Reproducibility and Peer Review

Daniel Nüst

Keynote at virtual *kick off meeting of the* ***ORDS network***, December 1st 2020

doi.org/10.5281/zenodo.4292263

d.n@wwu.de | nordholmen.net/ | orcid.org/0000-0002-0024-5046

About me

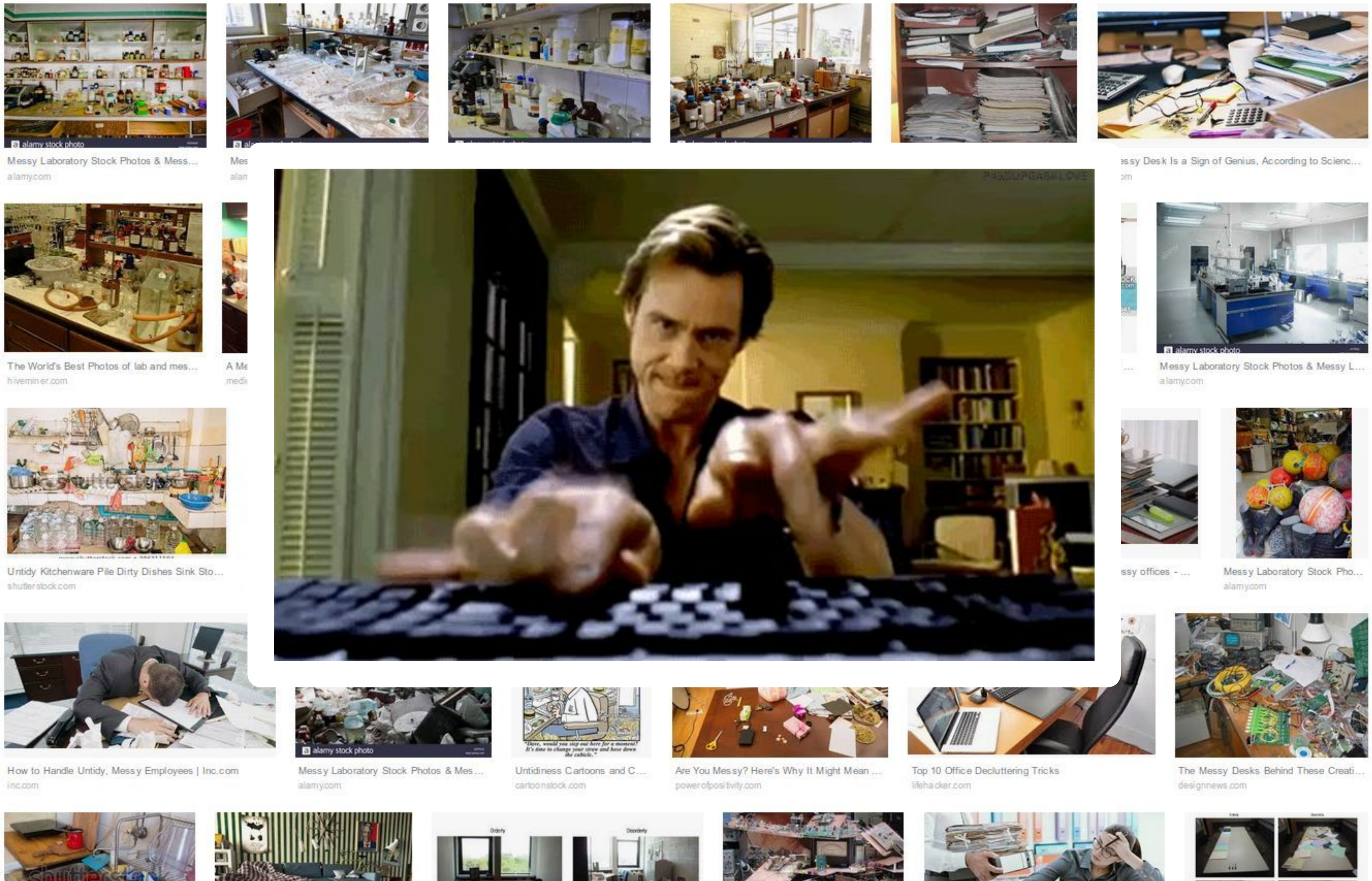
Self-identified research software engineer and *Open X* fan

Diploma Geoinformatics (Münster)
Software developer/consultant @ *52°North GmbH*, Münster
Researcher in project **Opening Reproducible Research** (o2r, https://o2r.info)
Founding member and deputy chair of de-**RSE** e. V., ms-RSE
Reproducibility Chair **AGILE conference** 2020, 2021
Co-PI of **CODECHECK**

https://orcid.org/0000-0002-0024-5046
GitHub.com/nuest | GitLab.com/nuest
@nordholmen

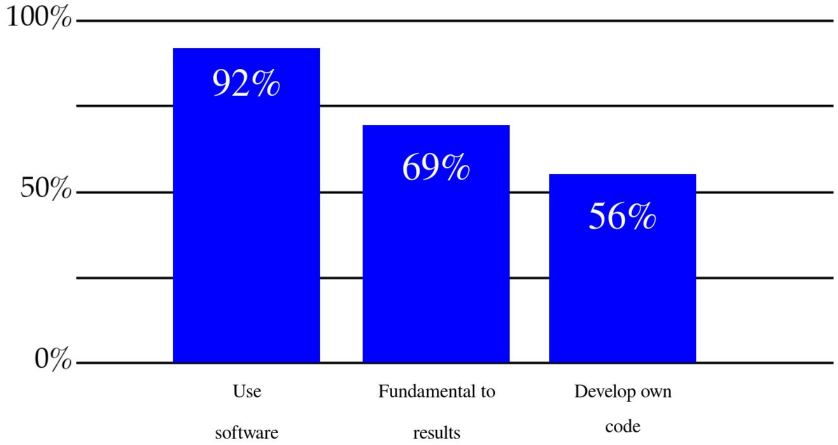
Open Science, #ReproducibleResearch, R, containerisation (Docker), Python, …

Ultimate Frisbee Endzonis


Messy Laboratory Stock Photos & Mess...
alamy.com
Mes...
alam...
...essy Desk Is a Sign of Genius, According to Scienc...
...om
The World's Best Photos of lab and mes...
hiveminer.com
A Me...
medi...
Messy Laboratory Stock Photos & Messy L...
alamy.com
Untidy Kitchenware Pile Dirty Dishes Sink Sto...
shutterstock.com
...ssy offices - ...
Messy Laboratory Stock Pho...
alamy.com
How to Handle Untidy, Messy Employees | Inc.com
inc.com
Messy Laboratory Stock Photos & Mes...
alamy.com
Untidiness Cartoons and C...
cartoonstock.com
Are You Messy? Here's Why It Might Mean ...
powerofpositivity.com
Top 10 Office Decluttering Tricks
lifehacker.com
The Messy Desks Behind These Creati...
designnews.com

Back to 2010 The Software Sustainability Institute (SSI, UK) run a study (1000 randomly chosen researchers) …

“It's impossible to conduct research without software, say 7 out of 10 UK researchers”



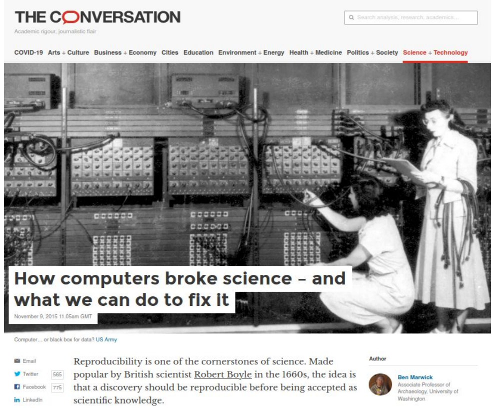
https://www.software.ac.uk/blog/2014-12-04-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers

A study of Nature papers from Jan-March 2016 reveals that

“32 of the 40 papers examined mention software, and the 32 papers contain 211 mentions of distinct pieces of software, for an average of 6.5 mentions per paper.”

Nangia, Udit; Katz, Daniel S. (2017): Understanding Software in Research: Initial Results from Examining Nature and a Call for Collaboration. doi:**10.1109/eScience.2017.78**

https://giphy.com/gifs/with-computers-fascination-PxSFAnuubLkSA



How computers broke science – and what we can do to fix it

November 9, 2015 11.05am GMT

Computer… or black box for data? US Army

Reproducibility is one of the cornerstones of science. Made popular by British scientist Robert Boyle in the 1660s, the idea is that a discovery should be reproducible before being accepted as scientific knowledge.

Author

Ben Marwick
Associate Professor of Archaeology, University of Washington

https://theconversation.com/how-computers-broke-science-and-what-we-can-do-to-fix-it-49938

https://giphy.com/gifs/david-hasselhoff-M3o3fL9nnxG4o

Crisis? Crisis of what?

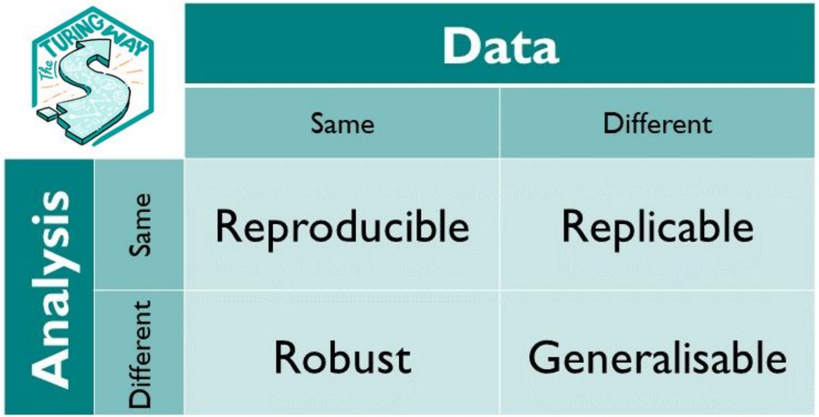
Credibility crisis?

Replicability crisis?

Reproducibility crisis?

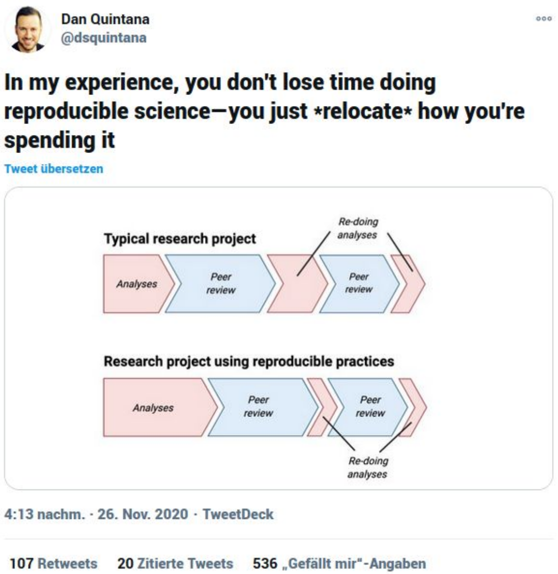
Robustness crisis?

Generalisability crisis?


