Guide to Water Balance R Code

# Overview

The water-balance repository is a series of R scripts that calibrates and models streamflow at a given USGS stream gage. The model has two parts: first, a simple water balance model is run to generate estimations of runoff, and second, the estimated runoff is used to calculate estimated streamflow using unit hydrograph theory, which approximates the catchment as a configuration of linear storages (cite IHACRES). This second part is known as the IHACRES catchment scale rainfall-streamflow modeling methodology.

Both components of this model require parameterization; the water balance model includes physical 12 parameters defining various characteristics of the watershed, while the IHACRES model includes 5 parameters describing the unit hydrograph. This model calibrates both sets of parameters on the historical time period, using recorded observations of streamflow from a given USGS gage. The initial version of the model existed in spreadsheet form; users could manually adjust each coefficient and evaluate the model’s accuracy through calculated NSE values and plots. Though this code can be run with provided coefficients, such as those produced by a calibrated spreadsheet version of the model, this R code automates the process of finding the optimal coefficients for both components of the model.

Optimization occurs in two steps. First, the optimal water balance variables are identified, using monthly NSE as the variable to optimize the parameters on. Because the water balance variables describe general characteristics of the watershed, they primarily impact streamflow over a longer time scale than daily. Additionally, using a monthly time scale increases the efficiency of the water balance optimization process, which is already complex because it involves 12 variables. The optimization function used for this process is a genetic algorithm, from the R package GA. Genetic algorithms (GAs) are stochastic search algorithms that mimic the processes of evolution and natural selection to determine the optimal solution. Specifically, the GA optimization function was selected for this purpose because it allows the user to place physically relevant constraints on parameter values and it searches the entire parameter space, allowing for a basic sensitivity analysis of the model. The optimization also runs more quickly than a manually coded optimization. Second, the optimization of IHACRES model parameters involves two optimizations. First, a GA optimization function is used to find the approximate values of the parameters. Second, optim() is used to search the parameter space near the approximate parameter values and find the most optimal parameters. In the test case, the adoption of the second optimization increased the NSE by 0.02. optim() is not used alone because it tends to only search locally, which can produce incorrect values if the initial values provided to the function are incorrect and does not allow for exploration of the parameter space. The optimization of the IHACRES model parameters uses the daily NSE as the optimization value. The IHACRES parameters contribute significantly to streamflow at a daily scale and tend to be more sensitive than the water balance variables, which is why two optimization functions are used for the IHACRES variables. Overall, optimization takes about 2? hours.

# Required User Input

The top of the @@ script includes a section that requires user input, titled @@. A range of parameters require user input for the model to run:

* PETMethod: “Oudin” is currently the only PET Method supported, but the WB() function has a switch statement to allow for other PET methods, including “Penman” and “Hamon”, should they eventually be implemented
* optimization: TRUE to run optimization code, replacing user-defined model parameters with optimal parameters. FALSE to run model with user-definedparameters.
* delayStart: TRUE to delay starting year by 1 year from the user-defined value, allowing for an assessment of the calibration accuracy. FALSE to use user-defined value as starting year.
* NonZeroDrainInitCoeff: TRUE to use nonzero initial drainage coefficients that are the average of the modeled historic January Flows after the Cutoff Year. If set to FALSE, initial drainage coefficients are 0. Typically set to FALSE.
* incompleteMonths: FALSE to remove incomplete months from monthly comparison of measured vs modeled flow, TRUE to include incomplete months. Set to TRUE for Excel R verification
* GridMET: TRUE causes model to use GridMET climate data, FALSE causes model to use Daymet climate data
* fillLeapDays: only applicable to Daymet data (which does not contain leap days; GridMET and USGS gage data do contain leap days). TRUE fills in missing Daymet leap days with previous days' data. FALSE removes leap days from USGS streamflow data.
* future\_analysis: TRUE runs the IHACRES flow model to generate future streamflow projections. FALSE does not.
* runFutureWB: TRUE to re-run entire water balance model for future; FALSE to use pre-existing water balance projections from a Mike Tercek spreadsheet. If Mike Tercek spreadsheet doesn’t exist in Data folder, this must be TRUE for future\_analysis to work.
* userSetJTemp: TRUE for the user-defined Jennings coefficient and FALSE for it to be automatically extracted from the Jennings tif based on the latitude and longitude of the watershed location. Typically set to FALSE
* make\_plots: TRUE generates plots that are stored in the Output folder. FALSE does not.
* provide\_coords: if TRUE, user provides lat/lon coords. if FALSE, lat/long coords are pulled from centroid of watershed with given gage id
* flow\_components: change the number of components that characterize the IHACRES flow. can be 2 or 3. 2: flow has quick and slow components; 3: flow has quick, slow, and very slow components.
* FolderName: name of folder created to store output from this run (including plots and results files).
* SiteID: name of location for analysis, used to make plots and folders to store output. Keep it simple. For example, “Redwood Creek”.
* GageSiteID: USGS gage number. [This website](https://www.usgs.gov/streamstats/getting-information-streamgages) provides guidance for finding the gage number on the USGS [StreamStats](https://streamstats.usgs.gov/ss/) map.
* startY, startM, startD: start year, month, and date, in the form 2025-01-01. The GridMET dataset begins on 1979-01-01 and the Daymet dataset begins on 1980-01-01, so these are the recommended respective start dates. This date does not need to match the start date of the USGS gage data; this will be adjusted for in the code.
* endY, endM, endD: end year, month, and date, in the form 2025-01-01. The GridMET dataset extends until yesterday and the Daymet dataset extends until the end of the most recent calendar year.

Optional parameters:

* Scaling factors for GridMET climate data (tmmx\_slope, tmmx\_bias, tmmn\_slope; tmmn\_bias, p\_slope, p\_bias): a linear regression model at a daily time step was used to determine scaling coefficients for tmin, tmax, and precipitation, based on comparisons between weather station data and GridMET data at a co-located gridcell. If this analysis has not been carried out, set slopes to 1 and biases to 0.
* Initial model parameters:
* Model parameter optimization upper and lower limits
* individual\_models: provide list of model names (GCM and RCP) in the format 'BNU-ESM.rcp45'; plots of future streamflow ensembles will highlight these individual model projections
* Latitude and longitude: if provide\_coords is TRUE, provide latitude and longitude corresponding to the location that climate data will be pulled for. This is most often the centroid of the watershed (as opposed to the location of the stream gage, which may not be representative of conditions across the entire watershed).

# Running with pre-provided variables

# Optimization

# Future Streamflow Projection

# Comparison with spreadsheet