

Table of Contents

Table of Contents	1
Opening Reproducible Research: research project website and blog	4
Blog posts	5
OJS plugin for enabling ERC journal submissions	5
o2r Workshop on Reproducible Remote Sensing	6
Talks, preprints, papers, and a leave of absence	7
Talks	7
Preprints, abstracts, reports & postprints	7
Papers	7
New team member	8
Daniel's Ph.D. Defense	9
geoextent presented at the 2021 EarthCube annual meeting	10
geoextent presented at the 2021 EarthCube	10
Exploring research data repositories with geoextent	10
EarthCube meeting 2021	11
New preprint on o2r architecture and implementation	12
Abstract	13
Discuss and share	13
Bachelor Thesis of student assistants Tom and Nick	15
Geospatial Metadata for Discovery in Scholarly Publishing	15
Combining Augmented Reality and Reproducibility to convey spatio-temporal results	17
New papers out about 'Practical Reproducibility in Geo' and the 'Rockerverse'	19
Practical Reproducibility in Geography and Geosciences	19
The Rockerverse: Packages and Applications for Containerisation with R	19
Beyond o2r: collaborations and community activity for more open and reproducible science	20
1. Citable and preserved AGILE short papers	20
2. CODECHECK	21
Low tech, community work, and technological advances go hand in hand in Opening Reproducible Research.	21
o2r student assistant about impressions of reproducibility ready to start a career in research	22
Introducing geoextent	23
Motivation	24
Origins	24
Process of creating the codebase	24
Current features	24
Next steps	25
Next generation journal publishing and containers	26
WWU workshop on Reproducible Research	27
o2r2 project proposal publiciation	28
o2r @ ECMWF Workshop in Reading, GB	30
Opening Reproducible Research with OJS	31
Procedure	31
User stories	31
ERC plug-in for OJS	32
Upload Executable Research Compendium	32
Review Executable Research Compendium	33
Examine Executable Research Compendium	35
Plug-in structure	38
Conclusion	39
Markus Konkol defends PhD Thesis	40
o2r on tour: eLife Sprint and JupyterHub/Binder workshop	41
Why PDFs are not suitable for communicating (geo)scientific results	43
4+1 quick incentives of open reproducible research	45
Discovery	45
Inspection	45
Manipulation	45
Substitution	45
+1	45
Reproducible Research and Geospatial Badges at AGILE 2019 conference in Limassol	47
o2r2 @ Conquaire Workshop	49
o2r2 - Putting ERC into practice	51
Archiving a Research Project Website on Zenodo	53
R&R Workshop at SPARC	55
How to increase reproducibility and transparency in your research	57
Reproducible research manuscripts	58
Sustainable access to supplemental data	60
Main takeaways	60
Acknowledgements	61
References	61
New article published in International Journal of Geographical Information Science	63
eLife sprint: Integrating Stencila and Binder	64
The idea and the sprint team	64
The building blocks	64
The challenge	64
The solution	65
Sprint breakthrough	65
Connecting Stencila to Jupyter kernels	65
Consolidation	67
The last mile	68

Summary and outlook	70
Demo server update	72
New article published in PeerJ	74
AGILE 2018 pre-conference workshop report	75
Report from EGU 2018	76
Digitisation of Science @ WWU	79
Open environmental data analysis	81
Events in 2018: Call for participation	84
Open Science, R, and FOSS sessions at EGU General Assembly 2018	84
Short course on reproducible papers at EGU General Assembly 2018	84
Reproducible Research Publications at AGILE 2018	85
Reference Implementation - Try it out!	86
tl;dr	87
Reproducible Research Badges	88
Introduction	88
Overview of badges for research	88
An independent API for research badges	90
Spread badges over the web	91
Outlook: Action integrations	93
Discussion	94
Future Work	94
References	95
useR!2017	96
C4RR workshop in Cambridge	98
Generating Dockerfiles for reproducible research with R	100
1. Introduction	100
2. Creating a Dockerfile	100
2.1 Basics	100
2.2 Packaging an interactive session	101
2.3 Packaging an external session	103
2.4 Packaging an R script	103
2.5 Packaging an R Markdown file	104
2.6 Packaging a workspace directory	104
3. Including resources	104
4. Image metadata	106
5. Further customization	107
6. CLI	107
7. Challenges	108
8. Conclusions and future work	108
Metadata	109
State of the project and next steps	110
Opening Reproducible Research at AGILE 2017 conference in Wageningen	112
Opening Reproducible Research at EGU General Assembly 2017	113
Reproducible Research at EGU GA - A short course recap	114
Docker for GEOBIA - new article published	116
o2r @ Open Science Conference 2017, Berlin	117
EGU short course scheduled and session programme upcoming	118
D-Lib Magazine Article Published	119
Reproducible Computational Geosciences Workshop at AGILE Conference	120
Investigating Docker and R	121
Dockerising R	121
Rocker	121
Bioconductor	121
CentOS-based R containers	121
MRO	122
Renjin	122
pqR	122
[WIP] FastR	122
Dockerising Research and Development Environments	122
Running Tests	123
Dockerising Documents and Workflows	123
Control Docker Containers from R	124
R and Docker for Complex Web Applications	124
Batch processing	124
"Reproducible research for big data in practice": call for abstracts EGU GA 2017 session	126
Open in Action	127
Workshop on Reproducible Open Science	128
Docker presentation at FOSS4G conference	129
Summer break technical post: ORCID OAuth with passport.js	131
Feedback on and Focus for the o2r Vision	132
Container Strategies for Data & Software Preservation that Promote Open Science (DASPOS workshop)	134
Looking back at EGU General Assembly	135
Join our first survey	136
Opening Reproducible Research at EGU General Assembly 2016	137
Introducing o2r	138
Website pages	139
About	139
Goal	139
Why Open Access?	139
Why Open Source?	139
The team	139
Former members	140
External partners	140
Funding	141

	Documents	142
	Pilots	143
⚙ Deployment		144
License		145



Opening Reproducible Research: research project website and blog

Blog post authors:

- Daniel Nüst
- Eftychia Koukouraki
- Jan Koppe
- Laura Goulier
- Lukas Lohoff
- Marc Schutzeichel
- Markus Konkol
- Matthias Hinz
- Nick Jakuschona
- Rémi Rampin
- Sebastian Garzón
- Tom Niers
- Vicky Steeves
- Yousef Qamaz

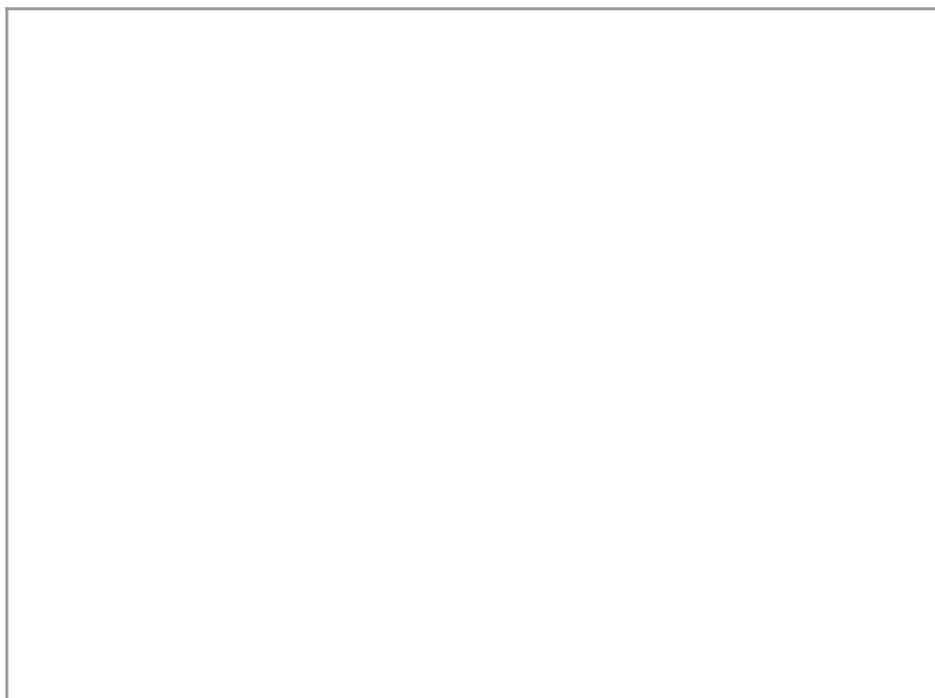
Blog posts

OJS plugin for enabling ERC journal submissions

26 Aug 2022 | By Eftychia Koukouraki

Our self-hosted pilot has new implemented functionalities! Developed by the o2r team, the *ojs-erc-plugin* uses the plugin mechanism provided by the [Open Journal System \(OJS\)](#) to create a completely ERC-based publishing workflow including submission, review, and publication. In the OJS submission process, we created the option to upload an ERC workspace via the [o2r service](#). After the submission, the assigned reviewer(s) can examine three components which are related to the (submitted) ERC: 1) the zipped ERC workspace which was uploaded by the author, 2) the html view of the ERC, and 3) the ERC Galley which shows the ERC in the usual o2r-service view. After the acceptance and publication of the submission, the reader of the journal can access all the abovementioned components of the (published) ERC and experience the interactive features that are enabled by ERCs.

The functionality of the *ojs-erc-plugin* is showcased in the following [video](#).



The *ojs-erc-plugin* is an Open Source project and is welcoming contributions and active participation in our [Github page](#)!

o2r Workshop on Reproducible Remote Sensing

08 Apr 2022 | By Eftychia Koukouraki

Just a few days ago, on March 23rd 2022, we had the pleasure of hosting a workshop for “Reproducible Remote Sensing: Challenges and Evolution Possibilities”. It was so exciting to get together again, even in a hybrid format, to discuss such an interesting and interdisciplinary topic! As long as the o2r project has been running, we have approached the matter of reproducibility of scientific results from several perspectives, and now it was the time to frame the complexities of reproducibility in Remote Sensing. With this workshop, we wanted to explore the current and future possibilities of publishing reproducible research in Remote Sensing as a subdomain of Geoinformatics and for this purpose, a group of 17 people involved in different positions of the Academia was assembled, with everyone contributing their point of view to the discussion.

The workshop consisted of 2 breakout sessions. In the first part, we talked about our thoughts, the obstacles we encounter and the wishes we have regarding scientific knowledge in Remote Sensing that works reproducibly. In the second part, the discussion was mainly focused on the idea of creating an independent Journal for publishing Reproducible Remote Sensing papers, as a means of establishing reproducibility in this scientific field. The participants shared their experiences with regards to the state of reproducibility in their own work, they commented about the usability of o2r developed platform (<https://o2r.uni-muenster.de>) and outlined the positive aspects and the potential pitfalls of launching a fully reproducible Journal on Remote Sensing. After every session, the breakout groups presented their discussed points to the rest.

There were many interesting points raised throughout the whole workshop. The data appeared to be a tricky issue, as the sensitivity, the licensing and the size are parameters that very frequently do not allow for an end-to-end reproducible pipeline. The o2r system was characterized as a nice solution for early career scientists, because of its simplicity, and as an interesting learning tool. In the o2r team, we felt that reproducibility in Remote Sensing is so engaging to a big part of the community, that we could be deliberating for quite some time! We deeply thank all the participants and we are looking forward to future collaborations!

Talks, preprints, papers, and a leave of absence

05 Apr 2022 | By Daniel Nüst

This blog has been a bit quieter than usual in the year 2021, not because of the thing that messes everything else up in the world, but because o2r team member Daniel has been focussing on wrapping up his PhD thesis ([it's done!](#)) and took a leave of absence. Most of the activities below have been shared via the project's Twitter feed, but we'd like to catch up the blog, too, little announcement at the end. So with no further ado, here are events and publications of the last 8+ months:

Talks

o2r team members were given several opportunities to present work stemming from the project at conferences and workshops.

- Sebastian presented the tool [geoextent](#) in the submission **Exploring Research Data Repositories with geoextent** at *EarthCube 2021*, see [archived notebook](#) and the [workshop proceedings](#).

It was great to participate in a scientific event that invited submissions of computational notebooks as the item of publication.

- Daniel presented the CODECHECK and Reproducible AGILE initiatives as well as the o2r approach to sharing computational workflows at several events with, admittedly, similar titles:
 - **Code execution during peer review** at *Collaborations Workshop 2022*, see [slides](#) and [recording](#)
 - **Code Execution and Code Review in the Scholarly Communication Process** at *Mannheim Open Science Meetup* on 2021-07-28, see [slides](#)
 - **Executing workflows during peer review for transparency, reproducibility, and reusability** at *Thüringer FDM-Tage 2021*, organised by Thüringer Kompetenznetzwerk Forschungsdatenmanagement on 2021-06-23, see <https://doi.org/10.5281/zenodo.5006378> and [slides](#)
- Daniel gave a talk on **Practical reproducibility and reproducibility vs. peer review** at *Spatial Data Science Hangout* organised by the Center for Spatial Studies at UCSB on 2021-05-20, see [slides](#)

Preprints, abstracts, reports & postprints

A chapter of Daniel's PhD thesis was published as a preprint:

Nüst, Daniel. 2021. **A Web service for executable research compendia enables reproducible publications and transparent reviews in geospatial sciences**. Zenodo. doi: [10.5281/zenodo.4818119](https://doi.org/10.5281/zenodo.4818119)

Daniel had the amazing opportunity to contribute to a *Knowledge Exchange* activity on *Publishing Reproducible Research Outputs*. Even though he is not an author, we want to point out the excellent reports from the group:

The Art of Publishing Reproducible Research Outputs: Supporting emerging practices through cultural and technological innovation. Chiarelli A, Loffreda, Lucia, Johnson R. 2021. Zenodo. <https://doi.org/10.5281/zenodo.5521076>

See also the [executive summary](#) if you are in a hurry.

As a continuation of the activities in Reproducible AGILE, Daniel contributed to a conference abstract or **Reproducible graduate theses in GIScience** with Carlos Granell and Rusne Sileryte, which was presented at *Research Reproducibility 2020*, University of Florida, on 2020-12-02, <https://doi.org/10.17605/OSF.IO/WCEXY>, see also [slides](#).

And finally, the embargo period of an *anon-OA article* by Daniel ended and he deposited the *AAM* as a postprint on EarthArXiv:

Practical Reproducibility in Geography and Geosciences Nüst D, Pebesma E. 2022. EarthArXiv. <https://doi.org/10.31223/X5XH0Z>

Papers

The assessment of the reproducibility of GIScience conference papers was continued for the GIScience conference series, and former team member Markus was also on board again. We are very happy that this paper won the Best Paper award at the GIScience conference 2021!

Reproducible Research and GIScience: An Evaluation Using GIScience Conference Papers Ostermann FO, Nüst D,

Granell C, Hofer B, Konkol M. 2021. <https://doi.org/10.4230/LIPIcs.GIScience.2021.II.2>; preprint

Daniel presents a non-technology driven approach to reproducibility in this collaboration with Stephen Eglen:

CODECHECK: an Open Science initiative for the independent execution of computations underlying research articles during peer review to improve reproducibility. Nüst D and Eglen SJ. [version 2; peer review: 2 approved]. F1000Research 2021, 10:253. <https://doi.org/10.12688/f1000research.51738.2>; preprint

The work was also featured on the F1000 blog [here](#).

Daniel contributed to the following review article on metadata - thanks Jeremy for an outstanding job and the opportunity to take part!

The role of metadata in reproducible computational research Leipzig J, Nüst D, Hoyt CT, Ram K, Greenberg J Article(journal). Patterns. 2021. <https://doi.org/10.1016/j.patter.2021.100322>; preprint

What if the o2r system is not flexible enough but you want to use containers to packaeg your data science workflow?

Ten simple rules for writing Dockerfiles for reproducible data science Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T, Evans BD. PLOS Computational Biology. 2020. <https://doi.org/10.1371/journal.pcbi.1008316>

New team member

The o2r project is excited to have found a new team member to continue making research more reproducible, and to take some new directions, too. We welcome [Eftychia Koukouraki](#) to the team! She joins us after graduating from the Erasmus Mundus Masters in Geospatial Technologies as a member of the SITCOM lab. Before joining the exciting field of Geoinformatics, she worked in the industry as a Software Engineer and she also holds a Diploma in Electrical and Computer Engineering.

Great to have you on board, Eftychia - looking forward to see where you take the project!

Daniel's Ph.D. Defense

14 Feb 2022 | By Markus Konkol

The o2r project started with quite a bold statement. I think it was Daniel's supervisor Edzer Pebesma who stated that "We want to change the scientific landscape". Well, yeah, let's try it - I thought. How could I know that this guy sitting next to me behind his Linux computer would indeed achieve such a change? He became CODECHECK's co-inventor and AGILE's reproducibility king. While I am not sure whether the latter is an official title, his contributions certainly have a real impact. His backend developments are the backbone of the success of the o2r project, though just one of the many publications, which not only comprise scientific papers but also software. Let's not forget that he successfully included a whale that swallows R's and poops containers in his defence!

`containerit`

Congratulations to Dr. rer. nat. Daniel Nüst for a great Ph.D. defence "*Infrastructures and Practices for Reproducible Research in Geography, Geosciences, and GIScience*"!



Read Daniel's work:

Nüst, Daniel. 2021. **Infrastructures and Practices for Reproducible Research in Geography, Geosciences, and GIScience**. Doctoral dissertation, University of Münster, Germany. <https://doi.org/10.5281/zenodo.4768096>.

geoextent presented at the 2021 EarthCube annual meeting

03 Aug 2021 | By Sebastian Garzón

The Python library `geoextent` by the o2r project team was selected for presentation at the 2021 EarthCube annual meeting in the peer-reviewed notebooks session. In this blog post, student assistant Sebastian reports from the event.

geoextent presented at the 2021 EarthCube

Notebooks as a scholarly object, database interoperability, FAIR workflows, connecting data and code, and tools for geosciences research are some of the topics discussed at the 2021 EarthCube annual meeting. At the event, o2r team members Sebastian and Daniel presented `geoextent`, a Python library designed to extract temporal and spatial extent from data files.

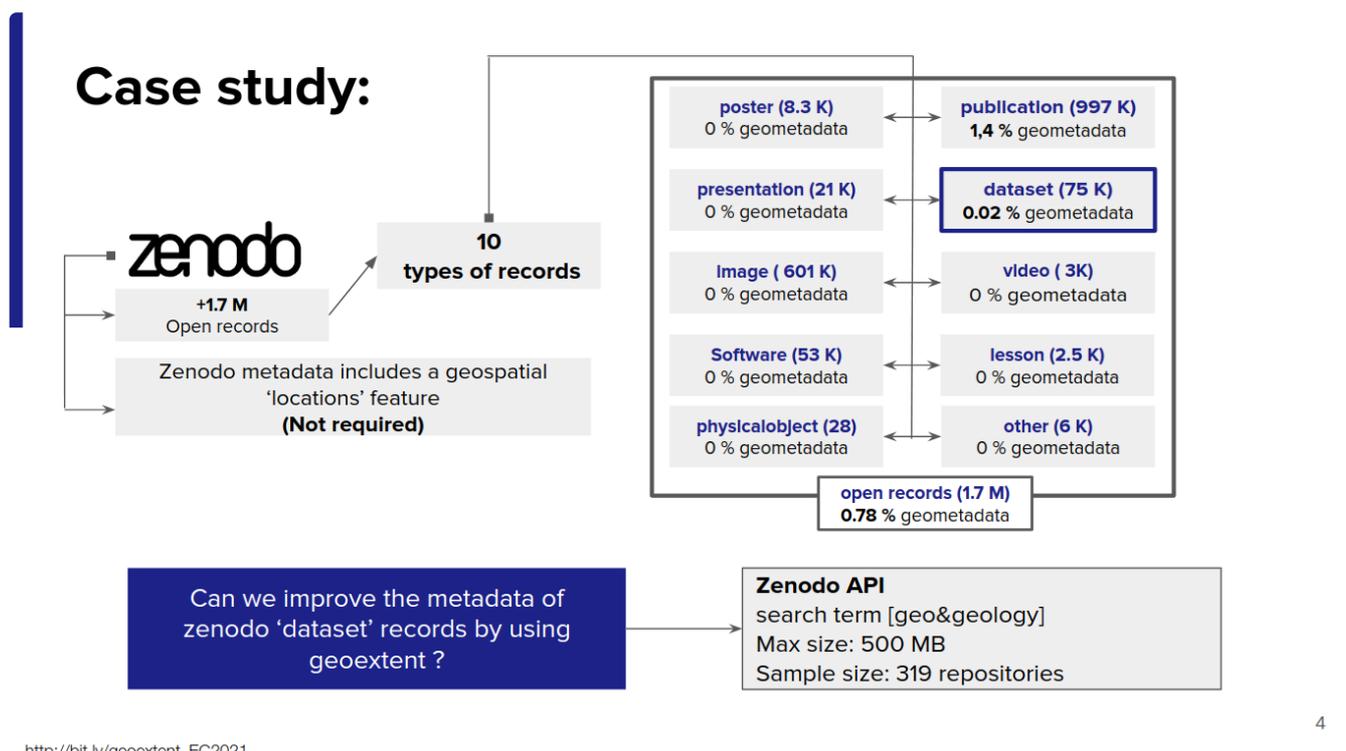
We presented the library as part of the 2nd call for Notebooks for a digital proceedings of the EarthCube annual meeting following the increased interest of the geosciences research community on reproducible workflows.

Exploring research data repositories with geoextent

Sebastian Garzón and Nüst, Daniel. 2021. **Exploring Research Data Repositories with geoextent**. EarthCube annual meeting. <https://doi.org/10.5281/zenodo.5496311>

launch binder

The notebook, accessible through Binder, includes an introduction of `geoextent`'s usage and a case study where we explored more than 300 Zenodo repositories (over 25,000 files!) with `geoextent`. An initial exploration of Zenodo's API showed that spatial metadata is rarely available, difficulting data integration and discovery. The objective of our case study was to verify if we can increase the current percentage of repositories with geospatial information on Zenodo by using `geoextent`.



4

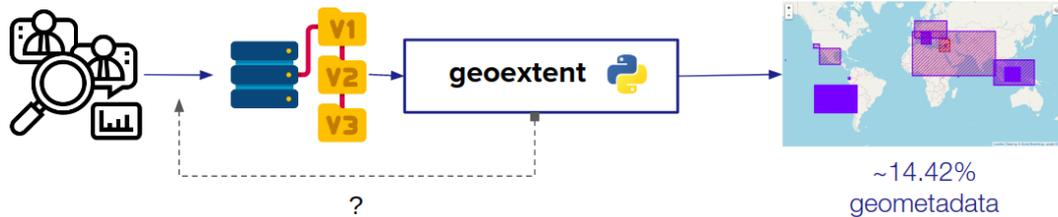
http://bit.ly/geoextent_EC2021

Screenshot of presentation showing the current state of spatial metadata in Zenodo

Our results suggest that `geoextent` could be used to increase spatial metadata of repositories by directly extracting information from the files deposited on them. However, we identified a series of challenges for this approach including geospatial information being stored in ambiguous formats (e.g., CSV and `.asc` files) or incorrectly georeferenced files in specialized formats (e.g., missing coordinate reference system or flipped coordinates). This case study also provides information for further development of `geoextent` to support more file formats and fix.

Conclusions / ideas

Research records contain files with potential geospatial information 51.4% of zenodo records (n=319) contain geotiffs formats	Ambiguous file formats are being used to store geospatial information. asc csv pdf png	Usage of specialized geospatial formats improve the preservation of information but do not completely guarantee it. NetCDF: 6.6% ext. success rate shapefile: 43.8 % ext. success rate	geoextent could be used to improve the rate of geotiffs of existing records. Additional human verification could be needed. Zenodo metadata: 0.2% Geoextent: 14.4%
--	---	--	--



http://bit.ly/geoextent_EC2021

8

Screenshot of presentation showing the results of our case study with geoextent

For more information about geoextent you can follow these links:

- [geoextent repository](#)
- [launch binder](#)
- [Exploring Research Data Repositories with geoextent notebook - Zenodo record](#)
- [Exploring Research Data Repositories with geoextent - presentation](#)

EarthCube meeting 2021

In addition to presenting geoextent, the participation in the event allowed us to get an insight into notebooks as research objects and scientific publications. Some of the reflections on the evolution of the guidelines, review process, and selected notebooks with respect to the first call were discussed in a [panel](#). In the same panel, representatives of the Jupyter, R Markdown, and Matlab communities presented different tools to share research results and how they could be integrated better within the context of scientific publications.

Among the other [18 accepted notebooks](#) we found interesting tools, for example [cf_xarray](#), used to simplify the usage of Climate and Forecast (CF) compliant datasets by improving the metadata of files [launch binder](#), a methodology to access to OpenTopography's Cloud Optimized GeoTIFF data for topography information [launch binder](#) or an educational platform to learn about glaciers [launch binder](#). All of these studies give us a picture of different geosciences research questions and how they are presented in fully reproducible workflows.

New preprint on o2r architecture and implementation

08 Jul 2021 | By Daniel Nüst

A new preprint is out!

With the first commit on [the manuscript repository](#) made on August 1st 2018, almost three years ago, this one is long overdue. In the article, o2r team member [Daniel](#) summarises over 5 years of designing and implementing the the o2r software infrastructure for *Executable Research Compendia* (ERCs). The article “*A Web service for executable research compendia enables reproducible publications and transparent reviews in geospatial science*” is now published on Zenodo:

Nüst, Daniel. (2021, July 8). A Web service for executable research compendia enables reproducible publications and transparent reviews in geospatial sciences (Version 1). Zenodo. <https://doi.org/10.5281/zenodo.4818120>

The preprint covers the ERC specification, o2r API specification, the o2r system architecture, and the o2r reference implementation - the *ERC Reproducibility Service*. It contains an extensive discussion and provides lessons learned from implementing ambitious ideas for increasing reproducibility. This article's most important section though is Acknowledgements: all o2r team members, past and present, contributed to making the vision of ERCs for scholarly communication a reality. *Thanks everyone!*

The screenshot below shows the preprints first page.

A Web service for executable research compendia enables reproducible publications and transparent reviews in geospatial sciences

 Daniel Nüst*

*Institute for Geoinformatics (ifgi), University of Münster, Germany (daniel.nuest@uni-muenster.de)

Preprint published on Zenodo at <https://doi.org/10.5281/zenodo.4818120> under CC-BY-4.0 license. This version was compiled on 2021-07-08 based on git commit d00c4205 from the repository https://zivgitlab.uni-muenster.de/d_nues01/architecture-paper.

The Executable Research Compendium (ERC) is a concept for packaging data, code, text, and user interface configurations in a single package to increase transparency, reproducibility, and reusability of computational research. This article introduces the ERC reproducibility service (ERS) for a publication workflow enhanced by ERCs. The ERS connects with existing scientific infrastructures and was deployed and tested with a focus on data and visualisation methods for open geospatial sciences. We describe the architecture of a reference implementation for the reproducibility service, including the created Web API. We critically discuss both the project set-up and features of ERC and ERS, and examine them in the light of various classifications for reproducible research. The ERC and ERS are found to be a powerful tool to improve reproducibility and thereby enable better investigating and understanding of computational workflows during peer review. We derive lessons learned and challenges for future scholarly publishing of computer-based geospatial research.

reproducible research | reproducibility | open science | executable research compendium | ERC | research infrastructure | research compendia | containerisation

1. Introduction

Open Science and reproducibility are enormous challenges for research, as computers and algorithms infuse all scientific disciplines, including geography and geosciences (David *et al.*, 2016; Nüst and Pebesma, 2020), and the scientific paper falls short in communicating the actual scholarship (Brammer *et al.*, 2011; Marwick, 2015; Gil *et al.*, 2016). The relevance of openness and reproducible reusable research are undisputed, just as the problems applying them in daily work, challenges around reproducibility, and handling in digital scholarly publishing workflows are real (e.g., Davison, 2012; Freire *et al.*, 2016). Software failures have led to wrong results and retractions (Miller, 2006; Gronenschild *et al.*, 2012) and “the lack of reported failures from geography and geosciences is not reassuring” (Nüst and Pebesma, 2020). Reproducibility in geospatial sciences, similar to most scientific disciplines, is low (e.g., Konkol *et al.*, 2019a; Nüst and Pebesma, 2020; Yan *et al.*, 2020; Nüst *et al.*, 2018). Although progress is made on openness in geospatial sciences, reproducibility has not been systematically addressed and increasing requirements for publication has just begun (Minghini *et al.*, 2020; cf. Peng and Hicks, 2021). A continued development of infrastructure sup-

ports for areas where change is needed are (a) requirements of funders and journals (cf. Hardwicke *et al.*, 2018, and Stodden *et al.* (2018); Nüst *et al.*, 2018), mechanisms to award recognition to all types of research outputs (Pillow, 2013), and (c) education and tools, so that all stakeholders have the means, i.e., resources, time, and knowledge, to create, examine, review, and publish reproducible open scientific workflows. To facilitate change on these levels, we have conceptualised and implemented an infrastructure to lower the barriers for creating, sharing, and reviewing reproducible publications. This work’s main contribution is a detailed description of that infrastructure and the demonstration of its functionality.

We present a Web service for open and reproducible publications for computational research in geography and geosciences: the ERC reproducibility service (ERS). The ERS is connected with the existing processes, services, and platforms for scholarly publications and serves the particular needs of geospatial data sciences. Examples and applications are taken from these domains as well, i.e., data-based workflows using observational data of the Earth. The ERS focuses on the third area of cultural change, education and tools, by putting the concept of the Executable Research Compendium (ERC, Nüst *et al.*, 2017) into practice as part of the scholarly publication process. The ERC at the centre of scholarly communication enables communicating, sharing, and collaborating on the actual scholarship as it includes data, software, and documentation (cf. Buckheit and Donoho, 1995; Davenport *et al.*, 2020). Previous work presented the benefits for authors and readers (Konkol *et al.*, 2019b). Here we describe the technical background and implementation of the ERS, and how it provides a missing functionality in scholarly publishing infrastructure.

In the remainder of this work, we first present related initiatives and approaches. Then we introduce a technical specification for the ERC followed by an architecture and reference implementation for a Web service for ERC creation and examination, which is connected with the existing landscape of scholarly publication infrastructures. Finally, we discuss limitations and lessons learned, and conclude with a summary and an outlook on future work.

Abstract

The Executable Research Compendium (ERC) is a concept for packaging data, code, text, and user interface configurations in a single package to increase transparency, reproducibility, and reusability of computational research. This article introduces the ERC reproducibility service (ERS) for a publication workflow enhanced by ERCs. The ERS connects with existing scientific infrastructures and was deployed and tested with a focus on data and visualisation methods for open geospatial sciences. We describe the architecture of a reference implementation for the reproducibility service, including the created Web API. We critically discuss both the project set-up and features of ERC and ERS, and examine them in the light of various classifications for reproducible research. The ERC and ERS are found to be a powerful tool to improve reproducibility and thereby enable better investigating and understanding of computational workflows during peer review. We derive lessons learned and challenges for future scholarly publishing of computer-based geospatial research.

Discuss and share

New preprint is out : "A Web service for executable research compendia enables reproducible publications and tr reviews in geospatial science"

<https://t.co/VAnRetcztL>

A sweeping blog about all specifications and implementations of 5+ years by team of [@o2r_project](#).
<pic.twitter.com/OGm7vlzVXn>

— Daniel Nüst (@nordholmen) July 8, 2021

Bachelor Thesis of student assistants Tom and Nick

22 Dec 2020 | By Nick Jakuschona, Tom Niers

Two of the o2r student assistants have just finished successful their bachelor theses. Congratulations!

This blog post introduces the excellent work by Tom and Nick - thank you for your dedication to advance Opening Reproducible Research.

Geospatial Metadata for Discovery in Scholarly Publishing

Tom Niers's thesis "*Geospatial Metadata for Discovery in Scholarly Publishing*" presents a novel approach to integrate well-defined geospatial metadata in a scholarly publishing platform to enhance discovery of scientific articles. For this purpose he developed the software [geoOJS](#), which offers a novel way for authors to provide spatial properties of research works when submitting an article to a journal based on the open source software [OJS](#).

Abstract

Many scientific articles are related to specific regions of the Earth. The connection is often implicit, although geospatial metadata has been shown to have positive effects, such as detecting biases in research coverage or enhancing discovery of research. Scholarly communication platforms lack an explicit modeling of geospatial metadata. In this work, we report a novel approach to integrate well-defined geospatial metadata into Open Journal Systems (OJS). Authors can create complex geometries to represent the related location(s) or region(s) for their submission and define the relevant time period. They are assisted by an interactive map and a gazetteer to capture high quality coordinates as well as a matching textual description with high usability. The geospatial metadata is published within the article pages using semantic tags, integrated in standardized publication metadata, and shown on maps. Thereby, the geoOJS plugin facilitates indexing by search engines, can improve accessibility, and provides a foundation for more powerful map-based discovery of research articles across journals.

Geospatial Metadata

Here the articles content can be specified in terms of location and time.

Temporal Properties

Define the temporal properties of the articles content by specifying date and time (time in GMT). The input is possible via the text field as well as via the calendar view, you just have to click the input field below this text. If you press "Apply" the result will be saved and with "Clear" nothing will be saved or in case something was already saved it will be deleted. The input needs to match the following format: "YYYY-MM-DD hh:mm:ss A", whereby "Y" stands for years, "M" for months, "D" for days, "h" for hours, "m" for minutes, "s" for seconds and "A" for AM or PM.

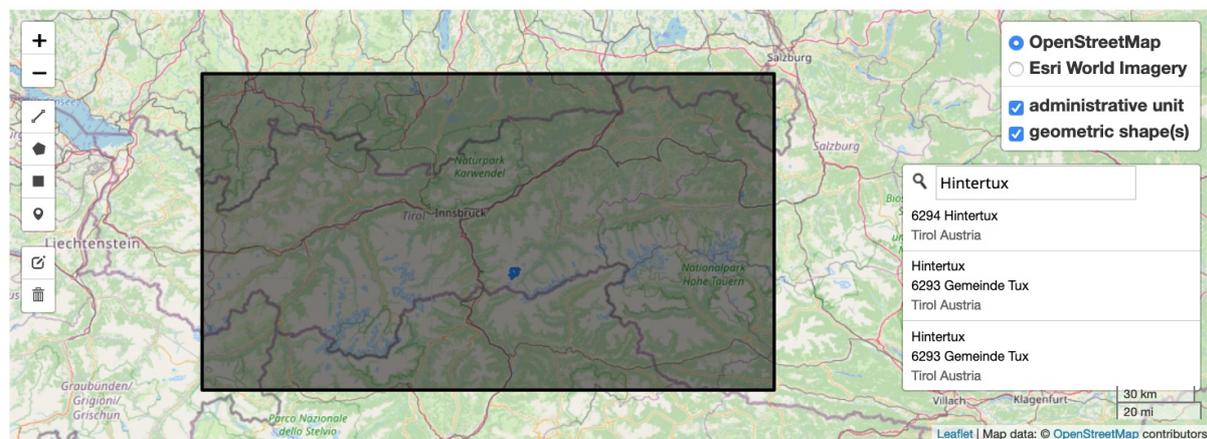
2013-01-01 12:00:00 AM - 2019-12-31 11:59:59 PM

< Jan 2013 Feb 2013 >

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
30	31	1	2	3	4	5	27	28	29	30	31	1	2
6	7	8	9	10	11	12	3	4	5	6	7	8	9
13	14	15	16	17	18	19	10	11	12	13	14	15	16
20	21	22	23	24	25	26	17	18	19	20	21	22	23
27	28	29	30	31	1	2	24	25	26	27	28	1	2
3	4	5	6	7	8	9	3	4	5	6	7	8	9

0 : 00 : 00 23 : 59 : 59

2013-01-01 12:00:00 AM - 2019-12-31 11:59:59 PM



Coverage Information

On basis of your input in the map, administrative unit(s) is/ are proposed which has/ have been selected according to your input in the map. Each time you update the map, the coverage information gets new calculated and updated correspondingly. You are able to delete administrative unit(s) by the red "x". If you hover over the administrative unit(s) the superior hierarchy of administrative unit(s) is displayed if available. Besides you can add further administrative units. You are only able to insert a further administrative unit if it fits to the already given hierarchy of administrative unit(s), and the given geometric shape(s) in the map. If you begin to insert, there are some suggestions you can accept by clicking, but nevertheless you can input your own administrative unit by hitting "Enter". The administrative unit (in black) which is the lowest common denominator for all geometric shape(s) is shown in the map. The administrative unit is not editable or deletable in the map, but here via the input field. If there are automatic changes in the map caused by changes in the coverage information and vice versa, this is indicated by a blue frame around the coverage element or the map.

Earth x Europe x Republic of Austria x Tirol x

Earth, Europe, Republic of Austria, Tirol

Screenshot of geoOJS – input of geospatial metadata, author can define temporal properties by calendar view, spatial properties by drawing or accept them by suggestions

The bachelor thesis was published on the University of Münster's document repository <http://nbn-resolving.de/urn:nbn:de:hbz:6-69029469735>.

Tom presented his work at the [Munin Conference of Scholarly Publishing 2020](#), a conference on scholarly publishing and communication with focus on open access, open data and open science. His ideas and the prototype were received very well, and the discussion yielded many ideas for further development. The abstract, the slides and a recording of the conference are available at <https://doi.org/10.7557/5.5590>. A video of the geoOJS presentation and the Q&A that followed is available on [Youtube](#).

Tom's experience in OJS is driving the development of the o2r OJS plugin for [Opening Reproducible Research in OJS](#) and we

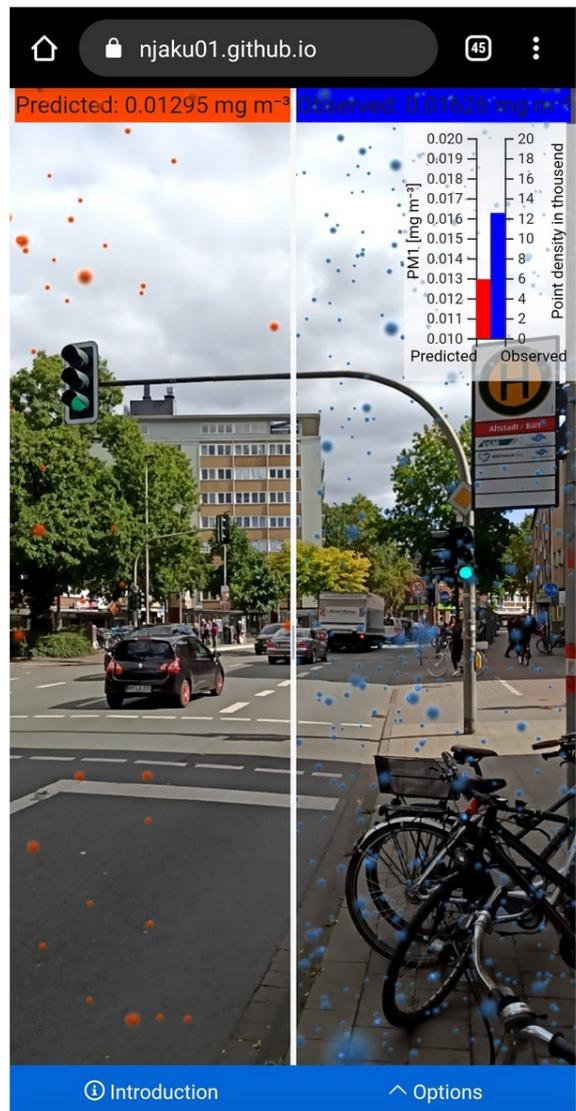
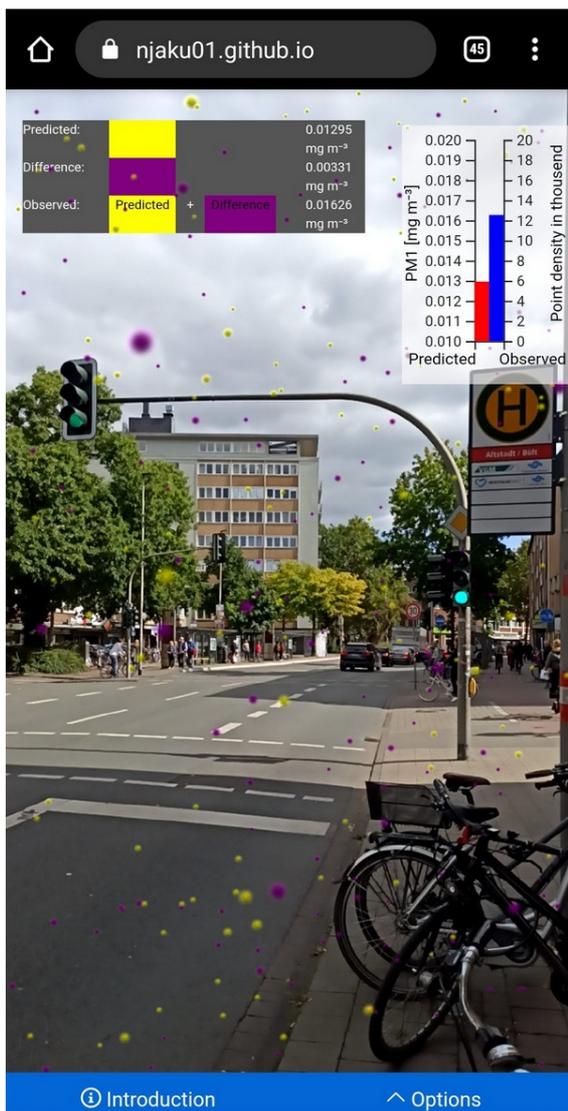
are looking forward to present our first developments soon.

Combining Augmented Reality and Reproducibility to convey spatio-temporal results

Nick Jakuschona researched a novel way of displaying spatio-temporal information in scientific articles and reported his findings in the thesis "*Combining Augmented Reality and Reproducibility to convey spatio-temporal results*". He makes use of the rising concept of Augmented Reality (AR) to display scientific data and shows how to create an AR application out of a reproducible article". You can try out the application on your own mobile device at <https://njaku01.github.io/>.

