Workflow Management

in HEP and Belle II Neutral B Mixing Analysis

Caspar Schmitt

Joint Seminar, LMU/MPP

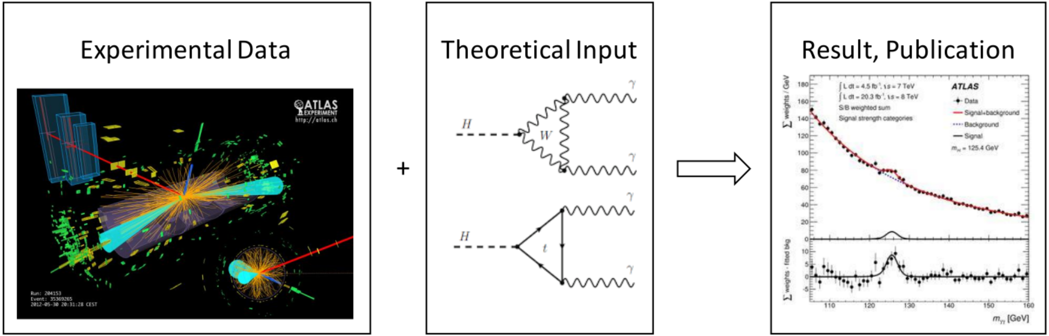
December 21, 2022

Overview

HEP Analysis Workflow

A **workflow**: a top-level entity of workload to accomplish a scientific objective.

Generally in HEP, we most have a workflow similar to



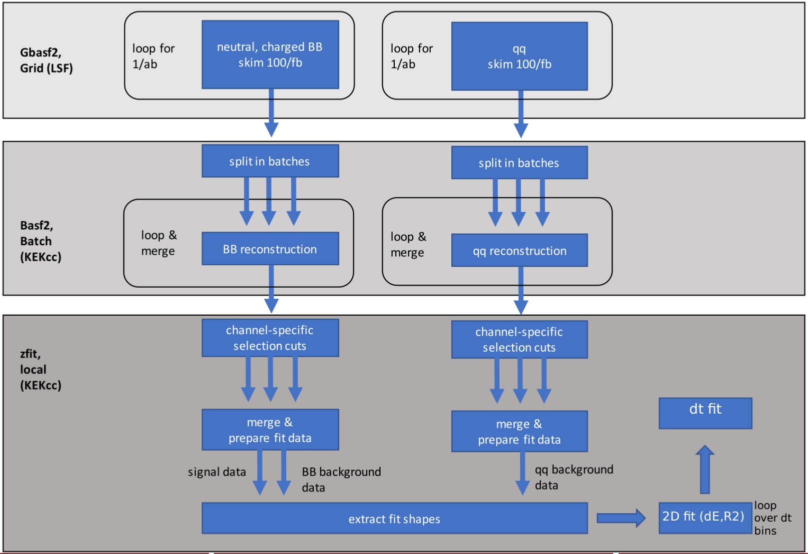
HEP Analysis Workflow

With modern HEP analyses being both increasingly

- complex (MVA Analysis, Training Interdependent NNs, ...)
- scaled (more data for e.g. rare processes, precision measurements, ...)
- distributed (multiple remote run locations, GPUs, ...)

many *undocumented dependencies* emerge.

Example: B^0 mixing analysis @ Belle II



HEP Analysis Workflow

With modern HEP analyses being both increasingly

- complex (MVA Analysis, Training Interdependent NNs, ...)
- scaled (more data for e.g. rare processes, precision measurements, ...)
- distributed (multiple remote run locations, GPUs, ...)

many *undocumented dependencies* emerge.

Manual execution and job steering by analyst is

- error-prone
- time-consuming
- deteriorates the reproducability of results and the transparency of collaborative reviews
- hinders data preservation efforts.

HEP Analysis Workflow

With modern HEP analyses being both increasingly

- complex (MVA Analysis, Training Interdependent NNs, ...)
- scaled (more data for e.g. rare processes, precision measurements, ...)
- distributed (multiple remote run locations, GPUs, ...)

many *undocumented dependencies* emerge.

Manual execution and job steering by analyst is

- error-prone
- time-consuming
- deteriorates the reproducability of results and the transparency of collaborative reviews
- hinders data preservation efforts.

Need for...

Workflow Management Systems!

Workflow Management can help!

