

Estimating Sediment Concentration and Load

Contents

Estimating Sediment Concentration and Load	1
Contents	1
General Guidelines.....	2
Software	2
Surrogate variables.....	2
Defining the relationship between surrogate and SSC	3
Annual loads	3
Storm event loads.....	3
Dividing the data.....	3
The form of the relationship	3
Linear regression.....	3
Transformed linear regression and bias corrections	4
Power function	5
Loess	5
Hysteresis and pairwise fitting.....	6
Linear time-interpolation	6
Variance of the estimated sediment load	7
Simple linear regression.....	7
Log-log regression	7
Power functions	7
Loess	8
Coefficient of variation	8
Statistical criteria for selecting among models	8
Coefficient of determination (r^2)	8
Residual standard error (s).....	9
Coefficient of the variation of the estimated load (CV).....	9
Other statistics.....	9
Subjective quality criterion	9
Procedures For Developing TTS Sediment Loads in R.....	11
Overview	11
General rules for specifying function arguments.....	12
Reading in the data: read.flo and read.lab	12
Required arguments	13
Optional argument	13
Merging the lab and electronic data: merge.flo	13
Required arguments	14
Optional arguments.....	14
Scatterplot functions: turbsscplot and qsseplot	14
Definitions of scatterplot functions' required arguments	14
Definitions of scatterplot functions' optional arguments.....	14
Prediction functions: turbsrc , flowsrc , lineartime , cubictime	15
Definitions of prediction functions' required arguments.....	16
Definitions of prediction functions' optional arguments.....	16

Summary stats: total , tts.tot , msc.tot , and sum.tot	17
Examples of summary stats on objects produced by prediction functions	18
Summary plots: ttsplot	18
Definitions of ttsplot required arguments	18
Definitions of ttsplot optional arguments.....	18
Functions to save results and export to text file: tts.ssc and write.ssc	18
Required arguments	19
Required arguments	19
Optional arguments	19
Cross-section coefficients (discoef).....	19
Customization	20
REFERENCES	Error! Bookmark not defined.

General Guidelines

After the complete data set has been corrected, the primary task required for computing sediment flux is to estimate SSC at the same frequency as discharge and turbidity were measured. Once the SSC has been estimated, the sediment load (load, flux, and yield are all synonymous terms) can be simply calculated as

$$L = \sum_i ktq_i c_i \quad (1)$$

where t is the time between measurements, and q_i and c_i are the instantaneous water discharge and estimated SSC for interval i . Generally, the conversion factor k will also be required to express the result in the desired measurement units. Before the advent of computers it was common to sum average fluxes, weighted by flow duration, over flow classes (i.e. the flow duration sediment rating curve method). However, with today's computing power, there is no longer any reason to use such an approximate method.

Software

Most of the methods described in this section are available in a set of procedures developed by Redwood Sciences Laboratory for use within [R, a free software package](#) for exploratory data analysis. Instructions for using the procedures, after they are installed, are described [below](#). The procedures are designed to be used for estimating sediment loads for input files created specifically by [TTS Adjuster](#). Contact the authors to obtain the software. The [FTS processing system](#), still under development at this writing is planned to include many of these methods as well.

Surrogate variables

Instantaneous SSC is normally estimated using one of three surrogate variables: turbidity, discharge, or time. Turbidity is the preferred surrogate. If the turbidity data are of high quality and if the range of turbidity is not too narrow, the relationship between turbidity and SSC is usually very good and certainly better than the relationship between discharge and SSC. However, when turbidity data are of poor quality or missing, it may be necessary to fall back on discharge as the surrogate. If enough pumped samples were collected during the period of bad turbidity data, then

instead of using a relationship between discharge and SSC, a better estimate of SSC can be obtained simply by interpolating SSC over time between samples. In the interpolation scenario, the surrogate variable is time.

Defining the relationship between surrogate and SSC

The selection of the data and determination of the surrogate relationship(s) are subjective, but must be guided by general principles. The data used to determine a surrogate relationship should ideally

1. be representative of the period whose flux is being estimated.
2. span the range of both variables in the period whose flux is being estimated
3. include enough samples to reasonably define the relationship

Annual loads

For calculating an annual sediment flux, assuming the turbidity data are complete, it may be reasonable to fit a single relationship between turbidity and SSC based on a whole year's data. Such a relationship is likely to overestimate some storm events and underestimate others, but these errors tend to balance one another and the annual flux may often be estimated accurately enough (within 10%) by the one relationship.

Replacing the estimated SSC during storm events using storm-based relationships can result in a more accurate estimate of annual load, and this can be used to validate or reject the first procedure.

Storm event loads

For calculating a storm event load, it is best to use only data from that storm event, assuming enough samples were collected. Using a relationship based on the entire year or, worse, from a prior year is likely to severely miscalculate a storm event load. If an inadequate number of samples were collected to reasonably define the relationship, or if the samples collected do not span the range of turbidity and/or SSC for the given event, then additional samples should be included from neighboring periods.

Dividing the data

Just as with annual loads, it may be more accurate to divide a storm event into multiple periods and multiple relationships. This decision should be based on an examination of scatterplots between turbidity or discharge and SSC. If the samples in the scatterplot are numbered, then it is easy to tell if a relationship has shifted during the event. If the relationship clearly shifted, and if enough samples were collected to define both relationships, it is usually best to divide the event into two periods. The drawback to dividing the data is that the individual relationships will be less precise because they are defined by fewer points.

The form of the relationship

Linear regression

Linear regression is usually the simplest relation considered. If several different models seem to fit the scatterplot adequately, the simplest relationship is usually preferred. However, there are other considerations. When dissolved substances are present that cause turbidity, or when SSC is determined by filtration and the filtrate contains fines that cause turbidity, concentrations near zero will have significantly positive turbidity and linear relationships between turbidity and SSC will

have a negative intercept. The result is usually that predictions of SSC are negative for some range of low turbidity values that have been recorded. The solution is to either set negative predictions to zero, or to adopt a model that never predicts negative SSC, e.g. log-log regression or a power function.

