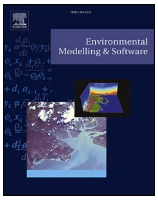
Environmental Modelling and Software 168 (2023) 105808

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



The PAVICS-Hydro platform: A virtual laboratory for hydroclimatic modelling and forecasting over North America

Richard Arsenault[a,*], David Huard[b], Jean-Luc Martel[a], Magali Troin[a,c], Juliane Mai[d], François Brissette[a], Christian Jauvin[b], Long Vu[b], James R. Craig[d], Trevor J. Smith[b], Travis Logan[b], Bryan A. Tolson[d], Ming Han[d], Francis Gravel[a], Sébastien Langlois[e]

[a] Hydrology, Climate and Climate Change Laboratory, École de technologie supérieure, Université du Québec, 1100 Notre-Dame Street West, Montreal, Quebec, H3C 1K3, Canada
[b] Ouranos, 550 Sherbrooke W., West Tower, 19th floor, Montreal, Quebec, Canada, H3A 1B9
[c] Hydroclimat, TVT, Maison du Numérique et de l'Innovation, place Georges Pompidou, 83 000, Toulon, France
[d] Department of Civil and Environmental Engineering, University of Waterloo, 200 University Ave W, Waterloo, Ontario, N2L 3G1, Canada
[e] Hydro-Québec, 75, René-Lévesque W., Montréal, QC, H2Z 1A4, Canada

ARTICLE INFO

Handling Editor: Daniel P Ames

Keywords:
Ravenpy
PAVICS-Hydro
Hydrological modeling
Flow forecasting
Climate change impacts
Open access
Scientific research platform

ABSTRACT

This paper presents the PAVICS-Hydro platform, the result of long-term strategic considerations and compromises between functionality and ease-of-use, which is developed to provide users with direct access to a comprehensive hydroclimatic modeling and analysis framework over North America. This includes a Python hydrological modelling package, efficient access to meteorological data, a large array of post-processed hydroclimatic data, numerical weather forecasts and bias-corrected CMIP6 climate projections at the scale of 5797 catchments, thus providing the basic components necessary for hydrological modeling and forecasting, as well as climate change impact studies. This paper presents the principles on which the PAVICS-Hydro platform was developed while explaining the choices behind its key features, with a specific focus on lessons learned for creators of the next generations' scientific research platforms. A case study using PAVICS-Hydro is given to illustrate how this platform can be used to support different needs in the field of water resources.

Software and data availability

PAVICS-Hydro is a set of software packages developed to add hydrological modelling capabilities to PAVICS, an existing platform for climate data analysis available at https://pavics.ouranos.ca/. Users may register a PAVICS account for free with the understanding that the platform be used for scientific purposes. Development of PAVICS started in 2015 and provides climate model data and processing capabilities to users. PAVICS-Hydro, which began services in 2019, created or made significant contributions to the packages shown in Table 1.

RavenPy is a Python wrapper for the Raven hydrological framework, meant to facilitate the configuration of hydrological models, the launch of simulations and retrieval of the results. RavenWPS is a server based on the Open Geospatial Consortium (OGC) Web Processing Service (WPS) Application Programming Interface (API), offering processes to extract watershed properties. Birdy is a Python client providing a user-friendly interface to remote WPS services. Jupyter notebooks (Fangohr et al., 2021) included with the RavenPy documentation show how to use these software libraries together to run typical hydrological modeling workflows. A support forum (Craig, 2023) is available for Raven, while for the other libraries, GitHub Issues and Pull requests are the best way to ask questions and propose enhancements.

Raven, RavenPy and Birdy can be run on a laptop or workstation running Linux, Windows or OSX. RavenWPS is a server component that has been developed and tested on Linux only. Various installation methods are available for these components, including compiling from source, installation with pip or conda. For those who cannot install the package locally for IT or administrative reasons, the PAVICS platform provides a JupyterLab environment where all these and more are available.

While the PAVICS platform itself is too complex to be described here, its components and their orchestration are defined in the birdhouse-deploy repository (Vu et al., 2023a). It currently runs on a 48-core Xeon server with 128 GB of RAM (32+ GB for the THREDDS data

[*] Corresponding author.
E-mail address: richard.arsenault@etsmtl.ca (R. Arsenault).

https://doi.org/10.1016/j.envsoft.2023.105808
Received 4 October 2022; Received in revised form 9 August 2023; Accepted 21 August 2023
Available online 23 August 2023

Table 1
Main software components of the PAVICS-Hydro contribution to PAVICS.

Component	Function	DOI	License
Raven	Hydrological modeling	10.5281/ zenodo.8229854	Artistic 2.0
RavenPy	Python wrapper for Raven	10.5281/ zenodo.7972347	MIT
RavenWPS	WPS server for watershed properties	10.5281/ zenodo.3239468	MIT
Birdy	WPS client	10.5281/ zenodo.7876310	Apache-2.0

server, 8+ GB for GeoServer and the remainder is used for other components and the JupyterLab user accounts), and over 100 TB of storage capacity. Questions regarding the PAVICS platform, including PAVICS-Hydro, should be addressed to pavics@ouranos.ca.

1. Introduction

There has been a growing interest over the last years in developing computational environmental modeling frameworks, with the emergence of multiple hydrology-oriented platforms or systems (e.g., among others Alfieri et al., 2013; Arsenault et al., 2020; Horsburgh et al., 2016; Tague and Band, 2004; Taylor et al., 2021; Werner et al., 2013; Zhang et al., 2016). For instance, specific developments in this field include the provision of web-based applications and services, online geospatial databases, model sharing technology, data analysis platforms, and software as a service including hydrological modeling (Choi et al., 2021; Gan et al., 2020; Khattar et al., 2021; Morsy et al., 2017; Souffront Alcantara et al., 2019; Werner et al., 2013). All these developments have the potential to support hydrological research by facilitating the use of high-end hardware architectures, easing access to large volumes of data, making information retrieval more interactive, and for ensuring the development, repeatability, and reproducibility of experiments while advancing hydrology in a more coherent way (Ceola et al., 2015).

Several hydrology-oriented platforms have recently been developed to support the Findable, Accessible, Interoperable, and Reproducible (FAIR) principles (Wilkinson et al., 2016). Among others, we can cite the HydroShare (Horsburgh et al., 2016), HydroDS (Gichamo et al., 2020), and eWaterCycle (Hut et al., 2022) platforms. All these platforms rely on the data, models, and codes sharing with the aim of enhancing the trustworthiness of research findings, while increasing the reusability of research results. HydroShare provides open access datasets and codes in an easily accessible and organized repository, also providing access through Jupyter notebook environments (e.g., Bandaragoda et al., 2019; Leonard et al., 2019). HydroDS is a web service that facilitates downloading data in a suitable format for hydrological models; the combination of both provides users a gateway to all required data for running hydrological models and performing hydrological analyses. The eWaterCycle platform gives open access to hydrological models of various complexity (conceptual to distributed structures) through a special interface that allows users to add their own models to the platform, written in different programming languages that interact from the Jupyter notebook environment. Also, eWaterCycle has the possibility of incorporating any hydrological model in any programming language through the Basic Modeling Interface (BMI), a modeling framework that allows interoperability between models and data through a common programming interface. The eWaterCycle platform currently includes the PCR-GLOBWB 2.0 (Sutanudjaja et al., 2018), wflow (Schellekens et al., 2020), Hype (Lindström et al., 2010), LISFLOOD (Van Der Knijff et al., 2010), TopoFlex HBV (Gao et al., 2014), MARRMoT (Knoben et al., 2019) and WALRUS (Brauer et al., 2014) models, with upcoming support for GlobWat (Hoogeveen et al., 2015), SUMMA (Clark et al., 2015) and mHM (Samaniego et al) models and modelling frameworks. These platforms provide users with a variety of methods and tools for performing hydrological research and applications, making hydrological sciences more accessible and reproducible. However, some applications and workflows are not currently trivial to implement, such as streamflow forecasting, climate change impact studies or regionalization at ungauged sites, due to the necessity of implementing specific processes and accessing particularly large datasets. For example, HydroShare is a repository for data and codes sharing, and is not in itself a modeling platform, and HydroDS allows generating model inputs but does not provide flexible modeling capabilities. eWaterCycle provides a comprehensive package by including flexibility in hydrological model structure and data inputs in an online environment where users can tailor workflows according to their needs. Choi et al. (2021) showed that it was possible to combine several existing platforms and tools for data access, geospatial processing, and hydrological modelling to provide reliable and reproducible studies. In their work, some example applications are provided, and tools include the ability to simulate streamflow and forecast flows. However, tools pertaining to climate change impact assessment are not included de facto and would have to be integrated from another source.

In Canada, a consortium initiative aims to create an innovative research software platform to streamline climate science research. The Power Analytics and Visualization for Climate Science (PAVICS) platform (https://pavics.ouranos.ca/) is a virtual laboratory first launched in 2015 facilitating the analysis of climate data over North America. It was originally developed jointly by Ouranos, a regional climate adaptation consortium, and the Centre de Recherche Informatique de Montréal, an applied research center on information technology and artificial intelligence. PAVICS provides access to various data collections ranging from observations, reanalyses, numerical weather forecasts and climate projections, in addition to ensuring quality control on the accompanying metadata. It also hosts a JupyterLab Python programming environment with the most up-to-date scientific libraries preinstalled. Acting as a climate data repository and processing suite, it allows researchers to access, process, analyze, and transform data into useable climate information to improve the quality and reproducibility of scientific research studies. Recently, under the umbrella of the PAVICS-Hydro project, services and capabilities were added to the platform with the aim of providing a comprehensive hydrological modeling framework, supporting simplified handling of large hydroclimatic data sets and multiple hydrological models with a minimal technical background from users, while making use of existing infrastructure as proposed by Choi et al. (2021). The development of PAVICS-Hydro was guided by five key considerations:

(1) Regional application: users should be able to run a hydrological model for any watershed in North America.
(2) Open access: all codes, models, and data should be open access, such that users can freely use the tools available on the platform for their research or use-cases.
(3) Interoperability: key functionalities should be made available through a standard Application Programming Interface (API), facilitating interoperability with other platforms and services.
(4) Flexibility: grant access to different models allowing users to set up models in the various hydroclimatological regimes in North America.
(5) Versatility: the same tools should support a wide range of activities, including hydrological model development, flow forecasting, and hydroclimatic projections.

