Pentaquarks and Open Data

LHCb-Germany meeting, Dortmund 2021

Adam Morris[1], with material from: Abhay Mehta[1], Mindaugas Šarpis[1], Marian Stahl[2], Chris Burr[3], Dillon Fitzgerald[4], Donijor Tropmann[5]

[1]University of Bonn, [2]University of Cincinnati, [3]CERN, [4]University of Michigan, [5]RWTH Aachen

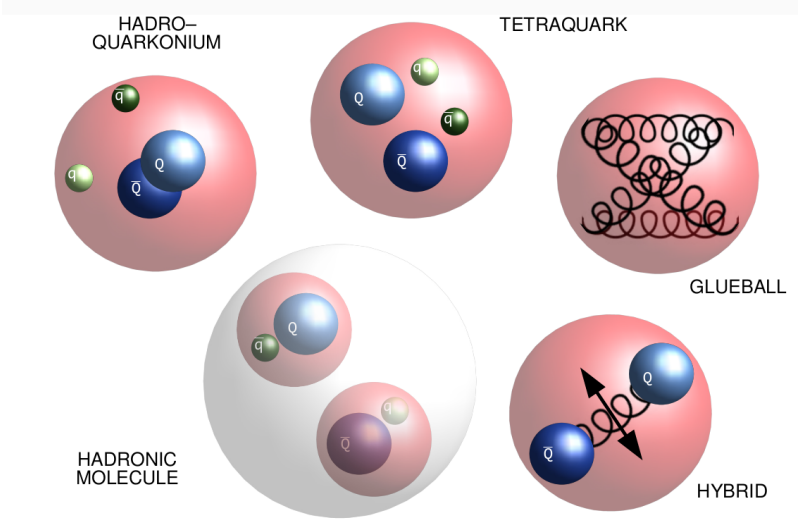
8th October 2021

Activities in the Bonn LHCb group

- **Searches for $J/\psi\, p$ pentaquarks in other channels**
- Mighty Tracker R&D
- Analysis Preservation & **Open Data**
- Simulation production

Pentaquarks

Exotic hadrons

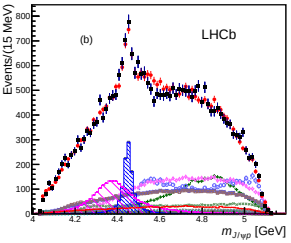


Graphics: C. Hanhart

Pentaquarks: some history

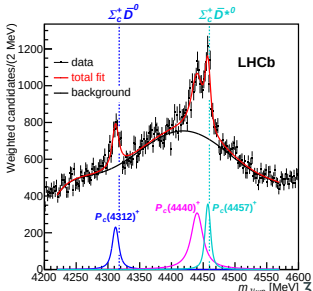
Discovery of pentaquarks [arXiv:1507.03414]

- Amplitude analysis of $\Lambda_b^0 \to J/\psi\, p K^-$ showed clear structure in $J/\psi\, p$ spectrum
  - $P_c(4450)^+$, $\Gamma \approx 40$ MeV
  - $P_c(4380)^+$, $\Gamma \approx 200$ MeV
- Proximity to $\Sigma_c^+ \bar{D}^{*0}$ threshold



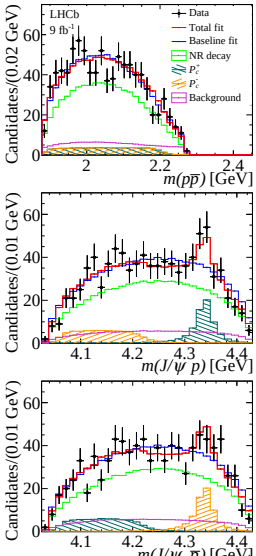
Adding the Run 2 data [arXiv:1904.03947]

- $P_c(4450)^+$ resolved to two-peak structure
  - $P_c(4440)^+$, $\Gamma \approx 21$ MeV
  - $P_c(4457)^+$, $\Gamma \approx 6$ MeV
- New pentaquark right on $\Sigma_c^+ \bar{D}^0$ threshold
  - $P_c(4312)^+$, $\Gamma \approx 10$ MeV



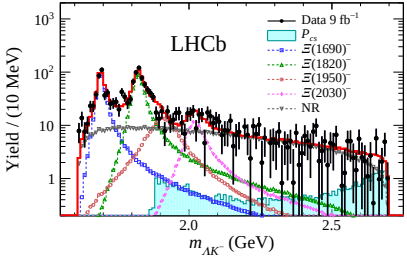
Evidence of a new $J/\psi\, p$ structure in $B_s^0 \to J/\psi\, p\overline{p}$ [arXiv:2108.04720]

