

An Overview of Platforms for Reproducible Research and New Ways of Publications

Miguel Colom¹, Bertrand Kerautret²

August 20, 2018, Beijing, 2nd RRPR Workshop

¹CMLA, ENS Paris-Saclay, France

²LORIA, Université de Lorraine, France

Plan

1. Introduction

1.1 Quick overview of RR

1.2 Benefits of RR

2. Main Platforms for Reproducible Research

Galaxy, IPython, Jupyter, Code Ocean, Research Compendia, RunMyCode, DAE, IPOL

3. New Ways of Publications

3.1 IPOL Journal

3.2 ReScience journal

3.3 JOSS Journal

4. Conclusion

1. Introduction

1.1 Quick overview of RR

- RR **redefines** the result of the research. Not just a paper, but also all the **procedures** needed to **obtain the** same published results.
- Which items? The **article** itself, the **source** code, and the **data**.
- Definition of a “**reproducible scientific publication**” given by Claerbout and followed also by Buckheit and Donoho [Buckheit & Donoho 95]: *“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”*

1.2 Benefits of RR

- **Less prone to errors**, since all the methods are **clear**, **public**, and therefore **auditable**.
- **Good practices**. **Reliable** science. The descriptions **match** what the published descriptions.
- The descriptions, methodology, source code, and data is **available** to the **scientific community**
- **Comparison** of methods easier (or even possible!).
- For authors, possibility of publishing research of **high quality**.
Counterexample: the case of a copy-pasted image in a disputed biology article.
- Also for authors: increase of the **visibility** of the publications.

2. Main Platforms for Reproducible Research

2. Main Platforms for Reproducible Research

- **Galaxy** - <https://galaxyproject.org>
- **IPython** - <https://ipython.org>
- **Jupyter** - <http://jupyter.org>
- **Code Ocean** - <https://codeocean.com>
- **Research Compendia** - <http://researchcompendia.science>
- **RunMyCode** - <http://www.runmycode.org>
- **DAE** - <http://dae.cse.lehigh.edu/DAE>
- **IPOL** - <https://www.ipol.im>

2. Main Platforms for Reproducible Research: Galaxy

Description:

- Platform for **genomic research**.
- It makes available **tools** which can be used by **non-expert** users too.
- Galaxy defines a **workflow** as a **reusable templates** which contains different algorithms applied to the input data.
- In order to achieve **reproducibility** the system stores:
 - the **input dataset**,
 - the **tools** and **algorithms** which were applied to the data along the **chain**,
 - the **parameters**,
 - the **output** result.

<https://galaxyproject.org>

2. Main Platforms for Reproducible Research: IPython

Description:

- **Generic tool** that can be used for **Reproducible Research**.
- **Mature** tool: created in **2001**.
- Allows creating **reproducible articles** by not only editing text in the notebook, but allowing **code execution** and creating figures *in situ*.
- Follows closely the definition of a “**reproducible scientific publication**” of **Claerbout**, Buckheit, and Donoho [**Buckheit & Donoho 95**].

<https://ipython.org>

2. Main Platforms for Reproducible Research: Jupyter

Description:

- Spin-off of IPython in 2014.
- Main goal: separate the Python language used in IPython from all the other functionalities needed to run the notebooks (for example, the notebook format, the web framework, or the message protocols).
- Languages: execution kernels in Jupyter.
- Nowadays it supports more than 40 languages that can be used as kernels.

<http://jupyter.org>

2. Main Platforms for Reproducible Research: Code Ocean

Description:

- Stated in 2014 as part *Runway Startup Postdoc Program* at the Jacobs Technion Cornell Institute.
- Sponsored by IEEE.
- Defined by themselves as a *computational reproducibility platform*.
- Not a journal itself. It only runs code, but not publishes articles.
- Assigns a DOI to each source code.
- Several languages accepted: Python, R, Julia, Matlab, Octave, C++, Fortran, Perl, Java
- They claim *to view and download for everyone for free*. The free plan is limited and the other are paid.
- Plans based on CPU time and storage usage. For example, the *researcher* plan allows 1h CPU and 5GB of storage per month.
- No statistics on usage. Seems low by website inspection.

2. Main Platforms for Reproducible Research: Research Compendia

Description:

- A **dissemination platform**, not a journal.
- **Reproducibility** in **computational research**.
- **Share** and **archive** the **data**, **codes**, **documentation**, **parameters**, and **environmental settings**.
- **Free**. **Non-profit** organization.
- No DOI for the moment. Future plan.
- **Supported languages**: R, MatLab, Python, and Cactus.
- No online demos, just **shared files**.

<http://researchcompendia.science>

2. Main Platforms for Reproducible Research: RunMyCode

Description:

- Confusing: two different services with the same name!
`runmycode.online`, `www.runmycode.org`.
- Both **dissemination platforms**, not a journals.
- `runmycode.online`: **run online** code from Github, Gitlab, BitBucket, Go by Example, Github Gist, Gitlab Snippets, Bitbucket Snippets.
- `runmycode.online` **languages**: C/C++, Java, Nodejs, Python 2/3, Ruby, PHP, Go, Kotlin, Scala.
- `runmycode.org`: **shares source code** and **data** associated to a **publication**.
- `runmycode.org`: **no code execution**. **All languages** and **data** formats **accepted**.

<http://www.runmycode.org>

2. Main Platforms for Reproducible Research: DAE

Description: [Lamiroy & Lopresti 16]

- Platform for **D**ocument **A**nalysis and **E**xploitation.
- Allows to run document analysis algorithms and apply comparisons.