Tools developed within the PAVICS-Hydro project have been in regular use since the initial version in 2019, and have been beneficial to many research areas, with the current registered user base including scientists, educators, governmental policy officers, agriculture industry specialists, engineers, water resources managers, students, and other stakeholders. Example applications utilizing core PAVICS-Hydro functionality include training students in the basics of hydrological modelling, investigating the impacts of climate change on water availability of hydropower systems, estimating flows in rivers for aquatic species resilience and flow forecasting for flood prediction. The sharing of IT infrastructure and tools reduces costs for the development and production of hydrological information, as users can develop their models and methods on the shared PAVICS server. Moreover, compliance with data standards in streamflow scenarios and forecasts promotes the integration of the generated information in many water planning and management decisions.

The PAVICS stack is part of Ouranos' long-term strategy to deliver climate services. It is embedded in multiple projects and used internally by the staff for the development of new climate products. It benefits from funding from the federal and provincial governments, and in-kind contributions from national and international collaborators. It has also become a core tool in the workflow of many of the authors' respective research teams, meaning it will continue to thrive and benefit from contributions of current and future research projects.

The main objective of this paper is to present the PAVICS-Hydro platform and make its capabilities known to the hydrology and environmental research communities. We denote by "PAVICS-Hydro platform" the software components of PAVICS that support hydrological modeling, and interactions with external hydroclimate data repositories. This paper touches on the development process and decisions leading to the platform's design. Lessons learned are presented for the benefit of platform developers.

In the following sections, a short overview of the PAVICS-Hydro platform is first provided (Section 2). Section 3 traces the key development choices of PAVICS-Hydro, and Section 4 presents its most important components and features. Section 5 presents an example application of the platform to display the scope and types of studies that it can tackle. Finally, an outlook on future developments of PAVICS-Hydro appears in Section 6.

2. Overview of the PAVICS-hydro platform

PAVICS-Hydro is an online hydrological modeling and forecasting service that is free of charge, open-source, and open-access. It is developed in the Python programming language and offers users the capacity to perform geospatial analysis, build hydrological models, apply various parameter estimation algorithms, estimate and predict streamflows, and/or evaluate the impacts of climate change on future streamflows via open access to data and PAVICS-Hydro software libraries. Fig. 1 presents a high-level overview of the PAVICS architecture, which includes a THREDDS data server for hydroclimatic data stored in the netCDF format, a GeoServer to store watershed polygons and land-use raster maps, a JupyterLab server, Open Geospatial Consortium (OGC) Web Processing Services (WPS) servers, and middleware responsible for authentication and authorization. The JupyterLab environment comes installed with a large collection of scientific libraries operating on xarray data structures. Novel components or substantial contributions to existing components made through the PAVICS-Hydro project are indicated by a color code.

The contributions of PAVICS-Hydro to the PAVICS architecture allow users to perform hydrological studies within PAVICS. The main contribution of PAVICS-Hydro is the RavenPy Python package, which wraps the Raven hydrological model framework (Craig et al., 2020). RavenPy provides a programmatic mechanism to configure models (setting parameters, options, and hydrological processes), run simulations, and retrieve results. Together, these capabilities allow hydrologists to design multiple experiments (e.g., ensemble forecasting, regionalization), run them in parallel and analyze results, something that would be time consuming to do manually. Actual simulations are performed by the Raven executable, RavenPy playing the role of an interface.

Raven and RavenPy can be installed and run on a laptop. However, setting up a programming environment with multiple scientific libraries can be difficult in some professional settings with IT restrictions. The PAVICS platform provides an online ready-to-use Python programming environment where RavenPy is pre-installed, along with a host of other packages that can be used to perform hydrological modelling workflows, such as bias-correction and time-series analysis, high bandwidth access to large data collection (e.g., ERA5 and CMIP6), hindcast and forecast databases (ECCC real-time and archived forecasts), HydroShare interoperability, and compute services such watershed delineation and spatial subsetting. In addition to this programming environment, PAVICS hosts a GeoServer storing watershed contours and land-use maps

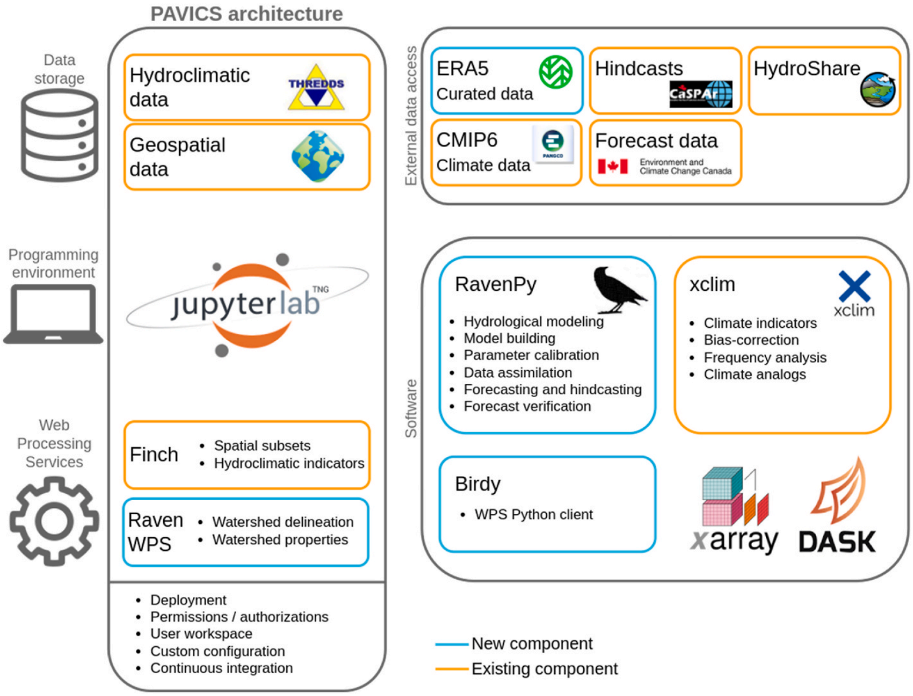


Fig. 1. PAVICS-Hydro environment components within the PAVICS platform, including newly developed packages (blue) and connections to pre-existing components that are made easily accessible through notebooks and code examples (orange).

which can be accessed through the RavenWPS server, a THREDDS server storing bias corrected climate data, observations and reanalyses, as well as multiple WPS climate-related services such as climate indicators, data subsetting and watershed land-use data extraction. It can be seen in Fig. 1 that many connections are also made towards data repositories such as ERA5 reanalysis data, CMIP6 climate model data, and archived and operational weather forecasts.

Tables 2–5 provide a detailed list of features and functionalities available in PAVICS-Hydro for hydrological model simulation and calibration (Table 2), data extraction, processing and geospatial analysis (Table 3), hydrological forecasting (Table 4) and climate change impacts assessments (Table 5). The tables also present the notebooks that include these features, which can be found in the "~/tutorial-notebooks/hydro" folder on the PAVICS-Hydro platform. They are also available on Zenodo within the RavenPy package release (as seen in Table 1).

3. Development of the PAVICS-hydro platform

3.1. Initial design choices

When the project team first proposed the idea of an online hydrological modeling service, a few requirements were obvious. The design of the platform would provide:

- geospatial analysis capabilities to evaluate catchment drainage areas, boundaries, slopes, and other relevant physiographic properties;
- at least one hydrological model, with the possibility to add others over time;
- easy access to meteorological databases and future climate scenarios to feed the hydrological model(s).

These three aspects are the basis of the platform on which other functionalities would eventually build upon. The main design choices are presented here, along with the reasons why these solutions were preferred to alternatives.

3.1.1. Geospatial analysis

Various research communities provide open-source software packages for geospatial analysis that could simply be integrated into PAVICS-Hydro. Python libraries, such as geopandas (Jordahl et al., 2020), shapely (Gillies et al., 2007), pyproj (Snow et al., 2021) and fiona (Gillies and contributors, 2023) were leveraged for processing geospatial data, as they are actively maintained and robust. These, together with GeoServers' built-in analytics provides the foundation for some of the remote data analytics offered on the platform, such as extracting land-use information over watersheds. One missing feature however was an efficient algorithm to average a long time series of gridded data over polygons (e.g., subbasins). Drawing inspiration from existing work done in OCGIS (Koziol et al., 2023), this functionality was contributed to xESMF (Zhuang et al., 2023), a package compatible with the xarray stack that exposes regridding algorithms implemented in the Earth System Modeling Framework (ESMF) Fortran library. The approach leverages a conservative regridding algorithm, which preserves the integral over the surface area between input and output grids. This xESMF. SpatialAverager is now used to convert gridded meteorological datasets into time series of subbasin averages, which simplifies hydrological model setup and speeds up computations.

3.1.2. Hydrological modeling

The core element of the PAVICS-Hydro is its hydrological modeling capability. Ideally, users should have the ability to simulate a wide range of hydroclimatic regimes. One option would have been to pre-install a suite of different hydrological models. Without commonalities regarding model configuration and parametrization, as well as common input and output data formats, this would have rapidly exceeded the developer's capacity to maintain and support users. Other platforms, such as eWaterCycle, have resolved this by way of a common model interface implemented for each hydrological model. This idea, together with a clear distinction between model agnostic and model specific configuration (Knoben et al., 2022), allows a single platform to support multiple independent hydrological models.

