Opening Reproducible Research: research project website and blog

Blog post authors:

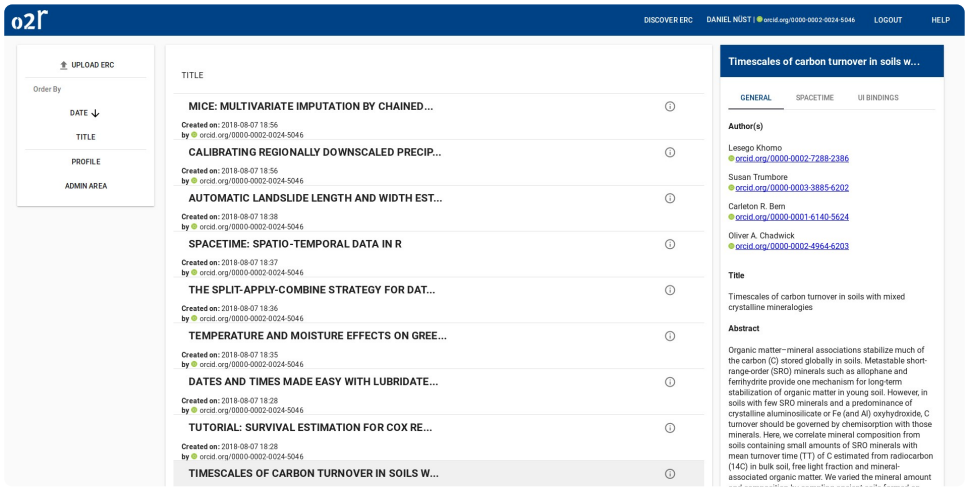
- Daniel Nüst
- Lukas Lohoff
- Marc Schutzeichel
- Markus Konkol
- Matthias Hinz
- Rémi Rampin
- Vicky Steeves

Blog posts

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.

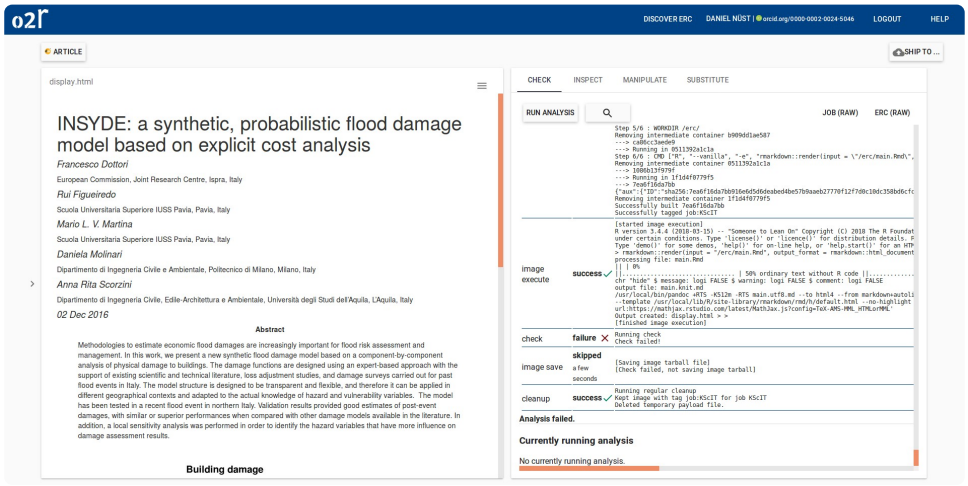


Screenshot 1: o2r reference implementation *listing of published Executable Research Compendia*. The right-hand side shows a metadata summary including original authors.

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 !

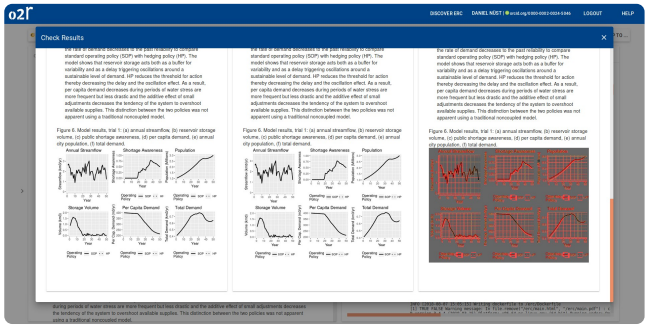


Screenshot 2: o2r reference implementation *ERC detail page* for compendium [SLVlQ](https://o2r.uni-muenster.de/#!/erc/5LVlQ). The link "Article" in the top left corner leads to the original article, the "magnifying glass" button takes you to a core feature: the reproduction result.

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 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://uni-muenster.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 memberDaniel was published in the journalPeerJ:

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**

The article is an outcome of a collaboration around the AGILE conference, seehttps://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 advise 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 continous 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 IE4.4, **"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 de-RSE's objectives). A more detailed report from the townhall will follow in due course on the de-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/Y0lDsAtaae
>
> — 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

> 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

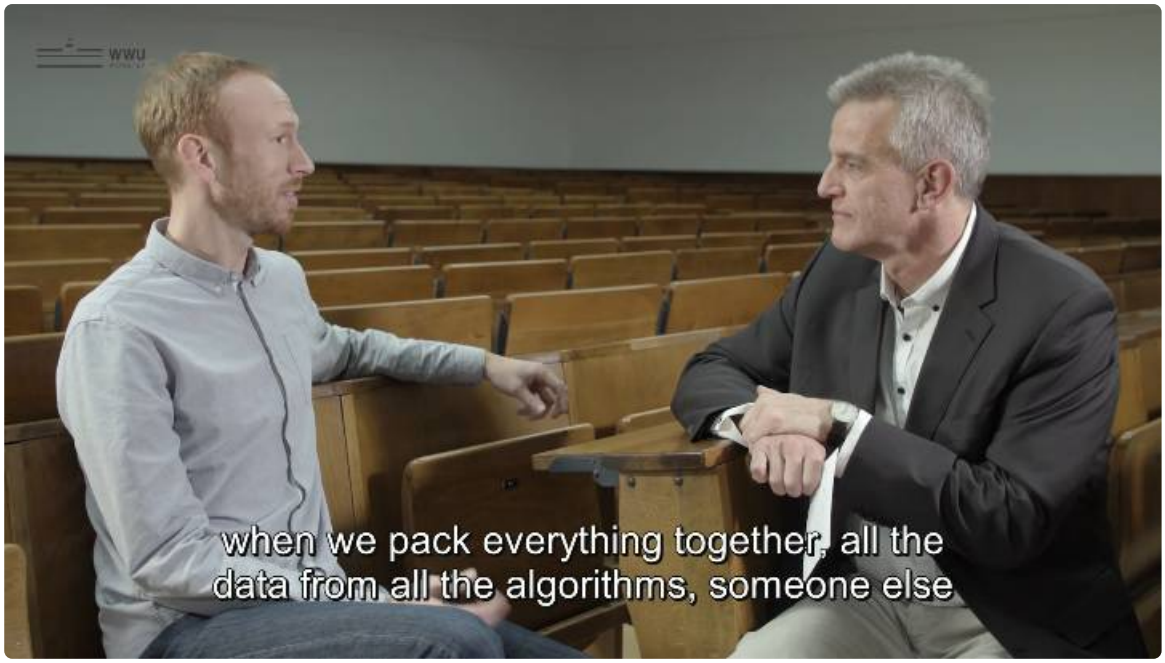
o2r Team member Daniel was invited by the university's press office to participate in a series of interviews and articles on digitisation or "digitalisation" at the WWU Münster:

The video is now available online in German (embedded below) and with English subtitles.