The screenshot shows the homepage of the Document Analysis and Exploitation (DAE) website. The browser address bar shows `dae.cse.lehigh.edu`. The page features a navigation menu on the left with categories like Project Description, Events and Initiatives, Analysis Services, Technical Issues, and Navigation. The main content area is titled "Document Analysis and Exploitation" and includes a "Home" section, a list of "Algorithms" (such as convert, Stanford-NER, Tesseract, ocrad, NCI-CADD segmentation, MergeImageList, NCI-CADD binarization, DICE, QGar Arc Detection, Kanungo Degradation, and ArcEval), and a "Recent blog posts" section. A sidebar on the right displays statistics: "Currently hosting 114,405 Document Images (1.3T), 9 Algorithms, 358,776 Data Items". At the bottom of the page, the URL `http://dae.cse.lehigh.edu` is displayed.

2. Main Platforms for Reproducible Research: DAE

Description: [Lamiroy & Lopresti 16]

- Platform for **D**ocument **A**nalysis and **E**xploitation.
- Allows to run document analysis algorithms and apply comparisons.
- RR framework for document analysis with image data base.

The screenshot shows the DAE website interface in a browser window. The URL is `dae.cse.lehigh.edu`. The page features a navigation menu on the left with sections like "Project Description", "Events and Initiatives", "Analysis Services", "Technical Issues", and "Navigation". The main content area is titled "Browse Data" and lists several datasets with their respective ratings and tags. The right sidebar contains "Recent blog posts" and a "Twitter" section. A top navigation bar includes "User login", "Contact us", "Copyright Alert", and "Job Offerings".

Document Analysis and Exploitation

Currently hosting
114,405 Document
Images (1.3T)
9 Algorithms
358,776 Data Items

Browse Data

Browse: [New](#) [Top Rated](#) [Popular](#) Search Names and Tags

Filter: All Datasets Root Datasets Page Images

Lehigh Notebook ★★★★★
0 TAGS: No Tags Yet

UNLV ★★★★★
2 TAGS: No Tags Yet

GREC 2011 Arc Segmentation Contest [Test Images] ★★★★★
0 TAGS: No Tags Yet

GREC 2011 Symbol Recognition Training Set ★★★★★
0 TAGS: No Tags Yet

GREC 2011 Symbol Recognition (OBSOLETE) ★★★★★

Recent blog posts

- Account Requests
- DAE is on Twitter
- Source Code Available
- DAE Public Live
- DAS 2010 Demo Poster

Twitter

2. Main Platforms for Reproducible Research: IPOL Journal

Description: [Arevalo *et al.* 16]

- A complete **peer-reviewed journal** can be considered as a platforms.
- Image Processing domain.
- Focused on **mathematical rigorness**. **Detailed** descriptions.
- **Fast** to create **new demos** for editors: **automatic system**.
- **Accepted languages**: C/C++, Python, MATLAB, Octave.
- **Free to use/submit**.
- **Next move**: **machine learning** applications. Servers with **GPU**.

<https://www.ipol.im>

3. New Ways of Publications

3. New Ways of Publications

Recent original journals

- IPOL (as presented the morning)

<https://www.ipol.im>

- ReScience

<http://rescience.github.io>

- JOSS

<https://joss.theoj.org>

3.1 IPOL Journal: Image Processing On Line

Origin: <http://www.ipol.im>

- Journal started in October 2009.
- Initiative of **Nicolas Limare** and **Jean-Michel Morel** (CMLA).
- First article published in 2010.
- Domain of **Image Processing**.



3.1 IPOL Journal: Image Processing On Line

Origin: <http://www.ipol.im>

- Journal started in October 2009.
- Initiative of **Nicolas Limare** and **Jean-Michel Morel** (CMLA).
- First article published in 2010.
- Domain of **Image Processing**.

Motivations

- Reproducible Research.
- New way to publish research results.
- Allows everybody to test the algorithms (with their own images).
- **Free online demonstration** (user-platform independant) and source code.

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing **algorithm** description, **source code**, online demonstration with experiment archives.

The screenshot shows a web browser window with the URL `ipol.im`. The page header includes the IPOL Journal logo and navigation links: HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH. The main title is "Automatic Detection of Internal Copy-Move Forgeries in Images" by Thibaud Ehret. Below the title are tabs for "article", "demo", and "archive". A green bar contains publication details: published 2018-07-25, reference to the journal issue, and a BibTeX link. The "Communicated by" and "Demo edited by" sections list Loïc Simon. The "Abstract" section describes the paper's contribution to copy-move forgery detection. The "Download" section offers a PDF manuscript (5.6M) and source code in TAR/GZ format. A "Preview" section notes that images are degraded for faster rendering.

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, **online demonstration** with experiment archives.

The screenshot shows a web browser window at `ipolcore.ipol.im`. The page title is "IPOL Journal · Image Processing On Line". The navigation menu includes "HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH". The main heading is "Automatic Detection of Internal Copy-Move Forgeries in Images". Below the heading are tabs for "Article", "Demo", and "Archive". A blue banner reads: "Please cite the reference article if you publish results obtained with this online demo." The interface has a "Select input(s)" section with an "Upload data" button and a row of five image thumbnails: a red maple tree, a marina with boats, autumn leaves, a torii gate, and a museum interior. Below this is an "Input(s)" section with a "Reset" button. The "Parameters" section includes a checkbox for "Test for flipped internal copies" (unchecked) and a slider for "Number of iterations of patchmatch" (set to 8, with a maximum of 16) and "Minimum distance".

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.

