OPEN WATER

An Open Source Package to Create Input Files for Simulations using the Stream Network Temperature (SNTEMP) Model

Software Introduction

Joseph A. Daraio[1*]

[1]*Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's NL A1B 3X5*
[*]*Corresponding Author: jadaraio@mun.ca*

ABSTRACT

The DOS based Stream Network Temperature (SNTEMP) model is a comprehensive physics-based model that requires seven formatted input files, and it can be challenging and time consuming to create these files. The open source package SNTempInput was developed to create the required SNTEMP input files from a defined stream network, meteorological, and hydrological data required to drive the SNTEMP simulations. The software is intended to reduce the amount of time required to set up and run SNTEMP simulations to facilitate stream temperature related research and practice. This paper describes how to format the required data files and use SNTempInput to create properly formatted input files for SNTEMP. The paper provides guidance as to how and where to make any necessary modifications to the source code so that SNTEMP can be coupled with output from any hydrologic modeling software.

Keywords
SNTEMP, stream temperature, modeling

1. Introduction

Throughout much of the world stream temperature has increased as a result of climate change and other human impacts [*Caissie,* 2006; *Webb and Nobilis,* 2007; *Kaushal et al.*, 2010; *Isaak et al.*, 2012; *Ficklin et al.*, 2013; *Hester and Bauman,* 2013], and this trend will continue into the future along with the potential for significant impacts on water quality and aquatic habitat [*Poole and Berman,* 2001; *Mantua et al.*, 2010; *Null et al.*, 2013]. Therefore there is a growing need for stream temperature modeling efforts to aid in continued research, applications, and impact analyses of stream temperature response to anthropogenic- and climate-driven changes.

There are many approaches to stream temperature modelling and many different models available [see *Daraio and Bales* 2014 and *Daraio et al.* 2014]. One of the most comprehensive physics-based stream temperature models available is the Stream Network Temperature (SNTEMP) model [*Theurer et al.*, 1984], and it has been used widely for a variety of applications. Other stream network temperature models, such as WET-Temp [*Cox and Bolte,* 2007], have been developed using approaches similar that used in SNTEMP. The SNTEMP model is a DOS based program that is written in FORTRAN and requires 9 separate input files, each of which requires strict formatting in order to be read by the program. Some researchers have opted for use of the Stream Segment Temperature (SSTEMP) model [*Bartholow,* 2002] because it is easier to work with and can be run using a spreadsheet [*Gaffield et al.,* 2007]. However SSTEMP is difficult to apply to larger stream networks at the watershed scale. The author is aware of several researchers and practitioners that had difficulties running SNTEMP which lead to the use of other models that are easier to run but not as accurate as SNTEMP. These difficulties have led to the development of a basin scale stream temperature model with a reduced the number of required parameters [*Cheng and Wiley,* 2016].

Daraio and Bales [2014] and *Daraio et al.* [2014] did climate impact assessment using the hydrologic model PRMS and linked it with SNTEMP. Concurrently, Markstrom [2012] developed P2S that allows users to internally link PRMS with SNTEMP, which is now available for download from the USGS [*USGS,* 2013]. The source code for P2S is not available for use and modification for researchers and practitioners. The author developed general source code, written in the Java® programming language, to couple output from PRMS to SNTEMP, which is described in this paper. The source code can be altered so that the user can couple output from any hydrologic model with SNTEMP. The purpose of this paper is (1) to describe how to use *SNTempInput* to create properly formatted input files for SNTEMP (2) to show how and where to make any necessary modifications to the source code for use with other models. This is not intended as a replacement for using the manual and other resources available on how to set up and run SNTEMP [*Theurer et al.,* 1984; *Bartholow,* 2002; *Markstrom,* 2012].