Abstract

Spatio-temporal research results are usually published in a static format, for example, as PDF. Here the results are not directly linked to their spatial reference. Therefore, it is difficult for the user to understand these results. To improve the user's understanding, we link these results with the real world. To archive this, we use the rising concept of Augmented Reality, where it is possible to integrate the results into the view of the user and to display the results on site. The results are often calculated out of a specified dataset. To ensure the data used for the application indicates the same result presented in the article, the outcome must be reproducible. The goal is to combine reproducibility and Augmented Reality to convey spatio-temporal results. We answered the research question about how to create an Augmented Reality application out of a reproducible article. Therefore, we performed a literature research and developed a concept which provides a guideline and explains the important steps. Starting with extracting the data used to calculate the results. Designing the app and deciding which types of visualization and devices fit best for the result and implementing the application. To show the feasibility of the concept, we created an application to convey the results of one scientific article. This application was evaluated with an expert user study, with the goal to indicate whether the application is understandable and easy to use. Furthermore, the general interest in using Augmented Reality applications to inspect spatio-temporal results got researched. The results of our research show that it is possible to convey spatio-temporal results through Augmented Reality. The results are displayed understandable. Overall, Augmented Reality is an interesting approach to display results out of scientific articles which should be depended in further research.



Screenshots of the Augmented Reality application showing results of the paper in two different views.

This application was evaluated with an expert user study. The results of this study are published as an [ERC](#). Nick hopes that this novel approach of visualizing results will be used to make research accessible and understandable for everyone. The future goal is to help authors, creating their own Augmented Reality applications. For example, with an Augmented-Reality-Application-Builder on the [o2r-homepage](#), which creates Augmented Reality applications out of ERCs.

New papers out about 'Practical Reproducibility in Geo' and the 'Rockerverse'

14 Oct 2020 | By Daniel Nüst

Two papers have been published this week by o2r team members Daniel and Edzer. They shed a light on the practical aspects of publishing computational geospatial research in a reproducible way and the enormous number of projects and opportunities of using containers with R to achieve reproducibility. *Please share widely!*

Practical Reproducibility in Geography and Geosciences

Daniel Nüst and Pebesma, Edzer. 2020. **Practical reproducibility in geography and geosciences**. Annals of the American Association of Geographers. doi:[10.1080/24694452.2020.1806028](https://doi.org/10.1080/24694452.2020.1806028)

The article is not Open Access, but there is an [author accepted manuscript PDF](#)

This paper is part of a collection of papers solicited after [a workshop early last year](#). We recommend to look at the other works, as they provide very interesting perspectives that complement the quite technical and practical approach in Daniel and Edzer's article. Join the discussion on Twitter:

Today is a fun week of announcements. Another paper is out! This time about "Practical Reproducibility in Geography and Geosciences" together with [@edzerpebesma](https://t.co/8WgKvuENwx) <https://t.co/8WgKvuENwx> [#ReproducibleResearch](#) [#Geography](#) [#Geosciences](#) pic.twitter.com/4NtPaxPuhF

— Daniel Nüst (@nordholmen) [October 14, 2020](#)

The Rockerverse: Packages and Applications for Containerisation with R

Daniel Nüst, Dirk Eddelbuettel, Dom Bennett, Robrecht Cannoodt, Dav Clark, Gergely Daróczi, Mark Edmondson, Colin Fay, Ellis Hughes, Lars Kjeldgaard, Sean Lopp, Ben Marwick, Heather Nolis, Jacqueline Nolis, Hong Ooi, Karthik Ram, Noam Ross, Lori Shepherd, Péter Sólymos, Tyson Lee Swetnam, Nitesh Turaga, Charlotte Van Petegem, Jason Williams, Craig Willis and Nan Xiao. **The Rockerverse: Packages and Applications for Containerisation with R** The R Journal (2020), 12:1, pages 437-461. doi:[10.32614/RJ-2020-007](https://doi.org/10.32614/RJ-2020-007)

This paper was actually inspired by [a post in this blog](#) four years ago and turned into a massive collaborative effort which is now published as Open Access in The R Journal. Join the discussion on Twitter:

A paper is out (or rather the DOI [10.32614/RJ-2020-007](https://doi.org/10.32614/RJ-2020-007) just now resolves..)! It feels anticlimactic with the preprint out for quite a while, but I'm grateful the [#RJournal](#) accepted this great and large collaborative effort: <https://t.co/qICMAU45SV> [#rstats](#) [#Docker](#) [@Docker](#) <https://t.co/VrO8En1HXx> pic.twitter.com/Nr0xJaAeE1

— Daniel Nüst (@nordholmen) [October 13, 2020](#)

Beyond o2r: collaborations and community activity for more open and reproducible science

26 Jun 2020 | By Daniel Nüst

The o2r project has its primary goals in providing tools to enhance scholarly communication. We build technology to help solving relevant problems. With the Executable Research Compendium and supporting software, we provide a different, more holistic take on how research output should look like in the future, especially if data and software are involved in the scientific workflow. However, tech is not all we do and o2r team members actively work with the GIScience community and the broader scientific community. This blog post briefly introduces two recent collaborations that are less about technology and more about community, culture, and people.

1. Citable and preserved AGILE short papers

We're a big fan of the AGILE conference: we have [organised several workshops](#), contributed to [stocktacking the reproducibility of AGILE conference publications](#), and co-authored the [AGILE Reproducible Paper Guidelines](#). These activities are collected under the umbrella of *Reproducible AGILE*:



The AGILE conference features a full paper and a short paper track. The *full paper proceedings 2020* are [published as Open Access \(yay!\) with Copernicus](#), which is a huge step towards more accessibility and openness in the community. Because of the cancellation of the conference, there are no *short paper proceedings* in 2020.

However, the **short papers** up to 2019 are published as so called “[bronze Open Access](#)”, meaning that they are published on a website and can be downloaded, but the license is unclear. The the question of preservation is not properly answered either, and referencing AGILE short papers is not possible up to today's standards because they lack a unique identifier. This is a huge shame, because, having published a few AGILE short papers, I know that the peer review process is solid and very helpful especially for ideas in an early stage. Furthermore, short papers are often written by early career researchs (in fact, my first ever scientific publication as first author was [at AGILE 2010](#)).

That is why I reached out to the [AGILE council](#) and suggested to add a statement to the [AGILE proceedings website](#), which clearly gives authors the permission to re-publish or “[self-archive](#)” the short paper PDFs in a proper repository. My initiative was triggered by a concrete event: one of my AGILE short papers was not accepted by my favourite preprint server [EarthArXiv](#) (which also hosts postprints, see also [on Wikipedia](#)) because it was not clear I had permission to submit a paper that was previously published elsewhere. This is a very reasonable [moderation policy](#), and the interaction with EarthArXiv advisory council member [Allison Enright](#) in the matter was extremely nice and helpful. After providing some good arguments via email and some endurance, I was very happy to learn that the council followed my suggestion and added the following statement to [the proceedings website](#):

Authors have permission to deposit AGILE short papers, published in the proceedings below and available as PDFs on the server <https://agile-online.org>, in a public repository, such as a preprint server or institutional repositories. Authors may only use repositories that provide a DOI for the published record.

Authors are strongly encouraged, and may be required by repositories e.g. EarthArXiv (<https://eartharxiv.org/>), to add a cover page to the uploaded PDF. The cover page should include name, time and place of the conference, the URL to the conference website, and a statement that the short paper is peer reviewed. If possible, authors should add the tags or keywords 'AGILE short paper' and 'AGILEGIS' and configure the recommended citation to include year and name of the conference.

How this all played out confirmed my trust and appreciation for the EarthArXiv and AGILE communities.

Did you author an AGILE short paper in the past? Please help to preserve the knowledge of the GIScience community. You

can do it today! Find step-by-step instructions and some more background here: <https://reproducible-agile.github.io/short-paper-postprints/> It really just takes 5 minutes.

AGILE #AGILEGIS short papers can now be self-archived/deposited in institutional repositories or preprint servers (recommendation: [@EarthArXiv](#)

♥ <https://t.co/FiRd5YWEDJ>

Preserve your work now! (and ensure others can cite you properly...)#postprint #OpenAccess pic.twitter.com/fj3INAMC8tU

— Daniel Nüst (@nordholmen) June 18, 2020

2. CODECHECK

o2r has an [approach and goals](#) grounded in the belief that technology can help to reduce barriers for reproducibility and make benefits of reproducible publications more readily available to the broad diversity of geoscientific researchers. Our focus lies in packaging code (scripts, runtime environment), data, and documentation together (the [ERC](#)) and integrate it into peer review and the scholarly publication process (our [pilots](#)).

While these goals and approach are true, and start to come to fruition, one can also take a completely different approach. Enter [CODECHECK](#).



CODECHECK is a joint initiative by Daniel Nüst and [Stephen J. Eglen](#), reader in Computational Neuroscience at the University of Cambridge. Daniel and Stephen were brought together by failure: both applied for a small Open Science grant with the Wellcome trust, but both were rejected. Luckily, they both took advantage of the option to publish their project proposals, so they could see they had similar ideas. Also starting out as technology driven, CODECHECK has developed into something completely different from o2r.

To introduce better recognition of computational workflows in the peer review process, Stephen and Daniel developed a set of [four principles](#). Based on these principles, scientific journals or the community can build a process, of which [many variants are imaginable](#), for executing code and data-based workflows during peer review.

1. Codecheckers record but don't investigate or fix.
2. Communication between humans is key.
3. Credit is given to codecheckers.
4. Workflows must be auditable.

These principles embrace openness ideals and the opportunity to introduce early career researchers and research software developers in peer review. Instead of trying to preserve and package everything, CODECHECK transfers the gist of peer reviewing articles to code execution: at one point in time, one fellow researcher or developer was able to execute a given workflow following the provided instructions. Some see this as a low bar, I see it as an option to break the current stagnancy of code review in science (see also the [CODECHECK FAQs](#)).

The currently most active implementation of these principles is the [community process](#), but the first successes of CODECHECKs conducted as part of journal publications are also already completed and the number of volunteering codecheckers is slowly rising. Stephen put in a lot of effort to contribute to the scientific knowledge by codechecking coronavirus simulations, which not only strengthens trust in science but lead to a nice [Nature News article](#). Please check out the CODECHECK website and the [CODECHECK register](#) for details, and see [how you can get involved](#) as author, reviewer, or journal editor.

Low tech, community work, and technological advances go hand in hand in Opening Reproducible Research.

o2r student assistant about impressions of reproducibility ready to start a career in research

15 Jun 2020 | By Laura Goulier

“Geoscientist with experience in or willingness to learn R programming for reproducible research wanted!”

I had just completed a beginner course in R programming for my master's thesis and saw my chance to further develop this knowledge and enter the field of geoinformatics, even get a little away from the pure ecology of my master studies in landscape ecology. I had never before heard of the words “reproducible research”, neither heard of any reason why this topic is of importance. So I took the job and worked my way in. After a couple of months I had to realise that in order to publish my master's thesis, it was the journals obligation to make all code and data openly available to enable other researchers so they could fully understand and reuse my analysis. And there I was, as a landscape ecologist who believed I had nothing to do with reproducible research. Apparently it is important after all, yet not that simple.

During my work in the o2r project I experienced first hand the whole range of reasons why people struggle so much making their work reproducible for others. The main argument, also for me, was this giant amount of additional work. Is it really worth it, I thought? I also believed I had my own structure while scripting and it would be much easier for me not to script in a way so other people understand my analysis, but to primarily make myself understand it. “I would have to spend an entire extra year for my PhD, just to prepare all scripts again for everyone to comprehend”, some PhD students from the atmospheric sciences told me. The desire for reproducibility in research is not always an open door. But maybe it is the same as for everything else. A clean method of working should always be the goal. Students in school should write cleanly so that the teacher can understand their essays. Every company needs a well organised structure to be successful. Scripting, so that only myself and no one else can understand what has been calculated, may in the short term have its benefits as I understand my own work because of the embedded history and context. After two years at the latest, however, not even I myself could look through my work and answer specific questions about my calculations. If we are honest, it happens far too often that we don't know exactly what we thought at that time, we made that one small change or attempted to fix that nasty bug. We tend to lose track of which scripts contain which results, how a certain parameter was calculated, or what the results would look like if we would change certain values. Getting it right from the beginning is not an extra effort though, it is just a change in the way we work, which saves us time in the long run. And not only for us, but also for many others who no longer need to find answers to the same questions or redo complex analyses themselves.

Now that I finished my master's thesis, my time in the o2r project is over and I am starting my PhD [terrestrial data analytics at Jülich Research Centre](#), investigating the impact of human water use on atmospheric extremes. During my job interview, they asked me quite a lot of details about my work at o2r, about limitations and obstacles, about difficulties and successes. This signals to me that reproducibility is not only gladly implemented, but is also an inevitable change that everyone must consider and adapt to, even if it is sometimes bothersome and entails some difficulties that we did not have to think about before. For me, reproducibility also has a social component. To do things not only for oneself, but for making others' work easier and letting them benefit from one's own method. For my PhD, I am taking along to further improve my method of working for best practice, because it certainly takes a lot of training. As a beginner in academia, I strongly hope to get help by detailed insights into the scripts of more experienced scientists in order to facilitate my own research.

The o2r team thanks Laura for her contribution to the project. She did great work bridging between geoinformatics and landscape ecology and contributed greatly, among other things, to [a paper on platforms for reproducible research](#). We wish her best of luck for her future academic career!

Introducing geoextent

26 Apr 2020 | By Yousef Qamaz, Daniel Nüst

`geoextent` is an easy to use library for extracting the geospatial extent from data files with multiple data formats.

Take a look at the [source code on GitHub](#), the [library on PyPI](#) and the [documentation website](#). You can view and test `geoextent` implementation through interactive notebooks on [mybinder.org](#) with a click on the following binder.

launch binder

Here is a small example how to use `geoextent`.

```
geoextent -b -t -input= 'cities_NL.csv'
```

The output will show the rectangular bounding box, time interval and crs extracted from file data, as follow:

```
{'format': 'text/csv',  
'crs': '4326',  
'tbox': ['30.09.2018', '30.09.2018'],  
'bbox': [4.3175, 51.434444, 6.574722, 53.217222]}
```

The input file used above was obtained from [Zenodo](#). The map below based on [OpenStreetMap](#) shows the area of extracted bounding box.



You can get quick usage help instructions on the command line, too:

```
geoextent --help
```

geoextent is a Python library for extracting geospatial and temporal extents of a file or a directory of multiple geospatial data formats.

usage: geoextent [-h] [-formats] [-b] [-t] [-input= '[filepath|input file]']

optional arguments:

```
-h, --help          show help message and exit
-formats           show supported formats
-b, --bounding-box  extract spatial extent (bounding box)
-t, --time-box     extract temporal extent
-input= INPUT= [INPUT= ...]
                   input file or path
```

By default, both bounding box and temporal extent are extracted.

Examples:

```
geoextent path/to/geofile.ext
geoextent -b path/to/directory_with_geospatial_data
geoextent -t path/to/file_with_temporal_extent
geoextent -b -t path/to/geospatial_files
```

Supported formats:

```
- GeoJSON (.geojson)
- Tabular data (.csv)
- Shapefile (.shp)
- GeoTIFF (.geotiff, .tif)
```

Motivation

Geospatial properties of data can serve as a useful integrator of diverse data sets and can improve discovery of datasets. However, spatial and temporal metadata is rarely used in common data repositories, such as [Zenodo](#). Users may ask *what data is available for my area of interest over a specific time interval?* This question formed the initial idea for creating a library that can serve as the basis for integration geospatial metadata in data repositories. Because a core function is the extraction of the geospatial extent, we named it **geoextent**. The data extracted using the library can be added to record metadata, which will allow users, specifically researchers, to find relevant data with less time and effort.

Origins

The library's source code is based on two groups projects ([Cerca Trova](#) and [Die Gruppe 1](#)) of the study project [Enhancing discovery of geospatial datasets in data repositories](#). We decided to develop the library with Python as we plan to integrate it with o2r's metadata extraction and processing tool [o2r-meta](#).

Process of creating the codebase

Luckily we did not have to start from scratch but could make **geoextent** a reimplementaion of existing prototypes. We roughly followed these steps:

- Evaluate the existing code of the [study project groups](#)
 - Review the code implementation
 - Identify parts of the code that are re-usable
- Integrate chosen parts
- Develop of core features
- Set up [tests on Travis CI](#)
- Publication of library on [PyPI](#)
- Writing library documentation using [Sphinx](#) and render it as part of the Travis CI process
- Adding introduction Notebooks for easy testing with [MyBinder](#)

Current features

- Extract bounding box.

```
geoextent -b -input= 'wf_100m_klas.tif'
```

Output:

```
{'format': 'image/tiff',  
'crs': '4326',  
'bbox': [5.91530075647532,  
50.3102519741084,  
9.46839871248415,  
52.5307755328733]}
```

- Extract time interval

```
geoextent -t -input= 'muenster_ring_zeit.geojson'
```

Output:

```
{'format': 'application/geojson',  
'crs': 4326,  
'tbox': ['2018-11-14', '2018-11-14']}
```

- Show coordinate reference system (CRS) used
- Supported formats:
 - GeoJSON (.geojson)
 - Tabular data (.csv)
 - Shapefile (.shp)
 - GeoTIFF (.geotiff, .tif)

For more examples, see [documentation](#).

Next steps

As an immediate next steps, we want to integrate the extraction of extents into `or2-meta` so that users creating an ERC will have to do less manual metadata creation. We also hope that `geoextent` is useful to others and have plenty ideas about extending the library. For example, being a Python project, we would like to explore integrating `geoextent` into Zenodo. Most importantly, we will add support for multiple files and directories, but also further data formats - see [project issues on GitHub](#). *We welcome your ideas, feature requests, comments, and of course contributions!*

Next generation journal publishing and containers

26 Feb 2020 | By Daniel Nüst, Tom Niers

Some challenges of working on the next generation of research infrastructures can be solved most effectively by talking to other people. That is why o2r team members Tom and Daniel were happy to learn about the [announcement](#) of an [Open Journal Systems \(OJS\) workshop organised by Heidelberg University Publishing \(heiUP\)](#)

The o2r team was a little bit the odd one out. Other workshop participants either had extensive OJS development experience, or were not developers at all but running production systems of many OJS journals across the German university landscape. But that could not keep us from telling everyone about [Executable Research Compendia](#), of course. We briefly summarised our plans to [extend OJS with ERC capabilities](#), *but we also had new stuff to share!* Tom is considering to put his *geo-informatics* skills to use and extend the metadata of OJS articles with geospatial features in his Bachelor thesis. This would allow to display the spatial area of articles on a map, and even browse articles by their location(s). **Learn more about these ideas in our slides.**

Today team members [@nordholmen](#) [@herrniers](#) meet the German [#OJS](#) developer and user community at a workshop organised by [@heiUP_HD](#) [@ojs_pkp](#)

Of course we want to talk [#spatial](#) data and [#ERC](#) in OJS!

♥ [#SpatialIsSpecial](#) [#ResearchCompendium](#) [#ScholComm](#) <https://t.co/5FenW7WUae>
pic.twitter.com/BrxkAiSE72

— o2r (@o2r_project) February 20, 2020

But Tom and Daniel also came with a mission: to jumpstart the struggling OJS developments with the help of some experienced OJS developers. None of our team has extensive experience with PHP, so getting control over the huge OJS codebase and setting up a proper **OJS development environment with debugging** was an important task they've been pushing aside since autumn last year. *And we got it!* [Note to self: don't forget to enable `remote_enable` and `remote_autostart` for Xdebug in the file `/etc/php/7.3/cli/conf.d/20-xdebug.ini` for debugging to work - then the default VSCode configuration with port `9000` will just work (-)]. On top of that, Tom got a very **helpful introduction to writing OJS plug-ins** and Daniel now has a good grasp on the **currently developed Docker images for OJS**. The Docker images are not a simple project, since the PKP team plans to support multiple webserver implementations, multiple PHP versions, and all OJS versions still in production somewhere... phew! Daniel even **opened a pull request** and suggests a different way to support both remotely and locally built images. This prepares us well for the moment when we want to run OJS on our own servers - in containers of course. So the expectations were high, but eventually they were not disappointed. Daniel was glad to see some familiar faces from the [OJS-de.net community](#) he met at a previous workshop in Heidelberg. The new contacts made were more just as important as the helpful practical tips towards becoming real "OJS devs" 😊

Other groups at the workshop reported very interesting results, for example on the connection of OJS with proper digital archives (with promising mentions of also archiving data... and code?), more flexible publishing workflows with own tools (mentioning Pandoc, which might make these flexible pipelines a first step towards (R) Markdown-based OJS publications 🌀), and using search indexes such as Solr and Elasticsearch within OJS (which also have geospatial capabilities). As you we're very hopeful future collaborations will spark from these educational and entertaining encounters.

WWU workshop on Reproducible Research

19 Feb 2020 | By Daniel Nüst

Reproducible research is a topic relevant for all scientific disciplines. We in the o2r project have a continued focus on the challenges originating in the software stacks and visualisations for the analysis of geospatial data. But that does not mean that our experiences may not be helpful for other disciplines. It does also not mean that our approaches for improving research reproducibility and reusability can not profit from learning about challenges and solutions in other domains.

That is why we decided to reach out to the local scientific community and talk about reproducibility. We invited all professors and post-docs of the University of Münster (WWU) to a workshop at the Institute for Geoinformatics. Why only senior researchers? One goal was to start discussions about collaborating on new projects and writing proposals, and we thought this group would be interested in that. We welcomed over 20 researchers across the full diversity of WWU, e.g., neuroscience, landscape ecology, business informatics, and psychology. The event was held in German and all material is available [on the workshop website](#).

Learning about concepts and experiences across these communities show the many different perspectives and challenges around the ideal of [#reproducibility](#). Now we continue in group discussions to identify common pain points and start new collaborations. [#interdisciplinary pic.twitter.com/tegKwBKjjo](#)

— o2r (@o2r_project) [February 11, 2020](#)

We thank our colleagues for the interesting discussions and new perspectives on a topic we thought we would have a good grasp of - there's so much more to learn! Special thanks go to our fellow researchers who prepared short talks on their experiences and ongoing work to improve reproducibility.

We thank the participants for a great day with interesting discussions and plenty of looking of the rims of one's own disciplines tea cups (if that makes sense). We collected creative and innovative ideas for inter/trans/cross-disciplinary projects to improve [#reproducibility](#). [pic.twitter.com/FUccyC4AOg](#)

— o2r (@o2r_project) [February 19, 2020](#)

Very special thanks go to [Dr. Lisanne Pauw](#), [[Dr. Nils Schuhmacher](#)], [Dr. Ben Stöver](#), and o2r team member [Daniel Nüst](#), who volunteered to write up a short story about their personal work connected with reproducible research. These stories are published on the university website [in English](#) and [German](#). We hope these spark the interest of fellow scientists or even the general public. Thanks to Kathrin Kottke from the WWU public relations team for making this happen.

Four researchers contributed short stories about [#reproducibility](#) in their fields, published now in German and English on [@WWU_Muenster](#)'s news page:

<https://t.co/SZDLAdURq3>

<https://t.co/ciYibkaTLp> [#spatialsciences](#)

[#psychology](#) [#bioinformatics](#) [pic.twitter.com/04GGnNeA9L](#)

— o2r (@o2r_project) [February 19, 2020](#)

o2r2 project proposal publication

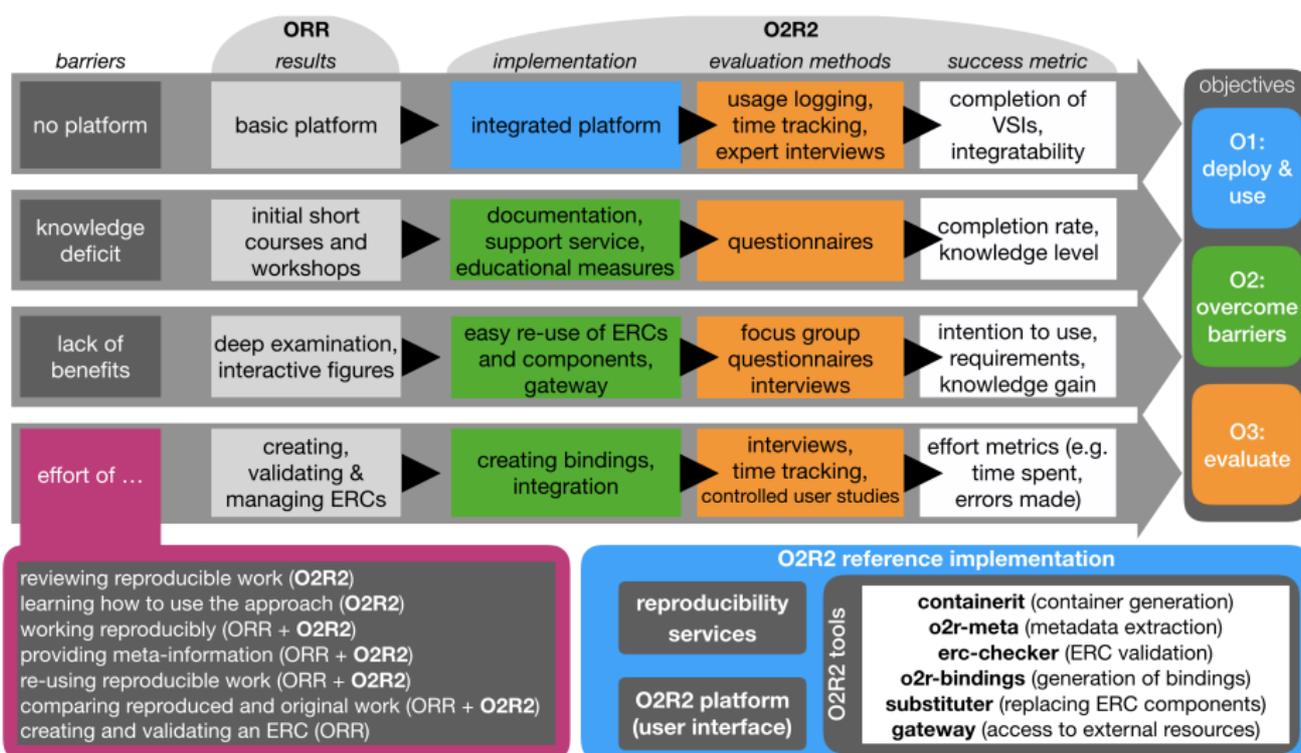
03 Feb 2020 | By Daniel Nüst

Ten months ago, we announced that the o2r team received funding for a second project phase. Today we publish our project proposal on the University of Münster's institutional repository MIAMI: <https://doi.org/10.17879/42149626934> (urn:nbn:de:hbz:6-42149629066)

We hope this publication of our proposal serves several purposes: it encourages fellow researchers to also share their plans openly (both funded and rejected), it motivates us to achieve the ambitious goals, and (we have to admit) the proposal's indexing in search engines hopefully leads to some attention for the o2r project and subsequent new contacts and collaborations. Science reinvents the wheel much too often, and the o2r project team wants not only to **increase openness, transparency and reusability of research workflows** but also in the bigger picture of research funding and building of research infrastructures. Our perspective on the distinction between o2r and other projects described in the proposals is also the motivation behind our recent paper: *"Publishing computational research - a review of infrastructures for reproducible and transparent scholarly communication"*.



Figure: Overview of the work programme and objectives. The table shows key barriers, how these were initially tackled during ORR, and which implementations are planned to overcome them during O2R2 (green and blue boxes). Orange boxes show how we plan to evaluate implementations and deployment. The success measures indicate how we determine success. The blue box (bottom) summarises the technical outcomes of O2R2.



The document is a shortened version of the second revision submitted in May 2018, without funding information and project planning details but with a slightly updated title page to include DOI, URN, and a subtitle marking the public version. The submission underwent a single-blind peer review by experts from information science and geosciences after first submission in August 2017. The first version originally included a thematic extension into life sciences, with a number of new collaborators at the

University of Münster, and the inclusion of Python as a second supported base software. A remaining point of critique was the description of our methodology from a research perspective. We were lucky that the information science reviewers and ultimately the deciding council agreed in the value of the experiences made from the perspective of infrastructure development, which can only be made with a concrete practical evaluation. The increased focus allowed us to reduce the project duration and continue with the existing approach, but also puts the remaining [pilots](#) under time pressure. More on the pilots soon!

o2r @ ECMWF Workshop in Reading, GB

21 Oct 2019 | By Markus Konkol

Claudia Vitolo from the European Centre for Medium-Range Weather Forecasts (ECMWF) had the brilliant idea to host a workshop about [building reproducible workflows for earth sciences](#). It is not surprising that weather forecasts strongly depend on computational analyses, statistics, and data. Wait! Isn't this exactly what o2r addresses? Well observed, that is certainly correct. For this reason, we were very happy to receive an invitation from Claudia for giving a keynote. But let's start from the beginning.

Ana Trisovic from the Harvard University opened the workshop with an interesting keynote about the reproducibility challenges in physics and the social sciences. She also conducted a reproducibility study to investigate if R scripts stored on Dataverse are actually reproducible. Her results were similarly worrying as those reported in our paper about [computational reproducibility in the geosciences](#). From the 3208 R files, only 502 could be executed successfully. The remaining scripts had issues such as a wrong file directory or a missing functionality.

The second keynote was given by Carol Willing from Project Jupyter. She argued that lives depend on scaling reproducible research and took the example of the typhoon that hit Japan recently. Her main point was that reproducible research improves prediction which is particularly necessary in the context of storms. Having reliable predictions can help people to prepare accordingly. Based on this use case, she presented some Jupyter-based tools such as Jupyter notebooks and Binder.

Many other talks presented approaches to address very specific reproducibility issues. Some of these approaches build on top of Jupyter notebooks and containers which demonstrates again that these two tools are probably the right way to go for the next few years. However, the speakers did not put much focus on user-related aspects and the publication process. As a consequence, the talks were more about the technical realization and less about creating a connection between the article and the reproducible analysis. This was a nice gap for o2r to fill. Markus presented our key concepts such as the [Executable Research Compendium](#) (ERC) and [bindings](#) and how these can be integrated into the process of publishing scientific articles.

All in all, the number of talks indicates that there is some need for exchange about reproducibility in weather forecasting. This is not surprising due to the intense and often emotional discussion about climate change. Reproducible research can help to demonstrate the robustness of the results and to find errors in the analysis before publication. Hence, this way of publishing research makes it easier to counteract two popular points of attack used by climate change deniers. We hope ECMWF is going for a second edition next year!

By the way, all slides and even the video recordings of the talks are available online:

<https://events.ecmwf.int/event/116/timetable/#20191014.detailed>



Opening Reproducible Research with OJS

15 Oct 2019 | By Daniel Nüst, Tom Niers

Data and software are crucial components of research. They go well beyond the workflows one would call *Data Science* today. Only openly available building blocks can ensure transparency, reproducibility, and reusability of computer-based research outputs. More and more researchers rely on small or large datasets and use analysis tools to analyse variables, create figures, and derive conclusions. That is why the project Opening Reproducible Research (*o2r*) implements the concept of the Executable Research Compendium (ERC) to capture all bits and pieces underlying a research article. In [a pilot study](#), we plan to connect the Open Journal Systems (OJS) with the ERC. On the one hand this connection enables submission, review, and publishing of [research compendia](#) and ERC. On the other hand while it leverages the publishing capabilities and workflow management of OJS. We will implement this integration in form of an [OJS plug-in](#) so it becomes readily available for all maintainers of OJS instances.

In this blog post [Tom](#) and [Daniel](#) describe our general procedure, the first concrete plug-in idea, and the planned plug-in structure.

o2r is a joint project by the Institute for Geoinformatics ([fgi](#)) and the University and State Library ([ULB](#)) at the University of Münster ([WWU](#)). The project is supported by the German Research Foundation ([DFG](#), see [About](#) page for details).

Procedure

After a first collection of ideas we started concretizing them in [user stories](#). The main user stories concern the idea of making research compendia, such as ERC, useable in the OJS-workflow (see details in the next paragraph). These stories may contain potentially generic features that could be realised as individual plug-ins for

- uploading multiple submission files, even from cloud storage, including large size files and public or authenticated shares, e.g. ownCloud, Dropbox, or GitHub,
- connecting articles with external data repositories (e.g. listing and preview of supplemental data published in [Open data repositories](#)),
- supporting [literate programming](#)-based article formats (e.g. ERC with R Markdown, Jupyter Notebooks) with rendering to HTML and/or PDF, or
- seamlessly connecting articles with interactive online workspaces with reusable data and code as an alternative to static fixed articles (e.g. using [Binder](#)).

However, the focus will initially be on the integration of a full ERC-based workflow into OJS. At a later stage, parts of this integration could be the starting point for the above individual plug-ins.

Based on the user stories, we then developed a few mockups (or [wireframes](#)) to get a better understanding how our ideas will likely look and to ease communication about the stories. The next step starts now: we develop the plug-in based on our mockups and user stories. To make sure we're on the right track we want to use this blog post to connect with the OJS community on our ideas and specifically search for feedback on the plans described below.

User stories

The full list of user stories can be found in [this spreadsheet](#). They are roughly sorted by priority. We even tried to guesstimate the efforts, though we expect to be quite far off during the first few stories until we get a better understanding of developing with OJS.

The following main user stories will be implemented first. They can be grouped into stories concerning creation (including upload) and examination (viewing, manipulating) ERCs.

ERC creation in OJS

- As author I want to upload all my files (data, code, text) directly from my computer, so that I save time (not each file individually) and the complete workflow is published.
- As author I want to insert the metadata for a submission at one location, so that I do not have to insert them several times.
- As editor I want my authors to be able to upload an (optional "executable") research compendium from their computer, so that data and software can be published as a unit and I can find suitable reviewers.
- As editor I want there to be a review step regarding reproducibility of the article, so that the quality of reproducibility of articles in my journal increases.
- As editor I want research compendia in general and ERCs to be automatically validated on the platform, so that I don't have trouble with them and the compendia are nevertheless complete.

- As site admin I would like to install a Research Compendium Upload from my computer as a plug-in in OJS, so that I can offer this feature to authors.

ERC examination in OJS

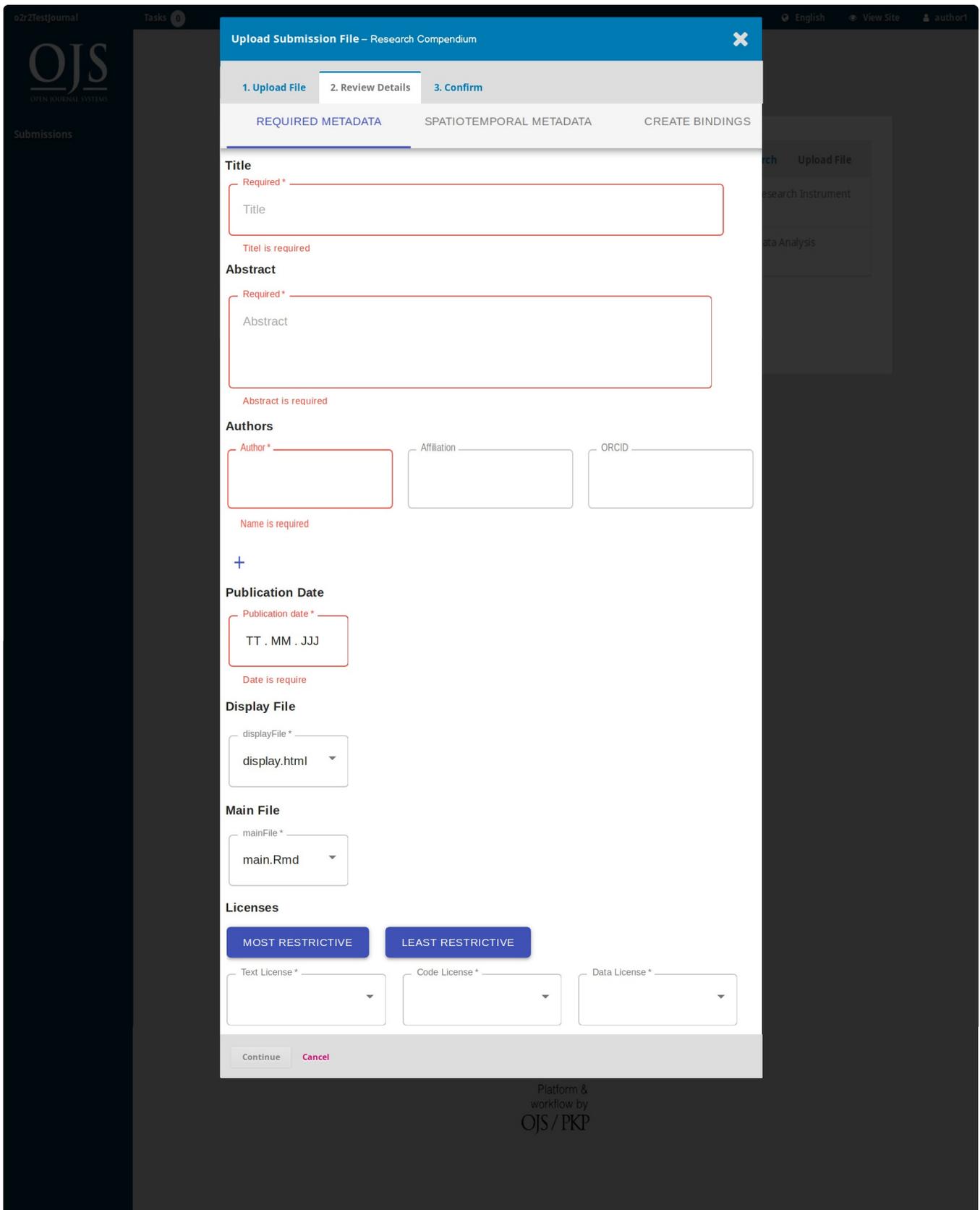
- As reviewer I want to view, download, survey and manipulate the ERC, so that I can check even complex workflows without much additional effort.
- As reader I want to view, download, survey and manipulate the ERC within the article page, so that I am able to understand the research work.
- As editor I would like the readers of the journal to be able to view the ERC in the issue of the journal, so that the quality of the journal increases.
- As site admin I want to be able to install a plug-in in OJS that allows you to view and manipulate ERCs, so that I can offer this feature to authors and reviewers.

ERC plug-in for OJS

How do we want to realize our user stories?

Upload Executable Research Compendium

To replace a regular article with an ERC in OJS, there is of course the need to upload it. The idea is to add a new file type for finished ERCs. But we also want to give the user the opportunity to create a ERC during the submission process within OJS. Therefore we plan to customize the upload process. The user will have the option to upload the files for the ERC and then to modify ERC metadata (publication metadata, spatio-temporal metadata). The authors will also be able to create [bindings](#). The following mockup shows how we imagine the upload process of an ERC.



Mockup 1.: Submit an ERC in OJS (metadata form)

Review Executable Research Compendium

After uploading the article, the next step in the OJS workflow is the review process. In this process the reviewer should be able to both download the ERC and to inspect the ERC online. Therefore a preview is needed, which does not differ from the view the reader is finally seeing. The preview only shows an additional link which brings the reviewer back to the review page. In this view the user can read the main text document of the ERC (PDF or HTML), look at data and code files and figures, and manipulate a

workflow with bindings. To provide feedback to the author, a new text area “Reproducibility Review” is added to the third step “Download & Review” in the review stage in OJS. Here the reviewer can comment on the understandability and reproducibility of the given workflow.

o2r2TestJournal Tasks 1 English View Site reviewer1

OJS
OPEN JOURNAL SYSTEMS

Submissions

Review: *submissionTitle*

1. Request 2. Guidelines 3. Download & Review 4. Completion

Review Files [Search](#)

Research Compendium	August 16, 2019	Article Text
---------------------	-----------------	--------------

[Download](#) [Preview](#)

Review
Enter (or paste) your review of this submission into the form below.

For author and editor

For editor only

Reproducibility Review
Enter (or paste) your review of this submission into the form below.

Upload
Upload files you would like the editor and/or author to consult, including revised versions of the original review file(s).

Reviewer Files [Search](#) [Upload File](#)

No Files

Review Discussions [Add discussion](#)

Name	From	Last Reply	Replies	Closed
No Items				

Recommendation
Select a recommendation and submit the review to complete the process. You must enter a review or upload a file before selecting a recommendation.

Choose One

[Submit Review](#) [Go Back](#)

* Denotes required field

Platform &
workflow by
OJS / PKP

Mockup 2.: Examine an ERC submission (download, preview links) and write review comments (reproducibility text box)

Examine Executable Research Compendium

The examination of an ERC in OJS, i.e. the viewing of compendium files and manipulation of workflows by reviewers and readers, is a core feature of the plug-in. The only differ in the link to get back to either the review form in the case of the reviewer or back to the article landing page in the case of the reader. We have two two different ideas how to realize ERC examination.

First, there is the possibility to integrate it directly on the main article page. The ERC with its file view and manipulation area is directly shown on the article page.

