IMPROVEMENT AND APPLICATION OF FLEXIBLE FRAMEWORKS FOR MODELLING REGIONAL STREAMFLOW VARIABILITY

Marco Dal Molin

Centre d'Hydrogéologie et de Géothermie (CHYN)
Faculté des Sciences

Supervisory Committee:
Prof. Mario Schirmer, University of Neuchâtel (Director of the thesis)
Dr. Fabrizio Fenicia, Eawag (Technical supervisor)
Prof. Riccardo Rigon, University of Trento (External reviewer)
Prof. Philip Brunner, University of Neuchâtel (Reviewer)

Université de Neuchâtel
Defended: 13.09.2021

UNIVERSITÉ DE NEUCHÂTEL

Faculté des Sciences
Secrétariat-décanat de Faculté
Rue Emile-Argand 11
2000 Neuchâtel – Suisse
Tél : + 41 (0)32 718 21 00
E-mail : secretariat.sciences@unine.ch

IMPRIMATUR POUR THESE DE DOCTORAT

La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par

Monsieur Marco DAL MOLIN

Titre:

"Improvement and application of flexible frameworks for modelling regional streamflow variability"

sur le rapport des membres du jury composé comme suit:

- Prof. tit. Mario Schirmer, directeur de thèse, Université de Neuchâtel, Suisse
- Prof. Philip Brunner, Université de Neuchâtel, Suisse
- Prof. Riccardo Rigon, Università degli studi di Trento, Italie
- Dr Fabrizio Fenicia, Eawag, Dübendorf, Suisse

Neuchâtel, le 11 octobre 2021

Le Doyen, Prof. A. Bangerter

ABSTRACT

River bodies influence the development of human society and economy, control where our settlements and cities are, represent one of the principal sources of drinking water, and provide clean and renewable energy. Improving our ability to understand and predict their behavior is a fundamental objective of hydrology as a science and as an engineering activity. Achieving precise and reliable streamflow predictions is challenging because the hydrological response of a catchment depends on numerous influence factors. This thesis makes a step forward in the study of streamflow spatial variability, developing a new framework for building hydrological models and applying it in two different studies, proposing a methodology to improve the formulation of models and the quantification of uncertainty in hydrology.

KEYWORDS

Hydrology, Modelling, Uncertainty, Bayesian statistics, Water resources

Les cours d'eau influencent le développement de la société et de l'économie, déterminent l'emplacement de nos agglomérations et de nos villes, constituent une source majeure d'eau potable et fournissent une énergie propre et renouvelable. L'amélioration de notre capacité à comprendre et à prévoir leur comportement est un objectif fondamental de l'hydrologie en tant que science et activité d'ingénierie. Obtenir des prédictions précises et fiables du débit des cours d'eau est un défi car la réponse hydrologique d'un bassin dépend de nombreux facteurs. Cette thèse fait un pas en avant dans l'étude de la variabilité spatiale des débits en développant un nouveau cadre pour la construction de modèles hydrologiques et en l'appliquant dans deux études différentes, proposant une méthodologie pour améliorer la formulation des modèles et la quantification de l'incertitude en hydrologie.

KEYWORDS

Hydrogéologie, Modélisation, Incertitude, Statistiques bayésiennes, Ressources en eau

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Rivers have always been at the center of human life. Since ancient times, civilizations have developed around big water bodies, which provided resources, energy, and a convenient mean of transportation (Macklin and Lewin, 2015). Uruk, Alexandria, Rome, Saint Petersburg, and New York are just some examples of large cities that, in different ages, developed along the course of major rivers. Modern civilizations have learned to manipulate rivers, adapting them to their needs and exploiting their resources. Additionally, water bodies represent the backbone of ecosystems, with several ecological communities developing closely with the aquatic environment (e.g., Arthington et al., 2010).

Many applications may benefit from understanding and modelling river dynamics, especially focusing on how the streamflow response to the meteorological forcing varies between regions. Examples of such applications are water resources management for agriculture and energy production (e.g., McInerney et al., 2018), protection against extreme events (e.g., Burn, 1997), procurement of drinking water (e.g., Bergion et al., 2017), projection of future scenarios (e.g., Eckhardt and Ulbrich, 2003), and protection of ecosystems (e.g., Hurford and Harou, 2014).

Hydrology is the science that studies the occurrence and movement of water on the Earth's surface and beneath the surface of the Earth (Bales, 2015). When interested in understanding the dynamics at the river scale, the focus includes the whole area draining into the river network, which is usually referred to by the name of "catchment". Catchments span a wide range of scales, ranging from a few square kilometers such as headwaters catchments (e.g., McDonnell et al., 2021) to large rivers of hundreds of thousands of square kilometers, such as the Rhine (e.g., Moser et al., 2018).

Hydrological models applied at the catchment scale are a powerful tool for investigating and describing streamflow variability (e.g., McMillan et al., 2014). As engineers, we need models that provide correct and reliable predictions that help us designing and managing infrastructures; as scientists, we use models also to gain a better understanding of the processes controlling the hydrological cycle (Kirchner, 2006). In this thesis, the focus is on the construction and usage of catchment-scale models finalized to understand and model regional streamflow variability.

1.1 CATCHMENT-SCALE HYDROLOGICAL MODELS

Catchment-scale hydrological models represent a sub-set of the models that are used in hydrology. Their main utility is when the interest is in capturing dynamics at the scale of a catchment (e.g., Dooge, 1988). Typical applications include management of water resources, information of agricultural practices, realization of predictions related to future climate change scenarios, identification of dominant hydrological processes, etc. (e.g., Eckhardt and Ulbrich, 2003; Clark et al., 2011b; Moser et al., 2018). Although catchment models are mainly used to simulate the streamflow response, their application can extend to the representation of other water or energy fluxes (e.g., Seibert and McDonnell, 2002; Matgen et al., 2012) or the simulation of transport processes (e.g., Bertuzzo et al., 2013; Ammann et al., 2020).

Catchment models can be classified based on the nature of their constitutive equations between conceptual and physically-based models. The former looks for a relation between inputs and outputs that is not based on specific measurable processes but a general understanding, at the macro scale, of the functioning of the entire system it simulates (e.g., a catchment). The latter, on the other hand, intends to represent all (or, at least, most of) the physical processes that may affect the output variables, upscaling the knowledge derived from lab experiments and using relations that are based on measurable quantities (e.g., Refsgaard, 1996). Common conceptual models include TopModel (Beven and Kirkby, 1979), HBV (Lindstrom et al., 1997), and GR4J (Perrin et al., 2003); common physically-based models include MIKE-SHE (Refsgaard and Storm, 1995), tRIBS (Ivanov et al., 2004), and GEOtop (Rigon et al., 2006).

A second classification can be done based on the spatial resolution and comprises models that range between lumped and grid-based configuration. Lumped models represent the entire system as a unit and therefore do not account for spatial variability (e.g., Kavetski and Fenicia, 2011). Semi-distributed models partition the catchment into subcatchments and produce response simulations at the subcatchment outlets (e.g., Feyen et al., 2008; Lerat et al., 2012). Such models often partition the catchments into "hydrological response units" (HRUs) which account for smaller-scale variability of catchment properties (e.g., Arnold et al., 1998; Gurtz et al., 1999; Viviroli et al., 2007; Fenicia et al., 2016). Grid-based models often represent the highest level of spatial discretization, with the domain that is divided using a grid; every cell has its unique hydrological response, which depends on its characteristics, and the total streamflow is obtained by integrating the contribution of all the cells (e.g., Samaniego et al., 2010; Maxwell, 2013).

In terms of model development approaches, two contrasting philosophies can be distinguished (Hrachowitz and Clark, 2017). The "bottom-up" approach is based on the characterization and upscaling of small-scale variability. This approach typically leads to so-called physically based, distributed models. A distinguishing advantage of this approach is that it can produce highly resolved or "hyper-resolution" models. Typical shortcomings of this approach are its reliance on upscaling assumptions that are often difficult to verify and its tendency to produce overparameterized models (e.g., Kampf and Burges, 2007). The "top-down" approach, on the other

hand, is based on the disaggregation of the system responses. This approach has the advantage of producing minimally parameterized models that are directly representative of the scale of interest. Its disadvantage is represented by the limits to which one can finger down into the smaller scale. Therefore, the top-down approach typically produces a coarser process representation than the bottom-up approach (e.g., Abbott and Refsgaard, 2012).

Although models resulting from both approaches have been successfully applied for simulating catchment hydrology (e.g., see review studies in Fatichi et al., 2016; Knoben et al., 2020), the choice of the proper modelling approach depends on the objectives of the study and must be supported by adequate data availability. When interested in capturing the aggregated response of a catchment (e.g., streamflow at a specific point of the river network), the degree of detail offered by physically-based models may be superfluous and counterproductive since their reliance on fine detailed processes may be not supported by the usually coarse resolution of the data when dealing with large catchments (e.g., Beven, 2006b). For these reasons, the studies in this thesis, which focus on streamflow simulation, will use conceptual models.

Following the top-down approach, the problem of model structure identification is typically tackled through model comparisons (e.g., Sivapalan et al., 2003a). The model comparisons need to be constructed in a controlled way, such that each model underlies a specific hypothesis on dominant processes which needs to be appraised (e.g., Fenicia et al., 2014).

The hydrological response of a catchment depends on multiple factors: climatic and meteorological conditions (e.g., precipitation, temperature, solar radiation, etc.) control the amount and form of water available in the catchment; once the fluxes reach the catchment, their fate depends on a plethora of catchment characteristics, namely topography, soil properties, geology, land use, etc. (e.g., Arnold et al., 1998; Mazvimavi et al., 2005; Bloomfield et al., 2009; Kuentz et al., 2017). However, when trying to isolate the individual factors that truly control the catchment response, it is often the case to end up with multiple alternative hypotheses that are able to explain the catchment response in a similar way, a problem known as "equifinality" (Beven, 2006a).

Given the lack of explicit knowledge of all the processes controlling the catchment response, uncertainty representation in the hydrological models is essential (e.g., Wagener and Gupta, 2005). In most cases, deterministic models are made stochastic by adding a lumped error term to the model output (e.g., McInerney et al., 2017); more advanced methods for representing the uncertainty consist in isolating the different sources of uncertainty (e.g., Renard et al., 2011) or in making model states or fluxes stochastic (e.g., Reichert and Mieleitner, 2009).

1.2 FLEXIBLE FRAMEWORKS

The selection of a hydrological model structure has motivated multiple research directions. On the one hand, many models have been developed following the "fixed" model paradigm, i.e., with the objective of creating a single general model that can be applied to every catchment; an emblematic example is given by the development of the GR4J model (Perrin et al., 2003) where

the authors have refined the model structure to provide a good performance on a wide range of different catchments. On the other hand, the "flexible" modelling paradigm pursues the idea of adapting the model structure to the peculiarities of the catchment and results in the development of modular toolkits (e.g., Fenicia et al., 2011; Leavesley et al., 1996).

Whether in search of a single model or multiple models, model selection necessarily relies on a process of model development, comparison, and refinement (e.g., Young, 1998; Sivapalan et al., 2003a; Clark et al., 2011a), which can be facilitated using flexible modelling frameworks. Compared to classical fixed-structure models like HBV (Lindstrom et al., 1997), GR4J (Perrin et al., 2003), or HYMOD (Boyle et al., 2001), where the only degree of freedom is represented by the choice of the parameters, flexible frameworks allow the exploration of different model structures that, given the high granularity of possible model formulations, vary in a gradual and controlled way, while maintaining other aspects (e.g., numerical implementation, software interface, etc.) fixed. Examples of flexible frameworks include FUSE (Clark et al., 2008), CMF (Kraft et al., 2011), SUPERFLEX (Fenicia et al., 2011; Kavetski and Fenicia, 2011), GEOframe-NewAge (Formetta et al., 2014), SUMMA (Clark et al., 2015a,b), RAVEN (Craig et al., 2020), etc.

As discussed for hydrological models in general, also flexible frameworks can be classified, based on the nature of their constitutive equations, between conceptual and physically-based frameworks and, based on the spatial resolution on which they operate, between lumped, semi-distributed, and distributed frameworks.

Open challenges related to the design of a flexible framework include the definition of its range of applications, the creation of a software solution that achieves the prescribed requirements, and keeping the balance between flexibility and ease of use. These aspects are discussed with greater detail in section 2.1.

1.3 UNDERSTANDING AND MODELLING REGIONAL STREAMFLOW VARIABILITY

Identifying dominant processes that control streamflow regional variability, i.e., how catchments respond differently to the meteorological forcing, is of fundamental importance for understanding better catchment functioning and for tackling other related open challenges in hydrology, such as prediction in ungauged catchments (e.g., Hrachowitz et al., 2013), model regionalization (e.g., Parajka et al., 2005), and catchment classification (e.g., Thorsten et al., 2007).

Current approaches for modelling streamflow variability can be divided into three categories:

- Statistical methods can be used to group similar catchments or to extrapolate streamflow properties; examples include regression of streamflow characteristics to catchment properties (e.g., Berger and Entekhabi, 2001), correlation analysis among multiple catchment characteristics (e.g., Trancoso et al., 2017; Carlier et al., 2018; Wirth et al., 2020), and clustering of catchments into hydrologically similar groups (e.g., Kuentz et al., 2017).

- Process-based hydrological models can be used to test the hypotheses about the dominant processes and, then, attribute them to known catchment characteristics; examples include synthetic numerical experiments (e.g., Carlier et al., 2019), the search of a lumped model structure that better represents processes (e.g., Fenicia et al., 2014), or the creation of spatial discretizations that focus on defining the controlling characteristics (e.g., soil, topology, land use, etc.) of the catchment (e.g., Antonetti et al., 2016; Fenicia et al., 2016).
- Combined usage of statistical methods with process-based models; in this case, statistical methods can be used, for example, to derive streamflow properties and process-based models can take advantage of such properties to make streamflow predictions (e.g., Prieto et al., 2019).

This thesis proposes two approaches for understanding and describing streamflow variability. The methodologies are applied to the Thur catchment (Switzerland), which is described in section 3.2.

The first approach uses process-based models to identify meteorological and catchment properties that control streamflow variability. This is done by building a semi-distributed model that represents in its structure only the processes controlling catchment functioning. Such a model, which captures the dominant processes, should be able to extrapolate well to other locations in the river network or other periods, becoming a valuable tool for making predictions in ungauged locations.

However, building a model that takes into account only the dominant processes brings several challenges. First, the identification of the controlling factors can be obscured by the presence of multiple characteristics that, being correlated among each other, can explain similarly well streamflow variability (e.g., Lacey and Grayson, 1998); this brings the necessity to introduce expert knowledge to guide this selection. Second, the definition of how to measure the goodness of the model is not trivial; common performance metrics (e.g., Nash-Sutcliffe efficiency) measure the fit between simulated and observed data; however, simulations with good performance metrics may not be able to represent correctly some specific hydrograph characteristics, usually called "signatures" (e.g., Sivapalan, 2006), that capture features that may be relevant for human activities or ecosystems (e.g., baseflow, hydrograph recessions, flow duration curves, etc.). Third, the model built should have a complexity proportional to the amount of data that is available to constrain the model during calibration. To this end, suitable calibration techniques that exploit the nested configuration of the catchment should be adopted (e.g., Fenicia et al., 2016).

The second approach uses statistical methods together with process-based models to describe and model streamflow variability. The first objective, here, is to estimate streamflow signatures and their related uncertainty using a combination of mechanistic and statistical models. Estimated signatures can then be used to calibrate process-based models, enabling the simulation of streamflow.

Extrapolating the signatures to another location requires the understanding of the variability between catchments. Moreover, the estimation of the uncertainty in the final streamflow predictions

requires the quantification of the uncertainty in the estimated signatures and the development of methodologies to transmit it from the signature space to the streamflow space (e.g., Fenicia et al., 2018; Kavetski et al., 2018).

1.4 THESIS OBJECTIVES

This thesis explores the design, implementation, and usage of flexible modelling frameworks as fundamental tools for studying streamflow variability. The objective of the thesis can be summarized as follows:

1. Improve flexible modelling frameworks. In particular:
    a) Analyze current flexible frameworks to understand their differences and limitations;
    b) Define a list of desirable requirements that a new flexible framework should satisfy, both from the software and the scientific perspective;
    c) Design and implement a new flexible framework.
2. Develop a methodology, based on the usage of flexible frameworks, for building hydrological models that include the main drivers of streamflow spatial variability. In particular:
    a) Analyze meteorological and catchment characteristics to understand their influence on streamflow response;
    b) Formulate hypotheses on drivers of streamflow spatial variability;
    c) Validate these hypotheses by building semi-distributed models that reflect the dominant processes.
3. Develop a methodology for the prediction of streamflow in ungauged locations. In particular:
    a) Estimate streamflow signatures and their uncertainty in ungauged locations;
    b) Calibrate a model to uncertain signatures to provide streamflow predictions in ungauged locations.

1.5 STRUCTURE OF THE THESIS

This thesis is divided into three main chapters addressing the three objectives listed in section 1.4, followed by the conclusion and outcome. Chapter 2 presents the SuperflexPy flexible modelling framework, describing the principles that guided its realization and giving examples of its usage. Chapter 3 presents an analysis of the characteristics of the Thur catchment and describes the process for constructing a semi-distributed model that represents only the dominant processes controlling streamflow. Chapter 4 presents a study on the calibration of hydrological models on estimated streamflow signatures as a potential approach to make predictions in ungauged

locations. Chapter 5, finally, summarizes the outcomes of these studies and discusses future research directions.

2

SUPERFLEXPY: AN OPEN SOURCE PYTHON FRAMEWORK FOR BUILDING, TESTING AND IMPROVING CONCEPTUAL HYDROLOGICAL MODELS

Catchment-scale hydrological models are widely used to represent and improve our understanding of hydrological processes and to support operational water resources management. Conceptual models, which approximate catchment dynamics using relatively simple storage and routing elements, offer an attractive compromise in terms of predictive accuracy, computational demands, and amenability to interpretation. This paper introduces SuperflexPy, an open-source Python framework implementing the SUPERFLEX principles (Fenicia et al., 2011) for building conceptual hydrological models from generic components, with a high degree of control over all aspects of model specification. SuperflexPy can be used to build models of a wide range of spatial complexity, ranging from simple lumped models (e.g. a reservoir) to spatially distributed configurations (e.g. nested sub-catchments), with the ability to customize all individual model components. SuperflexPy is a Python package, enabling modelers to exploit the full potential of the framework without the need for separate software installations, and making it easier to use and interface with existing Python code for model deployment. This paper presents the general architecture of SuperflexPy, discusses the software design and implementation choices, and illustrates its usage to build conceptual models of varying degrees of complexity. The illustration includes the usage of existing SuperflexPy model *elements*, as well as their extension to implement new functionality. Comprehensive documentation is available online and provided as supplementary material to this paper. SuperflexPy is available as open-source code, and can be used by the hydrological community to investigate improved process representations, for model comparison, and for operational work.

This chapter is based on the publication: Dal Molin, M., Kavetski, D., and Fenicia, F. (2021) "SuperflexPy 1.2.1: an open source Python framework for building, testing and improving conceptual hydrological models" In revision, *Geoscientific Model Development*.

NOTE

This chapter contains references to the documentation of SuperflexPy, often referred as supplementary material, which represented the supplementary material in the original publication. Given its large dimension, it has been decided to exclude the documentation from the thesis. The documentation can be found using the DOI 10.5281/zenodo.5069311.

2.1 INTRODUCTION

2.1.1 *Conceptual hydrological models*

Catchment-scale hydrological models are widely used to predict catchment behavior under natural and human-impacted conditions, as well as to represent and improve our understanding of internal catchment functioning (e.g., Beven, 1989). For example, catchment models underlie projections of climate change impact on groundwater recharge and streamflow (e.g., Eckhardt and Ulbrich, 2003), are used as tools for hypothesis testing to identify dominant hydrological processes (e.g., Clark et al., 2011b; Hrachowitz et al., 2014; Wrede et al., 2015), and are used to inform agricultural practices such as irrigation scheduling (e.g., McInerney et al., 2018) and pesticide application (e.g., Moser et al., 2018; Ammann et al., 2020). The typical use of hydrological models is to simulate or forecast the streamflow response (runoff) of a catchment to rainfall forcing; for this reason they are often referred to as rainfall-runoff models (e.g., Moradkhani and Sorooshian, 2008). However, their application extends to the simulation of other environmental variables such as groundwater levels (e.g., Seibert and McDonnell, 2002) and soil moisture (e.g., Matgen et al., 2012), as well as water chemistry (e.g., Bertuzzo et al., 2013; Ammann et al., 2020).

An important class of catchment models are "process based" models, which attempt to explicitly describe the cascade of processes transforming catchment inputs (e.g. precipitation) into outputs (e.g. streamflow). These models are an appealing choice due to their broad physical underpinnings, as well as their ability to represent internal catchment processes and potential for predicting catchment responses under changing environmental conditions. Process based models can be classified according to the nature of their constitutive equations (e.g. conceptual or physically based) and their spatial resolution (e.g. lumped or distributed) (e.g., Refsgaard, 1996).

Conceptual models, where catchment dynamics are approximated using relatively simple storage and routing elements (e.g., Fenicia et al., 2011), are common in practice because they offer an attractive compromise in terms of predictive accuracy, computational demands, and amenability to interpretation. Common conceptual models include TopModel (Beven and Kirkby, 1979), HBV (Lindstrom et al., 1997), GR4J (Perrin et al., 2003), and HyMod (Boyle et al., 2001).

In terms of spatial resolution, conceptual models can be applied in a lumped configuration (treating the entire catchment as a single unit) if the interest is in modelling integrated catchment outputs (e.g. streamflow at the catchment outlet). Alternatively, distributed configurations

can be used if the interest is in modelling hydrological behavior at internal locations (e.g., sub-catchments). In such distributed setups, the catchment is subdivided into spatial elements such as sub-catchments (e.g., Feyen et al., 2008; Lerat et al., 2012), Hydrological Response Units (HRUs) (e.g., Arnold et al., 1998; Fenicia et al., 2016; Dal Molin et al., 2020), or grids (e.g., Samaniego et al., 2010). A common strategy for developing distributed conceptual models is to represent individual landscape elements using independent (non-interacting) lumped models, and then obtain total catchment outflow by aggregating the outflows from these individual models, potentially incorporating flow routing elements to represent routing delays. This strategy is often referred to as "semi-distributed" modelling (e.g., Boyle et al., 2001), and typically employs discretization based on principles of "hydrological similarity" (e.g., Sivapalan et al., 1987); HRU-based discretization is particularly common (e.g., Leavesley, 1984). In many applications, semi-distributed modelling achieves good predictive ability – while greatly simplifying model representation and reducing computational demands compared to fully-integrated 2D/3D distributed models such as Parflow (Maxwell, 2013) or Mike She (Refsgaard and Storm, 1995), which typically use much smaller landscape elements and explicitly model lateral exchanges. For the purposes of this presentation, we consider semi-distributed modelling to be a special case of distributed modelling.

2.1.2 *Hydrological model structure and flexible modelling frameworks*

The selection of model structure has preoccupied researchers and practitioners since the early days of hydrological modelling (e.g., Ibbitt and O'Donnell, 1971; Moore and Clarke, 1981; Jakeman and Hornberger, 1993). Although in principle the physical laws governing hydrological processes are the same everywhere, the diversity of catchment conditions in terms of topography, soil, geology, vegetation, and anthropogenic influence results in remarkably different manifestations of these physical laws at the catchment scale. These local differences, also termed "uniqueness of place" (Beven, 2000), considerably limit our ability to develop generalizable hydrological hypotheses (e.g., Thorsten et al., 2007).

Model structure selection has motivated multiple research directions, including the search for a single model structure that achieves good prediction across all catchments (the "fixed" model paradigm), and the search for model structures best suited for specific locations and/or environmental conditions (the "flexible" model paradigm). Whether in search of a single model or multiple models, model selection necessarily relies on a process of model development, comparison, and refinement. Approaches to formalize this process include the top-down approach (e.g., Sivapalan et al., 2003a), the system identification approach (e.g., Young, 1998), and the method of multiple working hypotheses (e.g., Clark et al., 2011a). These approaches are not mutually exclusive, as the notion of comparing multiple model representations is ubiquitous in model development and empirical science in general.

The process of model development, comparison, and refinement can be facilitated using flexible modelling frameworks, which enable hydrologists to hypothesize, implement, and (eventually)

test and refine different model structures. Flexible frameworks have themselves developed along multiple directions according to their intended scopes of application. For example, GEOframe-NewAge (Formetta et al., 2014), SUMMA (Clark et al., 2015a,b), and CHM (Marsh et al., 2020) focus on the realm of physically based models. The CAPTAIN toolbox (Young et al., 2009) is a general toolkit for time series analysis. Machine learning frameworks such as scikit-learn (Pedregosa et al., 2011) and PyTorch (Paszke et al., 2019) can be used to construct data driven models.

In this paper, we focus on flexible frameworks intended for conceptual hydrological modelling. Examples of such frameworks include FUSE (Clark et al., 2008), SUPERFLEX (Fenicia et al., 2011; Kavetski and Fenicia, 2011), CMF (Kraft et al., 2011), PERSiST (Futter et al., 2014), ECHSE (Kneis, 2015), MARRMoT (Knoben et al., 2019), and RAVEN (Craig et al., 2020).

When discussing a mathematical model, it is relevant to distinguish its conceptual principles from its software implementation. In the hydrological literature, modelling concepts and their software implementation have been presented both jointly and separately. For example, the original FUSE publication (Clark et al., 2008) introduced the modelling concepts, while subsequent work (Vitolo et al., 2016) provided an R implementation. The original SUPERFLEX publications presented the modelling principles (Fenicia et al., 2011) and demonstrated its capabilities (Kavetski and Fenicia, 2011); while Fortran and Matlab implementations were developed as part of research work (e.g., David et al., 2019), these implementations have not been published or made available as standalone products. In contrast, some models, (e.g., MARRMoT, Knoben et al., 2019) have been presented with a publication describing both the theoretical principles and the software implementation.

A software implementation should fulfill the intended goals of the flexible framework, in particular supporting the envisaged flexibility in terms of processes representation, spatial distribution, numerical solution methods, etc. The software implementation should also be accessible to users in terms of ease of installation, operation, eventual extension, etc. Existing frameworks approach these conceptual and practical requirements with different priorities, e.g., focusing on selected modelling objectives (e.g., model mimicry) and/or limiting the range of applications (e.g., only to lumped setups), in order to simplify the model formulation and operation.

In terms of application scope of a flexible framework for conceptual hydrological modelling, we focus on the following "realms":

1. Lumped models;
2. Distributed setups, including simulation of sub-catchments and flows/processes at internal points;
3. Substance transport modelling, including water isotopes, pesticides, etc.;
4. Ability to reproduce existing models, when necessary.

In terms of software implementation, we consider the following practical criteria:

1. *Ease of use, including installation, learning, and operation.* Interoperability with external software, for example for model calibration and uncertainty analysis, is of obvious relevance because hydrological models are often used as parts of larger-scale projects and operations.

2. *Ease of modifications and extensions.* Even a comprehensive software implementation will eventually require extension. For example, a modelling framework intended to simulate streamflow may require extension to simulate water chemistry. Another type of modification might be a switch to a numerical implementation better suited for parallel computing, etc.

3. *Computational efficiency.* Hydrological model applications, especially including calibration and uncertainty quantification, may require thousands or even millions of model runs.

These criteria are challenging or even impossible to meet simultaneously. Hence, implementing a flexible framework entails juggling multiple obvious and less obvious tradeoffs. For example, the intended flexibility of a framework may come at the expense of ease of use, similar to how computer languages have varying degrees of abstraction from the hardware behavior. Implementing a practical flexible framework therefore requires careful code design, experimentation, and inevitably, some compromises.

This work pursues the flexible framework objectives defined above by building upon the concept of SUPERFLEX (e.g., Fenicia et al., 2011; Kavetski and Fenicia, 2011; Fenicia et al., 2014, 2016). A key attractive feature of SUPERFLEX as a modelling concept is the fine "granularity", i.e., the degree of flexibility of model structures it can support, which enables systematic and detailed hypothesis testing (Fenicia et al., 2011). For example, the hydrologist should have the ability to select and combine individual model elements (e.g., reservoirs, lag functions, etc.), as well as to build customized elements.

The development of the proposed framework capitalizes on the authors' collective experience in hydrological model design and application. The original Fortran implementation of SUPERFLEX, hereafter referred to as SUPERFLEX-F90, has been used in a series of case studies over the last decade, ranging from lumped model implementations (e.g., Kavetski and Fenicia, 2011; Fenicia et al., 2014), to distributed setups (e.g., Fenicia et al., 2016; Dal Molin et al., 2020), interpretation in the context of fieldwork insights (e.g., Wrede et al., 2015), large scale model intercomparisons (e.g., van Esse et al., 2013), and the inclusion of pesticide/substance transport (e.g., Ammann et al., 2020). The earlier Flex framework was used in studies exploring the use of multivariate data to refine the model structure (e.g., Fenicia et al., 2006, 2008). The modelling framework FUSE was used for a range of experiments in process representation (e.g., Clark et al., 2011b), data analysis (e.g., Henn et al., 2018), and numerical solution (e.g., Clark and Kavetski, 2010; Kavetski and Clark, 2010). The SUMMA framework represented an application of flexible modelling principles to physically based modelling. These applications have highlighted the versatility of the SUPERFLEX principles, and of flexible modelling approaches in general, to solve increasingly complex modelling problems – but have also highlighted implementation choices that limit the effectiveness and range of

application of current software. This work provides a new implementation of SUPERFLEX that addresses many of these limitations.

2.1.3 *Aims*

This paper introduces SuperflexPy, which is a new open-source Python software implementation of the SUPERFLEX principles for conceptual hydrological model development. Particular attention is given to the challenges of implementing a framework that achieves the flexibility envisaged by SUPERFLEX and flexible frameworks in general. Our objectives are as follows:

1. Present SuperflexPy and its basic building blocks (*components*): *elements*, *units*, *nodes*, and *network*;
2. Illustrate how SuperflexPy can help hydrologists implement a conceptual model structure at the desired level of internal complexity and spatial resolution – including recreating existing models and developing new models;
3. Provide a broad discussion of the hydrological modelling software implementation challenges and of how SuperflexPy contributes to the toolkits available to the hydrological community.

The paper is organized as follows. Section 2.2 describes the SuperflexPy architecture and building blocks, and provides a short demo (aims 1 and 2). Section 2.3 illustrates selected applications of the framework, including the setup of SUPERFLEX configurations used in earlier case studies and the use SuperflexPy to create new *elements* (aim 2). Section 2.4 provides more technical SuperflexPy details, useful for understanding the usage and general potential of the framework (aim 1). Section 2.5 discusses SUPERFLEX design choices in the context of existing flexible frameworks, including current limitations and future developments (aim 3). Finally, section 2.6 provides a brief overall summary and conclusions.

The examples presented in the paper are generally intended to provide the intuition and reasoning behind SuperflexPy. The model documentation provides detailed information and use instructions. The documentation is available and maintained online (refer to "code availability" section); references from the paper to the documentation point to the static PDF version that can be accessed at the DOI 10.5281/zenodo.5069311.

2.2 DESCRIPTION OF SUPERFLEXPY

2.2.1 *General organization*

The SuperflexPy framework has a hierarchical organization with four nested levels: "*element*", "*unit*", "*node*", and "*network*", collectively referred as "*components*". These *components* are shown

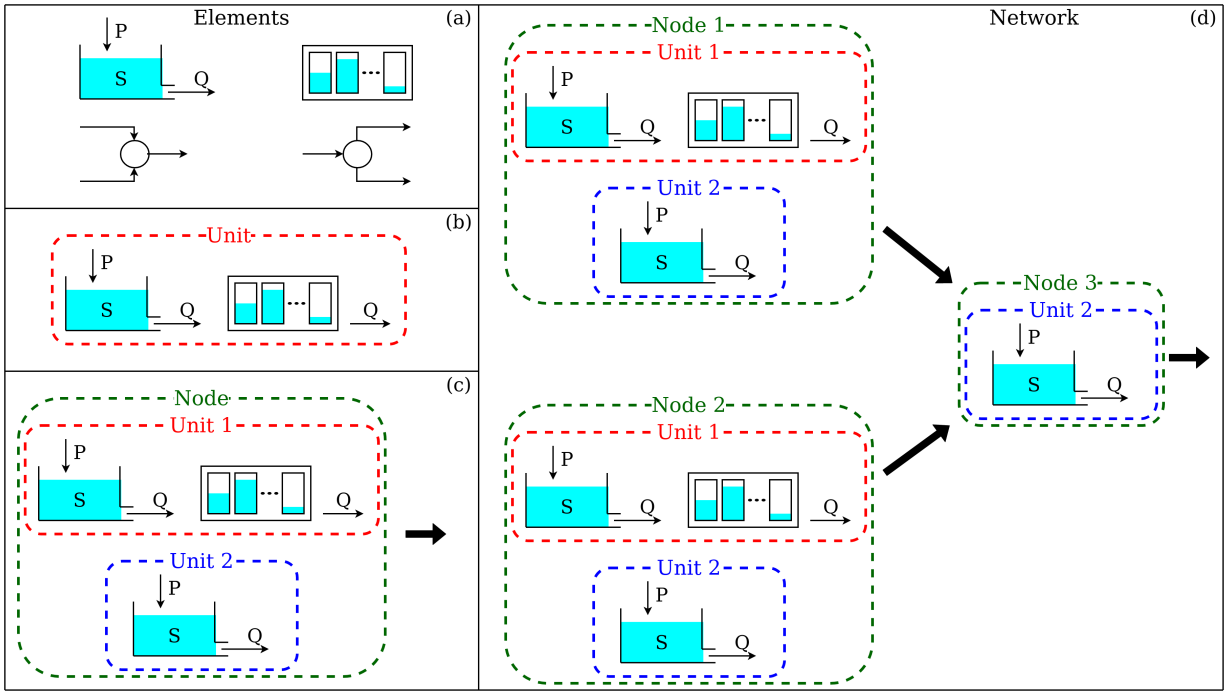

Figure 2.1: The four hierarchical levels of SuperflexPy and their respective *components*. **(a)** *Elements* (e.g. reservoirs, lags, connections) are used to represent individual hydrological processes/catchment response mechanisms; **(b)** *Units* connect multiple *elements* and are intended to implement lumped catchment models; **(c)** *Nodes* collect multiple *units* that operate in parallel representing different landscape elements within a catchment; **(d)** *Network* connects multiple *nodes* and is used to represent distributed setups.

in figure 2.1 and described below. Further practical details are provided in chapter 4 of the supplementary material:

1. *Element* (figure 2.1a). This level represents the basic model building block and is used to create reservoirs, lag functions, and connections. An *element* can be used to represent an entire catchment, or, more commonly, a specific hydrological process or response mechanism within the catchment.

   The **reservoir** element is used to conceptualize processes involving the storage and release of water and other fluxes. It is described mathematically by ordinary differential equations (ODEs),

   $$\frac{d\mathbf{S}(t)}{dt} = \mathbf{g}_S\left(\mathbf{S}(t), \mathbf{X}(t); \boldsymbol{\theta}\right) \tag{2.1}$$

   $$\mathbf{Y}(t) = \mathbf{g}_Y\left(\mathbf{S}(t), \mathbf{X}(t); \boldsymbol{\theta}\right) \tag{2.2}$$

   where $\mathbf{S}$ are the state variables (e.g., water storages), $\mathbf{X}$ are the inputs (e.g., precipitation), $\mathbf{Y}$ are the outputs (e.g., streamflow), and $\mathbf{g}_S$ and $\mathbf{g}_Y$ are specified constitutive functions (e.g., storage-discharge relationships).

   In most conceptual models, reservoir elements have a single state variable (representing water storage); multiple state variables can be accommodated if necessary (e.g., to keep

track of snow and liquid water separately). Mathematically, a multistate reservoir can be represented by a system of differential equations of the form of equations 2.1 and 2.2.

The solution of equation 2.1 is usually obtained numerically using external numerical procedures referred to as "numerical approximators" (see section 2.4.3).

The **lag function** element is used to represent delays in the transmission of the fluxes (e.g., routing). It is described mathematically by a convolution integral,

$$\mathbf{Y}(t) = \mathbf{X}(t) * \mathbf{g}_H(t;\theta) = \int_0^T \mathbf{X}(t-\tau)\mathbf{g}_H(\tau;\theta)d\tau \tag{2.3}$$

where $*$ denotes the convolution operator, $\mathbf{X}$ is the input (e.g., water flux), $\mathbf{g}_H$ is the impulse response function, and T is the time of influence of $\mathbf{g}_H$ (i.e. the maximum lag).

There is a general mathematical correspondence between reservoirs and lag functions (e.g., Nash, 1957). SuperflexPy users can select the element specification best suited to their specific context.

The **connection** element is used to connect two or more *elements* whenever a direct connection is not possible. For example, connection *elements* are used when a flux needs to be split among multiple *elements* downstream (splitter), or, vice versa, when multiple fluxes need to be aggregated (junction). A particular type of connection is represented by the "transparent" element, which simply outputs the same fluxes it receives as inputs, and is used to facilitate the connection between *elements* (see description of *unit* below).

All connection elements are stateless and can be represented mathematically as follows,

$$\mathbf{Y}(t) = \mathbf{g}_C\left(\mathbf{X}(t);\theta\right) \tag{2.4}$$

where $\mathbf{g}_C$ describes the connectivity between input fluxes and output fluxes, and θ represents connectivity parameters (if any).

2. *Unit* (figure 2.1b). A *unit* is a collection of multiple connected *elements*, and is generally intended to implement a lumped catchment model or an HRU in a distributed model. Multiple reservoir and lag function elements within a *unit* can be connected to each other, either directly (one-to-one connections), or using connection elements such as splitters and junctions (when a single *element* is connected to multiple *elements*). The multiple *elements* within a *unit* are arranged in *layers*, with the following restrictions: (i) feedback loops between the *elements* are not allowed and (ii) *elements* can be connected only if they belong to two consecutive *layers*. Fluxes between *elements* in nonconsecutive *layers* are passed using transparent elements. The concept of *layers* will be elaborated and illustrated in section 2.5.1.1; see also section 4.2 of the supplementary material. In technical terms, the structure formed by the *elements* must be a directional acyclic graph (DAG). The motivation and

implications of these design choices on model generality and computational efficiency are elaborated in sections 2.5.1.1 and 2.5.2.

3. *Node* (figure 2.1c). A *node* is a collection of multiple *units* that operate in parallel. In the context of distributed models, the *node* can be used to represent a single catchment and the *units* can be used to represent multiple landscape elements or HRUs within the catchment. Each *unit* within a *node* is characterized by a weight, which typically represents its area fraction or, more generally, its contribution to the total outflow of the *node*. The weights are used to combine the output fluxes from the *units* into the total output flux of the *node*. Another important attribute of a *node* is its "area", which is used when multiple *nodes* are combined into a *network* (see below).

4. *Network* (figure 2.1d). A *network* connects multiple *nodes* into a tree structure, and is typically intended to develop a distributed model that generates predictions at internal sub-catchment locations (e.g. to reflect a nested catchment setup). The *network* routes the fluxes from upstream *nodes* (leaves of the tree) to the final downstream *node* (root of the tree). Routing delays in the river network can be simulated by feeding *node* outputs into lag function elements. The area of each *node* is used to determine its contribution to the total outflow of the *network*. Only a single *network* can be used in a given SuperflexPy model.

The hierarchical organization of SuperflexPy makes the effort required to configure it to a new problem proportional to the problem complexity. In particular, many common model setups can be constructed without necessarily using all levels listed above, thus reducing configuration effort. Some representative examples are given below:

- Level 1 is sufficient to create single-*element* models, e.g., a single-reservoir model or a unit hydrograph model (e.g., Kirchner, 2009);

- Level 2 is sufficient to create a lumped model structure, such as GR4J (Perrin et al., 2003) or Hymod (Boyle et al., 2001);

- Level 3 is sufficient create a distributed model that represents spatial heterogeneity but generates predictions only at the catchment outlet (e.g., Beven and Kirkby, 1979; Gao et al., 2014; Nijzink et al., 2016);

- Level 4 is needed only in models that generate predictions at interior points, such as SWAT (Arnold et al., 2012), GEOframe-NewAge (Formetta et al., 2014), and distributed SUPERFLEX applications (e.g., Fenicia et al., 2016; Dal Molin et al., 2020).

Examples of SuperflexPy models implemented at Levels 2 and 4 are given later in section 2.3. Note that the association of specific SuperflexPy *components* to specific hydrological entities, e.g., the use of *units* for HRUs and *nodes* for sub-catchments, is not intended as a rigid prescription.

Other association choices may be favored by the modeler depending on the required model structure and spatial connectivity.

The clarity of visual model representation is particularly important in flexible frameworks because they can generate many subtly different configurations (e.g., Bancheri et al., 2019). The model schematics in this paper indicate explicitly every *element*, including reservoirs, lag functions, and junctions (e.g., figure 2.1).

From a software design prospective, SuperflexPy embraces the object-oriented paradigm (e.g., Meyer, 1988). All framework *components* are represented by objects that can operate either alone or together, interacting with each other and with external libraries (e.g. for calibration) through defined interfaces. More details are provided in section 2.4.2.

All SuperflexPy *components* are characterized by states and/or parameters, which are controlled programmatically using dedicated methods (refer to section 2.4.1).

2.2.2 *A simple illustration of SuperflexPy: creating a new model from existing components*

This section illustrates the key steps needed to configure and run a hydrological model using the SuperflexPy framework. The illustration presents a distributed model intended to represent a catchment with 2 HRUs and 3 sub-catchments. The model structure is shown in figure 2.1d. The catchment is represented using a *network*, the sub-catchments are represented using *nodes*, and the HRUs are represented using *units*. Two distinct HRU-specific model structures are specified and implemented using *elements*. The corresponding SuperflexPy code is shown in figure 2.2. An extended version of this demo is provided in section 6.5 of the supplementary material.

In this example, an implementation of the necessary *elements* with SuperflexPy already exists; therefore, the *elements* only need to be imported. The case where the model structure requires *elements* for which an implementation is not yet available is considered in section 2.2.3. More complex setups are described in section 2.3 and in the supplementary material.

We start by importing the model *components* required by the model structure, namely the *elements* (`LinearReservoir` and `HalfTriangularLag`), *unit*, *node*, and *network*. The numerical approximator `ImplicitEulerPython` and root finder `PegasusPython` needed to solve the ODEs associated with the reservoir elements are also imported (see section 2.4.3 for details). The import operation is shown in lines 1-7.

The imported *components* are then initialized, which entails specifying the model structure (connectivity between model *components*) and the initial values of parameters and states. The initialization sequence starts with the numerical procedures (Lines 10-11) and proceeds from the lowest-level *components* (*elements*) to the highest-level *component* (*network*).

More specifically:

L1 An *element* is initialized by specifying its parameters, states, and, where relevant, the numerical solver (lines 14-16). Each *element* is given an identifier (`id`) for subsequent use, as shown on line 23.

```
1  from superflexpy.implementation.elements.hymod import LinearReservoir
2  from superflexpy.implementation.elements.thur_model_hess import HalfTriangularLag
3  from superflexpy.framework.unit import Unit
4  from superflexpy.framework.node import Node
5  from superflexpy.framework.network import Network
6  from superflexpy.implementation.computation.pegasus_root_finding import PegasusPython
7  from superflexpy.implementation.computation.implicit_euler import ImplicitEulerPython
8
9  # Initialize computational tools
10 root_finder = PegasusPython()
11 numerical_approximator = ImplicitEulerPython(root_finder=root_finder)
12
13 # Initialize the elements
14 linear_reservoir = LinearReservoir(parameters={'k': 0.1}, states={'S0': 10.0},
15                                    approximation=numerical_approximator, id='LR')
16 lag = HalfTriangularLag(parameters={'lag-time': 3.5}, states={'lag': None}, id='LAG')
17
18 # Initialize the units
19 unit1 = Unit(layers=[[linear_reservoir], [lag]], id='U1')
20 unit2 = Unit(layers=[[linear_reservoir]], id='U2')
21
22 # Change parameters
23 unit2.set_parameters({'U2_LR_k': 0.2})
24
25 # Initialize the nodes
26 node1 = Node(units=[unit1, unit2], weights=[0.7, 0.3], area=5.0, id='N1')
27 node2 = Node(units=[unit1, unit2], weights=[0.9, 0.1], area=2.0, id='N2')
28 node3 = Node(units=[unit2], weights=[1.0], area=1.0, id='N3')
29
30 # Initialize the network
31 net = Network(nodes=[node1, node2, node3], topology={'N1': 'N3', 'N2': 'N3', 'N3': None})
32
33 # Assign the inputs to the nodes (assume P1, P2, P3 have been read)
34 node1.set_input([P1])
35 node2.set_input([P2])
36 node3.set_input([P3])
37
38 # Set the timestep
39 net.set_timestep(1.0)
40
41 # Run the model
42 net.get_output()
```

Figure 2.2: SuperflexPy code implementing the simple illustrative model in figure 2.1d

L2 A *unit* is initialized by specifying the *elements* that compose it and the identifier (lines 19-20). As noted earlier in section 2.2.1, the connectivity between *elements* is defined by conceptualizing the *unit* as a succession of *layers* that contain the *elements*. More complex examples are given in section 2.3. The parameters and states of *elements* can be changed after initialization using the methods `set_parameters` and `set_states` of the containing *units*. This operation is shown on line 23 for the `LinearReservoir` element.

L3 A *node* is initialized by specifying the *units* that compose it, their contribution (weight) to the *node* output, the influence area of the *node* (here, the area of the sub-catchment), and the identifier (lines 26-28).

L4 The *network* is initialized by specifying the *nodes* that compose it and their connectivity, called topology (line 31). The connectivity is defined indicating, for each *node*, the *node* downstream of it. A *network* identifier is not specified (as only a single *network* can be used).

The next step is to set the model inputs and time step. Lines 34-36 show how the inputs are assigned directly to the *nodes*, enabling the model to receive spatially varying rainfall and PET.

The time step is set on line 39 (variable time steps are also supported, see section 4.5.1 of the supplementary material).

The model can now be run by calling the `get_output` method of the highest-level *component*, as shown on line 42.

Note that all input quantities provided to SuperflexPy, including fluxes, time step length, parameters, states, areas, etc., must have consistent units. To reduce model code complexity and execution overhead, we take the perspective that unit checks represent pre-processing and are best handled by the user according to their own preferences and standards. Output fluxes have the same (assumed) units as input fluxes, e.g., if precipitation is in mm/h, then streamflow is also in mm/h, etc.

2.2.3 *Creating new model components with SuperflexPy*

We now consider the case where the intended model structure has *components* beyond those already available in SuperflexPy.

New model *components* can be created by extending existing SuperflexPy *components*. To this end, SuperflexPy provides a library of built-in high-level *components* that can be extended to achieve the desired functionality. We anticipate that the SuperflexPy *components* most likely to require extension are the *elements*, where new constitutive functions may be required in reservoir elements and new weight functions may be required in lag function elements. In contrast, it is less likely that *unit*, *node*, and *network* functionalities would require extension.

The extension of existing SuperflexPy *elements* takes advantage of the object-oriented paradigm underlying the SuperflexPy software design. The inheritance principle, one of the core concepts of the object-oriented paradigm, allows the user to construct new *components* by "inheriting" most of the functionalities (methods) from existing classes. Separate implementation is then required only for methods where the new model differences are to be introduced. This approach reduces substantially the amount of coding required to implement a new model *component*.

A detailed example of this procedure is given in section 2.3.2, which shows how to implement a reservoir with a new storage-discharge relationship. More examples are provided in chapters 8 and 9 of the supplementary material.

2.3 EXAMPLES OF BUILDING HYDROLOGICAL MODELS USING SUPERFLEXPY

This section provides more detailed examples of using SuperflexPy to implement hydrological models, including the use of built-in *elements* and the creation of new *elements*. We follow a progression from simple to complex. Section 2.3.1 shows the implementation of model M4, a lumped model built solely from reservoir elements and used in the original SUPERFLEX case study (Kavetski and Fenicia, 2011). Section 2.3.2 shows how to define a new *element* with a different storage-discharge relationship for one of the reservoirs of M4. Section 2.3.3 shows the

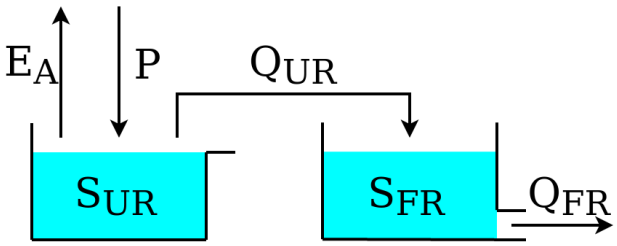

Figure 2.3: Schematic of model M4 used in the original SUPERFLEX case studies of Kavetski and Fenicia (2011).

implementation of a distributed model from a recent application of SUPERFLEX in the Thur catchment (Dal Molin et al., 2020).

