A Taxonomy for Reproducible and Replicable Research in Environmental Modelling

Bakinam T. Essawy[a,1], Jonathan L. Goodall[a*], Daniel Voce[b], Mohamed M. Morsy[a,c,1], Jeffrey M. Sadler[a,2], Young Don Choi[a] , David G. Tarboton[d], and Tanu Malik[e]

[a] Department of Engineering Systems and Environment, University of Virginia, 151 Engineers Way, P.O. Box 400747, Charlottesville, VA, 22904, USA

[b] Department of Electrical and Computer Engineering, University of Virginia, 351 McCormick Road, PO Box 400743, Charlottesville, VA, 22904, USA

[c] Irrigation and Hydraulics Engineering Department, Faculty of Engineering, Cairo University, P.O. Box 12211, Giza 12614, Egypt

[d] Utah Water Research Laboratory, Utah State University, Logan, UT 84322, USA

[e] College of Computing and Digital Media, DePaul University, Chicago, IL 60604, USA

* To whom correspondence should be addressed (E-mail: *goodall@virginia.edu*; Address: University of Virginia, Department of Engineering Systems and Environment, University of Virginia, 151 Engineers Way, P.O. Box 400747, Charlottesville, VA, 22904, USA; Tel: (434) 243-5019)

[1] Now with Dewberry, 8401 Arlington Boulevard, Fairfax, VA 22031-4666, USA

[2] Now with US Geological Survey, Middleton, WI, USA

This is the accepted version of the following published article. Please refer to the published article for the final version of the manuscript.

Essawy, B.T., Goodall, J.L., Voce, D., Morsy, M.M., Sadler, J.M., Choi, Y.D., Tarboton, D.G., Malik, T., 2020. A taxonomy for reproducible and replicable research in environmental modelling. Environ. Model. Softw. 104753. https://doi.org/10.1016/j.envsoft.2020.104753

Highlights:

- Defines a taxonomy describing levels of reproducibility for modelling studies
- Levels progress from repeatability to runnability, reproducibility, and replicability
- An analysis using the SUMMA hydrologic model is used to demonstrate the taxonomy
- Containerization of the conditions of analysis allows for achieving reproducibility
- Sciunit software facilitates this containerization of the conditions of analysis

Abstract

Despite the growing acknowledgment of reproducibility crisis in computational science, there is still a lack of clarity around what exactly constitutes a reproducible or replicable study in many computational fields, including environmental modelling. To this end, we put forth a taxonomy that defines an environmental modelling study as being either 1) repeatable, 2) runnable, 3) reproducible, or 4) replicable. We introduce these terms with illustrative examples using the Structure for Unifying Multiple Modeling Alternatives (SUMMA) hydrologic modelling framework along with cyberinfrastructure aimed at fostering reproducibility. Using this taxonomy as a guide, we argue that containerization is a key missing component in environmental modelling and is necessary to achieve the goal of computational reproducibility. The provided examples demonstrate how new tools, including a user-friendly tool for containerization of computational analyses called Sciunit, can lower the barrier to reproducibility and replicability in the environmental modelling community.

Keywords

reproducibility; replicability; containers; Docker; Singularity

Software Availability

The analysis example illustrated in this research is available free and open source, under an MIT license, from GitHub at https://github.com/uva-hydroinformatics/pysumma-sciunit.

1. Introduction

Emphasis on the importance of research reproducibility is steadily rising, yet many studies are still not reproducible (Bell et al., 2009; Garijo et al., 2013; Hothorn and Leisch, 2011; J. H. H. Stagge et al., 2019). One study found that 70% of researchers tried but failed to reproduce another researcher's experiments (Baker, 2016). This failure was largely due to lack of documentation and the omission of important details from the published article. Additionally, numerous studies have concluded that scientific articles commonly leave out details essential for reproduction (Ioannidis et al., 2009; Nekrutenko and Taylor, 2012; J. H. H. Stagge et al., 2019). Even when these details are in place, complex computational studies can take hundreds of hours to reproduce, especially for a scientist not involved in the original study (Baggerly and Coombes, 2009). Reproducibility in such computational studies is particularly difficult because they rely on nontrivial software systems with multiple software dependencies. As computational studies become more complex and dependent on a variety of software systems, this difficulty leads to the "reproducibility crisis" being experienced across fields where scientists are unable to reproduce analyses from the information provided in articles and software documentation alone (Baker, 2016; Baker and Penny, 2016). Some scientific disciplines (including biology, biostatistics, and biomedicine) have been discussing this reproducibility crisis for some time (Peng, 2011), while in other fields, such as Hydrology and Water Resources, the topic is still being brought to light (Hutton et al., 2016; Rosenberg et al., 2020; J. H. Stagge et al., 2019). Using Hydrology and Water Resources as case study, this paper addresses reproducibility challenges applicable in the broader field of Environmental Modelling by offering a taxonomy to aid in the interdisciplinary communication around reproducibility as well as an approach involving containerization of computational experiments to enhance reproducibility.

One of the challenges in discussing the reproducibility crisis in computational modelling fields is the lack of a standard definition for reproducibility. Several scientists have made efforts to define reproducibility (Easterbrook, 2014; Goodman et al., 2018; Gorgolewski and Poldrack, 2016; Ioannidis et al., 2009; Stodden et al., 2013). The problem has recently resulted in a high-level report by the United States (US) National Academies in collaboration with the US National Science Foundation (hereafter, referred to as the National Academies report) (National Academies of Sciences Engineering and Medicine, 2019). The National Academies report acknowledges the lack of a standard definition for computational reproducibility across scientific fields. To address this problem, it offers a standard definition for reproducibility along with a higher-level concept named replicability. The report defines computational reproducibility as "obtaining consistent results using the same input data, computational steps, methods, code, and conditions of analysis" and replicability as "obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data" (National Academies of Sciences, Engineering, and Medicine, 2019). We have adopted these definitions in this paper, but argue that there are stages in environmental modelling, and likely other computational modelling fields as well, before one reaches "reproducible" as defined in the National Academies report. Without consistent definitions of these earlier stages, there will continue to be confusion on what exactly constitutes a reproducible computational study.

As an example of this confusion, a common explanation for achieving reproducibility in computational fields is a study that has all code and data shared with the published article, but this explanation is incomplete for computationally complex studies. The idea is widely held as evidenced by many scientific journals now requiring authors to provide "Data Availability" statements providing Digital Object Identifiers (DOIs) for data and models used in the analysis. However, this requirement alone will not ensure that a computational study is reproducible. For

example, suppose a researcher publishes an article and shares the associated data and models through an open data repository. If other researchers have the necessary digital resources and try to reproduce this research using the paper, data, and models, they may be unable to do so if the first researcher did not share exact details of the "conditions of analysis," as the National Academies report (National Academies of Sciences, Engineering, and Medicine, 2019) puts it, used to complete the study. The conditions of analysis speak to the computation environment used to conduct the experiment including the operating system used, but also the exact versions of all software and software dependencies used in the study. Even in simple cases where only one software system is used for an analysis (e.g., a single simulation model or single analysis software like Python or R), because software is being constantly updated and dependencies may change, sharing the data and code for that simulation alone may not be sufficient for achieving reproducibility if not all software dependencies are shared too. One needs a way to create a self-contained package or container of the conditions of analysis that preserves these conditions that were present during experimentation so that, even if the container is moved to different computational environments, the experiment continues to re-execute as originally designed.

Any alternative way researchers have proposed to address the general confusion about different levels of reproducible research is by organizing studies into three categories: low, medium, and high reproducible research (Tatman et al., 2018; Peng, 2011). Low reproducible research occurs when a well written article, which may include a detailed methodology section, does not provide the digital resources (such as code, data, and computational environment) necessary to reproduce that research. When one of these resources, typically the data associated with the article, is published but the others are withheld, this research is classified as being "medium reproducibility" (Tatman et al., 2018). The highest level of reproducibility, referred as

the “gold standard of reproducibility” by Peng (2011), occurs when all codes, data, and environments used in the research are both described in the published article and shared.

These three levels of defining reproducibility provide a useful way to classify studies, but they tend to focus on the sharing of digital objects rather than on the functional aspects of reproducibility. We argue that the sharing of digital objects is a prerequisite to reproducible research. The real challenge and need for categorical distinctions along a reproducibility spectrum should target functional requirements showing the levels at which computational experiments can be rerun by the original authors and, ultimately, by others to reproduce and replicate the study results. For studies that require only minimal software tools, it seems feasible for researchers to achieve reproducibility without spending a significant amount of additional time and effort beyond gathering digital objects and setting up virtual environments for an analysis. For example, the concept of virtual environments in Python that allow one to package the specific version of Python and dependent libraries used in an analysis is becoming common place with tools like Anaconda (https://www.anaconda.com). However, when an analysis makes use of more than just a single software system, capturing the conditions of analysis (National Academies of Sciences Engineering and Medicine, 2019) or computational environment (Peng, 2011) that includes all software dependencies becomes more complicated. It can often be unclear even to the most sophisticated modeler exactly what software was used by a model or larger analysis workflow, given the complex dependencies and sub-dependencies with many scientific software systems, including modelling systems.

Container technologies offer a solution for this problem (Handigol et al., 2012; Knoth and Nüst, 2017; Piccolo and Frampton, 2016). Docker (Merkel, 2014) and Singularity (Kurtzer et al., 2017) are the most common container technologies, but these are programmable tools and, for many modelers, require a steep learning curve for how to work with them, which is often a

function of the complexity of the software program being contained. In addition, because it is unclear exactly what software to include in a container to make an analysis reproducible, scientists may overestimate the software needed, making for bloated containers that become more like virtual machines, thus losing one of the main benefits of container technologies. For this reason, there are efforts in scientific communities to create cyberinfrastructure technologies and approaches better suited for computational experiments (Brinckman et al., 2019; Marwick et al., 2018; Stodden et al., 2015) including efforts to lower the barrier to container technology adoption (Kjeldgaard, 2020; Nüst et al., 2017; Nüst and Hinz, 2019). The Geotrust project funded through the US National Science Foundation EarthCube program that is advancing Sciunit (http://sciunit.run) is one such effort (Chuah et al., 2020; That et al., 2017; Yuan et al., 2018). Using Sciunit, a tool for tracing and encapsulating dependencies in a Linux environment, researchers can create virtual environments for computational analyses that only include the software dependencies used within the analysis (That et al., 2017). Sciunit acts as a monitoring system, recording what software dependencies were used by an analysis and then packaging these dependencies into a virtual environment. In capturing dependencies during program execution, Sciunit is unlike Docker, which uses programmatic methods to capture dependencies. Sciunit also uses content-based data de-duplication (Yuan et al., 2018), and is thus more efficient to use than Docker (Meng et al., 2015).

