



## SWAT+ model setup verification tool: SWATdoctR

Svajunas Plunge<sup>a,b,\*</sup>, Christoph Schürz<sup>c</sup>, Natalja Čerkasova<sup>e,f</sup>, Michael Strauch<sup>c</sup>,  
Mikołaj Piniewski<sup>a</sup>

<sup>a</sup> Warsaw University of Life Sciences, Department of Hydrology, Meteorology and Water Management, Nowoursynowska st. 159, Warsaw, 02-776, Poland

<sup>b</sup> Institute of Water Resources Engineering, Vytautas Magnus University, Universiteto st. 10, Akademija, Kaunas district, LT-53361, Lithuania

<sup>c</sup> Helmholtz Centre for Environmental Research GmbH – UFZ, Department Computational Landscape Ecology, Permoserstraße 15, 04318, Leipzig, Germany

<sup>e</sup> Marine Research Institute, Klaipėda University, Universiteto ave. 17, 92294, Klaipėda, Lithuania

<sup>f</sup> Texas A&M AgriLife Research, Blackland Research and Extension Center, Temple, TX, USA

### ARTICLE INFO

Handling Editor: Daniel P Ames

#### Keywords:

SWAT+  
Verification  
R package  
SWATdoctR

### ABSTRACT

The Soil and Water Assessment Tool (SWAT) is applied worldwide for modeling basin-scale processes. Its latest version (SWAT+) adds new capabilities to the tool, and collectively with increasing computational power and availability of public datasets expands model complexity as well as provides pathways for serious errors in the model setup process. These models are used to run scenarios to support decision-making processes, hence undetected faults can have a substantial socio-economic and environmental impact. We propose a 5-step SWAT+ model setup verification workflow assessing the soundness of processes related to weather, water balance, management, plant growth, point source and tile drain flows. We developed an R package, called SWATdoctR, which guides the user through the model setup verification process, allowing the identification of typical, but repeatedly overlooked, issues. The workflow and the functionality of the tool is demonstrated in 4 SWAT+ setups in different catchments, at various stages of model setup.

### 1. Introduction

The semi-distributed process-based ecohydrologic modeling tool – Soil and Water Assessment Tool (SWAT) has been in use for more than two decades (Arnold et al., 1998). It was successfully applied worldwide to answer water quality and quantity, land use change, environmental impact, water management or hydro-climatic questions (Akoko et al., 2021; Gassman et al., 2007, 2014; Tan et al., 2020). It is the most extensively used river basin model with thousands of application examples (Gassman and Wang, 2015). SWAT Literature Database provides more than 5600 sources to peer-review publications (CARD&ISU, 2023) related to the SWAT model. Moreover, the model has been used to build national scale modeling systems for decision support in water management or agricultural sectors (Plunge et al., 2022; White et al., 2022). To provide a continuation of the SWAT model development, SWAT+, a new model version, brings a complete restructuring of inputs and code with new concepts into a well-established modeling tool (Bieger et al., 2017). For instance, increased flexibility of water routing across landscape was added (Bieger et al., 2019; White et al., 2022). Such enhancement

unlocks previously unavailable potential to represent any water routing objects or phenomena. Another important advancement is the implementation of decision tables (Arnold et al., 2018), which allowed the inclusion of complex land management (see e.g. Nkwasa et al., 2020; Nkwasa et al., 2021) and reservoir management operations in the model (Arnold et al., 2018).

However, increased flexibility and expansion of model setup options increases the potential for users to make errors (with input data, parametrization, conceptualization, setup design, etc.), which are common due to the complexity of processes represented by the SWAT tool (J. G. Arnold et al., 2012). The most widely reported procedure after model setup is the model parametrization and calibration step (Fu et al., 2020). The autocalibration of model parameters is usually computationally expensive, hence it is highly valuable to invest resources in the verification of model setup and input data before proceeding to the model calibration (Hydrology committee of ASABE, 2017).

We adapt the term “verification” from software engineering, where verification is a process of examining design, code, and functionality in order to check if the software has been built according to the

\* Corresponding author. Warsaw University of Life Sciences, Department of Hydrology, Meteorology and Water Management, Nowoursynowska st. 159, Warsaw, 02-776, Poland.

E-mail address: [svajunas.plunge@sggw.edu.pl](mailto:svajunas.plunge@sggw.edu.pl) (S. Plunge).

<https://doi.org/10.1016/j.envsoft.2023.105878>

Received 4 April 2023; Accepted 8 November 2023

Available online 16 November 2023

1364-8152/© 2023 Published by Elsevier Ltd.

requirements (Hydrology committee of ASABE, 2017). The main goal of the verification process is to ensure quality of the application, design, architecture, etc. In the case of SWAT+ model, we perceive model setup verification as a process of checking the soundness of critical input data, feasibility of the underlying process representations, such as water balance and plant growth, quality assurance of the interpretation of anthropogenic influence. Model setup verification should identify any potential inaccurate inputs, which would result in implausible model simulations. This is particularly relevant for complex input datasets such as management operations or decision rules, which may be interpreted by the model in a different way than intended. Verification may only require a few model simulations to identify problems. Unidentified issues in the model setup propagate into the following modeling steps, such as, model parametrization, calibration, validation, and scenario implementation and assessment. In the worst case, fixing the issues would require repeating model setup steps, which are affected by the faulty inputs. Eventually, a verified model setup contributes to a cost-efficient modeling effort, assures the quality of model design and implementation, and increases the confidence of stakeholders in the modeling results.

The importance of model setup verification was recognized by the SWAT model development team with the introduction of SWATCheck (White et al., 2014). This tool is a stand-alone program, which assists users in the identification of critical or fundamental flaws of a model setups by examining model outputs and identifying results which exceed typical or realistic ranges of aggregated simulation outputs and signature measures, such as the runoff ratio. SWATCheck is a part of the SWAT+ Editor program (Texas A&M AgriLife Research and USDA, 2023), which is a standard tool to set up SWAT+ models.

Although the analysis of temporally and spatially aggregated simulation outputs with SWATCheck provides a valuable initial overview of a model setup, it can be a limitation at the same time. Analyses of aggregated variables may average over the issues, such as temporal and spatial changes in simulated or input variables, issues in the scheduling of management operations and, in consequence, the simulated plant growth, or tile drain examination. As SWATCheck is a stand-alone tool which reports the model setup verification via a graphical user interface (GUI), further model troubleshooting is limited. Additional data processing using other software or tools requires substantial manual work and may be a barrier for SWAT+ modelers to adequately investigate their model setups. Integration of SWAT+ model setup verification step into a script programming language (including R) provides valuable support in the model setup process such as support the documentation of reproducible modeling projects and scripted workflows (Chawanda et al., 2020; Coon and Shuai, 2022). Moreover, it allows the utilization of the verification results in further analyses in the used script language and make use of available libraries for visualization, statistical analysis which are available in the respective programming environment.

In this study we develop a model setup verification tool in R programming language to provide SWAT+ users and the larger modeling community with additional capabilities in the identification of potential model application problems. The objectives are as follows: (1) Prepare a workflow for SWAT+ model setup verification; (2) Provide a new tool developed as R package with a set of functions which allow the identification of typical issues in SWAT+ model setups after the model setup and parameterization; (3) Develop easy to use tools for printing and visualizing model inputs and outputs for reporting of model setup verification; (4) Provide demonstration examples of the new tool by applying it in selected case studies.

## 2. Materials and methods

### 2.1. SWAT+ model

The Soil and Water Assessment Tool is a well-known and widely used basin-scale process-based model. Its popularity can be attributed to

several reasons, including versatility (ability to simulate a wide range of land-based and hydrological processes at various sizes), integration (SWAT can be coupled with other models and software), validity (SWAT has been shown to perform well in a variety of watersheds around the world), as well as in-depth documentation (J.G. Arnold et al., 2012; Neitsch et al., 2011) and user support directly from the developers and via user-groups. The model setup can be carried out using GUI tools (QSWAT+ and SWATEditor) or custom build solutions (Čerkasova et al., 2019; Plunge et al., 2023; White et al., 2022). The models' inputs are formatted as ASCII (American Standard Code for Information Interchange) files (text), which present the information about the watershed, underlying data and parameters in a structured way, similar to relational databases. The model outputs are structured as ASCII files as well, which makes model input/output file manipulation (reading – writing) easily accessible and straightforward, and the model itself interoperable and reusable. The SWAT user-community takes advantage of the flexibility by developing various tools for model re-use, and input/output manipulation and analysis.

### 2.2. R package

The scripting languages R (R Foundation, 2023) and Python (Python Software Foundation, 2023) are among the most popular open-source programming languages used by data analysts and data scientists (Bruce et al., 2020). For both R and Python, several libraries (packages) are available to work with SWAT and SWAT+ model setups. For example, SWAT+ AW (Chawanda et al., 2020) provides a SWAT+ model setup workflow in Python. Other libraries to process SWAT and SWAT+ model setups are available from GitHub, such as PySWAT (Ferreira, 2019) and swatpy (Kmoch, 2022) for SWAT2012, or SpotSWATplus (Barrensi, 2022) to integrate SWAT+ with the SPOTpy library (Houska et al., 2015). Two open source R packages are promoted on the official SWAT model website (available at <https://swat.tamu.edu/software/>): SWATplusR (Schürz, 2019) and R-SWAT (Nguyen et al., 2022). These packages can be applied for parallel model run, parameter sensitivity, calibration, validation and uncertainty analysis, as shown in several studies (Maref et al., 2022; Musyoka et al., 2021; Yang et al., 2022). SWATfarmR (Schuerz, 2022) can be utilized to develop agricultural management schedules for SWAT2012 and SWAT+ projects. SWAT-perpR (Plunge, 2023) provides a set of functions for SWAT+ model input data preparation. R packages, such as SWATmodel (Fuka et al., 2014) or R-SWAT-FME (Wu and Liu, 2012, 2014) were developed for running, assessing parameter sensitivity, or model calibration of SWAT2012 or earlier model versions, but are not maintained. An integration of supporting tools in the R environment provides an advantage compared to other, mainly standalone or GUI-based approaches, such as the possibility to take advantage of the large number of free packages (or libraries) developed, provided, and peer-reviewed by the R programming community. Moreover, it adds necessary means to facilitate design of scripted workflows, which helps solving reproducibility, transferability, transparency issues and is useful for data analysis, data visualization and complex statistical analysis tasks.