Transformed linear regression and bias corrections

Transformations can be used to produce models that cannot make negative predictions, to linearize relationships, to normalize residuals, to equalize variance, or a combination of the above.

Sometimes a transformation can accomplish more than one of these objectives, but there are no guarantees. Retransformed predictions from log-log regressions are always positive, but they have the draw back that they cannot be evaluated when the predictor is zero. The square root transformation may accomplish the same thing without eliminating predictions at zero.

Transformation of the response (SSC) has a drawback. The prediction must be retransformed back to the original units, and that step introduces a bias (Miller, 1984; Koch and Smillie, 1986). The bias arises because regression predicts the mean of a normal distribution for a given x , and the transformed mean of a normal distribution is not equivalent to the mean of the transformed distribution. To correct for retransformation bias, Duan (1983) introduced the non-parametric “smearing” estimator

$$\hat{y}_{sm} = \frac{1}{n} \sum_{i=1}^n h(\hat{y}_0 + e_i) \quad (2)$$

where \hat{y}_{sm} is the corrected prediction in original units, h is the inverse transformation, \hat{y}_0 is the prediction before retransformation, and e_i are the residuals from the regression.

Log transformation. For transformations by \ln (logarithm to the base e), the bias is always negative, and increases with regression variance. The smearing estimator becomes $\hat{y}_{sm} = \exp(\hat{y}_0)\bar{e}$ where $\bar{e} = \sum e_i / n$. If the \ln -transformed regression has normally distributed residuals, there is an exact, minimum variance unbiased estimator (Cohn et al., 1989), but it is quite complex and computationally demanding. A more widely known, approximate correction for \ln transformations with normally distributed residuals is

$$\hat{y}_{qmle} = \exp(0.5s^2 + \hat{y}_0) \quad (3)$$

where s is the residual standard error of the transformed regression. The QMLE subscript stands for quasi-maximum likelihood estimator (Cohn et al., 1989). This estimator has also been called the naive correction, or the Baskerville (1972) correction. More generally, for logarithms to the base b , the QMLE correction factor is $\exp[0.5\ln(b)^2 s^2]$, which, for base 10 logarithms, yields $\hat{y}_{qmle} = \exp(2.65s^2 + \hat{y}_0)$.

Square root transformation. For square root transformations, the smearing estimator takes the form $\hat{y}_{sm} = \hat{y}_0^2 + 2\hat{y}_0\bar{e} + \bar{e}^2$ where, as before \bar{e} is the mean residual.

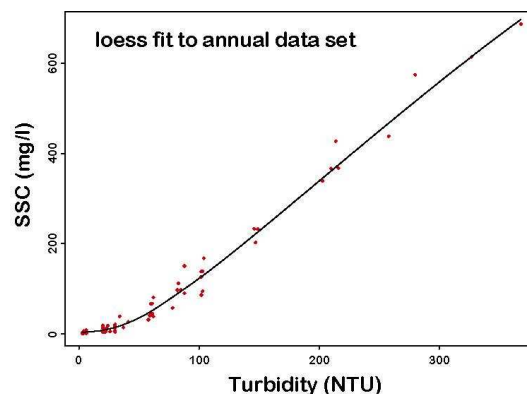
Power function

The power function $y = cx^k$ can be obtained by retransforming $\log_b(y) = a_0 + a_1 \log_b(x)$ to obtain $c = b^{a_0}$ and $k = a_1$, but it can also be fitted directly using non-linear least squares (NLS) estimation (Bates and Watts, 1988). The estimated coefficients will be different from the retransformed linear regression, with or without correction. The log-log regression gives more weight to data with small x . Algorithms for NLS are iterative calculations that are provided in many statistical packages such as Splus and R. Fitting a power function by NLS instead of log-transformed regression has the advantages that no bias-correction is necessary and data with large x are not de-emphasized. (Note: in Microsoft Excel, a power function is computed by retransforming the log-log regression. Excel ignores the bias correction, so predictions based on Excel power models are negatively biased).

Loess

Locally weighted regression (loess) is a non-parametric curve-fitting technique that can estimate a very wide class of regression functions without distortion (Cleveland, 1979; Cleveland et al., 1988). It is similar to a moving average. The fitted value at each x is the value of a regression fit to data near x using weighted least squares, with the closest points more heavily weighted. The amount of smoothing, which affects the number of points in the regression, is determined by the user. The advantage of loess is its flexibility - it nearly always can produce an acceptable fit to the data – but care must be taken not to over-fit the data. With SSC and turbidity, we recommend smoothing until no more than one inflection point (where curvature switches between convex and concave) is present. A statistic that characterizes the amount of smoothing is the *equivalent number of parameters*, which can be compared to the number of parameters estimated by a polynomial fit. The loess computation seems to require at least 2 more data points than the equivalent number of parameters, so it cannot generally be applied with less than about 5 samples.

At many sites, organic sediment becomes dominant at low values of turbidity, causing a reduction in the slope of the rating curve near zero. The slope change is most apparent in annual plots of turbidity and SSC. Loess is often the most satisfactory modeling method to capture this slope change.