In past work we have shown how Geotrust's Sciunit can be combined with HydroShare to improve reproducibility in computational analyses (Essawy et al., 2018). Here we extend on this idea to provide a more complete view of best practices for reproducible and replicable computational analyses. First, we present a taxonomy to capture the reproducibility spectrum for environmental modelling beginning with running an analysis on a single machine and ending with reproducing and replicating that study. Then, we demonstrate the path along this spectrum

with an analysis using the Structure for Unifying Multiple Modeling Alternatives (SUMMA) hydrology model (Clark et al., 2015a, 2015b) as an example application. As part of this example, we demonstrate how the Geotrust Sciunit tool and open data repositories, such as HydroShare, can be leveraged to achieve both reproducibility and the higher-level concept of replicability. A discussion follows that first speaks to nuances in the pursuit of reproducible computational analyses, namely the idea that reproducibility does not require an exact match in a computational environment or the results of the analysis, but instead requires results that are, as the National Academies report states, "consistent." This makes knowing when reproducibility has been achieved more challenging because quantifying an exact match is computationally simpler than quantifying a consistent match. This section also outlines gaps that remain in achieving reproducibility of environmental modelling analyses that should be addressed through future research and development efforts. Finally, we conclude with a summary of the contributions of this study, which are mainly a clearer definition of a taxonomy for describing environmental modelling studies as repeatable, runnable, reproducible, or replicable and an illustrative example for how to move analyses along this taxonomy so that modelers can be more confident that their studies are reproducible rather than being only repeatable or runnable.

2. Methodology

2.1. Defining a Taxonomy for Reproducible Environmental Modelling

The first objective in this study was to define an appropriate taxonomy for capturing the spectrum of reproducibility for computational environmental modelling. To do this, we reviewed and synthesized ideas from literature. We also adopted the definitions provided by the National Academies report as this is a high-level report aimed at clarifying two concepts in particular: reproducibility and replicability. We expanded on the definitions provided in this report by

adding additional explanation of these definitions and, importantly, the common steps in environmental modelling that come before the reproducibility step. The result of this work is a four level taxonomy of repeatability, runnability, reproducibility, and replicability described more fully in the results section of the paper and used to organize an example application where a hydrologic modelling analysis moves through the stages in the taxonomy each representing a step along the reproducibility spectrum.

In designing the taxonomy, we had two main goals. The first goal was to allow researchers to better distinguish the level of reproducibility of computational studies by providing clear definitions for placing a given study along a reproducibility spectrum. While past researchers have also used the concept of a reproducibility spectrum (Peng, 2011), designating studies as either reaching low, medium, or high reproducibility, we designed the taxonomy to focus on functionality rather than data availability. Our approach views availability as a prerequisite for reproducibility rather than a distinguishing characteristic. The second goal of the taxonomy was to not only place an analysis along the reproducibility spectrum, but also understanding what additional steps are required to advance to the next level in the reproducibility spectrum. Through an example application presented in this paper, we demonstrate how research can advance an analysis along the spectrum.

2.2. Steps for Creating a Reproducible Analysis

The second objective in this study was to design a general approach for advancing computational analyses common in environmental modelling along the reproducibility spectrum. The obvious goal of any modelling study is to get a simulation running on the researcher's own machine. Once this has been achieved, the next step should be to repeat this analysis on a new machine. In repeating an analysis on a different machine, the researcher will be forced to

overcome potential difficulties that another researcher may face when attempting to replicate the study. This step allows the researcher to document steps more clearly and completely by overcoming potential problems that another research may face when trying to setup and reproduce the study on a different machine. While a researcher must share these resources, reproducibility cannot be claimed until a third party has verified that it is possible to reproduce an analysis. The more researchers able to verify that an analysis can be reproduced in their own computational environment, the more confidence one can have that the work has achieved reproducibility.

While a researcher can make an analysis reproducible with sufficient documentation and sharing of digital resources, for complex analyses doing so is time consuming, challenging to do correctly, and difficult to keep up to date. Figure 1 illustrates this point with a simple example. Suppose Researcher A shares completed work in the form of a published article with associated code and data. Because the full computational environment capturing the conditions of analysis, including all dependencies, is not shared with this study, Researchers B, C, and D must try and recreate the computational environment and dependencies on their own. Research B has a different operating system, Research C has a different core software system for the analysis, and Researcher D has the wrong software dependencies. As a result, none of the researchers are able to reproduce the results of the analysis. Without detailed documentation on required software dependencies and the broader conditions of analysis, it may not even be clear to Researchers B, C, and D why they are unable to reproduce the results of Researcher A.

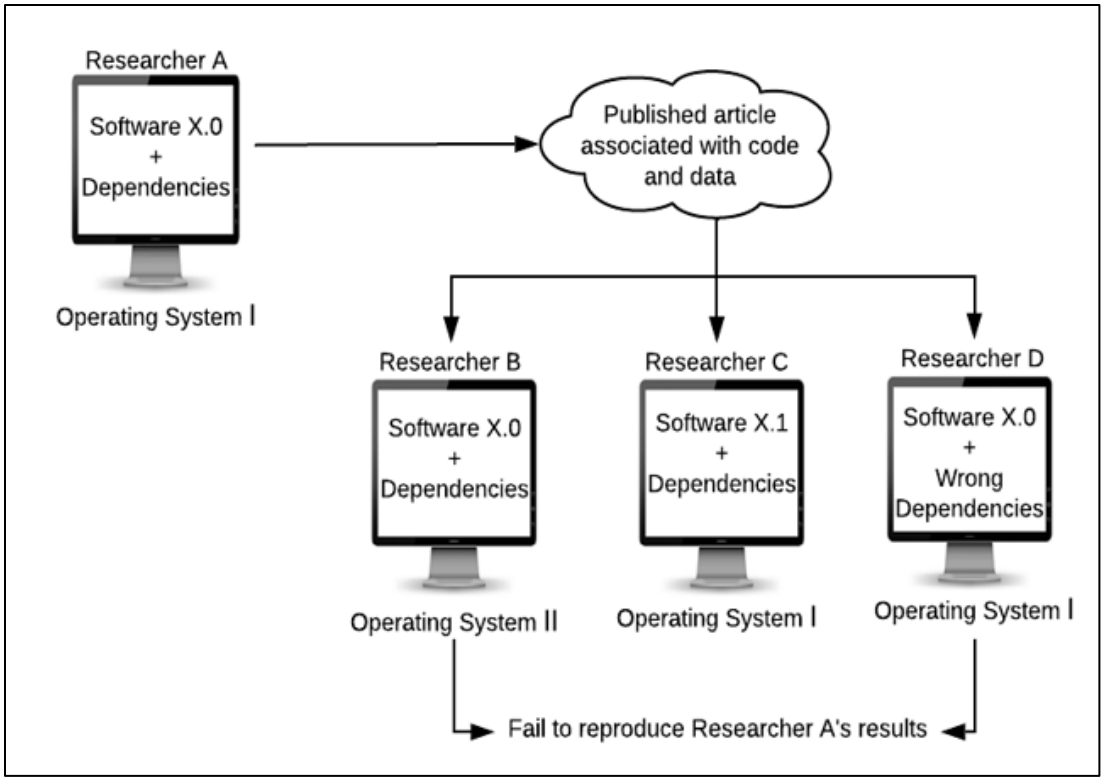

Figure 1. An example of why reproducibility can be challenging due to underlying software dependencies and simply sharing data and code is insufficient for achieving computational reproducibility.

The following methodology allows Researcher A to better share her analysis in a way that can be more easily reproduced. This method emphasizes the use of a container to encapsulate a virtual environment and open data repositories for sharing digital resources. These are critical enabling technologies and tools for advancing reproducibility in a research context where the software is part of the experiment and the environmental modelling is conducted with research simulation models that do not have a standard release schedule and installation package. The simulation models used by scientific teams for research within the scientific community are often

developed in a bespoke fashion--- often advanced by different research teams, and updated in a non-coordinated manner. Thus, unlike commercial simulation software that has more formalized release schedules, continuous integration testing, and installers, simply getting some scientific software properly installed while also knowing what version of the software is being used is a nontrivial exercise. For such bespoke software, it is increasingly important to snapshot a working execution and make the snapshots reproducible for others in different environments. The steps illustrated in Figure 2, and described below in a general, model agnostic way, demonstrate this need. The Results section includes results when applying of this methodology for a specific modelling analysis.

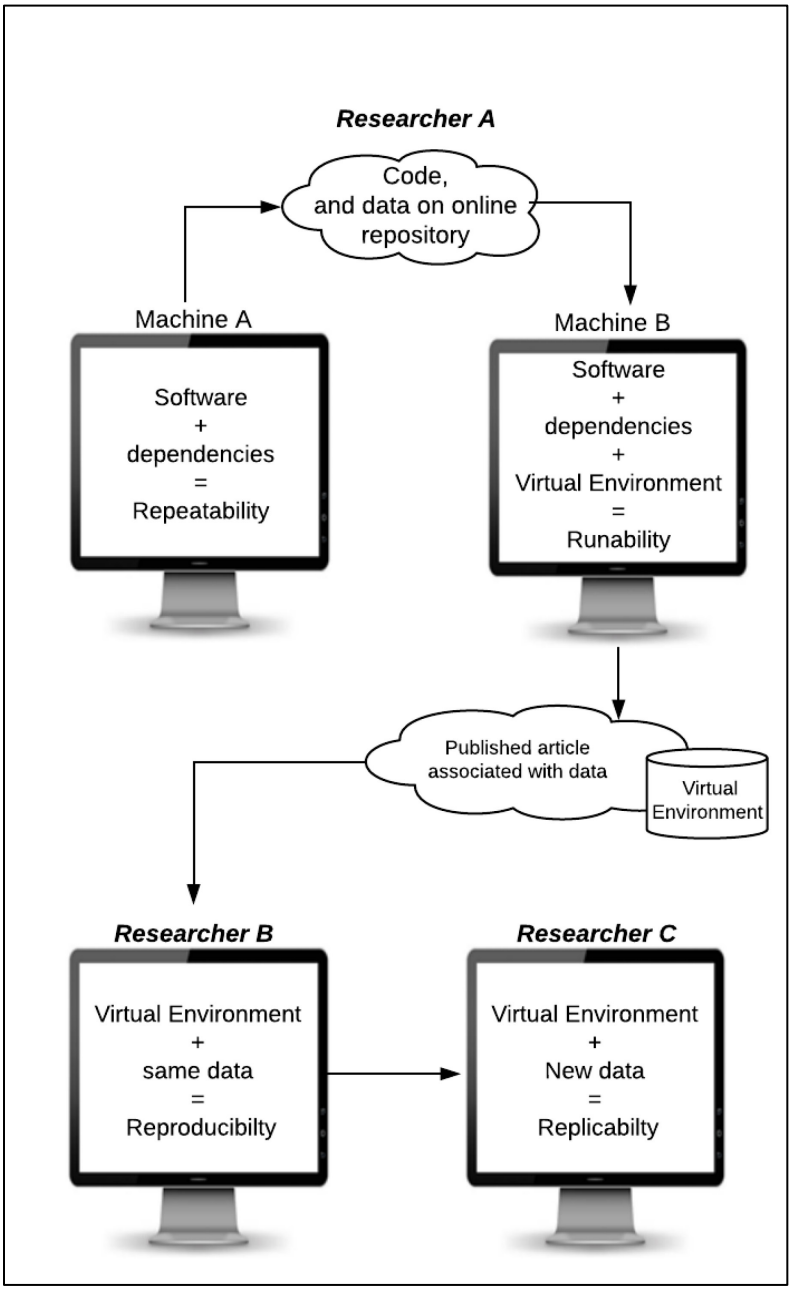

Figure 2. General steps necessary for ensuring that reproducibility is achieved.

