airGRteaching: an open-source tool for teaching hydrological modeling with R

Olivier Delaigue[1,*], Pierre Brigode[1,2,*], Guillaume Thirel[1], and Laurent Coron[3]

[1]Université Paris-Saclay, INRAE, HYCAR, Antony, France
[2]Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, OCA, IRD, Géoazur, Géoazur, Sophia-Antipolis, France
[3]EDF – PMC Hydrometeorological Center, Toulouse, France
[*]These authors contributed equally to this work.

Correspondence: Olivier Delaigue (olivier.delaigue@inrae.fr)

Abstract. Hydrological modeling is at the core of most studies related to water, especially for anticipating disasters, managing water resources, and planning adaptation strategies. Consequently, teaching hydrological modeling is an important, but difficult, matter. Teaching hydrological modeling requires appropriate software and teaching material (exercises, projects); however, although many hydrological modeling tools exist today, only few are adapted to teaching purposes. In this article, we present the `airGRteaching` package, which is an open-source R package. The hydrological models that can be used in `airGRteaching` are the GR rainfall-runoff models, i.e. lumped processed-based models allowing to simulate streamflows, including the GR4J model. In this package, thanks to a graphical user interface and a limited number of functions, numerous hydrological modeling exercises representing a wide range of hydrological applications are proposed. To ease its use by students and teachers, the package contains several vignettes describing complete projects that can be proposed to investigate various topics such as streamflow reconstruction, hydrological forecasting, and assessment of climate change impact.

1 Introduction

In order to anticipate and manage water conditions, outcomes of hydrological research are applied on a regular basis by water managers and stakeholders. These are aimed at addressing numerous challenges, such as:

- water resources management for hydropower, irrigation, and drinking water (e.g., Neumann et al., 2018);

- low-flow forecasting, to better manage water resources and to ensure that environmental flows are respected (e.g., Nicolle et al., 2014);

- flood forecasting, to protect people and property, to evacuate inhabitants, and to plan the allocation of rescue forces with sufficient anticipation (e.g., Furusho et al., 2016);

- flood protection, to define areas that cannot be built or to design dikes or dams (e.g., Paquet et al., 2013);

- assessing climate change impact, to better anticipate future risks and design adaptation measures (e.g., Dorchies et al., 2014);

- assessing water resources in catchments (e.g., Brigode et al., 2019).

- testing hypotheses about catchment processes since all fluxes are not easily measurable (Clark et al., 2011).

The consequences and damage of extreme events (floods and droughts) are more limited when such events are better anticipated
or managed. Hydrological science can also help to optimize profits in the hydropower sector (Cassagnole et al., 2021). In this
context, hydrological models are key tools because they help to transform meteorological variables into hydrological variables.

1.1 On the need (and relevance of) for teaching hydrology using models

For many years, teaching hydrology has implied teaching hydrological modeling (Wagener and McIntyre, 2007). As a conse-
quence, teaching hydrology can also imply programming, thereby raising the important issues of automatic calibration, sen-
sitivity analysis (AghaKouchak and Habib, 2010; Knoben and Spieler, 2022), and also reproducibility in hydrology (Hutton
et al., 2016). Given the advantages of applying hydrological models for the real-life cases listed above, there is a considerable
interest in and need for models to teach hydrology. First, hydrological modeling is a daily task for numerous practitioners and
thus it is an art that needs to be understood and mastered by students. Moreover, models are key tools for understanding the
hydrological cycle, the interactions between the processes involved, and how hydrological variables evolve. Lastly, models
represent an efficient way of proposing "active learning" courses to students. Thus, the impact of using hydrological models
with students while they are learning can be significant. Sanchez et al. (2016) showed that the use of a simple spreadsheet
with real hydrological data had a significant and positive impact on the civil engineering curriculum. AghaKouchak and Habib
(2010) also found significant learning gains for students using modeling tools in class. Nevertheless, the added value of using
models in class is not automatic and straightforward. For example, Marshall et al. (2015) demonstrated that the same hydrolog-
ical course offered using either (i) Microsoft Excel (Microsoft Corporation, 2019), (ii) MATLAB (2018) or (iii) the COMSOL
Multiphysics software (Zimmerman, 2006)[1] made no significant difference in student performances. This result highlights the
need to use tools tailored for teaching hydrology with models.

1.2 On the need for common tools for teaching hydrological (reproducible) modeling

Wagener and McIntyre (2007) and Merwade and Ruddell (2012) highlighted the large diversity of approaches available to
teach hydrology. Hutton et al. (2016) argued for the need for reproducible computational hydrology, to teach version-controlled
programming: "A key step to change this culture is to ensure that computational science training (e.g., http://software-carpentry.
org) is properly embedded within hydrological science curriculums, so that future generations of hydrologists have the skills to
build readable, version controlled and unit-tested software (McConnell, 2004), allowing them to engage more fully in an open
scientific community by reproducing and reusing each other's research outputs." This moves toward reproducible hydrology
(Hall et al., 2022) and leads to the emergence of experiments of virtual laboratories (Ceola et al., 2015; Tarboton et al., 2014),
open-source software (Coron et al., 2017; Slater et al., 2019), and open datasets (Addor et al., 2017; Irving et al., 2018). What
about open hydrological teaching?

---

[1]https://www.comsol.com/

1.3 A review of modeling tools designed for teaching hydrological modeling

The development of modeling tools dedicated to teaching hydrology began in the 1960s, with the pedagogic hydrological model `ABC` (Fiering, 1967; Kay et al., 1982; Burt and Butcher, 1986; Kirkby and Naden, 1988). Since the development of ABC, several software programs have been designed for teaching hydrology (see Special Issue in HESS entitled "Hydrology education in a changing world", Seibert et al., 2013). Elshorbagy (2005) used a system dynamics approach based on the `STELLA` visual programming language (Richmond et al., 1985) for teaching watershed hydrology. Pérez-Sánchez et al. (2022) described the use of Microsoft Excel (Microsoft Corporation, 2019) spreadsheets for teaching hydrological modeling and for estimating climate change impacts in a postgraduate civil engineering master's degree. The `HBV` rainfall-runoff model has been used several times as a basis to develop an education-dedicated version: AghaKouchak and Habib (2010) and AghaKouchak et al. (2013) developed the `HBV-EDU` toolbox in MATLAB to teach hydrology and uncertainty estimation[2], while Seibert and Vis (2012) created the `HBV-light` software. Mendez and Calvo-Valverde (2016) and Viglione and Parajka (2020) developed, respectively, `HBV-TEC` and `TUWmodel` within the R programming language (R Core Team, 2017), and several web applications designed for using HBV are available online[3]. This approach of simplifying an existing hydrological model for teaching purposes has been applied with `HBV` but also with other models such as `VIC` by Wi et al. (2017) with `VIC-ASSIST` (developed in MATLAB). The MATLAB-based `HMETS` model (Martel et al., 2017)[4], initially developed for teaching, has proved to be efficient over a large sample of 320 catchments located in the contiguous United States.

Numerous solutions exist to teach hydrological modeling, but they all have their limitations (see Carriba Demange et al., 2022), such as being a "light version" of a model (e.g., `HBV-light`), inability to import one's own data (e.g., `TUWteaching`[5]), inability to access and modify the source code (e.g., `RS MINERVE`; García Hernández et al., 2019), inability to manually or automatically calibrate the model parameters (e.g., `HBV.IANIGLA`; Toum et al., 2021), or being based on proprietary programming language (e.g., `VIC-ASSIST` developed in MATLAB).

1.4 R, a language increasingly used by hydrologists, especially for modeling...

The open-source programming language R is one of the most widely used languages in the hydrological community. It offers many open-source libraries useful, for example, for retrieving hydro-meteorological data, performing spatial analysis, analyzing hydrological statistics, etc. The whole workflow undertaken in hydrological studies can be done with R (see Slater et al., 2019), which is very useful for practical reasons. The reader is asked to refer to Slater et al. (2019) for further details about the advantages of R for all the steps of the workflow and to the R Hydrology TaskView[6] for a complete list of R packages linked to hydrology. The choice of hydrological modeling R packages is particularly large (see Astagneau et al., 2021, for a recent review), providing a variety of solutions adapted to the diverse problems or case studies that can be encoutered. Here again,

---

[2]https://fr.mathworks.com/matlabcentral/fileexchange/41395-hbv-edu-hydrologic-model?s_tid=FX_rc1_behav

[3]e.g., https://github.com/NikoZHAI/lumphydro

[4]https://fr.mathworks.com/matlabcentral/fileexchange/48069-hmets-hydrological-model?s_tid=FX_rc1_behav

[5]https://webaapptuwmodel.shinyapps.io/TUWteaching/

[6]https://cran.r-project.org/web/views/Hydrology.html

the reader is referred to Astagneau et al. (2021) for further details about the packages and models available. In addition, R facilitates interdisciplinary work in the other fields of geosciences in which R is also used (e.g., Bezak et al. (2019) who use the `airGR` package (Coron et al., 2017, 2022) for hydrological modeling and the prediction of landslides). One of the strengths of
85 R is its ability to incorporate geographic data and spatial analysis, such as in the use of the MODIS dataset, for example, for modeling of snow accumulation and melt (Riboust et al., 2019).

1.5 ... but not yet for teaching, even if attempts are being made

A basic search with the keywords "educ*" and "teach*" (last check on 1 August 2022) in the R Hydrology Task View (Zipper et al., 2020) only returns a couple of packages that address teaching aspects of hydrology: `TUWmodel` (Viglione and Parajka,
90 2020), which contains a hydrological model that is proposed for educational purposes, but does not contain actual exercises or an interface; and `EcoHydRology` (Fuka et al., 2018) which is aimed at providing a flexible framework for hydrology-related staff, students or researchers for basic exercises and `airGRteaching` (Delaigue et al., 2018, 2023b), which is the topic of the present article.

`airGRteaching` relies on the widely used GR hydrological models, a family of rainfall-runoff models simulating stream-
95 flows that are usually used in lumped mode (i.e., running at the basin scale with aggregated input), and that were recently incorporated into an R package (`airGR`; Coron et al., 2017, 2022). To provide teaching material, the `airGR` developers set up an add-on package dedicated to teaching hydrology, named "`airGRteaching`". This package contains a graphical user interface, simple functions, and hydrology exercises. Since then it has been used for teaching and for hands-on projects in various universities and engineering schools (see, for instance a master's degree project using `airGRteaching`: Roux and
100 Brigode, 2018).

Since `airGRteaching` relies on the widely-used GR models (see e.g., Perrin et al., 2003, which presents the widely-known GR4J model), and on the `airGR` package, which gained lots of interest over the past few years (see Coron et al., 2017, which presents the `airGR` package, or the list of publications on the airGR website[7], which lists all known uses of or references to `airGR`), we believe that this tool can pave the way to developing new hydrology teaching initiatives, developing
105 similar tools and to promote hydrology to a more general audience.

In this paper, after introducing the general concepts taught in hydrology, we present the main features of the `airGRteaching` package and introduce several exercises using this package.

2 Description of `airGRteaching`

2.1 The rationale behind `airGRteaching`: a glance backward

110 The GR models were initially developed in the 1980's by Claude Michel and his colleagues at Cemagref (that recently became Irstea and then INRAE). The main objective was to design efficient models, starting from a simple structure and gradually

---

[7]https://hydrogr.github.io/airGR/page_publications.html#Use_and_mention_of_airGR

adding complexity that proved useful for improving the model's predictive power (Michel, 1983). This approach prioritized predictive power over explanatory models (Shmueli, 2010), finding justification for this from results obtained using large data sets and not from predefined concepts. This led to the development of a family of models that are usually used in lumped mode (i.e., running at the basin scale with aggregated input).

To disseminate their models beyond the Fortran programming community, the developers of the GR models a long time ago proposed Microsoft Excel spreadsheets containing hydrological models, namely, the GR1A, GR2M and GR4J models, as well as the CemaNeige snow accumulation and melt model (see next section for a description of these models), accompanied by a dummy dataset[8]. The rationale behind this approach was dual: easily providing the GR models to external users (researchers and consultants from France and abroad) and illustrating the hydrological concepts to students with the models developed in-house. The relatively high efficiency and low computationnal time requirements of these models made them easily runnable with Microsoft Excel. In addition, the use of Excel macros enabled interactivity (e.g., the possibility to automatically update simulations when parameter values are modified by users) and graphs were predefined.

Later, the `airGR` R package was developed to propose additional GR models, and the `airGRteaching` R package was built as an add-on package of `airGR`. These tools are described in the next sections.

2.2 The GR models and the **airGR** package

To ease the implementation of the GR models, Coron et al. (2017, 2022) proposed the `airGR` package. Gathering seven hydrological models and one snow accumulation and melt model, `airGR` can be seen as a research tool, as an efficient way for its developers to share research results, and as a tool simple enough to be used by water managers. The hydrological models included in `airGR` differ in their complexity and time step, with a gradual increase in complexity as the time step decreases, and various application objectives:

- GR1A (Mouelhi, 2003; Mouelhi et al., 2006a): annual one-parameter model, used for water resources assessment (Baahmed et al., 2015; Kouassi et al., 2012). It consists of a single equation relating the annual streamflow to antecedent annual precipitation and potential evapotranspiration.

- GR2M (Mouelhi, 2003; Mouelhi et al., 2006b): monthly two-parameter model, used for water resources assessment and water regime modeling (Belarbi et al., 2017; Marchane et al., 2017). It consists of two stores: a production store used for calculating the part of rainfall transformed into discharge (effective rainfall) and a routing store used for distributing in time the effective rainfall toward the catchment outlet.

- GR4J (Perrin et al., 2003): daily four-parameter model, used for water resources assessment, floods and droughts simulation and forecasting and climate change impact (Chauveau et al., 2013). In addition to the GR2M components, it contains two unit hydrographs that refine the temporal distribution of effective rainfall.

---

[8]https://gitlab.irstea.fr/HYCAR-Hydro/ExcelGR

- GR5J (Le Moine, 2008): daily five-parameter model, used for similar applications as GR4J. Compared to GR4J, GR5J contains only one unit hydrograph and the intercatchment groundwater exchange function is slightly more general by two-way exchange fluxes between surface and regional groundwater.

- GR6J (Pushpalatha et al., 2011): daily six-parameter model, used for similar applications to GR4J and GR5J. Compared to GR5J, an additional exponential store improves the representation of low flows.

- GR4H (Mathevet, 2005): hourly four-parameter model, used for flood forecasting (Desclaux et al., 2018). Its structure is almost identical to that of GR4J.

- GR5H (Ficchì et al., 2019): hourly five-parameter model, mostly based on the GR5J model structure.

- CemaNeige (Valéry et al., 2014): daily two-parameter snow accumulation and melt model, used for snowy catchments. It consists of (i) a partition of precipitation into rainfall and snowfall upgraded with an extrapolation based on altitudinal gradients, (ii) a snow store that also represents the snow heat content, and (iii) a melt function. Optionally, satellite snow data can be used to calibrate an improved version of CemaNeige representing the Snow Water Equivalent - Snow Cover Area hysteresis relationship (Riboust et al., 2019).

- Semi-distribution: in order to represent spatially-heterogeneous catchments, the aforementioned models (except GR1A), which are lumped, can be used in a semi-distributed mode. The streamflow simulated for upstream catchments is propagated downstream using a lag function (de Lavenne et al., 2019).

2.3 The `airGRteaching` perspective

`airGRteaching` embeds the main features of `airGR` and offers simplified ergonomics. It therefore uses its basic tools, meaning that all models implemented in `airGR` are available in `airGRteaching`. Since these models have relatively simple structures and few parameters, they can be more easily understood by novice users such as students. `airGRteaching` does not provide "simplified" versions of existing GR models. Thus, students are able to learn hydrological modeling from the same models that are used in practice, not from degraded versions.

To ease hand-on experience, the choice was made to reduce the number of functions to implement a complete modeling exercise (an `airGRteaching` function therefore embeds several `airGR` functions). In addition, the number of modeling options has been reduced, which limits the number of arguments to be specified for running a simulation and simplifies the associated documentation. All these choices allow users to focus on the main questions that beginners ask themselves when they start dealing with hydrological modeling.

2.4 `airGRteaching` features

`airGRteaching` contains only a few functions, which can be split into two groups:

1. a small set of functions to prepare data, to calibrate and run hydrological models and to plot outputs, i.e. the basic functions needed to undertake a hydrological modeling study;

2. a function to launch a graphical user interface (GUI) to set up the hydrological models manually.

These two levels of use allow teachers to choose between different levels of technical difficulty. They can choose the most adapted use according to the time available for the exercises, the teaching objectives, and the students' skills.

To get started with the package, particular attention was given to the documentation. The user manual describes precisely and succinctly the implementation of functions, and provides simple examples (https://cran.r-project.org/web/packages/airGRteaching/airGRteaching.pdf). In addition, a website was created to explain step by step how to use the different features and to answer frequently asked questions (https://hydrogr.github.io/airGRteaching/).

Table 1 summarizes the `airGRteaching` (and `airGR`) features.

2.4.1 Basic functions for undertaking a hydrological modeling study

The main steps required to undertake a hydrological modeling study can be performed with `airGRteaching` with the help of a few simple functions:

– a data preparation function, `PrepGR()`: with only three main arguments, namely, the hydrometeorological input data as a data frame or independent vector time series, the name of the rainfall-runoff model to run, and a boolean indicating whether the `CemaNeige` snow model is activated, this function prepares all the necessary inputs in the correct format for the `airGRteaching` functions. If CemaNeige is activated, additional arguments are needed (e.g., catchment elevation distribution);

– a calibration function, `CalGR()`: with three main arguments, namely, the object produced by `PrepGR()`, the objective function name (i.e., which criterion is used to optimize the parameter values), and the calibration period start and end, this function calibrates the chosen GR model. If desired, a transformation of discharge can be chosen for the objective function calculation in order to give more weight to certain ranges of discharges (Santos et al., 2018) and a warm-up period can also be defined;

– a simulation function, `SimGR()`: with four main arguments, namely, the object produced by `PrepGR()`, the parameter values (output of `CalGR()` or defined by the user), the name of an efficiency criterion used to evaluate the simulation, and the simulation period start and end, this function runs the chosen GR model and assesses its performance. If desired, a transformation can be used for the criterion calculation and a warm-up period can be defined;

– static (`plot()`) and dynamic (`dyplot()`, which uses the functionalities of the `dygraphs` package Vanderkam et al., 2018) graphical functions: these functions take as main arguments any of the objects produced by `PrepGR()`, `CalGR()` and `SimGR()` (to be chosen). Graph tuning arguments are available, but optional. The dynamic graphs show the observed and simulated discharge time series. The static graphs render a choice of graphs to be selected with the `which` argument.

