

DaVinci tupling for Run 3

Albert López Huertas

Universitat de Barcelona

LHCb Starterkit 2024



Time to have Fun(Tuple)!

[All options scripts](#)
([gitlab](#))

Run 1&2:

DecayTreeTuple based on *TupleTools*.

➤ Some of the variables added might not be used.

Run 3:

Define variables to be in the ntuple by their **Functors** → FunTuple!

```
import Functors as F
```

[Functors Reference](#)

[DaVinci Docs](#)

What's new

Run 1&2: DecayTreeTuple

```
from Configurables import DecayTreeTuple  
dtt = DecayTreeTuple('TupleDstToD0pi_D0ToKpi')
```

Run 3

```
from FunTuple import FunTuple_Particles as Funtuple
```

```
funtuple = Funtuple(  
    name="myTuple",  
    tuple_name="DecayTree",  
    fields=fields,  
    variables=branches,  
    inputs=data,  
    event_variables=event_info,  
)
```

What's new: branches

Run 1&2: DecayTreeTuple

```
dt.addBranches({
  'Dstar': '[D*(2010)+ -> (D0 -> K- pi+) pi+]CC',
  'D0': '[D*(2010)+ -> ^(D0 -> K- pi+) pi+]CC',
  'Kminus': '[D*(2010)+ -> (D0 -> ^K- pi+) pi+]CC',
  'piplus': '[D*(2010)+ -> (D0 -> K- ^pi+) pi+]CC',
  'pisoft': '[D*(2010)+ -> (D0 -> K- pi+) ^pi+]CC'
})
```

Run 3

```
fields = {
  "Dstar" : "[D*(2010)+ -> (D0 -> K- pi+) pi+]CC",
  "D0" : "[D*(2010)+ -> ^(D0 -> K- pi+) pi+]CC",
  "Kminus": "[D*(2010)+ -> (D0 -> ^K- pi+) pi+]CC",
  "Kplus" : "[D*(2010)+ -> (D0 -> K- ^pi+) pi+]CC",
  "pisoft": "[D*(2010)+ -> (D0 -> K- pi+) ^pi+]CC",
}
```

What's new: adding variables

Run 1&2: DecayTreeTuple

```
track_tool = dtt.addTupleTool('TupleToolTrackInfo')
```

Run 3

```
variables_tracks += FunctorCollection({  
    "GHOSTPROB": F.GHOSTPROB,  
    "NHITS": F.VALUE_OR(-1) @ F.NHITS @ F.TRACK,  
    "NVPHITS": F.VALUE_OR(-1) @ F.NVPHITS @ F.TRACK,  
    "NFTHITS": F.VALUE_OR(-1) @ F.NFTHITS @ F.TRACK,  
    "QOVERP": F.QOVERP @ F.TRACK  
})
```

```
branches = {  
    "ALL" : variables,  
    "Dstar" : variables_comp,  
    "D0" : variables_comp,  
    "piplus" : variables_tracks,  
    "Kminus" : variables_tracks,  
    "pisoft" : variables_tracks  
}
```

What's new: Inputs

Run 1&2: DecayTreeTuple

```
stream = 'AllStreams'  
line = 'D2hhCompleteEventPromptDst2D2RSLine'  
dtt.Inputs = ['/Event/{0}/Phys/{1}/Particles'.format(stream, line)]
```

Run 3

```
line = "Hlt2Charm_DstpToD0Pip_D0ToKmPip"  
data = get_particles(f"/Event/HLT2/{line}/Particles")
```

If using sprucing line: Spruce instead of HLT2

Finding MC in bookkeeping

Bookkeeping tree

- MC
- 2009
- 2010
- 2011
- 2012
- 2013
- 2015
- 2016
- 2017
- 2018
- 2022
 - 10000000 (incl_b)
 - 11102202 (Bd_Kstgamma=HighPtGamma,DecProdCut)
 - 11104021 (Bd_phiKst0=BabarAmp,DecProdCut)
 - 11106111 (Bd_LambdabarLambdapbar=HELAMP,TightCut)
 - 11144001 (Bd_JpsiKst,mm=DecProdCut)
 - 11144011 (Bd_psi2SKst,mm=DecProdCut)
 - 11154001 (Bd_JpsiKst,ee=DecProdCut)

EventType Advanced Refresh

Data quality

- 27163002 (Dst_D0pi,KK=DecProdCut)
- 27163003 (Dst_D0pi,Kpi=DecProdCut)
 - Beam6800GeV-2022-MagDown-NoUT-Nu2.1-25ns-Pythia8
 - Sim10b
 - HLT2-pp-commissioning
 - DST
 - GAUSSHISTMERGED

We'll use the $D^{*+} \rightarrow D^0(\rightarrow K^- \pi^+) \pi^+$ MC.

