Introduction to DaVinci

- Introduction
  - overview of DaVinci structure
  - Input
- My first Selection
  - Read $J/\psi \to \mu\mu$ all in python, no C++ to write

LHCb Week
25 March 2011

Patrick Koppenburg
NIKHEF

DaVinci Links

- DaVinci web page:
  http://cern.ch/LHCb-release-area/DOC/davinci/
  From there you'll find :
  - Some documentation. Links to doxygen.
  - The DaVinci wiki and Tutorial page
- Bugs and feature requests are tracked in the LHCb software Savannah page. If you think you see a bug, go there first.
- Any DaVinci question can be asked at the DaVinci mailing list: lhcb-davinci@cern.ch .
  - You need to be registered to use it. You can do that online.
- Distributed analysis question should be asked at lhcb-distributed-analysis@cern.ch .
- General software questions should go to lhcb-soft-talk@cern.ch

Always give all information that is relevant to your question! (no "It stopped working. What has changed?")

DaVinci

INPUT: ProtoParticles

`ProtoParticles:` are the end of the reconstruction stage

- have all the links about how they have been reconstructed
- have a list of PID hypothesis with a probability
- contain the *kinematic* information

CHARGED `ProtoParticles:` One per `Track`

➜ Add Rich, Calo, Muon info

NEUTRAL `ProtoParticles:` One per Cluster

➜ Add Track info (could be an electron)

Particles

- `Particle` = `ProtoParticle` + one PID *choice*
  - ➜ one defined mass
- Physics analyses deal with `Particles`
  - You need to know the 4-vectors to compute the mass of a resonance
- The PID is your choice
  - The same `ProtoParticle` can be made as a π and as a μ...
    - This makes sense. Think of a pion from $B \to \pi\pi$ decaying in flight. Does it stop being a signal pion because it decayed before the Muon detector?
  - Some `ProtoParticles` can be ignored
  - All this is done by configuring a `ParticleMaker` algorithm
    - You don't need to worry about the configuration.
    - Many standards are pre-defined
    - But you need to choose which to use
    - ➜ Next slide

STANDARD Particles

- The `Particles` are actually already done for you. To ensure that everybody agrees on what is a K^+, a π or a K^0_S, we have a set of standard particles predefined.
- They are defined in `python/CommonParticles/*.py` (svn) in the `Phys/CommonParticles` package.
- All you need to know are the names of the algorithm that created them : `StdLooseKaons`, `StdTightProtons` ...

  STDNOPIDSXXXX: All tracks are made to `Xxxx`

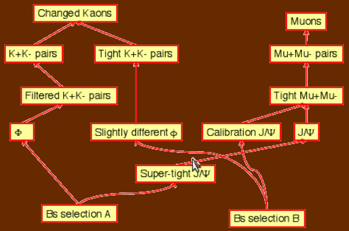
  STDLOOSEXXXX: Loose PID cuts for hypothesis `Xxxx` (no cuts for pions)

  STDTIGHTXXXX: Tight PID cuts for hypothesis `Xxxx`
- They are interfaces to the selection framework (see later)

```
from StandardParticles import StdLooseMuons, StdLoosePions
print StdLooseMuons.outputLocation()
```

The selection framework

- A selection is a sequence of algorithms reading in `Particles` and writing out other `Particles`
  - You want to make sure that the output of one algorithm is the input to another one
  - Some of this is enforced in C++, but not in python.
  - ✗ You can make mistakes



➜ Use a selection framework that tells you from the start that something is wrong

How to select Particles

There are actually several ways to quickly get a physics result:

- Plain C++: `DVAlgorithm` inherits from `GaudiAlgorithm` (and `GaudiTupleAlg` ...), some typing is saved
- LoKi: "loops and kinematics". Templated C++. More typing saved.
- GaudiPython, Bender: Interactive `python`.
- LoKi::Hybrid: **Used in Stripping.**

The common assumption is that physicists always do the same, hence any line of C++ you type is likely to be a duplication of what your office-mate is typing right now.

Example: Read J/ψ's

Using LoKi-Hybrid and the selection framework

What to do?