Many graphical outputs are available (see appendices A and B). Figure A1 provides a general overview of the precipitation and streamflow records to identify possible outliers and periods with missing data. Figure A2 adds to the previous graph the

Table 1. `airGR` and `airGRteaching` features.

	airGRteaching (GUI)	airGRteaching (code)	airGR
Datasets			
Example data	yes	yes	yes
User data	yes	yes	yes
Working environment			
Graphical user interface	yes	no	no
Use of programming	yes (1 command)	yes (simplified)	yes (advanced)
Dynamic graphics outputs	yes	yes	no
Static graphic outputs	yes	yes	yes
Models			
Hourly GR models (+ CemaNeige)	no	yes	yes
Daily GR models (+ CemaNeige)	yes	yes	yes
Monthly GR models	yes	yes	yes
Yearly GR models	no	yes	yes
Semi-distributed version of models	no	no	yes
CemaNeige with hysteresis using SCA & SWE	no	no	yes
Warm-up period disabling	no	yes	yes
Choice of initialization of internal states	no	no	yes
Criteria and calibration			
NSE criterion	yes	yes	yes
KGE criterion	yes	yes	yes
KGE' criterion	no	yes	yes
RMSE criterion	no	yes	yes
Composite criteria (defined by the user)	no	no	yes
Calculation of criteria over discontinuous periods	no	no	yes
Full freedom of parameter value ranges	no	yes	yes
Adaptation of the calibration options	no	yes (simplified)	yes (advanced)
Other calibration algorithms (defined by the user)	no	no	yes

205   simulated streamflow in order to have an overall view of the calibrated model behavior, and provides graphical diagnostic tools to check whether the simulated streamflow hydrograph fits the observed streamflow hydrograph. Figure A4 focuses on time series graphs (available in Fig. A2), and adds the potential evapotranspiration. Figure A3 focuses on the errors of the model compared to the observed streamflows. Figure A5 helps to understand the concept of parameter optimization by displaying the tested parameter values and the correspondence with the value of the criterion chosen as objective function. In general,

dynamic graphs (Figures B1, B2) help to read more precisely the values of time series and to zoom in on a particular event for each of the two axes (some options are available, e.g., to add a rolling average or a time range selector).

2.4.2 The graphical user interface

Using the functionalities of the `shiny` package (Chang et al., 2022), the `airGRteaching` graphical user interface (GUI) called with the `ShinyGR()` function allows one to use the GR models with no programming skills at all thanks to an intuitive interface. The `ShinyGR()` function takes as arguments hydrometeorological data and the simulation period start and end. Additional arguments can be provided if snow is present. Data can be provided for several catchments and the function offers the possibility to use different themes for the interface. The GR and CemaNeige models and the objective function are to be selected within the interface.

Figure 1 presents a commented example of the interface. Several intuitive elements can be found. On the left side:

- "Choose a dataset" to select a dataset from those provided by the user to the function;

- "Choose a model" to select a model among the GR2M monthly model and the GR4J, GR5J, and GR6J daily models according to the time step of the datasets provided (models at other time steps are not included in the GUI) and to activate the CemaNeige snow accumulation and melt model for the daily models;

- "Parameters values" using sliders to modify the parameters of the models. The parameters proposed are automatically adapted to the chosen model and the ranges are predefined. Changing any parameter value causes a real-time update of the plots and displayed scores (see below);

- "Automatic calibration" to perform an automatic calibration by optimizing a chosen objective function (among NSE and KGE, and with a squared root, inverse or no transformation of discharge).

At the top:

- "Choose a plot" to change the kind of plot that is displayed (cf. Fig. 2). Users can choose:

  - "Flow time series": dynamic plots of observed and simulated discharge as well as precipitation time series and discharge errors,

  - "Model performance": an ensemble of static plots of observed and simulated discharge as well as precipitation time series, and of annual regimes, flow duration curves, and a scatter plot between simulated and observed discharges,

  - "State variables": dynamic plots that show the time series of GR model store levels as well as the time series of internal model fluxes,

  - "Model diagram": a plot that can be dynamic and shows on the right the scheme of the chosen GR model and the dynamic evolution of all its fluxes with time and the related hydrometeorological data;

- "Select the time window" to zoom within the provided data period or to move slide the window;

240   – "Select the target date" to target a specific date (only for the "Model diagram" panel).

  In the center:

   – The plots proposed by the "Choose a plot" panel.

  On the right:

   – A table of criteria provides the values of seven performance criteria (NSE and KGE with use of squared root, inverse or
245   no transformation of discharge, in addition to the bias);

   – "Show previous simulation Qold": the previously obtained simulated time series appear on the plots provided in the
    center of the GUI as a dotted gray line. In addition, ticking this option makes criteria of this previous simulation appear
    in the criteria table introduced above. This option has no effect in the "Model performance" panel.

   – Two buttons allow users to download the displayed plot in a PNG file format, which can be useful for a report for
250   example (in order to ensure the tracking of the downloaded files, various information is automatically added to the
    file header: name of the dataset, name of the model, simulation period, parameter values, see Appendix C), and the
    hydrometeorological data (including the simulation) in a CSV file format, to be used externally for further analysis or to
    be saved.

  Figure 2 presents the airGRteaching GUI "Modeling" panels (a-d) and "Summary sheet" panels (e-f).
255 If R is not installed on the students' computers, it is possible to run the `airGRteaching` GUI online. Indeed, the graphical
user interface is available on the following website: https://sunshine.inrae.fr/app/airGRteaching with demo datasets.

2.4.3 Data associated with `airGRteaching`

Users are free to use their own datasets, but the `airGRteaching` package benefits from the `airGRdatasets` package
(Delaigue et al., 2023a), which contains a dataset of 19 different catchments located in France (Fig. 3 and Table 2). This
260 dataset is a subset of the larger CAMELS-FR dataset (Delaigue et al., 2022), and has been assembled to include various
French hydro-climatic regimes, with 12 rain-dominated catchments, one rain- and snow-dominated catchment, two snow-
dominated catchments, two Mediterranean catchments and two groundwater-dominated catchments. Figure 4 shows the main
characteristics of the catchment set. Catchment area ranges from 25 to 3917 km^2, with half of the catchment set draining less
than 686 km^2.

265 The dataset is composed of both static geomatic and physiographics catchment indices and hydro-climatic time series (solid
and liquid precipitation, potential evapotranspiration, air temperature, and streamflow time series). The climatic time series have
been extracted from the SAFRAN reanalysis (Vidal et al., 2010) and aggregated at the catchment scale, while streamflow series
have been extracted using the HydroPortail (https://hydro.eaufrance.fr/). These hydro-climatic temporal series are available at
the daily time step.
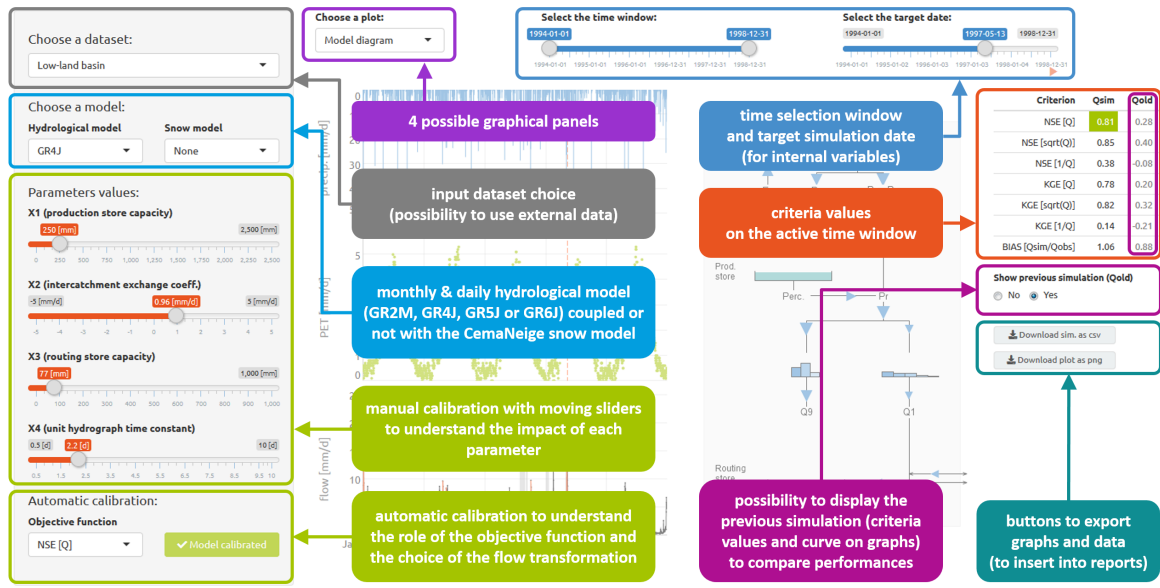
Figure 1. Overview of the `airGRteaching` GUI and identification of its main elements.

3 Teaching hydrology with `airGRteaching`

This section and the accompanying appendices present tests based on the `airGRteaching` package and designed to illustrate rainfall-runoff modeling, model calibration, evaluation, and robustness in hydrological classes. These tests are also available as a vignette in the `airGRteaching` package: users can thus recreate all these illustrations using their own datasets.

3.1 Understanding rainfall-runoff modeling

3.1.1 The role of model components and parameters

Rainfall-runoff models are composed of different components, e.g., reservoirs or unit hydrographs, whose behavior is defined by equations and parameters. Parameter estimation is a key step toward tailoring the models to a specific catchment. Understanding the role of model components and parameters is therefore an unavoidable preliminary step to performing hydrological modeling.

To illustrate the production and the routing parts of hydrological modeling that are present in any model, it is possible to use the different GR models included in `airGRteaching` and to produce rainfall-runoff transformations considering different model parameter values.

The GR4J model (Perrin et al., 2003) comprises a production store (X1 parameter), which determines the actual evapotranspiration and the net rainfall (see Appendix C4 for a GR4J flow chart). The routing of net rainfall is determined through

Table 2. List of the 19 catchments in France included in the `airGRdatasets` package (ID: identification letters displayed in Fig. 3).

	Station code	ID	Station Name	Area (km^2)	Hydrological regime
1	A273011002	A	the Bruche at Russ [Wisches]	224	Pluvial
2	A605102001	B	the Meurthe at Saint-Dié-des-Vosges	371	Pluvial
3	B222001001	C	the Meuse at Saint-Mihiel	2543	Pluvial
4	E540031001	D	the Canche at Brimeux	917	Groundwater
5	E645651001	E	the Nièvre at Étoile	270	Groundwater
6	F439000101	F	the Loing at Épisy	3917	Pluvial
7	H010002001	G	the Seine at Plaines-Saint-Lange	686	Pluvial
8	H120101001	H	the Aube at Bar-sur-Aube	1298	Pluvial
9	H622101001	I	the Aisne at Givry	2888	Pluvial
10	J171171001	J	the Trieux at Saint-Péver - Pont Locminé	184	Pluvial
11	J421191001	K	the Odet at Ergué-Gabéric - Treodet	203	Pluvial
12	K134181001	L	the Arroux at Rigny-sur-Arroux	2271	Pluvial
13	K265401001	M	the Couze Pavin at Saint-Floret	216	Pluvial
14	K731261001	N	the Indre at Saint-Cyran-du-Jambot	1707	Pluvial
15	V123521001	O	the Ire at Doussard	25	Nival-Pluvial
16	X031001001	P	the Durance at Embrun [La Clapière] - DREAL PACA	2283	Nival
17	X045401001	Q	the Ubaye at Lauzet-Ubaye [Roche-Rousse] - DREAL PACA	943	Nival
18	Y643401001	R	the Esteron at Broc [La Clave]	442	Mediterranean
19	Y862000101	S	the Taravo at Zigliara [Pont d'Abra]	332	Mediterranean

two unit hydrographs (X4 parameter) and a routing store (X3 parameter). A final component, representing the intercatchment groundwater exchange, is determined by the X2 parameter.

As an example, the command lines in Appendix D1 and Fig. 5 illustrate the role of the X2 parameter in the production part of the rainfall-runoff transformation, showing higher streamflow values simulated with higher X2 values, since higher X2 parameter values lead to more positive incoming water from groundwater. Moreover, Fig. 6 illustrates the role of the X4 parameter in the routing part of the rainfall-runoff transformation, with delayed flood peak values when considering higher X4 values (cf. command lines in Appendix D2).

The relative importance of the production and routing functions depends on the time step considered for the rainfall-runoff simulation. The production process is more important for the larger time steps (e.g., month or year) since it controls the catchment water balance. This can be easily illustrated by aggregating simulations performed at a daily time step and at a yearly time step (cf. command lines in Appendix D3). Figure 7 compares, at the annual time step, the GR4J daily simulations performed using different X2 parameter values with the simulations performed using different X4 parameter sets. We can observe that at the annual time step, the impact of considering different X4 parameter values is limited compared to the use of different X2 parameter values.

3.1.2 On the need to perform a model warm-up

300 Initial values of the model water storages must be specified at the beginning of a simulation. The way initial levels are defined can lead to potentially significant model errors. The most convenient way for modelers to initialize rainfall-runoff models is to perform a warm-up run of the model in order to limit the impact of this unknown.

This issue can be illustrated with `airGRteaching` by considering different warm-up period lengths (cf. command lines in Appendix D4). Figure 8 illustrates a portion of the streamflow simulations obtained considering (i) no warm-up period, (ii) a 1-month warm-up period, and (iii) a 1-year warm-up period of the two GR4J stores. Figure 8 shows that the three simulations converge after a bit more than 5 months, reinforcing the necessity of performing a sufficiently long warm-up. Please note that by default, `airGRteaching` initializes the production and the routing stores at 30 % and 50 % of their capacity, respectively.

3.2 Model calibration, evaluation, and robustness

3.2.1 Manual calibration

310 In the `airGRteaching` GUI (see Fig. 2), it is possible to test different parameter sets of the GR rainfall-runoff models and to estimate the performance of each tested parameter sets in order to perform a manual calibration. A classic way to do so through the `airGRteaching` GUI is to select a criterion as an objective function in the table showing the criteria values on the right, to activate the "Show previous simulations (Qold)", and to modify parameter values step by step until the simulation and criterion are satisfying.

3.2.2 Automatic calibration

Automatic calibration of model parameters is also possible in `airGRteaching` using the procedure described by Michel (1991) and by considering one objective function such as NSE (Nash and Sutcliffe, 1970) or KGE (Gupta et al., 2009). To do so, there are two options in `airGRteaching`:

1. clicking on the automatic calibration button in the `airGRteaching` GUI.

320 2. using the simple `airGRteaching` command line functions (`PrepGR()` and `CalGR()`, cf. command lines in Appendix D6).

The calibration algorithm available in `airGRteaching` comes from `airGR`, and is described into further details in (Coron et al., 2017, section 2.3). Two distinct steps are included in the procedure:

1. a systematic inspection of the parameter space is performed to determine the most likely zone of convergence. This is done either by direct grid-screening or by constrained sampling based on empirical parameter databases;

2. from the best parameter set of the previous step, a steepest descent local search procedure is carried out to find an estimate of the optimum parameter set.

`airGRteaching` allows to visualise the second step of this procedure (cf. command lines in Appendix A5).

3.2.3 How to evaluate model calibration?

330 Different ways to evaluate the model calibration performance may be conceived using `airGRteaching`: evaluating criteria on the calibration period, examining the graphical summary of the calibration performance (`airGR::plot()`), comparing simulated and observed streamflow temporal series, etc.

Analyzing simulated versus observed flow regimes is an informative indicator of model performance (cf. command lines in Appendix D7). Figure 9 compares regimes in a mountainous catchment (located in the French Alps), while the flow simulation 335 has been obtained with and without taking into account snow accumulation and melt. The regime comparison might be compelling for the students, hopefully leading them to use an additional snow accumulation and melt routine (such as CemaNeige (Valéry et al., 2014), available in `airGRteaching`).

3.2.4 Objective functions for model calibration

Oudin et al. (2006) and other authors showed the impact of using flow transformation in objective functions used for model 340 calibration. It is possible, in `airGRteaching`, to apply different flow transformations to the objective function used for model parameter calibration (cf. command lines in Appendix D8). Figure 10 compares the simulations performed considering GR4J parameter sets obtained after a calibration on (i) NSE calculated on natural flows (noted as NSE_Q hereafter), (ii) NSE calculated on square root transformed flows (noted as $NSE_{\sqrt{Q}}$ hereafter) and (iii) NSE calculated on logarithmic transformed flows (noted as $NSE_{\log Q}$ hereafter), emphasizing performance in high, mean and low flows, respectively. Logically, we can 345 observe that the model calibrated on NSE_Q performs better for high-flow periods, the model calibrated on $NSE_{\log Q}$ performs better for low-flow periods, while the model calibrated on $NSE_{\sqrt{Q}}$ performs in between.

Similarly to the use of different flow transformations during model calibration, the `airGRteaching CalGR()` function allows us to test several objective functions such as NSE or KGE (cf. command lines in Appendix D9).

3.2.5 Model evaluation and robustness

350 Split-sample tests, i.e., calibrating and evaluating a model on non-overlapping periods (Klemeš, 1986), is key for the assessment of model transferability in time, since in practice models are used outside their calibration conditions. Split-sample tests can be performed for model calibration and validation using both `CalGR()` and `SimGR() airGRteaching` functions, respectively (cf. command lines in Appendix D10).

The differential split-sample test, also introduced by Klemeš (1986), consists in identifying two climatically contrasted 355 periods in the available record and performing the split-sample test using these two periods. Table 3 presents the calibration / evaluation performance of the GR4J model obtained for two sub-periods, composed of the wettest and the driest hydrological years (based on the aridity index, i.e., the total annual precipitation divided by the total annual potential evapotranspiration, cf. command lines in Appendix D11).

Table 3. Example of differential split-sample results (KGE score) obtained for a given catchment.

