**The base flow model**

Let’s start with the idea used in the WRTDS model and then build from there. In WRTDS the he concept is that concentration (in this case of total nitrogen (TN)) is a function of three factors: Time (expressed in years), Discharge (that is, daily mean discharge on the day the sample was collected), and Season (expressed as a sine and cosine of time of year). Furthermore, it is understood that the function that relates these factors to concentration is gradually changing over time and thus we use windows and weighting functions to estimate the model repeatedly at any point in the space defined by time and discharge so that the coefficients of the model change continuously over the space.

The “base flow model” (BFM) introduced here follows a very similar approach to WRTDS but the set of variables that are used in the model is different. It still uses time and season just as in WRTDS, but in place of discharge it uses two discharge-related explanatory variables. The first of these is Qb, which is the base flow discharge for the day. This value comes from the output of an estimation scheme applied by Jeff Raffensperger. I actually don’t know any of the details of its computation although my general impression is that it rises very quickly with each high flow event and very quickly after the peak the estimated base flow becomes a very large fraction of total flow. I do wonder if this base flow estimate is a good one for this application.

The idea in the BFM is that there is a particular chemical signal for base flow water that changes gradually over the range of Qb values as well as a function of time and season. The other explanatory variable is Qp, which is defined as the percentage of the total discharge for the day that is base flow. Given that definition we know that Qp must fall within a range of values from close to zero (base flow can never be zero) to a maximum of 100 (signifying a day when all of the discharge is base flow).

Before introducing the equation that is used let me cover some underlying constraints to the formulation.

1. This model can only be applied to perennial streams. If discharge were to go to zero Qp would be undefined and, because we are going to take logarithms of Qb it would also be undefined.
2. There must be a set of Qp and Qb values for every day in the period for record. In this context “period of record” means the period for which the estimation is going to be done. There might be considerably more years of discharge information.
3. All of the concentration values in the data set lie within the period of record. In addition, the period of record does not extend very far beyond the time span of the concentration data set. For example we wouldn’t want a period of record that starts more than two years before the first sample and we wouldn’t want it to end more than two years after the last sample.
4. Concentrations are reported without censoring and are always greater than zero. This is a requirement that is imposed by the particular formulation used here, which uses weighted linear regression. Linear regression doesn’t allow for censored values and because the dependent variable, which is ln(C) (where C is concentration), zero values are impossible. This constraint could be removed if one moved to survival regression (used in WRTDS). But for now, the model does not accept censored values. If there is a very small amount of censoring in the data set, the censored values can be re-coded to be a value that is the average of the upper and lower bound on the censoring interval (usually this is half the reporting limit). That has been coded into the software provided here.

The estimates of concentration are determined as follows:

where:

*t* = time, in decimal years

*Qb* = base flow discharge, in m3/s

(a value between 0 and 100)

Note that takes on a different value for any location in the model space (as defined by the set of explanatory variables). Thus, there is no assumption of homoscedasticity in the formulation.

The model (consisting of the five beta values and ) is estimated using weighted linear regression. For any location in the model space the weights on the individual observations in the data set are determined by their distance from the estimation point (which is defined by its unique set of values of t, ln(Qb) and Qp ). The weight for each data point is computed as the product of 4 weights which are measured in the following dimensions: time, ln(Qb), Qp, and season. Season here has dimensions of time, but it measures the distance in the time of year between the estimation point and the observation, such that if they were both on the same day of the year (but in different years) the distance would be 0 years, but if the estimation point and the observation where half a year apart in time of year (say 2011-01-01 and 2011-07-01) then the distance would be 0.5 years. The weighting functions for each of the four distance-metrics is a tri-cubed weight function (the same function used in WRTDS). Their half-window widths are set by the user. If the distance in any one of these dimensions is greater than the half-window width then the weight assigned to that observation is zero. The code is designed so that it will only estimate the model when there are at least 100 observations that have a non-zero weight. But, because we need to make an estimate everywhere in the space the code incrementally widens all of the windows (except for the season window) until it is able to assign non-zero weights to at least 100 observations.

Thus far I’ve done a modest amount of experimentation with the choice of half-window widths. The defaults have been set to these values:

windowY (time window) is 7 (in units of years)

windowQb (window for ln(Qb)) is 1 (log discharge units)

windowQp (window for Qp) is 30 (in percentage units)

windowS (window for season) is 0.5 (in units of years)

Experimentation to date suggests that some reasonable values for windowY are in the range of 7 to 10 years, for windowQb the range might be 1 to 2, for windowQp the range might be 30 to 40, and for windowS I feel comfortable with the value of 0.5. The subject of “optimizing” these windows needs work. The software provided can be used to help with that because it computes a root mean square error (RMSE) for the model and the window widths can be changed and that RMSE recalculated.

Using the approach described here there are several quantities can be computed for each day in the period of record. In the following section the “hat” notation is eliminated for simplicity, but in all cases these are estimates and not measurements. Note that the units of the fluxes are kg/day, concentrations are mg/L, and discharges in m3/day. This means that the unit conversion factor is 86.4.

We can compute an estimate of total flux (Ft) for any given day.

We can also estimate a base-flow concentration (Cb) by selecting the estimation point for the day to have a Qp value of 100 (regardless of what the true Qp value was for that day), but the Qb value for that day remains at its true value. All of the other explanatory values at that estimation point are set to their true values for the day. Doing this will cause the beta values and the value to all be different from what they would be for that day with its explanatory variable values set to the true values. Having computed the base-flow concentration for the day we can compute a base flow flux (Fb) for the day.

One final note about this approach: It has the very desirable feature of extending the WRTDS concept to improve on the estimation of concentration and flux by adding information on the past history of discharge to the estimation process. It has been shown in various studies that nutrient concentrations can vary on the basis of the recent history of discharge. Typically if conditions have been relatively dry, then concentrations on a given day will be higher than would be the case if conditions had been relatively wet (for the same value of discharge on the current day). From the two test cases run the RMSE of the BFM is substantially lower than the RMSE for the WRTDS model. Of course some of that is simply due to having one more adjustable parameter, but it would appear that it is a difference that arises for realistic hydrologic reasons. On the other hand, at least superficially, the time series of estimated concentration values derived from WRTDS versus those derived from the BFM don’t look greatly different from each other.

**Working with the software**

Three scripts are provided. These are described below. There are also two functions that are used by one or more of the scripts (they are called “seasonSummary.R” and “runBaseReg.R”). The scripts depend on the EGRET software package as well as dataRetrieval and also the “lubridate” package (handy for handling various formulations of date information). These must be installed on the computer before running the software. All are available on CRAN. Here’s some information about the three scripts. It is best to run them in RStudio rather than the basic R interface, because RStudio allows for the saving of multiple graphs. Alternatively, they can be run in batch mode or on the console with the graphics written to a pdf or png file (but a small amount of additional coding is needed to do that).

1. A script called “FitScript.Potomac.R” ingests the necessary data files and estimates the BFM and turns out a variety of diagnostic information about the model and also compares it with WRTDS. (Information about the data sets is provided after the description of the three scripts). It allows the user to set window widths different from those specified as the defaults. The current version of the script has the names of the individual data files imbedded in it, so that in future applications those parts of the scripts would need to be rewritten to reflect the actual data files and the file structure on the computer being used. This script must be run before the others, because it saves an RData file that is needed for the subsequent functions. To see an example of the output, look at the file “Output from FitScript.Potomac.docx”.
2. A script called “ViewRecord.R” which produces example graphics showing how the BFM and WRTDS would depict the concentration time series for a given segment of the period of record and shows these daily estimated concentration traces along with the actual concentration values in the data set. This is interactive, and allows the user to pick out particular time periods to examine or to modify the windows. Note that if you try to view the entire record of 20 or 30 years, the structure will be hard to discern. You need to view a smaller part to see what is happening. There is no need to modify this script for different sites. But, for any given site it needs to be run after FitScript.xxxx.R so that the workspace is loaded. If you want to go back to a previously run example you can always load the workspace (e.g. PotomacCB.TN.RData) that was saved by FitScript. ViewRecord.R also produces a graph of total discharge and base flow discharge for the same time period. This is helpful in seeing how base flow responds to changes in total discharge. To see an example of the output see “Output from ViewRecord.docx”.
3. A script called “ComputeRecord.R” which computes a daily record for the entire data set and provides several forms of output. (Note that this is the one script that takes a significant amount of computer time. Something like 2 minutes, so be patient.) At a daily time step it provides concentrations and fluxes for each of three models (the BFM estimates, the BFM estimates with all discharge being equal to the daily baseflow, and WRTDS). These are stored in a data frame called allDaily that will end up being store in the file with a name like “PotomacCB.TN.RData”. The next form of output are seasonal summaries of that information. They are in a data frame called allResult and they reside in the same workspace but also in a smaller one with a name like “PotomacCB.TN.allResult.RData”. The results are the same six values mentioned above but the concentration values are averages for the season and the Flux values are total flux for the season (in kg). This same data set is also output to a csv file called something like “PotoamacCB.TN.allResult.csv”. In that data frame the column called “DecYear” is the mean value of DecYear for the particular season of the particular year. The column called season is just a counter of seasons with 1 = JFM, 2 = AMJ, 3 = JAS, and 4 = OND. For a dynamic SPARROW model that is attempting to model “base flow flux” the column of interest here is the one called BFMbfFlux. The values are totals, in kg, for the season. If there was an interest in using the flux associated with the “quick flow” (the part of discharge above the base flow) then this would be BFMFlux – BFMbfFlux.

**Data files required**

This example is for a model of total nitrogen for the Potomac River at Chain Bridge, Washington, DC. It assumes that we have two things to start with.

The first is a standard EGRET workspace, which contains the named list eList. eList has 4 component objects 1. is INFO which is all meta-data, 2. is Daily which is a data frame of daily discharges and daily results, 3. is Sample which is a data frame of sample values and various outputs for the sampled days, and 4. is a matrix called surfaces which contains the standard WRTDS model. All of this is described in the EGRET User Guide. If it doesn't exist, the user will need to create it using the instructions in the User Guide. Note that discharge is always in cubic meters per second. See note below about trimming the two data sets to the same start and end dates.

The second is a file that is structured as a .csv file that has five columns: station number, date, a total discharge, a base flow discharge (both of these are in cfs) and finally a percent of total flow that is base flow (a number between 0 and 100). You may need to format your date column to m/d/yyyy.

The two data sets need to be “trimmed” prior to running so that they cover the exact same time period. Here are the steps involved in the trimming process.

First load the EGRET workspace (in this example it is called PotomacCB.TN.RData). Once loaded, give the command summary(eList$Daily) and also summary(eList$Sample) take note of the first and last dates of the eList$Daily. Make sure that the first and last dates of eList$Sample do not lie outside the range of the dates for eList$Daily (this is a basic requirement for running EGRET).

Next, in Excel open up the daily discharge and base flow file (in this case it is flowsep.ChBridge.csv) and take note of the starting and ending dates for that record. You need to pick a start and end date for your analysis that is entirely within the range of the dates you see in these two files. Stated another way: both files need to start on the same date and end on the same date.

Let's suppose that the EGRET eList needs to be trimmed to fit within the flow separation record, and let's assume for this example that the back end of it needs trimming so that it ends with 2013-07-31. This would be the set of steps you would take.

eList$Sample <- subset(eList$Sample,Date<="2013-07-31")

eList$Daily <- subset(eList$Daily,Date<="2013-07-31")

If it needed to be trimmed at the front end the same process would take place there. For example:

eList$Sample <- subset(eList$Sample,Date>="1984-10-01")

eList$Daily <- subset(eList$Daily,Date>="1984-10-01")

When that is done the WRTDS model can be estimated and all the results stored in eList (this will take a couple of minutes of computer time). The command is:

eList <- modelEstimation(eList)

Then, at this point the workspace can be stored again. The command for that is

saveResults(savePath,eList)

Then you turn to the csv file and if it needs to be trimmed at either end, you do so in Excel by just deleting the rows that are before the starting date and after the end date and then store the csv file again. You may need to reformat the date column to m/d/yyyy.