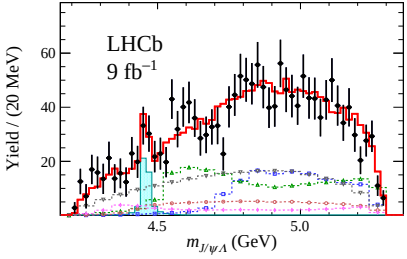
- Amplitude analysis of $B_s^0 \to J/\psi\, p\overline{p}$ decays
- Observe structure in $J/\psi\, p$ and $J/\psi\, \overline{p}$ spectra
- Potentially a new $P_c(4337)^+$ state
  - $m = 4337^{+7+2}_{-4-2}$ MeV, $\Gamma \approx 30$ MeV
- Significance 3.1σ to 3.7σ depending on J^P assignment
- Inconsistent with $P_c(4312)^+$
  - $m = 4311.9 \pm 0.7^{+6.8}_{-0.6}$ MeV



Evidence of a $c\bar{c}sud$ pentaquark [arXiv:2012.10380]

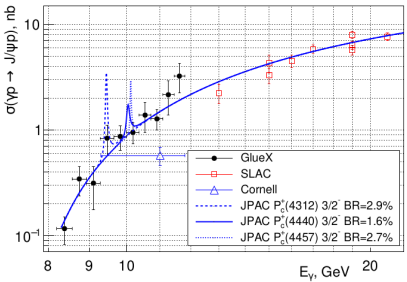
- Amplitude analysis of $\Xi_b^- \to J/\psi \Lambda K^-$ decays
- Evidence of structure in $J/\psi \Lambda$ spectrum
  - $P_{cs}(4459)^0$
- Significance 3.1σ
- Potentially a first charmonium-pentaquark with strangeness





Where do pentaquarks get their width?

- Photoproduction experiments put upper limit on partial width $\Gamma(P_c \to J/\psi\, p) < 2 \sim 5\%$
  [Phys. Rev. Lett. 123 (2019) 072001]
- Expect to see P_c in other channels with larger branching fractions
- Molecules should couple strongly to channels with open charm, *e.g.* $\Lambda_c^+ \overline{D}^0$, $\Sigma_c^+ \overline{D}^0$, $\Lambda_c^+ \overline{D}^{*0}$
- Focus of analyses in the Bonn LHCb group



P_c State	$\mathcal{B}(P_c \to J/\psi\, p)$
$P_c^+(4312)$	$< 4.6\%$ (90% CL)
$P_c^+(4440)$	$< 2.3\%$ (90% CL)
$P_c^+(4457)$	$< 3.8\%$ (90% CL)

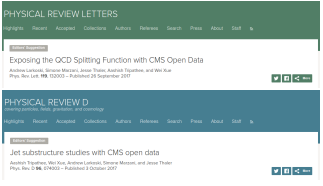
Open data

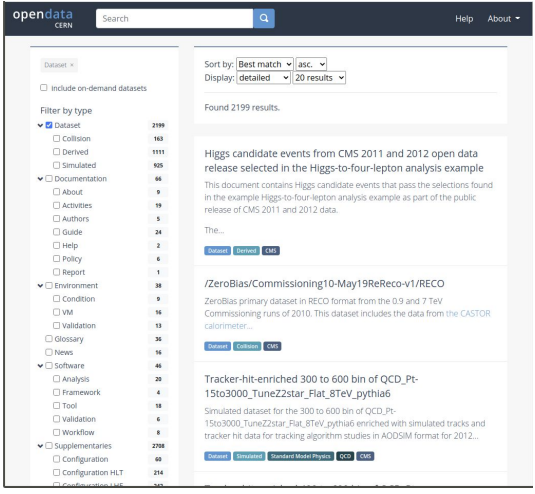
CERN Open Data Portal

https://cern.ch/opendata

CERN repository for public datasets

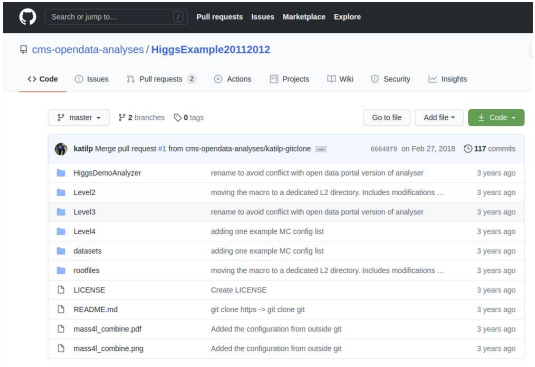
- $>$ 2 PB of data from ATLAS, CMS, ALICE, LHCb and OPERA
- Used in education, outreach and real physics research

