Data and analysis preservation at LHCb

Concezio Bozzi

INFN Sezione di Ferrara

CERN, November 26th 2019

Gratefully acknowledging: S. Neubert, S. Roiser, A. Pearce, B. Couturier, C. Burr, F. Redi, H. Cliff, D. Fitzgerald, C. Aidala

Outline

- Data and analysis preservation
  - Roadmap defined in internal LHCb document (LHCb-INT-2017-021)
  - Minimal requirements and best practices
  - Resources
- Open data
  - LHCb Open Data policy
    - LHCb collaboration (2013). LHCb External Data Access Policy. CERN Open Data Portal. DOI:10.7483/OPENDATA.LHCb.HKJW.TWSZ
  - Status and outlook

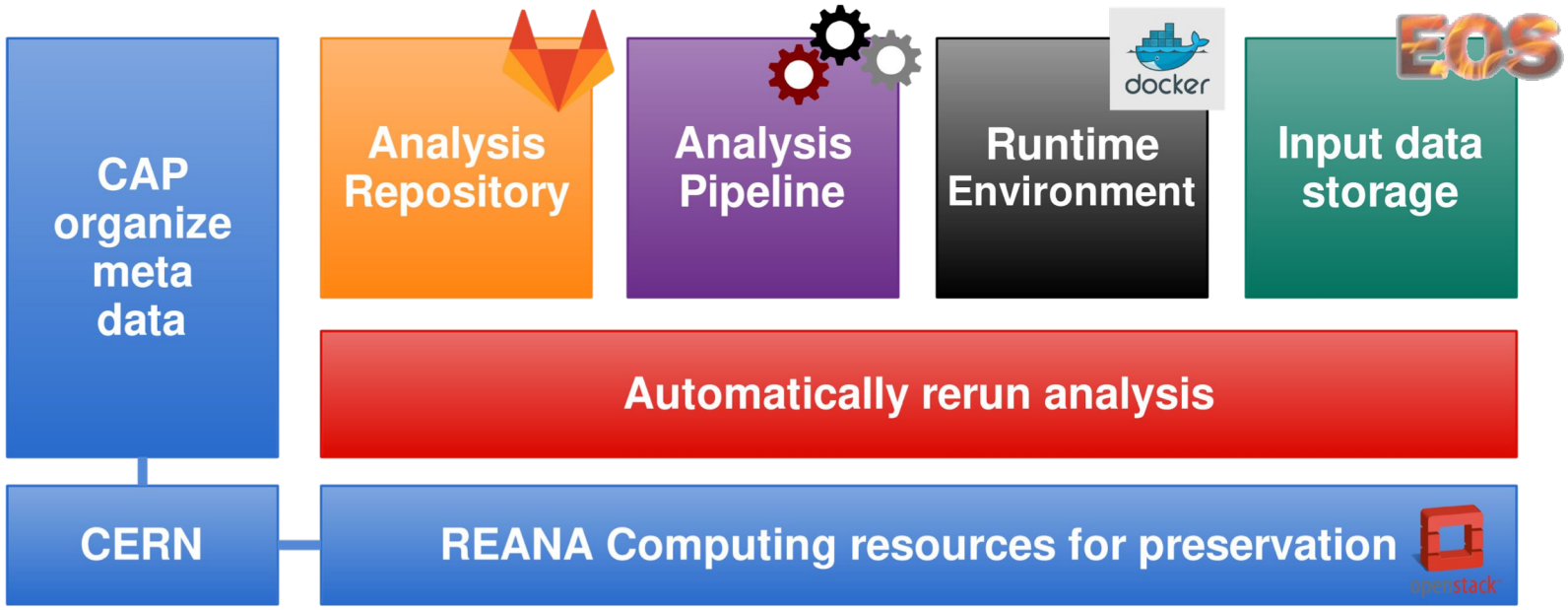
Analysis preservation: Objectives

- Collaborative working
  - allow members of analysis team to access the full software
  - Continuous development and integration
  - Newcomers: improve learning experience, get productive faster
- Knowledge preservation during review
- Knowledge transfer to other analysis teams
  - Analysis updates
  - Tools sharing
- Knowledge transfer to «future generations»
  - Reuse and testing of new models

