

Browsing into data: Jets, Tau, MET, etc

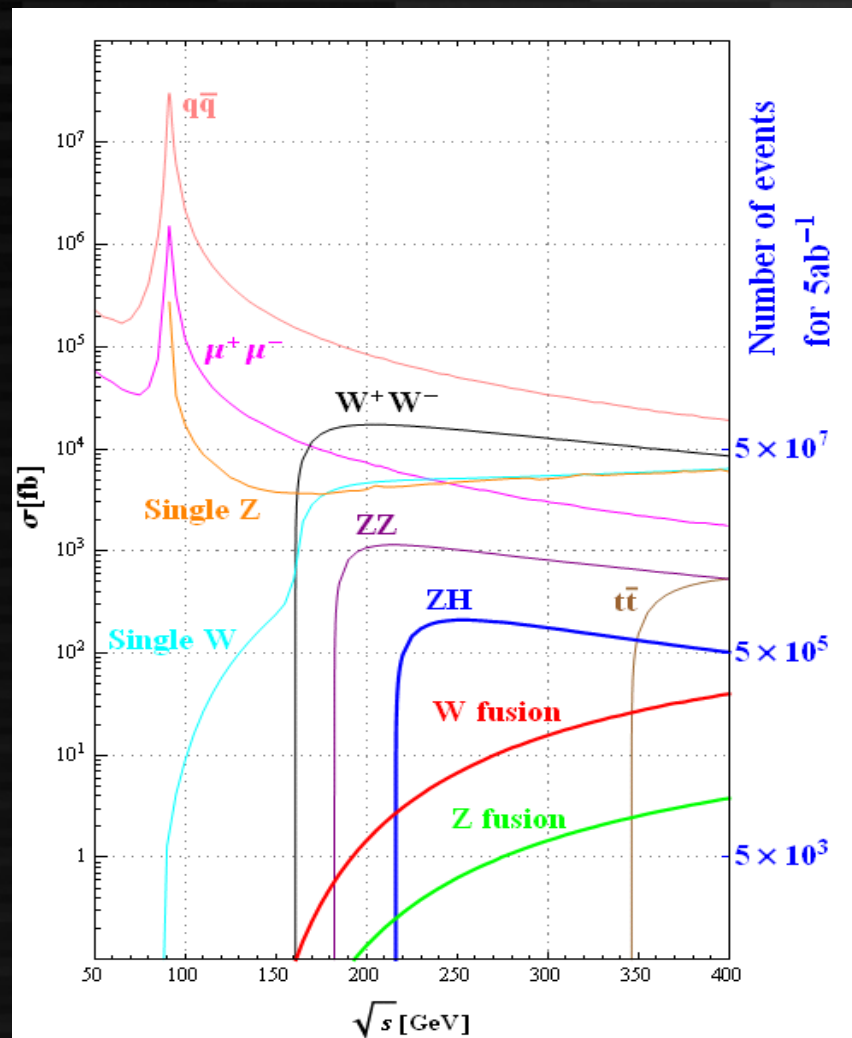
Gang Li

li.gang@mail.ihep.ac.cn

Summary table

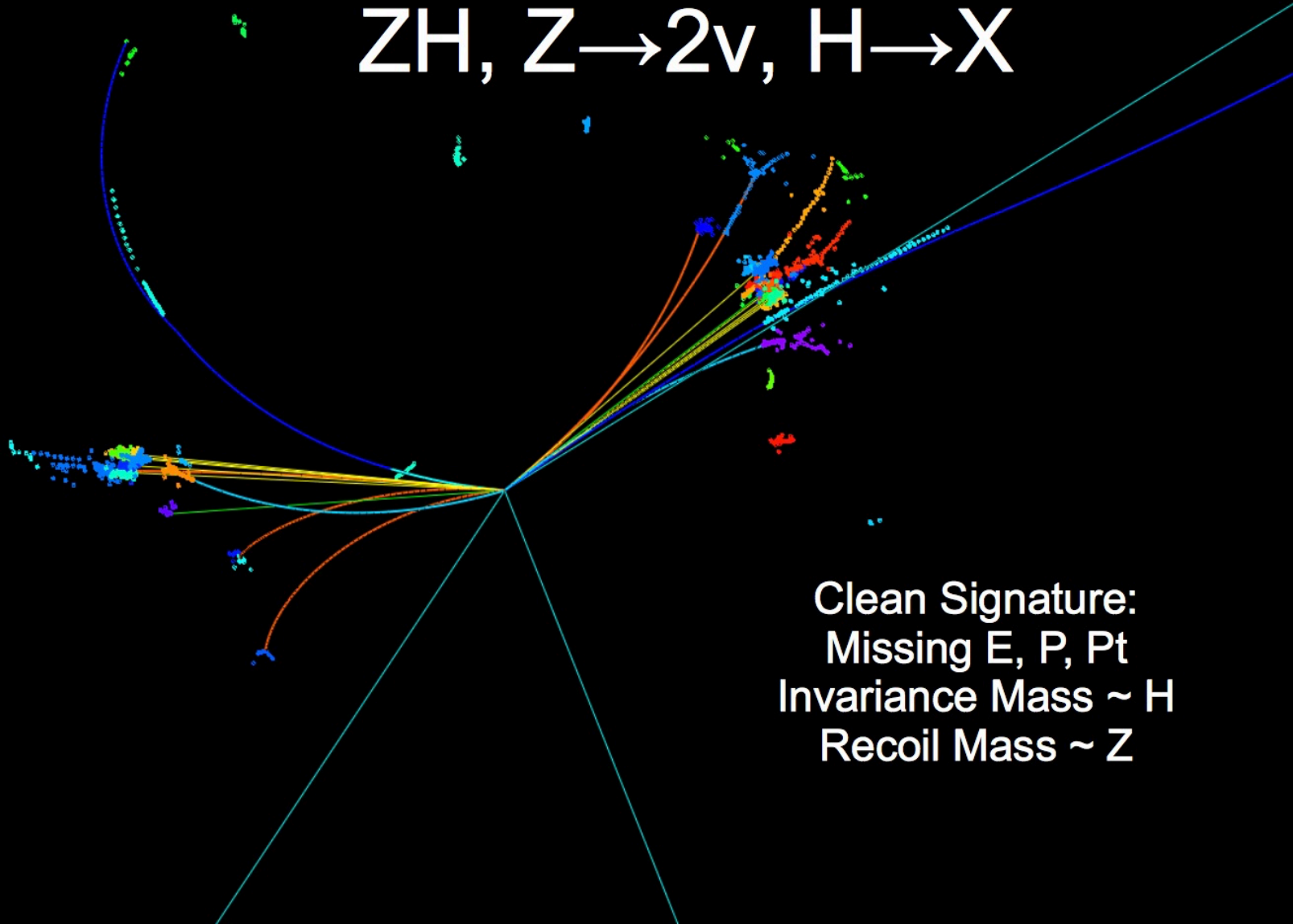
Cross sections [fb]

	240GeV	250GeV
qq	54662	50216
$\mu^+\mu^-$	4685	4405
single Z	4538	4734
single W	5086	5144
W^+W^-	16004	15484
ZZ	1079	1033
ZH	203	212
W fusion	5.36	6.72
Z fusion	0.50	0.63