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 excercise by the way.
>
> Although in this case, a non-expert could very well be an experienced scientists! 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 inpolitics 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

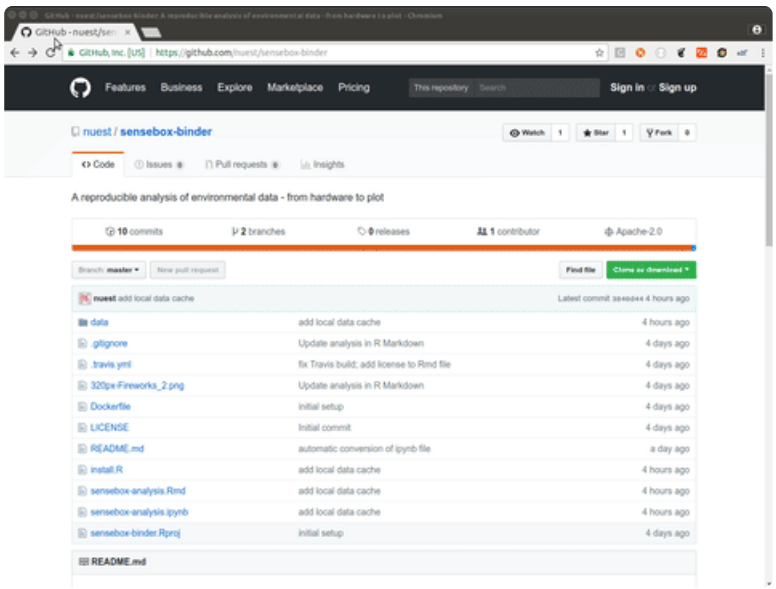
> workshops et cetera to try it out. You got to take every chance you can get to reach out to non-scientists (research resolution: Make sure all papers have non-specialist summaries). I applaud all bloggers and podcasters out there who do!

Open environmental data analysis

12 Jan 2018 | By Daniel Nüst

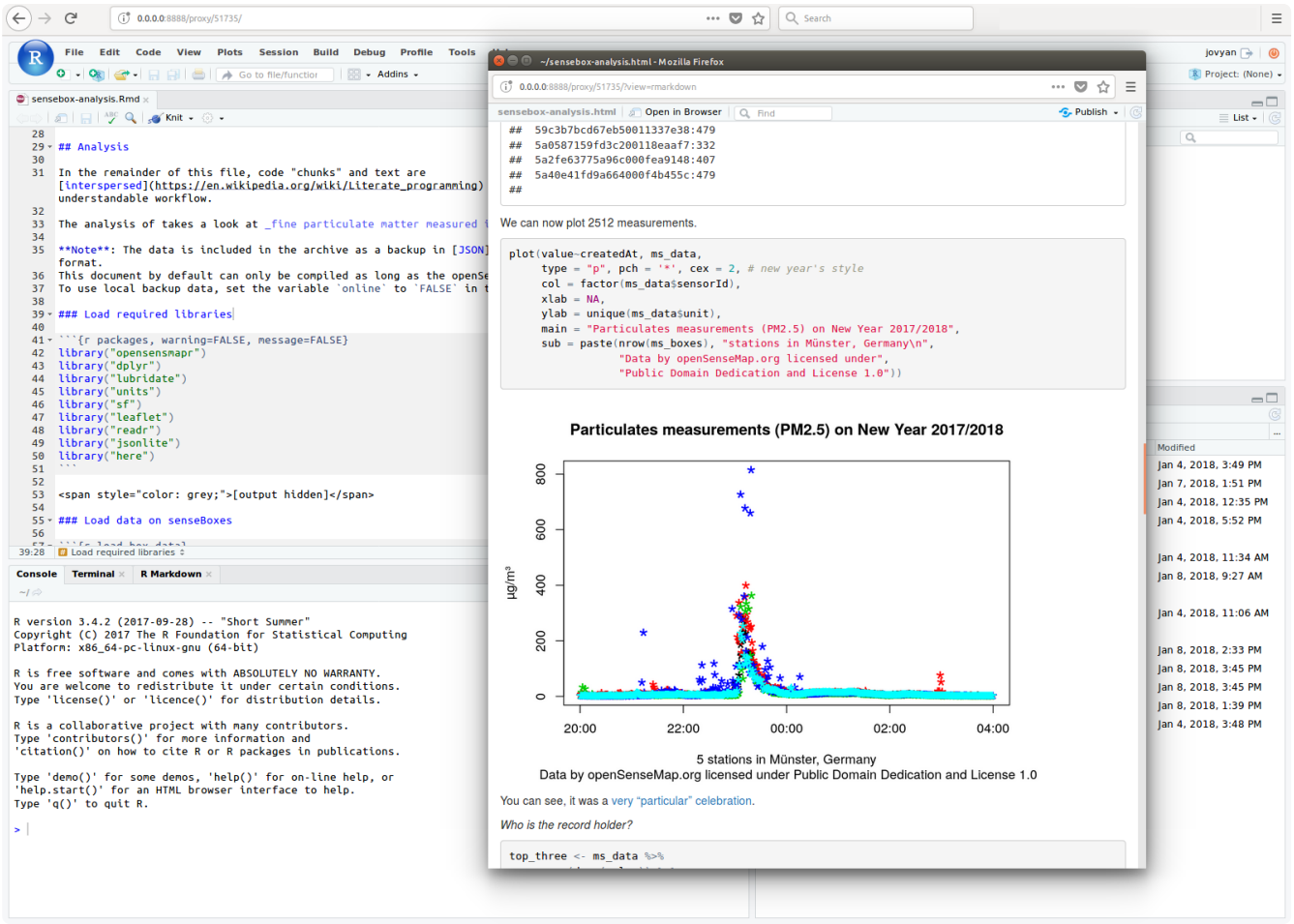
This article is cross-posted in German on the senseBox blog.