Analysis preservation: scope

- Active reproducibility
  - Analysis preservation means analysis re-use
  - Needs running code and its environment
    - Inspired by continuous integration
- Partial reproducibility
  - Typical two-stage process:
    - centralised production from RAW data or MC simulation to filtered DST/ntuples
    - Highly tailored, often hand-crafted processing by single analysts
  - Link between the two is common file format (typically: ROOT ntuples)
  - Physicists decoupled from LHCb software stack and using «ecosystem» of very specialised software
    - Lightweight file format can be shared easily between physicists
    - **reasonable starting point for analysis preservation**
  - Preservation of centralised data production accomplished by archiving data provenance stating how the datasets were produced
    - **DIRAC «steps» and «transformations» are always preserved**
    - Recent introduction of **containers** (Singularity) allows to preserve old (e.g. SLC5) workflows

Four domains of best practices



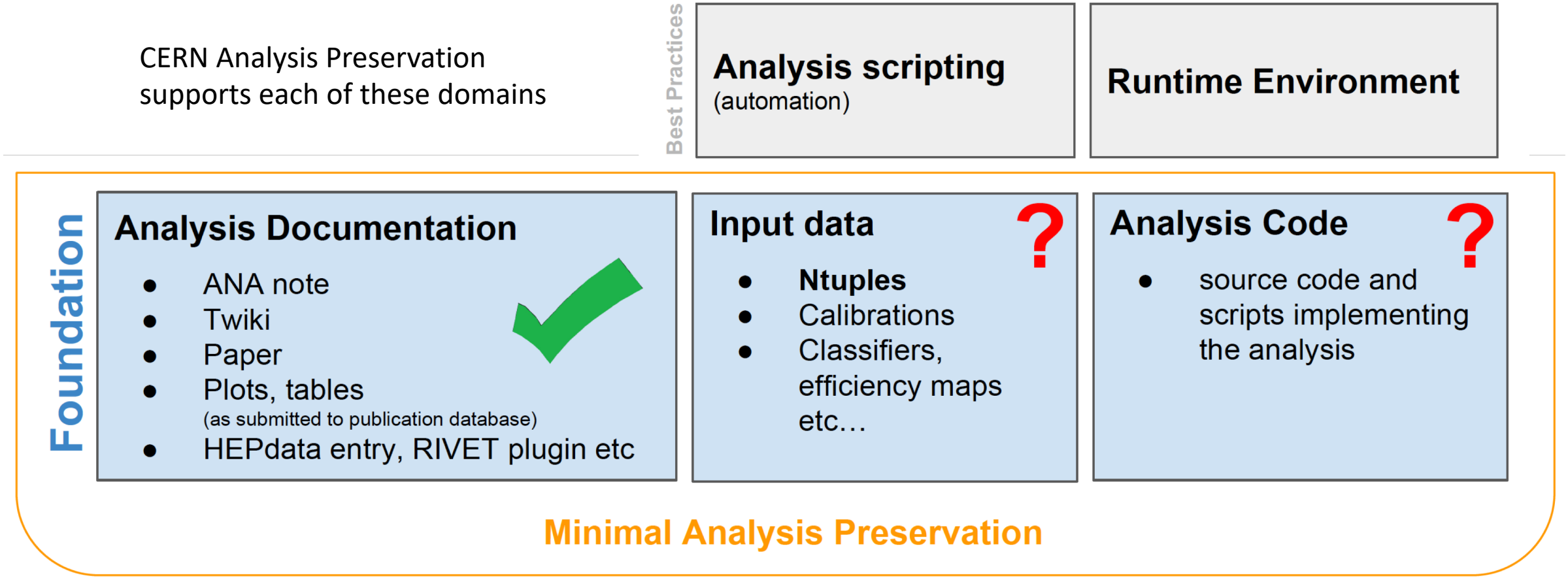
- **Analysis preservation can be adopted step-by-step**
- Synergies between the domains

Overview of recommendations

Domain	Minimal	Practices needed for CAP	Best Practices
Repository	Complete analysis code on gitlab.cern.ch readable by collaboration		Project is hosted in WG gitlab-group ANA and results in a separate repository Use fork&merge workflow Use submodules or subtrees for modularization Use gitlab CI
Pipeline	Full instructions to run analysis documented in ANA	Script to rerun analysis in batch mode producing every plot and table parametrize input data location	Dedicated pipeline tool with dependency-checking and caching Use pipeline early in analysis design Use same pipeline local and in cloud Use gitlab CI
Input data	All input data on EOS readable to collaboration		Input files in WG directory on EOS Code to produce input data accessible through analysis repo
Runtime environment	Documented installation instructions Setup script	Docker image containing everything the analysis needs, hosted in gitlab registry at CERN Dockerfile in analysis repo Specify resources needed to run container	Use LHCb provided base image Use package manager inside Dockerfile Use docker container in analysis CI

