Accepted ms – Hydrogeology J. – doi: 10.1007/s10040-020-02176-0

An agile and parsimonious approach to data management in groundwater science using open-source resources

Giovanna De Filippis[1*], Stefania Stevenazzi[2], Corrado Camera[2], Daniele Pedretti[2], Marco Masetti[2]

1. Milano, Italy; email: giovanna.df1989@libero.it

2. Università degli Studi di Milano, Via L. Mangiagalli 34, 20129 Milano (Italy)

*corresponding author – displayed in the article as the designated author to accept queries from readers

NOTE TO COPYEDITOR – PLEASE INSERT THE FOLLOWING AS A FIRST-PAGE FOOTNOTE:

This article was prepared in collaboration with the Early Career Hydrogeologists' Network (ECHN) of the Italian Chapter of the International Association of Hydrogeologists (IAH)

Abstract

National governments and international organizations, such as the European Commission, are promoting the increased use of information and communication technologies to assist scientists in the "mining" of knowledge. A typical workflow for a hydrogeologist consists of investigating and reporting hydrogeological processes in a study area, from data collection to model-based analysis. While hydrogeologists may feel insufficiently skilled to undertake the self-automatizing and digitizing process, the digitalization of the above-mentioned workflow can be easily obtained by means of agile and parsimonious methodologies based on free and open-source software, and by using existing standards. This route is demonstrated for the digitalization of a vadose-zone monitoring system, where a large number of raw data related to water infiltration through the vadose zone are collected. The main aspects of the proposed methodology are a structured database (DB) where field data are stored, and a Python script to manage and process the available data. The structured DB was designed to store data recorded by field sensors and to generate inputs to run a transfer-function-based model to simulate percolation to the water table. Field data and model outputs were also exploited to automatically generate summary reports, like plots and table statistics. The proposed methodology can be generalized to other hydrogeological processes and case studies, as it is based on commonly available standards, basic knowledge of data-storage and data-management, and elementary programming skills to connect the different components of its suite.

Keywords

Groundwater management · Groundwater monitoring · Groundwater modelling · Computer science · Free and Open Source Software

1. Introduction

During the last decades, scientists have been experiencing an exponential growth of collected data, due to the rapid rise in use of digital technologies. Management of large volumes of data requires efficient system automation in all kinds of applications, from industrial engineering to web-based services for smart cities, from building automation systems to robotics (e.g., Xiao and Fan 2014; Androniceanu 2019; Biffl et al. 2019). The field of geosciences is not an exception. This is mostly due to the advancements in data acquisition and transmission from ground sensors, remote sensing and satellite technologies. Besides raw data (i.e., field data), large amounts of model-derived data are also available (e.g., data about climate projections). In some cases, such data are shared by means of public data repositories (Essawy et al. 2016). Because of the increased complexity and resolution of the available datasets, there is a growing need for improved data analytics tools, in support of understanding geo-environmental processes (Guru et al. 2009). Information and communication technologies (ICTs) may assist geoscientists in "mining the knowledge", i.e., fully exploit information contained in the available data by means of computer science and programming (Babovic 2005).

Policies and standards to address integrated water resources management strategies have been formalized by governmental and non-governmental organizations, including the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI 2020a), the International Network of Basin Organizations (INBO 2018), and the

United Nations Educational, Scientific and Cultural Organization (UNESCO 2019)). In general, standards and protocols aim to achieve the integration of vast amounts of data collected from many different sources, while rendering the accumulated data easily shared among different users (Fletcher and Deletic 2007).

In hydrogeology, automatizing data processing is of outmost importance when dealing with large datasets, including spatially extended, heterogeneous, high frequency variables, such as those coming from satellite analysis (e.g., Brodaric et al. 2018) or from vadose-zone hydrology studies (e.g., Sophocleous 2002; Skolasińska 2006; Pedretti et al. 2011; Blackmore et al. 2018). ICT tools may facilitate the fulfillment of a digital workflow usually carried out by a hydrogeologist to investigate a physical process in a study area. A simple and general workflow may consist of multiple steps, including (1) data collection and pre-processing, (2) data sharing and publication, (3) data analysis and post-processing, and (4) reporting. All the workflow steps are strictly interconnected: for instance, any update in the available dataset or in its interpretation triggers enhancements in the conceptual and numerical models.

The authors contend that the hydrogeologist may feel insufficiently skilled from a computational perspective to digitalize water-related information on his/her own. On one hand, this suggestion could be justified by the fact that, traditionally, programming skills are necessary to interface data and models (e.g., Bakker 2014). On the other hand, the hydrogeologist may overestimate the resources required to create ICT-based tools, e.g., the estimated time to build up an efficient and reliable tool, or the estimated costs to purchase the needed tools.

The main objective of this paper is to show that the full digitalization of the hydrogeological workflow can be fulfilled by means of a lightweight and relatively agile deployment based on existing standards, open-source software and a straight-forward scripting-based approach. Such a simple parsimonious methodological approach for leveraging data can cover all the steps of the workflow, from data collection, to data analysis and hydrogeological modelling, to automated reporting. This bridges any gap likely to occur in the hydrogeological workflow, mostly due to data redundancy and lack of data harmonization.

To demonstrate the actual applicability of an agile and parsimonious digitalization methodology using a real-life problem, this paper presents an application to a simple case study focusing on the interpretation of infiltration through the vadose zone in a well-instrumented area in northern Italy. Here, multiple ground sensors and a weather sensor were installed to obtain time-series of hydrological and hydrogeological variables controlling the infiltration process, including soil moisture content and pressure at different depths and rainfall intensity. The data were used to run a simple transfer-function-based model to obtain a soil water balance.

2. Materials and methods

2.1 Four-steps workflow

A simple four-steps workflow was considered for descriptive purposes and to allow for generalization of the conclusions of this work, noticing however that other steps may be included in case of differently formulated, more complex workflows.

(1) *Data collection and pre-processing*. Accessing reliable, up-to-date and relevant data is of utmost importance when dealing with integrated water resources management (Patterson et al. 2017; INBO 2018). In most cases, such need is threatened by several issues: (a) production and use of data by different institutions, using different standards and tools for different purposes; (b) lack of protocols for data management and exchange, which makes data often sparse, non-homogeneous or non-comparable; (c) scarce financial and human resources to invest in data collection systems. Recently, advances in data logging technology have been made to assist field data collection. These advances involve equipment that is programmed and connected to sensors installed in the field. As an example, Ogilvy et al. (2009) developed the ALERT technology, which allows quasi real-time measurements of geoelectric, hydrologic and other properties. Such technology is based on the use of electrodes, permanently buried in shallow trenches or attached to borehole casing, which may be remotely interrogated by wireless telemetry to provide volumetric images of the subsurface. More recently, some researchers have developed automated systems with an Arduino

microcontroller for in-situ measurement of soil infiltration rate applied to single-ring (Di Prima 2015) or double-ring infiltrometers (Fatehnia et al. 2016). Also, Young et al. (2017) tested low-cost robots for remote surface data collection to support water balance computations. Furthermore, the use of standardized programming languages on small computing platforms represents an advancement in the field of software applications for data collection for mobile computers. These are developed within geographic information systems (GIS) frameworks to account for the spatial dimension of the data collection process (e.g., QField 2019). As an example, Vivoni et al. (2002) developed a prototype system, the ENVIT Field Notebook Data Collection System, to link real-time field data collected from mobile devices to a centralized data server located at a remote location. Such a link is guaranteed through wireless standards.

These advancements also have a role in building credibility of data use among stakeholders, by means of integrated information systems for Quality Assurance and Quality Control (QA/QC; Hudson et al. 1999; Refsgaard et al. 2010). In fact, a key role in data pre-processing is represented by QA/QC, including identification of missing data and errors caused by equipment malfunction, instrument drift, improper calibration, vandalism, or other causes (Fletcher and Deletic 2007). In some cases, raw data may not be used for defensible scientific analysis, unless they undergo a QA/QC process (e.g., Horsburgh et al. 2008). Documented experiences (e.g., Hudson et al. 1999) showed that the development and use of standards for quality system concepts, in conjunction with appropriate data management software, can result in increased data quality ("fitness for purpose") and significant cost reductions (e.g., by reducing the cost of rework).

(2) *Data sharing and publication*. This is of utmost importance to ensure reproducible science. As mentioned above, accessing data and information is often difficult because of the lack of harmonization in data formats, which prevents effective extraction of information for specific applications (Liu et al. 2016), and results in poor data exchange, the existence of sparse databases that are not freely accessible, and data redundancy (INBO 2018). These aspects often are exacerbated by the lack of an "*Internet of Water*" system (Patterson et al. 2017), i.e., a structured architecture involving (i) data producers collecting data, (ii) data hubs providing data structures according to privacy/security protocols for data use and exchange (e.g., Water Data Transfer Format; Walker et al. 2009; WaterML 2.0; OGC 2014), and (iii) data users.

In this framework, open repositories (e.g., HydroShare (Horsburgh et al. 2016; CUAHSI 2020b)) increase the availability of datasets to better support the long series of scientific data. Also, storing data in a coherent and logical structure (e.g., a database, DB) supported by a computing environment allows one to ensure validity and availability. Database management systems (DBMS), often integrated within GIS frameworks (Gogu et al. 2001), may serve this function, thereby avoiding data redundancy. Examples of application of geoDB and DBMS can be found in Georgiadis et al. (1970), Tipping (2002), Dìaz et al. (2008), Gao and Zhou (2008), Wu et al. (2008), Popielarczyk and Templin (2014). On a larger scale, WHYCOS (World Hydrological Cycle Observing System; WMO 2005) and WINS (Water Information Network System (IHP-WINS), UNESCO 2019) are examples of programmes dedicated to promoting and facilitating the collection, exchange, dissemination and use of water-related information, in order to reinforce international cooperation and promote the free exchange of water-related data.

(3) *Data analysis and post-processing*. This is needed to investigate the spatial and time dimensions of the involved physical processes, and to convert raw data into useful information for decision-making (Dìaz et al. 2008). Advanced ICT tools, like GIS and programming languages, may support this step through data querying and exploitation. Computational analysis has grown rapidly in the field of water management for data processing and quality analysis (Hutton et al. 2016). Furthermore, numerical modelling may serve for characterization or forecasting purposes, or to devise management strategies. According to the objectives and the scale of the investigation, a number of numerical models, either lumped or spatially-distributed, are available (Singh 2014).