A **workflow**: a top-level entity of workload to accomplish a scientific objective.
A **task**: a step in a workflow, processes input to produce output.
A **group of tasks** = a workflow, described by directed acyclic graph (DAG).

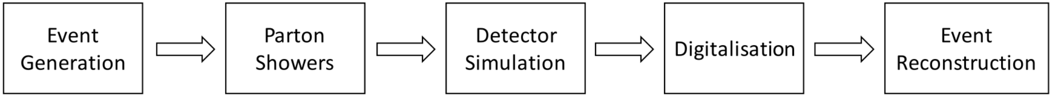
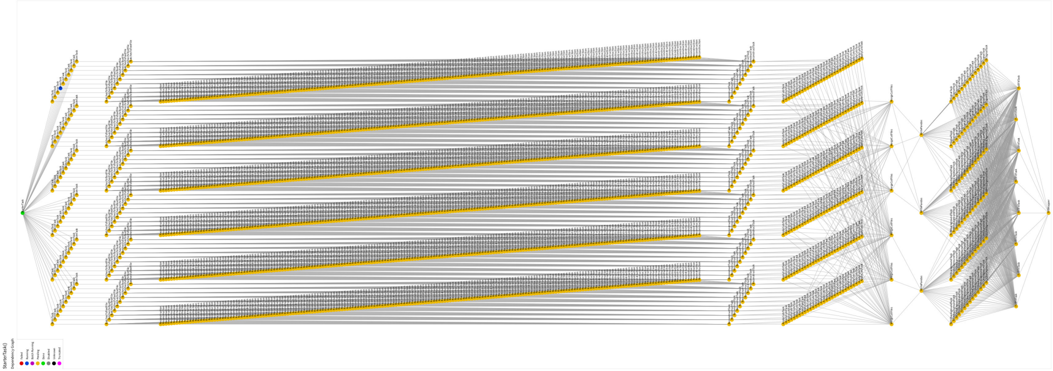


Figure 1: Existing Workflow: MC Production

Example: **Directed Acyclic Graph** in B^0 mixing analysis @ Belle II

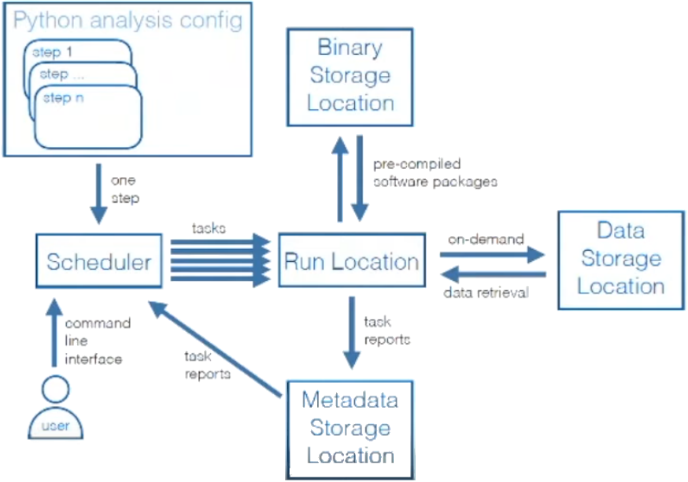
A **group of tasks** = a workflow, described by **directed acyclic graph (DAG)**.



Two possible realizations for Workflow Management

Report Based:

- store report file for each execution, evaluate reports of predecessors
- not-intensive on storage

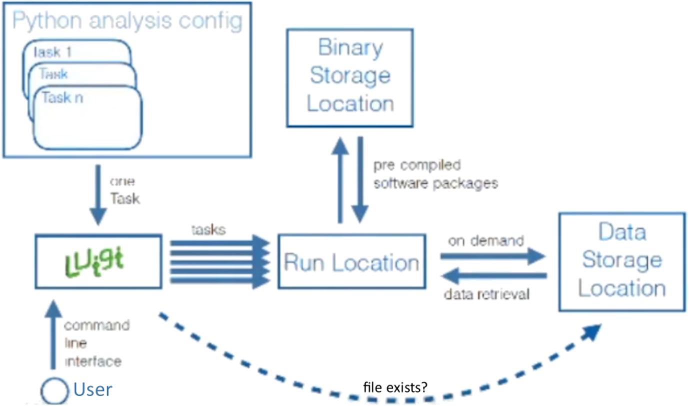


Robert Fischer. Workflow Management for User Analysis in Particle Physics, ACAT 2017.