The TURING WAY

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html



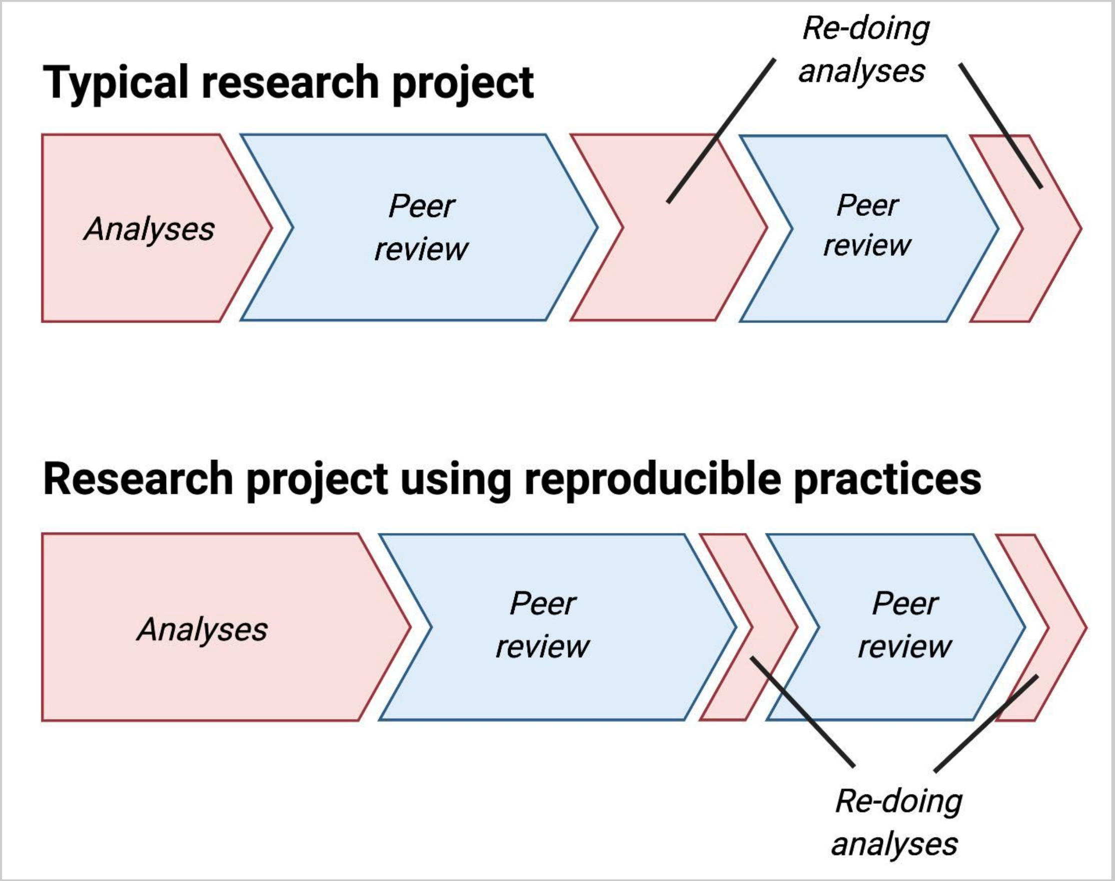
Dan Quintana
@dsquintana

In my experience, you don't lose time doing reproducible science—you just *relocate* how you're spending it

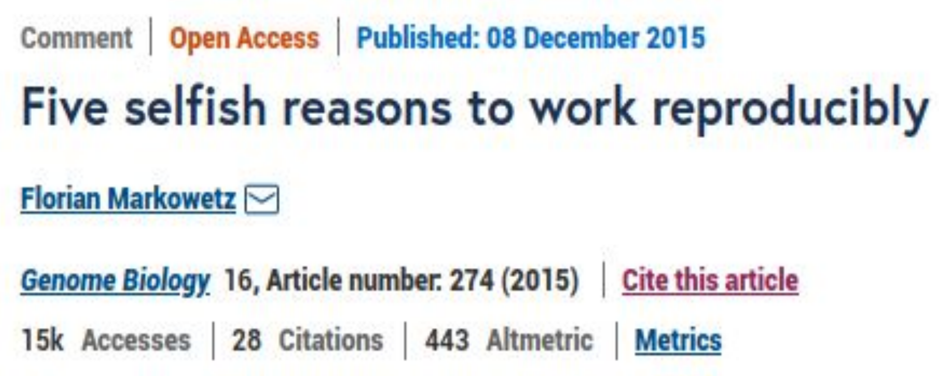
Tweet übersetzen

4:13 nachm. · 26. Nov. 2020 · TweetDeck

107 Retweets 20 Zitierte Tweets 536 „Gefällt mir"-Angaben



Quintana, D. S. (2020, November 28). Five things about open and reproducible science that every early career researcher should know. https://doi.org/10.17605/OSF.IO/DZTVQ



Comment | Open Access | Published: 08 December 2015

Five selfish reasons to work reproducibly

Florian Markowetz

Genome Biology 16, Article number: 274 (2015) | Cite this article

15k Accesses | 28 Citations | 443 Altmetric | Metrics

https://doi.org/10.1186/s13059-015-0850-7

1. reproducibility helps to avoid disaster
2. reproducibility makes it easier to write papers
3. **reproducibility helps reviewers see it your way**
4. reproducibility enables continuity of your work
5. reproducibility helps to build your reputation

Reproducible Research & research software

Peer Review

Reproducible research and peer review are cornerstones of science. But are they getting along?

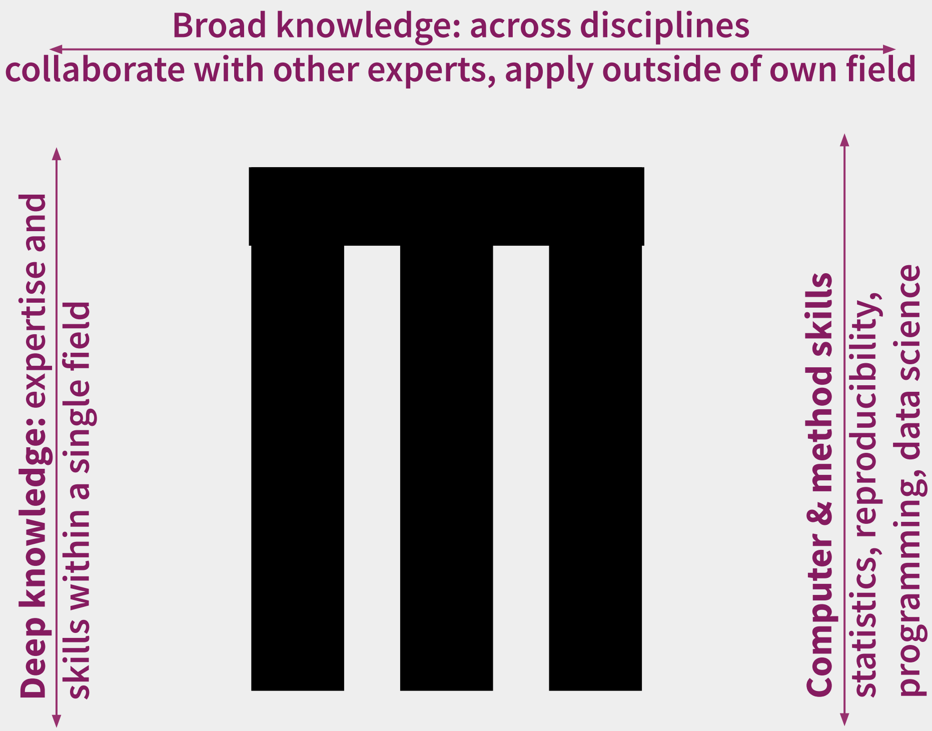
The many problems of science

Publish or perish
Broken metrics (citations, JIF)
Structural change not considering senior academics
Publication bias
Long-term funding for tools & infrastructure
HARKing
p-Hacking
Scholarly communication 1.0
Lack of reusability
Lack of transparency
Lack of reproducibility
Reinventing the wheel
Retraction practices
Not invented here syndrome
Fraud
Imposter syndrome
No “negative” citation
...

https://giphy.com/gifs/bbcamerica-cute-animals-lifestory-Ze3RpHue7qkwvcYOOf

Open Science (OER, OA, OS, OPR)
Registered reports/preregistration
Altmetrics
Preprints
Leiden Manifesto
DORA
Vienna Principles
Citing data and software
Software papers
Data and software as products of research
RSEng & RSEs (software sustainability)
CRediT
Research Compendia
Ten Hot Topics Around Scholarly Publishing
Code review (PyOpenSci, ROpenSci, JOSS)
...

Traditional and modern scientists



Broad knowledge: across disciplines

collaborate with other experts, apply outside of own field

Deep knowledge: expertise and

skills within a single field

Computer & method skills

statistics, reproducibility,

programming, data science

US RSE

US RESEARCH SOFTWARE ENGINEER ASSOCIATION

de RSE

RSE

RESEARCH SOFTWARE ENGINEERS ASSOCIATION

<ReSA>

Research Software Alliance

https://en.wikipedia.org/wiki/T-shaped_skills

https://doi.org/10.1007/s10816-015-9272-9

https://jakevdp.github.io/blog/2014/08/22/hacking-academia/
https://www.sciencemag.org/careers/2013/05/when-all-science-becomes-data-science
https://escience.washington.edu/community-level-data-science-and-its-spheres-of-influence-beyond-novelty-squared/

Code Review

rOpenSci

Boettiger, C., Chamberlain, S., Hart, E., & Ram, K. (2015). Building Software, Building Community: Lessons from the rOpenSci Project. Journal of Open Research Software, 3(1), e8. doi:10.5334/jors.bu

pyOpenSci

Journal of open research software

The Journal of Open Source Software

Code Review Community Working Group

How to get reproducible research and peer review to get along?

Reproducible computational research in journals & conferences

ACM Transactions on Mathematical Software

Journal of Statistical Software

Volume 220 March 2021 ISSN 0022-314X

ELSEVIER

JOURNAL OF NUMBER THEORY

Editor-in-Chief
Dorian Goldfeld

Managing Editors

Editorial Board
Andrew Booker
Gebhard Böckle
Stephen D. Miller
Ivan Mirkovic

ReproZip

Biostatistics

Information Systems

DATABASES: CREATION, MANAGEMENT AND UTILIZATION

ELECTRONIC ACCESS

SC19 PROGRAM

Reproducibility Initiative

SIGMOD/PODS AMSTERDAM 2019

Results Replicated acm

NEURAL INFORMATION PROCESSING SYSTEMS

AGILE Reproducible Paper Guidelines 🇬🇧 🇪🇸

https://doi.org/10.17605/OSF.IO/CB7Z8

Created by AGILE Initiative in 2019, see report at https://osf.io/hupxr/

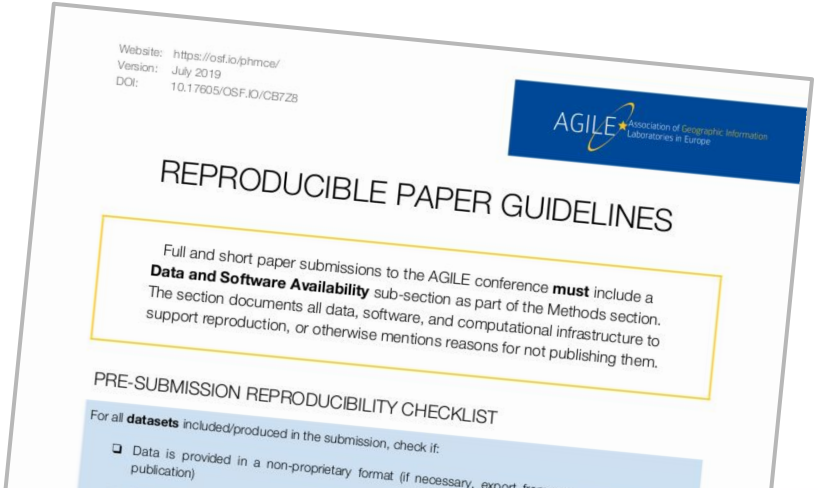
Transparency & Reproducibility

GIScience

https://osf.io/phmce/wiki/home/

Promotion

Acknowledge spectrum



Website: https://osf.io/phmce/
Version: July 2019
DOI: 10.17605/OSF.IO/CB7Z8

AGILE Association of Geographic Information Laboratories in Europe

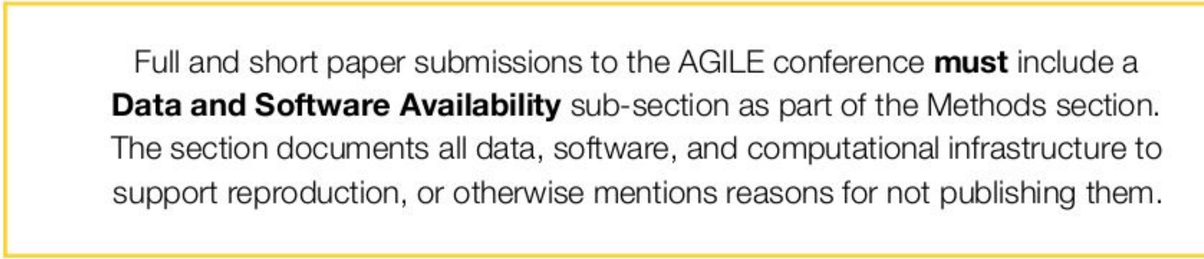
REPRODUCIBLE PAPER GUIDELINES

Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.

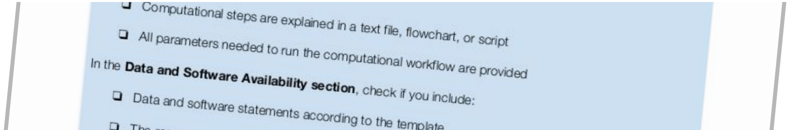
PRE-SUBMISSION REPRODUCIBILITY CHECKLIST

For all **datasets** included/produced in the submission, check if:

- ❑ Data is provided in a non-proprietary format (if necessary, export
publication)



Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.