The screenshot shows a journal article page with a dark blue header. The header contains the journal name, navigation links (Current, Archives, About), and a search bar. The main content area is divided into two columns. The left column displays the article title "INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis" and lists the authors: Francesco Dottori, Rui Figueiredo, Mario L. V. Martina, Daniela Molinari, and Anna Rita Scorzini. The right column features a metadata panel with tabs for "INSPECT", "CHECK", and "MANIPULATE". The "INSPECT" tab is active, showing a dropdown menu with "main.Rmd" selected. Below the dropdown is a code block containing metadata for the ERC submission, including title, author, affiliations, licenses, and dates. The code block is followed by the text "There is no data to display".

journalName

Current Archives About

Search

reader2

Home / Archives / Vol 1 No 1 (1): issueTitle / Articles

ERC: Title

SHOW PDF

INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis

Francesco Dottori
European Commission, Joint Research Centre, Ispra, Italy

Rui Figueiredo
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Mario L. V. Martina
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Daniela Molinari
Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy

Anna Rita Scorzini
Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy

02 Dec 2016

Abstract

Methodologies to estimate economic flood damages are increasingly important for flood risk assessment and management. In this work, we present a new synthetic flood damage model based on a component-by-component analysis of physical damage to buildings. The damage functions are designed using an expert-based approach with the support of existing scientific and technical

INSPECT CHECK MANIPULATE

main.Rmd

```
---
title: "INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis"
author:
- affiliation: "European Commission, Joint Research Centre, Ispra, Italy"
  name: "Francesco Dottori"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Rui Figueiredo"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Mario L. V. Martina"
- affiliation: "Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy"
  name: "Daniela Molinari"
- affiliation: "Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy"
  name: "Anna Rita Scorzini"
licenses:
  code: CC-BY-3.0
  data: CC-BY-3.0
  text: CC-BY-3.0
  date: "02 Dec 2016"
```

There is no data to display

Platform &
workflow by
OJS / PKP

Mockup 3.1: View of an ERC for a reader (idea 1)

ERC: Title

SHOW PDF

INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis

Francesco Dottori
European Commission, Joint Research Centre, Ispra, Italy

Rui Figueiredo
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Mario L. V. Martina
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Daniela Molinari
Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy

Anna Rita Scorzini
Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy

02 Dec 2016

Abstract

Methodologies to estimate economic flood damages are increasingly important for flood risk assessment and management. In this work, we present a new synthetic flood damage model based on a component-by-component analysis of physical damage to buildings. The damage functions are designed using an expert-based approach with the support of existing scientific and technical

INSPECT CHECK MANIPULATE

main.Rmd ▾

```

---
title: "INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis"
author:
- affiliation: "European Commission, Joint Research Centre, Ispra, Italy"
name: "Francesco Dottori"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
name: "Rui Figueiredo"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
name: "Mario L. V. Martina"
- affiliation: "Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy"
name: "Daniela Molinari"
- affiliation: "Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli S
name: "Anna Rita Scorzini"
licenses:
code: CC-BY-3.0
data: CC-BY-3.0
text: CC-BY-3.0
date: "02 Dec 2016"

```

There is no data to display

Platform &
workflow by
OJS / PKP

Mockup 3.2: View of an ERC for a reviewer (idea 1)

Second, a realization similar to [lensGalleyBits](#) is imaginable. In this case the reader is taken to a new page where can can show the regular o2r platform's user interface.

[Home](#) / [Archives](#) / [Vol 1 No 1 \(1\): issueTitle](#) / [Articles](#)

test o2r

author1

RC

Published
2019-08-14Issue
[Vol 1 No 1 \(1\): issueTitle](#)Section
Articles

Abstract

test

Platform &
workflow by
OJS / PKP

Mockup 4.1.1: View of an ERC for a reader - article view (idea 2)

o2r
[back to journal](#)

SHOW PDF
INSPECT
CHECK
MANIPULATE

INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis

Francesco Dottori
European Commission, Joint Research Centre, Ispra, Italy

Rui Figueiredo
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Mario L. V. Martina
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Daniela Molinari
Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy

Anna Rita Scorzini
Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy

02 Dec 2016

Abstract

Methodologies to estimate economic flood damages are increasingly important for flood risk assessment and management. In this work, we present a new synthetic flood damage model based on a component-by-component analysis of physical damage to buildings. The damage functions are designed using an expert-based approach with the support of existing scientific and technical literature, loss adjustment studies, and damage surveys carried out for past flood events in Italy. The model structure is designed to be transparent and flexible, and therefore it can be applied in different geographical contexts and adapted to the actual knowledge of hazard and vulnerability variables. The model has been tested in a recent

main.Rmd ▾

```

---
title: "INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis"
author:
- affiliation: "European Commission, Joint Research Centre, Ispra, Italy"
  name: "Francesco Dottori"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Rui Figueiredo"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Mario L. V. Martina"
- affiliation: "Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy"
  name: "Daniela Molinari"
- affiliation: "Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy"
  name: "Anna Rita Scorzini"
licenses:
code: CC-BY-3.0
data: CC-BY-3.0
text: CC-BY-3.0
date: "02 Dec 2016"
output: html_document
doi: 10.5194/nhess-16-2577-2016

```

There is no data to display

Mockup 4.1.2: View of an ERC for a reader - ERC view (idea 2)

SHOW PDF
INSPECT CHECK MANIPULATE

INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis

Francesco Dottori
European Commission, Joint Research Centre, Ispra, Italy

Rui Figueiredo
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Mario L. V. Martina
Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy

Daniela Molinari
Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy

Anna Rita Scorzini
Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy

02 Dec 2016

Abstract

Methodologies to estimate economic flood damages are increasingly important for flood risk assessment and management. In this work, we present a new synthetic flood damage model based on a component-by-component analysis of physical damage to buildings. The damage functions are designed using an expert-based approach with the support of existing scientific and technical literature, loss adjustment studies, and damage surveys carried out for past flood events in Italy. The model structure is designed to be transparent and flexible, and therefore it can be applied in different geographical contexts and adapted to the actual knowledge of hazard and vulnerability variables. The model has been tested in a recent

main.Rmd

```

***
title: "INSYDE: a synthetic, probabilistic flood damage model based on explicit cost analysis"
author:
- affiliation: "European Commission, Joint Research Centre, Ispra, Italy"
  name: "Francesco Dottori"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Rui Figueiredo"
- affiliation: "Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy"
  name: "Mario L. V. Martina"
- affiliation: "Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Milano, Italy"
  name: "Daniela Molinari"
- affiliation: "Dipartimento di Ingegneria Civile, Edile-Architettura e Ambientale, Università degli Studi dell'Aquila, L'Aquila, Italy"
  name: "Anna Rita Scorzini"
licenses:
code: CC-BY-3.0
data: CC-BY-3.0
text: CC-BY-3.0
date: "02 Dec 2016"
output: html_document
doi: 10.5194/nhess-16-2577-2016

```

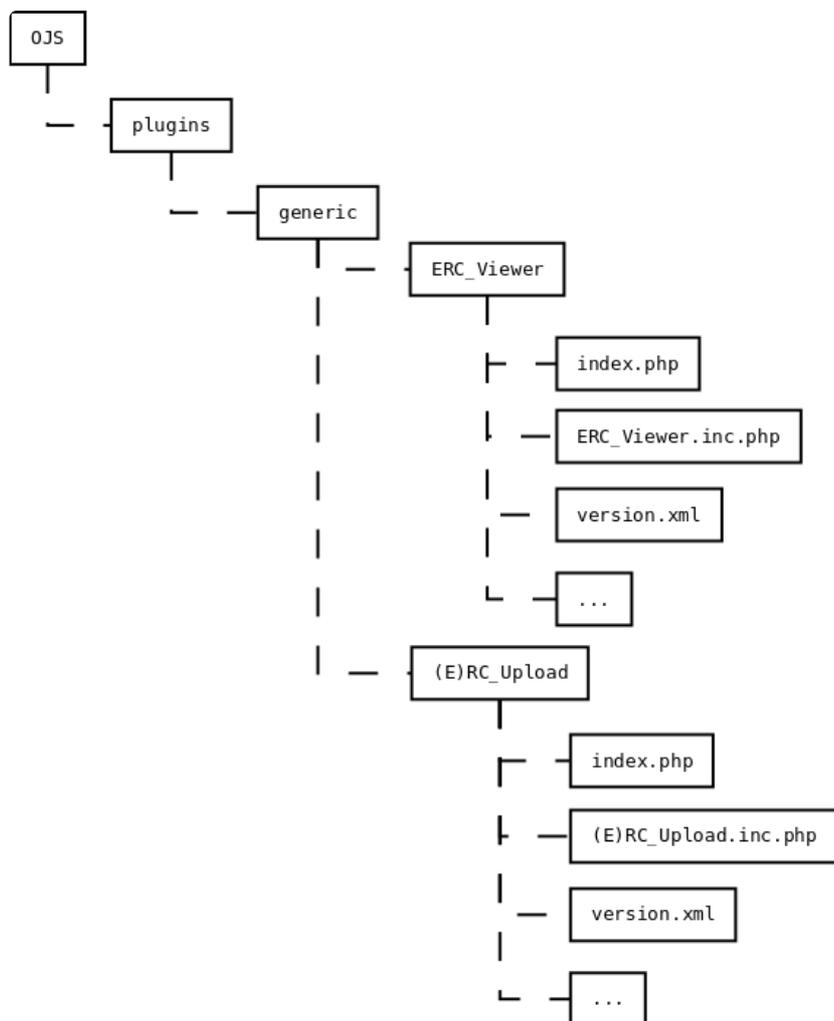
There is no data to display

Mockup 4.2: View of an ERC for a reader (idea 2)

In both cases the user has all possibilities concerning reading the PDF of the ERC and manipulating its figures and tables. In the first case we preserve the journal's branding at the top of the page, which might be desirable for editors and publishers. In the second case we only have the default o2r UI which might be easier to integrate as a standalone page.

Plug-in structure

We sketched a structure for our ERC plug-in. The plug-in consists of two parts, one for the examination of ERCs and one part for the creation/upload of ERCs. The **plug-in category** or type probably needs to be a "generic" plug-in to realise the deep integration of ERC into many different pages of OJS.



Plug-in structure of (E)RC in OJS

Conclusion

We hope this post gives you a good impression of our plans. As you may have noticed, some of the features we plan to implement for ERCs might also be interesting for OJS users who just want to upload multiple files, for journals who want to support other types of [research compendia](#), or for an OJS maintainer who wants to allow a Markdown based workflow. We can imagine several plug-ins could be extracted from the ERC plugin [as described above](#), depending on time left in our schedule and interest by other OJS users/developers. *What do you think?*

Please do not hesitate to comment on this blogpost with your ideas and questions, either below or in [related thread in the PKP Community Forum](#). We would be pleased to learn about your ideas and receive your feedback.

Markus Konkol defends PhD Thesis

11 Oct 2019 | By Daniel Nüst

Markus Konkol successfully defended his PhD thesis, “*Publishing Reproducible Geoscientific Papers: Status quo, benefits, and opportunities*”, today Friday Oct 11 at the Institute for Geoinformatics ([fgi](#)) at University of Münster ([WWU](#)).

Congratulations Markus on completing this important step in your career!

Dr. rer. nat. Markus Konkol is pictured with his Mentor Prof. Dr. Christian Kray and the examination committee: Jun. Prof. Dr. Judith Versteegen, Prof. Dr. Edzer Pebesma, Prof. Dr. Carsten Kessler and Prof. Dr. Harald Strauß.



Markus has been a core o2r team member since the project's start in January 2016. He lead the development of the user interface for creating and examining reproducible research and conducted comprehensive reproduction studies as well as several user studies and surveys, successfully connecting the o2r project with the needs of the geoscience communities. **His work** contributes great insights on the technical and individual challenges - the status quo in the geosciences - as well as incentives and solutions for making research more open and reproducible. His concept and implementation of *bindings* demonstrate a novel groundbreaking method for exposing the true core and value of research outputs that reach beyond geoscience applications and impact transparency, understandability, and discoverability. Markus is a welcome **advocator and speaker** on reproducible research in the geosciences at local and international events. The o2r project is fortunate that he will continue to take down barriers for openness and reproducibility and push towards better science.

Follow [@MarkusKonkol](#) on Twitter and [@MarkusKonk](#) on GitHub.

o2r on tour: eLife Sprint and JupyterHub/Binder workshop

10 Sep 2019 | By Daniel Nüst, Markus Konkol

This week, the o2r team was *on tour*. We put our o2r tasks aside for a few days to interact with and contribute to the awesome Open Science/publishing/research community.

Markus and Daniel were two of the fortunate few who were invited to participate in the **eLife Innovation Sprint 2019**. Thanks eLife! eLife is a non-profit Open Access publisher **with a mission to innovate** and push scholarly communication, peer review, and publication of reproducible articles to new heights. The **#eLifeSprint** is a two-day event and brings together scientists, developers, designers, architects, thinkers, community leaders, publishers, and early career researchers to come up with relevant challenges and promising ideas for the scientific community. It took place for the second time in Cambridge, UK, where eLife's headquarter is located, in the welcoming **Cambridge Junction**. Just like last year, the event was excellently organised and run by eLife staff.

And that's a wrap for **#eLifeSprint 2019**!

A huge thank you to everyone involved for making this event so productive and fun! You've all been amazing!

♥📷 pic.twitter.com/lAwi3pgaNY

— eLife Innovation (@eLifeInnovation) September 5, 2019

After introductions and pitching ideas, the participants formed into project groups and spent ~1.5 days on realising a first prototype. You can learn about the results in the **"time to shine"** presentation and on social media under **#eLifeSprint #timetoshine** : an Open Science card game, a user interface for generating citation files for software, extracting data from text such as the used instruments, a prototype for discovering preprints from authors with underrepresented backgrounds, or a template project for running a journal on GitHub, to name just a few. Daniel and Markus really enjoyed the event and contributed with their developer skills (containers, UI development, eating cake) to several projects.

#TimeToShine: Ankit, Stephen, Daniel have made a UI prototype on GitHub and Docker Hub, worked on UI development for Binder, written case studies and more principles for CODECHECK, as well as helping others with Docker projects at the **#eLifeSprint** pic.twitter.com/YJWrsvqWkN

— eLife Innovation (@eLifeInnovation) September 5, 2019

Team **#SoftwareCitation** ready for **#TimeToShine #eLifeSprint @eLifeInnovation @MarkusKonkol** Sarthak, me **@eScienceCenter** Jen **@ELIXIREurope** , Melissa **@eLife** , Sarala **@datacite** pic.twitter.com/jxjCDLFRIs

— Mateusz Kuzak (@matkuzak) September 5, 2019

While it was a little disappointing that Markus' idea of a JavaScript image comparison library (hopefully more on that soon!) did gain attention but did not end up in a team, the sprint was a great occasion to give back to the community, to broaden the horizon beyond the o2r project, to make new acquaintances, and to get to know potential collaborators. *And we did all that!*

After the **#eLifeSprint**, Daniel hopped on a plane to Oslo, Norway, to participate in a **Binder/BinderHub/MyBinder.org/JupyterHub** event generously organised by **Simula**. The event allowed long-term collaborators to meet in person, some for the first time, for some effective joint work. Participants happily hacked away on their own or formed discussion groups on specific topics for a few hours before taking on a new challenge. Ten to twelve developers of diverse backgrounds filled a hotel meeting room and turned coffee and delicious catering into pull requests, issues, and hackpads with new ideas and solutions in the Binder/Jupyter universe. It was a great experience to get to know the friendly faces and delightful personalities behind GitHub usernames. Daniel enjoyed participating in the discussions and picking the brains of the core developers of BinderHub and repo2docker, and the maintainers of mybinder.org. He was able to contribute a **few pull request** to **repo2docker** and enjoyed the discussions on future directions of the core tool in the Binderverse, such as a new user interface (a must to make BinderHub even more like magic), pinning the repo2docker version (a must for reproducibility) and re-enabling composability for all **supported configurations** (a must for many users).

My kind of crowd

♫ <https://t.co/q8fvlnhQnEL>

— Binder Team (@mybinderteam) September 8, 2019

Thanks to all participants for making the meeting so much fun and educational. Daniel's participation will surely help to pave the way for a Binder-powered scalable infrastructure for the [o2r pilots](#) and for [CODE CHECK](#). You can learn more about the numerous tasks tackled in the sprint in this HackMD pad: <https://hackmd.io/N-uffNhvRdOgt1OvTuoq5w?view>

Why PDFs are not suitable for communicating (geo)scientific results

28 Aug 2019 | By Markus Konkol

In 2016, Dottori et al. published [a paper](#) about a flood damage model. The model calculates the damage costs caused by a flood event, e.g., for repairing buildings or cleaning. This model is based on a number of parameters, such as **flow velocity** and **flood duration**. In the paper, the authors discuss a scenario in which a flood has a velocity of 2m/s and a duration of 24 hours. The resulting damage costs are shown in a figure and also alternative values are discussed in the text. This is where the paper format, i.e. a PDF file, is limited. A mere format change does not help - a static HTML rendering has the same issues. Describing within the article text how changes to the parameter set affect the damage costs might be possible but is surely a daunting and time-consuming task. Authors need to find the right words to briefly describe these changes, and readers need to imagine how the results change.

Wouldn't it be nice if readers, while reading the article, could also simply change the parameters in order to see how the figure changes? We recently published an article on how to achieve this by "[Creating Interactive Scientific Publications using Bindings](#)".

A **binding** describes which source code lines and data subsets were used to produce an individual computational result, such as a figure, table, or number in the text. A binding explicitly refers to single parameters in the code which influence the result. By specifying a user interface widget (e.g. a slider) for a parameter, a binding can then be used to create an interactive figure.

Ok, cool, that sounds just awesome, but how does it look like? Let's check both perspectives, the author who creates a binding, and the reader who uses the interactive figure. Just four steps are needed to create a binding for an interactive figure:

1. Specify the result, e.g. "Figure 3"
2. Mark the plot function in the code that creates the figure. From that plot function we automatically extract all relevant code lines, at least we plan to do so since this feature is currently under development.
3. Mark the parameter that should be made interactive, e.g. "duration" or "velocity"
4. Configure the user interface widget, e.g. a slider That's how authors can create interactive figures easily. Please note that we did not yet fully implement the functionality for specifying data subsets.

The readers' view was integrated into the implementation discussed in an article published last year ("[In-depth examination of spatio-temporal figures](#)") and the previous [blog post](#). The left side shows the static version of the paper. On the right side, readers

can use the slider to change the two parameters velocity and duration. The changes are immediately reflected in the figure. Since it might be difficult to spot differences, we also implemented a simple view to compare two figures created through parameter manipulation by the reader.

Such explorable papers are the next generation of scholarly communications. Being able to provide interactive figures is beneficial for authors, who can explain visually how changes to the parameters affect the figure, and for readers, who better understand complex models. They are also a sign of quality for the analysis workflow, because they demonstrate that all pieces (data, software) needed to create the figure are encapsulated in the Executable Research Compendium ([ERC](#)) on which the bindings are based.

By the way, we also presented the paper at the Engineering Interactive Computing Systems Conference 2019 in Valencia. Of course, the slides are available online on [Zenodo](#).

4+1 quick incentives of open reproducible research

15 Jul 2019 | By Markus Konkol

A few months ago, o2r team member Markus published the article “In-depth examination of spatiotemporal figures in open reproducible research” in the journal [Cartography and Geographic Information science](#). Our goal was to identify a set of concrete incentives for authors to publish open reproducible research, and for readers to engage with it. Based on semi-structured interviews, a focus group discussion, and an online survey with geoscientists, we summarised the incentives in a four-step workflow for readers who work with scientific papers (see figure below). Let’s see what these four workflow steps are who their **+1** is.

Discovery

By having all materials available in a publicly accessible way, we obtain additional capabilities to search for scientific papers which go beyond today’s keyword-based search engines. The materials underlying a paper include a bunch of information which can be extracted automatically (see [o2r-meta](#)) and put on display (see [geospatial data science badges](#)) to improve discovery. You were wondering how to use a specific software library in your R code in practice? Just search for papers with computations based on that library. Spatial information, temporal properties, models, parameters - this all becomes searchable which is good for readers, and findable which is good for the impact of authors.

Inspection

Once researchers found a suitable paper, they can continue with inspecting it. Parallel to reading the actual text of the paper, they can inspect the underlying source code and data. This is of particular interest for reviewers who want to check how the authors achieved the results reported in the paper. By the way, more and more reviewers [reject papers](#) reporting on computational results that do not contain code or data - Think about it! Again, this step is not only beneficial for readers and reviewers but also for the authors who can make their research workflows more reusable resulting in a higher research impact.

Manipulation

Many results in scientific papers are based on computational analyses. These calculations often include parameters which were set in a specific way by the author of the article. For example, a model that computes the damage costs caused by a flood strongly depends on the flow velocity (see [Dottori et al., 2016](#)) of the water. It is difficult to show in static papers, how changes to the flow velocity affect the final damage costs. One idea to solve this issue is an interactive figure. Readers and Reviewers can, for example, use a slider to change the parameter value interactively.

Substitution

Finally, other researchers can substitute, for instance, the original dataset by an own compatible dataset. This opportunity not only makes other researchers’ life easier as they can reuse existing materials, but might also bring citations, co-authorships, and cooperations for the original author.



+1

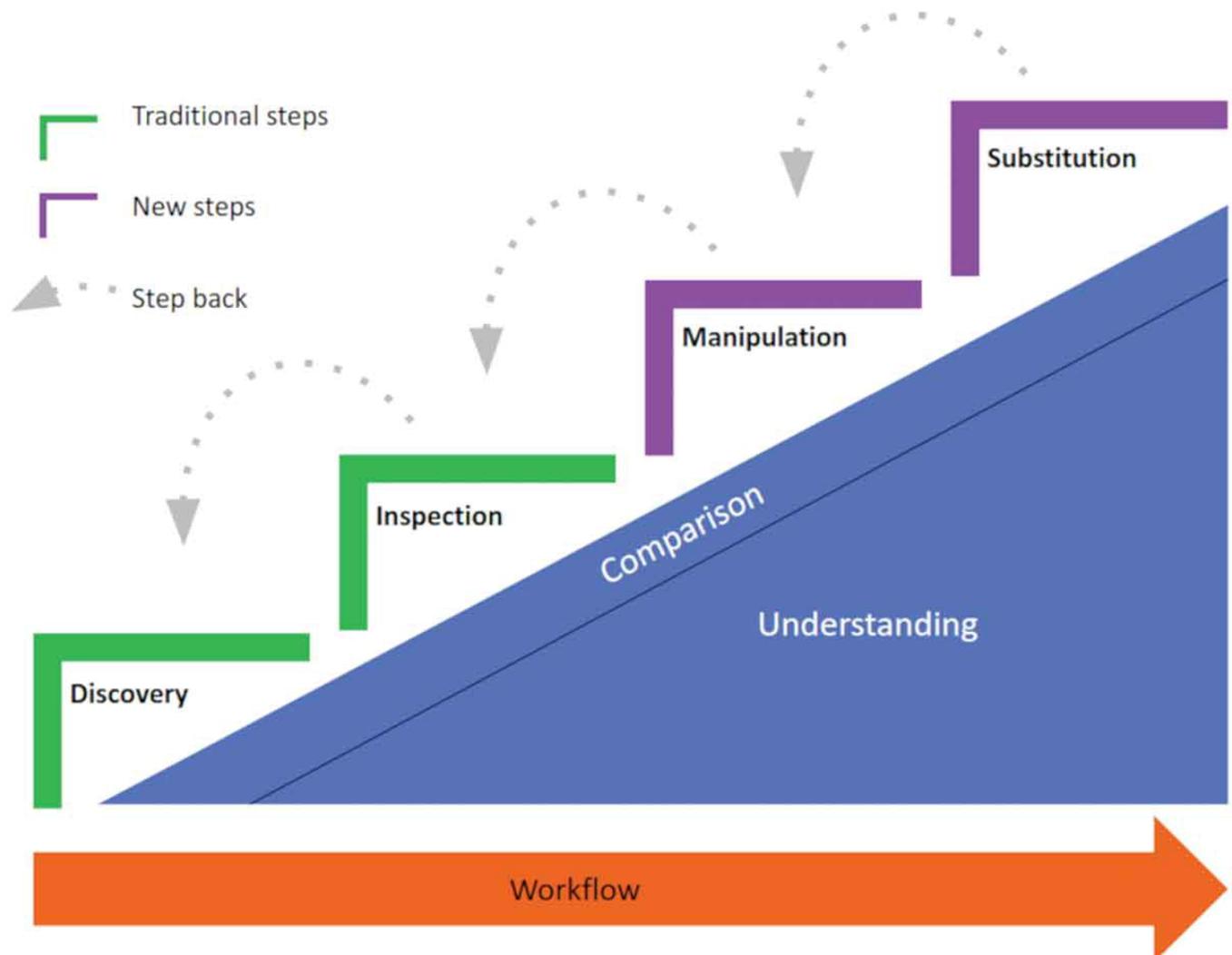
So who is this workflow steps’ +1?

It’s **understanding**.

In the paper, we argue that each of the steps contribute to a reader’s understanding in a better way than traditional papers could do. Already during the inspection phase, researchers get to know about spatio-temporal properties, used functions and so on.

During inspection, they can see how the authors produced a specific figure, experience the data from the analysts perspective, and finally understand how the authors came to their conclusions. By manipulating parameters, readers and reviewers can comprehend better how the model actually works. Substituting datasets provides insights into the applicability to other settings and evaluates robustness of an approach. A key requirement for the realization of understanding is being able to compare, for example, the original figure with one resulting from parameter manipulation.

You think that this was nice to read but difficult to realize? Correct, it is. And that is why the o2r team works hard to make the five incentives easier to achieve and received funding for two more years.



Reproducible Research and Geospatial Badges at AGILE 2019 conference in Limassol

01 Jul 2019 | By Daniel Nüst

Last week our team member Daniel went to Asia (or not?) to help a European conference with the transformation towards reproducible research. *How?*

The 2019 edition of the annual conference of the Association of Geographic Information Laboratories in Europe's (AGILE) took place in Limassol, Cyprus. It was excellently organised at the Cyprus University of Technology and consisted of a pre-conference day of workshops and three days of talks and posters across the full breadth of GI Science.

On the first day, Daniel contributed to the organisation of the third workshop in the “Reproducible Research @ AGILE Series of workshops”. Adjusting the scope of the workshop after the first two iterations, the participants learned first about the basics of reproducibility before being split up into a “beginners” and “advanced” group. The former continued with practical experiences in reproducing a tailored small manuscript with data and code, while the latter took on real world papers in a reproduction sprint. Starting only with a DOI, the participants skimmed real articles for practical instructions and shared how far they got after only 30 minutes. The results were mixed, as it could be expected, but the lessons that could be drawn were already very educational and could be connected directly with concrete steps towards reproducibility.

#agileconf2019 starting today in * Limassol with workshops. @f_ostermann is kicking things off at our #rragile19 workshop on reproducible research! #openscience #reproducibleresearch pic.twitter.com/SecPmEF16z

— Daniel Nüst (@nordholmen) 17. Juni 2019

After lunch, the groups joined again for getting to know the *AGILE Reproducible Paper Guidelines*. The guidelines were developed in online collaboration and a recent expert meeting at TU Delft (see [report](#)). They require authors to be transparent about the underlying building blocks of their work by adding a *Data and Software Availability* section. Beyond this minimal requirement of transparency, the guidelines intent to nudge authors towards higher degrees of reproducibility with concrete steps and recommendations for both data and software. The steps are illustrated by examples from the GI Science domain. [Leave your feedback about the guidelines](#) OSF! The ensuing discussion about the challenges, opportunities, and ethics of reproducible research made clear the participants were serious on their way to becoming experts in RR. They continued on this path in the final session, in which both groups took on the role of an author and applied practices of Open Science and reproducible research. Find all workshop material online at <https://osf.io/d9kcr/>.

Besides the workshop, the RR@AGILE team advertised and sought feedback on the guidelines in many small discussions and with a [dedicated poster](#). The feedback will be incorporated into a first release of the guidelines in the coming weeks, just in time for the call for papers for the [next AGILE conference in Chania, Crete!](#) The RR@AGILE team is proud that the AGILE council and next year's organising team support a transformation towards reproducible research publications and looks forward to working with authors, reviewers and organisers to making the move a success.

On the second day of the conference, Daniel presented the short paper “*Guerrilla Badges for Reproducible Geospatial Data Science*”. The paper is based on the work of a project seminar at the Institute for Geoinformatics from 2017, which explains the long list of co-authors. The article demonstrates how and what kind of novel badges can be created based on executable research compendia (ERC) and how they can be distributed on the web. The full postprint of the peer-reviewed article is available on [EarthArXiv](#) and it contains links to the related software projects.

Just presented short paper at #agileconf2019 - Thanks the #SDI session participants & fun questions. Slides: <https://t.co/N1OMj3iNjv> Paper (postprint w/ DOI pending): <https://t.co/gvkalccHt> Reproduction package: <https://t.co/SC4i99pk5l> #badges pic.twitter.com/Uyv8jyIAUS

— Daniel Nüst (@nordholmen) 18. Juni 2019

As always, AGILE was a delightful conference with many engaging discussions which may have started more collaborations to foster reproducible research. Daniel also continued the text analysis of all AGILE papers for this year's conference.

Obviously this needs advertising, as no one spotted the error in the "trends" analysis: Increase in "reproducibility" keywords can be largely awarded to mine and [@pjkedron](#)'s papers. Still lots of data, algorithms, and processing going on - clear need for reproducibility! pic.twitter.com/W4FMhtm3as

— Daniel Nüst (@nordholmen) 27. Juni 2019

A small rise in "reproducibility" terms can be traced to a couple of articles on the topic. Yet the stronger trend prevails: AGILE papers talk about data, processing, and algorithms - so the transformation for more transparency and reproducibility continues to be relevant.

Find the [full analysis online on RPubs](#) and see the [source code](#) on GitHub.

o2r2 @ Conquire Workshop

27 Jun 2019 | By Markus Konkol

Now that we have two more years to work on open reproducible research (see our [last blog post](#)), there is also some space for an exchange with related projects and to explore potential new collaborations. We were thus very happy to receive an invitation from the [Conquire](#) project at the University of Bielefeld for the workshop [on data quality and reproducibility](#) (03.04.2019). Conquire started about the same time as o2r and strives for similar goals, i.e. assisting scholars in making their research results reproducible and reusable. The workshop was located at the Center for Interdisciplinary Research in a very nice room that looked a bit like the United Nations headquarter - so it was good practice for the bigger goals we have in mind.



[Prof. Dr. Philipp Cimiano](#) gave the first talk of the day. He presented the general Conquire approach which focuses on storing all materials in a GitLab repository and running checks with the help of continuous integration based on [Jenkins](#). Researchers can thus create an incremental publication where each git commit triggers an automatic validation process. They also had a promising number of use cases. However, similar to us, they struggled a bit with the amount of effort needed from authors to make research reproducible.

[Christian Pietsch](#) then gave a quick introduction into versioning tools such as GitLab and which benefits users get. I particularly liked his answer to the question from the audience if the Conquire approach is also feasible with licensed software: Use free open source software! It's that easy.

Afterwards, Conquire team member Fabian Herrman talked about their validation approach by using continuous integration ([Slides](#)). They check, for example, if all files are available (including readme and license) and convey the result in two ways: First, by assigning a badge to the repository and second, by emailing the author of the repository.

The following talks were about **F**indable, **A**ccessible, **I**nteroperable, and **R**eusable data principles (by Silvia Wissel and Amrapali Zaveri) and the Jupyter Notebook, which was used in the context of history science by Malte Vogl. One benefit of Jupyter notebooks he mentioned stuck with us: it is also readable when the base software does not exist anymore. This is also one of the

essential advantages of the Dockerfiles and R Markdown documents used in our executable research compendia (ERCs).

Last but not least, we were allowed to present our approach and what we plan to achieve in the next two years. The slides are available online: <https://zenodo.org/record/2628278>.

o2r2 - Putting ERC into practice

15 Apr 2019 | By Daniel Nüst

The o2r project's journey continues.

On April 1st 2019 the o2r team started into a new phase ("o2r2"). In the next 30 months we plan to put our prototypes to the test with *real articles*, of course not without considerably improving them beforehand.

As detailed in the University of Münster's press releases ([English](#), [German](#)), we are fortunate to collaborate with publishers to achieve the following objectives:

- Use ERCs for actual scientific publications in **pilot studies** with original research manuscripts in a scholarly peer review
- **Eliminate barriers** for using ERCs as part of a publishing process
- **Evaluate** concept and pilots with user studies and monitoring to understand the costs and benefits of ERC-based authoring, publishing, and reading

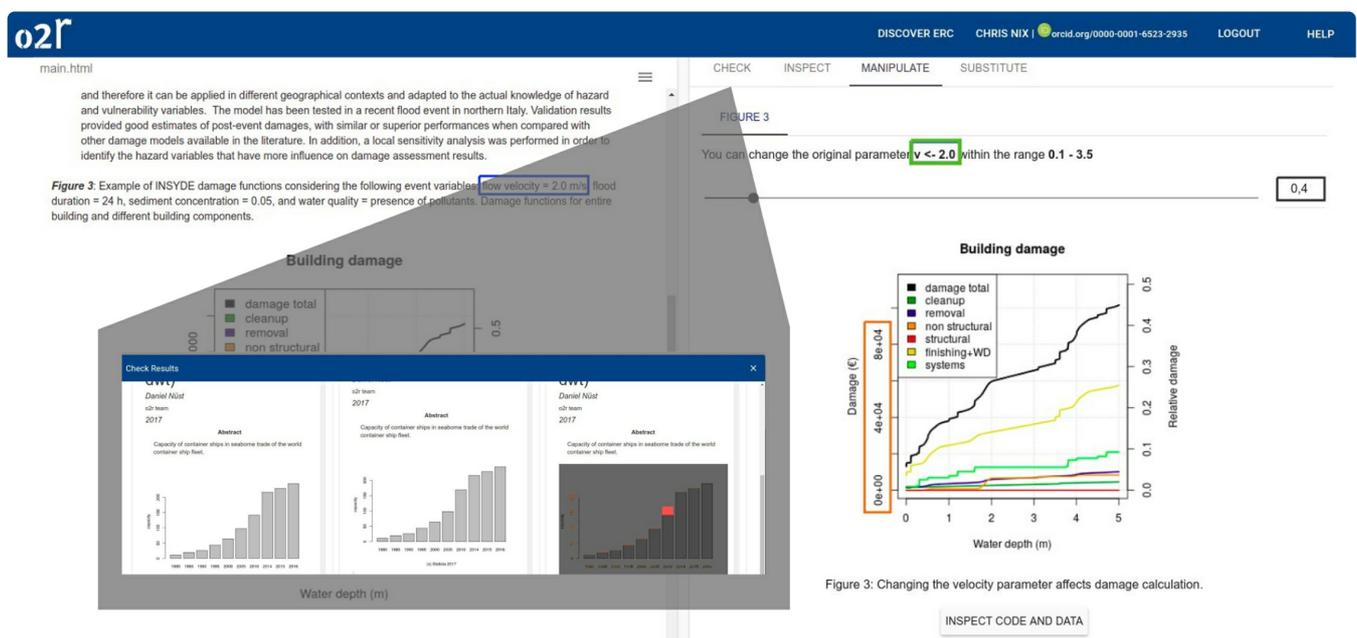


Figure 3: Changing the velocity parameter affects damage calculation.

This is how a scientific publication can be presented in the future: on the left is the original publication, on the right the readers can work with the research data themselves. © o2r - based on F. Dottori et al./ Nat. HazardsEarth Syst. Sci.

As the official press statement was edited for brevity, we'd like to use the opportunity to extend on it here:

The project "Opening Reproducible Research II" (o2r2) will be supported by the German Research Foundation (DFG) under the umbrella of the programme for Library Services and Information Systems (LIS). Building on the results of the predecessor project, it will test, evaluate, and further develop solutions for improving reproducibility of research results in practice over the next 30 months. The o2r project will conduct three pilot studies. The Open Access publisher *Copernicus Publications* and a large commercial publisher could be won for the project to conduct two pilot studies in the form of special issues for scientific journals, in which classic articles will be enriched with interactive and transparent analyses. In a third pilot, an open source software for the publication of scientific journals, Open Journal Systems by the Public Knowledge Project (OJS by PKP), which is widely used in the scientific community, will be connected to the o2r reproducibility service and piloted at the ULB Münster together with researchers and students from the geosciences. Developments for OJS will be contributed to the global and national communities, including [OJS-de.net](#). These pilots will be accompanied by various evaluations: with the help of authors, reviewers, and students, the transformation potential of reproducible scientific articles will be investigated. A focus lies on analyses based on geospatial data and the programming language R. Furthermore, the operation provides relevant insights into the costs and efforts for contemporary publishing of and interaction with data-based scientific research. All specifications and tools of o2r2 are published under free licenses and where possible are realised as contributions to existing Open Source projects. The project's developments and results provide building blocks and concepts for a future infrastructure for enhanced scholarly communication and academic publications.

■
We realise these goals are ambitious, but look forward confidently to work with the Open Science and Open Source communities to make them reality.

As a first action, o2r team member Daniel attended the [EGU General Assembly in Vienna last week](#) to start the conversation about the pilot with EGU journals with journal editors. With the support of Copernicus staff, we distributed flyers about the planned Virtual Special Issue to journal editors of the Copernicus journals. You can read more about the plans [on the Pilots page](#). Since we are very early in the project, it comes only as a little setback that Daniel could only speak to a handful of editors. We plan to intensify our outreach to journals and editors later this year, when the first working prototypes can tell the story much more convincing than a paper leaflet can - *isn't that what executable interactive publications are all about?*

Besides reaching out to editors, discussing [Open Access & preprints](#), and putting [research software](#) on the map, Daniel also presented [a poster](#) on *packaging research*, with many fun interactions and discussions.

One more (hopefully) [#betterposter](#) at [#EGU19](#) on advantages on packaging research using containers and community standards, powered by [@o2r_project](#) Find poster and abstract at <https://t.co/fKjQ57uuhp> pic.twitter.com/tjeJ582ZRq

— Daniel Nüst (@nordholmen) 12. April 2019

Archiving a Research Project Website on Zenodo

24 Feb 2019 | By Daniel Nüst

The o2r project website's first entry [Introducing o2r](#) was published [1132 days ago](#). Since then we've published short and long reports about events the o2r team participated in, advertised new scholarly publications we were lucky to have accepted in journals, and reported on results of workshops organised by o2r. But there has also been some original content from time to time, such as the extensive articles on [Docker and R](#), which received several updates over the last years (some still pending), on the [integration of Stencila and Binder](#), or on [writing reproducible articles for Copernicus Publications](#). These posts are a valuable output of the project, and contribute to the scholarly discussion. Therefore, when it came to writing a report on the project's activities and outputs, it was time to consider the [preservation](#) of the project website and blog. The website is built with [Jekyll](#) (with Markdown source files) and [hosted with GitHub pages](#), but GitHub may disappear and Jekyll might stop working at some point.

So how can we archive the blog post and website in a sustainable way, without any manual interference?

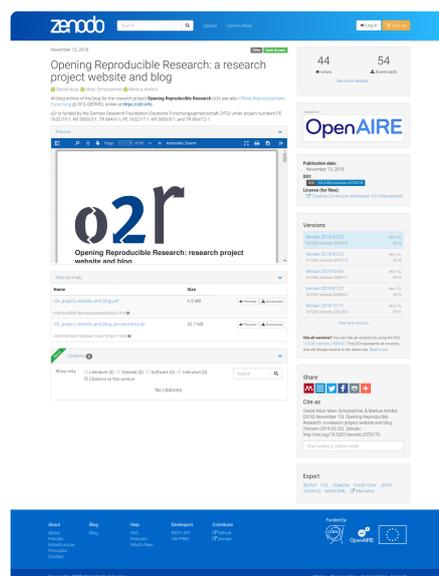
Today's blog post documents the steps to automatically deposit the sources, the HTML rendering, a PDF rendering, and the whole git repository in a new version of a [Zenodo deposit](#) with each new blog post using [Zenodo's DOI versioning](#). The PDF was especially tricky but is very important, because the format is established for archival of content, while using the Zenodo API was pretty straightforward. We hope the presented workflow might be useful for other websites and blogs in a scientific context. It goes like this:

1. The [Makefile](#) target `update_zenodo_deposit` starts the whole process with `make update_zenodo_deposit`. It triggers several other make targets, some of which require two processes to run at the same time:
2. Remove previously existing outputs ("clean").
3. Build the whole page with Jekyll and [serve](#) it using a local web server.
4. Create a PDF from the whole website from the local web server using `wkhtmltopdf` and the special page `/all_content`, which renders *all* blog entries in a suitable layout together with an automatically compiled list of author names and all website pages, unless they are excluded from the menu (e.g. manual redirection/shortened URLs) or excluded from "all pages" (e.g. the 404 page, blogroll, or publications list).
5. Create a ZIP archive with the sources, HTML rendering and PDF capture.
6. Run a Python script to upload the PDF and ZIP files to Zenodo using the [Zenodo API](#), which includes several requests to retrieve the latest version metadata, check that there really is a new blog post, create a new deposit, remove the existing files in the deposit, upload the new files, and eventually publish the record.
7. Kill the still running web server.

For these steps to run automatically, the [Travis CI](#) configuration file, `.travis.yml` (link to commit where Travis CI configuration was removed in favour of...) the GitHub action configuration `deposit.yml`, installs the required software environment to conduct all above steps during each change to the main branch. A secure environment variable for the repository is used to store a Zenodo API key, so the build system can manipulate the record. The first version of this record, including its description, authors, tags, etc., was created manually.

So what is possible now? As pointed out in the citation note at the bottom of pages and posts, the Digital Object Identifier (DOI) allows referencing the whole website or specific posts (via pages in the PDF) in scholarly publications. Manual archival from a local computer is still possible by triggering the same make target. As long as Zenodo exists, readers have access to all content published by the o2r research project on its website.

There are no specific next steps planned, but there's surely room for improvement as the current workflow is pretty complex. The post publication date is the trigger for a new version, so changes in a page such as [About](#) or in an existing post requires a manual triggering of the workflow (and commenting out the check for a new post) or wait for the next blog entry. The created PDF [could be made compliant with PDF/A](#). The control flow could also be implemented completely in Python instead of using multiple files and languages; a Python module might even properly manage the system dependencies. Though large parts of the process are not limited to pages generated with Jekyll (the capturing and uploading), it might be effectively wrapped in a standalone Jekyll [Jekyll plugin](#), or a combination of a Zenodo plugin together with the (stale?) [jekyll-pdf plugin](#)? *Your feedback is very welcome!*



R&R Workshop at SPARC

16 Feb 2019 | By Daniel Nüst

The capabilities of containerisation and the concept of the [Executable Research Compendium](#) form the basis for o2r's [reproducibility service](#). But the demonstration how latest technology may support a more open and transparent scholarly publication alone is only one half the battle. Breakthroughs in tools and infrastructure must be accompanied by outreach activities to highlight the need for opening reproducible research to all stakeholders (scientists, editors, publishers, funding agencies). That is why I was extremely glad to join some of the most renowned researchers of geography and GI Science at the "[Replicability and Reproducibility Workshop](#)" in Tempe, Arizona, on February 11 and 12, 2019. The event was organised by the Spatial Analysis Research Center ([SPARC](#)) at Arizona State University ([ASU](#)).