1.) Researcher A creates the analysis on her local machine and repeats it several times to ensure the analysis is working correctly. From here we can say repeatability is achieved.

2.) Researcher A moves the analysis to another machine and repeats Step 1. Completing this step will allow Researcher A to better understand and document the conditions of analysis so that anyone attempting to reproduce the study can also run the analysis on their own computational environment.

3.) Researcher A, acknowledging that it is time consuming to perfectly document the steps needed to recreate the conditions of the analysis based on performing Step 2, creates a containerized virtual environment that packages the conditions of analysis as a reproducible unit. This containerized virtual environment is shared with the community alongside the article in an open repository with its own digital object identifier (DOI) and metadata, as shown in Figure 3.

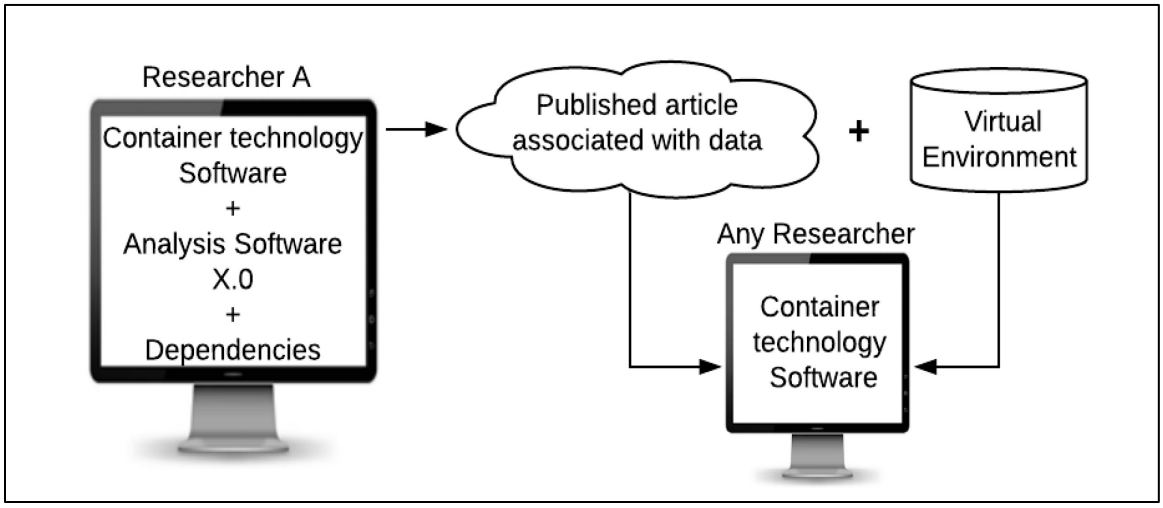

Figure 3. Details for Step 3 in the method for achieving reproducible analyses where a researcher publishes not only the article with associated data, but also a virtual environment that can be run using container technology on their own machine.

4.) Other researchers download this containerized virtual environment and required input data to their own machines to verify that they can obtain Researcher A's results. The more scientists able to reproduce the analysis in their own environment, the more confidence Researcher A has that her work has reached what Peng (2011) calls the reproducibility gold standard.

5.) Once researchers have been able to reproduce the analysis, they may now wish to replicate it by using the same analysis but with their own data sets. Because they were first able to reproduce the study on their own, they can be more confident in reusing the analysis to build from past work and advance their own research.

2.3. Leveraging Advances in Cyberinfrastructure to Create Reproducible Workflows

The third objective in this study was to effectively leverage cyberinfrastructure to reduce the burden scientists have in creating reproducible analyses. Achieving reproducibility remains time consuming and tedious due to a gap in technology. Recent advances in the broader information technology field can be used to assist researchers in better documenting and sharing code, data, metadata, and building a virtual environment to achieve reproducibility. Some scientists have begun using these new tools and approaches with the aim of making their research more reproducible, and they have discussed the reproducibility of their research in their published articles (Ivie and Thain, 2018; Sadler et al., 2018; Woodson et al., 2018). In these studies, online data repositories such as GitHub, Figshare, Zenodo, and HydroShare are becoming more commonly used by scientists to describe and share their data and code. Many of these repositories provide the ability to publish digital resources through their system assigning a digital object identifier (DOI) to the resource that can be used to uniquely identify that resource in perpetuity. Containers and virtual environments are still uncommon tools in scientific studies,

perhaps because of their steeper learning curve for use in computational modelling and analysis studies.

While there are a growing number of tools available to scientists to make their work more reproducible and replicable, we focus on two specific systems: (i) the Sciunit tool (http://sciunit.run) and (ii) the HydroShare online repository (http://www.hydroshare.org). The Sciunit tool is software that enables researchers to create virtual environments for reproducing their computational analyses with minimal user interaction. Sciunit advances the concept of a research object, which is an automatic aggregation of digital artifacts such as code, data, scripts, and temporary experiment results that together with any research paper provide an authoritative and far more complete record of a piece of research (Bechhofer et al., 2013). HydroShare is an open repository for sharing hydrologic data and models as digital resources, including detailed, hydrologic-specific resource metadata (Horsburgh et al., 2015; Tarboton and Idaszak, 2015).

We showed in prior work how combining these two tools, Sciunit and HydroShare, can improve reproducibility for computational analyses (Essawy et al., 2018). We showed how Sciunit can be used to create containerized virtual environments, but lacks mechanisms for sharing them within a community of users. HydroShare, on the other hand, allows for sharing codes, data, and descriptive metadata, but it does not address the challenge of packaging virtual environments for reproducing complex computational analyses. Building from this research, here we extend this work to the more complete taxonomy for describing repeatable, runnable, reproducible, and replicable studies, illustrating how an environmental modelling study advances along this taxonomy and how tools like Sciunit and HydroShare enable studies to move from being repeatable, to runnable, then reproducible, and finally replicable.

Applying the general five-step procedure outlined in the prior section, the Sciunit tool is used to capture, encapsulate, and make the model execution portable by creating Sciunit

containers while the HydroShare repository is used to share the resulting Sciunits and other key digital resources in the analysis. More specifically, the Sciunit tool is used to automatically containerize (or package) and share the virtual environment to HydroShare as a *sciunit* and then a HydroShare compute platform such as CUAHSI JuypterHub or CyberGIS Jupyter for Water is used to interact with the shared Sciunit through a Jupyter notebook (Figure 4). These JupyterHub implementations, because of their ability to read and write data from and to HydroShare, provides a powerful analysis and model execution environment (Bandaragoda et al., 2019). The advantage of this methodology is: 1) it reduces the time necessary to download, install, and run the workflow on local machines, 2) scientists don't face dependency issues such as missing or out-of-date dependencies, and 3) it avoids local computing restraints that scientists face such as limited memory or storage.

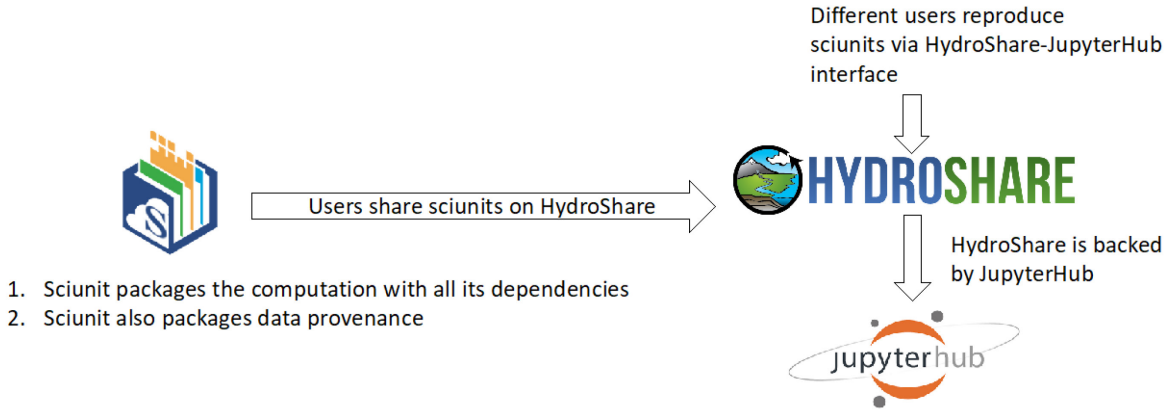

Figure 4. Interaction between the Sciunit Tool and HydroShare.

2.4. Example Application using the SUMMA Hydrologic Model

An example application is presented to illustrate how the five-step procedure presented in the prior section can be applied with Sciunit and HydroShare for an environmental modelling

analysis. The example uses the Structure for Unifying Multiple Modelling Alternatives (SUMMA) hydrologic modelling framework to simulate different hydrologic hypotheses and show how to reproduce the model simulation process (Clark et al., 2015a, 2015b). The SUMMA structure enables users to implement different modelling approaches with controlled and systematic analysis and provide insight for the advanced unified modelling framework. Figure 5 shows the SUMMA model construction and process flexibility. The SUMMA structure consists of a core (solver) with outer branches and produces a numerical solution with conservation equation from water and energy flux and state. The SUMMA model provides flexibility to evaluate the interplay between the choice of model parameters and the choice of process parameterizations, separating modelling decisions on process representation from their numerical implementation, and providing capabilities to experiment with different numerical solvers.

