Check for updates

METHOD ARTICLE

REVISED Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case

[version 3; peer review: 2 approved, 1 not approved]

Markus Konkol[1], Astra Labuce[2], Sami Domisch[3], Merret Buurman[3], Vanessa Bremerich[3]

[1]52°North Spatial Information Research, Münster, 48155, Germany
[2]Institute of Aquatic Ecology, Agency of Daugavpils University, Riga, LV-1007, Latvia
[3]Forschungsverbund Berlin e.V., Leibniz Institute of Freshwater Ecology and Inland Fisheries (IGB), Berlin, 12587, Germany

v3 **First published:** 06 May 2025, **5**:123
https://doi.org/10.12688/openreseurope.20221.1
Second version: 04 Jul 2025, **5**:123
https://doi.org/10.12688/openreseurope.20221.2
Latest published: 22 Sep 2025, **5**:123
https://doi.org/10.12688/openreseurope.20221.3

Abstract

The growing demand for reproducible research is based on the expectation that publishing research in this form will enable its reuse and the generation of new knowledge. However, reproducibility alone does not guarantee these benefits. Users still need to make considerable efforts to understand the data and analysis code before they can reuse these components in other contexts. To address this challenge, we introduce the Data-to-Knowledge Package (D2K-Package), a collection of research materials including source code and open FAIR data, virtual labs, web API services, and computational workflows. The D2K-Package's core is the reproducible basis composed of the data and source code on which an analysis is based. This core is designed such that the other components can be derived from it. The main goal of the package is to help researchers generate new knowledge by facilitating the understanding and encouraging the reuse of reproducible research. We demonstrate the applicability of the D2K-Package with a hydrological use case which can be also used for testing, and discuss its seamless integration into the research cycle.

Plain language summary

Researchers often collect data and analyse it using appropriate methods in order to answer a specific research question. The research results are then published in scientific articles. If other researchers have access to the data and the analysis used and can achieve the

Open Peer Review

Approval Status ✓ ✓ ✗

	1	2	3
version 3 (revision) 22 Sep 2025			
version 2 (revision) 04 Jul 2025	✓ view	✓ view	✗ view
version 1 06 May 2025	? view	? view	

1. **Miguel Colom**, ENS Paris-Saclay, Université de Paris, Gif-sur-Yvette, France
2. **Victoria Lush**, Aston University, Birmingham, UK
3. **David Tarboton**, Utah State University, Logan, USA
David Rosenberg, Utah State University, logan, USA

Any reports and responses or comments on the article can be found at the end of the article.

same results as in the article, this is called “reproducible research”. One of the main advantages of reproducible research is that other scientists can continue the work and reuse the materials. However, publishing the materials in a reproducible manner is a challenge for the developer of the analysis. In addition, reusing the material for others is a difficult task as the analysis can become very complex.

In this paper, we introduce the Data-to-Knowledge package (D2K-Package), which contains the data and analysis in a reproducible manner, as well as other components that help other researchers understand and reuse the analysis. One of these components is a workflow that visualizes the steps of the analysis pipeline as a first entry point. To demonstrate the feasibility of the D2K-Package, we applied the idea to a real use case in hydrology.

Keywords
FAIR Data, Reproducible Research, Workflows, OGC API, Galaxy, Hydrology, Gulf of Riga, Trend analysis

This article is included in the Earth and Environmental Sciences gateway.

This article is included in the Horizon Europe gateway.

Corresponding author: Markus Konkol (m.konkol@52north.org)

Author roles: Konkol M: Conceptualization, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Labuce A**: Data Curation, Formal Analysis, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation; **Domisch S**: Writing – Original Draft Preparation, Writing – Review & Editing; **Buurman M**: Formal Analysis, Resources, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Bremerich V**: Conceptualization

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Commission’s Horizon Europe Research and Innovation programme. Grant agreement No 101094434.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2025 Konkol M *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Konkol M, Labuce A, Domisch S *et al.* **Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case [version 3; peer review: 2 approved, 1 not approved]** Open Research Europe 2025, **5**:123 https://doi.org/10.12688/openreseurope.20221.3

First published: 06 May 2025, **5**:123 https://doi.org/10.12688/openreseurope.20221.1

REVISED **Amendments from Version 2**

- Figure 7 was added to show the final output of the workflow

- A paragraph was added (see end of section "A hydrological use case") to answer the research question

- The documentations in the D2K-Package and the toolbox were updated to facilitate reproducibility

- minor changes in the introduction

Any further responses from the reviewers can be found at the end of the article

Introduction

The scientific community must adapt to an environment in which research budgets are shrinking in many parts of the world, even though scientific advances are urgently needed to tackle global challenges such as climate change, biodiversity loss, and pandemics[1,2]. It is equally important today that scientific findings can be trusted and are not easily called into question. Sharing and publishing reusable research outputs is a crucial strategy to mitigate these challenges and foster new collaborative developments[3]. A key concept is computational reproducibility, which ensures that other researchers can validate and verify the results reported in a scientific article using the same data and code[4,5]. Consequently, reproducibility is a fundamental quality criterion in science, and one might assume it is widely practiced to strengthen the credibility and transparency of research. However, several studies have shown that many research articles across disciplines are in fact not reproducible[6–8] - a phenomenon often referred to as the "reproducibility crisis"[9]. There are various reasons for this, but a major obstacle is that publishing reproducible research results is time-consuming and some might argue it is not worth the investment[10]. Nevertheless, reproducible research should be valued not only for its intrinsic scientific merit, including verification and validation, but also for its wider benefits. One of these advantages is that readers do not have to blindly trust the results in an article, but can check them themselves on the basis of the accompanying data and analysis. A key expectation is that reproducible research results are inherently more reusable by other researchers, reducing redundant efforts and accelerating the creation of new knowledge[11–13]. Nevertheless, someone trying to reuse someone else's code has to invest a lot of time to understand the code and incorporate parts of it into their own analysis.

In response, funding agencies, peer reviewers, and academic journals are increasingly emphasizing the need for researchers to publish all materials required for reproducibility[14–16]. While this emphasis represents a step forward, it also entails changes in researchers' daily workflows, and this transition is still ongoing. Numerous tools, guidelines, and best practices have been developed to support this trend, including the FAIR (Findability, Accessibility, Interoperability, and Reusability) principles for data[17] and software[18] and the "Ten Simple Rules for Computational Reproducible Research"[19]. Gentleman and Lang[20] introduced the concept of a compendium as a package including a dynamic document (i.e., the paper) and the code and data needed to produce the computational results in that document. ReproZip[21] is an example implementation of the compendium concept and packages the computational environment with the source code and data so that users can start a virtual machine to re-run the analysis. The commercial publisher CodeOcean creates such virtual environments as supplements for research articles[22]. In addition, eLife is a scientific publisher that offers interactive papers based on a reproducible analysis[23].

Encapsulating all materials in a compendium might limit the findability of the individual components contained therein, unless the metadata of the individual elements can be found externally. Moreover, researchers may want or need to publish individual research assets in dedicated repositories. As an alternative to a compendium, a so-called Research Object can be created, which aggregates digital assets that are part of an analysis and are published in different repositories or archives[24,25]. Together with the Research Object, a particularly influential concept in the context of this paper is the Knowledge Package developed by the Group on Earth Observation (GEO), a "reusable application-sharing unit" that integrates geospatial datasets, code, tools, and the computational environment[26].

However, reusing research code requires a deep understanding of the underlying logic of the analysis pipeline, e.g., how the functions work, the meaning of input parameters, and how to adapt them to other contexts. Understanding complex source code requires knowledge of the programming language and can be a challenge even for experienced programmers[27]. Consequently, researchers often tend to reinvent the wheel and develop analyses from scratch, or rely on proprietary, point-and-click software with a graphical user interface (GUI) that lacks reproducibility[28]. This challenge is further complicated by the diverse needs and skills of different stakeholders, including reviewers, researchers, and science journalists. Workflow management systems can help with this problem by providing an entry point into the analysis pipeline. Examples include Nextflow[29], a command line-based tool that allows users to script workflows, and Snakemake[30], which defines workflow steps as rules. The Galaxy platform[31] also addresses users without programming expertise by offering an intuitive user interface for the development and execution of readily-sharable workflows.

In this paper, we introduce the Data-to-Knowledge Package (D2K-Package), a collection of links to digital research assets - such as data, code, virtual labs, web API services and workflows - that unlock the potential of open reproducible research and open FAIR data. Conceptually, the D2K-Package is built on top of the Research Object and GEO's Knowledge Package but focuses on creating a reproducible basis out of the code and data so that the other research assets (i.e., virtual labs, web API services, and workflows) can be inferred from it. A special feature of the code component is its structuring into self-contained and reusable functions that fulfil a specific task (e.g. data transformation) instead of enabling access to the analysis script as a whole. The primary aim of the D2K-Package

is to enhance the reusability of computational research outputs by enabling audiences with varying skill levels to follow the process from data analysis to knowledge generation.

In the following sections, we first outline the concept of the D2K-Package and show how researchers can develop a D2K-Package based on their reproducible analysis. Afterwards, we showcase its applicability using a hydrological use case, which we provide as a demonstrator so that the D2K-Package concept can be tested and explored. The paper concludes with a discussion on the integration of the D2K-Package into the research cycle, its benefits and limitations.

Methods

The concept of the Data-to-Knowledge Package

A D2K-Package (Figure 1) links to specific research assets to tackle key technical challenges encountered in a researcher's work:

- Verifying and reproducing the results presented e.g. in a publication needs access to the data and source code. The D2K-Package supports these tasks by linking to the ***Data*** and the ***Toolbox*** (Figure 1), which together form the reproducible basis. In a Toolbox, the code is structured in such a way that the entire analysis is divided into self-contained and reusable functions that can be chained together to reproduce the analysis or called individually in a different context. More details on the creation of a Toolbox are provided below.

- Exploring the analysis and evaluating its relevance and reusability in other use cases requires restoring the computational environment. This involves tasks like installing a specific R[32] version with required packages or Docker to build and run images. However, this process is time-consuming and may be challenging or even unfeasible for users lacking the necessary technical skills, which may discourage them from attempting to run the analysis. The D2K-Package supports these efforts by linking to a ***Virtual Lab*** allowing users to engage more deeply with the code.

- Reusing parts of the analysis in a custom script requires copying the relevant code snippets and the necessary libraries, which can be an error-prone task. The D2K-Package overcomes this issue by linking to a ***Web API Service*** to which calls to the functions contained in the Toolbox can be sent from a source code script.

- Understanding the role of each function in an analysis, which input parameters are needed, and how the functions are connected requires familiarity with the programming language and effort. This is particularly challenging if the analysis script contains several thousand lines of code. To provide an entry point to the analysis and help other researchers (with and without coding skills) receive an overview of the analysis workflow composed of the functions and the configuration, the D2K-Package links to an executable and readily-sharable ***Computational Workflow***.

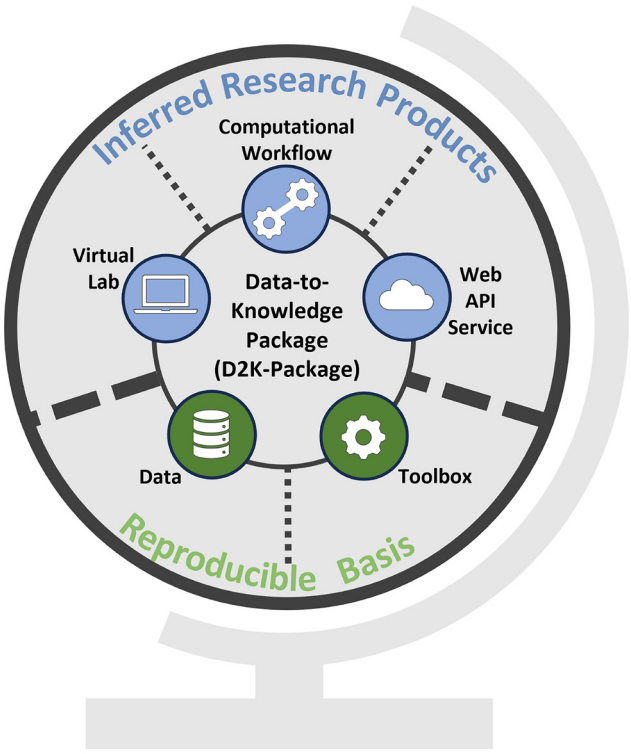
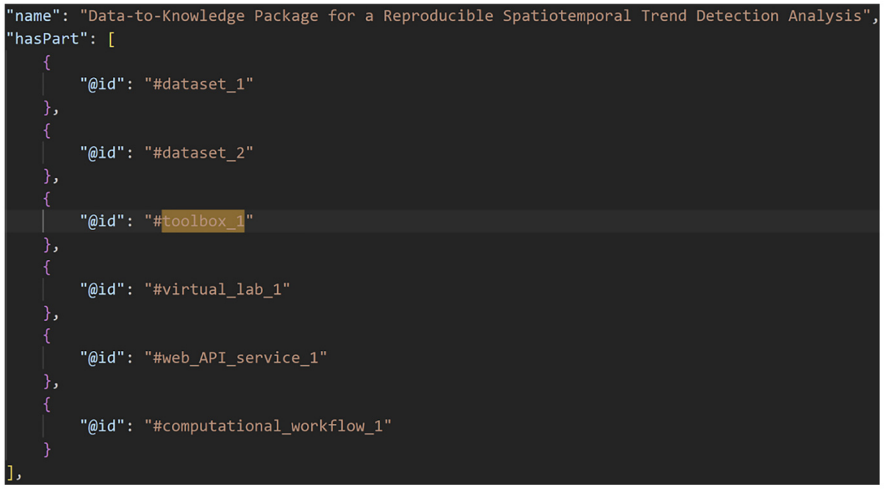


Figure 1. Conceptual overview of a Data-to-Knowledge-Package. Bottom: Reproducible basis composed of the *Data* & *Toolbox*. Top: Inferred research products built on top of the reproducible basis. All these components are linked in a Data-to-Knowledge Package.

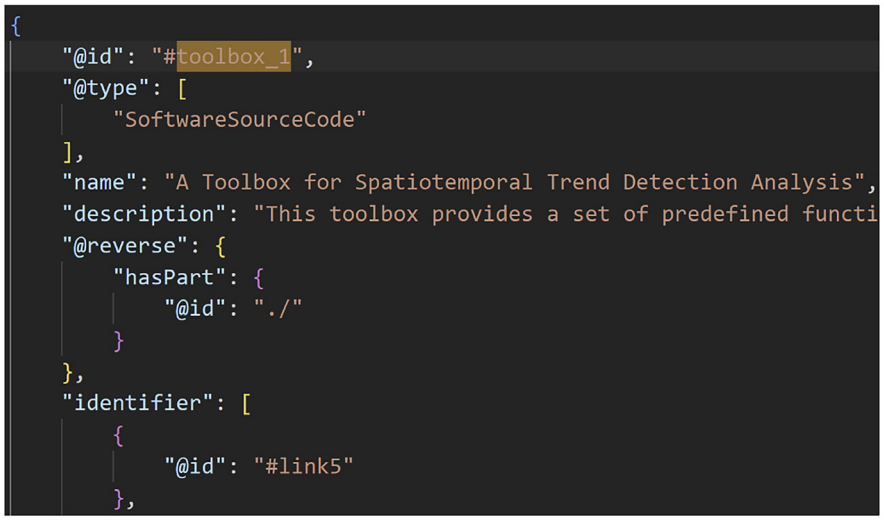
Consequently, a D2K-Package aims to facilitate various forms of reuse, including inspecting the analysis pipeline and running the analysis as-is. In addition, a D2K-Package places a particular emphasis on enabling reuse in new contexts by allowing users to customize the configuration of the analysis and extend it.