Period	Calibration	Evaluation
Wet	0.974	0.836
Dry	0.962	0.886

4 One step further: projects on flow reconstruction, forecasting, and climate change

360 The basic manipulations of the `airGRteaching` package illustrated in the previous sections can also be used in more comprehensive hydrological teaching projects, presented in a vignette format in the package (example in Fig. 11) available both in English and French. These three projects deal with flow reconstruction (i.e., producing simulated streamflow over periods for which records are missing), flow forecasting (i.e., anticipating streamflow conditions for days ahead from given initial conditions), and climate change applications (i.e., transforming climate projections into hydrological projections). These

365 three projects can be run as stand-alone with the dataset available in the `airGRdatasets` package, or run on other catchments by importing the necessary hydro-climatic series.

1. *Streamflow reconstruction*. The Estéron at Broc catchment presents flow observation from 1999 to 2018, but also several missing data in 2004. This project aims to use the hydro-climatic series available and the GR2M model to reconstruct the missing flow data through rainfall-runoff simulation. The concepts addressed and the skills developed with this
370 project are (i) parameter calibration (both manually and automatically) using an objective function, and (ii) calibration-evaluation methodology.

2. *Low-flow forecasting*. This project aims to use the hydro-climatic data available for the Meuse at Saint-Mihiel catchment and the GR6J rainfall-runoff model to forecast the flows for the autumn of 2018, using (i) the last observed streamflow value, (ii) historical rainfall observations and (iii) historical flow observations (cf. Fig. 12). The concepts addressed and
375 the skills developed with this project are (i) the definition of climatology, (ii) flow forecasting and (iii) flow assimilation.

3. *Impact of climate change on streamflow regime*. Using catchment-scale delta-change-derived future climate projections, this project aims at quantifying the impact of climate change on the flow regime of the Durance at Embrun catchment (cf. Fig. 13). The concepts addressed and the skills developed with this project are the (i) delta-change method, (ii) flow regime, (iii) bias correction, and (iv) impact of snow on flow regime.

380 Users of the `airGRteaching` package may also produce their own exercises as `airGRteaching` vignettes, based on the three examples provided.

We believe that the proposed exercises and projects are must-do if one wishes to learn hydrological modeling. They represent the core of many catchment-related studies.

5 Limitations and perspectives

5.1 Limitations

Like any tool, `airGRteaching` has its limitations. The first one is that so far only GR hydrological models are available in `airGRteaching`. Adding other models is feasible, but to do so, they should be implemented to be compatible with the `airGR` framework (which contains the basic components for `airGRteaching`). While for the command-line use of `airGRteaching` (i.e. use of the `PrepGR()`, `CalGR()`, and `SimGR()` functions), this should be easy to implement, the GUI implementation would require more efforts (for instance, it would require the adding of a model scheme for each model, the interface could become less handy with models presenting over 10 parameters to optimise, and calibration would be far less rapid).

In addition, it is not possible for the user to build its own hydrological model by adding, for example, reservoirs (e.g., with different discharge functions) and unit hydrographs, to help understand each compartment of a model. This is possible with the `RS MINERVE` software (García Hernández et al., 2019).

Other limitations, as mentioned in section 2.3, are that `airGRteaching` offers only a limited set of modeling options, compared to `airGR`. This however could also be seen as a strength, as proposing too many options could be cumbersome on a user's perspective, and these limitations are therefore voluntary.

Remote sensing data, other than meteorological or hydrological data, cannot be used in `airGRteaching` at the moment. In addition, the effect of land cover changes cannot directly be assets to `airGRteaching` as is the case in some physically-based models.

Finally, proper uncertainty exercises, apart from the calibration on different periods, do not belong so far to this tool, which we see as a simple way of starting hydrological modeling. However, it is easy enough to add noise to the input data to see assess input uncertainty. Uncertainty arising from model structure can only be studied by changing models (e.g., using GR4J and GR5J models). The uncertainty associated with parameter calibration methods cannot be tested, as only an optimization algorithm is provided (NB: other algorithms can be plugged into `airGR`). Finally, `airGRteaching` does not provide turnkey tools as tool for visualizing uncertainty (e.g., error bars or envelopes on streamflow simulation).

5.2 Perspectives

Exercises linking hydrology with other disciplines and scientific communities could be developed by coupling of the `airGRteaching` package with other numerical tools and models. First, using actual Global or Regional Climate model outputs as rainfall-runoff model inputs would illustrate the impact of climate variability or emission scenarios on catchment hydrology, linking climatology and hydrology. In a similar way, streamflows produced by the `airGRteaching` package could be used as inputs to hydraulic models to produce flood maps in teaching projects involving both hydrological and hydraulic skills. Finally, coupling `airGRteaching` with models of water uses (e.g., water withdrawal models for drinking water or irrigation) would have interesting teaching applications. Another valuable perspective is to use remote sensing data to perform data assimilation for hydrological forecasting by recovering real-time meteorological (e.g., precipitation measured in raingauges), hydrological (e.g.,

streamflow observed from gauging stations), or even satellite data (e.g., MODIS snow cover observations) and using these data as inputs of a rainfall-runoff model in the `airGRteaching` package, e.g., with the `airGRdatassim` package (Piazzi et al., 2021; Piazzi and Delaigue, 2021). Such applications would illustrate the added value of assimilating hydro-meteorological data for better modeling in hydrology. Other exercises could be centered around uncertainties, through coupling the airGRteaching package with sensitivity analysis methods.

Finally, the `airGRteaching` package could be used for the development of serious games devoted to hydro-meteorological applications, aiming, for example, to discuss the issues of making better decisions when considering probabilistic forecasts (Ramos et al., 2013).

Author's experience with different audiences has shown that `airGRteaching` is useful in helping students understand a variety of basic concepts: from the choice of an objective function, to the sensitivity of model simulations to individual parameters, the difference between model states and model parameters, the difference between automatic and manual calibration, and the informative and complementary value of a variety of plots. Projects that are more elaborate have been developed and are listed in section 4. For students, depending on the time allotted and their experience, we use the graphical interface with or without the use of computer code. For researchers, it is more a matter of introducing them specifically to GR models, and the interface is used as an introduction of the GR model structure. For engineers working in consulting firms, it is often somewhere in between, depending on their experience and their background. The GUI is frequently used to avoid being bogged down in problems of form and to concentrate exclusively on the underlying concepts of hydrological modeling. The simplified code version allows a smooth transition to the more complex `airGR` code. For the general public, the aim is usually to introduce them using the `airGRteaching` GUI to one of the fields of hydrology, to help them understand what a model is, and to raise their awareness of applications such as flood and low-flow forecasting, and global change.

The introduction to computer programming is ideal to teach these notions to students. If students are to take this tool into their own hands, they must gradually acquire the concepts without difficulty. It is therefore essential that this is done in a playful way so that they are not discarded. The use of a graphical interface allowing to acquire modeling notions, while putting aside the programming aspects, allows to separate the different problems: modeling on the one hand and programming on the other hand. As soon as they wish to go further in the understanding of their subject, students very quickly perceive the limitations that a graphical interface can represent (options too limited, etc.). In addition, use can quickly become daunting if tasks need to be repeated (for example, clicking a large number of times and in a well-defined order to reproduce the results on several datasets). Sometimes there is not enough time to learn programming, justifying the need to use simple tools.

As such, the `airGRteaching` tool is not intended to be used to realise extended hydrological research studies, and therefore it does not aim to be used to contribute to the actual solving of any of the 23 UPHs (Blöschl et al., 2019). However, as it permits to teach hydrology, to understand hydrological processes and to masterize hydrological modeling, we believe that `airGRteaching` could be used as a preliminary step in the solving of some UPHs. Namely, UPH19 (*How can hydrological models be adapted to be able to extrapolate to changing conditions, including changing vegetation dynamics?*) and UPH20 (*How can we disentangle and reduce model structural/parameter/input uncertainty in hydrological prediction?*), due to the many model parameter manipulations and calibration/evaluation exercises that `airGRteaching` proposes, are good

candidates. This tool can contribute to UPH21 (*How can the (un)certainty in hydrological predictions be communicated to decision makers and the general public?*) as it has already been used by several decision makers in hydrological trainings. `airGRteaching` can be seen as a gateway to mastering `airGR` and other airGR-dependent packages, and thus indirectly

455 helping to solve other UPHs. This is notably the case for questions UPH22 (*What are the synergies and tradeoffs between societal goals related to water management (e.g., water-environment-energy-food-health)?*) and UPH23 (*What is the role of water in migration, urbanisation and the dynamics of human civilisations, and what are the implications for contemporary water management?*), linked to water usage, thanks to the `airGRiwrm` package (Dorchies et al., 2022) which allows to integrated water resources management. This package could help to solve problems of spatial heterogeneity and change of scale,

460 namely UPH5 (*What causes spatial heterogeneity and homogeneity in runoff, evaporation, subsurface water and material fluxes (carbon and other nutrients, sediments), and in their sensitivity to their controls (e.g., snow fall regime, aridity, reaction coefficients)?*) and UPH6 (*What are the hydrologic laws at the catchment scale and how do they change with scale?*), because it simplifies the use of `airGR` in a semi-distributed mode. The `airGRdatassim` package, which enables data assimilation, could be link to questions of prediction uncertainty, namely UPH20.

465 ## 6 Conclusions

Teaching hydrological modeling requires hands-on experience with rainfall-runoff models. Dedicated tools need to be adapted to the skills of the students and users, and preferably developed in an open-source programming language to ensure the reproducibility of the results. In this context, the `airGRteaching` R package has been developed as an add-on to the `airGR` package, which gathers several lumped rainfall-runoff models widely used by hydrological researchers and practitioners.

470 `airGRteaching` contains a graphical user interface and allows teachers and students to import their own data and create their own exercises. A specific dataset of 19 different catchments in France is included in the add-on `airGRdatasets` package. This dataset is composed of hydro-climatic time series (solid and liquid precipitation, potential evapotranspiration, air temperature, and streamflow time series). Finally, three hydrological teaching projects are proposed aimed at (i) using a monthly rainfall-runoff model to reconstruct flow series, (ii) using a daily model to forecast low flows, and (iii) studying the

475 impact of climate change on streamflow of a mountainous catchment. Thanks to its open nature, other projects may be added to the package by `airGRteaching` users, based on the dataset provided or other datasets.

Code and data availability. The code and data used in this manuscript are included in the `airGRteaching` and `airGRdatasets` packages that are available from the CRAN (https://CRAN.R-project.org/package=airGRteaching)