By now only 2022 has DSTs in simulation! Only processed by HLT2, not Sprucing

- Use Hlt2 as input stream.
- Run moore in simulation to obtain DSTs (HLT and Sprucing if needed).

Finding data in bookkeeping

Bookkeeping tree

- Collision22
- Collision23
 - 90000000 (Full stream)
 - Beam2680GeV-VeloClosed-MagDown-Excl-UT
 - Beam6800GeV-VeloClosed-MagDown-Excl-UT
 - Real Data
 - Sprucing23r1
 - B2CC.DST
 - B2OC.DST
 - BANDQ.DST
 - BNOC.DST
 - QEE.DST
 - RD.DST
 - SL.DST
 - Sprucing23r2
 - RAW
 - 90400001 (TURBORAW)
 - 93000000 (Luminosity stream online)
 - 94000000 (Turbo stream online)
 - Beam2680GeV-VeloClosed-MagDown-Excl-UT
 - Beam6800GeV-VeloClosed-MagDown-Excl-UT
 - Real Data
 - SprucingPass23r1
 - AnaProd-v0r0p6251450-Tuples_CharmProdAsym
 - B2CC.DST
 - B2OC.DST
 - BANDQ.DST
 - BNOC.DST
 - CHARM.DST
 - IFT.DST
 - QEE.DST
 - RD.DST
 - RAW
 - 95100000 (TURCAL)

"Hlt2Charm_DstpToD0Pip_D0ToKmPip"

The Turbo line belongs to Charm stream

What's new: yaml config

Run 1&2

```
# Configure DaVinci
DaVinci().UserAlgorithms += [dtt]

DaVinci().InputType = 'DST'
DaVinci().TupleFile = 'DVntuple.root'
DaVinci().PrintFreq = 1000
DaVinci().DataType = '2012'
DaVinci().Simulation = True
# Only ask for luminosity information when not using simulated data
DaVinci().Lumi = not DaVinci().Simulation
DaVinci().EvtMax = -1

# Use the local input data
from GaudiConf import IOHelper
IOHelper().inputFiles([
    './00035742_00000002_1.allstreams.dst'
], clear=True)
```

Run 3

```
input_raw_format: 0.5
input_type: ROOT
output_type: ROOT
input_process: TurboPass
input_stream: charm
simulation: False
data_type: "Upgrade"
geometry_version: run3/trunk
conditions_version: master
scheduler_legacy_mode: False
input_files:
| - root://x509up_u137380@eos.lhcb.cern.ch//eos/lhcb/grid/prod/lhcb/LHCb/Collision23/CHARM.DST/
print_freq: 1000
evt_max: 10000
ntuple_file: DV_ntuple.root
```

If running in detdesc specify `ddddb_tag` and `conddb_tag`
From sprucing line: `input_process: Spruce`

We'll use this DST available in EOS:

MC: root://x509up_u137380@eos.lhcb.cern.ch//eos/lhcb/grid/prod/lhcb/MC/2022/DST/00193293/0000/00193293_00000008_3.dst

Data: root://x509up_u137380@eos.lhcb.cern.ch//eos/lhcb/grid/prod/lhcb/LHCb/Collision23/CHARM.DST/00198130/0000/00198130_00002134_1.charm.dst

Lbexec vs gaudirun

Gaudirun

```
lb-run DaVinci/v46r10 gaudirun.py options.py
```

New for run 3: Lbexec

➤ Analysis Productions uses it

Lbexec

```
lb-run DaVinci/latest lbexec options:main options.yaml
```

What's new: ThOr Functors

Run 1&2: DecayTreeTuple

```
d_hybrid = dtt.Dstar.addTupleTool('LoKi::Hybrid::TupleTool/LoKi_D')
d_hybrid.Variables = {
    'dira': 'BPVDIRA',
    'vtx_chi2': 'VFASPF(VCHI2)',
    'max_pt' : 'MAXTREE(ISBASIC & HASTRACK, PT, -1)',
    'n_highpt_tracks': 'NINTREE(ISBASIC & HASTRACK & (PT > 1500*MeV))'
}
```