The main feature of the D2K-Package is that it acts as an overarching meta-object including links to these components but it does not create another physical copy of them in a self-contained compendium as proposed in related approaches[20,33]. We chose this approach to avoid redundancy (e.g., copies of a given dataset) and possible synchronization issues caused by divergent code developments. Hence, several D2K-Packages can link to the same specific research asset (e.g., a dataset). Figure 2–Figure 4 show excerpts from a reference implementation of a D2K-Package in the form of a ro-crate metadata file[25]. The file is a human- and machine-readable way of describing research assets of different types based on the structured data format JSON-LD and vocabularies, e.g. schema.org. The D2K-Package components *Data*, *Toolbox*, *Virtual Lab*, *Web API Service* and *Computational Workflow* are listed under the metadata element "*hasPart*". The "*@id*" is a reference to the corresponding json object that contains the name, resource type and identifier (i.e. URL, DOI) of the D2K-Package component. The resulting ro-crate metadata file aggregates all components



```
"name": "Data-to-Knowledge Package for a Reproducible Spatiotemporal Trend Detection Analysis",
"hasPart": [
    {
        "@id": "#dataset_1"
    },
    {
        "@id": "#dataset_2"
    },
    {
        "@id": "#toolbox_1"
    },
    {
        "@id": "#virtual_lab_1"
    },
    {
        "@id": "#web_API_service_1"
    },
    {
        "@id": "#computational_workflow_1"
    }
],
```


Figure 2. A Data-to-Knowledge Package's ro-crate metadata file. See full file here. Extract of a ro-crate metadata file (part 1) showing the components ("hasPart") of a Data-to-Knowledge-Package.



```
{
    "@id": "#toolbox_1",
    "@type": [
        "SoftwareSourceCode"
    ],
    "name": "A Toolbox for Spatiotemporal Trend Detection Analysis",
    "description": "This toolbox provides a set of predefined functi
    "@reverse": {
        "hasPart": {
            "@id": "./"
        }
    },
    "identifier": [
        {
            "@id": "#link5"
        },
```


Figure 3. A Data-to-Knowledge Package's ro-crate metadata file. See full file here. Extract of a ro-crate metadata file (part 2) showing the Toolbox of a D2K-Package.



```
{
    "@id": "#link5",
    "@type": [
        "URL"
    ],
    "name": "https://doi.org/10.5281/zenodo.15112709",
    "@reverse": {
        "identifier": [
            {
                "@id": "#toolbox_1"
            },
```


Figure 4. A Data-to-Knowledge Package's ro-crate metadata file. See full file here. Extract of a ro-crate metadata file (part 3) showing the link to the Toolbox component.

needed to reproduce the analysis pipeline by their DOIs or URLs and is stored in a single public repository. We will see later in the section "User Interface" how this ro-crate file can be used as a central entry point for users to access the analysis pipeline.

Ideally, the components are published as citable assets in repositories using permanent identifiers (e.g. a Digital Object Identifier, DOI) to which the D2K-Package can link[34].

A D2K-Package can evolve and be expanded to include new materials in order to take account of the dynamic nature of publishing research assets. For example, a scientific paper under review can be supplemented by a D2K-Package and become part of the D2K-Package after publication. For this reason, the D2K-Package ro-crate file shown below should be published with, ideally versioned, DOIs. A D2K-Package can also refer to other research materials that facilitate the reusability of computational research. A poster, talk or similar can be a useful means of providing a detailed explanation of the context. Also, open educational resources (e.g. links to webinars, slides, tutorials) can contribute to a better understanding of the analysis and consequently facilitate reuse. Nevertheless, such creative outputs are not in the scope of this paper.

The following sections will elaborate on how the components can be created to achieve a D2K-Package that supports the reuse of computational research and the understanding of the path from data to knowledge generation. We refer to the *developer* who creates a D2K-Package and the *user* who works with it.

Data

The data is generated by the developer of a D2K-Package or integrated from a third-party data provider. Technically, a D2K-Package can link to any data. Conceptually, the D2K-Package is supposed to encourage reuse. Hence, the data should fulfil at least some of the FAIR Principles and be released openly (data complying with the FAIR principles is not necessarily openly accessible[35]). This means that the data should be described with rich metadata to ensure *findability*. In addition, *accessibility* should be granted by linking to each input data needed to reproduce the analysis. Ideally, the URL to the data is a permanent identifier (e.g., a DOI) to ensure the data is time-stamped, long-term accessible, and immutable. To ensure *interoperability*, the data should be stored in text-based formats (e.g., csv or json). Binary formats (e.g., GeoPackage, shapefile) are possible if they have an open specification.

Finally, the data should be released under an open license (ideally CC0[36]) to ensure *reusability*. Open licenses that restrict remixing, transforming, or building upon the data (e.g., CC-BY NoDerivatives[37]) should not be used in a D2K-Package. These aspects become relevant at the latest when the D2K-Package is published. However, during the development of a D2K-Package, the data can be kept private.

Toolbox

Besides the data, a D2K-Package links to a Toolbox created by the developer. The Toolbox contains the containerized analysis code (e.g., written in R or Python) used to generate the results in a research article and a specification of the computational environment. The source code is split into reusable functions that fulfil a specific task (e.g., processing data, running a statistical analysis). The functions follow the "input - processing - output" logic, meaning that each function takes some data (e.g., a table, figure) and a parameter configuration as inputs, runs the task on that data based on the configuration, and outputs the result (e.g., a table, figure). Furthermore, the functions are designed such that they do not depend on the other Toolbox functions and can be called individually. To ensure that the developer's original computational environment can be recreated in the container, the Toolbox contains a specification of the runtime (e.g., R version) and a list of libraries and their versions.

A D2K-Package links to at least one Toolbox but it can reference other developers' Toolboxes from which functions are reused. As with the data, the FAIR principles also apply to the Toolbox, considering software-specific characteristics[18] and the Toolbox should be made openly available at the latest when the D2K-Package is published.

The Toolbox and the data represent the central components in the concept of a D2K-Package and form the reproducible basis upon which the potential of reproducible research can be unlocked. In the following sections, we will see that virtual labs, web API services, and workflows can be created on top of the Toolbox without changing it.

Virtual lab

In the context of this work, a virtual lab is a web application where the runtime and all software dependencies are installed in a ready-to-use programming instance, such as a JupyterLab environment providing a certain python runtime and all necessary packages in a predefined version. A D2K-Package links to such a virtual lab that is created by the developer of a D2K-Package based on the description of the computational environment contained in a Toolbox. Users can follow the link to the virtual lab and immediately start interacting with the code in the browser without the challenge of restoring the computational environment locally. They can develop the code further, try it out and change it in an executable environment, or use it in hands-on classes together with students. Since all users work in the same environment, differences in the results caused by differing library or runtime versions[38,39] can also be avoided.

Web API service

A web API service allows users to send requests to that service and receive a response via HTTP. Based on this request-response mechanism (also known as client-server communication), the developer of a D2K-Package can expose the functions in a Toolbox by setting up a server and deploying a web API service. The concept of a Toolbox can be seamlessly integrated into this mechanism since its functions are encapsulated, containerized, and well-defined regarding input and output. The web API service acts as a wrapper that enables immediate reuse of a Toolbox function. The wrapper takes the input information sent by the user (i.e., data, parameter configuration), executes the function, and outputs the result as part of the response for

further processing in the user's source code. Every function in the Toolbox is wrapped in such a way and has a dedicated endpoint that can be reached via a URL. A D2K-Package links to such a web API service provided by the developer of the D2K-Package.

Computational workflow

A computational workflow is a chain of steps, each running a computational process. Such an analysis pipeline, which should be scripted according to the requirements in reproducible research, comprises steps such as data import, processing, analysis, and visualization. The concept of a Toolbox with its set of functions and the web API service exposing these functions seamlessly integrate with the idea of computational workflows. The developer of the D2K-Package can describe the analysis pipeline using a human-readable workflow language (e.g., Common Workflow Language). Such a workflow description can be created based on the Toolbox functions made available via the web API service, without having to change them. As the Toolbox functions are self-contained and independent, they can be extracted from the workflow context and combined with other functions. A workflow described in this form increases the reusability of the analysis by providing a quick overview and information on which functions are used in which order, the input parameters, and the outputs. The concrete implementation details of the functions are abstracted away from the user but can be explored in the virtual lab. A D2K-Package links to one or more computational workflows created by the developer of the D2K-Package. One can reuse the entire workflow or parts of it by combining it with functions used in other workflows.

Results

We have applied the concept of the D2K-Package to a real research use case to demonstrate its applicability. In the following, we briefly present the hydrological use case and then describe the realization of the D2K-Package based on this use case.

A hydrological use case

The use case explores hydrological dynamics by addressing changes in water optical properties in the Gulf of Riga — a semi-enclosed subbasin of the Baltic Sea. Changes in water transparency and colour in the Gulf of Riga are pressing ecological concerns[40], as they can significantly impact underwater habitats, biodiversity, and ecosystem services. However, the exact causes of these changes remain unclear, highlighting the need for a comprehensive investigation into the hydrological and biogeochemical processes driving these phenomena.

The use case is structured around several consecutive research questions aimed at exploring the interplay between land, freshwater, and marine ecosystems, with a focus on the River Daugava-Gulf of Riga continuum. For D2K-Package presentation purposes, we focus on the workflow addressing the first research question "*Do the optical properties in the Gulf of Riga water change in the long term?"* by analysing long-term changes in water transparency in-situ measurements.

Data: The use case is based on two input datasets from third-party providers, which are both released under an open license but not under a permanent identifier. For reproducibility reasons, both input datasets are made available via Zenodo (see Data Availability)[41]. One dataset is made available in geojson format and comes from an OGC API Features service[42]. It can be directly downloaded from a URL and passed to a function in the Toolbox or to a web API service. In the context of the use case, this data holds information about long-term changes in water optical properties of the Gulf of Riga. The dataset includes key variables such as geographical coordinates (latitude and longitude), sampling station identifiers, sampling dates, and corresponding water transparency (Secchi depth, m) and water colour measurements (expressed in Forel-Ule scale). The D2K-Package contains a link to the OGC API Features service as the original data source and a DOI to the Zenodo record.

The second dataset is provided as a shapefile by the Baltic Marine Environment Protection Commission, which is also known as the Helsinki Commission (HELCOM)[43]. This dataset includes information about assessment units for the Baltic Sea. The polygons are based on HELCOM open sea basins and are integrated with coastal water types as defined in the Water Framework Directive. The associated shapefile contains an attribute table with information, including country designation, subbasin identification (HELCOM_ID), subbasin level classification (e.g., Gulf of Riga), water type name (e.g., Gulf of Riga transitional waters), and additional data such as the area of the polygon.

For this specific dataset, users need to accept a data usage disclaimer before they can copy the download link and pass it to the function in the Toolbox. However, the download link becomes inactive after several minutes. Unfortunately, the inactive link does not redirect the user back to the page where they can accept the disclaimer again to reactivate the download link. Hence, it is disadvantageous to add the download link to the D2K-Package. Instead, we added the link to the website showing the data disclaimer, supplemented by the DOI to the Zenodo record.

Toolbox: The analysis pipeline developed in the use case is written in the open-source scripting language R[32]. In the context of the use case, this analysis employs a workflow for data processing, trend analysis, and visualization. By identifying temporal and spatial patterns of environmental variables, the workflow supports the exploration of underlying links between hydrological processes and water optical properties in the Gulf of Riga.

We divided the analysis into seven functions, each accepting URLs to input datasets and producing outputs that, except for the last functions in the pipeline producing visualizations, are further used as inputs in the next function. Hence, also intermediate results can be investigated or used in different contexts. The functions and the specification of the computational environment are encapsulated in a Docker container and stored on GitHub. Docker is an open-source solution to build and run containerized applications[44] and has proven to be particularly useful in the context of reproducible research, for example, when it comes to handling complex dependencies[29,45].

To specify the versions of the runtime and the libraries used in the Toolbox, we created an environment.yml file and used

Anaconda as a package hosting service. To also take other reproducibility guidelines into account, the repository contains an open-source license to clarify usage (i.e., Apache License 2.0) and a README to document the content. A DOI-linked and citable version of the repository is made available on Zenodo, which was chosen because of its seamless integration with GitHub. The D2K-Package contains a DOI to the Zenodo record and a URL to GitHub.

Virtual Lab: We chose MyBinder to create a virtual lab based on the Toolbox. MyBinder is an open-source web application that recreates a computational environment based on a git repository containing information on the runtime, libraries and their versions, and the source code scripts[46]. Users can engage with the code more deeply in a ready-to-use virtual lab providing the corresponding programming environment (in this case RStudio). The resulting virtual lab can be accessed and shared via a URL with other users and thus become part of a D2K-Package.

On the downside, MyBinder is a test instance with limited resources and does not guarantee to be operational when needed. However, it is also possible to use any other MyBinder web service or deploy and host an own instance.

Web API service: Interoperability is an essential pillar of the FAIR principles and the use of standards can help to achieve it. In order to further increase reusability and leverage community standards, the web API service was realized based on the OGC API Process standard[47]. Such a process offers an endpoint to execute an algorithm via HTTP and is well-defined concerning input parameters and outputs. Usually, these processes take a link to the input data and also generate a link to the output data. Hence, the concept of an OGC API Process can be used to make the Toolbox functions available as a web API service. To implement the web API service based on the OGC API standard, we made use of pygeoapi, an open-source solution to deploy RESTful OGC API services[48]. We have created an OGC API process file for each Toolbox function. The file is responsible for taking the input parameters and creating a Docker command to execute the function.

A D2K-Package contains a link to the OGC API service. From that link, users can receive a list of available processes (i.e., the Toolbox functions via GET /processes), execute a process (POST /{processId}/execution), and receive metadata (GET /processes/{processId}). The metadata also contains a link back to the GitHub repository containing the Toolbox and the corresponding Zenodo record enabling users to explore the underlying source code and cite the resource.

Computational workflow: An important selection criterion for the right workflow tool is the availability of a drag-and-drop workflow editor. Such a canvas can support users without programming knowledge in the development, execution and modification of readily-sharable workflows. For this requirement, the Galaxy platform[31] seems to be better suited than related software solutions such as Nextflow[29] and Snakemake[30]. Galaxy is an actively maintained open-source web application that provides such a user-friendly interface. In addition, Galaxy has a large number of pre-installed tools, is extensible, has a large user community, and runs on a shared public cloud-based infrastructure.

Integrating each Toolbox function separately as a tool into the Galaxy platform is possible but inefficient if the number of functions is large. For this reason, we developed a Galaxy tool that wraps the OGC API service and makes all exposed processes (i.e., each Toolbox function) available in one Galaxy tool (see Figure 5). Users can select the processes one after the other and fill the corresponding input parameters. The functions can then be connected to a workflow to show the entire analysis pipeline including all steps from data to knowledge generation (e.g., data input, pre/post-processing, analysis, and visualization).

Figure 6 shows the analysis pipeline of the hydrological use case implemented as a Galaxy workflow. The two datasets introduced above are used as inputs. The functions are executed one after the other, whereby the output of one function becomes the input of the subsequent function. Users can click on each function to check and modify the configuration and then re-run the workflow. The workflow can be shared with other users within the Galaxy platform and released on Zenodo. These two links are included in the final D2K-Package.

ro-crate metadata file: We created the ro-crate file for the D2K-Package using Describo[49], an open-source metadata editor for describing data. For the components of the D2K-Package, we reviewed the schema.org and bioschemas.org vocabularies to find the most appropriate resource types. As a result, we used "Dataset" for the input data, "SoftwareSourceCode" for the Toolbox, "SoftwareApplication" for the MyBinder-based virtual lab, "WebAPI" for OGC API-based web server, and "ComputationalWorkflow" for the Galaxy-based workflow. We published the file on Zenodo, an open-source repository hosted by CERN providing versioned DOIs and a profound set of metadata. Research assets can be added to the D2K-Package creating a new version and a new DOI of the record. The DOIs of previous versions are still available.

In summary, we have used the following technologies to realize the concept of a D2K-Package. The data is made available via DOIs provided by Zenodo. The code in the toolbox is written in R, containerized with Docker, and stored on GitHub. In addition, the virtual lab is created with MyBinder. The web API service is implemented in Python and is based on the pygeoapi framework, which follows the OGC API Processes standard. We used the Galaxy platform to implement the workflow. Finally, the ro-crate metadata file was used to implement the D2K-Package.