Daniel Nüst from the University of Munster presenting at our Replicability and Reproducibility in Geospatial Research: Open is not enough for reproducibility! pic.twitter.com/PwpVmwdQ7b

— SPARC at ASU (@SPARC_ASU) 11. Februar 2019

The events featured four full talks. I was invited to go first and report on the activities of o2r as well as the [Reproducible AGILE](#) conference series and initiative for developing new submission and reviewing guidelines. The expectations were high, but after three years of intense work by the o2r team, the allotted time was easily filled. After introducing *challenges* which disrupt scholarly publication practices, I reported on *observations* made by [our own surveys](#) and others on reproducibility, including the [two reproduction campaigns](#) let be the o2r team. The painted picture unsurprisingly left a lot of room for improvement, for which the talk provided technical as well as organisational *approaches*.

Just finished my talk on challenges, observations and approaches towards reproducible research at the SPARC workshop [#RandR #geospatial @ASU](#) Find the slides and material (including speaker notes in the source file with all the stuff I forgot!) at <https://t.co/cSRZ1DkwhO> pic.twitter.com/FXjNj0hT2K

— Daniel Nüst (@nordholmen) 11. Februar 2019

In the second talk, ASU's own [Peter Kedron](#) took a step back and surveyed the existing definitions of replicability and reproducibility. Peter then excellently connected and extended these terms with the intricacies and specifics of geospatial sciences. The technical and theoretical groundwork was laid, so the discussion following the first talks set the bar for the remainder of the workshop quite high. Many critical and thoughtful comments were made and viewpoints shared. One of the main take-home messages for me was that while geography/GI Science/related disciplines may take advantage of the hard lessons learned in other domains (which faced a 'replication crisis'), the uniqueness of geography as a science that always had to deal with uncertainty and *context* may also contribute a unique perspective on replicability and reproducibility. It was great to see that the topic of reproducibility is widely acknowledged as a relevant challenge and the interest to initiate improvement was unilateral.

The first day continued with *lightning talks*. As could be expected, the diverse backgrounds (eScience, ecology, political geography, ...) let to a very useful diversity in topics and perspectives. Afterwards the participants split up into three groups to tackle technical, organisational, and institutional aspects of replicability and reproducibility, which gave input for yet another thoughtful debate in the assembly to conclude day one. The discussions continued between old colleagues and new friends during a delightful evening reception and dinner.

Day two kicked off in a similar fashion with talks by [Daniel Sui](#), University of Arkansas, and Esri's [Dawn Wright](#) and Kevin Butler. Again two very different takes on the topic, with valuable new ideas. The following discussion was lively and included potential venues for the newly formed group to continue the collaboration, most importantly to increase the awareness of the topic across all communities working with spatial data. The whole meeting was nicely guided and framed by contributions from ASU's [Mike Goodchild](#) and [Stewart Fotheringham](#). All participants were united in their interest to advance transparency and openness and a realisation that there is a need for action from many different angles, including education and evaluation, if a 'crisis' shall be avoided. Despite some concerns how the topic might be received by critics, the meeting ended in a positive mood of newfound mutual support and of acknowledging the value of the work ahead.

My personal opinion is that the disruptions in science are more pressing than a traditional scholarly approach (organising a special issue for 2020, writing an editorial) can answer. Yet the old-school way may be able to bridge across the divide and different skill-set/mindset/needs between computational/junior/young/technical and theoretical/senior researchers, and is as such worth pursuing. For future discussions, I plan to frame reproducibility as an ideal that is worth striving for and worth to reward (e.g.

in evaluations, during reviews, using badges, in funding schemes), but to be careful with too simple checklists and dos/don'ts, because there will always be corner cases and limitations for specific circumstances. This is a core difference between reproducibility and openness - you can not be a little or partially open, but being almost reproducible is still an important achievement. Reproducibility and replicability will not be helped by whataboutism nor by pointing fingers, but the positive [spirit of preproducibility](#).

Luckily there is no need to echo all insights by talks and during the discussions: the sessions were recorded on video they will be published together with slides and position papers in an OSF project soon. This post will be updated then. [Follow us](#) on Twitter to not miss it. Until then you can take a look at my position paper [on GitLab](#), even the speaker notes in the presentation source file if you dare.

This post would be incomplete without a big *Thank You* to the sponsoring and excellent organisation provided by [Esri](#) and the hosting School of Geographical Sciences and Urban Planning ([SGSUP](#)). I am confident this workshop may spark new collaborations and be able to put replicability and reproducibility on the map for more researchers in geography and related disciplines.

On the last flight back home after a great week with [@SPARC_ASU](#). They brought together a great group of people to talk about [#reproducibleresearch](#). Thank you!

I enjoyed learning more about geospatial specialities with R&R, hiking Arizona, and starting new collaborations [#PhDlife](#)
pic.twitter.com/mb1lu9Nc0Q

— Daniel Nüst ([@nordholmen](#)) [15. Februar 2019](#)

How to increase reproducibility and transparency in your research

04 Feb 2019 | By Daniel Nüst

[This article is cross posted-on the [EGU GeoLog](#).]

Contemporary science faces many challenges in publishing results that are reproducible. This is due to increased usage of data and digital technologies as well as heightened demands for scholarly communication. These challenges have led to widespread [calls](#) for more research transparency, accessibility, and reproducibility from the science community. This article presents current findings and solutions to these problems, including recent new software that makes writing submission-ready manuscripts for journals of [Copernicus Publications](#) a lot easier. While it can be debated if science really faces a [reproducibility crisis](#), the challenges of computer-based research have sparked numerous articles on new [good research practices](#) and their [evaluation](#). The challenges have also driven researchers to develop infrastructure and tools to help scientists effectively write articles, publish data, share code for computations, and communicate their findings in a reproducible way, for example [Jupyter](#), [ReproZip](#) and [research compendia](#).

[Recent studies showed](#) that the geosciences and geographic information science are not beyond issues with reproducibility, just like other domains. Therefore, more and more [journals](#) have [adopted policies](#) on sharing data and code. However, it is equally important that scientists foster an [open research culture](#) and teach researchers how they adopt more transparent and reproducible workflows, for example at skill-building workshops at conferences offered by fellow researchers, such as the EGU short courses, community-led non-profit organisations such as the [Carpentries](#), [open courses for students](#), small discussion groups at research labs, or individual efforts of self-learning. In the light of prevailing [issues of a common definition](#) of reproducibility, [Philip Stark](#), a statistics professor and associate dean of mathematical and physical sciences at the University of California, Berkeley, recently coined the term [preproducibility](#): “An experiment or analysis is preproducible if it has been described in adequate detail for others to undertake it.” The neologism intends to reduce confusion and also to embrace a positive attitude for more openness, honesty, and helpfulness in scholarly communication processes.



“Science should be
‘show me’, not
‘trust me’.”

Before reproducibility must come preproducibility

Instead of arguing about whether results hold up, let's push to provide enough information for others to repeat the experiments, says Philip Stark.

[nature.com](#)

<https://twitter.com/NatureNews/status/999715421208104960>

In the spirit of these activities, this article describes a modern workflow made possible by recent software releases. The new features allow the EGU community to write preproducible manuscripts for submission to the large variety of academic journals published by [Copernicus Publications](#). The new workflow might require hard-earned adjustments for some researchers, but it pays

off because of an increase in transparency and effectivity. This is especially the case for early career scientists. An open and reproducible workflow enables researchers to build on others' and own previous work and better collaborate on solving the societal challenges of today.

Reproducible research manuscripts

Open digital [notebooks](#), which [interweave data and code](#) and can be exported to different output formats such as PDF, are powerful means to improve transparency and reproducibility of research. [Jupyter Notebook](#), [Stencila](#) and [R Markdown](#) let researchers combine long-form text of a publication and source code for analysis and visualisation in a single document. Having text and code side-by-side makes them easier to grasp and ensures consistency, because each rendering of the document executes the whole workflow using the original data. Caching for long-lasting computations is possible, and researchers working with supercomputing infrastructures or huge datasets may limit the executed code to purposes of visualisation using processed data as input. Authors can transparently expose specific code snippets to readers but also publish the complete source code of the document openly for collaboration and review.

The popular notebook formats are plain text-based, like [Markdown](#) in case of R Markdown. Therefore an R Markdown document can be managed with [version control software](#), which are programs for managing multiple versions and contributions, even by different people, to the same documents. Version control provides traceability of authorship, a time machine for going back to any previous “working” version, and online collaboration such as on [GitLab](#). This kind of workflow also stops [the madness of using file names for versions](#) yet still lets authors use [awesome file names](#) and apply domain-specific [guidelines for packaging research](#).

Final.doc <https://t.co/YXJaSacHWu> <pic.twitter.com/4bBDzn7TXt>

— PHD Comics (@PHDcomics) 1. Februar 2017

R Markdown supports [different programming languages](#) besides the popular namesake [R](#) and is a sensible solution even if you do not analyse data with scripts nor have any code in your scholarly manuscript. It is easy to write, allows you to [manage your bibliography](#) effectively, can be used for websites, [books](#) or [blogs](#), but most importantly *it does not fall short when it is time to submit a manuscript article to a journal*.

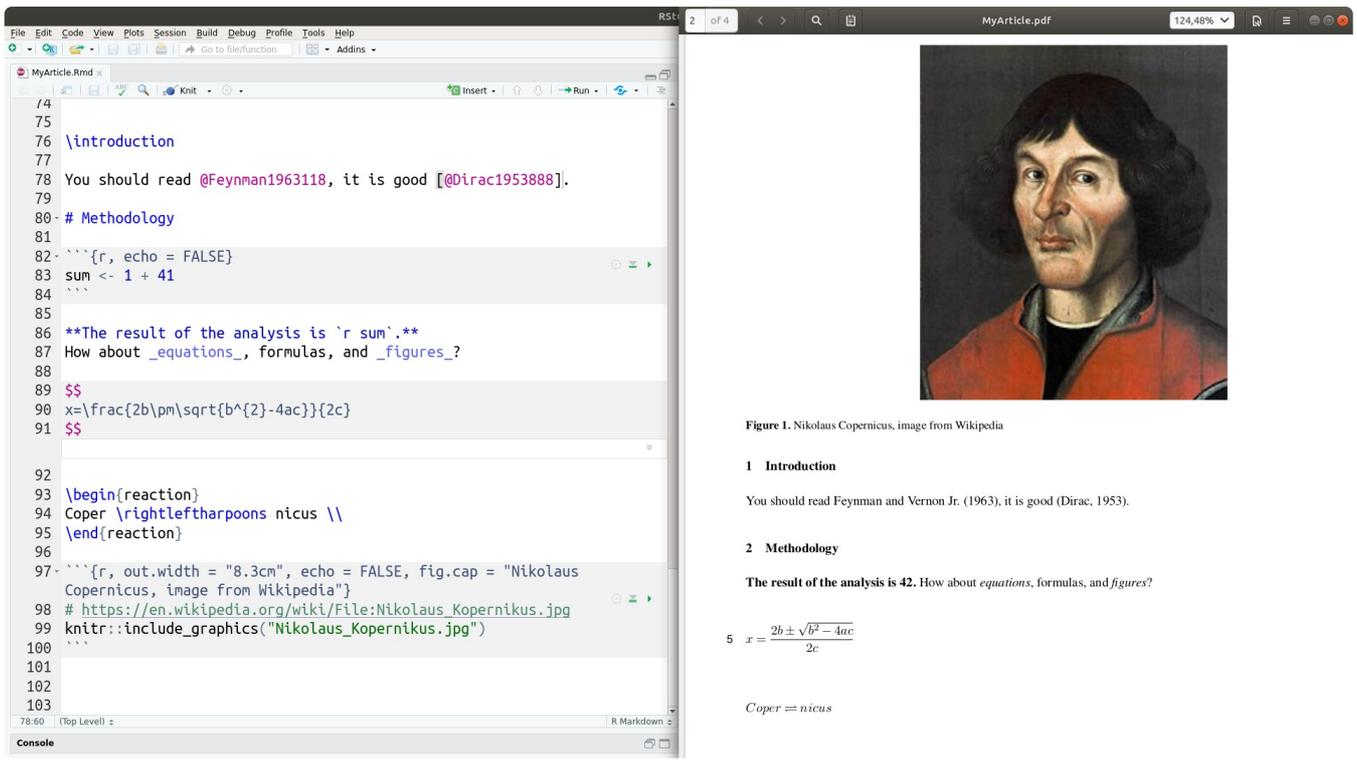
The [rticles](#) extension package for R provides a number of templates for popular journals and publishers. Since version [0.6](#) ([published Oct 9 2018](#)) these [templates include](#) the [Copernicus Publications Manuscript preparations guidelines for authors](#). The Copernicus Publications staff was kind enough to give a test document a quick review and all seems in order, though of course any problems and questions shall be directed to the software's vibrant community and not the publishers.

The following code snippet and screen shot demonstrate the workflow. Lines starting with <#> are code comments and explain the steps. Code examples provided here are ready to use and only lack the installation commands for required packages.

```
# load required R extension packages:
library("rticles")
library("rmarkdown")

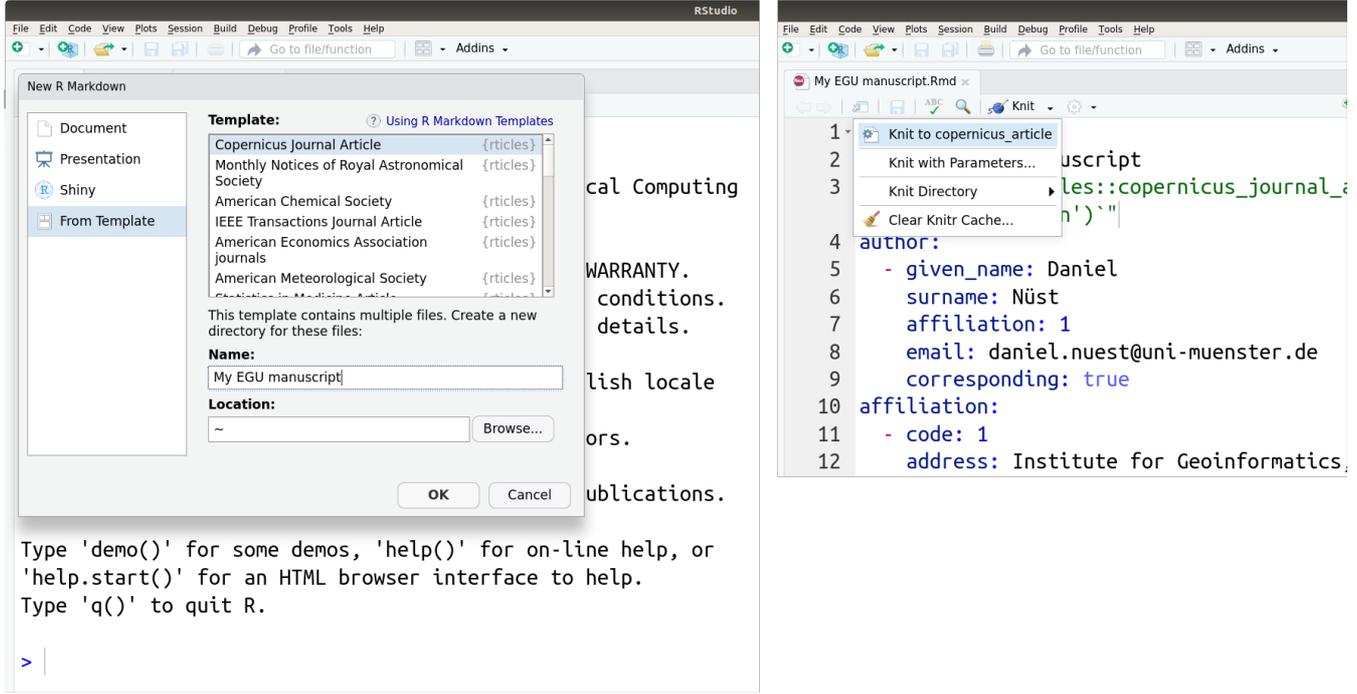
# create a new document using a template:
rmarkdown::draft(file = "MyArticle.Rmd",
  template = "copernicus_article",
  package = "rticles", edit = FALSE)

# render the source of the document to the default output format:
rmarkdown::render(input = "MyArticle/MyArticle.Rmd")
```



The commands created a directory with the Copernicus Publications template's files, including an R Markdown (.Rmd) file ready to be edited by you (left-hand side of the screenshot), a LaTeX (.tex) file for submission to the publisher, and a .pdf file for inspecting the final results and sharing with your colleagues (right-hand side of the screenshot). You can see how simple it is to format text, insert citations, chemical formulas or equations, and add figures, and how they are rendered into a high-quality output file.

All of these steps may also be completed with user-friendly forms when using RStudio, a popular development and authoring environment available for all operating systems. The left-hand side of the following screenshot shows the form for creating a new document based on a template, and the right-hand shows side the menu for rendering, called "knitting" with R Markdown because code and text are combined into one document like threads in a garment.



And in case you decide last minute to submit to a different journal, `rticles` supports many publishers so you only have to adjust the template while the whole content stays the same.

Sustainable access to supplemental data

Data published today **should** be published and properly cited using appropriate [research data repositories](#) following the [FAIR data principles](#). Journals require authors to follow these principles, see for example the [Copernicus Publications data policy](#) or a recent [announcement by Nature](#). Other publishers required, or still do today, to store supplemental information (SI), such as dataset files, extra figures, or extensive descriptions of experimental procedures, as part of the article. Usually only the article itself receives a digital object identifier (DOI) for long-term identification and availability. The DOI [minted](#) by the publisher is not suitable for direct access to supplemental files, because it points to a [landing page](#) about the identified object. This landing page is designed to be read by humans but not by computers.

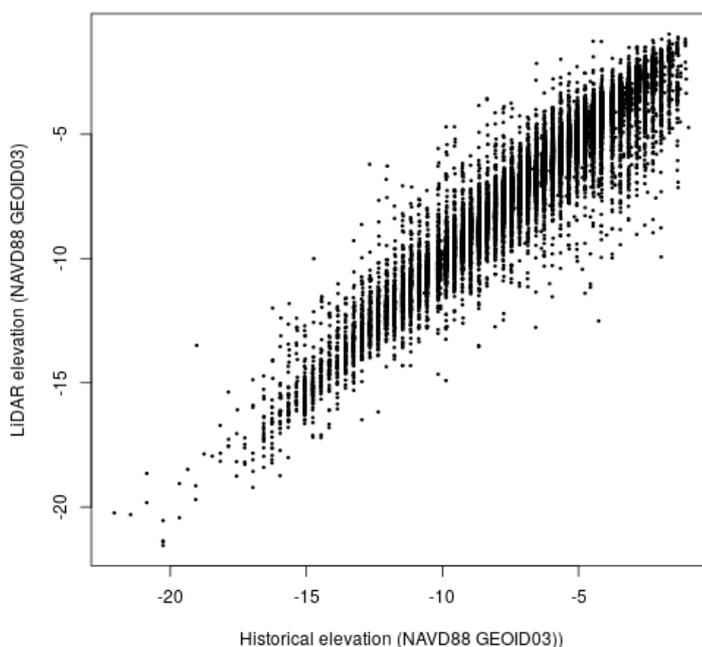
The R package `suppdata` [closes this gap](#). It supports downloading supplemental information using the article's DOI. This way `suppdata` enables long-term reproducible data access when data was published as SI in the past or in exceptional cases today, for example if you write about a reproduction of a published article. In the latest version available [from GitHub](#) (`suppdata` is [on its way to CRAN](#)) the [supported publishers](#) include Copernicus Publications. The following example code downloads a data file for the article “Divergence of seafloor elevation and sea level rise in coral reef ecosystems” by Yates et al. published in *Biogeosciences* in 2017. The code then creates a mostly meaningless plot shown below.

```
# load required R extension package:
library("suppdata")

# download a specific supplemental information (SI) file
# for an article using the article's DOI:
csv_file <- suppdata::suppdata(
  x = "10.5194/bg-14-1739-2017",
  si = "Table S1 v2 UFK FOR_PUBLICATION.csv")
supplemental

# read the data and plot it (toy example!):
my_data <- read.csv(file = csv_file, skip = 3)
plot(x = my_data$NAVD88_G03, y = my_data$RASTERVALU,
     xlab = "Historical elevation (NAVD88 GEOID03)",
     ylab = "LiDAR elevation (NAVD88 GEOID03)",
     main = "A data plot for article 10.5194/bg-14-1739-2017",
     pch = 20, cex = 0.5)
```

A silly plot for article 10.5194/bg-14-1739-2017



Main takeaways

Authoring submission-ready manuscripts for journals of Copernicus Publications just got a lot easier. Everybody who can write manuscripts with a word processor can learn quickly R Markdown and benefit from a preproducible data science workflow. Digital

notebooks not only improve day-to-day research habits, but the same workflow is suitable for authoring high-quality scholarly manuscripts and graphics. The interaction with the publisher is smooth thanks to the LaTeX submission format, but you never have to write any LaTeX. The workflow is based on an established [Free and Open Source](#) software stack and embraces the idea of preproducibility and the principles of [Open Science](#). The software is maintained by an [active, growing](#), and welcoming community of researchers and developers with a [strong connection to the geospatial sciences](#). Because of the complete and consistent notebook, [you](#), a colleague, or a student can easily pick up the work at a later time. The road to effective and transparent research begins with a first step - [take it!](#)

Acknowledgements

The software updates were contributed by [Daniel Nüst](#) from the project [Opening Reproducible Research \(o2r\)](#) at the Institute for Geoinformatics, University of Münster, Germany, but would not be able without the support of Copernicus Publications, the software maintainers most notably [Yihui Xie](#) and [Will Pearse](#), and the general awesomeness of the R, R-spatial, Open Science, and Reproducible Research communities. The blog text was greatly improved with feedback by EGU's [Olivia Trani](#) and Copernicus Publications' [Xenia van Edig](#). Thank you!

References

- [Announcement: FAIR data in Earth science](#), *Nature*, 565(7738), 134–134, doi:10.1038/d41586-019-00075-3, 2019.
- Baker, M.: [1,500 Scientists Lift the Lid on Reproducibility](#), *Nature*, 533(7604), 452–454, doi:10.1038/533452a, 2016.
- Barba, L. A.: [Terminologies for Reproducible Research](#), ArXiv:1802.03311 [Cs], February 9, 2018.
- Fanelli, D.: [Opinion: Is Science Really Facing a Reproducibility Crisis, and Do We Need It To?](#) *Proceedings of the National Academy of Sciences*, 115(11), 2628–2631, doi:10.1073/pnas.1708272114, 2018.
- Gil, Y., David, C. H., Demir, I., Essawy, B. T., Fulweiler, R. W., Goodall, J. L., Karlstrom, L., Lee, H., Mills, H. J., Oh, J.-H., Pierce, S. A., Pope, A., Tzeng, M. W., Villamizar, S. R. and Yu, X.: [Towards the Geoscience Paper of the Future: Best Practices for Documenting and Sharing Research from Data to Software to Provenance](#), *Earth and Space Science*, 3(10), 388–415, doi:10.1002/2015ea000136, 2016.
- Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsson, G., Banks, G. C., Kidwell, M. C., Hofelich Mohr, A., Clayton, E., Yoon, E. J., Henry Tessler, M., Lenne, R. L., Altman, S., Long, B. and Frank, M. C.: [Data availability, reusability, and analytic reproducibility: evaluating the impact of a mandatory open data policy at the journal Cognition](#), *Royal Society Open Science*, 5(8), 180448, doi:10.1098/rsos.180448, 2018.
- Konkol, M. and Kray, C.: [In-depth examination of spatiotemporal figures in open reproducible research](#) *Cartography and Geographic Information Science*, 1–16, doi:10.1080/15230406.2018.1512421, 2018.
- Konkol, M., Kray, C. and Pfeiffer, M.: [Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study](#), *International Journal of Geographical Information Science*, 1–22, doi:10.1080/13658816.2018.1508687, 2018.
- Markowetz, F.: [Five selfish reasons to work reproducibly](#), *Genome Biology*, 16(1), doi:10.1186/s13059-015-0850-7, 2015.
- Marwick, B., Boettiger, C. and Mullen, L.: [Packaging Data Analytical Work Reproducibly Using R \(and Friends\)](#), *The American Statistician*, 72(1), 80–88, doi:10.1080/00031305.2017.1375986, 2017.
- Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Percie du Sert, N., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J. and Ioannidis, J. P. A.: [A manifesto for reproducible science](#), *Nature Human Behaviour*, 1(1), 21, doi:10.1038/s41562-016-0021, 2017.
- Nüst, D., Granell, C., Hofer, B., Konkol, M., Ostermann, F. O., Sileryte, R. and Cerutti, V.: [Reproducible research and GIScience: an evaluation using AGILE conference papers](#), *PeerJ*, 6, e5072, doi:10.7717/peerj.5072, 2018.
- Ostermann, F. O. and Granell, C.: [Advancing Science with VGI: Reproducibility and Replicability of Recent Studies using VGI](#), *Transactions in GIS*, 21(2), 224–237, doi:10.1111/tgis.12195, 2016.
- Pearse, W. D. and A Chamberlain, S.: [Suppdata: Downloading Supplementary Data from Published Manuscripts](#), *Journal of Open Source Software*, 3(25), 721, doi:10.21105/joss.00721, 2018.
- [ReproZip: Computational Reproducibility With Ease](#), F. Chirigati, R. Rampin, D. Shasha, and J. Freire. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 2085-2088, 2016
- Sandve, G. K., Nekrutenko, A., Taylor, J. and Hovig, E.: [Ten Simple Rules for Reproducible Computational Research](#), edited by P. E. Bourne, *PLoS Computational Biology*, 9(10), e1003285, doi:10.1371/journal.pcbi.1003285, 2013.
- Stark, P. B.: [Before reproducibility must come preproducibility](#), *Nature*, 557(7707), 613–613, doi:10.1038/d41586-018-05256-0, 2018.
- Toelch, U. and Ostwald, D.: [Digital open science—Teaching digital tools for reproducible and transparent research](#) *PLOS Biology*, 16(7), e2006022, doi:10.1371/journal.pbio.2006022, 2018.
- Jupyter, P., Bussonnier, M., Forde, J., Freeman, J., Granger, B., Head, T., Holdgraf, C., Kelley, K., Nalvarte, G., Osheroff, A., Pacer, M., Panda, Y., Perez, F., Ragan-Kelley, B. and Willing, C.: [Binder 2.0 - Reproducible, interactive, sharable](#)

- [environments for science at scale](#), in Proceedings of the 17th Python in Science Conference, SciPy., 2018.
- Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L. and Teal, T. K.:[Good enough practices in scientific computing](#) PLOS Computational Biology, 13(6), e1005510, doi:10.1371/journal.pcbi.1005510, 2017.
 - Yates, K. K., Zawada, D. G., Smiley, N. A. and Tiling-Range, G.:[Divergence of seafloor elevation and sea level rise in coral reef ecosystems](#), Biogeosciences, 14(6), 1739–1772, doi:10.5194/bg-14-1739-2017, 2017.

New article published in International Journal of Geographical Information Science

17 Dec 2018 | By Markus Konkol, Daniel Nüst

A few weeks ago, a new journal article written by o2r team member Markus got published. In our [last article](#), we talked about the reproducibility of papers submitted to the AGILE conference. We checked if the papers had materials attached and if these materials were complete. The results were rather unfortunate. In our newest article, we took one further step and tried to *re-run the analyses of articles* which had code and data in the supplements.

Markus Konkol, Christian Kray & Max Pfeiffer (2019). **Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study**, *International Journal of Geographical Information Science*, 33:2, 408-429, DOI: [10.1080/13658816.2018.1508687](https://doi.org/10.1080/13658816.2018.1508687)

The article builds upon our [paper corpus for demonstrating the o2r platform](#) Feel free to distribute this piece of research to whoever might be interested. Feedback is always welcome.

Here is a non-specialist summary:

Recreating scientific data analysis is hard, but important. To learn more about the state of reproducibility in geosciences, we conducted several studies. We contacted over 150 geoscientists who publish and read articles based on code and data. We learned that as readers they often would like to have access to these materials, but as authors they often do not have the time or expertise to make them available. We also collected articles which use computational analyses and tried to execute the attached code. This was not as easy as it sounds! We describe these numerous issues in a structured way and our experiences in this publication. Some issues were pretty easy to solve, such as installing a missing library. Others were more demanding and required deep knowledge of the code which is, as you might imagine, highly time consuming. Further issues were missing materials (code snippets, data subsets) and flawed functionalities. In some cases, we contacted the original authors who were, and this was a positive outcome, mostly willing to help. We also compared the figures we got out of the code with those contained in the original article. Bad news: We found several differences related to the design of the figures and results that deviated from those described in the paper. OK, this is interesting, but why is it important? We argue, a key advantage of open reproducible research is that you can reuse existing materials. Apparently, this is usually not possible without some significant effort. Our goal is not to blame authors. We are very happy that they shared their materials. But they did that with a specific purpose in mind, i.e. making code and data available and reusable for others to build upon that. One incentive in this context is an increased number of citations, one of the main currencies for researchers. To facilitate that, we suggest some guidelines to avoid the issues we encountered during our reproducibility study, such as using Executable Research Compendia (ever heard of them? :)).

INTERNATIONAL JOURNAL OF GEOGRAPHICAL INFORMATION SCIENCE
2019, VOL. 33, NO. 2, 408-429
<https://doi.org/10.1080/13658816.2018.1508687>



RESEARCH ARTICLE

OPEN ACCESS [Check for updates](#)

Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study

Markus Konkol , Christian Kray and Max Pfeiffer
Institute for Geoinformatics, University of Münster, Münster, Germany

ABSTRACT

Reproducibility is a cornerstone of science and thus for geographic research as well. However, studies in other disciplines such as biology have shown that published work is rarely reproducible. To assess the state of reproducibility, specifically computational reproducibility (i.e. rerunning the analysis of a paper using the original code), in geographic research, we asked geoscientists about this topic using three methods: a survey ($n = 146$), interviews ($n = 9$), and a focus group ($n = 5$). We asked participants about their understanding of open reproducible research (ORR), how much it is practiced, and what obstacles hinder ORR. We found that participants had different understandings of ORR and that there are several obstacles for authors and readers (e.g. effort, lack of openness). Then, in order to complement the subjective feedback from the participants, we tried to reproduce the results of papers that use spatial statistics to address problems in the geosciences. We selected 41 open access papers from *Copernicus* and *Journal of Statistical Software* and executed the R code. In doing so, we identified several technical issues and specific issues with the reproduced figures depicting the results. Based on these findings, we propose guidelines for authors to overcome the issues around reproducibility in the computational geosciences.

ARTICLE HISTORY

Received 9 April 2018
Accepted 30 July 2018

KEYWORDS

Open reproducible research
computational research
spatial statistics

1. Introduction

Reproducibility is an essential element of scientific work in general, as it enables researchers to re-run and re-use experiments reported by others. Further benefits of working and publishing reproducibly include increased transparency and more efficient review processes (Gil *et al.* 2016). Despite these advantages, publishing results in a reproducible way is still not common practice (Reichman *et al.* 2011), which is part of the reason why some have proclaimed a 'reproducibility crisis' (Baker 2016). A recent study in economics (Gentler *et al.* 2018) has shown that even when authors make the data and code publicly accessible, it is not guaranteed that readers can successfully reproduce the results published in the paper. On top of that, the inconsistent usage of the terms *reproducibility* and *replicability* within and across disciplines can cause further

CONTACT Markus Konkol m.konkol@uni-muenster.de

Supplemental data for this article can be accessed here.

© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This article has been published under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

eLife sprint: Integrating Stencila and Binder

21 Nov 2018 | By Daniel Nüst

This article reports on a project, integrating Stencila and Binder, which started at the eLife Innovation Sprint 2018. It has been cross-posted on multiple blogs (eLife Labs, Stencila, Jupyter). We welcome comments and feedback on any of them!

eLife, an open science journal published by the non-profit organisation eLife Sciences Publications from the UK, hosted the first eLife Innovation Sprint 2018 as part of their Innovation Initiative in Cambridge, UK: “[.] a two-day gathering of 62 researchers, designers, developers, technologists, science communicators and more, with the goal of developing prototypes of innovations that bring cutting-edge technology to open research communication.” One of the 13 projects at the excellently organised event was an integration of Binder and Stencila.

This article reports on the project’s inception, building blocks, achievements at the sprint weekend, and work conducted in the months following the sprint. **Today, Binder has first class Stencila support.** You can open Stencila documents from any online code repository on mybinder.org with the click of a single button. Just try out the example below:

launch binder

The idea and the sprint team

The eLife Innovation Sprint started with brief introductions by all participants. Some of them prepared pitches for projects ideas, which quickly got little group discussions going. One table at the sprint attracted a few people with an interest in containerisation technology for research applications. Many ideas were floated and a helpful exchange around existing solutions and tools took place. When it was time to find a concrete task, two of the sprinters identified a worthwhile technological problem as their challenge for the next 1.5 days and the project “Jupyter+DAR compatibility exploration” started. Min from the Simula Research Laboratory, Norway, is a core developer of Binder and related tools. He was interested to get to know the Stencila project and explore the possibilities of having alternative user interfaces on Jupyter Hub. Daniel from the o2r project at the Institute for Geoinformatics, Germany, works on reproducible computations in the geosciences and had a keen interest in learning more about the Binder platform. They were joined remotely by Nokome, the initiator and one of the developers of Stencila.

The building blocks

Stencila Desktop is an office suite for reproducible research documents. It allows scientists to use languages like R and Python within familiar and intuitive word processor and spreadsheet user interfaces. By doing so, it aims to lower the barriers to reproducible research for those with little or no software development skills. At the same time, Stencila aims to make it easy for researchers versed in software development to collaborate with their colleagues without having to switch from R or Python. Stencila Desktop is built upon Texture, an editor for scientific content, which uses the Dar file format. Dar is an extension of the JATS publishing format which has been designed for reproducible research publications. It aims to serve researchers using computational methods for data, and publishers using digital workflows for publication and preservation of scholarly journals.

Binder makes it simple to generate reproducible computing environments from code repositories. The online service mybinder.org is the most prominent example for a platform based on the Binder project, a part of Project Jupyter. A user can run a Jupyter Notebook and other environments for their research projects, which are published in online repositories (e.g. GitHub or GitLab, see binder examples). In the spirit of the Unix philosophy, Binder combines several Open Source tools to achieve this goal: [repo2docker](https://github.com/jupyter/repo2docker), for generating Dockerfile s and building Docker images from software projects, [JupyterHub](https://jupyter.org) for executing a Docker image and user-facing web portal in a cloud environment, and [BinderHub](https://binderhub.org) for gluing the above together.

A **Dockerfile** is a human- and machine-readable recipe for setting up a computational environment, which is just fancy words for saying “installing and configuring software”. Dockerfile s are used by the popular Docker container software. They can be built into an executable image, which is portable between host computers. These properties make containers very interesting for capturing and sharing research involving data and software.

While containers have become a commodity for developers, researchers still struggle to grasp and control the complexity of computational environments. This is where the two building blocks join: **Running Stencila as part of a Binder helps researchers to communicate their work openly, to collaborate effectively with other scientists, and to ensure a high quality and transparency of their workflow and findings.**

The challenge

As Min and Daniel formulated their goals in the sprint project form the project was heavy with software titles:

Their goal was “[...] to connect them so that users can edit reproducible documents (DAR files) as part of a Binder project” with the following objectives: (i) understanding DAR [Dar Format], (ii) launching Stencila Editor on Binder (potentially not launching anything else, i.e. w/o the Jupyter Notebook start page), and (iii) repo2docker support for DAR files. The project was also part of the [Mozilla Global Sprint 2018](#), see [mozilla/global-sprint#317](#).

The solution

It took more than just the 1.5 days in Cambridge to really fulfil this challenge. First we describe the crucial breakthroughs that were actually made at the sprint, then the updates that happened until today.

Sprint breakthrough

Min and Daniel started by taking a close look at an existing solution, namely the integration of RStudio based on [nbrsessionproxy](#), i.e. the “Notebook R session proxy”. They learned two things:

1. a Jupyter notebook extension can be used to add a menu item to the Jupyter UI
2. a component is needed to route the traffic between the browser-based user interface and the server-side software

The first attempts utilised Binder’s feature of manually defining a bespoke [Dockerfile](#) (see [a first attempt](#)) and later also a [postBuild script](#) to install and configure all software. It was Daniel’s first task to transfer the first finding for Stencila. After setting up a local development environment and learning Jupyter/Binder, it just needed small adjustments to selected files from [nbrsessionproxy](#) to achieve this (see [commit](#) from the second day):



Min took on the second task while at the same time figuring out what parts of Stencila we really needed, and how to glue them together. He wrote a hard-wired proxy using [Python](#) and added some [JavaScript/HTML files](#) to serve Dar files and the Stencila UI itself.

Connecting Stencila to Jupyter kernels

Stencila has “execution contexts” (the equivalent of Jupyter’s “kernels”) for R, Python, SQL, Javascript (in the browser), and Node.js. Execution contexts differ from kernels in a number of ways including code dependency analysis and returning execution results as data values. Both of these are necessary for the reactive, functional execution model of Stencila.

We could install these execution contexts in the Docker image. However, Stencila also has a [JupyterContext](#) which acts as a bridge between Stencila’s API and Jupyter kernels. So, since the base [jupyter/minimal-notebook](#) image already has a Jupyter kernel for Python installed, we decided to use that. This did mean however, that some of the reactive aspects of the Stencila UI won’t work as expected.

We included the [stencila-node](#) Node.js package in the Docker image which provides the [JupyterContext](#) as well as a [NodeContext](#)

(for executing Javascript) and a `SQLiteContext` (for executing SQL) .

We first used Stencila's development build to run the JavaScript app using `node make -w -s -d /our/own/dir` , but struggled a bit to configure the file storage, i.e. the `dar-server` , to use the directory we want to, and to run it in a full path configured by us instead of `make.js` starting the `dar-server` relative to `__dirname` . Eventually we ended up implementing our own minimal JavaScript module (i.e. an `npm` package) that run (i) the `dar-server` and (ii) a static file server for the app using the distribution files (i.e. the `dist` directory). This gave us control of the paths and let us get rid of complex development features (e.g. `substance-bundler`).

We also made our own version of `app.js` , removing the virtual file storage (`vfs` , used to seamlessly integrate examples) and instead defaulting to a file system (`fs`) storage, because that is what is needed for Jupyter. In the same line, we built our own `index.html` (based on `example.html`) to serve as the entry page. This allowed us to directly render a single Dar document instead of a listing of examples and to use our own `app.js` . Relevant path configurations comprised the local storage paths *well* as the URLs used by the client, accessing the `dar-server` through the `nbserverproxy` .

At the end of the first day, the wiring was all there so we could open a repository and the Stencila document was shown! But the interactive execution of code cells did not work yet :-/.

Thanks to an international time-zone-difference-powered "overnight" contribution, Min and Daniel got a big surprise on Friday morning: Nokome [added the Stencila Node.js host for Jupyter execution context support](#) so that Python cells could be executed by connecting to the Jupyter Kernel (which of course was already there in the container). In doing so, he returned the "surprise" he had [when learning about the project](#) . The added "host" provides the single gateway for code cell contents to be forwarded to the respective execution contexts. Nokome showed everything works with the obligatory screenshot:

The image shows a GitHub commit history and a comment. The commit history includes:

- `nokome` added some commits on 11 May
 - Add the Stencila Node.js host for Jupyter execution context support (4f29c53)
 - Fix URL for host (1b4c44a)

The comment by `nokome` on 11 May is titled "Obligatory screenshot :)" and contains a screenshot of a Jupyter notebook interface. The notebook shows a code cell with the following Python code:

```
import sys
import time
'Hello this is Python %s.%s and it is %s' % (sys.version_info[0], sys.version_info[1],
time.strftime('%c'))
```

The output of the code cell is: "Hello this is Python 3.6 and it is Fri May 11 04:43:29 2018".

Below the comment, another commit is visible: "Fix the py-jupyter example to use Jupyter kernels and make default" (d0e81c3).

