LHCb Simulation Wizard

Alexia Alexander Wight
Supervisors: Adam Morris & Chris Burr

August 11, 2023

Abstract

The LHCb Simulation Wizard is a web tool that simplifies the process of finding available particle decays in the LHCb simulation software and samples which contain them. The app is directed both at LHCb physicists and users of LHCb Open Data. Whilst the LHCb software is currently public, the learning curve to use them is very steep: the aim of the Simulation Wizard is to reduce this via a user-friendly web tool. It produces an annotated version of the decay files, helping users to understand the content of the files without needing prior expertise; allows searching via particle decays (rather than by filenames of 8-digit event type identifiers); and instantly displays available samples.

1 Introduction

The focus of this project was the creation of the LHCb Simulation Wizard. The Wizard is a web tool that increases the accessibility of finding EvtGen user decay files (known as DecFiles) [1] and simulation samples from LHCb. Whilst all these things are currently technically accessible (at least to LHCb users — simulation samples are not available to Open Data users), the process can be difficult. The Wizard will provide a more reliable way to search for DecFiles by decay, and a quicker and easier way to find samples. Additionally, it lines up strongly with CERN's aim to make all research accessible and inclusive [2]. The core users for this tool would be LHCb users who have less experience with accessing decay files and simulation samples, but also Open Data users who are interested in LHCb simulations.

1.1 LHCb Simulation

The Large Hadron Collider beauty (LHCb) experiment is one of the four large detectors at the LHC.It focuses on heavy-quark flavour physics, specifically the decays of hadrons containing b and c quarks. As such, at LHCb simulations are generally samples of collision events containing a particular decay. The simulation is managed by the LHCb simulation framework, Gauss, that interfaces with a number of other applications [3]. Collisions are simulated using an event generator such as Pythia8 [4] until the desired particle is created, and then they are forced to decay a certain way by EvtGen. EvtGen is a Monte Carlo event generator that simulates the decays of heavy flavour particles [1]. A DecFile (decay file) configures the decays, and

```
# EventType: 27175000
#
# Descriptor: [D*(2010)+ -> (D0 ->
{K+ K- mu+ mu-}) pi+]cc
#
# NickName: Dst_D0pi,KKmumu=DecProdCut
#
# Cuts: DaughtersInLHCb
#
# Documentation: Forces the D* decay
in generic b-bbar / c-cbar events +
Requires products to be in LHCb acceptance
# EndDocumentation
#
# PhysicsWG: Charm
# Tested: Yes
# Responsible: Luisa Arrabito
# Email: unknown@<nospam>cern.ch
# Date: 20091215
#
```

```
Alias MyD0 D0
Alias MyantiD0 anti-D0
ChargeConj MyD0 MyantiD0

Decay D*+sig
1.000 MyD0  pi+    VSS;
Enddecay
CDecay D*-sig

Decay MyD0
1.000 K+ K- mu+ mu- PHSP;
Enddecay
CDecay MyantiD0
#
End
```

Figure 2: An example of an LHCb DecFile. The commented section contains important information and is parsed during compilation to created the `<event-type>.py` file that is given to Gauss as one of its options. The bottom part sof the file (here shown on the right) specifies the decay itself.

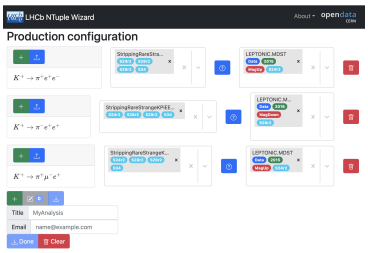
specifies the branching fraction, dynamics, and final states of particle decays. LHCb has it's own version fo EvtGen where additional options, such as generator cuts, are also specified. An example of a relatively simple DecFile can be seen in Fig. 2. One key aspect to highlight is the event type — this is a unique 8-digit identifier of the DecFile used in LHCb, and used throughout the Simulation Wizard to identify DecFiles. If these particle decay simulations pass some pre-determined criteria, detector simulation then occurs. Once the detector readout electronics are emulated (digitisation), this is processed in the same way as real data from the detector.

