PIDTools for Particle ID

Masakazu Kurata

The university of Tokyo

06/24/2015

INTRODUCTION

- We have released the first version of PIDTools for ilcsoft v01-17-07
- PIDTools module includes:
  - dE/dx calculation using TPC hit information
  - Shower profile extraction using hits in calorimeter
  - Particle Identification using all the information of tracks
- We will show usage of outputs of PIDTools modules

dE/dx

- Need to reconstruct Mokka simulation, because any information about TPC hits is lost in current public DST files.
  - After FullLDCTracking_MarlinTrk is necessary
  - Include the processor in steer file:

```
    <!-- ========== dE/dx   ================= -->
    <processor name="MyCompute_dEdxProcessor"/>
```

```
<!---  dedx calculation processor --->
<processor name="MyCompute_dEdxProcessor" type="Compute_dEdxProcessor">
  <!--Energy Loss Error for TPC-->
  <parameter name="EnergyLossErrorTPC" type="float">0.05 </parameter>
  <!--Cut on Opening Angle for merging Si and TPC segments-->
  <parameter name="LDCTrackCollection" type="string"> MarlinTrkTracks </parameter>
</processor>
```

  - Just use getdEdx() in EVENT::Track collection:

    like trk->getdEdx();

    trk->getdEdxError(); //dEdx error, but simple 5%estimation

SHOWER PROFILE

- Need to reconstruct Mokka simulation, because any information about Cluster hits is lost in current public DST files.
  - After MarlinPandora is necessary
  - Include the processor in steer file:

```
<!-- ========== Shower Profiles   ================= -->
<processor name="MyComputeShowerShapesProcessor"/>
```

```
<processor name="MyComputeShowerShapesProcessor" type="ComputeShowerShapesProcessor">
 <!--Performs Shower profile extraction-->
 <!--Debugging?-->
 <parameter name="Debug" type="int">0 </parameter>
 <!--Name of the PFO collection-->
 <parameter name="PFOCollection" type="string"> PandoraPFOs </parameter>
 <!--Name of the Cluster collection-->
 <parameter name="ClusterCollectionName" type="string"> PandoraClusters </parameter>
 <!-- Radiation Length of Ecal-->
 <parameter name="RadiationLength_Ecal" type="float">3.50 </parameter>
 <!-- Radiation Length of Hcal-->
 <parameter name="RadiationLength_Hcal" type="float">17.57 </parameter>
 <!-- Moliere radius of Ecal-->
 <parameter name="MoliereRadius_Ecal" type="float">9.00 </parameter>
 <!-- Moliere radius of Hcal-->
 <parameter name="MoliereRadius_Hcal" type="float">17.19 </parameter>
 </processor>
```

Shower profile

- Need to reconstruct Mokka simulation, because any information about Cluster hits is lost in current public DST files.
  - Just use getShape() in EVENT::Cluster collection:

    like clu->getShape();
  - getShape() returns FloatVec, so there are some variables as results of shower shape fitting
  - Useful info is:

    getShape()[0]: fitting $\chi 2$

    getShape()[1]: maximum energy deposit(GeV)

    getShape()[2]: showerMax(mm)

    getShape()[3]: transverse absorption Length(mm)

    (distance between shower axis and the point where energy deposit reduce to 1/e)

    getShape()[16]: xl20(mm)

    (distance at the point where accumulated energy deposit is 20% of total energy deposit along shower axis(from cluster start))
- So far, not so sophisticate, just for PID

PARTICLE ID

- Need to reconstruct Mokka simulation, because any information about TPC hits and Cluster hits are lost in current public DST files.
  - Include the processor in steer file:

```
  <!-- ========== particle ID ====================
  <!--processor name="MyPFOID" /-->
  <processor name="MyLikelihoodPID" />
```

```
<processor name="MyLikelihoodPID" type="LikelihoodPIDProcessor">
 <!--Performs particle identification-->
 <!--Debugging?-->
 <parameter name="Debug" type="int">0 </parameter>
 <!--Boundaries for energy binning-->
 <parameter name="EnergyBoundaries" type="FloatVec">0 1.0e+07  </parameter>
 <!--Name of files containing pdfs for charged particles-->
 <parameter name="FilePDFName" type="StringVec">
   LikelihoodPID_Histogram_v01.root
 </parameter>
 <!--Name of the PFO collection-->
 <parameter name="RecoParticleCollection" type="string">PandoraPFOs</parameter>
</processor>
```

