FAIR data Management plan

Miguel Colom
https://mcolom.perso.math.cnrs.fr/

CENTRE BORELLI

école normale supérieure paris—saclay

université PARIS-SACLAY

Data

Discussion

- Should we consider data as source code + “files”? Or should we separate source code and extra files?
- Different licenses for sources and data?
- Different management plans?

Data

Discussion: how do you manage your data?

- How many of you use “data“ in your projects?
- Do you or your lab have a specific “data management plan”?
- How do you reference your data from an article or source code?
- Where do you store your data?

Data

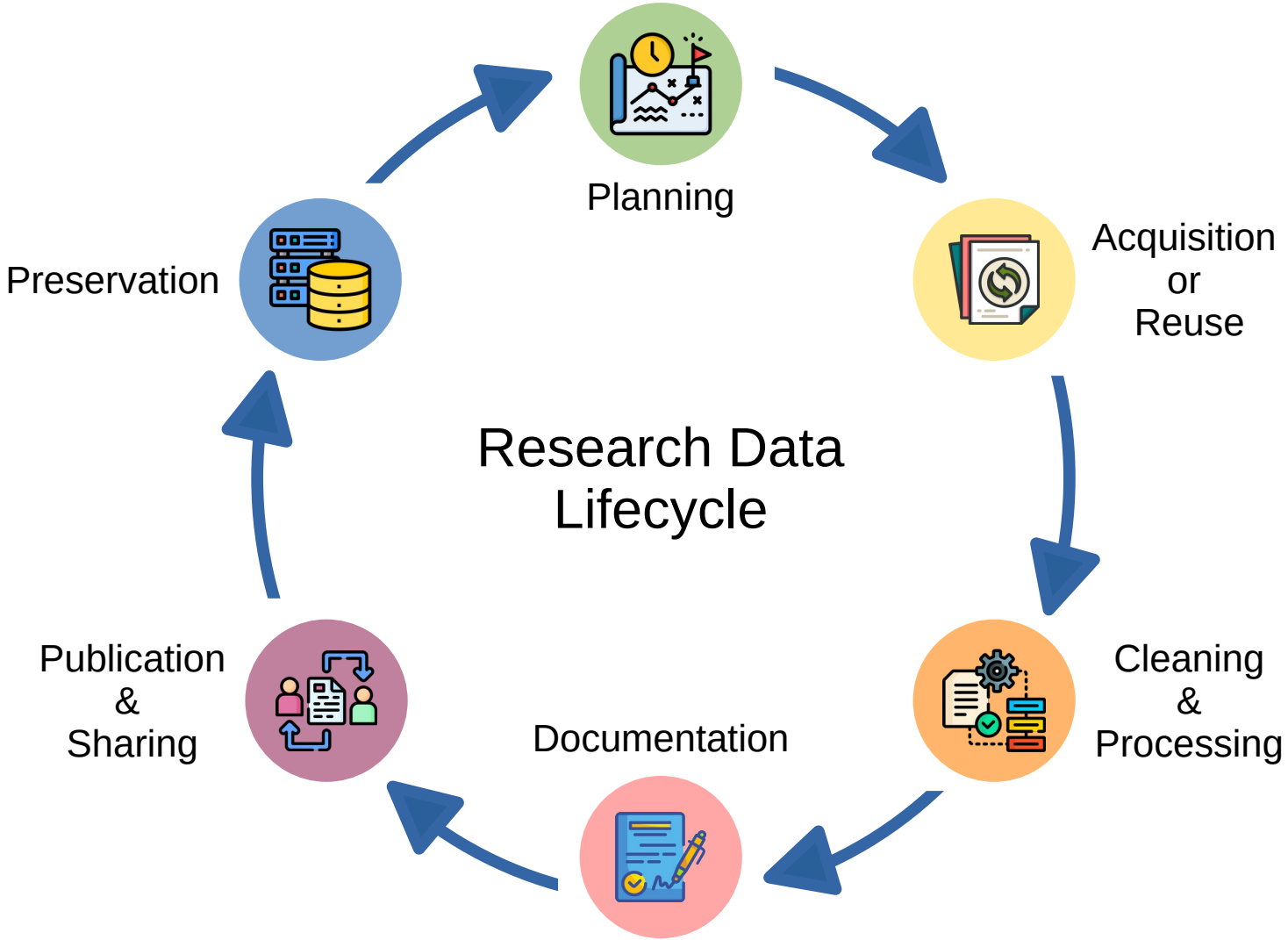
Discussion: how do you manage your data?

