CERN openlab

Data Platform for collection, storage, integration, analysis and distribution

CERN openlab Summer Student lightning talk

Diamantis Patsidis

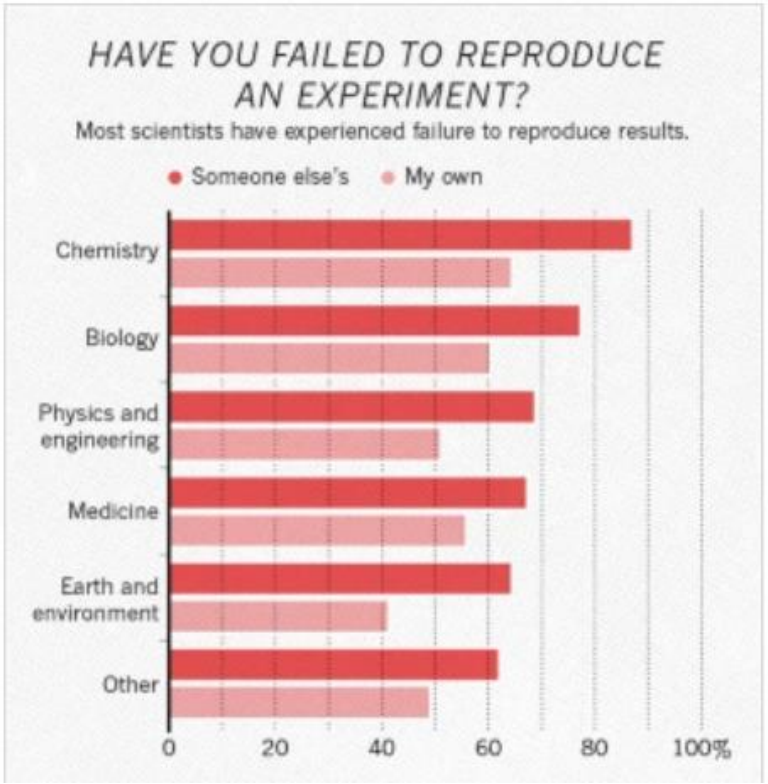
Supervisor – Anna Ferrari

06 / 09 / 2021

Reproducibility

What is reproducibility?

Enable others to reproduce results without difficulty



https://www.nature.com/articles/533452a (Monya Baker 2016.)

Reproducibility rates in biomedical research

11% (Begley and Ellis, 2012, Ioannidis 2005)

22% - 49% (Freedman 2015, from 5 independent research groups)

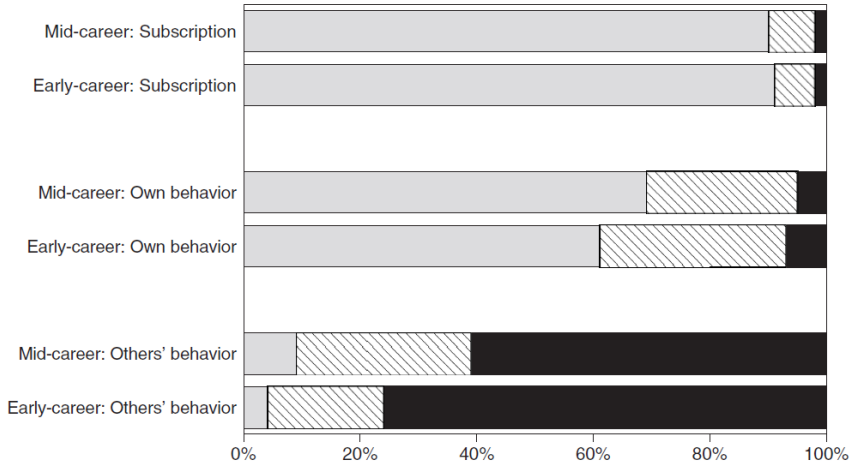


FIG. 3. Norm versus Counternorm Scores: Percent with Norm > Counternorm (dotted), Norm = Counternorm (striped), Norm < Counternorm (solid).