As a result, users can reproduce the analysis pipeline in three different ways: First, using Galaxy and the AquaINFRA Interaction Platform. The instructions for this are provided in the Zenodo repository hosting the D2K-Package. Second, using the OGC API Processes hosted on the pygeoapi server accessible

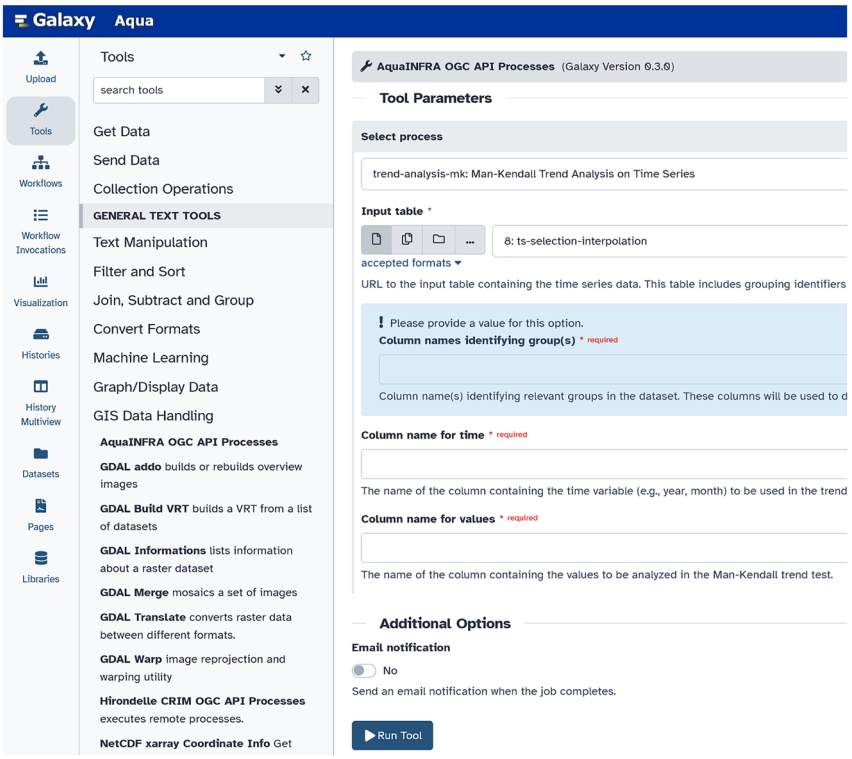


Figure 5. Integration of the OGC API Processes into the Galaxy platform (link to Galaxy, link to the code). Users can find the tools we developed under GIS Data Handling – AquaINFRA OGC API Processes. Then, users can select the function from the dropdown list and complete the corresponding form, which is different for every function.

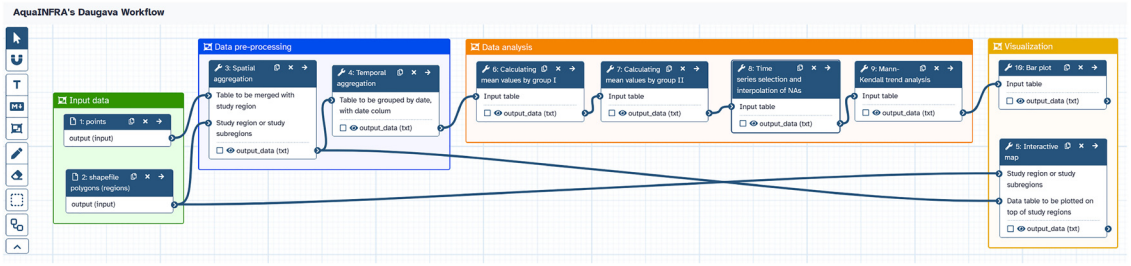


Figure 6. Galaxy workflow showing the analysis pipeline of the hydrological use case. The workflow is composed of seven steps (excluding input datasets, one function is used twice) and shows the entire analysis pipeline from data import to visualization.

via HTTP requests. For this, cURL commands are provided in the README of the toolbox. Finally, using Docker locally. To achieve that, users can run the Docker commands in the README of the toolbox locally one after the other after cloning the repository and building the Docker image. Consequently, the analysis pipeline is reproducible even if Galaxy is not available in the future or if the AquaINFRA Interaction Platform is shut down. Since the functions can be executed locally using Docker, reproducibility also does not depend on the running pygeoapi server.

The final output of the workflow is shown in Figure 7. The results of the Kendall's Tau analysis indicate that long-term changes in Secchi depth, which serves as a proxy for the bulk of optical properties, in the Gulf of Riga are negative in all statistically significant cases. Thus, the answer to the research question is affirmative: the optical properties of the Gulf of Riga do change in the long term, with the clearest signal emerging during summer.

User interface

Figure 8 shows a possible user interface for a D2K-Package based on the ro-crate metadata file. The landing page first shows the metadata such as title, authors, and a description. Below this is the so-called *Virtual Research Environment* (VRE), consisting of a Galaxy-based workflow, a

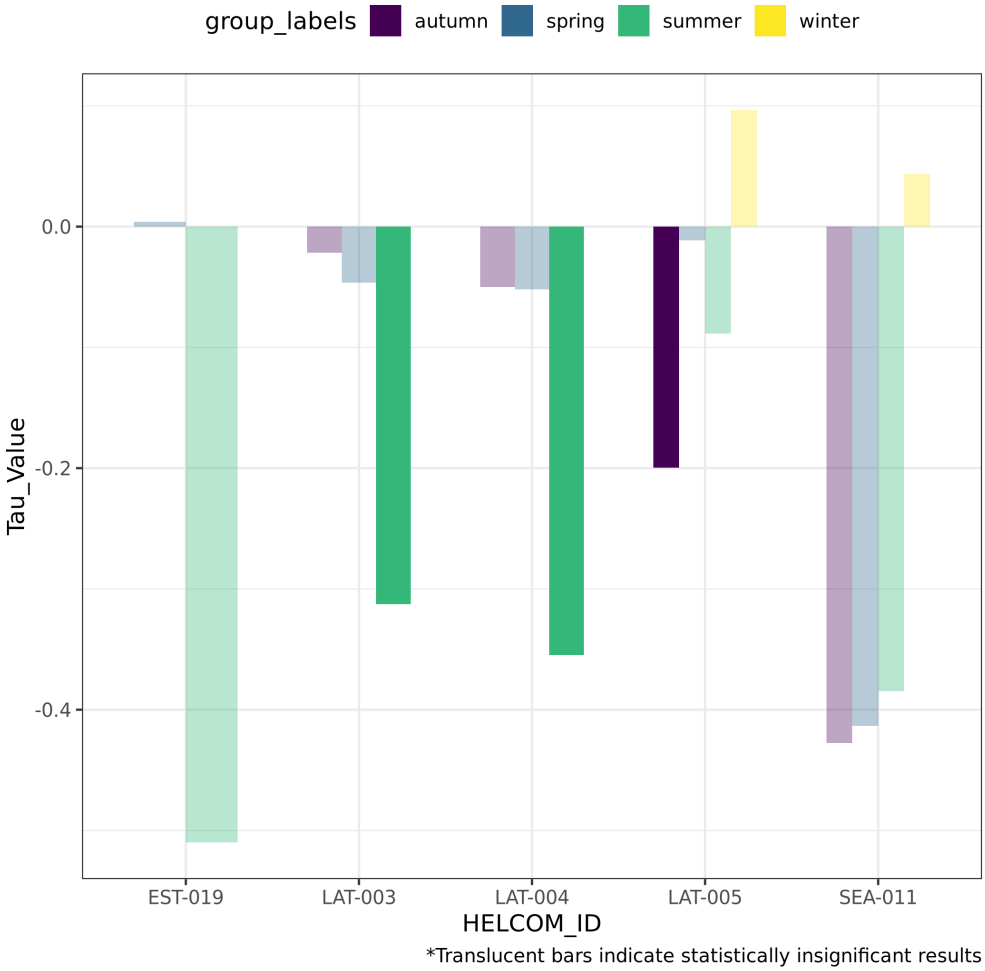


Figure 7. The final output of the workflow answering the research question.

MyBinder-based virtual lab and a pygeoapi-based web API service. The tile for the workflow was placed at the beginning, as it is intended to serve as the first entry point into the analysis. Clicking on it redirects the user to the Galaxy platform. Similarly, the virtual lab tile directs users to MyBinder and the Web API tile to the pygeoapi server. Below the VRE, users see the reproducible basis, consisting of input data tiles redirecting users to the website where they can find the download link and a Toolbox tile redirecting users to the GitHub repository.

Discussion

The goal of the Data-to-Knowledge Package (D2K-Package) is to help researchers understand the path from data to knowledge generation and reuse open FAIR data and code. The D2K-Package aims to foster collaboration among researchers and facilitate knowledge dissemination, reducing redundant efforts and accelerating scientific progress. It incorporates reproducible code and data at its core, while providing multiple opportunities to understand and build upon the work. We applied the D2K-Package concept to a real research use case and demonstrated how the D2K-Package and its components (i.e., *Data*, *Toolbox*, *Virtual Lab*, *Web API Service* and *Computational Workflow*) can be designed.

The question arises as to how the D2K-Package relates to other technologies that support reusability. Code Ocean, for example, offers a virtual environment in which users can execute and work with the source code. The D2K-Package does not compete with Code Ocean. Instead, one of its capsules could act as a virtual lab and be linked from a D2K-Package to become part of it. However, there is still a conflict between the commercial background of Code Ocean and the D2K-Package, which focuses on open-source technologies.

One might also argue that creating a Docker image integrating an execution environment and providing an own API service is an alternative to the web API service based on pygeoapi as suggested in this work. Similar to the previous case, such an approach is possible and could be linked from a D2K-Package. However, we suggest using frameworks such as

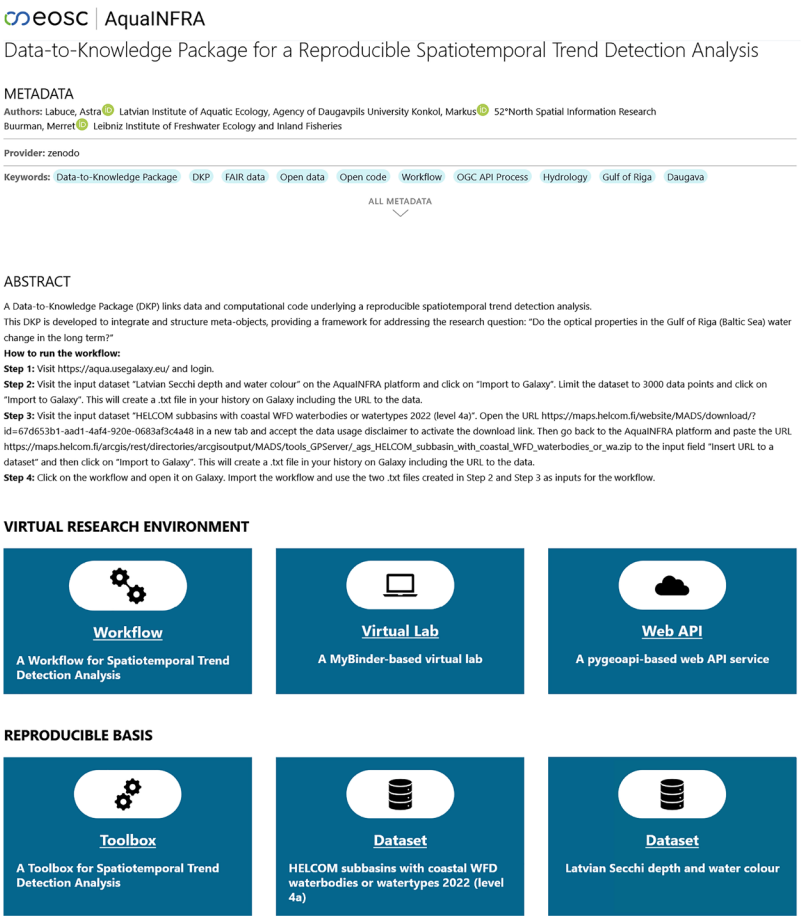


Figure 8. The landing page of a Data-to-Knowledge Package (https://aquainfra.dev.52north.org/result/zenodo:17143387). Top: Metadata and a description. Middle: Virtual Research Environment linking to the workflow, virtual lab, and web API. Bottom: Reproducible basis linking to the toolbox and the input datasets.

pygeoapi to ensure adherence with community standards, for instance, OGC API Processes.

Next, the advantages of developing a D2K-Package are discussed, but the challenges and limitations are also critically examined.

Research cycle

The D2K-Package offers several opportunities to unlock the potential of a reproducible analysis throughout the research cycle. The first entry point to the analysis is the workflow, which can help users understand the analysis and underlying configuration. By using tools such as Galaxy, users can re-run the workflow, modify it, or reuse parts of it in other contexts. This can be useful for researchers who are compiling a literature review at the start of a research project and trying to find related work to build on. Facilitating the reuse of an analysis entails the risk that the code will be reused without being scrutinized. The D2K-Package should not be understood as a shortcut to reusing research materials without having a thorough understanding of the implementation details. Instead, users can delve deeper into the code by exploring the virtual lab including the source code files. This virtual lab can be beneficial for other researchers who need a deeper understanding of the code before they can reuse parts of it. They do not have to spend time restoring the computational environment, but can start immediately. A researcher developing an analysis to answer a research question might want to reuse some of the functions provided by the Toolbox and can do so by integrating the remote OGC API Processes into the script. Experiencing the concrete benefits of this approach might incentivize researchers to develop a D2K-Package out of their own analysis before submitting the report to a conference or journal. Doing so can help them to ensure that the analysis is reproducible, verifiable, and reusable. Moreover, the resulting D2K-Package submitted for peer review can later be updated and expanded to also reference the peer-reviewed publication.

The research cycle shows the steps that can be supported by the Data-to-Knowledge Package.

A D2K-Package, especially the workflow, can also be a helpful tool for reviewers who already have to invest a considerable amount of time in reviewing the paper and do not have the time to familiarize themselves with the analysis underlying the results of the paper. Users who do not have the necessary skills can also benefit from the workflow, as no knowledge of a programming language is required to follow the analysis, modify the underlying configuration, and use a different dataset (also known as "replication"[50]).

Use case

Applying the D2K-Package to the use case revealed a number of challenges. The input data can become an issue if users need to accept data usage disclaimers or login before they can access the download link. This is an issue with respect to machine-readability as it is not possible to use the download link in the code without manual action.

The analysis pipeline was almost completed by the researcher working on the use case when we started to apply the D2K-Package to it. As a result, it was necessary to change the code. For example, the entire code had to be split into individual reusable functions that follow the logic "input - processing - output" instead of having a series of lines of code that can be executed one after the other. Consideration also needed to be given to how the code could be split into functions and how small the functions should be. It is important to find the right balance to design functions that are small enough to be reused in other contexts while still performing a meaningful task. Ultimately, each developer decides for themselves when a function is reusable. In addition, developers need to think carefully about which input parameters should be passed to the function and what the intermediate results are. Such problems are well known in connection with the development of software libraries, but can still pose a major challenge for non-software developers. For a D2K-Package to be practicable, it should therefore be considered from the start in order to avoid too much work afterwards, which is anyway recommended to ensure reproducibility[51]. The following list of prerequisites might help potential users to determine whether a D2K-Package fits their work.

- Data: While theoretically data in a D2K-Package can be kept private, it contradicts the idea of reusability and should be shared openly unless there are good reasons not to. This requires a repository with sufficient data storage.
- Toolbox: A D2K-Package can only include scripted analyses written in open-source programming languages. A sequence of clicks in a graphical user interface describing the analysis process cannot be considered.
- Virtual lab: In the case of MyBinder, the code needs to be stored on one of the integrated technologies, for instance, GitHub, GitLab or Zenodo.
- Web API Service: The developer of a D2K-Package needs access to a server that runs a service software, for instance, pygeoapi.
- Workflow: For a workflow, the code needs to be structured into independent and self-contained functions following the input-processing-output logic. Other developed software, such as a user interface cannot become a workflow.

