Course Notes Advanced SWAT: creating SWAT-CUP input simplified

Willem Vervoort

02-11-2017

Introduction

This is an introduction into developing SWAT CUP inout files using R, written for the "How do I use satellite and global reanalysis data for hydrological simulations in SWAT?" workshop in Montevideo between 7 - 11 August 2017, jointly organised by the University of Sydney, IRI (the University of Columbia) and INIA, Uruguay.

Functions to prepare SWAT-CUP files for the calibration and validation routine

This file demonstrates the use of three auxillary functions to create input data for SWAT calibration in SWAT-CUP, which can be called by using:

```r
source("functions/SWATCUPfunctions.R")
```

There are 3 functions.

- `swatcup_MFformat()` a function hat generates observed.txt and observed_rch.txt files depending on the inputs given for a list of data frames with either a single flow station, or with multiple flow stations.

- `MODIS_ts()` a function to transform the MODIS data to a timeseries for the different pixels in the MODIS image.

- `swatcup_ETformat()` a function that generates observed.txt and observed_rch.txt files depending on the inputs given for either a single flow station or a combination of a single flow station and ET data.

swatcup_MFformat

This function encapsulates a few separate functions and uses them to write the input files for SWAT-CUP in the right format for multiple flow data. It can write:

- observed.txt
- observed_rch.txt

This function takes the following input

- df_flow, this is a list of data frames with flow data, which have two columns: "Date" and "Flow".
- date.format is a definition of the date format in case the date format in the flow data is incorrect
- st.date: the starting date for the output
- end.date: the end date for the output
- outfile: the SWAT-CUP file you want to write
- infile: the SWAT-CUP file you use as template
- nlines: the number of lines in the header, this varies depending on the file you are trying to write and whether your project is a SUFI2 or PSO project.
- weight: a single number or a vector indicating the weight of the flow data relative to the other input data. The objective function weights will be $\sum weight/np * Obs_i$

Before you can run this code, it is wise to make a copy of your "observed.txt"/"observed_rch.txt" file from the SUFI2.in or PSO.in folder and put this in the working directory. This means that if things go wrong you still have a original file available.

nlines input
Note particulary the input `nlines`, which is a different number depending on whether you use the PSO or SUFI2 version.

```r
# Demonstrate: writing files for 2008 - 2011
# read in flow data
flow_df <-
  readRDS(file = "data/SantaLuciaFlow/AllSantaLuciaFlowdata.RDS")
# original Q is in m^3/sec
# SWAT also needs cumecs
# write observed.txt
# FOR SUFI2 use nlines = 16, for PSO use nlines = 14
# below is for SUFI2
swatcup_MFformat(df_flow = flow_df,
                 date.format = "%Y-%m-%d",
                 "2008-01-01", "2011-12-31",
                 outfile = "data/observed_MF.txt" ,
                 infile = "data/observed.txt", nlines = 16,
                 weight = 0.2)

# write observed_rch.txt
swatcup_MFformat(df_flow = flow_df,
                 date.format = "%Y-%m-%d",
                 "2008-01-01", "2011-12-31",
                 "data/observed_rch_MF.txt" ,
                 "data/observed_rch.txt", nlines = 6,
                 weight = 0.2)
```

MODIS_ts

This function reads the directory indicated by *MODISdir* and looks for files with the extension pattern given by *patt*. The output is a timeseries of values stacked for all the points that are available. There are 5 columns in the output:

- Year
- JDay (Julian Day)
- value (of actual ET in mm)
- Point (a number in the catchment)
- Date (the actual date in Y-m-d)

```
# demonstrate
# Create a single file with all the MODIS ET data for all points
ET_Data <- MODIS_ts("MODIS/SantaLucia")
# show the data
head(ET_Data)
```

```
##   Year JDay   ET Point       Date
## 1 2000    1 16.2     1 2000-01-01
## 2 2000    9 22.0     1 2000-01-09
## 3 2000   17 19.6     1 2000-01-17
## 4 2000   25 25.1     1 2000-01-25
## 5 2000   33 26.0     1 2000-02-02
## 6 2000   41 24.7     1 2000-02-10
```

swatcup_ETformat

This function encapsulates a few separate functions and uses them to write the input files for SWAT-CUP in the right format for both ET and flow data. It can write:

- observed.txt
- observed_sub.txt
- observed_rch.txt

This function takes the following input

- df, this is a data frame with flow data or ET data, the output of `MODIS_ts()`
- df_flow this is an optional dataframe with flow data if df is ET data
- date.format is a definition of the date format in case the date format in the flow data is incorrect
- st.date: the starting date for the output
- end.date: the end date for the output
- outfile: the SWAT-CUP file you want to write
- infile: the SWAT-CUP file you use as template
- nlines: the number of lines in the header
- Flow: a boolean indicating whether or not flow data is included
- weight: a single number or a vector indicating the weight of the flow data relative to the other input data. The objective function weights will be $weight * flow + \sum weight/np * Obs_i$

Before you can run this code, it is wise to make a copy of your "observed.txt"/"observed_rch.txt/observed_sub.txt" file from the SUFI2.in or PSO.in folder and put this in the working directory. This means that if things go wrong you still have a original file available.

nlines input
Note particulary the input `nlines`, which is a different number depending on whether you use the PSO or SUFI2 version.

```
# Demonstrate: writing files for 2008 - 2011
# read in flow data
flowdata <- readRDS(file="data/SantaLuciaflow/SantaLucia.RDS")
head(flowdata)
```

```
##         Date  flow
## 1 1983-03-16 3.964
## 2 1983-03-17 4.786
## 3 1983-03-18 6.226
## 4 1983-03-19 6.307
## 5 1983-03-20 5.788
## 6 1983-03-21 5.607
```

```
# original Q is in m^3/sec
# SWAT needs cumecs

# write observed_sub.txt
swatcup_ETformat(ET_Data, df_flow = NULL, date.format = "%Y-%m-%d",
                             "2008-01-01", "2011-12-31",
                 outfile = "data/observed_sub_ET.txt" ,
                 infile = "data/observed_sub.txt",
                 6, weight= 0.1)



# write observed.txt
# FOR SUFI2 use nlines = 16, for PSO use nlines = 14
# below is for SUFI2
swatcup_ETformat(ET_Data, df_flow = flowdata,
                 date.format = "%Y-%m-%d",
                 "2008-01-01", "2011-12-31",
                 outfile = "data/observed_ET.txt" ,
                 infile = "data/observed.txt",
                 nlines = 16, Flow = TRUE,
                 weight = 0.1)

# write observed_rch.txt
swatcup_ETformat(flowdata,df_flow=NULL,
                 date.format = "%Y-%m-%d",
                 "2008-01-01", "2011-12-31",
                 outfile = "data/observed_rch_ET.txt" ,
                 infile = "data/observed_rch.txt", nlines = 6,
                 Flow = TRUE)

# Now test putting in weights relative to the size of the subcatchment
subbasin_data <- read.csv("data/subbasins_SantaLucia_alldata.csv")

# calculate weights from relative areas
f_w <- 0.1 # flow weight
ET_w <- subbasin_data$Area/sum(subbasin_data$Area)*(1-f_w)
 w_in <- c(f_w, ET_w)

 # now try to write the file
# FOR SUFI2 use nlines = 16, for PSO use nlines = 14
# below is for SUFI2
swatcup_ETformat(ET_Data, df_flow = flowdata,
                  date.format = "%Y-%m-%d",
                  "2008-01-01", "2011-12-31",
                  "data/observed.txt" ,
                 "data/observed.txt", nlines = 16,
                 Flow = TRUE, weight = w_in)
```