- ❑ Computational steps are explained in a text file, flowchart, or script
- ❑ All parameters needed to run the computational workflow are provided

In the **Data and Software Availability section**, check if you include:

- ❑ Data and software statements according to the template

The guidelines

https://doi.org/10.17605/OSF.IO/CB7Z8

Author guidelines

Data in Research Papers

Computational workflows in Research Papers

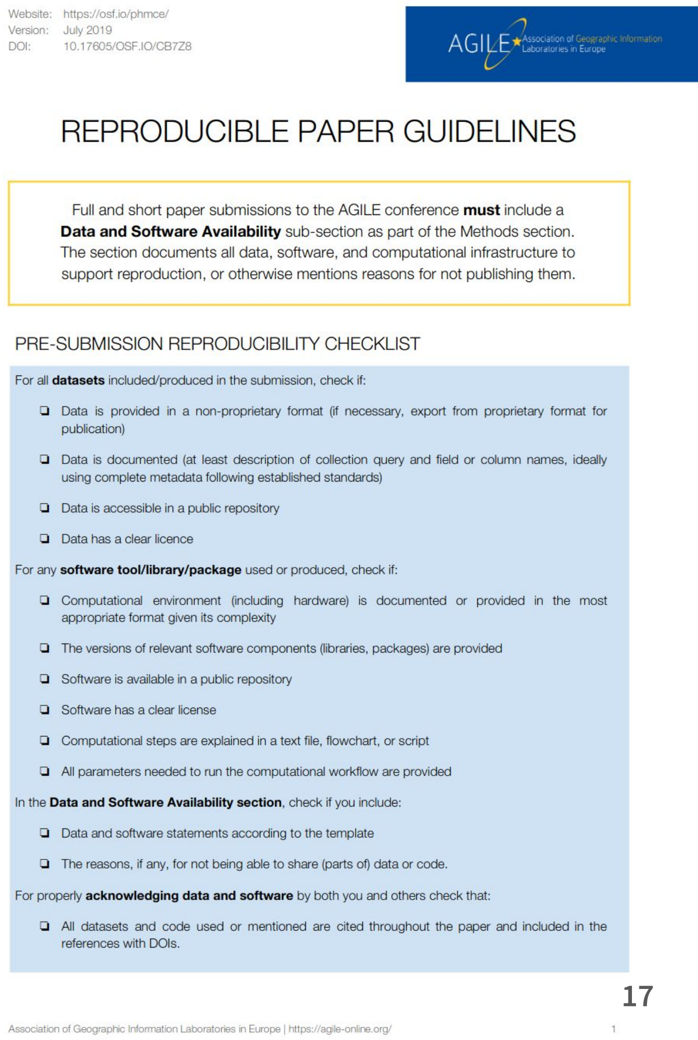
Pre-submission checklist

Writing DASA section

Rationale/Motivation/Vision

Reviewer guidelines

Reproducibility reviewer guidelines (WIP)



Website: https://osf.io/phmce/
Version: July 2019
DOI: 10.17605/OSF.IO/CB7Z8

REPRODUCIBLE PAPER GUIDELINES

Full and short paper submissions to the AGILE conference **must** include a **Data and Software Availability** sub-section as part of the Methods section. The section documents all data, software, and computational infrastructure to support reproduction, or otherwise mentions reasons for not publishing them.

PRE-SUBMISSION REPRODUCIBILITY CHECKLIST

For all **datasets** included/produced in the submission, check if:

- □ Data is provided in a non-proprietary format (if necessary, export from proprietary format for publication)
- □ Data is documented (at least description of collection query and field or column names, ideally using complete metadata following established standards)
- □ Data is accessible in a public repository
- □ Data has a clear licence

For any **software tool/library/package** used or produced, check if:

- □ Computational environment (including hardware) is documented or provided in the most appropriate format given its complexity
- □ The versions of relevant software components (libraries, packages) are provided
- □ Software is available in a public repository
- □ Software has a clear license
- □ Computational steps are explained in a text file, flowchart, or script
- □ All parameters needed to run the computational workflow are provided

In the **Data and Software Availability section**, check if you include:

- □ Data and software statements according to the template
- □ The reasons, if any, for not being able to share (parts of) data or code.

For properly **acknowledging data and software** by both you and others check that:

- □ All datasets and code used or mentioned are cited throughout the paper and included in the references with DOIs.

Association of Geographic Information Laboratories in Europe | https://agile-online.org/ 1

AGILE conference review process

Proceedings:
https://www.agile-giscience-series.net/review_process.html

Process documentation:
https://osf.io/7rjpe/

Reproducibility review *after* accept/reject decisions, triggered by regular reviewer

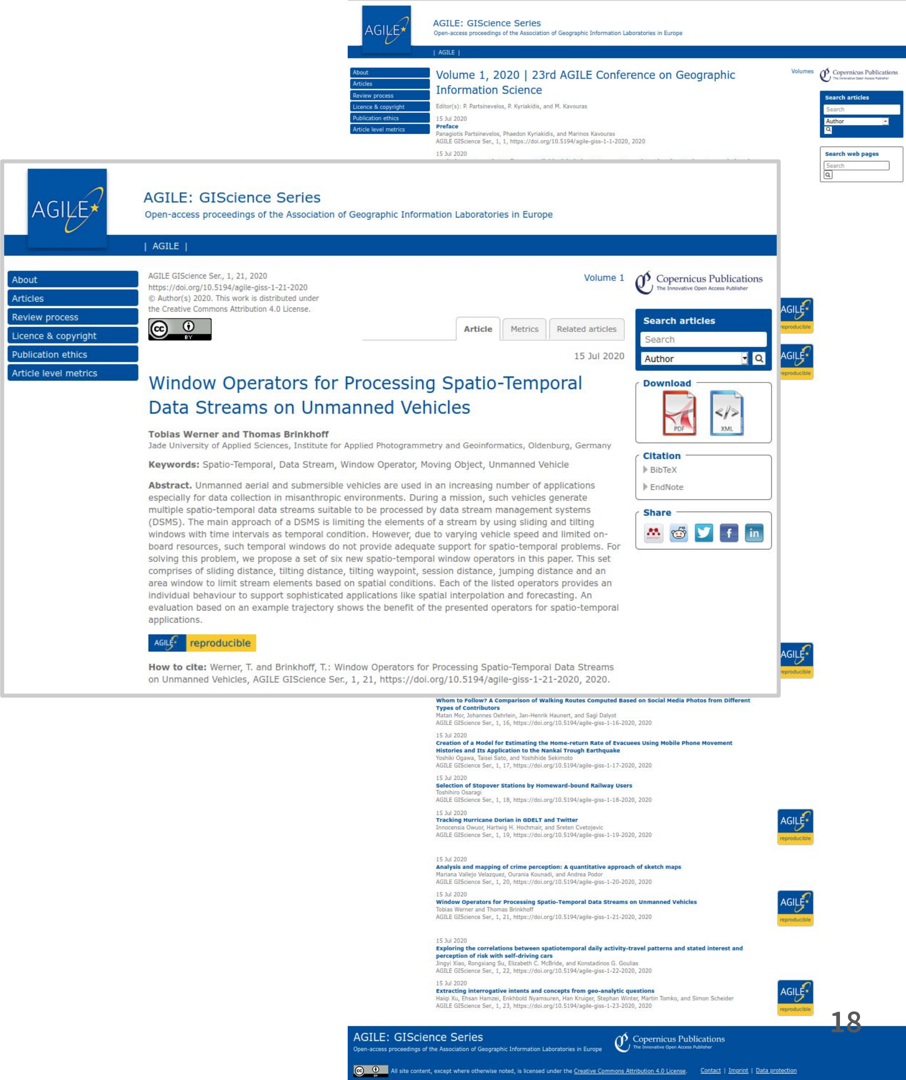
Reproducibility review & communication

Community conference & coronavirus

Badges on proceedings page

Presentation at conference

Read full report at https://osf.io/7rjpe/



AGILE: GIScience Series
Open-access proceedings of the Association of Geographic Information Laboratories in Europe
| AGILE |

Volume 1, 2020 | 23rd AGILE Conference on Geographic Information Science

Editor(s): P. Partsinevelos, P. Kyriakidis, and M. Kavouras

15 Jul 2020
Preface
Panagiotis Partsinevelos, Phaedon Kyriakidis, and Marinos Kavouras
AGILE GIScience Ser., 1, 1, https://doi.org/10.5194/agile-giss-1-1-2020, 2020

15 Jul 2020

AGILE: GIScience Series
Open-access proceedings of the Association of Geographic Information Laboratories in Europe
| AGILE |

AGILE GIScience Ser., 1, 21, 2020
https://doi.org/10.5194/agile-giss-1-21-2020
© Author(s) 2020. This work is distributed under
the Creative Commons Attribution 4.0 License.

15 Jul 2020

Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles

Tobias Werner and Thomas Brinkhoff
Jade University of Applied Sciences, Institute for Applied Photogrammetry and Geoinformatics, Oldenburg, Germany

Keywords: Spatio-Temporal, Data Stream, Window Operator, Moving Object, Unmanned Vehicle

Abstract. Unmanned aerial and submersible vehicles are used in an increasing number of applications especially for data collection in misanthropic environments. During a mission, such vehicles generate multiple spatio-temporal data streams suitable to be processed by data stream management systems (DSMS). The main approach of a DSMS is limiting the elements of a stream by using sliding and tilting windows with time intervals as temporal condition. However, due to varying vehicle speed and limited on-board resources, such temporal windows do not provide adequate support for spatio-temporal problems. For solving this problem, we propose a set of six new spatio-temporal window operators in this paper. This set comprises of sliding distance, tilting distance, tilting waypoint, session distance, jumping distance and an area window to limit stream elements based on spatial conditions. Each of the listed operators provides an individual behaviour to support sophisticated applications like spatial interpolation and forecasting. An evaluation based on an example trajectory shows the benefit of the presented operators for spatio-temporal applications.

How to cite: Werner, T. and Brinkhoff, T.: Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles, AGILE GIScience Ser., 1, 21, https://doi.org/10.5194/agile-giss-1-21-2020, 2020.

Whom to Follow? A Comparison of Walking Routes Computed Based on Social Media Photos from Different Types of Contributors
Matan Mor, Johannes Oehrlein, Jan-Henrik Haunert, and Sagi Dalyot
AGILE GIScience Ser., 1, 16, https://doi.org/10.5194/agile-giss-1-16-2020, 2020

15 Jul 2020
Creation of a Model for Estimating the Home-return Rate of Evacuees Using Mobile Phone Movement Histories and Its Application to the Nankai Trough Earthquake
Yoshiki Ogawa, Taisei Sato, and Yoshihide Sekimoto
AGILE GIScience Ser., 1, 17, https://doi.org/10.5194/agile-giss-1-17-2020, 2020

15 Jul 2020
Selection of Stopover Stations by Homeward-bound Railway Users
Toshihiro Osaragi
AGILE GIScience Ser., 1, 18, https://doi.org/10.5194/agile-giss-1-18-2020, 2020

15 Jul 2020
Tracking Hurricane Dorian in GDELT and Twitter
Innocensia Owuor, Hartwig H. Hochmair, and Sreten Cvetojevic
AGILE GIScience Ser., 1, 19, https://doi.org/10.5194/agile-giss-1-19-2020, 2020

15 Jul 2020
Analysis and mapping of crime perception: A quantitative approach of sketch maps
Mariana Vallejo Velazquez, Ourania Kounadi, and Andrea Podor
AGILE GIScience Ser., 1, 20, https://doi.org/10.5194/agile-giss-1-20-2020, 2020

15 Jul 2020
Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles
Tobias Werner and Thomas Brinkhoff
AGILE GIScience Ser., 1, 21, https://doi.org/10.5194/agile-giss-1-21-2020, 2020

15 Jul 2020
Exploring the correlations between spatiotemporal daily activity-travel patterns and stated interest and perception of risk with self-driving cars
Jingyi Xiao, Rongxiang Su, Elizabeth C. McBride, and Konstadinos G. Goulias
AGILE GIScience Ser., 1, 22, https://doi.org/10.5194/agile-giss-1-22-2020, 2020

15 Jul 2020
Extracting interrogative intents and concepts from geo-analytic questions
Haiqi Xu, Ehsan Hamzei, Enkhbold Nyamsuren, Han Kruiger, Stephan Winter, Martin Tomko, and Simon Scheider
AGILE GIScience Ser., 1, 23, https://doi.org/10.5194/agile-giss-1-23-2020, 2020

AGILE: GIScience Series
Open-access proceedings of the Association of Geographic Information Laboratories in Europe

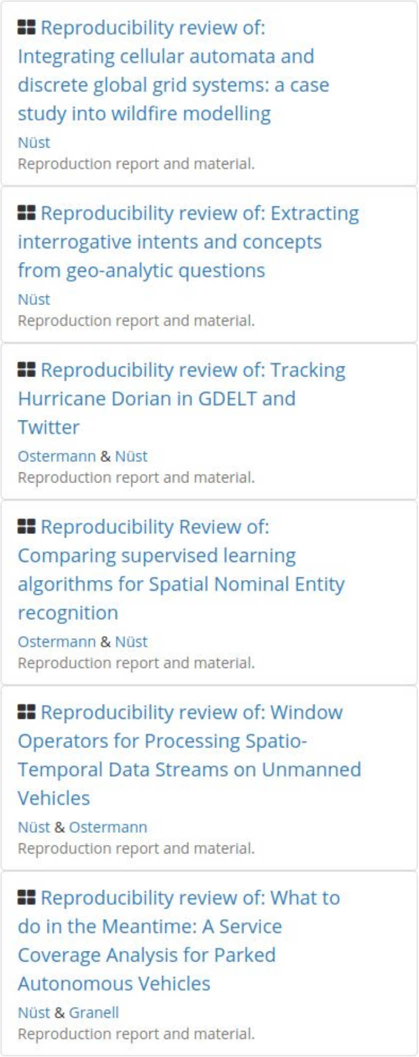
Reproducibility review results

6 reproducibility reports published

16 not possible/not attempted

(5 of which after communication with authors):

- **no starting point** in the paper
- **documentation insufficient** for third party
- **sensitive/confidential/commercial data**
- **proprietary** software
- software paper
- (conceptual papers)



Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling
Nüst
Reproduction report and material.

