

Rational gaze: Presenting the open-source HYPEtools R package for analysis, visualization, and interpretation of hydrological models and datasets

Conrad Brendel^{*}, René Capell, Alena Bartosova

Swedish Meteorological and Hydrological Institute, Norrköping, Sweden

ARTICLE INFO

Keywords:

R
Hydrological modeling
Data
Analysis
Visualization
Mapping

ABSTRACT

The open-source HYPEtools R package aims to address current challenges in hydrological modeling and data analysis by providing a suite of tools for data management, visualization, interpretation, and exploration. Originally developed as a supporting toolbox for the HYPE hydrological model, HYPEtools has since evolved into a standalone package with features ranging from diverse aggregation and evaluation routines to interactive apps and mapping. Specifically, functions are included for data manipulation and conversion, data summarization, plotting and mapping, and interactive data exploration. Two case studies are provided to highlight how HYPEtools can simplify hydrological workflows and analyses both independently of and within HYPE modeling contexts. First, HYPEtools is used for data exploration and mapping of water transfers in South Africa. Then, a case study for the Baltic Sea region of a pan-European HYPE model setup illustrates how HYPEtools can assist model calibration and validation.

1. Introduction

As simplified representations of real-world systems, hydrological models can improve understanding, prediction, and management of water resources across a wide range of applications (Moradkhani and Sorooshian, 2008). Models developed for research purposes can provide insights into the hydrological processes governing a system and allow everything to be examined, separated, one thing at a time. In contrast, models created for simulation and prediction — such as those for flood forecasting and climate change impact assessments — can guide planning and decision making while evaluating the consequences for society and the environment. Since its origins in the 1850s, hydrological modeling has steadily progressed through developments in concepts and theories, data collection and availability, computational and analysis tools, and integration with other technologies and sciences (Singh, 2018). In recent years, there has been a great increase in the amount of data available for model development and calibration due to new data sources such as earth observations (McCabe et al., 2017), low-cost Internet of Things (IoT) sensors (Drost et al., 2022), and citizen science networks (Lowry et al., 2019; Njue et al., 2019). Combined with increases in computational power, this increased data access has encouraged the application of hydrological models to larger

continental/global domains (Arheimer et al., 2020; Brendel et al., 2023; Sood and Smakhtin, 2015; Tijerina-Kreuzer et al., 2021) and modeling at higher temporal and spatial resolutions (Burek et al., 2020; Chen et al., 2017). As models grow in detail and complexity, there is an increased need for tools and methods to support the processing, analysis, visualization, and interpretation of hydrological models and datasets.

Advances in hydrological modeling have been matched by the expansion of open-source programming languages to support data processing and analyses in the hydrological sciences. In particular, the R programming language, which was originally developed for statistical computing (Ihaka and Gentleman, 1996), has evolved into a highly versatile multiplatform language with a multitude of packages for hydrological data retrieval, analysis, and modeling (Albers and Prosdoci, 2023). A review by Slater et al. (2019) highlights key R packages to facilitate typical hydrological workflows. R offers several specialized data retrieval packages — such as *dataRetrieval* (DeCicco et al., 2023) and *easyclimate* (Cruz-Alonso et al., 2023) — which provide tools to simplify the task of retrieving and pre-processing hydrological and meteorological datasets. In addition, R provides a wide variety of packages to assist with hydrological data analysis tasks such as data tidying and manipulation (e.g. *tidyverse* (Wickham et al., 2019)), performing hydrograph analyses (e.g. *grwat* (Samsonov, 2023) and *hydro-Events* (Wasko and Guo, 2022)), working with meteorological and

^{*} Corresponding author.

E-mail addresses: conrad.brendel@smhi.se (C. Brendel), rene.capell@smhi.se (R. Capell), alena.bartosova@smhi.se (A. Bartosova).

<https://doi.org/10.1016/j.envsoft.2024.106094>

Received 12 March 2024; Received in revised form 23 May 2024; Accepted 31 May 2024

Available online 2 June 2024

1364-8152/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Abbreviations

DWS	Department of Water and Sanitation
HRU	Hydrological Response Unit
KGE	Kling-Gupta Efficiency
MAE	Mean Absolute Error
NSE	Nash-Sutcliffe Efficiency
PBIAS	Percent Bias
SLC	Soil type – Land use Combination
SMHI	Swedish Meteorological and Hydrological Institute
VE	Volumetric Efficiency

climate data (e.g. *openair* (Carslaw and Ropkins, 2012)), and spatial data processing (e.g. *arcgisbinding* (Aydin et al., 2024), *sf* (Pebesma and Bivand, 2023), and *terra* (Hijmans, 2023)). R packages for hydrological modeling include those to help with activities such as preparing model inputs/outputs (e.g. *RavenR* (Chlumsky et al., 2022)), running process-based models (e.g. *agriwater* (de Oliveira Ferreira Silva et al., 2019)), and performing statistical modeling (e.g. *NPRED* (Sharma et al., 2016) and *WASP* (Jiang et al., 2021)). A review by Astagneau et al. (2021) compares the applicability and usability of several hydrological models implemented within R packages. A final noteworthy R package is *shiny* (Chang et al., 2023), which provides a framework for building interactive web apps and has been used to develop various apps for hydrological data retrieval, analysis, and visualization (Brendel et al., 2019, 2020; Cinkus et al., 2023; Rottler et al., 2021).

The convergence of hydrological models and the R programming language allows for expansion of the analysis and visualization capabilities of the models and as well as improvements in the reproducibility and replicability of the model construction and calibration. Several R packages have been developed as companion tools for hydrological modeling software in order to take advantage of R's features for data processing and visualization. The *swmmr* package developed by Leutnant et al. (2019) contains features to interface with the EPA Storm Water Management Model (SWMM) and includes functions to import/export files, initiate model simulations, and convert model files to/from GIS data. Likewise, the *RavenR* package (Chlumsky et al., 2022) provides utilities for processing inputs and outputs from the Raven Hydrological Modelling Framework, and the *geotopbricks* package (Cordano, 2024) includes features for working with files and simulation data from the GEOTOP distributed hydrological model. Other R packages take a different approach and re-implement existing hydrological models as R packages (Astagneau et al., 2021). For example, the *airGR* package developed by (Coron et al., 2017) includes various versions of the GR rainfall-runoff models, and the *HBV.IANIGLA* package developed by Toum et al. (2021) provides a modularized version of the popular Hydrologiska Byråns Vattenbalansavdelning (HBV) hydrological model.

The synthesis of hydrological models and R presents several challenges pertaining to interoperability. First, although efforts have been made to develop flexible hydrological modeling frameworks (Craig et al., 2020), most hydrological models have unique files with specific data formats and structures. Consequently, R packages and functions developed for one hydrological model may not necessarily be compatible with other models and file structures. Thus, there is an inherent need for R packages tailored to specific hydrological models. Another significant challenge in the integration of R and hydrological models is that the release of new versions of the R and/or modeling software can lead to unexpected results and behaviors or even complete loss of functionality. For example, R versions 3.X.X and 4.X.X use different toolchains, and in the case of the *swmmr* package, using different R versions on different systems lead to discrepancies when reading model result files (Leutnant, 2020). A common solution to address this issue is to develop separate R packages for predefined R/software versions. For

instance, the *rgrass* (Bivand, 2023) and *rQGIS* (Muenchow et al., 2017) families of R packages provide different packages when breaking changes occur. A final challenge in the interoperability of hydrological models and R is the presence of proprietary software. Some hydrological models, such as MIKE SHE (Hughes and Liu, 2008) or XPSWMM, must be operated through commercial software/platforms, and this imposes limits on how much the models can be integrated with R. In contrast, developing open-source tools for open-source hydrological models fosters the growth of the hydrological modeling community by promoting collaboration and innovation, increasing reproducibility, and eliminating financial barriers to access (Dile et al., 2016).

In this paper, we introduce HYPETools (Capell and Brendel, 2023), an open-source R package which was initially conceived as a companion package for the Hydrological Predictions for the Environment (HYPE) hydrological model (Lindström et al., 2010). HYPE is a widely used open-source, dynamic, semi-distributed hydrological model for simulating discharges, nutrients, and other substances. HYPE models have been used for a wide range of applications and domains such as (1) operational flood forecasting in Sweden (Girons Lopez et al., 2021), the Niger River Basin (Andersson et al., 2017), and other countries (e.g. Denmark, Greece, and Ireland), (2) climate change impact assessments in Europe and Sweden (e.g. Bartosova et al., 2021), the United Kingdom (Ockenden et al., 2017) and various other river drainage basins (Gelfan et al., 2017; Krysanova and Hattermann, 2017), (3) assessment of the effectiveness of pollution and nutrient mitigation strategies in England (Hankin et al., 2019) and the Baltic Sea drainage basin (Bartosova et al., 2019; Capell et al., 2021), (4) evaluating the impact of reservoirs on streamflow and sediment dynamics in Vietnam (Bui et al., 2023), and (5) hydropower production forecasting in Slovenia (Ogliari et al., 2020). Various HYPE models have also been developed for large-scale domains including the Hudson Bay Drainage Basin (Stadnyk et al., 2020), India (Pechlivanidis and Arheimer, 2015), Europe (Brendel et al., 2023; Hundedea et al., 2016), the Arctic basin, and the global domain (Arheimer et al., 2020). Developing and calibrating a new HYPE model can be challenging for those unfamiliar with the modeling software because HYPE does not provide a user interface and because the model input and setup files must adhere to certain formatting requirements. Thus, HYPETools was originally intended to provide functions for importing, manipulating, and exporting HYPE model files and performing basic data summarization. Since its initial development, HYPETools has evolved into a standalone package with additional features borne out of a basic need to quickly assess and visualize model dynamics, results, and performance in ways which were not readily available in other R packages. These unique HYPETools features allow users to take a rational gaze into hydrological models and datasets through tools such as interactive maps, plots for assessment of catchment properties and model performance, and apps for interactive data exploration. This paper describes the HYPETools implementation and presents a set of case studies demonstrating how HYPETools can be used to assist in both HYPE modeling and standalone (i.e. non-HYPE modeling) contexts.

In general, the HYPETools vision is to provide supporting functions for regular tasks which can then be used as building blocks in various hydrological modeling workflows and analyses. Features were chosen for inclusion in HYPETools according to needs identified by the HYPE modeling community, and users are encouraged to submit feature requests and contribute their own code. Overall, the features of the HYPETools package can be divided into five main categories: (1) file import/export, (2) data manipulation and conversion, (3) data summarization, (4) plotting and mapping, and (5) interactive data exploration. Functions in HYPETools are intended to be generic such that they can be used outside of a HYPE modeling context (e.g. with other hydrological models and/or workflows), provided that the function inputs are appropriately formatted. Furthermore, because it is an open-source package, people are free to adapt and modify the HYPETools code to better fit their needs and data formats/structures.

Table 1
HYPEtools dependencies.

R Package	Requirement
beepR	Optional
clipR	Mandatory
colorspace	Mandatory
data.table ($\geq 1.9.8$)	Mandatory
dplyr	Mandatory
DT	Optional
ggplot2	Mandatory
ggpubr	Mandatory
ggrepel	Mandatory
htmlwidgets	Optional
leaflet	Optional
leaflet.extras	Optional
lubridate	Mandatory
mapview	Optional
ncdf4	Mandatory
patchwork	Mandatory
pbapply	Mandatory
plotly	Optional
purrr	Mandatory
rlang	Mandatory
scales	Mandatory
sf	Optional
shiny	Optional
shinyalert	Optional
shinyFiles	Optional
shinyWidgets	Optional
stringr	Mandatory
tidyr	Mandatory
tidyselect	Mandatory
webshot	Optional
zoo	Mandatory

2. HYPEtools implementation

2.1. Architecture & dependencies

HYPEtools is implemented as a package developed entirely using the open-source R language. R was selected over alternatives, such as Python, to take advantage of R's advanced statistical functionality and exceptional data visualization features. The core HYPEtools features (See Section 0) are provided via 140+ exported functions. These exported functions are supported by several internal functions which, for example, extract file header information, fill data gaps, and prepare plot labels and positioning. R includes an object-oriented system called S3 which consists of classes that define object types and methods (e.g. print, summary, and plot) that are associated with particular classes (Wickham, 2019). These S3 methods are included in HYPEtools to provide summaries and descriptive statistics for various HYPEtools object classes. The HYPEtools documentation is produced using roxygen2 (Wickham et al., 2022). HYPEtools relies on several other R packages in order to provide its full suite of features (Table 1); however, the dependencies for certain HYPEtools visualization and interaction features are optional in order to reduce the package size and loading time for users who do not wish to use these features.

2.2. Installation

The HYPEtools package was developed to be run on any modern computer with R version 3.5 or greater and is compatible with HYPE modeling software version 5.0.0 or greater. HYPEtools is tested on a variety of Windows, Linux, and Mac OS distributions, and stable releases are available for installation from both CRAN (<https://cran.r-project.org/package=HYPEtools>) and conda-forge (<https://anaconda.org/package=HYPEtools>).

Table 2
Summary of common HYPE files.

File Name	Category	Contents	Data Format	Relevant HYPEtools Functions
allsim	Calibration File	Performance results (criteria and parameter values) of all runs during calibration	Tabular	<i>ReadAllsim</i>
basinoutput	Output File	Output variables for a single subbasin	Tabular	<i>AnnualRegime, ExtractFreq, PlotBasinOutput, PlotBasinSummary, ReadBasinOutput, AllDownstreamSubids, AllUpstreamSubids, DirectUpstreamSubids, PlotBasinOutput, PlotBasinSummary, PlotSubbasinRouting, ReadBranchData, SubidAttributeSummary, SumUpstreamArea, UpstreamGeoData, UpstreamGroupSLCClasses, WriteBranchData, ReadDescription</i>
BranchData	Model Setup File (Optional)	Bifurcations in the flow network	Tabular	
description	Model Setup File (Optional)	Names/IDs of land use, soil type, and crop types	List	<i>GroupSLCClasses, PlotBasinSummary, PlotPerformanceByAttribute, ReadGeoClass, SubidAttributeSummary, UpstreamGroupSLCClasses, WriteGeoClass, AllDownstreamSubids, AllUpstreamSubids, CleanSLCClasses, DirectUpstreamSubids, GroupSLCClasses, HeadwaterSubids, OutletNearObs, OutletSubids, PlotBasinOutput, PlotBasinSummary, PlotPerformanceByAttribute, PlotSubbasinRouting, ReadGeoData, RescaleSLCClasses, SortGeoData, SubidAttributeSummary, SumUpstreamArea, UpstreamGeoData, UpstreamGroupSLCClasses, WriteGeoData, ReadInfo, WriteInfo</i>
GeoClass	Model Setup File	HRU class definition	Tabular	
GeoData	Model Setup File	Subbasin characteristics and flow connections	Tabular	<i>PlotMapOutput, PlotMapPoints, PlotPerformanceByAttribute, ReadMapOutput, SubidAttributeSummary, VisualizeMapOutput, VisualizeMapPoints, MapRegionalSources, ReadMgmtData, WriteMgmtData</i>
info	Model Setup File	Model options and simulation settings	List	
mapoutput	Output File	Single output variable for all subbasins aggregated by time period	Tabular	<i>ReadOptPar, WriteOptPar</i>
MgmtData	Model Setup File (Optional)	Information about irrigation and water transfers	Tabular	
optpar	Calibration File	Settings and parameter ranges for calibration simulations	List	<i>ReadPar, WritePar</i>
par	Model Setup File	Model parameters	List	
Pobs	Observation Data	Precipitation forcing data	Tabular	<i>MergeObs, ReadObs, WriteObs</i>
Qobs	Observation Data	Discharge observations	Tabular	
subass	Output File	Model performance assessment for subbasins	Tabular	<i>EquallySpacedObs, MergeObs, OutletNearObs, ReadObs, WriteObs, PlotMapPoints, PlotPerformanceByAttribute, ReadSubass</i>
timeoutput	Output File	Time series of a single output variable for all subbasins	Tabular	
Tobs	Observation Data	Temperature forcing data	Tabular	<i>ExtractStats, ReadTimeOutput</i>
Xobs	Observation Data	Observations of evaluation variables (e.g. nutrient concentrations, lake water level, etc.)	Tabular	
				<i>MergeXobs, OutletNearObs, ReadXobs, WriteXobs</i>

[org/conda-forge/r-hypetools](https://github.com/rcapell/HYPETools)). The latest stable release can be installed in an R environment using R's `install.packages("HYPETools")` command or within a Conda environment using Conda's `conda install -c conda-forge r-hypetools` command. HYPETools is under constant development, and the development version is hosted on GitHub (<https://github.com/rcapell/HYPETools>) where instructions are provided to install the development version and users are encouraged to contribute, provide feedback, and submit feature requests. New HYPETools versions are released to CRAN and conda-forge every few months as new features become available.