Normative_Dissonance_in_Science_Results_from_a_National_Survey_of_US_Scientists (2007)

Reproducibility recipe

Four main ingredients

Data (location)
Cloud, local drive

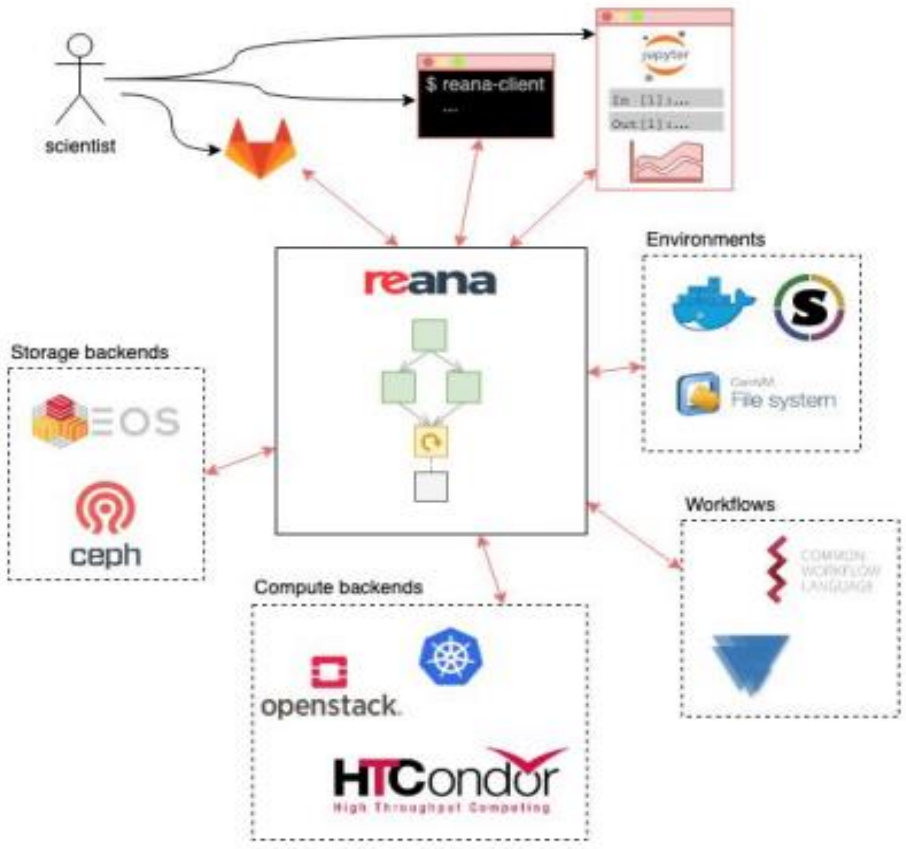
Code (location)
GitLab, local drive

Environment (software used, …)
Local computer, cloud computing

Workflow (order of scripts, …)
Bash script, README

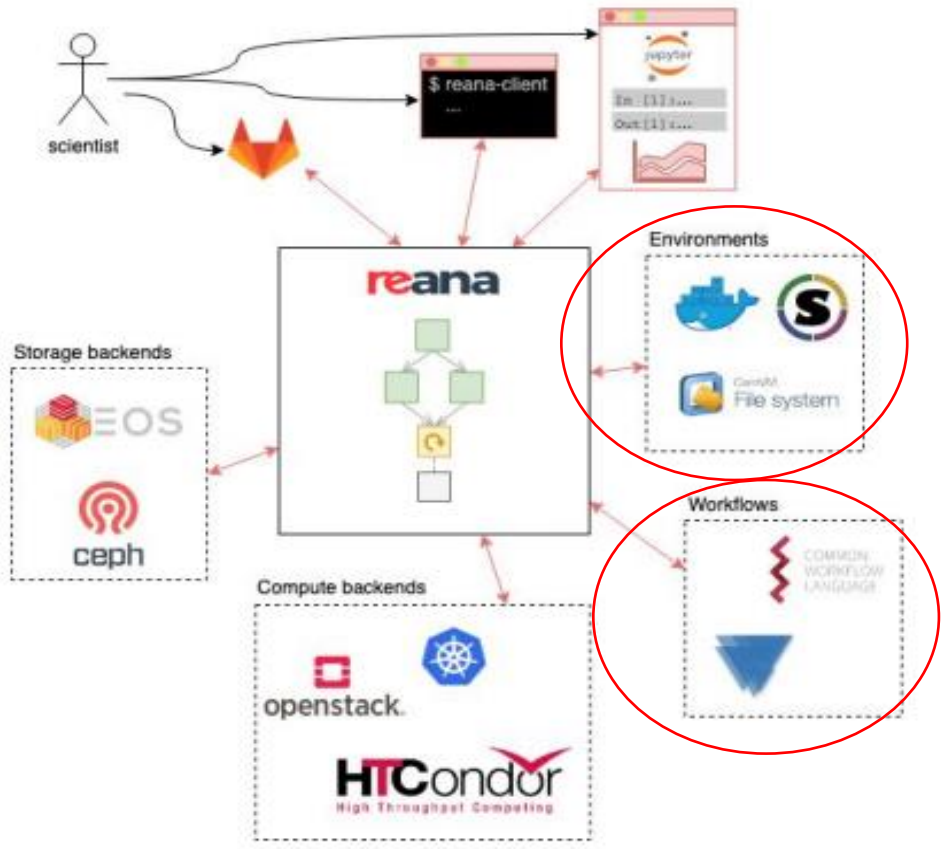
Analysis Reproducibility with REANA on Kubernetes (Diego Rodriguez - REANA team)

