

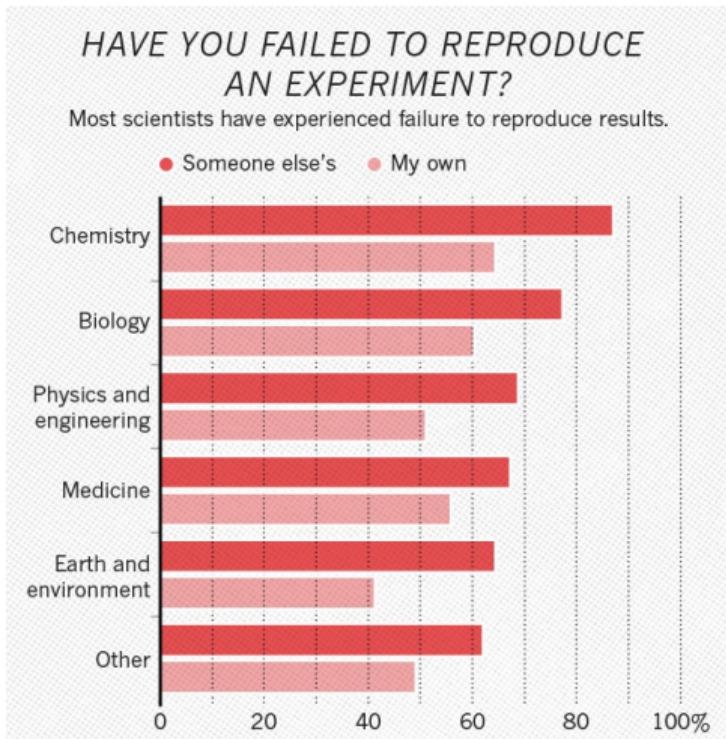
REANA reproducible analysis platform

Tibor Šimko

CERN

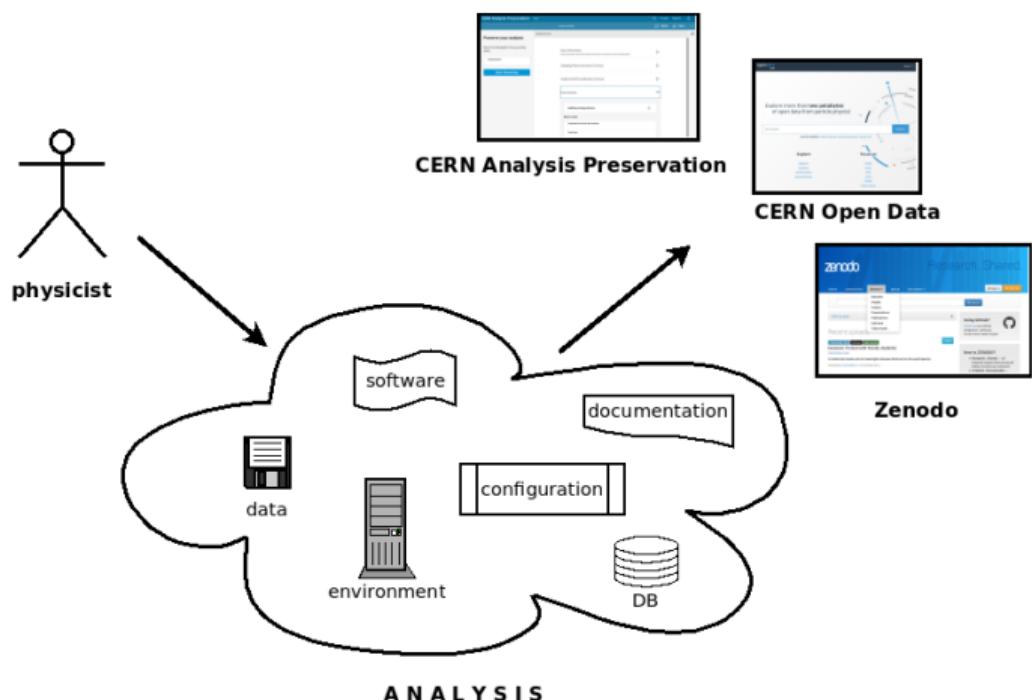
HSF Data Analysis Working Group Meeting, 8 December 2021

Half of researchers cannot reproduce their own results



<https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970>

Preserving analysis knowledge: data, code and more



The FAIR guiding principles for scientific data management

- ▶ Findable
- ▶ Accessible
- ▶ Interoperable
- ▶ Reusable

Capturing analysis assets in digital repositories to facilitate their future **reuse**

Reusability? Repeatability? Replicability? Reproducibility?