Minimal analysis preservation

CERN Analysis Preservation supports each of these domains



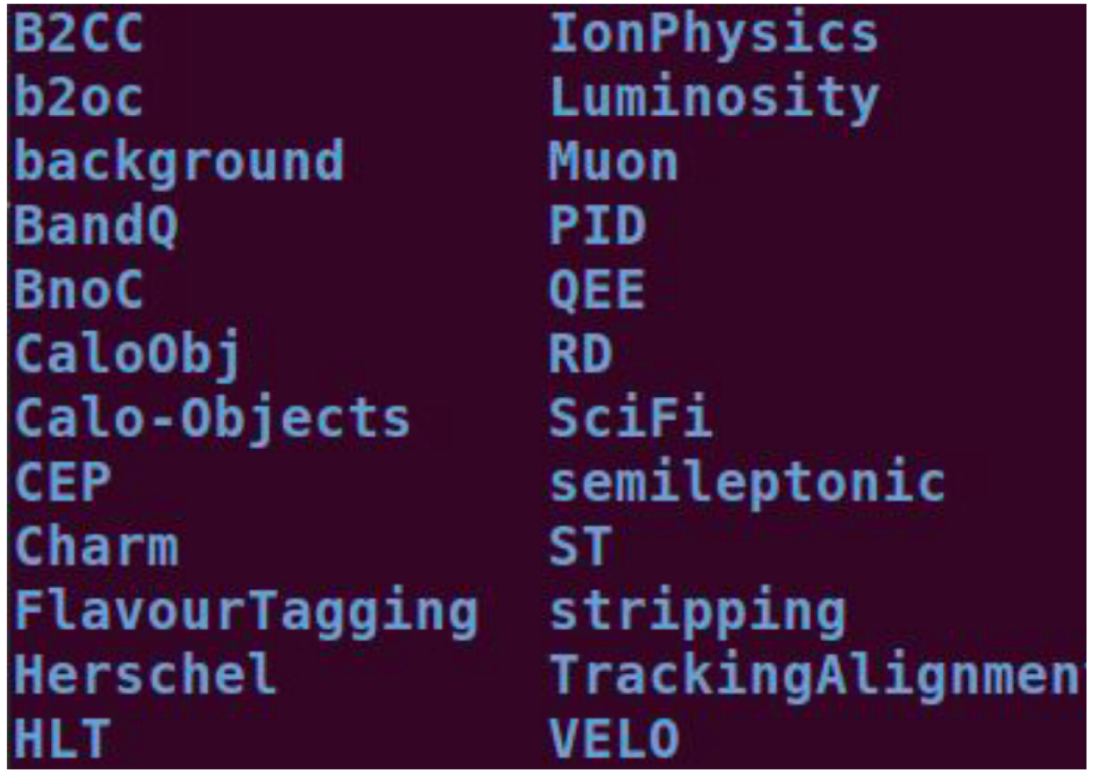
Minimal analysis preservation

- As of December 2017 LHCb has adopted a minimal set of mandatory AP practices as part of the internal analysis review process.
- Analysis code:
  - Source code and scripts are **made available in a git repository** at gitlab.cern.ch
  - Repository contains instructions to run the code
    - Including how to install runtime environment
  - Access regulated through CERN e-groups using WG gitlab groups
- Input data:
  - nTuples are **stored on EOS, using WG directories**, or
  - nTuples obtained by centrally-run Working Group Productions
    - **managed in the LHCb bookkeeping** and available to the entire collaboration

Existing gitlab groups and WG directories

- Gitlab groups (one per WG)
  - Bottomonium and Quarkonium
    - B&Q exotics
  - RD: Rare Decays
  - B2OC: b decays to Open Charm
  - B2CC: b decays to Charmonia
  - Charm
  - BnoC: charmless b decays
  - QEE: QCD & Electroweak physics
  - SLB: semileptonic b decays
  - IFT: Heavy-ion and fixed target

WG space: /eos/lhcb/wg



$\Delta A_{CP}(\Lambda_c^+)$

lc2pxx-dacp

Project ID: 6829

Star 1 | Fork 0 | Clone

MIT License | 511 Commits | 5 Branches | 0 Tags | 1.5 MB Files

master | lc2pxx-dacp / | History | Find file | Web IDE

README.md

lc2pxx-dacp

This is the repository for the analysis code for the Run 1 Lambdac ΔACP analysis. More information on the analysis in general can be found on the corresponding TWiki page.