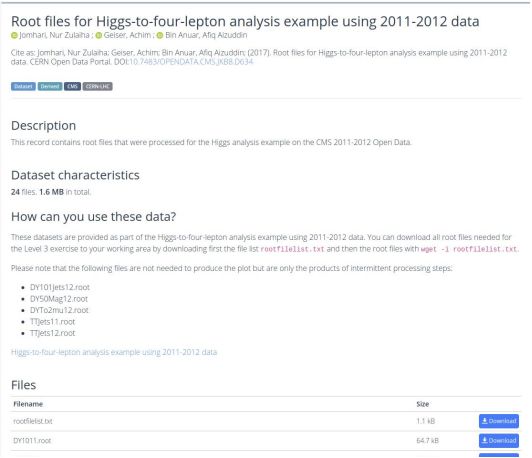




CERN Open Data Portal

https://cern.ch/opendata





Example analysis: CMS $H \rightarrow 4\ell$ discovery

CERN Open Data policy

Updated policy endorsed by the large LHC experiments CERN-OPEN-2020-013

Applies FAIR principles

Four levels of data complexity:

Level 1: Public results
e.g. HEPData

Level 2: Education and outreach
e.g. Masterclasses

Level 3: **Reconstructed data**
Open Data Portal

Level 4: Raw data
Not released

CERN Open Data policy

Updated policy endorsed by the large LHC experiments CERN-OPEN-2020-013

Applies FAIR principles

Four levels of data complexity:

Level 1: Public results
e.g. HEPData

Level 2: Education and outreach
e.g. Masterclasses

Level 3: **Reconstructed data**
Open Data Portal

Level 4: Raw data
Not released

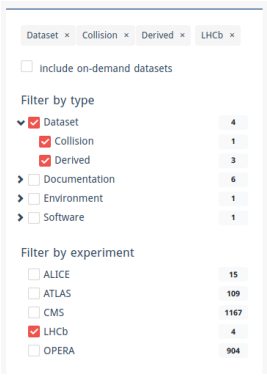
Level 3 data

- Data for main physics programme (protons and heavy ions)
- Special runs may not be included
- Software must also be available
- Data formats same as internal
- Collaborations will not review or sign third-party analyses

LHCb Open Data policy

- Open same datasets that we use internally
  - Stripping/Turbo/Sprucing streams
- 50% (100%) after 5 (10) years since the end of each LHC Run
  - Exceptionally withhold individual datasets still being used in ongoing analyses
- Provide tools to perform calibrations & corrections but no further support
  - PIDCalib, TrackCalib, etc…
- Require proper acknowledgement in papers using LHCb Open Data
  - Including disclaimer that LHCb has not reviewed the result
- LHCb collaborators are not permitted to sign papers using LHCb Open Data
  - Possible exceptions if no physics results shown (*e.g.* methodology papers)

LHCb Open Data: challenge #1



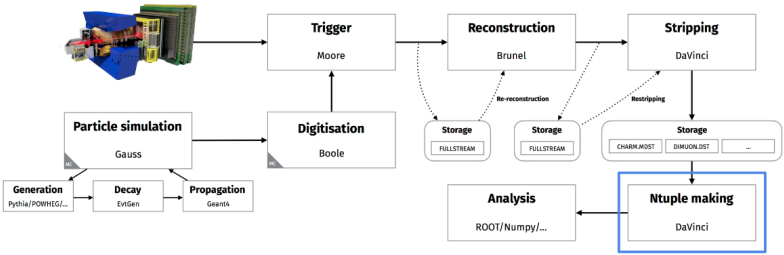
Why hasn't the Open Data been released yet?

Significantly larger filesizes than other LHC experiments!

	ALICE	ATLAS	CMS	LHCb
Run-2	2 PB	0.5 PB	2 PB	10 PB (including Run-1)
Run-3	4 PB	1 PB	4 PB	45 PB
Total	**6 PB**	**1.5 PB**	**6 PB**	**55 PB**

Not scalable: quickly hit storage quota on Open Data Portal.

LHCb Open Data: challenge #2



Ntuple making currently requires knowledge of the LHCb software stack

- Lots of documentation and support required to be successful
  - See LHCb Starterkit
- High barrier of entry for external analysts
- Complete overhaul of data processing (including ntupling) for Run 3
  - Knowledge required to access Run1+2 data will start fading “soon”

Ntuple Wizard

Make ntuples of LHCb data without knowledge of the software stack

- Render necessary option files from intuitive user inputs
- Create ntuples via Analysis Productions
- Return ntuples to user on the CERN Open Data Portal