Reproducibility review of: Extracting interrogative intents and concepts from geo-analytic questions
Nüst
Reproduction report and material.

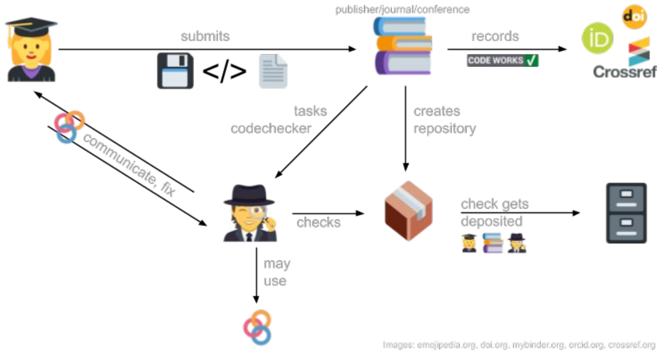
Reproducibility review of: Tracking Hurricane Dorian in GDELT and Twitter
Ostermann & Nüst
Reproduction report and material.

Reproducibility Review of: Comparing supervised learning algorithms for Spatial Nominal Entity recognition
Ostermann & Nüst
Reproduction report and material.

Reproducibility review of: Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles
Nüst & Ostermann
Reproduction report and material.

Reproducibility review of: What to do in the Meantime: A Service Coverage Analysis for Parked Autonomous Vehicles
Nüst & Granell
Reproduction report and material.

https://osf.io/6k5fh/



Images: emojipedia.org, doi.org, mybinder.org, orcid.org, crossref.org

CODE CHECK ✓

https://codecheck.org.uk/

Independent execution of computations underlying research articles.

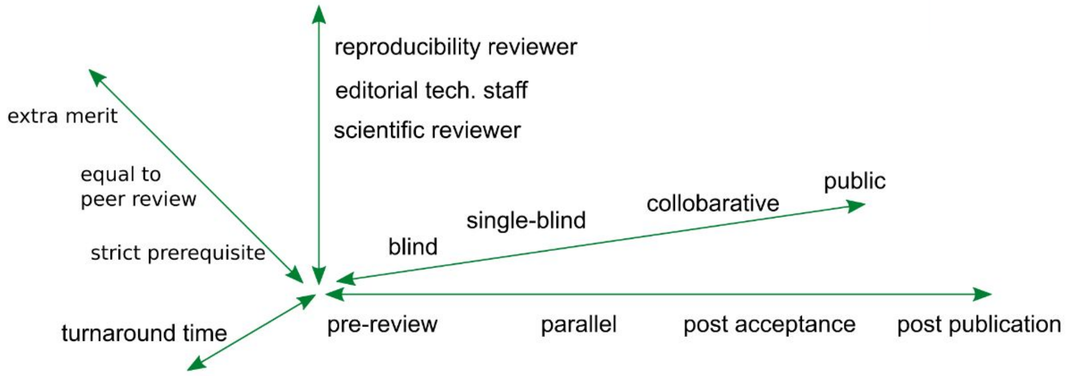
Codecheckers record but don’t investigate or fix.

Communication between humans is key.

Credit is given to codecheckers.

Workflows must be auditable.

CODE WORKS ✓



reproducibility reviewer

editorial tech. staff

scientific reviewer

extra merit

equal to peer review

strict prerequisite

turnaround time

blind

single-blind

collobarative

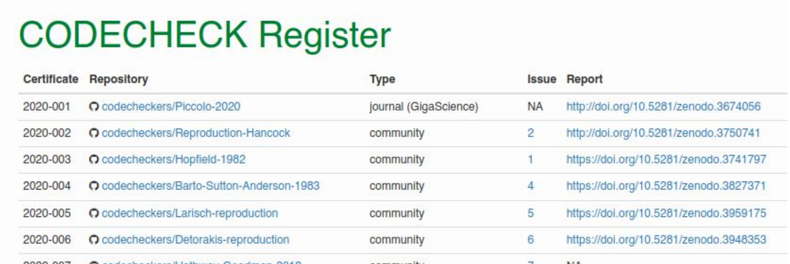
public

pre-review

parallel

post acceptance

post publication



CODECHECK Register

Certificate	Repository	Type	Issue	Report
2020-001	codecheckers/Piccolo-2020	journal (GigaScience)	NA	http://doi.org/10.5281/zenodo.3674056
2020-002	codecheckers/Reproduction-Hancock	community	2	http://doi.org/10.5281/zenodo.3750741
2020-003	codecheckers/Hopfield-1982	community	1	https://doi.org/10.5281/zenodo.3741797
2020-004	codecheckers/Barto-Sutton-Anderson-1983	community	4	https://doi.org/10.5281/zenodo.3827371
2020-005	codecheckers/Larisch-reproduction	community	5	https://doi.org/10.5281/zenodo.3959175
2020-006	codecheckers/Detorakis-reproduction	community	6	https://doi.org/10.5281/zenodo.3948353
2020-007	codecheckers/Hathway-Goodman-2018	community	7	NA

Reproducible AGILE and CODECHECK: Highlights of Lessons learned

CODE CHECK https://codecheck.org.uk/

AGILE reproducible

Spectrum or layers of reproducibility

Effect of guidelines at AGILE: **improved reproducibility**

Reproducibility reports/CODECHECK certificates full of **recommendations** for improvement, well received by authors, even included in revision before publication

Good practices spread slowly, establishing a **process** is tedious

Challenges for reproducibility reviewer: Inconsistencies and disconnects (figures), lack of documentation, unknown runtimes vs. no subsets of data, lack of repro guidance

Reproductions are **rewarding** and educational, matching expertises tricky

Safety net (👀), not **security**

What can you do today?

What can scientists do?

Do what Heidi tells you! [**slides**] One step at a time.

Create and publish **Research Compendia** (**Your code is good enough**!): **https://research-compendium.science/**

Become a **codechecker** or **reprohacker**.

Strive to be open science champion **especially** if you're in a senior position in your field (hiring, contributorship, funding & openness). [**RIOT Science Club Talk by Gavin Buckinham**; **preprint by Sam Westwood**]

What can communities and institutions do?

Introduce reproducibility reviews - CODECHECK (or not)!

Workshops on RCR, ReproHacks

Provide support (**R2S2**, Anja)

Rewards and incentives

Awareness > Change

https://giphy.com/gifs/chicagodancecrash-KCqjrcPfL55q3MkgHZ

How does the future of reproducible research in peer review look like?

Reproducibility is **possible**, but disciplines/communities must agree what “peer review” entails and acknowledge the **efforts** (ECRs, RSEs) in a **positive** way.

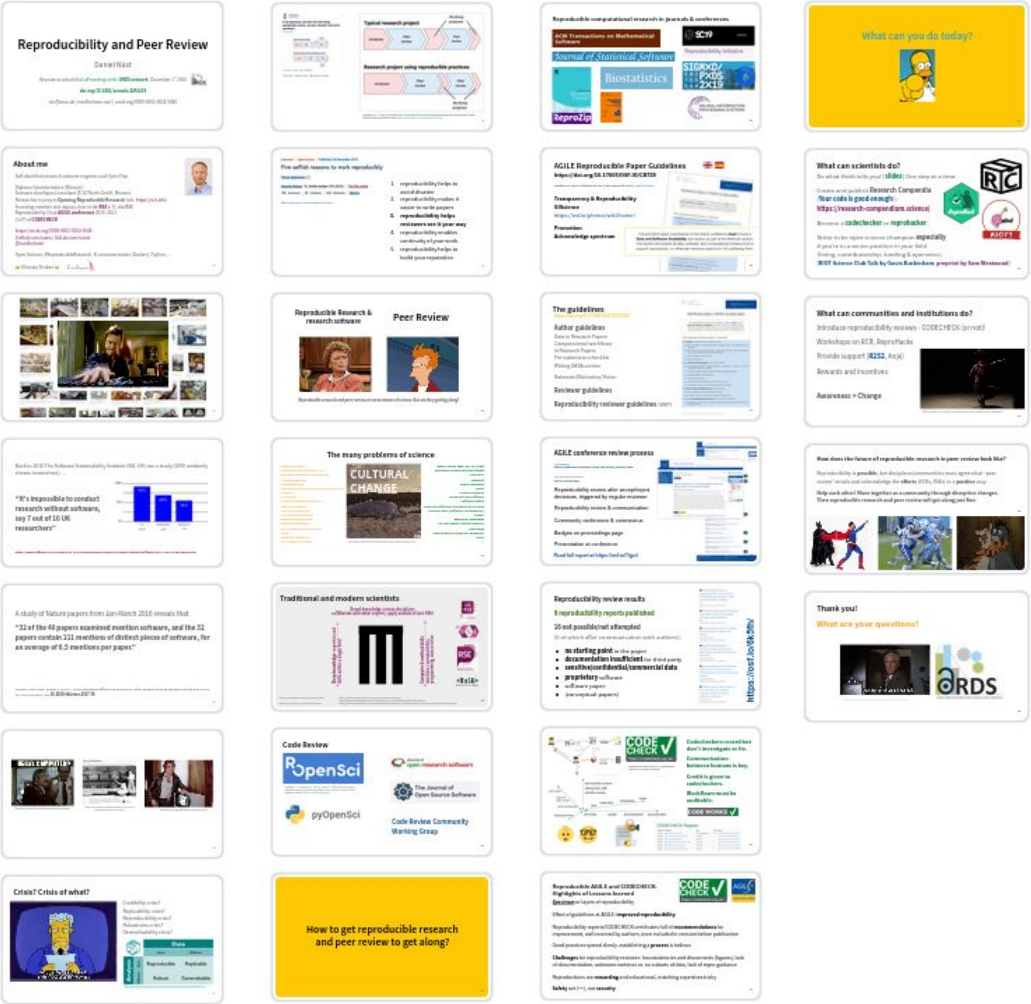
Help each other! Move together as a community through disruptive changes.
Then reproducible research and peer review will get along just fine.

Thank you!