### 2.3. Model setup verification workflow

We propose a five step workflow for SWAT+ model setup verification, which addresses common issues in model setups (Fig. 1). The first step analyzes the model simulated weather inputs and water balance components. A comparison to literature values and observation data allows the user to verify if the weather input data are interpreted correctly and water balance results are plausible. The steps 2 to 4 focus on the simulation of agricultural management operations and plant growth. Plant growth is a central part of SWAT+ simulation and controls the actual evapotranspiration (ETA), which is a substantial fraction of the hydrological water balance. Step 2 is designed to inspect if agricultural management operations are correctly executed during the

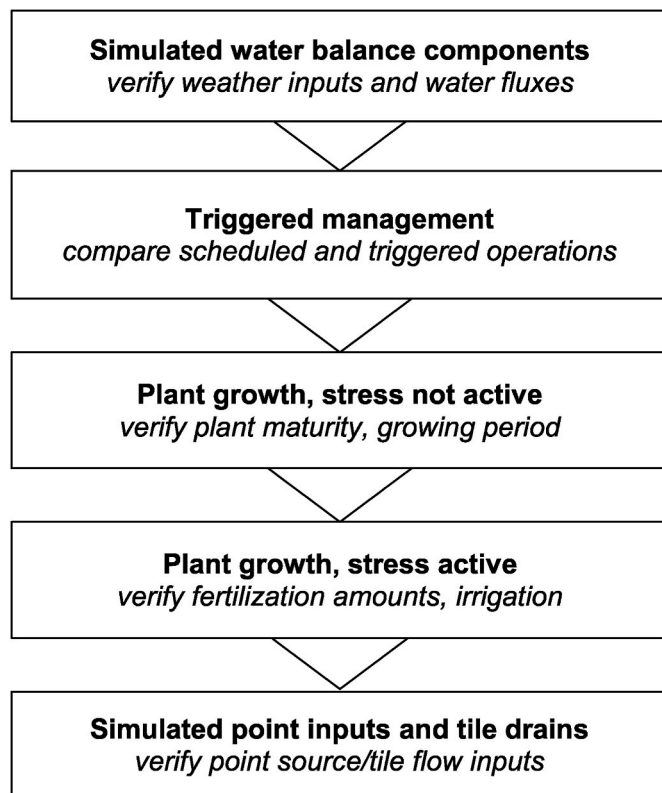


Fig. 1. Proposed workflow for SWAT+ model setup verification using SWATdoctr.

model run. A comparison to the scheduled management operations can identify errors in management inputs. Step 3 is used for the examination of plant growth simulation without activating various growth stresses that can be simulated with SWAT+ (water stress, aeration stress, temperature stress, nitrogen stress, phosphorous stress). Deactivating the stress factors for plant growth in a simulation results in a potential maximum biomass and yields. Low simulated values of biomass or yield identified during this verification step can indicate issues with plant parameters, or management, such as inadequate duration of plant growth. Step 4 is designed to investigate plant growth with the simulation of stress factors activated. A crop specific analysis of stress factors and yields can help to identify issues in the management inputs, such as low fertilizer inputs, or the requirement of irrigation or tile drainage. The final step 5 analyses the inputs into channels from point sources (i.e. wastewater treatment plants (WWTPs), water transfer) and from tile flow of agricultural land objects. step 5 determines if the point source inputs (units, order of magnitude, timing) and the tile drainage inputs (occurrence of tile flow, etc.) were parameterized correctly. The result of the verification process can be generalized and supported by visual analyses of simulation outputs.

To facilitate the 5-step verification process, we developed an open source R package `SWATdoctr`, which provides routines for model diagnostics. The package is currently under active development and testing, and will be applied in at least 14 small agricultural catchments in Europe in the project of “OPTimal strategies to reTAIn and re-use water and nutrients in small agricultural catchments across different soil-climatic regions in Europe” (OPTAIN), the EU-funded research and innovation project. Its functionality will be extended and updated throughout the project.

In this work we present a `SWATdoctr` in the version 0.1.11. The package is freely available on the UFZ GitLab and could be installed with following lines of code.

```
remotes::install_git('https://git.ufz.de/schuerz/
```

```
swatdoctr')
```

```
library(SWATdoctr)
```

The presented workflow for SWAT+ model setup verification can be adjusted to meet the specific project questions. The only requirement to run any of the `SWATdoctr` functions (Fig. 2) is to run SWAT+ model with `run_swat_verification()` function. Any other function from any step could be used independently. `SWATdoctr` package is an additional tool, which is intended to be applied side-by-side with existing `SWATCheck` tool during the model setup verification process.

### 3. Results

#### 3.1. Model setup verification workflow in action

##### 3.1.1. Step 0. Running the SWAT+ model and extracting outputs

After preparing a SWAT model, the first step in the model setup verification is to perform a single simulation and extract relevant simulation outputs. `SWATdoctr` provides the function `run_swat_verification()` to run a simulation of the SWAT+ model setup and to extract the relevant simulation outputs from a set of structured ASCII files (the reader is referred to the current SWAT+ I/O documentation (SWAT model website, n.d.) for a list of all possible outputs and their formats) for any further analyses. The minimum input which must be defined for `run_swat_verification()` is the `project_path` to define where the SWAT+ model setup is located on the local hard drive. An example code to run the function is provided below.

```
sim_nostress <- run_swat_verification(project_path =
'your/path', outputs = c('wb', 'mgt', 'plt'), nostress = 1)
```

Users can activate/deactivate reading of certain outputs. Not all outputs might be relevant to the selected analyses and some outputs can be too large to fit into the RAM of the computer on which the verification is performed. By default `run_swat_verification()` would read all outputs that are used in any of the verification steps. The simulation outputs are defined with the input argument `outputs = c('wb', 'mgt', 'plt')`, where the code `'wb'`, short for “water body”, defines reading `'basin_wb_day.txt'`, `'basin_pw_day.txt'`, `'hru_wb_aa.txt'` and `'recall_yr.txt'` output files. These files are required to analyze climate and water balance variables at the basin or Hydrological Response Unit (HRU) scale. Also this option sets the reading of point source data. The code `'mgt'`, short for “management”, defines reading `'mgt_out.txt'` and `'management.sch'`, which is required to analyze the management operations that were set in the simulation run. Additionally, the option `'mgt'` sets reading of input files such as `'landuse.lum'`, or `'hru-data.hru'` as the HRU properties are in some cases required to be linked with management operations. The code `'plt'`, short for “plant”, defines reading the output file `'hru_pw_day.txt'`, which provides HRU plant growth outputs at daily time steps. A file with daily simulation outputs for all HRUs can be large, if the number of HRUs is big and/or the simulation periods are long. In such cases, it may be necessary to exclude `'hru_pw_day.txt'` when reading the output files.

The setting that controls the plant stress can be manipulated with the input argument `nostress`, i.e., `nostress = 0` activates all stress factors for plant growth in the simulation, `nostress = 1` deactivates all stress factors, and `nostress = 2` deactivates only nutrient stresses. Deactivating plant stresses can be useful for the first three verification steps. If the first check of the climate variables and management were passed, the verification of plant growth without plant stresses can be immediately performed with the same simulation outputs without having to repeat the simulation runs.

Using this tool, SWAT+ simulations are never performed in the original project folder. A copy of the project (model) is generated in the subfolder `'run_verify'`, which is deleted after the simulation. An optional input argument `keep_folder` controls if the simulation folder should be kept and not deleted after the simulation runs. This option can be useful for debugging and checking if `run_swat_verification()` worked as intended.

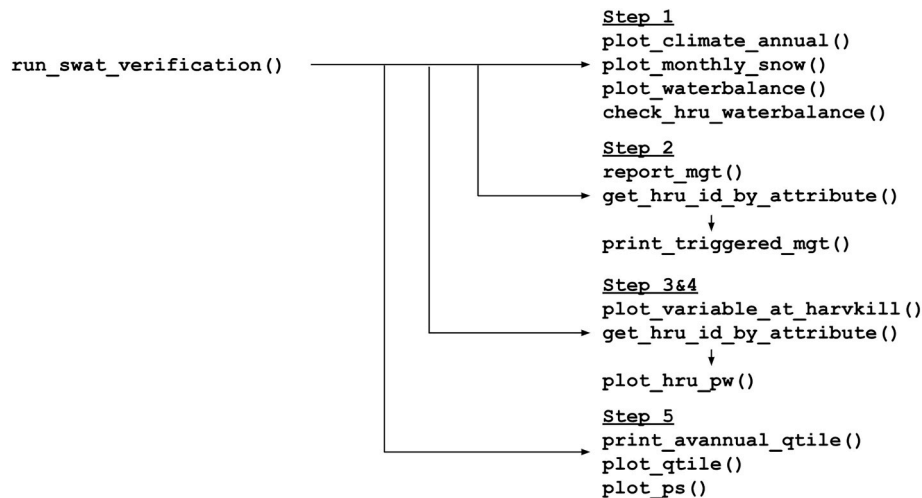


Fig. 2. SWATdoctr v. 0.1.11. package functions map.

By setting the input arguments `start_date`, `end_date`, and `years_skip` the simulation period and the years that are skipped (used for model warm-up) in printing simulated outputs can be controlled. If these input arguments are set to NULL the values which are defined in the model input files `'time.sim'` and `'print.prt'` will be used to define the simulation period.

### 3.1.2. Step 1. Analysis of simulated climate variables and water balance component

Daily precipitation, minimum, maximum temperatures, solar radiation, relative humidity and wind speed variables are used as inputs of a SWAT+ model setup. For different potential evapotranspiration methods (PET) methods, different climate variables are required. There are numerous issues, which could arise:

- If the data structure of the climate input, units of the climate variables, no data flags, etc. were incorrect, it could result in unrealistically small or large values of the climate variables in the simulation.
- The nearest neighbor assignment (weather stations in a model setup are assigned to spatial objects as HRUs, channels, reservoirs, etc.) allocates weather stations to spatial objects where the weather records do not represent the actual weather conditions in a spatial object well (i.e. in complex terrain).
- The selected method for the calculation of PET results in an under/overestimation of PET when compared to estimates of PET for the region. In such cases other methods for the simulation of PET included in SWAT+ should be tested for a better fit to regional conditions and available weather inputs.

An incorrect simulation of climate variables, but also errors in the model parametrization, can lead to issues in the simulation of water balance components. Thus, a first overall analysis of the average annual water balance components can be effective to identify potential issues at an early stage of a modeling study and can indicate necessary steps in a model calibration to result in a more plausible simulation of the water balance.

SWATdoctr provides a set of functions to analyze simulated climate variables and the overall water balance of a model setup. `plot_climate_annual()` (Fig. 3) and `plot_monthly_snow()` (Fig. 4) support the analysis of annual simulated basin averages of weather inputs (and snow processes). While `plot_waterbalance()` and `check_hru_waterbalance()` give an overview of the overall simulated catchment water balance components and report issues with the simulated water balance at the HRU level, respectively. Thus, the two opposite scales complement each other in the identification of potential

issues. The functionality requires the output resulting from `run_swat_verification()` with outputs = 'wb' activated.

```
plot_climate_annual(sim_nostress)
```

`plot_climate_annual()` will generate a multipanel plot with multi-annual distributions of PET, ETa parts, rainfall/snowfall, annual minimum, mean, and maximum temperature values, annual minimum, mean, and maximum relative humidity, annual minimum, mean, and maximum wind speeds, and solar radiation sums. The average values of respective variables for the simulation period are also provided. The comparison of these results to observation data and/or region specific literature values will help to identify implausibilities or discrepancies resulting from the model weather input.

