Technical Note: Hydrology modelling R packages: a unified analysis of models and practicalities from a user perspective

Paul C. Astagneau[1,2], Guillaume Thirel[1], Olivier Delaigue[1], Joseph H. A. Guillaume[3], Juraj Parajka[4], Claudia C. Brauer[5], Alberto Viglione[6], Wouter Buytaert[7], and Keith J. Beven[8]

[1]Université Paris-Saclay, INRAE, HYCAR Research Unit, Antony, France
[2]Sorbonne Université, Paris, France
[3]Institute for Water Futures and Fenner School of Environment & Society, Australian National University, Canberra, Australia
[4]Institute of Hydraulic and Water Resources Engineering, TU Vienna, Vienna, Austria
[5]Hydrology and Quantitative Water Management Group, Wageningen University and Research, Wageningen, The Netherlands
[6]Department of Environment, Land and Infrastructure Engineering, Politecnico di Torino, Turin, Italy
[7]Department of Civil and Environmental Engineering, Imperial College London, London, UK
[8]Lancaster Environment Centre, Lancaster University, Lancaster, UK

Correspondence: Paul Astagneau (paul.astagneau@inrae.fr)

Abstract. Following the rise of R as a scientific programming language, the increasing requirement for more transferable research, and the growth of data availability in hydrology, R packages containing hydrological models are becoming more and more available as an open source resource to hydrologists. Corresponding to the core of the hydrological studies workflow, their value is increasingly meaningful regarding the reliability of methods and results. Despite package and model distinctiveness, no study has ever provided a comparison of R packages for conceptual rainfall-runoff modelling from a user perspective, contrasting their philosophy, model characteristics and ease of use. We have selected eight packages based on our ability to consistently run their models on simple hydrology modelling examples. We have uniformly analysed the exact structure of seven of the hydrological models integrated in these R packages in terms of conceptual storages and fluxes, spatial discretisation, data requirements and output provided. The analysis showed that very different modelling choices are associated with these packages, which emphasises various hydrological concepts. These specificities are not always sufficiently well explained by the package documentation. Therefore a synthesis of the package functionalities was performed from a user perspective. This synthesis helps inform selection of what packages could/should be used depending on the problem at hand. In this regard, technical features, documentation, R implementations and computational times were investigated. Moreover, by providing a framework for package comparison, this study is a step forward towards supporting more transferable and reusable methods and results for hydrological modelling in R.

1 Introduction

Since the early 1960s, many hydrologists have been designing models to better understand water cycle processes controlling river flows (e.g. Todini, 2011; Beven, 2012). These models have enabled advances with respect to a wide variety of applications in hydrology, such as flood forecasting, climate change impact assessment or water resources management. The processes

20 involved in the motion of water at the catchment scale are complex (e.g. Wagener et al., 2010), mainly due to the heterogeneity and non-linearity of the involved physical properties. Hydrological modelling can therefore be of great use regarding many scientific challenges, as it relies on a threefold simplification of time, space and hydrological processes to either match the "average behaviour" of the water cycle (Singh et al., 2017) or focus on flow extremes (e.g. floods, Georgakakos, 2006, Rozalis et al., 2010 or low flows, Staudinger et al., 2011, Nicolle et al., 2014).

25 Various types of hydrological models exist according to their assumptions on the representation of natural processes, space and time dependencies (e.g. Clark et al., 2011; Beven, 2012; Clark et al., 2017). Various programming languages enable the use of these hydrological models. For example, ~~the R language (R Core Team, 2020a), which~~ some models are implemented in Python (e.g. EXP-HYDRO hydrological model, Patil and Stieglitz, 2014) or in Matlab with the MARRMoT toolbox (Knoben et al., 2019). A significant number of models like MIKE SHE (Danish Hydraulic Institute, 2017) can only be

30 operated through commercial software and platforms.

A large number of models can be found on the R platform, such as the HBV model (Bergström, 1976) or TOPMODEL (Beven and Kirby, 1979). The R language (R Core Team, 2020a) is an open source interpreted language~~, can significantly fulfill this task. R~~. It was originally designed for statistics (as an open source implementation of the S language, Becker et al., 1988) but has since been employed in many other scientific fields. The functionalities of the R language are con-

35 tained in packages, some of which include features related to hydrology topics. The R language is reasonably easy to use and has been taking advantage of a growing community of users. ~~It~~ A large range of documentation, tutorials, manuals and online discussion platforms are developed by the R-Hydro community, such as the R Hydrology Task View on hydrological data and modeling (https://cran.r-project.org/web/views/Hydrology.html) or the page related to R on the *AboutHydrology* blog (https://abouthydrology.blogspot.com). In addition, many short courses and workshops are regularly organised (e.g. the Using

40 R in Hydrology short course at the EGU General Assembly). The R-Hydro community is also very active on many R projects and websites, such as the rOpenSci project (https://ropensci.org) or the many code examples available on Stack Overflow (https://stackoverflow.com). R can be used at each step required for a basic study in hydrology (the hydrological workflow steps, see Fig. 3 of Slater et al., 2019). Consequently, there has been an important increase in the growth and use of hydrological R packages (see Fig. 1 of Slater et al., 2019). Some of these packages are designed for hydrological modelling.

45 ~~Various types of hydrological models exist according to their assumptions on the representation of natural processes, space and time dependence (e.g. Clark et al., 2011; Beven, 2012; Clark et al., 2017). A large number can be found on the R platform, such as the HBV model (Bergström, 1976) or TOPMODEL (Beven and Kirby, 1979), while some others are implemented in Python (e.g. EXP-HYDRO hydrological model, Patil and Stieglitz, 2014) or in Matlab with the MARRMoT toolbox (Knoben et al., 2019). A significant number of models like MIKE SHE (Danish Hydraulic Institute, 2017) can only~~

50 ~~be operated through commercial software and platforms.~~ In this study, we will restrict ourselves to the hydrological models that are available within the R environment.

At a time when data management is a key issue in many branches of science, R has taken a central place in hydrology (Slater et al., 2019). Dealing with the rise of available data can be achieved within the R environment through the numerous packages for data preprocessing, such as `rnrfa` (Vitolo et al., 2016a, 2018) to retrieve hydrological data from the UK National River

Flow Archive or `raster` (Hijmans, 2020) to manipulate spatial data. While this growing availability of open source data and methods is concomitant with the increasing development of open source models, there has never been any comparison of hydrological modelling R packages. Such a comparison is required to improve the usability of hydrological models included in the R environment. Comparison is a step towards overcoming reproducibility issues related to modelling in computational hydrology (Ceola et al., 2015; Hutton et al., 2016; Melsen et al., 2017). ~~In spite of the growing availability of data and methods, there are still impediments to acceptable reproducibility of the same experiment between operators (i.e. same method and data) in the hydrology field. According to Ceola et al. (2015), this is partly due i) to the impact of modellers' decisions on the outcomes and assessments of a model (e.g. Holländer et al., 2014; Crochemore et al., 2015; Melsen et al., 2019) — though different modelling choices are to be encouraged when, for instance, working with models considered as hypotheses about how a catchment is functioning (Beven, 2018), ii) to the obstacles to good practices in the transfer of data and methods (Viglione et al., 2010) despite open source platforms availability, or, iii) as highlighted by their virtual laboratories, to the importance of comprehensive and controlled protocols.~~ Furthermore, in addition to the struggle associated with the large number of hydrological models and the difficulty to find appropriate bases for model selection (Clark et al., 2011; Beven, 2012), there are many R packages related to hydrological modelling, making it even harder to select the model best suited for a specific case. Catching the modelling "philosophy" (Hrachowitz and Clark, 2017) or differences in perceptual model (Wrede et al., 2015) behind the packages, as well as the technical features offered by a package, therefore appears as relevant to hydrologists, whether it would aim at improving the reliability of inter-comparisons or simply at correctly selecting a model. By referring to the provided documentation, any user should be able to make a choice and use a model in full knowledge of its characteristics, thus guaranteeing good practices as suggested by Jakeman et al. (2006). Unfortunately, despite the wish to standardise package documentation, especially regarding the rules imposed by the main R packages repository, the Comprehensive R Archive Network (CRAN, https://cran.r-project.org/), it remains complicated and sometimes even daunting to select a package among the R packages containing hydrological models. Yet, to our knowledge, there has never been any published study dealing with the comparison of hydrological modelling R packages. This work should i) enable any newcomer in hydrology or more experienced hydrologist to knowledgeably employ one of the packages presented in this comparison and ii) highlight possible improvements for future developments of the packages.

The review paper published in the Hydrology and Earth System Sciences journal by Slater et al. (2019) on the place of R in Hydrology has reached a large part of the hydrological science community. Our work follows on from this review and aims at reaching a large part of hydrological modellers within the R-Hydro community from beginners to highly-skilled developers. The objective of this paper is to review the pros and cons of using hydrological models implemented as packages in the R environment and to compare and evaluate their applicability and ~~efficiency in terms of hydrological structure description and usability. We select packages that implement hydrological~~ usability. The objective of the paper is neither to describe new hydrological model developments nor to evaluate them. We present the package selection rationale and the comparison methodology in Sect. 2. We provide an overview of each package with their related models in Sect. ~~2.~~ 3. We examine these models in terms of implied conceptual storages and fluxes, spatial discretisation, model requirements and outputs in Sect. ~~3.~~ 4. The hydrological model packages are evaluated according to their functionalities, provided documentation, R implementation

90 and computational efficiency in Sect. ~~4.~~ 5. We discuss the usefulness of our analysis, possible improvements for future developments and aspects of practical implementation in Sect. ~~5.~~ 6. Simple hydrology modelling examples are provided in the form of R source code.

2 ~~R packages implementing hydrological models~~ Methodology

There is a wide variety of models contained in the R packages that we have selected for this study. To lay the foundations of

95 our analysis, we first present in this section how we have selected the packages, then we ~~give a summary of each package .~~ introduce the framework for analysing the models and the packages from a user perspective. In this framework for analysis, we separate model conceptualization (Sect. 2.2) from package practicalities (Sect. 2.3). The model conceptualization is analysed in terms of model structure (Sect. 2.2.1), how to break up a catchment (Sect. 2.2.2), and number of parameters, time steps, inputs and outputs required (Sect. 2.2.3). The package practicalities are analysed in terms of functionalities (Sect. 2.3.1) and

100 usability (Sect. 2.3.2, 2.3.3 and 2.3.4).

2.1 Selection of packages

Deciding upon the number of packages implies finding the right balance between including many packages and being able to conduct a thorough assessment. On the one hand, our aim was to select as many packages as possible in order to present an extensive comparison. On the other hand, to allow a comparison, only models with similar set-ups could be used, thus we

105 had to narrow our list to do so. In this regard, we have selected the packages containing conceptual (bucket-type) continuous rainfall-runoff models as they were the most frequently encountered during our ~~researches~~ search and are widely used for many applications in hydrology (e.g. Shin and Kim, 2016). Furthermore, conceptual models usually have lower data requirements (e.g. Clark et al., 2017; Knoben et al., 2019) and a smaller computational demand. It eases their application, which is consistent with the will to facilitate the work of any user.

110 ~~Many R packages are stored on the CRAN that contains more than 15000 packages. Many other packages (around 1500) are only available on~~ We based our search on four sources: the CRAN, GitHub (https://github.com/)~~and some packages are stored on~~, the R-Forge (https://r-forge.r-project.org/) ~~. Some of the packages stored on the CRAN are also available on GitHub or on the R-Forge. GitHub~~ and an R Task View dedicated to hydrology. GitHub is a development platform based on the Git version control software. The R-Forge is a development platform specific for R packages. It is based on the Subversion version

115 control software and offers tools, such as automatic build and check of packages or mailing lists and forums. R Task Views are guides — proposed by the CRAN — on the main packages related to a certain topic. Many R packages are stored on the CRAN that contains more than 15000 packages. Many other packages (around 1500) are only available on GitHub and some packages are stored on the R-Forge. Some of the packages stored on the CRAN are also available on GitHub or on the R-Forge. Among these packages, some were designed for hydrological purposes. To identify as many packages as possible,

120 we searched for packages on the CRAN, GitHub and the R-Forge by keywords. ~~Another source was the~~ Among the R Task Views, we used the work of Zipper et al. (2019) who established a list ~~(R Task View)~~ sorting the hydrology-related packages

by topics (data retrieval, statistical modelling...). ~~R Task Views are guides — proposed by the CRAN — on the main packages related to a certain topic.~~ We looked at the packages considered as aiming at modelling (process-based modelling category) in this Hydrology Task View. ~~A~~ The review paper by Slater et al. (2019) ~~has recently been published~~ about the place of R in hydrology ~~and~~ includes a section related to hydrological modelling where some of the packages compared in our study are briefly introduced. Despite our intention to create an exhaustive list, we might have missed some packages due to the poor organisation of repositories such as GitHub and the R-Forge compared to the CRAN.

~~We chose to exclude the packages that do not contain a model but rather are complex model storage without any function for data preparation or any explanatory document (e.g. `SWATmodel`, Fuka et al., 2014). Some packages such as `fuse` (Vitolo et al., 2016b) are not included in our work. Given the large number of model structure configurations available in this package, it was considered that its main purpose was not to conduct a basic hydrological study but more to understand errors arising from hydrological models .~~

2.2 Framework for analysing the hydrological models

Investigating the different hydrological characteristics behind the models contained in the R packages is a difficult but useful exercise. It aims at gathering information about the various hydrological visions available in a comparative framework. It is relevant to proceed with this comparison task to help any student or more experienced hydrologist to understand what is involved when using a specific model implemented in one of the R packages. The selected packages contain various hydrological models based on different assumptions. These assumptions can be sorted out into simplification options regarding storages, fluxes, time and space. In this comparative study, we first propose a unified comparison of the conceptual representation of storages and fluxes by the models included in the selected packages, then the spatial discretisation they imply and finally a description of model requirements and retrievable outputs via the packages. The unified representations should allow more consistent comparisons and therefore help the modellers in their choice of methodology for a specific case study. Package documentation and source codes were thoroughly screened to conduct these analyses. This work was carried out in accordance with the comments and recommendations of most of the package authors.

2.3 ~~Full list of selected hydrological modelling R packages~~

2.2.1 Conceptual representation of storages and fluxes

Each model has its own degree of complexity regarding the representation of storages and fluxes. The differences in model structure partly depend on the perceptual model of how a catchment is functioning (e.g. Wrede et al., 2015). In our list of seven models, these differences resulted in very different modelling characteristics. One of the goals here is to present the exact modelling structure contained in the selected packages that might sometimes have been adjusted from the original versions by the package authors. We have therefore adopted a comparison method that aims at representing the main principles behind the models — or differences in perceptual model — but that still keeps a certain degree of precision. This type of unified comparison was, for example, employed in the framework for understanding structural errors (FUSE, see Fig. 3 of Clark et al., 2008)

to compare the structures of four models, or to present the forty models included in the `MARRMoT` toolbox (see Fig. 2 of Knoben et al., 2019).

We selected an approach for this analysis that derives from the work of de Boer-Euser et al. (2017, Fig. 3). This analysis aims at depicting the conceptual storages and fluxes at a spatial unit scale. More details about the diagrams are given in Sect. 4.1 and Fig. 1. This analysis was reviewed by the package authors to ensure consistency.

2.2.2 Spatial distribution

Users must be aware of the spatial discretisation that is available. Furthermore, some packages offer the possibility to apply different types of catchment discretisations for the same model. We therefore present the different cases for the selected packages after introducing a special case that is snow modelling spatialisation.

2.2.3 Requirements and outputs

Since hydrological models do not always rely on the same assumptions, their requirements, i.e. data inputs and number of adjustable parameters, can differ. As data availability can sometimes be a restraining factor, it is essential for users to be informed on the model data requirements. The packages also allow operation of the models at different time steps and imply different types of numerical resolutions of model equations. The different equations of a hydrological model can be solved using different techniques. The equations are solved analytically (the exact solution is determined by integrating the equation for a given time step), explicitly (the solution is approximated by its derivative at the beginning of the time step) or implicitly (the solution is approximated by its derivative at the end of the time step). When the solution is analytical or explicit, the operator splitting technique is commonly applied to sequentially calculate processes such as evaporation, runoff and percolation (Santos et al., 2018b). Numerical solution in hydrology can be seen as part of the mathematical model rather than software implementation, as it changes the results substantially (Clark and Kavetski, 2010; Kavetski and Clark, 2010).

By making different outputs available, R packages allow modellers to better assess the suitability of applying a model for a specific problem. It can also facilitate the evaluation of appropriate parameter estimation, i.e finding a consistent set of parameters. Among the practical outputs for a modeller, time series of actual evapotranspiration estimates can be useful to understand the behaviour of the soil moisture accounting functions. Retrieving time series of runoff components (e.g. fast runoff, very quick runoff), which are highlighted by Sect. 4.1, makes it possible to relate the model simulations with catchment regimes (e.g. high baseflow well reproduced by the slow runoff exiting the groundwater store). Internal fluxes can inform a user on the internal consistency of a simulation, for example, to identify the fraction of effective rainfall exiting the root zone store and reaching the fast runoff routine compared to the fraction entering the groundwater store. Analysing time series of store levels can, for instance, enlighten the user on whether the root zone store capacity has been correctly estimated, which would then help to analyse the simulation of soil moisture seasonality by the model for the studied catchment.

Any modeller would need to understand these specificities in order to select and apply a model. We summarize these characteristics (requirements; time step and numerical resolution; outputs) in Sect. 4.3.

2.3 Framework for analysing package practicalities

To operate the models, the different packages implement a set of functionalities, which can be more or less in line with the hydrological workflow, i.e. from data preparation to result analysis. These functionalities aim at easing and sometimes constraining the use of the model. One would expect to be able to use all the functionalities required to consistently apply a specific model and avoid any supplementary source of errors. One of the specificities of R packages is the provided documentation. The related description and examples must be complete to ensure appropriate application of the models. The user is guided for basic examples and aware of potential errors that can occur. Following the analysis of functionalities and documentation, we present an analysis of R implementation that should foster more rigorous applications of the models. In an effort to contribute to a more extensive documentation relating to the packages and their models, we provide R scripts enabling the use of each package on simple examples. A short analysis on central processing unit (CPU) times is derived from the application of these scripts.

2.3.1 Functionalities

What a package provides in terms of functionalities is a distinguishing feature when selecting a specific software or another programming language for hydrological modelling. Among the main features, we usually find the careful preparation of input data to respect the right time references, initialisation period or specific R objects. Enabling an automatic calibration procedure to find a set of parameters consistent with the catchment of study can be an important step for some models as well (though some packages have specifically avoided automatic calibration for the reasons discussed in Beven, 2012, 2016). Functions to visualise and analyse the results are often appreciated by the users. Simple analyses can be the calculation of criteria assessing the overall performance. These criteria are regularly calculated on time series of transformed data to emphasise specific error characteristics. Hydrograph plots are also common for assessing hydrological models. Graphical user interfaces can ease the package usability. As some models enable snow calculations, implementing an independent snow function is necessary to avoid using a snow function on non snowy catchments.

We present in this analysis whether the selected packages integrate these basic functionalities to consistently apply the models. Inspections of the packages were conducted based on the different types of documents related to the packages and models. When judged necessary, the codes were analysed to ensure accurate results.

2.3.2 Documentation

To handle the complexity associated with the different hydrological models as well as with the functionalities provided by the packages, the documentation is obviously essential for any user. It is therefore important to assess whether looking at the overall documentation is sufficient to easily make use of package basics. In this regard, we propose a comparison based on our use of the available explanatory documents. This analysis is by definition subjective as it relies on our experience as users. However, we think that it can still give insights on the meaningful content of the documentation. Analysing the documentation explanation by explanation would indeed be very complicated to present. There are two different types of documentation related to these

packages: the R documentation that includes user manuals (functions explanations, mandatory) and sometimes vignettes ("long-form guides that illustrate how to use packages", Slater et al., 2019); the external documentation that comprises scientific

220 journal articles and sometimes websites. For each function of a package, the formal R user manual includes mandatory fields (e.g. name, value, title, description and arguments) and optional fields (e.g. details, examples and references, for more details see R Core Te
. We consider two types of scientific articles in this analysis: articles that were written in order to present the packages; and articles using the packages and made by one of the package authors. Websites usually contain elements such as video tutorials, list of publications mentioning the package, examples, user groups. Vignettes and external documentation are not required

225 when creating a package but can be very useful for thorough understanding of the packages and models.

2.3.3 User implementation

Package practicalities can also be assessed through an analysis of the links between the main functions of a package. Such an examination could be useful to provide guidance regarding package application. We try to put ourselves in the shoes of users who have to apply the models of the different packages and therefore need to understand which function they have to

230 use, where in the script and how. In this regard, we propose a unified diagram of the connections between the main functions that we have been able to run (see Fig. 4). We use the term "user function" which means that users have to write their own R function integrating, among others, the legacies illustrated on the diagrams. This analysis is intended for users familiar with R packages and aims at guiding users in their application of the hydrological modelling R packages. We therefore provide R scripts enabling the application of each package on a simple hydrology example (supplementary documentation). The provided

235 R scripts show the basic R commands required to test one parameter set on two different catchments (see Sect. 2.3.4).

2.3.4 R structures and CPU times

Package developers made several choices in terms of R implementation that can affect package usability. For that reason, we analyse the programming languages and external dependencies. We also perform a short analysis of package CPU times.

Some packages are entirely coded in R, which is an interpreted language, and some integrate models coded with a compiled

240 programming language interfaced with R. The different programming languages interfaced with R were identified by extracting the package sources, because they could not necessarily be identified by simply displaying the code from the R console. Regarding external dependencies, we considered a package as being dependent if one of its functions cannot be run without downloading another package. A package is not considered as depending on any other package when the use of an external package is only suggested in an example or in one of the related articles. R Core Team packages, such as `stats`, are not taken

245 into account in this assessment, as they are packages installed by default.