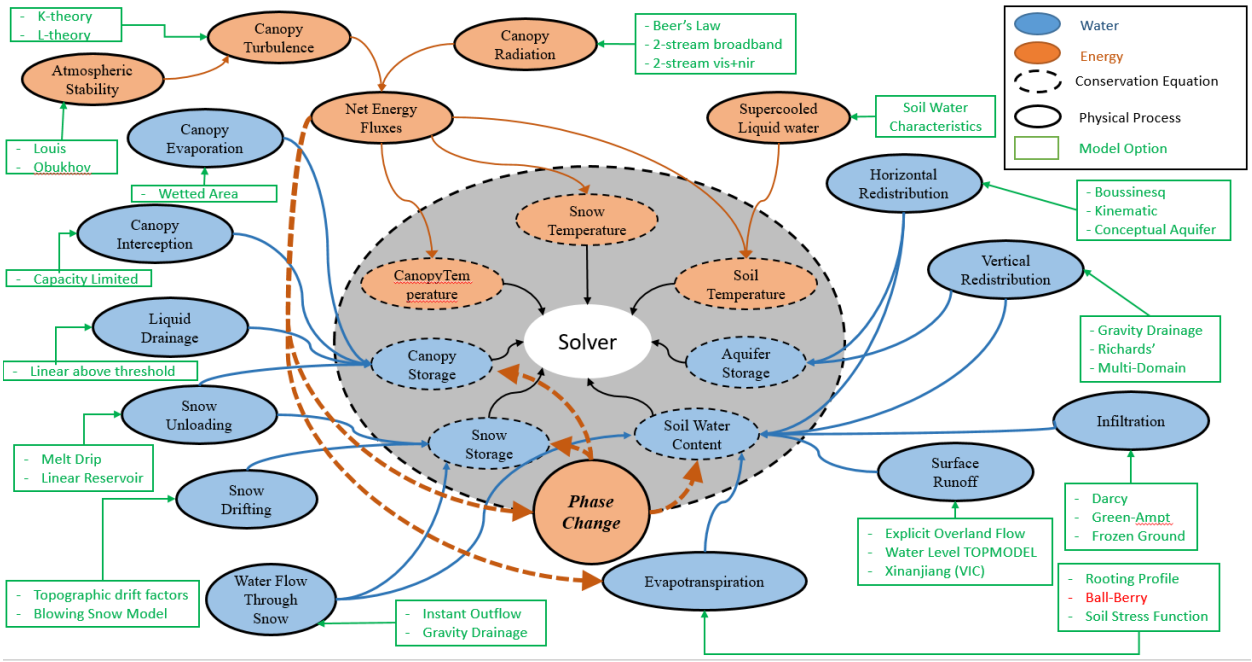

Figure 5. The SUMMA Model Construction and Process Flexibility (Clark et al., 2015a, 2015b).

Building from four synthetic and nine field study test cases of SUMMA described in Clark et al. (Clark et al., 2015b), we selected and automated two of the test cases (1) Field Data Test Case 5: Snow interception at Umpqua and (2) Field Data Test Case 7: Sensitivity of evapotranspiration to the stomatal resistance parameterization (Aspen stand at Reynolds Mountain East). These two test cases are described by in Clark et al. (Clark et al., 2015b) and the digital resources needed to reproduce the paper are available online. However, the test cases were not reproducible as defined by the National Academies report because the conditions of analysis were not completely described and documented. We demonstrate in the Results section how following the five-step procedure described earlier and using the Sciunit and HydroShare, one can move the SUMMA analysis from being repeatable, to being runnable, and finally to being reproducible and replicable.

3. Results

3.1. Taxonomy for Reproducibility in Environmental Modelling

The taxonomy resulting from this work organizes the reproducibility spectrum into four levels: repeatability, runnability, reproducibility, and replicability (Figure 6). These levels represent a progression where the base level, repeatability, is the first step to achieve, followed by runnability, reproducibility, and, finally, replicability. The terms reproducibility and replicability are consistent with those proposed in the National Academies report. We argue that the two lower-level concepts, repeatability and runnability, are needed as precursors to correctly capture the steps along the path to reproducibility and replicability. The four levels are defined below and illustrated more fully later in this section using the SUMMA model as a case study.

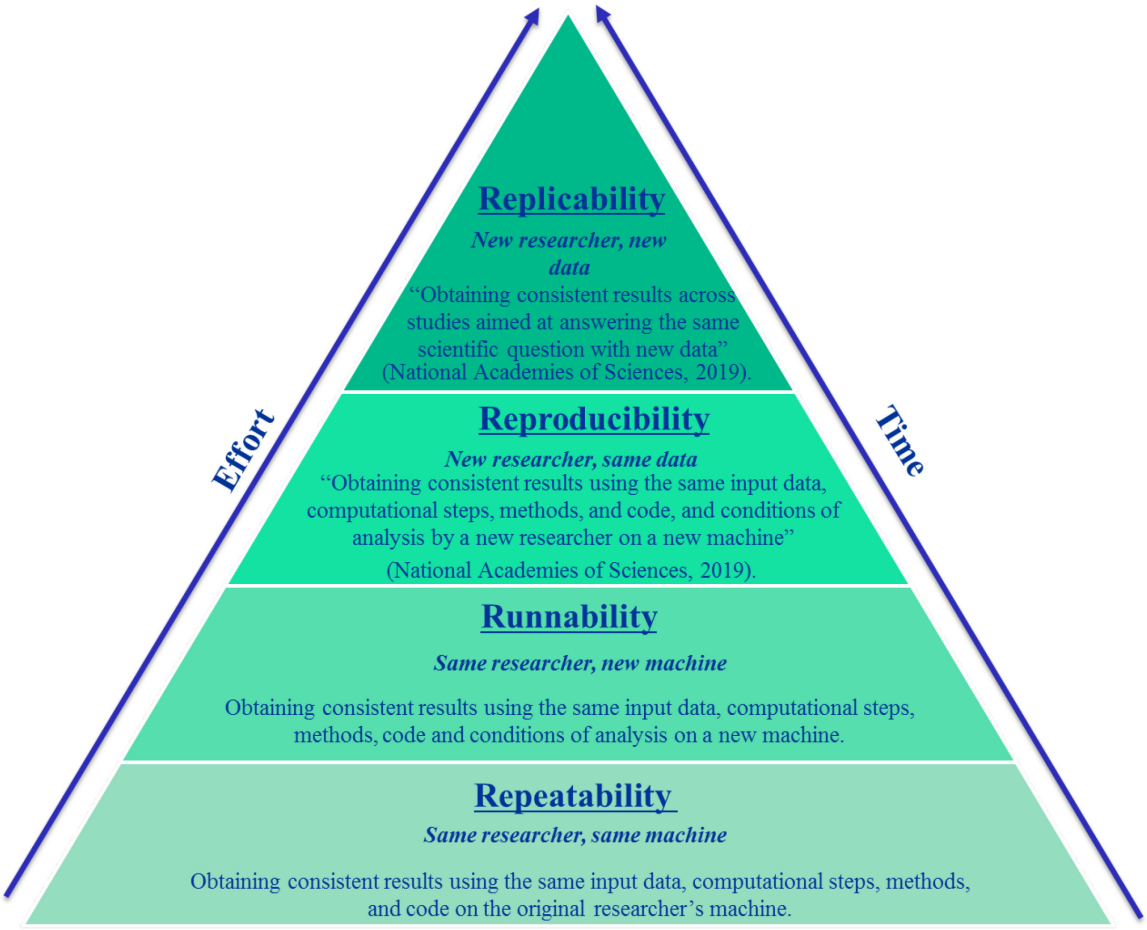

Figure 6. The reproducibility taxonomy for complex computational studies comprising a progression that requires increased effort and time from repeatability, through runnability, reproducibility, and replicability.

- *Repeatability* is achieved upon obtaining consistent results using the same input data, computational steps, methods, and code on the original researcher's machine. This level is normally achieved in scientific papers; the author is able to rerun the analysis on his or her own computer to obtain the analysis results. However, it does require detailed documentation and benefits greatly from automation of steps to make repeating the analysis in a consistent way possible.

- *Runnability* is achieved when the author of the research can obtain consistent results using the same input data, computational steps, methods, code *and conditions of analysis on a new machine*. Achieving this level requires thought and care to document the conditions of the analysis, meaning software dependencies and related details of the computational environment, in such a way that the analysis can be repeated correctly in a new computing environment.
- *Reproducibility* is achieved when a new researcher, not an original author of the analysis, is able to reproduce the analysis in their own computational environment. Achieving this step shows that one is able to achieve the National Academies report definition of reproducibility: "obtaining consistent results using the same input data, computational steps, methods, and code, and conditions of analysis."
- *Replicability* is the higher-level concept defined in the National Academies report as "obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data." Replicability also allows scientists not involved in the original study to build from and expand on research once they are first able to reproduce that research.

Note that the concept of results being consistent, used in these definitions, adopted from the National Academy report is less rigid than exact numerical equivalence. Rather it involves some degree of evaluation as to how close is good enough, acknowledging numerical differences that arise on different platforms. This concept is addressed more fully in the discussion section of the paper.

3.2. Modelling Example Case Study: Repeating the SUMMA Analysis

In this section we present a modelling example case study that steps through the four levels of the taxonomy using the general methodology described in Section 2.2 applied for the specific environmental modelling use case described in Section 2.4 that leverages the SUMMA model.

The first step is for the author to repeat an analysis on her own machine and obtain consistent results. We did this for Test Case 7: Sensitivity of evapotranspiration to the stomatal resistance parameterization (Aspen stand at Reynolds Mountain East described in Clark et al. (Clark et al., 2015b). For this test case, we created an automated analysis using the SUMMA Python wrapper software pySUMMA (Choi et al., 2018) to model the sensitivity of evapotranspiration to the stomatal resistance parameterization for the Aspen stand at the Reynolds Mountain East study site. We installed the SUMMA model by following the steps in the SUMMA Installation documentation (https://summa.readthedocs.io/en/latest/installation/SUMMA_installation/).