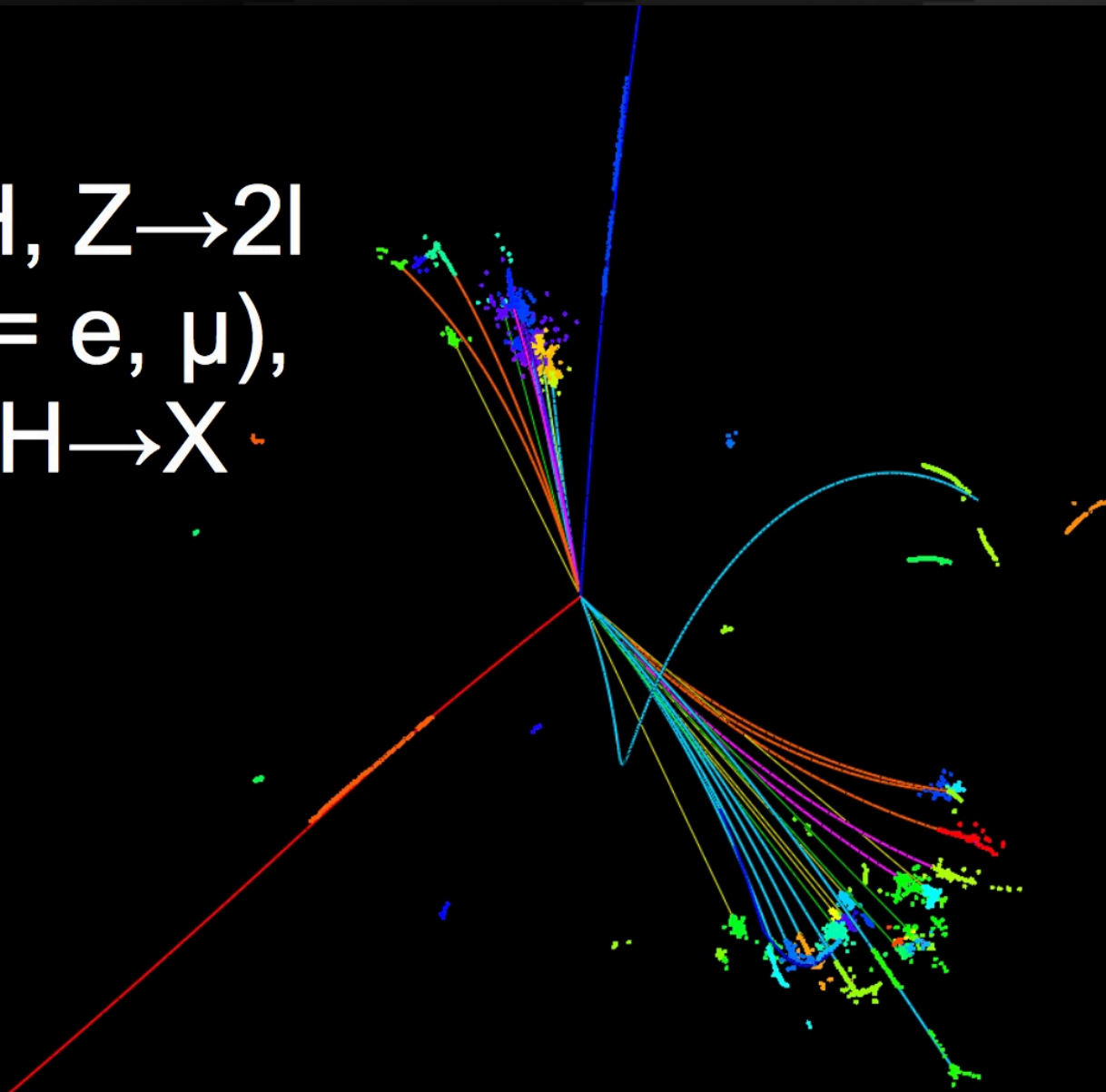
- Two types physics objects used in analysis
 - Single PFO: $e/\mu/\gamma$...
 - Composed objects: quarks and gluon jet, and τ
 - The four-momentum, **flavor**, **electrical charge**
- Now we move on and investigate composed objects

$ZH, Z \rightarrow 2\nu, H \rightarrow X$



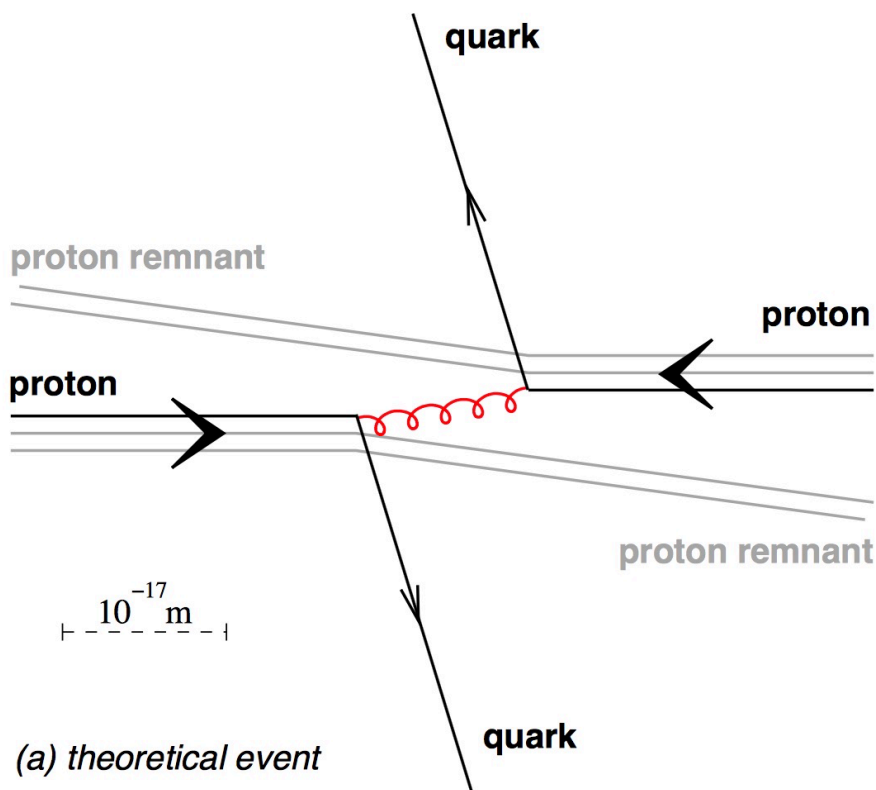
Clean Signature:
Missing E, P, Pt
Invariance Mass $\sim H$
Recoil Mass $\sim Z$

$ZH, Z \rightarrow 2l$
($l = e, \mu$),
 $H \rightarrow X$

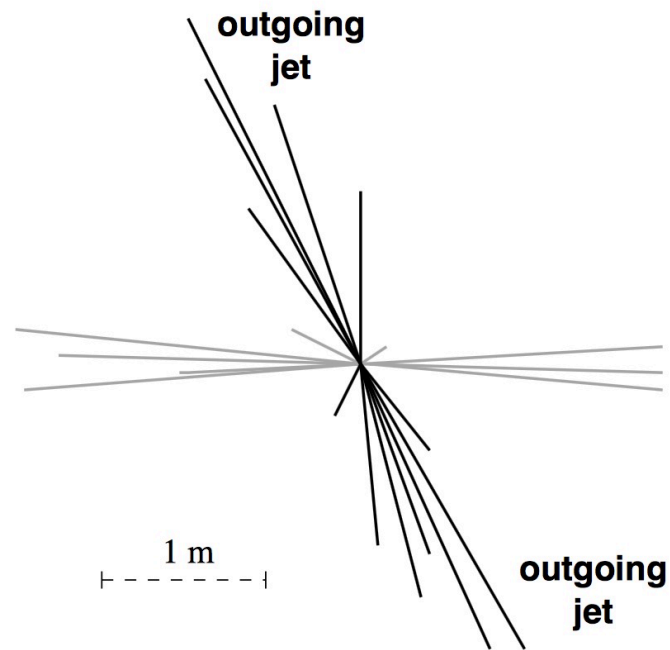


Jets

from partons to real objects



(a) theoretical event



(b) schematic view of real event

- The configuration of Jet-clustering is different from hadron collider
 - Exclusive jets: force the PFOs into fixed number of jets
 - Some special objects should be removed from FPO collection before jet-clustering, such as electrons, muons, ...

On LCFI+

- Standalone package on vertexfinder, Jet-clustering, and flavor identification
- Based on MVA method
- Potentially adopt the latest development in the industrial
- General introduction as user level

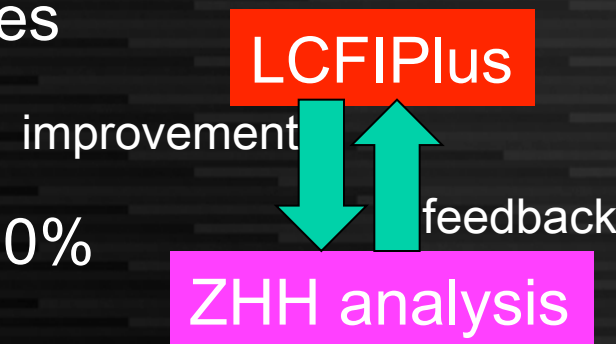
Direction of LCFIPlus development

LCFIVertex The first realistic flavor tagging in ILC

- Incorporating modern flavor tagging techniques to obtain reasonable performance
- No other algorithms to be compared...
- Mainly tuned with Z-pole qqbar samples

LCFIPlus the second version

- Clear target: Higgs self-coupling to $\sim 30\%$
➔ high demand for performance
- Focused on ≥ 4 jet environments
- Including jet clustering (performance driver for 6-jets)
- Trying many ideas for performance improvement



**LCFIPlus is more performance-driven,
mainly concentrated on many-jet processes**

Data/process flow

All in "lcfiplus" namespace

EventStore

singleton for data pool

vector<Track *> vector<Vertex *>
vector<Neutral *> vector<Jet *>
vector<MCParticle *> any other types

- Automatic type identification
(Allow one name with multiple types)
- Automatic creation/deletion
(using ROOT class dictionary)

Algorithm

PrimaryVertex JetVertexRefiner
BuildUpVertex FlavorTag TrainMVA
JetClustering MakeNtuple ReadMVA etc.