In addition, we have created a user guide with concrete steps for developing a D2K-Package as a living document, i.e. we will continue to improve it based on user feedback[52].

A persistent counterargument in connection with reproducible research is the time required for this. Based on our experience with the use case, it cannot be denied that this challenge also applies to the creation of a D2K-Package. Hence, it is unrealistic to expect that every kind of workflow is transformed into a D2K-Package, though a transformation is not required if the creation of a D2K-Package is considered from the beginning. Possible guiding questions to decide whether a D2K-Package is meaningful or not are:

- Should other researchers be able to reuse parts of the analysis in their own work?
- Should the analysis pipeline be presented as a workflow using, e.g., Galaxy?

If both answers are negative, the creation of a D2K-Package will probably not bring added value. However, if one or even both questions can be answered in the affirmative, a D2K-Package is a low-hanging fruit. The reason for this is that in order to create workflows or reusable code, the functions have to be developed in such a way that they can be called and executed individually anyway. In addition, containerization is necessary anyway to ensure executability. Consequently, the creation of a Toolbox is no longer a major additional expense.

Creating a MyBinder-based virtual lab is easy once a Toolbox has been created, as it contains all the necessary information. However, resolving dependency conflicts (e.g. incompatible library versions) can be a challenge. Moreover, a Toolbox also makes it easier to provide OGC API processes, as these only need to execute the containerized functions. Integrating the functions into the Galaxy platform, on the other hand, requires a certain amount of additional work, as the necessary tools have to be created.

Reproducibility

The core of a D2K-Package is the reproducible basis composed of data and code. However, the D2K-Package does not have an explicit mechanism to check and guarantee that the analysis generates the same results. Suppose a researcher uses a random number in the script but does not set a seed as recommended in literature[19]. The analysis script is still executable, but the results could be different, which does not meet the requirement for consistent results. A similar problem can occur if the developer of a D2K-Package does not specify the computational environment sufficiently, which can also lead to divergent numbers, figures, or tables[53].

One possibility to mitigate these issues is to add automatic tests to the toolbox. As the analysis is divided into individual, self-contained functions, unit tests could be suitable for checking reproducibility. This can be achieved with the help of GitHub Actions, a tool with which code can be executed after the occurrence of a certain event, e.g. after a commit or regularly at a certain point in time. Also, libraries can be used in the code to test for reproducibility, such as *testthat* for R[54].

Another option is to add a folder containing the results to the toolbox or even the raw data[19]. Users would then receive the chance to run the analysis and compare the new output to the original results. Tools such as Galaxy's GraphicsMagick[55] can help to spot differences between two images.

Finally, a review process could be implemented to check for adherence to the requirements of a D2K-Package.

Persistence

Making the D2K-Package persistent can be achieved by publishing it in a repository that assigns DOIs to its records. It is more difficult to ensure the persistence of the individual components linked from a D2K-Package. Ideally, all components generated by the developer of the D2K-Package are published in a repository offering DOIs. A Toolbox maintained on GitHub can be published permanently via the GitHub-Zenodo integration. The Galaxy workflow and the source code files responsible for the OGC API Processes can also be published on Zenodo.

However, we have seen in the hydrological use case that the data may only be available under a URL and not a DOI. Therefore, if the data is not maintained in the long term, either the data or the link to the data may change over time, which affects reproducibility. One way to mitigate this problem is to download the data and publish it in a repository such as Zenodo, if the license allows it. This approach would facilitate access and assign a permanent identifier to the data. However, this approach leads to duplication and is only possible if the file size restriction of the repository allows it, which is usually not the case for big data. This limitation is not specific to the D2K-Package, but shows where the idea of reproducible research and FAIR and open data conflicts with reality. Furthermore, the problem is not an exceptional case, but can always occur if the researcher has not generated the data and has no control over it. The responsibility lies with the data providers, who must provide open and FAIR data that is also machine-readable.

One might argue that DOIs are not enough for publishing source code persistently. While DOIs are a widely accepted and well-known solution in the scientific field, the Software Hash Identifier (SWHID, formally known as Software Heritage Identifier) is a globally accepted framework tailored to source code[56]. One of the benefits of using Zenodo is that it has Software Heritage integrated, meaning that every software record hosted on Zenodo is automatically made available via Software Heritage. This integration results in a bi-directional linking from Zenodo to Software Heritage and vice versa.

Portability

The D2K-Package is designed to facilitate the portability of individual components to ensure sustainability and longevity. The code is encapsulated in a Docker container, which can be deployed wherever needed and maintained on any web application based on git, such as GitHub or GitLab. The MyBinder instance used to realize the Virtual Lab is a test instance. It is not guaranteed to work at all times and has limited resources, which reduces its capacity when many users access the instance simultaneously. Nevertheless, the Toolbox is not tied to a specific MyBinder deployment but can run on any other instance offering the same functionality.

A similar problem exists with the pygeoapi-based web API service and the Galaxy platform which require a server. While the Galaxy tool wrapping the OGC API Processes can only run on a Galaxy platform, it is not bound to a particular Galaxy instance. Every operator hosting a Galaxy instance can also run that tool. Thus, also the workflows can be run on different Galaxy instances. We see the responsibility for providing server capabilities not with the researchers but with those who have reproducibility as a requirement, such as funders and publishers. Initiatives like the European Open Science Cloud and their concept of a "Node" can be a promising opportunity to provide access to such tools and services. Yet, the realization of the D2K-Package in this work is just a reference implementation. It can be realized using different technologies. The workflows, for instance could also be realized using other tools depending on the user requirements, such as Nextflow[29] to address experienced programmers.

In addition to sustainability, the development of a portable solution is advantageous in the case of big data. For example, the Toolbox could be deployed where the data is stored to avoid data transfer.

Ethics and consent

Ethical approval and consent were not required.

Data availability

Source data

Zenodo: Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case (Input data)[41]. https://doi.org/10.5281/zenodo.15234377

- The analysis used in the hydrological use case takes two input datasets coming from a third-party data provider released under an open license.
    1. The first input dataset[43] can be downloaded after accepting the data usage disclaimer.
    2. The second input dataset[42] of the workflow can be downloaded from an OGC API Features service.

All datasets are released under an open license (Creative Commons Attribution 4.0 International).

Software availability

Zenodo: Data-to-Knowledge Package[57]. https://doi.org/10.5281/zenodo.17143387

Zenodo: A Toolbox for Spatiotemporal Trend Detection Analysis[58]. https://doi.org/10.5281/zenodo.17143347

Zenodo: Galaxy workflow[59]. https://doi.org/10.5281/zenodo.15350861

Zenodo: Source code for the user interface[60]. https://doi.org/10.5281/zenodo.15129277

Zenodo: Galaxy tool[61]. https://doi.org/10.5281/zenodo.15234678

GitHub: A Toolbox for Spatiotemporal Trend Detection Analysis: https://github.com/AstraLabuce/aquainfra-usecase-Daugava/tree/containerize

Galaxy workflow: https://aqua.usegalaxy.eu/u/markus.konkol/w/copy-of-adj-aquainfras-daugava-workflow-shared-by-user-alabuce

Galaxy tool: https://github.com/galaxyecology/tools-ecology/tree/master/tools/aquainfra_ogc_api_processes

User interface: https://aquainfra.dev.52north.org/result/zenodo:17143387

OGC API server: https://aquainfra.ogc.igb-berlin.de/pygeoapi/

All software outputs are released under an open license (Apache License 2.0).

Acknowledgements

We are grateful for inspiring discussions with and support from Jörg Meyer (Karlsruhe Institute of Technology, DFG project ISABEL - 496155047) and Björn Grüning (Head of the Freiburg Galaxy Team).

References

1. Tollefson J, Kozlov M, Witze A, *et al.*: **Trump's siege of science: how the first 30 days unfolded and what's next.** *Nature.* 2025; **638**(8052): 865–867.
PubMed Abstract | Publisher Full Text
2. Matthews D: **Far-right governments seek to cut billions of euros from research in Europe.** *Nature.* 2024; **635**(8037): 15–16.
PubMed Abstract | Publisher Full Text
3. Costello MJ: **Motivating online publication of data.** *BioScience.* 2009; **59**(5): 418–427.
Publisher Full Text
4. Goodman SN, Fanelli D, Ioannidis JPA: **What does research reproducibility mean?** *Sci Transl Med.* 2016; **8**(341): 341ps12.
PubMed Abstract | Publisher Full Text
5. Stodden V, Bailey DH, Borwein J, *et al.*: **Setting the default to reproducible: reproducibility in computational and experimental mathematics**. 2013.
Reference Source
6. Konkol M, Kray C, Pfeiffer M: **Computational reproducibility in geoscientific papers: insights from a series of studies with geoscientists and a reproduction study.** *Int J Geogr Inf Sci.* 2019; **33**(2): 408–429.
Publisher Full Text
7. Culina A, van den Berg I, Evans S, *et al.*: **Low availability of code in ecology: a call for urgent action.** *PLoS Biol.* 2020; **18**(7): e3000763.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Hutson M: **Artificial Intelligence faces reproducibility crisis.** *Science.* 2018; **359**(6377): 725–726.
PubMed Abstract | Publisher Full Text
9. Baker M: **1,500 scientists lift the lid on reproducibility.** *Nature.* 2016; **533**(7604): 452–454.
PubMed Abstract | Publisher Full Text
10. McCullough BD, McGeary KA, Harrison TD: **Do economics journal archives promote replicable research?** *Can J Econ.* 2008; **41**(4): 1406–1420.
Publisher Full Text
11. Collberg C, Proebsting TA: **Repeatability in computer systems research.** *Commun ACM.* 2016; **59**(3): 62–69.
Publisher Full Text
12. Gil Y, David CH, Demir I, *et al.*: **Toward the Geoscience Paper of the Future: best practices for documenting and sharing research from data to software to provenance.** *Earth Space Sci.* 2016; **3**(10): 388–415.
Publisher Full Text
13. Bahaidarah L, Hung E, de Melo Oliveira AF, *et al.*: **Toward reusable science with readable code and reproducibility.** *International Conference on e-Science.* 2022; 437–439.
Publisher Full Text
14. European Commission: **Open science in horizon Europe**.
Reference Source
15. Stark PB: **Before reproducibility must come preproducibility.** *Nature.* 2018; **557**(7707): 613.
PubMed Abstract | Publisher Full Text
16. **Trust but verify**. *Nat Mater.* 2024; **23**(1): 1.
PubMed Abstract | Publisher Full Text
17. Wilkinson M, Dumontier M, Aalbersberg JJ, *et al.*: **The FAIR guiding principles for scientific data management and stewardship.** *Sci Data.* 2016; **3**: 160018.
PubMed Abstract | Publisher Full Text | Free Full Text
18. Barker M, Chue Hong NP, Katz DS, *et al.*: **Introducing the FAIR principles for research software.** *Sci Data.* 2022; **9**(1): 622.
PubMed Abstract | Publisher Full Text | Free Full Text
19. Sandve GK, Nekrutenko A, Taylor J, *et al.*: **Ten simple rules for reproducible computational research.** *PLoS Comput Biol.* 2013; **9**(10): e1003285.
PubMed Abstract | Publisher Full Text | Free Full Text
20. Gentleman R, Lang DT: **Statistical analyses and reproducible research.** *J Comput Graph Stat.* 2007; **16**(1): 1–23.
Publisher Full Text
21. Chirigati F, Rampin R, Shasha D, *et al.*: **ReproZip: computational reproducibility with ease**. *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16)*. Association for Computing Machinery, 2016; 2085–2088.
Publisher Full Text
22. **Seamless sharing and peer review of code.** *Nat Comput Sci.* 2022; **2**(12): 773.
PubMed Abstract | Publisher Full Text
23. Nogueira Alves A, Oliveira MM, Koyama T, *et al.*: **Ecdysone coordinates plastic growth with robust pattern in the developing wing.** *eLife.* 2022; **11**: e72666.
PubMed Abstract | Publisher Full Text | Free Full Text
24. Bechhofer S, Buchan I, De Roure D, *et al.*: **Why linked data is not enough for scientists.** *Future Gener Comput Syst.* 2013; **29**(2): 599–611.
Publisher Full Text
25. Soiland-Reyes S, Sefton P, Goble C, *et al.*: **Packaging research artefacts with RO-Crate.** *Data Sci.* 2022; **5**(2): 97–138.
Publisher Full Text
26. Group on Earth Observation: **Knowledge package**.
Reference Source
27. Shakil A, Lutteroth C, Weber G: **CodeGazer: making code navigation easy and natural with Gaze input**. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. 2019; 1–12.
Publisher Full Text
28. Nordmann E, McAleer P, Toivo W, *et al.*: **Data visualization using R for researchers who do not use R.** *Adv Methods Pract Psychol Sci.* 2022; **5**(2).
Publisher Full Text
29. Di Tommaso P, Chatzou M, Floden EW, *et al.*: **Nextflow enables reproducible computational workflows.** *Nat Biotechnol.* 2017; **35**(4): 316–319.
PubMed Abstract | Publisher Full Text
30. Mölder F, Jablonski KP, Letcher B, *et al.*: **Sustainable data analysis with**

Snakemake [version 2; peer review: 2 approved]. *F1000Res.* 2021; **10**: 33.
PubMed Abstract | Publisher Full Text | Free Full Text

31. Galaxy Community: **The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update.** *Nucleic Acids Res.* 2024; **52**(W1): W83–W94.
PubMed Abstract | Publisher Full Text | Free Full Text

32. R Core Team: **R: a language and environment for statistical computing**. R Foundation for Statistical Computing, 2025.
Reference Source

33. Nüst D, Konkol M, Schutzeichel M, *et al.*: **Opening the publication process with executable research compendia.** *D-Lib Magazine.* 2017; **23**(1/2).
Publisher Full Text

34. Nosek BA, Alter G, Banks GC, *et al.*: **SCIENTIFIC STANDARDS. Promoting an open research culture.** *Science.* 2015; **348**(6242): 1422–1425.
PubMed Abstract | Publisher Full Text | Free Full Text

35. Higman R, Bangert D, Jones S: **Three camps, one destination: the intersections of research data management, FAIR and open.** *Insights: the UKSG Journal.* 2019; **32**(1): 18.
Publisher Full Text

36. Hrynaszkiewicz I, Cockerill MJ: **Open by default: a proposed copyright license and waiver agreement for open access research and data in peer-reviewed journals.** *BMC Res Notes.* 2012; **5**: 494.
PubMed Abstract | Publisher Full Text | Free Full Text

37. Creative Commons: **Attribution-NoDerivatives 4.0 International**.
Reference Source

38. Piccolo SR, Frampton MB: **Tools and techniques for computational reproducibility.** *GigaScience.* 2016; **5**(1): 30.
PubMed Abstract | Publisher Full Text | Free Full Text

39. Garijo D, Kinnings S, Xie L, *et al.*: **Quantifying reproducibility in computational biology: the case of the tuberculosis drugome.** *PLoS One.* 2013; **8**(11): e80278.
PubMed Abstract | Publisher Full Text | Free Full Text

40. Aigars J, Suhareva N, Cepite-Frisfelde D, *et al.*: **From green to brown: two decades of darkening coastal water in the Gulf of Riga, the Baltic Sea.** *Front Mar Sci.* 2024; **11**: 1369537.
Publisher Full Text

41. Konkol M: **Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case (Input data).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.15234377

42. **Latvian Secchi depth and water colour**. AquaINFRA data discovery and access service, 2025.
https://vm4072.kaj.pouta.csc.fi/ddas/oapif/collections/lva_secchi/items?f=json&limit=5871

43. **HELCOM subbasins with coastal WFD waterbodies or watertypes 2022 (level 4a)**. The Baltic Marine Environment Protection Commission (HELCOM), Helsinki, Finland, 2022.
https://maps.helcom.fi/website/MADS/download/?id=67d653b1-aad1-4af4-920e-0683af3c4a48

44. Merkel D: **Docker: lightweight Linux containers for consistent development and deployment.** *Linux J.* 2014; **239**(2): 2.
Reference Source