3. HYPE hydrological model

3.1. Model description

The HYPE model is an open source, dynamic, semi-distributed hydrological model which integrates fluxes of water, nutrients, and other substances. HYPE is developed and maintained by the Swedish Meteorological and Hydrological Institute (SMHI), and model applications include operational forecasting, climate-change impact assessments, scenario analyses, and scientific research from the catchment to global scales. The spatial representation in HYPE is based on subbasins characterized according to hydrological response units (HRUs) called "Soil type – Land use Combination" (SLC) classes, which represent unique combinations of physical properties such as land use, soil type, elevation, soil depth, and/or crop/vegetation cover. HYPE requires input precipitation and air temperature forcing data, and models are typically run at the daily time step. However, HYPE also supports simulations at sub-daily time steps. Substances that can be simulated in HYPE include nitrogen, phosphorus, organic carbon, silica, sediment, water temperature, and a general constituent (tracer). Detailed descriptions of the HYPE model and its routines can be found in SMHI (2023).

3.2. Model files

There are four primary types of HYPE files: model setup files, observation data files, output files, and calibration files. However, HYPE also has several special files for performing data assimilation and specifying subbasin water balances. The majority of HYPE files are text files with a `"*.txt"` extension containing data in a tabular or list format. A full summary of all HYPE files can be found at http://hype.smhi.net/wiki/doku.php?id=start:hype_file_reference. Table 2 provides an overview of the most common HYPE files and their relevant HYPETools functions. HYPETools imports tabular HYPE files as data frames while HYPE files with list formats are imported as named lists.

4. HYPETools package features

HYPETools includes 140+ exported functions. Here, we highlight some HYPETools features and functionalities related to file import/export, data manipulation and conversion, data summarization, plotting and mapping, and interactive data exploration. Although these functions were developed with the HYPE model file structure and formats in mind, we believe that the functions can be valuable for people working with other hydrological models and/or data structures because they can reformat their data into data frames and lists in the HYPE file structure used by HYPETools to use the functions directly, or they can adapt the open source code to work with their own data formats. Example code for each highlighted function is provided in the Supplementary Material.

4.1. Help

There are several forms of help resources available for HYPETools. All core functions within HYPETools include associated help documentation and replicable examples which can be accessed through R's `help()` function and `"?"` operator. In addition, HYPETools is distributed with a

bundled demo HYPE model and example output files. These demo files and outputs are linked to the HYPETools help documentation so that users can test the various HYPETools functions using real data. Furthermore, six vignettes are provided to introduce users to the tools and become familiar with the functions; these vignettes can be accessed by entering `browseVignettes("HYPETools")` in the R Console or online at <https://cran.r-project.org/web/packages/HYPETools/vignettes/>.

Finally, HYPETools includes the `VariableInfo()` and `VariableSearch()` functions to help users lookup information (e.g. name, units, description, etc.) for a specific HYPE variable ID or query HYPE variable information for a given search term. This information is scraped from the HYPE wiki (SMHI, 2024) and stored internally within HYPETools each time a new package version is released, which ensures that the variable metadata can be accessed even if no internet connection is available.

4.2. File import/export

To support workflows and analyses involving HYPE model files, HYPETools includes dedicated functions to import HYPE files into R as well as complementary functions to export data from R into HYPE files with standardized formatting. The names for the import and export functions follow the pattern `Read<Filename>()` and `Write<Filename>()`, respectively. HYPETools reads and writes tabular HYPE files (See Table 2) using the `data.table` R package (Dowle and Srinivasan, 2023), which features parallelization to reduce the time required for file import and export. This is especially useful when working with HYPE observation data files, which can be several gigabytes in size. When reading certain tabular files, such as `GeoData` and `Xobs`, HYPETools provides `S3` class attributes for relevant information such as the subbasin `SUBID`, river length, and observation variable; this allows users to quickly access summaries and descriptive statistics using, for example, R's `summary()` function. For non-tabular HYPE model files (See Table 2), HYPETools reads the data into nested lists that can be queried by e.g. code or parameter name. HYPETools also offers support for reading HYPE time output files in `NetCDF` format. The export functions included in HYPETools ensure that exported files adhere to the HYPE model standardized formats with consideration for things such as list/column separators, decimal separator, number of digits, date/time format, and missing/empty entries. Without HYPETools, modelers would need to manually create HYPE model files using e.g. a text editor or spreadsheet program and manually check that the file formatting is correct. Thus, HYPETools encourages efficient, reproducible model development and modification.

4.3. Data manipulation & conversion

Functions are also provided in HYPETools to facilitate frequently needed data manipulations and conversions when working with hydrological models and datasets. To assist with the merging of time series observations for model forcing or calibration, HYPETools includes two functions, `MergeObs()` and `MergeXobs()`, to merge data frames with time series observations (typically imported HYPE `Obs` or `Xobs` files) while handling overlapping time periods, data gaps, and common column names. HYPETools also offers a related function, `EquallySpacedObs()`, for converting a data frame of observations with irregular time steps to a data frame of equally spaced time series with codes for missing observations. Another useful conversion function provided by HYPETools is `ConvertDischarge()`, which allows users to convert from volumetric discharges to unit area discharge and vice versa. A final noteworthy function is `SortGeoData()`, which allows users to sort subbasins in downstream order including bifurcations when provided a data frame with columns of subbasin IDs and their corresponding downstream subbasins (typically an imported HYPE `GeoData` file) and a data frame containing columns of source subbasin IDs and the corresponding receiving subbasin IDs for any bifurcations within the model domain (typically an imported HYPE `BranchData` file).

Another useful group of HYPEtools functions are those which were designed to assist with the partitioning of spatial attributes within model subbasins. During model development, the automatic characterization of spatial subbasin attributes (e.g. land use and soil type) using common GIS functions can result in subbasins which may contain very small fractions of some HRU classes (See Section 3). These small fractions are often negligible, and it can be desirable to simplify the model setup by eliminating them. HYPEtools provides the *CleanSLCClasses()* function to allow users to reassign these small HRU fractions to the predominant HRU classes for model subbasins according to user-provided area rules and thresholds. It may also be desirable to adjust the HRU distribution within a subbasin such that e.g. the fraction of water area matches the values from a separate lake attribute dataset. For these situations, HYPEtools includes a *RescaleSLCClasses()* function to re-scale the HRU fractions for each subbasin to a new target sum for all classes. Without HYPEtools, these data manipulations and conversions would be extremely tedious and time-consuming. Furthermore, these types of manipulations/conversions must be performed separately for each individual HYPE model. Thus, HYPEtools can greatly reduce the time spent performing these frequently needed data manipulation and conversion tasks.

4.4. Data summarization

4.4.1. Subbasin routing

HYPEtools also makes it easy to extract information on the subbasin routing for a model domain. With a data frame containing columns of subbasin IDs and their corresponding downstream subbasins (typically an imported HYPE GeoData file), users can apply the *AllUpstreamSubids()* and *AllDownstreamSubids()* functions to identify all upstream/downstream subbasins for a given subbasin. These functions can be useful for e.g. identifying which subbasins are affected by a given subbasin or for identifying subbasins for creating a submodel of the full domain. Users can also identify the subbasin(s) immediately upstream of a specific subbasin using the *DirectUpstreamSubids()* function. If the model domain contains bifurcations, then users can also pass a data frame containing columns of source subbasin IDs and the corresponding receiving subbasin IDs (typically an imported HYPE BranchData file) to the functions to account for these branches. HYPEtools also provides the *HeadwaterSubids()* and *OutletSubids()* functions to list the IDs for headwater subbasins and terminal subbasins (including sinks, outlets to coasts, etc.) in the model domain, respectively. Finally, the *OutletNearObs()* function is included to help users find observation stations located close to specified outlet subbasins using upstream area fractions as proximity thresholds. This can be especially useful, for example, when selecting calibration stations for large-scale models when calibrating to model performance for export to coastal areas (Section 5.2.1). Without HYPEtools it can be challenging to extract advanced routing relationships for hydrological models because routing information is typically provided only in terms of upstream-downstream pairs. To identify all subbasins upstream of a point, for example, one would need to search the individual upstream-downstream pairs to identify the subbasin(s) immediately upstream of the desired point and then search the upstream-downstream pairs to identify the subbasin(s) immediately upstream of those points, and so on. Without HYPEtools, this task would be prohibitively time-consuming for a large-scale model with many subbasins.

4.4.2. Time series

HYPEtools offers two functions, *ExtractFreq()* and *ExtractStats()*, for extracting aggregated statistics from time series datasets. *ExtractFreq()* calculates quantiles from time series data and is useful for exceedance probability analyses and the creation of flow duration curves. The *ExtractFreq()* function accepts input as a numeric vector (i.e. a single time series) or an all-numeric data frame containing columns with individual time series and returns a data frame with probabilities in the

first column and quantiles for the respective time series in the following columns. In contrast, *ExtractStats()* calculates summary statistics for time series data grouped according to various aggregation periods. The input to *ExtractStats()* is a data frame with a column of dates and columns for individual time series, and users can specify if the data should be aggregated by year, winter/summer half-years, winter/spring/summer/fall seasons, or month. The function then applies user defined functions — such as *min()*, *mean()*, or *max()* — across each time series and aggregation period and returns a data frame with columns for the starting dates of each aggregation period, a descriptive aggregation period name, and the aggregated results (e.g. minimum, mean, maximum) for each time series/aggregation period combination. The outputs from both *ExtractFreq()* and *ExtractStats()* include attributes for additional summary information such as the number of observations per time series, time step, and aggregation period.

HYPEtools also includes the *AnnualRegime()* function, which aggregates and summarizes a multi-year time series dataset by calendar day to produce a single representative annual time series (i.e. a one-year time series with summary values for each calendar day). This can be useful for analyses such as assessing the agreement between simulated and observed time series, comparing measurements from different sampling campaigns for a location, checking if the annual regimes for a location have shifted between different time periods, or determining if two locations have similar seasonal patterns. The input to the function is a data frame with a column of dates and columns of equally-spaced time series, and the output is a list of data frames containing the annual regimes — long-term arithmetic means, sums, medians, minima and maxima, and 5%, 25%, 75%, and 95% percentiles — for each respective time series. The output of *AnnualRegime()* also includes attributes for the time step and aggregation period of the results.

4.4.3. Catchment characteristics

Several functions are provided in HYPEtools for aggregating and summarizing catchment characteristics for both individual subbasins and their upstream areas. This information can be helpful for determining, for example, what processes and factors may influence model biases for a certain location or how water quality relates to land use. First, *SumUpstreamArea()* can be used to calculate the total upstream drainage area for individual model subbasins or all subbasins in the model domain. Then, *GroupSLCClasses()* allows users to calculate the land use, soil type, and crop fractions for each individual model subbasin, and *UpstreamGroupSLCClasses()* allows users to calculate these fractions based on the entire upstream area for each subbasin. Next, *UpstreamGeoData()* calculates upstream sums or averages of selected variables such as: total area and river length or mean elevation, slope, latitude, and longitude. Inputs to these functions include (1) a data frame containing columns with subbasin IDs and areas (typically an imported HYPE GeoData file), (2) a data frame containing columns with HRUs and corresponding land use, soil, and crop IDs (typically an imported HYPE GeoClass file), and, if bifurcations exist in the model domain, (3) a data frame containing columns of source subbasin IDs and the corresponding receiving subbasin IDs (typically an imported HYPE BranchData file). A final noteworthy function is *SubidAttributeSummary()*, which combines the *UpstreamGroupSLCClasses()* and *UpstreamGeoData()* functions to prepare a data frame of summary attributes such as the upstream land use, soil, and crop fractions or the upstream average elevation and slope. These summary attributes can then be used for plotting and mapping, assessment of model biases, and other similar analyses.

4.5. Plotting & mapping

4.5.1. Subbasin model performance evaluation

Visual inspection of model outputs is an essential step during model calibration and evaluation. To help with this, HYPEtools includes two functions, *PlotBasinSummary()* and *PlotBasinOutput()*, which generate

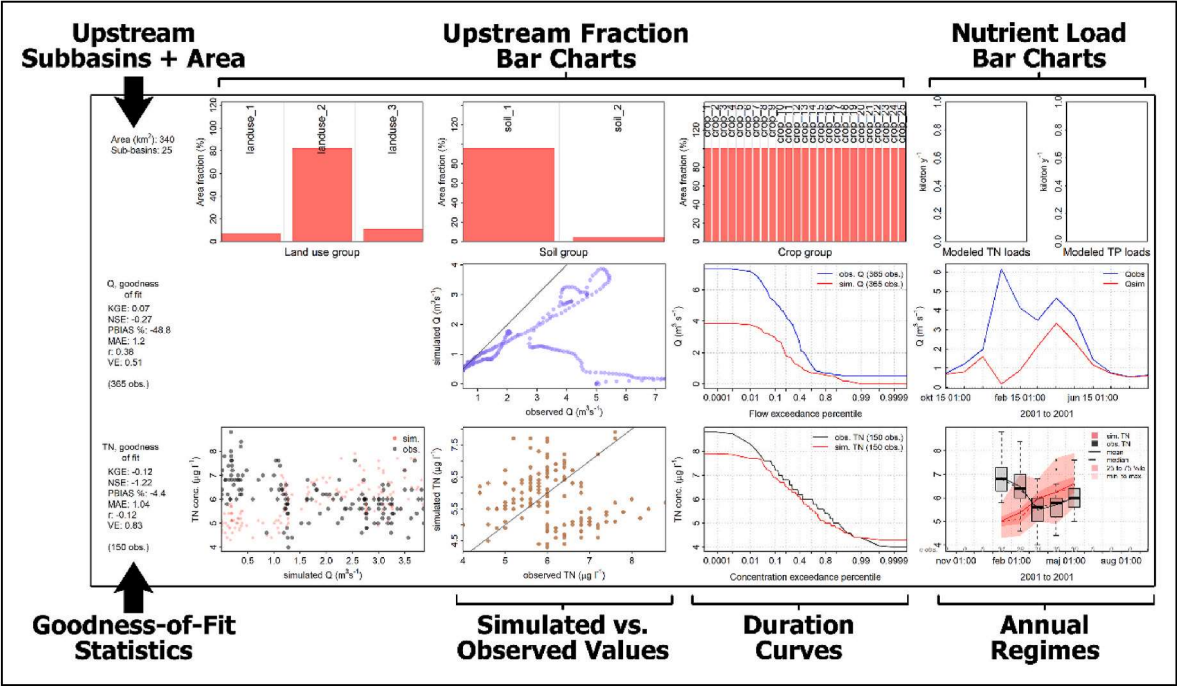


Fig. 1. PlotBasinSummary example.

