

1 **Using a Data Grid to Automate Data Preparation Pipelines**
2 **Required for Regional-Scale Hydrologic Modeling**

3 Mirza M. Billah¹, Jonathan L. Goodall^{2,3,*}, Ujjwal Narayan⁴, Bakinam T. Essawy²,
4 Venkat Lakshmi⁵, Arcot Rajasekar⁶, Reagan W. Moore⁶

5

6 ¹ *Department of Biological Systems Engineering, Virginia Tech, Blacksburg, Virginia*

7 ² *Department of Civil and Environmental Engineering, University of Virginia,*
8 *Charlottesville, Virginia*

9 ³ *Department of Civil and Environmental Engineering, University of South Carolina,*
10 *Columbia, SC*

11 ⁴ *CMNS-Earth System Science Interdisciplinary Center, University of Maryland, College*
12 *Park, MD*

13 ⁵ *Department of Earth and Ocean Sciences, University of South Carolina, Columbia, SC*

14 ⁶ *School of Information and Library Science, University of North Carolina, Chapel Hill,*
15 *NC*

16 * *To whom correspondence should be addressed (E-mail: goodall@virginia.edu;*
17 *Address: University of Virginia, Department of Civil and Environmental Engineering,*
18 *PO Box 400742, Charlottesville, Virginia 22904; Tel: (434) 243-5019)*

Abstract

Modeling a regional-scale hydrologic system introduces major data challenges related to the access and transformation of heterogeneous datasets into the information needed to execute a hydrologic model. These data preparation activities are difficult to automate, making the reproducibility and extensibility of model simulations conducted by others difficult or even impossible. This study addresses this challenge by demonstrating how the integrated Rule Oriented Data Management System (iRODS) can be used to support data processing pipelines needed when using data-intensive models to simulate regional-scale hydrologic systems. Focusing on the Variable Infiltration Capacity (VIC) model as a case study, data preparation steps are sequenced using rules within iRODS. VIC and iRODS are applied to study hydrologic conditions in the Carolinas, USA during the period 1998-2007 to better understand impacts of drought within the region. The application demonstrates how iRODS can support hydrologic modelers to create more reproducible and extensible model-based analyses.

Keywords: data management; workflows; hydrologic modeling, iRODS

Software availability: The software is provided open source and freely available on GitHub. Visit https://github.com/uva-hydroinformatics-lab/VIC_Pre-Processing_Rules for additional details.

1 INTRODUCTION

2 Motivation

3 Application of regional-scale hydrologic models presents a number of challenges
4 associated with handling and processing large datasets. These models are data intensive
5 and require a significant amount of time, effort, and resources in order to transform
6 available datasets into the form required by the model (Leonard and Duffy, 2013). The
7 information required by models is contained within multiple datasets maintained by
8 various data providers with each dataset having unique data access protocols, file
9 formats, and semantics (Horsburgh et al., 2014). The result is that the input datasets for a
10 hydrologic model require specific transformations before they can be used to setup,
11 calibrate, and validate the model (Leonard and Duffy, 2014).

12 Due to the level of heterogeneity across data sources and the inconsistent input
13 file formats required by various models, these data transformation steps are difficult to
14 automate. With the exception of a few models with robust data preparation tools, data
15 transformation steps associated with hydrologic model data preparation often require
16 significant manual intervention. Even for models with data preparation tools available,
17 these tools are often tightly coupled to pre-specified data sources that may not be the best
18 available information for a specific region or modeling objective. The end result is that
19 modelers (i) consume significant time on tasks that could be automated, (ii) lack the
20 ability to easily reproduce and extend past work completed by others, and (iii) are unable
21 to take full advantage of the most recently available information when creating models.

22 Within the information and computer science communities, there has been work
23 to create advanced data management and scientific workflow software (Foster, 2011; Gil

1 et al., 2007; Ludäscher et al., 2006; Moore and Rajasekar, 2014; Oinn et al., 2006). These
2 tools have been applied within many scientific communities including hydrology (Fitch
3 et al., 2011; Guru et al., 2009; Perraud et al., 2010; Piasecki and Lu, 2010). Scientific
4 workflow environments offer many benefits for implementing data preparation tasks. For
5 example, workflow environments coordinate and automate processing steps, as well as
6 track the provenance of the datasets generated through the processing steps. This is
7 important especially for reproducibility, transparency, and reuse of computational
8 analyses.

9 There remain challenges, however, in applying workflow tools for hydrologic
10 modeling support. One challenge is the lack of a common data model within hydrology
11 (Perraud et al., 2010). It is typical for hydrologic models to require gridded data in
12 NetCDF or GRIB formats, times series in CSV or WaterML format, and GIS layers in
13 shapefile, geodatabase, or raster formats. These data come from NASA, NOAA, USGS,
14 and others, with each agency adopting its own semantics and structure. Another challenge
15 is the large number of models used in the hydrology community, each with its own data
16 formats, structures, and semantics. There is significant heterogeneity across these
17 hydrologic models with some models adopting a gridded data structure and others
18 adopting a feature-oriented (e.g., watersheds and stream networks) data structure. In
19 short, there is very little commonality between input file structures and semantics across
20 hydrologic data and models.

21 For these reasons, we argue that data preparation for regional-scale hydrologic
22 modeling is a data-centric rather than process-centric task. That is to say, due to the lack
23 of a common data model in hydrology, the primary goal is to create data processing

1 pipelines that define the logic for integrating and transforming the information contained
2 within multiple datasets into a new product required by the hydrologic model as input.
3 While others have argued for a centralized and standardized data model for hydrology
4 (Goodall and Maidment, 2009; Leonard and Duffy, 2014; Maidment, 2002), we argue
5 here that the current level of heterogeneity in source data and model input requirements
6 makes this level of standardization, while an important long-term goal, difficult to obtain
7 in the short-term. Rather, data can remain decentralized and under the control of data
8 providers, but there should be a means for applying server-side data transformation
9 pipelines to reference datasets. These pipelines, which could be written and maintained
10 by model developers and users, will allow for on-demand access to derived data products
11 required by specific hydrologic models. In the spirit of Relational Database Management
12 Systems (RDMS), these pipelines can be thought of as views of the underlying data,
13 tailored for a specific model.

14 In this paper, we present a method for creating server-side data pre-processing
15 pipelines using the DataNet Federation Consortium (DFC) data grid, which is powered
16 by the Integrated Rule-Oriented Data System (iRODS). We illustrate the methodology
17 using the Variable Infiltration Capacity (VIC) regional-scale hydrologic model. These
18 systems are briefly introduced in the following background section. Then, in the design
19 and implementation section, the approach for automating VIC data preparation pipelines
20 using iRODS is presented. Finally, the pipelines are demonstrated for an example of
21 modeling drought in the Carolinas region of the United States.