From a user perspective, computation times can be meaningful to determine whether a package is suitable for a specific study. Short computation times are usually very well appreciated, especially when dealing with finer time steps or more complex spatial discretisations. Applying a model to a large database, generating an ensemble in operational (flood) forecasting or performing Monte Carlo runs for uncertainty analyses can also significantly increase computation times, hence some of the

250 packages include some compiled code to speed up the production runs of the model. We analysed the CPU time required for

one model run, which was estimated from 1000 runs with the `microbenchmark` package (Mersmann, 2019). We ran the packages on a computer with the following characteristics: RAM capacity: 8.00 GB; CPU: Intel i5-8250U 1.80 GHz; OS: Windows 10 (64-bit) using the 3.6.0 (64-bit) R version. The models were run at a daily time step on a catchment where high flows mostly result from precipitation events in winter, the Meuse River at Saint-Mihiel (2543 km^2, from January 1st 1990 to December 31th 1999) and, for the packages integrating a snow function, on a mountainous catchment where high flows mostly result from snowmelt in spring, the Ubaye River at Lauzet-Ubaye (943 km^2, from January 1st 1989 to December 31th 1998). The time series of precipitation and temperature at a daily time step were extracted by Delaigue et al. (2020b) from the SAFRAN countrywide climate reanalysis of Météo-France (Vidal et al., 2010). The PET time series were calculated using the Oudin et al. (2005) formula. The streamflow data were retrieved from the "Banque Hydro" database (Leleu et al., 2014). For the use of some packages, a DEM with a resolution of 25 m by 25 m was derived from the BD ALTI DEM (IGN, 2013). Only one parameter set is tested for each model.

3   An overview of the selected hydrological modelling R packages

The outcome of our selection is a list of eight packages that will be carefully compared throughout the paper. Here we give a first overview of these packages along with their related bucket-type hydrological models. The full list is presented in Table 1 with the main related documentation. Table 2 shows the snow models contained in the selected packages.

Table 1. List of the selected packages with their related models. The models included in the following analyses are in bold type. The models included in `hydromad` that are presented in this table only correspond to the main soil moisture accounting functions (for more details see Andrews and Guillaume, 2018).

Package	Version	Repository	Hydrological models	Main references of the models
airGR	1.4.3.65	CRAN	GR1A; GR2M; **GR4J**;	Mouelhi (2003); Mouelhi et al. (2006); Perrin et al. (2003);
			GR5J; GR6J;	Le Moine (2008); Pushpalatha et al. (2011);
			GR4H; GR5H	Mathevet (2005); Ficchì et al. (2019)
dynatopmodel	1.2.1	CRAN	**Dynamic TOPMODEL**	Beven and Freer (2001); Metcalfe et al. (2015)
HBV.IANIGLA	0.1.1	CRAN	**HBV**	Bergström (1976); Bergström and Lindström (2015)
hydromad	0.9-26	GitHub	**IHACRES-CMD**;	Croke and Jakeman (2004);
			IHACRES-CWI;	Jakeman and Hornberger (1993);
			Australian Water Balance;	Boughton (2004);
			GR4J; **Sacramento**	Perrin et al. (2003); Burnash (1995)
sacsmaR	0.0.1	GitHub	**Sacramento**	Burnash (1995)
topmodel	0.7.3	CRAN	**TOPMODEL 1995**	Beven and Kirby (1979); Beven et al. (1995)
TUWmodel	1.1-1	CRAN	**Modified HBV**	Parajka et al. (2007)
WALRUS	1.10	GitHub	**WALRUS**	Brauer et al. (2014a); Brauer et al. (2014b)

Table 2. List of the snow models contained in the selected packages.

Package	Snow model	Model reference
airGR	CemaNeige	Valéry et al. (2014)
dynatopmodel	None	
HBV.IANIGLA	HBV	Bergström and Lindström (2015)
hydromad	snow.sim	Andrews and Guillaume (2018)
sacsmaR	SNOW-17	Anderson (2006)
topmodel	None	
TUWmodel	Modified HBV	Parajka et al. (2007)
WALRUS	Degree-Day method;	Seibert (1997)
	Shortwave radiation method	Kustas et al. (1994)

We chose to exclude the following packages and we justify our choice:

- The `Ecohydmod` package (Souza, 2017) implements an ecohydrological model.

- The `LWF-BROOK90` package (Schmidt-Walter et al., 2020) implements a physically-based land-surface hydrological model.

- The `fuse` package (Vitolo et al., 2016b) proposes a large number of model structure configurations. It was considered that its main purpose was not to conduct a basic hydrological study but more to understand errors arising from hydrological models. It is also in need of active maintenance.

- The `RHMS` package (Arabzadeh and Araghinejad, 2019) implements several event-based hydrological models. This package is not included in this work as we chose to include only the continuous models.

- The `SWATmodel` package (**?**) implements a complex watershed hydrological transport model. This package does not provide any function for data preparation or any explanatory document.

airGR

The `airGR` package (Coron et al., 2020) implements the models constituting the suite of *Génie rural* (GR) hydrological models (Coron et al., 2017) which originate in the work of Claude Michel starting in the 1970s (Michel, 1983). These models are parsimonious conceptual rainfall-runoff models that consider a catchment as a single entity (lumped). Several versions were developed over the years from the well-known GR4J (Perrin et al., 2003) to the GR6J model (Pushpalatha et al., 2011) for improved low-flow simulations. A snow-accounting model called CemaNeige (Valéry et al., 2014) can be combined with the daily and hourly GR models or can also be operated independently. `airGR` includes a function to calculate potential evapotranspiration time series with the equation of Oudin et al. (2005). Various technical features associated with the hydrological

workflow from data pre-processing work to result analysis are offered. For the sake of brevity, only GR4J combined with CemaNeige will be assessed in the following analyses. `airGR` has a graphical user interface in the complementary package `airGRteaching` (Delaigue et al., 2018, 2020a), which will be analysed along with `airGR`.

topmodel

290 The Topography-based hydrological model (TOPMODEL, Beven and Kirby, 1979) has been largely employed for a variety of applications throughout the years (Beven et al., 2020). The TOPMODEL version included in `topmodel` (Buytaert, 2018) follows the version developed by Beven et al. (1995) that makes explicit assumptions about the nature of the near surface water table responses that lead to the possibility of using a topographic wetness index (TWI) as an index of hydrological similarity to calculate surface saturation and moisture deficits. Calculations are made for different increments of the distribution of the index,

295 making the model computationally fast to run. The pattern of the index can be derived from an analysis of a digital elevation model (DEM), and can be used to map the simulated response back into the space of the catchment. `topmodel` allows simple calculations from a DEM and basic data series required in conceptual hydrological modelling. TOPMODEL allows for saturated contributing areas to be predicted based on the spatial distribution of the topographic index. These assumptions mean that it is best suited to moderately sloping hillslopes with relatively shallow water tables (though see Quinn et al., 1991, for an

300 application to a deeper system).

dynatopmodel

Driven by a desire to relax some of the assumptions of TOPMODEL, the authors proposed a new version: the dynamic TOPMODEL (Beven and Freer, 2001). In the original version, simulations of subsurface flows depend on a quasi-steady state assumption for the redistribution of moisture at each time step (Beven, 1997). Dynamic TOPMODEL relaxes this assumption

305 to a non-steady kinematic wave solution for subsurface flows (between the similarity units), and allows other geographical information to be taken into account in the discretisation of the catchment, but with a similar aim of grouping parts of the catchment into computational units for efficiency. `dynatopmodel` (Metcalfe et al., 2018) includes this model and offers the technical features to prepare the basic data required to run the dynamic TOPMODEL. For instance, a function to calculate potential evapotranspiration time series following the equation of Calder (1986) is included.

310 ### HBV.IANIGLA

The HBV model (Bergström, 1976) has been improved over the years, one of its most employed version being the HBV-96 (Lindström et al., 1997). The `HBV.IANIGLA` package (Toum, 2019) enables application of each component (i.e. snow, soil moisture and routing) of the HBV model independently. Other types of snow, soil moisture functions and routing functions are also implemented, deriving from the HBV model (see Toum, 2019). This package also includes functions to calculate variables

315 such as potential evapotranspiration with the method of Calder et al. (1983) or glacier discharge with the equations of Jansson et al. (2003).

hydromad

The hydrological model assessment and development package, `hydromad` (Andrews et al., 2011; Andrews and Guillaume, 2018), suggests two ways for treating rainfall-runoff modelling: either a single rainfall-runoff model is considered, which can be a model such as Sacramento (Burnash, 1995), or an effective rainfall framework, distinguishing between a soil moisture accounting (SMA) and a routing step as in IHACRES (Jakeman and Hornberger, 1993). A user has to choose the combination that best suits their requirements. `hydromad` includes eleven soil moisture accounting functions and six routing modules. A snow-accounting function can be added to the calculations when the IHACRES-CMD SMA is selected. Several functions for data pre-processing, calibration and post-treatment are made available by the package. In our next analyses, for conciseness purposes, we will only apply the Sacramento, IHACRES and GR4J models.

sacsmaR

The `sacsmaR` package (Taner, 2019) implements the well-known Sacramento soil moisture accounting model (SAC-SMA). In its original version, the SAC-SMA model was set with lumped parameters. In the `sacsmaR` package, the model can be run in a semi-distributed way~~(see Sect. 4.2).~~ There is yet no preprocessing function included in the package to deal with the spatial discretisation required to run the semi-distributed version of SAC-SMA. A snow-accounting module, SNOW-17 (Anderson, 1976, 2006) can be run along with the SMA and will be considered in our applications. Two other functions are implemented in the package: a routing function based on Lohmann et al. (1996) and a function to calculate potential evapotranspiration time series based on the Hamon (1960) formulation.

TUWmodel

A modified version of the HBV rainfall-runoff model (Bergström, 1976) is implemented in `TUWmodel` (Parajka et al., 2007; Viglione and Parajka, 2020). HBV is composed of a snow routine, an SMA routine and a flow routing routine. The model can represent rainfall-runoff transformation in a lumped or semi-distributed way. In comparison to other HBV versions, it does not implement glacier melt modelling, refreezing of snow pack, separation of vegetation in different elevation zones or lake impact on river flow (https://www.smhi.se/en/research/research-departments/hydrology/hbv-1.90007).

WALRUS

The `WALRUS` package (Brauer et al., 2017) contains the Wageningen lowland runoff simulator (WALRUS), a water balance rainfall-runoff model that was specifically designed for catchments with shallow groundwater (Brauer et al., 2014a, b). This model assumes that each parameter has a physical meaning at the catchment scale (in a qualitative sense). The WALRUS authors introduced the model as an alternative to those mainly developed for sloping basins (Brauer et al., 2014a) to better account for essential processes in lowlands such as capillarity rise and groundwater-surface water interactions. The package offers several functions in line with the hydrological workflow. Snow accumulation and melt can be calculated with one of the package functions prior to the model simulations.

4 A unified analysis of the hydrological models proposed in R packages

Investigating the different hydrological characteristics behind the models contained in the R packages is a difficult but useful exercise. It aims at gathering information about the various hydrological visions available in a comparative framework. It is relevant to proceed with this comparison task to help any student or more experienced hydrologist to understand what is involved when using a specific model implemented in one of the R packages. The selected packages contain various hydrological models based on different assumptions. These assumptions can be sorted out into simplification options regarding storages, fluxes, time and space. In this comparative study, we first propose a unified comparison of the conceptual representation of storages and fluxes by the models included in the selected packages, then the spatial discretisation they imply and finally a description of model requirements and retrievable outputs via the packages.The unified representations should allow more consistent comparisons and therefore help the modellers in their choice of methodology for a specific case study. This work was carried out in accordance with the comments and recommendations of most of the package authors.

4.1 Conceptual representation of storages and fluxes through different model structures

Each model has its own degree of complexity regarding the representation of storages and fluxes. The differences in model structure partly depend on the perceptual model of how a catchment is functioning (e.g. Wrede et al., 2015). In our list of seven models, these differences resulted in very different modelling characteristics. One of the goals here is to present (Fig. 1) the exact modelling structure contained in the selected packages that might sometimes have been adjusted from the original versions by the package authors. We have therefore adopted a legend that aims at representing the main principles behind the models — or differences in perceptual model — but that still keeps a certain degree of precision. This type of unified comparison was, for example, employed in the framework for understanding structural errors (FUSE, see Fig. 3 of Clark et al., 2008) to compare the structures of four models. Their legend integrates the same store and flux designations and a similar layout to help visualising the different structures. In addition, the forty models included in the MARRMoT toolbox were uniformly compared using a table that summarises the main components of each model (e.g. which models contain a soil moisture store, see Fig. 2 of . As our comparison includes fewer modelsthan the MARRMoT toolbox, we selected a more precise approach based on a different legend from FUSE for a better visualisation of stores and fluxes. This legend was modified from de Boer-Euser et al. (2017, Fig. 3). The diagrams depict the conceptual

The diagrams of Fig. 1 depict the conceptual storages and fluxes at a spatial unit scale (e. spatial unit scale (e.g. at the catchment or sub-catchment scale). They were reviewed by the package authors to ensure consistencyFor these diagrams, the root zone storage corresponds to the soil moisture accounting or production function. Groundwater accounts for saturated soil zones and shallow aquifers involved in the catchment response. Fast runoff is similar to lateral flow or interflow. Bi-coloured rectangles are for two storages/fluxes modelled by the same store or by the same function simultaneously. Please note that for the semi-distributed models, the schemes only contain the storages and fluxes calculated on a single spatial unit. Details on the input data are given in section 4.3. We provide further explanations for each model hereafter.

GR4J of `airGR` and `hydromad`. The snow box is only for CemaNeige of `airGR`. Modified from de Boer-Euser et al. (2017).

Model of `WALRUS`, modified from de Boer-Euser et al. (2017).

TOPMODEL 1995 of `topmodel`.

Dynamic TOPMODEL of `dynatopmodel`.

Sacramento of `sacsmaR` and `hydromad`. The snow box is only for `sacsmaR`.

HBV of `TUWmodel` and `HBV.IANIGLA`.

IHACRES-CMD of `hydromad`.

Snow
Interception
Root zone storage
Very quick runoff
Fast runoff
Groundwater
Surface water
Lag function
Losses or gains
In: precipitation
Out: evaporation
Transpiration
Generated discharge
Internal fluxes

Figure 1. Unified diagrams illustrating the depiction of conceptual storages and fluxes by the main models contained in the selected packages. ~~The root zone storage usually corresponds to the soil moisture accounting or production function. Groundwater accounts for saturated soil zones and shallow aquifers involved in the catchment response. Fast runoff is similar to lateral flow or interflow. Bi-coloured rectangles are for two storages/fluxes modelled by the same store or by the same function simultaneously. Please note that for the semi-distributed models, the schemes only contain the storages and fluxes calculated on a single spatial unit.~~

GR4J-CemaNeige

The GR4J model combined with the CemaNeige snow model which are both included in `airGR`: total precipitation is first divided into solid and liquid precipitation by the snow function. Solid precipitation enters the snow accumulation store (light blue rectangle). Snow melt (from the light blue rectangle) and liquid precipitation are added together to calculate interception (blue rectangle) considering potential evapotranspiration (PET). Then, either a remaining PET component is used to calculate evapotranspiration withdrawn (blue ~~arrow~~and green arrows) from the production store (green rectangle) or a part of the liquid precipitation remaining from the interception calculations either fills the production store or enters the very quick runoff and fast runoff unit hydrographs (UHs). A percolation component from the production store also joins the very quick (yellow rectangle) and fast runoff (orange rectangle) UHs. The output of the fast runoff UH fills a routing store. Water volumes can be added or withdrawn to/from the routing store or the very fast runoff component. This function accounts for groundwater contribution to runoff. The flow rate from the routing store is then added to the very fast runoff component to form the final discharge value at a particular time. The GR4J model included in `hydromad` is almost identical to its implementation in `airGR`. The difference is about the fraction of water entering the fast runoff UH. This fraction was empirically set to 0.9 in `airGR`, whereas this default value can be modified by the user in `hydromad`. The `hydromad` package does not propose a snow function to be combined with GR4J.

~~WALRUS: precipitation~~

WALRUS

Precipitation is divided into solid or liquid water for calculation of snow accumulation and melt. Liquid precipitation and melt resulting from the snow function can either directly join the surface water reservoir (red rectangle) or enter the wetness index calculation. The wetness index determines the fraction of water infiltrating in the soil reservoir, which contains both the vadose zone and saturated zone (green/brown reservoir) or joining the linear quickflow reservoir (yellow to orange gradient rectangle) that supplies the surface water reservoir. Evapotranspiration is retrieved from the surface water reservoir and from the vadose zone both as a function of PET and water contents. WALRUS integrates an explicit representation of the dynamic water table in shallow groundwater of lowland areas. The vadose zone concurrently interacts with the groundwater through the dynamic water table in the same reservoir. The overall saturation of the soil reservoir is governed by the dryness of the vadose zone, which determines the wetness index. The groundwater table depth is compared to the surface water level to determine either drainage towards the surface water or infiltration from the surface water. Discharge is a function of the surface water level. Losses and gains can occur from/to the groundwater reservoir by seepage and from/to the surface water by extraction or surface water supply.

TOPMODEL 1995

As briefly introduced in Sect. ~~??~~3, two packages contain two different versions of TOPMODEL. Their singularities especially lie in the spatial distribution and calculations of subsurface contributions to streamflow. In terms of conceptual storages and

fluxes, some small differences are highlighted by our schematics. We first describe the water paths from inputs to outputs of TOPMODEL 1995 and then present the differences brought by the dynamic TOPMODEL. Spatial considerations are dealt with in Sect. 4.2.

415   TOPMODEL 1995 version of `topmodel`: precipitation infiltrates first in the interception/root zone store (green to blue colour gradient) where the actual evapotranspiration to be removed is calculated. When storage in the root zone is above a field capacity threshold, water is added to a drainage store (green rectangle) and recharge to the water table is calculated. At the end of each time step the configuration of the saturated zone (brown rectangle) is updated according to the topographic index distribution, as if the storage was in steady state with the drainage rate. On the saturated contributing area, or where

420   the unsaturated zone is filled from above, an excess flow is transmitted to the overland routine (yellow to orange colour gradient). Consequently, the overland routine deals with storage excess coming from the saturated zones, routes the runoff on the hillslopes and generates a part of the flow that will then be routed by the channel routing. The saturated zone drainage reaches the channel baseflow and will thus be routed along with the surface runoff. A constant celerity time delay function (or lag function) is applied to route the sum of these two flows to the catchment outlet.

425   ## Dynamic TOPMODEL

The dynamic version of TOPMODEL (Beven and Freer, 2001) is implemented in the `dynatopmodel` package: conceptual storages and fluxes of the dynamic TOPMODEL are represented without taking the semi-distributed spatialisation into account (i.e. on a single hydrological response unit). Spatial characteristics of the package models will be dealt with in Sect. 4.2. The difference in terms of storages and fluxes between the model in the `topmodel` package and the model in the `dynatopmodel`

430   package concerns the subsurface runoff and the water table. In the 1995 version of TOPMODEL, the water table is represented as a succession of quasi-steady states, whereas the dynamic TOPMODEL includes a ~~time-dependant~~ time-dependent kinematic routing (Beven and Freer, 2001; Metcalfe et al., 2015). The saturated zone (brown rectangle) water level is predicted using implicit kinematic routing between (and within) the spatial computational units. When, within a unit, the local storage capacity is reached, any excess water is routed to downslope units (as "run-on") or a connected river reach. Runoff components from the

435   interception/root zone store, the unsaturated zone and the saturated zone are added together (yellow to orange colour gradient rectangle) and then routed to the outlet by a constant celerity time-delay histogram.

Sacramento

The Sacramento model of `sacsmaR` and `hydromad`: snow calculations (precipitation separation and snow accumulation and melt) prior to liquid water inputs of the hydrological model are only operated when using the `sacsmaR` package. The

440   Sacramento model represents the soil with two main layers, a thin upper layer and a thicker lower layer. The upper layer contains two reservoirs (green to yellow and green to orange gradient rectangles) and the lower layer has three reservoirs (brown rectangles). Liquid water enters the first root zone store of the upper layer (green part of the green to yellow gradient rectangle), infiltrates through the second root zone store (green part of the green to orange gradient rectangle) and then reaches the lower soil layer where the three reservoirs are interconnected. Evaporation can occur from both the upper soil layer and

445   the channel (blue arrow from the final red arrow). Plant transpiration can exit the upper soil layer and the lower soil layer. A very quick runoff component originates from the first root zone store (yellow part of the rectangle), which accounts for impervious area runoff. The second root zone store produces interflow and another surface runoff component (both represented by fast runoff, i.e. the orange part of the rectangle). The lower layer contributes to the baseflow channel component and to a subsurface outflow lost by the model (brown arrow exiting the model). The baseflow channel component, the very quick runoff

450   and the two fast runoff flows are added together to form the final river discharge. A lag function can be applied on the final discharge. This function is based on Lohmann et al. (1996) when using the `sacsmaR` package. The `hydromad` package offers several routing functions that can be applied as well.

HBV

In terms of conceptual storages and fluxes, the HBV model of `TUWmodel` and `HBV.IANIGLA` are similar. Precipitation is first

455   divided into snowfall and rainfall. Snowfall goes to the snow routine (light blue rectangle) which calculates snow accumulation and melt. The part of snow that melts and rainfall become inputs of the root zone storage (blue to green gradient rectangle). The soil moisture accounting generates runoff and calculates actual evapotranspiration by taking potential evapotranspiration into account. The runoff generation routine consists from one upper and one lower reservoirs with three outflows representing overland flow, interflow and baseflow. These runoff components are then routed by a triangular transfer function (red triangle,

460   for more details please see Parajka et al., 2007). This function lags the overall flow volumes resulting from these three to form the final discharge value. The differences between HBV of `HBV.IANIGLA` and HBV of `TUWmodel` are as follows: `HBV.IANIGLA` offers the possibility to take glacier discharge into account in the snow calculations; `TUWmodel` distinguishes the temperature above which precipitation is liquid from the temperature below which precipitation is solid; the time constant of the triangular function corresponds to one parameter in `HBV.IANIGLA`, while it is derived from two different parameters

465   in `TUWmodel`.