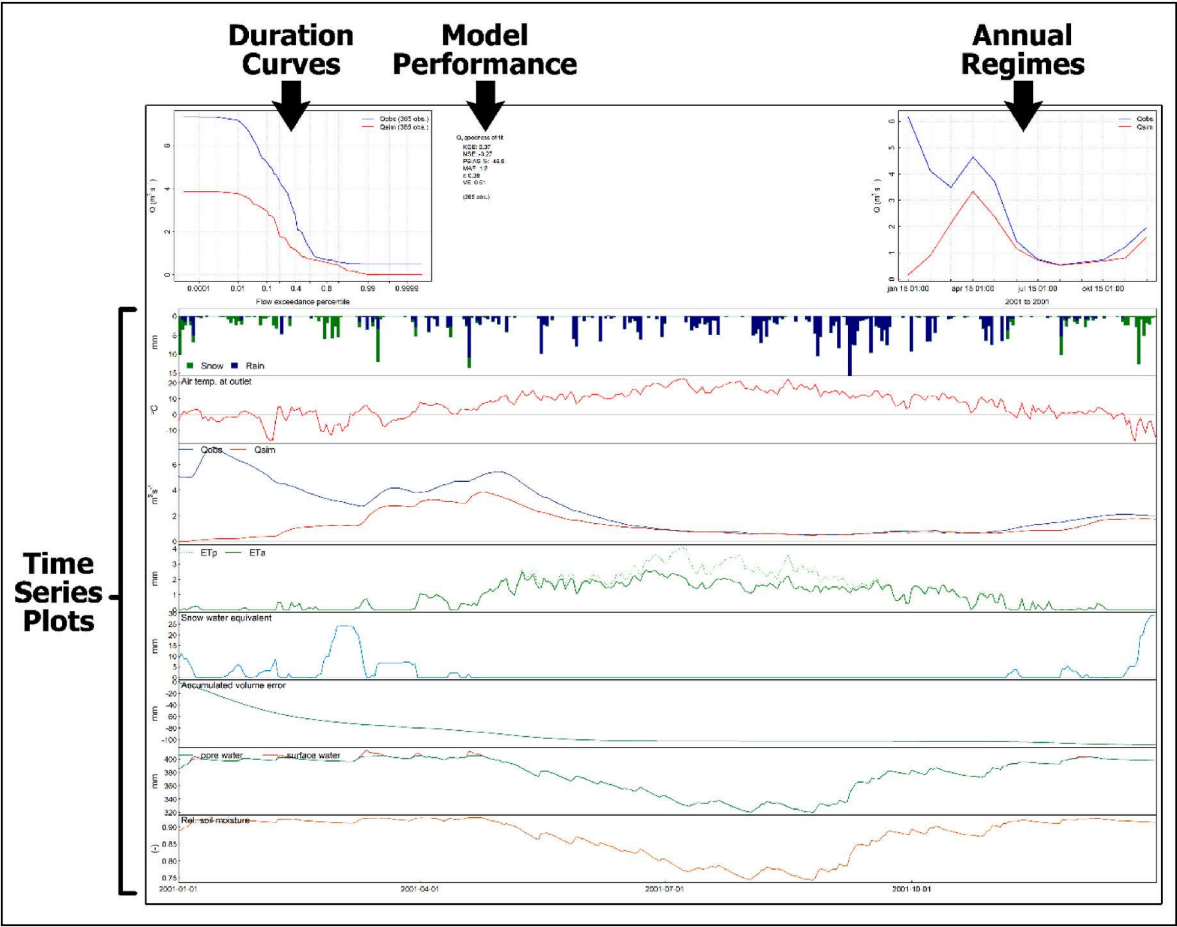


Fig. 2. PlotBasinOutput example.

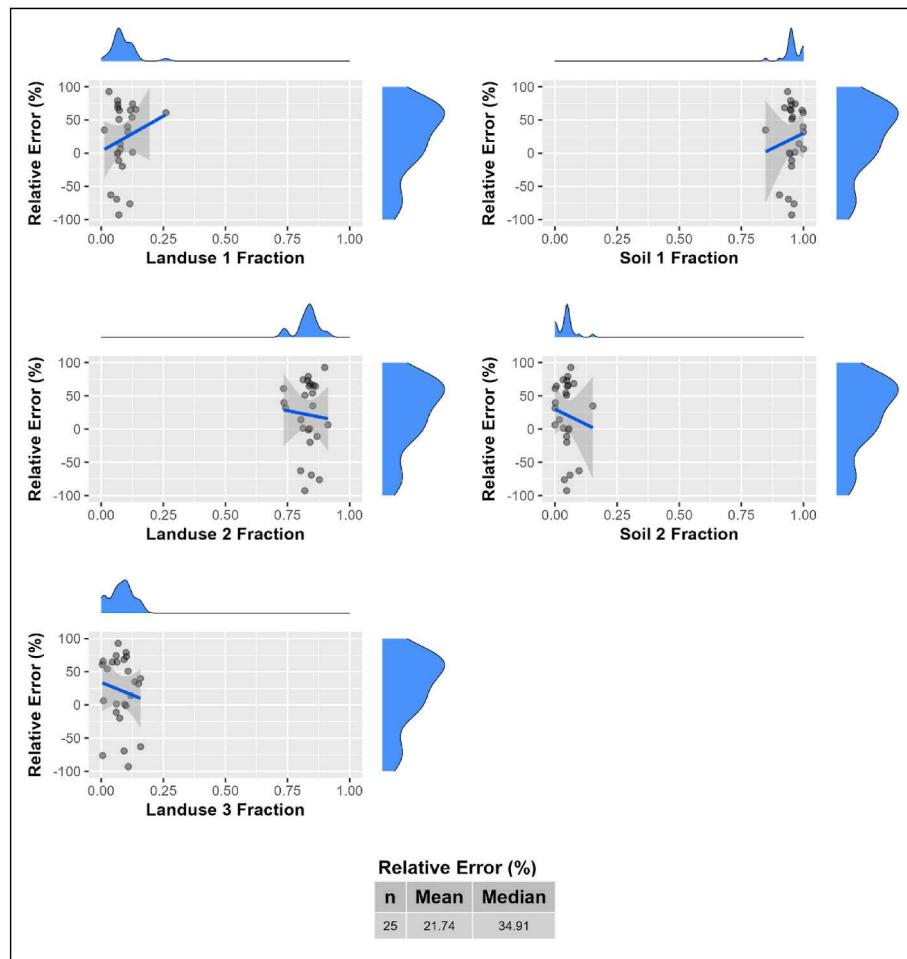


Fig. 3. PlotPerformanceByAttribute example.

grouped summary plots which can be used to assess catchment properties and model performance for a chosen subbasin. These summary plots consist of a variety of plotted panels for various model output variables. The inputs to both functions are (1) a data frame with a column of dates/times and columns for model output variable values (typically an imported HYPE basinoutput file), and (2) a data frame with columns for subbasin ID, subbasin area, and downstream subbasin ID (typically an imported HYPE GeoData file). A total of 36 unique HYPE output variable are recognized by the functions. By default, the functions will generate plots for all panels and available model variables, and any missing variables will be automatically omitted. However, users can adjust which panels and variables are plotted using function arguments. The summary plots can either be viewed in RStudio's Viewer pane or exported to a PNG or PDF file for external use.

The *PlotBasinSummary()* output (Fig. 1) provides an overview of the catchment properties (including upstream area) and model performance for discharge and concentrations of nutrient forms, sediments, and tracers for a selected subbasin. On the top row of the output plot, a panel summarizes the number of upstream subbasins and total upstream subbasin area, and bar charts present the upstream-averaged land use, soil type, and crop group fractions. These plots are useful for identifying the dominant HRU classes within the subbasin and therefore which land use and/or soil parameters may have the greatest impact on the model performance for the subbasin. The generic land use/soil/crop labels can be replaced with custom names by providing a named list of the custom labels (typically an imported HYPE description file) as an argument to the *PlotBasinSummary()* function. In addition, bar charts are included to summarize the modeled nutrient loads at the subbasin outlet and, if the

necessary variables are available in the input data frame, plots will be included for the total upstream gross nutrient loads from point sources and rural households. On the bottom row of the output plot, a panel presents a summary of goodness-of-fit statistics for discharge and concentrations in the subbasin which includes the Kling-Gupta Efficiency (KGE), Nash-Sutcliffe Efficiency (NSE), percent bias (PBIAS), mean absolute error (MAE), Pearson correlation coefficient (r), volumetric efficiency (VE), and number of observations. The bottom row also includes plots to summarize the relationships between the simulated vs. observed discharges and concentrations of nutrients, sediments, and tracers, respectively, as well as the simulated vs. observed concentration-discharge relationships. These plots are helpful for e.g. assessing model biases and errors for different parts of the flow regime. For further evaluation of the modeled dynamics, pairwise simulated and observed duration curves and annual regime plots are included for both discharge and concentrations. Finally, if the necessary variables are available in the input data frame, then plots are generated for the ratios of inorganic nitrogen to total nitrogen and soluble phosphorus to total phosphorus. It is worth noting that the majority of the individual plots/panels can be generated separately using HYPETOOLS functions such as *BarPlotUpstreamClasses()*, *PlotAnnualRegime()*, *PlotSimObsRegime()*, and *PlotDurationCurve()*. An example use case of the *PlotBasinSummary()* function is included in Section 5.2.3.

The *PlotBasinOutput()* output (Fig. 2) provides a quick and comprehensive overview of model dynamics for a subbasin of interest. These plots are useful for e.g. checking the water balance and for inspecting model performance during manual calibration. At the top of the grouped summary plot, panels present pairwise simulated and observed flow

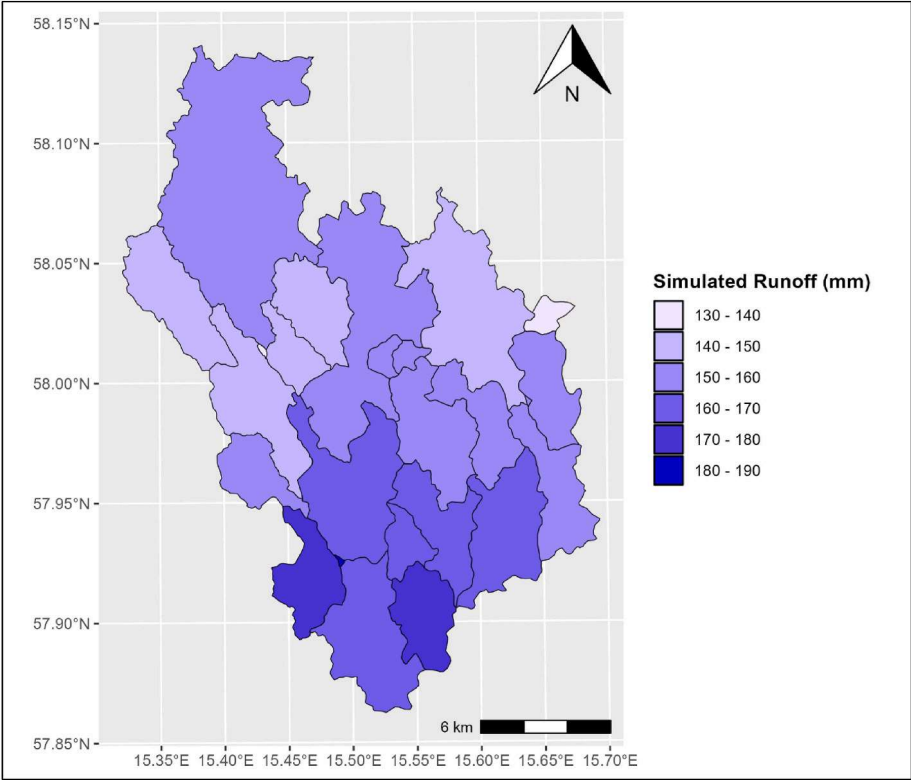


Fig. 4. PlotMapOutput example (static map).

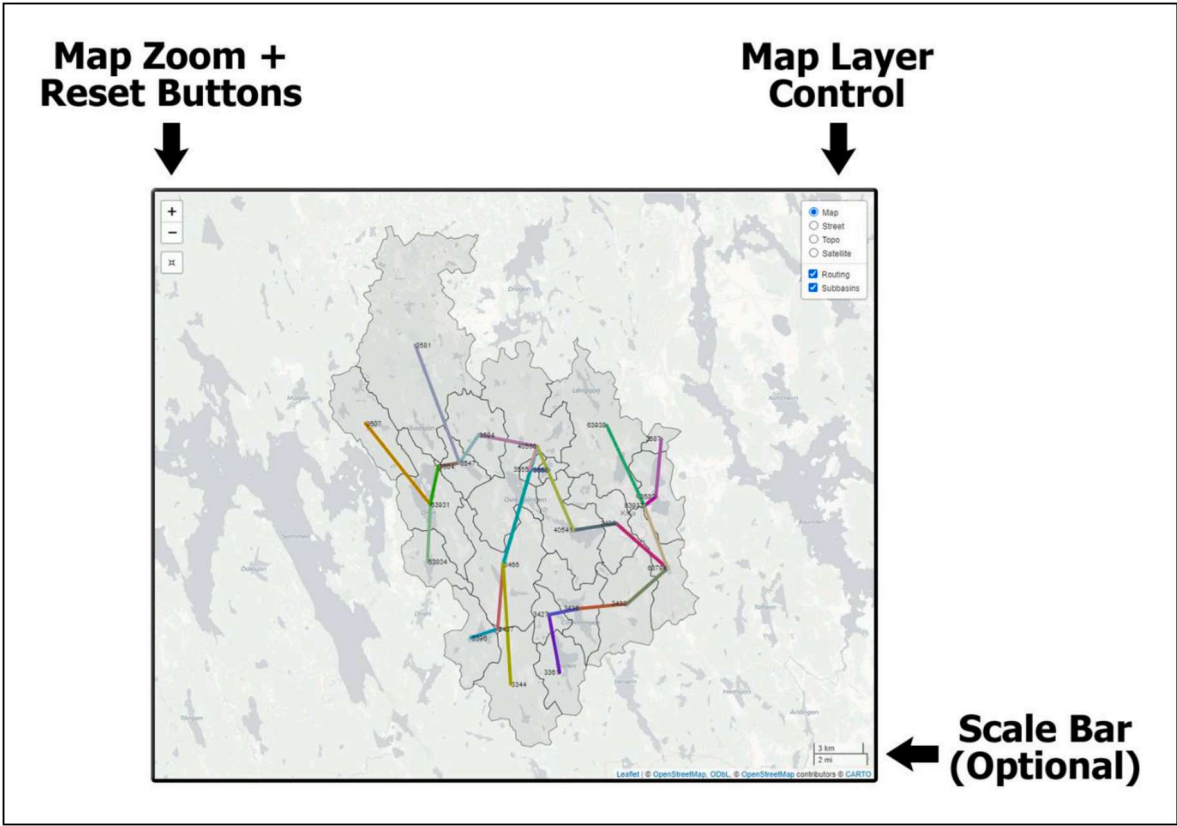


Fig. 5. PlotSubbasinRouting example.

