LCFIVertex tutorial – Flavour Tag Inputs

Erik Devetak – Oxford University (Orsay 03-05-07)
On behalf of the LCFI Collaboration

Flavour Tag Inputs Processor

- The flavour tag input processor is: **FlavourTagInputsProcessor**
  LCFIVertex/src/FlavourTagInputsProcessor.cc

- Default steering file is provided: **LCFIVertex/steering/fti.xml**
  Run command: Marlin fti.xml

- Slcio input file:
  zvresout.slcio (default – output of ZVTOPZVRESProcessor)
  alternatively one can use a different vertexing algorithm

- Input Collections:
  - JetRPCollection - collection representing the jets in the events
  - DecayChainRPCollection - DecayChain objects, matched to jets by ordering
  - IPVertexCollection (optional) - per event IP provided by PerEventIPFitter

Variables - Output

- Flavour tag variables calculated in separate classes. Advantages:
    - **Flexible code**
    - **Easy to read**
    - **Easy to add remove flavour tag inputs**

- All algorithm classes inherit from the algo.h. LCFI template for all algorithm code.

- There are flavour tag variables algorithms + helper ones:
    - procedures used by more than one flavour tag input
    - complex enough to be stand alone calculations

- **For a full list of Flavour tag input variables and algorithms please consult the documentation.**

- Output Collection: **FlavourTagInputsCollection –** collection holds all the calculated variables + vertex charge. **Same ordering as Jets collection**
Slcio output file: ftiout.slcio (default)

Parameters

- Excluding the input/output collection all parameters in FlavourTagInputsProcessor are parameters of the algorithm classes.

  - All these parameters are optional

  - If not present in steering files automatically given default value

  - All parameter names have a prefix indicating the object were used

Can be different values

Parameter	Object	Class
TrackAttachLoDCutmax	TrackAttach	TrackAttach
BChargeLoDCutmax	BAttach	TrackAttach
CChargeLoDCutmax	CAttach	TrackAttach

- **For further information on all available parameters please refer to the Processor or algorithm classes documentation.**

How to - Add Inputs

- Write the calculation algorithm in a separate class.

- LCFI standard inherits from algo.h. This provides:
  - Parameters handling (pointer double or string)
  - calculateFor - calculation routine called in processor

- **Once the algorithm is written:**
  - Initialize it at processor level
  - Set all needed parameters

```
_SecVertexProb = new SecVertexProb();
MemoryManager<Algo<DecayChain*, double > > ::Run()->registerObject(_SecVertexProb);
_SecVertexProb->setDoubleParameter("Chisquarecut",_SecondVertexProbChisquarecut);
_SecVertexProb->setDoubleParameter("Ntrackscut",_SecondVertexNtrackscut);
```

  - Add Variable name to _JetVariableNames and fill vaues in FlavourTagInputs. **Remember order matters!! Make sure you add the name and the fill values in the same position of the vectors!!**
  - You might want to define the parameters in the steering file. Use Marlin method: registerOptionalParameter.

Accessing Flavour Tag Inputs

- Names of flavour tag input variables are determined in lcio run header

- Values are filled and stored by order matching of the array

- However LCFI provides a more secure method for accessing variables by using name of variable

```
258        double NumVertices = (*FTInputs)[_IndexOf["NumVertices"]];
```

- Instead of “NumVertices” one can insert the name of any variable contained in the FlavourTagInputsCollection.

- This procedure is implemented in all processors downstream from FlavourtagInputsProcessor.

- **More details and list of variables can be found in documentation**

True Angular Jet Flavour

AIM: Find the “True Flavour of the Jet” – using MC PID data.

- The flavour tag input processor is: **TrueAngularJetFlavourProcessor**
  LCFIVertex/src/TrueAngularJetFlavourProcessor.cc

- Default steering file is provided: **LCFIVertex/steering/truejetflavour.xml**
  Run command: Marlin truejetflavour.xml (copy the xml file in Marlin/bin)

- Slcio input file: **cheatout.slcio (default – from tracking and jet finder)**

- Input Collections - Parameters:
  - JetRPCollection - collection representing the jets in the events
  - MCParticleCollection – collection containing MC particles of the event
  - MaximumAngle – Maximum angle for MC - closest jet matching

- Output Collection: **TrueJetFlavourCollection** – collection holds true jet flavours.

Same ordering as Jets collection. Slcio output file: jettypeout.slcio (default)