  - PDF file(Likelihood_Histogram_v01.root) is necessary to calculate likelihood

Particle ID

- Particle ID outputs are stored to ParticleID class, so we can use standard method:

  - Direct access(I think it is easy…)

```
lcio::ReconstructedParticle *pfo;                //get PFO particles
pfo->getParticleIDs()[0]->getPDG();              //get absolute PDG value(no charge)
pfo->getParticleIDs()[0]->getLikelihood();       //get posterior prob.
pfo->getParticleIDs()[0]->getParameters()[0]; //electron hypothesis
pfo->getParticleIDs()[0]->getParameters()[1]; //muon hypothesis
pfo->getParticleIDs()[0]->getParameters()[2]; //pion hypothesis
pfo->getParticleIDs()[0]->getParameters()[3]; //kaon hypothesis
pfo->getParticleIDs()[0]->getParameters()[4]; //pion hypothesis
```

  - So far Particle ID method is only one, so getParticleIDs()[0] is enough
  - But, in next version, it is not guaranteed

PARTICLE ID

- Particle ID outputs are stored to ParticleID class, so we can use standard method:

  - Using PIDHandler(easy?)

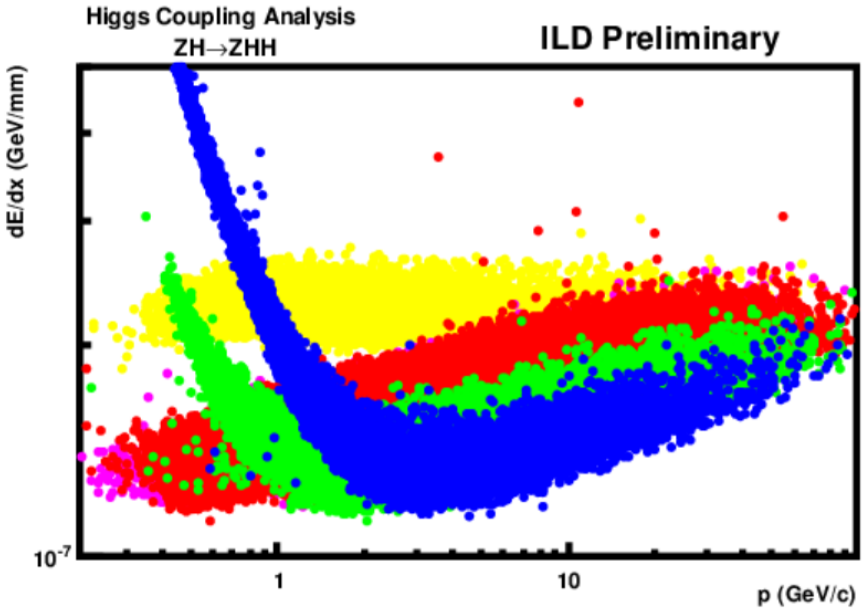
```
PIDHandler *pidh= new PIDHandler(pfocol); //register PFO collection to PIDHandler
lcio::ReconstructedParticle *pfo;  //get PFO particles
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getPDG(); //get absolute PDG value(no charge)
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getLikelihood(); //get posterior prob.
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getParameters()[0]; //electron hypothesis
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getParameters()[1]; //muon hypothesis
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getParameters()[2]; //pion hypothesis
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getParameters()[3]; //kaon hypothesis
pidh->getParticleID(pfo, pidh->getAlgorithmID("LikelihoodPID")).getParameters()[4]; //proton hypothesis
```