The Turing Way model

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

<https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html>

The PRIMAD model

Label	Data	Platform / Stack	Implementation	Method	Research Objective	Actor	Gain
Repeat	-	-	-	-	-	-	Determinism
Param. Sweep	x	-	-	-	-	-	Robustness / Sensitivity
Generalize	(x)	x	-	-	-	-	Applicability across different settings
Port	-	-	x	-	-	-	Portability across platforms, flexibility
Re-code	-	-	(x)	x	-	-	Correctness of implementation, flexibility, adoption, efficiency
Validate	(x)	(x)	(x)	(x)	x	-	Correctness of hypothesis, validation via different approach
Re-use	-	-	-	-	-	x	Apply code in different settings, Re-purpose
Independent x (orthogonal)						x	Sufficiency of information, independent verification

Figure 1 PRIMAD Model: Categorizing the various types of reproducibility by varying the (P)latform, (R)esearch Objective, (I)mplementation, (M)ethod, (A)ctor and (D)ata, analyzing the gain they bring to computational experiments. x denotes the variable primed i.e. changed, (x) a variable that may need to be changed as a consequence, whereas – denotes no change.

https://drops.dagstuhl.de/opus/volltexte/2016/5817/pdf/dagrep_v006_i001_p108_s16041.pdf

From “reproducible” to “reusable” analyses

Four pillars of reusable computational research

I. Input data

What is your input data?

- input files
- input parameters

II. Analysis code

Which code analyses it?

- user code
- software frameworks

III. Computing environment

What is your environment?

- operating system
- database calls

IV. Computational recipes

Which steps did you take?

- shell commands
- notebooks and workflows

I. Data and II. Code

[opendata CERN](#) Search

Help About ▾

Simulated dataset QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6 in AODSIM format for 2012 collision data

/QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6/Summer12_0853k-PU_R01_STARTS3_V7N_v1/AODSIM, CMS collaboration

Cite as: CMS collaboration (2017). Simulated dataset QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6 in AODSIM format for 2012 collision data. CERN Open Data Portal. DOI:10.4843/OPENDATA.CMS.2V17.M2N7

Dataset Standardized Student Model Physics GEN OMW ESD ESDMC

Description

Simulated dataset QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6 in AODSIM format for 2012 collision data.

See the description of the simulated dataset names in: [About CMS simulated dataset names](#).

These simulated datasets correspond to the collision data collected by the CMS experiment in 2012.

Dataset characteristics

3012569 events. 26958 files. 9.6 TB in total.

System details

Recommended [global tag](#) for analysis: STARTS3_V7N_v1

Recommended release for analysis: CMSSW_5_3_32

How were these data generated?

These data were generated in several steps (see also [CMS Monte Carlo production overview](#)):

Step SIM

Release: CMSSW_5_0_0_patch2
Global Tag: STARTS3_V13::All
Generators: pythia6
 Production script ([\[review\]](#))
 Generator parameters ([\[review\]](#)) ([\[link\]](#))
Output dataset: /QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6/Summer12-STARTS3_V13-v1/GEN-SIM

Step HLT RECO

Release: CMSSW_5_3_14
Global Tag: STARTS3_V7N::All
 Production script ([\[review\]](#))
 Configuration file for HLT ([\[link\]](#))
 Configuration file for RECO ([\[link\]](#))
Output dataset: /QCD_Pt_170_250_EMMerchanted_TuneZ2star_8TeV_pythia6/Summer12_0853k-PU_R01_STARTS3_V7N_v1/AODSIM

Data preserved in digital repositories

[zenodo](#) Search

Upload Communities Log In [\[sign up\]](#)

April 26, 2020 Software Open Access

mwasikom/seaborn: v0.10.1 (April 2020)

Mitch Weisert, Doga Bozovic, Joel Corbey, Marc Gefhar, Saku Lohuus, Paul Hobson, David C. Ernzerhof, Tom Augspurger, Yaroslav Hatchenko, John B. Cole, Jordi Wienierevsky, Julian de Ruiter, Cameron Pyke, Stephan Hoyer, Jake Vanderspey, Santi Vilalta, Gergo Kovacs, Eric Quinet, Peter Baumgartl, Marcel Martis, Kyle Meyer, Coban Suer, Alister Miles, Thomas Bruneau, Drew Dimmock, Tim Haysom, Ellen Lee Williams, Constantine Ercan, Clark Pierog, Ben Wittenberg

This is minor release with bug fixes for issues identified since v0.10.0.

- Fixed a bug that appeared with the bootstrapping algorithm on 32-bit systems.
- Fixed a bug where `seaborn` would crash on angular inputs. Now a crash is avoided and regression estimation/plotting is stopped.
- Fixed a bug where `seaborn` would ignore user-specified underline/overline values when rendering a colorstrip.
- Fixed a bug where `seaborn` would use values from the first cell when computing default colorstrip limits.
- Fixed a bug where `seaborn` would raise a warning when trying to plot on a manipulate categorical axis.
- Added a fix to a change in matplotlib that caused problems with single origin plots.
- Added the `showfliers` parameter to `boxplot` to suppress plotting of outlier data points, matching the API of `matplotlib`.
- Avoided seeing an error from statmodels when data with an ID of 0 is passed to `hexbin`.
- Added the `legend.style` keyword to the `plotting_context` definition.
- Deprecated several utility functions that are no longer used internally (`percentiles`, `sig_star`, `perf_size`, and `reset_AF`).

