Bacteria in Blood: Optimized Recovery of Bacterial DNA for Rapid Identification

Ryan Wood
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/8147

This Dissertation is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Bacteria in Blood: Optimized Recovery of Bacterial DNA for Rapid Identification

Ryan LaVar Wood

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

William G. Pitt, Chair
Jared P. Whitehead
Bradley C. Bundy
Douglas R. Tree

Department of Chemical Engineering
Brigham Young University

Copyright © 2020 Ryan LaVar Wood
All Rights Reserved
ABSTRACT

Bacteria in Blood: Optimized Recovery of Bacterial DNA for Rapid Identification

Ryan LaVar Wood
Department of Chemical Engineering, BYU
Doctor of Philosophy

Blood stream infections are challenging infections to rapidly diagnose. The current clinical diagnostic methods for blood stream infections require culturing the blood sample prior to identifying the bacteria and any resistance the bacteria may contain. Removing the culturing step from the bacterial identification process of a blood stream infection provides a significant reduction in the processing time. However, eliminating the culturing step shifts the difficulty from processing time to concentration, since clinical concentration levels can be as low as 10 CFU/mL in blood. This dissertation developed and evaluated many aspects of the process required to identify bacteria from a blood stream infection without culturing the bacteria. Two new methods of separating the bacteria from the blood cells were developed: inducing clotting using a centrifugal-sedimentation on a hollow disk, and filtering whole blood. Inducing clotting achieved 69% bacterial recovery from 7 mLs of whole blood in 117 s. Filtering whole blood achieved 100% bacterial removal from 5 mLs of whole blood in ≈ 90 s, but the bacteria were difficult to remove from the filter. Bacterial removal from the filter after blood filtration was also investigated. At a very low bacterial concentration of 200 CFU/mL, a blood lysis solution of 3% Tween 80 followed by a 3% Pluronic F108 backflush solution achieved 60% removal of the bacteria from the filter. In addition to developing two new methods, a previously developed technique using centrifugal-sedimentation on a hollow disk underwent a stability analysis in order to decrease the occurrence of mixing. This analysis yielded the development of the analytical solution to the Navier-Stokes equations for a two-fluid flow with a moving wall boundary and a free surface. The analysis also experimentally identified a stability boundary that was found to be in good agreement with the Kelvin-Helmholtz instability model. After exploring the methods to recover bacteria from blood, experiments were performed to identify a bacterial lysing solution that could lyse E. coli, E. cloacae and K. pneumoniae bacteria. The best bacterial lysing solution consisted of incubating the bacteria with 1 mg/mL lysozyme for 10 min followed by the addition of 6 M GHCl and 1% SDS. This solution obtained a 46% DNA recovery. The DNA were then fragmented by ultrasound to reduce the segment length for DNA labelling. In addition to lysing and fragmenting the DNA, a microfluidic device was prototyped and tested for incorporating the lysing, capturing, releasing, and fragmenting of the DNA all on a single device. Whole experiments were performed which extracted the bacteria from the blood, removed and collected the DNA from the bacteria, and fragmented the DNA. The best overall recovery from an experiment performing the whole process was 26.8%. The 26.8% recovery was achieved with a 68% recovery of the bacteria from spinning and a 54.1% removal of bacteria from off of the filter and a 72.9% recovery of the DNA from the bacteria.

Keywords: bacteria, blood, instability, filtering, DNA extraction, antibiotic resistance, microfluidic device
ACKNOWLEDGMENTS

I would like to thank Dr. William Pitt for his mentorship, guidance, and friendship while working on this project and studying here at Brigham Young University. Thank you for the many discussions we’ve had about research and life that have really taught me how to analyze and think about problems. Thank you for allowing me to be one of your students, I have enjoyed working with you on this project.

I would like to thank Dr. Jared Whitehead for invaluable help in analytically solving the Navier-Stokes equation for our given two layer flow. Without his knowledge and math checking, the analytical solution would still be unsolved and swamped with errors. I would also like to say thank you for your friendship and the many hours we spent together in meetings where you helped rekindle thirst for math knowledge.

I would like to thank my wife, Emma, and my sons, Theodore and Elliot, for their loving support throughout this entire process. I really appreciate the sacrifices that they made on my behalf to allow me to complete this project. There were many long nights and early mornings working on writing papers and debugging code that couldn’t have happened without your dedication, Emma. I love you all so much and am grateful to have you by my side. I would also like to thank my parents and siblings for your support throughout this project and my many years at school.

I would like to thank Dr. Julie Crockett for her insightful knowledge of fluid mechanics and instabilities. Without her direction on where to look, the instability analysis may not have succeeded. I would like to thank Dr. Richard Robison for his assistance and guidance on bacterial DNA extraction, and for the use of many of the machines and bacterial strains in his lab. I would like to thank Dr. Mark Clement for his expertise and guidance with machine learning. Without his knowledge, the 10mer DNA segment plasmid analysis with machine learning would have never made it over the first hurdle. I would like to thank Dr. Prashant Nagpal for the groundwork that he has laid down in using 10mer DNA segments for DNA analysis. Without our meeting at the
AICHE conference in 2017, using 10mer DNA segments for analyzing bacterial resistance would have never crossed my mind.

I would like to thank Olivia Brown for her efforts in supplying the project with bacteria, running qPCR and fluorometer, and for teaching me many of the important microbiologist techniques that I required to run this project. I would like to thank Dr. Radim Knob and Robert Hanson for their input of design changes and ease-of-use feedback with the microfluidic device, and in troubleshooting my process. I would like to thank Clifton Anderson for his help in troubleshooting many of my python codes and for sharing his knowledge of numerical methods and for his work on simulating the disk spinning process. I would like to thank Mahsa Alizadeh for her work on the design and testing of the disk for blood cell separation. I would like to thank Tanner Jensen and Cindy Wadsworth for their initial work on the machine learning technique for analyzing the 10mer DNA segments.

I would like to thank all of the students that helped conduct many of the experiments for this project, Daniel McClellan, Colin Bledsoe, Rae Blanco, Alex Hunter, Madison Wood, Clara Buchanan, Caroline Hickey, Alexandra Carter, Jacob Stepan, Cameron Beard, Evelyn Welling, Rebekah Torgesen, Magdalena Crofts, Nathan Bevan, Alayna Bedwell, Jordan Wood, Damian Teasley, Hope Callister, Alyson Ransom, Zach McAllister, and Dylan Howlett. This wouldn’t have gotten finished in a timely manner without their help.

This work was supported by the National Institute of Health grant number R01AI116989.
# TABLE OF CONTENTS

**LIST OF TABLES**  ............................................................... ix

**LIST OF FIGURES** ............................................................ xi

Chapter 1  Introduction ......................................................... 1

Chapter 2  Literature Review .................................................. 5
  2.1 Bacterial Extraction from Blood ......................................... 5
      2.1.1 Filtration .................................................. 6
      2.1.2 Centrifugation and Sedimentation ............................ 8
      2.1.3 Hydrodynamic focusing .................................... 11
      2.1.4 Chemical Capture of Bacteria from Blood .................. 12
      2.1.5 Field-Flow Fractionation and Dielectrophoresis ............ 14
      2.1.6 Outlook .................................................. 15
  2.2 Separation Instability .................................................. 17
      2.2.1 Outlook .................................................. 20
  2.3 Bacterial DNA Extraction ................................................ 20
      2.3.1 Mechanical Lysis .......................................... 21
      2.3.2 Electrical Lysis .......................................... 22
      2.3.3 Thermal Lysis ............................................ 22
      2.3.4 Chemical Lysis .......................................... 23
      2.3.5 Outlook .................................................. 23
  2.4 Bacterial DNA Collection ................................................ 24
      2.4.1 Density gradients ......................................... 24
      2.4.2 Electrophoresis .......................................... 25
      2.4.3 Organic extraction ....................................... 25
      2.4.4 Solid-phase Extraction ................................... 26
      2.4.5 Filtration ................................................ 27
      2.4.6 Electrostatic interactions ................................ 28
      2.4.7 Bead Adsorption ......................................... 29
      2.4.8 Outlook .................................................. 30
  2.5 Microfluidic Design .................................................... 30
      2.5.1 Micromixing .............................................. 30
      2.5.2 Microvalves .............................................. 33
      2.5.3 Outlook .................................................. 36
  2.6 Gene Identification ..................................................... 36
      2.6.1 Current Technique ......................................... 36
      2.6.2 qPCR ..................................................... 37
      2.6.3 in situ Hybridization .................................... 38
      2.6.4 DNA microarray .......................................... 38
      2.6.5 Sequencing ............................................... 39
      2.6.6 SERS ....................................................... 40
Chapter 3  Objectives  ......................................................... 42

Chapter 4  Experimental Methods ........................................... 44
  4.1 Bacterial Strains ......................................................... 44
  4.2 Bacterial Preparation and Growing .................................. 45
    4.2.1 Frozen Stocks .................................................. 45
    4.2.2 Stock Plates .................................................. 45
    4.2.3 Daily Growth .................................................. 45
  4.3 Measuring Bacteria ..................................................... 46
    4.3.1 Concentration Correlation .................................... 46
  4.4 Dilutions and Plating .................................................. 47
  4.5 Disk Spinning ........................................................ 47
    4.5.1 Disk and Blood Preparation .................................... 50
    4.5.2 Original Deceleration ......................................... 51
    4.5.3 Controlled Deceleration ....................................... 51
    4.5.4 Clotting Deceleration ......................................... 53
  4.6 Filtering .............................................................. 54
  4.7 Backflushing .......................................................... 55
  4.8 DNA Extraction Experiments ......................................... 56
    4.8.1 Ultrasound ...................................................... 57
  4.9 DNA Fragmentation ................................................... 58
    4.9.1 Gel Electrophoresis ............................................ 59
  4.10 DNA measurements ................................................... 60
    4.10.1 Fluorometer .................................................... 60
    4.10.2 qPCR ........................................................ 60
  4.11 Whole Process Experiments ......................................... 61
  4.12 Microfluidic Experiments ............................................ 62
  4.13 Statistics ............................................................ 62

Chapter 5  Bacterial Extraction from Blood  ................................ 63
  5.1 Spinning Disk ......................................................... 63
    5.1.1 Bacterial Separation through Clotting ....................... 71
  5.2 Filtering .............................................................. 72
    5.2.1 Plasma Filtering ................................................ 73
    5.2.2 Whole Blood Filtering ........................................ 75
    5.2.3 Removing Bacteria from Blood and Plasma .................. 83
  5.3 Backflushing .......................................................... 89
    5.3.1 Removing Bacteria from the Filter ......................... 96

Chapter 6  Separation Instability ........................................... 99
  6.1 Theoretical Velocity Profile ........................................ 99
    6.1.1 Derivation of 2-D Velocity Profile during Deceleration phase .. 102
  6.2 Linear Stability ....................................................... 108
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Disk Modeling</td>
<td>115</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Cell-pack layer</td>
<td>115</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Theoretical Velocity Profiles</td>
<td>119</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Instability Growth</td>
<td>120</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Onset of instabilities in a decelerating 2-fluid layer</td>
<td>124</td>
</tr>
<tr>
<td>6.4</td>
<td>Experimental Onset of Instability</td>
<td>126</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Onset of instabilities in a decelerating 2-fluid layer</td>
<td>129</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Clinical application and the experimental outcomes</td>
<td>131</td>
</tr>
<tr>
<td>7.1</td>
<td>DNA Extraction</td>
<td>134</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Bacterial Lysis for DNA Extraction</td>
<td>141</td>
</tr>
<tr>
<td>7.2</td>
<td>DNA Fragmentation</td>
<td>145</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Stir Bar</td>
<td>146</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Ultra-turrax</td>
<td>148</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Syringe</td>
<td>149</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Sonication</td>
<td>150</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Discussion</td>
<td>156</td>
</tr>
<tr>
<td>7.3</td>
<td>DNA Purification</td>
<td>157</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Capture and Release of DNA</td>
<td>159</td>
</tr>
<tr>
<td>7.4</td>
<td>Limits of Detection</td>
<td>161</td>
</tr>
<tr>
<td>8.1</td>
<td>Whole Process</td>
<td>164</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Discussion</td>
<td>166</td>
</tr>
<tr>
<td>8.2</td>
<td>Microfluidic Designs</td>
<td>168</td>
</tr>
<tr>
<td>8.3</td>
<td>Microfluidic Device Tests</td>
<td>171</td>
</tr>
<tr>
<td>8.3.1</td>
<td>DNA Recovery from Microfluidic Devices</td>
<td>172</td>
</tr>
<tr>
<td>9.1</td>
<td>Conclusion</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>181</td>
</tr>
<tr>
<td>A.1</td>
<td>Chapter 5 Statistics</td>
<td>215</td>
</tr>
<tr>
<td>A.2</td>
<td>Chapter 7 Statistics</td>
<td>224</td>
</tr>
<tr>
<td>B.1</td>
<td>Chapter 6 Mathematical Details</td>
<td>225</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Derivation of 2-D Velocity Profile during Spin Up phase</td>
<td>225</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Derivation of 2-D Velocity Profile during Hold phase</td>
<td>228</td>
</tr>
<tr>
<td>C.1</td>
<td>DNA Analysis</td>
<td>235</td>
</tr>
<tr>
<td>C.1.1</td>
<td>DNA Sequences</td>
<td>237</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>C.1.2 Generating Sequence-Specific FBC Spectrum</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td>C.1.3 Simulating Gene Mutations and Experimental Errors</td>
<td>241</td>
<td></td>
</tr>
<tr>
<td>C.1.4 Bias FBC Spectrum</td>
<td>243</td>
<td></td>
</tr>
<tr>
<td>C.1.5 PCA</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>C.1.6 Testing the MLAs</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>C.1.7 Model Performance Testing and Statistical Analysis</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>C.1.8 Simulation</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>C.2 Results</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>C.2.1 Principal Component Analysis</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>C.2.2 Classification Using MLA</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>C.3 Discussion</td>
<td>264</td>
<td></td>
</tr>
<tr>
<td>C.4 Additional Material</td>
<td>266</td>
<td></td>
</tr>
<tr>
<td>C.4.1 100 BOC reads</td>
<td>266</td>
<td></td>
</tr>
<tr>
<td>C.4.2 Plasmid NCBI Reference IDs</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>C.4.3 Additional PCA Figures</td>
<td>283</td>
<td></td>
</tr>
<tr>
<td>Appendix D Python Code</td>
<td>291</td>
<td></td>
</tr>
<tr>
<td>D.1 Chapter 6 Stability Codes</td>
<td>291</td>
<td></td>
</tr>
<tr>
<td>D.1.1 Velocity Profile Code</td>
<td>291</td>
<td></td>
</tr>
<tr>
<td>D.1.2 Non-Newtonian Modeling</td>
<td>301</td>
<td></td>
</tr>
<tr>
<td>D.1.3 Linear Stability Code</td>
<td>316</td>
<td></td>
</tr>
<tr>
<td>D.2 Appendix C Fractional Base Composition Codes</td>
<td>328</td>
<td></td>
</tr>
<tr>
<td>D.2.1 Complete Code</td>
<td>329</td>
<td></td>
</tr>
<tr>
<td>D.2.2 Confusion Matrix</td>
<td>390</td>
<td></td>
</tr>
<tr>
<td>D.2.3 G Plotting</td>
<td>399</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

2.1 Blood properties ................................................................. 6  
2.2 Sedimentation table ........................................................... 9  
2.3 Bacterial Separation ............................................................ 16

5.1 3000 rpm Clotting Results .................................................... 65  
5.2 4000 rpm Clotting Results .................................................... 67  
5.3 Fluid Configuration Clotting Results ....................................... 69  
5.4 ADP Clotting Results ............................................................ 70  
5.5 Alternate Disks Clotting Results ............................................. 72  
5.6 High SDS Plasma Filtering ..................................................... 73  
5.7 Low SDS Plasma Filtering ..................................................... 75  
5.8 Saponin Plasma Filtering ....................................................... 75  
5.9 Whole Blood SDS Filtering .................................................... 76  
5.10 Whole Blood Saponin Filtering ............................................... 76  
5.11 RBC Fixation Filtering through small pores ............................... 77  
5.12 RBC fixation filtering through large pores ................................ 78  
5.13 Whole blood filtering - different Triton X-100 concentrations .... 79  
5.14 Whole blood filtering - different salt concentrations ............... 80  
5.15 Whole blood filtering - acidic, basic, and neutral salt .............. 81  
5.16 Whole blood filtering - high hematocrit ................................ 82  
5.17 Whole blood filtering - varied hematocrit ............................... 83  

6.1 Fitted parameters for $\tau_C$ and $\mu_C$ ......................................... 117  
6.2 Blood parameters for modeling .............................................. 119  
6.3 Instability growth rate table ................................................. 123

7.1 List of bacterial lysis chemicals ............................................. 135  
7.2 Literature lysis - no lysozyme ................................................. 139  
7.3 Literature lysis - lysozyme .................................................... 140  
7.4 List of Lysis Solutions .......................................................... 142  
7.5 Sonication in Plastic Labels ................................................... 152  
7.6 Concentration of DNA in Different Concentrations of Bacteria .... 162

A.1 Clotting T-test Results-3000 rpm ........................................... 215  
A.2 Clotting T-test Results-4000 rpm ........................................... 216  
A.3 Clotting T-test Results-Layout .............................................. 217  
A.4 Clotting T-test Results-ADP .................................................. 217  
A.5 Clotting T-test Results-Disk Design ........................................ 218  
A.6 SDS Bactericidal T-test Results ............................................. 220  
A.7 Lysozyme T-test Results ....................................................... 224

C.1 Bacterial genomes .............................................................. 239  
C.2 Antibiotic Resistant Plasmids ................................................ 240  
C.3 Bacterial Species MLA .......................................................... 251
### LIST OF FIGURES

2.1 Bacteria separation disk ........................................ 17
4.1 Velocity profile for spinning disk ............................. 51
4.2 Layout for filming deceleration experiments ................. 53
4.3 Blood component separation instability ...................... 54
5.1 3000 rpm clotting results .................................... 65
5.2 4000 rpm hold speed variation ............................... 66
5.3 Fluid layout configuration experiments ...................... 68
5.4 Clotting with ADP ............................................. 70
5.5 Bactericidal effects of SDS ................................... 74
5.6 Backflush experiments with different bacterial suspension fluids ......................................................... 91
5.7 Backflush experiments with different blood components ................................................................. 92
5.8 Backflush Experiments with Hydrophilic filters ............ 94
5.9 Backflush Experiments with Hydrophobic filters .......... 95
5.10 Low Count Backflush Experiments ......................... 96
5.11 Low Count Backflush Experiments with Salt Variations 97
6.1 2-layer flow configuration ..................................... 100
6.2 Fitting the hematocrit dependence for the Casson viscosity 117
6.3 Fitting the hematocrit dependence for the critical yield stress ......................................................... 118
6.4 Deceleration velocity profile modeling ...................... 120
6.5 Mixing and Kelvin Helmholtz plots .......................... 122
6.6 Instability plots for different hematocrits ................... 127
6.7 Proposed disk deceleration profile ............................ 128
7.1 DNA extraction without lysozyme .............................. 136
7.2 Testing lysozyme parameters .................................. 137
7.3 DNA extraction with lysozyme ................................ 138
7.4 DNA extraction all bacteria comparison .................... 141
7.5 DNA Fragmenting - Stir bar .................................. 147
7.6 DNA Fragmenting - Stir bar with Microspheres ............ 147
7.7 DNA Fragmenting - Ultra-turrax .............................. 148
7.8 DNA Fragmenting - Needle .................................... 149
7.9 DNA Fragmenting - Sonication in Microcentrifuge Tubes 151
7.10 DNA Fragmenting - Sonication through Plastic .......... 153
7.11 DNA Fragmenting - Fluorescence Intensity ............... 154
7.12 DNA Fragmenting - Fragment Size .......................... 155
7.13 Magnetic Bead Experiments .................................. 158
7.14 DNA Recovery based on size ................................ 159
7.15 Size Based DNA Recovery Gel ................................ 160
7.16 Fluorometry Limit of Detection ............................... 162
7.17 qPCR Limit of Detection ..................................... 163
8.1 Whole Process Overview .................................................. 165
8.2 Whole Process Filter and DNA Recovery .............................. 166
8.3 Overall Recovery for Whole Process ................................. 167
8.4 Microfluidic device template ............................................ 168
8.5 Microfluidic devices ....................................................... 169
8.6 Microfluidic device comparison ........................................ 173
8.7 Microfluidic device to benchtop comparison .......................... 173
B.1 Time to uniformity plot for velocity hold ......................... 234
C.1 10mer DNA Analysis Overview ........................................ 238
C.2 10mer FBC Spectra ....................................................... 242
C.3 Bacterial Species PCA .................................................... 249
C.4 Antibiotic Resistance Genes PCA ...................................... 250
C.5 Species MLA ............................................................... 253
C.6 Species Taxonomy ........................................................ 254
C.7 Species Confusion Matrix ................................................ 255
C.8 Antibiotic Resistance Gene Individual Accuracy .................... 258
C.9 Antibiotic Resistance Gene Group Accuracy ......................... 261
C.10 Antibiotic Resistance Gene Individual Confusion Matrix ......... 262
C.11 Antibiotic Resistance Gene Group Confusion Matrix .............. 263
C.12 10 randomly sampled 100 BOC reads ............................... 267
C.13 100 BOC reads - bias comparison ................................... 267
C.14 Summed 100 BOC reads - bias comparison .......................... 268
C.15 Species PCA - 10^2 BOC Reads ....................................... 283
C.16 Species PCA - 10^3 BOC Reads ....................................... 283
C.17 Species PCA - 10^4 BOC Reads ....................................... 284
C.18 Species PCA - 10^5 BOC Reads ....................................... 284
C.19 Species PCA - 10^6 BOC Reads ....................................... 285
C.20 Individual Plasmid PCA - 10^2 BOC Reads .......................... 285
C.21 Individual Plasmid PCA - 10^3 BOC Reads .......................... 286
C.22 Individual Plasmid PCA - 10^4 BOC Reads .......................... 286
C.23 Individual Plasmid PCA - 10^5 BOC Reads .......................... 287
C.24 Individual Plasmid PCA - 10^6 BOC Reads .......................... 287
C.25 Group Plasmid PCA - 10^2 BOC Reads ............................... 288
C.26 Group Plasmid PCA - 10^3 BOC Reads ............................... 288
C.27 Group Plasmid PCA - 10^4 BOC Reads ............................... 289
C.28 Group Plasmid PCA - 10^5 BOC Reads ............................... 289
C.29 Group Plasmid PCA - 10^6 BOC Reads ............................... 290
CHAPTER 1. INTRODUCTION

With antibiotic resistant bacteria on the rise, patient treatment costs, hospitalizations, and mortality rates are increasing [1–3]. Furthermore, few new antibiotics are in development to treat these pathogens, leaving clinicians with an ever-diminished arsenal against these increasingly resistant opponents [4, 5]. Gram-negative pathogens are of particular concern, given their ease of resistance-gene acquisition, especially bacteria from the family Enterobacteriaceae due to the increase in the generation and spread of carbapenem-resistant strains [6, 7]. The carbapenem class of antibiotics is used as a last-resort antibiotic [8, 9]. Thus, carbapenem-resistant Enterobacteriaceae (CRE) are among some of the deadliest bacterial infections with mortality rates as high as 50% [8, 10]. Bacteremias (bloodstream infections) are of great concern, because of the difficulty in rapidly identifying the bacteria and any resistance they may contain. The difficulty in diagnosis significantly delays the start of appropriate treatment for bacteremias.

Current standards of bacteremia identification are based on culturing the blood and subsequently performing phenotypic assays which require at least 12-24 hours, and often longer, between sample acquisition and assay reporting [11]. This is not rapid enough to successfully treat CRE blood infections, where survival rates drop by as much as 7.6% per hour as effective treatment is delayed [12, 13]. In addition to the long bacteremia identification time, clinicians incorrectly diagnose bacteremias 25-33% of the time [14–16] which can lead to serious and often fatal consequences. It is critical that clinicians can rapidly acquire detailed information about both the infecting bacterium and its resistance profile to determine the most effective antibiotic regimen shortly after clinical suspicion of infection, which would lead to improved clinical outcomes. Many studies have shown that mortality rates decrease when timely administration of the correct antibiotics occurs for bloodstream infections [17–21]. However, without detailed information on both the infecting bacterium and its resistance profile, clinicians cannot reliably determine the most effective antimicrobial treatment.
Bacteremias may become a threat at concentration levels as low as 10 colony forming units (CFU) per mL [15, 22–24]. Currently, these low values require the time-consuming step of sample culturing before antibiotic resistance and bacterial type can be determined. In addition to the need for sample culturing, many of the current analysis methods struggle with the presence of non-bacterial material, such as blood cells, proteins and even the DNA from white blood cells (WBCs) [25]. Identifying the bacterial species and whether any resistance to antibiotics is present are challenging tasks at best.

This dissertation is part of a collaborative effort to develop a process and device which will identify bacteria and any antibiotic resistance from a blood stream infection within an hour from blood collection from a patient. Once the process and device are developed, they will be commercialized for use in hospitals and clinics around the world. The process includes separating the bacteria from the blood, lysing the bacteria, collecting the genomic and plasmid DNA from the bacteria, labelling the DNA with a fluorescent marker, and detecting the labelled DNA. Ideally, this process is performed upon a single device but may be performed upon two or three devices. Processes have already been developed for separating the bacteria from the blood, labelling the DNA with a fluorescent marker, and detecting the labelled DNA.

The process for separating the bacteria from the blood includes two techniques: a new blood filtering technique developed as part of this dissertation and a previously developed blood spinning technique [26]. These techniques are compared later by their efficiency in separating out the red and white blood cells (RBCs, WBCs) from the bacteria in a given amount of time. Filtering the blood requires the blood to be diluted in order to prevent filter clogging, and thus takes longer to filter a set amount of blood. Spinning the blood in a hollow disk allows for quick separation without the need for dilution, but stopping the disk from high speeds without remixing the plasma and cells is a challenge. The previously developed disk setup involved spinning a 12-cm-diameter disk at 3000 rpm for 54 seconds. This induces a centrifugal sedimentation which causes the RBCs and WBCs to spin out of the plasma and into a trough that holds the separated RBCs and WBCs. The disk is then slowed to a stop in about 3 minutes, which allows the separated plasma layer to be collected for further analysis. The disk process recovers about 70% of the bacteria found in the blood [26, 27].
The process for labelling the DNA includes flowing the released bacterial DNA through a monolith which contains complimentary, single-stranded DNA containing sequences for bacterial and antibiotic resistance identification. Multiple fluorescent molecules are then attached to the strands of DNA which have been captured by the monolith. A different fluorescent molecule is attached to each of the different types of bacteria and antibiotic resistance sequences on the monolith. Then the DNA is released for detection. The complimentary, single-stranded DNA are approximately 100 base pairs (bp) long [28]. Due to the shortness of the DNA strands on the monolith, the bacterial DNA needs to be broken down into 1-2 kbp segments in order to ensure that the monolith can capture the desired sequence.

The process for detecting the DNA includes directing a laser to the DNA to excite the fluorescent molecules while flowing the labelled DNA through a wave guide channel for detecting the light emitted from the fluorescent molecules. The wave guide channel is designed on a silicon wafer to convey the DNA, one piece at a time, through the interrogating laser beam, which creates a specific patterning for the different types of fluorescent molecules in the wave guide channel. A computer processes the pattern output from the wave guide channel allowing for the identification of the bacteria species and any antibiotic resistance genes present in the bacteria [29].

The scope of this dissertation includes 1) providing a stability analysis of the spinning disk to ensure that the RBCs and WBCs do not remix into the plasma while slowing the disk to a stop; 2) testing new bacterial separation techniques for comparison to the prior disk spinning technique; 3) isolating the bacteria from the spun plasma; 4) lysing the collected bacteria; 5) collecting the released genomic and plasmid DNA; 6) removing contaminants from the collected DNA; 7) releasing the collected DNA; 8) breaking down the DNA into the desired size range of 1-2 kbp for fluorescent labelling on the monolith; and 9) designing a microfluidic device which includes all processes after isolation of the bacteria from the plasma, including the labelling of the DNA on the monolith and detecting the labelled DNA using the wave guide channel. At the onset of this dissertation, the company in collaboration with the project already had a machine in hospitals that used injection-molded polypropylene cartridges for diagnostics. Therefore, the initial designs and prototypes of the microfluidic device were constrained to work on the existing cartridge. Shortly before the completion of my dissertation, the collaborating company changed and the constraints to the microfluidic device were removed. Due to time and material constraints,
new microfluidic device designs were not prototyped and tested but are included for future work. Due to the low number of pathogens in blood infections, the efficiency of the bacterial lysis, DNA collection, and DNA elution determines the amount of DNA that can be labelled and identified. It is important to obtain as high of a recovery of the DNA as possible in order to ensure that the bacteria and antibiotic resistance can be accurately identified.

This dissertation is organized as follows: Ch. 2 contains a literature review for each of the main aspects of this project, Ch. 3 presents the objectives, Ch. 4 presents the procedures used during the experiments, Ch. 5 discusses bacterial extraction from blood, Ch. 6 investigates the instabilities seen while separating bacteria from blood cells on the spinning disk, Ch. 7 discusses the extraction and collection of DNA from the bacteria, Ch. 8 discusses the microfluidic design for accomplishing the techniques presented in Ch. 7, and Ch. 9 concludes with the highlights from each chapter and future work.
CHAPTER 2. LITERATURE REVIEW

2.1 Bacterial Extraction from Blood

Blood is a very complex fluid. Table 2.1 presents some basic facts about blood components that are relevant to this literature review. Blood can be divided into plasma and cells, with the cells constituting between 40-55% of the blood volume [30]. The cellular components can be divided into three main components: erythrocytes or red blood cells (RBCs), leukocytes or white blood cells (WBCs), and platelets. RBCs make up > 99% of the volume of the cellular blood components (~ 5 × 10^9 RBCs per mL of whole blood); the rest of the blood components are a mixture of the WBCs, platelets, and in the case of disease, any contaminating cells or pathogens [30–34]. The threshold of contaminating cells or pathogens in blood that produces clinical symptoms differs for each patient and infection, but sepsis has been shown to be a threat at levels as low as 10 bacteria colony-forming units per milliliter (CFU/mL) [15, 22–24]. The extremely high concentration of RBCs is possibly the greatest challenge when separating bacteria from blood, since RBCs can outnumber the bacteria by nearly one billion to one. Another challenge is the ease at which platelets are activated, resulting in aggregates with each other or adherence to any surfaces. Platelets disabled by calcium-chelator anticoagulants, such as citrate or ethylenediaminetetraacetic acid (EDTA), can still stick to some surfaces [35]. In addition, anticoagulants can interfere with downstream processing of DNA, such as PCR [25]. Thus a technique needs to be able to not only deal with the high blood cell concentrations, but not interfere with any later processes wherein the pathogen and any antibiotic resistance is identified. An in depth review of blood separation techniques has been written by Pitt et al [7].
Table 2.1: Some properties of human blood components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Concentration [#/mL]</th>
<th>Size [µm]</th>
<th>Density [g/mL]</th>
<th>DNA</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red blood cell</td>
<td>4.0 - 5.7 x 10⁹</td>
<td>6.2 - 8.2</td>
<td>1.086 - 1.122</td>
<td>Generally no</td>
<td>[31]</td>
</tr>
<tr>
<td>White blood cell</td>
<td>4 - 11 x 10⁶</td>
<td>7 - 30</td>
<td>1.057 - 1.091</td>
<td>Yes</td>
<td>[32, 33]</td>
</tr>
<tr>
<td>Platelets</td>
<td>1.3 - 4.0 x 10⁸</td>
<td>2.0 - 4.0</td>
<td>1.072</td>
<td>No</td>
<td>[32]</td>
</tr>
<tr>
<td>Plasma</td>
<td>—</td>
<td>—</td>
<td>1.024</td>
<td>May contain DNA</td>
<td>[34]</td>
</tr>
</tbody>
</table>

2.1.1 Filtration

A search of the literature revealed no publications in which bacteria were directly separated from blood by filtration processes without altering the blood cells. However, there were several studies and good reviews [36, 37] describing the successful separation of blood components from each other. Separation by filtration is primarily based upon size. The greater the size differential between the particles to be separated, the easier it is to separate the particles. However, there are many other properties that can affect the separation, such as the aggregation of particles, adhesion to the filter and its pores, how particles pack with each other in a filter cake, their mechanical compliance (shape change under pressure), their interaction with the suspending liquid, and the viscosity of the suspending liquid [7].

Using a filter or series of filters, called screening, could be employed to separate suspended bacteria from blood. The pores of the filter would need to be sized to allow bacteria to pass, but prevent the passage of blood cells. Screening could be an effective technique since multiple filters are used and the diameter of the pores could progressively get smaller. However, once the red cells begin to get trapped by a filter, they can compress and pack together, creating a filter cake so tight there is nearly zero porosity [7, 38]. Thus, bacteria will have difficulty in finding a connected passage large enough to squeeze through. Additionally, filtering and screening will only work if the bacteria passing through the pores is sufficiently smaller than the retained blood cells, but the particle sizes between the bacteria and the blood components are relatively the same (see Table 2.1), especially between bacteria and platelets. The similar size ratio between bacteria and platelets, would be like separating 0.9 µm and 1.0 µm hard spherical particles from each other,
which has been done but in dilute flow [39]. With a normal blood hematocrit near 45%, filtering whole blood is not dilute enough to separate out the bacteria from the platelets. The main problem is that the blood cell filter cake will trap all bacteria that are not located right next to the filter pores at the commencement of filtering [7, 38]. Diluting the blood slows the build-up of the filter cake, but also dilutes the bacteria and increases the amount of fluid needed to be processed. Slow flow may also be used to slow the build-up of the filter cake, but neither dilution nor slow flow allow for rapid recovery of the bacteria [7].

Liu et al. separated white blood cells (WBCs) from red blood cells (RBCs) and platelets in diluted (1:20) blood by screening with filters of decreasing pore sizes [40]. A few other papers used arrays of posts and channels with gradients in size or spacing to separate WBCs from the other blood components (RBCs, platelets, and plasma) by redirecting the WBCs while allowing the other components to flow through [41–43]. However, these techniques separate out a larger particle rather than a smaller particle, and with the size limitations and filter cake, designing a filter or screen that can separate out bacteria from RBCs and platelets may be impossible.

Cross-flow filtration is a potential alternative to dead-end filtration [7]. In cross-flow filtration, there are two fluid flows perpendicular to each other. One fluid flows parallel to the filter surface to move the excluded particles away from the filter surface to keep them from forming a filter cake, and the other fluid flows through the filter carrying the small particles through the filter [44]. This process has been used in microfluidics to separate out blood cells from plasma [45] and WBCs from RBCs [46]. However, no reports of separating bacteria from blood by cross-flow filtration have been found in the literature. This may be because the same challenges exist, like the compressibility of RBCs and the small size of platelets. According to Pitt et al., this process might be feasible if the blood cells were rigid and the blood diluted [7]. A fixative could be added to the solution to create rigid RBCs, but this may also have unintended consequences with the bacteria present. Thus, cross-flow filtration may not be the best choice.

Another alternative to the previous types of filtration is to remove the WBCs, RBCs, and platelets by lysing them in solution and then just filtering out the bacteria from the cell-membrane debris and other cellular constituents. The initial studies performed separating bacteria from blood by filtration diluted the blood 1:20 in a Triton X-100 with sodium bicarbonate solution and then back flushed the bacteria into a culture bottle for identification [47, 48]. Zierdt et al began incorpo-
rating proteases along with the Triton X-100 and sodium bicarbonate solution for better filtration, but noticed that the solution was toxic to some species of bacteria if not filtered within a few minutes [49]. Tween 20 was then suggested as a replacement for the Triton X-100 and sodium bicarbonate combination due to its decrease in toxicity [50,51] and was verified clinically [52,53]. The use of a lysis-filtration method followed by filter incubation on agar plates was shown to be faster and just as sensitive as the standard blood culturing bottles [54–57]. Three recent articles applied this technique using Brij 97 to lyse the blood cells, but they performed the lysis step after growing the bacteria in blood culture bottles and removed the bacteria by scraping the filter [58–60]. Although this technique has always been combined with a culturing step, it has great potential for quickly separating out the bacteria from the blood cells, since culturing is skipped in our process. However, there are two major concerns with this process: first, the adherence of bacteria to the filter surface [61] and second, the large waste volumes since dilution is always used in the process. A great review on the factors that influence the filtering of bacteria [62] helps address the first concern. This process is investigated in Chapter 5.

2.1.2 Centrifugation and Sedimentation

Centrifugation is defined as the process of generating a centrifugal force when rotating a sample, which force operates on the particles and fluid. Sedimentation is defined as the movement of particles through a fluid in the presence of an external field. Centrifugation is often a convenient method to produce sedimentation, especially for small particles. Due to differences in properties, particles may sediment at different velocities allowing for separation to occur under the right conditions. The theoretical equation for sedimentation velocity, $v_s$, for small rigid spheres in Newtonian fluids is given by

$$v_s = \frac{D_p^2(\rho_p - \rho_f)(R\omega^2 + g)}{18\mu},$$

(2.1)

where $\rho_p$ is the particle density, $R$ is the rotational radius, $\omega$ is the rotational angular velocity, and $g$ is the gravitational constant [63]. This equation overestimates actual particle sedimentation velocities because the equation does not take into account interparticle interactions (such as aggregation, collisions, and hydrodynamic interactions) and non-spherical shapes. In addition to having particle interactions and non-spherical shapes, RBCs are compressible and blood is a non-Newtonian
fluid [7]. Nevertheless, a good first order estimation of sedimentation velocities is obtained by Eq 2.1 because blood plasma at low shear rates is nearly Newtonian [34] and the correction factors for spheroids are not large [63]. However, particle-particle interactions can produce a significant effect [64].

Clinically, blood is often separated into its components (RBCs, platelets and WBCs, and plasma) by centrifugation, but this process sediments the components until equilibrium is achieved, through long centrifugation times. At equilibrium, the components are separated into layers according to their density. This process will not work for separating bacteria from RBCs, due to the similarity of their densities (see Table 2.2). No reports of separating bacteria from blood by equilibrium centrifugation of blood in tubes have been found in the literature. Although equilibrium centrifugation will not work, Table 2.2 shows that the cells in blood have different sedimentation velocities from each other and from bacteria. Therefore, the principles of sedimentation velocity may be used to separate bacteria.

Table 2.2: Relative sedimentation properties of blood components and bacteria.**

<table>
<thead>
<tr>
<th>Component</th>
<th>Density range (g/cm$^3$)</th>
<th>Density used for calculation (g/cm$^3$)</th>
<th>Nominal size used for calculation ($\mu$m)</th>
<th>Relative Velocity*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red blood cell</td>
<td>1.086 - 1.122</td>
<td>1.098</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>White blood cell</td>
<td>1.057 – 1.091</td>
<td>1.092</td>
<td>15</td>
<td>96</td>
</tr>
<tr>
<td>Platelet</td>
<td>1.072</td>
<td>1.077</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Bacteria</td>
<td>1.08 - 1.10</td>
<td>1.095</td>
<td>1.5</td>
<td>1</td>
</tr>
</tbody>
</table>

* Relative to $E.\ coli$ (modeled as 1.095 g/cm$^3$, 1.5-\(\mu\)m sphere) using the nominal values in the table. ** This table is a modified version of Table 2 from [7].

One possibility, presented by Pitt et al, is the use of a separator tube, although no such reports for bacterial separation have been found in the literature. A separation tube contains a gel that has a density between that of plasma and cellular components [7]. The gels used in separation tubes usually have thixotropic viscosity, a viscosity that decreases under shear stress and returns to its initial state when the shear stress abates [65, 66]. One could then design the viscosity, density, and volume of the thixotropic gel such that it creates a layer between the plasma and RBCs after the
last sedimenting RBCs are below the gel when it completes the plug formation [7]. Because of the sedimentation velocity differences, the plug formation could be completed before an appreciable fraction of bacteria has been caught under the plug. However, this technique would not remove the majority of the platelets because of their similar sedimentation velocity to bacteria and has not been studied experimentally or theoretically.

A second possibility is the use of a microfluidic device. There have been several reports of microfluidic centrifugal separation processes in which microliter quantities of blood are placed within a chamber built into a disk the size of a compact disk (CD) and spun [67–69]. Gorkin et al have written a good review of blood separation devices on a CD size scale [69]. However, these processes have yet to be used to separate bacteria from blood. These processes invoke equilibrium centrifugation and would thus need some adaptations to be made in order to take advantage of the sedimentation velocity differences. Haeberle et al. designed a device using the centrifugal force to push blood past a hydrophobic stop into a carefully-sized cell chamber with a spill over chamber for the plasma [67]. Amasia et al. developed a spinning disk with a capillary duct drain channel that siphons the plasma into a collection chamber [68]. Both of these devices could work by controlling the timing and centrifugation of the disk. However, the Haeberle et al. device handles only 5 µL of whole blood [67] while the Amasia et al. device can handle 2 mLs of whole blood [68]. Being able to process the necessary 7-10 mLs of whole blood may be a problem for these devices. There is also the issue that platelets sediment at a similar rate to bacteria and would remain in the plasma with the bacteria.

A third possibility is the use of a lysis centrifugation method similar to that of the lysis filtration method mentioned above. Lysis centrifugation involves lysing the blood cells and then centrifuging the lysis mixture to be left with a pellet of bacteria along with cell-membrane pieces and other cellular constituents [70–75]. The studies that used this method would then streak the resulting centrifuge pellet onto several different types of agar. It has been shown that this is a much quicker form of detection than conventional blood culturing, but it still takes 16-24 hours for pathogen detection [70, 71, 73, 75]. But only the lysis centrifugation step is needed for pathogen separation from the blood cells. However, there could be residual human DNA and/or unlysed cells mixed in with the centrifuge pellet, which would require some kind of cleaning step before the bacteria could be broken open to retrieve their DNA.
A fourth possibility is the use of a hollow disk for bacterial separation that uses centrifugal separation. Our lab has been developing a 12-cm-diameter hollow disk that contains an RBC separation channel for separating out the unwanted blood cells from the plasma [7, 26, 76]. Using this device, 7 mLs of blood can be processed in one spin. This device takes advantage of large circumferential area that the hollow disk creates. When the device is spun, the blood is flung out against the outer wall creating a blood layer that is only about 3 mm thick. This means that the separation of the plasma and cells happens rapidly since the cells do not travel very far. Then the disk can be slowed to a stop, resulting in the RBCs being trapped in the separation channel while the plasma spills over the channel wall to be collected. While the disk can be spun up quickly to high rpms (≥ 3000 rpm) facilitating a very rapid separation of the cells and plasma (∼ 16 – 20 seconds depending on hematocrit), the disk must be slowly decelerated to prevent the separated layers from mixing. With the initial design and timing, the disk could achieve 70% bacterial recovery from 7 mLs of blood in 3.5 minutes [26,76] with concentrations as low as 6 CFU/mL [27]. Many aspects of this device are still under investigation and are presented in Chapters 5 and 6.

2.1.3 Hydrodynamic focusing

Hydrodynamic focusing is a technique which takes advantage of the secondary flows that can develop in channels during laminar flow to separate out particles. Flows and velocities are usually parallel with the channel walls, but channels that travel in a circular fashion or contain non-Newtonian fluids generate secondary flows toward or away from the walls of the channel. Depending on the properties of the particles, these secondary flows will have more affect upon some particles by moving them closer or further away from the walls than other particles. This allows for the separation of different types of particles in a multi-particle fluid, like bacteria in blood. These secondary flows, especially Dean’s flow [77,78], have been used to separate bacteria from blood cells [79].

Hou et al. developed a device which concentrated RBCs into the center of a 20 μm x 20 μm square channel, which pushed the bacteria towards the walls, and then expanded the channel into a 200 μm x 20 μm channel with side-stream bifurcations on each side to collect the bacteria [80]. This device collected about 80% of the bacteria, along with 80% of the WBCs and platelets, in the side-stream bifurcation. They demonstrated the applicability of their device to clinical samples.
by building and testing a 16-channel platform that processed 1.5 mL/min of blood and reduced bacterial concentration in the blood by >50% in 1 hr [81]. Mach et al. designed a device that focused the RBCs near the wall and then split off the RBCs in a bifurcation channel [82]. This device removed about 88% of the RBCs, recovered about 80% of the bacteria, and could be scaled to process 8 mL/min of diluted blood with a 1/200 dilution in a saline solution.

Wu et al. created a device in which diluted blood was sheathed with another flow and then deflected by a cross flow which caused blood cells to be deflected further from the center flow than the bacteria, leading to a separation [83]. They achieved a 300-fold enriched bacterial concentration in the collecting chamber, but diluted blood 1 to 1000 and had a whole blood flow rate of 18 µL/min. Hou et al. designed a similar device using a sheathed, diluted blood flow but employed Dean secondary flows that could separate and recover bacteria even at clinically relevant concentrations of 10-50 CFU/mL [79]. However, this device required a sheathing flow of 33 mLs for each mL of whole blood and processed whole blood at a rate of 50 µL/min.

While many of these devices are promising, most of them require blood dilutions that result in copious amounts of waste fluid that must be dealt with. Also, the time frame in which the devices could process the necessary amount of whole blood is not currently within the desired time constraints for rapidly identifying blood infections. But, just like our hollow disk device, many of these devices are still being investigated and improved.

2.1.4 Chemical Capture of Bacteria from Blood

The governing principle for chemical capture is the attachment of a substrate to a bacterium by either physical or chemical interactions, such that the bacterium-substrate complex is more easily separated from the blood than the bacterium alone. Any type of physical separation could then be used to separate out the bacteria-substrate complex from the blood. Magnetic separation is most common, but sedimentation or filtration could be used as well. The main requirement for chemical capture of an unidentified bacterium is that the capturing agent be general enough to attach to every type of bacteria but specific enough that it doesn’t bind to any of the other cells or proteins in the blood. While there is no known chemical group which is only found on all species of bacteria but not on mammalian cells, humans possess an immune protein called mannose binding lectin (MBL) which does bind all of the clinical pathogens of interest [84–86]. MBL has
been used to capture and separate bacteria from blood [87, 88]. An alternative to MBL could be lipopolysaccharides (LPS) found on the outer membrane of gram-negative bacteria [89], but this is still not general enough to collect gram-positive bacteria or mycobacteria.

The most common substrate for chemical capture of bacteria from blood is surface-modified superparamagnetic (SPM) beads. SPM beads are only magnetic when in the presence of a strong magnetic field. This allows the beads to be mixed in the solution with the bacteria without the beads sticking together; then after the allotted incubation time, a strong magnet can be applied to collect all of the beads in one location. These beads can be purchased with a surface coating that allows for easy attachment of biomolecules [90]. Sen et al. used SPM beads coated with binding ligands to capture *E. coli* from soil [91]. Kang et al. developed SPM beads coated with the Fc fragment from an IgG antibody attached to the binding end of MBL [87, 88, 92]. These beads obtained about 90% clearance of *E. coli* and *S. aureus* from whole blood at concentrations as low as $10^4$ CFU/mL. Lee et al. used SPM beads coated with bis-Zn-dipicolylamine to remove *E. coli* from diluted bovine blood [93]. At a concentration of $10^7$ CFU/mL, blood diluted 1 to 50, and a flow rate of 1 mL/min, the system removed about 80% of the bacteria in the first pass and 90-95% in the second pass. However, one challenge with SPM beads is the small size of SPM beads, usually between 100-500 nm, which cannot generate a sufficient force to move a bacteria through a fluid with a single bead [94]. Thus tens or hundreds of beads are needed to move the bacteria to the target region [87].

An alternative to using beads is to chemically capture bacteria onto a fixed surface. Investigators in Japan have clinically used a column of fibers coated with polymixin B, which binds to the LPS of gram-negative bacteria, to treat patients with septic shock [95, 96]. The blood passes through the column while the column binds the endotoxins released by the bacteria but not the bacteria themselves. Similarly, proinflammatory cytokines were removed from septic blood in a heparin-coated fiber column [97]. Mattsby-Baltzer et al. used a column of heparin-coated beads to successfully remove about 60% of *S. aureus* from spiked blood at a concentration 5 x $10^6$ CFU/mL [98]. Wang et al. was able to capture *E. coli* on a glass surface at a concentration as low as 50 CFU/mL using antibodies against the LPS binding protein which they mixed into the blood to capture the bacteria [89]. They achieved about a 70% binding efficiency, but the process required 3 hours to bind the *E. coli* and wash out the blood.
Similarly to the microfluidic and hydrodynamic devices, the chemical capture setups struggle with low bacteria concentrations or need long incubation times. Some even require significant dilutions to work. Also, none of the papers mention releasing the bacteria after capture, and while a high binding affinity is great for the capture of a bacterium, it presents a challenge for the release of a bacterium. However, if the substrate does not need to be removed for downstream processing, then tight binding is an advantage since it often facilitates more rapid adhesion and/or association. In their current state, these systems do not meet the necessary requirements for rapidly processing large volumes of blood with clinically relevant bacterial concentrations. But with continued research and improvements, these processes could become viable solutions.

2.1.5 Field-Flow Fractionation and Dielectrophoresis

Field-flow fractionation (FFF) flows suspended particles laminarly through a chamber and subjects them to some kind of force transverse to the flow direction. This causes the particles to migrate transversely in the channel with the amount of migration dependent on the particle properties. Common transverse forces are cross-flow fluid movement causing drag force, sedimentation force, thermophoretic force, electrical, magnetic and dielectric forces [99]. FFF separation has been used to separate blood components from each other [100–102], and separate different types of bacteria from each other [103–107], but no reports of FFF being used to separate bacteria from blood have been found in the literature. Since sedimentation forces can dynamically separate blood cells from bacteria, it may be possible to apply sedimentation-FFF to remove RBCs and WBCs from bacteria. However, blood separation involving FFF is usually diluted with saline or a buffer, which may create a challenge to rapidly process large volumes of blood.

Dielectrophoresis places polarized particles (no net charge) in a non-uniform electric field which then gain an electrokinetic force and move through the surrounding fluid creating separation. Thus differences in size, shape, density and dielectric properties of various particles can be exploited to create a separation. Cheng et al. separated *S. aureus* and *P. aeruginosa* from blood cells by migrating the bacteria to the center of a chamber using a quadruple electrode array [108, 109]. The blood cells migrated away from the center of the chamber. This was done in a non-flowing system with about 1/500 diluted blood and bacteria on the order of $10^7$ CFU/mL. In an initial study Cheng et al. was able to separate bacteria from blood cells using a flowing system op-
erated at 1 μL/min [108]. This system used a slightly higher blood dilution, 1/2000, and a slightly lower bacterial concentration, 10^6 CFU/mL.

Piacentini et al. used a combination of dielectrophoresis and FFF to separate platelets from RBCs in 1/10 diluted blood [101]. This was performed in a microfluidic channel using an alternating array of electrodes. This technique should be able to similarly separate bacteria from RBCs, with the possibility of also separating the bacteria from the platelets.

2.1.6 Outlook

There have been many techniques that have been used to separate bacteria from blood, with many others that could be applied to the problem. With the requirement of rapid identification, there is a significant challenge to be able to process the necessary 7 to 10 mLs of blood at a bacterial concentration of 10 CFU/mL. Examples from the review above for separating bacteria from blood are presented in Table 2.3. This table highlights the speed at which the techniques can process the blood, whether or not the blood needs to be diluted, the bacterial separation efficiency, and the lowest bacterial loading.

The techniques which can process at least 1 mL/min of whole blood are lectin on magnetic beads (8.9 mL/min) [87, 88], centrifugal sedimentation in a disk (2.5 mL/min) [26, 27, 76], parallel skimming channels (1.5 mL/min) [81], and Bis-Zn-dipicolylamine on magnetic beads (1 mL/min) [93]. The techniques with the recovery efficiencies of at least 80% are substrates on magnetic beads (80-90%) [87, 88, 93], RBC margination (80%) [80, 81], and hydrodynamic focusing in straight channels (80%) [82]. The techniques that can process clinical bacterial concentrations (10-100 CFU/mL) are centrifugal sedimentation in a disk (10 CFU/mL) [27], LPS antibodies on glass (50 CFU/mL) [89], and hydrodynamic focusing using Dean flow (100 CFU/mL) [79]. The techniques that can process whole blood without diluting are lectin on magnetic beads [87], centrifugal sedimentation [26, 27, 76], RBC margination [80, 81], Bis-Zn-dipicolylamine [93], and LPS antibodies on glass [89].
Table 2.3: Summary of sampling of successful techniques used to separate bacteria from blood.**

<table>
<thead>
<tr>
<th>Technique</th>
<th>Blood Flow Rate*</th>
<th>Dilution</th>
<th>Separation Efficiency (%)</th>
<th>Bacterial Loading</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lectin on magnetic beads</td>
<td>8,900 µL/min</td>
<td>Undiluted</td>
<td>90</td>
<td>10⁴ CFU/mL</td>
<td>[87]</td>
</tr>
<tr>
<td>Centrifugal sedimentation in hollow disk</td>
<td>2,500 µL/min</td>
<td>Undiluted</td>
<td>70</td>
<td>10 CFU/mL</td>
<td>[26, 27]</td>
</tr>
<tr>
<td>RBC margination in straight channels</td>
<td>1,500 µL/min (parallel)</td>
<td>Undiluted</td>
<td>60 (3 stages-parallel system)</td>
<td>10⁶ CFU/mL</td>
<td>[80, 81]</td>
</tr>
<tr>
<td>Bis-Zn-dipicolylamine on magnetic beads</td>
<td>30 µL/min (single)</td>
<td>Undiluted</td>
<td>80</td>
<td>10⁶ CFU/mL</td>
<td>[93]</td>
</tr>
<tr>
<td>Hydrodynamic focusing: Dean circulation</td>
<td>1,000 µL/min</td>
<td>Undiluted</td>
<td>80</td>
<td>10⁶ CFU/mL</td>
<td></td>
</tr>
<tr>
<td>Hydrodynamic focusing: channel flow</td>
<td>50 µL/min (parallel)</td>
<td>1:30</td>
<td>65</td>
<td>100 CFU/mL</td>
<td>[79]</td>
</tr>
<tr>
<td>Hydrodynamic focusing: sheath flow</td>
<td>40 µL/min (parallel)</td>
<td>1:200</td>
<td>80 (2 stages)</td>
<td>10⁸ CFU/mL</td>
<td>[82]</td>
</tr>
<tr>
<td>Hydrodynamic focusing: sheath flow</td>
<td>1 µL/min (single)</td>
<td>1:20</td>
<td>62</td>
<td>10⁷ CFU/mL</td>
<td>[83]</td>
</tr>
<tr>
<td>LPS antibodies on glass</td>
<td>0.5 µL/min</td>
<td>Undiluted</td>
<td>70</td>
<td>50 CFU/mL</td>
<td>[89]</td>
</tr>
<tr>
<td>Dielectrophoresis</td>
<td>0.005 µL/min</td>
<td>1:2000</td>
<td>Not reported</td>
<td>10⁶ CFU/mL</td>
<td>[108]</td>
</tr>
</tbody>
</table>

* Blood flow rate is adjusted according to blood dilution (e.g. a 200 µL/min system with 1:200 blood dilution is reported as 1 µL/min blood flow)

** This table is a modified version of Table 3 from reference [7].
It is hard to compare these processes, as one has to judge what is more important: speed, recovery, or loading. However, the results presented in Table 2.3 provide great hope for the future and for a solution to rapidly identifying bacterial infections and any antibiotic-resistance present. However, there is still much research and development that needs to be done. As the mortality rate from antibiotic-resistant blood infections continues to increase, there is an essential need and an increasing demand for this technology.

### 2.2 Separation Instability

The novel device developed by our lab [7, 26, 76] relies on centrifugal sedimentation in a hollow disk for the separation of red and white blood cells (RBCs and WBCs, respectively) from the plasma (See Figure 2.1). This technique relies on the creation of two layers during the centrifugal sedimentation: a cell-pack layer and a plasma layer (See Figure 6.1) [110]. The formation of these layers is dependent on the speed and duration of spinning. Thus, depending on the blood cell group of interest, the speed and duration of spinning can be manipulated to achieve the desired sedimentation rates and separation results. For example, the spinning hollow disk can separate out the large RBCs and WBCs from the plasma within seconds.

![Figure 2.1: A) Angled view of entire 12-cm disk. B) Cutaway of disk highlighting the RBC collection trough, the weir for maintaining separation during plasma flow down, the bowl for plasma collection, the baffles to help plasma flow down once the disk is stopped, and the lid of the disk. C) Plasma and blood cell distribution at the commencement of deceleration for 8.5 mLs of blood.](image-url)
In order for the device to be clinically relevant, the two layers must remain separated throughout the deceleration of the disk. However, it was observed on the device that during the deceleration of the blood after centrifugal sedimentation separation, the separated cell pack and plasma layers sometimes remix [26, 110, 111]. This remixing may occur in the separated layers through diffusion or through the growth of an instability. However, with the centrifugal force and the short duration of the spin (< 5 min), diffusive mixing is negligible. In order to maintain the usefulness of the device as a medical device, it is important to understand the stability criteria for the two-layer flow during deceleration.

Stability theory deals with the mathematical analysis of the evolution of disturbances to the system. If the systems response to a small or moderate amplitude disturbance is to return to its original state, the system is said to be stable. If the disturbance grows and changes the system, the system is said to be unstable. Neutrally stable systems receive a disturbance but the disturbance neither grows nor decays but is maintained by the system. Disturbances are usually assumed to be small, which allows the perturbation to be simplified, such that a linear equation governs the evolution of the disturbance. As the disturbance velocities grow, nonlinear effects become important and linear equations no longer accurately predict the disturbance evolution. While nonlinear stability analysis captures all of the evolution of the disturbance, linear stability is important in detecting physical growth mechanisms and identifying dominant disturbance types.

In addition to blood cells being more dense than plasma which allows them to be separated during centrifugal sedimentation, there is also a large difference in viscosity between packed blood cells and plasma. Instabilities in viscosity-stratified flow have been well studied [112]. Yih showed that the viscous interface between two immiscible fluids can be unstable to long wavelengths at any Reynolds number for bounded unidirectional flow with constant shear stress [113]. Analysis of semi-bounded flow (with the narrow layer bounded below the wider layer and against the wall), showed that if the narrow layer is the more viscous layer, then the viscosity difference is destabilizing; but if the wider layer is the more viscous layer, then the viscosity difference is stabilizing [114]. The analysis of unbounded flow proved to be stable to long wavelengths but unstable to short wavelengths [115]. Intermediate wavelengths were found to be unstable for sufficiently high Reynolds number (Re) and a less viscous narrow layer. These findings show that the loca-
tion of the interface is also an important parameter in determining stability of immiscible, viscous flows.

Work has been done by Charru and Hinch to determine the exact mechanisms for these instabilities [116]. Two important sets of experiments using two-layer rotating Couette flow showed that long-wave instabilities were of the Yih type and the shorter-wave instabilities were of the intermediate type, occurring at higher Re and wavelengths of the same magnitude as the thickness of the lower layer [117,118]. Sangalli et al. also demonstrated that the dominant-unstable wavelength increases as the more viscous fluid occupies less and less of the container [117]. Sangalli et al.’s experiments used density-matched fluids, while Barthelet et al.’s experiments did not [117,118]. While these studies and experiments provide great insight into viscosity-stratified instabilities, none of them analyzes instabilities in the presence of a free surface, which is the situation currently under consideration (see Figure 2.1).

For the two-layer system created by the hollow disk, it is noted that the flow is bounded by a moving wall on one side and is unbounded (a free surface, air-interface) on the other. As mentioned above, the literature search revealed that no previous stability study had been performed on such a flow. Similar flows are the two-layer inclined-plane flows, which are gravity driven flows and contain an air-interface. Kao investigated the stability of a discontinuous viscous stratification on an inclined, free-surface flow composed of two discrete layers of fluid with no surface tension [119]. He analyzed the long wavelength limit and identified two modes of instability. At small Re, one of the modes is unstable for a more viscous upper layer, while both modes are stable for a less viscous upper layer. He also noted that the critical Re for instability decreases as the difference in viscosity increases between the two layers.

Loewenherz and Lawerence investigated the effect of viscosity stratification on an inclined, free-surface flow composed of two discrete layers of fluid with no surface tension at low Re [120]. They determined that unlike Kao’s flow, the unstable wavelengths were the same magnitude as the thickness of the lower layer as Re approached zero. In both inclined, free-surface studies, neutral stability depends upon the angle of inclination only when inertial effects are present. In the absence of inertial effects, the free surface interface is always stable so that flow instabilities appear because of the liquid interface [119,120]. Hu et al. investigated the same scenario as Loewenherz and Lawerence except that they also included a density stratification in their analysis [121]. They
found that when the layer against the wall is more dense and has a viscosity greater than 1.25 times the upper fluid, the fluids are always stable.

Two recent papers performed stability analyses on a free-surface flow with a stationary wall boundary, but they analyzed the inviscid case with a density stratification [122, 123]. These papers found that for Poiseuille flow and flow with an inflection point, there is short wavelength instability with a long wave cutoff value that increases with increasing gravity. With separated blood flow, the denser and more viscous (at least 6-fold more viscous) fluid is the cell-pack which is the layer against the wall. All of these papers provide great insight into instabilities with a free surface, but most state that the given separated blood flow with the cell-pack against the wall should be stable. None of the above studies were performed for temporally varying base states of the flow, yet it is postulated that the rate of deceleration of the disk is the key cause of instability.

2.2.1 Outlook

When blood is spun on the hollow disk, the geometry of the system (see Figures 2.1 and 6.1) involves a two-fluid system that contains a moving wall and a free surface. A review of the literature revealed that this configuration has never been studied through mathematical analysis nor experimentation. However, the instabilities in such a configuration may resemble the instabilities some of the instabilities described above. Therefore, this configuration is analyzed in Chapter 6 in order to determine a stability criteria for the hollow disk.

2.3 Bacterial DNA Extraction

Nucleic acids are essential sources of information when studying and analyzing an organism. Nucleic acids have many uses, such as helping determine the similarities and differences between organisms in genetic studies [124], detecting organisms that are threats like antibiotic-resistance bacteria [125], or being used in therapies for disease prevention [126]. Current techniques of nucleic acid analysis need well-preserved nucleic acids that have received minimal damage due to degradation and/or shearing [41, 127]. Point-of-care (POC), chip-based, and other biosensing devices that detect or identify specific DNA signatures need effective yet simple means of obtaining high quality DNA [128]. Many analysis techniques also need a large quantity to en-
sure accuracy and completeness of the analysis. Large sample quantities can be difficult to obtain when working with clinical isolates of diseases [15, 24] or identifying all of the organisms in a soil or water extract [129]. In cases such as these, it is essential that the organisms undergo a complete lysis during the nucleic extraction procedure and that the lysis step releases all of the nucleic acid to ensure correct identification and/or detection. Even when large sample quantities can be easily obtained, a variety of contaminants can inhibit amplification and diminish the success of such analytical instruments [130]. In order to circumvent this problem, nucleic acids must be extracted and purified from a sample through a variety of lysis protocols and purification techniques [128]. Sample preparation (the series of lysis, extraction and purification steps) is still a major obstacle in developing better identification and detection devices [131].

There are 4 main mechanisms by which cells are lysed: mechanical, electrical, thermal and chemical [132–136]. Mechanical lysis uses shearing-forces, pressures and contact forces to lyse the cell [133, 137, 138]. Electrical lysis uses strong electric fields to facilitate cell rupture or low-strength fields to induce cell membrane pores [139, 140]. Thermal lysis uses high temperatures, extremely low temperatures, or temperature cycles to induce cell rupture and membrane disruption [141–143]. Chemical lysis uses detergents, enzymes, salts and other chemicals to disrupt or break down the cell membrane [144–147]. A couple of good reviews on all types of cell lysis specifically adapted to microfluidic devices can be found in [134–136]. Often multiple types of cell lysis are used in conjunction to achieve desired results [133, 148] such as boiling the cells in a detergent solution or using ultrasound in combination with a chemical solution.

2.3.1 Mechanical Lysis

For mechanical lysis, there are several different ways to break the cells. Some of these include the use of pressure or grinding to break the cells [132]. Di Carlo et al. forced cells through a filter containing sharp nanoscale barbs on the walls [149]. Others used high pressure flow into the wall of a divergent channel to break open the bacteria [150, 151]. Kim et al. used glass beads on a microfluidic disk to break up bacteria [152]. Kido et al. improved the idea and created a system that integrated the cell lysis step with nucleic acid purification and analysis [133]. Taylor et al. implemented sonication on a microfluidic system to lyse the cells [153].
Another form of mechanical lysis is the use of lasers, which can be used to produce cavitation bubbles which expand and contract and lyse cells in the process [154].

### 2.3.2 Electrical Lysis

For electrical lysis, the main technique is to expose cells to high-intensity pulsed electric fields [140]. The necessary exposure time for lysis is determined by field strength, with stronger fields lysing more quickly [134]. To induce lysis, the cell membrane potential needs to reach 1.1 V, on average, usually requiring an electric field of about 20 kV/cm [155]. Applications typically differ in geometry of electrode configuration. Cheng et al. used a checkerboard of 25 electrodes to separate and then lysis *E. coli* [156]. To minimize the distance between electrodes, Lee and Tai used saw-tooth electrodes for lysis [157]. Lu et al. further altered the geometry by incorporating three-dimensional cylinders as the electrodes in their system, which provides more volume in which the electric field affects the cell [158].

### 2.3.3 Thermal Lysis

For thermal lysis, there are a variety of ways to expose a sample to high temperatures. The most common way is the immersion of a sample tube in a boiling water bath for a short time period [141, 159]. Liu et al incorporated thermal lysis as part of the first step of a PCR thermal cycle and achieved sufficient lysis for the DNA present to be amplified [160]. Lee et al improved upon this by incorporating two heating chambers, one for lysis and the other for PCR [161]. The DNA was then electrokinetically transported from the lysis chamber to the PCR chamber, which improved amplification. This was still a time-consuming and labor intensive batch process. An alternative method was developed by Privorotskaya et al. using a microcantilever, where cells could be absorbed to the cantilever and then lysed by running a current through the cantilever to heat up the cells [162]. Zhu et al created another heating method by moving bacteria through two copper coils [163]. The first coil heated the fluid up to induce lysis while the second coil cooled the fluid down. An additional method was developed by Cho et al., using an optothermal method [164]. Cells are thermally lysed by using a laser to heat up metallic nano-particles in the fluid.
2.3.4 Chemical Lysis

For chemical lysis, there are many different types and combinations of chemicals and lytic agents used to induce lysis. The main chemical lysis for nucleic acids is the alkali-SDS method developed by Birnboim and Doly [132, 135, 136, 165]. The type of lytic agents used is often associated with the cell type being lysed and the target molecule. Triton X, a non-ionic detergent, is primarily used for mammalian cell lysis or if proteins are the target molecule [134]. SDS, an ionic detergent, is used for nucleic acid preparations, since SDS denatures DNAse and RNAse enzymes [166]. Guanidinium chloride and guanidinium isothiocynate (GSCN), chaotropic salts, are primarily used for nucleic acid preparations, since they denature the RNAse enzyme and assist in the binding of nucleic acids to silica [167].

Schilling et al. created a continuous flow channel where the sample fluid is injected into a channel and joins with a chemical lysing solution that disrupts the cells [168]. Heo et al. demonstrated chemical cell lysis by trapping cells in a hydrogel in a channel and flowing a SDS solution through the channel [169]. Mahalanabis et al. demonstrated single step cell lysis and DNA extraction by mixing the sample solution with a SDS-GSCN solution in the presence of a silica-coated monolith [170]. Strohmeier et al. created a centrifugal disk that could process a variety of samples depending on the loaded lysing solution [171]. The sample was loaded into a chamber near the center of the disk and then spun into the chemical lysing solution and mixed by moving magnetic beads back and forth through the sample.

2.3.5 Outlook

There are pros and cons to each type of lysis depending on the application. For microfluidic applications, it can be difficult to achieve the needed pressures for some of the mechanical lysis techniques. Mechanical and electrical techniques also often require additional equipment to accomplish lysis, but do not introduce additional substances into the sample. Chemical techniques do not require additional equipment, but the introduction of chemicals and lytic agents can often inhibit downstream applications requiring additional processing to remove such reagents later. Thermal lysis must be carefully controlled to ensure even heating and cooling since prolonged treatments can degrade the target molecules [172, 173]. Ideally, the optimal lysis induces both
high yield and high quality. But yield and quality are competing parameters [133]. Another consideration is the large assortment of bacterial cells that cause infections and the need for the lysis technique to be able to lyse all types of bacteria.

### 2.4 Bacterial DNA Collection

DNA collection is directly tied to the lysis of the bacteria [133]. Once the bacteria are lysed, the DNA is released into the solution along with many impurities (proteins, cell debris, RNA) that need to be removed [134, 167]. There are several different ways that DNA can be collected, but the optimum procedure needs to be easily implemented on a microfluidic device. Being able to perform the DNA collection on a microfluidic device enables the opportunity for integration of cell lysis, DNA collection and DNA analysis. This section of background will briefly review many different DNA collection techniques. These techniques include density gradients, electrophoresis, organic extraction, solid-phase extraction (SPE), filtration, electrostatic interaction, and adsorption on beads.

#### 2.4.1 Density gradients

There are two main density gradient methods for isolating nucleic acids from lysate: sucrose gradients [174] and cesium-chloride gradients [175]. In both cases, the lysate is layered on top of the gradient and then the tube is centrifuged using an ultracentrifuge [176]. In the case of cesium-chloride gradients, ethidium bromide is added to the lysate to distinguish intact double-stranded DNA from broken double-stranded DNA and single-stranded DNA [175]. The region of interest is then eluted from the tube by puncturing the tube wall and pipetting out increments of the solution using a syringe and needle [176, 177]. These can then be compared to known nucleic acid sediments for identification. An alternative to pipetting out the DNA is to perform the centrifugation and take absorbance measurements for determining amount of DNA collected at each gradient [178]. Due to the high gravity forces encountered in ultracentrifuges, density gradient separation is unlikely for implementation on a microfluidic device.
2.4.2 Electrophoresis

The standard method for electrophoresis is gel electrophoresis in a dc electric field [179]. Gel electrophoresis is performed in two standard media, agarose and polyacrylamide. Agarose gels form pore sizes on the order of hundreds of nanometers, where as the cross-linked polyacrylamide gels have pore sizes on the order of tens of nanometers [180]. In general, agarose gels are used to separate longer DNA and polyacrylamide gels used to separate short DNA [179]. Fujii demonstrated the use of an agarose gel in a microfluidic system for separation of DNA [181]. Pal demonstrated the use of a polyacrylamide gel in a microfluidic system for DNA separation [182].

An alteration to the standard method is to use pulsed field gel electrophoresis, in which the electric field switches between two directions [179]. This is mainly used to separate long DNA from each other, since there is a limitation to the size in which a constant electric field is effective in separation [179]. However, using a variant of this technique, Broemeling et al. were able to concentrate DNA together and remove PCR contaminants [183]. Another alteration is capillary electrophoresis, which performs electrophoresis in a capillary tube. This variation allows for higher electric fields for better separation [179]. Manz et al. [184] and Harrison et al. [185] demonstrated the use of an integrated microsystem employing capillary electrophoresis. Liu et al. developed a complete microfluidic system from loading of intact cell sample to DNA analysis that incorporated capillary electrophoresis [186].

A similar approach is isotachophoresis (ITP) which is the separation and concentration of charged components in an electric field [187]. ITP requires two different electrolyte solutions having a higher and lower electrophoretic mobility than the sample component of interest [187]. Wainright et al. used ITP to pre-concentrate and remove DNA from other contaminating substances [188]. Mei et al. [189] and Shintaku et al. [190] have successfully separated DNA from cell lysates and concentrated it for downstream processing using ITP.

2.4.3 Organic extraction

Organic extraction is the conventional benchtop method used in laboratories [191–193]. Once the sample is lysed, a phenol/chloroform solution is mixed into the sample and then centrifuged. After centrifuging, the supernatant is removed to a new tube and the DNA precipitated.
using ethanol, and the liquid is removed. These two steps can be repeated as needed to ensure
the necessary purity levels. Then, the precipitated DNA is dissolved into the elution buffer for
analysis [191–196]. Benchtop organic extraction requires larger volumes of lysate due to incom-
plete removal of supernatant and loss of material in the organic phase [191, 195]. This process is
also time consuming due to the multiple steps of adding chemicals and centrifuging [192, 193].
There is also the concern of the hazards of organic extractions, but microfluidics could help not
only reduce amounts but eliminate exposure to workers. Reddy and Zahn demonstrated the use
of a phenol/chloroform extraction in a microfluidic system [197]. Morales and Zahn improved
the process by creating small, aqueous droplets in an organic fluid to increase interfacial area and
diffusion [198]. They achieved good nucleic acid isolation, with only about 5% lost to the organic
phase.

2.4.4 Solid-phase Extraction

There are three main types of solid-phase extraction (SPE): normal phase, reverse phase,
and ion-exchange. Normal phase binds polar molecules to the solid substrate and washes away
non-polar molecules [199, 200]. Reverse phase binds non-polar molecules to the solid substrate
and washes away polar molecules [199, 200]. Cation exchange binds cationic substances to the
solid substrate but washes away anionic substances [199, 200]. Anionic exchange binds anionic
substances to the solid substrate but washes away cationic substances [199,200]. DNA purification
is compatible with normal phase and anionic extraction, but DNA is washed away with reverse
phase and cationic extraction. When using SPE, the composition of the cell lysate, the washing
buffer and the elution buffer greatly affect the yield and purity of the collected DNA.

SPE is commonly used for DNA isolation. Silica or glass fibers are the two main station-
ary phases used (normal phase) for DNA extraction [201]. DNA is able to bind to the silica or
glass in high ionic conditions, which decreases the electrostatic repulsion [134]. The DNA is then
washed with a non-polar solvent and eluted in a low ionic strength buffer. One study immobilized
silica beads in a sol-gel matrix and obtained $\sim$90% extraction efficiency of $\lambda$-DNA [202]. These
researchers further optimized the loading buffer condition and flow rate for a sol-gel/silica bead
matrix. Using these optimized procedures, they extracted genomic DNA from whole blood and
bacteria in less than 15 min [203]. However, Wu et al. showed that a decrease in surface area
greatly decreased the extraction efficiency of comparable sized sol-gel/silica bead matrices [204].
By adjusting the surface chemistry of the SPE, Chen et al. was able to increase the adsorption sites for nucleic acids [205].

Another study examined the binding and eluting conditions of DNA on solid-phase silica surfaces. They found that guanidine thiocyanate promoted the highest adsorption on the silica surface and that DNA had a greater affinity to silica surfaces under low pH solutions. However, only 50% of the adsorbed DNA was eluted in 5 mLs of elution buffer [127]. Another study combined a reverse phase and a normal phase extraction in a microfluidic device for better elimination of proteins and endotoxins. The cell lysate flowed through the reverse phase extraction where the hydrophobic substances were tightly bound and then onto a normal phase extraction where the DNA was bound. They were able to increase the DNA extraction efficiency compared to a single normal phase [206]. One research group was able to fully integrate cell lysis, SPE, PCR and an electrophoretic analysis onto a single microfluidic device [207]. The researchers demonstrated the ability to detect the presence of infectious pathogens using their device. The efficiency of SPEs depends on the matrix surface area, the nature of the functional group, the binding conditions (salt, pH, and contaminants), and the eluting buffer. The main challenge with SPE in microfluidic devices is the difficulty in immobilizing the solid-phase structures in the device [134].

2.4.5 Filtration

Filtration techniques are often combined with other techniques to achieve the necessary purity for analyzing the DNA after separation. A surface modification of the filter material can allow the filter to act like a SPE by capturing the DNA as it passes through, or one could modify the surface to involve electrostatic interactions. Filters can be used as a means to allow the DNA to flow through while blocking other cellular material or they can be used to block DNA while allowing other cellular material to flow through.

Elgort et al. demonstrated the use of nanoporous aluminum oxide membrane (AOM)-based purification of genomic DNA (gDNA) from human blood in which blood was lysed and gDNA was localized to the AOM surface by filtration and possible surface interactions [208]. Using this concept, Kim et al. were able to integrate the nanoporous AOM with microfluidic channels to extract the DNA [209]. They studied the physical and chemical interactions between
the DNA and the AOM by testing different lysis and elution buffers. Under high-salt conditions, the gDNA collection rate increased due to strengthened interactions between DNA, the surface and the aggregation of the gDNA itself. During elution, higher pH and anionic solutions had higher extraction efficiencies. Smaller nanopores were also found to increase the extraction efficiency with pure DNA samples, but mid-sized pores (100 nm) were found to be better when used with samples containing contamination from cell debris, blood plasma proteins or even residual red blood cell debris.

2.4.6 Electrostatic interactions

DNA collection by electrostatic interactions has been demonstrated with a variety of surface modifications. Because DNA is negatively charged, it will bind to positively charged surfaces. The most basic modifications rely on an amine surface coating. Amine groups below neutral pH have a positive charge, which decreases above neutral pH. However, DNA recovery from amine-coated surfaces depends on the concentration of phosphate and salt in the elution buffer and the temperature at elution. When an amine surface was tested with whole blood, the elution efficiency was determined to be 60% [210]. Changing the composition of the elution buffer may result in higher efficiencies, but the elution buffer can affect subsequent steps, such as DNA amplification [211]. Often in electrostatic methods a high pH is needed for elution (pH > 10) which can cause strand separation or affect downstream processes. One way to overcome this problem is to dilute the elution solution with the buffer needed for the downstream process, but this can over dilute the DNA causing problems with the process. Another way to overcome this problem is to use difference electrostatic surfaces. Cao et al. developed a chitosan surface coating to extract DNA from whole blood [212]. Chitosan is positively charged at pH 5.0 and is easily neutralized at pH 9.0. High density microfluidic channels coated with chitosan were fabricated and tested with lysed whole blood samples. DNA was captured at pH 5.0 through electrostatic interactions with the chitosan and eluted using pH 9.1 Tris buffer. The extraction efficiency for human genomic DNA was 75% [212]. The DNA was shown to be suitable for PCR following extraction.

DNA has also been collected on a polycarbonate (PC) surface with carbonyl/carboxyl functional groups attached by a polymerization reaction under ultraviolet light. Using this technique, Witek et al. demonstrated that about 85% of gDNA could be extracted from a lysed E. coli sam-
ple within 25 min using a 3% polyethylene glycol binding buffer [213]. They then created a high-throughput nucleic acid extraction system using a 96-well PC microfluidic bed. The loading capacity of nucleic acid from *E. coli* samples was 206 ng for gDNA and 165 ng for total RNA. The extraction efficiency was 63% for gDNA and 73% for total RNA [213]. PCR and reverse transcript PCR were successfully performed without any inhibition.

2.4.7 Bead Adsorption

Magnetic microparticles with DNA-adsorbent surfaces are the most common microparticles used for DNA extraction. The most common coatings for the magnetic particles are: silica [214], functionalized carboxyl groups [215], and amine groups [211]. Due to the paramagnetic quality of the particles, the particles are suspended in the fluid and then easily separated from the solution by applying a magnetic field. Because the beads are free floating, some washing steps can be eliminated that are common with other SPEs. A drawback to using magnetic particles in microfluidic systems is the need for a magnetic field to collect the particles. “On-device” magnetic fields can be difficult to incorporate and “off-device” fields mean that additional equipment is needed to manipulate the beads.

One microfluidic device incorporated a two-spiral micro-coil system for trapping the beads. The bead capture efficiency of this system was about 84% with a 20 /min flow rate [216]. Another device using an external magnetic field lysed cells from saliva and purified the DNA for genetic disease detection. These silica-coated beads were allowed to sit in a high salt buffer for 10 min with the lysed cells for DNA extraction. They had an extraction efficiency (combined efficiency of capture and elution) of about 60% and had good purification of the DNA samples [214]. Another study used sequence-specific probes attached to magnetic beads to extract *E. coli* gDNA. The beads were able to hybridize with the denatured gDNA and then remove all of the remainder of the cellular components by a single wash step. The captured gDNA was then released through heating without a specific elution buffer [217]. This technique achieved about 75% extraction efficiency. While this technique allowed for species-specific DNA, about 25 minutes were needed for the combined denaturing of the gDNA and hybridization between the target and the probe [217]. A similar concept used a microfluidic compact disk platform to collect pathogen DNA. Cell lysis was accomplished using a laser system without any lysis buffer, and gDNA was extracted using
biotinylated magnetic beads. The extraction efficiency produced results similar to that obtained with commercial DNA preparation kits [164].

### 2.4.8 Outlook

There are a variety of ways to collect DNA from samples. Often multiple techniques are combined to create the optimum system for DNA collection. The collection efficiency of each system is dependent on the salts and pH of the solution as well as the surface chemistry of the material. The optimal process would both separate and concentrate the DNA for downstream labelling, while the optimal solution would be a low salt solution to facilitate DNA labelling. As mentioned above, the process also needs to obtain both high yield and high purity, which are competing parameters in sample cleaning and collection.

### 2.5 Microfluidic Design

This section will focus on two main aspects of microfluidic devices, mixing and valves. When pressure limits are present along with laminar flow, mixing can be very challenging. With micromixing, there is both active and passive mixing. There are some good reviews on micromixing, [218–221], including the physics of mixing flows [222]. Microvalves can be one of the most important components for realization of a fully integrated microfluidic system [223, 224]. Like micromixing, there are both active and passive valves. Some good reviews on microvalves are [225, 226].

#### 2.5.1 Micromixing

At the macroscale, turbulent flow, which occurs around $\text{Re} \approx 2200$ in pipe flow, is a simple and easy method to achieve mixing [221]. However, at the microscale, most applications involve laminar flow with low Re, $\text{Re} \leq 100$ [221]. In order to achieve mixing at the microscale, both active and passive mixing schemes have been developed. Active mixing can be accomplished using acoustic/ultrasonic [227], pressure perturbation [228], electric [229–232], and magnetic [233] techniques. Due to the laminar flow, passive mixing predominantly relies on chaotic
advection [219]. Passive mixing involves schemes such as lamination [234], intersecting channels [235, 236], convergent-divergent channels [237], embedded structures [238], and 3D channels [239].

Acoustic/ultrasound mixing is usually accomplished by piezoelectric or ultrasound transducers. Liu et al. demonstrated the use of acoustic mixing by exciting trapped air bubbles at the top of the mixing chamber with a frequency of 5 kHz [240]. Yaralioglu et al. created acoustic mixing using embedded piezoelectric transducers to generate a strong acoustic streaming effect, which enhanced mixing within the channel [241]. Bezagu et al. enhanced stream mixing by applying ultrasound shortly after merging the streams for mixing [242].

Pressure perturbation mixers generate perturbations using velocity pulsing [228]. Afzal and Kim demonstrated the effects of pulsatile flow by modeling the effects in a channel with no pulsing, pulsing from one side, simultaneous pulsing from both sides, pulsing from both sides with one side 90° out of phase, and pulsing from both sides with one side 180° out of phase [243]. No pulsing and simultaneous pulsing achieved similar results, about 15% mixing, while pulsing from one side achieved about 40% mixing. Out of phase pulsing achieved similar results to one-sided pulsing when 90° out of phase, but jumped up to 83% mixed for 180° out of phase.

Electric mixers use electric fields to create electrokinetic or electrohydrodynamic driving forces which result in mixing [219]. Electrokinetic driving forces can be achieved by applying non-uniform alternating electrical fields [244] or by sending sine wave signals with varying frequencies [245]. Electrohydrodynamic forces rely on differences in electrical properties of the fluid themselves, which results in a transversal secondary flow and mixes the fluids. Campisi et al. developed a non-uniform alternating electrical field which electrokinetically displaced the liquids and created strong mixing [229]. Chen et al. demonstrated electrokinetic mixing using four electrodes with chaotic electric fields and achieved mixing efficiencies up to 95% [230]. El Moctar et al. demonstrated electrohydrodynamic mixing by flowing two fluids with identical viscosity and density but different electrical properties and achieved satisfactory mixing after 0.1 sec even at a Reynolds number of 0.02 [246].

Magnetic mixers can either rely on magneto-hydrodynamic forces or can move a magnetic object inside the fluid to stirring the fluid. Bau et al developed a magneto-hydrodynamic mixer by using both an electric field and a magnetic field to generate Lorentz forces causing mixing [247].
Wang et al. developed a magnetic particle driven mixer by alternating the magnetic field direction to move the particles back and forth, mixing the fluid [233].

Lamination operates on the principal of increasing the contact area and contact time between the different species and decreasing diffusion distances [219]. Buchegger et al. proposed a horizontal multilamination micromixer based on pre-splitting streams and then merging them all together in a wedge-shaped inlet channel [234]. Tofteberg et al. designed a mixer which slowly rotated the two streams 90°, split the streams, and then rotated both streams again 90° before recombining them to create a layered flow that mixed evenly [248]. Roudgar et al. performed a numerical study and found that uneven flow rates in a straight channel induced mixing [249]. Hsieh et al. demonstrated that optimal configuration for the merging of two streams is a reverse Y-channel (a reverse Y-channel has the two channels coming together like an arrow) with an angle of 60°, which increased mixing efficiency and reduced mixing length [250].

Intersecting channels have been shown to enhance the mixing performance within the microchannel through splitting, rearranging, and/or combining the component fluid flows [251–254]. Ansari et al. demonstrated the efficiency of using unblanced splits and collisions [235]. Li et al. developed a split-and-recombine (SAR) micromixer based on Anasari’s work and added dislocation sub-channels which obtained 86% mixing with low Reynolds numbers, 1 - 80 [255]. Chen and Shie presented a micromixer with staggered curved channels, which reduced the diffusion distance of the two fluids due to the staggered nature of the channel [256].

Convergent-divergent channels work similar to continously-stirred-tank reactors in that mixing is dependent on residence time, meaning that faster flows are less mixed [220]. Parsa et al. investigated the different amplitude-to-wavelength ratios for a convergent-divergent sinusoidal microchannel and found that diffusion is key in convergent-divergent flows [257]. Afzal and Kim developed a SAR micromixer with convergent-divergent walls and showed that secondary flows developed in the sub-channels [237]. Hong et al. demonstrated the effectiveness of a modified Tesla structure which contains a curved convergent-divergent main channel with a smaller sub-channel that wraps around to collide with the main stream [258]. Hossain et al. studied a modified Tesla structure and discovered that mixing was sensitive to the ratio of the curved gap to the channel width and the ratio varied with changes in Reynolds number [259].
Embedded structures include obstacles, such as pillars placed inside the channel, as well as uneven walls, such as ridges protruding into the flow path. Wong et al. conducted numerical investigations into the effectiveness of obstacles placed in the flow path and achieved almost complete mixing when only adding two square obstacles that reduced the channel width by 1/4 [260]. Tseng et al. tested the effects of diamond-shaped obstacles and boundary protrusions and found that the boundary protrusions induced intense vortices and secondary flows [261]. Kim et al. achieved 95% mixing when using slanted ribs placed throughout the flow channel [262]. Johnson et al. demonstrated successful mixing (≥75%) by placing wells into the bottom of the flow channel that were offset from horizontal [263]. Yang et al. investigated the effects of asymmetric staggered herringbone grooves and found that vortices with dissimilar scales were shown to provide better mixing along with increasing the groove depth [264].