Data from LHCb — both real and simulated — is stored in "the bookkeeping". This is an online file browser, and searching for samples in here can be quite cumbersome, as there are many folders, with multiple levels. It is possible to make this process much easier if you know the exact "bookkeeping path", as you can go directly to the sample, instead of having to go through layers of folders. It is also possible to access these samples via the LHCbDIRAC command-line [5]. The signal decays are reflected in the filename and a comment at the top of the file for DecFiles (as seen in Fig. 2), however there is no strict enforcement of any kind of convention, so searching via just these file names in not 100% reliable. The difficultly in searching for and navigating through these systems is the core motivation for the creation of the Simulation Wizard.

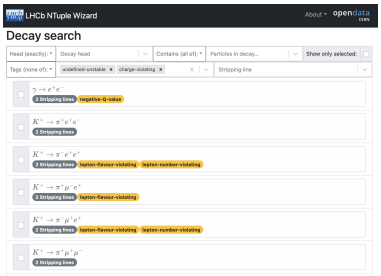
1.2 Open Data & Accessibility

Open science has been a core value at CERN from the start, when CERN's founding convention stated that all its results were to be published and made generally available [6]. Today, this philosophy continues, with CERN's open data policy aiming to make all CERN research fully accessible, inclusive, democratic and transparent [2]. LHCb has a 10-year embargo policy, where all data is released after 10 years, meaning all Run 1 data (2011-12) should be full available right now, and is viewable on CERN Open Data Portal [7]. Additionally, all LHCb software is open source, meaning it is freely available.

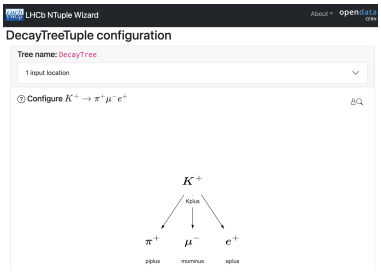
Whilst there is a lot of LHCb data available, this data is not in an easily accessible format, and trying to search for specific groups of data can be laborious. In order to overcome this issue with the Run 1 data, LHCb created the NTuple Wizard [8], which allows users to request the production of ntuples from LHCb



(a) Production configuration of the NTuple Wizard.



(b) Main decay search of the NTuple Wizard.



(c) DecayTreeTuple configuration of the NTuple Wizard.

Figure 3: Screenshots showing the NTuple Wizard in use.

data samples, simplifying an otherwise cumbersome process that requires large-scale computing resources and expertise [8]. Screenshots showing the NTuple Wizard can be seen in Fig. 3.

Similarly, in order to reduce this learning curve with simulations and decay files, this project works to build a Simulation Wizard, which allows users to access DecFiles and simulation samples more easily. Whilst these things are technically accessible, there is a large learning curve, with lots of room for user error. The Wizard hopes to remove learning barriers, both for Open Data users and for LHCb users who are less familiar with the relevant infrastructures. LHCb users will be able to access samples and put in simulation production requests. Open Data users will not be able to access these, but could generate their own samples using the Wizard output and the available LHCb software, provided they have the computer resources available. Producing LHCb simulation samples involves several steps that generate and process simulated data. Configuring this could most easily done with the Wizard output combined with LbMCSubmit, which takes a more basic file simulation "starter" file and produces a more complex one [9].

2 The LHCb Simulation Wizard

The main interface of the Simulation Wizard is the decay search, which is where the user starts. After searching for a specific decay, the user can then choose a specific event type (i.e. DecFile) and from that either see an annotated version of the DecFile, view bookkeeping paths that lead to simulation samples, or create a simulation production request. This workflow can see in Fig. 4.

The Wizard is built in React, and is made up of a number of components. These components and how they fit together can be seen in Fig. 5. The Simulation Wizard uses the NTuple Wizard as a dependency, and as such many of the components are taken directly from the NTuple Wizard (such as `SelectTag`, `SelectParticle`, and `DecayTag`), and others use child classes of NTuple Wizard classes (e.g. `DecfileDescriptorsSearch` is a child class of `DescriptorsSearch`). In order to do this it was necessary to rewrite some of the NTuple Wizard code in order to make it more modular, so that certain aspects of the classes could be overwritten whilst still keeping that general format.