- Could your data disappear and the links to it get broken if the provider disappears? Is it perpetually preserved?
- What is more important? The software or the data? In classic methods? In modern AI methods?
- Do you add a version number to your data?
- What if you need to change the data? Say, a bug in the code that generated it for an article

Data

Discussion

- What would you require to data to be useful for scientific research?


Research Data Lifecycle
Planning
Acquisition or Reuse
Cleaning & Processing
Documentation
Publication & Sharing
Preservation

FAIR data

- FAIR is an acronym for:
  - Findable
  - Accessible
  - Interoperable
  - Reusable

The "*FAIR*" term first appeared in:

Wilkinson, M. D., et al. (2016). "The FAIR Guiding Principles for scientific data management and stewardship." *Scientific Data*, 3, 160018. https://doi.org/10.1038/sdata.2016.18

FAIR data: FINDABLE

Discussion

- Isn’t Google o chatGPT just enough to find the data?
- What do you think is “findable” data?
- What do we need to make data “findable”?

FAIR data: FINDABLE

Requirements to be Findable, from Winkinson's article:

- F1. (meta)data are assigned a **globally unique** and **persistent identifier**
- F2. data are **described** with **rich metadata**
- F3. **metadata** clearly and explicitly **include the identifier of the data it describes**
- F4. **(meta)data** are **registered** or **indexed** in a **searchable resource**

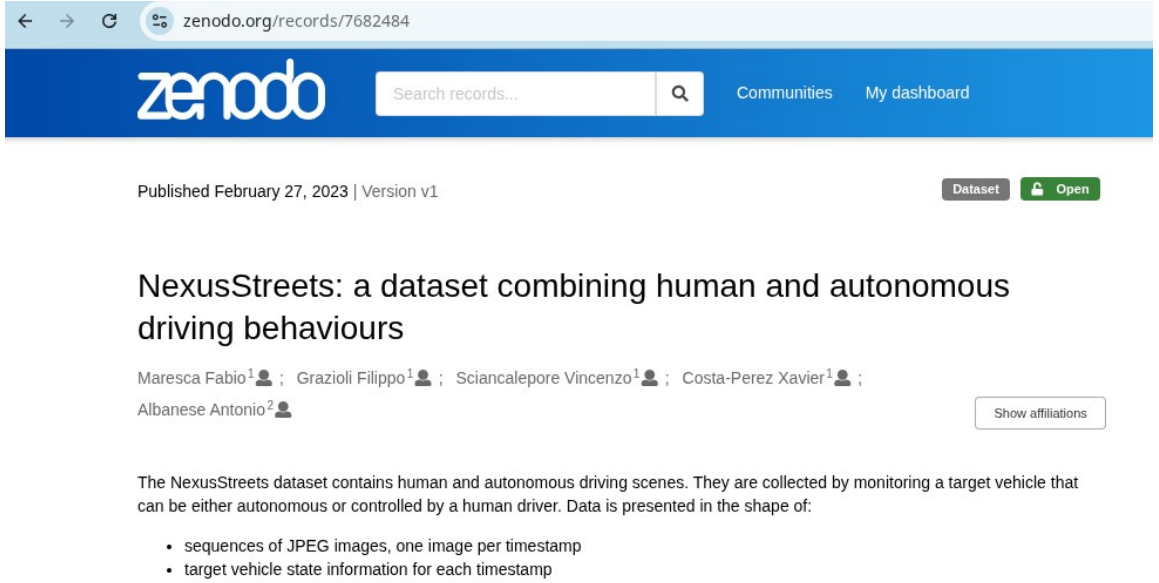
FAIR data: FINDABLE

Case study: Zenodo

- F1. (meta)data are assigned a globally unique and persistent identifier
  - OK, each record has a DOI
- F2. data are described with rich metadata
  - OK, it uses the DataCite's Metadata Schema
- F3. metadata clearly and explicitly include the identifier of the data it describes
  - OK, the DOI is included in the corresponding metadata record
- F4. (meta)data are registered or indexed in a searchable resource
  - OK, Zenodo search engine, and indexing with DataCite servers

FAIR data: FINDABLE

An example of retrieving metadata with Zenodo's REST API



FAIR data: FINDABLE

Record 7682484.