- Parameters class used
for type-safe configuration

LCIOStorer

- Automatic conversion from
LCIO to lcfiplus classes
(using hook in EventStore)
- Conversion to LCIO
is manually invoked by
LcfiplusProcessor

LcfiplusProcessor

- Marlin processor
- Process Marlin parameters
to be passed to Algorithm
- LCIO I/O configuration

LCIO

configuration

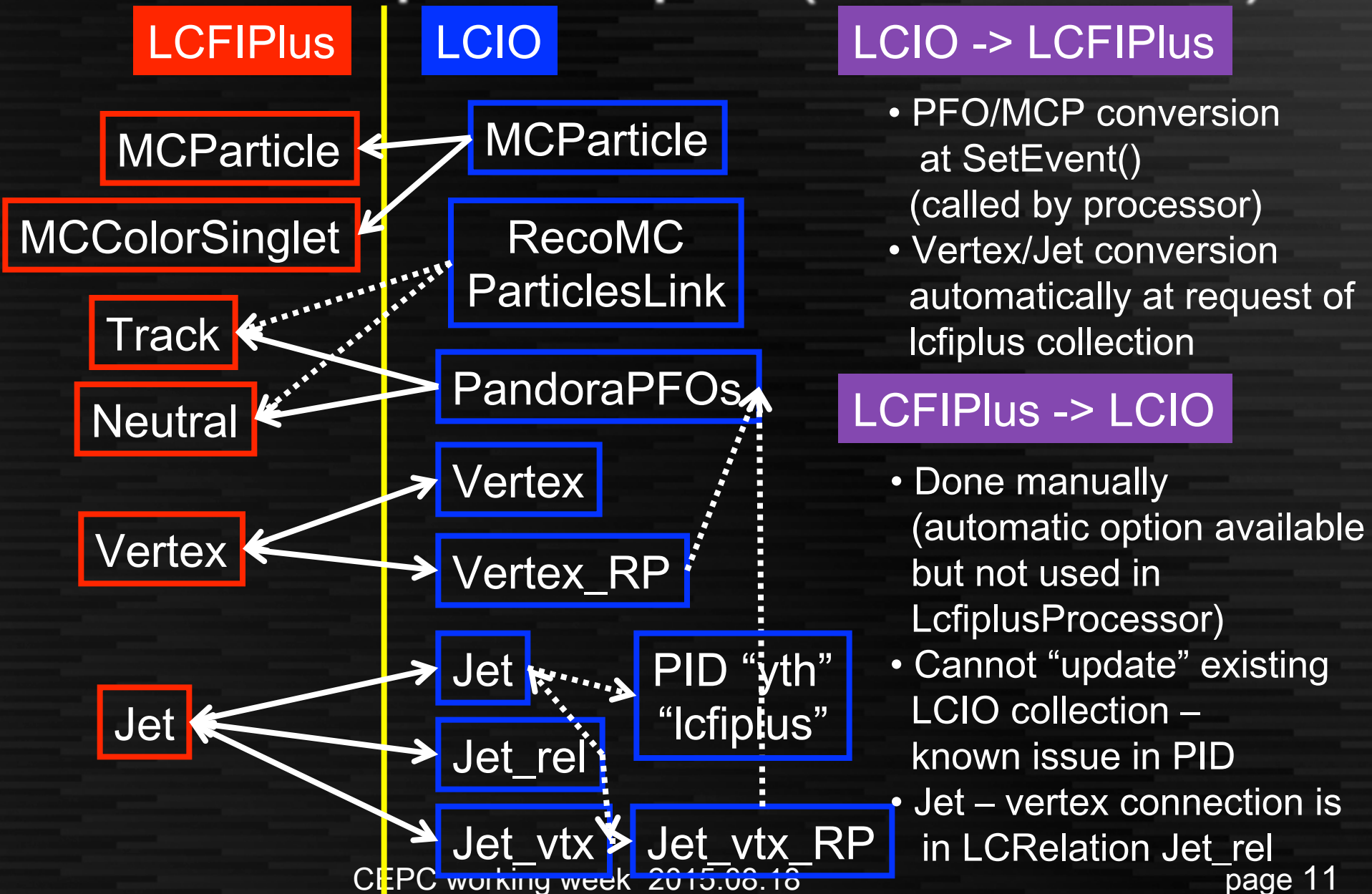
Marlin

Internal algorithms

C working week 2

Independent

LCIO input/output (LCIOStorer)



LcfixplusProcessor and parameters

Multiple LcfixplusProcessor possible now
Note: EventStore is singleton, so collection is converted only at first processor using the collection

List algorithms

General parameters:
see next slide for detail

See steer/README for samples

Output collections converted to LCIO at the end of this proc.

Algorithm specific parameters

Input PFO collection can be different

General parameters are treated by LcfixplusProcessor
Others sent to Algorithms

```
string" value="PandoraPFOs" /> <!-- input PFO collection -->+
  ue="0" /> <!-- MC info not used -->+
  string" value="" /> <!-- not used -->+
  string" value="" /> <!-- not used -->+
  s" type="int" value="1"/> <!-- true for ILD -->+
  s" type="int" value="0"/> <!-- false for non-updativ PandoraPFOs -->+
```

LcfiplusProcessor: global params

To use edep in subdetectors for jet-muon detection: assume ILD so specify 0 for SiD

List algorithms

```
13 <!-- primary and secondary vertex finders -->↓
14 <parameter name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex
15 ↓
16 <parameter name="UseMCP" type="int" value="0" />
17 <parameter name="MCPCollection" type="string" value="MCP" />
18 <parameter name="MCPFORelation" type="string" value="MCP" /> <!-- not used -->↓
19 <parameter name="ReadSubdetectorEnergies" type="int" value="1" /> <!-- true for ILD -->↓
20 <parameter name="UpdateVertexRPDaughters" type="int" value="0" /> <!-- false for r -->↓
21 <parameter name="PrintEventNumber" type="int" value="0" /> <!-- 0 for not printing -->↓
22 ↓
23 <!-- specify input collection names -->↓
24 <parameter name="PFOCollection" type="string" value="PandoraPFOs" />↓
25 ↓
```

UseMCP = 0: do not convert MCParticle

These two are global: do specify the same for all LcfiplusProcessor

Modify "StartVertex" in PFO collection or not
1 for mass production, 0 for user analysis

Print current event number every n events
Good for debug

Input PFO collection: can be different in each LcfiplusProcessor

Algorithm (1) vertex finders

- VertexFitterSimple
 - Original vertex fitter with Minuit2 in ROOT
 - Do not refit tracks
 - Slow – may need to improve
- PrimaryVertexFinder
 - TearDown method with VertexFitterSimple
 - Use beam vertex (fixed parameters now)
- BuildUpVertex
 - Secondary vertex finder with VertexFitterSimple
 - Tuned for JetClustering (many cuts included)
 - **Recent improvement in V0 rejection**

Vertex finder steering file

steer/vertex.xml

```
<processor name="VertexFinder" type="LcflplusProcessor">↓
  ↓
  <!-- run primary and secondary vertex finders -->↓
  <parameter name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex </parameter>↓
  <parameter name="ReadSubdetectorEnergies" type="int" value="1"/> <!-- true for ILD -->↓
  <parameter name="UpdateVertexRPDaughters" type="int" value="0"/> <!-- false for non-updative Pandora -->↓
  <parameter name="PrintEventNumber" type="int" value="0"/> <!-- 0 for not printing event number, n for n -->↓
  ↓
  <!-- specify input collection names -->↓
  <parameter name="PFOCollection" type="string" value="PandoraPFOs" />↓
  <parameter name="PrimaryVertexCollectionName" type="string" value="PrimaryVertex" />↓
  <parameter name="BuildUpVertexCollectionName" type="string" value="BuildUpVertex" />↓
  <parameter name="BuildUpVertex.V0VertexCollectionName" type="string" value="BuildUpVertex_V0" />↓
  ↓
  <!-- parameters for primary vertex finder -->↓
  <parameter name="PrimaryVertexFinder.TrackMaxD0" type="double" value="20." />↓
  <parameter name="PrimaryVertexFinder.TrackMaxZ0" type="double" value="20." />↓
  <parameter name="PrimaryVertexFinder.TrackMaxInnermostHitRadius" type="double" value="10." />↓
  <parameter name="PrimaryVertexFinder.TrackMinVtxFtdHits" type="int" value="5" />↓
  <parameter name="PrimaryVertexFinder.Chi2Threshold" type="double" value="25." />↓
  ↓
  <!-- parameters for secondary vertex finder -->↓
  <parameter name="BuildUpVertex.TrackMaxD0" type="double" value="10." />↓
  <parameter name="BuildUpVertex.TrackMaxZ0" type="double" value="20." />↓
  <parameter name="BuildUpVertex.TrackMinPt" type="double" value="0.1" />↓
  <parameter name="BuildUpVertex.TrackMaxD0Err" type="double" value="0.1" />↓
  <parameter name="BuildUpVertex.TrackMaxZ0Err" type="double" value="0.1" />↓
  <parameter name="BuildUpVertex.TrackMinTpcHits" type="int" value="20" />↓
  <parameter name="BuildUpVertex.TrackMinFtdHits" type="int" value="3" />↓
  <parameter name="BuildUpVertex.TrackMinVxdHits" type="int" value="3" />↓
  <parameter name="BuildUpVertex.TrackMinVxdFtdHits" type="int" value="0" />↓
  <parameter name="BuildUpVertex.PrimaryChi2Threshold" type="double" value="25." />↓
  <parameter name="BuildUpVertex.SecondaryChi2Threshold" type="double" value="9." />↓
  <parameter name="BuildUpVertex.MassThreshold" type="double" value="10." />↓
  <parameter name="BuildUpVertex.MinDistFromIP" type="double" value="0.3" />↓
  <parameter name="BuildUpVertex.MaxChi2ForDistOrder" type="double" value="1.0" />↓
  <parameter name="BuildUpVertex.AssocIPTracks" type="int" value="1" />↓
  <parameter name="BuildUpVertex.AssocIPTracksMinDist" type="double" value="0." />↓
  <parameter name="BuildUpVertex.AssocIPTracksChi2RatioSecToPri" type="double" value="2.0" />↓
  <parameter name="BuildUpVertex.UseV0Selection" type="int" value="1" />↓
  ↓
</processor>↓
```