Another solution for the standardization of model configurations, inputs and outputs is to adopt a general purpose hydrological framework such as SUMMA (Clark et al., 2015), MARRMoT (Knoben et al., 2019), wflow (Schellekens et al., 2020) and Raven (Craig et al., 2020). Raven is a flexible framework that can emulate a large suite of existing models or create new models by combining hydrological process (e.g., users can select from a series of snow accumulation and melt modules, infiltration methods, evapotranspiration formulas, routing schemes. etc.). Watershed configuration and model options can be shared across different emulators, facilitating multi-model experiments. Furthermore, Raven has a powerful and flexible data processing engine that supports both lumped and semi-distributed hydrological model structures. Input data can be provided as text or binary netCDF files, the latter supporting time series on a grid, at a gauge station, or integrated over an HRU or sub-basin. PAVICS-Hydro only targets and supports netCDF input and output time-series files to improve interoperability and limit development efforts going into model-specific workflows (Knoben et al., 2022). Therefore, the Raven modelling framework was selected for this platform. Other advantages of Raven are that the simulation time-step is easily adjustable, from minutes up to one day, and input time series are resampled automatically to match the model time step. Raven temporally downscales evapotranspiration, incoming energy, and potential melt using hourly incoming radiation estimates. Daily temperatures are converted to sub-daily using a sinusoidal mapping, influencing relative humidity, air pressure and air density. Precipitation and wind speed are uniformly discretized throughout the day. Error-checking routines identify problems with units, missing data, and other inconsistencies before launching the model. Therefore, by using Raven, we only needed to provide data in a single format to benefit from a multitude of hydrological models, flexibility in modelling choices and options (both for lumped and distributed models), and automatic error-checking. Together, those features convinced us to design the hydrological

Table 2
Features and functionalities available in PAVICS-Hydro for hydrological model simulation and training.

Process	Description	Notebook containing example
Hydrological model configuration	Programmatic generation of Raven configuration files based on predefined emulators (see Table 3)	04_Emulating_hydrological_models 05_Advanced_RavenPy_configuration
Hydrological model calibration	Estimation of an optimal parameter set based on an objective function and calibration hyperparameters	06_Raven_calibration
Running a hydrological model	Simulate streamflows based on model configuration and forcing dataset (meteorological or climate, see Tables 4 and 5)	04_Emulating_hydrological_models 05_Advanced_RavenPy_configuration
Streamflow regionalization	Estimate model parameters for ungauged basins using catchment descriptors and model parameters from pre-calibrated models on over 5000 catchments over North America.	Perform_Regionalization
Time-series and indicator analysis	Evaluate statistics of the generated hydrographs, including mean, peak and low flows for custom time periods, frequency analysis, and other such analyses.	time_series_analysis

Table 3
Features and functionalities for model input generation (geospatial analysis and data pre-processing).

Process	Description	Notebook containing example
Watershed delineation	Given latitude and longitude coordinates that point to a North American watershed, returns the watershed polygon containing the coordinates or the entire upstream water basin. Used to delineate watersheds based on small-scale watershed units.	01_Extracting_watershed_boundaries
Raster subsetting	Allows to clip raster files, such as a DEM, according to a watershed polygon shapefile, and return the raster subset.	02_Extract_geographical_watershed_properties
Terrain analysis	Extracts some basic properties of a DEM for a watershed, such as elevation, slope, and aspect.	02_Extract_geographical_watershed_properties
Shape properties	Extracts properties form a watershed polygon, such as area, perimeter, gravelius index and centroid coordinates.	02_Extract_geographical_watershed_properties
Zonal statistics	Allows extracting categorized data from rasters, such as for land cover ratios for a given watershed in North America (uses the North American Land Cover Monitoring System classification).	02_Extract_geographical_watershed_properties
Automated configuration for semi-distributed watersheds	Script to convert GIS layers from the BasinMaker Toolbox (Han et al., 2023) (http://hydrology.uwaterloo.ca/basinmaker/) into subbasin and routing configuration.	Distributed_hydrological_modelling
Extract ERA5 meteorological data	Allows users to efficiently obtain subsetted ERA5 precipitation and temperature data for anywhere over North America.	03_Extracting_forcing_data
Data subsetting, averaging, time shifting and other data processing tools.	Direct access to the python xarray package to process netcdf files and datasets allows users to tailor the data to the specific needs for their use-case.	03_Extracting_forcing_data 08_Getting_and_bias_correcting_CMIP6_data
Hydroshare Integration	Access HydroShare using the user's sign-in key to explore catalogs, gather data and integrate data into workflows.	HydroShare_integration

modeling capability around Raven, benefiting from its large feature set and active user community.

However, Raven has some drawbacks, such as the fact that it is designed as an executable to be run through a command line interface (CLI), rather than a library that can be imported and whose individual components executed on demand by external code. In Raven, processes, simulation parameters and input files are specified in text-based configuration files parsed by the executable, and simulation outputs are written to an output file on disk. In situations where multiple simulations are run using the same input data, such as model calibration, this unnecessary input/output (I/O) hampers performance. Work is underway to expose Raven components through the BMI interface, which could eventually allow on-demand execution of individual model components, in addition to making Raven compatible with the eWaterCycle platform.

To integrate the Raven modeling framework into an existing climate analytic programming environment written in Python, a Python wrapper for Raven was developed, called RavenPy (see Table 1). RavenPy offers an interface to a subset of Raven's configuration options and writes text-based Raven configuration files ready to be executed. It parses and validates configuration options to catch typos and mistakes before launching the model. It also includes utilities to launch model simulations, retrieve results and expose them as xarray data structures, the latter supporting the concatenation of results from multiple simulations. Together, these functionalities allow users to run a hydroclimatic analysis workflow in an interactive programming environment and avoid having to manually configure Raven for each simulation. The pre-configured hydrological models (emulators) available in RavenPy are detailed in section 4.1. RavenPy is designed with modularity in mind, such that model configuration, model execution and result retrieval can be used independently of the others.

At the start of the PAVICS-Hydro project, Raven was not meant to be exposed directly in a interactive programming environment. The initial plan was rather to host a remote server that would run Raven on behalf of users, in a bid to offload heavier computations to dedicated hardware close to where the climate forcing data was also stored. The initial implementation used a standard Application Programming Interface (API) called Web Processing Services (WPS), and a server backend based on PyWPS and the Birdhouse framework (Ehbrecht et al., 2018). The Python wrapper for Raven would be a layer hidden from users, simplifying the execution of those remote requests by the PyWPS server. This initial design choice proved to be cumbersome and ineffective. Running and debugging complex hydrological model simulations through web requests was unsuited to research work. Also, exposing the large number of model options through the API added complexity and friction in the development and maintenance of the software. Developer experience and user feedback led the team to convert the backend model wrapper into a user-facing Raven interface. This required extensive adaptation, but the RavenPy wrapper was thus developed and offered as a stand-alone library.

To cater to users with limited computing resources, RavenPy is installed on the PAVICS-Hydro platform. The original WPS server lost its hydrological modeling capabilities, but retained services for geospatial data processing. User interaction with WPS servers typically takes the form of HTML requests. To facilitate python-based interactions with WPS servers, we added a high-level Python client to an existing command line interface for WPS services (birdhouse-birdy, see Table 1). On connecting to the WPS server, the client creates a mock Python module, and each process available on the server is translated into a mock Python function within this module. The signature of these functions matches the input and outputs of the remote process. Calling a remote process becomes almost as simple as calling a Python function and relieves the user of writing HTML requests.

3.1.3. Meteorological and climate data access

Users can access meteorological and climate data in one of three ways on the PAVICS-Hydro platform:

1) Use one of the datasets stored on the platform;
2) Upload their own data to the platform in their personal account; or
3) Stream data from external online databases.

As for the basic set of data provided with the platform, we provide access to ERA5 reanalysis data, which is a high-quality dataset with a complete series of climate data over the entire world (0.25° × 0.25° spatial grid) from 1950 until today, in quasi-real time (Hersbach et al., 2020). The netCDF files distributed by ECMWF are however not designed for typical hydrological applications. Indeed, the chunking

Table 4
Features and functionalities available in PAVICS-Hydro for hydrological forecasting.

Process	Description	Notebook containing example
Hot-start file creation	Save final model states after a data-assimilation run or simulation as a hot-start file for the next simulation period.	07_Making_and_using_hotstart_files
Data assimilation	Generate initial hydrological model states in preparation of a forecast using an Ensemble Kalman Filter.	10_Data_assimilation
Extended streamflow prediction (ESP)	Generate streamflow forecasts using historical meteorological forcing data at the forecast Julian day as a proxy for long-term climate forecast probabilities.	11_Climatological_ESP_forecasting
Real-time streamflow forecasting	Generate real-time streamflow forecasts using the Environment and Climate Change Canada (ECCC) weather forecasts made accessible through PAVICS-Hydro using a pre-calibrated Raven model, ideally using the data assimilation implementation to prepare the model for the forecast.	Hydrological_realtime_forecasting
Streamflow hindcasting	Test forecasting methods and techniques on historical forecast data, using an archived ECCC database (CaSPAr).	12_Performing_hindcasting_experiments
Forecast verification	Evaluate the skill of forecasts using multiple common forecast verification metrics over the hindcasting period; compute statistics on the real-time forecasts.	11_Climatological_ESP_forecasting
Flood risk analysis	Predict the probability of flooding based on an ensemble streamflow forecast and a threshold flood value.	Assess_probabilistic_flood_risk

pattern used by ERA5 is designed for applications looking at large spatial areas one time step at a time. Hydrological applications rather look at long time periods over smaller areas. This mismatch can dramatically slow down data access. Tests showed that gathering 30 years of ERA5 climate data for a single medium-sized catchment (2500 km^2) required approximately 2 h of processing time.

To increase processing speed, a number of measures were put in place. First, commercial cloud-based object storage space with high throughput was purchased. Second, ERA5 files were rechunked along the spatial dimension, such that querying long time series is more efficient. Third, data were stored in zarr format and hosted on a public Amazon S3 bucket to take advantage of distributed object storage technology. An intake catalog indexes individual files and quickly and efficiently retrieves the data. Together, these measures make it such that the speed at which data is delivered is limited by internet bandwidth rather than computing power. This data can be accessed through the Jupyter Notebook titled "03_Extracting_forcing_data.ipynb" demonstrating ERA5 data access on the PAVICS-Hydro platform as well as in the supplementary information. The same data request now takes 2 s rather than 2 h. This was a breakthrough that was solved by community members that were outside the project team, highlighting the advantages of collaboration in open-source software. Note that this database currently only contains hourly ERA5 precipitation and temperatures covering the globe. There is an ongoing effort to also include ERA5-Land hourly precipitation and temperatures as well as snow-water equivalent (SWE), which should be made available later in 2023. More data could be added in time if an alternate funding source can cover the long-term data hosting costs. These datasets are continuously updated once per week with the available data as they become available from the official ERA5 database.

Table 5
Features and functionalities available in PAVICS-Hydro for climate change impact studies.