The 3D mixers that will be considered here are those that employ movement in the horizontal and vertical direction but do not contain any overlapping flows. Liu et al. investigated with numerical simulations a 3D serpentine mixer and found that the mixing performance varied inversely with the mass fraction of the sample at $Re = 1$ [265]. In another study, Liu et al. compared a repeating "C-shaped" 3D serpentine mixer to a zigzag channel and found the the serpentine mixer achieved 1.6 times better mixing than a zigzag channel [266]. Nimafar et al. presented a SAR micromixer with H-shaped subchannels and obtained 98% mixing at a Reynolds number of 0.083 [267]. Li et al. showed the effects of viscosity on mixing by simulating mixing for a less and more viscous fluid at the same Reynolds number and achieved 90% mixing for the less viscous fluid in 300 $\mu$m but needed 1000 $\mu$m for the more viscous fluid to achieve the same mixing [268]. Yang et al. developed a three-dimensional modified Tesla structure, by alternating a horizontal and a vertical Tesla structure, and achieved a 97% mixing result for a $Re = 0.1$ [269].

2.5.2 Microvalves

Both active and passive valves can contain mechanical or non-mechanical parts [225, 226, 270]. Active mechanical valves may contain magnetic, electric, or thermally activated parts, while active non-mechanical valves may be activated by a phase change or rheological materials [225, 226]. Passive mechanical valves are basically check valves, while passive non-mechanical valves rely on external forces, such as capillary forces, to restrict flow [225, 226].
2.5.2.1 Active Valves

Active systems require external parts, such as power systems or external magnets, in order to work and thus increase the overall size of the unit [270]. In addition to the external parts, a control scheme is needed to activate the valve at the desired times, which either requires a secondary analysis system for triggering activation or software [270]. Some of the most common active valves for liquid flows are magnetic valves, thermopneumatic valves, shape memory alloy (SMA) valves, hydrogel valves and paraffin valves [225,226]. Electric valves are usually employed for gas flow due to electrolysis of liquids at high voltages [271,272].

Magnetic valves operate by applying a magnetic force to either repel or attract a magnetic component resulting in a change in the position of the valve [273]. Meckes et al. developed a magnetic valve using a movable membrane with a gold coil attached, which deflected when a magnetic force was applied [274]. Bae et al. created a magnetic valve using a permanent magnet attached to a flexible membrane with an external solenoid for operating the valve [275]. Krusemark et al. used a metal ball placed at the outlet orifice to stop flow and was moved up by an external magnetic force [276]. Fu et al. developed a similar valve using an iron ball that blocked flow when a magnetic force was applied [277]. The above two examples show that magnetically operated valves can be operated in either the normally open or the normally closed position. Oh et al. developed a magnetic pinch valve that combined a solenoid, a spring, and a magnetic plunger [278]. The plunger stopped flow in the silicon tube due to the spring and allowed flow when the solenoid was turned on.

Thermopneumatic valves operate by heating a liquid and/or gas chamber which contains a deformable membrane [279]. Takao et al. used a PDMS membrane for high sealing performance and about leakage rates of less than 1 $\mu$L/min at a pressure of 30kPa [280]. Baechi et al. used an array of PDMS membrane thermopneumatic valves for separating and combining particle flows [281]. SMA valves operate by applying a force to deform the SMA component and then removing the force which allows it to return to its initial state [282]. Kohl et al. developed a SMA thin film that could move approximately 20 $\mu$m using 220 mW [283]. Reynaerts et al. developed a pinch-type valve using a SMA spring to pinch down rubber tubing about 0.6 mm using 120 mW [284].

Hydrogel valves operate by a change in the volume or phase of the hydrogel induced by a response to stimulus [285]. Stimuli can include pH [286], glucose [287], temperature [288], elec-
tric fields [289] and light [290]. Liu et al. developed a hydrogel valve that swelled and deformed a PDMS membrane based on a pH change of a secondary solution [291]. Hydrogels can have relatively slow response times. Eddington and Beebe took advantage of the slow response, due to a pH change, to create a valve that pushed fluid out of a reservoir chamber at a set flow rate [292]. Yu et al developed a hydrogel plug that changed phase at a temperature of 32°C [293].

Similar to hydrogel valves, paraffin valves operate by a change in phase. Since the paraffin solid-to-liquid phase transition is associated with a 10-30% volume expansion, these plugs can be used as either a meltable plug or a propellant for a membrane [225]. Mastrangelo et al. created a paraffin valve that moved a flexible membrane to block flow when melted [294]. Two groups created an paraffin plugs that allowed flow when heated and were moved down the channel by either an external pneumatic system [295] or upstream pressure [296].

### 2.5.2.2 Passive Valves

Passive systems do not require external parts or control schemes in order to work and thus can be contained within the microfluidic device [270]. Mechanical passive valves are check valves that only allow flow in one direction [225]. Some of the most common passive valves for liquid flows are flaps, membranes, spherical balls, and capillary forces [225, 226]. The flaps, membranes and spherical ball valves are all mechanical passive valves that allow flow when pressure is applied in the forward direction.

Cantilever-type flap valves have been made of silicon [297, 298], metals [299, 300], and polymers [301]. Membrane valves have been developed with bridges [302, 303], holes [304, 305], and bumps [306]. For spherical ball valves, a smaller circular orifice and a larger-than-the-ball cage is needed [307,308]. This allows the ball to move in the cage to allow forward flow but block the orifice with backward flows.

Capillary force valves can be created using localized potential differences on the surface [309], temperature gradients on the surface [225], differences in geometry [310, 311], and differences in surface properties [312]. Ahn et al. demonstrated the use of a geometry change capillary force by using an abrupt cross-sectional area change, which generates a pressure drop across the valve resulting in fluidic flow valving [310]. Many studies have demonstrated the use of a surface property change capillary force by using an hydrophobic coating in a channel.
which then requires a certain amount of pressure for the water to move across the hydrophobic region [313–317].

2.5.3 Outlook

There are a variety of ways to obtain mixing in laminar flows and open/close channels. Many of the above techniques depend on both the composition of the fluid as well as the downstream process. The vastness of the research that has been done on microfluidic techniques will enable the development of a microfluidic device that can incorporate cell lysis, DNA collection, DNA labelling, and fluorescent analysis with all of the necessary valves and mixing required for each process.

2.6 Gene Identification

Identification of antibiotic resistance and bacterial species from DNA hinges on the ability to tag, capture, or read a DNA sequence. An ideal process would be able to identify all antibiotic resistances and bacterial species present. However, this is not feasible in a process that involves tagging or capturing the DNA because a predefined sequence is needed and there are thousands of bacterial species and antibiotic resistance sequences. Since the collaborative project, which encompasses this dissertation, is focused on capturing and tagging DNA sequences, only a handful of antibiotic-resistance genes and species identification sequences can be targeted [318,319]. This section will review the current gene identification process being used for the collaborative project as well as some of the alternatives techniques that could be explored. These techniques include quantitative polymerase chain reaction (qPCR), in situ hybridization, microarrays, sequencing, and surface-enhance RAMAN scattering (SERS). A couple of good reviews on different methods for antibiotic-resistance detection are [320] and [321].

2.6.1 Current Technique

As mentioned in Chapter 1, the current technique involves capturing gene sequences on a monolith, attaching a fluorescent probe to the captured gene and then releasing the captured gene with it’s fluorescent probe for identification by a laser using optofluidics. This technique
involves needing a predefined sequence in order to design an immobilized compliment strand and a fluorescent probe. This limits the number of sequences that can be targeted at the same time.

This technique is very similar to the microarray technique described below. The monolith is loaded with a predefined segment of the target genes, which are then immobilized on the surface of the monolith. This step would allow for hundreds or thousands of different genes to be targeted. However, the next step of labeling the targets with a fluorescent probe limits the number of targets due to the limited number of fluorescent molecules that bind to DNA. Therefore, other techniques were explored to see if a technique could be found which could expand the capabilities of the system such that a predefined sequence is not needed for gene identification.

2.6.2 qPCR

The qPCR technique, also known as real-time PCR, measures the amount of fluorescent signal present after each cycle of PCR [322]. This technique requires primers, which are short DNA sequences that bind to the target DNA in a specific location, and thus limits the number of target sequences. Multiplexing is the simultaneous amplification of two or more target sequences using two or more pairs of primers in the same reaction [323]. Schwartz et al. were able to use multiplex qPCR to identify 3 different antibiotic-resistant sequences, but they did not identify bacterial species [324]. Dutka-Malen et al. used multiplex qPCR to identify 3 antibiotic-resistant sequences and 4 different bacterial species [325]. Depardieu et al. developed a multiplex qPCR that identified 6 antibiotic-resistant sequences and 4 different bacterial species [326]. Sandberg et al. used a miniaturized qPCR method which splits the sample DNA across 2304 wells (48 x 48 wells) to detect 48 different genes [327]. However, all of these were performed on bench-top units and not on a microfluidic device.

White et al. was able to integrate qPCR on a microfluidic device but only identified a single sequence [328]. Zhang et al. developed a microfluidic multiplexed qPCR system that was able to identify 3 bacterial species [329]. Zhong et al. was able to develop a qPCR microfluidic device which could identify 5 different targets [330]. Incorporating multiplexed qPCR on a microfluidic chip seems to be challenging as evidenced by the small amount of literature on the subject.
2.6.3 in situ Hybridization

The in situ hybridization technique involves the binding or hybridization of a probe sequence to the desired target—either a specific gene or RNA molecule [331]. The most common in situ hybridization detection method is fluorescence, but other methods such as magnetic or label-free in situ hybridization can be used [332]. Zwirglmaier et al. proved that fluorescent in situ hybridization (FISH) could be used to identify individual genes in an organism by identifying an antibiotic-resistant gene in an *Escherichia coli* model [333]. Laflamme et al. were able to identify antibiotic resistance in a *Bacillus cereus* model using FISH [332]. Bhattacharyya et al. were able to identify different bacterial species using a multiplexed FISH assay with 89% accuracy at the species level and 100% accuracy at the family level [334]. However, none of these were on a microfluidic device.

Sieben et al. developed a microfluidic device that integrated FISH as a way to detect the genetic content of individual cells [335]. Vedarethinam et al. used FISH for detecting chromosomal abnormalities on a microfluidic device as a means of reducing the cost and time for a normal laboratory protocol [336]. Liu et al. combined FISH with flow cytometry on a microfluidic device as a way to probe and then select desired bacteria [337]. This device could be a suitable alternative to the current gene identification process, since this would eliminate the need to break open the bacteria for DNA release.

2.6.4 DNA microarray

The microarray technique relies on target DNA hybridization to an immobilized probe which can then be measured by fluorescence, chemiluminescence, electrochemical detection, or radiochemical detection [338]. This technique relies on short DNA fragments, which are usually created by running a preliminary PCR, and probes which are the complimentary strand of the target sequence. Ma et al. developed a microarray which detected 11 antibiotic-resistant genes simultaneously [339]. Batchelor et al. developed a microarray which targeted 47 antibiotic-resistant genes [340]. Card et al. developed a microarray which screened organisms for 70 different antibiotic-resistant genes [341]. Frye et al. developed a microarray that detected 94 different antibiotic-resistant genes [342]. While this technique can detect large amounts of resistant genes,
hybridization usually takes multiple hours before the sample is ready for detection [338]. While the technique can detect genes at concentrations of 50 pM [338], 50 pM of genes translates into approximately $1 \times 10^8$ CFU/mL if there are 100 copies of the gene in a bacteria. This does not account for any concentration of the DNA nor loss of DNA, both of which are likely to happen during the extraction of the DNA from the bacteria, but it does give an estimate of how many bacteria are needed.

However, using microfluidics the hybridization times were decreased to inside a half hour. Liu and Rauch developed a microfluidic microarray that could detect 90 different targets while only needing 15 min at 50°C [343]. Pappaert et al. developed a microfluidic microarray that could detect 216 different targets with 30 minutes of hybridization time at 42°C [344]. Wang et al. developed an microfluidic array that only detected 4 targets but cut hybridization time down to 2 min at 50°C [345]. Peytavi et al. developed a microfluidic array that could detect 150 350-bp target segments with a 5 min hybridization time at room temperature [346]. Wang et al. developed a centrifugal-microfluidic device that detected 100 different targets with a hybridization time of 10 min at 45°C [347], which could potentially be combined with the previously developed centrifugal-sedimentation disk. While all of these demonstrate the usefulness of microarrays, none of these quicker microarrays used antibiotic-resistant sequences; but the potential is there.

2.6.5 Sequencing

Sequencing techniques relies on the ability of DNA polymerase to create the compliment strand of a DNA sequence [348]. Sequencing has both strengths and weaknesses. The main strength of sequencing is that it can detect all genes present in the DNA and that there is no need to selectively choose what antibiotic resistance genes to detect. A couple of the weaknesses of sequencing are the slow turn around that you get from sequencing, which takes multiple hours, and the need to piece together the sequence reads into a contiguous sequence, which makes it almost impossible if you have multiple genomes and plasmids present. However, some work has been done using sequencing to detect antibiotic resistance. Bennedsen et al. used sequencing on bacteria from a collection distributor to ensure that the correct genes were maintained in the collection strains [349].
Research is being done to incorporate sequencing on a microfluidic device. Kartalov and Quake developed a proof of concept microfluidic device that was able to read 4 consecutive bases [350]. Paegel et al. designed a microfluidic device that was able to sequence 500 bases on a known target molecule [351]. Blazej et al. were also able to sequence a continuous 500 bases on a microfluidic device [352]. Wang et al. developed a microfluidic device that could sequence 600 bases in 15 min [353]. A couple of good reviews on microfluidic sequencing are [354], [348], [355] and [356]. All of the above experiments used known vectors for sequencing at a set starting point to prove the capabilities of the system, which doesn’t provide insight into how the device would perform with unknown samples.

2.6.6 SERS

Surface enhanced Raman scattering (SERS) is an analytical spectroscopic technique that relies on photon scattering from the object of interest [357]. The spectra that is measured after interacting with the object provides a unique signature for particle identification [357]. Otto et al. showed that DNA bases could be detected and distinguished from each other using SERS [358]. Barhoumi et al. showed that you could detect and identify changes in DNA oligomers using SERS [359]. Several studies used SERS as a label-free method to detect target DNA sequences through hybridization to an immobilized probe in a microfluidic device [360–362]. Sagar et al. showed that SERS can do more than just detect oligomers: it can return the percentage of individual bases in the oligomer [363]. Korshoj and Nagpal provided a proof of concept paper on identifying specific genes from the base percentage obtained by SERS [364]. However, they did not test their hypothesis against background noise of other DNA being present. This idea is very intriguing since it obtains fast recovery of the base percentage in short oligomers. This technique is further analyzed in Appendix C.

2.6.7 Outlook

There are a variety of ways to identify genes. Most techniques rely on the hybridization of the target sequence to a probe, which means that only a limited amount of specific genes can be identified at once. These techniques can be very quick and efficient when only 1 or 2 genes need
to be identified. However, there are a few techniques which read DNA content. These techniques
could theoretically identify all genes present, but they are still under development and have current
limitations in implementation. Therefore, the ideal identification depends on how quickly and how
many genes need to be identified, as well as how much time can be devoted to development.
CHAPTER 3. OBJECTIVES

The overall goal of this dissertation is to collect the maximal amount of the DNA from the bacteria that have been separated out of the blood, and to concentrate the DNA in 100 µL within 7 minutes using a microfluidic device. The first two objectives below are related to the separation of the bacteria from the blood. The remaining objectives pertain to the recovery and concentration of the DNA. The specific objectives are:

1. To provide an analysis of the deceleration profile of the spinning disk to determine the stability criteria, which will provide the fastest stable deceleration in which bacteria can be recovered without significant remixing of the blood cells and plasma;

2. To isolate the bacteria for lysis after the bacteria have been spun on the disk by removing any remaining blood components that may interfere with subsequent analyses;

3. To effectively lyse the bacteria while minimizing the amount of time and the amount of reagents required for lysis;

4. To collect in 100 µL the DNA released from lysis and to decrease the size of the DNA to an average of 1-2 kbp;

5. To design a microfluidic device in which objectives 3 and 4 can be accomplished on device. The device may also include objective 2 if feasible.

To accomplish the first objective, the deceleration profile was solved for first. Once the profile was obtained, then a linear stability analysis was run to determine the stability criteria. In addition to the mathematical analysis, experiments were designed and executed to see if an experimental stability criteria can be established.

The second objective was accomplished by experimenting with the ability of a filter to concentrate and release bacteria. The literature was consulted to determine what had been done
to provide a starting point for experiments. The size of the filter, the starting solution and the backflush solution were parameters that were examined.

The third objective was accomplished by searching the literature to determine what types of lysis had been used for lysing the bacteria of interest for the collaborative project. Experiments were designed to test literature solutions and variations of the solutions to identify a best solution.

The fourth objective was accomplished by experimenting with techniques identified in the literature which worked with the cartridge from the collaborative company. Solutions, temperature, and volumes were examined to find the best combination.

The fifth objective was accomplished by searching the literature for the best means of mixing and valving a microfluidic device. Devices were prototyped and tested to determine what configurations worked with the cartridge from the collaborative company.
CHAPTER 4. EXPERIMENTAL METHODS

Due to the large number of experiments that use the same procedures, this chapter lists all methods used for experiments. Later chapters will simply reference the methods listed here when used, to limit the rewriting of procedures. When experiments deviate from what is listed in this chapter, the deviated procedure will be written and clarified in that chapter.

4.1 Bacterial Strains

All of the bacterial strains used were generously donated by Dr. Richard Robison’s lab (Microbiology, Brigham Young University).

- *Escheria coli* - DH5α containing pSCB3-EGFP plasmid
- *Escheria coli* - DH5α containing pSCB3-KPC plasmid, developed by Dr. Richard Robison’s lab for this project
- *Escheria coli* - DH5α containing pSCB3-VIM plasmid, developed by Dr. Richard Robison’s lab for this project
- *Escheria coli* - DH5α containing pSCB3-NDM plasmid, developed by Dr. Richard Robison’s lab for this project
- *Escheria coli* - BL21 DE3* containing pJ411 plasmid
- *Enterobacter cloacae* - ATCC 23355
- *Klebsiella pneumoniae* - ATCC 13883
- *Psuedomonas aurignosa* - ATCC 15442
- *Staphylococcus aeurus* - ATCC 6538
4.2 Bacterial Preparation and Growing

4.2.1 Frozen Stocks

All bacteria were received from the respective lab on an agar plate with any necessary antibiotics. A single colony was then scraped off the plate using a sterile wire loop and placed into a baffled Erlenmeyer flask with 20 mLs of Nutrient broth (Sigma Aldrich) along with any antibiotics if required. The culture was grown for 16 hours (overnight). Deep freeze cultures were made by placing 0.75 mL of filter-sterilized 80% glycerol along with 0.75 mL of the bacteria-nutrient broth solution into a sterilized cryogenic vial (2 mL) and vortexed. Then the cryogenic tube was stored in a -80 °C freezer. Stock freeze cultures were made bi-annually (or as necessary) by placing 0.175 mL of the filter-sterilized 80% glycerol along with 1.325 mL of the bacteria-nutrient broth solution into a sterilized cryogenic vial (2 mL) and vortexed. Then the cryogenic tube was stored in a -20 °C freezer. The bacteria-nutrient broth solution was then streaked onto a nutrient broth-agar plate to ensure no contamination was present.

4.2.2 Stock Plates

Working stock plates were made for daily bacteria growth. Plates were made by streaking the stock frozen culture on a nutrient broth-agar plate and grown in the incubator (Fisher Scientific Isotemp 637D Incubator) at 37 °C for 24 hours. Plates were then pulled out and placed in a zip lock bag in the fridge (4 °C). Nutrient agar plates were made fresh each week.

4.2.3 Daily Growth

The working stock plate for the desired bacteria was removed from the fridge and a single colony was then scraped off the plate using a sterile wire loop and placed into a baffled Erlenmeyer flask with 20 mLs of Nutrient broth along with any antibiotics if required. The flask was mixed by swirling at 120 rpm and the culture was then grown for 16 hours (overnight). The bacteria for experiments were pulled from the overnight culture.
4.3 Measuring Bacteria

For the experiments requiring known starting concentrations, the bacteria were measured using absorbance on a Cary-60 UV Spectrometer at 600 nm. The spectrometer was blanked with 3 mL of the solution the bacteria was suspended in (either clean PBS or clean nutrient broth). One milliliter was removed from the blank cuvette and 1 mL of the bacterial solution was added and mixed by pipetting up and down. The mixture was measured and then the concentration for the bacterial solution was determined using the correlation curves developed for each bacteria (procedure below). Dilutions are then performed to obtain the desired concentration, if necessary.

For bacterial spinning in the hollow disk, the bacteria were washed before measuring by centrifuging the 10 mLs of bacteria split into two 5 mL volumes in 15 mL conical vials in a Horizon 642E centrifuge (which centrifuges for 10 min at 3328 rpms, Fisher Healthcare). The supernatant was then discarded and replaced with 5 mLs of PBS and vortexed. The conical vials were then centrifuged again, the supernatant discarded, and 5 mLs of PBS added. The vials were then vortexed and 1 mL of the solution used for spectrometer measurements.

4.3.1 Concentration Correlation

In order to develop a concentration correlation, a single colony from the stock plate for the desired bacteria was used to inoculate 20 mLs of nutrient broth along with any antibiotics if required. The flask was mixed by swirling and then 1 mL was removed for absorbance measurements (at 600 nm) for quantifying the bacterial concentration using the spectrometer, and 0.1 mL was removed for dilutions and plating (procedure below). The flask was then incubated at 37 °C. The flask was then removed, swirled, and 1 mL was removed again for absorbance measurements for quantifying the bacterial concentration using the spectrometer, and 0.1 mL was removed for dilutions and plating every few hours for 24 hours. This experiment was repeated a couple of times to increase accuracy. The plate counts were plotted against the absorbance (at 600 nm) measurements and the trend line found, which was then used as the correlation curve to approximate concentration from absorbance measurements. This data also provides the growth curve for the bacterial strain.
4.4 Dilutions and Plating

In order to plate countable bacterial concentrations, the bacteria usually needs to be diluted. Dilutions are 1/10 dilutions by using 100 µLs of the starting solution and adding it to 900 µLs of PBS and vortexing. Dilutions are performed until the bacterial concentration was around 100 CFU/mL. This dilution along with the 3 previous dilutions are used for plating. Four different dilutions are plated to ensure that dilutions were performed properly (they should have approximately a 10-fold change in bacterial counts for each dilution) and to get a plate count between 40-400 CFU. This range was chosen based on observation; over 400 was hard to accurately count since the bacteria are so close together, and counts under 40 begin to lose accuracy since ±5 CFU/mL makes a difference when calculating recoveries. For experiments involving low bacterial concentrations (≤ 1000 CFU/mL), multiple plates were plated to ensure accuracy of the counts since bacterial counts sometimes were lower than 40 CFU on a plate.

Once all dilutions are performed the bacteria were then plated on nutrient agar plates. Plating was done by sectioning the plate into quadrants and then plating 50 µLs from 4 different 10-fold dilutions into each quadrant (ex: -1 dilution in 1st quadrant, -2 dilution in 2nd quadrant, and so on). The 50 µLs was plated by pipetting even sized drops in the quadrant (~15-20 drops/quadrant). The drops are allowed to dry and then the plate was placed upside down in the incubator and allowed to grow for 24-48 hours before counting the colonies.

4.5 Disk Spinning

There are a few variations of the rotating disks developed previously [7,26]. The variations will be discussed in their sections. The normal disks used for all spinning experiments, unless otherwise noted, are 12 cm in diameter and built of photopolymerizable acrylates (VeroClearTM Resin) using rapid-prototyping technology (Stratasys Objet30 Prime). The disks spun around a central axis. Each disk consisted of an annular collection chamber containing a retaining weir, a trough, a lid and a bowl region for the separation and collection of the plasma after spinning. The lid at the top of the annular collection chamber was a partial lid that did not enclose the entire disk but extended towards the center of the disk such that the chamber held 8.5 mL of fluid. The retaining weir was designed with the outer-most edge of the weir protruding into the trough to
separate the plasma and the red cell pack and prevent the red cell pack from spilling over the weir. The inner-most edge of the weir was designed with a rounded corner leading to a downward concave slope that allows the plasma to drain down into the bowl. The trough of the annular collection chamber was designed to hold 3.5 mL of fluid below the weir. The bowl was created by gently sloping down the bottom of the disk away from the center of the disk and away from the weir. The disks also contained wicking baffles which extended at an angle from the corner of the back wall and lid down to the inner-most edge of the weir to help drain the plasma into the bowl. The baffles did not extend down the inner surface of the back wall, which allowed for flow and circulation behind the baffle during the spinning process. Drawings of these disks have been published [7, 26, 76, 110].

A normal disk spin entailed spinning the disk up to 3000 rpm, the holding speed; holding the disk at 3000 rpm for 54 s, the holding time; and then decelerating the disk to a stop following a modified decaying exponential velocity profile. The rate of acceleration while spinning the disk up to the desired holding speed was always the same. The holding speed and the holding time were both varied during different experiments. Experiments in which these two parameters were varied will be labelled according to their holding speed and time. The deceleration of the disk varied greatly depending on the experiment but can be broken down into 3 major groups: original deceleration, controlled deceleration, and clotting deceleration. Each of these types of deceleration are detailed below.

Once the disk was stopped, the plasma layer that flowed over the weir into the bowl was collected by pipette into a pre-weighed test tube. The test tube was then reweighed and the difference between the values provides how much fluid was recovered in grams. Multiplying this value by 1.024 g/mL (the approximate density of the recovered plasma) gave the volume of recovered plasma. This solution was then plated for bacterial counts according to the dilution and plating procedure. The plate counts multiplied by the volume of plasma recovered gave the total amount of bacteria captured when spinning. This value divided by the total amount of bacteria delivered to the disk (see section 4.5.1 for calculation of this value) yielded the recovery rate of the bacteria from spinning. The recovery rate multiplied by 100 provided the percent bacteria recovered from spinning.
For the controlled deceleration experiments, the disk rotation was controlled by a Lab-
VIEW-powered motor control system. This allowed for reproducible and precise control of the
acceleration, hold time, maximum speed, and deceleration of the disk. The disks were mounted
on a platform connected to a Maxon 301039 Combination brushless DC motor. The speed of the
motor was read via an HEDL-5540 A12 digital encoder attached to the motor. The encoder output
signal was received into an National Instruments (NI) cRIO 9074 field-programmable gate array
(FPGA) board through an NI 9411 data acquisition (DAQ) module, which was powered by a GPS
3030D power supply. This signal was then processed by the FPGA board via a LabVIEW-based
proportional-integral-derivative (PID) controller. An analog control voltage between 0 and 10 V
(based on the PID calculations) was then transmitted via an NI 9263 DAQ module (also attached to
the cRIO FPGA board) to an Advanced Motion Controls (Camarillo, CA) CBE12A1C brushless
pulse width modulation (PWM) servo amplifier. The PWM servo amplifier, supplied with 30 V
from a GPS 2303 laboratory power supply, supplied power to the motor according to the 0 to 10 V
signal coming from the FPGA, 0 V corresponding to a resting state and 10 V corresponding to its
max speed. The motor and encoder were purchased from Maxon Motor. The FPGA board, DAQ
module, and LabVIEW software were purchased from National Instruments. The two GPS power
supplies came from Gw Instek. The custom LabVIEW program also recorded the speed of the
disk.

The motor system was updated after completing the controlled deceleration experiments.
The new system was used for clotting and complete experiments (complete experiments use a
normal deceleration). The new system was controlled by the Opto 22 PAC Control Pro software
using a custom program written in the software. The disks were mounted on a platform connected
to a Maxon EC-max 30 brushless motor. The speed of the motor was read via an HEDL-5540
A12 digital encoder attached to the motor. The motor was controlled by an ESCON Module 24/2
4-Q servo controller. Program instructions were sent to the controller and data received from the
encoder using the Opto 22 SNAP PAC hardware. The motor and encoder were purchased from
Maxon Motor. The Opto 22 software and hardware were purchased from Opto 22.
4.5.1 Disk and Blood Preparation

Unless otherwise denoted in the text, the blood and disks were prepared for spinning as follows. The disks were rinsed with water and sprayed with 85% ethanol and then immersed into a ethanol bath for about 10 seconds. The excess ethanol was shaken from the disk and the bottom side of the disk wiped dry with a paper towel. Then the top and inside of the disk were blown dry using compressed air until no residual ethanol could be detected.

The blood used for all experiments (not just spinning) was human blood donated by volunteers under an approved IRB protocol (X18-340) from Brigham Young University. In all experiments the blood was never stored in the fridge for longer than 6 hours to maintain its ”freshness”.

For spinning, the blood was removed from the fridge (if necessary) and allowed to equilibrate to room temperature. The blood was mixed by 5-7 hand inversions and then 8 mLs of blood removed and placed into a culture test tube. Bacteria was prepped according to section 4.2 and measured according to section 4.3. The blood was spiked with the necessary amount of bacteria to reach the desired bacterial loading, and mixed by hand inversions. Then 7 mLs of blood were removed from the culture test tube and place in circular ring around the center of the disk. Following the blood, 1.5 mLs of PBS were added to the disk in a ring inside the blood ring. Then the disk was placed on the motor and locked into place. The safety box (a clear plastic box) was then placed over the whole set up to protect the operator from both the blood and the disk should an accident occur.

To measure hematocrit, a portion of the blood was centrifuged in centrifuge tubes at 3328 rpm for 10 min to separate out the blood cells from the plasma. The interface of separation between the cell-pack and plasma layers was identified and the volume of each layer was calculated to determine the hematocrit. Plasma was then added to or removed from the blood tube to make a mixture with the desired hematocrit. When additional plasma was needed, a second tube of blood from the same donor was centrifuged at 3328 rpm for 10 min to provide the additional plasma, and the correct volume of plasma pipetted out from the second tube and into the first tube needing more plasma. Once the desired hematocrit was obtained, the blood was remixed by gentle hand inversions for 30 seconds.

Due to some problems encountered with the bacteria growing on agar plates after being spun in blood, a second culture test tube with 8 mLs of PBS was spiked with the same amount of
bacteria as the blood. This mixture was vortexed and then diluted and plated according to section 4.4. The plate counts multiplied by the volume of blood placed on the disk (e.g., 7 mL) gave the total amount of bacteria delivered to the disk.

### 4.5.2 Original Deceleration

The disk was then accelerated at 3.14 m/s² (500 rpm/s) until 18.85 m/s (3000 rpm) and then held for 54 seconds. During spinning two distinct vertical layers were observed to form, a transparent yellow cell-free plasma layer and a dark red cell pack layer. Then the disk was decelerated, quickly at first, and then slowly, see Figure 4.1A. Slow deceleration at low rpm was essential to avoid mixing of the clear plasma with the cell pack.

![Figure 4.1: A representative angular velocity profile for the disk under A) normal spinning conditions and under B) constant deceleration conditions. A) The disk acceleration and deceleration remained constant across all experiments. The holding phase (at 18.85 m/s) was shortened to 30 s for some experiments. B) The disk acceleration and holding phase remained constant across all experiments. Only the constant deceleration phase changed between the experiments, which only changed the rate of linear deceleration.](image)

### 4.5.3 Controlled Deceleration

The disk was accelerated at 3.14 m/s² (500 rpm/s) until 18.85 m/s (3000 rpm) and then held for 54 seconds. Once the spinning commenced, the blood was quickly flung to the inside
surface of the outer wall of the disk. The thickness of the unseparated blood layer was 2.97 mm for 8.5 mL of blood and 2.64 mm for 7 mL of blood. The blood then quickly sedimented into two layers, the cell pack layer and the plasma layer. The cell pack layer was observed as a dark red layer, while the plasma layer was observed as a transparent yellow layer. The disk was then decelerated at a constant deceleration rate of $0.628 \text{ m/s}^2$ (100 rpm/s), $0.503 \text{ m/s}^2$ (80 rpm/s), $0.314 \text{ m/s}^2$ (50 rpm/s), $0.188 \text{ m/s}^2$ (30 rpm/s), $0.126 \text{ m/s}^2$ (20 rpm/s), $0.063 \text{ m/s}^2$ (10 rpm/s), $0.031 \text{ m/s}^2$ (5 rpm/s), $0.013 \text{ m/s}^2$ (2 rpm/s), or $0.006 \text{ m/s}^2$ (1 rpm/s). Figure 4.1B shows three of the constant deceleration rates used in experiments in a disk velocity vs. time plot. The disk velocity vs. time profile differs between experiments only in the rate of deceleration used to slow the disk.

4.5.3.1 Image Acquisition

Prior to each experiment, the hollow disk was rinsed with water, sprayed copiously with 70% EtOH, agitated for 1 min, and then rinsed again with distilled water. The disks were then dried completely using clean compressed air. Once the disks were connected to a rotating platform, a Phantom v1610 camera (Vision Research, Wayne, NJ) was positioned and focused on the edge of the disk, viewing downward through the transparent, plastic upper surface of the disk into the trough region. Preliminary experiments were performed in order to mount the camera at the right height for focusing on the interface between the clarified plasma layer and the cell-pack layer. A halogen flood light was used to provide lighting for the camera. The halogen light was placed 4 inches to the right side of the disk at a 45° angle with a mirror below the disk at a 0° angle to provide optimal lighting for the camera, as shown in Figure 4.2.

The region of instability during the deceleration of a separated cell-pack layer and plasma layer in a two-layer flow was experimentally determined. A point of instability was defined as the rotational velocity (m/s) of the plasma/cell interface in the disk at the onset of a disturbance which causes mixing to occur. The compilation of the observed instability point for each deceleration rate in the deceleration versus velocity space separated stable from unstable regions. During deceleration, images of the two layers were recorded with the camera at 1000 frames per second. The camera images were stored in a continuous loop such that the previous 18 seconds were always stored in the camera memory. The operator visually observed the interface between plasma and cell-pack and stopped the camera immediately when mixing was observed. The LabVIEW pro-
gram was simultaneously stopped. Then the operator reviewed the previous 18 seconds of captured images and identified the time of the onset of instability of the interface. The video frame in which instability was first observed was designated as the time of onset of instability, and the LabVIEW recording provided the disk speed at this onset time. A disturbance was defined as a wave in the plasma/cell interface whose amplitude is equal to or greater than half the thickness of the plasma layer and resulted in mixing. Figure 4.3 shows an example of the interfacial instability. The photograph shows a mixing of the two layers by rolling billows at the interface, which is characteristic of mixing by Kelvin-Helmholtz instability [365, 366].

\[ \frac{\text{m}}{s^2} \]

4.5.4 Clotting Deceleration

The disk was accelerated at 3.14 m/s\(^2\) (500 rpm/s) and decelerated at 1.256 m/s\(^2\) (200 rpm/s). The disk was spun up and held at the separation speed and then decelerated and held
at the clotting speed. The idea here was to quickly separate the layers and then slow down the spin to maintain the bacteria in the plasma layer and allow enough time for the cell layer to clot. Thereafter, the disk can be slowed down to a stop rapidly without fear of remixing since the cell layer has clotted.

![Reflection of Cell Pack Layer through Disk](image)

Figure 4.3: Example of an instability appearing during deceleration. The faint red arcs repeating through the disk fluid are the interfacial instability, which are reminiscent of Kelvin-Helmholtz instabilities. (The dark red line near the air interface is a reflection of the cell-pack layer due to the presence of the weir.)

### 4.6 Filtering

Filtering was proposed as an additional method for separating bacteria from whole blood and as a method for bacterial concentration after plasma collection. Both whole blood and plasma were tested for their ability to be directly filtered. During the development of the filtering solution, a 25-mm track-etched filter with 0.8-µm pore diameter (Sterlitech) was used. Before filtering the blood, a filtering solution was made. This involved different salts and detergents and distilled water to a total of 45 mL of solution. The solution was vortexed and heated to 50°C to ensure everything was dissolved. Then 5 mL of the blood solution, either whole blood, 1% RBCs, or plasma, was added to the filtering solution and vortexed. The blood was then filtered through the filter using a reusable syringe filter holder (Advantec). The amount of solution that was filtered was recorded.
Experiments testing the filterability of a solution were performed without bacteria present, while the filtering for backflush experiments and whole process experiments were filtered with bacteria present.

4.7 Backflushing

For backflushing experiments, there was a blood-lysis solution, a backflush solution, and a bacterial suspension fluid. The bacterial suspension fluid varied for different experiments between PBS, a blood lysing solution (of different compositions), 1% lysed blood, 1% (v/v) lysed RBCs, and plasma. Bacteria was prepped according to section 4.2 and its concentration measured according to section 4.3. In each case, the bacterial suspension fluid was spiked with the bacteria to obtain the desired bacterial concentration. Then the bacterial suspension fluid was filtered as described in section 4.6. Three mLs of PBS (as a wash solution) were pushed through, and then the filter was “reverse filtered” (solution was pushed through backwards) using 2 mLs of the backflush solution being tested. The filters were a 25-mm track-etched filter with 0.4-µm pore diameter with either a hydrophilic (PVP-coated) or hydrophobic (uncoated) surface (Sterlitech). This is a smaller-pore filter than used during the development of the filtration solution because some preliminary whole process experiments revealed that 0.8-µm filters allowed about 5% of the bacteria to pass through the filter.

In order to determine the bacterial recovery from backflushing, the total bacteria delivered to the filter and the total bacteria removed from the filter had to be calculated. Total bacteria delivered was calculated by multiplying the bacterial concentration by the volume of the bacteria-blood solution pushed through the filter. The volume of the bacteria-blood solution delivered to the syringe was determined by sucking up into a syringe approximately 5 mLs of the bacteria-blood solution, emptying the syringe into the filtering solution (as described in section 4.6), weighing the empty syringe, and then multiplying the difference in the weights by the 1.061 g/mL (the approximate density of whole blood). The bacterial concentration of the bacteria-blood solution was determined by plate counts.

The total bacteria removed was calculated by multiplying the bacterial concentration by the volume of the backflush solution recovered in the second syringe. The volume of the recovered backflush solution recovered from the second syringe was determined by sucking up into the first
syringe approximately 2 mLs of the backflush solution, weighing the first syringe with solution and weighing the second syringe empty, pushing the solution through the filter from the first syringe into the second syringe, weighing the empty first syringe and the filled second syringe, and then multiplying the difference in the weights of the second syringe by the density of the solution (densities ranged from 1.004 g/mL to 1.009 g/mL). The bacterial concentration of the recovered backflush solution was determined by plate counts.

A comparison of the difference in the volumes (masses times density) between the first and second syringe yielded the percentage of fluid that passed through the filter. The ratio of the total bacteria recovered to the total bacteria delivered provides the recovery rate of the bacteria. This value multiplied by 100 gives the percent recovery for the experiment.

4.8 DNA Extraction Experiments

DNA removal from bacteria was developed through a bench-top method and then tested in various microfluidic device designs. The developmental bench-top procedure involved testing many different types of bacterial lysis solutions, three superparamagnetic bead types (silica-coated beads Spherotech – 2.5% w/v concentration, Dynabeads™ ThermoFisher Scientific – 40 mg/mL concentration, AbraMag™ Abraxis – 12 mg/mL concentration), and several elution buffers. The two wash solutions were constant throughout all of the tests. Wash 1 consisted of 6 M guanidine isothiocynate, 20 mM Tris-HCl, 40% (v/v) isopropanol, and distilled water up to 50 mL. Wash 2 consisted of 0.1 M NaCl, 10 mM Tris-HCl, 70% (v/v) ethanol, and distilled water up to 50 mL.

Bacteria was prepped according to section 4.2 and measured according to section 4.3. To begin, 200 µLs of washed bacteria suspended in PBS at a concentration of 1 x 10^8 CFU/mL was added to a microcentrifuge tube. Two hundred microliters of the bacterial lysis solution were then added to the microcentrifuge tube and vortexed for 5 seconds. The sample was then allowed to incubate for 5 min at room temp. Then 25 µLs of the superparamagnetic bead supension were added to the solution, followed by 200 µLs of isopropyl alcohol, and the mixture vortexed for 10 seconds. Following vortexing, the microcentrifuge tube was placed on a shaker (Gyrotory shaker model G2, New Brunswick Scientific) for 3 min, set at 100 Hz so that the microcentrifuge tubes were gently mixed. Following the shaking, the solution was then placed in a magnetic holder for 2 min to allow the superparamagnetic beads to collect in one location on the side wall of the
microcentrifuge tube. Then supernatant was then removed by pipetting and discarded. Next 450 µLs of Wash 1 was added and the microcentrifuge tube removed from the holder and vortexed for 30 seconds. The microcentrifuge tube was then placed back in the magnetic holder for 2 min for bead collection. The supernatant was removed again and discarded. Next, 450 µLs of Wash 2 was added and the microcentrifuge tube removed from the holder and vortexed for 30 seconds. The microcentrifuge tube was then placed back in the magnetic holder for 2 min for bead collection. The supernatant was removed again and discarded. Then the DNA was allowed to air dry for 5 minutes by leaving the microcentrifuge on the magnet with the top open. Finally, 50 µLs of the elution buffer was added and the microcentrifuge tube removed from the holder and vortexed for 2 min. The microcentrifuge tube was then placed back in the magnetic holder for 2 min for bead collection. The supernatant was removed into a clean microcentrifuge tube for analysis, and the microcentrifuge tube with the beads was discarded.

If the lysis solution included lysozyme, then prior to the addition of the 200 µLs of lysis buffer, 20 µLs of lysozyme (10 mg/mL) was added to the bacterial solution and vortexed for 5 s, and then incubated for 10 min on the shaker.

4.8.1 Ultrasound

Prior to the hand-off to the DNA labelling group for the NIH project, DNA was sheared using sonication. A Sonics Vibra-cell VCX400 with a model CV26 horn sonicator operating at 20 kHz with a 1/4” tip was used. First, the DNA was placed into a in-house-developed ultrasound chamber (described below). Then the sample was raised up to the ultrasound tip to be 1 mm away from the tip. Degassed distilled water was then added to fill in the gap between the tip and the plastic film in the chamber. Then the sample was sonicated for 30 s using 20% amplitude. Previous calibrations showed that 20% amplitude is about 1 W/cm² acoustic power density. The DNA solution was then pipetted out into a clean microcentrifuge tube.

The sonication chambers were built of photopolymerizable acrylates (VeroClearTM Resin) using rapid-prototyping technology (Stratasys Objet30 Prime). They consisted of a top and bottom piece. The bottom piece contained a disk-shaped well in the center of the circular piece that was designed to hold an exact amount of liquid (different sizes were used for different experiments). The top piece had a hole through the center of it, whose diameter was 1 mm larger than the tip.
of the ultrasound probe. Surrounding this hole in the top piece was a chamber built to hold the ultrasound-transmitting fluid, which was used as a medium to help couple the sonicating waves to the sample. Several different fluids were examined to obtain the best sonicating fluid. The whole in the center of the top piece allowed for different mediums to be tested for simulating the use of the sonicating system on a microfluidic device (see 4.9 for more details on the different mediums used). Strips (1” wide x 3” long) of the different polymers were placed in between the top and bottom pieces, such that the strips covered the well in the bottom piece and blocked the hole of the top piece once the device was assembled. This strip was a barrier between the lower sample chamber and the upper chamber of sonicating fluid.

**4.9 DNA Fragmentation**

The DNA samples for all fragmentation experiments were obtained by mixing 0.5 mL of 1% SDS solution with 0.5 mL of bacteria at an initial concentration of $1 \times 10^8$ CFU/mL. Bacteria was prepped according to section 4.2 and measured according to section 4.3. The lysed bacteria mixture was then used as the solution for testing fragmentation. After undergoing fragmentation, the mixture would be removed from the chamber or microcentrifuge tube and placed into a new microcentrifuge tube. The new tube would then be centrifuged (Eppendorf centrifuge 5417C) for 10 min at 11,000 rpm to spin down all cell debris and leave the DNA in the supernatant. Twenty microliters of the supernatant were used to visualize the fragment length on an agrose gel by electrophoresis. Four different methods of fragmentation were tested: stir bar, Ultra-turrax, needle, and ultrasound.

DNA was fragmented by means of a stir bar in a 3D printed chamber that was 18 mm in diameter and 4 mm deep. The stir bar was a 16.5 mm x 3.5 mm x 0.6 mm metal stir bar that could be moved by spinning a magnet underneath the chamber. The stir bar and magnet maintained magnetic coupling up to 2500 rpm, but lost coupling at higher speeds. Therefore, the speeds tested were 1000 rpm, 1275 rpm, 1575 rpm, 1850 rpm, 2150 rpm, and 2450 rpm. At all speeds the stir bar was stirred for 3 minutes. The fragmentation chamber was also used with 500-600 µm diameter polyethylene microspheres (Cospheric REDPMS - 0.98 g/cm³) added to the chamber in addition to the stir bar. The microspheres were tested at a spin speed of 1850 rpm for 3 minutes. The following amounts of microspheres were tested: 5 mg, 10 mg, 20 mg, 35 mg, and 100 µLs.
Fragmenting DNA using the Ultra-turrax (IKA T25) was performed in a microcentrifuge tube. The bacteria and SDS solution were added to a microcentrifuge tube and then the tip of the Ultra-turrax was dipped into the tube. The Ultra-turrax was run at several different speeds: 6500 rpm, 9500 rpm, 13500 rpm, and 17500 rpm for 3 min. The Ultra-turrax was also run at 6500 rpm for 15 s, 30 s, 1 min, 2 min and 10 min.

Fragmenting DNA using a syringe and needle was attempted by sucking up and expelling out the solution through the needle. Both a 25 and 30 gauge needle were used. For each needle gauge, the solution was either sucked up and expelled out twice or 5 times.

DNA was fragmented by ultrasound in two different ways: in a microcentrifuge tube and in a 3D printed chamber. Performing the ultrasound in the microcentrifuge tube was done to compare to the other types of DNA fragmentation, while the 3D printed chamber was used to determine the difference in applying ultrasound through different plastic films for eventual incorporation into the microfluidic device. The ultrasound and 3D printed chamber used are those described above in section 4.8.1. When used in conjunction with the microcentrifuge tube, a 20-kHz microtip was used at 10%, 20% and 30% amplitude, roughly corresponding to 0.5, 1.0 and 1.5 W/cm² as measured by hydrophone, respectively. The sample was then sonicated according to one of the following protocols: 15 s sonication only, 30 s sonication only, 30 s sonication followed by 90 s without sonication repeated 10 times, 45 s sonication only, 60 s sonication only, 60 s sonication followed by 90 s without sonication repeated 5 times, 75 s sonication only and 90 s sonication only. When used in conjunction with the 3D printed chamber, the 1/4-inch tip was used and the amplitude was set at 30%, 45% and 60% with different durations of time: 30 s, 60 s, 90 s, and 120 s.

4.9.1 Gel Electrophoresis

Gel electrophoresis was used to visualize the range of sizes for the fragmented DNA to determine the best fragmentation method. The gels were 1% (w/v) agarose and made by weighing out 3 g of agarose and mixing it with 300 mL of 1X TAE buffer. The mixture was then autoclaved and allowed to cool to 50 °C before being poured into the mold. This volume of agarose mixture made approximately 6-8 gels. Once the gels solidified, they were wrapped in plastic (saran wrap) and stored in the fridge until needed. A 50X TAE stock was made by mixing 242g of Tris-base
with 57.1 mL of acetic acid, 100 mL of a 0.5 M sodium EDTA solution, and then distilled water up to 1 L mark in a 1 L flask. The mixture was dissolved by stirring and, when needed, made to a 1X concentration by diluting 20 mL of the stock solution with 980 mL of distilled water.

For running the gels, the gel was placed into the electrophoresis box and the box filled with 1x TAE until the gel was covered with solution. Then 10 µLs of the samples and 6 µLs of the ladders were loaded into the wells in the gel. Prior to loading, 4 µLs of loading buffer were added to 20 µLs of sample, and the sample was mixed. Then the gels were run for 80 min at 100 V and 200 mA. The gels were then removed and soaked in a 5 µg/mL ethidium bromide solution for 30 min. After the ethidium bromide soak, the gels were destained by soaking in distilled water for 10 min. The gels were then placed under UV light and photographed. Two different ladders were used in fragmentation testing, a 100 bp ladder (N3231L New England BioLabs, 1500 bp-100 bp) and a 1 kbp ladder (N3232L New England BioLabs, 10 kbp-500 bp).

4.10 DNA measurements

4.10.1 Fluorometer

The extracted and fragmented DNA was quantitated by fluorometry. Following the Quant-iT Picogreen assay protocol (ThermoFisher Scientific) and using a 600 µL PCR tube (Fisher Scientific), 5-50 µLs of the extracted DNA was mixed with 95-50 µLs of TE buffer, followed by the addition of 100 µLs of Picogreen; and the mixture briefly vortexed. The tube was then placed in a TBS-300 fluorometer (Turner BioSystems) and the fluorescence intensity was read. The concentration of dsDNA (ng/µL) was determined using a calibrated Picogreen standard, which correlates the concentration of dsDNA with fluorescence intensity.

4.10.2 qPCR

Parameters for qPCR such as the number of cycles, cycle temperatures, and length of annealing were all optimized. Primers were evaluated with SYBR Green to optimize cycle temperatures and times. For every reaction, a master mix of 25 µLs was prepared using 2x SYBR Select Master Mix (Applied Biosystems) and the following: forward and reverse primers at 500 nM, 3
µLs target DNA, 13 µLs SYBR Green at 2x concentration and PCR H2O to 25 µLs. The mixtures were loaded into 25 µL Cepheid PCR tubes, and qPCR was performed using a SmartCycler II (Cepheid). The optimized procedure identified and used for the assay was 95°C initial denaturation for 120 s followed by 40 cycles of 95°C for 15 s, then 58°C for 30 s. These reactions were also carried out using an ABI 7900HT (Applied Biosystems) and 96 well plates. The reaction amounts and qPCR conditions for the ABI were the same as for the SmartCycler II.

4.11 Whole Process Experiments

In addition to testing and optimizing experiments individually, experiments were run to test the whole process from spinning the blood to collected DNA. The final DNA was then split into two parts and half was measured for concentration (by fluorometry or qPCR), and the other half was sent onto the next group in the grant process (Dr. Adam Woolley’s group in the Chemistry Department at BYU) for antibiotic-resistance labelling and analysis.

For these experiments, bacteria was prepped according to section 4.2 and measured according to section 4.3. The blood and disk were prepped according to section 4.5.1. The disk was spun according to section 4.5 using the deceleration profile in section 4.5.2 with a few changes: 1) the disk was only held at 3000 rpm for 30 seconds (for higher bacterial recovery); and 2) the hematocrit of the blood was measured and then adjusted to 40% hematocrit before putting the 8 mL of blood into the culture test tube. These adjustments are due to the results of another graduate student’s work [64]. After spinning, the recovered plasma underwent filtration according to section 4.6 and backflushing according to section 4.7, except in the case where less than 5 mL of plasma was recovered; then only the amount recovered (usually 4-4.5 mL) was added to the tube. After backflushing, the collected solution was split into 4 tubes (~0.4-0.5 mL per tube) and then each tube under went a DNA extraction experiment (see section 4.8) with the following modifications. The incubation time for the lysis-bacterial mixture was cut down to 1 min, Wash 2 was not used as it was found to not be necessary, and the samples were eluted into 15 µLs of a TE buffer. Everything else was done as stated in section 4.8. Following DNA collection, the 4 tubes were combined into 1 tube and sonicated according to section 4.8.1 with no alterations. After sonication, the DNA was split into 2 tubes, one for handing off to the other lab and one for measuring by fluorometer and qPCR as described in section 4.10.
4.12 Microfluidic Experiments

Experiments were performed with various microfluidic designs. These experiments mainly focused on finding the best way to lyse the bacteria, mix the lysate with beads, collect the beads, and elute the DNA—all on one device. Several experiments were run to compare the microfluidic device to the bench-top method. These experiments consisted of a whole process experiment (see section 4.11) where the sample was split into two after removal of the bacteria from the filter, with one tube going through the rest of the whole experiment process (bench-top) and the other tube going through the microfluidic device. The same volumes of each solution were used for both the bench-top and microfluidic device experiments.

4.13 Statistics

The statistical significance of experiments were analyzed using Welch’s t-test (unequal variance and unequal sample size) with statistical significance being $p \leq 0.05$. 
CHAPTER 5. BACTERIAL EXTRACTION FROM BLOOD

Removing bacteria from blood can be difficult, especially when dealing with very low clinical concentrations [7]. Chapter 2 discussed many of the different techniques that have been (or can be) used to separate bacteria from blood. Previous work in this lab [7,26] had already been performed when this dissertation began; therefore the technique of centrifugal-sedimentation was chosen as the main technique for separating bacteria from blood. Some limitations to this technique are the residual blood cells which remain in the recovered plasma [26], as well as only achieving 70% bacterial recovery, even at clinical concentrations [27]. This technique also produces about 4-5 mLs of plasma from spinning [26, 76]. The bacteria still needs to be concentrated because the 4-5 mLs of plasma is too much fluid for DNA extractions on a microfluidic device. Since spinning removes all of the WBCs present in the blood [26], lysing the remaining RBCs and then filtering the plasma was chosen as the desired technique for concentration, due to the simpleness of the process. Since filtering was chosen as the concentration technique, the filterability of whole blood was explored as an alternative to spinning the blood.

5.1 Spinning Disk

Many experiments on the spinning disk were previously performed by another graduate student [111]. The work previously done included work analyzing the chemical [26] and mechanical aspects [76] of spinning the blood on the disk. The average total bacterial recovery from these experiments is 69% and the time length of the spin was 282 s. While this is a great recovery, experiments were designed to determine if the recovery could be increased and/or the spin time shortened. Expounding upon the previous work was explored by inducing clotting while spinning the blood on the disk. Four main aspects of clotting were examined: amount of calcium added to the blood, amount of adenosine diphosphate (ADP) added to the blood, spinning parameters (speed and time), and disk style.
Ethylenediaminetetraacetic acid (EDTA) is the blood anticoagulant used in the blood collection tubes, which binds to the calcium in blood, inhibiting the coagulation and clotting cascade [367]. The first experiments tested the effects of adding a liquid solution of calcium chloride to the blood on the disk. Calcium chloride was added as an 8 M solution in either PBS or distilled water. The bacteria and disk were prepped according to sections 4.2 and 4.5.1, respectively. The disk was then spun according to section 4.5.4.

Figure 5.1 shows the total bacteria recovered in the plasma from spinning for both the PBS and distilled water trials for a total spin time of either 69 s or 117 s. The separation speed and a clotting speed for these experiments were maintained at 3000 rpm, with only the time held at the clotting speed different. These experiments were tested on both a normal disk and a disk with half the trough volume (achieved by reducing the weir height in half). A half-height weir was used because it was hypothesized that a smaller barrier could retain the semi-solid gelatinous coagulate while allowing the plasma to drain more freely.

There was only one statistically significant group different among the parameters tested: weir height for the PBS fluid with the 117 s spin. All other parameter combinations were not statistically significant, as seen in Appendix Table A.1. The average recovery for each weir height for each total spin time can be seen in Table 5.1. While the average recovery for the normal weir heights at 69 s (57.0%) and 117 s (55.6%) are lower than the previously optimized average disk recovery of 69% recovery, the total spin times of 69 s and 117 s are significantly shorter than the 282 s total spin time for the previously optimized disk spin. The total spin times of 69 s and 117 s are shorter because the blood is being clotted on the disk and can then be slowed down rapidly without any mixing occurring on the disk.

Since time did not significantly affect the recovery when spun at a single speed, different speeds were tested. With the goal of separation of the bacteria from the blood cells, the different speeds were examined by spinning the blood up to a high rpm to facilitate quick cell separation and then slowed down to a low rpm for holding at to facilitate clotting. The high rpm chosen for these experiments was 4000 rpm with a 20 s hold time to ensure good cell separation and then the disk was slowed down to either 2000 rpm, 1500 rpm, 1000 rpm or 500 rpm for the holding speed for clotting. The disk was held at the clotting speed for 60 s before being decelerated at 200 rpm/s. A normal disk was used with the regular weir height. Figure 5.2 shows the result of these
Figure 5.1: Total bacteria recovery from clotting experiments at a separation and clotting speed of 3000 rpm. Experiments use 7 mLs of whole blood with 1 mL of 8 M calcium chloride suspended in either PBS or distilled water for the clotting agent, and tests total spin times of either 69 s or 117 s for a normal disk or a half-height weir disk. Experiments using the calcium chloride suspended in PBS are blue, and calcium chloride suspended in water is grey. Squares represent a normal weir disk and triangles represent a half-height weir disk. * denotes statistical significance (p-value < 0.05).

Table 5.1: Bacterial recovery for the 3000 rpm separation and clotting speed experiments.

<table>
<thead>
<tr>
<th>Solution Type</th>
<th>Weir Height</th>
<th>Separation Speed (rpm)</th>
<th>Clotting Speed (rpm)</th>
<th>Spinning Time (s)</th>
<th>Average Bacterial Recovery</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS Normal</td>
<td>3000</td>
<td>3000</td>
<td>69</td>
<td>57.6%</td>
<td>16.7%</td>
<td></td>
</tr>
<tr>
<td>PBS Half Height</td>
<td>3000</td>
<td>3000</td>
<td>69</td>
<td>45.0%</td>
<td>21.4%</td>
<td></td>
</tr>
<tr>
<td>PBS Normal</td>
<td>3000</td>
<td>3000</td>
<td>117</td>
<td>58.7%</td>
<td>7.7%</td>
<td></td>
</tr>
<tr>
<td>PBS Half Height</td>
<td>3000</td>
<td>3000</td>
<td>117</td>
<td>40.5%</td>
<td>22.3%</td>
<td></td>
</tr>
<tr>
<td>H2O Normal</td>
<td>3000</td>
<td>3000</td>
<td>69</td>
<td>56.4%</td>
<td>14.1%</td>
<td></td>
</tr>
<tr>
<td>H2O Half Height</td>
<td>3000</td>
<td>3000</td>
<td>69</td>
<td>47.2%</td>
<td>20.7%</td>
<td></td>
</tr>
<tr>
<td>H2O Normal</td>
<td>3000</td>
<td>3000</td>
<td>117</td>
<td>53.2%</td>
<td>17.2%</td>
<td></td>
</tr>
<tr>
<td>H2O Half Height</td>
<td>3000</td>
<td>3000</td>
<td>117</td>
<td>54.3%</td>
<td>27.5%</td>
<td></td>
</tr>
</tbody>
</table>
experiments using both PBS and distilled water for the 8 M calcium chloride solution. Table 5.2 shows the average bacterial recovery for the different clotting speeds. The significantly different pairs were the 500 rpm and 2000 rpm averages for PBS, distilled water, and the fluids combined; the 1000 rpm and 2000 rpm averages for PBS, distilled water, and the fluids combined; and the 1000 rpm and 1500 rpm averages for PBS and the fluids combined. The highest average bacterial recovery achieving 65.0% recovery using PBS at a clotting speed of 1000 rpm and only took 104 s to complete. P-values can be found in Appendix Table A.2.

![Figure 5.2](image-url)

Figure 5.2: Clotting experiments using normal disk and calcium chloride, with a 4000 rpm cell separation speed and either a 500 rpm, 1000 rpm, 1500 rpm or 2000 rpm holding speed. Disk maintained at separation speed for 20 s and holding speed for 60 s. Experiments use 7 mLs of whole blood with 1 mL of 8 M calcium chloride suspended in either PBS or distilled water. Experiments using the calcium chloride suspended in PBS are blue, and calcium chloride suspended in water is grey. * denotes statistical significance (p-value < 0.05). The blue * represents significance for the PBS values, the grey * represents significance for the water values, and the black * represents significance between the rpm values grouping the PBS and water values together.
Table 5.2: Bacterial recovery for the experiments at 4000 rpm separation speed with different clotting speeds.

<table>
<thead>
<tr>
<th>Solution Type</th>
<th>Separation Speed (rpm)</th>
<th>Clotting Speed (rpm)</th>
<th>Spinning Time (s)</th>
<th>Average Bacterial Recovery</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBS</td>
<td>4000</td>
<td>500</td>
<td>111.5</td>
<td>52.7%</td>
<td>10.7%</td>
</tr>
<tr>
<td>PBS</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>65.0%</td>
<td>15.5%</td>
</tr>
<tr>
<td>PBS</td>
<td>4000</td>
<td>1500</td>
<td>105.5</td>
<td>38.2%</td>
<td>12.0%</td>
</tr>
<tr>
<td>PBS</td>
<td>4000</td>
<td>2000</td>
<td>107</td>
<td>31.8%</td>
<td>3.2%</td>
</tr>
<tr>
<td>H2O</td>
<td>4000</td>
<td>500</td>
<td>111.5</td>
<td>58.2%</td>
<td>6.5%</td>
</tr>
<tr>
<td>H2O</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>53.7%</td>
<td>11.0%</td>
</tr>
<tr>
<td>H2O</td>
<td>4000</td>
<td>1500</td>
<td>105.5</td>
<td>49.5%</td>
<td>26.6%</td>
</tr>
<tr>
<td>H2O</td>
<td>4000</td>
<td>2000</td>
<td>107</td>
<td>36.5%</td>
<td>8.7%</td>
</tr>
</tbody>
</table>

Because the 8 M calcium chloride solution in either PBS or distilled water is hypertonic, some lysing of the RBCs was noticed during the spinning. Experiments were then conducted in PBS using different solution configurations on the disk to see how this affected both RBC lysing and the bacterial recovery after clotting. RBC lysing was a value between 0 and 5 given by the operator based on how pink the recovered plasma was, with 0 representing no lysing (yellowish color) and 5 representing complete lysis (deep pink color). The different fluid configurations were 1) placing blood in a semi-circle on one half of the disk and the calcium chloride in a semi-circle on the other half; 2) placing blood in 2 dots with the calcium chloride placed in 2 dots without touching the blood; 3) placing blood in a full circle and the calcium chloride in a full circle inside the blood circle, without touching the blood circle. For the full circle configuration, pre-mixing of the solutions on the disk was also tested. On-disk mixing consisted of 3, 6 and 12 mixes before spinning up to a separation speed of 4000 rpm with a clotting speed of 1000 rpm. One mix consists of spinning the disk counter-clockwise on the disk for 1 s at 100 rpm/s and then switching directions and spinning clockwise for 1 s at 100 rpm/s. Three mixes would then consist of repeating the 1 s counter-clockwise followed by 1 s clockwise three times before spinning up to the separation speed counter-clockwise at the normal 500 rpm/s.

Figure 5.3A shows the total bacteria recovery results for the different fluid layouts and mixes. Figure 5.3B shows the RBC lysing results for the different fluid layouts and mixes, with a 0 representing no lysing and a 10 representing complete lysis. Figure 5.3C shows representa-
Figure 5.3: Experiments testing different configurations for applying blood and clotting solution to the disk. A) The total bacteria recovery results for the different fluid configurations. B) The RBC lysing results for the different fluid configurations, with a 0 representing no lysing and a 5 representing complete lysis. All experiments maintained a 4000 rpm separation speed for 20 s and a 1000 rpm clotting speed for 60 s. * denotes statistical significance (p-value < 0.05). C) The blood layout for a non-clotting experiment (PBS overlaid on top of the blood ring); the 4 dots layout, the 2 semi-circles layout, the double ring layout (calcium solution inside blood ring).

tive photos of what each layout looks like. Table 5.3 shows the average bacterial recovery for the different fluid configurations. The recovery from the full circle configuration with no pre-mixing was statistically different from the other two configurations as well as all of the full circle configurations with pre-mixing. None of the other groups differed from each other. The full circle configuration with no pre-mixing had the highest bacterial recovery of 65.0%. The recovery from the two semi-circle configuration was statistically different from the full circle configuration both with and without pre-mixing, but was not different from the 4 dots configuration. The two semi-circle configuration had the lowest RBC lysing values at an average of 1.7. P-values can be found in Appendix Table A.3.

After establishing that the full circle fluid configuration was the best configuration for bacterial recovery, experiments using adenine diphosphate (ADP) were tested with the full circle configuration. ADP is a known platelet aggregator [368]. It was hypothesized that the combination of
Table 5.3: Bacterial recovery for experiments testing different fluid configurations.

<table>
<thead>
<tr>
<th>Liquid Placement</th>
<th>Separation Speed (rpm)</th>
<th>Clotting Speed (rpm)</th>
<th>Spinning Time (s)</th>
<th>Average Bacterial Recovery</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Dots</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>42.7%</td>
<td>10.4%</td>
</tr>
<tr>
<td>2 Half Circles</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>44.3%</td>
<td>15.3%</td>
</tr>
<tr>
<td>Full Circle</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>65.0%</td>
<td>15.5%</td>
</tr>
<tr>
<td>0 mixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Circle</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>46.3%</td>
<td>15.6%</td>
</tr>
<tr>
<td>3 mixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Circle</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>39.5%</td>
<td>13.3</td>
</tr>
<tr>
<td>6 mixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Circle</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>36.7%</td>
<td>9.6%</td>
</tr>
<tr>
<td>12 mixes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Replacing the calcium and aggregating the platelets would create larger clotting aggregates faster, which would sediment out faster and trap less bacteria in the clot allowing for better bacterial recovery. The normal 1 mL of 8 M calcium chloride was used in these experiments along with a separation speed of 4000 rpm and a clotting speed of 1000 rpm. ADP was added at different volumes of a concentration of 2 g/L. The volumes were 0.25 mL, 0.5 mL and 1 mL. Table 5.4 shows the results for the different ADP amounts. Figure 5.4 shows the results for the experiments. The 0 mL ADP experiments were significantly different than the 0.5 mL and the 1 mL ADP experiments. The 0.25 mL ADP experiments were also significantly different than the 0.5 mL and the 1 mL ADP experiments. Adding ADP does not increase bacterial recovery in these experiments. P-values can be found in Appendix Table A.4.

In addition to using the regular spinning disk with a full-height weir and half-height weir, there were 6 additional disk designs examined. The first disk design was a normal disk with a full weir but the weir was altered so that none of the weir stuck into the trough region and the weir had no shelf between the trough region and the sloped region; the sloped region was pushed back so that the weir came to a point with the sloped region on one side and the trough region on the other. The second disk design was a normal disk with full weir but the weir did not go all the way around the disk, but had cut out sections where the weir was missing. The disk was divided into 12 equal sections which then alternated between having a weir and missing the weir. The third through the sixth disk designs were normal disks without a weir, and in place of the weir there
were small triangles that extended from the bottom of the disk to the top of the disk that had either 2 mm side lengths or 5 mm side lengths; there were either 16 or 32 triangles. These triangles were placed such that the volume behind the triangles would hold the same volume as the trough. The hypothesis was that this “pickett fence” would hold back clotted blood but freely allow plasma to flow down, similar to the half-height wier.

Table 5.4: Bacterial recovery for the experiments using different volumes of ADP.

<table>
<thead>
<tr>
<th>Amount of ADP (mL)</th>
<th>Separation Speed (rpm)</th>
<th>Clotting Speed (rpm)</th>
<th>Spinning Time (s)</th>
<th>Average Bacterial Recovery</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>36.5%</td>
<td>9.6%</td>
</tr>
<tr>
<td>0.5</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>29.7%</td>
<td>13.8%</td>
</tr>
<tr>
<td>0.25</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>58.7%</td>
<td>13.3%</td>
</tr>
<tr>
<td>0</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>65.0%</td>
<td>15.5%</td>
</tr>
</tbody>
</table>
These disks were tested using 1 mL of 8 M calcium chloride in PBS with separation speeds of either 4000 rpm, 6000 rpm, 7000 rpm or 8000 rpm and a clotting speed of 1000 rpm. It was hypothesized that at a certain point increasing the separation speed would begin to have adverse effects on bacterial recovery. The results are shown in Table 5.5. The normal disk with a full weir spun at the separation speeds of 4000 rpm and 6000 rpm were statistically different than the separation speeds of 7000 rpm and 8000 rpm as well as from all other disk designs at every separation speed. P-values can be found in Appendix Table A.5. Spinning the normal disk at a separation speed of 6000 rpm achieved an average bacterial recovery of 69.0% with a total spin time of 117 s. This average bacterial recovery is the same as the recovery that was achieved through the previous work in the lab, but the 117s is significantly shorter than the 282s. Spinning with the 2 mm “pickett fence” design gave high values of bacterial recovery (60% and 87%), but the large standard deviation precludes this design from any recommendation.

5.1.1 Bacterial Separation through Clotting

While there were never any clotting experiments with a reasonable standard deviation that achieved a higher average total bacteria recovery than the recovery found with previous spinning experiments, a separation speed of 6000 rpm with a clotting speed of 1000 rpm achieved the same bacterial recovery rate of 69.0% at a much faster spin time of 117 s. Applications that have a critical time component to them but are not as dependent on bacterial recovery would greatly benefit from these experiments. From the ADP experiments, it seems that the addition of ADP does not help in bacterial recovery, but it may speed up clotting. Such experiments were not tried as our application needs to recover as much bacteria as possible. It seems that mixing the calcium chloride on the disk is detrimental to the recovery of the bacteria as well as pre-exposure to calcium chloride since that starts clotting early and could potentially trap even more bacteria in the clot. One thing of note is the length of the time spinning at the desired speeds. Spinning for 20 s at 4000 rpm and then 60 s at 1000 rpm could be spinning the bacteria close to the clot such that some of the bacteria are getting caught in the clot. Also, another potential reason for the lower recovery of bacteria is that not all of the plasma could be recovered; there is a thin layer of plasma that clings to the clot which could retain some of the bacteria.
Table 5.5: Bacterial recovery for experiments using alternative disk designs.

<table>
<thead>
<tr>
<th>Weir Type</th>
<th>Disk Type</th>
<th>Separation Speed (rpm)</th>
<th>Clotting Speed (rpm)</th>
<th>Spinning Time (s)</th>
<th>Average Bacterial Recovery</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>65.0%</td>
<td>15.5%</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>69.0%</td>
<td>7.4%</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>49.6%</td>
<td>4.7%</td>
</tr>
<tr>
<td>Normal</td>
<td>Normal</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>39.9%</td>
<td>26.8%</td>
</tr>
<tr>
<td>Pointed</td>
<td>Normal</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>24.7%</td>
<td>15.2%</td>
</tr>
<tr>
<td>Pointed</td>
<td>Normal</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>44.9%</td>
<td>10.7%</td>
</tr>
<tr>
<td>Pointed</td>
<td>Normal</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>19.6%</td>
<td>8.5%</td>
</tr>
<tr>
<td>Pointed</td>
<td>Normal</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>21.6%</td>
<td>12.2%</td>
</tr>
<tr>
<td>Segmented</td>
<td>Normal</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>25.8%</td>
<td>10.7%</td>
</tr>
<tr>
<td>Segmented</td>
<td>Normal</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>48.2%</td>
<td>27.9%</td>
</tr>
<tr>
<td>Segmented</td>
<td>Normal</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>48.9%</td>
<td>13.3%</td>
</tr>
<tr>
<td>Segmented</td>
<td>Normal</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>48.1%</td>
<td>24.3%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 5mm Triangles</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>28.5%</td>
<td>13.2%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 5mm Triangles</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>55.0%</td>
<td>6.8%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 5mm Triangles</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>43.9%</td>
<td>12.7%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 5mm Triangles</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>47.6%</td>
<td>15.6%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 5mm Triangles</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>44.4%</td>
<td>20.0%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 5mm Triangles</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>28.5%</td>
<td>13.0%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 5mm Triangles</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>32.9%</td>
<td>18.3%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 5mm Triangles</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>44.3%</td>
<td>13.8%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 2mm Triangles</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>59.6%</td>
<td>45.2%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 2mm Triangles</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>42.7%</td>
<td>17.5%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 2mm Triangles</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>87.2%</td>
<td>77.4%</td>
</tr>
<tr>
<td>None</td>
<td>32 - 2mm Triangles</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>41.4%</td>
<td>16.2%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 2mm Triangles</td>
<td>4000</td>
<td>1000</td>
<td>104</td>
<td>42.0%</td>
<td>12.8%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 2mm Triangles</td>
<td>6000</td>
<td>1000</td>
<td>117</td>
<td>37.3%</td>
<td>11.2%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 2mm Triangles</td>
<td>7000</td>
<td>1000</td>
<td>117</td>
<td>23.5%</td>
<td>13.3%</td>
</tr>
<tr>
<td>None</td>
<td>16 - 2mm Triangles</td>
<td>8000</td>
<td>1000</td>
<td>117</td>
<td>45.2%</td>
<td>22.5%</td>
</tr>
</tbody>
</table>

5.2 Filtering

After spinning the blood on the disk, there is approximately 4-4.5 mLs of recovered plasma with about 1% of the RBCs still present [26]. This can be a lot of fluid to process on a microfluidic device, especially when the flow rate is 100 µL/min or less. Also, the residual blood cells have the potential to interfere with downstream processes. Therefore, to remove the remaining blood cells and concentrate the bacteria into a smaller volume, a blood-cell-lysis filtration method was developed. In addition to filtering recovered plasma, whole blood filtering was examined as a potential replacement of the spinning disk.
5.2.1 Plasma Filtering

Plasma filtering experiments were done to determine if the bacteria in the plasma could be concentrated by filtering. Plasma filtering was initially examined using the anionic detergent sodium dodecyl sulphate (SDS). This detergent was chosen since it was on the list of detergents used for bacterial lysis. Different amounts of SDS and different volumes of recovered plasma were pushed through a 0.8-µm polycarbonate track-etched (PCTE) filters (see section 4.6 for details). Table 5.6 presents the results of filtering with different SDS concentrations from 15% down to 1%.

Table 5.6: Results of filtering plasma recovered from spinning with added SDS.

<table>
<thead>
<tr>
<th>% SDS Solution (mL)</th>
<th>Amount of Plasma (mL)</th>
<th>% SDS in mixture</th>
<th>% Plasma in mixture</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>2</td>
<td>2</td>
<td>7.5</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
<td>1</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2.5</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
<td>1</td>
<td>1.2</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.67</td>
<td>67</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
<td>3</td>
<td>0.67</td>
<td>67</td>
</tr>
<tr>
<td>1</td>
<td>1.5</td>
<td>1</td>
<td>0.6</td>
<td>40</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0.33</td>
<td>67</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>0.33</td>
<td>67</td>
</tr>
<tr>
<td>1</td>
<td>1.5</td>
<td>3</td>
<td>0.33</td>
<td>67</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>0.33</td>
<td>67</td>
</tr>
</tbody>
</table>

* This percentage of the suspension was successfully filtered.

However, the ideal concentration of SDS would be a concentration that did not lyse the bacteria while still lysing the RBCs and allowing for filtration. Therefore, some bactericidal tests were performed in order to determine the highest concentration of SDS that the bacteria could withstand for a short (10 min) and long (1 hr) period of time. Figure 5.5 shows the results of this bactericidal study. SDS begins to have a significant effect on bacteria at a concentration of 0.111% (see Appendix Table A.6). This is much lower than any of the previous experiments of Table 5.6. Therefore, new experiments were performed to determine whether or not this low of a concentration of SDS would even allow recovered plasma to be filtered. Table 5.7 shows the results of filtering 1 mL of recovered plasma combined with 1 mL of the bacterial suspension at
a concentration of $1 \times 10^8$ CFU/mL in PBS with different amounts of 0.111% SDS. These results show that too much detergent can prevent filtering.

Figure 5.5: Concentration of viable bacteria measured by plate counting after suspension in different concentrations of SDS for either 10 min or 1 hr.

Another detergent evaluated for plasma filtering was saponin with and without a protease present. Saponins interact with lipids and proteins causing hemolysis when introduced into blood [369]. With this knowledge, experiments were performed examining the ability of a saponin solution to lyse RBCs well enough to filter plasma. Table 5.8 shows the results of filtering different volumes of recovered plasma with different amounts of saponin with and without *Aspergillus melleus* proteinase. The results show that the proteinase is required to get good filtration of plasma treated with saponin.
Table 5.7: Results of low SDS concentrations on filtering plasma recovered from spinning.

<table>
<thead>
<tr>
<th>% SDS</th>
<th>Amount of SDS Solution (mL)</th>
<th>Amount of Plasma (mL)</th>
<th>Amount of Bacteria (mL)</th>
<th>% SDS in mixture</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.111</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>0.096</td>
<td>66.7%</td>
</tr>
<tr>
<td>0.111</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0.083</td>
<td>34.4%</td>
</tr>
<tr>
<td>0.111</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0.067</td>
<td>100%</td>
</tr>
<tr>
<td>0.111</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.056</td>
<td>100%</td>
</tr>
<tr>
<td>0.111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.037</td>
<td>100%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.

Table 5.8: Results of recovered plasma filtering using Saponin with or without a proteinase.

<table>
<thead>
<tr>
<th>Filtering Solution</th>
<th>Amount of Filtering Solution (mL)</th>
<th>Amount of Plasma (mL)</th>
<th>% Saponin in mixture</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>2% Saponin</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4.2%</td>
</tr>
<tr>
<td>2% Saponin</td>
<td>2</td>
<td>1</td>
<td>1.33</td>
<td>6.7%</td>
</tr>
<tr>
<td>2% Saponin</td>
<td>4</td>
<td>1</td>
<td>1.6</td>
<td>8.3%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>100%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>94.6%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.

5.2.2 Whole Blood Filtering

Experiments were designed to examine processes to directly filter whole blood and skip the need for spinning the blood on the disk, since less technician involvement means less opportunity for contamination and faster processing times [133]. Also, it is assumed that if a solution is found that can allow whole blood to be filtered, then that same solution should allow the recovered plasma to be filtered as well.

Based on the results of the plasma filtering tests above, different amounts of 0.111% SDS and 2% Saponin were evaluated for whole blood filtration to determine if whole blood could also be filtered. The results for 0.111% SDS showed that even at a dilution of 1 mL of whole blood in 49 mL of 0.111% SDS only 15.6% of the solution was filtered before it clogged. From these results,
the necessary concentration of SDS for effective filtering was investigated. These results are shown in Table 5.9. It was found that 13% and 15% SDS always allowed filtering when combined with 1 mL of whole blood, but when 15% SDS was combined with the 7 mL of whole blood (compare to spinning) only about 38% of the solution could be filtered. The results for 2% Saponin with and without proteinase are presented in Table 5.10. Only 2 mLs of whole blood could be filtered if the blood dilution was smaller than 1/2 and proteinase was included in the filtration solution.

Table 5.9: Results of whole blood filtering with different concentrations of SDS.

<table>
<thead>
<tr>
<th>% SDS</th>
<th>Amount of SDS Solution (mL)</th>
<th>Amount of Blood (mL)</th>
<th>% SDS in mixture</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.67</td>
<td>1%</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>3.33</td>
<td>41.6%</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1</td>
<td>6.67</td>
<td>65%</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>1</td>
<td>8.67</td>
<td>100%</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>100%</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>7</td>
<td>3.33</td>
<td>9.2%</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>7</td>
<td>6.67</td>
<td>19.7%</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>7</td>
<td>10</td>
<td>38.1%</td>
</tr>
</tbody>
</table>

* This percentage of the suspension was successfully filtered.

Table 5.10: Results of whole blood filtering using Saponin with or without a proteinase.

<table>
<thead>
<tr>
<th>Filtering Solution</th>
<th>Amount of Solution (mL)</th>
<th>Amount of Blood (mL)</th>
<th>% Saponin in mixture</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>2% Saponin</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0%</td>
</tr>
<tr>
<td>2% Saponin</td>
<td>2</td>
<td>1</td>
<td>1.33</td>
<td>0%</td>
</tr>
<tr>
<td>2% Saponin</td>
<td>4</td>
<td>2</td>
<td>1.33</td>
<td>0%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>2</td>
<td>1</td>
<td>1.33</td>
<td>100%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>17.5%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>3</td>
<td>2</td>
<td>1.2</td>
<td>100%</td>
</tr>
<tr>
<td>2% Saponin w/ 33 mg proteinase</td>
<td>6</td>
<td>3</td>
<td>1.33</td>
<td>50%</td>
</tr>
</tbody>
</table>

* This percentage of the suspension was successfully filtered.
As mentioned in Ch. 2, RBCs are very flexible and can squeeze through spaces much smaller than their diameter. They might be squeezing into and clogging pores. Experiments were designed to test this hypothesis by creating rigid RBCs. This was done by RBC exposure to both glutaraldehyde and formaldehyde. Aldehydes cross-link the proteins in the RBCs, which increases the osmotic stability and reduces the deformability of the cells while maintaining the size, morphology, and internal structure of the cells [370, 371]. According to the results in Alizadeh’s thesis [111], WBCs are not present in the recovered plasma, only RBCs. Since RBCs are usually un-nucleated, the presence of RBCs does not present DNA contamination while collecting the bacterial DNA. Therefore, the first experiments were designed to concentrate the bacteria by simply collecting the bacteria and any RBCs present on the filter. Different concentrations of glutaraldehyde and formaldehyde were tested. Table 5.11 shows the results of filtering experiments using glutaraldehyde and formaldehyde solutions with a 0.8-µm filter. The RBCs were mixed with the aldehydes 5 minutes prior to filtering. These results were not very promising since only two of the solutions allowed some of the solution to filter. The rest of the solutions immediately clogged the filter.

Table 5.11: Results of whole blood filtering after RBC fixation using different concentrations of either glutaraldehyde or formaldehyde and a 0.8-µm filter.

<table>
<thead>
<tr>
<th>% Fixative</th>
<th>Amount of Fixative Solution (mL)</th>
<th>Amount of Blood (mL)</th>
<th>% aldehyde in solution</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>50</td>
<td>0%</td>
</tr>
<tr>
<td>50% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>25</td>
<td>0%</td>
</tr>
<tr>
<td>25% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>12.5</td>
<td>0%</td>
</tr>
<tr>
<td>12.5% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>6.25</td>
<td>0%</td>
</tr>
<tr>
<td>6.25% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>3.13</td>
<td>0%</td>
</tr>
<tr>
<td>1% Glutaraldehyde</td>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0%</td>
</tr>
<tr>
<td>38% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>19</td>
<td>0%</td>
</tr>
<tr>
<td>19% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>9.5</td>
<td>12.5%</td>
</tr>
<tr>
<td>9.5% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>4.75</td>
<td>0%</td>
</tr>
<tr>
<td>4.75% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>2.38</td>
<td>55%</td>
</tr>
<tr>
<td>2.38% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>1.19</td>
<td>0%</td>
</tr>
<tr>
<td>0.38% Formaldehyde</td>
<td>2</td>
<td>2</td>
<td>0.19</td>
<td>0%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.
The alternative idea with creating rigid RBCs was to filter out the RBCs first by size exclusion and then concentrate the bacteria on a smaller second filter. Due to the results of the previous experiments, only formaldehyde was used in these experiments. Both 5-µm and 3-µm filters were tested for filtering out the rigid RBCs. Table 5.12 shows the results for using different formaldehyde solutions on either a 5-µm or 3-µm filter. Formaldehyde volumes were found which allowed 100% of the blood-formaldehyde solution to be filtered through both 5-µm and 3-µm filters. These volumes were then used in a screening process in which the blood-formaldehyde solution was filtered through a 5-µm filter and/or a 3-µm filter. The filtrate was collected and then filtered through a 0.8-µm filter. For these experiments, all of the solution passed through the 5-µm filter and/or 3-µm filter and the percent of the filtrate that passed through the 0.8-µm filter was recorded in Table 5.12. These results show that either whole mammalian cells or fragments, whether RBCs or platelets, were getting through the exclusion filters, since only 25% of the filtrate was able to pass through the 0.8-µm filter. Since this blood cell fixation method was not performing as intended, it was abandoned in favor of searching for a blood cell lysis solution.

Table 5.12: Results of whole blood filtering after RBC fixation using different concentrations of formaldehyde and different filter sizes.

<table>
<thead>
<tr>
<th>% Formaldehyde</th>
<th>Amount of Formaldehyde Solution (mL)</th>
<th>Amount of Blood (mL)</th>
<th>% Formaldehyde in mixture</th>
<th>Size of Filter (µm)*</th>
<th>Percent Filtered**</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.5%</td>
<td>2</td>
<td>2</td>
<td>4.75</td>
<td>5</td>
<td>0%</td>
</tr>
<tr>
<td>4.75%</td>
<td>2</td>
<td>2</td>
<td>2.38</td>
<td>5</td>
<td>62.5%</td>
</tr>
<tr>
<td>2.4%</td>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>5</td>
<td>100%</td>
</tr>
<tr>
<td>2.4%</td>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>2.4%</td>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>5, 0.8</td>
<td>25%</td>
</tr>
<tr>
<td>2.4%</td>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>3, 0.8</td>
<td>25%</td>
</tr>
<tr>
<td>1.2%</td>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>5</td>
<td>100%</td>
</tr>
<tr>
<td>1.2%</td>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>1.2%</td>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>5, 0.8</td>
<td>25%</td>
</tr>
<tr>
<td>1.2%</td>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>3, 0.8</td>
<td>25%</td>
</tr>
<tr>
<td>1.2%</td>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>5, 3, 0.8</td>
<td>25%</td>
</tr>
</tbody>
</table>

* More than one value indicates that solution was filtered through a filter of the size of the first number with the filtrate then being filtered through another filter of the size of the second number, and so forth.
** This percentage of the suspension was successfully filtered.
Table 5.13: Results of whole blood filtration using Triton X-100 and different salts.

<table>
<thead>
<tr>
<th>Salt (1 M)</th>
<th>2% Triton X-100</th>
<th>4% Triton X-100</th>
<th>6% Triton X-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammonium Persulfate</td>
<td>2%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Calcium Chloride</td>
<td>100%</td>
<td>2%</td>
<td>0%</td>
</tr>
<tr>
<td>Copper Sulfate</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Lithium Chloride</td>
<td>30%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Magnesium Chloride</td>
<td>0%</td>
<td>48%</td>
<td>34%</td>
</tr>
<tr>
<td>Magnesium Oxide</td>
<td>24%</td>
<td>6%</td>
<td>30%</td>
</tr>
<tr>
<td>Magnesium Sulfate</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Manganese Chloride</td>
<td>0%</td>
<td>100%</td>
<td>28%</td>
</tr>
<tr>
<td>Nickel Chloride</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Potassium Carbonate</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Potassium Chloride</td>
<td>4%</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>Potassium Hydroxide</td>
<td>100%</td>
<td>100%</td>
<td>10%</td>
</tr>
<tr>
<td>Sodium Acetate</td>
<td>65%</td>
<td>35%</td>
<td>2%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>12%</td>
<td>6%</td>
<td>0%</td>
</tr>
<tr>
<td>Sodium Borohydride</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Sodium Carbonate</td>
<td>100%</td>
<td>100%</td>
<td>2%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>100%</td>
<td>100%</td>
<td>2%</td>
</tr>
<tr>
<td>Sodium Hydroxide</td>
<td>46%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Zinc Sulfate</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.