V0 collection is used in later algorithms

Parameters are highly tuned: please contact us if you need to modify them

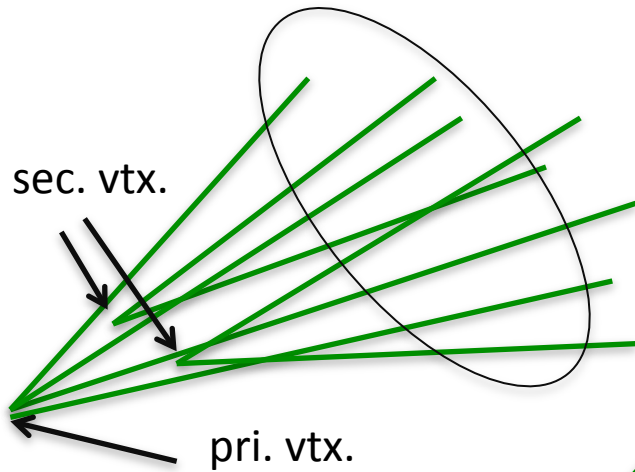
Algorithm(2) JetClustering

- Jet clustering with vertex information (see next slide)
- Using jet muons as vertex with UseMuonID = 1
 - Using edep in muon detector: only for ILD...
- Plane Durham is possible by specifying “0” for InputVertexCollectionName (do not specify “”) & UseMuonID=0
- Multiple output collections can be done such as NJetsRequested = 8 6 4, (must be descending order), OutputJetCollectionName = Jets8 Jets6 Jets4
- Add ParticleID yth with y23, y34,..., y910 parameters for ycuts
 - Issue: yn(n+1) is obtained only if NJetRequested <= n is done

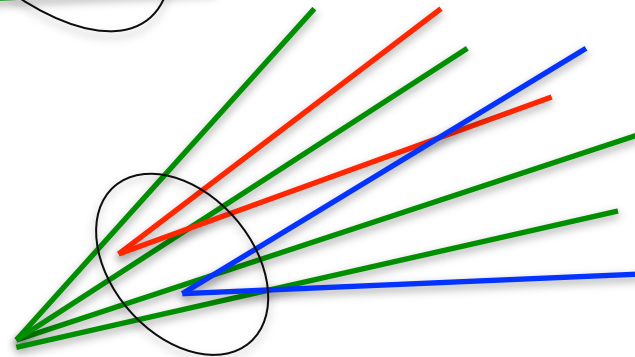
```
<!-- jet clustering parameters -->↓
<parameter name="JetClustering.InputVertexCollectionName" type="string" value="BuildUpVertex" />
<parameter name="JetClustering.OutputJetCollectionName" type="stringVec" value="VertexJets" /> <!--
<parameter name="JetClustering.NJetsRequested" type="intVec" value="2" /> <!-- Multiple NJets can
↓
<parameter name="JetClustering.YCut" type="doubleVec" value="0." /> <!-- specify 0 if not used --
<parameter name="JetClustering.UseMuonID" type="int" value="1" /> <!-- jet-muon ID for jet cluste
<parameter name="JetClustering.VertexSelectionMinimumDistance" type="double" value="0.3" /> <!--
<parameter name="JetClustering.VertexSelectionMaximumDistance" type="double" value="30." /> <!--
<parameter name="JetClustering.VertexSelectionKOMassWidth" type="double" value="0.02" /> <!-- in
↓
```


Vertex-Jet Finding Overview

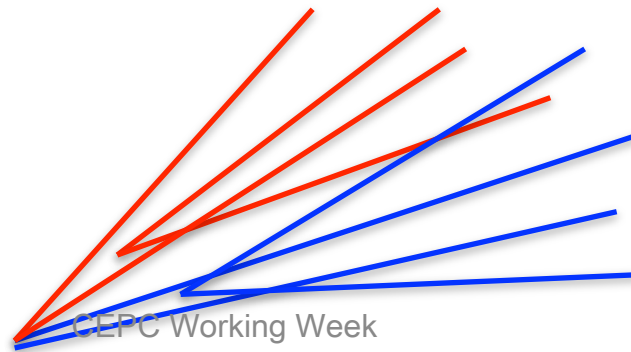
arXiv:1110.5785



1. Difficult to separate two b-jets which are close. Ordinary kt algorithm tends to merge them.



2. To overcome this, find secondary vertices first, and use them as *seeds* for jet finding.



3. This results in an increased chance of correct jet separation.

This effect is particularly pronounced for final states with many b jets, e.g. Zhh

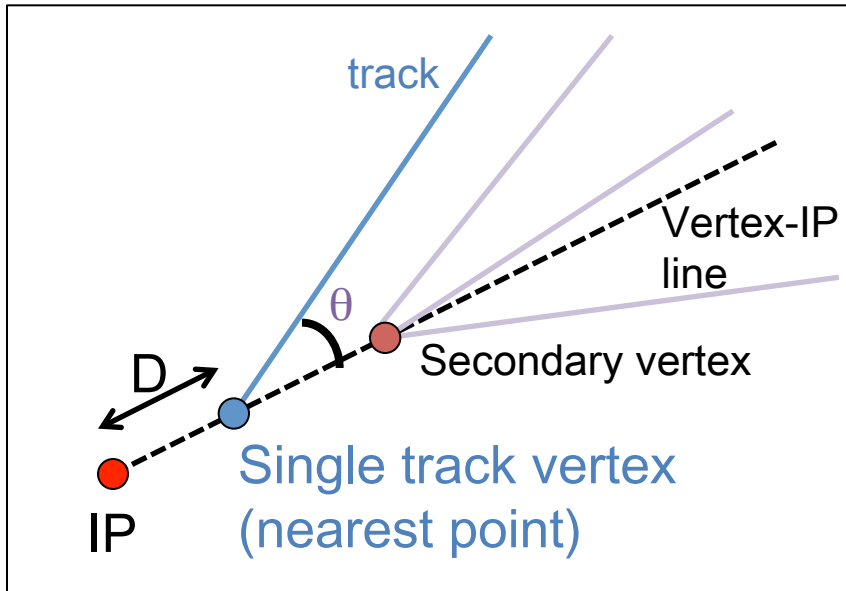
Algorithm(3) JetVertexRefiner

- Consists of two algorithms
 - SingleTrackVertexFinder & VertexCombiner
- SingleTrackVertexFinder: reconstruct single-track vertices using existing vertex directions (see next slide)
- VertexCombiner: combine vertices into two at most aiming at combining multi+single vertices which are from same b or c – tuned for b/c separation
- Jet & vertex collection are specified separately, so this can be used after other jet clustering method (Durham etc.)

```
<!-- vertex refiner parameters -->↓
<parameter name="JetVertexRefiner.InputJetCollectionName" type="string" value="VertexJets" />↓
<parameter name="JetVertexRefiner.OutputJetCollectionName" type="string" value="RefinedJets" />↓
<parameter name="JetVertexRefiner.PrimaryVertexCollectionName" type="string" value="PrimaryVertex" />↓
<parameter name="JetVertexRefiner.InputVertexCollectionName" type="string" value="BuildUpVertex" />↓
<parameter name="JetVertexRefiner.VOVertexCollectionName" type="string" value="BuildUpVertex_VO" />↓
<parameter name="JetVertexRefiner.OutputVertexCollectionName" type="string" value="RefinedVertex" />↓
↓
<parameter name="JetVertexRefiner.MinPosSingle" type="double" value="0.3" />↓
<parameter name="JetVertexRefiner.MaxPosSingle" type="double" value="30." />↓
<parameter name="JetVertexRefiner.MinEnergySingle" type="double" value="1." />↓
<parameter name="JetVertexRefiner.MaxAngleSingle" type="double" value="0.5" />↓
<parameter name="JetVertexRefiner.MaxSeparationPerPosSingle" type="double" value="0.1" />↓
<parameter name="JetVertexRefiner.minD0sigSingle" type="double" value="5." />↓
<parameter name="JetVertexRefiner.minzD0sigSingle" type="double" value="5." />↓
<parameter name="JetVertexRefiner.OneVertexProbThreshold" type="double" value="0.001" />↓
<parameter name="JetVertexRefiner.MaxCharmFlightLengthPerJetEnergy" type="double" value="0.1" />↓
↓
```