After we installed the SUMMA model, we developed a workflow using pySUMMA to model the impact of stomatal resistance parameterizations on total evapotranspiration. Figure 7 shows the script used to set up and run the Test Case 7 SUMMA model simulation using pySUMMA and create a visualization of the model. This script shows how a SUMMA model can be initialized, how the model configuration can be adjusted with single lines (e.g., setting the stomata resistance function to 'Jarvis'), how SUMMA can be executed, and how the results can be visualized all from a single Python script. While these steps can be completed without pySUMMA through manipulating input files for a SUMMA model directly, pySUMMA acts as a wrapper for these text files and as a means for documenting a model configuration for a specific experiment.

```
from pysumma.Simulation import Simulation
import subprocess
import matplotlib.pyplot as plt
from pysumma.Plotting import  Plotting

# create a pySUMMA simulation object using the SUMMA 'file manager' input file
S = Simulation('/home/ubuntu/figure07/summa_fileManager_riparianAspenSimpleResistance.txt')

# set the simulation start and finish times
S.decision_obj.simulStart.value = "2007-10-01 01:00"
S.decision_obj.simulFinsh.value = "2008-10-01 00:00"
S.executable = "/home/ubuntu/summa/bin/summa.exe"
S.decision_obj.stomResist.value = 'Jarvis'
results_simpleResistance, out_file = S.execute(run_suffix="rootDistExp", run_option = 'local')

# Visualization
P = Plotting(out_file)
P.plt('scalarCanopyTranspiration')
```

Figure 7. The script "simulation_object.py" is used to run SUMMA Test Case 7 model simulation and create a visualization.

The result from running the script shown in Figure 7 shows the plot generated from the analysis. This can be compared to the plot that appears in Clark et al. (Clark et al., 2015b) to gauge if the results are consistent. In this case, consistent results were achieved and, therefore, we can conclude that the analysis has reached the repeatable stage. We are assuming for the sake of this paper that we are playing the role of the original author in this study as the original paper did not provide details on repeatability.

3.3. Making the Analysis Runnable using Sciunit and CUAHSI HydroShare JupyterHub

Once the analysis was repeated, the next step was to make the analysis runnable on a separate machine. To do this, we made use of the Sciunit tool to package the analysis along with its dependencies and provenance metadata. Figure 8 shows the steps to package the workflow analysis using the Sciunit tool. These steps are: 1.) create a new *Sciunit* "MyAnalysis." This will create a virtual directory, which will include the captured execution of the computational workflow with all the dependencies and provenance metadata associated with it; 2.) open the "MyAnalysis" Sciunit to begin working in the desired Sciunit; 3) execute the code required to be

packaged as a virtual environment in order to repeat the analysis; 4.) show the details of the virtual environment encapsulated in a Sciunit package; 5.) place the packaged Sciunit on HydroShare as a digital resource, and 6.) test the runnability of the package by executing the Sciunit on the CUAHSI HydroShare JupyterHub app linked to HydroShare and configured to open and execute scripts acting on content from Resources in HydroShare (Note: To run a Sciunit again requires the Sciunit tool, which is installed on CUAHSI HydroShare JupyterHub). This initial test of this execution by us (playing the role of the original author) on the separate JupyterHub app computer demonstrating runnability is detailed in the following paragraph. Once the Sciunit is shared on HydroShare, users of HydroShare can interact and execute the shared Sciunit from HydroShare using JupyterHub, thereby achieving reproducibility. Doing so would achieve the reproducibility gold standard (Peng, 2011).

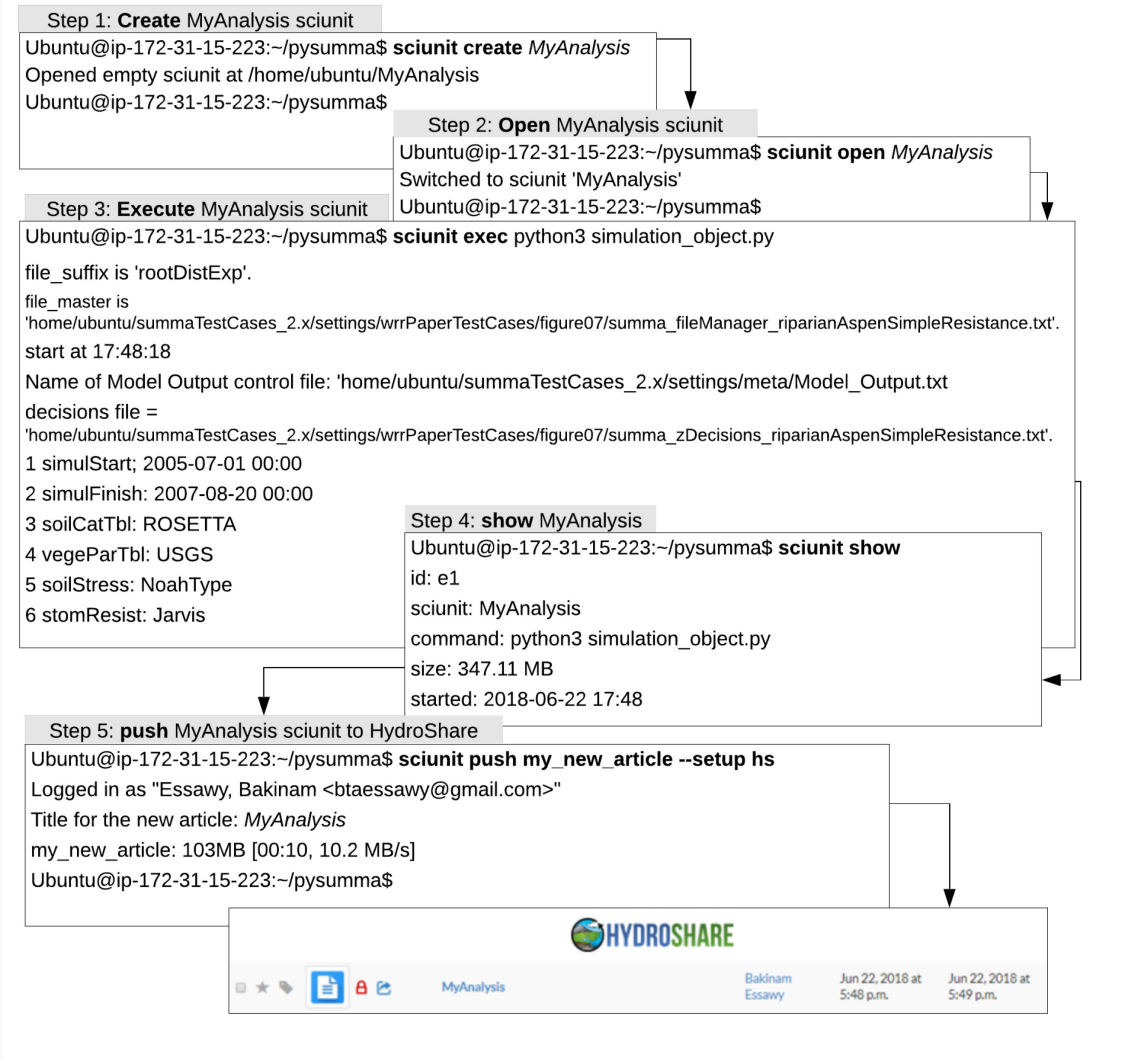

Figure 8. The steps followed to package the workflow analysis using the Sciunit tool.

The CUAHSI HydroShare JupyterHub app is a deployment of JupyterHub (Tarboton et al., 2018) hosted at http://jupyter.cuahsi.org that is part of the overall HydroShare cyberinfrastructure linked to, but separate from, the HydroShare repository (www.hydroshare.org). JupyterHub enables Literate programming, an approach that describes the written program so that the user can understand it (Piccolo and Frampton, 2016). Literate programing allows the researcher to split the code into fragments that can be executed

independently, and it enables the researcher to organize the fragments in a way that is easily understandable. Jupyter gives scientists the capability of combing data and visualizations in powerful ways when communicating their research (Pérez and Granger, 2007). Scientists use Jupyter notebooks as a supplement to the published manuscripts so that others can replicate their analysis and re-generate the published results. For example, Sadler et al., (2018) took a step toward creating reproducible research in the field of hydrology by publishing all codes and data on HydroShare. Additionally, Sadler et al., 2018 used the Jupyter functionality within HydroShare to create notebooks for some of the scripts used in his study, and he shared these notebooks with other researchers so they could interact with the notebooks using Jupyter.

Step 5 resulted in a HydroShare resource named MyAnalysis (Figure 9) (Choi, 2020). To run this in JupyterHub, the user would first click the "Open with" button from the resource's landing page in HydroShare and select CUAHSI JupyterHub. This step will open an instance of JupyterHub where the user can reproduce the workflow saved in the Sciunit package. Figure 10 shows these steps for reproducing the analysis with commands issued through the Jupyter user interface. The user must issue Sciunit commands by using the "!" expression in Jupyter. The sciunit open MyAnalysis command will open the Sciunit. The sciunit show command will show the latest packaged experiment within "MyAnalysis" Sciunit. While the sciunit list command will list the all the packaged experiments within "MyAnalysis" Sciunit. Finally, the sciunit repeat e1 command will rerun the experiment 1 (e1) analysis on the host machine. Examining the plot generated validates that results are consistent with the original work and demonstrates runnability.

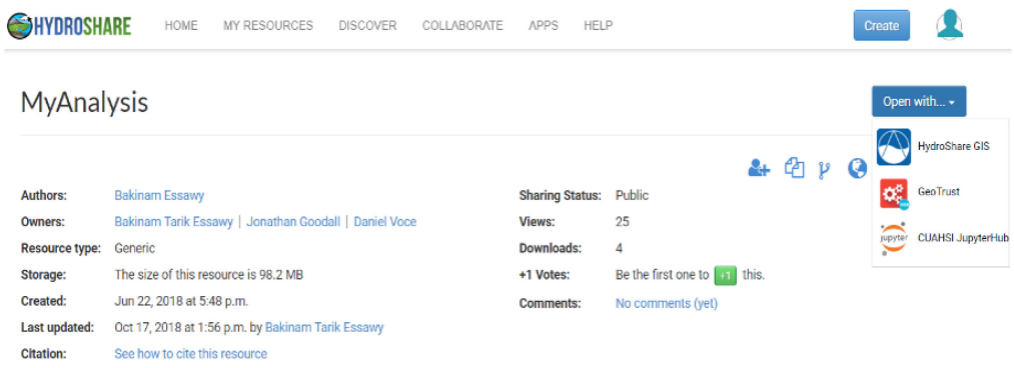

Figure 9. Screen shot of HydroShare Resource landing page holding the Sciunit package. CUAHSI JupyterHub appears under the “Open with” button.