2.0 Input File Creation

There are 7 required and 2 optional input files for SNTEMP, and *SNTempInput* creates 8 of these files, including the 7 required files (Figure 1). Each file has a rigid formatting requirement and the primary objective of the *SNTempInput* source code is to create properly for-

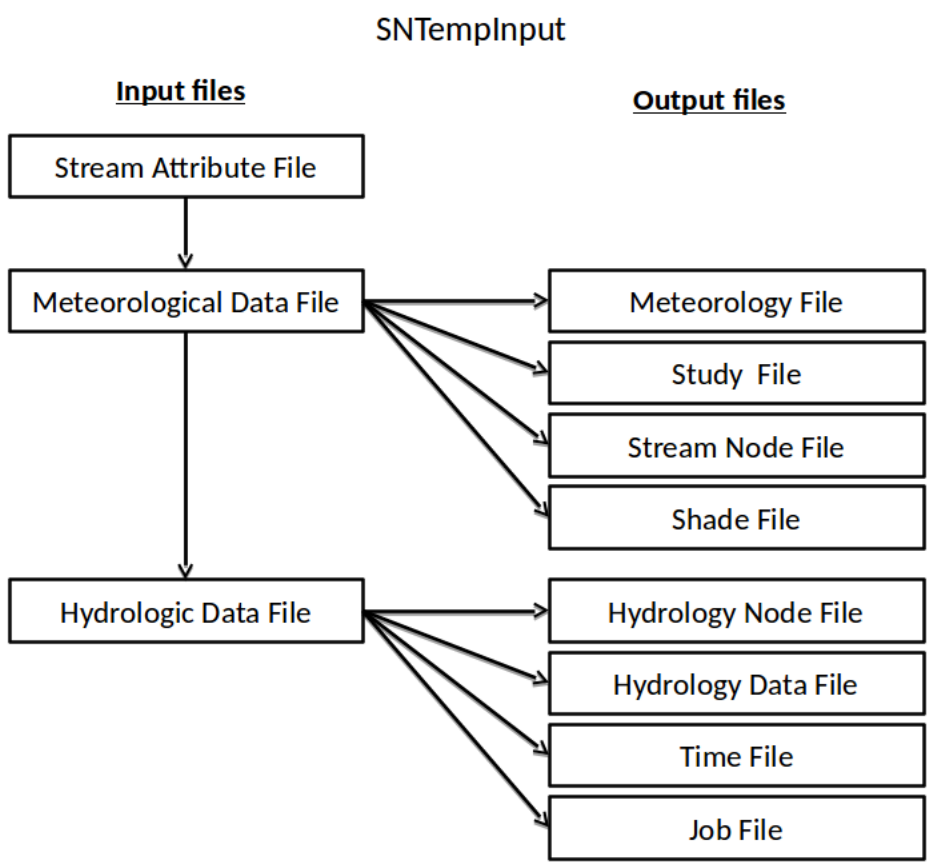

Figure 1. Sequence of files to be built for SNTEMP input. Files on the left are required user supplied data containing stream attributes, meteorological, and hydrologic data. Files are created by running SNTempInput. These files are fully formatted for input into SNTEMP.

matted SNTEMP input files. The user must supply three different input files to *SNTempInput*; a stream attributes file, a meteorological data file, and a hydrologic data file. Each of these files must be properly formatted to by read by *SNTempInput*, or the source code must be changed by the user to read a differently formatted input file. The sequence of SNTEMP input file creation and user supplied data files are shown in in Figure 1. The following section describes the formatting of each of the user supplied data files required as input to *SNTempInput*.

3.0 Input Files Required for *SNTempInput*

3.1 Stream Attributes

The stream attribute file contains one line of data for each stream segment. Each stream segment has 33 columns with information relevant for each stream segment. Table 1 provides a list of all stream attributes including variable name, units, and data type along with a brief description of each attribute. The format of the stream attributes file was adapted from *Markstrom* [2012] and Markstrom's report provides more information about many of the attributes. *Daraio and Bales* [2014] and *Markstrom* [2012] provide some information on how to calculate some of the stream attributes, and it is highly recommended that the user consult *Bartholow* [2000, 2002] or *Theurer et al.* [1984] for more details on stream attributes.

SNTempInput will properly format and create the SNTEMP input files based on the ordering provided in the stream attribute file, but if the stream segments are not ordered properly SNTEMP will not run. Error messages will be generated that provide an indication of potential errors in stream sequences. The details of properly ordering stream segments and stream nodes can be found in [*Theurer et al.,* 1984] and [*Bartholow,* 2002]. The version of *SNTempInput* available here must have stream numbering that does not exceed the total number of segments in the network. That is, if there are 25 segments in the network, they must be numbered 1-25.