Compared to the demo in section 2.2.2, which was intended to give a general sense of model building with SuperflexPy, the examples in this section represent "realistic" applications of SuperflexPy, including setting up a spatially distributed model with multiple HRUs and more complex model structure. Further technical details and additional examples, including the implementation of popular conceptual models (e.g., GR4J, HYMOD), are provided in the supplementary material (chapters 8 to 11).

2.3.1 *Implementing SUPERFLEX configuration M4*

M4 is a simple lumped model presented in Kavetski and Fenicia (2011). As shown in figure 2.3, M4 comprises two reservoirs connected in series: an "unsaturated" reservoir (UR) intended to represent the partitioning of precipitation between evaporation and runoff, and a "fast" reservoir (FR) intended to represent subsequent streamflow generation mechanisms.

UR partitions precipitation $P^{(UR)}$ into a portion that enters the UR storage and eventually evaporates through flux $E_A^{(UR)}$, and a portion $Q^{(UR)}$ that is directed to the downstream FR reservoir:

$$\frac{dS^{(UR)}}{dt} = P^{(UR)} - E_A^{(UR)} - Q^{(UR)} \tag{2.5}$$

where

$$\bar{S}^{(UR)} = \frac{S^{(UR)}}{S_{max}^{(UR)}} \tag{2.6}$$

$$Q^{(UR)} = P^{(UR)} \times \left(\bar{S}^{(UR)}\right)^{\beta^{(UR)}} \tag{2.7}$$

$$E_A^{(UR)} = E_P^{(UR)} \times \frac{\bar{S}^{(UR)}\left(1 + m^{(UR)}\right)}{\bar{S}^{(UR)} + m^{(UR)}} \tag{2.8}$$

In equations 2.6-2.8, $S_{max}^{(UR)}$ and $\beta^{(UR)}$ are model parameters. The quantity $m^{(UR)}$ is used to approximate a "smooth" threshold behavior; we typically fix $m^{(UR)} = 0.01$.

FR is a power-law reservoir,

$$\frac{dS^{(FR)}}{dt} = P^{(FR)} - Q^{(FR)} \tag{2.9}$$

```
1  from superflexpy.implementation.computation.pegasus_root_finding import PegasusPython
2  from superflexpy.implementation.computation.implicit_euler import ImplicitEulerPython
3  from superflexpy.implementation.elements.hbv import UnsaturatedReservoir, PowerReservoir
4  from superflexpy.framework.unit import Unit
5  import numpy as np
6
7  root_finder = PegasusPython()
8  numeric_approximator = ImplicitEulerPython(root_finder=root_finder)
9
10 ur = UnsaturatedReservoir(parameters={'Smax': 50.0, 'Ce': 1.0, 'm': 0.01, 'beta': 2.0},
11                            states={'S0': 25.0}, approximation=numeric_approximator, id='UR')
12 fr = PowerReservoir(parameters={'k': 0.1, 'alpha': 1.0}, states={'S0': 10.0},
13                     approximation=numeric_approximator, id='FR')
14
15 model = Unit(layers=[[ur], [fr]], id='M4')
16
17 P = np.loadtxt('precipitation.txt')
18 EP = np.loadtxt('evap_pot.txt')
19
20 model.set_input([P, EP])
21 model.set_timestep(1.0)
22
23 output = model.get_output()
```

Figure 2.4: SuperflexPy code implementing model M4 in figure 2.3

with the storage-discharge relationship given by

$$Q^{(FR)} = k^{(FR)} \left(S^{(FR)} \right)^{\alpha^{(FR)}} \tag{2.10}$$

where $k^{(FR)}$ and $\alpha^{(FR)}$ are model parameters.

The inflow $P^{(FR)}$ is given by the outflow from UR, i.e., $P^{(FR)} = Q^{(UR)}$.

M4 is a lumped model with multiple *elements*, and hence can be implemented using SuperflexPy levels L1 and L2 (*element* and *unit*, see section 2.2.1). Figure 2.4 shows the code needed to implement M4. The numerical procedures are imported and initialized on lines 1-2 and 7-8 respectively. Similar to the model described in section 2.2.2, the two model *elements* (UR and FR) are already implemented. Hence, the user only needs to import the *elements* (lines 1-3) and initialize their parameters (lines 7-13). Next, the *unit* is imported (line 4) and initialized to contain the two reservoirs (line 15). The model configuration is then complete.

The loading of input data from text file(s), databases, etc. is separate from the configuration of SuperflexPy, and can be carried out using any suitable Python library or function. In this example, we use Numpy to read time series of precipitation and PET from a text file, as shown in lines 17-18. The corresponding SuperflexPy inputs are set using these Numpy arrays, as shown on line 20. Further practical details on input-output are provided in section 4.5.5 of the supplementary material.

The model can now be run with the given input data to produce the model outputs, as shown on line 23. The outputs contain streamflow time series in the form of Numpy arrays.

```python
class NewFastReservoir(ODEsElement):

    def __init__(self, parameters, states, approximation, id):

        ODEsElement.__init__(self, parameters=parameters, states=states,
                             approximation=approximation, id=id)

        # _fluxes_python is used to calculate the fluxes doing vector operations
        self._fluxes_python = [self._fluxes_function_python]
        # _fluxes is used to solve the ODE and it is specific to the architecture of the numerical_approximator
        self._fluxes = [self._fluxes_function_python]

    def set_input(self, input):

        self.input = {'P': input[0]}

    def get_output(self, solve=True):

        if solve:
            self._solver_states = [self._states[self._prefix_states + 'S0']]
            self._solve_differential_equation()
            self.set_states({self._prefix_states + 'S0': self.state_array[-1, 0]})

        fluxes = self._num_app.get_fluxes(fluxes=self._fluxes_python,
                                          S=self.state_array,
                                          S0=self._solver_states,
                                          **self.input,
                                          **{k[len(self._prefix_parameters):]: self._parameters[k] for k in self._parameters})

        return [- fluxes[0][1]]

    @staticmethod
    def _fluxes_function_python(S, S0, ind, P, k, alpha, b):

        if ind is None:
            return ([P, -(k * S**alpha)/(S + b)], 0.0, S0 + P)
        else:
            return ([P[ind], -(k[ind] * S**alpha[ind])/(S + b[ind])], 0.0, S0 + P[ind])

```

Figure 2.5: General approach for implementing a new reservoir element `NewFastReservoir` by extending the class `ODEsElement`.

2.3.2 *Changing the equations of the fast reservoir in M4*

Suppose the modeler wishes to modify model M4 by changing the storage-discharge equation of the fast reservoir given in equation 2.10 to a new relationship

$$Q^{(FR)} = \frac{k^{(FR)} \left(S^{(FR)}\right)^{\alpha^{(FR)}}}{S^{(FR)} + b^{(FR)}} \tag{2.11}$$

where $k^{(FR)}$, $\alpha^{(FR)}$, and $b^{(FR)}$ are model parameters.

An *element* with this storage-discharge relationship has not been implemented in SuperflexPy yet (as of version 1.2.1). The following sections give two approaches for creating such an *element*.

2.3.2.1 *General approach for creating a new reservoir with SuperflexPy*

The general approach for creating a new reservoir in SuperflexPy is to define a new class that inherits most of its functionality (methods) from the class `ODEsElement`. This operation is illustrated in the code snippet in figure 2.5 (see section 8.1 of the supplementary material for full details). The new class must override the following methods:

```
1 class NewFastReservoir(PowerReservoir):
2
3     @staticmethod
4     def _fluxes_function_python(S, S0, ind, P, k, alpha, b):
5
6         if ind is None:
7             return ([P, -(k * S**alpha)/(S + b)], 0.0, S0 + P)
8         else:
9             return ([P[ind], -(k[ind] * S**alpha[ind])/(S + b[ind])], 0.0, S0 + P[ind])
```

Figure 2.6: Simplified approach for implementing the `NewFastReservoir` by inheriting directly from class `PowerReservoir`

- `__init__`: constructor of the class. Its main purpose is to invoke the constructor of the parent class (lines 5-6) and to point to the method used to calculate the fluxes, here, `_fluxes_function_python` (see also section 2.4.3, which illustrates the efficiency benefits of using Numba-optimized methods for calculating the fluxes);
- `set_input`: takes the input fluxes in a predefined order (here, just precipitation) and assigns them a key (line 15) that is then used when setting up and solving the model equations;
- `get_output`: invokes the functionalities implemented by the ODEsElement to solve the *element* equation over the entire simulation (all time steps). Lines 20-22 get the current state of the reservoir, invoke the ODE solver, and set the state to its final value. Lines 24-28 get the output flux arrays from the numerical approximator (see section 2.4.3). Line 30 returns a list with the output of the *element* (here, the streamflow);
- `_fluxes_function_python`: calculates the fluxes for a given state, inputs, and parameters. Line 36 implements the vector version while line 38 implements the scalar version. Both versions are needed by the numerical approximator (see section 2.4.3; further practical details are provided in section 8.1 of the supplementary material).

The new *element* `NewFastReservoir` is now defined and can be used in the "new" version of M4, in lieu of the previous *element* `PowerReservoir`. The Object-Oriented features of Python are very useful here to enable the new class `NewFastReservoir` to inherit most of the methods from the base class `ODEsElement`. Otherwise, in addition to the methods listed above, we would have needed to implement many other methods, e.g., for interfacing with numerical solvers, for setting *element* parameters and states, etc.

2.3.2.2 *Simplified approach for creating a new reservoir element (from an existing element)*

The same new reservoir element can be implemented in a simpler way by noting that `NewFastReservoir` differs from `PowerReservoir` solely in the definition of the outflow equation. This difference affects only one of the four methods implemented in figure 2.5, namely `_fluxes_function_python`. A simpler implementation of `NewFastReservoir` can be therefore achieved by inheriting this class directly from class `PowerReservoir` rather than from class `ODEsElement`. The code in figure 2.6

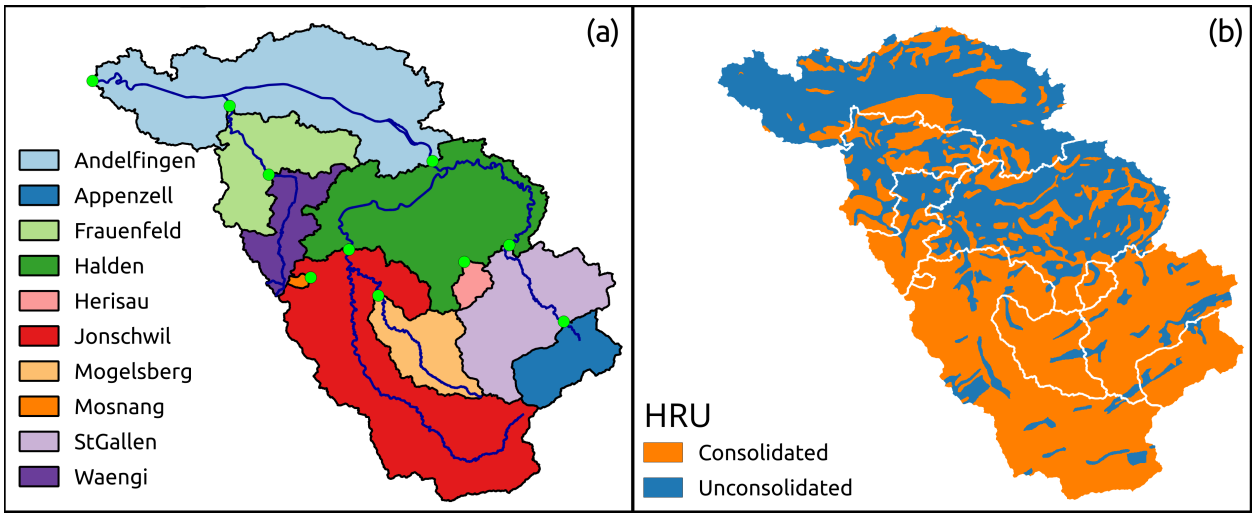

Figure 2.7: Illustration of catchment discretization used for a distributed application of SuperflexPy in the Thur catchment: **(a)** discretization into sub-catchments and **(b)** discretization into hydrological response units (HRUs) as presented in model M02 in Dal Molin et al. (2020). The panels of this figure were originally published in figures 1a and 6 of Dal Molin et al. (2020). The HRU model structure is shown in figure 2.8.

illustrates this approach and implements only the method `_fluxes_function_python`. All other methods are inherited from class `PowerReservoir`.

Note that this simplified implementation is a consequence of the required modification being relatively minor, i.e., a change solely in the constitutive function equation. More complex modifications, such as the inclusion/exclusion of input/output fluxes (e.g. inclusion of evapotranspiration into the `PowerReservoir`), would require the general implementation approach described in section 2.3.2.1.

2.3.3 *Implementing a distributed model*

This section illustrates the implementation of an HRU-based, distributed hydrological model, intended to simulate streamflow in a nested catchment. This implementation requires the entire workflow illustrated in section 2.2.2. The example is provided by model M02, developed in Dal Molin et al. (2020) to provide streamflow predictions at 10 sub-catchments of the Thur catchment in Switzerland (figure 2.7a).

Each sub-catchment receives its own forcing, namely precipitation, potential evapotranspiration, and temperature. Two HRU types are defined based on geology: consolidated and unconsolidated formations (figure 2.7b). Both HRU types are characterized by the same model structure, which is shown in figure 2.8. This HRU model structure differs from model structure M4 (section 2.3.1) in the following additional *elements*: (i) a "snow" reservoir, WR, which controls the partition of incoming precipitation between rainfall and snowfall based on temperature, (ii) a lag function between UR and FR, and (iii) a "slow" reservoir, SR, which acts in parallel to FR and is controlled by the same equations as FR but with different parameter values.

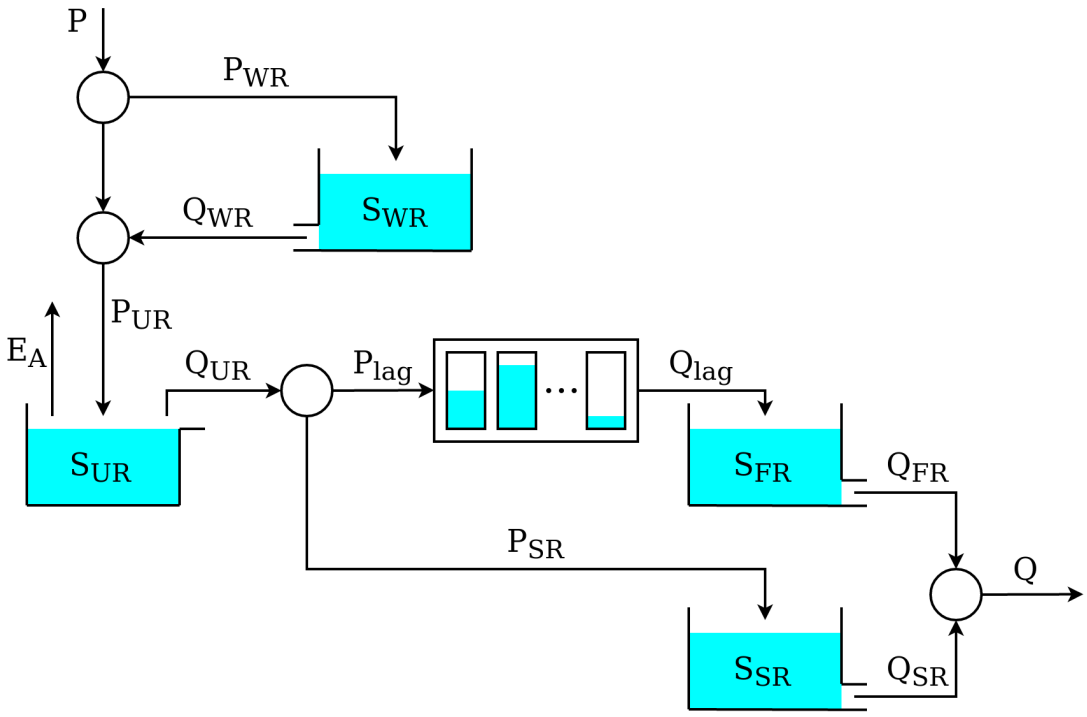

Figure 2.8: Model structure used to represent the HRUs in model M02 in Dal Molin et al. (2020). Refer to figure 2.7 for the corresponding HRU discretization of the Thur catchment.

Similar to the simpler previous example in section 2.3.1, this "lumped" model structure is implemented as a *unit*. However, a key difference is that in the previous example the *unit* represented the entire system, whereas here it is part of a more complex system.

Given the spatial organization of the model, *nodes* are used to represent sub-catchments and *units* are used to implement HRU types. Note that the sub-catchments may share (one or more) HRU types, which in SuperflexPy translates into the *nodes* sharing (one or more) *units*. The *network* level is used to connect multiple *nodes*, and enables predictions at internal catchment locations. Figure 2.9 shows the SuperflexPy representation of the spatial organization shown in figure 2.7.

We start by implementing the *units*. As seen in figure 2.8, the HRU model structure has *elements* operating in parallel and, therefore, requires the use of connections. Figure 2.10 shows how the HRU model structure is "translated" into a SuperflexPy *unit*. Recall, from section 2.2.1, that *elements* can be connected only if they belong to two consecutive layers, which implies that "gaps" in the structure must be filled using transparent elements, which output the same fluxes they receive as inputs. Splitters and junctions are used to divide and merge the fluxes to implement the parallel flow paths.

Comparing figure 2.8 with figure 2.10, we see how the HRUs structure has been implemented within SuperflexPy. The following implementation aspects are noted:

1. The incoming precipitation is partitioned into rainfall and snowfall. This partitioning is done internally in the WR element. The SuperflexPy implementation of WR takes care of two processes: (i) partitioning of precipitation into rainfall and snowfall; and (ii) simulation

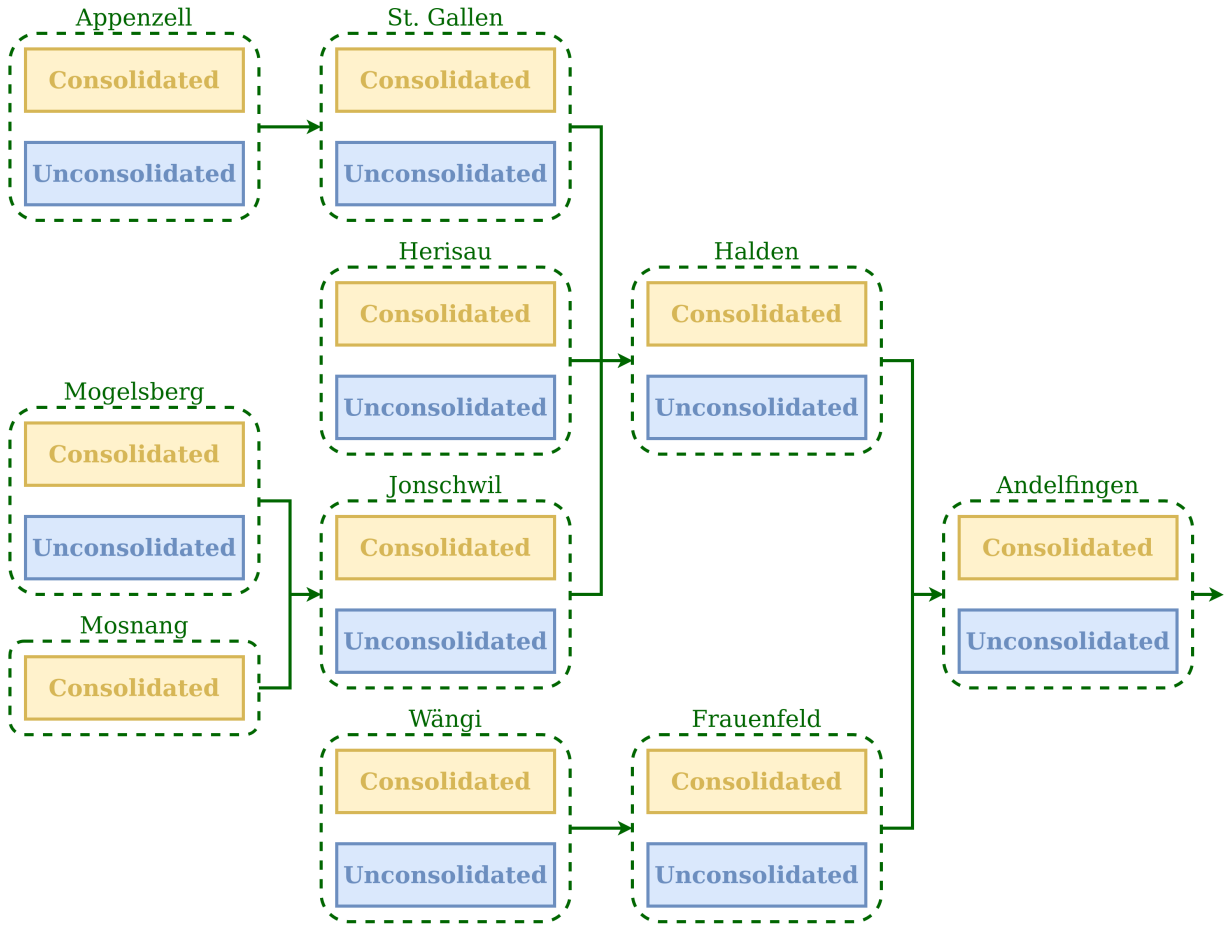

Figure 2.9: Spatial organization of the SuperflexPy model configuration used to simulate water fluxes in the Thur catchment (M02 in Dal Molin et al., 2020). The *units*, used to represent the HRUs, are shown using the blue and yellow boxes. The *nodes*, used to represent the sub-catchments, are shown using the green dashed boxes. The group of *nodes* connected together (green arrows) creates a *network*.

of snow processes (accumulation and melting). The output of WR is, logically, the sum of rainfall and snowmelt. Alternatively, a (new) splitter element could have been defined to partition the fluxes between UR (rainfall) and WR (snowfall) based on temperature.

2. WR, as currently implemented, does not receive as input the potential evapotranspiration (PET), which is needed by the downstream *element* UR. Therefore, the transfer of PET values to the UR element is implemented using a separate path composed by three *elements*, labelled "upper splitter", "upper transparent", and "upper junction" (figure 2.10). This choice simplifies the interface of element WR at the expense of a somewhat more complicated model structure with additional *elements*.

3. The parallel part of the structure is composed by two *elements* on one branch (lag and FR) and only one *element* on the other branch (SR). To satisfy the requirement of not having "gaps" in the *unit* structure, a transparent element ("lower transparent") is added after the SR.

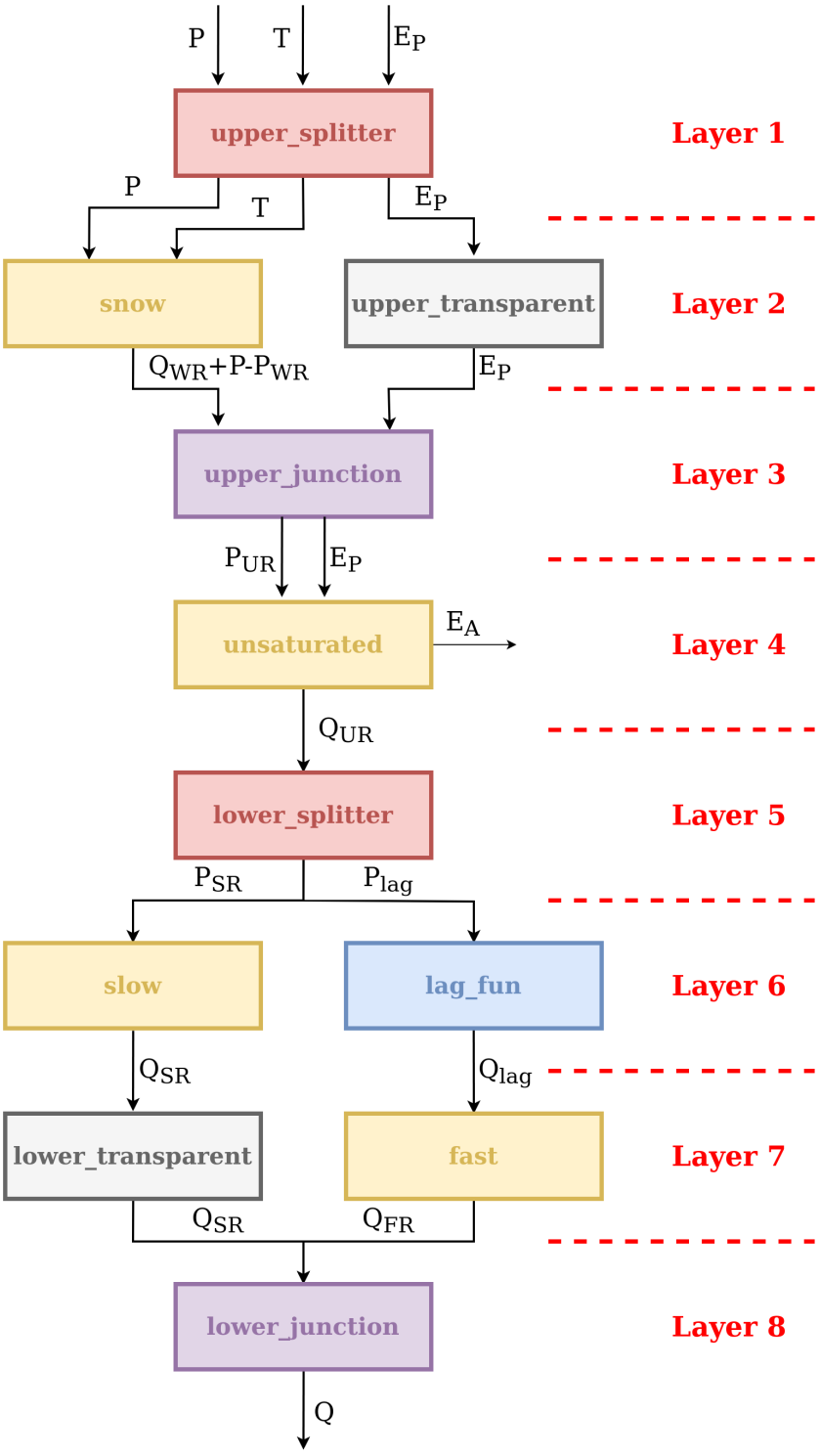

Figure 2.10: SuperflexPy representation of the model structure M02 in figure 2.8.

```python
from superflexpy.implementation.computation.pegasus_root_finding import PegasusPython
from superflexpy.implementation.computation.implicit_euler import ImplicitEulerPython
from superflexpy.implementation.elements.thur_model_hess import SnowReservoir, UnsaturatedReservoir, HalfTriangularLag, PowerReservoir
from superflexpy.implementation.elements.structure_elements import Transparent, Junction, Splitter
from superflexpy.framework.unit import Unit
from superflexpy.framework.node import Node
from superflexpy.framework.network import Network

# Initialize the elements
solver = PegasusPython()
approximator = ImplicitEulerPython(root_finder=solver)

upper_splitter = Splitter(direction=[[0, 1, None], [2, None, None]],
                          weight=[[1.0, 1.0, 0.0], [0.0, 0.0, 1.0]],
                          id='upper-splitter')
snow = SnowReservoir(parameters={'t0': 0.0, 'k': 0.01, 'm': 2.0}, states={'S0': 0.0},
                     approximation=approximator, id='snow')
upper_transparent = Transparent(id='upper-transparent')
upper_junction = Junction(direction=[[0, None], [None, 0]], id='upper-junction')
unsaturated = UnsaturatedReservoir(parameters={'Smax': 50.0, 'Ce': 1.0, 'm': 0.01, 'beta': 2.0},
                                   states={'S0': 10.0}, approximation=approximator, id='unsaturated')
lower_splitter = Splitter(direction=[[0], [0]], weight=[[0.3],  [0.7]], id='lower-splitter')
lag_fun = HalfTriangularLag(parameters={'lag-time': 2.0}, states={'lag': None}, id='lag-fun')
fast = PowerReservoir (parameters={'k': 0.01, 'alpha': 3.0}, states={'S0': 0.0},
                     approximation=approximator, id='fast')
slow = PowerReservoir (parameters={'k': 1e-4, 'alpha': 1.0}, states={'S0': 0.0},
                     approximation=approximator, id='slow')
lower_transparent = Transparent(id='lower-transparent')
lower_junction = Junction(direction=[[0, 0]], id='lower-junction')

# Initialize the HRUs
consolidated = Unit(layers=[[upper_splitter], [snow, upper_transparent], [upper_junction],
                            [unsaturated],  [lower_splitter],  [slow, lag_fun],
                            [lower_transparent, fast], [lower_junction]],
                    id='consolidated')
unconsolidated = Unit(layers=[[upper_splitter], [snow, upper_transparent], [upper_junction],
                              [unsaturated],  [lower_splitter],  [slow, lag_fun],
                              [lower_transparent, fast], [lower_junction]],
                      id='unconsolidated')

# Create the catchments
andelfingen = Node(units=[consolidated, unsaturated], weights=[0.24, 0.76], area=403.3, id='andelfingen')
appenzell = Node(units=[consolidated, unsaturated], weights=[0.92, 0.08], area=74.4, id='appenzell')
frauenfeld = Node(units=[consolidated, unsaturated], weights=[0.49, 0.51], area=134.4, id='frauenfeld')
halden = Node(units=[consolidated, unsaturated], weights=[0.34, 0.66], area=314.3, id='halden')
herisau = Node(units=[consolidated, unsaturated], weights=[0.88, 0.12], area=16.7, id='herisau')
jonschwil = Node(units=[consolidated, unsaturated], weights=[0.9, 0.1], area=401.6, id='jonschwil')
mogelsberg = Node(units=[consolidated, unsaturated], weights=[0.92, 0.08], area=88.1, id='mogelsberg')
mosnang = Node(units=[consolidated], weights=[1.0], area=3.1, id='mosnang')
stgallen = Node(units=[consolidated, unsaturated], weights=[0.87, 0.13], area=186.6, id='stgallen')
waengi = Node(units=[consolidated, unsaturated], weights=[0.63, 0.37], area=78.9, id='waengi')

# Create the network
thur_catchment = Network(nodes=[andelfingen, appenzell, frauenfeld, halden, herisau,
                                jonschwil, mogelsberg, mosnang, stgallen, waengi],
                         topology={'andelfingen': None, 'appenzell': 'stgallen',
                                   'frauenfeld': 'andelfingen', 'halden': 'andelfingen',
                                   'herisau': 'halden', 'jonschwil': 'halden',
                                   'mogelsberg': 'jonschwil', 'mosnang': 'jonschwil',
                                   'stgallen': 'halden',  'waengi': 'frauenfeld'})
```

Figure 2.11: SuperflexPy code implementing the distributed model in figures 2.9 and 2.10.

The code to setup this model is detailed in figure 2.11. Similar to the earlier example in section 2.2.2, the user initializes and connects all model *components*, proceeding sequentially from the lowest level (*elements*) to the highest level (*network*). The procedure can be summarized as follows:

1. lines 10-29: Initialize the *elements* needed for the lumped model structures used in the HRUs;

2. lines 32-39: Initialize the *units* used to represent the HRUs, linking all the *elements*;

3. lines 42-51: Initialize the *nodes* used to represent the sub-catchments. Both *units* are assigned to 9 *nodes*; the Mosnang sub-catchment contains a single HRU and hence only a single *unit* is assigned to the corresponding *node* (line 49).

4. lines 54-60: Connect the *nodes* using a *network*. The topology of the *network* is defined by indicating, for each *node*, the downstream one.

The *network* runs the *nodes* from upstream to downstream, collects their outputs, and routes them to the outlet. Customized routing functions can be implemented, as shown in section 9.1 of the supplementary material. The output of the *network* is a Python dictionary, with keys given by the *node* identifiers and values given by the list of Numpy arrays representing the time series of output fluxes over the simulation period.

2.4 IMPLEMENTATION DETAILS OF SUPERFLEXPY

This section presents additional technical details of SuperflexPy needed to understand better some aspects of the functioning of the framework. A more detailed and practical description is provided in the supplementary material.

2.4.1 *Parameters and states*

All SuperflexPy *components* can have parameters and states. Parameters specify *component* characteristics, whereas states keep track of the *component* history. States and parameters are set as part of initializing the model *components* and can be manipulated using `get` and `set` methods provided by the framework at all levels of its hierarchy (see the example in section 2.2.2).

The parameters can be either constant or variable in time. Constant parameters represent the most common set up of hydrological models. In conceptual hydrological modelling, time-varying parameters have been proposed to represent "deterministic" system variability (e.g. seasonality, Westra et al., 2014) and/or "stochastic" system variability (e.g., Reichert and Mieleitner, 2009; Renard et al., 2011; Kuczera et al., 2006); see also earlier work in data-based mechanistic modelling (e.g., Young, 2000).

2.4.2 *Modular design following the Object-Oriented paradigm*

As noted in section 2.2.1, SuperflexPy embraces the object-oriented paradigm (e.g., Meyer, 1988), which is widely used in general software and is increasingly adopted in scientific software.

The object-oriented design provides several advantages in the context of SuperflexPy:

- The inheritance principle enables the creation of new classes by extending existing ones. Inheritance reduces drastically the amount of new code that needs to be generated to implement a new model *component* (see example in section 2.3.2);
- Changes to a class (e.g. a *component*) and the creation of new classes can be carried out in isolation from the rest of the code, as long as the interfaces between classes are respected;
- When creating a model, only the necessary objects need to be initialized and used. This principle makes the model configuration effort roughly proportional to required model complexity, i.e., simple model structures can be constructed from the minimal set of required *components*;
- Objects retain their history (states), which can be accessed post-run to undertake model analysis and/or subsequent computation;
- The modular nature of objects facilitates the development and testing of new code.

These benefits make it easier to achieve clean and maintainable code, which is essential for any practical modelling framework.

2.4.3 *Numerical solution of ODEs*

The mass balance of reservoir elements is described using ordinary differential equations (ODEs), which are typically solved (approximately) using numerical time-stepping algorithms. Many such algorithms have been described in the numerical methods literature, e.g. Euler methods, Runge-Kutta methods, etc. (e.g., Butcher, 2016).

SuperflexPy separates the formulation of model equations from the solution of these equations. More specifically, flux equations are defined internally as methods of the *elements* (as shown in section 2.3.2), while the numerical algorithm is specified externally (to the *element*) by defining a so-called "numerical approximator". The numerical approximator is a procedure, which constructs a numerical approximation of the differential equation(s) controlling the *element*. If the numerical approximator implements an implicit time stepping scheme, it will generally require an auxiliary "root finder", which is a procedure that solves nonlinear algebraic equation(s). The separation of equations and solvers in the model specification enables the modeler to select the numerical method without making any changes to the governing model equations. Further details are provided in section 5.1 of the supplementary material.

SuperflexPy provides two built-in numerical approximators, namely the fixed-step implicit and explicit Euler time stepping schemes (e.g., Clark and Kavetski, 2010). The implicit Euler equations are solved using the Pegasus root finder (Dowell and Jarratt, 1972). The user can implement additional numerical algorithms, either by coding them directly or by interfacing with external code (e.g. ODE solvers from SciPy). As detailed next in section 2.4.4, the choice of numerical implementation, and its compatibility with optimizing compilers, may have a strong impact on the overall computational speed of the model.

2.4.4 *Computational efficiency and language choice*

Computational efficiency is a key requirement of a practical modelling framework. Model calibration via parameter optimization is a common computationally demanding task required by most hydrological models, typically requiring hundreds or thousands of model runs. Moreover, conceptual hydrological models are often used in Monte Carlo uncertainty quantification, with comparable or even larger computational cost (up to millions of model runs in some cases).

The choice of programming language inevitably requires trade-offs between computational efficiency and ease of use. The choice of Python for SuperflexPy was motivated by the attraction of a flexible and widely used scripting language in conjunction with two efficient numerical libraries: Numpy (van der Walt et al., 2011) and Numba (Lam et al., 2015). Numpy provides highly efficient arrays for vectorized operations (i.e. element-wise operations between arrays). Numba provides a "just-in-time compiler" that compiles (at runtime) a Python method into machine code that interacts efficiently with Numpy arrays.

The combined use of Numpy and Numba is particularly effective when solving ODEs, where the numerical algorithm performs element-wise sequential operations. The built-in SuperflexPy approaches for solving ODEs are compatible with such numerical infrastructure, and therefore enable fast computation times. Note that switching to ODEs solvers that do not take advantage of such libraries might dramatically increase the model runtime.

Numba offers drastic computational speed ups compared to native Python; our experimentation suggests runtime reductions by factors of up to 30. However, a drawback of Numba is the requirement to compile the code each time it is executed (run). For a lumped model composed of a few reservoirs, the Numba compilation time is of the order of a few seconds. Therefore, Numba will outperform Python when the simulation is long (e.g. multiple years of hourly data) and/or when the model needs to be run a large number of times. For example, as a broad illustration of runtimes on a standard laptop, calibration of a HYMOD-like SuperflexPy model to observed daily data, requiring 1000's of model runs each with 1000 time steps, takes a few seconds with the Numba implementation compared to a couple of minutes with native Python execution. Note that here we refer to the runtime of the SuperflexPy model itself, and exclude the runtime of the calibration tool procedures; more details on benchmarking are given in section 5.3 of the supplementary material. Examples of interoperability of SuperflexPy with external libraries for

model calibration (e.g., SPOTPY, Houska et al., 2015) are given in chapter 14 of the supplementary material.

2.4.5 *Ability to represent multiple fluxes and states*

SuperflexPy can operate with multiple fluxes and state variables. In particular, connection elements, *units*, *nodes*, and the *network* can accommodate an arbitrarily large number of fluxes. The use of multiple fluxes has been already shown in the model structure described in section 2.3.3, where the `upper_splitter` handles three different variables (precipitation, temperature, and PET). Additional examples are provided in the supplementary material (e.g., chapters 10, 11).

The capability to simulate multiple fluxes and states is intended to support the extension of SuperflexPy to new modelling scenarios. Several such scenarios may be of interest, including the transport of chemical substances (e.g., Fenicia et al., 2010; Ammann et al., 2020), the interaction between frozen and liquid water in a snow element (e.g., Jansen et al., 2021), interactions in the saturated/unsaturated soil zones (e.g., Seibert et al., 2003), and so forth.

While the current examples in SuperflexPy do not include all the cases listed above, the framework architecture anticipates the need for more general simulation functionality, and has been designed to support extension to accommodate such multi-state processes.

2.5 DISCUSSION

2.5.1 *Balancing functionality, scope, and usability in a flexible model implementation*

A software implementation that maximizes flexibility and usability is challenging to achieve, because flexible modelling functionality may increase configuration effort and computational cost. Existing flexible frameworks have approached this tradeoff with different priorities, based on their respective modelling objectives and paradigms.

The following sections offer a brief discussion of the design choices made by SuperflexPy in the context of selected existing frameworks with a similar scope. The discussion makes use of table 2.1 and table 2.2, which summarize key design choices related to usability and simulation capabilities respectively.

Table 2.1: Summary of usability characteristics of SuperflexPy in the context of selected flexible frameworks for conceptual hydrological modelling.*

	Availability	Distribution and installation	Documentation	Interface and setup	I/O format for settings and data	Possibility of customization	Built-in calibration and uncertainty analysis
SuperflexPy	Open source	Python package	Available	Python package. Python script to setup	Direct I/O with Python. No binding to particular formats	Possible with moderate programming expertise	Not present
FUSE (Fortran) (2008)	Exe or code, by request from authors	Standalone exe/code	Comments in code (limited)	Executable with/without GUI, or Fortran subs. Setup files	Structured text files	Possible but not supported systematically	Some versions are coupled with optimization and MCMC sampling tools
SUPERFLEX-F90 (2011)	Exe or code, by request from authors	Standalone exe	Comments in code (limited)	CLI or DLL or Fortran subs. Setup files	Structured text files	Possible but not supported systematically	Not present
CMF (2011)	Open source	Python package. Code compilation for enhancements	Available	Python package. Python script to setup. GUI only for lumped models	Direct I/O with Python. No binding to particular formats	Customization using C++. Possibility with Python under development	No. Developers recommend to use the SPOTpy package from the same group

Continued on next page

Continued from previous page

	Availability	Distribution and installation	Documentation	Interface and setup	I/O format for settings and data	Possibility of customization	Built-in calibration and uncertainty analysis
PERSiST (2014)	Exe/webapp after registration	Standalone executable or webapp	Exists. Not public at the moment	Desktop app or webapp. Setup files or GUI	Structured text files and XMLs	Possible but not supported systematically	Incorporates MCMC toolkit
ECHSE (2015)	Open source	R package to generate C code that has to be compiled	Available	CLI. Setup through text file or CLI	Delimited text files	Possible with moderate programming expertise	Not present
MARRMoT (2019)	Open source	Matlab/Octave package	Available	Collection of scripts and functions. Setup with script.	Direct I/O with Matlab/Octave. No binding to particular formats	Possible with moderate programming expertise	Not present
RAVEN (2020)	Open source	Standalone executable. May require NetCDF	Available	Executable without GUI. Setup files	Structured text files	Possible but requires developer-level expertise. Instructions in the documentation	DDS optimization. Reports model performance metrics usable by external software

*This information was collated based on published information. A brief informal review was provided by the framework authors.
Abbreviations: exe = binary executable, subs = subroutines, GUI = graphical user interface, CLI = command line interface, DLL = dynamic link library, MCMC = Monte Carlo Markov Chain, DDS = Dynamic Dimensioned Search

Table 2.2: Summary of simulation capabilities of SuperflexPy in the context of selected flexible frameworks for conceptual hydrological modelling.*

	Structural flexibility	Spatial flexibility	Hydrological processes	Numerical solution options	Pre and post processing	Programming language
SuperflexPy	Components can be connected freely	Lumped; semi-distributed	Water fluxes; Designed to handle multiple fluxes	Fixed step implicit and explicit Euler. Possibility to use custom solvers	Not available	Python
FUSE (Fortran) (2008)	Master structure; components selected for each model decision	Lumped	Water fluxes	Implicit, semi-implicit, explicit schemes; fixed and adaptive step solvers	Not available	Fortran
SUPERFLEX-F90 (2011)	Master structure; components can be turned on/off	Lumped; semi-distributed	Water fluxes; transport processes	Fixed step implicit and explicit Euler	Not available	Fortran
CMF (2011)	Components can be connected freely	Lumped; semi-distributed; fully-distributed	Water fluxes; transport processes	Implicit and explicit schemes; single or multi-step solvers	Calculation methods for PET	Python wrapping of C++ code

Continued on next page

Continued from previous page

	Structural flexibility	Spatial flexibility	Hydrological processes	Numerical solution options	Pre and post processing	Programming language
PERSiST (2014)	Components can be connected freely	Semi-distributed	Water fluxes; designed to be coupled with transport models (INCA)	Implemented as a series of first order difference equations	PET calculated internally	C++
ECHSE (2015)	Components can be connected freely	Lumped; semi-distributed; grids	Water fluxes; transport processes	To be implemented by user when defining the components	Not available	C++; R package to generate C++ code
MARRMoT (2019)	Library of model structures. Possibility to combine different components	Lumped	Water fluxes	Fixed step implicit and explicit Euler. Possibility to use custom solvers	Not available	Matlab/Octave
RAVEN (2020)	Components can be connected freely	Grids; subbasin/HRUs; triangulated irregular network	Water fluxes; transport processes	Ordered series, Euler, and predictor/corrector global methods; local methods at process level	Calculation and interpolation (spatial and temporal) of derived fluxes and other variables	C++

*This information was collated based on published information. A brief informal review was provided by the framework authors.

2.5.1.1 *Structural flexibility*

Structural flexibility refers to the flexibility in how *elements* can be connected to compose the structure of the model (i.e., of the *unit*, following SuperflexPy's terminology). This consideration applies both to lumped and distributed models; the flexibility in specifying the spatial organization of the model is considered separately in section 2.5.1.2.

Some flexible frameworks are implemented using a master structure that incorporates all supported model configurations. In these implementations, the user can choose the flux equation(s) (e.g., FUSE, SUPERFLEX-F90) and/or activate/deactivate specific *elements* (e.g., SUPERFLEX-F90), but cannot change the overall connectivity of model *elements*. To the extent that the master structure is sufficiently general, it may not unduly restrict the practical usage of the framework.

Other frameworks (e.g., MARRMoT) propose a collection of existing conceptual model structures ready to use, which have been implemented following the same design rules in order to allow for a fair comparison. Such frameworks are typically intended for model intercomparison studies.

The most general frameworks allow connecting the *elements* freely without constraints. A distinction can be made between frameworks that allow for mutual interactions between the *elements* (e.g., CMF) and frameworks that do not allow such interactions (e.g., ECHSE).

SupeflexPy adopts the latter philosophy, allowing to connect the *elements* freely within the *unit* but restricting mutual interactions, i.e., constraining the structure to be a DAG (see section 2.5.2). Moreover, we have chosen to define the DAG as a succession of *layers*, listing the *elements* in order from upstream to downstream and allowing for parallel flow paths (e.g., see the model structure in figure 2.10). This "list" formulation has been selected in preference to other methods for defining a graph, e.g., connectivity matrix, adjacency list, etc., for the following reasons: (i) simplicity/scalability, as the list dimension scales linearly with the number of *elements*, in contrast to the connectivity matrix approach where this scaling is quadratic; (ii) arguably better readability, as the *elements* are listed in the order they appear in the DAG; and (iii) it guarantees a graph topology without loops. Note that other popular modelling tools (e.g., neural networks) adopt this type of formulation.

2.5.1.2 *Spatial flexibility*

Most frameworks (e.g., CMF, ECHSE, SUPERFLEX-F90, etc.) support multiple types of spatial discretization (e.g., lumped, HRUs, sub-catchments, grids, etc.). Some frameworks (e.g., FUSE, MARRMoT) support solely lumped models.

SuperflexPy uses 4 hierarchical levels of *components*, intended to facilitate the formulation of models that range in spatial complexity from a simple lumped model, to a composition of lumped models intended for prediction at a single location (e.g. a catchment with several HRUs), and ultimately to a distributed model capable of making predictions at multiple internal locations. The use of a hierarchical set of *components* could be contrasted to a framework based solely on the lowest level *components*, here, *elements*. The use of higher level *components* enables the modeler to

capture explicitly the natural groupings in the catchment of interest, e.g., sub-catchments, HRUs, etc.

2.5.1.3 *Usability*

The usability of a framework can be judged according to several aspects.

The first aspect is how a framework is operated. Some frameworks are standalone and operated through a graphical interface (e.g., PERSiST) or the command line interface (e.g., SUPERFLEX-F90). Other frameworks are designed as libraries that can be called from the user code in a specific programming language to initialize, configure and run the model (e.g., CMF, MARRMoT; SUPERFLEX-F90 also allows this option when using the source code from Fortran). SuperflexPy is implemented as a Python package. Models can be created using a Python script and interfaced easily with external libraries (examples are provided in chapter 14 of the supplementary material).

The second aspect is the scope of the framework. Most frameworks (e.g., SUPERFLEX-F90, ECHSE) adopt, by design, the philosophy of "one tool per problem" and limit their functionality to the simulation of hydrological processes. Other frameworks integrate tools for parameter calibration and sensitivity analysis, uncertainty quantification, pre- and post-processing tasks such as input unit checks/conversions, etc. (e.g., RAVEN, PERSiST). SuperflexPy adopts the first philosophy: it limits its functionality to hydrological simulation.

Finally, documentation is another key aspect in the usability of a framework. Virtually all considered frameworks provide such documentation to a varying degree of detail. SuperflexPy documentation is available online and explains in detail how to use and further develop the framework.

Figure 2.12 shows the online software management tools that are used to develop and deploy SuperflexPy. The framework itself, including source code, documentation, examples, etc., is hosted on GitHub. Automated workflows are then used to create new releases (PyPI), get DOIs for the software releases (Zenodo), host the documentation (ReadTheDocs), and run the examples (Jupyter and Binder). From a general user perspective, this setup improves model accessibility and reproducibility. From a developer and contributor perspective, it reduces the effort needed to maintain and extend the framework.

2.5.1.4 *Possibility of extension and customization*

Most frameworks have open source code and permissive licenses, making it possible to modify and extend their codebase. Within this category, some frameworks are specifically intended to be customized (e.g., implementing new functionalities) as part of their regular usage without an expectation of "developer-level" skills (e.g., ECHSE). Other frameworks do not envisage customization in their primary scope, but can still be modified by modelers with appropriate programming expertise in consultation with available developer guides (e.g., RAVEN).