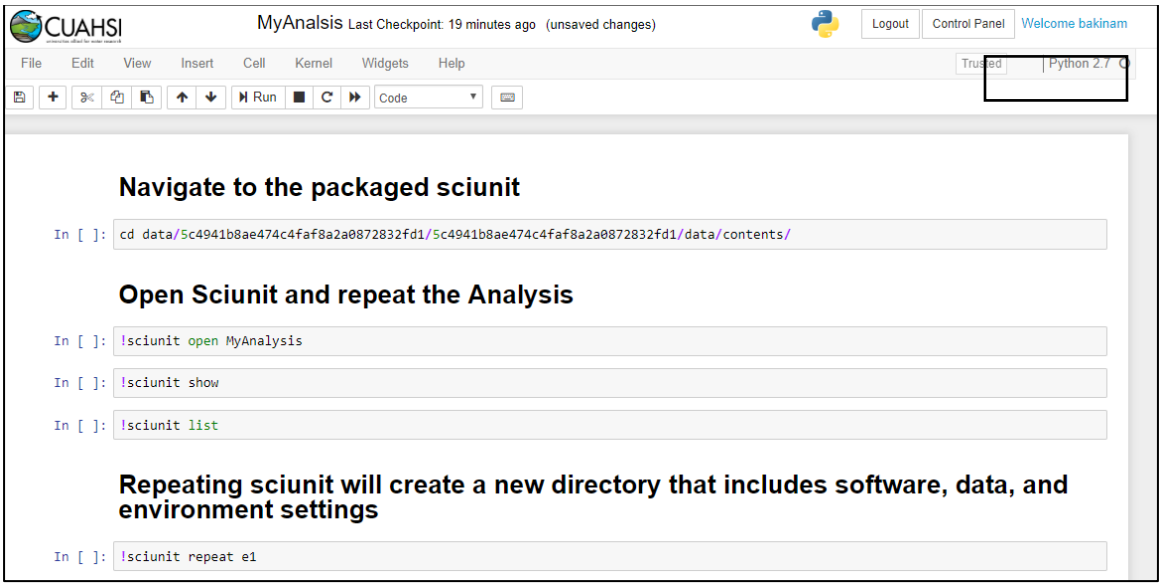

Figure 10. The steps taken on Jupyter notebook to repeat the analysis using the Sciunit tool commands.

3.4. Reproducing the SUMMA Analysis using Sciunit and HydroShare

Once runnability has been validated, the original researcher can make the HydroShare resource holding the Sciunit package public and accessible to other interested researchers. It can also be permanently published and be given a citable DOI. A new researcher can then access this resource to reproduce the analysis. One way the new researcher can do this is to also use the CUAHSI HydroShare JupyterHub app, essentially repeating the validation of runnability done by the original author above. This is, however, only a weak demonstration of reproducibility, because it uses the same environment (CUAHSI HydroShare JupyterHub) where runnability was tested.

The following steps describe how other researchers can reproduce the analysis on their own computer using the Sciunit package shared on HydroShare, a stronger test for reproducibility. These steps assume the new machine has Sciunit installed but makes no additional assumptions of software availability on the new machine. 1.) Download the Sciunit resource from HydroShare. 2.) Navigate to the location where the Sciunit resides. 3.) Use the command line to unpack and initiate the selected Sciunit using `sciunit open`. 4.) Use the command `sciunit list` to see what packages are within the Sciunit and which package will be used in the process (since some Sciunits may contain multiple packages, for example, Sciunit may use the same datasets but employ different methods). 5.) Repeat the "e1" package to generate the analysis. 6.) Once the analysis is repeated, a directory will be created with a copy of the analysis and all associated files. Examining the plot generated validates that results are consistent with the original work and demonstrates reproducibility.

3.5. Replicating the SUMMA Analysis

Once the user achieves reproducibility following the steps outlined in the prior section, the user is now in a position to replicate the work by modifying the Sciunit package and running it using a new dataset or code aimed at answering the same research question using the different data, or exploring whether a different model or approach to the analysis produces results that support the conclusions. A different implementation of the model, here may be new code, but the same underlying equations or principles, and this would serve to evaluate whether the code and solvers actually implement the equations properly. While replicability could be achieved without first reproducing another's work, we suggest that in many cases the path to replicability is best traveled by first being able to reproduce past work. This puts a researcher in a better position to interpret any differences that may arise and at the same time be more efficient and take advantage of and reuse code from the original researchers.

The steps for achieving replicability within CUAHSI HydroShare JupyterHub using Sciunit build off of the steps for reproducibility. Once the five steps needed for reproducing an analysis have been completed, the next step is to 1.) navigate to the directory that includes the analysis created and edit the script to use different data or a different approach (Figure 11). Next, 2.) the user would commit changes and repeat Sciunit execution (package) "e2" with the sciunit given command, as shown in Figure 12. After completing these steps, the user has replicated the study by creating a copy of a past analysis, changing that analysis's input data, and rerunning the analysis using the new input file. Because all software dependencies are handled by the Sciunit tool, the analysis will be reproducible on any machine with a compatible OS and one that has the Sciunit client software installed. This same process could be repeated outside of the CUAHSI HydroShare JupyterHub environment following a similar procedure to the one described in Section 3.4.

CUAHSI simulation_object.py 06/21/2018

File Edit View Language

```
from pysumma.Simulation import Simulation
import subprocess
import pylab
#import seaborn as sns
import xarray as xr
import matplotlib.pyplot as plt
from pysumma.Plotting import  Plotting
import netCDF4 as nc
import numpy as np
import glob
# create a pySUMMA simulation object using the SUMMA 'file manager' input file
S = Simulation('/home/ubuntu/figure07/summa_fileManager_riparianAspenSimpleResistance.txt')
# set the simulation start and finish times
S.decision_obj.simulStart.value = "2005-07-01 00:00"
S.decision_obj.simulFinsh.value = "2007-08-20 00:00"
S.executable = "/home/ubuntu/summa/bin/summa.exe"
S.decision_obj.stomResist.value = 'BallBerry'
results_simpleResistance, out_file = S.execute(run_suffix="rootDistExp", run_option = 'local')

# Visualization

P= Plotting(out_file)
P.plot_1d("pptrate")
#savefig("scalarCanopyTranspiration.pdf")
#pylab.savefig('test')
```

Figure 11. The modified software resulting from the replication analysis.

Commit the changes to create a new package

```
In [ ]: !sciunit commit
```

Repeat the newly created sciunit package using different dataset

```
In [ ]: !sciunit given ../../f4a84771e68c4abd90c135ce4f72cfbd/f4a84771e68c4abd90c135ce4f72cfbd/data/contents repeat e2
```

Figure 12. Commit changes to the software into the Sciunit and repeating the “e2” (experiment 2) run.

4. Discussion

4.1. Defining and Assuring Consistent Results

The National Academies report’s definition for reproducibility includes the phrase “consistent results,” acknowledging that achieving reproducibility does not require obtaining the

exact same results as the original study. Very often in computational modelling, numerical differences due to a variety of factors including rounding and precision mean that the output from an analysis does not have to be exactly the same as prior runs of the analysis, but it should be "close enough." Furthermore, some computational modelling strategies are stochastic so achieving the exact result from a prior run is impossible. Obviously, close enough is a relative measure that requires expert judgment based on the specific requirements of the study and the tolerance for differences in numerical results. For this reason, it is a more difficult standard in some ways that achieving the exact results from a prior analysis.

The "consistent results" requirement also points to the complicating factor that it is not necessary to have an exact recreation of the computational environment in order to achieve reproducibility. Thus, while we made the claim at the start of this paper that software dependency differences could be the reason for an inability to reproduce a prior computational study (see Figure 1), this may not always be true. For example, consider the case illustrated in Figure 13 which presents a more detailed view of case described in Figure 1. In this more detailed view, two researchers, A and B, have slight differences in their computational environments yet runnability can still be achieved if the results from Researcher B's setup (different libraries and operating system) can produce results that are close enough to Researcher A's results. There is risk here, however, because just because the results are close enough for one run (reproducibility), there is no guarantee that a slight change in the input data could still produce results that are close enough for some other run (replication). However, there is value in allowing for software inconsistencies because, if setup B is close to the setup A, setup B has migrated the model to later versions of dependency software. There clearly needs to be the ability to move to new software versions while also achieving reproducibility along the way. At the same time, it is well known that some software mitigations, like the migration from GDAL

v2 to GDAL v3 illustrated in Figure 13, can cause breaking changes, so documenting these dependency changes is important. To test for possible breaking changes, ideally the analysis is run across a set of different input data to increase confidence in reproducibility as new software updates are incorporated into the analysis.

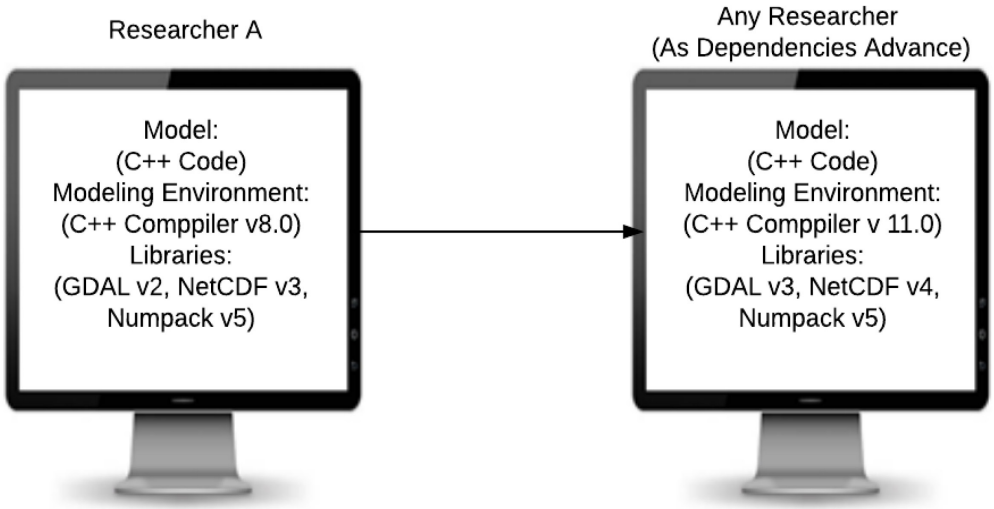

Figure 13. An example for how runnability can be achieved if the results from researcher B setup is different than the setup of researcher A

4.2. Limitations and Remaining Challenges

While computational resources (data, code, and environment) are important, these alone are insufficient to ensure reproducibility. The researcher must also include documentation and metadata about the software they use in order to run their programs correctly. Documentation in the journal article itself is often lacking and must be accompanied with user manuals or other resources, including Jupyter notebooks that more fully describe the computational analysis. Metadata is needed to uniquely identify computational resources within the growing ecosystem

of data and software. While metadata for data is well established, recent efforts have been made towards capturing software metadata. OntoSoft is an example that provides an ontology and portal for addressing the challenge of capturing metadata for scientific software in a formal way (Gil et al., 2016, 2015). In hydrology, the HydroShare system can be used to describe metadata for data and models. HydroShare defines two key computational modelling concepts: a model program and a model instance. The model program is the software for executing the model and the model instance is the input files required for executing the model (Horsburgh et al., 2015; Morsy et al., 2017, 2014; Tarboton et al., 2014).