The `plot_monthly_snow()` function can be used to get insight into seasonal dynamics of the precipitation input, which is particularly important in snow-impacted catchments. Verification of snowfall is valuable to analyze how precipitation in solid form is simulated, how snow storage is built-up and how it causes increased spring runoff through snow melt. The plot can be helpful to verify the snow processes and provide guidance to adjust the snow parameters if necessary (Fig. 4).

```
plot_monthly_snow(sim_nostress)
```

`plot_waterbalance()` plots average annual basin values for a wide range of water balance components into a "flow diagram". The arrows visually support the analysis of water fluxes and their connections in the water cycle. In addition to the values of water fluxes and storages a few relevant statistics are printed such as the ET-precipitation ratio (et/precip), the water yield ratio (wyld\_cha/precip), or the base flow ratio (flo/wyld) (Fig. 5).

The function `check_hru_waterbalance()` uses a selection of heuristics with a focus on water balance output variables which were proposed by White et al. (2014). White et al. (2014) defined ranges for average annual water balance variables where simulations outside of these ranges can be implausible and may need further attention. `check_hru_waterbalance()` employs these variable ranges and checks average annual simulated water balance outputs at the HRU level. Variables which can be analyzed are for example the precipitation, different components of evapotranspiration, fast surface runoff, percolation, soil water storage, or the average curve number of an HRU. The function allows to include/exclude checks in the HRU analysis. The function returns a table which flags identified issues for all individual HRUs. Only HRUs are shown for which at least one check identified a potential issue. Function options allow to exclude specific land uses from the analysis which may be flagged due to the land uses properties (e.g. runoff from urban areas). Fig. 6 shows an example output of `check_hru_waterbalance()` which may be helpful to identify

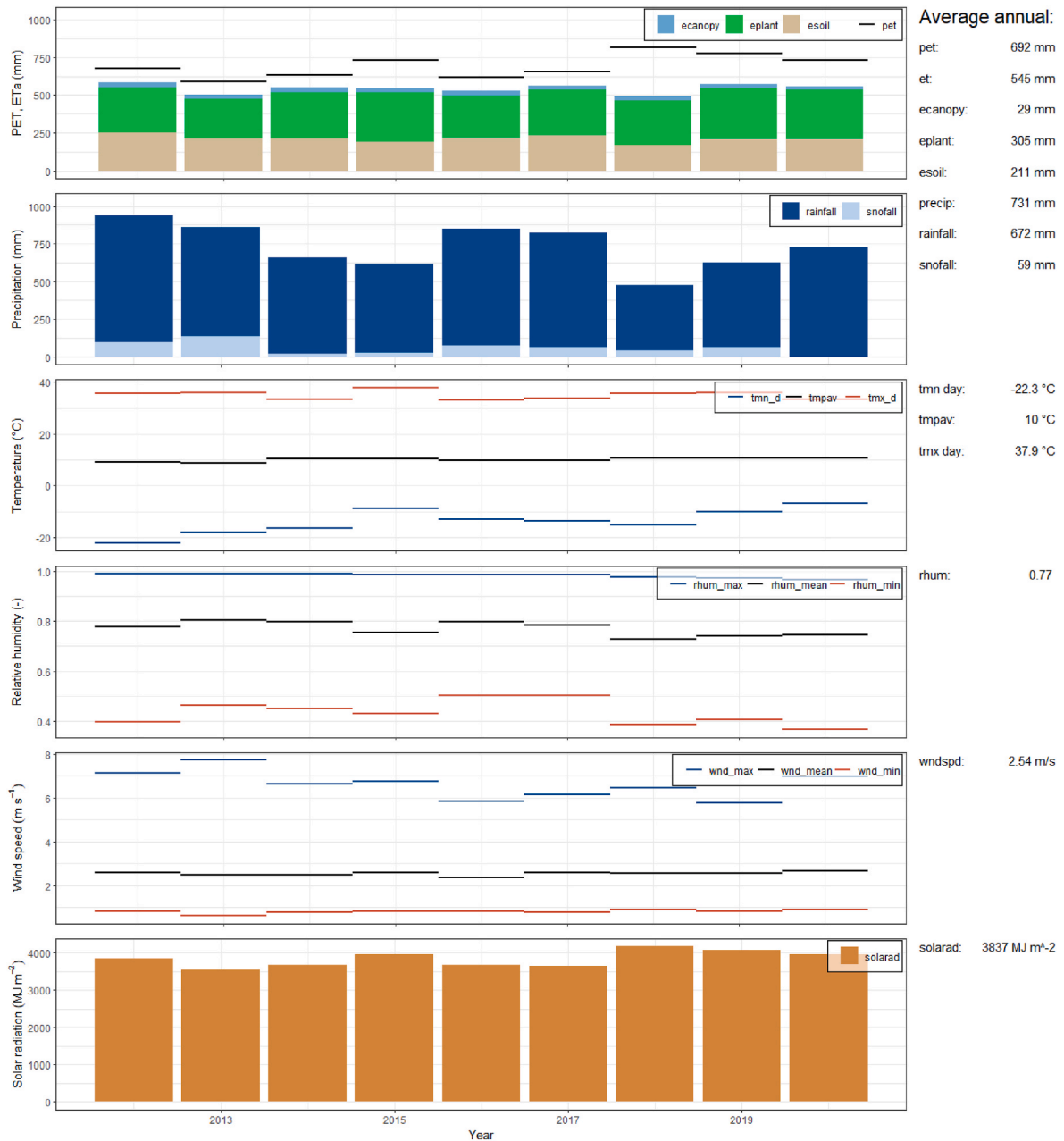


Fig. 3. Example plot of simulated basin climate variables with the function `plot_climate_annual()`.

potential issues in the simulation of land phase processes at the HRU scale.

```
plot_waterbalance(sim_nostress)
```

### 3.1.3. Step 2. Simulation of management operations

Management operation inputs in a SWAT+ model setup can be complex and comprehensive (i.e. could include multiple years, plants, operations, even set to each HRU separately, etc). Scheduled operations point to several other input files which define the parameters of operations or inputs, such as fertilizer or tillage types. Hence, the development of management schedules is highly error prone. Mistakes in management inputs usually do not stop a simulation or produce warnings in the model diagnostics, but lead to skipping certain operations in a simulation run. These circumstances can impede the validation of simulated management schedules and it can become difficult to identify single erroneous lines in the scheduled management operations.

All operations, which are triggered in a SWAT+ simulation run are

written into the `'mgt_out.txt'` output file. To verify the correct triggering of the scheduled operations, a tabular comparison between scheduled and simulated operations is the most robust approach. Yet such a procedure can be cumbersome and only a few HRUs could be used in a comparison each time. In most cases it might be enough to select a few cases for comparison to see if the scheduled operations work properly.

SWATdoctr offers two approaches to investigate management operations in a tabular form. SWATdoctr command `report_mgt()` generates an overview report for each management schedule that was implemented in the simulations, where the scheduled and triggered operations are matched and compared. The function prepares the scheduled management operations that were written in the input file `'management.sch'` in tabular form and randomly samples one HRU for each defined schedule from the triggered management operations (from the output file `'mgt_out.txt'`). The comparison is only done for operations that were defined with a fixed date in the management schedule. The operations, which are triggered by decision tables, will be excluded.

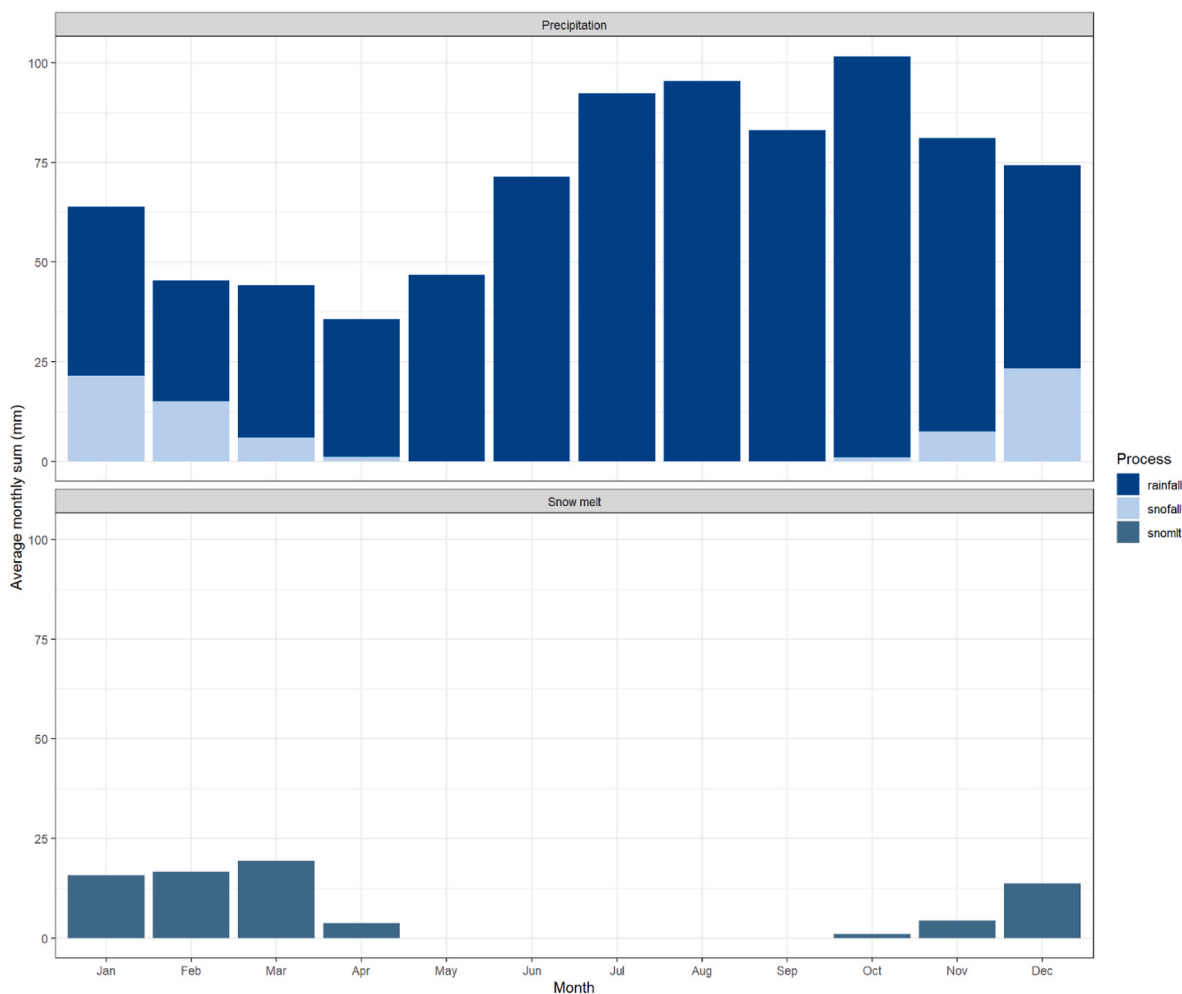


Fig. 4. Example plot of simulated basin variables precip, snowfall, and snowmelt with the function `plot_monthly_snow()`.