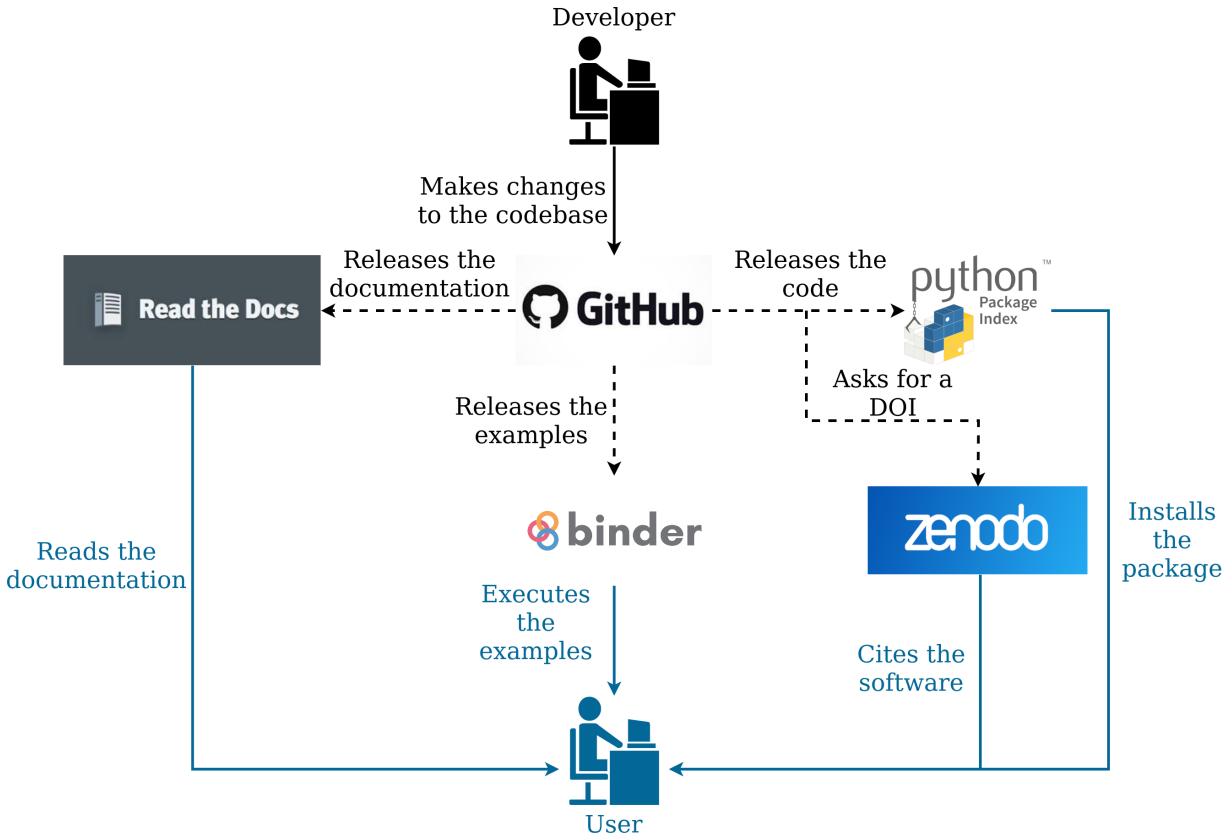

Figure 2.12: Organization of the SuperflexPy project, indicating the online software management tools used to develop the source code and documentation, release product versions with associated DOIs, and provide general open access to all project components. Dashed lines represent automated workflows.

Some frameworks have not been released as open source, and the only way to access their codebase for customization and extension is by contacting their developers (e.g., SUPERFLEX-F90, PERSiST).

SuperflexPy is designed to facilitate extension and customization as part of its regular usage. New *components* can be created by extending or modifying existing *components*, as demonstrated in section 2.3.2.

2.5.1.5 *Computational efficiency*

The computational efficiency of a model code, i.e., the time required to run a simulation, depends primarily on two aspects, namely the programming language and the numerical algorithms.

In terms of programming languages, most frameworks have been implemented in C/C++ and Fortran, which enable very fast computation. These implementations can be either purely single-language (e.g., FUSE, RAVEN), or wrapped within a scripting language to provide a more suitable interface (e.g., CMF). Among the considered existing frameworks, only MARRMoT is implemented entirely in an interpreted language (Matlab/Octave).

In terms of numerical algorithms, a wide range of options are available for solving differential equations. Broadly speaking, time stepping algorithms can be classified as implicit or explicit, and may employ fixed or adaptive step size. The choice of algorithm and its settings brings tradeoffs between solution accuracy, algorithm complexity and computational cost. In the context of model development and comparison, it is important to separate the specification of model equations from the choice of numerical solution and to use robust numerical methods to avoid spurious artefacts (e.g., Kavetski and Clark, 2010). The majority of frameworks implement this separation and provide a choice of built-in numerical algorithms.

SuperflexPy, while written entirely in Python (a nominally "slow" language), makes several implementation choices to reduce computational costs. These choices include the use of efficient numerical libraries (section 2.4.4) and the solution of the *elements* in succession (DAG, section 2.5.2). The choice of numerical algorithm for individual *elements* is left to the user (section 2.4.3). The (recommended) built-in approximators include the implicit Euler scheme with fixed step size, which offers stability and smoothness benefits valuable in parameter estimation contexts.

2.5.2 *Current restrictions in model structure specification*

As part of balancing the flexibility, ease of use, and computational performance of SuperflexPy, some restrictions have been imposed on the connectivity between model *components*.

The first restriction is that *elements* within a *unit* must form a directional acyclic graph (DAG), with no feedback loops from downstream to upstream elements (section 2.2.1). This restriction enables the numerical solvers to proceed, at each time step, in a single pass from upstream to downstream elements and improves the computational performance of the framework. The restriction on internal model feedbacks is not expected to be overly limiting when developing conceptual hydrological models, where the fluxes from a given *element* typically depend only on the state in that *element* and not on downstream elements. In such systems, flows occur only in one direction, e.g. in model M4 the water flows from UR to FR but not vice versa. A counter-example where internal model feedbacks are required is given by the bidirectional interaction between surface water and groundwater in the hyporheic zone, where the exchange flux (or fluxes) depends on both states. Such interactions can still be modelled in SuperflexPy by introducing *elements* that embed feedbacks internally. For example, the hyporheic zone can be represented using a two-state reservoir with interacting states (e.g., Seibert et al., 2003). In other words, the SuperflexPy restriction on model feedbacks applies to interactions between *elements*, but not to interactions within an *element*.

The second restriction, which also applies at the *unit* level, derives from the decision to define the DAG as a succession of *layers* (section 2.5.1.1). This choice simplifies the model definition in typical use cases, when there are many *elements* with relatively few connections (i.e., the DAG is "sparse" rather than "dense"). However, the definition of a DAG as a succession of *layers* requires the *elements* to be connected directly one to the other, without skipping *layers*. Hence

the need for transparent elements, which output the inputs they receive and are used to fill the gaps that arise when two or more parallel flow paths have a different number of *elements*. An example of such model configuration is given in figure 2.10, where a transparent element (labeled "`lower_transparent`") is used to fill the gap in *layer* 7.

The third restriction is that the topology of a *network* must represent a tree where any given *node* can connect and transfer fluxes only to a single downstream *node* (section 2.2.1). This restriction has a similar motivation to the restriction of a *unit* structure to a DAG, and allows for a simple and efficient computational implementation, which starts from the headwater *nodes* and proceeds downstream one *node* at a time. Typical distributed conceptual models meet this restriction, for example as illustrated in section 2.3.3. However, fully integrated distributed models, such as Parflow and Mike-She, do include mutual dependencies between spatial elements, e.g., leading to 2D or 3D groundwater flows. Such configurations are considered beyond the scope of conceptual distributed models, and therefore are not currently supported in SuperflexPy.

2.5.3 *Current usage and future developments*

SuperflexPy is easy to install and run; it is written in pure Python and its dependencies are limited to the packages Numpy and Numba (section 2.4.4). Installation can be done directly using the package installer for Python (pip) and does not require (additional) external libraries. We stress that SuperflexPy is not a wrapper of earlier SUPERFLEX-F90 code but offers a completely new implementation that is not constrained by choices taken in the earlier code versions.

SuperflexPy has already been used for research applications. Jansen et al. (2021) performed a "model mimicry" study where similarities and differences within the HBV family of models were investigated. SuperflexPy was used to construct a set of HBV-like models and compare them in terms of the behavior of individual model components, the impact of numerical implementation, and so forth. A list of publications using SuperflexPy is maintained on the documentation website.

In terms of future developments, we hope that SuperflexPy offers the broader hydrological community a versatile new tool for research work and practical applications. Further SuperflexPy developments are likely to follow from such work and collaborations, including: (i) expansion of the library of model *components* beyond the ones here presented (as shown in the example in section 2.3.2), and (ii) more fundamental developments in response to future model applications. It is important to highlight that SuperflexPy can be used to create and combine new model *components*, thereby enabling experimentation with new model structures and general conceptualizations. The framework, therefore, is not limited to *components* and structures taken from existing models – though such collections could be also produced. The SuperflexPy model library may grow as new users share their implementations with the community. In order to facilitate the use of SuperflexPy, its code is accessible on GitHub with license LGPL-3.0 and distributed using the Python package installer PyPI (see the code availability section at the end of this paper). The

online documentation provides a guide for colleagues interested in contributing to the framework (section 2.1 of the supplementary material).

2.6 SUMMARY AND CONCLUSIONS

SuperflexPy is a new Python flexible modelling framework for building conceptual catchment-scale hydrological models ranging from lumped to distributed configurations. SuperflexPy offers detailed control over each aspect of model configuration, and caters to a wide range of typical conceptual model applications. In order to facilitate the model building process, the framework defines its *components* (building blocks) at four hierarchical levels, namely *element*, *unit*, *node*, and *network*. These *components* support conceptual model setups of increasing levels of complexity, including but not limited to: a single element model (e.g. a reservoir), a typical lumped model (e.g. a collection of interconnected reservoirs), a semi-distributed model designed to provide prediction at a single outlet, and a semi-distributed model designed to provide predictions at internal sub-catchments. The construction of a model from *components* up to a given hierarchical level does not require specifying *components* at higher levels, which makes the model configuration effort proportional to the complexity of the application and reduces configuration/computational overheads. The framework supports multiple states and fluxes in each *component*, which facilitates future extension to applications where such functionality is needed.

SuperflexPy offers an open source implementation of the SUPERFLEX principles (Fenicia et al., 2011) that builds on the collective experience of the authors and their colleagues in hydrological model design and application. The paper discusses the key design choices made in SuperflexPy, with emphasis on the ease of use and interfacing, availability, amenability of extensions, and computational efficiency.

The use of the SuperflexPy framework is illustrated using two examples that represent typical tasks in conceptual hydrological modelling: the implementation of a lumped model to simulate an entire catchment, and the implementation of a distributed model to simulate a system of multiple sub-catchments with spatially varying landscape characteristics. We hope the framework will contribute to ongoing efforts in the hydrological modelling community to develop more robust and representative models. The framework is open source, available with license LGPL-3.0 on GitHub.

CODE AVAILABILITY

The source code of SuperflexPy, together with documentation and examples, is hosted in the public GitHub repository `https://github.com/dalmo1991/superflexPy`. Github is used for issue-tracking. Package releases are distributed using the Python package index `https://pypi.org/project/superflexpy`. Releases are identified using a version number based on Semantic Versioning 2.0.0 and assigned a DOI through Zenodo. The release associated with this

paper represents version 1.2.1 and has DOI https://doi.org/10.5281/zenodo.4698469. Detailed documentation is available through Read the Docs at https://superflexpy.readthedocs.io. The supplementary material to this paper represents a snapshot of the documentation at the time reported on the front page.

SuperflexPy is implemented using Python 3.7 and depends on Numpy (version 1.19) and Numba (version 0.50).

SuperflexPy is available under the license LGPL-3.0. Users of the framework are invited to share their modelling solutions with the community by contributing to the GitHub repository.

3

UNDERSTANDING DOMINANT CONTROLS ON STREAMFLOW SPATIAL VARIABILITY TO SET UP A SEMI-DISTRIBUTED HYDROLOGICAL MODEL: THE CASE STUDY OF THE THUR CATCHMENT

This study documents the development of a semi-distributed hydrological model aimed at reflecting the dominant controls on observed streamflow spatial variability. The process is presented through the case study of the Thur catchment (Switzerland, 1702 km^2), an alpine and pre-alpine catchment where streamflow (measured at 10 subcatchments) has different spatial characteristics in terms of amounts, seasonal patterns, and dominance of baseflow. In order to appraise the dominant controls on streamflow spatial variability and build a model that reflects them, we follow a two-stage approach. In a first stage, we identify the main climatic or landscape properties that control the spatial variability of streamflow signatures. This stage is based on correlation analysis, complemented by expert judgement to identify the most plausible cause–effect relationships. In a second stage, the results of the previous analysis are used to develop a set of model experiments aimed at determining an appropriate model representation of the Thur catchment. These experiments confirm that only a hydrological model that accounts for the heterogeneity of precipitation, snow-related processes, and landscape features such as geology produces hydrographs that have signatures similar to the observed ones. This model provides consistent results in space–time validation, which is promising for predictions in ungauged basins. The presented methodology for model building can be transferred to other case studies, since the data used in this work (meteorological variables, streamflow, morphology, and geology maps) are available in numerous regions around the globe.

This chapter is based on the publication: Dal Molin, M., Schirmer, M., Zappa, M., and Fenicia, F. (2020) "Understanding dominant controls on streamflow spatial variability to set up a semi-distributed hydrological model: the case study of the Thur catchment." *Hydrology and Earth System Sciences*, 24, 1319–1345.

3.1 INTRODUCTION

Semi-distributed rainfall–runoff models are widely applied in operation for applications such as flood forecasting (e.g., Ajami et al., 2004) or developing sustainable irrigation practices (e.g., McInerney et al., 2018). The main purpose of these models is to simulate streamflow at a limited number of fixed points along river channels (e.g., Boyle et al., 2001), and for this reason they are characterized by a coarser spatial resolution than fully distributed models, which allow a very detailed representation of the spatial variability of catchment processes. Compared to fully distributed models, they are characterized by lower data and computational requirements, which represents a valuable practical advantage in their operational use.

Similarly to the case of lumped models, the parameters of semi-distributed models are estimated via calibration. Therefore, it is important that the structure of these models is commensurate with the available data, including length, timescale, and spatial distribution (Wooldridge et al., 2001). However, semi-distributed models, even when used for similar applications such as streamflow predictions, differ significantly in terms of their process representation as well as the number of calibration parameters. For example, Gao et al. (2014) assume topography to be a dominant control on hydrological processes, whereas the SWAT model (Arnold et al., 1998) emphasizes the role of soil properties. These differences raise the question of how to select an appropriate model structure for the data at hand, which requires understanding of the association between model parameters and the climatological and geomorphological characteristics of the catchment.

Understanding the relationship between climate, landscape, and catchment response is a common objective of many research areas in hydrology, including comparative hydrology (e.g., Falkenmark and Chapman, 1989), model regionalization (e.g., Parajka et al., 2005), catchment classification (e.g., Thorsten et al., 2007), and prediction in ungauged basins (e.g., Hrachowitz et al., 2013). In the case of streamflow, the attempt to explain its spatial variability is typically accomplished either using statistical approaches, which are designed to regionalize selected characteristics of the hydrograph (streamflow signatures), or through hydrological models that account for relevant spatial information. In particular, statistical approaches such as regression analysis (e.g., Berger and Entekhabi, 2001; Bloomfield et al., 2009) and correlation analysis (e.g., Trancoso et al., 2017), or machine learning techniques like clustering (e.g., Sawicz et al., 2011; Toth, 2013; Kuentz et al., 2017), are used to group together catchments that present similar characteristics and to extrapolate the signatures where unknown. Such approaches have been useful for quantifying the hydrological variability and identifying its principal drivers. However, they are often not designed to discover causality links and can be affected by multicollinearity, which arises when multiple factors are correlated internally and with the target variable (Kroll and Song, 2013).

By incorporating spatial information about meteorological forcings and landscape characteristics, semi-distributed hydrological models have the ability to mimic the mechanisms that influence hydrograph spatial variability. However, identifying the relevant mechanisms is challenging. One

possibility is to be as inclusive as possible in accounting for all the catchment properties that are, in principle, important in controlling catchment response. However, this approach leads to models that tend to be data demanding and contain many parameters. For example, Gurtz et al. (1999) considered several landscape characteristics (elevation, land use, etc.) in their application of a semi-distributed model to the Thur catchment (Switzerland), which resulted in a model with hundreds of hydrological response units (HRUs) that were defined a priori based on the complexity of the catchment. The other option is to try to identify the most relevant processes and neglect others in order to control model complexity. For example, Fenicia et al. (2016) compared various model hypotheses to determine an appropriate discretization of the catchment in HRUs and appropriate structures for different HRUs. Antonetti et al. (2016) used a map of dominant runoff processes following Scherrer and Naef (2003) for defining HRUs. However, these approaches require a good experimental understanding of the area, which is not always available.

Convincing model calibration–validation strategies are essential to provide confidence that the model ability to fit observations is a reflection of model realism and not a consequence of calibrating an overparameterized model (e.g., Andréassian et al., 2009). A common approach for the calibration of semi-distributed models is the so-called "sequential" approach, where subcatchments are calibrated sequentially from upstream to downstream (e.g., Verbunt et al., 2006; Feyen et al., 2008; Lerat et al., 2012; De Lavenne et al., 2016). Although this approach may provide good fits and therefore has its practical utility where data are available, it does not provide understanding of the causes of streamflow spatial variability and results in models that are not spatially transferable. Moreover, such models are prone to contain many parameters, as each subcatchment would be represented by its own set of parameters. Alternative calibration–validation approaches that enable model validation not only in time but also in space are conceptually preferable, particularly when the modelling is used for process understanding or prediction in ungauged locations (e.g., Wagener et al., 2004; Fenicia et al., 2016).

The objective of this study is to develop a semi-distributed hydrological model with the appropriate level of functional complexity to reproduce streamflow spatial variability in the Thur catchment. For this purpose, we use a two-stage approach, the first one dedicated to an in-depth analysis of the available data and the second one focused on hydrological modelling.

Our specific objectives are to (1) explore the spatial variability present in the Swiss Thur catchment regarding landscape characteristics, meteorological forcing, and streamflow signatures; (2) identify the main climate and landscape controls that explain the variability of the hydrological response; (3) based on this analysis, build a set of model experiments aimed to test the relative importance of dominant processes and their effect on the hydrograph; and (4) appraise model assumptions against competing alternatives using a stringent validation strategy.

The paper is organized as follows: section 3.2 presents the study area and gives information about data availability; section 3.3 illustrates the methodology; section 3.4 shows the results; section 3.5 analyses the results and puts them in perspective, showing what other similar studies have found; section 3.6, finally, summarizes the main conclusions.

3.2 STUDY AREA

This study is carried out in the Thur catchment (figure 3.1), located in the north-east of Switzerland, south-west of Lake Constance. With a total length of 127 km and a catchment area of 1702 km^2, the Thur is the longest Swiss river, without any natural or artificial reservoir along its course. The Thur River is very dynamic, with streamflow values that can change by 2 orders of magnitude within a few hours (Schirmer et al., 2014).

The Thur catchment has been the subject of several studies in the past: Gurtz et al. (1999) performed the first modelling study on the entire catchment using a semi-distributed hydrological model; Abbaspour et al. (2007) modelled hydrology and water quality using the SWAT model; Fundel et al. (2013) and Jorg-Hess et al. (2015) focused on low flows and droughts; Jasper et al. (2004) investigated the impact of climate change on the natural water budget. Other modelling studies also include Melsen et al. (2014, 2016), who investigated parameter estimation in data-limited scenarios and parameter transferability across spatial and temporal scales, and Brunner et al. (2019), who studied the spatial dependence of floods. The Thur also includes a small-sized experimental subcatchment (Rietholzbach, called Mosnang in this paper after the name of the gauging station) that was the subject of many field studies at the interface between process understanding and hydrological modelling (e.g., Menzel, 1996; Gurtz et al., 2003; Seneviratne et al., 2012; von Freyberg et al., 2014, 2015).

The topography of the catchment is presented in figure 3.1b; the elevation ranges between 356 m a.s.l. at the outlet and 2502 m a.s.l. at Mount Säntis. The majority of the catchment lies below 1000 m a.s.l. (75%) and only 0.6% is above 2000 m a.s.l. (Gurtz et al., 1999). Based on topography (figure 3.1b), the catchment can be visually subdivided into two distinct regions: the northern part, with low elevation and dominated by hills and flat land, and the southern part, which presents a mountainous landscape.

The land use (figure 3.1c) is dominated by pasture and sparsely vegetated soil (60%) and forest (25%); urbanized and cultivated areas are located mainly in the north and cover 7% and 4% of the catchment respectively.

Most of the catchment is underlain by conglomerates, marl incrustations, and sandstone (Gurtz et al., 1999). For the purpose of this study, the geological formations have been divided into three classes (figure 3.1d): "consolidated", covering mainly the mountainous part of the catchment, "unconsolidated", located in the north, and "alluvial", located in the proximity of the river network, mainly in the plateau area; the latter formation constitutes the main source of groundwater in the region (Schirmer et al., 2014). The soil depth (figure 3.1e) is shallower in the mountainous part of the catchment and deeper in the northern part.

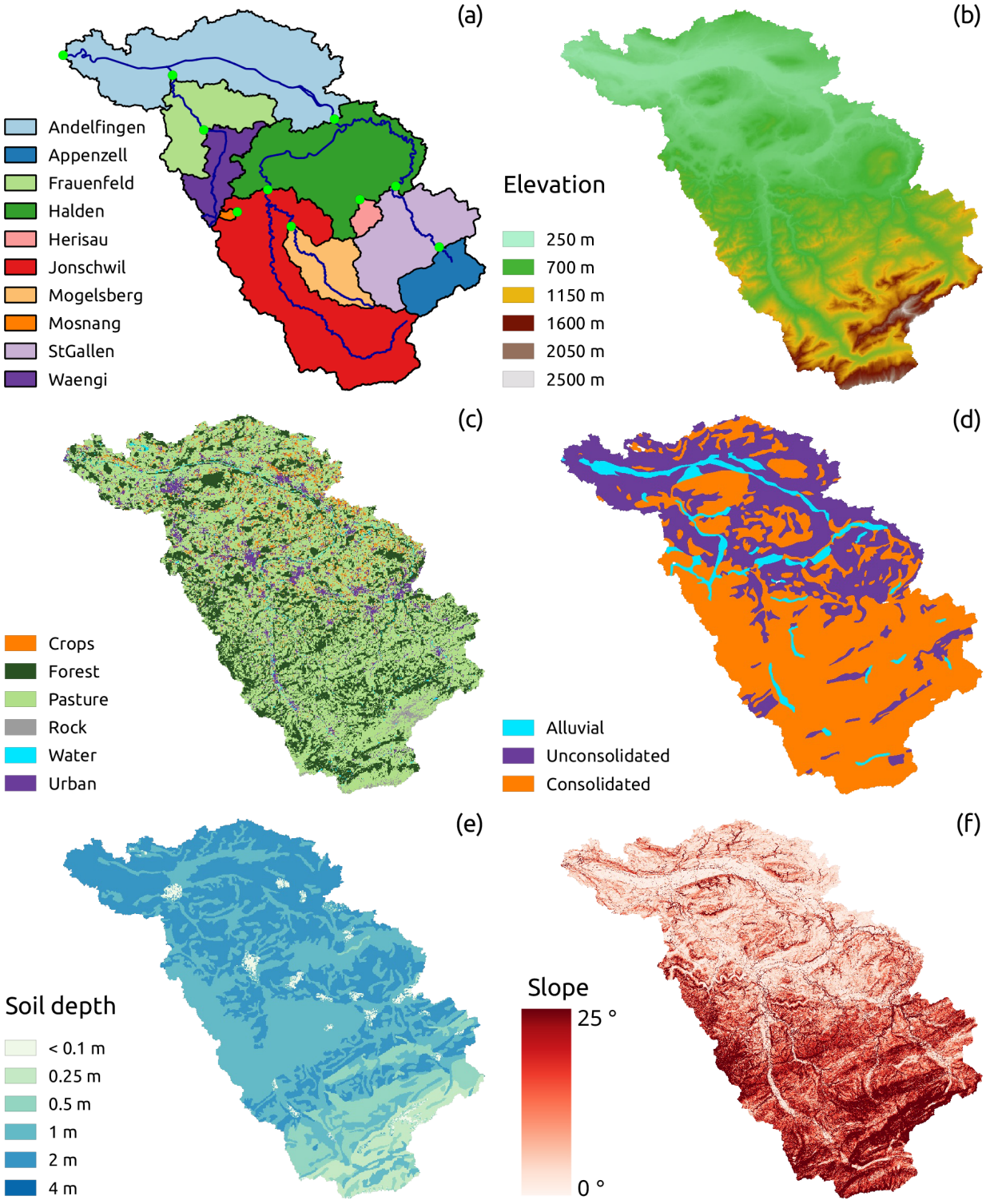

Figure 3.1: Landscape characteristics of the Thur catchment: **(a)** subdivision into subcatchments, river network, and gauging stations; **(b)** elevation map; **(c)** land-use map; **(d)** simplified geology map; **(e)** soil depth map; **(f)** slope map (derived from the elevation map).

Table 3.1: Identification of the gauging stations and description of the river network.

Catchment	Index	Code*	Upstream catchments
Andelfingen	1	2044	2 – 10
Appenzell	2	2112	–
Frauenfeld	3	2386	10
Halden	4	2181	2, 3, 5 – 10
Herisau	5	2305	–
Jonschwil	6	2303	7, 8
Mogelsberg	7	2374	–
Mosnang	8	2414	–
St. Gallen	9	2468	2
Wängi	10	2126	–

*Code of the gauging station, as defined by the Federal Office for the Environment, FOEN.

Based on the availability of gauging stations (table 3.1), the catchment was divided into 10 subcatchments (figure 3.1a), with a total drained area that ranges between 3.2 km^2 (Mosnang) and 1702 km^2 (Andelfingen). Streamflow time series are obtained from the Federal Office for the Environment FOEN, and the data are available from 1974 to 2017 but are used only from 1981 to 2005 to match the precipitation, temperature, and potential evapotranspiration (PET) time series. In the considered range, the streamflow data are relatively continuous, with two gaps, one in St. Gallen, from 31 December 1981 to 1 January 1983, and the other one in Herisau, from 31 December 1982 to 9 May 1983.

The raw maps (topography, land use, geology, and soil) are obtained from the Federal Office of Topography (swisstopo). The meteorological data are obtained from the Federal Office of Meteorology and Climatology MeteoSwiss. Precipitation and temperature are interpolated, as done in Melsen et al. (2016), with the WINMET pre-processing tool (Viviroli et al., 2009) using inverse distance weight (IDW) and detrended IDW respectively; while the first method considers only the horizontal variability (related to the distance from the meteorological stations), the second adds a vertical component to the variability related to the elevation (Garen and Marks, 2001). PET data are then obtained, as done in Gurtz et al. (1999), starting from meteorological and land-use data, using the Penman–Monteith equation (Monteith, 1975) implemented as part of the PREVAH hydrological model (Viviroli et al., 2009). All these values are calculated at pixel (100 m) scale and then averaged over the subcatchments. All the time series are used at daily resolution in the subsequent analyses, aggregating the available hourly data. This choice was influenced on the one hand by the need to limit the computational demand for the model experiments, for which a coarser temporal resolution is preferable, and on the other hand by the need to represent relevant hydrograph dynamics, for which finer temporal resolution is desirable (e.g., Kavetski et al., 2011). A daily data resolution, although it may obscure subdaily process dynamics, appeared to be a good compromise, and it is a typical choice in distributed model applications at such spatial scales (e.g., Kirchner et al., 2004).

3.3 METHODS

The methodology follows a two stages approach. The first stage aims at determining the climatic and landscape controls on streamflow signatures. The second stage uses this understanding to configure the structure of a semi-distributed model, whose functional suitability is tested through a set of model experiments. Section 3.3.1 describes the first stage of the analysis, that is, the identification of influence factors on the spatial variability of streamflow signatures. Section 3.3.2 describes the general structure of the semi-distributed model, and the model evaluation approach. The design of the model experiments, which is dependent on the outcomes of the first stage of analyses, is presented directly in the results section 3.4.2.2.

3.3.1 *Identification of influence factors on the spatial variability of streamflow signatures*

The purpose of the analysis presented in this section is to understand the influence of climatic conditions and landscape characteristics on streamflow. Climatic conditions are represented by precipitation, potential evaporation, and temperature time series. Landscape characteristics are represented by maps of topography, land use, geology, and soil.

Table 3.2: List of streamflow signatures, climatic indices, and subcatchment characteristics considered in the study.

Symbol	Name
	Streamflow signatures
ζ_Q (mm d^{-1})	Average daily streamflow
ζ_{RR} (-)	Runoff ratio
ζ_{EL} (-)	Streamflow elasticity
ζ_{FDC} (-)	Slope of the flow duration curve
ζ_{BFI} (-)	Baseflow index
ζ_{HDF} (day of the year)	Mean half streamflow date
ζ_{Q_5} (mm d^{-1})	5th percentile of the streamflow
$\zeta_{Q_{95}}$ (mm d^{-1})	95th percentile of the streamflow
ζ_{HQF} (d yr^{-1})	Frequency of high-flow events
ζ_{HQD} (d)	Mean duration of high-flow events
ζ_{LQF} (d yr^{-1})	Frequency of low-flow events
ζ_{LQD} (d)	Mean duration of low-flow events
	Climatic indices
ψ_P (mm d^{-1})	Average daily precipitation
ψ_{PET} (mm d^{-1})	Average daily potential evapotranspiration

Continued on next page

Continued from previous page

Symbol	Name
ψ_{AI} (-)	Aridity index
ψ_{FS} (-)	Fraction of snow
ψ_{HPF} (d yr^{-1})	Frequency of high-precipitation events
ψ_{HPD} (d)	Mean duration of high-precipitation events
ψ_{HPS} (-)	Season with most high-precipitation events
ψ_{LPF} (d yr^{-1})	Frequency of low-precipitation events
ψ_{LPD} (d)	Mean duration of low-precipitation events
ψ_{LPS} (-)	Season with most low-precipitation events
	Subcatchments characteristics
ξ_{A} (km^2)	Subcatchment area
ξ_{TE} (m)	Average elevation
ξ_{TSm} (°)	Average slope
ξ_{TSs} (%)	Fraction of the subcatchment with steep areas
ξ_{TAs} (%)	Fraction of the subcatchment facing south
ξ_{TAn} (%)	Fraction of the subcatchment facing north
ξ_{TAew} (%)	Fraction of the subcatchment facing east or west
ξ_{SM} (m)	Average soil depth
ξ_{SD} (%)	Fraction of the subcatchment with deep soil
ξ_{LF} (%)	Fraction of the subcatchment with forest land use
ξ_{LC} (%)	Fraction of the subcatchment with crops land use
ξ_{LU} (%)	Fraction of the subcatchment with urbanized land use
ξ_{LP} (%)	Fraction of the subcatchment with pasture land use
ξ_{GA} (%)	Fraction of the subcatchment with alluvial geology
ξ_{GC} (%)	Fraction of the subcatchment with consolidated geology
ξ_{GU} (%)	Fraction of the subcatchment with unconsolidated geology

Climatic conditions, landscape characteristics and streamflow are represented through a set of statistics (listed in table 3.2). In the following, statistics calculated based on streamflow data will be called streamflow "signatures", as it is often done in catchment classification literature (e.g., Sivapalan, 2006). We will refer to climatic and landscape "indices" for statistics calculated based on climatic data and landscape characteristics. A broad list of signatures and indices is presented in section 3.3.1.1; section 3.3.1.2 presents the approach for reducing such list to a set of meaningful variables; section 3.3.1.3 illustrates the approach for determining the indices that mostly control streamflow signatures; section 3.3.1.4 describes how the signature analysis is used to set-up the model experiments.

3.3.1.1 *Catchment indices for representing streamflow, climate, and landscape*

Streamflow signatures (ζ) and climatic indices (ψ) were obtained using streamflow, precipitation, PET, and temperature time series. The values were calculated using 24 years of data, between 01 September 1981 and 31 August 2005; we considered the 01 September as the beginning of the hydrological year. The periods with gaps in the data (refer to section 3.2 for details) were discarded from the analysis of the specific subcatchment. Landscape indices were obtained using the maps described in section 3.2.

Addor et al. (2017) recently compiled a comprehensive list of streamflow signatures and climatic indices for characterizing catchment behaviour (see table 3 in Addor et al., 2017). Here, we adopted their selection: while being originally introduced for a study about large sample hydrology, we believe that the indices proposed are also able to capture several different aspects of the time series and are therefore suitable also for this regional study. The streamflow signatures here considered are described hereafter, followed by an explanation of their rationale:

- average daily streamflow $\zeta_Q = \bar{\mathbf{q}}$, where $\mathbf{q}$ is the streamflow time series and the overbar represents the average over the observation period;
- runoff ratio $\zeta_{RR} = \frac{\bar{\mathbf{q}}}{\bar{\mathbf{p}}}$, where $\mathbf{p}$ is the precipitation time series;
- streamflow elasticity (ζ_{EL}) defined as

$$\zeta_{EL} = \text{med}\left(\left(\frac{\Delta\bar{\mathbf{q}}}{\bar{\mathbf{q}}}\right)/\left(\frac{\Delta\bar{\mathbf{p}}}{\bar{\mathbf{p}}}\right)\right) \tag{3.1}$$

  where $\Delta\bar{\mathbf{q}}$ and $\Delta\bar{\mathbf{p}}$ represent the streamflow and precipitation difference between two consecutive years and med is the median function;

- slope of the flow duration curve (ζ_{FDC}) defined as the slope between the log-transformed 33rd and 66th streamflow percentiles;
- baseflow index $\zeta_{BFI} = \frac{\bar{\mathbf{q}}^{(b)}}{\bar{\mathbf{q}}}$, where $\bar{\mathbf{q}}^{(b)}$ represents the baseflow and was calculated using a low-pass filter as illustrated in Ladson et al. (2013) with the equations

$$\mathbf{q}_t^{(f)} = \left(0,\ \theta_b \mathbf{q}_{t-1}^{(f)} + \frac{1+\theta_b}{2}\left(\mathbf{q}_t^{(f)} - \mathbf{q}_{t-1}^{(f)}\right)\right) \tag{3.2}$$

$$\mathbf{q}_t^{(b)} = \mathbf{q}_t - \mathbf{q}_t^{(f)} \tag{3.3}$$

  with $\bar{\mathbf{q}}^{(f)}$ representing the quickflow. The settings of the filter were taken according to the findings of Ladson et al. (2013) and, in particular, three filter passes were applied (forward, backward, and forward), the parameter θ_b was chosen to be equal to 0.925, and a reflection of 30 time steps at the beginning and at the end of the time series was used;

- mean half streamflow date (ζ_{HFD}) (Court, 1962), defined as the number of days needed in order to have a cumulated streamflow that reaches 50% of the total annual streamflow;

- 5th and 95th percentiles of the streamflow (ζ_{Q_5} and $\zeta_{Q_{95}}$ respectively);
- frequency (ζ_{HQF}) and mean duration (ζ_{HQD}) of high-flow events: they are defined as the days when the streamflow is bigger than 9 times the median daily streamflow;
- frequency (ζ_{LQF}) and mean duration (ζ_{LQD}) of low-flow events: they are defined as the days when the streamflow is smaller than 0.2 times the mean daily streamflow.

The frequency of days with zero streamflow, present in Addor et al. (2017), was not considered in this study because there are no ephemeral subcatchments in the study area.

This group of streamflow signatures is capable of capturing various characteristics of the hydrograph: ζ_Q measures the overall water flows, ζ_{RR} represents the proportion of precipitation that becomes streamflow, ζ_{EL} measures the sensitivity of the streamflow to precipitation variations, with a value greater than 1 indicating an elastic subcatchment (i.e. sensitive to change in precipitation) (Sawicz et al., 2011), ζ_{FDC} measures the variability of the hydrograph, with a steeper flow duration curve indicating a more variable streamflow, ζ_{BFI} measures the magnitude of the baseflow component of the hydrograph and can be considered a proxy for the relative amount of groundwater flow in the hydrograph, ζ_{HFD} measures the streamflow seasonality, ζ_{Q_5}, ζ_{LQF}, and ζ_{LQD} measure low-flow dynamics, and $\zeta_{Q_{95}}$, ζ_{HQF}, and ζ_{HQD} measure high-flow dynamics.

Climatology was represented through the following indices (see Addor et al., 2017, table 2):

- average daily precipitation $\psi_P = \bar{\mathbf{p}}$;
- average daily PET $\psi_{PET} = \bar{\mathbf{e}}_{pot}$, where $\mathbf{e}_{pot}$ is the potential evapotranspiration time series;
- aridity index $\psi_{AI} = \frac{\bar{\mathbf{e}}_{pot}}{\bar{\mathbf{p}}}$;
- fraction of snow (ψ_{FS}), defined as the volumetric fraction of precipitation falling as snow (i.e. on days colder than 0°C);
- frequency (ψ_{HPF}) and mean duration (ψ_{HPD}) of high-precipitation events: they are defined as days when the precipitation is more than 5 times the mean daily precipitation;
- season (ψ_{HPS}) with most high-precipitation events (defined as above);
- frequency (ψ_{LPF}) and mean duration (ψ_{LPD}) of dry days: they are defined as days when the precipitation is lower than 1 mm d^{-1};
- season (ψ_{LPS}) with most dry days (defined as above).

The seasonality of precipitation used in Addor et al. (2017) was not considered in this study as it relied on fitting a sinusoidal function to the precipitation values, which in our case did not produce reliable results.

Nevertheless, these climatological indices are able to comprehensively represent the climatic conditions of the subcatchment, with ψ_P representing average water input, ψ_{PET} representing

average evaporative demand, ψ_{AI} measuring the dryness of the climate, ψ_{FS} measuring the relative importance of snow, ψ_{HPF}, ψ_{HPD}, and ψ_{HPS} measuring the importance of intense precipitation events, and ψ_{LPF}, ψ_{LPD}, and ψ_{LPS} measuring the importance of dry days.

The landscape characteristics were divided into four categories: topography, land use, soil, and geology. In order to quantify the characteristics of each category, a set of indices (ξ) was defined. It is important to notice that all the areas calculated in this analysis were normalized by the respective subcatchment area (ξ_A) in order to get comparable values between subcatchments of different sizes.

Topography was represented with the following indices, calculated based on the digital elevation model:

- average elevation (ξ_{TE});
- average slope (ξ_{TSm});
- fraction of the subcatchment with steep areas (ξ_{TSs}), with slope larger than 10°;
- aspect, i.e. fraction of the subcatchment facing north (ξ_{TAn}), south (ξ_{TAs}), or east and west (ξ_{TAew}).

Land use was represented with the following characteristics, obtained by reclassifying the land-use map into four categories (from 22 original classes):

- fraction of the subcatchment with crop land use (ξ_{LC});
- fraction of the subcatchment with pasture land use (ξ_{LP});
- fraction of the subcatchment with forest land use (ξ_{LF});
- fraction of the subcatchment with urbanized land use (ξ_{LU}).

Soil type was represented with the following indices, derived by the soil map:

- fraction of the subcatchment with deep soil (soil depth greater than 2 m) (ξ_{SD});
- average soil depth (ξ_{SM}).

Geology was represented by the following indices, obtained by reclassifying the original map into three categories (from 22 original classes):

- fraction of the subcatchment with alluvial geology (ξ_{GA});
- fraction of the subcatchment with consolidated geology (ξ_{GC});
- fraction of the subcatchment with unconsolidated geology (ξ_{GU}).

The reclassification of the land use and of the geology maps consisted in aggregating specific classes into general classes (e.g. combining different types of forests into a unique forest class) with the objective of reducing their number, in order to facilitate subsequent analyses.

The consideration of topography, land use, soil, and geology for defining landscape indices was based on their potential influence on hydrological processes and, in turn, on the shape of the hydrograph. These landscape characteristics can all play an important role in controlling hydrological processes: land use can, for example, influence the infiltration of water in the substrate; soil thickness can affect the partitioning between water storage and runoff; vegetation is typically assumed to affect evaporation, and geology can affect groundwater dynamics. Indeed, these characteristics are used by many semi-distributed hydrological models, for example for determining parameter values or for dividing the catchment into areas with a homogenous hydrological response (e.g., Gurtz et al., 1999).

3.3.1.2 *Selection of meaningful streamflow signatures, climatic indices, and catchment indices*

The sets of statistics presented in section 3.3.1.1 were designed to be comprehensive. However, they may also be redundant, for example by containing metrics that express similar characteristics of the underlying data. In order to facilitate subsequent correlation analyses between the various sets of statistics, it is important to reduce each set to a short list of meaningful variables. The reduction of each set of streamflow signatures, climatic indices, and landscape indices was achieved through the following steps.

- All the statistics that did not show sufficient variability between the subcatchments were eliminated. We were in fact interested in identifying causes of spatial variability in the streamflow dynamics of the subcatchments of the Thur. Therefore, statistics that had a low variability were not of interest in this analysis. The variability was assessed using the coefficient of variation (defined by the ratio between the standard deviation and the average) and statistics with a value lower than 5% were discarded.

- All the catchment indices (e.g. a certain type of land use) that account for a limited part of any subcatchment were discarded. This point was motivated by the expectation that landscape characteristics covering a very small fraction of the subcatchment should not have a strong influence on the streamflow signatures considered. Here, landscape indices accounting for less than 5% of any subcatchment area were discarded.

- Within each set of streamflow signatures, climatic indices, and catchment indices, we retained only relatively independent metrics, if these are believed to represent the same underlying features of the time series. This step was motivated by the need to remove redundant information within each set. The selection of independent metrics was aided by Spearman's rank score between each pair of metrics, which represents (also non-linear) correlation between variables. Pairs of metrics with high absolute values of Spearman's rank score

are potentially redundant. In eliminating potentially redundant variables, we adopted the following criteria.

  - Among highly correlated metrics, we preferred those depending on single variables (e.g. only precipitation or only streamflow) to those containing multiple variables (e.g. combining precipitation and streamflow or evaporation, such as the runoff ratio or the aridity index), as this may be a problem when looking for correlations between metrics.
  - With respect to landscape indices, in many cases the high correlation is due to the fact that they are complementary (the areal fractions sum up to unity). In such cases, we kept one index per class (e.g. a single index for geology).

- A high correlation between metrics does not always mean that the metrics represent the same information. Therefore, the final selection of relevant metrics within each set was guided by expert judgement.

Based on this process, we compiled a reduced list of signatures, climatic indices, and landscape indices, which was used in subsequent analyses.

3.3.1.3 *Identification of climate and landscape controls on streamflow and consequences for model development*

This analysis aimed to identify climatic and landscape indices that mostly control streamflow signatures. In order to identify causality links between indices (ψ and ξ) and signatures (ζ), we proceed as follows:

- we calculated the correlation between indices and signatures using Spearman's rank score and identified pairs of variables with high correlation;
- we scrutinized pairs of variables with high correlations using expert judgement to decide whether a causality link between variables is justified.

The outcome of this process will be used to inform the semi-distributed model setup. The expert judgement is a critical step in the elicitation of causality from correlation (e.g., Antonetti and Zappa, 2018), and it is clearly subjective, being dependent on personal experience and subject matter knowledge. Although personal and subjective, expert decisions are based on an attempt to interpret data rather than being a priori defined, which is typically the case in the application of semi-distributed hydrological models.

3.3.1.4 *Semi-distributed model setup and model experiments*

We assumed a generic structure for a semi-distributed hydrological model, described in section 3.3.2.1, where some model structure characteristics are defined a priori and others are to be defined. In order to motivate the open decisions, we proceeded as follows:

- we used the identified causality links to interpret the dominant processes influencing signature spatial variability;
- we designed model experiments aimed to confirm the hypothesized climatic and landscape controls on streamflow spatial variability.

The overall objective of the model experiments is to prove that only models that incorporate the correct dependencies are able to correctly predict regional streamflow variability. In order to test this assumption, the model experiments will include cases where the assumed dependencies are not incorporated. Omitting an assumed dependency leads to a structurally simpler model, which may raise doubt that potential differences in model performance might be due to differences in model complexity. For this reason, the model experiments will include cases where alternative dependencies are incorporated, which do not reduce model complexity. In order to keep the study and presentation tractable, the model experiments will be limited to a few cases, illustrated in section 3.4.2.2, which we judge relevant for this specific application.

3.3.2 *General structure of the semi-distributed hydrological model and model evaluation approach*

This section describes the approach for building and testing a semi-distributed hydrological model designed to represent the observed streamflow and particularly the observed spatial variability of streamflow signatures. The general model structure is explained in section 3.3.2.1, the error model and the calibration procedure are described in Sections 3.3.2.2 and 3.3.2.3, and the metrics utilized to assess the performance are shown in section 3.3.2.4.

3.3.2.1 *General structure of the hydrological model*

We describe here the general model structure; the definition of specific model experiments, which depends on the results of the signature analysis done in the first step, will be described in section 3.4.2.2.

The model uses a two-layer decomposition of the catchment.

1. Subcatchments are defined by the presence of the gauging stations; this subdivision is due to the necessity of having locations in the model where the streamflow is both observed and simulated and, therefore, it is possible to calibrate and evaluate the parameters of the hydrological model.
2. HRUs are defined based on catchment characteristics (e.g. topography, geology, or vegetation); they represent parts of the catchment that are supposed to have a similar hydrological response to the meteorological forcing. Each HRU is characterized by its own parameterization. Different definitions of HRUs are tested, as described in section 3.4.2.2.

Each HRU has a unique parameterization. However, depending on how the inputs are discretized, the same HRU can have different states in different parts of the catchment. Therefore,

the same HRU needs its own model representation whenever the spatial variability of states needs to be considered. For example, if the inputs are discretized per subcatchment, the same HRU needs a separate model representation in each subcatchment where it is present. For more details about our model implementation of HRUs, refer to figure 4 of Fenicia et al. (2016).

In order to limit the levels of decisions of the semi-distributed models, some of the aspects of the distributed models are fixed a priori, and others are left open. In particular,

- the structure chosen to represent the various HRUs is kept fixed. That is, differences between HRUs will be reflected only through the parameter values.

- The definition of HRUs is left open. In particular, we do not a priori specify which approach is used to discretize the landscape.

- The spatial discretization of the model inputs is left open. Hence, we do not decide in advance which spatial discretization of the inputs is most appropriate.

Only the fixed decision about the HRU model structure is described here, whereas the open decisions are described in the results section 3.4.2.2. The spatial organization of the model structure is represented in figure 3.6, with the equations listed in appendix 3.A. The structure includes a snow reservoir (WR), with inputs distributed per subcatchments. Snowmelt and rainfall are input to an unsaturated reservoir (UR), which determines the portion of precipitation that produces runoff. This flux is split through a fast reservoir (FR), designed to represent the peaks of the hydrograph, proceeded by a lag function to offset the hydrograph, and a slow reservoir (SR), designed to represent baseflow. This structure was chosen to be parsimonious while general enough to reproduce typical hydrograph behaviour; it was tested in previous applications (e.g., van Esse et al., 2013; Fenicia et al., 2014, 2016), demonstrating its suitability for reproducing a wide range of catchment responses. It also resembles popular conceptual hydrological models such as HBV (Lindstrom et al., 1997) and HyMod (Boyle et al., 2001), which have been shown to have wide applicability. The model was built using the SUPERFLEX modelling framework (Fenicia et al., 2011).

3.3.2.2 *Error model*

As commonly done in hydrological modelling (e.g., McInerney et al., 2017), we here account for uncertainties by considering a probabilistic model of the observations $\mathbf{Q}(\theta, \mathbf{x})$, where θ is the vector of parameters and $\mathbf{x}$ the model input, which is composed of a deterministic hydrological model $\mathbf{h}(\theta_h, \mathbf{x})$ (illustrated in section 3.3.2.2) and a random residual error term $\mathbf{E}(\theta_E)$ that accounts for all data and model uncertainties (θ_h and θ_E represent the hydrological and error parameters):

$$z\left[\mathbf{Q}(\theta, \mathbf{x}), \lambda\right] = z\left[\mathbf{h}(\theta_h, \mathbf{x}), \lambda\right] + \mathbf{E}(\theta_E) \tag{3.4}$$

where $z[\mathbf{y}, \lambda]$ represents the Box–Cox transformation (Box and Cox, 1964) with parameter λ, which is used to account for heteroscedasticity (stabilize the variance). For $\lambda \neq 0$,

$$z[\mathbf{y}_t, \lambda] = \frac{\mathbf{y}_t^{\lambda} - 1}{\lambda} \tag{3.5}$$

The residual error term is assumed to follow a Gaussian distribution with zero mean and variance σ^2:

$$\mathbf{E}_t \sim \mathbf{N}\left(0, \sigma^2\right) \tag{3.6}$$

The error model has, therefore, two parameters (λ and σ^2); the first one was fixed to 0.5 (McInerney et al., 2017) and the second one was inferred.