Whilst the Simulation Wizard has the NTuple Wizard as a dependency and shares many characteristics with it, it is also quite different. In the Simulation Wizard, the core page of the web app is the search, seen in Fig. 7, whereas in the NTuple Wizard it's the production configuration (Fig. 3a). The key aspects of the Simulation Wizard, such as the annotated DecFile, the bookkeeping paths, and the simulation production request, can all be accessed from this search page. Additionally, the whilst the search itself is in terms of decays, the selection is in terms of event types.

2.1 Metadata

The metadata for the Simulation Wizard came from parsing all 7800 Decfiles using the decaylanguage package from scikit-hep [11]. There are three needed metadata files for the Simulation Wizard: one that

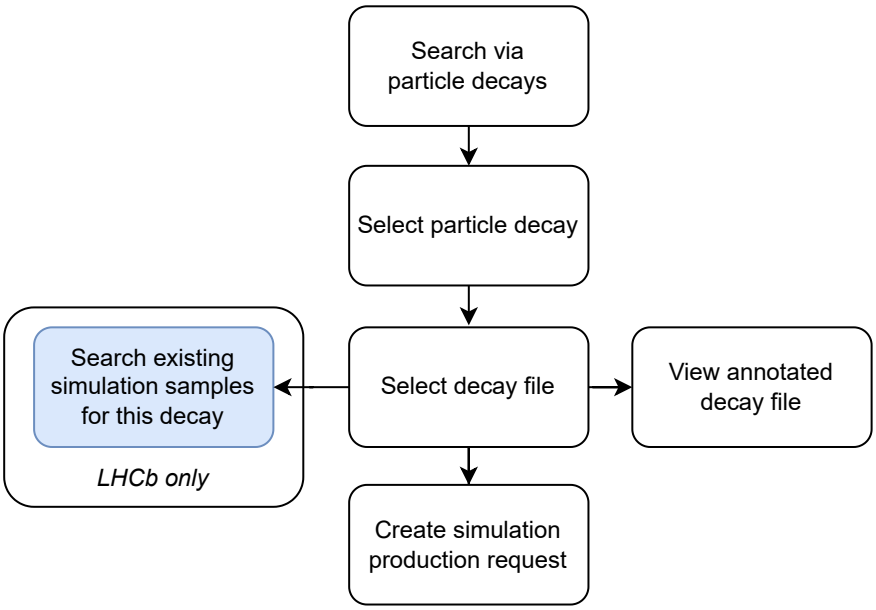


Figure 4: Flow chart showing the workflow of the Simulation Wizard web tool.The blue box indicates features only available for LHCb users. To note, the simulation production request is only the creation of the request — only LHCb users would then be able to submit it via the relevant repository [10].

