ABSTRACT: The Soil and Water Assessment Tool (SWAT) model (Arnold et al., 1998) is a popular watershed management tool. Currently, the SWAT model, actively supported by the U.S. Department of Agriculture and Texas A&M, operates only on Microsoft® Windows, which hinders modelers that use other operating systems (OS). This technical note introduces the Comprehensive R Archive Network (CRAN) distributed “SWATmodel” package which allows SWAT 2005 and 2012 to be widely distributed and run as a linear model-like function on multiple OS and processor platforms. This allows researchers anywhere in the world using virtually any OS to run SWAT. In addition to simplifying the use of SWAT across computational platforms, the SWATmodel package allows SWAT modelers to utilize the analytical capabilities, statistical libraries, modeling tools, and programming flexibility inherent to R. The software allows watershed modelers to develop a simple hydrological watershed model conceptualization of the SWAT model and to obtain a first approximation of the minimum expected results a more complicated model should deliver. As a proof of concept, we test the SWAT model by initializing and calibrating 314 U.S. Geological Survey stream gages in the Chesapeake Bay watershed and present the results.

(KEY TERMS: hydrological model; CRAN; R; R-Project; SWAT; SWATmodel; EcoHydRology.)


INTRODUCTION

The Soil and Water Assessment Tool (SWAT) model (Arnold et al., 1998), which is in the public domain, is commonly used by scientists and natural resource managers to predict the effect of Best Management Practices on water, sediment, nutrient, and pesticide yields from agricultural watersheds. The model is used by many planning and regulatory agencies both in the United States and abroad. Currently, SWAT is only actively supported for Microsoft® Windows OS. Also, the model is developed using Fortran, which is currently taught less frequently in college curriculum than higher level languages. This contributes to a gap between “new” research and modeling, as new research increasingly relies more heavily on higher level programming languages than on “legacy” models, which are often developed in Fortran or C. The open source, relatively high-level R-language (RCD Team, 2012) may provide a bridge that allows modelers around the world to easily access and run “legacy” models, like SWAT, and share “new” libraries, packages, data,
code, and model output that can help to further the field of watershed modeling.

R is an open source, freely available computational system, initially developed as a statistical analysis tool to serve as an alternative to the commercially available statistical analysis software, S. R has evolved into a sophisticated programming language supporting object-oriented programming. As a computational tool, the performance of R is comparable to MATLAB; in statistical functioning, it resembles widely used software like S and SAS. R also offers excellent language interoperability and can, through various extensions, access codes written in C, C++, Fortran 77, Fortran 9x, Objective C, Objective C++, Java, MATLAB, and others. The platform flexibility fosters multiple model applications (e.g., TOPMODEL; Buytaert, 2011). The increasing frequency with which R appears in peer-reviewed literature indicates that it is gaining significant use by the scientific community. Because it is used by so many different disciplines that regularly contribute to the Comprehensive R Archive Network (CRAN), R may uniquely facilitate interdisciplinary collaborations among scientists in numerous fields (e.g., socioeconomic, political, hydrological, and biological sciences), thus encouraging cross-fertilization of ideas among the traditionally disparate disciplines.

The SWATmodel package we developed provides a linear model-like R interface to the SWAT modeling system, transforming weather data through a multiparameter modeling space into a hydrological output response. A valuable feature of R analysis packages is their ability to work on most OS and system architectures. SWATmodel contains the public domain SWAT 2005 and 2012 Fortran code, slightly modified to be GNU (a recursive acronym for GNU’s not Unix free software project) licensed, multi-architecture, and Fortran compiler compliant. This way CRAN can confirm compliance, compile binaries, and distribute the SWATModel for most OS.

As a proof of concept we utilize this new software to develop initial SWAT models for 314 U.S. Geological Survey (USGS) stream gages located across the Chesapeake Bay watershed. Models were initialized and then autocalibrated against USGS measured data using the DEoptim algorithm (Ardia and Mullen, 2009). Installation of R on any platform is as simple as downloading the software and running the installation program. Available from http://www.r-project.org/.

> install.packages("SWATmodel")
> library(SWATmodel)

# Loads the needed packages and libraries into the workspace
> flowgage_id='01540500'

# Creates a directory named '01540500', which is the USGS flowgage ID
> flowgage=get_usgs_gage(flowgage_id)

# This is a function which links to the USGS WaterWatch website and finds the USGS flowgage '01540500'

In our demonstration, we use functions in R to download and parse streamflow and elevation data from the USGS WaterWatch website. For meteorological data (min/max temperature and precipitation, humidity, solar radiation, wind) we use the Climate Forecast Systems Reanalysis (CFSR) dataset. Recently, Fuka et al. (2013c) demonstrated that CFSR data do as well or better at representing the weather occurring over a watershed than National Climatic Data Center (NCDC) weather stations, particularly when the NCDC stations are more than 10 km from the watershed. Note, however, that we are not advocating any one weather data source, and testing various weather data is not our focus here, rather we are demonstrating an efficient watershed model initialization procedure, and the CFSR data are formatted specifically for this task. Both of the functions to download and parse the USGS gage data as well as the CFSR data are available from the CRAN EcoHydRology package (Fuka et al., 2013b), which is a dependency in SWATmodel and automatically installed when the SWATmodel package is installed. SWAT parameters are then calibrated using the DEoptim algorithm (Ardia and Mullen, 2009). Installation of R on any platform is as simple as downloading the software and running the installation program. Available from http://www.r-project.org/.