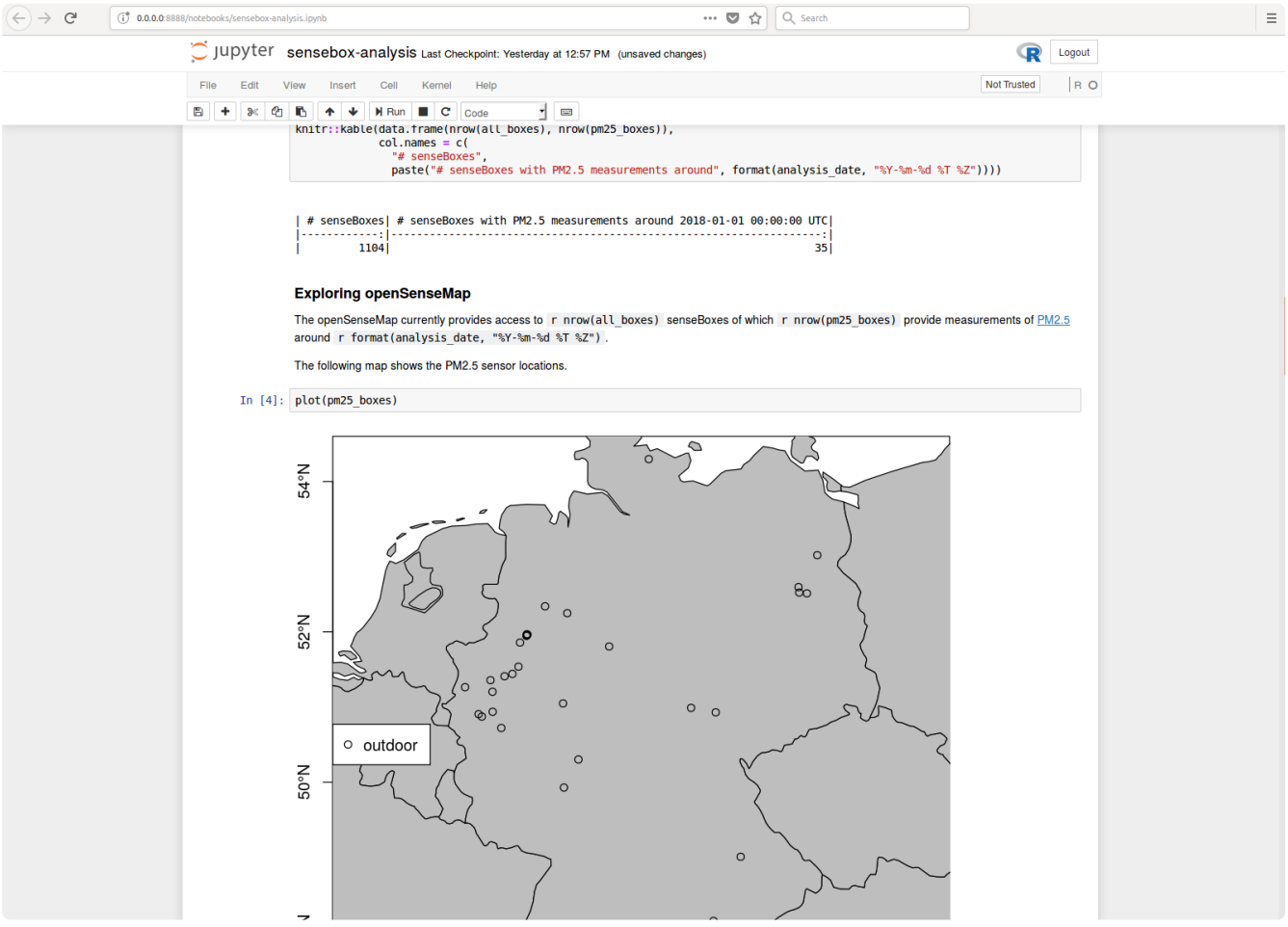
It's the time of the year to make resolutions and to see beyond one's own nose. For o2r team member Daniel, this meant to explore what he could do with his brand new *senseBox:home* and the awesome *BinderHub* instead of putting it on the back burner.



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.


sensebox-analysis.Rmd
Analysis
In the remainder of this file, code "chunks" and text are [interspersed](https://en.wikipedia.org/wiki/Literate_programming) understandable workflow.
The analysis of takes a look at _fine particulate matter measured
Note: The data is included in the archive as a backup in [JSON] format.
This document by default can only be compiled as long as the openSe
To use local backup data, set the variable `online` to `FALSE` in t
Load required libraries
```{r packages, warning=FALSE, message=FALSE}
library("opensensmapr")
library("dplyr")
library("lubridate")
library("units")
library("sf")
library("leaflet")
library("readr")
library("jsonlite")
library("here")
```
[output hidden]
Load data on senseBoxes
Load required libraries
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.
~/sensebox-analysis.html - Mozilla Firefox
sensebox-analysis.html
Open in Browser
Publish
59c3b7bcd67eb50011337e38:479
5a0587159fd3c200118eaaf7:332
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
0
200
400
600
800
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


jupyter sensebox-analysis Last Checkpoint: Yesterday at 12:57 PM (unsaved changes)
Logout
File Edit View Insert Cell Kernel Help
Not Trusted
Run
Code
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 RJohannes 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/ngn9fqvhfkp3haui**

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 anothercontinuation, a pre-conference workshop at the21st AGILE International Conference on Geographic Information Science in Lund, Sweden:**"Reproducible Research Publications"**

> The half day workshop attempts to provides 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 announcedon the conference website soon. Please also check theworkshop 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!

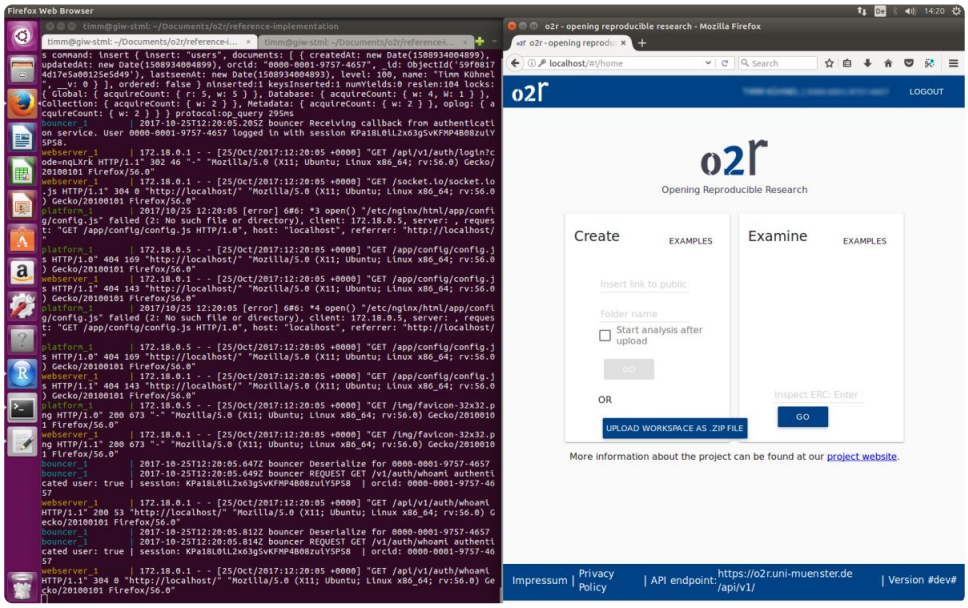
31 Oct 2017 | By Daniel Nüst

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.