Two possible realizations for Workflow Management

Target Based:

- check for output target before execution
- more intensive on storage
- No intermediate metadata storage with potential failure

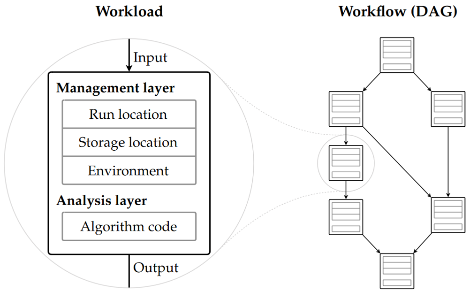


Robert Fischer. Workflow Management for User Analysis in Particle Physics, ACAT 2017.

Requirements for Workflow Management System

Wishlist:

- *Encapsulation* in logically separated steps.
- *Factorization* within a workload: management and analysis layer are fully decoupled
- *Interchangeability* of workloads storage and run locations
- *Universality* in programming language, workload outputs etc.

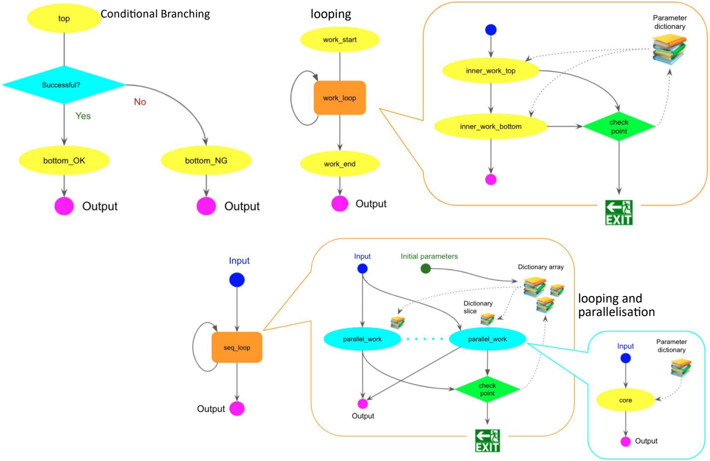


M. Rieger. Design pattern for analysis automation on distributed resources using luigi analysis workflows, EPJ Web of Conferences, EDP Sciences 2020.

Requirements for Workflow Management System

Wishlist (continued):

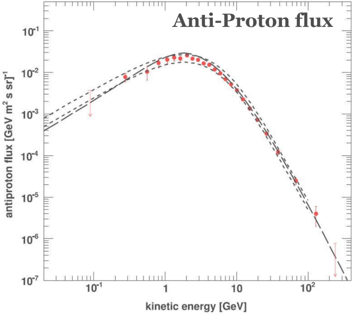
- *Automatization* in data retrieval, dependency resolution (and bookkeeping).
- *Detachable* in VE, Containers, Sandboxes, ...
- *Complex* Workflows: non-static, dynamic run-dependencies, conditional branching, parallelisation, looping, ...



Tadashi Maeno. PanDA Evolution for ATLAS and Beyond, Software Comp. Table 2022.

Digression: Physics in neutral B mixing

Astrophysical signatures for anti-matter:
anti-nuclei and γ fluxes from annihilation.



Experiments (e.g. AMS, Pamela):
$\overline{p}$ flux purely secondary, no $\overline{He}$ flux, γ flux insufficient.

Need CP violation to explain baryon asymmetry.

From cosmic microwave background anisotropies we infer
$\eta_B \equiv \frac{n_B - n_{\overline{B}}}{n_\gamma^{CMB}} = (6.1 \pm 0.3) \cdot 10^{-10} \neq 0.$

Sakharov (1967)

- baryon-number violation
- charge-parity violation (CPV)
- thermal non-equilibrium

Are there additional sources of CPV beyond the standard model? Precise tests of SM predictions!

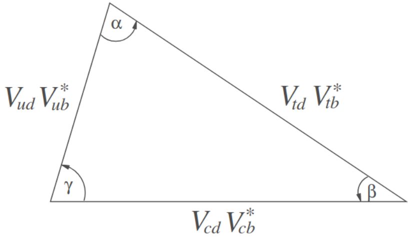
CP violation in the Standard Model

$$V_{CKM} \equiv \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}$$

In standard model, only weak interaction allows for C, P, CP, T violation through weak phase in V_{CMK} mixing matrix.