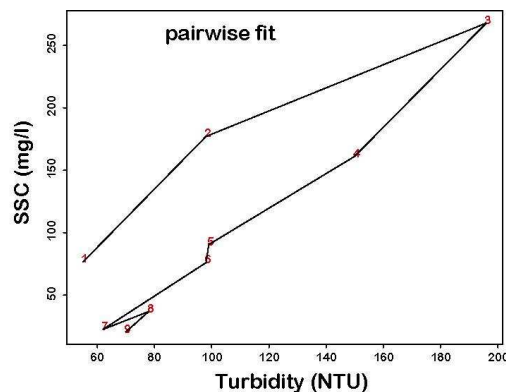
Prediction. Since there is no simple algebraic expression for a loess curve, prediction is usually done within the statistical package that computed the curve. Splus and R have prediction capabilities, but they do not permit extrapolation beyond the range of the data used to fit the model.

Extrapolation can be done manually by computing regression lines for the last few points of the curve at either end of the range.

Hysteresis and pairwise fitting

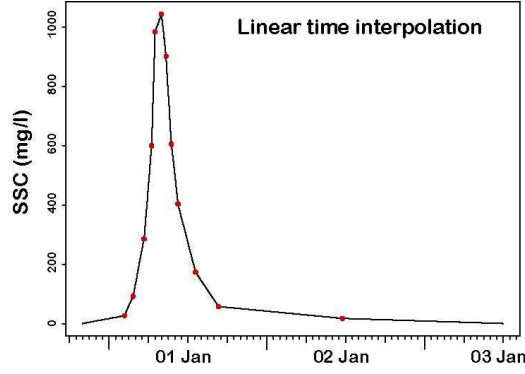
It is fairly common for the relationship between surrogate and target variable to shift during an event. The shift is usually near the turbidity or discharge peak and, on a scatterplot, the shift often appears as a loop pattern. This is known as hysteresis. Hysteresis loops for SSC and turbidity are less common and narrower than loops for SSC and discharge. It might be possible to approximate the loop as two curves, in which case the data could be divided into two periods, [as discussed above](#), applying two of the previous methods to estimate the load.

Another approach to hysteresis loops, which is occasionally useful, is to connect each consecutive pair of samples with a line segment, using each segment to predict the period between that pair's sample collection times. The reason this method is only occasionally useful is that, if turbidity or discharge is the surrogate variable, all the line segments must have positive slopes to produce a reasonable result. If any of the segments have negative slope, and some usually do have negative slope near the peak, the surrogate time series for that segment will be inverted, i.e. surrogate peaks will be converted to SSC troughs and vice versa.



Linear time-interpolation

The one situation in which pairwise fitting is more commonly useful is when time is the surrogate variable. In that case, pairwise fitting becomes linear interpolation between consecutive sample times. SSC need not always move in the same direction as time, so the positive slope restriction, which makes sense when turbidity or discharge is the surrogate, does not apply. As mentioned when [surrogate variables](#) were first discussed, if enough pumped samples were collected during a period of bad turbidity data, then instead of using a relationship between discharge and SSC, a better estimate of SSC can often be obtained simply by linear time-interpolation.



Variance of the estimated sediment load

The variance of the estimated sediment load is the variance of the sum of the estimated fluxes for all the intervals in the period being estimated. The variance of a sum is the sum of the variances and covariances among the items being summed. The covariances cannot be ignored because flux estimates for similar values of turbidity are highly correlated. The covariance matrix of estimated fluxes is derived from the covariance matrix of the regression coefficients, assuming that the surrogate variable for intervals to be predicted is just a set of constants (not random).

Simple linear regression

The derivation is not difficult for simple linear regression, but the computation is demanding because the dimension of the covariance matrix of estimated fluxes is $N \times N$, where N is the number of intervals being predicted. The elements of this covariance matrix must be summed to obtain the estimated variance of the load:

$$V = s^2 Z'(X'X)^{-1} Z \quad (4)$$

where s is the residual standard error of the regression; X is the $n \times 2$ design matrix whose rows are $(1, x_i)$, x_i is the i th sampled turbidity value; and Z is the $N \times 2$ matrix whose rows are $(ktq_j, ktq_j x_j)$, q_j and x_j are the discharge and turbidity for the j th interval to be predicted, t is the length of the interval between instantaneous discharge readings, and k is a units conversion factor that expresses the load in the desired units.

Log-log regression

Variance computation is much more complicated for retransformed predictions from log-log regression (Gilroy et al., 1990).

Power functions

An approximate covariance matrix can be obtained for power functions using the delta method for vector-valued functions (Rice, 1994). If the power function is $b_0 X^{b_1}$, where X is an $N \times 1$ vector of surrogate variables, the $N \times N$ covariance matrix of estimated concentrations is

$$C \equiv [X^{b_1}, b_0 \ln(X) X^{b_1}] V(B) [X^{b_1}, b_0 \ln(X) X^{b_1}]' \quad (5)$$

where $\mathbf{B}=(b_0, b_1)'$ and $V(\mathbf{B})$ is the coefficient covariance matrix estimated using non-linear least squares. The dimensions of the first bracketed matrix is $N \times 2$, $V(\mathbf{B})$ is 2×2 , and the last matrix is $2 \times N$. The covariance matrix of estimated fluxes also includes the discharge, time interval, and units conversion factor:

$$V = (kt)^2 \mathbf{Q} \mathbf{C} \mathbf{Q} \quad (6)$$

where \mathbf{Q} is an $N \times N$ diagonal matrix containing the discharges for each interval, and k and t are scalars defined as before.

Loess

The author is not aware of any methods that have been developed for estimating the covariance matrix for loess predictions or derived fluxes.

Coefficient of variation

A more intuitive way of expressing the variance of the estimated sediment load is the coefficient of variation, which is the standard error as a proportion or percentage of the estimated load. The standard error is just the square root of the variance, so the coefficient of variation is given by

$$CV = 100 \frac{\sqrt{V}}{L} \quad (7)$$

where V is the sum of the elements of the matrix V defined by [equation \(4\)](#), or [equations \(5\) and \(6\)](#), above and L is the estimated sediment load defined by [equation \(1\)](#) above. In the R sediment load procedures a slightly different version of CV is calculated for QMLE and Duan's smearing estimator. Since they are both slightly biased, an estimate of mean square error from Gilroy et al. (1990) is substituted for V . Mean square error is the sum of variance and the square of bias.