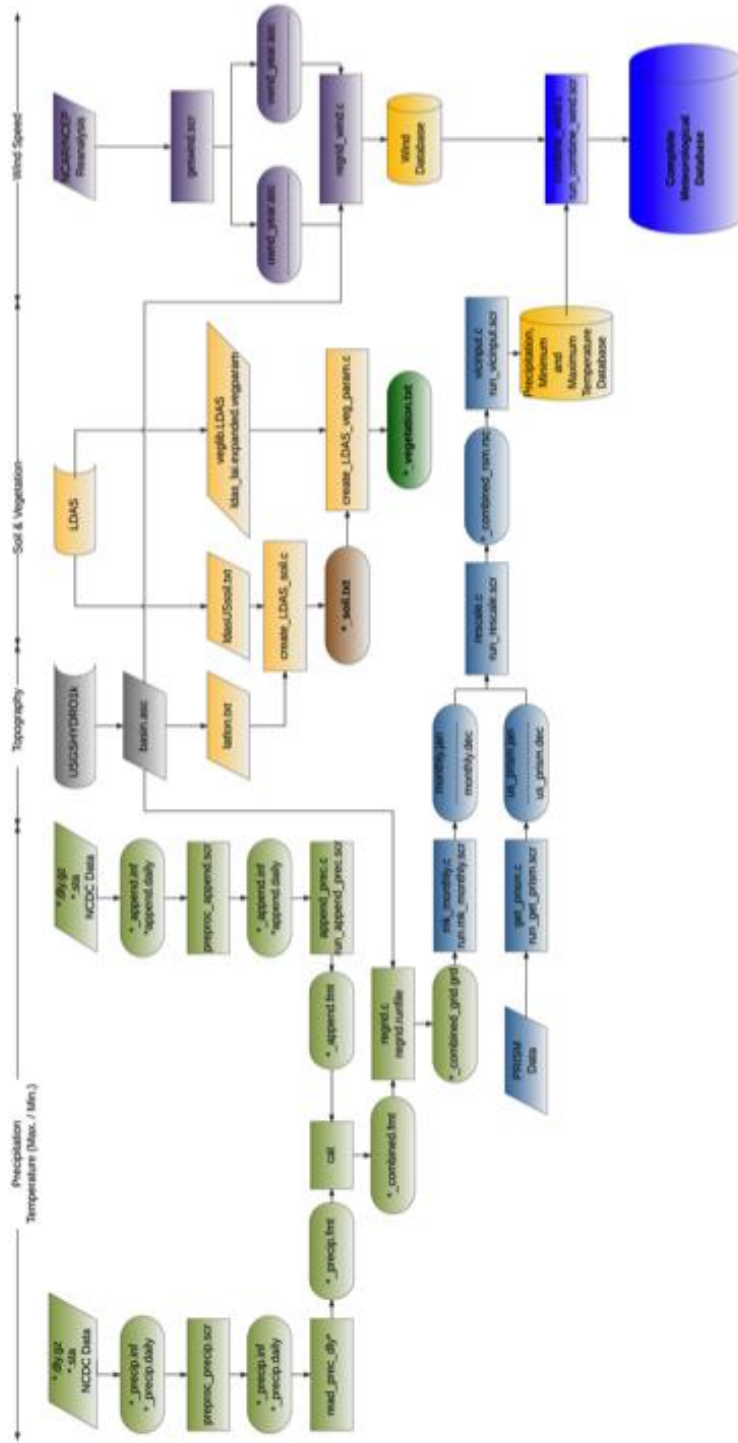
1 **Background**

2 The DFC grid (<http://www.datafed.org>) was built as part of an NSF-funded
3 research project to provide storage and compute resources that allow for long-term access
4 to the stored datasets. DFC is enabled by iRODS, an open source, policy-based
5 cyberinfrastructure developed by the Data Intensive Cyber Environments (DICE) group
6 for distributed data management (Rajasekar et al., 2010b) (<http://www.irods.org>). It is
7 used by a wide variety of end users including various scientific communities (Chiang et
8 al., 2011; Goff et al., 2011; NASA, 2015). Data management tasks and policies are
9 implemented within iRODS as rules. Rules specify a sequence of lower-level micro-
10 services that operate on datasets within the data grid. Users can specify sequences of data
11 collection, transformation, curation, preservation, and processing steps within one or
12 more rules. iRODS includes a Rule Engine (RE) that allows for remote execution of rules
13 on iRODS resource servers, which is especially beneficial for large-datasets. iRODS,
14 therefore, provides a means for uploading processing routines to data resource servers,
15 whereas the typical approach used now in hydrology is to download data for local
16 processing.

17 The Variable Infiltration Capacity (VIC) model is a large-scale hydrologic model
18 that applies water and energy balances to simulate terrestrial hydrology at a regional
19 spatial scale (Liang et al., 1996a). The scientific background of the model is summarized
20 in the case study section, while here the model is presented from a data management
21 perspective. The model requires several input datasets, and these datasets must be
22 generated through a sequence of data processing steps. Figure 1 shows the data
23 processing pipeline that is typically done manually to create the meteorological and land

1 surface variable inputs for a VIC model simulation. The figure illustrates the difficulties
2 in performing this workflow, which include the fact that different datasets are required,
3 each with a different data model, and processing steps are written in different languages
4 and sometimes require legacy compilers due to the time that they were developed.
5 Executing these data processing scripts currently requires significant effort, especially for
6 new users that must learn the steps and configure often complicated legacy software to
7 correctly execute the sequence of steps. While Figure 1 depicts the process as an
8 integrated workflow, in practice it is instead a sequence of steps often performed
9 independently with software developed at different times by different scientists. While
10 this description is for VIC, we believe it describes a general problem faced by other data
11 intensive models.

12



1
 2 Figure 1: Data pre-processing steps for VIC model depicted as a flow chart of legacy
 3 processing scripts written in different languages, operating on different input datasets,
 4 and creating different output datasets.

1 **Study Objective and Scope**

2 Given these data challenges in running data-intensive hydrologic models, and
3 given the potential advantages of advanced data management systems such as iRODS,
4 the objective of this study is to apply iRODS to support hydrologic modeling. More
5 specifically, the focus of this research is on the data preparation pipeline used to create
6 the input files required for running VIC. The goal is to automate these steps as iRODS
7 rules. The rules provide a means for server-side execution of data processing pipelines,
8 moving closer to the long-term goal of reproducible end-to-end model simulations. The
9 study is built on prior work where the VIC model was used for simulating a period of
10 drought in South Carolina, USA (Billah and Goodall, 2011; Billah et al., 2015). VIC is a
11 widely used model for investigating regional-scale hydrologic systems (Abdulla et al.,
12 1996; Lakshmi et al., 2004; Lohmann et al., 1998; Sheffield and Wood, 2007; Sheffield
13 et al., 2004), and the software resulting from the prototyping work can be used by this
14 community. More generally, the outcome of this work is a methodology for creating
15 server-side data processing pipelines that could be applied for other data-intensive
16 hydrologic, environmental, and Earth system models.