The most important parts of the repository are: an `lc2pxx_dacp` folder which contains a Python module; and an `analysis` folder which contains scripts the use the module. All the output is stored in the `output` folder.

Running the analysis

The [`run_all.sh`][run_all.sh] script will run everything that's needed for the analysis note. It can take a couple of hours to complete.

```
$ ./run_all.sh
```

The individual Python scripts in the `analysis` folder, which are run in a particular order by `run_all.sh` to produce the final results, can be run individually. For example:

```
$ ./run python analysis/download_data.py --help
usage: download_data.py [-h] -m {LcTopKpi,LcTopKK,LcToppipi} -y {2011,2012} -p
                        {MagUp,MagDown,MagBoth} [-v]
```

Setting up the dependencies

The analysis dependencies are mostly Python packages, along with the dependencies of those packages. This amounts to a reasonably up-to-date set of build tools, like a compiler, and a Unix-like OS. ROOT is required for loading and saving the data. A recent version is better, such as 6.06/02.

There are a couple of ways the dependencies can be installed.

On your machine

The Python packages upon which this code depends can be installed with pip, a Python package manager.

```
$ pip install --user numpy matplotlib scipy hep_ml uncertainties tqdm jug tables
$ pip install --user mcerp
$ pip install --user git+git://github.com/ibab/root_pandas@0cc031f5fa01ff3cfe78fl
```

Using Docker

The analysis can also be run inside a Docker container. This removes the burden of satisfying the dependencies from the user. To setup and run the container, you need to have installed Docker and have the Docker daemon running.

```
# Inside this repository
$ mkdir -p output/{data,luminosity,fits,plots,tables}
$ docker build -t lc2pxx-dacp .
$ docker run -i -v `pwd`:/work -e LOCAL_USER_ID=`id -u $USER` -e LOCAL_GROUP_ID=`
```

This builds a container with the name `lc2pxx-dacp`, using the definitions in the [Dockerfile][Dockerfile] in this repository, and then runs the command `bash run_all.sh` inside the container.

Going beyond minimal AP

- Automated analysis workflows
  - Dedicated lesson on **Snakemake** in the **LHCb Starterkit**

Tutorial

Snakemake allows you to create a set of rules, each one defining a "step" of your analysis need to be written in a file called `Snakefile`. For each step you need to provide:

- The *input*: Data files, scripts, executables or any other files.
- The expected *output*. It's not required to list all possible outputs. Just those that you want to monitor or that are used by a subsequent step as inputs.
- A *command* to run to process the input and create the output.

The basic rule is:

```
rule myname:
    input: ['myinput1', 'myinput2']
    output: ['myoutput']
    shell: 'Some command to go from in to out'
```


Analysis automation with snakemake

Learning Objectives

- Learn what analysis automation is and how it helps with analysis preservation
- Learn how to create a pipeline with Snakemake

Motivation

"The Snakemake workflow management system is a tool to create reproducible and scalable data analyses"

- A workflow management system allows you to:
  - Keep a record of how your scripts are used and what their input dependencies are
  - Run multiple steps in sequence, parallelising where possible
  - Automatically detect if something changes and then reprocess data if needed
- Using a workflow management forces you to:
  - Keep your code and your locations in order
  - Structure your code so that it is user-independent
  - Standardise your scripts
    - Bonus: Standardised scripts can sometimes be used across analyses!