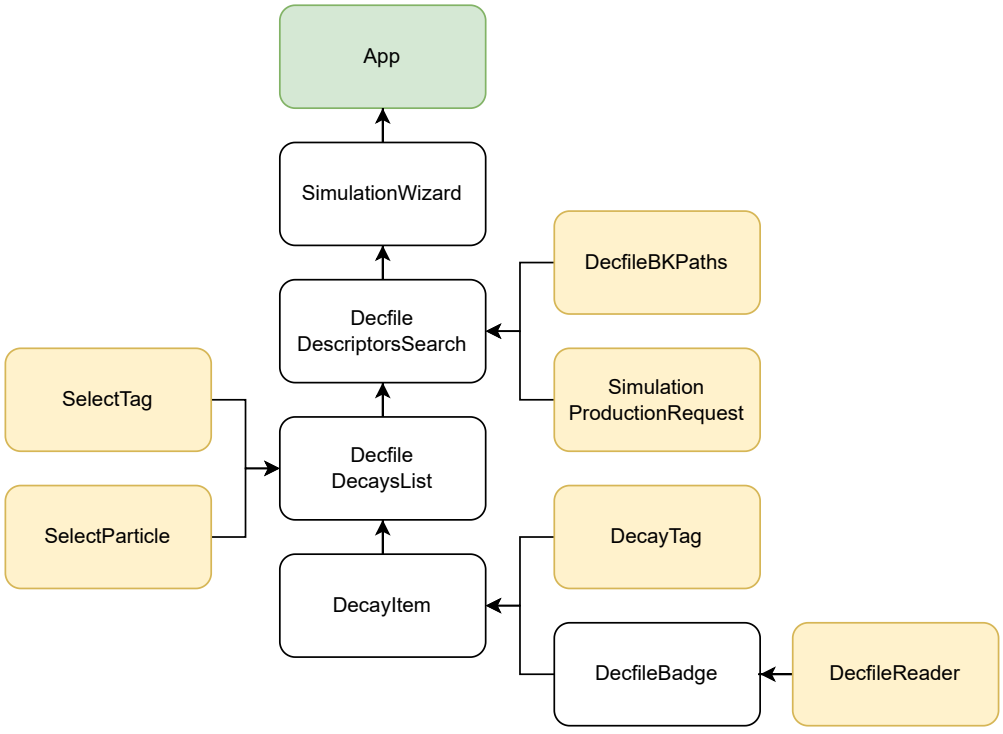


Figure 5: Flowchart showing how the different React components of the Simulation Wizard fit together. Yellow components have no child dependencies on other components, and the green *App* component does not feed into any other components.

```
"11102203": {
"text": "# EventType: 11102203  \n#\n#
Descriptor: [Beauty -> (K*(892)0 -> K+ pi-
) gamma]cc\n#\n#
NickName: Bd_Kstgamma=TightCut,PHSP\n#\n#
Cuts: LoKi::GenCutTool/TightCut\n#
CPUTime: 1 min\n#\n# ...",
"signal_particles": [
"B0",
"B~0"
],
"full_decays": [
"B0 -> (K*(892)0 -> K+ pi-) gamma",
"B~0 -> (K*(892)~0 -> K- pi+) gamma"
],
"production": "Pythia8",
"cuts": [
"LoKi::GenCutTool/TightCut"
]
},
```


(a) Example entry of `decfiles.json`. Note that the `"text"` entry has been cut short to save space, but this would contain the full raw text for the DecFile.


```
"K- -> e+ e- pi-": {
"descriptors": {
"plain": "K- -> e+ e- pi-",
"template": "K- ->
${eplus_1}e+ ${eminus_1}e- ${piminus}pi-",
"list": ["K-", ["e+", "e-", "pi-"]],
"mapped_list":
[{"branch": "Kminus_0", "particle": "K-"},
[{"branch": "eplus_2", "particle": "e+"},
{"branch": "eminus_2", "particle": "e-"},
{"branch": "piminus_0", "particle": "pi-
"}]],
"latex": "K^{-} \\rightarrow e^{+} e^{-
} \\pi^{-}"
},
"event_types": [
"37123000",
"37123001"
]
},
```


(b) Example entry of `simulation_decays.json`.

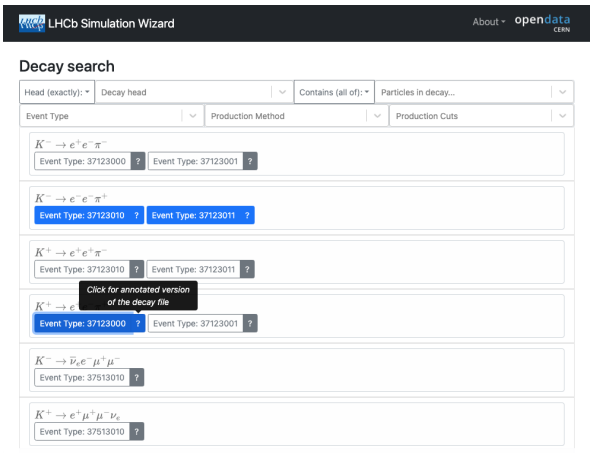
Figure 6: Examples of entries of the metadata files.

gives information on the different decays, called `simulation_decays.json`; one that gives information on different Decfiles, called `decfiles.json`; and one that gives the different bookkeeping paths for each DecFile, called `bkpaths.json`. The format of the latter is simply a list of bookkeeping paths for each event type, which is then processed further within the code. The format of the other two `.json` files can be seen in Fig. 6. We can see that a single decay, as seen in Fig. 6b, may have multiple linked event types, and a single DecFile (i.e. event type), as seen in Fig. 6a, may describe multiple decays.