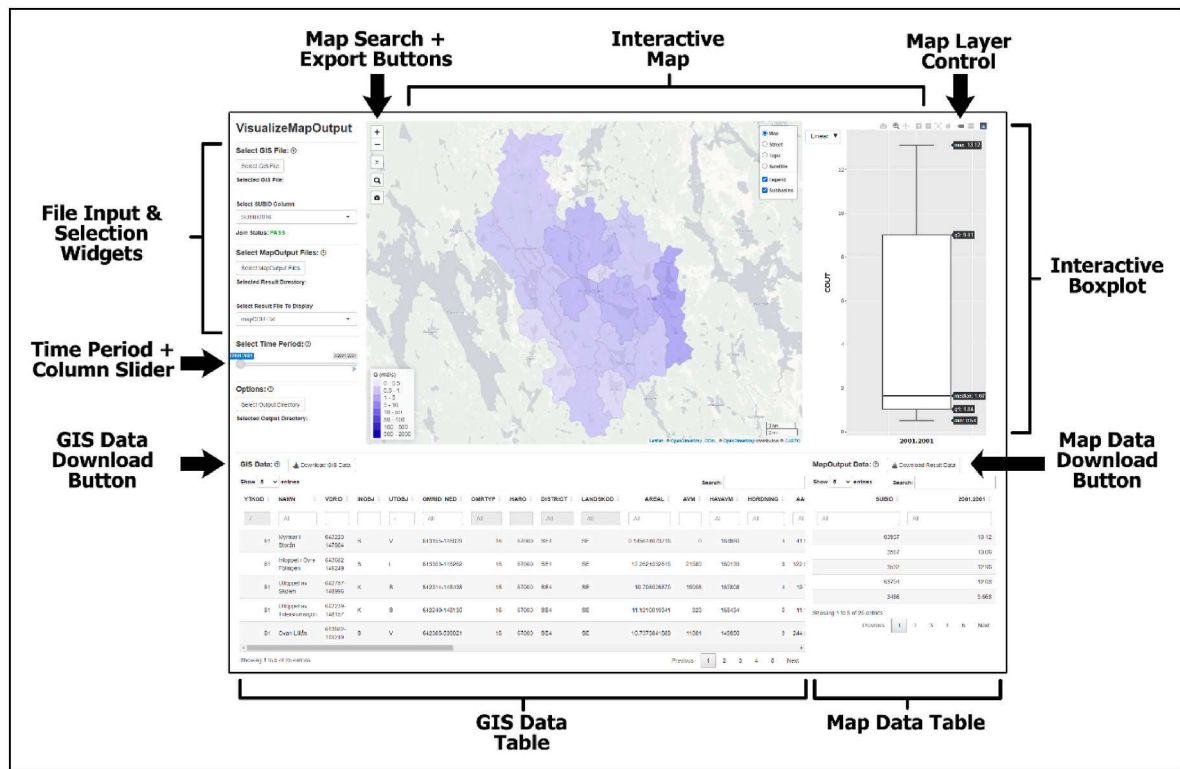


Fig. 6. VisualizeMapOutput Shiny app interface.

duration curves and annual flow regime plots as well as a table summarizing the number of discharge observations and the model performance for discharge in the subbasin according to the following goodness-of-fit statistics: KGE, NSE, PBIAS, MAE, Pearson correlation coefficient, and VE. If desired, users can also generate the duration curve and annual regime plots separately using the *PlotDurationCurve()* and *PlotAnnualRegime()* functions, respectively. Beneath these panels, the *PlotBasinOutput()* output provides a suite of time series plots which can be used for assessing dynamics and evaluating errors over time. Noteworthy plots include those for the amount/distribution of snow and rainfall, observed subbasin air temperature, simulated vs. observed discharge, simulated actual vs. potential evapotranspiration, snow water equivalent, accumulated volume error, pore vs. surface water, and relative soil moisture.

4.5.2. Model performance by catchment attribute

To aid in identifying relationships between model performance and subbasin catchment attributes, HYPEtools includes the *PlotPerformanceByAttribute()* function. This function requires two primary inputs: (1) a data frame with columns with subbasin ID and model performance statistics (typically an imported HYPE subbasin file) and (2) a data frame containing subbasin ID and catchment attributes, typically an output from the *SubidAttributeSummary()* function (Section 4.4.3). Using this data, the *PlotPerformanceByAttribute()* function generates scatterplots with the model performance metrics — such as relative error, NSE, and KGE — plotted on the y-axis and the catchment characteristics — such as land use fractions, upstream area, mean slope, etc. — plotted on the x-axis (Fig. 3). These sorts of plots are useful for not only determining if, for example, model biases are correlated with the fraction of certain land use or soil types, but also for comparing the relative fractions of the different model land use and soil types. Marginal density plots can be added to visualize the distribution of the data points along each axis, and trendlines can be added to convey patterns. In addition, a table can be included below the plot to present summary statistics such as n, mean, median, min, and max. To facilitate comparisons between e.g. different

regions or modeled substances (Section 5.2.2), the data points can be also grouped and symbolized separately.

4.5.3. Spatial patterns

HYPEtools offers the *PlotMapOutput()* and *PlotMapPoints()* functions to easily produce maps of subbasin polygons or points, respectively, symbolized according to e.g. model outputs or performance statistics. These maps are helpful for identifying spatial patterns in model outputs (Fig. 4) and model performance across the model domain. The functions provide “pretty” scale discretizations and default color ramps including sequential color ramps with less saturated colors for smaller values and more saturated values for higher values, suitable for e.g. concentration or volume ranges, and diverging color ramps suitable for calculated differences, e.g. between two model runs. In addition, users have the option of generating static maps using the ggplot2 R package (Wickham, 2016) or interactive maps using the leaflet R package (Cheng et al., 2023). The interactive maps allow users to pan/zoom the map and include features to select different basemaps, search for and highlight map features, and view labels when hovering over map features. Inputs to the functions are (1) a GIS file (typically a GeoPackage or Shapefile) containing a column with subbasin ID and (2) a tabular data file (typically an imported HYPE mapoutput or subbasin file) containing columns for subbasin ID and the e.g. temporally aggregated model results or model performance statistics that should be mapped. One could also create these types of maps using GIS software, but HYPEtools makes the process more efficient because one does not need to manually specify e.g. how to join the GIS data to the model outputs or the color ramps and intervals.

4.5.4. Subbasin connections

HYPEtools includes two functions, *PlotSubbasinRouting()* and *MapRegionalSources()*, to generate interactive maps showing the various connections between the model subbasins. Whereas *PlotSubbasinRouting()* produces a map with lines indicating the routing of flow (including bifurcations) between the different model subbasins (Fig. 5),

MapRegionalSources() creates a map with lines representing the regional connections (e.g. irrigation, water transfers, etc.) between source and target subbasins. These functions make it easy to, for example, perform visual inspection of model delineations or gain an understanding of the important connections between regions (Section 5.1.2). Inputs to the functions are (1) a GIS data file providing subbasin locations as points or polygons with an attribute for subbasin ID, (2) a data frame containing columns with the IDs of the source and target/downstream subbasins (Typically an imported HYPE GeoData or MgmtData file), and, if bifurcations are present in the model domain, (3) a data frame containing columns of source subbasin IDs and the corresponding receiving subbasin IDs (typically an imported HYPE BranchData file). The interactive mapping is handled by the leaflet R package (Cheng et al., 2023), and each connection line is assigned a random color using the HYPEtools *distinctColorPalette()* function. Users can pan/zoom the maps, and layer control panes are included to select between four different basemap options and to toggle on/off the routing lines and subbasin labels. The interactive output maps can be viewed within RStudio's Viewer pane or exported as self-contained HTML files for distribution. Because the different connection lines are each rendered separately, interactive mapping performance is best for smaller model domains with fewer connections. However, an option is included to export the maps as static image files (i.e. without the interactive features).

4.6. Interactive data exploration

In order to eliminate the constant need for adapting and rewriting code to perform repeated quick analyses for different models and/or outputs, HYPEtools provides a suite of in-built Shiny apps for interactive analysis and visualization. The Shiny app framework allows developers to construct web apps using entirely R code and provides a collection of pre-built input and output widgets for controlling apps and displaying outputs. In addition, the Shiny framework utilizes a reactive programming model which updates outputs instantly as app inputs/controls are modified, eliminating the need for messy event handling code (Chang et al., 2023). The HYPEtools Shiny apps can be initialized by calling a function in R upon which the apps can be viewed within either an RStudio environment or in the web browser of the user's choice.

HYPEtools currently contains two Shiny apps, "VisualizeMapOutput" and "VisualizeMapPoints", which can be initialized by calling the *VisualizeMapOutput()* and *VisualizeMapPoints()* functions, respectively. At their core, these apps are interactive versions of the *PlotMapOutput()* and *PlotMapPoints()* functions (Section 4.5.2) and allow users to visualize model outputs and performance in a map-based interface. However, the true added value of the Shiny apps is that they couple interactive mapping with sortable/filterable tables and interactive plots. This allows users to quickly explore their data to identify spatial trends, outliers, and relationships with catchment characteristics (e.g. area or soil/land use fractions). Without the interactive Shiny apps, data exploration is less intuitive and effective because one needs to, for example, generate a large number of plots and perform many separate comparisons. Inputs to the app are (1) a GIS file (typically a GeoPackage or Shapefile) containing a column with subbasin ID and (2) tabular data files (typically imported HYPE mapoutput or subass files) containing columns for subbasin ID and the data that should be mapped and plotted (typically temporally aggregated model results or model performance statistics). The apps link these two input files together by the subbasin ID to determine the spatial mapping for the tabular data. For convenience, optional input arguments can be passed when calling the app functions to preload the app with certain desired data sets and settings. An example use case of the Shiny apps included in Section 5.1.1.

The HYPEtools Shiny app interfaces consist of several different interactive components (Fig. 6). On the left edge of the apps, sidebar menus provide widgets to upload the input files and select the data series to plot. In addition, a slider input allows users to select the column in the tabular data files that should be mapped and plotted (typically different

time periods or model performance statistics). Users can even click a button to animate through all columns so they can see, for example, how the spatial patterns change over time. In the center of the app, an interactive Leaflet map displays the GIS data symbolized according to the selected data values, and default color palettes are provided for certain HYPE model variables. Users can hover their mouse cursor over a map feature to view a label with the feature ID and its corresponding data value. A search bar allows users to search for and highlight specific map features, and a button is provided to export the current map view to an image file. On the right side of the interface, an interactive Plotly boxplot shows the distribution of the mapped values, and users can hover over the plot to display labels indicating the quantile, median, and min/max values. A button is also included to toggle the display between a continuous y-axis and a log-scale y-axis. At the bottom of the interface, two interactive data tables present the imported GIS data and tabular map data. The columns in the data tables can be sorted and filtered, and selections made within the GIS data table will subset and update the data displayed in the map and boxplot. Download buttons are included to export the data in the tables to CSV files. Finally, help buttons are provided throughout the app to provide users with additional information on the various app inputs, controls, and outputs.

5. Case studies

The purpose of this section is to demonstrate several example applications of the HYPEtools package for various workflows and analyses. Although HYPEtools was initially developed as a support tool for working with the HYPE hydrological model, we reiterate that the packages functions are intended to be generic such that they can be used outside of a HYPE modeling context (e.g. with other hydrological models and/or workflows) and that the HYPEtools source-code is publicly available to be tailored to different data formats and structures as needed (Capell and Brendel, 2023). Thus, two case studies are included to highlight how HYPEtools can simplify workflows and analyses within both HYPE and non-HYPE applications. First, a case study in South Africa demonstrates how HYPEtools can be used for due diligence data exploration and describes the identification of observation stations for a location and period of interest and the mapping of inter-basin water transfers. Then, a case study in the Baltic Sea drainage basin illustrates how HYPEtools can contribute to a typical HYPE model calibration and evaluation workflow.