45. Boettiger C: **An introduction to Docker for reproducible research.** *SIGOPS Oper Syst Rev.* 2015; **49**(1): 71–79.
Publisher Full Text

46. Bussonnier M, Forde J, Freeman J, *et al.*: **Binder 2.0 - reproducible, interactive, sharable environments for science at scale.** *Proceedings of the 17th Python in Science Conference.* 2018.
Publisher Full Text

47. Open Geospatial Consortium: **OGC API - processes**.
Reference Source

48. Kralidis T, Webb B, Tzotsos A, *et al.*: **geopython/pygeoapi: 0.19.0 (0.19.0).** *Zenodo.* 2025.
Publisher Full Text

49. La Rosa M, *et al.*: **Describo.** 2025.
Reference Source

50. The Turing Way Community: **The turing way: a handbook for reproducible, ethical and collaborative research (1.0.2).** *Zenodo.* 2022.
Publisher Full Text

51. Alston JM, Rick JA: **A beginner's guide to conducting reproducible research.** *Bull Ecol Soc Am.* 2021; **102**(2): e01801.
Publisher Full Text

52. Konkol M, Buurman M: **User guide: creating a Data-to-Knowledge package.** *Zenodo*. 2025.
Publisher Full Text

53. Gronenschild EHBM, Habets P, Jacobs HIL, *et al.*: **The effects of FreeSurfer version, workstation type, and Macintosh operating system version on anatomical volume and cortical thickness measurements.** *PLoS One.* 2012; **7**(6): e38234.
PubMed Abstract | Publisher Full Text | Free Full Text

54. Wickham H: **testthat: get started with testing.** *The R Journal.* 2011; **3**: 5–10.
Reference Source

55. Group G: **GraphicsMagick image processing system.** GraphicsMagick Group, 2017.
Reference Source

56. Di Cosmo R, Gruenpeter M, Zacchiroli S: **Identifiers for digital objects: the case of software source code preservation.** *iPRES 2018 - 15th International Conference on Digital Preservation*. 2022; 1–9.
Publisher Full Text

57. Labuce A, Konkol M, Buurman M: **Data-to-Knowledge package for a reproducible spatiotemporal trend detection analysis (1.4).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.17143387

58. Labuce A, Konkol M, Buurman M: **A toolbox for spatiotemporal trend detection analysis (1.0.1).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.17143347

59. Labuce A, Konkol M, Buurman M: **A workflow for spatiotemporal trend detection analysis (1.0.1).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.15350861

60. Konkol M: **AquaINFRA interaction platform v1.0.1 (1.0.1).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.15129277

61. Konkol M: **AquaINFRA OGC API processes to galaxy (1.0).** *Zenodo.* 2025.
http://www.doi.org/10.5281/zenodo.15234678

Open Peer Review

Current Peer Review Status: ✔ ✔ ✘

Version 2

Reviewer Report 02 September 2025

https://doi.org/10.21956/openreseurope.22545.r56724

© **2025 Tarboton D et al.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✘

David Tarboton
[1] Civil and Environmental Engineering, Utah State University, Logan, Utah, USA
[2] Civil and Environmental Engineering, Utah State University, Logan, Utah, USA
David Rosenberg
[1] Civil and Environmental Engineering, Utah State University, logan, Utah, USA
[2] Civil and Environmental Engineering, Utah State University, logan, Utah, USA

Failed Attempt to Reproduce Results in "Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case."
https://doi.org/10.12688/openreseurope.20221.2.
Reviewer Report
David Rosenberg, David Tarboton

General comments
This manuscript presents "data to knowledge packages" as a solution to encourage reusability and reproducibility of computational research in hydrology. The manuscript is well written. The work is needed. The descriptions of elements of reproducibility are compelling. The key component is a web platform (GALAXY) that allows users to undertake common data management, import, analysis, and visualization activities. These activities include:

- Spatial aggregation: Assigns in situ environmental data points to spatial units based on polygon boundaries
- Temporal aggregation: Groups data points by time intervals for structured trend analysis
- Time Series Processing and Selection such as calculating mean values by group, selecting time series with sufficient data and interpolating missing values (NAs) to generate structured time series datasets optimized for trend detection.
- Trend Detection: such as Mann-Kendall analysis for statistically assessing long-term trends in environmental parameters across spatial units.
- Visualization: Interactive map generation for visualizing in situ data points over spatial assessment units, Trend representation in a static bar plot summarizing detected significant trends.

The manuscript aims to present a hydrological use case for the Gulf of Riga. The motivating

question is: *Do the optical properties in the Gulf of Riga water change in the long term*?

We find that the manuscript falls short in the following aspects:

1. The hydrological use case is not actually given.
2. The web platform (GALAXY) and tools presented are too complex and fragile and end up being a barrier to, rather than helpful towards reproducibility and reusability.
3. We were unable to get the tools to work.
4. We could not see the analysis or results.
5. We did not gain any (new) understanding of the data , hydrologic system, or management of the system.

Thus our overall finding is that while the manuscript encourages reusability and reproducibility, the work was not reproducible itself. We believe it will take a major redesign of how the data, web platform, tools, analysis, and workflow engine are integrated to achieve the reusability and reproducibility goals. The paper fails to offer a contribution either in technical substance, reusability, or reproducibility. There needs to be significant attention to leading and helping a reuser/reproducer establish or gain access to the run time environment and from there on for the process to be automated, while at the same time transparent for this to become a helpful contribution.

A framework for assessing research reusability and reproducibility.

We find this framework helpful for assessing research reproducibility and reusability:

1. Publicly available. Research artifacts should be in a public repository with persistent identifiers (preferably DOI's).
2. Documentation. Writing should overview the project, explain the contents, and list results the authors intend to be reproduced. The documentation should be in human readable file format that will persist (e.g., text or markdown).
3. Organization. Distinguish between Data, Digital Artifacts that operate on Data, and Author generated results.
4. Computational requirements. Provide requirements and instructions to create or set up the run-time environment for the analysis.
5. Workflow. Provide instructions to use artifacts in the runtime environment to generate results.
6. Independent check. A non-author follows the prior steps to generate and validate the authors results.

The first 5 steps address findability, accessibility, interoperability, and reusability (FAIR) principles, while step 6 validates reproducibility.

Steps (1) through (3) are generally well done in this paper. The digital artifacts are available with a persistent digital object identifier in Zenodo (step 1) and included human-readable markdown (README.md) or Text (.txt) files that overviewed the purpose and explain the organization of the contents (Step 2). Across the project, the digital artifacts are nicely separated into data and the software that operates on the data (Step 3).

Our concerns with this paper rest with steps (4) to (6). The complex technical requirements for reproducing what should be standard statistical analysis in an open-source procedural programming language is daunting and seems unnecessary.

Suggestions to improve

1. The paper should show the key computational outputs and results in maps, figures, tables,

etc., so that someone intending to reproduce or reuse this work knows what they are looking for, and so that the motivating question is answered.
2. Simplify the technological requirements. It seems that the work of this paper could be done solely in an open-source procedural language such as Python or R.
3. Make a well documented script that automates all steps of the workflow, including presentation of results.
4. Use a computational environment that will persist. For example, web platforms such as GALAXY may go away, change, or be inaccessible to the reproducer.
5. Provide guidance on more general approaches for the work to be reproduced.
6. Provide instructions to recreate the run time environment, specifically for Docker and Binder. Explain how a person trying to reproduce the work should set up these tools on their computer, particularly on one or more common platforms (Windows/Mac/Linux).
7. Aggregate all the components needed to undertake this work in a single public repository with digital object identifier so that users can find everything in one place. Perhaps the data to knowledge package (https://doi.org/10.5281/zenodo.15354757) is this repository, but make this clear and lead a user on when they get to this.

Below we provide more detailed notes on the technical efforts by each of us to work with the web platform, data, and code provided. These notes can help identify where reproducibility broke down.

Writing

Shrinking research budgets are not in our opinion the leading motivator for reproducible research. The leading motivator is trust in the findings brought about by seeing and understanding the rationale for the findings. We suggest not opening the paper with a statement about shrinking research budgets.

The paper states that a major obstacle is the “perception” that publishing reproducible results is time consuming. This is not a just perception, it is real.

Notes on technical reproducibility efforts by Rosenberg

Source Data

1. Zenodo: Encouraging reusability of computational research through Data-to-Knowledge Packages - A hydrological use case (Input data)[41]. https://doi.org/10.5281/zenodo.15234377

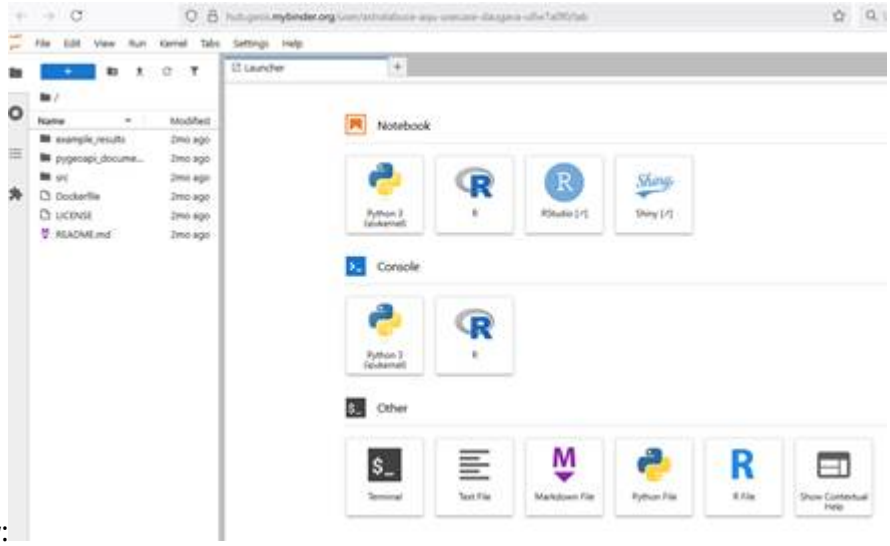
1. I was able to access and download both data sets.
2. This is simply a json file with no meta data or explanation of contents. Insufficient meta data.
    1. No instructions, interpretation on the Zenodo repository.
    2. No explanation of Latvian Secchi depth, units of measurement. Human readable format
3. Similar in the HELCOM dataset.

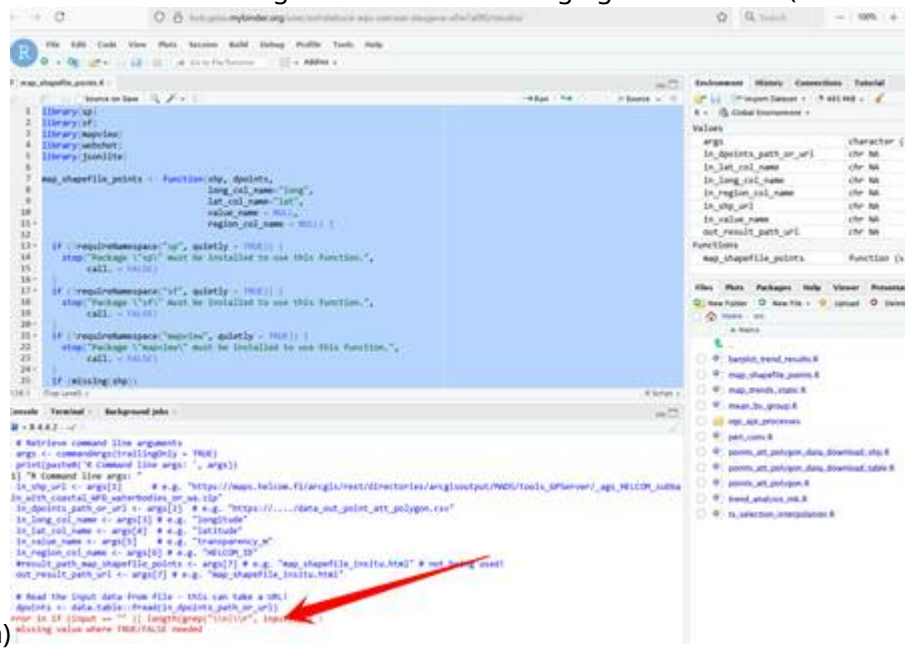
Software availability

a. Zenodo: Data-to-Knowledge Package[57]. https://doi.org/10.5281/zenodo.15354757

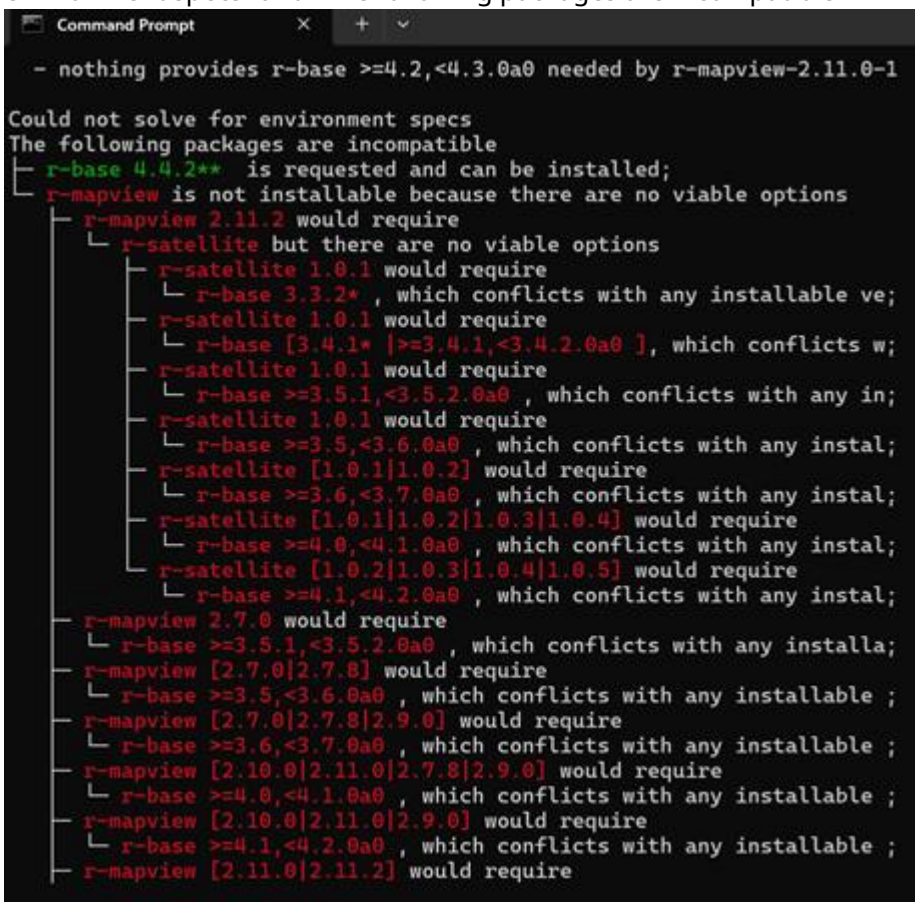
1. Good directions and meta data.
2. No explanation of the file ro-crate-metadata.json.

b. Zenodo: A Toolbox for Spatiotemporal Trend Detection Analysis[58].
https://doi.org/10.5281/zenodo.15350797

1. Repository with DOI – good. Also excellent description of contents in the meta data for the repository.
2. README.md file provides a good overview and explanation of the contents.
3. I followed the link on line 4 to the Binder. This is my first time using Binder. The link opened the following window: 

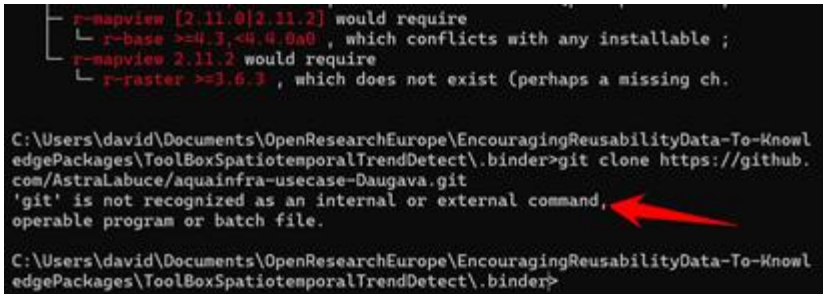
4. What do I do next???
5. Poking around, I found my way to a virtual RStudio Server webpage that looks a lot like the R Studio platform I use a lot on my desktop. Cool!!
6. I was unclear which file to open first.
7. So I opened the map_shapefile_points.R.
8. I selected all the code and ran it. That generated the following figure and error (see red arrow at bottom) 

9. At this point I gave up.
10. Back in the README.md file that I downloaded from Zenodo onto my desktop, starting on line 6, I found the ordering of the sections in the README.md file confusing. It seems the last section, “Create conda environment” needs to happen first?? Order the remaining sections in the order of the steps to reproduce the results.
11. So I tried the following: I opened a cmd prompt (I have windows 11) and was able to execute the 3 lines of code at the bottom of the README.md file under the section “Create conda environment.”
12. The last command executed but gave a bunch of warnings such as “could not solve for environment specs” and “The following packages are incompatible”



Can one still continue even with these errors/warnings?