Ntuple Wizard

Make ntuples of LHCb data without knowledge of the software stack

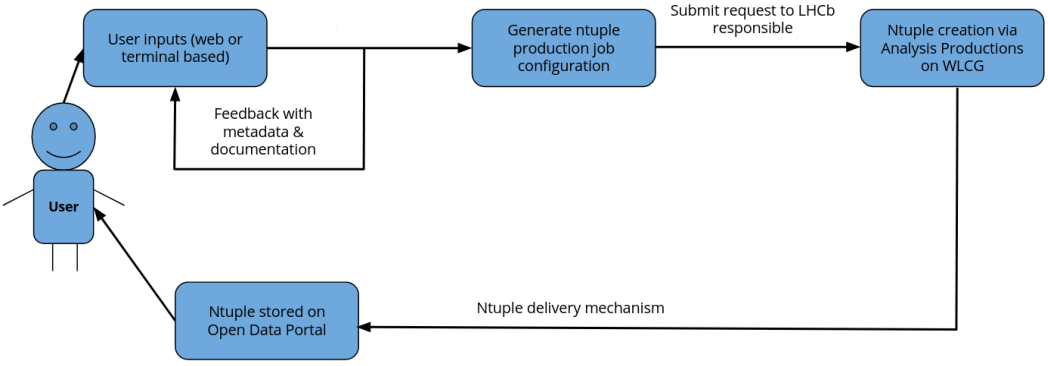
- Render necessary option files from intuitive user inputs
- Create ntuples via Analysis Productions
- Return ntuples to user on the CERN Open Data Portal

This solves several problems:

- ✓ Much smaller storage and bandwidth requirements on Open Data Portal
- ✓ Significantly flattened learning curve for accessing LHCb data
- ✓ No need for a computing cluster just to download and process the data

NB: *Also benefits LHCb members*

Overview


User inputs (web or terminal based)
Generate ntuple production job configuration
Submit request to LHCb responsible
Ntuple creation via Analysis Productions on WLCG
Feedback with metadata & documentation
User
Ntuple stored on Open Data Portal
Ntuple delivery mechanism

Core components

Backend

- Collect metadata from various sources and serve as static files

  `lhcb-dpa/wp6-analysis-preservation-and-open-data/lhcb-ntuple-wizard`

Web frontend

- Guide the user through choosing a dataset and configuring algorithms

  `lhcb-temp-transfer/tuple-wizard-frontend`

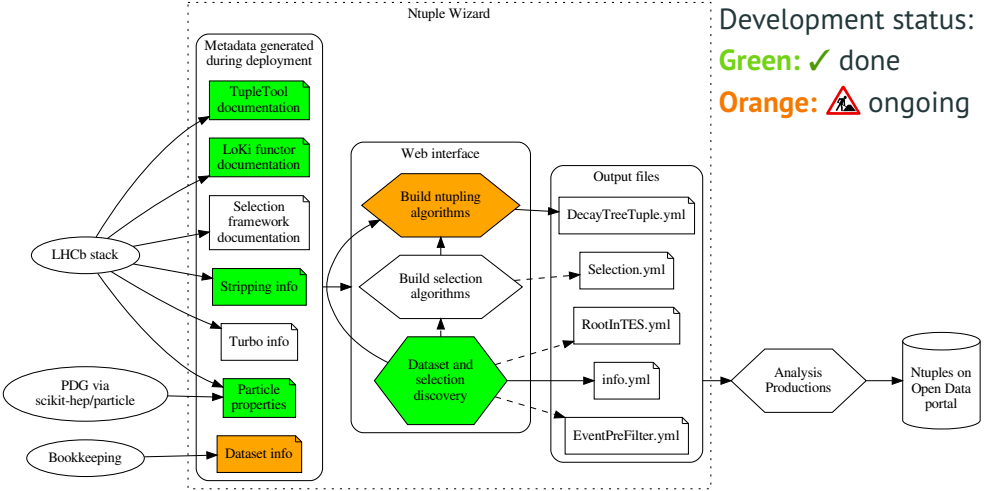
Parsers

- Configure DaVinci with the output of the Wizard

  `admorris/analysishelpers`

***NB:** repo locations subject to change*

In more detail



Backend: Sources of metadata

DaVinci

- Access via Gaudi python scripts
- TupleTool interfaces
- Stripping lines
  - Decay descriptors
  - Streams, TES locations

PDG

- `scikit-hep/particle`
  - Physical properties
  - LaTeX names
- `ParticleTable.txt` from DDDB
  - Gaudi names
  - Custom particles

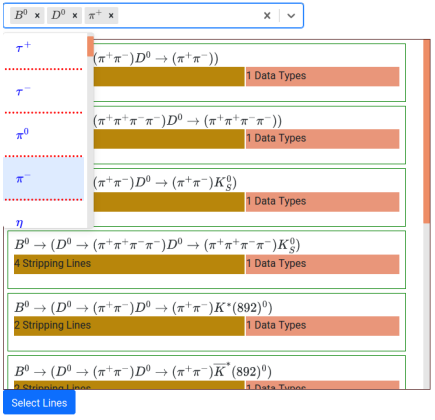
Doxygen

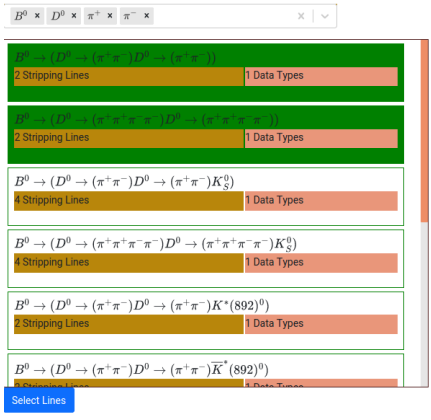
- Non-trivial URL discovery
- HTML pages parsed with `bs4`
- TupleTool and LoKi functor docs