Review

2020-04-26 11:16:16 UTC

mwasikom/seaborn-3c98751

- ✓ `seaborn-3c98751`
 - ✓ `coverage`
 - ✓ `CONTRIBUTING.md`
 - ✓ `gridspec`
 - ✓ `mainmap`
 - ✓ `pyplot`
 - ✓ `LICENSE`
 - ✓ `MANIFEST.in`
 - ✓ `README`
 - ✓ `REACHING.md`
 - ✓ `tests`
 - ✓ `gridspec`
 - ✓ `pyplot`
 - ✓ `tests.py`
 - ✓ `tests_seaborn.py`

142 Bytes 1.3 kB 109 Bytes 145 Bytes 1.4 kB 3.5 kB 87 Bytes 343 Bytes 2.7 kB 256 Bytes 6.8 kB 2.7 kB 270.4 kB 1.9 kB

Publication date: April 26, 2020

DOI: [https://doi.org/10.5281/zenodo.3771777](#)

Related identifiers: Supplement to [https://gitlab.cern.ch/mwasikom/seaborn/tree/v0.10.1](#)

Contributors: Zenodo

License (for file): CC-BY (Other) (Open)

Versions

Version	Date
v0.10.1	Apr 26, 2020
v0.10.0	Jan 24, 2020
v0.9.1	Jan 24, 2020
v0.9.0	Jul 16, 2019
v0.9.1	Aug 3, 2017

Files (205.0 kB)

Name	Size
seaborn-3c98751.zip	205.0 kB

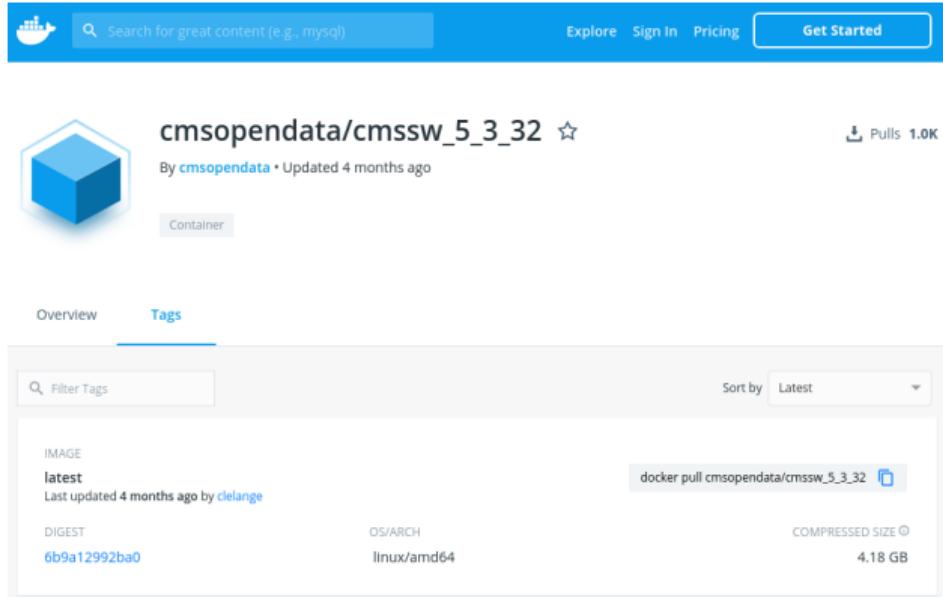
View all 11 versions

Downloads

Show only: Literature (10) Unknown (4) Dataset (8) Software (5) Search

... as is the code

III. Environment



Search for great content (e.g., mysql)

Explore Sign In Pricing Get Started

cmsopendata/cmssw_5_3_32 ☆

By cmsopendata • Updated 4 months ago

Container Tags

Overview Tags

Filter Tags Sort by Latest

IMAGE

latest Last updated 4 months ago by delange

DIGEST

6b9a12992ba0

OS/ARCH

linux/amd64

COMPRESSED SIZE

4.18 GB

docker pull cmsopendata/cmssw_5_3_32

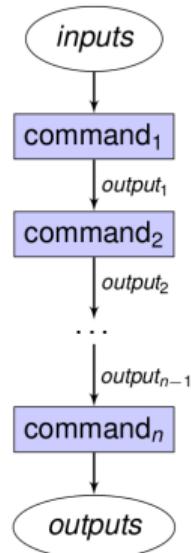
Containerised CMS software framework

```
> ls -l /cvmfs/cms-opendata-condb.cern.ch/
total 1655262
drwxr-xr-x. 2 cvmfs cvmfs          24 Jan 21 2016 FT_53_LV5_AN1
drwxr-xr-x. 2 cvmfs cvmfs          24 Feb 22 2016 FT_53_LV5_AN1_RUNA
drwxr-xr-x. 2 cvmfs cvmfs          366 Jun 21 2017 FT53_V21A_AN6
drwxr-xr-x. 2 cvmfs cvmfs          365 Nov 29 2017 FT53_V21A_AN6_FULL
drwxr-xr-x. 2 cvmfs cvmfs          365 Jun 23 2017 FT53_V21A_AN6_RUNC
drwxr-xr-x. 2 cvmfs cvmfs          3 Oct 20 2017 FT_R_42_V10A
drwxr-xr-x. 2 cvmfs cvmfs          248 Nov 9 2018 START42_V17B
drwxr-xr-x. 2 cvmfs cvmfs          282 Jan 21 2016 START53_LV6A1
drwxr-xr-x. 2 cvmfs cvmfs          394 Jun 21 2017 START53_V27
drwxr-xr-x. 2 cvmfs cvmfs          296 Nov 30 2018 START53_V7N
-rw-r--r--. 1 cvmfs cvmfs 1002414080 Oct 31 2018 102X_upgrade2018_design_v9.db
-rw-r--r--. 1 cvmfs cvmfs 691593216 Oct 31 2018 80X_mcRun2_asymptotic_2016_TrancheIV_v8.db
-rw-r--r--. 1 cvmfs cvmfs 82944 Jan 21 2016 FT_53_LV5_AN1.db
-rw-r--r--. 1 cvmfs cvmfs 82944 Feb 22 2016 FT_53_LV5_AN1_RUNA.db
-rw-r--r--. 1 cvmfs cvmfs 119888 Jun 21 2017 FT53_V21A_AN6.db
-rw-r--r--. 1 cvmfs cvmfs 120832 Nov 29 2017 FT53_V21A_AN6_FULL.db
-rw-r--r--. 1 cvmfs cvmfs 120832 Jun 23 2017 FT53_V21A_AN6_RUNC.db
-rw-r--r--. 1 cvmfs cvmfs 64512 Oct 20 2017 FT_R_42_V10A.db
-rw-r--r--. 1 cvmfs cvmfs 72704 Nov 9 2018 START42_V17B.db
-rw-r--r--. 1 cvmfs cvmfs 84992 Jan 21 2016 START53_LV6A1.db
-rw-r--r--. 1 cvmfs cvmfs 130048 Jun 21 2017 START53_V27.db
-rw-r--r--. 1 cvmfs cvmfs 89088 Nov 30 2018 START53_V7N.db
```