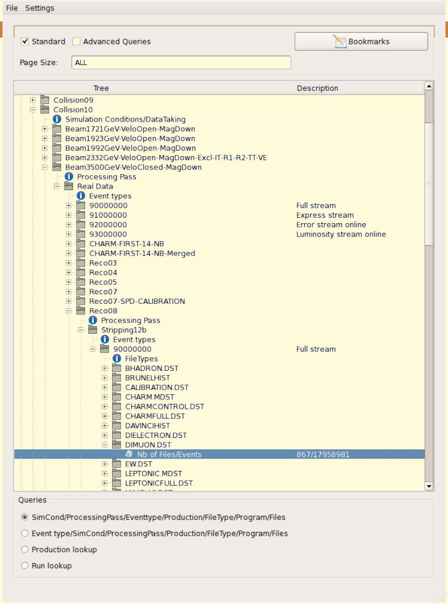
The workflow of you analysis should be:

1. Read your candidates
2. Refine your candidates (more cuts)
3. TisTos your candidate
    - If TIS, you're lucky ➜ keep it
    - If TOS, keep it only if Tossed by one of "your" lines
4. Store (or fit) it somehow
    1. DST
    2. MicroDST (better)
    3. `DecayTreeTuple` (not good, but sometimes useful)

➜ Make sure you know what you need before starting

The Data

- No reconstructed 2011 data available yet
- Look at `Collision10`. You can take magnet down or up. No off!
- Then `Reco08/ 90000000/ Stripping12b/ DIMUON.DST`
  - ➜ One file per run (if size permits)



How to Read your Candidates

I want the candidates of the tight $J/\psi \to \mu\mu$ selection

1. Let's look at the Dimuon stream. Do:

```
SetupProject DaVinci  v26r3p2
python
from StrippingSettings.Stripping12.StreamDimuon import stream
locations = {}
for line in stream.lines :
   locations[ line.name()] = stream.name() + '/' + line.outputLocation()

for line, loc in locations.iteritems() :
   if 'DiMuon' in line : print line, loc
...
StrippingNeuroBayesJPsiLine Dimuon/Phys/NeuroBayesJPsiLine
StrippingNeuroBayesMuMuLine Dimuon/Phys/NeuroBayesMuMuLine
StrippingDiMuonHighMassSameSignLine Dimuon/Phys/DiMuonHighMassSameSignLine
StrippingBd2KstarMuMu_SignalTriggered Dimuon/Phys/Bd2KstarMuMu_SignalTriggered
StrippingBs2MuMuNoMuIDLooseLine Dimuon/Phys/Bs2MuMuNoMuIDLooseLine
```

➜ Line `StrippingNeuroBayesMuMuLine` puts candidates in `/Event/Dimuon/Phys/NeuroBayesMuMuLine/Particles`

See https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbStripping

How to Read your Candidates

I want the candidates of the tight $J/\psi \to \mu\mu$ selection

1. Line `StrippingNeuroBayesMuMuLine` puts candidates in `/Event/Dimuon/Phys/NeuroBayesMuMuLine/Particles`
2. Actually, the candidates are moved there from `/Event/Phys/NeuroBayesMuMuLine/Particles`

See `https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbStripping`

How to Read your Candidates

I want the candidates of the tight $J/\psi \to \mu\mu$ selection

1. Line `StrippingNeuroBayesMuMuLine` puts candidates in `/Event/Dimuon/Phys/NeuroBayesMuMuLine/Particles`
2. Actually, the candidates are moved there from `/Event/Phys/NeuroBayesMuMuLine/Particles`
3. Try `PrintDecayTree`:

```
from Configurables import DaVinci, PrintDecayTree
DaVinci().appendToMainSequence( [ PrintDecayTree(
            Inputs = ['/Event/Dimuon/Phys/NeuroBayesMuMuLine/Particles' ])  ] )
```

```
PrintDiMuons.PrintDecay                                                                   INFO
<-------------------------------- Particle -------------------------------->
               Name          E          M          P         Pt        phi         Vz
                           MeV        MeV        MeV        MeV       mrad         mm
J/psi(1S)             91562.06    1758.35   91545.18    4543.92   -1294.93      -5.05
+-->mu+               32183.09     105.66   32182.92     961.42   -1736.07     -36.29
+-->mu-               59364.25     105.66   59364.16    3696.71   -1186.00     -24.25
```

See https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbStripping

How to Read your Candidates

I want the candidates of the tight $J/\psi \to \mu\mu$ selection

1. Line `StrippingNeuroBayesMuMuLine` puts candidates in `/Event/Dimuon/Phys/NeuroBayesMuMuLine/Particles`
2. Actually, the candidates are moved there from `/Event/Phys/NeuroBayesMuMuLine/Particles`
3. Try `PrintDecayTree`:
4. What if I have no candidates? Still stick to you line!