Applying the function `report_mgt()` for the model setup verification simulation outputs returns a table with an overview of the operations which were scheduled. The output table summarizes all operations which are found in either one of the SWAT+ files `'management.sch'` (management inputs) or `'mgt.out.txt'` (triggered management), or operations where the type of operations in the two files differ. Thus, `report_mgt()` is an efficient way to filter and identify potential issues in the management operation scheduling of a SWAT+ simulation. An example of a `report_mgt()` function application is provided below.

```
##Obtain report for problemating schedules
mgt_report <- report_mgt(sim_nostress)
##Print report
mgt_report
##Provide detail report for specific case
mgt_report$schedule_report[3]
```

Another way to analyze the triggered and the scheduled management is to compare the input and output files. `SWATdoctR` provides the function `print_triggered_mgt()` to print the triggered managements for individual HRUs. The helper function `get_hru_id_by_attribute()` is useful for selecting HRUs with a specific management. In the example below the `id` for an HRU was selected that uses the management `mgt = 'agrr_wwht'`. `print_triggered_mgt(sim_verify = sim_nostress, hru_id = hru_agr$id[1])` then shows the management which was triggered for a selected HRU. This table can be visually compared with the management input table (`'management.sch'`).

```
hru_wwht <- get_hru_id_by_attribute(sim_nostress,
```

```
mgt = 'agrr_wwht')
print_triggered_mgt(sim_verify = sim_nostress,
hru_id = hru_wwht$id[1])
```

Operations which are not triggered in the simulated management schedules must be checked in the `'management.sch'` input file. By answering the following questions for the scheduled management operations their proper implementation in the model setup can be verified:

- Are the date sequences in the scheduled operations correct and in the right order (mistakes in assigned month and day values)?
- Does the variable `op_data1` point to the correct entry in the respective input data file? Does the key exist in the input file? E.g. does defined `op_data1` exist in `'tillage.til'` for tillage operations, or does defined `op_data1` exist in `'plant.plt'` for plant operations?
- Does the variable `op_data2` point to the correct entry in the respective operations file (`'ops'`)? Does the harvest operation defined with `op_data2` exist in `'harv.ops'`?

### 3.1.4. Step 3. Analysis of unconstrained plant growth

The verification of plant growth is a two-tiered approach. In a first step plant growth is simulated and analyzed without simulating any stress factors. Such analysis illustrates the potential biomass and/or yields, which plant communities can gain given the climatic and soil conditions of the simulated catchment. Moreover, it allows the modeler to verify the duration of the scheduled growing period, or if the selected crop parametrization meets the climatic conditions. Plant stress can be activated/deactivated in the simulations by selecting the appropriate

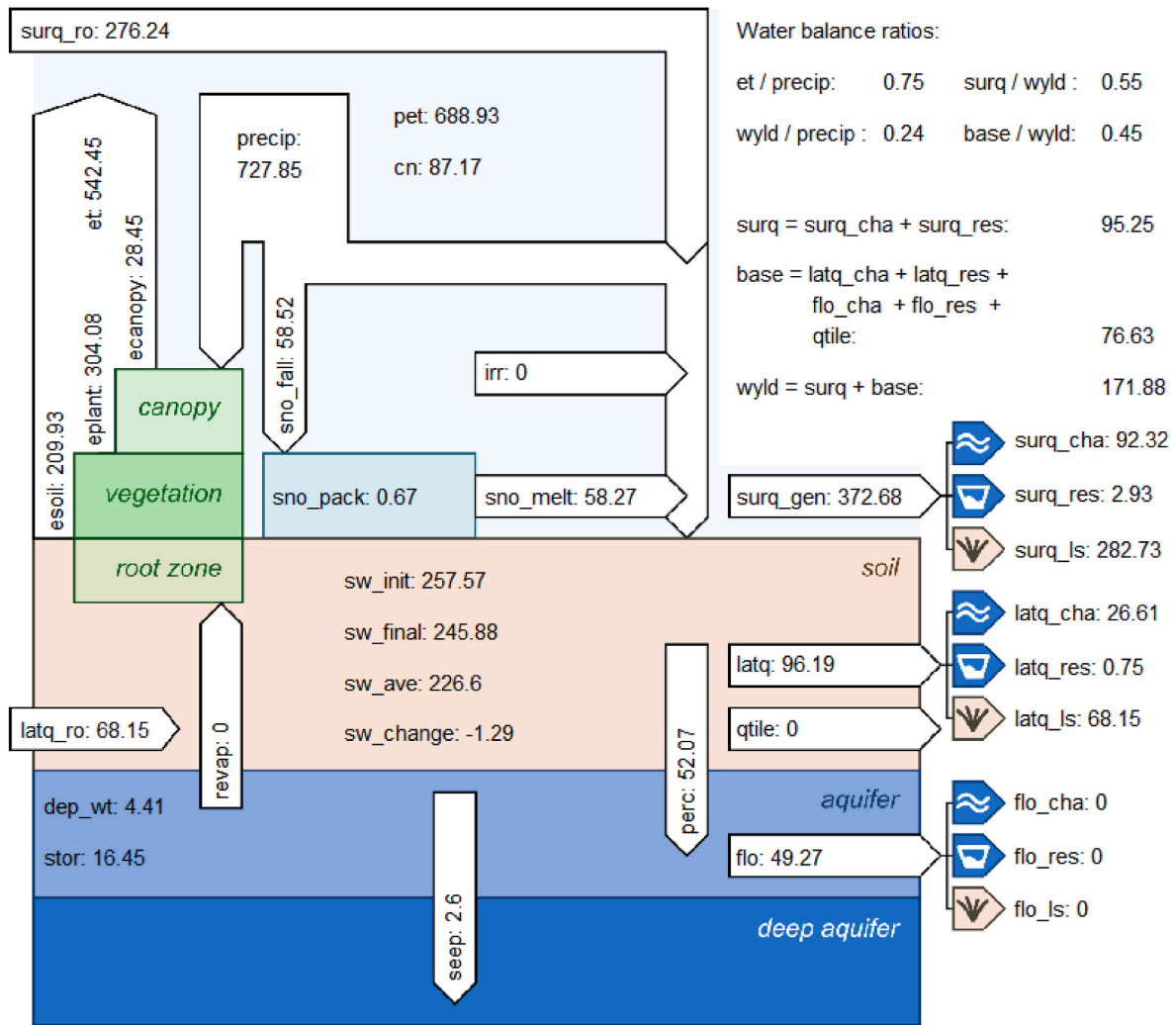


Fig. 5. Example plot of the simulated average annual basin water balance generated with the function plot\_waterbalance().

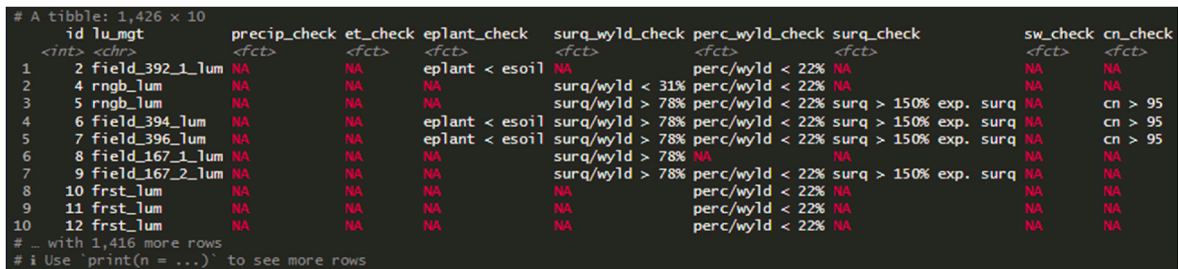


Fig. 6. Screenshot of analysis of HRU level water balance components with check\_hru\_waterbalance().

nostress parameter value (0/1/2 - all stress factor activated/all deactivated/only nutrient stress activated) in running run\_swat\_verification() function, which updates 'codes.bsn' model file.

SWAT+ simulations employ the heat unit (HU) concept (Barnard, 1948) to simulate the stages of plant development. HUs are units of temperature, which exceed a certain plant specific base temperature and are accumulated over a growing period. A certain budget of potential heat units (PHUs) must be accumulated in order to reach plant maturity.

Plant maturity at the harvest (crop yields reported by SWAT+ are in dry weight) can be investigated with plot\_variable\_at\_harvkill() function, which could be used to provide a general overview of the state of variables (outputs from 'mgt.out.txt') at the time of harvest/kill operations.<sup>1</sup> Following lines could be applied to generate such information, and Fig. 7 provides examples.

```
plot_variable_at_harvkill(sim_nostress, variable = 'yield')
```

<sup>1</sup> SWAT+ user must define the harvest/kill operation of a crop as two separate operations harvest and kill in the management schedule to be able to read the variable states at the last harvest operation before the plant is killed.

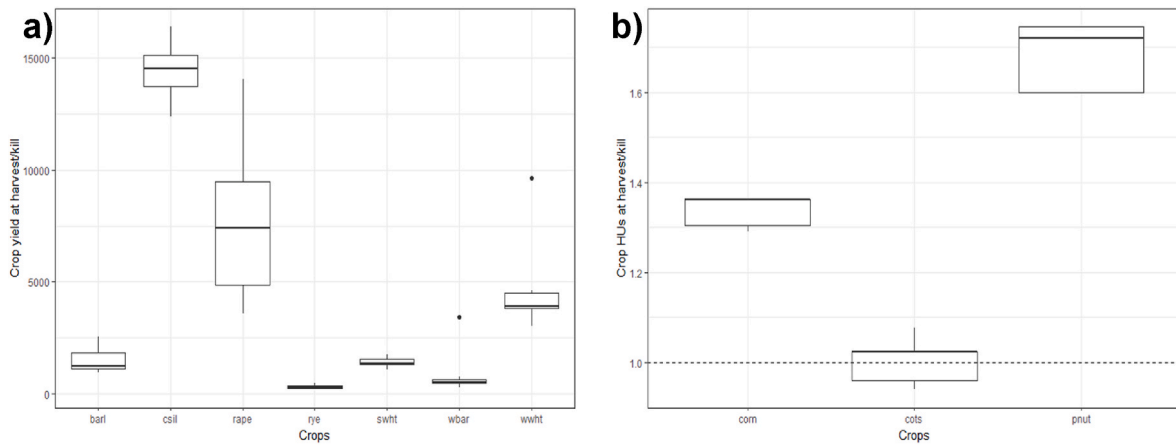


Fig. 7. a) example plot of the crop yields and b) example of crop heat unit fractions at harvest without simulated plant stress plotted with the function `plot_variable_at_harvkill()`.