17 **SYSTEM DESIGN AND IMPLEMENTATION**

18 **The DFC-Hydrology Grid**

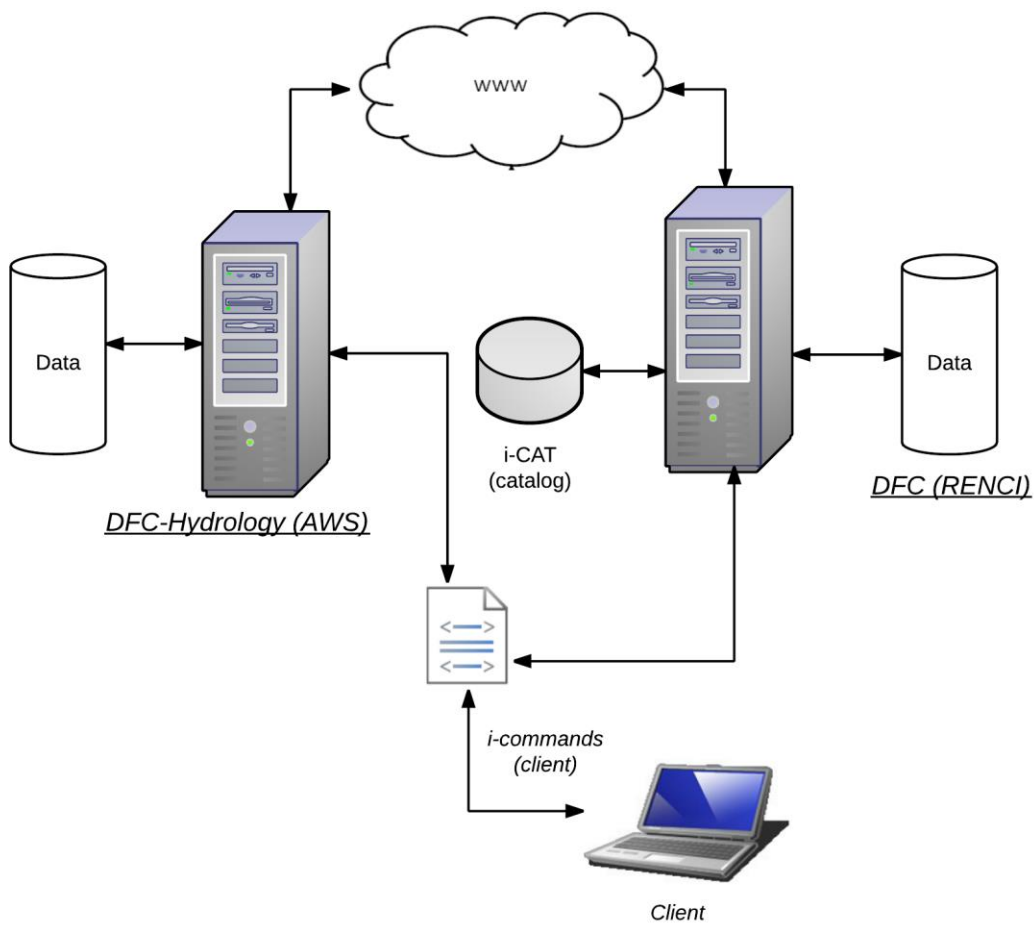
19 The server-side data processing software shown in Figure 1 is deployed on an
20 iRODS resource server that is part of the DFC-hydrology grid. The DFC-hydrology grid
21 is part of the larger DFC grid (Figure 2), which is funded by the National Science
22 Foundation (NSF) to support data collection, analysis, preservation, sharing, and

1 publication of scientific data and models (<http://www.datafed.org>). The DFC-hydrology
2 grid consists of an iRODS resource server that communicates with an iRODS metadata
3 catalog (iCAT) server located in the DFC grid. The iCAT server is administered by
4 RENCI (Renaissance Computing Institute) in Chapel Hill, North Carolina. iRODS also
5 includes a Rule Engine (RE) that interprets rules executed on the server, but initiated
6 from a client program. The RE connects to the catalog server in the DFC grid and updates
7 the iCAT database if new files are generated through the execution of a rule.

8 An iRODS client application is used to initiate data management tasks including
9 rules within the federated grid. A commonly used iRODS client application is the i-
10 commands utility that consists of several command-line utilities for managing datasets
11 and executing data processing commands (Rajasekar et al., 2010a). These i-commands
12 can be used to put and get data into/from the DFC grids, as well as execute a set of other
13 commands for data management, analysis, and sharing. A user has the opportunity to
14 create and execute rules to chain commands for tasks such as gaining access to
15 heterogeneous data sources and processing datasets into the formats required by models
16 or scientific communities.

17 The hydrology related data collected from various external sources are stored and
18 preserved in the DFC-hydrology grid in their native formats. This data gathering process
19 can be automated using iRODS functionality and different servers within the grid. For
20 instance, the precipitation data collection can be obtained from remote servers by one
21 iRODS resource server and data transformations could be performed by a second iRODS
22 resource server in the DFC-hydrology grid. Because these servers are part of the same
23 data grid, the distributed data stored across the servers appears to client applications as

1 part of the same logical file directory. iRODS includes services for accessing remote
2 resources using different access protocols (e.g., HTTP or FTP). Rules and policies can be
3 established for the grid to provide continuous data synchronization between data in the
4 grid and data stored on remote servers. Many of these more advanced features were not
5 included in the prototyping research described in this paper, but could be added through
6 future work and refinement of the system.



7
8 Figure 2: Schematic diagram of the NSF supported Data Federation Consortium (DFC)
9 data management system showing the connections between the DFC-hydrology grid with
10 the DFC grid.

1 **Micro-Services**

2 Micro-services are the building blocks for implementing policy-based data
3 management within the DFC grid (Rajasekar et al., 2010a). A micro-service is a well-
4 defined function that performs a specific task as part of a distributed workflow system. A
5 number of micro-services are available to automate data collection, processing, and
6 storage in the DFC federated resource servers. These micro-services are primarily
7 developed by system or application programmers, but could also be written by scientists.
8 The micro-services that are applied for this study are listed in Table 1. These micro-
9 services are included in the latest iRODS release and can be chained together using rules.
10 Although the flexibility to chain a number of micro-services provides multiple ways to
11 complete a series of tasks, iRODS applies priorities and validation conditions to select the
12 best micro-service to complete a given task.

13 Table 1: Micro-services applied for VIC model application using iRODS.

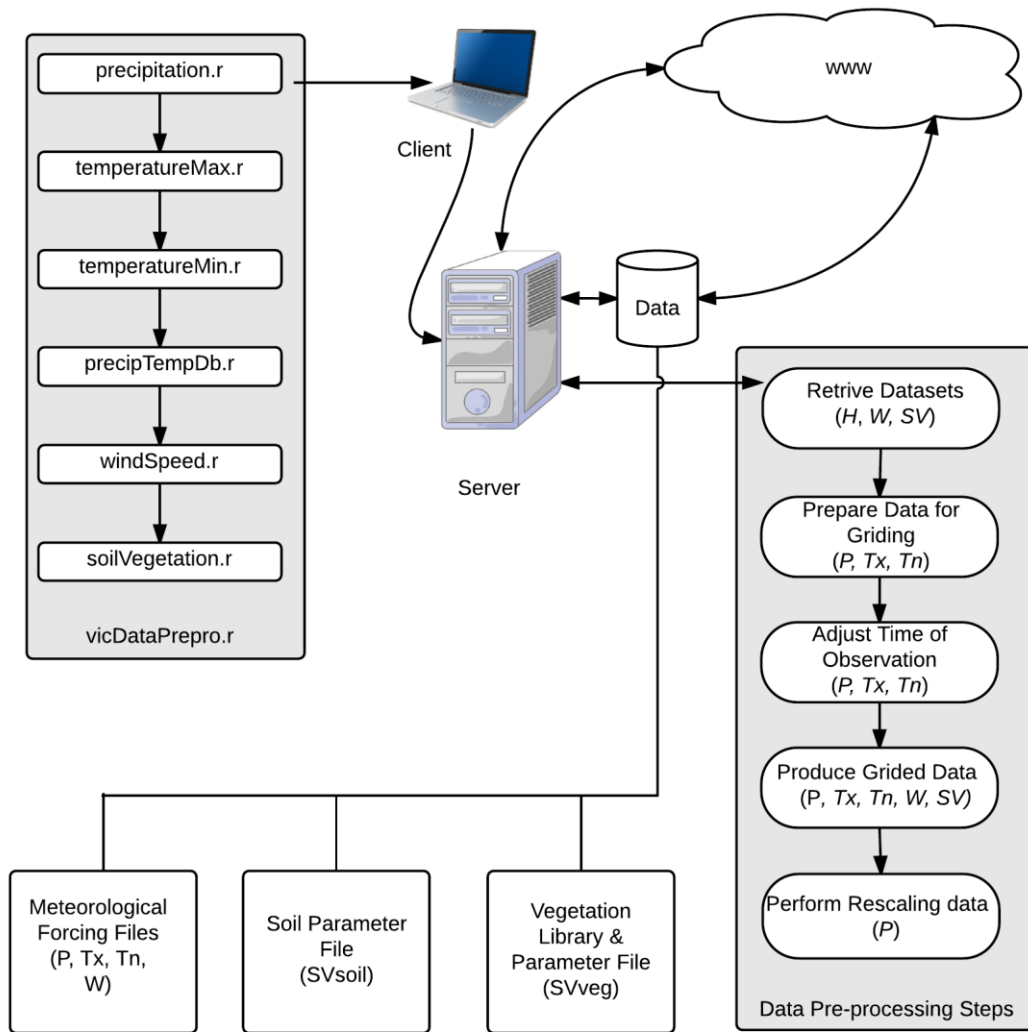
No.	Micro-service	Purpose
1	msiExecCmd	Execute commands
2	msiCollCreate	Make data collection
3	msiDataObjCreate	Create data object
4	msiDataObjWrite	Write data object
5	msiDataObjClose	Close data object
6	msiAddSelectFieldToGenQuery	Make query for data using field
7	msiAddConditionToGenQuery	Make query for data using condition
8	msiExecGenQuery	Execute Query
9	msiGetValByKey	Extract value from query result
10	msiSplitPath	Get directory path
11	msiDataObjUnlink	Delete temporary file
12	msiRmColl	Remove data collection
13	msiGetSystemTime	Get time stamp