Bookkeeping

- Access via LHCbDIRAC API scripts
- Delegated to sister project: `lhcb-opendata-curation`
  - Generates the ODP records

Frontend: Decay descriptor search





- Filtering of decay descriptors by multiple particle types
  - ⚠ More advanced filtering options (e.g. decay head)
- Selection of multiple decay descriptors

Frontend: Stripping line selection

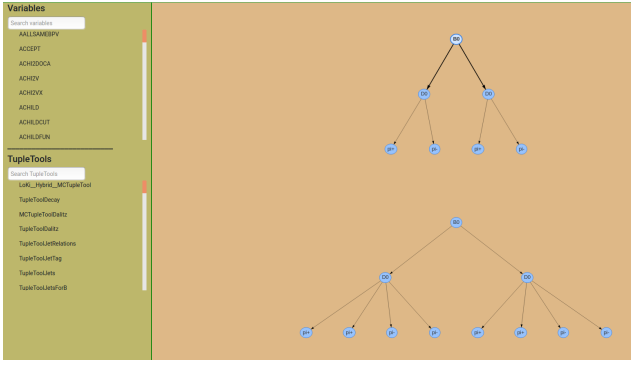
Lines Search

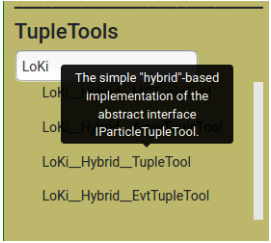
Decay	Line	Datatype	
$B^0 \to (D^0 \to (\pi^+\pi^-)D^0 \to (\pi^+\pi^-))$	StrippingB02D0D0D02HHD02H...	2011	Remove
$B^0 \to (D^0 \to (\pi^+\pi^+\pi^-\pi^-)D^0 \to (\pi^+\pi^+\pi^-\pi^-))$	StrippingB02D0D0D02HHD02K3...	2011	Remove

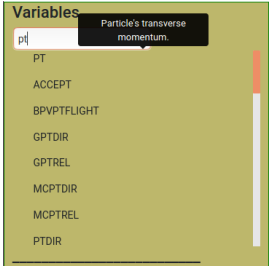
Build Tree

- List available stripping lines and running years for selected decay descriptors
-  Select multiple running years for the same decay descriptor
-  Provide enough documentation for users to choose the appropriate stripping line for their physics interest

Frontend: DecayTreeTuple configuration







- Visualisation of decay tree as interactive DAGs
- LoKi functor and TupleTool documentation in info blurbs
- ⚠ CRUD form for TupleTools on each node

Problem

- Output of Ntuple Wizard must be data-structures to prevent code injection
- Gaudi apps can be configured natively with JSON or a bespoke `.opts` format
  - ✗ Requires explicit configuration of all options incl. default values
- Want to configure only certain parts *à la* python options files → **custom parsers**

AnalysisHelpers

- *Importable* data-package, inspired by a trick used in AnalysisProductions
  - Not the same use-case as AppConfig
- Principal aim: **reduce boilerplate code** within options files
- Contains general helpful code for configuring DaVinci
  - Restripping, MC filtering
  - **DecayTreeTuple from YAML**
  - *etc…* (open for contributions)
- Release cycle decoupled from DaVinci
- Will be made available to AnalysisProductions

Conclusion

Conclusion

Pentaquarks

- Hints of new charmonium-pentaquark species, including one with strangeness

Open Data

- LHCb data format poses a challenge for Open Data release
  - Size and ease-of-use
- Ntuple Wizard offers a solution
  - Development of key features for first release in mature state
  - Our experience will provide input to Run 3 ntupling

Thanks for listening

Appendix

Open Data

FAIR Guiding Principles for scientific data management and stewardship

https://go-fair.org

Findable: Persistent identifiers, rich metadata, indexed and searchable.

Accessible: Retrievable over standard protocols. Metadata accessible even if data isn't.