3.2 Meteorological Data

The meteorological data file is read after the stream attribute file, and serves to define the simulation period in SNTEMP. The SNTEMP model requires basin average daily temperature (C°), basin average daily incoming solar radiation (Wm^{-2}), basin average daily wind speed (ms^{-1}), and basin average daily relative humidity (unitless). The provided source code is not written to read wind speed and relative humidity. Variables for wind speed and relative humidity are included in the source code and a minor change to the **readMetData** method in the **ReadData** class will allow for these variables to be read from a meteorological data file.

Below is a sample of the header and several lines of data from a PRMS formatted meteorological data file. The meteorological data are basin averaged daily precipitation (in), basin averaged daily temperature (C°), and basin averaged daily solar radiation (langleys). Basin averaged precipitation is not needed in SNTEMP. Basin averaged daily wind speed and relative humidity can be included in the meteorological data file with a simple alteration of the source code, as mentioned above.

```
# Begin DBF
"# timestamp,#FIELD_ISODATETIME,19,0"
"# one,#FIELD_DECIMAL,10,2"
"# basin_ppt,#FIELD_DECIMAL,10,2"
"# basin_temp,#FIELD_DECIMAL,10,2"
"# orad,#FIELD_DECIMAL,10,2"
# End DBF
#
timestamp one basin_ppt basin_temp orad
19d 10n 10n 10n 10n
2013-7-02:00:00:00 1 0.08    24.61 69.53
2013-7-03:00:00:00 1 0.01    25.45 84.26
2013-7-04:00:00:00 1 0     26.85 147.31
2013-7-05:00:00:00 1 0     27.37 145.90
2013-7-06:00:00:00 1 0     28.66 149.30
```

The SNTEMP meteorology input file is written after the *SNTempInput* meteorological data input file since all the required data are available to create this SNTEMP input file. Note that leap years with 366 days of data in the *SNTempInput* meteorological data input file are shortened by one day when writing the SNTEMP meteorological input file. Source code included in the utilities package will find leap years and trim one day of data from the file.

Below is a sample of source code to read meteorological data from the *SNTempInput* meteorological input data file. A line of data is read as a string then parsed into components. Wind speed and relative humidity can be included by changing the code on lines 11 and 12 below. (The code snippet is from class ReadData method readMetData.) The format to read the meteorological data file can be adjusted in the code as needed, however do not change the variable names unless you use refactor in Netbeans so that all of the instances of the variable are changed.

Col	Field	Description	Units	Data Type
1	arcid	Identifier of stream segment automatically generated in ArcMap	NA	integer
2	streamID	Stream segment ID number	NA	integer
3	toNode	ID number of downstream segment	NA	integer
4	length	Length of stream segment	meters	double
5	upNodeAlt	Elevation of upstream node	meters	double
6	downNodeAlt	Elevation of downstream node	meters	double
7	upNodeLat	Latitude of upstream node	decimal degrees[1]	double
8	downNodeLat	Latitude of downstream node	decimal degrees[1]	double
9	upNodeX	x-coordinate of upstream node	meters	double
10	upNodeY	y-coordinate of upstream node	meters	double
11	downNodeX	x-coordinate of downstream node	meters	double
12	downNodeY	y-coordinate of downstream node	meters	double
13	width	Channel width	meters	double
14	widthExp	Exponent in channel width equation[2]	NA	double
15	manning	Manning roughness coefficient *(n)*	NA	double
16	grdTemp	Mean annual ground temperature	C	double
17	thermGrad	Thermal gradient across stream	$Jm^{-2}s^{-1}C^{o-1}$	double
18	eastAlt	Topological altitude of east side of stream. Angle from stream center to top of stream bank	radians[2]	double
19	eastCrown	Height to forest crown on east side of stream segment	meters	double
20	eastVegHt	Height of understory vegetation on east side of stream segment	meters	double
21	eastVegOff	Distance of vegetation from channel banks on east side of stream segment	meters	double
22	eastSumDen	Summer vegetation density on east side of stream segment	NA	double
23	eastWinDen	Winter vegetation density on east side of stream segment	NA	double
24	westAlt	Topological altitude of west side of stream. Angle from stream center to top of stream bank[2]	radians	double
25	westCrown	Height to forest crown on west side of stream segment	meters	double
26	westVegHt	Height of understory vegetation on west side of stream segment	meters	double
27	westVegOff	Distance of vegetation from channel banks on west side of stream segment	meters	double
28	westSumDen	Summer vegetation density on west side of stream segment	NA	double
29	westWinDen	Winter vegetation density on west side of stream segment	NA	double
30	streamName	Name of stream (limit 12 characters)	NA	string
31	streamAz	Orientation of stream segment[2,3]	decimal degrees	double
32	upNode	Type of node at upstream end of segment	NA	string
33	downNode	Type of node at downstream end of segment	NA	string

Table 1. Variable name and description of stream segment attributes. The order of the variable list represents the order in which stream attributes are read in SNTempInput. Developed and adapted from *Markstrom* [2012].