This study focuses on studies where input data has already been prepared for an analysis and all required data for running the analysis are stored locally (i.e., there are no references to external data or services). Once this is the case, the remaining goal is to perform model simulations and analyze the model output as described in this paper. Following this work, a next step needed to achieve reproducibility in complex computational studies is to have workflows that automate the end-to-end process of reproducing the analysis from raw data to publication-ready figures and tables. The general workflow should automate the steps to: 1) obtain the raw data published in an online repository, preferably from a published, immutable external resource with a DOI for reproducibility, 2) run one or more scripts needed to prepare the raw data for the model, 3) run the model using the prepared datasets, and 4) post-process the model output to generate publication ready figures and tables. This four-step workflow should be containerized into a virtual environment that can be repeated by other researchers. Figure 14 shows the proposed approach to achieve reproducible research.

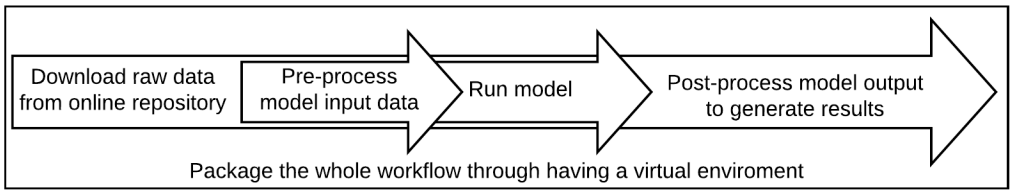

Figure 14. A proposed approach to achieve reproducible research.

Some efforts have already been made to create generic processes to automate the larger workflow required for hydrologic modelling. One example is work using the Variable Infiltration Capacity (VIC) hydrological model, a macro-scale hydrologic model that applies water and energy balances to simulate terrestrial hydrology at a regional level (Liang et al., 1996). The VIC model, like many hydrologic models, requires significant effort to prepare its input data. Billah et al. (2016) demonstrated a single-step process to create a workflow that automated the preparation of the input data for the VIC. While workflow software can help to better capture the provenance, it is still important to have sufficient metadata for each step within the workflow (Essawy et al., 2017). An example of workflows focused on metadata capture is work using MODFLOW-NWT as a demonstration model (Essawy et al., 2018). The MODFLOW-NWT is a version of the United States Geological Survey's groundwater model, MODFLOW (Niswonger et al., 2011). The work done for the MODFLOW-NWT, because it leverages the Sciunit concept provided through the GeoTrust project, improves reproducibility in the way it guarantees the replicability of research. Because this workflow is automated, the user is unable to change the parameters of the workflow execution, and all data run through this workflow will be executed according to the same parameters specified by the automated workflow. Work is needed to merge these ideas and concepts into a complete end-to-end workflow for repeatable, runnable, reproducible, and replicable environment modelling that stretches from data preparation, to model execution, to post-processing of model outputs, all containerized, well documented with

appropriate metadata, easy to understand and reuse, and discoverable through online data repository systems.

5. Conclusions

This research has shown a path to achieving reproducibility and replicability of environmental modelling studies by first providing a more formal taxonomy that defines studies as being repeatable, runnable, reproducible, or replicable. After defining this taxonomy, we focused on a methodology that uses this taxonomy in describe steps needed to move research along the spectrum from being repeatable to being replicable. Using a hydrologic modelling analysis as an example, we demonstrate these steps and highlight the important role of containers and software tools that enable scientists to use containers in achieving reproducible and replicable studies.

The hydrologic modelling analysis demonstrating this process uses the SUMMA modelling framework and different parameterizations of stomatal resistance to estimate total evapotranspiration for a study site. We demonstrated how to achieve the steps along the taxonomy with existing cyberinfrastructure software tools as follows. 1.) ***Repeating*** an analysis on the same computer with the same inputs as its original application. 2.) ***Running*** the same analysis on another machine and getting consistent results by packaging it as a container using the Sciunit software. 3.) ***Reproducing*** the analysis by sharing it to HydroShare and allowing other researchers to re-execute it through the literate programing JupyterHub. 4.) ***Replicating*** the analysis by allowing researchers to easily change the shared Sciunit package to use their own data or modelling ideas and then repeating the 1-4 sequence of steps.

While this work moves closer to achieving reproducible and replicable environmental modelling studies, there are still remaining limitations and challenges to be addressed. First, it is

difficult to know precisely when an analysis has been reproduced because achieving this goal does not require an exact match of the computational environment but rather requires a similar computational environment. That is, perhaps some dependences have been updated to new versions, but it is essentially the same computational environment and set up. Therefore, the results may not be exactly the same but are consistent with results in the prior analysis run. Creating automated tools for judging this potentially vague criteria of "consistent results" for reproducibility remains a challenge. Literate programming tools like Jupyter are a major step forward in providing documentation alongside analysis, but more work is needed to better document computational analyses in ways that foster reproducibility and replicability. This is especially true for efforts to document and automate end-to-end workflows that operate from raw input data as inputs, create model input files, execute models, and produce publication-ready publications. Capturing these entire end-to-end workflows in a way that can be not only reproduced and replicated by others, but also easy to understand and reuse, remains an open challenge.

6. Acknowledgements

We gratefully acknowledge the National Science Foundation for support of this work under awards ICER-1639759, ICER-1661918, ICER-1540901, and OAC-1664061.

7. References

Baker, M., 2016. Muddled meanings hamper efforts to fix reproducibility crisis. Nature. https://doi.org/10.1038/nature.2016.20076

Baker, M., Penny, D., 2016. Is there a reproducibility crisis? Nature. https://doi.org/10.1038/533452A

Bandaragoda, C., Castronova, A., Istanbulluoglu, E., Strauch, R., Nudurupati, S.S., Phuong, J., Adams, J.M., Gasparini, N.M., Barnhart, K., Hutton, E.W.H.H., Hobley, D.E.J.J., Lyons, N.J., Tucker, G.E., Tarboton, D.G., Idaszak, R., Wang, S., 2019. Enabling Collaborative Numerical Modeling in Earth Sciences using Knowledge Infrastructure. Environ. Model. Softw. 120, 104424. https://doi.org/https://doi.org/10.1016/j.envsoft.2019.03.020

Bell, A.W., Deutsch, E.W., Au, C.E., Kearney, R.E., Beavis, R., Sechi, S., Nilsson, T., Bergeron, J.J.M., Beardslee, T.A., Chappell, T., Meredith, G., Sheffield, P., Gray, P., Hajivandi, M., Pope, M., Predki, P., Kullolli, M., Hincapie, M., Hancock, W.S., Jia, W., Song, L., Li, L., Wei, J., Yang, B., Wang, J., Ying, W., Zhang, Y., Cai, Y., Qian, X., He, F., Meyer, H.E., Stephan, C., Eisenacher, M., Marcus, K., Langenfeld, E., May, C., Carr, S.A., Ahmad, R., Zhu, W., Smith, J.W., Hanash, S.M., Struthers, J.J., Wang, H., Zhang, Q., An, Y., Goldman, R., Carlsohn, E., van der Post, S., Hung, K.E., Sarracino, D.A., Parker, K., Krastins, B., Kucherlapati, R., Bourassa, S., Poirier, G.G., Kapp, E., Patsiouras, H., Moritz, R., Simpson, R., Houle, B., LaBoissiere, S., Metalnikov, P., Nguyen, V., Pawson, T., Wong, C.C.L., Cociorva, D., Yates, J.R., Ellison, M.J., Lopez-Campistrous, A., Semchuk, P., Wang, Y., Ping, P., Elia, G., Dunn, M.J., Wynne, K., Walker, A.K., Strahler, J.R., Andrews, P.C., Hood, B.L., Bigbee, W.L., Conrads, T.P., Smith, D., Borchers, C.H., Lajoie, G.A., Bendall, S.C., Speicher, K.D., Speicher, D.W., Fujimoto, M., Nakamura, K., Paik, Y.K., Cho, S.Y., Kwon, M.S., Lee, H.J., Jeong, S.K., Chung, A.S., Miller, C.A., Grimm, R., Williams, K., Dorschel, C., Falkner, J.A., Martens, L., Vizcaíno, J.A., 2009. A HUPO test sample study reveals common problems in mass spectrometry-based proteomics. Nat. Methods 6, 423–430. https://doi.org/10.1038/nmeth.1333

Billah, M.M., Goodall, J.L., Narayan, U., Essawy, B.T., Lakshmi, V., Rajasekar, A., Moore, R.W., 2016. Using a data grid to automate data preparation pipelines required for regional-

scale hydrologic modeling. Environ. Model. Softw. 78, 31–39. https://doi.org/10.1016/j.envsoft.2015.12.010

Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M.B., Kowalik, K., Kulasekaran, S., Ludäscher, B., Mecum, B.D., Nabrzyski, J., Stodden, V., Taylor, I.J., Turk, M.J., Turner, K., 2019. Computing environments for reproducibility: Capturing the "Whole Tale." Futur. Gener. Comput. Syst. 94, 854–867. https://doi.org/10.1016/j.future.2017.12.029

Choi, Y.-D., 2020. MyAnalysis | CUAHSI HydroShare [WWW Document]. HydroShare. URL https://www.hydroshare.org/resource/7d1403636fd3444c87e3c5b40b000b91/ (accessed 4.28.20).