~~IHACRES-CMD of `hydromad`:~~

IHACRES-CMD

A simple degree-day factor snow model (light blue) feeds, with melt or liquid precipitation, into a catchment moisture deficit model that represents soil moisture accounting (green). Evapotranspiration occurs from this store. The resulting effective

470   rainfall is passed to a unit hydrograph, typically consisting of two flowpaths (very quick/fast, and slower groundwater), but with potential for other configurations. These two runoff components are then added together to form the final discharge value.

Synthesis

This unified representation of the model structures in terms of conceptual storages and fluxes reveals certain trends in the different modelling choices. Although it is clear that each structure has its own specificities, the schematics highlight sev-

475   eral modelling similarities. When snow is taken into account (WALRUS, GR4J-CemaNeige, Sacramento, TUWmodel and

IHACRES-CMD), the related calculations respect similar steps where total rainfall (solid + liquid) is divided into solid precipitation, which supplies a snow cover storage, and liquid precipitation joining the hydrological model. These calculations follow a degree-day approach except for the snow model included in the `sacsmaR` package, which relies on a snow energy balance equation. WALRUS allows either a degree-hour-factor method or a shortwave-radiation-factor method to be used. Both methods do not solve the energy balance equation. Three models, dynamic TOPMODEL, TOPMODEL 1995 and TUWmodel take the interception process into account with the root zone store related calculations to reduce the number of parameters to be determined. Fast and very quick runoff are considered as two distinct components for GR4J-CemaNeige, TUWmodel and Sacramento. Apart from GR4J-CemaNeige, discharge sources are separated into a slow contribution from groundwater that can be identified as baseflow and a surface runoff input. These two components are added together to form the final river discharge value and sometimes, if not applied separately before the addition (dynamic TOPMODEL and IHACRES-CMD), a lag function is employed on the overall resulting flow. WALRUS does not include such a function. Sacramento has a finer representation of soil layers compared to the other models.

4.2 Which spatial distribution for which model?

Users must be aware of the spatial discretisation that is available. Furthermore, some packages offer the possibility to apply different types of catchment discretisations for the same model. We therefore present the different cases for the selected packages after introducing a special case that is snow spatialisation.

4.2.1 The case of snow

As presented in the previous section, some packages enable application of a snow function along with the hydrological models they include (`airGR`, `HBV.IANIGLA`, `hydromad` only for IHACRES-CMD, `sacsmaR`, `TUWmodel` and `WALRUS`). Depending on the package of interest, taking The influence of snow processes on streamflow can vary with elevation, as snow accumulation and melt into account does not involve the same spatial distribution mainly depend on air temperature that usually decreases with elevation and precipitation that usually increases with elevation. For that reason, a spatial discretization within the catchment may be needed to better account for snow influence when modelling streamflow at the outlet of a catchment. Some packages propose a spatial discretization to account for the influence of snow processes on streamflow. Four configurations were found possible.

All these packages allow to proceed with snow calculations considering the catchment as a single unit. In that case, input data are aggregated at the catchment scale. `HBV.IANIGLA`, `hydromad` and `WALRUS` do not offer any other possibility regarding the spatial distribution of snow processes. The CemaNeige model of `airGR` is applied on different elevation zones of the catchment in order to take into account the important heterogeneity of snow. The elevation bands have the same surface area (see Fig. 2). They are derived from the quantiles of the basin hypsometric curve that must be provided to `airGR`. Precipitation and temperature data are interpolated for each zone and become inputs of the CemaNeige model. There is one set of parameters for the whole basin. The spatial distribution of snow processes by the TUWmodel follows a similar TUWmodel and `sacsmaR` (SNOW-17 module) packages follow another principle, the difference being that the elevation zones are determined by dividing

the maximal elevation into equal ranges ~~can be set with different ranges and with different surface areas~~ (e.g. ~~each 20 m on~~ Fig.
2)~~, thus forming different surface areas. Precipitation, potential evapotranspiration and temperature data inputs are spatially interpolated by the user to get values for each elevation zone. This pre-processing step has to be completed before using the main function of the `TUWmodel` package.~~. Model parameters can be differentiated across elevation zones.

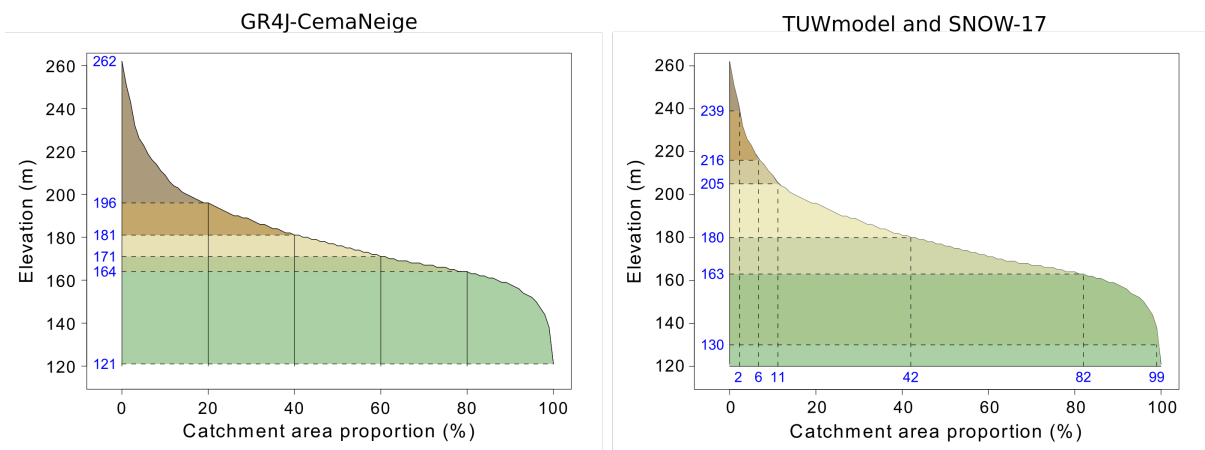



Figure 2. Example of GR4J-CemaNeige elevation zones (left) and TUWmodel/SNOW-17 elevation zones (right) both following the hypsometric curve of the Couetron River at Souday (France). Each colour indicates a different elevation zone.

~~The spatial discretisation required to use the snow function contained in `sacsmaR` follows the spatial distribution of the Sacramento model, which will be further discussed in this section. In order to use the snow function, elevation bands can be specified. These elevation bands do not have to be set with equal ranges or equal areas.~~

4.2.2 From lumped models to complex semi-distributions

In the case of our selected models, the packages theoretically allow one or more of the spatial discretisation configurations illustrated on Fig. 3. Table 3 summarises the possible configurations for each model contained in the selected packages.

When the models are applied with a lumped spatial configuration, inputs of precipitation and potential evapotranspiration are aggregated on the whole catchment. There is one set of parameters, which means that the model reservoirs represent the water content at the catchment scale. The model simulates a discharge output at the catchment outlet where the hydrometric record station is located.

TOPMODEL 1995 does not rely on the same calculations as dynamic TOPMODEL, especially regarding the computational units. In this implementation of TOPMODEL 1995, inputs of precipitation and potential evapotranspiration are aggregated over the entire catchment (as a lumped model) although in the original paper (Beven and Kirby, 1979) different inputs and TWI distributions were applied in different subcatchments. A single parameter set is defined at the catchment scale. Routines are provided for processing a digital elevation model to calculate a topographic wetness index for each grid cell (as a distributed model). The digital elevation model should have a resolution of less than 30 m for the results to be meaningful (Beven,

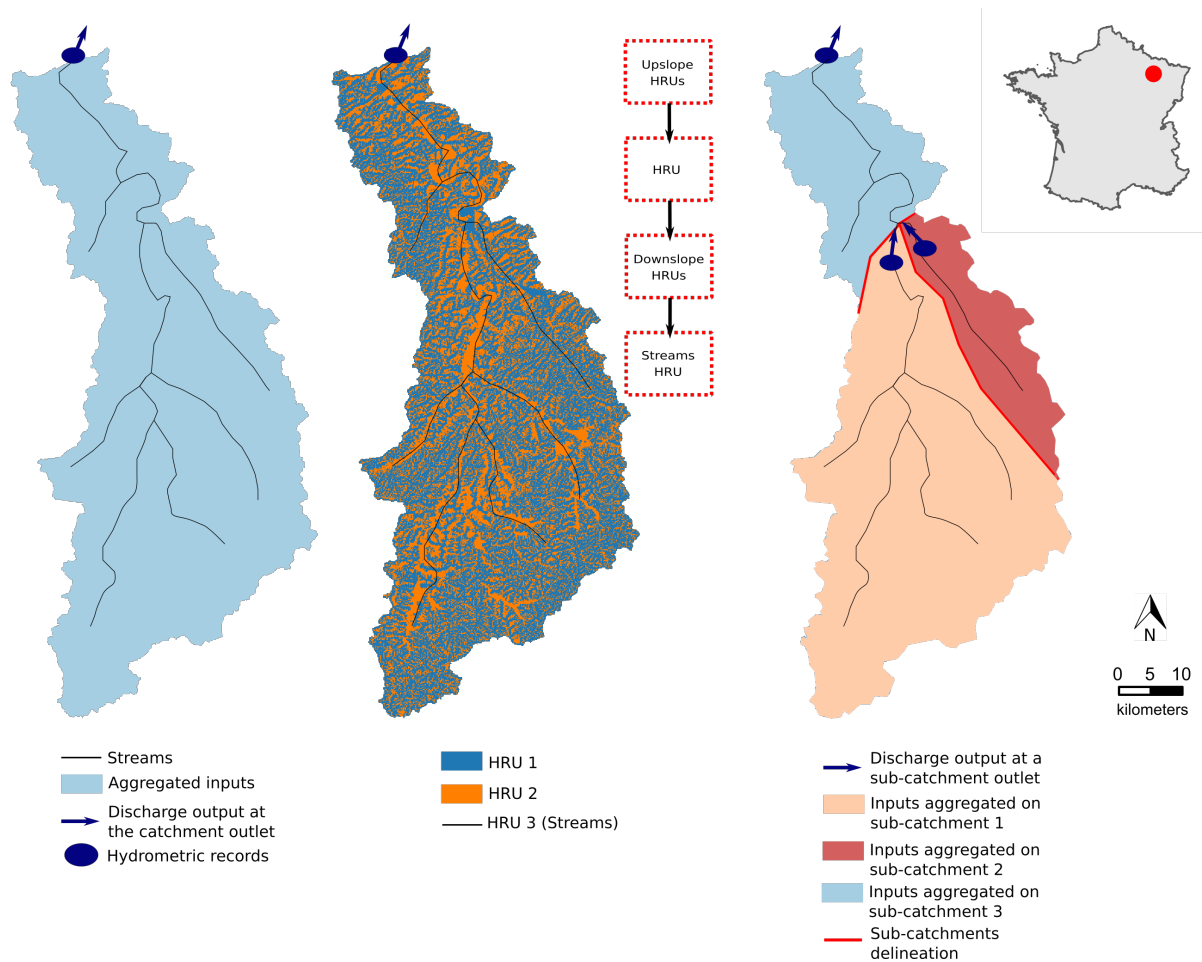



Figure 3. Illustration of the three possible spatial discretisations concerning the models contained in the selected packages for this study. From left to right, lumped configuration, hydrological response units (HRUs) configuration and sub-catchments configuration. The catchment outline is from the Meuse River at Saint-Mihiel (France). The HRUs were generated using a function of the `dynatopmodel` package and are defined by the upslope area contribution based on the topographic index.

2012). Cells with similar values of the topographic index are then bundled to create computational units. Each unit has specific reservoirs while the saturation zone is represented as a global saturation value at the catchment scale. These units are interconnected through the subsurface store updating based on the TOPMODEL theory and produce runoff and baseflow values to generate the final discharge time series (see Fig. 1). They can be seen as a particular case of hydrological response units (HRUs) (HRUs or Hydrological Similarity Units, HSUs in Beven and Freer, 2001) resulting from explicit assumptions about the process response. Dynamic TOPMODEL enables application of other types of HRUs HSUs that can be dependent on very different conditions, such as soil properties, land use but also the components of the topographic index. Fluxes between HRUs HSUs are controlled by a "flux-distribution" matrix based on the connectivity between the grid squares of the base digital

Table 3. Possible spatial configurations for each model. If a package allows a specific configuration (✓), it means that the model is coded for this configuration in the related package but it does not mean that the necessary pre-processing functions are provided. (∽) is for models following a spatial discretisation close to one of the categories.

Package	Model	Lumped	HRUs	Sub-catchments	Routing between HRUs and/or subcatchments
airGR	GR4J	✓	✗	✗	
dynatopmodel	TOPMODEL (dynamic)	✗	✓	✗	flux-distribution matrix (Metcalfe et al., 2015)
HBV.IANIGLA	HBV	✓	✗	✗	
hydromad	GR4J	✓	✗	✗	
hydromad	IHACRES-CMD	✓	✗	✗	
hydromad	Sacramento	✓	✗	✗	
sacsmaR	Sacramento	✓	✓	✓	hydraulic routing (Lohmann et al., 1996)
topmodel	TOPMODEL (1995)	∽	∽	✗	subsurface store updating (Beven and Kirby, 1979)
TUWmodel	Modified HBV	✓	∽	∽	weighted sum
WALRUS	WALRUS	✓	✗	✗	

elevation map contributing to the ~~HRUs~~ HSUs (for more details see Metcalfe et al., 2015, 2018). This also allows for connectivity between grids within the same ~~HRU~~ HSU. Inputs can be spatially distributed if needed by associating each ~~HRU~~ HSU with different rainfall and evapotranspiration data. ~~HRUs~~ HSUs thus have their own reservoirs. When it is required, a different

540 parameter set can be assigned to every ~~HRU~~ HSU.

The HBV model of `TUWmodel` enables a very straightforward spatial configuration where the model is run independently on different zones (with different parameters and inputs), which can be subbasins, elevation zones or any area defined by the user. ~~The~~ For example, a catchment can be divided into three subbasins, with one subbasin divided into five elevation zones. The relative contribution of each spatial entity to the entire catchment is defined by the user with a weighting coefficient.

545 The discharge outputs from each zone are then summed up ~~based on their relative area to the entire catchment~~ using these coefficients. The Sacramento model of `sacsmaR` can be applied in different ways. During a preprocessing step (not provided by the package functions), the catchment can be divided into sub-catchments that can also include hydrological response units. The `sacsmaR` package then enables assignment of a different set of parameters to each HRU as well as different data inputs. The water is run upstream to downstream through a hydraulic routing function based on Lohmann et al. (1996).

550 A large proportion of the packages that we have selected contain models that can be run as lumped models though some of them can rely on more complex spatial distribution with very specific characteristics. The most complex level of spatial distribution is enabled by the `sacsmaR` package (HRUs + sub-basins). Theoretically, it would be possible to run every lumped model on subbasins independently and sum the outputs with weights as permitted by one of the `TUWmodel` functions. We

have noticed that there are thin boundaries between the different spatial configurations. One would hardly acknowledge the differences between a computational unit of TOPMODEL 1995 and ~~HRUs~~ HSUs of dynamic TOPMODEL defined by the upslope area calculations. Nevertheless, these specificities can have a great influence on the final result and consequently the interpretations deriving from it.

4.3 ~~Model requirements and outputs~~

~~Since the models contained in the different packages of this study do not rely on the same assumptions, their requirements, i.e. data inputs and number of adjustable parameters, differ. Furthermore, the packages allow operation of the models at different time steps and enable the retrieval of a variety of outputs. As data availability can sometimes be a restraining factor, it is essential for users to be informed on the model data requirements. By making different outputs available, R packages allow modellers to better assess the suitability of applying a model for a specific problem. It can also facilitate the evaluation of appropriate parameter estimation, i.e finding a consistent set of parameters. Among the practical outputs for a modeller, time series of actual evapotranspiration can be useful to understand the behaviour of the soil moisture accounting functions. Retrieving time series of runoff components~~ Please note that the high level of spatial discretization enabled by some of the packages sometimes requires a demanding pre-processing to be carried out outside of the corresponding package (e.g. ~~fast runoff, very quick runoff), which are highlighted by Fig. 1, makes it possible to relate the model simulations with catchment regimes (e.g. high baseflow well reproduced by the slow runoff exiting the groundwater store) . Internal fluxes can inform a user on the internal consistency of a simulation, for example, to identify the fraction of effective rainfall exiting the root zone store and reaching the fast runoff routine compared to the fraction entering the groundwater store. Analysing time series of store levels can, for instance, enlighten the user on whether the root zone store capacity has been correctly estimated, which would then help to analyse the simulation of soil moisture seasonality by the model for the studied catchment. Any modeller would need to understand these specificities in order to select and apply a model. Table **??** summarises the requirements associated with the studied models and the retrievable outputs managed by the packages~~`sacsmaR`, see section 5 and Table 6). In addition, please note that the recently released version of `airGR` (v. 1.6.10.4) allows for semi-distributed modelling at the sub-catchment scale using a simple lag. As this version of airGR was released after the realisation of the present analysis, it is not included in Table 3.

4.3 Model requirements and outputs

4.3.1 Inputs and number of adjustable parameters

~~Table **??**~~ Table 4 highlights the minimum requirements that have to be supplied to run one of the models. Other inputs can be used to increase model accuracy such as satellite snow data to constrain the calibration of the snow function in `airGR`, snow cover areas for enhanced snow inclusion in `HBV.IANIGLA` or data concerning groundwater and surface water supply or withdrawals when running `WALRUS`. A digital river network can be used to set up the dynamic TOPMODEL in digital terrain analysis.

The `HBV.IANIGLA` package includes 5 configurations of the HBV routing routine. These functions rely on either 3 or 5 parameters to be estimated. The number of adjustable parameters associated with IHACRES-CMD depends on the selected routing function. When applying the exponential components transfer function with the structure identified in Jakeman et al. (1990), 6 parameters need to be estimated. For some models, several parameters may not require parameter estimation procedures but physical determination depending on the user's need and access to additional data. For instance, in the `WALRUS` package, a minimum of three parameters requires the use of estimation procedures, three parameters can either be calibrated or physically determined and the other ones are derived from physical properties of the catchment. The snow function of `WALRUS` has fixed parameters.

~~Through this analysis of model requirements and outputs, we have seen that all the~~ The two versions of TOPMODEL require an analysis of digital terrain data hence more preprocessing work. TUWmodel and Sacramento have the highest number of parameters to adjust, however, 5 parameters out of 15 for `TUWmodel` and 10 out of 23 for `sacsmaR` control the snow routine.

4.3.2 Time steps and numerical resolution of model equations

The differences in terms of time step and resolution of model equations are summarized in Table 4. All the packages give the possibility to use their models at a daily time step. Some of them allow a total flexibility (`dynatopmodel`, `HBV.IANIGLA`, `topmodel`, `WALRUS`), which might result in errors if not correctly handled by the user. `WALRUS` runs with adaptive computational time steps that may differ from the input/output time steps. It is recommended to use `dynatopmodel` and `topmodel` only at a sub-daily time step (Beven, 1997; Metcalfe et al., 2015). `airGR`, `hydromad` and `TUWmodel` also allow time step flexibility but with some constraints. ~~The two versions of TOPMODEL require an analysis of digital terrain data hence more preprocessing work. TUWmodel and Sacramento have the highest number of parameters to adjust, however, 5 parameters out of 15 for~~ `dynatopmodel` implements an explicit resolution of the root and unsaturated zones equations but an implicit resolution of the kinematic wave equation between HSUs. In `TUWmodel` ~~and 10 out of 23 for~~, part of the equations are solved analytically (e.g. the outflows of the upper and lower reservoirs, see Fig. 1). The other equations are solved explicitly (e.g. root zone storage). Most of the packages rely on operator splitting for the resolution of model equations (all except ~~sacsmaR~~`dynatopmodel` ~~correspond to snow considerations.~~ and `topmodel`).

4.3.3 Outputs

Table 5 summarises the retrievable outputs managed by the packages. While few packages allow the retrieval of all internal fluxes, most allow the user to retrieve time series of actual evapotranspiration and runoff components. Packages implementing semi-distributed models except `sacsmaR` enable retrieval of outputs at the spatial unit scale (e.g. topographic grid for TOPMODEL 1995).

The variety of models presented in this comparison are based on similar but specific assumptions in terms of storages, fluxes and spatial discretisation. The models are formulated based on our knowledge of these properties and their spatial implications. For instance, predictions of TOPMODEL and dynamic TOPMODEL can be mapped back into space because of the direct rout-

Table 4. Requirements to run the models and associated numerical resolutions. D = daily; H = hourly; M = monthly; A = annual; FL = flexible; Num. res. = Numerical resolution; OS = operator splitting; Ana = analytic; Exp = explicit; Imp = implicit; P = precipitation; T = air temperature; PET = potential evapotranspiration; DEM = digital elevation model; SA = subbasins area; hypso = hypsometric curve; TS = time series. Between parentheses: parameters or inputs of the corresponding snow routine. It is not compulsory to provide the snow routines with the hypsometric curve but it is strongly recommended when enabled by one of the packages. In this table, for the semi-distributed models, the parameters are considered uniform over the spatial units; in case they are considered distributed, the amount of parameters should be multiplied by the number of spatial units (i.e. HRUs, subbasins...).