The screenshot shows a web browser window at `ipolcore.ipol.im`. The page title is "IPOL Journal · Image Processing On Line". The navigation menu includes "HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH". The main article title is "Automatic Detection of Internal Copy-Move Forgeries in Images". There are tabs for "Article", "Demo", and "Archive". A note says "Please cite the reference article if you publish results obtained with this online demo." Below that, it states "146 public experiments since 2017-03-29". A warning says "This archive is not moderated. In case you uploaded images that you don't want that appear in the archive, please contact the editor in charge. In case of copyright infringement or similar problems, please contact us to request the removal of some images. Some archived content may be deleted by the editorial board for size matters, inadequate content, user requests, or other reasons." There is a pagination bar with numbers 1 through 15, where 15 is selected. The article content shows "Experiment #17091" dated "2018-08-06 03:44:07". A table of parameters is shown:

Parameters	
flip	false
iter	8
th1	64
thd	2500
the	300
ths	1200
rdm	4
rde	6
mtd	0

Below the parameters are four image thumbnails:

- Suspect**: A grayscale image of a city skyline with a boat in the water.
- Patchmatch displacement map**: A colorful heatmap showing displacement vectors.
- Error map**: A grayscale image showing detected errors or distortions.
- Initial mask**: A black image with a few white pixels.

At the bottom, there is a fifth thumbnail showing a white mask on a black background.

Ouvrir #ipolcore.ipol.im/demo/clientApp/archive.html?id=213# sur cette page, dans un nouvel onglet

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.
- Like classic journal: ISSN, DOI, indexed by:
SCOPUS, DBLP, Scirus, Google Scholar, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek and NewJour.

3.1 IPOL Journal: short overview

Characteristics

- Journal publishing algorithm description, source code, online demonstration with **experiment archives**.
- The peer-review process includes the article, and source code.
- Open Science journal and Reproducible Research.
- Like classic journal: ISSN, DOI, indexed by:
SCOPUS, DBLP, Scirus, Google Scholar, DOAJ, SHERPA/RoMEO, Héloïse, WorldCat, CrossRef, Ulrich, Index Copernicus, PBN, JGate, VisionBib, CVonline, JournalSeek and NewJour.

Recent evolution

- New system to automatically create our own online demonstrations (see next session).
- Extended topics to **sound** and **video** processing with **3D** processing.

3.2 ReScience Journal

Philosophy (<http://rescience.github.io>)

- Context of Reproducible research [Buckheit & Donoho 95].
- Explicit **replication**: propose a **new implementation** of an existing work.
- Motivated from replication problems in computational science [Hinsen 15], [Topalidou *et al.* 15], [Hinsen 14].



3.2 ReScience Journal

Philosophy (<http://rescience.github.io>)

- Context of Reproducible research [Buckheit & Donoho 95].
- Explicit **replication**: propose a **new implementation** of an existing work.
- Motivated from replication problems in computational science [Hinsen 15], [Topalidou *et al.* 15], [Hinsen 14].

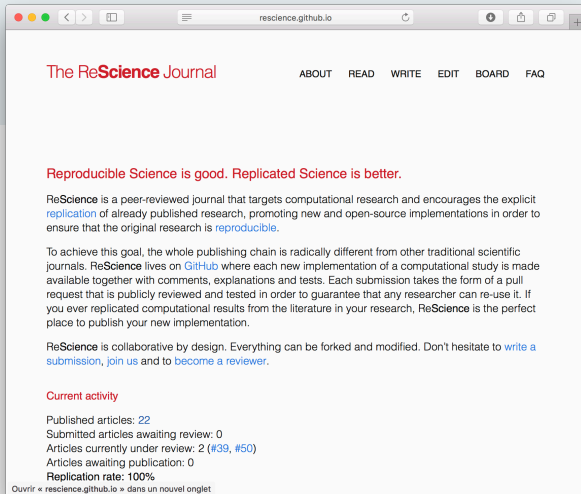
Details:

- Origin: first volume in 2015.
- Editorial Board:
 - **Editors-in-Chief**:
 - Konrad Hinsen (Molecular Biophysics - Python, C, Racket, Clojure).
 - Nicolas P. Rougier (Comp. Neuroscience, Computer Science - Python, C/C++).
 - **11 Associate Editors** with roles in: Bioinformatics; Cognitive Modelling; Computational Ecology; Computational Physics; Image processing; Ecology, High-Performance Computing; Physics; Robotics; Signal Processing

3.2 ReScience Journal: short overview

Characteristics

- Same presentation as in a “classic” journal.



The screenshot shows a web browser window displaying the homepage of the ReScience Journal. The browser's address bar shows the URL `rescience.github.io`. The page features a navigation menu with links for ABOUT, READ, WRITE, EDIT, BOARD, and FAQ. The main heading is "The ReScience Journal". Below the heading, a red banner states "Reproducible Science is good. Replicated Science is better." The text describes the journal as a peer-reviewed platform for computational research, emphasizing reproducibility and open-source implementations. It mentions that submissions are made as pull requests on GitHub, which are publicly reviewed and tested. The page also includes a section for "Current activity" with statistics: 22 published articles, 0 submitted articles awaiting review, 2 articles currently under review (with 39 and 50 in parentheses), and 0 articles awaiting publication. The replication rate is listed as 100%. At the bottom, there is a small footer: "Ouvrir « rescience.github.io » dans un nouvel onglet".

The ReScience Journal

ABOUT READ WRITE EDIT BOARD FAQ

Reproducible Science is good. Replicated Science is better.

ReScience is a peer-reviewed journal that targets computational research and encourages the explicit [replication](#) of already published research, promoting new and open-source implementations in order to ensure that the original research is [reproducible](#).