The overarching goal of data analysis is the interpretation of the involved physical processes. The first fundamental result of data analysis consists of the development of a hydrogeological conceptual model. The quality of a conceptual model affects the reliability of any decision. Any gap in the conceptual comprehension of the hydrogeological behavior of a system may pose serious issues while setting up a numerical model for management purposes. Also, any inconsistency in the hydrogeological conceptual model puts in crisis the interpretation of the available data. In this context, a numerical model can provide useful information on the inadequacy of data analysis and conceptual model (Foglia et al. 2007; Guillaume et al. 2016) and can help address the integration of the available datasets (Chen et al. 2012). This step also includes post-processing of the model's results, by means of mathematical methods, to quantify the uncertainty related to model input data and, as a consequence, to model outcomes as well (Linde et al. 2017).

(4) *Reporting*. This is of utmost importance, either if the objective of the investigation is to make an advancement in the knowledge of a hydrogeological system, or to support a decision-making process.

2.2 Components of the workflow

The proposed agile and parsimonious methodology is based on the use of commonly available devices for data collection and requires basic knowledge about data storage and management, and basic programming skills to connect all the components. To provide an example of its application, an idealized problem based on the general four-steps workflow is addressed. It is noted however that the approach is general and any additional phases may be easily added. Once the workflow is established, scripting to digitalize all steps is needed. The specific components forming the idealized workflow to be resolved are divided into two groups (Fig.1), "Data collection and pre-processing" and "Data storage and exploitation", corresponding to the steps of the workflow reported in section '*Four-steps workflow*'.

“Data collection and pre-processing” embeds actions related to data acquisition in the field and QA/QC. Data can be of any type, either related to groundwater or surface-water quantity, to physical/chemical status of the saturated/unsaturated zone (e.g., pH, electrical conductivity, dissolved oxygen), or to climate conditions (e.g., rainfall rate, air temperature). Data may be recorded by a data logger with any time frequency, according to the scope of the investigation. This work considered sensors installed in the field for data acquisition with a 10-minute frequency, a data logger installed in the field communicating with the sensors and saving data, and an R (R Core Team 2013) script to periodically download data in text file formats and to assign QA/QC flags.

“Data storage and exploitation” includes a structured DB and a scripting tool. The DB is where field data are long-term stored. It is constantly updated as soon as new data are downloaded. Scripting based on Python language (van Rossum and de Boer 1991) was considered, which offers a free and versatile means to: (i) transfer the downloaded data from text files to the DB; (ii) query the DB; (iii) provide a graphical user interface (GUI) for preparing files needed to run a numerical model; (iv) produce a report in pdf format containing some statistics about the field data and results of the numerical model.

The structured DB may be conceived adopting any DBMS, and the structure of such DB may be adapted to the amount of data available (or likely to be acquired), and to the specific objective of the investigation. A DB may store data of any type, e.g.:

- literature data;
- data derived from laboratory analysis;
- data collected in the field by means of analogic instruments (e.g., groundwater level data measured through freatimeters in wells/piezometers);
- data acquired by means of sensors and probes installed in the field (e.g., rainfall gauge stations, divers).

The structure of the DB should at least include:

- a table with the basic information related to the measurement points/devices (i.e., wells, piezometers, sensors, etc.). Information may include the identifier (ID) assigned to the measurement points/devices, coordinates of their location, the time frequency of the measurement, and any other information specifically related to the measurement points/devices and their installation;
- a table with values of a certain variable measured by a certain measurement point/device on a certain date and time and a QA/QC flag column. In a relational DB, such a table is connected to the above-mentioned one by means of a primary key field, which usually contains the ID assigned to the measurement points/devices.

The structured DB may then be queried, and the data stored within it may be used for statistics and analysis, according to the objectives of the investigation. Data may also be used for feeding numerical models. In this regard, the Python script allows one to retrieve data from the DB and use them to write model files needed to run a certain simulation code. Similarly, the Python script allows one to retrieve data from the DB and use them to produce a report with plots and statistics about the data queried. Python libraries are available to serve this function. Among these, Matplotlib (Hunter 2007) allows one to manage plotting, while Reportlab (ReportLab 2019) allows one to produce documents in pdf format.

Connecting the above components is relatively inexpensive in terms of costs and scientific effort. Indeed, costs only derive from installation and maintenance of sensor devices and data loggers, while running the R and Python scripts requires installation of free and open source tools. Scientific effort is mostly related to: (a) programming the R script for data download in a text file format and for management of the acquisition system, by means of QA/QC flags in case of malfunctioning; (b) designing the DB structure; (c) programming the Python script to read data from a text file and transfer them to the DB, to query the DB and to exploit data for feeding models and for automatic reporting.

3. Application

This study demonstrated the utility of the proposed agile and parsimonious approach for the management of the hydrogeological data collected from a vadose-zone monitoring station located close to the city of Milan (northern Italy). The case study provides a useful opportunity to test the proposed solution, as vadose zone hydrology is notoriously characterized by a large number of parameters and strong nonlinearity of the governing flow equations (e.g., Richards 1931).

The monitoring site is located in the plain surrounding Pozzuolo Martesana (20 km east of Milan, Fig.2a). The site covers a green (grass) area of approximately 6x6 m^2 and is located within a well field managed by the local water authority (CAP Holding S.p.A.). The geological framework involves a sandy soil with a thickness of about 1.5 m, above Quaternary gravelly-sandy deposits, which host the shallow aquifer. The average groundwater depth below the ground surface is about 6 m, with seasonal changes of ±1 m. The base of the aquifer, represented by a continuous clay layer, is found at about 30 m below the ground surface (Fig.2b).

The monitoring site was equipped to collect data on ever-changing dynamics of infiltration rate in the vadose zone, to relate them with the rainfall pattern and to validate the soil-water characteristic curve of the local soil, initially derived from pedo-transfer functions. Moreover, through the integration of information from the saturated zone and from air and rainfall chemistry, the monitoring site is used to support the development of an unsaturated-saturated flow numerical model to estimate recharge rates of the shallow aquifer. Ultimately, the site shall provide information on the air-rainfall-infiltration-groundwater path of nitrogen, in order to target the topic of vulnerability of the Po Plain aquifers to nitrate contamination (Masetti et al. 2009).

3.1 Field data collection and pre-processing

The site is instrumented with a compact weather sensor (WS, Lufft WS501-UMB) measuring air temperature, relative humidity, air pressure, wind speed, wind direction, and solar radiation. An automatic tipping bucket (0.2 mm) rain gauge (RG, Davis Rain Collector Vantage Pro2) complements the WS. The WS and the RG are installed on different poles, 2.0 m apart, at 2.0 m and 1.5 m above ground surface, respectively. To monitor water infiltration and redistribution processes in the topsoil layer, six soil moisture and temperature sensors (SMT, Truebner SMT100) and six tensiometers (Ts, Soilmoisture 2725ARL Jet Fill) equipped with pressure transducers (Soilmoisture 5302 Transducer) were installed. Soil moisture sensors were positioned along two vertical profiles at a depth of 20 cm, 40 cm and 80 cm below the ground surface. The two vertical profiles are located at a distance of 2.0 m from the WS pole to the south-east and to the south-west. Next to each profile, southward, a set of three Ts was installed. These were located at an approximate distance of 0.6 m from each other. Ideally, each set should have been made up of instruments at a depth of 20 cm, 40 cm, 80 cm below the ground surface, as for the soil moisture sensors. However, due to stones and pebbles present in the soil, it was not possible to dig holes of appropriate diameter (around 4 cm) below 40 cm. Therefore, the south-east set of Ts counts one instrument at 20 cm of depth and two instruments at 40 cm of depth. Conversely, the south-west set of Ts counts two instruments at 20 cm of depth and one instrument at 40 cm of depth.

Ts, WS, and RG are connected to a Campbell Scientific data logger, while the SMTs are attached to a Truebner TrueLog100. Data are acquired with a 10-minute frequency. An illustrative sketch of the monitoring site, including all the installed instruments, is presented in Fig.2c.

3.1.1 Quality Assurance and Quality Control (QA/QC)

QA/QC could include any kind of potential variable that could bias the quality of the data. Here, some of the key aspects that affect directly the test site's specific system were considered.

For QA, one essential item of data to retrieve is the voltage status of the instruments system in the field. If voltage falls below a certain threshold, the correct functioning of sensors and data loggers is jeopardized and data are not recorded, resulting in an array of missing values in the text files of field data. The threshold depends on the configuration of the acquisition system (e.g., type of sensor, type of data logger). Anyway, since the acquisition system is still in the testing phase, the voltage is recorded every 24 hours. As such, any intervention to restore any voltage issue is not immediate.

In order to overcome such limitation, the QC phase is adopted. It consists of checking the text files with field data recorded and notifying any occurrence of missing or untrusted values. This is done by assigning each data item recorded in the field a *quality check flag*, i.e., an integer value, which can assume three values, "0", "-1" or "1". Accordingly, "0" is assigned in case of missing data, "-1" in case of untrusted data (e.g., this occurs when the porous cup dries and the capillary rise drops to values close to zero with decreasing water content), or "1" in case data are correctly recorded and validated. Since the acquisition system is still in the testing phase, the QC procedure is executed manually by the field data manager. Anyway, an automatic assignment of a *quality check flag* is foreseen, in order to consolidate the recording procedure.

The *quality check flag* introduced in this phase is used in the subsequent steps (see the following sub-sections) to estimate the uncertainty related to the model's results. In this sense, the whole procedure is set to take into account the uncertainty associated to field data, which could be further processed.

3.2 Data storage and exploitation

This phase is entirely managed through a script written in Python3 programming language. The script is based on three main pillars, i.e., functions that allow one to:

- transfer sensor data from text files to a structured DB;
- provide a GUI for writing the model files needed to run the AquiMod modelling code (Mackay et al. 2014a) for simulating the infiltration process through the soil profile and the unsaturated zone;

- produce a pdf report with statistics and plots of the available data and results obtained after running the AquiMod software.

3.2.1 DB structure

The structure of the DB (Fig.3) has been conceived for storing data monitored by sensors in the field, as described in section '*Field data collection and pre-processing*'. Such structure has been set using the SpatiaLite DBMS (SpatiaLite Development Team 2011). With reference to Fig.3, the SpatiaLite DB includes the following tables:

- table *sensors*: this table lists all the sensors installed in the field. For each sensor, the following is reported: a name associated to that sensor, the type of that sensor, the depth of installation below the ground surface (if applicable), the time frequency adopted for data acquisition;
- table *sensor_type*: this table lists some types of sensors, reporting for each of them the measured variable with the related units of measurement. The content of the field reporting the type of sensor in the *sensors* table is retrieved from this table through a relation between the two tables involved;
- table *pore_pressure*: this table reports pore pressure values measured by tensiometers T1 through T8 on a certain date and time (one measurement every 10 minutes). For each measurement, the following is reported: the name of the tensiometer which recorded that value (retrieved from the *sensors* table), the date and time of the measurement, the measured value for pore pressure;

• table *unsat_props*: this table reports values for water content in the unsaturated zone, soil temperature and electric potential measured by SMT sensors, named TDR0 through TDR5, on a certain date and time (one measurement every 10 minutes). For each measurement, the following is reported: the name of the SMT sensor which recorded that value (retrieved from the *sensors* table), the date and time of the measurement, the measured value for water content in the unsaturated zone, soil temperature and electric potential;

• table *rainfall*: this table reports rainfall rate values measured by the RG station on a certain date and time (one measurement every 10 minutes). For each measurement, the following is reported: the name assigned to the RG station (retrieved from the *sensors* table), the date and time of the measurement, the measured value for rainfall rate;

• table *meteo_climate*: this table reports values for air temperature, relative humidity, atmospheric pressure, wind direction, wind velocity and solar radiation measured by the WS on a certain date and time (one measurement every 10 minutes). For each measurement, the following is reported: the name assigned to the WS (retrieved from the *sensors* table), the date and time of the measurement, the measured value for the above variables.