What are your questions?

https://giphy.com/gifs/mrw-save-counting-wvWJOoYmFnSp2

Open Reproducible Data Science and Statistics



Reproducibility and Peer Review

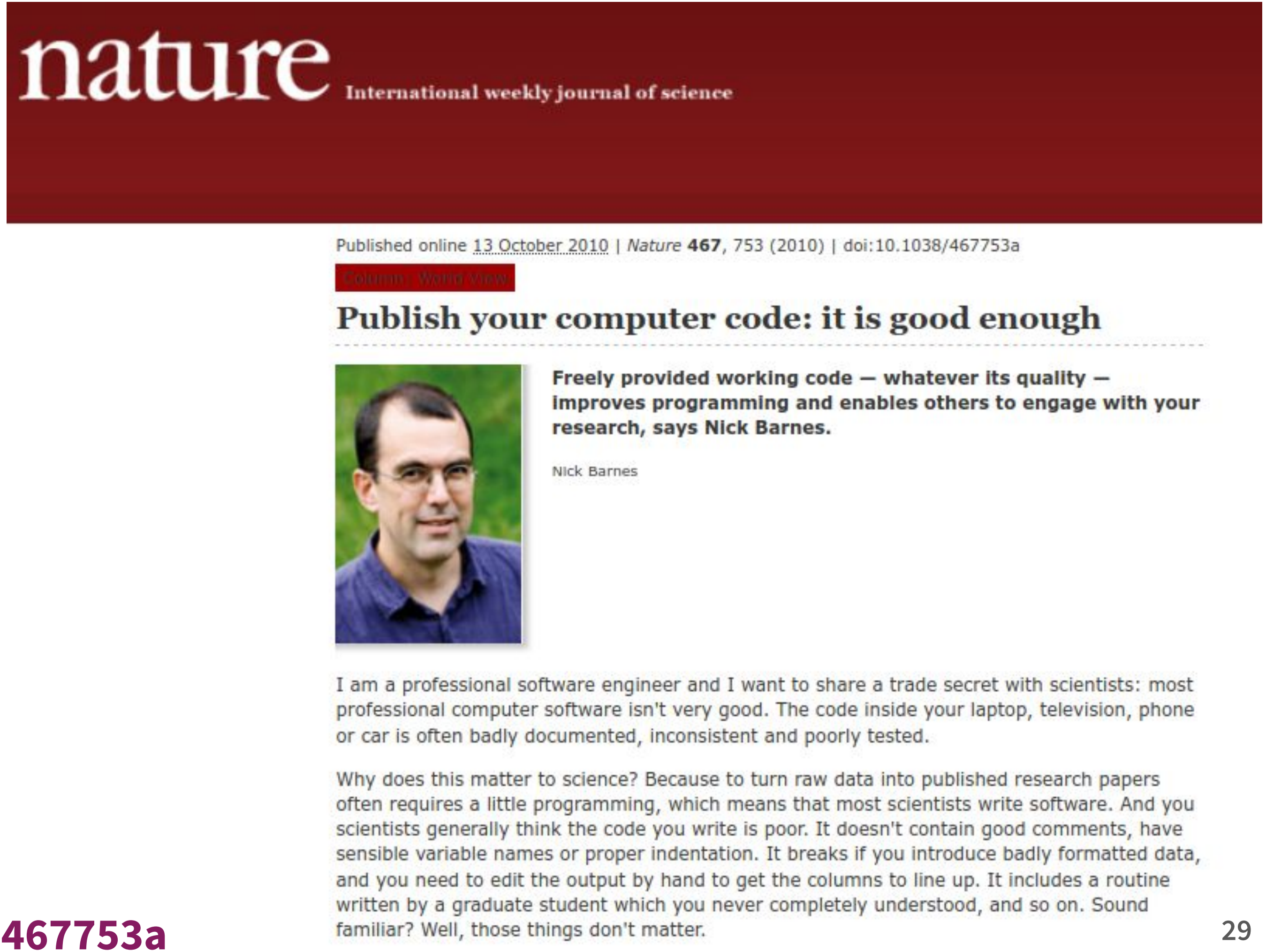
Daniel Nüst

Keynote at virtual *kick off meeting of the* ***ORDS network***, December 1st 2020

doi.org/10.5281/zenodo.4292263

orcid.org/0000-0002-0024-5046

bonus slides



nature International weekly journal of science

Published online 13 October 2010 | *Nature* **467**, 753 (2010) | doi:10.1038/467753a

Column : World View

Publish your computer code: it is good enough

Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says Nick Barnes.

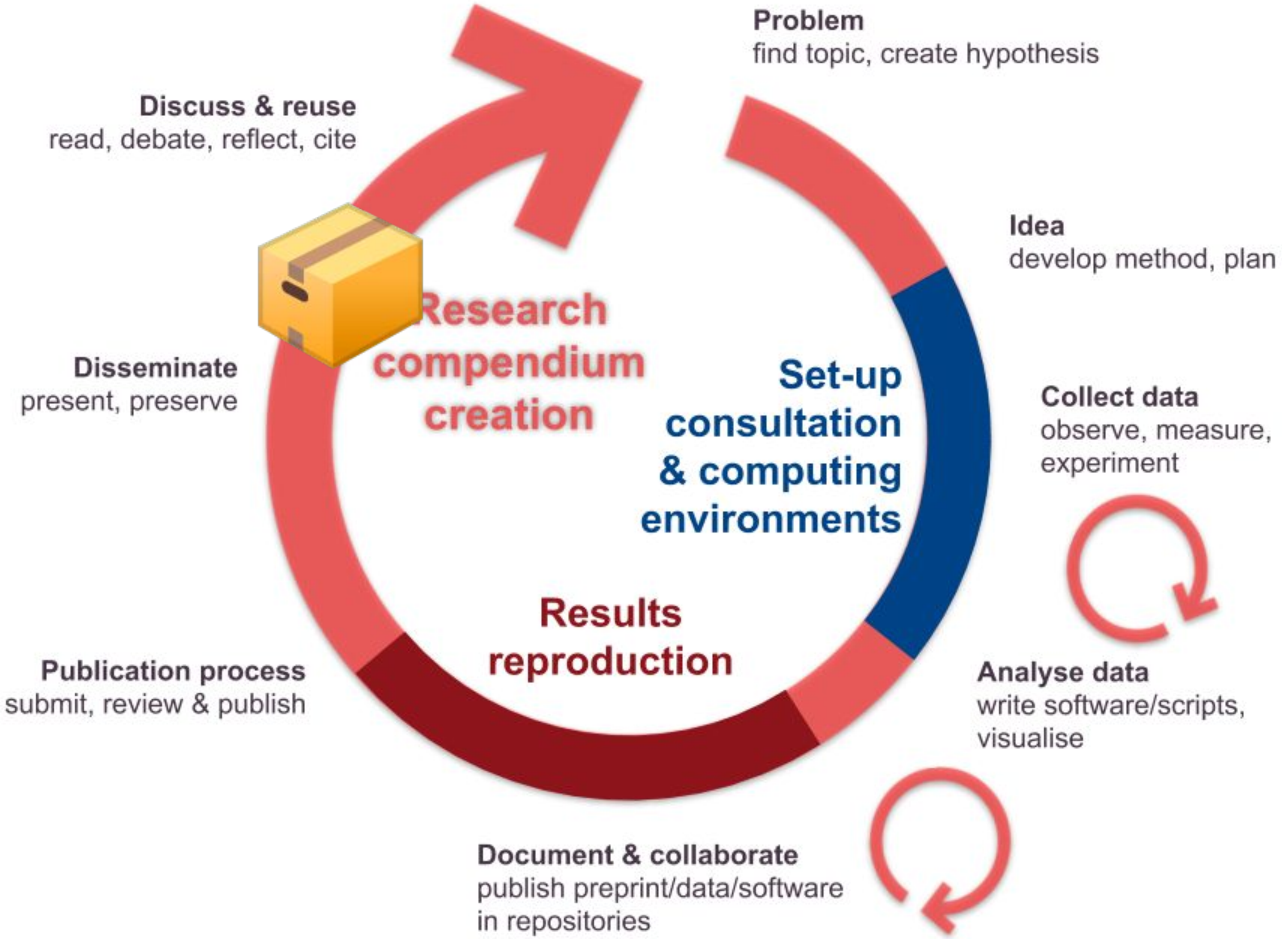
Nick Barnes

I am a professional software engineer and I want to share a trade secret with scientists: most professional computer software isn't very good. The code inside your laptop, television, phone or car is often badly documented, inconsistent and poorly tested.

Why does this matter to science? Because to turn raw data into published research papers often requires a little programming, which means that most scientists write software. And you scientists generally think the code you write is poor. It doesn't contain good comments, have sensible variable names or proper indentation. It breaks if you introduce badly formatted data, and you need to edit the output by hand to get the columns to line up. It includes a routine written by a graduate student which you never completely understood, and so on. Sound familiar? Well, those things don't matter.

https://doi.org/10.1038/467753a

Reproducible Research Support Services in the Research Lifecycle



Problem
find topic, create hypothesis

Idea
develop method, plan

Collect data
observe, measure, experiment

Analyse data
write software/scripts, visualise

Document & collaborate
publish preprint/data/software in repositories

Publication process
submit, review & publish

Disseminate
present, preserve

Discuss & reuse
read, debate, reflect, cite

Research compendium creation

Set-up consultation & computing environments

Results reproduction

R2S2

Reproducible **R**esearch
Support **S**ervice

https://go.wwu.de/r2s2

EDUCATION OF

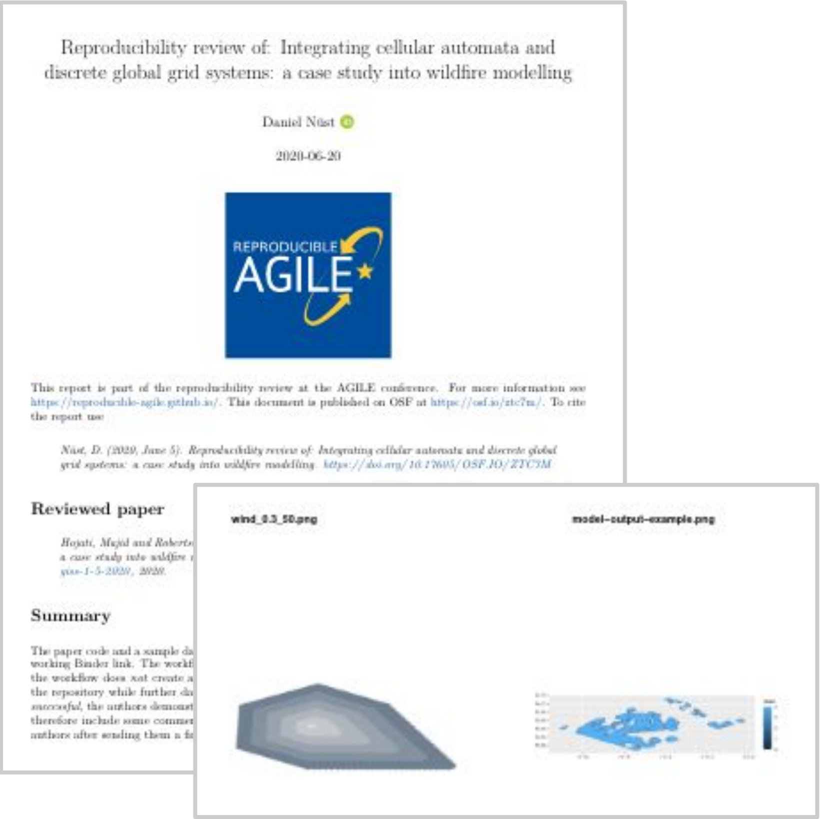
BETTER CONDITIONS FOR

RESEARCHERS USING SOFTWARE DEVELOPERS

LEAD TO BETTER

RESEARCH

Reproducibility review reports



Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling

Daniel Nüst

2020-06-20

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.gitlab.io/. This document is published on OSF at https://osf.io/ztc7m/. To cite the report use

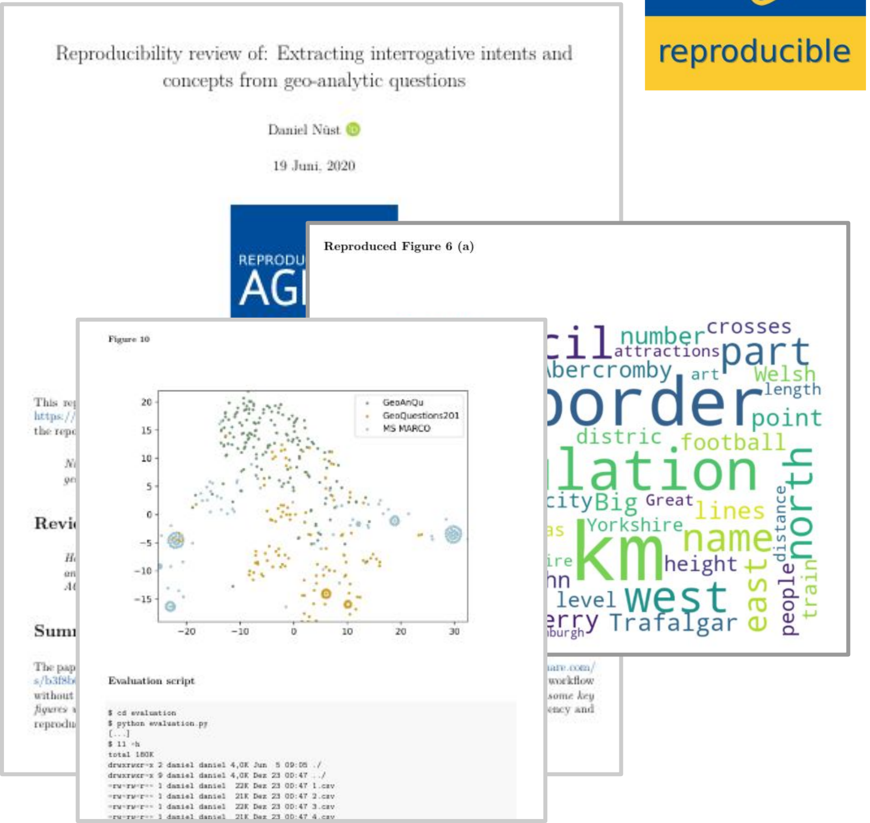
Nüst, D. (2020, June 5). Reproducibility review of: Integrating cellular automata and discrete global grid systems: a case study into wildfire modelling. https://doi.org/10.17605/OSF.IO/ZTC7M

Reviewed paper

Hojati, Majid and Roberts…
a case study into wildfire …
gisc-1-5-2020, 2020.