14

1 **Rules**

2 The rule is a critical and fundamental component for iRODS. It provides a
3 flexible mechanism to integrate external systems for specialized processing and metadata
4 management (Hedges et al., 2009, 2007). In this system, we use rules to implement data
5 processing pipelines on the DFC grid. Data pre-processing rules involve collecting and
6 transforming datasets from heterogeneous sources into the inputs required by VIC. For
7 our purposes, data were collected from the United States Geological Survey (USGS),
8 National Climatic Data Center (NCDC), National Center For Atmospheric Research
9 (NCAR), National Centers for Environmental Prediction (NCEP), and Land Data
10 Assimilation System (LDAS). The DFC catalog is updated automatically when data are
11 put into the grid. This catalog functions as an information center and enables the
12 discovery of distributed data stored within the grid.

13 Data processing workflows can be implemented as rules that transform the
14 collected datasets from the external sources into model readable inputs (Figure 3). These
15 rules are a combination of multiple step-based routines, each of which performs a
16 particular task. The steps are shown in Figure 3 and include tasks such as retrieving data,
17 preparing data for gridding, adjusting observation times, and transforming gridded and
18 rescaled datasets. For our study, the routines for executing these steps are installed on a
19 resource server that is part of the DFC-hydrology grid and integrated into data-specific
20 rules used to complete a series of tasks from collection and transformation of the datasets
21 into model inputs. Separate rules were created to perform the data transformation tasks
22 grouped into logical divisions as shown in Table 2.



1
 2 Figure 3: Model pre-processing workflows showing the major steps for transforming
 3 datasets to set up the VIC model for a specific study area. Rules are initiated from a client
 4 but executed on a server using micro-services.

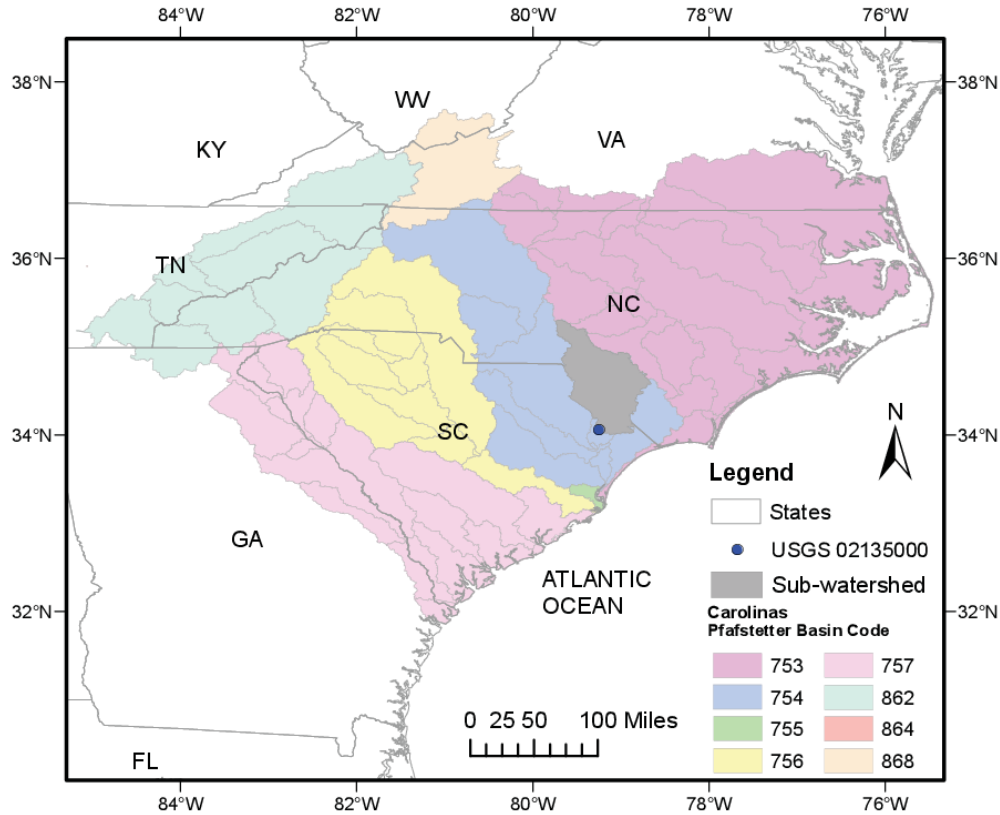
1 Table 2: Rules created within iRODS for VIC data pre-processing.

No.	Micro-service	Purpose
1	vicDataPrepro.r	Rules all subrules for VIC data pre-processing
2	precipitation.r	Collect and process daily precipitation data
3	temeperatureMax.r	Collect and process daily maximum temperature data
4	temeperatureMin.r	Collect and process daily minimum temperature data
5	precipTempDb.r	Combines precipitation and temperature data
6	windSpeed.r	Collect and process annual wind speed data
7	soilVegetation.r	Collect and process soil and vegetation data

2 EXAMPLE APPLICATION

3 Study Area

4 The region used for the case study application covers all the major river basins in
5 both North and South Carolina with a total area of 280,736 km² (108,393 mi²) (Figure 5).
6 The Pfafstetter Basin Code system was used to define this “Carolinas” study region from
7 the HYDRO1k basin dataset. The codes were incorporated in the hydro1kCarolina.r rule
8 to extract the area during execution of the workflow.



