

Table of Contents

DaVinci Tutorial 9b - Applying arbitrary functors to a refitted decay tree.....	1
---	---

DaVinci Tutorial 9b - Applying arbitrary functors to a refitted decay tree

For basic info see DecayTreeFitter.

This is a very advanced tutorial. Make sure you understand DaVinciTutorial9 and it's prerequisites.

Using the DTFDict the DecayTreeFitter will only be called once per candidate (not once per variable).

```
#####  
# The NEW Dictionary Tools - DTFDict example  
#####  
# Imports  
from Configurables import LoKi__Hybrid__DictOfFunctors  
from Configurables import LoKi__Hybrid__Dict2Tuple  
from Configurables import LoKi__Hybrid__DTFDict as DTFDict  
  
# Define some branches  
tuple.addBranches({  
    "Bs" : "B_s0 -> (phi(1020) -> K+ K-) J/psi(1S) ",  
    "Phi" : "B_s0 -> ^phi(1020) -> K+ K-) J/psi(1S) ",  
})  
  
# Let's start with something traditional: The traditional TupleTool  
LoKi_All = tuple.Bs.addTupleTool("LoKi::Hybrid::TupleTool/LoKi_All")  
LoKi_All.Variables = {  
    "lokiP" : "P",  
    "lokiPT" : "PT",  
    "lokiM" : "M",  
    "loki_Psi_M" : "CHILD(M, 'B_s0 -> phi(1020) ^J/psi(1S)')",  
    "loki_Psi_PE" : "CHILD(E, 'B_s0 -> phi(1020) ^J/psi(1S)')",  
    "loki_Psi_PX" : "CHILD(PX, 'B_s0 -> phi(1020) ^J/psi(1S)')",  
    "loki_Psi_PY" : "CHILD(PY, 'B_s0 -> phi(1020) ^J/psi(1S)')",  
    "loki_Psi_PZ" : "CHILD(PZ, 'B_s0 -> phi(1020) ^J/psi(1S)')",  
}  
  
# The old way to apply functors to a refitted decay tree was to use the DTF_FUN  
# with these functors you can get the values but the fit is rerun for every variable from scratch  
  
LoKi_DTFFun = tuple.Bs.addTupleTool("LoKi::Hybrid::TupleTool/LoKi_DTFFun")  
LoKi_DTFFun.Variables = {  
    "DTFFun_Bs_P" : "DTF_FUN(P, True, 'J/psi(1S)')",  
    "DTFFun_Bs_PT" : "DTF_FUN(PT, True, 'J/psi(1S)')",  
    "DTFFun_Bs_M" : "DTF_FUN(M, True, 'J/psi(1S)')",  
    "DTFFun_DTF_CH2" : "DTF_CHI2( True, 'J/psi(1S)')",  
    "DTFFun_DTF_NDOF" : "DTF_NDOF( True, 'J/psi(1S)')",  
    "DTFFun_Psi_M" : "DTF_FUN(CHILD(M, 'B_s0 -> phi(1020) ^J/psi(1S)'), True, 'J/psi(1S)')",  
    "DTFFun_Psi_PE" : "DTF_FUN(CHILD(E, 'B_s0 -> phi(1020) ^J/psi(1S)'), True, 'J/psi(1S)')",  
    "DTFFun_Psi_PX" : "DTF_FUN(CHILD(PX, 'B_s0 -> phi(1020) ^J/psi(1S)'), True, 'J/psi(1S)')",  
    "DTFFun_Psi_PY" : "DTF_FUN(CHILD(PY, 'B_s0 -> phi(1020) ^J/psi(1S)'), True, 'J/psi(1S)')",  
    "DTFFun_Psi_PZ" : "DTF_FUN(CHILD(PZ, 'B_s0 -> phi(1020) ^J/psi(1S)'), True, 'J/psi(1S)')",  
}  
  
# Now, let's start to build the tool chain for getting the refitted decay tree once and apply a  
  
# Start by adding the Dict2Tuple to the Bs branch - this will write the values we are going to re  
DictTuple = tuple.Bs.addTupleTool(LoKi__Hybrid__Dict2Tuple, "DTFTuple")  
  
# We need a DecayTreeFitter. DTFDict will provide the fitter and the connection to the tool chain  
# we add it as a source of data to the Dict2Tuple  
DictTuple.addTool(DTFDict, "DTF")  
DictTuple.Source = "LoKi::Hybrid::DTFDict/DTF"  
DictTuple.NumVar = 10 # reserve a suitable size for the dictionary
```

```

# configure the DecayTreeFitter in the usual way
DictTuple.DTF.constrainToOriginVertex = True
DictTuple.DTF.daughtersToConstrain = ["J/psi(1S)"]

# Add LoKiFunctors to the tool chain, just as we did to the Hybrid::TupleTool above
# these functors will be applied to the refitted(!) decay tree
# they act as a source to the DTFDict
DictTuple.DTF.addTool(LoKi__Hybrid__DictOfFunctors,"dict")
DictTuple.DTF.Source = "LoKi::Hybrid::DictOfFunctors/dict"

DictTuple.DTF.dict.Variables = {
    "DTFDict_Bs_PT"      : "PT",
    "DTFDict_Bs_M"      : "M",
    "DTFDict_Psi_PT"    : "CHILD(PT,1)",
    "DTFDict_Psi_M"     : "CHILD(M,'B_s0 -> phi(1020) ^J/psi(1S)')",
    "DTFDict_Psi_PE"    : "CHILD(E,'B_s0 -> phi(1020) ^J/psi(1S)')",
    "DTFDict_Psi_PX"    : "CHILD(PX,'B_s0 -> phi(1020) ^J/psi(1S)')",
    "DTFDict_Psi_PY"    : "CHILD(PY,'B_s0 -> phi(1020) ^J/psi(1S)')",
    "DTFDict_Psi_PZ"    : "CHILD(PZ,'B_s0 -> phi(1020) ^J/psi(1S)')",
}

#####
# End
#####

```

This topic: LHCb > DaVinciTutorial9b
 Topic revision: r7 - 2018-06-05 - JohannesHeuel



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback