netherlands eScience center

Open science and software best practices

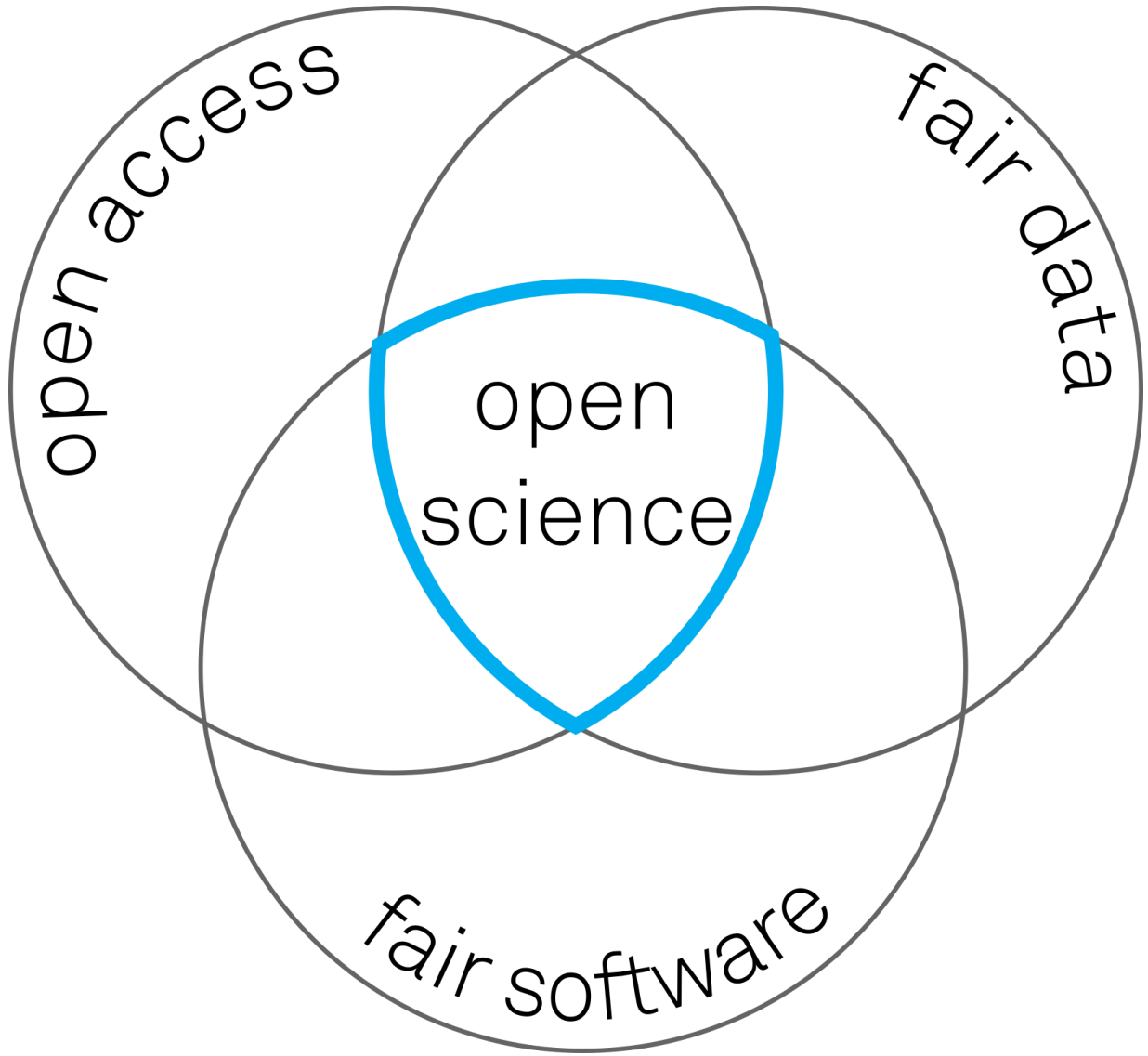
Dafne van Kuppevelt & Faruk Diblen

Photography: Elodie Burrillon

Netherlands eScience Center

Enabling digitally enhanced research through efficient utilization of data, software and e-infrastructure