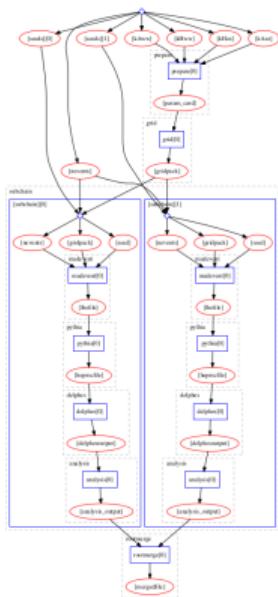
Condition DB snapshot living on
CernVM File System



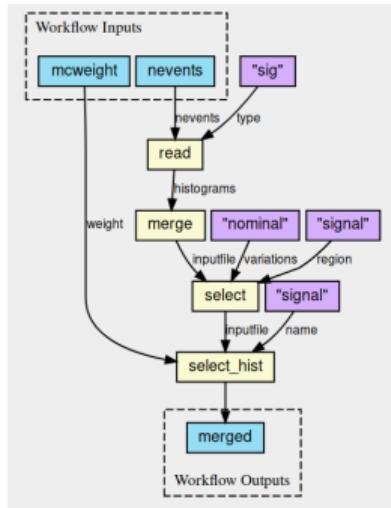
IV. Computational workflows



Serial



Yadage



CWL



Snakemake

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.



Scalable

Support for remote compute clouds.



Reusable

Containerise once, reuse elsewhere. Cloud-native.



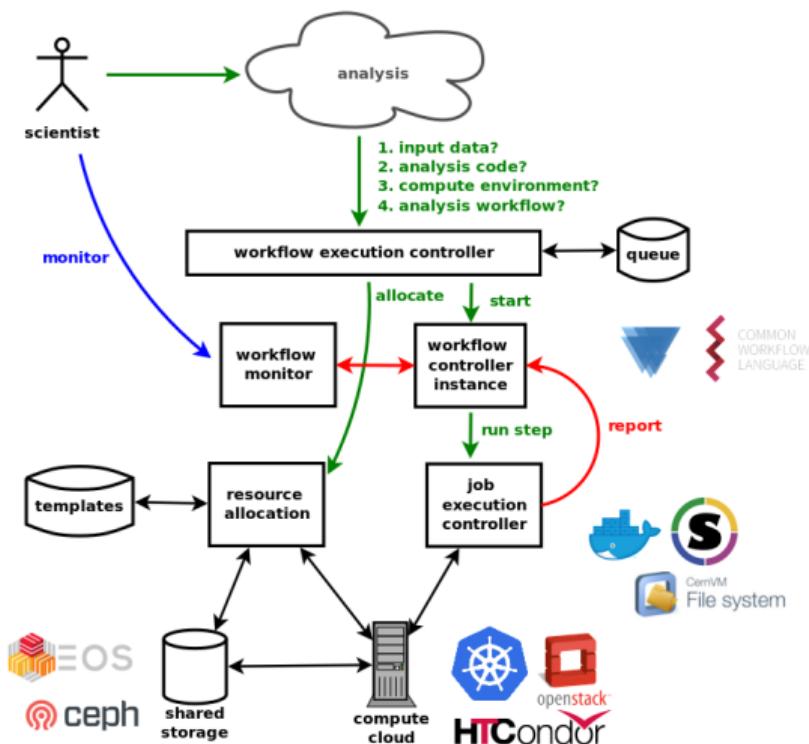
Free

Free Software. GPL licence.
Made with ❤ at CERN.