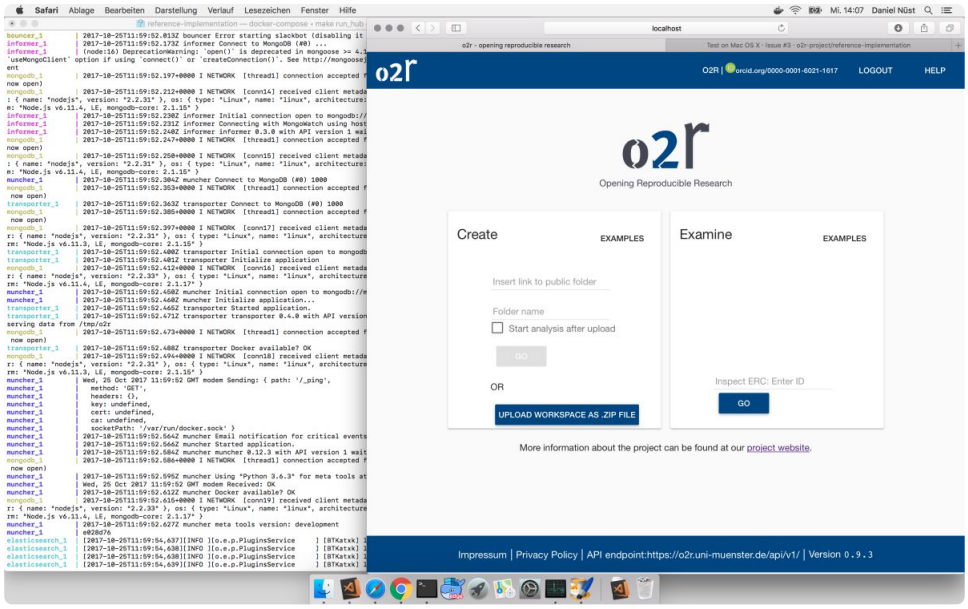


Screenshot: o2r reference implementation on *Ubuntu*.

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.

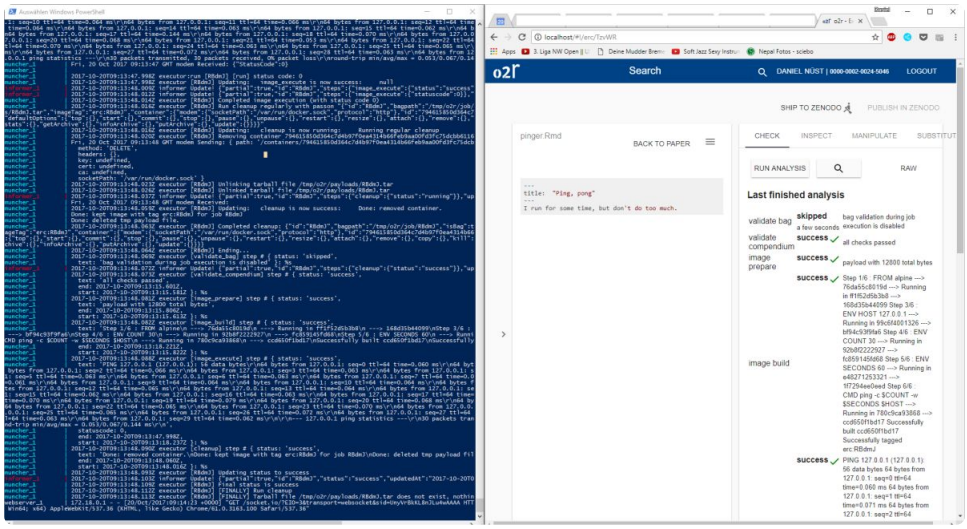


Screenshot: o2r reference implementation on *OS X*.

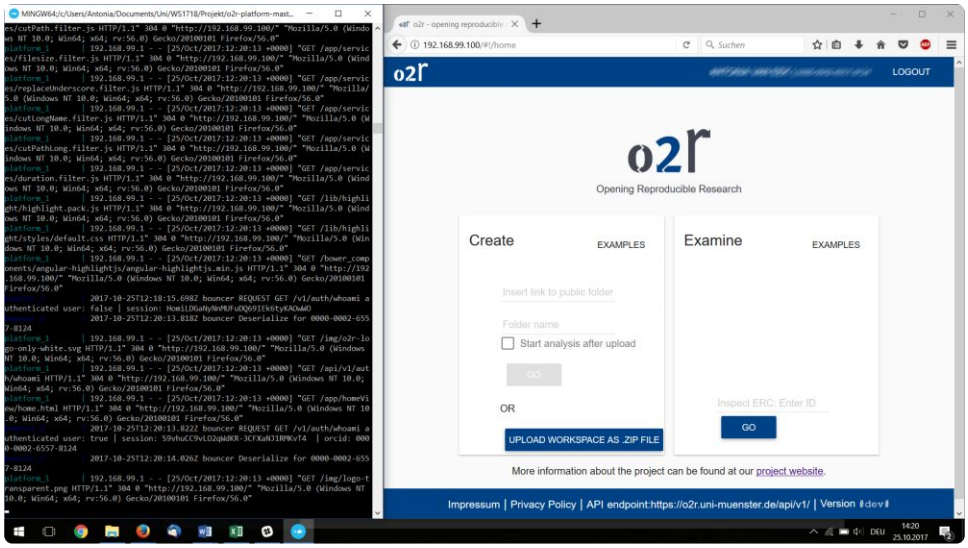
We look forward to your feedback!

`tl;dr`

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

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.

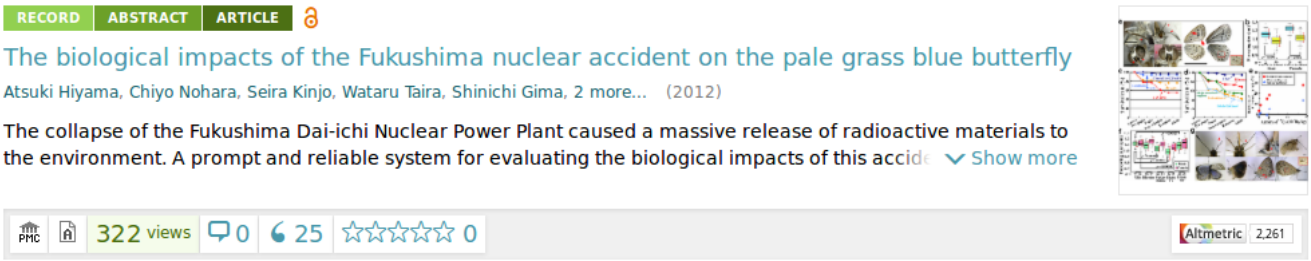


Figure 1: *ScienceOpen* badges in a search result listing.

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]).


Biostatistics (2009), **10**, 3, *pp.* 409–423
doi:10.1093/biostatistics/kxp010
Advance Access publication on April 17, 2009

R

Air pollution and health in Scotland: a multicity study

DUNCAN LEE*, CLAIRE FERGUSON
Department of Statistics, University of Glasgow, Glasgow, G12 8QQ UK
duncan@stats.gla.ac.uk
RICHARD MITCHELL
Public Health and Health Policy, University of Glasgow, Glasgow, G12 8QQ UK

SUMMARY

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. The 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).