This choice of error model (Gaussian noise applied to the Box–Cox transformation of the streamflow) allows for an explicit definition of the likelihood function (McInerney et al., 2017)

$$p\left(\mathbf{q}_{obs}|\theta_h, \theta_E, \mathbf{x}\right) = \prod_{t=1}^{T} z'\left(\mathbf{q}_{obs,t}\theta_E\right) f_{\mathbf{N}}\left(\mathbf{E}_t|0, \sigma^2\right) \tag{3.7}$$

where T represents the length of the time series, $f_{\mathbf{N}}$ is the Gaussian probability density function (PDF) and $z'\left(\mathbf{q}_{obs,t}\theta_E\right)$ is the derivative of $z\left(\mathbf{q}_{obs,t}\theta_E\right)$ with respect to $\mathbf{q}$ evaluated at the observed data $\mathbf{q}_{obs}$. Specifying equation 3.7 for the case where $z\left(\mathbf{q}_{obs,t}\theta_E\right)$ is defined by equation 3.5, the expression of the likelihood function becomes

$$p\left(\mathbf{q}_{obs}|\theta_h, \theta_E, \mathbf{x}\right) = \prod_{t=1}^{T} \mathbf{q}_{obs,t}^{(\lambda-1)} f_{\mathbf{N}}\left(\mathbf{E}_t|0, \sigma^2\right) \tag{3.8}$$

Equation 3.8 represents the likelihood function that is then used, together with a uniform prior distribution, to calibrate the parameters of the model as described in section 3.3.2.3.

3.3.2.3 *Calibration*

Parameter calibration is performed with the objective of maximizing their posterior density. According to the Bayes equation, the posterior distribution of model parameters is expressed as the product between the prior distribution and the likelihood function; since a uniform prior is used for the parameters, this is equivalent to maximizing the likelihood function in the defined parameter space; the optimization procedure is performed with a multi-start quasi-Newton method (Kavetski et al., 2007) with 20 independent searches. We empirically established that with models of our complexity (about 10 parameters), 20 independent searches provide good confidence that a global optimum will be found.

The evaluation of the model ability to reproduce streamflow is carried out in space–time validation (see also Fenicia et al., 2016). For this purpose, the time domain is divided into two periods of 12 years each (from 1 September 1981 to 1 September 1993 and from 1 September 1993 to 1 September 2005) and the subcatchments are split into two groups (A and B), according

to a spatial alternation (subcatchment in group A flows into a subcatchment in group B that flows into one in group A and so on); the subcatchments belonging to group A are Andelfingen, Herisau, Jonschwil, St. Gallen, and Wängi and the ones in group B are Appenzell, Frauenfeld, Halden, Mogelsberg, and Mosnang. This method implies a division of the space–time domain into four quadrants, such that the model can be calibrated in one quadrant and validated in the other three. For space–time validation, the model is calibrated using each group of subcatchment and period and validated using the other group of subcatchment and period. That is, the model calibrated using group A and period 1 was validated using group B and period 2, and so on for the other three combinations of subcatchments and groups. The model output in the four space–time validation periods is then combined to calculate model performance using various indicators (see section 3.3.2.4). Results are presented for space–time validation, which represents the most challenging test of model performance.

3.3.2.4 *Performance assessment*

Model performance is assessed using the following metrics.

1. Time series metrics, which evaluate the ability to reproduce streamflow time series. The metrics used for this assessment are the following.

   - Normalized log likelihood (F_{LL}), that is, the logarithm of equation 3.8 normalized by the number of time steps present in the time series. This metric corresponds to the objective function used for model optimization. It can be observed that, since λ is fixed at 0.5 in the Box–Cox transformation, model calibration is equivalent to maximizing the Nash–Sutcliffe efficiency (F_{NS}) calculated with the square root of the streamflow. F_{LL} is not bounded, but a higher value means a better match between two time series since, in this case, the absolute value of the residual is smaller and, thus, their PDF higher.

   - Nash–Sutcliffe efficiency:

$$F_{NS}(\mathbf{q}_{obs}, \mathbf{q}_{sim}) = 1 - \frac{\sum_{t=1}^{T} \left(\mathbf{q}_{sim,t} - \mathbf{q}_{obs,t}\right)^2}{\sum_{t=1}^{T} \left(\mathbf{q}_{obs,t} - \bar{\mathbf{q}}_{obs}\right)^2} \tag{3.9}$$

     which is often used in hydrological applications and provides a sense of the general quality of the simulations. F_{NS} is bounded between $-\infty$ and 1, with 1 meaning a perfect match.

2. Signature metrics, which determine the ability to reproduce the streamflow signatures (ζ) selected using the procedure illustrated in section 3.3.1.2. The agreement between simulated

and observed signatures is assessed using two metrics: Spearman's rank correlation (r) and the normalized root mean square error:

$$F_{RMSE} = \frac{\sqrt{\frac{\sum_{n=1}^{N} \left(\zeta_{sim,n} - \zeta_{obs,n}\right)^2}{N}}}{\frac{\sum_{n=1}^{N} \zeta_{obs,n}}{N}} \tag{3.10}$$

While r assesses how well the simulated signatures can be described using a monotonic function, F_{RMSE} imposes a more stringent requirement, as it assesses how well the simulated and observed signatures line up on the diagonal line.

The use of multiple metrics for assessing model performance enables a comprehensive assessment of various characteristics of the simulations. Time series metrics are designed to appraise the general quality of the model fit. Signatures, instead, are designed to highlight selected characteristics of the data at the expense of others.

3.4 RESULTS AND INTERPRETATION

3.4.1 *Influence factors on the spatial variability of streamflow signatures*

This section illustrates the results of the correlation analysis complemented by expert judgement aimed to identify influence factors that control the spatial variability of streamflow signatures; section 3.4.1.1 presents the results of the selection of meaningful statistics; section 3.4.1.2 identifies climate and landscape indices controlling streamflow signatures and presents consequences for model development.

3.4.1.1 *Selection of meaningful streamflow signatures, climatic indices, and catchment indices*

The streamflow signatures defined in section 3.3.1.1 were calculated for each subcatchment and the values are shown in table 3.3 together with the coefficient of variation. All the signatures have a coefficient of variability bigger than the threshold value of 5%, with the most variable signature being ζ_{LQF} (71%) and the least variable ζ_{HQD} (6%). Therefore, none of these signatures was discarded.

Figure 3.2 shows the correlations between the streamflow signatures: the lower triangle contains Spearman's rank correlation and the upper triangle the p-value associated with the correlations. Based on correlations and on its interpretation, a subset of ζ can be defined as follows.

- ζ_Q, ζ_{RR} and ζ_{EL} are strongly correlated ($r > 0.72$). We retained ζ_Q and discarded ζ_{RR} and ζ_{EL} because both contain climatic information (precipitation) in their definition.

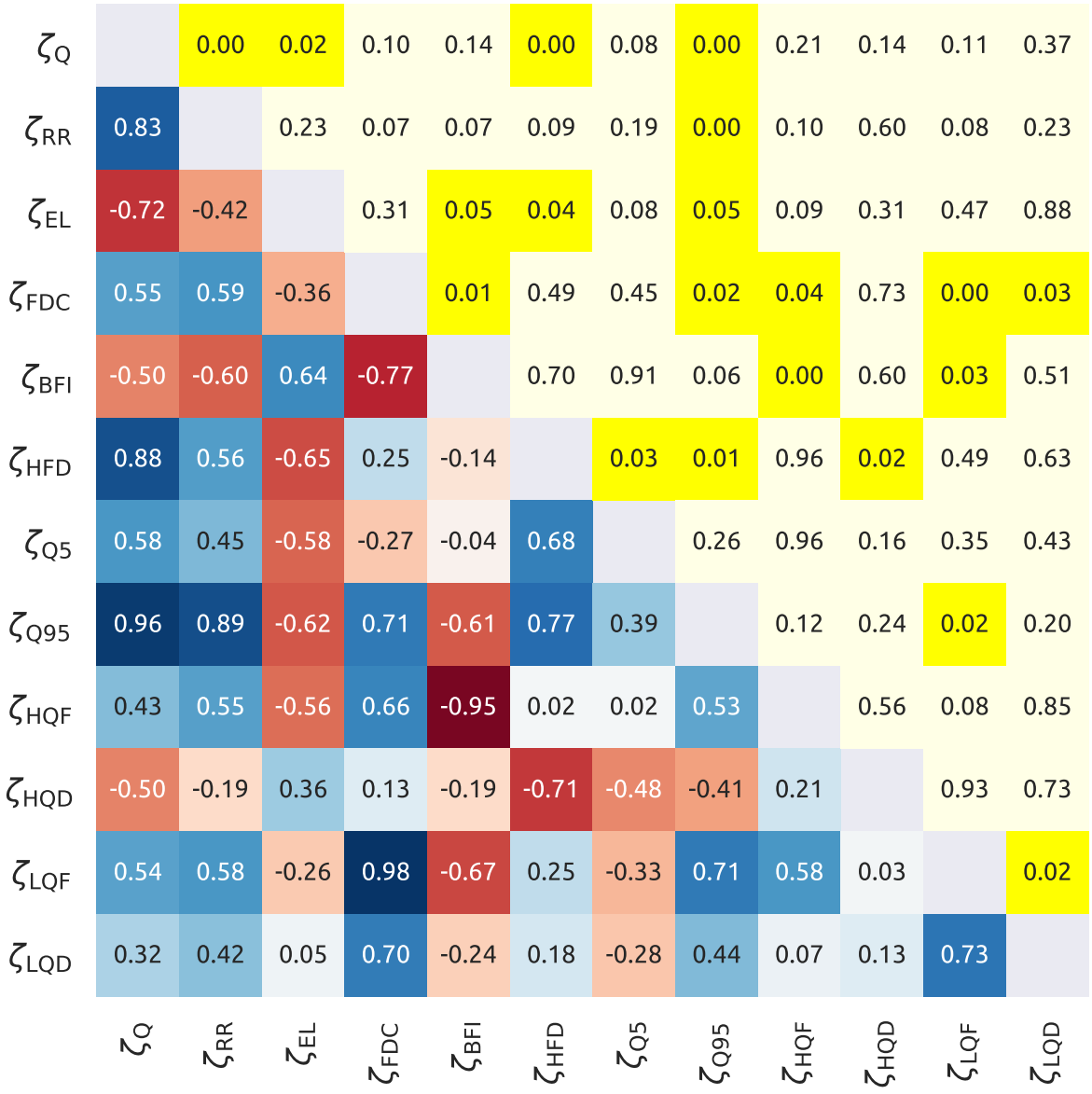

Figure 3.2: Internal correlation between the streamflow signatures. The lower triangle shows Spearman's rank score with the red colour that indicates negative correlations and the blue that indicates positive correlations. The upper triangle reports the corresponding p-values, where yellow colour indicates a statistically significant correlation (p-value < 0.05). The symbols used in the figure are reported in table 3.2

Table 3.3: Values of the streamflow signatures. The names of the subcatchments are abbreviated using the first three letters, the symbols are reported in table 3.2. The last column contains the coefficient of variation of each signature.

	Subcatchment										
	And	App	Fra	Hal	Her	Jon	Mog	Mos	StG	Wän	CV
ζ_Q	2.46	4.14	1.64	3.08	2.95	3.71	3.21	2.91	3.43	2.03	0.25
ζ_{RR}	0.63	0.80	0.49	0.70	0.71	0.80	0.70	0.72	0.71	0.56	0.14
ζ_{EL}	1.35	1.22	1.68	1.24	1.17	1.35	0.97	1.37	0.99	1.54	0.17
ζ_{FDC}	2.12	2.41	2.11	2.30	2.08	2.49	2.76	2.78	2.47	2.02	0.12
ζ_{BFI}	0.55	0.50	0.56	0.52	0.50	0.50	0.45	0.42	0.48	0.57	0.10
ζ_{HDF}	194.21	220.63	170.38	202.00	193.87	205.38	196.96	168.33	209.36	173.17	0.09
ζ_{Q_5}	0.50	0.70	0.35	0.57	0.74	0.54	0.44	0.28	0.60	0.49	0.27
$\zeta_{Q_{95}}$	6.96	12.85	4.83	9.23	9.17	11.19	10.57	10.46	11.00	5.98	0.28
ζ_{HQF}	2.21	5.17	3.50	3.67	6.34	4.46	6.54	12.96	5.87	2.96	0.57
ζ_{HQD}	1.39	1.25	1.45	1.35	1.40	1.39	1.37	1.58	1.35	1.29	0.06
ζ_{LQF}	17.50	31.92	12.92	24.21	2.62	37.21	49.42	66.92	28.35	7.25	0.71
ζ_{LQD}	6.67	6.18	3.69	6.53	2.00	7.44	6.38	7.11	4.53	4.35	0.32

- ζ_{BFI} and ζ_{FDC} are strongly correlated (r=-0.77). We decided to retain ζ_{BFI} as it is of easier interpretation (it is a proxy for the importance of groundwater flow, which is a potentially important process for the subsequent model development).
- ζ_{HFD} was kept because it measures the seasonality of the streamflow. Note that ζ_{HFD} is strongly correlated with ζ_Q (r=0.88). However, they reflect different properties of the hydrograph. In particular, ζ_{HFD} can be a useful indicator of the effect of snow-related processes.
- ζ_{Q_5} and ζ_{HQD} were retained because they have low correlation ($r < 0.71$) with the other selected signatures and because the first represents low flows and the second high flows.
- $\zeta_{Q_{95}}$, ζ_{HQF}, ζ_{LQD}, and ζ_{LQF} were discarded because they all show correlations with the selected signatures.

In summary, the original set of streamflow signatures was reduced to a set of five meaningful signatures, which will be used in the subsequent analyses: average daily streamflow (ζ_Q), baseflow index (ζ_{BFI}), half streamflow period (ζ_{HFD}), 5th percentiles of the streamflow (ζ_{Q_5}), and duration of high-flow events (ζ_{HQD}).

Table 3.4: Values of the climatic indices. The names of the subcatchments are abbreviated using the first three letters, the symbols are reported in table 3.2. The last column contains the coefficient of variation of each index.

	Subcatchment										
	And	App	Fra	Hal	Her	Jon	Mog	Mos	StG	Wän	CV
ψ_P	3.91	5.15	3.36	4.38	4.13	4.64	4.57	4.04	4.80	3.62	0.13
ψ_{PET}	1.60	1.37	1.70	1.55	1.61	1.54	1.57	1.69	1.49	1.71	0.07
ψ_{AI}	0.41	0.27	0.50	0.35	0.39	0.33	0.34	0.42	0.31	0.47	0.19
ψ_{FS}	0.04	0.21	0.04	0.05	0.09	0.15	0.13	0.09	0.13	0.05	0.57
ψ_{HPF}	15.21	14.38	17.67	14.58	15.82	14.54	14.58	16.13	14.31	17.50	0.08
ψ_{HPD}	1.20	1.17	1.17	1.18	1.22	1.20	1.19	1.22	1.17	1.19	0.01
ψ_{HPS}	Summer	Summer	Summer	Summer	Summer	Summer	Summer	Summer	Summer	Summer	0.00
ψ_{LPF}	201.67	195.79	216.83	198.54	205.04	197.21	198.92	205.75	197.69	213.17	0.04
ψ_{LPD}	3.57	3.50	3.83	3.50	3.63	3.51	3.51	3.66	3.51	3.76	0.03
ψ_{LPS}	Fall	Fall	Fall	Fall	Fall	Fall	Fall	Fall	Fall	Fall	0.00

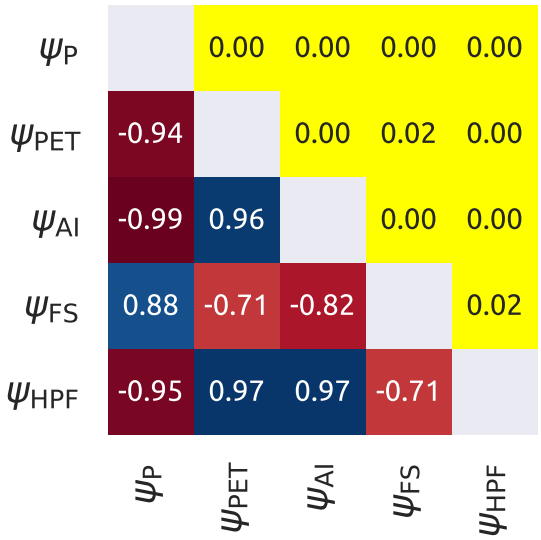

Figure 3.3: Internal correlation between the climatic indices. The lower triangle shows Spearman's rank score with the red colour that indicates negative correlations and the blue that indicates positive correlations. The upper triangle reports the corresponding p-values, where yellow colour indicates a statistically significant correlation (p-value < 0.05). The symbols used in the figure are reported in table 3.2

In terms of climatic indices, table 3.4 shows their values together with the coefficient of variation. It can be seen that there are some indices that show very little or no variation at all and, therefore, they could already be excluded from the subsequent correlation analysis; they are ψ_{HPD} (1%), ψ_{HPS} (0%), ψ_{LPF} (4%), ψ_{LPD} (3%), and ψ_{LPS} (0%).

Figure 3.3 shows the correlation between the remaining indices. It can be observed that they all have strong internal correlation ($r > 0.71$). For this reason it was decided to retain only ψ_P and ψ_{FS}, as they have lower correlation. The former represents an important term of the water budget, and the latter captures snow dynamics.

Table 3.5 shows the values of the catchment characteristics considered in this study. All of them have a coefficient of variation larger than the minimum threshold of 5%. Therefore, none of them was excluded based on this criterion. The second criterion for the pre-exclusion of the catchments characteristics, consisting in removing ξ occupying less than 5% of the subcatchments, led to the suppression of ξ_{LC} (which occupies 4% of the subcatchment).

Figure 3.4 shows the correlations between catchment characteristics; in many cases the high correlation is due to the fact that many indices are complementary (e.g. different types of geology). The following ξ were selected (one index per class):

- ξ_A because it is low correlated with the other features;
- ξ_{TE} and ξ_{TAs} in representation of the topography;
- ξ_{LF} for the land use;

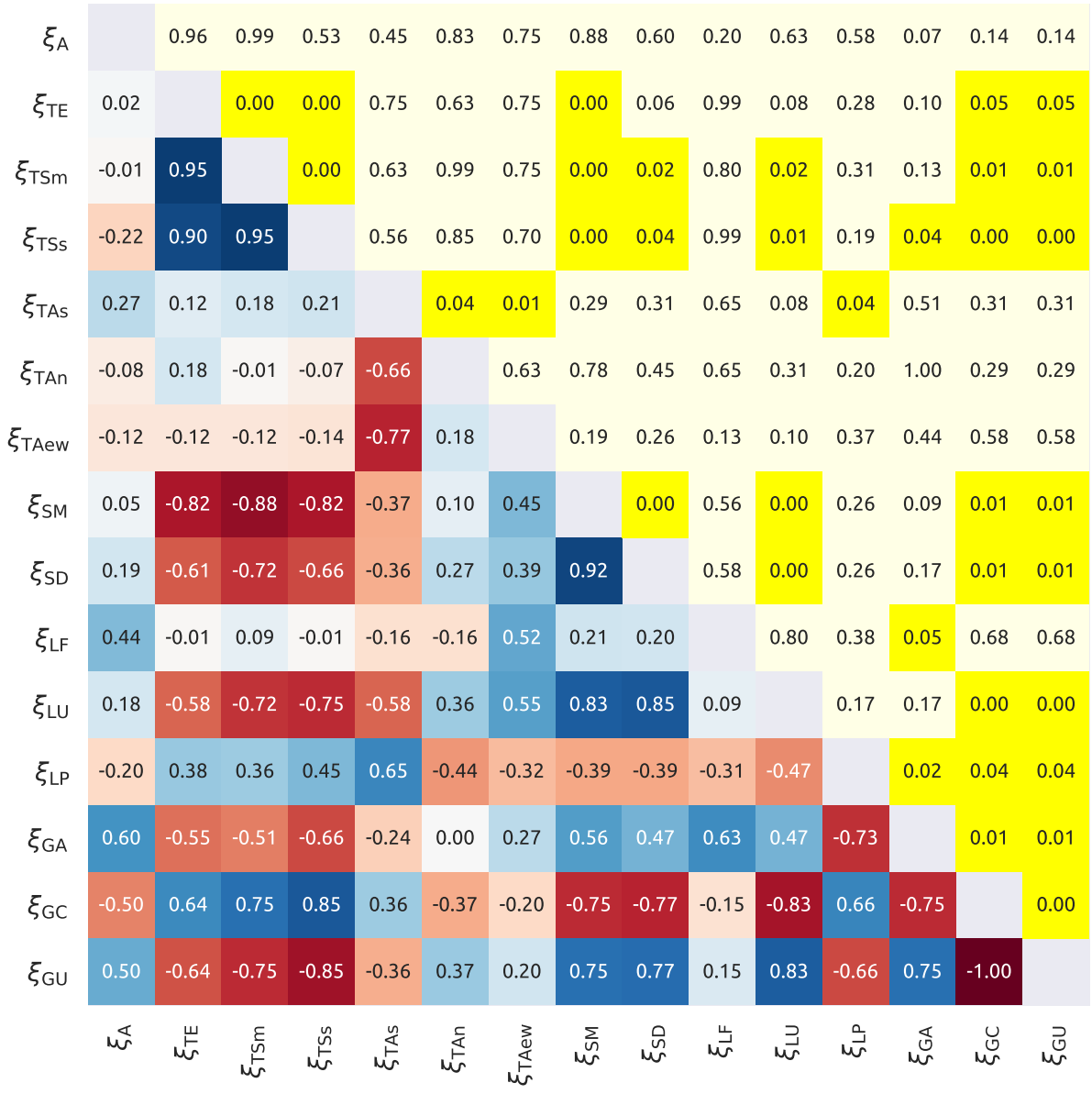

Figure 3.4: Internal correlation between the catchment characteristics. The lower triangle shows Spearman's rank score with the red colour that indicates negative correlations and the blue that indicates positive correlations. The upper triangle reports the corresponding p-values, where yellow colour indicates a statistically significant correlation (p-value < 0.05). The symbols used in the figure are reported in table 3.2

Table 3.5: Values of the subcatchment characteristics. The names of the subcatchments are abbreviated using the first three letters, the symbols are reported in table 3.2. The last column contains the coefficient of variation of each characteristic.

	Subcatchment										
	And	App	Fra	Hal	Her	Jon	Mog	Mos	StG	Wän	CV
ξ_A	1701	74.46	213.34	1085	16.72	493.0	88.11	3.19	261.1	78.96	1.40
ξ_{TE}	768	1250	591	908	831	1020	954	797	1039	650	0.22
ξ_{TSm}	13.32	25.23	9.70	16.87	15.44	20.66	19.77	15.68	19.72	12.49	0.27
ξ_{TSs}	0.47	0.81	0.33	0.62	0.69	0.77	0.79	0.71	0.73	0.45	0.26
ξ_{TAs}	0.25	0.22	0.23	0.23	0.21	0.23	0.24	0.40	0.24	0.21	0.23
ξ_{TAn}	0.32	0.35	0.33	0.32	0.33	0.32	0.31	0.24	0.33	0.32	0.09
ξ_{TAew}	0.43	0.43	0.44	0.44	0.46	0.44	0.45	0.36	0.43	0.47	0.07
ξ_{SM}	1.30	0.56	1.48	1.10	1.32	0.93	1.17	1.00	1.03	1.35	0.23
ξ_{SD}	0.40	0.04	0.49	0.25	0.41	0.13	0.28	0.00	0.26	0.36	0.63
ξ_{LF}	0.26	0.25	0.28	0.27	0.21	0.31	0.34	0.18	0.27	0.30	0.17
ξ_{LC}	0.04	0.00	0.04	0.03	0.03	0.01	0.01	0.01	0.01	0.04	0.79
ξ_{LU}	0.08	0.03	0.10	0.06	0.15	0.04	0.03	0.03	0.05	0.10	0.63
ξ_{LP}	0.60	0.59	0.57	0.61	0.61	0.61	0.62	0.77	0.63	0.55	0.09
ξ_{GA}	0.06	0.01	0.09	0.03	0.00	0.02	0.02	0.00	0.01	0.11	1.05
ξ_{GC}	0.59	0.92	0.54	0.73	0.88	0.90	0.92	1.00	0.88	0.63	0.20
ξ_{GU}	0.35	0.07	0.36	0.23	0.12	0.07	0.06	0.00	0.10	0.26	0.79

- ξ_{SD} representing the soil characteristics;
- ξ_{GC} for the geology.

In summary, the original set of catchment indices was reduced to a set of six indices.

3.4.1.2 *Selection of controlling factors on streamflow signatures*

Figure 3.5 reports the results of Spearman's correlation between climatic indices plus catchment characteristics and streamflow signatures. Panel (a) contains Spearman's rank coefficients and panel (b) shows p-values associated with them.

The following results can be noted.

- The three statistics average precipitation (ψ_P), fraction of snow (ψ_{FS}), and average elevation (ξ_{TE}) correlate strongly with average streamflow (ζ_Q) and seasonality (ζ_{HFD}) ($r > 0.64$ and p-value < 0.05). This correlation can be interpreted as follows: subcatchments with high elevation (ξ_{TE}) tend to have higher precipitation (ψ_P) due to orographic effects, which leads to higher streamflow (ζ_Q). They also tend to have more snow (ψ_{FS}) due to lower temperatures, which influences the seasonality (ζ_{HFD}).
- There are then some catchment characteristics that have no correlation ($r < 0.45$) with the streamflow signatures (catchment area (ξ_A) and land use (ξ_{LF})) or limited correlation (aspect (ξ_{TAs}) and deep soil (ξ_{SD}), with $r < 0.64$).

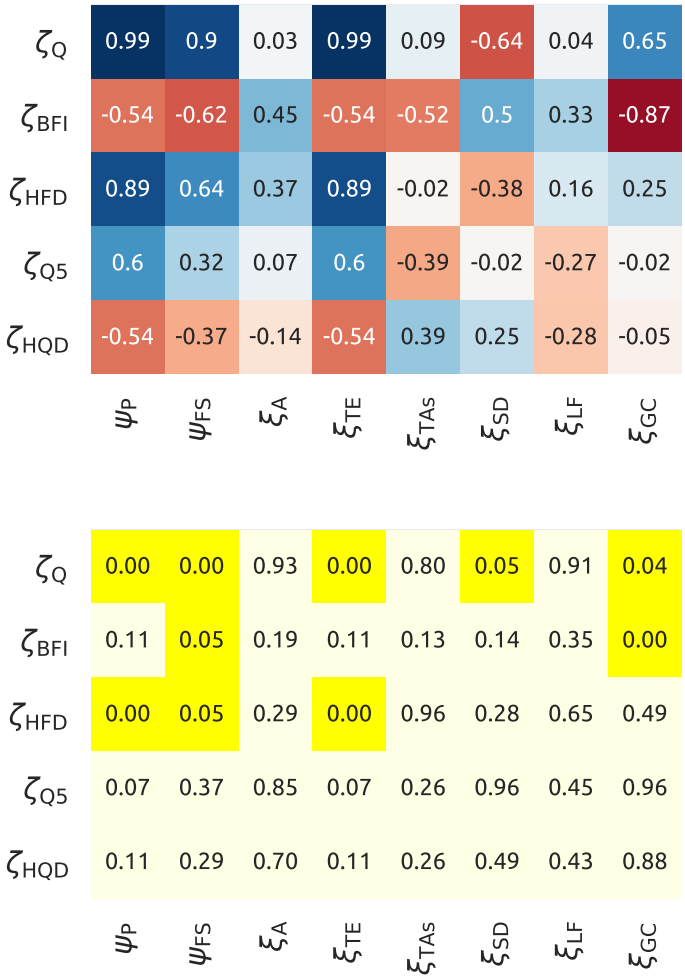

Figure 3.5: Correlation between the selected streamflow signatures (rows) and the selected climatic indices and catchment characteristics (columns). Panel **(a)** shows Spearman's rank score with the red colour that indicates negative correlations and the blue that indicates positive correlations. Panel **(b)** reports the corresponding p-values, where yellow colour indicates a statistically significant correlation (p-value < 0.05). The symbols used in the figure are reported in table 3.2

- The consolidated geology (ξ_{GC}) presents a strong correlation (r=-0.87) only with the baseflow index (ζ_{BFI}); that is not captured by the other indices.
- The streamflow signatures of low and high flows (ζ_{Q5} and ζ_{HQD}) cannot be explained by any index, with little correlation only with ψ_P and ξ_{TE} ($r < 0.60$) that is not sufficient to reach a p-value lower than 0.05.

These results are the premise for designing meaningful model experiments.

3.4.2 *Hypotheses for model building*

This section interprets the results found in section 3.4.1.2 and formulates some hypotheses regarding the hydrological functioning of the catchment (section 3.4.2.1). Section 3.4.2.2, then, presents the model alternatives designed for testing those hypotheses.

3.4.2.1 *Hypotheses on catchment functioning*

The results of the correlation analysis can be interpreted to formulate the following hypotheses regarding the drivers of streamflow variability.

1. The precipitation is the first driver of the differences in the water balance of the subcatchments. The effect of topographic variability manifests itself primarily as an influence on precipitation (amount and type). Accounting for variability of precipitation therefore implicitly reflects such effect of topography on the hydrograph, since some inputs were interpolated taking into account the effect of the elevation (section 3.2). Other phenomena potentially altering the water balance (e.g. regional groundwater flow) do not have a significant role and should not be considered.
2. Snow-related processes (e.g. amount of snow, timing of snowmelt) control differences in streamflow seasonality between subcatchments. Hence, the model needs to account for snow-related processes and their spatial variability.
3. Geology exerts an important control on the partitioning between quickflow and baseflow. Hence, the model should distinguish the different response behaviours of distinct geological areas.
4. The other catchment characteristics (e.g. soil, vegetation) show little or no correlation with the streamflow signatures, and therefore they should not be considered if the idea is to keep the model as simple as possible.

The streamflow signatures ζ_{Q5} and ζ_{HQD}, which have been selected as part of the analysis shown in section 3.4.1.1, do not manifest a strong correlation with any of the indices (r is always less than 0.60), meaning that the identification of their potential controls is not obvious with

the chosen approach. Hence, we have not been able to build model hypotheses that specifically target those signatures. As a result, we expect that the chosen models will not excel and will perform similarly in reproducing these signatures. The model comparisons used to test the four hypotheses listed above are described in section 3.4.2.2.

3.4.2.2 *Modelling experiments for testing the hypotheses*

Using the model structure described in section 3.3.2.1, four model configurations were compared by varying the number and the definition of the HRUs, and changing the structure of the HRUs (figure 3.6). The objective of the experiments was to test the hypotheses 1–4 in section 3.4.2.1 using semi-distributed hydrological models.

For all the models, the meteorological inputs (precipitation, PET, temperature) are aggregated at the subcatchment scale. Based on the first hypothesis, we assume that this discretization is sufficient to capture the regional difference in water balance between subcatchments. This hypothesis is tested with model M0, with uniform parameters in the catchment (i.e. a single HRU) and distributed precipitation input. This model does not consider snow processes. We expect that this model will be able to reproduce differences in streamflow averages between subcatchments.

The second hypothesis (snow controls seasonality) is tested with model M1. Relative to M0, M1 accounts for snow processes, represented by a simple degree-day snow module (see Kavetski and Kuczera, 2007), with inputs (temperature) distributed per subcatchment. We expect that this model will be able to reproduce differences in streamflow seasonality between subcatchments.

The third hypothesis (geology controls baseflow) is tested with model M2. Relative to M1, M2 considers two HRUs, defined based on geology type. One HRU contains the areas with consolidated geology, while the other HRU contains the rest of the catchment (unconsolidated and alluvial geology together). We expect that M2 will be able to reproduce differences in the baseflow index between subcatchments.

The fourth hypothesis (other catchment characteristics should not be considered if the idea is to keep the model as simple as possible) is exemplified by model M3. M3 is analogous to M2 in terms of complexity, but the HRUs are based on catchment characteristics that did not show correlation with the streamflow signatures. Among those characteristics, we have selected land use and considered an HRU based on forest and crops and the second one that occupies the rest of the catchment. This model is as complex as M2 (therefore it is more complex than M1); hence it has the same dimensions of flexibility to fit the data. However, since the structure of this model does not incorporate the cause–effect relationships derived from the signature analysis, we expect that its predictive performance will be poorer than M2.

The total number of the calibrated parameters depends on the number of HRUs and on the structure used to represent them: it was 8 for M0, 9 in M1, and 13 in M2 and M3, of which 5 parameters are common to all the HRUs (figure 3.6 and table 3.6); these parameters are C_e that governs the evapotranspiration, t_{rise}^{OL} and t_{rise}^{IL} that control the routing in the river network, that

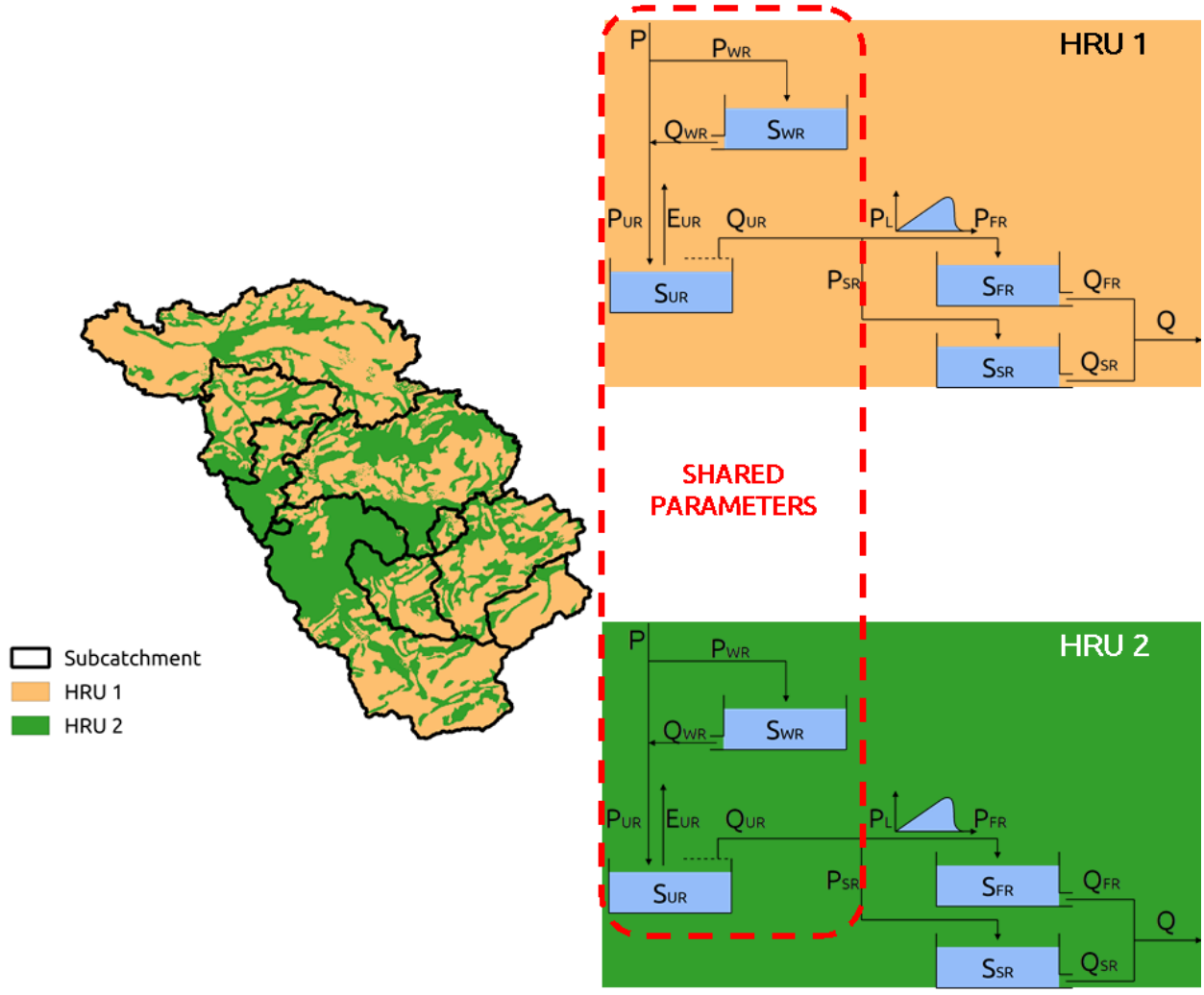

Figure 3.6: Spatial organization of the model structure: the catchment is divided into subcatchments (black lines), based on the location of the gauging stations, and HRUs (background colour), based on the catchment characteristics. All the HRUs have the same structure, but each HRU has its own parameterization except for some shared parameters. In the case of a single-HRU model (i.e. M0 and M1), the model maintains the subdivision into subcatchments but loses the subdivision into multiple HRUs.

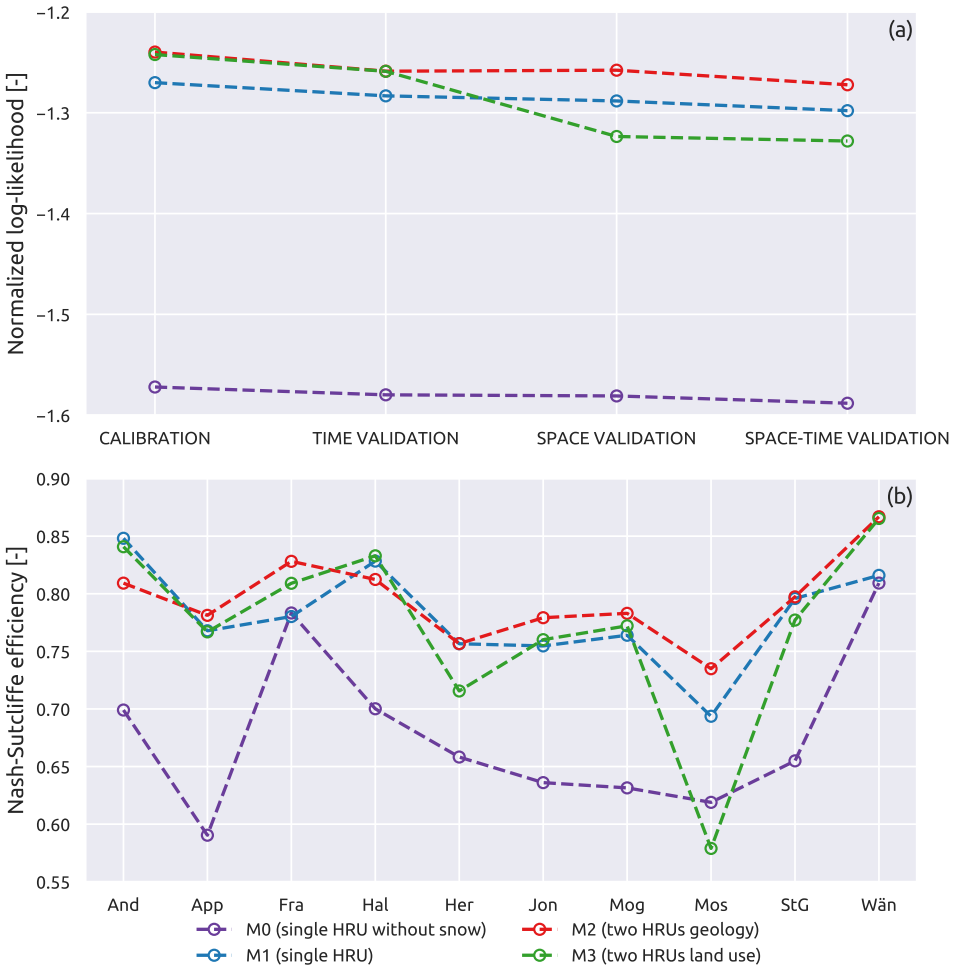

Figure 3.7: Normalized log likelihood **(a)** and Nash–Sutcliffe efficiency **(b)** for the three model configurations. **(a)** reports the variation between calibration and validation of the average of the 10 subcatchments; **(b)** shows the variation between subcatchments during space–time validation

regulates the outflow of the snow reservoir, and that determines the behaviour of the unsaturated reservoir.

3.4.3 *Modelling results*

The models presented in section 3.4.2.2 are evaluated in terms of hydrograph metrics (section 3.4.3.1) and signature metrics (section 3.4.3.2).

3.4.3.1 *Model performance in terms of hydrograph metrics*

Figure 3.7a shows the values of the likelihood function (corresponding to the calibration objective function) for the four models in calibration and validation. It can be observed that Mo is, by far, the worst model, with the lowest value of the likelihood function. Regarding the other three models, it can be seen that, during calibration, M1, which has the lowest number of calibration parameters, has the lowest performance, whereas M2 and M3 have higher and similar likelihood

values. This behaviour persists in time validation, with M2 and M3 that outperform M1. In space and space–time validation, however, M3 has the lowest likelihood value of the three, whereas M1 and M2 limit their decrease in performance, ranking respectively second and first in terms of optimal likelihood value.

The likelihood function represents an aggregate metric of model performance; in order to get a sense of appreciation of model fit on individual subcatchments, figure 3.7b reports the values of Nash–Sutcliffe efficiency in space–time validation for each of the subcatchments. On average, M2 has the best performance of all models (F_{NS}=0.79), followed by M1 (F_{NS}=0.78), M3 (F_{NS}=0.77), and M0 (F_{NS}=0.68). M3 and M0 have the highest variability of performance, with F_{NS} values between 0.58 and 0.86 and between 0.59 and 0.81. M1 and M2 have similar spread of F_{NS} values, ranging from 0.69 to 0.85 for M1 and from 0.73 to 0.87 for M2. Therefore, M1 and M2 have a more stable performance across subcatchments than M3. M3 obtains a significantly worse performance than the other three models on Mosnang, where it reaches a F_{NS} value of 0.58 (M0, M1, and M2 have values of 0.62, 0.69, and 0.73 respectively).

It can also be observed that M2 is generally better than M1, with F_{NS} values that are higher or approximately equal except for subcatchments Andelfingen and Halden, where the F_{NS} is slightly worse (however still higher than 0.80). M3 is clearly better than M1 in Andelfingen, Frauenfeld, and Wängi, and clearly worse in Herisau and Mosnang. In particular, in Mosnang (the smallest basin), M3 reaches the worst performance of all the models on all the subcatchments.

Regarding M0, it is interesting to observe that it has the worst performance (among all the subcatchments) in Appenzell, which is the subcatchment that is most affected by snow (ψ_{FS}=0.21), while it reaches a performance similar to M1 in Frauenfeld and Wängi, which are two subcatchments with almost no snow.

3.4.3.2 *Model performance in terms of signature metrics*

Figure 3.8 compares the ability of M0 and M1 to capture the signatures representing average streamflow (ζ_Q) and seasonality (ζ_{HFD}). The analysis is presented for space–time validation and, for ζ_{HFD}, focuses only on the four subcatchments that are most affected by the snow (ψ_{FS} >0.10), to emphasize the differences between the results of the two models. Each colour represents a different subcatchment and each dot a year; the red dashed line has a 45° slope and represents where the dots should align in case of perfect simulation results. The normalized root mean square error and Spearman's rank score are also reported. It is important to stress that the models have not been calibrated using any of the signatures as an objective function, which therefore represent independent evaluation metrics.

It can be observed that M0 represents ζ_Q equally well as M1, with almost no difference between the two models (r is 0.95 in both cases, whereas F_{RMSE} is 0.11 for M0 and 0.10 for M1). Focusing on the ability to capture ζ_{HFD}, it can be seen that the points corresponding to M0 all lie in the upper-left part of the plot, meaning that this model underestimates the signature values. With respect to M1, instead, the points are more aligned around the diagonal. This difference in

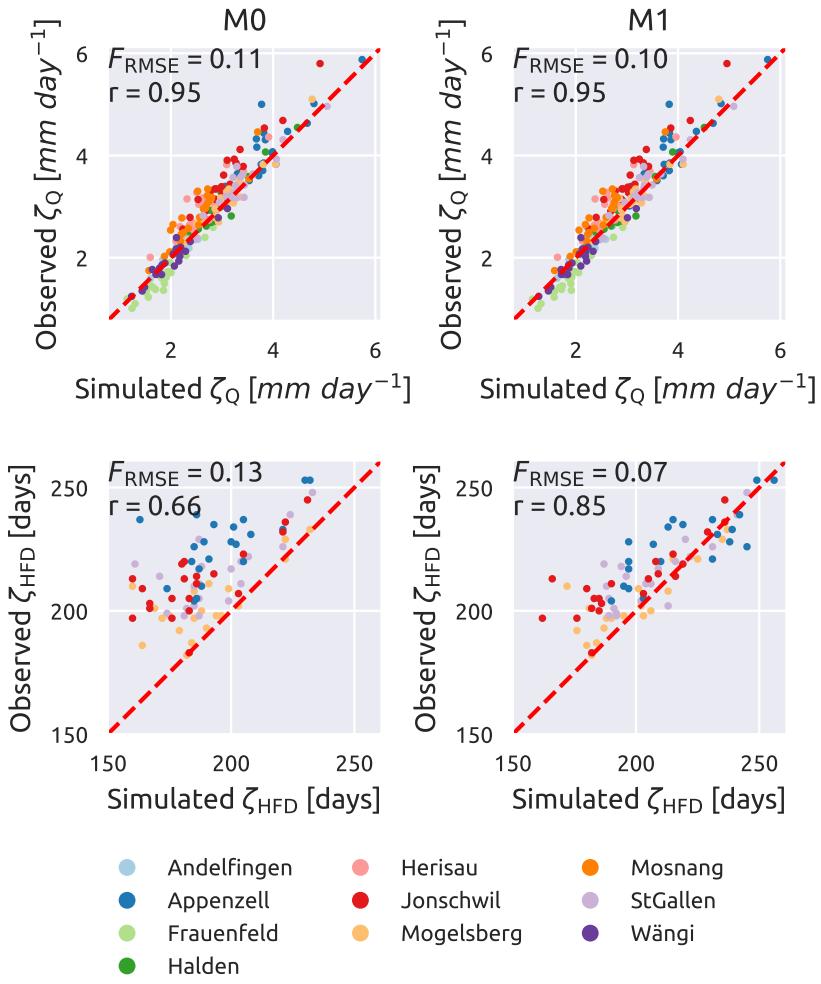

Figure 3.8: Influence of the model structure on the representation of the average streamflow (ζ_Q) and the mean half streamflow day (ζ_{HFD}). Single-HRU model without snow reservoir in **(a, c)** (M0) and single-HRU model with snow reservoir in **(b, d)** (M1). Each dot represents a year and each colour a subcatchment. For ζ_{HFD}, only the four subcatchments with the fraction of snow (ψ_{FS}) larger than 10% are plotted. The red dashed line has a 45° slope and indicates where all points should align in case of a perfect match. Spearman's rank score (r) is also reported.

performance is captured by the values of F_{RMSE} (0.13 for M0 and 0.07 for M1) and of r (0.66 for M0 and 0.85 for M1).

Figure 3.9 compares the observed and simulated signatures for the other three models (M1, M2, and M3). All of them are equally good in representing ζ_Q (F_{RMSE} is 0.10, 0.10, and 0.11, and r is 0.95, 0.96, and 0.95 for M1, M2, and M3 respectively) and ζ_{HFD} (F_{RMSE} is 0.07, 0.07, and 0.05 and r is 0.85, 0.84, and 0.87 for M1, M2, and M3 respectively). In all cases the cloud of points appears to be aligned to the diagonal, meaning that the three models are able to capture the values of the signatures each year. Moreover, there is no sensible difference in the various models in representing those signatures.

The performance of all the models decreases for ζ_{Q_5} where the models have a similar performance, with F_{RMSE} equal to 0.32, 0.28, and 0.33, and r equal to 0.62, 0.66, and 0.61 for M1, M2, and M3 respectively. The points are still aligned along the diagonal but are quite dispersed, especially if compared with ζ_Q and ζ_{HFD}, meaning that the models capture the general tendency but have deficiencies in capturing the inter-annual variability.

In terms of ζ_{BFI}, M2 performs clearly better than the other models. It is the only model that is able to represent this signature, with F_{RMSE}=0.07, r=0.83, and the points that align compactly with the diagonal. The other two models have a lower performance (F_{RMSE} equal to 0.11 and 0.10, and r equal to 0.31 and 0.52 for M1 and M3 respectively), with points that are quite dispersed and align almost vertically, implying that the simulated values have a range of variability that is definitely smaller than the observed data.