The following is an example of a simple implementation of the SWAT2005 (or 2012) model within an R session that obtains the necessary information from the USGS Water Watch website for stream gage ID: 01540500 to initialize a SWATmodel run, and, using the gage coordinates, obtains the necessary weather variables with which to run the model. We demonstrate the simple steps required to install R on any operating system, install the SWATmodel package, download streamflow and meteorological data, run the model, and visualize the results.

In this example, we also demonstrate the coupling of SWAT legacy code with other models in R, which is fairly common among community models and allows users to assemble models in different ways or create new models that augment or replace existing ones. The first step in our example is to obtain the required data to represent the hydrologic system. In our case, this is the land surface characterization, the meteorological forcing data, and the hydrological response data used for calibration and corroboration. In our demonstration, we use functions in R to download and parse streamflow and elevation data from the USGS WaterWatch website. For meteorological data (min/max temperature and precipitation, humidity, solar radiation, wind) we use the Climate Forecast Systems Reanalysis (CFSR) dataset. Recently, Fuka et al. (2013c) demonstrated that CFSR data do as well or better at representing the weather occurring over a watershed than National Climatic Data Center (NCDC) weather stations, particularly when the NCDC stations are more than 10 km from the watershed. Note, however, that we are not advocating any one weather data source, and testing various weather data is not our focus here, rather we are demonstrating an efficient watershed model initialization procedure, and the CFSR data are formatted specifically for this task. Both of the functions to download and parse the USGS gage data as well as the CFSR data are available from the CRAN EcoHydRology package (Fuka et al., 2013b), which is a dependency in SWATmodel and automatically installed when the SWATmodel package is installed. SWAT parameters are then calibrated using the DEoptim algorithm (Ardia and Mullen, 2009). Installation of R on any platform is as simple as downloading the software and running the installation program. Available from http://www.r-project.org/.

> install.packages("SWATmodel")
> library(SWATmodel)

# Loads the needed packages and libraries into the workspace
> flowgage_id='01540500'

# Creates a directory named '01540500', which is the USGS flowgage ID
> flowgage=get_usgs_gage(flowgage_id)

# This is a function which links to the USGS WaterWatch website and finds the USGS flowgage '01540500'
> hist_wx = get_cfsr_latlon(flowgage$declat, flowgage$declon)
# This is a function that parses and formats the CFSR data
# for the Lat/Lon of the USGG flowgage
> build_swat_basic(dirname = flowgage_id, iyr = '2006',
  nbyr = 4, wsarea = flowgage$area, elev = flowgage$elev,
  declat = flowgage$declat, declon = flowgage$declon,
  hist_wx = hist_wx)
# This function sets up a simple SWAT model for the USGS
# flowgage, getting the watershed area and elevations to
# build the model and integrates the CFSR forcing data
> data(change_params)
# This displays a list of parameters that are available
# for calibration by SWATmodel
> setup_swatcal(change_params)
# This sets up the calibration
> calib_range = c('1997-12-31', '2005-12-31')
# This tells SWATmodel the range of data to calibrate
# against
> parm_key = paste(change_params$parameter,
  change_params$filetype)
# This tells SWATmodel what to do with the results
> calib_params = change_params[match(select.list(
  parm_key, multiple = T, graphics = F), parm_key),]
1: GW_DELAY 0*.gw 2: ALPHA_BF 0*.gw
3: GWQMN 0*.gw 4: GW_REVAP 0*.gw
5: REVAPMN 0*.gw 6: RCHRG_DP 0*.gw
7: SFTMP basins.bsn 8: SMTMP basins.bsn
9: SMFMX basins.bsn 10: SMFMN basins.bsn
11: TIMP basins.bsn 12: ESCO basins.bsn
13: EPCO basins.bsn 14: SURLAG basins.bsn
15: ADJ_PKR basins.bsn 16: PRF basins.bsn
17: SPCON basins.bsn 18: SPEXP basins.bsn
19: CN2 0*.mgt 20: USLE_P 0*.mgt
21: Depth 0*.sol 22: Bulk Density 0*.sol
23: Ave\._AW 0*.sol 24: Ksat\._est 0*.sol
25: Soil Name 0*.sol 26: Clay 0*.sol
27: Rock Fragments 0*.sol 28: Erosion K 0*.sol
29: OV_N 0*.hru 30: LAT_TTIME 0*.hru
31: LAT_SED 0*.hru 32: ESCO 0*.hru
33: EPCO 0*.hru 34: SLSOIL 0*.hru
35: CH_N2 0*.rte 36: CH_K2 0*.rte
37: CH_EROD 0*.rte 38: CH_COV 0*.rte
39: CH_SAN 0*.rte 40: CH_SIL 0*.rte
41: SUB_KM 0*.sub
Enter one or more numbers separated by spaces, or an
empty line to cancel
1:
# This allows users to select which parameters to
calibrate
1 2 3 4 5 6 7 8 9 10 11 12 13 14 19 21 23 24 32 33
# These are the flow parameters we select
>x = calib_params$xcurrent
# This sets up a vector of the selected calibration
parameters
>swat_objective_function(x, calib_range, calib_params,
  flowgage)
# This tests that the model runs
> outDEoptim <-
  DEoptim(swat_objective_function, calib_params$min,
  calib_params$max, DEoptim.control(NP = 16,
  itermax = 200), calib_range, calib_params, flowgage)
# This starts the calibration using DEoptim
> x = outDEoptim$optim$bestmem
> save(outDEoptim, file = 'flowgage_id')
> file = paste('DEoptim', flowgage_id, sep = '')
# The preceding four lines are used to write out the best
simulation