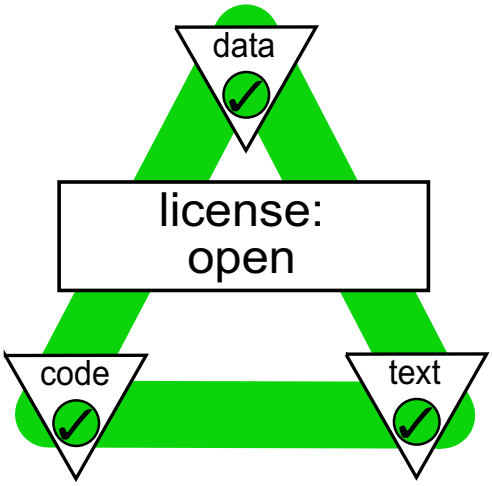


Figure 5: An extended licence badge reporting open data, text and code.

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`. 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 if 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:

executable yes

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 releasetime 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.

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 a userscript, 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 respetive 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).

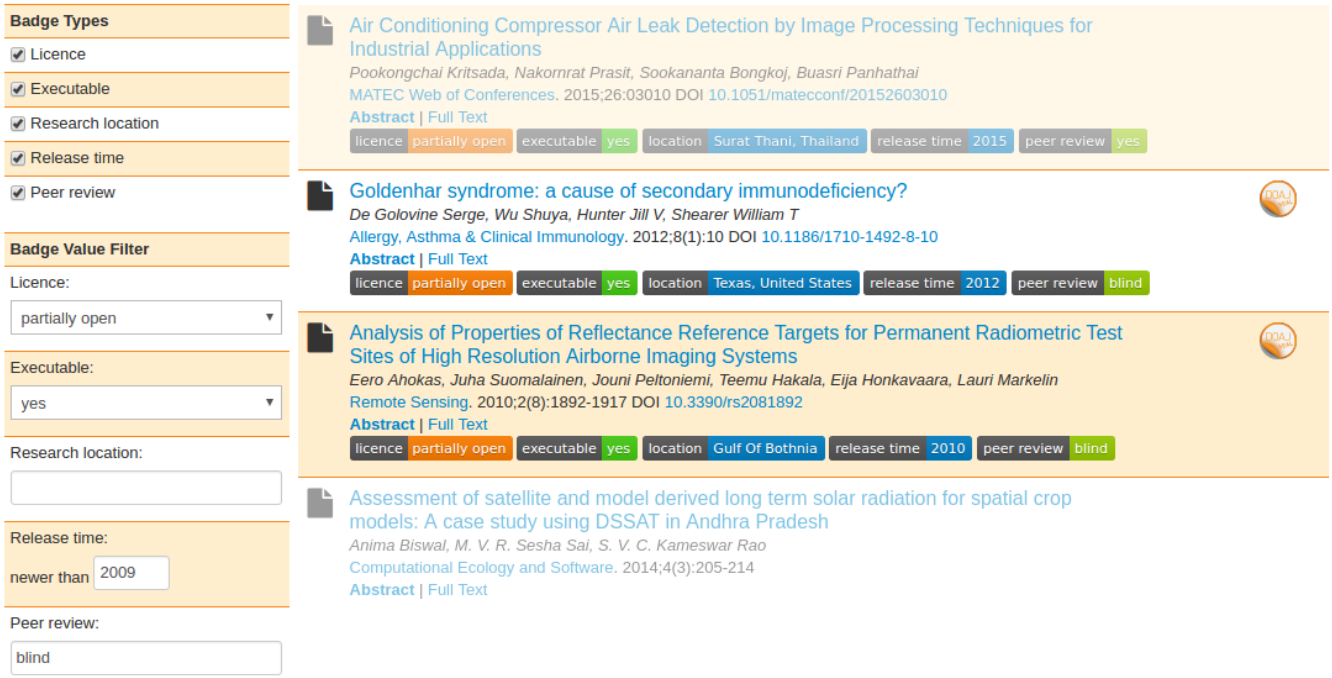


Figure 8: Filtering search results on *DOAJ*. Results not matching the filter or articles where the DOI could not be detected are greyed out.

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).

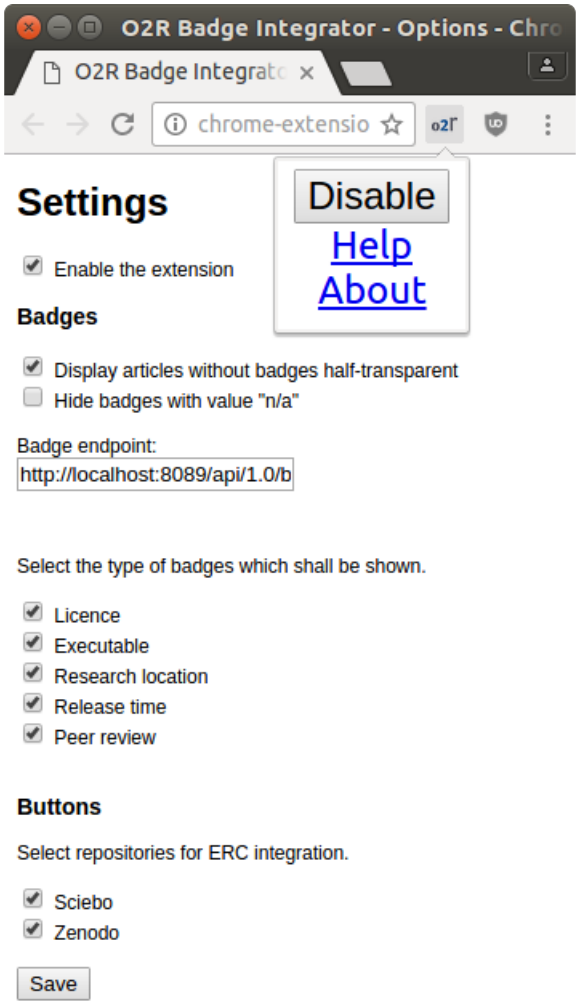


Figure 9: extender configuration.

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.

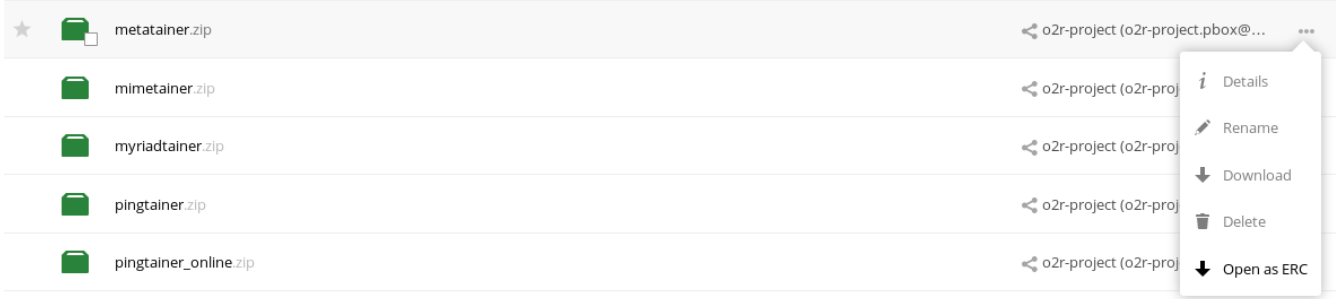


Figure 10: *Sciebo* upload integration.