<https://www.reana.io/>

REANA architecture



Respecting diverse habits of diverse research groups

- ▶ multiple workflow systems
(CWL, Serial, Snakemake, Yadage)
- ▶ multiple container technologies
(Docker, Singularity)
- ▶ multiple compute backends
(Kubernetes, HTCondor, Slurm)
- ▶ multiple shared storage platforms
(Ceph, EOS, NFS)

REANA command-line and web interface

```
1 version: 0.6.0
2 inputs:
3   files:
4     - code/gendata.C
5     - code/fitdata.C
6   parameters:
7     events: 20000
8   data: results/data.root
9   plot: results/plot.png
10 workflow:
11   type: serial
12   specification:
13     steps:
14       - name: gendata
15         environment: 'reanahub/reana-env-root6:6.18.04'
16         commands:
17           - mkdir -p results && root -b -q 'code/gendata.C(${events}),"${{data}}")'
18       - name: fitdata
19         environment: 'reanahub/reana-env-root6:6.18.04'
20         commands:
21           - root -b -q 'code/fitdata.C("${{data}}","${{plot}}")'
22 outputs:
23   files:
24     - results/plot.png
```

The image displays the REANA command-line interface and the REANA web interface. The top part shows terminal sessions:

- A terminal session showing the execution of a workflow script.
- A terminal session showing the status of the workflow, indicating it has started and is currently running.
- A terminal session showing the results of a command to list files in a directory.

The bottom part shows the REANA web interface:

- A "Job Details" page for a workflow named "rootfit" which has completed successfully in 2 min 27 sec.
- A "Workspace" browser showing the contents of the workspace, including files like "gendata.C", "fitdata.C", "results/data.root", and "results/plot.png".
- A "Plot" viewer displaying a histogram titled "Fit example" with a single peak centered around 5.

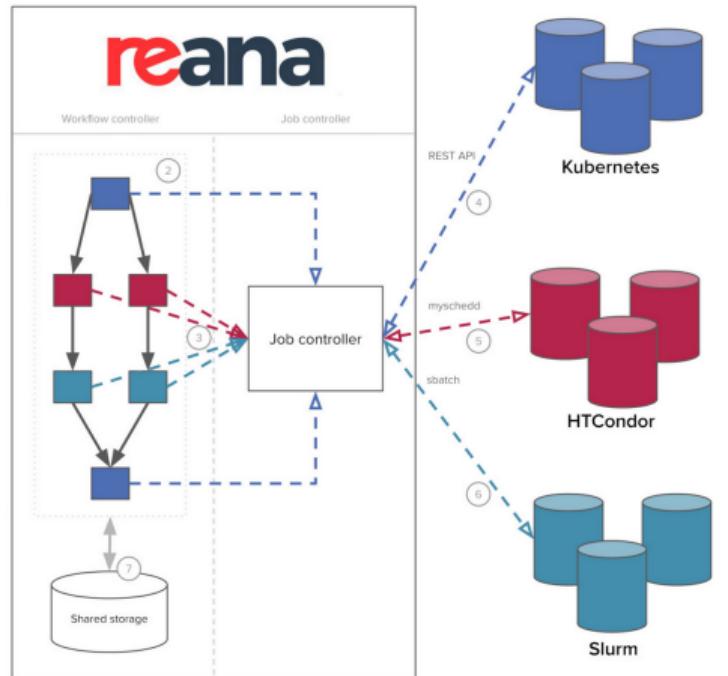
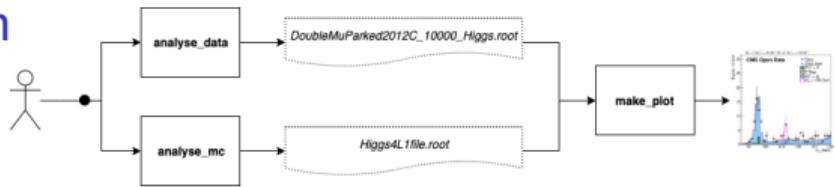
Structure data analysis by means of declarative workflows

Use command-line and web interfaces to run analysis on remote compute clusters

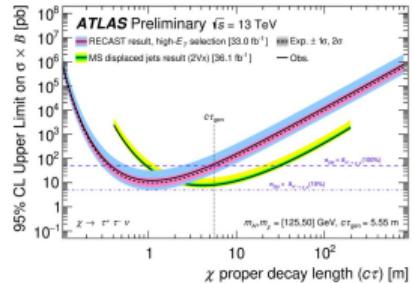
An advantage of declarative approach

```
steps:  
  analyse_data:  
    run: analyse_data.cwl  
    hints:  
      reana:  
        compute_backend: slurmcern  
    out: [DoubleMuParked2012C_10000_Higgs.root]  
  analyse_mc:  
    run: analyse_mc.cwl  
    hints:  
      reana:  
        compute_backend: htcondorcern  
    out: [Higgs4Lfile.root]  
  make_plot:  
    run: make_plot.cwl  
    hints:  
      reana:  
        compute_backend: kubernetes  
in:  
  DoubleMuParked2012C_10000_Higgs: >  
    analyse_data/DoubleMuParked2012C_10000_Higgs.root  
  Higgs4Lfile: >  
    analyse_mc/Higgs4Lfile.root  
out: [mass4l_combine_userlvl3.pdf]
```