Process	Description	Notebook containing example
Extract CMIP6 climate model data	Subset CMIP6 climate model data anywhere in the world. Uses Pangeo catalogs to query data in the cloud.	08_Getting_and_bias_correcting_CMIP6_data
Bias-correction	Bias-correct climate model data using a reference dataset such as ERA5. Uses the xclim Python package to expose many bias-correction algorithms.	08_Getting_and_bias_correcting_CMIP6_data
Simulating impacts of climate change	Built-in support for climate model input data, including non-standard calendars.	09_Hydrological_impacts_of_climate_change

It is important to note that running a very large (e.g., continental-sized) watershed in semi-distributed mode over long periods with distributed ERA5 meteorological data could overwhelm the PAVICS system memory in certain cases. In those cases, users should consider running their model locally using the RavenPy package, but can also gather the ERA5 data they need from the platform independently. For example, Raven can run in distributed mode over a 275,000 km^2 watershed and be calibrated with 10,000 model evaluations without any problems on a standard computer (Brown and Craig, 2020), but larger catchments or models remain untested. The ERA5 database is also accessible to users outside of PAVICS-Hydro and is used by many research groups throughout the world to ensure quick and efficient access to this data.

Another option that has been included in PAVICS-Hydro is the possibility of getting data directly from the HydroShare database. This allows for a greater level of interoperability and allows users to build workflows with archived data that makes workflows more repeatable. An example workflow for HydroShare access using Notebooks is available in the "HydroShare_Integration.ipynb" notebook.

3.2. *User interaction: the struggle between user-friendliness and complexity*

A crucial design element of an online scientific platform is the method by which users will interact with the software. Many platforms opt for easy-to-use graphical user interfaces (GUIs), which typically widens access to less code-savvy scientists, with mouse-clicks, checkboxes, and option buttons. This was first considered for PAVICS-Hydro; however, discussions with user groups showed that more experienced users disliked the fact that GUIs make batch operations more difficult, or even impossible. Furthermore, reproducing research work based on interaction with a GUI is a process that can be inconsistent due to the required human intervention. Finally, GUI development requires considerable software engineering resources, including expensive front-end expertise. Given a finite budget, investing in a front-end necessarily reduces the number of scientific applications that can be developed and maintained. On the opposite end of the spectrum, some platforms use APIs to gather requests and return results directly, without any GUI. Users must learn all the options through the documentation. This makes the learning curve very steep, but gives complete control over the operations to users, including batch-processing and duplicating results easily.

For PAVICS-Hydro, a compromise between user-friendliness and the ability to customize workflows, reproduce methodologies and maximize efficiency was found in the adoption of Jupyter Notebooks, which can be seen as an intermediate solution between a GUI and pure API or code access. They can contain widgets that make certain graphical interactions possible, such as displaying and interacting with figures, maps, and simple menu items. Multiple interacting widgets may be combined to build dashboards, and the underlying notebook may be hosted on the platform as a stand-alone web page, providing a low-cost solution to prototype graphical application interfaces. JupyterLab and the use of Jupyter Notebooks were found to provide excellent properties for complex use cases and power-users, as they can build upon existing codes and be customized as needed, while they allow users to also perform batch calculations for larger-scale studies. Some examples of this flexibility are given in the Jupyter Notebook "Managing_Jupyter_-Environments.ipynb" on the platform and summarized here:

- Containerized JupyterLab environments means users can install Python packages in their instance by pip/mamba, including other hydrological models coded in Python;
- Deleting/shutting down your server instance will reset the environment but preserve all files, such that breaking the environment is not permanent;
- The exact environment setup can be exported to a file such that it is possible to replicate it exactly if there are ever upgrades, updates or other modifications that break user code.
- Data is preserved permanently in the user account and can be downloaded. Storage availability is based on a good faith approach to resources management. Users that request disproportionate amounts of storage or computing power for long periods of time will be contacted by platform managers to find alternative solutions.
- Unlimited use of the available computing power when launching simulations, including no timeout or wall-time for simulations, except for synchronous WPS geoprocessing requests that have a timeout of 4 min. Of course, the more users are present, the fewer resources are available and requests can be queued in case of very busy periods.

This solution is also more flexible than GUIs, in which users are constrained to using the available control options. This is a useful feature for researchers that use the platform for developing new methods and tools, which would be difficult or expensive to maintain on a GUI-based platform.

Example notebooks are included in the documentation as tutorials and are tested nightly on the same environment running on the Jupyter server to ensure that the regular updates and feature additions to the platform are backwards-compatible with existing Notebooks. This integration testing adds to the suite of unit tests running automatically with each code change. The testing environment can be found at this link (Vu et al., 2023b).

3.3. Other lessons learned

The initial thrust of the project was to provide hydrological modeling capability through WPS services. The interface to the Raven modeling framework was then a backend component, entirely hidden from end-users. This interface consisted in a mechanism where the parameters given through WPS requests were plugged into model configuration text templates. Because the WPS interface limits user interactions, some additional logic in the backend was necessary to infer configuration information that would be too tedious to ask of users.

As the project advanced and we realized that running model simulations from a remote server through WPS requests made experimentation more difficult, not less, we decided to bring the Raven interface closer to users, and made the Raven framework interface available as a standalone library (RavenPy). Code previously in the backend was exposed to end users, but without major refactoring to avoid breaking WPS processes depending on it. Because the backend was not initially designed to be exposed, its interface was not intuitive for existing Raven users. In addition, the hidden logic that was necessary in the backend now created additional difficulties due to the much wider range of use-cases it had to handle. Ultimately, we entirely refactored RavenPy with the view that it would primarily be used by Raven users, rather than as a cog in the backend. This considerably simplifies the code base and will hopefully make further maintenance much easier.

In the same vein, we initially wanted to host all data internally, propose custom codes for all features we wished to expose to users, and generally control as much of the environment as possible. However, we eventually realized that it's preferable to develop modular components that end users are responsible for putting together, rather than trying to maintain solutions for every combination of data, model and configuration. We removed code perceived as too specific, and increasingly relied on other packages to provide functionality. This allowed us to focus our attention on fewer components and spread the maintenance effort over a larger developer community. In retrospect, the pressure to reach development milestones might have contributed to feature creep. Hopefully this experience will also help future platform developers avoid similar pitfalls.

4. PAVICS-hydro features and structure

In the following section, a few features of the PAVICS-Hydro platform are presented. These are also demonstrated in Jupyter Notebooks that provide step-by-step workflows to perform specific research and/or data processing tasks and should be consulted for more details (see Tables 2–5). Notebooks cover a wide range of possible applications, including general hydrological modeling, streamflow forecasting, and climate change impact studies on hydrology over North American catchments.

4.1. General hydrological modeling

General hydrological modeling requires the extraction of hydroclimatic data, geographical catchment properties (latitude, longitude, area, mean altitude, land use, etc.) and catchment boundaries before setting-up a hydrological model. All this information can be obtained using remote services through PAVICS-Hydro. These services connect to a digital elevation model (DEM) and a land-use dataset to extract relevant information for the studied catchment. The DEM used is the EarthEnv-DEM90 (Robinson et al., 2014) while the land-use dataset is the North American Land Change Monitoring System (Latifovic et al., 2016). Other data sources could also be used depending on their availability as the platform allows the use of other packages or databases according to user specifications, as long as these are accessible online. Catchments whose boundaries are to be defined make use of the worldwide and most detailed level of the HydroSHEDS database to provide drainage basin contours (Lehner et al., 2008), using the closest subbasin to the pour point as the basin outlet. A GeoJSON file with the catchment boundaries is then made available for download and useable for other modeling steps, such as extracting meteorological input data. This meteorological dataset can be extracted within the catchment's boundaries or in the surrounding region after specifying the required spatial and temporal extents.

The implementation of the hydrological model is done through the Raven modeling framework. Eight hydrological models are available in a lumped or distributed mode: GR4J (Perrin et al., 2003), HBV (Bergström, 1995), HMETS (Martel et al., 2017), HYPR (Ahmed et al., 2020), MOHYSE (Fortin and Turcotte, 2007), SAC-SMA (Burnash et al., 1973), Canadian Shield (an in-house model that represents the Canadian Shield hydrology particularly well), and the Blended Model (Mai et al., 2020a). While Raven does not explicitly "contain" these models, it includes processes that allow it to emulate them either perfectly or almost perfectly. Therefore, running a Raven implementation of HMETS or GR4J is essentially the same thing as running the original source code for those models. One drawback, however, is that our implementations makes some assumptions (GR4JCN implements the CemaNeige snow module automatically with a single altitude band for example). The model parameters can be calibrated using spotpy (Houska et al., 2015). Ravenpy sets defaults for the lower and upper boundaries of the parameter search space, the optimization methods (using the default option of the Dynamically Dimensioned Search algorithm; Tolson and Shoemaker, 2007) and the maximum number of model evaluations. At the end of the model calibration step, the best parameter set as well as the score of the objective Nash-Sutcliffe Efficiency (or other user-defined functions) are provided, and users can analyze the calibration results.

General statistics of the streamflow time-series can be computed in addition to flow indicators (e.g., base flow index) and specific analyses of times series (e.g., frequency analysis). A regionalization package in RavenPy is available to provide estimated streamflow (best estimate and ensemble) at an ungauged site using three pre-calibrated hydrological models (GR4JCN, HMETS, and MOHYSE) and a large hydrometeorological database with catchment attributes and observed streamflow (Extended CANOPEX; Arsenault et al., 2016). Multiple regionalization strategies are proposed to estimate flows at the ungauged sites, as described in Table 6 (Arsenault et al., 2015).

4.2. *Streamflow forecasting*

PAVICS-Hydro supports data assimilation of streamflow to prepare the model states of a forecast by wrapping the assimilation tools available in Raven. The Ensemble Kalman Filter (EnKF) data assimilation method is used for the initialization of the hydrological model states. This also helps improve the forecast skill for shorter-term forecasts (i.e., up to a few days lead-time), and in some instances, can also improve longer-term forecasts. In the data assimilation notebook (see Table 4), two simulations are performed to demonstrate the ability of the data assimilation method to improve the model states and reduce initial biases: 1) an open-loop simulation without data assimilation, and 2) simulation with EnKF data assimilation. Users can define the experimental conditions for the assimilation (i.e., the state variables to be assimilated), the distribution and shape of random perturbations, the number of members, the assimilation period and frequency. Assimilated states can be used as a starting point for hydrological forecasts, and performance can be validated using a hindcasting dataset.