Figure 2. `airGRteaching` GUI "Modeling" panels (a-d) and "Summary sheet" panels (e-f) that can be reached through diverse clicking. In the following, the center column of the GUI is described for each possible panel; all other elements of the GUI were described in Fig. 1. (a) "Flow time series": precipitation, observed and simulated hydrographs (top), and flow error time series (bottom). (b) "Model performance": precipitation (top), observed and simulated hydrographs (middle), simulated and observed regime hydrographs (bottom left), flow duration curves (bottom center), and a scatter plot between simulated and observed discharges (bottom right). (c) "State variables": time series of reservoir levels (top) and runoff components (bottom). (d) "Model diagram": time series (left) of precipitation, potential evapotranspiration, simulated and observed flows (from top to bottom), and interactive model diagram (right; with updating of the flows, the size and the level of the reservoirs). (e) Hydrometeorological and topographical characteristics of the selected catchment (Brigode et al., 2020, only available for French catchments). (f) Same as (e) when the catchment characteristics are not available.
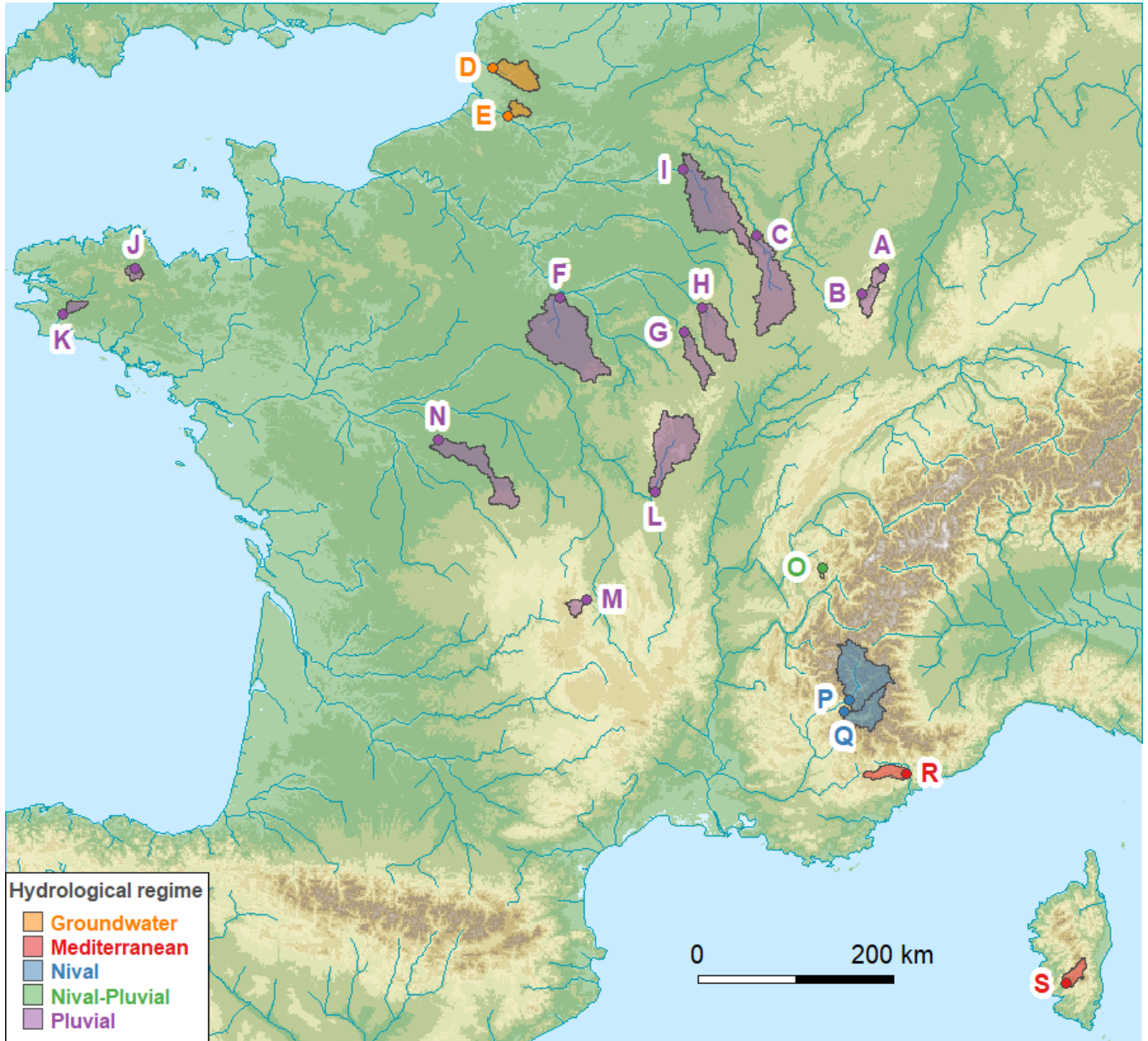
Figure 3. Location of the 19 catchments in France included in the `airGRdatasets` package (map from the `airGRdatasets` package documentation (Delaigue et al., 2023a), using: hydrometric station coordinates and catchment boundaries: Delaigue et al., 2022; river network: Lehner and Grill, 2013); DEM: GEBCO Bathymetric Compilation Group 2021, 2021).
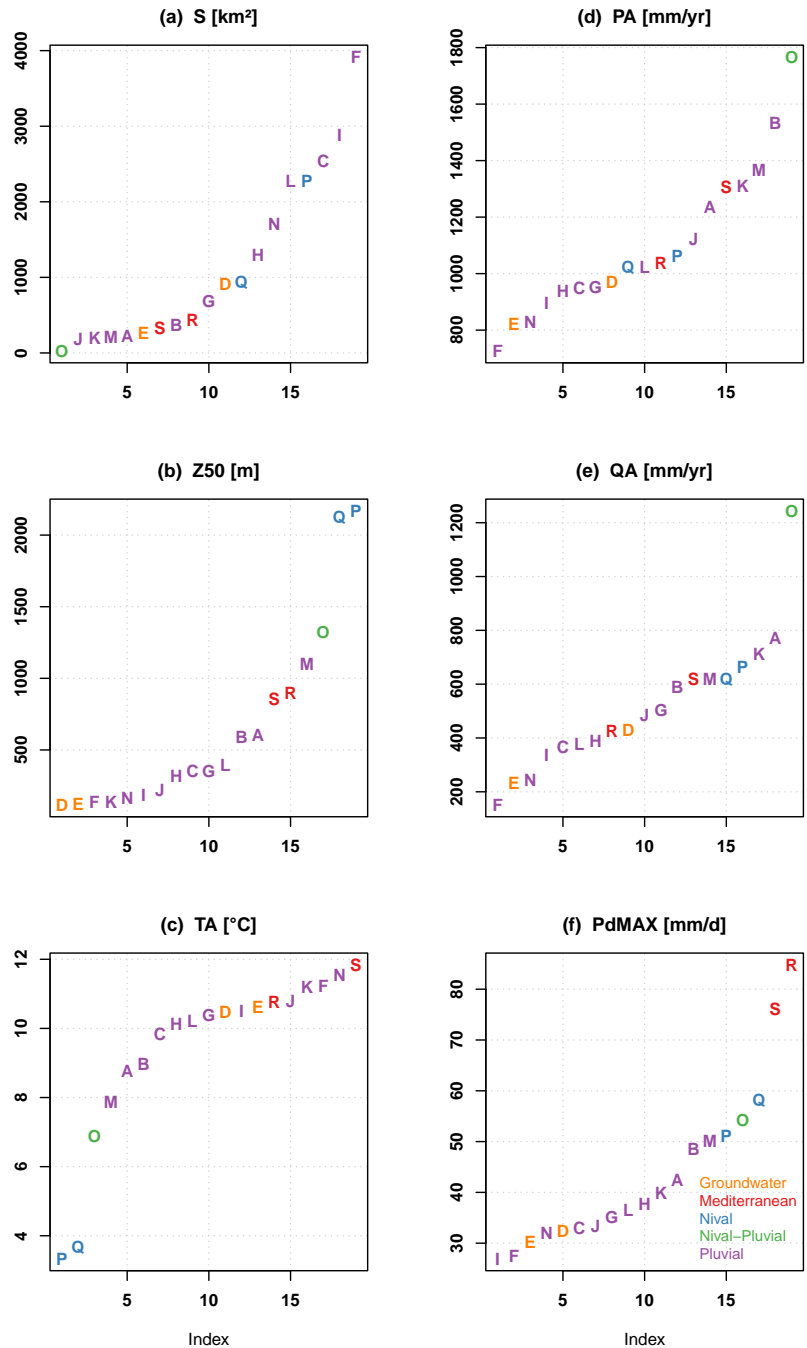
Figure 4. Distribution of the characteristics of the 19 catchments included in the `airGRdatasets` package. (a) "S": area (km^2); (b) "Z50": median altitude (m a.s.l.); (c) "TA": median of the mean annual air temperature (°C); (d) "PA": median of the annual precipitation (mm/year); (e) "QA": median of the annual flow (mm/year); (f) "PdMAX": median of the maximum annual daily precipitation (mm/day), versus the catchment indexes. The statistics have been calculated over the available daily time series in the `airGRdatasets` package (i.e. from 1999-01-01 to 2018-12-31, only the years with less than 10 % of missing streamflow values have been considered).
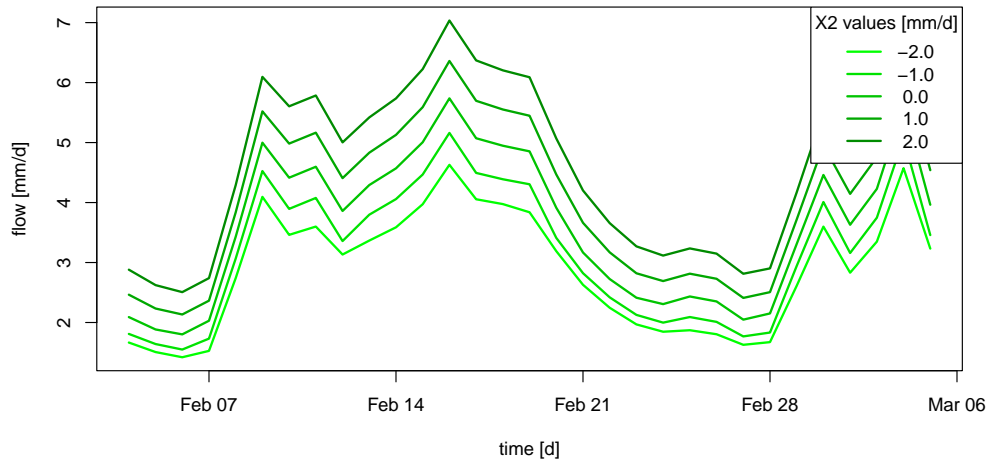
Figure 5. The role of the production component in GR4J illustrated by an example of flow simulation sensitivity to the X2 parameter values (groundwater exchange coefficient, mm/day).
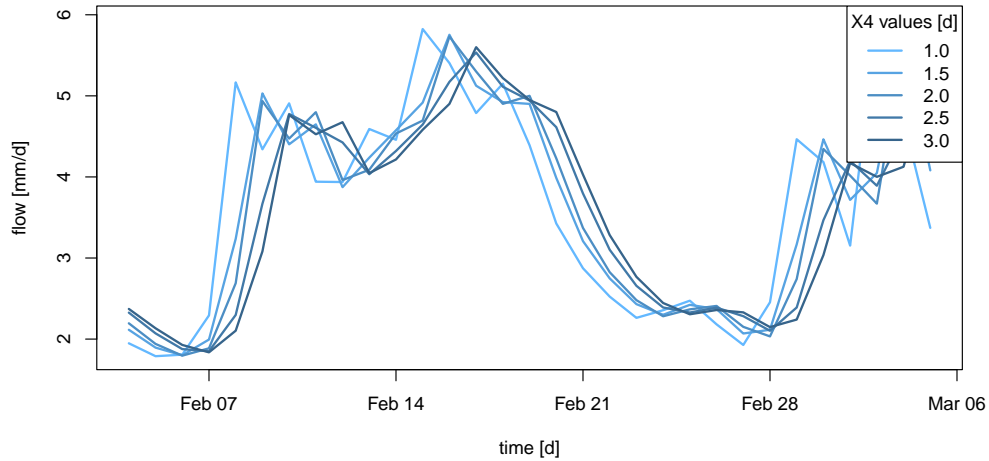
Figure 6. The role of the routing component in GR4J illustrated by an example of flow simulation sensitivity to the X4 parameter values (time base of unit hydrographs, in days).
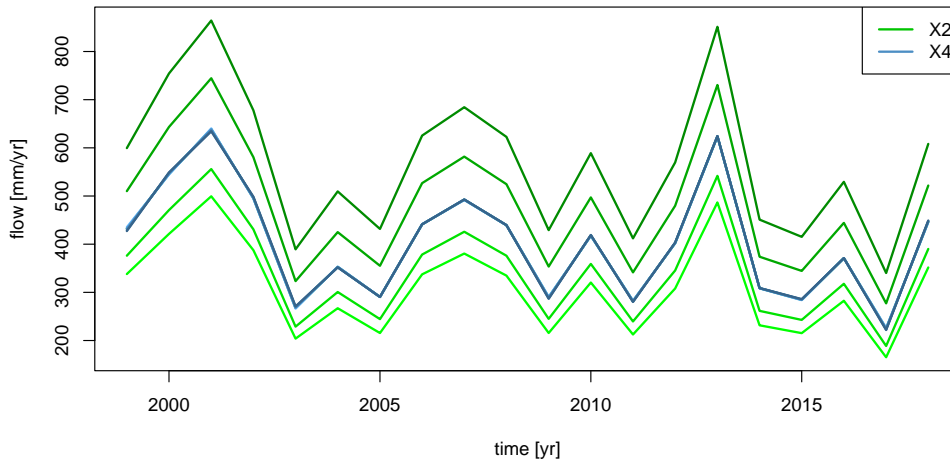
Figure 7. Comparison, at the annual time step, between GR4J daily simulations performed with different X2 parameter values (in green gradient) and simulations performed with different X4 parameter sets (in blue gradient).
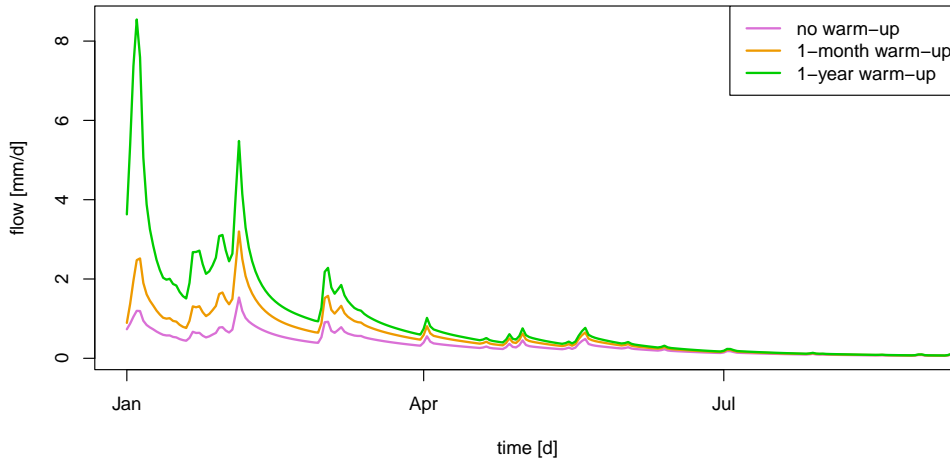
Figure 8. Example of streamflow simulations obtained considering no warm-up period (in purple), a 1-month warm-up period (in orange), and a 1-year warm-up period (in green) of the GR4J two stores.
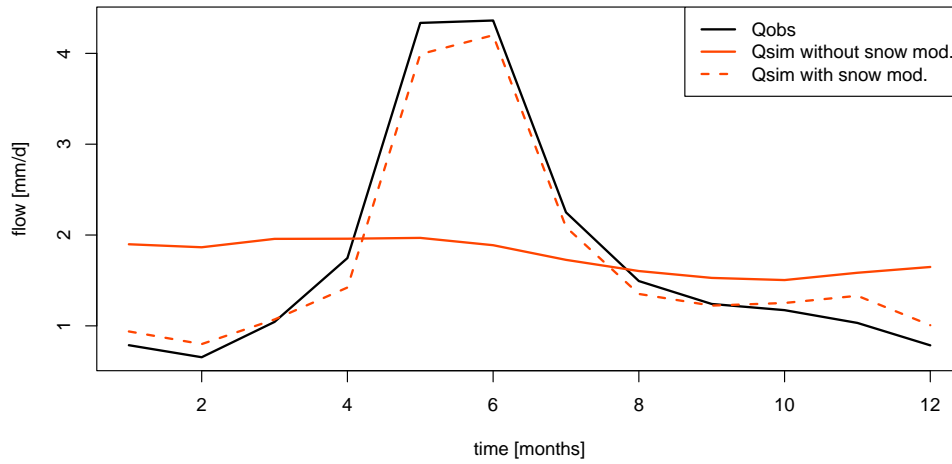
Figure 9. Example of flow regimes observed for a catchment located in the French Alps (in black), and flow regimes simulated by GR4J without considering snow accumulation and melting (solid red line) or when a snow accumulation and melting routine is used (dashed red line).
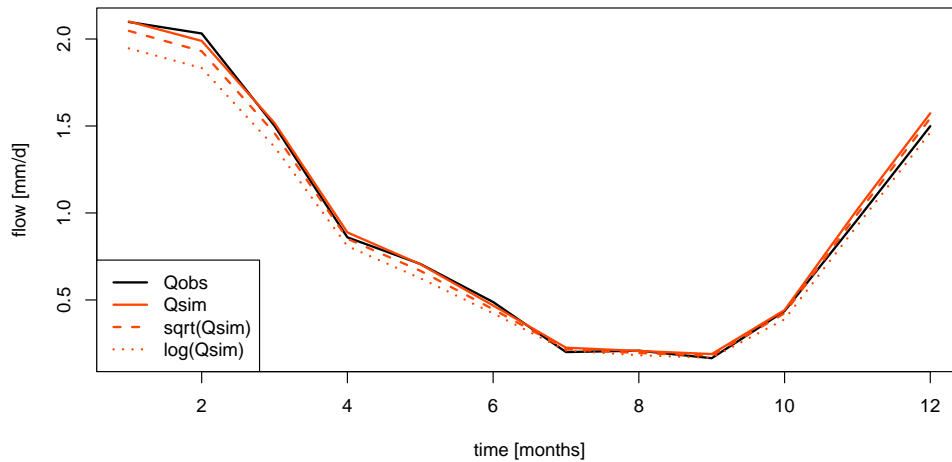
Figure 10. Example of observed flow regimes (in black), and flow simulations obtained when GR4J is calibrated on NSE calculated on untransformed flows (solid red line), NSE calculated on square root transformed flows (dashed red line), and NSE calculated on logarithmic transformed flows (dotted red line).

Model calibration

Manual calibration

In the **airGRteaching** GUI, it is possible to test different parameter set of the GR rainfall-runoff models and to estimate the performance of each tested parameter sets in order to perform a manual calibration. A classical way to do so through the **airGRteaching** GUI is to pick an objective function in the table showing the criteria values on the right, to activate the "Show previous simulations (Qold)" button, and to modify parameter values little by little until the simulation and the criterion are satisfying.

Note that the manual calibration of model parameters is facilitated using the **airGRteaching** graphical user interface launched by the `ShinyGR()` function.

You can embed the following code in a loop (except the data processing step using `PrepGR()`). At each iteration you test a new parameter set and compute corresponding the criterion. This way you can find the "best" parameter set.

```
# Data processing for GR4J (with Q for calibration)
prep <- PrepGR(DatesR    = ts_obs$Date,
               Precip    = ts_obs$Ptot,
               PotEvap   = ts_obs$Evap,
               Qobs      = ts_obs$Qmmd,
               HydroModel = "GR4J",
               CemaNeige = FALSE)

# Parameter set to test
i_param_gr4j <- c(X1 = 350, X2 = 0, X3 = 90, X4 = 1.4)

# Rainfall-runoff simulation on the calibration period
i_sim_manu <- SimGR(PrepGR  = prep,
                    Param   = param_gr4j,
                    WupPer  = c("1999-01-01", "2000-12-31"),
                    SimPer  = c("2001-01-01", "2010-12-31"),
                    EffCrit = "NSE",
                    verbose = TRUE)
```

```
## Crit. NSE[Q] = 0.6254
```

```
# Get the criterion value
GetCrit(i_sim_manu)
```

Automatic calibration

Automatic calibration of model parameters is also possible in **airGRteaching** using the procedure described by Michel (1991) and by considering one objective function such as NSE (Nash and Sutcliffe 1970) or KGE (Gupta et al. 2009). To do so, there are two options in airGRteaching:

1. clicking on the "Automatic calibration button" in the **airGRteaching** GUI.
2. using the simple **airGRteaching** command line functions (`PrepGR()` and `CalGR()`), see following commands.

```
# Calibration using NSE score
cal_auto <- CalGR(PrepGR  = prep,
                  CalCrit = "NSE",
                  WupPer  = c("1999-01-01", "2000-12-31"),
                  CalPer  = c("2001-01-01", "2010-12-31"))
```

```
## Grid-Screening in progress (0% 20% 40% 60% 80% 100%)
##    Screening completed (81 runs)
##       Param = 432.681,   -0.649,   83.096,    2.384
##       Crit. NSE[Q]    = 0.7327
## Steepest-descent local search in progress
##    Calibration completed (27 iterations, 280 runs)
##       Param = 242.257,   -0.888,   63.434,    5.155
##       Crit. NSE[Q]    = 0.9098
```

```
# Get parameter and criteria values at the end of the calibration step
GetParam(cal_auto)
```

```
## [1] 242.257207  -0.888106  63.434000   5.155405
```

```
GetCrit(cal_auto)
```

```
##     NSE[Q]
## 0.9097516
```

Figure 11. Example of a vignette explaining how to perform both manual (left) and automatic (right) calibration of model parameters using the `airGRteaching` package.
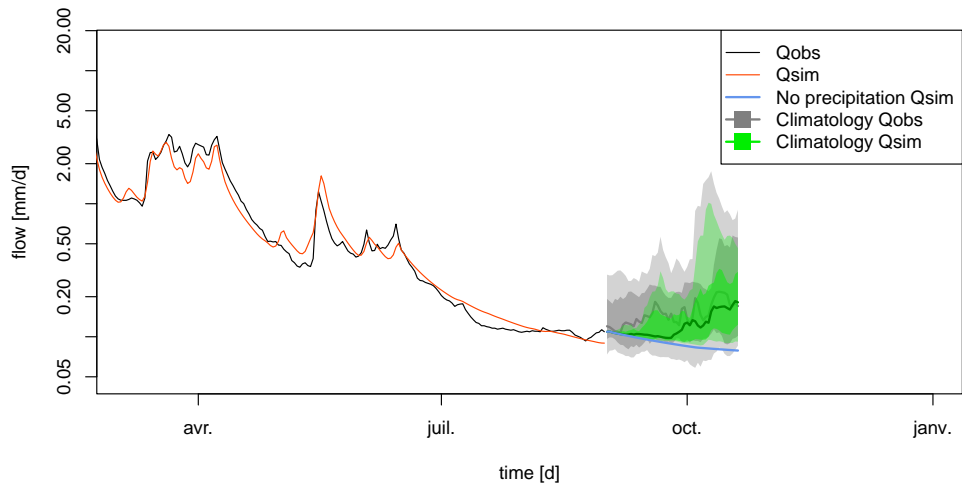
Figure 12. Final output of the `airGRteaching` "Low-flow forecasting" vignette: observed flow (in black), simulated flow (in red), and different forecast scenarios (in blue: simulated streamflow based on the pessimistic zero precipitation scenario; in gray: streamflow quantiles (10, 25, 50, 75 and 90 %) based of historical past flow observations; in green: simulated streamflow quantiles (10, 25, 50, 75 and 90 %) based on the precipitation climatology).
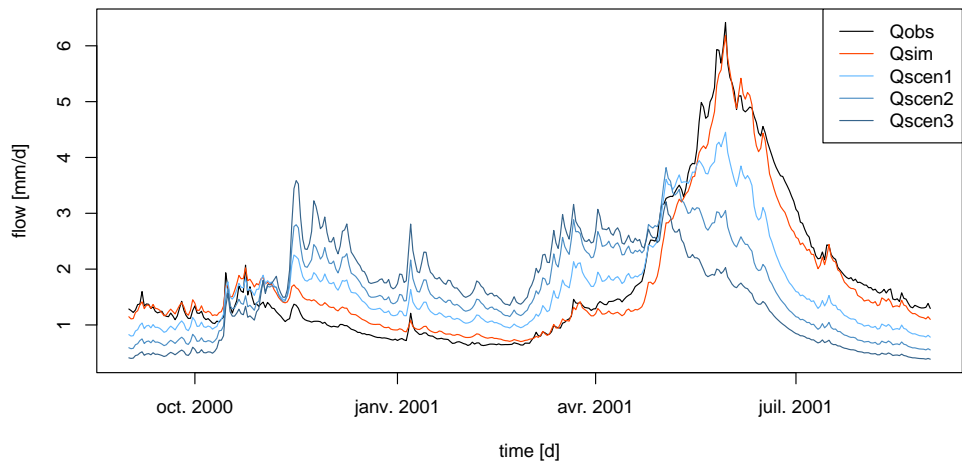
Figure 13. Final output of the `airGRteaching` "Impact of climate change on streamflow regime" vignette: flow regimes observed (in black), calibrated over the historical period (in red), and simulated using different climate change scenarios (in blue gradient).