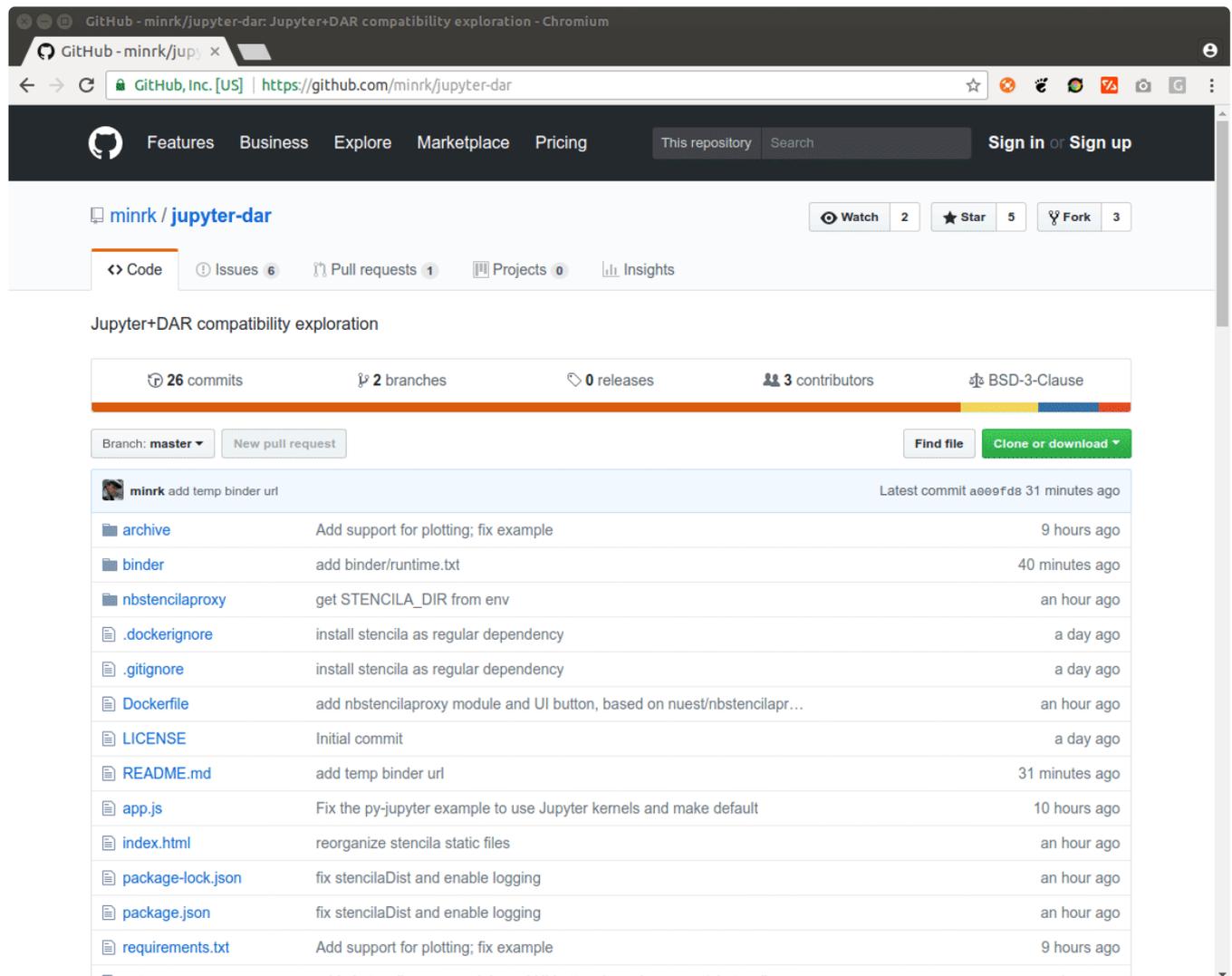
Since you can run any commit in a Binder, you can also try out that particular state [from the repository](#) yourself:

[launch binder](#)

The second day of the sprint involved many iterations of improvements, including changes to `repo2docker` . These updates could not simply be thrown upon mybinder.org, so Min set up a test server for the demonstrations at the sprint's final day. Daniel continued his work on supporting R code cells, but albeit [small contributions](#) to the Stencila codebase, he could not complete this task in time.

The sprint ended with [presentations by all projects](#), some of which are still continuing today, for example [Appstract](#), [Citation Gecko](#), [PREreview](#), or [Octopus](#). The results were truly awesome, ranging from ambitious concepts, case studies, design concepts, completely new tools with great UX design, to technical demonstrators. It's an easy guess where on the spectrum our project can be placed... You're invited to catch a glimpse of the sprint, its results, and the people behind all of it on Twitter under the hashtag [#eLifeSprint](#) and read the [project roundup](#).

The following screencast and Binder link show the **status at the end of the sprint** a Stencila document could be opened on a bespoke Binder deployment and the contained Python code could be interactively edited. The code is re-run on the server and the figure updated.



[[Watch video on YouTube](#)]

launch [binder](#)

You can view the Python example document by appending `?archive=py-jupyter` to the URL of Stencila in the Binder, e.g. <https://hub.mybinder.org/.../stencila/?archive=py-jupyter>.

Consolidation

A couple of weeks after the sprint, a second less intensive development period started. Daniel continued his work on adding support for the R context, and also managed to get plain Python cells running (see pull requests [#15](#) and [#16](#)). Min restructured the whole project and gave it the name it still bears: `nbstencilaproxy` - a Jupyter notebook server extension and proxy for Stencila.

The projects GitHub repository holds a **Python module** with the Jupyter notebook server and “non-server” extensions of the same name, and a **bundled JavaScript module** (of the same name).

The Python module allows proper versioned installation, dependency management, and installation from an established software

repository. It takes care of the plumbing between the user interface and the services in the background, so that the binder is viewable over one port in the browser, while the many different background components run on their own ports. The “no server” extension adds the “Stencila session” menu entry and conveniently lives in the same directory structure as the server extension.

The JavaScript module manages the required JavaScript dependencies and provides an well-defined structure for the code files. It serves the Dar document and provides access to the Stencila host (see above).

While complex at first sight, this modularity hopefully makes maintenance for future developments and new collaborators easier. For now, the JavaScript module and its installation are bundled with the Python module instead of being published independently, because the code and configuration is very much specific to the Jupyter integration.

Min also extended `repo2docker` with [automatic detection of Dar documents](#) (as part of a “build pack”), so that no configuration is required for most common use cases. As with most Binder repositories, a user could simply open a Dar document on Binder and trust the required environment to provide all required software.

On July 20th the `nbstencilaproxy` [was published on PyPI](#) and on August 1st, the new developments [made it into `repo2docker`](#) . Soon after Stencila was available for all users on mybinder.org, which was a great achievement for a little project started at a community sprint. However, the big announcement was still not made, since some things were still hard-wired and, for example, to use R, the author of a repository [had to manually add a configuration file](#) although the information that R is needed is already part of the Stencila document.

The last mile

In October, Daniel took on the final tasks of writing this blog post and fixing the R installation, including the automatic detection of the required execution contexts of a given Dar document. This included some [housekeeping](#) in `nbstencilaproxy` and more importantly new [contributions to `repo2docker`](#) (thanks to [Tim](#) for review and help) to (i) properly detect the languages used in a Stencila document, (ii) extend the R build pack to install R if it is used in a Stencila document, and (iii) add documentation and tests. `repo2docker` now detects Dar documents based on their `manifest.xml` files and uses the location of the first discovered one as the base directory to start Stencila. If a Dar manifest is found, then `nbstencilaproxy` is installed and the languages are extracted from code cells from the document. Authors can install extra dependencies using the [repo2docker's existing mechanisms](#).

Daniel also created a few **example repositories** to provide a starting point for users. Thankfully the binder team generously welcomed [the changes to mybinder.org](#) and and the examples to the [binder examples organisation](#) on GitHub. The following repositories contain single or multiple Stencila documents with code chunks in different programming languages.

<https://github.com/binder-examples/stencila-py> contains Python code cells, using both the Jupyter and plain Python execution contexts:

launch binder

Stencila Project

https://hub.mybinder.org/user/binder-examples-stencila-py-qh60tvu5/stencila/?token=SlXrjfeLSj2W8AgSpxPZ1Q

Abstract

Introduction

Markdown cells

Code cells

Metadata

Citations and references

References

```
'Hello this is Python via Jupyter %s.%s and it is %s' % (sys.version_info[0], sys.version_info[1], time.strftime('%c'))
```

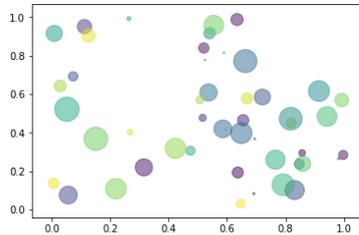
"Hello this is Python via Jupyter 3.6 and it is Mon Nov 12 17:10:35 2018"

Stencila also support Jupyter code cells that produce plots. The cell below produces a simple plot based on the example from [the Matplotlib website](#). Try changing the code below (for example, the variable `N`).

```
import numpy as np
import matplotlib.pyplot as plt

N = 50
N = min(N, 1000) # Prevent generation of too many numbers :)
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2 # 0 to 15 point radii

plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```



We are currently working on supporting [Jupyter's magic commands](#) in Stencila via a bridge to Jupyter kernels.

py-jupyter.ipynb

<https://github.com/binder-examples/stencila-r> contains R code cells and two plots:

launch binder

Stencila Project

https://hub.mybinder.org/user/binder-examples-stencila-r-eszhsgoq/stencila/?token=N1KPQjd6SycEoXeoEFVdw

(although that is not working 100% right now). Other options are placed in a comment at the top of the cell so that they are preserved (and eventually will be used to apply those options within the R execution context).

Abstract

Introduction

Code chunks

Figures

References

Figure 1

```

# : fig.width=7,fig.height=6
s <- min(1000, n)
x <- runif(s)
y <- x + runif(s)
z <- y + rnorm(s)
hist(z, breaks=40, col=hsv(0.6, 0.9, 1), xlab="Value", main="")

```

Figure title

rmarkdown.Rmd

<https://github.com/binder-examples/stencila-multi> demonstrates how to access specific Dar projects if multiple are found within a repository.

launch binder

In each case you can see the available execution environments by clicking on the icon in the bottom right corner.

One of the cool features of Stencila are the reactive cells, as demonstrated in a tweet following the feature release:

Thanks to @nordholmen working on @stencila support for <https://t.co/Zlj6FrYgBw> you now have reactive cells with Python code on @mybinderteam! Give it a go <https://t.co/ToluQPq0Fy> pic.twitter.com/Wjyf1kiH9B

— Tim Head (@betatim) 12. November 2018

Summary and outlook

Thanks for reading so far! This blog post is a [long planned](#) write-up of the history of the tool and decisions mostly relevant to developers, but also an demonstration of the power that the Open Source and Open Science community can foster. Many people are working together on the (technological) challenges of science today towards full research transparency and reproducibility, even if we use computers to an unprecedented level. Many small contributions on “side projects” such as these can make a difference, and connecting these two great projects hopefully helps to solve some problem in science down the road.

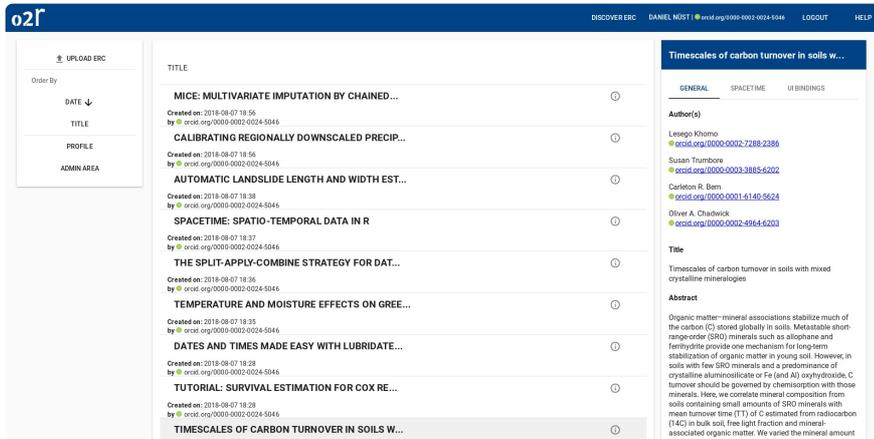
What's next? While there are no concrete plans, there are of course some ideas listed on the [project's issue tracker](#), such as an automatic Jupyter notebook to Dar conversion when there is no Dar archive in a repository. In any case you can keep an eye out on GitHub for projects being [tagged stencila and binder](#) and join the public [Stencila](#) and [binder](#) chats to stay in touch or get help. We look forward to see scientists using [nbstencilaproxy](#) for communicating their work and new challenges that come with it.

#eLifeSprint-ers @minrk and @nordholmen are working to connect #JupyterNotebooks / #Binder with DAR / #Texture / @Stencila, so that users can edit reproducible documents as part of a Binder project <https://t.co/2GoGNydsMx> (@mybinderteam @ProjectJupyter @_substance) pic.twitter.com/sZ8bbE9SsM

Demo server update

14 Aug 2018 | By Daniel Nüst

We've been working on demonstrating our reference-implementation during spring an managed to create a number of example workspaces. We now decided to publish these workspaces on our demo server.

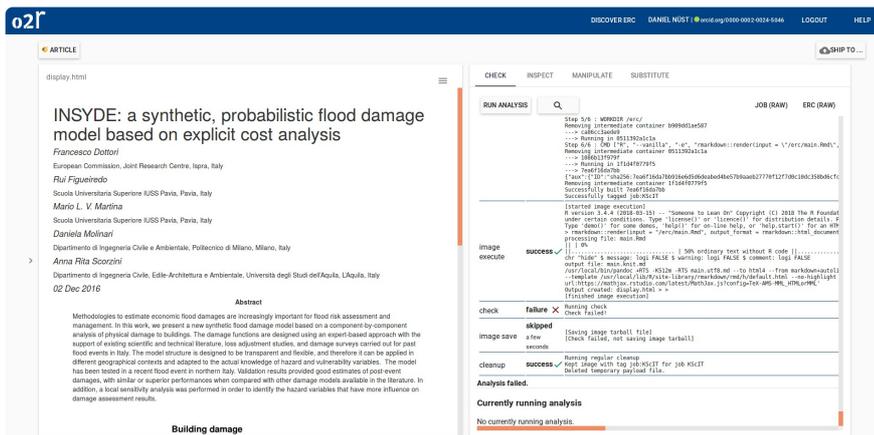


Screenshot 1: o2r reference implementation listing of published Executable Research Compendia. The right-hand side shows a metadata summary including original authors.

The papers were originally published in [Journal of Statistical Software](#) or in a [Copernicus Publications](#) journal under open licenses. We have created an R Markdown document for each paper based on the included data and code following the [ERC specification](#) for naming core files, but only included data, an R Markdown document and a HTML display file. The publication metadata, the runtime environment description (i.e. a [Dockerfile](#)), and the runtime image (i.e. a Docker image tarball) were all created during the ERC creation process without any human interaction (see the used [R code for upload](#)), since required metadata were included in the R Markdown document's front matter.

The documents include selected figures or in some cases the whole paper, if runtime is not extremely long. While the paper's authors are correctly linked in the workspace metadata (see right hand side in [Screenshot 1](#)), the "o2r author" of all papers is o2r team member Daniel since he made the uploads. You can find all publications on his author page (this is the link you definitely want to try out!):

<https://o2r.uni-muenster.de/#!/author/0000-0002-0024-5046>

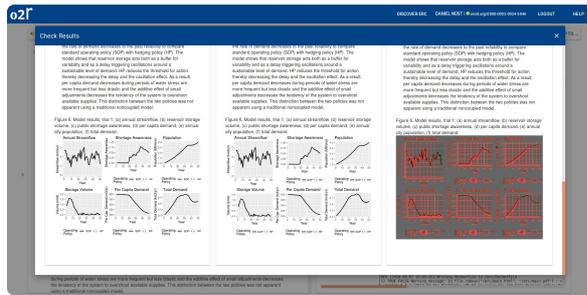


Screenshot 2: o2r reference implementation ERC detail page for compendium [SLVIQ](https://o2r.uni-muenster.de/#!/erc/SLVIQ). The link "Article" in the top left corner leads to the original article, the "magnifying glass" button takes you to a core feature: the reproduction result.

You can get to the original publication by clicking the "Article" button in the top left corner (see [Screenshot 2](#)). The workspaces demonstrate a variety of issues and are a great source for future work on architecture and implementation. Here are some examples of the power of a reproducible research service and publishing platform:

- The ERC for "Tidy Data" by Hadley Wickham completes the reproduction successfully, so no differences between the uploaded and reproduced HTML file were found! You can even download the image tarball (just bear with our demo - not production - server it takes some time).
- The ERC for "A question driven socio-hydrological modeling process" by Garcia et al. "fails" due to differences in the created

figure. A human can now judge if these differences are minor, or the author can try to tweak rendering parameters to fix this.



- A demo ERC with randomised output shows how things can really go wrong. Feel free to click “Run Analysis” and see how the differences changes with each execution.

If you want to go through the creation process yourself, register on the platform (this requires a short manual interaction by us) and upload one of selected workspaces, which you can find in our public demo share at <https://unimuenster.sciebo.de/s/G8vxQ1h50V4HpuA> (just look for zip files starting with `corpus_...`). Please take care to choose appropriate licenses and be aware that we might remove compendia from the demo platform without prior notice.

We welcome your feedback on Twitter, in the reference implementation GitHub project, or in the comments below.

New article published in PeerJ

13 Jul 2018 | By Daniel Nüst

Today a new journal article lead by o2r team member Daniel was published in the journal PeerJ:

article peer-reviewed **Reproducible research and GIScience: an evaluation using AGILE conference papers** by *Daniel Nüst, Carlos Granell, Barbara Hofer, Markus Konkol, Frank O. Ostermann, Rusne Sileryte, Valentina Cerutti*
PeerJ. 2018. doi: [10.7717/peerj.5072](https://doi.org/10.7717/peerj.5072)

The article is an outcome of a collaboration around the AGILE conference, see <https://o2r.info/reproducible-agile/> for more information. Please [retweet](#) and spread the word! [Your questions & feedback](#) are most welcome.

Here is Daniel's attempt at a **non-specialist summary**:

More and more research use data and algorithms to answer a question. That makes it harder for researchers to understand a scientific publication, because you need more than just the text to understand what is really going on. You need the software and the data to be able to tell if everything is done correctly, and to be able to re-use new and exciting methods. We took a look at the existing guides for such research and created our own criteria for research in sciences using environmental observations and maps. We used the criteria to test how reproducible a set of papers from the AGILE conference actually are. The conference is quite established and the papers are of high quality because they were all suggested for the “best paper” awards at the conference.

The results are quite bad! We could not re-create any of the analyses. Then we asked the authors of the papers we evaluated if they had considered that someone else might want to re-do their work. While they all think the idea is great, many said they do not have the time for it.

The only way for researchers to have the time and resources to work in a way that is transparent to others and reusable openly is either to convince them of the importance or to force them. We came up with a list of suggestions to publishers and scientific conference organisers to create enough reasons for researchers to publish science in a re-creatable way.

AGILE 2018 pre-conference workshop report

21 Jun 2018 | By Daniel Nüst

Last week o2r team member Daniel co-organised a workshop at the 21st AGILE International Conference on Geographic Information Science in Lund, Sweden. The workshop went very well and Daniel together with his colleagues was able to spread the word about reproducible research and Open Science. They are pretty sure they convinced some new scientists to reconsider their habits!

Daniel wrote a short report about the workshop: <https://o2r.info/reproducible-agile/2018/#workshop-report>

We are ready for the 2nd workshop on [#reproducibility](#) [#reproducibleresearch](#) at AGILE conference in Lund [#agileconf2018](#)
<pic.twitter.com/bbBok1SnRm>

— Daniel Nüst (@nordholmen) 12. Juni 2018

The workshop series will probably be continued at the next AGILE conference in Limassol, Cyprus. For o2r participating in such a workshop is a great way to stay in touch with users of reproducibility tools and practices, and to give back to the communities not only with technology but with education.

Report from EGU 2018

18 Apr 2018 | By Daniel Nüst

Last week EGU General Assembly (GA) 2018 took place in Vienna, Austria, and it was packed with interesting sessions and inspiring presentations. The o2r team humbly tried to contribute to a massive conference: 15075 participants from 106 countries gave 17323 presentations in 666 sessions (it's been reported the programme committee briefly discussed adding a session...), taught 68 short courses, and collaborated in 294 side events. Let's go through the events with o2r participation in chronological order.



Image courtesy of EGU website.

On *Monday*, Daniel joined the first ever **EarthArXiv townhall meeting**. He was happy to share his stickers with a small but engaged crowd and experienced an open-minded discussion about the young community-led EarthArXiv (already over 300 pre- and postprints after a little over 6 months), preprints, postprints, and the bigger picture of Open Access in the context of the two large conferences in the geosciences, EGU GA and the AGU Fall Meeting. AGU's cooperation with ESSOAr on abstracts and poster publications was presented at the meeting by Brooks Hanson. The event went really well and it was fun to meet fellow Open Science enthusiasts Friedrich Hawemann and David Fernandez-Blanco (the mastermind behind the gif-loaden entertaining tweets by @EarthArXiv).

Friedrich Hawemann making the case for preprints at the @EarthArXiv Townhall at #EGU18 #preprint #postprint #openaccess pic.twitter.com/TgfoPSVksD

— Daniel Nüst (@nordholmen) 9. April 2018

On *Tuesday* the evening events continued with the always enjoyable **OSGeo townhall meeting**. Its theme was "Open Science demystified" and organiser Peter Löwe nicely connected the spirit and goals of an Open Source organisation with Open Science. As usual, it did not take long until newcomers could be helped with concrete advice on software, development, and transformation to FOSS for organisations by the attending mix of FOSS users, developers, contributors, and old-timers.

It's that time of year again: @OSGeo townhall meeting at #EGU18 @EuroGeosciences@drpeterloewe continues his tremendous outreach activity (4th time convening?) and connects #OpenScience with #OpenSource - OSGeo is not limited to the latter! pic.twitter.com/UuFe0fAdxZ

— Daniel Nüst (@nordholmen) 10. April 2018

On *Wednesday* Daniel had to shift his attention to the early morning. In the PICO session **E4.4, "R and the benefit of low-cost solutions - democratic participation to face challenges in Earth science"**, he demonstrated the usefulness of **rocker/geospatial** for science with a number of showcases in a PICO presentation slot packed with exciting projects and software presentation.

- Abstract: <https://meetingorganizer.copernicus.org/EGU2018/EGU2018-8500-1.pdf>
- Slides: <https://doi.org/10.5281/zenodo.1217911>
- Thanks: [showcase authors](#)

PICO uploaded! Don't get to bed too late today or you'll miss "rocker/geospatial: a flexible runtime environment for geoscientific data analysis" at #EGU18 - PICO spot 4 tomorrow at 08:30 hrs. I showcase the community work headed by @cboettig & @eddelbuettel for R in @Docker pic.twitter.com/F8KK453fFx

— Daniel Nüst (@nordholmen) 10. April 2018

In the same session, [Edzer](#) presented “R vector and raster data cubes for openEO”, his latest work to continue the evolution for spatial data handling in R and connecting it to [openEO](#). Both o2r team members could welcome many interested scientists at their PICO screens and had to stay until the very end answering questions and discussing the depths of the respective implementations.

- Abstract: <https://meetingorganizer.copernicus.org/EGU2018/EGU2018-8198.pdf>
- [stars](#) R package: <https://r-spatial.github.io/stars/>

Steadily growing crowd gathering at [#EGU18](#) session on [#R](#) and the benefit if low-cost solutions - democratic participation to face challenges in Earth sciences [#rstats](#) [#PICO](#) [@EuroGeosciences](#) [#rspatial](#) pic.twitter.com/MdgpJLnu4y

— Daniel Nüst (@nordholmen) 11. April 2018

On *Thursday afternoon* Daniel was joined by [Markus](#) and good friends of o2r from New York, [Vicky](#) and [Remi](#) from [NYU Center for Data Science](#) and [ReproZip](#), to [continue the collaboration](#) on teaching tools for Open Science and Reproducible Research at EGU. They welcomed a large audience (70+ people) to the [short course “Writing reproducible geoscience papers using R Markdown, Docker, and GitLab”](#). The course was hands-on, so participants worked with their own laptops. Hopefully most of them arrived safely home with a working research environment for [R](#) and [git](#). The course’s contents and speed are adjusted to accommodate the diverse previous knowledge from a multidisciplinary conference such as EGU. Inspired by the great [Carpentry courses](#), but considerably shorter and more dense, all conveners/teachers were active at the same time to lead through the instructions and help fixing the many little issues during software installations. We tried hard to leave no one hanging behind and albeit being confronted with an estimated number of 6 operating systems the [RStudio](#)-based instructions stood their ground excellently and we are glad to have received numerous positive feedbacks from participants.

Almost 70 people at our [#egu18repro](#) [#egu18](#) session!! So many folks eager to learn about best practices for reproducible research in geoscience! <https://t.co/XSxc0s53uk>

— Vicky Steeves (joinmastodon.org) (@VickySteeves) 12. April 2018

The *course material* is available openly online at <https://vickysteeves.gitlab.io/repro-papers/> and if you could not be there, be sure to try and check out the Twitter hashtag [#egu18repro](#) for some impressions. The course is roughly split in two sessions à 90 minutes:

- Introduction to Open Science, using git and GitLab
- R Markdown for reproducible papers and rendering of R Markdown manuscripts with GitLab CI

We sincerely thank the attendees for the useful questions and the positive atmosphere at the course! People were helping each other and showed patience when little breaks had to be taken to solve individual issues. We welcome comments, ideas and suggestions in the [GitLab repository of the course](#). We hope it’s not the last time we can use the material ourselves but also invite everybody to use it. It contains numerous links to more detailed courses and we thank the R and Open Science communities for the breadth of existing tutorials and the inspiration they provide.

Our short course on writing reproducible geoscience papers is DONE! [#egu18repro](#) [#EGU18](#) pic.twitter.com/V3vwojpk11

— Vicky Steeves (joinmastodon.org) (@VickySteeves) 12. April 2018

The *evening* belonged to yet another townhall meeting: “[Research Software Engineers in the Geosciences](#)”. Daniel initiated this meeting to bring together researchers developing software, or software developers doing research, to get to know the existing national chapters and initiatives as well as each other. A diverse group came together from different countries and [scientific divisions](#) to share their experiences and to discuss how to improve the situation for RSEs in the geosciences (see [the RSE’s objectives](#)). A more detailed report from the townhall will follow in due course on [the RSE Blog](#), until then see [Daniel’s full thread on Twitter](#).

First research software engineers meeting at the EGU just started. Great to see people engaging with the role behind a crucial part of science. [#rse](#) [#RSEng](#) [@SoftwareSaved](#) [@RSE_de](#) [@nordic_rse](#) [@nl_rse](#) [#EGU18](#) [@EGU_ESSI](#) pic.twitter.com/Y0IDsAtaae

— Daniel Nüst (@nordholmen) 12. April 2018

On *Friday* it was time for PICOs and posters. Daniel and Markus presented “[Open Environmental Data Analysis](#)” and “[Reproducible research bindings](#)” respectively in the session “**Open Data, Reproducible Research, and Open Science**”. Again we enjoyed fruitful discussions and missed out on the interesting other presentations as we were fortunate enough to be visited by interested people throughout the whole viewing time.

- Daniel's slides:

DOI [10.5281/zenodo.1217912](https://doi.org/10.5281/zenodo.1217912)

Last day at #EGU18 and what a great week it has been so far @EGU_ESSI @EuroGeosciences. Now at PICO spot 1 in session #OpenData #OpenScience and #ReproducibleResearch incl. presentation by @thomas_barto and me on open env. analysis w/ @openSenseMap & @SenseBox_De pic.twitter.com/tRglBkhiMv

— Daniel Nüst (@nordholmen) 13. April 2018

@MarkusKonkol presents results from @o2r_project : reproducible research bindings, increasing research transparency and understanding. #OpenScience #reproducibleResearch #EGU18 #EGU18ESSI pic.twitter.com/UY7PqU4sA2

— Daniel Nüst (@nordholmen) 13. April 2018

Later that morning Edzer presented a poster on “openEO: an open API for cloud-based big Earth Observation processing platforms” ([abstract](#)) in the session “Data cubes of Big Earth Data - a new paradigm for accessing and processing Earth Science Data”.

Drawing a big crowd at @EGU_ESSI poster session: @edzerpebesma presenting @open_EO project, an open API for Earth observation data and processing. #EGU18 #DataCubes #opensource pic.twitter.com/1rDoexZgPJ

— Daniel Nüst (@nordholmen) 13. April 2018

We are happy to thank the great people from Copernicus, the organisers of the conference and a great supporter of Open Science as well as a [partner of o2r](#), who we got to meet and catch up with. The conference has been great and here we only scratch the surface of fun, entertaining and educational experiences where o2r team members presented or convened and lack the many times we met [new people and communities](#), colleagues, and friends to talk science.

Thanks for reading!

Digitisation of Science @ WWU

27 Feb 2018 | By Daniel Nüst

o2r Team member Daniel was invited by the university's press office to participate in a series of interviews and articles on digitisation or "digitalisation" at the WWU Münster:

The video is now available online in German (embedded below) and with English subtitles. You can also watch it on Facebook or in the WWU video portal.

Daniel wrote a brief summary for our blog and shares his experience:

Interview summary

First we talked about how digitisation is a familiar topic for computer scientists professionally (digital data, algorithms), but also something we encounter as citizens. Next I explained the importance of reproducibility in science and when asked if that was not the case in the past, I outlined the new challenges of a completely digital research workflow. I summarise the idea of the o2r project and use the term "*digital laboratory*" as a metaphor for our Executable Research Compendium, which collects all research artefacts and opens them up in a transparent way and allows collaboration. We then briefly touch on the fact that the concepts and ideas are relevant for all sciences, but how o2r (and geoinformatics) focuses on geoscience applications. Looking ahead I mention our plans to bring our idea of a reproducible article into the publishing workflow of regular scientists. Is digitisation a blessing or a curse? It's both, because it creates new challenges and exciting applications in geoinformatics, but the amount of information in large datasets is not always clear and requires critical dealing.



What was it like?

Shooting the interview was fun and a great experience. Having two cameras pointed at you and 4 people standing critically observing in the background could have been intimidating, but it wasn't - big thanks to the team!

The film shooting took only about 40 minutes (some technical preparations, two full takes, a little break and small talk) and was prepared with a one-hour conversation some weeks ahead. I wrote down answers to some questions I expected - but the actual questions were not shared, for the better I think because the spontaneity makes it a conversation and less of a lecture. The video published online is from the second, shorter take. I wish they would have used the first one, because it was longer (around 10 minutes, so double the target time) and I could make more good points and feel like I got the message across better. Researchers can go on for hours about topics they care about, and I hope my enthusiasm about Open Science and Reproducible Research does come across. Though I am partly unhappy with the content, I hope the brevity spikes interest by fellow researchers and students at University of Münster. Next time you feel like cutting down your paper from 6000 to 5000 words is hard, try bringing it down to 2 minutes of talking to a non-expert :-). A worthwhile exercise by the way.

Although in this case, a non-expert could very well be an experienced scientist! The lack of a common terminology for reproducible/replicable etc. became very apparent during the preparations. For the next video I'll make every spectator read "[Terminologies for Reproducible Research](#)" first ...

"[Digitalisierung](#)" is a very hot topic in Germany, both in [politics](#) and [economy](#). It regularly makes it into national news and more and more also into small talk. There is so much connected with digitisation I did not touch on, like artificial intelligence (AI). How can we ensure research transparency and reproducibility when we don't even know how something works? *The one thing I regret* not saying in the interview is the fact that having studied computer science, I rarely grasp the difficulties non-programmers must have with digitisation. While I do sometimes have to "explain computers" to friends and family, I don't do it often enough, and must show more patience when I get the chance. AI, web services, cloud computing - it is complex stuff! Let's help non-techies close to us more in understanding them (reading recommendation: "[Tubes: A Journey to the Center of the Internet](#)" by Andrew Blum) !

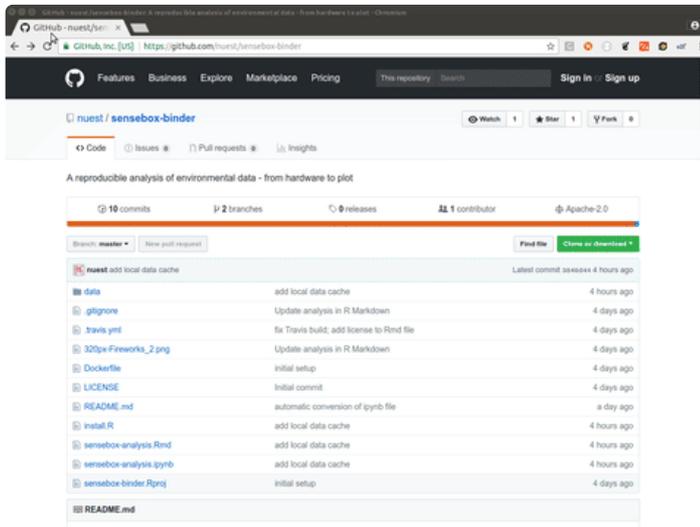
Formulating my view on digitisation in research and its impact on research reproducibility in "plain language" was a worthwhile challenge and I can only recommend every researcher to participate in public relations workshops et cetera to try it out. You got to take every chance you can get to reach out to non-scientists (research resolution: [Make sure all papers have non-specialist summaries](#)). I applaud all bloggers and podcasters out there who do!

Open environmental data analysis

12 Jan 2018 | By Daniel Nüst

This article is cross-posted in German on the [senseBox blog](#).

It's the time of the year to make resolutions and to see beyond one's own nose. For o2r team member Daniel, this meant to explore what he could do with his brand new [senseBox:home](#) and the awesome [BinderHub](#) instead of putting it on the back burner.



Building on a deep stack of Open Hardware, Free and Open Source Software, and Open Data, he created a fully open analysis of particulate measurements at New Year's Eve in Münster. With just a few clicks you can open the exact computational environment which he utilized to retrieve historic sensor data from the openSenseMap API, and to analyse and visualise it with R. And all that without installing any software to your computer, all you need is a web browser.

The following screenshots show the RStudio and Jupyter Notebook renderings of the workflow.

0.0.0.8888/proxy/51735/

File Edit Code View Plots Session Build Debug Profile Tools

sensebox-analysis.Rmd

```

28
29 ## Analysis
30
31 In the remainder of this file, code "chunks" and text are
[interspersed](https://en.wikipedia.org/wiki/Literate_programming)
understandable workflow.
32
33 The analysis of takes a look at fine particulate matter measured
34
35 Note: The data is included in the archive as a backup in [JSON]
format.
36 This document by default can only be compiled as long as the openSe
37 To use local backup data, set the variable 'online' to 'FALSE' in t
38
39 Load required libraries
40
41 ## (r packages, warning=FALSE, message=FALSE)
42 library("opensemapr")
43 library("dplyr")
44 library("lubridate")
45 library("units")
46 library("sf")
47 library("leaflet")
48 library("readr")
49 library("jsonlite")
50 library("here")
51 ...
52
53 <span style="color: grey;">[output hidden]</span>
54
55 Load data on senseBoxes
56
57 ## Load required libraries
58
59

```

Console Terminal R Markdown

R version 3.4.2 (2017-09-28) -- "Short Summer"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY. .
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

0.0.0.8888/proxy/51735/view=markdown

sensebox-analysis.html

```

## 59c3b7bcd67eb5001137e38:479
## 5a0507159fd3c200118eaf7:332
## 5a2fe63775a96c000fea9148:407
## 5a40e41fd9a664000f4b455c:479
##

```

We can now plot 2512 measurements.

```

plot(value-createdAt, ms_data,
type = "p", pch = "*", cex = 2, # new year's style
col = factor(ms_data$sensorId),
xlab = NA,
ylab = unique(ms_data$unit),
main = "Particulates measurements (PM2.5) on New Year 2017/2018",
sub = paste(nrow(ms_boxes), "stations in Münster, Germany\n",
"Data by openSenseMap.org licensed under",
"Public Domain Dedication and License 1.0"))

```

Particulates measurements (PM2.5) on New Year 2017/2018

µg/m³

20:00 22:00 00:00 02:00 04:00

5 stations in Münster, Germany
Data by openSenseMap.org licensed under Public Domain Dedication and License 1.0

You can see, it was a very "particular" celebration.
Who is the record holder?

```
top_three <- ms_data %>
```

Modified

- Jan 4, 2018, 3:49 PM
- Jan 7, 2018, 1:51 PM
- Jan 4, 2018, 12:35 PM
- Jan 4, 2018, 5:52 PM
- Jan 4, 2018, 11:34 AM
- Jan 8, 2018, 9:27 AM
- Jan 4, 2018, 11:06 AM
- Jan 8, 2018, 2:33 PM
- Jan 8, 2018, 3:45 PM
- Jan 8, 2018, 3:45 PM
- Jan 8, 2018, 1:39 PM
- Jan 4, 2018, 3:48 PM

0.0.0.8888/notebooks/sensebox-analysis.ipynb

Jupyter sensebox-analysis Last Checkpoint: Yesterday at 12:57 PM (unsaved changes)

File Edit View Insert Cell Kernel Help

```

knitr::kable(data.frame(nrow(all_boxes), nrow(pm25_boxes)),
col.names = c(
"# senseBoxes",
paste("# senseBoxes with PM2.5 measurements around", format(analysis_date, "%Y-%m-%d %T %Z"))))

```