Package	Model(s)	Time step(s)	Num. res.	OS	Inputs TS	Inputs Static	Nb. of param.
airGR	GR models	H; D; M; A	Ana	✓	P; PET; (T)	(hypso)	[1 ; 6] (+2)
dynatopmodel	Dynamic TOPMODEL	FL	Imp & Exp	✗	P; PET;	DEM	8
HBV.IANIGLA	HBV	FL	Exp	✓	P; PET; (T)		[7 ; 9] (+4)
hydromad	GR4J	D	Ana	✓	P; PET		4
	IHACRES-CMD	FL	Ana	✓	P; PET; (T)		6 (+7)
	Sacramento	≥ H	Exp	✓	P; PET		13
sacsmaR	Sacramento	D	Exp	✓	P; PET; (T)	SA; (hypso)	13 (+10)
topmodel	TOPMODEL 1995	FL	Exp	✗	P; PET	DEM	10
TUWmodel	Modified HBV	≤ D	Exp & Ana	✓	P; PET; (T)	SA; (hypso)	10 (+5)
WALRUS	WALRUS	FL	Exp	✓	P; PET; (T)	soil type	3

Table 5. Model outputs made available by the packages. TS = time series; AET = actual evapotranspiration; RC = runoff components; ⌢: only some of the time series of runoff components, internal fluxes or store levels are provided. Between parentheses: outputs of the corresponding snow routine. All the packages return time series of discharge.

Package	Model(s)	TS of AET and TS of RC	TS of internal fluxes		TS of store levels	Spatially distributed	
airGR	GR models	✓	✓	(✓)	✓	✗	(✓)
dynatopmodel	Dynamic TOPMODEL	✓	⌢		✓	✓	
HBV.IANIGLA	HBV	✓	⌢	(✓)	✓	✗	(✗)
hydromad	GR4J	✓	⌢		✓	✗	
	IHACRES-CMD	✓	⌢	(✓)	✓	✗	(✗)
	Sacramento	✓	✗		✓	✗	
sacsmaR	Sacramento	✗	✗	(✗)	✗	✗	(✗)
topmodel	TOPMODEL 1995	✓	⌢		✗	✓	
TUWmodel	Modified HBV	✓	⌢	(✓)	✓	✓	(✓)
WALRUS	WALRUS	✓	✓	(✗)	✓	✗	(✗)

ing on the hillslopes — either implicit for TOPMODEL or explicit for dynamic TOPMODEL. It is a significant difference in terms of representing "processes" in relation to "catchment characteristics" with other models relying on independent HRUs. These assumptions are not valid for every catchment. Consistency with a perceptual model of catchment processes should be assessed before applying one of the models contained in the R packages. Whether it is through a complex representation of shallow groundwater contribution to runoff (leading to a higher number of parameters to estimate), more conceptual calculations of soil moisture or the discretisation of a catchment into different response areas (hence more preprocessing operations), any user will now have more materials related to what the models really imply and how these specificities are made available as outputs by the packages.

5 A critical analysis of package practicalities

To operate the models, the different packages implement a set of functionalities, which can be more or less in line with the hydrological workflow, i.e. from data preparation to result analysis. These functionalities aim at easing and sometimes constraining the use of the model. One would expect to be able to use all the functionalities required to consistently apply a specific model and avoid any supplementary source of errors. One of the specificities of R packages is the provided documentation. The related description and examples must be complete to ensure appropriate application of the models. The user is guided for basic examples and aware of potential errors that can occur. Following the analysis of functionalities and documentation, we present a comparison of R implementation that should foster more rigorous applications of the models. In an effort to contribute to a more extensive documentation relating to the packages and their models, we provide R scripts enabling the use of each package on simple examples. A short analysis on central processing unit (CPU) times is derived from the application of these scripts.

5.1 An uneven set of functionalities and documentation

5.1.1 Package functionalities

What a package provides in terms of functionalities is a distinguishing feature when selecting a specific software or another programming language for hydrological modelling. Among the main features, we usually find the careful preparation of input data to respect the right time references, initialisation period or specific R objects. Enabling an automatic calibration procedure to find a set of parameters consistent with the catchment of study can be an important step for some models as well (though both the `topmodel` and `dynatopmodel` packages have specifically avoided automatic calibration for the reasons discussed in . Functions to visualise and analyse the results are often appreciated by the users. Simple analyses can be the calculation of criteria assessing the overall performance. These criteria are regularly calculated on time series of transformed data to emphasise specific error characteristics. Hydrograph plots are also common for assessing hydrological models. Graphical user interfaces can ease the package usability. As some models enable snow calculations, implementing an independent snow

~~function is necessary to avoid using a snow function on non snowy catchments. Here we present whether the selected packages~~
~~integrate these~~

Table 6 presents whether the different packages integrate several basic functionalities to ~~consistently apply the models(Table 6). Inspections of the packages were conducted based on the different types of documents related to the packages and models. When judged necessary, the codes were analysed to ensure accurate results.~~ apply their models. We provide further explanations hereafter.

Table 6. Functionalities provided by the packages. ✓: the item is offered by the package; ✗: not included; ≈: under development; ∽: suggested in detailed examples or presented in one of the related articles. "RMSE" stands for root mean square error, "MSE" for mean square error, "NSE" for Nash-Sutcliffe efficiency criterion (Nash and Sutcliffe, 1970), "KGE" for Kling-Gupta efficiency criterion (Gupta et al., 2009), "KGE'" for a modified version of the KGE (Kling et al., 2012), "MAE" for mean absolute error criterion, "CP" for coefficient of persistence criterion (Kitanidis and Bras, 1980), "ARPE" for average relative parameter error criterion (Jakeman et al., 1990), "X" for correlation of modelled flow with model residuals (Littlewood, 2002), "NSEseas" for NSE with mean of each month as the reference model instead of the overall mean. "combi." means that the criterion is a weighted combination of several criteria. "custom." means that any available criterion can be customised by the user. This table does not aim to provide guidelines for model evaluation. Other R packages can be used to assess model performance.

Package	Data prepro-cessing fun.	Criteria	Data transfo.	Automatic calibration	Plot fun.	Graphical user interface	Independent snow fun.
airGR	✓	KGE; KGE'; NSE; bias; RMSE; combi.	✓	✓	✓	✓	✓
dynatopmodel	✓	NSE	✗	✗	✓	✗	✗
HBV.IANIGLA	✓	✗	✗	✗	✗	✗	✓
hydromad	✓	NSE; RMSE; MAE; CP; ARPE; X; NSEseas; bias; combi.; custom.	✓	✓	✓	✗	✗
sacsmaR	✗	✗	✗	✗	✗	✗	✓
topmodel	✓	NSE	✗	✗	✗	✗	✗
TUWmodel	✗	∽	✗	∽	✗	✓	✗
WALRUS	✓	NSE; MSE	✓	∽	✓	≈	✓

~~The~~ `sacsmaR` ~~and~~ `TUWmodel` ~~packages do not contain preprocessing functions to use their models with the spatialisation specificities explained in the previous section.~~

Criteria

Weighted combinations of criteria are possible with the `airGR` and `hydromad` packages. These combinations can be derived from the implemented criteria. A combination of several criteria can be, for instance, a weighted sum of three criteria. For example, in airGR, users can average the KGE calculated on discharge, the KGE calculated on square root of the discharge

and the KGE calculated on the inverse of discharge, and different weights can be chosen for each of these three individual criteria. `hydromad` also offers the possibility to implement other combinations through a customisable function. `airGR` and `hydromad` include many different transformations of discharge time series. `hydromad` enables calculation of the NSE on the following transformations: square root; logarithm; Box-Cox (Box et al., 2015); successive differences; monthly aggregation; triangular kernel (Silverman, 1986); time-delay correction (Andrews and Guillaume, 2018). The user can apply other criteria on transformed data when using the customisable criterion. `airGR` enables calculation of the criteria listed in Table 6 on the following transformations: square root; logarithmic (not advised for KGE and KGE', for more details see Santos et al., 2018a); inverse; sorting from lowest to highest; Box-Cox; power. One of the `WALRUS` postprocessing functions returns the NSE of the logarithm of the discharges. Various R packages, such as `hydroGOF` (Zambrano-Bigiarini, 2020), implement model evaluation techniques that are not provided by the selected hydrological modelling packages. Fuzzy measures implemented in the `fuzzyR` package (Chen et al., 2019) can be used to evaluate the outputs of `topmodel` and `dynatopmodel` (for an example of application of fuzzy measures with dynamic TOPMODEL see Freer et al., 2004). This is one way of allowing for the concept of equifinality of model parameter sets in calibration, rather than trying to identify an optimum parameter set (see, for example, Beven, 2006).

Parameter estimation

Automatic calibration in the packages either ~~correspond~~ corresponds to functions permitting the use of calibration algorithms from other packages with the package specific R objects, or to the package own algorithm. Complete examples of automatic calibration with `TUWmodel` and `WALRUS` can be found in the package documentation but do not correspond to one of the functions of these packages. The automatic calibration algorithm included in `airGR` derives from Michel (1991). Calibration algorithms implemented in other R packages can be used within `airGR`, as documented in a vignette. In the `hydromad` package, nine automatic calibration algorithms are proposed, either using built-in R functions (`optim()`), implementations within the package (Shuffled Complex Evolution), or external packages, e.g. the differential evolution algorithm (Storn and Price, 1997) enabled through the use of the `DEoptim` package ~~(?).~~ (Ardia et al., 2020). Parameter estimation and uncertainty quantification are important steps of the hydrological workflow. Multiple sources of epistemic uncertainty are associated with simulations of hydrological models (Beven, 2016), such as uncertainty in the available catchment data that can lead to incorrect model inference (Beven, 2019) or uncertainty arising from the difficulty of models to represent the properties affecting river flows. Several methods can be used to take uncertainty into account when estimating model parameters. For example, `hydromad` includes a function to determine feasible parameter sets and estimate prediction quantiles by applying the GLUE method. The `FME` package (Soetaert and Petzoldt, 2010) enables estimation of parameters within a Bayesian Markov Chain Monte Carlo (MCMC) framework.

Plot functions

The plot function of `airGR` can display several variables (e.g. Fig. A1): time series of precipitation, potential evapotranspiration, actual evapotranspiration, temperature, snow water equivalent, simulated discharge, observed discharge and simulated

minus observed discharge (residuals); interannual monthly median; correlation between observed and simulated discharge;
cumulative frequency. `dynatopmodel` contains a function to plot the observed and simulated hydrographs along with the
precipitation and actual evapotranspiration time series (mainly designed for short time series, e.g. Fig. A2). `hydromad` inte-
grates a function to plot simulated and observed hydrographs (including different simulation configurations and rainfall time
series, e.g. Fig. A3 and A4). This function can also plot the flow error, i.e. the criterion value for each point of the time series.
A function to select and plot discrete events is also included in `hydromad`. `WALRUS` includes two functions to display the
model outputs. The first one enables to plot time series of observed discharge, simulated discharge, precipitation, potential
evapotranspiration, modelled groundwater drainage, modelled actual evapotranspiration, wetness index, soil reservoir and sur-
face reservoir levels, seepage and surface water supply or extraction (e.g. Fig. A5). The second function displays the model
residuals, the autocorrelation of residuals and the cross correlation between residuals and the precipitation time series.

Graphical user interfaces

`airGR` and `TUWmodel` offer a graphical interface for manipulating the models. `WALRUS` GUI is currently under development
(see Sect. 4.6 of Brauer et al., 2017). The `airGR` and `TUWmodel` GUIs rely on the `shiny` package (Chang et al., 2019)
that implements an R framework to build a web application. The GUI developed for the `airGR` package is proposed in the
`airGRteaching` package (Delaigue et al., 2018, 2020a). This GUI is either available online (https://sunshine.irstea.fr/) or
by launching the interface from R. This GUI integrates several features (see Fig. B1), e.g. easily estimating the parameters,
either manually or automatically. To perform an automatic calibration, users have to select an objective function among the
NSE and KGE, calculated on flow time series (TS), flow inverse TS or square root of flow TS. They can then directly visualise
the impacts on seven criteria values (including those mentioned previously and the flow bias) and through several graphics.
For both types of calibration, it is possible to adjust the temporal window on which the model performs (using a slider or by
selecting the period on the graphic). Four graphical panels are available: precipitation TS, simulated and observed hydrographs
and flow error TS (e.g. Fig. B2a); a concise performance graphic displaying TS of precipitation, simulated and observed
hydrographs, interannual monthly median, correlation between observed and simulated discharge and cumulative frequency
(e.g. Fig. B2b); TS of store levels and runoff components (e.g. Fig. B2c); a model diagram displaying store levels, fluxes and
unit hydrographs at each time of a selected temporal window (e.g. Fig. B2d); a fact sheet with several hydrometeorological
characteristics of the selected catchment (e.g. Fig. B2e and B2f). The simulation results and plots can be downloaded by users.

The `TUWmodel` GUI (Sleziak, 2019) is available online (https://webaapptuwmodel.shinyapps.io/TUWteaching/). This in-
terface includes five example datasets on which the TUWmodel parameters can be manually adjusted (see Fig. B3). Users can
adjust the parameters and directly observe the impacts on a graphical panel that displays the simulated hydrograph compared
to the observed hydrograph and TS of three state variables (e.g. Fig. B4a). This interface also offers a second panel that allows
to visualise the localisation of the five catchment outlets on a map along with their area, mean elevation, mean slope, forest
cover percentage, mean annual precipitation, mean annual air temperature and mean annual runoff (e.g. Fig. B4b).

5.1.2 Package documentation and support

~~To handle the complexity associated with the different hydrological models as well as with the functionalities provided by the packages , the documentation is obviously essential for any user. It is therefore important to assess whether looking at the overall documentation is sufficient to easily make use of package basics. In this regard, we propose a comparison based on our use of the available explanatory documents(Table 7). This analysis is by definition subjective as it relies on our experience as users. However, we think that it can still give insights on the meaningful content of the documentation . Analysing the documentation explanation by explanation would indeed be very complicated to present. There are two different types of documentation related to these packages: the R documentation that includes user manuals (functions explanations, mandatory) and sometimes vignettes ("long-form guides that illustrate how to use packages", Slater et al., 2019); the external documentation that comprises scientific journal articles and sometimes websites. For each function of a package, the formal R user manual includes mandatory fields (e.g. name, value, title, description and arguments) and optional fields (e.g. details, examples and references) . We consider two types of scientific articles in this analysis: articles that were written in order to present the packages; and articles using the packages and made by one of the package authors. Websites usually contain elements such as video tutorials, list of publications mentioning the package, examples, user groups. Vignettes and external documentation are not required when creating a package but can be very useful for thorough understanding of the packagesand models (Table 8)~~Table 7 presents whether the main functionalities of the packages are provided with sufficient explanatory documents. Table 8 summarizes the available additional documentation that can help to better understand the characteristics of the models and the packages.

Table 7. Assessment of package documentation. Description: information on the general purpose of the function and the associated possibilities. Details: precise explanations of the function. Arguments: description of the function arguments that requires details about the unit, the R object class and how to obtain it. Value: description of the function outputs. References: related documentation where users can find more information on the function. Examples: R commands to use the function. Examples are considered as "comprehensive" if they cover most of the functionalities and if there is an example for each function. Dataset: an example dataset is provided and can be used with the package functions. Steps between functions: explanations of the required stages to run the main functions. The last two fields are not explicit parts of the package manuals but can help to understand and use a package.

Package	Description	Details	Arguments	Value	References	Examples	Dataset	Steps between fun.
airGR	✓	✓	✓	✓	✓	✓	✓	✓
dynatopmodel	✓	✓	✓	✓	✓	✓	✓	✗
HBV.IANIGLA	✗	✗	✓	✓	✓	✗	✗	✗
hydromad	✓	✓	✓	✓	✓	✓	✓	✓
sacsmaR	✗	✗	✗	✗	✗	✗	✓	✓
topmodel	✓	✓	✗	✓	✓	⌣	✓	✗
TUWmodel	✓	✓	✓	✓	✓	✓	✓	−
WALRUS	✓	✓	✓	✓	✓	✓	✓	✓

The `topmodel`'s main example does not inform on the preprocessing steps to discretise a catchment. It may be complicated to use `sacsmaR` to run the semi-distributed version of Sacramento because the example found in one of the vignettes does not illustrate the required discretisation. Coherence between `dynatopmodel` functions is not made explicit by the provided function, especially for the catchment spatial discretisation. All the packages except `HBV.IANIGLA` include datasets that can be used with the examples provided in the documentation. `HBV.IANIGLA` provides an example on how to generate random time series of inputs.

Table 8. Additional available package documentation and support. @: e-mail. "GL" stands for GitLab and "GH" for GitHub.

Package	Website	Vignette(s)	Article(s) about the package	User group	Bug report
airGR	https://hydrogr.github.io/airGR/	✓	Coron et al. (2017)	✗	@ and GL
dynatopmodel	✗	✗	Metcalfe et al. (2015)	✗	@
HBV.IANIGLA	✗	✗	✗	✗	@
hydromad	http://hydromad.catchment.org/	✓	Andrews et al. (2011)	Google	@ and GH
sacsmaR	✗	✓	✗	✗	@ and GH
topmodel	https://github.com/ICHydro/topmodel	✗	✗	✗	@
TUWmodel	✗	✗	⌣	✗	@
WALRUS	✗	✗	⌣	✗	@ and GH

`airGR` includes several vignettes, for example on how to estimate parameters within a Bayesian Markov Chain Monte Carlo (MCMC) framework. The `WALRUS` package is stored on the GitHub platform where a complete set of documents, tutorials and data can be found (e.g. an R script to run a Monte Carlo parameter estimation procedure). A comprehensive user manual, whose structure is different from the usual R documentation, can also be found on GitHub. `sacsmaR` is stored on GitHub with a vignette on how to use the different functions. `hydromad` offers a vignette and nine demos are available and deal with subjects such as how to estimate the model parameters or how to conduct a sensitivity analysis. Examples of sensitivity analysis and generalised likelihood uncertainty estimation (GLUE) method (Beven and Binley, 1992) are available on the `topmodel`'s website. Articles related to the `TUWmodel` and the `WALRUS` packages were not written to present the packages themselves but the models included in these packages. Other examples of `TUWmodel` were found in the appendixes of Ceola et al. (2015). Users are invited to report bugs or ask for additional support via email and, for some packages, by creating GitHub (`hydromad`, `sacsmaR`, `WALRUS`) or GitLab (`airGR`) issues. A user group related to the `hydromad` package was created for additional support.

Looking at Table 6, it appears that there is an important heterogeneity in the availability of package functionalities. Packages such as `airGR`, `hydromad` and `WALRUS` integrate many functionalities from data management to result analyses whereas the other packages mostly contain the main functions to run the associated model. We can differentiate two types of packages in our study: packages guiding the user with functionalities in line with the hydrological workflow and, to a certain extent,

constraining the use to reduce potential errors and packages allowing more flexibility but less guidance and thus potentially more errors. Regarding the documentation, packages with the most important number of functionalities also provide more comprehensive documentation and additional documents. Even if `dynatopmodel` and `TUWmodel` offer more explanatory documents and examples than `sacsmaR` and `topmodel`, there is still a lack of information concerning the spatial distribution these four packages permit (see Sect. 4.2). Any user has to be able to use every functionality of a package and the models they contain. While the more complex the models are, the more important the functionalities and therefore the provided explanatory documentation become, strengthening the documentation associated with some of the presented packages seems to be necessary to assure more rigorous applications of the models.

5.2 A guide for user implementation

Considering the previous analysis along with package distinctiveness in terms of R implementation raises the question of how to use the packages containing hydrological models. Indeed, one would expect consistent inter-comparison studies between the different models, leading to more reliable analyses. ~~Package practicalities can also be assessed through an analysis of~~ Figure 4 shows the links between the main functions ~~of a package. Such an examination could be useful to provide guidance regarding package application. Here, we try to put ourselves in the shoes of a user who has to apply the models of the different packages and therefore needs to understand which function they have to use, where in the script and how. In this regard, we propose a unified diagram of the connections between the main functions that we have been able to run (Fig. 4). This figure is intended for users familiar with R packages and aims at guiding users in their application of the hydrological modelling R packages. We therefore provide R scripts enabling the application of~~ required to apply each package on a simple hydrology example (~~supplementary documentation). The provided R scripts show the basic R commands required to test one parameter set on a French midland catchment (The Meuse River at Saint-Mihiel) and, for the packages integrating a snow-accounting routine, on a French mountainous catchment (The Ubaye River at Lauzet-Ubaye). Parameter estimation and uncertainty quantification are important steps of the hydrological workflow. Multiple sources of epistemic uncertainty are associated with simulations of hydrological models (Beven, 2016), such as uncertainty in the available catchment data that can lead to incorrect model hypotheses (Beven, 2019) or uncertainty arising from the difficulty of models to represent the properties affecting river flows. Several methods can be used to take uncertainty into account when estimating model parameters. For example, `hydromad` includes a function to determine feasible parameter sets and estimate prediction quantiles by applying the GLUE method. The `FME` package (Soetaert and Petzoldt, 2010) enables estimation of parameters within a Bayesian Markov Chain Monte Carlo (MCMC)framework~~R scripts are provided in the supplementary documentation).

The diagrams of Fig. 4 indicate three main groups of packages in terms of R implementation:

1. `airGR`, `hydromad` and `WALRUS` integrate many R functions with complex connections and inner consistency. These choices regarding the R implementation follow similar steps from data formatting to result analysis. They aim at reducing potential errors arising from R codes written by the modeller, whose choices are guided throughout the hydrological workflow. It also eases the application of a new model by end-users.