13. Next I tried to the build the docker image. I continued in the same virtual environment and entered the first line of code in under the section “Building Docker image.” I got an error about unable to run the git command even though I have Git Desktop installed on my machine that I regularly use.



14. I got the same error when I tried to run in the root cmd prompt outside the environment.
15. I stopped here. There is evidently specialized knowledge or directions needed to execute these commands. Please add more directions/instructions.
16. At this point I was also unclear about the order or other steps needed to run the R code in src/ directory, the .json and .py files in the directory src/ogc_api_processes
17. For example, I ran the file test_post_requests.py in the folder pygeoapi_documentation which executed. But the window closed quickly before I could see any output. When I opened the same file in a text editor I could see generally a bunch of code, but it was hard to understand what is actually happening (or supposed to happen). Suggestions to improve:
    1. In the header provide an overview of the steps taken in the code.
    2. For each code section provide more information as comments about what is happening.
18. I opened the example_results folder and found a bunch of csv files. Excellent as this is a common and persistent data sharing format. I opened two of the output files. These can be improved by providing more meta data about the contents of the columns and units of measurement. It will also help to show the mapping between code files in the other folders and the results files. Which code file/section generates which result file?
19. OGC processes – I did not follow this section. It would help to organize this section as the steps the user needs to take.

c. Zenodo: Galaxy workflow[59]. https://doi.org/10.5281/zenodo.15350861

1. I was able to download and extract the files in this repository.
2. The README.md file had nothing in it.
3. Some of the txt files were either empty, had a short instance such as “{"galaxy_export_version": "2"} “, or had a long line of indecipherable code.
4. I stopped here.

d. Zenodo: Source code for the user interface[60]. https://doi.org/10.5281/zenodo.15129277

1. It looks like all the code is here, but there is not meta data in the Zenodo resource or explanation of the contents.
2. Nonetheless, I downloaded and opened the zip file.
    1. The README.md in the root folder provides a nice summary of the platform. Under Installation, I ran into the same error when I tried to execute the first command git clone https:// as above.
    2. The README.md in the docs folder looks like a nice organization of the user guide.
    3. I appreciated the instructions in the GettingStarted.md but did not follow them because they were long and required a lot of manual entries.

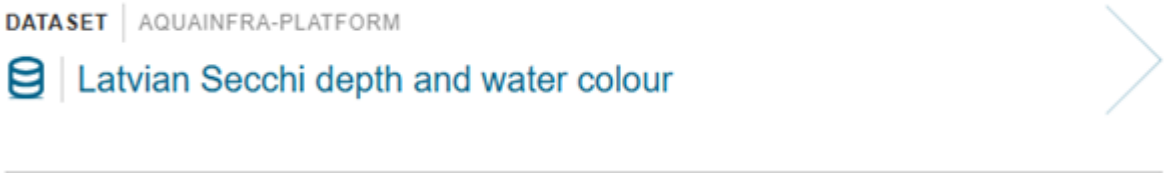
Notes on technical reproducibility efforts by Tarboton

This is a record of the work I did to attempt to reproduce the results.

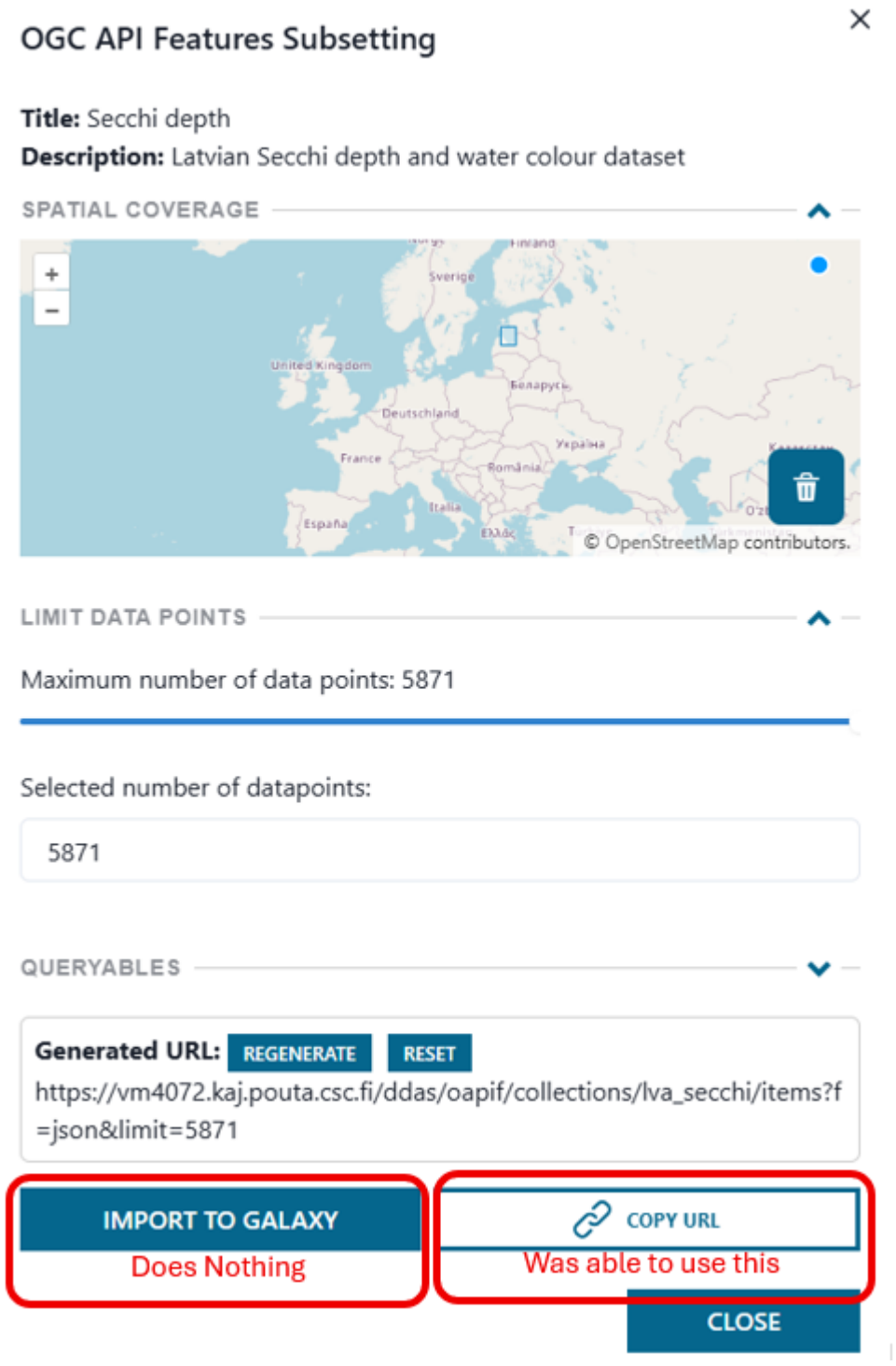
I started from the D2K package given at https://doi.org/10.5281/zenodo.15354757. I tried to follow the steps given (given steps in italics, my comments in plain text):

Step 1: *Visit https://aqua.usegalaxy.eu/ and login.*
I was able to create an account. The dependency on a specific site like this is a limitation. This site may be updated and the procedure may no longer work.

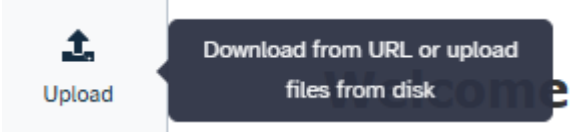
Step 2: *Visit the input dataset "Latvian Secchi depth and water colour" on the AquaINFRA platform and click on "Import to Galaxy". Limit the dataset to 5871 data points and click on "Import to Galaxy". This will create a .txt file in your history on Galaxy including the URL to the data.*
How to achieve this is not clear from the statement. I went back to the paper and guessed based on aquainfra being in the URL that the platform was what was given in the paper as *User interface: https://aquainfra.dev.52north.org/result/zenodo:15354757* This has the same set of instructions, but a search bar at the top. I guessed and put in. One dataset was located



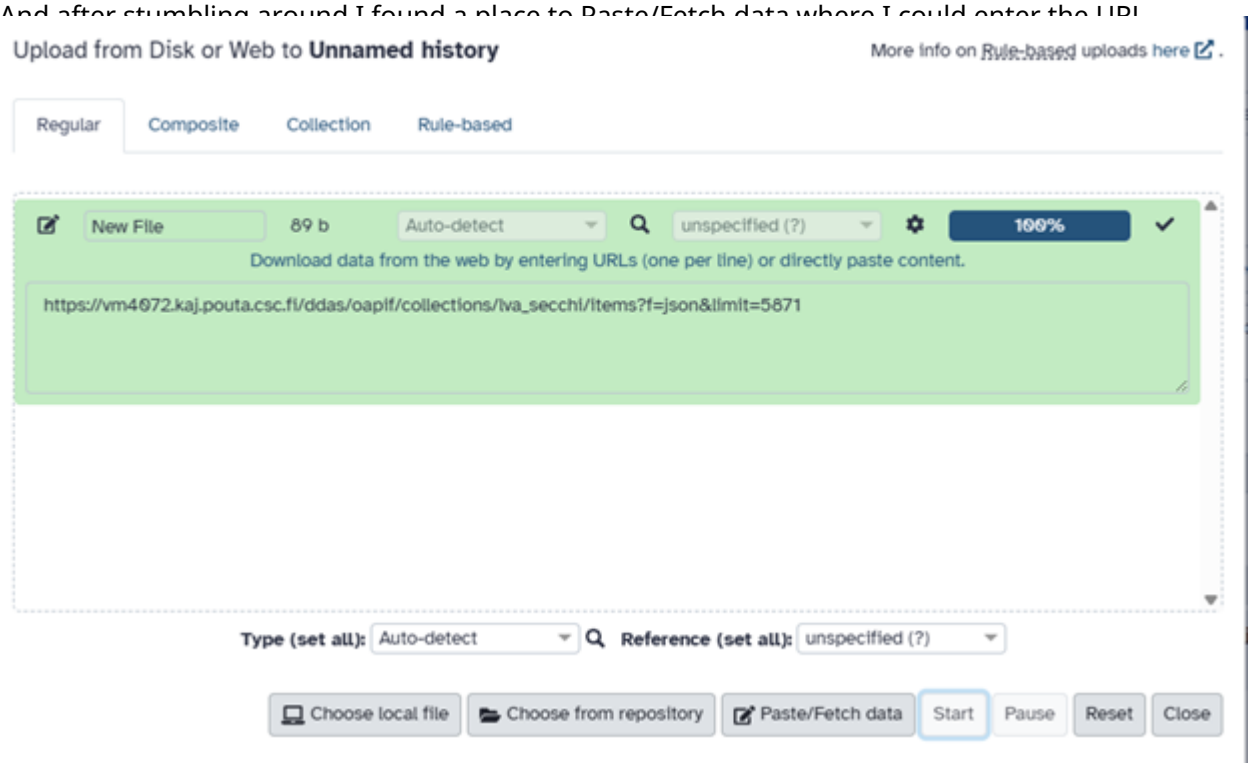
I clicked on this and clicked on Import to Galaxy. I set the limit data points to 5871 and clicked on the next Import to Galaxy button. Nothing happened. I then clicked Copy URL



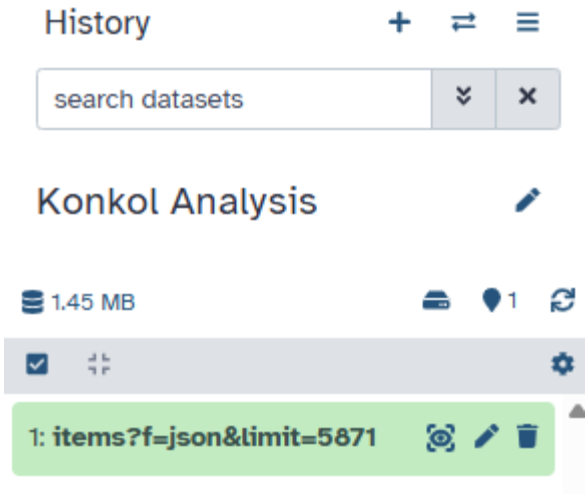
Then in Galaxy I clicked “Upload”



(noting the confusion related to terminology that the Upload button says Download from URL)
And after stumbling around I found a place to Paste/Fetch data where I could enter the URL

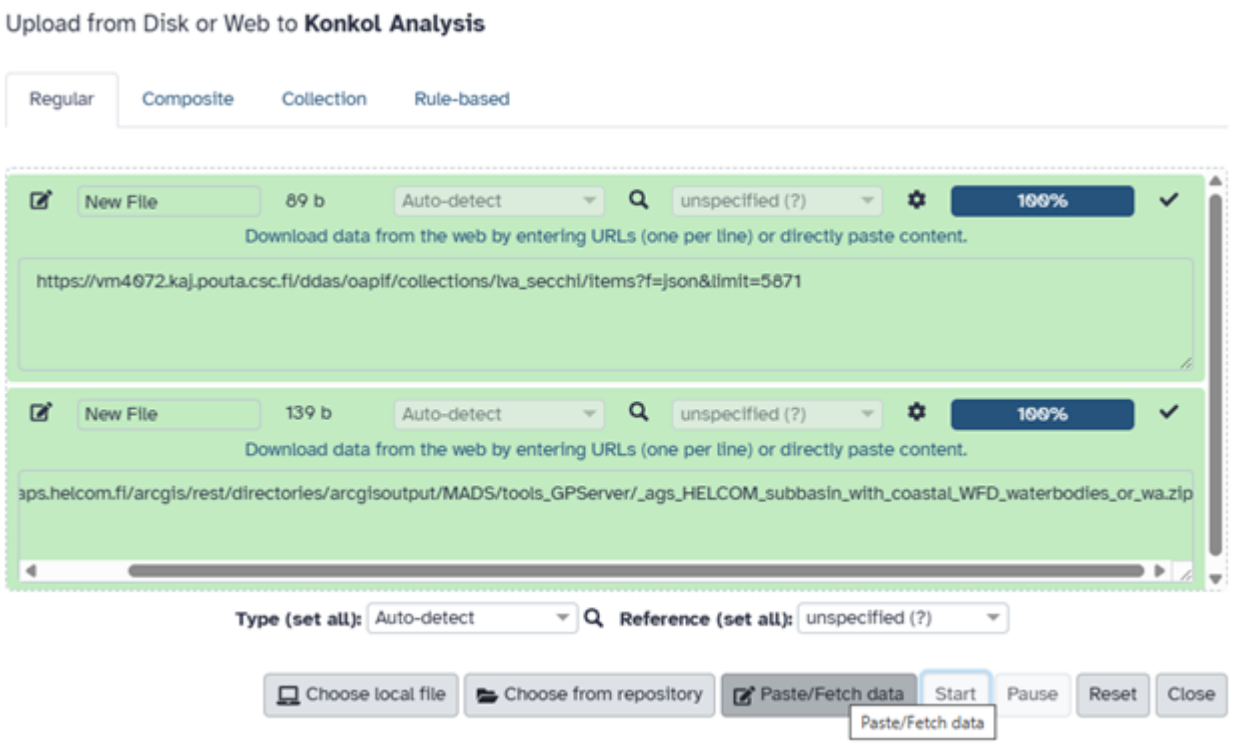


After clicking Start, this appeared to be successful and in Galaxy



Step 3: *Visit the input dataset "HELCOM subbasins with coastal WFD waterbodies or watertypes 2022 (level 4a)". Open the URL https://maps.helcom.fi/website/MADS/download/?id=67d653b1-aad1-4af4-920e-0683af3c4a48 in a new tab and accept the data usage disclaimer to activate the download link. Then go back to the AquaINFRA platform and paste the URL https://maps.helcom.fi/arcgis/rest/directories/arcgisoutput/MADS/tools_GPServer/_ags_HELCOM_subbasin_with_coast to the input field "Insert URL to a dataset" and then click on "Import to Galaxy". This will create a .txt file in your history on Galaxy including the URL to the data.*