5.1. South Africa

A first step of any research project involving observational data is to identify and map the observational data that exists and to assess the quantity and quality of the available data. In South Africa, the Department of Water and Sanitation (DWS) maintains a portal for accessing verified data for hydrological and meteorological observation stations across the country (https://www.dws.gov.za/Hydrology/Verified/hy_main.aspx). Data is available for over 2800 observations stations in 22 different drainage regions, and stations are categorized into eight different types: canal, dam, eye, lake/pan, meteorological, pipeline, reservoir, and river. The portal provides tables of station metadata with columns for station ID, station description, latitude, longitude, and the date range of available data. For canal, river, and pipeline stations, the portal also provides information on the location of any downstream connections. In general, downloading data from the portal is rather tedious as one can only access data for one drainage region and station type at a time and there are limits on the amount of data one may retrieve with one query. Furthermore, the portal does not include any mapping features, so it is difficult to determine exactly what data is available for where and when without clicking through several menus. Several research projects conducted by SMHI were interested in using data from the portal, but there was a need to identify what data was available for specific regions and time periods (Section 5.1.1). In

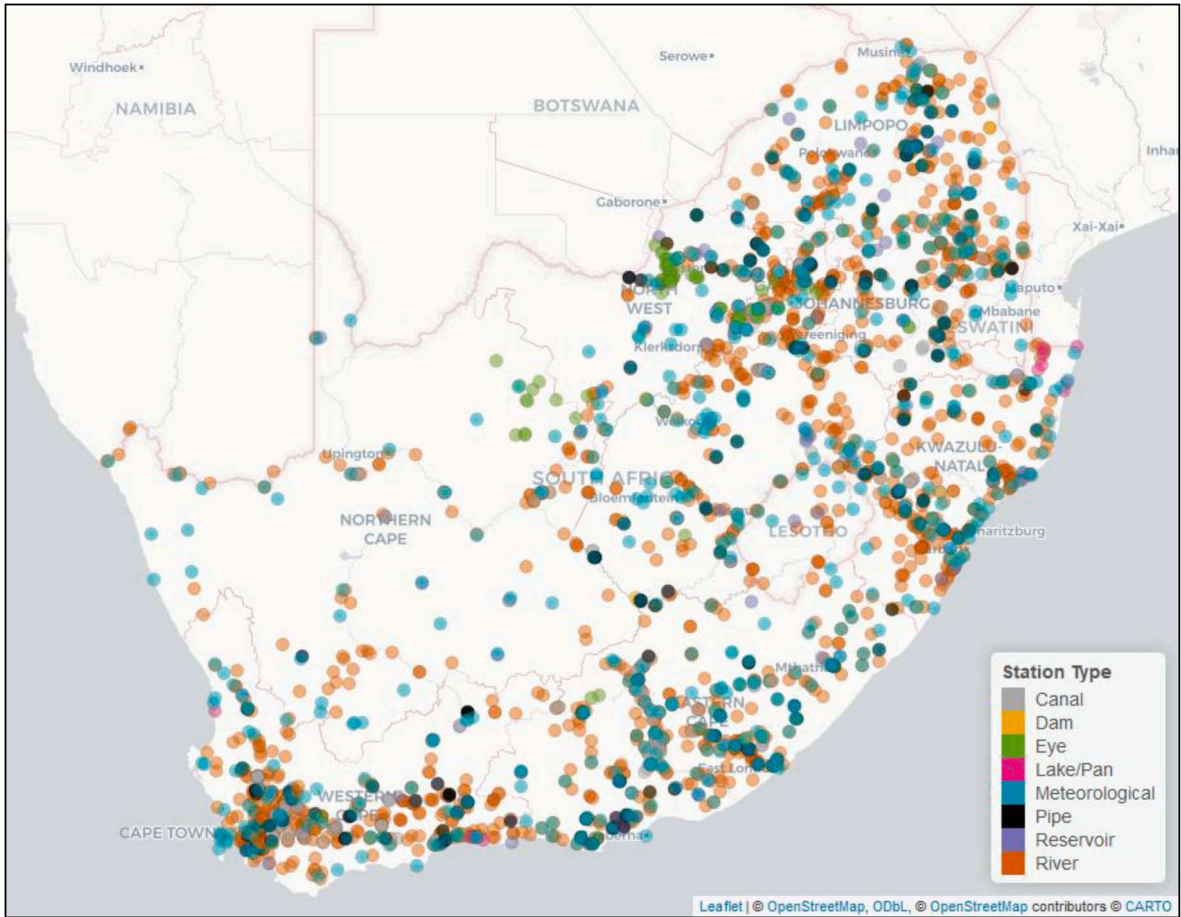


Fig. 7. South Africa observation stations by station type.

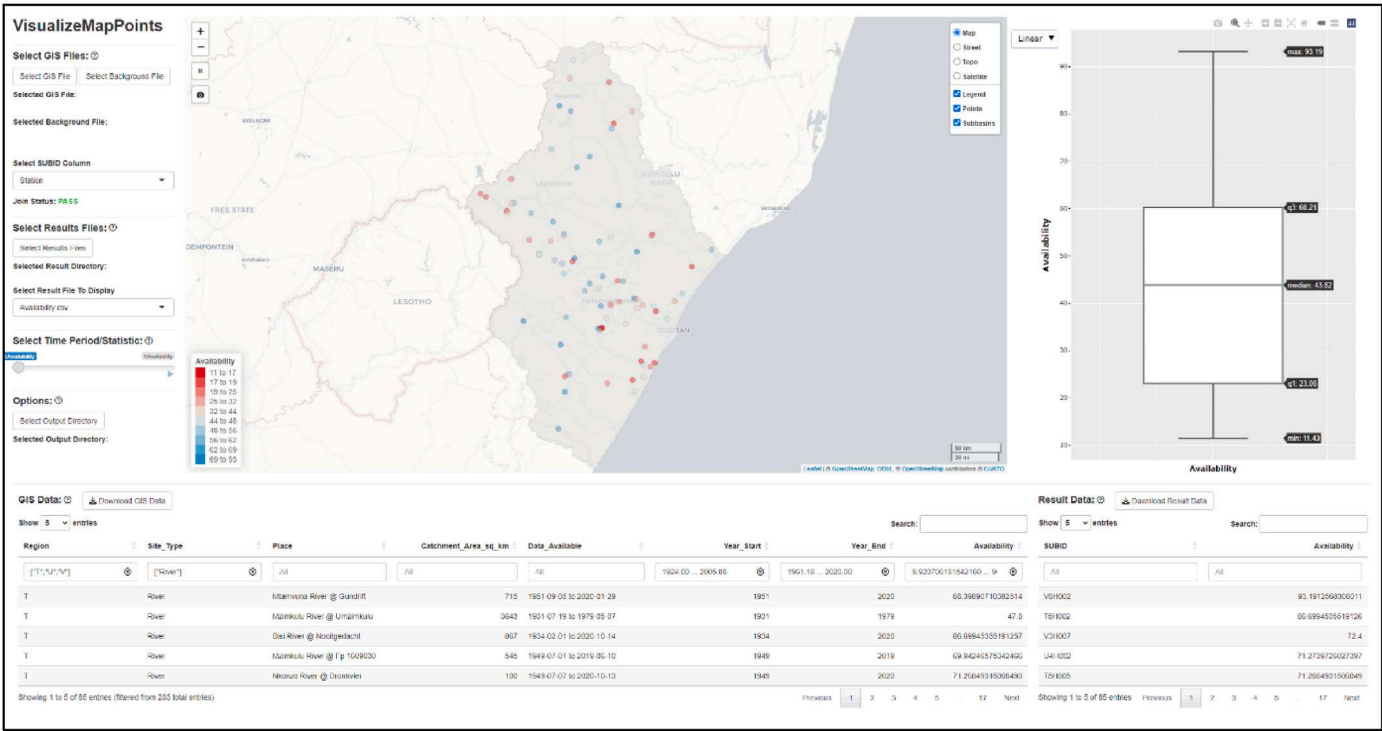


Fig. 8. VisualizeMapPoints app showing river observation stations in the uMgeni drainage region (displayed as a gray background layer) with data available from 1961 to 2006 and a minimum 10 years of available data.



Fig. 9. Water transfers in South Africa as recorded in the DWS portal. Intra-basin connections are symbolized in gray, and inter-basin transfers are symbolized with random colors.

addition, there was interest in mapping water transfers across the country (Section 5.1.2). In order to do this, an R script was written outside of HYPEtools to scrape the station metadata from the portal, and various HYPEtools features were used to help with mapping and analysis.

5.1.1. Identify observation stations for a location and period of interest

In order to assess the spatial distribution of the available DWS observational data, metadata for each of the DWS observation stations was scraped from the portal in December 2020. This metadata was then summarized and formatted into an R data frame that could be used by the HYPEtools *PlotMapPoints()* function to generate a Leaflet map of all available observation stations symbolized by station type (Fig. 7). This interactive map provided an intuitive way to gain an understanding of the distribution of observation stations and station types throughout South Africa and showed that there was an abundance of river and meteorological data in populous areas. One notable feature of Leaflet maps created using *PlotMapPoints()* is the ability to toggle on/off different layer groups on the map. In the case of the DWS station map, individual layer groups were included for each station type so users could choose which station types to show/hide on the map. This facilitated visual inspection of specific locations of interest and made it easy to determine if data for specific station types were available in a given area. Overall, the Leaflet map was best suited for quick visualization and review of the station metadata, and more comprehensive tools were needed for further in-depth data exploration.

For a particular research project, there was interest in identifying a group of river discharge observations stations in the uMngeni drainage basin with data available between 1961 and 2006. The project had no

strict requirements on the length of the period of available data for the group of selected stations. However, it was desirable to identify a group which maximized the number of stations and the length of available data periods while simultaneously ensuring good spatial coverage throughout the drainage region. In order to identify observation stations meeting these criteria, the same station metadata dataset used to generate the *PlotMapPoints()* map was used as input to HYPEtools' *VisualizeMapPoints()* Shiny app, which provides more advanced features for interactive data exploration (See Section 4.6). Using the interactive tables in the Shiny app, the observation stations could be filtered to include only those with "River" site type that were located in the uMngeni drainage region and had a data availability start date before 2006 and a data availability end date after 1961. The stations were then filtered in the app according to the length of the available data periods, and it was decided that using a minimum period length of 10 years provided an acceptable group of 85 observation stations that were well distributed throughout the drainage region. As shown by the app's interactive boxplot, the period of available data for the 85 stations ranged from 11.43 years to 93.19 years with a median of 43.82 years (Fig. 8). The list of station IDs could then be downloaded to a CSV file and used for further analyses and retrieval of the time series datasets.

Overall, the combination of the HYPEtools *PlotMapPoints()* function and the *VisualizeMapPoints()* app made performing exploratory data analysis and visualization much simpler, and they greatly expedited the process of identifying observation stations meeting the desired criteria. In addition, the use of these HYPEtools features significantly reduced the amount of custom coding required for generating the maps and performing the analyses. Furthermore, the app made it easy to test alternative selection criteria for the observation stations and assess the



Fig. 10. Classification of nutrient observation stations within the Baltic Sea drainage basin.

tradeoff between obtaining data for a greater number of stations vs. obtaining longer data records, all without the need to write additional code. Finally, these interactive maps and analysis tools were valuable assets for engaging with stakeholders and presenting the results in an approachable format.

5.1.2. Mapping water transfers

In water-scarce regions, inter-basin water transfers can be used to reconcile problems with water availability. However, these transfers can be vulnerable to severe large-scale droughts which simultaneously affect multiple neighboring drainage regions. As part of a research project to assess the vulnerability of inter-basin transfers to droughts in certain regions of South Africa, there was a need to gain a quick overview of the location of water transfers across the country. Using the information on the downstream connections for the canal, river, and pipeline stations from the Department of Water and Sanitation portal, an R data frame was constructed with columns for the station IDs of the upstream and downstream connections. This data frame was then used as input to the *MapRegionalSources()* function to produce an interactive Leaflet map of the water transfers (Fig. 9) with intra-basin connections symbolized in gray and inter-basin transfers symbolized with random colors. From this map, it is apparent that the majority of the transfers are short local

connections from e.g. dams to downstream locations in the vicinity. However, several notable inter-basin transfers are also visible such as the Orange-Fish River Tunnel (symbolized in magenta) which diverts water from the Gariep Dam to the Fish River valley and is the longest continuous enclosed aqueduct in the southern hemisphere. Ultimately, the *MapRegionalSources* function made mapping the water transfers very straightforward, and the ability to zoom and pan throughout the map made it easy to identify connections in the areas of interest.

5.2. Baltic Sea drainage basin

In Europe, anthropogenically driven eutrophication in the Baltic Sea is a well-documented problem, and large-scale hydrological models have been used to investigate the impacts of climate change and the implementation of mitigation measures (Bartosova et al., 2019; Capell et al., 2021). However, the calibration of large-scale hydrological models presents many challenges including how to select representative calibration stations and how to determine which processes may be affecting biases in the model results. Fortunately, HYPEtools provides tools to assist with the various steps of a typical workflow for model calibration and evaluation. The fourth-generation version of the pan-European HYPE model setup, E-HYPE4, was recently calibrated for

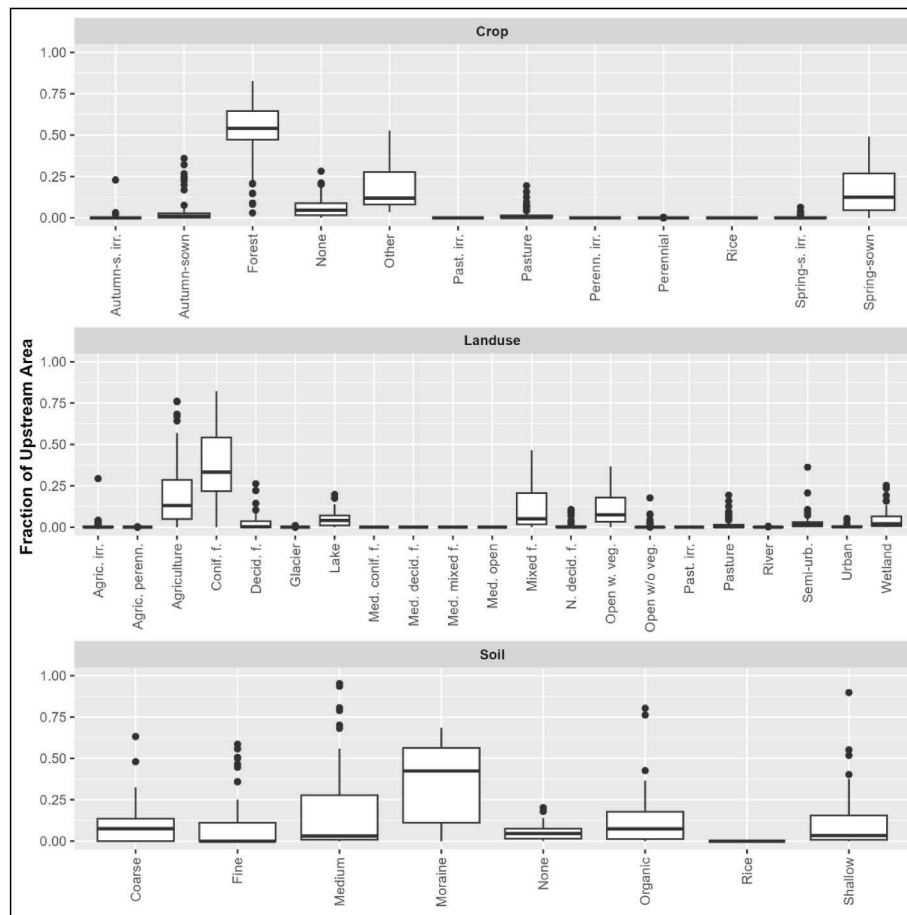


Fig. 11. Upstream crop, land use, and soil fractions for coastal observation stations ($n = 69$) in the Baltic Sea drainage basin.

discharge and sediments (Brendel et al., 2023), and the model is now being calibrated for simulation of nitrogen (inorganic + organic) and phosphorus (soluble + particulate) to study nutrient export to the Baltic Sea and to provide inputs to oceanographic models. Here, we use the ongoing E-HYPE4 nutrient calibration to provide examples demonstrating how HYPETOOLS can be used to simplify common model calibration tasks such as selecting and classifying calibration stations (Section 5.2.1), assessing model biases by catchment attributes (Section 5.2.2), and evaluating model dynamics (Section 5.2.3).

