Tracking FOSS contributions

Jesus M. Gonzalez-Barahona

Universidad Rey Juan Carlos
https://floss.social/@jgbarah
https://jgbarah.github.io/presentations

Workshop: "Open science, a landscape under
construction with a horizon of possibilities"
CIEM, Castro Urdiales, Spain, November 11-13th 2022

The plan

1 FOSS and open science
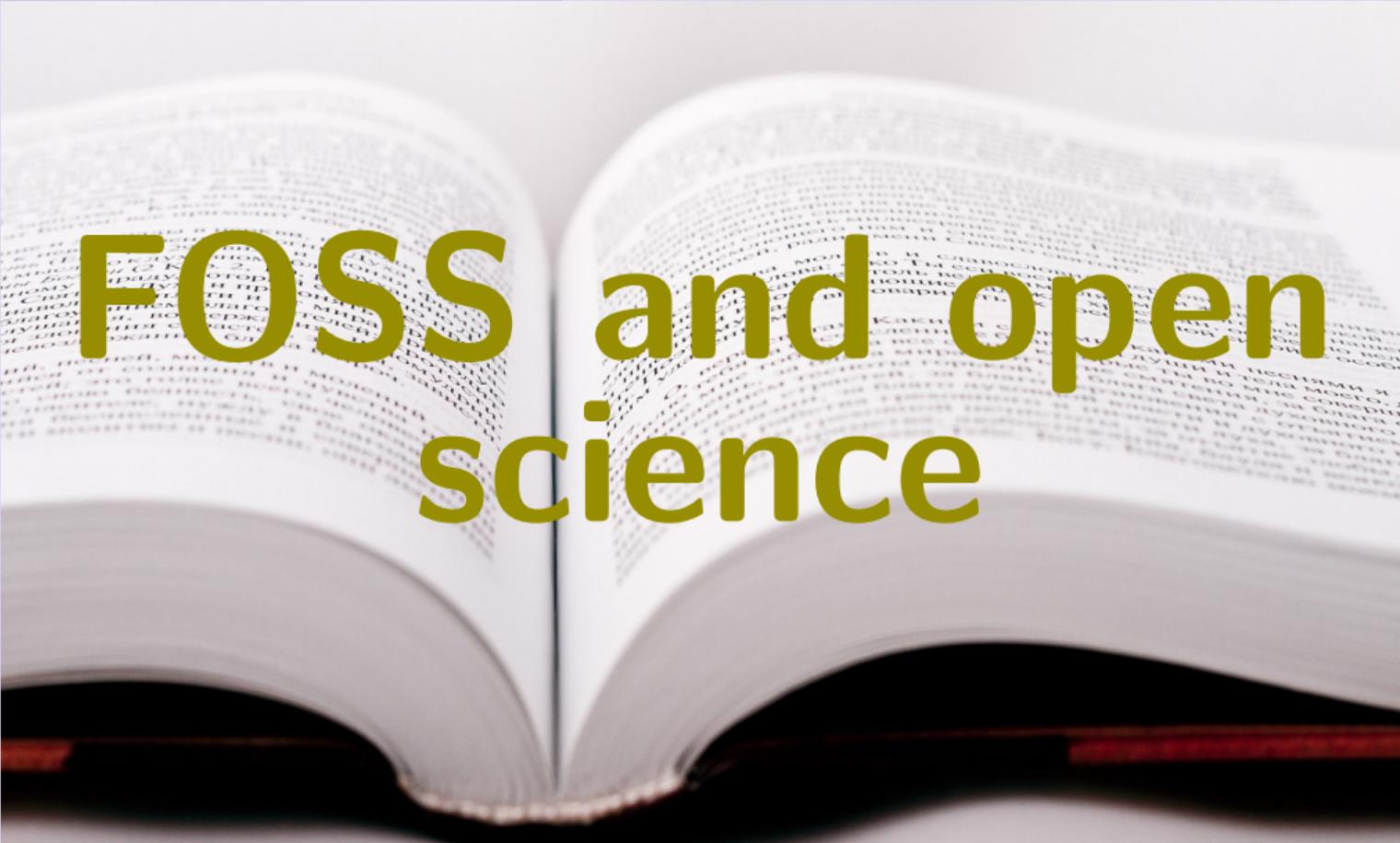
2 Recognition

3 Summarizing

FOSS and open science

What is FOSS(*)?

Anyone can use it

Anyone can study it and modify it

Anyone can redistribute it

Anyone can redistribute modified versions

https://www.gnu.org/philosophy/free-sw.en.html
https://www.debian.org/social_contract#guidelines
https://opensource.org/osd
(*) FOSS: Free, open source software

What else is FOSS?

- Development models
  (including review and issue reporting)
- Common infrastructure and tools
- Recognition model
  (sometimes based in metrics)
- Communities
  (with digital meeting points, conferences…)
- Governance models

Tracking FOSS
contributions

Jesus M.
Gonzalez-Barahona

FOSS and open
science

Recognition

Summarizing

References

Open science before open science

(to some extent)

FOSS and open
science

Recognition

Summarizing

References

Science and FOSS (1)

- Software is more and more critical for research: software as base infrastructure, and specialized software to obtain research results
- FOSS is the only chance for open science
- But "releasing as FOSS" is not good enough
- Embracing of FOSS good practices is needed

FOSS practices to embrace

Software used in research should be:

- shared $\Rightarrow$ **Preservation, Reuse**

- run (today, and tomorrow) $\Rightarrow$ **Reproduction**

- adapted, improved $\Rightarrow$ **Modification**

- scrutinized, criticized $\Rightarrow$ **Review**

- maintained, nurtured $\Rightarrow$ **Community**

Recognition

Who will do all of this?

Incentives

Actually, incentives are against!

- No "points" collected by doing the right thing

- Consequence: publish and abandon

- Remarkable exceptions: looking for trouble

- Marginal (scientific) recognition:
  only very successful cases (eg: scikit.learn)

Can this be reversed?

- Software as a "first class" artifact (as papers)
- Push for really reusable reproduction packages
- Recognize production of reused software
- Recognize participation in communities

How to recognize?

Software is...:

- • is incremental: identifying contributions
  ( )

- • is collaborative: individualizing contributions
  ( )

- • needs "gray work": visualizing contributions
  ( )

Not an easy task

How to recognize?

Software is...:

- is incremental: identifying contributions (but research is too)
- is collaborative: individualizing contributions (but research is too)
- needs "Gray work": visualizing contributions (but research is too)

Not an easy task

Possible approach

(Almost) everything is public

so...

collect data and evidences from public repositories

Possible approach: tools

- Development stats: commits, etc.
- Usage stats: downloads, cites, etc.
- Community stats: issues, pull requests, Q&A, etc.

All of this informing a qualitative assessment

Summarizing

FOSS is fundamental for research, but incentives don't help

The situation must be reversed

References, credits, license

- "Évaluation des Logiciels" (Inria)
  https://hal.inria.fr/hal-03110723

- "Artifact Review and Badging"
  https://www.acm.org/publications/policies/
  artifact-review-and-badging-current

- "Attributing and Referencing (Research) Software:
  Best Practices and Outlook from Inria"
  https://hal.archives-ouvertes.fr/hal-02135891

- "Software citation principles"
  https://peerj.com/articles/cs-86/

Credits

 Book, by NikolayFrolochkin, Pixabay.
License: Creative Commons CC0

Tracking FOSS
contributions

Jesus M.
Gonzalez-Barahona

FOSS and open
science

Recognition

Summarizing

References