A more in depth search of the literature yielded a few papers where bacteria was removed from blood by filtration. Farmer et al. [47] and Sullivan et al. [48] showed that Triton X-100 and sodium bicarbonate could allow filtration of whole blood at a 1:20 ratio. This means that 100-140 mLs would need to be filtered in order to filter the necessary 5-7 mLs of whole blood, which is a lot of fluid to process. Therefore, experiments were designed to test whether the blood dilution ratio could be dropped to 1:10, as well as investigate if there were other salts that could be mixed with the Triton X-100 to allow filtration of whole blood. Table 5.13 lists the salts tested with Triton X-100 at a 1 M concentration and how well they performed. These results suggest that there is an interplay between the concentration of the detergent and the concentration of the salt that is dependent on the chemical makeup of the salt. To test how changing the salt concentration affected the ability to filter whole blood, the salts that allowed 100% filtration of the whole blood at 4% Triton X-100 were tested at 4% Triton X-100 with 0.5 M and 2 M salt concentrations. Table 5.14 shows the results of testing different salt concentrations with 4% Triton X-100. These
results confirm that the ability of a detergent-salt solution to filter whole blood is dependent on the detergent concentration, the salt concentration, and the chemical makeup of the salt.

Table 5.14: Results of whole blood filtration using 4% Triton X-100 and different salt concentrations.

<table>
<thead>
<tr>
<th>Salt</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5 M</td>
</tr>
<tr>
<td>Lithium Chloride</td>
<td>16%</td>
</tr>
<tr>
<td>Magnesium Chloride</td>
<td>70%</td>
</tr>
<tr>
<td>Potassium Carbonate</td>
<td>2%</td>
</tr>
<tr>
<td>Potassium Chloride</td>
<td>100%</td>
</tr>
<tr>
<td>Potassium Hydroxide</td>
<td>6%</td>
</tr>
<tr>
<td>Sodium Carbonate</td>
<td>2%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>100%</td>
</tr>
<tr>
<td>Sodium Hydroxide</td>
<td>14%</td>
</tr>
</tbody>
</table>

* This percentage of the suspension was successfully filtered.

The results from Table 5.13 for sodium bicarbonate suggest that the dilution of the blood may also play a role in the ability to filter whole blood, since a 1:10 blood dilution ratio did not yield the same results presented in the papers at a 1:20 blood dilution ratio [47,48]. Sodium bicarbonate was then further explored to see if the right combination of sodium bicarbonate and Triton X-100 could be found which rendered whole blood filterable at the 1:10 dilution. In addition, to the sodium bicarbonate, sodium chloride and sodium acetate were further explored to compare the difference between a basic, neutral, and acidic salt. Table 5.15 shows the results of testing different Triton X-100 amounts with different concentrations of each of the 3 salts at a 1:10 ratio with whole blood. These data show that the neutral salt sodium chloride is a more versatile salt than either the acidic salt sodium acetate or the basic salt sodium bicarbonate. Not all of the different salt and Triton X-100 mixtures that were combined with whole blood at a 1:10 ratio were successful in producing 100% filtration of the mixture. Of those combinations that produced 100% filtration of the mixture, there were 9 combinations using sodium chloride, while there were only 3 sodium bicarbonate combinations and 2 sodium acetate combinations.
Table 5.15: Results of whole blood filtering varying Triton X-100 and salt concentrations for an acidic, basic, and neutral salt.

<table>
<thead>
<tr>
<th>Salt</th>
<th>Salt Concentration</th>
<th>1% Triton X-100</th>
<th>2% Triton X-100</th>
<th>3% Triton X-100</th>
<th>4% Triton X-100</th>
<th>5% Triton X-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sodium Acetate</td>
<td>0.25 M</td>
<td>16%</td>
<td>34%</td>
<td>44%</td>
<td>22%</td>
<td>6%</td>
</tr>
<tr>
<td>Sodium Acetate</td>
<td>0.5 M</td>
<td>22%</td>
<td>26%</td>
<td>100%</td>
<td>10%</td>
<td>12%</td>
</tr>
<tr>
<td>Sodium Acetate</td>
<td>1 M</td>
<td>18%</td>
<td>65%</td>
<td>100%</td>
<td>57%</td>
<td>22%</td>
</tr>
<tr>
<td>Sodium Acetate</td>
<td>2 M</td>
<td>32%</td>
<td>72%</td>
<td>92%</td>
<td>64%</td>
<td>28%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>0.25 M</td>
<td>15%</td>
<td>100%</td>
<td>76%</td>
<td>72%</td>
<td>100%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>0.5 M</td>
<td>8%</td>
<td>100%</td>
<td>10%</td>
<td>44%</td>
<td>34%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>1 M</td>
<td>20%</td>
<td>12%</td>
<td>12%</td>
<td>6%</td>
<td>16%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>2 M</td>
<td>10%</td>
<td>6%</td>
<td>8%</td>
<td>2%</td>
<td>0%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>0.25 M</td>
<td>49%</td>
<td>3%</td>
<td>44%</td>
<td>38%</td>
<td>32%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>0.5 M</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>1 M</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>2 M</td>
<td>100%</td>
<td>79%</td>
<td>66%</td>
<td>63%</td>
<td>19%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.

All of the previous whole blood filtration experiments were performed with a blood hematocrit between 42% and 46%, which is inside the normal range. However, this technology will be used in diagnosing sick people, whose hematocrit levels can be abnormal due to dehydration and other factors [372, 373]. Therefore, filtering whole blood with exceptionally high hematocrit was also examined by adding RBCs to the blood to raise the hematocrit to 65%. While this is an unrealistic hematocrit, even for an ill person, it was postulated that a solution that would allow 65% hematocrit blood to be filtered would be useful for all hematocrit levels encountered in a clinical setting. The 16 salt and Triton X-100 combinations, which allowed 100% of the whole blood to be filtered from the table above (Table 5.15), were tested to see if they allowed successful filtration of a 1:10 dilution of 65% hematocrit whole blood. Similar to all previous whole blood filtration experiments, 5 mLs of the 65% hematocrit whole blood was filtered through a 0.8-µm filter. Table 5.16 shows the results for the filtration of high hematocrit whole blood. Only 2 of the salt and Triton X-100 combinations with 65% hematocrit blood were filterable: 0.5 M sodium chloride with 4% Triton X-100 and 1 M sodium chloride with 5% Triton X-100.

Because of the low number of salt and Triton X-100 combinations that were completely filterable when combined with 65% hematocrit blood, the influence of hematocrit was studied by
Table 5.16: Results of testing combinations of Triton X-100 and sodium carbonate, sodium chloride, or sodium acetate on 65% hematocrit blood.

<table>
<thead>
<tr>
<th>Salt</th>
<th>Molarity of Salt</th>
<th>Percent of Triton X-100</th>
<th>Percent Filtered*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sodium Acetate</td>
<td>0.5 M</td>
<td>3%</td>
<td>47%</td>
</tr>
<tr>
<td></td>
<td>1 M</td>
<td>3%</td>
<td>56%</td>
</tr>
<tr>
<td>Sodium Bicarbonate</td>
<td>0.25 M</td>
<td>2%</td>
<td>55%</td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>2%</td>
<td>100%</td>
</tr>
<tr>
<td>Sodium Chloride</td>
<td>0.5 M</td>
<td>1%</td>
<td>12%</td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>3%</td>
<td>64%</td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>4%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>5%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>1 M</td>
<td>2%</td>
<td>30%</td>
</tr>
<tr>
<td></td>
<td>1 M</td>
<td>4%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>1 M</td>
<td>5%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>2 M</td>
<td>1%</td>
<td>55%</td>
</tr>
</tbody>
</table>

*This percentage of the suspension was successfully filtered.

using the salt and Triton X-100 combinations at 3 different hematocrit levels in further filtration experiments. These hematocrit levels were 55%, 58% and 65%. For these experiments, sodium chloride was chosen as the salt.

In addition to varying the hematocrit of the blood, the sodium chloride concentration and Triton X-100 concentration were varied slightly to see how a small variation in either salt or detergent affected the filtration of the sample. Based on the previous experiments in the lab, it was hypothesized that both hematocrit and small variations in both the salt and detergent would affect the filtration of the blood. Table 5.17 shows the results of these experiment. The results show that the 0.5 M sodium chloride and 4% Triton X-100 combination is very robust when it comes to variable hematocrit, as it produced completely filterable solutions when combined with 5 mLs of whole blood with hematocrit levels of ≈ 45% (Table 5.15), 55%, 58%, and 65%. However, the results also show that slight deviations in salt concentration greatly impact the solution’s ability to allow complete filtration of 5 mLs of whole blood. However, this impact is not seen with de-
viations in detergent concentration. Five milliliters of whole blood can be filtered even with small deviations in the detergent concentration.

Table 5.17: Results of whole blood filtering using a 4% Triton X-100 and 0.5 M NaCl lysing solution at hematocrit levels of 55%, 58%, 65%.

<table>
<thead>
<tr>
<th>Hematocrit</th>
<th>NaCl Concentration</th>
<th>3.7% Triton X-100</th>
<th>4% Triton X-100</th>
<th>4.3% Triton X-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>55%</td>
<td>0.45 M</td>
<td>2%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>0.55 M</td>
<td>4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58%</td>
<td>0.45 M</td>
<td>4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>0.55 M</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>65%</td>
<td>0.45 M</td>
<td>4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5 M</td>
<td>10%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>0.55 M</td>
<td>2%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* This percentage of the suspension was successfully filtered.

5.2.3 Removing Bacteria from Blood and Plasma

Using filters to remove bacteria from blood and plasma was more challenging than initially anticipated. Too high of a detergent concentration can prematurely lyse the bacteria while not enough detergent can incompletely lyse the mammalian cells, allowing them to block the filter. Another potential problem is the ability of detergents to make complexes with the released proteins and cell membranes [374]. These complexes can precipitate out of solution depending on the concentration and amount of unbound detergent in solution [375]. This would explain why 3 mLs of a 0.111% SDS concentration in blood could be completely filtered, but 6 and 13 mLs of the same solution could not produce complete filtration. It is postulated that precipitation of lipid/protein-SDS complexes caused the decrease in the amount filtered in the experiments with higher volumes of SDS from Table 5.7. Thus, the optimum salt and detergent concentration are those that do not lyse bacteria but completely lyse the mammalian cells while not causing complexes that precipitate out of solution and clog the filter. This is a difficult challenge.
However, there are ways around the lipid/protein-SDS complex problem. Chaotropes, which are substances such as urea and guanidinium chloride that interfere with hydrophobic interactions, can weaken protein-SDS interactions, which increases the solubility of the complexes [376, 377]. In addition to making protein-detergent complexes more soluble, chaotropes also disrupt cell membranes [167, 170], which has both positive and negative outcomes—positive in that less detergent may be needed with the addition of the chaotrope, but negative in that this will increase the potential of prematurely lysing the bacteria. Another remedy would be to incorporate proteases into the solution. Table 5.8 shows the effects of adding a protease into the lysis solution. There is a drastic difference in the amount filtered with and without a protease in solution. The solutions containing a protease were almost completely filtered using the detergent-protease-plasma solution, while the solutions without a protease had less than 10% of their volume filtered. Since proteases cleave proteins, the detergents can only make complexes with portions of proteins instead of full proteins, meaning that the complexes, if any, are smaller and do not aggregate as well, thus allowing the solution to be filtered more easily.

The main difference between the filterability of whole blood and plasma recovered from spinning is the number of RBCs in suspension. Spinning the blood removes approximately 99% of the RBCs from the plasma, essentially dropping the concentration of RBCs by 2 orders of magnitude. For whole blood, not only is there 2 orders of magnitude more proteins, but there is 2 orders of magnitude more lipid membranes, which potentially may contribute to the problem. The RBC membrane composition is 19.5% (w/w) water, 39.5% protein, 35.1% lipid and 5.8% carbohydrates [378]. According to Jones [379], there is a four step interaction between cellular membranes and detergents: 1) detergent accumulates in the membrane until saturation, 2) additional detergent lyses the membrane, 3) additional detergent solubilizes the membrane, releasing lipid-detergent micelles and protein-lipid-detergent complexes, and 4) additional detergent then releases protein-detergent complexes from the protein-lipid-detergent complexes. However, there is only a two step interaction between proteins (not membrane-bound proteins) and detergents: 1) detergent accumulates on the protein saturating binding sites and 2) additional detergent unravels the protein [379]. The differences between the detergent-membrane and detergent-protein interactions suggest that more detergent is needed to interact with the membranes than with the proteins. The large increase
in the amount of membrane would then require a significant increase in the amount of detergent needed to solubilize the blood cells sufficiently enough for filtration.

The complex interactions of detergents with membranes, lipids and proteins would help explain the results of Tables 5.9 and 5.10, showing that 15% SDS and 2% saponin with proteinase could lyse 1-2 mLs of whole blood but could not lyse larger volumes of whole blood (3+ mLs). These results suggest that higher concentrations of detergent are needed, which increases the likelihood of pre-maturely lysing the target bacteria in the blood. Thus, further experiments were run that examined the ability of filters to exclude aldehyde-cross-linked RBCs. These results are in Tables 5.11 and 5.12. Table 5.11 shows that the RBCs still appear to be blocking the filter even though they are cross-linked, and thus more rigid. Vassar et al. found that the RBCs incur an initial decrease in cell volume upon aldehyde addition but they maintain their shape and lose some of their deformability at the time of fixation [370]. It was hypothesized that the RBCs would become rigid spheres from the addition of aldehyde, but the results of Table 5.11 supports the findings of Vassar et al., since the a cross-linked RBC that maintained its discoidal shape would create a filter cake that would stop fluid flow. However, Table 5.12 allowed all of the blood-aldehyde solution to pass through the 5-μm and 3-μm filters but not through the 0.8-μm filter. If the 5-μm and 3-μm filters were excluding the aldehyde-cross-linked RBCs then more solution should have been filtered through the 0.8-μm filter without any problems, unless platelets were also getting through the larger pore-sized filters and then blocking the smaller pore-sized filter. Other possibilities to consider are that either the RBC membrane is only slightly hindered by the aldehyde fixation, allowing the RBCs to still change shape enough to pass through the larger pore-sized filters, or that the RBCs are being lysed and the rigid fragments breaking into small enough sizes to pass through the larger pore-sized filters but large enough to block the smaller pore-sized filter. No matter the reason, experiments using cross-linked blood cells were stopped, and the pursuit of a blood cell lysis solution resumed.

Tables 5.13 - 5.15 show the results of testing different salts and different concentrations of salt and Triton X-100. From these results, it is clear that the combination of detergent concentration, salt concentration, and salt chemistry greatly affects how well the lysed blood cells are filtered. No studies were found in the literature which directly tested salt effects in the presence of detergents with membranes or proteins. However, insights found in the literature for salt effects on
proteins confirm that salt chemistry and concentration are key factors in the system. According to Moelbert et al., sodium chloride is a weak kosmotrope, which stabilizes proteins and hydrophobic aggregates (the opposite of chaotropes) in aqueous solutions, and can have a stabilizing effect on the detergent-protein/detergent-protein-lipid complexes that form [377]. Sammalkorpi et al. found that the concentration of the salt was important in stabilizing aggregates with increased concentrations of salt leading to larger micelle aggregates for SDS for both NaCl and CaCl₂ [380]. In studying salts made up of different anions and cations, Arakawa and Timasheff found that preferential hydration for the protein generally follows SO₄²⁻ > COO⁻ > CH₃ > Cl⁻ and Na⁺ > Mg²⁺ > Ca²⁺, Ba²⁺ > Mn²⁺, Ni²⁺ > Gu⁺ (Gu⁺ = guanidine) [381, 382]. However, these relationships are also dependent on both concentration of the salt and pH of the solution [383]. For example, MgCl₂ preferentially hydrates at concentrations below 2.5 M, while Gu₂SO₄ preferentially hydrates above 0.5 M [384]. Preferential hydration means that the protein prefers to be surrounded by water that maintains protein function and shape (stability), while preferential binding means that the protein prefers to be surrounded by the salt that causes destabilization and dissociation of the protein [384]. At a certain salt concentration, preferentially hydrated proteins maintain a liquid layer that is no longer in equilibrium with the bulk solution causing the protein to be precipitated; preferentially bound proteins can transition from preferentially bound to preferentially hydrated as the protein unravels, which can also cause the protein to be precipitated [384, 385]. Therefore, from the Arakawa and Timasheff experiments, the best salt for ensuring that the protein stays in solution would be guanidinium chloride. However, guanidium chloride can lyse bacteria and should be used with caution.

In searching the literature for detergent-membrane interactions, several studies were found. Hansen et al. found that protein-containing membranes in the presence of non-ionic detergents never completely solubilized and contained large particles between 0.5-1 μm in diameter [386]. However, this was not the case when using SDS as it completely solubilized both the proteins and the lipids [386]. This report shows that the chemical nature of the detergent is an important factor to consider when solubilizing protein-lipid membranes. Other factors which affect the solubilization of the membrane include pH, ionic strength, and temperature [387]. Another factor to consider for blood filtration is the size of the micelles formed by the detergent [387]. High molecular weight
micelles form larger structures, which when complexed to proteins or membrane fragments, have the potential to clog filters with small-diameter pores.

In experiments that directly tested the solubilization of RBC membranes, solubilization by Triton X-100 was found to be dependent on membrane composition [388–394]. Using 300 mg Triton X-100 per milliliter of packed RBC ghosts (≈ 10^{10} cells) reduced RBC ghosts from ≈ 7 µm down to 2.5-3-µm in diameter, and increasing the detergent concentration had no affect [388]. That study also found that using > 0.3 M KCl increased the amount of protein removal from the membrane [388]. Another group found that increasing Triton X-100 detergent concentrations only changed the rate at which the RBC membrane separates into an insoluble section and a soluble section. After the soluble section breaks off, the insoluble section reseals and forms a smaller but intact membrane [390]. This study also found that artificial membranes made up of lipids, sphingomyelin, and cholesterol also created insoluble sections that were temperature dependent with smaller insoluble sections at higher temperatures [390]. In a study using a 40% hematocrit RBC solution, the supernatant contained mainly detergent-membrane fragments when the Triton X-100 concentration was below 10 mM, while mixed micelles did not dominate the supernatant until Triton X-100 concentration exceeded 50 mM [395]. Two studies found that Triton X-100 only solubilized 85% of the membranes of a 20% hematocrit RBC solution even at a concentration of 80 mM [393, 394]. In comparison to Triton X-100 (non-ionic) using the zwitterionic detergent 3-[(3-Cholamidopropyl)dimethylammonio]-1-propanesulfonate (CHAPS), a 20% hematocrit RBC solution only required a 60 mM concentration of CHAPS to solubilize 95% of the RBC membranes; higher detergent concentrations of CHAPS did not increase solubilization [394].

Tables 5.16 and 5.17 show that the hematocrit affects the lysis and filtration of blood cells. In fact, some of the detergent-salt concentrations used may not have been enough to even lyse all of the blood cells at high hematocrit. Hagerstrand et al. discovered that sublytic concentrations of detergents cause shape deformations in RBC membranes, sometimes resulting in flattened discs that were thinner but larger in diameter [396]. This highlights the importance for ensuring that the detergent (with or without salt) is at an adequate level to completely lyse all of the blood cells with minimal impact upon the bacteria.

In studies on the effect of hematocrit, Prete et al. examined Triton X-100 interactions with blood at hematocrit levels of 15%, 30%, and 45%; they found that 213-µm and 255-µm of
Triton X-100 was needed for 100% hemolysis for a 30% and a 45% hematocrit level, respectively [397]. This result can be compared to the results from Casadei et al. who evaluated different concentrations of the non-ionic detergents, Brij 58 and Brij 98, against 4 different hematocrit levels 15%, 30%, 45% and 60%. They found that 37.6 µM, 43.1 µM and 47.0 µM of Brij 58 was needed to obtain 100% hemolysis for a 30%, 45% and 60% hematocrit level, respectively [398]. Furthermore, 20.4 µM, 22.2 µM and 25.1 µM of Brij 98 was needed to obtain 100% hemolysis for a 30%, 45% and 60% hematocrit level, respectively [398]. In another study by Prete et al., they tested polyoxyethylene alkyl ethers, from C_{10}E_8 (8 polyoxyethylene decyl ether) to C_{18}E_8 (8 polyoxyethylene stearoyl ether), and found that 475-µm of C_{10}E_8, 43.5-µm of C_{12}E_8, 12.5-µm of C_{14}E_8, 8.2 µM of C_{16}E_8, and 3.2 µM of C_{18}E_8 was needed to obtain 100% hemolysis for a 45% hematocrit level [399]. This shows that the Brij detergents are better hemolytic agents than Triton X-100 and C_{10}E_8, and the larger carbon-chain alkyl groups were better hemolytic agents than the Brij detergents. However, these studies used 100% hemolysis as the criteria for complete solubilization which does not account for membrane pores created by detergents that cause 100% hemolysis while the membrane has not been solubilized [395].

All of my filtration experiments above were performed on 0.8-µm track-etched filters, but it was revealed later during the initial whole process experiments (see section 8.1) that smaller-pore filters were needed as an average of about 5% of the bacteria could pass through a 0.8-µm track-etched filter, with a maximum of 20% (data not shown). Another problem that plagued some experiments was the lack of consistency in the amount of RBCs present in the recovered plasma solution. The hematocrit of the recovered plasma solution varied from less than 1% to up to 5%. Because of this some of our plasma filtration experiments varied greatly between runs, highlighting the effect hematocrit has on filterability. The filtration of blood cells depends on many factors, some of which are hematocrit, temperature, pH, salt chemistry, salt concentration, detergent chemistry, and detergent concentration.

To summarize all this research, the best plasma filtration solution was a 2% saponin with 33 mg protease which could filter 4 mLs of spun plasma at a 1:1 ratio (plasma:solution). The best whole blood filtration solution was a 4% Triton X-100 with 0.5 M NaCl which could filter whole blood at a 10:1 ratio (solution:blood). However, in light of the salt studies from the literature a 0.5
M guanidinium chloride solution could be a better salt than NaCl, but this would need to be studied further.

5.3 Backflushing

After finding a blood lysis-filtration solution, experiments were performed encompassing the whole process from blood spinning through DNA removal and recovery from bacteria (see Ch. 8). During some of the initial “whole process” experiments, removal of the bacteria from the filter after filtering recovered plasma from disk spinning was very poor (≤ 1%). Therefore experiments were designed to optimize the removal of the bacteria. Another observation indicated that higher concentrations of bacteria being filtered achieved better bacterial removal, up to 40-50%. In designing experiments to optimize and test the removal of bacteria when backflushing, it was hypothesized that the observation of better removal at higher concentrations was due to the presence of multiple layers of bacteria stacked on the filter. Therefore, subsequent experiments were designed to identify the optimum when there was a single layer or less of bacteria on the filter. The filters used are 25-mm-diameter track-etched filters with 0.4-µm pores (for details see section 4.7). According to the manufacturer of the syringe filter holder, the filtration area of the holder is 3.5 cm$^2$. *E. coli* bacteria were used for this experiment, and to simplify calculations the bacteria were oversized to approximately 3.5 µm long by 1 µm wide for an area of 3.5 µm$^2$. This means that a single layer of side-by-side bacteria on the filter would contain approximately 1 x 10$^8$ CFU. Therefore to ensure only single layer coverage with 5 mLs of solution put onto the filter, the maximum desired concentration of bacteria was 2 x 10$^7$ CFU/mL.

Initial experiments to identify trends for further analysis were performed with the bacteria suspended in PBS and only used 2 mLs of a bacterial concentration of 2 x 10$^7$ CFU/mL. The removal solutions (n=3 for all solutions) were pure water, a 3 M NaCl solution (high salt concentration), a 0.9% (w/v) NaCl solution (low salt concentration, physiological saline), a 3% (v/v) Triton-X 100 solution (high detergent concentration), and a 0.5% Triton-X 100 solution (low detergent concentration). Triton-X 100 was used because it is a non-ionic detergent. It was hypothesized that either of the detergent solutions would remove the most bacteria; however, the results came back opposite to what was expected. Figure 5.6A shows that the pure water was the best removal solution, achieving an average removal of 60%, while the detergents only achieved average recov-
eries of \( \approx 5\% \). The salt solutions were comparable to each other with an average removal of nearly 50%.

These results showed that the suspension solution (present when filtering the \textit{E. coli} through the filter) for the bacteria played an important role in being able to subsequently remove the bacteria from the filter. Therefore, the bacteria were suspended in both a 4\% Triton X-100 with 0.5 M NaCl solution and a 1\% lysed blood solution (5 mLs of whole blood suspended in 45 mLs of 4\% Triton X-100 with 0.5 M NaCl) to determine whether the presence of the detergent-salt solution or the lysed blood cell solution was preventing the removal of bacteria from the filter. Three additional detergents were added to the list of backflush solutions employed: Brij 58, Pluronic F108, and Tween 20. After testing a few of the solutions, backflushing was also performed while vibrating the filter holder apparatus to see if vibration increased removal. This was accomplished by mounting a cell-phone vibrator to the filter holder and turning it on while injecting the backflush solution.

Figure 5.6B shows the results for the 4\% Triton X-100 with 0.5 M NaCl solution and Figure 5.6C shows the results for the 1\% lysed blood solution. The results of these experiments (\( n=3 \) for each combination of suspension fluid, lysis fluid and whether the vibrator was present) showed that the lysed blood solution caused the highest retention of the bacteria on the filter. Figure 5.6D compares the backflush solutions that were similar between the water, the optimized blood lysis solution and the 1\% lysed blood solution.

When depositing bacteria from the filtering solution of 4\% Triton X-100 with 0.5 M NaCl solution, pure water achieved an average backflush removal of \( < 1\% \), while low salt achieved above 30\% removal on average. The high salt and low Triton X-100 solutions achieved an average of about 10\% removal, while the high Triton X-100 solution only achieved about 2\% removal. The other detergent solutions achieved between 40-80\% removal, with vibration usually recovering more than the non-vibrated counterpart.

When depositing bacteria from the 1\% lysed whole blood solution, pure water, low and high salt, and low and high Triton X-100 did not achieve higher than 3\% removal on average for any of them. Tween 20 and NP-40 also showed lower than 3\% removal, while Brij 58 and Pluronic F108 achieved about 16\% removal on average. Adding vibration to Pluronic F108 raised the average up to 27\% but did not increase the Brij 58 average. In addition to testing the detergents,
a combination of salt and detergent was explored with the Pluronic F108 with vibration, but adding the salt brought the average removal back down to less than 3%.

Figure 5.6: Percent bacterial removal for backflushing when applying the bacteria to the filter in different mediums: A) water, B) 4% Triton X-100 with 0.5 M NaCl, C) 1% lysed blood, D) comparative graph of the results from similar solutions in A-C. The backflushing solutions are listed on the y-axis. The x-axis is the percentage of *E. coli* removed from the filter.

The next set of experiments investigated whether the lysed cells in the solution or the proteins in the plasma were the main cause of bacteria adherence to the filter. These tests were performed by centrifuging the blood for 10 min at 3328 rpm in a Horizon 642E centrifuge (Fisher
Healthcare). This separated the RBCs from the plasma phase. Either 5 mLs of the plasma (recovered from spinning on a disk) or 5 mLs of a 1% (v/v) RBC in PBS solution was then used in place of the 5 mLs of whole blood for the 1% lysed blood solution as the suspending solution during filtration deposition. The 1% lysed RBC and 1% lysed plasma solutions were only tested with the removal solutions Brij 58 and Pluronic F108, both with and without vibration, based upon the results from the previous set of experiments. Figure 5.7A shows the removal efficiencies for Brij 58 and Pluronic F108, both with and without vibration, with either the 1% lysed RBCs, 1% lysed plasma or 1% lysed whole blood as the bacterial deposition fluid. Figure 5.7B shows the results of pre-soaking the filter in detergent solution for 5 min. In general, pre-soaking the filter greatly enhanced bacterial backflushing from the filter.

![Figure 5.7](image)

**Figure 5.7:** A) Percent bacterial removal for backflushing when using either lysed blood, lysed RBCs or recovered plasma after spinning mixed with lysing solution as the medium for bacterial filtering. B) Comparison of pre-soaking in the respective backflush solution or not soaking the filters before filtering and backflushing when delivering the bacteria in a 1% lysed RBC solution.

Figure 5.7 revealed that deposition from the 1% RBC in PBS solution produced more adhesion to the filter than adhesion from the 1% whole blood solution or the 1% lysed plasma solution. Therefore, subsequent experiments were focused on challenging the most adhesive deposition solution, a 1% lysed RBC solution, for further optimization of bacterial removal from the filter. In addition to testing backflushing solutions, subsequent experiments included evaluating different de-
tergents in the deposition solution and evaluating different types of filters. The different detergents tested for the deposition solution were Brij 58 at 1% and 3% (w/v), NP-40 at 1% and 3% (v/v), Pluronic F108 at 1% and 3% (w/v), Triton X-100 at 1% and 3% (v/v), Tween-20 at 1% and 3% (v/v), and Tween-80 at 1% and 3%(v/v). The deposition solution also contained 0.5 M NaCl along with the detergent. The two different types of filters tested were 0.4 \mu m hydrophilic track-etched filter (polycarbonate filter with a PVP surface-coating) and 0.4 \mu m hydrophobic track-etched filter (polycarbonate filter with no surface-coating). Figure 5.8 shows the removal efficiencies for each detergent combination using the hydrophilic filter. Figure 5.9 shows the removal efficiencies for each detergent combination using the hydrophobic filter.

From the results in Figures 5.8 and 5.9, the best non-pre-soaked detergent combinations were chosen for testing at bacterial concentrations of 20,000 CFU/mL, 2,000 CFU/mL, and 200 CFU/mL. Only non-pre-soaked combinations were chosen for the low count experiments because pre-soaking the filters requires extra steps which increases the chance for mistakes and contamination in clinical settings. The results for the low bacterial concentration experiments can be seen in Figure 5.10. Additional experiments were performed in which the salt concentration in the deposition solution was varied from the usual 0.5 M NaCl concentration to 0 M and 2 M to see how salt affected the removal of the bacteria when backflushing. It was hypothesized that the presence of salt in the deposition fluid reduced the length-scale of charge repulsion and allowed the bacteria to get closer to the surface of the filter and thus adhered to or trapped on the filter. The results verified this hypothesis as the higher salt solutions correlated with decreased bacterial removal from the filter while no salt increased bacterial removal from the filter. Figure 5.11 shows the results of the high, low and no salt experiments for the three best solutions from the low count experiments.

Because the end goal is to collect the bacterial DNA, backflushing with a bacterial lysis solution was examined to determine if the two steps could be combined. However, these experiments (data not shown) resulted in less than 1% of the DNA being collected from the bacteria on the filter, as compared to the DNA collected in a control tube in which the same amount of bacteria were suspended and lysed. It appears that either the DNA may associate with the filter and not easily come into solution, or that the lysis solution does not mix well enough with the bacteria stuck on the filter to fully lyse them.
Figure 5.8: Percent bacterial removal when testing different detergents in the deposition solution with 1% lysed RBCs in PBS and different backflushing solutions on a normal hydrophilic (PVP-coated) 0.4 \( \mu \text{m} \) track-etched filter. Solid colors represent not soaked filters and dashed colors represent pre-soaked filters in the backflush solution.
Figure 5.9: Percent bacterial removal when testing different detergents in the deposition solution with 1% lysed RBCs in PBS and different backflushing solutions on a hydrophobic (PVP-free) 0.4 \( \mu \text{m} \) track-etched filter. Solid colors represent not soaked filters and dashed colors represent pre-soaked filters in the backflush solution.
5.3.1 Removing Bacteria from the Filter

The are many factors which affect bacterial adhesion to filter surfaces, such as bacterial concentration, time of exposure, shear stress, temperature, surface chemistry of both filter and bacteria, pH, ionic strength of solution, presence of proteins or other molecules, and roughness of the filter surface [62, 400]. The filtration solution which contains the bacteria will consist of many proteins and lipids from the lysed blood cells (see the discussion for section 5.2) as well as the detergent and salt present. Along with the bacteria, each of these substrates will interact with the filter surface, potentially modifying the surface. With so many variables present, it is hard to determine what is the major factor in preventing the removal of the bacteria from the filter. However, the results of Figure 5.6 lead to the conclusion that the proteins and lipids from the lysed blood cells are a key component in bacterial adherence to the filter. How these molecules are affecting the bacteria and/or filter surface can not be determined from the experiments performed, but the results of Figure 5.7B show that the interaction of the detergent with the filter does counteract...
some of the protein and lipid interactions that affect bacterial adherence. Pretreating filters with nonionic detergents has been shown to reduce the adhesion of particles onto the filter surface [61]. Figure 5.7A also shows that the use of vibration positively correlates with bacterial removal, which is similar to the results of [401].

![Graph](image)

Figure 5.11: Percent bacterial removal using the 3 best deposition and backflush solution combinations from Figure 5.10 containing no salt, 0.5 M NaCl or 2 M NaCl along with 1% lysed RBCs in the deposition solution. The bacterial count was 200 CFU/mL. The deposition solution is listed on top and the backflush solution is listed on bottom on the y-axis. Error bars are the standard deviation.

An interesting phenomena is that bacteria usually adhere better to hydrophobic surfaces than hydrophilic surfaces [402], but Figures 5.8 and 5.9 show that higher bacterial removal was achieved with the hydrophobic filters. From this data, it is postulated that the proteins and lipids in solution favorably interact with the PVP on the surface of the hydrophilic filters, while the lack of the PVP on the surface of the hydrophobic filters limits protein and lipid interaction. A published study on plasma proteins and their effects on bacterial adherence to polycarbonate surfaces showed that the proteins decreased adherence to the surface while increasing bacterial surface charge [403]. However, another published study on protein-coated polymer surfaces showed that surfaces with proteins had greater bacterial adherence [404]. Also, as the time of contact between
the bacteria and the surface increased, bacterial adherence to the surface increased, but increases in ionic strength decreased bacterial adherence to the surface [404]. In another study analyzing the difference between reversible and irreversible adherence to glass surfaces, high ionic strength was shown to be associated with irreversible adherence [405]. Thus, it is hard to say exactly what is happening at the filter surface.

Other studies have shown that polymeric substances, lipopolysaccharides, and proteins in the outer membrane of bacteria increase bacterial adhesion to surfaces [406–408]. Thus, the partial solubilization of the bacterial outer membrane may be advantageous in the removal of bacteria from the filter, as long as the bacteria are not lysed. Or potentially the inclusion of protein denaturing substances, like urea and guanidinium chloride, could help improve bacterial removal.

To summarize the experiments performed, the best solution for increasing bacterial removal would be a deposition solution of 3% Tween 80 without any salt with a backflush solution of 3% Pluronic F108 with presoaking the filter in the backflush solution and the inclusion of vibration while backflushing. The recommended filter is a hydrophilic track-etched 0.4-µm-pore filter.
CHAPTER 6. SEPARATION INSTABILITY

In this chapter, we solve and model the equations of motion for two-fluid flow with a moving wall boundary and a free surface. We mathematically explore the presence of instabilities in the system through a linear stability analysis. We empirically develop a model for non-Newtonian fluid adjacent to the wall. Then we experimentally determine stability for two fluid flow with a moving wall boundary and a free surface in a hollow disk. Multiple deceleration rates for the moving wall boundary are investigated experimentally, along with the parameters of cell pack thickness, plasma thickness, and total thickness. The majority of the material in this chapter was published in AICHE Journal volume 65 pages 1376-1386, 2019 (see reference [110]).

6.1 Theoretical Velocity Profile

The velocity profile for a two-layer flow with a moving wall boundary and a free surface is mathematically determined from spin up through the hold phase to deceleration. As mentioned in previous chapters, there are multiple types of cells in blood. Two layers are created during the centrifugation-sedimentation of the blood, a cell-pack layer and a plasma layer. These layers are created after about 20 seconds of spinning on the disk. Each layer has a different density and viscosity with the cell-pack layer being denser and more viscous than the plasma layer. The cell-pack layer consists mainly of RBCs and WBCs, while the plasma layer consists mainly of the plasma. Figure 6.1 shows the layout of the fluid on the disk used in the development of the velocity profile.

In order to analytically solve the velocity profile for the disk, several assumptions and simplifications are used. First, as mentioned previously, the disk has a 12 cm diameter, while the total thickness of the fluid on the disk is 2.97 mm for 8.5 mL of fluid and 2.64 mm for 7 mL of fluid. The ratio of the fluid thickness to the radius at the disk wall (5.9 cm) is 0.05. The disk is maintained at a speed of 3000 rpm for 60 s prior to the deceleration of disk, sufficient time
Figure 6.1: Geometry for modeling the two-layer-flow configuration.

to create separation and damp out any residual flows from centripetal acceleration. The small thickness-to-radius ratio and the combination of no centripetal effects allows the velocity profile to be developed in 2-D Cartesian coordinates. Second, the cell-pack layer is treated as an immiscible continuum fluid with zero surface tension between the two layers. Third, both fluids are treated as Newtonian fluids with the analytical equation. While this is a good approximation of the plasma layer [409–411], this approximation does not hold for the cell-pack layer. In order to obtain a more realistic approximation for modeling the cell-pack layer, the Casson equation is used (see the Disk Modeling section for details).

With the above mentioned assumptions, the equations of motion for the system become:

$$\frac{\partial u_j}{\partial t} + u_j \frac{\partial u_j}{\partial x} + v_j \frac{\partial u_j}{\partial y} = -\frac{1}{\rho_j} \frac{\partial p_j}{\partial x} - \frac{1}{\rho_j} \frac{\partial \tau_{yxj}}{\partial y},$$  \hspace{1cm} (6.1)

$$\nabla \cdot u_j = 0,$$  \hspace{1cm} (6.2)

$$\frac{\partial v_j}{\partial t} + u_j \frac{\partial v_j}{\partial x} + v_j \frac{\partial v_j}{\partial y} = -g_c - \frac{1}{\rho_j} \frac{\partial p_j}{\partial y} - \frac{1}{\rho_j} \frac{\partial \tau_{xyj}}{\partial x},$$  \hspace{1cm} (6.3)

$$\nabla \cdot v_j = 0,$$  \hspace{1cm} (6.4)

in which $u$ and $v$ are the velocity components in the $x$- and $y$-directions, respectively, $t$ is the time, $\rho$ is the fluid density, $p$ is the pressure, $g_c$ is the centrifugal acceleration produced by spinning,
τ is the shear stress, and the subscript \( j \) (\( j = 1, 2 \)) represents the plasma and cell-pack fluids, respectively. \( g_c \) has a magnitude of \( \Omega^2 R \), in which \( \Omega \) is the angular velocity and \( R \) is the local radius of the rotating disk. The direction of increasing \( y \) is the direction towards the center of the disk. The \( x \)-direction is then the horizontal flow in the \( \Theta \)-direction of the rotating disk. The motion of the disk, and the fluids themselves, is inherently 2-D, and stability analyses done by Yih [412] and Schaflinger [413] indicate that two dimensional systems accurately predict the instabilities. For these reasons, the velocity profile is developed in 2-D, and the \( z \)-direction (perpendicular to the disk) is ignored.

The basic flow state has only one velocity component \( u \), in the \( x \)-direction and only depends spatially on the \( y \)-direction, i.e. \( u_j = u_j(y,t) \) and \( v_j = 0 \). The pressure in the system \( p \) is constant in the \( x \)-direction and changes in the \( y \)-direction, \( p_j = p_j(y) \). Plugging these assumptions into the equations of motion leaves us with the following equations for the flow,

\[
\frac{\partial u_j}{\partial t} = -\frac{1}{\rho_j} \frac{\partial \tau_{xy}}{\partial y}, \tag{6.5}
\]

\[-\rho_j g_c = \frac{\partial p_j}{\partial y}. \tag{6.6}\]

This equation of motion is the starting point for solving the flow velocities during spin up, the hold phase and deceleration. For spin up, the flow is a single fluid and the cells do not begin sedimenting out until the fluid cannot move in the \( y \)-direction anymore (see section B.1.1). This occurs sometime during the spin up phase and is used as the starting point for solving the spin up velocity profile. However, sedimentation is ignored for all phases due to treating the cell-pack layer as a fluid and not cells. During the hold phase, the cell-pack layer forms about 15-20 seconds (depending on hematocrit [64]) into the hold phase and maintains the separation for the duration of the phase (see section B.1.2). Both the spin up and the hold phase velocity profiles are solved using only a single layer, the plasma layer. This is because the momentum per volume of each fluid is defined by \( \rho u \), where \( \rho \) is the density and \( u \) is the velocity. Therefore the cell-pack layer, which has a larger density (\( 1108 \text{ kg/m}^3 \)), will transfer momentum faster than the plasma layer, which has a smaller density (\( 1024 \text{ kg/m}^3 \)), by a factor of approximately 1.08. The equations for solving the spin up and hold phase velocity profiles can be found in Appendix B. The deceleration phase then begins with the two separated layers and the velocity profile is solved with the two separated layers.
For all phases, the boundary condition at the wall will employ a “no-slip” condition, and the air interface will employ a free surface boundary condition, which will be modeled as no momentum transfer. For the deceleration phase with two layers, the liquid-liquid interface boundary conditions will employ continuity in both velocity and shear stress.

6.1.1 Derivation of 2-D Velocity Profile during Deceleration phase

6.1.1.1 Problem Set Up

We seek solutions to (6.5) for \( j = 1, 2 \) with the four boundary conditions, two for each fluid, given by (6.7)-(6.10), and initial condition \( u_j(t = 0) = U_0 \) everywhere (included in the wall boundary condition). As mentioned above, the deceleration phase begins with 2 separated layers. It is important to note that the denser “fluid” is the cell-pack layer and will form against the moving wall of the disk, and the lighter fluid is the plasma layer and will form against the cell-pack layer with the other side remaining unbounded (air interface, see Figure 6.1). The velocity equations are developed with the wall boundary decelerating at a constant rate, to match experiments that were performed with constant rotational deceleration. The constant deceleration results in a constant wall shear stress, meaning that the wall shear rate for the denser fluid is not dependent on time. Since this is a transient problem initially the \( \tau \) and \( \dot{\gamma} \) away from the wall are dependent upon both time and space. After a long enough time of deceleration (\( \approx 10 \) seconds), the \( \tau \) and \( \dot{\gamma} \) have nearly approached steady state values, so they are not dependent on time (see Figure 6.5C).

These considerations lead to the following boundary conditions: at the wall,

\[
 u_2(-d_2) = U_0 - kt; \quad (6.7)
\]

at the liquid-liquid interface,

\[
 u_2(0) = u_1(0), \quad (6.8)
\]

\[
 \mu_2 \frac{\partial u_2(0)}{\partial y} = \mu_1 \frac{\partial u_1(0)}{\partial y}; \quad (6.9)
\]
and at the air interface,
\[ \frac{\partial u_1(d_1)}{\partial y} = 0; \]  
(6.10)

where \( k \) is a deceleration rate, \( \mu_j \) is the viscosity of the respective fluid, and \( d_j \) is the thickness of the respective fluid. The initial condition for the flow is \( u_2(t = 0) = u_1(t = 0) = U_0 \), where \( U_0 \) is the wall velocity prior to the onset of deceleration (see application of the constant velocity hold derivation section, section B.1.2), and this condition is included in the wall boundary condition.

The following parameters for 2-layer flow are used to non-dimensionalize the flow equations.

\[ \kappa_j = \frac{u_j}{\sqrt{kd_2}}, \quad \eta = \frac{y}{d_2}, \quad \xi = \frac{d_1}{d_2}, \quad \zeta = \frac{\mu_1}{\mu_2}, \quad \beta = \frac{v_1}{v_2}, \quad \epsilon = \frac{v_2t}{d_2}, \quad \text{Re} = \frac{k^2d_2^3}{\nu_2} \]

where \( \nu_1 \) is the kinematic viscosity for fluid 1 (\( \mu_1/\rho_1 \)) and \( \nu_2 \) is the kinematic viscosity for fluid 2 (\( \mu_2/\rho_2 \)). The non-dimensional equations for (6.5) for \( j = 1, 2 \) and for the boundary conditions (6.7)-(6.10) are:

\[ \frac{\partial \kappa_1}{\partial \epsilon} = \beta \frac{\partial^2 \kappa_1}{\partial \eta^2}, \]  
(6.11)

\[ \frac{\partial \kappa_2}{\partial \epsilon} = \frac{\partial^2 \kappa_2}{\partial \eta^2}, \]  
(6.12)

\[ \kappa_2(-1) = \frac{U_0}{\sqrt{kd_2}} - \epsilon \text{Re}, \]  
(6.13)

\[ \kappa_2(0) = \kappa_1(0), \]  
(6.14)

\[ \frac{\partial \kappa_2(0) }{\partial \eta} = \zeta \frac{\partial \kappa_1(0) }{\partial \eta}, \]  
(6.15)

\[ \frac{\partial \kappa_1(\xi) }{\partial \eta} = 0. \]  
(6.16)

In order to accommodate the inhomogeneous boundary condition, we will let

\[ \kappa_2 = w + v_2, \quad \kappa_1 = w + v_1, \]  
(6.17)

\[ w(-1) = \frac{U_0}{\sqrt{kd_2}} - \epsilon \text{Re}, \]  
(6.18)

\[ v_1(-1) = 0, \]  
(6.19)
The simplest solution to the boundary conditions for \( w \) is a linear relationship between them, \( w = c_0 + c_1 \eta \). Applying the boundary conditions, we obtain:

\[
\begin{align*}
  w'(\xi) &= c_1 = 0 \\
  w(-1) &= c_0 = \frac{U_0}{\sqrt{k d_2}} - \epsilon Re \\
  w &= \frac{U_0}{\sqrt{k d_2}} - \epsilon Re
\end{align*}
\]  

(6.20)  
(6.21)  
(6.22)

We then split \( \kappa_1 \) and \( \kappa_2 \) into their homogeneous, \( v \), and inhomogeneous, \( w \), components to obtain:

\[
\begin{align*}
  \frac{\partial v_1}{\partial \epsilon} &= \beta \frac{\partial^2 v_1}{\partial \eta^2} - \frac{\partial w}{\partial \epsilon} \\
  \frac{\partial v_2}{\partial \epsilon} &= \frac{\partial^2 v_2}{\partial \eta^2} - \frac{\partial w}{\partial \epsilon}
\end{align*}
\]  

(6.23)  
(6.24)

We first solve the eigenvalue problem associated with the homogeneous form of (6.23) and (6.24), and then use that information to solve the non-homogeneous case which includes the term \( \frac{\partial w}{\partial \epsilon} = Re \) as well as the temporal derivative.

### 6.1.1.2 Eigenfunction expansion technique

We are interested in solving the eigenvalue problem given by:

\[
\begin{align*}
  \partial_\eta \eta \phi_m &= -\lambda_m^2 \phi_m \\
  \beta \partial_\eta \eta \theta_m &= -\lambda_m^2 \theta_m,
\end{align*}
\]  

(6.25)  
(6.26)

where \( \phi(\eta) \) is the eigenfunction for the plasma-layer velocity and \( \theta(\eta) \) is the eigenfunction for the cell-pack-layer velocity, with homogeneous boundary conditions:

\[
\begin{align*}
  \phi_m(-1) &= 0, & \theta_m'(\xi) &= 0, \\
  \phi_m(0) &= \theta_m(0), & \phi_m'(0) &= \xi \theta_m'(0).
\end{align*}
\]  

(6.27)
Once we have determined \( \phi_m(\eta) \) and \( \theta_m(\eta) \), we note that the full solution of (6.23) and (6.24) can be written as an expansion of these functions, i.e.:

\[
v_1(\epsilon, \eta) = \sum_m c_m(\epsilon) \phi_m(\eta)
\]
(6.28)

\[
v_2(\epsilon, \eta) = \sum_m c_m(\epsilon) \theta_m(\eta),
\]
(6.29)

which will reduce (6.23) and (6.24) to

\[
\sum_m c'_m(\epsilon) \phi_m(\eta) = - \sum_m \lambda^2_m c_m(\epsilon) \phi_m(\eta) + Re
\]
(6.30)

\[
\sum_m c'_m(\epsilon) \theta_m(\eta) = - \sum_m \lambda^2_m c_m(\epsilon) \theta_m(\eta) + Re.
\]
(6.31)

Using the orthogonality of \( \theta_m(\eta) \) and \( \phi_m(\eta) \) we can then determine the temporal coefficients \( c_m(\epsilon) \).

### 6.1.1.3 Finding the eigenfunctions and eigenvalues

The solutions to the eigenvalue problem (6.25) and (6.26) have the form:

\[
\phi_m(\eta) = A \sin \left[ \frac{\lambda}{\sqrt{\beta}} (\xi - \eta) \right] + B \cos \left[ \frac{\lambda}{\sqrt{\beta}} (\xi - \eta) \right],
\]
(6.32)

\[
\theta_m(\eta) = C \sin [\lambda (\eta + 1)] + D \cos [\lambda (\eta + 1)].
\]
(6.33)

Applying \( \phi'_m(\xi) = 0 \) and \( \theta_m(-1) = 0 \) leaves us with:

\[
\phi_m(\eta) = B \cos \left[ \frac{\lambda}{\sqrt{\beta}} (\xi - \eta) \right],
\]
(6.34)

\[
\theta_m(\eta) = C \sin [\lambda (\eta + 1)],
\]
(6.35)

and using \( \phi_m(0) = \theta_m(0) \) yields:

\[
C = B \frac{\cos \left[ \frac{\lambda}{\sqrt{\beta}} \xi \right]}{\sin [\lambda]},
\]
(6.36)
so that
\[
\theta_m(\eta) = B \cos \left[ \frac{\lambda_m \xi}{\sqrt{\beta}} \right] \sin \left[ \lambda (\eta + 1) \right].
\] (6.37)

Applying the final boundary condition: \( \zeta \phi_m' = \theta_m' \) at \( \eta = 0 \) gives an equation for \( \lambda \):
\[
\frac{\sqrt{\beta}}{\zeta} = \tan \left[ \frac{\lambda_m \xi}{\sqrt{\beta}} \right] \tan[\lambda_m].
\] (6.38)

This means that \( \lambda_m \) (there are a countably infinite number of these) is dependent on the parameters: \( \xi \), \( \zeta \), and \( \beta \). The problem is that (6.38) is very ill-conditioned numerically due to the singularity of the tangent function. To fix this, we use some carefully configured trigonometric identities to see that (6.38) is equivalent to:
\[
(1 - \sqrt{\beta}) \cos \left( \lambda_m \left[ \frac{\xi}{\sqrt{\beta}} - 1 \right] \right) = (1 + \sqrt{\beta}) \cos \left( \lambda_m \left[ \frac{\xi}{\sqrt{\beta}} + 1 \right] \right).
\] (6.39)

The constant \( B \) (really \( B_m \) from (6.37)) is chosen to normalize these eigenfunctions so that
\[
\int_{\xi}^{0} \phi_m^2(\eta) d\eta + \int_{-1}^{0} \theta_m^2(\eta) d\eta = 1.
\] (6.40)

Solving this equation for \( B_m \) yields:
\[
B_m = \left\{ \frac{\xi}{2} + \frac{\sqrt{\beta}}{4\lambda_m} \sin \left( \frac{2\lambda_m \xi}{\sqrt{\beta}} \right) + \frac{\cos^2 \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right)}{\sin^2(\lambda_m)} \left[ \frac{1}{2} - \frac{1}{4\lambda_m} \sin(2\lambda_m) \right] \right\}^{-1/2}.
\] (6.41)

6.1.1.4 Determining the temporal coefficients

Now, returning to the temporal coefficients, we identify
\[
f_m = B_m Re \left[ \frac{\sqrt{\beta}}{\lambda_m} \sin \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) - \cos \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) \right],
\] (6.42)
so that each of the \( c_m(\varepsilon) \) satisfy the differential equation

\[
c'_m(\varepsilon) = -\lambda_m^2 c_m(\varepsilon) + f_m,
\]

which has solution given by

\[
c_m(\varepsilon) = \frac{f_m}{\lambda_m^2} \left( 1 - e^{-\lambda_m^2 \varepsilon} \right).
\]

Hence, putting all the pieces together we arrive at the solution given by (6.45) and (6.46). Written out in its final form gives:

\[
\begin{align*}
\kappa_1(\varepsilon, \eta) &= \frac{U_0}{\sqrt{kd_2}} - \varepsilon Re + \sum_m \frac{Re \left( \frac{\sqrt{\beta}}{\lambda_m} \sin \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) - \cos \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) \left[ \cos (\lambda_m) - 1 \right] \right)}{\lambda_m^2 \left\{ \frac{\xi}{2} + \sqrt{\frac{\beta}{4\lambda_m}} \sin \left( \frac{2\lambda_m \xi}{\sqrt{\beta}} \right) + \cos^2 \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) \left[ \frac{1}{2} - \sin \left( 2\lambda_m \right) \right] \right\}} \left( 1 - e^{-\lambda_m^2 \varepsilon} \right) \cos \left( \frac{\lambda_m}{\sqrt{\beta}} (\xi - \eta) \right), \\
\kappa_2(\varepsilon, \eta) &= \frac{U_0}{\sqrt{kd_2}} - \varepsilon Re + \sum_m \frac{Re \left( \frac{\sqrt{\beta}}{\lambda_m} \sin \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) - \cos \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) \left[ \cos (\lambda_m) - 1 \right] \right)}{\lambda_m^2 \left\{ \frac{\xi}{2} + \sqrt{\frac{\beta}{4\lambda_m}} \sin \left( \frac{2\lambda_m \xi}{\sqrt{\beta}} \right) + \cos^2 \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right) \left[ \frac{1}{2} - \sin \left( 2\lambda_m \right) \right] \right\}} \left( 1 - e^{-\lambda_m^2 \varepsilon} \right) \sin \left[ \lambda_m (\eta + 1) \right] \frac{\cos \left( \frac{\lambda_m \xi}{\sqrt{\beta}} \right)}{\sin (\lambda_m)}.
\end{align*}
\]
The dimensional equations, which are used for plotting, are:

\[
\begin{align*}
&u_1(t, y) = U_0 - kt + k \sum_m \frac{\sqrt{V_1}}{\lambda_m} \sin \left( \frac{\lambda_m d_1}{\sqrt{V_1}} \right) - \cos \left( \frac{\lambda_m d_1}{\sqrt{V_1}} \right) \frac{\sqrt{V_1}}{\lambda_m} \left[ \cos \left( \frac{\lambda_m d_2}{\sqrt{V_2}} \right) - 1 \right] \\
&\quad \cdot \left( 1 - e^{-\lambda_m^2 t} \right) \cos \left[ \frac{\lambda_m}{\sqrt{V_1}} (d_1 - y) \right] \\
&u_2(t, y) = U_0 - kt + k \sum_m \frac{\sqrt{V_1}}{\lambda_m} \sin \left( \frac{\lambda_m d_1}{\sqrt{V_1}} \right) - \cos \left( \frac{\lambda_m d_1}{\sqrt{V_1}} \right) \frac{\sqrt{V_1}}{\lambda_m} \left[ \cos \left( \frac{\lambda_m d_2}{\sqrt{V_2}} \right) - 1 \right] \\
&\quad \cdot \left( 1 - e^{-\lambda_m^2 t} \right) \sin \left[ \frac{\lambda_m}{\sqrt{V_2}} (y + d_2) \right] \frac{\cos \left( \frac{\lambda_m d_1}{\sqrt{V_1}} \right)}{\cos \left( \frac{\lambda_m d_2}{\sqrt{V_2}} \right)}
\end{align*}
\]

where \( \lambda_m \) is obtained by solving the dimensional form of (6.39), which is shown below.

\[
\left( 1 - \frac{\mu_2 \sqrt{V_1}}{\mu_1 \sqrt{V_2}} \right) \cos \left( \frac{\lambda_m}{\sqrt{V_1}} \left[ \frac{d_1 \sqrt{V_2}}{d_2 \sqrt{V_1}} - 1 \right] \right) = \left( 1 + \frac{\mu_2 \sqrt{V_1}}{\mu_1 \sqrt{V_2}} \right) \cos \left( \frac{\lambda_m}{\sqrt{V_1}} \left[ \frac{d_1 \sqrt{V_2}}{d_2 \sqrt{V_1}} + 1 \right] \right)
\]

6.2 Linear Stability

Since the experiment is conducted at a constant deceleration rate, the system resembles a pseudo-steady state function which is independent of time. The pseudo-steady state velocity profile is described above.

Assuming a small perturbation to the basic flow, the non-dimensionalized velocities and pressures can be separated into the basic flow and the perturbation (continuing the non-dimensionalized variables from Section 6.1.1 along with \( \chi_j = v_j / \sqrt{kd_2} \), \( \sigma_j = p_j / \rho_2 kd_2 \), and \( \gamma = x / d_2 \)).

\[
\kappa_j = K_j + \hat{k}_j, \quad \chi_j = \hat{\chi}_j, \quad \sigma_j = \Sigma_j + \hat{\sigma}_j
\]

Substituting these new equations into the NS equations, keeping only the perturbation terms (the base case was solved above in Section 6.1), and neglecting second order terms because of the linear
analysis, we get the following for fluid 1:

\[
\frac{1}{Re} \frac{\partial \hat{k}_1}{\partial \epsilon} + K_1 \frac{\partial \hat{k}_1}{\partial \gamma} + \frac{\partial K_1}{\partial \eta} = -\frac{\beta}{\zeta} \frac{\partial \hat{\sigma}_1}{\partial \gamma} + \frac{\beta}{Re} \Delta \hat{k}_1, \quad (6.50)
\]

\[
\frac{1}{Re} \frac{\partial \hat{\chi}_1}{\partial \epsilon} + K_1 \frac{\partial \hat{\chi}_1}{\partial \gamma} = -\frac{\beta}{\zeta} \frac{\partial \hat{\sigma}_1}{\partial \eta} + \frac{\beta}{Re} \Delta \hat{\chi}_1, \quad (6.51)
\]

\[
\frac{\partial \hat{k}_1}{\partial \gamma} + \frac{\partial \hat{\chi}_1}{\partial \eta} = 0, \quad (6.52)
\]

and the following for fluid 2:

\[
\frac{1}{Re} \frac{\partial \hat{k}_2}{\partial \epsilon} + K_2 \frac{\partial \hat{k}_2}{\partial \gamma} + \hat{\chi}_2 \frac{\partial K_2}{\partial \eta} = -\frac{\partial \hat{\sigma}_2}{\partial \gamma} + \frac{1}{Re} \Delta \hat{k}_2, \quad (6.53)
\]

\[
\frac{1}{Re} \frac{\partial \hat{\chi}_2}{\partial \epsilon} + K_2 \frac{\partial \hat{\chi}_2}{\partial \gamma} = -\frac{\partial \hat{\sigma}_2}{\partial \eta} + \frac{1}{Re} \Delta \hat{\chi}_2, \quad (6.54)
\]

\[
\frac{\partial \hat{k}_2}{\partial \gamma} + \frac{\partial \hat{\chi}_2}{\partial \eta} = 0. \quad (6.55)
\]

The equation of continuity, (6.52) and (6.55), permits the use of the stream function.

\[
\hat{k}_j = \frac{\partial \psi_j}{\partial \eta}, \quad \hat{\chi}_j = -\frac{\partial \psi_j}{\partial \gamma}
\]

We shall now assume that the disturbance can be described as a sinusoidal disturbance and write

\[
\psi_j = \phi_j(\eta) \exp[i\alpha(\gamma - c\epsilon)], \quad \hat{\sigma}_j = f_j(\eta) \exp[i\alpha(\gamma - c\epsilon)]
\]

in which \( c \) is the dimensionless wave velocity and can be split into its real and imaginary parts, \( c = c_r + ic_i \), and \( \alpha \) is the dimensionless wave number defined by \( 2\pi d_2/\lambda \) where \( \lambda \) is the wave length.

Substituting the sinusoidal form in the disturbance NS equations (6.50) and (6.51) for fluid 1, we get

\[
i\alpha \left( K_1 - \frac{c}{Re} \right) \phi'_1 - i\alpha K'_1 \phi_1 = \frac{-i\alpha\beta}{\zeta} f_1 + \frac{\beta}{Re} \left( \phi'''_1 - \alpha^2 \phi'_1 \right) \quad (6.56)
\]

\[
\alpha^2 \left( \frac{c}{Re} - K_1 \right) \phi_1 = \frac{\beta}{\zeta} f'_1 + \frac{i\alpha\beta}{Re} \left( \phi''_1 - \alpha^2 \phi_1 \right) \quad (6.57)
\]
in which the primes on $\phi$, $K$ and $f$ denote differentiation with respect to $y$. Eliminating $f_1$ from the previous two equations, (6.56) and (6.57), through cross-differentiation, produces an Orr-Sommerfeld equation [414].

$$\beta \left( \phi_1^{'''} - 2\alpha^2 \phi_1'' + \alpha^4 \phi_1 \right) = i \alpha Re \left[ \left( \frac{c}{Re} - K_1 \right) \left( \alpha^2 \phi_1 - \phi_1'' \right) - K_1'' \phi_1 \right]$$

(6.58)

Substituting the sinusoidal form in the disturbance NS equations (6.50) and (6.51) for fluid 2, we get

$$i \alpha \left( K_2 - \frac{c}{Re} \right) \phi_2' - i \alpha K_2' \phi_2 = -i \alpha f_2 + \frac{1}{Re} \left( \phi_2''' - \alpha^2 \phi_2' \right)$$

(6.59)

$$\alpha^2 \left( \frac{c}{Re} - K_2 \right) \phi_2 = f_2' + \frac{i \alpha}{Re} \left( \phi_2'' - \alpha^2 \phi_2 \right).$$

(6.60)

Eliminating $f_2$ from the previous two equations, (6.59) and (6.60), through cross-differentiation, produces an Orr-Sommerfeld equation.

$$\phi_2^{'''} - 2\alpha^2 \phi_2'' + \alpha^4 \phi_2 = i \alpha Re \left[ \left( \frac{c}{Re} - K_2 \right) \left( \alpha^2 \phi_2 - \phi_2'' \right) - K_2'' \phi_2 \right]$$

(6.61)

Now that we have developed the flow equations for the perturbations, we need to address the boundary conditions. There are eight boundary conditions for $\hat{\kappa}_j$ and $\hat{\chi}_j$; four for each of the two fluids. Fluid 1 has a boundary condition for both $\hat{\kappa}_1$ and $\hat{\chi}_1$ at the air-liquid interface and the liquid-liquid interface. Fluid 2 has a boundary condition for both $\hat{\kappa}_2$ and $\hat{\chi}_2$ at the wall and the liquid-liquid interface.

The boundary conditions for the system are as follows. At the wall ($\eta = -1$), we have

$$\hat{\kappa}_2 = 0,$$

(6.62)

$$\hat{\chi}_2 = 0.$$
At the liquid-liquid interface \((\eta = 0)\), we have

\[
\hat{\kappa}_1 + m \frac{\partial K_1}{\partial \eta} = \hat{\kappa}_2 + m \frac{\partial K_2}{\partial \eta}, \tag{6.64}
\]

\[
\hat{\chi}_1 = \hat{\chi}_2, \tag{6.65}
\]

\[
\mu_1 \left( \frac{\partial \hat{\kappa}_1}{\partial \eta} + \frac{\partial \hat{\chi}_1}{\partial \gamma} + m \frac{\partial^2 K_1}{\partial \eta^2} \right) = \mu_2 \left( \frac{\partial \hat{\kappa}_2}{\partial \eta} + \frac{\partial \hat{\chi}_2}{\partial \gamma} + m \frac{\partial^2 K_2}{\partial \eta^2} \right), \tag{6.66}
\]

\[
\tilde{\sigma}_1 - \frac{2\zeta}{Re} \frac{\partial \hat{\chi}_1}{\partial \eta} + \left( 1 - \frac{\zeta}{\beta} \right) mF = \tilde{\sigma}_2 - \frac{2}{Re} \frac{\partial \hat{\chi}_2}{\partial \eta} + S_2^2 \frac{\partial^2 m}{\partial \gamma^2}, \tag{6.67}
\]

where \(m\) represents the deviation of the liquid-liquid interface from \(\eta = 0\), \(S_2\) represents the non-dimensional surface tension between the two fluids and \(F = g_c/k\). But for our case, we do not have surface tension between the red blood cell and plasma layers so \(S_2 = 0\). At the air-liquid interface \((\eta = \xi)\), we have

\[
\mu_1 \left( \frac{\partial \hat{\kappa}_1}{\partial \eta} + \frac{\partial \hat{\chi}_1}{\partial \gamma} + r \frac{\partial^2 K_1}{\partial \eta^2} \right) = 0, \tag{6.68}
\]

\[
\frac{\zeta}{\beta} rF + \frac{2\zeta}{Re} \frac{\partial \hat{\chi}_1}{\partial \eta} - \hat{\sigma}_1 = S_1 \frac{\partial^2 r}{\partial \gamma^2}, \tag{6.69}
\]

where \(r\) represents the deviation of the air-liquid interface from \(\eta = \xi\), \(S_1 = T_1/\rho_2 kd_2^2\), \(T_1\) represents the surface tension between the air and fluid 1. In our case, we do have surface tension but it is dependent on the amount of proteins in the plasma layer. Therefore, we will test the surface tension at the two extremes and the average value of physiologically relevant protein levels to determine how the change in surface tension affects the stability.

Now that the boundary conditions are established, we will convert them into the equations representing the stream function and the pressure. We then obtain the following boundary conditions to be solved in conjunction with the two flow equations. At the wall, we now have

\[
\phi_2(-1) = 0 \tag{6.70}
\]

\[
\phi_2'(-1) = 0. \tag{6.71}
\]

In order to be able to determine the boundary conditions at the liquid-liquid interface and air-liquid interface, we need to first look at the kinematic conditions at both interfaces. The linearized
kinematic conditions for each interface are

\[
\frac{\partial r}{\partial \varepsilon} + K_1(\xi) \frac{\partial r}{\partial \gamma} = \dot{\xi}(\xi) = -i\alpha \phi_1(\xi) \exp[i\alpha (\gamma - c\varepsilon)] \tag{6.72}
\]

\[
\frac{\partial m}{\partial \varepsilon} + K_1(0) \frac{\partial m}{\partial \gamma} = \dot{\chi}_2(0) = \dot{\chi}_1(0) = -i\alpha \phi_1(0) \exp[i\alpha (\gamma - c\varepsilon)] . \tag{6.73}
\]

Solving (6.72) for \( r \) and (6.73) for \( m \), we obtain

\[
r = \frac{\phi_1(\xi)}{c \overline{K} - K_1(\xi)} \exp[i\alpha (\gamma - c\varepsilon)] \tag{6.74}
\]

\[
m = \frac{\phi_1(0)}{c \overline{K} - K_1(0)} \exp[i\alpha (\gamma - c\varepsilon)] . \tag{6.75}
\]

The boundary conditions for the interface are then

\[
\phi_1(0) = \phi_2(0) , \tag{6.76}
\]

\[
\phi_1'(0) + \frac{\phi_1(0)}{c \overline{K} - K_1(0)} K_1'(0) = \phi_2'(0) + \frac{\phi_1(0)}{c \overline{K} - K_1(0)} K_2'(0) , \tag{6.77}
\]

\[
\zeta \left[ \phi_1''(0) + \alpha^2 \phi_1(0) + \frac{\phi_1(0)}{c \overline{K} - K_1(0)} K_1''(0) \right] = \phi_2''(0) + \alpha^2 \phi_2(0) + \frac{\phi_1(0)}{c \overline{K} - K_1(0)} K_2''(0) , \tag{6.78}
\]

\[
\frac{\alpha \Re}{c \overline{K} - K_1(0)} \left( 1 - \frac{\zeta}{\beta} \right) F + \zeta \phi_1''(0) + \phi_1'(0) \left( i\alpha c \frac{\zeta}{\beta} - 3\alpha^2 \zeta - i\alpha \Re K_1(0) \frac{\zeta}{\beta} \right) = \phi_2'(0) (i\alpha c - 3\alpha^2 - i\alpha \Re K_2(0)) + \phi_2''(0) + i\alpha \Re \left( \frac{\phi_2(0) K_2'(0) - \zeta}{\beta} \phi_1(0) K_1'(0) \right) . \tag{6.79}
\]

At the air interface, we have

\[
\phi_1''(\xi) + \alpha^2 \phi_1(\xi) + \frac{\phi_1(\xi)}{c \overline{K} - K_1(\xi)} K_1''(\xi) = 0 , \tag{6.80}
\]

\[
3\alpha^2 \zeta \phi_1'(\xi) - \zeta \phi_1''(\xi) - \frac{i\alpha \Re}{\beta} \left( \frac{c}{\overline{K}} - K_1(\xi) \right) \phi_1'(\xi) - \frac{i\alpha \Re}{\beta} \phi_1(\xi) K_1'(\xi) + \frac{i\alpha \Re}{\beta} \left( \alpha^2 S_1 + \zeta F \right) = 0 . \tag{6.81}
\]

For running the problem using a numerical solver, we need to change both fluids be from

\[
0 \leq \eta \leq 1 . \tag{6.79}
\]

In order to do this we need to re-evaluate all equations involving fluid 1 with \( \eta = \xi \hat{\eta} \).
and fluid 2 with $\eta = -\hat{\eta}$. After adjusting $\eta$ for the respective equations, the new equations are:

$$\frac{\phi_1'''}{\xi^4} - \frac{2\alpha^2}{\xi^2} \phi_1'' + \alpha^4 \phi_1 = \frac{i\alpha Re}{\beta} \left[ \left( \frac{c}{Re} - K_1 \right) \left( \alpha^2 \phi_1 - \phi_1'' \right) - \frac{K_1''}{\xi^2} \phi_1 \right],$$  \hspace{1cm} (6.82)

$$\frac{\phi_2'''}{\xi^4} - \frac{2\alpha^2}{\xi^2} \phi_2'' + \alpha^4 \phi_2 = i\alpha Re \left[ \left( \frac{c}{Re} - K_2 \right) \left( \alpha^2 \phi_1 - \phi_2'' \right) - K_2' \phi_2 \right],$$  \hspace{1cm} (6.83)

$$\phi_2(1) = 0,$$  \hspace{1cm} (6.84)

$$\phi_2'(1) = 0,$$  \hspace{1cm} (6.85)

$$\phi_1(0) = \phi_2(0),$$  \hspace{1cm} (6.86)

$$\frac{\phi_1''(0)}{\xi} + \frac{\phi_1(0)}{\xi \left( \frac{c}{Re} - K_1(0) \right) K_1'(0)} = -\phi_2'(0) - \frac{\phi_1(0)}{\xi \left( \frac{c}{Re} - K_1(0) \right)} K_2'(0),$$  \hspace{1cm} (6.87)

$$\zeta \left( \frac{\phi_2''(0)}{\xi^2} + \alpha^2 \phi_1(0) + \frac{\phi_1(0)}{\xi \left( \frac{c}{Re} - K_1(0) \right) K_1'(0)} \right) = \phi_2''(0) + \alpha^2 \phi_2(0) + \frac{\phi_1(0)}{\xi \left( \frac{c}{Re} - K_1(0) \right)} K_2''(0),$$  \hspace{1cm} (6.88)

$$i\alpha Re \frac{\phi_1(0)}{\xi \left( \frac{c}{Re} - K_1(0) \right)} \left( 1 - \frac{c}{\beta} \right) F + \frac{\xi \phi_1''(0)}{\xi^3} + \phi_1(0) \left( \frac{i\alpha c}{\beta} - 3\alpha^2 \zeta - i\alpha Re K_1(0) \frac{\zeta}{\beta} \right) =$$  \hspace{1cm} (6.89)

$$\phi_2'(0) \left( 3\alpha^2 - i\alpha c + i\alpha Re K_1(0) \right) - \phi_2''(0) - i\alpha Re \left( \phi_2(0) K_2'(0) + \frac{\zeta}{\beta \xi} \phi_1(0) K_1'(0) \right),$$

$$\frac{\phi_1''(1)}{\xi^2} + \alpha^2 \phi_1(1) + \frac{\phi_1(1)}{\xi \left( \frac{c}{Re} - K_1(1) \right)} K_1''(1) = 0,$$  \hspace{1cm} (6.90)

$$\frac{3\alpha^2 \zeta}{\xi} \phi_1(1) - \frac{\zeta}{\xi^3} \phi_1''(1) - \frac{i\alpha Re \zeta}{\xi} \left( \frac{c}{Re} - K_1(1) \right) \phi_1(1)$$

$$- \frac{i\alpha Re \zeta}{\xi \beta} \phi_1(1) K_1'(1) + \frac{i\alpha Re \phi_1(1)}{\xi \left( \frac{c}{Re} - K_1(1) \right)} \left( \alpha^2 S_1 + \frac{\zeta}{\beta} F \right) = 0.$$  \hspace{1cm} (6.91)

The problem is then defined as an eigenvalue problem for $c$. The Orr-Sommerfeld equations for fluid 1 and fluid 2, (6.82) and (6.83), together with the eight boundary conditions, (6.84) - (6.91), governs the stability of the problem. $c$ can be split into its real $c_r$ and its imaginary $c_i$ parts. Both of these are dependent on $\zeta$, $\xi$, $\beta$, $Re$, $S_1$, and $F$. The stability of the flow is then determined by the value of $c_i$, with $c_i = 0$ being the neutral stability curve. The fluid is then either stable if $c_i$ is negative or unstable if $c_i$ is positive.
The further complication is that the eigenvalue problem for $c$ must be linear in the eigenvalue. Hence, we must rewrite (6.89) as:

$$
\left( \frac{\zeta \phi'''(0)}{\xi^3} - \frac{3\alpha^2 \zeta \phi'(0)}{\xi} \right) \left( \frac{c}{\text{Re}} - K_1(0) \right) + \frac{i\alpha \text{Re} \zeta \phi'_1(0)}{\xi \beta} \left( \frac{c^2}{\text{Re}^2} - \frac{2cK_1(0)}{\text{Re}} + K_1(0)^2 \right) \\
+ i\alpha \text{Re} \phi_1(0) \left( 1 - \frac{\zeta}{\beta} \right) F + i\alpha \text{Re} \left( \phi_2(0)K_2'(0) + \frac{\zeta}{\beta \xi} \phi_1(0)K_1'(0) \right) \left( \frac{c}{\text{Re}} - K_1(0) \right) = \\
(3\alpha^2 \phi_2'(0) - \phi_2'''(0)) \left( \frac{c}{\text{Re}} - K_1(0) \right) - i\alpha \text{Re} \phi_2'(0) \left( \frac{c^2}{\text{Re}^2} - \frac{2cK_1(0)}{\text{Re}} + K_1(0)^2 \right). \quad (6.92)
$$

Similarly we rewrite (6.91) as:

$$
\left( \frac{3\alpha^2 \zeta \phi'(1)}{\xi} - \frac{\zeta \phi'''(1)}{\xi^3} \right) \left( \frac{c}{\text{Re}} - K_1(1) \right) - \frac{i\alpha \text{Re} \zeta \phi'_1(1)}{\xi \beta} \left( \frac{c^2}{\text{Re}^2} - \frac{2cK_1(1)}{\text{Re}} + K_1(1)^2 \right) \\
- \frac{i\alpha \text{Re} \zeta \phi_1(1)K_1'(1)}{\xi \beta} \left( \frac{c}{\text{Re}} - K_1(1) \right) + i\alpha \text{Re} \phi_1(1) \left( \alpha^2 S_1 + \frac{\zeta}{\beta} F \right) = 0. \quad (6.93)
$$

Now in order to make these boundary conditions linear in the eigenvalue $c$ (a requirement for Dedalus, the numerical solver package), we need to make a change of variables that increases the size of the system, but does not alter the actual state. To this end, let $\delta_j(y) = c \phi_j'(y)$. Incorporating this change into (6.92) and (6.93) yields

$$
\left( \frac{\zeta \phi'''(0)}{\xi^3} - \frac{3\alpha^2 \zeta \phi'(0)}{\xi} \right) \left( \frac{c}{\text{Re}} - K_1(0) \right) + \frac{i\alpha \text{Re} \zeta \phi'_1(0)}{\xi \beta} \left( \frac{c^2}{\text{Re}^2} - \frac{2c \delta_1(0)K_1(0)}{\text{Re}} + \phi_1'(0)K_1(0)^2 \right) \\
+ i\alpha \text{Re} \phi_1(0) \left( 1 - \frac{\zeta}{\beta} \right) F + i\alpha \text{Re} \left( \phi_2(0)K_2'(0) + \frac{\zeta}{\beta \xi} \phi_1(0)K_1'(0) \right) \left( \frac{c}{\text{Re}} - K_1(0) \right) = \\
(3\alpha^2 \phi_2'(0) - \phi_2'''(0)) \left( \frac{c}{\text{Re}} - K_1(0) \right) - i\alpha \text{Re} \left( \frac{c \delta_2(0)}{\text{Re}^2} - \frac{2c \phi_2'(0)K_1(0)}{\text{Re}} + \phi_2'(0)K_1(0)^2 \right). \quad (6.94)
$$

and

$$
\left( \frac{3\alpha^2 \zeta \phi'(1)}{\xi} - \frac{\zeta \phi'''(1)}{\xi^3} \right) \left( \frac{c}{\text{Re}} - K_1(1) \right) - \frac{i\alpha \text{Re} \zeta \phi'_1(1)}{\xi \beta} \left( \frac{c^2}{\text{Re}^2} - \frac{2c \delta_1(1)K_1(1)}{\text{Re}} + \phi_1'(1)K_1(1)^2 \right) \\
- \frac{i\alpha \text{Re} \zeta \phi_1(1)K_1'(1)}{\xi \beta} \left( \frac{c}{\text{Re}} - K_1(1) \right) + i\alpha \text{Re} \phi_1(1) \left( \alpha^2 S_1 + \frac{\zeta}{\beta} F \right) = 0. \quad (6.95)
$$

Wherever necessary, the boundary conditions for $\zeta_j(y)$ can be imitated from that for $\phi_j'(y)$. 

114
Using Dedalus to solve the equations resulted in a positive $c_i$ for every wavelength tested (0.1 $\mu$m through 1 m) for all deceleration rates tested (0.628 m/s$^2$ down to 0.006 m/s$^2$). This means that according to the linear stability analysis the system is always unstable. We postulate that a linear stability analysis does not adequately model the instabilities present in the system either because a linear analysis cannot adequately represent the non-linear instabilities present or the time-independence of the linear stability analysis is incompatible with the time-dependent velocity equations, even with a pseudo-steady state velocity profile; thus, a non-linear stability analysis is needed.

However, a non-linear stability analysis was not performed because of the complexity of the problem, limited amount of time to devote to the analysis, and because the experimental analysis of the instabilities yielded a working solution. A free-surface boundary condition requires significant attention in a non-linear stability analysis, and since a system with a moving boundary and a free surface has not been previously studied, this would require devoting many hours and much effort to a small portion of the overall dissertation and was therefore not performed.

6.3 Disk Modeling

6.3.1 Cell-pack layer

As mentioned above, the Casson equation, Eq. 6.96, is used for modeling the cell-pack layer viscosity [415]. This equation is fit to high hematocrit data from Chien et al. [416] in order to obtain the critical yield stress and Casson viscosity values as a function of the hematocrit, $Hct$. An initial shear rate is determined by solving the model as two Newtonian layers, and then the model is numerically interpolated using the Casson equation to obtain parameters for the cell-pack layer.

\[
\left( \tau^\frac{1}{2} - \tau_C^\frac{1}{2} \right)^2 = \mu_C \dot{\gamma} ,
\]

(6.96)

where $\dot{\gamma}$ is the shear rate, $\tau$ is the shear stress, $\tau_C$ is the critical yield stress with $\tau_C = f (Hct)$, and $\mu_C$ is the Casson viscosity with $\mu_C = f (Hct)$. The hematocrit value for the cell-pack layer was experimentally determined to be between 85 – 90%. The hematocrit value used in the subsequent calculations is 88%, for which $\tau_C = 47.6mPa$ and $\mu_C = 20.7mPa\cdot s$. 

115
The Casson equation \((\sqrt{\tau} - \sqrt{\mu C} = \sqrt{\mu_C \dot{\gamma}})\) was fit to the data from Chien et al [416]. The data were presented in graphical form as apparent viscosity vs hematocrit with separate data sets for the different shear rates. The apparent viscosity values were multiplied by their respective shear rate to obtain the shear stress values. The square roots of the shear stress values and the shear rates were calculated to obtain \(\tau_C\) and \(\mu_C\). Using the square roots of \(\tau\) and \(\dot{\gamma}\), the Casson equation takes the form of the linear equation: \(y = mx + b\). Using the least squares approach, \(\tau_C\) and \(\mu_C\) were calculated for each hematocrit across the range of shear rates. The resulting values for \(\tau_C\) and \(\mu_C\) were then fit to different linear equations to obtain a single \(\tau_C\) and \(\mu_C\) that were functions of hematocrit. Table 6.1 along with Figures 6.2 and 6.3 show the results of fitting the data to the equations below. The following equations were tested in order to determine the best \(\tau_C\) and \(\mu_C\) (in the equations \(y\) stands for either \(\tau_C\) or \(\mu_C\) and \(x\) stands for the hematocrit, between 0-1):

\[
\begin{align*}
y &= ax + b \quad (6.97) \\
y &= a \ln x + b \quad (6.98) \\
y &= a \ln (1 - x) + b \quad (6.99) \\
y &= a (\ln x + x) + b \quad (6.100) \\
y &= a (\ln (1-x) + x) + b \quad (6.101) \\
\ln y &= ax + b \quad (6.102) \\
\ln y &= a \ln x + b \quad (6.103) \\
\ln y &= a \ln (1 - x) + b \quad (6.104) \\
y &= ax^c + b \quad (6.105) \\
y &= a(1-x)^c + b \quad (6.106)
\end{align*}
\]
Table 6.1: Fitted parameters for $\tau_C$ and $\mu_C$.

<table>
<thead>
<tr>
<th>Equation</th>
<th>$\mu_C$</th>
<th>$\mu_C$</th>
<th>$\mu_C$</th>
<th>L1 Norm</th>
<th>$\tau_C$</th>
<th>$\tau_C$</th>
<th>$\tau_C$</th>
<th>L1 Norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equation (6.97)</td>
<td>51.68</td>
<td>-15.57</td>
<td>254</td>
<td></td>
<td>56.25</td>
<td>-14.79</td>
<td>163</td>
<td></td>
</tr>
<tr>
<td>Equation (6.98)</td>
<td>14.69</td>
<td>25.08</td>
<td>290</td>
<td></td>
<td>17.67</td>
<td>30.68</td>
<td>275</td>
<td></td>
</tr>
<tr>
<td>Equation (6.99)</td>
<td>-21.02</td>
<td>-10.62</td>
<td>183</td>
<td></td>
<td>-16.41</td>
<td>-1.74</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td>Equation (6.100)</td>
<td>11.96</td>
<td>16.16</td>
<td>281</td>
<td></td>
<td>13.92</td>
<td>19.88</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td>Equation (6.101)</td>
<td>-29.78</td>
<td>-3.79</td>
<td>129</td>
<td></td>
<td>-20.85</td>
<td>5.05</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>Equation (6.102)</td>
<td>3.87</td>
<td>-0.31</td>
<td>119</td>
<td></td>
<td>9.36</td>
<td>-3.95</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>Equation (6.103)</td>
<td>1.27</td>
<td>2.86</td>
<td>128</td>
<td></td>
<td>3.96</td>
<td>4.37</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>Equation (6.104)</td>
<td>-1.18</td>
<td>0.53</td>
<td>59</td>
<td></td>
<td>-2.11</td>
<td>-1.04</td>
<td>257</td>
<td></td>
</tr>
<tr>
<td>Equation (6.105)</td>
<td>129.57</td>
<td>4.41</td>
<td>30.09</td>
<td>84</td>
<td>54.64</td>
<td>-1.45</td>
<td>5.14</td>
<td>76</td>
</tr>
<tr>
<td>Equation (6.106)</td>
<td>3.17</td>
<td>-3.96</td>
<td>-0.78</td>
<td>66</td>
<td>-103.88</td>
<td>89.42</td>
<td>0.23</td>
<td>106</td>
</tr>
</tbody>
</table>

*The smaller the L1 Norm value, the better the fit.

Figure 6.2: Plot of the data vs the fitted equations for $\mu_C$. Experimental data, ⬤ Eq. (6.97), ⬤ Eq. (6.98), ⬤ Eq. (6.99), ⬤ Eq. (6.100), ⬤ Eq. (6.101), ⬤ Eq. (6.102), ⬤ Eq. (6.103), ⬤ Eq. (6.104), ⬤ Eq. (6.105), ⬤ Eq. (6.106)
Figure 6.3: Plot of the data vs the fitted equations for $\tau_C$. ▲ Experimental data, - - - Eq. (6.97), ····· Eq. (6.98), ····· Eq. (6.99), ····· Eq. (6.100), - - - Eq. (6.101), - - - Eq. (6.102), - - - Eq. (6.103), - - - Eq. (6.104), - - - Eq. (6.105), - - - Eq. (6.106)

From the data, the logarithmic equation (6.104) was the best fit line for $\mu_c$ with hematocrit dependence, and the logarithmic equation (6.103) was the best fit line for $\tau_C$ with hematocrit dependence. Plugging in the values, we obtain:

$$\ln \mu_c = -1.18 \ln (1 - Hct) + 0.53$$  \hspace{1cm} (6.107)

$$\ln \tau_C = 3.96 \ln Hct + 4.37$$  \hspace{1cm} (6.108)

The Casson equation for the cell-pack layer with hematocrit dependence becomes:

$$\sqrt{\tau} - \sqrt{e^{3.962 \ln Hct + 4.369}} = \sqrt{e^{-1.18 \ln (1 - Hct) + 0.529} \dot{\gamma}}.$$  \hspace{1cm} (6.109)
6.3.2 Theoretical Velocity Profiles

Table 6.2 provides the parameters used in plotting the calculated velocity profiles of Figure 6.4. The total volume of fluid is 8.5 mL, giving a total thickness of 2.97 mm. The blood and cell-pack layer had hematocrits of 45% and 88%, respectively. The viscosity of blood is dependent on both hematocrit and shear rate, and values were taken from the literature [410]. The wall shear rates range from about 0.71 s\(^{-1}\) at a deceleration rate of 0.063 m/s\(^2\) to about 55.61 s\(^{-1}\) at a deceleration rate of 0.628 m/s\(^2\), giving corresponding viscosities at the wall of about 161 mPa·s and 30 mPa·s, respectively, and giving corresponding viscosities at the interface of about 255 mPa·s and 34 mPa·s, respectively.