Parameters are highly tuned

Single Track Selection



- Normal vertex finder needs > 2 tracks
-> loose many vertices
- Single track vertex can be found by using other vertex direction
- Improves b-tagging performance

Algorithm(4) flavor tagging

- Based on TMVA Boosted Decision Trees
 - MLP neural net under development
 - Four categories: #vtx = 0, 1, 1+singletrack, 2
- Algorithms
 - FlavorTag: obtain input variables
 - MakeNtuple: making ROOT ntuple for training
 - TrainMVA: training MVA with b/c/s ntuples
 - ReadMVA: obtain BTag/CTag variables with weight file
- Procedure
 1. FlavorTag + MakeNtuple for each training sample
 2. TrainMVA with all ntuples (output: weight file)
 3. FlavorTag + ReadMVA with the weight file

Algorithm(4) input variables

```
<parameter name="FlavorTag.CategoryDefinition1" type="string">nvtx==0</parameter>↓
<parameter name="FlavorTag.CategoryVariables1" type="stringVec">↓
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz↓
</parameter>↓
<parameter name="FlavorTag.CategorySpectators1" type="stringVec">↓
  aux nvtx↓
</parameter>↓
↓
<parameter name="FlavorTag.CategoryDefinition2" type="string">nvtx==1&&nvtxall==1</parameter>↓
<parameter name="FlavorTag.CategoryVariables2" type="stringVec">↓
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz↓
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob↓
</parameter>↓
<parameter name="FlavorTag.CategorySpectators2" type="stringVec">↓
  aux nvtx↓
</parameter>↓
↓
<parameter name="FlavorTag.CategoryDefinition3" type="string">nvtx==1&&nvtxall==2</parameter>↓
<parameter name="FlavorTag.CategoryVariables3" type="stringVec">↓
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz↓
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob↓
  1vtxprob 2ndfit vtxmassall↓
</parameter>↓
<parameter name="FlavorTag.CategorySpectators3" type="stringVec">↓
  aux nvtx↓
</parameter>↓
↓
<parameter name="FlavorTag.CategoryDefinition4" type="string">nvtx>=2</parameter>↓
<parameter name="FlavorTag.CategoryVariables4" type="stringVec">↓
  trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr jprobz↓
  vtxlen1_jete vtxsig1_jete vtxdirang1_jete vtxmom1_jete vtxmass1 vtxmult1 vtxmasspc vtxprob↓
  vtxlen2_jete vtxsig2_jete vtxdirang2_jete vtxmom2_jete vtxmass2 vtxmult2↓
  vtxlen12_jete vtxsig12_jete vtxdirang12_jete vtxmom_jete vtxmass vtxmult↓
</parameter>↓
<parameter name="FlavorTag.CategorySpectators4" type="stringVec">↓
  aux nvtx↓
</parameter>↓
```

LcfiplusProcessor and parameters

```
1 <marlin>↓
2 ↓
3 <execute>↓
4   <processor name="VertexFinder"/>↓
5   <processor name="JetClusteringAndFlavorTag"/>↓
6   <processor name="MyLCIOOutputProcessor"/>↓
7 </execute>↓
8 ↓
9 <global> <!-- nip --> </global>↓
10 ↓
11 <processor name="VertexFinder" type="LcfiplusProcessor">↓
12 ↓
13 <!-- run primary and secondary vertex finders -->↓
14 <parameter name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex
15 ↓
16 <parameter name="UseMCP" type="int" value="0" /> <!-- MC info not used -->↓
17 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
18 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
19 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
20 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
21 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
22 ↓
23 <!-- specify input collection names -->↓
24 <parameter name="PFOCollection" type="string" value="PandoraPFOs" />↓
25 ↓
26 <!-- algorithm specific parameters -->↓
27 <parameter name="PrimaryVertexCollectionName" type="string" value="PrimaryVertex" />↓
28 <parameter name="BuildUpVertexCollectionName" type="string" value="BuildUpVertex" />↓
29 <parameter name="BuildUpVertex.VOVertexCollectionName" type="string" value="BuildUpVertex_V
30 <!-- nip -->↓
31 ↓
32 <processor name="JetClusteringAndFlavorTag" type="LcfiplusProcessor">↓
33 ↓
34 <!-- run primary and secondary vertex finders -->↓
35 <parameter name="Algorithms" type="stringVec"> JetClustering JetVertexRefiner FlavorTag ReadMVA</parameter>↓
36 ↓
37 <!-- general parameters -->↓
38 ↓
39 <parameter name="PFOCollection" type="string" value="PandoraPFOs" /> <!-- input PFO collection -->↓
40 <parameter name="UseMCP" type="int" value="0" /> <!-- MC info not used -->↓
41 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
42 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
43 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
44 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
45 <parameter name="UseMCP" type="int" value="0" /> <!-- not used -->↓
```

Multiple LcfiplusProcessor possible now
Note: EventStore is singleton, so collection is converted only at first processor using the collection

List algorithms

General parameters:
see next slide for detail

See steer/README for samples

Output collections converted to LCIO at the end of this proc.

Algorithm specific parameters

Input PFO collection can be different

General parameters are treated by LcfiplusProcessor
Others sent to Algorithms

Known issue: JetVertexRefiner & ReadMVA cannot be separated since we need to modify jet collection

Example

General information

- LCFI+ v00-05-02 used in vo1-17-05
- Z pole samples $Z \rightarrow b\bar{b}$, $c\bar{c}$, and light quark pairs of 1M each
- Input PFOs: Arbor

Step by step: samples

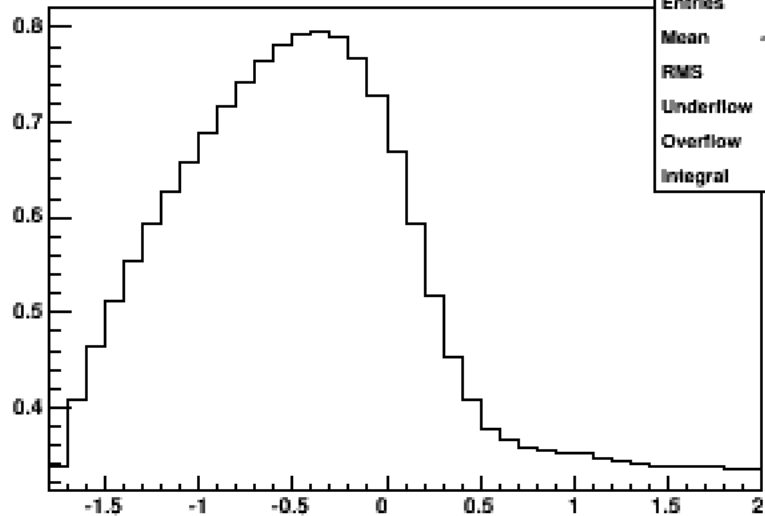
- Generator samples: $Z \rightarrow QQ\bar{q}$:
 - /besfs/groups/higgs/data/Fast_Simulation/wo_beamstrahlung/background/Z-pole
 - Sqrt(s)=91.18GeV
 - with ISR
- Simulation and reconstruction (CEPC_v1)
 - /besfs/groups/higgs/data/SimReco/wo_BS/FlavorTag/CEPC_v1_zqq
 - SimData
 - RecData_ArborDHCAL_5_ILD