Documentation and installation

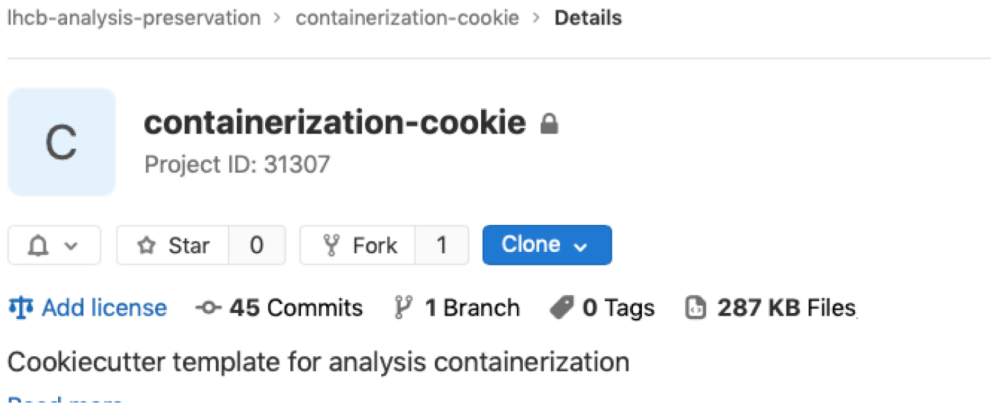
You can find full documentation for Snakemake at this link, you can also ask any questions you have on the ~reproducible channel on mattermost.

Snakemake requires Python 3, if you already have this available it can be easily installed using pip:

```
python3 -m pip install --user snakemake
# Depending on your PATH variable you may also need to use:
alias snakemake='python3 -m snakemake'
```

Going beyond minimal AP

- Containerisation template available in gitlab
  - template allowing easy setup of a new gitlab repository with automatic containerisation and continuous integration, including safe data access on EOS



Thanks a lot to Lukas Heinrich @lheinric for pointing out cookiecutter.

Purpose

This template will **enable automatic containerization and continuous integration** on your analysis repository. It also demonstrates **how to access data on EOS from a cloud service** using a save authentification mechanism.

You will get an executable Docker image containing your analysis, which can be directly handed to analysis preservation. No setup work has to be done by the analysts. By default you will get a full ROOT 6.11 setup. Additional, custom software (e.g. scikit-learn, the GSL, snakemake etc) can be easily added. Instructions below.

Using this template

To initialize your repository with this template, follow these steps:

- Create a gitlab access token by navigating to your user settings menu (click on your avatar). Make sure the token covers the scope "api".
- On you local machine Install cookiecutter: `pip install cookiecutter`
- Download template and fill in your information: `cookiecutter ssh://git@gitlab.cern.ch:7999/lhcb-analysis-preservation/containerization-cookie.git`
- cookiecutter will ask you to fill in some basic information about the analysis and provide your user name and access token
- You will be given the option to create the repository on gitlab.cern.ch for you. If you decline the template will only be created locally and in that case you do not need the gitlab token
- It will create a new project on gitlab (or locally) and initialize it with the material in this template
- You still need to add authentification to access data on EOS as described below

If you are using the option to create the project on gitlab you immediately will note a gitlab-ci runner starting to build your docker image and test it with the `HelloWorld` example. While the docker build should work out of the box, the test will fail on the first try. The reason is that this test demonstrates how to access data on EOS, using a keytab belonging to the `lbanadat` service account. However, for security reasons no secrets are part of the template. Instead follow these steps:

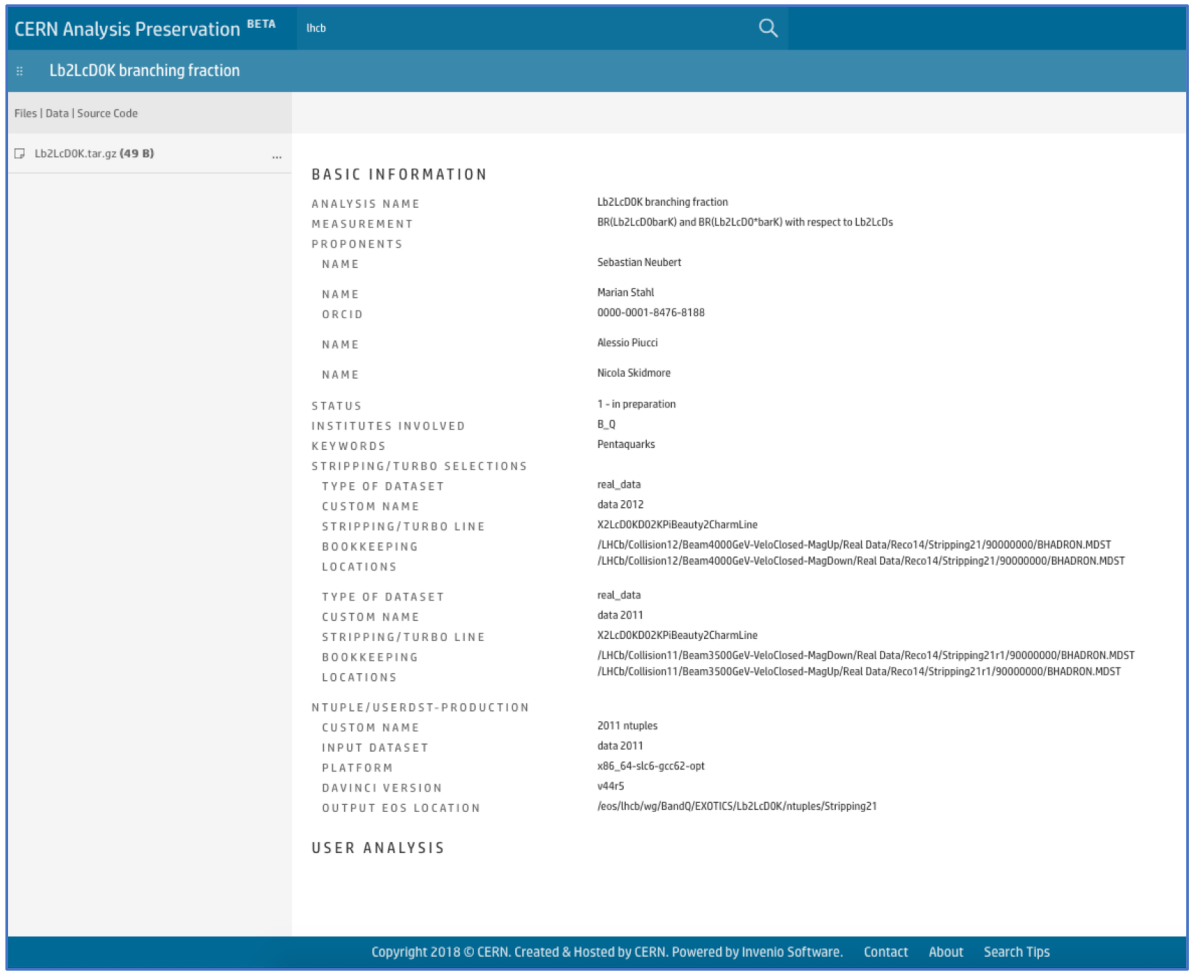
- In you newly created gitlab project go to the Menu `Settings --> CI/CD --> Secret Variables`
- Add a new secret variable with the variable key `ENCODED_KEYTAB`
- Copy the contents of the text file `/eos/lhcb/wg/BandQ/Test/anadat.txt` into the variable value
- Go to the menu `CI/CD` and restart the `test` stage of the CI pipelinem, which should now work.

Watch a video how to make a new analysis repo: https://asciinema.org/a/fQcyG7Pvjsil5ouCpXFoVS98B
To run the containerized analysis: https://asciinema.org/a/0gfEE1IhMfSQm7C9nvNDcDZKf

You can inspected available images for your analysis in the `Registry` tab.

As soon as the image is build you can run the analysis inside docker as described below.

Testing CAP and REANA



REANA example - LHCb Rare Charm Decay Search

gitter join chat license MIT

About

This analysis example attempts to reproduce LHCb rare charm decay study published in Phys. Lett. B 724 (2013) 203-212.

$$D^+_{(s)} \to \pi^+\mu^+\mu^-$$

These decays are very rare, which makes their observation extremely challenging. The LHC produces a lot of charm particles `D`, but it also produces a much greater number of other particles which can be mistaken for the signal. It is necessary to develop an effective strategy to identify the signal events in the large data sample. The event selection strategy is implemented in three stages: the trigger selection, stripping selection and the multivariate analysis.

In the paper, you can lean more about the theoretical background and motivation to study this decay.

This example is based on this analysis-case-study.

Analysis structure

Making a research data analysis reproducible basically means to provide "runnable recipes" addressing (1) where is the input data, (2) what software was used to analyse the data, (3) which computing environments were used to run the software and (4) which computational workflow steps were taken to run the analysis. This will permit to instantiate the analysis on the computational cloud and run the analysis to obtain (5) output results.

1. Input data

The data being used is currently only available on request. It consists of a 10 GB ROOT file.