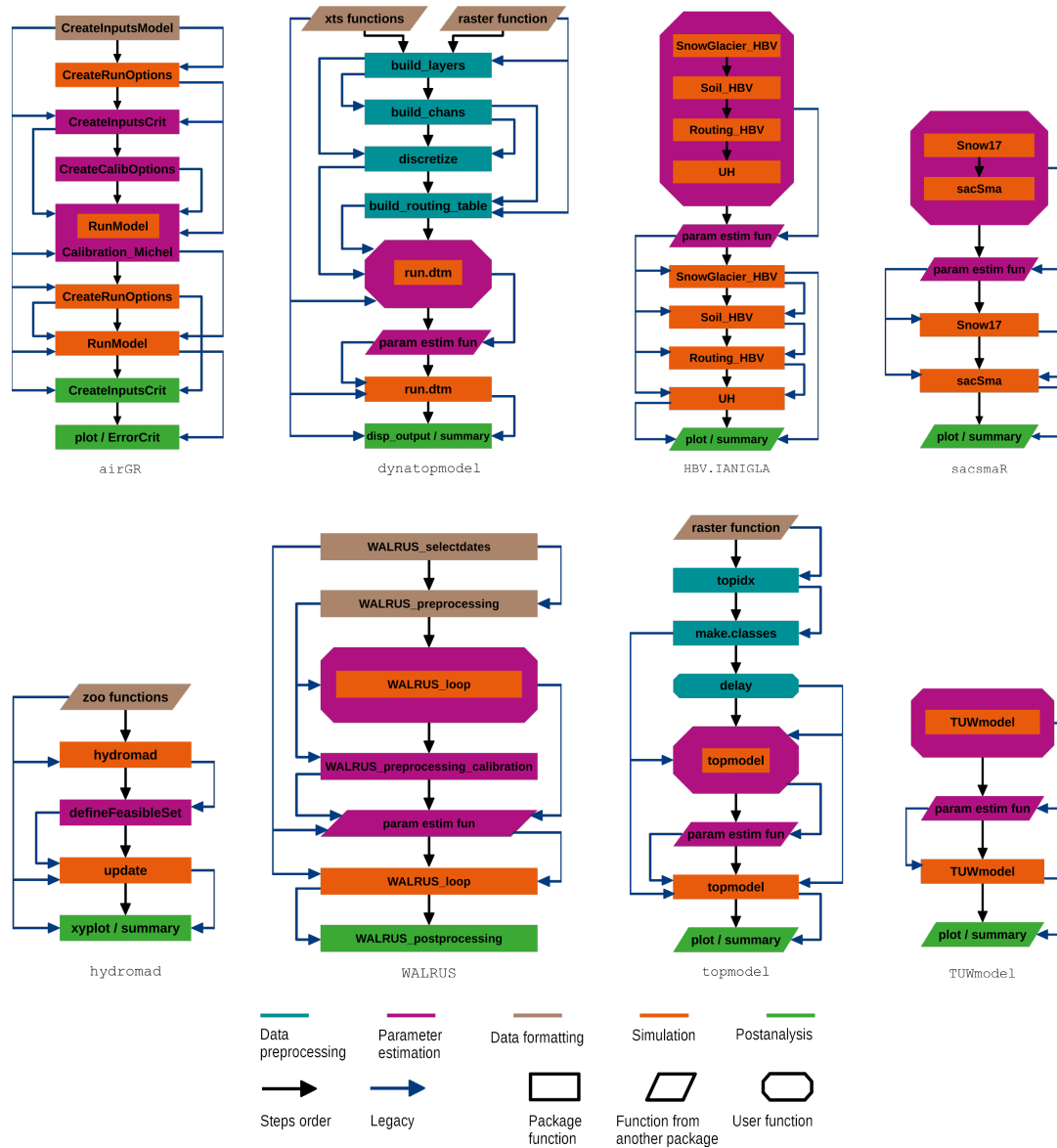



Figure 4. Unified diagrams illustrating the package main functions that are necessary to proceed with parameter estimation and validation on a basic example in hydrology modelling. R core functions are not explicitly illustrated. Basic data preprocessing steps, such as checking data gaps and consistency, are not displayed on this diagram but it is strongly recommended to adopt such practices before operating a model. Users can run `hydromad.options` and `hydromad.stats` to change or visualise the options for parameter estimation in `hydromad`. Many different parameter estimation methods are available within the R environment or within the selected packages. "Legacy" means that a function depends on other functions to be operable (e.g. its arguments are obtained by running another function). "user function" means that users have to write their own R function integrating, among others, the legacies illustrated on the diagrams.

2. `sacsmaR`, `TUWmodel` and `HBV.IANIGLA`, on the other hand, whose functionalities are less extensive, rely on a very simple R structure. R objects preparation are left to the user (i.e. without particular specifications or requirements) who has to provide vectors of data time series. The `sacsmaR` and `TUWmodel` packages do not contain preprocessing functions to use their models with the spatialisation specificities explained Sect. 4.2. As the packages only integrate functions allowing to run the core models, the R structure only depends on the models inputs and outputs, whether it is with one function integrating all the modules of the model (`TUWmodel`) or with separate components (`HBV.IANIGLA` and `sacsmaR`). Unlike what is offered by the packages of the first group and `dynatopmodel`, the user has to code the plots they desire by using the runoff outputs.

3. As pointed out in Sect. 4.2, operating the two versions of TOPMODEL requires ~~spatialisation~~ spatial discretisation preprocessing steps. `dynatopmodel` and `topmodel` integrate these steps at the beginning of the workflow, through more (`dynatopmodel`) or less (`topmodel`) complex functions and connections. The spatial operations also require the use of an external package to deal with raster data. The remaining steps of the workflow are then very similar to those of the second group of packages.

`airGR` and `hydromad` contain parameter estimation functions and objective functions that are integral parts of their structure, whereas users have to build their own objective function and combine it with an external algorithm when operating the other packages. `WALRUS` includes the structural functions to facilitate parameter estimation. `TUWmodel` and `WALRUS` provide examples on how to proceed with these steps.

~~Unified diagrams illustrating the package main functions that are necessary to proceed with parameter estimation and validation on a basic example in hydrology modelling. R core functions are not explicitly illustrated. Basic data preprocessing steps, such as checking data gaps and consistency, are not displayed on this diagram but it is strongly recommended to adopt such practices before operating a model. Users can run `hydromad.options` and `hydromad.stats` to change or visualise the options for parameter estimation in `hydromad`. Many different parameter estimation methods are available within the R environment or within the selected packages. "Dependency" means that a function depends on other functions to be operable (e.g. its arguments are obtained by running another function). "user function" means that the user has to write their own R function integrating, among others, the dependencies illustrated on the diagrams.~~

The path to better guidance to avoid mistakes in application of the models — which implies more rigorous methods that include, for instance, uncertainty analyses — is a tricky road facing the wide heterogeneity of packages in terms of R structure and models in terms of modelling choices. One could imagine better harmonisation of packages taking advantage of other packages' strengths (e.g. using a similar structure for the objective function or managing time complexities with the same R functions). This goal is not out of reach and would considerably improve the usability and scope of these packages. It would require defining sampling strategies for parameter estimation (e.g. with the differential evolution adaptive metropolis (DREAM) algorithm, Vrugt and Beven, 2018) and post-processing techniques for performance and uncertainty assessments. Some hydrological modelling R packages include these strategies, some leave it external (see Sect. 5.1.1). This first attempt

at categorising the packages in terms of R implementation along with the provided R scripts should ease the application of packages containing hydrological models.

5.3 Analysis of R structures and CPU times

5.3.1 Programming languages and dependencies

835

Some packages are entirely coded in R, which is an interpreted language, and some integrate models coded with a compiled programming language interfaced with R: ~~FORTRAN~~ Fortran (Backus et al., 1957), C (Kernighan and Ritchie, 1978) or C++ (Stroustrup, 1984, see Table 9). When the model functions are coded in a compiled language interfaced with R, users may not have direct access to all the code if required. Computation times tend to be lower when using a compiled programming

840 language rather than using an interpreted one (see Sect. 5.3.2). As the most part of necessary Central Processing Unit (CPU) time is dedicated to the actual hydrological model run, some package developers chose compiled languages for the model calculations. ~~The different programming languages interfaced with R were identified by extracting the package sources, because they could not necessarily be identified by simply displaying the code from the R console.~~ Three packages integrate models entirely coded with the R language and four packages with a compiled programming language (C, C++ or ~~FORTRAN~~ Fortran).

845 The `hydromad` package integrates some models coded in R, some in C or C++, and some with both implementations to facilitate both debugging and fast execution.

Table 9. Programming languages of the models used in the selected R packages.

| Package | Interpreted | Compiled | | |
	R	C	C++	~~FORTRAN~~ Fortran
airGR				✓
dynatopmodel	✓			
HBV.IANIGLA			✓	
hydromad	✓	✓	✓	
sacsmaR	✓			
topmodel		✓		
TUWmodel				✓
WALRUS	✓			

Table 10 summarises whether packages require other packages to run the models. ~~A package is considered as being dependant if one of its functions cannot be run without downloading another package. A package is not considered as depending on any other package when the use of an external package is only suggested in an example or in one of the related articles. R Core~~

850 ~~Team packages, such as `stats`, are not taken into account in this assessment, as they are packages installed by default.~~ Three of the selected packages integrate functions requiring the use of external packages. `dynatopmodel` has the highest number of dependencies (nine packages), `hydromad` has six dependencies and `WALRUS` has one dependency. These three packages

rely on several types of external packages. Some of these external packages (e.g. `zoo`, Zeileis and Grothendieck, 2005) integrate functions to manage time series. `dynatopmodel` relies on packages integrating functions to deal with spatial data (e.g. `raster`, Hijmans, 2020). Some packages are employed to solve equations or proceed with specific calculations (e.g. calculations on univariate polynomials with `polynom`, Venables et al., 2019).

Table 10. Package dependencies. R Core Team packages are not explicitly taken into account in this table.

Package	Dependencies
`airGR`	✘
`dynatopmodel`	`deSolve`; `lubridate`; `raster`; `rgdal`; `rgeos`; `sp`; `topmodel`; `xts`; `zoo`
`HBV.IANIGLA`	✘
`hydromad`	`car`; `Hmisc`; `latticeExtra`; `polynom`; `reshape`; `zoo`
`sacsmaR`	✘
`topmodel`	✘
`TUWmodel`	✘
`WALRUS`	`zoo`

5.3.2 CPU times

~~From a user perspective, computation times can be meaningful to determine whether a package is suitable for a specific study. Short computation times are usually very well appreciated, especially when dealing with finer time steps or more complex spatial discretisations. Applying a model on a large database, generating an ensemble in operational (flood) forecasting or performing Monte Carlo runs for uncertainty analyses can also significantly increase computation times, hence some of the packages include some compiled code to speed up the production runs of the model.~~ We hereby present the CPU time required for one model run ~~, which was estimated from 1000 runs~~ with the selected packages (Fig. 5)~~with the `microbenchmark` package (Mersmann, 2019). The models were run at a daily time step on a midland catchment where high flows mostly result from precipitation events in winter, the Meuse River at Saint-Mihiel (2543 km^2, from January 1st 1990 to December 31th 1999) and, for the packages integrating a snow function, on a mountainous catchment where high flows mostly result from snowmelt in spring, the Ubaye River at Lauzet-Ubaye (943 km^2, from January 1st 1989 to December 31th 1998). The time series of precipitation and temperature at a daily time step were extracted by Delaigue et al. (2020b) from the SAFRAN countrywide climate reanalysis of Météo-France (Vidal et al., 2010). The PET time series were calculated using the Oudin et al. (2005) formula. The streamflow data were retrieved from the "Banque Hydro" database (Leleu et al., 2014). For TOPMODEL and dynamic TOPMODEL, a DEM with a resolution of 25 m by 25 m was derived from the BD ALTI DEM (IGN, 2013).~~ As the `hydromad` package includes two implementations of the GR4J and IHACRES-CMD models, one coded in R and one coded in C (see Sect. 5.3.1), computation times were estimated for both implementations. ~~Only one parameter set is tested for each model.~~ Note that `WALRUS` computation times can vary depending on the selected parameter set.

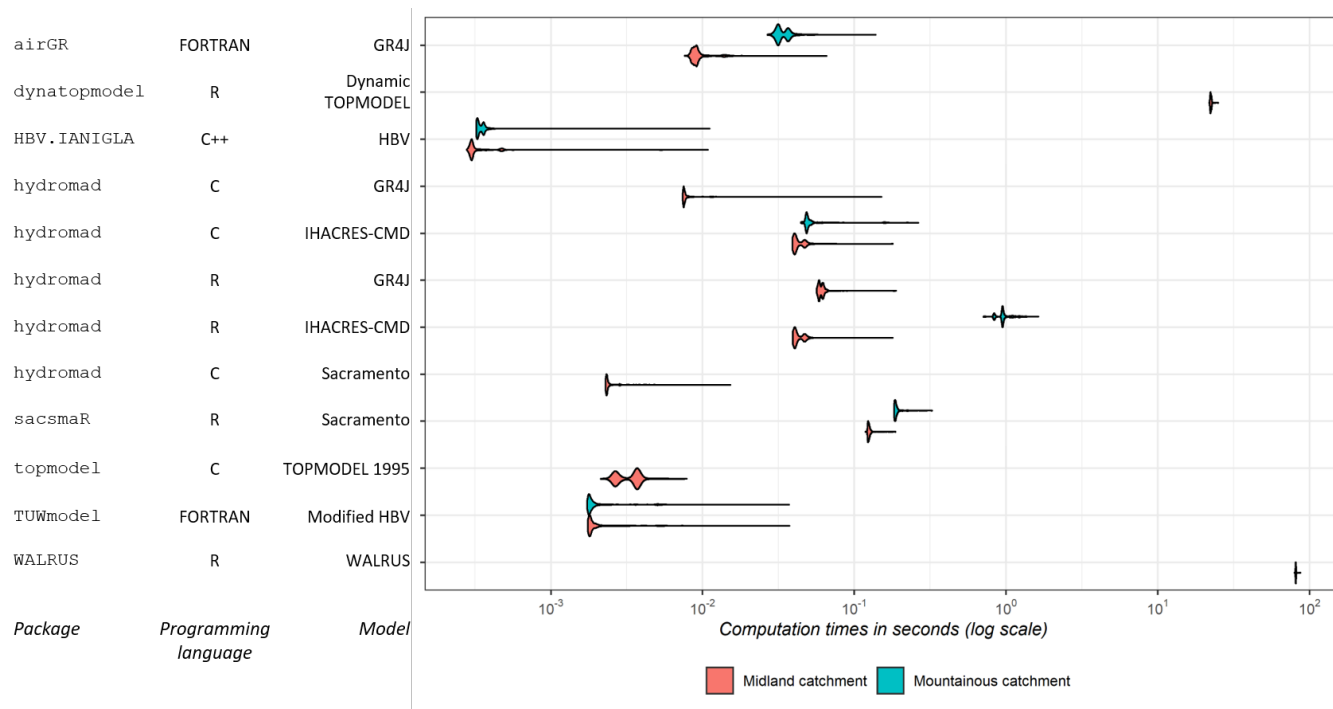



Figure 5. Computation times of the packages (time of one run estimated from 1000 runs using the same parameter set with the `microbenchmark` package). The models were run on a ~~midland~~ French catchment, "The Meuse River at Saint-Mihiel" and a mountainous catchment, "The Ubaye River at Lauzet-Ubaye", for a 10-year period at a daily time step. The related package functions were applied using the default settings, i.e. on a single spatial unit except for `dynatopmodel` and `topmodel`. Runs on the mountainous catchment were performed only with the packages that integrate a snow function and using the default settings (i.e. five elevation bands for `airGR` and one for the other packages). `hydromad` includes an R version and a C version of GR4J and IHACRES-CMD. Data preprocessing and formatting are not taken into account in the estimation of computation times. Computer characteristics: RAM capacity: 8.00 ~~Go~~GB; CPU: Intel i5-8250U 1.80 GHz; OS: Windows 10 (64-bit). R version: 3.6.0 (64-bit).

875   The results of Fig. 5 show that the packages based on models coded with a compiled programming language have lower CPU times than the packages integrating models coded in R. The CPU times associated with the packages integrating models coded with a compiled language have the same order of magnitude. The `HBV.IANIGLA` package has lower CPU times than the other selected packages. The CPU times associated with `sacsmaR` and the models coded in R of `hydromad` are lower than the CPU times of `dynatopmodel` and `WALRUS` (CPU times can also depend on the model implementation). `WALRUS` has

880   the highest CPU times, which is probably caused by the flexible time step approach: the computational time step is decreased automatically to improve numerical stability. For all these packages, the runs performed with the associated snow function resulted in higher CPU times than without the snow function, except for `TUWmodel` that does not integrate an independent snow function. Note that computation times may differ when selecting other function settings and model configurations.

6   Discussion

885   Following these analyses of hydrology modelling R packages, we discuss how users can actually apply one of the models offered by the selected packages to a specific application or research question, we identify what improvements could be brought to the packages and we discuss the reasons and the implications of the implementation choices that were made on the packages.

6.1   Usefulness of the proposed analysis for end-users

890   While our analysis focuses on models that can be considered as conceptual rainfall-runoff models, we have highlighted different approaches, assumptions, and choices underlying these models, both in terms of structure and spatial discretisation, which imply distinct numbers of parameters and sometimes specific inputs.

6.1.1   Choosing the fit-for-purpose model and package

This attempt to simplify the users' selection process does not aim at labelling the "good" and "bad" models or packages and
895   therefore cannot shed light on which models hydrologists should use today. In our opinion, two major points should be drawn from this study in terms of hydrological modelling. First, this work should be considered as a tool to determine which models, within the R environment, best fit the specific requirements of the end-user and their perceptions about the dominant processes in a catchment (see the stages in the modelling process in Beven, 2012). For instance, we do not advise to run the WALRUS model on steep mountainous catchments, rather on lowland catchments with shallow groundwater. WALRUS was also de-
900   signed for catchments where ditches and channels have great influence on river flows and can take into account major water withdrawals or surface water supply. Interestingly, only WALRUS integrates modelling aspects related to catchments affected by human activities and none of the selected packages offer functions to take the regulation of dams into account. Simulations on catchments with a high proportion of solid precipitation should be performed with models integrating functions to take snow into account (e.g. the HBV models of `HBV.IANIGLA` and `TUWmodel`, the Sacramento model of `sacsmaR` and the
905   CemaNeige model of `airGR`). The dynamic TOPMODEL can be relevant to simulate river flows in catchments with a high spatial heterogeneity of precipitation (e.g. with frequent storms) since it integrates a finer spatial discretisation. This discretisation requires at least a digital elevation model. These examples certainly do not cover the numerous applications enabled by these models but can help users to reason when selecting a hydrological model implemented in the R environment. Second, our analyses address what is enabled by the packages when considering these modelling specificities. For instance, users can
910   theoretically run the complex semi-distributed version of `sacsmaR` but would need to prepare the spatial discretisation with external R packages or softwares. While `HBV.IANIGLA` implements the HBV model with separated functions, `TUWmodel` includes one function for the whole model — although calculations on separate zones are implemented, which can improve snow calculations. `dynatopmodel` enables a different set of parameters and inputs for each computational unit (8 parameters), whereas `topmodel` integrates a version based on 10 adjustable parameters (including 2 initialisation parameters) but
915   the model can be run with a less complex spatial version requiring fewer preprocessing procedures.

6.1.2 Towards better helping users for appropriately using new models

Appropriately using a new model is fundamentally difficult. Understanding the complexity resulting from the different modelling choices, the model inner consistency, and whether it is appropriate for a specific research problem also depends on the perceptual model of the hydrology of a catchment (e.g. Wrede et al., 2015; Beven and Chappell, 2020). Adding the difficulty of implementation in terms of software or programming language complicates this task even more. Flexible modelling frameworks, such as `SuperflexPy` (Dal Molin et al., 2020) and FUSE (Clark et al., 2008), have partially addressed these difficulties. However, their complex conceptualisation makes them more accessible to advanced modellers than to newcomers. The FUSE implementation for R available on GitHub is in need of active maintenance by the community (https://github.com/cvitolo/fuse). The `hydromad` R package provides such a flexible framework to a certain point, as it enables combining different soil moisture accounting functions with several routing modules. It is one of the reasons that should encourage users to model within the same framework. Such a framework can be the R environment with the available modelling packages. But for these packages to be powerful tools and still maintain a certain degree of practicality and accessibility, they need to be strongly documented and user-friendly. In this regard, implementing modelling functionalities can guide users towards good practices and help them to interpret results. However, enough flexibility should be maintained to allow other applications not considered by authors. Package developers must find the right balance when adding functionalities, in an attempt to help users run models with more possibilities, without making the package implementation excessively complex. It means that packages are made accessible for newcomers but are still of interest for more experienced hydrologists. In this respect, GUIs are interesting features to make packages more accessible to newcomers and to users that are not familiar with coding, especially for students. They allow users to understand what the models imply in terms of fluxes, parameters and stores by providing easy tools to perform basic simulations. Given that these GUIs are less flexible than the packages themselves and do not enable all the modelling aspects, they do not intend to replace the packages. They can be considered as convenient tools to help users in their understanding of the models and therefore facilitate the selection process. These tools are improvements to guide users in their applications of the different models and could be considered for future developments of the selected packages, as we have seen that only two packages propose a GUI (`airGR` and `TUWmodel`). Learning the idea of calling functions is however important in a world requiring reusable methods between operators, as it is a more transferable way of thinking. GUIs should therefore guide users in doing so.

6.2 Possible improvements for the packages

6.2.1 Fifty shades of R package functionalities

With respect to our list of packages, we have seen that three main groups of packages emerged from our analyses of practicalities. `airGR`, `hydromad` and `WALRUS` offer the most functionalities to run their associated models and therefore rely on more functions integrated in a complex structure and associated with an extensive documentation. Providing functions such as graphical outputs or performance criteria is a major advantage for users that can easily and quickly have access to an overview of how the models performed on a simulation. This does not restrain users in their modelling assessments, as graphics and criteria can

be coded with the R environment when accessing the model outputs made available by the packages (see Table ~~??~~5). Do these packages yet allow enough flexibility for other applications and users? On the one hand, we have been able to combine these three packages with the same automatic calibration algorithm, namely the `DEoptim` package ~~(?)~~(Ardia et al., 2020), which is an indicator of adaptability. On the other hand, there are still improvements to be made. For instance, the performance criteria included in `airGR` require specific `airGR` objects to be calculated. Users cannot directly use time series of other variables in the related functions that would, for instance, enable to compare simulations from two different packages. `hydromad` enables the estimation of adjustable parameters with different algorithms via several external R packages but documentation does not yet integrate examples on how to use its structure with other algorithms, i.e. new R algorithms that are not included in the structure of `hydromad`.