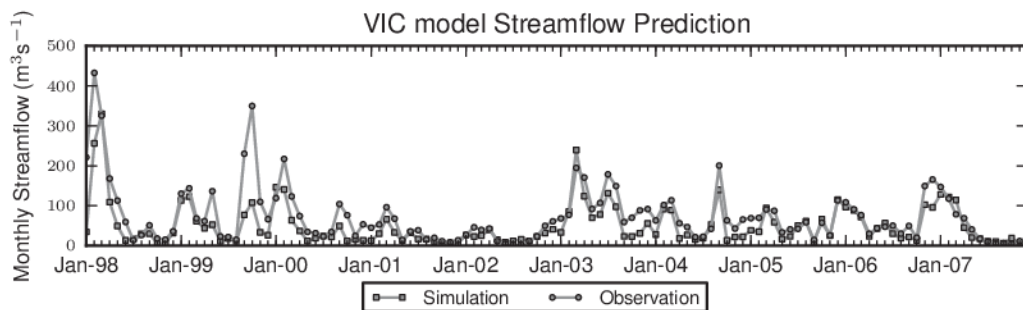
1
 2 Figure 5: Study area with major river basins in North and South Carolina, USA. These
 3 subwatersheds were extracted using selected Pfafstetter Basin Code from the Hydro1K
 4 dataset. The watershed and stream gage station used in the VIC model calibration are also
 5 shown.

6 Model Application

7 VIC is able to simulate the land surface portion of the hydrologic cycle by solving
 8 the full water and surface energy balance equations (Liang and Lettenmaier, 1994; Liang
 9 et al., 1996b). VIC was calibrated for the study region and period using the following
 10 seven parameters: variable infiltration curve (b), maximum base flow (D_{smax}), fraction
 11 of base flow where base flow occurs (D_s), fraction of maximum soil moisture content
 12 above which nonlinear base flow occurs (W_s), mid (d_2) and deep (d_3) soil layer depth,

1 and minimum stomatal resistance (r_0) (Abdulla and Lettenmaier, 1997a, 1997b; Crow,
2 2003; Troy et al., 2008). The range investigated and the final values of the parameters
3 applied in this study are described in more detail in Billah e al. (2015).

4 A comparison of the monthly average streamflow predicted by the calibrated VIC
5 model and streamflow observations are provided in Figure 6. The streamflow
6 observations are for the Little Pee Dee River at Galivants Ferry station that is part of the
7 USGS National Water Information System (NWIS) network (USGS 02135000; Figure 5)
8 for the period of 1998 to 2007. This streamflow station has a drainage area of 7,257 km²
9 and includes portions of both North and South Carolina. This station was selected based
10 on its available time series record and because it is on an unmanaged portion of a river
11 network. The Nash-Sutcliffe Efficiency (NSE) index of the final calibration is 0.6 at this
12 and other stations used for calibration and validation of the model but not shown here.
13 This NSE value is considered to be a satisfactory calibration by watershed-scale
14 hydrologic modelers (Moriiasi et al., 2007). Further details on the model calibration and
15 validation are provided in Billah et al. (2015).



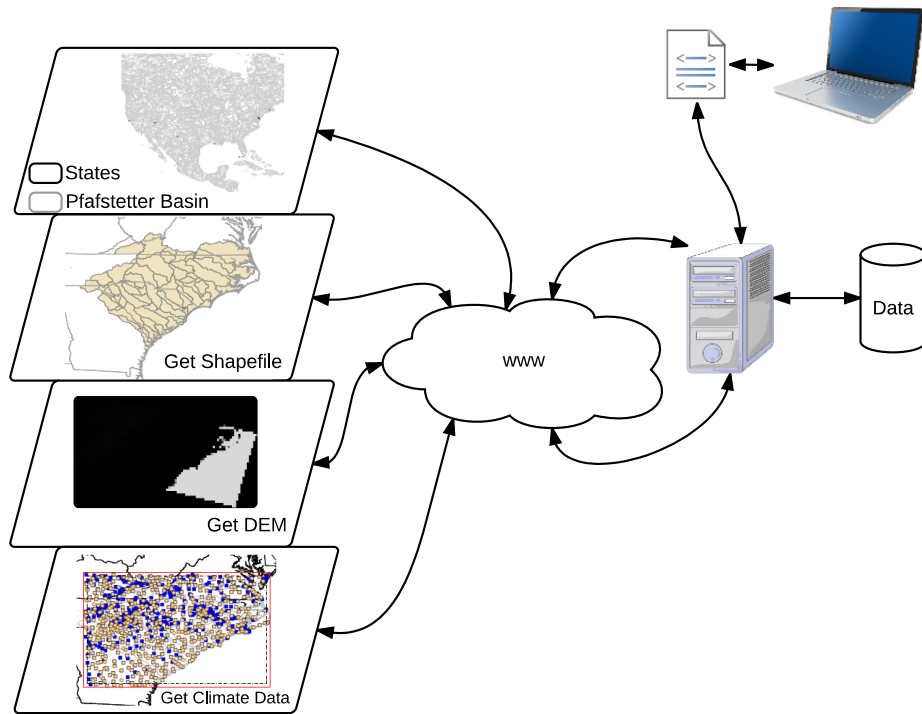
16
17 Figure 6: Streamflow comparison between the VIC model predictions and USGS
18 observations. The comparison is performed at the USGS station Little Pee Dee River at
19 Galivants Ferry, SC (Station Number 02135000).

1 **Data Pre-Processing Pipeline**

2 When executing the data pre-processing rules for a specific area of
3 interest, the first step is to retrieve climate data (precipitation, maximum and minimum
4 temperature data) from the DFC federated grid by initiating the hydro1kCarolinas.r rule.
5 This rule uses a series of micro-service calls initiated on the DFC federated grid to extract
6 and register data for the area of interest from national-scale datasets (Figure 4). We use the
7 Pfafstetter basin numbering system as described in Furans and Olivera (2001) for
8 defining study area from HYDRO1k basin and DEM datasets. The climate datasets for
9 the defined study area were downloaded via FTP from the NCDC Global Historical
10 Climatology Network (GHCND) database. While downloading the climate data, a buffer
11 of 0.25° was considered around the defined study area to collect sufficient climate data.
12 The climate data contained precipitation, maximum and minimum temperature, and wind
13 speed datasets.

14 The precipitation data is then processed using the precipitation.r rule, which uses
15 the GHCND data and converts the station specific datasets into gridded datasets with a
16 spatial resolution of 1/8°. Similarly, temperatureMax.r and temperatureMin.r rules
17 convert station specific temperature values into gridded datasets with 1/8° spatial
18 resolution. Rules are also used to process wind speed, soil, and vegetation data from their
19 respective sources. The annual wind data were collected from NCAR/NCEP and
20 processed to generate gridded datasets of 1/8° resolution for the study area using
21 windSpeed.r. The soilVegetation.r rule was applied to transform the LDAS soil and
22 vegetation information into information required for the model. The result is a collection

1 of VIC input files derived from national-scale reference datasets using iRODS rules that
2 largely wrap legacy data processing routines.