  - Seems safe for extension of PID

PLOTS

- @500GeV
- Momentum dep. of dE/dx

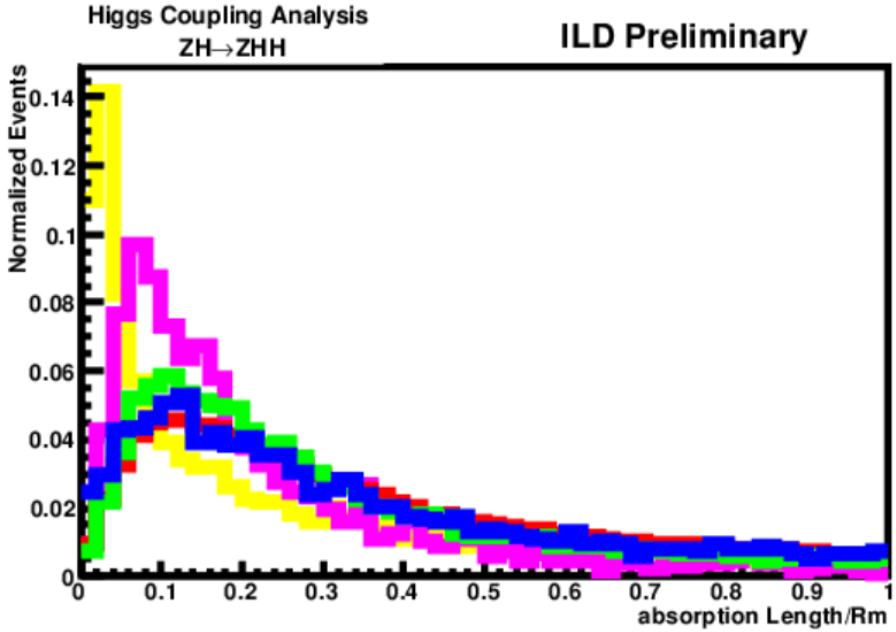
  Electron
  Muon
  Pion
  Kaon
  Proton



- Shower profile:

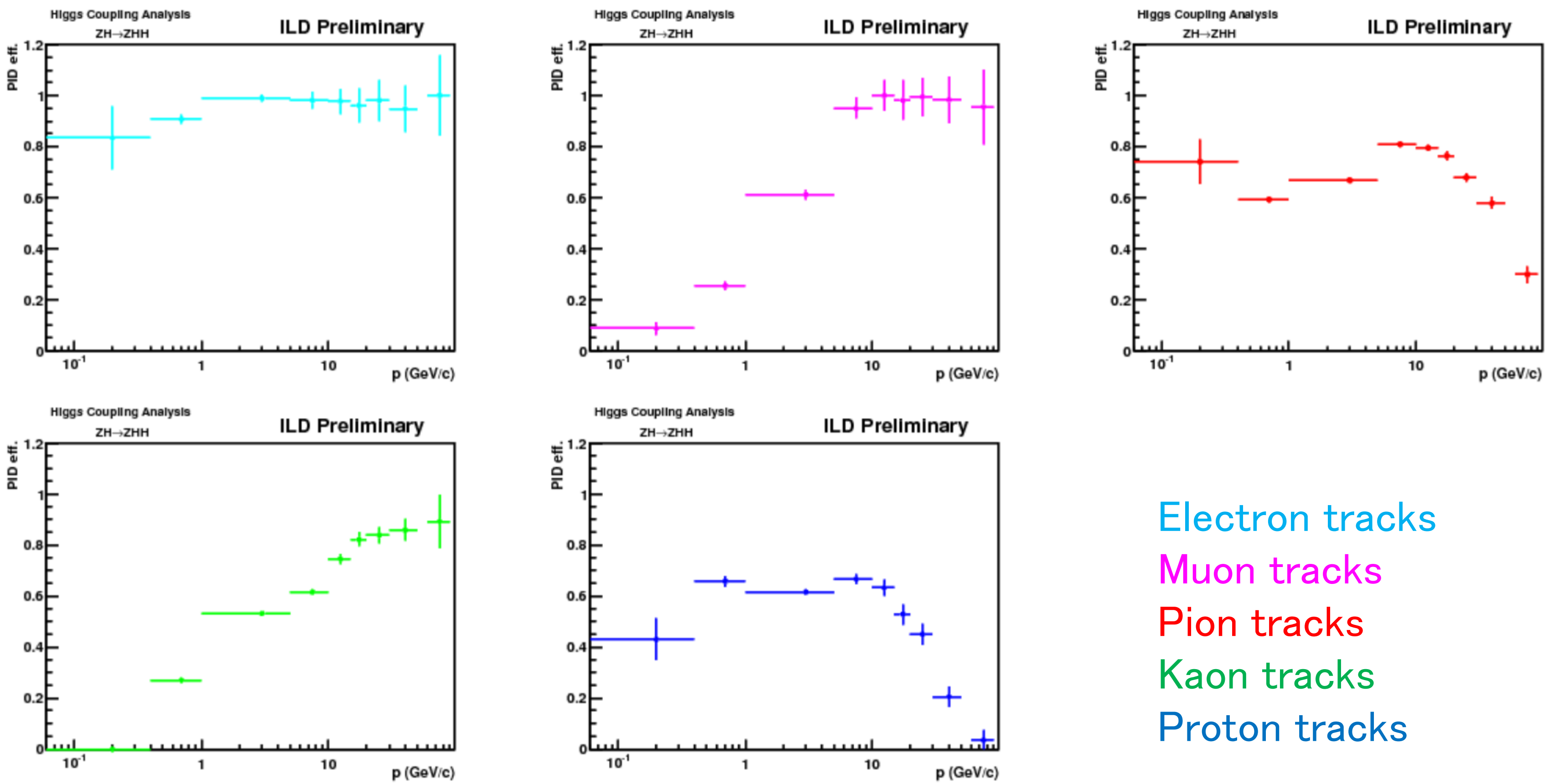
  ex) Absorption length

  Electron
  Muon
  Pion
  Kaon
  Proton



PLOTS

- @500GeV
- Momentum dep. of PID efficiency



Electron tracks
Muon tracks
Pion tracks
Kaon tracks
Proton tracks

Summary & Todo

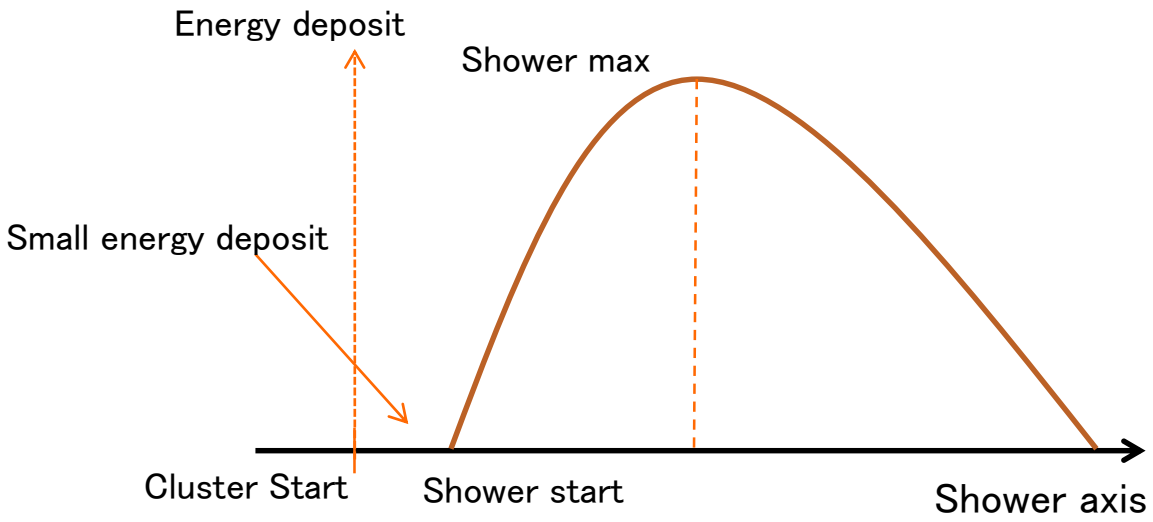
- We have released the first version of PIDTools
  - We can use dE/dx information
  - We can use shower profile information(but still messy)
  - We can use Particle ID using all the track information