The second group of packages, offering fewer functionalities with a simpler R structure, includes `HBV.IANIGLA`, `sacsmaR` and `TUWmodel` though `TUWmodel` provides a more extensive documentation. This type of implementation, as flexible and easy to use as it seems (i.e. fewer steps, dependencies and functions on the diagrams of Fig. 4), does not always imply that it is easy to use and apply to specific uses. When guidelines are not sufficiently provided, it can lead to more errors and misuse of the models. For example, the documentation of `HBV.IANIGLA` does not inform on how to manage missing values, which usually occur in flow time series. `sacsmaR` does not provide functions for spatial discretisation or even examples on how to proceed by operating other R packages. This exposes users to misuse and errors. Even if the package is theoretically adaptable, using the Sacramento model with a finer spatial discretisation is very complicated with this package. While `TUWmodel` does not offer many functionalities, its documentation provides guidelines for basic uses.

`dynatopmodel` and `topmodel` (third group) include the basic ~~spatialisation~~ spatial discretisation functions to run their version of TOPMODEL. These functions guide users for simple applications of the model. However, few guidelines are given for more complex applications that are enabled by the packages, for example, information on how to create hydrological response units with maps of land use or soil type when using `dynatopmodel`. As `dynatopmodel` integrates a more complex spatial distribution, its structure is more complex and ~~less adaptable to other applications. Nevertheless, more~~ therefore harder to implement. More functionalities could have improved its usability, although it is more adaptable to other applications in its present form.

6.2.2 On the importance of accounting for uncertainties

`airGR` and `hydromad` include parameter estimation functions as an integral part of their structure. `TUWmodel` and `WALRUS` provide some examples on how to combine external packages to do so. However, the four other packages do not present guidelines regarding this step of the hydrological workflow. Even if it is sometimes important to understand how to manually estimate the parameters of a model and even more relevant for some models to not obtain unrealistic parameter sets, parameter estimation is an important step for most of these conceptual hydrological models. As imperfect as some of these procedures might be, as highlighted by Beven (2012, 2016), for instance regarding the important inclusion of epistemic uncertainty (Beven, 2019), the related documentation lacks in some of the selected packages (note that `topmodel` includes an example to test different sets of parameters).

Taking different sources of uncertainty into account, especially epistemic uncertainty in data inputs (Beven, 2019), is important when working with models considered as hypotheses about how a catchment is functioning (e.g. Andréassian et al., 2009; Clark et al., 2011; Beven, 2016; Blöschl, 2017). It enables, for a specific case study, to reject a model that would yield poor predictions due to wrong hypotheses not brought to light by highly uncertain evaluation data (Beven, 2016, 2018). This important step of the hydrological workflow is mentioned in the documentation of four of the selected packages (`airGR`, `hydromad`, `topmodel` and `WALRUS`). Uncertainty analyses could therefore be more strongly documented in future versions of the packages to guide users towards more rigorous applications of the models. Although the proper way of including these uncertainties is debated among hydrologists, for example, regarding the non-stationarity of epistemic uncertainty (e.g. Koutsoyiannis and Montanari, 2015; Beven, 2016) or the definition of limits of acceptability for model rejection (e.g. Clark et al., 2011; Beven, 2018), this highly debated question is beyond the scope of this paper. Nevertheless, hydrologists that intend to apply one of the packages presented in this study should be encouraged to perform uncertainty analyses. In this regard, several methods can be used within the R environment such as GLUE or MCMC analyses.

6.3 Aspects of practical implementation and maintenance of packages

The adaptability and efficiency of packages can also be assessed in terms of R structure. Among the choices that developers have made for their R packages, programming languages used for the core functions of the models and dependence on external packages for some functions resulted in very different R structures. Using a compiled programming language for the core model functions (`airGR`, `HBV.IANIGLA`, `hydromad`, `topmodel` and `TUWmodel`) tends to lower CPU times and ease parameter estimation procedures, though computation times can also depend on algorithm efficiency, choice of parameter estimation technique, number of parameters to be estimated and operations not related to the core model runs. As a result, applications on studies relying on a large database of catchments and uncertainty analysis procedures such as Monte Carlo realisations are easier with these packages. On the other hand, this does not facilitate the comprehensibility of the packages, as users who would want to understand the core function algorithms would need to learn another programming language. Coding all the functions in R (`dynatopmodel`, `sacsmaR` and `WALRUS`) improves the comprehensibility of the core functions of a model. This also allows users to change the model algorithms directly within the R environment. However, depending on the implementation, this alternative can increase computation times and therefore restrain users in their application of the hydrological models. Relying on external packages is a way of taking advantage of the many possibilities offered by the packages of the R environment, which can improve the functionalities of a package. However, package dependencies can become a liability when the versions of the external packages change (an issue that is not specific to the R language). This sometimes induces errors with functions that can no longer be run by users, which could thus reduce the durability of R scripts and therefore the transferability and understanding of methods and results.

Slater et al. (2019) identified key points for future developments of R in hydrology that would improve the transferability of methods. Structured approach when developing a package, user-friendly functions and documentation are in line with the specific results of our analyses of practicalities. Developers are advised to consider these analyses, as R packages implementing hydrological models are great improvements avoiding to recode a model or buy a specific software. Even if users need to learn

the basics of the R language to employ one of these packages, this allows many modellers to save time and focus on modelling rather than algorithm issues. Furthermore, despite the need for some improvements, using a package for hydrological modelling within the R environment is greatly enhanced by package follow-up maintained by a strong community of developers and users. Bugs and requests can be reported via e-mail for all the packages (see Table 8). Ensuring follow-up to help users understand package specificities, deal with errors and misuse is a way for more reusable methods and more rigorous applications of the models. It also allows users to follow the latest developments through version controlling, thus ensuring R script durability (version controlling of `airGR` is accessible via the GitLab platform; `WALRUS`, `hydromad` and `sacsmaR` version controlling are accessible via GitHub; older versions of the four other packages are archived on the CRAN). This two-way interaction between users receiving help with their specific applications and developers that can improve their packages is essential to keep the R environment a helpful framework for hydrologists. In this sense, several developers within the hydrological science community have been committed to allow the open source use of hydrological models in the form of R packages.

7   Conclusions

Given that the R language can easily be operated for hydrological purposes, the growth of available modelling packages makes the choice of an appropriate package more complicated. In our will to encourage any hydrologist to employ one of the packages, we have first proceeded with a careful analysis of a selection of models contained in eight packages. These models were examined in terms of conceptual storages and fluxes, spatial discretisation, requirements and retrievable outputs. We have then evaluated the packages regarding their practicality, i.e. the integrated technical features, the related documentation, the R hydrological workflow, CPU times and programming languages interfaced with R. The results of our unified analyses confirmed that the selected models rely on different assumptions with regards to the conceptualisation of the water cycle and therefore their emphasis on the main physical processes contributing to river flows. A model structure can be selected depending on our knowledge of the physical properties of a catchment. As our understanding of these properties is limited, hydrological models are subject to epistemic uncertainties (e.g. Beven, 2016), making them imperfect but improvable tools for different purposes in hydrology. The spatial discretisation options that are theoretically enabled by the packages range from lumped conceptualisation to a finer spatial discretisation integrating both HRUs and sub-basins. These modelling specificities result in variations concerning the requirements of each model. Packages enable more or less flexibility with respect to these modelling characteristics, in terms of time steps and retrievable outputs. While it was expected to find differences in model conceptualisations and technical features offered by the packages, we did not expect such a heterogeneity regarding what the packages enable. Indeed, some packages provide functions from data preparation to result analysis in line with the hydrological workflow, whereas some others do not provide enough features and guidance to allow the user to operate the models with their complete specificities. Such choices by the package authors raise the question of the initial purpose behind the development of these packages. Nevertheless, whether it is for teaching purposes or complex hydrological studies, detailed documentation is always important to ensure reusable methods and appropriate use of the packages.

Our attempt at easing the preliminary research process should help the R users to select the packages that best suit their requirements. The provided framework containing examples on how to use the models included in the selected packages represents a first step towards more comprehensible and operable R packages for hydrological modelling. With the same aim, the unified representations of models and packages usability result in strengthened materials associated with these specific packages. While some limitations regarding package practicalities have arisen from our analysis, we hope that our framework will help developers in improving their packages by meeting the needs of more transferable methods. In this regard, numerous features that were not examined in this article are currently under development.

Although we have tried to thoroughly select the packages, this work does not attempt at being exhaustive regarding the number of packages. Indeed, as pointed out in Sect. 2.1, some hydrological models with more data requirements and very different purposes exist in the R environment but were not included in our study. Furthermore, one might ask whether the choice of package should exclusively rely on the performance of the models. We could argue that in conceptual hydrology modelling, models are mainly assessed by their ability to reproduce river discharges at the gauging station of a catchment (Singh et al., 2017). Therefore, a hydrologist who would need to use an R package for modelling might favour the efficiency of a given model in their selection procedure. Even though we have run the models contained in the selected packages several times on different datasets, we have chosen to not show any comparison of model performance, which would be dependent on specific catchment characteristics and more representative of the model than of the package itself. We provide with this comparison R scripts to test one parameter set for each model on a simple hydrology example, whereas it is becoming increasingly important to be able to use sensitivity analysis to explore location-specific behaviour of a model (Haghnegahdar et al., 2017; Blair et al., 2019), uncertainty and identifiability analysis to understand ability to estimate parameters (Guillaume et al., 2019), testing of alternative model structure hypotheses (Clark et al., 2011) and defining limits of acceptability for model rejection (Beven, 2016, 2018). ~~Numerical properties of the model implementations have also not been investigated (Clark and Kavetski, 2010).~~ Comparison of models implemented in R with others in Python, Matlab or proprietary packages is similarly left to future work.

This work could be considered as an early stage version of a meta-package that would manage to run all the packages through the same R architecture, thus improving guidance, appropriate constraint and reliable comparisons when using hydrological models with R.

Code availability. R scripts to use the packages on simple hydrology examples are provided under the terms of the GNU General Public License 2.0 and available at: https://doi.org/10.15454/3PPKCL. They enable the application of each package on a simple hydrology example. These scripts consist of basic input data or spatial data preparation and data check procedures, R formatting functions, run of a single parameter set for each model, calculation of the KGE criterion and plot of results. For the packages that include a snow-accounting function, the R scripts include the related steps. We do not provide the hydrometeorological data that were used in these scripts. However, all the packages except `HBV.IANIGLA` include one or several example datasets. These scripts do not provide R commands to run parameter estimation procedures or to perform uncertainty analyses. These procedures are important steps of the hydrological workflow and should be considered in light of the specificities of the different models and the extensive literature on the subject.

Appendix A: Plots by some of the selected packages

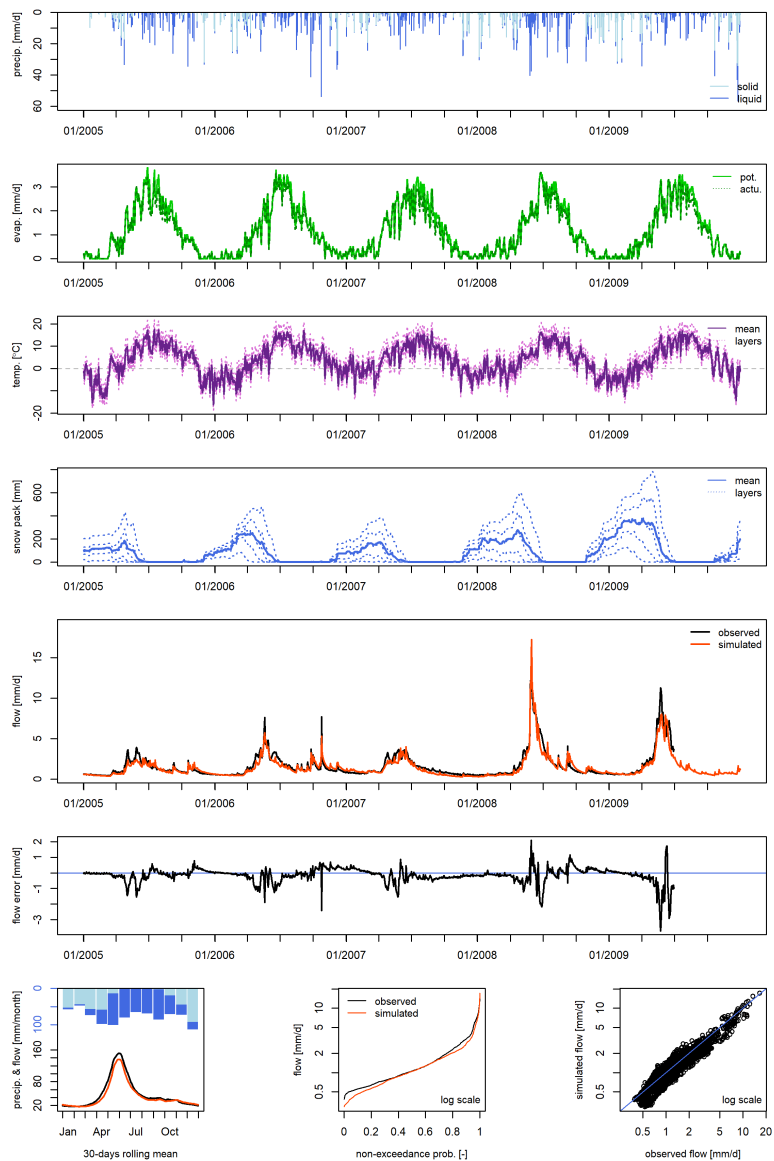



Figure A1. Example of plots enabled by the `airGR` package. This example was generated by using one of the datasets included in the package and simulations with the GR4J-CemaNeige hydrological model. From top to bottom: solid and liquid precipitation time series (TS); potential and actual evapotranspiration TS; temperature TS for each layer; snow pack TS for each layer; simulated and observed hydrographs; flow error (or residuals). On the bottom line from left to right: interannual monthly median; cumulative frequency; bias. The hydrographs and the flow error charts can be visualised with a log scale.

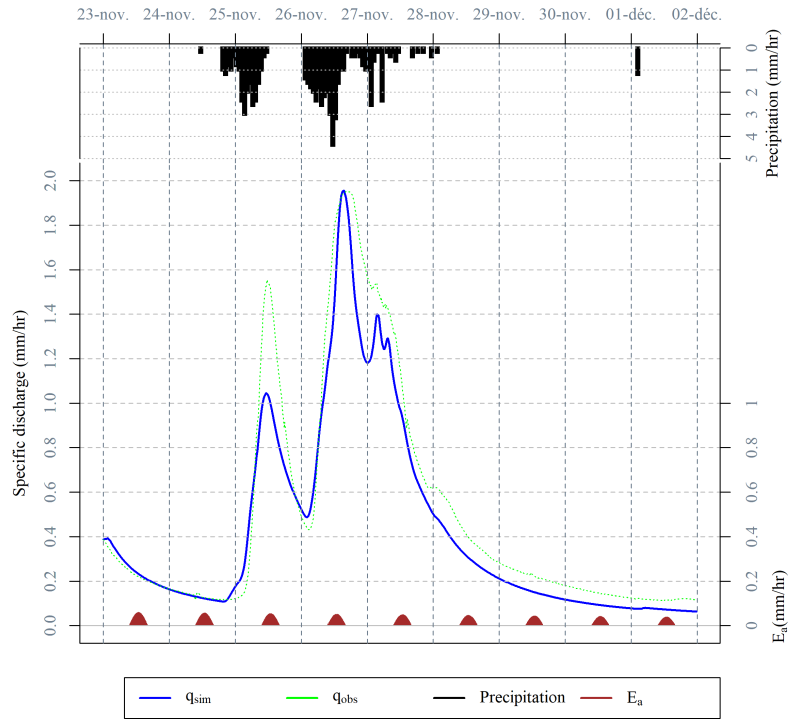



Figure A2. Example of a plots enabled by the `dynatopmodel` package. This example was generated by using the dataset included in the package. This figure combines plots of actual evapotranspiration time series, precipitation time series, simulated and observed hydrograph.

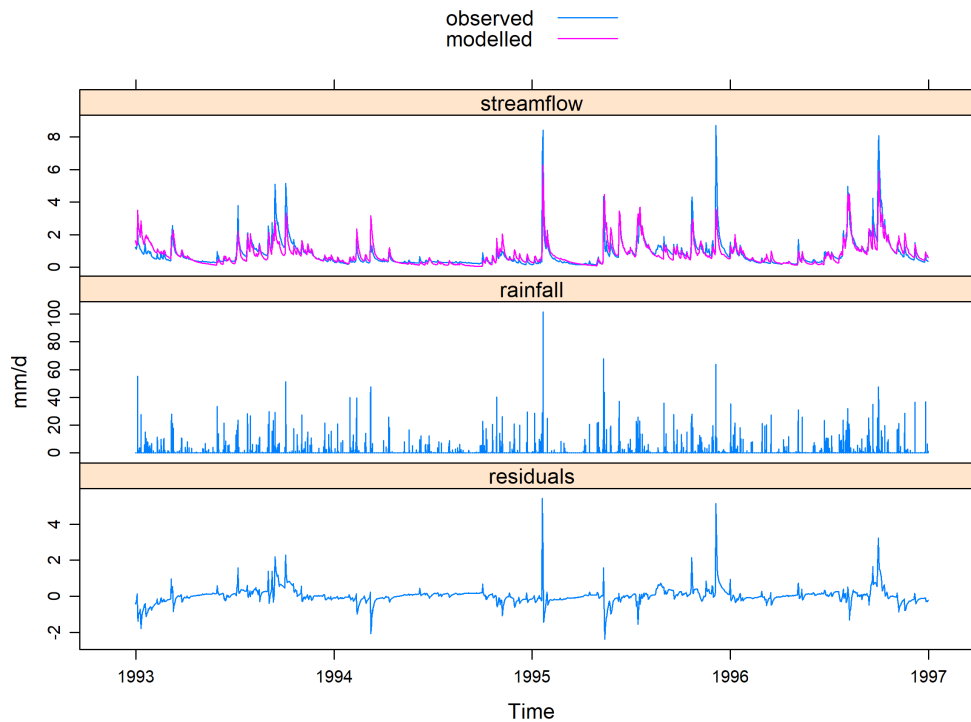



Figure A3. Example of plots enabled by the `hydromad` package. This example was generated by using one of the datasets included in the package and simulations with the IHACRES-CMD model. From top to bottom: observed and simulated hydrographs; precipitation time series; flow error (or residuals). These charts can be visualised with a log scale.

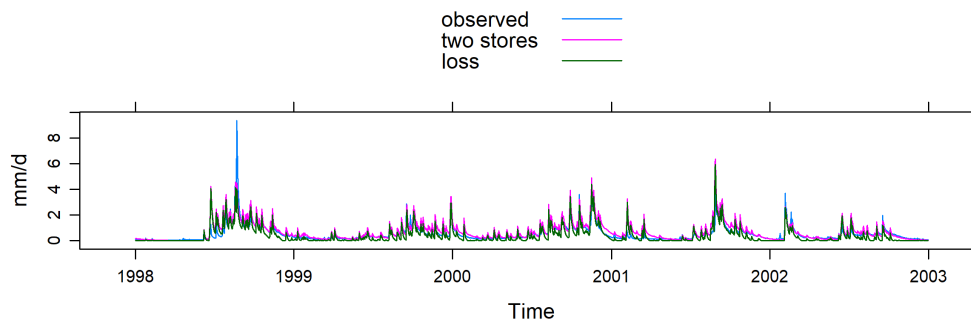



Figure A4. Example of plots of simulation results from two model configurations enabled by the `hydromad` package. This example was generated by using one of the datasets included in the package and simulations with the IHACRES-CMD model. This chart can be visualised with a log scale.