`plot_variable_at_harvkill(sim_nostress, variable = 'phu')`

Another function in `SWATdoctR` is `plot_hru_pw()`. It allows the

analyses of the daily time series of HRU-related plant variables (outputs from `'hru_pw_day.txt'`), which then can only be performed for a few HRUs of a model setup. Examples for its use can be temporal

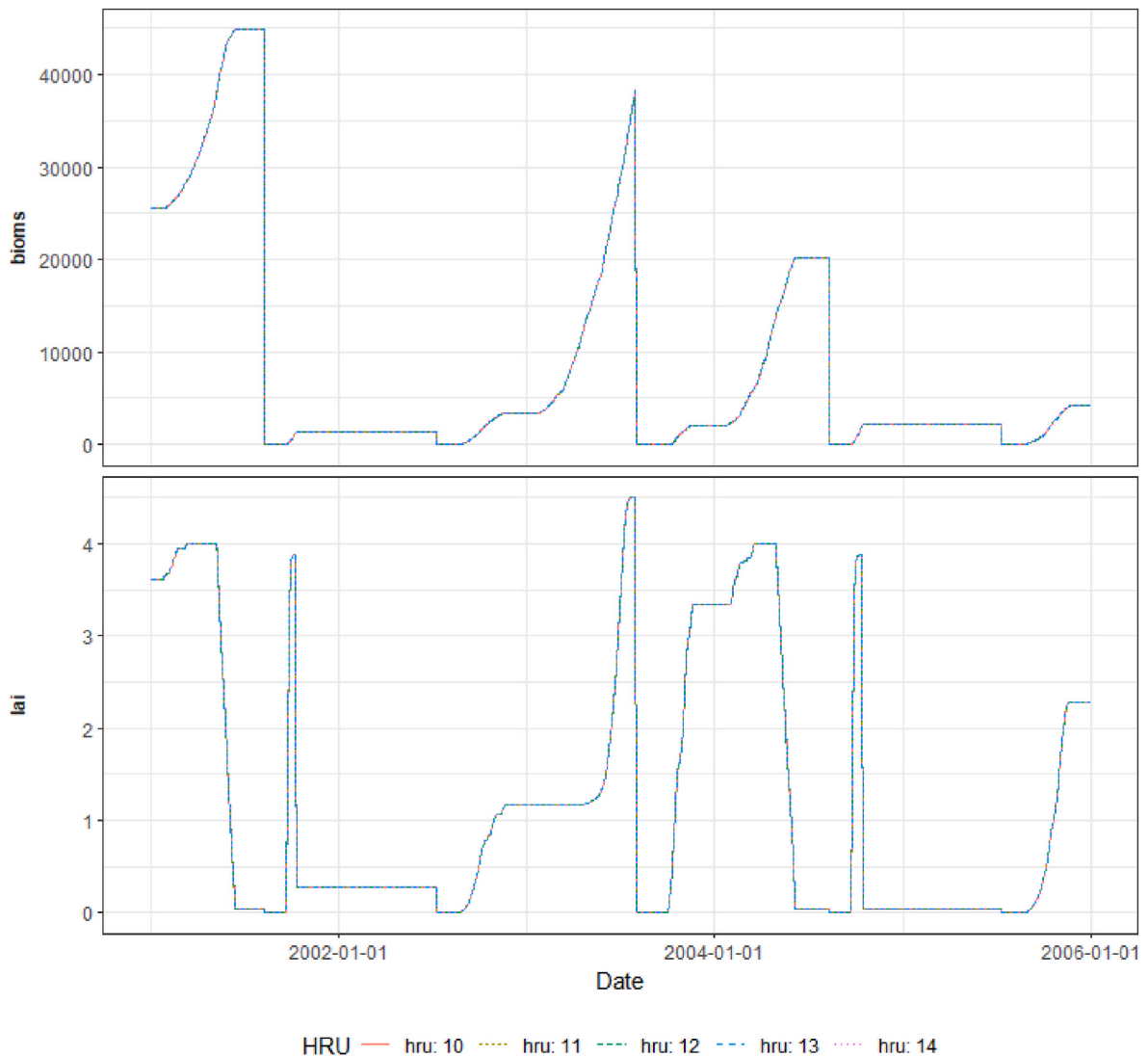


Fig. 8. Example plot of the daily LAI and biomass development showing problems with plant growth in the years 2001–2005 for 5 HRUs that implement the management schedule `'agrr_wwht'`.



examination of changes in simulated leaf area index (LAI), or biomass of a selected crop, which could be used to identify any abnormalities in plant growth. Particularly, for perennial plant land uses (e.g. forests, pastures) the temporal development of the plant biomass is a relevant variable to examine. An example of possible application of this function and its results are provided below (Fig. 8).

```
hru_agr <- get_hru_id_by_attribute(sim_nostress,
mgt = 'agrr_wvht')
plot_hru_pw(sim_verify = sim_nostress, hru_id =
sample(hru_agr$id, 5), var = c('lai', 'bioms'), years =
2001:2005)
```

### 3.1.5. Step 4. Model simulations with plant stress active

The next step includes activating potential sources for plant growth stresses, such as nutrient stress due to limited fertilizer inputs, or water stress due to limited water availability. An analysis of plant growth with active stresses can indicate issues in quantities of scheduled operations, such as the amounts of fertilizer inputs, or problems with definition of irrigation schedules, or decision table rules.

In a model simulation plant growth is often limited by the stress factors such as water stress, aeration stress, temperature stress, nitrogen stress or phosphorus stress. If any or several of those stress factors are significant in the simulation of the crop development, the simulated biomass and yields can be strongly reduced. The five different stresses are printed as the variables *strsw*, *strsa*, *strstmp*, *strsn*, *strsp*, into the file '*<scale>\_pw\_<time>.txt*'. Additionally, these variables are written as the variables *var4* (*strsw*), *var5* (*strsa*), *var3* (*strstmp*), *var1* (*strsn*), and *var2* (*strsp*) for harvest operations in the '*mgt.out.txt*', respectively. Both `plot_hru_pw()` and `plot_variable_at_harvkill()` can be applied to examine plant growth with stresses active. The function `plot_variable_at_harvkill()` could be used to summarize impacts of stress factors on plants as well (Fig. 9).

Turning off only the nutrient plant stress only can as well be a useful option for analyses (*nostress* = 2). This is particularly useful for eliminating the fertilization impact on the plant growth and focusing

only on the weather/climate and structural setting of the land use. Aeration, temperature, and water stress, alongside yields are relevant outputs to be analyzed. A simulation with inactive nutrient stress will provide a good approximation of possible yields with an optimal fertilization and ideal plant nutrient supply. All other stresses will indicate the need of irrigation, drainage or plant-specific parameter adjustments for a plant to grow. The following line example will activate stresses in simulation.

```
sim_stress <- run_swat_verification(project_path =
model_path, outputs = c('wb', 'mgt', 'plt'), nostress = 0)
plot_variable_at_harvkill(sim_stress, variable =
'stress')
```

The `plot_variable_at_harvkill()` function could be applied on a list holding several `run_swat_verification()` function run results. Results would be plotted on one figure for easy comparison (Fig. 10).

```
sim_list <- list(no_stress = sim_nostress1,
except_nutrient = sim_nostress2, all_stress =
sim_nostress0)
plot_variable_at_harvkill(sim_list, variable =
'yield')
```

To summarize steps 3–4, the following full procedure can be recommended to verify the appropriate functioning of plant growth:

- Simulate the plant growth with all the stresses turned off (*nostress* is set to 1) and check if the plant is growing (LAI and biomass are increasing). This will show if the plant is actually set-up in a way that the model is simulating the growth cycle.
- Simulate the plant growth with only fertilizer stress off (*nostress* is set to 2) and check if the plant is growing (LAI and biomass are increasing). If the plant/crop is harvested, at this stage.
- Simulate the plant growth with enabled stresses (*nostress* is set to 0). If unreasonable values for crop growth (LAI, biomass, yields, stresses) are produced, this is an indicator of possible setup or

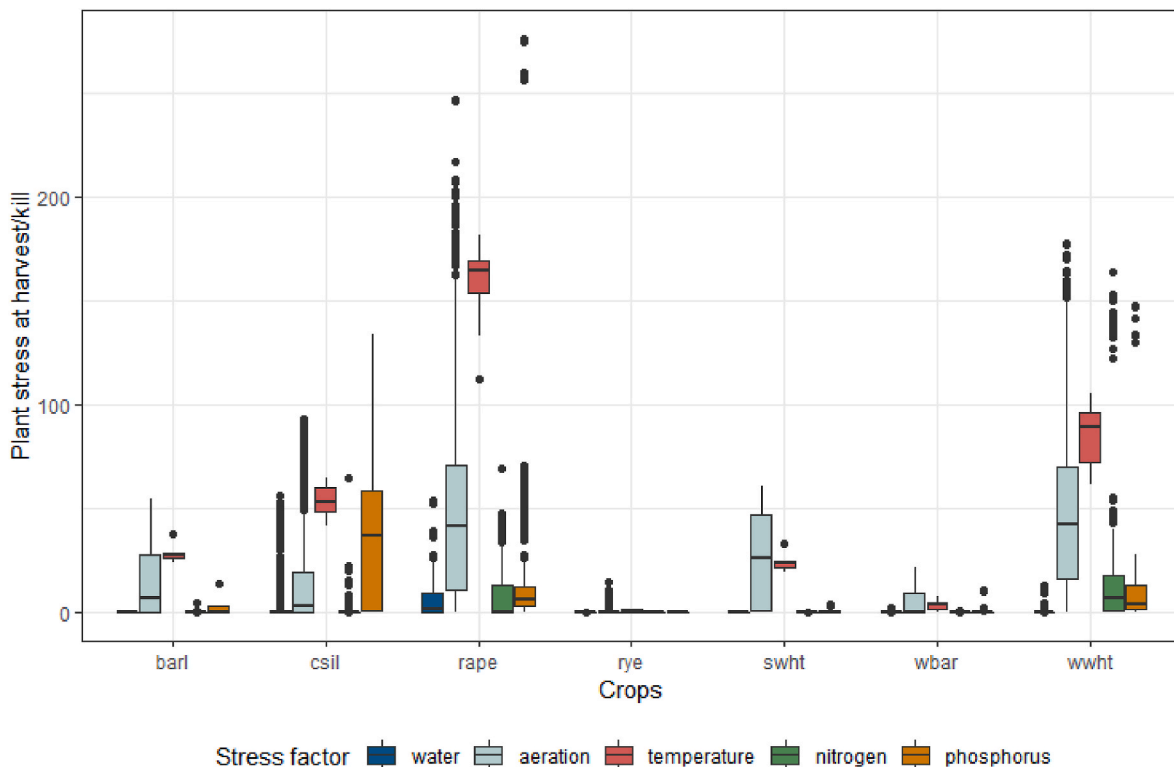


Fig. 9. Example of plant stress factors per crop for the simulations with active plant stress plotted with the function `plot_variable_at_harvkill()`.