To achieve this goal, the whole publishing chain is radically different from other traditional scientific journals. ReScience lives on [GitHub](#) where each new implementation of a computational study is made available together with comments, explanations and tests. Each submission takes the form of a pull request that is publicly reviewed and tested in order to guarantee that any researcher can re-use it. If you ever replicated computational results from the literature in your research, ReScience is the perfect place to publish your new implementation.

ReScience is collaborative by design. Everything can be forked and modified. Don't hesitate to [write a submission](#), [join us](#) and to [become a reviewer](#).

Current activity

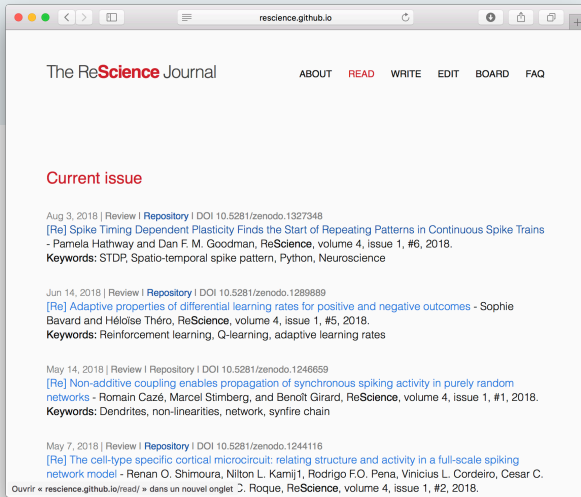
Published articles: 22
Submitted articles awaiting review: 0
Articles currently under review: 2 (#39, #50)
Articles awaiting publication: 0
Replication rate: 100%

Ouvrir « rescience.github.io » dans un nouvel onglet

3.2 ReScience Journal: short overview

Characteristics

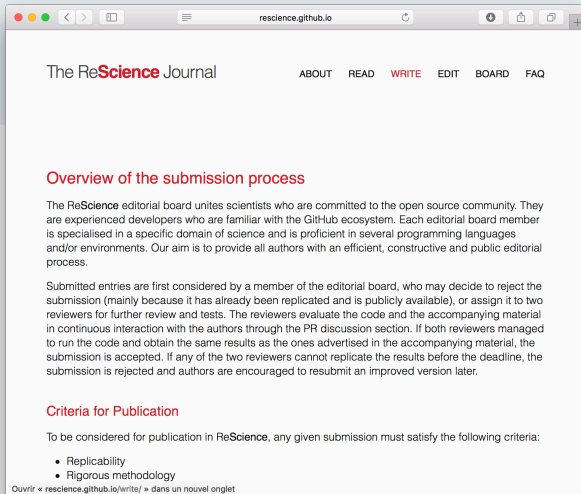
- Same presentation as in a “classic” journal.



3.2 ReScience Journal: short overview

Characteristics

- Same presentation as in a “classic” journal.



The screenshot shows a web browser window with the URL `rescience.github.io`. The page title is "The ReScience Journal" and the navigation menu includes "ABOUT", "READ", "WRITE", "EDIT", "BOARD", and "FAQ". The main content area features a red heading "Overview of the submission process" followed by a paragraph describing the editorial board and their role. Below this is another paragraph detailing the submission process, including the role of reviewers and the criteria for acceptance or rejection. A red heading "Criteria for Publication" is followed by a list of criteria: "Replicability" and "Rigorous methodology". At the bottom of the page, there is a small footer: "Ouvrir « rescience.github.io/write/ » dans un nouvel onglet".

The ReScience Journal

ABOUT READ **WRITE** EDIT BOARD FAQ

Overview of the submission process

The ReScience editorial board unites scientists who are committed to the open source community. They are experienced developers who are familiar with the GitHub ecosystem. Each editorial board member is specialised in a specific domain of science and is proficient in several programming languages and/or environments. Our aim is to provide all authors with an efficient, constructive and public editorial process.

Submitted entries are first considered by a member of the editorial board, who may decide to reject the submission (mainly because it has already been replicated and is publicly available), or assign it to two reviewers for further review and tests. The reviewers evaluate the code and the accompanying material in continuous interaction with the authors through the PR discussion section. If both reviewers managed to run the code and obtain the same results as the ones advertised in the accompanying material, the submission is accepted. If any of the two reviewers cannot replicate the results before the deadline, the submission is rejected and authors are encouraged to resubmit an improved version later.

Criteria for Publication

To be considered for publication in ReScience, any given submission must satisfy the following criteria:

- Replicability
- Rigorous methodology

Ouvrir « rescience.github.io/write/ » dans un nouvel onglet

3.2 ReScience Journal: short overview

Characteristics

- Same presentation as in a “classic” journal.

The screenshot shows a web browser window displaying a ReScience article. The browser's address bar shows the file path 'Hathway-Goodman-2018.pdf (page 1 sur 11)'. The page features the ReScience logo at the top left. The main title is '[Re] Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains'. The authors are Pamela Hathway¹ and Dan F. M. Goodman¹. The affiliation is the Department of Electrical and Electronic Engineering, Imperial College, London, UK. The article includes an editor (Nicolas P. Rougier), reviewers (Damien Drix, Julien Vitay), and dates (Received May 29, 2018; Accepted Jul 5, 2018; Published Aug 3, 2018). A highlighted box contains a reference implementation link. The introduction section begins with a discussion on neuronal communication through spike patterns.

ReScience

[Re] Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains

Pamela Hathway¹ and Dan F. M. Goodman¹

¹ Department of Electrical and Electronic Engineering, Imperial College, London, UK
p.hathway16@imperial.ac.uk

Editor
Nicolas P. Rougier

Reviewers
Damien Drix
Julien Vitay

Received May, 29, 2018
Accepted Jul, 5, 2018
Published Aug, 3, 2018

Licence [CC-BY](#)

Competing Interests:
The authors have declared that no competing interests exist.