Figure 1 displays the result of the 314 model ini-
itializations for USGS gages across the Chesapeake
Bay watershed. The mean daily Nash Sutcliffe Effi-
ciency (NSE) (Nash and Sutcliffe, 1970) across all
watersheds was 0.53, but there was a large range in
model performance statistics, from NSE = 0.20 in
basins with impoundments, complex hydraulic struc-
tures, urban, or regulated flow, to an NSE approach-
ing 0.99 in watersheds where there was a limited
record of flow measurements (e.g., less than one
year). It is also interesting to note that model initiali-
zation and full calibration takes less than one hour
for each gage on a laptop with 3.06 GHz Intel Core 2
Duo, and 8 GB 1067 MHz DDR3 SDRAM, and less
than 2.5 min on the National Center for Atmospheric
Research Geyser analysis cluster, making this model-
system a fast and valuable method for a first
approximation of how a more robust (or complicated)
model with more calibration degrees of freedom
should perform. Figure 2 shows the spatial distribu-
tion of the NSE values across the Chesapeake Bay

FIGURE 1. Distribution of Daily Nash Sutcliffe Efficiencies for 314
USGS Stream Gages in the Chesapeake Bay Watershed Initialized
and Calibrated Using the SWATModel Package in R.
watershed. Notice that the best NSE values are in less developed regions of the watershed, where there are fewer reservoirs and urban areas, which complicate the modeling process. Figure 3 shows a typical hydrograph created automatically by the SWATmodel system, and updated for each calibration loop, allowing one to track the calibration progress visually.

It should be noted that the CFSR access function was utilized in this study due to the rapid data access from the EcoHydRology get_cfsr_latlon() function that only requires a latitude and longitude. This is available from the get_usgs_gage() function, and creates a seamless 32-year historical weather dataset formatted specifically for SWAT. When good quality weather data are available, such as from a nearby NCDC Global Summary of Day (GSOD) station, it is important to assess the quality of both the CFSR and NCDC datasets. This is easily accomplished by linking the GSOD function in the EcoHydRology library (get_gsod_stn()) to the SWATmodel package. An example
distance vs. weather representation analysis performed using the SWATmodel package is presented in Fuka et al. (2013a).

CONCLUSIONS

We develop and demonstrate a multi-OS simplified version of the popular watershed model, SWAT implemented in the computing framework, R. This version of SWAT allows modelers to rapidly develop, run, and calibrate a watershed model to obtain a first approximation of watershed parameters. This information can be subsequently used to inform parameter selection and initialization of more complicated watershed models.

SOFTWARE AVAILABILITY

<table>
<thead>
<tr>
<th>Name of software</th>
<th>SWATmodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept</td>
<td>D.R. Fuka, Z.M. Easton</td>
</tr>
<tr>
<td>Programing</td>
<td>D.R. Fuka, Z.M. Easton, J.A. Archibald, M.T. Walter</td>
</tr>
<tr>
<td>Availability</td>
<td>install.packages(&quot;SWATmodel&quot;), All CRAN, <a href="http://cran.r-project.org/">http://cran.r-project.org/</a></td>
</tr>
<tr>
<td>Year first available</td>
<td>2011</td>
</tr>
<tr>
<td>Software required</td>
<td>R</td>
</tr>
</tbody>
</table>

LITERATURE CITED