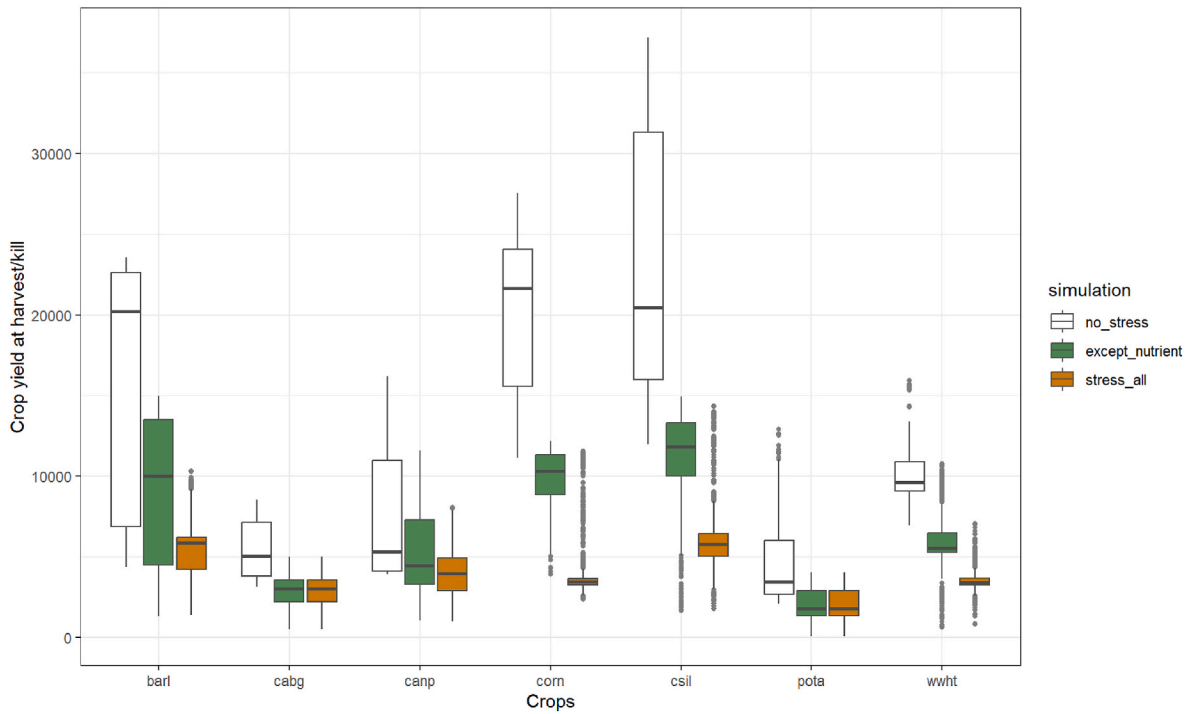


Fig. 10. Example of several run\_swat\_verification() function results plotted with the function plot\_variable\_at\_harvkill().

parametrization errors in the model and should be investigated and fixed.

plants (WWTPs), on-site wastewater systems or water transfers are defined with the files 'recall.rec' and 'recall.con' and corresponding time series records '.rec' files in a SWAT+ model setup. The point source time series inputs define the water, sediment, and nutrient loads which are emitted by a point source into a spatial object. The common issues are

### 3.1.6. Step 5. Simulation of point inputs and tile drains

Point sources, such as municipal/industrial waste water treatment

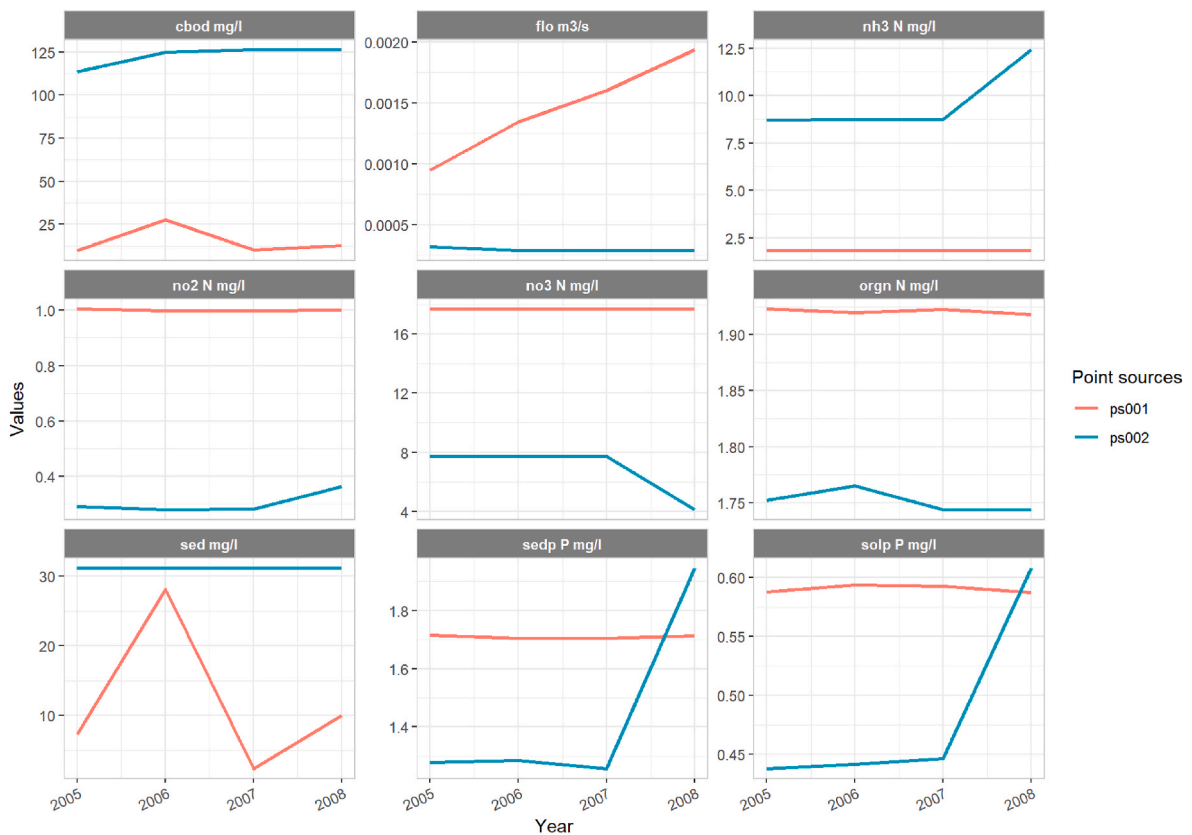


Fig. 11. Example of simulated point source data plotted with the plot\_ps() function.

wrong units of the defined fluxes or wrong time intervals for a certain accumulated flux. SWAT+ projects can potentially have a large number of unique point sources, which may be another reason for errors. Thus, it is good practice to verify the simulated influxes from point sources into the respective spatial objects.

Verification of point source simulation can be done with the `plot_ps()` function (Fig. 11) to plot simulated point source data. It requires only objects from `run_swat_verification()` and allows examining point source loading amounts or concentrations (by switching `conc = TRUE`).

```
plot_ps(sim_stress, conc = TRUE)
```

Flow from tile drainage systems into channels is defined in the file `'rout_unit.con'` by sending a certain flow fraction (`frac`) as tile flow (`hyd_typ` defined as `til`) to a channel (`obj_typ` defined as `sdc`) with the respective `obj_id`. Further, the defined land use and management (`'landuse.lum'`) of a tile drained land object must point to the parametrization of a tile drainage network (parameter tile points to entry in `'tiledrain.str'`).

The verification of tile flow should in first place focus on whether tile flow occurs or not. The occurrence of tile flow can be verified with the output variable `qtile` for the respective land objects in the output file `'hru_wb_aa.txt'`. If no tile flow occurs for an HRU for which tile flow was parameterized the model inputs above have to be checked for any errors. `SWATdoctR` function `print_avannual_qtile()` selects all HRUs for which the land use definition uses a tile flow parametrization (the variable `'tile'` in `'landuse.lum'`) and prints average annual tile flows in tabular form. Additionally, `plot_qtile()` function could be applied to plot tile drain water flow distribution in the model (Fig. 12). It may be useful to compare the simulated tile flow values with the values of other water balance components (Fig. 5).

```
plot_qtile(sim_stress_all)
```

### 3.2. Examples of verification results for model setups

The following are issues identified during the verification process with `SWATdoctR` in different case study SWAT+ model setups from different countries. Locations, details on model input data and model

setup information are intentionally excluded as the main purpose of this section is to show examples of model setup issues, which could be identified during the verification process with the `SWATdoctR` package. General characteristics of tested SWAT+ model setups is provided in Table 1.