A reference implementation of

→ Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains, Masquelier T, Guyonneau R, Thorpe SJ, PLoS ONE 3(1): e1377, 2008. <https://doi.org/10.1371/journal.pone.0001377>

Introduction

Neurons communicate through repeated, specifically timed action potential sequences (spike patterns) to convey information [3, 8]. Since neuronal activity is noisy and neurons are likely involved in a multitude of spike patterns of various lengths and extent, it can be hard to find spike patterns at first glance. The more neurons are recorded, the more difficult the task becomes due to the exponential increase of possible combinations of spikes that could make up a pattern [2]. It is unclear how neurons in the brain may extract relevant information from such input. In a 2008 paper, Masquelier and colleagues demonstrated that a single neuron with afferent synapses exhibiting spike timing dependent plasticity (STDP) is able to find the start of a

[Article repository](#)

[Code repository](#)

3.2 ReScience Journal: short overview

Characteristics

- Same presentation as in a “classic” journal.
- Journal living on *GitHub*.
- Original submission process through *Pull Request* on *GitHub*.

3.2 ReScience Journal: short overview

Characteristics

- Same presentation as in a “classic” journal.
- Journal living on *GitHub*.
- Original submission process through *Pull Request* on *GitHub*.
- Peer reviewed journal (**reviews** and **reviewer name** given in the paper).

3.2 ReScience Journal: example of publication process

Example of publication [Rougier 17]

- Already published work "Weighted Voronoi Stippler" [Secord 02]
- Code no more available on author webpage:
<https://mrl.nyu.edu/~ajsecord/stipples.html>.

3.2 ReScience Journal: example of publication process

Example of publication [Rougier 17]

- Already published work "Weighted Voronoi Stippler" [Secord 02]
- Code no more available on author webpage:
<https://mrl.nyu.edu/~ajsecord/stipples.html>.
- Authors propose a new implementation defined in another context.

3.2 ReScience Journal: example of publication process

Example of publication [Rougier 17]

- Already published work "Weighted Voronoi Stippler" [Secord 02]
- Code no more available on author webpage:
<https://mrl.nyu.edu/~ajsecord/stipples.html>.
- Authors propose a new implementation defined in another context.
- New description of the choice made to replicate the initial work.

3.2 ReScience Journal: example of publication process

Example of publication [Rougier 17]

- Already published work "Weighted Voronoi Stippler" [Secord 02]
- Code no more available on author webpage:
<https://mrl.nyu.edu/~ajsecord/stipples.html>.
- Authors propose a new implementation defined in another context.
- New description of the choice made to replicate the initial work.

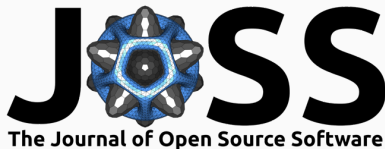
Conclusion of ReScience Journal

- Original new way of publish replication.
- Contains actually 22 published papers and 2 under review.
- Author of original work can not submit their own replication.
- Can potentially present works that were not successfully replicated ...
→ not yet present

3.3 JOSS Journal

Journal of Open Source Software (<https://joss.theoj.org>)

- Origin: founded by Arfon M. Smith in May 2016.
- Free and Open Access.
- Peer reviewing.
- Motivated by the fact that [Smith et al. 17]: *"Current publishing and citation do not acknowledge software as a first-class research output"*.



3.3 JOSS Journal

Journal of Open Source Software (<https://joss.theoj.org>)

- Origin: founded by Arfon M. Smith in May 2016.
- Free and Open Access.
- Peer reviewing.
- Motivated by the fact that [Smith et al. 17]: *"Current publishing and citation do not acknowledge software as a first-class research output"*.

Details

- **Design:** defined in the current merit system of science.
- **Aim:** can be considered as a "journal for research software packages".
- **Editorial Board:**
 - Arfon Smith (@arfon), Editor-in-Chief.
 - 19 Topic Editors: representing: Astronomy; Biodiversity Informatics; Bioinformatics; Computational Science; Data Science; Engineering, Computational Combustion; Computational Social Science; Fluid Dynamics; Energy Engineering; Geophysics; Geoscience; High Perf. Computing; Image; Information Sciences; Machine Learning; Neuroimaging; Nuclear Engineering; Open Science; Psychology; Semantic Web; Social Sciences; Software Deployment; Reproducible Research.

3.3 JOSS Journal: motivations

Motivation of JOSS Editor in chief [Smith *et al.* 17]:

- Software more and more present in numerous disciplines:
⇒ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick *et al.* 14].
- Software lack of scholarship support: no ecosystem of publication, citation, acknowledge.
- JOSS is the contribution to offer modern computational research results.

3.3 JOSS Journal: motivations

Motivation of JOSS Editor in chief [Smith *et al.* 17]:

- Software more and more present in numerous disciplines:
⇒ from a 2014 survey 90% mention to use software and 70% indicates that they were obligatory [Hettrick *et al.* 14].
- Software lack of scholarship support: no ecosystem of publication, citation, acknowledge.
- JOSS is the contribution to offer modern computational research results.

As mentioned by Buckheit and Dohono [Buckheit & Donoho 95]:

"An article about computational science in a scientific publication is not the scholarship itself it is merely advertising of the scholarship"

3.3 JOSS Journal: characteristics

Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).