Let's do it!
On Linux:
$ curl https://zenodo.org/api/records/7682484 > record.json
$ firefox record.json

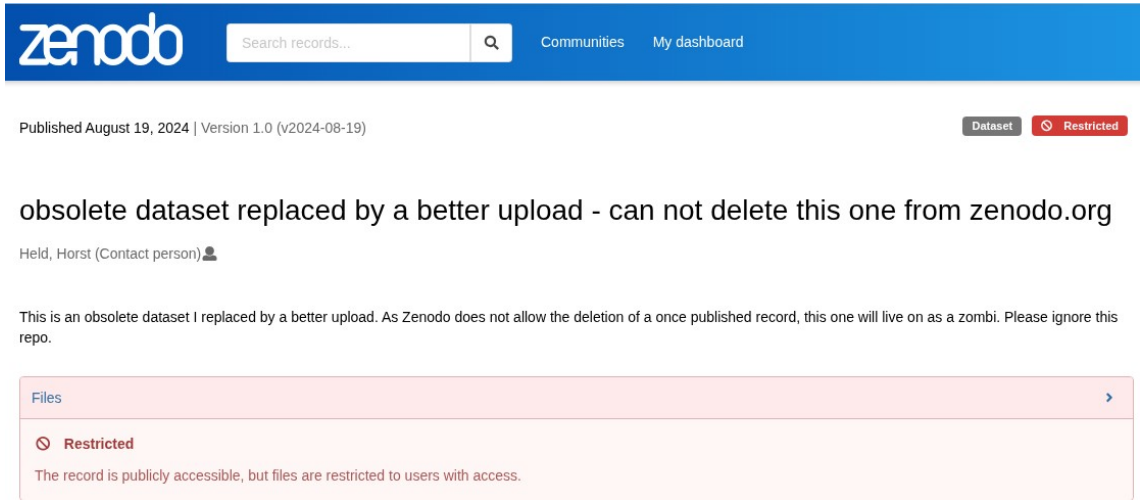
Discussion: do we have everything we need for findability?
Would you add anything else?

FAIR data: ACCESSIBLE

- A1. (meta)data are retrievable by their identifier using a standardized communications protocol → OK, in Zenodo we used the ID 7682484.
  - A1.1. The protocol is open, free, and universally implementable → OK, we used HTTPS and JSON
  - A1.2. The protocol allows for an authentication and authorization procedure, where necessary → OK, to modify our own records, but it was not needed for reading
- A2. Metadata are accessible, even when the data are no longer available → Let's see an example...

FAIR data: ACCESSIBLE

A2. metadata are accessible, even when the data are no longer available → Let's see an example: https://zenodo.org/records/13340731



FAIR data: ACCESSIBLE

- Let’s see another example: https://www.kaggle.com/datasets/sanjanchaudhari/facebook-dataset?select=FB.csv

- Discussion: is the data accessible?

FAIR data: ACCESSIBLE

$ curl https://zenodo.org/api/records/13340731 > obsolete.json

$ firefox obsolete.json

Discussion: why would you keep an obsolete record?

Why not simply removing everything?

FAIR data: INTEROPERABLE

- I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation → For example, the Fast Healthcare Interoperability Resources (HL7) has proper keys for medical data

- I2. (meta)data use vocabularies that follow FAIR principles → That's kind of a circular definition ;)

- I3. (meta)data include qualified references to other (meta)data

  curl -X GET "https://hapi.fhir.org/baseR4/Observation?subject=Patient/example" -H "Accept: application/fhir+json"

- The response refers to a vocabulary and interoperable attributes: https://hl7.org/fhir/vitalsigns.html

FAIR data: INTEROPERABLE

- An example of a non-interoperable format:
  - An Excel spreadsheet
    - The format is controlled by a company and could change. Not an standard
    - It requires the use of proprietary software
    - It's got limited machine-readability capability
    - Not necessarily linked to a particular and standard vocabulary
- Discussion: can you think about other non-interoperable formats?
  What is the current situation?
  Have you ever faced problems related to the format of the data?