Table 6.2: Blood parameters used for the determining the calculated velocity profiles.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Cell-pack layer</th>
<th>Plasma layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density (g/cm(^3))</td>
<td>1.108</td>
<td>1.024</td>
</tr>
<tr>
<td>Viscosity (mPa·s)**</td>
<td>35***</td>
<td>1.2</td>
</tr>
<tr>
<td>Thickness (mm)</td>
<td>1.519</td>
<td>1.451</td>
</tr>
</tbody>
</table>

* Parameters came from [417]
** Parameters came from [416, 418]
*** 35 mPa·s is the initial viscosity used for the cell-pack layer, and then the Casson equation is used as shear develops.

Figure 6.4A shows the development of the calculated velocity profile once the deceleration of the disk commences for a deceleration rate of 0.063 m/s\(^2\). The amount of time that it takes for the velocity profile to fully develop is dependent on the rate at which the disk is decelerating. For a deceleration rate of 0.063 m/s\(^2\), the pseudo-steady state velocity profile develops in \(\approx\)11 seconds. For a deceleration rate of 0.628 m/s\(^2\), the pseudo-steady state velocity profile develops in \(\approx\)13.5 seconds. Once the pseudo-steady state velocity profile is developed, the shape of the profile remains constant until an instability appears or the wall of the disk attains zero velocity, after which a transient deceleration of the fluid occurs, which we have not modeled. Figure 6.4B shows the difference between treating the blood flow as a Newtonian fluid with a viscosity of 35 mPa·s and
treat the blood as a non-Newtonian fluid with a Casson viscosity model at a deceleration rate of 0.063 m/s^2.

![Image](image_url)

Figure 6.4: A) A representative development of the velocity profile as the disk begins to decelerate (constant deceleration of 0.063 m/s^2 with a hematocrit of 45% and using the Casson equation shown). The profile develops faster for slower decelerations. –– $t = 0$ seconds, –– $t = 0.4$ seconds, –– $t = 0.5$ seconds, –– $t = 1$ seconds, –– $t = 3$ seconds, –– $t = 5$ seconds. B) The Newtonian vs the non-Newtonian fluid at a deceleration rate of 0.063 m/s^2. –– Newtonian, –– non-Newtonian

### 6.3.3 Instability Growth

The shape of the velocity profile for each deceleration rate is self-similar and scaled by the velocity difference between the wall and the free surface. In analyzing the shape of the profile, there is a sharp discontinuity in the shear rate at the interface between the two layers. This discontinuity increases as the deceleration rate increases in magnitude. This abrupt change in the shear rate may be a source of initiation or growth for the instabilities. For a deceleration rate of 0.063 m/s^2, the observed instabilities occur when the disk speed is about 1.3 m/s (see Figure 6.5B) and the disk has decelerated for $\approx 280$ seconds. At this deceleration rate, the pseudo-steady state velocity profile is developed in $\approx 11$ seconds, but the instabilities do not occur until $\approx 280$ seconds. This is long after the velocity profile has reached the pseudo-steady state, but the instability may be a combination of both the size of the shear rate discontinuity and the centripetal force felt by the fluid.
A model to describe the instability at the interface is the Kelvin-Helmholtz instability model. The Kelvin-Helmholtz model employs a Richardson number (Ri), which is a ratio of the stratification of the fluid to the shear rate of the fluid squared. If the Ri number meets the Ri criteria, \( Ri < 0.25 \), then the fluid has met a necessary but not sufficient condition for instability. The Ri formula is:

\[
Ri = \frac{-\frac{g_c}{\rho_0} \left( \frac{\partial \rho}{\partial y} \right)}{\left( \frac{\partial u}{\partial y} \right)^2},
\]  

(6.110)

where \( \rho_0 \) is the average density of the system. For our system, \( \rho_0 = 1061 \text{ kg/m}^3 \) which gives a value of 26.66 m\(^{-1} \) for \( -\left( \frac{\partial \rho}{\partial y} \right) / \rho_0 \). Since \( g_c = \Omega^2 R \) and \( u_c = \Omega R \), we can relate the disk velocity to the shear rate and determine when the system meets the Ri criteria at the interface. Figure 6.5C shows the velocity versus shear rate plot. This plot shows that the shear rate is transient for the first few seconds and then reaches a steady state value. Only in one case does the Ri criteria happen while the interface is in the transient state. Figure 6.5B shows the relation of the Ri criteria to the experimental data on a velocity versus log time plot. This plot shows how quickly each deceleration rate reached zero along with how far apart the Ri criteria and the experimental instability (mixing) values are.

We can also analyze the instability for a growth rate. Because we believe the instability to be shear driven, we will propose that the growth rate (\( z \)) scales as interfacial friction velocity divided by the cell-pack layer thickness:

\[
z = \frac{\sqrt{\tau}}{d_2}. \]  

(6.111)

This equation shows us that the faster deceleration rates (higher shear stress) grow at a faster rate, which is consistent with Figure 6.5A. To validate this growth rate model, we analyzed both the theoretical velocity profile and the experimental data. For the experimental data, we do not know the interfacial shear stress. Therefore, in order to fit this instability growth model to the experimental data, we used growth ratios which compare each constant deceleration growth rate to the 0.628 m/s\(^2 \) constant deceleration growth rate. The amount of time in which the interface is unstable ("time mixed") is calculated by subtracting the time it took for the two layers to mix from the total time it took for the disk velocity to reach zero. Then the amount of "time mixed" is divided
Figure 6.5: A) Comparison of the rpm at the onset of instability for the baffled and non-baffled disk for both genders. The closer the data points are to the vertical axis the greater the stable region of the disk. ○ Baffle Male, □ Baffle Female, ▲ No Baffle Male, ◆ No Baffle Female B) Semi-log plot of velocity vs time for the constant deceleration rates: 0.031 m/s², 0.063 m/s², 0.126 m/s², 0.188 m/s², 0.314 m/s², 0.503 m/s², 0.628 m/s². The plot also contains the Ri criteria line (—) and the experimental mixing values (▲). C) Plot of the Richardson criteria (—) with the theoretical velocity profiles and experimental data on a velocity vs shear rate plot. 0.031 m/s², 0.063 m/s², 0.126 m/s², 0.188 m/s², 0.314 m/s², 0.503 m/s², 0.628 m/s² D) Plot of the instability growth rate comparisons for the theoretical velocity profile, ▲ experimental data, and × Kelvin-Helmholtz instability model.

by the total time it took for the disk velocity to reach zero to calculate the percentage of time that the two layers are mixed. If the growth model is correct then the ratio of each constant deceleration rates’ percentage of “time mixed” to the 0.628 m/s² constant deceleration rate’s percentage of “time mixed”, will be the same ratio as the ratio of the corresponding constant deceleration rate’s growth time to the 0.628 m/s² constant deceleration rate’s growth time. This analysis validates
the growth rate model because the instability will cause mixing sooner if it grows faster; thus the layers will have spent a larger percent of the total time mixed compared to a slower growth rate. The theoretical growth rates were calculated using the steady state interfacial shear stress. Each of these values was then compared to the value for the 0.628 m/s² constant deceleration rate to obtain similar ratios to those from the experimental data. These ratios can be found in Table 6.3 and in Figure 6.5D.

Table 6.3: Instability growth rate values for the theoretical velocity profile, the experimental data, and the Kelvin-Helmholtz model.

<table>
<thead>
<tr>
<th>Deceleration Ratio</th>
<th>Theoretical Velocity Profile</th>
<th>Experimental Data</th>
<th>Kelvin-Helmholtz Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.628 m/s²</td>
<td>4.388</td>
<td>5.335</td>
<td>4.952</td>
</tr>
<tr>
<td>0.031 m/s²</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628 m/s²</td>
<td>3.170</td>
<td>3.305</td>
<td>3.143</td>
</tr>
<tr>
<td>0.063 m/s²</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628 m/s²</td>
<td>2.238</td>
<td>2.287</td>
<td>2.645</td>
</tr>
<tr>
<td>0.126 m/s²</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628 m/s²</td>
<td>1.826</td>
<td>1.876</td>
<td>2.329</td>
</tr>
<tr>
<td>0.188 m/s²</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628 m/s²</td>
<td>1.415</td>
<td>1.330</td>
<td>1.893</td>
</tr>
<tr>
<td>0.314 m/s²</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.628 m/s²</td>
<td>1.115</td>
<td>1.116</td>
<td>1.401</td>
</tr>
</tbody>
</table>

To see if the Kelvin-Helmholtz model also fit the instability growth rate model, we took the theoretical growth rate values and divided them by the difference between the time that the Ri criteria was met for each deceleration rate and the time that mixing was observed to calculate the percentage of time that the instability was growing. Each of these values was then compared to the value for the 0.628 m/s² constant deceleration rate. These ratios can be found in Table 6.3 and in Figure 6.5D.

For each of the instability growth rate comparisons, only the deceleration rates between 0.031 m/s² and 0.628 m/s² were used because there was no mixing in the majority of the experimental trials for the 0.013 m/s² and 0.006 m/s² constant deceleration rates.
6.3.4 Onset of instabilities in a decelerating 2-fluid layer

The rate of deceleration determines the shear stress at the interface, which creates a discontinuity of the shear rate at the fluid-fluid interface. It is proposed that this discontinuity instigates the instability found in the system. There were two proposed models for representing the instability in the system: the Kelvin-Helmholtz model and the interfacial growth rate model. The growth rate model explains why the instability is seen later for the slower constant decelerations but does not give a reason as to why the instability is happening. The Kelvin-Helmholtz model was chosen to explain why the instability is happening because it takes into account three of the major factors of our system, the centrifugal force, the density discontinuity and the shear rate discontinuity.

When looking at Figure 6.5B, we see that all the experimental values of instability fall below the Richardson criteria line. The higher constant deceleration rates fall farther below the line than the lower constant deceleration rates. This highlights the fact that the Richardson criteria is necessary but not sufficient, which means that there are other factors at play in instigating the instability. But it does not mean that the model is invalid. Looking at Figure 6.5D and Table 6.3, we see that the Ri values for the Kelvin-Helmholtz model follow the same trend and are within the deviation of the experimental values from the theoretical values. At a deceleration rate of 0.031 m/s$^2$, there is $\approx$25% deviation from the theoretical growth curve. We postulate that this is due to diffusion of the momentum because the disk is spinning very slowly at that time. Another detail that points to the possibility that the instabilities are governed by the Richardson criteria is that not all of the trials demonstrated mixing. The velocity at which the Richardson criteria is met for 0.006 m/s$^2$ and 0.013 m/s$^2$ is 0.182 m/s and 0.344 m/s, respectively, and the time is 2971 s and 1471 s, respectively. This is slow enough and close enough to the end of the disk rotation that the instability may not be able to propagate fast enough to actually cause mixing unless disturbed by another force.

As noted in the literature, there are 3 primary types of instabilities: long-wavelength, short-wavelength and intermediate-wavelength. Because these experiments are performed in a rotating disk, there are limits on the allowed sizes of the instability wavelengths. In our system, the long-wavelength instabilities are limited to the diameter of the disk. The short-wavelength instabilities are limited to a percentage of the size of the RBCs ($\approx$8 $\mu$m), approximately 0.1 $\mu$m, because anything smaller than that will fall below the continuum scale of the fluid and not be able to
disrupt the cell-pack layer. Therefore the instabilities most likely to perturb the system are the intermediate-wavelength instabilities. In previous studies it has been noted that these intermediate-wavelength instabilities occur at higher Re with wavelengths the same magnitude as the thickness of the lower layer. The Re for the cell-pack layer ranges from 0.005 to 1.5, while the Re for the plasma layer ranges from 9 to 40, for deceleration rates of 0.031 m/s² and 0.628 m/s². In our hematocrit experiments, we could not show that the thickness of the lower layer affects the stability of the system. According to the experiments of Sangalli et al. [117] and Barthelet et al. [118] and the observations of Hu et al. [121], our 30% hematocrit experiments should have been more unstable than the 50% hematocrit since the 30% hematocrit experiments have the more viscous layer occupying less of the container. We did not see this result. Also, Kao noted that having a less viscous upper layer was always stable [119], which is not the case for our experiments. Our system is similar to the results that Yih obtained for semibounded flow, which state that the velocity difference is destabilizing if the narrow layer is the more viscous layer [113]. The linear stability analysis yielded a similar result that the system is always unstable. The stability analysis does not guarantee that the system will mix, but that the system is susceptible to all disturbance wavelengths. In the event of a disturbance, the system will also allow the disturbance to grow. However, how the disturbance grows and progresses is determined by a non-linear stability analysis which was not undertaken.

While the system is susceptible to disturbances, the induced centrifugal force appears to have a stabilizing effect that the viscosity difference must overcome to produce the instability. This is consistent with the Kelvin-Helmholtz model, where for a given change in density, the instability goes as $g_c/\gamma^2$. For our system we have a critical $g_c/\gamma^2$ ratio of $9.4 \times 10^{-3}$. Without a formal wave-length instability analysis, it is hard to say what sizes of wavelengths affect the system and whether or not the wavelength size that affects the system changes with either centrifugal gravity or shear rate. There is a possibility that the results of the Renardy [122] and Bresch [123] studies, which state that as gravity increased there was an increase in the wavelength at which the short-wavelength instabilities could occur, could apply to our system. Another prevailing component of instabilities in two-phase flow is the interplay between the two-layer interface and the presence of a critical layer which can contain the interface or be inside either of the two layers [112]. It is in this critical layer that the instability begins and can contribute to an instability below the critical
long-wavelength or above the critical short-wavelength. None of these developments have been examined for a two-layer flow bounded by a moving boundary and a free-surface, so while we may infer some phenomenological predictions, such discussions are not diagnostic.

6.4 Experimental Onset of Instability

Experiments were performed comparing the onset of instability associated with the baffled disk and the non-baffled disk using the 7 mL of blood with 1.5 mL of PBS. These experiments examined whether the presence of baffles in the disk were associated with a change in the disk velocity at the onset of instability. There were 46 trials run in this set of experiments. Twenty-one of the experiments were performed on the baffled disk (Disk 1) with 11 of those experiments using female blood and 10 of them using male blood. Twenty-five of the experiments were performed on the non-baffled disk (Disk 2) with 10 of those experiments using female blood and 15 of them using male blood. Male and female blood have different numbers of the cellular components (RBCs, WBCs, and platelets) and protein content [26]. This design also allowed us to determine if the instabilities were sensitive to slight changes in the difference of the blood make up between genders. Figure 6.5A shows the results of these tests. At all deceleration rates for the experiments using 7 mL blood and 1.5 mL PBS, there was no statistically significant difference in the speed of the disk at the onset of mixing between Disk 1 and Disk 2 ($p \geq 0.15$). Gender of the blood donor (and general hematocrit) did not make a difference in the speed of the disk at the onset of mixing in the blood at any deceleration rate ($p \geq 0.12$). The blood of both genders had some trials that did not mix at deceleration rates of 0.031 m/s$^2$, 0.013 m/s$^2$ and 0.006 m/s$^2$ and in both types of disk.

A second set of experiments was performed to examine the effect of the the total volume of fluid in the disk and the effect of different hematocrit levels in the blood. These experiments provided different thicknesses of the cell-pack layer ($d_2$) and plasma layer ($d_1$). The non-baffled disk was used for these experiments. The different total volumes of blood tested were: 8.5 mL of blood; 7 mL of blood mixed with 1.5 mL of PBS; and 7 mL of blood. Those experiments performed with 7 mL of blood mixed with 1.5 mL of PBS allowed for direct comparison with the 8.5 mL of blood with the same total volume in the disk but a different volume of RBCs. The experiments using 7 mL of blood mixed with 1.5 mL of PBS also allowed for direct comparison to the 7 mL of blood with the same value of $d_2$ but with a different total volume (and value of $d_1$) in
the disk. In these experiments, only three deceleration rates were tested: 0.628 m/s$^2$, 0.126 m/s$^2$, and 0.031 m/s$^2$. Both male and female blood were tested and three hematocrit values were used: 30%, 40%, and 50%. Eight trials were performed for each hematocrit at each deceleration rate for the different total volumes of fluid. Each gender was used equally in these trials.

Figure 6.6: Comparison of the rpm at the onset of instability for hematocrit values of 30% ($d_1 = 1.81 \text{ mm}$, $d_2 = 0.83 \text{ mm}$), 40% ($d_1 = 1.53 \text{ mm}$, $d_2 = 1.11 \text{ mm}$), and 50% ($d_1 = 1.25 \text{ mm}$, $d_2 = 1.39 \text{ mm}$) at deceleration rates of $\triangle 0.628 \text{ m/s}^2$, $\blacksquare 0.126 \text{ m/s}^2$, and $\bullet 0.031 \text{ m/s}^2$ for (A) 7 mL of blood, (B) 7 mL of blood mixed with 1.5 mL PBS, and (C) 8.5 mL of blood. In C, there are 6 data points at a deceleration rate of 0.031 m/s$^2$ and 30% hct in which mixing was never observed.

Figure 6.6 shows the results depicting the disk speed at the onset of mixing for the different total volumes of fluid at each hematocrit for the different rates of deceleration. The graphs show slightly offset data points for the 0.628 m/s$^2$ and 0.031 m/s$^2$ deceleration rates for better comparison of the data points. For each of the three total volumes of fluid (7 mL blood, 7 mL blood mixed with 1.5 mL PBS, and 8.5 mL blood), there is much scatter and no statistically significant difference in the disk speed at the onset of instability between the different hematocrit values (cell pack
layer thickness) except for the case of a total volume of 8.5 mL of blood at 30% hematocrit where 6 of the 8 trials never displayed mixing ($p \leq 0.023$). There is also a much larger difference in the disk speed at the onset of instability between 0.628 m/s$^2$ and 0.126 m/s$^2$ than there is between 0.126 m/s$^2$ and 0.031 m/s$^2$. The total volume of fluid had no statistically significant effect upon the critical disk speed at the onset of instability observed in these experiments ($p \leq 0.1$).

![Figure 6.7: Comparison of the mixing rpm at the onset of instability for the baffled and non-baffled disk for both genders. The closer the data points are to the vertical axis the greater the stable region of the disk.](image)

Since the hematocrit of the blood had no effect on the stability of the system, the data shown in 6.5A was fitted to obtain equations for decelerating. The fitted equations were then adjusted to ensure that the deceleration of the disk would maintain stability. Below are the equations:

\[
k = 0.6283 \\
k = 0.1677u_2(y = 0) - 0.2004 \\
k = 0.0048e^{1.686u_2(y=0)}
\]

\[
u_2(y = 0) \geq 4.94 \text{ m/s} \quad (6.112)
\]

\[
1.9 \text{ m/s} \leq u_2(y = 0) \leq 4.94 \text{ m/s} \quad (6.113)
\]

\[
u_2(y = 0) \leq 1.9 \text{ m/s} \quad (6.114)
\]
where $u_2(y = 0)$ is the wall speed and $k$ is the rate of deceleration. These lines together make up the proposed deceleration line and have a smooth transition from one line to the next since the first two lines converge to the same value at 4.94 m/s and the last two lines converge to the same value at 1.9 m/s. These equations were tested and implemented into the current motor setup for decelerating the disk. The original disk deceleration from 18.85 m/s (3000 rpm) took 282 s [26, 76] but these new deceleration equations only take 150.4 s when decelerating from 18.85 m/s. The whole process experiments (see Ch. 8) were tested using these equations for controlling the deceleration of the disk.

### 6.4.1 Onset of instabilities in a decelerating 2-fluid layer

The rate of deceleration determines the shear stress at the interface, which creates a discontinuity of the shear rate at the fluid-fluid interface. It is proposed that this discontinuity instigates the instability found in the system. There were two proposed models for representing the instability in the system: the Kelvin-Helmholtz model and the interfacial growth rate model. The growth rate model explains why the instability is seen later for the slower constant decelerations but does not give a reason as to why the instability is happening. The Kelvin-Helmholtz model was chosen to explain why the instability is happening because it takes into account three of the major factors of our system, the centrifugal force, the density discontinuity and the shear rate discontinuity.

When looking at Figure 6.5B, we see that all the experimental values of instability fall below the Richardson criteria line. The higher constant deceleration rates fall farther below the line than the lower constant deceleration rates. This highlights the fact that the Richardson criteria is necessary but not sufficient, which means that there are other factors at play in instigating the instability. But it does not mean that the model is invalid. Looking at Figure 6.5D and Table 6.3, we see that the Ri values for the Kelvin-Helmholtz model follow the same trend and are within the deviation of the experimental values from the theoretical values. At a deceleration rate of 0.031 m/s$^2$, there is $\approx 25\%$ deviation from the theoretical growth curve. We postulate that this is due to diffusion of the momentum because the disk is spinning very slowly at that time. Another detail that points to the possibility that the instabilities are governed by the Richardson criteria is that not all of the trials demonstrated mixing. The velocity at which the Richardson criteria is met for 0.006 m/s$^2$ and 0.013 m/s$^2$ is 0.182 m/s and 0.344 m/s, respectively, and the time is 2971 s and
1471 s, respectively. This is slow enough and close enough to the end of the disk rotation that the instability may not be able to propagate fast enough to actually cause mixing unless disturbed by another force.

As noted in the literature, there are 3 primary types of instabilities: long-wavelength, short-wavelength and intermediate-wavelength. Because these experiments are performed in a rotating disk, there are limits on the allowed sizes of the instability wavelengths. In our system, the long-wavelength instabilities are limited to the diameter of the disk. The short-wavelength instabilities are limited to a percentage of the size of the RBCs ($\approx 8 \mu m$), approximately 0.1 $\mu m$, because anything smaller than that will fall below the continuum scale of the fluid and not be able to disrupt the cell-pack layer. Therefore the instabilities most likely to perturb the system are the intermediate-wavelength instabilities. In previous studies it has been noted that these intermediate-wavelength instabilities occur at higher Re with wavelengths the same magnitude as the thickness of the lower layer. The Re for the cell-pack layer ranges from 0.005 to 1.5, while the Re for the plasma layer ranges from 9 to 40, for deceleration rates of 0.031 m/s$^2$ and 0.628 m/s$^2$. In our hematocrit experiments, we could not show that the thickness of the lower layer affects the stability of the system. According to the experiments of Sangalli et al. [117] and Barthelet et al. [118] and the observations of Hu et al. [121], our 30% hematocrit experiments should have been more unstable than the 50% hematocrit since the 30% hematocrit experiments have the more viscous layer occupying less of the container. We did not see this result. Also, Kao noted that having a less viscous upper layer was always stable [119], which is not the case for our experiments. Our system is similar to the results that Yih obtained for semibounded flow, which state that the velocity difference is destabilizing if the narrow layer is the more viscous layer. [113]

The induced centrifugal force appears to have a stabilizing effect that the viscosity difference must overcome to produce the instability. This is consistent with the Kelvin-Helmholtz model, where for a given change in density, the instability goes as $g_c/\dot{\gamma}^2$. For our system we have a critical $g_c/\dot{\gamma}^2$ ratio of $9.4 \times 10^{-3}$. Without a formal wave-length instability analysis, it is hard to say what sizes of wavelengths affect the system and whether or not the wavelength size that affects the system changes with either gravity or shear rate. There is a possibility that the results of the Renardy [122] and Bresch [123] studies (which state that as gravity increased there was an increase in the wavelength at which the short-wavelength instabilities could occur) could apply to our sys-
tem. Another prevailing component of instabilities in two-phase flow is the interplay between the two-layer interface and the presence of a critical layer which can contain the interface or be inside either of the two layers [112]. It is in this critical layer that the instability begins and can contribute to an instability below the critical long-wavelength or above the critical short-wavelength. None of these developments have been examined for a two-layer flow bounded by a moving boundary and a free-surface, so while we may infer some phenomenological predictions, such discussions are not diagnostic.

6.4.2 Clinical application and the experimental outcomes

For the spinning disk apparatus to be clinically relevant, it is important to maintain the stability of the separated layer of blood in the disk during deceleration. Because the separation of blood components is produced by sedimentation, the duration and speed of spinning will affect the number of blood cells that are separated out from the plasma. For the clinical application of spinning to separate out pathogens (such as bacteria) from the other blood cells, it is necessary to decelerate the disk as quickly as possible. If the disk is decelerated too quickly, then mixing will occur. This leads to a narrow window of clinically useful deceleration rates. Thus, it is important to be able to determine when mixing will occur so that the disk can be decelerated quickly to minimize total processing time while maintaining a stable separation of the cell-pack and plasma layers.

In order to best prevent phase mixing, it is important to know both the cause of mixing and when the mixing will occur. As stated above, without a formally developed theory for this physical setting, we can only postulate as to the mechanisms behind the instabilities, and experiments are the only sure way to determine exactly when the mixing occurs. The experiments performed in this study were aimed at determining when the mixing will occur for a given deceleration rate and if any of the adjustable parameters (deceleration rate and thickness of the cell pack and plasma layers) could be identified as a source of the instabilities. The data of Figure 6.5 was aimed at determining both when the mixing was occurring and whether or not the baffles in the disk were causing the mixing. The baffles were seen as the design feature most likely to cause mixing. The results suggested that any contribution to mixing caused by baffles was not discernible.
The experiments represented in Figure 6.6 were aimed at determining whether the thicknesses of the cell pack \( (d_2) \) and plasma \( (d_1) \) layers had any effect upon the mixing. Experiments with differences in hematocrit identify the effect of blood cell thickness and plasma thickness with respect to the total thickness. The comparison of the 7 mL blood group with the 7 mL blood plus 1.5 mL PBS group tested the hypothesis that the plasma layer thickness \( (d_1) \) made a significant contribution to the stability. The comparison of the 7 mL blood plus 1.5 mL PBS group and the 8.5 mL blood group tested the hypothesis that the blood cell layer thickness \( (d_2) \) made a significant contribution to the stability. The comparison of the 7 mL blood group and the 8.5 mL blood group tested the hypothesis that the overall thickness \( (d_1 + d_2) \) made a significant contribution to the stability. As there were no statistically significant differences in any of these groups, we posit that the parameters of blood cell thickness, plasma thickness, and total thickness do not play a sufficiently significant role to affect the instabilities during deceleration in our limited and noisy system. The insensitivity of the critical disk speed at the onset of instability to hematocrit is very beneficial for future clinical applications because there will be no need to measure the hematocrit for each person and adjust the parameters of the diagnostic instrument. The insensitivity of the critical disk speed at the onset of instability to the plasma thickness and total thickness is also beneficial for future clinical applications because it allows for a range in the amount of blood and PBS that is placed in the disk. Thus, if a patient wasn’t able to give a full vial of blood, the volume could be adjusted by adding in some PBS to help with blood separation without the concern that it will increase the likelihood of mixing during the deceleration of the disk.

The instability is not highly correlated with the thicknesses of the layers, which is consistent with the observation that changing the thickness of either the cell pack layer or the plasma layer does not greatly affect the size of the shear rate discontinuity—only the location of the discontinuity with respect to the total thickness of the two layers. Recognizing that the onset of mixing is reliant on the deceleration rate only and not on the hematocrit, or relative size of the given layers, we developed a velocity deceleration curve taking into account the Richardson criteria boundary to define a region in which the mixing will not occur. The viability of this curve was tested on the system by doing dozens of decelerations with 7mL blood and 1.5 mL PBS. Following this curve during deceleration decreased the amount of time it takes to stop the disk by more than one minute from previous studies [26]. A decrease in spinning time means a quicker diagnosis.
which hopefully leads to a better patient outcome. The decrease in spin time also increases the bacterial recovery in the plasma, and an increase in bacterial recovery increases the signal-to-noise ratio in the diagnostics. It will also allow a smaller blood draw for pediatric and geriatric patients. The future application of theory to this problem may help improve the experimentally determined “window of stable deceleration,” thus allowing for even quicker decelerations.
CHAPTER 7. CAPTURING BACTERIAL DNA

This chapter deals with the extraction of DNA from the bacteria after it has been separated and concentrated as described in Ch. 5. This chapter explores the lysis of multiple types of bacteria with a variety of chemicals, multiple ways of breaking the DNA, different magnetic beads to collect the DNA, and recovery solutions for the DNA on the magnetic beads.

7.1 DNA Extraction

Lysis of the bacteria and subsequent release of the DNA was investigated using many different chemical and/or enzymatic treatments. There were 84 different lysis combinations examined using the chemicals and enzymes listed in Table 7.1. This project tested 3 major bacteria: E coli, K pneumoniae, and E cloacae. Bacterial concentrations used for all experiments were 1 x 10^8 CFU/mL. All DNA extractions were performed according to section 4.8, which included lysing the bacteria, capturing the DNA on magnetic beads, and eluting the DNA off of the beads. When testing lysing solutions, the only change between the experiments was the lysis solution; the capture and elution of the DNA remained constant. Therefore, it was hypothesized that the amount of DNA collected represented the efficiency of the lysis solution to release DNA from bacteria.

The lysis procedure developed by Birnboim and Doly [165] has become the most widespread bacterial lysis approach for plasmid preparations [419]; it uses a 1% sodium dodecyl sulfate (SDS) and 0.5 M sodium hydroxide (NaOH) solution. This lysis solution, along with variations in the concentration of both SDS and NaOH, was tested for its ability to release both genomic and plasmid DNA. Guanidine hydrochloride is a chaotropic salt that has been used both as a bacterial lysis solution [167] and to increase DNA binding to silica beads [420]. Because of this dual nature of guanidine hydrochloride (GHCl), it was employed in conjunction with the SDS and NaOH solutions. The efficacy of lysis solutions were quantified by the percent of DNA recovered. The percent of DNA recovered was determined by dividing the amount of DNA recovered from each
Table 7.1: List of bacterial lysis chemicals.

<table>
<thead>
<tr>
<th>Trade Name (or abbreviation)</th>
<th>Chemical name (or description)</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDS</td>
<td>sodium dodecyl sulphate</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>NaOH</td>
<td>sodium hydroxide</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>GHCl</td>
<td>guanidinium chloride</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>Sarkosyl</td>
<td>sodium lauroyl sarcosinate</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>Brij 58</td>
<td>polyoxyethylene (20) cetyl ether</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>Triton X-100</td>
<td>polyethylene glycol tert-octylphenyl ether</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>Tergitol NP-40</td>
<td>polyethylene glycol nonylphenyl ether</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>DOC</td>
<td>sodium deoxycholate</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>CTAB</td>
<td>cetyltrimethylammonium bromide</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>TCA</td>
<td>trichloroacetic acid</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>EtOH</td>
<td>ethanol</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>CA</td>
<td>citric acid</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>SAS</td>
<td>sodium 4-animosalicylate</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>lysozyme</td>
<td>mucopeptide N-acetyl muramoylhydrolase</td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>pronase</td>
<td>lytic enzyme from <em>Streptomyces griseus</em></td>
<td>Sigma Aldrich</td>
</tr>
<tr>
<td>Chelex</td>
<td>cross-linked polystrene</td>
<td>Sigma Aldrich</td>
</tr>
</tbody>
</table>

experiment by the theoretical amount of DNA contained within a bacteria, either *E. coli* or *K. pneumoniae* or *E. cloacae*. Figure 7.1 shows the DNA recovery percents for the lysis solutions testing different concentrations of SDS and NaOH, without GHCl on the left and with GHCl on the right. The graphic is shaded from green to red, with green representing low DNA recovery and red representing high DNA recovery. In Figure 7.1, NaOH increases going downward; SDS increases moving to the right; GHCl was used in experiments on the right half but not the left. Most chemical combinations resulted in a high DNA recovery for *E. coli* but a low DNA recovery for *E. cloacae*, suggesting that *E. coli* lyses fairly easily while *E. cloacae* is very difficult to lyse.
The presence of GHCl improved DNA recovery for most chemical combinations for *E. coli*, while it only improved some chemical combinations for *K. pneumoniae* and *E. cloacae*.

<table>
<thead>
<tr>
<th>No Lysozyme</th>
<th>E. coli</th>
<th>E. cloacae</th>
<th>K. pneumonia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NaOH</strong></td>
<td>0%</td>
<td>0.5%</td>
<td>1%</td>
</tr>
<tr>
<td><strong>SDS</strong></td>
<td>2%</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td><strong>GHCl</strong></td>
<td>0%</td>
<td>0.5%</td>
<td>1%</td>
</tr>
<tr>
<td>0 M</td>
<td>19.3%</td>
<td>24.0%</td>
<td>26.0%</td>
</tr>
<tr>
<td>0.05 M</td>
<td>5.3%</td>
<td>6.5%</td>
<td>14.9%</td>
</tr>
<tr>
<td>0.2 M</td>
<td>11.5%</td>
<td>34.7%</td>
<td>44.0%</td>
</tr>
<tr>
<td>0.5 M</td>
<td>24.9%</td>
<td>8.2%</td>
<td>5.9%</td>
</tr>
<tr>
<td>1 M</td>
<td>12.3%</td>
<td>6.9%</td>
<td>8.1%</td>
</tr>
<tr>
<td>2 M</td>
<td>11.2%</td>
<td>41.2%</td>
<td>46.9%</td>
</tr>
<tr>
<td><strong>SDS</strong></td>
<td>2%</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td><strong>GHCl</strong></td>
<td>0%</td>
<td>0.5%</td>
<td>1%</td>
</tr>
<tr>
<td>0 M</td>
<td>34.7%</td>
<td>43.3%</td>
<td>50.9%</td>
</tr>
<tr>
<td>0.05 M</td>
<td>12.7%</td>
<td>68.2%</td>
<td>62.1%</td>
</tr>
<tr>
<td>0.2 M</td>
<td>6.8%</td>
<td>45.4%</td>
<td>77.5%</td>
</tr>
<tr>
<td>0.5 M</td>
<td>7.5%</td>
<td>7.4%</td>
<td>12.6%</td>
</tr>
<tr>
<td>1 M</td>
<td>18.2%</td>
<td>70.1%</td>
<td>29.8%</td>
</tr>
<tr>
<td>2 M</td>
<td>11.0%</td>
<td>76.3%</td>
<td>75.5%</td>
</tr>
</tbody>
</table>

Figure 7.1: DNA recovery for each of the different combinations of SDS and NaOH for *E. coli*, *E. cloacae* and *K. pneumoniae*. The left side of the figure is without GHCl and the right side of the figure is with 6 M GHCl.

Lysozyme is an enzyme used to disrupt the bacterial cell wall of gram-negative bacteria [421], but lacks efficacy on most Gram-positive bacteria [422]. Several studies analyzing the effects of lysozyme on DNA purity and yield found that lysozyme improved both the purity and yield of the isolated DNA [423–425]. Experiments were performed which examined different concentrations and different incubation times to determine which combination was the best to use for multiple organisms. Lysozyme does not release DNA from bacteria; it enzymatically digests the peptidoglycan of the cell wall, which then allows other chemicals to release the DNA. Since the results from using the different combinations of SDS and NaOH showed that DNA was easily
recovered from *E. coli*, lysozyme was tested with *K. pneumoniae* and *E. cloacae* to determine the concentration and incubation time. Figures 7.2A and 7.2B show the percent of DNA recovered for different concentrations of lysozyme combined with a 0.5% SDS solution. Figures 7.2C and 7.2D show the percent of DNA recovered for different time lengths for a solution of 0.5 mg/mL lysozyme and 0.5% SDS.

Since the first set of experiments (Figure 7.1) showed that *E. cloacae* was the most difficult organism to lyse, a lysozyme concentration of 1 mg/mL was selected due to the result of Figure 7.2A. Experiments with variations in SDS, NaOH and GHCl were performed again, but this time the cells were first incubated in a 1 mg/mL lysozyme solution for 10 min and then received the lysis solution. Figure 7.3 shows the DNA recovery percents for the lysis with lysozyme ex-
experiments, testing different concentrations of SDS and NaOH, without GHCl on the left and with GHCl on the right.

<table>
<thead>
<tr>
<th></th>
<th>0 M</th>
<th>0.5 M</th>
<th>1 M</th>
<th>2 M</th>
<th>3 M</th>
<th>4 M</th>
<th>0 M</th>
<th>0.5 M</th>
<th>1 M</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaOH</td>
<td>55.1%</td>
<td>44.8%</td>
<td>48.0%</td>
<td>21.6%</td>
<td>31.4%</td>
<td>33.6%</td>
<td>45.7%</td>
<td>50.4%</td>
<td>54.9%</td>
</tr>
<tr>
<td>SDS</td>
<td>18.1%</td>
<td>1.1%</td>
<td>0.9%</td>
<td>2.0%</td>
<td>0.5%</td>
<td>1.7%</td>
<td>21.6%</td>
<td>17.7%</td>
<td>5.1%</td>
</tr>
<tr>
<td>GHCl</td>
<td>14.5%</td>
<td>20.1%</td>
<td>25.8%</td>
<td>33.0%</td>
<td>44.0%</td>
<td>0.2%</td>
<td>17.1%</td>
<td>18.2%</td>
<td>8.4%</td>
</tr>
<tr>
<td>SDS</td>
<td>75.3%</td>
<td>3.6%</td>
<td>11.1%</td>
<td>0.2%</td>
<td>0.3%</td>
<td>0.5%</td>
<td>13.7%</td>
<td>39.7%</td>
<td>9.7%</td>
</tr>
<tr>
<td>GHCl</td>
<td>6.2%</td>
<td>6.0%</td>
<td>21.6%</td>
<td>22.5%</td>
<td>30.5%</td>
<td>1 M</td>
<td>16.8%</td>
<td>25.3%</td>
<td>2.9%</td>
</tr>
<tr>
<td>SDS</td>
<td>0.9%</td>
<td>15.9%</td>
<td>17.2%</td>
<td>22.4%</td>
<td>21.3%</td>
<td>2 M</td>
<td>14.0%</td>
<td>3.4%</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

**Figure 7.3:** DNA recovery for each of the different combinations of SDS and NaOH for *E. coli*, *E. cloacae* and *K. pneumoniae* after receiving treatment with 1 mg/mL of Lysozyme for 10 min. The left side of the figure is without GHCl and the right side of the figure is with 6 M GHCl.

In addition to testing the SDS, NaOH, and guanidine hydrochloride, other chemical combinations found in the literature were examined with and without lysozyme. Tables 7.2 and 7.3 show the results for each bacteria against all of the literature chemical combinations without and with lysozyme, respectively. Figure 7.4 shows the average DNA recovered from all bacteria for all lysis solutions for a comparison of lysis with and without lysozyme. The lysis solution numbers found on the x-axis in Figure 7.4 correspond to Table 7.4. The results from the overall comparison reveal that the best lysis solution was 1 mg/mL of lysozyme for 10 min followed by 6 M GHCl and 1% SDS.
Table 7.2: DNA recovery for the different literature lysis solutions for *E. coli*, *E. cloacae* and *K. pneumoniae* without receiving lysozyme.

<table>
<thead>
<tr>
<th>Bacterial Lysis</th>
<th>DNA Recovery (%)*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>E. coli</em></td>
</tr>
<tr>
<td>5% Sarkosyl</td>
<td>3.3%</td>
</tr>
<tr>
<td>10% SDS</td>
<td>22.8%</td>
</tr>
<tr>
<td>0.5% SDS + Pronase</td>
<td>5.9%</td>
</tr>
<tr>
<td>1% SDS + 0.5% Triton X-100</td>
<td>2.2%</td>
</tr>
<tr>
<td>1.5% SDS + 5% citric acid</td>
<td>1.5%</td>
</tr>
<tr>
<td>2% SDS + 4.5% Ethanol</td>
<td>6.9%</td>
</tr>
<tr>
<td>0.5% Brij 58 + 0.2% SDS</td>
<td>1.5%</td>
</tr>
<tr>
<td>0.5% Brij 58 + 0.2% DOC</td>
<td>6.0%</td>
</tr>
<tr>
<td>1% Brij 58 + 0.4% DOC</td>
<td>19.2%</td>
</tr>
<tr>
<td>0.5% Brij 58</td>
<td>12.2%</td>
</tr>
<tr>
<td>0.5% Triton X-100</td>
<td>5.8%</td>
</tr>
<tr>
<td>2% Triton X-100</td>
<td>3.7%</td>
</tr>
<tr>
<td>5% Triton X-100</td>
<td>0.5%</td>
</tr>
<tr>
<td>10% Triton X-100</td>
<td>4.9%</td>
</tr>
<tr>
<td>0.6% Triton X-100 + 5M GuSCN</td>
<td>26.6%</td>
</tr>
<tr>
<td>1.2% Triton X-100 + 4.7M GuSCN</td>
<td>31.4%</td>
</tr>
<tr>
<td>1% NP-40 + 0.1% SDS</td>
<td>1.2%</td>
</tr>
<tr>
<td>1% NP-40</td>
<td>1.4%</td>
</tr>
<tr>
<td>2% NP-40</td>
<td>7.6%</td>
</tr>
<tr>
<td>1% CTAB</td>
<td>4.6%</td>
</tr>
<tr>
<td>6% sodium 4-animosalicylate</td>
<td>12.3%</td>
</tr>
<tr>
<td>10% TCA</td>
<td>0.7%</td>
</tr>
<tr>
<td>15% Cheelex + 0.1% SDS</td>
<td>8.4%</td>
</tr>
<tr>
<td>15% Cheelex + 0.1% SDS + 1% NP-40</td>
<td>4.1%</td>
</tr>
<tr>
<td>20% ethanol + 75mM citric acid</td>
<td>19.2%</td>
</tr>
</tbody>
</table>

* Yellow highlighted cells represent recoveries from 20% to 40% and red highlighted cells represent recoveries greater than 40%.
Table 7.3: DNA recovery for the different literature lysis solutions for *E. coli*, *E. cloacae* and *K. pneumoniae* after receiving treatment with 1 mg/mL of lysozyme for 10 min.

<table>
<thead>
<tr>
<th>Bacterial Lysis</th>
<th>DNA Recovery (%)*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>E. coli</em></td>
</tr>
<tr>
<td>5% Sarkosyl</td>
<td>31.1%</td>
</tr>
<tr>
<td>10% SDS</td>
<td>27.4%</td>
</tr>
<tr>
<td>0.5% SDS + Pronase</td>
<td>19.7%</td>
</tr>
<tr>
<td>1% SDS + 0.5% Triton X-100</td>
<td>12.3%</td>
</tr>
<tr>
<td>1.5% SDS + 5% citric acid</td>
<td>10.3%</td>
</tr>
<tr>
<td>2% SDS + 4.5% Ethanol</td>
<td>15.9%</td>
</tr>
<tr>
<td>0.5% Brij 58 + 0.2% SDS</td>
<td>1.2%</td>
</tr>
<tr>
<td>0.5% Brij 58 + 0.2% DOC</td>
<td>49.3%</td>
</tr>
<tr>
<td>1% Brij 58 + 0.4% DOC</td>
<td>14.6%</td>
</tr>
<tr>
<td>0.5% Brij 58</td>
<td>48.6%</td>
</tr>
<tr>
<td>0.5% Triton X-100</td>
<td>11.3%</td>
</tr>
<tr>
<td>2% Triton X-100</td>
<td>13.0%</td>
</tr>
<tr>
<td>5% Triton X-100</td>
<td>9.8%</td>
</tr>
<tr>
<td>10% Triton X-100</td>
<td>14.7%</td>
</tr>
<tr>
<td>0.6% Triton X-100 + 5M GuSCN</td>
<td>36.6%</td>
</tr>
<tr>
<td>1.2% Triton X-100 + 4.7M GuSCN</td>
<td>43.6%</td>
</tr>
<tr>
<td>1% NP-40 + 0.1% SDS</td>
<td>8.2%</td>
</tr>
<tr>
<td>1% NP-40</td>
<td>6.6%</td>
</tr>
<tr>
<td>2% NP-40</td>
<td>12.8%</td>
</tr>
<tr>
<td>1% CTAB</td>
<td>10.3%</td>
</tr>
<tr>
<td>6% sodium 4-aminosalicylate</td>
<td>17.7%</td>
</tr>
<tr>
<td>10% TCA</td>
<td>8.5%</td>
</tr>
<tr>
<td>15% Cheelex + 0.1% SDS</td>
<td>3.0%</td>
</tr>
<tr>
<td>15% Cheelex + 0.1% SDS + 1% NP-40</td>
<td>15.7%</td>
</tr>
<tr>
<td>20% ethanol + 75mM citric acid</td>
<td>2.2%</td>
</tr>
</tbody>
</table>

* Yellow highlighted cells represent recoveries from 20% to 40% and red highlighted cells represent recoveries greater than 40%.
Figure 7.4: Average DNA recovery from all bacteria (E. coli, E. cloacae and K. pneumoniae) for all tested lysis solutions (Figures 7.1 and 7.3, and Tables 7.2 and 7.3), both with and without lysozyme. The lysis solution numbers found on the x-axis correspond to Table 7.4. Error bars represent standard deviation.

7.1.1 Bacterial Lysis for DNA Extraction

As mentioned in Chapter 2, there are many ways that bacterial organisms can be lysed. However, due to the vast number of bacterial organisms, lysing an unknown organism presents a significant challenge. For example, the cell wall structure significantly varies between Gram-negative, Gram-positive, and Mycobacterium [426]. Figure 7.1 shows that even inside the Gram-negative classification, the effectiveness of a lysis solution varies. E. coli is generally considered to be an easily lysed organism, while it appears that E. cloacae is a difficult organism, as seen in Figures 7.1 and 7.3. Thus, efficiently recovering DNA from all 3 gram-negative bacteria was
Table 7.4: List of Lysis Solutions

<table>
<thead>
<tr>
<th>Lysis Solution</th>
<th>Chemicals</th>
<th>Lysis Solution</th>
<th>Chemicals</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>6 M GHCl</td>
<td>L18</td>
<td>0.5% Brij 58 + 0.2% DOC</td>
</tr>
<tr>
<td>L2</td>
<td>0.5% SDS</td>
<td>L19</td>
<td>1% Brij 58 + 0.4% DOC</td>
</tr>
<tr>
<td>L3</td>
<td>1% SDS</td>
<td>L20</td>
<td>0.5% Brij 58</td>
</tr>
<tr>
<td>L4</td>
<td>2% SDS</td>
<td>L21</td>
<td>0.5% Triton X-100</td>
</tr>
<tr>
<td>L5</td>
<td>4% SDS</td>
<td>L22</td>
<td>2% Triton X-100</td>
</tr>
<tr>
<td>L6</td>
<td>0.05 M NaOH</td>
<td>L23</td>
<td>5% Triton X-100</td>
</tr>
<tr>
<td>L7</td>
<td>0.2 M NaOH</td>
<td>L24</td>
<td>10% Triton X-100</td>
</tr>
<tr>
<td>L8</td>
<td>0.5 M NaOH</td>
<td>L25</td>
<td>0.6% Triton X-100 + 5M GuSCN</td>
</tr>
<tr>
<td>L9</td>
<td>1 M NaOH</td>
<td>L26</td>
<td>1.2% Triton X-100 + 4.7M GuSCN</td>
</tr>
<tr>
<td>L10</td>
<td>2 M NaOH</td>
<td>L27</td>
<td>1% NP-40 + 0.1% SDS</td>
</tr>
<tr>
<td>L11</td>
<td>5% Sarkosyl</td>
<td>L28</td>
<td>1% NP-40</td>
</tr>
<tr>
<td>L12</td>
<td>10% SDS</td>
<td>L29</td>
<td>2% NP-40</td>
</tr>
<tr>
<td>L13</td>
<td>0.5% SDS + Pronase</td>
<td>L30</td>
<td>1% CTAB</td>
</tr>
<tr>
<td>L14</td>
<td>1% SDS + 0.5% Triton X-100</td>
<td>L31</td>
<td>6% SAS</td>
</tr>
<tr>
<td>L15</td>
<td>1.5% SDS + 5% CA</td>
<td>L32</td>
<td>10% TCA</td>
</tr>
<tr>
<td>L16</td>
<td>2% SDS + 4.5% EtOH</td>
<td>L33</td>
<td>15% Chelex + 0.1% SDS</td>
</tr>
<tr>
<td>L17</td>
<td>0.5% Brij 58 + 0.2% SDS</td>
<td>L34</td>
<td>15% Chelex + 0.1% SDS + 1% NP-40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L35</td>
<td>20% EtOH + 75mM CA</td>
</tr>
</tbody>
</table>

very challenging. In fact, the best lysis solution, 1 mg/mL lysozyme for 10 min followed by 6 M GHCl and 1% SDS, only achieved an overall average DNA recovery of 46%, see Figure 7.4. While this is the best result for these experiments, there are still several variables that can be optimized: the wash solution, the type and amount of magnetic beads, the elution solution, and temperature. Section 7.3 investigates the type and amount of magnetic beads needed to recover the maximum amount of DNA. Depending on the results of the magnetic bead experiments, the amount of DNA recovered may be improved due to a difference in type or amount of the beads from the type and amount used in these DNA experiments.

Figure 7.2 shows that the highest average DNA recovery was achieved at 1 mg/mL lysozyme for *E. cloacae* (7.2A), at 0.5 mg/mL lysozyme for *K. pneumoniae* (7.2B), at 12 min for *E. cloacae* (7.2C), and at 10 min for *K. pneumoniae* (7.2D). However the using the Welch’s t-test, there was no statistical difference in time or concentration (except 0.05 mg/mL was statistically lower than others for both *E. cloacae* and *K. pneumoniae*); see Appendix Table A.7. Comparison of Figure 7.3 with Figure 7.1 reveals that lysozyme improved DNA yield for some chemical combinations but not for others. There are two important factors that affect the reproducibility
and completeness of the lysis: pH and growth phase. The pH of the resulting lysis solution affects the released DNA because DNA changes its structure above pH 12, which may affect the recovery of the DNA [427]. Thus, the reproducibility of the method, without any external buffering during lysis, depends on having enough cell mass to counteract a given volume of high-pH NaOH solution [419]. The phase of growth affects the completeness of lysis because cells become increasingly difficult to lyse as they transition into the stationary phase [428]. In an experiment testing the effects of growth phase on susceptibility to lysozyme, complete cell wall digestion was achieved for cells harvested in the early logarithmic phase within 5 min, but required 45-60 min for cells harvested in the stationary phase, and cells from very old cultures never achieved complete wall digestion [429]. Thus, 1-2 hours difference in cell harvesting could mean the difference between complete lysis and partial lysis. In order to grow, bacteria require many different metabolites in various concentrations. As a growth media for bacteria, blood contains an abundance of some metabolites, such as amino acids and nucleotides, but lacks others, such as free iron [430]. Nutrient concentration is patient specific, and makes it unclear as to which nutrients are freely available and which are limited for bacterial growth [430]. However, bacteria in blood would be expected to have a cell surface similar to the logarithmic phase instead of the stationary phase. Thus, the optimized lysis solution which was developed with bacteria in the stationary phase should be able to lyse all gram-negative bacteria isolated from blood.

In addition to the findings above, it is important to note that lysozyme is denatured by detergents, and therefore bacterial lysis involving lysozyme must be performed in two-steps, a cell wall removal step and then a cell membrane disruption step [431]. Performing such a two-step process, increases the number of chemicals that now affect the bacteria. Many of the non-ionic detergents have limited cell wall disruption capabilities on their own, but are excellent cell membrane disruptors [432]. Table 7.2 shows the limited lysis capabilities of non-ionic detergents on their own, such as Brij 58, Triton X-100, and NP-40. Comparing Tables 7.2 and 7.3 shows the increased affects that lysozyme has on DNA recovery for the non-ionic detergents. In fact almost all of the lysis solutions in Table 7.3 increased DNA recovery in the presence of lysozyme. This increase in DNA recovery from using lysozyme between Tables 7.2 and 7.3 contrasts the mediocre response from using lysozyme between Figures 7.1 and 7.3. It was postulated that part of the difference between the sets came from the lysozyme interacting with many of the macromolecules.
that the NaOH would interact with, decreasing the NaOH-macromolecule interactions causing an increase in the amount of unreacted OH\(^-\) ions, raising the pH of the solution. It should be noted that this hypothesis does not fully describe what is happening because the most concentrated NaOH solutions are not the worst DNA recovery solutions, a corollary to the above hypothesis. Other factors, such as the growth phase described above and chemical-enzyme interactions, may contribute to the noticed difference. Specific experiments controlling for the age of the bacteria as well as testing alternative enzymes could determine whether these factors were contributing to the lack of an expected increase in DNA recovery.

While alternative enzymes may or may not yield similar results to lysozyme, there are other enzymes, such as pronase, which do not require a two-step process but can remove the cell wall in the presence of detergents. Pronase has been shown to be an effective cell wall disruptor in the presence of SDS [431, 433, 434]. This would decrease the number of process steps, and potentially process time, needed to recover DNA from bacteria. Another enzyme alternative to lysozyme is mutanolysin. While mutanolysin is also susceptible to detergents, mutanolysin is an effective Gram-positive cell wall remover [435] and has been shown to work on both Gram-negative and Gram-positive bacteria [436]. This would allow for DNA recovery from an unknown bacterium since the structure of the cell wall does not affect the lysis solution’s ability to lyse the bacteria. Proteinase K is another enzyme often used in DNA extraction experiments that has been found to increase the yield of DNA [129, 437]. Proteinase K can be used by itself as well as in conjunction with lysozyme. One study found that both Proteinase K and lysozyme increased DNA recovery with the best recovery coming from a combined use of Proteinase K and lysozyme [423]. However, the combined use of Proteinase K and lysozyme adds an additional step in the process for Proteinase K incubation at an elevated temperature. Also, it was found that Proteinase K is very effective on Gram negative bacteria, but lacks efficacy on Gram positive bacteria [422].

One of the variables not addressed by experiments in this chapter is the chemical composition of the bacterial suspension fluid. For all of the experiments in this chapter, the bacteria were suspended in a PBS solution. Several studies have shown that the presence of EDTA and Tris improve the yield of DNA [423, 425, 438]. However, one study did find that while EDTA increased the DNA yield, it also decreased the purity of the DNA [423]. Furthermore, they did not use silica in their DNA purification step, which means that increasing the EDTA concentration may not
lower the purity of DNA collected from silica-coated magnetic beads. But this can only be confirmed through experiments controlling the EDTA concentration using two different purification methods, one with silica and one without it. Another study, which analyzed the effects of organic amines in the presence of lysozyme, found that Tris is far less effective in sensitising the cell wall to lysis than n-dodecylamine [425]. After incubating the bacterial cells in the lysozyme-EDTA-dodecylamine solution, complete lysis occurred almost instantaneously upon the addition of the detergent, whereas controls without the n-dodecylamine in solution frequently failed to complete lysis [425]. Thus including EDTA and either Tris or n-dodecylamine in the bacterial suspension fluid would increase the yield of DNA. However, the bacterial suspension fluid that will contain the bacteria in a whole process experiment (from blood to DNA, see Chapter 8) would be the back-flush solution. Therefore experiments would need to be performed to determine how the presence of EDTA and Tris would affect the recovery of bacteria from off of the filter.

The ideal bacterial lysis solution would be a single step process that effectively lysed all bacteria. Since this lysis solution has yet to be discovered, the recommended lysis solution would be a single step process with the solution consisting of EDTA, n-dodecylamine, pronase, 6 M GHCl and 1% SDS. The inclusion of pronase should allow the solution to effectively lyse any bacteria.

### 7.2 DNA Fragmentation

Dr. Adam Wooley’s group (Chemistry Department, Brigham Young University) is charged with the task of labelling segments of the collected DNA for antibiotic resistance genes and species genes. In their process, the length of DNA passed from this process to them is constrained to be below 1000 bp with an ideal length around 500 bp. Therefore, different ways to consistently fragment DNA to this length were investigated. These methods include shearing the DNA with a magnetic stir bar, a magnetic stir bar with polyethylene beads, an Ultra-turrax, a 34 gauge needle, and a sonicator. Due to the initial requirements for the microfluidic device, the device could not handle more than 1 atm, relative pressure, so forced shearing by very small channels or objects on the device was not investigated.
7.2.1 Stir Bar

A chamber with a stir bar (detailed in section 4.9) was tested because of the ease of application and because it could be used for mixing solutions together, such as the backflushed-bacterial solution and the lysis solution. Different stir bar speeds were examined as well as the inclusion of microspheres in order to determine the best protocol for DNA fracture (see section 4.9 for details). Figure 7.5 shows the image of the agarose gel with each lane having DNA sheared by the stir bar at different speeds. The results are labelled as follows: Ld represents a 100 bp ladder (1500 bp - 100 bp range), 0 represents unsheared lysate, 1 represents spinning in the chamber for 3 min at 1000 rpm, 2 represents spinning in the chamber for 3 min at 1275 rpm, 3 represents spinning in the chamber for 3 min at 1575 rpm, 4 represents spinning in the chamber for 3 min at 1850 rpm, 5 represents spinning in the chamber for 3 min at 2150 rpm, 6 represents spinning in the chamber for 3 min at 2450 rpm. These results highlight two main findings. First, there is no smear of DNA but distinct bands appear, indicating that either no DNA was shredded or that it only breaks DNA of a certain size. In talking with some microbiologists, the smear appearing at the top of the wells (7.5) was suggested to be protein that absorbed some of the DNA staining dye. Second, the main DNA band near the top of the gels represents the genomic DNA released from the bacteria. The faint bands seen at the level of the the ladder represent RNA and the di-mers and tri-mers formed by ribosomes. A successful DNA fracture environment will eliminate the genomic DNA band. From the gel staining, it appears that the BL21 bacteria of this study did not contain a plasmid.

Adding microspheres helps increase the shear in the fluid by having solid objects that move past one another, but they appear to produce the same effect as shearing the DNA without the microspheres. Figure 7.6 shows the image of the agarose gel with each lane having DNA sheared by microspheres in conjunction with the stir bar. The results are labelled as follows: Ld represents a 100 bp ladder, 0 represents unsheared lysate, 1 represents spinning in the chamber for 3 min at 1850 rpm with 10 µL of microspheres, 2 represents spinning in the chamber for 3 min at 1850 rpm with 20 µL of microspheres, 3 represents spinning in the chamber for 3 min at 1850 rpm with 35 µL of microspheres, 4 represents spinning in the chamber for 3 min at 1850 rpm with 100 µL of microspheres. This gel contains a similar DNA pattern to those in Figure 7.5. However, the larger amount of smear on the gel, under which the normal DNA profile can be seen, reveals that the microspheres weren’t completely cleared from the DNA solution after use and can be stained by
Figure 7.5: The image of the agarose gel showing the results of shearing DNA using a spinning stir bar for 3 mins at different speeds. The description for each label can be found in the text. The process for making and running the gel can be found in section 4.9.1. Part of the DNA ladder is labelled. RNA can be seen at the bottom of the gel.

Figure 7.6: The image of the agarose gel showing the results of shearing DNA using a spinning stir bar for 3 mins at different speeds. The description for each label can be found in the text. The process for making and running the gel can be found in section 4.9.1. Part of the DNA ladder is labelled. RNA can be seen at the bottom of the gel.
the dye since the unsheared lysate (0) was incubated with the spheres but not stirred. The results suggest that the stir bar is not capable of creating the needed shear to tear the DNA apart, with or without the microspheres.

### 7.2.2 Ultra-turrax

The Ultra-turrax was employed because it can generate much higher mixing speeds and shear stresses than can the stir bar. The Ultra-turrax was tested both at different speeds for 3 mins and at a set speed for different durations of time (see section 4.9 for details). Figure 7.7 shows the image of the agarose gel with each lane having DNA sheared by the Ultra-turrax at different speeds, and at a constant speed for different lengths of time. The results are labelled as follows:

![Image of agarose gel](image)

**Figure 7.7**: The image of the agarose gel showing the results of shearing DNA using an Ultra-turrax at different speeds for 3 min and at 6500 rpm for different amounts of time. The description for each label can be found in the text. The process for making and running the gel can be found in section 4.9.1. Part of the DNA ladder is labelled. RNA can be seen at the bottom of the gel.

Ld represents a 100 bp ladder, 0 represents unsheared lysate, 1 represents shearing the lysate for 3 min at 6500 rpm, 2 represents shearing the lysate for 3 min at 9500 rpm, 3 represents shearing the lysate for 3 min at 13500 rpm, 4 represents shearing the lysate for 3 min at 17500 rpm, 5 represents shearing the lysate for 15 s at 6500 rpm, 6 represents shearing the lysate for 30 s at 6500 rpm, 7
represents shearing the lysate for 2 min at 6500 rpm, 8 represents shearing the lysate for 1 min at 6500 rpm, and 9 represents shearing the lysate for 10 min at 6500 rpm. Once again the image of the agarose gel shows that there is no smear of DNA, but distinct bands of DNA.

### 7.2.3 Syringe

DNA fragmentation was also examined using a syringe and needle because it is a simple method that does not require any expensive pieces of equipment. Figure 7.8 shows the image of the agarose gel with each lane having DNA sheared by a syringe and narrow needle. DNA fragmentation by needle was explored by multiple needle expulsions and through 2 different needle gauges. The results are labelled as follows: Ld represents a 100 bp ladder, 1 represents shearing the lysate by a 30 gauge needle with sucking up and expelling the solution twice, 2 represents shearing the lysate by a 30 gauge needle with sucking up and expelling the solution 5 times, 3

![Figure 7.8: The image of the agarose gel showing the results of shearing DNA using a needle. Different guages and multiple expulsions were tested. The description for each label can be found in the text. The process for making and running the gel can be found in section 4.9.1. Part of the DNA ladder is labelled. RNA can be seen at the bottom of the gel.](image-url)
represents shearing the lysate by a 25 gauge needle with sucking up and expelling the solution twice, 4 represents shearing the lysate by a 25 gauge needle with sucking up and expelling the solution 5 times. Neither needle gauge nor the number of times the solution was expelled was able to create a smear of DNA but again had distinct bands.

7.2.4 Sonication

Sonication was also explored as a possible method for DNA fragmentation. For comparison with the previous DNA fragmentation methods, the sonication was performed in a microcentrifuge tube with a 3 mm microtip. Different sonication amplitudes and different amount of sonication time were tested (see section 4.9 for details). Figure 7.9 shows the image of the agarose gel with each lane having DNA sheared by sonication for different lengths of time and at different amplitudes. The results are labelled as follows: Ld represents a 100 bp ladder, 0 represents unsheared DNA, 1 represents shearing the lysate by repeating 10 times – 30 s of sonication at 30% amplitude followed by 90 s of no sonication, 2 represents shearing the lysate by sonicating for 60 s at 30% amplitude, 3 represents shearing the lysate by repeating 5 times – 60 s of sonication at 30% amplitude followed by 90 s of no sonication, 4 represents shearing the lysate by sonicating for 60 s at 30% amplitude in an ice bath ($\approx 0 \, ^\circ C$), 5 represents shearing the lysate by repeating 5 times in an ice bath ($\approx 0 \, ^\circ C$) – 60 s of sonication at 30% amplitude followed by 90 s of no sonication, 6 represents shearing the lysate by sonicating for 60 s at 10% amplitude, 7 represents shearing the lysate by sonicating for 60 s at 20% amplitude, 8 represents shearing the lysate by sonicating for 15 s at 30% amplitude, 9 represents shearing the lysate by sonicating for 15 s at 20% amplitude, 10 represents shearing the lysate by sonicating for 45 s at 20% amplitude, 11 represents shearing the lysate by sonicating for 45 s at 30% amplitude, 12 represents shearing the lysate by sonicating for 75 s at 30% amplitude, 13 represents shearing the lysate by sonicating for 75 s at 20% amplitude, 14 represents shearing the lysate by sonicating for 90 s at 20% amplitude, 15 represents shearing the lysate by sonicating for 90 s at 30% amplitude. As can be seen in the gels, there is a distinct DNA smear present indicating that the DNA has been broken into smaller fragments. The gel lanes that do not contain any residual genomic DNA bands indicate that approximately all DNA has undergone some fragmentation; DNA bands with a concentration less than 0.3 ng/$\mu$L cannot be seen on an agarose gel.
These results are promising; not only does sonication break up approximately all of the DNA, it breaks it up into the size range of approximately 200 bp to 3000 bp (3000 bp is approximately the top of the DNA smear in the labelled gel lanes 9). The results reveal that a 10% amplitude ($\approx 0.5 \text{ W/cm}^3$, gel lane labelled 6) does not fragment the DNA while increasing the amplitude to 20% ($\approx 1 \text{ W/cm}^3$, gel lane labelled 7) does fragment the DNA. The results also reveal
that temperature does not affect the sonication results because the labelled gel lanes 2 (room temp) and 4 (0°C) as well as 3 (room temp) and 5 (0°C) have approximately the same fragmented DNA size range. From the gels shown in Figure 7.9, it is hard to discern how the amplitude and time length affects the size range of the fragmented DNA.

Because sonication was the only method that indicated that DNA was being fragmented, it was further explored to determine how well it could fragment the DNA when being implemented on a microfluidic device. This meant that the sonication would need to be performed through a plastic medium to replicate the environment of a microfluidic device. Two different plastic films were tested, a thin 0.5 mm sheet of 3D printed acrylic and a thin 0.1 mm sheet of polypropylene (PP). Due to the resolution of the printer, a 3D printed sheet thinner than 0.5 mm could not be reproducibly obtained between prints. The setup and device used for the sonication experiments are detailed in sections 4.8.1 and 4.9. Figure 7.10 shows the results for the tests using the two types of plastics at different amplitudes for different lengths of sonication time. The sonication times, amplitudes, and material for Figure 7.10 can be found in Table 7.5. The ladders used in Figure 7.10 are 1 kbp ladders (10 kbp - 500 bp range).

<table>
<thead>
<tr>
<th>Sonication Time (s)</th>
<th>Amplitude</th>
<th>Acrylic</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>30%</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>30</td>
<td>45%</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>60%</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>60</td>
<td>30%</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>60</td>
<td>45%</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>60</td>
<td>60%</td>
<td>18</td>
<td>22</td>
</tr>
<tr>
<td>90</td>
<td>30%</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>90</td>
<td>45%</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>90</td>
<td>60%</td>
<td>19</td>
<td>23</td>
</tr>
<tr>
<td>120</td>
<td>30%</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>120</td>
<td>45%</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>120</td>
<td>60%</td>
<td>20</td>
<td>24</td>
</tr>
</tbody>
</table>
Figure 7.10: The image of the agarose gel showing the results of shearing DNA using sonication through different plastic mediums at different amplitudes and for different lengths of time. The description for each label can be found in the text. The process for making and running the gel can be found in section 4.9.1. Part of the DNA ladder is labelled. RNA can be seen in the bottom right gel in lanes 21 & 22.
These results reveal the increased difficulty for the ultrasound to fragment the DNA when the sound waves have to pass through a medium. The size range of the DNA fragments extend from around 200 bp up to 10 kbp, a much larger range than when the ultrasound was applied to the DNA solution in a microcentrifuge tube. Even when sonicating for 2 min, the fragmented DNA still maintained a large spread in the size of the fragments. In comparing the size ranges between the different parameters in Figure 7.10, the best parameters appear to be 30% amplitude for 1 min using either the acrylic or PP sheet, 60% amplitude for 1 min using the acrylic sheet, or 60% amplitude for 2 min using either the acrylic or PP sheet. However, achieving reproducibility when running replicate samples on a different days turned out to be very difficult.

Figure 7.11: A representative plot (the gel containing labels 17-20 in Figure 7.10) of the pixel values for each well on the agarose gel. The higher values represent brighter fluorescent spots. Since the gels are stained with a nucleic acid dye, the brighter the fluorescence in an area represents more DNA in that area.

In order to determine the average DNA size for the bulk region (the brightest region in the DNA smear), the images of the agarose gels were uploaded into ImageJ and the relative intensity of
the fluorescence was calculated by converting the image to a grey scale. By converting the image to grey scale, the brightest pixel on the image received a value of 255 and the darkest pixel on the image received a value of 0. Figure 7.11 shows the results of plotting the values of the pixels for a representative plot (the gel containing labels 17-20 in Figure 7.10). The relatively low fluorescence and the dispersiveness of the DNA smears can be seen in Figure 7.11. From the pixel values, the location of the maximum value could be found for each well. This value was then compared to the bands on the ladder and a size determined. Figure 7.12 shows the approximate size (in kbp) of the maximum value for the different amplitudes at each sonication time for both plastics. These results revealed that there were additional parameters that needed to be controlled but were not. There was a large spread across all of the parameters and the statistics showed that no set of parameters were statistically different from any other set of parameters. The set of parameters with the smallest average across the 4 repeats was 60% amplitude for 2 min using the PP sheet which achieved an average fragment sized of 890 bp.

Figure 7.12: The fragment size of the region with the brightest fluorescence for each of the different parameter combinations (amplitude and duration of sonication) for both A) the PP sheet and B) the acrylic sheet.
7.2.5 Discussion

While several methods for fragmenting DNA were investigated, only sonication fragmented the DNA. The ultrasound performed best when the tip of the device could be inserted into the DNA solution. However, when being performed on a microfluidic device, the ultrasound tip will not be able to be inserted into the fluid. In order to simulate sonicating on a microfluidic device, a sonication chamber was built that could incorporate plastic strips which would provide a barrier between the tip of the ultrasound device and the DNA suspension. The results from sonicating the DNA through a plastic strip showed the plastic decreased the fragmentation of the DNA. From comparing Figures 7.9 and 7.10, we can see the full extent of the hindrance of the plastic strip. The size of the largest DNA band on the ladders in Figure 7.9 (direct contact) is the same size as the 3rd DNA band from the bottom of the ladders in Figure 7.10 (plastic film separation).

In addition to the diffuse DNA smear created by sonicating through plastic, Figure 7.12 revealed that the fragmentation was not reproducible across different experiments. There are many factors which are hard to control when sonicating through plastic. Some of these factors are the sonicating coupling fluid, the distance that the ultrasound tip is to the plastic sheet, and the sonication chamber set up. The sonicating coupling fluid refers to the fluid needed to transmit the ultrasound waves from the tip to the plastic sheet since the tip could not be in contact with the plastic. Degassed distilled water was used for the sonicating fluid, but depending on how vigorous the fluid was added and/or how recent the fluid was degassed, air could be introduced back into the fluid and limit the ability of the sonicator to get ultrasound waves to the DNA. Also, the initial set up of the ultrasound device involved raising the sonication chamber up to the tip. In an effort to increase the reproducibility of the distance the ultrasound tip was from the plastic, a stop was put in place that the platform was raised into, but depending on how soft or hard the user raised the platform into the stop would allow some discrepancy between the trials. Lastly, depending on exactly how the plastic strip was placed over the DNA solution in the sonication chamber, air bubbles could be introduced into the system and block the ultrasound waves. While all of these variables were controlled as best as possible, there will always be slight variations in every set up. Because there was no statistical significance between the ultrasound parameters evaluated, the whole process experiments performed in Chapter 8 were sonicated at 30% amplitude for 30 s using the polypropylene sheet. While Figure 7.12 shows the large variation in DNA fragmentation
at 30% amplitude for 30 s using the polypropylene sheet, it also reveals that these parameters can achieve the desired fragmentation.

7.3 DNA Purification

There are pros and cons to each of the DNA purification techniques reviewed in Ch. 2. Because this process initially needed to be compatible with a pre-existing microfluidic system, DNA purification by superparamagnetic beads was chosen. There are several different commercially available superparamagnetic beads. Three different superparamagnetic beads were bought and their ability to recover DNA from our lysis solution was evaluated. These experiments were performed as follows: DNA was extracted from bacteria according to section 4.8 and pooled together until 2 mLs of a DNA solution was obtained. Then, 200 µL of the solution was removed into a new tube and the desired amount of beads were added along with 100 µL of 6 M GHCl and 200 µL of isopropyl alcohol. The beads were incubated for 3 min in the solution and then a magnet was applied to side of the tube. The solution was removed into a new microcentrifuge tube and 50 µL of TE buffer were added to the beads. The beads were vortexed for 2 min in the solution and then a magnet applied to the tube and the solution removed into a new microcentrifuge tube. The original DNA solution and the eluted DNA solution were analyzed by fluorometer to determine the percent of the DNA that the beads captured and released. Figure 7.13 shows the percent of the DNA that was captured and released by the 3 different types of commercially available beads. The results show that there is an optimum amount of beads for DNA capture and release at 25 µL of beads. The results also show that the beads are not equal as the AbraMag™ beads struggled to collect any DNA. The other two beads were very similar although the Dynabeads™ beads did have a higher average at 25 µL than the Spherotech beads. Part of this discrepancy between bead type may be due to the differences in concentration between the beads, since Dynabeads™ had the highest concentration of 40 mg/mL of 1-µm beads, Spherotech had a concentration of 25 mg/mL of 0.5-µm beads, and AbraMag™ had a concentration of 12 mg/mL of 1-µm beads. At 25 µL of beads, there is 1.0 g of Dynabeads™, 0.6 g of Spherotech, and 0.3 g of AbraMag™.
Figure 7.13: Percent DNA recovery for different amounts of 3 commercially available superparamagnetic beads. Bead 1 is the Dynabeads™ beads, bead 2 is the AbraMag™ beads, and bead 3 is the Spherotech beads.

After determining that either the Dynabeads™ or the Spherotech superparamagnetic beads were acceptable to use and that the optimum amount of beads was 25 µL, the next set of experiments were aimed at determining whether the size of the DNA fragment mattered for bead collection. These experiments were performed in a similar fashion to the experiments above, except that the DNA used was either 250 bp antibiotic resistance gene fragments, 500 bp antibiotic resistance gene fragments, 2800 bp plasmid containing the antibiotic resistance gene, or the DNA ladders used in the agarose gels. Figure 7.14 shows the amount of DNA recovered from testing a 2800 bp plasmid, a 500 bp segment enzymatically cut from a plasmid, a 250 bp segment enzymatically cut from a plasmid, and a DNA ladder with segment lengths ranging from 15 kbp down to 100 bp. Figure 7.15 shows the image of the agarose gel with each lane consisting of the recovered DNA from DNA ladders. The results of the experiments show that the beads struggled to collect the smaller sizes of DNA segments. The plasmid was collected and released with ≈ 70% recovery,
Figure 7.14: Percent DNA recovery when testing the capture and release of different segment lengths using 25 µL of Dynabeads™ beads in 200 µL of DNA solution. Blue circles represent the plasmid, 500 bp and 250 bp segments that were collected individually by the beads. The red line is drawn at 1000 bp segment for reference.

while the 250 bp segments were collected and released with ≈ 30% and 500 bp segments were collected and released with ≈ 10% recovery. The DNA ladders confirmed the results with the DNA bands at or below 1000 bp showing up very faintly or not at all. The collection and release of the 250 bp and 500 bp segments had higher recoveries when collected by themselves than the 200-300 bp and 500 bp segments that were collected and released as part of the DNA ladder.

7.3.1 Capture and Release of DNA

An important aspect of the purification technique is the trade off between purity and yield. The ultimate goal is to have 100% yield with 100% purity, but this is an unattainable goal. Therefore it is important to understand how each part of the process affects the purity and/or yield. For
example, it was hypothesized that all of the commercial beads would be able to capture and release the DNA within a couple of percents of each other. However, there was one type of bead that struggled to capture and release the DNA regardless of the amount of beads added to the solution. It was postulated that the DNA recovery using superparamagnetic beads would follow a bell curve, in that the amount of DNA recovered would increase until the optimum amount of beads was obtained and then the DNA recovery would decrease since too many beads could physically trap the DNA between beads preventing release as well as struggle to release the DNA because of the low concentration of DNA on the beads. However, the DNA recovery did not follow a bell curve, but the optimum amount was established at 25 μL of beads (see Figure 7.13) which corresponds to about $\approx 1.6 \times 10^9$ beads.
Figures 7.14 and 7.15 show that it is important to maintain the size integrity of the DNA in order to increase the yield. If the DNA is broken up too soon, either by sonication or by accidental fragmenting during the process, into very small pieces, then a significant decrease in the yield will occur. From Figure 7.14, the drop off in yield comes below 1500 bp segments. This could potentially be due to the DNA being too small to overcome the repulsion force felt by the DNA from the beads which contain a slightly negative charged beads or to too strong of an interaction between the beads and the DNA such that the DNA cannot be removed from beads. Therefore, in order to maximize the recovery of the DNA, sonicating the DNA should wait until after or during the DNA elution step.

### 7.4 Limits of Detection

The importance of this project relies on its ability to be able to deal with extremely low clinical concentrations of bacteria and DNA. Bacterial concentrations of 10 CFU/mL yield very small quantities of DNA, even if it was possible to recover 100% of the DNA. On many of the current laboratory instruments that measure DNA, this amount of DNA is below their limit of detection. Therefore, a small study was performed to determine the limits of detection for the two main instruments used to measure DNA: fluorometry and quantitative polymerase chain reaction (qPCR). A bacteria contains around 5 million base pairs of DNA in it’s genome and some bacteria can carry upwards of 500 copies of plasmids. Using an average molecular weight of 675 g/mol for the DNA bases, a bacterium would contain approximately $5.6 \times 10^{-6}$ ng of genomic DNA. Using this value, Table 7.6 shows the resulting concentration of DNA for different concentrations of bacteria. Some of these DNA concentrations are extremely small. The limit of detection for fluorometry is determined by calculating the DNA concentration whose average fluorescence value is just above the highest fluorescence value for the 0 ng/mL standard. Figure 7.16 shows the limit of detection plot using the fluorometry standards for the fluorometer. The limit of detection for the fluorometer was determined to be approximately 1 ng/mL of DNA, which is approximately $1.8 \times 10^5$ CFU/mL. This is an unrealistically high concentration of bacteria in a blood infection.
Table 7.6: Concentration of DNA in different concentrations of bacteria.

<table>
<thead>
<tr>
<th>Bacterial Concentration (CFU/mL)</th>
<th>DNA Concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^8$</td>
<td>560 ng/mL</td>
</tr>
<tr>
<td>$10^7$</td>
<td>56 ng/mL</td>
</tr>
<tr>
<td>$10^6$</td>
<td>5.6 ng/mL</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.56 ng/mL</td>
</tr>
<tr>
<td>$10^4$</td>
<td>56 pg/mL</td>
</tr>
<tr>
<td>$10^3$</td>
<td>5.6 pg/mL</td>
</tr>
<tr>
<td>100</td>
<td>0.56 pg/mL</td>
</tr>
<tr>
<td>10</td>
<td>56 fg/mL</td>
</tr>
</tbody>
</table>

Figure 7.16: Plot of the average, highest and lowest fluorometer reading for different lengths of DNA segments.
The limit of detection for qPCR is determined by calculating the concentration where the qPCR instrument begins to give the same (or nearly the same) CT values to lower concentrations. Figure 7.17 shows the limit of detection plot using the EGFP plasmid for the qPCR instrument. The limit of detection for qPCR was determined to be approximately $1.25 \times 10^3$ copies/mL ($\approx 3.9$ fg/mL), which is lower than fluorometry limit of detection. One copy corresponds to one bacterium when there is only one copy of the target sequence in the bacterium. If the target sequence is found multiple times on the bacterial genome or the bacteria harbors multiple plasmids which contain the target sequence, then a lower concentration of bacteria could be detected. While $1.4 \times 10^3$ copies/mL ($\approx 4.4$ fg/mL) is lower than fluorometry limit of detection, this is still not low enough to detect clinical concentrations of bacteria in blood which have only one copy of the target sequence per bacterium. Because of this limitation, very low clinical concentrations were never used in experiments.

![Figure 7.17: The CT value (the PCR cycle at which the relative fluorescence of the solution surpasses a pre-defined threshold) for 10-fold dilutions of the EGFP plasmid. The limit of detection can be seen in the overlap of the three plasmid concentrations which can be found in the bottom right corner of the graph. The plasmids were extracted from *E. coli* using a commercial plasmid extraction kit (Qiagen).](image-url)
CHAPTER 8. MICROFLUIDIC DESIGN

This chapter describes experiments that measure the efficiency of the complete process from bacterial extraction out of blood to the elution of DNA from off of the beads. This chapter also presents different microfluidic designs and the results of DNA recovery using different microfluidic prototypes.

8.1 Whole Process

To measure the efficiency of the whole process, experiments were performed according to section 4.11. Figure 8.1 provides an overview of the procedure for the experiments covering the whole process. During the weeks of running whole process experiments, several changes were made. These changes include disk design alterations, disk holding time changes, different backflushing solutions and different ultrasound procedures. Disk design alterations and disk holding time changes were made due to the results of work performed by another graduate student [64]. Whole process experiments began while the optimum backflush solution chemistry was being investigated. Only the initial backflush experiment had been completed (see Figure 5.6A and section 5.3 for details), and the results of the first few complete experiments showed that the bacteria did not come off as easily in the presence of blood and detergents as it did in water. Thus, more backflush experiments were performed and eventually an optimum solution was found and implemented in the whole process experiments. Also, as microfluidic device testing began, the ultrasound procedure was adjusted to create a more reproducible set up in order to decrease the variability between runs.

The results of the whole process experiments are presented in Figures 8.2 and 8.3. Figure 8.2 shows how the recovery compares between backflushing the bacteria from the filter and recovery of the bacterial DNA after bacterial lysis. Figure 8.3 shows the bacterial recovery after filtering, the DNA recovery from the filtered bacteria, and the overall recovery of the whole process.
from the starting bacteria concentration to the eluted DNA. The overall recovery is calculated by multiplying the spinning, filter, and DNA recoveries together. The spinning recoveries ranged from 54% to 75% bacterial recovery. The spinning recoveries are not shown explicitly in the graphs.

As seen in Figure 8.2, the DNA recoveries range from near 0% recovery to near 100% recovery, even at bacterial concentrations as low as $5 \times 10^3$ CFU/mL. Bacterial concentrations lower than this value are lower than the detection limit of the fluorometer. There is no trend in the data, and it is unknown why one experiment recovered 90% and another experiment only recovered 10%. The filter recoveries range from near 0% recovery to 80% recovery. However, majority of the recoveries are near 0% with some data points clustered around 30% and only a few higher than 50% recovery. It is important to note that the higher filtration recoveries mainly correlate with the higher bacterial concentrations.

As seen in Figure 8.3, the DNA recoveries are spread across the testing time frame with no trend. However, the DNA recoveries in the month of December 2019 have a higher average than the preceding months, which is due in part to the standardization of the ultrasound process, in part to the optimized backflush solution, and in part to the experience gained over the preceding months. The filter recoveries increased in the last 3 months of running the whole process, averaging about 15% recovery over those 3 months, while the average for the rest of the experiments was 0.5%, with the exception of 3 days between April and May which are discussed below. The best overall recovery for any experiment was 26.8%, which comes from a 68% recovery of the bacteria from
Figure 8.2: Filter and DNA recovery results (triangle and circle symbols, respectively) from the whole process experiments. A) The filtration recovery and DNA recovery plotted against the amount of bacteria recovered from filtering. B) The DNA recovery results plotted against the filtration recovery results. For both A and B, blue represents bacterial counts $< 10^4$ CFU/mL recovered from backflushing, orange represents bacterial counts between $10^4$ and $10^5$ CFU/mL recovered from backflushing, grey represents bacterial counts between $10^5$ and $10^6$ CFU/mL recovered from backflushing, and yellow represents bacterial counts $> 10^6$ CFU/mL recovered from backflushing. The DNA recovery shows the amount of DNA recovered from the bacteria that were recovered from backflushing, not all bacteria.

spinning, a 54.1% recovery of bacteria from off of the filter and a 72.9% recovery of the DNA. The group of higher overall recoveries seen in Figure 8.3C were all achieved in the last month of experiments (December 2019).

8.1.1 Discussion

As can be seen in Figures 8.2 and 8.3, the bacterial recovery from the filter is the most difficult part of the process and a higher filter recovery usually leads to a higher overall recovery. The anomalies were the 3 days between April and May in which more than 50% recovery was obtained from backflushing off of the filter while the DNA recovery was only about 2%. Those experiments had more than $1 \times 10^7$ CFU/mL of bacteria recovered in the plasma for filtering. With the tested filtering set up, more than $1 \times 10^7$ CFU/mL of bacteria completely covers the filter and creates additional layers of bacteria coating the filter. It is hypothesized that these additional layers are easier to remove because the bacteria in the additional layers do not interact with the filter membrane and therefore can be removed simply by the flow of the fluid. Therefore a bacterial
Figure 8.3: Filter, DNA and overall recovery (triangle, circle, diamond symbols, respectively) from the whole process experiments. A) The recovery results for DNA, filtration, and the whole process are plotted by date of experiment. B) The same figure as A except that data points are color coded by bacterial counts. C) The recovery results for DNA and filtration plotted against the overall recovery from the whole process experiment. For B and C, blue represents bacterial counts $< 10^4$ CFU/mL recovered from backflushing, orange represents bacterial counts between $10^4$ and $10^5$ CFU/mL recovered from backflushing, grey represents bacterial counts between $10^5$ and $10^6$ CFU/mL recovered from backflushing, and yellow represents bacterial counts $> 10^6$ CFU/mL recovered from backflushing.

Count of $1 \times 10^8$ CFU/mL could obtain a 90% recovery by only removing the additional layers of bacteria. This leads to the conclusion that minimizing the surface area of the filter will increase the bacterial recovery of the filter. However, it is unknown what effects shrinking the filter area will have on the filterability of the spun plasma solution.

The filter recoveries ranging from 15% to 30% combined with a bacterial concentration between $1 \times 10^5$ and $1 \times 10^6$ are truly recovering bacteria in contact with the filter, since this concentration of bacteria does not completely cover the filter. However, this is still not as high of a recovery as was achieved in the dedicated filtering experiments in Chapter 5. This discrep-
ancy of the recovery results (between the dedicated filtering experiments and the whole process experiments) may exist because the amount of RBCs was controlled for the dedicated filtering experiments, but there was a variable amount of RBCs in the whole process experiments. It is hypothesized that the lower bacterial recovery from off of the filter in the whole process experiments is due to more RBCs being retained in the spun plasma. There are many variables that affect the amount of RBCs retained in the spun plasma, making it hard to standardize this part of the lab procedure [111]. Yet, each step of the whole process is improving with time and the kinks are continually being worked out, which help make the process smoother, faster, and more reproducible. Integration into the microfluidic device may also improve the process as that would allow for better control and standardization of the fluid flow through the filter and the manipulation of the DNA.

8.2 Microfluidic Designs

As stated in Chapter 1, the first company that collaborated with the project already had a cartridge that they used for diagnostics in their machines, see Figure 8.4. This cartridge was the basis for developing the microfluidic devices for our process. There were four different designs that were used to identify the best setup for mixing and DNA retrieval. Three of the designs came from reading the literature and one design was developed to test the effectiveness of vibration (Figure 8.5B). The four designs can be seen in Figure 8.5.

Figure 8.4: Great Basin’s cartridge for using with their analysis machines.
Figure 8.5: The microfluidic prototypes tested in the lab. A) Initial channel design, inset shows Y-channel used to inject solutions. B) Vibration testing design. C) Magnetic moving design. D) Final channel design. See text for description of number labels.