TrackNtuple: prior-prob

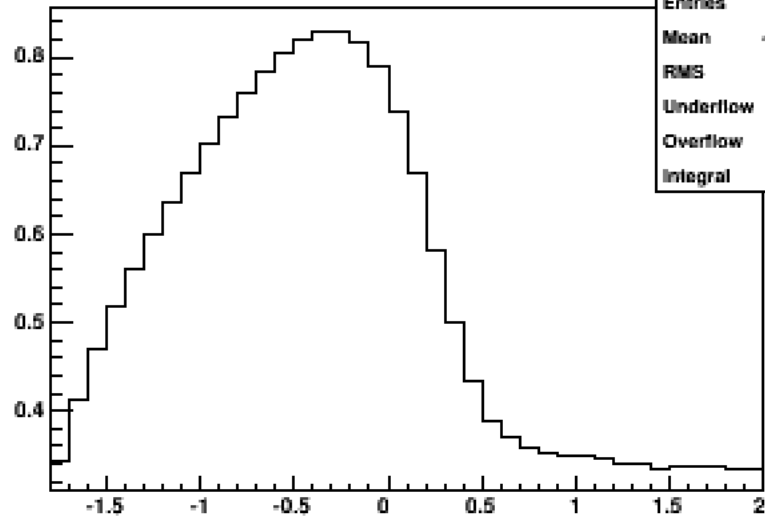
- Based on MC samples, All tracks in jets are used

```
for(unsigned int n=0;n<tracks.size();n++){  
    const Track *tr = tracks[n];  
  
    float sd0 = signedD0(tr, j, privtx, true);  
    float sd0sig = signedD0Significance(tr, j, privtx, true);  
    float sz0 = signedZ0(tr, j, privtx, true);  
    float sz0sig = signedZ0Significance(tr, j, privtx, true);  
  
    bool hitCut = trackSelectionForFlavorTag(tr, _hitcutJprob);  
  
    _tree->Fill(sd0,sd0sig,sz0,sz0sig,hitCut);  
}  
}
```

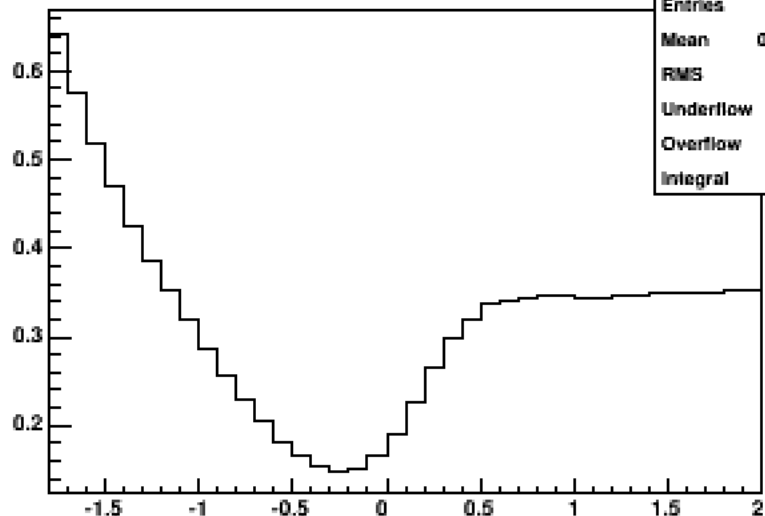
- - /home/bes/lig/higgs/higgs/analysis/LCFIplus/FT_Arbor_cepc_v1/
TrackNtuple

hb

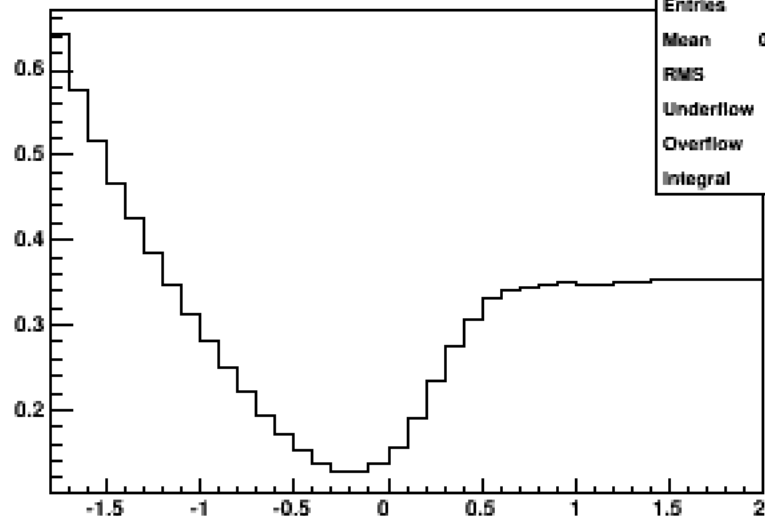
hb	
Entries	20
Mean	-0.1189
RMS	0.9885
Underflow	0.2942
Overflow	0.3383
Integral	19.58

hbp

hbp	
Entries	20
Mean	-0.1223
RMS	0.9759
Underflow	0.3024
Overflow	0.338
Integral	20.22

hc

hc	
Entries	12
Mean	0.03605
RMS	1.218
Underflow	0.6852
Overflow	0.3498
Integral	12.13

hcp

hcp	
Entries	12
Mean	0.04432
RMS	1.232
Underflow	0.6832
Overflow	0.3514
Integral	11.87

Step by step: Fill Ntuple

- Based on PFOs, the variables for MVA calculated and some prior distribution generated
- Three processors used:
- VertexFinder(if not called),

```
<processor name="VertexFinder" type="LcfiplusProcessor">  
  <!-- run primary and secondary vertex finders -->  
  <parameter name="Algorithms" type="stringVec"> PrimaryVertexFinder BuildUpVertex </parameter>
```

- JetClustering

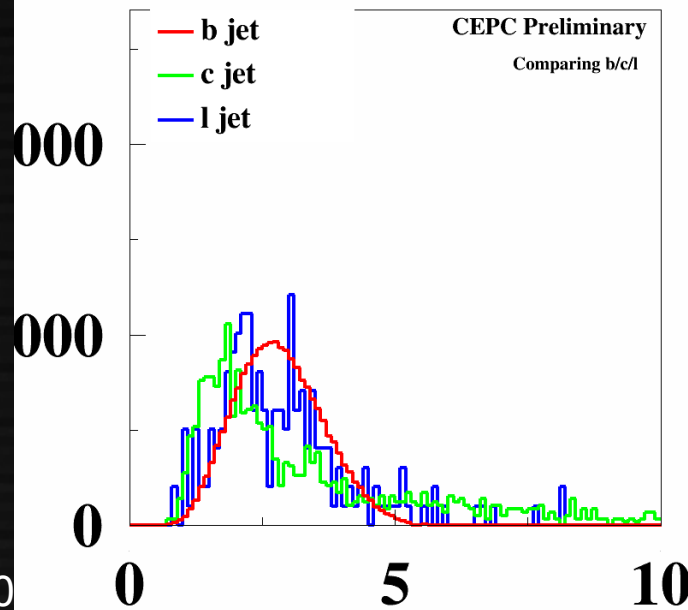
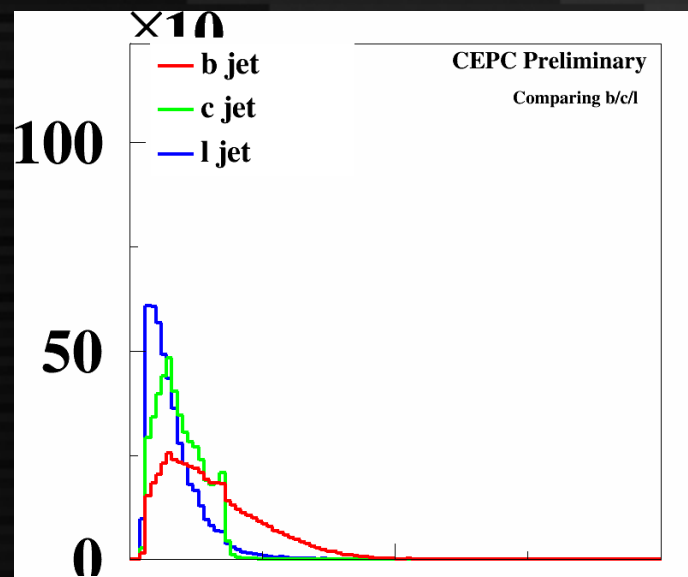
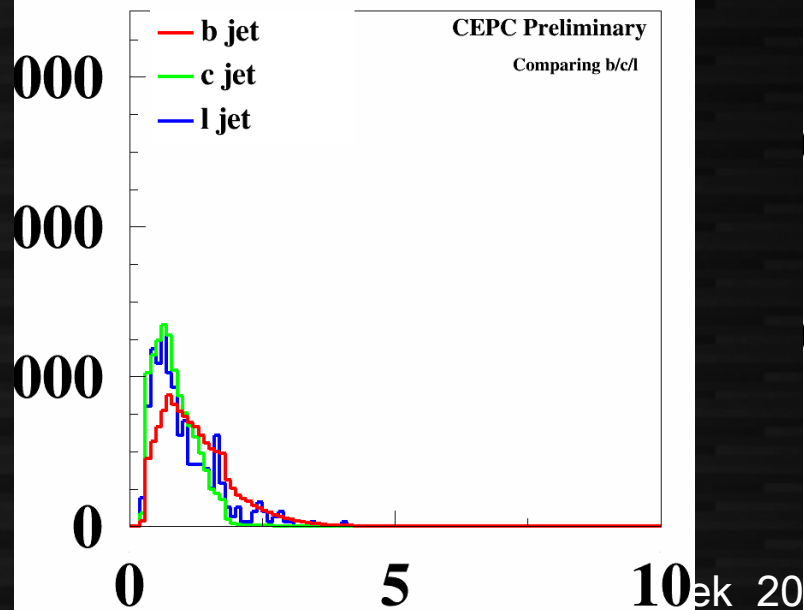
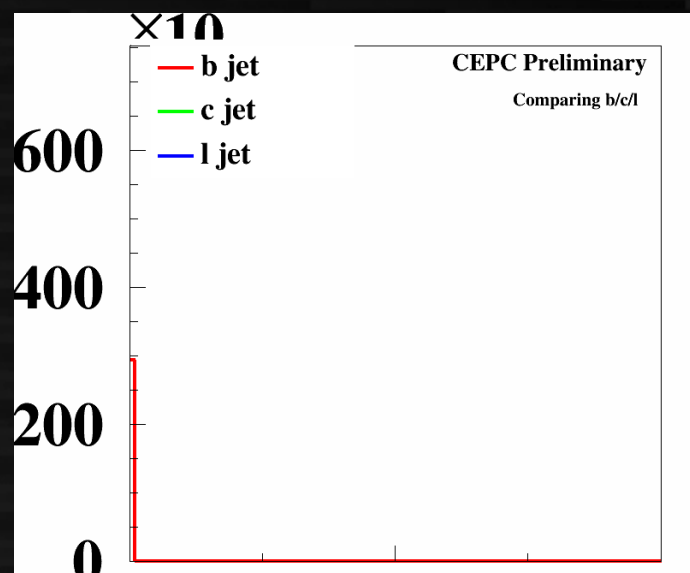
```
<processor name="JetClustering" type="LcfiplusProcessor">  
  
  <!-- run primary and secondary vertex finders -->  
  <parameter name="Algorithms" type="stringVec"> JetClustering JetVertexRefiner</parameter>
```