LoKi

Run 3

```
d_variables = {
    'dira': F.BPVDIRA(pvs),
    'vtx_chi2': F.CHI2,
    'max_pt': F.MAXTREE(F.ISBASICPARTICLE & (F.HAS_VALUE @ F.TRACK), F.PT),
    'n_highpt_tracks': F.NINTREE(F.ISBASICPARTICLE & (F.HAS_VALUE @ F.TRACK) & (F.PT > 1500*MeV)),
}
```

ThOr

[More! AllFunctors.py](#)

Adding HLT info

Functor collections for HLT decisions

```
Hlt1_decisions = ["Hlt1TrackMVA", "Hlt1TwoTrackMVA"]
variables+=FC.HltTisTos(
    selection_type="Hlt1",
    trigger_lines=[f"{x}Decision" for x in Hlt1_decisions],
    data=data)
```

Similar procedure for HLT2

Add global event info

```
event_info += FC.SelectionInfo(selection_type="Hlt1", trigger_lines=Hlt1_decisions)
```

Decay Tree Fitter

- The reconstruction in LHCb is made leaf-by-leaf: fit vertices most downstream and build up until decay head.
- Decay Tree Fitter: fit the entire decay chain globally. Can also apply *constraints*:
 - PV.
 - Mass.
 - Change particle hypothesis.

Run 2: LoKi algorithm & TupleTool

```
dtf.Dstar.addTupleTool('TupleToolDecayTreeFitter/ConsD')
dtf.Dstar.ConsD.constrainToOriginVertex = True
dtf.Dstar.ConsD.Verbose = True
dtf.Dstar.ConsD.daughtersToConstrain = ['D0']
```

```
LoKi_DTFFun = dtf.Dstar.addTupleTool("LoKi::Hybrid::TupleTool/LoKi_DTFFun")
LoKi_DTFFun.Variables = {
  "DTFFun_Dstart_P" : "DTF_FUN(P, True, 'D0')",
  "DTFFun_Dstar_PT" : "DTF_FUN(PT, True, 'D0')",
  "DTFFun_Dstar_M" : "DTF_FUN(M, True, 'D0')",
  "DTFFun_DTF_CHI2" : "DTF_CHI2(True, 'D0')",
  "DTFFun_DTF_NDOF" : "DTF_NDOF(True, 'D0')",
  "DTFFun_D0_M" : "DTF_FUN(CHILD(M, '[D*(2010)+ ->^D0 pi+CC']), True, 'D0')",
  "DTFFun_D0_PE" : "DTF_FUN(CHILD(E, '[D*(2010)+ ->^D0 pi+CC']), True, 'D0')",
  "DTFFun_D0_PX" : "DTF_FUN(CHILD(PX, '[D*(2010)+ ->^D0 pi+CC']), True, 'D0')",
  "DTFFun_D0_PY" : "DTF_FUN(CHILD(PY, '[D*(2010)+ ->^D0 pi+CC']), True, 'D0')",
  "DTFFun_D0_PZ" : "DTF_FUN(CHILD(PZ, '[D*(2010)+ ->^D0 pi+CC']), True, 'D0')"
}
```

[Starterkit lesson](#)

[More info](#)

Decay Tree Fitter

RUN 3: ThOr Algorithm!

```
from DecayTreeFitter import DecayTreeFitter

DTF_D0 = DecayTreeFitter(
    name="DTF_D0andPVconstr",
    input_particles=data,
    input_pvs=pvs,
    mass_constraints=["D0"]
)

DTF_subs = DecayTreeFitter(
    name="DTF_PVKtopi",
    input_particles=data,
    input_pvs=pvs,
    substitutions=["D*(2010)+ -> (D0 -> K- pi+{{K+}}) pi+"]
)

variables.update({"DTF_M": DTF_D0(F.MASS)})
variables.update({"DTF_M_subst_pi2K": DTF_subs(F.MASS)})
```

[More info](#)