   You can check the result of the selection with

```
from Configurables import LoKi__HDRFilter   as StripFilter
MySequencer = GaudiSequencer('Sequence')
MySequencer.Members += [ StripFilter( 'StripPassFilter',
                         Code="HLT_PASS('StrippingNeuroBayesMuMuLineDecision')",
                         Location="/Event/Strip/Phys/DecReports" ) ]
```

   Note the location (the default is the HLT).

See https://twiki.cern.ch/twiki/bin/view/LHCb/LHCbStripping

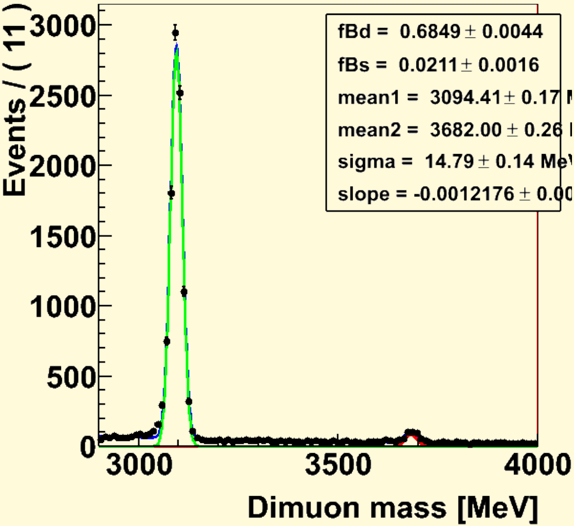
How to Refine your Candidates

```
from Gaudi.Configuration import *
line = 'NeuroBayesMuMuLine'
location = '/Event/Dimuon/Phys/'+line+'/Particles'
from Configurables import DaVinci

# get classes to build the SelectionSequence
from PhysSelPython.Wrappers import AutomaticData, Selection, SelectionSequence
# Get the Candidates from the DST.
# AutomaticData is for data on the DST
JpsiSel = AutomaticData(Location = location)
# Filter the Candidate.
from Configurables import FilterDesktop
_jpsiFilter = FilterDesktop('jpsiFilter', Code = '(M>2500*MeV) & (M<4000*MeV)')

# make a Selection
JpsiFilterSel = Selection(name = 'JpsiFilterSel',
                          Algorithm = _jpsiFilter,
                          RequiredSelections = [ JpsiSel ])

# build the SelectionSequence
JpsiSeq = SelectionSequence('SeqJpsi',
                             TopSelection = JpsiFilterSel,
                            )

DaVinci().appendToMainSequence( [ JpsiSeq.sequence() ] )
```

See https://twiki.cern.ch/twiki/bin/view/LHCb/ParticleSelection

How to Refine your Candidates

```
from Gaudi.Configuration import *
line = 'NeuroBayesMuMuLine'
location = '/Event/Dimuon/Phys/'+l
from Configurables import DaVinci

# get classes to build the Selecti
from PhysSelPython.Wrappers import
# Get the Candidates from the DST.
# AutomaticData is for data on the
JpsiSel = AutomaticData(Location =
# Filter the Candidate.
from Configurables import FilterDe
_jpsiFilter = FilterDesktop('jpsiF

# make a Selection
JpsiFilterSel = Selection(name = '
                          Algorith
                          Required


# build the SelectionSequence
JpsiSeq = SelectionSequence('SeqJp
                            TopSe
                           )


DaVinci().appendToMainSequence( [
```



See https://twiki.cer

LoKi-Hybrid

- The previous slide contained the line `Code = '(M>2500*MeV) & (M<4000*MeV)'`
- This is a bit of `python` that manipulates LoKi functors
  - Commonly used are `M, P, PT, TRCHI2`, but also `BPVLTCHI2, CHILDCUT`. Some take arguments, some don't.
  - But also `VFASPF(VCHI2/VDOF) ...`
  - The idea is that any *reasonable* selection requirement can be coded this way.
- The list can be viewed here : https://twiki.cern.ch/twiki/bin/view/LHCb/LoKiHybridFilters
- It's very sensitive to syntax orthodoxy. The parentheses are not optional. And it's &, not && or `and`.
- But as it's `python` it can be tested at the `python` prompt.
  ➜ Demo!