All tables include a quality check field devoted to recording any error that occurred during the automatic recording of data in the field. Such a quality check field may assume three integer values: "0" in case of non-recorded data (e.g., in case of voltage problems for the instruments), “-1” in case of untrusted data (e.g., if the porous cup dries), or “1” in case data are correctly recorded.

Fig.3 reports also the relations among the tables. Such relations are represented through arrows oriented from the table where a certain information is retrieved from, towards the table where that information is needed.

3.2.2 Data storage

The structured DB is a **.sqlite* file containing all the above-listed tables. A Python script was coded to read the text files containing the sensors data downloaded and to fill the above-listed tables with the information needed. A GUI was also designed using the Qt framework (Qt Company 2019; Fig.4). Such GUI requires the path of the SpatiaLite DB to be updated and the paths of the text files containing sensors data. Such paths may be retrieved using the *Browse…* buttons.

While the Python script reads the text files containing the sensors’ data in the due format, a check is performed on the date-and-time field of such files: if the measurement recorded on a certain date and time is already present in the due table of the SpatiaLite DB, the corresponding line of the text file is skipped. In this way, clicking *OK* in the GUI reported in Fig.4, the SpatiaLite DB may be periodically updated with new measurements. Fig.4 also reports the tables of the SpatiaLite DB, which are automatically updated after reading the corresponding text file.

3.2.3 Connecting the DB to a mathematical model

The data stored in the SpatiaLite DB were connected to a mathematical model for the simulation of groundwater recharge through the soil profile and the unsaturated zone. In the example of this study, the groundwater recharge process was simulated through transfer functions by running the AquiMod modelling code (Mackay et al. 2014a), although any other mathematical model (e.g., a process-based numerical model) could be connected to the DB. AquiMod was selected as a free and open source lumped model that simulates soil drainage, water flow through the unsaturated zone and groundwater flow. AquiMod can be applied to simulate groundwater level time-series at observation boreholes. Such levels can be calibrated against field data. AquiMod includes three sub-routines:

- a soil water balance module: rainfall is partitioned among crop evapotranspiration, runoff and drainage through the soil profile. The algorithm is a simplification of the one developed by Allen et al. (1998);
- an unsaturated-zone balance module: the soil drainage is attenuated through the unsaturated zone, using a two-parameter Weibull transfer function. As a result of this module, percolation to the water table is calculated;

• a saturated-zone balance module: percolation to the water table is input as a recharge term to the saturated portion of the aquifer. Groundwater storage and discharge are simulated, and a mass balance equation is solved to calculate groundwater level at observation boreholes over the length of the simulation.

The conceptualization of the AquiMod code fits perfectly with the objective of the case study, i.e., the need to understand the hydraulic behavior of the aquifer, in view of investigating the spreading of pollutants (e.g., nitrate) through the unsaturated zone. Furthermore, the extent of the study area makes the use of lumped codes particularly convenient, at least in a first attempt to achieve the objective above by adopting poorly parameterized modelling tools.

Regarding the time dimension of the simulated processes, AquiMod allows one to discretize the whole simulation into time-steps, whose length is input by the modeler, according to the scope of the investigation and to the time scale of the variation of the involved processes.

AquiMod can be run in *calibration* or *evaluation* mode. In *calibration* mode, a Monte Carlo parameter sampling algorithm is run using multiple unique parameter sets, in order to compare the simulated groundwater level time-series against observational data, and to evaluate the best fit by means of objective functions. In *evaluation* mode, fixed parameter sets are specified by the modeler and AquiMod is run to calculate groundwater level time-series for that specific parameter set. In both cases, the AquiMod source code needs model files, i.e., formatted text files containing information about the running settings.

Model files must have a specific structure. As such, a Python script was coded to retrieve and process data stored in the SpatiaLite DB, in order to generate the model files with the required structure. A GUI was also designed using the Qt framework (Fig.5). Such GUI requires the User to input the path of the SpatiaLite DB to be queried, and parameters and settings needed to run the AquiMod code.

The GUI reported in Fig.5 is made of five parts:

- in the upper part, the User must select the time-step length (*days* or *months* options are available), the starting and lasting dates of the simulation, and the simulation mode (*evaluation* or *calibration*). The time-step length and the starting and lasting dates of the simulation are used to define the time discretization of the model: the simulation starts from the starting date selected by the User and ends in the lasting date selected by the User. The starting and lasting dates of time-steps in the middle are determined according to the time-step length defined by the User. Data stored in the SpatiaLite DB are averaged (e.g., soil moisture, pore pressures, air temperature) or summed (rainfall) over the length of each time-step. While retrieving the data from the SpatiaLite DB, a check is made over the quality check fields of each table of the averaged variables (i.e., soil moisture, pore pressures, air temperature): if the content of the quality check field is "0" (meaning missing data) or "-1" (meaning untrusted data), a linear interpolation is made to fill the gap, using the last value recorded right before the gap and the first one recorded right after the gap. It must be noted that this interpolation procedure is not applicable for the rainfall values, which are summed over the length of each time-step. As such, a check is made internally to the Python script: if some gaps occur in the rainfall time-series (i.e., if the content of the quality check field is "0" or "-1"), then the User is asked to edit the starting and lasting dates in the GUI, in order to get a simulation period including trusted rainfall data only;

- the following three parts are related to the simulation of: the soil water balance module, the unsaturated-zone balance module and the saturated-zone balance module. These are checkable, meaning that defining the needed parameters for simulating each module is not mandatory. As such, if the *Soil profile* option is not checked, rainfall will be applied to the unsaturated zone as a whole. Similarly, if the *Unsaturated zone* option is not checked, the drainage term through the soil profile or the rainfall as a whole will be an input to the water table. Finally, if the *Saturated zone* option is not checked, the groundwater level will not be simulated. Parameters needed in the *Soil profile* section are soil and crop properties, to determine evapotranspiration and runoff rates. The Python script internally computes potential evapotranspiration required by the AquiMod algorithm by using the Thornthwaite equation (Thornthwaite 1948), exploiting the air temperature values stored in the SpatiaLite DB. Parameters needed in the *Unsaturated zone* section are related to the computation of the Weibull probability density function. Parameters needed in the *Saturated zone* section are related to the geometry and hydraulic properties of the aquifer.

For each of the required parameters, in case the model is run in *calibration* mode, the User may define minimum and maximum values to let parameters vary between such limits. Also, for each of these three sections, the User may define a *Parameters variation (%)* factor. This will be used in the Python script to internally vary each of the defined parameters by such percentage, in order to perform runs with different sets of parameters;

• the last part is related to the solver settings. Here the needed parameters are:

- the *Number of runs of the model*;
- the *Spin-up period*, i.e., the number of time-steps during which an initial wetting-up period of the soil profile and of the unsaturated zone occurs;
- the *Objective function* for evaluating the model outputs against the observed levels;
- the *Threshold for the objective function* that must be reached in *calibration* mode, for the parameter set of the simulation to be stored;
- the *Maximum number of acceptable models*, i.e., the maximum number of parameter sets that exceed the acceptable threshold that will be output when running in *calibration* mode.

After filling the GUI reported in Fig.5 and clicking *OK*, a sub-folder renamed *AquiMod_files* is created by the Python script within the folder where the **.sqlite* DB file is located. Such sub-folder is populated with sub-folders and model files containing information defined by the User through the GUI reported in Fig.5 and formatted as required by the AquiMod algorithm. A sub-folder renamed *Output* is also created within the *AquiMod_files* one. This will be populated with output files generated after running the AquiMod executable. The latter must be run independently from the GUI reported in Fig.5. To do so, the AquiMod executable, freely available for download from the AquiMod website, must be installed as detailed in the AquiMod User Manual (MacKay et al. 2014b), and run specifying the whole path of the *AquiMod_files* sub-folder.

The approach detailed in this sub-section allows one to connect a hydrological model on one side to a relational DB on the other, through a loose coupling strategy (Brimicombe 2003; Goodchild 1992; Nyerges 1991), where the two components are treated independently and interaction between them is managed through manually-enabled file exchange.

Due to the specific aims of the case study, the following limitations result from the application of the GUI (Fig.5):

- the Python script only implements the *Q1K1S1* component of AquiMod for the saturated zone (MacKay et al. 2014b), where the aquifer is represented as a single layer with a single discharge outlet. Also, the *α* parameter which determines the hydraulic conductivity variation with depth has been set to 1, meaning that the hydraulic conductivity increases with elevation;

- potential evapotranspiration is computed using the Thornthwaite equation, by exploiting the air temperature values stored in the SpatiaLite DB;
- different sets of model parameters may be defined just using the *Parameters variation (%)* factor;
- time-steps length is the same, as defined by the User, all over the simulation length.

Anyway, these limitations concern only the GUI shown in Fig.5. The User can manually edit the model files generated within the *AquiMod_files* sub-folder as needed.

3.2.4 Modelling results

Fig.6 reports results of the AquiMod application to the case study between February 4th, 2019 and June 10th, 2019. The model was run in *evaluation* mode and the three components (soil, unsaturated zone, saturated zone) were simulated. The simulation period was discretized into time-steps 15-days long and average values of climate data stored in the SpatiaLite DB were calculated internally to the Python script. No gaps occurred in the time-series of rainfall and climate data in the simulation period.

As a result of the soil component (Fig.6a), the following may be inferred about the water budget at the end of the simulation period, considering that the budget terms were averaged over 15-day time-steps. The total rainfall was worth about 290.0 mm. This was partitioned between superficial runoff (23.5 mm) and percolation to the unsaturated zone (54.8 mm), while a water deficit of about 29.4 mm occurred, which did not satisfy the vegetation need for evapotranspiration (226.8 mm). The water deficit term is calculated in AquiMod as the difference between the actual evapotranspiration from the soil profile and the rainfall.