From a quick search in the literature, the initial microfluidic device (see Figure 8.5A) was developed, which included a zig-zag mixing channel and mixing posts. For this device, 1) marks an inlet; 2) marks the zig-zag channel for mixing the solutions; 3) marks the mixing posts in the zig-zag channel where the magnetic beads were preloaded—25 µL of the bead suspension were pipetted into the zig-zag channel at the posts before the top and bottom pieces were bolted together; 4) marks the first chamber where a magnet can be applied to the backside of the bottom piece; 5) marks an outlet/inlet; 6) marks the channel between the two chambers; 7) marks the second chamber where a magnet can be applied to the backside of the bottom piece; and 8) marks an outlet/inlet. The device worked as follows: the separate bacteria and bacterial lysis solution were simultaneously injected into the device at 1 using the Y-connector (inset); the solutions were mixed in 2 and mixed with the beads in 3 and 4, by blocking 5 the mixed solution passed through 6 and into 7 where the magnet was applied to trap the magnetic beads with DNA attached; then the magnet could be removed and the wash solution injected from 8; the magnet was then applied to 4...
to catch the beads while allowing the washing solution pass out 5 (with 1 blocked); then the magnet could be removed from 4 and applied to 7 while the second washing solution was injected at 5 with 1 still blocked; then the magnet could be removed from 7 and applied to 4 while the elution solution was injected at 8 allowing the elution solution to push through such that the magnet catches the beads in 4 while the eluted DNA comes out at 5.

The second microfluidic device (see Figure 8.5B) differed in that it had no mixing channels but a mixing chamber in which vibration could be applied to mix the solution chamber. For this device, 1) marks an inlet/outlet; 2) marks the mixing chamber; 3) marks the magnet chamber where a magnet can be applied, this chamber also contains mixing posts; 4) marks an outlet/inlet. The device worked as follows: the bacteria, bacterial lysis solution, and beads were load into 2 one at a time; then a cell-phone vibrator (3V Micro DC Vibrating Motor, Amazon) was applied to the device to mix the solutions for 2 minutes; then the magnet was applied to 3 and the washing solution was injected in 1 to move the DNA-coated beads to 3 for collection; then the magnet was removed and one of the outlets (4) was blocked and the elution solution was injected through the other 4; once 2 was full, the cell-phone vibrator was applied again for 2 minute; then the magnet was applied to 3 and more elution solution was injected from 1 and pushed through 2 and 3, where the beads were caught while the free DNA came out 4.

The third microfludic device (see Figure 8.5C) studied whether the magnets could be easily moved from one chamber to the next rather than pushing solutions in and out. For this device, 1) marks an inlet channel; 2) marks the lysis mixing chamber; 3) marks a solution inlet channel; 4) marks the wash mixing chamber; 5) marks the elution mixing chamber; 6) marks the bead chamber; 7) marks an inlet/outlet. In theory, the device worked as follows: 2, 4, and 5 were loaded with the solutions (bacterial lysis solution and beads in 2, washing solution in 4, and elution solution in 5); then the bacteria are injected through 1 into 2 and mixed by moving the beads around using a magnet; then the beads are moved into 4 by magnet by moving the beads around the corner separating the chambers and into the wall of channel 3, which breaks the magnetic hold on the beads; then the beads are mixed with the washing solution in 4 by moving the magnet on and off the device and around in the chamber; then the beads are moved into 5 (same way as from 2 to 4) and once again mixed by magnet; then the beads are moved out of 5 and into 6, which contains no
solution; then the eluted DNA is removed by needle through 7. One drawback with this idea was that a needle had to be inserted to retrieve the eluted DNA. This idea came from reference [171].

The fourth microfluidic device (see Figure 8.5D) tested an improved channel mixing design. For this device, 1) marks an inlet/outlet; 2) marks the "J" channels for solution mixing; 3) marks a solution inlet channel; 4) marks the chamber where a magnet can be applied to the backside of the bottom piece. This device worked similarly to the first device as follows: 25 μL of magnetic beads were injected into one of the inlets (1); then the bacteria and the bacterial lysis solution were simultaneously inserted into each side of the Y-channel (1); then the solutions would mix in the "J" channels (2) along with the magnetic beads; then the solution would pass into the chamber (3) where a magnet was applied to the backside of the bottom piece; next the magnet could be removed and applied to the base of the Y-channel just before the split while the wash solution was injected from 4; then the magnet could be removed from the channel and applied to 3 while the elution solution was injected from both sides of the Y-channel (1) and pushed through such that the magnet catches the beads in 3 but the eluted DNA comes out 4. This channel idea came from reference [258].

As mentioned in Section 8.1.1, the microfluidic device can not only provided better standardization and control of the process, but can allow the process to become more reproducible, have less room for error, and eliminate many of the different sources of contamination since fluid isn’t being transferred from one tube to the next.

### 8.3 Microfluidic Device Tests

Each of the microfluidic devices described in the previous section were examined for their ability to lyse the bacteria and capture and elute the DNA. Working with the microfluidic devices were challenging. There were many sources of error in the experiments, such as fluid leaking between the top and bottom pieces causing a loss and dilution of the sample as additional wash or elution solution was then needed. Other problems consisted of not all beads being trapped in the magnetic field applied by the magnet as well as bead clumping (and lack of dispersion) in device A and D. Device C failed altogether since we could not get the beads to break the surface tension in order to travel from chamber 2 to chamber 4. The beads were tested both preloaded into chamber 2 as well as loaded with the bacterial lysis solution, but both experiments yielded the same result:
the beads could move around within the fluid but not out of it. In addition to the 3D printed top for device C, glass slides were tested to see if it was the polymeric material that was preventing the beads from moving out of the solution, but even on the smooth glass surface the beads still could not break the surface tension. Due to this problem, no DNA could ever be recovered from device C.

The results for the experiments for each device (A-D) are shown in Figure 8.6. The recoveries for device A and B are really low while the recovery for device D is higher but less reproducible. Another difficulty that potentially lead to such low recoveries is that a much larger volume of the elution solution was used to move the beads and the DNA across the device and the DNA out of the solution, thus diluting the solution and losing some solution to the inside walls of the device. These difficulties lead to imprecise measurements of the amount of solution that was recovered from the device, most likely lower than was expected. However, additional experiments were performed using device D to directly compare the microfluidic device to the benchtop method. Figure 8.7 shows the results of 4 direct comparison experiments between the microfluidic device and bentop method. These experiments were performed according to section 4.12, so that each method saw approximately the same number of bacteria. These disparate results highlight the inconsistency experienced when performing the whole process experiments, whether benchtop or microfluidic. The microfluidic device performed better on 2 experiments while the benchtop method performed better on the other two.

8.3.1 DNA Recovery from Microfluidic Devices

Comparing the DNA recoveries of Figure 8.6 to Figure 7.4 reveals that the microfluidic devices were acheiving similar results to the DNA extraction experiments. This is promising since the microfluidic experiments were tested on bacteria that had been in the presence of blood and other detergents, which means that the spinning, filtering and backflushing process do not greatly impact the lysis of the bacteria. However, 41% recovery is still low for the microfluidic device to perform in a clinical setting. When less DNA is recovered, there is a decrease in the signal-to-noise ratio and a greater chance of small errors producing a false negative result.
Figure 8.6: DNA recovery for each microfluidic device. Error bars represent standard deviation.

Figure 8.7: Comparison of microfluidic device D to the benchtop method.
CHAPTER 9. CONCLUSION

In conclusion, this dissertation developed and evaluated many aspects of the process required to identify bacteria from a blood stream infection without culturing the bacteria to increase the bacterial concentration. Removing the culturing step from the bacterial identification process of a blood stream infection provides a significant reduction in the processing time. As mentioned in the introduction, at least 12 hours, often 24-36 hours, is needed specifically for the culturing step in the bacterial identification process, and there is a large increase in the mortality rate every hour that effective treatment is delayed [12, 13]. However, eliminating the culturing step shifts the difficulty from time to concentration. There are many problems associated with very low bacterial concentrations, such as an increase in the ratio of contaminants to sample and a decrease in the amount of DNA available for identification. Therefore, the overall goal of this dissertation was to collect the maximal amount of DNA from the bacteria that have been separated out from the blood, and to concentrate the DNA in 100 µL within 7 minutes using a microfluidic device.

In order to accomplish this goal, several aspects of the bacterial identification process were investigated. The initial step of removing the bacteria from the blood had been previously developed by this lab (Dr. Willliam Pitt’s lab, Chemical Engineering, Brigham Young University) using centrifugal-sedimentation on a hollow disk [26, 76]. However, this dissertation explored two alternatives to the previously developed bacterial separation process: inducing clotting on the centrifugal-sedimentation disk (section 5.1) and filtering the whole blood (section 5.2.2). In order to determine if one of the alternative processes was better, the alternative process needed either to achieve a bacterial recovery higher than the 69% recovery obtained through the original process or to achieve a similar bacterial recovery with a significant reduction in processing time from the original processing time of 282 s.

Through the many experiments investigating the clotting process, parameters were identified that matched the previous bacterial recovery of 69% while achieving that recovery in only 117
s, a 58.5% reduction in processing time. These parameters were spinning 7 mLs of whole blood with 1 mL of 8 M calcium chloride up to a separation speed of 6000 rpm at a rate of 500 rpm/s and then held for 5 s, then decelerating at a rate of 100 rpm/s until a clotting speed of 1000 rpm was reached, which was held for 60 s; then decelerating at 200 rpm/s until the disk speed reached 0 rpm. Another advantageous aspect of this clotting process is that it used the existing disk style and motor set up of the original process and could be implemented simply by changing the spinning parameters (computer input parameters) and including the 1 mL of 8 M calcium chloride. However, this process was only ever tested at high bacterial concentrations ($\geq 1 \times 10^7$ CFU/mL) and very low bacterial concentrations ($\leq 200$ CFU/mL) would need to be further explored before a switch should be made. Also, the speed and length of time at which the disk was held at the clotting speed of 1000 rpm should be further investigated. The speed of 1000 rpm and the hold time of 60 s were used for comparison to other experiments, but these parameters were never varied to determine what the optimum conditions were.

The filtration process proved to be a little more difficult than the clotting process. Parameters were originally identified which allowed for the lysis and filtration of 5 mLs of whole blood while capturing 100% of the bacteria on the filter. These parameters included a blood lysis solution containing 4% Triton X-100 and 0.5 M NaCl and a 9:1 solution-to-blood ratio. Filtering all 50 mLs of the solution took between 60 s - 90 s, which is a 68% reduction in processing time at 90 s. While this process is significantly faster and captures all of the bacteria, 50 mLs of solution is a large amount of waste that would require bio-hazard processing. It was also noted later that while 100% of the bacteria are caught on the filter, not 100% of the bacteria come off of the filter. Therefore, additional experiments were performed in which the removal of bacteria from the filter, following filtering, was investigated using several different chemical solutions. While performing backflush experiments, it was discovered that the blood lysis solution was just as important to the bacteria removal as the backflush solution used to push the bacteria off of the filter.

Additional blood lysis solutions were tested in order to test different combinations of blood lysis and backflush solutions. The optimum parameters were able to achieve 80% recovery in about 150 s (this time only includes the filtering and backflushing time, no prep time), which is a 15.9% increase in bacterial recovery with a 46.8% decrease in processing time. These parameters included a 3% Pluronic F108 backflush solution, a 1% Brij 58-1% Tween 80-1% Pluronic F108
with 0.5 M NaCl blood lysis solution, and a hydrophobic track-etched filter with 0.4 \( \mu \)m pore diameters that was soaked in the backflush solution for 3 min before filtering. However, these parameters were determined using a bacterial concentration of \( 1 \times 10^7 \) CFU/mL. At a very low bacterial concentration of 200 CFU/mL, the optimum parameters could only recover 60% of the bacteria. These low-count optimal parameters included a 3% Pluronic F108 backflush solution, a 3% Tween 80 with no salt blood lysis solution, and a hydrophilic track-etched filter with 0.4 \( \mu \)m pore diameters that was not pre-soaked in the backflush solution. Because of the problems with the bacteria remaining on the filter and the huge volume of waste fluid generated from filtering whole blood, this alternative process is not recommended at this time. However, this recommendation could change depending on the results from future studies.

Future filtering studies should include exploring the solubilization of the RBCs for the different detergent solutions at several hematocrit values (1%, 5%, 20%, 40%, and 55%) in conjunction with the filterability of the mixture to create a correlation between solubilization and filtration. Other studies should include exploring coated filter surfaces, chemicals with alternative structures and chain-lengths such as zwitterionic detergents, enzymes such as proteinases and other protein denaturing compounds, and perhaps organic solutions (all of which are known to induce micelles in lipids); but it should be noted that organic solutions increase the health risks of the user and clinical settings try to avoid them. In addition to the above studies, chemical combinations should also be explored as the backflushing solution since only single chemicals were tried.

In addition to evaluating two alternatives to the original centrifugal-sedimentation process, an objective of this dissertation was to analyze the stability of the original process in order to decrease the occurrence of re-mixing of the separated plasma and cells, and to potentially decrease the processing time. This analysis yielded the development of the analytical solution to the Navier-Stokes equations for a two-fluid flow with a moving wall boundary and a free surface (section 6.1). Using the analytical solution, a linear stability analysis was performed. The linear stability analysis yielded an unexpected result in that the results showed that the system was always unstable. However, a linear stability analysis involves a time-independent analysis of the system. Thus, it is postulated that the deceleration aspect of the system is integral in determining whether or not the system is stable, which would result in the linear stability analysis yielding an unstable result because deceleration is not time-independent. Therefore, a non-linear stability analysis is needed to
mathematically determine the stability criteria for the system. But due to time constraints and the complexity of a non-linear stability analysis involving a free surface, a non-linear stability analysis was not performed. However, the Kelvin-Helmholtz instability model can incorporate a deceleration into the model for stability analysis. But this model only provides a necessary condition for stability not a sufficient condition, which means that it predicts at what point the system *could go* unstable but not whether the state of the system *is* stable or unstable. This model yielded a stability criteria for the system which was compared to the results of experiments, which analyzed at what time point the 2 layers (plasma and cells) mixed on the disk; it was found to be in good agreement with the experimental results. Using this criteria and the data from the stability experiments, a new velocity deceleration profile was developed which decreased the spinning time down from 282 s to 150 s. This new velocity deceleration was implemented into the centrifugal-sedimentation process.

Although the centrifugal-sedimentation process separates out the bacteria from the blood cells, it does not remove 100% of the blood cells and it leaves the bacteria suspended in approximately 4 mLs of plasma with about 1% RBCs. Another objective of this dissertation was to remove any remaining blood components since these can interfere with downstream processes, such as labeling the bacterial DNA. To accomplish this objective, blood-lysis filtration was employed (section 5.2.1). As discussed above, filtration of whole blood had already been investigated. However, additional experiments were performed to determine if there was a blood-lysis solution that could allow filtration of the recovered plasma in a 1:1 ratio. A blood-lysis solution was found that allowed filtration of 4 mLs of plasma in a 1:1 ratio. This solution consisted of 2% saponin with 33 mg of a protease. However, when tested with whole blood, the solution could not filter more than 2 mLs of whole blood in a 1:1 ratio. Further exploration was not pursued because a working solution had been found for recovered plasma. However, it would be advantageous to perform solubilization studies in the future that consisted of analyzing the relationship of different detergent concentrations to the percent of the blood cells that are solubilized at different blood hematocrits for multiple detergents.

After isolating the bacteria, the next objective consisted of formulating a single effective lysing solution that could lyse the 3 bacteria of this study (*E. coli*, *E. cloacae* and *K. pneumoniae*). This objective turned out to be more challenging than anticipated. There are many methods of lysing bacteria, but mechanical and electrical methods were foregone by the constraint that the
lysing technique needed to be compatible with the Great Basin microfluidic cartridge. Therefore, chemical and enzymatic methods were investigated (section 7.1). After examining several different chemicals and chemical combinations, the resulting best bacterial lysing solution consisted of incubating the bacteria with 1 mg/mL lysozyme for 10 min followed by the addition of 6 M GHCl and 1% SDS. However, this solution only achieved an average DNA recovery of 46% across all 3 species. But this is a minimum value in that additional parameters were later studied which could improve the DNA recovery of 46%. It was determined from experiments that the size of the DNA affected the ability of the superparamagnetic beads to both capture and release the DNA (section 7.3). The beads were ineffective in the capture and release DNA segments \( \leq 1000 \) bp in length, averaging only about 15% for these segments, while the average capture and release was about 55% for segment lengths between 1500 bp and 15 kbp. In addition to the size of the DNA segment, the supplier of superparamagnetic bead was of significant import. The best superparamagnetic beads were the Dynabeads\textsuperscript{TM} from ThermoFischer. One parameter not experimentally studied was the inclusion of Tris and EDTA in the lysis buffer, since it is well documented in the literature that these salts increase DNA recovery and it was assumed that these salts could simply be added in after the best detergent was found. However, this was a poor assumption to make since the bacteria which needed to be lysed will be suspended in the backflush solution. Thus, future experiments need to be performed to determine how these salts affect the bacterial recovery when backflushing. This also brings up another concern which is that the non-ionic detergents present in the backflush solution will inactivate any lysozyme added to the solution. Thus, the bacterial lysis solution cannot involve any enzymes that are sensitive to the detergents found in the backflush solution. Future experiments will also need to examine the lysis of the bacteria when suspended in the backflush solution. However, not all is lost, since the bacterial cell walls will already be weakened by suspension in a non-ionic detergent solution. When performing future bacterial lysis studies, the studies should analyze bacterial lysis by light scattering with a spectrometer; then take that same solution and spin out the bacteria for plate counts and measure the supernatant for any DNA. Additional future studies should explore different temperature ranges, varied concentrations of guanidine, how the stage of bacterial growth affects lysis by different detergents, and how different surface coatings to the silica-coated beads such as an amine coating affect DNA collection.
After extracting the DNA from the bacteria, the next objective involved concentrating the DNA into 100 $\mu$L or less and fragmenting it into $\leq 1000$ bp segments. The first part of the objective was very easy using superparamagnetic beads. Because the beads are magnetic, any solution can be replaced while holding the beads in one place. Thus, the final volume of the elution solution can be varied by simply adding in more or less volume. For the DNA extraction experiments above, 50 $\mu$L of distilled water were used for removing the DNA from the beads, well below the 100 $\mu$L maximum. The second part of the objective was more challenging.

Experiments with several different methods for fragmenting the DNA resulted in sonication being the method of choice (section 7.2). When sonicating the DNA in a microcentrifuge tube, a fragment size range of 200 bp - 1500 bp was achieved, effectively meeting the objective. However, when applied within a microfluidic device, the sonicating probe cannot be stuck into the DNA suspension. Thus, experiments were performed which analyzed the ability of sonication to fragment DNA when there was a plastic film barrier between the ultrasound device and the DNA suspension. These results yielded a wide spread of DNA fragment size ranges. The smallest size ranges were between 200 bp and about 3000 bp, while the larger size ranges ranged from 1000 bp to 15,000 bp. The data contained a lot of scatter, indicating that not all of the variables were well-controlled. The conclusion was that a 30% amplitude for 30 s with polyproplyene being the separation plastic film was sufficient to achieve the smallest size range for the DNA fragments. However, future experiments need to be performed in which the distance between the sonicating probe tip and the separation plastic is varied, in order to understand how that affects the size of the DNA fragments. Another future experiment could examine different well configurations to evaluate whether a small and deep well achieves smaller DNA fragments than a wide and shallow well.

Now that bacteria were isolated from the spun plasma and broken open and it’s DNA captured, released, and fragmented, the final goal was to put the entire process together. The whole process from spinning the blood through fragmented DNA, was evaluated (section 8.1). These experiments revealed that while great efficiencies and recoveries may be obtained in isolated controlled experiments, putting it all together introduced new obstacles. One of these obstacles was the inconsistency of spinning the blood to remove the same amount of RBCs from solution because variations in the amount of RBCs in the recovered plasma greatly affected the blood lysis.
solution’s ability to completely lyse all of the RBCs, which affected the bacterial removal from the filter and the subsequent bacterial lysis. Another obstacle is that the sub-perfect efficiencies multiply together. The best overall recovery from an experiment performing the whole process was 26.8%. The whole process is only \( \approx 25\% \) efficient; the individual recoveries are much higher but because there are so many steps in the process each sub-perfect performance increases the inefficiency of previous step until the overall system is not very efficient. For example, the overall recovery of 26.8% was achieved with a 68% recovery of the bacteria from spinning and a 54.1% removal of bacteria from off of the filter and a 72.9% recovery of the DNA from the bacteria. Thus, experiments on the whole process need to be continued in order for observations and progress to be made that will help tweak the process to increase the overall recovery of DNA.

Once experiments were being performed on the whole process, the next objective was to implement as much of the process as possible in a single microfluidic device. Several different prototypes were created and tested for their ability to perform the bacterial lysis, DNA capture, DNA release, and DNA fragmentation (section 8.3). Experiments were performed with the different prototypes, and the best prototype (see Figure 8.5D) was selected for direct comparisons of DNA recovery from the whole process using either the individualized benchtop method or the microfluidic device. These experiments showed that overall the microfluidic device performed similarly to the benchtop method. Although improvement would have been ideal, performing the same is better than the microfluidic device performing worse. Due to time constraints, a microfluidic device was not prototyped which incorporated the filtering into the device. Therefore, future experiments are needed to determine the feasibility of incorporating the filtering into the device.

This dissertation represents a significant contribution to the development of diagnostics for blood stream infections. While I designed and analyzed most of the experiments, there were several contributions to this project made by others. Dr. William Pitt recognized the need to analyze the stability of the spinning disk and thought that the instability seemed like a Kelvin-Helmholtz instability which guided the mathematical analysis. He also helped design the experimental stability analysis for the disk, helped design the clotting experiments, and helped guide the direction of the DNA identification simulations. Dr. Jared Whitehead contributed to the analytical solution of the Navier Stokes equation as well as analyzing the linear stability equations using python. These contributions were greatly appreciated and most helpful in accomplishing this project.
REFERENCES


[66] , 1979., Dec 19, 1977. 9


for detection of escherichia coli in produce and blood.” *Int J Nanomedicine*, 7, pp. 2591–600. 13, 15, 16


189


192


193


195


Kim, J., and Gale, B. K., 2008. “Quantitative and qualitative analysis of a microfluidic dna extraction system using a nanoporous alo(x) membrane.” Lab Chip, 8(9), pp. 1516–23. 27


206


207


[414] , 1982. 110


ph during plasmid preparation by alkaline lysis of escherichia coli.” Anal Biochem, 378(2),
pp. 224–5. 134, 143


the preparation of bacterial dna that employs dimethyl sulfoxide to induce the lysis of cells.”

ple method for the isolation of chromosomal dna from gram-positive or acid-fast bacteria.”

isolation and purification of total community dna from soil.” Journal of Microbiological
Methods, 39(1), pp. 1–16. 136, 144


positive bacteria.” Trends Genet, 11(6), pp. 217–8. 141

superinfected immune bacterium.” Journal of Molecular Biology, 14(2), pp. 399–+. 143

detergent.” Biochim Biophys Acta, 149(2), pp. 476–88. 143

terium. 1. the isolation of a nuclear material.” Biochem J, 88, pp. 176–82. 143

[430] Samant, S., Lee, H., Ghassemi, M., Chen, J., Cook, J. L., Mankin, A. S., and Neyfakh,
PLoS pathogens, 4(2). 143

[431] Pulleyblank, D., Michalak, M., Oaisley, S. L., and Glick, R., 1983. “A method for the purifi-
cation of e. coli plasmid dna by homogeneous lysis and polyethylene glycol precipitation.”
Mol Biol Rep, 9(3), pp. 191–5. 143, 144

Royal Society Interface, 5, pp. S131–S138. 143

[433] Smith, M. G., 1967. [69] isolation of high molecular weight DNA from normal and phage-
infected escherichia coli., Vol. 12 Elsevier, pp. 545–550. 144


APPENDIX A. STATISTICAL DATA

A.1 Chapter 5 Statistics

Table A.1: P-values for the 3000 rpm clotting spins.

<table>
<thead>
<tr>
<th>Category</th>
<th>Grouping</th>
<th>Evaluating</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>water + PBS</td>
<td>69s</td>
<td>H v F</td>
<td>0.108</td>
</tr>
<tr>
<td>water + PBS</td>
<td>117 s</td>
<td>H v F</td>
<td>0.21</td>
</tr>
<tr>
<td>water + PBS</td>
<td>H</td>
<td>69 v 117</td>
<td>0.916</td>
</tr>
<tr>
<td>water + PBS</td>
<td>F</td>
<td>69 v 117</td>
<td>0.719</td>
</tr>
<tr>
<td>water</td>
<td>69s</td>
<td>H v F</td>
<td>0.35</td>
</tr>
<tr>
<td>water</td>
<td>117 s</td>
<td>H v F</td>
<td>0.922</td>
</tr>
<tr>
<td>water</td>
<td>H</td>
<td>69 v 117</td>
<td>0.593</td>
</tr>
<tr>
<td>water</td>
<td>F</td>
<td>69 v 117</td>
<td>0.588</td>
</tr>
<tr>
<td>PBS</td>
<td>69s</td>
<td>H v F</td>
<td>0.232</td>
</tr>
<tr>
<td>PBS</td>
<td>117 s</td>
<td>H v F</td>
<td>0.043</td>
</tr>
<tr>
<td>PBS</td>
<td>H</td>
<td>69 v 117</td>
<td>0.705</td>
</tr>
<tr>
<td>PBS</td>
<td>F</td>
<td>69 v 117</td>
<td>0.82</td>
</tr>
<tr>
<td>69s</td>
<td>H</td>
<td>W v P</td>
<td>0.86</td>
</tr>
<tr>
<td>69s</td>
<td>F</td>
<td>W v P</td>
<td>0.828</td>
</tr>
<tr>
<td>117s</td>
<td>H</td>
<td>W v P</td>
<td>0.281</td>
</tr>
<tr>
<td>117s</td>
<td>F</td>
<td>W v P</td>
<td>0.322</td>
</tr>
</tbody>
</table>
Table A.2: P-values for the clotting spins at a separation speed of 4000 rpm with different clotting speeds.

<table>
<thead>
<tr>
<th>Category</th>
<th>Evaluating</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>water + PBS</td>
<td>500 v 1000</td>
<td>0.113</td>
</tr>
<tr>
<td>water + PBS</td>
<td>500 v 1500</td>
<td>0.093</td>
</tr>
<tr>
<td>water + PBS</td>
<td>500 v 2000</td>
<td>1.63e-6</td>
</tr>
<tr>
<td>water + PBS</td>
<td>1000 v 1500</td>
<td>0.015</td>
</tr>
<tr>
<td>water + PBS</td>
<td>1000 v 2000</td>
<td>1.19e-8</td>
</tr>
<tr>
<td>water + PBS</td>
<td>1500 v 2000</td>
<td>0.142</td>
</tr>
<tr>
<td>PBS</td>
<td>500 v 1000</td>
<td>0.053</td>
</tr>
<tr>
<td>PBS</td>
<td>500 v 1500</td>
<td>0.052</td>
</tr>
<tr>
<td>PBS</td>
<td>500 v 2000</td>
<td>0.004</td>
</tr>
<tr>
<td>PBS</td>
<td>1000 v 1500</td>
<td>0.001</td>
</tr>
<tr>
<td>PBS</td>
<td>1000 v 2000</td>
<td>8.02e-8</td>
</tr>
<tr>
<td>PBS</td>
<td>1500 v 2000</td>
<td>0.251</td>
</tr>
<tr>
<td>water</td>
<td>500 v 1000</td>
<td>0.418</td>
</tr>
<tr>
<td>water</td>
<td>500 v 1500</td>
<td>0.468</td>
</tr>
<tr>
<td>water</td>
<td>500 v 2000</td>
<td>0.001</td>
</tr>
<tr>
<td>water</td>
<td>1000 v 1500</td>
<td>0.729</td>
</tr>
<tr>
<td>water</td>
<td>1000 v 2000</td>
<td>0.014</td>
</tr>
<tr>
<td>water</td>
<td>1500 v 2000</td>
<td>0.299</td>
</tr>
<tr>
<td>500</td>
<td>P v W</td>
<td>0.316</td>
</tr>
<tr>
<td>1000</td>
<td>P v W</td>
<td>0.077</td>
</tr>
<tr>
<td>1500</td>
<td>P v W</td>
<td>0.376</td>
</tr>
<tr>
<td>2000</td>
<td>P v W</td>
<td>0.253</td>
</tr>
</tbody>
</table>
Table A.3: P-values for the clotting spins at a separation speed of 4000 rpm and a clotting speed of 1000 rpm with different fluid layouts on the disk.

<table>
<thead>
<tr>
<th>Evaluating</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 D v 2 H</td>
<td>0.788</td>
</tr>
<tr>
<td>4 D v F 0</td>
<td>0.0001</td>
</tr>
<tr>
<td>4 D v F 3</td>
<td>0.461</td>
</tr>
<tr>
<td>4 D v F 6</td>
<td>0.588</td>
</tr>
<tr>
<td>4 D v F 12</td>
<td>0.244</td>
</tr>
<tr>
<td>2 H v F 0</td>
<td>0.003</td>
</tr>
<tr>
<td>2 H v F 3</td>
<td>0.734</td>
</tr>
<tr>
<td>2 H v F 6</td>
<td>0.489</td>
</tr>
<tr>
<td>2 H v F 12</td>
<td>0.231</td>
</tr>
<tr>
<td>F 0 v F 3</td>
<td>0.001</td>
</tr>
<tr>
<td>F 0 v F 6</td>
<td>0.001</td>
</tr>
<tr>
<td>F 0 v F 12</td>
<td>0.0001</td>
</tr>
<tr>
<td>F 3 v F 6</td>
<td>0.267</td>
</tr>
<tr>
<td>F 3 v F 12</td>
<td>0.079</td>
</tr>
<tr>
<td>F 6 v F 12</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table A.4: P-values for the clotting spins at a separation speed of 4000 rpm and a clotting speed of 1000 rpm with different volumes of ADP (2g/L) added to the blood.

<table>
<thead>
<tr>
<th>Evaluating</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 v 0.25</td>
<td>0.255</td>
</tr>
<tr>
<td>0 v 0.5</td>
<td>4.36e-7</td>
</tr>
<tr>
<td>0 v 1</td>
<td>6.0e-7</td>
</tr>
<tr>
<td>0.25 v 0.5</td>
<td>1.97e-5</td>
</tr>
<tr>
<td>0.25 v 1</td>
<td>8.76e-5</td>
</tr>
<tr>
<td>0.5 v 1</td>
<td>0.139</td>
</tr>
</tbody>
</table>
Table A.5: P-values for the clotting spins with different disk designs at different separation speeds with a clotting speed of 1000 rpm.

<table>
<thead>
<tr>
<th>Category</th>
<th>Evaluating</th>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>4000 v 6000</td>
<td>0.513</td>
</tr>
<tr>
<td>Normal</td>
<td>4000 v 7000</td>
<td>6.20e-3</td>
</tr>
<tr>
<td>Normal</td>
<td>4000 v 8000</td>
<td>0.241</td>
</tr>
<tr>
<td>Normal</td>
<td>6000 v 7000</td>
<td>0.026</td>
</tr>
<tr>
<td>Normal</td>
<td>6000 v 8000</td>
<td>0.194</td>
</tr>
<tr>
<td>Normal</td>
<td>7000 v 8000</td>
<td>0.593</td>
</tr>
<tr>
<td>Pointed</td>
<td>4000 v 6000</td>
<td>0.141</td>
</tr>
<tr>
<td>Pointed</td>
<td>4000 v 7000</td>
<td>0.699</td>
</tr>
<tr>
<td>Pointed</td>
<td>4000 v 8000</td>
<td>0.844</td>
</tr>
<tr>
<td>Pointed</td>
<td>6000 v 7000</td>
<td>0.034</td>
</tr>
<tr>
<td>Pointed</td>
<td>6000 v 8000</td>
<td>0.068</td>
</tr>
<tr>
<td>Pointed</td>
<td>7000 v 8000</td>
<td>0.829</td>
</tr>
<tr>
<td>Segmented</td>
<td>4000 v 6000</td>
<td>0.155</td>
</tr>
<tr>
<td>Segmented</td>
<td>4000 v 7000</td>
<td>0.025</td>
</tr>
<tr>
<td>Segmented</td>
<td>4000 v 8000</td>
<td>0.089</td>
</tr>
<tr>
<td>Segmented</td>
<td>6000 v 7000</td>
<td>0.961</td>
</tr>
<tr>
<td>Segmented</td>
<td>6000 v 8000</td>
<td>0.994</td>
</tr>
<tr>
<td>Segmented</td>
<td>7000 v 8000</td>
<td>0.945</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>4000 v 6000</td>
<td>0.168</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>4000 v 7000</td>
<td>0.356</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>4000 v 8000</td>
<td>0.318</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>6000 v 7000</td>
<td>0.419</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>6000 v 8000</td>
<td>0.624</td>
</tr>
<tr>
<td>Category</td>
<td>Parameters</td>
<td>P-values</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>32-5mm Triangles</td>
<td>7000 v 8000</td>
<td>0.820</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>4000 v 6000</td>
<td>0.448</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>4000 v 7000</td>
<td>0.612</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>4000 v 8000</td>
<td>0.994</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>6000 v 7000</td>
<td>0.689</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>6000 v 8000</td>
<td>0.147</td>
</tr>
<tr>
<td>16-5mm Triangles</td>
<td>7000 v 8000</td>
<td>0.467</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>4000 v 6000</td>
<td>0.692</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>4000 v 7000</td>
<td>0.715</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>4000 v 8000</td>
<td>0.677</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>6000 v 7000</td>
<td>0.563</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>6000 v 8000</td>
<td>0.961</td>
</tr>
<tr>
<td>32-2mm Triangles</td>
<td>7000 v 8000</td>
<td>0.558</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>4000 v 6000</td>
<td>0.327</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>4000 v 7000</td>
<td>0.180</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>4000 v 8000</td>
<td>0.798</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>6000 v 7000</td>
<td>0.792</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>6000 v 8000</td>
<td>0.307</td>
</tr>
<tr>
<td>16-2mm Triangles</td>
<td>7000 v 8000</td>
<td>0.197</td>
</tr>
<tr>
<td>4000 Normal v Pointed</td>
<td></td>
<td>0.031</td>
</tr>
<tr>
<td>4000 Normal v Segmented</td>
<td></td>
<td>3.48e-3</td>
</tr>
<tr>
<td>4000 Normal v 32-5mm Triangles</td>
<td></td>
<td>0.119</td>
</tr>
<tr>
<td>4000 Normal v 16-5mm Triangles</td>
<td></td>
<td>0.372</td>
</tr>
<tr>
<td>4000 Normal v 32-2mm Triangles</td>
<td></td>
<td>0.894</td>
</tr>
<tr>
<td>4000 Normal v 16-2mm Triangles</td>
<td></td>
<td>0.053</td>
</tr>
<tr>
<td>Category</td>
<td>Parameters</td>
<td>P-values</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v Pointed</td>
<td>0.038</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v Segmented</td>
<td>0.177</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v 32-5mm Triangles</td>
<td>0.138</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v 16-5mm Triangles</td>
<td>3.86e-3</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v 32-2mm Triangles</td>
<td>0.253</td>
</tr>
<tr>
<td>6000</td>
<td>Normal v 16-2mm Triangles</td>
<td>0.020</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v Pointed</td>
<td>0.012</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v Segmented</td>
<td>0.904</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v 32-5mm Triangles</td>
<td>0.641</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v 16-5mm Triangles</td>
<td>0.278</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v 32-2mm Triangles</td>
<td>0.617</td>
</tr>
<tr>
<td>7000</td>
<td>Normal v 16-2mm Triangles</td>
<td>0.070</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v Pointed</td>
<td>0.365</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v Segmented</td>
<td>0.679</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v 32-5mm Triangles</td>
<td>0.709</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v 16-5mm Triangles</td>
<td>0.810</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v 32-2mm Triangles</td>
<td>0.920</td>
</tr>
<tr>
<td>8000</td>
<td>Normal v 16-2mm Triangles</td>
<td>0.792</td>
</tr>
</tbody>
</table>

Table A.6: P-values for the SDS bactericidal experiment.

<table>
<thead>
<tr>
<th>Category</th>
<th>Evaluating</th>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0% SDS</td>
<td>10 min v 1 hr</td>
<td>0.824</td>
</tr>
<tr>
<td>1/243% SDS</td>
<td>10 min v 1 hr</td>
<td>0.482</td>
</tr>
<tr>
<td>1/81% SDS</td>
<td>10 min v 1 hr</td>
<td>0.916</td>
</tr>
<tr>
<td>1/27% SDS</td>
<td>10 min v 1 hr</td>
<td>0.962</td>
</tr>
<tr>
<td>1/9% SDS</td>
<td>10 min v 1 hr</td>
<td>0.573</td>
</tr>
</tbody>
</table>

220
<table>
<thead>
<tr>
<th>Category</th>
<th>Parameters</th>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3% SDS</td>
<td>10 min v 1 hr</td>
<td>0.533</td>
</tr>
<tr>
<td>1% SDS</td>
<td>10 min v 1 hr</td>
<td>0.505</td>
</tr>
<tr>
<td>3% SDS</td>
<td>10 min v 1 hr</td>
<td>0.453</td>
</tr>
<tr>
<td>9% SDS</td>
<td>10 min v 1 hr</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1/243% SDS</td>
<td>0.353</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1/81% SDS</td>
<td>0.797</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1/27% SDS</td>
<td>0.198</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1/9% SDS</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1/3% SDS</td>
<td>0.022</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 1% SDS</td>
<td>0.019</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 3% SDS</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>10 min 0% SDS v 9% SDS</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 1/81% SDS</td>
<td>0.185</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 1/27% SDS</td>
<td>0.273</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 1/9% SDS</td>
<td>0.026</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 1/3% SDS</td>
<td>0.036</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 1% SDS</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 3% SDS</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>10 min 1/243% SDS v 9% SDS</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 1/27% SDS</td>
<td>0.19</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 1/9% SDS</td>
<td>0.048</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 1/3% SDS</td>
<td>0.008</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 1% SDS</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 3% SDS</td>
<td>0.012</td>
</tr>
<tr>
<td></td>
<td>10 min 1/81% SDS v 9% SDS</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>10 min 1/27% SDS v 1/9% SDS</td>
<td>0.288</td>
</tr>
<tr>
<td></td>
<td>10 min 1/27% SDS v 1/3% SDS</td>
<td>0.196</td>
</tr>
<tr>
<td></td>
<td>10 min 1/27% SDS v 1% SDS</td>
<td>0.175</td>
</tr>
<tr>
<td>Category</td>
<td>Parameters</td>
<td>P-values</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------</td>
<td>----------</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/27%$ SDS v $3%$ SDS</td>
<td>0.185</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/27%$ SDS v $9%$ SDS</td>
<td>0.167</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/9%$ SDS v $1/3%$ SDS</td>
<td>0.6</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/9%$ SDS v $1%$ SDS</td>
<td>0.469</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/9%$ SDS v $3%$ SDS</td>
<td>0.532</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/9%$ SDS v $9%$ SDS</td>
<td>0.418</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/3%$ SDS v $1%$ SDS</td>
<td>0.65</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/3%$ SDS v $3%$ SDS</td>
<td>0.84</td>
</tr>
<tr>
<td>10 min</td>
<td>$1/3%$ SDS v $9%$ SDS</td>
<td>0.485</td>
</tr>
<tr>
<td>10 min</td>
<td>$1%$ SDS v $3%$ SDS</td>
<td>0.76</td>
</tr>
<tr>
<td>10 min</td>
<td>$1%$ SDS v $9%$ SDS</td>
<td>0.589</td>
</tr>
<tr>
<td>10 min</td>
<td>$3%$ SDS v $9%$ SDS</td>
<td>0.512</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1/243%$ SDS</td>
<td>0.448</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1/81%$ SDS</td>
<td>0.37</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1/27%$ SDS</td>
<td>0.134</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1/9%$ SDS</td>
<td>0.049</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1/3%$ SDS</td>
<td>0.047</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $1%$ SDS</td>
<td>0.046</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $3%$ SDS</td>
<td>0.046</td>
</tr>
<tr>
<td>1 hr</td>
<td>$0%$ SDS v $9%$ SDS</td>
<td>0.046</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $1/81%$ SDS</td>
<td>0.319</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $1/27%$ SDS</td>
<td>0.117</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $1/9%$ SDS</td>
<td>0.045</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $1/3%$ SDS</td>
<td>0.042</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $1%$ SDS</td>
<td>0.042</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $3%$ SDS</td>
<td>0.042</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/243%$ SDS v $9%$ SDS</td>
<td>0.042</td>
</tr>
<tr>
<td>1 hr</td>
<td>$1/81%$ SDS v $1/27%$ SDS</td>
<td>0.124</td>
</tr>
<tr>
<td>Category</td>
<td>Parameters</td>
<td>P-values</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/81% SDS v 1/9% SDS</td>
<td>0.041</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/81% SDS v 1/3% SDS</td>
<td>0.038</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/81% SDS v 1% SDS</td>
<td>0.037</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/81% SDS v 3% SDS</td>
<td>0.037</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/81% SDS v 9% SDS</td>
<td>0.037</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/27% SDS v 1/9% SDS</td>
<td>0.031</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/27% SDS v 1/3% SDS</td>
<td>0.021</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/27% SDS v 1% SDS</td>
<td>0.02</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/27% SDS v 3% SDS</td>
<td>0.02</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/27% SDS v 9% SDS</td>
<td>0.02</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/9% SDS v 1/3% SDS</td>
<td>0.521</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/9% SDS v 1% SDS</td>
<td>0.428</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/9% SDS v 3% SDS</td>
<td>0.418</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/9% SDS v 9% SDS</td>
<td>0.392</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/3% SDS v 1% SDS</td>
<td>0.582</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/3% SDS v 3% SDS</td>
<td>0.531</td>
</tr>
<tr>
<td>1 hr</td>
<td>1/3% SDS v 9% SDS</td>
<td>0.414</td>
</tr>
<tr>
<td>1 hr</td>
<td>1% SDS v 3% SDS</td>
<td>0.883</td>
</tr>
<tr>
<td>1 hr</td>
<td>1% SDS v 9% SDS</td>
<td>0.504</td>
</tr>
<tr>
<td>1 hr</td>
<td>3% SDS v 9% SDS</td>
<td>0.516</td>
</tr>
</tbody>
</table>
### A.2 Chapter 7 Statistics

Table A.7: P-values for the experiments evaluating different lysozyme concentrations.

<table>
<thead>
<tr>
<th>Evaluating</th>
<th>EC P-values</th>
<th>KP P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 mg/mL Lysozyme v 5 mg/mL Lysozyme</td>
<td>0.071</td>
<td>0.457</td>
</tr>
<tr>
<td>10 mg/mL Lysozyme v 1 mg/mL Lysozyme</td>
<td>0.080</td>
<td>0.341</td>
</tr>
<tr>
<td>10 mg/mL Lysozyme v 0.5 mg/mL Lysozyme</td>
<td>0.081</td>
<td>0.248</td>
</tr>
<tr>
<td>10 mg/mL Lysozyme v 0.1 mg/mL Lysozyme</td>
<td>0.446</td>
<td>0.669</td>
</tr>
<tr>
<td>10 mg/mL Lysozyme v 0.05 mg/mL Lysozyme</td>
<td>0.034</td>
<td>0.375</td>
</tr>
<tr>
<td>10 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.682</td>
<td>0.970</td>
</tr>
<tr>
<td>5 mg/mL Lysozyme v 1 mg/mL Lysozyme</td>
<td>0.928</td>
<td>0.893</td>
</tr>
<tr>
<td>5 mg/mL Lysozyme v 0.5 mg/mL Lysozyme</td>
<td>0.360</td>
<td>0.765</td>
</tr>
<tr>
<td>5 mg/mL Lysozyme v 0.1 mg/mL Lysozyme</td>
<td>0.265</td>
<td>0.608</td>
</tr>
<tr>
<td>5 mg/mL Lysozyme v 0.05 mg/mL Lysozyme</td>
<td>0.010</td>
<td>0.092</td>
</tr>
<tr>
<td>5 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.193</td>
<td>0.492</td>
</tr>
<tr>
<td>1 mg/mL Lysozyme v 0.5 mg/mL Lysozyme</td>
<td>0.353</td>
<td>0.844</td>
</tr>
<tr>
<td>1 mg/mL Lysozyme v 0.1 mg/mL Lysozyme</td>
<td>0.260</td>
<td>0.423</td>
</tr>
<tr>
<td>1 mg/mL Lysozyme v 0.05 mg/mL Lysozyme</td>
<td>0.013</td>
<td>0.031</td>
</tr>
<tr>
<td>1 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.187</td>
<td>0.396</td>
</tr>
<tr>
<td>0.5 mg/mL Lysozyme v 0.1 mg/mL Lysozyme</td>
<td>0.561</td>
<td>0.250</td>
</tr>
<tr>
<td>0.5 mg/mL Lysozyme v 0.05 mg/mL Lysozyme</td>
<td>4.74e-3</td>
<td>7.17e-3</td>
</tr>
<tr>
<td>0.5 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.304</td>
<td>0.316</td>
</tr>
<tr>
<td>0.1 mg/mL Lysozyme v 0.05 mg/mL Lysozyme</td>
<td>0.039</td>
<td>0.060</td>
</tr>
<tr>
<td>0.1 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.463</td>
<td>0.692</td>
</tr>
<tr>
<td>0.05 mg/mL Lysozyme v 0 mg/mL Lysozyme</td>
<td>0.545</td>
<td>0.500</td>
</tr>
</tbody>
</table>
APPENDIX B. MATHEMATICAL ANALYSIS

B.1 Chapter 6 Mathematical Details

B.1.1 Derivation of 2-D Velocity Profile during Spin Up phase

B.1.1.1 Problem set up

We seek a solution to (6.5) with the “no-slip” and free-surface boundary conditions. The fluid begins at rest and is accelerated at a constant velocity until it reaches the hold velocity and begins the hold phase. The equation of motion Eq.(B.1), along with the boundary conditions Eqs.(B.3) and (B.4) and initial condition Eq.(B.2), are solved using Newtonian mechanics. The solution of (B.32) is then the starting condition for the hold phase.

The equation of motion, boundary conditions, and initial condition for the fluid is:

\[ \frac{\partial v}{\partial t} = \nu \frac{\partial^2 v}{\partial y^2}, \quad 0 < y < d, \]  
\[ v(t = 0) = 0 \]  
\[ v(0) = kt \]  
\[ v'(d) = 0. \]

We will be using the following parameters for single layer flow to non-dimensionalize the equations as used,

\[ \kappa = \frac{v}{\sqrt{kd}}, \quad \eta = \frac{y}{d}, \quad \varepsilon = \frac{vt}{d^2}, \quad Re = \frac{k^\frac{1}{2}d^\frac{3}{2}}{\nu}. \]
The non-dimensional equations for (B.1)-(B.4) are:

\[
\frac{\partial \kappa}{\partial \varepsilon} = \frac{\partial^2 \kappa}{\partial \eta^2}, \quad 0 < \eta < 1, \tag{B.5}
\]

\[\kappa(\varepsilon = 0) = 0 \tag{B.6}\]

\[\kappa(0) = \varepsilon Re \tag{B.7}\]

\[\kappa'(1) = 0. \tag{B.8}\]

In order to accommodate the non-homogeneous boundary condition, we will let \(\kappa\) be given by the following:

\[\kappa = w + \theta \tag{B.9}\]

where \(w\) accounts for the non-homogeneous part of (B.3). The simplest solution to the boundary conditions for \(w\) is a linear relationship between them, \(w = c_0 + c_1 \eta\). Applying the boundary conditions, we get:

\[w = c_0 + c_1 \eta \tag{B.10}\]

\[w'(1) = c_1 = 0 \tag{B.11}\]

\[w(0) = c_0 = \varepsilon Re \tag{B.12}\]

\[w = \varepsilon Re \tag{B.13}\]

We then split \(\kappa\) into its homogeneous, \(\theta\), and non-homogeneous, \(w\), components to obtain:

\[
\frac{\partial \theta}{\partial \varepsilon} = \frac{\partial^2 \theta}{\partial \eta^2} - \frac{\partial w}{\partial \varepsilon}, \quad 0 < \eta < 1, \tag{B.14}
\]

\[\theta(\varepsilon = 0) = 0 \tag{B.15}\]

\[\theta(0) = 0 \tag{B.16}\]

\[\theta'(1) = 0. \tag{B.17}\]

We first solve the eigenvalue problem associated with the homogeneous form of (B.14), and then use that information to solve the non-homogeneous case which includes the term \(\frac{\partial w}{\partial \varepsilon} = Re\) as well as the temporal derivative.
B.1.1.2 Eigenfunction expansion technique

We are interested in solving the eigenvalue problem given by:

\[ \partial_{\eta\eta} f = -\lambda^2 f, \]  

where \( f(\eta) \) is the eigenfunction with homogeneous boundary conditions. Once we have determined \( f(\eta) \), we note that the full solution of (B.14) can be written as an expansion of these functions, i.e.:

\[ \theta(\varepsilon, \eta) = \sum_{m} g_m(\varepsilon) f_m(\eta), \]  

which will reduce (B.14) to

\[ \sum_{m} g_m'(\varepsilon) f_m(\eta) = -\sum_{m} \lambda^2 g_m(\varepsilon) f_m(\eta) - Re. \]  

The solution to (B.18) has the form:

\[ f = A_m \sin (\lambda \eta) + B_m \cos (\lambda \eta). \]  

Applying \( f(0) = 0 \) and \( f'(1) = 0 \) leaves us with:

\[ f = A_m \sin (\lambda \eta) \]  

\[ \lambda = \frac{m\pi}{2} \]  

where \( m \) is odd.

The constants \( A_m \) are chosen to normalize the eigenfunction so that:

\[ \int_{0}^{1} f_m^2(\eta) d\eta = 1. \]  

Solving this equation yields:

\[ A_m = \sqrt{2} \]
B.1.1.3 Determining the temporal coefficients

Converting $Re$ with a Fourier series into an infinite sum, we obtain:

$$Re = \sum_{m}^{\text{odd}} D_m A_m \sin \left( \frac{m \pi}{2} \eta \right) \quad (B.27)$$

$$D_m = \int_{0}^{1} Re \sin \left( \frac{m \pi}{2} \eta \right) d\eta \quad (B.28)$$

$$= \frac{2Re}{m \pi} \quad (B.29)$$

Now, returning to the temporal coefficients, we require $g_m$ to satisfy the differential equation

$$\dot{\varepsilon} g_m = -\lambda^2 g_m - \frac{2Re}{m \pi} \quad (B.30)$$

which has the solution given by

$$g_m = \frac{2Re}{m \pi \lambda^2} \left( 1 - \exp^{-\lambda^2 \varepsilon} \right) \quad (B.31)$$

Putting all of the pieces together, we arrive at the solution to (B.5)

$$\kappa = \varepsilon Re - \sum_{m}^{\text{odd}} \frac{8 \sqrt{2} Re}{m^3 \pi^3} \left( 1 - \exp^{-\frac{m^2 \pi^2}{4} \varepsilon} \right) \sin \left( \frac{m \pi}{2} \eta \right) \quad (B.32)$$

B.1.2 Derivation of 2-D Velocity Profile during Hold phase

B.1.2.1 Problem set up

As stated above, this problem commences in time where the spin up velocity profile ends (6 s for a constant acceleration rate of 500 rpms/s until 3000 rpm). The above velocity profile, Eq.(B.32), defines the shape of the starting velocity profile for the hold phase. Thus $\kappa$ is now at a set value of $\varepsilon$, which we will designate as $T_\varepsilon$ to denote that it is constant in time but changing in $\eta$. This phase continues for 60 seconds at which the disk is then decelerated, beginning the deceleration phase. We again seek a solution to (6.5) with the “no-slip” and free-surface boundary conditions. This time the fluid begins according Eq.(B.32) with $T_\varepsilon$ substituted for $\varepsilon$. The equation
of motion Eq.(B.33), along with the boundary conditions Eqs.(B.35) and (B.36) and initial condition Eq.(B.34), is solved using Newtonian mechanics. The solution of (B.33) is then the starting condition for the deceleration phase.

We are now solving the equations of motion for the fluid being held at a constant velocity, \( \zeta \), for cell separation. Continuing with the same non-dimensional parameters as above, we now have:

\[
\frac{\partial \omega}{\partial \varepsilon} = \frac{\partial^2 \omega}{\partial \eta^2}, \quad 0 < \eta < 1, \quad (B.33)
\]

\[
\omega(\varepsilon = 0) = \kappa(\varepsilon = T_\varepsilon, \eta) \quad (B.34)
\]

\[
\omega(0) = \zeta \quad (B.35)
\]

\[
\omega'(1) = 0. \quad (B.36)
\]

In order to accommodate the non-homogeneous initial condition and the constant velocity boundary condition, we will let \( \omega \) be given by the following:

\[
\omega = \theta + \phi \quad (B.37)
\]

where \( \theta \) accounts for of the non-homogeneous initial condition, and \( \phi \) accounts for the constant velocity boundary condition. The equations for \( \theta \) are:

\[
\frac{\partial \theta}{\partial \varepsilon} = \frac{\partial^2 \theta}{\partial \eta^2}, \quad 0 < \eta < 1, \quad (B.38)
\]

\[
\theta(\varepsilon = 0) = \kappa(\varepsilon = T_\varepsilon, \eta) \quad (B.39)
\]

\[
\theta(0) = 0 \quad (B.40)
\]

\[
\theta'(1) = 0. \quad (B.41)
\]
The equations for $\phi$ are:

$$\frac{\partial \phi}{\partial \varepsilon} = \frac{\partial^2 \phi}{\partial \eta^2}, \quad 0 < \eta < 1,$$ (B.42)

$$\phi(\varepsilon = 0) = 0$$ (B.43)

$$\phi(0) = \zeta$$ (B.44)

$$\phi'(1) = 0.$$ (B.45)

In order to solve (B.38) and (B.42), we will first solve the eigenvalue problem associated with the homogeneous form of each equation and then use that information to solve the non-homogeneous case and temporal derivative.

**B.1.2.2 Eigenfunction expansion technique**

We are then interested in solving the eigenvalue problems given by:

$$\partial_{\eta \eta} s = -\lambda^2 s$$ (B.46)

$$\partial_{\eta \eta} q = -\lambda^2 q$$ (B.47)

where $s(\eta)$ is the eigenfunction for $\theta$ and $q(\eta)$ is the eigenfunction for $\phi$.

Once we have determined $s(\eta)$ and $q(\eta)$, we note that the full solutions to (B.38) and (B.42) can be written as an expansion of these functions, i.e.:

$$\theta(\varepsilon, \eta) = \sum_m c_m(\varepsilon) s_m(\eta)$$ (B.48)

$$\phi(\varepsilon, \eta) = \sum_m d_m(\varepsilon) q_m(\eta)$$ (B.49)

which will reduce (B.38) and (B.42) to

$$\sum_m c'_m(\varepsilon) s_m(\eta) = -\sum_m \lambda^2 c_m(\varepsilon) s_m(\eta)$$ (B.50)

$$\sum_m d'_m(\varepsilon) q_m(\eta) = -\sum_m \lambda^2 d_m(\varepsilon) q_m(\eta).$$ (B.51)
The solutions to the eigenvalue problems (B.46) and (B.47) have the form:

\[ s = A \sin (\lambda \eta) + B \cos (\lambda \eta) \]  \hspace{1cm} (B.52)

\[ q = E \sin (\lambda \eta) + F \cos (\lambda \eta) \]  \hspace{1cm} (B.53)

Applying the boundary conditions, \( s(0) = 0, s'(1) = 0, q(0) = 0, \) and \( q'(1) = 0, \) leaves us with:

\[ s = A \sin (\lambda \eta) \]  \hspace{1cm} (B.54)

\[ q = E \sin (\lambda \eta) \]  \hspace{1cm} (B.55)

\[ \lambda = \frac{m \pi}{2} \]  \hspace{1cm} (B.56)

where \( m \) is odd.

Due to the non-homogeneous conditions of both \( s \) and \( q, \) we will solve for \( A \) and \( E \) below.

**B.1.2.3 Determining the temporal coefficients**

Now, returning to the temporal coefficients, we want \( c_m \) and \( d_m \) to satisfy the differential equations

\[ c'_m(\varepsilon) = -\lambda^2 c_m(\varepsilon) \]  \hspace{1cm} (B.57)

\[ d'_m(\varepsilon) = -\lambda^2 d_m(\varepsilon) \]  \hspace{1cm} (B.58)

which have the solutions given by

\[ c_m(\varepsilon) = Ge^{-\lambda^2 \varepsilon} \]  \hspace{1cm} (B.59)

\[ d_m(\varepsilon) = He^{-\lambda^2 \varepsilon} \]  \hspace{1cm} (B.60)
Plugging in the solutions from (B.54), (B.55), (B.59), (B.60) and (B.56) into (B.48) and (B.49), we obtain:

\[
\theta = \sum_{m}^{\text{odd}} J_m \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right) \sin\left(\frac{m \pi}{2} \eta\right) \quad \text{(B.61)}
\]

\[
\phi = \sum_{m}^{\text{odd}} K_m \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right) \sin\left(\frac{m \pi}{2} \eta\right) \quad \text{(B.62)}
\]

where \(J_m = B \cdot G\) and \(K_m = E \cdot H\).

Now addressing the non-homogeneous part for each equation, for \(\theta\) we have \(\theta(\epsilon = 0) = \kappa(\epsilon = T_\epsilon, \eta)\) and for \(\phi\) we have \(\phi(\eta = 0) = \zeta\). Solving for \(J_m\), we get:

\[
\kappa(\eta) = J_m \sin\left(\frac{n \pi}{2} \eta\right) \quad \text{(B.63)}
\]

\[
T_\epsilon Re - \sum_{m}^{\text{odd}} \frac{8 \sqrt{2} Re}{m^3 \pi^3} \left(1 - \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right)\right) \sin\left(\frac{m \pi}{2} \eta\right) = J_m \sin\left(\frac{m \pi}{2} \eta\right) \quad \text{(B.64)}
\]

\[
J_m \int_{0}^{1} \left(T_\epsilon Re - \sum_{m}^{\text{odd}} \frac{8 \sqrt{2} Re}{m^3 \pi^3} \left(1 - \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right)\right) \sin\left(\frac{m \pi}{2} \eta\right)\right) \sin\left(\frac{n \pi}{2} \eta\right) d\eta = J_m^2 \int_{0}^{1} \sin\left(\frac{m \pi}{2} \eta\right) \sin\left(\frac{n \pi}{2} \eta\right) d\eta \quad \text{(B.65)}
\]

\[
J_m = \frac{4T_\epsilon Re}{m \pi} - \frac{8 \sqrt{2} Re}{m^3 \pi^3} \left(1 - \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right)\right) \quad \text{(B.66)}
\]

which yields

\[
\theta = \sum_{m}^{\text{odd}} \left[ \frac{4T_\epsilon Re}{m \pi} - \frac{8 \sqrt{2} Re}{m^3 \pi^3} \left(1 - \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right)\right) \right] \exp\left(-\frac{m^2 \pi^2 \epsilon}{4}\right) \sin\left(\frac{m \pi}{2} \eta\right) \quad \text{(B.67)}
\]

Because \(\zeta\) is a constant we can just add it at the end, but then our initial condition must equal \(-\zeta\). Solving for \(K_m\), then becomes:

\[
-\zeta = K_m \sin\left(\frac{m \pi}{2} \eta\right) \quad \text{(B.68)}
\]

\[
-\zeta K_m \int_{0}^{1} \sin\left(\frac{n \pi}{2} \eta\right) d\eta = K_m^2 \int_{0}^{1} \sin\left(\frac{n \pi}{2} \eta\right) \sin\left(\frac{m \pi}{2} \eta\right) d\eta \quad \text{(B.69)}
\]

\[
K_m = -\frac{4 \zeta}{m \pi} \quad \text{(B.70)}
\]

232
which yields

\[ \phi = \zeta - \sum_{m} \frac{\text{odd}}{m\pi} \exp \left( \frac{-m^2 \pi^2 \epsilon}{4} \right) \sin \left( \frac{m\pi}{2}\eta \right) \] (B.71)

Putting \( \theta \) and \( \phi \) together gives us the solution to (B.33).

\[ \omega = \zeta + \sum_{n} \frac{\text{odd}}{4T_{c} Re - 4\zeta} \left[ 4T_{c} Re - 4\zeta - \frac{8 \sqrt{2} Re}{m^2 \pi^2} \left( 1 - \exp \left( \frac{-m^2 \pi^2 \epsilon}{4} \right) \right) \right] \frac{1}{m\pi} \exp \left( \frac{-m^2 \pi^2 \epsilon}{4} \right) \sin \left( \frac{m\pi}{2}\eta \right) \] (B.72)

Converting this non-dimensional equation back into a dimensional equation, we get

\[ u = U_0 + \sum_{n} \frac{\text{odd}}{8 \sqrt{2} kd^2 \left( 1 - \exp \left( \frac{-n^2 \pi^2 \nu C}{4d^2} \right) \right) \left[ 4kt_{C} - 4U_0 \right]}{n^3 \pi^3 \nu} \left( \exp \left( \frac{-n^2 \pi^2 \nu C}{4d^2} \right) \right) \exp \left( \frac{-n^2 \pi^2 \nu C}{4d^2} \right) \sin \left( \frac{n\pi}{2d}\eta \right) \] (B.73)

**B.1.2.4 Application of the constant velocity hold derivation**

Now that we have an equation for \( \omega \), we will apply it to our system, using the dimensionalized version \( u \), Eq.(B.73). We know that we transition from the spin up profile to the hold profile when we reach the hold velocity, meaning that \( kt_{C} = U_0 \). This reduces \( u \) to:

\[ u = U_0 + \sum_{n} \frac{\text{odd}}{8 \sqrt{2} kd^2} \left( 1 - \exp \left( \frac{-n^2 \pi^2 \nu C}{4d^2} \right) \right) \exp \left( \frac{-n^2 \pi^2 \nu C}{4d^2} \right) \sin \left( \frac{n\pi}{2d}\eta \right) \] (B.74)

Plotting \( u(y) \) at \( t = 0 \) seconds, \( t = 10 \) seconds and \( t = 60 \) seconds shows the development of the profile over time (see Figure B.1). The velocity profile reached at \( t = 60 \) seconds is the initial velocity profile for the deceleration phase. The difference between the velocity at the air interface and the velocity at the wall is \( 2.13 \times 10^{-8} \) m/s, a negligible value compared to the velocity of 18.85 m/s. Thus, the velocity profile for \( \omega \) is essentially uniform at the end of the holding phase and the deceleration commences with a uniform velocity profile.
Figure B.1: Plot of the velocity profile using the solution to the constant velocity hold derivation. The \( y \)-axis is the distance from the disk wall, with \( y = 0 \) being the wall of the disk. The start of the constant velocity hold profile (\( t=0 \)) is the end of the spin up velocity profile. \(-\cdots-\ t = 0 \) seconds, \( \cdots\cdots\ t = 10 \) seconds, \( -\longrightarrow\ t = 60 \) seconds
APPENDIX C. DNA ANALYSIS

Rapidly identifying DNA sequences can be difficult, especially when dealing with very small quantities of DNA. Chapter 2 discussed many of the different techniques that are being used to identify DNA sequences. This appendix explores the feasibility of using a block optical sequencing (BOS) method employing surface-enhanced Raman spectroscopy (SERS) to obtain a spectrum of short DNA oligomers of length $k$, called $k$-mers [363, 364]. This genomic analysis technique that has some data compression and data loss, but compensates by very rapid analysis of very short reads of DNA-sufficiently short length and suitably fast analysis—that the species and resistance genes can be identified in about an hour. While a working device with many thousands of pyramid tips is still in development, such a process has been proposed and demonstrated for the identification of resistance genes in bacteria associated with bloodstream infections [363,364,439].

Because the Raman spectrum of each A, T, G and C base is known, the overall ATGC content of a single $k$-mer can be calculated by mathematical analysis of the $k$-mer spectrum. Sequence information is lost, but the base content-called block optical content (BOC)-is preserved. For example, the 10 bp DNA segment ATATGGCCTT would become a BOC datum of $A_2T_4G_2C_2$. For very rapid analysis, this BOC technique can be multiplexed by creating an array of thousands of pyramidal peaks on a silicon wafer whose entire peak field can be imaged by a sensitive CCD camera. Using band-pass filters at discrete spectral windows, optical intensity at specific wavelengths can be obtained simultaneously from all peaks. Finally the optical spectra are processed to obtain the ATGC content of the DNA on each pyramid peak. The size of the tips is such that 10 bases, but not any longer length, fit within the SERS electromagnetic field “hot-spot” [363]. It is estimated that using a 1,000 x 1,000 array of SERS pyramids on a silicon wafer, 1,000,000 reads of DNA 10-mers can be done in about 100 seconds, using high-throughput Raman spectroscopy with quantum dot optical filters [440] and digital processing of the resulting spectra.
Such a technique is ideally suited for genomic identification of bacteria, as a typical bacterial genome is about 5,000,000 bp, or 10 Mbase of single-stranded DNA (ssDNA). Clipping this genome into 10-mer lengths would provide enough ssDNA from a single bacterium to cover the SERS pyramids on a 1,000 x 1,000 array. Bloodstream infections contain very low counts of bacteria, often on the order of 10 colony forming units (CFU) per mL of blood. Thus a 10 mL sample of blood would provide 100-fold more DNA than needed to place a 10-mer on each pyramid. Bacteria (and their DNA) can be collected from blood in minutes [7, 26, 76], and the SERS analysis can commence immediately, followed by computations for identification of species and antibiotic resistance.

Compared to species identification, analysis of resistance genes on plasmids is more challenging since those genes are usually contained within 500 to 1,000 bases while the average plasmid is roughly 200,000 bases in length. Thus, there is a much smaller signal to background ratio making detection of resistance genes more difficult. Nevertheless, we show herein that our technique can still identify specific genes.

While such technology seems promising, this appendix only explores the feasibility of such a device through simulation. These simulations allow us to answer many of the theoretical questions, the largest of which is whether the loss of sequence data (the BOC reads give only content, not sequence) will make it difficult to uniquely identify a bacterial species or state unequivocally whether a known resistance gene is present. Another theoretical question is whether random mutations in the bacterial genomes will compromise correct identification, or whether random noise from the experimental optical measurements will reduce accuracy. While a simulation provides a quick proof-of-concept, it leaves the experimental questions unanswered.

Therefore, the main goal of this appendix was to determine whether data produced in 10-mer blocks could be used in a diagnostic device, meaning that the data could correctly identify species and antibiotic resistance genes with a realistic number of pyramid tips (≤ 1,000,000 tips). To answer this question, this appendix addressed four main objectives: 1) to determine how many BOC reads are needed for species identification; 2) to determine how many BOC reads are needed for single gene detections (such as an antibiotic resistance gene of around 800 bp); 3) to analyze how accuracy is affected by noise from the detecting instrument and from random gene mutations; 4) to identify which learning algorithms are best at accurately identifying the species and genes.
Because this appendix only involves simulation and no experimental work, the methods for this appendix are included below instead of with Chapter 4. This appendix exists because I started a collaboration with Dr. Prashant Nagpal (University of Boulder Colorado) about combining our bacteria-DNA extraction process with the technology he is working on developing as described above. The majority of this appendix was published in Frontiers in Microbiology volume 11 article 257, 2020 (see reference [439]).

C.1 Fractional Base Composition Analysis Methods

Figure C.1 provides an overview of the process for simulating the experimental data reads, converting these into a spectrum, and testing the machine learning algorithms (MLAs). A brief outline follows: a sequence was obtained (see section C.1.1); the sequence was broken into every possible 10-mer for both strands of DNA (see section C.1.2); the 10-mers are binned according to the percent $A$, $T$, $G$, and $C$, resulting in the sequence-specific fractional base content (FBC) spectrum (see section C.1.2); experimental noise was simulated by introducing random errors into the sequence-specific FBC spectrum resulting in the simulated experimental FBC spectrum (see section C.1.3); the spectrum from a purely random sequence (bias) was subtracted from the simulated experimental FBC spectrum, producing a deviation spectrum (see section C.1.4); the FBC deviation spectra from many DNA samples are then analyzed by both principal component analysis (PCA) (see section C.1.5) and the MLAs (see section C.1.6); the MLAs are trained and cross-validated on one set of FBC deviation spectra and then tested against another set of never-before-seen FBC deviation spectra (see section C.1.6).

C.1.1 DNA Sequences

While awaiting experimental data from the SERS instrument being built at the University of Colorado Boulder, simulated SERS BOC data was generated to determine the feasibility of the device in identifying bacterial species and antibiotic resistance genes. Reference genomes for twelve bacterial species, 728 plasmids containing 4 different types of carbapenem antibiotic-resistant genes, and 600 control plasmids not containing any carbapenem resistance genes were collected from the National Center of Biotechnology Informations reference sequence (NCBI Ref-
Figure C.1: Overview of the identification algorithm. In actual experiments, bacterial DNA (genomic or plasmid) was digested into 10-base lengths. In simulated experiments and in teaching the learning algorithms, known DNA sequences are randomly broken into 10-base lengths. The block optical content (BOC) was measured for each 10-mer and put into bins corresponding to the fractional base composition (FBC) of each 10-mer, producing a distribution of FBC for the entire set of BOC measurements. In some cases random error was generated and added to (or subtracted from) the distribution. Finally the distribution of purely random ATCG 10-mers was subtracted from the FBC spectrum to produce a spectrum of deviation from randomness. This deviation spectrum was processed by PCA or the MLAs as described herein.

Seq; see Table C.1 and Appendix Tables C.6 and C.7 for NCBI reference IDs for each genome and plasmid, respectively) [441]. The DNA sequences were separated into genomic and plasmid
DNA (gDNA and pDNA, respectively) and studied separately using principle component analysis (PCA) and several MLAs.

Of the gDNA sequences, 10 of the 12 species are common organisms producing bloodstream infections (Bacteroides fragilis, Campylobacter jejuni, Enterococcus hirae, Escherichia coli, Escherichia fergusonii, Klebsiella pneumoniae, Salmonella enterica, Staphylococcus aureus, Streptococcus pneumoniae, and Streptococcus pyogenes) and were used in both training and testing (different genomic sequences were used for training and testing); and 2 additional species (Klebsiella aerogenes and Mycobacterium tuberculosis) were only used in testing (for taxonomy testing). Median GC% content for the chosen species are included in Table C.1. One NCBI RefSeq genome was used for each species in the training set and one NCBI RefSeq genome was used for each species in the testing set. Only one was used because 1000 genomes are generated from each genome using the original genome FBC spectrum as the probability distribution for creating new genomes (see sections C.1.2 and C.1.3). This means that no 2 simulated genomes are identical, even in the absence of error, and that the MLAs see the original genome and 999 variations for both the training and testing sets for each species, resulting in a similar analysis to one made with multiple NCBI RefSeq genomes per species.

Table C.1: NCBI Reference IDs and GC% for the bacterial genomes.

<table>
<thead>
<tr>
<th>Bacteria</th>
<th>Training</th>
<th>Testing</th>
<th>% GC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bacteroides fragilis</td>
<td>NC_006347.1</td>
<td>CP011073.1</td>
<td>43.4%</td>
</tr>
<tr>
<td>Campylobacter jejuni</td>
<td>NC_002163.1</td>
<td>NC_003912.7</td>
<td>30.4%</td>
</tr>
<tr>
<td>Escherichia coli</td>
<td>U00096.3</td>
<td>AE005174.2</td>
<td>50.6%</td>
</tr>
<tr>
<td>Escherichia fergusonii</td>
<td>NC_011740.1</td>
<td>NZ_CP040805.1</td>
<td>49.9%</td>
</tr>
<tr>
<td>Salmonella enterica</td>
<td>NC_003197.2</td>
<td>CP006631.1</td>
<td>52.1%</td>
</tr>
<tr>
<td>Klebsiella pneumoniae</td>
<td>NC_012731.1</td>
<td>FO834906.1</td>
<td>57.2%</td>
</tr>
<tr>
<td>Staphylococcus aureus</td>
<td>NC_007622.1</td>
<td>BX571856.1</td>
<td>32.7%</td>
</tr>
<tr>
<td>Enterococcus hirae</td>
<td>LR134297.1</td>
<td>NC_018081.1</td>
<td>36.8%</td>
</tr>
<tr>
<td>Streptococcus pyogenes</td>
<td>NC_002737.1</td>
<td>AE014074.1</td>
<td>38.4%</td>
</tr>
<tr>
<td>Streptococcus pneumoniae</td>
<td>NC_003028.3</td>
<td>CP000919.1</td>
<td>39.6%</td>
</tr>
<tr>
<td>Klebsiella aerogenes</td>
<td>-</td>
<td>NC_015663.1</td>
<td>55.0%</td>
</tr>
<tr>
<td>Mycobacterium tuberculosis</td>
<td>-</td>
<td>AP018035.1</td>
<td>65.6%</td>
</tr>
</tbody>
</table>
For the pDNA, all sequenced plasmids from the NCBI RefSeq for the 4 carbapenem resistance plasmids (imipenemase 4 [IMP-4], *Klebsiella pneumoniae* carbapenemase 2 [KPC-2], New Delhi metallo-beta-lactamase 1 [NDM-1], and Verona integron-encoded metallobeta-lactamase 1 [VIM-1]) were used in either training or testing (see Table C.2 for specific number used in training and testing). No variant plasmid data was used (a variant being KPC-4 or VIM-2), and all control plasmids were checked to make sure they did not contain the 4 carbapenem resistance plasmids or any variants. Two different tests were performed on the pDNA: the first test investigated whether the MLAs are able to identify the particular type of resistance (out of 4 types) or identify that none of these are present; the second test grouped the carbapenem resistance plasmids together and investigated whether the MLAs are able to identify the presence (or not) of any carbapenem resistance.

<table>
<thead>
<tr>
<th>Table C.2: Carbapenemase-gene-containing plasmids used in this study.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Plasmid Type</strong></td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>KPC-2</td>
</tr>
<tr>
<td>NDM-1</td>
</tr>
<tr>
<td>VIM-1</td>
</tr>
<tr>
<td>IMP-4</td>
</tr>
<tr>
<td>No-resistance plasmids</td>
</tr>
</tbody>
</table>

*Plasmids were tested against individual resistant types and not resistant (5 groups), or as a resistant vs non-resistant grouping (2 groups)

C.1.2 Generating Sequence-Specific FBC Spectrum

The physical optical instrument reads the BOC of each DNA k-mer bound to r number of SERS pyramids on a silicon chip in the instrument [363]. Since the size (i.e., k bases) of each read was known, these base fractions are converted into specific integer counts of nucleotides for each k-mer. BOC reads are written in the form $A_w T_x G_y C_z$ where $0 \leq w, x, y, z \leq k$, and $w + x + y + z = k$. For traditional sequencing, there are $4^k$ possible reads for a single k-mer; however, there are only $(k+3)!/(k!3!)$ BOC reads corresponding to the different ways of assigning the variables $w, x, y,$
and \( z \), given the previous constraints. The distribution of BOC reads, hereafter called the fractional base content (FBC) spectrum, was defined as the probability distribution function of sampling any BOC read of a specific base composition (see Figure C.2).

In order to create the FBC spectrum for each gDNA and pDNA sequences of interest, each sequence was decomposed into every possible 10-mer block using both complementary strands of DNA (since both strands will be present in a physical system). These blocks are then grouped into their corresponding bins to produce the FBC spectrum. Once all blocks are binned for the given sequence, each bin count was divided by the total 10-mer count of all 286 bins for that DNA sequence to get the sequence-specific probability distribution function, or FBC spectrum.

To simulate BOC reads on a multi-pyramid chip, the FBC spectrum for the selected plasmid or genome was randomly sampled 2.5 million times, using the sequence-specific probability distribution. This represents having more than one copy of the DNA sequence present in a physical experiment. From this 2.5-million-value array, the first \( r \) number of values are selected and distributed into bins to produce a simulated experimental FBC spectrum, where \( r \) represents the number of pyramid tips on the SERS-BOC device. This was done for both the training and testing sets.

C.1.3 Simulating Gene Mutations and Experimental Errors

Due to mutations present in bacteria, any given bacteria species or plasmid will not have a perfectly identical FBC spectrum as that of the corresponding NCBI reference sequence. These natural gene mutations, which are on the order of \( 5 \times 10^{-4} \) to \( 5 \times 10^{-9} \) in bacteria [442, 443], are expected to be overwhelmed by the experimental errors produced during experimental BOC reads and assignments. There are several sources for error in the optical sequencing reads. A few examples are under-digestion and/or over-digestion which results in non-uniform length \( k \)-mer sequences, \( k \)-mer sequences adhering to the pyramid tips such that not all bases can be read, portions of multiple \( k \)-mer sequences adhering to the same pyramid tip, and optical noise from the instrumentation [363]. While creating more realistic training and test samples, a single error rate parameter was introduced to modify the BOC read that accounts for both the expected bacterial mutations and the instrument errors, producing FBC spectra with various levels of random error.
Figure C.2: FBC spectra and deviation from bias FBC for *Klebsiella aerogenes* (▼), *Klebsiella pneumoniae* (■), and *Escherichia coli* (○). The x-axis shows the range of the 286 different FBC 10-mers starting with $A_0T_0G_0C_10$ and ending with $A_1T_0G_0C_0$. The $A$, $T$, $G$, and $C$s are color-coded on a light to dark scale with the lightest shade representing 0 and the darkest shade representing 10. A) Y-axis shows the frequency at which that FBC 10-mer appears in the genome sequence as a percentage of the total FBC 10-mer count; the black line indicates the normal random bias calculated from all possible FBC 10-mers. B) Y-axis represents the deviation from normal randomness of the FBC 10-mer frequencies calculated by subtracting the bias from the frequencies counted in A. The graph shows that even closely related sequences have significantly different FBC spectra.
We define an error rate $m$ (where $0 \leq m \leq 1$) to be the fraction of bases in the reference sequence that are expected to contain an error. Assuming that the errors are randomly distributed throughout the reference sequence, the number of errors in a randomly selected 10-mer is the same as the number of errors in 10 randomly selected bases. The probability of selecting a 10-mer without any errors was determined by a binomial distribution in which the number of trials was the same as the number of pyramid tips, $r$, and the probability of being errorless was one minus the error rate, $1 - m$. To simulate errors, a value of either [0,1] was sampled from the binomial distribution $B(r,m)$. If a 0 was chosen, then a value was chosen from the sequence-specific FBC spectrum (the 2.5-million-value array described above in section C.1.2). If a 1 was chosen, then a random value from the bias spectrum (see section C.1.4 for details) was chosen. This was repeated until a list of values was created that was $r$ in length; next, the list was distributed into bins to produce the “noisy” FBC bins. The resulting FBC 10-mer bin counts are divided by $r$ to obtain the “noisy” FBC spectrum for the given plasmid or genome.

C.1.4 Bias FBC Spectrum

We discovered that a key to enhancing the differences in the FBC spectra of various DNA sequences was to subtract from each FBC spectrum the spectrum of totally random ATGC, leaving a spectrum of deviations from randomness. The resulting spectrum was called the “deviation spectrum” for a particular sequence.

Because the $k$-mer size was known, a purely random spectrum (called the bias spectrum) can be generated by including every possible $k$-mer once. Since there are $4^k$ possible sequential $k$-mers, and given that any BOC read $(A_wT_xG_yC_z)$ has $k!/(w!x!y!z!)$ ways of permuting the base counts to create nucleotide-specific $k$-mers, the bias spectrum can be calculated as:

$$bias(A_wT_xG_yC_z) = \frac{k!}{4^k w!x!y!z!}$$

(C.1)

This bias spectrum was subtracted from the FBC spectrum of the particular DNA sequence to yield a unique-DNA-sequence FBC deviation spectrum. This unique-DNA-sequence deviation spectrum was the deviation from pure randomness and should oscillate around zero.
For our given $k$-mer size of 10, there are 286 bins in the FBC spectrum. The FBC spectrum for a sequence can be visualized by plotting the frequency of a 10-mer in the bins of the spectrum, as shown in Figure C.2. Figure C.2A shows the corresponding FBC spectrum for *E. coli*, *K. pneumoniae*, *K. aerogenes*, and the bias spectrum. Figure C.2B shows the resulting deviation spectra for *E. coli*, *K. pneumoniae*, and *K. aerogenes* produced by subtracting the bias spectrum. In Figure C.2A, the commonness of the sequence was indicated by the height of the peak, with taller peaks being more common, such as $A_3T_2G_3C_2$ and $A_3T_3G_2C_2$. A few of these peaks are labeled for easy comparison with Figure C.2B. As seen in Figure C.2B, some FBC 10-mers appear more often than expected from a random sequence and some FBC 10-mers appear less than expected from a random sequence. While obvious that these bacteria do not have random sequences, it was useful and informative to observe that randomly breaking their DNA into all possible 10-mers does not produce a random FBC spectrum; in fact, unique features appear that suggest a species may be identified by its deviation spectrum.

To test the robustness of identification of species or genes by deviation spectra in the presence of real experimental noise and random mutations in the DNA sequence, random errors are introduced into the FBC spectra (see section C.1.3 for details) and then the resulting noisy data are divided by $r$ and the bias spectrum was subtracted to obtain a simulated FBC deviation spectrum (with noise) for the given plasmid or genome. For each gDNA and pDNA sequence (22 bacterial genomes and 1329 plasmids), 1000 simulated FBC deviation spectra were created from each specific sequence for both the training set and testing set.

### C.1.5 PCA

PCA is a useful analysis that reduces data dimensions while retaining trends and patterns [444]. This technique, which can reduce computational expense, is often used with biological data. For these reasons, PCA was used to investigate whether a simple data reduction analysis could easily identify the different species and antibiotic resistance genes using the noiseless and noisy FBC deviation spectra. As seen in Figures C.3 and C.4, visualizing the first 2 principal components allowed for some of the data to be easily classified into the correct groups, while other data was unclassifiable. Adding a third principal component may have helped cluster the data, but information was not easily retrieved from 3 dimensional plots, especially with thousands of data...
points. From this initial result, MLAs were subsequently examined to determine if supervised learning methods could classify the data.

C.1.6 Testing the MLAs

After creating the FBC deviation spectra for noiseless and noisy DNA BOC reads, the data sets were split into a training set and a never-before-seen test set (both sets of FBC spectra are created through the process detailed by sections C.1.2 - C.1.4) and run through several MLAs to classify the bacterial species or the carbapenem resistance status of plasmids based on their FBC deviation spectra. The attributes used in the classification model were the values corresponding to the probability distribution from each FBC deviation spectrum. The training set was then randomly split into 900 and 100 deviation spectra for each DNA sequence for running 10-fold cross-validation. The MLAs were trained on the 900 deviation spectra from each DNA sequence and then tested against the remaining 100 deviation spectra from each DNA sequence of the training set and then tested against all of the never-before-seen sequences. For the gDNA, that meant that the MLAs were trained on 9,000 training deviation spectra (900 FBC spectra for each of the 10 training genome sequences) and validated against the remaining 1,000 training deviation spectra from the training set and then tested against all the 12,000 testing deviation spectra (1,000 FBC spectra for each of the 12 testing genome sequences). The MLAs were trained and tested similarly for the pDNA.

For the gDNA, the labels used for classifying were: *B. fragilis, C. jejuni, E. hirae, E. coli, E. fergusonii, K. pneumoniae, S. enterica, S. aureus, S. pneumonia*, and *S. pyogenes*. The other two species, *K. aerogenes* and *M. tuberculosis*, are used to investigate how the MLAs group unknown species. For the pDNA, the labels used for the individual classification tests were: KPC, NDM, VIM, IMP and No Resistance; and the labels used for the group classification tests were: Resistance and No Resistance.

For both the gDNA and pDNA, the FBC deviation spectra were created with the following parameters: $k = 10; r = [10^2, 10^3, 10^4, 10^5, 10^6]; m = [0, 0.01, 0.05, 0.1, 0.25, 0.33, 0.5, 0.75, 0.9, 1];$ and $s = 1000,$ where $k$ was the size of the $k$-mer, $r$ was the number of pyramid tips for generating the sample FBC spectra, $m$ was the fractional error rate, and $s$ was the number of FBC
deviation spectra created per DNA sequence (genome or plasmid). All 50 combinations of $r$ and $m$ were tested.

From the machine learning python package, Sci-kit learn, (version 0.20.3) [445], 11 different machine learning algorithms were tested from the following categories: linear machine learning algorithms, decision tree learning algorithms, Nave Bayes learning algorithms, discriminant analyses, and a neural network. For this initial study, default parameters were used for all of these algorithms. Each of these classification models were chosen for their ability to fit data with positive and negative values and to fit data to the model using out-of-core fitting, except for the discriminant analyses as explained below. The results focus on the best algorithm from each category; however, the results for all algorithms can be found in Tables C.3 - C.5.

The following 11 machine learning algorithms were tested: 3 linear machine learning algorithms were tested – Stochastic Gradient Descent Classifier (SGD), Passive Aggressive Classifier (PA), and Perceptron (Per); 3 decision tree machine learning algorithms were tested – Random Forest (RF), Gradient Boosting Classifier (GB), and Extra Trees Classifier (ET); 2 nave bayes algorithms were tested – Bernoulli Nave Bayes (BNB) and Gaussian Nave Bayes (GNB); 2 discriminant analyses were tested – Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA); and 1 neural network whose default is 100 layers and the number of nodes determined by the number of input features.

The 2 discriminant analyses were tested due to the results of the PCA figures in an initial analysis of the genomes. Theses algorithms can only train on 250,000 data samples and cannot use out-of-core learning. Since there was only 10 bacterial species, the LDA and QDA were trained on the full training set. Due to there being 1329 plasmids, the LDA and QDA could not be trained on the full training set. Therefore, the data was randomly selected, such that there was an even number of each type of antibiotic resistance (or resistance grouping) in the training set. However, the LDA and QDA were tested against the full never-before-seen set.

C.1.7 Model Performance Testing and Statistical Analysis

The robustness of each classification model was studied by measuring the predictive accuracy as a function of the parameters of the simulated BOC data. For each optical sequencing read number ($r$) and each error rate ($m$), the performance of the model was quantified by the pre-
dictive accuracy and a confusion matrix, which keeps track of the true positives, true negatives, false positives and false negatives for the sample [446]. The accuracies and confusion matrix presented for each MLA are the average of 10 trials (n=10) for the given r and m. Three different cutoff accuracies (95% for species, 90% for group plasmid, and 75% for individual plasmid) for the never-before-seen sequences were chosen as criteria for assessing the effects of the different error rates and sequencing read numbers.

C.1.8 Simulation

To generate large amounts of simulated experimental data on which to test the different MLAs, the data was produced using code written in Python 3.7 as described above. Reference genomes were downloaded directly from the NCBI RefSeq database. All code used for running the simulation is contained in Appendix D. While these simulations were tested only on plasmids having a carbapenem antibiotic resistance gene, the simulation will work for any set of antibiotic resistance genes in which the full sequence of the gene-containing plasmid is known.

C.2 Results

C.2.1 Principal Component Analysis

Figures C.3 and C.4 visually display the first 2 principal components for the deviation spectra of the species and resistance genes. For species, the PCA data is visually distinct, even when significant noise is added to the BOC data. For resistance gene detection, the PCA could not produce a clear distinction. Details are discussed below.

