LHCb Starterkit 2022 - Practice Session
Part 1 - Producing your own Tuple

2^{nd} December 2022

Goals of the session

Starting from a DST files with simulated events that passed HLT2 selections, we want to tuple the information using DaVinci.

Decay of interest : $D^0 \to K^- \pi^+$

Simulation sample used :
`/MC/Upgrade/Beam7000GeV-Upgrade-MagDown-Nu7.6-25ns-Pythia8/Sim10aU1/27163003/XDIGI`

You can find the dst files (2 files with different statistics) and the tck.json files in this directory.
`/afs/cern.ch/work/r/reamalri/public/StarterKit/` PLEASE COPY THE FILES TO YOUR WORKING AREA.

1 Setting the configuration

Working from lxplus.

Create a file python `tupling.py`, that will contain the algorithm used to config DaVinci.

Create a function that will be called when executing DaVinci (typically `alg_config()`), taking `options` as argument.

This function return `make_config(options, algs)`. (This function needs to be imported from DaVinci).

To make things easier here are the list of imports we will be using.

```
from PyConf.components import force_location
from DaVinci.algorithms import add_filter
from DaVinci import make_config
from FunTuple import FunTuple_Particles as Funtuple
from FunTuple import FunctorCollection
import Functors as F
from FunTuple.functorcollections import (
    MCHierarchy,
    Kinematics,
    SelectionInfo,
    MCVertexInfo,
    MCKinematics,
    EventInfo,
)
from Functors.math import log

from DaVinci.truth_matching import (
    configured_MCTruthAndBkgCatAlg,
)
from DaVinci.reco_objects import make_pvs_v2
from DaVinci.algorithms import add_filter, get_odin, get_decreports
```

Now define the algorithm `algs = {"Tuple(NAME)": [FILTER,FunTuple]}`.
Require the filter to be "`Hlt2Charm_D0ToKmPip_XSec_Line`".
Now to the FunTuple, which has the following structure:

```
FunTuple = Funtuple(
    # give a name to the Tuple
    name= ... ,

    # give a name to the Tree
    tuple_name= ... ,

    # define fields -> identify the particle in the decay descriptor
    fields=fields,

    # assign varaibles to each of the particles
    variables=variables,

    # give the path in the dst where the information is
    inputs=force_location(f"/Event/HLT2/{line_name}/Particles"),

    # used to HLT level information (skipped here, feel free to ask)
    event_variables=None,
)
```

Advice for adding variables, start with Kinematics, test the whole thing with the options file (see later), and then slowly add other variables and MCTruth information.

2 Setting the options

Create the `yaml` file and fill with the following:

```
input_files: ... # the dst file
annsvc_config: PATH/hlt2.tck.json
input_type: ...
input_raw_format: ...
data_type: Upgrade
simulation: true
dddb_tag: ...
conddb_tag: ...
ntuple_file: ...
process: ...
```

3 Execution

To run the tupling:

```
lb-run DaVinci/v62r0p1 lbexec CONFIG OPTION
```