The effective infiltration reaching the water table (i.e., the net aquifer recharge) is a fraction of the percolation. The effective infiltration results from the application of a Weibull-based transfer function, which simulates the attenuation of the soil drainage through the unsaturated zone. For this modeling exercise, the shape (μ) and scale (σ) parameters of the adopted Weibull distribution were set to μ = 6.0 and $\sigma =$ 3.0. Fig.6b reports the time plot of the effective infiltration with respect to rainfall. An effective infiltration of about 45.5 mm to the water table was simulated, resulting in decreasing groundwater head from 117.0 m to 115.8 m with respect to the mean sea level (Fig.6c).

3.2.5 Data analytics and reporting

The final step of the proposed methodology concerns the automated production of a report with statistics and plots of data and the models' results. To this end, a Python script was coded to generate a pdf report for the case study. Two Python libraries were used:

Matplotlib (Hunter 2007) for 2D plotting, and ReportLab (ReportLab 2009) for generating pdf files.

A GUI was also designed using the Qt framework (Fig.7). Such GUI requires:

- the path of the SpatiaLite DB to be queried;
- the logo to be printed in the first page of the pdf report. This may be retrieved using the *Browse…* button;
- the list of authors of the report, with the related institutions;
- an image of the study area. This may be retrieved using the *Browse…* button.

After clicking *OK* in the GUI reported in Fig.7, the Python script:

- queries the different tables of the SpatiaLite DB;
- computes statistics (i.e., minimum, maximum and average values) about rainfall rate, air temperature, pore pressure and soil water content over the monitoring period;
- produces time plots of the above variables over the monitoring period;
- accesses the AquiMod model files and the output files produced after running the AquiMod executable;
- produces plots of (a) the water budget of the soil profile, (b) the water budget of the unsaturated zone, and (c) the groundwater level simulated in the study area, over the simulation period (Fig.6).

All the plots and statistics listed above are printed in a **.pdf* file generated and saved in a sub-folder renamed *pdf_report* within the folder where the **.sqlite* DB file is located. The plots produced are also saved as **.png* files in a sub-folder renamed *plots* within the *pdf_report* sub-folder.

The generated pdf report includes:

- a first page displaying the logo selected through the GUI, a title of the report internally specified in the Python script, the list of authors and institutions defined through the GUI, and the current date internally specified in the Python script;
- a section reporting a description of the study area, including the image selected through the GUI, and of the objectives of the investigation;
- a part about the analysis of the sensors' data. This includes:
  - statistics and time plots of rainfall and air temperature over the monitoring period;
  - statistics of pore pressure and a composite time plot of pore pressure and rainfall values over the monitoring period;
  - statistics and time plots of water content values over the monitoring period;
- a part about the AquiMod model. This includes:
  - a general description of model settings, including a table with time discretization;
  - a description and a bar chart with the water balance of the soil profile (Fig.6a);

- a description and a composite time plot of the effective infiltration and rainfall through the unsaturated zone over the simulation period (Fig.6b);
- a description and a time plot of the groundwater level simulated over the simulation period (Fig.6c);
- a plot which compares the groundwater level simulated and the groundwater level measured over the simulation period. Such plot is complemented by an analysis of residuals. This part is included in the pdf report in case the model is run in *calibration* mode. This is not the case of this example, but the structured DB is set to include also a table renamed *gw_level*, with the same general structure as the tables described above (Fig.3), and a field containing groundwater level data. If the model is run in *calibration* mode, groundwater level data are averaged over the length of each time-step, in order to perform the comparison with the simulated values. Also, if missing or untrusted groundwater level data occur in the *gw_level* table, a linear interpolation is performed internally to the Python script to fill the gaps;
- a post-processing section, where a preliminary evaluation of the error introduced in the model is made, in cases of missing or untrusted climate data. Should this be the case, such error is estimated by calculating, for each time-step, the percentage of records in the *meteo_climate* table where the content of the quality check field is "0" or "-1".

Model settings are retrieved from the AquiMod model files generated in the *AquiMod_files* sub-folder. Similarly, the different components of the water balance of the soil profile, and time-series of the effective infiltration through the unsaturated zone and of the simulated groundwater level are retrieved from output files generated after running the AquiMod executable independently of the Python GUI.

Fig.8 shows two extracts of the pdf report produced. The whole report is presented in the electronic supplementary material (ESM).

The Python script is coded so that the User may easily modify the content of the pdf report (i.e., sentences, sections ordering). Since Python is an open source programming language, this can be done by simply editing a Python file with any text editor.

Even if the selected time-series to test the approach and to run the AquiMod analysis were complete and not affected by errors, this study finally addressed a hypothetical application in which some gaps occur in the time-series of air temperature data to showcase the ability of the proposed approach to account for QA/QC analysis.

Identical model settings and input data were adopted, as the ones described in the previous sections (i.e., model running between February 4th, 2019 and June 10th, 2019, with time-steps 15 days long, and the same parameter values set for the soil, unsaturated zone and saturated zone components, and for the solver). The time-series of the reference air temperature data were then modified, assuming a first gap due to "missing data" lasting two days, on February 10th-12th, 2019 (during time-step 1), and a second gap due to "untrusted data" lasting five days, on March 20th-25th, 2019 (during time-steps 3 and 4; Fig.9a).

A framed text including a printout of the percentage of missing/untrusted data over the length of each time-step where the above gaps occur is included in the last section of the generated pdf report (Fig.9b). The main purpose of this printed percentage is to warn the reader about the magnitude of the potential errors affecting the water budget calculations, providing a first-cut and preliminary evaluation of the uncertainty that may affect the model's results. The modeler is required to run a thorough post-processing analysis and to develop more robust uncertainty indicators to assess and quantify the types of errors that can affect the water budget calculation.

4. Conclusions

The workflow carried out by a hydrogeologist may consist of multiple steps, from data collection to data exploitation and analysis, from the definition of a hydrogeological conceptual model to mathematical modelling and reporting the results of the investigation. As governments and administrations are increasingly fostering data digitalization and ICTs, developing a fully-digital workflow requires integrating the expert knowledge of a groundwater scientist with computing and programming skills.

The present work demonstrates that simple and inexpensive tools may be applied to create an agile and parsimonious methodology that connects multiple parts of the workflow. The proposed approach is based on existing data management standards and free and open source software tools, like DBMS and Python programming language.

Basic programming skills and a wise use of existing standards and open source resources can be applied to:

- harmonize data, usually available in different formats and from different sources, by means of DBMS, thus avoiding data redundancy;
- easily query and manage data, by means of data analytics tools for computing statistics and plotting. The automation of data analysis by means of script coding allows one to rapidly update statistics and plots as soon as new data are available, and to timely highlight any inconsistency in the time/space trends of the collected data;
- facilitate the interaction between the available data and numerical models, as script coding allows one to retrieve data from a structured DB and to organize them in model files with specific formats, as required by the selected simulation model;
- automate and customize the reporting step, generating data plots, statistics and the models' results, according to the objectives of the investigation. This also results in facilitating the timely update of a report as soon as new data are available;
- provide a preliminary estimate of the uncertainty that affects the models' results, based on the calculation of the percentage of

missing/untrusted data occurring in the time-series of climate data. The proposed approach was demonstrated by applying it to a simple, real-world case study, with the aim of investigating the infiltration mechanism through the unsaturated zone.

The Python programming language for script coding can interact with libraries suited for designing GUIs (e.g., the Qt framework). As such, the source code can be used straightforwardly. This guarantees the portability of the proposed methodology, as it is based on existing data management standards and on the application of free and open source tools. This means that the suite of the methodology (i.e., the structured DB where data are stored, and the Python script with its GUI) may be easily shared and used by water managers and experts of the water sector. The proposed approach may be easily reproduced and adapted to any case study, as it is based on the use of commonly available devices for data collection, and it requires basic knowledge about the commonly available standards for data storage and management, and basic programming skills to connect the different components of its suite.

Acknowledgements

The content of this article uses materials of a short course on the hydrogeology of the vadose zone, which was held on 10th - 11th June 2019 in Milan (Italy), within the framework of the "Flowpath – National Meeting on Hydrogeology" Italian congress, organized every two years by the Italian Chapter of the International Association of Hydrogeologists (IAH-Italy). The short course was organized by the Italian Chapter of the Early Career Hydrogeologists' Network (ECHN-Italy), and lectures were given by ECHN-Italy members and by researchers of the Department of Earth Sciences of the University of Milan. The whole suite described in this paper was tested by young Italian hydrogeologists attending the short course. CAP Holding S.p.A., the water authority managing the well field installed in the monitoring site, is kindly acknowledged.

Code availability

The source code of the Python script developed for data management and processing is freely available through the following GitHub repository: https://github.com/gdefilippis/pozzuolo_martesana (GitHub Inc (2020)).

The pdf report related to the application reported in this paper is freely available for download through the following link: https://www.dropbox.com/s/gz3qwo9pxv27mkf/pozzuolo_martesana.pdf?dl=0.

References

Allen RG, Pereira LS, Raes D, Smith M (1998) Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56. Food and Agriculture Organization of the United Nations.

Androniceanu A (2019) The social sustainability of smart cities: urban technological innovation, big data management, and the cognitive internet of things. Geopolitics, History, and International Relations 11(2):110–115.

Babovic V (2005) Data mining in hydrology. Hydrological Processes: An International Journal 19(7):1511-1515.

Bakker M (2014) Python scripting: The return to programming. Groundwater 52(6):821-822. https://doi.org/10.1111/gwat.12269

Biffl S, Lüder A, Rinker F, Waltersdorfer L, Winkler D (2019) Engineering Data Logistics for Agile Automation Systems Engineering. In: Biffl S, Eckhart M, Lüder A and Weippl E (eds) Security and Quality in Cyber-Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb. Springer International Publishing: Cham, 187–225.

Blackmore S, Pedretti D, Mayer KU, Smith L, Beckie RD (2018) Evaluation of single-and dual-porosity models for reproducing the release of external and internal tracers from heterogeneous waste-rock piles. Journal of contaminant hydrology 214:65-74. https://doi.org/10.1016/j.jconhyd.2018.05.007

Brimicombe A (2003) GIS, Environmental Modelling and Engineering. Taylor and Francis, London.

Brodaric B, Boisvert E, Chery L, Dahlhaus P, Grellet S, Kmoch A, Létourneau F, Lucido J, Simons B, Wagner B (2018) Enabling global exchange of groundwater data: GroundWaterML2 (GWML2). Hydrogeology Journal 26(3):733-741. https://doi.org/10.1007/s10040-018-1747-9

Chen Q, Wu W, Blanckaert K, Ma J, Huang G (2012) Optimization of water quality monitoring network in a large river by combining measurements, a numerical model and matter-element analyses. Journal of environmental management 110:116-124.