Appendix A: Static plots produced by the package

480   In this appendix, we have used the time series of the X045401001 catchment (the Ubaye at Lauzet-Ubaye [Roche-Rousse] -
DREAL PACA). The GR5J model, coupled to CemaNeige, was calibrated on the raw flows of the period from 2001-01-01 to
2004-12-31. The objective function used is the KGE.
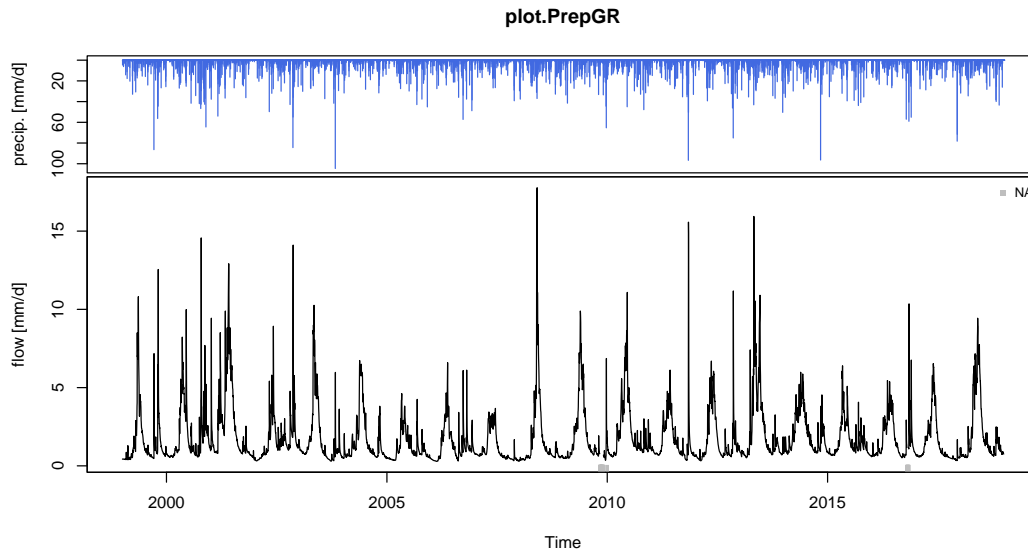
Figure A1. Plot generated using the outputs of the `PrepGR()` function: precipitation time series (top); observed hydrograph (bottom).
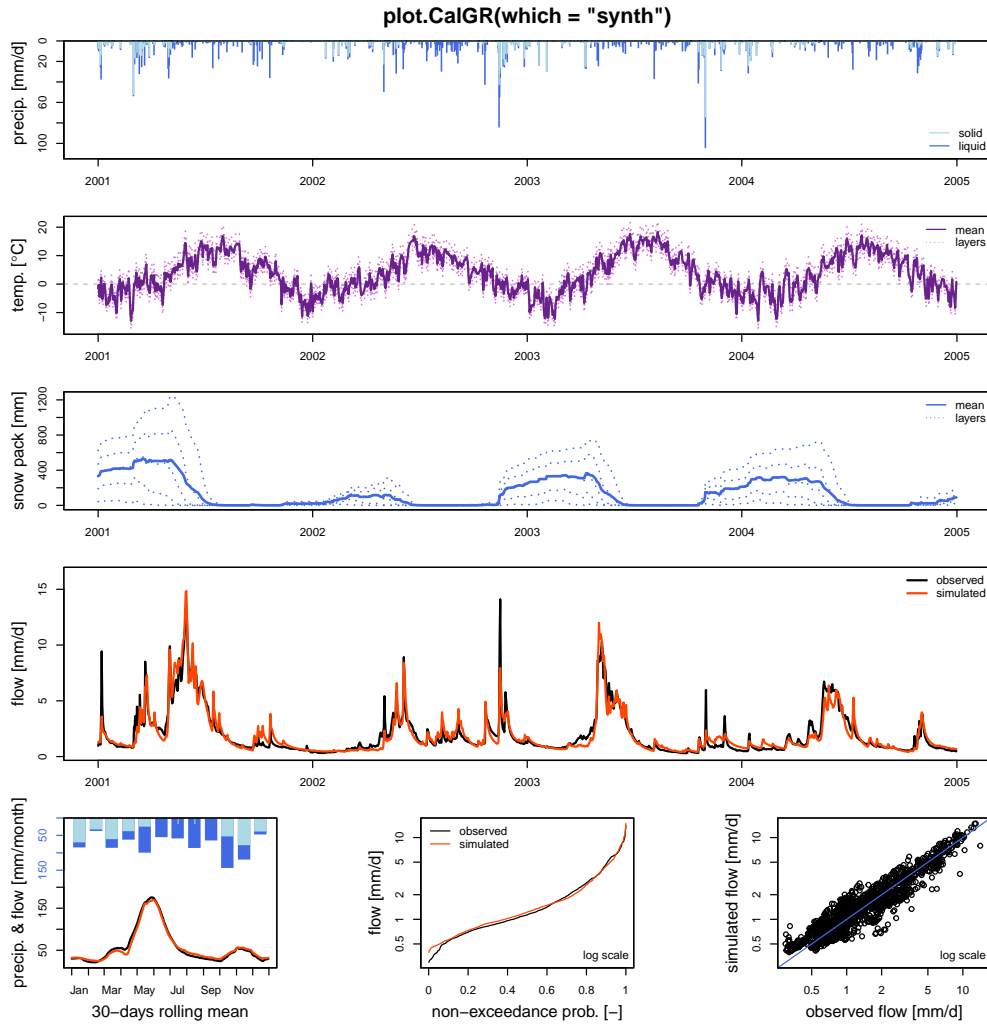
Figure A2. Plot generated using the outputs of the `CalGR()` or the `SimGR()` functions, when argument `which = "synth"` (synthesis; default value). From top to bottom and from left to right: precipitation time series (liquid and solid, if CemaNeige is used); potential evapotranspiration time series; temperature time series for each layer (if CemaNeige is used); snow pack time series for each layer (if CemaNeige is used); monthly average precipitation (liquid and solid, if CemaNeige is used) and 30-day rolling mean of interannual mean daily streamflow; observed and simulated flow duration curves; scatter plot between between observed and simulated discharges. The hydrographs can also be plotted with a log scale.
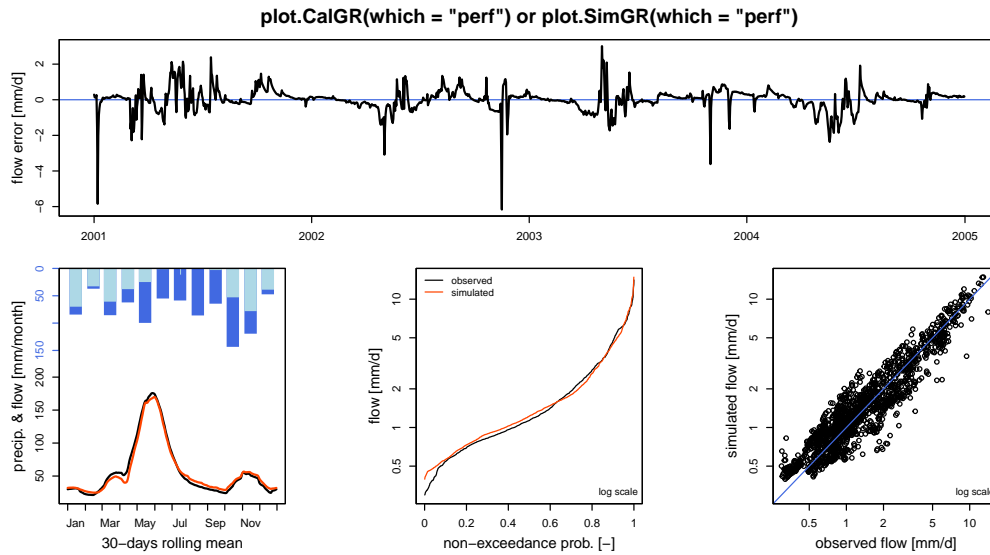
Figure A3. Plot generated using the outputs of the `CalGR()` or the `SimGR()` functions, when argument `which = "perf"` (performance). From top to bottom and from left to right: flow error (or residuals); monthly average liquid and solid (if CemaNeige is used) precipitation and 30-day rolling mean of interannual mean daily streamflow; cumulative frequency; correlation plot. The flow error chart can also be plotted with a log scale.
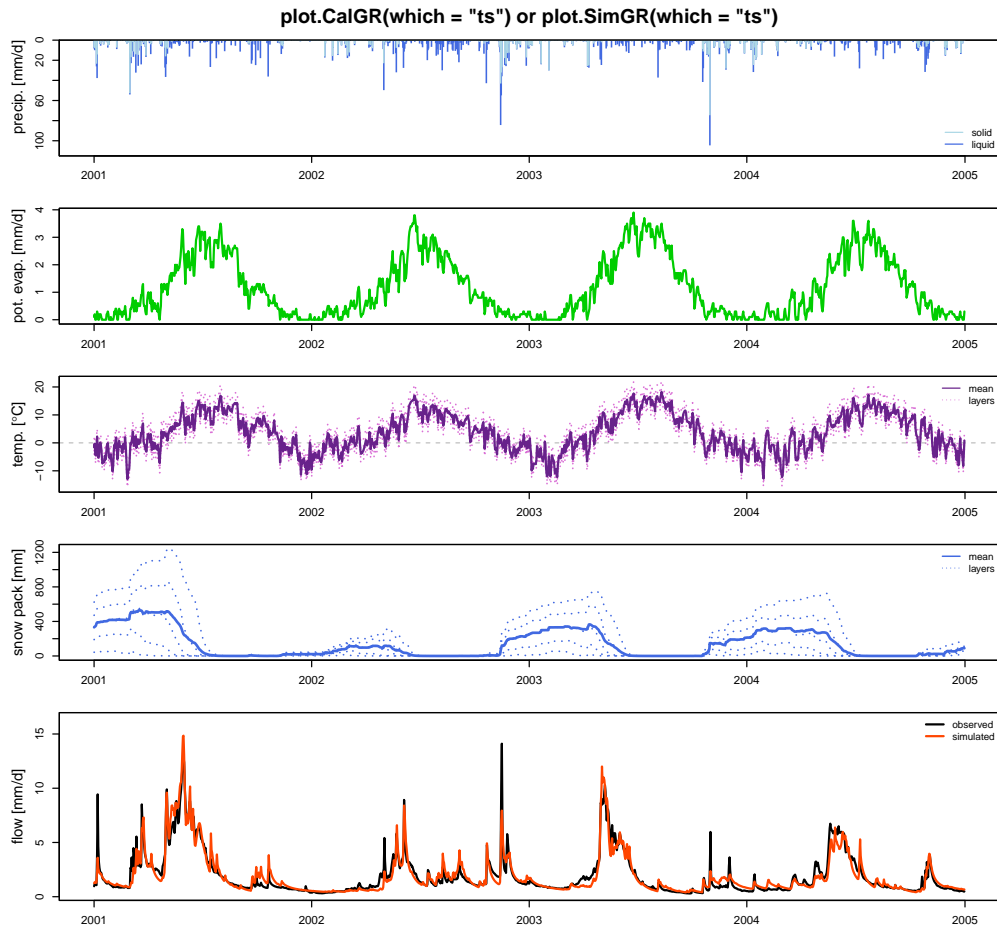
Figure A4. Plot generated using the outputs of the `CalGR()` or the `SimGR()` functions, when argument `which = "ts"` (time series). From top to bottom: precipitation time series (liquid and solid if CemaNeige is used); potential evapotranspiration time series; air temperature time series for each layer (if CemaNeige is used); snow pack time series for each layer (if CemaNeige is used); observed and simulated hydrographs. The hydrographs can also be plotted with a log scale.
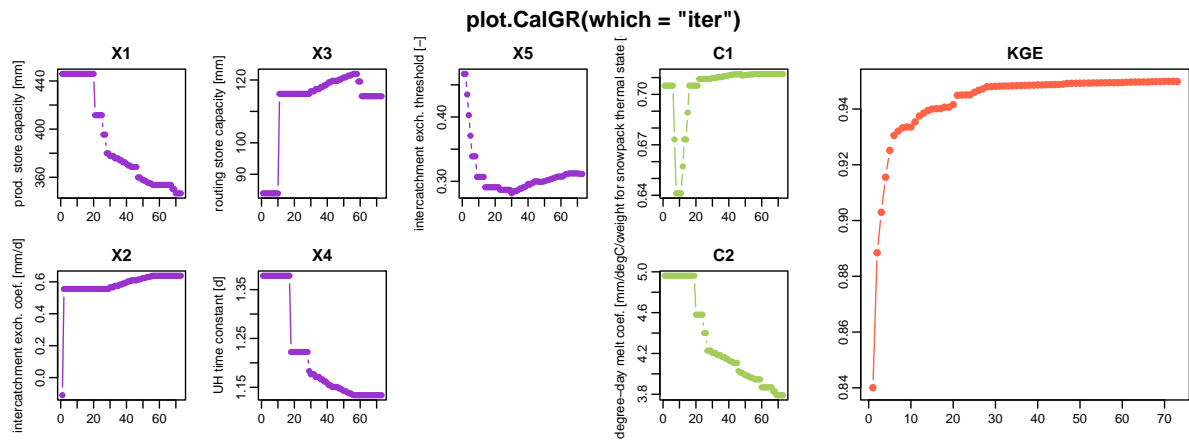
Figure A5. Plot generated using the outputs of the `CalGR()` function, when argument `which = "iter"` (iterations). From left to right: evolution of parameters of the GR5J model (in purple) and CemaNeige model (in green) and of the efficiency criterion (in orange) during the iterations of the calibration steepest-descent step.

Appendix B: Dynamic plots returned by the package

In this appendix, we have used the time series of the X045401001 catchment (the Ubaye at Lauzet-Ubaye [Roche-Rousse] -

485   DREAL PACA). The GR5J model, coupled to CemaNeige, was calibrated on the raw flows of the period from 2001-01-01 to 2004-12-31. The objective function used is the KGE.
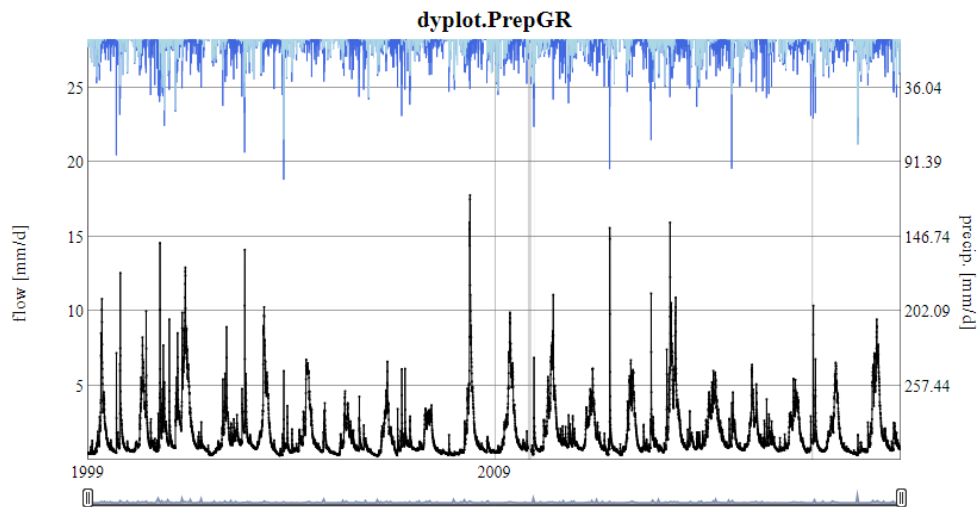
Figure B1. Dynamic plot generated using the outputs of the `PrepGR()` function: precipitation time series (liquid and solid, if CemaNeige is used) (top); observed hydrograph (bottom).
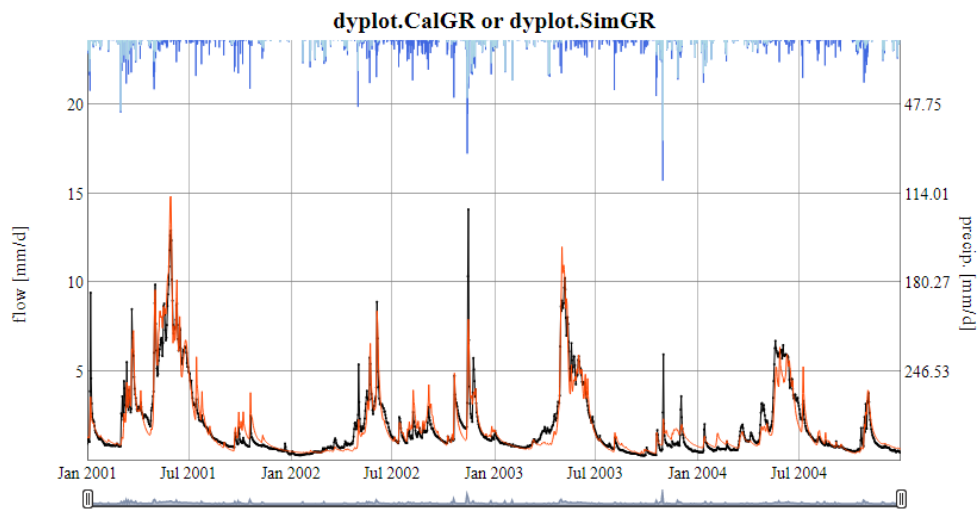
Figure B2. Dynamic plot generated using the outputs of the `CalGR()` or the `SimGR()` functions: precipitation time series (liquid and solid, if CemaNeige is used) (top); observed and simulated hydrographs (bottom).