FAIR data: REUSABLE

- An example of a non-interoperable format:
  - An Excel spreadsheet
    - The format is controlled by a company and could change. Not an standard
    - It requires the use of proprietary software
    - It's got limited machine-readability capability
    - Not necessarily linked to a particular and standard vocabulary
- Discussion: can you think about other non-interoperable formats?
  What is the current situation?
  Have you ever faced problems related to the format of the data?

FAIR data: REUSABLE

- R1. meta(data) are richly described with a plurality of accurate and relevant attributes
  - R1.1. (meta)data are released with a clear and accessible data usage license → Discussion: why a license is needed for reusability?
  - R1.2. (meta)data are associated with detailed provenance → Indeed, without the details of the origin of the data, we cannot trust it. Or to cite it properly.
- R1.3. (meta)data meet domain-relevant community standards → For example, a medical dataset has a specialized vocabulary, which wouldn't be useful to describe the dataset of images for autonomous driving

FAIR data: REUSABLE

- An example of a non-reusable dataset
- https://up42.com/goingup/high-resolution-satellite-data
  - Commercial license, restricting derivative works and redistribution
  - Very limited provenance metadata, or information about the processing chain

FAIR data: REUSABLE

- An example of a satellite image dataset which is reusable: Sentinel 2 data
- See, for example: https://eos.com/products/crop-monitoring/satellite-data-api


```
{
  "id": "S2A_MSIL2A_20230515T105031_N0509_R094_T30TVK_20230515T132812",
  "platform": "Sentinel-2A",
  "instrument": "MSI",
  "processingLevel": "Level-2A",
  "creationDate": "2023-05-15T13:28:12Z",
  "processingBaseline": "05.09",
  "datatakeIdentifier": "GS2A_20230515T105031_033010_N05.09",
  "producer": "European Space Agency",
  "orbitNumber": 33010,
  "tileId": "T30TVK",
  "cloudCover": 5.12,
  "processingCenter": "IPF",
  "license": "Copernicus Sentinel data (2015-present), open access"
}
```


- Provenance: platform, instrument, processingLevel, dates, …
- License (open access)

The data management plan

FAIR formats

- Text: JSON, XML, CSV, TXT, ...
- Datasets: they depend on the discipline. For example, geoJSON for spatial data, FASTA for genomic sequencing, or HL7 FHIR for clinical records
- Databases: they structure internally the data with some binary format. Make sure you use a free/open source program that defines the format and allows for reading it. Examples: MongoDB (for JSON-like documents), Redis (for high performance, storage in RAM), Neo4j (network analysis)

FAIR formats

- Images: use standard lossless format, to avoid loss of information. For example, PNG, TIFF. But not JPEG!

- Audio: again, standard lossless formats. For example, avoid MP3 (lossy, proprietary), and use instead FLAC or WAV.

FAIR license

- Discussion: do you add a license to your released data? Why?
  Would it be OK not to add a license, if you don’t case about what people can do with your data?

FAIR license

- Any dataset you produce must come along a license
- Without a license, users don’t have any rights! All is forbidden by default: no reuse, no redistribution, no modification, no derivative works, …
- Certainly, this is not what you want…

FAIR license

- Which license?
- Data is different from software. Avoid software-specific licenses (GPL, BSD, …)
- Creative Commons (CC) are adequate licenses. For example, CC-BY (with attribution, and full reuse)
- Specifically for datasets, you have ODC-BY
- And many others. It’s important to pick a license that allows to get credit (that’s important for you as a researcher) and also allows reuse

Organization of the data

- You need to organize your files in folders, and give names. Check how this is done by your community in your field
- Make the structure and naming of your files reflect the organization of the data. For example: europe/spain/madrid/temperature/12-06-2025/15h00.csv
  - This allows to navigate through the data easily
- Use the expected formats and be consistent. For example, for the dates or times.
- Adhere to the corresponding vocabulary. For example, “temperature”, and not “degrees”
- Ensure that you’ve added all needed files: for example LICENSE, VERSION or AUTHORS

Organization of the data

- Add metadata
- There are several tools that can help you

- DataCite Metadata Generator.
  - https://dhvlab.gwi.uni-muenchen.de/datacite-generator/

- Metadata Wizard
  - https://www.usgs.gov/software/metadata-wizard

- Research Object Crate (RO-Crate)
  - https://www.researchobject.org/ro-crate/

- CKAN
  - https://ckan.org/

- Zenodo has its own tool for adding metadata
  - https://zenodo.org/

Organization of the data

Discussion

- Do you know what is a hash function?
- How it might be useful for data consistency or identification?