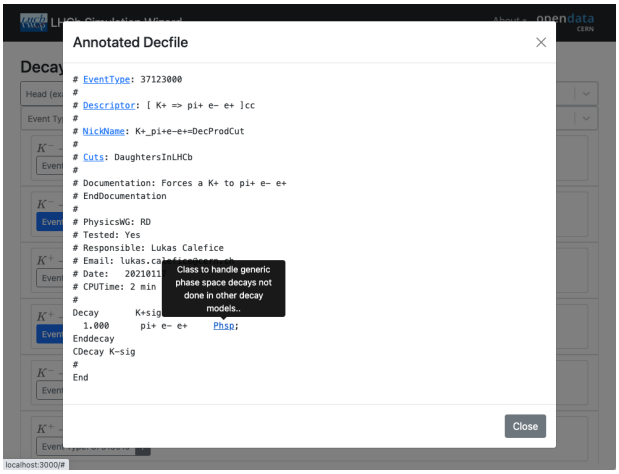
2.2 Decay search & annotated Decfiles

The Wizard decay search has various filters, which allow th user to narrow down the list of available decays. The filters include: decay head, particles in decay, event type, production method, and production cuts. The first two of these are decay specific, whilst the latter three are event type filters. The key feature to note here is that whilst the decay search is for decays themselves, and decays are what are shown in the result, the selection is via event type, i.e. DecFile. This can be seen in Fig. 7a, where the chosen event types are highlighted in blue.

In order to help the user chose the correct DecFile, next to the event type number is a small question mark, which when clicked shows an annotated version of the DecFile (seen in Fig. 7a). The annotations are on keywords in the DecFile syntax, and are a combination of a number of handwritten definitions, based on those given in the LHCb Starterkit [12], and also descriptions of models given in `.hh` files in the `EventGenModels` folder, which were scraped from those files with a `.py` code. An example of an annotated DecFile can be seen in Fig. 7b.

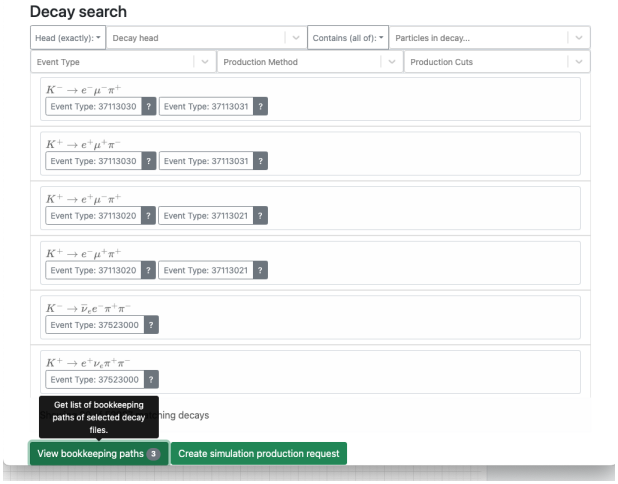


(a) Screenshot of decay search with some event types highlighted.

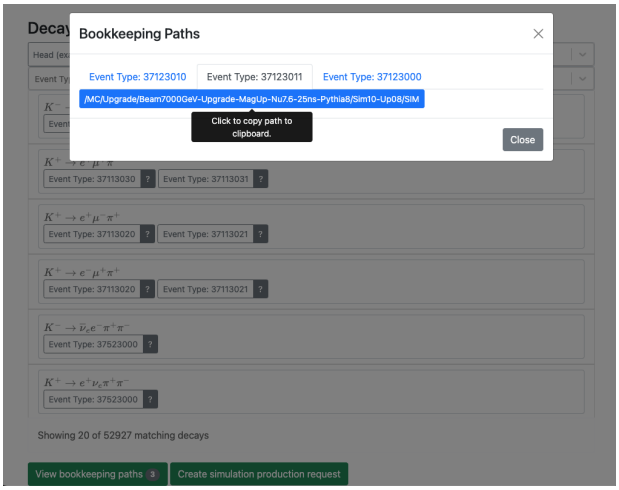


(b) Screenshot of annotated DecFile.

Figure 7: Screenshots of the decay search and annotated Decfiles.



(a) Screenshot of decay search with some event types highlighted.



(b) Screenshot of annotated DecFile.

Figure 8: Screenshots of the decay search and annotated Decfiles.