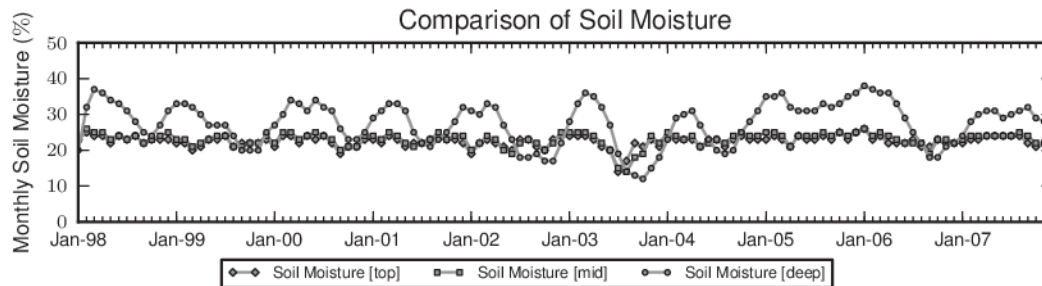


3
4 Figure 4: Data flow in the hydro1kCarolinas.r rule that extracts climate data from NCDC
5 GHCND using HYDRO1K basin/DEM datasets to define a study region.

6 **Model Results**

7 VIC model runs generate a number of grid-based hydrologic flux and state
8 variable outputs. One of these outputs is soil moisture estimated for each grid cell in the
9 simulation domain and for each soil layer within the model. Soil moisture is an important
10 indicator of drought, so it is used as an example model output. It is possible to create data
11 post-processing rules similar to what was done for data pre-processing to extract model
12 results from the various output files generated by the model. For example, a rule could be
13 created to summarize the soil moisture over the study region for each soil layer and to

1 display this information as a time-series plot of monthly soil moisture within the three
2 VIC soil layers (Figure 7). The plot of soil moisture provides a way of depicting the
3 impact of the drought on soil moisture in 2003 and its recovery in 2005, particularly the
4 deep soil layer which is more sensitive to long term trends in water availability compared
5 to the middle and upper soil layers.



6
7 Figure 7: Comparison of monthly averaged soil moisture in the three soil layers predicted
8 by the VIC model in the Carolinas for the periods of 1998 to 2007.

9 **Benchmarking**

10 We performed a benchmarking experiment to quantify the time required to
11 execute the data pre-processing pipeline and the overhead introduced through remote
12 execution using iRODS. The experiment was conducted using virtual machines (VMs) in
13 the Amazon Web Service (AWS) cloud. VMs were created using the AWS Elastic
14 Compute Cloud (EC-2) service. We used a m3.medium instance type, which at the time
15 of this research had a high frequency Intel Xeon E5-2670 v2 (Ivy Bridge) 64-bit
16 processor, a Solid State Drive (SDD), and 3.75 GB of RAM for the iRODS resource
17 server. For the iRODS client machine we used a t2.micro instance type, which at the time
18 of this research had a high frequency Intel Xeon processor with turbo and up to 3.3 GHz
19 and 1 GB of RAM.

1 We ran each rule on the resource server and initiated execution of the rule from
2 the client machine. We also ran the shell scripts that the rules wrap directly on the
3 m3.medium instance in order to measure the time lag introduced by iRODS and the Rule
4 Engine. We ran the processing script for the Carolinas study region for the period 1997-
5 2008 at a 1/8-degree resolution. We tracked the wall time required to complete each of
6 the processing steps and executed each process three times to measure variability in the
7 execution time. The data processing pipeline makes use of data from five different federal
8 data providers, it touches 35 different files and file directories, and it generates a 1.6GB
9 file directory that represents the input files for VIC for the study region and time period
10 of interest.

11 Results of this benchmarking experiment show that the overall rule takes
12 approximately 10 minutes to run for the area and period of interest (Table 3). This
13 assumes reference data have already been gathered and loaded into the data grid. The vast
14 majority of the processing time is spent on meteorological data transformations compared
15 to the soil and vegetation data transformations. This is because the rule processes
16 precipitation, minimum and maximum air temperature, and wind speed directly from data
17 providers (NCDC and NCAR). For soil and vegetation data, the rule makes use of pre-
18 processed data created specifically for VIC models provided through the North American
19 Land Data Assimilation System (NLDAS).

20 The iRODS Rule Engine (RE) does appear to introduce some overhead compared
21 to directly calling the process using shell scripts on the server. Considering the overall
22 rule processing times and averaging across the three runs for each case, the RE case was
23 11 seconds (2%) slower. Given that the RE allows for remote execution of the data

1 processing pipeline, we feel this is a fairly minimal cost to pay for the added benefit.
 2 Because the focus of this research was on creating the data processing pipeline rather
 3 than optimizing its execution time, there are likely opportunities to reduce these
 4 execution times through modifications to the rule structure and underlying source code.

5 Table 3: Wall time to complete for each data processing step when the logic internal to
 6 the rule is initiated directly on the server using a shell script and when the rule is initiated
 7 from a client machine and executed on the server using the iRODS Rule Engine (RE).
 8 For each case, the rule was executed three runs to capture the execution time variability.

Execution Time (sec)

Rule	On Server/ Shell Script			From Client / Rule Engine		
	Run 1	Run 2	Run 3	Run 1	Run 2	Run 3
precipitation.r	82	75	76	71	71	69
temeperatureMax.r	162	162	162	161	162	162
temeperatureMin.r	161	161	156	161	160	159
precipTempDb.r	39	38	38	38	39	38
windSpeed.r	115	122	116	144	132	125
soilVegetation.r	27	20	27	27	27	27
vicDataPrepro.r (parent rule)	586	578	575	602	591	580

9

10 SUMMARY, DISCUSSION, AND CONCLUSIONS

11 The hydrologic modeling process involves many steps from data access and
 12 transformation, to model setup, calibration, and validation, to analysis and visualization
 13 of model outputs. This entire “end-to-end” process involves some steps that are easily
 14 automated and others that require intervention by expert modelers. The goal in this and
 15 related work is to automate those steps that are straightforward but tedious, while still
 16 allowing experts to guide the process and intervene when needed. Currently too many
 17 steps that can be automated are not. As a result, modelers are unable to focus on the
 18 important tasks that require their expertise and insights because time must be spent on

1 more basic data gathering and transformation steps. Furthermore, the steps that could be
2 automated are typically not thoroughly documented and, even if they are thoroughly
3 documented, are time consuming to repeat. This makes independent reproducibility of
4 model results, a requirement for scientific progress and water resource management
5 objectives, difficult or even impossible.

6 It is important to note the specific data challenges required for a hydrologic model
7 application. First, VIC requires a large amount of data when applied to a region the size
8 of the Carolinas, and of course even more data would be required for Continental scale
9 model executions, which are not uncommon when applying VIC. The data used in the
10 VIC model pre-processing steps included meteorological datasets at point stations and on
11 grids, topography datasets available as grids, and soil and vegetation datasets also
12 available as grids. Transformation of these raw input data resulted in intermediate
13 datasets with different spatial projections, filled gaps, and other modifications required
14 before initiating the model simulation. Over the years, researchers have created scripts for
15 completing many of these data pre-processing steps, but less effort has been devoted to
16 schemes for integrating these scripts into reproducible workflows. As data volumes
17 continue to increase, more sophisticated ways of handling these data are needed.

18 This work addresses these challenges by leveraging the iRODS technology and
19 the DataNet Federation Consortium (DFC) cyberinfrastructure to create data processing
20 pipelines that automate pre-processing steps for VIC. The workflows developed for VIC
21 include sufficient information to allow others to independently reproduce the model
22 results from well-known reference data products. The workflows, therefore, act as a
23 means for forced documentation of the steps used to create model input files including

1 the provenance of data as it is transformed from the form provided by federal and
2 academic data repositories into the form required by models.

