# How we have made our data publicly available

Jana Hozzová

Institute of Computer Science

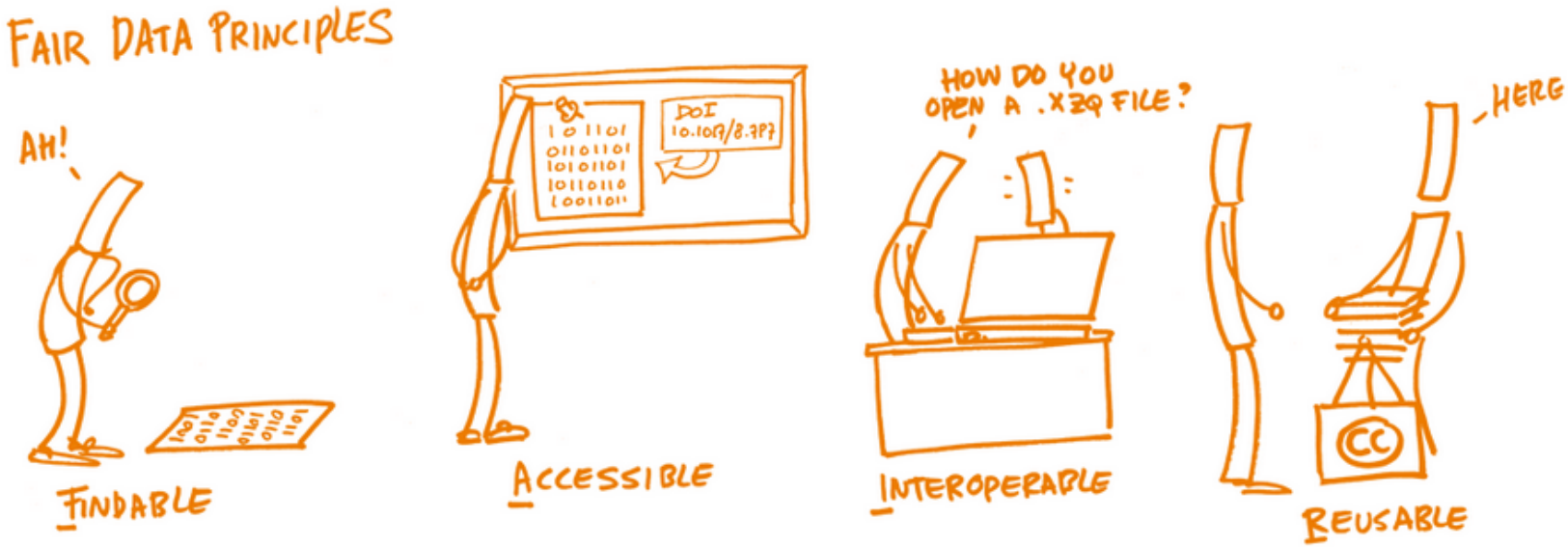
# What is this about?

We all know that we “should” make our scientific data publicly available.
We (HiPerCore group) have actually done that.
Here is what we have learned.

# Open science

- open data: tables, images, videos, text files
- open source: code, scripts, frameworks, web services
- open reproducible research: pre-registration, lab protocols, workflows
- open access

# Open science




https://book.fosteropenscience.eu/en/02OpenScienceBasics/02OpenResearchDataAndMaterials.html

# Why you should do it?

- useful research $\gg$ publishable research
- diminishes reproducibility crisis in science
- handles publication bias to some extent
- enables faster progress
- makes it cheaper
- supports innovation
- engages public

https://www.fosteropenscience.eu/

# What's in it for me?

- makes research more visible
- increases number of citations
- gives you credit for your work
- creates funding opportunities
- facilitates networking

It makes you a better, more conscientious researcher.

# Data suitable for opening

- demanding to acquire (time, resources, people)
- potential for re-use
- others can use them
  - to reproduce your research
  - to explore other hypotheses
  - in their methodology (training data, baseline, default parameter values, scripts for processing)
  - in their evaluation (evaluation data, comparable results, the truth, metrics for comparison)

# Make them usable for computers

- create a clear directory structure
- name data files/scripts/options consistently
- format files for easy machine processing
- make the output/logs easy to process by a machine
- replace hard-coded parameters with options
- fix the code or make bugs known
- make building and deployment easy
- handle dependencies' versions

# Make data usable for people

- describe metadata: file structure, columns' names, units, cmd options
- make data readable for humans if possible
- describe how data were measured/can be recomputed
- describe the processing: filtering, outliers, omitted data, corner cases, conversions
- provide scripts for processing the data
- deal with sensitive data (anonymization, consent)
- decide licencing

# Make code usable for people

- start with readable, refactored, commented code
- add logging
- provide examples of usage and/or tutorials
- write documentation or at least README
- set reasonable default values for parameters
- set contribution rules
- decide licencing

All of this takes quite some time.

# Ways to make data public

- supplementary material alongside the research article
- data repository
  - DOI, reference in your research article
  - Mendeley Data
  - Zenodo
  - OSF

# Ways to make data public

- data repository + data article
  - no hypotheses, no conclusions, just data and their description
  - two DOIs, reference in your research article
  - general: Data in Brief (Elsevier), Journal of Open Research Software (Ubiquity Press), CODATA Data Science Journal (CODATA), Data (MDPI)
  - life sciences: Scientific Data (Nature), GigaScience (BioMed Central), BMC Research Notes (BioMed Central)

# Data in Brief

- research journal for data articles
- all research areas
- template for article structure
- separate submission or co-submission
- review: complete, well-documented data
- preprints allowed, open access for USD 700

# Lets' look at repositories and articles

https://github.com/HiPerCoRe/KTT
nonpublic repository on https://gitlab.ics.muni.cz/
https:
//data.mendeley.com/datasets/nn53dskr7z/2
https://arxiv.org/abs/2102.05299
https://arxiv.org/abs/2102.05297

# Where should * go?

- github: C++ application, some scripts, description of that
- non-public gitlab: scripts, data
- data repository: subset of data, some scripts
- data article: description of data and its processing
- research article: hypotheses and conclusions

# Support at ICS

- Data Security and Management Department within Cybersecurity and Data Management Division
- Jiří Marek as manager of Open Access
- provide support especially when dealing with sensitive data
- provide money for open access fees

# So, how can I even start with this?

- the best way to learn is by doing
- supplementary material: scripts to generate the figures and raw data tables
- data repository: a small subset of your data (one method, one GPU, one instance)
- data repository + data article: consistent dataset with all scripts

# Conclusion

- as writing a paper forces you to tidy up and "finish" your research
- as open source forces you to tidy up and "finish" your code
- open data force you to tidy up and "finish" our data
- always easier to keep things tidy along the way
- but even if not, do it after paper/grant/* submission
- it's good for you, good for others, good for science
- it makes you a better researcher