```

| # senseBoxes | # senseBoxes with PM2.5 measurements around 2018-01-01 00:00:00 UTC |
|-----|-----|
| 1104 | 35 |

```

Exploring openSenseMap

The openSenseMap currently provides access to `r nrow(all_boxes)` senseBoxes of which `r nrow(pm25_boxes)` provide measurements of **PM2.5** around `r format(analysis_date, "%Y-%m-%d %T %Z")`.

The following map shows the PM2.5 sensor locations.

In [4]: `plot(pm25_boxes)`

54°N

52°N

50°N

○ outdoor

And of course he worried about reproducibility and put in several layers of backup! Learn all about it at the GitHub repository:

<https://github.com/nuest/sensebox-binder/>

Or get a peak at the output of the analysis here:<https://nuest.github.io/sensebox-binder/sensebox-analysis.html>

And we were not the only ones taking a look at particulate matter in Germany using R. [Johannes Friedrich](#), researcher at [University of Bayreuth](#), used his R package [senseBox](#) to download and plot data of over 400 senseBoxes. See [his blog](#) for his findings.

Events in 2018: Call for participation

05 Jan 2018 | By Daniel Nüst

As everyone is slowly coming back to work, the o2r team wishes *Happy New Year*. What better way to start the year with planning some fun trips? Here are our recommendations for upcoming events:

- EGU sessions on Reproducible Research, R, and FOSS
- EGU short course “Writing reproducible geoscience papers”
- AGILE pre-conference workshop “Reproducible Research Publications”

Please share this information with potentially interested parties (and retweet). Thanks!

Update! Added two more sessions and the OSGeo Townhall.



Image courtesy of EGU website.

Open Science, R, and FOSS sessions at EGU General Assembly 2018

The European Geophysical Union's General Assembly (EGU GA) takes place once more in April in Vienna #EGU18. The deadline for abstracts is 10 Jan 2018, 13:00 CET, so don't delay, **submit your abstract today** to one of the following sessions:

- Open Data, Reproducible Research, and Open Science (ESSI3.5)
- R's deliberate role in Earth sciences (PICO Session)
(IE4.4/GM2.8/AS5.8/BG1.17/CL5.28/GD10.10/GMPV10.5/HS3.5/SSS13.77/TS11.12)
- Free and Open Source Software (FOSS) for Geoinformatics and Geosciences (PICO Session, ESSI3.1)

Other sessions without o2r team members convening, but looking very interesting are

- Leveraging data-driven workflows to accelerate Earth Science research (ESSI3.3)
- Data science, Analytics and Visualization: The challenges and opportunities for Earth and Space Science (ESSI4.3)
- Future of (hydrological) publishing (PICO session) (HS1.16)
- Web-based Exchange and Processing of Environmental Data (ESSI2.6)
- Emerging Computational Technology (PICO session, IE4.2/AS5.5/BG1.32/CL5.17)
- Virtual Research Environments: creating online collaborative environments to support research in the Earth Sciences and beyond (co-organised with American Geophysical Union, ESSI2.4)

After our previous participations we look forward to yet another event with interesting presentations and great conversations. If you're going, too, make sure to join the **OSGeo Townhall: Open Science demystified** (TM8, on Tuesday) and the townhall meetings **“Research Software Engineers in the Geosciences”** (TM13, room L8 on Thursday) and **“EarthArXiv - a preprint server for the Earth Sciences”** (TM4, room L2 on Monday).

See you at EGU!

Short course on reproducible papers at EGU General Assembly 2018

After organising a [workshop on reproducible computational research in the publication cycle](#) last year, o2r is teaming up again with [ReproZip](#) to help geoscientists tackling the challenges of reproducible papers. This year we organise the short course [SC1.13 - Writing reproducible geoscience papers using R Markdown, Docker, and GitLab](#). We plan to go guide participants through the steps of writing an Open Science publication and managing its rendering, publication, and archival by using free and open online platforms.

Let us know you're interested to join with only two clicks: <https://doodle.com/poll/ngn9fqvhfkp3hau>

Other short courses without o2r participation, but looking promising (they both use **R!**), are

- [SC1.34 - Improving statistical evaluations in the geosciences](#)
- [SC1.17 - Using R for natural hazard risk modelling, with applications to wildfire risk forecasting](#)

Reproducible Research Publications at AGILE 2018

We are happy to announce another [continuation](#), a pre-conference workshop at the [21st AGILE International Conference on Geographic Information Science](#) in Lund, Sweden: **“Reproducible Research Publications”**

The half day workshop attempts to provide a hands-on introduction to reproducible research by reproducing a provided real-world publication. Together with the instructors they create a reproducible document from text, code, and data of a scholarly publication and publish it in a data repository.

The workshop is accepted and will be announced [on the conference website](#) soon. Please also check the [workshop website](#) for detailed information on registration and scheduling.



Image courtesy of AGILE website.

Submit your registration *both* at the [conference website](#) (will open soon!) and the workshop repository (see [instructions](#)): <https://github.com/o2r-project/reproducible-agile/issues/new>

The workshop is co-organized by o2r team members and Frank Osterman (ITC, Enschede), Barbara Hofer (Z_GIS), Carlos Granell (Jaume I), Valentina Cerutti (ITC), and Rusne Sileryte (OTB, TU Delft). *We look forward to your registration!*

Reference Implementation - Try it out!

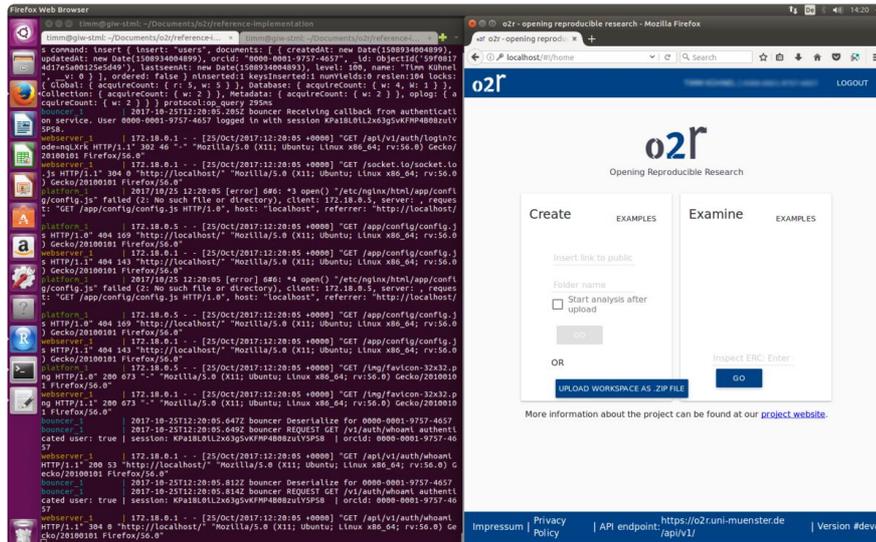
31 Oct 2017 | By Daniel Nüst

Post updated on March 15 2018 to reflect simplified run commands.

Our project is going into its final phase. We are working on integrating our latest experiences and discussions into the ERC specification and constantly add new features to the implementation of the reproducibility service.

We also try to keep our demo server up to date. *But what good is a reproducibility platform, when you can only try it online?*

Inspired by the just passed Open Access Week (#oaweek), we've started a new repository [reference-implementation](#) to expose our developments, which have been open source from the start, to the interested public.

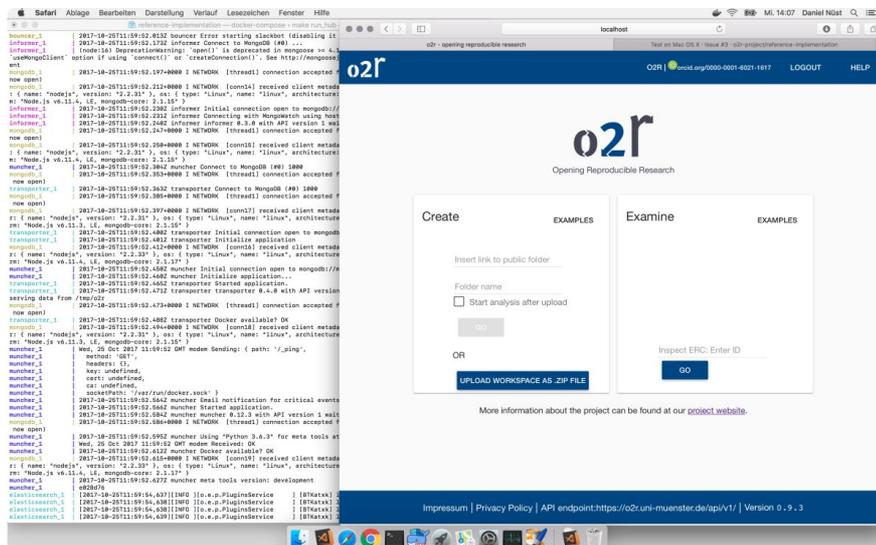


Screenshot: o2r reference implementation on Ubuntu.

It comprises documentation for run o2r software on a completely new machine:

- Run o2r locally with pre-build Docker images (the regular approach, let's you easily update to later versions)
- Download all source code, build Docker images, and then run o2r locally (the investigative approach)
- Upload a demo workspace or ERC

The only efforts besides a few commands on your computer is registering a client application with ORCID to be able to log in, because there is no other way to authenticate within the o2r platform and microservices. You may also get an access token from Zenodo to "ship" your completed ERC. Eventually this repository will be the basis for a citable package of our software.

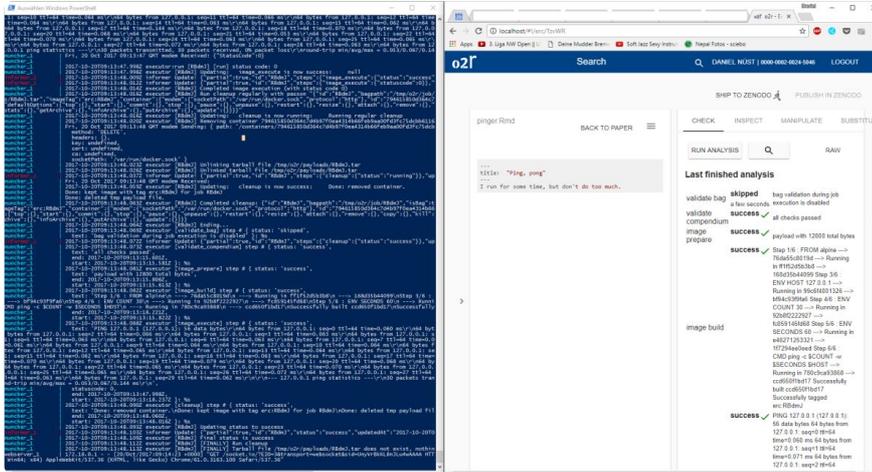


Screenshot: o2r reference implementation on OS X.

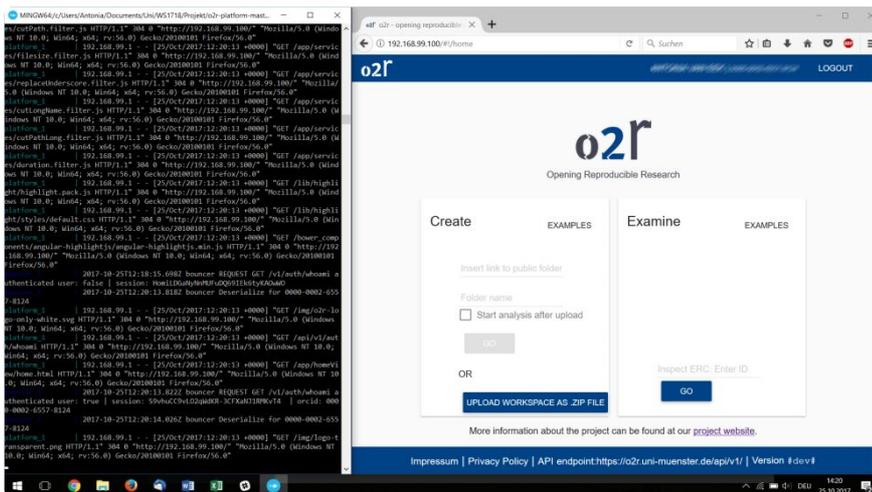
We look forward to your feedback!

tl;dr

1. Install **Docker** and **docker-compose**
2. Download the o2r reference implementation repository and run it with
 - o `git clone https://github.com/o2r-project/reference-implementation`
 - o `docker-compose up`



Screenshot: o2r reference implementation on *Windows 10*.



Screenshot: o2r reference implementation on *Windows 10* (Docker Toolbox), contributed by Antonia - Thanks!

Reproducible Research Badges

12 Sep 2017 | By Lukas Lohoff, Daniel Nüst

This blog post presents work based on the study project *Badges for computational geoscience containers at ifgi*. We thank the *project team* for their valuable contributions!

This blog post was extended and presented and published as a peer-reviewed short paper at the [AGILE Conference 2019](#).

Find the article [here on EarthArXiv](#) and the presentation [here on OSF](#). The citation is

Nüst, Daniel, Lukas Lohoff, Lasse Einfeldt, Nimrod Gavish, Marlena Götza, Shahzeib T. Jaswal, Salman Khalid, et al. 2019. "Guerrilla Badges for Reproducible Geospatial Data Science (AGILE 2019 Short Paper)." EarthArXiv. June 19. doi:10.31223/osf.io/xtsqh.

Introduction

Today badges are widely used in open source software repositories. They have a high recognition value and consequently provide an easy and efficient way to convey up-to-date metadata. Version numbers, download counts, test coverage or container image size are just a few examples. The website [Shields.io](#) provides many types of such badges. It also has an API to generate custom ones.

Now imagine similar badges, i.e. succinct, up-to-date information, not for software projects but for modern research publications. It answers questions such as:

- When was a research paper published?
- Is the paper openly accessible?
- Was the paper published in a peer reviewed journal?
- What is the research's area of interest?
- Are the results reproducible?

These questions cover basic information for publications (date, open access, peer review) but also advanced concepts: the *research location* describes the location a study is focusing on. A publication with *reproducible results* contains a computation or analysis and the means to rerun it - ideally getting the same results again.

We developed a back-end service providing badges for reproducible research papers.

Overview of badges for research

We are however not the first nor the only ones to do this. [ScienceOpen](#) is a search engine for scientific publications. It has badges for open access publications, content type, views, comments and the [Altmetric](#) score as displayed in Figure 1.



Figure 1: ScienceOpen badges in a search result listing.

These are helpful when using the ScienceOpen website, but they are not available for other websites. Additional issues are the inconsistent style and missing information relevant for reproducible geosciences, e.g. reproducibility status or the research location.

Badges are also used directly on publications, without the search portal "middleman". The published document, poster or presentation contains a badge along with the information needed to access the data or code. The [Center for Open Science designed badges](#) for acknowledging open practices in scientific articles accompanied by guidelines for [incorporating them into journals' peer review workflows](#) and [adding them to published documents](#), including large colored and small black-and-white variants. The badges are for *Open Data*, *Open Materials*, and *Preregistration* of studies (see Figure 2) and are adopted by over a

dozen of journals to date (cf. [Adoptions and Endorsements](#)).



Figure 2: COS badges.

University of Washington's [eScience Institute](#) created a peer-review process for open data and open materials badges <https://github.com/uwescience-open-badges/about> based on the COS badges. The service is meant for faculty members and students at the University of Washington, but external researchers can also apply. The initiative also has a list of relevant [publications on the topic](#).

A study by Kidwell et al. [1] demonstrates a positive effect by the introduction of open data badges in the journal *Psychological Science*: After the journal started awarding badges for open data, more articles stating open data availability actually published data (cf. [2]). They see badges as a simple yet effective way to promote data publishing. The argument is very well summarized in the tweet below:

Simple rewards are sufficient to see the change we want to occur #SSP2017 pic.twitter.com/P1H4hpQeqN

— David Mellor (@EvoMellor) 1. Juni 2017

Peng [3, 4] reports on the efforts the journal *Biostatistics* is taking to promote reproducible research, including a set of “kite marks”, which can easily be seen as minimalistic yet effective badges. **D** and **C** if data respectively code is provided, and **R** if results were successfully reproduced during the review process (implying D and C). Figure 3 shows the usage of **R** on an article's title page (cf. [5]).



Figure 3: *Biostatistics* kite mark **R** rendering in the PDF version of the paper.

The Association for Computing Machinery (ACM) provides a common terminology and standards for artifact review processes for its conferences and journals, see their policies website section on [Artifact Review Badging](#). They have a system of three badges with several levels accompanied by specific criteria. They can be independently awarded:

- *Artifacts Evaluated* means artifacts were made available to reviewers and awarded the level *Functional* or *Reusable*
- *Artifacts Available* means a deposition in a repository ensures permanent and open availability (no evaluation)
- *Results Validated* means a third party successfully obtained the same results as the author at the levels *Results Replicated* (using, in part, artifacts provided by the author) or *Results Reproduced* (without author-supplied artifacts)

Figure 4 shows a rendering of the ACM badges.



Figure 4: ACM badges, from left to right: Artifacts Evaluated – Functional, Artifacts Evaluated – Reusable, Artifacts Available, Results Replicated, and Results Reproduced. (Copyright © 2017, ACM, Inc)

Although these examples are limited to a specific journal, publisher, or institution, they show the potential of badges. They also show the diversity, limitations, and challenges in describing and awarding these badges.

For this reason, our goal is to explore sophisticated and novel badge types (concerning an article's reproducibility, research location, etc.) and to find out how to provide them independently from a specific journal, conference, or website.

An independent API for research badges

Advanced badges to answer the above questions are useful for literature research, because they open new ways of exploring research by allowing to quickly judge the relevance of a publication, and they can motivate efforts towards openness and reproducibility. Three questions remain: How can the required data for the badges be found, ideally automatically? How can the information be communicated? How can it be integrated across independent, even competitive, websites?

Some questions on the data, such as the publication date, the peer review status and the open access status can already be answered by online research library APIs, for example those provided by [Crossref](#) or [DOAJ](#). The [o2r API](#) can answer the remaining questions about reproducibility and location: Knowing if a publication is reproducible is a core part of the o2r project. Furthermore, the location on which a research paper focuses can be extracted from spatial files published with an Executable Research Compendium [6]. The metadata extraction tool [o2r-meta](#) provides the latter feature, while the [ERC specification](#) and [o2r-muncher](#) micro service enable the former.

How can we integrate data from these different sources?

[o2r-badger](#) is a *Node.js* application based on the [Express](#) web application framework. It provides an API endpoint to serve badges for reproducible research integrating multiple online services into informative badges on scientific publications. Its [RESTful API](#) has routes for five different badge types:

- *executable*: Information about executability and reproducibility of a publication
- *licence*: licensing information
- *spatial*: a publication's area of interest
- *releasetime*: publication date
- *peerreview*: if and by which process the publication was peer reviewed

The API can be queried with URLs following the pattern `/api/1.0/badge/:type/:doi`. `:type` is one of the aforementioned types, and `:doi` is a publication's [Digital object identifier](#) (DOI).

The badger currently provides badges using two methods: internally created SVG-based badges, and redirects to [shields.io](#). The redirects construct a simple shields.io URL. The SVG-based badges are called *extended* badges and contain more detailed information: the extended *license* badge for example has three categories (*code*, *data* and *text*, see Figure 5), which are [aggregated](#) to single values (open, partially open, mostly open, closed) for the shields.io badge (see Figure 6).

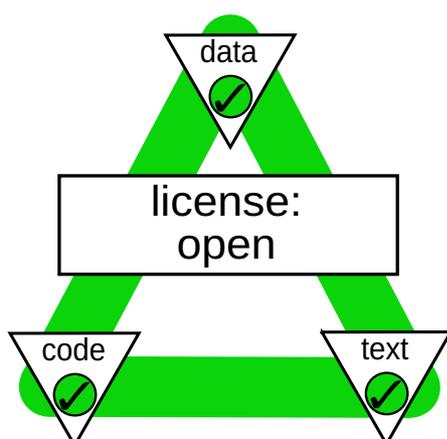


Figure 5: An extended licence badge reporting open data, text and code.

Extended badges are meant for websites or print publications of a single publication, e.g. an article's title page. They can be resized and alternatively provided pre-rendered as a PNG image. In contrast, the standard shields.io badges are smaller, text based badges. They still communicate the most important piece of information:



Figure 6: An shields.io based small badge, based on the URL <https://img.shields.io/badge/licence-open-44cc11.svg>.

They excel at applications where space is important, for example search engines listing many research articles. They are generated on the fly when a URL is requested (e.g. <https://img.shields.io/badge/licence-open-44cc11.svg>) which specifies the text (e.g. `licence` and `open`) and the color (`44cc11` is a [HTML color code](#) for green).

Let's look at another example of an *executable* badge and how it is created. The badge below is requested from the badger demo instance on the o2r server by providing the DOI of the publication for the `:doi` element in the above routes:

<https://o2r.uni-muenster.de/api/1.0/badge/executable/10.1126%2Fscience.1092666>

This URL requests a badge for the reproducibility status of the paper "Global Air Quality and Pollution" from *Science* magazine identified by the DOI [10.1126/science.1092666](https://doi.org/10.1126/science.1092666). When the request is sent, the following steps happen in o2r-badger:

1. The badger tries to find a reproducible research paper (called Executable Research Compendium [ERC](#)) via the o2r API. Internally this searches the database for ERC connected to the given DOI.
2. If it finds an ERC, it looks for a matching *job*, a report of a reproduction analysis.
3. Depending on the reproduction result (`success`, `running`, or `failure`) specified in the job, the badger generates a green, yellow or red badge. The badge also contains text indicating the reproducibility of the specified research publication.
4. The request is redirected to a [shields.io](#) URL link containing the color and textual information..

The returned image contains the requested information, which is in this case a successful reproduction:

URL: <https://img.shields.io/badge/executable-yes-44cc11.svg>

Badge:



If an extended badge is requested, the badger itself generates an SVG graphic instead.

Badges for reproducibility, peer review status and license are color coded to provide visual aids. They indicate for example (un)successful reproduction, a public peer review process, or different levels of open licenses. These badges get their information from their respective external sources: the information for peer review badges is requested from the external service *DOAJ*, a community-based website for open access publications. The *Crossref* API provides the dates for the relesetime badges. The spatial badge also uses the o2r services. The badger service converts the spatial information provided as coordinates into textual information, i.e. place names, using the [Geonames API](#).

Spread badges over the web

There is a great badge server, and databases providing manifold badge information, but how to get them displayed online? The sustainable way would be for research website operators to agree on a common badge system and design, and then incorporate these badges on their platforms. But we know it is unrealistic this ever happens. So instead of waiting, or instead of engaging in a lengthy discourse with all stakeholders, we decided to create a [Chrome extension](#) and augment common research websites. The *o2r-extender* automatically inserts badges into search results or publication pages using client-side browser scripting. It is [available in the Chrome Web Store](#) and ready to be tried out.

The extender currently supports the following research websites:

- Google Scholar <https://scholar.google.de/>
- DOAJ.org <https://doaj.org/>
- ScienceDirect.com <http://www.sciencedirect.com/>
- ScienceOpen.com <https://scienceopen.com/>
- PLOS.org <https://www.plos.org/>
- Microsoft Academic <https://academic.microsoft.com/>
- Mendeley <https://www.mendeley.com/>

For each article display on these websites, the extender requests a set of badges from the badger server. These are then inserted into the page's HTML code after rendering the regular website as shown exemplary in the screenshot in Figure 7.

Environmental quality and development: is there a Kuznets curve for air pollution emissions?

licence n/a executable running location Pará, Brazil release time 1994 peer review n/a

TM Selden, D Song - *Journal of Environmental Economics and ...*, 1994 - Elsevier
Abstract Several recent studies have identified inverted-U relationships between pollution and economic development. We investigate this question using a cross-national panel of data on emissions of four important air pollutants: suspended particulate matter, sulfur
Zitiert von: 2551 Ähnliche Artikel Alle 16 Versionen Web of Science: 688 Zitieren Speichern

Figure 7: Badges integrated into Google Scholar search results (partial screenshot).

When the badger does not find information for a certain DOI, it returns a grey “not available” - badge instead. This is shown in the screenshot above for the outermost license and peer review badges.

The extender consists of a content script, similar to [auserscript](#), adjusted to each target website. The content scripts insert badges at suitable positions in the view. A set of common functions defined in the Chrome extension for generating HTML, getting metadata based on DOIs, and inserting badges are used for the specific insertions. A good part of the extender code is used to extract the respective DOIs from the information included in the page, which is a lot trickier than interacting with an API. Take a look at the source code on [GitHub](#) for details.

But the extender is not limited to inserting static information. The results of searches can also be filtered based on badge value and selected badge types can be turned on or off directly from the website with controls inserted into the pages' navigation menus (see left hand side of Figure 8).

Badge Types

- Licence
- Executable
- Research location
- Release time
- Peer review

Badge Value Filter

Licence: partially open

Executable: yes

Research location:

Release time: newer than 2009

Peer review: blind

Air Conditioning Compressor Air Leak Detection by Image Processing Techniques for Industrial Applications
Pookongchai Kritsada, Nakornrat Prasit, Sookananta Bongkoj, Buasri Panhathai
MATEC Web of Conferences. 2015;26:03010 DOI 10.1051/mateconf/20152603010
Abstract | Full Text
licence partially open executable yes location Surat Thani, Thailand release time 2015 peer review yes

Goldenhar syndrome: a cause of secondary immunodeficiency?
De Golovine Serge, Wu Shuya, Hunter Jill V, Shearer William T
Allergy, Asthma & Clinical Immunology. 2012;8(1):10 DOI 10.1186/1710-1492-8-10
Abstract | Full Text
licence partially open executable yes location Texas, United States release time 2012 peer review blind

Analysis of Properties of Reflectance Reference Targets for Permanent Radiometric Test Sites of High Resolution Airborne Imaging Systems
Eero Ahokas, Juha Suomalainen, Jouni Peltoniemi, Teemu Hakala, Eija Honkavaara, Lauri Markelin
Remote Sensing. 2010;2(8):1892-1917 DOI 10.3390/rs2081892
Abstract | Full Text
licence partially open executable yes location Gulf Of Bothnia release time 2010 peer review blind

Assessment of satellite and model derived long term solar radiation for spatial crop models: A case study using DSSAT in Andhra Pradesh
Anima Biswal, M. V. R. Sessa Sai, S. V. C. Kameswar Rao
Computational Ecology and Software. 2014;4(3):205-214
Abstract | Full Text

Figure 8: Filtering search results on DOAJ. Results not matching the filter or articles where the DOI could not be detected are greyed out.

The extender is easily configurable: it can be enabled and disabled with a click on the icon in the browser toolbar. You can select the badge types to be displayed in the extension settings. Additionally it contains links to local info pages (“Help” and “About”, see Figure 9).

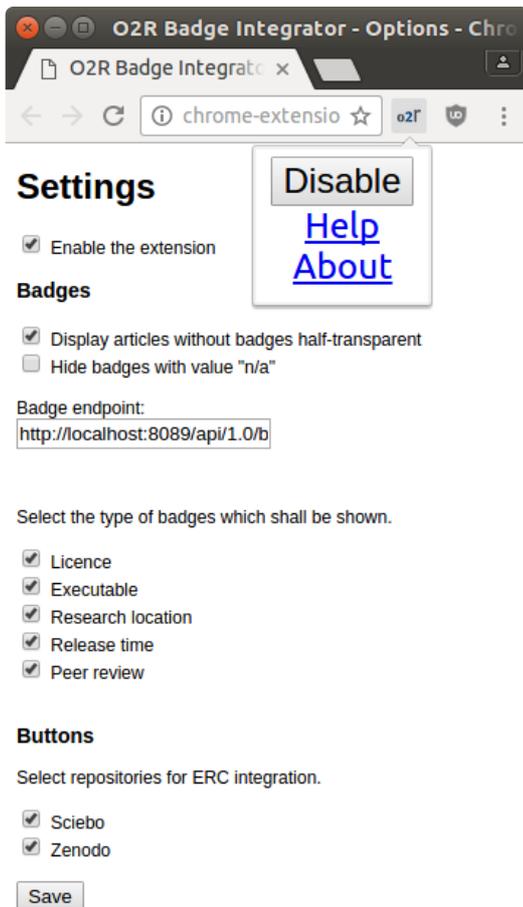


Figure 9: extender configuration.

Outlook: Action integrations

The *extender* also has a feature unrelated to badges. In the context of open science and reproducible research, the reproducibility service connects to other services in a larger context as described in the [o2r architecture](#) (see section Business context).

Two core connections are loading research workspaces from cloud storage and connecting to suitable data repositories for actual storage of ERCs. To facilitate these for users, the extender can also augment the user interfaces of the non-commercial cloud storage service [Sciebo](#) and the scientific data repository [Zenodo](#) with reproducibility service functionality.

When using *Sciebo*, a button is added to a file's or directory's context menu. It allows direct interaction with the o2r platform to upload a new reproducible research paper (ERC) from the current file or directory as shown in Figure 10.



Figure 10: Sciebo upload integration.

When you are viewing an *Executable Research Compendium* on [Zenodo](#), a small badge links directly to the corresponding inspection view in the o2r platform (see Figure 11):

February 28, 2017

ERC Software Open Access

Metatainer Test ERC

Lukas Lohoff

Test Executable Research Compendium (ERC). Used to develop an Zenodo loader for the O2R Project (<http://o2r.info>)**Publication date:**

February 28, 2017

DOI:DOI [10.5072/zenodo.69114](https://doi.org/10.5072/zenodo.69114)**ERC:**ERC [inspect](#)**License (for files):**[Creative Commons Attribution 4.0](#)

Preview

metatainer.zip

bag-info.txt	67 Bytes
bagit.txt	55 Bytes
data	
o Dockerfile	71 Bytes
o bagtainer.yml	62 Bytes
o document.Rmd	4.3 kB
o document.html	805.4 kB
o document.tex	5.3 kB
manifest-md5.txt	260 Bytes

Figure 11: Link to inspection view and tag "ERC" on Zenodo.

Share**Cite as**Lukas Lohoff. (2017, February 28). Metatainer Test ERC. <http://doi.org/10.5072/zenodo.69114>

Start typing a citation style...

Discussion

The study project [Badges for computational geoscience containers](#) initially implemented eight microservices responsible for six different badge types, badge scaling and testing. A microservice architecture using Docker containers was not chosen because of the need for immense scaling capabilities, but for another reason: developing independent microservices makes work organization much easier. This is especially true for a study project where students prefer different programming languages and have different skill sets.

However, for o2r the microservices were integrated into a single microservice for easier maintainability. This required refactoring, rewriting and bug fixing. Now, when a badge is requested, a [promise chain](#) is executed (see [source code example](#)). The chain reuses functions across all badges where possible, which were refactored from the study project code into small chunks to avoid [callback hell](#).

A critical feature of extender is the detection of the DOI from the website's markup. For some websites, such as [DOAJ.org](#) or [ScienceOpen.com](#), this is not hard because they provide the DOI directly for each entry. When the DOI is not directly provided, the extender tries to retrieve the DOI from a request to [CrossRef.org](#) using the paper title (see [source code for the DOI detection](#)). This is not always successful or may find incorrect results.

The Chrome extension supports nine different websites. If there are changes to one of these, the extender has to be updated as well. For example, [Sciebo](#) (based on [ownCloud](#)) recently changed their URLs to include a "fileid" parameter which resulted in an error when parsing the current folder path.

As discussed above, in an ideal world the Chrome extension would not be necessary. While there are a few tricky parts with a workaround like this, it nevertheless allows o2r as a research project to easily demonstrate ideas and prototypes stretching beyond the project's own code to even third party websites. Moreover, the combination of extender client and badger service is suitable for embedding a common science badge across multiple online platforms. It demonstrates a technical solution how the scientific community can create and maintain a cross-publisher, cross-provider solution for research badges. What it clearly lacks is a well-designed and transparent workflow for awarding and scrutinizing badges.

Future Work

One of the biggest source of issues for *badger* currently is the dependence on external services such as *Crossref* and *DOAJ*. While this cannot be directly resolved, it can be mitigated by requesting multiple alternative back-end services, which can provide the same information (e.g. *DOAJ* for example also offers licence information at least for publications), or even by caching. Furthermore, the newness of the o2r platform itself is another issue: *licence*, *executable*, and *spatial* badges are dependent on an existing ERC, which must be linked via DOI to a publication. If a research paper has not been made available as an ERC then a users will get a lot of "n/a" badges.

The *extender* is only available for Google Chrome and Chromium. But since Firefox is switching to [WebExtensions](#) and moving away from their old "add-ons" completely with [Firefox 57](#), a port from a Chrome Extension to the *openWebExtensions* makes the

extender available for more users. The port should be possible with a few changes due to only minor differences between the two types of extensions.

Other ideas for further development and next steps include:

- Interactive badges can provide additional information when hovering over them or when the badges are clicked, most importantly why and by who the badge was assigned.
- Provide the information behind the badges via an API.
- Create a common design for extended badges.
- Conduct a user study on extended and basic badges within a discovery scenario.
- Evaluating usage of badges in print applications and for visually impaired people (cf. COS badges)

For more see the GitHub issues pages of [o2r-badger](#) and [o2r-extender](#). Any feedback and ideas are appreciated, either on the GitHub repositories or in [this discussion thread](#) in the Google Group *Scientists for Reproducible Research*. We thank the group members for pointing to some of the resources referenced in this post.

References

- [1] Kidwell, Mallory C., et al. 2016. Badges to Acknowledge Open Practices: A Simple, Low-Cost, Effective Method for Increasing Transparency. *PLOS Biology* 14(5):e1002456. doi:<https://doi.org/10.1371/journal.pbio.1002456>.
- [2] Baker, Monya, 2016. Digital badges motivate scientists to share data. *Nature News*. doi:[10.1038/nature.2016.19907](https://doi.org/10.1038/nature.2016.19907).
- [3] Peng, Roger D. 2009. Reproducible research and Biostatistics. *Biostatistics*, Volume 10, Issue 3, Pages 405–408. doi:[10.1093/biostatistics/kxp014](https://doi.org/10.1093/biostatistics/kxp014).
- [4] Peng, Roger D. 2011. Reproducible Research in Computational Science. *Science* 334 (6060): 1226–27. doi:[10.1126/science.1213847](https://doi.org/10.1126/science.1213847).
- [5] Lee, Duncan, Ferguson, Claire, and Mitchell, Richard. 2009. Air pollution and health in Scotland: a multicity study. *Biostatistics*, Volume 10, Issue 3, Pages 409–423, doi:[10.1093/biostatistics/kxp010](https://doi.org/10.1093/biostatistics/kxp010).
- [6] Nüst, D., Konkol, M., Pebesma, E., Kray, C., Schutzeichel, M., Przibytzin, H., and Lorenz, J. Opening the Publication Process with Executable Research Compendia. *D-Lib Magazine*. 2017. doi:[10.1045/january2017-nuest](https://doi.org/10.1045/january2017-nuest).

useR!2017

07 Jul 2017 | By Daniel Nüst



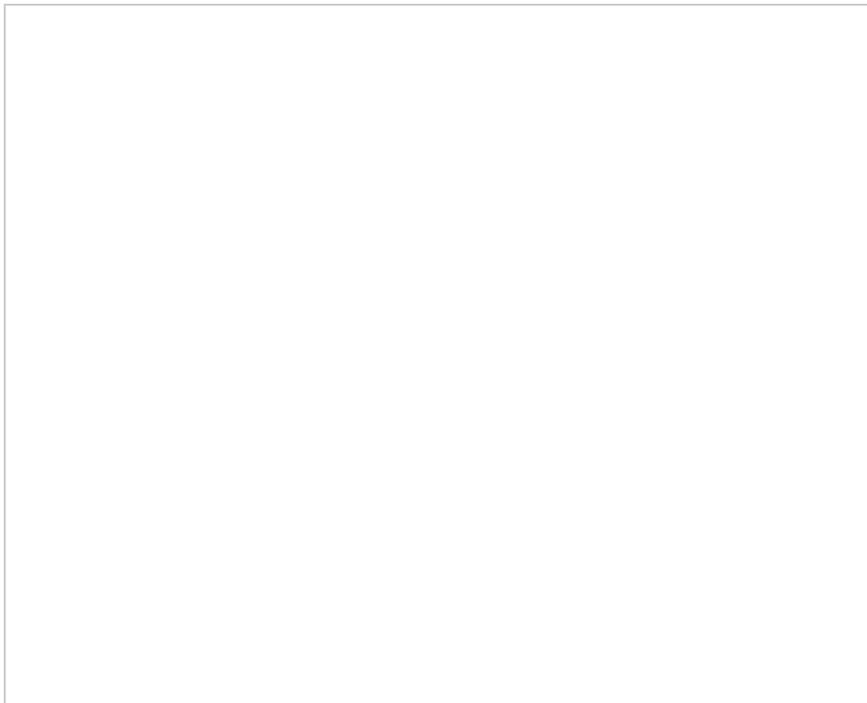
This o2r team members [Daniel](#) and [Edzer](#) had the pleasure to participate in the largest conference of R developers and users, [useR!2017](#) in Brüssel, Belgium.

Daniel Nüst [@nordholmen](#) presenting containerit, creates a docker img from an R session to archive reproducibly [@o2r_project](#) [@cboettig](#) pic.twitter.com/o65O8s8jXY

— Edzer Pebesma ([@edzerpebesma](#)) 6. Juli 2017

Daniel presented a new R extension package, [containerit](#), in the *Data reproducibility* session. It can automatically create a container manifest, i.e. a Dockerfile, from different sources, such as sessions or scripts.

If you want to learn more about [containerit](#), read [this blog post](#) and take a look at Daniel's presentation (also on [Zenodo](#)).



[containerit at useR!2017 conference, Brussels](#) from [Daniel Nüst](#)

Fortunately the presentation was very well-attended and assured our understanding that the importance of reproducibility is widespread in the R community. The interest in using containers for this challenge is growing, as shown by the numerous questions Daniel received after the session and the remainder of the conference.

[containerit](#) is Open Source Software and we invite you to [try it out](#), [inform us about bugs](#), and even [participate in the development](#). In the near future, we will use the package to automatically create [Executable Research Compendia](#) in our [reproducibility service](#), but the package also has an [independent roadmap](#) and it hopefully proves useful for many useRs outside of our project.

The workshop presentations were recorded and are published on [Channel 9](#), including [Daniel's talk](#):

C4RR workshop in Cambridge

28 Jun 2017 | By Daniel Nüst

Today o2r team member Daniel had the pleasure to present work from the o2r project at the two day [Docker Containers for Reproducible Research Workshop](#) held in Cambridge, UK.

It was a full packed [two days of talks and demos](#) (see also [#C4RR](#)). People from a large [variety of disciplines](#) shared how they use containers for making research transparent, scalable, transferable, and reproducible.

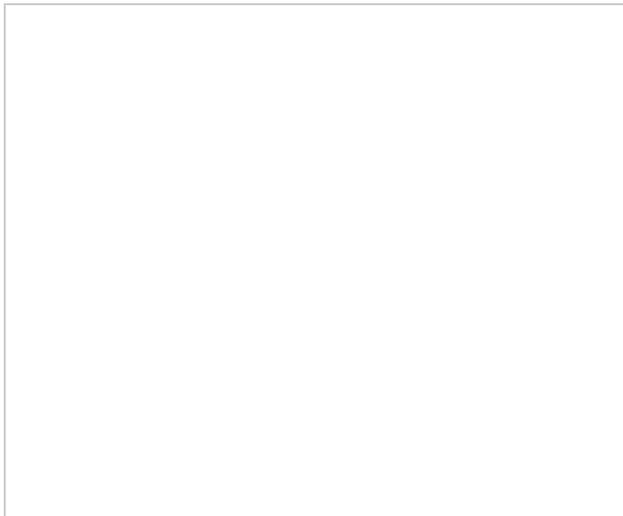
Getting ready for two exciting days at [#C4RR](#) workshop in Cambridge. [@nordholmen](#) presenting o2r tomorrow afternoon. [@SoftwareSaved](#) pic.twitter.com/lhYqmjddPD

— o2r (@o2r_project) 27. Juni 2017

Unlike the workshop's title, Docker was not the only container solution. [Singularity](#) made some important appearances, especially with the different groups working with clusters of thousands of nodes in HPC (high performance computing) and GPGPUs (general processing on graphical processing units). Further topics included deep learning, statistical reports by governments, using containers for teaching, scientific workflows in the cloud, virtual clusters and “best before” dates for software.

Daniel had the hard job of giving the final presentation. After all the previous talks, which comprises many different aspects of reproducible research also somehow part of o2r, this was a threatening task and felt a bit like like “imposters syndrome”. However, the commonalities in motivation, challenges, and ideas are also a sign of the increasing popularity for using containers across [diverse domains](#). Eventually it is a very positive fact an event such as C4RR took place in Europe and had more than 50 people in attendance!

Take a look and Daniel's slides and a video recording below.



[Creating Executable Research Compendia to Improve Reproducibility in the Geosciences](#) from [Daniel Nüst](#)

The workshop was a great experience and very well organized by the [Software Sustainability Institute](#). We learned about both related and quite similar projects, but also acknowledged that o2r's focus on "Desktop-sized" data and computing as well as supporting the geosciences domain does set us apart.

Thanks for a great workshop to [@SoftwareSaved](#) [@rgaiacs](#) [@StephenEglen](#) Taking home stickers and many new ideas
[#C4RR](#) [#reproducibleresearch](#) pic.twitter.com/tEMG34drqp

— o2r (@o2r_project) 28. Juni 2017

Generating Dockerfiles for reproducible research with R

30 May 2017 | By Daniel Nüst, Matthias Hinz

This post is the draft of the vignette for a new R package by o2r team members [Matthias](#) and [Daniel](#). Find the original file in the package repository on [GitHub](#).

- 1. Introduction
- 2. Creating a Dockerfile
- 3. Including resources
- 4. Image metadata
- 5. Further customization
- 6. CLI
- 7. Challenges
- 8. Conclusions and future work
- Metadata

1. Introduction

Even though R is designed for open and reproducible research, users who want to share their work with others are facing challenges. Sharing merely the R script or R Markdown document should warrant reproducibility, but many analyses rely on additional resources and specific third party software as well. An R script may produce unexpected results or errors when executed under a different version of R or another platform. Reproducibility is only assured by providing complete setup instructions and resources. Long-term reproducibility can be achieved by either regular maintenance of the code, i.e. keeping it always working with the latest package versions from CRAN. It can be supported by packages such as [packrat](#) and platforms such as [MRAN](#), which provide means to capture a specific combination of R packages. An alternative to updating or managing packages explicitly is providing the full runtime environment in its original state, using [virtual machines](#) or [software containers](#).

The R extension package [containerit](#) aims to facilitate the latter approach by making reproducible and archivable research with containers easier. The development is supported by the DFG-funded project Opening Reproducible Research (o2r, <https://o2r.info>). [containerit](#) relies on [Docker](#) and automatically generates a container manifest, or “recipe”, with setup instructions to recreate a runtime environment based on a given R session, R script, R Markdown file or workspace directory. The resulting [Dockerfile](#) can not only be read and understood by humans, but also be interpreted by the Docker engine to create a software container containing all the R packages and their system dependencies. This way all requirements of an R workflow are packaged in an executable format.

The created Dockerfiles are based on the [Rocker](#) project ([Rocker on Docker Hub](#), [introduction](#)). Using the stack of version-stable Rocker images, it is possible to match the container’s R version with the local R installation or any R version the user requires. [containerit](#) executes the provided input workspace or file first locally on the host machine in order to detect all dependencies. For determining external software dependencies of attached packages, [containerit](#) relies (a) on the [sysreqs database](#) and makes use of the corresponding web API and R package, and (b) on internally defined rule sets for challenging configurations.

The Dockerfile created by [containerit](#) can then be used to build a Docker image. Running the image will start an R session that closely resembles the creating systems runtime environment. The image can be shared and archived and works anywhere with a compatible Docker version.

To build images and run containers, the package integrates with the [harbor](#) package and adds a few convenience functions for interacting with Docker images and containers. For concrete details on reading, loading, or installing the exact versions of R packages including their system dependencies/libraries, this project focuses on the geospatial domain. [containerit](#) uses the package [futile.logger](#) to provide information to the user at a configurable level of detail, see [futile.logger documentation](#).

In the remainder of this vignette, we first introduce the main usage scenarios for [containerit](#) and document current challenges as well as directions for future work.

2. Creating a Dockerfile

2.1 Basics

The easiest way to generate a Dockerfile is to run an analysis in an interactive R session and create a Dockerfile for this session by loading [containerit](#) and calling the `dockerfile()` - method with default parameters. As shown in the example below, the result can be pretty-printed and written to a file. If no `file` argument is supplied to `write()`, the Dockerfile is written to the current working

directory as `./Dockerfile` , following the typical naming convention of Docker.

When packaging any resources, it is essential that the R working directory is the same as the build context, to which the Dockerfile refers. All resources must be located below this directory so that they can be referred to by relative paths (e.g. for copy instructions). This must also be considered when packaging R scripts that use relative paths, e.g. for reading a file or sourcing another R script.

2.2 Packaging an interactive session

```
library("containerit")

##
## Attaching package: 'containerit'

## The following object is masked from 'package:base':
##
## Arg

# do stuff, based on demo("krige")
library("gstat")
library("sp")

data(meuse)
coordinates(meuse) = ~x+y
data(meuse.grid)
gridded(meuse.grid) = ~x+y
v <- variogram(log(zinc)~1, meuse)
m <- fit.variogram(v, vgm(1, "Sph", 300, 1))
plot(v, model = m)

# create Dockerfile representation
dockerfile_object <- dockerfile()

## INFO [2017-05-30 14:49:20] Trying to determine system requirements for the package(s) 'sp, gstat, knitr, Rcpp, intervals, lattice, FNN, spacetime, zoo, digest, rprojroot, futile.options, backports, magrittr, evaluate, stringi, futile.logger, xts, rmarkdown, lambda.r, stringr, yaml, htmltools' from sysreq online DB
## INFO [2017-05-30 14:49:21] Adding CRAN packages: sp, gstat, knitr, Rcpp, intervals, lattice, FNN, spacetime, zoo, digest, rprojroot, futile.options, backports, magrittr, evaluate, stringi, futile.logger, xts, rmarkdown, lambda.r, stringr, yaml, htmltools
## INFO [2017-05-30 14:49:21] Created Dockerfile-Object based on sessionInfo
```

The representation of a Dockerfile in R is an instance of the S4 class `Dockerfile` .

```
dockerfile_object

## An object of class "Dockerfile"
## Slot "image":
## An object of class "From"
## Slot "image":
## [1] "rocker/r-ver"
##
## Slot "postfix":
## An object of class "Tag"
## [1] "3.4.0"
##
##
## Slot "maintainer":
## An object of class "Label"
## Slot "data":
## $maintainer
## [1] "daniel"
##
##
## Slot "multi_line":
## [1] FALSE
##
##
## Slot "instructions":
## [[1]]
## An object of class "Run_shell"
## Slot "commands":
## [1] "export DEBIAN_FRONTEND=noninteractive; apt-get -y update"
```

```
## [2] "apt-get install -y pandoc \\n\tpandoc-citeproc"
##
##
## [[2]]
## An object of class "Run"
## Slot "exec":
## [1] "install2.r"
##
## Slot "params":
## [1] "-r 'https://cloud.r-project.org'" "sp"
## [3] "gstat" "knitr"
## [5] "Rcpp" "intervals"
## [7] "lattice" "FNN"
## [9] "spacetime" "zoo"
## [11] "digest" "rprojroot"
## [13] "futile.options" "backports"
## [15] "magrittr" "evaluate"
## [17] "stringi" "futile.logger"
## [19] "xts" "rmarkdown"
## [21] "lambda.r" "stringr"
## [23] "yaml" "htmltools"
##
##
## [[3]]
## An object of class "Workdir"
## Slot "path":
## [1] "/payload/"
##
##
## Slot "cmd":
## An object of class "Cmd"
## Slot "exec":
## [1] "R"
##
## Slot "params":
## [1] NA
```

The printout below shows the rendered Dockerfile. Its instructions follow a pre-defined order:

1. define the base image
2. define the maintainer label
3. install system dependencies and external software
4. install the R packages themselves
5. set the working directory
6. copy instructions and metadata labels (see examples in later sections)
7. **CMD** instruction (final line) defines the default command when running the container

Note that the maintainer label as well as the R version of the base image are detected from the runtime environment, if not set to different values manually.