Statistical criteria for selecting among models

There are several statistical criteria available for selecting between competing models. But no statistic should be used as a yardstick without considering the other factors mentioned previously in the sections [Storm event loads](#) and [The form of the relationship](#). Models that are based on different data sets (e.g. with or without certain data pairs) should not be selected based on a statistic. And for small sample sizes, statistics are not very reliable. All other things being equal, the model that expresses the relationship in simplest terms should be the model chosen. In particular, models with too many parameters will tend to have greater prediction error.

Coefficient of determination (r^2)

The coefficient of determination, r^2 , is an expression of the shape of a scatterplot. It is the squared correlation of predicted and observed values and can be interpreted as the proportion of total variance of y that is explained by x , a value always between 0 and 1. r^2 depends strongly on the range of the data sampled. If the range of x is very small, on the order of the residual standard error, then r^2 will be close to 0. If the range of x is much larger than the residual standard error, even due to one extreme value, then r^2 will be close to 1. r^2 can be heavily influenced by a few points, so it is best to not to use r^2 if the values of x are not evenly distributed. Because r^2 is

unitless, it is possible to compare r^2 between transformed and untransformed regressions. It is available from loess as well as linear and non-linear regression models.

Residual standard error (s)

Residual standard error is an expression of the average deviation of points from the regression line. It is expressed in the same units as the y measurement, therefore it cannot be used to compare regressions on transformed and untransformed data. It is available from loess as well as linear and non-linear regression models.

Coefficient of the variation of the estimated load (CV)

Coefficient of variation is a good yardstick for comparing loads estimated by transformed and/or untransformed linear regression, but may not be available for models other than linear regression. For small sample sizes, CV may be a poor error estimate, and is likely to be the least reliable of the criteria mentioned here.

Other statistics

There are other statistical criteria for selecting among competing regression models that are beyond the scope of this report. These include PRESS (prediction residual sum of squares), Mallows' C_p , and Akaike's information criterion (AIC). In recent years, various versions of AIC have become widely accepted for selecting models (with the same response) that have differing numbers of parameters. A corrected AIC has been formulated for loess models (Hurvich et al., 1998).

Subjective quality criterion

There are many considerations that determine the quality of a sediment load estimate determined using the methods described in this report. The following factors can be used to qualitatively rate a sediment load estimate.

A. Sediment sample coverage

1. How many samples were collected?
 - a. For a simple turbidigraph with one peak and no spikes, need at least 4 samples, and more is better.
 - b. Extra samples are desirable for complex turbidigraphs with multiple peaks or spikes. We'd like to have at least 2-3 samples for each additional extended peak and one sample for each spike.
2. Are there samples covering the whole range of turbidity for the storm? If the range of samples is too small, the regression slope will be unreliable, and extrapolation errors are likely. If the range or number of samples is too small, samples from a neighboring storm event should be considered for inclusion.
3. Are samples well-distributed temporally?
 - a. Were samples collected near the start, peak, and end of the storm?
 - b. Are there any extended periods with no samples?
4. If there were periods of bad or questionable turbidity, were samples collected during those times? These are needed to validate the turbidity and to permit time-interpolation if they show that the turbidity was bad.

B. Relationship of SSC to turbidity

1. What is the variance about the regression line? Note well: low variance doesn't mean a thing if you have only 2 data points or if one end of the regression line is based on a single point.
2. What is the coefficient of variation (*CV*) of the estimated load? This expresses the error of the estimate as a percentage. As with regression variance, if the sample coverage is poor this measure is meaningless.
3. Can the relation be expressed by a single regression? For a given data set, using more relationships divides the data into smaller groups, so there is greater uncertainty about each relationship.
4. Can the relation be expressed as a simple *linear* regression? Curvilinear fits are more susceptible to extrapolation error and require more data for support.

C. Quality of recorded turbidity

D. Quality of pumped samples

1. Were the volumes too low or too high?
2. Are there known problems in the transport or processing of the samples?
3. Were there any conditions (e.g. particle sizes, stream velocity, freezing conditions, pumping sampler problems) that might have compromised the quality or representativeness of pumped samples?