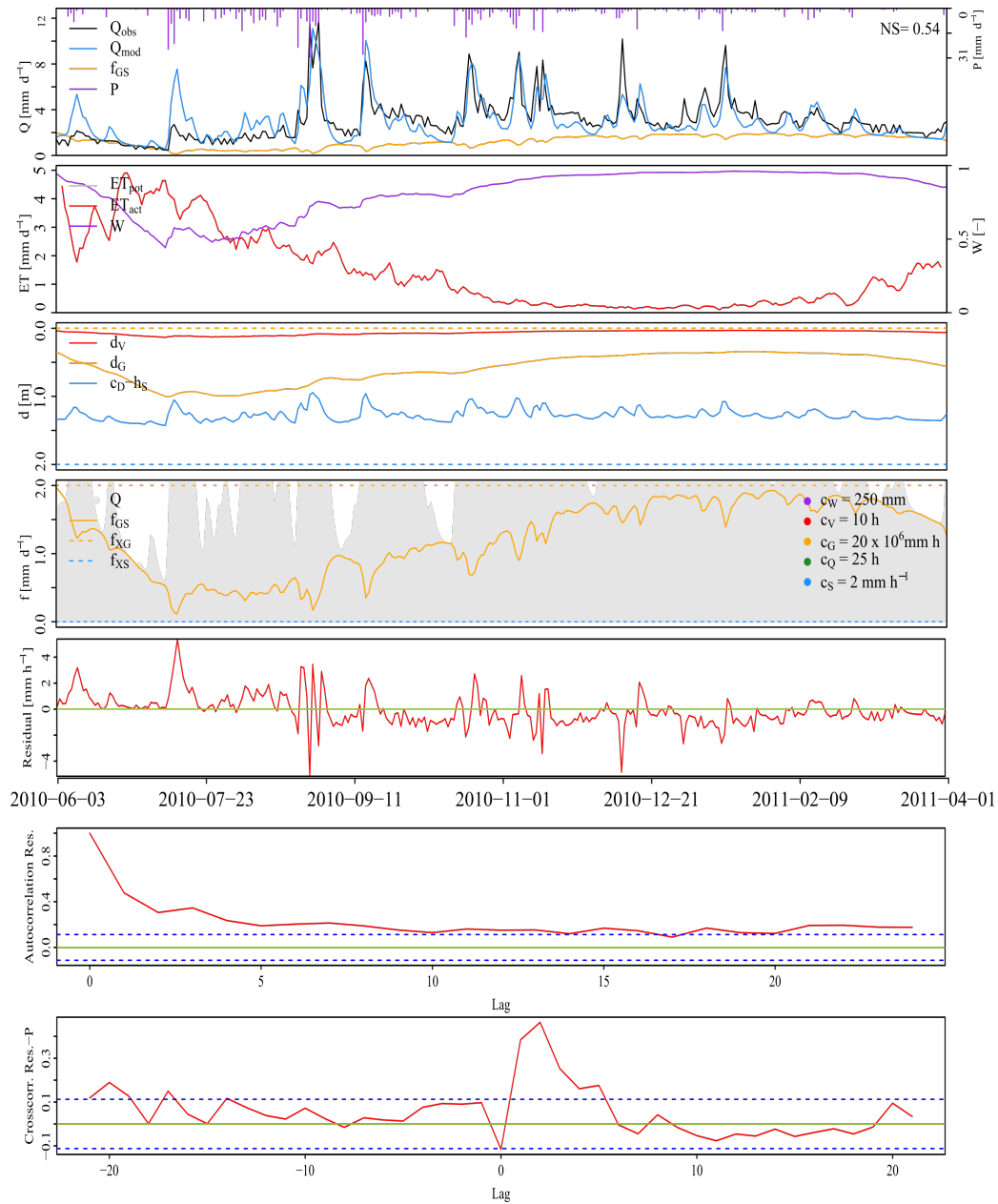



Figure A5. Example of plots enabled by the `WALRUS` package. This example was generated by using one of the datasets included in the package and simulations with the WALRUS model. From top to bottom: precipitation time series (TS), simulated hydrograph, observed hydrograph and modelled groundwater drainage; potential evapotranspiration TS, modelled actual evapotranspiration TS, wetness index TS; soil reservoir level TS and surface reservoir level TS; seepage TS and surface water supply or extraction TS; flow error (or residuals); autocorrelation of residuals; cross correlation between residuals and precipitation.

Appendix B: Graphical user interfaces

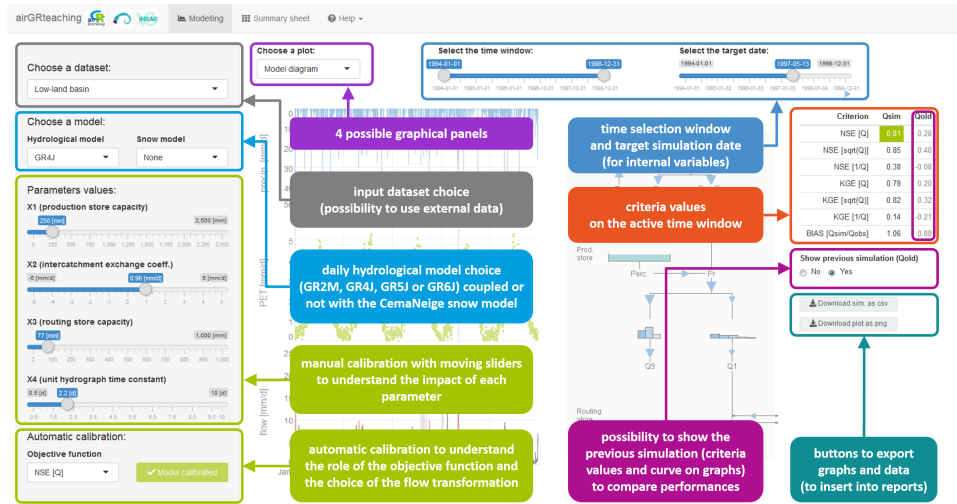



Figure B1. Graphical user interface of `airGR` using the `airGRteaching` package: diagram describing the different functionalities.

(a) (b) (c)

(d) (e) (f)

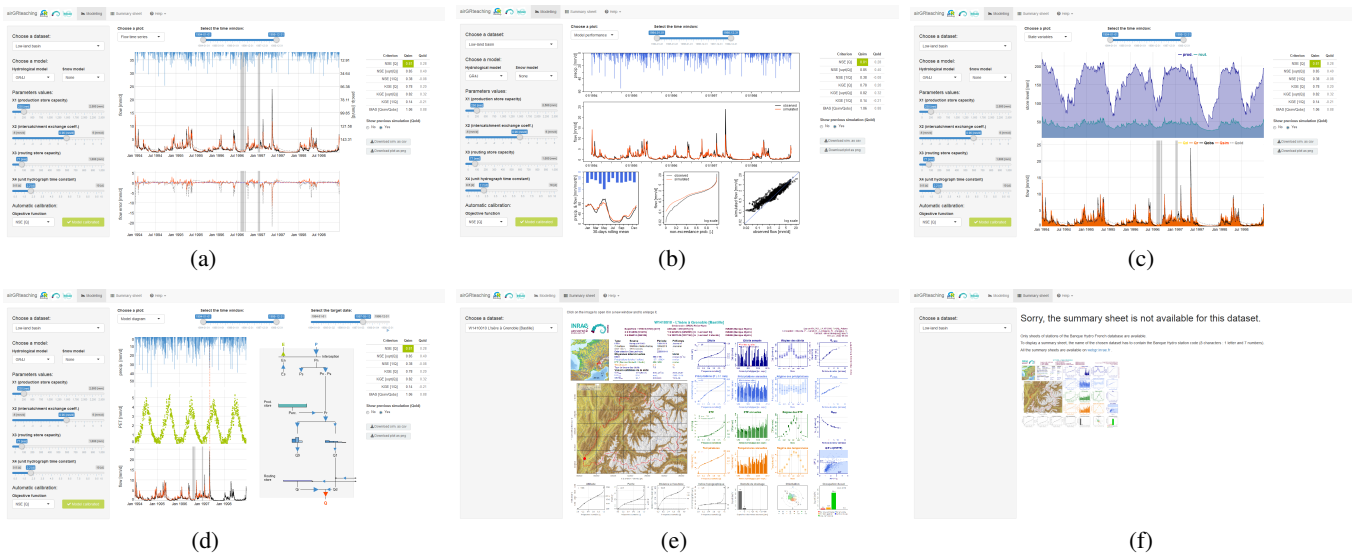



Figure B2. Graphical user interface of `airGR` using the `airGRteaching` package: illustration of the six graphical panels. (a) panel displaying the time series of flow error (bottom graphic) and simulated and observed hydrographs (top graphic). (b) performance graphics panel. (c) panel displaying the time series of runoff components (bottom graphic) and reservoir levels (top graphic). (d) panel displaying the interactive model diagram (right graphic) and the time series (left graphic) of precipitation, potential evapotranspiration, simulated flows and observed flows (from top to bottom). (e) panel displaying several hydrometeorological characteristics of the selected catchment. (f) when catchment characteristics are not available.

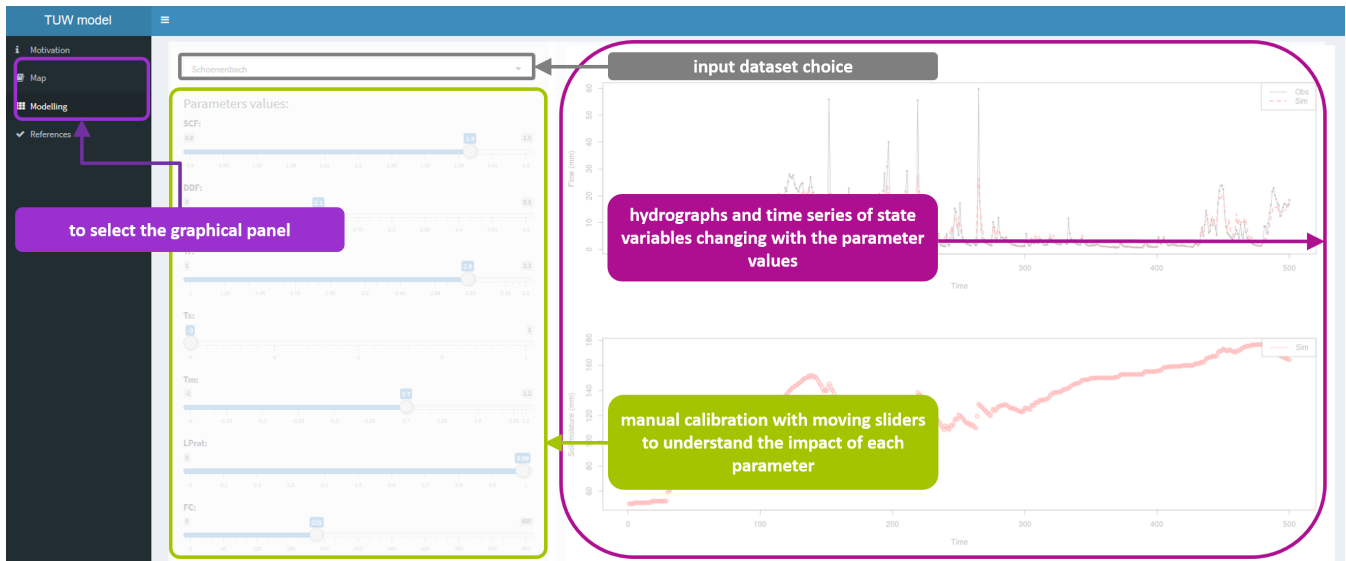



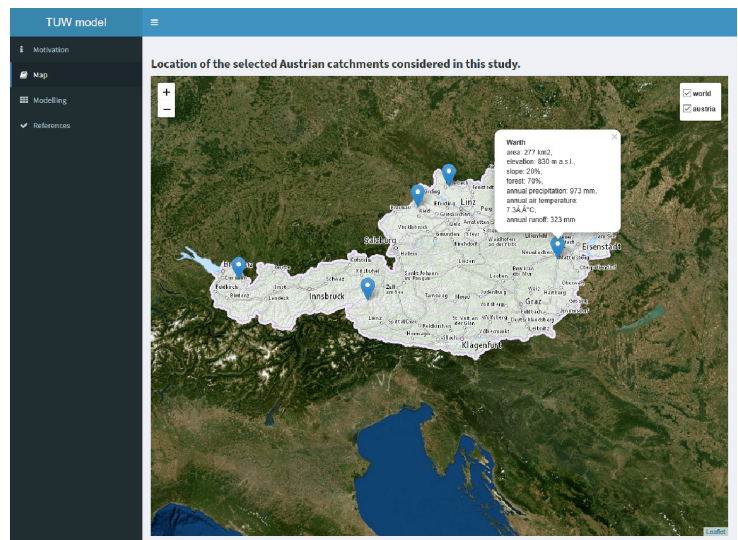
Figure B3. Graphical user interface of `TUWmodel`: diagram describing the different functionalities.

(a)               (b)

Figure B4. Graphical user interface of `TUWmodel`. (a) tab where users can adjust the model parameters and see the impacts on (from top to bottom): the simulated hydrograph compared to the observed hydrograph; the simulated time series of soil moisture; the simulated time series of melt; the simulated time series of snow water equivalent. (b) tab where users can visualise the localisation of the five available catchment datasets on a map of Austria.

Author contributions. PA, GT and OD designed the study. PA conducted the analyses and wrote the manuscript. All authors discussed the design, the results and contributed to the final manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors thank Météo-France and the hydrometric services of the French government (SCHAPI) for respectively providing the climatic and hydrological data used to calculate the computation times. The SCHAPI is also acknowledged for providing financial support to the first author. We also thank the French national geographic institute (IGN) for providing the digital elevation model data. `hydromad` development has been supported by the Sydney Informatics Hub, a Core Research Facility of the University of Sydney. We would like to thank Charles Perrin, Vazken Andréassian, Maria-Helena Ramos, François Bourgin and Léonard Santos for their useful remarks and suggestions on the manuscript. Finally, we thank the Editor Jan Seibert, Elena Toth Mattia Neri and one anonymous referee for their comments that significantly improved the manuscript.

References

Anderson, E. A.: A point energy and mass balance model of a snow cover, vol. 19, US Department of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, Office of Hydrology., 1976.

Anderson, E. A.: Snow accumulation and ablation model SNOW-17, NOAA's National Weather Service Hydrology Laboratory NWSRFS user manual, 61 pp, 2006.

Andréassian, V., Perrin, C., Berthet, L., Le Moine, N., Lerat, J., Loumagne, C., Oudin, L., Mathevet, T., Ramos, M.-H., and Valéry, A.: HESS Opinions "Crash tests for a standardized evaluation of hydrological models", Hydrology and Earth System Sciences, 13, 1757–1764, https://doi.org/10.5194/hess-13-1757-2009, 2009.

Andrews, F. T. and Guillaume, J. H.: hydromad: Hydrological Model Assessment and Development, http://hydromad.catchment.org/, R package version 0.9-26, 2018.

Andrews, F. T., Croke, B. F. W., and Jakeman, A. J.: An open software environment for hydrological model assessment and development, Environmental Modelling and Software, 26, 1171–1185, https://doi.org/10.1016/j.envsoft.2011.04.006, 2011.

Arabzadeh, R. and Araghinejad, S.: RHMS: Hydrologic Modelling System for R Users, https://CRAN.R-project.org/package=RHMS, R package version 1.6, 2019.

Ardia, D., Mullen, K. M., Peterson, B. G., and Ulrich, J.: DEoptim: Differential Evolution in R, https://CRAN.R-project.org/package=DEoptim, R package version 2.2-5, 2020.

Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., Nelson, R. A., Sayre, D., Sheridan, P. B., Stern, H., Ziller, I., Hughes, R. A., and Nutt, R.: The FORTRAN Automatic Coding System, in: Papers Presented at the February 26-28, 1957, Western Joint Computer Conference: Techniques for Reliability, IRE-AIEE-ACM '57 (Western), pp. 188–198, ACM, New York, NY, USA, https://doi.org/10.1145/1455567.1455599, 1957.

Becker, R. A., Chambers, J. M., and Wilks, A. R.: The New S Language: A Programming Environment for Data Analysis and Graphics, Wadsworth and Brooks/Cole Advanced Books & Software, USA, 1988.

Bergström, S.: Development and Application of a Conceptual Runoff Model for Scandinavian Catchments, vol. 134 pp., SMHI Rep. RHO 7, 1976.

Bergström, S. and Lindström, G.: Interpretation of runoff processes in hydrological modelling—experience from the HBV approach, Hydrological Processes, 29, 3535–3545, https://doi.org/10.1002/hyp.10510, 2015.

Beven, K. J.: TOPMODEL: a critique, Hydrological Processes, 11, 1069–1085, https://doi.org/10.1002/(SICI)1099-1085(199707)11:9<1069::AID-HYP545>3.0.CO;2-O, 1997.

Beven, K. J.: A manifesto for the equifinality thesis, Journal of Hydrology, 320, 18–36, https://doi.org/10.1016/j.jhydrol.2005.07.007, 2006.

Beven, K. J.: Rainfall-runoff modelling: the primer (2nd edition), John Wiley & Sons, https://doi.org/10.1002/9781119951001, 2012.

Beven, K. J.: Facets of uncertainty: epistemic uncertainty, non-stationarity, likelihood, hypothesis testing, and communication, Hydrological Sciences Journal, 61, 1652–1665, https://doi.org/10.1080/02626667.2015.1031761, 2016.

Beven, K. J.: On hypothesis testing in hydrology: Why falsification of models is still a really good idea, WIREs Water, 5, e1278, https://doi.org/10.1002/wat2.1278, 2018.

Beven, K. J.: Towards a methodology for testing models as hypotheses in the inexact sciences, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 475, 20180 862, https://doi.org/10.1098/rspa.2018.0862, 2019.

Beven, K. J. and Binley, A.: The future of distributed models: Model calibration and uncertainty prediction, Hydrological Processes, 6, 279–298, https://doi.org/10.1002/hyp.3360060305, 1992.

Beven, K. J. and Chappell, N.: Perceptual perplexity and parameter parsimony, WIREs Water, under review, 2020.

Beven, K. J. and Freer, J.: A dynamic TOPMODEL, Hydrological Processes, 15, 1993–2011, https://doi.org/10.1002/hyp.252, 2001.

Beven, K. J. and Kirby, M. J.: A physically based, variable contributing area model of basin hydrology, Hydrological Sciences Bulletin, 24, 43–69, https://doi.org/10.1080/02626667909491834, 1979.

Beven, K. J., Lamb, R., Quinn, P., Romanwvic, R., and Freer, J.: TOPMODEL, in: Computer models of watershed hydrology, edited by Singh, V. P., p. 627, Water Resources Publications, Colorado, 1995.

Beven, K. J., Lamb, R., Kirkby, M. J., and Freer, J. E.: A history of TOPMODEL, Hydrology and Earth System Sciences Discussions, pp. 1–44, https://doi.org/10.5194/hess-2020-409, 2020.

Blair, G. S., Beven, K. J., Lamb, R., Bassett, R., Cauwenberghs, K., Hankin, B., Dean, G., Hunter, N., Edwards, L., Nundloll, V., Samreen, F., Simm, W., and Towe, R.: Models of everywhere revisited: A technological perspective, Environmental Modelling and Software, 122, 104 521, https://doi.org/10.1016/j.envsoft.2019.104521, 2019.

Blöschl, G.: Debates—Hypothesis testing in hydrology: Introduction, Water Resources Research, 53, 1767–1769, https://doi.org/10.1002/2017WR020584, 2017.

Boughton, W.: The Australian water balance model, Environmental Modelling and Software, 19, 943–956, https://doi.org/10.1016/j.envsoft.2003.10.007, 2004.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M.: Time series analysis: forecasting and control, John Wiley & Sons, 2015.

Brauer, C. C., Teuling, A. J., Torfs, P. J. J. F., and Uijlenhoet, R.: The Wageningen Lowland Runoff Simulator (WALRUS): a lumped rainfall-runoff model for catchments with shallow groundwater, Geoscientific Model Development, 7, 2313–2332, https://doi.org/10.5194/gmd-7-2313-2014, 2014a.

Brauer, C. C., Torfs, P. J. J. F., Teuling, A. J., and Uijlenhoet, R.: The Wageningen Lowland Runoff Simulator (WALRUS): application to the Hupsel Brook catchment and Cabauw polder, Hydrology and Earth System Science, 18, 4007–4028, https://doi.org/10.5194/hess-18-4007-2014, 2014b.

Brauer, C. C., Torfs, P. J. J. F., Teuling, A. J., and Uijlenhoet, R.: The Wageningen Lowland Runoff Simulator WALRUS 1.10, User manual, www.github.com/ClaudiaBrauer/WALRUS, R package version 1.10, 2017.

Burnash, R. J. C.: The NWS River Forecast System – Catchment Modeling, in: Computer models of watershed hydrology, edited by Singh, V. P., pp. 311–366, Water Resources Publications, Colorado, 1995.

Buytaert, W.: topmodel: Implementation of the Hydrological Model TOPMODEL in R, https://CRAN.R-project.org/package=topmodel, R package version 0.7.3, 2018.

Calder, I. R.: A stochastic model of rainfall interception, Journal of Hydrology, 89, 65–71, https://doi.org/10.1016/0022-1694(86)90143-5, 1986.

Calder, I. R., Harding, R. J., and Rosier, P. T. W.: An objective assessment of soil-moisture deficit models, Journal of Hydrology, 60, 329–355, https://doi.org/10.1016/0022-1694(83)90030-6, 1983.

Ceola, S., Arheimer, B., Baratti, E., Blöschl, G., Capell, R., Castellarin, A., Freer, J., Han, D., Hrachowitz, M., Hundecha, Y., Hutton, C., Lindström, G., Montanari, A., Nijzink, R., Parajka, J., Toth, E., Viglione, A., and Wagener, T.: Virtual laboratories: new opportunities for collaborative water science, Hydrology and Earth System Sciences, 19, 2101–2117, https://doi.org/10.5194/hess-19-2101-2015, 2015.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J.: shiny: Web Application Framework for R, https://CRAN.R-project.org/package=shiny, R package version 1.4.0, 2019.

Chen, C., Garibaldi, J., and Razak, T.: FuzzyR: Fuzzy Logic Toolkit for R, https://CRAN.R-project.org/package=FuzzyR, R package version 2.3, 2019.

Clark, M. P. and Kavetski, D.: Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes, Water Resources Research, 46, https://doi.org/10.1029/2009WR008894, 2010.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., and Hay, L. E.: Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, Water Resources Research, 44, https://doi.org/10.1029/2007WR006735, 2008.

Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the method of multiple working hypotheses for hydrological modeling, Water Resources Research, 47, https://doi.org/10.1029/2010WR009827, 2011.

Clark, M. P., Bierkens, M. F. P., Samaniego, L., Woods, R. A., Uijlenhoet, R., Bennett, K. E., Pauwels, V. R. N., Cai, X., Wood, A. W., and Peters-Lidard, C. D.: The evolution of process-based hydrologic models: historical challenges and the collective quest for physical realism, Hydrology and Earth System Sciences, 21, 3427–3440, https://doi.org/10.5194/hess-21-3427-2017, 2017.

Coron, L., Thirel, G., Delaigue, O., Perrin, C., and Andréassian, V.: The Suite of Lumped GR Hydrological Models in an R Package, Environmental Modelling and Software, 94, 166–171, https://doi.org/10.1016/j.envsoft.2017.05.002, 2017.

Coron, L., Delaigue, O., Thirel, G., Perrin, C., and Michel, C.: airGR: Suite of GR Hydrological Models for Precipitation-Runoff Modelling, https://doi.org/10.15454/EX11NA, https://CRAN.R-project.org/package=airGR/, R package version 1.4.3.65., 2020.

