

Heavy Flavor Tools

New Modules and Updates

Cameron Dean

04/19/2021

HF TG Meeting

- MDC1 $c\bar{c}$ and $b\bar{b}$ production has finished
 - **A min-bias production macro has been developed and is getting tested locally now**
- Backwards-compatibility of software and MDC DSTs became an issue
 - **I produced some new modules as a work around**
 - **They may also function as general efficiency tools**
- This presentation: two new tools and KFParticle updates

New tool 1: Decay Finder

- Backwards compatibility caused problems with associating reconstructed objects to generated objects
 - Somewhat solution: Avoid reco. and only use HepMC record
- The idea of [DecayFinder](#)
 - Write a decay string which can be interpreted
 - Allow or reject association photons and π^0
 - Skip over resonances such as J/ψ or $\phi(1020)$
 - Trigger further processing if decay exists
 - Run basic acceptance for efficiency

New tool 1: Decay Finder

- Create as many finders as you like for efficiency study
 - Only make one if you want to trigger as the rest of the processing gets skipped if the decay isn't there!
- The finder will only be registered if the string can be parsed
 - Run a small test first to ensure this, Fun4All will still run
- List of particles that can be understood is read from KFParticle_particleList.cc
- There is an internal list of resonances to skip over
 - If you use a particle from this list in your string, it will be removed from the skip-list

```
DecayFinder* myFinder = new DecayFinder("MyDecayFinder");  
myFinder->Verbosity(verbosity);  
myFinder->setDecayDescriptor(B_s0 -> {J/psi -> mu^+ mu^-} {phi -> K^+ K^-});  
myFinder->allowPi0(true);  
myFinder->allowPhotons(true);  
myFinder->triggerOnDecay(true);  
se->registerSubsystem(myFinder);
```

New tool 1: Decay Finder

Decaying particles
are left of ->

`[B+ -> {D0 -> pi^+ K^-} pi^+]cc`

Intermediate decays
are kept inside {}

Declare a charge-
conjugate search by
putting descriptor
inside []CC (case-
insensitive)

Charges are to the
right of ^
(only works to the right
of -> and accepts
neutrals)

- If a particle is in my resonance list and your string, it will be removed from my list
- While untested, in theory just writing a mother with -> will find all instances of that mother in the HepMC record

New tool 1: Decay Finder

```
Fun4AllServer::run - processing event 47541 from run 0
Fun4AllServer::End: End for MyD02KpiFinder
```

```
-----DecayFinder information-----
```

```
Module name: MyD02KpiFinder
```

```
Decay descriptor: [D0 -> kaon^- pion^+]cc
```

```
Number of generated decays: 6962
```

```
    Number of decays that failed pT requirement: 788
```

```
    Number of decays that failed eta requirement: 1358
```

```
    Number of decays that failed pT and eta requirements: 777
```

```
Number of decays that could be reconstructed: 4039
```

```
-----
```