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):

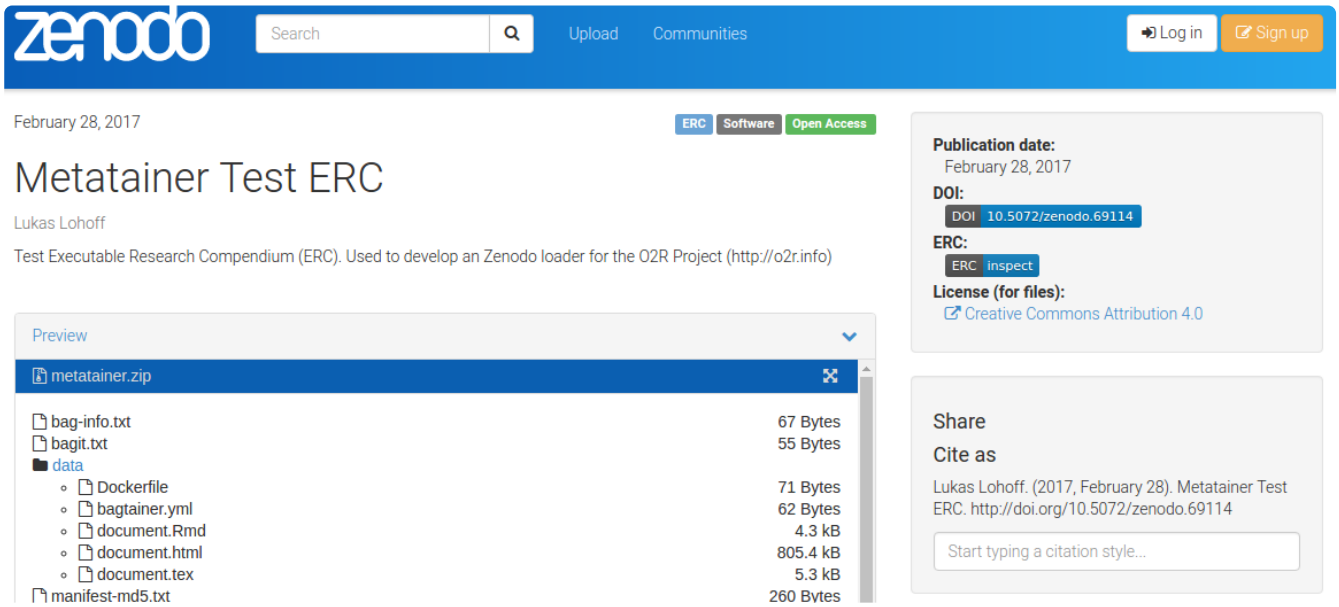


Figure 11: Link to inspection view and tag "ERC" on *Zenodo*.

Discussion

The study project Badges for computational geoscience containers initially implemented eight microservices responsible for six different badges 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 toWebExtensions and moving away from their old “add-ons” completely with Firefox 57, a port from a Chrome Extension to the open *WebExtensions* 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 ofo2r-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.

[3] Peng, Roger D. 2009. Reproducible research and Biostatistics. Biostatistics, Volume 10, Issue 3, Pages 405–408. doi:10.1093/biostatistics/kxp014.

[4] Peng, Roger D. 2011. Reproducible Research in Computational Science. Science 334 (6060): 1226–27. doi: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.

[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.

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 Brüssels, 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 wide-spread 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/IhYqmjddPD
>
> — 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/tEMG34drgp
>
> — 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

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. Reproduciblility 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 refered 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, k
nitr, Rcpp, intervals, lattice, FNN, spacetime, zoo, digest, rprojroot, futile.options, backports, mag
rittr, evaluate, stringi, futile.logger, xts, rmarkdown, lambda.r, stringr, yaml, htmltools' from sysr
eq online DB
## INFO [2017-05-30 14:49:21] Adding CRAN packages: sp, gstat, knitr, Rcpp, intervals, lattice, FNN, s
pacetime, zoo, digest, rprojroot, futile.options, backports, magrittr, evaluate, stringi, futile.logge
r, 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", "l
attice", "FNN", "spacetime", "zoo", "digest", "rprojroot", "futile.options", "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 seperate 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 arguemen `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 zxf 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 occurence of an R script, or otherwise the first occurence 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 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 `dockefile()` 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 zxf 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", "un
its", "lattice", "rjson", "FNN", "udunits2", "stringr", "xts", "DBI", "lambda.r", "futile.logger", "ht
mltools", "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 zxf 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", "ma
grittr", "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 dublicate 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 seperate 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_ImageName", "org.label-schema.name"="ImageName",
                "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 = "ImageName", 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"
WORKDIR /payload/
LABEL key1="this" key2="that" otherKey="content"
LABEL my.label.ns.name="A name" my.label.ns.description="A description"
LABEL info.o2r.name="myProject_ImageName" \
    org.label-schema.name="ImageName" \
    yet.another_labelname="true"
LABEL org.label-schema.schema-version="1.0.0-rc.1" \
    org.label-schema.build-date="2017-05-30T14:49:39+0200" \
    org.label-schema.name="ImageName" \
    org.label-schema.description="Description of the image"
LABEL R.session-info="R version 3.4.0 (2017-04-21)\nPlatform: x86_64-pc-linux-gnu (64-bit)\nRunning un
der: Ubuntu 16.04.2 LTS\n\nMatrix products: default\nBLAS: /usr/lib/libblas/libblas.so.3.6.0\nLAPACK:
/usr/lib/lapack/liblapack.so.3.6.0\n\nlocale:\n [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
\n [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_US.UTF-8    \n [5] LC_MONETARY=en_GB.UTF-8    LC_MESSA
GES=en_US.UTF-8   \n [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C                  \n [9] LC_ADDRESS=C
LC_TELEPHONE=C             \n[11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C       \n\nattached bas
e packages:\n[1] stats     graphics  grDevices utils     datasets  methods   base     \n\nloaded via a
namespace (and not attached):\n[1] compiler_3.4.0"
LABEL R.session-info="{\"platform\":{\"version\":\"R version 3.4.0 (2017-04-21)\",\"system\":\"x86_64,
linux-gnu\",\"ui\":\"X11\",\"language\":\"en\",\"collate\":\"en_US.UTF-8\",\"tz\":\"Europe/Berlin\",\"
date\":\"2017-05-30\"},\"packages\":{\"package\":[\"backports\",\"base\",\"compiler\",\"containerit\",
\"datasets\",\"DBI\",\"devtools\",\"digest\",\"evaluate\",\"FNN\",\"fortunes\",\"futile.logger\",\"fut
ile.options\",\"graphics\",\"grDevices\",\"grid\",\"gstat\",\"htmltools\",\"intervals\",\"knitr\",\"la
mbda.r\",\"lattice\",\"magrittr\",\"memoise\",\"methods\",\"Rcpp\",\"rjson\",\"rmarkdown\",\"rprojroot
\",\"sf\",\"sp\",\"spacetime\",\"stats\",\"stringi\",\"stringr\",\"tools\",\"udunits2\",\"units\",\"ut
ils\",\"withr\",\"xts\",\"yaml\",\"zoo\"],\"*\":[\"\",\"*\",\"\",\"*\",\"*\",\"\",\"\",\"\",\"\",\"\",\"\",\"\",
\"*\",\"\",\"\",\"*\",\"*\",\"\",\"*\",\"\",\"\",\"*\",\"\",\"\",\"\",\"\",\"\",\"*\",\"\",\"\",\"\",\"\",\"\",\
"\",\"*\",\"\",\"*\",\"\",\"\",\"\",\"\",\"\",\"\",\"*\",\"\",\"\",\"\",\"\",\"\"],\"version\":[\"1.0.5\",\"3.4.
0\",\"3.4.0\",\"0.2.0\",\"3.4.0\",\"0.6-1\",\"1.13.1\",\"0.6.12\",\"0.10\",\"1.1\",\"1.5-4\",\"1.4.3\"
,\"1.0.0\",\"3.4.0\",\"3.4.0\",\"3.4.0\",\"1.1-5\",\"0.3.6\",\"0.15.1\",\"1.16\",\"1.1.9\",\"0.20-35\"
,\"1.5\",\"1.1.0\",\"3.4.0\",\"0.12.11\",\"0.2.15\",\"1.5\",\"1.2\",\"0.4-3\",\"1.2-4\",\"1.2-0\",\"3.
4.0\",\"1.1.5\",\"1.2.0\",\"3.4.0\",\"0.13\",\"0.4-4\",\"3.4.0\",\"1.0.2\",\"0.9-7\",\"2.1.14\",\"1.8-
0\"],\"date\":[\"2017-01-18\",\"2017-04-21\",\"2017-04-21\",\"2017-05-30\",\"2017-04-21\",\"2017-04-01
\",\"2017-05-13\",\"2017-01-27\",\"2016-10-11\",\"2013-07-31\",\"2016-12-29\",\"2016-07-10\",\"2010-04
-06\",\"2017-04-21\",\"2017-04-21\",\"2017-04-21\",\"2017-03-12\",\"2017-04-28\",\"2015-08-27\",\"2017
-05-18\",\"2016-07-10\",\"2017-03-25\",\"2014-11-22\",\"2017-04-21\",\"2017-04-21\",\"2017-05-22\",\"2
014-11-03\",\"2017-04-26\",\"2017-01-16\",\"2017-05-15\",\"2016-12-22\",\"2016-09-03\",\"2017-04-21\",
\"2017-04-07\",\"2017-02-18\",\"2017-04-21\",\"2016-11-17\",\"2017-04-20\",\"2017-04-21\",\"2016-06-20
\",\"2014-01-02\",\"2016-11-12\",\"2017-04-12\"],\"source\":[\"CRAN (R 3.4.0)\",\"local\",\"local\",\"
local\",\"local\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (
R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"local\",\"local\",\"local\",\"CR
AN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"cran (@1.16)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.3
.3)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"local\",\"cran (@0.12.11)\",\"CRAN (R 3.4.0)\",\"CRAN (R
3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"local\",\"CRAN
(R 3.4.0)\",\"CRAN (R 3.4.0)\",\"local\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"local\",\"CRAN (R 3.4
.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\",\"CRAN (R 3.4.0)\"]}}"
CMD ["R"]
```

5. Further customization

The `dockerfile()` function allows further customization regarding the R version or the used base image (cf. Rocker stack). Note that while choosing an R version for the Dockerfile explicitly is possible, the session to generate the required information (i.e. which packages are attached etc.) is still running the R version of the generating machine.

The following examples show usage of these options and the respective `FROM` statements in the Dockerfile.