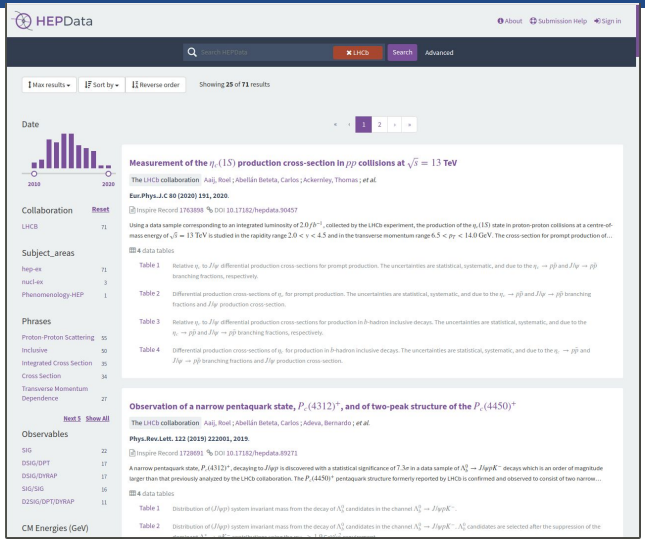
Interoperable: Appropriate language/format. Qualified references to other data.

Reusable: Clear usage licences. Detailed provenance. Meet community standards.

HEPData

https://hepdata.net

- Mainly high-level observables / analysis results:
  - cross-sections
  - particle properties
  - SM parameters
  - limits and confidence intervals
- Machine-readable format



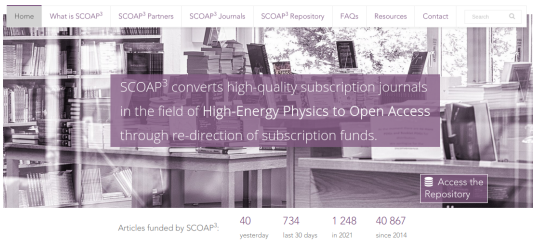
CERN and Open Science

Article II

Purposes

1. The Organization shall provide for collaboration among European States in nuclear research of a pure scientific and fundamental character, and in research essentially related thereto. The Organization shall have no concern with work for military requirements and the results of its experimental and theoretical work shall be published or otherwise made generally available.

SCOAP^3 – Sponsoring Consortium for Open Access Publishing in Particle Physics



'International cooperation and dissemination of information have been enshrined in the CERN Convention for sixty years...'

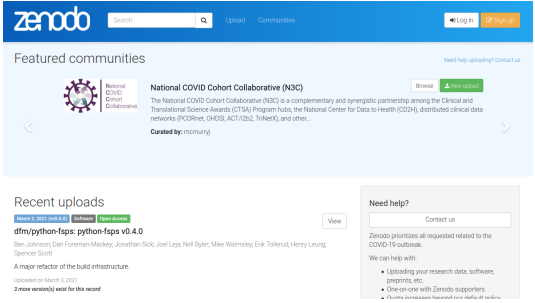
Rolf Heuer, former CERN Director General

CERN and Open Science

Article II

Purposes

1. The Organization shall provide for collaboration among European States in nuclear research of a pure scientific and fundamental character, and in research essentially related thereto. The Organization shall have no concern with work for military requirements and the results of its experimental and theoretical work shall be published or otherwise made generally available.



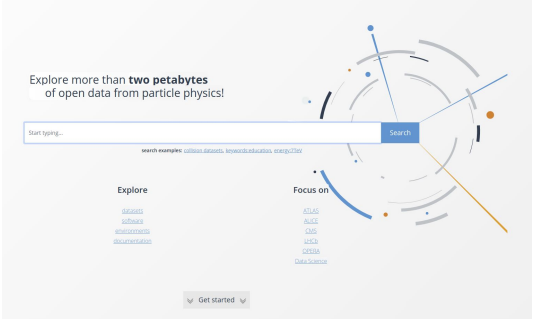
General-purpose open-access repository developed under the European OpenAIRE program and operated by CERN

CERN and Open Science

Article II

Purposes

1. The Organization shall provide for collaboration among European States in nuclear research of a pure scientific and fundamental character, and in research essentially related thereto. The Organization shall have no concern with work for military requirements and the results of its experimental and theoretical work shall be published or otherwise made generally available.



‘ The open data policy reflects CERN’s commitment to open science, which was already asserted in the CERN Convention over 60 years ago.’