Figure 3.10 shows the comparison between observed and simulated ζ_{HQD}; since this signature requires a long time window to be computed, it is not calculated year by year (as done with the other signatures) but as an aggregated value over the 24 years. In terms of performance, M2 still remains the best among the three models, with F_{RMSE} of 0.09 and r of 0.69; in second place comes M1, which outperforms M2 in terms of r (0.77) but has a higher F_{RMSE} (0.19), meaning that M1 has the points that are more aligned but on a line that is farther from the diagonal compared to M2; M3, finally, has a bad performance, with high F_{RMSE} (0.18) and low r (0.48). All the models tend to slightly overestimate the duration of high-flow events with most of the points that lie on the right-hand side of the diagonal.

3.4.4 *Hypotheses testing*

The results of the hydrological model experiments appear to support our general hypothesis that only models that account for the influence factors that affect the streamflow signatures are able to reproduce streamflow spatial variability (see section 3.4.2.1). This provides confidence that those models are a realistic representation of dominant processes in the catchment. The implications of the modelling results with respect to the evaluation of the four hypotheses are explained as follows.

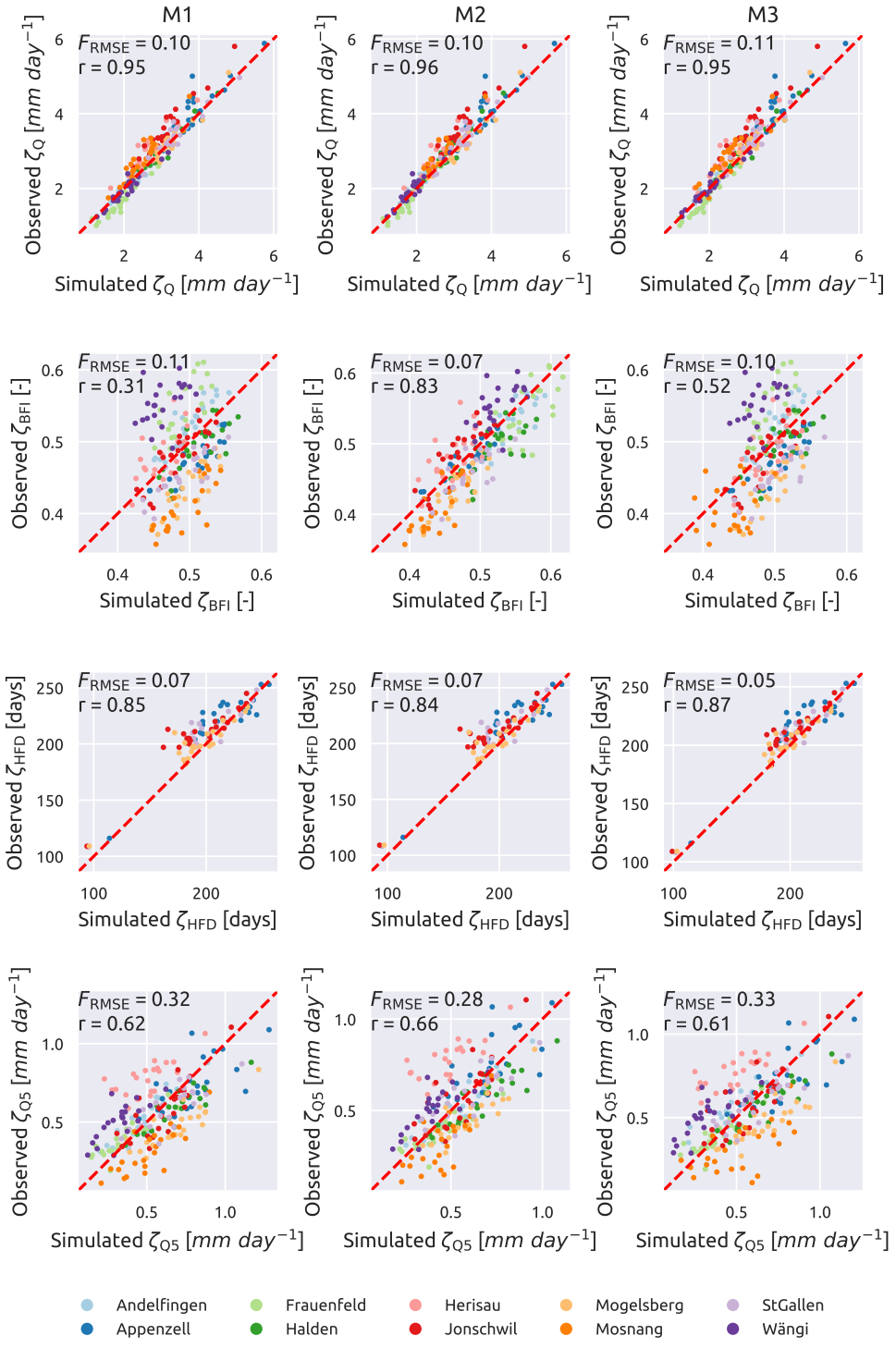

Figure 3.9: Simulated vs. observed streamflow signatures. Single-HRU model on the left (M1), two-HRU model based on geology in the centre (M2), and two-HRU model based on land use on the right (M3). Each dot represents a year and each colour a subcatchment. From up to bottom, mean daily streamflow (ζ_Q), baseflow index (ζ_{BFI}), mean half streamflow date (ζ_{HFD}, only the catchment with ψ_{FS} larger than 10%), and 5th percentile of the streamflow (ζ_{Q5}). The red dashed line has a 45° slope and indicates where all points should align in case of a perfect match. Spearman's rank score (r) is also reported

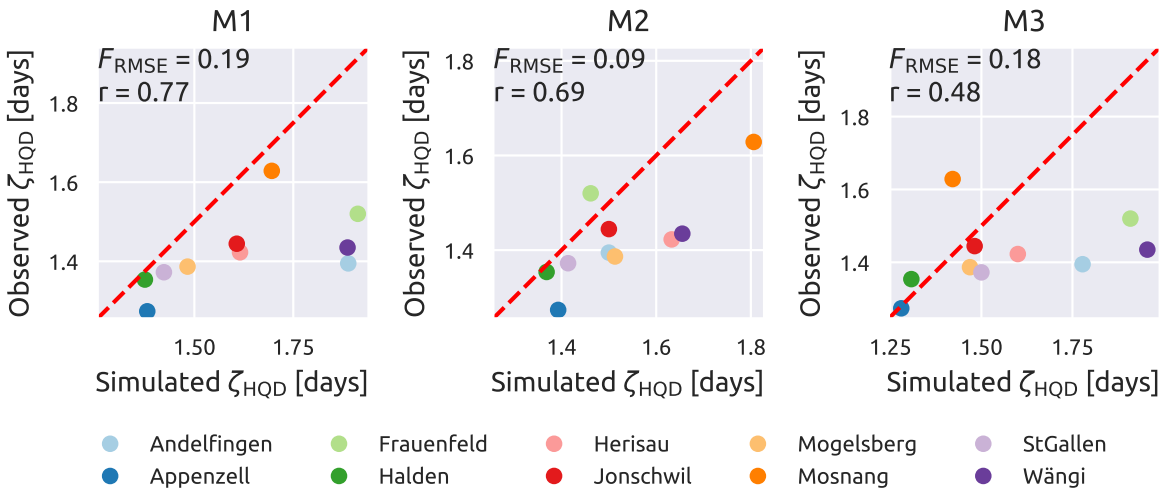

Figure 3.10: Ability of the hydrological models to represent the signature duration of low-flow events (ζ_{HQD}). Single-HRU model **(a)**, two-HRU model based on geology **(b)**, and two-HRU model based on land use **(c)**.

1. *Hypothesis 1: precipitation is the first driver of differences in the water balance.* The good performance of model Mo in the representation of the mean annual streamflow (ζ_Q) suggests that accounting for the spatial heterogeneity of the precipitation alone is sufficient to achieve a good representation of the annual water balance. More complex models, with more HRUs, processes, and parameters, while resulting in an overall improvement of time series metrics, do not result in any improvement in simulating the water balance signature ζ_Q.

2. *Hypothesis 2: snow-related processes control differences in streamflow seasonality.* The improvement in the representation of the streamflow seasonality ζ_{HFD} by M1 can be largely attributed to the (spatially variable) effect of snow accumulation and melting. More complex models (M2 and M3) do not demonstrate an improvement in this signature, indicating that the structural differences between these models do not have an influence on this signature.

3. *Hypothesis 3: geology controls the partitioning between quickflow and baseflow.* The ability of M2 to match the signature ζ_{BFI}, which quantifies the separation between quickflow and baseflow, much better than the other models, supports the hypothesis that geology has a strong control on the partitioning between quickflow and baseflow. M2 is also the model with the average best performance in terms of streamflow metrics.

4. *Hypothesis 4: characteristics that do not show correlations do not influence streamflow variability.* The overall lower performance of M3 compared to M2, in terms of both signatures and streamflow metrics, reassures us that the relatively good results of M2 are not just due to increasing complexity and confirms that adding characteristics that do not show correlations does not improve the representation of spatial variability.

In summary, distributing the inputs in space and accounting for the spatial distribution of snow-related processes are sufficient to get good performance metrics of water balance and seasonality, confirming the fact that only the precipitation rate and the partitioning between rainfall and snow are the first controls on these hydrograph characteristics. However, in order to capture other important characteristics of the hydrograph, described by signatures such as ζ_{BFI}, the discretization of the catchment in HRUs is necessary. This discretization has to be carefully made and a preliminary analysis to understand dominant influence factors on signatures can help in this decision. As shown in figure 3.9, if such discretization uses landscape characteristics that are not strongly correlated with the signatures (e.g. land use), the results are worse than if we choose characteristics that show a strong correlation with signatures (e.g. geology). This means that M2 is capable of capturing the signatures not just because it is more complex than M1, but because it incorporates the causality link between the geology and the streamflow signatures into its structure.

3.5 GENERAL DISCUSSION

Explaining the spatial variability observed in catchment response is a major focus of catchment hydrology and a central theme in classification studies (e.g., McDonnell and Woods, 2004; Thorsten et al., 2007). A common approach for interpreting the spatial variability of catchment response is to identify relationships between climatic or catchment characteristics and streamflow signatures. This is typically done through correlation-based analyses (e.g., Lacey and Grayson, 1998; Bloomfield et al., 2009), which however carry the limitations that correlation does not always imply causality and that the presence of multiple correlated variables can obscure process interpretation.

In this study, we combine a correlation analysis for identifying the dominant influence factors on streamflow signatures with hydrological modelling by using the interpretation of the first analysis as an inspiration for generating testable model hypotheses. The combination of correlation analysis on streamflow signatures and hydrological modelling is beneficial because on the one hand, the speculations on dominant processes resulting from the correlation analyses can be verified in the modelling process. Specifically, we developed model experiments to test the influence of precipitation spatial distribution on streamflow average and seasonality and the influence of geology on quickflow vs. baseflow partitioning. On the other hand, model building benefits from the guidance resulting from the preliminary signature analysis. The construction of a distributed model requires several decisions (e.g., Fenicia et al., 2016), including how to "break up" the catchment in a meaningful way, and preliminary signature analysis can motivate some of these decisions. For example, the definition of HRUs based on geology, which was suggested by the signature analysis, resulted in models with better performance than models using HRUs defined on the basis of land use, particularly in the representation of streamflow signatures that reflect the baseflow vs. quickflow partitioning.

Although several modelling decisions were guided by data analysis, it should be noted that alternative decisions would have been similarly consistent with the data. For example, both precipitation and elevation are correlated with average streamflow, and geology, topography, and soil-type characteristics are correlated between each other and with the baseflow index (section 3.4.1.2 and figure 3.5). The correlation of catchment characteristics (e.g. geology, soil, and topography) can be attributed to the fact that they have evolved together in the shaping of the catchment morphology (e.g. mountainous regions have impervious topography with shallower soil and, for these reasons, are less suitable for human activities, influencing land use). The decisions on which variables are chosen to reflect a causality link are not always obvious from correlation analysis alone, and they require expert judgement, which is necessarily subjective. Although subjectivity is difficult to avoid, it is important to be transparent about the decision taken and the argumentations on which they are based, how weak or strong they may be, so that they can be reappraised and revised if new evidence is acquired.

Although our results in terms of hypotheses 1–4 described in the previous section appear justifiable based on previous work, they are not a priori obvious. In terms of the first hypothesis, although it is known that precipitation has a strong control on the average streamflow, it is less clear whether the spatial variability in the streamflow average can only be attributed to precipitation: some authors, for example, pointed to the role of regional groundwater flow in affecting the water balance (e.g., Bouaziz et al., 2018); GR4J (e.g., Perrin et al., 2003), for example, has a parameter that quantifies catchment gains and losses. Our modelling experiments, in particular through M0, have shown that groundwater processes, which potentially alter the water balance, do not influence the mean streamflow spatial variability of the Thur catchment.

In terms of the snow processes, although it is clear that, when there is snow (as in this case), the model needs to have a snow component, it is less obvious (at least just by looking at hydrographs) how many of the differences in the seasonality of the streamflow response between catchments are due to snow. The objective of the comparison between M0 and M1 is not to show that adding a snow component improves the overall performance, but that the differences in seasonality are captured by the model only when the snow component is integrated.

In terms of the effect of geology, Kuentz et al. (2017) made a classification study over more than 40'000 catchments across all Europe (of which almost 2700 are gauged) and found that geology controls the BFI, topography the flashiness index, and, for most of the cases, land use is the second control of them; Bloomfield et al. (2009) used a linear regression model and linked the lithology of the Thames Basin (UK) with the BFI; Lacey and Grayson (1998) noted that geology controls the BFI in two ways, storing the water and impacting the soil formations; Fenicia et al. (2016) compared different model structures and catchment discretization methods in the Attert Basin (Luxemburg) and discovered that the best model was the one that incorporates a spatial representation of the meteorological inputs and of the geology. On the other hand, this general tendency should not be generalized to all places. For example, Mazvimavi et al. (2005) found that geology was not important for the BFI, as in their case study the aquifer was deep and disconnected from the

river. Bouaziz et al. (2018) found a strong influence of regional groundwater flow in the Meuse catchment which altered the water balance.

The choice of streamflow signatures is based on the large-sample study from Addor et al. (2017), which provides a broad range of signatures typically used in hydrology. Our analyses showed that this selection is rather inclusive, with several strongly correlated signatures. For this reason, we eventually used a much smaller selection of the original set of signatures (12 in the original set vs. 5 in the final set). Although hundreds of signatures have been proposed in the literature (e.g., Olden and Poff, 2003), the apparent inclusivity of the set from Addor et al. (2017) provides confidence that the main properties of streamflow are captured in our study. However, it does not guarantee that this set of signatures is sufficient in representing streamflow time series.

One of the main limitations of this work is the restricted number of catchments involved and the limited spatial extension of the study. For this reason, it is difficult to generalize the results to other climatic regions. The subcatchments all belong to the same region, and the landscape and climatic characteristics, while varying substantially within the basin, are still a small sample of the characteristics found elsewhere. Moreover, although the model evaluation uses validation in space and time, which is a relatively incisive test, the spatial validation is carried out in a nested setup. The application of systematic model development strategies to other places and scales, and spatial validation to entirely different regions, is necessary to obtain more generalizable insights.

The small number of subcatchments involved in this study (10) limits the range of viable methods for identifying relationships between landscape and climatic indices and streamflow signatures (section 3.3.1) to rather simple approaches. In particular, our correlation analysis, although accounting for non-linearity, is limited to monotonic correlations between variables, and it is unable to identify other forms of relationship, including the mutual interaction between various influence factors. The usage of more advanced techniques, including machine learning approaches such as random forest or clustering analyses, is most efficient when larger samples are available and could represent a more suitable choice in these situations.

3.6 CONCLUSION

In this study, we presented the development process of a distributed model where model hypotheses, instead of being made a priori, are informed by preliminary analysis on determining the dominant climatic and landscape controls on streamflow spatial variability. Besides providing guidance to model development, the proposed approach is useful in the fact that modelling can be used to test specific hypotheses on dominant processes resulting from such preliminary analysis. Our analysis was applied to the Thur catchment, subdivided into 10 subcatchments based on available stream gauging stations. The main findings are summarized in the following points.

1. We found large spatial variability between the subcatchments of the Thur in terms of various streamflow signatures reflecting multiple temporal scales: yearly, seasonal, and event scale. In terms of climatic characteristics, indices reflecting fraction of snow, precipitation totals,

and aridity varied considerably among catchments. Other precipitation characteristics such as season, frequency, and duration of dry and wet days did not vary significantly among catchments. In terms of landscape characteristics, there is large variability of topography (e.g. from upstream mountainous to downstream flat areas), geology (with unconsolidated, more permeable, and consolidated, relatively impermeable formations), and soils (with low depths in the mountains and large depths in the floodplains) in all the catchments.

2. Based on correlation analysis and expert judgement, we determined that climatic variables, especially the precipitation average, are the main controls on streamflow average yearly values; the fraction of snow is responsible for streamflow seasonality by delaying the release of winter precipitation to the spring season, and geology controls the baseflow index, with a higher fraction of unconsolidated material determining higher baseflow.

3. The results of the signature analysis were translated into a set of model hypotheses: a model with uniform parameters and distributed precipitation input (M0), the addition of a snow component (M1), the subdivision of the catchment into geology-based HRUs (M2), and the alternative subdivision of the catchment using land-use-based HRUs (M3).

4. Using model comparison and a validation approach that considers model performance (also in terms of signatures) in space–time validation, we found that it is necessary to account for the heterogeneity of precipitation, snow-related processes, and landscape features such as geology in order to produce hydrographs that have signatures similar to the observed ones. In particular, we confirmed that M0, in spite of a generally poor performance, is sufficient to capture signatures of streamflow average, showing that only distributing the meteorological inputs is sufficient to explain regional differences in average streamflow and that other phenomena potentially altering the water balance (e.g. regional groundwater flows) do not play a significant role. M1 improves signatures of streamflow seasonality, showing that snow is the main cause of the variability of the seasonality among the catchments. M2 enables signatures such as the baseflow index to be reproduced, showing that incorporating the geology of the catchment is important for reproducing regional differences in baseflow. Model modifications that are not in line with the results of the signature analysis, such as subdividing the catchment using land-use-based HRUs (M3), despite leading to the same complexity as M2, cause deterioration in model performance in space–time validation. Overall, these results confirm the hypotheses based on the signature analysis and suggest that the causality relationships, explaining the influence of climate and landscape characteristics on streamflow signatures, can be constructively used for distributed model building.

The relatively good performance obtained in space–time validation suggests that the proposed approach could be used for the prediction of the streamflow in other ungauged locations within the Thur catchment. The method proposed uses data that are commonly available in many gauged catchments (e.g. meteorological data, streamflow measurements, maps of elevation, geology, land use, and soil); therefore, it is easily transferable to other locations.

APPENDICES

3.A HYDROLOGICAL MODEL DETAILS

3.A.1 *Model equations*

The equations of the model are listed in this Appendix; the model structure in presented in figure 3.6. Table 3.6 contains the model parameters with the range of variability used in calibration, table 3.7 lists the water-budget equations, and table 3.8 and presents the constitutive functions used.

Table 3.6: Hydrological model parameters with the range of variation used for the definition of the uniform prior distribution. The "component" column indicates the element (reservoir, lag, or network) where the parameter belongs.

Parameter	Unit	Component	Range of variability
C_e	-	Unsaturated reservoir (UR)	0.1 – 3.0
S_{max}^{UR}	mm	Unsaturated reservoir (UR)	0.1 – 500.0
k_{WR}	d^{-1}	Snow reservoir (WR)	0.1 – 10.0
t_{rise}^{IL}	d	Network lag	0.5 – 10.0
t_{rise}^{OL}	d	Network lag	0.5 – 10.0
D	-	Structure	0.0 – 1.0
k_{FR}	$d^{-1}mm^{-2}$	Fast reservoir (FR)	10–6 – 10.0
k_{SR}	d^{-1}	Slow reservoir (SR)	10–6 – 1.0
t_{rise}^{lag}	d	Structure lag	1.0 – 20.0

Table 3.7: Water–budget equations (see model schematic in figure 3.6).

Component	Equation
Snow reservoir (WR)	$\frac{dS_{WR}}{dt} = P_{WR} - Q_{WR}$
Unsaturated reservoir (UR)	$\frac{dS_{UR}}{dt} = P_{UR} - E_{UR} - Q_{UR}$
Splitter	$Q_{UR} = P_{SR} + P_{lag}$
Slow reservoir (SR)	$\frac{dS_{SR}}{dt} = P_{SR} - Q_{SR}$
Fast reservoir (FR)	$\frac{dS_{FR}}{dt} = P_{FR} - Q_{FR}$
Outflow	$Q = Q_{FR} + Q_{SR}$

Table 3.8: Constitutive functions of the model.

Component	Equation
Snow reservoir (WR)	$P_{WR} = \begin{cases} P & \text{if } T \leqslant 0 \\ 0 & \text{if } T > 0 \end{cases}$
Snow reservoir (WR)	$M_{WR}^{max} = \begin{cases} 0 & \text{if } T \leqslant 0 \\ k_{WR} T & \text{if } T > 0 \end{cases}$
Snow reservoir (WR)	$Q_{WR} = M_{WR}^{max} \left[1 - \exp\left(-\frac{S_{WR}}{2}\right)\right]$
Unsaturated reservoir (UR)	$\overline{S_{UR}} = \frac{S_{UR}}{S_{UR}^{max}}$
Unsaturated reservoir (UR)	$E_{UR} = C_e\,(PET)\left[\frac{\overline{S_{UR}}(1+0.01)}{1+0.01}\right]^{\beta_{UR}}$
Unsaturated reservoir (UR)	$Q_{UR} = P_{UR}\left(\overline{S_{UR}}\right)$
Slow reservoir (SR)	$P_{SR} = DQ_{UR}$
Slow reservoir (SR)	$Q_{SR} = k_{SR} S_{SR}$
Lag function	$P_{FR} = \left(P_{lag} * h_{lag}\right)(t)$
Lag function	$h_{lag} = \begin{cases} 2t \Big/ \left(t_{rise}^{lag}\right)^2 & \text{if } t \leqslant t_{rise}^{lag} \\ 0 & \text{if } t > t_{rise}^{lag} \end{cases}$
Fast reservoir (FR)	$Q_{FR} = k_{FR} S_{FR}^3$
Lags in the network	$h_{lag}^{net} = \begin{cases} 2t \Big/ \left(t_{rise}^{OL/IL}\right)^2 & \text{if } t \leqslant t_{rise}^{OL/IL} \\ \left(1 \Big/ t_{rise}^{OL/IL}\right)\left(1 - \left(\left(t - t_{rise}^{OL/IL}\right) \Big/ t_{rise}^{OL/IL}\right)\right) & \text{if } t_{rise}^{OL/IL} < t \leqslant 2t_{rise}^{OL/IL} \\ 0 & \text{if } t > 2t_{rise}^{OL/IL} \end{cases}$

4

EXPLORING SIGNATURE-BASED CALIBRATION OF HYDROLOGICAL MODELS FOR PREDICTION IN UNGAUGED BASINS

Achieving reliable and precise streamflow predictions in ungauged basins (PUB) remains an open challenge in catchment hydrology. A possible approach to PUB is to calibrate rainfall-runoff models to estimated streamflow "signatures". Estimating parameter and prediction uncertainty in this case is not trivial because: (1) calibration takes place in the signature domain, while predictions are necessary in the time domain, and (2) the streamflow signatures are estimated (e.g. extrapolated from donor catchments) rather than measured, and therefore subject to modelling uncertainty. To overcome these challenges, this paper investigates model calibration using estimated signatures in a Bayesian framework. First, we calibrate a stochastic signature transfer model, based on seasonal flow duration curves. Then, we calibrate a hydrological model to the estimated signatures, thus accounting for both regionalization and hydrological model uncertainties. Approximate Bayesian Computation (ABC) is employed to avoid the need to explicitly derive and evaluate the likelihood function. The proposed method is tested in six subcatchments of the Thur catchment (Switzerland). Three data availability scenarios are considered: (1) concomitant scenario, where signatures are measured, (2) non-concomitant scenario, where signatures are transposed from a different time period, and (3) regionalized scenario, where signatures are regionalized using a model. Results show limited degradation in the quality of streamflow predictions when moving from concomitant to regionalized scenarios. The overall performance is slightly worse than classic time-domain calibration. These results indicate a useful advance in probabilistic prediction in ungauged basins.

This chapter is based on the publication: Dal Molin, M., Kavetski, D., Albert, C., Schirmer, M., and Fenicia, F. (2021) "Exploring Signature-Based Calibration of Hydrological Models for Prediction in Ungauged Basins" In preparation, *Water Resources Research*.

4.1 INTRODUCTION

Prediction in Ungauged Basins (PUB) is an important research theme in catchment hydrology. Many human activities (e.g. flood protection, water resources management, ecosystem preservation, etc.) require reliable and precise hydrological predictions that are difficult to obtain when observed data, typically used to calibrate the models used to make predictions, is not available. PUB has motivated a large body of work (e.g., review studies of Parajka et al., 2013; Spence et al., 2013), including during the scientific decade 2003-2012 of the International Association of Hydrological Sciences (IAHS) (Sivapalan et al., 2003b; Hrachowitz et al., 2013). Another decade later, PUB continues to attract research attention, due to its practical importance and its unresolved challenges (e.g., Kratzert et al., 2019; Prieto et al., 2019).

One of the main objectives of PUB is predicting streamflow and its uncertainty in locations where historical observations are not available (e.g., He et al., 2011; Parajka et al., 2013). The common approach for generating streamflow predictions is through a hydrological ("rainfall-runoff") model, which contains parameters that need to be estimated. When streamflow observations are available, the model parameters are typically estimated by calibration to the observed time series. In ungauged basins, this approach is not applicable, and alternative solutions are necessary. These alternative approaches can be classified into three main categories:

1. *Calibration to estimated signatures*. Here, the term "signatures" refers to summary statistics that reflect meaningful traits of catchment streamflow time series, such as the timing of peaks, relative baseflow volume, shape of hydrograph recessions, etc. (e.g., Sivapalan, 2006; Addor et al., 2017). Streamflow signatures based on the Flow Duration Curve (FDC) have received particular attention as streamflow signatures, as they convey key properties of the marginal distribution of streamflow and are used widely in hydrological engineering applications (e.g., Castellarin et al., 2013). Once the signatures at the ungauged location are estimated, model calibration then proceeds by optimizing an objective function that quantifies the mismatch between regionalized and simulated signatures (e.g., Yu and Yang, 2000; Yadav et al., 2007; Lombardi et al., 2012; Shafii and Tolson, 2015; Kim et al., 2017). Alternatively, Bayesian inference approaches have also been proposed, using the estimated signatures either to construct the prior distribution of model parameters (e.g., Bulygina et al., 2009, 2011) or to create an approximation of the likelihood function (e.g., Castiglioni et al., 2010; Bulygina et al., 2012; Almeida et al., 2016; Prieto et al., 2019).

2. *Regionalization of model parameters*. In the case of conceptual models, the parameters can be estimated by calibration in gauged catchments that are "similar" (in terms of either physical properties or geographical proximity), or by regression to observable catchment attributes (constructed across a large number of catchments) (e.g., Seibert, 1999; Mwakalila, 2003; Merz and Blöschl, 2004; Parajka et al., 2005; Zhang and Chiew, 2009; Singh et al., 2014). When using physically based models, which in principle may not require calibration (e.g., Dornes

et al., 2008; Fang et al., 2010), or regional models, the parameters can be directly measured in the field, which however require appropriate fieldwork campaigns (e.g., Abdulla and Lettenmaier, 1997).

3. *Use of complementary sources of data*. Data such as satellite observations and/or spot measurements, can constrain the model parameters without using continuous streamflow measurements (e.g., Mohamed et al., 2006; Winsemius et al., 2008; Hingray et al., 2010; Lerat et al., 2012; Pool et al., 2017; Nijzink et al., 2018).

This study concentrates on the first approach, namely the estimation of streamflow signatures in the ungauged catchment using a signature model followed by the calibration of a hydrological model to these estimated signatures. Within this approach, it is convenient to distinguish three scenarios:

1. *Transfer of signatures in time*, i.e., from a time period where streamflow observations are available to a time period where such data is not available (in the same catchment). This scenario is often referred to as "non-concomitant calibration" (NCC), and represents arguably the simplest ungauged scenario. NCC was explored by Montanari and Toth (2007), who proposed the use of a likelihood based on the spectral properties of the time series, and by Hingray et al. (2010), who showed how NCC can compensate for the absence of useful data in catchments heavily impacted by hydraulic works.

2. *Transfer of signatures in space*, i.e., from a gauged location to an ungauged location. These approaches are typically based on regression to catchment properties (e.g., Berger and Entekhabi, 2001; Yadav et al., 2007; Castiglioni et al., 2009; Addor et al., 2018; Prieto et al., 2019);

3. *Estimation of signatures using mechanistic models* that, instead of searching for similar periods or catchments to extrapolate the signatures from, derive process-based equations that relate catchment and meteorological characteristics with streamflow signatures. Examples of this approach include the studies by Betterle et al. (2017); Booker and Woods (2014); Doulatyari et al. (2015).

An important challenge associated with data availability and model approximations is the treatment of streamflow predictive uncertainty. Predictive uncertainty quantification when the model is calibrated to estimated signatures is particularly challenging. First, calibration takes place in the signature domain, while predictions are necessary in the time domain. Second, streamflow signatures are extrapolated from other catchments rather than calculated directly from local observed streamflow data, and hence may contain appreciable additional uncertainty.

Uncertainty estimation in the context of PUB has relied mainly on heuristic "limits of acceptability" approaches. For example, Westerberg et al. (2011) searched for hydrological model parameters that produce FDCs similar to the observed ones at some selected quantiles. Winsemius et al. (2009)

used several observed signatures to impose hard and soft constrains to select suitable parameter sets. Yadav et al. (2007) used regionalized signatures and accepted only model parameters that produce signatures that fall into an a priori estimated range of variability of the regionalized signatures. A limitation of these approaches from our perspective is that they yield predictive uncertainty estimates that are not interpretable in a statistical sense.

Although the Bayesian approach could in principle address these limitations, its applications to-date have not focused on the quantification of the different sources of streamflow uncertainty in PUB. For example, previous studies have explored different methodologies for conditioning model parameters on regionalized signatures (e.g., Bulygina et al., 2009, 2011; Castiglioni et al., 2010; Prieto et al., 2019) but have considered mainly the uncertainty related to the regionalization process and neglected other sources (e.g., uncertainly related to the hydrological model); usually, these studies weight model parameter sets based on their ability to produce simulated signatures that are close to the regionalized ones and, then, report the parametric predictive uncertainty of the simulated hydrographs. Other studies (e.g., Almeida et al., 2016) report general performance metrics of the model (e.g., Nash-Sutcliffe efficiency) and its ability to represent the signatures, i.e. without providing an explicit documentation of the hydrograph predictive uncertainty.

The estimation of uncertainty in the time domain while performing calibration in the signature domain can be pursued using Approximate Bayesian Computation (ABC), as demonstrated in previous publications (e.g., Vrugt and Sadegh, 2013; Nott et al., 2014; Fenicia et al., 2018; Kavetski et al., 2018). However, these previous studies focused on scenarios where signatures are computed directly from available streamflow observations, which is not the case for ungauged catchments.

In this study, we focus on streamflow prediction and uncertainty estimation at ungauged locations, calibrating a rainfall-runoff model to signatures derived from estimated seasonal FDCs within a Bayesian inference framework implemented using ABC. The proposed approach provides separate treatment of two key sources of uncertainty that arise in the modelling process, namely regionalization uncertainty and hydrological uncertainty.

The specific objectives are as follows:

1. Introduce a novel methodology for the calibration of hydrological models to signatures that are estimated rather than computed from observed streamflow;

2. Assess the ability of the proposed methodology to provide reliable and precise predictions, in the following data availability scenarios:

   a) calibration to concomitant signatures, i.e. using observed FDCs;

   b) calibration to non-concomitant signatures, i.e. extrapolating the FDCs from another time period;

   c) calibration to regionalized signatures, i.e. using a regionalization model to estimate the FDCs.

3. Assess the performance of signature-domain calibration in broader contexts, including in comparison to

a) classical time-domain calibration (to appraise potential loss of quality in model predictions);

b) prediction using basic prior parameter ranges (to demonstrate that calibration to regionalized signatures is a viable and useful approach for PUB).

The case study is based on data from six sub-catchments of the Thur catchment in Switzerland, which are used to simulate multiple ungauged scenarios. A lumped conceptual hydrological model is applied separately in each of the sub-catchments.

The remainder of the paper is organized as follows. Section 4.2 presents the theory behind the proposed methodology and section 4.3 details the algorithm implementations. Section 4.4 describes the case study setup, section 4.5 reports the case study results, and section 4.6 discusses these results and their implications. Section 4.7 draws the conclusions.

4.2 THEORY

4.2.1 *General overview*

Figure 4.1 shows a schematic of the theoretical framework used for the inference of the parameters of the hydrological model. The following scenarios are central to this study:

- Concomitant signatures (CS), where streamflow observations are available at the location and time period of interest. This scenario provides the baseline for the other two scenarios. The signatures are calculated directly from the observed data.
- Non-concomitant signatures (NCS), where streamflow observations are available at the location of interest but not in the time period of interest. The signatures in the time period of interest are estimated using data from the available time period.
- Regionalized signatures (RS), where streamflow observations at the location of interest are not available. The signatures at the location of interest are estimated using a signature model and data from other locations.

Panel (a) describes scenario CS, where signatures are observed, and panel (b) describes the scenarios NCS and RS, where in both cases, the signatures are estimated. We use the term *target signatures* to indicate, regardless of the scenario considered, the signatures that the hydrological model has to "match" during the calibration. The method used to estimate the target signatures varies depending on the scenario.

In scenario CS (panel a), the target signatures $\tilde{\mathbf{y}}$ are calculated directly from observed streamflow data $\tilde{\mathbf{q}}$ using a deterministic function g, hence $\tilde{\mathbf{y}} = g(\tilde{\mathbf{q}})$ (refer to section 4.2.3.1). The parameters $\boldsymbol{\theta}^{(H)}$ of the hydrological model $\mathbf{Q}^{(H)}$ are inferred in the signature domain. The inference process compares the prior signatures $\mathbf{Y}^{(H)}_{prior} = g\left(\mathbf{Q}^{(H)}\right)$ to the observed ones $\tilde{\mathbf{y}} = g(\tilde{\mathbf{q}})$. The result is the

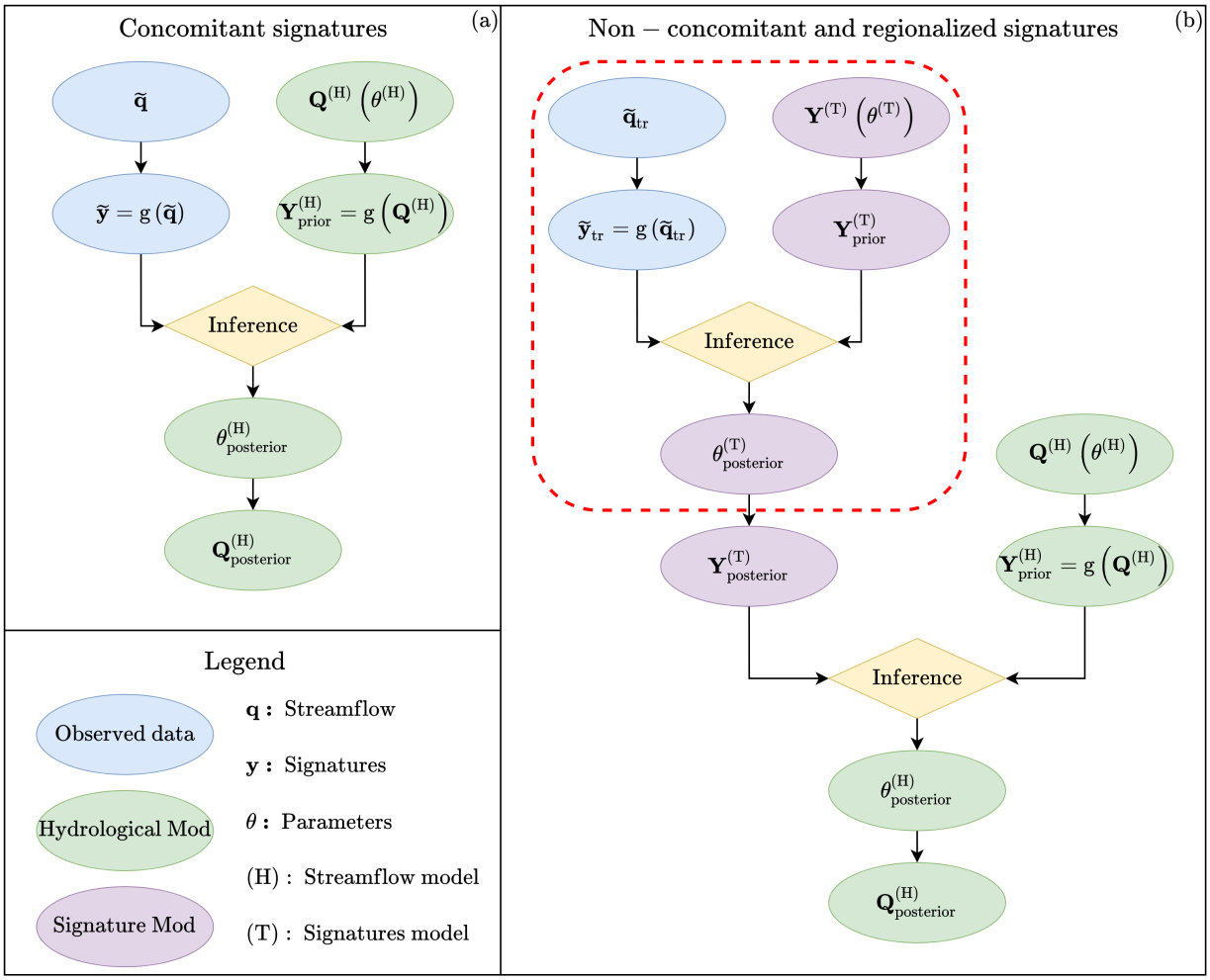

Figure 4.1: Schematic of the inference setup used in this work to estimate the streamflow model parameters $\theta^{(H)}$. Panel **(a)** illustrates the inference in a scenario CS, where signatures can be computed directly from observed streamflow. Panel **(b)** illustrates the inference in scenarios NCS and RS, where signatures are estimated using a model, which itself requires calibration (red dashed box).

posterior distribution of the parameters $\theta^{(H)}_{posterior}$, which can then be used to generate posterior streamflow distributions $\mathbf{Q}^{(H)}_{posterior}$.

In scenarios NCS and RS (panel b), a signature transfer model $\mathbf{Y}^{(T)}\left(\theta^{(T)}\right)$ (refer to sections 4.2.3.2 and 4.2.3.3) is employed to estimate the target signatures. This results in a two-step inference process. In the first step (shown in the red dashed box), the parameters $\theta^{(T)}$ of the transfer model are inferred using signatures $\tilde{\mathbf{y}}_{tr}$ calculated from observed data $\tilde{\mathbf{q}}_{tr}$ from another time period (scenario NCS) or catchment (scenario RS). This step estimates the target signatures $\mathbf{Y}^{(T)}_{posterior}$. In the second step, the posterior distribution of the parameters of the hydrological model $\theta^{(H)}_{posterior}$ is estimated. This step is similar to the one shown in panel (a) and ultimately produces the posterior streamflow distributions $\mathbf{Q}^{(H)}_{posterior}$. The two inference steps are carried out separately.

In addition to the three principal scenarios, the case study also considers two auxiliary scenarios, concomitant hydrograph (CH) and prior simulation (PS), which are used to put the current

analysis in a broader context, as described in section 4.4.6. Scenario CH represents the classical time-domain calibration where the parameters of the hydrological model are inferred using the streamflow observations directly; the inference follows the same procedure used, for example, in the study by Fenicia et al. (2018) and it is based on the application of the Bayes equation (see equation 4.5 in section 4.2.4) in the time domain. Scenario PS does not involve any inference of model parameters as they are sampled directly from their prior distribution to simulate streamflow. Given their marginality w.r.t. the theoretical development of this paper, these two scenarios are omitted from the methodological description.

The procedures depicted in figure 4.1 are detailed next.

4.2.2 *Hydrological model*

The uncertain streamflow predictions $\mathbf{Q}^{(H)}$ are generated by a stochastic hydrological model. For example, typical stochastic models in hydrological modelling are obtained by combining a deterministic model $\mathbf{m}^{(H)}$ with a random residual error term $\boldsymbol{\epsilon}^{(H)}$; for simplicity our presentation is based on this prototypical example.

The stochastic model equation is hence

$$z\left[\mathbf{Q}^{(H)}\left(\theta^{(H)}, \mathbf{x}\right), \theta_z^{(H)}\right] = z\left[\mathbf{m}^{(H)}\left(\theta_m^{(H)}, \mathbf{x}\right), \theta_z^{(H)}\right] + \boldsymbol{\epsilon}^{(H)}\left(\theta_\epsilon^{(H)}\right) \tag{4.1}$$

where z is a transformation (e.g. Box-Cox) that accounts for the heteroscedasticity and skew in the residuals (e.g., McInerney et al., 2017); its parameters are denoted as $\theta_z^{(H)}$. The hydrological model parameters are denoted as $\theta^{(H)}$, and in turn are composed by parameters of the deterministic model $\theta_m^{(H)}$ and parameters of the residual error model $\theta_\epsilon^{(H)}$. The term $\mathbf{x}$ denotes all fixed inputs to the model (e.g., precipitation, potential evapotranspiration, etc.).

Specific modelling choices taken in the case studies are explained section 4.4.2.

4.2.3 *Streamflow signatures*

4.2.3.1 *Scenario CS (concomitant signatures)*

In scenario CS, the hydrological model is calibrated to streamflow signatures computed directly from the observed streamflow time series in the calibration period and catchment. The signature estimation process can be represented as

$$\tilde{\mathbf{y}} = g(\tilde{\mathbf{q}}) \tag{4.2}$$

where g is a deterministic function that transforms the observed streamflow $\tilde{\mathbf{q}}$ in signatures $\tilde{\mathbf{y}}$.

The choice and definition of g, i.e. the selection of signatures, is a specific modelling choice; our choices for the case study are detailed in section 4.4.3.

4.2.3.2 *Scenario NCS (non-concomitant signatures)*

In scenario NCS, the hydrological model is calibrated to estimated streamflow signatures derived from the observed streamflow time series $\tilde{\mathbf{q}}_{tr}$ observed in the same catchment but in another time period. The signature estimation process, which introduces uncertainty related to the transfer of the signatures, is represented as

$$z\left[\mathbf{Y}^{(T)}\left(\theta_{\epsilon}^{(T)}, \tilde{\mathbf{q}}_{tr}\right), \theta_{z}^{(T)}\right] = z\left[g\left(\tilde{\mathbf{q}}_{tr}\right), \theta_{z}^{(T)}\right] + \boldsymbol{\epsilon}^{(T)}\left(\theta_{\epsilon}^{(T)}\right) \tag{4.3}$$

where $\mathbf{Y}^{(T)}$ are the estimated signatures and $\theta_{\epsilon}^{(T)}$ are the parameters of the signatures transfer error model $\boldsymbol{\epsilon}^{(T)}$.

The choice of transformation z and of the transfer error model are application-specific; our choices for the case study are detailed in section 4.4.4.1.

4.2.3.3 *Scenario RS (regionalized signatures)*

In scenario RS, the hydrological model is calibrated to streamflow signatures calculated using a deterministic signature model $\mathbf{m}^{(T)}$ that is constructed (offline) using streamflow data $\tilde{\mathbf{q}}_{tr}$ from "donor" catchments. The uncertainty introduced by the signature model is represented using a residual error model $\boldsymbol{\epsilon}^{(T)}$, also constructed using data from the donor catchments.

The stochastic signature model is defined as the sum of a deterministic model and a transfer error term:

$$z\left[\mathbf{Y}^{(T)}\left(\theta^{(T)}, \tilde{\mathbf{q}}_{tr}, \mathbf{x}\right), \theta_{z}^{(T)}\right] = z\left[\mathbf{m}^{(T)}\left(\theta_{m}^{(T)}, \tilde{\mathbf{q}}_{tr}, \mathbf{x}\right), \theta_{z}^{(T)}\right] + \boldsymbol{\epsilon}^{(T)}\left(\theta_{\epsilon}^{(T)}\right) \tag{4.4}$$

The specific choices of the model $\mathbf{m}^{(T)}$, of the transformation z, and of the transfer error made in this study are detailed in section 4.4.4.2.

4.2.4 *Bayesian inference approach*

The posterior distribution of hydrological model parameters, $p\left(\theta^{(H)}|\mathbf{x}, \tilde{\mathbf{y}}\right)$, is given by Bayes equation as follows

$$p\left(\theta^{(H)}|\mathbf{x}, \tilde{\mathbf{y}}\right) = \frac{p\left(\tilde{\mathbf{y}}|\theta^{(H)}, \mathbf{x}\right) p\left(\theta^{(H)}\right)}{p\left(\tilde{\mathbf{y}}\right)} \tag{4.5}$$

where $p\left(\tilde{\mathbf{y}}|\theta^{(H)}, \mathbf{x}\right)$ is the likelihood function, $p\left(\theta^{(H)}\right)$ is the prior distribution of the model parameters, and $p\left(\tilde{\mathbf{y}}\right)$ is a normalization constant (given by the marginal distribution of the observed signatures).

In scenario CS, where the target signatures are deterministic, equation 4.5 can be used directly to estimate the posterior of $\theta^{(H)}$; the algorithmic implementation is described in section 4.3.

In contrast, in scenarios NCS and RS, where the target signatures are themselves explicitly formulated as the output of a stochastic model, equation 4.5 cannot be applied directly.

To accommodate scenarios NCS and RS, we formulate the posterior distribution in "expanded" form using the total probability integral,

$$p\left(\theta^{(H)}|\mathbf{x},\tilde{\mathbf{y}}\right) = \int p\left(\theta^{(H)}|\mathbf{x},\mathbf{z}\right) p\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}}\right) d\mathbf{z} \tag{4.6}$$

where $p\left(\theta^{(H)}|\mathbf{x},\tilde{\mathbf{y}}\right)$ and $p\left(\theta^{(H)}|\mathbf{x},\mathbf{z}\right)$ are the posterior distribution of the hydrological model parameters $\theta^{(H)}$ w.r.t. the observed signatures $\tilde{\mathbf{y}}$ and the simulated signatures $\mathbf{z}$, respectively; $p\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}}\right)$ is the posterior distribution of the simulated signatures w.r.t. the observed ones.

Substituting equation 4.5 into equation 4.6 yields

$$p\left(\theta^{(H)}|\mathbf{x},\tilde{\mathbf{y}}\right) = \int \frac{p\left(\mathbf{z}|\theta^{(H)},\mathbf{x}\right) p\left(\theta^{(H)}\right)}{p\left(\mathbf{z}\right)} p\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}}\right) d\mathbf{z} \tag{4.7}$$

The three scenarios differ in the formulation of the term $p\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}}\right)$ in equation 4.7.

In scenario CS, the uncertainty in the observed signatures is treated as part of the error term already present in the hydrological model in equation 4.1, as that error includes also the observational uncertainty on the streamflow measurements. Therefore, in this case, the probability distribution collapses to a Dirac function and the integral simplifies back to the Bayes equation 4.5

$$p\left(\theta^{(H)}|\mathbf{x},\tilde{\mathbf{y}}\right) = \int \frac{p\left(\mathbf{z}|\theta^{(H)},\mathbf{x}\right) p\left(\theta^{(H)}\right)}{p\left(\mathbf{z}\right)} \delta\left(\mathbf{z}-\tilde{\mathbf{y}}\right) d\mathbf{z} = \frac{p\left(\tilde{\mathbf{y}}|\theta^{(H)},\mathbf{x}\right) p\left(\theta^{(H)}\right)}{p\left(\tilde{\mathbf{y}}\right)} \tag{4.8}$$

In scenarios NCS and RS this simplification cannot be made. The term $p\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}}\right)$ represents the probability density function of the stochastic signature model $\mathbf{Y}^{(T)}$, whose parameters are pre-calibrated to the value $\hat{\theta}^{(T)}$. Therefore, equation 4.7 becomes

$$p\left(\theta^{(H)}|\mathbf{x},\tilde{\mathbf{y}}\right) = \int \frac{p\left(\mathbf{z}|\theta^{(H)},\mathbf{x}\right) p\left(\theta^{(H)}\right)}{p\left(\mathbf{z}\right)} p_{\mathbf{Y}^{(T)}}\left(\mathbf{z}|\mathbf{x},\tilde{\mathbf{y}},\hat{\theta}^{(T)}\right) d\mathbf{z} \tag{4.9}$$

The posterior parameter distribution in equations 4.8 and 4.9 is sampled using an ABC procedure as described next in section 4.3.