```
df_custom <- dockerfile(from = NULL, r_version = "3.1.0", silent = TRUE)
print(df_custom@image)
```

```
FROM rocker/r-ver:3.1.0

df_custom <- dockerfile(from = NULL, image = "rocker/geospatial", silent = TRUE)
print(df_custom@image)

FROM rocker/geospatial

df_custom <- dockerfile(from = NULL, image = "rocker/verse:3.0.0", silent = TRUE)@image
print(df_custom@image)

[1] "rocker/verse"
```

6. CLI

A command line interface to the package functions is also available for Linux based on docopt.R. This allows integration into workflows and tools written in other programming languages than R.

You can make the command `containerit` available on your maching by linking the R script file delivered with the package as follows:

```
ln -s $(Rscript -e "cat(system.file(\"cli/container_it.R\", package=\"containerit\"))")
/usr/local/bin/containerit
```

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` alows to create and costumize Dockerfiles with minimal effort, which are suitable for packaging R analyses in the persistant 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

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

Thanks to the valuable questions and comments by our external partners, the project tasks were assessed and refocussed. 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 intented 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:

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.	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.
We know the potential of user interface bindings as the connecting entity of research compendia.	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.

We show the improvements in discovery and understanding of research when all aspects of research are explicitly linked in a meaningful way.	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.

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 shourt 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 it's 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 they 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 proofed 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 as 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

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 a call 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

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.


D-Lib Magazine

The Magazine of Digital Library Research

HOME | ABOUT D-LIB | CURRENT ISSUE | ARCHIVE | INDEXES | CALENDAR | AUTHOR GUI

D-Lib Magazine

January/February 2017
Volume 23, Number 1/2
Table of Contents

Opening the Publication Process with Executable Research Compendia

Daniel Nüst*
Institute for Geoinformatics, Münster
daniel.nuest@uni-muenster.de

Markus Konkol*
Institute for Geoinformatics, Münster
m.konkol@uni-muenster.de

Marc Schutzeichel
University and State Library, Münster
m.schutzeichel@uni-muenster.de

Edzer Pebesma
Institute for Geoinformatics, Münster
edzer.pebesma@uni-muenster.de

Christian Kray
Institute for Geoinformatics, Münster
c.kray@uni-muenster.de

Holger Przibytzin
University and State Library, Münster
holger.przibytzin@uni-muenster.de

Jörg Lorenz
University and State Library, Münster
holger.przibytzin@uni-muenster.de

*Shared co-first authorship

Nüst, D., Konkol, M., Pebesma, E., Kray, C., Schutzeichel, M., Przibytzin, H., Lorenz, J., 2017. Opening the Publication Process with Executable Research Compendia. D-Lib Magazine 23. doi:10.1045/january2017-nuest

Reproducible Computational Geosciences Workshop at AGILE Conference

05 Jan 2017

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.

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 (andretweet). Thanks!*

We look forward to your submission!

Investigating Docker and R

15 Dec 2016 | By Daniel Nüst

*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.

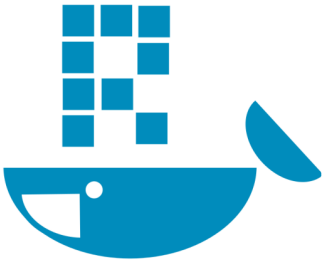
Thanks to Ben Marwick for contributing to this post! You know about a project using Docker and R? Get in touch.

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 it's 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 exists 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. In 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 `rize` 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

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 Interdisplinary 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 Schutzeichel

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 constitutating 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 rebuild 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 orginial 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 suppored 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 CERNs 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.

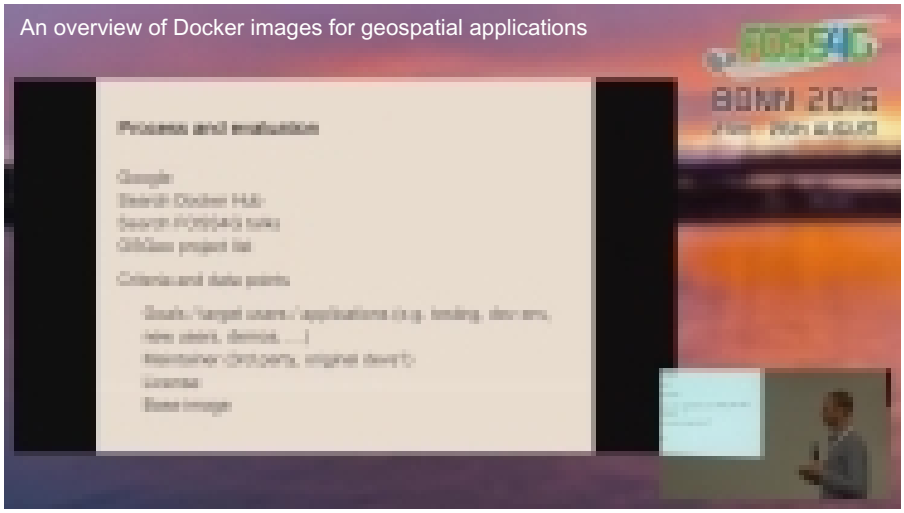