A climatological Extended Streamflow Prediction (ESP; Day, 1985) forecast can also be performed by using historical weather as a proxy for future weather. The general idea is to initialize the state of the hydrological model to represent current conditions, but instead of using weather forecasts to predict future flows, the hydrological model is run with observed historical weather series from past years. For instance, 30 years of weather observations give 30 different forecasts. The accuracy and reliability of the forecast ensemble are then evaluated by different probabilistic metrics, such as the CRPS and the rank histogram metric using the climpred package (Brady and Spring 2021). A climatology ESP hindcast can also be performed over multiple years to assess the forecast skill. Other streamflow forecasts can be done through PAVICS-Hydro, such as real-time streamflow forecasts with the Environment and Climate Change Canada (ECCC) weather forecasts and streamflow hindcasts using the Canadian Surface Prediction Archive (CaSPAr; Mai et al., 2020b) archived weather forecasts.

The risk of flooding can be estimated from a threshold extracted from a frequency analysis of observed data. To do so, the forecasting abilities and the time series analysis capabilities are combined in a single seamless process to estimate the flood risk of a probabilistic forecast. A frequency analysis is first performed on an observed time series, then the streamflow associated to a given return period is estimated. A climatological ESP forecast is then performed, and the probability of flooding (exceeding the threshold) is estimated given the ensemble of members in the probabilistic forecast.

4.3. *Hydrological impact studies*

PAVICS-Hydro provides a method to gather climate simulations quickly and efficiently from the Coupled Model Intercomparison Project Phase 6 (CMIP6; O'Neill et al., 2016) with as little user-input as possible through the Pangeo catalog (Pangeo Team, 2023). It allows the automatic extraction of the climate model simulations in the CMIP6 experiment as defined by the watershed boundaries. The users need to choose the climate variables, the shared socioeconomic pathway (SSP) scenarios, the future time periods of interest as well as the model member.

A Jupyter Notebook is available to guide the users on how to conduct

Table 6
Description of the regionalization methods available in PAVICS-Hydro.

Name of the regionalization method	Description
Multiple linear regression	Ungauged catchment parameters are estimated individually by a linear regression against catchment properties.
Spatial proximity	The ungauged hydrograph is an average of the *n* closest catchments' hydrographs.
Physical similarity	The ungauged hydrograph is an average of the *n* most similar catchments' hydrographs.
Spatial proximity with inverse distance weighting	The ungauged hydrograph is an average of the *n* closest catchments' hydrographs, and the average is weighted using inverse distance weighting.
Physical similarity with inverse distance weighting	The ungauged hydrograph is an average of the *n* most similar catchments' hydrographs, and the average is weighted using inverse distance weighting.
Spatial proximity with inverse distance weighting and regression-based augmentation	The ungauged hydrograph is an average of the *n* closest catchments' hydrographs, and the average is weighted using inverse distance weighting. The method uses the Extended CANOPEX dataset to estimate model parameters using multiple linear regression. Parameters whose regression r^2 is higher than 0.5 are replaced by the multiple linear regression estimated value (Arsenault and Brissette, 2014).
Physical similarity with inverse distance weighting and regression-based augmentation	The ungauged hydrograph is an average of the *n* most similar catchments' hydrographs, and the average is weighted using inverse distance weighting. The method uses the CANOPEX dataset to estimate model parameters using multiple linear regression, as for the spatial proximity variation.

bias correction of climate model simulations that will be fed as inputs to the hydrological model in order to perform climate change impact studies on hydrology. PAVICS-Hydro supports various methods to bias-correct CMIP6 climate simulations, such as quantile mapping, detrended quantile mapping, multivariate quantile mapping bias correction algorithm (MBCn; Cannon, 2018) and a bias-adjustment method based on the principles of Principal Components Analysis (PCA; Hnilica et al., 2017). The bias correction methods are proposed through the "xclim" Python package (Smith et al., 2023), which provides tools and utilities for hydro-climate indicator calculations and bias-adjustment algorithms. Users need to extract the historical period and future period data of the climate model to perform the bias correction method using the provided templates. The reference data used as the baseline for the bias correction is the ERA5 reanalysis, but users can easily substitute their own dataset. Then, correction relationships between the reference dataset (ERA5) and historical dataset (climate model simulations for the same time period) are found. The correction relationships obtained are then applied to both reference and future climate simulations to correct them. Finally, the hydrological model can be run with the bias-corrected climate data for both user-defined reference and future periods to generate streamflows and evaluate the impacts of climate change on the hydrology of the catchment of interest.

4.4. Adaptability, customization and performance

While PAVICS-Hydro provides a set of tools that are ready-to-use for many use-cases and applications, it also allows users to modify codes and processes for custom workflows. This requires users to install RavenPy locally using the developer installation method. Installing the software in developer mode can be done by following the instructions available on the RavenPy readthedocs documentation (Huard et al., 2023). This allows users to modify the RavenPy package and add, modify and customize codes that can be useful, such as defining new emulators, implementing specific algorithms for optimization or data assimilation, adding new forecasting datasets with particular features, regionalization options, and so forth. Such changes can also be submitted on the project GitHub for possible integration into a future RavenPy release, which would then be available on the RavenPy installation on the server. Furthermore, users can install other Python packages (as demonstrated in the "Managing_Jupyter_Envrionments.ipynb" notebook), including other Python-based models, which can be integrated into workflows for data processing and output analysis. As long as these packages are available on the Python Package Index (PyPI) or Conda, they are also accessible to the server and to users. Finally, another method for adding hydrological models is to modify the Raven source code directly as described in Craig et al. (2020).

In terms of performance, results will depend on the hardware running Raven and on the server load when running on the online portal. Nonetheless, the platform can handle distributed hydrological modelling well, even for medium to large size catchments. One limitation is that of basin size when invoking the land-use properties WPS service. Due to the resolution of the underlying dataset, catchments with sizes above 100,000 km^2 will display an error message if attempting to submit such a WPS request. Another limitation to be aware of is that of model calibration, which sees some small performance hits due to all of the overhead related to file I/O required by Raven, but that is usually negligible for larger models and/or longer simulations, where I/O becomes less significant. We provide an analysis of simulation times for some common examples in Table 7.

We can see that the runtime is affected by two components: the I/O process and the actual model runtime. These examples are for a daily timestep and would scale linearly according to the timestep. For single model simulations, the I/O can be a significant portion of the runtime. This can be seen in the scaling of the lumped GR4J simulation, which shows that for short simulations, the runtime can be as high as 0.18 s per year of simulation, whereas the same model run on a 30 year period has a runtime of [3.27s/30 =] 0.11 s per year of simulation. Thus, the more years of simulation, the lower the impact of the I/O overhead. In terms of calibration, it can be seen that the runtime shows the same patterns, but since the I/O (new configuration files with updated parameter values as input, objective function as output) must be rewritten at each model evaluation, the runtime increases linearly with the number of model evaluations. Note that the model outputs are minimized in calibration such that only the objective function is written to disk. Hydrographs and model states are not written in these cases as they are not necessary for the calibration procedure. Finally, simulation time for the distributed model shows longer runtimes than the lumped model, which is expected, and also shows the same I/O overhead effect. Here, a single-year simulation has a runtime of 1.24s per simulation year, and a 30-year simulation takes [15.53s/30 =] 0.52s per simulation year.

5. Applications

To illustrate the use of PAVICS-Hydro, an example application is briefly presented. It illustrates the ability of PAVICS-Hydro to simply and efficiently setup a hydrological model and all required data and calibrate the model to obtain calibrated parameter sets. It also extracts and bias-corrects climate model data to allow climate change impact assessments on the catchment's hydrology. The Jupyter Notebook and required data are available either on the PAVICS-Hydro platform (in the "paper" folder when logging in), or in the official release of RavenPy as shown in Table 1. This notebook can be uploaded to the PAVICS-Hydro server into any user account and run to obtain the results shown in this paper. The example case study is applied to the Mistassini catchment in Canada as shown in Fig. 2. All data were taken from the HYSETS database (Arsenault et al., 2020).

The notebook first presents the geographical extraction tools and applies them to the Mistassini catchment. The code returns catchment properties, such as land-use fractions, area, slope, elevation, aspect, and other characteristics. These can be used to set up the hydrological model or for any other research use. The platform lets users download the extracted datasets in raster format, but only the basin-averaged values are computed in this example for simplicity. The required weather data to drive the model are then extracted using a call to the ERA5 reanalysis dataset we store on Wasabi, a data hosting service running on Amazon S3. Data is spatially averaged over the catchment and saved to the NetCDF format for the Raven hydrological modeling framework to be able to access it. This means the data is also downloadable locally for users that only want to extract data quickly for their projects. Data for climate change impact assessments are then extracted. To do so, the ERA5 dataset is used as the reference dataset, and the MIROC6 climate model using the SSP5-8.5 scenario is selected for the climate change aspect. The notebook presents a boilerplate code to extract the climate model data from the Pangeo catalog on Google Cloud which can be easily adapted to use another climate model. Data for the reference

Table 7
Run times of common hydrological model simulations and calibrations as a function of model complexity and number of simulation years.

Simulation duration (years)	Lumped GR4JCN model			Distributed GR4JCN model (176 HRUs)
	Simulation time (s)	Calibration time - 100 evaluations (s)	Calibration time - 1000 evaluations (s)	Simulation time (s)
1	0.18	–	–	1.24
5	0.59	13.84	145.61	3.41
10	1.08	17.51	180.10	5.62
20	2.15	25.25	274.17	11.49
30	3.27	33.28	340.15	15.53

period (1981–1990) and future period (2081–2090) are extracted and spatially averaged over the catchment using the “clisops” subsetting tools. These shorter periods were used purely for demonstration purposes and to keep computation times low. Actual research should use more years, ideally 30-year windows to capture climatology (Navarro-Racines et al., 2020). All data are processed to convert units to those used by Raven. The “xclim” post-processing package is used to bias-correct the climate model data with a detrended quantile mapping approach. This is one of many available methods in “xclim” that is also integrated into PAVICS-Hydro; therefore, users can adapt the code to their use-case. Finally, processed datasets are saved to the NetCDF format for both the reference and future periods.