Appendix C: Static plots downloaded from the `airGRteaching` GUI

In this appendix, we have used the time series of the X045401001 catchment (the Ubaye at Lauzet-Ubaye [Roche-Rousse] - DREAL PACA). The GR5J model, coupled to CemaNeige, was calibrated on the raw flows of the period from 2001-01-01 to 2004-12-31. The objective function used is the KGE.
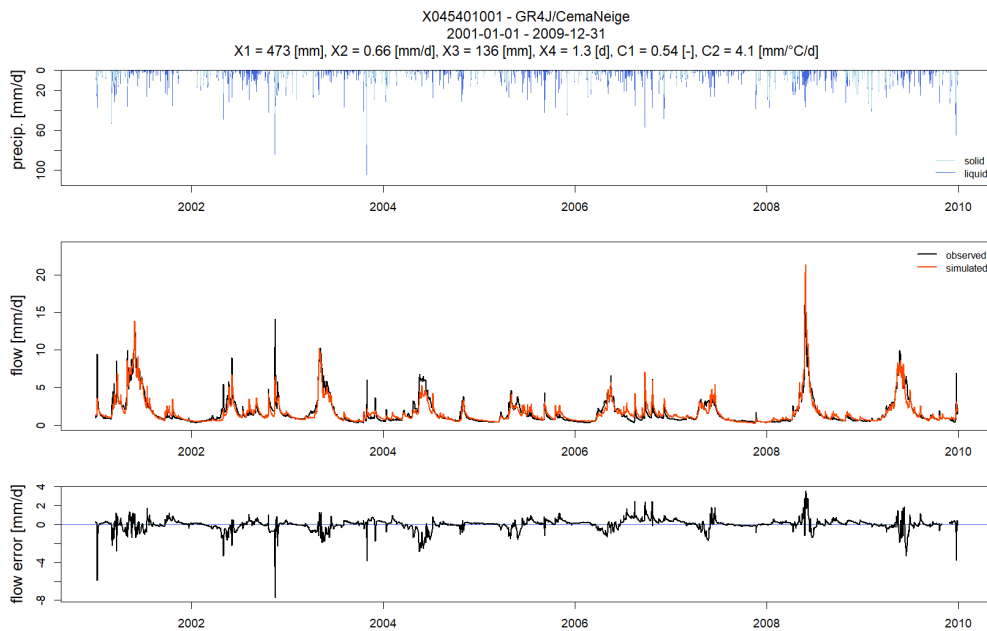
Figure C1. Static plot downloaded from the "Flow time series" tab of the GUI. From top to bottom: solid and liquid precipitation; observed and simulated hydrographs; flow error time series.
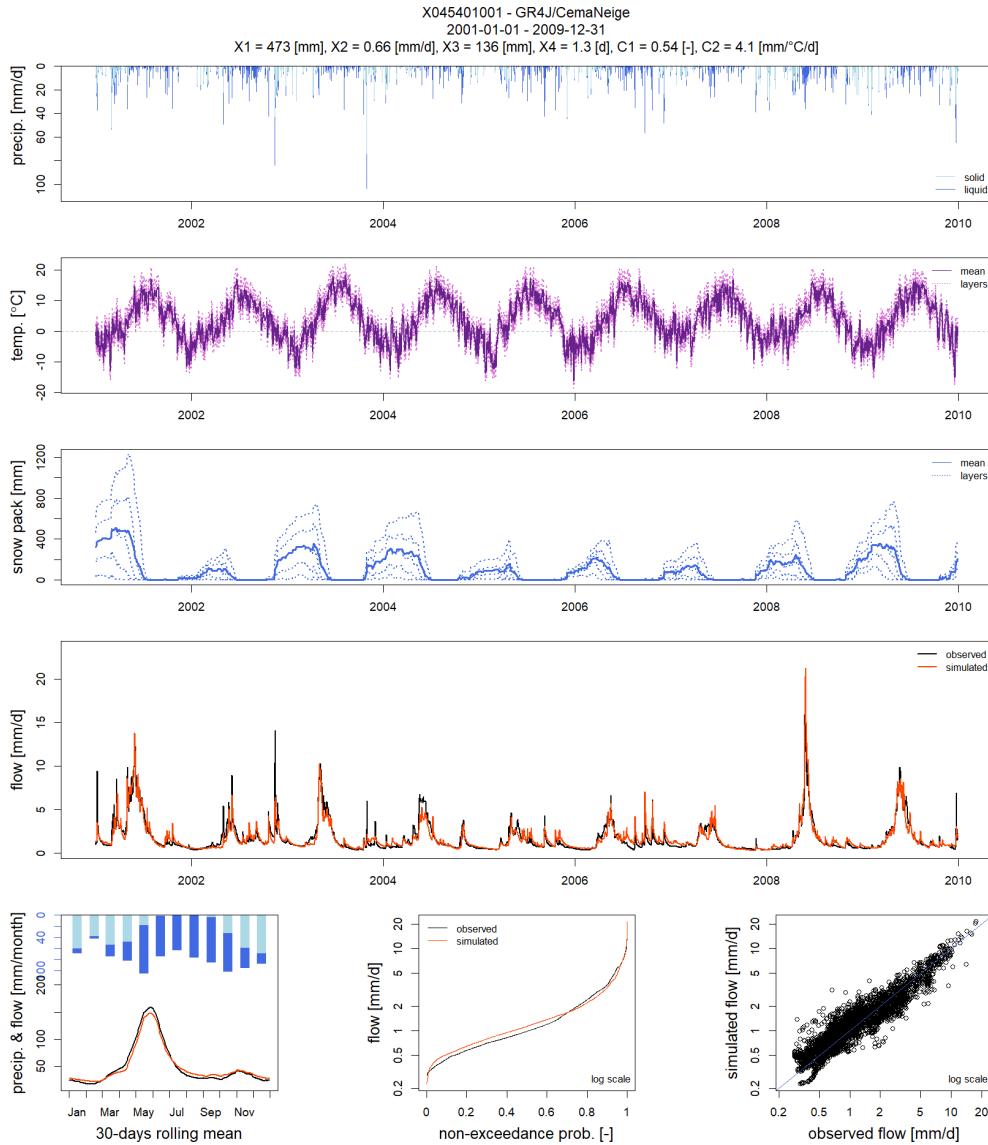
Figure C2. Static plot downloaded from the "Model performance" tab of the GUI. From top to bottom and from left to right: precipitation, temperature time series for each layer (if CemaNeige is used); snow pack time series for each layer (if CemaNeige is used), observed and simulated hydrographs, monthly average precipitation (liquid and solid, if CemaNeige is used), and 30-day rolling mean of interannual mean daily streamflow; observed and simulated flow duration curves; scatter plot between observed and simulated discharges.
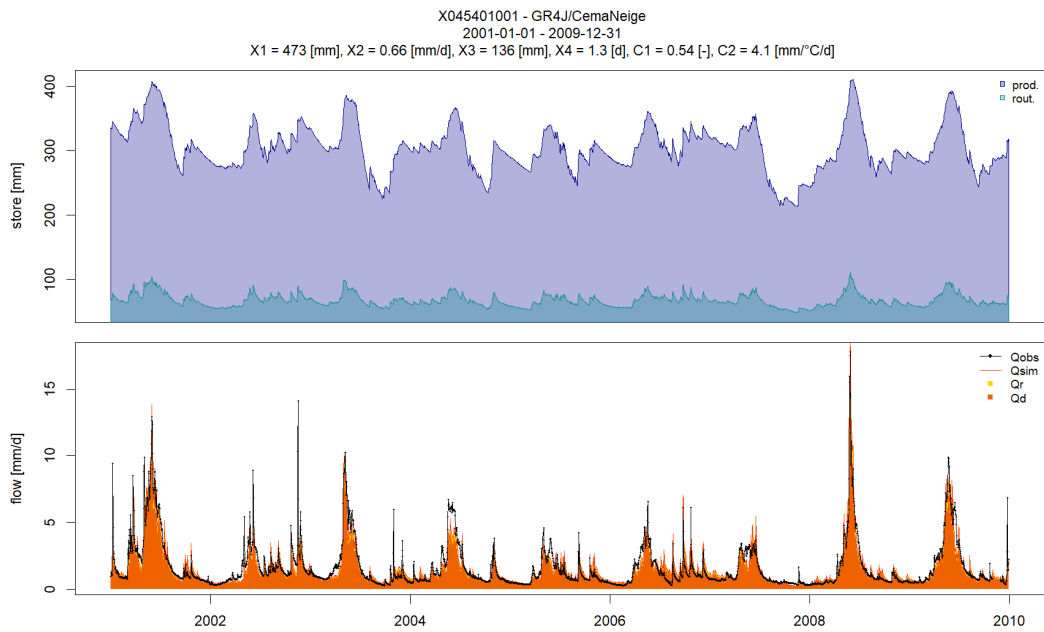
Figure C3. Static plot downloaded from the "State variables" tab of the GUI: time series of store levels (top); runoff components (bottom).
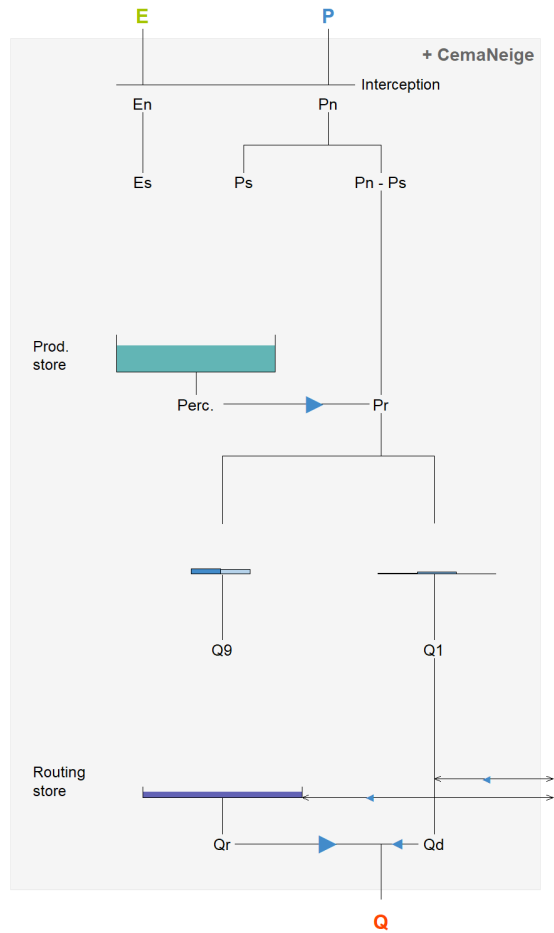
Figure C4. Static plot downloaded from the "Model diagram" tab of the GUI. Model diagram with adaptation of the arrows representing the different fluxes and of the maximal size and the level of the reservoirs according to the actual parameter values and to the values of all internal variables of the model.

Appendix D: "Teaching hydrology with airGRteaching" vignette command lines

D1   Role of the production component in GR4J

```
1: # Different X2 values around its median values (0 [mm/day])
2: param_x2 <- seq(from = -2, to = 2, by = 1)
3:
4: # Combination of parameter values (X1, X3 and X4 are fixed; X2 changes)
5: param_gr4j <- expand.grid(X1 = 350,
6:                           X2 = param_x2,
7:                           X3 = 90,
8:                           X4 = 1.4)
9:
10: # Streamflow simulations using parameter sets
11: sim_x2 <- apply(param_gr4j, MARGIN = 1, FUN = function(i_param_gr4j) {
12:    i_sim <- SimGR(PrepGR = prep_no_q,
13:                   Param = i_param_gr4j,
14:                   SimPer = per_sim,
15:                   verbose = FALSE)
16:    i_sim$OutputsModel$Qsim
17: })
18:
19: # Graphical comparison
20: ind_zoom <- 400:430
21: col_param_x2 <- colorRampPalette(c("green1", "green4"))(ncol(sim_x2))
22: matplot(x = as.POSIXct(prep_no_q$InputsModel$DatesR[ind_zoom]),
23:         y = sim_x2[ind_zoom, ],
24:         xlab = "time [days]", ylab = "flow [mm/day]",
25:         type = "l", lty = 1, lwd = 2, col = col_param_x2)
26: legend("topright",
27:        legend = sprintf("% .1f", param_x2),
28:        lwd = 2, col = col_param_x2,
29:        title = "X2 values [mm/day]")
```

D2 Role of the routing component in GR4J

```
1: # Different X4 values around its median values (1.4 [days])
2: param_x4 <- seq(from = 1.0, to = 3.0, by = 0.5)
3:
4: # Combination of parameter values (X1, X2 and X3 are fixed; X4 changes)
5: param_gr4j <- expand.grid(X1 = 350,
6:                           X2 = 0,
7:                           X3 = 90,
8:                           X4 = param_x4)
9:
10: # Streamflow simulations using parameter sets
11: sim_x4 <- apply(param_gr4j, MARGIN = 1, FUN = function(i_param_gr4j) {
12:   i_sim <- SimGR(PrepGR = prep_no_q,
13:                  Param = i_param_gr4j,
14:                  SimPer = per_sim,
15:                  verbose = FALSE)
16:   i_sim$OutputsModel$Qsim
17: })
18:
19: # Graphical comparison
20: ind_zoom <- 400:430
21: col_param_x4 <- colorRampPalette(c("steelblue1", "steelblue4"))(ncol(sim_x4))
22: matplot(x = as.POSIXct(prep_no_q$InputsModel$DatesR[ind_zoom]),
23:         y = sim_x4[ind_zoom, ],
24:         xlab = "time [days]", ylab = "flow [mm/day]",
25:         type = "l", lty = 1, lwd = 2, col = col_param_x4)
26: legend("topright",
27:        legend = sprintf("% .1f", param_x4),
28:        lwd = 2, col = col_param_x4,
29:        title = "X4 values [days]")
```

D3 Relative importance of the production and routing functions

```r
1: # Aggregation of the simulated streamflow at the yearly time step
2: sim_x2_y <- cbind(DatesR = as.POSIXct(prep_no_q$InputsModel$DatesR),
3:                   as.data.frame(sim_x2))
4: sim_x2_y <- SeriesAggreg(x = sim_x2_y,
5:                          Format = "%Y",
6:                          ConvertFun = rep("sum", ncol(sim_x2_y) - 1))
7: sim_x4_y <- cbind(DatesR = as.POSIXct(prep_no_q$InputsModel$DatesR),
8:                   as.data.frame(sim_x4))
9: sim_x4_y <- SeriesAggreg(x = sim_x4_y,
10:                         Format = "%Y",
11:                         ConvertFun = rep("sum", ncol(sim_x4_y) - 1))
12:
13: # Graphical comparison
14: matplot(x = sim_x2_y$DatesR, y = sim_x2_y[, -1],
15:         type = "l", lty = 1, lwd = 2, col = col_param_x2,
16:         xlab = "time [years]", ylab = "flow [mm/year]")
17: matlines(x = sim_x4_y$DatesR, y = sim_x4_y[, -1],
18:          type = "l", lty = 1, lwd = 2, col = col_param_x4)
19: legend("topright",
20:        legend = c("X2", "X4"),
21:        lwd = 2, col = c(median(col_param_x2), median(col_param_x4)))
```

D4 On the need to perform a model warm-up

```
 1: # Warm-up and simulation periods
 2: per_wup1m <- c("2002-12-01", "2002-12-31")
 3: per_wup1y <- c("2002-01-01", "2002-12-31")
 4: per_sim   <- c("2003-01-01", "2006-12-31")
 5:
 6: # Parameter set
 7: param_gr4j <- c(X1 = 350, X32 = 0, X3 = 90, X4 = 1.4)
 8:
 9: # Simulation without warm-up period
10: sim_wup0d <- SimGR(PrepGR  = prep_no_q,
11:                    Param   = param_gr4j,
12:                    WupPer  = 0L,
13:                    SimPer  = per_sim)
14:
15: # Simulation with a 1-month warm-up period
16: sim_wup1m <- SimGR(PrepGR  = prep_no_q,
17:                    Param   = param_gr4j,
18:                    WupPer  = per_wup1m,
19:                    SimPer  = per_sim)
20:
21: # Simulation with a 1-year warm-up period
22: sim_wup1y <- SimGR(PrepGR  = prep_no_q,
23:                    Param   = param_gr4j,
24:                    WupPer  = per_wup1y,
25:                    SimPer  = per_sim)
26:
27: # Graphical comparison
28: col_wup <- c("orchid", "orange2", "green3")
29: matplot(x = as.POSIXct(sim_wup0d$OutputsModel$DatesR),
30:         y = cbind(sim_wup0d$OutputsModel$Qsim,
31:                   sim_wup1m$OutputsModel$Qsim,
32:                   sim_wup1y$OutputsModel$Qsim),
33:         xlab = "time [days]", ylab = "flow [mm/day]",
34:         type = "l", lty = 1, lwd = 2, col = col_wup,
35:         xlim = as.POSIXct(x = c("2003-01-01", "2003-09-01"), tz = "UTC"))
36: legend("topright",
```

```
37:        legend = c("no warm-up", "1-month warm-up", "1-year warm-up"),
38:        col = col_wup, lwd = 2)
```

620

D5 Manual calibration

```
1: # Data processing for GR4J (with Q for calibration)
2: prep <- PrepGR(DatesR    = ts_obs$Date,
3:                Precip    = ts_obs$Ptot,
4:                PotEvap   = ts_obs$Evap,
5:                Qobs      = ts_obs$Qmmd,
6:                HydroModel = "GR4J",
7:                CemaNeige = FALSE)
8:
9: # Parameter set to test
10: i_param_gr4j <- c(X1 = 350, X2 = 0, X3 = 90, X4 = 1.4)
11:
12: # Rainfall-runoff simulation on the calibration period
13: i_sim_manu <- SimGR(PrepGR  = prep,
14:                     Param   = param_gr4j,
15:                     WupPer  = c("1999-01-01", "2000-12-31"),
16:                     SimPer  = c("2001-01-01", "2010-12-31"),
17:                     EffCrit = "NSE",
18:                     verbose = TRUE)
19:
20: # Get the criterion value
21: GetCrit(i_sim_manu)
22:
23: # Graphical assessment of the calibration performance
24: plot(i_sim_manu)
```

D6 Automatic calibration

```
1: # Calibration using NSE score
2: cal_auto <- CalGR(PrepGR  = prep,
3:                   CalCrit = "NSE",
4:                   WupPer  = c("1999-01-01", "2000-12-31"),
5:                   CalPer  = c("2001-01-01", "2010-12-31"))
6:
7: # Get parameter and criteria values at the end of the calibration step
8: GetParam(cal_auto)
9: GetCrit(cal_auto)
10:
11: # Graphical assessment of the calibration performance
12: plot(cal_auto)
```