For both bacterial species and antibiotic resistance gene identification, it is noted that the PCA produced in all cases contains an arch effect, which indicates that the principal components are not completely independent of each other and are thus not completely orthogonal to each other [447]. Since the PCA assumes independence and orthogonality between the principal components, it is not completely reliable as a means of identification without first adjusting for the arch effect. However, PCA was used here to reveal the extent of differences that might be learned by the MLAs.
C.2.1.1 Bacterial Species

For the bacterial species, we found that the PCA revealed significant differences between the various species, even with a 90% error rate for $10^6$ reads. As a control, using a 100% error rate produced no differences (data not shown) because the PCA yields the same value for all species, regardless of the number of reads. We selected *E. coli* and *E. fergusonii*, two genetically very similar bacteria, as a stringent test for sensitivity and discrimination. Because of the similarity of *E. coli* and *E. fergusonii*, the graphical plot can only reveal the difference between these two species for error rates of 0%-33% for $10^5$ reads and 0-90% for $10^6$ reads (see difference between Figure C.3A and C.3B). Figure C.3 contains an example of a non-overlapping (significantly different) PCA (A) and an overlapping (some difference to no difference) PCA (B). Figure C.3 shows the principal component analysis of the FBC deviation spectra data for the bacteria species for 25% error at $10^6$ reads (A) and at $10^4$ reads (B) (see Appendix A for additional PCA figures). This figure shows that with enough BOC reads even in the presence of error, there exists significant differences between the species when using only the first two principal components. This indicates that the FBCs of each of these 12 species are distinct enough that the MLAs should be able to easily classify each species, potentially perfectly, even in the presence of noise, even differentiating very similar species such as *E. coli* and *E. fergusonii*.

C.2.1.2 Antibiotic Resistance

For the plasmids, the PCA revealed that there is a greater spread across the plasmid sequences with no clear differences using only two principal components for any number of reads, even with no error. This is not surprising because the antibiotic resistant gene content could be affecting less than 1% of the FBC deviation spectrum (800 bases on a 200,000-base plasmid) for some of the samples. This indicates that the MLAs will need to detect the small resistance gene signal from a dominating background. In addition to a low signal competing with a strong background, there is a wide range of different DNA signatures in plasmids, increasing the complexity of the task. Figure C.4 highlights this varying range of DNA signatures; both control plasmids and plasmids containing resistance genes span the entire space of the PCA plot. Neither the individual resistance categorization (Figure C.4A & C.4B) nor the group resistance categorization (Figure
C.4C & C.4D) provided any insights into clustering or separation. Figure C.4 shows the principal component analysis of the FBC spectra data for the plasmids for 0% error at $10^6$ reads (A & C) and at $10^4$ reads (B & D) (see Appendix A for additional PCA figures).

Figure C.3: Visualization of bacterial species principal component analysis. Each of the twelve species tested is clustered individually. There are 1000 data points per species and each datum represents a single experiment with random error. The clusters indicate the distinctions in the FBC spectra. A: $10^6$ BOC reads; B: $10^4$ BOC reads. The error rate is 25% for both A and B. In both A and B, the darker triangles represent the never-before-seen genomes for the given species with the lighter circles representing the genomes trained on. The two additional triangles represent the extra species that the MLAs try to categorize according to the learning from the other nine species.
Figure C.4: Visualization of the individual and group plasmid principal component analysis. The plasmid FBC spectra cannot be distinguished using only 2 principal components (500 FBC spectra per plasmid). A: Individual $10^6$ BOC reads; B: Individual $10^4$ BOC reads; C: Group $10^6$ BOC reads; D: Group $10^4$ BOC reads. The error rate is 0% for A-D. In A-D, the darker squares represent the never-before-seen plasmids for the given resistance type with the lighter circles representing the plasmids trained on. The non-resistant plasmids are represented by a darker diamond for the never-before-seen plasmids with a lighter triangle for training plasmids.

C.2.2 Classification Using MLA

The outputs from thousands of simulated SERS-BOC experiments were produced to examine a wide range of combinations of error rates and the number of reads from pyramid tips. The simulation creates hypothetical BOC reads from different bacterial genomes or plasmids which are then converted to FBC deviation spectra and handed to the different MLAs for categorizing. Simulating the experiments allows us to show feasibility and test the robustness of the algorithm in the presence of noise and genetic variation, and to examine the device as a diagnostic tool for bacteria classification and resistance profiling.

The results detailed below showed that the best algorithms from each category are: Passive-Aggressive Classifier (PA, linear machine learning algorithm); Extra Trees Classifier (ET, decision tree algorithm); Gaussian Nave Bayes (GNB, Nave Bayes algorithm); Linear Discriminant Ana-
ysis (LDA, discriminant analysis algorithm); and the neural network (NN). The results for each of these MLAs are examined in detail below and the results for the rest of the MLAs tested can be found in Tables C.3 - C.5.

C.2.2.1 Bacterial Species

The MLA analysis showed that we were able to accurately classify greater than 96% of the simulated unseen bacterial genome data sets (1000 samples per species) using the ET, LDA and NN MLAs employing as low as \(10^4\) BOC reads and up to 50% error. The PA MLA was the worst algorithm (of the best 5 analyzed) and never got better than 78% accuracy (see Table C.3). As for the best algorithms, even at 90% error, the LDA and NN could accurately classify greater than 98% of the bacterial species at \(10^5\) BOC reads. This surprising result at very high noise levels is postulated to occur because random error added to the BOC generates random noise in the FBC spectrum, but the deviation spectrum has the bias spectrum subtracted out which removes random noise. Thus, the MLA models working on the FBC deviation spectra operate on data with a good signal-to-noise level, even up through 90% random error in the original BOC data.

Table C.3: MLA accuracy for the different error rates and number of BOC reads for bacterial species for the never before seen genomes.

<table>
<thead>
<tr>
<th># of BOC reads</th>
<th>Error</th>
<th>PA</th>
<th>Per</th>
<th>SGD</th>
<th>ET</th>
<th>GB</th>
<th>RF</th>
<th>BNB</th>
<th>GNB</th>
<th>LDA</th>
<th>QDA</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100%</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>1000</td>
<td>100%</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>10000</td>
<td>100%</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>100000</td>
<td>100%</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>1000000</td>
<td>100%</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>100</td>
<td>90%</td>
<td>0.13</td>
<td>0.13</td>
<td>0.20</td>
<td>0.67</td>
<td>0.29</td>
<td>0.64</td>
<td>0.32</td>
<td>0.24</td>
<td>0.36</td>
<td>0.47</td>
<td>0.39</td>
</tr>
<tr>
<td>1000</td>
<td>90%</td>
<td>0.14</td>
<td>0.17</td>
<td>0.17</td>
<td>0.75</td>
<td>0.45</td>
<td>0.55</td>
<td>0.45</td>
<td>0.25</td>
<td>0.62</td>
<td>0.61</td>
<td>0.65</td>
</tr>
<tr>
<td>10000</td>
<td>90%</td>
<td>0.12</td>
<td>0.19</td>
<td>0.17</td>
<td>0.83</td>
<td>0.72</td>
<td>0.68</td>
<td>0.75</td>
<td>0.59</td>
<td>0.91</td>
<td>0.77</td>
<td>0.92</td>
</tr>
<tr>
<td>100000</td>
<td>90%</td>
<td>0.14</td>
<td>0.17</td>
<td>0.16</td>
<td>0.94</td>
<td>0.93</td>
<td>0.88</td>
<td>0.91</td>
<td>0.77</td>
<td>0.98</td>
<td>0.74</td>
<td>0.99</td>
</tr>
<tr>
<td>1000000</td>
<td>90%</td>
<td>0.12</td>
<td>0.33</td>
<td>0.18</td>
<td>0.97</td>
<td>0.89</td>
<td>1.00</td>
<td>0.94</td>
<td>0.61</td>
<td>1.00</td>
<td>0.68</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>75%</td>
<td>0.30</td>
<td>0.22</td>
<td>0.22</td>
<td>0.73</td>
<td>0.38</td>
<td>0.60</td>
<td>0.43</td>
<td>0.25</td>
<td>0.48</td>
<td>0.40</td>
<td>0.52</td>
</tr>
<tr>
<td># of BOC reads</td>
<td>Error</td>
<td>PA</td>
<td>Per</td>
<td>SGD</td>
<td>ET</td>
<td>GB</td>
<td>RF</td>
<td>BNB</td>
<td>GNB</td>
<td>LDA</td>
<td>QDA</td>
<td>NN</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1000</td>
<td>75%</td>
<td>0.28</td>
<td>0.27</td>
<td>0.22</td>
<td>0.76</td>
<td>0.56</td>
<td>0.54</td>
<td>0.66</td>
<td>0.42</td>
<td>0.79</td>
<td>0.68</td>
<td>0.80</td>
</tr>
<tr>
<td>10000</td>
<td>75%</td>
<td>0.23</td>
<td>0.36</td>
<td>0.20</td>
<td>0.91</td>
<td>0.84</td>
<td>0.79</td>
<td>0.87</td>
<td>0.82</td>
<td>0.96</td>
<td>0.89</td>
<td>0.97</td>
</tr>
<tr>
<td>100000</td>
<td>75%</td>
<td>0.24</td>
<td>0.36</td>
<td>0.21</td>
<td>0.99</td>
<td>0.97</td>
<td>0.97</td>
<td>0.95</td>
<td>0.84</td>
<td>0.99</td>
<td>0.90</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>75%</td>
<td>0.23</td>
<td>0.34</td>
<td>0.21</td>
<td>1.00</td>
<td>0.94</td>
<td>0.99</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>50%</td>
<td>0.44</td>
<td>0.32</td>
<td>0.38</td>
<td>0.75</td>
<td>0.45</td>
<td>0.55</td>
<td>0.55</td>
<td>0.26</td>
<td>0.61</td>
<td>0.58</td>
<td>0.63</td>
</tr>
<tr>
<td>1000</td>
<td>50%</td>
<td>0.52</td>
<td>0.43</td>
<td>0.40</td>
<td>0.81</td>
<td>0.67</td>
<td>0.64</td>
<td>0.80</td>
<td>0.63</td>
<td>0.90</td>
<td>0.75</td>
<td>0.91</td>
</tr>
<tr>
<td>10000</td>
<td>50%</td>
<td>0.50</td>
<td>0.58</td>
<td>0.39</td>
<td>0.96</td>
<td>0.92</td>
<td>0.88</td>
<td>0.94</td>
<td>0.95</td>
<td>0.98</td>
<td>0.85</td>
<td>0.98</td>
</tr>
<tr>
<td>100000</td>
<td>50%</td>
<td>0.52</td>
<td>0.49</td>
<td>0.44</td>
<td>0.99</td>
<td>0.98</td>
<td>0.97</td>
<td>0.96</td>
<td>0.92</td>
<td>1.00</td>
<td>0.97</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>50%</td>
<td>0.51</td>
<td>0.44</td>
<td>0.43</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>33%</td>
<td>0.53</td>
<td>0.43</td>
<td>0.42</td>
<td>0.75</td>
<td>0.47</td>
<td>0.54</td>
<td>0.62</td>
<td>0.29</td>
<td>0.68</td>
<td>0.61</td>
<td>0.72</td>
</tr>
<tr>
<td>1000</td>
<td>33%</td>
<td>0.59</td>
<td>0.56</td>
<td>0.49</td>
<td>0.84</td>
<td>0.74</td>
<td>0.67</td>
<td>0.84</td>
<td>0.70</td>
<td>0.92</td>
<td>0.79</td>
<td>0.94</td>
</tr>
<tr>
<td>10000</td>
<td>33%</td>
<td>0.60</td>
<td>0.60</td>
<td>0.49</td>
<td>0.97</td>
<td>0.92</td>
<td>0.91</td>
<td>0.94</td>
<td>0.97</td>
<td>0.99</td>
<td>0.88</td>
<td>0.99</td>
</tr>
<tr>
<td>100000</td>
<td>33%</td>
<td>0.59</td>
<td>0.59</td>
<td>0.50</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
<td>0.96</td>
<td>0.93</td>
<td>1.00</td>
<td>0.97</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>33%</td>
<td>0.58</td>
<td>0.50</td>
<td>0.53</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
<td>0.95</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>25%</td>
<td>0.55</td>
<td>0.43</td>
<td>0.47</td>
<td>0.75</td>
<td>0.49</td>
<td>0.54</td>
<td>0.63</td>
<td>0.30</td>
<td>0.69</td>
<td>0.62</td>
<td>0.75</td>
</tr>
<tr>
<td>1000</td>
<td>25%</td>
<td>0.70</td>
<td>0.52</td>
<td>0.49</td>
<td>0.85</td>
<td>0.75</td>
<td>0.69</td>
<td>0.85</td>
<td>0.78</td>
<td>0.92</td>
<td>0.76</td>
<td>0.94</td>
</tr>
<tr>
<td>10000</td>
<td>25%</td>
<td>0.70</td>
<td>0.68</td>
<td>0.55</td>
<td>0.97</td>
<td>0.94</td>
<td>0.92</td>
<td>0.95</td>
<td>0.97</td>
<td>0.99</td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>100000</td>
<td>25%</td>
<td>0.61</td>
<td>0.61</td>
<td>0.50</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>0.97</td>
<td>0.96</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>25%</td>
<td>0.64</td>
<td>0.43</td>
<td>0.55</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
<td>0.90</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>10%</td>
<td>0.56</td>
<td>0.44</td>
<td>0.48</td>
<td>0.74</td>
<td>0.51</td>
<td>0.53</td>
<td>0.66</td>
<td>0.30</td>
<td>0.71</td>
<td>0.62</td>
<td>0.76</td>
</tr>
<tr>
<td>1000</td>
<td>10%</td>
<td>0.72</td>
<td>0.57</td>
<td>0.53</td>
<td>0.86</td>
<td>0.77</td>
<td>0.71</td>
<td>0.85</td>
<td>0.80</td>
<td>0.93</td>
<td>0.82</td>
<td>0.94</td>
</tr>
<tr>
<td>10000</td>
<td>10%</td>
<td>0.74</td>
<td>0.62</td>
<td>0.57</td>
<td>0.97</td>
<td>0.93</td>
<td>0.93</td>
<td>0.96</td>
<td>0.97</td>
<td>0.99</td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>100000</td>
<td>10%</td>
<td>0.68</td>
<td>0.62</td>
<td>0.55</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.96</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>10%</td>
<td>0.62</td>
<td>0.47</td>
<td>0.59</td>
<td>1.00</td>
<td>0.91</td>
<td>0.99</td>
<td>0.89</td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>5%</td>
<td>0.57</td>
<td>0.48</td>
<td>0.49</td>
<td>0.74</td>
<td>0.50</td>
<td>0.53</td>
<td>0.65</td>
<td>0.30</td>
<td>0.71</td>
<td>0.63</td>
<td>0.77</td>
</tr>
<tr>
<td>1000</td>
<td>5%</td>
<td>0.69</td>
<td>0.58</td>
<td>0.53</td>
<td>0.86</td>
<td>0.78</td>
<td>0.71</td>
<td>0.86</td>
<td>0.80</td>
<td>0.94</td>
<td>0.78</td>
<td>0.95</td>
</tr>
<tr>
<td>10000</td>
<td>5%</td>
<td>0.73</td>
<td>0.57</td>
<td>0.57</td>
<td>0.98</td>
<td>0.94</td>
<td>0.93</td>
<td>0.96</td>
<td>0.97</td>
<td>0.99</td>
<td>0.88</td>
<td>0.99</td>
</tr>
<tr>
<td>100000</td>
<td>5%</td>
<td>0.69</td>
<td>0.61</td>
<td>0.49</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1000000</td>
<td>5%</td>
<td>0.69</td>
<td>0.49</td>
<td>0.50</td>
<td>1.00</td>
<td>0.91</td>
<td>0.99</td>
<td>0.87</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>100</td>
<td>1%</td>
<td>0.57</td>
<td>0.48</td>
<td>0.46</td>
<td>0.74</td>
<td>0.50</td>
<td>0.53</td>
<td>0.66</td>
<td>0.31</td>
<td>0.72</td>
<td>0.62</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Figure C.5: Species model accuracy for different number of BOC reads and error rates. The accuracy of the 5 MLAs is shown with (A) 10% error, and (B) 25% error in identifying the never-before-seen bacterial species sets at different number of BOC reads. The accuracy of the 5 MLAs is shown with (C) $10^4$ BOC reads, and (D) $10^3$ BOC reads in identifying the never-before-seen bacterial species sets at different error rates. The red line represents 95% accuracy threshold.
The results suggest that a SERS-BOC device only needs a 100 x 100 pyramid array (10^4 reads per experiment) to accurately identify bacterial species. Figure C.5 shows the MLA accuracy for an error rate of 10% (A) and 25% (B) for all of the BOC reads. This figure shows that greater than 97% accuracy is maintained at 10^4 BOC reads for both 10% and 25% error. Figure C.5 also shows how the various MLAs hold up to error for 10^4 BOC reads (C) and 10^3 BOC reads (D).

![Taxonomy of bacterial species](image)

**Figure C.6**: Taxonomy of the bacterial species pulled from the NCBI database to show how the species are related.

The confusion matrix for the MLAs shows how the models mislabel the species they were trained on, and how they label the 2 species which are not defined in the model. Figure C.7 shows the species confusion matrix for the 5 MLAs for 10^4 BOC reads and 10% error. While most of the species in this study are not closely related (see Figure C.6 for details), the best 3 MLAs only had problems identifying between *E. coli* and *E. fergusonii*, which are genetically very similar (same family). The ET had a sensitivity rate of 82% for *E. coli* and a specificity rate of 94% for *E. fergusonii* when comparing the two genomes. The LDA had a sensitivity rate of 92% for *E. coli* and a specificity rate of 100% for *E. fergusonii* when comparing the two genomes; meaning that the LDA could correctly identify *E. fergusonii* but not *E. coli*. The NN had a sensitivity rate of 93% for *E. coli* and a specificity rate of 100% for *E. fergusonii* when comparing the two genomes. All other genomes trained on had 100% identification for the ET, LDA, and NN MLAs. As for the classification of the two extra species (*K. aerogenes* and *M. tuberculosis*), all 3 MLAs identify *M. tuberculosis* as *K. pneumoniae* (unrelated) 100% of the time. *K. aerogenes* was identified as both *K.*
pneumoniae (same genus) and S. enterica (same family) by all 3 MLAs. Other details are found in Figure C.7.

Figure C.7: Confusion matrix for the 5 MLAs for species identification. $r = 10^4$, $m = 0.10$ The x-axis represents the label predicted by the algorithm and the y-axis represents the true label.

C.2.2.2 Antibiotic Resistance

In the study of the individual resistance categorization (distinguishing 4 carbapenem resistance genes), the MLAs had difficulty achieving better than 75% accuracy. Only with $10^5$ BOC reads (or more) did the ET algorithm attain better than 75% accuracy, and even this algorithm only achieved 80% accuracy at best. The LDA and ET algorithms had similar accuracy at $10^4$ reads (see Figure C.8D) but the LDA did not perform as well at $10^5$ reads (Figure C.8C). The GNB and PA MLAs were the worst but were still able to achieve 64% accuracy (see Figure C.8 and Table C.4). Figure C.8 shows the MLA accuracy for an error
rate of 10% (A) and 25% (B) for all of the BOC reads for the individual scenarios. Figures C.5A and C.5B show that greater than 75% accuracy is maintained at $10^5$ BOC reads for both 10% and 25% error. Figure C.8 also shows how the MLAs hold up to error for $10^5$ BOC reads (C) and $10^4$ BOC reads (D).

Table C.4: MLA accuracy for the different error rates and number of BOC reads for the antibiotic resistances with individual resistance categorization for the never before seen plasmids.

<table>
<thead>
<tr>
<th># of BOC reads</th>
<th>Error</th>
<th>PA</th>
<th>Per</th>
<th>SGD</th>
<th>ET</th>
<th>GB</th>
<th>RF</th>
<th>BNB</th>
<th>GNB</th>
<th>LDA</th>
<th>QDA</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100%</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>1000</td>
<td>100%</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>10000</td>
<td>100%</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>100000</td>
<td>100%</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>1000000</td>
<td>100%</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>100</td>
<td>90%</td>
<td>0.36</td>
<td>0.23</td>
<td>0.29</td>
<td>0.57</td>
<td>0.37</td>
<td>0.54</td>
<td>0.48</td>
<td>0.35</td>
<td>0.46</td>
<td>0.20</td>
<td>0.49</td>
</tr>
<tr>
<td>1000</td>
<td>90%</td>
<td>0.41</td>
<td>0.23</td>
<td>0.38</td>
<td>0.60</td>
<td>0.51</td>
<td>0.54</td>
<td>0.53</td>
<td>0.45</td>
<td>0.53</td>
<td>0.29</td>
<td>0.52</td>
</tr>
<tr>
<td>10000</td>
<td>90%</td>
<td>0.40</td>
<td>0.39</td>
<td>0.39</td>
<td>0.64</td>
<td>0.56</td>
<td>0.60</td>
<td>0.56</td>
<td>0.56</td>
<td>0.65</td>
<td>0.49</td>
<td>0.55</td>
</tr>
<tr>
<td>100000</td>
<td>90%</td>
<td>0.36</td>
<td>0.31</td>
<td>0.39</td>
<td>0.71</td>
<td>0.64</td>
<td>0.67</td>
<td>0.59</td>
<td>0.59</td>
<td>0.74</td>
<td>0.63</td>
<td>0.56</td>
</tr>
<tr>
<td>1000000</td>
<td>90%</td>
<td>0.37</td>
<td>0.29</td>
<td>0.40</td>
<td>0.77</td>
<td>0.70</td>
<td>0.72</td>
<td>0.63</td>
<td>0.63</td>
<td>0.75</td>
<td>0.79</td>
<td>0.56</td>
</tr>
<tr>
<td>100</td>
<td>75%</td>
<td>0.48</td>
<td>0.26</td>
<td>0.51</td>
<td>0.60</td>
<td>0.47</td>
<td>0.53</td>
<td>0.52</td>
<td>0.42</td>
<td>0.49</td>
<td>0.26</td>
<td>0.51</td>
</tr>
<tr>
<td>1000</td>
<td>75%</td>
<td>0.50</td>
<td>0.39</td>
<td>0.54</td>
<td>0.61</td>
<td>0.53</td>
<td>0.56</td>
<td>0.56</td>
<td>0.52</td>
<td>0.58</td>
<td>0.38</td>
<td>0.55</td>
</tr>
<tr>
<td>10000</td>
<td>75%</td>
<td>0.48</td>
<td>0.43</td>
<td>0.55</td>
<td>0.68</td>
<td>0.58</td>
<td>0.65</td>
<td>0.59</td>
<td>0.59</td>
<td>0.70</td>
<td>0.50</td>
<td>0.57</td>
</tr>
<tr>
<td>100000</td>
<td>75%</td>
<td>0.49</td>
<td>0.42</td>
<td>0.55</td>
<td>0.75</td>
<td>0.68</td>
<td>0.70</td>
<td>0.62</td>
<td>0.62</td>
<td>0.75</td>
<td>0.71</td>
<td>0.58</td>
</tr>
<tr>
<td>1000000</td>
<td>75%</td>
<td>0.50</td>
<td>0.42</td>
<td>0.56</td>
<td>0.78</td>
<td>0.73</td>
<td>0.74</td>
<td>0.63</td>
<td>0.64</td>
<td>0.72</td>
<td>0.85</td>
<td>0.58</td>
</tr>
<tr>
<td>100</td>
<td>50%</td>
<td>0.52</td>
<td>0.43</td>
<td>0.57</td>
<td>0.60</td>
<td>0.50</td>
<td>0.54</td>
<td>0.54</td>
<td>0.49</td>
<td>0.52</td>
<td>0.36</td>
<td>0.53</td>
</tr>
<tr>
<td>1000</td>
<td>50%</td>
<td>0.55</td>
<td>0.50</td>
<td>0.62</td>
<td>0.64</td>
<td>0.55</td>
<td>0.60</td>
<td>0.58</td>
<td>0.55</td>
<td>0.63</td>
<td>0.38</td>
<td>0.59</td>
</tr>
<tr>
<td>10000</td>
<td>50%</td>
<td>0.61</td>
<td>0.52</td>
<td>0.62</td>
<td>0.72</td>
<td>0.63</td>
<td>0.68</td>
<td>0.61</td>
<td>0.59</td>
<td>0.73</td>
<td>0.58</td>
<td>0.65</td>
</tr>
<tr>
<td>100000</td>
<td>50%</td>
<td>0.60</td>
<td>0.53</td>
<td>0.63</td>
<td>0.76</td>
<td>0.70</td>
<td>0.72</td>
<td>0.64</td>
<td>0.63</td>
<td>0.73</td>
<td>0.76</td>
<td>0.66</td>
</tr>
<tr>
<td>1000000</td>
<td>50%</td>
<td>0.59</td>
<td>0.53</td>
<td>0.63</td>
<td>0.79</td>
<td>0.75</td>
<td>0.75</td>
<td>0.64</td>
<td>0.65</td>
<td>0.70</td>
<td>0.87</td>
<td>0.66</td>
</tr>
<tr>
<td>100</td>
<td>33%</td>
<td>0.52</td>
<td>0.35</td>
<td>0.58</td>
<td>0.60</td>
<td>0.51</td>
<td>0.54</td>
<td>0.55</td>
<td>0.50</td>
<td>0.54</td>
<td>0.36</td>
<td>0.55</td>
</tr>
<tr>
<td>1000</td>
<td>33%</td>
<td>0.58</td>
<td>0.57</td>
<td>0.62</td>
<td>0.65</td>
<td>0.56</td>
<td>0.61</td>
<td>0.60</td>
<td>0.57</td>
<td>0.65</td>
<td>0.48</td>
<td>0.64</td>
</tr>
<tr>
<td>10000</td>
<td>33%</td>
<td>0.63</td>
<td>0.59</td>
<td>0.63</td>
<td>0.73</td>
<td>0.65</td>
<td>0.69</td>
<td>0.62</td>
<td>0.60</td>
<td>0.74</td>
<td>0.62</td>
<td>0.68</td>
</tr>
<tr>
<td>100000</td>
<td>33%</td>
<td>0.63</td>
<td>0.60</td>
<td>0.64</td>
<td>0.77</td>
<td>0.71</td>
<td>0.73</td>
<td>0.64</td>
<td>0.64</td>
<td>0.73</td>
<td>0.77</td>
<td>0.69</td>
</tr>
<tr>
<td># of BOC reads</td>
<td>Error</td>
<td>PA</td>
<td>Per</td>
<td>SGD</td>
<td>ET</td>
<td>GB</td>
<td>RF</td>
<td>BNB</td>
<td>GNB</td>
<td>LDA</td>
<td>QDA</td>
<td>NN</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1000000</td>
<td>33%</td>
<td>0.62</td>
<td>0.60</td>
<td>0.64</td>
<td>0.79</td>
<td>0.75</td>
<td>0.75</td>
<td>0.64</td>
<td>0.65</td>
<td>0.70</td>
<td>0.87</td>
<td>0.69</td>
</tr>
<tr>
<td>100</td>
<td>25%</td>
<td>0.52</td>
<td>0.43</td>
<td>0.58</td>
<td>0.60</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.50</td>
<td>0.54</td>
<td>0.37</td>
<td>0.56</td>
</tr>
<tr>
<td>1000</td>
<td>25%</td>
<td>0.60</td>
<td>0.57</td>
<td>0.63</td>
<td>0.65</td>
<td>0.56</td>
<td>0.61</td>
<td>0.60</td>
<td>0.57</td>
<td>0.66</td>
<td>0.48</td>
<td>0.65</td>
</tr>
<tr>
<td>10000</td>
<td>25%</td>
<td>0.63</td>
<td>0.62</td>
<td>0.64</td>
<td>0.74</td>
<td>0.65</td>
<td>0.69</td>
<td>0.63</td>
<td>0.60</td>
<td>0.74</td>
<td>0.63</td>
<td>0.70</td>
</tr>
<tr>
<td>100000</td>
<td>25%</td>
<td>0.64</td>
<td>0.64</td>
<td>0.65</td>
<td>0.78</td>
<td>0.72</td>
<td>0.72</td>
<td>0.64</td>
<td>0.64</td>
<td>0.72</td>
<td>0.78</td>
<td>0.71</td>
</tr>
<tr>
<td>1000000</td>
<td>25%</td>
<td>0.64</td>
<td>0.64</td>
<td>0.65</td>
<td>0.80</td>
<td>0.75</td>
<td>0.74</td>
<td>0.64</td>
<td>0.65</td>
<td>0.70</td>
<td>0.87</td>
<td>0.71</td>
</tr>
<tr>
<td>100</td>
<td>10%</td>
<td>0.52</td>
<td>0.40</td>
<td>0.58</td>
<td>0.60</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.50</td>
<td>0.55</td>
<td>0.36</td>
<td>0.57</td>
</tr>
<tr>
<td>1000</td>
<td>10%</td>
<td>0.62</td>
<td>0.59</td>
<td>0.63</td>
<td>0.66</td>
<td>0.57</td>
<td>0.62</td>
<td>0.60</td>
<td>0.57</td>
<td>0.67</td>
<td>0.49</td>
<td>0.66</td>
</tr>
<tr>
<td>10000</td>
<td>10%</td>
<td>0.64</td>
<td>0.66</td>
<td>0.64</td>
<td>0.74</td>
<td>0.66</td>
<td>0.69</td>
<td>0.63</td>
<td>0.60</td>
<td>0.74</td>
<td>0.64</td>
<td>0.71</td>
</tr>
<tr>
<td>100000</td>
<td>10%</td>
<td>0.66</td>
<td>0.67</td>
<td>0.65</td>
<td>0.78</td>
<td>0.72</td>
<td>0.73</td>
<td>0.64</td>
<td>0.64</td>
<td>0.72</td>
<td>0.79</td>
<td>0.71</td>
</tr>
<tr>
<td>1000000</td>
<td>10%</td>
<td>0.64</td>
<td>0.67</td>
<td>0.65</td>
<td>0.80</td>
<td>0.75</td>
<td>0.74</td>
<td>0.64</td>
<td>0.65</td>
<td>0.69</td>
<td>0.87</td>
<td>0.72</td>
</tr>
<tr>
<td>100</td>
<td>5%</td>
<td>0.54</td>
<td>0.40</td>
<td>0.58</td>
<td>0.60</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.50</td>
<td>0.55</td>
<td>0.36</td>
<td>0.57</td>
</tr>
<tr>
<td>1000</td>
<td>5%</td>
<td>0.62</td>
<td>0.59</td>
<td>0.63</td>
<td>0.66</td>
<td>0.57</td>
<td>0.62</td>
<td>0.60</td>
<td>0.57</td>
<td>0.67</td>
<td>0.50</td>
<td>0.67</td>
</tr>
<tr>
<td>10000</td>
<td>5%</td>
<td>0.65</td>
<td>0.66</td>
<td>0.65</td>
<td>0.74</td>
<td>0.66</td>
<td>0.69</td>
<td>0.63</td>
<td>0.60</td>
<td>0.74</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>100000</td>
<td>5%</td>
<td>0.66</td>
<td>0.67</td>
<td>0.65</td>
<td>0.78</td>
<td>0.72</td>
<td>0.73</td>
<td>0.63</td>
<td>0.64</td>
<td>0.72</td>
<td>0.79</td>
<td>0.72</td>
</tr>
<tr>
<td>1000000</td>
<td>5%</td>
<td>0.65</td>
<td>0.67</td>
<td>0.65</td>
<td>0.80</td>
<td>0.76</td>
<td>0.74</td>
<td>0.64</td>
<td>0.65</td>
<td>0.69</td>
<td>0.87</td>
<td>0.72</td>
</tr>
<tr>
<td>100</td>
<td>1%</td>
<td>0.55</td>
<td>0.41</td>
<td>0.58</td>
<td>0.60</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.50</td>
<td>0.55</td>
<td>0.36</td>
<td>0.57</td>
</tr>
<tr>
<td>1000</td>
<td>1%</td>
<td>0.62</td>
<td>0.59</td>
<td>0.63</td>
<td>0.66</td>
<td>0.57</td>
<td>0.62</td>
<td>0.61</td>
<td>0.57</td>
<td>0.67</td>
<td>0.50</td>
<td>0.67</td>
</tr>
<tr>
<td>10000</td>
<td>1%</td>
<td>0.65</td>
<td>0.66</td>
<td>0.65</td>
<td>0.74</td>
<td>0.66</td>
<td>0.69</td>
<td>0.63</td>
<td>0.60</td>
<td>0.74</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>100000</td>
<td>1%</td>
<td>0.67</td>
<td>0.68</td>
<td>0.65</td>
<td>0.78</td>
<td>0.72</td>
<td>0.74</td>
<td>0.63</td>
<td>0.64</td>
<td>0.72</td>
<td>0.79</td>
<td>0.72</td>
</tr>
<tr>
<td>1000000</td>
<td>1%</td>
<td>0.66</td>
<td>0.68</td>
<td>0.65</td>
<td>0.80</td>
<td>0.76</td>
<td>0.73</td>
<td>0.64</td>
<td>0.65</td>
<td>0.69</td>
<td>0.87</td>
<td>0.72</td>
</tr>
<tr>
<td>100</td>
<td>0%</td>
<td>0.54</td>
<td>0.43</td>
<td>0.58</td>
<td>0.60</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.50</td>
<td>0.55</td>
<td>0.37</td>
<td>0.57</td>
</tr>
<tr>
<td>1000</td>
<td>0%</td>
<td>0.62</td>
<td>0.59</td>
<td>0.63</td>
<td>0.66</td>
<td>0.57</td>
<td>0.62</td>
<td>0.61</td>
<td>0.57</td>
<td>0.67</td>
<td>0.50</td>
<td>0.67</td>
</tr>
<tr>
<td>10000</td>
<td>0%</td>
<td>0.64</td>
<td>0.66</td>
<td>0.65</td>
<td>0.74</td>
<td>0.66</td>
<td>0.69</td>
<td>0.63</td>
<td>0.60</td>
<td>0.74</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>100000</td>
<td>0%</td>
<td>0.66</td>
<td>0.68</td>
<td>0.65</td>
<td>0.78</td>
<td>0.72</td>
<td>0.74</td>
<td>0.63</td>
<td>0.64</td>
<td>0.72</td>
<td>0.79</td>
<td>0.72</td>
</tr>
<tr>
<td>1000000</td>
<td>0%</td>
<td>0.66</td>
<td>0.68</td>
<td>0.65</td>
<td>0.80</td>
<td>0.77</td>
<td>0.73</td>
<td>0.64</td>
<td>0.65</td>
<td>0.69</td>
<td>0.87</td>
<td>0.72</td>
</tr>
</tbody>
</table>
Figure C.8: Individual antibiotic-resistance model accuracy for different number of BOC reads and error rates. The accuracy of the 5 MLAs is shown with (A) 10% error, and (B) 25% error in identifying the never-before-seen plasmid sets for individual identification. The accuracy of the 5 MLAs is shown with (C) $10^5$ BOC reads, and (D) $10^4$ BOC reads in identifying the never-before-seen plasmid sets for individual identification. The red line represents a 75% accuracy threshold.

For the group resistance categorization, the MLA analysis was able to accurately classify 90% of the simulated unseen data set using the ET, LDA and NN MLAs down to $10^4$ BOC reads and up to 50% error. The GNB MLA performed the worst of those examined but was still able to achieve 86% accuracy (see Figure C.9 and Table C.5). Figure C.9 shows the MLA accuracy for an error rate of 10% (A) and 25% (B) for all of the BOC reads for the group scenario. Figure C.9 shows that greater than 90% accuracy is maintained at $10^4$ BOC reads for both 10% and 25% error. Figure C.9 also shows how the MLAs hold up to error for $10^5$ BOC reads (C) and $10^4$ BOC reads (D).

Table C.5: MLA accuracy for the different error rates and number of BOC reads for the antibiotic resistances with group resistance categorization for the never before seen plasmids.

<table>
<thead>
<tr>
<th># of BOC reads</th>
<th>Error</th>
<th>PA</th>
<th>Per</th>
<th>SGD</th>
<th>ET</th>
<th>GB</th>
<th>RF</th>
<th>BNB</th>
<th>GNB</th>
<th>LDA</th>
<th>QDA</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100%</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>1000</td>
<td>100%</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td># of BOC reads</td>
<td>Error</td>
<td>PA</td>
<td>Per</td>
<td>SGD</td>
<td>ET</td>
<td>GB</td>
<td>RF</td>
<td>BNB</td>
<td>GNB</td>
<td>LDA</td>
<td>QDA</td>
<td>NN</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>10000</td>
<td>100%</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>100000</td>
<td>100%</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>1000000</td>
<td>100%</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>100</td>
<td>90%</td>
<td>0.76</td>
<td>0.69</td>
<td>0.37</td>
<td>0.78</td>
<td>0.71</td>
<td>0.76</td>
<td>0.74</td>
<td>0.76</td>
<td>0.76</td>
<td>0.69</td>
<td>0.76</td>
</tr>
<tr>
<td>1000</td>
<td>90%</td>
<td>0.79</td>
<td>0.70</td>
<td>0.37</td>
<td>0.82</td>
<td>0.80</td>
<td>0.76</td>
<td>0.81</td>
<td>0.82</td>
<td>0.81</td>
<td>0.78</td>
<td>0.82</td>
</tr>
<tr>
<td>10000</td>
<td>90%</td>
<td>0.78</td>
<td>0.75</td>
<td>0.37</td>
<td>0.86</td>
<td>0.84</td>
<td>0.79</td>
<td>0.83</td>
<td>0.83</td>
<td>0.86</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td>100000</td>
<td>90%</td>
<td>0.78</td>
<td>0.71</td>
<td>0.37</td>
<td>0.92</td>
<td>0.88</td>
<td>0.85</td>
<td>0.81</td>
<td>0.84</td>
<td>0.91</td>
<td>0.88</td>
<td>0.85</td>
</tr>
<tr>
<td>1000000</td>
<td>90%</td>
<td>0.77</td>
<td>0.71</td>
<td>0.37</td>
<td>0.94</td>
<td>0.90</td>
<td>0.90</td>
<td>0.80</td>
<td>0.85</td>
<td>0.91</td>
<td>0.91</td>
<td>0.86</td>
</tr>
<tr>
<td>100</td>
<td>75%</td>
<td>0.76</td>
<td>0.65</td>
<td>0.52</td>
<td>0.80</td>
<td>0.75</td>
<td>0.76</td>
<td>0.79</td>
<td>0.80</td>
<td>0.80</td>
<td>0.73</td>
<td>0.81</td>
</tr>
<tr>
<td>1000</td>
<td>75%</td>
<td>0.81</td>
<td>0.75</td>
<td>0.47</td>
<td>0.83</td>
<td>0.83</td>
<td>0.77</td>
<td>0.82</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>10000</td>
<td>75%</td>
<td>0.81</td>
<td>0.77</td>
<td>0.46</td>
<td>0.89</td>
<td>0.86</td>
<td>0.82</td>
<td>0.82</td>
<td>0.84</td>
<td>0.88</td>
<td>0.85</td>
<td>0.87</td>
</tr>
<tr>
<td>100000</td>
<td>75%</td>
<td>0.79</td>
<td>0.80</td>
<td>0.46</td>
<td>0.94</td>
<td>0.89</td>
<td>0.89</td>
<td>0.78</td>
<td>0.85</td>
<td>0.92</td>
<td>0.91</td>
<td>0.89</td>
</tr>
<tr>
<td>1000000</td>
<td>75%</td>
<td>0.80</td>
<td>0.81</td>
<td>0.46</td>
<td>0.94</td>
<td>0.90</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.90</td>
<td>0.91</td>
<td>0.89</td>
</tr>
<tr>
<td>100</td>
<td>50%</td>
<td>0.80</td>
<td>0.74</td>
<td>0.82</td>
<td>0.82</td>
<td>0.79</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.81</td>
<td>0.79</td>
<td>0.82</td>
</tr>
<tr>
<td>1000</td>
<td>50%</td>
<td>0.82</td>
<td>0.77</td>
<td>0.83</td>
<td>0.85</td>
<td>0.84</td>
<td>0.79</td>
<td>0.82</td>
<td>0.83</td>
<td>0.85</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td>10000</td>
<td>50%</td>
<td>0.83</td>
<td>0.81</td>
<td>0.83</td>
<td>0.91</td>
<td>0.88</td>
<td>0.85</td>
<td>0.79</td>
<td>0.84</td>
<td>0.90</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>100000</td>
<td>50%</td>
<td>0.84</td>
<td>0.82</td>
<td>0.83</td>
<td>0.94</td>
<td>0.90</td>
<td>0.91</td>
<td>0.78</td>
<td>0.85</td>
<td>0.92</td>
<td>0.92</td>
<td>0.90</td>
</tr>
<tr>
<td>1000000</td>
<td>50%</td>
<td>0.84</td>
<td>0.81</td>
<td>0.83</td>
<td>0.94</td>
<td>0.92</td>
<td>0.90</td>
<td>0.78</td>
<td>0.86</td>
<td>0.90</td>
<td>0.88</td>
<td>0.90</td>
</tr>
<tr>
<td>100</td>
<td>33%</td>
<td>0.79</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.81</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.81</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>1000</td>
<td>33%</td>
<td>0.81</td>
<td>0.78</td>
<td>0.83</td>
<td>0.86</td>
<td>0.85</td>
<td>0.80</td>
<td>0.82</td>
<td>0.83</td>
<td>0.86</td>
<td>0.84</td>
<td>0.86</td>
</tr>
<tr>
<td>10000</td>
<td>33%</td>
<td>0.85</td>
<td>0.83</td>
<td>0.83</td>
<td>0.92</td>
<td>0.88</td>
<td>0.86</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>33%</td>
<td>0.86</td>
<td>0.86</td>
<td>0.83</td>
<td>0.94</td>
<td>0.90</td>
<td>0.91</td>
<td>0.78</td>
<td>0.85</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>33%</td>
<td>0.86</td>
<td>0.86</td>
<td>0.83</td>
<td>0.94</td>
<td>0.92</td>
<td>0.90</td>
<td>0.78</td>
<td>0.86</td>
<td>0.91</td>
<td>0.88</td>
<td>0.91</td>
</tr>
<tr>
<td>100</td>
<td>25%</td>
<td>0.80</td>
<td>0.76</td>
<td>0.83</td>
<td>0.82</td>
<td>0.81</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>1000</td>
<td>25%</td>
<td>0.81</td>
<td>0.77</td>
<td>0.83</td>
<td>0.86</td>
<td>0.85</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.87</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>10000</td>
<td>25%</td>
<td>0.86</td>
<td>0.83</td>
<td>0.83</td>
<td>0.93</td>
<td>0.89</td>
<td>0.86</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>25%</td>
<td>0.84</td>
<td>0.87</td>
<td>0.83</td>
<td>0.95</td>
<td>0.90</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>25%</td>
<td>0.87</td>
<td>0.86</td>
<td>0.83</td>
<td>0.94</td>
<td>0.91</td>
<td>0.89</td>
<td>0.78</td>
<td>0.86</td>
<td>0.91</td>
<td>0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>100</td>
<td>10%</td>
<td>0.81</td>
<td>0.77</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>1000</td>
<td>10%</td>
<td>0.82</td>
<td>0.79</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.87</td>
<td>0.84</td>
<td>0.87</td>
</tr>
</tbody>
</table>

259
<table>
<thead>
<tr>
<th># of BOC reads</th>
<th>Error</th>
<th>PA</th>
<th>Per</th>
<th>SGD</th>
<th>ET</th>
<th>GB</th>
<th>RF</th>
<th>BNB</th>
<th>GNB</th>
<th>LDA</th>
<th>QDA</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>10%</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.93</td>
<td>0.89</td>
<td>0.86</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>10%</td>
<td>0.87</td>
<td>0.87</td>
<td>0.83</td>
<td>0.94</td>
<td>0.91</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>10%</td>
<td>0.89</td>
<td>0.88</td>
<td>0.83</td>
<td>0.94</td>
<td>0.92</td>
<td>0.89</td>
<td>0.77</td>
<td>0.86</td>
<td>0.91</td>
<td>0.87</td>
<td>0.92</td>
</tr>
<tr>
<td>100</td>
<td>5%</td>
<td>0.80</td>
<td>0.76</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>1000</td>
<td>5%</td>
<td>0.82</td>
<td>0.80</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.87</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>10000</td>
<td>5%</td>
<td>0.86</td>
<td>0.87</td>
<td>0.83</td>
<td>0.93</td>
<td>0.89</td>
<td>0.87</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>5%</td>
<td>0.86</td>
<td>0.87</td>
<td>0.83</td>
<td>0.95</td>
<td>0.91</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>5%</td>
<td>0.88</td>
<td>0.88</td>
<td>0.83</td>
<td>0.94</td>
<td>0.92</td>
<td>0.90</td>
<td>0.77</td>
<td>0.86</td>
<td>0.91</td>
<td>0.87</td>
<td>0.92</td>
</tr>
<tr>
<td>100</td>
<td>1%</td>
<td>0.79</td>
<td>0.76</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>1000</td>
<td>1%</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.87</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>10000</td>
<td>1%</td>
<td>0.85</td>
<td>0.87</td>
<td>0.83</td>
<td>0.93</td>
<td>0.89</td>
<td>0.87</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>1%</td>
<td>0.85</td>
<td>0.87</td>
<td>0.83</td>
<td>0.95</td>
<td>0.91</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>1%</td>
<td>0.87</td>
<td>0.88</td>
<td>0.83</td>
<td>0.94</td>
<td>0.91</td>
<td>0.90</td>
<td>0.77</td>
<td>0.86</td>
<td>0.91</td>
<td>0.87</td>
<td>0.92</td>
</tr>
<tr>
<td>100</td>
<td>0%</td>
<td>0.80</td>
<td>0.76</td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>0.76</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.82</td>
<td>0.83</td>
</tr>
<tr>
<td>1000</td>
<td>0%</td>
<td>0.81</td>
<td>0.82</td>
<td>0.83</td>
<td>0.87</td>
<td>0.85</td>
<td>0.80</td>
<td>0.81</td>
<td>0.83</td>
<td>0.87</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>10000</td>
<td>0%</td>
<td>0.86</td>
<td>0.87</td>
<td>0.83</td>
<td>0.93</td>
<td>0.89</td>
<td>0.87</td>
<td>0.78</td>
<td>0.84</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>100000</td>
<td>0%</td>
<td>0.87</td>
<td>0.88</td>
<td>0.83</td>
<td>0.94</td>
<td>0.91</td>
<td>0.91</td>
<td>0.78</td>
<td>0.86</td>
<td>0.92</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>1000000</td>
<td>0%</td>
<td>0.87</td>
<td>0.88</td>
<td>0.83</td>
<td>0.94</td>
<td>0.91</td>
<td>0.91</td>
<td>0.77</td>
<td>0.86</td>
<td>0.91</td>
<td>0.87</td>
<td>0.92</td>
</tr>
</tbody>
</table>
Figure C.9: Group antibiotic-resistance model accuracy with different number of BOC reads and error rates. The accuracy of the 5 MLAs is shown with (A) 10% error, and (B) 25% error in identifying the never-before-seen plasmid sets for group identification. The accuracy of the 5 MLAs is shown with (C) $10^5$ BOC reads, and (D) $10^3$ BOC reads in identifying the never-before-seen plasmid sets for group identification. The red line represents a 90% accuracy threshold.

The ET MLA model for the individual resistance categorization maintains 74% accuracy for up to 25% error for $10^4$ BOC reads and maintains 80% accuracy for up to 25% error for $10^6$ BOC reads. The individual results suggest that a SERS-BOC device would need to be at least a 100 x 100 pyramid array ($10^4$ reads per experiment), but a 1,000 x 1,000 pyramid array ($10^6$ reads per experiment) would be optimal for the best results to accurately identify antibiotic resistance genes on plasmids. For the group resistance categorization, the three best MLA models maintain a good signal-to-noise levels up through 50% error. The group results suggest that a SERS-BOC device with a 100 x 100 pyramid array would be sufficient to accurately identify whether any carbapenem resistance gene was present in the presence of 50% experimental noise.
The confusion matrix for the MLAs shows how the models mislabel the plasmids, which allows us to calculate the false-negative rate for the antibiotic-resistant plasmids as a group, as well as the sensitivity of each individual plasmid. The false-negative rate metric is clinically important as it is incorrectly labeling resistant plasmids as not being resistant, which could result in the wrong (ineffective) antibiotics being given to the patient, possibly leading to death. The sensitivity metric shows which type of resistance gene is harder to identify. Figure C.10 shows the confusion matrix for the 5 MLAs for $10^5$ BOC reads and 10% error for the plasmid sets for individual identification. Figure C.11 shows the confusion matrix for the 5 MLAs for $10^5$ BOC reads and 10% error for the plasmid sets for group identification.

The sensitivity rate at $10^5$ BOC reads and 10% error is the values on the diagonal on the confusion matrices shown in Figures C.10 and C.11. The clinically important false negative rates at $10^5$ BOC reads...
Figure C.11: Confusion matrix for the 5 MLAs for group identification of the plasmid sets. \( r = 10^5, m = 0.10 \) The x-axis represents the label predicted by the algorithm and the y-axis represents the true label.

with 10% error from the individual carbapenem resistance identification are 0.50% for LDA, 1.00% for ET, 1.25% for NN, 2.50% for PA, and 4.75% for GNB. The false-negative rates at \( 10^5 \) BOC reads with 10% error from the grouped carbapenem resistance identification are 2.6% for LDA, 4.5% for ET, 5.4% for NN, 5.9% for GNB, and 8.2% for PA. Thus, the LDA MLA appears best at avoiding false-negative errors when identifying resistance genes on plasmids, while the ET most accurately identifies the resistance genes.
C.3 Discussion

This research is the first of its kind and validates the SERS-BOC instrument (currently in development) as an excellent predictor of genetic signatures, even in the presence of genetic mutations and experimental noise. The use of FBC deviation spectra was able to achieve greater than 99% accuracy in classifying bacterial species using several types of MLAs and only $10^4$ BOC reads, even in the presence of mutations or experimental noise.

We were able to detect individual antibiotic resistance genes on plasmids from their FBC deviation spectra with 80% accuracy and to detect the presence of a carbapenem resistance gene with 95% accuracy using MLAs with $10^5$ BOC reads. The implications of our research suggest that the use of MLA classifiers on fractional base composition data generated from a SERS-BOC type instrument [363] has tremendous potential in accurately identifying both bacterial species and antibiotic resistance genes. With respect to bloodstream infection diagnosis, the creation of FBC models has the potential to help determine the bacterial species and the antibiotic-resistance profile associated with a bloodstream infection in a cost-efficient and time-efficient way, thus improving the outcomes for patients. Of most import, the false negative evaluations for carbapenem resistance genes were less than 3% using the LDA algorithm with $10^5$ BOC reads.

We note with surprise the lack of spread within a species data cluster in PCA analysis even when introducing a 25% error rate. This indicates that the inclusion of error does not introduce enough variance to cause these species to have overlap in the principal component space until the number of BOC reads get low ($10^2$ and $10^3$ reads). Because we are sampling at least a million different $k$-mers to generate the FBC spectrum, the distribution of any single training sample does not deviate significantly from the FBC of the non-mutated genome. The high number of reads available on a 1-million-pyramid SERS array would enable this high predictive ability, and may be more than necessary. Even with experimental error causing deviation from the reference FBC, the FBCs of different species are distinct enough that the simulated experimental noise had no negative effect on clustering in principal component space. The random error in BOC data from genomic mutation and experimental noise in the FBC spectra are subtracted out since we know what the random spectrum looks like. Thus, random errors reduce the signal levels but have little effect on noise levels.

The MLA analysis shows how robust the species model is at handling experimental noise in the FBC, allowing for substantial error rates while maintaining its predictive power. Again, one reason why the model can still accurately classify species even with high error rates is that the experimental noise mimics the random bias distribution. Since we subtract out this random bias before running the machine learning
classifier, we are essentially “subtracting out” the effects of the mutation or noise, and what is left is a smaller size of the accurate FBC, which the MLA can still use to accurately classify the species of bacteria. For example, $10^6$ BOC reads with 90% random errors still leaves $10^5$ good reads once the randomness is subtracted.

This study challenged the algorithms with much more random error in the data than expected in actual experiments. The largest source of error is anticipated to be wrong assignment of the BOC values due to spectroscopic noise. Currently the accuracy of calling correct BOC is about 90% [363], but accuracy is improving with time and experience and optical quality. Thus 25% error is probably a gross overestimate of actual experimental error, which we anticipate will be on the order of 10% or less. Random point mutations in bacteria are far less than 0.1% and have no bearing on the accuracy of species identification.

Plasmids come in all different sizes ranging from 2,000 to 200,000 bp long (and longer), and the carbapenem-resistant genes are about 800 bp. This means that the FBC that comes from the resistant genes has varying weight upon the FBC spectrum of the entire plasmid, which would explain why there is so much spread from one resistant plasmid to another in the PCA analysis. The ability for the MLAs to pick up on that 800-bp region amongst all of the other data and noise shows great promise for this type of antibiotic resistance gene identification. Overall, the never-before-seen results show that the good performance of the models is not a result of overfitting; rather, the models are actually learning how to recognize the characteristic signatures in an FBC deviation spectrum when a specific target gene is included. Of course the application of the algorithm for antibiotic resistance identification requires knowledge of an antibiotic resistance gene sequence, so this technique would not identify novel antibiotic resistance evolved by mutations.

In a real-world application, these models would need to be tuned to perform optimally, but the low sensitivity of the accuracy to error rates ranging from 0% to 90% proves the robustness of the models in dealing with noise. As previously stated, we postulate that this robustness stems from the subtraction of randomness from within the data set. A high number of BOC reads with high random error is similar to a model having less SERS-BOC readings, which our results show is still effective down to the $10^4$ BOC reads. Additional sources of error in a real-world setting could include DNA contamination (human or bacterial), lack of good separation between plasmids and genomes (which would make the resistant genes on plasmids less distinguishable), and untested or new bacterial species causing the blood infections. We anticipate the continued development of the SERS-BOC instrument that will provide real BOC data from which FBC data can be created to further test our assumptions and postulates. Further application of this method could go beyond analyzing bacterial bloodstream infections into other clinical scenarios that need fast and reliable analysis of only a few key genes of known sequence.
There are known limitations to this study. The first and foremost is that all of the data had to be simulated because a large enough device to produce experimental data is still being developed. Another limitation is that only a single group of antibiotic resistance genes (carbapenem resistant genes) was tested. Therefore, the results cannot be broadcasted to other groups of resistance genes or to identifying other genes. Also, this technique can only identify antibiotic resistance through known DNA sequences. Another limitation is the limited number of bacterial species used. Because of the focus on testing the technology for clinical use in bloodstream infections, a select group of bacterial species was used for this initial study. Future studies will test the ability of this technique to identify between all species. This study also did not tune the MLAs or perform a cost-benefit analysis to identify the factors influencing the accuracy of the MLAs. However, this study provides the initial groundwork for further exploration in using a SERS-BOC device to identify species and genes based on 10-mer DNA sequences.

I would like to thank Dr. Mark Clement and Dr. Prashant Nagpal for their contributions to this appendix. Dr. Mark Clement made contributions to the DNA identification simulations by helping us understand how to use the machine learning algorithms as well as the development of how to present the data. Dr. Prashant Nagpal made contributions to the DNA identification simulations in helping understand how the DNA is analyzed using SERS and the data that results from the SERS reading.

C.4 Additional Material

C.4.1 100 BOC reads

In creating the random bias, there are 4k possible sequential \( k \)-mers, which for \( k=10 \) gives 1,048,576 \( k \)-mers. When we get down to 100 and 1000 BOC reads, we are greatly under-sampling the bias FBC. Therefore, in order to ensure that the bias remains true to the bias sequence, we checked the bias by sampling 100 different random 100 BOC reads (see Supplemental Figure C.12) and compared them against the bias FBC (see Supplemental Figure C.13). Then we created an FBC from the 100 random samples by summing the bins for the 100 samples and compared that FBC to the bias FBC (see Supplemental Figure C.14). From Supplemental Figure C.12, we see that no one sample covers the entire FBC space, but collectively they begin to sample the entire space. From Supplemental Figure C.13, we see that the individual samples do not match the bias FBC. It is higher in some regions and lower in others. But from Supplemental Figure C.14, the summed FBC from the 100 samples matches the bias FBC.
Figure C.12: Ten of the 100 randomly sampled 100 BOC reads from the bias FBC.

Figure C.13: Four of the 100 randomly sampled 100 BOC reads from the bias FBC compared to the bias FBC.
Figure C.14: The FBC from summing the 100 randomly sampled 100 BOC reads from the bias FBC compared to the bias FBC.

C.4.2 Plasmid NCBI Reference IDs

Table C.6: NCBI Reference IDs for the antibiotic resistant plasmids.

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_019368.1</td>
<td>NC_011382.1</td>
<td>NC_015872.1</td>
<td>NC_014368.1</td>
</tr>
<tr>
<td>NC_022652.1</td>
<td>NC_011383.1</td>
<td>NC_016974.1</td>
<td>NC_019375.1</td>
</tr>
<tr>
<td>NG_049203.1</td>
<td>NC_014312.1</td>
<td>NC_016980.1</td>
<td>NC_019906.1</td>
</tr>
<tr>
<td>NZ_CP016763.1</td>
<td>NC_016846.1</td>
<td>NC_018994.1</td>
<td>NC_021576.1</td>
</tr>
<tr>
<td>NZ_CP021328.1</td>
<td>NC_019152.1</td>
<td>NC_019045.2</td>
<td>NC_021622.1</td>
</tr>
<tr>
<td>NZ_CP022533.1</td>
<td>NC_019161.1</td>
<td>NC_019063.1</td>
<td>NC_022242.1</td>
</tr>
<tr>
<td>NZ_CP024192.1</td>
<td>NC_019161.1</td>
<td>NC_019069.1</td>
<td>NC_024987.1</td>
</tr>
<tr>
<td>NZ_CP024522.1</td>
<td>NC_019384.1</td>
<td>NC_019089.1</td>
<td>NC_025186.1</td>
</tr>
<tr>
<td>NZ_CP024529.1</td>
<td>NC_020132.1</td>
<td>NC_019153.1</td>
<td>NC_032101.1</td>
</tr>
<tr>
<td>NZ_CP024557.1</td>
<td>NC_020893.1</td>
<td>NC_019158.1</td>
<td>NG_050336.1</td>
</tr>
<tr>
<td>NZ_CP025964.2</td>
<td>NC_021238.1</td>
<td>NC_019162.1</td>
<td>NZ_ATHX01000046.1</td>
</tr>
</tbody>
</table>
Table C.6: (continued)

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZ_CP028486.1</td>
<td>NC_021660.2</td>
<td>NC_019163.1</td>
<td>NZ_CM010663.1</td>
</tr>
<tr>
<td>NZ_CP028952.1</td>
<td>NC_021664.2</td>
<td>NC_019268.1</td>
<td>NZ_CP023419.1</td>
</tr>
<tr>
<td>NZ_CP031121.1</td>
<td>NC_022078.1</td>
<td>NC_019281.1</td>
<td>NZ_CP023442.1</td>
</tr>
<tr>
<td>NZ_JQNT01000104.1</td>
<td>NC_022346.1</td>
<td>NC_019360.1</td>
<td>NZ_CP023926.1</td>
</tr>
<tr>
<td>NZ_JRFQ01000062.1</td>
<td>NC_022885.1</td>
<td>NC_019889.1</td>
<td>NZ_CP025467.1</td>
</tr>
<tr>
<td>NZ_JTEB01000058.1</td>
<td>NC_023331.1</td>
<td>NC_019985.2</td>
<td>NZ_CP026661.1</td>
</tr>
<tr>
<td>NZ_JTGO01000050.1</td>
<td>NC_023903.1</td>
<td>NC_020552.1</td>
<td>NZ_CP029718.1</td>
</tr>
<tr>
<td>NZ_JUBD01000040.1</td>
<td>NC_023904.1</td>
<td>NC_020811.1</td>
<td>NZ_CP030081.1</td>
</tr>
<tr>
<td>NZ_JUBE01000054.1</td>
<td>NC_023905.1</td>
<td>NC_020818.1</td>
<td>NZ_CP031570.1</td>
</tr>
<tr>
<td>NZ_JUGC01000052.1</td>
<td>NC_023906.1</td>
<td>NC_021180.1</td>
<td>NZ_CP031573.1</td>
</tr>
<tr>
<td>NZ_KQ089505.1</td>
<td>NC_023907.1</td>
<td>NC_021198.1</td>
<td>NZ_CP031576.1</td>
</tr>
<tr>
<td>NZ_LAIY01000158.1</td>
<td>NC_024967.1</td>
<td>NC_021501.1</td>
<td>NZ_CP031584.1</td>
</tr>
<tr>
<td>NZ_LFXT01000042.1</td>
<td>NC_025019.1</td>
<td>NC_022589.1</td>
<td>NZ_CP031610.1</td>
</tr>
<tr>
<td>NZ_LFXU01000060.1</td>
<td>NC_025166.1</td>
<td>NC_022609.1</td>
<td>NZ_CP032168.1</td>
</tr>
<tr>
<td>NZ_LRJV01000033.1</td>
<td>NC_025167.1</td>
<td>NC_023322.1</td>
<td>NZ_CP032176.1</td>
</tr>
<tr>
<td>NZ_LRJW01000124.1</td>
<td>NC_025183.1</td>
<td>NC_023908.1</td>
<td>NZ_CP032178.1</td>
</tr>
<tr>
<td>NZ_LRJX01000101.1</td>
<td>NC_025187.1</td>
<td>NC_023911.1</td>
<td>NZ_CP034084.1</td>
</tr>
<tr>
<td>NZ_NGAG01000014.1</td>
<td>NC_032098.1</td>
<td>NC_023914.1</td>
<td>NZ_FWYI01000067.1</td>
</tr>
<tr>
<td>NZ_NGAH01000052.1</td>
<td>NC_032103.1</td>
<td>NC_024954.1</td>
<td>NZ_JYGA02000003.1</td>
</tr>
<tr>
<td>NZ_NGAI01000045.1</td>
<td>NG_049253.1</td>
<td>NC_024959.1</td>
<td>NZ_LIDY01000091.1</td>
</tr>
<tr>
<td>NZ_NGAJ01000013.1</td>
<td>NZ_ANGI02000157.1</td>
<td>NC_024999.1</td>
<td>NZ_LLWB01000129.1</td>
</tr>
<tr>
<td>NZ_NGAK01000052.1</td>
<td>NZ_AOGQ01000049.1</td>
<td>NC_025000.1</td>
<td>NZ_LNG0O10000036.1</td>
</tr>
<tr>
<td>NZ_NGAL01000047.1</td>
<td>NZ_AP018674.1</td>
<td>NC_025106.1</td>
<td>NZ_LNHV01000004.1</td>
</tr>
<tr>
<td>NZ_NII01000041.1</td>
<td>NZ_APVO01000006.1</td>
<td>NC_025116.1</td>
<td>NZ_LPOA01000052.1</td>
</tr>
<tr>
<td>NZ_NIIJ01000054.1</td>
<td>NZ_APVP01000035.1</td>
<td>NC_025130.1</td>
<td>NZ_LPOE01000075.1</td>
</tr>
<tr>
<td>NZ_NIJK01000054.1</td>
<td>NZ_APVQ01000147.1</td>
<td>NC_025184.1</td>
<td>NZ_LPOF01000059.1</td>
</tr>
<tr>
<td>NZ_NIIR01000049.1</td>
<td>NZ_APVR01000051.1</td>
<td>NG_049326.1</td>
<td>NZ_LPOH01000021.1</td>
</tr>
<tr>
<td>NZ_NPGF01000059.1</td>
<td>NZ_APVS01000045.1</td>
<td>NZ_AGFH01000030.1</td>
<td>NZ_LPOI01000051.1</td>
</tr>
<tr>
<td>NZ_NWEC01000032.1</td>
<td>NZ_APVT01000059.1</td>
<td>NZ_AOCV01000085.1</td>
<td>NZ_LPOQ01000121.1</td>
</tr>
</tbody>
</table>
Table C.6: (continued)

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZ_PCFT01000003.1</td>
<td>NZ_APVV01000098.1</td>
<td>NZ_AP014649.1</td>
<td>NZ_LPAA01000096.1</td>
</tr>
<tr>
<td>NZ_PTXQ01000049.1</td>
<td>NZ_APVY01000038.1</td>
<td>NZ_AP018750.1</td>
<td>NZ_LPPL01000133.1</td>
</tr>
<tr>
<td>NZ_PWAQ01000079.1</td>
<td>NZ_APVZ01000026.1</td>
<td>NZ_AP018753.1</td>
<td>NZ_LPPN01000057.1</td>
</tr>
<tr>
<td>NZ_PWAR01000086.1</td>
<td>NZ_APWA01000117.1</td>
<td>NZ_AVAN01000031.1</td>
<td>NZ_LPPS01000099.1</td>
</tr>
<tr>
<td>NZ_PWAS01000080.1</td>
<td>NZ_APWB01000186.1</td>
<td>NZ_BDHK01000043.1</td>
<td>NZ_LPPT01000038.1</td>
</tr>
<tr>
<td>NZ_RDRD01000120.1</td>
<td>NZ_APWD01000124.1</td>
<td>NZ_BFCJ01000033.1</td>
<td>NZ_LPQA01000052.1</td>
</tr>
<tr>
<td>NZ_RXPP01000071.1</td>
<td>NZ_APWF01000060.1</td>
<td>NZ_BFCK01000011.1</td>
<td>NZ_LPQK01000015.1</td>
</tr>
<tr>
<td>NZ_RXPQ01000059.1</td>
<td>NZ_APWG01000101.1</td>
<td>NZ_BFCL01000054.1</td>
<td>NZ_LPQM01000037.1</td>
</tr>
<tr>
<td>NZ_RXPS01000070.1</td>
<td>NZ_APWJ01000039.1</td>
<td>NZ_BFCM01000058.1</td>
<td>NZ_LPQP01000039.1</td>
</tr>
<tr>
<td>NZ_RXQD01000071.1</td>
<td>NZ_AQQA01000132.1</td>
<td>NZ_BFCO01000016.1</td>
<td>NZ_LPQQ01000053.1</td>
</tr>
<tr>
<td>NC_019368.1</td>
<td>NZ_AQRD01000004.1</td>
<td>NZ_BFCQ01000022.1</td>
<td>NZ_LPQR01000048.1</td>
</tr>
<tr>
<td>NC_022652.1</td>
<td>NZ_ARPR01000194.1</td>
<td>NZ_BFCR01000070.1</td>
<td>NZ_LPQ01000029.1</td>
</tr>
<tr>
<td>CP016763.1</td>
<td>NZ_ARRU01000101.1</td>
<td>NZ_BFCS01000096.1</td>
<td>NZ_LPQW01000035.1</td>
</tr>
<tr>
<td>AF244145.1</td>
<td>NZ_ARRV01000080.1</td>
<td>NZ_BFCT01000116.1</td>
<td>NZ_LRJI01000043.1</td>
</tr>
<tr>
<td>AF445082.1</td>
<td>NZ_ARRZ01000061.1</td>
<td>NZ_BFCU01000014.1</td>
<td>NZ_LRJ01000086.1</td>
</tr>
<tr>
<td>AY590475.1</td>
<td>NZ_ARSI01000101.1</td>
<td>NZ_BFCY01000147.1</td>
<td>NZ_LRJL01000082.1</td>
</tr>
<tr>
<td>AY795963.1</td>
<td>NZ_ARSJ01000057.1</td>
<td>NZ_BIFN01000180.1</td>
<td>NZ_LRJM01000044.1</td>
</tr>
<tr>
<td>DQ532122.1</td>
<td>NZ_ARSN01000117.1</td>
<td>NZ_BIFO01000144.1</td>
<td>NZ_LRJN01000051.1</td>
</tr>
<tr>
<td>JQ818252.1</td>
<td>NZ_ARSO01000220.1</td>
<td>NZ_CBYG01000041.1</td>
<td>NZ_LRJP01000075.1</td>
</tr>
<tr>
<td>JX101693.1</td>
<td>NZ_ARSP01000198.1</td>
<td>NZ_CCFB01000005.1</td>
<td>NZ_LXMI01000062.1</td>
</tr>
<tr>
<td>KF250428.1</td>
<td>NZ_ARSQ01000087.1</td>
<td>NZ_CDEL01000093.1</td>
<td>NZ_MCOM01000077.1</td>
</tr>
<tr>
<td>HQ419285.1</td>
<td>NZ_ARVM01000006.1</td>
<td>NZ_CDNA01000040.1</td>
<td>NZ_MCON01000196.1</td>
</tr>
<tr>
<td>KM977631.1</td>
<td>NZ_ARVP01000148.1</td>
<td>NZ_CDQG01000062.1</td>
<td>NZ_MCOO01000183.1</td>
</tr>
<tr>
<td>KF680003.1</td>
<td>NZ_ARVQ01000146.1</td>
<td>NZ_CGHE01000020.1</td>
<td>NZ_MCO01000210.1</td>
</tr>
<tr>
<td>KT989376.1</td>
<td>NZ_ATAK01000080.1</td>
<td>NZ_CM001802.1</td>
<td>NZ_MCOQ01000136.1</td>
</tr>
<tr>
<td>KT982613.1</td>
<td>NZ_ATEZ01000138.1</td>
<td>NZ_CM008471.1</td>
<td>NZ_MDCW01000200.1</td>
</tr>
<tr>
<td>KT982618.1</td>
<td>NZ_AXLE01000003.1</td>
<td>NZ_CM008823.1</td>
<td>NZ_MDCX01000052.1</td>
</tr>
<tr>
<td>KT982615.1</td>
<td>NZ_AXLI01000004.1</td>
<td>NZ_CM008879.1</td>
<td>NZ_MDCA01000044.1</td>
</tr>
<tr>
<td>KT982615.1</td>
<td>NZ_AXLI01000051.1</td>
<td>NZ_CM008882.1</td>
<td>NZ_MDCZ01000168.1</td>
</tr>
<tr>
<td>IMP-4</td>
<td>KPC-2</td>
<td>NDM-1</td>
<td>VIM-1</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>KT982616.1</td>
<td>NZ_AZAP01000068.1</td>
<td>NZ_CM008884.1</td>
<td>NZ_MDDA01000130.1</td>
</tr>
<tr>
<td>KT982617.1</td>
<td>NZ_BFCA01000004.1</td>
<td>NZ_CM008885.1</td>
<td>NZ_MLFV01000055.1</td>
</tr>
<tr>
<td>KT989598.1</td>
<td>NZ_BFCN01000044.1</td>
<td>NZ_CM008891.1</td>
<td>NZ_MLFW01000070.1</td>
</tr>
<tr>
<td>KT989599.1</td>
<td>NZ_BFCP01000043.1</td>
<td>NZ_CM008895.1</td>
<td>NZ_MSAA01000028.1</td>
</tr>
<tr>
<td>KU051707.1</td>
<td>NZ_BFCX01000068.1</td>
<td>NZ_CM008904.1</td>
<td>NZ_NFFC01000079.1</td>
</tr>
<tr>
<td>KU051708.1</td>
<td>NZ_BFDC01000027.1</td>
<td>NZ_CM008917.1</td>
<td>NZ_NFFG01000090.1</td>
</tr>
<tr>
<td>KU051709.1</td>
<td>NZ_BFDD01000065.1</td>
<td>NZ_CM012324.1</td>
<td>NZ_NFFO01000062.1</td>
</tr>
<tr>
<td>KU051710.1</td>
<td>NZ_BFDE01000174.1</td>
<td>NZ_CM012324.1</td>
<td>NZ_NFFU01000013.1</td>
</tr>
<tr>
<td>KU886034.1</td>
<td>NZ_BFDF01000149.1</td>
<td>NZ_CP006661.1</td>
<td>NZ_NFFV01000038.1</td>
</tr>
<tr>
<td>KX711879.1</td>
<td>NZ_BFDG01000080.1</td>
<td>NZ_CP006799.1</td>
<td>NZ_NFFX01000040.1</td>
</tr>
<tr>
<td>KX711880.1</td>
<td>NZ_BGLK01000081.1</td>
<td>NZ_CP008933.1</td>
<td>NZ_NFFY01000029.1</td>
</tr>
<tr>
<td>KU726588.1</td>
<td>NZ_BGNU01000071.1</td>
<td>NZ_CP009115.1</td>
<td>NZ_NFFZ01000073.1</td>
</tr>
<tr>
<td>KU862632.1</td>
<td>NZ_BGNV01000079.1</td>
<td>NZ_CP010370.2</td>
<td>NZ_NFGB01000058.1</td>
</tr>
<tr>
<td>KY913900.1</td>
<td>NZ_BGNW01000137.1</td>
<td>NZ_CP010373.2</td>
<td>NZ_NFGC01000059.1</td>
</tr>
<tr>
<td>KY887590.1</td>
<td>NZ_BGNX01000052.1</td>
<td>NZ_CP010391.1</td>
<td>NZ_NFGM01000055.1</td>
</tr>
<tr>
<td>KY887595.1</td>
<td>NZ_BGNY01000121.1</td>
<td>NZ_CP010399.1</td>
<td>NZ_NFGN01000084.1</td>
</tr>
<tr>
<td>KY887596.1</td>
<td>NZ_BGNZ01000083.1</td>
<td>NZ_CP011839.1</td>
<td>NZ_NFGU01000072.1</td>
</tr>
<tr>
<td>KY887591.1</td>
<td>NZ_BGOA01000106.1</td>
<td>NZ_CP012754.1</td>
<td>NZ_NFGV01000057.1</td>
</tr>
<tr>
<td>CP021328.1</td>
<td>NZ_BGBO01000073.1</td>
<td>NZ_CP012901.1</td>
<td>NZ_NPIX01000118.1</td>
</tr>
<tr>
<td>CP022533.1</td>
<td>NZ_BGOC01000031.1</td>
<td>NZ_CP013483.1</td>
<td>NZ_NSMB01000103.1</td>
</tr>
<tr>
<td>CP024192.1</td>
<td>NZ_CCFQ01000037.1</td>
<td>NZ_CP014478.1</td>
<td>NZ_OOHS01000036.1</td>
</tr>
<tr>
<td>CP024522.1</td>
<td>NZ_CCFR01000024.1</td>
<td>NZ_CP014757.1</td>
<td>NZ_OOHT01000065.1</td>
</tr>
<tr>
<td>CP024529.1</td>
<td>NZ_CCFT01000028.1</td>
<td>NZ_CP015835.1</td>
<td>NZ_PDDP01000062.1</td>
</tr>
<tr>
<td>CP024557.1</td>
<td>NZ_CCGC01000012.1</td>
<td>NZ_CP016035.1</td>
<td>NZ_PDLU01000121.1</td>
</tr>
<tr>
<td>CP025964.2</td>
<td>NZ_CCGH01000106.1</td>
<td>NZ_CP016921.1</td>
<td>NZ_PEHD01000103.1</td>
</tr>
<tr>
<td>KY884003.1</td>
<td>NZ_CCHA01000013.1</td>
<td>NZ_CP017672.1</td>
<td>NZ_PUTD01000036.1</td>
</tr>
<tr>
<td>MF344557.1</td>
<td>NZ_CCHD01000041.1</td>
<td>NZ_CP018366.1</td>
<td>NZPUTE01000036.1</td>
</tr>
<tr>
<td>MF344559.1</td>
<td>NZ_CCHE01000047.1</td>
<td>NZ_CP018817.1</td>
<td>NZ_PUTF01000036.1</td>
</tr>
<tr>
<td>MF344561.1</td>
<td>NZ_CCHF01000083.1</td>
<td>NZ_CP019053.1</td>
<td>NZ_PXKU01000056.1</td>
</tr>
</tbody>
</table>
Table C.6: (continued)

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF344562.1</td>
<td>NZ_CCHS01000075.1</td>
<td>NZ_CP020049.1</td>
<td>NZ_QDDT01000068.1</td>
</tr>
<tr>
<td>MF344564.1</td>
<td>NZ_CCHT01000060.1</td>
<td>NZ_CP020056.1</td>
<td>NZ_QDDU01000072.1</td>
</tr>
<tr>
<td>MF344565.1</td>
<td>NZ_CCHU01000057.1</td>
<td>NZ_CP020068.1</td>
<td>NZ_QDDW01000067.1</td>
</tr>
<tr>
<td>MF344567.1</td>
<td>NZ_CCHW01000023.1</td>
<td>NZ_CP020090.1</td>
<td>NZ_QDDX01000055.1</td>
</tr>
<tr>
<td>MG516907.1</td>
<td>NZ_CDFP0100006.1</td>
<td>NZ_CP020703.1</td>
<td>NZ_QFDS01000057.1</td>
</tr>
<tr>
<td>MG516908.1</td>
<td>NZ_CDFR01000233.1</td>
<td>NZ_CP020704.1</td>
<td>NZ_QKNQ01000003.1</td>
</tr>
<tr>
<td>MG516909.1</td>
<td>NZ_CM010658.1</td>
<td>NZ_CP020848.1</td>
<td>NZ_QKRX01000089.1</td>
</tr>
<tr>
<td>MG516910.1</td>
<td>NZ_CM010665.1</td>
<td>NZ_CP020854.1</td>
<td>NZ_QLNR01000077.1</td>
</tr>
<tr>
<td>MG516912.1</td>
<td>NZ_CP007558.1</td>
<td>NZ_CP021206.1</td>
<td>NZ_QWWE01000104.1</td>
</tr>
<tr>
<td>MF072962.1</td>
<td>NZ_CP007732.1</td>
<td>NZ_CP021345.1</td>
<td>NZ_RDWT01000101.1</td>
</tr>
<tr>
<td>CP027603.1</td>
<td>NZ_CP008791.1</td>
<td>NZ_CP021345.1</td>
<td>NZ_RYX01000077.1</td>
</tr>
<tr>
<td>CP028952.1</td>
<td>NZ_CP008798.1</td>
<td>NZ_CP021536.1</td>
<td>NZ_UISC01000060.1</td>
</tr>
<tr>
<td>CP028486.1</td>
<td>NZ_CP008824.1</td>
<td>NZ_CP021551.1</td>
<td>NZ_UISE01000058.1</td>
</tr>
<tr>
<td>CP031121.1</td>
<td>NZ_CP008833.1</td>
<td>NZ_CP021687.1</td>
<td>NZ_UISK01000113.1</td>
</tr>
<tr>
<td>MH727565.1</td>
<td>NZ_CP008901.1</td>
<td>NZ_CP021699.1</td>
<td>NZ_UITI01000068.1</td>
</tr>
<tr>
<td>MH824848.1</td>
<td>NZ_CP008901.1</td>
<td>NZ_CP021709.1</td>
<td>NZ_UIVF01000087.1</td>
</tr>
<tr>
<td>MH829594.1</td>
<td>NZ_CP009853.1</td>
<td>NZ_CP021720.1</td>
<td>NZ_UIXR01000051.1</td>
</tr>
<tr>
<td>AB636651.1</td>
<td>NZ_CP009858.1</td>
<td>NZ_CP021936.1</td>
<td>NZ_UIXU01000060.1</td>
</tr>
<tr>
<td>LC107424.1</td>
<td>NZ_CP009862.1</td>
<td>NZ_CP021941.1</td>
<td>NZ_UJAY01000132.1</td>
</tr>
<tr>
<td>LC107606.1</td>
<td>NZ_CP009867.1</td>
<td>NZ_CP021947.1</td>
<td>NZ_UJHZ01000086.1</td>
</tr>
<tr>
<td>LC198842.1</td>
<td>NZ_CP009875.1</td>
<td>NZ_CP021952.1</td>
<td>NZ_UJIF01000084.1</td>
</tr>
<tr>
<td>AJ609296.3</td>
<td>NZ_CP009876.1</td>
<td>NZ_CP021961.1</td>
<td>NZ_UJKO01000087.1</td>
</tr>
<tr>
<td>KQ089505.1</td>
<td>NZ_CP009881.1</td>
<td>NZ_CP021962.1</td>
<td>NZ_UJME01000055.1</td>
</tr>
<tr>
<td>NIII01000041.1</td>
<td>NZ_CP010362.1</td>
<td>NZ_CP022126.1</td>
<td>NZ_UJML01000088.1</td>
</tr>
<tr>
<td>NIIIR01000049.1</td>
<td>NZ_CP010374.1</td>
<td>NZ_CP022350.1</td>
<td>NZ_UJWL01000026.1</td>
</tr>
<tr>
<td>NPGF01000059.1</td>
<td>NZ_CP010379.1</td>
<td>NZ_CP022612.1</td>
<td>NZ_UJWM01000030.1</td>
</tr>
<tr>
<td>NGAG01000014.1</td>
<td>NZ_CP010396.1</td>
<td>NZ_CP023187.1</td>
<td>NZ_UJWN01000027.1</td>
</tr>
<tr>
<td>NGAL01000047.1</td>
<td>NZ_CP010575.1</td>
<td>NZ_CP023274.1</td>
<td>NZ_UJWQ01000026.1</td>
</tr>
</tbody>
</table>

272
Table C.6: (continued)

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGAJ01000013.1</td>
<td>NZ_CP011571.1</td>
<td>NZ_CP023488.1</td>
<td>NZ_UJXD01000027.1</td>
</tr>
<tr>
<td>PCFT01000003.1</td>
<td>NZ_CP011573.1</td>
<td>NZ_CP023554.1</td>
<td>NZ_UKNV01000033.1</td>
</tr>
<tr>
<td>RDRD01000120.1</td>
<td>NZ_CP011580.1</td>
<td>NZ_CP023910.1</td>
<td>NZ_UKRC01000036.1</td>
</tr>
<tr>
<td>RXQD01000071.1</td>
<td>NZ_CP011582.1</td>
<td>NZ_CP023914.1</td>
<td>NZ_UKXI01000027.1</td>
</tr>
<tr>
<td>RXPP01000071.1</td>
<td>NZ_CP011595.1</td>
<td>NZ_CP023923.1</td>
<td>NZ_UKXU01000032.1</td>
</tr>
<tr>
<td>RXPS01000070.1</td>
<td>NZ_CP011598.1</td>
<td>NZ_CP023948.1</td>
<td>NZ_UKXU01000032.1</td>
</tr>
<tr>
<td>NZ_CP011608.1</td>
<td>NZ_CP024039.1</td>
<td>NZ_ULAU01000072.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011611.1</td>
<td>NZ_CP024814.1</td>
<td>NZ_ULAA01000085.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011615.1</td>
<td>NZ_CP024828.1</td>
<td>NZ_ULAU01000072.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011622.1</td>
<td>NZ_CP024879.1</td>
<td>NZ_ULAU01000072.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011635.1</td>
<td>NZ_CP026127.1</td>
<td>NZ_ULAW01000053.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011639.1</td>
<td>NZ_CP026127.1</td>
<td>NZ_ULDL01000044.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011649.1</td>
<td>NZ_CP026127.1</td>
<td>NZ_ULDM01000044.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011656.1</td>
<td>NZ_CP026590.1</td>
<td>NZ_ULDP01000090.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011661.1</td>
<td>NZ_CP026617.1</td>
<td>NZ_ULDS01000088.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011981.1</td>
<td>NZ_CP027177.1</td>
<td>NZ_ULEA01000084.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011986.1</td>
<td>NZ_CP027528.1</td>
<td>NZ_UWWA01000041.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP011991.1</td>
<td>NZ_CP027532.1</td>
<td>NZ_UWXE01000002.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP013325.1</td>
<td>NZ_CP027679.1</td>
<td>NC_014368.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP015347.1</td>
<td>NZ_CP028171.1</td>
<td>NC_019375.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP015382.1</td>
<td>NZ_CP028178.1</td>
<td>NC_021576.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP015395.1</td>
<td>NZ_CP028560.1</td>
<td>NC_021622.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP015990.1</td>
<td>NZ_CP028786.2</td>
<td>NC_022242.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP015991.1</td>
<td>NZ_CP028800.1</td>
<td>NC_024987.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP016381.1</td>
<td>NZ_CP028929.1</td>
<td>NC_025186.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP017083.1</td>
<td>NZ_CP029118.1</td>
<td>NC_032101.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP017086.1</td>
<td>NZ_CP029397.1</td>
<td>AF317511.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP017929.1</td>
<td>NZ_CP029610.1</td>
<td>AY152821.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP017937.1</td>
<td>NZ_CP029631.1</td>
<td>AY339625.2</td>
<td></td>
</tr>
<tr>
<td>IMP-4</td>
<td>KPC-2</td>
<td>NDM-1</td>
<td>VIM-1</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>NZ_CP017993.1</td>
<td>NZ_CP029731.1</td>
<td>AY648125.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018356.1</td>
<td>NZ_CP029737.1</td>
<td>AY970968.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018362.1</td>
<td>NZ_CP030345.1</td>
<td>AY987853.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018426.1</td>
<td>NZ_CP030858.1</td>
<td>DQ112355.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018432.1</td>
<td>NZ_CP031216.1</td>
<td>DQ489717.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018436.1</td>
<td>NZ_CP031297.1</td>
<td>DQ522239.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018455.1</td>
<td>NZ_CP031322.1</td>
<td>EF078697.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018668.1</td>
<td>NZ_CP031716.1</td>
<td>EF093145.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018669.1</td>
<td>NZ_CP031736.1</td>
<td>EF093146.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018884.1</td>
<td>NZ_CP031884.1</td>
<td>EF494743.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018887.1</td>
<td>NZ_CP032132.1</td>
<td>EF577408.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018945.1</td>
<td>NZ_CP032142.1</td>
<td>EF690695.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018949.1</td>
<td>NZ_CP032193.1</td>
<td>EF690696.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018956.1</td>
<td>NZ_CP032278.1</td>
<td>EU118150.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018959.1</td>
<td>NZ_CP032284.1</td>
<td>FJ172675.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018963.1</td>
<td>NZ_CP032284.1</td>
<td>GQ422826.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018968.1</td>
<td>NZ_CP032286.1</td>
<td>GQ422827.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018974.1</td>
<td>NZ_CP033133.1</td>
<td>GQ422828.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018977.1</td>
<td>NZ_CP033439.1</td>
<td>GQ422829.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018989.1</td>
<td>NZ_CP034323.1</td>
<td>GQ161847.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP018999.1</td>
<td>NZ_CP034369.1</td>
<td>GU724868.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP019001.1</td>
<td>NZ_CP034398.1</td>
<td>GU724869.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP019006.1</td>
<td>NZ_CP034403.1</td>
<td>GU724870.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP019017.1</td>
<td>NZ_CP034409.1</td>
<td>GU724871.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP019026.1</td>
<td>NZ_CP035739.1</td>
<td>GU724872.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP020110.1</td>
<td>NZ_CTBD01000091.1</td>
<td>GU585907.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP021328.1</td>
<td>NZ_CVLD01000041.1</td>
<td>HM126016.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP021743.1</td>
<td>NZ_CVLE01000125.1</td>
<td>JQ690541.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP021750.1</td>
<td>NZ_CVLT01000087.1</td>
<td>JQ824049.1</td>
<td></td>
</tr>
</tbody>
</table>
Table C.6: (continued)

<table>
<thead>
<tr>
<th>IMP-4</th>
<th>KPC-2</th>
<th>NDM-1</th>
<th>VIM-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZ_CP021835.1</td>
<td>NZ_CZWB01000007.1</td>
<td>CP003739.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP021853.1</td>
<td>NZ_FKLP01000165.1</td>
<td>KC140564.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022275.1</td>
<td>NZ_FLXD01000026.1</td>
<td>KC417377.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022276.1</td>
<td>NZ_FLXG01000027.1</td>
<td>KC417378.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022349.1</td>
<td>NZ_FNXW01000145.1</td>
<td>KC417379.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022574.1</td>
<td>NZ_FWFB01000125.1</td>
<td>KC430094.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022693.1</td>
<td>NZ_FWWO01000105.1</td>
<td>JX424614.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022698.1</td>
<td>NZ_FXMF01000024.1</td>
<td>KF856617.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022917.1</td>
<td>NZ_FXMG01000023.1</td>
<td>KF850516.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022920.1</td>
<td>NZ_FXMP01000036.1</td>
<td>KF977034.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022923.1</td>
<td>NZ_FXNB01000025.1</td>
<td>KJ547703.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP022926.1</td>
<td>NZ_FXNE01000025.1</td>
<td>KF998104.1</td>
<td></td>
</tr>
<tr>
<td>NZ_CP023186.1</td>
<td>NZ_FXNP01000028.1</td>
<td>KM514484.1</td>
<td></td>
</tr>
</tbody>
</table>

Table C.7: NCBI Reference IDs for the plasmids without a resistance.