The screenshot shows the JOSS journal website. The main header includes the JOSS logo and the text 'The Journal of Open Source Software'. Navigation links for 'Submit', 'Papers', 'About', and 'Sign in' are visible. The article title is 'jstor: Import and Analyse Data from Scientific Texts'. The author is Thomas Klebel, from the Department of Sociology, University of Graz. The article is dated 07 August 2018. The DOI is 10.21105/joss.00883. The article is licensed under a Creative Commons Attribution 4.0 International License. The abstract discusses the interest in text-as-data and the development of the jstor package for R.

Article details

- View review
- Download paper
- Software repository
- Software archive

Submitted: 07 August 2018
Accepted: 08 August 2018

Cite as:
Klebel, (2018). jstor: Import and Analyse Data from Scientific Texts. Journal of Open Source Software, 3(28), 883.
<https://doi.org/10.21105/joss.00883>

Status badge
JOSS 10.21105/joss.00883

License
Authors of JOSS papers retain copyright.

Software

- Review IT
- Repository IT
- Archive IT

Submitted: 07 August 2018
Published: 08 August 2018

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

Summary

The interest in text-as-data has seen a sharp increase in the past few years, mostly due to the advent of methods for automated text analysis. At the same time, researchers within the field of scientometrics have analysed citations and other aspects of the scholarly literature with great sophistication. The archival content of JSTOR offers a rich and diverse set of primary sources like research articles or book chapters for both approaches. Data for Research (DR) by JSTOR gives all researchers, regardless of whether they have access to JSTOR or not, the opportunity to analyse metadata, n-grams and, upon special request, full-text materials about all available articles and books from JSTOR. The package jstor (Klebel, 2018) helps in analyzing these datasets by enabling researchers to easily import the metadata to R (R Core Team, 2018), a task, for which no other integrated solution exists to date.

The metadata from DR can either be analysed on their own or be used in conjunction with n-grams or full-text data. Commonly, metadata from DR include information on the article's authors, their title, journal, date of publishing, and quite frequently all footnotes and references. All this information can be of interest for specific research questions. For the analysis of n-grams or full-texts, the metadata imported with jstor allow the researchers to filter articles based on specific journals, the dates of publication, the authors, keywords in titles and other aspects.

3.3 JOSS Journal: characteristics

Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.

3.3 JOSS Journal: characteristics

Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.
- Same characteristics than other journals: ISSN, Crossref DOI.
- Code Review: **direct visibility**
⇒ collaboration with classic development tools (based on *GitHub*).

3.3 JOSS Journal: characteristics

Specific form:

- Form: **voluntary short**: short abstract length (author names, list of key references, a link to software repository and a short description of the content).
- **Not allowed**: **API** or **novel research** descriptions.
- Same characteristics than other journals: ISSN, Crossref DOI.
- Code Review: **direct visibility**
⇒ collaboration with classic development tools (based on *GitHub*).

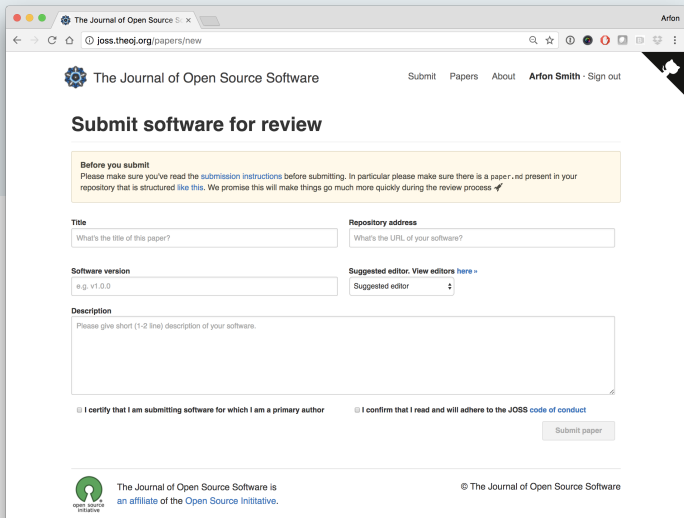
Content requested:

- Software need to be **open source**.
- **Research** application.
- Submitter needs to be **main software contributor**.
- Significant new contribution.
- **Feature-complete** (not partial).

3.3 JOSS Journal: publication process and cost (1)

Review Process

- Submission page: simple web application.



The screenshot shows a web browser window with the URL `joss.theoj.org/papers/new`. The page header includes the JOSS logo and name, and navigation links for 'Submit', 'Papers', 'About', and 'Arfon Smith · Sign out'. The main heading is 'Submit software for review'. A yellow box contains instructions: 'Before you submit Please make sure you've read the [submission instructions](#) before submitting. In particular please make sure there is a `PAPER.md` present in your repository that is structured [like this](#). We promise this will make things go much more quickly during the review process ✓'. Below this are four input fields: 'Title' (placeholder: 'What's the title of this paper?'), 'Repository address' (placeholder: 'What's the URL of your software?'), 'Software version' (placeholder: 'e.g. v1.0.0'), and 'Suggested editor' (with a dropdown arrow and a link to 'View editors here'). A 'Description' section has a text area with the placeholder 'Please give short (1-2 line) description of your software.'. At the bottom, there are two checkboxes: 'I certify that I am submitting software for which I am a primary author' and 'I confirm that I read and will adhere to the JOSS code of conduct'. A 'Submit paper' button is located to the right of these checkboxes. The footer contains the JOSS logo and text: 'The Journal of Open Source Software is an affiliate of the Open Source Initiative.' and '© The Journal of Open Source Software'.

3.3 JOSS Journal: publication process and cost (1)

Review Process

- Submission page: simple web application.
- Each submission generate an associated *GitHub* issue.

