FAIR+ reproducible workflows NIH/NCI

Pjotr Prins & Arun Isaac (UTHSC/Tennessee)

October 8, 2021

Contents

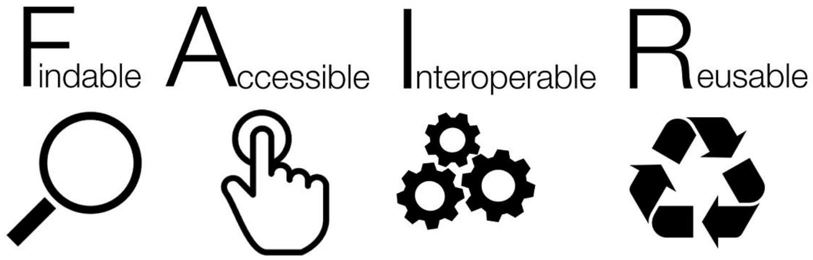
1 Introduction

Pjotr Prins

- Assistant Professor UTHSC/GGI
- Software solutions for genomics/genetics
- Working on pangenomics, genotyping and genomewide-association

2 FAIR

What is FAIR?



3 FAIR Data

What is FAIR not?

- FAIR is not 'Open Data'
- FAIR is not reproducible data

(part of the appeal)

4 FAIR+

What is FAIR+?

- FAIR+ brings in compute
- To have reproducible analysis we need 'Open data'
  - data that can be downloaded
- Online analysis

5 Homomorphic encryption

Encrypt data in such a way that you can still do (reproducible) analysis. We published HEGP for GWA

=> `https://hegp.genenetwork.org/`

Homomorphic Encryption of Genotypes and Phenotypes (HEGP) Genetics June 1, 2020 vol. 215 no. 2 359-372 by Richard Mott, Christian Fischer, Pjotr Prins, Robert William Davies

6 Privacy preserving graphs

Publish data in such a way that you can still do analysis, but you can't trace back to the individual built on formal models of differential privacy:

=> `https://nlnet.nl/project/VariationGraph/` by Erik Garrison (UTHSC)

7 PubSeq Example

=> https://covid19.genenetwork.org/

- PubSeq started simply as a public data resource for SARS-CoV-2 sequencing with live workflows
- No all so-called public repositories are open
- This hampers research

Scientists call for fully open sharing of coronavirus genome data Nature. 2021 Feb;590(7845):195-196.

8 PubSeq

- Online compute using common workflow language (CWL)
- Cloud platform sponsored by Amazon OpenData, AWS and Curii
- Virtual HPC at SARA Amsterdam (Sas Swart & Michael Crusoe)
- Permanent identifiers and federated data using IPFS
- Amazon Open Data initiative (TCGA, NCBI Sequence Read Archive etc)

9 Innovation

PubSeq as an initiative triggered innovation:

- Pangenome work on SARS-CoV-2 - methods for large scale phylogeny
- Demonstration platform for best practices (CWL, RDF, Arvados)
- Metadata normalisation (Elixir/EBI)

=> https://covid19.genenetwork.org/ => https://github.com/pubseq/bh20-seq-resource/tree/master/workflows => https://github.com/common-workflow-library/bio-cwl-tools

10 FAIR data should be as free and open as possible

- FAIR does not mean open
- FAIR does not mean reproducible
- So called ’public’ repositories are often not free and open
- Even proper public repositories, such as GenBank and EBI/ENA lack support for metadata and online compute
- FAIR+ is a step up with Open Data and compute

11 Holy grail of FAIR+

- Publish data + analysis in a journal
- Rerun analysis on demand [button]
- Be able to tweak parameters

12 Jupyter notebooks?

- What software is running?
- Is the data content addressed?
- Is the notebook itself captured in git?

Notebook or workflow in a Docker container?

- Maybe

13 Containers

- kernel namespaces - ipc, uts, mount, pid, network and user
- Docker = whales
- Singularity = cows
- GNU Guix = GNUs (!?)

- Reproducible containers
- GNU Guix can create Docker containers -> Singularity

14 Hands on

- Create GNU Guix container
- Explore inside of container
- Run tools in container

15 Guix container

```
~/opt/guix/bin/guix environment -C --ad-hoc python
```

```
python3
import os
```

```
len(os.listdir("/gnu/store"))
  19 # packages
```

16 Guix container added

```
~/opt/guix/bin/guix environment -C --ad-hoc python bash coreutils less vim binutils gl
```

includes 35 packages

```
python3 points to /gnu/store/hc2nql01h78qqxlcg4qril9c314m33zg-python-3.8.2/bin/python3
ldd python3
```

```
libc.so.6 => /gnu/store/fa6wj5bxkj5ll1d7292a70knmyl7a0cr-glibc-2.31/lib/libc.so.6
```

all paths are hard coded(!)
Try ruby, python-numpy

```
Dir.children("/gnu/store")
```

17 Run command with container

```
time ~/opt/guix/bin/guix environment -C --ad-hoc python python-numpy -- python3 -c 'pr
```

I do most of my development inside containers
Pure dependency control all the way down to glibc

- no software bleeding in from environment

18 Guix + Docker

~/opt/guix/bin/guix pack -f docker -S /usr/bin=/bin python python-numpy

```
docker load --input /gnu/store/42v185rm4pxmmbjg0sgl1bddj3mradgk-docker-pack.tar.gz
```

19 Fully reproducible

- The version of Guix includes the (content-addressed) package tree
- Install anywhere - native, container, Docker, Singularity…
- The same binary software is reproducibly deployed

20 GNU Guix Links

=> https://github.com/pjotrp/guix-notes/ => https://github.com/pjotrp/guix-notes/blob/master/CONTAINERS.org

21 Reproducibility

=> https://hpc.guix.info/blog/2019/01/creating-a-reproducible-workflow-with-cwl/

- Content addressable deployment (GNU Guix)
- Content addressable data (IPFS, S3, Arvados Keep)
- Hashed workflows stored in git (CWL, Nextflow, Jupyter, or whatever)

We have a fully reproducible stack!

22 Arun's presentation

- CWL is a bit unwieldy
- Can we create a CWL generator?

23 CWL conclusions

- Nextflow, snakemake, WDL, bash...
- CWL is the only one that can generate the others!
  - compile time type checking for tools and parameters
  - promise of sharing components
  - promise of run anywhere
- We quickly came to realise we should generate CWL (Brad Chapman)
- CCWL benefits from Lisp -> the language can generate graphs easily
- That does not mean you can't do this from Python

24 Conclusion

GNU Guix, CCWL are advanced concepts showing the way forward

- GNU Guix
  - Reproducible deployment
  - Light-weight containers
- CCWL
  - like shell scripting but with the benefits of CWL