Procedures For Developing TTS Sediment Loads in R

Overview

1. Enter laboratory data into the *stnhy.isc* file, where *stn* is the 3-letter station name and *hy* is the 2-digit water year. File it in the proper locations as per the TTS Adjuster manual and help file. Double check for data entry errors.
2. Clean up stage and turbidity data using TTS Adjuster. This creates the *stnhy.flo* file.
3. Define the storms for which you want to estimate loads.
4. The rest is done in R.
5. Read the corrected data for one water year from the *stnhy.flo* file using **read.flo**, and from the *stnhy.isc* file using **read.lab**. Save the results in R objects named *stnhy.flo* and *stnhy.lab*.
6. Check for unmatched samples in both the electronic data and lab data using **mismatches**.
7. Merge the electronic data with each sediment sample using **merge.flo** to create an R object named *stnhy.sed*.
8. Plot concentration vs turbidity for a storm event (**turbscplot**).
9. If turbidity is good for the whole storm, and if SSC vs. turbidity is linear, then
 - a. Use **turbsrc** to estimate the load.
 - b. Save the results as an R object named however you like.
 - c. Plot the results with **ttsplot**.
10. If turbidity is good for whole storm but some predicted concentrations are negative or the relationship between SSC and turbidity is nonlinear or inconsistent, choose between the following alternative methods for estimating SSC from turbidity.
 - a. Use **turbsrc** with **type="logxy"** to avoid negative predictions. Zero is not permitted in the data. *CV* is calculated.
 - b. Use **turbsrc** with **type="sqrt"** to avoid negative predictions. Zero is permitted in the data. *CV* is not calculated.
 - c. Use **turbsrc** with **type="power"** option to fit a power function by non-linear least squares (similar to **type="logxy"** but no bias-correction is needed and calculations are simpler and faster). *CV* is approximated using the delta method.
 - d. Use **turbsrc** with **type="loess"** to fit a nonparametric loess model (locally weighted regression). Calculation of *CV* awaits further study and may not be possible.
 - e. Use **turbsrc** twice with the default linear regression, if there are 2 different linear relations.
 - f. Use **turbsrc** with **type="pairs"** if there is nonlinear hysteresis present and all pairs have positive slope. This does a piecewise fit using one pair of points at a time. It is likely to overfit the data and, in practice, is not used very often. If some pairs have a negative slope this method won't work well, because for those pairs predictions will have the opposite shape as turbidity. No statistics are available with this method.
 - g. Use **turbsrc** with the **type="logx"** option to try to linearize the scatter. This option is rarely helpful.
11. If turbidity is not good for whole storm, but there is a decent relationship between SSC and discharge for the periods when turbidity is not good, try using **flowsrc** to estimate loads from discharge for those periods. It is perfectly analogous to **turbsrc** in its usage except that the default method is **type = "logxy"** instead of **"linear"**. The two functions take all the same arguments. Save the results as an R object named however you like.

12. If neither **turbsrc** nor **flowsrc** does a good job of estimating SSC, you may still be able to get a good estimate using **lineartime** (linear time-interpolation between consecutive pairs) or **cubictime** (cubic spline time-interpolation). This works well when the samples are fairly close together in time, which often occurs in TTS when the turbidity sensor is fouled. Save the results as an R object.
13. Assign the results of each fit to an object, using the argument **long=T** (default) to include the time, predicted SSC, and (for **turbsrc** only) turbidity with the returned object, for use by **ttsplot()**.
 - a. Pass the objects returned by **turbsrc**, **flowsrc**, **lineartime**, and/or **cubictime** as arguments to **ttsplot** to get a plot of the whole storm and the total estimated sediment load. **ttsplot** can handle up to 5 estimation periods (a maximum of two objects created by **turbsrc** and three others).
 - b. If the concentration curves for two methods do not meet properly, a continuous transition can often be achieved by moving the boundary between the two periods forward or backward in time. You can get a continuous transition whenever methods applied to adjacent periods intersect, and this can be determined by including overlapping time periods in adjacent objects. Once the intersection, if any, has been determined, the objects should be recreated for non-overlapping periods that are contiguous at the intersection point.
14. After computing the sediment load for an event, save the predicted concentrations in an object using **tts.ssc**. Name the object *stnxx.ssc* or *stnxxx.ssc*, where *stn* is a 3-letter station designator, and *xx* or *xxx* is typically a storm number (but could be any 2 or 3 character identifying information). Run **write.ssc** to combine and export all objects named using that convention to a text file. The output will include date, time, flow, turbidity, turbidity quality, SSC, and method. The method identifies the surrogate used to estimate SSC: 1=turbidity, 2=discharge, 3=time.

General rules for specifying function arguments

To see function arguments at any time in R, use the **args** command (see below for examples). The **args** command shows the order and names of arguments that a function can take. The **args** function shows default values as *argname = defaultvalue*. When evoking a function in R, arguments can be specified in order or by name. If they are specified in order, the names of the arguments are not needed, but then to use an argument all preceding arguments must also be included. If they are specified by name, any order can be used. Often the first few arguments are specified in order and subsequent arguments are specified by name, for example:

```
> turbsrc("ftr", 99, 990201, 1200, 990204, 1200, interval=15, adj=F)
```

In the example, the first 6 arguments are specified in order, but *interval* and *adj* are specified by name.

Reading in the data: **read.flo** and **read.lab**

read.flo reads in a comma-delimited file containing year, Campbell julian day, time, dump, bottle, discharge, turbidity, and turbidity code. The input file is assumed to be in the *stn* folder, as per TTS Adjuster's file organization. Result is a data frame that you should name *stnhy.flo*.

read.lab reads in the comma-delimited *stnhy.isc* file, containing dump, bottle number, and lab concentration in the first 3 columns. The input file is assumed to be in the *stn/rawhy* subfolder, as per TTS Adjuster's file organization. Result is a data frame that you should name *stnhy.lab*.

```
> args(read.flo)
function (stn, hy, ttshome = ".")
```

```
> args(read.lab)
function (stn, hy, ttshome = ".")
```

Required arguments

stn 3-character string in quotes identifying the station (e.g. "nfc")
hy 2-digit water year

Optional argument

ttshome TTS data location in quotes, using double backslash ("\\") as subfolder/file delimiter. File **.RData** in the TTS home directory in order to use the default. The file system is assumed to be that used by TTS Adjuster and described in the Adjuster User's Manual. That is, the *stnhy.flo* file is located in a 3-letter station subfolder beneath *ttshome*, and the *stnhy.isc* file is located in a raw data subfolder, named **rawhy**, beneath the station subfolder. For example, **ttshome = "C:\\TTS"** specifies that Eel River 2004 data should be stored as **C:\\TTS\\eel\\eel04.flo** and **C:\\TTS\\eel\\raw04\\eel04.isc**.