5.2.1. Select and classify observation stations

If the primary use of a certain hydrological model is to simulate fluxes to coastal areas, then it can be advantageous to focus calibration efforts on optimizing the model performance for stations near the coasts. The HYPETOOLS `OutletNearObs()` function makes it easy to identify coastal stations by classifying stations according to a drainage fraction threshold; stations are classified as coastal if the ratio between the station's upstream drainage area and the total drainage area upstream of the respective river outlet (i.e. the discharge point into the sea) is greater

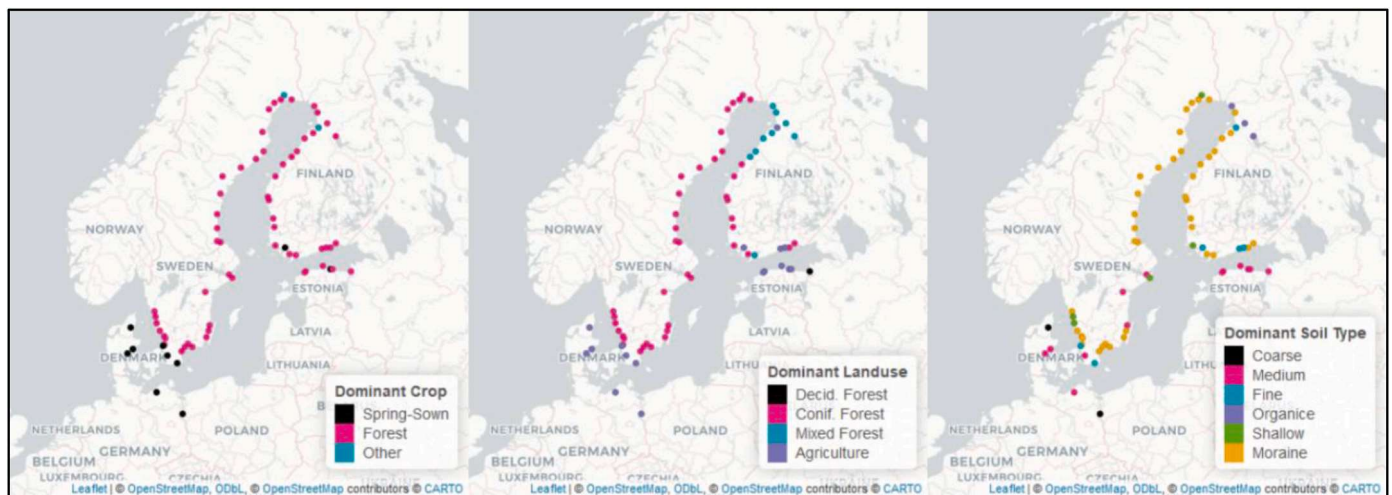


Fig. 12. Dominant crop, land use, and soil types for coastal observation stations ($n = 69$) in the Baltic Sea drainage basin.

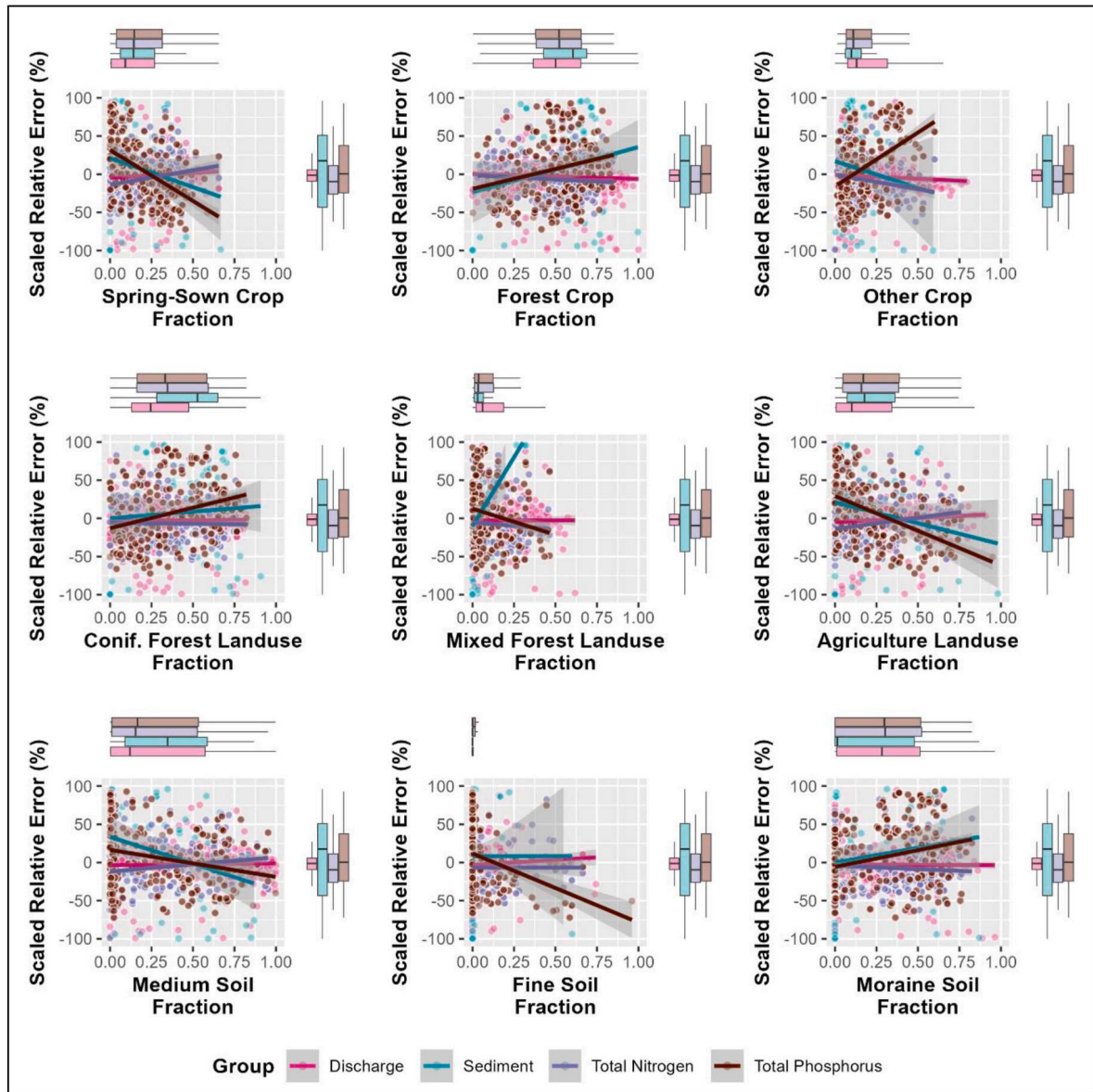


Fig. 13. Scaled relative error for simulated discharge and concentrations of sediments, total nitrogen, and total phosphorus versus upstream fractions of dominant crop, land use, and soil types for observation stations within the Baltic Sea drainage basin.

than or equal to the threshold fraction. In the E-HYPE4 model, nutrient observations are available for 234 stations within the Baltic Sea drainage basin, and using a threshold drainage fraction of 0.8, there are 69 coastal observation stations that could be prioritized during model calibration. Fig. 10 presents the categorization of all 234 stations in the Baltic Sea drainage basin on a map generated using the *PlotMapPoints()* function. In the case of multiple observation stations located on the same river meeting the minimum drainage fraction criteria, then only the furthest downstream station was considered coastal. Overall, the *OutletNearObs()* and *PlotMapPoints()* functions make it simple to identify and map coastal stations by automating (1) the assessment of the subbasin routing to determine the corresponding outlets for each observation station, (2) the calculation of the drainage area ratios and classification of the stations, and (3) the map generation.

In order to tune various processes in large-scale hydrological models, it can be helpful to first identify groups of calibration stations with drainage areas that are representative of certain features such as crop, land use, and soil types. This allows for (1) the isolation and calibration of key processes and characteristics individually for each group, and (2)

the evaluation of the model performance for each group separately. The HYPEtools *UpstreamGroupSLCClasses()* function aggregates the individual HYPE HRU fractions for a station's drainage area to calculate the total fractions corresponding to each modeled crop, land use, and soil type. In E-HYPE4, there are 97 unique HRU classes derived from 13 crop types, 21 land use categories, and 8 soil types (Brendel et al., 2023). Fig. 11 shows the upstream crop, land use, and soil type fractions for the 69 coastal observation stations as calculated by *UpstreamGroupSLCClasses()*. Overall, there are many crop, land use, and soil classes which have negligible contributions to drainage areas of the coastal observation stations (Fig. 11). Consequently, while calibrating the Baltic Sea region of E-HYPE4 for nutrients, it may be beneficial to use the *CleanSLCClasses()* function to reassign these small fractions to the dominant classes to reduce the number of classes and parameters which must be calibrated. Fig. 12 shows the most dominant crop, land use, and soil type (i.e. the respective classes with the largest upstream fraction) for each observation station as mapped using *PlotMapPoints()*. From this map, it is apparent that there were two crop types (spring-sown and forest) and three land uses (coniferous forest, mixed forest, and

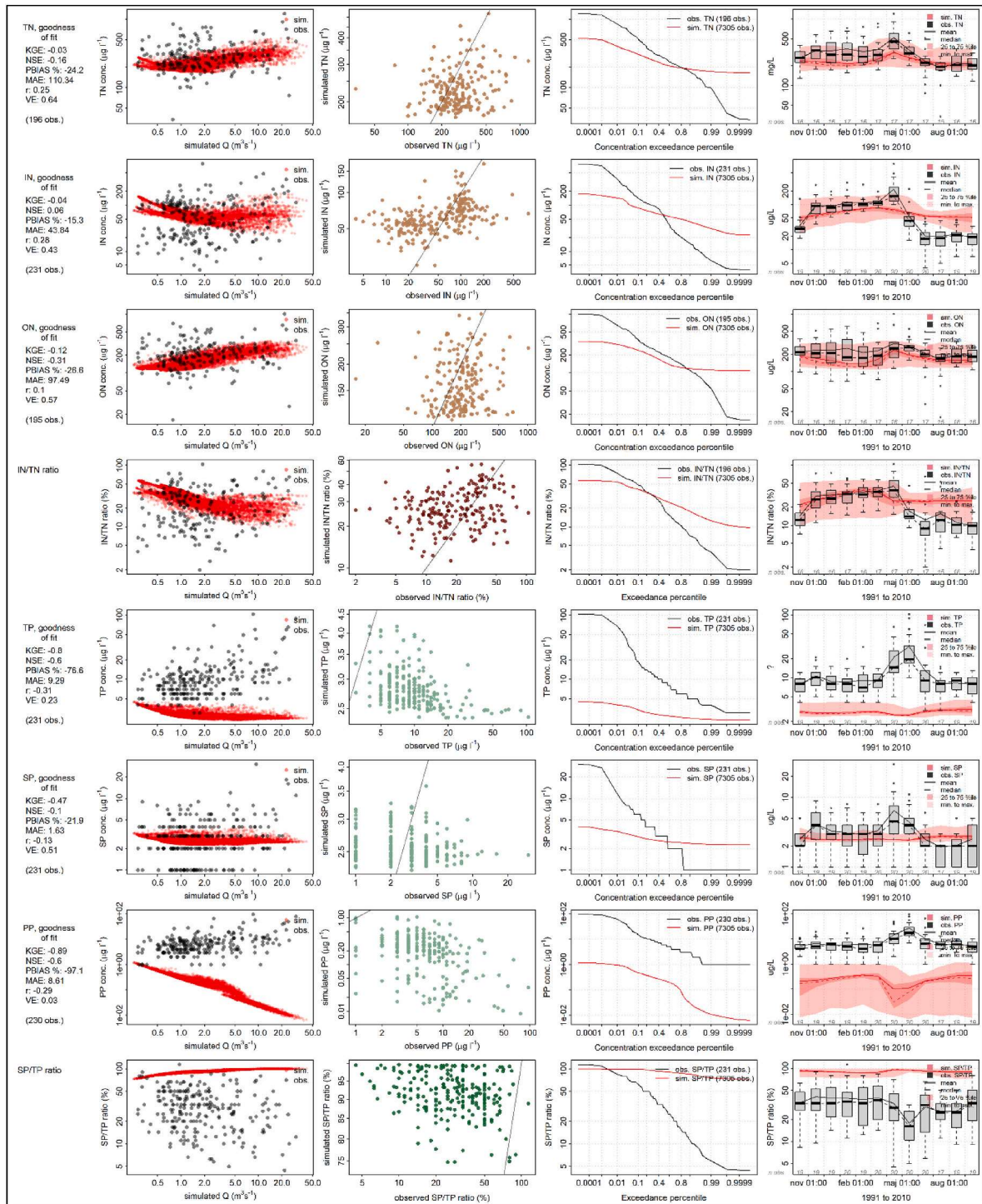


Fig. 14. PlotBasinSummary output for subbasin #38564 before calibration.

agriculture) which were most dominant. In contrast, there was more variety in the dominant soil types; although moraine soils were the most dominant soil type, there were also many stations which had medium, fine, organic, or shallow soils as their most dominant soil type. For the E-HYPE4 nutrient calibration, this means that the soil parameters will be especially important for capturing the spatial variations in the nutrient fluxes. Altogether, HYPEtools expedites the identification of the dominant crop, land use, and soil types for a domain of interest which can, in turn, guide the selection of calibration parameters and the development of a model calibration strategy.