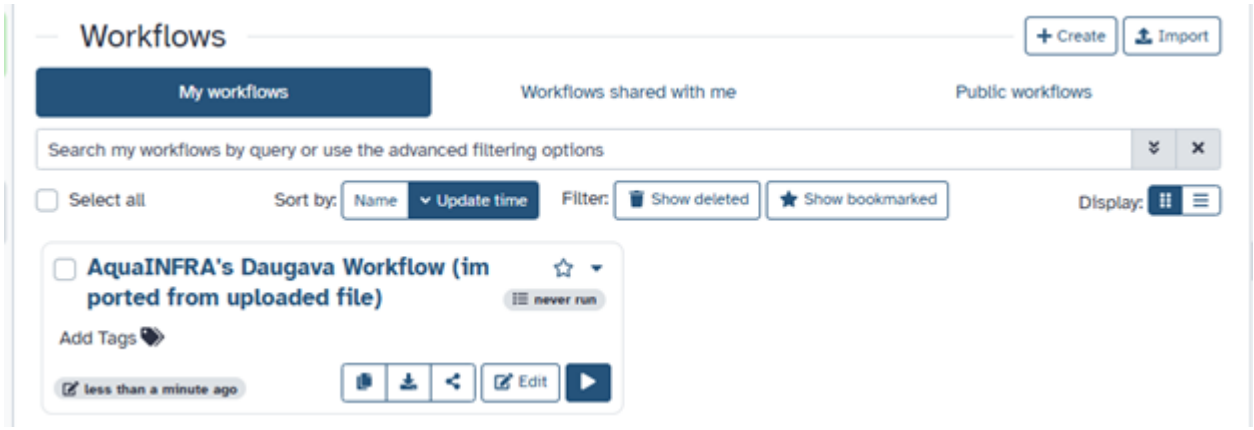
I used the URL in the Upload tool in Galaxy to load this dataset too.



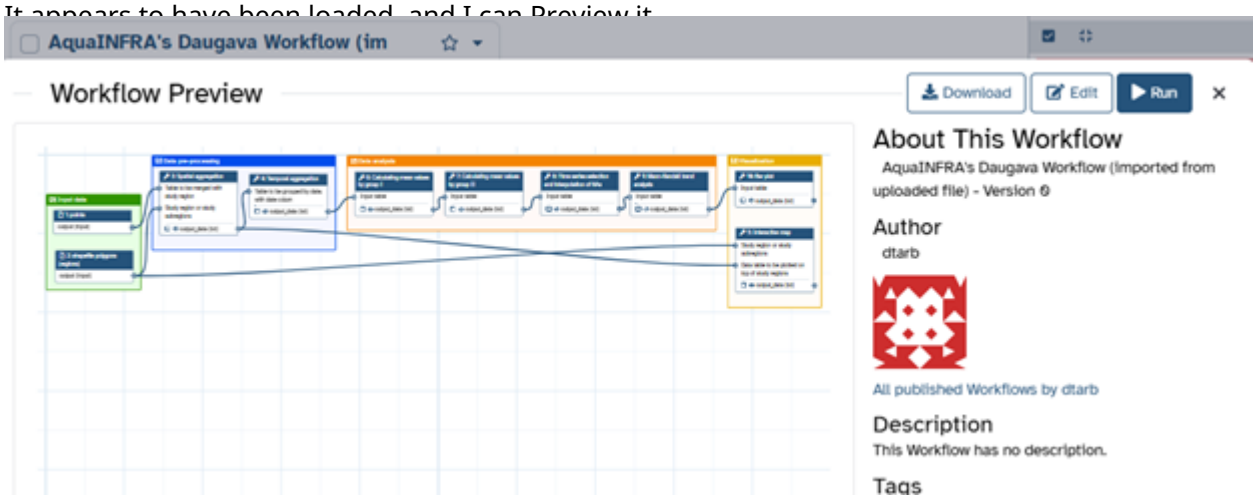
Step 4: *Click on the workflow and open it on Galaxy. Import the workflow and use the two .txt files created in Step 2 and Step 3 as inputs for the workflow.*
It was not clear where I was supposed to click on the workflow. I tried *Galaxy workflow: https://aqua.usegalaxy.eu/u/markus.konkol/w/copy-of-adj-aquainfras-daugava-workflow-shared-by-user-alabuse* but when I click on Run I get the error shown below and do not know what to do.



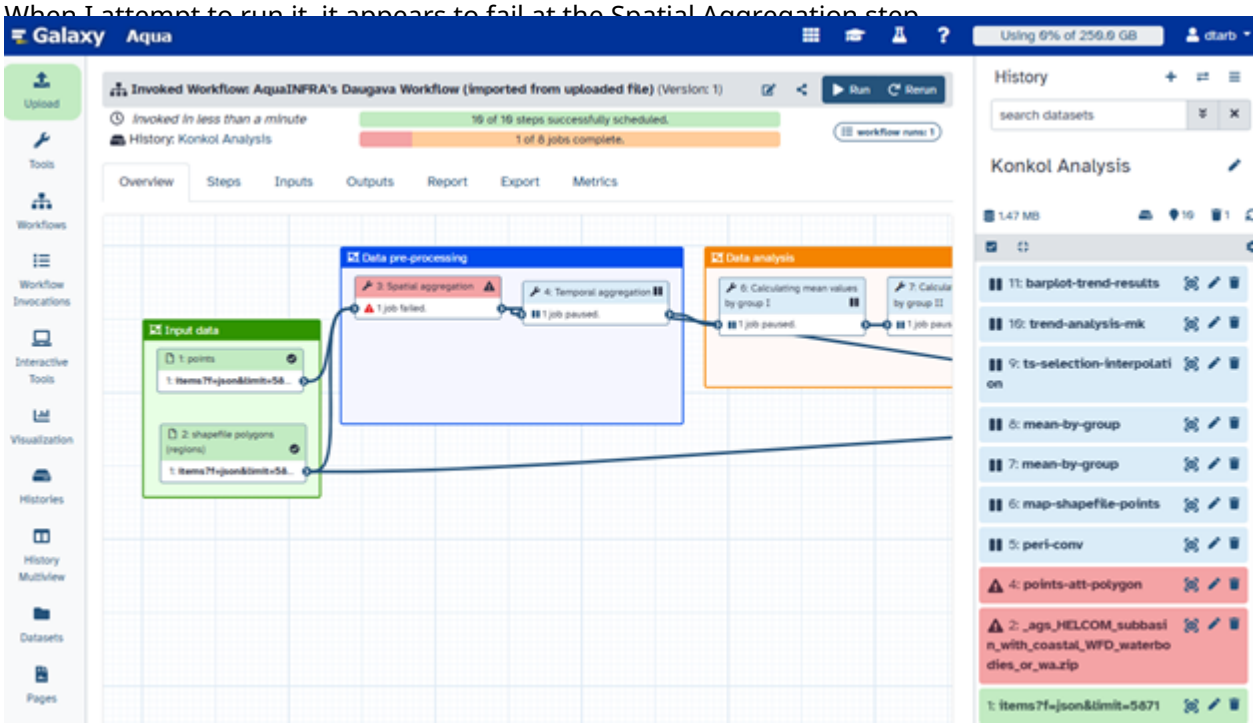
I also tried going to *Zenodo: Galaxy workflow. https://doi.org/10.5281/zenodo.15350861*. There I downloaded *Galaxy-Workflow-AquaINFRA_s_Daugava_Workflow.ga*, which I then uploaded from my local file.



It appears to have been loaded, and I can Preview it



When I attempt to run it, it appears to fail at the Spatial Aggregation step



I do not know what is wrong and do not know where to start looking to fix it.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
No

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
No

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Hydrology and Hydrologic Information Systems to support data sharing

We confirm that we have read this submission and believe that we have an appropriate level of expertise to state that we do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 18 Sep 2025

Markus Konkol

First of all, we would like to thank the two reviewers who jointly wrote the review. We are impressed by the effort they invested in reviewing the implementation of the proposed D2K-Package and the level of detail of the reproducibility report. First, the good news: We checked the functions and the workflow once more and they are working and generating the expected results. However, we need to reflect critically why it was not possible for the two reviewers to reproduce the results. The major obstacle we see was an unclear and shallow documentation, which caused confusion regarding what to do and how. For this reason, we revised the documentations to improve the comprehensibility. The analysis can be reproduced in several ways:

1. Using Galaxy and the AquaINFRA Interaction platform. For this, please visit the updated documentation in the repository hosting the D2K-Package, i.e., the ro-crate metadata file (https://doi.org/10.5281/zenodo.17143387). You might also want to check the video number 7 on the starting page of the AquaINFRA platform (https://aquainfra.dev.52north.org/) showing how this is working. However, some elements of the video are outdated and look different but it still shows the general

idea.

2. Using the OGC API Processes hosted on the pygeoapi server accessible via HTTP requests. For this, we added cURL commands to the README of the toolbox: https://github.com/AstraLabuce/aquainfra-usecase-Daugava/tree/containerize?tab=readme-ov-file#running-functions-via-curl-commands (https://doi.org/10.5281/zenodo.17143347)
3. Using Docker locally. To achieve that, we updated the Docker commands in the README of the toolbox so that they can simply be executed locally one after the other after cloning the repository and building the Docker image: https://github.com/AstraLabuce/aquainfra-usecase-Daugava/tree/containerize?tab=readme-ov-file#building-docker-image (https://doi.org/10.5281/zenodo.17143347)

Consequently, the analysis pipeline is reproducible even if Galaxy is not available in the future or if the AquaINFRA Interaction Platform is shut down. Since the functions can be executed locally using Docker, reproducibility also does not depend on the running pygeoapi server. We added a paragraph to clarify this issue, see end of the section "A hydrological use case". Next, we would like to address the concrete suggestions:

1. "The paper should show the key computational outputs and results in maps, figures, tables, etc., so that someone intending to reproduce or reuse this work knows what they are looking for, and so that the motivating question is answered."

Answer: We agree that this should be included in the paper. Therefore, we have included the final figure generated by the workflow in the paper (see Fig. 7) and added additional text to answer the research question (see end of the section "A hydrological use case"). In addition, we added the intermediate results to the toolbox folder "example_results" so that users can check after every step if the results are still the same (https://github.com/AstraLabuce/aquainfra-usecase-Daugava/tree/containerize/example_results).

1. "Simplify the technological requirements. It seems that the work of this paper could be done solely in an open-source procedural language such as Python or R."

Answer: The reproducible basis is done solely in R and users can run the Docker commands one after the other without using any other tools (see above). The other technologies (Galaxy, OGC API Processes, Binder) build on top of that basis and are intended to encourage reuse and reproducibility. Whether this goal was fully achieved may be open for discussion, but we believe this is the case.

1. "Make a well documented script that automates all steps of the workflow, including presentation of results"

Answer: We addressed this by providing detailed instructions to reproduce the analysis pipeline in three different ways (see above).

1. "Use a computational environment that will persist. For example, web platforms such as GALAXY may go away, change, or be inaccessible to the reproducer."

Answer: As explained above, the analysis does not depend on Galaxy, the AquaINFRA Interaction Platform, or the pygeoapi server. Users can run the functions locally using Docker. Nevertheless, in my opinion, we can use tools like Galaxy even if they might not be available in the future. What we need to ensure is that we're not locked in by a certain software as is the case in many commercial products. This is not the case when using the Galaxy platform. The workflow file (.ga) is an open data format and readable without Galaxy. So even if Galaxy will not exist anymore, users can still open the file and see the workflow

configuration including the steps and parameter settings. Based on this information, users can still create a workflow themselves using the OGC API Processes that are accessible via HTTP requests. Moreover, even if the OGC API Processes server is shut down, users find all materials on Zenodo (data, toolbox etc.) and can run the functions using Docker. Of course, this scenario comes with an increasing amount of effort but the reproducibility information is not lost. As explained in the section "Persistence", the reproducible basis is made persistent by using Zenodo for the data and the toolbox (incl. source code + computational environment).

1. "Provide guidance on more general approaches for the work to be reproduced."

Answer: Not sure what is meant by "more general approaches" but hope that is addressed by the instructions listed above.

1. "Provide instructions to recreate the run time environment, specifically for Docker and Binder. Explain how a person trying to reproduce the work should set up these tools on their computer, particularly on one or more common platforms (Windows/Mac/Linux)."

Answer: You have already recreated the computational environment by following the URL to MyBinder. This is exactly what Binder is there for. The instructions for Docker are in the README of the toolbox. I don't know what went wrong on your machine but making sure that Git, Docker etc. are running on your local computer is beyond the scope of this work. Moreover, running everything locally is possible but the actual goal of this work is to encourage reusability by providing workflows, web API servers, and virtual labs.

1. "Aggregate all the components needed to undertake this work in a single public repository with digital object identifier so that users can find everything in one place. Perhaps the data to knowledge package is this repository, but make this clear and lead a user on when they get to this."

Answer: Indeed, the ro-crate file in the D2K-Package is supposed to take this role. We added some text to clarify this issue in the section "The Concept of the Data-to-Knowledge Package". In the following, we would like to address some of the technical difficulties: "I followed the link on line 4 to the Binder. This is my first time using Binder. The link opened the following window. What do I do next?" Answer: We agree that first-time users might need a bit of time to use these tools but we also refer to the excellent documentation provided by Binder (https://mybinder.readthedocs.io/en/latest/introduction.html). Explaining the tools in the paper or the supplements is beyond the scope of this work. "I was unclear which file to open first." Answer: The idea is that users start with workflow as an entry point (see last bullet point in section "The concept of the Data-to-Knowledge Package"). The workflow indicates the order of the functions. "I selected all the code and ran it. That generated the following figure and error" Answer: The key benefit of Binder is that it provides a computational environment with all libraries and the runtime pre-installed. So, users who want to dive deep in the code have an environment where they can work. The scripts are designed such that they can be called from the outside with a command like R_SCRIPT "toolbox_function1.R" "URL to input dataset_1" "URL to input dataset_2" "parameter_1" "parameter_2" "output filename" Such a command is also what is passed into the Docker container to run the script. So it is indeed not possible just to select the code and run it. It requires some adjustments to the file paths and the parameters in code to run the scripts in the Binder environment. "Back in the README.md file that I downloaded from Zenodo onto my desktop, starting on line 6, I found the ordering of the

sections in the README.md file confusing. It seems the last section, “Create conda environment” needs to happen first?? Order the remaining sections in the order of the steps to reproduce the results.” Answer: This is indeed confusing and only needed for creating a local environment outside the Docker container. In fact, creating that environment is done within the Docker container and users working with the D2K-Package never need to get in touch with this, unless they want to dive really deep. We removed it from the README. “I got an error about being unable to run the git command even though I have Git Desktop installed on my machine that I regularly use.” Answer: It is difficult to judge from a distance what exactly the error is. It sounds like there is an issue with the local installation of git. However, for the correct installation of Git and refer you to the available online resources. “I clicked on this and clicked on Import to Galaxy. I set the limit data points to 5871 and clicked on the next Import to Galaxy button. Nothing happened.” Answer: I was not able to reproduce this error. I tried it with Firefox and Chrome and it worked. Maybe you can provide me further information on your browser. “Then in Galaxy I clicked “Upload” (noting the confusion related to terminology that the Upload button says Download from URL) And after stumbling around I found a place to Paste/Fetch data where I could enter the URL.” Answer: This step is not necessary to run the workflow presented in this paper. The workflow takes the two .txt files as input data, see description in https://doi.org/10.5281/zenodo.17143387. “When I attempt to run it, it appears to fail at the Spatial Aggregation step.” Answer: The problem was probably that you uploaded the data directly to Galaxy and used this as input to the tool. However, the tools used to run the workflow only work with URLs. If a tool fails, you can click on the tool (highlighted in red), click on the small bug. In one of the black boxes you will find an error message. “Shrinking research budgets are not in our opinion the leading motivator for reproducible research. The leading motivator is trust in the findings brought about by seeing and understanding the rationale for the findings.“ Answer: We added a sentence on trust in the introduction. “The paper states that a major obstacle is the “perception” that publishing reproducible results is time consuming. This is not a just perception, it is real.” Answer: We removed “perception”.