CUAHSI (2020a) Data & Models. Universities Allied for Water Research. https://www.cuahsi.org/data-models. Accessed 2019

CUAHSI (2020b) HydroShare. https://www.hydroshare.org/. Accessed 2019

Díaz L, Granell C, Gould M (2008) Case Study: Geospatial Processing Services for Web based Hydrological Applications.

Di Prima S (2015) Automated single ring infiltrometer with a low-cost microcontroller circuit. Computers and Electronics in Agriculture 118: 390-395.

Essawy BT, Goodall JL, Xu H, Rajasekar A, Myers JD, Kugler TA, Mirza MB, Whitton MC, Moore RW (2016) Server-side workflow execution using data grid technology for reproducible analyses of data-intensive hydrologic systems. Earth and Space Science 3(4):163-175.

Fatehnia M, Paran S, Kish S, Tawfiq K (2016) Automating double ring infiltrometer with an Arduino microcontroller. Geoderma 262:133-139.

Fletcher T, Deletic A (2007) Data Requirements for Integrated Urban Water Management: Urban Water Series-UNESCO-IHP. CRC Press.

Foglia L, Mehl SW, Hill MC, Perona P, Burlando P (2007) Testing alternative ground water models using cross-validation and other methods. Groundwater 45(5):627-641.

Gao Y, Zhou W (2008) Advances and challenges of GIS and DBMS applications in karst.

Georgiadis N, Sidiropoulos E, Tolikas P (1970) Organising information related to groundwater hydrology. WIT Transactions on Ecology and the Environment 8.

GitHub Inc (2020) Pozzuolo Martesana plugin, version 0.2. https://github.com/gdefilippis/pozzuolo_martesana. Accessed 2019

Gogu R, Carabin G, Hallet V, Peters V, Dassargues A (2001) GIS-based hydrogeological databases and groundwater modelling. Hydrogeology Journal 9(6):555-569.

Goodchild M (1992) Integrating GIS and spatial data analysis: problems and possibilities. Int. J. Geogr. Inf. Syst. 6(5):407–423.

Guillaume JH, Hunt RJ, Comunian A, Blakers RS, Fu B (2016) Methods for exploring uncertainty in groundwater management predictions. In Integrated Groundwater Management (pp. 711-737). Springer, Cham.

Guru SM, Kearney M, Fitch P, Peters C (2009) Challenges in using scientific workflow tools in the hydrology domain. 18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009.

Horsburgh JS, Tarboton DG, Maidment DR, Zaslavsky I (2008) A relational model for environmental and water resources data. Water Resources Research 44(5).

Horsburgh JS, Morsy MM, Castronova AM, Goodall JL, Gan T, Yi H, Stealey MJ, Tarboton DG (2016) Hydroshare: Sharing diverse environmental data types and models as social objects with application to the hydrology domain. JAWRA Journal of the American Water Resources Association 52(4):873-889

Hudson HR, McMILLAN DA, Pearson CP (1999) Quality assurance in hydrological measurement. Hydrological Sciences Journal 44(5):825-834. https://doi.org/10.1080/02626669909492276

Hunter JD (2007) Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9(3):90-95.

Hutton C, Wagener T, Freer J, Han D, Duffy C, Arheimer B (2016) Most computational hydrology is not reproducible, so is it really science? Water Resources Research 52(10):7548-7555.

INBO (International Network of Basin Organizations; 2018) The handbook on water information system – Administration, processing and exploitation of water-related data. ISBN: 978-2-9563656-0-0.

Linde N, Ginsbourger D, Irving J, Nobile F, Doucet A (2017) On uncertainty quantification in hydrogeology and hydrogeophysics. Advances in Water Resources 110:166-181.

Liu H, van Oosterom P, Hu C, Wang W (2016) Managing large multidimensional array hydrologic datasets: a case study comparing NetCDF and SciDB. Procedia Engineering 154:207-214.

Mackay JD, Jackson CR, Wang L (2014a) A lumped conceptual model to simulate groundwater level time-series. Environmental Modelling and Software 61:229-245.

Mackay JD, Jackson CR, Wang L (2014b) AquiMod user manual (v1. 0).

Masetti M, Sterlacchini S, Ballabio C, Sorichetta A, Poli S (2009) Influence of threshold value in the use of statistical methods for groundwater vulnerability assessment. Science of the total environment 407(12):3836-3846. https://doi.org/DOI: 10.1016/j.scitotenv.2009.01.055

Nyerges T (1991) GIS for environmental modellers: an overview. In: First International Conference/Workshop on Integrating GIS and Environmental Modeling. NCGIA, Boulder.

OGC (Open Geospatial Consortium; 2014). OGC WaterML. https://www.opengeospatial.org/standards/waterml. Accessed 2019.

Ogilvy RD, Meldrum PI, Kuras O, Wilkinson PB, Chambers JE, Sen M, Pulido-Bosch A, Gisbert J, Jorreto S, Frances I, Tsourlos P (2009)

Automated monitoring of coastal aquifers with electrical resistivity tomography. Near Surface Geophysics 7(5-6):367-376.

Patterson L, Doyle M, King K, Monsma D (2017) Internet of water: Sharing and integrating water data for sustainability. The Aspen Institute, Washington, DC.

Pedretti D, Fernàndez-Garcia D, Sanchez-Vila X, Barahona-Palomo M, Bolster D (2011) Combining physical-based models and satellite images for the spatio-temporal assessment of soil infiltration capacity. Stochastic environmental research and risk assessment 25(8):1065-1075. https://doi.org/10.1007/s00477-011-0486-4

Popielarczyk D, Templin T (2014) Application of integrated GNSS/hydroacoustic measurements and GIS geodatabase models for bottom analysis of Lake Hancza: the deepest inland reservoir in Poland. Pure and Applied Geophysics 171(6):997-1011.

QField (2019) QField Project Management, https://qfield.org/docs/project-management/. Accessed 2019

Qt Company (2019) https://www.qt.io/. Accessed 2019

R Core Team (2013) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

Refsgaard JC, Højberg AL, Møller I, Hansen M, Søndergaard V (2010) Groundwater modeling in integrated water resources management—visions for 2020. Groundwater 48(5):633-648.

ReportLab (2019) https://www.reportlab.com/. Accessed 2019

Richards LA (1931) Capillary conduction of liquids through porous mediums. Physics 1(5):318-333.

Singh A (2014) Groundwater resources management through the applications of simulation modeling: a review. Science of the Total Environment 499:414-423.

Skolasińska K (2006) Clogging microstructures in the vadose zone—laboratory and field studies. Hydrogeology Journal 14(6):1005-1017. https://doi.org/10.1007/s10040-006-0027-2

Sophocleous M (2002) Interactions between groundwater and surface water: the state of the science. Hydrogeology Journal 10(1):52-67. https://doi.org/10.1007/s10040-001-0170-8

SpatiaLite Development Team (2011) The Gaia-SINS federated projects home-page. http://www.gaia-gis.it/gaia-sins/. Accessed 2019

Thornthwaite CW (1948) An approach toward a rational classification of climate. Geographical review 38(1):55-94.

Tipping RG (2002) The development of a karst feature database for southeastern Minnesota. Journal of Cave and Karst Studies 51.

UNESCO (2019) The Water Information Network System (IHP-WINS). https://en.unesco.org/ihp-wins. Accessed 2019

van Rossum G, de Boer J (1991) Interactively Testing Remote Servers Using the Python Programming Language. CWI Quarterly 4(4):283–303.

Vivoni ER, Camilli R, Rodriguez MA, Sheehan DD, Entekhabi D (2002) Development of mobile computing applications for hydraulics and water quality field measurements. WIT Transactions on Ecology and the Environment 52.

Walker G, Taylor P, Cox S, Sheahan P (2009) Water Data Transfer Format (WDTF): Guiding principles, technical challenges and the future. In Proc. 18th World IMACS Congress and MODSIM09 Int. Congress on Modelling and Simulation, pp. 4381-4387.

WMO (World Meteorological Organization; 2005) Hydrological information systems for integrated water resources management - WHYCOS Guidelines for development, implementation and governance. WMO/TD-No. 1282.

Wu Q, Xu H, Zhou W (2008) Development of a 3D GIS and its application to karst areas. Environmental geology 54(5):1037-1045.

Xiao F, Fan C (2014) Data mining in building automation system for improving building operational performance. Energy and Buildings 75:109–118. https://doi.org/10.1016/j.enbuild.2014.02.005

Young S, Peschel J, Penny G, Thompson S, Srinivasan V (2017) Robot-assisted measurement for hydrologic understanding in data sparse regions. Water 9(7):494.

FIGURE CAPTIONS:

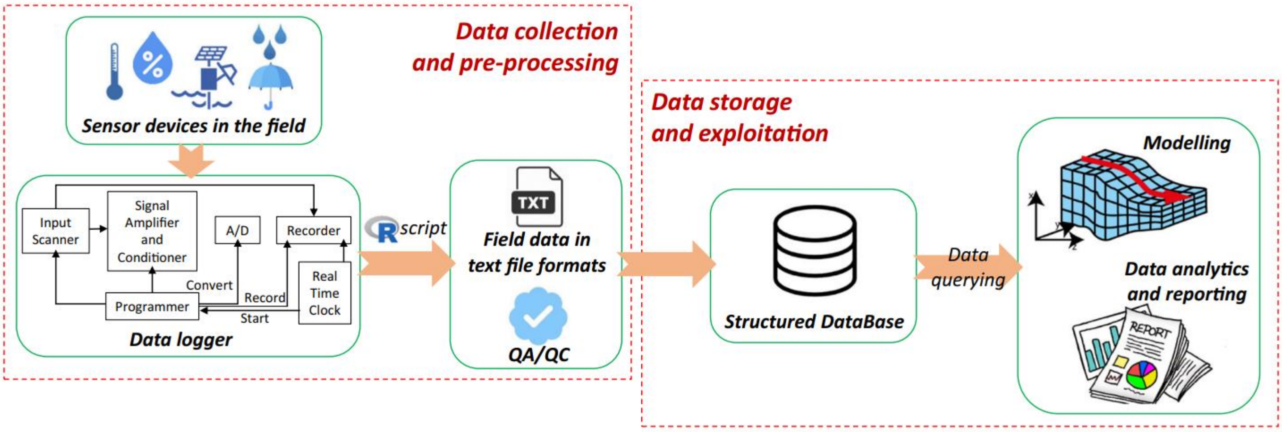

Data collection and pre-processing
Sensor devices in the field
Input Scanner
Signal Amplifier and Conditioner
A/D
Recorder
Convert
Record
Real Time Clock
Programmer
Start
Data logger
R script
TXT
Field data in text file formats
QA/QC
Data storage and exploitation
Structured DataBase
Data querying
Modelling
Data analytics and reporting
REPORT