Crochemore, L., Perrin, C., Andréassian, V., Ehret, U., Seibert, S. P., Grimaldi, S., Gupta, H. V., and Paturel, J.-E.: Comparing expert judgement and numerical criteria for hydrograph evaluation, Hydrological Sciences Journal, 60, 402–423, https://doi.org/10.1080/02626667.2014.903331, 2015.

Croke, B. F. W. and Jakeman, A. J.: A catchment moisture deficit module for the IHACRES rainfall-runoff model, Environmental Modelling and Software, 19, 1–5, https://doi.org/10.1016/j.envsoft.2003.09.001, 2004.

Dal Molin, M., Fenicia, F., and Kavetski, D.: SuperflexPy: the flexible language of hydrological modelling, https://superflexpy.readthedocs.io/en/latest/index.html, version 1.2.0, 2020.

Danish Hydraulic Institute: MIKE SHE: Volume 2: Reference guide, https://manuals.mikepoweredbydhi.help/2017/Water_Resources/MIKE_SHE_Printed_V2.pdf, DHI, Netherlands, 2017.

de Boer-Euser, T., Bouaziz, L., De Niel, J., Brauer, C. C., Dewals, B., Drogue, G., Fenicia, F., Grelier, B., Nossent, J., Pereira, F., Savenije, H. H. G., Thirel, G., and Willems, P.: Looking beyond general metrics for model comparison – lessons from an international model intercomparison study, Hydrology and Earth System Sciences, 21, 423–440, https://doi.org/10.5194/hess-21-423-2017, 2017.

Delaigue, O., Thirel, G., Coron, L., and Brigode, P.: airGR and airGRteaching: two open source tools for rainfall-runoff modeling and teaching hydrology, HIC2018 proceedings, 13th International conference of Hydroinformatics, July 2018, Palermo, Italy, 2018.

Delaigue, O., Coron, L., and Brigode, P.: airGRteaching: Teaching Hydrological Modelling with GR (Shiny Interface Included), https://doi.org/10.15454/W0SSKT, https://CRAN.R-project.org/package=airGRteaching/, R package version 0.2.8.69., 2020a.

Delaigue, O., Génot, B., Lebecherel, L., Brigode, P., and Bourgin, P.: Database of watershed-scale hydroclimatic observations in France, https://webgr.inrae.fr/base-de-donnees, INRAE, HYCAR Research Unit, Hydrology group, Antony, 2020b.

Ficchì, A., Perrin, C., and Andréassian, V.: Hydrological modelling at multiple sub-daily time steps: Model improvement via flux-matching, Journal of Hydrology, 575, 1308–1327, https://doi.org/10.1016/j.jhydrol.2019.05.084, 2019.

Freer, J. E., McMillan, H., McDonnell, J. J., and Beven, K. J.: Constraining dynamic TOPMODEL responses for imprecise water table information using fuzzy rule based performance measures, Journal of Hydrology, 291, 254–277, https://doi.org/10.1016/j.jhydrol.2003.12.037, 2004.

Fuka, D. R., Walter, M. T., Steenhuis, T. S., and Easton, Z. M.: SWATmodel: A multi-OS implementation of the TAMU SWAT model, https://CRAN.R-project.org/package=SWATmodel, R package version 0.5.9, 2014.

Georgakakos, K. P.: Analytical results for operational flash flood guidance, Journal of Hydrology, 317, 81–103, https://doi.org/10.1016/j.jhydrol.2005.05.009, 2006.

Guillaume, J. H., Jakeman, J. D., Marsili-Libelli, S., Asher, M., Brunner, P., Croke, B., Hill, M. C., Jakeman, A. J., Keesman, K. J., Razavi, S., and Stigter, J. D.: Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose, Environmental Modelling and Software, 119, 418–432, https://doi.org/10.1016/j.envsoft.2019.07.007, 2019.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F.: Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling, Journal of Hydrology, 377, 80–91, https://doi.org/10.1016/j.jhydrol.2009.08.003, 2009.

Haghnegahdar, A., Razavi, S., Yassin, F., and Wheater, H.: Multicriteria sensitivity analysis as a diagnostic tool for understanding model behaviour and characterizing model uncertainty, Hydrological Processes, 31, 4462–4476, https://doi.org/10.1002/hyp.11358, 2017.

Hamon, W.: Estimating potential evapotranspiration, Master's thesis, Massachusetts Institute of Technology, 82 pages, 1960.

Hijmans, R. J.: raster: Geographic Data Analysis and Modeling, https://CRAN.R-project.org/package=raster, R package version 3.0-12, 2020.

Holländer, H. M., Bormann, H., Blume, T., Buytaert, W., Chirico, G. B., Exbrayat, J.-F., Gustafsson, D., Hölzel, H., Krauße, T., Kraft, P., Stoll, S., Blöschl, G., and Flühler, H.: Impact of modellers' decisions on hydrological a priori predictions, Hydrology and Earth System Sciences, 18, 2065–2085, https://doi.org/10.5194/hess-18-2065-2014, 2014.

Hrachowitz, M. and Clark, M. P.: HESS Opinions: The complementary merits of competing modelling philosophies in hydrology, Hydrology and Earth System Sciences, 21, 3953–3973, https://doi.org/10.5194/hess-21-3953-2017, 2017.

Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., and Arheimer, B.: Most computational hydrology is not reproducible, so is it really science?, Water Resources Research, 52, 7548–7555, https://doi.org/10.1002/2016WR019285, 2016.

IGN: BD ALTI: modèle numérique de terrain maillé qui décrit le territoire français à moyenne échelle, https://professionnels.ign.fr/bdalti, version 1.0, 2013.

Jakeman, A. J. and Hornberger, G. M.: How much complexity is warranted in a rainfall-runoff model?, Water Resources Research, 29, 2637–2649, https://doi.org/10.1029/93WR00877, 1993.

Jakeman, A. J., Littlewood, I. G., and Whitehead, P. G.: Computation of the instantaneous unit hydrograph and identifiable component flows with application to two small upland catchments, Journal of Hydrology, 117, 275–300, https://doi.org/10.1016/0022-1694(90)90097-H, 1990.

Jakeman, A. J., Letcher, R. A., and Norton, J. P.: Ten iterative steps in development and evaluation of environmental models, Environmental Modelling and Software, 21, 602–614, https://doi.org/10.1016/j.envsoft.2006.01.004, 2006.

Jansson, P., Hock, R., and Schneider, T.: The concept of glacier storage: a review, Journal of Hydrology, 282, 116–129, https://doi.org/10.1016/S0022-1694(03)00258-0, 2003.

Kavetski, D. and Clark, M. P.: Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction, Water Resources Research, 46, https://doi.org/10.1029/2009WR008896, 2010.

Kernighan, B. W. and Ritchie, D. M.: The C Programming Language, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1978.

Kitanidis, P. K. and Bras, R. L.: Real-time forecasting with a conceptual hydrologic model: 2. Applications and results, Water Resources Research, 16, 1034–1044, https://doi.org/10.1029/WR016i006p01034, 1980.

1245   Kling, H., Fuchs, M., and Paulin, M.: Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios, Journal of Hydrology, 424–425, 264–277, https://doi.org/10.1016/j.jhydrol.2012.01.011, 2012.

Knoben, W. J. M., Freer, J. E., Fowler, K. J. A., Peel, M. C., and Woods, R. A.: Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v1.2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations, Geoscientific Model Development, 12, 2463–2480, https://doi.org/10.5194/gmd-12-2463-2019, 2019.

1250   Koutsoyiannis, D. and Montanari, A.: Negligent killing of scientific concepts: the stationarity case, Hydrological Sciences Journal, 60, 1174–1183, https://doi.org/10.1080/02626667.2014.959959, 2015.

Kustas, W. P., Rango, A., and Uijlenhoet, R.: A simple energy budget algorithm for the snowmelt runoff model, Water Resources Research, 30, 1515–1527, 1994.

Le Moine, N.: Le bassin versant de surface vu par le souterrain: une voie d'amélioration des performances et du réalisme des modèles
1255   pluie-débit ?, Ph.D. thesis, University of Pierre and Marie Curie (Paris), CEMAGREF (Antony), France, 2008.

Leleu, I., Tonnelier, I., Puechberty, R., Gouin, P., Viquendi, I., Cobos, L., Foray, A., Baillon, M., and Ndima, P.-O.: La refonte du système d'information national pour la gestion et la mise à disposition des données hydrométriques, La Houille Blanche, pp. 25–32, http://hydro.eaufrance.fr/, 2014.

Lindström, G., Johansson, B., Persson, M., Gardelin, M., and Bergström, S.: Development and test of the distributed HBV-96 hydrological
1260   model, Journal of Hydrology, 201, 272–288, https://doi.org/10.1016/S0022-1694(97)00041-3, 1997.

Littlewood, I. G.: Improved unit hydrograph characterisation of the daily flow regime (including low flows) for the River Teifi, Wales: towards better rainfall-streamflow models for regionalisation, Hydrology and Earth System Sciences, 6, 899–911, https://doi.org/10.5194/hess-6-899-2002, 2002.

Lohmann, D., Nolte-Holube, R., and Raschke, E.: A large-scale horizontal routing model to be coupled to land surface parametrization
1265   schemes, Tellus A, 48, 708–721, https://doi.org/10.1034/j.1600-0870.1996.t01-3-00009.x, 1996.

Mathevet, T.: Quels modèles pluie-débit globaux pour le pas de temps horaire? Développement empirique et comparaison de modèles sur un large échantillon de bassins versants, Ph.D. thesis, CEMAGREF (Antony), ENGREF (Paris), France, 463 pages, 2005.

Melsen, L. A., Torfs, P. J. J. F., Uijlenhoet, R., and Teuling, A. J.: Comment on "Most computational hydrology is not reproducible, so is it really science?" by Christopher Hutton et al., Water Resources Research, 53, 2568–2569, https://doi.org/10.1002/2016WR020208, 2017.

1270   Melsen, L. A., Teuling, A. J., Torfs, P. J. J. F., Zappa, M., Mizukami, N., Mendoza, P. A., Clark, M. P., and Uijlenhoet, R.: Subjective modeling decisions can significantly impact the simulation of flood and drought events, Journal of Hydrology, 568, 1093–1104, https://doi.org/10.1016/j.jhydrol.2018.11.046, 2019.

Mersmann, O.: microbenchmark: Accurate Timing Functions, https://CRAN.R-project.org/package=microbenchmark, R package version 1.4-7, 2019.

1275   Metcalfe, P., Beven, K. J., and Freer, J.: Dynamic TOPMODEL: A new implementation in R and its sensitivity to time and space steps, Environmental Modelling and Software, 72, 155–172, https://doi.org/10.1016/j.envsoft.2015.06.010, 2015.

Metcalfe, P., Beven, K. J., and Freer, J.: dynatopmodel: Implementation of the Dynamic TOPMODEL Hydrological Model, https://CRAN.R-project.org/package=dynatopmodel, R package version 1.2.1, 2018.

Michel, C.: Que peut-on faire en hydrologie avec modèle conceptuel à un seul paramètre ?, La Houille Blanche, pp. 39–44,
1280   https://doi.org/10.1051/lhb/1983004, 1983.

Michel, C.: Hydrologie appliquée aux petits bassins ruraux, Hydrology handbook, 1991.

Mouelhi, S.: Vers une chaîne cohérente de modèles pluie-débit conceptuels globaux aux pas de temps pluriannuel, annuel, mensuel et journalier, Ph.D. thesis, ENGREF (Paris), CEMAGREF (Antony), France, 2003.

Mouelhi, S., Michel, C., Perrin, C., and Andréassian, V.: Stepwise development of a two-parameter monthly water balance model, Journal of Hydrology, 318, 200–214, https://doi.org/10.1016/j.jhydrol.2005.06.014, 2006.

Nash, J. and Sutcliffe, J.: River flow forecasting through conceptual models part I — A discussion of principles, Journal of Hydrology, 10, 282–290, https://doi.org/10.1016/0022-1694(70)90255-6, 1970.

Nicolle, P., Pushpalatha, R., Perrin, C., François, D., Thiéry, D., Mathevet, T., Le Lay, M., Besson, F., Soubeyroux, J.-M., Viel, C., Regimbeau, F., Andréassian, V., Maugis, P., Augeard, B., and Morice, E.: Benchmarking hydrological models for low-flow simulation and forecasting on French catchments, Hydrology and Earth System Sciences, 18, 2829–2857, https://doi.org/10.5194/hess-18-2829-2014, 2014.

Oudin, L., Hervieu, F., Michel, C., Perrin, C., Andréassian, V., Anctil, F., and Loumagne, C.: Which potential evapotranspiration input for a lumped rainfall–runoff model?: Part 2—Towards a simple and efficient potential evapotranspiration model for rainfall–runoff modelling, Journal of Hydrology, 303, 290–306, https://doi.org/10.1016/j.jhydrol.2004.08.026, 2005.

Parajka, J., Merz, R., and Blöschl, G.: Uncertainty and multiple objective calibration in regional water balance modelling: case study in 320 Austrian catchments, Hydrological Processes, 21, 435–446, https://doi.org/10.1002/hyp.6253, 2007.

Patil, S. and Stieglitz, M.: Modelling daily streamflow at ungauged catchments: what information is necessary?, Hydrological Processes, 28, 1159–1169, https://doi.org/10.1002/hyp.9660, 2014.

Perrin, C., Michel, C., and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, Journal of Hydrology, 279, 275–289, https://doi.org/10.1016/S0022-1694(03)00225-7, 2003.

Pushpalatha, R., Perrin, C., Le Moine, N., Mathevet, T., and Andréassian, V.: A downward structural sensitivity analysis of hydrological models to improve low-flow simulation, Journal of Hydrology, 411, 66–76, https://doi.org/10.1016/j.jhydrol.2011.09.034, 2011.

Quinn, P., Beven, K. J., Chevallier, P., and Planchon, O.: The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models, Hydrological Processes, 5, 59–79, https://doi.org/10.1002/hyp.3360050106, 1991.

R Core Team: R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/, 2020a.

R Core Team: Writing R extensions, R Foundation for Statistical Computing, Vienna, Austria, https://cran.r-project.org/doc/manuals/r-release/R-exts.pdf, 2020b.

Rozalis, S., Morin, E., Yair, Y., and Price, C.: Flash flood prediction using an uncalibrated hydrological model and radar rainfall data in a Mediterranean watershed under changing hydrological conditions, Journal of Hydrology, 394, 245–255, https://doi.org/10.1016/j.jhydrol.2010.03.021, 2010.

Santos, L., Thirel, G., and Perrin, C.: Technical note: Pitfalls in using log-transformed flows within the KGE criterion, Hydrology and Earth System Sciences, 22, 4583–4591, https://doi.org/10.5194/hess-22-4583-2018, 2018a.

Santos, L., Thirel, G., and Perrin, C.: Continuous state-space representation of a bucket-type rainfall-runoff model: a case study with the GR4 model using state-space GR4 (version 1.0), Geoscientific Model Development, 11, 1591–1605, https://doi.org/10.5194/gmd-11-1591-2018, 2018b.

Schmidt-Walter, P., Trotsiuk, V., Meusburger, K., Zacios, M., and Meesenburg, H.: Advancing simulations of water fluxes, soil moisture and drought stress by using the LWF-Brook90 hydrological model in R, Agricultural and Forest Meteorology, 291, 108 023, https://doi.org/10.1016/j.agrformet.2020.108023, 2020.

Seibert, J.: Estimation of parameter uncertainty in the HBV model: Paper presented at the Nordic Hydrological Conference (Akureyri, Iceland-August 1996), Hydrology Research, 28, 247–262, 1997.

Shin, M. J. and Kim, C. S.: Assessment of the suitability of rainfall–runoff models by coupling performance statistics and sensitivity analysis, Hydrology Research, 48, 1192–1213, https://doi.org/10.2166/nh.2016.129, 2016.

Silverman, B. W.: Density Estimation for Statistics and Data Analysis, Chapman and Hall, 1986.

Singh, S. K., Ibbitt, R., Srinivasan, M., and Shankar, U.: Inter-comparison of experimental catchment data and hydrological modelling, Journal of Hydrology, 550, 1–11, https://doi.org/10.1016/j.jhydrol.2017.04.049, 2017.

Slater, L. J., Thirel, G., Harrigan, S., Delaigue, O., Hurley, A., Khouakhi, A., Prosdocimi, I., Vitolo, C., and Smith, K.: Using R in hydrology: a review of recent developments and future directions, Hydrology and Earth System Sciences, 23, 2939–2963, https://doi.org/10.5194/hess-23-2939-2019, 2019.

Sleziak, P.: Vývoj webovej aplikácie pre potreby hydrologického modelovania, Master's thesis, Vysoká škola báňská-Technická univerzita Ostrava, 68 pages, 2019.

Soetaert, K. and Petzoldt, T.: Inverse Modelling, Sensitivity and Monte Carlo Analysis in R Using Package FME, Journal of Statistical Software, 33, 1–28, https://doi.org/10.18637/jss.v033.i03, 2010.

Souza, R.: Ecohydmod: Ecohydrological Modelling, https://CRAN.R-project.org/package=Ecohydmod, R package version 1.0.0, 2017.

Staudinger, M., Stahl, K., Seibert, J., Clark, M. P., and Tallaksen, L. M.: Comparison of hydrological model structures based on recession and low flow simulations, Hydrology and Earth System Sciences, 15, 3447–3459, https://doi.org/10.5194/hess-15-3447-2011, 2011.

Storn, R. and Price, K.: Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization, 11, 341–359, https://doi.org/10.1023/A:1008202821328, 1997.

Stroustrup, B.: The C++ programming language: reference manual, Tech. rep., Bell Lab., 1984.

Taner, M. U.: sacsmaR: SAC-SMA Hydrology Model, R package version 0.0.1, 2019.

Todini, E.: History and perspectives of hydrological catchment modelling, Hydrology Research, 42, 73–85, https://doi.org/10.2166/nh.2011.096, 2011.

Toum, E.: HBV.IANIGLA: Decoupled Hydrological Model for Research and Education Purposes, https://CRAN.R-project.org/package=HBV.IANIGLA, R package version 0.1.1, 2019.

Valéry, A., Andréassian, V., and Perrin, C.: "As simple as possible but not simpler": What is useful in a temperature-based snow-accounting routine? Part 2 – Sensitivity analysis of the CemaNeige snow accounting routine on 380 catchments, Journal of Hydrology, 517, 1176–1187, https://doi.org/10.1016/j.jhydrol.2014.04.058, 2014.

Venables, B., Hornik, K., and Maechler, M.: polynom: A Collection of Functions to Implement a Class for Univariate Polynomial Manipulations, https://CRAN.R-project.org/package=polynom, R package version 1.4-0, 2019.

Vidal, J.-P., Martin, E., Franchistéguy, L., Baillon, M., and Soubeyroux, J.-M.: A 50-year high-resolution atmospheric reanalysis over France with the Safran system, International Journal of Climatology, 30, 1627–1644, https://doi.org/10.1002/joc.2003, 2010.

Viglione, A. and Parajka, J.: TUWmodel: Lumped Hydrological Model for Education Purposes, https://CRAN.R-project.org/package=TUWmodel, R package version 1.1-1, 2020.

Viglione, A., Borga, M., Balabanis, P., and Blöschl, G.: Barriers to the exchange of hydrometeorological data in Europe: Results from a survey and implications for data policy, Journal of Hydrology, 394, 63–77, https://doi.org/10.1016/j.jhydrol.2010.03.023, 2010.

Vitolo, C., Fry, M., and Buytaert, W.: rnrfa: an R package to retrieve, filter and visualize data from the UK National River Flow Archive, The R Journal, 8, 102–116, https://journal.r-project.org/archive/2016/RJ-2016-036/index.html, 2016a.

Vitolo, C., Wells, P., Dobias, M., and Buytaert, W.: fuse: An R package for ensemble Hydrological Modelling, The Journal of Open Source Software, 1, https://doi.org/10.21105/joss.00052, 2016b.

Vitolo, C., Fry, M., Buytaert, W., Spencer, M., and Gauster, T.: rnrfa: an R package to retrieve, filter and visualize data from the UK National River Flow Archive, R package version 2.0.3, 2018.

Vrugt, J. A. and Beven, K. J.: Embracing equifinality with efficiency: Limits of Acceptability sampling using the DREAM(LOA) algorithm, Journal of Hydrology, 559, 954–971, https://doi.org/j.jhydrol.2018.02.026, 2018.

Wagener, T., Sivapalan, M., Troch, P. A., McGlynn, B. L., Harman, C. J., Gupta, H. V., Kumar, P., Rao, P. S. C., Basu, N. B., and Wilson, J. S.: The future of hydrology: An evolving science for a changing world, Water Resources Research, 46, https://doi.org/10.1029/2009WR008906, 2010.

Wrede, S., Fenicia, F., Martínez-Carreras, N., Juilleret, J., Hissler, C., Krein, A., Savenije, H. H. G., Uhlenbrook, S., Kavetski, D., and Pfister, L.: Towards more systematic perceptual model development: a case study using 3 Luxembourgish catchments, Hydrological Processes, 29, 2731–2750, https://doi.org/10.1002/hyp.10393, 2015.

Zambrano-Bigiarini, M.: hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series, https://doi.org/10.5281/zenodo.839854, https://github.com/hzambran/hydroGOF, R package version 0.4-0, 2020.

Zeileis, A. and Grothendieck, G.: zoo: S3 Infrastructure for Regular and Irregular Time Series, Journal of Statistical Software, 14, 1–27, https://doi.org/10.18637/jss.v014.i06, 2005.

Zipper, S., Albers, S., and Prosdocimi, I.: CRAN Task View: Hydrological Data and Modeling, https://cran.r-project.org/view=Hydrology, 2019.

1360

1365

1370