4.3 IMPLEMENTATION OF SAMPLING ALGORITHM: MODIFIED SABC ALGORITHM

This section describes the sampling algorithm used to estimate the posterior distribution of the parameters $\theta^{(H)}$ of the hydrological model. Compared to classical time-domain calibration, signature-domain calibration brings the complication that the likelihood function is not available in closed form. Moreover, the signatures themselves can be treated either as a fixed given value (scenario CS) or as random variables to represent their estimation uncertainty (scenarios NCS and RS).

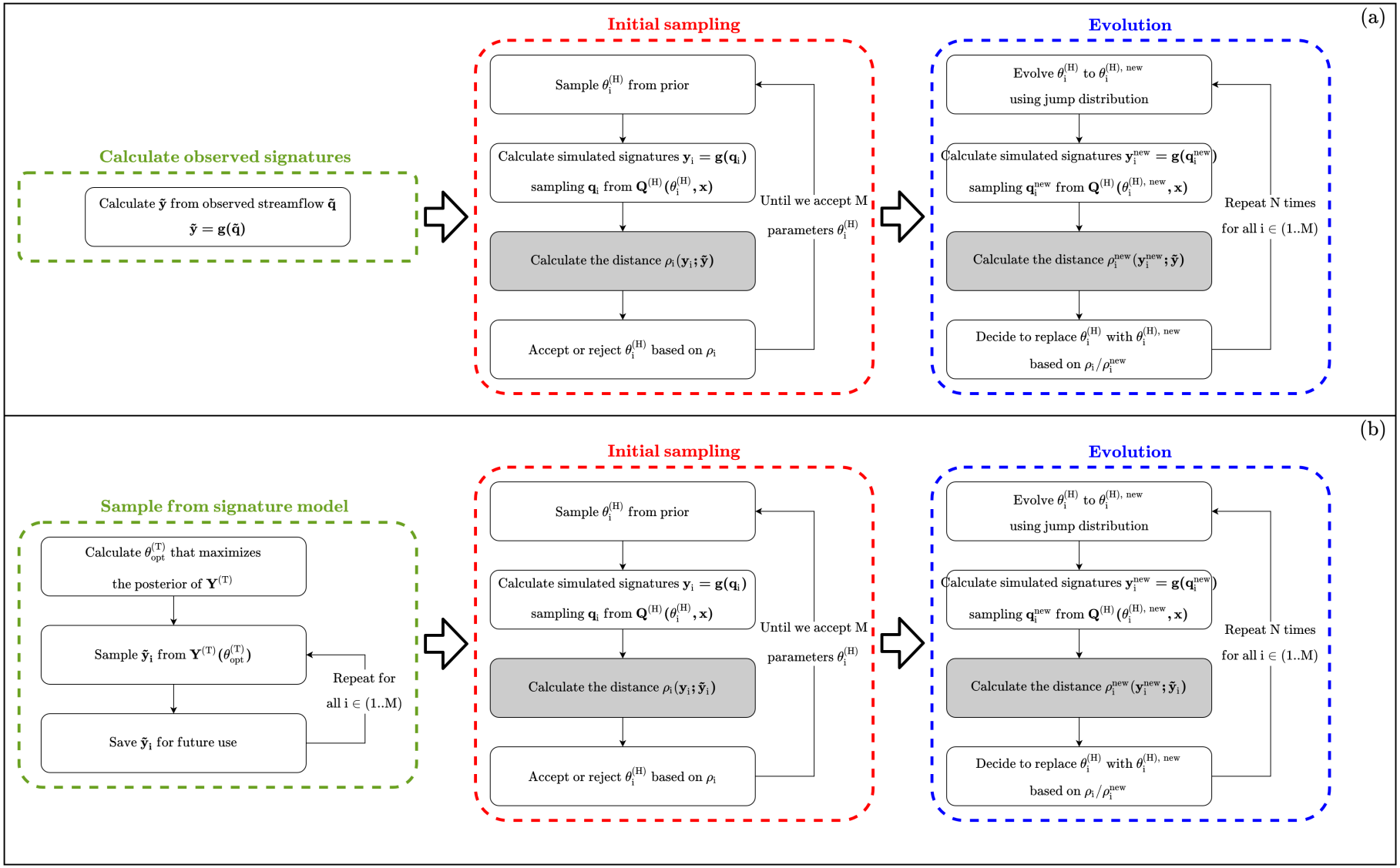

Figure 4.2: Schematic of the algorithm used to infer the parameters of the hydrological model. Panel **(a)** shows the algorithm used in scenario CS; panel **(b)** shows the one used in scenarios NCS and RS.

The Approximate Bayesian Computation (ABC) sampling algorithm used in this work is based on the SABC algorithm proposed in Albert et al. (2015), but employs some modifications to take into account the usage of stochastic (rather than fixed) target signatures.

Figure 4.2 outlines the algorithm. Panel (a) shows the workflow applied in the case of deterministic signatures (scenario CS); for this scenario procedure is the same as in the earlier study by Kavetski et al. (2018) and Fenicia et al. (2018). Panel (b) shows the algorithm modifications developed to handle stochastic signatures (scenarios NCS and RS).

In both cases, the algorithm is composed by three separate phases:

1. Calculation of the target signatures (green box);
2. Sampling of an initial population of M model parameters $\theta_i^{(H)}$ using a constant acceptance/rejection threshold based on the distance ρ (defined in appendix 4.A) between simulated and target signatures (red box);
3. Evolution of the population of particles $\theta_i^{(H)}$ using a Metropolis step with the acceptance/rejection threshold evolving according to the annealing schedule described in Albert et al. (2015) (blue box).

A detailed description of the standard algorithm shown in panel (a) is provided in Albert et al. (2015); see also Kavetski et al. (2018). The presentation here focuses on the modifications designed in this work to adapt the SABC algorithm to the case of stochastic target signatures (panel b). Two important algorithmic differences are noted.

The first difference is in the input signatures themselves. In the deterministic case (panel a, green box), a fixed set of signatures is calculated from observed streamflow data. In the stochastic case (panel b, green box), a population of M different target signatures is sampled from the pre-calibrated signature model (see appendix 4.B for details on the calibration of this model).

The second difference is in the calculation of the distance between simulated and target signatures (red and blue boxes). The step is indicated with a grey background in figure 4.2. In the deterministic case the target signatures always have the same value $\tilde{\mathbf{y}}$ (note the absence of the subscript i). In the stochastic case, the target signatures $\tilde{\mathbf{y}}_i$ are attached to each particle $\theta_i^{(H)}$. This key modification allows for the incorporation of the uncertainty of the stochastic signatures into the SABC sampling algorithm.

4.4 CASE STUDY MATERIALS AND METHODS

4.4.1 *Catchment*

The study is carried out in the Thur catchment (figure 4.3), an alpine and pre-alpine catchment located in north-eastern Switzerland, south of Lake Constance. Its subcatchments, while belonging to the same watershed, present appreciable variability in terms of physical characteristics, meteorological inputs, and streamflow response (Dal Molin et al., 2020).

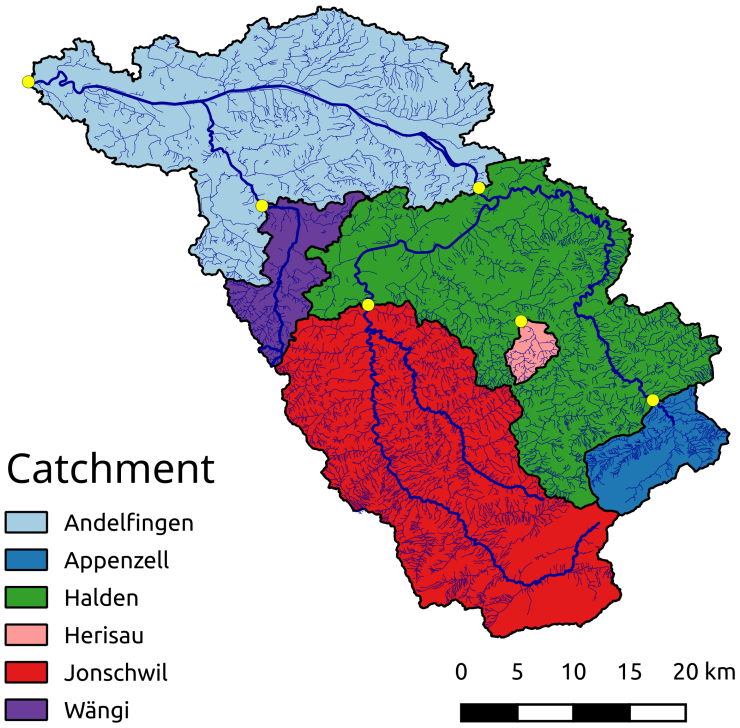

Figure 4.3: Map of the Thur catchment. Subcatchments are indicated with different colors, the yellow dots represent the streamflow gauging stations.

The catchment has high quality observation data, having been studied intensively in the last 40 years. Detailed physical characteristics of the Thur catchment and a summary of previous investigations can be found in Dal Molin et al. (2020). The work of Doulatyari et al. (2017) on the estimation of FDCs is particularly relevant to this case study, as detailed in section 4.4.4.2.

The streamflow time series are obtained from the Swiss Federal Office for the Environment (FOEN). The precipitation, temperature, and potential evapotranspiration (PET) time series are derived from data provided by the Swiss Federal Office of Meteorology and Climatology MeteoSwiss; details about the methodology used to interpolate the data from the meteorological stations and the calculation of the PET are given in section 2 of Dal Molin et al. (2020).

All time series used in this study are daily and span the period 1981-2005, with only one gap in the streamflow data in Herisau, from 31 December 1982 to 9 May 1983. The 24 years of data are divided into 3 periods of 8 years each for creating "virtual" ungauged scenarios (details on the subdivision are provided in appendix 4.B).

4.4.2 *Hydrological model*

The deterministic hydrological model $\mathbf{m}^{(H)}$ used in equation 4.1 is based on the popular hydrological model HyMod (Boyle, 2001), with the inclusion of a snow reservoir.

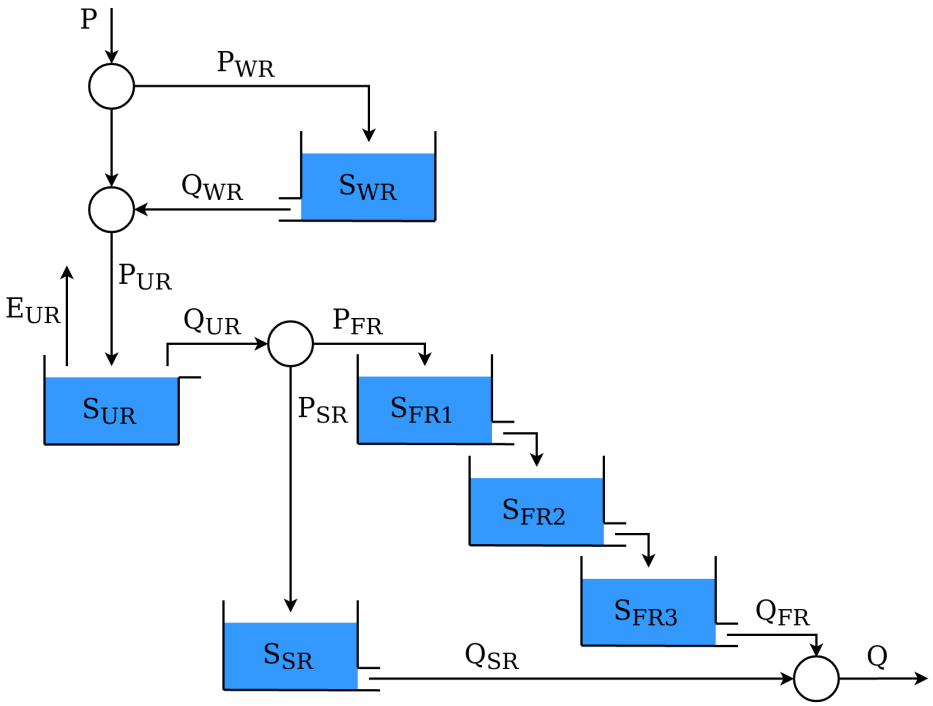

Figure 4.4: Schematic representation of the lumped streamflow model used in the case study. "P" represents the precipitation entering in the reservoirs, "E" the evaporation, and "Q" the outflow from the reservoirs. The subscripts indicate the reservoirs: WR = snow reservoir, UR = unsaturated reservoir, FR = fast reservoir, SR = slow reservoir. The governing equations are reported in 4.C.

This model is chosen for its low computational and data requirements, and for being able to provide reasonable fits to the data based on previous studies in the Thur catchment (Dal Molin et al., 2020).

The model has four elements, shown schematically in figure 4.4. The snow reservoir (WR) intercepts the incoming precipitation and releases it according to the input temperature, in order to simulate snow accumulation and melting. The unsaturated reservoir (UR) partitions the combined precipitation and snowmelt into a portion that builds storage, which eventually evaporates, and a portion that eventually produces streamflow. The latter component is partitioned between a cascade of three fast reservoirs (FR) and a slow reservoir (SR). The fast reservoirs are intended to generate the peaks of the hydrograph and their offset, while the slow reservoir is intended to produce the baseflow.

The model is implemented using the SUPERFLEX modelling framework (Fenicia et al., 2011). A fixed step implicit Euler time stepping scheme is used for numerical stability. The equations and the calibrated parameters are detailed in appendix 4.C.

The residual error model is built to take into account the heteroscedastic and autocorrelated nature of the residuals. The Box-Cox transformation (Box and Cox, 1964) defined as

$$BC[x, \lambda] = \frac{x^{\lambda} - 1}{\lambda} \tag{4.10}$$

is used in equation 4.1 ($z = BC$). The value of λ is fixed to 0.2 (McInerney et al., 2017).

The autocorrelation in the residuals is represented using a first-order autoregressive model (AR1),

$$\epsilon_t^{(H)} = \phi^{(H)} \epsilon_{t-1}^{(H)} + W_t^{(H)} \tag{4.11}$$

where $\epsilon_t^{(H)}$ is the residual error at the time step t, $\phi^{(H)}$ is the autoregressive parameter and $W_t^{(H)}$ is the innovation (random noise term).

The innovations are assumed to follow a truncated Gaussian distribution

$$W_t^{(H)} \sim \mathcal{N}_T \left(0, \sigma_\Omega^{(H)}, L_{\omega,t}, +\infty\right) \tag{4.12}$$

with zero mean, variance $\sigma_\Omega^{(H)}$, and lower bound $L_{\omega,t}$ set such that $\mathbf{Q}_t{}^{(H)} \geqslant 0$.

The residual error model is, therefore, defined by two parameters $\theta_\epsilon^{(H)} = \left[\phi^{(H)}, \sigma_\Omega^{(H)}\right]$; the value of $\phi^{(H)}$ is fixed to 0.8 while σ_Ω is inferred. These choices follow the findings of Evin et al. (2014) and McInerney et al. (2017).

4.4.3 *Definition of streamflow signatures*

The streamflow signatures are derived from the seasonal FDCs, which can be either calculated from observed streamflow (scenarios CS and NCS) or obtained using a model (scenario RS). In the first case, the streamflow time series $\mathbf{q}$ is divided in four seasonal partitions $\mathbf{q}_s$, with $s = 1, ..., 4$ referring to the season (i.e., September, October, and November for Autumn, etc.).

In the following equations, FDCs are expressed as cumulative distribution functions (CDFs) and indicated with the letter F. The signatures $\mathbf{y}_i$ are defined as slopes between consecutive quantiles of the seasonal CDFs. Mathematically, the function g used to calculate the signatures can be defined as follows:

$$\mathbf{y}_i = g(\mathbf{q}_s, i) = \begin{cases} F^{-1}(\zeta_i, \mathbf{q}_s) / \zeta_i & i = 1 \\ \frac{F^{-1}(\zeta_i, \mathbf{q}_s) - F^{-1}(\zeta_{i-1}, \mathbf{q}_s)}{\zeta_i - \zeta_{i-1}} & i > 1 \end{cases} \tag{4.13}$$

where $F^{-1}(\zeta, \mathbf{q}_s)$ is the inverse cumulative distribution function of a data set $\mathbf{q}_s$ evaluated at percentile ζ_i.

Each season is represented by a vector of N slopes, with N fixed to 19. The percentiles ζ_i are selected to have higher resolution for higher flows while excluding maxima (adapted from Westerberg et al., 2011):

$$\frac{\zeta_i}{100} = 1 - \left(\frac{i+2}{N+3}\right)^2, \quad \text{for } i = 1, ..., N \tag{4.14}$$

Compared to direct usage of streamflow quantiles, the usage of slopes (equation 4.13) facilitates the definition of the signature error model in scenarios NCS and RS since it relaxes some requirements like monotonicity and autocorrelation. Moreover, this choice does not reduce

the amount of information available, compared to the direct usage of streamflow quantiles, as thanks to the definition of the first slope ($i = 1$ in equation 4.13), "parallel" FDCs can still be distinguished.

4.4.4 *Signature models*

4.4.4.1 *Scenario NCS (non-concomitant signatures)*

The model for non-concomitant signatures is formulated based on equation 4.3 as follows:

$$\mathrm{BC}\left[\mathbf{Y}^{(T)}\left(\theta_{\epsilon}^{(T)}, \tilde{\mathbf{q}}_{tr}\right), \lambda_{BC}^{(T)}\right] = \mathrm{BC}\left[g\left(\tilde{\mathbf{q}}_{tr}\right), \lambda_{BC}^{(T)}\right] + N\left(0, \sigma_{\epsilon}^{(T)}\right) \tag{4.15}$$

where $\lambda_{BC}^{(T)} = 0.2$, and $\sigma_{\epsilon}^{(T)}$ is inferred in the same catchment in a period where data is assumed to be available (see appendix 4.B).

4.4.4.2 *Scenario RS (regionalized signatures)*

This study employs the deterministic model proposed by Doulatyari et al. (2015) and subsequently applied in the Thur catchment (Doulatyari et al., 2017). This model defines the probability density function (pdf) of streamflow $\mathbf{q}^{(T)}$, up to a constant, with the following expression:

$$\mathrm{pdf}\left(q_s{}^{(T)}\right) \propto \left(q_s{}^{(T)}\right)^{-a} \exp\left(-\frac{\left(q_s{}^{(T)}\right)^{2-a}}{\alpha k(2-a)} + \frac{l\left(q_s{}^{(T)}\right)^{1-a}}{k(1-a)}\right) \tag{4.16}$$

where α and l represent the mean precipitation and the frequency of effective precipitation events (i.e. precipitation events intense enough to generate streamflow), respectively; k and a represent a the coefficient and the exponent of the hydrograph recession, respectively.

The parameters in equation 4.16 are estimated as follows. Parameters α and l are estimated from daily rainfall (and snowfall, since the model uses the precipitation without distinguishing by its form) time series. Parameters a and k are estimated from geomorphic properties of the catchment. This study uses the values of the parameters reported in Table 2 of Doulatyari et al. (2017) for the Thur catchment, where l is indicated with the symbol λ.

The model $\mathbf{m}^{(T)}$ used in equation 4.4 is built integrating numerically equation 4.16 in order to calculate the CDF of the streamflow $F(\mathbf{q}_s)$ that is, then, used in equation 4.13 to calculate the signatures.

The specific choices regarding transformation z and error model $\epsilon^{(T)}\left(\theta_{\epsilon}^{(T)}\right)$ in equation 4.4 are motivated by residual analysis of $\mathbf{m}^{(T)}$ in the gauged catchments. Based on these analyses, it was observed that residuals are heteroscedastic, and characterized by a consistent difference (bias) between the observed and the modeled FDCs (e.g., Figure 4 in Doulatyari et al., 2017). The heteroscedasticity is represented by making the variance of the residuals proportional to the magnitude of modelled signatures,

$$\boldsymbol{\epsilon}^{(T)}\left(b_{\epsilon}^{(T)}\right) = N\left(0, b_{\epsilon}^{(T)}\mathbf{m}^{(T)}\right) \tag{4.17}$$

As a consequence of this choice, the transformation z in equation 4.4 is not needed.

The signature model is "bias-corrected" by applying a multiplier $a_{\epsilon}^{(T)}$ to the modelled signatures. Equation 4.4 is therefore simplified to

$$\mathbf{Y}^{(T)}\left(\theta^{(T)}, \tilde{\mathbf{q}}_{tr}, \mathbf{x}\right) = a_{\epsilon}^{(T)}\mathbf{m}^{(T)}\left(\theta_{m}^{(T)}, \tilde{\mathbf{q}}_{tr}, \mathbf{x}\right) + \boldsymbol{\epsilon}^{(T)}\left(\theta_{\epsilon}^{(T)}, \mathbf{m}^{(T)}\right) \tag{4.18}$$

The inferred model parameters are $\theta_{\epsilon}^{(T)} = \left[a_{\epsilon}^{(T)}, b_{\epsilon}^{(T)}\right]$.

4.4.5 *Performance metrics*

The quality of model predictions in each scenario is evaluated by calculating some performance metrics both on the time series (hydrograph) and the cumulative distribution (FDC). The metrics considered are:

- Nash-Sutcliffe efficiency of the median of the predictive distribution, which measures the fit of the simulated data to the observations;
- Precision, which measures the spread of the simulated time series;
- Reliability, which measures the consistency of the observations with the predictive distribution;
- Volumetric bias, which measures the long-term water balance error of simulations.

The metrics definition is detailed in appendix 4.D.

4.4.6 *Experiments*

Three experiments are carried out:

- *Experiment 1* compares the three signatures-domain calibration scenarios (CS, NCS, and RS) in order to appraise potential loss of quality in model predictions when moving from gauged to ungauged conditions.
- *Experiment 2* compares the best signatures-domain calibration scenario (CS) with the time-domain calibration (CH), in order to appraise potential loss of quality in model predictions when moving from time series to signatures calibration.
- *Experiment 3* compares the results of scenario RS with the prior predictive distribution, i.e., hydrographs generated using parameters sampled from the prior distribution (PS), in order

to assess the extent to which calibration to regionalized signatures is informative beyond what is already known a priori.

Experiment 1 is the main experiment of this study. Experiments 2 and 3 are auxiliary experiments to help put the findings in better overall context.

The results are shown with two levels of detail. First, we consider a representative catchment and time period, namely Andelfingen and time period P3 (see appendix 4.B), where we report detailed results including hydrographs and annual FDCs. Second, we consider all catchments and time periods, and report performance using aggregated metrics.

For experiment 1, we include a comparison of the posterior distribution of the model parameters in the three scenarios. Specifically, we illustrate cases of parameters being well identified or weakly identified in all the scenarios and cases where the identifiability of the parameters varies among scenarios.

4.5 RESULTS

4.5.1 *Experiment 1: signature calibration in multiple scenarios*

4.5.1.1 *Representative catchment: hydrograph and FDC representation*

Figure 4.5 shows the results of scenarios CS, NCS and RS for the representative Andelfingen catchment in period P3. Panel (a) shows the simulated hydrographs and panel (b) shows the simulated FDCs. Panels (c) and (d) show the values of the performance metrics.

The hydrographs in panel (a) show a consistent difference in the width of the uncertainty bounds between the scenario CS and the other two. Scenario CS has the narrowest predictive uncertainty, scenario RS comes second, followed closely by scenario NCS, whose uncertainty bands enclose the ones of the other two scenarios. This difference in uncertainty is also confirmed by the precision metric of the hydrograph (panel d), which is the lowest in the scenario CS (0.42) and assumes comparable values in the other two scenarios (0.57 for scenario NCS and 0.56 for scenario RS).

Still in panel (a), the median simulated hydrograph (dashed lines) is more dynamic (i.e. more reactive to external forcing) and closer to the observed data, especially on high flows, in the scenario CS. In scenarios NCS and RS, on the other hand, the median hydrograph is less dynamic, with deeper recession and limited ability to reach the observed peaks. This qualitative assessment is confirmed by the value of the Nash-Sutcliffe efficiency (panel c), which is highest in the scenario CS (0.64), followed by scenarios RS and NCS, both with a value of 0.59.

In terms of FDC performance, it can be observed that all the scenarios tend to underestimate the peak flows. Apart from that, the comparison of the FDCs shows results similar to the ones emerged from the comparison of the hydrographs. In particular, the simulation of scenario CS produces the most precise predictive distributions and has the median closest to the observed

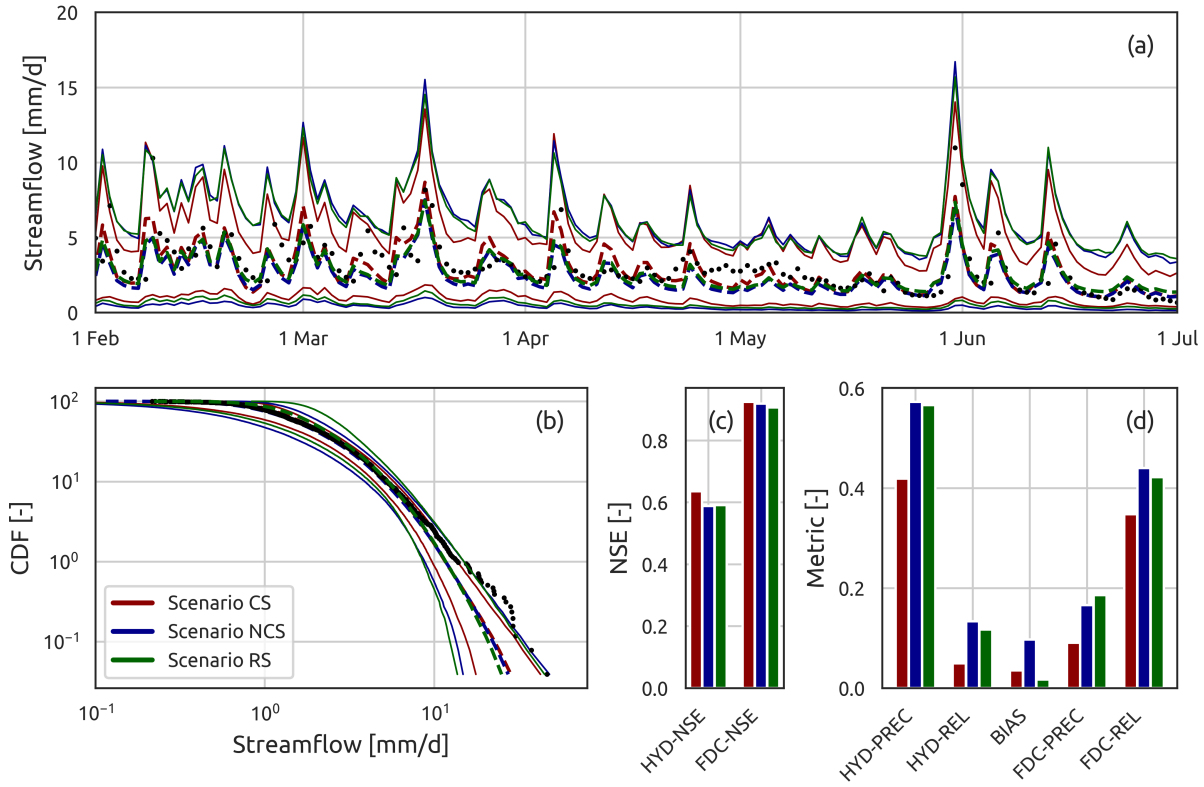

Figure 4.5: Results of experiment 1 for a representative catchment (Andelfingen) and time period (period P3). Panel **(a)** shows the hydrograph, panel **(b)** the annual FDC, panels **(c)** and **(d)** the metrics used for evaluating the simulations. In panels **(a)** and **(b)**, the dashed line represents the median simulation; the solid lines represent 95% uncertainty bands

values. The predictive distributions of scenarios NCS and RS have comparable precision and intersect each other, with scenario NCS simulation that underestimates the medium-low flows. Note that, due to the logarithmic scale of the vertical axis, the highest 1% of the streamflow values occupies almost half of the plot area – hence a large visual mismatch may result in a limited difference in the metrics.

These qualitative findings are confirmed by the quantitative FDC metrics shown in panels (c) and (d):

- Nash-Sutcliffe efficiency slightly declines when going from scenario CS (0.92) to NCS (0.92) and RS (0.91);

- the precision metric has similar values for scenarios NCS and RS (0.17 and 0.18) while it improves for scenario CS (0.09);

- the volumetric bias is substantially higher in scenario NCS and confirms the general tendency of the simulations, seen in the FDCs, to underestimate the flow values.

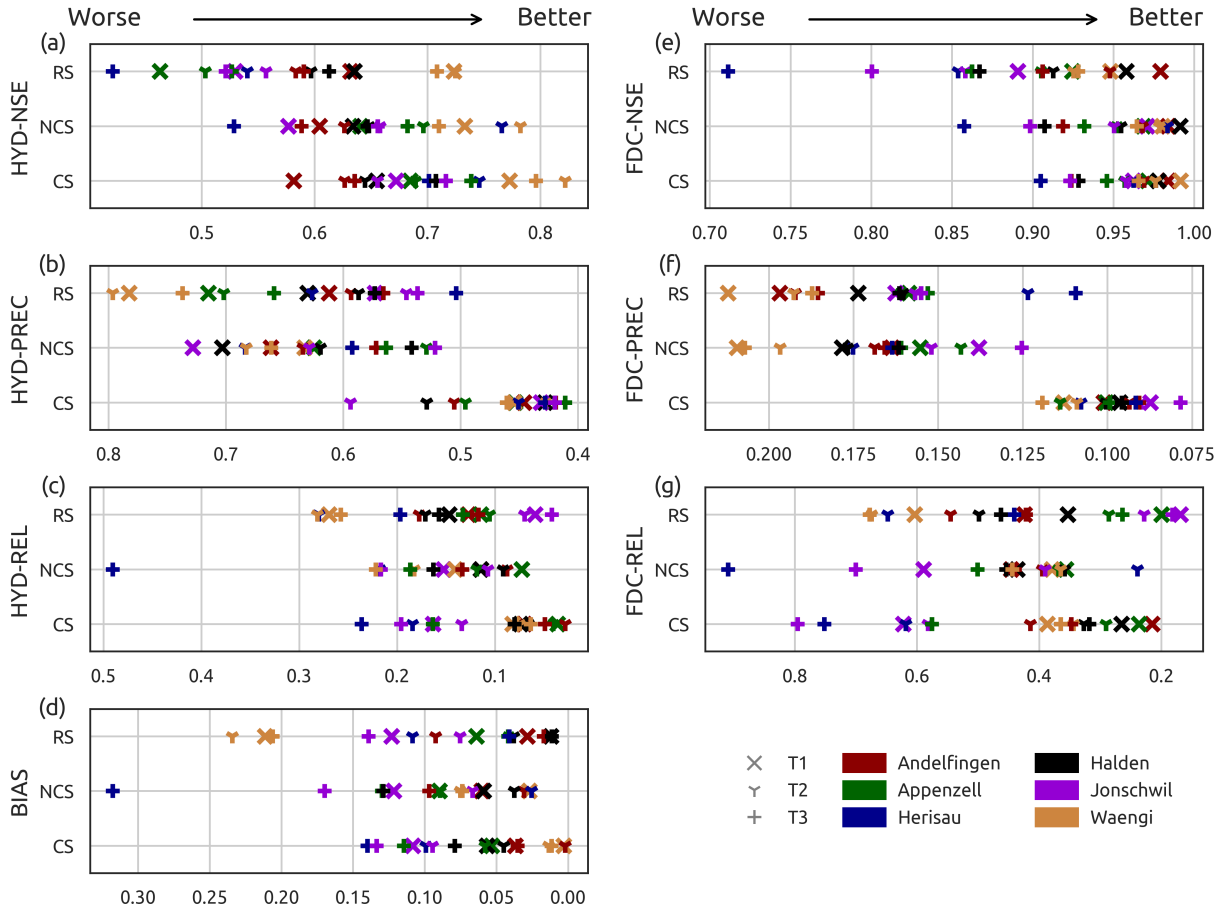

Figure 4.6: Performance metrics achieved by the simulations in experiment 1. The left column reports the metrics calculated on the hydrograph; the right column reports the metrics calculated on the annual FDC. Catchments are distinguished by color, periods are distinguished by symbol. Note that some horizontal axes are reversed so that "better" metric values appear consistently on the right in all plots.

4.5.1.2 *Performance over all catchments*

Figure 4.6 shows the performance metrics calculated for all the catchments in all periods. The figure is subdivided in seven panels, with the left column showing the metrics calculated on the hydrograph and the right column showing the FDCs metrics. Each point in the plots represents the value of the metric for the simulations of a specific catchment and period. Scenarios are evaluated based on the compactness and mean position of the group of points. Compactness indicates the consistency of the simulations composing the scenario; a position towards the right-hand side of the plot indicates good mean performance (note that for metrics precision, reliability, and volumetric bias the horizontal axes are flipped so that “better” results are always on the right, for consistency with the other metrics).

Panel (a) shows the Nash-Sutcliffe efficiency of the median hydrograph. All scenarios have similar spread but the mean position of the points show a clear ranking: scenario CS in the first place (mean value 0.70), NCS second (0.66), and RS third (0.58). In all scenarios, the points that refer to the same catchment (same color) tend to be clustered together.

Panel (b) shows the precision metric of the hydrograph. In this case, there is a clear trend among the scenarios, both in terms of mean position and compactness, with scenario CS that is the best (mean precision 0.46 and variance of $2.2x10^{-3}$), NCS that comes second (mean 0.62, variance $3.5x10^{-3}$) and RS third (mean 0.63, variance $7.1x10^{-3}$).

Panel (c) and (d) show the reliability metric of the hydrograph and the volumetric bias. For both metrics, all three scenarios perform similar in terms of mean value and spread. It is noted that the simulations in the catchment Herisau in period P3 in scenario NCS have considerably worse performance metrics than the other catchments in the same scenario.

Panel (e) shows Nash-Sutcliffe efficiency of the median FDC. There is a clear trend in the scenarios: CS has the highest mean (0.96) and compactness (variance $5.5x10^{-4}$), NCS keeps a similar mean (0.95) but increases the spread (variance $1.2x10^{-3}$), and RS deteriorates both in the mean (0.89) and in the variance ($3.9x10^{-3}$).

Panel (f) shows the precision metric of the FDC. Scenarios NCS and RS have a similar performance in terms of mean (0.17 for both) and spread (variance $4.8x10^{-4}$ and $6.4x10^{-4}$), with scenario RS that has two simulations that perform substantially better than the rest. Both scenarios NCS and RS are strongly outperformed by scenario CS (mean 0.10, variance $1.1x10^{-4}$), with almost no simulation performing better in scenarios NCS and RS than in scenario CS.

Panel (g), finally, shows the reliability metric of the FDC. All the scenarios perform similarly, with the presence of one bad outlier in the scenario NCS (Herisau, period P3).

4.5.1.3 *Posterior distribution of the model parameters*

Figure 4.7 shows the posterior distributions of selected parameters in the three scenarios. The parameters have been selected as representative of three common behaviors:

1. *Parameters that are identifiable in all scenarios*, with posterior distributions that are quite distinct from the priors. An example is given by parameter D, which controls the split of outflow from the unsaturated reservoir between the fast and the slow reservoirs.

2. *Parameters that are well identified in the scenario CS but lose identifiability in the scenarios NCS and RS*. An example is given by parameter k_{WR}, which controls the outflow of the snow reservoir;

3. *Parameters that are weakly identified by all the scenarios*, with the posterior distributions that differ moderately from the priors. An example is given by parameter k_{SR}, which controls the outflow of the slow reservoir;

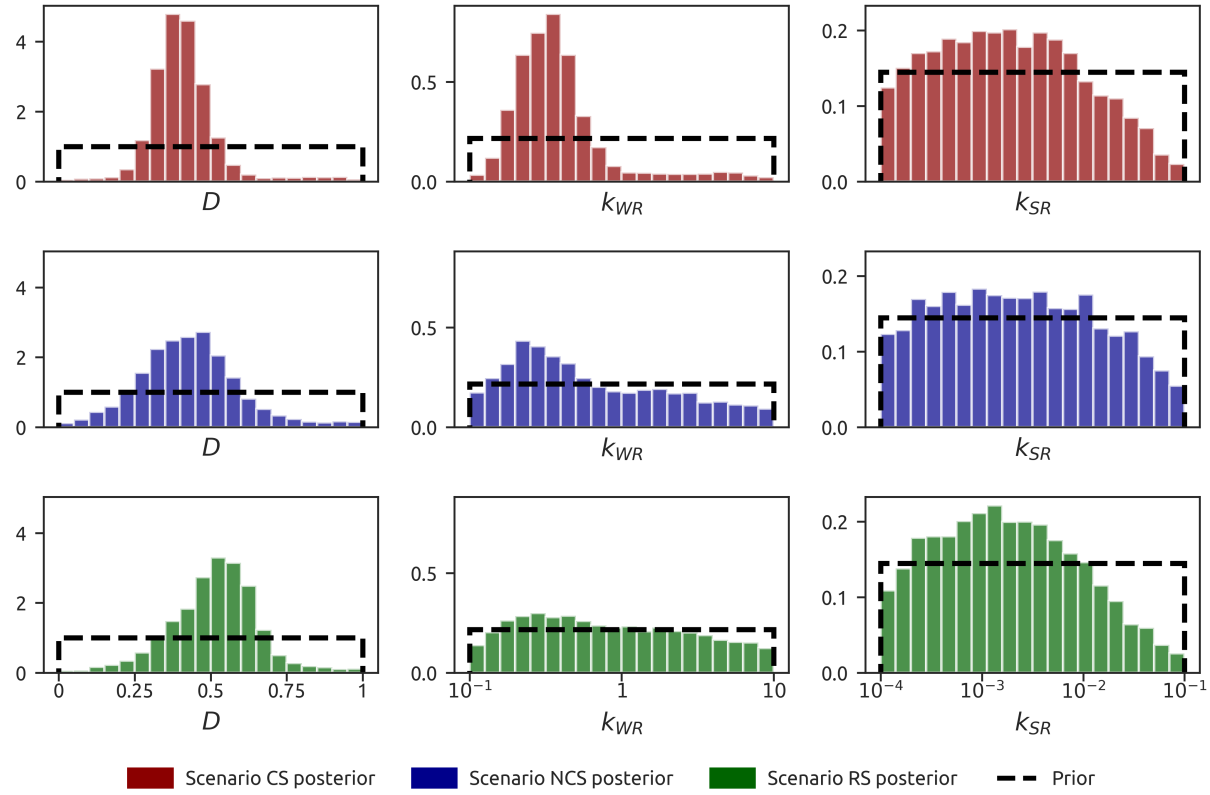

Figure 4.7: Posterior distributions for three selected parameters, demonstrating qualitatively different behaviors in the scenarios. The histogram bars represent the posterior distributions. The black dashed lines represent the prior distributions.

4.5.2 *Experiment 2: comparison with time-domain calibration*

4.5.2.1 *Representative catchment: hydrograph and FDC representation*

Figure 4.8 compares the results of the two gauged scenarios in the subcatchment Andelfingen, in the period P3; the two scenarios are model calibration to observed signatures (scenario CS) and model calibration to the observed time series (scenario CH).

Panel (a) shows the simulated hydrographs. The simulation appears, overall, more precise in the scenario CS, with scenario CH that has a large uncertainty in the high flows and the lower uncertainty bound that is higher than the one of scenario CS for most of the time. This tendency is also confirmed by the precision metric shown in panel (d), where the scenario CS performs better (0.42 compared to 0.56 of scenario CH).

The median hydrograph (dashed line) is more dynamic in the scenario CS and represents better the magnitude of the peaks. On the other hand, the median hydrograph in scenario CH captures better the timing of some peaks (e.g., in the month of March). The Nash-Sutcliffe efficiency (panel c) shows a slightly better performance of the scenario CH (0.69 vs. 0.64 for scenario CS).

It is worth noting the "strange" behavior of scenario CH from the end of April and during May where the median hydrograph is almost flat and the uncertainty bounds shrink. This behavior is

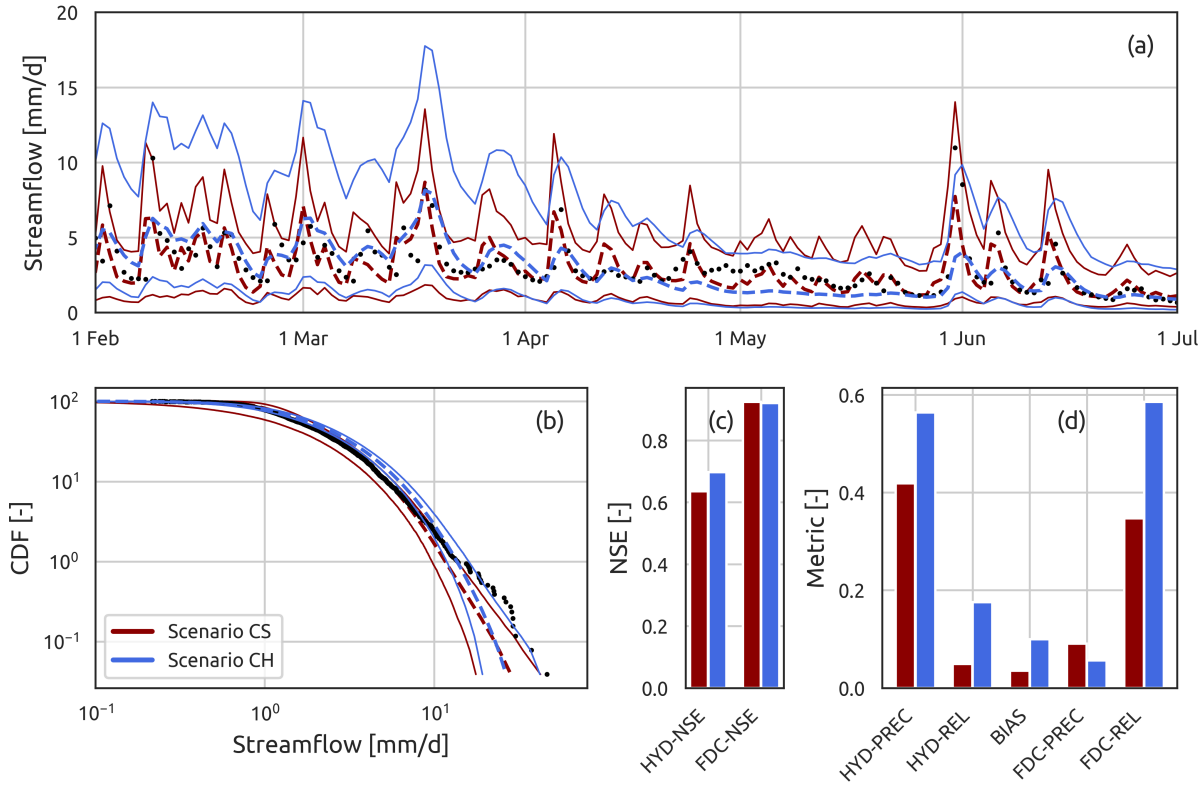

Figure 4.8: Results of experiment 2 for a representative catchment (Andelfingen) and time period (period P3). Panel **(a)** shows the hydrograph, panel **(b)** the annual FDC, panels **(c)** and **(d)** the metrics used for evaluating the simulations. In panels **(a)** and **(b)**, the dashed line represents the median simulation; the solid lines represent 95% uncertainty bands

not a peculiarity of the selected catchment and period but it is quite common, especially in the years when the precipitation is scarce during the late spring and may be due to a misrepresentation of the snow melting process.

Panel (b) shows the simulated FDCs. In terms of precision, the predictive distribution of scenario CH tends to be more precise, especially for low flows; the median of scenario CS, on the other hand, is closer to the data, with the one of scenario CH that tends to overestimate the observed values, particularly in the low flows. Both simulations tend to underestimate high flow values. These results are also confirmed by the metrics (panels c and d), with the precision metric of the FDC that is lower in the scenario CH (0.06 compared to 0.09), the volumetric bias that is higher in the scenario CH (0.10 compared to 0.04), and NS of the FDCs that are similar.

4.5.2.2 *Performance over all catchments*

Figure 4.9 shows the values of the performance metrics for all the catchments and periods considered for this experiment. From these plots, it is difficult to define which scenario produces, overall, the best results. In particular:

- Three metrics (volumetric bias, FDC Nash-Sutcliffe and reliability; panels d, e, and g) show a comparable performance, both in terms of spread and mean value.

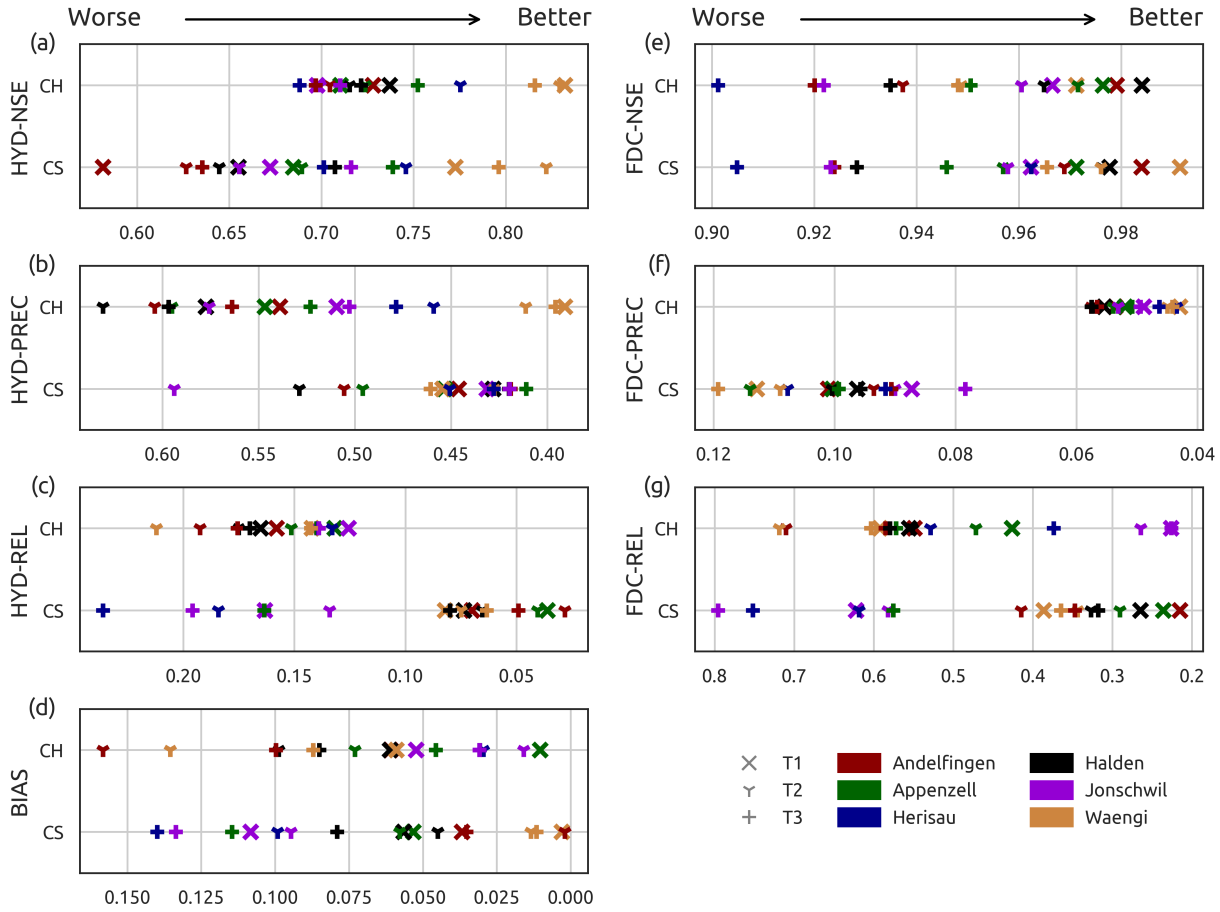

Figure 4.9: Performance metrics achieved by the simulations in experiment 2. The left column reports the metrics calculated on the hydrograph; the right column reports the metrics calculated on the annual FDC. Catchments are distinguished by color, periods are distinguished by symbol. Note that some horizontal axes are reversed so that "better" metric values appear consistently on the right in all plots.

- In two metrics (hydrograph Nash-Sutcliffe and FDC precision; panels a and f) the simulations of scenario CH outperform the ones of scenario CS, showing a better mean value and a lower spread for both metrics.

- In the two remaining metrics (hydrograph precision and reliability; panels b and c) the simulations of scenario CS outperform the ones of scenario CH. In particular, in the hydrograph precision, scenario CS shows better mean and spread; in the hydrograph reliability, scenario CS has a better mean but the presence of some outliers increase broadly the spread.