<table>
<thead>
<tr>
<th>Non-resistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_018988.1</td>
</tr>
<tr>
<td>NC_013379.2</td>
</tr>
<tr>
<td>NC_013393.1</td>
</tr>
<tr>
<td>NC_013383.1</td>
</tr>
<tr>
<td>NC_013313.1</td>
</tr>
<tr>
<td>NC_018997.1</td>
</tr>
<tr>
<td>NC_019005.1</td>
</tr>
<tr>
<td>NC_013391.1</td>
</tr>
<tr>
<td>NC_018965.1</td>
</tr>
<tr>
<td>NC_013341.1</td>
</tr>
<tr>
<td>NC_013305.1</td>
</tr>
<tr>
<td>Non-resistant</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>NC_013307.1 NC_010427.1 NC_007773.1</td>
</tr>
<tr>
<td>NC_013345.1 NC_009781.1 NC_002776.1</td>
</tr>
<tr>
<td>NC_013346.1 NC_002147.1 NC_003969.1</td>
</tr>
<tr>
<td>NC_013308.1 NC_010021.1 NC_011227.1</td>
</tr>
<tr>
<td>NC_013312.1 NC_010077.1 NC_012220.1</td>
</tr>
<tr>
<td>NC_013292.1 NC_010099.1 NC_001763.1</td>
</tr>
<tr>
<td>NC_013293.1 NC_003099.1 NC_004334.1</td>
</tr>
<tr>
<td>NC_013299.1 NC_008821.1 NC_004653.1</td>
</tr>
<tr>
<td>NC_013301.1 NC_001399.1 NC_011409.1</td>
</tr>
<tr>
<td>NC_019008.1 NC_005570.1 NC_007206.1</td>
</tr>
<tr>
<td>NC_013340.1 NC_006868.1 NC_008486.1</td>
</tr>
<tr>
<td>NC_019009.1 NC_006904.1 NC_014233.1</td>
</tr>
<tr>
<td>NC_013342.1 NC_001272.2 NC_015068.1</td>
</tr>
<tr>
<td>NC_013344.1 NC_006399.1 NC_015849.1</td>
</tr>
<tr>
<td>NC_018982.1 NC_010616.1 NC_015860.1</td>
</tr>
<tr>
<td>NC_018983.1 NC_010684.1 NC_015861.1</td>
</tr>
<tr>
<td>NC_018984.1 NC_010685.1 NC_015863.1</td>
</tr>
<tr>
<td>NC_018986.1 NC_010686.1 NC_015864.1</td>
</tr>
<tr>
<td>NC_018987.1 NC_010687.1 NC_015862.1</td>
</tr>
<tr>
<td>NC_018998.1 NC_010726.1 NC_015872.1</td>
</tr>
<tr>
<td>NC_013392.2 NC_007635.1 NC_015900.1</td>
</tr>
<tr>
<td>NC_013300.1 NC_004959.1 NC_015901.1</td>
</tr>
<tr>
<td>NC_013395.1 NC_001774.1 NC_015902.1</td>
</tr>
<tr>
<td>NC_019033.1 NC_010851.1 NC_015912.1</td>
</tr>
<tr>
<td>NC_018985.1 NC_010857.1 NC_001446.1</td>
</tr>
<tr>
<td>NC_019040.1 NC_010861.1 NC_009137.1</td>
</tr>
<tr>
<td>NC_019041.1 NC_010862.1 NC_011266.1</td>
</tr>
</tbody>
</table>

276
Table C.7: (continued)

<table>
<thead>
<tr>
<th>Non-resistant</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_011917.1</td>
<td>NC_010870.1</td>
<td>NC_011522.1</td>
</tr>
<tr>
<td>NC_019037.1</td>
<td>NC_010885.1</td>
<td>NC_010098.1</td>
</tr>
<tr>
<td>NC_019039.1</td>
<td>NC_010886.1</td>
<td>NC_004971.1</td>
</tr>
<tr>
<td>NC_019044.1</td>
<td>NC_010898.1</td>
<td>NC_004163.1</td>
</tr>
<tr>
<td>NC_019043.1</td>
<td>NC_010913.1</td>
<td>NC_005247.1</td>
</tr>
<tr>
<td>NC_019053.1</td>
<td>NC_010919.1</td>
<td>NC_006979.1</td>
</tr>
<tr>
<td>NC_019054.1</td>
<td>NC_010929.1</td>
<td>NC_013121.1</td>
</tr>
<tr>
<td>NC_019059.1</td>
<td>NC_010937.1</td>
<td>NC_014369.1</td>
</tr>
<tr>
<td>NC_019060.1</td>
<td>NC_010980.1</td>
<td>NC_014959.1</td>
</tr>
<tr>
<td>NC_019061.1</td>
<td>NC_011101.1</td>
<td>NC_015054.1</td>
</tr>
<tr>
<td>NC_019070.1</td>
<td>NC_011102.1</td>
<td>NC_002123.1</td>
</tr>
<tr>
<td>NC_019080.1</td>
<td>NC_011136.1</td>
<td>NC_010907.1</td>
</tr>
<tr>
<td>NC_019081.1</td>
<td>NC_011139.1</td>
<td>NC_011073.1</td>
</tr>
<tr>
<td>NC_019082.1</td>
<td>NC_011140.1</td>
<td>NC_004771.1</td>
</tr>
<tr>
<td>NC_019083.1</td>
<td>NC_011207.1</td>
<td>NC_011795.1</td>
</tr>
<tr>
<td>NC_019085.1</td>
<td>NC_011228.1</td>
<td>NC_015950.1</td>
</tr>
<tr>
<td>NC_019086.1</td>
<td>NC_011363.1</td>
<td>NC_015972.1</td>
</tr>
<tr>
<td>NC_019087.1</td>
<td>NC_011364.1</td>
<td>NC_016009.1</td>
</tr>
<tr>
<td>NC_019088.1</td>
<td>NC_011378.1</td>
<td>NC_016039.1</td>
</tr>
<tr>
<td>NC_011514.1</td>
<td>NC_005013.1</td>
<td>NC_016053.1</td>
</tr>
<tr>
<td>NC_019092.1</td>
<td>NC_002517.1</td>
<td>NC_016600.1</td>
</tr>
<tr>
<td>NC_019013.1</td>
<td>NC_002120.1</td>
<td>NC_016431.1</td>
</tr>
<tr>
<td>NC_019062.1</td>
<td>NC_001431.1</td>
<td>NC_016849.1</td>
</tr>
<tr>
<td>NC_019065.1</td>
<td>NC_011405.1</td>
<td>NC_016852.1</td>
</tr>
<tr>
<td>NC_019066.1</td>
<td>NC_011406.1</td>
<td>NC_016151.1</td>
</tr>
<tr>
<td>NC_019067.1</td>
<td>NC_011418.1</td>
<td>NC_016966.1</td>
</tr>
<tr>
<td>NC_019068.1</td>
<td>NC_011497.1</td>
<td>NC_016973.1</td>
</tr>
<tr>
<td>Non-resistant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_019071.1</td>
<td>NC_006977.1</td>
<td>NC_016974.1</td>
</tr>
<tr>
<td>NC_019072.1</td>
<td>NC_011521.1</td>
<td>NC_016977.1</td>
</tr>
<tr>
<td>NC_019073.1</td>
<td>NC_011605.1</td>
<td>NC_016979.1</td>
</tr>
<tr>
<td>NC_019074.1</td>
<td>NC_011610.1</td>
<td>NC_016976.1</td>
</tr>
<tr>
<td>NC_019075.1</td>
<td>NC_011617.1</td>
<td>NC_017213.1</td>
</tr>
<tr>
<td>NC_019076.1</td>
<td>NC_011640.1</td>
<td>NC_017197.1</td>
</tr>
<tr>
<td>NC_019077.1</td>
<td>NC_011642.1</td>
<td>NC_017489.1</td>
</tr>
<tr>
<td>NC_019079.1</td>
<td>NC_011752.1</td>
<td>NC_017478.1</td>
</tr>
<tr>
<td>NC_009131.1</td>
<td>NC_011754.1</td>
<td>NC_017772.1</td>
</tr>
<tr>
<td>NC_019093.1</td>
<td>NC_011799.1</td>
<td>NC_005705.1</td>
</tr>
<tr>
<td>NC_019094.1</td>
<td>NC_011812.1</td>
<td>NC_002799.1</td>
</tr>
<tr>
<td>NC_019095.1</td>
<td>NC_011964.1</td>
<td>NC_003101.1</td>
</tr>
<tr>
<td>NC_019096.1</td>
<td>NC_012006.1</td>
<td>NC_004058.1</td>
</tr>
<tr>
<td>NC_019097.1</td>
<td>NC_012031.1</td>
<td>NC_004772.1</td>
</tr>
<tr>
<td>NC_019056.1</td>
<td>NC_012208.1</td>
<td>NC_003201.1</td>
</tr>
<tr>
<td>NC_019057.1</td>
<td>NC_012209.1</td>
<td>NC_004769.1</td>
</tr>
<tr>
<td>NC_019058.1</td>
<td>NC_012215.1</td>
<td>NC_004770.1</td>
</tr>
<tr>
<td>NC_019084.1</td>
<td>NC_012216.1</td>
<td>NC_008444.1</td>
</tr>
<tr>
<td>NC_019078.1</td>
<td>NC_012222.2</td>
<td>NC_008436.1</td>
</tr>
<tr>
<td>NC_009132.1</td>
<td>NC_012487.1</td>
<td>NC_008445.1</td>
</tr>
<tr>
<td>NC_019139.1</td>
<td>NC_012547.1</td>
<td>NC_010290.1</td>
</tr>
<tr>
<td>NC_019140.1</td>
<td>NC_012625.1</td>
<td>NC_010291.1</td>
</tr>
<tr>
<td>NC_019141.1</td>
<td>NC_012626.1</td>
<td>NC_002798.1</td>
</tr>
<tr>
<td>NC_019142.1</td>
<td>NC_012627.1</td>
<td>NC_002150.1</td>
</tr>
<tr>
<td>NC_019143.1</td>
<td>NC_012628.1</td>
<td>NC_005017.1</td>
</tr>
<tr>
<td>NC_019144.1</td>
<td>NC_012630.1</td>
<td>NC_005088.1</td>
</tr>
<tr>
<td>NC_019145.1</td>
<td>NC_012631.1</td>
<td>NC_013506.1</td>
</tr>
</tbody>
</table>
Table C.7: (continued)

<table>
<thead>
<tr>
<th>Non-resistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_019146.1</td>
</tr>
<tr>
<td>NC_019147.1</td>
</tr>
<tr>
<td>NC_019148.1</td>
</tr>
<tr>
<td>NC_019149.1</td>
</tr>
<tr>
<td>NC_019150.1</td>
</tr>
<tr>
<td>NC_018956.1</td>
</tr>
<tr>
<td>NC_018957.1</td>
</tr>
<tr>
<td>NC_018958.1</td>
</tr>
<tr>
<td>NC_006130.1</td>
</tr>
<tr>
<td>NC_004464.2</td>
</tr>
<tr>
<td>NC_006427.1</td>
</tr>
<tr>
<td>NC_001370.1</td>
</tr>
<tr>
<td>NC_002810.1</td>
</tr>
<tr>
<td>NC_004965.1</td>
</tr>
<tr>
<td>NC_000906.2</td>
</tr>
<tr>
<td>NC_005010.1</td>
</tr>
<tr>
<td>NC_004562.1</td>
</tr>
<tr>
<td>NC_002637.1</td>
</tr>
<tr>
<td>NC_005565.1</td>
</tr>
<tr>
<td>NC_006976.1</td>
</tr>
<tr>
<td>NC_002092.1</td>
</tr>
<tr>
<td>NC_002130.1</td>
</tr>
<tr>
<td>NC_006997.1</td>
</tr>
<tr>
<td>NC_005923.1</td>
</tr>
<tr>
<td>NC_005324.1</td>
</tr>
<tr>
<td>NC_004958.1</td>
</tr>
<tr>
<td>NC_006994.1</td>
</tr>
</tbody>
</table>
Table C.7: (continued)

<table>
<thead>
<tr>
<th>Non-resistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_002487.1</td>
</tr>
<tr>
<td>NC_004652.1</td>
</tr>
<tr>
<td>NC_002119.1</td>
</tr>
<tr>
<td>NC_001537.1</td>
</tr>
<tr>
<td>NC_004944.1</td>
</tr>
<tr>
<td>NC_005098.1</td>
</tr>
<tr>
<td>NC_005322.1</td>
</tr>
<tr>
<td>NC_006278.1</td>
</tr>
<tr>
<td>NC_002145.1</td>
</tr>
<tr>
<td>NC_005019.1</td>
</tr>
<tr>
<td>NC_000937.1</td>
</tr>
<tr>
<td>NC_000938.1</td>
</tr>
<tr>
<td>NC_001949.1</td>
</tr>
<tr>
<td>NC_004960.1</td>
</tr>
<tr>
<td>NC_003893.1</td>
</tr>
<tr>
<td>NC_003894.1</td>
</tr>
<tr>
<td>NC_009034.1</td>
</tr>
<tr>
<td>NC_009353.1</td>
</tr>
<tr>
<td>NC_009739.1</td>
</tr>
<tr>
<td>NC_005327.1</td>
</tr>
<tr>
<td>NC_003350.1</td>
</tr>
<tr>
<td>NC_004999.1</td>
</tr>
<tr>
<td>NC_007365.1</td>
</tr>
<tr>
<td>NC_005249.1</td>
</tr>
<tr>
<td>NC_006323.1</td>
</tr>
<tr>
<td>NC_008275.1</td>
</tr>
<tr>
<td>NC_002525.1</td>
</tr>
</tbody>
</table>
Table C.7: (continued)

<table>
<thead>
<tr>
<th>Non-resistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_005211.1</td>
</tr>
<tr>
<td>NC_005205.1</td>
</tr>
<tr>
<td>NC_004564.1</td>
</tr>
<tr>
<td>NC_005024.1</td>
</tr>
<tr>
<td>NC_005250.1</td>
</tr>
<tr>
<td>NC_008357.1</td>
</tr>
<tr>
<td>NC_006671.1</td>
</tr>
<tr>
<td>NC_008460.1</td>
</tr>
<tr>
<td>NC_008490.1</td>
</tr>
<tr>
<td>NC_008613.1</td>
</tr>
<tr>
<td>NC_006277.2</td>
</tr>
<tr>
<td>NC_002377.1</td>
</tr>
<tr>
<td>NC_001911.1</td>
</tr>
<tr>
<td>NC_009139.1</td>
</tr>
<tr>
<td>NC_009435.1</td>
</tr>
<tr>
<td>NC_004998.1</td>
</tr>
<tr>
<td>NC_007675.1</td>
</tr>
<tr>
<td>NC_009602.1</td>
</tr>
<tr>
<td>NC_010076.1</td>
</tr>
<tr>
<td>NC_010111.1</td>
</tr>
<tr>
<td>NC_007772.1</td>
</tr>
<tr>
<td>NC_006828.1</td>
</tr>
<tr>
<td>NC_006829.1</td>
</tr>
<tr>
<td>NC_007928.1</td>
</tr>
<tr>
<td>NC_007931.1</td>
</tr>
<tr>
<td>NC_005018.1</td>
</tr>
<tr>
<td>NC_002144.1</td>
</tr>
</tbody>
</table>
Table C.7: (continued)

<table>
<thead>
<tr>
<th>Non-resistant</th>
<th>NC_002096.1</th>
<th>NC_014312.1</th>
<th>NC_013377.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC_005020.1</td>
<td>NC_014356.1</td>
<td>NC_013380.1</td>
<td></td>
</tr>
<tr>
<td>NC_006843.1</td>
<td>NC_013381.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_001767.1</td>
<td>NC_014382.1</td>
<td>NC_013382.1</td>
<td></td>
</tr>
<tr>
<td>NC_005015.1</td>
<td>NC_013389.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_005567.1</td>
<td>NC_013294.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_001277.1</td>
<td>NC_013295.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_007100.1</td>
<td>NC_013296.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008356.1</td>
<td>NC_013298.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008487.1</td>
<td>NC_013332.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008488.1</td>
<td>NC_013333.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008489.1</td>
<td>NC_013334.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_004981.2</td>
<td>NC_013306.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008597.1</td>
<td>NC_013330.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_004988.1</td>
<td>NC_013307.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_005243.1</td>
<td>NC_013297.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_005564.1</td>
<td>NC_013310.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_002757.1</td>
<td>NC_019010.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_005000.1</td>
<td>NC_013376.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_002013.1</td>
<td>NC_013321.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_008768.1</td>
<td>NC_013329.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_010113.1</td>
<td>NC_013291.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_010114.1</td>
<td>NC_013328.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_010177.2</td>
<td>NC_018977.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_003265.1</td>
<td>NC_00843.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NC_010230.1</td>
<td>NC_00846.1</td>
<td>NC_018981.1</td>
<td></td>
</tr>
</tbody>
</table>

282
C.4.3 Additional PCA Figures

Figure C.15: PCA for genomes at $m = 0$ and $r = 10^2$. Circles (lighter colors) represent the training genome and triangles (darker colors) represent the testing genomes.

Figure C.16: PCA for genomes at $m = 0$ and $r = 10^3$. Circles (lighter colors) represent the training genome and triangles (darker colors) represent the testing genomes.
Figure C.17: PCA for genomes at $m = 0$ and $r = 10^4$. Circles (lighter colors) represent the training genome and triangles (darker colors) represent the testing genomes.

Figure C.18: PCA for genomes at $m = 0$ and $r = 10^5$. Circles (lighter colors) represent the training genome and triangles (darker colors) represent the testing genomes.
Figure C.19: PCA for genomes at $m = 0$ and $r = 10^6$. Circles (lighter colors) represent the training genome and triangles (darker colors) represent the testing genomes.

Figure C.20: PCA for individual plasmid identification at $m = 0$ and $r = 10^2$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
Figure C.21: PCA for individual plasmid identification at \( m = 0 \) and \( r = 10^3 \). Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.

Figure C.22: PCA for individual plasmid identification at \( m = 0 \) and \( r = 10^4 \). Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
Figure C.23: PCA for individual plasmid identification at $m = 0$ and $r = 10^5$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.

Figure C.24: PCA for individual plasmid identification at $m = 0$ and $r = 10^6$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
Figure C.25: PCA for group plasmid identification at $m = 0$ and $r = 10^2$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.

Figure C.26: PCA for group plasmid identification at $m = 0$ and $r = 10^3$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
Figure C.27: PCA for group plasmid identification at $m = 0$ and $r = 10^4$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.

Figure C.28: PCA for group plasmid identification at $m = 0$ and $r = 10^5$. Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
Figure C.29: PCA for group plasmid identification at \( m = 0 \) and \( r = 10^6 \). Circles and triangles (lighter colors) represent the training plasmids and squares and diamonds (darker colors) represent the testing plasmids.
APPENDIX D. PYTHON CODE

The formatting of the Python code is as follows: comments are highlighted in green, Python keywords are highlighted in blue, strings are highlighted in purple, and the red arrow indicates line wrapping.

D.1 Chapter 6 Stability Codes

D.1.1 Velocity Profile Code

def spin_profile(disk_radius, t):
    if t <= 6:
        time = t
        decel = disk_radius * 500 / 9.5492966
        F_spin = decel * time
    elif t < 60:
        time = t - 6
        decel = 0
        F_spin = disk_radius * 3000 / 9.5492966
    elif t >= 60:
        time = t - 60
        decel = -disk_radius * 10 / 9.5492966
F_spin = disk_radius * 3000 / 9.5492966 + decel * time
return F_spin, decel

def rbc_velocity(disk_radius, d1, d2, nu1, nu2, lmda, t, y):
    # Finds the velocity for the RBC layer
    u2 = np.zeros((y.shape[0], t.shape[0]))

    for kk in range(t.shape[0]):
        if t[kk] < 16:
            nu1 = 0.0000048
            nu2 = nu1
        for jj in range(y.shape[0]):
            lmda_sum = 0
            for ii in range(lmda.shape[0]):
                F, decel = spin_profile(disk_radius, t[kk])
                B = B_constant(d1, d2, nu1, nu2, lmda[ii])
                fm = f_coefficient(B, decel, d1, d2, nu1, nu2, lmda[ii])
                theta = make_theta(B, y[jj], d1, d2, nu1, nu2, lmda[ii])
                lmda_sum = lmda_sum + (fm / lmda[ii] ** 2) * (1 - np.exp(-t[kk] * lmda[ii] ** 2)) * theta
            u2[jj, kk] = F + lmda_sum
    return u2

def plasma_velocity(disk_radius, d1, d2, nu1, nu2, lmda, t, y):
    # Finds the velocity for the plasma layer
u1 = np.zeros((y.shape[0], t.shape[0]))

for kk in range(t.shape[0]):
    if t[kk] < 16:
        nu1 = 0.0000048
        nu2 = nu1
    for jj in range(y.shape[0]):
        lmda_sum = 0
        for ii in range(lmda.shape[0]):
            F, decel = spin_profile(disk_radius, t[kk])
            B = B_constant(d1, d2, nu1, nu2, lmda[ii])
            fm = f_coefficient(B, decel, d1, d2, nu1, nu2, lmda[ii])
            phi = make_phi(B, y[jj], d1, nu1, lmda[ii])
            lmda_sum = lmda_sum + (fm/lmda[ii]**2) * (1-np.exp(-t[kk]*lmda[ii]**2)) * phi
        u1[jj, kk] = F + lmda_sum

return u1

def acceleration_singlelayer(k, d, nu_single, t, y):

    v = np.zeros((y.shape[0], t.shape[0]))

    for kk in range(t.shape[0]):
        for jj in range(y.shape[0]):
            lmda_sum = 0
            for ii in range(1, 200):
                lmda_sum = lmda_sum + 16*2**0.5*k*d**2/(nu_single**2 * ii**3*np.pi**3)*np.sin(ii*np.pi/4)**2 * (1
\[ - \exp(-t[kk]*\pi^2*i[i]^2*\nu\text{single}/(4*d^2)) * \sin(i[i]*\pi*y[j]/(2*d)) \]

\[ v[jj,kk] = \text{lmda}_\text{sum} \]

```python
def corrected_constant_vel(F, k, d, nu_single, t, y, t_up):
    u = np.zeros((y.shape[0], t.shape[0]))
    for kk in range(t.shape[0]):
        for jj in range(y.shape[0]):
            lmda_sum = 0
            for ii in range(1, 300, 2):
                lmda_sum = lmda_sum + (16*2**0.5*k*d**2/(nu_single*i[ii]**2*\pi**2) * (1-\exp(-\pi^2*i[ii]**2*nu_single*t_up/(4*d^2))) * \sin(\pi/4)**2/i[ii] * \exp(-t[kk]*\pi^2*\pi*y[j])/\sin(y[j]/(2*d))
            u[jj, kk] = F - lmda_sum
    return u
```

```python
def lmda_func(lmda, d1, d2, nu1, nu2, m, q):
    answer = (1-m*q)*np.cos(lmda*(d1/\sqrt{nu1}-d2/\sqrt{nu2})) - (1+m*q)*np.cos(lmda*(d2/\sqrt{nu2}+d1/\sqrt{nu1}))
    return answer
```

294
```python
def find_lmda(guess_low, guess_high, d1, d2, nu1, nu2, m, q):
    M = len(guess_low)
    lmda = np.zeros(M)
    for kk in range(M):
        lmda[kk] = opt.brentq(lmda_func, guess_low[kk], guess_high[kk], args=(d1, d2, nu1, nu2, m, q))
    return lmda

def B_constant(d1, d2, nu1, nu2, lmda):
    c = np.sqrt(nu1) / (4 * lmda) * np.sin(2 * lmda * d1 / np.sqrt(nu1))
    d = np.sqrt(nu2) / (4 * lmda) * np.sin(2 * lmda * d2 / np.sqrt(nu2))
    B = (d1 / 2 + c + ((np.cos(lmda * d1 / np.sqrt(nu1))) ** 2 / (np.sin(lmda * d2 / np.sqrt(nu2))) ** 2) * (d2 / 2 - d)) ** (-1/2)
    return B

def coefficient(B, k, d1, d2, nu1, nu2, lmda):
    fm = -B * k * ((np.sqrt(nu1) / lmda) * np.sin(lmda * d1 / np.sqrt(nu1)) -
                   (np.cos(lmda * d1 / np.sqrt(nu1)) / (np.sin(lmda * d2 / np.sqrt(nu2))) * (np.sqrt(nu2) / lmda) * np.cos(lmda * d2 / np.sqrt(nu2))) - 1))
    return fm

def make_phi(B, y, d1, nu1, lmda):
    phi = B * np.cos(lmda * (d1 - y) / np.sqrt(nu1))
    return phi

def make_theta(B, y, d1, d2, nu1, nu2, lmda):
    theta = B * np.cos(lmda * d1 / np.sqrt(nu1)) * np.sin(lmda * (y + d2) / np.sqrt(nu2)) / np.sin(lmda * d2 / np.sqrt(nu2))
    return theta
```
```python
### Constants ###

disk_radius = 0.06
t_hold = np.array([0, 10, 60])#, 65, 70, 90, 120, 150, 200, 250,
                     300, 360])
# t = np.linspace(0, 6, 20)
F = disk_radius * 300 / 9.5492966
k = disk_radius * 500 / 9.5492966
t_up = F / k

t_up_array = np.linspace(0, 6, 7)
D = 2.97e-3
hct = 0.45
hct_pack = 0.88
d1 = (1 - hct / hct_pack) * D
d2 = hct / hct_pack * D
mu1 = 0.0012
rho1 = 1024
# mu2 = np.array([0.025, 0.02, 0.0144, 0.0085, 0.0083, 0.0082,
              0.0081, 0.008, 0.008])
mu2 = 0.035
rho2 = 1108
# mu1 = mu2
# rho1 = rho2
nu1 = mu1 / rho1
nu2 = mu2 / rho2
nu_single = nu1
m = mu2 / mu1
q = np.sqrt(nu1) / np.sqrt(nu2)
y_up_graph = np.linspace(d1, 0, 31)
y_low_graph = np.linspace(0, -d2, 31)
```

```python
y = np.linspace(0,D,61)

# guess = np.zeros(200)
# guess_low = np.zeros(len(guess))
# guess_high = np.zeros(len(guess))
# value_low = np.zeros(len(guess))
# value_high = np.zeros(len(guess))
# guess[0] = 1
# guess_low[0] = 0
#for ii in range(len(guess)-1):
  # guess[ii+1] = guess[ii] + 2
  # guess_high[ii] = guess[ii] + 1
  # value_high[ii] = lmda_func(guess_high[ii],d1,d2,nu1,nu2,m,q)
  # value_low[ii] = lmda_func(guess_low[ii],d1,d2,nu1,nu2,m,q)
  # while (np.sign(value_high[ii]) == np.sign(value_low[ii])):
  #   guess_high[ii] = guess[ii] + 3*random.random() -1.5
  #   value_high[ii] = lmda_func(guess_high[ii],d1,d2,nu1,nu2,m,q)
  # guess_low[ii+1] = guess_high[ii]
  # if ii == len(guess)-2:
  #   guess_high[ii+1] = guess[ii+1] + 1
  #   value_high[ii+1] = lmda_func(guess_high[ii+1],d1,d2,nu1,nu2,m,q)
  #   value_low[ii+1] = lmda_func(guess_low[ii+1],d1,d2,nu1,nu2,m,q)
  # while (np.sign(value_high[ii+1]) == np.sign(value_low[ii+1])):
  #   guess_high[ii+1] = guess[ii+1] + 3*random.random() -1.5
```

297
value_high[i+1] = lmda_func(guess_high[i+1], d1, d2, nu1, nu2, m, q)

#
# n = np.linspace(0, 30, 301)
# graph_lmda = lmda_func(n, d1, d2, nu1, nu2, m, q)
# plt.plot(n, graph_lmda)
# value_high = lmda_func(guess_high, d1, d2, nu1, nu2, m, q)
# value_low = lmda_func(guess_low, d1, d2, nu1, nu2, m, q)
# lmda = find_lmda(guess_low, guess_high, d1, d2, nu1, nu2, m, q)

# for ii in range(len(decel_rate)):
# ii = 0
# u1_hct = plasma_velocity(disk_radius, d1, d2, nu1, nu2, lmda, t_hold, y_up_graph)
# u2_hct = rbc_velocity(disk_radius, d1, d2, nu1, nu2, lmda, t_hold, y_low_graph)
# velocity_profile = np.vstack((u1_hct, u2_hct))

velocity_accel_single = acceleration_singlerey(k, D, nu_single, t_up_array, y)
velocity_corrected_single = corrected_constant_vel(F, k, D, nu_single, t_hold, y, t_up)

print(velocity_corrected_single[0,2] - velocity_corrected_single[[[-1,2]])

# v = np.tile(y_up_graph.reshape(31,1), len(t))
# w = np.tile(y_low_graph.reshape(31,1), len(t))
# y_profile = np.vstack((v,w))
# y_profile = np.tile(y.reshape(61,1), len(t_hold))
```python
y_wall = y_profile[0,:]
y_interface = y_profile[30,:]
y_air = y_profile[-1,:]

# fig = plt.figure(frameon=False)
# ax = fig.add_axes([0,0,1,1])
# ax.axis('off')
ax = plt.subplot(111)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.get_xaxis().tick_bottom()
ax.yaxis.set_ticks([])
ax.spines['left'].set_visible(False)

# ax.plot(np.linspace(velocity_profile[-1,-1]-0.01,
#       velocity_profile[0,0]+0.01,len(y_air)), y_air, 'k--', lw=2.5)
# ax.plot(np.linspace(velocity_profile[-1,-1]-0.01,
#       velocity_profile[0,0]+0.01,len(y_interface)), y_interface,
#       'k--', lw=2.5)
# ax.plot(np.linspace(velocity_corrected_single[-1,-1]-0.01,
#       velocity_corrected_single[0,0]+0.01,len(y_wall)), y_wall, 'k--', lw=1)
ax.plot(fix, y_profile[:,0], 'k', lw=2, ls='--')
ax.plot(velocity_corrected_single[:,0], y_profile[:,0], 'k', lw=2, ls='--')
ax.plot(velocity_corrected_single[:,1], y_profile[:,1], 'k', lw=2, ls=':')
ax.plot(velocity_corrected_single[:,2], y_profile[:,2], 'k', lw=2, ls='-')

# for ii in range(1,len(t_hold)):
#   jj = (ii) % 6 - 1
```
# color_array = ['darkorange', 'b', 'dimgrey', 'g', 'r', 'saddlebrown', 'm']
# ax.plot(velocity_corrected_single[:, ii], y_profile[:, ii],
#          color=color_array[jj], lw=2)
plt.rc('font', size=18)  # controls default text sizes
plt.rc('axes', titlesize=26)  # fontsize of the axes title
plt.rc('axes', labelsize=22)  # fontsize of the x and y labels
plt.rc('xtick', labelsize=20)  # fontsize of the tick labels
plt.rcParams['font.weight'] = 'bold'
# plt.ylim((-0.002, 0.0017))
plt.ylim((0, D))
plt.xlabel('Fluid Velocity (m/s)', weight='bold')
# plt.annotate('t=' + str(t[0]), xy=(velocity_constant_single[-1, 0],
#                                   y_profile[-1, 0]), xytext=(velocity_constant_single
#                                                                [-1, 0] + 0.005, y_profile[-1, 0] - 0.00016), ha='right',
#               rotation=62, weight='bold')
# plt.annotate('t=' + str(t[1]), xy=(velocity_constant_single[-1, 1],
#                                   y_profile[-1, 1]), xytext=(velocity_constant_single
#                                                                [-1, 1] + 0.003, y_profile[-1, 1] - 0.00016), ha='right',
#               rotation=61, weight='bold')
# for jj in range(2, len(t)):
#     plt.annotate('t=' + str(t[jj]), xy=(velocity_constant_single
#                                          [-1, jj], y_profile[-1, jj]), xytext=(
#                                          velocity_constant_single[-1, jj] + 0.001, y_profile[-1, jj]
#                                          - 0.00016), ha='right', rotation=60, weight='bold')
# plt.annotate('Plasma', xy=(velocity_profile[0, 0] + 0.002, y_profile[15, 0]),
#               xytext=(velocity_profile[0, 0] + 0.0057, y_profile[15, 0]), xycoords='data', va='center', rotation='vertical', arrowprops=dict(arrowstyle='[-[, widthB=3.8, lengthB=0.4'])
D.1.2 Non-Newtonian Modeling

```python
import pandas as pd
import numpy as np
import scipy.optimize as opt
import matplotlib.pyplot as plt

def rbc_velocity(F, Re, gamma, beta, lmda, tau, eta):
    # Finds the velocity for the RBC layer
    u2 = np.zeros((eta.shape[0]))
```

301
```python
for jj in range(eta.shape[0]):
    lmda_sum = 0
    for ii in range(lmda.shape[0]):
        B = B_constant(gamma, beta, lmda[ii])
        fm = f_coefficient(B, Re, gamma, beta, lmda[ii])
        theta = make_theta(B, eta[jj], gamma, beta, lmda[ii])
        lmda_sum = lmda_sum + (fm/lmda[ii]**2) * (1-np.exp(-
                                tau*lmda[ii]**2)) * theta
    u2[jj] = F - Re*tau + lmda_sum
return u2

def rbc_shear_rate(F, Re, gamma, beta, lmda, tau, eta):
    # Finds the velocity for the RBC layer
    u2_deriv = np.zeros((eta.shape[0]))
    for jj in range(eta.shape[0]):
        lmda_sum = 0
        for ii in range(lmda.shape[0]):
            B = B_constant(gamma, beta, lmda[ii])
            fm = f_coefficient(B, Re, gamma, beta, lmda[ii])
            theta = theta_deriv(B, eta[jj], gamma, beta, lmda[ii])
            lmda_sum = lmda_sum + (fm/lmda[ii]**2) * (1-np.exp(-
                                tau*lmda[ii]**2)) * theta
        u2_deriv[jj] = lmda_sum
    return u2_deriv

def plasma_velocity(F, Re, gamma, beta, lmda, tau, eta):
    # Finds the velocity for the plasma layer
```
u1 = np.zeros((eta.shape[0]))

for jj in range(eta.shape[0]):
    lmda_sum = 0
    for ii in range(lmda.shape[0]):
        B = B_constant(gamma, beta, lmda[ii])
        fm = f_coefficient(B, Re, gamma, beta, lmda[ii])
        phi = make_phi(B, eta[jj], gamma, beta, lmda[ii])
        lmda_sum = lmda_sum + (fm / lmda[ii]**2) * (1 - np.exp(-tau*lmda[ii]**2)) * phi
    u1[jj] = F - Re*tau + lmda_sum
return u1

def plasma_shear_rate(F, Re, gamma, beta, lmda, tau, eta):
    # Finds the velocity for the plasma layer
    u1_deriv = np.zeros((eta.shape[0]))

    for jj in range(eta.shape[0]):
        lmda_sum = 0
        for ii in range(lmda.shape[0]):
            B = B_constant(gamma, beta, lmda[ii])
            fm = f_coefficient(B, Re, gamma, beta, lmda[ii])
            phi = phi_deriv(B, eta[jj], gamma, beta, lmda[ii])
            lmda_sum = lmda_sum + (fm / lmda[ii]**2) * (1 - np.exp(-tau*lmda[ii]**2)) * phi
        u1_deriv[jj] = lmda_sum
    return u1_deriv

def lmda_func(lmda, gamma, beta, zeta):

303
answer = (1 - zeta / np.sqrt(beta)) * np.cos(lmda * (gamma / np.sqrt(bet - 1))) + (1 + zeta / np.sqrt(beta)) * np.cos(lmda * (gamma / np.sqrt(beta) + 1))
return answer

def find_lmda(guess_low, guess_high, gamma, beta, zeta):
    M = len(guess_low)
lmda = np.zeros(M)
    for kk in range(M):
        lmda[kk] = opt.brentq(lmda_func, guess_low[kk], guess_high[kk], args=(gamma, beta, zeta))
    return lmda

def B_constant(gamma, beta, lmda):
c = np.sqrt(beta) / (4 * lmda) * np.sin(2 * lmda * gamma / np.sqrt(beta))
B = (gamma / 2 + c + ((np.cos(lmda * gamma / np.sqrt(beta))) ** 2 / np.sin(lmda)) ** 2) * (1 / 2 - np.sin(2 * lmda) / (4 * lmda)) ** (-1/2)
return B

def f_coefficient(B, Re, gamma, beta, lmda):
fm = B * Re * ((np.sqrt(beta) / lmda) * np.sin(lmda * gamma / np.sqrt(beta)) - (np.cos(lmda * gamma / np.sqrt(beta))) / (lmda * np.sin(lmda))) * (np.cos(lmda) - 1))
return fm

def make_phi(B, eta, gamma, beta, lmda):
    phi = B * np.cos(lmda * (gamma - eta) / np.sqrt(beta))
    return phi
def phi_deriv(B, eta, gamma, beta, lmda):
    phi = B*(lmda/np.sqrt(beta))*np.sin(lmda*(gamma-eta)/np.sqrt(beta))
    return phi

def make_theta(B, eta, gamma, beta, lmda):
    theta = B*np.cos(lmda*gamma/np.sqrt(beta))*np.sin(lmda*(eta+1))/np.sin(lmda)
    return theta

def theta_deriv(B, eta, gamma, beta, lmda):
    theta = B*np.cos(lmda*gamma/np.sqrt(beta))*lmda*np.cos(lmda*(eta+1))/np.sin(lmda)
    return theta

time = [10]
rates = [5.2]
data = np.zeros((len(time), len(rates)*2))
for ii in rates:
    ### Constants ###
disk_radius = 0.06
U0 = disk_radius*3000/9.5492966
ramp_rate = ii
decel_rate = disk_radius*ramp_rate/9.5492966
total_time = U0/(decel_rate)
t = time
#t = np.linspace(1, total_time, (total_time*2+1))
D = 2.97e-3
hct = .45
hct_pack = .88  
\[ d1 = (1 - hct/hct\_pack) \times D \]  
\[ d2 = hct/hct\_pack \times D \]  
\[ \mu1 = 0.0012 \]  
\[ \rho1 = 1024 \]  
\[ \nu1 = \mu1 / \rho1 \]  
\[ uc = \exp(-1.18016 \times \log(1-hct\_pack) + 0.52904) \]  
\[ tc = \exp(3.964 \times \log(hct\_pack) + 4.36911) \]  
\[ F = U0 / \sqrt{decel\_rate \times d2} \]  
\[ \gamma = d1 / d2 \]  
\[ \text{viscosity} = \text{np.array}([2.530684041128413719e-01,} \]  
\[ \quad \rightarrow 2.468171970643559809e-01, 2.409746296818014999e-01,} \]  
\[ \quad \rightarrow 2.35455257233765961e-01, 2.302912593995248836e-01,} \]  
\[ \quad \rightarrow 2.253747979606624818e-01, 2.207732302534949065e-01,} \]  
\[ \quad \rightarrow 2.163651544035880003e-01, 2.122333798030327512e-01,} \]  
\[ \quad \rightarrow 2.082596400549800442e-01, 2.045211610164151261e-01,} \]  
\[ \quad \rightarrow 2.009158572087768324e-01, 1.975107397877999826e-01,} \]  
\[ \quad \rightarrow 1.942328488062659564e-01, 1.911158102514392276e-01,} \]  
\[ \quad \rightarrow 1.881246084063777080e-01, 1.852561670967086682e-01,} \]  
\[ \quad \rightarrow 1.825172157309251086e-01, 1.798653352814224560e-01,} \]  
\[ \quad \rightarrow 1.773489334047538279e-01, 1.748876765897088292e-01,} \]  
\[ \quad \rightarrow 1.72567771990819509e-01, 1.702762147286097094e-01,} \]  
\[ \quad \rightarrow 1.681296176506243689e-01, 1.659912150562865840e-01,} \]  
\[ \quad \rightarrow 1.639965622621839536e-01, 1.619993349014665074e-01,} \]  
\[ \quad \rightarrow 1.601342266823596594e-01, 1.582774218802297794e-01,} \]  
\[ \quad \rightarrow 1.564888498781972503e-01, 1.551105349483969942e-01]) \]  
\[ \rho2 = 1108 \]  
\[ \zeta1 = \mu1 / 0.0144 \]  
\[ \beta1 = \nu1 / (0.0144 / \rho2) \]  
\[ \eta\_up\_graph = \text{np.linspace}(\gamma, 0.31) \]
eta_low_graph = np.linspace(0, -1, 31)

u1_hct = np.zeros((len(eta_up_graph), len(t)))
u2_hct = np.zeros((len(eta_low_graph), len(t)))
u1_deriv = np.zeros((len(eta_up_graph), len(t)))
u2_deriv = np.zeros((len(eta_low_graph), len(t)))
u1_non = np.zeros((len(eta_up_graph), len(t)))
u2_non = np.zeros((len(eta_low_graph), len(t)))

for jj in range(len(time)):
    if time[jj] > 3000/ii:
        continue

    for jj in range(len(t)):
        print(jj)
    if jj == 0:
        mu2 = 0.035
        nu2 = mu2/rho2
        zeta = mu1/mu2
        beta = nu1/nu2
        tau = nu2*t[jj]/d2**2
        Re1 = np.sqrt(decel_rate*d2**3)/nu2
        Re2 = Re1/beta
        Re = np.sqrt(decel_rate*d2**3)/nu2

        x = np.linspace(0, 10, 5001)
        value_x = lmda_func(x, gamma, beta, zeta)
        guess_low = [x[aa] for aa in range(len(value_x)) if
                     np.sign(value_x[aa]) != np.sign(value_x[aa+1])]
        guess_high = np.array(guess_low) + x[2] - x[1]
lmda = find_lmda(guess_low, guess_high, gamma, beta, zeta)

u1_hct[:, jj] = plasma_velocity(F, Re, gamma, beta, lmda, tau, eta_up_graph)
u2_hct[:, jj] = rbc_velocity(F, Re, gamma, beta, lmda, tau, eta_low_graph)

# u1_non[:, jj] = plasma_velocity(F, Re, gamma, beta, lmda, tau, eta_up_graph)
# u2_non[:, jj] = rbc_velocity(F, Re, gamma, beta, lmda, tau, eta_low_graph)

print("finished 0")

# else:
#     if time[jj] < 5:
#         mu2 = np.tile(200, 31)
#     else:
#         mu2 = viscosity
#
#     mu2 = np.tile(180, 31)
u2 = mu2/rho2
zeta = mu1/mu2[0]
beta = nu1/nu2[0]
tau = nu2[0]*t[jj]/d2**2
Re = np.sqrt(decel_rate*d2**3)/nu2[0]
x = np.linspace(0, 5, 5001)
value_x = lmda_func(x, gamma, beta, zeta)
guess_low = [x[aa] for aa in range(len(value_x)-1) if np.sign(value_x[aa]) != np.sign(value_x[aa+1])]

guess_high = np.array(guess_low)+x[2]-x[1]
\[ lmda = \text{find}_lmda(\text{guess}_\text{low}, \text{guess}_\text{high}, \gamma, \beta, \zeta) \]

# \[ u1_{\text{hct}}[:, jj] = \text{plasma_velocity}(F, Re, \gamma, \beta, \rightarrow lmda, \tau, \eta_{\text{up}\_\text{graph}}) \]
# \[ u2_{\text{hct}}[:, jj] = \text{rbc_velocity}(F, Re, \gamma, \beta, \rightarrow lmda, \tau, \eta_{\text{low}\_\text{graph}}) \]
# \[ u1_{\text{deriv}}[:, jj] = \text{plasma_shear_rate}(F, Re, \gamma, \beta, \rightarrow lmda, \tau, \eta_{\text{up}\_\text{graph}}) \]
# \[ u2_{\text{deriv}}[:, jj] = \text{rbc_shear_rate}(F, Re, \gamma, \beta, lmda, \rightarrow \tau, \eta_{\text{low}\_\text{graph}}) \]

\[ n_n = 0 \]
\[ k_k = 1 \]

while \( k_k > 0.000001 \): 
    \[ \mu_{2\_\text{new}} = \frac{(u_c + 2u_c^{0.5} \ast t_c^{0.5}) / \text{np.abs}(u_{2\_\text{deriv}}[:, jj] \ast \text{np.sqrt}(d2 \ast \text{decel\_rate}) / d2)^{0.5} + t_c / \text{np.abs}(u_{2\_\text{deriv}}[:, jj] \ast \text{np.sqrt}(d2 \ast \text{decel\_rate}) / d2))}{1000} \]
    \[ k_k = \text{np.max}(\text{np.abs}(\mu_{2\_\text{new}} - \mu_2)) \]
    if \( n_n \% 10 == 0 \):
        print(kk)
        # print(\( \mu_{2\_\text{new}}[0] \))
        # print(\( u_{2\_\text{deriv}} \))
        # print(\( \zeta \ast u_{1\_\text{deriv}} \))
    \[ n_n += 1 \]
    \[ \mu_2 = \mu_{2\_\text{new}} \]
    \[ \nu_2 = \mu_2 / \rho_2 \]
    \[ \zeta = \mu_1 / \mu_{2[0]} \]

309
\[
\beta = \frac{\nu_1}{\nu_2[0]}
\]
\[
\tau = \frac{\nu_2[0] \cdot t[jj]}{d^2 \cdot 2}
\]
\[
Re = \sqrt{\frac{\text{decel rate} \cdot d^2}{\nu_2[0]}}
\]
\[
Re_3 = \frac{Re}{\beta}
\]

if \( t[jj] < 0.5 \):
    end = 5
elif \( t[jj] < 0.5 \):
    end = 1
else:
    end = 50

guess = 0
guess_low = 0
guess_high = 0

x = np.linspace(0, end, 5001)
value_x = lmda_func(x, gamma, beta, zeta)
guess_low = [x[aa] for aa in range(len(value_x))]
    if np.sign(value_x[aa]) != np.sign(value_x[aa+1])]
guess_high = np.array(guess_low) + x[1]

lmda = find_lmda(guess_low, guess_high, gamma, beta, zeta)
u2_deriv[:, jj] = rbc_shear_rate(F, Re, gamma, beta, lmda, tau, eta_low_graph)

if \( t[jj] < 0.5 \) and nn > 50:
    kk = 0.000001
if \( t[jj] >= 0.5 \) and nn > 200:
    kk = 0.000001
\[
\begin{align*}
\text{u1\_non}[\ldots, jj] &= \text{plasma\_velocity}(F, Re, gamma, beta, lmda, \tau, \eta_{\text{up\_graph}}) \\
\text{u2\_non}[\ldots, jj] &= \text{rbc\_velocity}(F, Re, gamma, beta, lmda, \tau, \eta_{\text{low\_graph}}) \\
\text{u1\_deriv}[\ldots, jj] &= \text{plasma\_shear\_rate}(F, Re, gamma, beta, lmda, \tau, \eta_{\text{up\_graph}})
\end{align*}
\]

```
# u1_non[:, jj] = plasma_velocity(F, Re, gamma, beta, lmda, tau, eta_up_graph) * np.sqrt(d2 * decel_rate)
## u2_non[:, jj] = rbc_velocity(F, Re, gamma, beta, lmda, tau, eta_low_graph) * np.sqrt(d2 * decel_rate)
# u1_deriv[:, jj] = plasma_shear_rate(F, Re, gamma, beta, lmda, tau, eta_up_graph) * np.sqrt(d2 * decel_rate) / d2
## u2_deriv[:, jj] = rbc_shear_rate(F, Re, gamma, beta, lmda, tau, eta_low_graph) * np.sqrt(d2 * decel_rate) / d2
## velocity_profile_org = np.vstack((u1_hct, u2_hct)) * np.sqrt(d2 * decel_rate)
## velocity_profile_non = np.vstack((u1_non, u2_non)) * np.sqrt(d2 * decel_rate)
## velocity_profile_nd = np.vstack((u1_non, u2_non))
## rate_profile = np.vstack((u1_deriv, u2_deriv)) * np.sqrt(d2 * decel_rate) / d2
## stress1 = zeta * u1_deriv * np.sqrt(d2 * decel_rate) / d2 * mu2
## [0]
## stress2 = u2_deriv * np.sqrt(d2 * decel_rate) / d2 * mu2 * 1000
## stress_profile = np.vstack((stress1.reshape(31,1), stress2.reshape(31,1)))
# data[:, ii*2] = u1_non[30,:]
# data[:, ii*2+1] = u1_deriv[30,:]
```
# v = np.tile(eta_up_graph.reshape(31,1),len(t))
# w = np.tile(eta_low_graph.reshape(31,1),len(t))
## eta_profile = np.vstack((eta_up_graph.reshape(31,1),
← eta_low_graph.reshape(31,1)))
# col_namel =[['Velocity ' + str(aa) + ' RPM', 'Shear Rate ' + str(aa) + ' RPM']
← for aa in rates]
# col_name = [bb for item in col_namel for bb in item]
# print('saving')
# df_Final = pd.DataFrame(data, columns=col_name, index=time)
## df_Final['Velocity Profile'] = velocity_profile_non
## df_Final['Velocity Profile ND'] = velocity_profile_nd
## df_Final['Shear Rate'] = rate_profile
## df_Final['Shear Stress'] = stress_profile
## df_Final['Instability Growth Lower'] = np.sqrt(stress_profile /
← rho2)/d2
## df_Final['Instability Growth Upper'] = np.sqrt(stress_profile /
← rho1)/d2
## df_Final['Viscosity'] = np.vstack((np.tile(mu1,31).reshape
← (31,1),mu2.reshape(31,1))))
## df_Final['nu'] = np.vstack((np.tile(nu1,31).reshape(31,1),nu2.
← reshape(31,1))))
## df_Final['Re'] = np.tile(Re,62).reshape(62,1)
## df_Final['d2'] = np.tile(d2,62).reshape(62,1)
## df_Final['tau'] = np.tile(tau,62).reshape(62,1)
## df_Final['decel rate'] = np.tile(decel_rate,62).reshape(62,1)
#
# df_Final.to_csv('Kelvin Helmholtz Data5.csv')
velocity_profile_org = np.vstack((u1_hct, u2_hct))*np.sqrt(d2*decel_rate)
velocity_profile_non = np.vstack((u1_non, u2_non))*np.sqrt(d2*decel_rate)

velocity_profile_nd[:, jj] = np.vstack((u1_non[:, jj].reshape(31,1), u2_non[:, jj].reshape(31,1)))
rate_profile = np.vstack((u1_deriv, u2_deriv))*np.sqrt(d2*decel_rate)/d2

stress1[:, jj] = zeta * u1_deriv[:, jj]*np.sqrt(d2*decel_rate)/d2*mu2[0]
stress2[:, jj] = u2_deriv[:, jj]*np.sqrt(d2*decel_rate)/d2*mu2*1000
stress_profile = np.vstack((stress1, stress2))

v = np.tile(eta_up_graph.reshape(31,1), len(t))
w = np.tile(eta_low_graph.reshape(31,1), len(t))

eta_profile = np.vstack((eta_up_graph.reshape(31,1), eta_low_graph.reshape(31,1)))
y_profile = np.vstack((v,w))

# print('saving')
# df_Final = pd.DataFrame(eta_profile, columns=['y'])
# df_Final['Velocity Profile'] = velocity_profile_non
# df_Final['Velocity Profile ND'] = velocity_profile_nd
# df_Final['Shear Rate'] = rate_profile
# df_Final['Shear Stress'] = stress_profile

# df_Final.to_csv('Datafor'+str(ramp_rate)+'RPMat'+str(t)+'seconds.csv')
# eta_wall = eta_profile[-1,:]
# eta_interface = eta_profile[30,:]
# eta_air = eta_profile[0,:]

ax = plt.subplot(111)
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.get_xaxis().tick_bottom()
ax.yaxis.set_ticks([])
ax.spines['left'].set_visible(False)

# ax.plot(np.linspace(velocity_profile[-1,-1]-0.01,
#                         velocity_profile[0,0]+0.01,len(eta_air)), eta_air, 'k--',
#                         lw=2.5)
# ax.plot(np.linspace(velocity_profile[-1,-1]-0.01,
#                         velocity_profile[0,0]+0.01,len(eta_interface)),
#                         eta_interface, 'k--', lw=2.5)
# ax.plot(np.linspace(velocity_profile[-1,-1]-0.01,
#                         velocity_profile[0,0]+0.01,len(eta_wall)), eta_wall, 'k--',
#                         lw=1)

ax.plot(velocity_profile_org[:,0], y_profile[:,0], 'r', lw=2,
#                         dashes=[5, 3, 5, 3])
# ax.plot(velocity_profile_org[:,0], y_profile[:,0], 'k', lw=2)
# for ii in range(0,len(t)):
#     jj = (ii) % 7
#     color_array = ['k', 'darkorange', 'b', 'dimgrey', 'g', 'r',
#                     'saddlebrown', 'm']
# ax.plot(velocity_profile_non[:,ii], eta_profile[:,ii], color
#                     =color_array[jj], lw=2)
ax.plot(velocity_profile_non[:,0], y_profile[:,0], 'k', lw=2)
# ax.plot(velocity_profile_non[::1, y_profile[::1]] , 'k', lw=2,
              dashes=[5, 3, 5, 3])
# ax.plot(velocity_profile_non[::2, y_profile[::2]] , 'k', lw=2,
              dashes=[2, 2, 2, 2])
# ax.plot(velocity_profile_non[::3, y_profile[::3]] , 'k', lw=2,
              dashes=[8, 2, 2, 2])
# ax.plot(velocity_profile_non[::4, y_profile[::4]] , 'k', lw=2,
              dashes=[10, 3, 10, 3])
# ax.plot(velocity_profile_non[::5, y_profile[::5]] , 'k', lw=2,
              dashes=[8, 2, 2, 2, 2])
## ax.plot(velocity_profile_org, eta_profile , 'k', lw=2)
plt.rc('font', size=18)  # controls default text sizes
plt.rc('axes', titlesize=26)  # fontsize of the axes title
plt.rc('axes', labelsize=22)  # fontsize of the x and y labels
plt.rc('xtick', labelsize=20)  # fontsize of the tick labels
plt.rcParams['font.weight'] = 'bold'
plt.ylim((−1.1, gamma+.1))
plt.xlabel('Fluid Velocity (m/s)', weight='bold')
# plt.annotate('t=' + str(t[0]), xy=(velocity_profile[−1,0],
              y_profile[−1,0]), xytext=(velocity_profile[−1,0]+0.005,
              y_profile[−1,0]−0.00016), ha='right', rotation=62, weight='bold')
# plt.annotate('t=' + str(t[1]), xy=(velocity_profile[−1,1],
              y_profile[−1,1]), xytext=(velocity_profile[−1,1]+0.003,
              y_profile[−1,1]−0.00016), ha='right', rotation=61, weight='bold')
# for jj in range(2, len(t)):
#     plt.annotate('t=' + str(t[jj]), xy=(velocity_profile[−1,jj],
              y_profile[−1,jj]), xytext=(velocity_profile[−1,jj]+0.001,
D.1.3 Linear Stability Code

ttfamily

# This is a Dedalus script.
# Make sure that the python package dedalus is installed before running this script.
# For more information, go to http://dedalus-project.org/ or https://dedalus-project.readthedocs.io/en/latest/.

import time
import numpy as np
import matplotlib.pyplot as plt
from math import *
from mpi4py import MPI
from scipy import optimize as opt
import h5py
from dedalus import public as de
from dedalus.extras import flow_tools
from dedalus.core import operators
import logging
logger = logging.getLogger(__name__)

# Create bases and domain
# Use COMM_SELF so keep calculations independent between processes
Ny = 64
y_basis = de.Chebyshev('y', Ny, interval=(0, 1))
domain = de.Domain([y_basis], grid_dtype=np.complex128, comm=MPI.COMM_SELF)

# Next 8 functions are for calculating the RBC and plasma velocities

def B_constant(eta, beta, lmda):
    c = np.sqrt(beta)/(4*lmda)*np.sin(2*lmda*eta/np.sqrt(beta))
    d = (np.cos(lmda*eta/np.sqrt(beta))**2/(np.sin(lmda))**2
    B = (eta/2 + c + d*(1/2 - np.sin(2*lmda)/(4*lmda)))**(-1/2)
    return B

def f_coefficient(B,Re,eta,beta,lmda):
\[ g = \left( \frac{\sqrt{\beta}}{\lambda} \right) \cdot \sin \left( \frac{\lambda \eta}{\sqrt{\beta}} \right) \]
\[ h = \cos \left( \frac{\lambda \eta}{\sqrt{\beta}} \right) / \left( \lambda \sin \lambda \right) \]
\[ f_m = B \cdot R \cdot (g - h \cdot (\cos \lambda - 1)) \]
\[ \text{return } f_m \]

def make_U1(eta, beta, Re, tau, U0, lmda):
    # Finds the velocity for the Plasma layer
    y = domain.grid(0)
    value = domain.new_field(name='U1')
    value['g'] = 0
    U1_calc = [0] * len(y)
    for jj in range(len(y)):
        lmda_sum = 0
        for ii in range(lmda.shape[0]):
            B = B_constant(eta, beta, lmda[ii])
            fm = f_coefficient(B, Re, eta, beta, lmda[ii])
            lmda_sum = lmda_sum + (fm/lmda[ii]**2) * (1 - np.exp(-tau*lmda[ii]**2)) * B * np.cos(lmda[ii]*eta*(1 - y[jj]/np.sqrt(beta)))
        U1_calc[jj] = U0 - Re*tau + lmda_sum
    value['g'] = U1_calc
    return value

def make_U2(eta, beta, Re, tau, U0, lmda):
    # Finds the velocity for the RBC layer
    y = domain.grid(0)
    value = domain.new_field(name='U2')
    value['g'] = 0
    U2_calc = [0] * len(y)
for jj in range(len(y)):
    lmda_sum = 0
    for ii in range(lmda.shape[0]):
        B = B_constant(eta, beta, lmda[ii])
        fm = f_coefficient(B, Re, eta, beta, lmda[ii])
        lmda_sum = lmda_sum + (fm/lmda[ii]**2)*(1-np.exp(-tau
            *lmda[ii]**2))*B*np.cos(lmda[ii]*eta)/np.sqrt(beta)
            /np.sin(lmda[ii])*np.sin(lmda[ii]*(1-y[jj])/np.sqrt(beta))
        U2_calc[jj] = U0 - Re*tau + lmda_sum
    value['g'] = U2_calc
return value

def make_U1y(eta, beta, Re, tau, lmda):
    y = domain.grid(0)
    value = domain.new_field(name='U1y')
    value['g'] = 0
    U1y_calc = [0] * len(y)
    for jj in range(len(y)):
        lmda_sum = 0
        for ii in range(lmda.shape[0]):
            B = B_constant(eta, beta, lmda[ii])
            fm = f_coefficient(B, Re, eta, beta, lmda[ii])
            lmda_sum = lmda_sum + (fm/lmda[ii]**2)*(1-np.exp(-tau
                *lmda[ii]**2))*B*lmda[ii]*eta*np.sin(lmda[ii]*
                *eta*(1-y[jj])/np.sqrt(beta))/np.sqrt(beta)
        U1y_calc[jj] = lmda_sum
    value['g'] = U1y_calc
```python
return value

def make_U2y(eta, beta, Re, tau, lmda):
y = domain.grid(0)
value = domain.new_field(name='U2y')
value['g'] = 0
U2y_calc = [0] * len(y)
for jj in range(len(y)):
    lmda_sum = 0
    for ii in range(lmda.shape[0]):
        B = B_constant(eta, beta, lmda[ii])
        fm = f_coefficient(B, Re, eta, beta, lmda[ii])
        lmda_sum = lmda_sum - (fm/lmda[ii]**2) * (1-np.exp(-
            tau*lmda[ii]**2)) * B*np.cos(lmda[ii]*eta/np.sqrt
            (beta))/np.sin(lmda[ii]) * lmda[ii]*np.cos(lmda[
            ii]*(1-y[jj]))
    U2y_calc[jj] = lmda_sum

value['g'] = U2y_calc
return value

def make_U1yy(eta, beta, Re, tau, lmda):
y = domain.grid(0)
value = domain.new_field(name='U1yy')
value['g'] = 0
U1yy_calc = [0] * len(y)
for jj in range(len(y)):
    lmda_sum = 0
    for ii in range(lmda.shape[0]):
        B = B_constant(eta, beta, lmda[ii])
```

**Python Code**

```python
fm = f_coefficient(B, Re, eta, beta, lmda[ii])
lmda_sum = lmda_sum - (fm/lmda[ii]**2) * (1-np.exp(-tau*lmda[ii]**2))*B*lmda[ii]**2*eta**2*np.cos(lmda[ii]*eta*(1-y[jj]) / np.sqrt(beta))/beta
U1yy_calc[jj] = lmda_sum

value['g'] = U1yy_calc
return value

def make_U2yy(eta, beta, Re, tau, lmda):
y = domain.grid(0)
value = domain.new_field(name='U2yy')
value['g'] = 0
U2yy_calc = [0] * len(y)
for jj in range(len(y)):
lmda_sum = 0
for ii in range(lmda.shape[0]):
    B = B_constant(eta, beta, lmda[ii])
    fm = f_coefficient(B, Re, eta, beta, lmda[ii])
    lmda_sum = lmda_sum - (fm/lmda[ii]**2) * (1-np.exp(-tau*lmda[ii]**2))*B*np.cos(lmda[ii]*eta/np.sqrt(beta)*np.sin(lmda[ii]*(1-y[jj])))
U2yy_calc[jj] = lmda_sum

value['g'] = U2yy_calc
return value

def max_evalue(alpha, Re, F, eta, zeta, s1, U1, U2, Uly, U2y, U1yy, U2yy):
```

321
"""Calculate smallest e-value for linear growth."""

# Orr–Sommerfeld equation with appropriate (nasty) boundary conditions

```python
problem = de.EVP(domain, variables=['phi1', 'phi1y', 'phi1yy',
                                     'phi1yyy', 'phi2', 'phi2y', 'phi2yy', 'phi2yyy', 'delta1',
                                     'delta2'], eigenvalue='c', tolerance=1e-10)
```

```python
problem.parameters['alpha'] = alpha
problem.parameters['Re'] = Re
problem.parameters['F'] = F
problem.parameters['eta'] = eta
problem.parameters['beta'] = beta
problem.parameters['zeta'] = zeta
problem.parameters['S1'] = s1
problem.parameters['U1'] = U1
problem.parameters['U2'] = U2
problem.parameters['U1y'] = U1y
problem.parameters['U2y'] = U2y
problem.parameters['U1yy'] = U1yy
problem.parameters['U2yy'] = U2yy
```

```python
problem.add_equation('''dy(phi1yyy) / eta**4 - 2*alpha**2 * phi1y / eta**2 + alpha**4 * phi1 - Re - U1) * (alpha**2 * phi1 - phi1yy / eta**2) - U1yy * phi1 / eta**2) = 0'''
```

```python
problem.add_equation('''dy(phi2yyy) - 2*alpha**2 * phi2yy + alpha**4 * phi2 - 1j*alpha*Re*((c/Re - U2)*(alpha**2 * phi2 - phi2yy) - U2yy * phi2) = 0'''
```

```python
problem.add_equation('''c * phi1y - delta1 = 0'''
```

```python
problem.add_equation('''c * phi2y - delta2 = 0'''
```

322
problem.add_equation("dy(phi1) - phi1y = 0")
problem.add_equation("dy(phi1y) - phi1yy = 0")
problem.add_equation("dy(phi1yy) - phi1yyy = 0")
problem.add_equation("dy(phi2) - phi2y = 0")
problem.add_equation("dy(phi2y) - phi2yy = 0")
problem.add_equation("dy(phi2yy) - phi2yyy = 0")

#Now for the boundary conditions
#Here are the BCs on the solid wall
problem.add_bc("right(phi2) = 0")
problem.add_bc("right(phi2y) = 0")

#Here are the BCs for the fluid--fluid interface
problem.add_bc("left(phi1) - left(phi2) = 0")
problem.add_bc("(c/Re-left(U1))*(left(phi1y)/eta + left(phi2y)
→ ) + left(phi1)*left(U1y)/eta + left(phi1)*left(U2y) =
→ 0")
problem.add_bc("(c/Re-left(U1))*(zeta*left(phi1yy)/eta**2 +
→ zeta*alpha**2 *left(phi1) - left(phi2yy) - alpha**2 *
→ left(phi2)) + left(phi1)*left(U1yy) - left(phi1)*left(U2yy) = 0")
problem.add_bc("(c/Re-left(U1))*(zeta*left(phi1yyy)/eta**3 -
→ 3*zeta*alpha**2 *right(phi1y)/eta + left(phi2yyy) - 3*
→ alpha**2 *left(phi2y) + 1j*alpha*Re*(left(phi2)*left(U2y) + zeta/(eta*beta)*left(phi1)*left(U1y)) + 1j*
→ alpha*Re*left(phi1)*(1-zeta/beta)*F + 1j*alpha*Re*zeta
→ /(eta*beta)*(c*left(delta1)/Re**2 - 2*c*left(phi1y)*
→ left(U1)/Re + left(phi1y)*left(U1)**2) + 1j*alpha*Re*(c
→ *left(delta2)/Re**2 - 2*c*left(phi2y)*left(U1)/Re +
→ left(phi2y)*left(U1)**2) = 0")
Here are the BCs for the air interface

```python
problem.add_bc("(c/Re-right(U1))*(right(phi1yy)/eta**2 + alpha**2*right(phi1)) + right(phi1)*right(U2yy) = 0")

problem.add_bc("(c/Re-right(U1))*(3*zeta*alpha**2/eta*right(phi1yy) - zeta/eta**3*right(phi1yy) - 1j*Re*zeta*alpha/(eta*beta)*right(phi1)*right(U1y)) - 1j*alpha*Re*zeta/(eta*beta)*(c*right(delta1)/Re**2 - 2*c*right(phi1)*right(U1)**2) + 1j*alpha*Re*right(phi1)*(alpha**2*S1 + zeta/beta*F) = 0")
```

An additional BC for delta1 and delta2 there are choices to be made here, we could have tried a number of different options (and maybe should)

```python
problem.add_bc("left(delta1) - left(delta2) = 0")
```

# Solve for eigenvalues

```python
solver = problem.build_solver()
solver.solve_dense(solver.pencils[0])
```

# Return largest imaginary part

```python
ev = solver.eigenvalues

# We have to remove the spurious eigenvalues first

ev = ev[np.isfinite(ev)]

return np.max(ev.imag)
```

# Now we have to define the relevant constants (which is a lot)

disk_radius = 0.06
```python
decel_rate = 5
decel = disk_radius * decel_rate / 9.5492966

time = np.linspace(300, 600, 11)
D = 2.97e-3
hct = .45
hct_pack = .88
d1 = (1 - hct / hct_pack) * D
d2 = hct / hct_pack * D
eta = d1 / d2
mu1 = 0.0012
rho1 = 1024
mu2 = 0.035
rho2 = 1108
nu1 = mu1 / rho1
nu2 = mu2 / rho2
zeta = mu1 / mu2
beta = nu1 / nu2
t1 = np.linspace(0.0062044, 0.0193887, 2)

# Lambda function for getting lmda values

def lmda_func(n, eta, beta, zeta):
    answer = (1 - np.sqrt(beta) / zeta) * np.cos(n * (eta / np.sqrt(beta) - 1)) - (1 + np.sqrt(beta) / zeta) * np.cos(n * (eta / np.sqrt(beta) - beta) + 1))
```

```
# Function for iterating to find lmda

def find_lmda(guess_low, guess_high, eta, beta, zeta):
    M = len(guess_low)
    lmda = np.zeros(M)
    for kk in range(M):
        lmda[kk] = opt.brentq(lmda_func, guess_low[kk], guess_high[kk], args=(eta, beta, zeta))
    return lmda

# Getting the guess values for lmda

guess_low = np.zeros(200)
guess_high = np.zeros(200)
value = lmda_func(np.linspace(0, 150, 1501), eta, beta, zeta)
jj = 0
kk = 1
for ii in range(len(guess_low)):
    while (np.sign(value[kk]) == np.sign(value[jj))):
        kk += 1
    guess_low[ii] = (kk - 1)/10
    guess_high[ii] = kk/10
    jj = kk
    kk += 1

lmda = find_lmda(guess_low, guess_high, eta, beta, zeta)

# Creating the list to store the values for all of the different parameter combinations
```
final data = [[0] * 2 for i in range(len(alpha)*len(t1)*len(time))
              + len(time)*((len(t1)+1)+1))]
final_alpha = [0] * len(alpha)*len(t1)*len(time)
final_ci = [0] * len(alpha)*len(t1)*len(time)
index = 0
aindex = 0

# Finding out what c is for a given alpha
for jj in range(len(time)):
    final data[index][0] = 'Velocity'
    final data[index][1] = U0 - Re*tau[jj]
    F = ((U0-Re*tau[jj])*np.sqrt(decel*d2)**2/disk_radius
     index = index + 1
for ii in range(len(t1)):
    s1 = t1[ii]/(rho2*decel*d2**2)
    final data[index][0] = 's1'
    final data[index][1] = s1
    index = index + 1
for kk in range(len(alpha)):
    U1 = make_U1(eta, beta, Re, tau[jj], F, lmda)
    U2 = make_U2(eta, beta, Re, tau[jj], F, lmda)
    U1y = make_U1y(eta, beta, Re, tau[jj], lmda)
    U2y = make_U2y(eta, beta, Re, tau[jj], lmda)
    U1yy = make_U1yy(eta, beta, Re, tau[jj], lmda)
    U2yy = make_U2yy(eta, beta, Re, tau[jj], lmda)
    something = max_value(alpha[kk], Re, F, eta, beta, zeta,
                           s1, U1, U2, U1y, U2y, U1yy, U2yy)
    final data[index][0] = alpha[kk]
    final data[index][1] = something
    final_alpha[aindex] = alpha[kk]
```
final_ci[aindex] = something
index = index + 1
aindex = aindex + 1

# structures that will contain the final wave–numbers alpha, and
→ the corresponding growth factor c
print('now saving data')
with open("data_linear_analysis.txt", "a") as f:
   for row in finaldata:
      print(''.join([str(elem) for elem in row]), file=f)

D.2 Appendix C Fractional Base Composition Codes

Scripts for analyzing DNA fragments in a lossy style format using machine learning This
code has been developed for the analysis of bacterial DNA (both genomic and plasmidic) when
broken up into different segment lengths to be analyzed in a lossy format in which the total A,
T, G, and C content of the segment is known but not the order of the segment. This code was
developed at Brigham Young University under the direction of Dr. William Pitt and Dr. Mark
Clement (http://csl.cs.byu.edu/) while working on a joint project with Dr. Prashant Nagpal from
the University of Colorado Boulder (https://www.colorado.edu/lab/nagpal/). Dr. Nagpal developed
a device which can read DNA bases in segments and print out the total number of A’s, T’s, G’s,
and C’s found in the segment.

The purpose of this project was to determine whether or not you could identify bacterial
species and known antibiotic resistances using DNA segment data in the lossy format from Dr.
Nagpal’s device. The project is still ongoing and the code is still under development.

The current working code is called Complete Code. The only thing that needs to be updated
is the paths for where you want files stored to. Looking at the code, you need to adjust the path
to the Bacteria folder, the path to the folder to store the simulated BOC reads (program does not
create a folder), and the path to the folder to store the prediction arrays and confusion matrix arrays
(program does not create a folder).
To plot the confusion matrices for the data, you will need to use the Confusion Matrix code and adjust the path to the folder where you stored the confusion matrix arrays. You can also change the error rate and number of reads to plot your desired confusion matrix for the specified parameters.

The G Plotting code plots the original FBC spectra for the genomes and the bias subtracted spectra for the genomes. You will need to change the location to the BOC files in order to run this code.

D.2.1 Complete Code

```python
import os
import math
import time
import datetime
import dask.dataframe as ddf
import dask.array as da
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.ensemble import GradientBoostingClassifier,
                          ExtraTreesClassifier, RandomForestClassifier
from sklearn.model_selection import ShuffleSplit
from sklearn.linear_model import SGDClassifier,
                         PassiveAggressiveClassifier, Perceptron
from sklearn.naive_bayes import BernoulliNB, GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix
from sklearn.externals import joblib
from dask.distributed import Client, LocalCluster
```

329
from dask_ml.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis
from collections import Counter

''' You will need to change these folder locations to be the
folder where you
want to save the DNA sequences, the files of the created BOC
reads, and
the files of the created prediction arrays and confusion
matrices. '''

# Establishing where things are being run and where things are
being saved
bacteria = 'C:/Usr/Bacteria/'
local_BOC = 'C:/Usr/BOC/'
pred_arrays = 'C:/Usr/NPY/'

def pca_plot(data, name, dna_type, err, read):
    ymax_data = 1.1 * max(data[:, 1])
    ymin_data = 1.1 * min(data[:, 1])
    xmax_data = 1.1 * max(data[:, 0])
    xmin_data = 1.1 * min(data[:, 0])

    fig = plt.figure(figsize=(20, 10))
    ax = plt.subplot()
    ax.spines["right"].set_visible(False)
    ax.spines["top"].set_visible(False)
if dna_type == 'G':
    cb = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '0']
    cc = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '1']
    ceh = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '8']
    cec = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '2']
    cef = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '9']
    ck = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '3']
    cse = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '4']
    csa = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '5']
    cspn = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '6']
    csy = [int_p for int_p, ii in enumerate(name[:10000]) if ii == '7']
```python
ax.scatter(data[cb,0], data[cb,1], s=200, marker='o', c='xkcd:lime green', label='Bacteroides fragilis')
ax.scatter(data[cc,0], data[cc,1], s=200, marker='o', c='xkcd:sienna', label='Campylobacter jejuni')
ax.scatter(data[ceh,0], data[ceh,1], s=200, marker='o', c='xkcd:blue', label='Enterococcus hirae')
ax.scatter(data[cec,0], data[cec,1], s=200, marker='o', c='xkcd:orange', label='Escherichia coli')
ax.scatter(data[cef,0], data[cef,1], s=200, marker='o', c='xkcd:yellow', label='Escherichia fergusonii')
ax.scatter(data[ck,0], data[ck,1], s=200, marker='o', c='xkcd:light pink', label='Klebsiella pneumoniae')
ax.scatter(data[cse,0], data[cse,1], s=200, marker='o', c='xkcd:red', label='Salmonella enterica')
ax.scatter(data[csa,0], data[csa,1], s=200, marker='o', c='xkcd:violet', label='Staphylococcus aureus')
ax.scatter(data[cspn,0], data[cspn,1], s=200, marker='o', c='xkcd:light green', label='Streptococcus pneumoniae')
ax.scatter(data[cspy,0], data[cspy,1], s=200, marker='o', c='xkcd:light cyan', label='Streptococcus pyogenes')

cb = [int(int_p+10000) for int_p, ii in enumerate(name[10000:20000]) if ii == '0']
cc = [int(int_p+10000) for int_p, ii in enumerate(name[10000:20000]) if ii == '1']
ceh = [int(int_p+10000) for int_p, ii in enumerate(name[10000:20000]) if ii == '8']
cec = [int(int_p+10000) for int_p, ii in enumerate(name[10000:20000]) if ii == '2']
```

332
cef = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}9\texttt{')}
ck = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}3\texttt{')}
cse = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}4\texttt{')}
csa = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}5\texttt{')}
cspn = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}6\texttt{')}
cspy = [\texttt{int}(\texttt{int.p} +10000) \texttt{for} \texttt{int.p}, \texttt{ii} \texttt{in} \texttt{enumerate(name [10000:20000]) \texttt{if} \texttt{ii} == \texttt{'}7\texttt{')}
ax.scatter(data[cb,0], data[cb,1], s=200, marker='^', c='\texttt{xkcd\textcolon dark lime green}')
ax.scatter(data[cc,0], data[cc,1], s=200, marker='^', c='\texttt{xkcd\textcolon brown}')
ax.scatter(data[ceh,0], data[ceh,1], s=200, marker='^', c='\texttt{xkcd\textcolon darkblue}')
ax.scatter(data[cec,0], data[cec,1], s=200, marker='^', c='\texttt{xkcd\textcolon orangered}')
ax.scatter(data[cef,0], data[cef,1], s=200, marker='^', c='\texttt{xkcd\textcolon goldenrod}')
ax.scatter(data[ck,0], data[ck,1], s=200, marker='^', c='\texttt{xkcd\textcolon magenta}')
ax.scatter(data[cse,0], data[cse,1], s=200, marker='^', c='\texttt{xkcd\textcolon crimson}')
ax.scatter(data[csa,0], data[csa,1], s=200, marker='^', c='\texttt{xkcd\textcolon plum}')
ax.scatter(data[cspn,0], data[cspn,1], s=200, marker='^', c='\texttt{xkcd\textcolon darkgreen}')
```python
ax.scatter(data[cspy,0], data[cspy,1], s=200, marker='^', c='xkcd:teal')

ax.scatter(data[20000:21000,0], data[20000:21000,1], s=200, marker='^', label=name[20000], c='xkcd:salmon')

ax.scatter(data[21000:22000,0], data[21000:22000,1], s=200, marker='^', label=name[21000], c='xkcd:grey')

save_png = 'GPCA%s%s.png' %(str(err), str(read))

plt.legend(loc='upper center', fontsize=16)

elif dna_type == 'PI':

    ci = [int(int_p+250000) for int_p, ii in enumerate(name) if ii == 0]
    ck = [int(int_p+250000) for int_p, ii in enumerate(name) if ii == 1]
    cn = [int(int_p+250000) for int_p, ii in enumerate(name) if ii == 2]
    cnr = [int(int_p+250000) for int_p, ii in enumerate(name) if ii == 3]
    cv = [int(int_p+250000) for int_p, ii in enumerate(name) if ii == 4]

    ax.scatter(data[ci,0], data[ci,1], s=50, marker='s', c='xkcd:crimson')
    ax.scatter(data[ck,0], data[ck,1], s=50, marker='s', c='xkcd:orangered')
    ax.scatter(data[cn,0], data[cn,1], s=50, marker='s', c='xkcd:darkgreen')
```

334
ax.scatter(data[cv,0], data[cv,1], s=50, marker='s', c='xkcd:darkblue')
ax.scatter(data[cnr,0], data[cnr,1], s=60, marker='D', c='xkcd:grape')

for ii in enumerate(name[:250000]) if ii == 0:
    ci = int_p
for ii in enumerate(name[:250000]) if ii == 1:
    ck = int_p
for ii in enumerate(name[:250000]) if ii == 2:
    cn = int_p
for ii in enumerate(name[:250000]) if ii == 3:
    cnr = int_p
for ii in enumerate(name[:250000]) if ii == 4:
    cv = int_p

i = ax.scatter(data[ci,0], data[ci,1], s=60, marker='o', c='xkcd:red')
k = ax.scatter(data[ck,0], data[ck,1], s=60, marker='o', c='xkcd:orange')
n = ax.scatter(data[cn,0], data[cn,1], s=60, marker='o', c='xkcd:green')
v = ax.scatter(data[cv,0], data[cv,1], s=60, marker='o', c='xkcd:blue')

nr = ax.scatter(data[cnr,0], data[cnr,1], s=50, marker='^', c='xkcd:violet')

save_png = 'P_Ind_PCA_%s_%s.png'%(str(err), str(read))
plt.legend([i,k,n,v,nr], ['IMP', 'KPC', 'NDM', 'VIM', 'No Resistance'], loc='upper center', fontsize=16)
elif dna_type == 'PG':
    cnr = [int(int_p + 250000) for int_p, ii in enumerate(name)
           [250000:500000]) if ii == 0]
    cr = [int(int_p + 250000) for int_p, ii in enumerate(name
           [250000:500000]) if ii == 1]
    ax.scatter(data[cr, 0], data[cr, 1], s=50, marker='s', c=
                'xkcd:crimson')
    ax.scatter(data[cnr, 0], data[cnr, 1], s=60, marker='D', c= '
                'xkcd:darkgreen')

    cnr = [int_p for int_p, ii in enumerate(name[:250000]) if ii == 0]
    cr = [int_p for int_p, ii in enumerate(name[:250000]) if ii == 1]
    r = ax.scatter(data[cr, 0], data[cr, 1], s=60, marker='o',
                    c='xkcd:red')
    nr = ax.scatter(data[cnr, 0], data[cnr, 1], s=50, marker='^'
                    ', c='xkcd:green')

    save_png = 'P_Group_PCA_%s_%s.png' % (str(err), str(read))
    plt.legend([r, nr], ['Resistance', 'No Resistance'], loc='
                upper center', fontsize=16)

else:
    return

plt.rc('xtick', labelsize=20)
plt.rc('ytick', labelsize=20)
plt.savefig(save_png, dpi='figure', bbox_inches='tight')
plt.close()
```python
def str_count(str_part):
    A = str_part.count('A')
    T = str_part.count('T')
    G = str_part.count('G')
    C = str_part.count('C')
    return (A, T, G, C), (T, A, C, G)

def kmer_fingerprints(whole_str, dna_length, kmer_range):
    str_part = (whole_str[ii:ii+dna_length] for ii in range(len(whole_str)-dna_length+1))
    kmer_list = [item for string in str_part for item in str_count(string)]
    kmer_dict = Counter(kmer_list)
    results = [kmer_dict[val] for val in kmer_range]
    return results

def title_extraction(header, resistance):
    heading = header.split(' ')
    seq_record_id = heading[0]
    genus = heading[1]
    if len(heading) >= 3:
        species = heading[2]
        name = genus + ' ' + species
    else:
        name = genus
    return name
```

for word in heading:
    if 'plasmid' in word:
        dna_type = 'Plasmid'
        break
    elif 'genome' in word:
        dna_type = 'Genome'
        break
    elif 'sequence' in word:
        dna_type = 'Sequence'
        break
    else:
        dna_type = ""

bacteria_type = ""
strain = ""
data_add = [ seq_record_id[1:], resistance, name, genus, dna_type, strain, bacteria_type, header ]

return data_add

def multiple_str_check(file):
    header = []
    start = []
    with open(file, 'r') as f:
        for ii, line in enumerate(f):
            if '>' == line[0]:
                header.append(line)
                start.extend([ ii+1 ])
            end = ii+1
def str_extraction(file, start, end, dna_length):
    string = []
    with open(file, 'r') as f:
        for ii, line in enumerate(f):
            if ii >= start and ii < end:
                string.append(line)
            elif ii == end:
                break

    section = ''.join(line.strip() for line in string)
    whole_string = ''.join([section, section[0:dna_length-1]])
    return whole_string

def kmer_main(file, dna_length, kmer_range, data_index, df_WG, df_Plasmid, resistance):
    header, start = multiple_str_check(file)
    if len(header) > 1:
        for ii in range(len(header)):
            whole_str = str_extraction(file, start[ii], start[ii] + 1 - 1, dna_length)
            species_data = title_extraction(header[ii], resistance)
            kmer_results = kmer_fingerprints(whole_str, dna_length, kmer_range)
            species_data.extend(kmer_results)
df_kmer = pd.DataFrame(species_data, index=data_index)

if species_data[4] != 'Plasmid':
    df_Genome = df_Genome.append(df_kmer.T, ignore_index=True)
elif species_data[4] == 'Plasmid':
    df_Plasmid = df_Plasmid.append(df_kmer.T, ignore_index=True)
else:
    whole_str = str_extraction(file, start[0], start[1]−1,
        dna_length)
    species_data = title_extraction(header[0], resistance)
    kmer_results = kmer_fingerprints(whole_str, dna_length,
        kmer_range)
    species_data.extend(kmer_results)
    df_kmer = pd.DataFrame(species_data, index=data_index)
    if species_data[4] != 'Plasmid':
        df_Genome = df_Genome.append(df_kmer.T, ignore_index=False)
    elif species_data[4] == 'Plasmid':
        df_Plasmid = df_Plasmid.append(df_kmer.T, ignore_index=True)

    type_change = {}
for ii, name in enumerate(data_index):
    if ii < 8:
        type_change[name] = 'object'
    else:
        type_change[name]= 'int32'
    df_Genome = df_Genome.astype(type_change)
df_Plasmid = df_Plasmid.astype(type_change)

return df_Genome, df_Plasmid

def kmer_length():
    k = None
    while k is None:
        input_value = input("Please enter DNA segment length ↪ (5–100): ")
    try:
        # try and convert the string input to a number
        k = int(input_value)
        if k < 5:
            print("{input} is not a valid integer, please ↪ enter a valid integer between 5–100".format(input=input_value))
            k = None
        elif k > 100:
            print("{input} is not a valid integer, please ↪ enter a valid integer between 5–100".format(input=input_value))
            k = None
    except ValueError:
        # tell the user off
        print("{input} is not a valid integer, please enter a ↪ valid integer between 5–100".format(input=input_value))
        k = None
    return k

def SERS_values(sers, categories, num_reads, arr, mutate, mutation):
jj = 0
while 1:
    # Randomizing the pyramid array
    mr = np.random.permutation(arr)
    # Adding in the mutations
    mr = np.where(mutate > 0, mutation, mr)
    # Getting the respective kmer counts and dividing the values
    # by the total value to get the frequencies
    for nn in range(len(num_reads)):
        # Setting the limits for how much of the sequence it's using
        if num_reads[nn] > 10000:
            ff = 10000
            max_depth = int(num_reads[nn]/10000)
        else:
            ff = num_reads[nn]
            max_depth = 1
        for mm in range(mr.shape[1]):
            for kk in range(max_depth):
                sers[nn,mm,jj,:] += np.bincount(mr[kk,mm,:ff
                                      ],minlength=categories)

jj+=1
    if jj >= sers.shape[2]:
        break
sers /= np.array(num_reads).reshape((-1,1,1,1))
return sers
def SERS_reads(dna_length, df, group, DNAType, data_categories, bias,
             num_training_samples, num_reads, error_rate, SERS_location):

    # Dividing the number of occurrences of each bin by the total
    # number of occurrences
    df_prob = df.loc[:, data_categories[0]:data_categories[-1]].div(df.loc[:, data_categories[0]:data_categories[-1]].sum(axis=1), axis=0)

    for ii in range(len(df_prob.index)):
        # Getting just the probability values from the kmer counts
        prob = df_prob.iloc[ii, :].values
        # Getting the probability for the largest pyramid of interest
        # and setting the data type
        read = np.random.RandomState(seed=231).choice(len(data_categories), (int(2*max(num_reads)/10000), 10000), p=prob)

        read = np.stack([read for _ in range(len(error_rate))], axis=1)

        # Creating the mutation array
        mutate = np.zeros((int(2*max(num_reads)/10000), len(error_rate), 10000), dtype=np.int16)

        for int_mut, mut in enumerate(error_rate):
            mutations = np.concatenate([np.random.RandomState(seed=123).choice([0, 1], min(num_reads), p=[1-mut, mut]) for _ in range(int(2*max(num_reads)/min(num_reads)))]).reshape((-1, 10000))

            mutate[:, int_mut, :] = mutations
mutation = np.random.RandomState(seed=321).choice(len(data_categories), int(2*max(num_reads)/10000), 10000), p=bias)
mutation = np.stack([mutation for _ in range(len(error_rate))], axis=1)

# Getting the training samples for the species
sers = np.empty((len(num_reads), len(error_rate), num_training_samples, len(data_categories)))
sers_results = SERS_values(sers, len(data_categories), num_reads, read, mutate, mutation)

# Subtracting off the random bias
sers_results -= bias

# Cycling through each mutation array
for mut_int in range(len(error_rate)):
    # Cycling through each pyramid size
    for read_int in range(len(num_reads)):
        # Storing the files as an hdf5 file
        with pd.HDFStore(os.path.join(local_BOC[
            SERS_location], 'SERS%s%s%s%s%s%smer_data.h5' % (str(int(error_rate[mut_int]*100)),
            str(num_reads[read_int]), group, DNAtype,
            str(dna_length))), complevel=9, complev='zlib') as store:
            # Putting it into a pandas dataframe and storing it
            df_SERS = pd.DataFrame(sers_results[read_int, mut_int,:,:], columns=data_categories)
df_SERS['Name'] = [df.iloc[ii,1]]*len(df_SERS).index
store.append('df%s' % (str(ii)), df_SERS, min_itemsize = {'values': 50})

if ii % 10 == 0:
    print(ii)
    print('Saved %s' % (datetime.datetime.now().isoformat()))
return

#dna_length=kmer_length()
dna_length = 10
# Recording the time it takes to run everything
begin = datetime.datetime.now().isoformat()
start = time.perf_counter()
# number of samples per species
num_training_samples = 1000
# error rate
error_rate = [0, 0.01, 0.05, 0.1, 0.25, 0.33, 0.5, 0.75, 0.9, 1]
# number of optical sequencing reads
num_reads = [100, 1000, 10000, 100000, 1000000]

# creating the correct tuples for how many A, T, G and C's are in each bin
kmer_range = [(aa, tt, gg, cc) for aa in range(dna_length + 1) for tt in range(dna_length + 1) for gg in range(dna_length + 1) for cc in range(dna_length + 1) if aa + tt + cc + gg == dna_length]
# setting dna length based bias