Competing Interests: No competing interests were disclosed.

Reviewer Report 08 August 2025

https://doi.org/10.21956/openreseurope.22545.r56467

© 2025 Lush V. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Victoria Lush
[1] Aston University, Birmingham, England, UK
[2] Aston University, Birmingham, England, UK

I would like to thank the authors for review responses. I have no further issues with the article.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Expertise include Human Computer Interaction, Data Analytics, Software Engineering, Citizen Science data, Decision Support systems.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 08 August 2025

https://doi.org/10.21956/openreseurope.22545.r56466

© 2025 Colom M. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Miguel Colom
[1] ENS Paris-Saclay, Université de Paris, Gif-sur-Yvette, France
[2] ENS Paris-Saclay, Université de Paris, Gif-sur-Yvette, France

In this revised version 2 of the article, the authors have given more importance to the reusability part of their work, rather than reproducibility. This is fair, since indeed from previous reviews it became clear that the proposed tool has a good potential mainly for reusability.

A brief discussion comparison D2K with existing platforms such as Code Ocean has been added, as well as an explanation why D2K is pertinent, despite the existence of virtualization tools suck as Docker.

Regarding PIDs, along with DOIs, now identifiers tailored specifically for software such as the SWHID are included in the discussion, as well as the integration with Zenodo.

In conclusion, the authors have address all my comments from the previous review, and therefore I think the article can be approved.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computational reproducible research

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 14 June 2025

https://doi.org/10.21956/openreseurope.21881.r53925

© **2025 Lush V.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Victoria Lush
[1] Aston University, Birmingham, England, UK
[2] Aston University, Birmingham, England, UK

This article presents a method designed to enable reproducible research and as such support re-use of published scientific research. The authors propose the Data-to-Knowledge Package (D2KPackage) method which packages together a collection of research materials including source code, data, virtual labs, web API services, and computational workflows. The authors also demonstrate a practical example of applying D2KPackage to a hydrological use case.
As a whole, the article is well written and is supported with visual materials. The limitations of the proposed method are discussed.
My main concern with the article is that the authors state that their method supports reproducibility of research, whereas D2KPackage method offers a potential solution for designing re-usable research studies. The title and some parts of the article refer to “re-usable research”, while other parts strongly suggest “reproducible research”. The authors should review their claims regarding D2KPackage support for reproducible research.
I also do not fully agree with a discussion that reproducible research means reusable research, while usually the case, it is not guaranteed. Reusability is not the only reason for a strong push towards reproducible research, it is also about trustworthiness of the results.
Since D2KPackage solution will not fit all research studies, it would be beneficial to clearly outline what are the prerequisites of applying this method. E.g., the authors already state that some data might not be openly available hence D2KPackage providers could crate own copy in a public repository, yet some datasets will exceed 100 files and 50GB Zenodo allowance (also duplication of data is not an ideal solution). A clear list of requirements would help potential users of D2KPackage solution to decide whether it fits their planned research study.
Regarding the diagrams, while the authors offer a generic D2KPackage method diagram (Figure 1), it would be beneficial to include a full overview of the technologies involved to produce the D2KPackage for the hydrological case study. This would help to better appreciate what resources are typically needed to adapt a research study to fit into D2KPackage framework.
In Figures 2 and 3 most text is cut off, the authors should use wrapped text to show content of the files.
I am not sure if Figure 8 really adds useful visual information to this paper. This is a generic research study lifecycle diagram.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Expertise include Human Computer Interaction, Data Analytics, Software Engineering, Citizen Science data, Decision Support systems.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 30 Jun 2025

Markus Konkol

We are grateful for the constructive criticism. The comments and recommendations helped us to enhance our paper considerably. In the following, we briefly outline how we addressed the raised issues. We hope that we have taken sufficient account of the comments. Don't hesitate to request further improvements.

"*My main concern with the article is that the authors state that their method supports reproducibility of research, whereas D2KPackage method offers a potential solution for designing re-usable research studies. [...] The authors should review their claims regarding D2KPackage support for reproducible research.*"

Answer: This is indeed an important issue. For this reason, we added the section "Reproducibility" to the discussion to elaborate on this aspect.

"*I also do not fully agree with a discussion that reproducible research means reusable research, while usually the case, it is not guaranteed.*"

Answer: We fully agree with this statement and it was not our intention to convey this message. We assume this impression comes from the sentence in the introduction

"*A key expectation is that reproducible research results are inherently more reusable by other*

researchers, reducing redundant efforts and accelerating the creation of new knowledge"? We added another sentence to clarify this: "...creation of new knowledge.

Nevertheless, someone trying to reuse someone else's code has to invest a lot of time to understand the code and incorporate parts of it into their own analysis."

"*Reusability is not the only reason for a strong push towards reproducible research, it is also about trustworthiness of the results.*"

Answer: That is correct. We added a sentence to the introduction: "Nevertheless, reproducible research should be valued not only for its intrinsic scientific merit, including verification and validation, but also for its wider benefits. **One of these advantages is that readers do not have to blindly trust the results in an article, but can check them themselves on the basis of the accompanying data and analysis.** A key expectation is that..."

"*Since D2KPackage solution will not fit all research studies, it would be beneficial to clearly outline what are the prerequisites of applying this method. E.g., the authors already state that some data might not be openly available hence D2KPackage providers could crate own copy in a public repository, yet some datasets will exceed 100 files and 50GB Zenodo allowance (also duplication of data is not an ideal solution). A clear list of requirements would help potential users of D2KPackage solution to decide whether it fits their planned research study.*"

Answer: We added a list of *prerequisites* to the section Discussion – Use case. In the meantime, we also created a user guide with concrete steps to develop a D2K-Package as a living document, meaning that we further improve it based on user feedback.

"*Regarding the diagrams, while the authors offer a generic D2KPackage method diagram (Figure 1), it would be beneficial to include a full overview of the technologies involved to produce the D2KPackage for the hydrological case study.*"

Answer: We added a list of all technologies used to implement the hydrological use case at the end of the chapter "Results". We have refrained from creating another figure analogous to Figure 1, as this would have led to overloading.

"*In Figures 2 and 3 most text is cut off, the authors should use wrapped text to show content of the files.*"

Answer: Showing the whole content of the file as wrapped text is probably too much as it contains more than 500 lines. How about adding a link to the file to the caption of the Figures 2-4?

"*I am not sure if Figure 8 really adds useful visual information to this paper. This is a generic research study lifecycle diagram*."

Answer: We removed the figure from the article.

Competing Interests: No competing interests were disclosed.

Author Response 18 Jul 2025

Markus Konkol

We are grateful for the constructive criticism. The comments and recommendations helped us to enhance our paper considerably. In the following, we briefly outline how we addressed the raised issues. We hope that we have taken sufficient account of the comments. Don't hesitate to request further improvements. "*My main concern with the article is that the authors state that their method supports reproducibility of research, whereas D2KPackage method offers a potential solution for designing re-usable research studies. [...] The authors should review their claims regarding D2KPackage support for reproducible research.*" **Answer:** This is indeed an important issue. For this reason, we added the section "Reproducibility" to the discussion to elaborate on this aspect. "*I also do not fully agree with a discussion that reproducible research means reusable research, while usually the case, it is not guaranteed.*" **Answer:** We fully agree with this statement and it was not our intention to convey this message. We assume this impression comes from the sentence in the introduction "*A key expectation is that reproducible research results are inherently more reusable by other researchers, reducing redundant efforts and accelerating the creation of new knowledge*"? We added another sentence to clarify this: "...creation of new knowledge. **Nevertheless, someone trying to reuse someone else's code has to invest a lot of time to understand the code and incorporate parts of it into their own analysis**." " *Reusability is not the only reason for a strong push towards reproducible research, it is also about trustworthiness of the results.*" **Answer:** That is correct. We added a sentence to the introduction: "Nevertheless, reproducible research should be valued not only for its intrinsic scientific merit, including verification and validation, but also for its wider benefits. **One of these advantages is that readers do not have to blindly trust the results in an article, but can check them themselves on the basis of the accompanying data and analysis.** A key expectation is that..." "*Since D2KPackage solution will not fit all research studies, it would be beneficial to clearly outline what are the prerequisites of applying this method. E.g., the authors already state that some data might not be openly available hence D2KPackage providers could crate own copy in a public repository, yet some datasets will exceed 100 files and 50GB Zenodo allowance (also duplication of data is not an ideal solution). A clear list of requirements would help potential users of D2KPackage solution to decide whether it fits their planned research study.*" **Answer:** We added a list of *prerequisites* to the section Discussion – Use case. In the meantime, we also created a user guide with concrete steps to develop a D2K-Package as a living document, meaning that we further improve it based on user feedback. "*Regarding the diagrams, while the authors offer a generic D2KPackage method diagram (Figure 1), it would be beneficial to include a full overview of the technologies involved to produce the D2KPackage for the hydrological case study.*" **Answer:** We added a list of all technologies used to implement the hydrological use case at the end of the chapter "Results". We have refrained from creating another figure analogous to Figure 1, as this would have led to overloading. "*In Figures 2 and 3 most text is cut off, the authors should use wrapped text to show content of the files.*" **Answer:** Showing the whole content of the file as wrapped text is probably too much as it contains more than 500 lines. How about adding a link to the file to

the caption of the Figures 2-4? *"I am not sure if Figure 8 really adds useful visual information to this paper. This is a generic research study lifecycle diagram."* **Answer:** We removed the figure from the article.

Competing Interests: No competing interests were disclosed.

Reviewer Report 03 June 2025

https://doi.org/10.21956/openreseurope.21881.r53927

© 2025 Colom M. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Miguel Colom
[1] ENS Paris-Saclay, Université de Paris, Gif-sur-Yvette, France
[2] ENS Paris-Saclay, Université de Paris, Gif-sur-Yvette, France

The submitted article proposes a the Data-to-Knowledge Package ("D2K-Package") which aims to facilitate reusability of computational methods, focusing on reproducible research. To this purpose, the package integrates several components, including virtual labs (such as MyBinder), API services, FAIR data, and workflows.
The aim of the project is to facilitate reusability, duplication of efforts, and a easier peer review in scientific publications. A concrete use case, the hydrological dynamics of the Gulf of Riga, is presented.

The work is clear and well written, but there are some points I think it's worth to for the authors to explain:

In the article, reproducible research is mentioned in different parts, but however it seems that the main contribution of this article is actually reusability, as clearly highlighted in the title.

The presented work is useful as a tool for reusability as it gathers data and code together, and allows for the creation of the virtual lab, workshops, of API services, as shown in Figure 1, but I have concerns about its potential for reproducibility. For example, it should contain automatic tests to ensure that the experiments always give the same results, or have a mandatory peer-review to ensure that the pseudocodes corresponds to the implementation, among other minimal requirements. I would, therefore, suggest the authors to relax the claims on reproducibility, since although the proposed tool has potential for it, it remains far from this objective.

From the point of view of a reusability tool, the article lacks of comparisons with reference techniques and tools. For example: how does it compare to a Docker image which integrates an execution environment and provides its own API services? Is D2K more convenient because the "research products" in Fig. 1 would be external to the image?
How D2K compares to a Docker image using a Jupyter notebook or any other visualization tool,

with respect to using the VirtualLab? How D2K compares to existing solutions such as Code Ocean and others?

The authors mention the use of PIDs, but a DOI is not enough for code (although the accepted solution for indexing). I'd suggest to explore other more suitable PIDs, such as the SWHID, among others.

My conclusion is that, even if the proposed work has some potential for a reusability work, the reasons why it would facilitate reproducibility are not well justified in this version of the article. Given that it also lacks a proper comparison with other alternatives to reuse code, I think it needs to be completed following these recommendations, before being indexed as a method article, or to be proposed as a general solution to the community. I strongly recommend the authors to improve this word and prepare a new revision.

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Partly

Are sufficient details provided to allow replication of the method development and its use by others?
Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
No source data required

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computation reproducible research

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 30 Jun 2025

Markus Konkol

We are grateful for the constructive criticism. The comments and recommendations helped us to enhance our paper considerably. In the following, we briefly outline how we addressed the raised issues. We hope that we have taken sufficient account of the

comments. Don't hesitate to request further improvements. *"I have concerns about its potential for reproducibility. For example, it should contain automatic tests to ensure that the experiments always give the same results, or have a mandatory peer-review to ensure that the pseudocodes correspond to the implementation, among other minimal requirements. I would, therefore, suggest the authors to relax the claims on reproducibility, since although the proposed tool has potential for it, it remains far from this objective."*

Answer: We fully agree with this comment and have therefore added a further section "Reproducibility" to the discussion, in which this aspect is highlighted.

"*From the point of view of a reusability tool, the article lacks of comparisons with reference techniques and tools. For example: how does it compare to a Docker image which integrates an execution environment and provides its own API services? Is D2K more convenient because the "research products" in Fig. 1 would be external to the image? How D2K compares to a Docker image using a Jupyter notebook or any other visualization tool, with respect to using the VirtualLab? How D2K compares to existing solutions such as Code Ocean and others?*"

Answer: This is indeed a crucial gap. We have therefore added a corresponding section at the beginning of the discussion.

"*The authors mention the use of PIDs, but a DOI is not enough for code (although the accepted solution for indexing). I'd suggest to explore other more suitable PIDs, such as the SWHID, among others.*"

Answer: During our research on PIDs for source code and SWHID we found out that fortunately there is already an integration between Zenodo and Software Heritage. We have explained this aspect in more detail in the section "Persistence".

Competing Interests: No competing interests were disclosed.

Author Response 18 Jul 2025

Markus Konkol

We are grateful for the constructive criticism. The comments and recommendations helped us to enhance our paper considerably. In the following, we briefly outline how we addressed the raised issues. We hope that we have taken sufficient account of the comments. Don't hesitate to request further improvements. *"I have concerns about its potential for reproducibility. For example, it should contain automatic tests to ensure that the experiments always give the same results, or have a mandatory peer-review to ensure that the pseudocodes correspond to the implementation, among other minimal requirements. I would, therefore, suggest the authors to relax the claims on reproducibility, since although the proposed tool has potential for it, it remains far from this objective."* **Answer:** We fully agree with this comment and have therefore added a further section "Reproducibility" to the discussion, in which this aspect is highlighted. *"From the point of view of a reusability tool, the article lacks of comparisons with reference techniques and tools. For example: how does it compare to a Docker image which integrates an execution environment and provides its own API services? Is*

D2K more convenient because the "research products" in Fig. 1 would be external to the image? How D2K compares to a Docker image using a Jupyter notebook or any other visualization tool, with respect to using the VirtualLab? How D2K compares to existing solutions such as Code Ocean and others?" **Answer:** This is indeed a crucial gap. We have therefore added a corresponding section at the beginning of the discussion. "*The authors mention the use of PIDs, but a DOI is not enough for code (although the accepted solution for indexing). I'd suggest to explore other more suitable PIDs, such as the SWHID, among others.*" **Answer:** During our research on PIDs for source code and SWHID we found out that fortunately there is already an integration between Zenodo and Software Heritage. We have explained this aspect in more detail in the section "Persistence".

Competing Interests: No competing interests were disclosed.