4.5.3 *Experiment 3: Goodness of the regionalization study*

4.5.3.1 *Representative catchment: hydrograph and FDC representation*

Figure 4.10 compares the results of scenario RS with the prior simulations (scenario PS) in subcatchment Andelfingen, period P3.

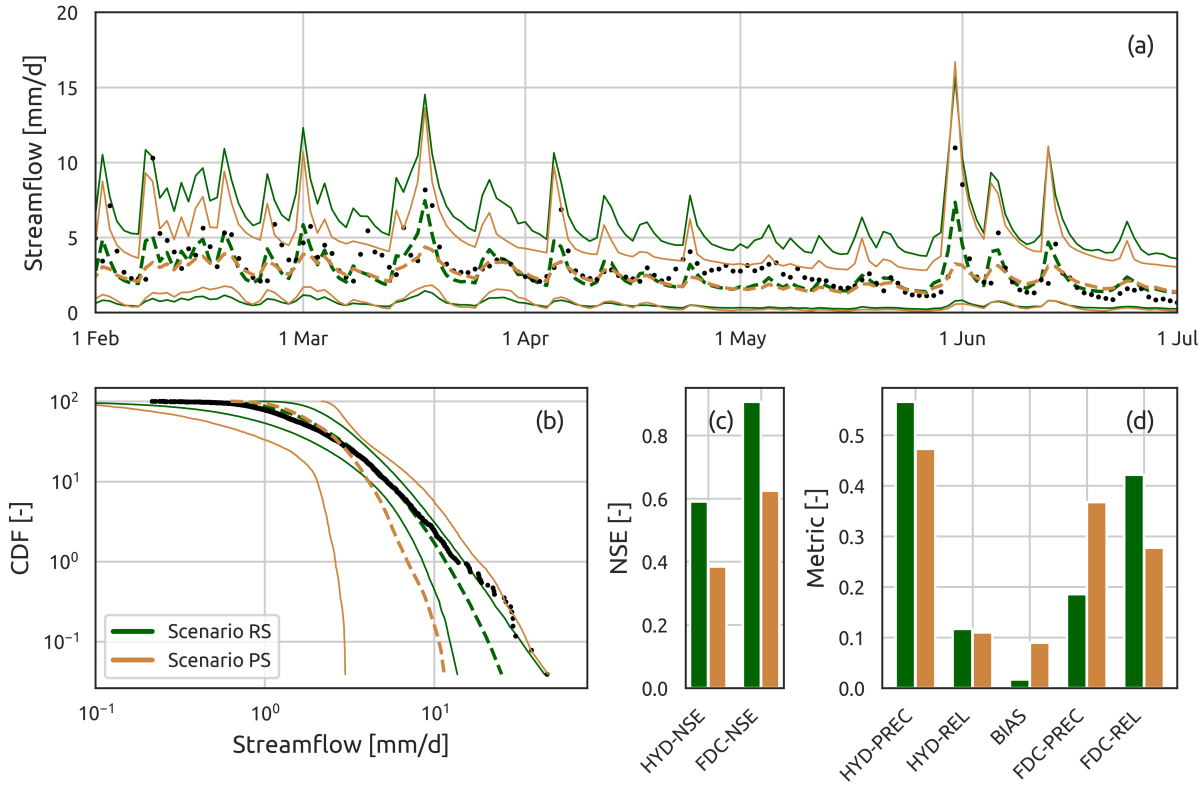

Figure 4.10: Results of experiment 3 for a representative catchment (Andelfingen) and time period (period P3). Panel **(a)** shows the hydrograph, panel **(b)** the annual FDC, panels **(c)** and **(d)** the metrics used for evaluating the simulations. In panels **(a)** and **(b)**, the dashed line represents the median simulation; the solid lines represent 95% uncertainty bands

Panel (a) shows the simulated hydrographs. Although scenario PS has narrower uncertainty bands, we have to notice that, in this scenario, the standard deviation of the residual error model (σ_Ω) has been set to zero and, therefore, the plot of the scenario PS is showing only the parametric uncertainty.

The median hydrograph (dashed line) is more dynamic and closer to the observed data in the scenario RS; this behavior is confirmed by the large difference in Nash-Sutcliffe efficiency (panel c) between the two simulations (0.59 for scenario RS, 0.38 for scenario PS).

The biggest difference between the two scenarios can be observed in panel (b), which shows the simulated FDCs. The predictive distribution of the scenario RS, as already analyzed in section 4.5.1.1, is consistent with the observations, both in terms of uncertainty and fit of the median. The predictive distribution of the scenario PS, on the other hand, has extremely wide uncertainty bounds and a median that is far off the observed values.

4.5.3.2 *Performance over all catchments*

Figure 4.11 presents the performance metrics for the scenarios CH and PS considered in this experiment. Excluding hydrograph precision (panel b) and FDC reliability (panel g), the scenario RS has performance metrics that are always better, or at least similar, to the ones of scenario

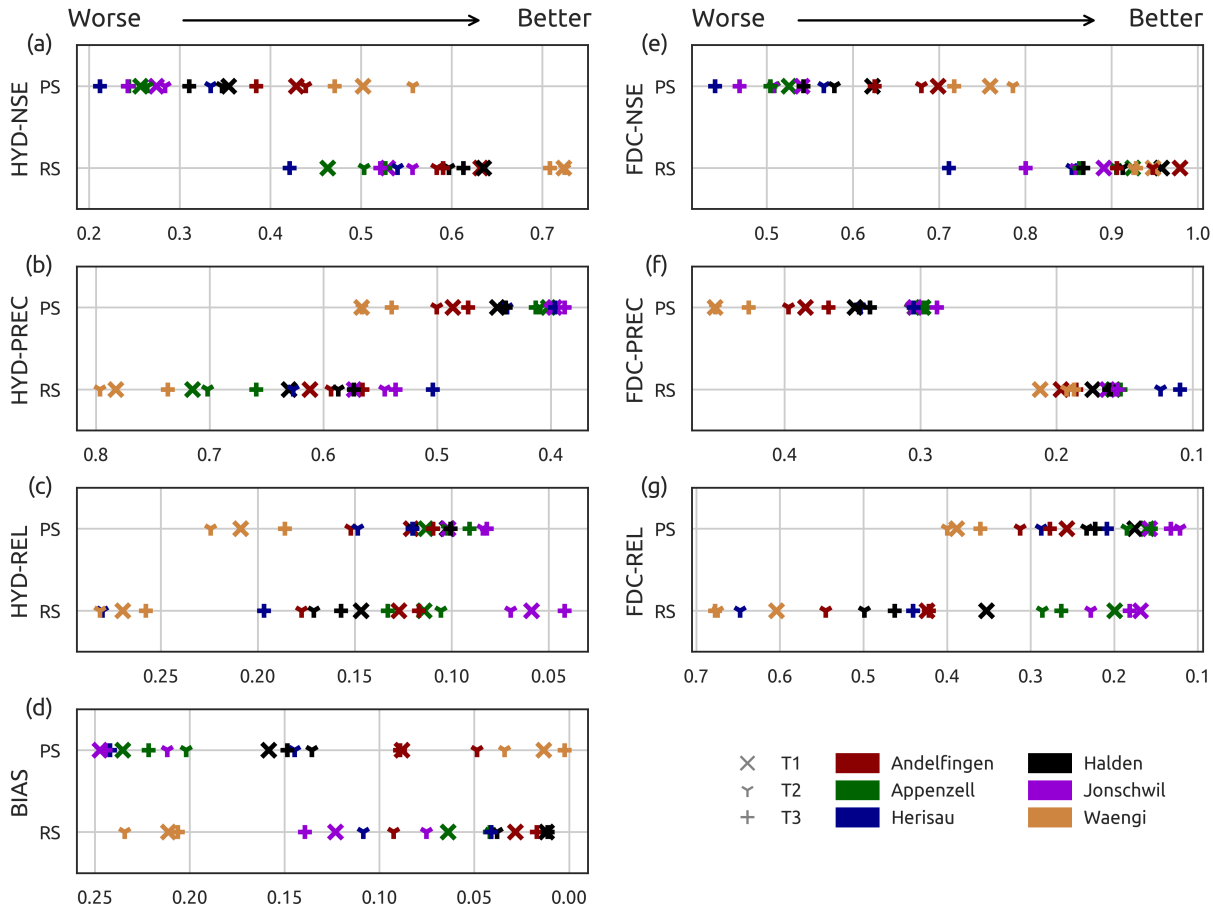

Figure 4.11: Performance metrics achieved by the simulations in experiment 3. The left column reports the metrics calculated on the hydrograph; the right column reports the metrics calculated on the annual FDC. Catchments are distinguished by color, periods are distinguished by symbol. Note that some horizontal axes are reversed so that "better" metric values appear consistently on the right in all plots.

PS. This is particularly evident in the two Nash-Sutcliffe plots (panels a and e) and in the FDC precision (panel f), which demonstrate that the FDC of the scenario RS is much more compact and with a mean value that is closer to the observations. Hydrograph reliability and volumetric bias (panels c and d), on the other hand, show a similar performance for scenarios PS and RS. In particular, hydrograph reliability has the same mean in the two cases, and less spread in scenario PS than in scenario RS; volumetric bias has a lower mean and larger spread for scenario PS than for scenario RS. Panels (b) and (g), finally, show that scenario PS outperforms scenario RS with, in both cases, the first scenario having a higher mean and a lower spread.

4.6 DISCUSSION

4.6.1 *Experiment 1: signature calibration in multiple scenarios*

This experiment compares the results of the three signature-calibration scenarios: concomitant (CS), non-concomitant (NCS), and regionalized (RS). We first discuss aspects associated with model performance, then the impact of errors in the signatures, and, finally, parameters identifiability.

4.6.1.1 *Performance of the model*

The performance obtained in scenario CS is generally higher than the one in scenario NCS, which is, in turn, slightly higher than the one in scenario RS. This relative ranking is particularly evident in the hydrographs (e.g., figure 4.5), which demonstrate this ranking in terms of the accuracy of the median and the uncertainty bounds. The ranking is less clear in the FDCs, where scenario CS still has the lowest uncertainty but the fit of the medians is similar, with the inference in the different scenarios performing better at different flow regimes. Analyzing the performance metrics (figure 4.6), scenario CS is clearly the best scenario (sometimes sharing the first place with others) while scenarios NCS and RS have a similar performance.

These results appear reasonable, as:

- Scenario CS should have the best performance since it uses signatures derived from streamflow observed directly at the catchment and period of interest. Such signatures, therefore, are not affected by the usage of stochastic target signatures, which would increase the total uncertainty, and of possible errors in the estimated signatures, which may be largely different from the observed ones. The models belonging to this scenario are evaluated during the calibration period, making their performance as high as possible.

- Scenario NCS uses signatures that are calculated based on streamflow observations from the same catchment but in a different time-period. This should decrease the performance compared to scenario CS, because transferring the signatures in time requires the use of stochastic target signatures, with the associated increase in uncertainty. This increased uncertainty could be attributable to the signature transfer model not being able to account

for the variability in hydrological conditions in the two periods (the one used for estimating the signatures and the one when the meteorological inputs are available). For example, the presence (or absence) of extreme events or trends in the data would force the models to calibrate to signatures that are not coherent with the forcing data, decreasing the performance of the predicted hydrographs;

- Scenario RS uses signatures that are calculated using a FDCs model and streamflow measurements from neighboring catchments. Similarly to the previous scenario (NCS), this model is subject to uncertainty, which is reflected in the estimation of the signature at the ungauged location. Moreover, we have noted (section 4.4.4.2) the presence of a systematic difference between the estimated and observed seasonal FDCs, which we have been able to correct by adding a bias in the transfer error model.

Only a few studies in the past have compared PUB under different data availability scenarios. Montanari and Toth (2007) compared the results of time-domain calibration, concomitant (scenario CS, in this study) and non-concomitant (scenario NCS) signatures calibration, and prior simulations; while their methodology is different from the one used here, as based on different signatures and inference procedure, their findings are aligned with the ones of this study. In particular, they show an average decrease of Nash-Sutcliffe efficiency of 30% moving from the usage of concomitant to the usage of non-concomitant signatures, and that the magnitude of this decrease is highly dependent on the period considered.

Biondi and De Luca (2016), on the other hand, compared the usage of observed signatures (scenario CS) with regionalized signatures (scenario RS), with the objective of estimating design floods for an assigned return period. Their results show that, for their objective, the use of regionalized signatures does not deteriorate the performance. Our study, while showing a better performance of the simulations of the scenario CS on the ones of the scenario RS, has also shown that this better performance is not consistent for all the evaluation metrics (e.g., volumetric bias or FDC reliability) and that, when present, the difference in performance is not too strong.

4.6.1.2 *Impact of errors in the signatures*

When signatures are estimated, their value can be largely different from the observed one. This difference is due to the uncertainty associated to the signature model, attributable, in the present case, either to the different conditions in the period used to calculate the signatures or to deficiencies in the signature regionalization model.

This study shows the effects of errors associated with the transfer of the signatures in time and with the transfer of the signatures in space:

- *Transfer in time*. Focusing on the performance metrics, scenario NCS shows often the presence of a catchment that, for one period, has a performance that is, by far, worse than all the others. This is the case of the catchment Herisau in period P3 (blue crosses in figure 4.6). Looking closely at this simulation, the poor performance in this catchment may be due to the

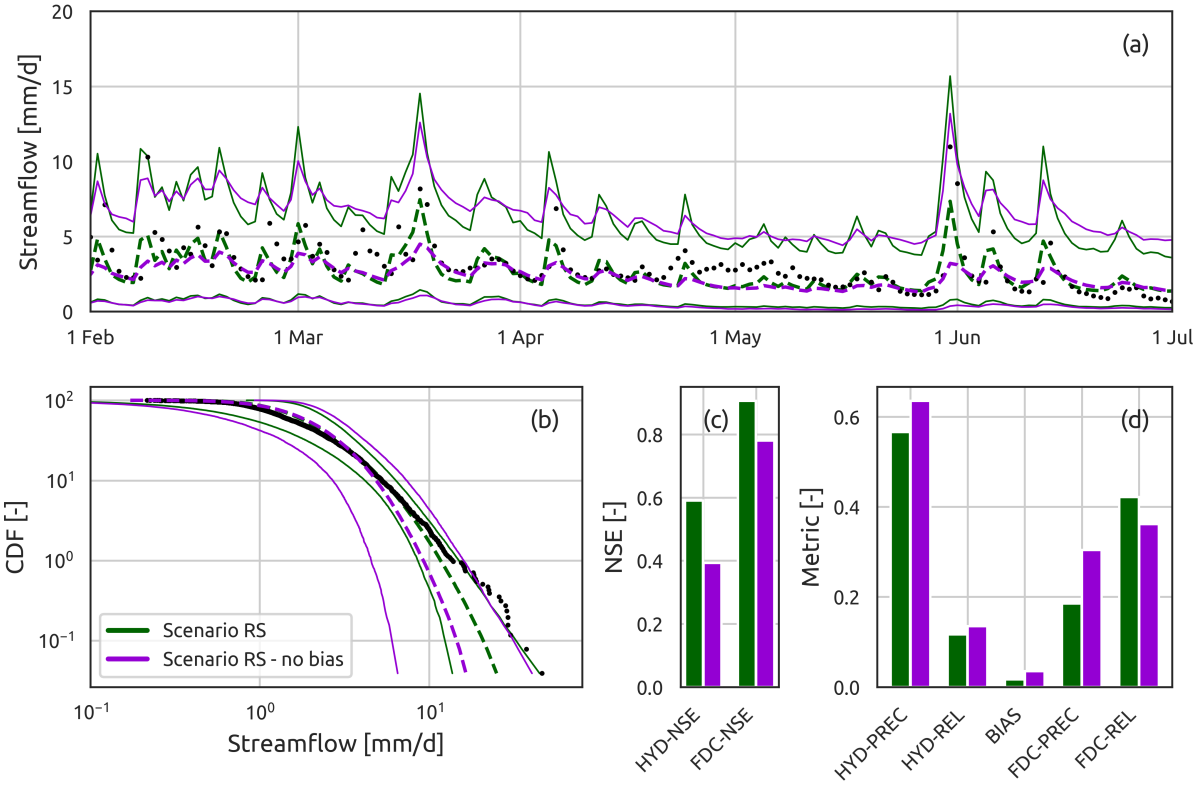

Figure 4.12: Results of the comparison of two regionalization scenarios (with and without bias correction) for a representative catchment (Andelfingen) and time period (period P3). Panel **(a)** shows the hydrograph, panel **(b)** the annual FDC, panels **(c)** and **(d)** the metrics used for evaluating the simulations. In panels **(a)** and **(b)**, the dashed line represents the median simulation; the solid lines represent 95% uncertainty bands

presence of a large flood in 2002: such extreme high flows have a large impact on the FDCs, especially considering that they usually last more than a single day. Therefore, the model is forced with meteorological inputs that generated the flood but calibrated to signatures that do not contain information about this flood, leading to unrealistic parameters values and a mediocre fit.

- *Transfer in space*. The FDCs model has some apparent deficiencies in capturing the observed FDCs, already discussed in section 4.4.4.2. This has led to the decision of correcting the FDCs adding a bias in the error term. Figure 4.12 illustrates the effect of not making this correction, comparing the results of scenario RS with and without bias correction ($a_\epsilon^{(T)} = 1$ in equation 4.18). It can be seen that this new simulation produces, overall, worse results, increasing the uncertainty of hydrograph and FDC, reducing the fit of their medians to observed data, and getting worse in almost all the metrics.

The two examples presented above highlight that a small difference in the estimated signatures could lead to a large difference in the streamflow predictions. This problem has also been noted in previous studies. For example, Westerberg et al. (2011) discussed the role of "disinformative" signatures, Castiglioni et al. (2010) showed the impact of errors in the regionalized signatures due to deficiencies in the regionalization model, and Fenicia et al. (2018) demonstrated the effect of time shifts in the time series that are not captured by the signatures. All these studies pointed out that the presence of small errors in the signatures may lead to large differences in the model output, as shown in this study.

4.6.1.3 *Parameters identifiability*

The calibration should identify the values of the parameters that maximize the objective function. However, in some cases, this is not possible when the effect of a parameter on the model output is not measurable by the objective function. This is shown in figure 4.7, which illustrates how some parameters lose identifiability when calibrating to estimated signatures. This behavior depends largely on the parameter considered, and can be interpreted for specific cases:

- *Parameters that remain identifiable in all the scenarios*. These are parameters that have a strong influence on the model output. This is the case of D, which controls the splitting of the flow between the “fast” and the “slow” part of the model. A different value of this parameter would strongly affect the FDCs in all scenarios, since the slope of the FDC can be related to the proportion of quick and slow response in the catchment.

- *Parameters that lose their identifiability when moving to an ungauged scenario*. These are parameters whose influence on the signatures may be blurred by higher uncertainty in the target signatures. This is the case, for example, of k_{WR}, which controls the outflow of the snow reservoir. The transfer error may contribute in obscuring this behavior, but the regionalization model can also be responsible, not representing correctly snow dynamics;

this is the case of the FDCs model, which does not distinguish explicitly the separation between rainfall and snowfall.

- *Parameters that are weakly identifiable, regardless the scenario*. These are parameters that have an effect on the model output that is not captured by the signatures. For example, consider parameter k_{SR}, which controls the release rate of the "slow" part of the model. This parameter has a strong effect on the hydrograph, by affecting the baseflow component, but its effect on the FDCs may be less evident. This change in behavior compromises its identifiability when the model is calibrated to FDCs. A different choice of signatures (e.g., including the baseflow index) would have probably helped in this specific case, but this analysis is not in the scope of this study.

Similar results regarding parameters identifiability have been highlighted by Biondi and De Luca (2016), who showed that using observed signatures makes the posterior distributions sharper, compared to the usage of modelled signatures.

4.6.2 *Experiment 2: comparison with time-domain calibration*

The aim of experiment 2 is to assess the overall performance of signature calibration, in its best scenario (CS), using the classical time-domain calibration as a benchmark.

In our cases study, scenarios CS and CH produce similar results in terms of fit to the observed data, uncertainty, and performance metrics. This finding indicates that seasonal FDCs can represent a suitable choice of signatures since when used for the inference they have summarized the information contained in the hydrograph with a relatively limited loss.

Our findings are in line with previous studies (e.g., Castiglioni et al., 2010; Kavetski et al., 2018), which have shown that signatures calibration can achieve similar or slightly inferior performance to time-domain calibration and that, in some cases, the FDCs are a viable choice for this. On the other hand, Kim et al. (2017) pointed out how the lack of time information if the FDC makes them, in some cases, a poor choice for signatures calibration, suggesting to add other signatures that account for that information (e.g. flashiness index).

The study of loss of information due to the usage of different signatures is, however, a broad and important topic that goes beyond the scope of this study; this topic has been partially covered in other studies (e.g., Fenicia et al., 2018) though future work is still required.

4.6.3 *Experiment 3: Goodness of the regionalization study*

The last experiment consisted in comparing the scenario RS with simulations run sampling the parameters from the prior distribution. The objective was to assess the performance of the regionalization approach, comparing it with an easy benchmark (easy in the sense that the effort

and the data needed to run prior simulations are limited and can be done in any ungauged catchment as long as the meteorological forcing is available).

The results show that the simulations of the scenario RS generally outperform the ones of the scenario PS. On the hydrograph, the two scenarios have similar performance, with scenario RS that fits better the observed data and scenario PS that is more precise. On the FDC, however, the performance of scenario PS drops substantially with the simulated FDC that looks unrealistic, with huge uncertainty and no fit to the observed data.

Hence, similar uncertainty bands in the hydrograph space can correspond to very different uncertainty bands in the signatures space. This result indicates that scenario PS produces hydrographs with very different dynamics, whereas scenario RS produces hydrographs with much more similar dynamics. The cause of this difference in behavior is that in scenario PS the uncertainty is purely parametric, and the uncertainty bands in the hydrograph space are generated by deterministic model predictions, whereas in scenario RS it is also contributed by the residual error, and the uncertainty bands in the hydrograph space are generated by stochastic model predictions. Such difference between scenarios PS and RS is not apparent when considering the total uncertainty bands in the hydrograph space. However, it becomes obvious in the signature space, where the similarities or differences between individual hydrographs manifest themselves as similarity or differences in corresponding signatures. Thus, the FDCs corresponding to scenario PS have much wider uncertainty bands than the FDCs corresponding to scenario RS.

The difference in behavior between scenarios PS and RS is apparent when considering two metrics where scenario PS has better performance than scenario RS: hydrograph precision and FDC reliability. The first result can be explained considering that in scenario PS the variance of the residual error model ($\sigma_{\Omega}^{(H)}$) is set to zero and, therefore, only the parametric uncertainty is considered. This choice was motivated by the fact that sampling $\sigma_{\Omega}^{(H)}$ from its prior would have resulted in unrealistic (extremely high) uncertainties for most of the simulations. The better performance of scenario PS in terms of FDC reliability is because the observed data cross the simulated uncertainty bound and this gives them almost a uniform distribution, making the reliability metric low.

4.6.4 *Limitations and future work*

This study has a number of limitations that warrant follow-up research. First, the case study catchments in this work all belong to the same geographic region. For this reason, they can be thought to be similar, or at least less varied than catchments in different climatic regions. Nevertheless, the chosen Thur catchments show a relatively large variability in physical and local hydroclimate characteristics, which in turn manifest as appreciable differences in mean streamflow, in seasonality, and in baseflow (refer to Dal Molin et al., 2020, for further details). Future studies could include a wider range of catchments, in particular, dry catchments.

Second, the case study catchments are in a well-studied area, where previous work could be exploited for the selection of the streamflow model (Dal Molin et al., 2020) and the FDCs model (Doulatyari et al., 2017). The choice of neighboring catchments also gives an opportunity, in scenario RS, to assume a similar behavior of the regionalization model in the subcatchments and, therefore, to estimate the parameters of the signatures error model using other subcatchments of the network. The application of the proposed methodology in less instrumented catchments may be more challenging and deserves future attention.

Finally, the FDCs model does not account for snow dynamics. In other words, when precipitation is used in the signature model for calculating its parameters (refer to section 4.4.4.2), no distinction is made between rainfall and snowfall. This limitation does not appear problematic, given the limited amount of snow in the case study (at most 20% of the annual precipitation volume). Importantly, note that the models calibrated in snow affected catchments (e.g., Appenzell) do not perform systematically worse than the ones of catchments with little snow (e.g., Wängi). Nevertheless, future work should move in the direction of signatures and regionalization models that incorporate snow dynamics.

4.7 CONCLUSION

This work presented a new approach for calibrating a rainfall-runoff model to regionalized streamflow signatures. The approach is able to propagate the uncertainties in the signatures estimation to the streamflow predictions. The approach relies on Approximate Bayesian Computation (ABC), which spares from the direct calculation of the likelihood function. The main intended application of the new method is for prediction in ungauged catchments, where current methods are limited by not explicitly accounting for the uncertainty in estimated signatures on model predictions.

The proposed framework is evaluated in a case study based on six subcatchments of the Thur catchment (Switzerland). This catchment has high quality data, a long history of hydrological studies, and high operational importance in the water supply of the region. In order to test the quality of predictions in ungauged conditions, we followed a progression from gauged to ungauged, with three scenarios: "concomitant", where signatures are observed, "non-concomitant", where signatures are transferred in time, and "regionalized", where signatures are transferred in space. For reference, we also calibrated the model in the time domain and run simulations using parameters sampled from the prior distribution.

The results of the experiments suggest that:

1. In line with expectations, the progression from concomitant to regionalized signatures increases uncertainties in reproducing time series and flow duration curves. However the change is generally small, e.g. a limited loss of precision (38% more uncertainty) and fit to the observed data (20% decrease in NSE). These findings provide confidence that the proposed approach has general utility;

2. Poor quality of model predictions could be attributed to cases where regionalized signatures differ strongly from the observed ones. In such cases, the error in estimating the signatures results into streamflow predictions with poor performance metrics (e.g. doubling the volumetric bias);

3. The use of regionalized signatures may reduce the identifiability of some parameters of the streamflow model (e.g. the release rate of the snow reservoir). This behavior is likely explained by the representation of some processes (e.g. snow) being lost when calibrating to regionalized rather than observed signatures;

4. Signatures based on seasonal FDCs appear to be an adequate choice, as they result into streamflow predictions that are comparable to those obtained in time-domain calibration. However this finding may be catchment-specific;

5. There is a clear improvement moving from prior simulations to calibration to regionalized signatures, which shows that regionalized signatures contain useful information to constrain model predictions.

The proposed methodology represents a step towards better predictions in data scarce regions. Future work is needed to generalize the approach to more diverse areas, especially where neighboring catchments are not available for the calibration of the signature regionalization model, to focus on the impact of the signatures selected on the results, and to understand how this methodology compares to other approaches for prediction in ungauged basins.

APPENDICES

4.A CHOICE OF THE DISTANCE MEASURE BETWEEN MODELLED AND TARGET SIGNATURES

The choice of distance metric ρ to measure the distance between the target and the simulated signatures is intended to give the same "weight" to all the signatures. Therefore, a normalization factor, which acts as a weight, is used to balance distances between signatures with different ranges of variability.

The normalization term $\rho_{norm}^{(j)}$ for each signature is calculated as the empirical standard deviation of the absolute difference between simulated and target signatures

$$\rho_{norm}^{(j)} = \text{std}\left(\left|\mathbf{y}^{(j)} - \tilde{\mathbf{y}}^{(j)}\right|\right) \tag{4.19}$$

where the index j represents the specific signature (i.e. the slope between two percentiles of one of the four FDCs) and std represent the standard deviation function.

Since the signatures can be stochastic, the normalization term in equation 4.19 is calculated as follows: (i) generate N sets of simulated signatures $\mathbf{y}^{(j)}$ from the hydrological model, sampling the parameters $\theta^{(H)}$ from their prior distribution, and (ii) sample N sets of target signatures $\tilde{\mathbf{y}}^{(j)}$ from the signature model. The value of N is set to 5000.

The term $\rho_{norm}^{(j)}$ should represent a reliable estimate of the variability of each signature. Therefore, it can be used to normalize the distance metric in order to give the same importance to each signature. The distance metric, used in the ABC algorithm, is defined as the maximum value, over the four seasons, of the mean distance between target and simulated FDCs, as expressed by the equation

$$\rho(\mathbf{y}_i, \tilde{\mathbf{y}}_i) = \max_{\text{season}}\left[\text{mean}_{\text{FDC}}\left(\frac{\left|\mathbf{y}_i^{(j)} - \tilde{\mathbf{y}}_i^{(j)}\right|}{\rho_{norm}^{(j)}}\right)\right] \tag{4.20}$$

This normalization of the distance metric is different form the normalization used in previous studies, where $\rho_{norm}^{(j)}$ was simply set equal to the target signatures $\tilde{\mathbf{y}}^{(j)}$ (e.g., Fenicia et al., 2018). That earlier normalization approach would not have worked in this study because the target signatures may be stochastic, and in some cases take the value of 0, in which case $\rho \to \infty$.

4.B CONSTRUCTION OF VIRTUAL UNGAUGED SCENARIOS FOR THE CALIBRATION OF THE SIGNATURE MODELS

In the scenarios NCS and RS, the target signatures are generated using the probabilistic model defined in sections 4.2.3.2 and 4.2.3.3.

The inference of the parameters $\theta^{(T)}$ is done before inferring the parameters $\theta^{(H)}$ and using observations that are independent (meaning that they come from a different time period or from

different catchments, depending on the scenario) from the ones used to infer the parameters of the streamflow model.

In order to construct "virtual" ungauged scenarios, the 24 years-long period of streamflow data is divided into three periods: P1, from 1 September 1981 to 31 August 1989; P2, from 1 September 1989 to 31 August 1997; P3, from 1 September 1997 to 31 August 2005.

The following procedure is then applied for the two scenarios:

- scenario NCS: when $\theta^{(H)}$ are needed in P1 for a given catchment, the signatures transfer model in equation 4.3 uses inputs $\tilde{\mathbf{q}}_{tr}$ from P2, and is conditioned on signatures $\tilde{\mathbf{y}}$ observed in P3. We then rotate the three periods in order to run the inference for all the time series of a given catchment. We then apply this procedure to all catchments.

- scenario RS: when $\theta^{(H)}$ are needed in a specific catchment for a given period, the signature model in equation 4.4 is conditioned on signatures $\tilde{\mathbf{y}}$ observed in the other five catchments. We then rotated the six catchments in order to run the inference in all catchments for a given period. We then applied this procedure to all periods.

This procedure generates a set of calibrated parameters $\theta^{(T)}_{opt}$ that is specific for each catchment and each time period. These parameters are kept fixed at their calibrated values when inferring $\theta^{(H)}$ with the procedure described in section 4.3.

The underlying assumptions of this approach are: (i) the behavior of each catchment does not change substantially moving from one period to another, and (ii) the discrepancies between the signatures generation model (Doulatyari et al., 2017) and the observed signatures is similar among the catchments.

4.C HYDROLOGICAL MODEL DETAILS

This appendix lists the hydrological model equations (section 4.C.1) and the prior parameter distribution (section 4.C.2).

4.C.1 *Model equations*

4.C.1.1 *Snow reservoir (WR)*

The mass balance equation is

$$\frac{dS_{WR}}{dt} = P_{WR} - Q_{WR} \tag{4.21}$$

with fluxes

$$P_{WR} = \begin{cases} P & \text{if } T \leqslant 0 \\ 0 & \text{if } T > 0 \end{cases} \tag{4.22}$$

$$M_{WR}^{max} = \begin{cases} 0 & \text{if } T \leqslant 0 \\ k_{WR} T & \text{if } T > 0 \end{cases} \tag{4.23}$$

$$Q_{WR} = M_{WR}^{max} \left[1 - \exp\left(-\frac{S_{WR}}{m_{WR}} \right) \right] \tag{4.24}$$

m_{WR} is fixed to 2.

4.C.1.2 *Unsaturated reservoir (UR)*

The mass balance equation is

$$\frac{dS_{UR}}{dt} = P_{UR} - Q_{UR} - E_{UR} \tag{4.25}$$

with fluxes

$$P_{UR} = P - P_{WR} + Q_{WR} \tag{4.26}$$

$$\overline{S_{UR}} = \frac{S_{UR}}{S_{UR}^{max}} \tag{4.27}$$

$$E_{UR} = (PET) \left[\frac{\overline{S_{UR}} (1 + m_{UR})}{1 + m_{UR}} \right] \tag{4.28}$$

$$Q_{UR} = P_{UR} \left(\overline{S_{UR}} \right)^{\beta_{UR}} \tag{4.29}$$

m_{UR} is fixed to 0.01.

4.C.1.3 *Fast reservoirs (FR)*

The mass balance equation is

$$\frac{dS_{FR}}{dt} = P_{FR} - Q_{FR} \tag{4.30}$$

with fluxes

$$P_{FR} = (1 - D)\, Q_{UR} \tag{4.31}$$

$$Q_{FR} = k_{FR} S_{FR} \tag{4.32}$$

4.C.1.4 *Slow reservoir (SR)*

The mass balance equation is

$$\frac{dS_{SR}}{dt} = P_{SR} - Q_{SR} \tag{4.33}$$

with fluxes

$$P_{SR} = D Q_{UR} \tag{4.34}$$

$$Q_{SR} = k_{SR} S_{SR} \tag{4.35}$$

Table 4.1: Details of parameters of the hydrological model

Parameter	Unit	Model component	Bounds	Logarithmic transformation applied
k_{WR}	d^{-1}	WR	$10^{-1} - 10$	Yes
S_{UR}^{max}	mm	UR	$10^{-1} - 10^{3}$	Yes
β_{UR}	–	UR	$10^{-3} - 8$	Yes
D	–	Structure	$0 - 1$	No
k_{FR}	d^{-1}	FR	$10^{-1} - 10$	Yes
k_{SR}	d^{-1}	SR	$10^{-4} - 10^{-1}$	Yes

4.C.2 *Prior distribution of parameters*

Table 4.1 lists the calibrated parameters and the limits of their uniform prior distributions. For parameters that are log-transformed, the uniform prior distribution is applied to the transformed value.

4.D PERFORMANCE METRICS

This appendix defines the model performance metrics listed in section 4.4.5. Symbol $\mathbf{x}$ refers to simulated data and symbol $\tilde{\mathbf{x}}$ refers to observed data. Note that $\mathbf{x}$ is a matrix, containing ensemble of model predictions generated following the procedure in appendix 4.E. In contrast, $\tilde{\mathbf{x}}$ is a vector (e.g., representing a time series). In both cases, the index i indicates the time step.

4.D.1 *Nash-Sutcliffe efficiency*

The Nash-Sutcliffe efficiency measures the fit of the simulated data to the observations.

$$NS(\mathbf{x}, \tilde{\mathbf{x}}) = 1 - \frac{\sum_{i=1}^{N} \left(\text{med}\left(\mathbf{x}^i\right) - \tilde{\mathbf{x}}^i\right)^2}{\sum_{i=1}^{N} \left(\tilde{\mathbf{x}}^i - \text{mean}\left(\tilde{\mathbf{x}}\right)\right)^2} \tag{4.36}$$

where $\text{med}\left(\mathbf{x}^i\right)$ is the median of the predictive distribution and $\text{mean}\left(\tilde{\mathbf{x}}\right)$ is the mean of the observed streamflow time series.

The NS varies between $-\infty$ and 1, with 1 denoting a perfect match.

4.D.2 *Precision*

The precision metric quantifies the spread of the simulated time series.

$$P(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{\sum_{i=1}^{N} \text{sdev}\left(\mathbf{x}^i\right)}{\sum_{i=1}^{N} \tilde{\mathbf{x}}^i} \tag{4.37}$$

where sdev $\mathbf{x}^i$ indicates the standard deviation of simulated data.

Not that the value of the metric does not depend on the fit between modelled and observed data. The "best" value of this metric is 0, which occurs for a prediction with no uncertainty, i.e., where all predictive replicates fall on a single curve. Higher values denote worse precision.

4.D.3 *Reliability*

The reliability (R) measures the consistency of the observations with the predictive distribution. The metric is defined as the area between the diagonal and the predictive quantile-quantile curve (PQQ plots). The mathematical derivation can be found in McInerney et al. (2017). The best value of this metric is 0. Higher values denote worse reliability.

4.D.4 *Volumetric bias*

The volumetric bias measures the long-term water balance error of the predicted data.

$$VB(\mathbf{x}, \tilde{\mathbf{x}}) = \left| \frac{\sum_{i=1}^{N} \text{mean}\left(\mathbf{x}^i\right) - \sum_{i=1}^{N} \tilde{\mathbf{x}}^i}{\sum_{i=1}^{N} \tilde{\mathbf{x}}^i} \right| \tag{4.38}$$

where mean $\left(\mathbf{x}^i\right)$ is the mean of the predictive distribution over a time step. The best value of this metric is 0 and higher values denote an increasing water balance discrepancy.

4.E STREAMFLOW PREDICTIVE DISTRIBUTION

The posterior and prior predictive distributions of the streamflow time series are obtained by sampling from the probabilistic model with given parameter sets $\theta^{(H)}$.

A replicate streamflow time series is generated as follows:

1. Take parameter set $\theta_i^{(H)}$
2. Run the deterministic hydrological model $\mathbf{m}^{(H)}\left(\theta_{m,i}^{(H)}, \mathbf{x}\right)$ to generate a streamflow time series;
3. Sample a time series of residual errors from the residual error model $\boldsymbol{\epsilon}^{(H)}\left(\theta_{\epsilon,i}^{(H)}\right)$;

4. Use the results of the previous two points in equation 4.1 to compute a realization of the hydrological model $\mathbf{Q}^{(H)}$

5. Repeat for $i = 1, ..., N_{sam}$

5

CONCLUSIONS

The development of tools and methodologies to quantify regional streamflow variability is fundamental to improve the understanding of hydrological dynamics at the catchment scale. This thesis has presented three different works that contributed with tools and methodologies to help scientists and practitioners understand and simulate regional streamflow variability across catchments.

Chapter 2 has introduced SuperflexPy, a new flexible modelling framework for building conceptual hydrological models. SuperflexPy is a powerful software tool that enables modelers to build customized models, encompassing a wide range of applications and maintaining the possibility for further extensions. This study has provided an extensive overview of existing modelling frameworks and has highlighted the contribution of SuperflexPy in addressing current limitations. The development process of SuperflexPy has unveiled the inevitable compromises that need to be taken in order to create a feasible software implementation that satisfies the theoretical requirements.

Chapter 3 has presented a methodology to guide the construction of semi-distributed hydrological models that reflect the dominant processes controlling streamflow response. In particular, the methodology avoids making a priori assumptions when building a model, basing all the choices on the interpretation of hydrological data. This methodology becomes particularly important when the objective is to create parsimonious models that do not try to simulate every single aspect of the hydrological cycle but focus on only the relevant processes. The effectiveness of the procedure is demonstrated with a modelling application on the Thur catchment, where it provides valuable insights on the hydrological processes controlling streamflow regional variability. Specific results on the Thur catchment, which are in agreement with previous research, have highlighted the importance of including the variability of meteorological inputs and geological features in the model formulation; other case studies may indicate different dominant processes but, nevertheless, the significance of the work remains in the proposed methodology rather than in the catchment-specific findings.

Chapter 4, finally, has explored the calibration of hydrological models to estimated streamflow signatures, which enables streamflow predictions with uncertainty quantification in ungauged catchments. The parameters of conceptual models, in fact, cannot be linked to measurable quantities in the field and, therefore, must be inferred through calibration to observed streamflow.

Calibration to hydrological signatures, as an alternative to calibration to time series, has been shown to be a promising approach since signatures and the associated uncertainty can be estimated even in ungauged catchments. However, previous studies did not focus directly on hydrograph uncertainty quantification. The proposed procedure takes advantage of our understanding of streamflow variability to estimate the signatures and uses advanced numerical techniques to calibrate the hydrological model and quantify the uncertainty of the streamflow predictions. The results in the Thur catchment show satisfactory predictions in different data availability scenarios, with the performance of the models calibrated on regionalized signatures that does not degrade substantially compared to classical time-domain calibration.

Thus, chapter 3 and chapter 4 both deal with the problem of predicting streamflow across catchments, but this objective is achieved in different ways. In chapter 3, we regionalize the processes and build a model that captures processes variability. In chapter 4, we extrapolate the streamflow signatures and calibrate a model to the estimated signatures. Both approaches represent viable solutions to the problem of simulating streamflow regional variability, and more generally, to the challenge of making predictions in ungauged catchments. Apart from their methodological contribution, both works have discovered valuable insights on the hydrology of the Thur catchment, in particular underlining the importance of the geological formulations in controlling the variability of streamflow signatures related to baseflow dynamics. Chapter 2, finally, provides a methodological contribution to both approaches by providing a platform that facilitates the model building process. The full potential of this flexible framework is exploited in chapter 3, which deals with a distributed model application, whereas chapter 4 uses a lumped conceptual model.

It is important to underline that the progress of hydrological modelling goes along with the development of more advanced and integrated codes that facilitate the construction and usage of numerical models. Furthermore, the software used in science should be open source and widely accessible; its development should be the outcome of the joint effort of small groups of core developers together with the broader contribution of the community of scientists and modelers.

Besides the process-oriented insights of the dedicated applications in the Thur catchment, an important contribution of this thesis is represented by the software that has been produced and made available. In particular, this thesis contributed SuperflexPy (chapter 2), which is a general modelling platform to construct tailor-made hydrological models, and aided the further development of ABCpy (Dutta et al., 2017) to perform inference of the model parameters to estimated signatures, as detailed in section 4.3. Additional code is currently being processed and will be made available in the near future. Thus, the works in this thesis have embraced the ideals to both promote the realization of new open-source code (i.e., SuperflexPy) and customizing existing solutions to pursue the research objectives.

5.1 PERSPECTIVES

This thesis has been particularly oriented to the development of new tools and methods to advance research in hydrology. The hope is that these can become the starting point for future research.

As explained in section 2.5.3, the development of SuperflexPy is left open to further contributions from the community that can push it towards new directions. Examples of developments are the creation of new model components that reflect new ideas on streamflow processes, the realization of new couplings between model elements (e.g., reservoirs) and spatial components (e.g., HRUs), the extension of the framework beyond the simulation of water fluxes, the experimentation of new spatial configurations, etc.

SuperflexPy has been designed to be easy to interface with other Python inference frameworks. Such tools can interact deeply with the internal functioning of the hydrological model, making it possible to explore research directions such as usage of internal fluxes and states for calibration, simulation with time-variable parameters, alteration of the internal states during the simulation, etc.

In terms of model development approaches, the guidelines proposed to create a semi-distributed model that reflects catchment dynamics can become a blueprint for the creation of robust models that are more resilient to changes in conditions and that can generalize well to ungauged locations. The algorithm developed for calibration to estimated signatures can be tested in different conditions (e.g., different choices of signatures or catchments) to understand better its applicability to provide streamflow predictions and uncertainty quantification in ungauged locations in different regions.

Given the focus of the thesis on the development of methods, we have limited their application and testing to the Thur catchment. This catchment has been chosen because of the large availability of hydrological data, its low level of human impact on the natural streamflow regime, the presence of variegate characteristics within a relatively small area, as well as the available hydrological expertise in this area. However, the emergence of large datasets of catchments (e.g., the CAMELS series of national datasets), which were less available and popular when this work started, represents a big opportunity to test these methodologies to more diversified conditions. This could lead to new insights regarding drivers of streamflow variability and catchment functioning.

BIBLIOGRAPHY

Abbaspour, K. C., J. Yang, I. Maximov, R. Siber, K. Bogner, J. Mieleitner, J. Zobrist, and R. Srinivasan (2007), "Modelling hydrology and water quality in the pre-alpine/alpine thur watershed using swat." *Journal of Hydrology*, 333, 413–430, URL https://www.doi.org/10.1016/j.jhydrol.2006.09.014.

Abbott, M. B. and J. C. Refsgaard (2012), *Distributed hydrological modelling*, volume 22. Springer Science & Business Media.

Abdulla, F. A. and D. P. Lettenmaier (1997), "Development of regional parameter estimation equations for a macroscale hydrologic model." *Journal of Hydrology*, 197, 230–257, URL https://www.doi.org/10.1016/S0022-1694(96)03262-3.

Addor, N., G. Nearing, C. Prieto, A. J. Newman, N. Le Vine, and M. P. Clark (2018), "A ranking of hydrological signatures based on their predictability in space." *Water Resources Research*, 54, 8792–8812, URL https://www.doi.org/10.1029/2018wr022606.

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

Ajami, K. N., H. Gupta, T. Wagener, and S. Sorooshian (2004), "Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system." *Journal of Hydrology*, 298, 112–135, URL https://www.doi.org/10.1016/j.jhydrol.2004.03.033.

Albert, C., H. R. Künsch, and A. Scheidegger (2015), "A simulated annealing approach to approximate bayes computations." *Statistics and Computing*, 25, 1217–1232, URL https://www.doi.org/10.1007/s11222-014-9507-8.

Almeida, S., N. Le Vine, N. McIntyre, T. Wagener, and W. Buytaert (2016), "Accounting for dependencies in regionalized signatures for predictions in ungauged catchments." *Hydrology and Earth System Sciences*, 20, 887–901, URL https://www.doi.org/10.5194/hess-20-887-2016.

Ammann, L., T. Doppler, C. Stamm, P. Reichert, and F. Fenicia (2020), "Characterizing fast herbicide transport in a small agricultural catchment with conceptual models." *Journal of Hydrology*, 586, 124812, URL https://www.doi.org/10.1016/j.jhydrol.2020.124812.

Andréassian, V., C. Perrin, L. Berthet, N. Le Moine, J. Lerat, C. Loumagne, L. Oudin, T. Mathevet, M. H. Ramos, and A. Valéry (2009), "Hess opinions "crash tests for a standardized evaluation

of hydrological models"." *Hydrology and Earth System Sciences*, 13, 1757–1764, URL https://www.doi.org/10.5194/hess-13-1757-2009.

Antonetti, M., R. Buss, S. Scherrer, M. Margreth, and M. Zappa (2016), "Mapping dominant runoff processes: an evaluation of different approaches using similarity measures and synthetic runoff simulations." *Hydrology and Earth System Sciences*, 20, 2929–2945, URL https://www.doi.org/10.5194/hess-20-2929-2016.

Antonetti, M. and M. Zappa (2018), "How can expert knowledge increase the realism of conceptual hydrological models? a case study based on the concept of dominant runoff process in the swiss pre-alps." *Hydrology and Earth System Sciences*, 22, 4425–4447, URL https://www.doi.org/10.5194/hess-22-4425-2018.

Arnold, J. G., D. N. Moriasi, P. W. Gassman, K. C. Abbaspour, M. J. White, R. Srinivasan, C. Santhi, R. D. Harmel, A. van Griensven, M. W. Van Liew, N. Kannan, and M. K. Jha (2012), "Swat: Model use, calibration, and validation." *Transactions of the ASABE*, 55, 1491–1508, URL https://www.doi.org/10.13031/2013.42256.

Arnold, J. G., R. Srinivasan, R. S. Muttiah, and J. R. Williams (1998), "Large area hydrologic modeling and assessment. part i: model development." *JAWRA Journal of the American Water Resources Association*, 34, 73–89, URL https://www.doi.org/10.1111/j.1752-1688.1998.tb05961.x.

Arthington, A. H., R. J. Naiman, M. E. McClain, and C. Nilson (2010), "Preserving the biodiversity and ecological services of rivers: new challenges and research opportunities." *Freshwater Biology*, 55, 1–16, URL https://www.doi.org/10.1111/j.1365-2427.2009.02340.x.

Bales, R. C. (2015), "Hydrology, floods and droughts | overview." In *Encyclopedia of Atmospheric Sciences (Second Edition)* (G. R. North, J. Pyle, and F. Zhang, eds.), second edition edition, 180–184, Academic Press, Oxford, URL https://www.doi.org/10.1016/B978-0-12-382225-3.00166-3.

Bancheri, M., F. Serafin, and R. Rigon (2019), "The representation of hydrological dynamical systems using extended petri nets (epn)." *Water Resources Research*, 55, 8895–8921, URL https://www.doi.org/10.1029/2019WR025099.

Berger, K. P. and D. Entekhabi (2001), "Basin hydrologic response relations to distributed physiographic descriptors and climate." *Journal of Hydrology*, 247, 169–182, URL https://www.doi.org/10.1016/S0022-1694(01)00383-3.

Bergion, V., E. Sokolova, J. Åström, A. Lindhe, K. Sörén, and L. Rosén (2017), "Hydrological modelling in a drinking water catchment area as a means of evaluating pathogen risk reduction." *Journal of Hydrology*, 544, 74–85, URL https://www.doi.org/10.1016/j.jhydrol.2016.11.011.

Bertuzzo, E., M. Thomet, G. Botter, and A. Rinaldo (2013), "Catchment-scale herbicides transport: Theory and application." *Advances in Water Resources*, 52, 232–242, URL https://www.doi.org/10.1016/j.advwatres.2012.11.007.