Organization of the data

- A hierarchical organizing allows to define intrinsic Permanent Identifiers (PIDs)
  - An intrinsic identifier is computed from the contents of the data itself
  - We'll see later an example of an intrinsic identifier (but for code), the SWHID

Consistency of your data

- Before publishing your data, it's important to ensure its consistency
- Computing hash-based control sums such as SHA1 is a good way
  - Some PIDs use them to provide *intrinsic* identifiers

Consistency of your data

- Ensure that the authors are well identified. For example, with the ORCID
- Add hints on what you would expect if the data is re-regenerated
- If you’re using JSON: validate against a JSON schema
- If you’re using a database, always enforce relational rules with foreign keys. Rules
- Ensure that you’re using only terms from your chosen vocabulary
- Use automatic tools.
- For example FAIRshake: https://fairshake.cloud/

Sensitive data

- Some of your data might be sensitive
  - For example, you're building an anonymized dataset from medical data
  - After being anonymized, it can be FAIRly share

- There is specific regulation for sensitive data
  - The General Data Protection Regulation (GDPR), arts. 6 and 9 on medical data

- Check if you're even allowed to deal with a particular type of sensitive data! Designate an authorized responsible.

Sensitive data

- As a general rule, use encryption to protect private date
  - Multiple solutions for data encryption, both software and hardware
- When transmitting data from one place to another (say, a backup), always encrypt
  - Straightforward with openSSH (scp, rsync under ssh, etc.)
  - Avoid 3rd party services
- In general, the corresponding regulation state clearly the conditions for storage and transmission of sensitive data

Backup your data

- Especially when building a dataset, make regular backups of your data
- Compute hash-based control sums of your hierarchical data
  This allows to detect inconsistencies
- Make incremental copies of your data.
  This allows to get back to a consistent state
- The consistency should be checked not only at binary level, but at entity level.
  For example: is the ORCID of all the researchers listed in the dataset valid?
  Do the computed intrinsic identifiers correspond to the contents of the data?
- Use a RAID-1 (mirroring) storage to avoid losing data in case of a disk failure

Preserve your data

- A local copy is needed, but FAIR data needs to be made available to everybody

- Just making the dataset available online is not enough, since it needs
  - to be preserved for the long term (perpetually, ideally)
  - to be referenced with PIDs
  - to be searchable

- Thus, you need a proper platform to fulfill these minimal FAIR requirements
  - For data, Zenodo is a good choice, among others
  - For source code, Software Heritage is an excellent choice.

- Which data should I preserve?
  - It depends on the policy of each platform
  - For example, Software Heritage wishes to preserve all source code, including support or pre-processing code you might think it's not that relevant
  - Other platforms, especially for data, might expect some data curation or filtering
  - Check your funder requirements. Some will request to make source and data available, whereas others might ask for a time-limited embargo.

A few references

- Data Management Guida, Univ. of Cambridge
  https://www.data.cam.ac.uk/data-management-guide

- FAIR Data, A Quick Guide for Researchers
  https://think.f1000research.com/wp-content/uploads/2021/02/F1000Research-Open-Data-FAIR-Data-Quick-Guide.pdf

- Managing and sharing data. UK data archive.
  https://dam.ukdataservice.ac.uk/media/622417/managingsharing.pdf

- Data Management Plan checklist
  https://www.dcc.ac.uk/sites/default/files/documents/data-forum/documents/docs/DCC_Checklist_DMP_v3.pdf

Preserve your data

- A local copy is needed, but FAIR data needs to be made available to everybody

- Just making the dataset available online is not enough, since it needs
  - to be preserved for the long term (perpetually, ideally)
  - to be referenced with PIDs
  - to be searchable

- Thus, you need a proper platform to fulfill these minimal FAIR requirements
  - For data, Zenodo is a good choice, among others
  - For source code, Software Heritage is an excellent choice.

- Which data should I preserve?
  - It depends on the policy of each platform
  - For example, Software Heritage wishes to preserve all source code, including support or pre-processing code you might think it's not that relevant
  - Other platforms, especially for data, might expect some data curation or filtering
  - Check your funder requirements. Some will request to make source and data available, whereas others might ask for a time-limited embargo.