ARC-Finder – A simple, locally-deployed tool to find your peer's research data

Projektmodul (Modul 9) des Zertifikatskurs FDM (15.07)
2021 / 2022

Dominik Brilhaus,
https://orcid.org/0000-0001-9021-3197

2022-06-09

Contents

Motivation

Research is a highly collaborative endeavor that builds on synergistic interaction between different stakeholders enabled by efficient knowledge exchange. Gaining a prompt overview of the ongoing research efforts – both pre- and post-publication – is oftentimes hindered (for social, legal or technical reasons) even between parties of spatially closest and well trusted surroundings of a collaborative consortium such as the Cluster of Excellence on Plant Sciences (CEPLAS[1]). The key to enable discussion on and exchange of research data is *findability*, the first layer of the FAIR principles of data stewardship. The project presented here aims to address this layer, by making CEPLAS research easily findable and visible amongst CEPLAS researchers and showcase the beauty and ease of data sharing to spike fruitful collaborations with peers.

State of the art

Research data management within CEPLAS is closely aligned with DataPLANT[2], the NFDI consortium for plant sciences. DataPLANT has developed the Annotated Research Context (ARC[3]), a directory structure for research objects. Annotation of research data in the ARC is based on the metadata schema ISA[4] (for investigation – study – assay). Serialized in spread sheet format as *ISA-tab* this allows intuitive, flexible and yet structured and conclusive metadata annotation of the versatile data types produced in plant sciences. ARCs are git[5] repositories that can be shared via DataPLANT's DataHUB[6], a customized GitLab[7] instance with a federated authentication interface to allow controlled access across institute borders. Although the ARC environment is continuously being developed, the choice of these key technical pillars are set: (a) ARC as the structure, (b) ISA as the metadata language, (c) git as version control logic and (d) gitlab for ARC collaboration and user management. This allows to leverage the ARC and develop at least intermediate solutions for data findability, knowing that time and efforts are well-invested, since both (meta)data inputs in as well as secondary outputs dependent on the ARC will be adoptable and migratable in the future.

[1] https://ceplas.eu "CEPLAS"
[2] https://nfdi4plants.de "DataPLANT"
[3] https://github.com/nfdi4plants/ARC "ARC specifications"
[4] https://isa-tools.org/ "ISA Metadata Schema"
[5] https://git-scm.com/ "Git"
[6] https://git.nfdi4plants.org "ARC DataHUB"
[7] https://gitlab.com "GitLab"

Approach

This project focuses on metadata at the highest project and least sensitive (i.e. ISA's "investigation") level to minimize user input or possible discomfort with data sharing and will be achieved in four concerted, but independent modules of metadata

1. collection,
2. retrieval,
3. restructure, and
4. representation.

First, metadata is collected – manually or supported by automation – in the ISA investigation spread sheet, packaged in ARCs and submitted to the DataHUB by individual volunteers. Here, access to the ARCs can be controlled to share them publicly or with invited collaborators. The CEPLAS-ARC-Finder selectively retrieves, downloads and dumps the metadata locally on the user's machine. The CEPLAS-ARC-Finder then restructures the metadata into a simple spreadsheet-based database. From the database the investigation data is finally read and represented by a user interface that enables finding the data available to the individual user.

Caveats and places for future improvements

isa.investigation.xlsx

- read from isa.json rather than isa.investigation.xlsx
- Rationale: yet another detour / dependency to produce isa.json
- direct user-input to isa.investigation.xlsx can be read immediately
- xlsx can become big and needs to be dumped
- json could be read on-the-fly

branches

- reading only from default git branch `main` (not e.g. master or others)

efficiency

- tool dumps, pulls freshly every time it is called
  - nothing memorized and updated

  - by design sqlite db is always overwritten -> data could be appended

- selectivity
  - not all ARCs, but just selection (e.g. group)
  - error-prone: non-clean ARCs

Appendix

Dependencies

Softwares

The softwares and platforms listed below are those used during development and testing.

Software	Version	Platform
GNU bash	3.2.57(1)-release	x86_64-apple-darwin21
curl	7.79.1	x86_64-apple-darwin21.0
R	4.2.0	x86_64-apple-darwin17.0

R libraries

To provide best reproducibility, R dependencies are handled via package `renv`[8] (version 0.15.3) and stored in the root file "renv.lock". In the first step of `arcFinder`, the virtual environment is automatically restored, including installation of all required dependencies. Depending on the local setup (installation of R and packages), this may take some time. However, `renv` prevents interference with the local setup, thus keeping your system intact. The following lists packages specifically loaded for individual R scripts:

Script name	Package	Main purposes
03_parse_isaInvxlsx.R	readxl_1.4.0 (part of `tidyverse`)	Reading data from Microsoft Excel workbooks

Platform

- The DataPLANT's DataHUB[9] is a customized instance of GitLab[10], currently running under ### TODO version ###
- Data is retrieved via GitLab API version 4

[8] [<https://rstudio.github.io/renv/) "renv"

[9] https://git.nfdi4plants.org "ARC DataHUB"

[10] https://gitlab.com "GitLab"

After registration[11] with DataPLANT, users can share and access non-public ARCs via DataPLANT's DataHUB[12]. As explained in the `arcFinder`'s README, a GitLab private access token (PAT) needs to be generated within the DataHUB[13] and provided to `arcFinder`.

Checks and tests

Currently tested only under the following constellation

- bash and zsh
- macOS Monterey 12.3.1, Platform: x86_64-apple-darwin17.0 (64-bit)
- R version 4.2.0 (2022-04-22)

User instructions

gitlab token

1. Generate a personal access token (PAT) at DataHUB
2. Two options to use the PAT
    - store the PAT in a file and use this in the script XXX {### TODO}
    - directly paste the token as argument to script XXX
3. Supplying a "wrong token" (i.e. any non-sense string) currently breaks the script.

Permissions

- make all bash scripts executable?

Lessons learned

- first time really making use of an API
- first time thinking about
    - dependencies
    - where to put what
    - logs
- ARCs in (sub)groups

[11] https://register.nfdi4plants.org/ "DataPLANT registration"
[12] https://git.nfdi4plants.org "ARC DataHUB"
[13] https://git.nfdi4plants.org "ARC DataHUB"