Fig. 1 Components of the proposed agile and parsimonious methodology

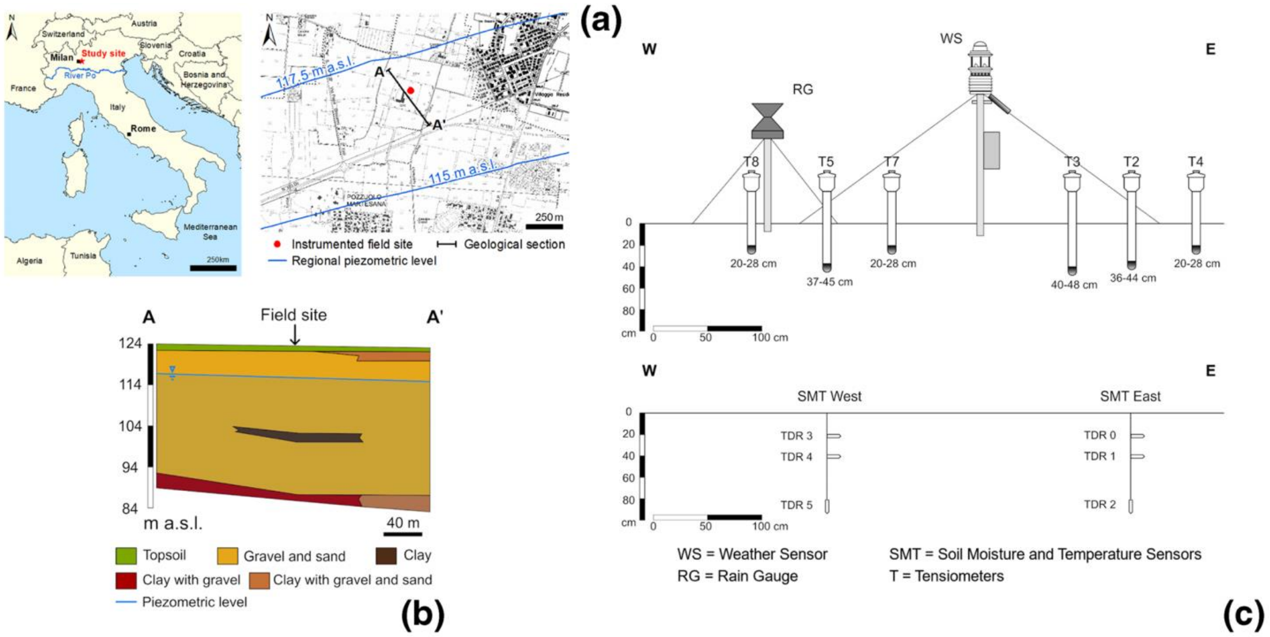


Fig. 2 a) Topographical map of the study area; b) simplified geological section; c) illustrative sketch of the monitoring site; depths indicated next to each tensiometer represent the exact location of the respective porous cup (length = 9 cm)

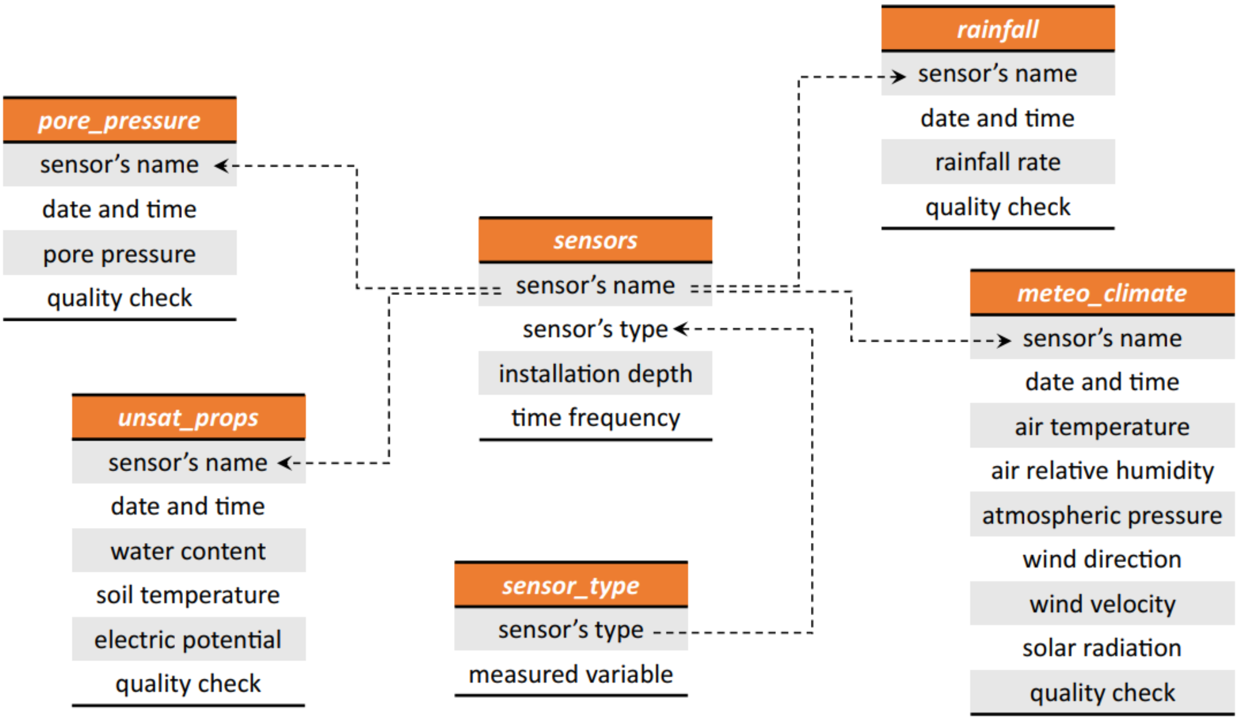
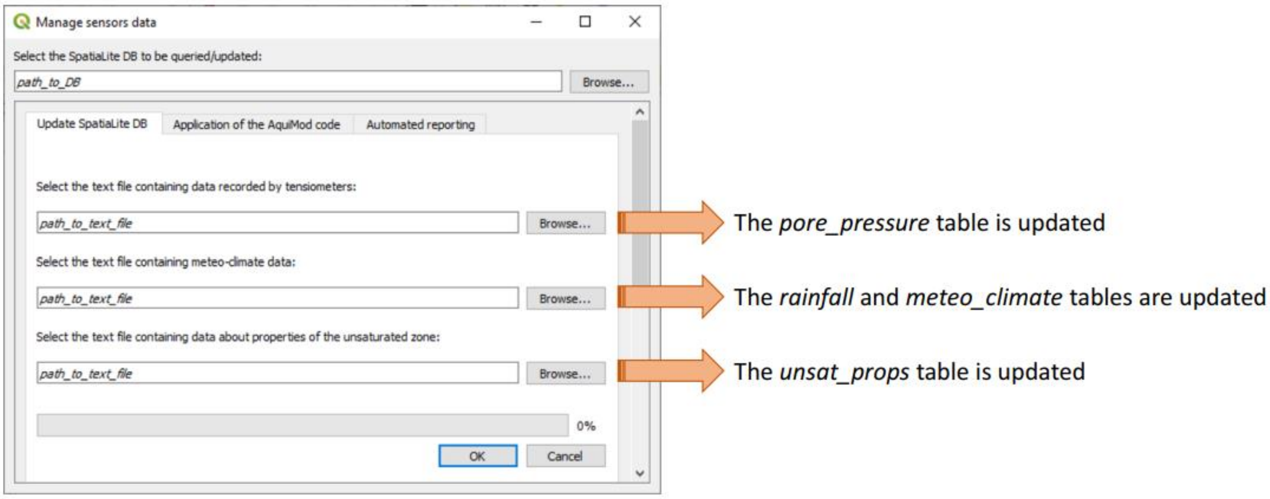


Fig. 3 The structure of the SpatiaLite DB adopted to store the sensors' data


Manage sensors data

Select the SpatiaLite DB to be queried/updated:

path_to_DB Browse...

Select the text file containing data recorded by tensiometers:

path_to_text_file Browse...

Select the text file containing meteo-climate data:

path_to_text_file Browse...

Select the text file containing data about properties of the unsaturated zone:

path_to_text_file Browse...

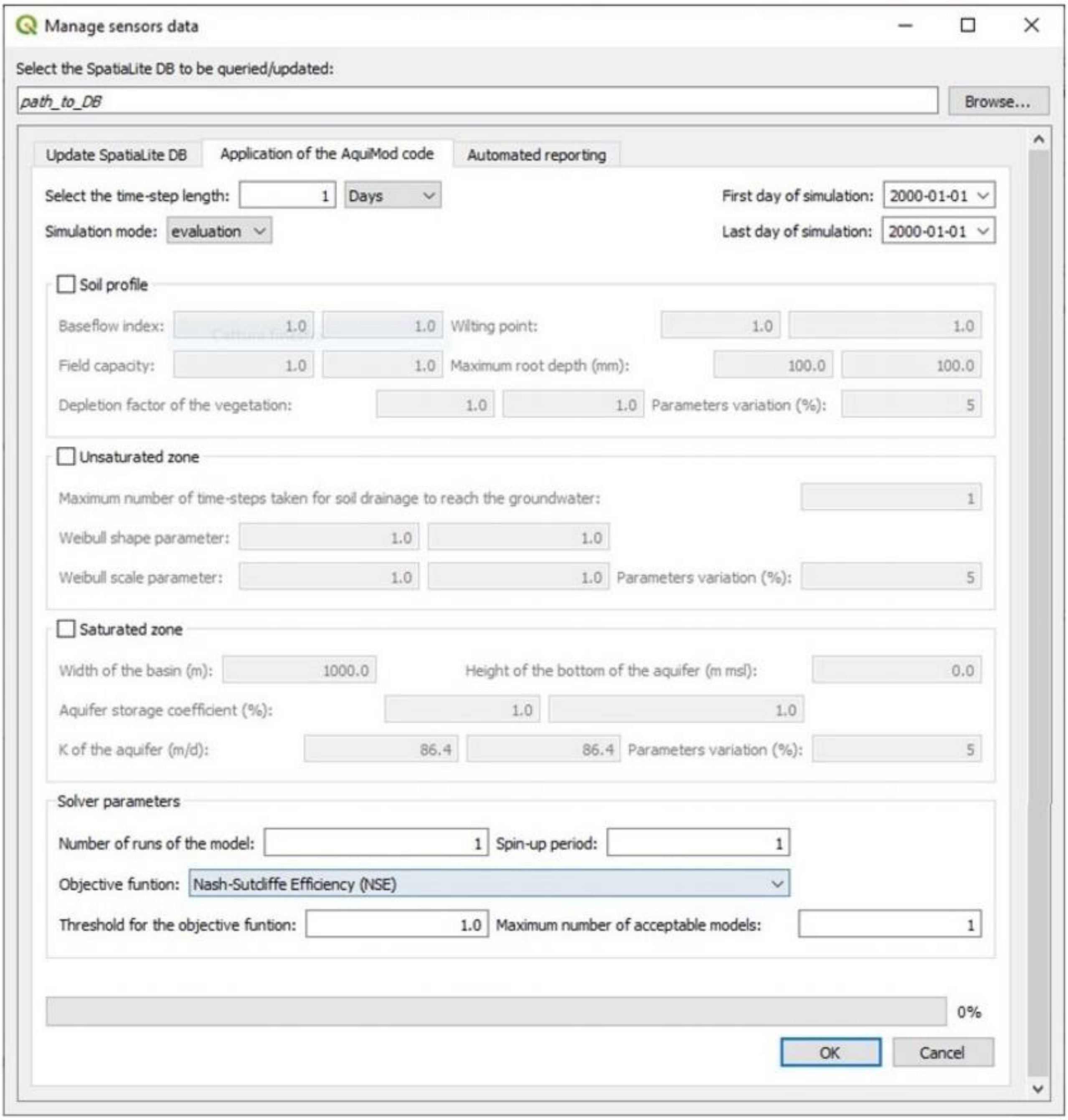
0%

The *pore_pressure* table is updated

The *rainfall* and *meteo_climate* tables are updated

The *unsat_props* table is updated

Fig. 4 The graphical user interface (GUI) used to update the SpatiaLite DB



Manage sensors data

Select the SpatiaLite DB to be queried/updated: path_to_DB [Browse...]