A three-step CWL hybrid workflow

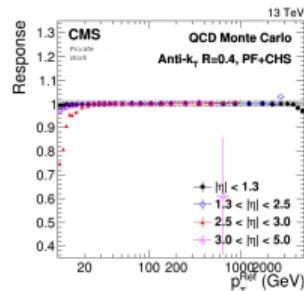
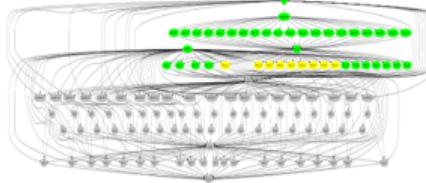


Data analysis and data production examples



ATLAS <https://cdsweb.cern.ch/record/2714064>

Data analysis example: ATLAS displaced jet search reinterpretation



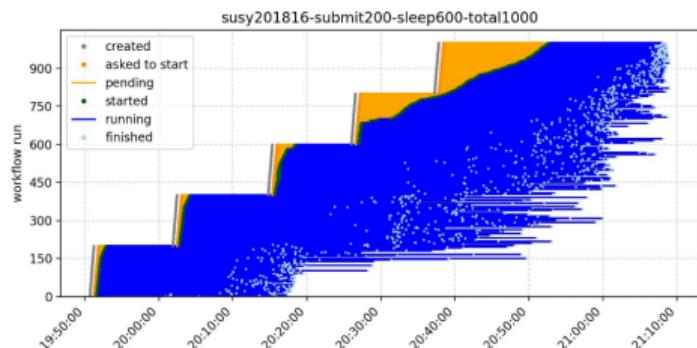
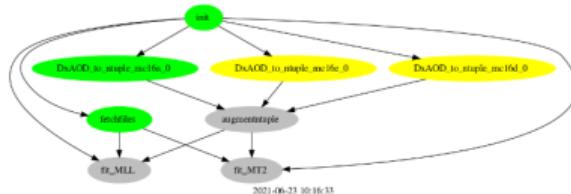
CMS <https://github.com/alintulu/reana-demo-JetMETAnalysis>

Data production example: CMS jet energy resolution and corrections

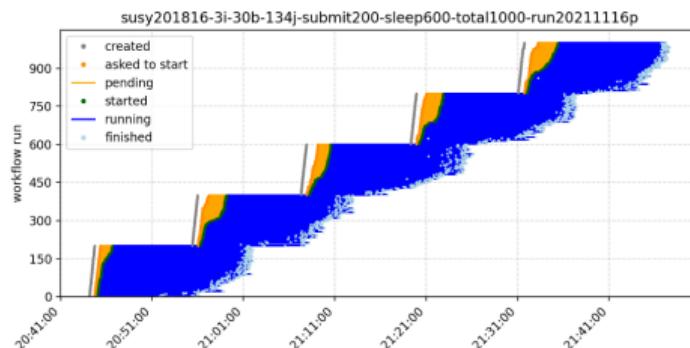
Scalability: running 100k ATLAS pMSSM workflows

ATL-SUSY-2018-16 analysis:

- ▶ NoSys: O(10 minutes); "test" payload
- ▶ AllSys: O(10 hours); "real" payload



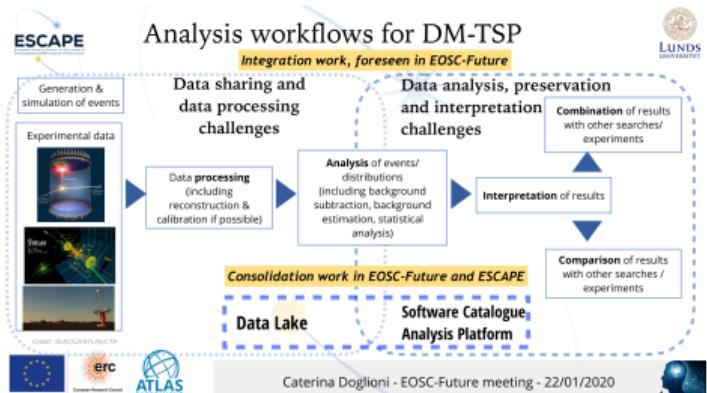
Old cluster (448 cores)



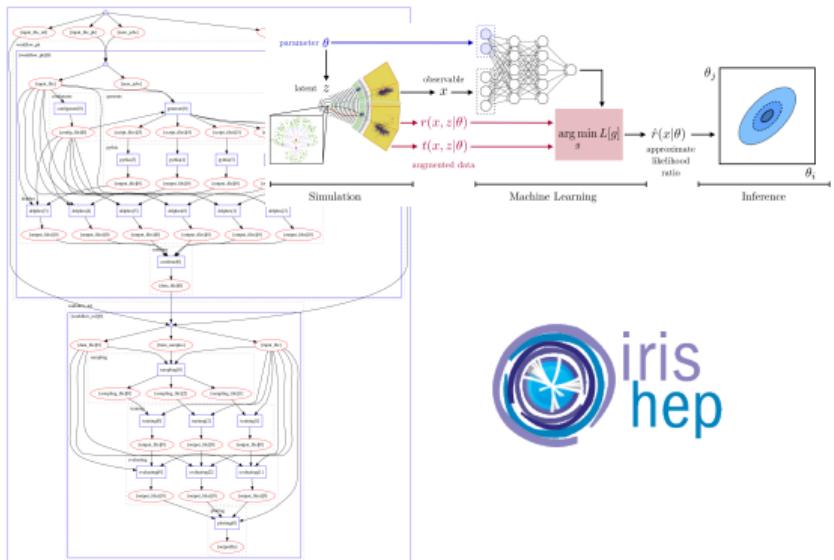
New cluster (1072 cores)