- Todo:
  - Forming several types of PID, using just part of track information
    e.g.) use dE/dx only, use shower profile only, etc.
  - Some variables are not yet included, so we need to use them(backups)
    - μ / π separation will become better
  - Many minor updates…

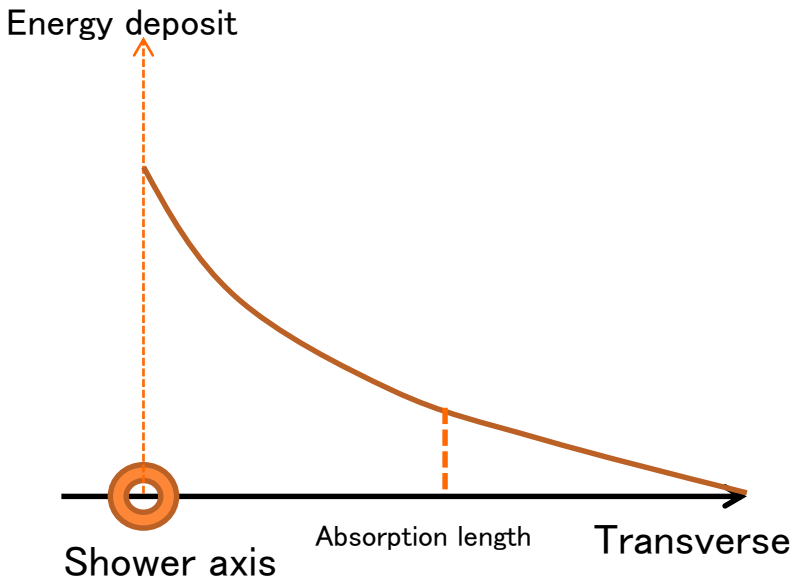
BACKUPS

SHOWER PROFILE –STRUCTURE IN THE CLUSTER

longitudinal

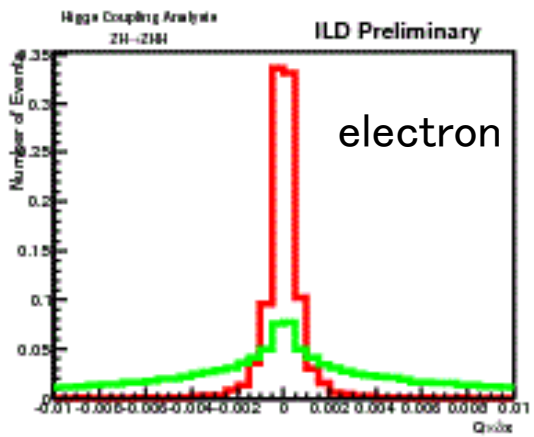


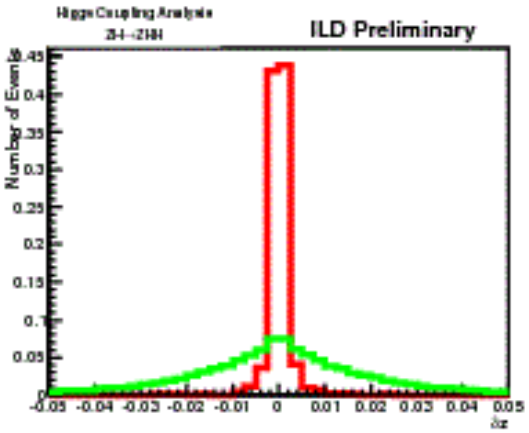
transverse



Introduce variables

- Variables are almost same as cut based:
  - E/P, EM/(EM+HAD), |d0|,|Z0|, cone energy
  - Using these variables as pdf
- Introduce new variables:
  - Q × Δx, Δz – distance between the cluster position and expected position when tracks are extrapolated to the radius of the cluster position(Q is charge)





Isolated lepton

Fakes