V_{CKM} unitarity yields 3 relations in complex plane, e.g.

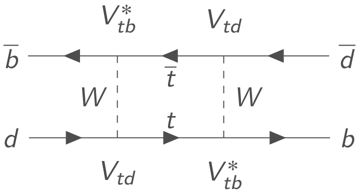
$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0.$$



unitary triangle

Neutral B meson mixing

New sources of CPV in the quark sector? Testing V_{CKM} unitarity by over-constraining unitarity triangle!



Extract mixing parameter Δm from time-dependent mixing asymmetry in decays to flavor-specific final states f:

$$\mathcal{A}_f(t) \equiv \frac{\Gamma(B^0(t) \to f) - \Gamma(\overline{B}^0(t) \to f)}{\Gamma(B^0(t) \to f) + \Gamma(\overline{B}^0(t) \to f)} = \cos(\Delta m t)$$

with $\Gamma(B^0/\overline{B}^0(t) \to f) = |A_f|^2 \cdot \frac{\exp[-t/\tau_B]}{2\tau_B} \cdot [1 \pm \cos(\Delta m t)]$.

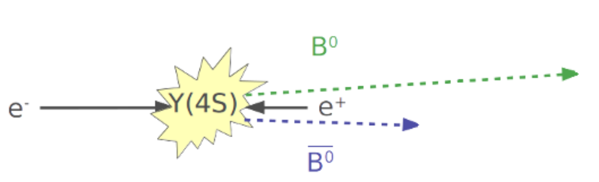
Analysis strategy at Belle II experiment

1. Produce **large B meson quantities** at accelerator.
2. Reconstruct B mesons in **4 hadronic decay channels** $B \rightarrow \pi D^{(*)}$
3. Fit to **separate backgrounds from signal**.
4. **Reconstruct B flavor and decay time** t.
5. Fit background-substracted data to **extract** Δm **and** τ_B.

B mixing at e^+e^- colliders

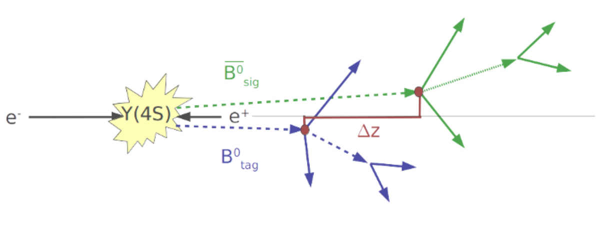
- e^+ and e^- are collided with $E = m(\Upsilon(4S))c^2 \approx 10.5\mathrm{GeV}$.
- Exploit $Br(\Upsilon(4S) \to B\overline{B}) > 96\%$ of bottonium state.
- Need excellent time resolution to resolve oscillations of B^0 mesons with $\Delta m \approx 0.5\mathrm{ps}^{-1}$.
- Translates into exceptional vertex resolution at detectors.

Asymmetric colliders boost the $\Upsilon(4S)$ restframe, and dilatate decay lengths $z_{lab} = \beta\gamma c\tau_B > z_{com}$.



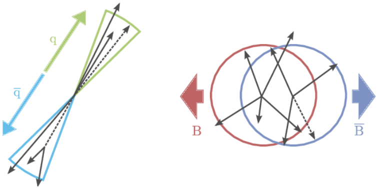
Mixing measurement in hadronic $B \to \pi D^{(*)}$

We measure $\Delta z \approx \beta\gamma(t_{sig} - t_{tag})c = \beta\gamma\Delta t\, c$



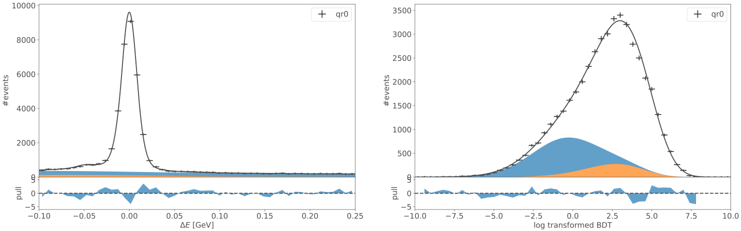
We either flavor-tag both B mesons in same flavor (SF) or opposite flavor (OF), with wrong-tag fraction w:

$$\mathcal{A}_f(\Delta t) \to \frac{N_{\rm OF}(\Delta t) - N_{\rm SF}(\Delta t)}{N_{\rm OF}(\Delta t) + N_{\rm SF}(\Delta t)} = (1-2w)\cdot\cos(\Delta m \Delta t)$$