The next step is to set-up the hydrological model. RavenPy allows emulating eight hydrological models, and for this case study, the GR4J hydrological model coupled with the CemaNeige snow routine (Valéry et al., 2014), henceforth called GR4JCN, is selected. GR4JCN is calibrated on half of the reference period using the spotpy calibration package and ERA5 data as observations, and validated on the other half of available data. The results for the validation period are presented in Fig. 3. The model underestimates the flow during the onset of spring snowmelt but overestimates the winter low flow. The timing of all peak flows is adequately captured, returning an overall Nash-Sutcliffe efficiency score of 0.49. However, some aspects, such as the representation of the peak flow magnitude, could be improved. In this application, a limited number of model evaluations (200) was used in the calibration to keep computing times short for demonstration purposes. Increasing the number of model evaluations will help improve the models’ skill.

The final calibrated model parameters and catchment properties are extracted and can be saved by the user for future tasks. In this example, the next steps after the model creation and calibration are to run the model with both the bias-corrected reference and future climate model datasets to evaluate changes in future hydrographs. The resulting hydrographs can be compared for both periods in terms of peak, median or low-flow indicators, for example. The mean annual hydrograph for both periods are presented in Fig. 4, along with the validation period and observed flows. It can be seen that the future period hydrograph has an earlier melt season, a dryer spring season, and a wetter fall season compared to the reference period, but that the reference and validation flows are similar to the observations, as expected.

Users can download the generated streamflow files and figures, or they can simply analyze the data directly on the PAVICS-Hydro platform by using one of the many available statistical analysis tools made available to users.

6. Discussion and conclusion

The PAVICS-Hydro platform is an open-source, user-friendly, flexible, and powerful research tool for hydrology and hydroclimatology that is free to use for anyone, and operational at https://pavics.ouranos.ca. It allows users to simplify many aspects of their research and to develop training materials that can be easily shared and replicated. One of its advantages is to grant professionals access to a programming environment that is continually updated with the latest version of scientific libraries. This is especially useful when office policies and IT staff control which pieces of software can and cannot be installed on individual computers.

The complexities involved in developing and maintaining an active platform are, however, not to be underestimated. One early mistake was to package hydrological services behind a WPS API while they were still in heavy development. For one, client-server interactions bring their own set of bugs, independent from the service logic. Also, debugging and testing the services through all the different software layers adds friction and complexity that slows the development of the actual scientific content. For example, for security reasons, servers typically do not display the full stack traceback when failures occur, making bugs more difficult to diagnose. We eventually decided to focus our WPS interface efforts on simple operations that absolutely required server interactions, such as extracting watershed properties from databases hosted on the server. Complex operations, such as hydrological model calibration, simulations, and analyses, would run directly in the Jupyter Notebook engine, instead of being offered through the WPS API as initially envisaged.

Conversely, one early decision that proved beneficial was to insist on using netCDF as the standard data format, and the CF-Convention as the metadata standard. Although this required conversion efforts, enforcing data and metadata standards allows for greater interoperability, and reduces the amount of boilerplate code needed to write complex scientific workflows, as suggested by Miles (2014) and Knoben et al. (2022).

This paper attempts to shed some light on the decision choices and challenges faced during the platform development and provides guidance for the next generation of scientific research platforms. This, hopefully, will help other developers in the design and implementation of similar platforms, as more and more online research portals come online. One of the most important decisions that shaped the platform into its current form was the selection of an interaction mechanism. We opted for Jupyter Notebooks, as a compromise between fully-fledged GUIs and powerful but unwieldy API strategies, and we are seeing positive feedback both from novice and power users. Developers of platforms in similar situations should consider these compromises carefully during the design phase and ensure that they have the ability, time and resources to develop, adapt and maintain similar dependencies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

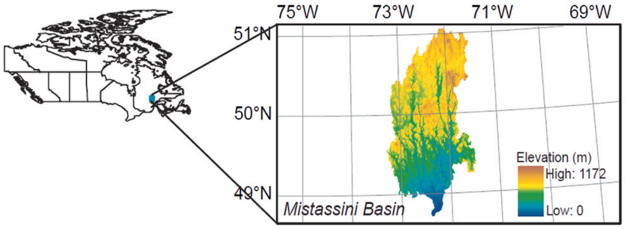


Fig. 2. Location map of the Mistassini catchment (Modified from Troin et al., 2016).

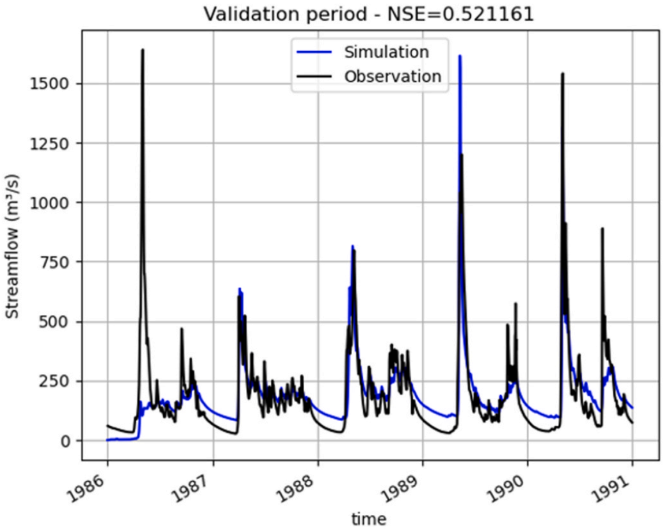


Fig. 3. Validation period hydrographs for the observed and simulated streamflow for the Mistassini catchment in this case study.

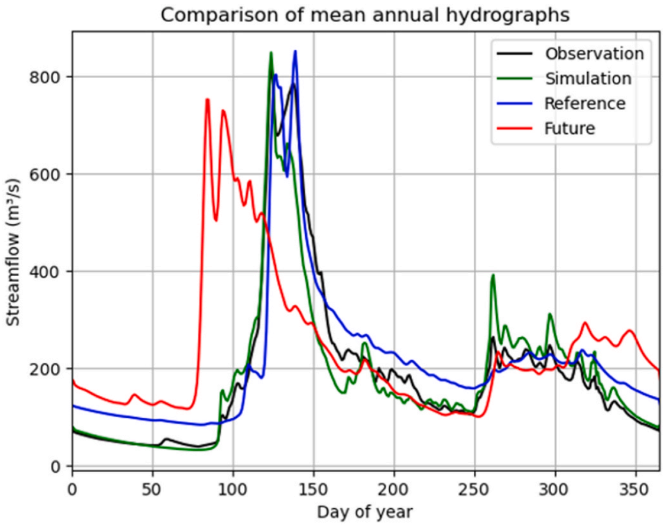


Fig. 4. Mean annual hydrograph for the observed (1981–1990), validation (1986–1990), reference (1981–1990), and future (2081–2090) periods of the Mistassini catchment in this case study.

the work reported in this paper.

Data availability

All data and codes are shared at the links in the data availability statement.

Acknowledgements

This work was supported by CANARIE inc., a Canadian funding agency that funded the development and maintenance of PAVICS-Hydro. The authors would also like to thank the developers of the original PAVICS platform at Ouranos for making the platform possible and for sharing the computing infrastructure. Many thanks to the Pangeo project team for making the CMIP6 data efficiently accessible, as well as the climate modeling centers that made their data available in the CMIP6 project. We would also like to thank all the contributors to the platform that provided feature requests, updates, suggestions and packages on the project GitHub and that were eventually integrated into PAVICS-Hydro. Finally, the ERA5 and data were initially downloaded from the Copernicus Climate Data Store, available at: https://climate.copernicus.eu/climate-reanalysis. Note that the dataset we provide contains modified ERA5 reanalysis data from the Copernicus Climate Change Service Information and that neither the European Commission nor ECMWF is responsible for any use that may be made of the Copernicus Information or Data it contains.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2023.105808.

References

Ahmed, M.I., Elshorbagy, A., Pietroniro, A., 2020. Toward simple modeling practices in the complex canadian prairie watersheds. J. Hydrol. Eng. 25 (6), 04020024 https://doi.org/10.1061/(ASCE)HE.1943-5584.0001922.

Alfieri, L., Burek, P., Dutra, E., Krzeminski, B., Muraro, D., Thielen, J., Pappenberger, F., 2013. GloFAS-global ensemble streamflow forecasting and flood early warning. Hydrol. Earth Syst. Sci. 17 (3), 1161–1175. https://doi.org/10.5194/hess-17-1161-2013.

Arsenault, R., Bazile, R., Ouellet Dallaire, C., Brissette, F., 2016. CANOPEX: a Canadian hydrometeorological watershed database. Hydrol. Process. 30 (15), 2734–2736. https://doi.org/10.1002/hyp.10880.

Arsenault, R., Brissette, F., Martel, J.-L., Troin, M., Lévesque, G., Davidson-Chaput, J., Castañeda Gonzalez, M., Ameli, A., Poulin, A., 2020. A comprehensive, multisource database for hydrometeorological modeling of 14,425 North American watersheds. Sci. Data 7 (1), 243. https://doi.org/10.1038/s41597-020-00583-2.

Arsenault, R., Brissette, F.P., 2014. Continuous streamflow prediction in ungauged basins: the effects of equifinality and parameter set selection on uncertainty in regionalization approaches. Water Resour. Res. 50 (7), 6135–6153. https://doi.org/10.1002/2013WR014898.

Arsenault, R., Poissant, D., Brissette, F., 2015. Parameter dimensionality reduction of a conceptual model for streamflow prediction in Canadian, snowmelt dominated ungauged basins. Adv. Water Resour. 85, 27–44. https://doi.org/10.1016/j.advwatres.2015.08.014.

Bandaragoda, C., Castronova, A., Istanbulluoglu, E., Strauch, R., Nudurupati, S.S., Phuong, J., Adams, J.M., Gasparini, N.M., Barnhart, K., Hutton, E.W., Hobley, D.E.J., 2019. Enabling Collaborative Numerical Modeling in Earth Sciences Using Knowledge Infrastructure, vol. 120. Environmental Modelling & Software, 104424.