<https://github.com/openjournals/joss-reviews>

The screenshot shows the GitHub interface for the repository `openjournals/joss-reviews`. The 'Issues' tab is selected, showing 90 issues. A filter is applied: `is:issue is:open`. The list of issues includes:

- [REVIEW]: medical image visualization library in python `review` #887 opened 15 hours ago by whedon
- [PRE REVIEW]: hoggorm: a python library for explorative multivariate statistics `Python` `TeX` `pre-review` #886 opened 21 hours ago by whedon
- [PRE REVIEW]: idpflex: Analysis of Intrinsically Disordered Proteins by Comparing Simulations to Small Angle Scattering Experiments `Jupyter Notebook` `Makefile` `Python` `pre-review` #885 opened 2 days ago by whedon
- [PRE REVIEW]: compboost: Modular Framework for Component-Wise Boosting `C++` `R` `Rebol` `pre-review` #894 opened 2 days ago by whedon
- [REVIEW]: Estimation statistics from multi-state models using simulation with `multistateutil`

3.3 JOSS Journal: publication process and cost (1)

Review Process

- Submission page: simple web application.
- Each submission generate an associated *GitHub* issue.
<https://github.com/openjournals/joss-reviews>
- Review interface: *joss-review* *GitHub* repository

openjournals / joss-reviews

[REVIEW]: medical image visualization library in python #897

Open whedon opened this issue 16 hours ago · 3 comments

whedon commented 16 hours ago

Submitting author: @raamana (Pradeep Reddy Raamana)
 Repository: <https://github.com/raamana/mivis>
 Version: 0.3
 Editor: @sankam
 Reviewer: @mlykael
 Archive: Pending

Status

JOSS Under Review

Status badge code:

```
HTML: <a href="https://joss.theo.org/papers/86f56834fa3b4c812f96dcb2787b69?
Markdown: [!status|https://joss.theo.org/papers/86f56834fa3b4c812f96dcb2787b69/status:under-review]
```

Reviewers and authors:

Please avoid lengthy details of difficulties in the review thread. Instead, please create a new issue in the target repository and link to those issues (especially acceptance-blockers) in the review thread below. (For completists: if the target issue tracker is also on GitHub, linking the review thread in the issue or vice versa will create corresponding breadcrumb trails in the link target.)

Assignees: broken

Labels: Review

Projects: None yet

Milestones: No milestones

Notifications: Subscribe

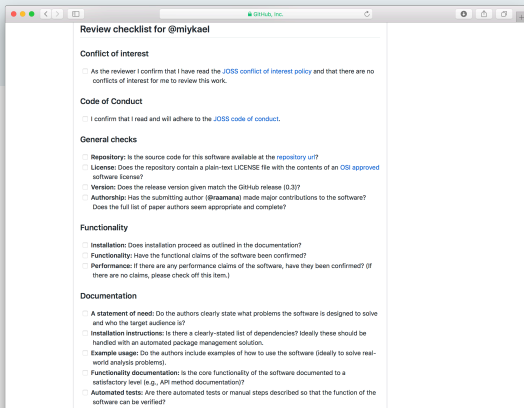
You're not receiving notifications from this thread.

2 participants

3.3 JOSS Journal: publication process and cost (1)

Review Process

- Submission page: simple web application.
- Each submission generate an associated *GitHub* issue.
<https://github.com/openjournals/joss-reviews>
- Review interface: `joss-review` *GitHub* repository



Review checklist for @miykael

Conflict of Interest

- As the reviewer I confirm that I have read the *JOSS conflict of interest policy* and that there are no conflicts of interest for me to review this work.

Code of Conduct

- I confirm that I read and will adhere to the *JOSS code of conduct*.

General checks

- Repository:** Is the source code for this software available at the *repository url*?
- License:** Does the repository contain a plain-text LICENSE file with the contents of an *OSI approved software license*?
- Version:** Does the release version given match the *GitHub release (0.3)?*
- Authorship:** Has the submitting author (*@raamana*) made major contributions to the software? Does the full list of paper authors seem appropriate and complete?

Functionality

- Installation:** Does installation proceed as outlined in the documentation?
- Functionality:** Have the functional claims of the software been confirmed?
- Performance:** If there are any performance claims of the software, have they been confirmed? (If there are no claims, please check off this item.)

Documentation

- A statement of need:** Do the authors clearly state what problems the software is designed to solve and who the target audience is?
- Installation instructions:** Is there a clearly-stated list of dependencies? Ideally these should be handled with an automated package management solution.
- Example usage:** Do the authors include examples of how to use the software (ideally to solve real-world analysis problems).
- Functionality (documentation):** Is the core functionality of the software documented to a satisfactory level (e.g., API method documentation)?
- Automated tests:** Are there automated tests or manual steps described so that the function of the software can be verified?

3.3 JOSS Journal: publication process and cost (1)

Review Process

- Submission page: simple web application.
- Each submission generate an associated *GitHub* issue.
<https://github.com/openjournals/joss-reviews>
- Review interface: `joss-review` *GitHub* repository
- Submission handled by a collection tools from a RubyGem library: Whedon.
<https://github.com/openjournals/whedon>

3.3 JOSS Journal: publication process and cost (2)

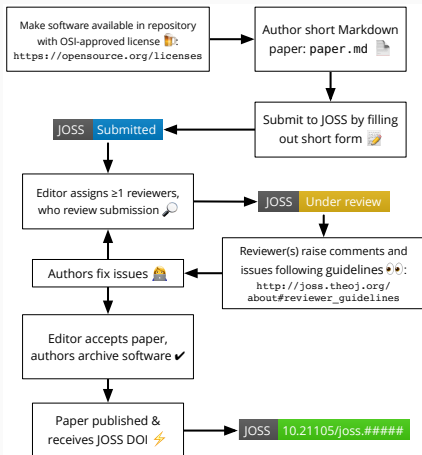
Main review steps

- Mainly handled from issue labels.
- Example of review flow (extracted from [Smith *et al.* 17]).