2. Analysis code

You can straightforwardly run the analysis step by step as explained below:

```
$ mkdir -p results/tmp && mkdir -p logs
$ root -b -q 'Optimise.C("${data}", "results")' | tee logs/optimise.log
$ root -b -q 'ModelFixing.C("${data}", "results/tmp/PhiModels.root")' | tee logs/model_fixing.log
$ root -b -q 'OSMassFit.C("${data}", "results/tmp/PhiModels.root", "results")'  | tee logs/massfit.log
```

Resources

- Current usage of EOS WG space: 340 TB!
  - Way more than AP data
  - Including material from past ~454 analyses that were not in AP
  - Estimate O(2TB) per analysis on Run1+Run2 data
  - LHCb publishes ~60 papers per year
- During the development of an analysis, expect significant resources needed to test preservation and set up continuous integration
  - Managed per working group
  - Estimate 50CPU cores per WG, 4GB RAM/core
  - Provisioned with cloud resources on the CERN infrastructure?

Open data policy DOI:10.7483/OPENDATA.LHCb.HKJW.TWSZ

- Adopted by the LHCb Collaboration Board on Feb 27th 2013

Data Access Policy for LHCb

1. Data preservation is fundamentally important for the collaboration itself, regardless of any external requirements. This is to enable collaboration members to access data for many years after it was taken and requires a consistent set of the data, associated software, metadata and conditions and documentation to be preserved. LHCb will seek to develop such a data preservation capability as soon as practical. We will need to identify additional resources for this.

2. LHCb supports the principle of open access. In principle we can envisage providing some such open access based upon the work needed internally for data preservation (point 1 above).

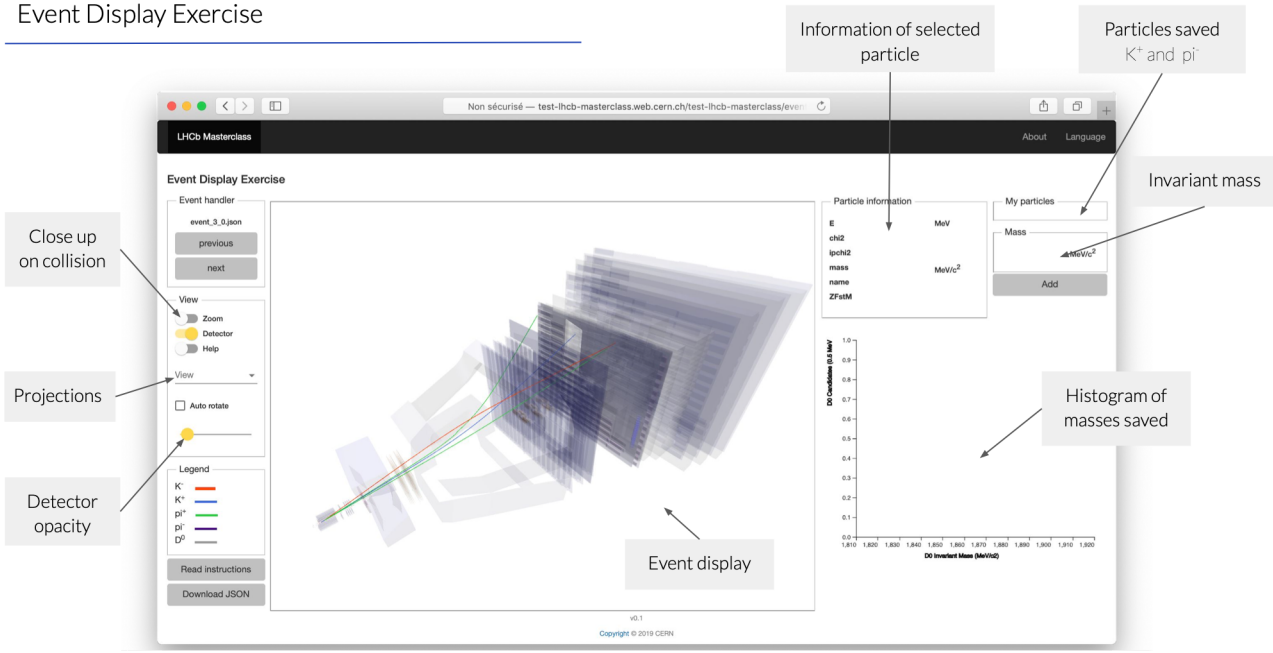
Open data policy DOI:10.7483/OPENDATA.LHCb.HKJW.TWSZ