Bergström, S., 1995. The HBV model. Computer models of watershed hydrology 443–476.

Brady, R.X., Spring, A., 2021. climpred: verification of weather and climate forecasts. J. Open Source Softw. 6 (59), 2781.

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

Brown, G., Craig, J.R., 2020. Multi-gauge calibration of a hydrological model of the Liard river basin. Can. Water Resour. J. 45 (4), 287–303. https://doi.org/10.1080/07011784.2020.1803143.

Burnash, R.J.C., Ferral, R.L., McGuire, R.A., 1973. A Generalized Streamflow Simulation System: Conceptual Modeling for Digital Computers. US Department of Commerce, National Weather Service, and State of California.

Cannon, A.J., 2018. Multivariate quantile mapping bias correction: an N-dimensional probability density function transform for climate model simulations of multiple variables. Clim. Dynam. 50 (1), 31–49. https://doi.org/10.1007/s00382-017-3580-6.

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., Wagener, T., 2015. Virtual laboratories: new opportunities for collaborative water science. Hydrol. Earth Syst. Sci. 19 (4), 2101–2117. https://doi.org/10.5194/hess-19-2101-2015.

Choi, Y.-D., Goodall, J.L., Sadler, J.M., Castronova, A.M., Bennett, A., Li, Z., Nijssen, B., Wang, S., Clark, M.P., Ames, D.P., Horsburgh, J.S., Yi, H., Bandaragoda, C., Seul, M., Hooper, R., Tarboton, D.G., 2021. Toward open and reproducible environmental modeling by integrating online data repositories, computational environments, and model Application Programming Interfaces. Environ. Model. Software 135, 104888. https://doi.org/10.1016/j.envsoft.2020.104888.

Clark, N.P., Nijssen, B., Lundquist, J.D., Kavetski, D., Rupp, D.E., Woods, R.A., Freer, J.E., Gutmann, E.D., Wood, A.W., Brekke, L.D., Arnold, J.R., Gochis, D.J., Rasmussen, R. M., 2015. A unified approach for process-based hydrologic modeling: 1. Modeling concept. Water Resour. Res. 51 (4), 2498–2514. https://doi.org/10.1002/2015WR017198.

Craig, J.R., 2023. Support forum for the Raven hydrological modelling framework. http://www.civil.uwaterloo.ca/raven_forum/. (Accessed 3 August 2023).

Craig, J.R., Brown, G., Chlumsky, R., Jenkinson, R.W., Jost, G., Lee, K., Mai, J., Serrer, M., Sgro, N., Shafii, M., Snowdon, A.P., Tolson, B.A., 2020. Flexible watershed simulation with the Raven hydrological modelling framework. Environ. Model. Software 129, 104728. https://doi.org/10.1016/j.envsoft.2020.104728.

Day, G.N., 1985. Extended streamflow forecasting using NWSRFS. J. Water Resour. Plann. Manag. 111 (2), 157–170. https://doi.org/10.1061/(ASCE)0733-9496(1985)111:2(157).

Ehbrecht, C., Landry, T., Hempelmann, N., Huard, D., Kindermann, S., 2018. Projects based on the web processing service framework birdhouse. Int. Arch. Photogram. Rem. Sens. Spatial Inf. Sci. 43–47. https://doi.org/10.5194/isprs-archives-XLII-4-W8-43-2018. XLII-4/W8.

Fangohr, H., Kluyver, T., DiPierro, M., 2021. Jupyter in computational science. Comput. Sci. Eng. 23 (2), 5–6. https://doi.org/10.1109/MCSE.2021.3059494.

Fortin, V., Turcotte, R., 2007. Le Modèle Hydrologique MOHYSE (in French). Université du Québec à Montréal Note SCA7420, p. 14.

Gan, T., Tarboton, D.G., Dash, P., Gichamo, T.Z., Horsburgh, J.S., 2020. Integrating hydrologic modeling web services with online data sharing to prepare, store, and execute hydrologic models. Environ. Model. Software 130, 104731. https://doi.org/10.1016/j.envsoft.2020.104731.

Gao, H., Hrachowitz, M., Fenicia, F., Gharari, S., Savenije, H.H.G., 2014. Testing the realism of a topography-driven model (FLEX-Topo) in the nested catchments of the Upper Heihe, China, Hydrol. Earth Syst. Sci. 18, 1895–1915. https://doi.org/10.5194/hess-18-1895-2014.

Gichamo, T.Z., Sazib, N.S., Tarboton, D.G., Dash, P., 2020. HydroDS: data services in support of physically based, distributed hydrological models. Environ. Model. Software 125, 104623. https://doi.org/10.1016/j.envsoft.2020.104623.

Gillies, S., Bierbaum, A., Lautaportti, K., Tonnhofer, O., 2007. Shapely: manipulation and analysis of geometric objects. toblerity (Org).

Gillies, S., contributors, 2023. PyPi project for the Fiona python package. https://pypi.org/project/Fiona/. (Accessed 3 August 2023).

Han, M., Shen, H., Tolson, B.A., Craig, J.R., Mai, J., Lin, S.G.M., Basu, N.B., Awol, F.S., 2023. BasinMaker 3.0: A GIS Toolbox for Distributed Watershed Delineation of Complex Lake-River Routing Networks. Environmental Modelling & Software, 105688. https://doi.org/10.1016/j.envsoft.2023.105688.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R.J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., Thépaut, J.-N., 2020. The ERA5 global reanalysis. Q. J. R. Meteorol. Soc. 146 (730), 1999–2049. https://doi.org/10.1002/qj.3803.

Hnilica, J., Hanel, M., Puš, V., 2017. Multisite bias correction of precipitation data from regional climate models. Int. J. Climatol. 37 (6), 2934–2946. https://doi.org/10.1002/joc.4890.

Hoogeveen, J., Faurès, J.-M., Peiser, L., Burke, J., van de Giesen, N., 2015. GlobWat – a global water balance model to assess water use in irrigated agriculture, Hydrol. Earth Syst. Sci. 19, 3829–3844. https://doi.org/10.5194/hess-19-3829-2015.

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M. J., Tarboton, D.G., 2016. HydroShare: sharing diverse environmental data types and models as social objects with application to the hydrology domain. J. Am. Water Resour. Assoc. 52 (4), 873–889. https://doi.org/10.1111/1752-1688.12363.

Houska, T., Kraft, P., Chamorro-Chavez, A., Breuer, L., 2015. SPOTting model parameters using a ready-made Python package. PLoS One 10 (12), e0145180. https://doi.org/10.1371/journal.pone.0145180.

Huard, D., Smith, T.J., Jauvin, C., Arsenault, R., Mai, J., Gravel, F., 2023. RavenPy project readthedocs.io documentation. url: https://ravenpy.readthedocs.io/en/latest/. (Accessed 3 August 2023).

Hut, R., Drost, N., van de Giesen, N., van Werkhoven, B., Abdollahi, B., Aerts, J., Albers, T., Alidoost, F., Andela, B., Camphuijsen, J., Dzigan, Y., van Haren, R., Hutton, E., Kalverla, P., van Meersbergen, M., van den Oord, G., Pelupessy, I., Smeets, S., Verhoeven, S., de Vos, M., Weel, B., 2022. The eWaterCycle platform for open and FAIR hydrological collaboration. Geosci. Model Dev. (GMD) 15 (13), 5371–5390. https://doi.org/10.5194/gmd-15-5371-2022.

Jordahl, K., Van den Bossche, J., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Garcia Badaracco, A., Farmer, C., 2020. Geopandas/Geopandas: V0. 8.1. Zenodo.

Khattar, R., Hales, R., Ames, D.P., Nelson, E.J., Jones, N.L., Williams, G., 2021. Tethys App Store: simplifying deployment of web applications for the international GEOGloWS initiative. Environ. Model. Software 146, 105227. https://doi.org/10.1016/j.envsoft.2021.105227.

Knoben, W.J.M., Clark, M.P., Bales, J., Bennett, A., Gharari, S., Marsh, C.B., Nijssen, B., Pietroniro, A., Spiteri, R.J., Tang, G., Tarboton, D.G., 2022. Community workflows to advance reproducibility in hydrologic modeling: separating model-agnostic and model-specific configuration steps in applications of large-domain hydrologic models. Water Resour. Res. 58 (11) e2021WR031753.

Knoben, W.J.M., Freer, J.E., Fowler, K.J.A., Peel, M.C., Woods, R.A., 2019. Modular Assessment of Rainfall–Runoff Models Toolbox (MARRMoT) v1.2: an open-source, extendable framework providing implementations of 46 conceptual hydrologic models as continuous state-space formulations. Geosci. Model Dev. (GMD) 12 (6), 2463–2480. https://doi.org/10.5194/gmd-12-2463-2019.

Koziol, B., Endsley, A.K., Huard, D., Erickson, T., Murphy, S., Sims, J., O'Kuinghttons, R., Cera, T., 2023. GitHub repository for the OCGIS Python package. url: https://github.com/NCPP/ocgis. (Accessed 3 August 2023).

Latifovic, R., Homer, C., Ressl, R., Pouliot, D., Hossain, S.N., Colditz, R.R., Olthof, I., Giri, C.P., Victoria, A., 2016. 20 North American land-change monitoring system. Remote sensing of land use and land cover 303.

Lehner, B., Verdin, K., Jarvis, A., 2008. New global hydrography derived from spaceborne elevation data. Eos, Transactions American Geophysical Union 89 (10), 93–94. https://doi.org/10.1029/2008EO100001.

Leonard, L., Miles, B., Heidari, B., Lin, L., Castronova, A.M., Minsker, B., Lee, J., Scaife, C., Band, L.E., 2019. Development of a participatory Green Infrastructure design, visualization and evaluation system in a cloud supported jupyter notebook computing environment. Environ. Model. Software 111, 121–133.

Lindström, G., Pers, C., Rosberg, J., Strömqvist, J., Arheimer, B., 2010. Development and testing of the HYPE (Hydrological Predictions for the Environment) water quality model for different spatial scales. Nord. Hydrol 41, 295–319. https://doi.org/10.2166/nh.2010.007.

Mai, J., Craig, J.R., Tolson, B.A., 2020a. Simultaneously determining global sensitivities of model parameters and model structure. Hydrol. Earth Syst. Sci. 24 (12), 5835–5858. https://doi.org/10.5194/hess-24-5835-2020.