[1]Converted to radians within SNTempInput.
[2]See [*Theurer et al.*, 1984] or [*Bartholow,* 2002] for details.
[3]Decimal degrees ranging from -90 to 90.

```
for (i = 0; i < line.length; i++) {x
line[i] = in.next();
String[] token = line[i].split("[-:]");
basinYear[i] = Integer/parseInt(token[0]);
basinMonth[i] = Integer.parseInt(token[1]);
basinDay[i] = Integer.parseInt(token[2]);
int discard = in.nextInt();
basin_ppt[i] = in.nextDouble();
basin_temp[i] = in.nextDouble();
orad[i] = in.nextDouble();
wind[i] = 0;
rh[i] = 0;}
```

3.3 Hydrologic Data

The hydrologic data file is read after the meteorological data file is read and after the following SNTEMP input files are written: meteorology file, study file, stream node file, and shade file. The code was developed to read hydrologic simulation output from PRMS (see *Daraio and Bales* 2014), therefore the hydrologic data input file format follows that of the PRMS output file. Below is a sample of header and lines of data from a PRMS formatted hydrologic data file. Each line of the data file is for daily averaged values for each stream segment. The order in which stream segments are listed in this file does not matter. The time stamp is year-month-day:hour:minute:second; only year, month, and day are read. The next column is the stream segment ID, followed by mean daily total flow in the stream segment (fts^{-1}), mean daily groundwater flow into the stream segment (fts^{-1}), mean daily surface runoff into the stream segment (fts^{-1}), mean daily sub-surface or interflow into the stream segment (fts^{-1}), and mean daily incoming solar radiation for the stream segment (langleys)

```
#
 # From HMS output file W:\SNTempInput\hmsOut.txt
# Only surface runoff is provided
# Assume constant baseflow
#
timestamp nsegment segment_cfs gwflow runoff
ssflow srad
Some meaningless line from PRMS is here
2013-7-02:00:00:00 1 2.178 0.500 1.678 0.00
200.00
2013-7-02:00:00:00 10 1.100 0.500 0.600 0.00
200.00
2013-7-02:00:00:00 11 1.029 0.500 0.529 0.00
200.00
2013-7-02:00:00:00 12 0.932 0.500 0.432 0.00
200.00
2013-7-02:00:00:00 13 0.660 0.500 0.160 0.00
200.00
```

It was determined that it was easier to develop code to convert simulated stream flow output data from other models into the PRMS format rather than develop code to read the formatted output from other hydrologic software. As an example, a HEC-HMS model for a small watershed was linked to SNTEMP model. The source code provided has a method that converts output from the HMS model to PRMS formatted data to read into SNTEMP. The output was created using HEC-DSSVue. The HEC-HMS model output file (*.dss) is read using HEC-DSSVue [USACE, 2014] using the "Math Functions" menu and printing the 1 day average flow for each reach (stream segment) in the model. Users can use the code developed for this purpose in the method **readHMShydrologyData** in the **ReadData** class to reformat hydrology data for input to *SNTempInput*. The following is a snippet of code for writing the properly formatted hydrology data file. Note that "year" and "day" are strings while "month" is an integer, which is a result of how the HEC-DSSVue data file was read. The year month, and day are read as integers from the hydrology data file, but the user can easily change the data types as needed.

```
for(i=0;i<numSegments;i++){
   dout.printf("%s-%d-%s:00:00:00\t%d\t%.4f\t%f\
t%f\t%f\t%f%n",
   year[i], month[i], day[i], segment[i],
totalSegmentFlow_cfs[i],
   groundwaterInflow_cfs[i], surfaceInlow_cfs[i],
   interflowInflow_cfs[i], solarRad[i]);
}
```

4.0 Software Description

The source code was written in the Java® programming language using Netbeans IDE 8.1, and the source code is available as a Netbeans project. Netbeans is free cross-platform software that runs on Windows, Mac OS X, and Linux operating systems, and is available from Oracle [2013]. Working from the source code, it is recommended that the code be compiled and run in Netbeans.