D7 Model evaluation

```
665  1: # Catchment data loading
     2: data("X031001001", package = "airGRdatasets")
     3:
     4: # Observed daily time series
     5: ts_obs <- X031001001$TS
670  6:
     7: # Catchment elevation distribution
     8: hypso <- X031001001$Hypso
     9:
    10: # Temporal subset
675 11: is_per <- ts_obs$Date >= as.POSIXct("1999-01-01", tz = "UTC") &
    12:   ts_obs$Date <= as.POSIXct("2009-12-30", tz = "UTC")
    13: ts_obs <- ts_obs[is_per, ]
    14:
    15: # Data processing for GR4J (without snow module)
680 16: prep_snow_n <- PrepGR(DatesR    = ts_obs$Date,
    17:                       Precip    = ts_obs$Ptot,
    18:                       PotEvap   = ts_obs$Evap,
    19:                       Qobs      = ts_obs$Qmmd,
    20:                       HydroModel = "GR4J",
685 21:                       CemaNeige  = FALSE)
    22:
    23: # Data processing for GR4J with snow module
    24: prep_snow_y <- PrepGR(DatesR    = ts_obs$Date,
    25:                       Precip    = ts_obs$Ptot,
690 26:                       PotEvap   = ts_obs$Evap,
    27:                       Qobs      = ts_obs$Qmmd,
    28:                       TempMean  = ts_obs$Temp,
    29:                       ZInputs = median(hypso),
    30:                       HypsoData = hypso,
695 31:                       NLayers = 5,
    32:                       HydroModel = "GR4J",
    33:                       CemaNeige  = TRUE)
    34:
    35: # Calibration using NSE score (without snow module)
700 36: cal_snow_n <- CalGR(PrepGR  = prep_snow_n,
```

```
37:                     CalCrit = "NSE",
38:                     WupPer  = c("1999-01-01", "2000-12-31"),
39:                     CalPer  = c("2001-01-01", "2009-12-30"),
40:                     verbose = TRUE)
41:
42: # Calibration using NSE score (with snow module)
43: cal_snow_y <- CalGR(PrepGR  = prep_snow_y,
44:                     CalCrit = "NSE",
45:                     WupPer  = c("1999-01-01", "2000-12-31"),
46:                     CalPer  = c("2001-01-01", "2009-12-30"),
47:                     verbose = TRUE)
48:
49: # Combination of observed and simulated streamflow
50: tab_cal <- data.frame(Date        = cal_snow_n$OutputsModel$DatesR,
51:                       QOobs       = cal_snow_n$Qobs,
52:                       Qsim_snow_n = cal_snow_n$OutputsModel$Qsim,
53:                       Qsim_snow_y = cal_snow_y$OutputsModel$Qsim)
54:
55: # Computation of regime streamflow
56: tab_cal_reg <- SeriesAggreg(tab_cal,
57:                             Format = "%m",
58:                             ConvertFun = rep("mean", ncol(tab_cal) - 1))
59:
60: # Graphical comparison between simulated and observed streamflow regimes
61: col_snow <- c("black", rep("orangered", 2))
62: lty_snow <- c(1, 1:2)
63: matplot(y = tab_cal_reg[, grep("^Q", colnames(tab_cal))],
64:         xlab = "time [months]", ylab = "flow [mm/day]",
65:         type = "l", lty = lty_snow, lwd = 2, col = col_snow)
66: legend("topright",
67:        legend = c("Qobs", "Qsim without snow mod.", "Qsim with snow mod."),
68:        lty = lty_snow, lwd = 2, col = col_snow)
```

D8 Using flow transformation in objective functions

```
 1: # Catchment data loading
 2: data("B222001001", package = "airGRdatasets")
 3: ts_obs <- B222001001$TS
 4:
 5: # Data processing for GR4J (with Q for calibration)
 6: prep <- PrepGR(DatesR     = ts_obs$Date,
 7:                Precip     = ts_obs$Ptot,
 8:                PotEvap    = ts_obs$Evap,
 9:                Qobs       = ts_obs$Qmmd,
10:                HydroModel = "GR4J",
11:                CemaNeige  = FALSE)
12:
13: # Calibration using NSE score on raw Q
14: cal_raw <- CalGR(PrepGR  = prep,
15:                  CalCrit = "NSE",
16:                  transfo = "",
17:                  WupPer  = c("1999-01-01", "2001-12-31"),
18:                  CalPer  = c("2002-01-01", "2016-12-31"))
19:
20: # Calibration using NSE score on sqrt(Q)
21: cal_sqrt <- CalGR(PrepGR  = prep,
22:                   CalCrit = "NSE",
23:                   transfo = "sqrt",
24:                   WupPer  = c("1999-01-01", "2001-12-31"),
25:                   CalPer  = c("2002-01-01", "2016-12-31"))
26:
27: # Calibration using NSE score on log(Q)
28: cal_log <- CalGR(PrepGR  = prep,
29:                  CalCrit = "NSE",
30:                  transfo = "log",
31:                  WupPer  = c("1999-01-01", "2001-12-31"),
32:                  CalPer  = c("2002-01-01", "2016-12-31"))
33:
34: # Combination of simulated streamflow
35: tab_sim_trsf <- data.frame(Date     = cal_raw$OutputsModel$DatesR,
36:                            QSIM_rawQ = cal_raw$OutputsModel$Qsim,
```

```
37:                              QSIM_sqrtQ = cal_sqrt$OutputsModel$Qsim,
38:                              QSIM_logQ  = cal_log$OutputsModel$Qsim)
39: tab_sim_trsf <- merge(x = ts_obs[, c("Date", "Qmmd")],
40:                       y = tab_sim_trsf,
41:                       by = "Date",
42:                       all.y = TRUE)
43:
44: # Computation of regime streamflow
45: tab_sim_reg <- SeriesAggreg(tab_sim_trsf,
46:                             Format = "%m",
47:                             ConvertFun = rep("mean", ncol(tab_sim_trsf) - 1))
48:
49: # Graphical comparison between simulated and observed streamflow regimes
50: col_trsf <- c("black", rep("orangered", 3))
51: lty_trsf <- c(1, 1:3)
52: matplot(y = tab_sim_reg[, -1],
53:         xlab = "time [months]", ylab = "flow [mm/day]",
54:         type = "l", lty = lty_trsf, lwd = 2, col = col_trsf)
55: legend("bottomleft",
56:        legend = c("Qobs", "Qsim", "sqrt(Qsim)", "log(Qsim)"),
57:        lty = lty_trsf, lwd = 2, col = col_trsf)
```

D9   Using different objective functions

```r
1: # Calibration using NSE score on Q
2: cal_nse <- CalGR(PrepGR  = prep,
3:                  CalCrit = "NSE",
4:                  transfo = "",
5:                  WupPer  = c("1999-01-01", "2001-12-31"),
6:                  CalPer  = c("2002-01-01", "2016-12-31"))
7:
8: # Calibration using KGE score on Q
9: cal_kge <- CalGR(PrepGR  = prep,
10:                  CalCrit = "KGE",
11:                  transfo = "",
12:                  WupPer  = c("1999-01-01", "2001-12-31"),
13:                  CalPer  = c("2002-01-01", "2016-12-31"))
14:
15: # Combination of observed and simulated streamflow
16: tab_crit <- data.frame(Date     = as.POSIXct(cal_nse$OutputsModel$DatesR),
17:                        Qobs     = cal_nse$Qobs,
18:                        Qsim_nse = cal_nse$OutputsModel$Qsim,
19:                        Qsim_kge = cal_kge$OutputsModel$Qsim)
20:
21: # Graphical comparison
22: col_crit <- c("black", rep("orangered", 2))
23: lty_crit <- c(1, 1:2)
24: matplot(x = tab_crit$Date, y = tab_crit[, -1],
25:         xlab = "time [days]", ylab = "flow [mm/day]",
26:         type = "l", lty = lty_crit, lwd = 2, col = col_crit,
27:         xlim = as.POSIXct(x = c("2004-01-01", "2004-03-01"), tz = "UTC"))
28: legend("topleft",
29:        legend = c("Qobs", "Qsim NSE", "Qsim KGE"),
30:        lty = lty_crit, lwd = 2, col = col_crit)
```

D10 Split-sample test

```r
1: # Calibration and evaluation sub-periods
2: per1_wup <- c("1999-01-01", "2001-12-31")
3: per1_sim <- c("2002-01-01", "2008-12-31")
4: per2_wup <- c("2009-01-01", "2011-12-31")
5: per2_sim <- c("2012-01-01", "2018-12-31")
6:
7: # Calibration on per1 and per2
8: cal_per1 <- CalGR(PrepGR = prep,
9:                    CalCrit = "KGE",
10:                   transfo = "",
11:                   WupPer  = per1_wup,
12:                   CalPer  = per1_sim,
13:                   verbose = TRUE)
14: cal_per2 <- CalGR(PrepGR = prep,
15:                   CalCrit = "KGE",
16:                   transfo = "",
17:                   WupPer  = per2_wup,
18:                   CalPer  = per2_sim,
19:                   verbose = TRUE)
20:
21: # Get parameter values at the end of the calibration step
22: param_per1 <- GetParam(cal_per1)
23: param_per2 <- GetParam(cal_per2)
24:
25: # Get criteria values at the end of the calibration step
26: crit_cal_per1 <- GetCrit(cal_per1)
27: crit_cal_per2 <- GetCrit(cal_per2)
28:
29: # Evaluation over per1 and per2
30: eva_per1 <- SimGR(PrepGR = prep,
31:                   Param   = param_per2,
32:                   WupPer  = per1_wup,
33:                   SimPer  = per1_sim,
34:                   EffCrit = "KGE",
35:                   verbose = TRUE)
36:
```

```
37: eva_per2 <- SimGR(PrepGR  = prep,
38:                    Param   = param_per1,
39:                    WupPer  = per2_wup,
40:                    SimPer  = per2_sim,
41:                    EffCrit = "KGE",
42:                    verbose = TRUE)
43:
44: # Get criteria values
45: crit_eva_per1 <- GetCrit(eva_per1)
46: crit_eva_per2 <- GetCrit(eva_per2)
47:
48: # Cleveland dot plot of the criteria
49: dotchart(c(crit_eva_per1, crit_cal_per1, crit_eva_per2, crit_cal_per2),
50:          labels = c("eva (per1)", "cal (per1)", "eva (per2)", "cal (per2)"),
51:          groups = rep(1:2, each = 2),
52:          col = rep(c("darkred", "darkblue"), each = 2), pch = 19,
53:          xlab = "KGE [-]")
```

D11 Differential split-sample test

```
885   1: # Estimation of annual aridity index (PE/P)
      2: ts_obs_y <- SeriesAggreg(x = ts_obs[, c("Date", "Ptot", "Evap")],
      3:                               Format = "%Y",
      4:                               ConvertFun = c("sum", "sum"),
      5:                               YearFirstMonth = 10)
890   6: ts_obs_y$Arid <- ts_obs_y$Evap / ts_obs_y$Ptot
      7:
      8: # Identification of wetter and dryer hydrological years
      9: barplot(height = ts_obs_y$Arid,
     10:         names.arg = format(ts_obs_y$Date, format = "%Y"),
895  11:         xlab = "time [years]", ylab = "aridity index [-]",
     12:         col = "royalblue")
     13:
     14: # Wet and dry periods
     15: per_wet <- c("2016-10-01", "2017-09-30")
900  16: per_dry <- c("2000-10-01", "2001-09-30")
     17:
     18: # Calibration over the wet and the dry periods
     19: cal_wet <- CalGR(PrepGR  = prep,
     20:                  CalCrit = "KGE",
905  21:                  CalPer  = per_wet,
     22:                  verbose = TRUE)
     23: cal_dry <- CalGR(PrepGR  = prep,
     24:                  CalCrit = "KGE",
     25:                  CalPer  = per_dry,
910  26:                  verbose = TRUE)
     27:
     28: # Get parameter values at the end of the calibration step
     29: param_dry <- GetParam(cal_dry)
     30: param_wet <- GetParam(cal_wet)
915  31:
     32: # Get criteria values at the end of the calibration step
     33: crit_cal_dry <- GetCrit(cal_dry)
     34: crit_cal_wet <- GetCrit(cal_wet)
     35:
920  36: # Evaluation over the wet and the dry periods
```

```
37: eva_wet <- SimGR(PrepGR  = prep,
38:                  Param   = cal_dry,
39:                  SimPer  = per_wet,
40:                  EffCrit = "KGE",
41:                  verbose = TRUE)
42: eva_dry <- SimGR(PrepGR  = prep,
43:                  Param   = cal_wet,
44:                  SimPer  = per_dry,
45:                  EffCrit = "KGE",
46:                  verbose = TRUE)
47:
48: # Get criteria values
49: crit_eva_dry <- GetCrit(eva_dry)
50: crit_eva_wet <- GetCrit(eva_wet)
51:
52: # Cleveland dot plot of the criteria
53: dotchart(c(crit_eva_dry, crit_cal_dry, crit_eva_wet, crit_cal_wet),
54:          labels =  c("eva (dry)", "cal (dry)", "eva (wet)", "cal (wet)"),
55:          col = rep(c("darkorange", "deepskyblue3"), each = 2), pch = 19,
56:          xlab = "KGE [-]")
```

Author contributions. OD, PB, and GT conceptualized the work; All authors contributed to the `airGRteaching` package development (LC implemented a first version of the GUI; OD created and maintains the package (features added, GUI improved, and wrote documentation and vignettes); PB coded the model diagram graph and wrote the vignettes containing the exercises; GT beta-tested the package and provided documentation and code improvements); OD, PB, and GT drafted the manuscript; all authors reviewed and edited the manuscript.

945

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

Acknowledgements. The authors would like to thank Météo-France (https://www.data.gouv.fr/en/organizations/meteo-france/) and the SCHAPI (https://hydro.eaufrance.fr/) for providing the SAFRAN meteorological series and the streamflow series included in the `airGRdatasets` package.

References

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: catchment attributes and meteorology for large-sample studies, Hydrology and Earth System Sciences, 21, 5293–5313, https://doi.org/10.5194/hess-21-5293-2017, 2017.

AghaKouchak, A. and Habib, E.: Application of a conceptual hydrologic model in teaching hydrologic processes, International Journal of Engineering Education, 26, 963–973, 2010.

AghaKouchak, A., Nakhjiri, N., and Habib, E.: An educational model for ensemble streamflow simulation and uncertainty analysis, Hydrology and Earth System Sciences, 17, 445–452, 2013.

Astagneau, P. C., Thirel, G., Delaigue, O., Guillaume, J. H. A., Parajka, J., Brauer, C. C., Viglione, A., Buytaert, W., and Beven, K. J.: Technical note: Hydrology modelling R packages – a unified analysis of models and practicalities from a user perspective, Hydrology and Earth System Sciences, 25, 3937–3973, https://doi.org/10.5194/hess-25-3937-2021, 2021.

Baahmed, D., Oudin, L., and Errih, M.: Current runoff variations in the Macta catchment (Algeria): is climate the sole factor? [Le facteur climatique est-il la seule cause des modifications actuelles de l'écoulement dans le bassin versant de la Macta (Algérie ?], Hydrological Sciences Journal, 60, 1331–1339, https://doi.org/10.1080/02626667.2014.975708, 2015.

Belarbi, H., Touaibia, B., Boumechra, N., Amiar, S., and Baghli, N.: Drought and modification of the rainfall-runoff relation: case of Wadi Sebdou basin (western Algeria) [Sécheresse et modification de la relation pluie–débit: cas du bassin versant de l'Oued Sebdou (Algérie Occidentale)], Hydrological Sciences Journal, 62, 124–136, https://doi.org/10.1080/02626667.2015.1112394, 2017.

Bezak, N., Jemec Auflič, M., and Mikoš, M.: Application of hydrological modelling for temporal prediction of rainfall-induced shallow landslides, Landslides, https://doi.org/10.1007/s10346-019-01169-9, 2019.

Blöschl, G., Bierkens, M. F. P., Chambel, A., Cudennec, C., Destouni, G., Fiori, A., Kirchner, J. W., McDonnell, J. J., Savenije, H. H. G., Sivapalan, M., Stumpp, C., Toth, E., Volpi, E., Carr, G., Lupton, C., Salinas, J., Széles, B., Viglione, A., Aksoy, H., Allen, S. T., Amin, A., Andréassian, V., Arheimer, B., Aryal, S. K., Baker, V., Bardsley, E., Barendrecht, M. H., Bartosova, A., Batelaan, O., Berghuijs, W. R., Beven, K., Blume, T., Bogaard, T., Borges de Amorim, P., Böttcher, M. E., Boulet, G., Breinl, K., Brilly, M., Brocca, L., Buytaert, W., Castellarin, A., Castelletti, A., Chen, X., Chen, Y., Chen, Y., Chifflard, P., Claps, P., Clark, M. P., Collins, A. L., Croke, B., Dathe, A., David, P. C., de Barros, F. P. J., de Rooij, G., Di Baldassarre, G., Driscoll, J. M., Duethmann, D., Dwivedi, R., Eris, E., Farmer, W. H., Feiccabrino, J., Ferguson, G., Ferrari, E., Ferraris, S., Fersch, B., Finger, D., Foglia, L., Fowler, K., Gartsman, B., Gascoin, S., Gaume, E., Gelfan, A., Geris, J., Gharari, S., Gleeson, T., Glendell, M., Bevacqua, A. G., González-Dugo, M. P., Grimaldi, S., Gupta, A. B., Guse, B., Han, D., Hannah, D., Harpold, A., Haun, S., Heal, K., Helfricht, K., Herrnegger, M., Hipsey, M., Hlaváčiková, H., Hohmann, C., Holko, L., Hopkinson, C., Hrachowitz, M., Illangasekare, T. H., Inam, A., Innocente, C., Istanbulluoglu, E., Jarihani, B., Kalantari, Z., Kalvans, A., Khanal, S., Khatami, S., Kiesel, J., Kirkby, M., Knoben, W., Kochanek, K., Kohnová, S., Kolechkina, A., Krause, S., Kreamer, D., Kreibich, H., Kunstmann, H., Lange, H., Liberato, M. L. R., Lindquist, E., Link, T., Liu, J., Loucks, D. P., Luce, C., Mahé, G., Makarieva, O., Malard, J., Mashtayeva, S., Maskey, S., Mas-Pla, J., Mavrova-Guirguinova, M., Mazzoleni, M., Mernild, S., Misstear, B. D., Montanari, A., Müller-Thomy, H., Nabizadeh, A., Nardi, F., Neale, C., Nesterova, N., Nurtaev, B., Odongo, V. O., Panda, S., Pande, S., Pang, Z., Papacharalampous, G., Perrin, C., Pfister, L., Pimentel, R., Polo, M. J., Post, D., Sierra, C. P., Ramos, M.-H., Renner, M., Reynolds, J. E., Ridolfi, E., Rigon, R., Riva, M., Robertson, D. E., Rosso, R., Roy, T., Sá, J. H. M., Salvadori, G., Sandells, M., Schaefli, B., Schumann, A., Scolobig, A., Seibert, J., Servat, E., Shafiei, M., Sharma, A., Sidibe, M., Sidle, R. C., Skaugen, T., Smith, H., Spiessl, S. M., Stein, L., Steinsland, I., Strasser, U., Su, B., Szolgay, J., Tarboton, D., Tauro, F., Thirel, G., Tian, F., Tong, R., Tussupova, K., Tyralis, H., Uijlenhoet, R., van Beek, R., van der Ent, R. J., van der Ploeg, M., Van Loon, A. F., van Meerveld, I., van Nooijen, R., van Oel,

P. R., Vidal, J.-P., von Freyberg, J., Vorogushyn, S., Wachniew, P., Wade, A. J., Ward, P., Westerberg, I. K., White, C., Wood, E. F., Woods, R., Xu, Z., Yilmaz, K. K., and Zhang, Y.: Twenty-three unsolved problems in hydrology (UPH) – a community perspective, Hydrological Sciences Journal, 64, 1141–1158, https://doi.org/10.1080/02626667.2019.1620507, publisher: Taylor & Francis, 2019.