5.2.2. Assess model biases by catchment attributes

During calibration of hydrological models with many calibration parameters, it can often be difficult to determine which processes may have the greatest influence on the biases in the model results. The *PlotPerformanceByAttribute()* function simplifies the visual identification of relationships between model biases and catchment characteristics with a series of grouped scatter plots. Knowledge of these relationships can then be used to determine which attributes, processes, and parameters may have the greatest impact on the simulation results. In E-HYPE4, observations for the Baltic Sea drainage basin are available for 545 discharge stations, 57 stations measuring sediment concentrations,

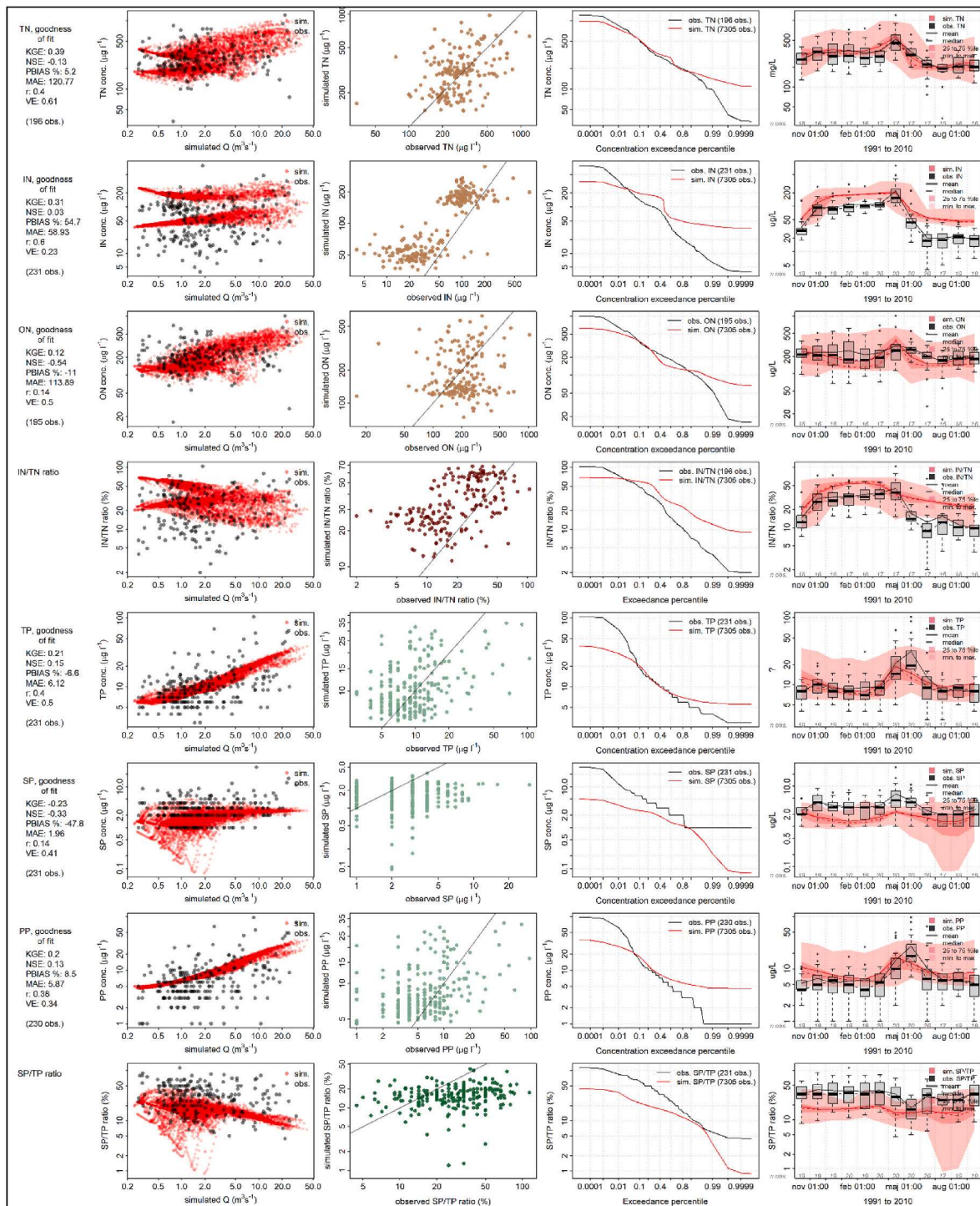


Fig. 15. PlotBasinSummary output for subbasin #38564 after calibration.

219 stations measuring total nitrogen concentrations, and 229 stations measuring total phosphorus concentrations. For each station, a scaled relative error ranging from -100% to 100% can be calculated between the simulated and observed time series. Then, using *PlotPerformanceByAttribute()*, the scaled relative errors for each station from an initial E-HYPE4 nutrient calibration can be plotted against the stations' respective upstream fractions of the three most dominant crop, land use, and soil types (Fig. 13).

Analysis of the trendlines in Fig. 13 provides an indication of the correlation between the model biases and the crop/land use/soil fractions. Generally, the biases in discharge and nitrogen concentrations did

not display a strong correlation with the upstream fractions of any of the dominant crops, land uses, or soil types. However, the trendlines for discharge and nitrogen concentrations were very similar, which suggests that biases in the nitrogen concentrations could be tied to the biases in discharge. The trendlines for sediments and total phosphorus were often similar, which highlights the impact of the connection between sediments and the particulate forms of phosphorus. However, there were large differences in the trendlines between the two substances in the plots for other crops, mixed forest land use, and fine soils, and this may indicate that there are errors in the partitioning of total phosphorus between the soluble and particulate forms. Because the relative error for

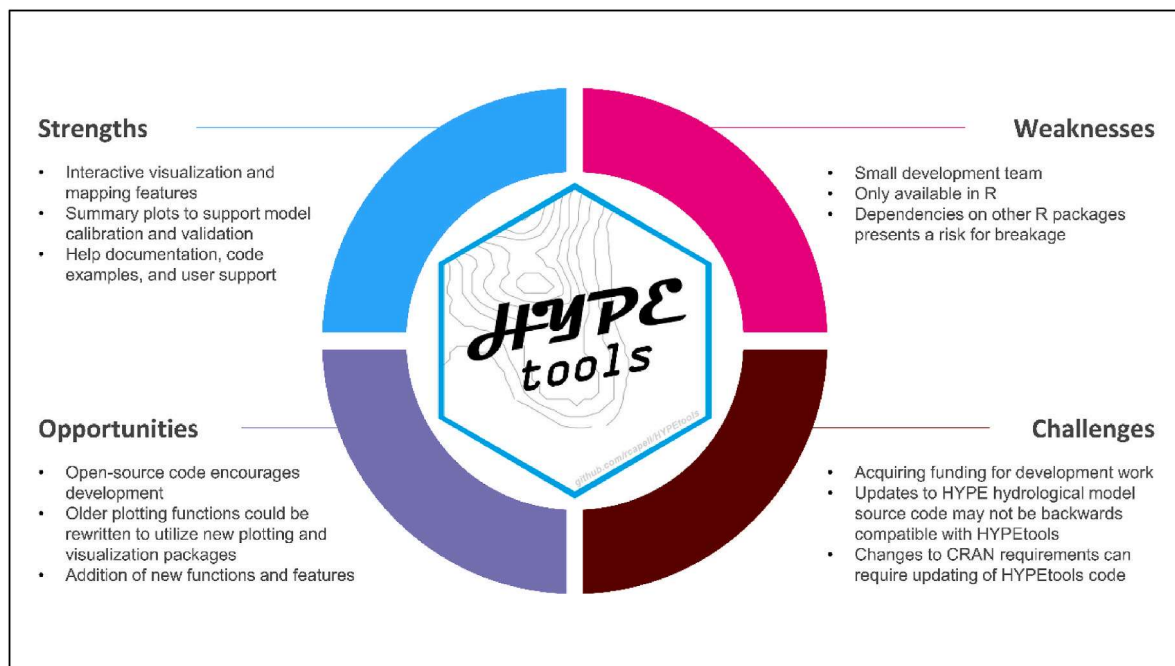


Fig. 16. SWOC analysis of the HYPEtools package.

total phosphorus decreases with increasing proportions of agricultural land and spring sown crops, this may indicate that the parameters controlling total phosphorus export should be adjusted to provide more export. Overall, the `PlotPerformanceByAttribute()` function allows for the quick assessment of the correlations and relationships between model biases and catchment characteristics, and the insights gained from these plots can be used to guide and develop a model calibration strategy.

5.2.3. Evaluate model dynamics

Visual inspection of the agreement between observations and model simulations is an important step of any model calibration and evaluation workflow. The HYPEtools `PlotBasinSummary()` function provides a variety of comparison and diagnostic plots which can be used to spot-check model dynamics for particular sites of interest. Figs. 14 and 15 respectively present the basin summary plots for one selected model subbasin in Sweden with observed data before and after an initial E-HYPE4 nutrient calibration. The plots consist of a matrix with rows for total nitrogen (TN), inorganic nitrogen (IN), organic nitrogen (ON), IN/TN ratio, total phosphorus (TP), soluble phosphorus (SP), particulate phosphorus (PP), and SP/TP ratio and columns presenting (1) model goodness of fit statistics, (2) the simulated and observed concentrations versus the simulated discharge, (3) the simulated vs observed concentrations/ratios, (4) the simulated vs observed exceedance curves, and (5) the simulated vs. observed annual regimes.

The performance for the model subbasin in the uncalibrated model was, as expected, rather poor. Before calibration, there were large differences between the simulated and observed exceedance curves (Fig. 14). In general, the model underestimated the high concentrations and overestimated the low concentrations. For total and particulate phosphorus, however, the model underestimated the observed concentrations in the subbasin by several orders of magnitude across the full range of concentrations. In terms of the annual regimes, the uncalibrated model did not capture the seasonal variations in the nitrogen and phosphorus concentrations.

During the initial model calibration, the parameters related to the dominant crop, land use, and soil types were tuned with the goal of minimizing the biases between the simulation and observed nutrient concentrations. Overall, the calibration resulted in significant improvements in the KGE values for total nitrogen and total phosphorus as

well as large decreases in the magnitude of the percent biases (PBIAS) for the model subbasin (Fig. 15). Most notably, the calibration fixed the underestimation of the total phosphorus and particulate phosphorus concentrations. In general, the calibrated model also improved the simulation of high concentrations and the matching of the seasonal variations in annual regimes for each substance. However, upon closer inspection of the plots for inorganic nitrogen and soluble phosphorus, it is apparent that there are still large biases in the model simulations for these forms in the subbasin. As shown in the IN/TN and SP/TP ratio plots, these biases may be due to errors in the partitioning of nitrogen and phosphorus between the different forms. In the case of inorganic nitrogen, the model overestimates the observed concentrations at low flows, which could also indicate that the model is overestimating the loading from upstream point sources. All things considered, the `PlotBasinSummary()` function is a convenient and impactful tool for visual assessment of model performance and dynamics during model calibration, and it provides insights to the model processes which could be influencing model biases.

6. Summary

Overall, the HYPEtools R package provides a collection of functions to simplify data summarization, analysis, and exploration for not only HYPE modeling applications, but also for other hydrological models and/or non-modeling contexts. Features included in HYPEtools range from diverse aggregation and evaluation routines to interactive mapping and apps, and the package functions can be used as standalone tools or as building blocks for e.g. scripts, automated workflows, and interactive apps.

A Strengths, Weaknesses, Opportunities, and Challenges (SWOC) analysis was performed for the HYPEtools package (Fig. 16). As currently implemented, HYPEtools has demonstrated its value for calibration and analysis of hydrological models as well as for providing insights through data visualization. In addition, the HYPEtools package source code is provided as open-source so that others can adapt the code to fit their needs and requirements. However, there are several notable limitations to the HYPEtools package:

- HYPEtools is only implemented in R, so the package may be of limited use to those developing workflows in other programming languages
- The dependence of HYPEtools on other R packages presents a risk for breakage and could make using the package tricky within session specific virtual container situations.
- At the time of writing, new HYPEtools releases can only be used in a Conda environment on Linux systems because the latest R version supported for Windows Conda environments is R version 4.1.3 and conda-forge only builds the HYPEtools package for R versions $\geq 4.2.0$.

SMHI plans to continue the active development and maintenance of HYPEtools and offer guidance to those who would like to get started with HYPE or HYPEtools. Potential updates to the package include interactive plotting using e.g. the *plotly* package (Sievert, 2020), tools to provide standardized comparisons between multiple time series and/or model outputs, functions for converting between NetCDF files and HYPE model files, expanded support for converting HYPE models between different simulation time steps, and the development of Shiny apps for time series analyses.

CRedit authorship contribution statement

Conrad Brendel: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Conceptualization. **René Capell:** Writing – review & editing, Visualization, Supervision, Software, Methodology, Formal analysis, Conceptualization. **Alena Bartosova:** Writing – review & editing, Supervision.

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

Code is available on GitHub and example code is provided as supplementary material

Acknowledgments

The development of HYPEtools was performed at the SMHI Hydrological Research unit, where much of our work is done jointly, and we would like to acknowledge all of our co-workers — especially Jafet Andersson, David Gustafsson, and Jude Musuza — for their contributions. This work was partially funded by several projects and efforts including Bilateral Cooperation in South Africa funded by the Swedish Environmental Protection Agency and the EU projects DIRT-X (funded by JPI-AXIS, Grant No. 776608) and FOCCUS (funded by Horizon, Grant No. 101133911). In addition, we would like to thank everyone who has contributed to the HYPE/HYPEtools open source community. Finally, special thanks to Meshuggah for the source of inspiration \m/

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.envsoft.2024.106094>.

References

Albers, S., Prosdociimi, I., 2023. CRAN task view: hydrological data and modeling. Version 2023-05-28 [WWW Document]. URL: <https://CRAN.R-project.org/view=Hydrology>. (Accessed 12 November 2023).