Summary

The paper code and a sample da…
working Binder link. The workf…
the workflow does not create a…
the repository while further da…
successful, the authors demonst…
therefore include some commen…
authors after sending them a fi…



Reproducibility review of: Extracting interrogative intents and concepts from geo-analytic questions

Daniel Nüst

19 Juni, 2020

This re…
https://…
the repo…

Revi…

Summ…

The pap… …are.com/
s/b3f8b… …workflow
without … *…some key*
figures … …ency and
reprodu…

Reproduced Figure 6 (a)

Figure 10

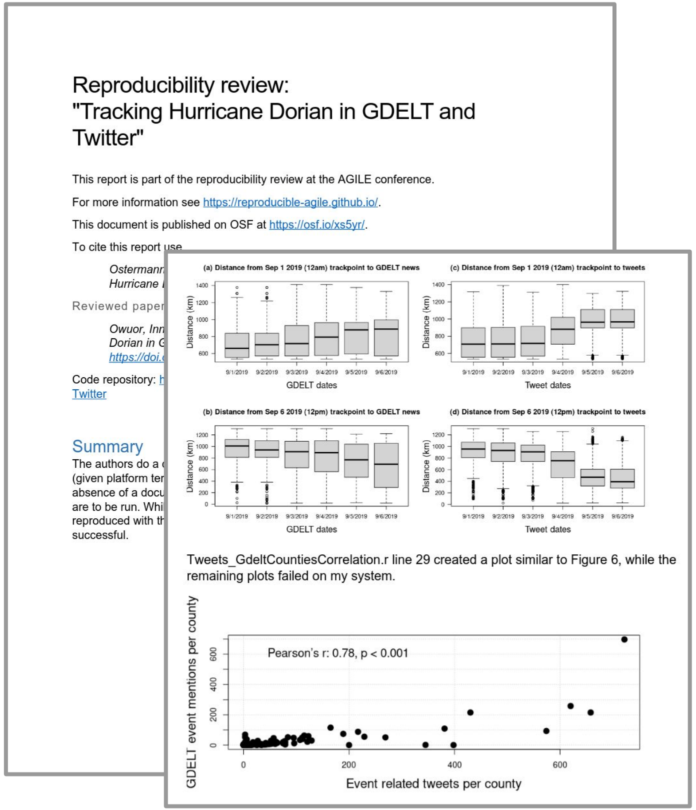
Evaluation script

```
$ cd evaluation
$ python evaluation.py
[...]
$ ll -h
total 180K
drwxrwxr-x 2 daniel daniel 4,0K Jun  5 09:05 ./
drwxrwxr-x 9 daniel daniel 4,0K Dez 23 00:47 ../
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 1.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 2.csv
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 3.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 4.csv
```

https://doi.org/10.17605/OSF.IO/ZTC7M

https://doi.org/10.17605/OSF.IO/7XRQG

Reproducibility review reports



Reproducibility review:
"Tracking Hurricane Dorian in GDELT and
Twitter"

This report is part of the reproducibility review at the AGILE conference.

For more information see https://reproducible-agile.github.io/.

This document is published on OSF at https://osf.io/xs5yr/.

To cite this report use

Ostermann
Hurricane

Reviewed paper

Owuor, Inn
Dorian in G
https://doi.

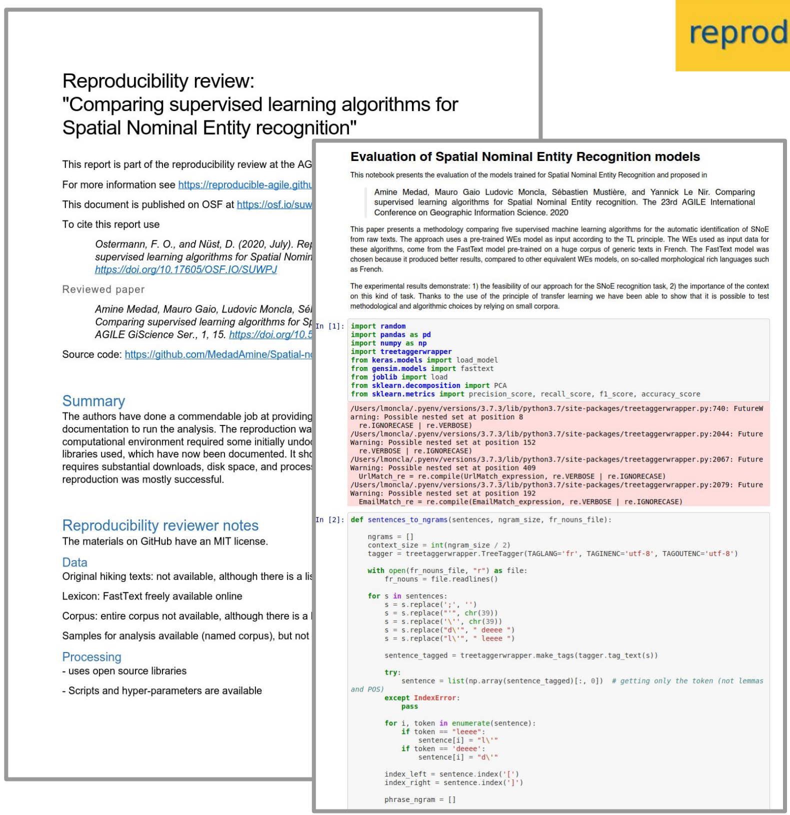
Code repository: h
Twitter

Summary

The authors do a
(given platform ter
absence of a docu
are to be run. Whi
reproduced with th
successful.

Tweets_GdeltCountiesCorrelation.r line 29 created a plot similar to Figure 6, while the remaining plots failed on my system.

Pearson's r: 0.78, p < 0.001
GDELT event mentions per county
0
200
400
600
Event related tweets per county
0
200
400
600



Reproducibility review:
"Comparing supervised learning algorithms for
Spatial Nominal Entity recognition"

This report is part of the reproducibility review at the AG

For more information see https://reproducible-agile.githu

This document is published on OSF at https://osf.io/suw

To cite this report use

Ostermann, F. O., and Nüst, D. (2020, July). Rep
supervised learning algorithms for Spatial Nomin
https://doi.org/10.17605/OSF.IO/SUWPJ

Reviewed paper

Amine Medad, Mauro Gaio, Ludovic Moncla, Sé
Comparing supervised learning algorithms for Sp
AGILE GIScience Ser., 1, 15. https://doi.org/10.5

Source code: https://github.com/MedadAmine/Spatial-n

Summary

The authors have done a commendable job at providing
documentation to run the analysis. The reproduction wa
computational environment required some initially undoc
libraries used, which have now been documented. It sho
requires substantial downloads, disk space, and process
reproduction was mostly successful.

Reproducibility reviewer notes

The materials on GitHub have an MIT license.

Data

Original hiking texts: not available, although there is a lis

Lexicon: FastText freely available online

Corpus: entire corpus not available, although there is a l

Samples for analysis available (named corpus), but not

Processing

- uses open source libraries
- Scripts and hyper-parameters are available

Evaluation of Spatial Nominal Entity Recognition models

This notebook presents the evaluation of the models trained for Spatial Nominal Entity Recognition and proposed in

> Amine Medad, Mauro Gaio Ludovic Moncla, Sébastien Mustière, and Yannick Le Nir. Comparing supervised learning algorithms for Spatial Nominal Entity recognition. The 23rd AGILE International Conference on Geographic Information Science. 2020

This paper presents a methodology comparing five supervised machine learning algorithms for the automatic identification of SNoE from raw texts. The approach uses a pre-trained WEs model as input according to the TL principle. The WEs used as input data for these algorithms, come from the FastText model pre-trained on a huge corpus of generic texts in French. The FastText model was chosen because it produced better results, compared to other equivalent WEs models, on so-called morphological rich languages such as French.

The experimental results demonstrate: 1) the feasibility of our approach for the SNoE recognition task, 2) the importance of the context on this kind of task. Thanks to the use of the principle of transfer learning we have been able to show that it is possible to test methodological and algorithmic choices by relying on small corpora.

```
In [1]: import random
        import pandas as pd
        import numpy as np
        import treetaggerwrapper
        from keras.models import load_model
        from gensim.models import fasttext
        from joblib import load
        from sklearn.decomposition import PCA
        from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

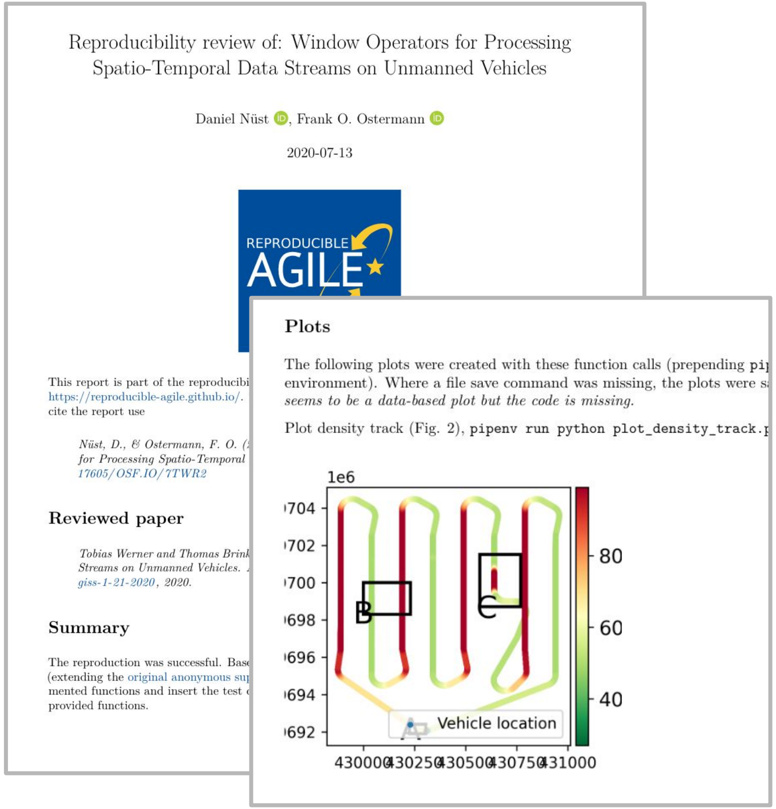
```
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:740: FutureW
arning: Possible nested set at position 8
  re.IGNORECASE | re.VERBOSE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2044: Future
Warning: Possible nested set at position 152
  re.VERBOSE | re.IGNORECASE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2067: Future
Warning: Possible nested set at position 409
  UrlMatch_re = re.compile(UrlMatch_expression, re.VERBOSE | re.IGNORECASE)
/Users/lmoncla/.pyenv/versions/3.7.3/lib/python3.7/site-packages/treetaggerwrapper.py:2079: Future
Warning: Possible nested set at position 192
  EmailMatch_re = re.compile(EmailMatch_expression, re.VERBOSE | re.IGNORECASE)
```