#### 3.2.1. Case I

Climate variables investigation indicated issues with model evaporation as the results showed suspiciously high canopy evaporation (larger than plant and soil evaporation combined). Examining management outputs indicated many potential problems in 43 out of 49

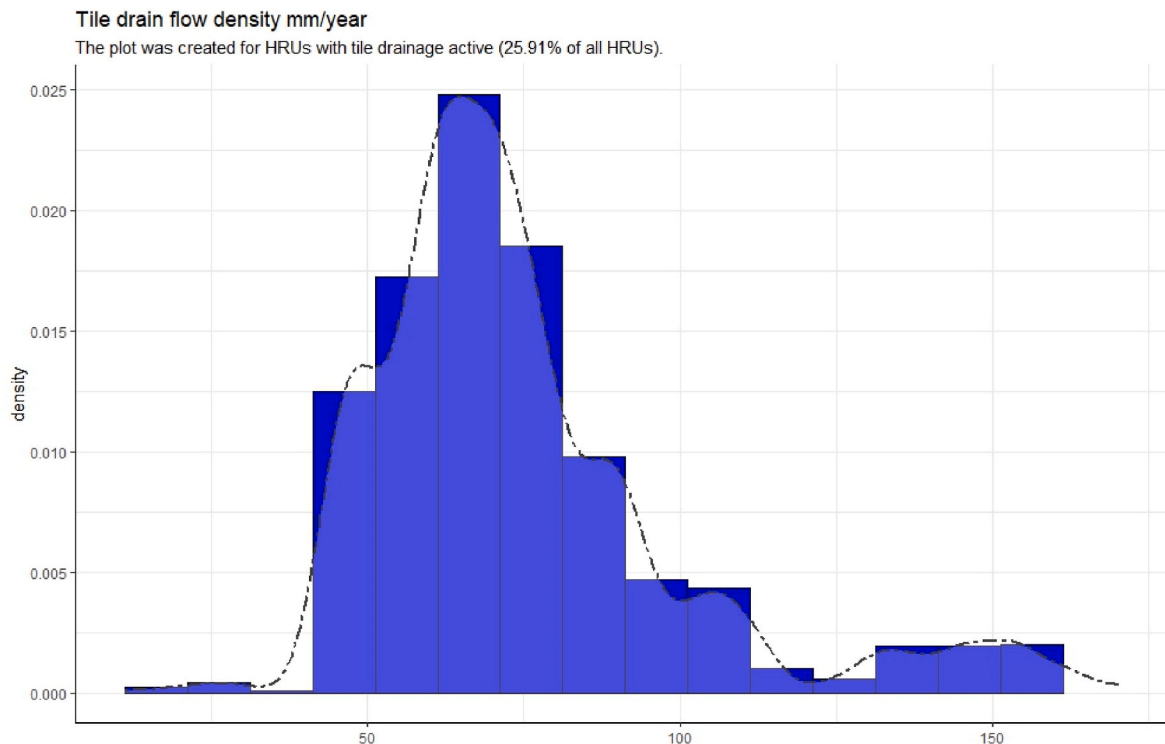
**Table 1**  
Characteristics of SWAT+ model setups used in case comparison.

	Case I	Case II	Case III	Case IV
<b>Catchment size (km<sup>2</sup>)</b>	417	386	150	2926
<b>Land use (%):</b>				
- forest	22	16	5.5	1
- pastures	45	8	2	4
- arable land	30	67 (39 <sup>a</sup> )	89	83 (77 <sup>b</sup> )
- urban and water	3	7 (3 + 4)	3.5	12 (7 + 5)
<b>HRU number</b>	1197	5391	10240	7437
<b>Routing units</b>	9	93	10240	68
<b>Channels</b>	9	93	128	522
<b>Reservoirs</b>	3	0	27	67
<b>Aquifers</b>	10	94	1	34
<b>Point sources</b>	6	93	2	34
<b>Management</b>	HU-based	HU-based	Date-based	Date-based
<b>HRUs with tile drains (%)</b>	83	26	59	83
<b>Calibration status</b>	Calibrated flows and water quality	In process	Not started	Finished soft calibration <sup>c</sup>

<sup>a</sup> Orchards.

<sup>b</sup> Soybeans and corn rotation.

<sup>c</sup> Soft calibrated for water balance and crop yields.



**Fig. 12.** Example of tile drain functioning plotted with the `plot_qtile()` function.

schedules in the model. The main problem was that some operations, which were supposed to be triggered in a single year, were actually triggered in the following year. These issues caused multiple problems in the model such as very high temperature stress levels for summer crops, which were planted at the wrong time. Also yields were quite low for some plants affected by this issue. Some plants were also affected by fertilization operations occurring at the wrong timing, e.g. in winter, when plants should not be growing. This issue was caused by a seemingly minor error in SWAT+ 'management.sch' file, providing the wrong number of operations for each schedule. Median water flow value for tile drains was around 163 mm/year (full range in 0–401 mm/year), which is very high for this area and requires investigation of water balance processes. Point sources are active in the model. However, unexplainable spikes in flows and loads of some parameters brings into question the quality of model input data.

### 3.2.2. Case II

There were no issues identified for climatic variables. However, the examination of management identified issues in the 6 out of 10 management schedules. The main problem, again, was connected to some operations spilling over to the next year. This was caused by the same mistake in SWAT+ 'management.sch' file. It caused plants to be planted and fertilized at the wrong timing and caused huge temperature and nitrogen stresses. Nutrient stress activation reduced yields significantly, however activation of all stresses reduced yields by around three times.

Water flow for drained HRUs was not indicating any issues (median - 61, min - 13, max - 158 mm/year). However, point sources were missing actual data about flows and loads, which indicated potential issues. In this case point sources were added to several channels as "average annual" (exco in SWAT files) inputs. If this option is used, the point source inputs are not reported in the "recall" output files. SWATdoctr made us aware of this issue of not writing point source outputs by SWAT+, which was reported to the SWAT+ development team for further debugging.

### 3.2.3. Case III

The assessment of simulated climatic variables indicated problems with PET, which were far too low for the catchment. This problem was caused by incorrect relative humidity units in input data. Water partition figure also allowed us to identify some HRUs with extreme surface runoff numbers. This was caused by incorrect parametrization of wetland areas.

The investigation of management issues showed that there were problems in 1065 management schedules out of 7682 existing in the model. This model had individual date based management schedules for each field in the catchment created with the SWATfarmR package. It caused the management operation files for this model to be very large ('management.sch' was more than 200 Mb). Thus SWATdoctr helped to identify multiple problems, which were nearly impossible to identify manually. Among the main problems was a mis-placement of harvest operations for some crops. Another problem was some typos in the management files. These errors caused some crops not being harvested or some schedules not being triggered. Additionally, such problems could be identified from plant harvest plots as well.

Water flows in tile drains are not represented correctly, this has an impact on tile drain flows too. The median tile drain flow was 0 mm/year for this setup, while the average was 5 mm/year (0 min, max 209 mm/year). This shows that tile drains are active, however water balance simulation should be thoroughly analyzed and improved before examining tile drain flows in more detail. Point source examination allowed us to notice that point source data was provided in wrong units to this model.

### 3.2.4. Case IV

SWATdoctr helped pinpoint problems, which were complex to identify manually, considering the complexity and the size of the model.

The most significant was the analysis of crop yields and crop heat unit fractions at harvest without simulated plant stress, as described in Step 3. The simulated yields and plant accumulated HUs at harvest for winter wheat were very low in a normal simulation (with plant stresses). After running the model without any stresses (`nostress = 1`), SWAT+ still reported low yields, with accumulated HU averaging at 0.6 fraction of fully mature plant values. After adjusting the management schedule for harvest/kill operation, the simulated yields and accumulated HU did not improve. The potential issue in winter crop HU accumulation was identified and reported to the SWAT+ development team, who acknowledged the fact and are further investigating the issue. In this case, SWATdoctr assisted in both, troubleshooting the model setup, as well as diagnose issues in the modeling tool itself.

## 4. Discussion and conclusions

We have developed and tested SWAT+ model setup verification workflow together with the SWATdoctr package on four SWAT+ model setups with different levels of model preparation (not-calibrated, soft-calibrated and calibrated). The proposed verification procedure and SWATdoctr tool proved to be valuable in identifying various issues with model setups that would have otherwise gone unnoticed in all of these cases. Addressing these issues resulted in enhanced reliability of the model applications, improved performance, and reduced time requirements. This, in turn, prevented possible problematic issues from arising during the later stages of model calibration and application.

The model setup verification procedure, supported by the SWATdoctr package or similar tools, should be seen as a necessary step right after completing the model setup and before starting the model calibration. The model setup verification procedure helps to identify and eliminate input or structural errors before the model is calibrated, reducing the likelihood of problems occurring in later stages of model application. By verifying the setups, modelers can ensure that their applications are of high quality, reliable, and meet the needs of its users. This helps to build user confidence in the models, reduce support costs, and increase stakeholder satisfaction. In addition, verifying the model functionality can help to identify areas for improvement, as demonstrated in Case IV, which can lead to enhancements in functionality, performance, and usability over time.

We foresee that the usage of verification tools, such as SWATdoctr may go beyond and serve as a quality control of model calibration. Highly parameterised models, such as SWAT+, may be overfitted during the automatic calibration process. SWATdoctr can guide the users to identify such pitfalls from provided plots.

Compared to the SWATCheck tool, the SWATdoctr R package offers additional functionalities. Its functions enable a detailed investigation of the underlying causes of specific issues found in tested models. Moreover, the package functions can be easily integrated with R markdown documents to generate automated and customized model setup verification reports. This greatly expands the potential applications of the tool, such as generating setup verification reports for rapid problem screening or as supplements to model preparation documentation, utilizing package functions directly in the creation of articles, books, or scripted workflows, and more.

This tool is still in the development stage and many functionalities are being developed, tested and improved. We invite the SWAT+ modeling community for testing, providing feedback, contributing to adding and improving SWATdoctr package features, thus ensuring SWAT+ model reliability and helping to develop more sound model applications.

### Software availability

Name of the software: SWATdoctr

Developer: Christoph Schürz [aut, cre]

Svajunas Plunge [aut] Contact information: [christoph.schuerz@ufz.de](mailto:christoph.schuerz@ufz.de).

## de

Year first available: 2022.

Program language: R

Cost: free

Software availability: <https://git.ufz.de/schuerz/swatdoctr>

Program size: 685 KB

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgements

This work was accomplished within the OPTAIN project (Optimal strategies to reTAIN and re-use water and nutrients in small agricultural catchments across different soil-climatic regions in Europe, cordis.europa.eu) which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 862756.