Update SpatiaLite DB | Application of the AquiMod code | Automated reporting

Select the time-step length: 1 Days

First day of simulation: 2000-01-01

Simulation mode: evaluation

Last day of simulation: 2000-01-01

☐ Soil profile

Baseflow index: 1.0 1.0

Wilting point: 1.0 1.0

Field capacity: 1.0 1.0

Maximum root depth (mm): 100.0 100.0

Depletion factor of the vegetation: 1.0 1.0

Parameters variation (%): 5

☐ Unsaturated zone

Maximum number of time-steps taken for soil drainage to reach the groundwater: 1

Weibull shape parameter: 1.0 1.0

Weibull scale parameter: 1.0 1.0

Parameters variation (%): 5

☐ Saturated zone

Width of the basin (m): 1000.0

Height of the bottom of the aquifer (m msl): 0.0

Aquifer storage coefficient (%): 1.0 1.0

K of the aquifer (m/d): 86.4 86.4

Parameters variation (%): 5

Solver parameters

Number of runs of the model: 1

Spin-up period: 1

Objective funtion: Nash-Sutcliffe Efficiency (NSE)

Threshold for the objective funtion: 1.0

Maximum number of acceptable models: 1

0%

[OK] [Cancel]

Fig. 5 The GUI used to generate model files for AquiMod

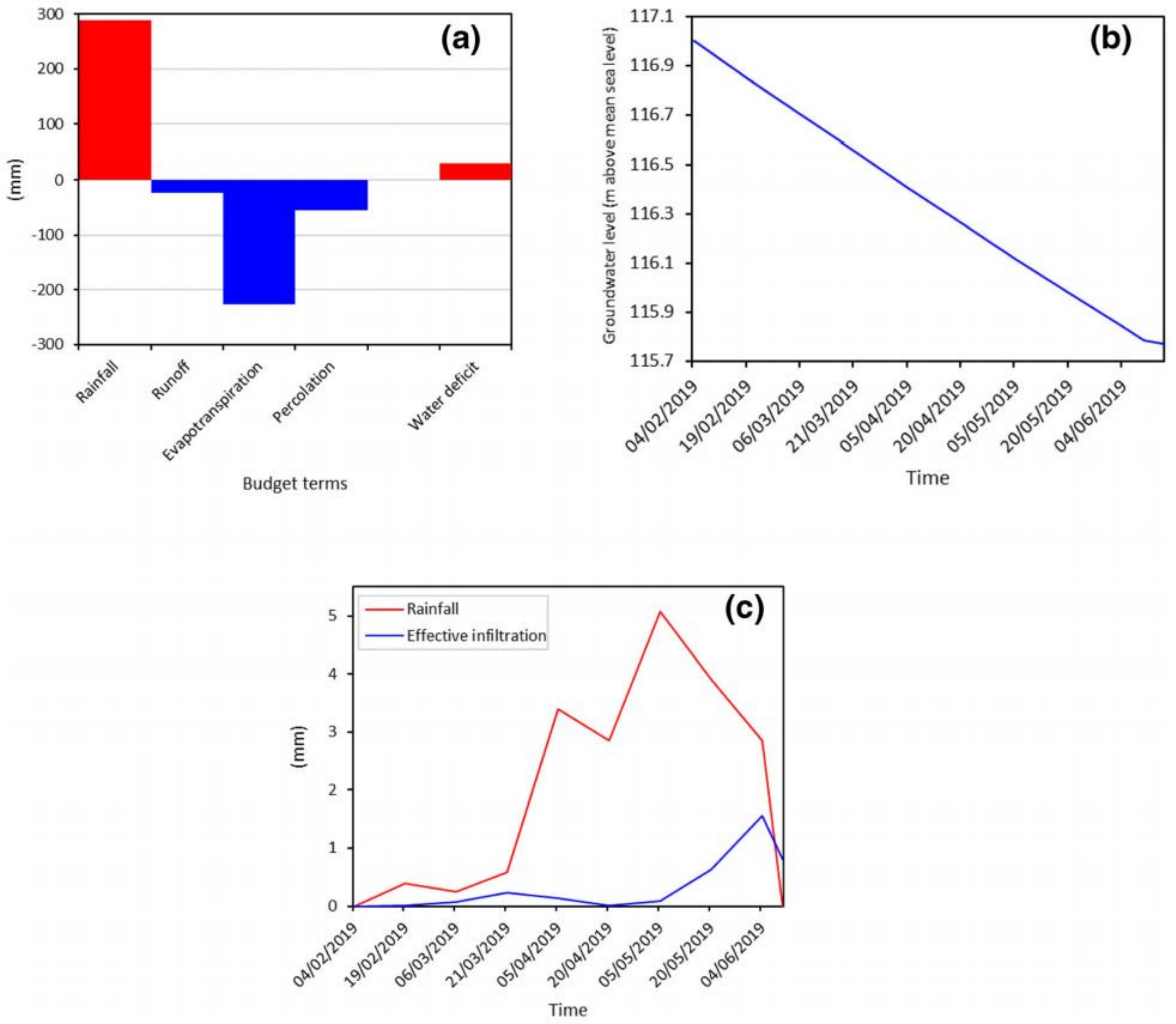
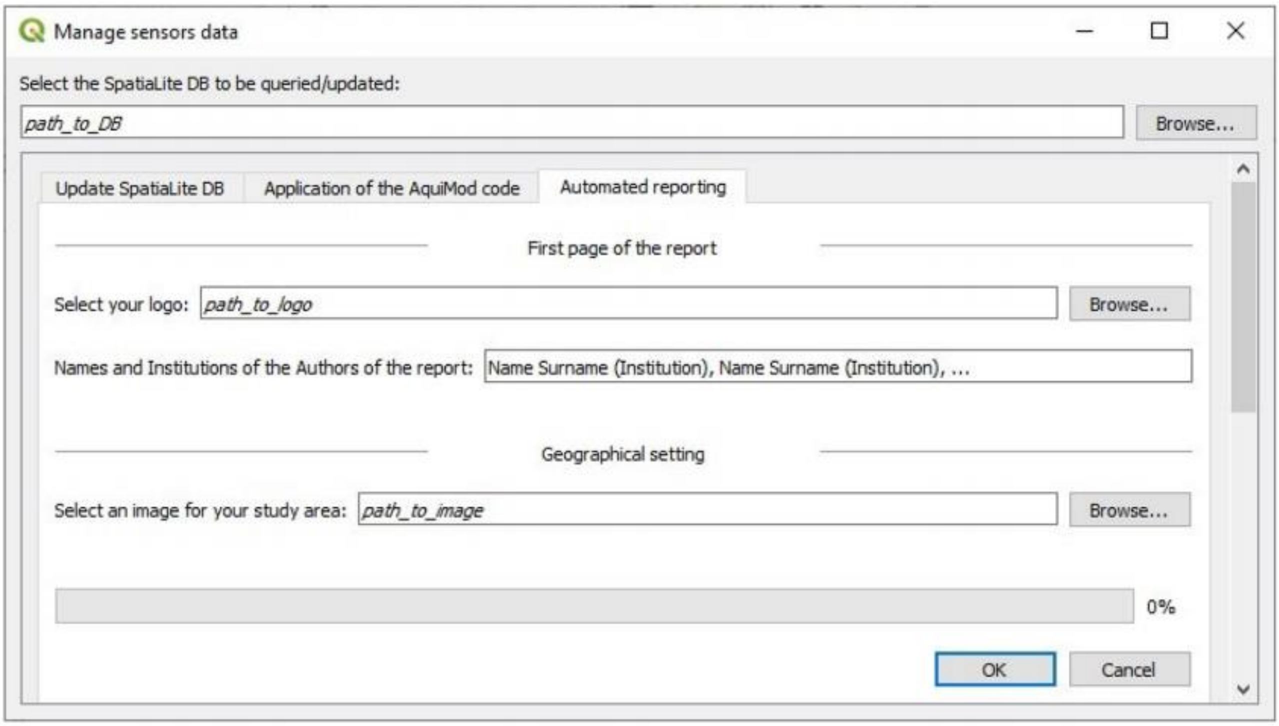


Fig. 6 Results of the AquiMod application. (a) Water budget of the soil profile; (b) time plot of the effective infiltration through the unsaturated zone with respect to rainfall; (c) time plot of the groundwater level simulated