- MakeNtuple

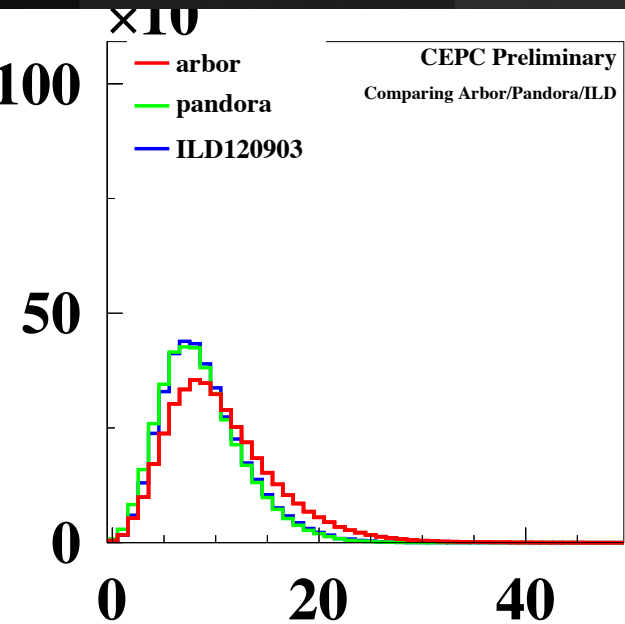
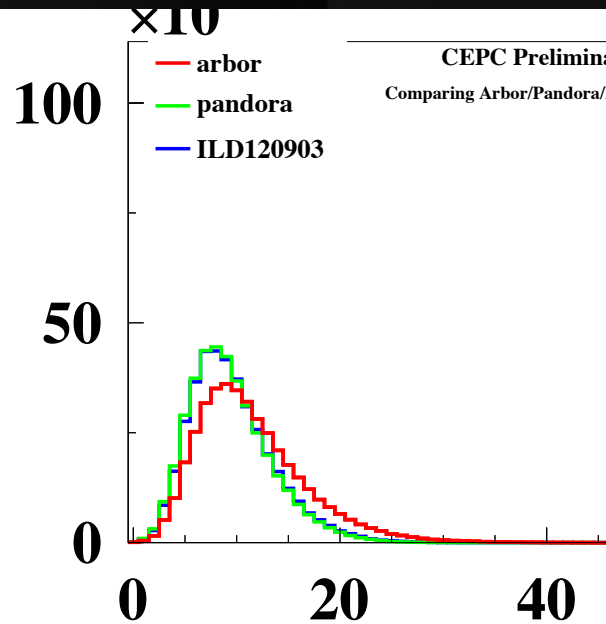
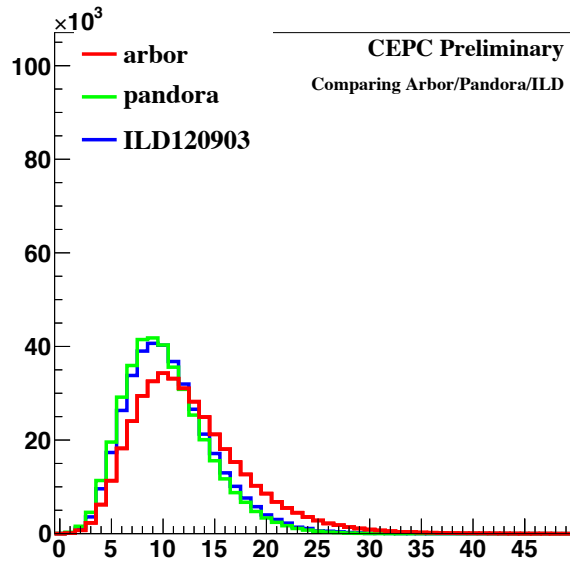
```
<processor name="MakeNtuple" type="LcfiplusProcessor">  
  <parameter name="Algorithms" type="stringVec">FlavorTag MakeNtuple</parameter>
```