```
In [2]: def sentences_to_ngrams(sentences, ngram_size, fr_nouns_file):

            ngrams = []
            context_size = int(ngram_size / 2)
            tagger = treetaggerwrapper.TreeTagger(TAGLANG='fr', TAGINENC='utf-8', TAGOUTENC='utf-8')

            with open(fr_nouns_file, "r") as file:
                fr_nouns = file.readlines()

            for s in sentences:
                s = s.replace(';', '')
                s = s.replace("'", chr(39))
                s = s.replace('\'', chr(39))
                s = s.replace("d\'", " deeee ")
                s = s.replace("l\'", " leeee ")

                sentence_tagged = treetaggerwrapper.make_tags(tagger.tag_text(s))

                try:
                    sentence = list(np.array(sentence_tagged)[:, 0])  # getting only the token (not lemmas
        and POS)
                except IndexError:
                    pass

                for i, token in enumerate(sentence):
                    if token == "leeee":
                        sentence[i] = "l\'"
                    if token == 'deeee':
                        sentence[i] = "d\'"

                index_left = sentence.index('[')
                index_right = sentence.index(']')

                phrase_ngram = []
```

https://doi.org/10.17605/OSF.IO/XS5YR

https://doi.org/10.17605/OSF.IO/SUWPJ

Reproducibility review reports



Reproducibility review of: Window Operators for Processing Spatio-Temporal Data Streams on Unmanned Vehicles

Daniel Nüst, Frank O. Ostermann

2020-07-13

This report is part of the reproducibi
https://reproducible-agile.github.io/.
cite the report use

Nüst, D., & Ostermann, F. O. (
for Processing Spatio-Temporal
17605/OSF.IO/7TWR2

Reviewed paper

Tobias Werner and Thomas Brink
Streams on Unmanned Vehicles.
giss-1-21-2020, 2020.

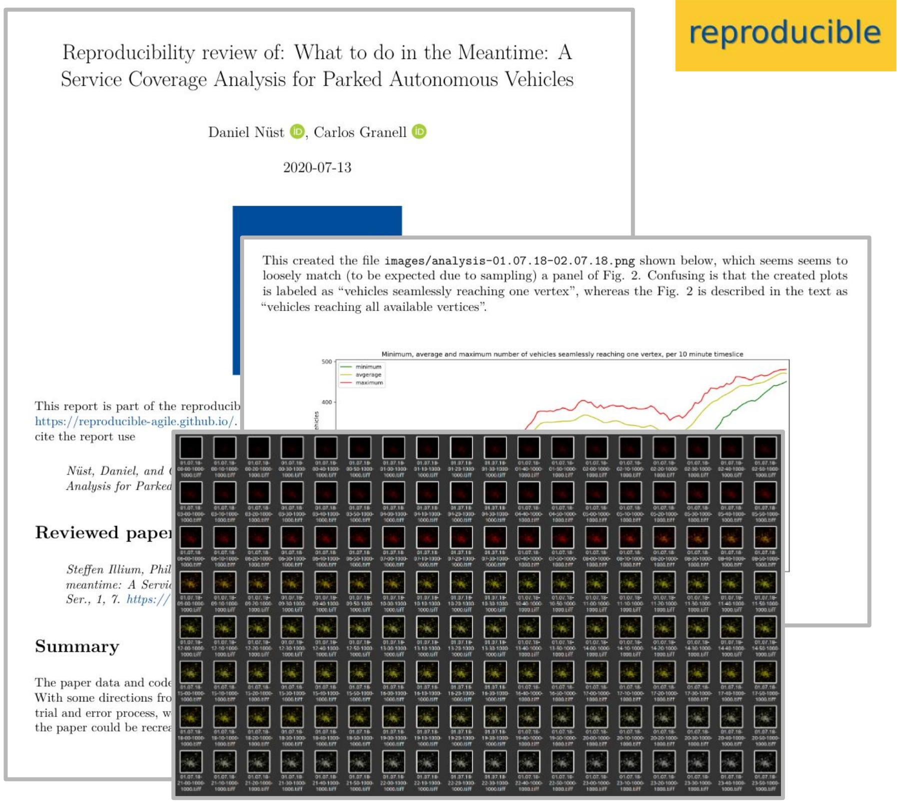
Summary

The reproduction was successful. Base
(extending the original anonymous sup
mented functions and insert the test
provided functions.

Plots

The following plots were created with these function calls (prepending pi
environment). Where a file save command was missing, the plots were s
seems to be a data-based plot but the code is missing.

Plot density track (Fig. 2), `pipenv run python plot_density_track.p`



Reproducibility review of: What to do in the Meantime: A Service Coverage Analysis for Parked Autonomous Vehicles

Daniel Nüst, Carlos Granell

2020-07-13

This created the file `images/analysis-01.07.18-02.07.18.png` shown below, which seems seems to loosely match (to be expected due to sampling) a panel of Fig. 2. Confusing is that the created plots is labeled as "vehicles seamlessly reaching one vertex", whereas the Fig. 2 is described in the text as "vehicles reaching all available vertices".

Minimum, average and maximum number of vehicles seamlessly reaching one vertex, per 10 minute timeslice
minimum
avgerage
maximum
500
400

This report is part of the reproducib
https://reproducible-agile.github.io/.
cite the report use

Nüst, Daniel, and C
Analysis for Parked

Reviewed paper

Steffen Illium, Phil
meantime: A Servic
Ser., 1, 7. https://

Summary

The paper data and code
With some directions fro
trial and error process, w
the paper could be recrea

https://doi.org/10.17605/OSF.IO/7TWR2

https://doi.org/10.17605/OSF.IO/5SVMT

The guidelines for reproducibility reviewers (WIP)

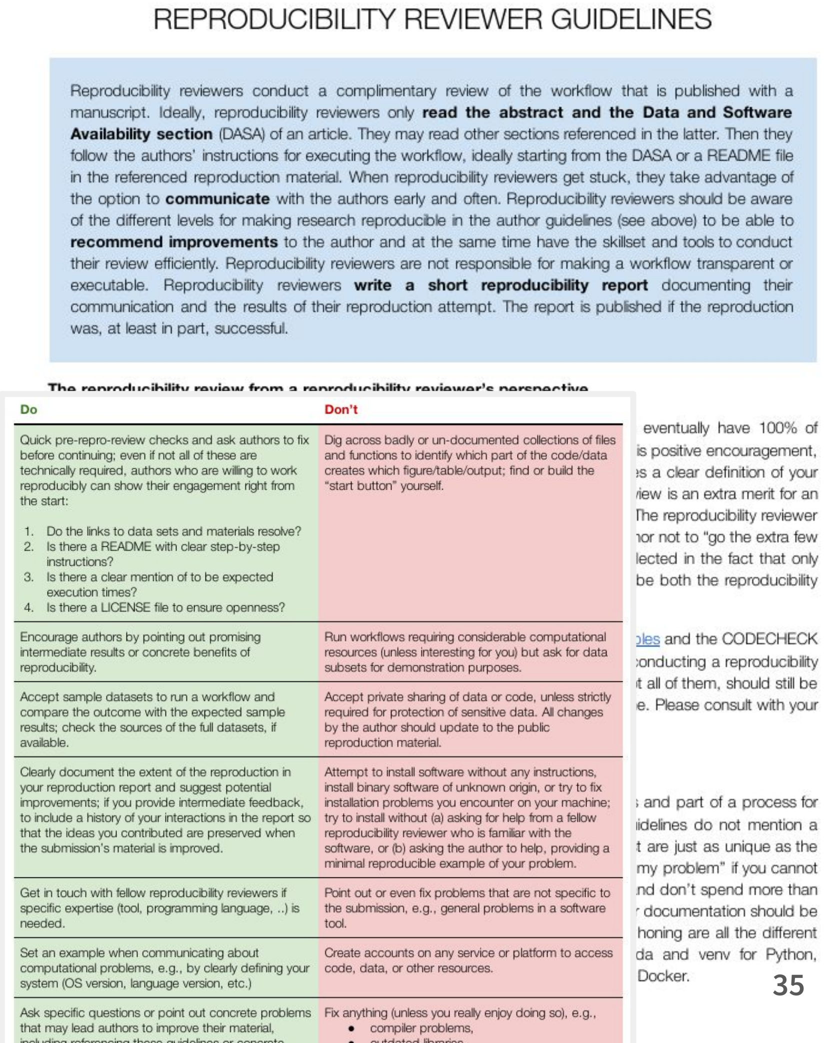
Ideal vs. realistic

Role

Skills

Do’s & dont’s

https://docs.google.com/document/d/1Kc-ToUVcrdsq6aB8Qy2J_rlluFwDniv6GHGtZuPvIEo/edit#



REPRODUCIBILITY REVIEWER GUIDELINES

Reproducibility reviewers conduct a complimentary review of the workflow that is published with a manuscript. Ideally, reproducibility reviewers only **read the abstract and the Data and Software Availability section** (DASA) of an article. They may read other sections referenced in the latter. Then they follow the authors’ instructions for executing the workflow, ideally starting from the DASA or a README file in the referenced reproduction material. When reproducibility reviewers get stuck, they take advantage of the option to **communicate** with the authors early and often. Reproducibility reviewers should be aware of the different levels for making research reproducible in the author guidelines (see above) to be able to **recommend improvements** to the author and at the same time have the skillset and tools to conduct their review efficiently. Reproducibility reviewers are not responsible for making a workflow transparent or executable. Reproducibility reviewers **write a short reproducibility report** documenting their communication and the results of their reproduction attempt. The report is published if the reproduction was, at least in part, successful.

The reproducibility review from a reproducibility reviewer’s perspective

Do	Don’t
Quick pre-repro-review checks and ask authors to fix before continuing; even if not all of these are technically required, authors who are willing to work reproducibly can show their engagement right from the start: 1. Do the links to data sets and materials resolve? 2. Is there a README with clear step-by-step instructions? 3. Is there a clear mention of to be expected execution times? 4. Is there a LICENSE file to ensure openness?	Dig across badly or un-documented collections of files and functions to identify which part of the code/data creates which figure/table/output; find or build the “start button” yourself.
Encourage authors by pointing out promising intermediate results or concrete benefits of reproducibility.	Run workflows requiring considerable computational resources (unless interesting for you) but ask for data subsets for demonstration purposes.
Accept sample datasets to run a workflow and compare the outcome with the expected sample results; check the sources of the full datasets, if available.	Accept private sharing of data or code, unless strictly required for protection of sensitive data. All changes by the author should update to the public reproduction material.
Clearly document the extent of the reproduction in your reproduction report and suggest potential improvements; if you provide intermediate feedback, to include a history of your interactions in the report so that the ideas you contributed are preserved when the submission’s material is improved.	Attempt to install software without any instructions, install binary software of unknown origin, or try to fix installation problems you encounter on your machine; try to install without (a) asking for help from a fellow reproducibility reviewer who is familiar with the software, or (b) asking the author to help, providing a minimal reproducible example of your problem.
Get in touch with fellow reproducibility reviewers if specific expertise (tool, programming language, ..) is needed.	Point out or even fix problems that are not specific to the submission, e.g., general problems in a software tool.
Set an example when communicating about computational problems, e.g., by clearly defining your system (OS version, language version, etc.)	Create accounts on any service or platform to access code, data, or other resources.
Ask specific questions or point out concrete problems that may lead authors to improve their material, including referencing these guidelines or concrete	Fix anything (unless you really enjoy doing so), e.g., • compiler problems, • outdated libraries