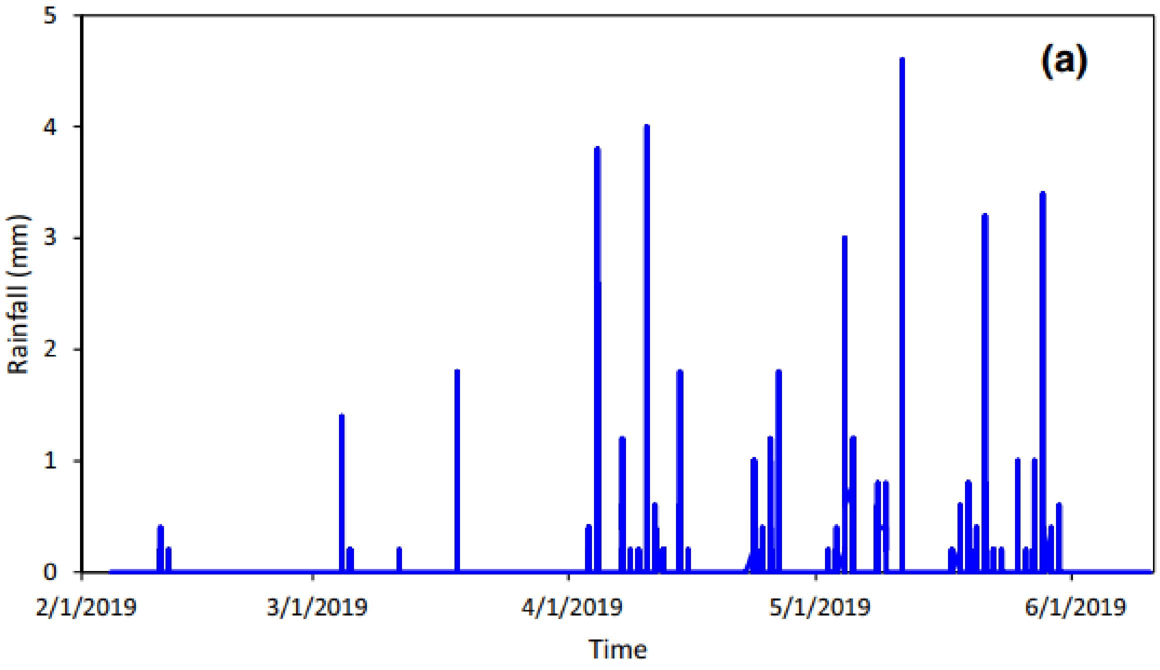
Select the SpatiaLite DB to be queried/updated: *path_to_DB*

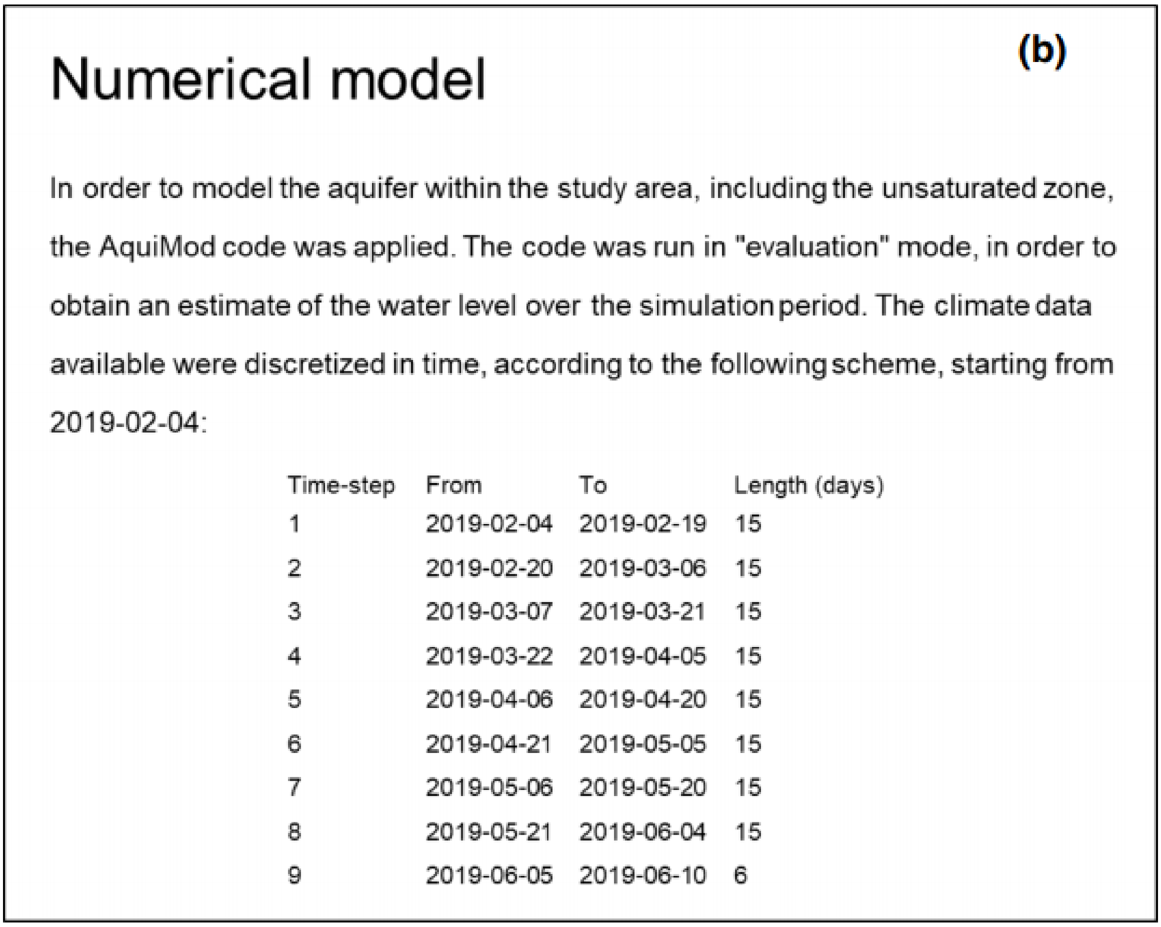
Select your logo: *path_to_logo*

Names and Institutions of the Authors of the report: Name Surname (Institution), Name Surname (Institution), ...

Select an image for your study area: *path_to_image*

Fig. 7 The GUI used to generate an automated pdf report



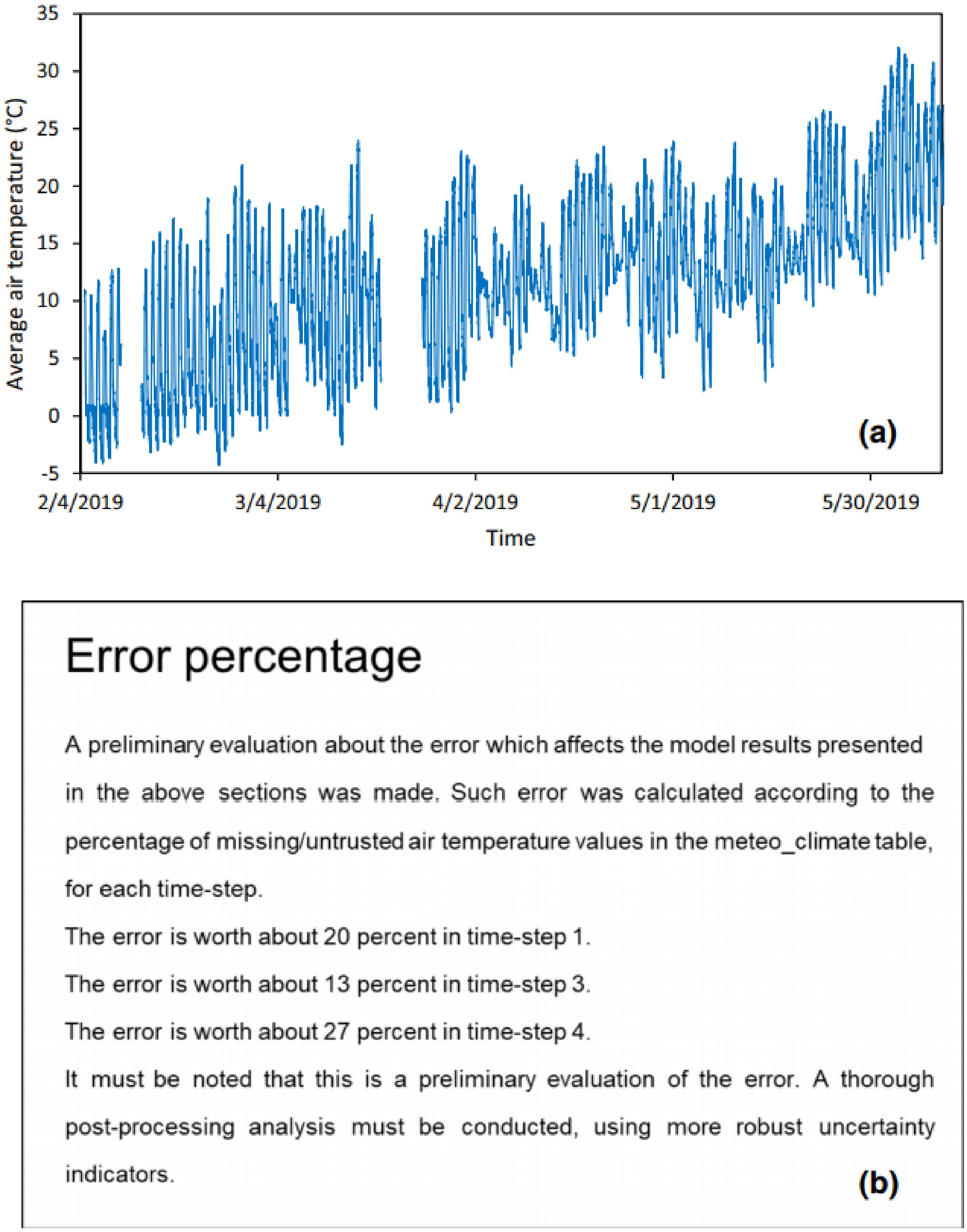


Numerical model

In order to model the aquifer within the study area, including the unsaturated zone, the AquiMod code was applied. The code was run in "evaluation" mode, in order to obtain an estimate of the water level over the simulation period. The climate data available were discretized in time, according to the following scheme, starting from 2019-02-04:

Time-step	From	To	Length (days)
1	2019-02-04	2019-02-19	15
2	2019-02-20	2019-03-06	15
3	2019-03-07	2019-03-21	15
4	2019-03-22	2019-04-05	15
5	2019-04-06	2019-04-20	15
6	2019-04-21	2019-05-05	15
7	2019-05-06	2019-05-20	15
8	2019-05-21	2019-06-04	15
9	2019-06-05	2019-06-10	6

Fig. 8 Two extracts of the pdf report produced. (a) Plot of the rainfall data over the simulation period; (b) extract of the part related to the AquiMod model



Error percentage

A preliminary evaluation about the error which affects the model results presented in the above sections was made. Such error was calculated according to the percentage of missing/untrusted air temperature values in the meteo_climate table, for each time-step.

The error is worth about 20 percent in time-step 1.

The error is worth about 13 percent in time-step 3.

The error is worth about 27 percent in time-step 4.

It must be noted that this is a preliminary evaluation of the error. A thorough post-processing analysis must be conducted, using more robust uncertainty indicators.

Fig. 9 (a) Plot of the air temperature data over the simulation period. The gaps in the time-series are evident; (b) extract of the pdf report related to the estimation of the error that affects the models' results