It was great for us to share our ideas of an *Executable Research Compendium* with the workshop attendees. The discussions and feedback was very helpful. We especially realized that we need to sharpen the distinctive aspects of our project when we talk about it. We're now working hard to implement this 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 recoriding 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

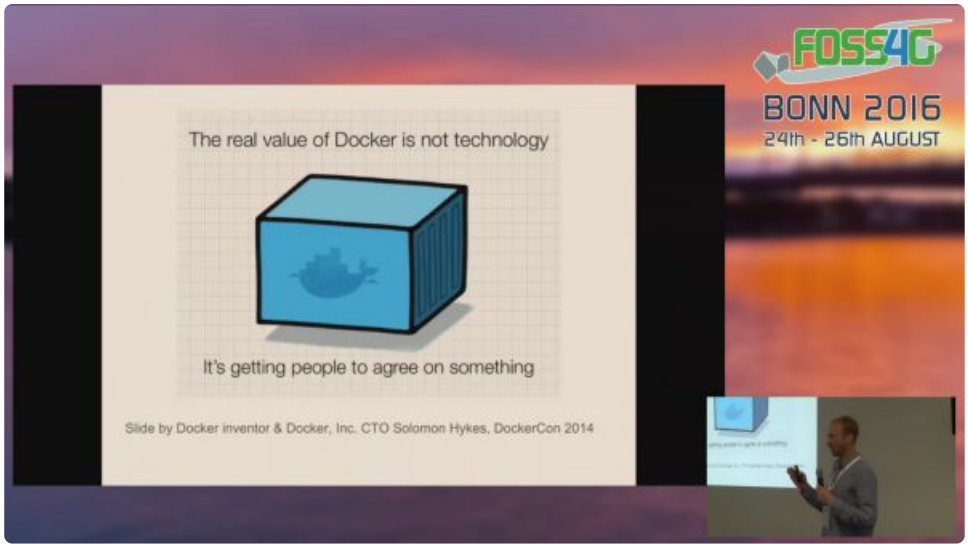
The main part of the work is published in the OSGeo wiki: a comprehensive list of Docker containers published by projects or third parties to use a large variety of tools, libraries, or Desktop applications in Docker containers. Check out the list at **https://wiki.osgeo.org/wiki/DockerImages**. Contributions 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.

Geocontainers

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

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 today[1], and because of the potential for useful integrations in the future[2].

The solution[3] 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.

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.

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

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 places 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

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

Motivation and goals

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. Collected or generated research data as well as procedures are published alongside a research paper.

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. As opposed to the collection of experimental data in labs or nature, computational experiments lend themselves very well for reproduction. Some of the reasons why scientists do not publish data and computational procedures that allow reproduction will be hard to change, e.g. privacy concerns in the data, fear for embarrassment or of losing a competitive advantage. Others reasons however involve technical aspects, and include the lack of standard procedures to publish such information and the lack of benefits after publishing them. *We aim to resolve these two technical aspects.*

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, this project develops standards and tools for executable research documents, and will demonstrate and evaluate these, initially focusing on the geosciences domains. 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.*

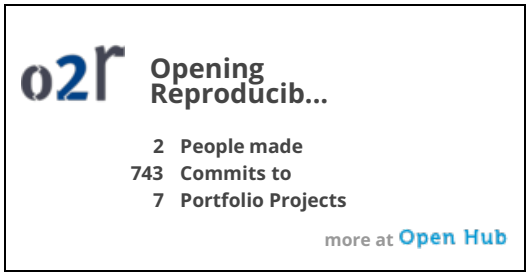
Learn about the accomplishment of these goals on the**results page**.

Open Source

For Open Science and reproducible research, we see not 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. Even the software of supported workflows (i.e. R) and underlying technologies (i.e. Docker) are 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.

Learn more about our projects on Open Hub and GitHub (currently [NA] repos having [NA] forks using [NA] languages).



People

o2r team members, supporting university staff, and external advisory board members in alphabetical order.

Team

- Dr. Stephanie Klötgen (ULB)
- Markus Konkol (ifgi)
- Prof. Dr. Christian Kray (ifgi)
- Jörg Lorenz (ULB)
- Daniel Nüst (ifgi)
- Prof. Dr. Edzer Pebesma (ifgi)
- Holger Przibytzin (ULB)
- Dr. Beate Tröger (ULB)

ulb Münster

ifgi
Institute for Geoinformatics
University of Münster

Former team members

- Rehan Chaudhary (ifgi, intern, 2017-01-17 to 2017-07-17)
- 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)
- 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 regularly enrich discussions with their insights. These are the partners in alphabetical order.

- Maarten Cleeren (Director of Product Management, Enriched Content at Elsevier; previouslyDr. Hylke Koers, Head of Content Innovation, Elsevier)
- 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, University of California Santa Barbara, Santa Barbara, CA)

ELSEVIER

Copernicus Publications
The Innovative Open Access Publisher

Aalto University
School of Science

spatial@ucsb
CENTER FOR SPATIAL STUDIES

Funding

This project *Opening Reproducible Research* (see also Offene Reproduzierbare Forschung @ DFG GEPRIS) is funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under project numbers PE 1632/10-1, KR 3930/3-1 and TR 864/6-1.

Results

Please find a project description and information about the team and partners on the [about page](/about), and a complete list of publications and presentations on the [publications page](/publications). ## Specifications & documentation o2r is an open project, so all our components 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 reprocibility 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. ## Implementation & demo We develop a reference implementation of the mentioned specification as Open Source software on GitHub: **{{ site.github.org }}** **Try the online demo at 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/)** (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!):

License