Merging the lab and electronic data: **mismatches** and **merge.flo**

mismatches checks for unmatched bottles in *stnhy.flo* and *stnhy.lab*. The user should review the output to identify bottle numbering errors or data that still need to be entered. If changes to the data are needed, they should be made outside of R, and the data should be read back into R using **read.flo** and/or **read.lab** before proceeding.

merge.flo merges the data from *stnhy.flo* and *stnhy.lab* to create *stnhy.sed*. Result contains one line per pumped sample, with all pertinent info. Result is a data frame that you should name *stnhy.sed*.

```
> args(mismatches)
function (stn, hy)
```

```
> args(merge.flo)
function (stn, hy, all.lab = F, all.flo = F)
```

Required arguments for **mismatches** and **merge.flo**

stn 3-letter station name (e.g. "yoc")
hy 2-digit water year

Optional arguments for **merge.flo** (used primarily within the **mismatches** function)

all.lab logical; if **TRUE**, then extra rows will be added to the output, one for each pumped sample that has no matching **.flo** record. These rows will have **NA** in those columns that are usually filled with values from the **.flo** object. The default is **FALSE**, so that only rows with data from both **.flo** and **.lab** are included in the output.

all.flo logical; analogous to **all.lab** above, but output includes **.flo** records that have no matching **.lab** records.

Scatterplot functions: **turbsscplot** and **qsscplot**

turbsscplot plots SSC vs. turbidity for a specified period or any set of dumps and bottles.

qsscplot plots SSC vs. discharge for a specified period or any set of dumps and bottles.

```
> args(turbsscplot)
function(stn, hy, sdate, stime = 0, edate, etime = 2400, dumps, bottles,
        exclude = NULL, type = "linear", col = T, textsize = 0.6, span = 1,
        degree = 1, txt = "bottle", ...)
NULL
> args(qsscplot)
function(stn, hy, sdate, stime = 0, edate, etime = 2400, dumps, bottles,
        exclude = NULL, type = "logxy", col = T, textsize = 0.6, span = 1,
        degree = 1, txt = "bottle", units = "cfs", ...)
```

Definitions of scatterplot functions' required arguments

stn 3-letter station name (e.g. "nfc")
hy 2-digit water year
sdate start date (*yymmdd*), not required if dumps and bottles are specified
edate end date (*yymmdd*), not required if dumps and bottles are specified
dumps vector of dump numbers, same length as bottles. Not required if start and end date are specified.
bottles vector of bottle numbers, same length as dumps. Overrides the date/time selection of samples to use in the model. Not required if start and end date are specified.

Definitions of scatterplot functions' optional arguments

stime start time (*hhmm*), defaults to 0000
etime end time (*hhmm*), defaults to 2400

exclude	the plot excludes data with turbidity codes matching these (The NULL default causes nothing to be excluded)
type	character string: " linear ", " logx ", " logxy ", " sqrt ", " power ", " loess ", or " pairs "
col	logical value: whether or not to use color to identify data dumps
textsize	relative text size for plotting txt symbols
span	when type="loess" , this is the smoothing parameter used (positive values usually no greater than 1). Specifying " span=0 " will select the span, among values from 0.5 to 1.0, that minimizes corrected AIC (Hurvich et al., 1998). For storm events, this seems to always result in span=1.0 because of the relatively small sample sizes. Optimization is therefore most useful for annual data sets.
degree	when type="loess" , this is the degree of the local regression (1 or 2).
txt	name of column in sediment data frame to use as plotting symbol for samples (usually " bottle " or " dump ")
units	discharge units in input (sed) object. default is " cfs " (ft ³ /sec). The alternative is " cumecs " (m ³ /sec) (qsccplot only)
...	other graphical parameters that are passed through to plot() . Use help(par) for more information about graphical parameters.

Prediction functions: **turbsrc**, **flowsrc**, **lineartime**, and **cubictime**

turbsrc fits a turbidity/SSC rating curve to samples from any specified time period and uses the curve to predict SSC from turbidity. Can fit a linear function (**type="linear"**), linear function to log(x) (**type="logx"**), linear function to log(x) and log(y) (**type="logxy"**), linear function to sqrt(x) and sqrt(y) (**type="sqrt"**), power function fitted by non-linear least squares (**type="power"**), loess model (**type="loess"**), or pairwise linear functions (**type="pairs"**, see below). If only one set of dates and times are given it uses samples from the period being estimated. If two sets of dates and times are given, samples from the second period are used to fit a curve that is applied to the first period. *Or* dumps and bottles can be specified by number for the rating curve.

turbsrc(type="pairs") requires a little more explanation. It fits pairwise line segments, SSC as a function of turbidity, using **approx** (a built-in Splus function), to consecutive pairs of samples. Will not extrapolate however beyond the range of 2 samples before and after the point being estimated. Below the minimum turbidity represented by the pumped samples, it extrapolates using a line segment to (0,0) at the lower end, unless **opt=2**, in which case it extrapolates using a simple linear regression based on all samples in the specified period. Above the maximum turbidity in the set of pumped samples, extrapolation always uses the regression based on all samples. This method can be used to model hysteresis, but performs poorly unless the loop is smooth and all pairwise segments have positive slopes. It can invert peaks if these conditions are not met. Therefore it is not commonly used.

flowsrc is exactly analogous to **turbsrc** in its usage, except the predictor of SSC is discharge instead of turbidity, *and the default type is "logxy" instead of "linear"*.

lineartime interpolates SSC linearly between samples. Specify **ssc1** for the starting SSC if there is no sample at the start time and **ssc2** for the ending SSC if there is no sample at the end time. You

can get the start and end SSC of an adjacent modeled segment, by using the **endpoints** function, e.g. **endpoints(tts1)** gives the predicted SSC at the start and end of the period represented by the object **tts1**.

cubictime interpolates SSC using a natural cubic interpolating spline between samples. Otherwise it works just like **lineartime**.

Note: The prediction functions have similar arguments to the scatterplot functions, so once you have settled on an appropriate model, the argument list can often be copied verbatim from a scatterplot function to the corresponding prediction function.