- Combinations of random tracks from $e^+e^- \to q\bar{q}$ and
- mis-reconstructed $e^+e^- \to B^0\overline{B}^0, B^+B^-$

deteriorate measurement precision. Background-subtract data from 2d fit in BDT output and $\Delta E \equiv E_{\mathrm{B}}^{\mathrm{com}} - E_{\mathrm{beam}}^{\mathrm{com}}$.



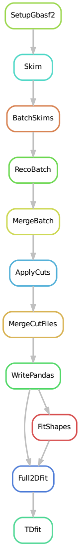
Now let us recreate this analysis using workflow management!

Workflow Management in Belle II B^0 mixing analysis

Single executable, full analysis automatized and portable!

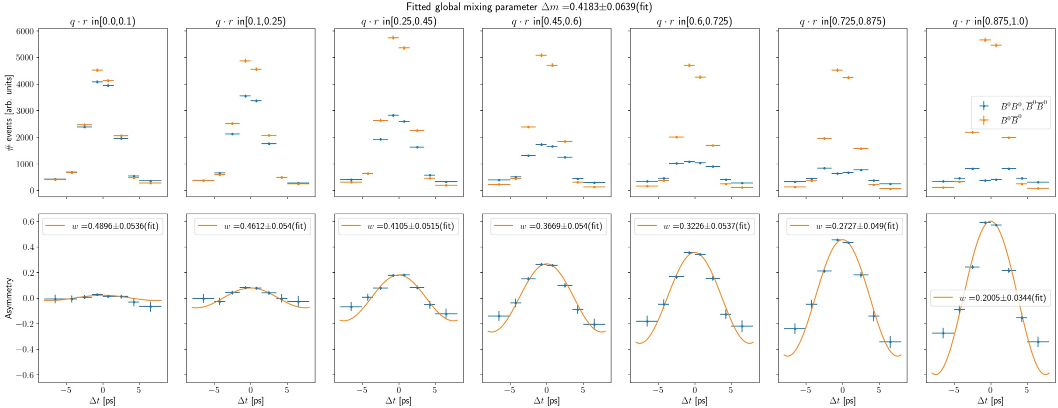
Fully automatized tasks:

1. submitting gbasf2 jobs on LSF Grid
2. job steering, dataset downloading and error handling
3. batching basf2 jobs, job steering, file merging and error handling
4. selection cuts, event selection and fit shape extraction
5. event binning and multivariate fitting
6. plotting final result.



Result in Belle II B^0 mixing analysis

Fit results binned in flavor tag figure of merit $q \cdot r$:



Globally fitted mixing frequency 1.4σ from PDG,
despite simplifications (resolution functions, BDT, systematics etc.)

Comparison of Selected Workflow Management Systems

Single applications:

Luigi

- developed by Spotify, open sourced
- Python syntax
- target based
- integrated analysis code and WF logic
- focus on dynamic DAG visualization and remote execution support
- automatised job steering on LSF and KEKcc. . .

Snakemake

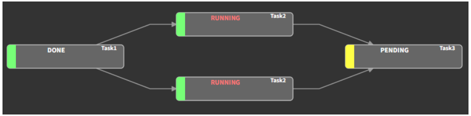
- developed at Uni Duisburg-Essen
- Custom Python-based syntax
- target based
- Analysis code and WF logic factorize
- focus on environment management and remote execution support
- automatised job steering on LSF and KEKcc. . .

Server-based: e.g. **Reproducible Analysis Platform** developed by CERN.