2.3 Bookkeeping paths

The component for the bookkeeping paths is partially built, as can be seen in Fig. 8. The button has a small badge that indicates how many events have been selected, and when the button is clicked it takes you to a modal with several tabs, one for each event type. In each tab there is a list of bookkeeping paths (in Fig. 8b we can see only one, but some events will have multiple simulations), and if the user clicks on a bookkeeping path it will be copied to their clipboard so they can access it. Although not currently implemented, it would be useful to add in filters to this modal, such as simulation version, year, collision type, etc. This information would also help users differentiate between the different paths.

2.4 Future developments

One of the key future developments is the creation of a mini-wizard that helps you produce a production request. Thise has not been started on yet, but the space for it is within the structure, and there is a button at the bottom of the decay search that will be used for it, as seen in Fig. 8a and in the component structure in Fig. 5. Eventually this button will open a modal or page within which there will be a "mini-wizard" that helps the user create a simulation production request. This would be in the format of a `stage0.yaml` file that can be used with LbMCSubmit [9] to create a higher level "stage 6" file which can be directly submitted to the MC requests repository [**lhcbsimulationMCRequestsGitLab2023**]. Submitting this request would be something that is obly available to LHCb users, however Open Data users do have access to all the necessary software (it is all publicly available) so could generate their own simulations if they have the required computational resources.

Another possible future development would be another mini-wizard that helps to user build a new DecFile, in the case where a searched for particle decay does not exist. Additionally, more filters could potentially be added, both in the main search and in the bookkeeping paths modal.

3 Conclusion

The LHCb Simulation Wizard will help improve the accessibility of LHCb decay files and simulation samples for both LHCb users and Open Data users. The current state of the Wizard has a useable search function (although additional filtering could be applied), as well as DecFile annotations, with the links to simulation samples and the ability to build production requests is currently under development. This tool is not only, but also helps LHCb be fully true to the CERN aim of making all research accessible and inclusive.

References

[1] John Back et al. *EvtGen Project*. EvtGen – Hepforge. 2020. URL: `https://evtgen.hepforge.org/` (visited on 08/11/2023).

[2] CERN. *CERN Open Science Policy*. Geneva: CERN, 2022. DOI: `10.17181/CERN.RTIF.Z8BO`.

[3] LHCb Computing. *The GAUSS Project*. LHCb Computing. 2023. URL: `https://lhcbdoc.web.cern.ch/lhcbdoc/gauss/` (visited on 08/11/2023).

[4] C. Bierlich et al. "A Comprehensive Guide to the Physics and Usage of PYTHIA 8.3". In: *SciPost Physics Codebases* (Nov. 10, 2022), p. 8. DOI: `10.21468/SciPostPhysCodeb.8`.

[5] LHCb Simulation. *LHCb Dirac*. GitLab. July 7, 2023. URL: `https://gitlab.cern.ch/lhcb-dirac` (visited on 08/11/2023).

[6] CERN. *Open Science at CERN: Home | OpenScience at CERN*. 2023. URL: `https://openscience.cern/` (visited on 08/10/2023).

[7] CERN. *CERN Open Data Portal*. 2023. URL: `https://opendata.cern.ch/` (visited on 08/10/2023).

[8] C. A. Aidala et al. "Ntuple Wizard: An Application to Access Large-Scale Open Data from LHCb". In: *Computing and Software for Big Science* 7.1 (Dec. 2023), p. 6. ISSN: 2510-2036, 2510-2044. DOI: 10.1007/s41781-023-00099-5. arXiv: 2302.14235 [hep-ex, physics:hep-ph].

[9] C. Burr. "LbMCSubmit: A New Flexible and Scalable Request Submission System for LHCb Simulation". In: 26th International Conference on Computing in High Energy & Nuclear Physics (CHEP2023). May 9, 2023.

[10] LHCb Simulation. *MC Requests*. GitLab. Aug. 9, 2023. URL: https://gitlab.cern.ch/lhcb-simulation/mc-requests (visited on 08/10/2023).

[11] E. Rodrigues and H. Schreiner. *DecayLanguage*. July 25, 2023. DOI: 10.5281/zenodo.3257423.

[12] LHCb. *The LHCb Starterkit Lessons — LHCb Starterkit Lessons Documentation*. URL: https://lhcb.github.io/starterkit-lessons/index.html (visited on 08/10/2023).

