Containers aren't a panacea, but a useful technology, with a range of tools to assist with

Fantastic containers and how to tame them

Yaroslav O. Halchenko[1], Kyle Meyer[1], Matt Travers[2], Dorota Jarecka[3], Satrajit Ghosh[3], Jakub Kaczmarzyk[3], Michael Hanke[4,5]

Contemporary neuroimaging research is computationally intensive and heavily relies on analysis software. Results of an analysis might depend not only on the version of the software the scientist is using, but also on the overall computational environment (including the operating system, preinstalled software, environment variables, and data). Containerized computing environments, using **Docker** (https://www.docker.com) and **Singularity** (https://www.sylabs.io/singularity), gained popularity as a means to maximize portability of complete environments, minimize operational variance across available infrastructure (laptops/HPC/cloud), and share complete environments. While containers are a very powerful technology, **additional care and tooling are necessary to use them efficiently and reproducibly**. We explore containers-oriented computing across the container life cycle—creation, validation (QA), storage (dissemination), and usage—along with approaches and tooling to assist at each step. This overview will focus on the reproducibility aspects, including the reproducibility of the containers themselves, and highlight recent technological developments.

Visit https://github.com/myyoda for more info

Track all input data, code, and computational environments needed to produce analysis outputs in version controlled datasets — and reproducibility you will achieve!

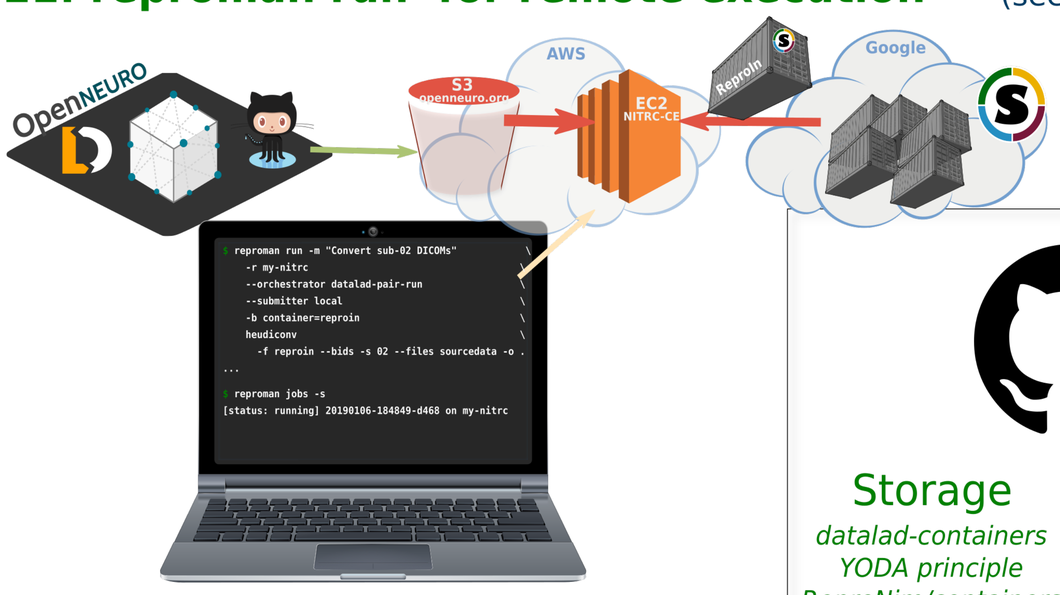
Containers ARE NOT the …
- reproducibility panacea
- solution for software sustainability or integration
- replacement for software distributions (e.g., GNU/Linux distributions, Conda, etc)

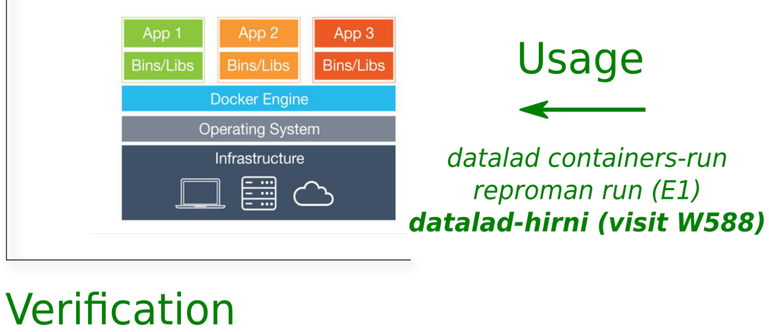
Containers are great for …
- Collaboration
- Computation
- Experimentation
- Reproducibility
- Teaching
- Troubleshooting

Containers are great because they ...
- are portable and can be used on most operating systems
- provide easy access to broad range of existing environments from container hubs
- can be used for “quick&dirty” bundling of computing elements
- are isolated and easy to install/remove
- simplify Continuous Integration
- could be kept under the same version control system as for code and data (DataLad)
- facilitate the introduction of a unified interface to a wide range of separate environments (e.g. BIDS apps http://bids-apps.neuroimaging.io, Boutiques, Flywheel Gears)