Software's role in open science

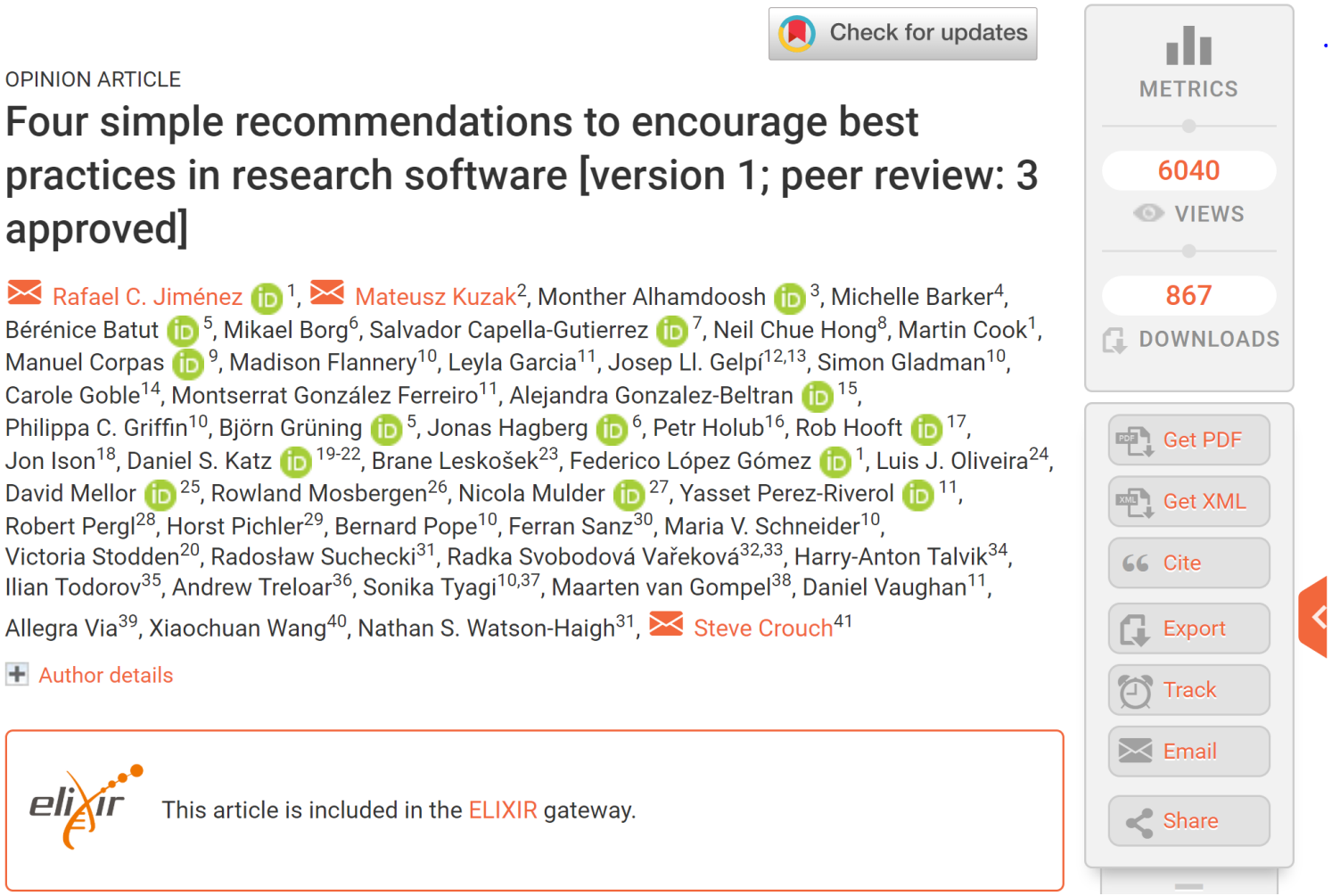


- 4 recommendations to encourage best practices in research software

- Practical guidelines

- Do you develop software?
- Do you develop on your own, or together with others?
- How do you share your software?
- What are the + and – of making code and data ‘open’?