> **args(turbsrc)**

```
function(stn, hy, sdate1, stime1, edate1, etime1, sdate2 = sdate1, stime2 =
  stime1, edate2 = edate1, etime2 = etime1, dumps, bottles, interval = 10,
  exclude = NULL, long = T, adj = F, var = T, type = "linear", bias
    = "mvue", units = "cfs", span = 1, degree = 1)
```

> **args(flowsrc)**

```
function(stn, hy, sdate1, stime1, edate1, etime1, sdate2 = sdate1, stime2 =
  stime1, edate2 = edate1, etime2 = etime1, dumps, bottles, interval = 10,
  exclude = NULL, long = T, adj = F, var = T, type = "linear", bias
    = "mvue", units = "cfs", span = 1, degree = 1)
```

> **args(lineartime)**

```
function(stn, hy, sdate, stime, edate, etime, interval = 10, ssc1 = 0, ssc2 = 0,
  long = T, adj = F, units = "cfs")
```

> **args(cubictime)**

```
function(stn, hy, sdate, stime, edate, etime, interval = 10, ssc1 = 0, ssc2 = 0,
  long = T, adj = F, units = "cfs")
```

Definitions of prediction functions' required arguments

stn	3-letter station name (e.g. " nfc ")
hy	2-digit water year
sdate	start date (<i>yymmdd</i>)
stime	start time (<i>hhmm</i>)
edate	end date (<i>yymmdd</i>)
etime	end time (<i>hhmm</i>)

(If 2 sets of start and end dates are given, the first is the period estimated and the second is the period defining the samples to use in the model)

Definitions of prediction functions' optional arguments

interval	data logger interval in minutes (default=10)
dumps	vector of dump numbers, same length as bottles
bottles	vector of bottle numbers, same length as dumps. Overrides the date/time selection of samples to use in the model

exclude	vector of turbidity codes flagging data to be omitted from prediction model. The default (NULL) omits nothing.
long	logical value (T or F), indicates long output if T , short if F
adj	logical value (T or F), indicates whether to adjust SSC to cross-section mean. If T , then a data frame called discoef must be present. See bottom of this document for a description of discoef .
var	logical value (T or F), indicates whether to compute variance (turbsrc and flowsrc only). If estimating a record of more than a few days using log-transformed regression and bias-correction method " mvue " (see bias argument above), variance computations are very tedious and may appear to hang the computer. To proceed without computing variance, specify var = F , or try using bias = "duan" or bias = "qmle" . There are no variance calculations for type = "sqrt" or type = "loess" in the current implementation. For loess models, I am not aware of an accepted computational method for covariance of predictions.
type	character string: " linear ", " logx ", " logxy ", " sqrt ", " power ", " loess ", or " pairs "
bias	" mvue " (default), " qmle ", or " duan " specifies bias-correction method when type="logxy" . " mvue " is the minimum-variance unbiased estimate, hence is preferred but is computationally more demanding. " qmle " is the more easily computed and well-known quasi-maximum likelihood estimate. " duan " specifies Duan's non-parametric smearing estimate. All three estimates have similar mean squared error, but only MVUE is unbiased. Duan's estimate may be preferred in some instances because it makes no assumptions about normality of residuals. See Gilroy et al. (1990) for brutal details. Duan's smearing correction is the only bias correction option for type="sqrt" , in which case no bias argument is needed.
units	input discharge units. default is " cfs " (ft ³ /sec), alternative is " cumecs " (m ³ /sec). (Output will always be in metric)
ssc1	starting SSC for lineartime and cubictime functions (default = 0)
ssc2	ending SSC for lineartime and cubictime functions (default = 0)
span	when type="loess", this is the smoothing parameter used (positive values usually no greater than 1).
degree	when type="loess", this is the degree of the local regression (1 or 2). For TTS applications, degree = 1 is recommended.

Summary stats: **total**, **tts.tot**, **msc.tot**, and **sum.tot**

total replaces the old functions **tts.tot**, **msc.tot**, and **sum.tot**, but places no restrictions on the names of the objects. The old functions do still work however.

tts.tot may be used when objects are named **tts1**, **tts2**, etc.

msc.tot may be used when objects are named **msc1**, **msc2**, etc.

sum.tot may be used when mixtures of the above objects are present.

Examples of summary stats on objects produced by prediction functions

total(obj1, obj2, obj3) would summarize the objects **obj1**, **obj2**, and **obj3**.

tts.tot(1:2) would summarize objects **tts1** and **tts2**

mssc.tot(1:3) would summarize **mssc1**, **mssc2**, and **mssc3**

sum.tot(tts=1:2, mssc=1:3) would summarize all of the above

sum.tot(mssc=1:3) is equivalent to **mssc.tot(1:3)**

Summary plots: **ttsplot**

```
> args(ttsplot)
function(stn, hy, ..., stagetics = seq(0, 4, 0.1), adj = F, number = T,
        units = "cfs", split = 0.35)
```

Definitions of **ttsplot** required arguments

stn	3-letter station name (e.g. "nfc")
hy	2-digit water year
...	One or more objects created by turbsrc , flowsrc , lineartime , and cubictime , but no more than two objects produced by turbsrc , and no more than three produced by flowsrc , lineartime , and cubictime . Multiple objects are delimited by commas.

Definitions of **ttsplot** optional arguments

adj	logical value (T or F), indicates whether to adjust SSC to cross-section average. If T , looks for an object called discoef containing regression coefficients for each station
number	logical value (T or F) denoting whether to plot sediment samples using sample number (T) or a solid circle (F)
units	input discharge units: " cfs "=ft ³ /sec or " cumecs "=m ³ /sec. Discharge will be converted to metric if input is " cfs ".
split	the proportion of the figure region that will be devoted to the discharge hydrograph.