[13] A. Alexander Wight. *Poster Session 2023*. Indico. July 25, 2023. URL: https://indico.cern.ch/event/1293251/ (visited on 08/09/2023).

[14] A. Alexander Wight. *LHCb Tuesday Meeting*. Indico. Aug. 8, 2023. URL: https://indico.cern.ch/event/1310606/ (visited on 08/09/2023).

[15] A. Alexander Wight. *Student Sessions 2023*. Indico. Aug. 10, 2023. URL: https://indico.cern.ch/event/1293260/ (visited on 08/09/2023).

Appendices

A Poster

The poster seen in Fig. 9 was presented at the Summer Student Poster Session on the 25th July [13].

B Presentation

The presentation seen in Fig. 10 was given twice: once at the LHCb Tuesday meeting on the 8th August [14] and once at the Summer Student Sessions on the 10th August [15].

Making LHCb decay files and simulations easily accessible.

Alexia Alexander Wight, King's College London, UK

The LHCb Simulation Wizard

The LHCb Simulation Wizard is a web tool that simplifies the process of finding or producing decay files and simulation samples linked to them. The app is directed both at LHCb physicists and users of LHCb Open Data. The full workflow of the web tool can be seen in Figure 1. Whilst the decay files and LHCb software are currently public, the learning curve to use them is very steep: the aim of the Simulation Wizard is to make this learning curve a bit smaller via a user-friendly web tool. It also produces an annotated version of the decay files, which is helps users to understand the content of the files without needing prior expertise.

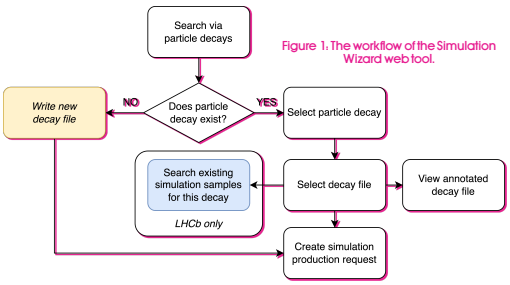


Figure 1: The workflow of the Simulation Wizard web tool.

Figure 2: A screenshot of the main decay search of the Simulation Wizard.

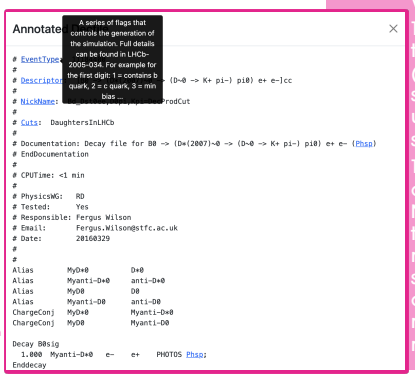


Figure 3: An example of the annotation of the decay files.

alexia.a.w@cern.ch
alexia.alexander.wight@kcl.ac.uk
@missneutrino

Open Data at CERN

Open data has been a core value at CERN from the start, when CERN's founding convention stated that all its results were to be published and made generally available. Today, this philosophy continues, with CERN's open data policy aiming to make all CERN research fully accessible, inclusive, democratic and transparent.

LHCb Wizards

Whilst there is a lot of LHCb data available at opendata.cern.ch, this data is not in an easily accessible format, and trying to search for specific groups of data can be laborious.

In order to overcome this issue, LHCb has made two wizards that allow researchers a more intuitive interface with which to find data, and also get the chance to request or work with LHCb software.

The Simulation Wizard allows users to search through decay files (publicly available) and simulation data (only for LHCb users), and allows them to create a simulation production request.

The Simulation Wizard is complementary to the NTupleWizard, which allows users to request the production of ntuples from LHCb data samples, simplifying an otherwise cumbersome process that requires large-scale computing resources and expertise.