Container gotchas! They ...
- can be challenging
- can only contain GNU/Linux, and Singularity can only be run on GNU/Linux
- (Singularity) might leak host environment into containers
- might be incompatible with elderly host systems
- are a black box and typically there are no easy ways to figure out what and how was installed into a container and what licenses cover the pieces, etc (see e.g. https://lwn.net/Articles/786066/):
- security and legal implications
- difficult to inspect/compare different containers on how they differ (use https://github.com/GoogleContainerTools/container-diff,

E1: reproman run for remote execution





Verification

Automated testing of containers should be performed!
- **Boutique** specification that can contain **test definitions** to validate containerized pipelines [Glatard et al., 2018], https://github.com/boutiques/boutiques/#test-your-tool
- **Flywheel Gears** generated by the **gearificator** (https://github.com/yarikoptic/gearificator) carry regression tests, see https://github.com/yarikoptic/gearificated-nipype
- **Niflows** (https://github.com/niflows) use Testkraken (https://github.com/ReproNim/testkraken) to test neuroimaging workflows in various containerized environments, see e.g. testkraken's workflows4regtests/basic_examples/simple_workflow/parameters.yaml

E2: Neurodocker + nd_freeze

ReproIn reproducible container recipe generation

```
$ docker run --rm kaczmarj/neurodocker generate "(docker|singularity)" \
  --base=neurodebian:stretch \
  --pkg-manager=apt \
  --ndfreeze date=20190513 \
  --install vim wget strace time ncdu gnupg curl datalad pigz \
      ... \
  --run "curl -sL https://deb.nodesource.com/setup_6.x | bash - "\
  --install nodejs npm \
  --run "npm install -g bids-validator@1.1.1" \
  --run "mkdir /afs /inbox" \
  --run "echo '#!/bin/bash ... /neurodocker/heudiconv.sh && ... \
  --user=reproin \
  --entrypoint "/neuro
```

Dockerfile

```
FROM neurodebian:stretch

ARG DEBIAN_FRONTEND="noninteractive"

RUN apt-get update -qq \
   && apt-get install -y -q --no-install-recommends \
        neurodebian-freeze \
   && apt-get clean \
   && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/* \
   && nd_freeze  20190513

ENV LANG="en_US.UTF-8" \
   LC_ALL="en_US.UTF-8" \
   ND_ENTRYPOINT="/neurodocker/startup.sh"
RUN export ND_ENTRYPOINT="/neurodocker/startup.sh" \
....
```

Singularity

```
Bootstrap: docker
From: neurodebian:stretch

%post
apt-get update -qq
apt-get install -y -q --no-install-recommends \
   neurodebian-freeze
apt-get clean
rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
nd_freeze  20190513

export ND_ENTRYPOINT="/neurodocker/startup.sh"
apt-get update -qq
apt-get install -y -q --no-install-recommends \
   apt-utils bzip2 ca-certificates \
....
%environment
export LANG="en_US.UTF-8"
export LC_ALL="en_US.UTF-8"
export ND_ENTRYPOINT="/neurodocker/startup.sh"
```

snapshot.debian.org
snapshot-neuro.debian.net

1 Dartmouth College, NH, USA
2 TCG, USA
3 Massachusetts Institute of Technology, Cambridge, MA, USA
4 Institute of Neuroscience and Medicine, Brain & Behaviour INM-7, Research Centre Jülich, Germany
5 Institute of Systems Neuroscience, Medical Faculty, Heinrich Heine University Düsseldorf, Germany

corresponding author: yoh@dartmouth.edu

https://github.com/ReproNim/artwork

References and Acknowledgements

Glatard T, et al (2018): Boutiques: ... http://dx.doi.org/10.1093/gigascience/giy016.
Gorgolewski KJ, et al (2017): BIDS apps: ... http://dx.doi.org/10.1371/journal.pcbi.1005209.
Halchenko Y (2018): yarikoptic/gearificator 0.1. https://doi.org/10.5281/zenodo.2431133.
Halchenko Y, et al (2018): ReproNim/reproman https://doi.org/10.5281/zenodo.2403222.
Halchenko YO, et al (2012): Open is Not Enough. ... http://dx.doi.org/10.3389/fninf.2012.00022.
Hanke M, et al (2018a): datalad/datalad-container 0.4.0. https://doi.org/10.5281/zenodo.2431915.
Hanke M, et al (2018b): YODA: ... https://doi.org/10.7490/f1000research.1116363.1
Kaczmarzyk J, et al (2019): kaczmarj/neurodocker: https://zenodo.org/record/1477094.
Rampin R, et al (2016): ReproZip: ... http://adsabs.harvard.edu/abs/2016JOSS.2016..107R.
Sochat V (2018): The Scientific Filesystem. http://dx.doi.org/10.1093/gigascience/giy023.
DataLad development was supported by the NSF (CRCNS 1429999), German federal state of Saxony-Anhalt and the European Regional Development Fund (ERDF), Project: Center for Behavioral Brain Sciences, Imaging Platform. YH, KM and development of NICEMAN, nd_freeze and parts of DataLad were supported by ReproNim (NIH-NIBIB P41 EB019936).