Functions to save results and export to text file: **tts.ssc** and **write.ssc**

tts.ssc saves an object containing data and time, concentration, and sediment surrogate (turbidity = 1, discharge = 2, time = 3). You should save the result as *stn...ssc* or *stn....ssc*, where *stn* is the 3-letter station name and "." represents any character. For example, the first two dots could represent a storm number (e.g. **cas09.ssc**)

write.ssc searches for all objects named *stn...ssc* or *stn....ssc*, where *stn* is the 3-letter station name and "." represents any character. For example, the first two dots could represent a storm number (e.g. **cas09.ssc**). These objects, presumably created by **tts.ssc**, are then appended together and

sorted. **write.ssc** then looks up the discharge, turbidity and turbidity code in *stnhy.flo*, and merges that information with all the rest. The output file is a text file containing date, time, flow, turbidity, turbidity code, SSC, and method. The method identifies the surrogate used to estimate SSC: 1=turbidity (from **turbsrc**), 2=discharge (from **flowsrc**), 3=time (from **lineartime** or **cubictime**).

```
> args(tts.ssc)
function(...)
```

Required arguments

... One or more objects created by **turbsrc**, **flowsrc**, **lineartime**, and **cubictime**. These are the same objects passed to functions **ttsplot** and **total**. Multiple objects must be delimited by commas.

```
> args(write.ssc)
function (file, stn, hy, path = ".")
```

Required arguments

file Name of output file
stn 3-letter station name
hy 2-digit water year (needed to find **flo** object)

Optional arguments

path location on system where output file is to be written. Default is same location as .RData

Cross-section coefficients (**discoef**)

discoef is an optional data frame that must be present in the working directory if you want to have concentrations adjusted to a cross-sectional mean value. The coefficients are presumably from a regression of point SSC versus cross-sectionally and vertically integrated SSC. The format of **discoef** is as follows:

	n	a	b	max
ARF	0	0.0000	1.0000	NA
CAR	39	0.5313	0.9268	450
DOL	53	0.2547	0.9592	2000
EAG	47	0.3101	0.9645	2000
FTR	0	0.0000	1.0000	NA
HEN	46	0.2663	0.9819	900
IVE	0	0.0000	1.0000	NA
MUN	0	0.0000	1.0000	NA
NFC	55	0.2624	0.9424	800

The first column contains the row names of the data frame and must consist of the first 2 characters of each station name. The column names are **n**, **a**, and **b**. The **n** column contains the number of data points in the regression and is not required. The **a** and **b** columns are the cross-section

coefficients. **max** should normally be the highest point SSC represented in the regression. For point SSC less than **max**, the adjustment is computed as follows:

$$y = e^a x^b$$

where y is the adjusted SSC and x is the laboratory SSC. The constant **a** is assumed to include a bias correction, if needed (i.e. if the coefficients were computed by log-log regression, $\ln(y) = \mathbf{b}_0 + \mathbf{b}_1 \ln(x)$, then $\mathbf{a} = \mathbf{b}_0 + 0.5*s^2$ and $\mathbf{b} = \mathbf{b}_1$, where s is the residual standard error from the regression).

For point SSC higher than **max**, extrapolation error may become large as the predicted y diverges from x . Therefore, for extrapolation above **max**, the adjustment is computed as:

$$y = (e^a x) \mathbf{max}^{b-1}$$

This modification changes the regression line to a slope of 1, parallel to the line $y=x$, above the point where $x = \mathbf{max}$. The modification for extrapolations has been implemented for load estimation under R 2.2.0, but has *not* been implemented in procedures for R 1.3.0.

Customization

It may be useful to customize the default file locations in **read.flo**, **read.lab**, and **write.ssc**. To do so, run **fix(funcname)** at the R command line, where *funcname* is the name of one of the functions; then modify the argument list in NotePad, save and exit. Finally, save workspace from the R file menu, or quit and save when prompted.

Add your discharge rating equations to the function called **qcalc**. This is not critical. Its only use is in placing stage tick marks on the right-hand axis of the hydrograph produced by **ttsplot**. Modify the existing function using **fix(qcalc)**. Save and exit NotePad, then save workspace.

REFERENCES

Baskerville, G. L. (1972). "Use of logarithmic regression in the estimation of plant biomass." Canadian Journal Forestry 2: 49-53.

Bates, D.M. and Watts, D.G. (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley.

Cleveland, W. S. (1979). "Robust locally weighted regression and smoothing scatterplots." Journal of the American Statistical Association 74(368): 829-836.

Cleveland, W. S., S. J. Devlin, et al. (1988). "Regression by local fitting: methods, properties, and computational algorithms." Journal of Econometrics 37: 87-114.

- Cohn, T. A., L. L. DeLong, et al. (1989). "Estimating constituent loads." *Water Resources Research* 25(5): 937-942.
- Duan, N. (1983). "Smearing estimate: a nonparametric retransformation method." *Journal of the American Statistical Association* 383(78): 605-610.
- Gilroy, E. J., R. M. Hirsch, et al. (1990). "Mean square error of regression-based constituent transport estimates." *Water Resources Research* 26(9): 2069-2077.
- Hurvich, C. M., J. S. Simonoff, et al. (1998). "Smoothing parameter selection in nonparametric regression using an improved Akaike Information Criterion". *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 60(2): 271-293.
- Koch, R. W. and G. M. Smillie (1986). "Bias in hydrologic prediction using log-transformed regression models." *Water Resources Research* 22(5): 717-723.
- Miller, D. M. (1984). "Reducing transformation bias in curve fitting." *American Statistician* 38: 124-126.
- Rice, J. (1994). *Mathematical Statistics and Data Analysis*. 2nd ed. Duxbury.