4.1 Class Descriptions

The *SNTempInput* Netbeans project is composed of two source packages, ***sntempinput*** and utilities, each with several classes. A javadoc is available in the package folder that summarizes the classes (`/SNtempInput_dist/SNtempInput/dist/javadoc`). The main class in ***sntempinput*** is SNTempInput, which calls methods to read the required input files and write the SNTEMP input files (Table 2). All variables are declared in the superclass CreateFiles from which the subclasses Calculations, ReadData, and WriteInputFile are derived. The ***utilities*** source package contains several classes that contain methods that are peripheral to *SNTempInput* but required to run the program (Table 3).

4.2 Sample Files

Sample input files for *SNTempInput* are provided along with the source code including a stream attributes file, a meteorological data file, and a hydrologic data file. The source code contains comments that will guide the user through the program.

5.0 Summary

The SNTEMP model is time consuming to set up and get running given the large amount of data required for the model and, in particular, because the model requires the creation of at least 7 formatted input files. A general open source package, *SNTempInput* was developed to create the required SNTEMP input files from a defined stream network that includes several physical attributes of stream segments along with meteorological and hydrological data to drive the SNTEMP simulations. This paper has described how to format the required data files and use *SNTempInput* to create properly formatted input files for SNTEMP and provide guidance as to how and where to make any necessary modifications to the source code as required by the user. While all SNTEMP input files will be properly formatted for use with the SNTEMP model, there is no guarantee that the SNTEMP model will run properly as there are many other structural requirements for the model. The *SNTempInput* source package is intended to reduce the amount of time required to set up and run SNTEMP simulations to facilitate stream temperature related research and practice. Please contact the author with any questions about the source code.

Software Availability

The source code with example *SNTempInput* input files can be downloaded in zipped format (110 kb) free of charge at http://www.hydroshare.org/resource/18e0a54ff30c4ca1bbe-af71223a790ed, or by contacting the developer. Joseph Daraio, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, 240 Prince Phillip Drive, St. John's, NL, A1B 3X5. Phone: 1-709-864-2756. Email: jadaraio@mun.ca. The software was first available in 2014 and will run with minimal hardware requirements. The unzipped package includes the base directory `SNtempInput_dist` that includes the Netbeans project folder and sample files described below. The Java source code can be found in the `/SNtempInput_dist/SNtempInput/src/sntempinput/` and `/SNtempInput_dist/SNtempInput/src/utilities/` folders. The Windows executable jar file `/SNtempInput_dist/SNtempInput/dist/SntempInput.jar` (44 kb) was compiled using Netbeans on Windows 10.

References

Bartholow, J. M. (2000), The Stream Segment and Stream Network Temperature Models: A Self-Study Course, *Tech. rep.*, U.S. Geological Survey Open-File Report 99–112.

Bartholow, J. M. (2002), SSTEMP for Windows: The Stream Segment Temperature Model (Version 2.0), *Tech. rep.*, US Geological Survey.

Caissie, D. (2006), The thermal regime of rivers: a review, *Freshwater Biol., 51,* 1389–1406.

Cheng, S.-T., and M. J. Wiley (2016), A Reduced Parameter Stream Temperature Model (RPSTM) for basin-wide simulations, *Environ. Modell. Software, 82*(C), 295–307.

Cox, M. M., and J. P. Bolte (2007), A spatially explicit network-based model for estimating stream temperature distribution, *Environ. Modell. Software, 22*(4), 502–514.

Class	Description
Calculations	Contains methods to calculate the distance from outlet node E of the SNTEMP model to the upstream and downstream nodes of each stream segment.
CreateFiles	SNTEMP input files.
LandSurfaceTemper-ature Calculation	Calculates the lateral inflow temperature of water that is added to a stream segment in SNTEMP
ReadData	Contains methods to read stream attributes file, read meteorological data, and read hydrologic data.
SNtempInput	This is the main program that calls methods to read output from a hydrologic model and create the required input files for SNTEMP.
WriteInputFile	This is the main program that calls methods to read output from a hydrologic model and create the required input files for SNTEMP.