Submitting 200 NoSys analyses every 20 minutes

Examples of other recent activities



ATLAS Dark Matter searchers



MadMiner ML workflows

REANA installations

Release	Created
0.8.0-alpha.1	2020-11-25T14:44:01.074Z
0.7.2	2021-02-04T17:45:36.640Z
0.7.1	2020-11-11T09:40:55.784Z
0.7.0	2020-10-21T09:05:18.049Z
0.7.0-alpha.2	2020-10-05T14:25:00.271Z
0.7.0-alpha.1	2020-08-14T16:28:38.333Z

Helm makes it easy to install REANA at scale

Workloads [REFRESH](#) [DEPLOY](#) [DELETE](#)

Cluster: reana Namespace: default [RESET](#) [SAVE](#) [BETA](#)

Workloads are deployable units of computing that can be created and managed in a cluster.

[Filter workloads](#)

Name ↑	Status	Type	Pods	Namespace	Cluster
reana-cache	OK	Deployment	1/1	default	reana
reana-db	OK	Deployment	1/1	default	reana
reana-message-broker	OK	Deployment	1/1	default	reana
reana-server	OK	Deployment	1/1	default	reana
reana-traefik	OK	Deployment	1/1	default	reana
reana-workflow-controller	OK	Deployment	1/1	default	reana

 Google Cloud

REANA on Google Cloud



ILLINOIS
NCSA | National Center for
Supercomputing Applications

REANA on US supercomputers

Sociology challenges: adopting containerised workflow paradigm

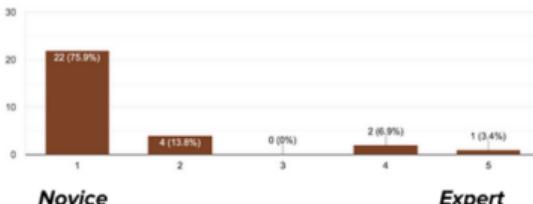


ATLAS/CMS analysis preservation workshop



Before

I am confident I can write a containerized workflow that can run my full analysis on the cloud.
29 responses

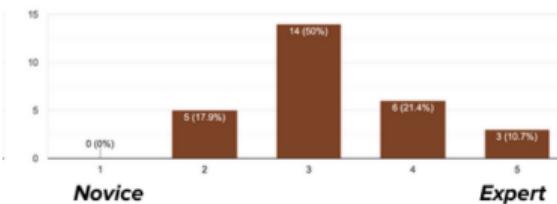


Novice

Expert

After

I am confident I can write a containerized workflow that can run my full analysis on the cloud.
28 responses