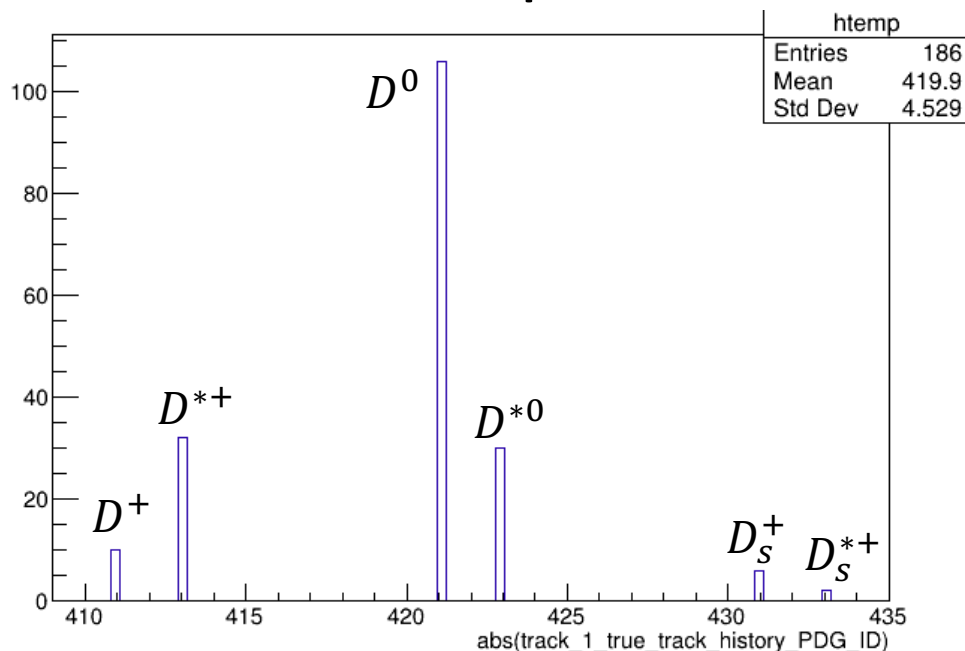
New Tool 2: The anti-trigger

- This is essentially the opposite the the decay finder
- You supply a list of particles, if any of them exist in the HepMC record the event is skipped
- Sounds strange, why make this?
 - I locally produced a min-bias sample for study but only wanted background
 - I used [this module](#) to reject any event with a heavy flavor particle

```
AntiTrigger* myFinder = new AntiTrigger("myTestAntiTrigger");  
myFinder->Verbosity(verbosity);  
std::vector<std::string> particleList =  
    {"D0", "D+", "Ds+", "Lambdac+", "B+", "B0", "Bs0", "Lambdab0"};  
myFinder->setParticleList(particleList);  
se->registerSubsystem(myFinder);
```

Adding truth MC history

- Han-Sheng requested we have the whole decay history of a track
 - Implemented separately for each reconstructed track, should then be possible to check if all tracks are from same parent



N.B. The information here is a vector so we can see potential parents to the D^0 . D^+ and D_s^+ is probably partially reconstructed

Decay parser

- I added a version of the DecayFinder string parser to KFParticle this morning
 - Local testing should now be finished
 - After some final code clean up a PR will be made, probably tomorrow
- NOTE: KFParticle will not be registered if your string cannot be parsed!

```
decayDescriptor = "[D*+ -> {D0 -> K^- pi^+} pi^+cc";  
//D*->D0(->Kpi)pi  
if (reconstructionChannel["Dstar2D0pi"])  
{  
    kfparticle->setMinimumMass(1.9);  
    kfparticle->setMaximumMass(2.1);  
    kfparticle->constrainToPrimaryVertex(true);  
  
    if (use_decay_descriptor)  
    {  
        kfparticle->setDecayDescriptor(decayDescriptor);  
    }  
    else  
    {  
        kfparticle->setMotherName("Dstar");  
        kfparticle->setNumberOfTracks(3);  
        kfparticle->hasIntermediateStates(true);  
        kfparticle->getChargeConjugate(true);  
  
        kfparticle->setNumberOfIntermediateStates(1);  
        intermediateList[0] = make_pair("D0", 0);  
        nIntTracks[0] = 2;  
        daughterList[0] = make_pair("kaon", -1);  
        daughterList[1] = make_pair("pion", +1);  
        daughterList[2] = make_pair("pion", +1);  
    }  
  
    intermediateMassRange[0] = make_pair(1.75, 1.95);  
    intPt[0] = 0.2;  
    intIP[0] = 0;  
    intermediateIPchi2Range[0] = make_pair(0., 100.);  
    intDIRA[0] = 0.98;  
    intFDchi2[0] = 0;  
}
```

- Min-bias production macro, based on MDC1 production, is being locally tested
 - **Hopefully ask for production at tomorrow's software meeting**
- Updates to KFParticle to get more truth info and better secondary selection are ready to go
 - **Local testing shows truth-matching is compatible with ana.241**
- Just finishing testing of decay descriptors to simplify UI
- Two new tools [are available](#) to help with efficiency and/or background studies
- [Twiki](#) and [examples](#) have already been updated