Table 2. Classes And Class Descriptions In The ***Sntempinput Source Package.***

Class	Description
DecisionBox	Defines a decision box as an interface for the user to select the proper files and/or directories
executeDosCommand	Opens and executes the command sent to it in a DOS window
GetNumLines	Holds the method to get the number of lines in a file; use to allocate memory for arrays
JulianDate	Methods to convert a Gregorian date (month, day, year) into a Julian date or day of the year
LeapYear	Contains method to determine if a given year is a leap year

Table 3. Classes and class descriptions in the ***utilities source package.***

Daraio, J. A., and J. D. Bales (2014), Effects of Land Use and Climate Change on Stream Temperature I: Daily Flow and Stream Temperature Projections, *J. Am. Water Resour. Assoc., 50*(5), 1155–1176. doi:10.1111/jawr.12179

Daraio, J. A., J. D. Bales, and T. J. Pandolfo (2014), Effects of Land Use and Climate Change on Stream Temperature II: Threshold Exceedance Duration Projections for Freshwater Mussels, *J. Am. Water Resour. Assoc., 50*(5), 1177–1190. doi: 10.1111/jawr.12179

Ficklin, D. L., I. T. Stewart, and E. P. Maurer (2013), Effects of climate change on stream temperature, dissolved oxygen, and sediment concentration in the Sierra Nevada in California, *Water Resour. Res., 49*(5), 2765–2782.

Gaffield, S. J., T. W. Rayne, L. Wang, and K. R. Bradbury (2007), Impacts of land use and groundwater flow on the temperature of Wisconsin trout streams: Final report, *Tech. rep.*, Wisconsin Geological and Natural History Survey, Open-File Report 2007-09.

Hester, E. T., and K. S. Bauman (2013), Stream and retention pond thermal response to heated summer runoff from urban impervious surfaces, *J. Am. Water Resour. Assoc., 49*(2), 328–342, doi:10.1111/j.1752-1688.2011.00525.x

Isaak, D. J., S. Wollrab, D. Horan, and G. Chandler (2012), Climate change effects on stream and river temperatures across the northwest U.S. from 1980—2009 and implications for salmonid fishes, *Clim. Change, 113*, 499–524.

Kaushal, S. S., G. E. Likens, N. a. Jaworski, M. L. Pace, A. M. Sides, D. Seekell, K. T. Belt, D. H. Secor, and R. L. Wingate (2010), Rising stream and river temperatures in the United States, *Front. Ecol. Environ.,* 8(9), 461–466, doi:10.1890/090037.

Mantua, N., I. Tohver, and A. Hamlet (2010), Climate change impacts on streamflow extremes and summertime stream temperature and their possible consequences for freshwater salmon habitat in Washington State, *Clim. Change, 102*(1-2), 187–223, doi:10.1007/s10584-010-9845-2.

Markstrom, S. L. (2012), P2S—coupled simulation with the Precipitation-Runoff Modeling System (PRMS) and the Stream Temperature Network (SNTemp) Models, *Tech. rep.*, U.S. Geological Survey Open-File Report 2012–1116.

Null, S. E., J. H. Viers, M. L. Deas, S. K. Tanaka, and J. F. Mount (2013), Stream temperature sensitivity to climate warming in California's Sierra Nevada: impacts to coldwater habitat, *Clim. Change, 116*(1), 149–170, doi:10.1007/s10584-012-0459-8

Oracle (2013), Netbeans IDE.

Poole, G. C., and C. H. Berman (2001), An ecological perspective on in-stream temperature: Natural heat dynamics and mechanisms of human-causedthermal degradation, *Envir. Manag., 27*(6), 787–802, doi:10.1007/s002670010188

Theurer, F. D., K. A. Voos, and W. J. Miller (1984), Instream water temperature model., *IFI Paper 16 FWS/OBS-84/15,* U S Fish and Wildlife Service.

USACE (2014), HEC-DSSVue http://www.hec.usace.army.mil/software/hec-dssvue/ (Accessed May 26, 2014).

USGS (2013), P2S-coupled simulation with the Precipitation-Runoff Modeling System (PRMS) and the Stream Temperature Network (SNTemp) Models.

Webb, B. W., and F. Nobilis (2007), Long-term changes in river temperature and the influence of climatic and hydrological factors, *Hydro. Sci. J.*, 52(1), 74–85.