Eckhard Elsen, former CERN Director for Research and Computing

Ntuple Wizard

Considerations

Security & permissions

- ! DaVinci typically configured with python scripts
  - ✗ **Running arbitrary code from outside is a bad idea**
  - Wizard saves configuration as data structures, to be interpreted by our parsers
- ! Dataset discovery requires metadata from Bookkeeping and LHCb stack
  - Pull all metadata at "deployment time"
  - **Read static files at runtime**, no interaction with DIRAC
- ! Locations of "unreleased" datasets easily guessable
  - Check input data against allowed list
- ! LHCb policy reserves right to withhold part of a dataset (e.g. ongoing analysis)
  - Require **fine-grained control** over:
    - building/accessing decay candidates
    - Stripping lines or equivalent selections
  - No elegant/agreed solution yet

Ntuple Wizard backend

Doxygen scraping

URL discovery in two parts:

- Prefix for a specific application and version found from `docs/db.json`
  - e.g. DaVinci/v45r8 → DOC_008155
- Pages for all classes and namespaces extracted from `{prefix}/search/search.idx` into a JSON file, e.g.:

```
{
  ...
  "class TupleToolPid": "d8/ddd/class_tuple_tool_pid.html",
  "class TupleToolPIDCalib": "d9/db9/class_tuple_tool_p_i_d_calib.html",
  "Namespace LoKi__Cuts": "d7/dae/namespace_lo_ki_1_1_cuts.html",
  ...
}
```

- Pages parsed with BeautifulSoup

LoKi functors

- Extract URL for `LoKi::Cuts` namespace:

https://lhcb-doxygen.web.cern.ch/lhcb-doxygen/davinci/v45r8/d7/dae/namespace_lo_ki_1_1_cuts.html

- Iterate through all `<div class="memitem">`
  - Name from `<td class="memname">`
  - Description from `<div class="memdoc">`
- Save `{name: description}` pairs to JSON file

TupleTool documentation

- Extract URL for specific TupleTool class, e.g.:

https://lhcb-doxygen.web.cern.ch/lhcb-doxygen/davinci/v45r8/d8/ddd/class_tuple_tool_pid.html

- Take description from the first `<div class="textblock">`

TupleTool interfaces

Complete list of TupleTools discoverable from python:

```
from Configurables import __all__ as all_configurables
all_tupletools = list(filter(lambda t: "TupleTool" in t, all_configurables))
```

Interfaces extracted using `getPropertiesWithDescription`:

```
interfaces = {t: {} for t in all_tupletools}
for tool_name in all_tupletools:
    tool = getattr(Configurables, tool_name)()
    interfaces[tool_name] = tool.getPropertiesWithDescription()
```

Output contains option names, default values and descriptions:

```
"LoKi__Hybrid__MCTupleTool": {
  "Variables": [{}, "The {'name':'functor'}-map of columns for N-tuple"],
  "Preambulo": [[], "The preambulo to be used for Bender/Python script"],
  ...
},
```

TupleTool metadata

Description (from Doxygen) and interface (from DaVinci) are combined into JSON entry:

```
"LoKi__Hybrid__MCTupleTool": {
  "description": "Simple implementation of IMCParticleTupleTool interface. ",
  "interface": [
    {
      "name": "Variables",
      "default": {},
      "description": "The {'name':'functor'}-map of columns for N-tuple",
      "type": "dict"
    },
    {
      "name": "Preambulo",
      "default": [],
      "description": "The preambulo to be used for Bender/Python script",
      "type": "list"
    },
    ...
  ]
},
```

Stripping lines: step 1

- Streams and lines for a specific Stripping version from `strippingArchive`
  - c.f. any restripping example
- Record output location, decay descriptors, stream, MDST flag
- Iterate recursively through dependency tree of algorithms
  - decay descriptiors
  - PID substitutions
  - input locations
- Prune empty nodes, consolidate and save to JSON:

```
"StrippingBsPhiRhoLine": {
  "MDST": false,
  "decay": ["B0 -> phi(1020) rho(770)0"],
  "inputs": [{
      "decay": ["phi(1020) -> K+ K-", "rho(770)0 -> pi+ pi-"],
      "inputs": []
  }],
  "output": "Phys/BsPhiRhoLine/Particles",
  "stream": ["BhadronCompleteEvent"]
},
```

Stripping lines: step 2

- Recursively substitute decaying particles with their descriptors until all combinations of decays are built
- Apply substitutions extracted from `SubstitutePID` and `SubPIDMMFilter`
- Parsing and substitution code from scratch with `pyparsing`
  - Tools within `Phys.SelPy.utils` broken and incomplete

```
"StrippingBsPhiRhoLine": {
  "MDST": false,
  "decay": ["B0 -> phi(1020) rho(770)0"],
  "output": "Phys/BsPhiRhoLine/Particles",
  "stream": ["BhadronCompleteEvent"],
  "full_decays": ["B0 -> (phi(1020) -> K+ K-) (rho(770)0 -> pi+ pi-)"]
},
```