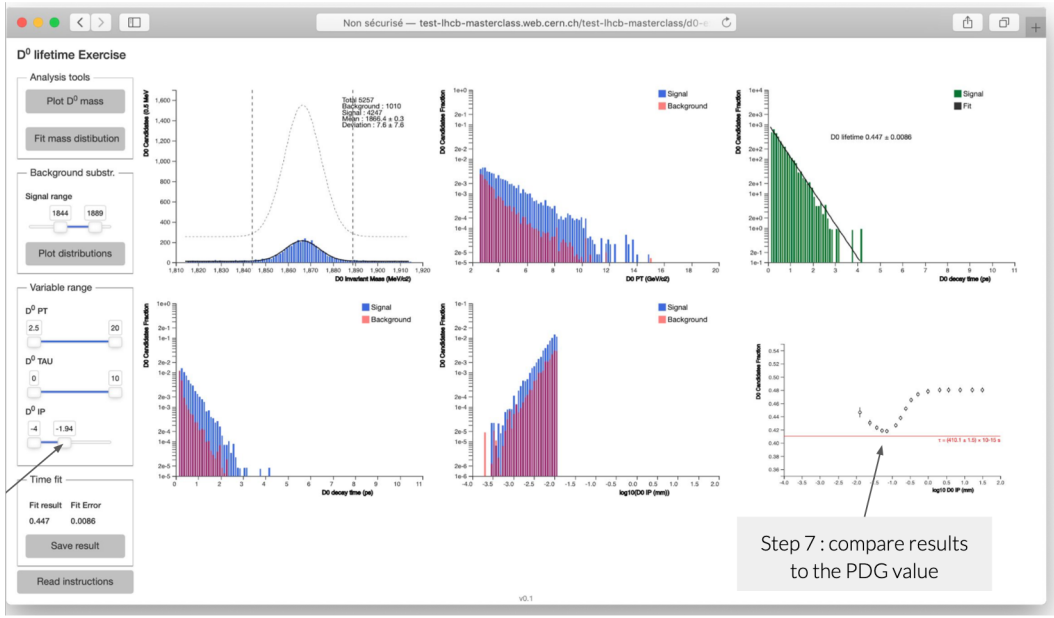
- Adopted by the LHCb Collaboration Board on Feb 27th 2013
- Open data for Outreach and education:
  - »… LHCb already participates in outreach activities and will continue to do so. […] **The data are provided for educational purposes only**, and are not considered suitable for publication. Only a limited fraction of the complete LHCb data-set may be used. »
- Open data for Research:
  - «… **access will be granted to portions of the DST data five years after data is taken**. The portion of the data which LHCb would normally make available is 50% after 5 years, rising to 100% after 10 years. All requests will be considered by the CB and the period and proportion may be varied for specific requests.»
- Big caveat
  - «… **LHCb is extremely resource limited** at present. Therefore whilst this policy expresses a spirit of intent, **we cannot commit to implementation of any capability on any specific timescale**. Specifically in respect of open access we will not be able to undertake any significant development to support this without injection of additional resources. »

Pragmatic approach

Open data initiatives limited to **outreach and education**

- International physics masterclasses
  - Virtual machine image (being replaced by web application)
  - Getting started with LHCb Open Data
  - Event files for event display
  - Event file for D0 lifetime measurement
- Matter-antimatter differences (B meson decays to three hadrons)
  - Data file
  - Project notebook
- **New initiatives** on CP violation in charmless two-body decays and kaon decays for high-school students





Pragmatic approach

A test case of open data for research:

- static, open access jet database/dataset along with all necessary tools and documentation for conducting analyses
- provide access to reconstructed data (DSTs and/or Ntuples)
- main target: theorists and phenomenologists, such that we can have more analyzers to help advance the field of QCD
- Propose to use containers
- Funding proposal submitted

Open Data: storage resources

- Required storage depends on whether we will publish
  - FULL.DST, i.e. the outcome of the event reconstruction, or
  - (μ)DST, i.e. the outcome of the selections aka «stripping»
- For all Run1 data:
  - FULL.DST: 500TB
  - (μ)DST: 120TB
- MC samples: same order of magnitude
- **Current Open Data proposals require much less**

Conclusion

- Data and analysis preservation
  - **Minimal recommendations** defined; all analyses entering review must comply with them
  - **Best practices** needed to **interface to the CAP** infrastructure have been identified
  - **Further best practices** that provide guidance how to profit most from the adoption of AP have been proposed
- Open data
  - External data access policy identified
  - **Pragmatic approach**: limited person-power, use well-crafted proposals from a few LHCb institutes to steer the developments