LoKi-Hybrid

- The previous slide containe
  (M<4000*MeV)'
- This is a bit of python tha
  - Commonly used are M,
    CHILDCUT. Some take ar
  - But also VFASPF(VCHI2
  - The idea is that any *rea*
    this way.
- The list can be viewed here
  //twiki.cern.ch/twiki/
- It's very sensitive to syntax
  optional. And it's &, not &
- But as it's python it can b
  - ➜ Demo!

```
from math import sqrt
from LoKiPhys.decorators import *
from LoKiCore.functions import monitor
p = LHCb.Particle()
p.setParticleID( LHCb.ParticleID(11) )
m = p.momentum()
m.SetPx (   1000 )
m.SetPy (  -1000 )
m.SetPz (  10000 )
m.SetE  ( sqrt( m.P2() + 5000*5000 ) )
p.setMomentum ( m )
fun = PX+PY
print PX(p), PY(p) , fun(p)
fun2 = PX>750
print fun2(p)
fun3 = monitor(fun2)
print fun3(p)
from LoKiCore.doxygenurl import browse
browse(PT)
```

How to Select a Trigger Line

I assume you already know which L0/Hlt1/Hlt2 lines you want

- You can use the `DecReports` to refine your selection

```
from Configurables import LoKi__HDRFilter   as HltFilter
MySequencer = GaudiSequencer('Sequence')
MySequencer.Members += [ HltFilter( 'HltPassFilter',
            Code="HLT_PASS('Hlt1Photon.*Decision')"  ]
```

See https://twiki.cern.ch/twiki/bin/view/LHCb/TriggerTisTos

How to Select a Trigger Line

I assume you already know which L0/Hlt1/Hlt2 lines you want

- You can use the `DecReports` to refine your selection
- But to really understand what the trigger is doing to your signal you need to TisTos it.
  - See Tomasz' tutorial
  - If you use `DecayTreeTuple`, configure `TupleToolTISTOS`

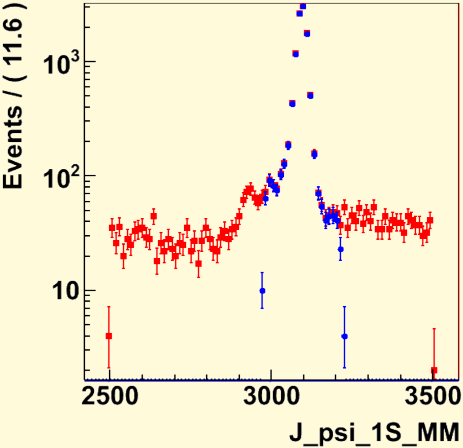
```
############# DecayTreeTuple
from Configurables import DecayTreeTuple, TupleToolTrigger,
tuple = DecayTreeTuple("Jpsi_Tuple")
tuple.ToolList +=  [
    "TupleToolGeometry"
    , "TupleToolKinematic"
    , "TupleToolPrimaries"
    , "TupleToolEventInfo"
    , "TupleToolTrackInfo"
    , "TupleToolTISTOS"
    , "TupleToolAngles"
    , "TupleToolPid"
    , "TupleToolPropertime"
    ]
tuple.Decay = "J/psi(1S) -> ^mu+ ^mu-"
tuple.InputLocations = [ pt, JpsiSeq.outputLocation() ]
tuple.addTool(TupleToolTISTOS)
tuple.TupleToolTISTOS.TriggerList = [ "Hlt2DiMuonUnbiasedJP
tuple.TupleToolTISTOS.VerboseHlt2 = True
##################

DaVinci().appendToMainSequence( [ JpsiSeq.sequence(), tuple

DaVinci().DataType = "2010"
DaVinci().EvtMax = -1
DaVinci().PrintFreq = 100
DaVinci().TupleFile = "Jpsi.root"
```

See https://twiki.cern.ch/twiki/bin/view/LHCb/TriggerTisTos

How to Select a Trigger Line

This shows the dimuon mass

- For B candidates that are `Hlt2Hlt2DiMuon-UnbiasedJPsi` triggered
  - TOS
  - all



See https://twiki.cern.ch/twiki/bin/view/LHCb/TriggerTisTos

CombineParticles

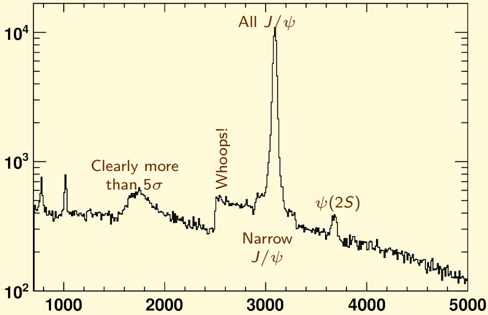
```
_motherCut = "(acos(BPVDIRA) < 0.01 & " \
             "(BPVIPCHI2() < 100"
_combinationCut = "ADAMASS('B0') < 500*MeV"
_Bd = CombineParticles(DecayDescriptor="[B0 -> K*(892)0 gamma]cc",
                       CombinationCut=_combinationCut,
                       MotherCut=_motherCut,
                       ReFitPVs=False)#True)
return Selection(name, Algorithm=_Bd,
                 RequiredSelections=[gammaSel, kstSel])
```