Mai, J., Kornelsen, K.C., Tolson, B.A., Fortin, V., Gasset, N., Bouhemhem, D., Schäfer, D., Leahy, M., Anctil, F., Coulibaly, P., 2020b. The Canadian Surface Prediction Archive (CaSPAr): a platform to enhance environmental modeling in Canada and globally. Bull. Am. Meteorol. Soc. 101 (3), E341–E356. https://doi.org/10.1175/bams-d-19-0143.1.

Martel, J.-L., Demeester, K., Brissette, F., Poulin, A., Arsenault, R., 2017. HMETS—a simple and efficient hydrology model for teaching hydrological modelling, flow forecasting and climate change impacts. Int. J. Eng. Educ. 33 (4), 1307–1316.

Miles, B., 2014. SMALL-SCALE Residential Stormwater Management in Urbanized Watersheds: A Geoinformatics-Driven Ecohydrology Modeling Approach (Unpublished Doctoral Dissertation). A Dissertation Submitted to the Faculty at the University of North Carolina at Chapel Hill. Chapel Hill, 2014.

Morsy, M.M., Goodall, J.L., Castronova, A.M., Dash, P., Merwade, V., Sadler, J.M., Rajib, M.A., Horsburgh, J.S., Tarboton, D.G., 2017. Design of a metadata framework for environmental models with an example hydrologic application in HydroShare. Environ. Model. Software 93, 13–28. https://doi.org/10.1016/j.envsoft.2017.02.028.

Navarro-Racines, C., Tarapues, J., Thornton, P., Jarvis, A., Ramirez-Villegas, J., 2020. High-resolution and bias-corrected CMIP5 projections for climate change impact assessments. Sci. Data 7 (1), 7.

O'Neill, B.C., Tebaldi, C., van Vuuren, D.P., Eyring, V., Friedlingstein, P., Hurtt, G., Knutti, R., Kriegler, E., Lamarque, J.F., Lowe, J., Meehl, G.A., Moss, R., Riahi, K., Sanderson, B.M., 2016. The scenario model Intercomparison project (ScenarioMIP) for CMIP6. Geosci. Model Dev. (GMD) 9 (9), 3461–3482. https://doi.org/10.5194/gmd-9-3461-2016.

Pangeo Team, 2023. Pangeo: a community platform for Big Data geoscience. url: https://pangeo.io/. (Accessed 3 August 2023).

Perrin, C., Michel, C., Andréassian, V., 2003. Improvement of a parsimonious model for streamflow simulation. J. Hydrol. 279 (1), 275–289. https://doi.org/10.1016/S0022-1694(03)00225-7.

Robinson, N., Regetz, J., Guralnick, R.P., 2014. EarthEnv-DEM90: a nearly-global, void-free, multi-scale smoothed, 90m digital elevation model from fused ASTER and SRTM data. ISPRS J. Photogrammetry Remote Sens. 87, 57–67. https://doi.org/10.1016/j.isprsjprs.2013.11.002.

Samaniego, L., Brenner, J., Craven, J., Cuntz, M., Dalmasso, G., Demirel, C. M., Jing, M., Kaluza, M., Kumar, R., Langenberg, B., Mai, J., Müller, S., Musuuza, J., Prykhodko, V., Rakovec, O., Schäfer, D., Schneider, C., Schrön, M., Schüler, L., Schweppe, R., Shrestha, P. K., Spieler, D., Stisen, S., Thober, S., Zink, M., and Attinger, S.: mesoscale Hydrologic Model – mHM v5.11.1, Zenodo [code], https://doi.org/10.5281/zenodo.4575390.

Schellekens, J., van Verseveld, W., Visser, M., hcwinsemius, laurenebouaziz, tanjaeuser, sandercdevries, cthiange, hboisgon, DirkEilander, DanielTollenaar, aweerts, Baart, F., Pieter9011, Pronk, M., arthur-lutz, ctenvelden, Imme1992, Jansen, M., 2020. Openstreams/Wflow: Bug Fixes and Updates for Release 2020.1.2. Zenodo. https://doi.org/10.5281/zenodo.4291730.

Smith, T.J., Huard, D., Bourgault, P., Dupuis, É., Lierhammer, L., Aoun, A., Logan, T., Lavoie, J., Labonté, M.P., Alegre, R., Keel, T., Rondeau-Genesse, G., Biner, S., Squire, D., et al., 2023. Xclim project readthedocs.io documentation for xclim's "statistical downscaling and bis-adjustment" toolkit. https://xclim.readthedocs.io/en/stable/notebooks/sdba.html. (Accessed 3 August 2023).

Snow, A.D., Whitaker, J., Cochran, M., Van den Bossche, J., Mayo, C., Miara, I., de Kloe, J., Karney, C., Couwenberg, B., Lostis, G., Dearing, J., Ouzounoudis, G., Filipe, Jurd, B., Kgohlke, C., Hoese, D., Itkin, M., May, R., Heitor, Vwiedermann, B.M., Barker, C., Willoughby, C., Haberthür, D., Popov, E., de Maeyer, J., Ranali, J., Evers, K., da Costa, M.A., mmodenesi, Raspaud, M., 2021. pyproj4/pyproj: 3,2,1 Release (3.2.1). Zenodo. https://doi.org/10.5281/zenodo.5514417.

Souffront Alcantara, M.A., Nelson, E.J., Shakya, K., Edwards, C., Roberts, W., Krewson, C., Ames, D.P., Jones, N.L., Gutierrez, A., 2019. Hydrologic modeling as a service (HMaaS): a new approach to address hydroinformatic challenges in developing countries. Front. Environ. Sci. 7 https://doi.org/10.3389/fenvs.2019.00158.

Sutanudjaja, E.H., van Beek, R., Wanders, N., Wada, Y., Bosmans, J.H.C., Drost, N., van der Ent, R.J., de Graaf, I.E.M., Hoch, J.M., de Jong, K., Karssenberg, D., López López, P., Peßenteiner, S., Schmitz, O., Straatsma, M.W., Vannametee, E., Wisser, D., Bierkens, M.F.P., 2018. PCR-GLOBWB 2: a 5 arcmin global hydrological and water resources model. Geosci. Model Dev. (GMD) 11, 2429–2453. https://doi.org/10.5194/gmd-11-2429-2018.

Tague, C.L., Band, L.E., 2004. RHESSys: regional hydro-ecologic simulation system—an object-oriented approach to spatially distributed modeling of carbon, water, and nutrient cycling. Earth Interact. 8 (19), 1–42. https://doi.org/10.1175/1087-3562(2004)8<1:RHRSSO>2.0.CO;2.

Taylor, P., Rahman, J., O'Sullivan, J., Podger, G., Rosello, C., Parashar, A., Sengupta, A., Perraud, J.-M., Pollino, C., Coombe, M., 2021. Basin futures, a novel cloud-based system for preliminary river basin modelling and planning. Environ. Model. Software 141, 105049. https://doi.org/10.1016/j.envsoft.2021.105049.

Tolson, B.A., Shoemaker, C.A., 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. Water Resour. Res. 43 (1) https://doi.org/10.1029/2005WR004723.

Troin, M., Poulin, A., Baraer, M., Brissette, F., 2016. Comparing snow models under current and future climates: uncertainties and implications for hydrological impact studies. J. Hydrol. 540, 588–602. https://doi.org/10.1016/j.jhydrol.2016.06.055.

Valéry, A., Andréassian, V., Perrin, C., 2014. '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. J. Hydrol. 517, 1176–1187. https://doi.org/10.1016/j.jhydrol.2014.04.058.

Van Der Knijff, J.M., Younis, J., De Roo, A.P.J., 2010. LISFLOOD: a GIS-based distributed model for river basin scale water balance and flood simulation. Int. J. Geogr. Inf. Sci. 24, 189–212. https://doi.org/10.1080/13658810802549154.

Vu, L., Charette-Migneault, Francis, Cummings, Charles-William, Schwartz, Misha, Huard, David, Smith, Trevor James, Provencher, Mathieu, Pelletier, Francis, Byrns, David, Radoš, Nik, Gagnon-Grenier, Félix, Perron, Louis-David, Jauvin, Christian, Bourgault, Pascal, 2023a. Bird-House/Birdhouse-Deploy (1.26.2). Zenodo. https://doi.org/10.5281/zenodo.7968841.

Vu, L., Charette-Migneault, F., Handan, A., Smith, T.J., Bourgault, P., Byrns, D., Cummings, C.W., Huard, D., 2023. GitHub repository for the Pavics testing suite. https://github.com/Ouranosinc/PAVICS-e2e-workflow-tests. (Accessed 3 August 2023).

Werner, M., Schellekens, J., Gijsbers, P., van Dijk, M., van den Akker, O., Heynert, K., 2013. The Delft-FEWS flow forecasting system. Environ. Model. Software 40, 65–77. https://doi.org/10.1016/j.envsoft.2012.07.010.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., Bouwman, J., Brookes, A.J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C.T., Finkers, R., Gonzalez-Beltran, A., Gray, A.J.G., Groth, P., Goble, C., Grethe, J.S., Heringa, J., Hoen, P.A.C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S.J., Martone, M.E., Mons, A., Packer, A.L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., Mons, B., 2016. The FAIR guiding principles for scientific data management and stewardship. Sci. Data 3 (1), 160018. https://doi.org/10.1038/sdata.2016.18.

Zhang, D., Chen, X., Yao, H., James, A., 2016. Moving SWAT model calibration and uncertainty analysis to an enterprise Hadoop-based cloud. Environ. Model. Software 84, 140–148. https://doi.org/10.1016/j.envsoft.2016.06.024.

Zhuang, J., Huard, D., Bourgault, P., Dussin, R., Banihirwe, A., Malevich, B., Raynaud, S., Schupfner, M., Fernandes, F., Levang, S., Almansi, M., Jüling, A., Gauthier, C., Smith, T.J., Scott, R., Rondeau-Genesse, G., Rasp, S., Caneill, R., Plough, M., Manchon, P., Stachelek, J., Li, X., Bell, R., Cherian, D., Hauser, M., 2023. GitHub repository for the xESMF Python package. url: https://github.com/pangeo-data/xESMF. (Accessed 3 August 2023).