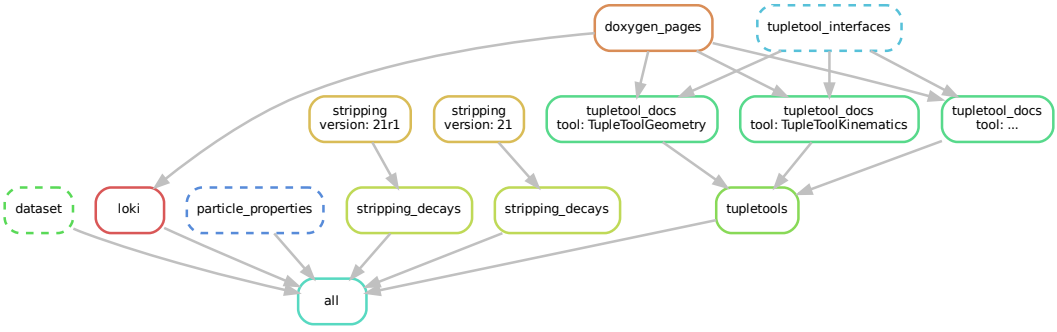
Particle properties

- Gaudi names from `ParticleTable.txt`
- Properties, LaTeX string and antiparticle relation from `scikit-hep/particle`
- Save to JSON:

```
"B0": {
  "mass": 5279.65,
  "charge": 0,
  "stable": false,
  "pid": 511,
  "latex": "B^{0}",
  "antiparticle": "B~0",
  "categories": ["Hadron", "Meson", "Down", "Beauty", "Bottom", "Scalar", "LongLived"]
},
```

Workflow automation

- Snakemake used to execute the workflow
  - Adopting suggested practices from `snakemake-workflows` repo



- GitLab CI executes the Snakemake worklow, `pytest` tests and linting

Package structure

For those interested in developing this package:

```
od-ntuple-wizard/
|-- {COPYING, README.md}                  # documentation
|-- {pyproject.toml, setup.cfg, setup.py} # standard python packaging stuff
|-- {deployment/, data/, logs/}           # output directories (created by snakemake)
|-- config.yml                            # versions, urls, paths etc
|-- scripts/
|   |-- {dump_*.sh} {dump_*.py}           # shell and Gaudi python scripts
|-- src/
|   |-- lhcb_ntuple_wizard/ # the main python module
|       |-- __init__.py
|       |-- __main__.py     # command line entry point (argparse)
|       |-- deployment.py   # metadata acquisition, parsing and formatting
|       |-- decays.py       # decay descriptor manipulation
|-- tests/
|   |-- test_*.py   # tests of all functions and classes (pytest)
|-- workflow/
    |-- __init__.py # makes utils.py importable from snakemake
    |-- Snakefile   # workflow automation rules
    |-- utils.py    # helper functions for the snakemake rules
```

AnalysisHelpers

The “magic”

```
<?xml version="1.0" encoding="UTF-8"?>
<env:config xmlns:env="EnvSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="EnvSchema EnvSchema.xsd ">
  <env:set variable="ANALYSIS_HELPERS_BASE">${.}</env:set>
  <!-- Allow data package to be imported in python -->
  <env:prepend variable="PYTHONPATH">${.}</env:prepend>
</env:config>
```

AnalysisHelpers.xenv

Use case: restripping

```
from Configurables import DaVinci
from AnalysisHelpers.stripping import line_name_contains, restripping
from AnalysisHelpers.dst_writer import write_dst

restrip_sequence = restripping(
    version = "34",
    line_filter = line_name_contains("B2XTauNu"),
    stream_name = "AllStreams"
)
dst_writer = write_dst(
    selection_sequences = restrip_sequence.activeStreams(),
    extra_items = ["/Event/pRec/ProtoP#99", "/Event/pRec/Calo#99"]
)

DaVinci.appendToMainSequence([restrip_sequence, dst_writer.sequence()])
```

examples/Restrip_RXcHad_s34.py

Drastic reduction in amount of boilerplate code compared to StarterKit example.

Use case: MC filtering on a stripping line

```
from Configurables import DaVinci
from AnalysisHelpers.stripping import line_name_contains, stripping_mc_filter
from AnalysisHelpers.dst_writer import write_dst

filter_sequence = stripping_mc_filter(
    version = "34",
    line_filter = line_name_contains("B2XTauNu"),
    stream_name = "RXcHad.Strip"
)
dst_writer = write_dst(
    selection_sequences = filter_sequence.activeStreams(),
    extra_items = ["/Event/pRec/ProtoP#99", "/Event/pRec/Calo#99"]
)

DaVinci.appendToMainSequence([filter_sequence, dst_writer.sequence()])
```

examples/Filter_RXcHad_s34.py

Very similar interface to restripping, with sensible defaults for MCFiltering production.

Use case: DecayTreeTuple from YAML

```
from Configurables import DaVinci
from AnalysisHelpers.decaytreetuple import configure_dtt
import yaml
config = yaml.safe_load(open("dtt_options.yml", "rb"))
for key in config:
    dtt = configure_dtt(key, config[key], verbose = True)
    DaVinci().UserAlgorithms += [dtt]
```

examples/dtt_from_yaml.py

Crucial component of Ntuple Wizard