Simple example for workflow in Luigi

```python
#luigi_workflow.py
import numpy
import subprocess

class Task3(luigi.Task):
    def requires(self):
        return [Task2(NumberOfRandoms = 10), Task2(NumberOfRandoms = 5)]

    def output(self):
        return luigi.LocalTarget("final/finalResult.txt")

    def run(self):
        command = ["cat"]
        for input in self.input(): command.append(input.path)
        with self.output().open("w") as output:
            output.write(subprocess.check_output(command).decode())

class Task2(luigi.Task):
    NumberOfRandoms = luigi.IntParameter()
    def requires(self):
        return Task1()

    def output(self):
        return luigi.LocalTarget(f"intermediate/intermediateResult_{self.
                                        NumberOfRandoms}.txt")

    def run(self):
        with self.input().open("r") as input:
            numpy.random.seed(int(input.readlines()[0]))
        with self.output().open("w") as output:
            for i in range (self.NumberOfRandoms): output.write(f"{numpy.
                                                random.random()}\n")

class Task1(luigi.Task):
    def output(self):
        return luigi.LocalTarget("initial/seed.txt")

    def run(self):
        with self.output().open("w") as output:
            output.write(str(42))
```

Dynamic DAG visualization:



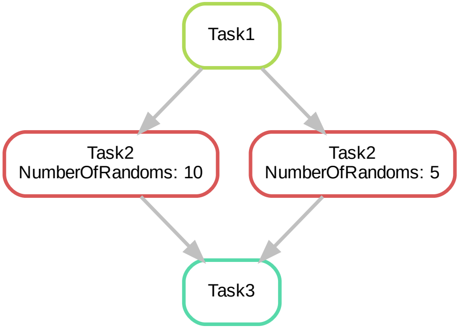
Automatic parallelization:

`luigi --module my module Task3 --workers 2 --local-scheduler`

Simple example for workflow in Snakemake

```
#snakefile
rule Task3:
    input:
        "intermediate/intermediateResult_10.txt",
        "intermediate/intermediateResult_5.txt"
    output:
        "final/finalResult.txt"
    shell:
        "cat {input} > {output}"

rule Task2:
    input:
        "initial/seed.txt"
    output:
        "intermediate/intermediateResult_{NumberOfRandoms}.txt"
    conda:
        "environment.yaml"
    containerized:
        "container.sif"
    params:
        current_NumberOfRandoms = lambda wildcards: int(wildcards.
                                                    NumberOfRandoms)
    script:
        "python_script.py"

rule Task1:
    output:
        "initial/seed.txt"
    shell:
        "echo 42 > {output}"
```

```
#python_script.py
input_file = snakemake.input[0]
output_file = snakemake.output[0]
number_of_randoms = snakemake.params.current_NumberOfRandoms

import numpy
with open(input_file,"r") as input:
    numpy.random.seed(int(input.readlines()[0]))
with open(output_file,"w") as output:
    for i in range(number_of_randoms): output.write(f"{numpy.random.random
                                                    ()}\n")
```



Automatic parallelization, containerization, virtual environment and package management:

`snakemake --cores 2 --use-singularity --use-conda`

Extensive comparison on arXiv:2212.01422

	Criteria	Luigi	Snakemake	Reana
Project	Learning curve	intermediate	simple	simple
	Programming languages	single	multiple	multiple
	Relation to analysis code	integrated	complete factorization	complete factorization
	Workflow language	Python	custom Python-based	Python-based
	Boilerplate code	minimal	minimal	intermediate
	Data formats	any	any	any
	Dependency management	automatic	automatic	automatic
	State management	target based	target based	target based
Interface	Visualization and monitoring	extensive	no dynamic DAG	no dynamic DAG
	Execution control	extensive	extensive	limited
	Error handling and debugging	failed output remains	failed output deleted	single-use clean environment
	Architecture	single application	single application	server
Features	Scalability	easy	easy	easy
	Portability	easy	easy	easy
	Environment management	minimal	extensive	limited
	Storage systems support	extensive	extensive	extensive
	Remote execution support	extensive	extensive	extensive
	Authentication mechanism	environment variables	environment variables	access tokens
	Version control	external	external and internal	external and internal
Integration	Installation	pip, non-root	conda, non-root	pip, non-root
	Documentation	extensive	extensive	incomplete
	Support	extensive	extensive	satisfactory
	System developers	Spotify Group	academic team	CERN
	History and activity	very active	very active	less active
	User community	large	large	significant
	Long term perspective	good	good	acceptable
	Lock-in	yes	no	no
	License	Spotify Group, free use	MIT, free use	CERN
	Use in Punch	Belle 2	LHCb, Radioastronomy	CERN

Conclusions

- Workflow management should become standard for transparency, reproducibility and data preservation.
- Variety of tools for various use-cases.
- We implemented simplified B^0 mixing analysis in Snakemake and Luigi.
- We are providing a comparison of tools.