```
print(dockerfile_object)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
&& apt-get install -y pandoc \
  pandoc-citeproc
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "sp", "gstat", "knitr", "Rcpp", "intervals", "lattice", "FNN", "spacetime", "zoo", "digest", "rprojroot", "futile.opti
ons", "backports", "magrittr", "evaluate", "stringi", "futile.logger", "xts", "rmarkdown", "lambda.r", "stringr", "yaml", "htmltools"]
WORKDIR /payload/
CMD ["R"]
```

Instead of printing out to the console, you can also write to a file:

```
write(dockerfile_object, file = tempfile(fileext = ".dockerfile"))
```

```
## INFO [2017-05-30 14:49:21] Writing dockerfile to /tmp/Rtmp25OKLi/file1a9726e56459.dockerfile
```

2.3 Packaging an external session

Packaging an interactive session has the disadvantage that unnecessary dependencies might be added to the Dockerfile and subsequently to the container. For instance the package `futile.logger` is a dependency of `containerit`, and it will be added to the container because it was loaded into the same session were the analyses was executed. It cannot be removed by default, because other packages in the session *might* use it as well (even unintentionally in case of generic methods). Therefore, it is safer not to tamper with the current session, but to run the analysis in an isolated *vanilla* session, which does not have `containerit` in it. The latter will batch-execute the commands in a separate instance of R and retrieves an object of class `sessionInfo`. The session info is then used as input to `dockerfile()`. This is also how `dockerfile()` works internally when packaging either expressions, scripts or R markdown files.

The following code creates a Dockerfile for a list of expressions in a vanilla session.

```
exp <- c(expression(library(sp)),
          expression(data(meuse)),
          expression(mean(meuse[["zinc"]])))
session <- clean_session(exp, echo = TRUE)

## INFO [2017-05-30 14:49:21] Creating an R session with the following arguments:
## R --silent --vanilla -e "library(sp)" -e "data(meuse)" -e "mean(meuse[["zinc"]])" -e "info <- sessionInfo()" -e "save(list = \"info\", file = \"/tmp/Rtmp25OKLi/rdata-sessioninfo1a9714893e92\")"

dockerfile_object <- dockerfile(from = session)

## INFO [2017-05-30 14:49:23] Trying to determine system requirements for the package(s) 'sp, lattice' from sysreq online DB
## INFO [2017-05-30 14:49:24] Adding CRAN packages: sp, lattice
## INFO [2017-05-30 14:49:24] Created Dockerfile-Object based on sessionInfo

print(dockerfile_object)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "sp", "lattice"]
WORKDIR /payload/
CMD ["R"]
```

2.4 Packaging an R script

R scripts are packaged by just supplying the file path or paths to the argument `from` of `dockerfile()`. They are automatically copied into the container's working directory. In order to run the R script on start-up, rather than an interactive R session, a `CMD` instruction can be added by providing the value of the helper function `CMD_Rscript()` as an argument to `cmd`.

```
# create simple script file
scriptFile <- tempfile(pattern = "containerit_", fileext = ".R")
writeLines(c("library(rgdal)",
            'nc <- rgdal::readOGR(system.file("shapes/", package="maptools"), "sids", verbose = FALSE)',
            'proj4string(nc) <- CRS("+proj=longlat +datum=NAD27")',
            'plot(nc)', scriptFile)

# use a custom startup command
scriptCmd <- CMD_Rscript(basename(scriptFile))

# create Dockerfile for the script
dockerfile_object <- dockerfile(from = scriptFile, silent = TRUE, cmd = scriptCmd)

print(dockerfile_object)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
&& apt-get install -y gdal-bin \
  libgdal-dev \
  libproj-dev
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "rgdal", "sp", "lattice"]
WORKDIR /payload/
COPY [".", "."]
```

```
CMD ["R", "--vanilla", "-f", "containerit_1a977e2dcdea.R"]
```

2.5 Packaging an R Markdown file

Similarly to scripts, R Markdown files can be passed to the `from` argument. In the following example, a vignette from the Simple Features package `sf` is packaged in a container. To render the document at startup, the Dockerfile's `CMD` instruction must be changed. To do this, the `cmd` argument passed to `dockerfile()` is constructed using the function `CMD_Render`. Note that, as shown in the Dockerfile, the GDAL library has to be build from source for `sf` to work properly, because a quite recent version of GDAL is required. This adaptation of the installation instruction is based on an internal ruleset for the package `sf`.

```
response <- file.copy(from = system.file("doc/sf3.Rmd", package = "sf"),
  to = temp_workspace, recursive = TRUE)
vignette <- "sf3.Rmd"

dockerfile_object <- dockerfile(from = vignette, silent = TRUE, cmd = CMD_Render(vignette))

## Loading required namespace: sf

print(dockerfile_object)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
&& apt-get install -y gdal-bin \
  libgeos-dev \
  libproj-dev \
  libudunits2-dev \
  make \
  pandoc \
  pandoc-citeproc \
  wget
WORKDIR /tmp/gdal
RUN wget http://download.osgeo.org/gdal/2.1.3/gdal-2.1.3.tar.gz \
&& tar xzf gdal-2.1.3.tar.gz \
&& cd gdal-2.1.3 \
&& ./configure \
&& make \
&& make install \
&& ldconfig \
&& rm -r /tmp/gdal
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "dplyr", "sf", "Rcpp", "assertthat", "digest", "rprojroot", "R6", "DBI", "backports", "magrittr", "evaluate", "units", "rlang", "stringi", "rmarkdown", "udunits2", "stringr", "yaml", "htmltools", "knitr", "tibble"]
WORKDIR /payload/
COPY ["sf3.Rmd", "sf3.Rmd"]
CMD ["R", "--vanilla", "-e", "rmarkdown::render(\"sf3.Rmd\", output_format = rmarkdown::html_document())"]
```

2.6 Packaging a workspace directory

A typical case expected to be interesting for `containerit` users is packaging a local directory with a collection of data and code files. If providing a directory path to the `dockerfile()` function, the package searches for the first occurrence of an R script, or otherwise the first occurrence of an R markdown file. It then proceeds to package this file along with all other resources in the directory, as shown in the next section.

3. Including resources

Analyses in R often rely on external files and resources that are located in the workspace. When scripts or R markdown files are packaged, they are copied by default into the same location relative to the working directory. The argument `copy` influences how `dockerfile()` behaves in this matter. It can either have the values `script` (default behaviour), `script_dir` (copies the complete directory in which the input file is located), or a custom list of files and directories inside the current working directory

```
response <- file.copy(from = system.file("simple_test_script_resources",
  package = "containerit"),
  to = temp_workspace, recursive = TRUE)

dockerfile_object <- dockerfile("simple_test_script_resources",
  copy = "script_dir",
  cmd = CMD_Rscript("simple_test_script_resources/simple_test.R"))
```

```

print(dockerfile_object)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
WORKDIR /payload/
COPY ["simple_test_script_resources", "simple_test_script_resources/"]
CMD ["R", "--vanilla", "-f", "simple_test_script_resources/simple_test.R"]

```

Including R objects works similar to resources, using the argument `save_image`. The argument can be set to `TRUE` to save *all* objects of the current workspace to an `.RData` file, which is then copied to the container's working directory and loaded on startup (based on `save.image()`).

```

df <- dockerfile(save_image = TRUE)
print(df)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
&& apt-get install -y gdal-bin \
  libgeos-dev \
  libproj-dev \
  libudunits2-dev \
  make \
  pandoc \
  pandoc-citeproc \
  wget
WORKDIR /tmp/gdal
RUN wget http://download.osgeo.org/gdal/2.1.3/gdal-2.1.3.tar.gz \
&& tar xzf gdal-2.1.3.tar.gz \
&& cd gdal-2.1.3 \
&& ./configure \
&& make \
&& make install \
&& ldconfig \
&& rm -r /tmp/gdal
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "sp", "gstat", "knitr", "Rcpp", "magrittr", "units", "lattice", "rjson", "FNN", "udunits2", "stringr", "xts", "DBI", "lambdabeta", "futile.logger", "htmltools", "intervals", "yaml", "rprojroot", "digest", "sf", "futile.options", "evaluate", "rmarkdown", "stringi", "backports", "spacetime", "zoo"]
WORKDIR /payload/
COPY ["/.RData", "/."]
CMD ["R"]

```

Alternatively, a object names as well as other arguments can be passed as a list, which then are passed to the `save()` function.

```

require(fortunes)

## Loading required package: fortunes

rm(list = ls())
calculation <- 41 + 1
frtn <- fortunes::fortune()
original_sessionInfo <- sessionInfo()

df <- dockerfile(silent = TRUE,
  save_image = list("original_sessionInfo", "frtn"))

print(df)

FROM rocker/r-ver:3.4.0
LABEL maintainer="daniel"
RUN export DEBIAN_FRONTEND=noninteractive; apt-get -y update \
&& apt-get install -y gdal-bin \
  libgeos-dev \
  libproj-dev \
  libudunits2-dev \
  make \
  pandoc \
  pandoc-citeproc \

```

```
wget
WORKDIR /tmp/gdal
RUN wget http://download.osgeo.org/gdal/2.1.3/gdal-2.1.3.tar.gz \
&& tar xzf gdal-2.1.3.tar.gz \
&& cd gdal-2.1.3 \
&& ./configure \
&& make \
&& make install \
&& ldconfig \
&& rm -r /tmp/gdal
RUN ["install2.r", "-r 'https://cloud.r-project.org'", "fortunes", "sp", "gstat", "knitr", "Rcpp", "magrittr", "units", "lattice", "rjson", "FNN", "udunits2", "stringr", "xts", "DBI", "lambda.r", "futile.logger", "htmltools", "intervals", "yaml", "rprojroot", "digest", "sf", "futile.options", "evaluate", "rmarkdown", "stringi", "backports", "spacetime", "zoo"]
WORKDIR /payload/
COPY ["/payload.RData", "/payload.RData"]
CMD ["R"]
```

4. Image metadata

Metadata can be added to Docker images using [Label instructions](#). Label instructions are key-value pairs of arbitrary content. A duplicate key overwrites existing ones. Although it is up to the user how many labels are created, it is recommended to bundle them into one Label instruction in the Dockerfile. Each use of the `Label()` function creates a separate instruction in the Dockerfile.

As shown in section 2, the maintainer label is set by default to the top as the dockerfile and contains the username of the current host system. The maintainer can be changed with the `maintainer` argument of `dockerfile()` :

```
labeled_dockerfile <- dockerfile(from = clean_session(), maintainer = "Jon_Doe@example.com")
```

Labels can be applied to the existing Dockerfile object using the `addInstructions()` function, which adds any newly created instructions to the end of the Dockerfile but before the CMD statement. The `Label()` constructor can be used for creating labels of arbitrary content and works similar to creating named lists in R.

```
# A simple label that occupies one line:
label1 <- Label(key1 = "this", key2 = "that", otherKey = "content")
addInstruction(labeled_dockerfile) <- label1

#label with fixed namespace for all keys
label2 <- Label("name"="A name", "description" = "A description", label_ns = "my.label.ns.")

# A multiline label with one key/value pair per line
label3 <- Label("info.o2r.name" = "myProject_ImgName", "org.label-schema.name"="ImgName",
"yet.another_labelname"="true", multi_line = TRUE)
addInstruction(labeled_dockerfile) <- list(label2, label3)
```

Metadata according to the [Label Schema](#) conventions can be created with a function constructed by the helper factory

`LabelSchemaFactory()` .

```
Label_LabelSchema <- LabelSchemaFactory()
label <- Label_LabelSchema(name = "ImgName", description = "Description of the image", build_date = Sys.time())
addInstruction(labeled_dockerfile) <- label
```

You can also put session information, using either base R or `devtools` , into a label as plain text or as json:

```
addInstruction(labeled_dockerfile) <- Label_SessionInfo(session = clean_session())
addInstruction(labeled_dockerfile) <- Label_SessionInfo(session = devtools::session_info(), as_json = TRUE)
```

The resulting Dockerfile with all the labels:

```
print(labeled_dockerfile)

FROM rocker/r-ver:3.4.0
LABEL maintainer="Jon_Doe@example.com"
```


CLI Examples:

```
containerit --help

# runs the first R markdown or R script file locally
# prints Dockerfile without writing a file
containerit dir -p --no-write

# Packages R-script
# saves a workspace image (-i parameter)
# Writes Dockerfile (overwrite with -f)
# execute the script on start-up
containerit file -ifp --cmd-R-file path/example.R

# Creates an empty R session with the given R commands
# Set R version of the container to 3.3.0
containerit session -p -e "library(sp)" -e "demo(meuse, ask=FALSE)" --r_version 3.3.0
```

7. Challenges

We encountered several challenges during `containerit`'s development. First and foremost, a well known limitation is that R packages don't define system dependencies and do not provide explicit versions for R package dependencies. The `sysreqs` package is a promising approach towards handling system requirements, but so far lists package names but does not provide version information. The `shinyapps-package-dependencies` demonstrate a (currently system dependent) alternative. The high value of R might well lie in the fact that "packages currently on CRAN" should work well with each other.

An unmet challenge so far is the installation of specific versions of external libraries (see [issue](#)). A package like `sf` relies on well-tested and powerful system libraries, see `sf::sf_extSoftVersion()`, which ideally should be matched in the created container.

And of course users may do things that `containerit` cannot capture from the session state "after the analysis is completed", such as detaching packages or removing relevant files, and unknown side-effects might occur.

All software is presumed to be installed and run on the host system. Although it is possible to use deviating versions of R or even create Dockerfiles using sessionInfo-objects created on a different host, this may lead to unexpected errors because the setup cannot be tested locally.

8. Conclusions and future work

`containerit` allows to create and customize Dockerfiles with minimal effort, which are suitable for packaging R analyses in the persistent runtime environment of a software container. So far, we were able to reproduce complete R sessions regarding loaded and attached packages and mitigate some challenges towards reproducible computational research.

Although we are able to package different versions of R, we still do not fully support the installation of specific versions of R packages and external software libraries, which R itself does not support. This should be tested in the future by evaluating version-stable package repositories like MRAN and GRAN or utility packages such as packrat – see the [GitHub issues](#) for the status of these plans or provide your own ideas there.

Related to installing specific versions is support for other package repositories, such as Bioconductor, git, BitBucket, or even local files. For now, it is recommended that users have all software up-to-date when building a software container, as the latest version are installed from CRAN during the image build, to have matching package versions between the creation runtime environment and the container. All Dockerfiles and instructions are adjusted to the Rocker image stack and assume a Debian/Linux operating system. As we are not yet supporting the build of Docker images from scratch, we are restricted to this setup.

The package is a first prototype available via GitHub. While a publication on CRAN is a goal, it should be preceded by feedback from the user community and ideally be accompanied by related packages, such as `harbor`, being available on CRAN, too. The prototype of `containerit` was developed and tested only on Ubuntu/Linux, which should be extended before releasing a stable version on CRAN.

As part of the o2r project, it is planned to integrate `containerit` in a [web service](#) for creating archivable research in form of [Executable Research Compendia \(ERC\)](#). Making `containerit` itself easier to use for end-users is a secondary but worthwhile goal, for example by building a graphical user interface for metadata creation. Country locales are also not supported yet. We may want to support other container OS (e.g. windows container or other Linux distributions) or even containerization solutions such as [Singularity](#) or the [Open Container Initiative's \(OCI\) Image Format](#).

Feedback and contributions are highly welcome on [GitHub](#) or [o2r_project](#) on Twitter.

Metadata

```
sessionInfo()

## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8   LC_NUMERIC=C
## [3] LC_TIME=en_GB.UTF-8     LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_GB.UTF-8 LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_GB.UTF-8   LC_NAME=C
## [9] LC_ADDRESS=C           LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats  graphics  grDevices  utils  datasets  methods  base
##
## other attached packages:
## [1] fortunes_1.5-4  sp_1.2-4  gstat_1.1-5  containerit_0.2.0
## [5] knitr_1.16
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.11  rstudioapi_0.6  magrittr_1.5
## [4] devtools_1.13.1  units_0.4-4  lattice_0.20-35
## [7] rjson_0.2.15  FNN_1.1  udunits2_0.13
## [10] stringr_1.2.0  tools_3.4.0  xts_0.9-7
## [13] grid_3.4.0  DBI_0.6-1  withr_1.0.2
## [16] lambda.r_1.1.9  futile.logger_1.4.3  htmltools_0.3.6
## [19] intervals_0.15.1  yaml_2.1.14  rprojroot_1.2
## [22] digest_0.6.12  sf_0.4-3  futile.options_1.0.0
## [25] memoise_1.1.0  evaluate_0.10  rmarkdown_1.5
## [28] stringi_1.1.5  compiler_3.4.0  backports_1.0.5
## [31] spacetime_1.2-0  zoo_1.8-0
```

State of the project and next steps

17 May 2017 | By Daniel Nüst, Markus Konkol, Marc Schutzeichel

Yesterday the o2r team met for the second time with a group of experts to request feedback on the state of the project.



Image is licensed under a [CC BY-NC-ND 4.0 International](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.

Thanks to the valuable questions and comments by our external partners, the project tasks were assessed and refocused. On top of it, we agreed to collaborate even further and sketched first ideas for putting o2r's prototypes into real-world settings.

This workshop was only possible thanks to our partner's commitment, enthusiasm, and continued interest in the project. Our heartfelt thanks go to

- [Xenia van Edig](#), Business Development Manager, [Copernicus Publications](#),
- [Maarten Cleeren](#), Director of Product Management, Enriched Content at [Elsevier](#), and
- [Tomi Kauppinen](#) from the [Department of Computer Science at Aalto University](#)

As last year, the full day meeting took place in the countryside at the lovely Wersehaus. Unlike last year, we skipped lightning talks and profited from the existing understanding of the project. Instead we dove right into the project's significant progress: survey results which motivated our design decisions, a critical view on the project schedule and completed/open tasks, the [specification for executable research compendia \(ERC\)](#), our [architecture](#), the [API](#), and most importantly the Open Source [reference implementation](#) and its integration with [Sciebo](#) and [Zenodo](#).

Just as intended these topics were merely started as presentations and led to an active discussion. They were evaluated and connected to the partners perspectives, not the least by putting more ambitious goals (*"let's completely change the way scholarly publishing works!"*) into perspective and defining concrete steps ahead to (i) spread understanding of reproducible research, and (ii) show the potential for enhancements by computational reproducibility with ERC. Many valuable insights will keep the o2r team busy in the following weeks.

In the [blog post of the first workshop](#) we included some statements on *what will we understand in two years time that we do not know now?*, and here is the original (left) and updated version:

<i>We have a good understanding of how far the process of creating research compendia can be automated, and what efforts remain for authors or preservationists that must be counterbalanced with incentives.</i>	Our understanding is consolidated in specifications, in well-defined user workflows, and is demonstrated by a reference implementation. On the topic of incentives, the need for a cultural change ("it takes a generation") was re-stated at the workshop but we can better communicate o2r's actual contributions.
<i>We know the potential of user interface bindings as the connecting entity of research compendia.</i>	By conducting a survey and interviews with geoscientists, we identified promising use cases for UI bindings. e.g. change an analysis variable and update a diagram. The conceptual description (an ontology) underlying these use cases is in progress. It is an open question if we can realise a generic solution to generate UI bindings automatically, and how much effort by the author is required.
<i>We show the improvements in discovery and understanding of research when all aspects of research are explicitly linked in a meaningful way.</i>	Thanks to feedback by last year's workshop and continued interaction with other researchers at conferences and workshops, we decided to concentrate on these challenging topics first: easily packaging research into ERC, integrating with data repositories, and interacting with ERC. Therefore discovery is a topic for the second half of 2017, including a recently started master thesis.

We get to know the common language as well as points of contact for the involved parties as we create a closer connection between research, preservation, and publication communities.

Success! The prototypes are received well by all of the parties. They provide unifying concepts and workflows and are even seen as ready for pilot studies.

We hope to have another inspirational meeting like this in 2018! To keep in touch, follow us on [Twitter](#) or [GitHub](#).

Opening Reproducible Research at AGILE 2017 conference in Wageningen

10 May 2017 | By Daniel Nüst

This week o2r participates in another conference, which is partly a repetition but also a contrast to the last one:

Again, o2r team members (Markus and Daniel) are fortunate to co-organize a workshop about reproducible research: “Reproducible Geosciences Discussion Forum” at the 20th AGILE International Conference on Geographic Information Science in Wageningen, The Netherlands, took place yesterday. **Read the short recap on the workshop website.**

Thx! Fun, educational & productive workshop today on [#reproducible](#) [#geosciences](#) at [#agilewag2017](#) [#agile2017nl](#) Report soon via [@o2r_project](#) [pic.twitter.com/MjrWPQyoQ2](#)

— Daniel Nüst (@nordholmen) [May 9, 2017](#)

Daniel will also present a poster on o2r titled “*An Architecture for Reproducible Computational Geosciences*”. **Please visit the AGILE 2017 poster session tomorrow at 15:00** and discuss with us how our [ERC](#) fits into the geosciences landscape.

Update: Download the [abstract](#) and the [poster](#).

agile conference banner

Image courtesy of AGILE website.

Completely different is the scale of this weeks conference: unlike [EGU general assembly](#), AGILE is a small conference with an informal feeling. While the attendees represent diverse topics, the common connection to GI Science is strong and while the programme is packed at times (5 parallel tracks - hard to choose!), there is ample room to focus, for example in the single track keynote or poster sessions, but also to chat and learn, which we hope to do by spreading questions on reproducibility of the works presented at AGILE.

Opening Reproducible Research at EGU General Assembly 2017

04 May 2017 | By Daniel Nüst

Last week the largest European geosciences conference of the year took place in Vienna: the [European Geophysical Union General Assembly 2017](#).

o2r took part by co-organising a workshop on reproducible research and co-convening the session *IE2.4/ESSI3.10 Open Data, Reproducible Research, and Open Science*.

o2r team member Daniel Nüst presented the abstract *“Executable research compendia in geoscience research infrastructures”* ([download poster](#)) and supported Marius Appel and Edzer Pebesma in their work on *“Reproducible Earth observation analytics: challenges, ideas, and a study case on containerized land use change detection”* ([download poster](#)) in the ESSI3.10’s poster session.

It was a great experience to meet fellow scientists interested in, and worried about, reproducibility of scholarly works. We got useful feedback on our practical work and are encouraged again to continue spreading the word on the general topic of reproducibility alongside our research.

Poster in EGU17 final session is ready for business. Come to X4.123 at 17:00 and talk [#openscience](#) [#reproduciblersearch](#)
Survey going well! pic.twitter.com/at2eem44Md

— o2r (@o2r_project) April 28, 2017



Reproducible Research at EGU GA - A short course recap

03 May 2017 | By Daniel Nüst, Vicky Steeves, Rémi Rampin

At last week's EGU general assembly members of the o2r and ReProZip projects organized the short course "*Reproducible computational research in the publication cycle*". This post is a recap of the course by Daniel Nüst, Vicky Steeves, and Rémi Rampin.

All materials for the course are published in an Open Science Framework repository at <https://osf.io/umy6g/> and you can learn about the motivation for the course in the [course page at EGU](#).

The short was divided into two parts: a practical introduction to selected tools supporting computational reproducibility, and talks by stakeholders in the scientific publication process followed by a lively panel discussion.



In the first part, Daniel and Vicky began with sharing some literature on reproducible research (RR) with the roughly 30 participants. After all, the participants should take home something useful, so a reading list seems reasonable for RR newcomers but also for researchers writing about the reproducibility aspects in upcoming papers.

Then Daniel fired up a console and took a deep dive into **using containers to encapsulate environments for reproducible computational research**. He started with a very quick introduction to Docker and then demonstrated some containers useful to researchers, i.e. Jupyter Notebook and RStudio.

The material presented by Daniel is a [starting point for an Author Carpentry lesson](#), which is currently [developed on GitHub](#), so he highly appreciates any feedback, especially by short course attendees. We were surprised to learn a good portion of the participants had already some experience with Docker. But even better was realizing a few actually hacked along as Daniel raced through command-line interface examples! This "raw" approach to packaging research in containers was contrasted in the second section.

.@nordholmen forked author carpentry to make a lesson for us today! About to look at rstudio & jupyter notebooks w/ Docker! #egu2017 pic.twitter.com/ekgYuJPKs6

— Vicky Steeves (@VickySteeves) April 24, 2017

Under the title "**ReProZip for geospatial analyses**", Vicky and Rémi showcased [ReProZip](#), a tool for automatically tracing and packaging scientific analyses for easily achieved computational reproducibility. The resulting file is a ReProZip package (`.rpz`), which can be easily shared due to its small size, and contains everything necessary to reproduce research (input files, environmental information etc.) across different operating systems. They demonstrated their various unpackers and showed how these `.rpz` files can be used for reproducibility and archiving. They also demoed their brand new user interface for the first time in Europe.

The materials presented by Vicky and Rémi are also available on both the Open Science Framework [here](#) and on the [ReProZip examples website](#).

@edzerpebesma @benmarwick @o2r_project And @VickySteeves and @remram44 showing #reprozip pic.twitter.com/4hxpEsmqPN

— Daniel Nüst (@nordholmen) April 24, 2017

The practical demonstrations paved the way for the **second part** of the short course, which was more abstract yet proved to excellently demonstrate the breadth of reproducible research. Selected speakers provided their perspectives on the topic of reproducing scientific papers in the broader context of the scientific publication cycle. In short talks they wore a specific role of the scholarly publication process and shared their experience as a researcher, infrastructure provider, publisher, reviewer, librarian, or editor. The speakers:

- [Edzer Pebesma](#) talked about his experiences as journal editor for [JStatSoft](#) as well as [Computers & Geosciences](#), and his original motivation to enter the area of reproducible research with his [prize-winning "one-click reproduce" concept](#) and initiator of [o2r](#): annoyance by not being able to share the full integrated material of his works easily.

- [Tobias Weigel](#) from the [german national climate computing center](#) introduced the challenges and limitations for a supercomputer facility which provides crucial resources for reproducibility.
- [David Ham](#) shared the priorities of the [journal Geoscientific Model Development \(GMD\)](#) where he is editor, when it comes to reproducibility and the issues they face. Proper provenance and citations are examples for the former, the ephemerality of code and data for the latter.
- [Xenia van Edig](#) lead us through the stages of Open Access that [Copernicus](#) went and is going through as a publisher: from public data (1.0) via interactive articles and public peer review (2.0) to the future of open science and executable papers (3.0)
- [Vicky Steeves](#) advertised the expertise of librarians worldwide in supporting research in all aspects, including reproducibility, writing grants, or data management plans, but also pointed out the necessity to support scientists with proper tools and teach the required skills.
- [Daniel Nüst](#) (research software engineer perspective)

All speakers touched on the topic of *scientific culture*, which was seen in a process of changing towards more openness, but with still quite some way to go. The cultural aspects and larger scale challenges were a recurring topic in the panel discussion after the short talks. These aspects included resistance to share supplemental material, so that journals cannot make sharing everything mandatory, for example because of unwillingness (fear of stealing) or because authors might not be allowed to do so. A member of the audience could share that in their experience as a publisher, requiring data and software publication did not result in a decrease in submissions when accompanied by transparent and helpful author guidelines. Such guidelines for both data and code are lacking for many journals but are a means to improve the overall situation - and make the lives of editors simpler. When the progress of the last years on *Open Data* was pointed out as largely a top down political endeavour, the contrast to *Open Source* as a bottom-up grassroots initiative became clear. Nevertheless, the hope was phrased that with the success of *Open Data*, things might go smoother with *Open Source* in science.



A further topic the discussion covered for some time was *creditation*, and the need to update the ways researchers *get and give* credit as part of grant-based funding and publishing scholarly articles. Though it was pointed out that RR is also about “doing the right thing”. Credit and culture were seen as closely linked topics, which can only be tackled by improving the education of scientists, both as authors and reviewers(!), and spreading the word about the importance of reproducibility for all of science, not least in the light of the marches for sciences taking place just a few days before the short course.

While one could say we were mostly preaching to the choir, it was great to see an interest in the topic of reproducible research amongst EGU attendees. **This workshop being the first of its kind at the EGU general assembly hopefully was a step towards even higher visibility and interest for RR as a crucial topic in today’s research.**

We thank the short course attendees and invited speakers for turning the first afternoon of EGU 2017 into an instructive and diverting few hours. *Will there be a reproducible research short course next year at EGU?* We don’t know yet, but please do get in touch if you would like to support the planning. It could be worth providing a longer course targeted as [early career scientists](#), giving the [next generation](#) the tools to work reproducibly.

Docker for GEOBIA - new article published

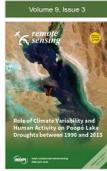
30 Mar 2017 | By Daniel Nüst

We are happy to announce that o2r team member [Daniel](#) published a new article together with [Christian Knoth](#) in the journal Remote Sensing. The special issue “Advances in Object-Based Image Analysis—Linking with Computer Vision and Machine Learning” comprising six papers was published in connection with the 6th GEOBIA conference (2016), where Daniel and Christian’s work was previously honoured with the best student paper award.

The article **Reproducibility and Practical Adoption of GEOBIA with Open-Source Software in Docker Containers** is available as Open Access:
doi:10.3390/rs9030290



Knoth, C., Nüst, D., 2017. Reproducibility and Practical Adoption of GEOBIA with Open-Source Software in Docker Containers. Remote Sensing 9, 290. doi:10.3390/rs9030290



Remote Sens. 2017, 9(3), 290; doi:10.3390/rs9030290 (page/figure DOI)

Reproducibility and Practical Adoption of GEOBIA with Open-Source Software in Docker Containers

Christian Knoth ¹ and Daniel Nüst ^{1*}

Institute for Geoinformatics, University of Münster, Hohenringstraße 2, 48149 Münster, Germany

* Author to whom correspondence should be addressed.

Academic Editor: Norman Kerle, Markus Gerke, Sebastian Lefrow and Prasad S. Thenkabail

Received: 30 December 2016 / Revised: 22 February 2017 / Accepted: 6 March 2017 / Published: 18 March 2017

(This article belongs to the Special Issue Advances in Object-Based Image Analysis—Linking with Computer Vision and Machine Learning)

View Full Text | Download PDF (5561 KB, updated 20 March 2017) | Browse Figures

Article Versions

- Abstract
- Full-Text HTML
- Full-Text HTML (PDF)
- Full-Text XML
- Full-Text ePub
- Article Versions Notes

Related Info

- Article Statistics
- Google Scholar
- CrossRef

More by Authors

- on DOI
- on Google Scholar
- on PubMed

Export Article

- HTML
- EndNote
- RIS

Alerts

See more details

- Twittered by 3
- 1 readers on Mendeley

Facebook | LinkedIn | Twitter | Mendeley

Abstract

Geographic Object-Based Image Analysis (GEOBIA) mostly uses proprietary software. This hinders access not only from cost barriers, but also from benefits concerning reproducibility and collaboration. Technical challenges hamper practical reproducibility, especially when multiple software packages are required to conduct an analysis. In this study, we use containerisation to package a GEOBIA workflow in a well-defined FOSS environment. We explore the approach using test software stacks to perform an exemplary analysis detecting destruction of buildings in historical images of a conflict area. The analysis combines feature extraction techniques with segmentation and object-based analysis to detect changes using automatically defined local reference systems and to distinguish disappeared buildings from non-target structures. The resulting workflow is published as FOSS comprising both the model and data in a ready-to-use Docker image and a user interface for interaction with the containerised workflow. The presented solution enhances GEOBIA in the following aspects: higher transparency of methodology; easier reuse and adaptation of workflow; better transferability between operating systems; complete description of the software environment; and easy replication of workflow by image analysis experts and non-experts. As a result, it promotes and aids the reproducibility of GEOBIA, but also its practical adoption. View Full Text

Keywords: reproducibility; GEOBIA; Docker; workflow; containerisation; reproducible research; object-based image analysis; GIS; containerisation

Figures



Figure 4

This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. (CC BY 4.0)

o2r @ Open Science Conference 2017, Berlin

24 Mar 2017 | By Markus Konkol



Foto: open-science-conference.eu

Due to the overall topic of our project, we felt the [Open Science Conference \(#osc2017\)](#) taking place this week in Berlin would be a great chance to share our ideas and meet like-minded folks. We were happy about the notification that our poster was accepted and even made it into the top ten (of altogether 57 submissions), which allowed o2r team member Markus to give a three-minute [lightning talk](#) and present a project [poster](#). Both days included interesting talks given by international speakers (see [full programme](#)) and in this post Markus reports on the trip. The first day covered several topics related to o2r, for example, data infrastructures (see [European Open Science Cloud](#)). Speakers also mentioned social challenges such as a new reward system and incentives required to motivate scientists to conduct Open Science – a key issue in the Executable Research Compendium-concept as well. In times of *fake news* and the *credibility crisis*, [keynote](#) speaker Prof. Johannes Vogel strongly encouraged in his opening talk to set a good example in the field of Open Science and convincingly put scientists in charge of the issue.

A few people I talked to liked the idea of making the dataset the actual publication and the paper being “only” the supplementary material. It might be interesting to play around with some thoughts on that: Will institutes focus on publishing datasets instead of papers? Is “data collector” a new job title?

The lightning talks and the poster session were a success. Several visitors were keen to ask questions and to get explanations on technical and conceptual details. I hope that I was able to answer all of them in sufficient detail. If you think I didn't, please don't hesitate to ask me or in case of doubts, my colleagues Docker Daniel and Metadata Marc. You should also take a look at [Conquaire](#), an interesting project in the context of reproducible research.

One highlight was a visit by [Open Science Radio](#), who also published a [short interview on opening reproducible research](#).

In the evening, we had a wonderful dinner next to dinosaurs (I am not talking about the scientists 😊) organized by [Museum für Naturkunde Berlin](#), a museum of natural science. In this impressive atmosphere, we were able to network a bit and to continue discussions.

The second day was rather education-driven. However, we do also want to enhance and extend the understanding of scientists when examining a paper by using our ERC. Why not addressing students, too? We still dream of a reproducible and interactive atlas.

It was interesting to see that the great majority of guests and speakers focused on open data when discussing challenges in Open Science. Mentioning source-code was rather the exception although reproducibility was perceived as being part of Open Science. For this reason, I think that our contribution to the conference was relevant as we treat (open) code and software as being equally important. I mentioned this aspect in my lightning talk, too, and tried to highlight the importance of source code during the poster presentation. One might argue that open code is implicitly included in open data or open methodology. However, we should not rely on vague interpretations and make explicit what is required to rerun analyses. In the future, submitting, for example, analysis scripts should be as mandatory as it is demanded for datasets.

To conclude, here a few **take home messages**:

1. Rewards and incentives that motivate to conduct Open Science are key issues
2. We have to engage people from society to increase trust in scientific results (tackle credibility crisis)
3. Problems are social – not technical. BUT: we have to provide scientists with working examples, otherwise they don't know why to use it and how.
4. Open Science strongly focuses on data and educational aspects.

P.S. The next time you read about guidelines, recommendations on open data, try to replace it by source code. The argument still works, right?

EGU short course scheduled and session programme upcoming

14 Feb 2017 | By Daniel Nüst

Join our short course "#Reproducible #computational #research in the publication cycle" at #EGU2017 #SC81
<https://t.co/zPbvGUDsCy>

— o2r (@o2r_project) January 23, 2017

The short course **Reproducible computational research in the publication cycle** (SC81) at the [EGU general assembly](#) was accepted by the short course programme group and scheduled **Monday, April 24th, 2017** in the afternoon. Thanks!

We are grateful for the change to share our work on reproducible research together with members of the [ReproZip](#) team in a practical, hands-on short course. Afterwards we welcome a number of esteemed speakers to share their views on reproducibility as researcher, reviewer, editor, publisher, and preservationist.

See the [full session description](#) in the EGU programme.

Please register for the short course for free by filling in your name and email in this Doodle poll:
<http://doodle.com/poll/2yvi7y9tine2x3pf>.

Earlier this year we also announced [acall for a session on reproducibility at EGU general assembly](#). The contributions to this session was merged with other sessions to create the session **IE2.4/ESSI3.10 Open Data, Reproducible Research, and Open Science**, see [session description in the EGU GA programme](#)

The session programme will be published March 1st, 2017, so stay tuned for the official announcements.

D-Lib Magazine Article Published

16 Jan 2017 | By Daniel Nüst

We are happy to announce that our article **Opening the Publication Process with Executable Research Compendia** is now published in D-Lib Magazine's current issue:

<https://doi.org/10.1045/january2017-nuest>

This paper was originally presented at the [RepScience Workshop](#) in September 2016 and was peer-reviewed as part of the workshop submission. It is published as Open Access along with [other papers from the conference](#).

Nüst, D., Konkol, M., Pebesma, E., Kray, C., Schutzzeichel, M., Przybytzin, H., Lorenz, J., 2017. Opening the Publication Process with Executable Research Compendia. D-Lib Magazine 23. doi:10.1045/january2017-nuest



The screenshot shows the D-Lib Magazine website interface. At the top, there is a blue header with the 'D-Lib Magazine' logo and the subtitle 'The Magazine of Digital Library Research'. Below the header is a navigation bar with links for HOME, ABOUT D-LIB, CURRENT ISSUE, ARCHIVE, INDEXES, CALENDAR, and AUTHORS. The main content area displays the article title 'Opening the Publication Process with Executable Research Compendia' and lists the authors: Daniel Nüst*, Markus Konkol*, Marc Schutzzeichel, Edzer Pebesma, Christian Kray, Holger Przybytzin, and Jörg Lorenz. Each author's name is followed by their affiliation and email address. A note at the bottom indicates that * denotes shared co-first authorship.

Reproducible Computational Geosciences Workshop at AGILE Conference

05 Jan 2017 | By Daniel Nüst

We are happy to announce that a pre-conference workshop “Reproducible Computational Geosciences” at the [20th AGILE International Conference on Geographic Information Science](#) will be held on May 9 2017 in Wageningen, The Netherlands.

With this half day workshop we want to introduce the topic of reproducible research to the AGILE conference series, the most prominent and long-standing GIScience and GIS conference in Europe. The 3-day conference is accompanied by [13 workshops on diverse topics](#).

agile conference banner

Image courtesy of AGILE website.

Submit your abstract [here](#) and share your experiences in reproducibility of geospatial analysis. Challenges, reproducibility studies, archiving, educational or legal aspects are among the welcomed topics.

The workshop is co-organized by o2r team members and Frank Osterman from ITC, Enschede. Contributions and a public peer review are done via GitHub and supported by a great programme committee of distinguished researchers. *Please share this information with potentially interested parties (and [retweet](#)). Thanks!*

We look forward to your submission!

Investigating Docker and R

15 Dec 2016 | By Daniel Nüst

This post gave the idea for the following

PUBLISHED PAPER

. It will not be updated anymore

The Rockerverse: Packages and Applications for Containerisation with R

Daniel Nüst, Dirk Eddelbuettel, Dom Bennett, Robrecht Cannoodt, Dav Clark, Gergely Daróczi, Mark Edmondson, Colin Fay, Ellis Hughes, Lars Kjeldgaard, Sean Lopp, Ben Marwick, Heather Nolis, Jacqueline Nolis, Hong Ooi, Karthik Ram, Noam Ross, Lori Shepherd, Péter Sólymos, Tyson Lee Swetnam, Nitesh Turaga, Charlotte Van Petegem, Jason Williams, Craig Willis and Nan Xiao. *The R Journal* (2020) 12:1, pages 437-461. doi:10.32614/RJ-2020-007

This post is regularly updated (cf. [GH issue](#)) and available under the URL <http://bit.ly/docker-r>. Last update: 11 Jan 2018.

Docker and R: How are they used and could they be used together? That is the question that we regularly ask ourself. And we try to keep up with other people's work! In this post, we are going to share our insights with you.



Thanks to [Ben Marwick](#) for *contributing* to this post! You know about a project using Docker and R? [Get in touch](#).

Dockerising R

Several implementations of besides the one by R-core exist today, together with numerous integrations into open source and proprietary software (cf. [English](#) and [German](#) Wikipedia pages). In the following we present the existing efforts for using *open source* R implementation with Docker.

Rocker

The most prominent effort in this area is the **Rocker** project (<http://rocker-project.org/>). It was initiated by [Dirk Eddelbuettel](#) and [Carl Boettiger](#) and containerises the main R implementation based on [Debian](#). For an introduction, you may read their blog post [here](#) or follow [this tutorial](#) from rOpenSci.

With a big choice of pre-build Docker images, Rocker provides optimal solutions for those who want to run R from Docker containers. Explore it on [Github](#) or [Docker Hub](#), and soon you will find out that it takes just one single command to run instances of either [base R](#), [R-devel](#) or [Rstudio Server](#). Moreover, you can run [specific versions of R](#) or use one of the many bundles with commonly used R packages and other software, namely [tidyverse](#) and [rOpenSci](#).

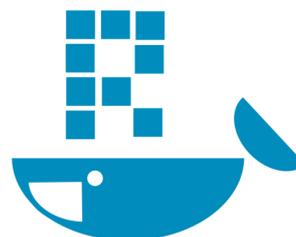
Images are build monthly on Docker Hub, except *devel* tags which are build nightly. Automated builds are disabled, instead builds are triggered by CRON jobs running on a third party server (cf. [GitHub comment](#)).

Bioconductor

If you come from bioinformatics or neighboring disciplines, you might be delighted that **Bioconductor** provides several images based on Rocker's [rocker/rstudio](#) images. See the [help page](#), [GitHub](#), and [Open Hub](#) for more information. In short, the Bioconductor core team maintains *release* and *devel* images (e.g. [bioconductor/release_base2](#)), and contributors maintain image with different levels of pre-installed packages (each in *release* and *devel* variants), which are based on Bioconductor views (e.g. [bioconductor/devel_proteomics2](#) installs the views [Proteomics](#) and [MassSpectrometryData](#)).

Image updates occur with each Bioconductor release, except the *devel* images which are build weekly with the latest versions of R and Bioconductor based on [rocker/rstudio-daily](#).

CentOS-based R containers



Jonathan Lisic works on a collection of Dockerfiles building on CentOS (6 and 7) and other operating systems as an alternative to the Debian-based Rocker stack. The Dockerfiles are on GitHub: <https://github.com/jlisic/R-docker-centos>

MRO

Microsoft R Open (MRO) is an “enhanced R distribution”, formerly known as Revolution R Open (RRO) before [Revolution Analytics](#) was acquired by Microsoft. MRO is compatible with main R and its packages. “It includes additional capabilities for improved performance, reproducibility, and platform support.” (source); most notably these are the [MRAN repository](#) a.k.a. CRAN Time Machine, which is also used by versioned Rocker images, and the (optional) integration with [Intel® Math Kernel Library \(MKL\)](#) for [multi-threaded performance](#) in linear algebra operations ([BLAS](#) and [LAPACK](#)).



o2r team member Daniel created a Docker image for MRO including MKL. It is available on [Docker Hub](#) as [nuest/mro](#), with [Dockerfile on GitHub](#). It is inspired by the Rocker images and can be used in the same fashion. Please note the extended licenses printed at every startup for MKL.

Jonathan Lisic published a Dockerfile for a CentOS-based MRO on [GitHub](#).

Ali Zaidi published [Dockerfiles on GitHub](#) and [images on Docker Hub](#) for [Microsoft R Client](#), which is based on MRO.

R Client adds to MRO by including a couple of “ScaleR” machine learning algorithms and packages for parallelisation and remote computing.

Renjin

Renjin is a JVM-based interpreter for the R language for statistical computing developed by [BeDataDriven](#). It was developed for big data analysis using existing R code seamlessly in cloud infrastructures, and allows Java/Scala developers to easily combine R with all benefits of Java and the JVM.



While it is not primarily build for interactive use on the command line, this is possible. So o2r team member Daniel created a Docker image for Renjin for you to try it out. It is available [on Docker Hub](#) as [nuest/renjin](#), with [Dockerfile on GitHub](#).

pqr

pqr tries to create “a pretty quick version of R” and fixing some perceived issues in the R language. While this is a one man project by [Radford Neal](#), it’s worth trying out such contributions to the open source community and to the discussion on how R should look like in the future (cf. [a recent presentation](#)), even if things might get [personal](#). As you might have guess by now, Daniel created a Docker image for you to try out pqr: It is available [on Docker Hub](#) as [nuest/pqr](#), with [Dockerfile on GitHub](#).

[WIP] FastR

Also targeting performance, **FastR** is “is an implementation of the R Language in Java atop [Truffle](#), a framework for building self-optimizing AST interpreters.” FastR is planned as a drop-in replacement for R, but [relevant limitations](#) apply.

While GraalVM has a [Docker Hub user](#), no images are published probably because of licensing requirements, as can be seen in the GitHub repository [oracle/docker-images](#), where users must manually download a GraalVM release, which requires an Oracle Account... so the current tests available in [this GitHub repository](#), trying to build FastR from source based on the newest OpenJDK Java 9.

Dockerising Research and Development Environments

So why, apart from the incredibly easy usage, adoption and transfer of typical R environments, would you want to combine R with Docker?

Ben Marwick, Associate Professor at the University of Washington, explains in [this presentation](#) that it helps you manage dependencies. It gives a computational environment that is isolated from the host, and at the same time transparent, portable, extendable and reusable. Marwick uses Docker and R for **reproducible research** and thus bundles up his works to a kind of *Research Compendium*; an instance is available [here](#), and a template [here](#).

Carl Boettiger, Assistant Professor at UC Berkeley, wrote in detail about using Docker for reproducibility in his ACM SIGOPS paper 'An introduction to Docker for reproducible research, with examples from the R environment'.



Both Ben and Carl contributed case studies using Docker for research compendia in the book "The Practice of Reproducible Research - Case Studies and Lessons from the Data-Intensive Sciences": Using R and Related Tools for Reproducible Research in Archaeology and A Reproducible R Notebook Using Docker.

An R extension you may want to dockerise is **Shiny**. Flavio Barros dedicated two articles on R-bloggers to this topic: [Dockerizing a Shiny App](#) and [Share Shiny apps with Docker and Kitematic](#). The majority of talks at [useR!2017](#) presenting [real-world deployments of Shiny](#) mentioned using dockerised Shiny applications for reasons of scalability and ease of installation.

The company [Seven Bridges](#) provides an example for a public container encapsulating a specific research environment, in this case the product [Seven Bridges Platform](#) ("a cloud-based environment for conducting bioinformatic analyses"), its tools and the Bioconductor package [sevenbridges](#). The published image [sevenbridges/sevenbridges-r](#) includes both RStudio Server and Shiny, see the [vignette "IDE Container"](#).

A new solution to ease the creation of Docker containers for specific research environments is [containerit](#). It creates [Dockerfile](#)s (using [Rocker](#) base images) from R sessions, R scripts, R Markdown files or R workspace directories, including the required system dependencies. The package was [presented at useR!2017](#) and can currently only be installed from [GitHub](#).

While Docker is made for running tools and services, and providing user interfaces via web protocols (e.g. via a local port and a website opened in a browser, as with [rocker/rstudio](#) or Jupyter Notebook images), several activities exist that try to package **GUI applications in containers**. Daniel explores some alternatives for running RStudio in [this GitHub repository](#), just for the fun of it. In this particular case it may not be very sensible, because *RStudio Desktop* is already effectively a browser-based UI (unlike other GUI-based apps packages this way), but for users with reluctance to a browser UI and/or command line interfaces, the "Desktop in a container" approach might be useful.

Running Tests

The package [dockertest](#) makes use of the isolated environment that Docker provides: R programmers can set up test environments for their R packages and R projects, in which they can rapidly test their works on Docker containers that only contain R and the relevant dependencies. All of this without cluttering your development environment.

The package [gitlabr](#) does not use Docker itself, but wraps the [GitLab API](#) in R functions for easy usage. This includes starting continuous integration (CI) tests (function [gl_ci_job](#)), which [GitLab can do using Docker](#), so the function has an argument [image](#) to select the image run to perform a CI task.

In a completely different vein but still in the testing context, [sanitizers](#) is an R package for testing the compiler setup across different compiler versions to detect code failures in sample code. This allows testing completely different environments on the same host, without touching the well-kept development environment on the host. The packages' images are now *deprecated* and superseded by [Rocker](#) images ([rocker/r-devel-san](#) and [rocker/r-devel-ubsan-clang](#)).

Dockerising Documents and Workflows

Some works are dedicated to *dockerising R-based documents*.

The package [liftr](#) (on [CRAN](#)) for R lets users enhance Rmd files with YAML-metadata ([example](#)), which enables rendering R Markdown documents in Docker containers. Unlike [containerit](#), this metadata must be written by the author of the R Markdown document.



[liftr](#) is used in the [DockFlow](#) initiative to containerise a selection of [Bioconductor workflows](#) as presented in [this poster](#) at [BioC 2017](#) conference. [Liftr](#) also supports [Rabix](#), a Docker-based toolkit for portable bioinformatics workflows. That means that users can have [Rabix](#) workflows run inside the container and have the results integrated directly into the final document.

The Bioconductor package [sevenbridges](#) (see also above) has a [vignette on creating reproducible reports with Docker](#). It recommends a reproducible script or report with [docopt](#) respectively R markdown (parametrised reports). The cloud-based [Seven Bridges](#) platform can fulfill requirements, such as required Docker images, within their internal JSON-based workflow and "Tool"

description format ([example](#)), for which the package provides helper functions to create Tools and execute them, see [this example in a vignette](#). Docker images are used for [local testing of these workflows](#) based on Rabix (see above), where images are started automatically in the background for a user, who only uses R functions. Automated builds for workflows on Docker Hub are also encouraged.

RCloud is a collaborative data analysis and visualization platform, which you can not only try out online but also host yourself with Docker. Take a look at [their Dockerfiles](#) or try out their image `rcl0ud/rcloud`.

Control Docker Containers from R

Rather than running R inside Docker containers, it can be beneficial to call Docker containers from inside R. This is what the packages in this section do.

The `harbor` package for R (only available via GitHub) provides all Docker commands with R functions. It may be used to control Docker containers that run either locally or remotely.

A more recent alternative to `harbor` is the package `docker`, also available on CRAN with source code on GitHub. Using a DRY approach, it provides a thin layer to the Docker API using the Docker SDK for Python via the package `reticulate`. The package is best suited for apt Docker users, i.e. if you know the Docker commands and life cycle. However, thanks to the abstraction layer provided by the Docker SDK for Python, `docker` also runs on various operating systems (including Windows).

`dockermachine` provides a convenient R interface to the `docker-machine` command, so you can provision easily local or remote/cloud instances of containers.

Selenium provides tools for browser automation, which are also available as Docker images. They can be used, amongst others, for testing web applications or controlling a headless web browser from your favorite programming language. In [this tutorial](#), you can see how and why you can use the package `RSelenium` to interact with your Selenium containers from R.

`googleComputeEngineR` provides an R interface to the Google Cloud Compute Engine API. It includes a function called `docker_run` that starts a Docker container in a Google Cloud VM and executes R code in it. Read [this article](#) for details and examples. There are similar ambitions to implement Docker capabilities in the `analogsea` package that interfaces the Digital Ocean API. `googleComputeEngineR` and `analogsea` use functions from `harbor` for container management.

R and Docker for Complex Web Applications

Docker, in general, may help you to build complex and scalable web applications with R.

If you already have a Shiny app, then Cole Brokamp's package `rise` makes you just one function call away from building and viewing your dockerised Shiny application.

If you want to get serious with Shiny, take a look at ShinyProxy by Open Analytics. ShinyProxy is a Java application (see GitHub) to deploy Shiny applications. It creates a container with the Shiny app for each user to ensure scalability and isolation and has some other "enterprise" features.

Mark McCahill presented at an event of the Duke University in North Carolina (USA) how he provided 300+ students each with private RStudio Server instances. In his presentation ([PDF / MOV \(398 MB\)](#)), he explains his **RStudio farm** in detail.

If you want to use **RStudio with cloud services**, you may find delight in these articles from the SAS and R blog: [RStudio in the cloud with Amazon Lightsail and docker](#), [Set up RStudio in the cloud to work with GitHub](#) [RStudio in the cloud for dummies, 2014/2015 edition](#).

The platform **R-hub** helps R developers with solving package issues prior to submitting them to CRAN. In particular, it provides services that build packages on all CRAN-supported platforms and checks them against the latest R release. The services utilise backends that perform regular R builds inside of Docker containers. Read the [project proposal](#) for details.

The package `plumber` ([website](#), [repository](#)) allows creating web services/HTTP APIs in pure R. The maintainer provides a ready to use Docker image `trestletech/plumber` to run/host these applications with [excellent documentation](#) including topics such as multiple images under one port and load balancing.

Batch processing

The package `batchtools` ([repository](#), [JOSS paper](#)) provides a parallel implementation of Map for HPC for different schedulers, including Docker Swarm. A job can be executed on a Docker cluster with a [single R function call](#), for which a Docker CLI command

is constructed as a string and executed with `system2(..)` .

"Reproducible research for big data in practice": call for abstracts EGU GA 2017 session

09 Nov 2016 | By Daniel Nüst

We are happy to announce that a session convened by o2r team member [Edzer Pebesma](#) along with co-conveners [Yolanda Gil](#), [Kerstin Lehnert](#), [Jens Klump](#), [Martin Hammitzsch](#), and [Daniel Nüst](#) was accepted at next year's European Geosciences Union General Assembly.

The **call for abstracts** is now open. The abstract submission deadline is 11 Jan 2017, 13:00 CET. So there is plenty of time to contribute, prepare an abstract and share your experience of reproducible research.

Please [spread the word](#) and find out more at <https://bit.ly/rregu17>.

From the session description:

This session will showcase papers that focus on big data analysis and take reproducibility and openness into account. It is open to members of all programme groups and scientific disciplines to present how they conduct data-based research in a reproducible way. They are welcome to share practical advice, lessons learned, practical challenges of reproducibility, and report on the application of tools and software that support computational reproducibility.

The session is co-organized as part of the Interdisciplinary Event "Big Data in the Geosciences" (IE 3.3), and the [division on Earth & Space Science Informatics](#) (ESSI ESSI4.11). "Using computers" is the unifying feature of many a researcher in the [scientific divisions](#), so we look forward to meet a diverse group of people next year in Vienna. In the session description the conveners point out that...

[c]omputational reproducibility is especially important in the context of big data. Readers of articles must be able to trust the applied methods and computations because [...] data are also unique, observed by a single entity, or synthetic and simulated. Contributions based on small datasets are of special interest to demonstrate the variety in big data. Topics may include, but are not limited to, reproducibility reports and packages for previously published computational research, practical evaluations of reproducibility solutions for a specific research use case, best practices towards reproducibility in a specific domain such as publishing guidelines for data and code, or experiences from teaching methods for computational reproducibility.

Open in Action

24 Oct 2016 | By Marc Schutzzeichel

The Open Access movement has improved the foundation for research reproducibility in that it has greatly advanced the accessibility of research data and text. This year's theme for the [International Open Access Week](#) is "Open in Action". The o2r team joins in by creating [local awareness](#) for what may come beyond Open Access.



Image by openaccessweek.org, licensed under [CC BY 4.0 Int.](#)

To transform access into action, the o2r team is working towards the implementation of a simple technical solution. A "one click reproduce" button is one of the extremes within the continuum of reproducibility. It enables the user to recreate the original results of a study with only a mouse click. In order to realize that, a new format for the publication of research findings has to be created and integrated into the publication cycle.

In o2r we envision a container format that implements the *executable research compendium* (ERC) to encapsulate any information relevant to constituting a complete set of research data, code, text and UI. This includes any necessary specification of the working and run time environments.

Towards the other end of the continuum of reproducibility we find examples of published code and data that are openly accessible and yet fail to be rebuilt easily by another scholar. By being dependent on other software, vanished packages and specific versions or environments, such cases leave it to the user to reconstruct the individual computational dependency architectures. This strongly increases the efforts to rebuild, run, or compile the code and thus effectively blocks *Open Action*.

With the use of ERCs such obstacles can be resolved: The original analysis underlying a scientific publication becomes fully reproducible for independent researchers and anyone interested. Opening reproducibility is where we see the biggest need for Open Action in science.

Workshop on Reproducible Open Science

23 Sep 2016 | By Daniel Nüst, Markus Konkol

Just two weeks ago, o2r team members [Daniel](#) and [Markus](#) proudly presented the project's first workshop paper "*Opening the Publication Process with Executable Research Compendia*" at the [First International Workshop on Reproducible Open Science](#) held in conjunction with the [20th International Conference on Theory and Practice of Digital Libraries](#) and supported by [RDA Europe](#).

The workshop was a great event with contributions from very diverse backgrounds, ranging from computer science to library technology, and use cases, from big data to metadata interoperability or microscopic experiments.

The talks we're accompanied by excellent **keynotes** given by [Carole Goble](#) on the "R* Brouhaha" and [Sünje Dallmeier-Tiessen](#) on CERN's hard [work towards reproducible research](#).

The presentations were followed by a **general discussion session**, which touched, for example, the topics of publication bias/negative results, education having a higher potential than yet another infrastructure ("software data carpentry works", says [Carole Goble](#)), and the necessity to communicate better about reproducible research. The latter lead to the idea of "five stars of reproducibility" inspired by the tremendously useful [5 ★ Open Data](#) and also the [FAIR principles](#).

All the **slides** are [available online](#), including [our own](#).

It was great for us to share our ideas of an *Executable Research Compendium* with the workshop attendees. The discussions and feedback was very helpful. We especially realized that we need to sharpen the distinctive aspects of our project when we talk about it. We're now working hard to implement this in a paper draft we're working on.

We thank the [organizers Amir Aryani, Oscar Corcho, Paolo Manghi, and Jochen Schirrwagen](#) for the well-run event! Hopefully there is going to be a second edition next year.

Docker presentation at FOSS4G conference

06 Sep 2016 | By Daniel Nüst

Update: A video recording of the presentation is now published on the TIB AV-Portal <http://dx.doi.org/10.5446/20330>

o2r team member [Daniel Nüst](#) recently participated in the worlds largest conference for geospatial open source software. The [FOSS4G 2016](#) was hosted by the Open Source Geospatial Foundation ([OSGeo](#)) and took place close to home, namely in Bonn. Therefore Daniel was extremely happy that his [talk](#) “An overview of Docker images for geospatial applications” was voted to be presented by the OSGeo community. Daniel presented an evaluation into the existing containers for FOSS4G software. After an introduction into Docker and some live demos, the takeaway was that everybody should use Docker more, and many different application scenarios (development, demos, training, cloud deployment) exist.

The presentation was very well attended (~ 120 people), albeit taking place in the first session on Friday morning after the conference dinner the night before. [Reactions on Twitter](#) were also quite positive, several [good questions](#) were asked, and great discussions followed throughout the day.

Much interest in Docker containerization, [#foss4g pic.twitter.com/i55KphJwKv](#)

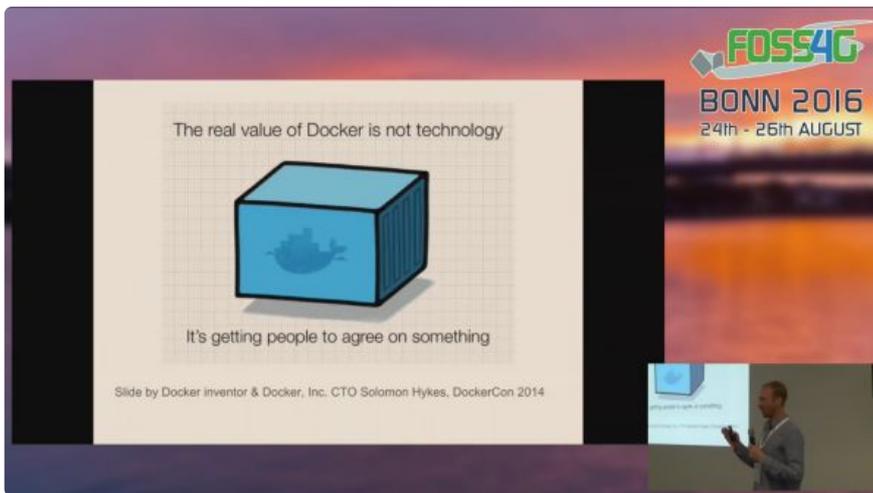
— michael GOULD (@0mgould) [August 26, 2016](#)

The main part of the work is published in the OSGeo wiki: a comprehensive list of Docker containers published by projects or third parties to use a large variety of tools, libraries, or Desktop applications in Docker containers. Check out the list at <https://wiki.osgeo.org/wiki/DockerImages>. Contributions are welcome!

How is this related to the o2r project? The expertise build up around Docker should be shared with the communities we know. And more concretely, many applications in the geospatial world are build upon services and APIs, so scientific work building upon these APIs will require to archive such services, too. This is a topic we will experiment on in the second year of o2r.

As some popular projects surprisingly did not have Docker images yet, Daniel started a new independent project [on GitHub](#) to provide a place for FOSS4G-related containers and to expand the knowledge and application of containers for geospatial applications: [geocontainers](#). Inspired by [Biodocker](#), geocontainers is intended to be a place to experiment and collaborate on containers without any initial rules or guidelines.

All of this is described in detail in [his presentation](#), which is also available as a [video recording](#). Feedback welcome!



The conference was excellently organized in a [great venue](#) which includes the former Plenary Chambers of the Bundestag. Indeed a very special place to meet the people behind the projects of Free and Open Source Software for Geospatial.



Summer break technical post: ORCID OAuth with passport.js

12 Aug 2016 | By Daniel Nüst, Jan Koppe

With the University in a rather calm state during summer, the o2r team continues to work on the first prototypes for testing and demonstrating our ideas. This is the first post on a technical topic, and we will occasionally write about topics that are not related to the scientific work but either kept us busy for some time or might be useful to others.

Last week o2r team member Jan struggled with the implementation of the **login feature** for a **Node.js microservice**. *Why would we bother with that?* Because we want to share our prototypes publicly and invite you to try them out, but at the same time not have to worry about one of your most valuable possessions: your password.

Therefore we decided early on to rely on **three legged OAuth 2.0** for handling user authentication. We opted for **ORCID** as the authorization server because it is the most widespread identification for researchers today¹, and because of the potential for useful integrations in the future².

The solution³ required to dig a bit deeper into the code of the used libraries, namely **passport.js** with the plugin **passport-oauth4**. Jan summarizes everything nicely in **this Gist** and the working implementation is part of our component **o2r-bouncer**. The ORCID support team was even so kind to include our solution on their **code examples page** and we shared it with the **ORCID API Users mailing list** in the hope that future developers will find this information helpful.

So in the end, a full day of work to figure out two missing lines of code, but still many days saved on bullet-proofing standalone authentication and password storage.

1. The used libraries would allow us to quickly add more authorization services, such as Google or GitHub. ↩
2. Wouldn't you like to have a research container be automatically added to your publication list? ↩
3. In a nutshell, the `passReqToCallback` option must be enabled when creating the `OAuth4Strategy` and the used callback function must include 6 arguments. Only then the **function with the largest number of arguments** is used and the content of the accessToken-request answer, which includes the ORCID id and user name, is accessible in your own code. They can be found in the `params` parameter of the function, not as part of `profile` as one is used to with other OAuth servers. This seems to be a slight deviation from the standard by the ORCID folks. ↩

Feedback on and Focus for the o2r Vision

07 Jun 2016 | By Daniel Nüst

A couple of weeks ago the [o2r team](#) met with a group of experts to discuss the project's outline and scope. Being a few months into the project, the team members were eager to get feedback on their plans, which they created based on the original project proposal, the first practical evaluations, and extensive reviews of research literature. To give this feedback, we invited a group of external partners to a full day meeting at the [Wersehaus](#), a small boathouse in the countryside next to the Werse river.



Image is licensed under a [CC BY-NC-ND 4.0 International](#) license.

This workshop was already planned in the project proposal and proved to be worth the preparation and, first and foremost, the efforts of our guests to travel to Münster. The external participants were [Xenia van Edig](#) from Copernicus's business development team, [Hylke Koers](#), Head of Content Innovation at Elsevier, [Simon Scheider](#) from the Department of Humany Geography and Spatial Planning at Utrecht University, [Tomi Kauppinen](#) from the Department of Computer Science at Aalto University, and [Werner Kuhn](#) from the Center for Spatial Studies at University of California, Santa Barbara. The photo above also shows the o2r team members participating: [Edzer Pebesma](#), [Daniel Nüst](#), [Markus Konkol](#), [Chris Kray](#) (all ifgi), [Holger Przibytzin](#), and [Marc Schutzeichel](#) (both ULB).

We started the day with talks by the external partners and project grantees. With such a select group, we were not surprised to get an excellent discussion rolling from the first talk on! You can download the talks' slides below if available, just click on the person's name.

- [Edzer](#) set the context of the project and took a look back at the motivation for the project (among which is personal annoyance!)
- [Werner](#) discussed the products of research (hypotheses, software, data, narratives) and provided some claims on these that fueled a lively discussion.
- [Tomi](#) approached reproducibility from the question "How science works?" and connected it to his work on Linked Open Science (see also the original [Prezi](#)).
- Holger introduced the interests and role of the university library in the project.
- [Xenia](#) presented different levels of open access publication workflows and shared experiences from the publication domain and enforcement of openness.
- [Hylke](#) talked about content innovation's relation to reproducibility and showed a variety of work around interactivity in the article of the future.
- [Chris](#) related reproducible research to ifgi's vision of an open geoinformatics platform, and critically discussed benefits and challenges.

We continued the day with intensive discussions on the project's schedule for the first year, stretching across all aspects such as preservation metadata, usability and user interaction, and compendium specification. After lunch these areas were explored more deeply in an Open Space setting prepared by the projects full-time employees Marc, Markus, and Daniel. Afterwards we drilled deeper to identify potentials risks and their mitigations, as well as answering the crucial question: Where can the project have the largest impact?

We found that a narrow focus is crucial for the project to succeed. Since we're not going to change the publishing landscape in one step and we want to make an impact in the community we know best, geoinformatics, we see these high priority goals for the foreseeable project's future:

- New means of *interaction with and exploration of scientific spatio-temporal data, analyses, and visualisations* based on

linked research compendia contents.

- *Automatic* (bordering on [magical](#)) *creation of executable research compendia* based on typical science workspaces for R-based geosciences.
- Specification of an *executable research compendium rooted firmly* in users' requirements, preservation requirements, the currently dominating procedures in scientific publications, and reality of highly diverse scientific workflows.
- *New ways for searching scientific work* based on the integrated and linked parts of a research compendium (text, code, data, user interface bindings).

So what will we understand in two years time that we do not know now?

- We have a good understanding of how far the process of creating research compendia can be automated, and what efforts remain for authors or preservationists that must be counterbalanced with incentives.
- We know the potential of user interface bindings as the connecting entity of research compendia.
- We show the improvements in discovery and understanding of research when all aspects of research are explicitly linked in a meaningful way.
- We get to know the common language as well as points of contact for the involved parties as we create a closer connection between research, preservation, and publication communities.

What do you think? Ambitious goals, or nothing new? Give the new discussion feature below this post a try!

We thank again our guests for their valuable inputs. Having their backgrounds in research as well as scientific publishing, their critical evaluation helps us to shape a clear direction for our work. To keep in touch, follow us on [Twitter](#) or [GitHub](#).

Container Strategies for Data & Software Preservation that Promote Open Science (DASPOS workshop)

20 May 2016 | By Daniel Nüst

In the last two days o2r team member [Daniel](#) participated in a workshop organized by the project “Data and Software Preservation Open Science” ([DASPOS](#)) at the University of Notre Dame, USA. It was organized in an excellent fashion and in perfect amenities by the Center for Research Computing ([CRC](#) at the University of Notre Dame).

The workshop title “*Container Strategies for Data & Software Preservation that Promote Open Science*” fits perfectly with our own project goals, so Daniel was not surprised learn about a lot of great initiatives and projects. A great group of researchers and librarians, mostly from the US, presented diverse topics almost all of which had one connection or another with o2r. The general connecting feature between all participants were (Docker) containers and the interest in preservation. Different approaches were presented in hands-on sessions, for example [ReproZip](#), [Umbrella](#) and the NDS Dashboard. You can check the [full schedule and participant list](#) for details.

A few highlights from Daniel's perspective were the shared understanding that a common language and terms would be needed going forward when containerisation is applied more widely for openness and transparency of research. But at the same time, and certainly at the current point in time of implementations, diversity is good and some amount of re-doing existing “features” is unavoidable.

You can find [Daniel's presentation](#) (also on [SlideShare](#)), as well as [all other's slides and other \(reading\) material](#), on the Open Science Framework (OSF) website. The presentations were recorded and you can watch Daniel's talk as well as [all the others](#):

Daniel would like to thank the DASPOS team for the invitation and the excellent filming and transcribing. It was a great experience to meet the leaders in the field. The workshop was an awesome opportunity to share the o2r vision and to learn about other projects, ideas and concepts at this stage of our project.

Looking back at EGU General Assembly

02 May 2016

o2r team members Edzer Pebesma and Daniel Nüst published a short blog article on [r-spatial](#) about the project in general, and more specifically about the poster presented at EGU General Assembly [a couple of weeks ago](#).

Read the blog here: <https://r-spatial.org/r/2016/04/29/o2r.html>

The EGU poster is now also [available for download on the EGU website](#). The survey is also still running - please participate [here](#)!

Join our first survey

21 Apr 2016 | By Markus Konkol

Getting user input and evaluating our ideas is a crucial part of the project. Therefore, starting today, we run an **online questionnaire** investigating user interaction in the context of reproducible research. The survey is also advertised **this week at the EGU General Assembly**.

Please take a few minutes to help understanding reproducibility in geoscience research by participating in the **first o2r survey** at <https://o2r.info/survey>.

Opening Reproducible Research at EGU General Assembly 2016

08 Apr 2016 | By Daniel Nüst

Next week the largest European geosciences conference of the year will take place in Vienna: the [European Geophysical Union General Assembly 2016](#). It takes place in the Austria Center Vienna for a full week and expects to welcome over [thirteen thousand scientists](#) from all over the world. A vast variety of research across all disciplines of the Earth, planetary and space sciences will be presented in a [meeting programme](#) featuring workshops, lectures, talks, and posters.



One of the participants will be o2r team member Edzer Pebesma ([@edzerpebesma](#) and <http://r-spatial.org>).

Edzer presents our abstract "[Opening Reproducible Research](#)" in the poster session [ESSI3.4 Open Access to Research Data and Public Sector Information towards Open Science](#). The session takes place on *Thursday, April 21st, from 17:30 to 19:00 in Hall A*. Make sure to drop by and get a glance at our first plans and the many other [talks](#) and [poster presentations](#) in this session.

We look forward to the discussions about reproducible research and to get feedback about the project.

Introducing o2r

19 Jan 2016 | By Daniel Nüst, Markus Konkol

Welcome to the new website of the research project *Opening Reproducible Research*.

You can learn the basics of the project and get to know the participants on the [About](#) page.

In short, we will develop new methods to make geosciences research reproducible. We will create open source tools and standards to compile text, data, and code (both sources and binary executables) into research compendia. These compendia will be easy to create for non-developers, executable in a web-based infrastructure, and allow exchanging of data and methods between compatible compendia.

You can follow our work on [GitHub](#).

Website pages

About

Opening Reproducible Research (o2r) is a project by the Institute for Geoinformatics (ifgi) and University and Regional Library (ULB) at the University of Münster, Germany.



Goal

We propose a system that supports the evolution of scientific publications from static papers into dynamic, executable research documents and aim for the main aspects of open access: improving the exchange of, facilitating productive access to, and simplifying reuse of research results that are published over the internet.

Building on existing open standards and software, o2r develops standards and tools for executable research documents (the so-called *Executable Research Compendia* or *ERCs*) and evaluates them, initially focusing on the Geosciences domain. Building on recent advances in mainstream IT, o2r envisions a new architecture for storing, executing and interacting with the original analysis environment alongside the corresponding research data and manuscript. *o2r bridges the gaps between long-term archiving, practical geoscientific research, and publication media.* The o2r team collaborates with publishers to achieve the following goals:

- Identify key barriers to working reproducibly
- Design and evaluate ways to overcome these barriers
- Develop approach to reap the benefits of reproducible research
- Implement platform that realises approach and test it

Learn about the accomplishment of these goals on the [Deployment page](#) and stay updated [via Twitter](#).

Why Open Access?

Open access is not only a form of publishing such that research papers become available to the large public free of charge, it also refers to a trend in science that the act of doing research becomes more open and transparent when it comes to data and methods. Increasingly, scientific results are generated by numerical manipulation of data that were already collected, and may involve simulation experiments that are entirely carried out computationally. Reproducibility of research findings, the ability to repeat experimental procedures and confirm previously found results, is at the heart of the scientific method.

Why Open Source?

For Open Science and reproducible research, we see no alternative to Open Source software. All scripts, code, and libraries supporting a computational analysis must be open for scrutiny by fellow scientists. We publish current code online, instead of holding back until publication of a paper, to profit from interaction with the Free and Open Source Software (FOSS) community. The software of supported workflows (i.e. R) and underlying technologies (i.e. Docker) are also published under FOSS principles. Already in the project proposal, we set a clear agenda on the question of software licenses:

All software developed by project staff will be distributed under a permissive open source license that allows reuse, modification and integration in commercial systems (e.g. Apache 2.0). Development happens openly at GitHub and all developments are visible directly instead of after the end of the project.

See our [Deployment](#) page for more information about all software projects.

The team

- Dr. Stephanie Klötgen (ULB)
- Eftychia Koukouraki (ifgi)
- Prof. Dr. Christian Kray (ifgi)
- Jörg Lorenz (ULB)
- Dr. Daniel Nüst (ifgi)
- Prof. Dr. Edzer Pebesma (ifgi)
- Holger Przybytzin (ULB)
- Dr. Beate Tröger (ULB)

Contact: o2r.team@uni-muenster.de



Former members

- Fabian Fermazin (student assistant, 2020-10 to 2022-03)
- Nick Jakuschona (student assistant, 2019-04 to 2022-03)
- Tom Niers (student assistant, 2019-04 to 2022-03)
- Dr. Markus Konkol (ifgi, 2016-2020)
- Rehan Chaudhary (ifgi, intern, 2017-01 to 2017-07)
- Juan Sebastian Garzon (student assistant, 2020-02 to 2021-12)
- Philipp Glahe (student assistant, 2019-04 to 2019-09)
- Laura Goulier (student assistant, 2019-06 to 2020-05)
- Matthias Hinz (ifgi, research assistant, 2016-12 to 2017-03)
- Jan Koppe (ifgi, student assistant, 2016-03 to 2016-08)
- Torben Kraft (ifgi, student assistant, 2017-01 to 2017-12)
- Timm Kühnel (ifgi, student assistant, 2017-01 to 2018-06)
- Lukas Lohoff (ULB, student assistant, 2016-12 to 2018-03)
- Yousef Qamaz (ifgi, student assistant, 2019-06 to 2020-03)
- Dr. Marc Schutzeichel (ULB, research associate, 2016-02 to 2018-01)
- Jan Suleiman (ifgi, student assistant, 2016-04 to 2017-12)

External partners

The o2r project is connected to external partners since its inception, and the group has been extended since then. They come from different disciplines and provide valuable feedback on project plans and decisions.

Dr. Xenia van Edig (Business Development, Copernicus.org)



Dr. Tomi Kauppinen (Department of Computer Science, Aalto University School of Science, Finland)



Prof. Dr. Werner Kuhn (Center for Spatial Studies, [spatial@ucsb](http://spatial@ucsb.edu) University of California Santa Barbara, Santa Barbara, CA)



Dr. Albert Geukes, [CeDIS](http://CeDIS.org), FU Berlin, Germany & OJS-de.net



Dr. Simon Scheider, Department of Human Geography and Spatial Planning, Universiteit Utrecht, The Netherlands

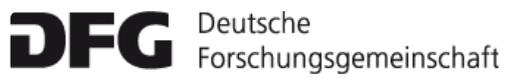
Dr. Wolfgang Peters-Kottig, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany

Laura Hassink, Senior Vice President Publishing Transformation at RELX (previously Maarten Cleeren - Director of Product Management, Enriched Content at Elsevier; Dr. Hylke Koers - Head of Content Innovation, Elsevier)



Funding

This project *Opening Reproducible Research* (see also [Offene Reproduzierbare Forschung](#) and [Offene Reproduzierbare Forschung II @ DFG GEPRIS](#)) receives funding by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under project numbers PE 1632/10-1, KR 3930/3-1 and TR 864/6-1 from 2016/01 to 2018/06, and under project numbers PE 1632/17-1, KR 3930/8-1, and TR 864/12-1 from 2019/04 to 2021.



Documents

All components of o2r are openly developed [on GitHub]({{ site.github.org }}). The project's findings manifest themselves in the following core specifications and documents, all of which are under development. - **[ERC specification](https://o2r.info/erc-spec)** ([source](https://github.com/o2r-project/erc-spec)) formally defines the Executable Research Compendium and provides some background. - **[Architecture](https://o2r.info/architecture)** ([source](https://github.com/o2r-project/architecture)) describes multiple levels of architecture, from the relation of our reproducibility service with other platforms down to internal microservices. - **[Web API](https://o2r.info/api)** ([source](https://github.com/o2r-project/api)) defines a RESTful API for our reproducibility service, also used by our platform client. To cite the specifications and documentations please use > Nüst, Daniel, 2018. Reproducibility Service for Executable Research Compendia: Technical Specifications and Reference Implementation. Zenodo. doi:[10.5281/zenodo.2203844](http://doi.org/10.5281/zenodo.2203844)

Pilots

The primary objective of the second phase of the project [Opening Reproducible Research](https://o2r.info) (o2r) is to use Executable Research Compendia (ERCs) for actual scientific publications. The o2r team is implementing pilots for several work packages: - In [collaboration pilots](#collaboration-pilots), we work with publishers on Virtual Special Issues (VSIs). In these VSIs, submitted articles will have [ERCs](/results) as supplemental material - an evolutionary use of ERCs. - In a [self-hosted pilot](#self-hosted-pilot), we extend [Open Journal Systems](https://pkp.sfu.ca/ojs/) (OJS) to make the ERC itself the item under review in a demonstration journal. The pilots are supported by operating infrastructure and we will support authors, reviewers, and readers to enhance their scholarly communication with ERCs. The three pilots cover current publication practices from large scale publishers to independent journals. The pilots are accompanied user studies and monitoring, in order to gain insights about the costs and benefits of ERC-based research publications. All pilots require the integration of o2r services and tools in existing, established IT systems and organisational workflows - we thank the collaborating publishers and editors as well as all authors and editors for their participation and open-mindedness!_ ----- ## Collaboration pilots The diversity of the VSIs, potentially covering journals from a variety of geoscience disciplines, allows to reach the broad authorship and readership of the involved journals. It evaluates the creation and inspection process for ERCs outside of a prototypical lab setting, so the technical infrastructure for ERCs is completely overhauled to support a growing number of users. ### [Information for authors](https://docs.google.com/document/d/1skV3niWpQDYrtLWHob3UbP-Ejgbx1sG6opLcJ1WjZng/edit?usp=sharing) ### call for almost reproducible papers](/almost) ### [Information for reviewers](https://docs.google.com/document/d/1cV62UWCohHstclDisrtNYZmjr2E1YuHxMw7O6dk/edit?usp=sharing) In case of questions, you can always [contact us](#get-help). ### Copernicus Publications Virtual Special Issue**Status:** First review started with loose integration or ERC (communication of ERC URL via coverletter and handling editor). We are currently collecting more showcases. **Participating journals** and papers: - [**Earth System Science Data**](https://www.earth-syst-sci-data.net/) (ESSD) - [Deep-sea sediments of the global ocean_ by Markus Diesing](https://doi.org/10.5194/essd-2020-22) (see RC1 for ERC information) **Next steps:** - We are looking for more Copernicus journals to participate in the virtual special issue!_ See the [Open Call for Participation in Virtual Special Issue for Reproducible Research](https://drive.google.com/file/d/1WltyoZQz7UpfTqU396JdZ3zVe0gZHv1/view?usp=sharing). - Integration of ERC form_ field to submission forms and adding of extra reviewer questions to the review form. - Direct display of ERCs from the article reading page. ### Elsevier Virtual Special Issue**Status:** Looking for collaborating journals See the [informative leaflet for a virtual special issue for Elsevier journals based on containers and R](https://drive.google.com/file/d/1uIm6hA_Ke7bWmgiDYdaG2gsZAVwsArwB/view?usp=sharing). ----- ## Self-hosted pilot The self-hosted pilot replaces the traditional paper with ERCs. It demonstrates ERCs' potential to stakeholders and provides a platform for evaluation of ERCs in education. The self-hosted pilot is also used to investigate the impact of ERCs on understanding, complementing or even replacing traditional reading material in Geoinformatics courses. The students also have the opportunity of gaining first experiences as academic authors by submitting they final project report as an ERC. _Example usage of the self-hosted pilot based on OJS is described in this [blog post](https://o2r.info/2019/10/15/Opening-Reproducible-Research-with-OJS/)._ ##### We are open for collaborations with OJS journals _We are always looking for feedback and are open collaboration with OJS developers and OJS journal maintainers who are interested in enhanced reviews and publications powered by ERC on OJS. _ ### Get help If you have any questions please do not hesitate to contact us: - Ask a question on [Stackoverflow](https://stackoverflow.com) using the tag [erc](https://stackoverflow.com/questions/tagged/erc) or [executable-research-compendium](https://stackoverflow.com/questions/tagged/executable-research-compendium). - an email to o2r.support@uni-muenster.de if you have specific questions you prefer not to share publicly. - Join our [public pilots chat room](https://gitter.im/o2r-project/pilots) on Gitter: [Gitter]({{ 'public/pilots.svg' | absolute_url }})(https://gitter.im/o2r-project/pilots) </div>

⚙️ Deployment

Find a list of all **papers and presentations** at <https://o2r.info/publications/>. Try the **online demo** at <https://o2r.uni-muenster.de> (<https://o2r.uni-muenster.de>) and if you are a developer find the web API endpoint at <https://o2r.uni-muenster.de/api/v1/> (<https://o2r.uni-muenster.de/api/v1/>). Try it out on your own machine with the **reference implementation** ([/2017/10/31/reference-implementation/](https://2017/10/31/reference-implementation/)) (only Docker required!): `git clone https://github.com/o2r-project/reference-implementation` `docker-compose up`` Watch a short **video** of our platform prototype (turn on subtitles!):

To cite the reference implementation please use > Nüst, Daniel, 2018. Reproducibility Service for Executable Research Compendia: Technical Specifications and Reference Implementation. Zenodo. doi:[10.5281/zenodo.2203844](https://doi.org/10.5281/zenodo.2203844) (<http://doi.org/10.5281/zenodo.2203844>) ## Software Learn more about our projects on [\[Open Hub\]](https://www.openhub.net/orgs/o2r) (<https://www.openhub.net/orgs/o2r>) and [\[GitHub\]](https://github.com/o2r-project) (<https://github.com/o2r-project>), where we currently have [\[NA\]](#) repositories with [\[NA\]](#) forks using [\[NA\]](#) languages.

License



Except where otherwise noted site content created by the o2r project is licensed under a Creative Commons Attribution 4.0 International License.