Novice

Expert

"Prereproducible" analyses

Nature 557 (2018) 613

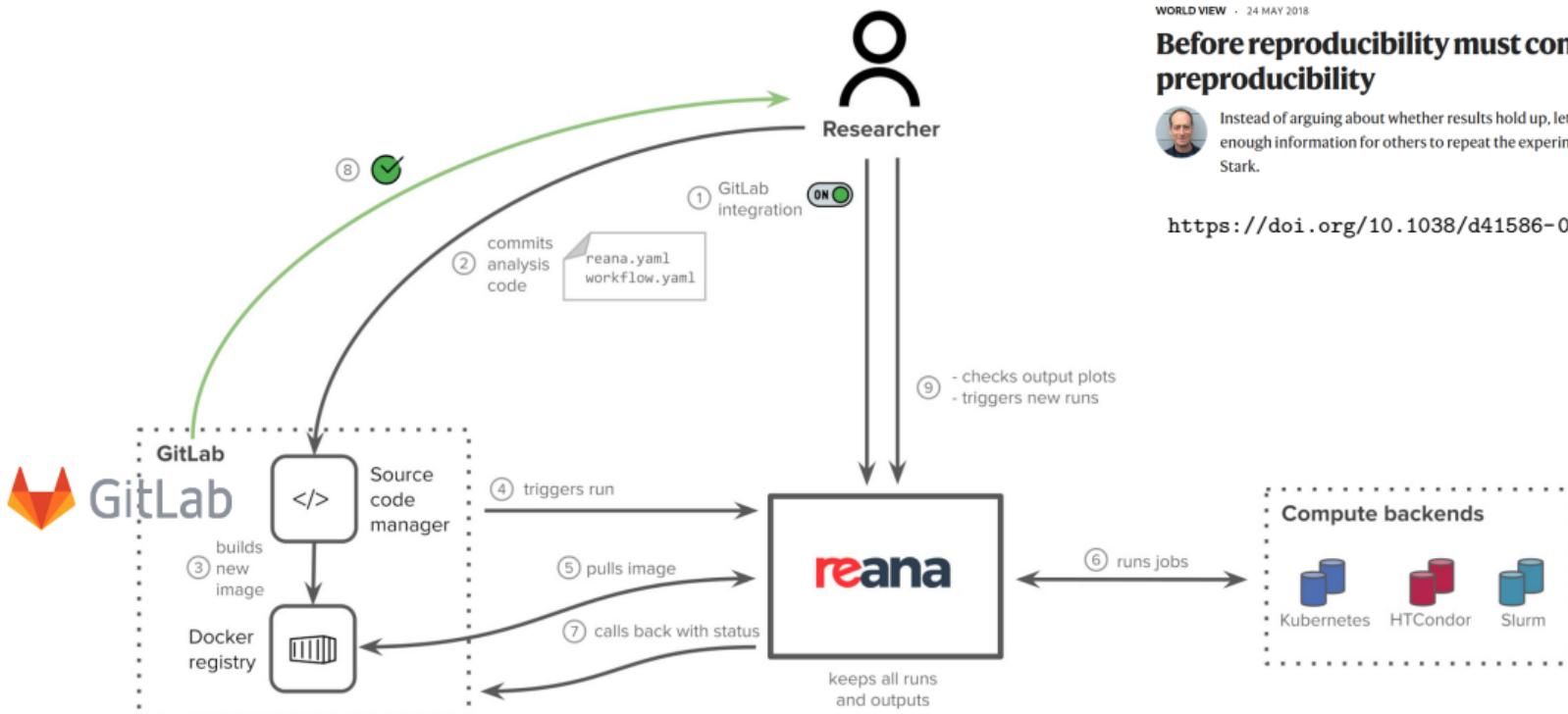
WORLD VIEW · 24 MAY 2018

Before reproducibility must come prereproducibility



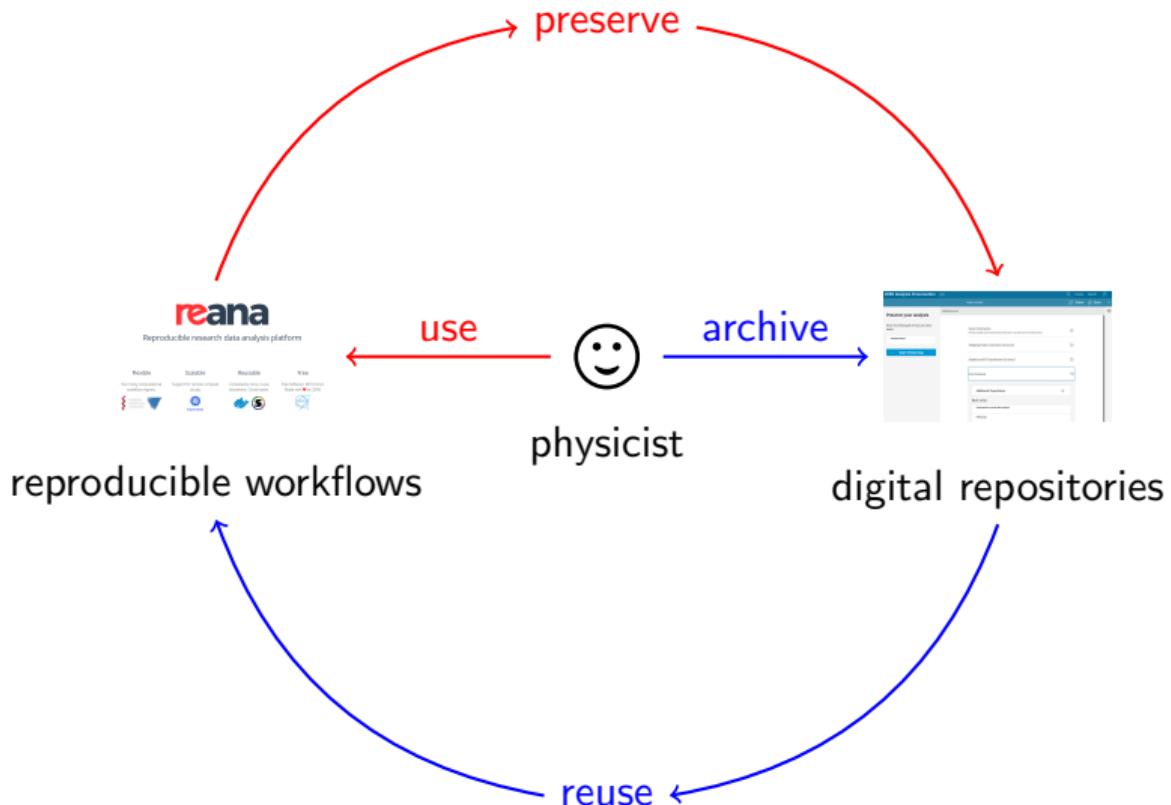
Instead of arguing about whether results hold up, let's push to provide enough information for others to repeat the experiments, says Philip Stark.

<https://doi.org/10.1038/d41586-018-05256-0>



Driving prereproducibility via Continuous Integration with source code management systems

Reproducibility \rightleftharpoons Preservation



Conclusions

- ▶ driving reuse through preproducibility
- ▶ data + code + environment + workflow
→ reproducible analyses
- ▶ technology challenges: large containers, complex computational workflows
- ▶ sociology challenges: declarative programming, paradigm shifting, publish-or-perish culture
- ▶ synergies with computational reproducibility needs in astronomy, life sciences