```python
bias = np.array([(1/4)**dna_length) * math.factorial(dna_length) / (math.factorial(kmer[0]) * math.factorial(kmer[1]) * math.factorial(kmer[2]) * math.factorial(kmer[3])) for kmer in kmer_range])
```

# creating the categorical labels for storing data

data_categories = ["A%sT%sG%sC%s" % (str(aa), str(tt), str(gg), str(cc)) for aa in range(dna_length + 1) for tt in range(dna_length + 1) for gg in range(dna_length + 1) for cc in range(dna_length + 1) if aa + tt + cc + gg == dna_length]

# creating the labels for the non-numerical information

data_index = [ 'Seq Record ID', 'Resistance', 'Name', 'Genus', 'DNA Type', 'Strain', 'Bacteria Type', 'Notes']

# combining the the non-numerical and categorical labels for the pandas dataframe

data_index.extend(data_categories)

# getting the list of all of the folders and files that have the DNA sequences in them

```python
file_list = [(os.path.join(root, name), root[3:].split('/')[1], name) for root, dirs, files in os.walk(bacteria) for name in files if name.endswith(".txt") or name.endswith(".fna"))
```

# Running through all of the DNA sequence files

```python
for int_file, file in enumerate(file_list):
    # creating the empty dataframes to store the data in
    df_Genome = pd.DataFrame(columns=data_index)
    df_Plasmid = pd.DataFrame(columns=data_index)

    # creating the 10mer data files for the DNA sequences
resistance = file[2][:-4]
df_Genome, df_Plasmid = kmer_main(file[0], dna_length,
    kmer_range, data_index, df_Genome, df_Plasmid, resistance
)
print('Completed %s' % file[2][:-4])

# saving the dataframe
if len(df_Genome.index) > 0:
    df_Genome.to_hdf('PandasDataFrame%s_Genome%smer_data.h5
        % (file[1], str(dna_length)), 'df%s' % (int_file),
        mode='a', format='table')
if len(df_Plasmid.index) > 0:
    df_Plasmid.to_hdf('PandasDataFrame%s_Plasmid%smer_data.
        % (file[1], str(dna_length)), 'df%s' % (int_file)
        , mode='a', format='table')
print('file saved')

del(df_Genome)
del(df_Plasmid)

# Running the code for creating the simulated SERS data
for file in os.listdir(os.getcwd()):
    """ Splitting file name up into parts to determine if its
    genomic dna and to get the dna length """
    file_split = file[:3].split('_')
    if file_split[0] == 'PandasDataFrame':
        # get dna length from file name
        # dna_length = int(file_split[3].replace('mer', ''))
        # collecting the dataframe from the stored file for analysis
df = ddf.read_hdf(os.path.join(os.getcwd(), file), 'df').

compute()
SERS_reads(dna_length, df, file_split[1], file_split[2],
          data_categories, bias, num_training_samples, num_reads,
          error_rate, local_BOC)

end = time.perf_counter()
print('# of hours to run code: %s' % ((end - start)/3600))
print(datetime.datetime.now().isoformat())

# Getting the list of all the data files that need to be tested
file_list = [file for dirr in local_BOC for file in os.listdir(dirr)]

# Getting the list of the files for the specific reads and errors
working_genome_list = [file for error in error_rate for reads in num_reads for file in file_list if '%s%s' % (int(100 * error), reads) in file and 'Genome' in file]
working_plasmid_list = [file for error in error_rate for reads in num_reads for file in file_list if '%s%s' % (int(100 * error), reads) in file and 'Plasmid' in file]

# Getting each file in the list of files to use
train_file = []
test_file = []
extra_file = []
for file in working_genome_list:
    if file.split('_')[3] == 'Training':
        train_file.append(file)
    elif file.split('_')[3] == 'Testing':
        test_file.append(file)
elif file.split('_')[3] == 'Extras':
    extra_file.append(file)

# Number of testing plasmids
num_plasmid = 100

# Number of training bacterial species
bac_type = 10

# Number of training plasmid types
plasmid_type = 5

# Number of SERS readings to grab at a time from the training samples
num_SERS = 50

# Number of cross fold validation
cfv = 10

# Number of samples to train on
samples = 250000

# Determining how many times you want to test the ML algorithm against different unseen sets of plasmids/genomes
unseen_tests = 2

# Split for looking at resistance vs non-resistance grouping for non-resitant plasmids
unseen_group_split = ShuffleSplit(unseen_tests, None, num_plasmid, 123)

# Creating the list of all of the classifiers for storing things
classifier_list = ['SGD', 'Perceptron', 'Passive-Aggressive', 'Neural Network', 'GNB', 'BNB', 'RF', 'ET', 'GB', 'PCA_svd', 'LDA', 'QDA']
g_class = ['Bacteroides fragilis', 'Campylobacter jejuni', 'Escherichia coli', 'Klebsiella pneumoniae', 'Salmonella enterica', 'Staphylococcus aureus', 'Streptococcus pneumoniae', 'Streptococcus pyogenes', 'Enterococcus hirae', 'Escherichia fergusonii', 'Klebsiella aerogenes', 'Mycobacterium tuberculosis']
g_abr_class = ['B. fragilis', 'C. jejuni', 'E. coli', 'K. pneumoniae', 'S. enterica', 'S. aureus', 'S. pneumoniae', 'S. pyogenes', 'E. hirae', 'E. fergusonii', 'K. aerogenes', 'M. tuberculosis']
p_class = ['IMP', 'KPC', 'NDM', 'No Resistance', 'VIM']
p_g_class = ['No Resistance', 'Resistance']

# Initializing dask to run things in parallel
with LocalCluster(processes=False) as cluster, Client(cluster) as client:

# Retrieving whether the file is genomic or plasmid
file_type = 'Genome'

# Opening up each of the HDF5 files in the working list
store_train = ddf.read_hdf([os.path.join(local_BOC, file) for file in train_file], 'df*')
store_test = ddf.read_hdf([os.path.join(local_BOC, file) for file in test_file], 'df*')
store_extra = ddf.read_hdf([os.path.join(local_BOC, file) for file in extra_file], 'df*')

# Splitting the files into the unseen sets and the training sets
train_set = store_train.drop('Name', axis='columns').to_dask_array(True)
unseen_test_set = store_test.drop('Name', axis='columns').to_dask_array(True)
check_set = store_extra.drop('Name', axis='columns').to_dask_array(True)

bac_train = np.char.array(store_train.loc[:, 'Name'].compute().tolist())
bac_test = np.char.array(store_test.loc[:, 'Name'].compute().tolist())
bac_ex = np.char.array(store_extra.loc[:, 'Name'].compute().tolist())

# Converting the names into numbers for the analysis
bac_train = np.where(bac_train.rfind('_') != bac_train.find('_'), bac_train.rpartition('_')[0].replace('_', '', ''), bac_train.replace('_', '', ''))
bac_train = np.where(bac_train == 'Bacteroides fragilis', g_class[0], bac_train)
bac_train = np.where(bac_train == 'Streptococcus pneumonia', g_class[6], bac_train)
bac_test = np.where(bac_test.rfind('_') != bac_test.find('_'), bac_test.rpartition('_')[0].replace('_', '', ''), bac_test.replace('_', '', ''))
bac_ex = np.where(bac_ex.rfind('_') != bac_ex.find('_'), bac_ex.rpartition('_')[0].replace('_', '', ''), bac_ex.replace('_', '', ''))
bac_ex = np.where(bac_ex == 'Enterobacter aerogenes', g_class[10], bac_ex)
bac_train = np.where(bac_train == g_class[0], 0, bac_train)
bac_train = np.where(bac_train == g_class[1], 1, bac_train)
bac_train = np.where(bac_train == g_class[2], 2, bac_train)
bac_train = np.where(bac_train == g_class[3], 3, bac_train)
bac_train = np.where(bac_train == g_class[4], 4, bac_train)
bac_train = np.where(bac_train == g_class[5], 5, bac_train)
bac_train = np.where(bac_train == g_class[6], 6, bac_train)
bac_train = np.where(bac_train == g_class[7], 7, bac_train)
bac_train = np.where(bac_train == g_class[8], 8, bac_train)
bac_train = np.where(bac_train == g_class[9], 9, bac_train)
bac_train = bac_train.astype('int')

bac_test = np.where(bac_test == g_class[0], 0, bac_test)
bac_test = np.where(bac_test == g_class[1], 1, bac_test)
bac_test = np.where(bac_test == g_class[2], 2, bac_test)
bac_test = np.where(bac_test == g_class[3], 3, bac_test)
bac_test = np.where(bac_test == g_class[4], 4, bac_test)
bac_test = np.where(bac_test == g_class[5], 5, bac_test)
bac_test = np.where(bac_test == g_class[6], 6, bac_test)
bac_test = np.where(bac_test == g_class[7], 7, bac_test)
bac_test = np.where(bac_test == g_class[8], 8, bac_test)
bac_test = np.where(bac_test == g_class[9], 9, bac_test)
bac_test = bac_test.astype('int')

# Creating a testing data array for the unseen tests and the
# training data

train_data = da.concatenate([train_set[int(num_SERS*ii+jj*}
                          num_training_samples+kk*bac_type*num_training_samples):
                          int(num_SERS*(ii+1)+jj*num_training_samples+kk*bac_type
                          *num_training_samples) for kk in range(int(len(}
error_rate) * len(num_reads)) for ii in range(int(num_training_samples / num_SERS)) for jj in range(bac_type)], axis=0).rechunk({0: 500})

unseen_data = da.concatenate([unseen_test_set[int(num_SERS * ii + jj * num_training_samples + kk * bac_type * num_training_samples): int(num_SERS * (ii + 1) + jj * num_training_samples + kk * bac_type * num_training_samples)] for kk in range(int(len(error_rate) * len(num_reads))) for ii in range(int(num_training_samples / num_SERS)) for jj in range(bac_type)], axis=0).rechunk({0: 500})

# Creating the labels/categories for the data

y_train_data = da.concatenate([bac_train[int(num_SERS * ii + jj * num_training_samples + kk * bac_type * num_training_samples): int(num_SERS * (ii + 1) + jj * num_training_samples + kk * bac_type * num_training_samples)] for kk in range(int(len(error_rate) * len(num_reads))) for ii in range(int(num_training_samples / num_SERS)) for jj in range(bac_type)], axis=0).rechunk({0: 500})

y_unseen_data = da.concatenate([bac_test[int(num_SERS * ii + jj * num_training_samples + kk * bac_type * num_training_samples): int(num_SERS * (ii + 1) + jj * num_training_samples + kk * bac_type * num_training_samples)] for kk in range(int(len(error_rate) * len(num_reads))) for ii in range(int(num_training_samples / num_SERS)) for jj in range(bac_type)], axis=0).rechunk({0: 500})

print('Moving on to classifying')

# Cycling through the 24 combinations of error rate and number of reads

for int_err, error in enumerate(error_rate):
# Recording the time it takes to run everything
start1 = time.perf_counter()

for int_read, reads in enumerate(num_reads):
    # Creating a pandas dataframe for the results
    results = pd.DataFrame(index=np.arange(cfv))
    # Getting the values for the unseen testing
    xu, yu = unseen_data[int(bac_type*]
        num_training_samples*(int_read+len(num_reads)*
        int_err)) : int(bac_type*num_training_samples*(1+
        int_read+len(num_reads)*int_err))].compute(),
        y_unseen_data[int(bac_type*num_training_samples
        *(int_read+len(num_reads)*int_err)) : int(
        bac_type*num_training_samples*(1+int_read+len(
        num_reads)*int_err))].compute()

    # Getting the values for the extras testing
    xe, ye = check_set[int(2*num_training_samples*(
        int_read+len(num_reads)*int_err)) : int(2*
        num_training_samples*(1+int_read+len(num_reads)
        *int_err))].compute(), np.tile(np.array
        ([0,1,2,3,4,5,6,7,8,9]), int(np.ceil(2*
        num_training_samples/9)))[ : int(2*
        num_training_samples)]

    # Creating the split for the 90% train, 10% validate
    split for the training set
    for cv in range(cfv):
        # Here are the classifiers that support the 'partial_fit' method
        partial_fit_classifiers = {

354
'SGD': SGDClassifier(max_iter=1000, tol=1e-3, random_state=123),
'Perceptron': Perceptron(max_iter=1000, tol=1e-3, random_state=123),
'Passive-Aggressive':
  PassiveAggressiveClassifier(max_iter=1000, tol=1e-3, random_state=123),
'Neural Network': MLPClassifier(max_iter=1000, random_state=123),
'GNB': GaussianNB(),
'BNB': BernoulliNB(),
'RF': RandomForestClassifier(n_estimators = 10, max_features=None, bootstrap=False,
  random_state=123, warm_start=True),
'ET': ExtraTreesClassifier(n_estimators = 10,
  max_features=None, bootstrap=False,
  random_state=123, warm_start=True),
'GB': GradientBoostingClassifier(n_estimators = 10,
  max_features=None, random_state =123, warm_start=True),
'LDA': LinearDiscriminantAnalysis(solver='svd'),
'QDA': QuadraticDiscriminantAnalysis()
}

# Timing how long it takes to run through all models
	tic = time.perf_counter()

# Getting the split for this cv
	x_train, x_test, y_train, y_test =
  train_test_split(train_data[int(bac_type*
  num_training_samples*(int_read+len)
num reads) * int error): int (bac_type*
num training samples * (1 + int read + len(
num reads) * int error)), y_train_data [int(
bac_type * num training samples * (int read + len
num reads) * int error)): int (bac_type*
num training samples * (1 + int read + len(
num reads) * int error)), random_state=123+cv)

# Getting the values for the model for training
x, y = x_train.compute(), y_train.compute()

# Getting the values for the model for testing
xt, yt = x_test.compute(), y_test.compute()

np.save(os.path.join(pred_arrays, '%s %s Genome
Train-test values %s.npy' % (int(error*100)
, reads, cv)), yt)

np.save(os.path.join(pred_arrays, '%s %s Genome
Unseen values %s.npy' % (int(error*100),
reads, cv)), yu)

# Iterating through the different models
for cls_name, cls in partial_fit_classifiers:
    items():

    # timing each classifier
tick1 = time.perf_counter()

    # Using the client to parallelize the fits
    with joblib.parallel_backend('dask'):
        # update estimator with data
        cls.fit(x, y)

        # getting how long it takes to fit the data
        fit_time = time.perf_counter() - tick1

        # scoring the testing set
        acc = cls.score(xt, yt)
# getting the prediction arrays

```python
if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive Aggressive':
pred = cls.decision_function(xt)
else:
pred = cls.predict_proba(xt)
```

# Recording results

```python
if cv == 0:
    # model fitting stats
    results.loc[cv, '%s %s %s Genome Train Time' % (error, reads, cls_name)] =
    np.mean(fit_time)
    results.loc[cv, '%s %s %s Genome Train Sample Number' % (error, reads, cls_name)] = len(y)

    # accumulate test accuracy stats
    results.loc[cv, '%s %s %s Genome Train Accuracy' % (error, reads, cls_name)] =
    np.mean(acc)

# Calculating Confusion Matrix
```

```python
conf_mat = confusion_matrix(yt, pred.argmax(axis=1))
np.save(os.path.join(pred_arrays, '%s %s %s Genome Prediction Array Train %s.npy' %
                     (int(error*100), reads, cls_name, cv)), pred)
np.save(os.path.join(pred_arrays, '%s %s %s Genome Confusion Matrix Train %s.npy' %
                     (int(error*100), reads, cls_name, cv)), conf_mat)
```
# Printing the time the model testing has been running
print('Train set tested. Total time model testing has been running %s minutes' % ((time.perf_counter() - tick) / 60))

# timing each classifier
tick1 = time.perf_counter()

# Using the client to parallelize the fits
with joblib.parallel_backend('dask'):
    # update estimator with data
    acc = cls.score(xu, yu)
    if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive - Aggressive':
        pred = cls.decision_function(xu)
        check = cls.decision_function(xe)
    else:
        pred = cls.predict_proba(xu)
        check = cls.predict_proba(xe)

test_time = time.perf_counter() - tick1

if cv == 0:
    # model fitting stats
    results.loc[cv, '%s %s %s Genome Test Time' % (error, reads, cls_name)] = np.mean(test_time)

    results.loc[cv, '%s %s %s Genome Test Number' % (error, reads, cls_name)] = len(yu)
# accumulate test accuracy stats

```python
cv, genome_test_accuracy = np.mean(acc)
```

# Calculating Confusion Matrix

```python
cnf_mat = confusion_matrix(yu, pred.argmax(axis=1))
np.save(os.path.join(pred_arrays, f'Genome Prediction Array Test {error*100}%', reads, cls_name, cv), pred)
np.save(os.path.join(pred_arrays, f'Genome Confusion Matrix Test {error*100}%', reads, cls_name, cv), cnf_mat)
cnf_mat = confusion_matrix(ye, check.argmax(axis=1))
np.save(os.path.join(pred_arrays, f'Genome Extras Prediction Array Test {error*100}%', reads, cls_name, cv), check)
np.save(os.path.join(pred_arrays, f'Genome Extras Confusion Matrix Test {error*100}%', reads, cls_name, cv), cnf_mat)
```

```python
print('Took %s minutes to fit, validate and test all models for 1 cross-fold validation' % ((time.perf_counter() - tick)/60))
print('%s fold Cross Validation finished' % (cv))
```
if cv == 0:
    # Timing how long it takes to run through all
discriminant analysis models
    tick = time.perf_counter()

    ## Creating component analysis data
    cls = PCA(random_state=123)
    g_pca_train = np.array(cls.fit_transform(np.concatenate((x, xt), axis=0)))
    g_pca_unseen = np.array(cls.transform(xu))
    g_pca_check = np.array(cls.transform(xe))
    g_pca = np.concatenate((g_pca_train, g_pca_unseen, g_pca_check), axis=0)
    np.save(os.path.join(pred_arrays, '%s %s Genome PCA.npy' % (int(100*error), reads)), g_pca)

    y_name = np.concatenate((y, yt, yu), axis=0)
    g_name = np.concatenate((y_name, bac_ex[int(2*num_training_samples*(int_read+len(num_reads)*int_err))/int(2*num_training_samples*(1+int_read+len(num_reads)*int_err))], axis=0)
    np.save(os.path.join(pred_arrays, '%s %s Genome PCA Names.npy' % (int(100*error), reads)), g_name)

    pca_plot(g_pca[:, :2], g_name, 'G', int(100*error), reads)

    # Saving the results

    360
results.to_csv('%s_ML_Results_%s_%s.csv' % (file_type, reads, int(100*error)))

print('Results saved')

print('%s error rate and %s number of reads tests finished' % (error, reads))

print('Took %s minutes to run all number of reads tests' % ((time.perf_counter() - start1)/60))

print('Genome Finished. Taken %s hours to run code.' % ((time.perf_counter() - start)/(3600)))

print(datetime.datetime.now().isoformat())

# Creating a dictionary for the results
pred = {}

for cls_name in classifier_list:
    pred[cls_name] = []

# Testing the machine learning techniques on different sets of data being the never before seen test set
for int_data in range(unseen_tests):
    # Retrieving wether the file is genomic or plasmid
    file_type = 'Plasmid'
    # Determining how long this takes
    tick = time.perf_counter()
    # Splitting the files into the unseen sets and the training sets
    train_set = []
    unseen_test_set = []
    train_group_set = []
    unseen_group_set = []
    file_info = []
# Getting each file in the list of files to use
for int_list, file in enumerate(working_plasmid_list):
    # Opening up each of the HDF5 files in the working list
    store = ddf.read_hdf(os.path.join(local_BOC, file), 'df*')
    # Getting the file type and the number of plasmids/
genomes in the file
    file_info.append((file.split('_')[3], store.npartitions))
    # Creating a split of 100 training samples and the rest
    for testing as never before seen
    split = unseen_split.split(np.zeros((file_info[int_list][1], 1)))
    # Grabbing 400 training samples for Group testing for the
    # NonResistant sample
    if file_info[int_list][0] == 'NonResistant':
        split_group = unseen_group_split.split(np.zeros((
            file_info[int_list][1], 1))))
    # Getting the split of data for this learning test with
    # never before seen sets
    for int_next in range(int_data+1):
        # Getting the correct data split
        train, unseen_ind = next(split)
        if file_info[int_list][0] == 'NonResistant':
            train_group, unseen_group = next(split_group)
        # Storing just the values of the data split
        train_set.append(store.drop('Name', axis='columns').
                         partitions[train].to_dask_array(True))
unseen_test_set.append(store.drop(’Name’, axis=’
columns’).partitions[unseen_ind].to_dask_array(
True))
if file_info[int_list][0] == ’NonResistant’:
    train_group_set.append(store.drop(’Name’, axis=’
columns’).partitions[train_group].
to_dask_array(True))
unseen_group_set.append(store.drop(’Name’, axis=’
columns’).partitions[unseen_group].
to_dask_array(True))
else:
    train_group_set.append(store.drop(’Name’, axis=’
columns’).partitions[train].to_dask_array(
True))
unseen_group_set.append(store.drop(’Name’, axis=’
columns’).partitions[unseen_ind].
to_dask_array(True))
print(’Took %s hours to extract data’ % ((time.
    perf_counter()–tick)/3600))
# Removing unwanted variables
del(train)
del(unseen_ind)
del(split)
del(train_group)
del(unseen_group)
del(split_group)
del(int_list)
del(int_next)

# Getting what set of files to grab
int_set = 0
# Cycling through the 24 combinations of error rate and
→ number of reads
for error in error_rate:
# Recording the time it takes to run everything
→ start1 = time.perf_counter()
for reads in num_reads:
# Creating a pandas dataframe for the results
→ results = pd.DataFrame(index = np.arange(cfv))
→ results_group = pd.DataFrame(index = np.arange(
→ → cfv))
# Creating lists to save the dictionaries into for
→ cross fold validation
→ cv_stats = []

# Creating a testing data array for the unseen tests
→ and the training data
→ train_data = da.concatenate([[arr[int(num_SERS*ii +
→ jj*num_training_samples):int(num_SERS*(ii +1)+jj*num_training_samples)] for ii in
→ range(int(num_training_samples/num_SERS))
→ for arr in train_set[int(int_set* 
→ plasmid_type):int((int_set+1)*plasmid_type)
→ ] for jj in range(len(arr.chunks[0])),axis 
→ =0).rechunk({0: 25000})
→ unseen_data = da.concatenate([[arr[int(num_SERS*ii +
→ jj*num_training_samples):int(num_SERS*(ii +1)+jj*num_training_samples)] for ii in
→ range(int(num_training_samples/num_SERS))
→ for arr in unseen_test_set[int(int_set* 
→ plasmid_type):int((int_set+1)*plasmid_type)
→ ] for jj in range(len(arr.chunks[0])),axis 
→ =0).rechunk({0: 25000})

→ import time
→ start1 = time.perf_counter()
→ for reads in num_reads:
→ results = pd.DataFrame(index = np.arange(cfv))
→ results_group = pd.DataFrame(index = np.arange(
→ cfv))
→ # Creating lists to save the dictionaries into for
→ cross fold validation
→ cv_stats = []

→ train_data = da.concatenate([[arr[int(num_SERS*ii +
→ jj*num_training_samples):int(num_SERS*(ii +1)+jj*num_training_samples)] for ii in
→ range(int(num_training_samples/num_SERS))
→ for arr in train_set[int(int_set* 
→ plasmid_type):int((int_set+1)*plasmid_type)
→ ] for jj in range(len(arr.chunks[0])),axis 
→ =0).rechunk({0: 25000})
→ unseen_data = da.concatenate([[arr[int(num_SERS*ii +
→ jj*num_training_samples):int(num_SERS*(ii +1)+jj*num_training_samples)] for ii in
→ range(int(num_training_samples/num_SERS))
→ for arr in unseen_test_set[int(int_set* 
→ plasmid_type):int((int_set+1)*plasmid_type)
→ ] for jj in range(len(arr.chunks[0])),axis 
→ =0).rechunk({0: 25000})
plasmid_type): int((int_set+1)*plasmid_type)
] for jj in range(len(arr.chunks[0])), axis=0).rechunk({0: 25000})

train_group_data = da.concatenate([arr[int(num_SERS*ii+jj*num_training_samples):int(num_SERS*(ii+1)+jj*num_training_samples)]
for ii in range(int(num_training_samples/num_SERS)) for arr in train_group_set[int(int_set*plasmid_type):int((int_set+1)*plasmid_type)] for jj in range(len(arr.chunks[0])), axis=0).rechunk({0: 25000})

unseen_group_data = da.concatenate([arr[int(num_SERS*ii+jj*num_training_samples):int(num_SERS*(ii+1)+jj*num_training_samples)]
for ii in range(int(num_training_samples/num_SERS)) for arr in unseen_group_set[int(int_set*plasmid_type):int((int_set+1)*plasmid_type)] for jj in range(len(arr.chunks[0])), axis=0).rechunk({0: 25000})

# Creating the labels/categories for the data
y_train_data = da.concatenate([np.ones(int(num_SERS))*int_arr for ii in range(int(num_training_samples/num_SERS)) for int_arr, arr in enumerate(train_set[int(int_set*plasmid_type):int((int_set+1)*plasmid_type)]]) for jj in range(len(arr.chunks[0])), axis=0).rechunk({0: 25000})

y_unseen_data = da.concatenate([np.ones(int(num_SERS))*int_arr for ii in range(int(num_training_samples/num_SERS)) for int_arr,
for jj in range(len(arr.chunks[0])), axis=0).rechunk({0: 25000})

y_train_group = da.concatenate([np.ones(int(
    num_SERS)) if file_info[int_arr][0] != 'NonResistant' else np.zeros(int(num_SERS))
    for ii in range(int(num_training_samples/
    num_SERS)) for int_arr, arr in enumerate(
    train_group_set[int(int_set*plasmid_type):
    int((int_set+1)*plasmid_type)]) for jj in
    range(len(arr.chunks[0])), axis=0).rechunk
    ({0: 25000})

y_unseen_group = da.concatenate([np.ones(int(
    num_SERS)) if file_info[int_arr][0] != 'NonResistant' else np.zeros(int(num_SERS))
    for ii in range(int(num_training_samples/
    num_SERS)) for int_arr, arr in enumerate(
    unseen_group_set[int(int_set*plasmid_type):
    int((int_set+1)*plasmid_type)]) for jj in
    range(len(arr.chunks[0])), axis=0).rechunk
    ({0: 25000})

# Upping the set of files to grab
int_set += 1

# Getting the # of times the model needs to be
iterated on
if len(unseen_data) % samples != 0:
    model_unseen_count = int(len(unseen_data)/
samples) + 1
else:
    model_unseen_count = int(len(unseen_data) / samples)

# Creating the split for the 90% train, 10% validate
for cv in range(cfv):
    # Here are the classifiers that support the 'partial_fit' method
    partial_fit_classifiers = {
        'SGD': SGDClassifier(random_state=123),
        'Perceptron': Perceptron(tol=1e-3, random_state=123),
        'Passive-Aggressive': PassiveAggressiveClassifier(tol=1e-3, random_state=123),
        'Neural Network': MLPClassifier(random_state=123),
        'GNB': GaussianNB(),
        'BNB': BernoulliNB(),
        'RF': RandomForestClassifier(n_estimators=10, max_features=None, bootstrap=False, random_state=123, warm_start=True),
        'ET': ExtraTreesClassifier(n_estimators=10, max_features=None, bootstrap=False, random_state=123, warm_start=True),
'GB': GradientBoostingClassifier(
    n_estimators=10, max_features=None,
    random_state=123, warm_start=True)
}

# Timing how long it takes to run through all models

tick = time.perf_counter()

# Getting the split for this cv

x_train, x_test, y_train, y_test =
    train_test_split(train_data,
                      y_train_data, random_state=123+cv)

# Getting the # of times the model needs to be iterated on

if len(x_train) % samples != 0:
    model_fit_count = int(len(x_train)/
                          samples) + 1
else:
    model_fit_count = int(len(x_train)/
                          samples)

if len(x_test) % samples != 0:
    model_test_count = int(len(x_test)/
                          samples) + 1
else:
    model_test_count = int(len(x_test)/
                          samples)

# Iterating through the # of fits

for int_model in range(model_fit_count):

    # Getting the values for the model for testing

    if int_model != model_fit_count - 1:
x, y = x_train[samples*int_model:
    samples*(int_model+1)].compute
    (), y_train[samples*int_model:
    samples*(int_model+1)].compute
    ()
else:
    x, y = x_train[samples*int_model:].
    compute(), y_train[samples*
    int_model:].compute()

# Creating the timing variable
    fit_time = [np.zeros(model_fit_count) for
    ii in partial_fit_classifiers]
# Iterating through the different models
    for int_cls, (cls_name, cls) in enumerate
    (partial_fit_classifiers.items()):
# timing each classifier
    tick1 = time.perf_counter()
# Using the client to parallelize the
    fits
    with joblib.parallel_backend('dask'):
# update estimator with data
    if cls_name == 'RF' or cls_name
    == 'ET' or cls_name == 'GB':
    cls.fit(x, y)
    if int_model !=
    model_fit_count - 1:
        cls.n_estimators += 1
    else:
```python
classes.partial_fit(x, y, classes
        = np.array(plasmid_type
        ))
fit_time[int_cls][int_model] = time.
        perf_counter() - tick1

# Printing the time it took to fit all of the
# models
print(’Took %s minutes to fit all models’ %
        ((time.perf_counter() - tick) / 60))

# Creating the test variables
acc = [np.zeros(model_test_count) for ii in
        range(len(partial_fit_classifiers))]

# Iterating through the # of tests
for int_model in range(model_test_count):
    if int_model != model_test_count - 1:
        xt, yt = x_test[samples*int_model:
        samples*(int_model+1)].compute
    else:
        xt, yt = x_test[samples*int_model:].
        compute(), y_test[samples*
        int_model:].compute()

# Iterating through the different models
for int_cls, (cls_name, cls) in enumerate
        (partial_fit_classifiers.items()):
    # Using the client to parallelize the
    fits
    with joblib.parallel_backend(’dask’):
```

370
# update estimator with data
acc[int_cls][int_model] = cls.
score(xt, yt)

if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive-Aggressive':
pred[cls_name].append(cls.
decision_function(xt))
else:
pred[cls_name].append(cls.
predict_proba(xt))

# Creating labels for AUROC
for int_cls, (cls_name, cls) in enumerate(
partial_fit_classifiers.items()):

# model fitting stats
results.loc[cv, '%s %s %s Individual Train Time' % (error, reads, cls_name)] = np.mean(fit_time[int_cls])
results.loc[cv, '%s %s %s Individual Train Sample Number' % (error, reads, cls_name)] = len(y_train)

# accumulate test accuracy stats
results.loc[cv, '%s %s %s Individual Train Accuracy' % (error, reads, cls_name)] = np.mean(acc[int_cls])

# Calculating Confusion Matrix
conf_mat = confusion_matrix(y_test,compute(), np.concatenate(pred[
cls_name ]).argmax(axis=1))
np.save(os.path.join(pred_arrays, '%s %s 
↪ %s Plasmid Prediction Array Train % 
↪ s.npy' % (int(error*100), reads, 
↪ cls_name, cv)), np.concatenate(pred 
↪ [cls_name]))

np.save(os.path.join(pred_arrays, '%s %s 
↪ %s Plasmid Confusion Matrix Train % 
↪ s.npy' % (int(error*100), reads, 
↪ cls_name, cv)), conf_mat)
pred[cls_name] = []

# Printing the time the model testing has been running
print('Train set tested. Total time model testing has been running %s minutes' % ((time.perf_counter()-tick)/60))

# Creating the test variables
acc_t = [np.zeros(model_unseen_count) for ii in partial_fit_classifiers]
test_time = [np.zeros(model_unseen_count) for ii in partial_fit_classifiers]

# Iterating through the # of tests
for int_model in range(model_unseen_count):
  if int_model != model_unseen_count - 1:
    xu, yu = unseen_data[samples* 
    ↪ int_model:samples*(int_model+1) 
    ↪ ].compute(), y_unseen_data[ 
    ↪ samples*int_model:samples* ( 
    ↪ int_model+1)].compute()
  else:
xu, yu = unseen_data[samples*int_model].compute(),
→ y_unseen_data[samples*int_model].compute()
# Iterating through the different models
for int_cls, (cls_name, cls) in enumerate(partial_fit_classifiers.items()):
  # timing each classifier
  tick1 = time.perf_counter()
  # Using the client to parallelize the fits
  with joblib.parallel_backend('dask'):
    # update estimator with data
    acc_t[int_cls][int_model] = cls.
    score(xu, yu)
    if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive-Aggressive':
      pred[cls_name].append(cls.
      decision_function(xu))
    else:
      pred[cls_name].append(cls.
      predict_proba(xu))
    test_time[int_cls][int_model] = time.
    perf_counter() - tick1
# Creating labels for AUROC
for int_cls, (cls_name, cls) in enumerate(partial_fit_classifiers.items()):
  # model fitting stats
results.loc[cv, '%s %s %s Individual Test Time' % (error, reads, cls_name)] = np.mean(test_time[int_cls])
results.loc[cv, '%s %s %s Individual Test Number' % (error, reads, cls_name)] = len(y_unseen_data)

# accumulate test accuracy stats
results.loc[cv, '%s %s %s Individual Test Accuracy' % (error, reads, cls_name)] = np.mean(acc_t[int_cls])

# Calculating Confusion Matrix
conf_mat = confusion_matrix(y_unseen_data.compute(), np.concatenate(pred[cls_name]).argmax(axis=1))
np.save(os.path.join(pred_arrays, '%s %s %s Plasmid Prediction Array Test %s.npy' % (int(error*100), reads, cls_name, cv)), np.concatenate(pred[cls_name]))

np.save(os.path.join(pred_arrays, '%s %s %s Plasmid Confusion Matrix Test %s.npy' % (int(error*100), reads, cls_name, cv)), conf_mat)
pred[cls_name] = []

print('Took %s minutes to fit, validate and test all models for 1 cross-fold validation' % ((time.perf_counter() - tick)/60))
print('%s fold Cross Validation finished' % (cv))
# Timing how long it takes to run through all
discriminant analysis models

tick0 = time.perf_counter()

## Creating component analysis data on the first 500
SERS sequences out of the 1000

```python
x_train, x_test, y_train, y_test =
    train_test_split(train_data, y_train_data, random_state=123)
```

```python
xt, yt = x_test.compute(), y_test.compute()
x, y = x_train[:int(samples - len(xt))].compute(),
    y_train[:int(samples - len(yt))].compute()
xu, yu = unseen_data[:samples].compute(),
    y_unseen_data[:samples].compute()
```

```python
classifier = {
    'PCA_svd': PCA(random_state=123),
    'LDA': LinearDiscriminantAnalysis(solver=
        'svd'),
    'QDA': QuadraticDiscriminantAnalysis()
}
```

```python
for cls_name, cls in classifier.items():
    if cls_name == 'PCA_svd':
        pi_pca_train = np.array(cls.fit_transform
            (np.concatenate((x, xt), axis=0)))
        pi_pca_unseen = np.array(cls.transform(xu
            ))
        pi_pca = np.concatenate((pi_pca_train,
            pi_pca_unseen), axis=0)
        pi_name = np.concatenate((y, yt, yu)).
            astype('int8')
```

375
pi_var = np.array([.32, .32, .36])

np.save(os.path.join(pred_arrays, '%s %s Plasmid PCA.npy' % (int(100*error), reads)), pi_pca)
pca_plot(pi_pca[:, :2], pi_name, 'PI', int((100*error), reads))

else:
    # fit model
    cls.fit(x, y)

# CV Testing
results.loc[0, '%s %s %s Individual Train Accuracy' % (error, reads, cls_name)] = np.mean(cls.score(xt, yt))
pred[cls_name].append(cls.predict_proba(xt))

# Calculating Confusion Matrix
conf_mat = confusion_matrix(yt, np.concatenate(pred[cls_name]).argmax(axis=1))

np.save(os.path.join(pred_arrays, '%s %s Plasmid Prediction Array Train %s.npy' % (int(error*100), reads, cls_name, cv)), np.concatenate(pred[cls_name]))
np.save(os.path.join(pred_arrays, '%s %s Plasmid Confusion Matrix Train %s.npy' % (int(error*100), reads, cls_name, cv)), conf_mat)
pred[cls_name] = []

# Unknown Testing
Accuracy = np.mean(cls.score(xu, yu))

# Calculating Confusion Matrix

cnf_mat = confusion_matrix(yu, np.concatenate(pred[cls_name]).argmax(axis=1))

np.save(os.path.join(pred_arrays, f'{int(error*100)} Plasmid Prediction Array Test.npy', reads, cls_name, cv), np.concatenate(pred[cls_name]))

np.save(os.path.join(pred_arrays, f'{int(error*100)} Plasmid Confusion Matrix Test.npy', reads, cls_name, cv), cnf_mat)

pred[cls_name] = []

print('Took %s minutes to fit, validate and test all discriminant analysis models' % ((time.perf_counter()-tick0)/60))

print('Took %s hours to run all models' % ((time.perf_counter()-tick)/3600))

# Saving the results

results.to_csv('ML Results.csv' % (file_type, reads, int(100*error), int_data))

print('Results saved')

print(datetime.datetime.now().isoformat())
# Getting the # of times the group model needs to be iterated on

```python
if len(unseen_group_data) % samples != 0:
    model_unseen_count = int(len(unseen_group_data)/samples) + 1
else:
    model_unseen_count = int(len(unseen_group_data)/samples)
```

# Creating the split for the 90% train, 10% validate split for the training set

```python
for cv in range(cfv):
    # Here are the classifiers that support the `partial_fit` method
    partial_fit_classifiers = {
        'SGD': SGDClassifier(random_state=123),
        'Perceptron': Perceptron(tol=1e-3,
                                 random_state=123),
        'Passive-Aggressive':
                                 PassiveAggressiveClassifier(tol=1e
                                                  -3,random_state=123),
        'Neural Network': MLPClassifier(
                                 random_state=123),
        'GNB': GaussianNB(),
        'BNB': BernoulliNB(),
        'RF': RandomForestClassifier(n_estimators =10,max_features=None, bootstrap=
                                     False ,random_state=123,warm_start=
                                     True),
    }
```
'ET': ExtraTreesClassifier(n_estimators =10, max_features=None, bootstrap= False, random_state=123, warm_start= True),
'GB': GradientBoostingClassifier(
    n_estimators=10, max_features=None,
    random_state=123, warm_start=True)
}

# Timing how long it takes to run through all models

    tick = timer.perf_counter()

    # Getting the split for this cv
    x_train, x_test, y_train, y_test =
        train_test_split(train_group_data,
                          y_train_group, random_state=123+cv)

    # Getting the # of times the model needs to be iterated on

        if len(x_train) % samples != 0:
            model_fit_count = int(len(x_train)/samples) + 1
        else:
            model_fit_count = int(len(x_train)/samples)

        if len(x_test) % samples != 0:
            model_test_count = int(len(x_test)/samples) + 1
        else:
            model_test_count = int(len(x_test)/samples)

    # Iterating through the # of fits
for int_model in range(model_fit_count):
    # Getting the values for the model for testing
    if int_model != model_fit_count - 1:
        x, y = x_train[samples*int_model:
            samples*(int_model+1)].compute()
        y_train[samples*int_model:
            samples*(int_model+1)].compute()
    else:
        x, y = x_train[samples*int_model:].compute(), y_train[samples*
            int_model:].compute()
    # Creating the timing variable
    fit_time = [np.zeros(model_fit_count) for
        ii in partial_fit_classifiers]
    # Iterating through the different models
    for int_cls, (cls_name, cls) in enumerate(partial_fit_classifiers.items()):
        # timing each classifier
        tick1 = time.perf_counter()
        # Using the client to parallelize the fits
        with joblib.parallel_backend('dask'):
            # update estimator with data
            if cls_name == 'RF' or cls_name == 'ET' or cls_name == 'GB':
                cls.fit(x, y)
if int_model !=
  → model_fit_count − 1:
    cls.n_estimators += 1
  else:
    cls.partial_fit(x, y, classes
      ↦ = [0, 1])
    fit_time[int_cls][int_model] = time.
      ↦ perf_counter() − tick1
# Printing the time it took to fit all of the
  ↦ models
  print('Took %s minutes to fit all models' %
    (time.perf_counter() − tick) / 60))
# Creating the test variables
  acc = [np.zeros(model_test_count) for ii in
    → range(len(partial_fit_classifiers))]  
# Iterating through the # of tests
  for int_model in range(model_test_count):
    if int_model != model_test_count − 1:
      xt, yt = x_test[samples*int_model:
        ↦ samples*(int_model+1)].compute
        ↦ (), y_test[samples*int_model:
          ↦ samples*(int_model+1)].compute
          ↦ ()
    else:
      xt, yt = x_test[samples*int_model:].
        ↦ compute(), y_test[samples*
          → int_model:].compute()
# Iterating through the different models
  for int_cls, (cls_name, cls) in enumerate
    → (partial_fit_classifiers.items()):
  381
# Using the client to parallelize the fits

```python
classifier_w(client, use_dask=True, **kwargs)
```

# update estimator with data
```python
classifier_w.update(data=x, y=y)
```

# Creating labels for AUROC
```python
for i, (cls_name, cls) in enumerate(partial_fit_classifiers.items):
    # model fitting stats
    results_group.loc['%s %s %s Group' % (error, reads, cls_name)] = np.mean(fit_time[
    # accumulate test accuracy stats
    results_group.loc['%s %s %s Group' % (error, reads, cls_name)] = len(y_train)

# Calculating Confusion Matrix
```
if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive - Aggressive':
    conf_mat = confusion_matrix(y_test.compute(), np.where(np.concatenate(pred[cls_name]) > 0, 1, 0))
else:
    conf_mat = confusion_matrix(y_test.compute(), np.concatenate(pred[cls_name]).argmax(axis=1))
np.save(os.path.join(pred_arrays, '%s %s %s Plasmid Group Prediction Array' % (int(error * 100), reads, cls_name, cv)), np.concatenate(pred[cls_name]))
np.save(os.path.join(pred_arrays, '%s %s %s Plasmid Group Confusion Matrix' % (int(error * 100), reads, cls_name, cv)), conf_mat)
pred[cls_name] = []

# Printing the time the model testing has been running
print('Train set tested. Total time model testing has been running %s minutes' % ((time.perf_counter() - tick) / 60))

# Creating the test variables
acc_t = [np.zeros(model_unseen_count) for ii in partial_fit_classifiers]
test_time = [np.zeros(model_unseen_count) for ii in partial_fit_classifiers]

# Iterating through the # of tests
for int_model in range(model_unseen_count):
    if int_model != model_unseen_count - 1:
        xu, yu = unseen_group_data[samples*int_model:samples*(int_model+1)].compute(), y_unseen_group[samples*int_model:samples*(int_model+1)].compute()
    else:
        xu, yu = unseen_group_data[samples*int_model:].compute(), y_unseen_group[samples*int_model:].compute()

# Iterating through the different models
for int_cls, (cls_name, cls) in enumerate(partial_fit_classifiers.items()):
    # timing each classifier
    tick1 = time.perf_counter()
    # Using the client to parallelize the fits
    with joblib.parallel_backend('dask'):
        # update estimator with data
        acc_t[int_cls][int_model] = cls.
score(xu, yu)
        if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive-Aggressive':
pred[cls_name].append(cls.
    \rightarrow decision_function(xu))
else:
pred[cls_name].append(cls.
    \rightarrow predict_proba(xu))
test_time[int_cls][int_model] = time.
    \rightarrow perf_counter() - tick1

# Creating labels for AUROC
for int_cls, (cls_name, cls) in enumerate(
    \rightarrow partial_fit_classifiers.items()):
  # model fitting stats
  results_group.loc[cv, '%s %s %s Group Test
  \rightarrow Time' % (error, reads, cls_name)]
  \rightarrow = np.mean(test_time[int_cls])
  results_group.loc[cv, '%s %s %s Group Test
  \rightarrow Number' % (error, reads, cls_name)
  \rightarrow ] = len(y_unseen_group)

  # accumulate test accuracy stats
  results_group.loc[cv, '%s %s %s Group Test
  \rightarrow Accuracy' % (error, reads,
  \rightarrow cls_name)] = np.mean(acc_t[int_cls
  \rightarrow ])

# Calculating Confusion Matrix
if cls_name == 'SGD' or cls_name == 'Perceptron' or cls_name == 'Passive
  \rightarrow -Aggressive':
  conf_mat = confusion_matrix(
    \rightarrow y_unseen_group.compute(), np.
    \rightarrow where(np.concatenate(pred[
    \rightarrow cls_name]) >0,1,0))
else:

    conf_mat = confusion_matrix(
        y_unseen_group.compute(), np.
        concatenate(pred[cls_name]).
        argmax(axis=1))

np.save(os.path.join(pred_arrays, '%s %s
        Plasmid Group Prediction Array
        Test %s.npy' % (int(error*100),
        reads, cls_name, cv)), np.
        concatenate(pred[cls_name]))

np.save(os.path.join(pred_arrays, '%s %s
        Plasmid Group Confusion Matrix
        Test %s.npy' % (int(error*100),
        reads, cls_name, cv)), conf_mat)

    pred[cls_name] = []

print('Took %s minutes to fit, validate and
    test all models for 1 cross-fold
    validation' % ((time.perf_counter() -
    tick)/60))

print('%s fold Cross Validation finished' % (cv))

# Timing how long it takes to run through all
    discriminant analysis models
tick0 = time.perf_counter()

## Creating component analysis data on the first 500
    SERS sequences out of the 1000
    x_train, x_test, y_train, y_test =
        train_test_split(train_group_data,
        y_train_group, random_state=123)
x, y = x_train[:samples].compute(), y_train[:samples].compute()
x_t, y_t = x_test.compute(), y_test.compute()
x_u, y_u = unseen_group_data[:samples].compute(),
y_unseen_group[:samples].compute()
classifier = {
    'PCA_svd': PCA(random_state=123),
    'LDA': LinearDiscriminantAnalysis(solver='svd'),
    'QDA': QuadraticDiscriminantAnalysis()
}
for cls_name, cls in classifier.items():
    if cls_name == 'PCA_svd':
        pg_pca_train = np.array(cls.fit_transform(np.concatenate((x, x_t), axis=0)))
        pg_pca_unseen = np.array(cls.transform(x_u))
        pg_pca = np.concatenate((pg_pca_train, pg_pca_unseen), axis=0)
        pg_name = np.concatenate((y, y_t, y_u)).astype('int8')
        pg_var = np.array([.32, .32, .36])
        np.save(os.path.join(pred_arrays, 'Plasmid Group PCA.npy', '%s %s
            Plasmid Group PCA.npy' % (int(100*error), reads)), pg_pca)
        pca_plot(pg_pca[:, :2], pg_name, 'PG', int((100*error), reads))
    else:
        # fit model
        cls.fit(x, y)
# CV Testing

results_group.loc[0, '%s %s %s Group Train

\rightarrow Accuracy' % (error, reads, cls_name)] = np.mean(cls.score(xt, yt))
pred[cls_name].append(cls.predict_proba(xt))

# Calculating Confusion Matrix

conf_mat = confusion_matrix(yt, np.concatenate(pred[cls_name]).argmax(axis=1))

np.save(os.path.join(pred_arrays, '%s %s Plasmid Group Prediction Array Test %s.npy' % (int(error*100), reads, cls_name, cv)), np.concatenate(pred[cls_name]))

np.save(os.path.join(pred_arrays, '%s %s Plasmid Group Confusion Matrix Test %s.npy' % (int(error*100), reads, cls_name, cv)), conf_mat)
pred[cls_name] = []

# Unknown Testing

results_group.loc[0, '%s %s %s Group Test

\rightarrow Accuracy' % (error, reads, cls_name)] = np.mean(cls.score(xu, yu))
pred[cls_name].append(cls.predict_proba(xu))

# Calculating Confusion Matrix
```python
conf_mat = confusion_matrix(yu, np.
    → concatenate(pred[cls_name]).argmax(
    → axis=1))
np.save(os.path.join(pred_arrays, '%s %s
    → %s Plasmid Group Prediction Array
    → Test %s.npy % (int(error*100),
    → reads, cls_name, cv)), np.
    → concatenate(pred[cls_name]))
np.save(os.path.join(pred_arrays, '%s %s
    → %s Plasmid Group Confusion Matrix
    → Test %s.npy % (int(error*100),
    → reads, cls_name, cv)), conf_mat)
pred[cls_name] = []
print('Took %s minutes to fit, validate and test
    → all discriminant analysis models' % ((time.
    → perf_counter()-tick0)/60))
print('Took %s hours to run Individual and Group
    → models' % ((time.perf_counter()-tick)/3600))

# Saving the results
results_group.to_csv('Plasmid_ML_Group_Results_%
    → s_%s_%s.csv' % (reads, int(100*error),
    → int(data)))
print('Results saved')
print(datetime.datetime.now().isoformat())
print('%s error rate and %s number of reads tests
    → finished' % (error, reads))
```
print('Took %s hours to run all number of reads tests' % ((time.perf_counter() - start1)/3600))
print('Took %s days to run all error rate tests' % ((time.perf_counter() - start)/(3600*24)))

client.close()
close()
print(begin)
print(datetime.datetime.now().isoformat())

D.2.2 Confusion Matrix
ttfamily
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
from matplotlib import colors

'You need to change the path to the folder where you saved the NPY files.
You can also change the error rate and number of reads to the desired combination''
path = 'C:/Usr/NPY/'
error_rate = 10
num_of_reads = 10000

plt.rc('xtick', labelsize=22)
```python
plt.rc('ytick', labelsize=22)

def plot_conf_matrix(conf, classes, save_title, fmt='.3f', cmap=
                      plt.cm.Blues):

    norm = colors.Normalize(vmin=0, vmax=1)

    fig = plt.figure(figsize=(18,17))
    gs = grd.GridSpec(2,7, figure=fig, width_ratios
                      = [25,25,25,25,25,25,3])
    ax1 = fig.add_subplot(gs[0,:2])
    ax1.imshow(conf[0], norm=norm, interpolation='nearest', cmap=
                cmap)
    ax1.set(xticks=np.arange(conf[0].shape[1]),
            yticks=np.arange(conf[0].shape[0]),
            xticklabels=classes[:conf[0].shape[1]],
            yticklabels=classes)
    plt.setp(ax1.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")
    ax1.set_title('ET', fontdict={'fontsize':24})

    ax2 = fig.add_subplot(gs[0,2:4])
    ax2.imshow(conf[1], norm=norm, interpolation='nearest', cmap=
                cmap)
    ax2.set(xticks=np.arange(conf[0].shape[1]),
            xticklabels=classes[:conf[0].shape[1]])
    plt.setp(ax2.get_yticklabels(), visible=False)
    plt.setp(ax2.get_xticklabels(), rotation=45, ha="right",
             rotation_mode="anchor")

    # Rotate the tick labels and set their alignment.
```

391
```python
ax2.yaxis.set_tick_params(which='both', length=0)
ax2.set_title('NN', fontdict={'fontsize':24})

ax3 = fig.add_subplot(gs[0, 4:6])
ax3.imshow(conf[2], norm=norm, interpolation='nearest', cmap=cmap)
ax3.set(xticks=np.arange(conf[0].shape[1]),
        xticklabels=classes[:conf[0].shape[1]])
# Rotate the tick labels and set their alignment.
plt.setp(ax3.get_yticklabels(), visible=False)
plt.setp(ax3.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")
ax3.yaxis.set_tick_params(which='both', length=0)
ax3.set_title('LDA', fontdict={'fontsize':24})

ax4 = fig.add_subplot(gs[1, 1:3])
ax4.imshow(conf[3], norm=norm, interpolation='nearest', cmap=cmap)
ax4.set(xticks=np.arange(conf[0].shape[1]),
        yticks=np.arange(conf[0].shape[0]),
        xticklabels=classes[:conf[0].shape[1]],
        yticklabels=classes)
# Rotate the tick labels and set their alignment.
plt.setp(ax4.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")
ax4.set_title('PA', fontdict={'fontsize':24})

ax5 = fig.add_subplot(gs[1, 3:5])
im5 = ax5.imshow(conf[4], norm=norm, interpolation='nearest',
                 cmap=cmap)
```
ax5.set(xticks=np.arange(conf[0].shape[1]),
        xticklabels=classes[:, conf[0].shape[1]])

# Rotate the tick labels and set their alignment.
plt.setp(ax5.get_xticklabels(), visible=False)
plt.setp(ax5.get_yticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")
ax5.yaxis.set_tick_params(which='both', length=0)
ax5.set_title('GNB', fontdict={'fontsize':24})

ax6 = fig.add_subplot(gs[:,6])
ax6.figure.colorbar(im5, cax=ax6)

# We want to show all ticks...
for axe in [ax1, ax2, ax3, ax4, ax5]:
    axe.set(xticks=np.arange(conf[0].shape[1]),
            yticks=np.arange(conf[0].shape[0]),
            xticklabels=classes[:, conf[0].shape[1]],
            yticklabels=classes)

# Rotate the tick labels and set their alignment.
plt.setp(axe.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(conf[0].shape[0]):
    for j in range(conf[0].shape[1]):
        for int_ax, axe in enumerate([ax1, ax2, ax3, ax4, ax5]):
            if conf[int_ax][i, j] <= 0.005:
                axe.text(j, i, 0,
                          ha="center", va="center"
color="white" if conf[int_ax][i, j] > 
\[
\rightarrow 0.45 \text{ else } "black", \text{ fontsize} \n\rightarrow =20\)

elif conf[int_ax][i, j] >= 0.995:
    axe.text(j, i, 1,
    ha="center", va="center",
    color="white" if conf[int_ax][i, j] > 
\[
\rightarrow 0.45 \text{ else } "black", \text{ fontsize} \n\rightarrow =20\)

else:
    axe.text(j, i, format(conf[int_ax][i, j], fmt 
\[
\rightarrow ),
    ha="center", va="center",
    color="white" if conf[int_ax][i, j] > 
\[
\rightarrow 0.45 \text{ else } "black", \text{ fontsize} \n\rightarrow =20\)

fig.text(0.02, .5, 'True Label', fontsize=30, rotation='vertical' 
\[
\rightarrow )

fig.text(.4, 0.02, 'Predicted Label', fontsize=30)

plt.tight_layout()

plt.savefig('%s.png' % (save_title), dpi='figure', 
\[
\rightarrow bbox_inches='tight')

plt.savefig(os.path.join('J:/groups/dnafingers/Fingerprints 
\[
\rightarrow Analysis/Graphics/eps/', '%s.eps' % (save_title)), dpi=' 
\[
\rightarrow figure', bbox_inches='tight')

plt.close()

return
```python
import numpy as np

# Initialize matrices

G = np.zeros((12, 10))  # GNN = np.zeros((12, 10))
L = np.zeros((12, 10))  # PA = np.zeros((12, 10))
D = np.zeros((12, 10))  # GNB = np.zeros((12, 10))

# Initialize smaller matrices

G = np.zeros((5, 5))  # ET = np.zeros((5, 5))
L = np.zeros((5, 5))  # PA = np.zeros((5, 5))
D = np.zeros((5, 5))  # GNB = np.zeros((5, 5))

for ii in range(10):
    g_ET[:, :10, :10] += np.load('%s%s %s ET Genome Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
    g_NN[:, :10, :10] += np.load('%s%s %s Neural Network Genome Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
    g_PA[:, :10, :10] += np.load('%s%s %s Passive Aggressive Genome Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
    g_GNB[:, :10, :10] += np.load('%s%s %s GNB Genome Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
    g_LDA[:, :10, :10] += np.load('%s%s %s LDA Genome Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
```

395
p_ET += np.load('%s%s %s ET Plasmid Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
p_NN += np.load('%s%s %s Neural Network Plasmid Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
p_PA += np.load('%s%s %s Passive-Aggressive Plasmid Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
p_GNB += np.load('%s%s %s GNB Plasmid Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
pg_ET += np.load('%s%s %s ET Plasmid Group Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
pg_NN += np.load('%s%s %s Neural Network Plasmid Group Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
pg_PA += np.load('%s%s %s Passive-Aggressive Plasmid Group Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
pg_GNB += np.load('%s%s %s GNB Plasmid Group Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
pg_LDA += np.load('%s%s %s LDA Plasmid Group Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
p_LDA += np.load('%s%s %s LDA Plasmid Confusion Matrix Test 9.npy' % (path, error_rate, num_of_reads))
ge_ET = np.zeros((10,10))
ge_NN = np.zeros((10,10))
ge_LDA = np.zeros((10,10))
ge_PA = np.zeros((10,10))
ge_GNB = np.zeros((10,10))
for ii in range(10):
ge_ET += np.load('%s%s %s ET Genome Extras Confusion Matrix' % (path, error_rate, num_of_reads, ii))
g_NN += np.load('%s%s %s Neural Network Genome Extras Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
g_PA += np.load('%s%s %s Passive-Aggressive Genome Extras Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
g_GNB += np.load('%s%s %s GNB Genome Extras Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))
g_LDA += np.load('%s%s %s LDA Genome Extras Confusion Matrix Test %s.npy' % (path, error_rate, num_of_reads, ii))

g_ET[11,:] += [0., 0., 0., 10000, 0., 0., 0., 0., 0., 0.]
g_NN[11,:] += [0., 0., 0., 10000, 0., 0., 0., 0., 0., 0.]
g_PA[11,:] += [0., 0., 0., 10000, 0., 0., 0., 0., 0., 0.]
g_GNB[11,:] += [0., 0., 0., 10000, 0., 0., 0., 0., 0., 0.]
g_LDA[11,:] += [0., 0., 0., 10000, 0., 0., 0., 0., 0., 0.]

g_ET[10,:] += ge_ET.sum(axis=0)-g_ET[11,:]
g_NN[10,:] += ge_NN.sum(axis=0)-g_NN[11,:]
g_PA[10,:] += ge_PA.sum(axis=0)-g_PA[11,:]
g_GNB[10,:] += ge_GNB.sum(axis=0)-g_GNB[11,:]
g_LDA[10,:] += ge_LDA.sum(axis=0)-g_LDA[11,:]

g_ET = g_ET.astype('float')/g_ET.sum(axis=1)[..., np.newaxis]
g_NN = g_NN.astype('float')/g_NN.sum(axis=1)[..., np.newaxis]
g_LDA = g_LDA.astype('float')/g_LDA.sum(axis=1)[..., np.newaxis]
g_PA = g_PA.astype('float')/g_PA.sum(axis=1)[..., np.newaxis]
g_GNB = g_GNB.astype('float')/g_GNB.sum(axis=1)[..., np.newaxis]
p_ET = p_ET.astype('float')/p_ET.sum(axis=1)[..., np.newaxis]
p_NN = p_NN.astype('float')/p_NN.sum(axis=1)[..., np.newaxis]
p_LDA = p_LDA.astype('float')/p_LDA.sum(axis=1)[..., np.newaxis]
p_PA = p_PA.astype('float')/p_PA.sum(axis=1)[..., np.newaxis]
p_GNB = p_GNB.astype('float')/p_GNB.sum(axis=1)[..., np.newaxis]
pg_ET = pg_ET.astype('float')/pg_ET.sum(axis=1)[..., np.newaxis]
pg_NN = pg_NN.astype('float')/pg_NN.sum(axis=1)[..., np.newaxis]
pg_LDA = pg_LDA.astype('float')/pg_LDA.sum(axis=1)[..., np.newaxis]
pg_PA = pg_PA.astype('float')/pg_PA.sum(axis=1)[..., np.newaxis]
pg_GNB = pg_GNB.astype('float')/pg_GNB.sum(axis=1)[..., np.newaxis]

g_class = ['B. fragilis', 'C. jejuni', 'E. coli', 'K. pneumoniae', 'S. enterica', 'S. aureus', 'S. pneumoniae', 'S. pyogenes', 'E. hirae', 'E. fergusonii', 'K. aerogenes', 'M. tuberculosis']
p_class = ['IMP', 'KPC', 'NDM', 'No Resistance', 'VIM']
pg_class = ['No Resistance', 'Resistance']

config = [g_ET, g_NN, g_LDA, g_PA, g_GNB]
confp = [p_ET, p_NN, p_LDA, p_PA, p_GNB]
confpg = [pg_ET, pg_NN, pg_LDA, pg_PA, pg_GNB]

for it, cl, titl in [(config, g_class, 'Genome Confusion Matrix 10000'), (confp, p_class, 'Plasmid Confusion Matrix 100000'), (confpg, pg_class, 'Plasmid Group Confusion Matrix 100000')]:
    plot_conf_matrix(it, cl, titl)

# [['true negative', 'false positive'],
#     ['false negative', 'true positive']]
# So anything that is in the same row as a true label
# but not correct is a false negative.
# Which means things in the same column as a predicted label
# but not correct are a false positive.

## D.2.3 G Plotting

ttfamily
import os
import json
import time
import datetime
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as grd
from dask.distributed import Client, LocalCluster
import dask.dataframe as ddf
import dask.array as da

' You need to change the path to the folder where you saved the
↩ SERS files. ' 

local_SERS = 'C:/Usr/SERS/'

plt.rcParams['font.weight'] = 'bold'
plt.rcParams['font.size'] = 14
plt.rcParams['axes.labelweight'] = 'bold'
def plotting_data(color_array, data_array, names, data_range,
                   label_int, title='no title', save='no', normalized_type=''
                   Random'):  

399
This determines how many points are being plotted and then creates the necessary gradients for creating the A,T,C,G colorbars that go with the plot. ""

```python
index_values = list(range(1, len(color_array[0]) + 1))
s = [60 for n in range(len(color_array[0]))]  # the size of the markers in plots
gradient1 = np.vstack((color_array[0], color_array[0]))
gradient2 = np.vstack((color_array[1], color_array[1]))
gradient3 = np.vstack((color_array[2], color_array[2]))
gradient4 = np.vstack((color_array[3], color_array[3]))

fig = plt.figure(figsize=(20, 10))

""" Creates a set of 5 graphs inside the figure for the actual plot and then the 4 colorbar plots for representing the ATCG percentages """

# gs = grd.GridSpec(5, 1, height_ratios=[15, 1, 1, 1, 1], hspace=0.05)
gs = grd.GridSpec(6, 1, height_ratios=[15, 0.5, 1, 1, 1, 1], hspace=0.05)

""" Working on formatting the actual plot. Getting rid of the x-axis, setting the necessary boundaries for the plot, adding a legend and a title (if applicable) """
ax = plt.subplot(gs[0], xticklabels=data_range, xticks=
label_int)
```

400
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)

# ax.xaxis.set_ticks([])
ax.set_xlim(index_values[0]−1, index_values[−1]+1)
plt.setp(ax.spines.values(), linewidth=2)
ax.xaxis.set_tick_params(width=2)
ax.yaxis.set_tick_params(width=2)

if normalized_type == 'no':
    ax.set_ylabel('10-mer Frequency', fontdict={'size':20})
else:
    # ax.set_ylim(0,1.01)
    ax.set_ylabel('10-mer Frequency Deviation from Bias',
               fontdict={'size':20})

# looping through each data set for plotting
max_data = 0
min_data = 0
mark = ['o', 'o', 's', 'v', '*', 'x']
color = ['k', 'tab:orange', 'tab:purple', 'tab: cyan', 'tab:gray',
         'tab:brown', 'tab:pink']
for row in range(1, len(data_array)):
    if max(data_array[row]) > max_data:
        max_data = max(data_array[row]) + 0.05*max(data_array
               [row])
    if min(data_array[row]) < min_data:
        if min(data_array[row]) > 0:
            min_data = min(data_array[row]) − 0.05*min(data_array[row])
        else:

401
min_data = \text{min}(\text{data_array}[\text{row}]) + 0.05 \times \text{min}(\text{data_array}[\text{row}])

ax.set_ylim(min_data, max_data)

ax.scatter(index_values, data_array[\text{row}], s=s, c=color[\text{row}], marker=mark[\text{row}], label=names[\text{row}])

ax.plot(index_values, data_array[0], 'k-', label=names[0], linewidth=2)

if not title == 'no title':
    plt.title(title)
    plt.legend(prop={'size':16})

""" Creating the colorbars for visualizing representing the ATCG percenatages"""

colorAx1 = plt.subplot(gs[2])
colorAx1.imshow(gradient1, aspect='auto', cmap=plt.get_cmap('Blues'))
pos = list(colorAx1.get_position().bounds)
x_text = pos[0] - 0.01
y_text = pos[1] + pos[3]/2
fig.text(x_text, y_text, 'C', va='center', ha='center')
colorAx1.set_axis_off()

colorAx2 = plt.subplot(gs[3])
colorAx2.imshow(gradient2, aspect='auto', cmap=plt.get_cmap('Greens'))
pos = list(colorAx2.get_position().bounds)
x_text = pos[0] - 0.01
y_text = pos[1] + pos[3]/2
label_text = ['T', '0', '1', '0', '2', '1', '0', '3', '2', '1']
for ii in range(len(label_text)):
    fig.text(x_text[ii], y_text[ii], label_text[ii], va='center', ha='center')
colorAx3.set_axis_off()

colorAx4 = plt.subplot(gs[5])
colorAx4.imshow(gradient4, aspect='auto', cmap=plt.get_cmap('Greys'))
pos = list(colorAx4.get_position().bounds)
x_text = [pos[0] - 0.01, pos[0] + pos[2]*5.5/286, pos[0] +
            pos[2]*16/286, pos[0] + pos[2]*26/286, pos[0] + pos
            [2]*36/286, pos[0] + pos[2]*44.75/286, pos[0] + pos
            [2]*53.5/286, pos[0] + pos[2]*62/286, pos[0] + pos
            [2]*70/286, pos[0] + pos[2]*78.25/286, pos[0] + pos
            [2]*86/286, pos[0] + pos[2]*93.5/286, pos[0] + pos
            [2]*100.5/286, pos[0] + pos[2]*107.75/286, pos[0] + pos
            404

label_text = [ 'A', '0', '0', '1', '0', '1', '2', '0', '1', '2
    ', '3', '0', '1', '2', '3', '4', '0', '1', '2', '3', '4
    ', '5', '0', '1', '2', '3', '4', '5', '6', '0', '1', '2
    ', '3', '4', '5', '6', '7']
for ii in range(len(label_text)):
    fig.text(x_text[ii], y_text[ii], label_text[ii], va='center', ha='center')
colorAx4.set_axis_off()
x_text1 = pos[0] + pos[2]/2
y_text1 = pos[1] - 0.03
fig.text(x_text1, y_text1, 'Fractional Base Composition Spectrum', va='center', ha='center', fontdict={'size': 20})

""" Either showing the plot or just directly saving it """
if save == 'yes':
    if not normalized_type == 'no':
        plt.savefig(normalized_type + ' ' + title + '.png', dpi='figure', bbox_inches='tight')
        plt.close()
    else:
        plt.savefig('Not Normalized ' + title + '.png', dpi='figure', bbox_inches='tight')
        plt.close()
else:
    plt.show()

return

start = time.perf_counter()

# creating the categorical labels for storing data
dna_length = 10
# Creating the correct tuples for how many A, T, G and C's are in each bin

kmer_range = [(aa, tt, gg, cc) for aa in range(dna_length + 1)
               for tt in range(dna_length + 1) for gg in range(dna_length + 1)
               for cc in range(dna_length + 1) if aa + tt + cc + gg == dna_length]

# Setting dna_length based bias
bias = np.array([(1/4**dna_length) * math.factorial(dna_length)/
                 math.factorial(kmer[0]) * math.factorial(kmer[1]) * math.
                 factorial(kmer[2]) * math.factorial(kmer[3])) for kmer in
                 kmer_range])

# Creating the categorical labels for storing data

data_categories = ["A%sT%sG%sC%s" % (str(aa), str(tt), str(gg),
                      str(cc)) for aa in range(dna_length + 1) for tt in range(dna_length + 1)
                      for gg in range(dna_length + 1) for cc in
                      range(dna_length + 1) if aa + tt + cc + gg == dna_length]

# Creating the labels for the non-numeric information

data_index = ["Seq Record ID", "Resistance", "Name", "Genus", 
               "DNA Type", "Strain", "Bacteria Type", "Notes"]

# Combining the the non-numeric and categorical labels for the panda's dataframe

data_index.extend(data_categories)

# Number of samples per species
num_training_samples = 1000

# List of error testing rates
error_rate = [0]

# List of number of optical sequencing reads
num_reads = [1000000]
# Getting the list of all the data files that need to be tested
file_list = [file for file in os.listdir(local_SERS)]

# Initializing dask to run things in parallel
with LocalCluster(processes=False) as cluster, Client(cluster) as client:

# Cycling through the 24 combinations of error rate and number of reads
for error in error_rate:
    # Recording the time it takes to run everything
    for reads in num_reads:
        # Getting the error and reads values in string form for file identification
        str_err = '%s' % (int(100*error))
        str_read = '%s' % (reads)

        # Getting the list of the files for the specific reads and errors
        working_genome_list = [file for file in file_list if str_err in file and str_read in file and 'Genome' in file]

        # Making sure that the lists aren't empty
        run_list = []
        if len(working_genome_list) != 0:
            run_list.append(working_genome_list)

        for use_list in run_list:
            # Retrieving whether the file is genomic or plasmid
            file_type = 'Genome'
# Determining how long this takes

tick = time.perf_counter()

# Splitting the files into the unseen sets and the
# training sets
train_set = []
unseen_test_set = []
check_set = []
file_info = []

# Getting each file in the list of files to use
for file in use_list:
    if file.split('_')[3] == 'Training':
        train_file = file
    elif file.split('_')[3] == 'Testing':
        test_file = file
    elif file.split('_')[3] == 'Extras':
        extra_file = file

# Opening up each of the HDF5 files in the working
# list
store_train = ddf.read_hdf(os.path.join(
    local_SERS, train_file), 'df*')
store_test = ddf.read_hdf(os.path.join(local_SERS
    , test_file), 'df*')
store_extra = ddf.read_hdf(os.path.join(
    local_SERS, extra_file), 'df*')

# Storing just the values of the data split
train_set.append(store_train.drop('Name', axis='columns').to_dask_array(True))
unseen_test_set.append(store_test.drop('Name',
    axis='columns').to_dask_array(True))
check_set = check_set.append(store_extra.drop('Name', axis='columns').to_dask_array(True))

bac_train = np.char.array(store_train.loc[:, 'Name'].compute().to_list())

bac_test = np.char.array(store_test.loc[:, 'Name'].compute().to_list())

bac_ex = np.char.array(store_extra.loc[:, 'Name'].compute().to_list())

bac_train = np.where(bac_train.rfind('_') != bac_train.find('_'), bac_train.rpartition('_')[0:].replace('_', ''), bac_train.replace('_', ' '))

bac_test = np.where(bac_test.rfind('_') != bac_test.find('_'), bac_test.rpartition('_')[0:].replace('_', ''), bac_test.replace('_', ' '))

bac_ex = np.where(bac_ex.rfind('_') != bac_ex.find('_'), bac_ex.rpartition('_')[0:].replace('_', ''), bac_ex.replace('_', ' '))

bac_ex = np.where(bac_ex == 'Enterobacter aerogenes', 'Klebsiella aerogenes', bac_ex)

# Creating a testing data array for the unseen tests and the training data

train_data = da.concatenate([arr[int(jj*num_training_samples):int(1+jj*num_training_samples)] for arr in train_set for jj in range(len(arr.chunks[0]))], axis=0).compute()

unseen_data = da.concatenate([arr[int(jj*num_training_samples):int(1+jj*num_training_samples)] for arr in unseen_set for jj in range(len(arr.chunks[0]))], axis=0).compute()
num_training_samples) for arr in unseen_test_set for jj in range(len(arr.chunks[0])), axis=0).compute()

check_data = da.concatenate([arr[int(jj* num_training_samples):int(1+jj* num_training_samples)] for arr in check_set for jj in range(len(arr.chunks[0])), axis=0).compute()

# Creating the labels/categories for the data

train_names = da.concatenate([np.array(bac_train[int(jj* num_training_samples):int(1+jj* num_training_samples)]) for arr in train_set for jj in range(len(arr.chunks[0])), axis=0).compute()

unseen_names = da.concatenate([np.array(bac_test[int(jj* num_training_samples):int(1+jj* num_training_samples)]) for arr in unseen_test_set for jj in range(len(arr.chunks[0])), axis=0).compute()

check_names = da.concatenate([np.array(bac_ex[int(jj* num_training_samples):int(1+jj* num_training_samples)]) for arr in check_set for jj in range(len(arr.chunks[0])), axis=0).compute()

data = np.concatenate((train_data, unseen_data, check_data, np.zeros((1, 286))))

df = pd.DataFrame(data=data, columns=data_categories)
pdf = df.iloc[[22, 3, 17, 20], :]

data_range = ["A0T0G0C10", "A0T0G1C9", "A0T0G2C8"
            , "A0T0G3C7", "A0T0G4C6", "A0T0G5C5", "A0T0G6C4", "A0T0G7C3", "A0T0G8C2", "A0T0G9C1", "A0T1G1C8", "A0T1G2C7", "A0T1G3C6", "A0T1G4C5", "A0T1G5C4", "A0T1G6C3", "A0T1G7C2", "A0T1G8C1", "A0T1G9C0", "A1T0G0C9", "A1T0G1C8", "A1T0G2C7", "A1T0G3C6", "A1T0G4C5", "A1T0G5C4", "A1T0G6C3", "A1T0G7C2", "A1T0G8C1", "A1T0G9C0", "A0T2G0C8", "A0T2G1C7", "A0T2G2C6", "A0T2G3C5", "A0T2G4C4", "A0T2G5C3", "A0T2G6C2", "A0T2G7C1", "A0T2G8C0", "A1T1G0C8", "A1T1G1C7", "A1T1G2C6", "A1T1G3C5", "A1T1G4C4", "A1T1G5C3", "A1T1G6C2", "A1T1G7C1", "A1T1G8C0", "A2T0G0C8", "A2T0G1C7", "A2T0G2C6", "A2T0G3C5", "A2T0G4C4", "A2T0G5C3", "A2T0G6C2", "A2T0G7C1", "A2T0G8C0", "A0T3G0C7", "A0T3G1C6", "A0T3G2C5", "A0T3G3C4", "A0T3G4C3", "A0T3G5C2", "A0T3G6C1", "A0T3G7C0", "A1T2G0C7", "A1T2G1C6", "A1T2G2C5", "A1T2G3C4", "A1T2G4C3", "A1T2G5C2", "]

data_range += bias

p2df = df.iloc[[22, 3, 17, 20], :]
p2df += bias

pnames = ["Bias", train_names[3], unseen_names[7],
            check_names[0]]
A0T6G1C3 , A0T6G2C2 , A0T6G3C1 , A0T6G4C0 , A1T5G0C4 , A1T5G1C3 , A1T5G2C2 , A1T5G3C1 , A1T5G4C0 , A2T4G0C4 , A2T4G1C3 , A2T4G2C2 , A2T4G3C1 , A2T4G4C0 , A3T3G1C3 , A3T3G2C2 , A3T3G3C1 , A3T3G4C0 , A4T2G0C4 , A4T2G1C3 , A4T2G2C2 , A4T2G3C1 , A4T2G4C0 , A5T1G0C4 , A5T1G1C3 , A5T1G2C2 , A5T1G3C1 , A5T1G4C0 , A6T0G1C3 , A6T0G2C2 , A6T0G3C1 , A6T0G4C0 , A0T7G0C3 , A0T7G1C2 , A0T7G2C1 , A0T7G3C0 , A1T6G1C2 , A1T6G2C1 , A1T6G3C0 , A1T6G4C0 , A2T5G0C3 , A2T5G1C2 , A2T5G2C1 , A2T5G3C0 , A3T4G0C3 , A3T4G1C2 , A3T4G2C1 , A3T4G3C0 , A4T3G1C2 , A4T3G2C1 , A4T3G3C0 , A5T2G0C3 , A5T2G1C2 , A5T2G2C1 , A5T2G3C0 , A6T1G0C3 , A6T1G1C2 , A6T1G2C1 , A6T1G3C0 , A7T0G0C3 , A7T0G1C2 , A7T0G2C1 , A7T0G3C0 , A0T8G0C2 , A0T8G1C1 , A0T8G2C0 , A1T7G0C2 , A1T7G1C1 , A1T7G2C0 , A2T6G0C2 , A2T6G1C1 , A2T6G2C0 , A3T5G0C2 , A3T5G1C1 , A3T5G2C0 , A4T4G0C2 , A4T4G1C1 , A4T4G2C0 , A5T3G0C2 , A5T3G1C1 , A5T3G2C0 , A6T2G0C2 , A6T2G1C1 , A6T2G2C0 , A7T1G0C2 , A7T1G1C1 , A7T1G2C0
\[
\text{color_array} = [[1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0], 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0]]
\]
\[ \begin{array}{cccccccccccccccc}
0.3, & 0.2, & 0.1, & 0, & 0.5, & 0.4, & 0.3, & 0.2, & 0.1, \\
0, & 0.4, & 0.3, & 0.2, & 0.1, & 0, & 0.4, & 0.3, & 0.2, \\
0.1, & 0, & 0.4, & 0.3, & 0.2, & 0.1, & 0, & 0.4, & 0.3, \\
0.2, & 0.1, & 0, & 0.4, & 0.3, & 0.2, & 0.1, & 0, & 0.4, \\
0.3, & 0.2, & 0.1, & 0, & 0.4, & 0.3, & 0.2, & 0.1, & 0, \\
0.3, & 0.2, & 0.1, & 0, & 0.3, & 0.2, & 0.1, & 0, & 0.3, \\
0.2, & 0.1, & 0, & 0.3, & 0.2, & 0.1, & 0, & 0.3, & 0.2, \\
0.1, & 0, & 0.3, & 0.2, & 0.1, & 0, & 0.3, & 0.2, & 0.1, \\
0.3, & 0.2, & 0.1, & 0, & 0.2, & 0.1, & 0, & 0.2, & 0.1, \\
0, & 0.2, & 0.1, & 0, & 0.2, & 0.1, & 0, & 0.2, & 0.1, \\
0.2, & 0.1, & 0, & 0.2, & 0.1, & 0, & 0.2, & 0.1, & 0, \\
0.1, & 0, & 0.1, & 0, & 0.1, & 0, & 0.1, & 0, & 0.1, \\
0.1, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, \\
[0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, & 0.7, & 0.8, \\
0.9, & 1, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, \\
0.7, & 0.8, & 0.9, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, \\
0.6, & 0.7, & 0.8, & 0.9, & 0, & 0.1, & 0.2, & 0.3, & 0.4, \\
0.5, & 0.6, & 0.7, & 0.8, & 0, & 0.1, & 0.2, & 0.3, & 0.4, \\
0.5, & 0.6, & 0.7, & 0.8, & 0, & 0.1, & 0.2, & 0.3, & 0.4, \\
0.5, & 0.6, & 0.7, & 0.8, & 0, & 0.1, & 0.2, & 0.3, & 0.4, \\
0.6, & 0.7, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, \\
0.7, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, & 0.7, \\
0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, & 0, & 0.1, \\
0.2, & 0.3, & 0.4, & 0.5, & 0.6, & 0, & 0.1, & 0.2, & 0.3, \\
0.4, & 0.5, & 0.6, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, \\
0.6, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0.6, & 0, \\
0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0, & 0.1, & 0.2, & 0.3, \\
0.4, & 0.5, & 0, & 0.1, & 0.2, & 0.3, & 0.4, & 0.5, & 0, \\
416
\end{array} \]
0.1, 0.2, 0.3, 0.4, 0.5, 0, 0.1, 0.2, 0.3,
0.4, 0.5, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0,
0.1, 0.2, 0.3, 0.4, 0, 0.1, 0.2, 0.3, 0.4,
0, 0.1, 0.2, 0.3, 0.4, 0, 0.1, 0.2, 0.3,
0.4, 0, 0.1, 0.2, 0.3, 0.4, 0, 0.1, 0.2,
0.3, 0.4, 0, 0.1, 0.2, 0.3, 0.4, 0, 0.1,
0.2, 0.3, 0, 0.1, 0.2, 0.3, 0, 0.1, 0.2,
0.3, 0, 0.1, 0.2, 0.3, 0, 0.1, 0.2, 0.3, 0,
0.1, 0.2, 0.3, 0, 0.1, 0.2, 0.3, 0, 0.1,
0.2, 0.3, 0, 0.1, 0.2, 0, 0.1, 0.2, 0, 0.1,
0.2, 0, 0.1, 0.2, 0, 0.1, 0.2, 0, 0.1, 0.2,
0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0,
0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0.1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3,
0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
0.4, 0.4, 0.4, 0.4, 0.3, 0.3, 0.3, 0.3, 0.3,
0.3, 0.3, 0.3, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
0.2, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0, 0, 0, 0, 0, 0, 0, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
0.4, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.2,
0.2, 0.2, 0.2, 0.2, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0.6, 0.6,
0.6, 0.6, 0.6, 0.5, 0.5, 0.5, 0.5,
0.4, 0.4, 0.4, 0.4, 0.3, 0.3, 0.3,
0.3, 0.3, 0.2, 0.2, 0.2, 0.2, 0.1,
0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0.7,
0.7, 0.7, 0.7, 0.6, 0.6, 0.6, 0.5,
0.5, 0.5, 0.5, 0.4, 0.4, 0.4, 0.4,
0.3, 0.3, 0.3, 0.2, 0.2, 0.2, 0.2,
0.1, 0.1, 0.1, 0, 0, 0, 0.8, 0.8, 0.8,
0.7, 0.7, 0.7, 0.6, 0.6, 0.6, 0.5,
0.5, 0.4, 0.4, 0.4, 0.3, 0.3, 0.3,
0.2, 0.1, 0.1, 0.1, 0, 0, 0.9, 0.9,
0.8, 0.8, 0.7, 0.7, 0.6, 0.6, 0.5,
0.4, 0.4, 0.3, 0.3, 0.2, 0.2, 0.1,
0.1, 0.1, 0, 0.9, 0.8, 0.7, 0.6,
0.5, 0.4, 0.3, 0.2, 0.2, 0.2, 0.2,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0, 0, 0, 0, 0, 0, 0, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2,
0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2,
0.1, 0.1, 0.1, 0.1, 0.2, 0.2, 0.2,
0.2, 0.2, 0.2, 0.2, 0.3, 0.3, 0.3,
0.3, 0.3, 0.3, 0.3, 0, 0, 0, 0, 0,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.2, 0.2, 0.2, 0.2, 0.2, 0.3, 0.3,
0.3, 0.3, 0.3, 0.3, 0.3, 0.4, 0.4,
0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4,
0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4.

418
bplot = pdf[data_range].to_numpy()
nplot = p2df[data_range].to_numpy()

plotting_data(color_array, bplot, pnames, [
    data_range[0], data_range[44], data_range[77],
    data_range[107], data_range[146],
    data_range[178], data_range[210], data_range[241],
    data_range[285]],
    [1, 45, 78, 108, 147, 179, 211, 242, 286], save='no',
    normalized_type='Split')
plotting_data(color_array, nplot, pnames, [data_range[0], data_range[44], data_range[77], data_range[107], data_range[146], data_range[178], data_range[210], data_range[241], data_range[285]], [1, 45, 78, 108, 147, 179, 211, 242, 286], save='no', normalized_type='no')

print('Took %s minutes' % ((time.perf_counter() - start) / 60))

client.close()
cluster.close()

print(datetime.datetime.now().isoformat())