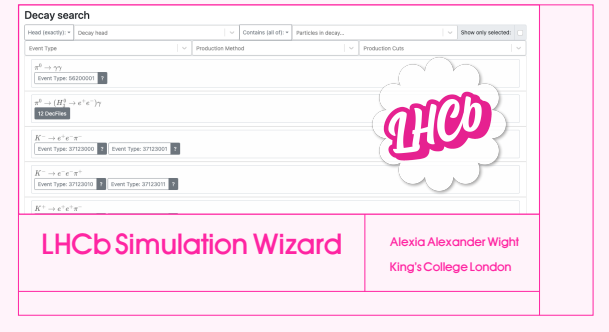
Thank you to my supervisors Adam Morris and Chris Burr for supporting me in this project, as well as to the CERN Summer Student programme and team for the privilege of being at CERN this summer.

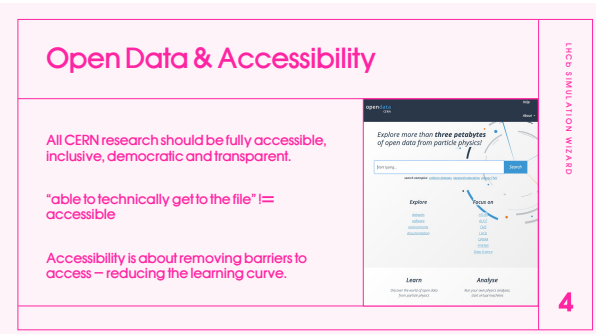
Aidala CA, Burr C, Cattaneo M, Fitzgerald DS, Morris A, Neubert S, et al. Ntuple Wizard: An Application to Access Large-Scale Open Data from LHCb. Comput Softw Big Sci (Internet). 2023 Dec (cited 2023 Jul 20);7(1):6. Available from: http://arxiv.org/abs/2302.14235

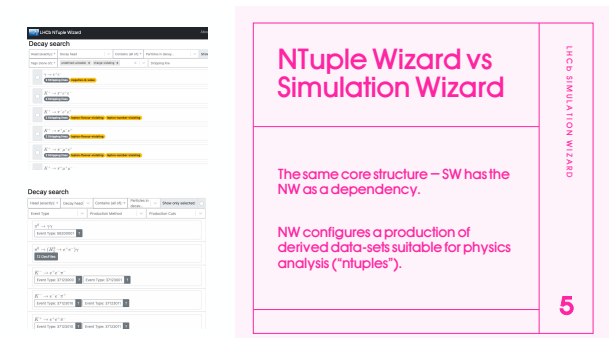
CERN Open Science Policy (Internet). Geneva: CERN; 2022 (cited 2023 Jul 20). Available from: https://cds.cern.ch/record/2835057

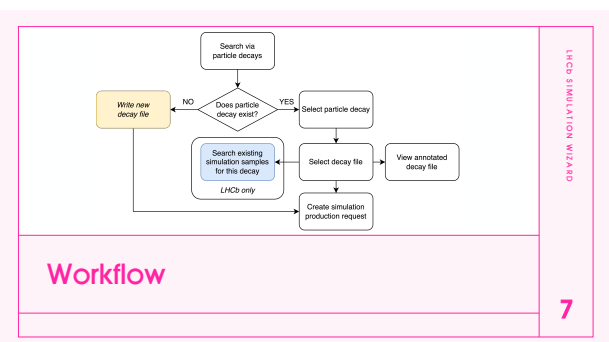
CERN (Internet). 2023 (cited 2023 Jul 20). CERN publishes comprehensive open science policy. Available from: https://home.cern/news/news/knowledge-sharing/cern-publishes-comprehensive-open-science-policy

Figure 9: Project poster from the Summer Student poster session.









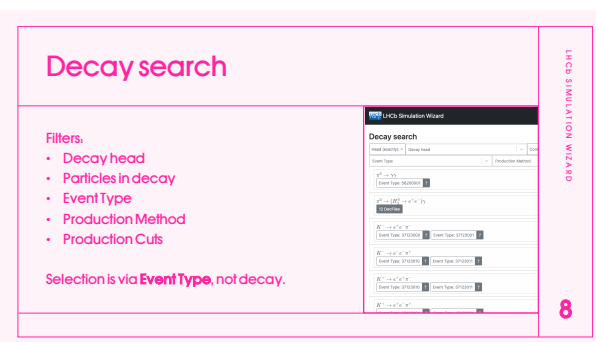


Figure 10: Slides from presentation given in the LHCb meeting on the 8th August.

Figure 10: Slides from presentation given in the LHCb meeting on the 8th August.