4 Recommendations



Check for updates

OPINION ARTICLE

Four simple recommendations to encourage best practices in research software [version 1; peer review: 3 approved]

Rafael C. Jiménez [1], Mateusz Kuzak[2], Monther Alhamdoosh [3], Michelle Barker[4], Bérénice Batut [5], Mikael Borg[6], Salvador Capella-Gutierrez [7], Neil Chue Hong[8], Martin Cook[1], Manuel Corpas [9], Madison Flannery[10], Leyla Garcia[11], Josep Ll. Gelpí[12,13], Simon Gladman[10], Carole Goble[14], Montserrat González Ferreiro[11], Alejandra Gonzalez-Beltran [15], Philippa C. Griffin[10], Björn Grüning [5], Jonas Hagberg [6], Petr Holub[16], Rob Hooft [17], Jon Ison[18], Daniel S. Katz [19-22], Brane Leskošek[23], Federico López Gómez [1], Luis J. Oliveira[24], David Mellor [25], Rowland Mosbergen[26], Nicola Mulder [27], Yasset Perez-Riverol [11], Robert Pergl[28], Horst Pichler[29], Bernard Pope[10], Ferran Sanz[30], Maria V. Schneider[10], Victoria Stodden[20], Radosław Suchecki[31], Radka Svobodová Vařeková[32,33], Harry-Anton Talvik[34], Ilian Todorov[35], Andrew Treloar[36], Sonika Tyagi[10,37], Maarten van Gompel[38], Daniel Vaughan[11], Allegra Via[39], Xiaochuan Wang[40], Nathan S. Watson-Haigh[31], Steve Crouch[41]

Author details

This article is included in the ELIXIR gateway.

METRICS

6040 VIEWS

867 DOWNLOADS

Get PDF

Get XML

Cite

Export

Track

Email

Share

https://softdev4research.github.io/recommendations/

https://softdev4research.github.io/4OSS-lesson/

1. Develop publicly accessible open source code from day one

GitHub

Bitbucket

GitLab

2. Make software easy to discover by providing software metadata via a popular community registry

What is useful metadata?

3. Adopt a license and comply with the licence of third-party dependencies

Your work is automatically protected by copyright.
No license = no reuse by others!

Source code licenses: Apache 2, MIT, BSD, GPL, …

Data licenses: CC0, CC-BY-4, ODbL, …

https://choosealicense.com/

https://blog.esciencecenter.nl/a-license-to-science-cd8030a4a145

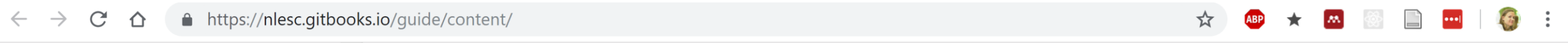
4. Have a clear and transparent contribution, governance and communication processes

How can people contribute to your project ?

How can they contact you ?

How can people ask for new features?

Can anyone push changes to your repo?



Type to search

Netherlands eScience Center Guide

Introduction
Best practices
Checklist
Version Control
Code Quality
Code Review
Licensing
Communication
Testing
Releases
Documentation
Standards
Language Guides

EDIT

Guide

This is a guide to software development and projects at the Netherlands eScience Center. It both serves as a source of information for exactly how we work at the eScience Center, and as a basis for discussions and reaching consensus on this topic.

This Guide is a work in progress

The source of this book can be found on GitHub: https://www.github.com/nlesc/guide

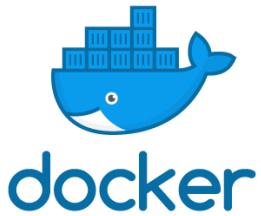
https://nlesc.gitbooks.io/guide/content/

- User documentation (README, tutorial…)
- Code documentation:
  - Python: docstring, Sphinx
  - R: roxygen

Use a consistent coding style:

- Python: pep-8
- Install *linters* for your editor (R: lintr)

- Make your code installable
  - Python: setup.py
- On GitHub: version tagging
  - Semantic versioning
- Create DOI: e.g. Zenodo
- Add Citation file
- Distribute your software:
  - Python: PyPi, Conda
  - R: CRAN
  - Generic: Docker



https://guide.esciencecenter.nl/best_practices/releases.html

- Write (unit) tests
  - Python: pytest / nose
  - R: testthat
- Continuous Integration: Travis
  - Automatically runs tests after new commit
  - Different operating systems / versions
  - Can also check code style, test coverage, generate documentation…

.travis.yml file:

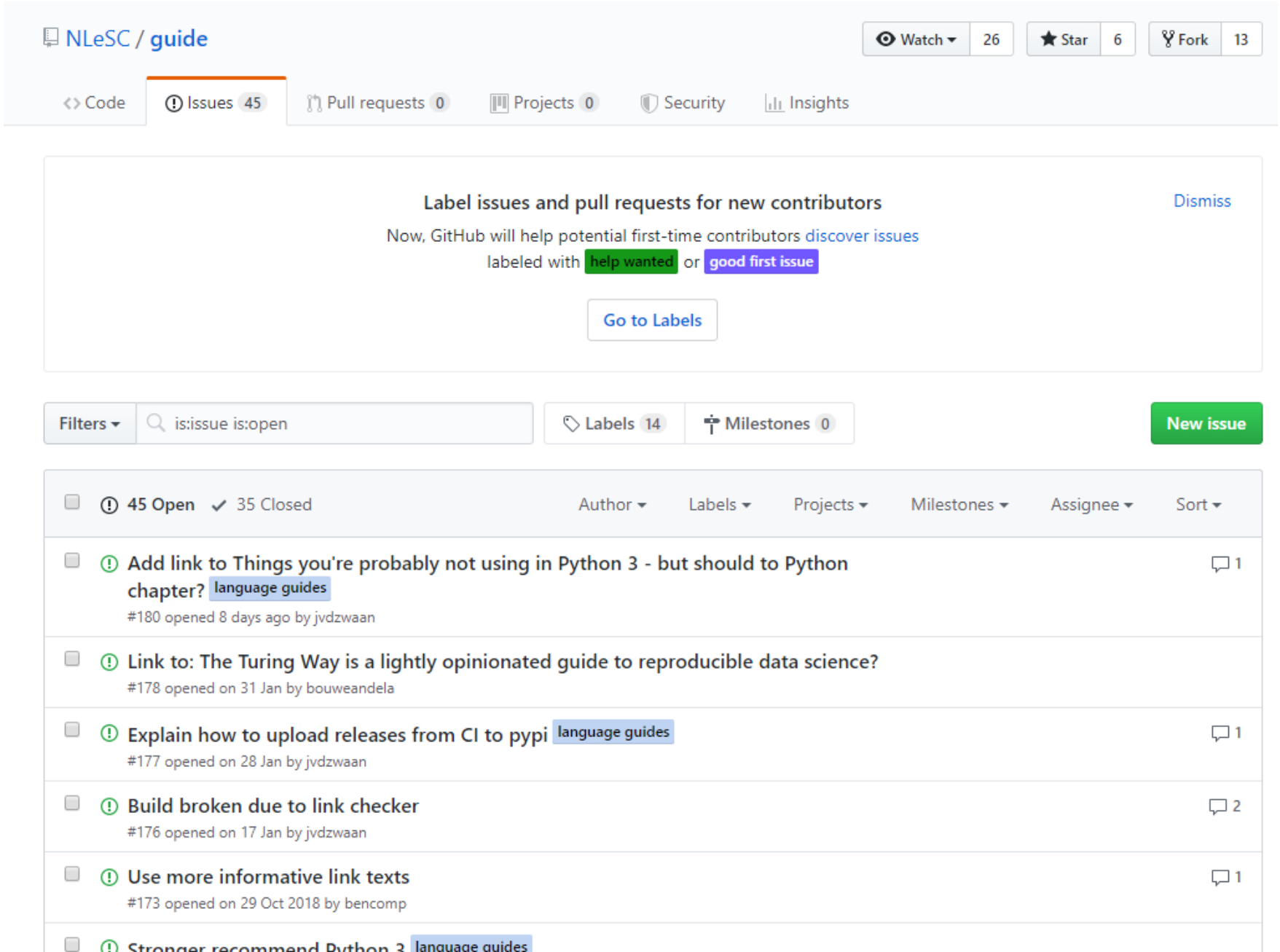
```
# Config file for automatic testing at travis-ci.org
language: python
python:
  - "3.4"
  - "3.5"
  - "3.6"

# command to install dependencies
install:
  - python setup.py install

# command to run tests
script: pytest
```

Set up empty (Python) project:

- license,
- documentation,
- Readme
- Installation
- etc…

https://github.com/NLeSC/python-template

Anything missing? Create issue



Let’s stay in touch

+31 (0)20 460 4770

info@esciencecenter.nl

www.esciencecenter.nl

blog.esciencecenter.nl

eScienceCenter

eScienceCenter

linkd.in/1j2uS8S

netherlands eScience center