REANA



Analysis Reproducibility with REANA on Kubernetes (Diego Rodriguez - REANA team)

REANA



Analysis Reproducibility with REANA on Kubernetes (Diego Rodriguez - REANA team)

Defining workflows and environments

The creation of workflows and environments usually requires knowledge in specialized technologies

```yaml
- name: prepare
  dependencies: []
  scheduler:
    scheduler_type: 'singlestep-stage'
    parameters:
      model: sm
      parametercard: '{workdir}/param.dat'
      inputpars: defaultparam.yml
    step: {$ref: 'preparestep.yml'}
- name: madgraph
  dependencies: ['prepare','init']
  scheduler:
    scheduler_type: 'singlestep-stage'
    parameters:
      outputlhe: '{workdir}/output.lhe'
      events: {stages: init, output: nevents, unwrap: true}
      paramcard: {stages: prepare, output: parcard, unwrap: true}
    step: {$ref: 'madgraph.yml'}
- name: pythia
  dependencies: ['madgraph']
  scheduler:
    scheduler_type: 'singlestep-stage'
    parameters:
      outputhepmc: '{workdir}/output.hepmc'
      events: {stages: init, output: nevents, unwrap: true}
      lhefile: {stages: madgraph, output: lhefile, unwrap: true}
    step: {$ref: 'pythia.yml'}
```

Three stage workflow

https://yadage.readthedocs.io/en/latest/definingworkflows.html

```dockerfile
# Environment: Jupyter 1.0.0 with IPython 5.0.0 kernel on CentOS7
FROM centos:7

# Set system locale
ENV LC_ALL=en_US.UTF-8
ENV LANG=en_US.UTF-8

# hadolint ignore=DL3033
RUN yum install -y epel-release && yum clean all

# hadolint ignore=DL3033
RUN yum install -y \
      gcc \
      python3-devel \
      python3-pip \
      && yum clean all

# hadolint ignore=DL3013
RUN pip3 install --upgrade pip
RUN pip3 install --no-cache-dir \
      ipython==7.16.1 \
      jupyter==1.0.0 \
      jupyter-client==6.1.12 \
      jupyter-console==6.4.0 \
      jupyter-core==4.7.1 \
      matplotlib==3.3.4 \
      nbconvert==6.0.7 \
      pandas==1.1.5 \
      papermill==2.3.3 \
      # FIXME: Pin black to be able to create the cache folders manually and avoid errors.
      # Related issue: https://github.com/nteract/papermill/issues/498
      black==21.7b0 \
      ipykernel==5.5.5

# FIXME: Create `black` cache folder manually
# hadolint ignore=SC2174
RUN mkdir -m 770 -p /.cache/black/21.7b0/
```