- `CombineParticles` combines `Particles` to produce new `Particles`
- That's what is used in the stripping
- You can also use it yourself
  - To add something to stripped candidates (look for $B^0 \to J/\psi\pi^0$ using NeuroBayes stripped J/ψ's and π^0 from `CommonParticles`)
  - To develop a new stripping line on MC
  - But not to redo combinatorics on a stripped DST!

Observation of a Resonance at 1.8 GeV

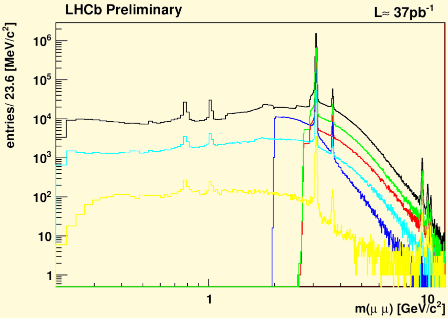
This is the dimuon spectrum on the dimuon stream with $\mu\mu$ combinatorics redone

It is biased by all selections. Including "trivial" ones (thresholds) and complicated one, like $B \to \mu\mu K^*$.



Observation of a Resonance at 1.8 GeV

- Use the candidates from one of the dimuon lines
- If you want $X \to (\mu\mu) Y$ you may make the Y yourself **if** you use the $(\mu\mu)$ from the stipping
- For any other use-case, use the MB stream
- The same applies to HLT



NEVER REDO COMBINATORICS ON A STREAM!

You will not be able to unfold the biases

ALWAYS USE THE CANDIDATES!

The only exception is the MB stream

NEVER RUN ON SEVERAL STREAMS MIXED-UP!

Exercises!

Ex. 0: **Set things up**

Ex. 1: Loop over muons and make some plots

Ex. 2: Extend the algorithm to make a J/ψ (if you have time)

Ex. 3: Make your algorithm more generic: select also a ϕ

Do Ex. 1 to 3 if you plan to develop `C++` in DaVinci.

Ex. 4: The recommended way of writing a selection

- Everything you need is on the wiki page
- The main difficulty is to figure out what to copy-paste where.
- Don't be afraid to ask if you are unsure

Ex. 5: Debugging

Ex. 6: MC truth, Trigger, Tagging, and much more

Ex. 7: More Tuples

Ex. 8: **Read Stripped DSTs**

Backup

Making a Selection

Any (flavour-) physics analysis is a sequence of A $\rightarrow$ B C (...), with some cuts in between.

Hence: An analysis algorithm should know:

Making a Selection

Any (flavour-) physics analysis is a sequence of A $\rightarrow$ B C (...), with some cuts in between.

Hence: An analysis algorithm should know:

1. Where to get the particles

4. Where to put the data

Handled by `DVAlgorithm`

Making a Selection

Any (flavour-) physics analysis is a sequence of A → B C (...), with some cuts in between.

Hence: An analysis alg

1. Where to get the
2. What decay to re
4. Where to put the

```
for { LHCb::Particle::ConstVector::const_iterator mK = KMinus.begin() ;
      mK != KMinus.end() ; ++mK ){
  for { LHCb::Particle::ConstVector::const_iterator pK = KPlus.begin()
        pK != KPlus.end() ; ++pK ){
    for { LHCb::Particle::ConstVector::const_iterator pi = Pions.begin(
        pi != Pions.end() ; ++pi ){
     [...]
```

Can be shorter:

```
for ( Loop Ds = loop( "K K pi", "D_s+", FitVertex ) ; Ds ; ++Ds ) {
```

Or:

```
DsForBs2DsPi.DecayDescriptor = "[D_s+ -> K+ K- pi+]cc"
```

Making a Selection

Any (flavour-) physics analysis is a sequence of A $\rightarrow$ B C (...), with some cuts in between.

Hence: An analysis algorithm should know:

1. Where to get the particles
2. What decay to reconstruct
3. What cuts to apply
4. Where to put the data

Hard-coding cuts is a bad idea...

Better to use options of the algorithm

...or predefined filters configurable by options:

```
DsForBs2DsPi.MotherCut = "PT > 2000*MeV"
```

Anything else?