3.3 JOSS Journal: publication process and cost (2)

Main review steps

- Mainly handled from issue labels.
- Example of review flow (extracted from [Smith *et al.* 17]).



3.3 JOSS Journal: publication process and cost (2)

Main review steps

- Mainly handled from issue labels.
- Example of review flow (extracted from [Smith *et al.* 17]).

Cost [Smith *et al.* 17]

- Minimum cost from volunteer editors and reviewers;
- Around 6\$ per papers (with a base of 100 paper/year)
→ cross ref membership 275\$ + crossref DOI 1\$/paper + 19\$/month.

3.3 JOSS Journal: publication process and cost (2)

Main review steps

- Mainly handled from issue labels.
- Example of review flow (extracted from [Smith *et al.* 17]).

Cost [Smith *et al.* 17]

- Minimum cost from volunteer editors and reviewers;
- Around 6\$ per papers (with a base of 100 paper/year)
→ cross ref membership 275\$ + crossref DOI 1\$/paper + 19\$/month.

Comparisons with other journal

- Journal of Open Research Software (openresearchsoftware.metajnl.com)
- SoftwareX (journals.elsevier.com/softwarex/)
 - both journals review papers and software.
 - anonymous reviews in the contrary to the contrary public review and public authors/reviewers interaction in JOSS.

3.3 JOSS Journal: one year overview and future [Smith *et al.* 17]

Synthetic review of first year anniversary

- 110 published articles for the first year (**actually 350**).
- 45.6 days between submission and publication.
- 1.11 reviewers (total of reviewer count 93).

3.3 JOSS Journal: one year overview and future [Smith *et al.* 17]

Synthetic review of first year anniversary

- 110 published articles for the first year (**actually 350**).
- 45.6 days between submission and publication.
- 1.11 reviewers (total of reviewer count 93).

Future plan and open orientations

- Complete the `Whedon RubyGem` library for an automatic final paper publication.
- Question about what to do to handle version number and publication ?
→ Select the MAJOR version from a normalized MAJOR.MINOR.PATCH ?
- JOSS is now under *NumFOCUS* (charity supporting "world-class, innovative, open source scientific computing").
- Make more adverts in various communities like in Pattern Recognition.

4. Conclusion

4. Conclusion

Reproducible Research in Science

- Essential for credible science.
- In particular in Computational Science.
- Motivated by the credibility crisis pointed out by Donoho.
- Allows to follow the scientific method: claims can be verified (and eventually disproved).

4. Conclusion

Reproducible Research in Science

- Essential for credible science.
- In particular in Computational Science.
- Motivated by the credibility crisis pointed out by Donoho.
- Allows to follow the scientific method: claims can be verified (and eventually disproved).

Evolution

- Development of several platforms and journals.
- New tools allows comparisons and establish the real state of the art.
- New metric of assessing the impact of the research (instead classic citation indices).
- Increasing interest in this area.

References

Thank you for your attention!

感谢您的关注!



[Smith *et al.* 17] M Smith, A., E Niemeyer, K., Katz, D.S., A Barba, L., Githinji, G., Gymrek, M., Huff, K., Madan, C., Cabunoc Mayes, A., Moerman, K., Prins, P., Ram, K., Rokem, A., K Teal, T., Valls Guimera, R., T Vanderplas, J.:

Journal of open source software (joss): Design and first-year review.
4 (2017)



[Hettrick *et al.* 14] Hettrick, S., Antonioletti, M., Carr, L., Chue Hong, N., Crouch, S., De Roure, D., Emsley, I., Goble, C., Hay, A., Inupakutika, D., Jackson, M., Nenadic, A., Parkinson, T., Parsons, M.I., Pawlik, A., Peru, G., Proeme, A., Robinson, J., Sufi, S.:

UK Research Software Survey 2014.
(2014)



[Buckheit & Donoho 95] Buckheit, J.B., Donoho, D.L.

In: WaveLab and Reproducible Research. Springer New York, New York, NY (1995) 55–81

References (2)



[Hinsen 2015] Hinsen, K.:

Writing Software Specifications.

Computing in Science & Engineering (2015)



[Topalidou *et al.* 15] Topalidou, M., Leblois, A., Boraus, T., Rougier, N.:

A long journey into reproducible computational neuroscience.

Frontiers in Computational Neuroscience (2015)



[Hinsen 14] Hinsen, K.:

Computational science shifting the focus tools to models.

F1000Research (2014)



[Rougier 17] Rougier, P, N.:

Weighted Voronoi Stippling.

ReScience 3 (2017) 1–8

References (3)



[Secord 02] Secord, A.:

Weighted Voronoi stippling.

In: Proceedings of the second international symposium on Non-photorealistic animation and rendering, ACM Press (2002) 37–43



[Lamiroy & Lopresti 16] Lamiroy, B., Lopresti, D.P.:

The DAE platform: A framework for reproducible research in document image analysis.

In: RRPR@ICPR. Volume 10214 of Lecture Notes in Computer Science. (2016) 17–29



[Arevalo *et al.* 16] Arévalo, M., Escobar, C., Monasse, P., Monzón, N., Colom, M.:

The IPOL demo system: A scalable architecture of microservices for reproducible research.

In: RRPR@ICPR. Volume 10214 of Lecture Notes in Computer Science. (2016) 3–16