Dockerfile for Jupyter notebooks

https://github.com/reanahub/reana-env-jupyter/blob/master/Dockerfile

Automating workflows

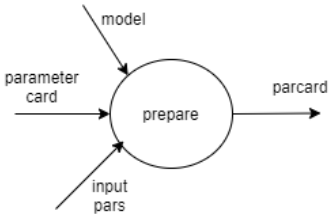
User defines for each stage

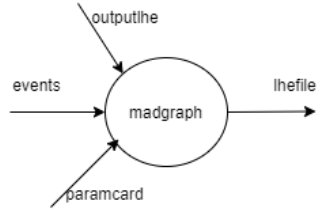
data/code files

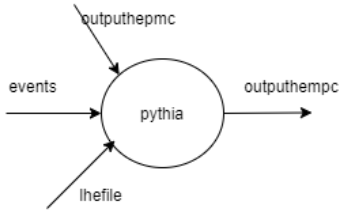
script parameters

expected input values

expected outputs

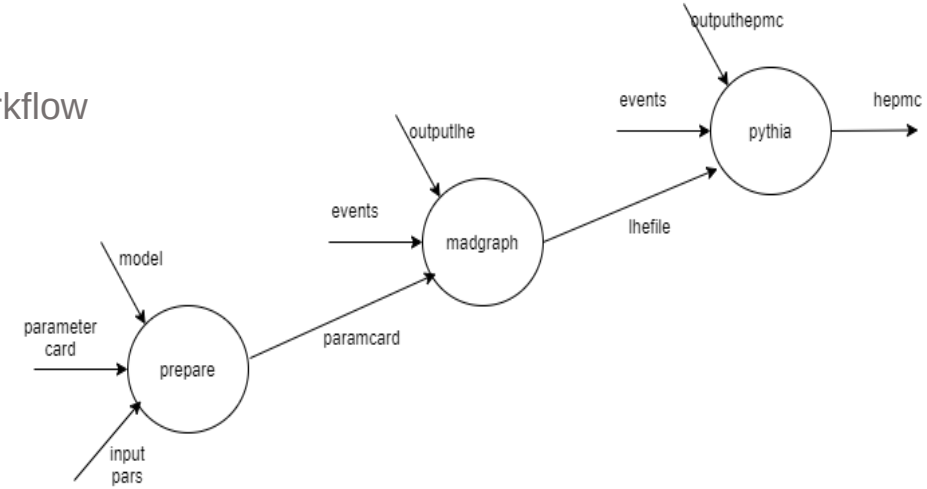






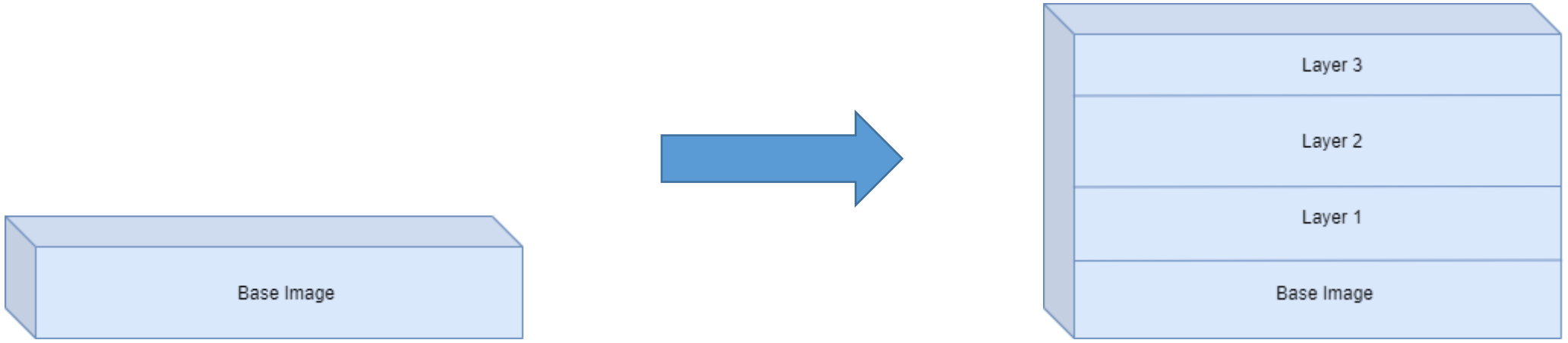
automatically define workflow

(create yaml file)



Extent environments

Add software and packages to an environment without manually changing any files (Dockerfile)



QUESTIONS?

diamantis.patsidis@cern.ch
diampats@ee.duth.gr