- Andersson, J.C.M., Ali, A., Arheimer, B., Gustafsson, D., Minoungou, B., 2017. Providing peak river flow statistics and forecasting in the Niger River basin. *Phys. Chem. Earth Parts ABC, Infrastructure Planning for Water Security in Eastern and Southern Africa* 100, 3–12. <https://doi.org/10.1016/j.pce.2017.02.010>.
- Arheimer, B., Pimentel, R., Isberg, K., Crochemore, L., Andersson, J.C.M., Hasan, A., Pineda, L., 2020. Global catchment modelling using World-Wide HYPE (WWH), open data, and stepwise parameter estimation. *Hydrol. Earth Syst. Sci.* 24, 535–559. <https://doi.org/10.5194/hess-24-535-2020>.
- Astagneau, P.C., Thirel, G., Delaigue, O., Guillaume, J.H.A., Parajka, J., Brauer, C.C., Viglione, A., Buytaert, W., Beven, K.J., 2021. Technical note: hydrology modelling R packages – a unified analysis of models and practicalities from a user perspective. *Hydrol. Earth Syst. Sci.* 25, 3937–3973. <https://doi.org/10.5194/hess-25-3937-2021>.
- Aydin, O., Pavlushko, D., Walbridge, S., Kopp, S., Janikas, M.V., 2024. arcgisbinding: an R package for integrating R and ArcGIS. *Environ. Model. Software* 172, 105904. <https://doi.org/10.1016/j.envsoft.2023.105904>.
- Bartosova, A., Arheimer, B., de Lavenne, A., Capell, R., Strömqvist, J., 2021. Large-scale hydrological and sediment modeling in nested domains under current and changing climate. *J. Hydrol. Eng.* 26, 05021009 [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0002078](https://doi.org/10.1061/(ASCE)HE.1943-5584.0002078).
- Bartosova, A., Capell, R., Olesen, J.E., Jabloun, M., Refsgaard, J.C., Donnelly, C., Hyttiäinen, K., Pihlainen, S., Zandersen, M., Arheimer, B., 2019. Future socioeconomic conditions may have a larger impact than climate change on nutrient loads to the Baltic Sea. *Ambio* 48, 1325–1336. <https://doi.org/10.1007/s13280-019-01243-5>.
- Bivand, R., 2023. Rgrass: Interface between “GRASS” Geographical Information System and “R”.
- Brendel, C., Capell, R., Bartosova, A., 2023. To tame a land: limiting factors in model performance for the multi-objective calibration of a pan-European, semi-distributed hydrological model for discharge and sediments. *J. Hydrol. Reg. Stud.* 50, 101544 <https://doi.org/10.1016/j.ejrh.2023.101544>.
- Brendel, C.E., Dymond, R.L., Aguilar, M.F., 2020. Integration of quantitative precipitation forecasts with real-time hydrology and hydraulics modeling towards probabilistic forecasting of urban flooding. *Environ. Model. Software* 134, 104864. <https://doi.org/10.1016/j.envsoft.2020.104864>.
- Brendel, C.E., Dymond, R.L., Aguilar, M.F., 2019. An interactive web app for retrieval, visualization, and analysis of hydrologic and meteorological time series data. *Environ. Model. Software* 117, 14–28. <https://doi.org/10.1016/j.envsoft.2019.03.003>.
- Bui, T.T.P., Kantoush, S., Kawamura, A., Du, T.L.T., Bui, N.T., Capell, R., Nguyen, N.T., Du Bui, D., Saber, M., Tetsuya, S., Lee, H., Saleh, A., Lakshmi, V., Bartosova, A., Van Binh, D., Nguyen, B.Q., Nguyen, T.T.T., 2023. Reservoir operation impacts on streamflow and sediment dynamics in the transboundary river basin, Vietnam. *Hydrol. Process.* 37, e14994 <https://doi.org/10.1002/hyp.14994>.
- Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., Guillaumot, L., Zhao, F., Wada, Y., 2020. Development of the Community Water Model (CWatM v1.04) – a high-resolution hydrological model for global and regional assessment of integrated water resources management. *Geosci. Model Dev. (GMD)* 13, 3267–3298. <https://doi.org/10.5194/gmd-13-3267-2020>.
- Capell, R., Bartosova, A., Tonderski, K., Arheimer, B., Pedersen, S.M., Zilans, A., 2021. From local measures to regional impacts: modelling changes in nutrient loads to the Baltic Sea. *J. Hydrol. Reg. Stud.* 36, 100867 <https://doi.org/10.1016/j.ejrh.2021.100867>.
- Capell, R., Brendel, C., 2023. HYPEtools: tools for processing and analyzing files from the hydrological catchment model HYPE. <https://zenodo.org/doi/10.5281/zenodo.7627955>.
- Carlsaw, D.C., Ropkins, K., 2012. Openair — an R package for air quality data analysis. *Environ. Model. Software* 27–28, 52–61. <https://doi.org/10.1016/j.envsoft.2011.09.008>.
- Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., Borges, B., 2023. Shiny: web application framework for R. R package version 1.7.5.1. <https://CRAN.R-project.org/package=shiny>.
- Chen, Y., Li, J., Wang, H., Qin, J., Dong, L., 2017. Large-watershed flood forecasting with high-resolution distributed hydrological model. *Hydrol. Earth Syst. Sci.* 21, 735–749. <https://doi.org/10.5194/hess-21-735-2017>.
- Cheng, J., Schloerke, B., Karambelkar, B., Xie, Y., 2023. Leaflet: create interactive web maps with the JavaScript 'leaflet'. R package version 2.2.0. <https://CRAN.R-project.org/package=leaflet>.
- Chlumsky, R., Craig, J.R., Lin, S.G.M., Grass, S., Scantlebury, L., Brown, G., Arabzadeh, R., 2022. RavenR v2.1.4: an open-source R package to support flexible hydrologic modelling. *Geosci. Model Dev. (GMD)* 15, 7017–7030. <https://doi.org/10.5194/gmd-15-7017-2022>.
- Cinkus, G., Mazzilli, N., Jourde, H., 2023. KarstID: an R Shiny application for the analysis of karst spring discharge time series and the classification of karst system hydrological functioning. *Environ. Earth Sci.* 82, 136. <https://doi.org/10.1007/s12665-023-10830-5>.
- Cordano, E., 2024. Geotopbricks: an R Plug-In for the Distributed Hydrological Model GEOTop.
- Coron, L., Thirel, G., Delaigue, O., Perrin, C., Andréassian, V., 2017. The suite of lumped GR hydrological models in an R package. *Environ. Model. Software* 94, 166–171. <https://doi.org/10.1016/j.envsoft.2017.05.002>.
- Craig, J.R., Brown, G., Chlumsky, R., Jenkinson, R.W., Jost, G., Lee, K., Mai, J., Serrero, M., Sgro, N., Shafii, M., Snowdon, A.P., Tolson, B.A., 2020. Flexible watershed simulation with the Raven hydrological modelling framework. *Environ. Model. Software* 129, 104728. <https://doi.org/10.1016/j.envsoft.2020.104728>.

- Cruz-Alonso, V., Pucher, C., Ratcliffe, S., Ruiz-Benito, P., Astigarraga, J., Neumann, M., Hasenauer, H., Rodríguez-Sánchez, F., 2023. The easyclimate R package: easy access to high-resolution daily climate data for Europe. *Environ. Model. Software* 161, 105627. <https://doi.org/10.1016/j.envsoft.2023.105627>.
- de Oliveira Ferreira Silva, C., Heriberto de Castro Teixeira, A., Lilla Manzione, R., 2019. agriwater: an R package for spatial modelling of energy balance and actual evapotranspiration using satellite images and agrometeorological data. *Environ. Model. Software* 120, 104497. <https://doi.org/10.1016/j.envsoft.2019.104497>.
- DeCicco, L., Hirsch, R., Lorenz, D., Watkins, D., Johnson, M., 2023. dataRetrieval: R packages for discovering and retrieving water data available from U.S. federal hydrologic web services. *U.S. Geol. Surv.* <https://doi.org/10.5066/P9X4L3GE>.
- Dile, Y.T., Daggupati, P., George, C., Srinivasan, R., Arnold, J., 2016. Introducing a new open source GIS user interface for the SWAT model. *Environ. Model. Software* 85, 129–138. <https://doi.org/10.1016/j.envsoft.2016.08.004>.
- Dowle, M., Srinivasan, A., 2023. data.table: extension of 'data.frame'. R package version 1.14.8. <https://CRAN.R-project.org/package=data.table>.
- Drost, S., Malewski, C., Jirka, S., 2022. Sensor web and internet of things technologies for hydrological measurement data. In: Di Mauro, A., Scozzari, A., Soldovieri, F. (Eds.), *Instrumentation and Measurement Technologies for Water Cycle Management*. Springer Water. Springer International Publishing, Cham, pp. 279–293. https://doi.org/10.1007/978-3-031-08262-7_12.
- Gelfan, A., Gustafsson, D., Motovilov, Y., Arheimer, B., Kalugin, A., Krylenko, I., Lavrenov, A., 2017. Climate change impact on the water regime of two great Arctic rivers: modeling and uncertainty issues. *Clim. Change* 141, 499–515. <https://doi.org/10.1007/s10584-016-1710-5>.
- Girons Lopez, M., Crochemore, L., Pechlivanidis, I.G., 2021. Benchmarking an operational hydrological model for providing seasonal forecasts in Sweden. *Hydrol. Earth Syst. Sci.* 25, 1189–1209. <https://doi.org/10.5194/hess-25-1189-2021>.
- Hankin, B., Strömquist, J., Burgess, C., Pers, C., Bielby, S., Revilla-Romero, B., Pope, L., 2019. A new national water quality model to evaluate the effectiveness of catchment management measures in England. *Water* 11, 1612. <https://doi.org/10.3390/w11081612>.
- Hijmans, R.J., 2023. Terra: Spatial Data Analysis.
- Hughes, J.D., Liu, J., 2008. MIKE SHE: software for integrated surface water/groundwater modeling. *Groundwater* 46, 797–802. <https://doi.org/10.1111/j.1745-6584.2008.00500.x>.
- Hundecha, Y., Arheimer, B., Donnelly, C., Pechlivanidis, I., 2016. A regional parameter estimation scheme for a pan-European multi-basin model. *J. Hydrol. Reg. Stud.* 6, 90–111. <https://doi.org/10.1016/j.ejrh.2016.04.002>.
- Ihaka, R., Gentleman, R., 1996. R: a language for data analysis and graphics. *J. Comput. Graph. Stat.* 5, 299–314. <https://doi.org/10.1080/10618600.1996.10474713>.
- Jiang, Z., Rashid, MdM., Johnson, F., Sharma, A., 2021. A wavelet-based tool to modulate variance in predictors: an application to predicting drought anomalies. *Environ. Model. Software* 135, 104907. <https://doi.org/10.1016/j.envsoft.2020.104907>.
- Krysanova, V., Hattermann, F.F., 2017. Intercomparison of climate change impacts in 12 large river basins: overview of methods and summary of results. *Clim. Change* 141, 363–379. <https://doi.org/10.1007/s10584-017-1919-y>.
- Leutnant, D., 2020. Dleutnant/swmmr - BugReport: total inflow not correctly read from out File [WWW Document]. URL. <https://github.com/dleutnant/swmmr/issues/48>, 1.10.24.
- Leutnant, D., Döring, A., Uhl, M., 2019. Swmmr - an R package to interface SWMM. *Urban Water J.* 16, 68–76. <https://doi.org/10.1080/1573062X.2019.1611889>.
- Lindström, G., Pers, C., Rosberg, J., Strömquist, J., Arheimer, B., 2010. Development and testing of the HYPE (Hydrological Predictions for the Environment) water quality model for different spatial scales. *Hydrol. Res.* 41, 295–319. <https://doi.org/10.2166/nh.2010.007>.
- Lowry, C.S., Fienen, M.N., Hall, D.M., Stepenuck, K.F., 2019. Growing pains of crowdsourced stream stage monitoring using mobile phones: the development of CrowdHydrology. *Front. Earth Sci.* 7.
- McCabe, M.F., Rodell, M., Alsdorf, D.E., Miralles, D.G., Uijlenhoet, R., Wagner, W., Lucieer, A., Houborg, R., Verhoest, N.E.C., Franz, T.E., Shi, J., Gao, H., Wood, E.F., 2017. The future of Earth observation in hydrology. *Hydrol. Earth Syst. Sci.* 21, 3879–3914. <https://doi.org/10.5194/hess-21-3879-2017>.
- Moradkhani, H., Sorooshian, S., 2008. General review of rainfall-runoff modeling: model calibration, data assimilation, and uncertainty analysis. In: Sorooshian, S., Hsu, K.-L., Coppola, E., Tomassetti, B., Verdecchia, M., Visconti, G. (Eds.), *Hydrological Modelling and the Water Cycle: Coupling the Atmospheric and Hydrological Models*, Water Science and Technology Library. Springer, Berlin, Heidelberg, pp. 1–24. https://doi.org/10.1007/978-3-540-77843-1_1.
- Muenchow, J., Schratz, P., Brenning, A., 2017. RQGIS: integrating R with QGIS for statistical geocomputing. *R J* 9, 409. <https://doi.org/10.32614/RJ-2017-067>.
- Njue, N., Stenfort Kroese, J., Gräf, J., Jacobs, S.R., Weeser, B., Breuer, L., Rufino, M.C., 2019. Citizen science in hydrological monitoring and ecosystem services management: state of the art and future prospects. *Sci. Total Environ.* 693, 133531. <https://doi.org/10.1016/j.scitotenv.2019.07.337>.
- Ockenden, M.C., Holloway, M.J., Beven, K.J., Collins, A.L., Evans, R., Falloon, P.D., Forber, K.J., Hiscock, K.M., Kahana, R., Macleod, C.J.A., Tych, W., Villamizar, M.L., Wearing, C., Withers, P.J.A., Zhou, J.G., Barker, P.A., Burke, S., Freer, J.E., Johnes, P.J., Snell, M.A., Surridge, B.W.J., Haygarth, P.M., 2017. Major agricultural changes required to mitigate phosphorus losses under climate change. *Nat. Commun.* 8, 161. <https://doi.org/10.1038/s41467-017-00232-0>.
- Ogliari, E., Nespoli, A., Mussetta, M., Pretto, S., Zimbardo, A., Bonfanti, N., Aufiero, M., 2020. A hybrid method for the run-of-the-river hydroelectric power plant energy forecast: HYPE hydrological model and neural network. *Forecasting* 2, 410–428. <https://doi.org/10.3390/forecast2040022>.
- Pebesma, E., Bivand, R., 2023. Spatial Data Science: with Applications in R. Chapman and Hall/CRC, Boca Raton. <https://doi.org/10.1201/9780429459016>.
- Pechlivanidis, I.G., Arheimer, B., 2015. Large-scale hydrological modelling by using modified PUB recommendations: the India-HYPE case. *Hydrol. Earth Syst. Sci.* 19, 4559–4579. <https://doi.org/10.5194/hess-19-4559-2015>.
- Rottler, E., Vormoor, K., Francke, T., Bronstert, A., 2021. Hydro Explorer: an interactive web app to investigate changes in runoff timing and runoff seasonality all over the world. *River Res. Appl.* 37, 544–554. <https://doi.org/10.1002/rra.3772>.
- Samsonov, T., 2023. Grwat: River Hydrograph Separation and Analysis.
- Sharma, A., Mehrotra, R., Li, J., Jha, S., 2016. A programming tool for nonparametric system prediction using Partial Informational Correlation and Partial Weights. *Environ. Model. Software* 83, 271–275. <https://doi.org/10.1016/j.envsoft.2016.05.021>.
- Sievert, C., 2020. Interactive Web-Based Data Visualization with R, Plotly, and Shiny. Chapman and Hall/CRC, Florida.
- Singh, V.P., 2018. Hydrologic modeling: progress and future directions. *Geosci. Lett.* 5, 15. <https://doi.org/10.1186/s40562-018-0113-z>.
- Slater, L.J., Thirel, G., Harrigan, S., Delaigue, O., Hourakhi, A., Prosdocimi, I., Vitolo, C., Smith, K., 2019. Using R in hydrology: a review of recent developments and future directions. *Hydrol. Earth Syst. Sci.* 23, 2939–2963. <https://doi.org/10.5194/hess-23-2939-2019>.
- SMHI, 2024. HYPE model documentation. <http://hype.smhi.net/wiki/doku.php>.
- Sood, A., Smakhtin, V., 2015. Global hydrological models: a review. *Hydrol. Sci. J.* 60, 549–565. <https://doi.org/10.1080/02626667.2014.950580>.
- Stadnyk, T.A., MacDonald, M.K., Tefs, A., Déry, S.J., Koenig, K., Gustafsson, D., Isberg, K., Arheimer, B., 2020. Hydrological modeling of freshwater discharge into Hudson Bay using HYPE. *Elem. Sci. Anthr.* 8, 43. <https://doi.org/10.1525/elementa.439>.
- Tijerina-Kreuzer, D., Condon, L., FitzGerald, K., Dugger, A., O'Neill, M.M., Sampson, K., Gochis, D., Maxwell, R., 2021. Continental hydrologic intercomparison project, phase 1: a large-scale hydrologic model comparison over the continental United States. *Water Resour. Res.* 57, e2020WR028931. <https://doi.org/10.1029/2020WR028931>.
- Toum, E., Masiokas, M.H., Villalba, R., Pitte, P., Ruiz, L., 2021. The HBV.IANIGLA hydrological model. *R J* 13, 378. <https://doi.org/10.32614/RJ-2021-059>.
- Wasko, C., Guo, D., 2022. Understanding event runoff coefficient variability across Australia using the hydroEvents R package. *Hydrol. Process.* 36, e14563. <https://doi.org/10.1002/hyp.14563>.
- Wickham, H., 2019. *Advanced R*, second ed. CRC Press.
- Wickham, H., 2016. ggplot2: elegant graphics for data analysis. <https://ggplot2.tidyverse.org>.
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H., 2019. Welcome to the tidyverse. *J. Open Source Softw.* 4, 1686. <https://doi.org/10.21105/joss.01686>.
- Wickham, H., Csárdi, G., Eugster, M., 2022. roxygen2: in-Line Documentation for R. <https://roxygen2.r-lib.org/>. <https://github.com/r-lib/roxygen2>.