3 In the case of VIC, and likely in the case of other hydrology applications as this
4 work is extended to include other hydrologic models, the workflows were created by
5 leveraging existing scripts written to complete specific data pre-processing tasks. The
6 approach used provides a means for placing these scripts in larger data processing
7 pipelines and removes the need to access and understand the details of the original scripts
8 to reproduce model results or to reuse the tools for a new study. In the latter case,
9 modification of the rules is only required when selecting a new area of interest. The
10 Pfafstetter Basin codes are replaced in the hydro1kCarolinas.r rule. However, all other
11 workflows are used without modification to automate the remaining data processing
12 steps, which effectively build the input files required for a VIC model.

13 A primary focus of this work is demonstrating a methodological approach to
14 assist in data-intensive hydrologic modeling. Using iRODS has advantages that include
15 workflow automation, access control, data transfers, and data synchronization. iRODS is
16 flexible and robust; it was possible to extend the software by developing rules specific for
17 the data processing pipelines associated with running the VIC model. Because of the data
18 grid concept used by iRODS, it was possible to design workflows in such a way that
19 distributed computers could be leveraged to perform the data gathering and preparation
20 steps. It was also possible to make use of legacy and new software tools for server-side
21 processing of reference datasets. For instance, we applied ecohydroworkflow (Miles, B.,
22 Band, 2013), a shareable workflow for data management for hydrologic models, to

1 collect and register GHCND and HYDRO1k datasets from NCDC and USGS,
2 respectively, in the DFC grid.

3 The rules created within iRODS to enable data pre-processing steps are a key step
4 to achieving reproducible hydrologic model runs. It was possible to chain the data pre-
5 processing routines by creating a rule named vicDataPreprocessing.r in iRODS. This
6 overall rule was designed to call a series of sub-rules, with each sub-rule performing a
7 series of steps required to transform the reference datasets into the specific form required
8 by the model. Doing so provided a level of granularity so that the subrules could later be
9 re-used within other applications in the DFC grid. Having this capability will reduce
10 human errors introduced by manual data transforms. It will also free researchers to devote
11 more time to enhancing, calibrating, and validating models, rather than on tedious steps
12 required to set-up first iterations of the model.

13 A longer-term goal is to allow researchers a means for publishing post-processing
14 workflows that can be used to recreate publication figures using reference datasets stored
15 in a data grid. Creating data post-processing workflows would supplement this work by
16 providing reproducible ways of visualizing large collections of model results. Post-
17 processing rules could result in publication-ready figures that could be reproduced by
18 other researchers. Various stakeholder groups have unique needs, so creating general
19 visualization tools will not always be possible. Creating rules that leverage lower-level
20 micro-services to visualize model results in customized ways could provide a powerful
21 tool provided through iRODS and the DFC cyberinfrastructure.

22 While our approach was to use nested subrules when creating the overall data pre-
23 processing rule to foster reuse, questions remain as to the level of reuse that will be

1 practical across hydrologic simulation models. Hydrologic models can be grouped into
2 classes based on the use cases considered when developing the model. VIC falls into the
3 “macro-scale” class of hydrologic models, meaning it is typically applied to regional,
4 continental, or even global scale hydrologic systems. Other hydrologic models focus on
5 more local scale systems such as a single catchment. We know that different classes of
6 hydrologic models will require different schemes for pre-processing tasks such as
7 discretizing the landscape, and will make use of different reference datasets to set up and
8 parameterize the model. A key challenge moving forward, however, will be to determine
9 the correct level of granularity of rules for data pre-processing that will provide a flexible
10 environment able to support the wide variety of reference datasets and models used by
11 the hydrologic community.

12 **ACKNOWLEDGMENTS**

13 The authors wish to acknowledge support from the National Science Foundation
14 (NSF) under the project DataNet Full Proposal: DataNet Federation Consortium (Award
15 Number:094084 and from Amazon through an Amazon Web Services (AWS) in
16 Education Research grant.

REFERENCES

- 1
- 2 Abdulla, F. a., Lettenmaier, D.P., 1997a. Application of regional parameter estimation schemes to simulate
3 the water balance of a large continental river. *J. Hydrol.* 197, 258–285. doi:10.1016/S0022-
4 1694(96)03263-5
- 5 Abdulla, F. a., Lettenmaier, D.P., 1997b. Development of regional parameter estimation equations for a
6 macroscale hydrologic model. *J. Hydrol.* 197, 230–257. doi:10.1016/S0022-1694(96)03262-3
- 7 Abdulla, F. a., Lettenmaier, D.P., Wood, E.F., Smith, J. a., 1996. Application of a macroscale hydrologic
8 model to estimate the water balance of the Arkansas-Red River Basin. *J. Geophys. Res.* 101, 7449.
9 doi:10.1029/95JD02416
- 10 Billah, M.M., Goodall, J.L., 2011. Annual and interannual variations in terrestrial water storage during and
11 following a period of drought in South Carolina, USA. *J. Hydrol.* 409, 472–482.
12 doi:10.1016/j.jhydrol.2011.08.045
- 13 Billah, M.M., Goodall, J.L., Narayan, U., Reager, J.T., Lakshmi, V., Famiglietti, J.S., 2015. A methodology
14 for evaluating evapotranspiration estimates at the watershed-scale using GRACE. *J. Hydrol.* 523,
15 574–586. doi:10.1016/j.jhydrol.2015.01.066
- 16 Chiang, G.-T., Clapham, P., Qi, G., Sale, K., Coates, G., 2011. Implementing a genomic data management
17 system using iRODS in the Wellcome Trust Sanger Institute. *BMC Bioinformatics* 12, 361.
18 doi:10.1186/1471-2105-12-361
- 19 Crow, W.T., 2003. Multiobjective calibration of land surface model evapotranspiration predictions using
20 streamflow observations and spaceborne surface radiometric temperature retrievals. *J. Geophys. Res.*
21 doi:10.1029/2002JD003292
- 22 Fitch, P., Perraud, J.-M., Cuddy, S., Seaton, S., Bai, Q., Hehir, D., Sims, J., Merrin, L., Ackland, R.,
23 Herron, N., 2011. The Hydrologists Workbench: more than a scientific workflow tool, in:
24 Proceedings, Water Information Research and Development Alliance Science Symposium.
- 25 Foster, I., 2011. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services.
26 *IEEE Comput. Soc.* 15, 70–73. doi:doi:10.1109/MIC.2011.64
- 27 Furnans, J., Olivera, F., 2001. Watershed Topology - The Pfafstetter System. *Esri User Conf.* Vol. 21.
- 28 Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L.,
29 Myers, J., 2007. Examining the Challenges of Scientific Workflows. *Computer (Long. Beach. Calif.)*.
30 40, 24–32. doi:10.1109/MC.2007.421
- 31 Goff, S.A., Vaughn, M., McKay, S., Lyons, E., Stapleton, A.E., Gessler, D., Matasci, N., Wang, L.,
32 Hanlon, M., Lenards, A., Muir, A., Merchant, N., Lowry, S., Mock, S., Helmke, M., Kubach, A.,
33 Narro, M., Hopkins, N., Micklos, D., Hilgert, U., Gonzales, M., Jordan, C., Skidmore, E., Dooley, R.,
34 Cazes, J., McLay, R., Lu, Z., Pasternak, S., Koesterke, L., Piel, W.H., Grene, R., Noutsos, C.,
35 Gendler, K., Feng, X., Tang, C., Lent, M., Kim, S.-J., Kvilekval, K., Manjunath, B.S., Tannen, V.,
36 Stamatakis, A., Sanderson, M., Welch, S.M., Cranston, K.A., Soltis, P., Soltis, D., O’Meara, B., Ane,
37 C., Brutnell, T., Kleibenstein, D.J., White, J.W., Leebens-Mack, J., Donoghue, M.J., Spalding, E.P.,
38 Vision, T.J., Myers, C.R., Lowenthal, D., Enquist, B.J., Boyle, B., Akoglu, A., Andrews, G., Ram, S.,