990 Brigode, P., Lilas, D., Andréassian, V., Nicolle, P., Le Moine, N., Perrin, C., Gremminger, S., and Augeard, B.: Une cartographie de l'écoulement des rivières de Corse, La Houille Blanche, pp. 68–77, https://doi.org/10.1051/lhb/2019009, 2019.

Brigode, P., Génot, B., Lobligeois, F., and Delaigue, O.: Summary sheets of watershed-scale hydroclimatic observed data for France, https://doi.org/10.15454/UV01P1, 2020.

Burt, T. and Butcher, D.: Stimulation from simulation? A teaching model of hillslope hydrology for use on microcomputers, Journal of 995 Geography in Higher Education, 10, 23–39, https://doi.org/10.1080/03098268608708953, 1986.

Carriba Demange, L., Chanoual, A., and Gazull, A.: Evaluation des logiciels, modèles et packages disponibles pour l'enseignement de la modélisation hydrologique, Projet d'ingénierie GE5, Polytech Nice Sophia, Université Côte d'Azur, 2022.

Cassagnole, M., Ramos, M.-H., Zalachori, I., Thirel, G., Garçon, R., Gailhard, J., and Ouillon, T.: Impact of the quality of hydrological forecasts on the management and revenue of hydroelectric reservoirs – a conceptual approach, Hydrology and Earth System Sciences, 25, 1000 1033–1052, https://doi.org/10.5194/hess-25-1033-2021, 2021.

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., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B.: shiny: Web 1005 Application Framework for R, https://CRAN.R-project.org/package=shiny, R package version 1.7.2, 2022.

Chauveau, M., Chazot, S., Perrin, C., Bourgin, P.-Y., Sauquet, E., Vidal, J.-P., Rouchy, N., Martin, E., David, J., Norotte, T., Maugis, P., and De Lacaze, X.: Quels impacts des changements climatiques sur les eaux de surface en France à l'horizon 2070 ?, La Houille Blanche, pp. 5–15, https://doi.org/10.1051/lhb/2013027, 2013.

Clark, M. P., Kavetski, D., and Fenicia, F.: Pursuing the method of multiple working hypotheses for hydrological modeling, Water Resources 1010 Research, 47, https://doi.org/10.1029/2010WR009827, 2011.

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., Dorchies, D., Perrin, C., and Michel, C.: airGR: Suite of GR Hydrological Models for Precipitation-Runoff Modelling, https://doi.org/10.15454/EX11NA, R package version 1.7.0, 2022.

1015 de Lavenne, A., Andréassian, V., Thirel, G., Ramos, M.-H., and Perrin, C.: A Regularization Approach to Improve the Sequential Calibration of a Semidistributed Hydrological Model, Water Resources Research, 55, 8821–8839, https://doi.org/10.1029/2018WR024266, 2019.

Delaigue, O., Thirel, G., Coron, L., and Brigode, P.: airGR and airGRteaching: Two Open-Source Tools for Rainfall-Runoff Modeling and Teaching Hydrology, in: HIC 2018. 13th International Conference on Hydroinformatics, edited by La Loggia, G., Freni, G., Puleo, V., and De Marchis, M., vol. 3 of *EPiC Series in Engineering*, pp. 541–548, EasyChair, https://doi.org/10.29007/qsqj, 2018.

1020 Delaigue, O., Brigode, P., Andréassian, V., Perrin, C., Etchevers, P., Soubeyroux, J.-M., Janet, B., and Addor, N.: CAMELS-FR: A large sample hydroclimatic dataset for France to explore hydrological diversity and support model benchmarking, https://hal.inrae.fr/hal-03687235, 2022.

Delaigue, O., Brigode, P., and Thirel, G.: airGRdatasets: Hydro-Meteorological Catchments Datasets for the 'airGR' Packages, https://doi.org/10.57745/3SPJ4B, R package version 0.2.1, 2023a.

1025 Delaigue, O., Coron, L., Brigode, P., and Thirel, G.: airGRteaching: Teaching Hydrological Modelling with GR (Shiny Interface Included), https://doi.org/10.15454/W0SSKT, R package version 0.3.2, 2023b.

Desclaux, T., Lemonnier, H., Genthon, P., Soulard, B., and Gendre, R. L.: Suitability of a lumped rainfall–runoff model for flashy tropical watersheds in New Caledonia, Hydrological Sciences Journal, 63, 1689–1706, https://doi.org/10.1080/02626667.2018.1523613, 2018.

Dorchies, D., Thirel, G., Jay-Allemand, M., Chauveau, M., Dehay, F., Bourgin, P.-Y., Perrin, C., Jost, C., Rizzoli, J.-L., Demerliac, S., and
1030 Thépot, R.: Climate change impacts on multi-objective reservoir management: case study on the Seine River basin, France, International Journal of River Basin Management, 12, 265–283, https://doi.org/10.1080/15715124.2013.865636, 2014.

Dorchies, D., Delaigue, O., and Thirel, G.: airGRiwrm: 'airGR' Integrated Water Resource Management, https://doi.org/10.15454/3CVD1I, r package version 0.6.1, 2022.

Elshorbagy, A.: Learner-centered approach to teaching watershed hydrology using system dynamics, International Journal of Engineering
1035 Education, 21, 1203, http://www.hydropyramids.com/files/IJEE1612-final.pdf, 2005.

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.

Fiering, M. B.: Streamflow Synthesis, Harvard University Press, Cambridge, Mass., 1967.

Fuka, D., Walter, M., Archibald, J., Steenhuis, T., and Easton, Z.: EcoHydRology: A Community Modeling Foundation for Eco-Hydrology,
1040 https://CRAN.R-project.org/package=EcoHydRology, R package version 0.4.12.1, 2018.

Furusho, C., Perrin, C., Viatgé, J., Lamblin, R., and Andréassian, V.: Synergies entre acteurs opérationnels et scientifiques au service de l'amélioration de la prévision des crues, La Houille Blanche, pp. 5–10, https://doi.org/10.1051/lhb/2016033, 2016.

García Hernández, J., Paredes Arquiola, J., Foehn, A., Roquier, B., and Fluixá-Sanmartín, J.: RS MINERVE–technical manual v2. 17, Tech. rep., RS MINERVE Group, Sion, Switzerland, 2019.

1045 GEBCO Bathymetric Compilation Group 2021: The GEBCO_2021 Grid - a continuous terrain model of the global oceans and land., https://doi.org/10.5285/c6612cbe-50b3-0cff-e053-6c86abc09f8f, 2021.

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.

Hall, C. A., Saia, S. M., Popp, A. L., Dogulu, N., Schymanski, S. J., Drost, N., van Emmerik, T., and Hut, R.: A hydrologist's guide to open
1050 science, Hydrology and Earth System Sciences, 26, 647–664, https://doi.org/10.5194/hess-26-647-2022, 2022.

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, https://doi.org/10.1002/2016WR019285, 2016.

Irving, K., Kuemmerlen, M., Kiesel, J., Kakouei, K., Domisch, S., and Jähnig, S. C.: A high-resolution streamflow and hydrological metrics dataset for ecological modeling using a regression model, Scientific Data, 5, https://doi.org/10.1038/sdata.2018.224, 2018.

1055 Kay, D., Kay, N., and McDonald, A.: Teaching Catchment Hydrology: Two Dynamic Models for Classroom Use, Teaching Geography, 7, 118–124, https://www.jstor.org/stable/23751886, 1982.

Kirkby, M. and Naden, P.: The use of simulation models in teaching geomorphology and hydrology, Journal of Geography in Higher Education, 12, 31–49, https://doi.org/10.1080/03098268808709023, 1988.

Klemeš, V.: Operational testing of hydrological simulation models, Hydrological Sciences Journal, 31, 13,
1060 https://doi.org/10.1080/02626668609491024, 1986.

Knoben, W. J. M. and Spieler, D.: Teaching hydrological modelling: illustrating model structure uncertainty with a ready-to-use computational exercise, Hydrology and Earth System Sciences, 26, 3299–3314, https://doi.org/10.5194/hess-26-3299-2022, 2022.

Kouassi, A., Koffi, Y., Kouame, K., Lasm, T., and Biemi, J.: Modeling of annual flows using a conceptual model and an artificial neural network model in the N'zi-Bandama watershed (Côte d'Ivoire), Agris On-line Papers in Economics and Informatics, 2, 2082–2094, https://doi.org/10.6088/ijes.00202030092, 2012.

Le Moine, N.: Le bassin versant de surface vu par le souterrain : une voie d'amélioration des performances et du réalisme des modèles pluie-débit ?, Ph.D. thesis, Université Pierre et Marie Curie, Paris 6, http://webgr.inrae.fr/wp-content/uploads/2012/07/2008-LE_MOINE-THESE.pdf, 2008.

Lehner, B. and Grill, G.: Global river hydrography and network routing: baseline data and new approaches to study the world's large river systems, Hydrological Processes, 27, 2171–2186, https://doi.org/10.1002/hyp.9740, 2013.

Marchane, A., Tramblay, Y., Hanich, L., Ruelland, D., and Jarlan, L.: Climate change impacts on surface water resources in the Rheraya catchment (High Atlas, Morocco), Hydrological Sciences Journal, 62, 979–995, https://doi.org/10.1080/02626667.2017.1283042, 2017.

Marshall, J. A., Castillo, A. J., and Cardenas, M. B.: The Effect of Modeling and Visualization Resources on Student Understanding of Physical Hydrology, Journal of Geoscience Education, 63, 127–139, https://doi.org/10.5408/14-057.1, 2015.

Martel, J.-L., Demeester, K., Brissette, F., Poulin, A., and Arsenault, R.: HMETS - A simple and efficient hydrology model for teaching hydrological modelling, flow forecasting and climate change impacts, International Journal of Engineering Education, 33, 1307–1316, https://www.ijee.ie/contents/c330417.html, 2017.

Mathevet, T.: Quels modèles pluie-débit globaux au pas de temps horaire ? Développements empiriques et comparaison de modèles sur un large échantillon de bassins versants, Ph.D. thesis, ENGREF, Paris, http://webgr.inrae.fr/wp-content/uploads/2012/07/2005-MATHEVET-THESE.pdf, 2005.

MATLAB: 9.7.0.1190202 (R2019b), The MathWorks Inc., Natick, Massachusetts, 2018.

McConnell, S.: Code Complete, 2nd ed., Providence, R. I., Cambridge, Mass., interfaces edn., 2004.

Mendez, M. and Calvo-Valverde, L.: Development of the HBV-TEC Hydrological Model, Procedia Engineering, 154, 1116–1123, https://doi.org/10.1016/j.proeng.2016.07.521, 2016.

Merwade, V. and Ruddell, B. L.: Moving university hydrology education forward with community-based geoinformatics, data and modeling resources, Hydrology and Earth System Sciences, 16, 2393–2404, https://doi.org/10.5194/hess-16-2393-2012, 2012.

Michel, C.: How to use single-parameter conceptual model in hydrology?, La Houille Blanche, 69, 39–44, https://doi.org/10.1051/lhb/1983004, 1983.

Michel, C.: Hydrologie appliquée aux petits bassins ruraux, Cemagref, Antony, 1991.

Microsoft Corporation: Microsoft Excel, https://office.microsoft.com/excel, 2019.

Mouelhi, S.: Vers une chaîne cohérente de modèles pluie-débit conceptuels globaux aux pas de temps pluriannuel, annuel, mensuel et journalier, Ph.D. thesis, Paris, ENGREF, http://webgr.inrae.fr/wp-content/uploads/2012/07/2003-MOUELHI-THESE.pdf, 2003.

Mouelhi, S., Michel, C., Perrin, C., and Andréassian, V.: Linking stream flow to rainfall at the annual time step: The Manabe bucket model revisited, Journal of Hydrology, 328, 283–296, https://doi.org/10.1016/j.jhydrol.2005.12.022, 2006a.

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, 2006b.

Nash, J. E. and Sutcliffe, J. V.: 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.

Neumann, J. L., Arnal, L., Emerton, R. E., Griffith, H., Hyslop, S., Theofanidi, S., and Cloke, H. L.: Can seasonal hydrological forecasts inform local decisions and actions? A decision-making activity, Geoscience Communication, 1, 35–57, https://doi.org/10.5194/gc-1-35-2018, 2018.

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., Andréassian, V., Mathevet, T., Perrin, C., and Michel, C.: Dynamic averaging of rainfall-runoff model simulations from complementary model parameterizations, Water Resources Research, 42, https://doi.org/10.1029/2005WR004636, 2006.

Paquet, E., Garavaglia, F., Garçon, R., and Gailhard, J.: The SCHADEX method: A semi-continuous rainfall–runoff simulation for extreme flood estimation, Journal of Hydrology, 495, 23–37, https://doi.org/10.1016/j.jhydrol.2013.04.045, 2013.

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.

Piazzi, G. and Delaigue, O.: airGRdatassim: Suite of Tools to Perform Ensemble-Based Data Assimilation in GR Hydrological Models., https://doi.org/10.15454/WEYYVZ, R package version 0.1.3, 2021.

Piazzi, G., Thirel, G., Perrin, C., and Delaigue, O.: Sequential Data Assimilation for Streamflow Forecasting: Assessing the Sensitivity to Uncertainties and Updated Variables of a Conceptual Hydrological Model at Basin Scale, Water Resources Research, 57, https://doi.org/10.1029/2020WR028390, 2021.

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.

Pérez-Sánchez, J., Senent-Aparicio, J., and Jimeno-Sáez, P.: The application of spreadsheets for teaching hydrological modeling and climate change impacts on streamflow, Computer Applications in Engineering Education, https://doi.org/10.1002/cae.22541, 2022.

R Core Team: R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/, 2017.

Ramos, M. H., van Andel, S. J., and Pappenberger, F.: Do probabilistic forecasts lead to better decisions?, Hydrology and Earth System Sciences, 17, 2219–2232, https://doi.org/10.5194/hess-17-2219-2013, 2013.

Riboust, P., Thirel, G., Moine, N. L., and Ribstein, P.: Revisiting a Simple Degree-Day Model for Integrating Satellite Data: Implementation of Swe-Sca Hystereses, Journal of Hydrology and Hydromechanics, 67, 70–81, https://doi.org/10.2478/johh-2018-0004, 2019.

Richmond, B., Aspinwall, D., Vescuso, P., Peterson, S., and High Performance Systems, Inc.: STELLA, High Performance, Lyme, N.H., OCLC: 14639320, 1985.

Roux, Q. and Brigode, P.: How long would we have to wait before (re)filling the Malpasset dam reservoir? An example of a teaching project done using R and airGR modeling packages, 2018.

Sanchez, C. A., Ruddell, B. L., Schiesser, R., and Merwade, V.: Enhancing the T-shaped learning profile when teaching hydrology using data, modeling, and visualization activities, Hydrology and Earth System Sciences, 20, 1289–1299, https://doi.org/10.5194/hess-20-1289-2016, 2016.

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, 2018.

Seibert, J. and Vis, M. J. P.: Teaching hydrological modeling with a user-friendly catchment-runoff-model software package, Hydrology and Earth System Sciences, 16, 3315–3325, https://doi.org/10.5194/hess-16-3315-2012, 2012.

Seibert, J., Uhlenbrook, S., and Wagener, T.: Preface "Hydrology education in a changing world", Hydrology and Earth System Sciences, 17, 1393–1399, https://doi.org/10.5194/hess-17-1393-2013, 2013.

Shmueli, G.: To Explain or to Predict?, Statistical Science, 25, 289–310, https://doi.org/10.1214/10-STS330, publisher: Institute of Mathematical Statistics, 2010.

Slater, L. J., Thirel, G., Harrigan, S., Delaigue, O., Hurley, A., Khouakhi, A., Prodoscimi, I., Vitolo, C., and Smith, K.: Using R in hydrology: a review of recent developments and future directions, Hydrology and Earth System Sciences Discussions, pp. 1–33, https://doi.org/10.5194/hess-2019-50, 2019.

Tarboton, D., Idaszak, R., Horsburgh, J., Heard, J., Ames, D., Goodall, J., Band, L., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., and Maidment, D.: HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing, International Congress on Environmental Modelling and Software, https://scholarsarchive.byu.edu/iemssconference/2014/Stream-A/7, 2014.

Toum, E., Masiokas, M. H., Villalba, R., Pitte, P., and Ruiz, L.: The HBV.IANIGLA Hydrological Model, The R Journal, 13, 378–395, https://journal.r-project.org/archive/2021/RJ-2021-059/index.html, 2021.

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, pp. 1176–1187, https://doi.org/10.1016/j.jhydrol.2014.04.058, 2014.

Vanderkam, D., Allaire, J., Owen, J., Gromer, D., and Thieurmel, B.: dygraphs: Interface to 'Dygraphs' Interactive Time Series Charting Library, https://CRAN.R-project.org/package=dygraphs, R package version 1.1.1.6, 2018.

Vidal, J., Martin, E., Franchistéguy, L., Baillon, M., and Soubeyroux, J.: 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/Semi-Distributed Hydrological Model for Education Purposes, https://CRAN.R-project.org/package=TUWmodel, R package version 1.1-1, 2020.

Wagener, T. and McIntyre, N.: Tools for teaching hydrological and environmental modeling, Computers in Education Journal, 17, 16–26, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.493.5729&rep=rep1&type=pdf, 2007.

Wi, S., Ray, P., Demaria, E. M. C., Steinschneider, S., and Brown, C.: A user-friendly software package for VIC hydrologic model development, Environmental Modelling and Software, 98, 35–53, https://doi.org/10.1016/j.envsoft.2017.09.006, 2017.

Zimmerman, W. B. J.: Multiphysics Modeling with Finite Element Methods, vol. 18 of *Series on Stability, Vibration and Control of Systems, Series A*, WORLD SCIENTIFIC, https://doi.org/10.1142/6141, 2006.

Zipper, S., Albers, S., and Prosdocimi, I.: CRAN Task View: Hydrological Data and Modeling, https://cran.r-project.org/view=Hydrology, https://cran.r-project.org/view=Hydrology, 2020.