eventually have 100% of
is positive encouragement,
es a clear definition of your
iew is an extra merit for an
The reproducibility reviewer
hor not to "go the extra few
lected in the fact that only
be both the reproducibility

ples and the CODECHECK
conducting a reproducibility
t all of them, should still be
e. Please consult with your

and part of a process for
idelines do not mention a
t are just as unique as the
my problem" if you cannot
nd don’t spend more than
documentation should be
honing are all the different
da and venv for Python,
Docker.

🙌

How to put your community on a path towards more reproducibility in 5 ~~easy~~ hard steps

1. Build a team of enthusiasts (workshop, social events)
2. Assess the current state and raise awareness (workshop, paper)
3. Institutional support (🙏 AGILE Council 🙏 + committee chairs)
4. Positive encouragement (no reproduction != bad science)
5. Keep at it!

Next steps

Do it again in 2021 🎉

Revise guidelines 🛠️ 🇮🇹 🇫🇷 🇨🇳

Grow reproducibility reviewer team

ECRs, credit @ ORCID, skills

Continue research 🕵️

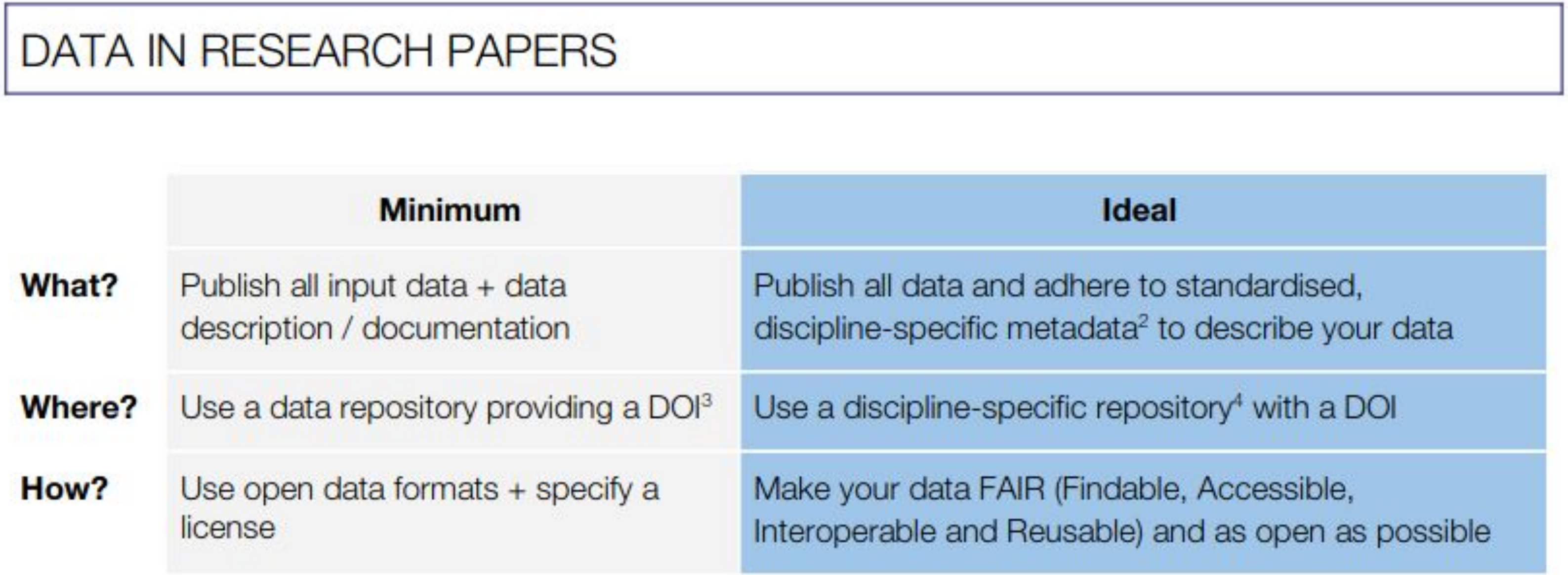
Ostermann, F., Nüst, D., Granell, C., Hofer, B., & Konkol, M. (2020). *Reproducible Research and GIScience: an evaluation using GIScience conference papers*. EarthArXiv. https://doi.org/10.31223/x5zk5v

Continue community engagement towards opening scholarship

Scope
Requirements
Acceptance condition?

Open review if tenured
Format-free submission
CRediT
Phase out when standard practice...

The guidelines for data



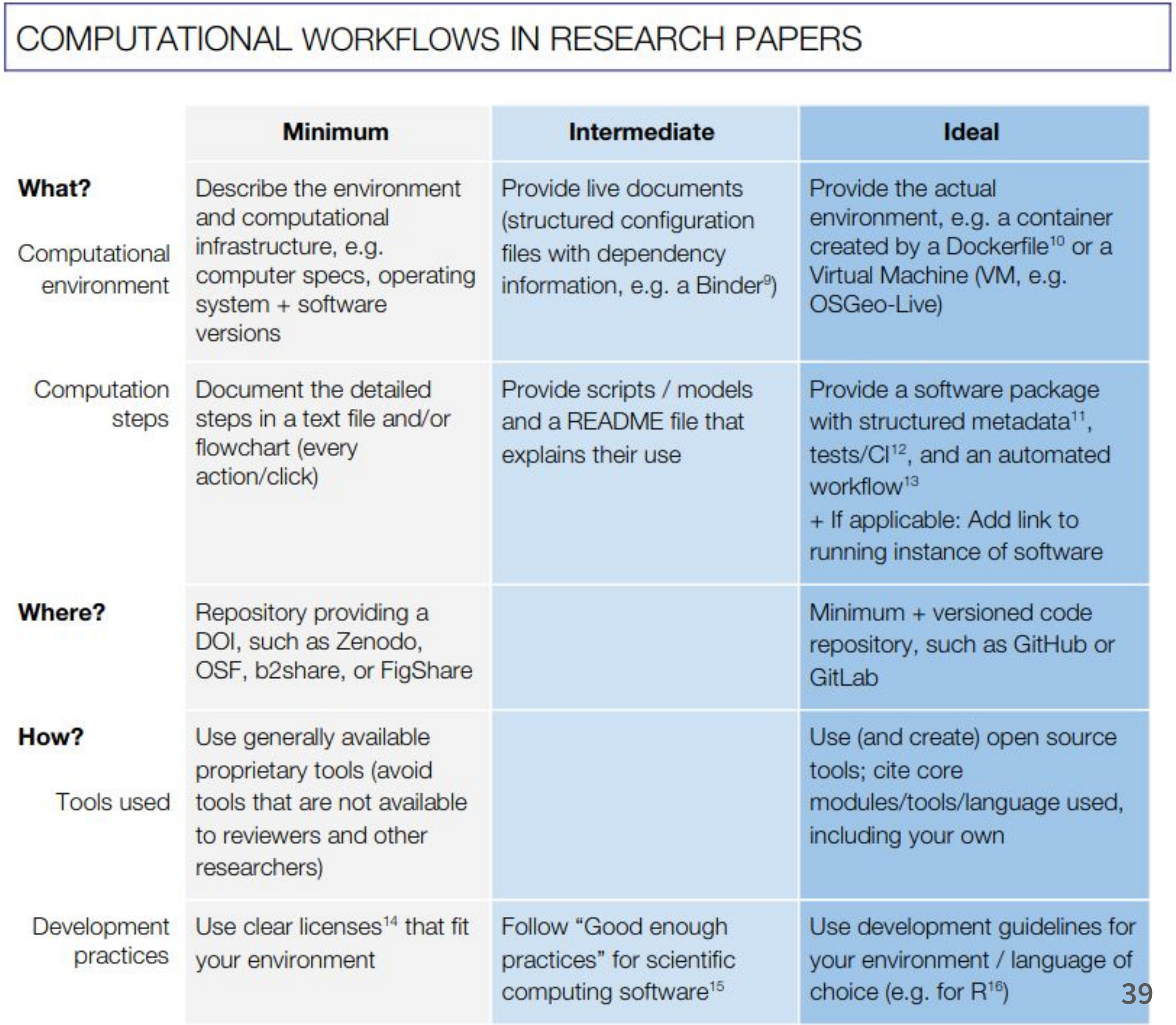
DATA IN RESEARCH PAPERS

	Minimum	Ideal
What?	Publish all input data + data description / documentation	Publish all data and adhere to standardised, discipline-specific metadata[2] to describe your data
Where?	Use a data repository providing a DOI[3]	Use a discipline-specific repository[4] with a DOI
How?	Use open data formats + specify a license	Make your data FAIR (Findable, Accessible, Interoperable and Reusable) and as open as possible

“What if…” and Examples (not shown)

https://doi.org/10.17605/OSF.IO/CB7Z8

The guidelines for workflows



COMPUTATIONAL WORKFLOWS IN RESEARCH PAPERS

	Minimum	Intermediate	Ideal
What? Computational environment	Describe the environment and computational infrastructure, e.g. computer specs, operating system + software versions	Provide live documents (structured configuration files with dependency information, e.g. a Binder[9])	Provide the actual environment, e.g. a container created by a Dockerfile[10] or a Virtual Machine (VM, e.g. OSGeo-Live)
Computation steps	Document the detailed steps in a text file and/or flowchart (every action/click)	Provide scripts / models and a README file that explains their use	Provide a software package with structured metadata[11], tests/CI[12], and an automated workflow[13] + If applicable: Add link to running instance of software
Where?	Repository providing a DOI, such as Zenodo, OSF, b2share, or FigShare		Minimum + versioned code repository, such as GitHub or GitLab
How? Tools used	Use generally available proprietary tools (avoid tools that are not available to reviewers and other researchers)		Use (and create) open source tools; cite core modules/tools/language used, including your own
Development practices	Use clear licenses[14] that fit your environment	Follow "Good enough practices" for scientific computing software[15]	Use development guidelines for your environment / language of choice (e.g. for R[16])

Examples (not shown)

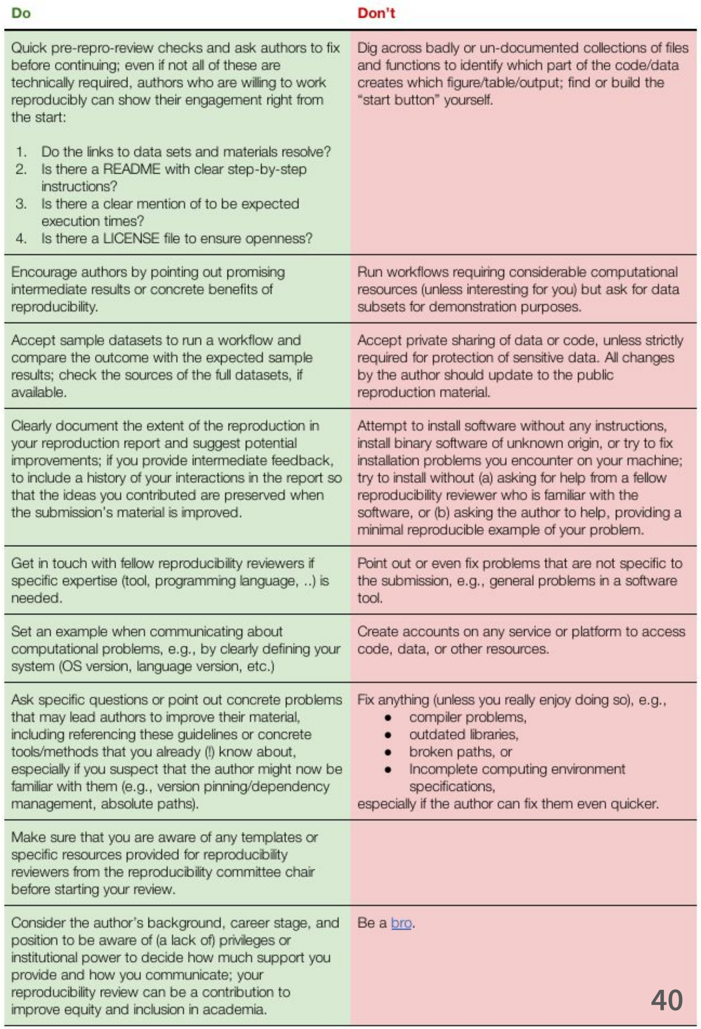
https://doi.org/10.17605/OSF.IO/CB7Z8

The guidelines for reproducibility reviewers (WIP)

Examples for “Do’s and Don’ts”:

- Do shift burden to author
- Do encourage and s
- Do *not* accept private data sharing
- Document your work in report (impact)
- Be kind (career stage, knowledge, privileges)
- No rummaging

https://docs.google.com/document/d/1Kc-ToUVcrdsq6aB8Qy2J_rlluFwDniv6GHGtZuPvlEo/edit#



Do	Don’t
Quick pre-repro-review checks and ask authors to fix before continuing; even if not all of these are technically required, authors who are willing to work reproducibly can show their engagement right from the start: 1. Do the links to data sets and materials resolve? 2. Is there a README with clear step-by-step instructions? 3. Is there a clear mention of to be expected execution times? 4. Is there a LICENSE file to ensure openness?	Dig across badly or un-documented collections of files and functions to identify which part of the code/data creates which figure/table/output; find or build the “start button” yourself.
Encourage authors by pointing out promising intermediate results or concrete benefits of reproducibility.	Run workflows requiring considerable computational resources (unless interesting for you) but ask for data subsets for demonstration purposes.
Accept sample datasets to run a workflow and compare the outcome with the expected sample results; check the sources of the full datasets, if available.	Accept private sharing of data or code, unless strictly required for protection of sensitive data. All changes by the author should update to the public reproduction material.
Clearly document the extent of the reproduction in your reproduction report and suggest potential improvements; if you provide intermediate feedback, to include a history of your interactions in the report so that the ideas you contributed are preserved when the submission’s material is improved.	Attempt to install software without any instructions, install binary software of unknown origin, or try to fix installation problems you encounter on your machine; try to install without (a) asking for help from a fellow reproducibility reviewer who is familiar with the software, or (b) asking the author to help, providing a minimal reproducible example of your problem.
Get in touch with fellow reproducibility reviewers if specific expertise (tool, programming language, ..) is needed.	Point out or even fix problems that are not specific to the submission, e.g., general problems in a software tool.
Set an example when communicating about computational problems, e.g., by clearly defining your system (OS version, language version, etc.)	Create accounts on any service or platform to access code, data, or other resources.
Ask specific questions or point out concrete problems that may lead authors to improve their material, including referencing these guidelines or concrete tools/methods that you already (!) know about, especially if you suspect that the author might now be familiar with them (e.g., version pinning/dependency management, absolute paths).	Fix anything (unless you really enjoy doing so), e.g., • compiler problems, • outdated libraries, • broken paths, or • Incomplete computing environment specifications, especially if the author can fix them even quicker.
Make sure that you are aware of any templates or specific resources provided for reproducibility reviewers from the reproducibility committee chair before starting your review.	
Consider the author’s background, career stage, and position to be aware of (a lack of) privileges or institutional power to decide how much support you provide and how you communicate; your reproducibility review can be a contribution to improve equity and inclusion in academia.	Be a bro.