Betterle, A., M. Schirmer, and G. Botter (2017), "Characterizing the spatial correlation of daily streamflows." *Water Resources Research*, 53, 1646–1663, URL https://www.doi.org/10.1002/2016wr019195.

Beven, K. (1989), "Changing ideas in hydrology — the case of physically-based models." *Journal of Hydrology*, 105, 157–172, URL https://www.doi.org/10.1016/0022-1694(89)90101-7.

Beven, K. (2006a), "A manifesto for the equifinality thesis." *Journal of Hydrology*, 320, 18–36, URL https://www.doi.org/10.1016/j.jhydrol.2005.07.007.

Beven, K. (2006b), "Searching for the holy grail of scientific hydrology: Q t=(s, r, δt) a as closure." *Hydrology and earth system sciences*, 10, 609–618, URL https://doi.org/10.5194/hess-10-609-2006.

Beven, K. J. (2000), "Uniqueness of place and process representations in hydrological modelling." *Hydrology and Earth System Sciences*, 4, 203–213, URL https://www.doi.org/10.5194/hess-4-203-2000.

Beven, K. J. and M. J. Kirkby (1979), "A physically based, variable contributing area model of basin hydrology / un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant." *Hydrological Sciences Bulletin*, 24, 43–69, URL https://www.doi.org/10.1080/02626667909491834.

Biondi, D. and D. L. De Luca (2016), "Rainfall-runoff model parameter conditioning on regional hydrological signatures: application to ungauged basins in southern italy." *Hydrology Research*, 48, 714–725, URL https://www.doi.org/10.2166/nh.2016.097.

Bloomfield, J. P., D. J. Allen, and K. J. Griffiths (2009), "Examining geological controls on baseflow index (bfi) using regression analysis: An illustration from the thames basin, uk." *Journal of Hydrology*, 373, 164–176, URL https://www.doi.org/10.1016/j.jhydrol.2009.04.025.

Booker, D. J. and R. A. Woods (2014), "Comparing and combining physically-based and empirically-based approaches for estimating the hydrology of ungauged catchments." *Journal of Hydrology*, 508, 227–239, URL https://www.doi.org/10.1016/j.jhydrol.2013.11.007.

Bouaziz, L., A. Weerts, J. Schellekens, E. Sprokkereef, J. Stam, H. Savenije, and M. Hrachowitz (2018), "Redressing the balance: quantifying net intercatchment groundwater flows." *Hydrology and Earth System Sciences*, 22, 6415–6434, URL https://doi.org/10.5194/hess-22-6415-2018.

Box, G. E. P. and D. R. Cox (1964), "An analysis of transformations." *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 26, 211–252, URL www.jstor.org/stable/2984418.

Boyle, D. P. (2001), *Multicriteria calibration of hydrologic models*. The University of Arizona.

Boyle, D. P., H. V. Gupta, S. Sorooshian, V. Koren, Z. Zhang, and M. Smith (2001), "Toward improved streamflow forecasts: value of semidistributed modeling." *Water Resources Research*, 37, 2749–2759, URL https://www.doi.org/10.1029/2000wr000207.

Brunner, M. I., R. Furrer, and A. C. Favre (2019), "Modeling the spatial dependence of floods using the fisher copula." *Hydrology and Earth System Sciences*, 23, 107–124, URL https://www.doi.org/10.5194/hess-23-107-2019.

Bulygina, N., C. Ballard, N. McIntyre, G. O'Donnell, and H. Wheater (2012), "Integrating different types of information into hydrological model parameter estimation: Application to ungauged catchments and land use scenario analysis." *Water Resources Research*, 48, URL https://www.doi.org/10.1029/2011wr011207.

Bulygina, N., N. McIntyre, and H. Wheater (2009), "Conditioning rainfall-runoff model parameters for ungauged catchments and land management impacts analysis." *Hydrology and Earth System Sciences*, 13, 893–904, URL https://www.doi.org/10.5194/hess-13-893-2009.

Bulygina, N., N. McIntyre, and H. Wheater (2011), "Bayesian conditioning of a rainfall-runoff model for predicting flows in ungauged catchments and under land use changes." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2010wr009240.

Burn, D. H. (1997), "Catchment similarity for regional flood frequency analysis using seasonality measures." *Journal of Hydrology*, 202, 212–230, URL https://www.doi.org/10.1016/S0022-1694(97)00068-1.

Butcher, J. C. (2016), *Numerical methods for ordinary differential equations*. John Wiley & Sons, URL https://www.doi.org/10.1002/9781119121534.

Carlier, C., S. B. Wirth, F. Cochand, D. Hunkeler, and P. Brunner (2018), "Geology controls streamflow dynamics." *Journal of Hydrology*, 566, 756–769, URL https://doi.org/10.1016/j.jhydrol.2018.08.069.

Carlier, C., S. B. Wirth, F. Cochand, D. Hunkeler, and P. Brunner (2019), "Exploring geological and topographical controls on low flows with hydrogeological models." *Groundwater*, 57, 48–62, URL https://doi.org/10.1111/gwat.12845.

Castellarin, A., G. Botter, D. A. Hughes, S. Liu, T. B. M. J. Ouarda, J. Parajka, D. A. Post, M. Sivapalan, C. Spence, A. Viglione, and R. M. Vogel (2013), *Prediction of flow duration curves in ungauged basins*, 135–162. Cambridge University Press, Cambridge, URL https://www.doi.org/10.1017/CBO9781139235761.010.

Castiglioni, S., A. Castellarin, and A. Montanari (2009), "Prediction of low-flow indices in ungauged basins through physiographical space-based interpolation." *Journal of Hydrology*, 378, 272–280, URL https://www.doi.org/10.1016/j.jhydrol.2009.09.032.

Castiglioni, S., L. Lombardi, E. Toth, A. Castellarin, and A. Montanari (2010), "Calibration of rainfall-runoff models in ungauged basins: A regional maximum likelihood approach." *Advances in Water Resources*, 33, 1235–1242, URL https://www.doi.org/10.1016/j.advwatres.2010.04.009.

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

Clark, M. P., D. Kavetski, and F. Fenicia (2011a), "Pursuing the method of multiple working hypotheses for hydrological modeling." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2010wr009827.

Clark, M. P., B. Nijssen, J. D. Lundquist, D. Kavetski, D. E. Rupp, R. A. Woods, J. E. Freer, E. D. Gutmann, A. W. Wood, L. D. Brekke, J. R. Arnold, D. J. Gochis, and R. M. Rasmussen (2015a), "A unified approach for process-based hydrologic modeling: 1. modeling concept." *Water Resources Research*, 51, 2498–2514, URL https://www.doi.org/10.1002/2015wr017198.

Clark, M. P., B. Nijssen, J. D. Lundquist, D. Kavetski, D. E. Rupp, R. A. Woods, J. E. Freer, E. D. Gutmann, A. W. Wood, D. J. Gochis, R. M. Rasmussen, D. G. Tarboton, V. Mahat, G. N. Flerchinger, and D. G. Marks (2015b), "A unified approach for process-based hydrologic modeling: 2. model implementation and case studies." *Water Resources Research*, 51, 2515–2542, URL https://www.doi.org/10.1002/2015WR017200.

Clark, M. P., A. G. Slater, D. E. Rupp, R. A. Woods, J. A. Vrugt, H. V. Gupta, T. Wagener, and L. E. Hay (2008), "Framework for understanding structural errors (fuse): A modular framework to diagnose differences between hydrological models." *Water Resources Research*, 44, URL https://www.doi.org/10.1029/2007wr006735.

Clark, Martyn P., Hilary K. McMillan, Daniel B. G. Collins, Dmitri Kavetski, and Ross A. Woods (2011b), "Hydrological field data from a modeller's perspective: Part 2: process-based evaluation of model hypotheses." *Hydrological Processes*, 25, 523–543, URL https://www.doi.org/10.1002/hyp.7902.

Court, A. (1962), "Measures of streamflow timing." *Journal of Geophysical Research (1896-1977)*, 67, 4335–4339, URL https://www.doi.org/10.1029/JZ067i011p04335.

Craig, J. R., G. Brown, R. Chlumsky, R. W. Jenkinson, G. Jost, K. Lee, J. Mai, M. Serrer, N. Sgro, M. Shafii, A. P. P. Snowdon, and B. A. Tolson (2020), "Flexible watershed simulation with the raven hydrological modelling framework." *Environmental Modelling & Software*, 129, 104728, URL https://www.doi.org/10.1016/j.envsoft.2020.104728.

Dal Molin, M., M. Schirmer, M. Zappa, and F. Fenicia (2020), "Understanding dominant controls on streamflow spatial variability to set up a semi-distributed hydrological model: the case

study of the thur catchment." *Hydrology and Earth System Sciences*, 24, 1319–1345, URL https://www.doi.org/10.5194/hess-24-1319-2020.

David, P. C., D. Y. Oliveira, F. Grison, M. Kobiyama, and P. L. B. Chaffe (2019), "Systematic increase in model complexity helps to identify dominant streamflow mechanisms in two small forested basins." *Hydrological Sciences Journal*, 64, 455–472, URL https://www.doi.org/10.1080/02626667.2019.1585858.

De Lavenne, A., G. Thirel, V. Andréassian, C. Perrin, and M. H. Ramos (2016), "Spatial variability of the parameters of a semi-distributed hydrological model." In *7th International Water Resources Management Conference of ICWRS*, volume 373, 87–94, URL https://doi.org/10.5194/piahs-373-87-2016.

Dooge, J. C. I. (1988), "Hydrology in perspective." *Hydrological Sciences Journal*, 33, 61–85, URL https://doi.org/10.1080/02626668809491223.

Dornes, P. F., B. A. Tolson, B. Davison, A. Pietroniro, J. W. Pomeroy, and P. Marsh (2008), "Regionalisation of land surface hydrological model parameters in subarctic and arctic environments." *Physics and Chemistry of the Earth, Parts A/B/C*, 33, 1081–1089, URL https://www.doi.org/10.1016/j.pce.2008.07.007.

Doulatyari, B., A. Betterle, S. Basso, B. Biswal, M. Schirmer, and G. Botter (2015), "Predicting streamflow distributions and flow duration curves from landscape and climate." *Advances in Water Resources*, 83, 285–298, URL https://www.doi.org/10.1016/j.advwatres.2015.06.013.

Doulatyari, B., A. Betterle, D. Radny, E. A. Celegon, P. Fanton, M. Schirmer, and G. Botter (2017), "Patterns of streamflow regimes along the river network: The case of the thur river." *Environmental Modelling & Software*, 93, 42–58, URL https://www.doi.org/10.1016/j.envsoft.2017.03.002.

Dowell, M. and P. Jarratt (1972), "The "pegasus" method for computing the root of an equation." *BIT Numerical Mathematics*, 12, 503–508, URL https://www.doi.org/10.1007/BF01932959.

Dutta, R., M. Schoengens, J. P. Onnela, and A. Mira (2017), "Abcpy: A user-friendly, extensible, and parallel library for approximate bayesian computation." In *Proceedings of the Platform for Advanced Scientific Computing Conference*, PASC '17, 8:1–8:9, ACM, New York, NY, USA, URL https://www.doi.org/10.1145/3093172.3093233.

Eckhardt, K. and U. Ulbrich (2003), "Potential impacts of climate change on groundwater recharge and streamflow in a central european low mountain range." *Journal of Hydrology*, 284, 244–252, URL https://www.doi.org/10.1016/j.jhydrol.2003.08.005.

Evin, G., M. Thyer, D. Kavetski, D. McInerney, and G. Kuczera (2014), "Comparison of joint versus postprocessor approaches for hydrological uncertainty estimation accounting for error

autocorrelation and heteroscedasticity." *Water Resources Research*, 50, 2350–2375, URL https://www.doi.org/10.1002/2013WR014185.

Falkenmark, M. and T. Chapman (1989), *Comparative hydrology: An ecological approach to land and water resources*. The Unesco Press.

Fang, X., J. W. Pomeroy, C. J. Westbrook, X. Guo, A. G. Minke, and T. Brown (2010), "Prediction of snowmelt derived streamflow in a wetland dominated prairie basin." *Hydrology and Earth System Sciences*, 14, 991–1006, URL https://www.doi.org/10.5194/hess-14-991-2010.

Fatichi, S., E. R. Vivoni, F. L. Ogden, V. Y. Ivanov, B. Mirus, D. Gochis, C. W. Downer, M. Camporese, J. H. Davison, B. A. Ebel, N. Jones, J. Kim, G. Mascaro, R. Niswonger, P. Restrepo, R. Rigon, C. Shen, M. Sulis, and D. Tarboton (2016), "An overview of current applications, challenges, and future trends in distributed process-based models in hydrology." *Journal of Hydrology*, 537, 45–60, URL https://doi.org/10.1016/j.jhydrol.2016.03.026.

Fenicia, F., D. Kavetski, P. Reichert, and C. Albert (2018), "Signature-domain calibration of hydrological models using approximate bayesian computation: Empirical analysis of fundamental properties." *Water Resources Research*, 54, 3958–3987, URL https://www.doi.org/10.1002/2017wr021616.

Fenicia, F., D. Kavetski, and H. H. G. Savenije (2011), "Elements of a flexible approach for conceptual hydrological modeling: 1. motivation and theoretical development." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2010wr010174.

Fenicia, F., D. Kavetski, H. H. G. Savenije, M. P. Clark, G. Schoups, L. Pfister, and J. Freer (2014), "Catchment properties, function, and conceptual model representation: is there a correspondence?" *Hydrological Processes*, 28, 2451–2467, URL https://www.doi.org/10.1002/hyp.9726.

Fenicia, F., D. Kavetski, H. H. G. Savenije, and L. Pfister (2016), "From spatially variable streamflow to distributed hydrological models: Analysis of key modeling decisions." *Water Resources Research*, 52, 954–989, URL https://www.doi.org/10.1002/2015wr017398.

Fenicia, F., H. H. G. Savenije, P. Matgen, and L. Pfister (2006), "Is the groundwater reservoir linear? learning from data in hydrological modelling." *Hydrology and Earth System Sciences*, 10, 139–150, URL https://www.doi.org/10.5194/hess-10-139-2006.

Fenicia, F., H. H. G. Savenije, P. Matgen, and L. Pfister (2008), "Understanding catchment behavior through stepwise model concept improvement." *Water Resources Research*, 44, URL https://www.doi.org/10.1029/2006WR005563.

Fenicia, F., S. Wrede, D. Kavetski, L. Pfister, L. Hoffmann, H. H. G. Savenije, and J. J. McDonnell (2010), "Assessing the impact of mixing assumptions on the estimation of streamwater mean residence time." *Hydrological Processes*, 24, 1730–1741, URL https://www.doi.org/10.1002/hyp.7595.

Feyen, L.U.C., M. Kalas, and J. A. Vrugt (2008), "Semi-distributed parameter optimization and uncertainty assessment for large-scale streamflow simulation using global optimization." *Hydrological Sciences Journal*, 53, 293–308, URL https://www.doi.org/10.1623/hysj.53.2.293.

Formetta, G., A. Antonello, S. Franceschi, O. David, and R. Rigon (2014), "Hydrological modelling with components: A gis-based open-source framework." *Environmental Modelling & Software*, 55, 190–200, URL https://www.doi.org/10.1016/j.envsoft.2014.01.019.

Fundel, F., S. Jorg-Hess, and M. Zappa (2013), "Monthly hydrometeorological ensemble prediction of streamflow droughts and corresponding drought indices." *Hydrology and Earth System Sciences*, 17, 395–407, URL https://www.doi.org/10.5194/hess-17-395-2013.

Futter, M. N., M. A. Erlandsson, D. Butterfield, P. G. Whitehead, S. K. Oni, and A. J. Wade (2014), "Persist: a flexible rainfall-runoff modelling toolkit for use with the inca family of models." *Hydrology and Earth System Sciences*, 18, 855–873, URL https://www.doi.org/10.5194/hess-18-855-2014.

Gao, H., M. Hrachowitz, F. Fenicia, S. Gharari, and H. H. G. Savenije (2014), "Testing the realism of a topography-driven model (flex-topo) in the nested catchments of the upper heihe, china." *Hydrology and Earth System Sciences*, 18, 1895–1915, URL https://www.doi.org/10.5194/hess-18-1895-2014.

Garen, D. C. and D. Marks (2001), "Spatial fields of meteorological input data including forest canopy corrections for an energy budget snow simulation model." *IAHS PUBLICATION*, 349–354.

Gurtz, J., A. Baltensweiler, and H. Lang (1999), "Spatially distributed hydrotope-based modelling of evapotranspiration and runoff in mountainous basins." *Hydrological Processes*, 13, 2751–2768, URL https://www.doi.org/10.1002/(Sici)1099-1085(19991215)13:17<2751::Aid-Hyp897>3.3.Co;2-F.

Gurtz, J., M. Verbunt, M. Zappa, M. Moesch, F. Pos, and U. Moser (2003), "Long-term hydrometeorological measurements and model-based analyses in the hydrological research catchment rietholzbach." *Journal of Hydrology and Hydromechanics*, 51, 162–174.

He, Y., A. Bárdossy, and E. Zehe (2011), "A review of regionalisation for continuous streamflow simulation." *Hydrology and Earth System Sciences*, 15, 3539–3553, URL https://www.doi.org/10.5194/hess-15-3539-2011.

Henn, B., M. P. Clark, D. Kavetski, A. J. Newman, M. Hughes, B. McGurk, and J. D. Lundquist (2018), "Spatiotemporal patterns of precipitation inferred from streamflow observations across the sierra nevada mountain range." *Journal of Hydrology*, 556, 993–1012, URL https://www.doi.org/10.1016/j.jhydrol.2016.08.009.

Hingray, B., B. Schaefli, A. Mezghani, and Y. Hamdi (2010), "Signature-based model calibration for hydrological prediction in mesoscale alpine catchments." *Hydrological Sciences Journal*, 55, 1002–1016, URL https://www.doi.org/10.1080/02626667.2010.505572.

Houska, T., P. Kraft, A. Chamorro-Chavez, and L. Breuer (2015), "Spotting model parameters using a ready-made python package." *PloS one*, 10, e0145180, URL https://doi.org/10.1371/journal.pone.0145180.

Hrachowitz, M. and M. P. Clark (2017), "Hess opinions: The complementary merits of competing modelling philosophies in hydrology." *Hydrology and Earth System Sciences*, 21, 3953–3973, URL https://www.doi.org/10.5194/hess-21-3953-2017.

Hrachowitz, M., O. Fovet, L. Ruiz, T. Euser, S. Gharari, R. Nijzink, J. Freer, H. H. G. Savenije, and C. Gascuel-Odoux (2014), "Process consistency in models: The importance of system signatures, expert knowledge, and process complexity." *Water Resources Research*, 50, 7445–7469, URL https://www.doi.org/10.1002/2014WR015484.

Hrachowitz, M., H.H.G. Savenije, G. Blöschl, J.J. McDonnell, M. Sivapalan, J.W. Pomeroy, B. Arheimer, T. Blume, M.P. Clark, and U. Ehret (2013), "A decade of predictions in ungauged basins (pub)—a review." *Hydrological sciences journal*, 58, 1198–1255, URL https://doi.org/10.1080/02626667.2013.803183.

Hurford, A. P. and J. J. Harou (2014), "Balancing ecosystem services with energy and food security - assessing trade-offs from reservoir operation and irrigation investments in kenya's tana basin." *Hydrology and Earth System Sciences*, 18, 3259–3277, URL https://www.doi.org/10.5194/hess-18-3259-2014.

Ibbitt, R. P. and T. O'Donnell (1971), "Designing conceptual catchment models for automatic fitting methods." *IAHS Publication*, 101, 462–475.

Ivanov, V. Y., E. R. Vivoni, R. L. Bras, and D. Entekhabi (2004), "Preserving high-resolution surface and rainfall data in operational-scale basin hydrology: a fully-distributed physically-based approach." *Journal of Hydrology*, 298, 80–111, URL https://doi.org/10.1016/j.jhydrol.2004.03.041. The Distributed Model Intercomparison Project (DMIP).

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

Jansen, K. F., A. J. Teuling, J. R. Craig, M. Dal Molin, W. J. M. Knoben, J. Parajka, M. Vis, and L. A. Melsen (2021), "Mimicry of a conceptual hydrological model (hbv): What's in a name?" *Water Resources Research*, 57, e2020WR029143, URL https://www.doi.org/10.1029/2020WR029143.

Jasper, K., P. Calanca, D. Gyalistras, and J. Fuhrer (2004), "Differential impacts of climate change on the hydrology of two alpine river basins." *Climate Research*, 26, 113–129, URL https://www.doi.org/10.3354/cr026113.

Jorg-Hess, S., S. B. Kempf, F. Fundel, and M. Zappa (2015), "The benefit of climatological and calibrated reforecast data for simulating hydrological droughts in switzerland." *Meteorological Applications*, 22, 444–458, URL https://www.doi.org/10.1002/met.1474.

Kampf, S. K. and S. J. Burges (2007), "A framework for classifying and comparing distributed hillslope and catchment hydrologic models." *Water Resources Research*, 43, URL https://doi.org/10.1029/2006WR005370.

Kavetski, D. and M. P. Clark (2010), "Ancient numerical daemons of conceptual hydrological modeling: 2. impact of time stepping schemes on model analysis and prediction." *Water Resources Research*, 46, URL https://www.doi.org/10.1029/2009wr008896.

Kavetski, D. and F. Fenicia (2011), "Elements of a flexible approach for conceptual hydrological modeling: 2. application and experimental insights." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2011wr010748.

Kavetski, D., F. Fenicia, and M. P. Clark (2011), "Impact of temporal data resolution on parameter inference and model identification in conceptual hydrological modeling: Insights from an experimental catchment." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2010wr009525.

Kavetski, D., F. Fenicia, P. Reichert, and C. Albert (2018), "Signature-domain calibration of hydrological models using approximate bayesian computation: Theory and comparison to existing applications." *Water Resources Research*, 54, 4059–4083, URL https://www.doi.org/10.1002/2017wr020528.

Kavetski, D. and G. Kuczera (2007), "Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration." *Water Resources Research*, 43, URL https://www.doi.org/10.1029/2006wr005195.

Kavetski, D., G. Kuczera, M. Thyer, and B. Renard (2007), "Multistart newton-type optimisation methods for the calibration of conceptual hydrological models." *Modsim 2007: International Congress on Modelling and Simulation*, 2513–2519.

Kim, D., I. W. Jung, and J. A. Chun (2017), "A comparative assessment of rainfall–runoff modelling against regional flow duration curves for ungauged catchments." *Hydrology and Earth System Sciences*, 21, 5647–5661, URL https://www.doi.org/10.5194/hess-21-5647-2017.

Kirchner, J. W. (2006), "Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology." *Water Resources Research*, 42, URL https://www.doi.org/10.1029/2005wr004362.

Kirchner, J. W. (2009), "Catchments as simple dynamical systems: Catchment characterization, rainfall-runoff modeling, and doing hydrology backward." *Water Resources Research*, 45, URL https://www.doi.org/10.1029/2008wr006912.

Kirchner, J. W., X. H. Feng, C. Neal, and A. J. Robson (2004), "The fine structure of water-quality dynamics: the (high-frequency) wave of the future." *Hydrological Processes*, 18, 1353–1359, URL https://www.doi.org/10.1002/hyp.5537.

Kneis, D. (2015), "A lightweight framework for rapid development of object-based hydrological model engines." *Environmental Modelling & Software*, 68, 110–121, URL https://www.doi.org/10.1016/j.envsoft.2015.02.009.

Knoben, W. J. M., J. E. Freer, K. J. A. Fowler, M. C. Peel, and R. A. Woods (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." *Geoscientific Model Development*, 12, 2463–2480, URL https://www.doi.org/10.5194/gmd-12-2463-2019.

Knoben, W. J. M., J. E. Freer, M. C. Peel, K. J. A. Fowler, and R. A. Woods (2020), "A brief analysis of conceptual model structure uncertainty using 36 models and 559 catchments." *Water Resources Research*, 56, e2019WR025975, URL https://doi.org/10.1029/2019WR025975.

Kraft, P., K. B. Vaché, H. Frede, and L. Breuer (2011), "Cmf: A hydrological programming language extension for integrated catchment models." *Environmental Modelling & Software*, 26, 828–830, URL https://www.doi.org/10.1016/j.envsoft.2010.12.009.

Kratzert, F., D. Klotz, M. Herrnegger, A. K. Sampson, S. Hochreiter, and G. S. Nearing (2019), "Toward improved predictions in ungauged basins: Exploiting the power of machine learning." *Water Resources Research*, URL https://www.doi.org/10.1029/2019wr026065.

Kroll, C. N. and P. Song (2013), "Impact of multicollinearity on small sample hydrologic regression models." *Water Resources Research*, 49, 3756–3769, URL https://www.doi.org/10.1002/wrcr.20315.

Kuczera, G., D. Kavetski, S. Franks, and M. Thyer (2006), "Towards a bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters." *Journal of Hydrology*, 331, 161–177, URL https://www.doi.org/10.1016/j.jhydrol.2006.05.010.

Kuentz, A., B. Arheimer, Y. Hundecha, and T. Wagener (2017), "Understanding hydrologic variability across europe through catchment classification." *Hydrology and Earth System Sciences*, 21, 2863–2879, URL https://www.doi.org/10.5194/hess-21-2863-2017.

Lacey, G. C. and R. B. Grayson (1998), "Relating baseflow to catchment properties in south-eastern australia." *Journal of Hydrology*, 204, 231–250, URL https://www.doi.org/10.1016/S0022-1694(97)00124-8.

Ladson, A. R., R. Brown, B. Neal, and R. Nathan (2013), "A standard approach to baseflow separation using the lyne and hollick filter." *Australasian Journal of Water Resources*, 17, 25–34, URL https://www.doi.org/10.7158/13241583.2013.11465417.

Lam, S. K., A. Pitrou, and S. Seibert (2015), "Numba: A llvm-based python jit compiler." In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, Association for Computing Machinery, New York, NY, USA, URL https://www.doi.org/10.1145/2833157.2833162.

Leavesley, G. H. (1984), *Precipitation-runoff modeling system: User's manual*, volume 83. US Department of the Interior.

Leavesley, G. H., S. L. Markstrom, M. S. Brewer, and R. J. Viger (1996), "The modular modeling system (mms)—the physical process modeling component of a database-centered decision support system for water and power management." *Water, Air, & Soil Pollution*, 90, 303–311, URL https://doi.org/10.1007/BF00619290.

Lerat, J., V. Andreassian, C. Perrin, J. Vaze, J. M. Perraud, P. Ribstein, and C. Loumagne (2012), "Do internal flow measurements improve the calibration of rainfall-runoff models?" *Water Resources Research*, 48, URL https://doi.org/10.1029/2010WR010179.

Lindstrom, G., B. Johansson, M. Persson, M. Gardelin, and S. Bergstrom (1997), "Development and test of the distributed hbv-96 hydrological model." *Journal of Hydrology*, 201, 272–288, URL https://www.doi.org/10.1016/S0022-1694(97)00041-3.

Lombardi, L., E. Toth, A. Castellarin, A. Montanari, and A. Brath (2012), "Calibration of a rainfall–runoff model at regional scale by optimising river discharge statistics: Performance analysis for the average/low flow regime." *Physics and Chemistry of the Earth, Parts A/B/C*, 42-44, 77–84, URL https://www.doi.org/10.1016/j.pce.2011.05.013.

Macklin, M. G. and J. Lewin (2015), "The rivers of civilization." *Quaternary Science Reviews*, 114, 228–244, URL https://www.doi.org/10.1016/j.quascirev.2015.02.004.

Marsh, C. B., J. W. Pomeroy, and H. S. Wheater (2020), "The canadian hydrological model (chm) v1.0: a multi-scale, multi-extent, variable-complexity hydrological model – design and overview." *Geoscientific Model Development*, 13, 225–247, URL https://www.doi.org/10.5194/gmd-13-225-2020.

Matgen, P., F. Fenicia, S. Heitz, D. Plaza, R. de Keyser, V. R. N. Pauwels, W. Wagner, and H. Savenije (2012), "Can ascat-derived soil wetness indices reduce predictive uncertainty in well-gauged

areas? a comparison with in situ observed soil moisture in an assimilation application." *Advances in Water Resources*, 44, 49–65, URL https://www.doi.org/10.1016/j.advwatres.2012.03.022.

Maxwell, R. M. (2013), "A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling." *Advances in Water Resources*, 53, 109–117, URL https://www.doi.org/10.1016/j.advwatres.2012.10.001.

Mazvimavi, D., A. M. J. Meijerink, H. H. G. Savenije, and A. Stein (2005), "Prediction of flow characteristics using multiple regression and neural networks: A case study in zimbabwe." *Physics and Chemistry of the Earth*, 30, 639–647, URL https://www.doi.org/10.1016/j.pce.2005.08.003.

McDonnell, J. J., C. Gabrielli, A. Ameli, J. Ekanayake, F. Fenicia, J. Freer, C. Graham, B. McGlynn, U. Morgenstern, A. Pietroniro, T. Sayama, J. Seibert, M. Stewart, K. Vache, M. Weiler, and R. Woods (2021), "The maimai m8 experimental catchment database: Forty years of process-based research on steep, wet hillslopes." *Hydrological Processes*, 35, e14112, URL https://www.doi.org/10.1002/hyp.14112.

McDonnell, J. J and R. Woods (2004), "On the need for catchment classification." *Journal of Hydrology*, 299, 2–3, URL https://doi.org/10.1016/j.jhydrol.2004.09.003.

McInerney, D., M. Thyer, D. Kavetski, F. Githui, T. Thayalakumaran, M. Liu, and G. Kuczera (2018), "The importance of spatiotemporal variability in irrigation inputs for hydrological modeling of irrigated catchments." *Water Resources Research*, 54, 6792–6821, URL https://www.doi.org/10.1029/2017wr022049.

McInerney, D., M. Thyer, D. Kavetski, J. Lerat, and G. Kuczera (2017), "Improving probabilistic prediction of daily streamflow by identifying pareto optimal approaches for modeling heteroscedastic residual errors." *Water Resources Research*, 53, 2199–2239, URL https://www.doi.org/10.1002/2016wr019168.

McMillan, H., M. Gueguen, E. Grimon, R. Woods, M. Clark, and D. E. Rupp (2014), "Spatial variability of hydrological processes and model structure diagnostics in a 50 km(2) catchment." *Hydrological Processes*, 28, 4896–4913, URL https://www.doi.org/10.1002/hyp.9988.

Melsen, L., A. Teuling, P. Torfs, M. Zappa, N. Mizukami, M. Clark, and R. Uijlenhoet (2016), "Representation of spatial and temporal variability in large-domain hydrological models: case study for a mesoscale pre-alpine basin." *Hydrology and Earth System Sciences*, 20, 2207–2226, URL https://www.doi.org/10.5194/hess-20-2207-2016.

Melsen, L. A., A. J. Teuling, S. W. van Berkum, P. J. J. F. Torfs, and R. Uijlenhoet (2014), "Catchments as simple dynamical systems: A case study on methods and data requirements for parameter identification." *Water Resources Research*, 50, 5577–5596, URL https://www.doi.org/10.1002/2013WR014720.

Menzel, L. (1996), "Modelling canopy resistances and transpiration of grassland." *Physics and Chemistry of the Earth*, 21, 123–129, URL https://www.doi.org/10.1016/S0079-1946(97)85572-3.

Merz, R. and G. Blöschl (2004), "Regionalisation of catchment model parameters." *Journal of Hydrology*, 287, 95–123, URL https://www.doi.org/10.1016/j.jhydrol.2003.09.028.

Meyer, B. (1988), *Object-oriented software construction*, volume 2. Prentice hall New York.

Mohamed, Y. A., H. H. G. Savenije, W. G. M. Bastiaanssen, and B. J. J. M. van den Hurk (2006), "New lessons on the sudd hydrology learned from remote sensing and climate modeling." *Hydrology and Earth System Sciences*, 10, 507–518, URL https://www.doi.org/10.5194/hess-10-507-2006.

Montanari, A. and E. Toth (2007), "Calibration of hydrological models in the spectral domain: An opportunity for scarcely gauged basins?" *Water Resources Research*, 43, URL https://www.doi.org/10.1029/2006wr005184.

Monteith, J. L. (1975), *Vegetation and the atmosphere*. Academic Press, London ; New York.

Moore, R. J. and R. T. Clarke (1981), "A distribution function approach to rainfall runoff modeling." *Water Resources Research*, 17, 1367–1382, URL https://www.doi.org/10.1029/WR017i005p01367.

Moradkhani, H. and S. Sorooshian (2008), *General Review of Rainfall-Runoff Modeling: Model Calibration, Data Assimilation, and Uncertainty Analysis*, 1–24. Springer Berlin Heidelberg, Berlin, Heidelberg, URL https://www.doi.org/10.1007/978-3-540-77843-1_1.

Moser, A., D. Wemyss, R. Scheidegger, F. Fenicia, M. Honti, and C. Stamm (2018), "Modelling biocide and herbicide concentrations in catchments of the rhine basin." *Hydrology and Earth System Sciences*, 22, 4229–4249, URL https://www.doi.org/10.5194/hess-22-4229-2018.

Mwakalila, S. (2003), "Estimation of stream flows of ungauged catchments for river basin management." *Physics and Chemistry of the Earth, Parts A/B/C*, 28, 935–942, URL https://www.doi.org/10.1016/j.pce.2003.08.039.

Nash, J. E. (1957), "The form of the instantaneous unit hydrograph." *International Association of Scientific Hydrology, Publ*, 3, 114–121.

Nijzink, R. C., S. Almeida, I. G. Pechlivanidis, R. Capell, D. Gustafssons, B. Arheimer, J. Parajka, J. Freer, D. Han, T. Wagener, R. R. P. van Nooijen, H. H. G. Savenije, and M. Hrachowitz (2018), "Constraining conceptual hydrological models with multiple information sources." *Water Resources Research*, 54, 8332–8362, URL https://www.doi.org/10.1029/2017wr021895.

Nijzink, R. C., L. Samaniego, J. Mai, R. Kumar, S. Thober, M. Zink, D. Schäfer, H. H. G. Savenije, and M. Hrachowitz (2016), "The importance of topography-controlled sub-grid process heterogeneity and semi-quantitative prior constraints in distributed hydrological models." *Hydrology and Earth System Sciences*, 20, 1151–1176, URL https://www.doi.org/10.5194/hess-20-1151-2016.

Nott, D. J., Y. Fan, L. Marshall, and S. A. Sisson (2014), "Approximate bayesian computation and bayes' linear analysis: Toward high-dimensional abc." *Journal of Computational and Graphical Statistics*, 23, 65–86, URL https://www.doi.org/10.1080/10618600.2012.751874.

Olden, J. D. and N. L. Poff (2003), "Redundancy and the choice of hydrologic indices for characterizing streamflow regimes." *River Research and Applications*, 19, 101–121, URL https://www.doi.org/10.1002/rra.700.

Parajka, J., R. Merz, and G. Blöschl (2005), "A comparison of regionalisation methods for catchment model parameters." *Hydrology and Earth System Sciences*, 9, 157–171, URL https://www.doi.org/10.5194/hess-9-157-2005.

Parajka, J., A. Viglione, M. Rogger, J. L. Salinas, M. Sivapalan, and G. Blöschl (2013), "Comparative assessment of predictions in ungauged basins - part 1: Runoff-hydrograph studies." *Hydrology and Earth System Sciences*, 17, 1783–1795, URL https://www.doi.org/10.5194/hess-17-1783-2013.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga (2019), "Pytorch: An imperative style, high-performance deep learning library." In *Advances in Neural Information Processing Systems*, 8024–8035.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg (2011), "Scikit-learn: Machine learning in python." *the Journal of machine Learning research*, 12, 2825–2830.

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

Pool, S., D. Viviroli, and J. Seibert (2017), "Prediction of hydrographs and flow-duration curves in almost ungauged catchments: Which runoff measurements are most informative for model calibration?" *Journal of Hydrology*, 554, 613–622, URL https://www.doi.org/10.1016/j.jhydrol.2017.09.037.

Prieto, C., N. Le Vine, D. Kavetski, E. García, and R. Medina (2019), "Flow prediction in ungauged catchments using probabilistic random forests regionalization and new statistical adequacy tests." *Water Resources Research*, 55, 4364–4392, URL https://www.doi.org/10.1029/2018wr023254.

Refsgaard, J. C. (1996), *Terminology, Modelling Protocol And Classification of Hydrological Model Codes*, 17–39. Springer Netherlands, Dordrecht, URL https://www.doi.org/10.1007/978-94-009-0257-2_2.

Refsgaard, J. C. and B. Storm (1995), *MIKE SHE*, chapter 23, 809–846. Water Resources Publications.

Reichert, P. and J. Mieleitner (2009), "Analyzing input and structural uncertainty of nonlinear dynamic models with stochastic, time-dependent parameters." *Water Resources Research*, 45, URL https://www.doi.org/10.1029/2009wr007814.

Renard, B., D. Kavetski, E. Leblois, M. Thyer, G. Kuczera, and S. W. Franks (2011), "Toward a reliable decomposition of predictive uncertainty in hydrological modeling: Characterizing rainfall errors using conditional simulation." *Water Resources Research*, 47, URL https://www.doi.org/10.1029/2011WR010643.

Rigon, R., G. Bertoldi, and T. M. Over (2006), "Geotop: A distributed hydrological model with coupled water and energy budgets." *Journal of Hydrometeorology*, 7, 371 – 388, URL https://www.doi.org/10.1175/JHM497.1.

Samaniego, L., R. Kumar, and S. Attinger (2010), "Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale." *Water Resources Research*, 46, URL https://www.doi.org/10.1029/2008wr007327.

Sawicz, K., T. Wagener, M. Sivapalan, P. A. Troch, and G. Carrillo (2011), "Catchment classification: empirical analysis of hydrologic similarity based on catchment function in the eastern usa." *Hydrology and Earth System Sciences*, 15, 2895–2911, URL https://www.doi.org/10.5194/hess-15-2895-2011.

Scherrer, S. and F. Naef (2003), "A decision scheme to indicate dominant hydrological flow processes on temperate grassland." *Hydrological Processes*, 17, 391–401, URL https://www.doi.org/10.1002/hyp.1131.

Schirmer, M., J. Luster, N. Linde, P. Perona, E. A. D. Mitchell, D. A. Barry, J. Hollender, O. A. Cirpka, P. Schneider, T. Vogt, D. Radny, and E. Durisch-Kaiser (2014), "Morphological, hydrological, biogeochemical and ecological changes and challenges in river restoration; the thur river case study." *Hydrology and Earth System Sciences*, 18, 2449–2462, URL https://www.doi.org/10.5194/hess-18-2449-2014.

Seibert, J. (1999), "Regionalisation of parameters for a conceptual rainfall-runoff model." *Agricultural and Forest Meteorology*, 98-99, 279–293, URL https://www.doi.org/10.1016/S0168-1923(99)00105-7.

Seibert, J. and J. J. McDonnell (2002), "On the dialog between experimentalist and modeler in catchment hydrology: Use of soft data for multicriteria model calibration." *Water Resources Research*, 38, 23–1–23–14, URL https://www.doi.org/10.1029/2001WR000978.

Seibert, J., A. Rodhe, and K. Bishop (2003), "Simulating interactions between saturated and unsaturated storage in a conceptual runoff model." *Hydrological Processes*, 17, 379–390, URL https://doi.org/10.1002/hyp.1130.

Seneviratne, S. I., I. Lehner, J. Gurtz, A. J. Teuling, H. Lang, U. Moser, D. Grebner, L. Menzel, K. Schroff, T. Vitvar, and M. Zappa (2012), "Swiss prealpine rietholzbach research catchment and lysimeter: 32 year time series and 2003 drought event." *Water Resources Research*, 48, URL https://www.doi.org/10.1029/2011wr011749.

Shafii, M. and B. A. Tolson (2015), "Optimizing hydrological consistency by incorporating hydrological signatures into model calibration objectives." *Water Resources Research*, 51, 3796–3814, URL https://www.doi.org/10.1002/2014wr016520.

Singh, R., S. A. Archfield, and T. Wagener (2014), "Identifying dominant controls on hydrologic parameter transfer from gauged to ungauged catchments – a comparative hydrology approach." *Journal of Hydrology*, 517, 985–996, URL https://www.doi.org/10.1016/j.jhydrol.2014.06.030.

Sivapalan, M. (2006), *Pattern, Process and Function: Elements of a Unified Theory of Hydrology at the Catchment Scale*. URL https://www.doi.org/10.1002/0470848944.hsa012.

Sivapalan, M., K. Beven, and E. F. Wood (1987), "On hydrologic similarity: 2. a scaled model of storm runoff production." *Water Resources Research*, 23, 2266–2278, URL https://www.doi.org/10.1029/WR023i012p02266.

Sivapalan, M., G. Blöschl, L. Zhang, and R. Vertessy (2003a), "Downward approach to hydrological prediction." *Hydrological Processes*, 17, 2101–2111, URL https://www.doi.org/10.1002/hyp.1425.

Sivapalan, M., K. Takeuchi, S. W. Franks, V. K. Gupta, H. Karambiri, V. Lakshmi, X. Liang, J. J. McDonnell, E. M. Mendiondo, P. E. O'Connell, T. Oki, J. W. Pomeroy, D. Schertzer, S. Uhlenbrook, and E. Zehe (2003b), "Iahs decade on predictions in ungauged basins (pub), 2003–2012: Shaping an exciting future for the hydrological sciences." *Hydrological Sciences Journal*, 48, 857–880, URL https://www.doi.org/10.1623/hysj.48.6.857.51421.

Spence, C., P. H. Whitfield, J. W. Pomeroy, A. Pietroniro, D. H. Burn, D. L. Peters, and A. St-Hilaire (2013), "A review of the prediction in ungauged basins (pub) decade in canada." *Canadian Water Resources Journal / Revue canadienne des ressources hydriques*, 38, 253–262, URL https://www.doi.org/10.1080/07011784.2013.843867.

Thorsten, W., M. Sivapalan, P. Troch, and R. Woods (2007), "Catchment classification and hydrologic similarity." *Geography Compass*, 1, 901–931, URL https://www.doi.org/10.1111/j.1749-8198.2007.00039.x.

Toth, E. (2013), "Catchment classification based on characterisation of streamflow and precipitation time series." *Hydrology and Earth System Sciences*, 17, 1149–1159, URL https://www.doi.org/10.5194/hess-17-1149-2013.

Trancoso, R., S. Phinn, T. R. McVicar, J. R. Larsen, and C. A. McAlpine (2017), "Regional variation in streamflow drivers across a continental climatic gradient." *Ecohydrology*, 10, URL https://www.doi.org/10.1002/eco.1816.

van der Walt, S., S. C. Colbert, and G. Varoquaux (2011), "The numpy array: A structure for efficient numerical computation." *Computing in Science & Engineering*, 13, 22–30, URL https://www.doi.org/10.1109/MCSE.2011.37.

van Esse, W. R., C. Perrin, M. J. Booij, D. C. M. Augustijn, F. Fenicia, D. Kavetski, and F. Lobligeois (2013), "The influence of conceptual model structure on model performance: a comparative study for 237 french catchments." *Hydrology and Earth System Sciences*, 17, 4227–4239, URL https://www.doi.org/10.5194/hess-17-4227-2013.

Verbunt, M., M. Zappa, J. Gurtz, and P. Kaufmann (2006), "Verification of a coupled hydrometeorological modelling approach for alpine tributaries in the rhine basin." *Journal of Hydrology*, 324, 224–238, URL https://www.doi.org/10.1016/j.jhydrol.2005.09.036.

Vitolo, C., P. Wells, M. Dobias, and W. Buytaert (2016), "fuse: An r package for ensemble hydrological modelling." *Journal of Open Source Software*, 1, 52, URL https://www.doi.org/10.21105/joss.00052.

Viviroli, D., J. Gurtz, and M. Zappa (2007), *The Hydrological Modelling System PREVAH*. Geographica Bernensia P40. Berne: Institute of Geography, University of Berne.

Viviroli, D., M. Zappa, J. Gurtz, and R. Weingartner (2009), "An introduction to the hydrological modelling system prevah and its pre- and post-processing-tools." *Environmental Modelling & Software*, 24, 1209–1222, URL https://www.doi.org/10.1016/j.envsoft.2009.04.001.

von Freyberg, J., C. Moeck, and M. Schirmer (2015), "Estimation of groundwater recharge and drought severity with varying model complexity." *Journal of Hydrology*, 527, 844–857, URL https://www.doi.org/10.1016/j.jhydrol.2015.05.025.

von Freyberg, J., D. Radny, H. E. Gall, and M. Schirmer (2014), "Implications of hydrologic connectivity between hillslopes and riparian zones on streamflow composition." *Journal of Contaminant Hydrology*, 169, 62–74, URL https://www.doi.org/10.1016/j.jconhyd.2014.07.005.

Vrugt, J. A. and M. Sadegh (2013), "Toward diagnostic model calibration and evaluation: Approximate bayesian computation." *Water Resources Research*, 49, 4335–4345, URL https://www.doi.org/10.1002/wrcr.20354.

Wagener, T. and H. V. Gupta (2005), "Model identification for hydrological forecasting under uncertainty." *Stochastic Environmental Research and Risk Assessment*, 19, 378–387, URL https://doi.org/10.1007/s00477-005-0006-5.

Wagener, T., M. Sivapalan, J. McDonnell, R. Hooper, V. Lakshmi, X. Liang, and P. Kumar (2004), "Predictions in ungauged basins as a catalyst for multidisciplinary hydrology." *Eos, Transactions American Geophysical Union*, 85, 451–457.

Westerberg, I. K., J. L. Guerrero, P. M. Younger, K. J. Beven, J. Seibert, S. Halldin, J. E. Freer, and C. Y. Xu (2011), "Calibration of hydrological models using flow-duration curves." *Hydrology and Earth System Sciences*, 15, 2205–2227, URL https://www.doi.org/10.5194/hess-15-2205-2011.

Westra, S., M. Thyer, M. Leonard, D. Kavetski, and M. Lambert (2014), "A strategy for diagnosing and interpreting hydrological model nonstationarity." *Water Resources Research*, 50, 5090–5113, URL https://www.doi.org/10.1002/2013wr014719.

Winsemius, H. C., H. H. G. Savenije, and W. G. M. Bastiaanssen (2008), "Constraining model parameters on remotely sensed evaporation: justification for distribution in ungauged basins?" *Hydrology and Earth System Sciences*, 12, 1403–1413, URL https://www.doi.org/10.5194/hess-12-1403-2008.

Winsemius, H. C., B. Schaefli, A. Montanari, and H. H. G. Savenije (2009), "On the calibration of hydrological models in ungauged basins: A framework for integrating hard and soft hydrological information." *Water Resources Research*, 45, URL https://www.doi.org/10.1029/2009wr007706.

Wirth, S. B., C. Carlier, F. Cochand, D. Hunkeler, and P. Brunner (2020), "Lithological and tectonic control on groundwater contribution to stream discharge during low-flow conditions." *Water*, 12, URL https://doi.org/10.3390/w12030821.

Wooldridge, S., J. Kalma, and G. Kuczera (2001), "Parameterisation of a simple semi-distributed model for assessing the impact of land-use on hydrologic response." *Journal of Hydrology*, 254, 16–32, URL https://www.doi.org/10.1016/S0022-1694(01)00489-9.

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

Yadav, M., T. Wagener, and H. Gupta (2007), "Regionalization of constraints on expected watershed response behavior for improved predictions in ungauged basins." *Advances in Water Resources*, 30, 1756–1774, URL https://www.doi.org/10.1016/j.advwatres.2007.01.005.

Young, P. (1998), "Data-based mechanistic modelling of environmental, ecological, economic and engineering systems." *Environmental Modelling & Software*, 13, 105–122, URL https://www.doi.org/10.1016/S1364-8152(98)00011-5.

Young, P. C (2000), "Stochastic, dynamic modelling and signal processing: time variable and state dependent parameter estimation." *Nonlinear and nonstationary signal processing*, 74–114.

Young, P. C., W. Tych, and C. J. Taylor (2009), "The captain toolbox for matlab." *IFAC Proceedings Volumes*, 42, 758–763, URL https://www.doi.org/10.3182/20090706-3-FR-2004.00126.

Yu, P.-S. and T.-C. Yang (2000), "Using synthetic flow duration curves for rainfall–runoff model calibration at ungauged sites." *Hydrological Processes*, 14, 117–133, URL https://www.doi.org/10.1002/(sici)1099-1085(200001)14:1<117::Aid-hyp914>3.0.Co;2-q.

Zhang, Y. and F. H. S. Chiew (2009), "Relative merits of different methods for runoff predictions in ungauged catchments." *Water Resources Research*, 45, URL https://www.doi.org/10.1029/2008wr007504.