### References

- Texas A&M AgriLife Research and USDA, 2023. SWAT+ Editor Documentation. <https://swatplus.gitbook.io/docs/user/editor>.
- Akoko, George, Le, Tu Hoang, Gomi, Takashi, Kato, Tasuku, 2021. A review of SWAT model application in Africa. *Water* 13 (9), 1313. <https://doi.org/10.3390/w13091313>.
- Arnold, J.G., Srinivasan, R., Muttiah, R.S., Williams, J.R., 1998. Large area hydrologic modeling and assessment Part I: model development. *JAWRA J. Am. Water Resour. Associat.* 34 (1), 73–89. <https://doi.org/10.1111/j.1752-1688.1998.tb05961.x>.
- Arnold, J.G., Moriasi, D.N., Gassman, P.W., Abbaspour, K.C., White, M.J., Srinivasan, R., Santhi, C., et al., 2012. SWAT: model use, calibration, and validation. *J. Name Chang. Trans. ASABE* 55 (4), 1491–1508.
- Arnold, J.G., Kiniry, J.R., Srinivasan, R., Williams, J.R., Haney, E.B., Neitsch, S.L., 2012. Soil & Water Assessment Tool, Input/Output Documentation. Texas Water Resources Institute-TR-439. <https://swat.tamu.edu/media/69296/swat-io-documentation-2012.pdf>.
- Arnold, Jeffrey G., Bieger, Katrin, White, Michael J., Srinivasan, Raghavan, Dunbar, John A., Allen, Peter M., 2018. Use of decision tables to simulate management in SWAT+. *Water* 10 (6), 713. <https://doi.org/10.3390/w10060713>.
- Barnard, J.D., 1948. Heat units as a measure of canning crop maturity. *Canner* 106 (16), 28.
- Barrenis, Luis Osvaldo, 2022. Spotswatplus: Link the New Version of the Soil and Water Assessment Tool Plus with the Spotpy Library in Order to Perform Sensitivity and Uncertainty Analysis and Calibration. July. <https://github.com/osvaluis/Spotswatplus>.
- Bieger, Katrin, Arnold, Jeffrey G., Rathjens, Hendrik, White, Michael J., Bosch, David D., Allen, Peter M., Volk, Martin, Srinivasan, Raghavan, 2017. Introduction to SWAT+, A completely restructured version of the soil and water assessment tool. *JAWRA J. Am. Water Resour. Associat.* 53 (1), 115–130. <https://doi.org/10.1111/1752-1688.12482>.
- Bieger, Katrin, Arnold, Jeffrey G., Rathjens, Hendrik, White, Michael J., Bosch, David D., Allen, Peter M., 2019. Representing the connectivity of upland areas to floodplains and streams in SWAT+. *JAWRA J. Am. Water Resour. Associat.* 55 (3), 578–590. <https://doi.org/10.1111/1752-1688.12728>.
- Bruce, Peter, Bruce, Andrew, Gedeck, Peter, 2020. *Practical Statistics for Data Scientists: 50 Essential Concepts Using R and Python*. O'Reilly Media, Inc. Nicole Tache. 978-1492072942.
- CARD&ISU, 2023. SWAT Literature Database for Peer-Reviewed Journal Articles. [http://www.card.iastate.edu/swat\\_articles/](http://www.card.iastate.edu/swat_articles/).
- Čerkasova, Natalja, Umgieser, Georg, Ali, Ertürk, 2019. Assessing climate change impacts on streamflow, sediment and nutrient loadings of the minija river (Lithuania): a hillslope watershed discretization application with high-resolution spatial inputs. *Water* 11 (4), 676. <https://doi.org/10.3390/w11040676>.
- Chawanda, Celray James, George, Chris, Thiery, Wim, Ann van Griensven, Tech, Jaclyn, Arnold, Jeffrey, Srinivasan, Raghavan, 2020. User-friendly workflows for catchment modelling: towards reproducible SWAT+ model studies. *Environ. Model. Software* 134, 104812. <https://doi.org/10.1016/j.envsoft.2020.104812>.
- Coon, Ethan T., Shuai, Pin, 2022. Watershed workflow: a toolset for parameterizing data-intensive, integrated hydrologic models. *Environ. Model. Software* 157, 105502. <https://doi.org/10.1016/j.envsoft.2022.105502>.
- Ferreira, David Bispo, 2019. "PySWAT: A Python Application for Input/Output Analysis for the Soil and Water Assessment Tool(SWAT). December. <https://github.com/davidbispo/PySWAT>.
- Fu, B., Horsburgh, J.S., Jakeman, A.J., Gualtieri, C., Arnold, T., Marshall, L., Green, T.R., et al., 2020. Modeling water quality in watersheds: from here to the next generation. *Water Resour. Res.* 56 (11) <https://doi.org/10.1029/2020wr027721>, 10.1029/2020wr027721.
- Fuka, Daniel R., Todd Walter, M., MacAlister, Charlotte, Steenhuis, Tammo S., Easton, Zachary M., 2014. SWATmodel: a multi-operating system, multi-platform SWAT model package in R. *JAWRA J. Am. Water Resour. Associat.* 50 (5), 1349–1353. <https://doi.org/10.1111/jawr.12170>.
- Gassman, Philip W., Wang, Yingkuan, 2015. IJABE SWAT special issue: innovative modeling solutions for water resource problems. *Int. J. Agric. Biol. Eng.* 8.
- Gassman, P.W., Reyes, M.R., Green, C.H., Arnold, J.G., 2007. The soil and water assessment tool: historical development, applications, and future research directions. *Trans. ASABE* 50, 1211–1250.
- Gassman, Philip W., Sadeghi, Ali M., Srinivasan, Raghavan, 2014. Applications of the SWAT model special section: overview and insights. *J. Environ. Qual.* 43 (1), 1–8. <https://doi.org/10.2134/jeq2013.11.0466>.
- Houska, Tobias, Kraft, Philipp, Chamorro-Chavez, Alejandro, Breuer, Lutz, 2015. SPOTting model parameters using a ready-made Python package. *PLoS One* 10 (12), e0145180. <https://doi.org/10.1371/journal.pone.0145180>.
- Kmoch, Alexander, 2022. Swatpy: A Set of Python Modules to Work with SWAT2012 Models. <https://doi.org/10.5281/zenodo.6322023>. March.
- Maref, Noureddine, Baahmed, Djelloul, Bemoussat, Kader, Mahfoud, Zakaria, 2022. SWAT Model Application for Sediment Yield Modeling and Parameters Analysis in Wadi K'sob (Northeast of Algeria). <https://doi.org/10.21203/rs.3.rs-2069353/v1>.
- Musyoka, Kilundu, Francis, Strauss, Peter, Zhao, Guangju, Srinivasan, Raghavan, Andreas, Klik, 2021. Multi-step calibration Approach for SWAT model using soil moisture and crop yields in a small agricultural catchment. *Water* 13 (16), 2238. <https://doi.org/10.3390/w13162238>.
- Neitsch, Susan L., Arnold, Jeffrey G., Kiniry, Jim R., Williams, Jimmy R., 2011. Soil and Water Assessment Tool Theoretical Documentation Version 2009. Texas Water Resources Institute. <https://swat.tamu.edu/media/99192/swat2009-theory.pdf>.
- Nguyen, Tam V., Dietrich, Jörg, Dang, Thanh Duc, Tran, Dang An, Doan, Binh Van, Sarrazin, Fanny J., Abbaspour, Karim, Srinivasan, Raghavan, 2022. An interactive graphical interface tool for parameter calibration, sensitivity analysis, uncertainty analysis, and visualization for the soil and water assessment tool. *Environ. Model. Software* 156, 105497. <https://doi.org/10.1016/j.envsoft.2022.105497>.
- Nkwasa, Albert, Chawanda, Celray James, Anna, Msigwa, Komakech, Hans C., Verbeiren, Boud, Griensven, Ann van, 2020. How can We represent seasonal land use dynamics in SWAT and SWAT+ models for African cultivated catchments? *Water* 12 (6), 1541. <https://doi.org/10.3390/w12061541>.
- Nkwasa, Albert, Chawanda, Celray James, Jonas, Jägermeyr, Griensven, Ann van, 2021. Improved representation of agricultural land use and crop management for large-scale hydrological impact simulation in Africa using SWAT+. *Hydrol. Earth Syst. Sci.* 26 (1), 71–89. <https://doi.org/10.5194/hess-26-71-2022>.
- NRES-21 Hydrology committee of ASABE, 2017. Guidelines for Calibrating, Validating, and Evaluating Hydrologic and Water Quality (H/WQ) Models. American Society of Agricultural and Biological Engineers, June.
- Plunge, Svajunas, 2023. SWATprepR - tools for SWAT+ input data preparation, visualization and model output assessment. <https://biopsichas.github.io/SWATprepR/>.
- Plunge, Svajunas, Gudas, Mindaugas, Povilaitis, Arvydas, 2022. Expected climate change impacts on surface water bodies in Lithuania. *Ecohydrol. Hydrobiol.* 22 (2), 246–268. <https://doi.org/10.1016/j.ecohyd.2021.11.004>.
- Plunge, Svajunas, Gudas, Mindaugas, Povilaitis, Arvydas, Piniowski, Mikołaj, 2023. Evaluation of the costs of agricultural diffuse water pollution Abatement in the context of Lithuania's water protection goals and climate change. *Environ. Manag.* 71 (4), 755–772. <https://doi.org/10.1007/s00267-022-01745-1>.
- Python Software Foundation, 2023. Python Language Reference, Version 3.11. <http://www.python.org>.
- R Foundation, 2023. The R Project for Statistical Computing, 4.2. <https://www.r-project.org/>.
- Schuerz, Christoph, 2022. SWATfarmR: Simple Rule Based Scheduling of Management Operations in SWAT. R. <https://github.com/chrissschuerz/SWATfarmR>.
- Schürz, Christoph, 2019. SWATplusR: Running SWAT2012 and SWAT+ Projects in R. <https://github.com/chrissschuerz/SWATplusR>.
- SWAT model website (n.d.). SWAT+ IO Documentation. <https://swatplus.gitbook.io/io-docs/>.
- Tan, Leong, Mou, Gassman, Philip, Yang, Xiaoying, James, Haywood, 2020. A review of SWAT applications, performance and future needs for simulation of hydro-climatic extremes. *Adv. Water Resour.* 143, 103662 <https://doi.org/10.1016/j.advwatres.2020.103662>.
- White, Michael J., Daren Harmel, R., Arnold, Jeff G., Williams, Jimmy R., 2014. SWAT check: a screening tool to assist users in the identification of potential model application problems. *J. Environ. Qual.* 43 (1), 208–214. <https://doi.org/10.2134/jeq2012.0039>.
- White, Michael J., Arnold, Jeffrey G., Bieger, Katrin, Allen, Peter M., Gao, Jungang, Čerkasova, Natalja, Gambone, Marilyn, et al., 2022. Development of a field scale SWAT+ modeling framework for the contiguous U.S. *JAWRA J. Am. Water Resour. Associat.* <https://doi.org/10.1111/1752-1688.13056>.
- Wu, Yiping, Liu, Shuguang, 2012. Automating calibration, sensitivity and uncertainty analysis of complex models using the R package flexible modeling environment

- (FME): SWAT as an example. Environ. Model. Software 31, 99–109. <https://doi.org/10.1016/j.envsoft.2011.11.013>.
- Wu, Yiping, Liu, Shuguang, 2014. Improvement of the R-SWAT-FME framework to support multiple variables and multi-objective functions. Sci. Total Environ. 466, 455–466. <https://doi.org/10.1016/j.scitotenv.2013.07.048>.
- Yang, Chengde, Xu, Min, Fu, Congsheng, Kang, Shichang, Luo, Yi, 2022. Glacier Hydrological Process Modeling Based on Improved SWAT+: A Case Study in the Upper Yarkant River Basin. <https://doi.org/10.22541/au.164512280.00856493/v1>.