Chuah, J., Deeds, M., Malik, T., 2020. Documenting Computing Environments for Reproducible Experiments, in: Ian Foster, Gerhard R. Joubert, Luděk Kučera, Wolfgang E. Nagel, F.P. (Ed.), Volume 36: Parallel Computing: Technology Trends. IOS Press, pp. 756–765. https://doi.org/10.3233/APC200106

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

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

Easterbrook, S.M., 2014. Open code for open science? Nat. Geosci. https://doi.org/10.1038/ngeo2283

Essawy, B.T., Goodall, J.L., Xu, H., Gil, Y., 2017. Evaluation of the OntoSoft Ontology for describing metadata for legacy hydrologic modeling software. Environ. Model. Softw. 92, 317–329. https://doi.org/10.1016/j.envsoft.2017.01.024

Essawy, B.T., Goodall, J.L., Zell, W., Voce, D., Morsy, M.M., Sadler, J., Yuan, Z., Malik, T., 2018. Integrating scientific cyberinfrastructures to improve reproducibility in computational hydrology: Example for HydroShare and GeoTrust. Environ. Model. Softw. 105, 217–229. https://doi.org/10.1016/j.envsoft.2018.03.025

Garijo, D., Kinnings, S., Xie, Li, Xie, Lei, Zhang, Y., Bourne, P.E., Gil, Y., 2013. Quantifying reproducibility in computational biology: The case of the tuberculosis drugome. PLoS One 8. https://doi.org/10.1371/journal.pone.0080278

Gil, Y., David, C.H., Demir, I., Essawy, B.T., Fulweiler, R.W., Goodall, J.L., Karlstrom, L., Lee, H., Mills, H.J., Oh, J.H., Pierce, S.A., Pope, A., Tzeng, M.W., Villamizar, S.R., Yu, X., 2016. Toward the Geoscience Paper of the Future: Best practices for documenting and sharing research from data to software to provenance. Earth Sp. Sci. https://doi.org/10.1002/2015EA000136

Gil, Y., Ratnakar, V., Garijo, D., 2015. OntoSoft: Capturing scientific software metadata, in: Proceedings of the 8th International Conference on Knowledge Capture, K-CAP 2015. https://doi.org/10.1145/2815833.2816955

Goodman, S.N., Fanelli, D., Ioannidis, J.P.A., 2018. What does research reproducibility mean?, in: Getting to Good: Research Integrity in the Biomedical Sciences. pp. 96–102. https://doi.org/10.1126/scitranslmed.aaf5027

Gorgolewski, K.J., Poldrack, R.A., 2016. A Practical Guide for Improving Transparency and

Reproducibility in Neuroimaging Research. PLoS Biol. 14, 1–13. https://doi.org/10.1371/journal.pbio.1002506

Handigol, N., Heller, B., Jeyakumar, V., Lantz, B., McKeown, N., 2012. Reproducible Network Experiments Using Container-based Emulation, in: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '12. ACM, New York, NY, USA, pp. 253–264. https://doi.org/10.1145/2413176.2413206

Horsburgh, J.S., Morsy, M.M., Castronova, A.M., Goodall, J.L., Gan, T., Yi, H., Stealey, M.J., Tarboton, D.G., 2015. Hydroshare: Sharing Diverse Environmental Data Types and Models as Social Objects with Application to the Hydrology Domain. JAWRA J. Am. Water Resour. Assoc. 52, 873–889. https://doi.org/10.1111/1752-1688.12363

Hothorn, T., Leisch, F., 2011. Case studies in reproducibility. Brief. Bioinform. 12, 288–300. https://doi.org/10.1093/bib/bbq084

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 Resour. Res. 52, 7548–7555. https://doi.org/10.1002/2016WR019285

Ioannidis, J.P.A., Allison, D.B., Ball, C.A., Coulibaly, I., Cui, X., Culhane, A.C., Falchi, M., Furlanello, C., Game, L., Jurman, G., Mangion, J., Mehta, T., Nitzberg, M., Page, G.P., Petretto, E., Van Noort, V., 2009. Repeatability of published microarray gene expression analyses. Nat. Genet. 41, 149–155. https://doi.org/10.1038/ng.295

Ivie, P., Thain, D., 2018. Reproducibility in scientific computing. ACM Comput. Surv. https://doi.org/10.1145/3186266

Kjeldgaard, L., 2020. smaakage85/dockr: create lightweight docker image for an R package [WWW Document]. URL https://github.com/smaakage85/dockr (accessed 4.28.20).

Knoth, C., Nüst, D., 2017. Reproducibility and practical adoption of GEOBIA with open-source

software in Docker containers. Remote Sens. 9, 290. https://doi.org/10.3390/rs9030290

Kurtzer, G.M., Sochat, V., Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute. PLoS One 12, e0177459. https://doi.org/10.1371/journal.pone.0177459

Liang, X., Lettenmaier, D.P., Wood, E.F., 1996. One-dimensional statistical dynamic representation of subgrid spatial variability of precipitation in the two-layer variable infiltration capacity model. J. Geophys. Res. 101, 21403–21422. https://doi.org/10.1029/96JD01448

Marwick, B., Boettiger, C., Mullen, L., 2018. Packaging Data Analytical Work Reproducibly Using R (and Friends). Am. Stat. 72, 80–88. https://doi.org/10.1080/00031305.2017.1375986

Meng, H., Kommineni, R., Pham, Q., Gardner, R., Malik, T., Thain, D., 2015. An invariant framework for conducting reproducible computational science. J. Comput. Sci. 9, 137–142. https://doi.org/10.1016/j.jocs.2015.04.012

Merkel, D., 2014. Docker: lightweight Linux containers for consistent development and deployment. Linux J. 2014, 2. https://doi.org/10.1097/01.NND.0000320699.47006.a3

Morsy, M.M., Goodall, J.L., Bandaragoda, C., Castronova, A.M., Greenberg, J., 2014. Metadata for describing water models, in: Proceedings - 7th International Congress on Environmental Modelling and Software: Bold Visions for Environmental Modeling, IEMSs 2014. pp. 53–59. https://doi.org/10.13140/2.1.1314.6561

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

National Academies of Sciences Engineering, Medicine, 2019. Reproducibility and Replicability

in Science. The National Academies Press, Washington, DC. https://doi.org/10.17226/25303

Nekrutenko, A., Taylor, J., 2012. Next-generation sequencing data interpretation: Enhancing reproducibility and accessibility. Nat. Rev. Genet. https://doi.org/10.1038/nrg3305

Niswonger, R.G., Panday, S., Motomu, I., 2011. MODFLOW-NWT , A Newton Formulation for MODFLOW-2005, USGS reports.

Nüst, D., Hinz, M., 2019. containerit: Generating Dockerfiles for reproducible research with R. J. Open Source Softw. 4, 1603. https://doi.org/10.21105/joss.01603

Nüst, D., Konkol, M., Schutzeichel, M., Pebesma, E., Kray, C., Przibytzin, H., Lorenz, J., 2017. Opening the publication process with executable research compendia. D-Lib Mag. 23. https://doi.org/10.1045/january2017-nuest

Peng, R.D., 2011. Reproducible research in computational science. Science (80-.). https://doi.org/10.1126/science.1213847

Pérez, F., Granger, B.E., 2007. IPython: A system for interactive scientific computing. Comput. Sci. Eng. 9, 21–29. https://doi.org/10.1109/MCSE.2007.53

Piccolo, S.R., Frampton, M.B., 2016. Tools and techniques for computational reproducibility. Gigascience. https://doi.org/10.1186/s13742-016-0135-4

Rosenberg, D.E., Filion, Y., Teasley, R., Sandoval-Solis, S., Hecht, J.S., van Zyl, J.E., McMahon, G.F., Horsburgh, J.S., Kasprzyk, J.R., Tarboton, D.G., 2020. The Next Frontier: Making Research More Reproducible. J. Water Resour. Plan. Manag. 146, 01820002. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001215

Sadler, J.M., Goodall, J.L., Morsy, M.M., Spencer, K., 2018. Modeling urban coastal flood severity from crowd-sourced flood reports using Poisson regression and Random Forest. J. Hydrol. 559, 43–55. https://doi.org/10.1016/J.JHYDROL.2018.01.044

Stagge, J.H., Rosenberg, D.E., Abdallah, A.M., Akbar, H., Attallah, N.A., James, R., 2019. Assessing data availability and research reproducibility in hydrology and water resources. Sci. Data 6. https://doi.org/10.1038/sdata.2019.30

Stagge, J.H.H., Rosenberg, D.E.E., Abdallah, A.M.M., Akbar, H., Attallah, N.A.A., James, R., 2019. Author Correction: Assessing data availability and research reproducibility in hydrology and water resources. Sci. Data 6, 35. https://doi.org/10.1038/s41597-019-0039-0

Stodden, V., Bailey, D.H., Borwein, J., Leveque, R.J., Rider, W., Stein, W., 2013. Setting the Default to Reproducible Reproducibility in Computational and Experimental Mathematics, in: ICERM Workshop. p. 19.

Stodden, V., Miguez, S., Seiler, J., 2015. ResearchCompendia.org: Cyberinfrastructure for reproducibility and collaboration in computational science. Comput. Sci. Eng. 17, 12–19. https://doi.org/10.1109/MCSE.2015.18

Tarboton, D.G., Idaszak, R., 2015. HydroShare: Advancing Hydrology through Collaborative Data and Model Sharing. iRODS User Gr. Meet.

Tarboton, D.G., Idaszak, R., Horsburgh, J.S., Heard, J., Ames, D., Goodall, J.L., Band, L.L.E., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D.R., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D.R., Goodall, J.L., Band, L.L.E., Merwade, V., Couch, A., Arrigo, J., Hooper, R., Valentine, D., Maidment, D.R., 2014. Hydro share: Advancing collaboration through hydrologic data and model sharing. Proc. - 7th Int. Congr. Environ. Model. Softw. Bold Visions Environ. Model. iEMSs 2014 1, 23–29. https://doi.org/10.13140/2.1.4431.6801

Tatman, R., Vanderplas, J., Dane, S., 2018. A Practical Taxonomy of Reproducibility for Machine Learning Research. Reprod. ML Work. ICML'18.

That, D.H.T., Fils, G., Yuan, Z., Malik, T., 2017. Sciunits: Reusable research objects, in:

Proceedings - 13th IEEE International Conference on EScience, EScience 2017. Institute of Electrical and Electronics Engineers Inc., pp. 374–383. https://doi.org/10.1109/eScience.2017.51

Woodson, C., Hayes, J.H., Griffioen, S., 2018. Towards reproducible research: Automatic classification of empirical requirements engineering papers, in: Proceedings of the ACMSE 2018 Conference. p. 8. https://doi.org/10.1145/3190645.3190689

Yuan, Z., Ton That, D., Kothari, S., Fils, G., Malik, T., 2018. Utilizing Provenance in Reusable Research Objects. Informatics 5, 14. https://doi.org/10.3390/informatics5010014