- 1 Ware, D., Stein, L., Stanzione, D., 2011. The iPlant Collaborative: Cyberinfrastructure for Plant
2 Biology. *Front. Plant Sci.* 2, 34. doi:10.3389/fpls.2011.00034
- 3 Goodall, J.L., Maidment, D.R., 2009. A spatiotemporal data model for river basin- scale hydrologic
4 systems. *Int. J. Geogr. Inf. Sci.* 23, 233–247. doi:10.1080/13658810802032193
- 5 Guru, S.M., Kearney, M., Fitch, P., Peters, C., 2009. Challenges in using scientific workflow tools in the
6 hydrology domain, in: 18th World IMACS Congress and MODSIM09 International Congress on
7 Modelling and Simulation. Cairns, Qld., pp. 3514–3520.
- 8 Hedges, M., Blanke, T., Hasan, A., 2009. Rule-based curation and preservation of data: A data grid
9 approach using iRODS. *Futur. Gener. Comput. Syst.* 25, 446–452. doi:10.1016/j.future.2008.10.003
- 10 Hedges, M., Hasan, A., Blanke, T., 2007. Management and preservation of research data with iRODS.
11 Proc. ACM first Work. CyberInfrastructure Inf. Manag. eScience CIMS 07 17–22.
12 doi:10.1145/1317353.1317358
- 13 Horsburgh, J.S., Tarboton, D.G., Hooper, R.P., Zaslavsky, I., 2014. Managing a community shared
14 vocabulary for hydrologic observations. *Environ. Model. Softw.* 52, 62–73.
15 doi:10.1016/j.envsoft.2013.10.012
- 16 Lakshmi, V., Piechota, T., Narayan, U., Tang, C., 2004. Soil moisture as an indicator of weather extremes.
17 *Geophys. Res. Lett.* 31, 2–5. doi:10.1029/2004GL019930
- 18 Leonard, L., Duffy, C.J., 2013. Essential Terrestrial Variable data workflows for distributed water
19 resources modeling. *Environ. Model. Softw.* 50, 85–96. doi:10.1016/j.envsoft.2013.09.003
- 20 Leonard, L., Duffy, C.J., 2014. Automating data-model workflows at a level 12 HUC scale: Watershed
21 modeling in a distributed computing environment. *Environ. Model. Softw.* 61, 174–190.
22 doi:10.1016/j.envsoft.2014.07.015
- 23 Liang, X., Lettenmaier, D.P., 1994. A simple hydrologically based model of land surface water and energy
24 fluxes for general circulation models. *J. Geophys. Res.* 99, 14,415–14,428. doi:10.1029/94JD00483
- 25 Liang, X., Lettenmaier, D.P., Wood, E.F., 1996a. One-dimensional statistical dynamic representation of
26 subgrid spatial variability of precipitation in the two-layer variable infiltration capacity model. *J.*
27 *Geophys. Res.* 101, 21403. doi:10.1029/96JD01448
- 28 Liang, X., Wood, E.F., Lettenmaier, D.P., 1996b. Surface soil moisture parameterization of the VIC-2L
29 model: Evaluation and modification. *Glob. Planet. Change* 13, 195–206. doi:10.1016/0921-
30 8181(95)00046-1
- 31 Lohmann, D., Raschke, E., Nijssen, B., Lettenmaier, D.P., 1998. Regional scale hydrology: I. Formulation
32 of the VIC-2L model coupled to a routing model. *Hydrol. Sci. J.* 43, 131–141.
33 doi:10.1080/02626669809492107
- 34 Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.,
35 2006. Scientific workflow management and the Kepler system. *Concurr. Comput. Pract. Exp.* 18,
36 1039–1065. doi:10.1002/cpe.994
- 37 Maidment, D.R., 2002. *Arc Hydro: GIS for Water Resources*. ESRI Press.
- 38 Miles, B., Band, L., 2013. EcohydroWorkflowLib. <[http://pythonhosted.
39 org/ecohydroworkflowlib/index.html](http://pythonhosted.org/ecohydroworkflowlib/index.html)> (verified 05.29.2013).

- 1 Moore, R., Rajasekar, A., 2014. Reproducible Research within the DataNet Federation Consortium, in:
2 International Environmental Modelling and Software Society 7th International Congress on
3 Environmental Modelling and Software. San Diego, CA.
- 4 Moriasi, D.N., Arnold, J.G., Liew, M.W. Van, Bingner, R.L., Harmel, R.D., Veith, T.L., 2007. *M e g s q a*
5 *w s* 50, 885–900.
- 6 NASA, 2015. NASA Climate Data Service (NCDS) [WWW Document]. URL <http://cds.nccs.nasa.gov>
7 (accessed 10.16.15).
- 8 Oinn, T., Greenwood, M., Addis, M., Alpdemir, M.N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull,
9 D., Marvin, D., Li, P., Lord, P., Pocock, M.R., Senger, M., Stevens, R., Wipat, A., Wroe, C., 2006.
10 Taverna: lessons in creating a workflow environment for the life sciences. *Concurr. Comput. Pract.*
11 *Exp.* 18, 1067–1100. doi:10.1002/cpe.993
- 12 Perraud, J., Fitch, P.G., Bai, Q., 2010. Challenges and Solutions in Implementing Hydrological Models
13 within Scientific Workflow Software. AGU Fall Meet. Abstr. -1, 06.
- 14 Piasecki, M., Lu, B., 2010. Development of a Hydrologic Modeling Platform Using a Workflow Engine, in:
15 AGU Fall Meeting Abstracts. p. 1239.
- 16 Rajasekar, A., Moore, R., Hou, C.-Y., Lee, C. a., Marciano, R., de Torcy, A., Wan, M., Schroeder, W.,
17 Chen, S.-Y., Gilbert, L., Tooby, P., Zhu, B., 2010a. iRODS Primer: Integrated Rule-Oriented Data
18 System, Synthesis Lectures on Information Concepts, Retrieval, and Services.
19 doi:10.2200/S00233ED1V01Y200912ICR012
- 20 Rajasekar, A., Moore, R., Wan, M., Schroeder, W., 2010b. Policy-based Distributed Data Management
21 Systems. *JODI J. Digit. Inf.* 11, 1–16.
- 22 Sheffield, J., Goteti, G., Wen, F., Wood, E.F., 2004. A simulated soil moisture based drought analysis for
23 the United States. *J. Geophys. Res. D Atmos.* 109, 1–19. doi:10.1029/2004JD005182
- 24 Sheffield, J., Wood, E.F., 2007. Characteristics of global and regional drought, 1950-2000: Analysis of soil
25 moisture data from off-line simulation of the terrestrial hydrologic cycle. *J. Geophys. Res. Atmos.*
26 112, 1–21. doi:10.1029/2006JD008288
- 27 Troy, T.J., Wood, E.F., Sheffield, J., 2008. An efficient calibration method for continental-scale land
28 surface modeling. *Water Resour. Res.* 44, 1–13. doi:10.1029/2007WR006513
- 29