- Variable choice first time here

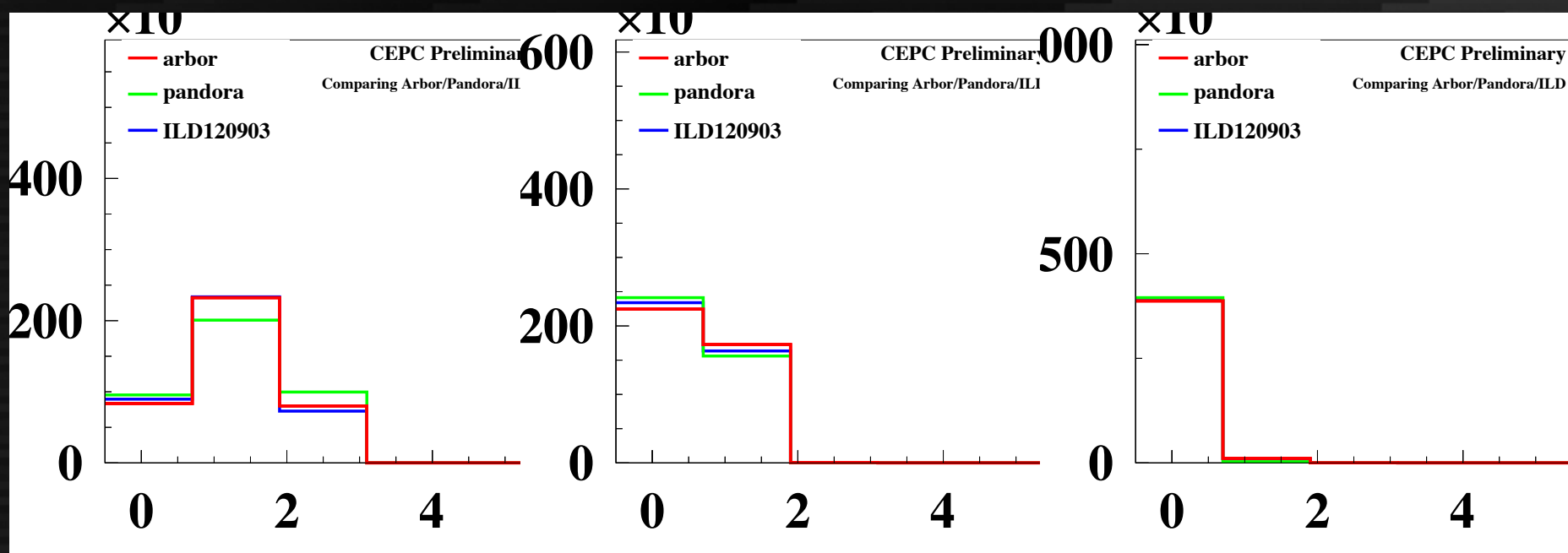
vtxmass



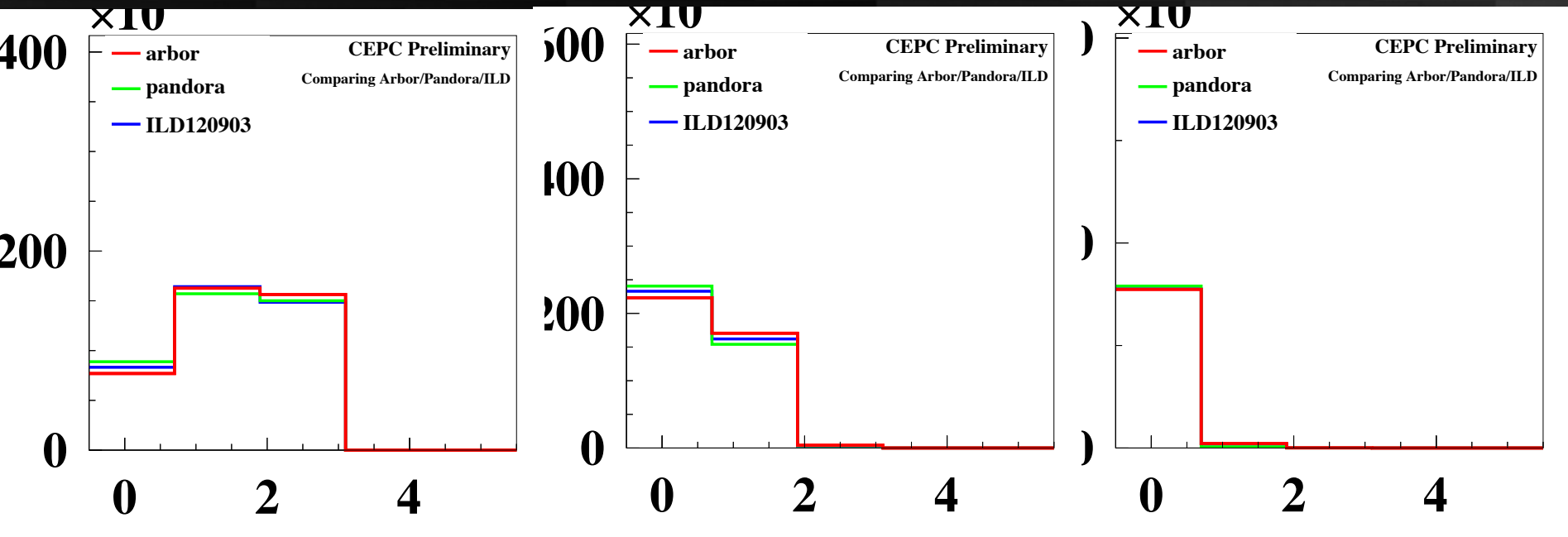
Ntrks in jets



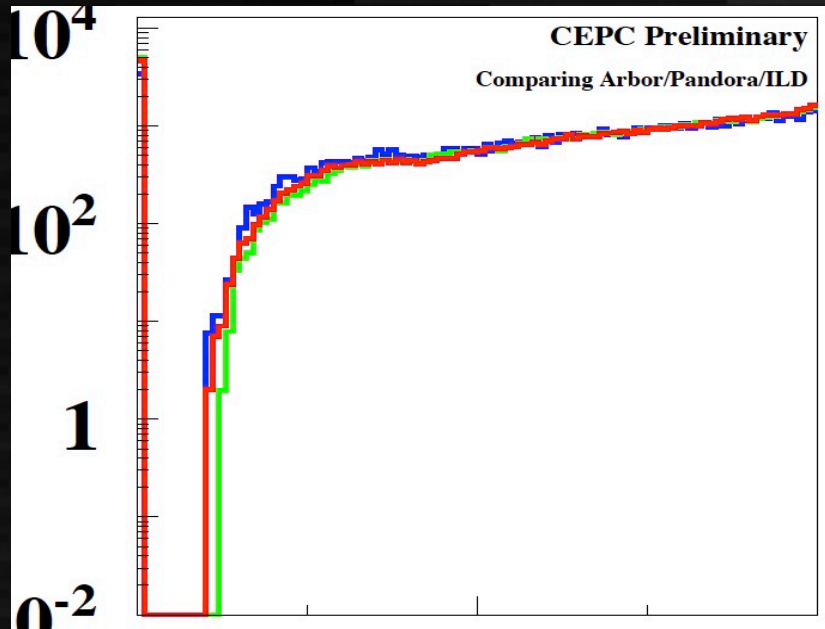
Nvtx



Nvtxall

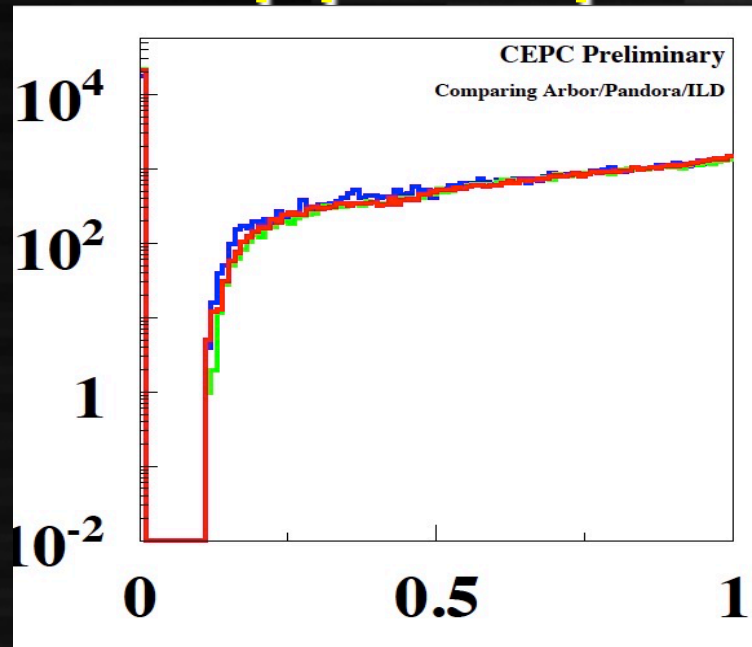


qq:trk1pt



```
class FtTrk1Pt : public FTAlgo {  
public:  
    FtTrk1Pt() : FTAlgo("trk1pt") {}  
    void process() {  
        double sigVec[6];  
        findMostSignificantTrack(_jet, _privtx, _nhitsMostSignificantTrack, sigVec);  
        _result = sigVec[4];  
    }  
};
```

qq:trk2pt



```
class FtTrk2Pt : public FTAlgo {  
    public:  
        FtTrk2Pt() : FTAlgo("trk2pt") {}  
        void process() {  
            double sigVec[6];  
            findMostSignificantTrack(_jet,_privtx,_nhitsMostSignificantTrack,sigVec);  
            _result = sigVec[5];  
        }  
};
```

Step by step: training

- Train the MVA algorithm based on (subset) of the variables in the ntuples

```
<processor name="MyLcfiplusProcessor" type="LcfiplusProcessor">
  <parameter name="Algorithms" type="stringVec">TrainMVA</parameter>
  <parameter name="PFOCollection" type="string" value="ArborPFOs" />
  <parameter name="FlavorTag.WeightsDirectory" type="string" value="lcfiweights" />
  <parameter name="FlavorTag.WeightsPrefix" type="string" value="zpole_v00" />
  <parameter name="FlavorTag.BookName" type="string" value="bdt" />

  <parameter name="FlavorTag.CategoryDefinition1" type="string">nvtx==0</parameter>
  <parameter name="FlavorTag.CategoryPreselection1" type="string">trk1d0sig!=0</parameter>
  <parameter name="FlavorTag.CategoryVariables1" type="stringVec">
    trk1d0sig trk2d0sig trk1z0sig trk2z0sig trk1pt_jete trk2pt_jete jprobr5sigma jprobz5sigma
    d0bprob d0cprob d0qprob z0bprob z0cprob z0qprob nmuon nelectron trkmass
  </parameter>
```

Information in log file

```
: Ranking input variables (method specific)...  
: Ranking result (top variable is best ranked)
```

```
: -----  
: Rank : Variable      : Variable Importance
```

```
: -----  
: 1 : trkmass      : 9.397e-02  
: 2 : d0qprob      : 8.894e-02  
: 3 : trk2d0sig    : 7.816e-02  
: 4 : z0qprob      : 7.780e-02  
: 5 : trk2z0sig    : 7.771e-02  
: 6 : trk1d0sig    : 7.428e-02  
: 7 : jprobr5sigma : 7.009e-02  
: 8 : jprobz5sigma : 6.519e-02  
: 9 : trk1z0sig    : 6.366e-02  
: 10 : nelectron    : 5.663e-02  
: 11 : d0cprob      : 5.324e-02  
: 12 : trk1pt_jete  : 5.200e-02  
: 13 : trk2pt_jete  : 4.957e-02  
: 14 : z0cprob      : 3.987e-02  
: 15 : z0bprob      : 3.140e-02  
: 16 : d0bprob      : 2.749e-02  
: 17 : nmuon        : 0.000e+00  
: -----
```

```
: Ranking result (top variable is best ranked)
```

```
: -----  
: Rank : Variable      : Variable Importance
```

```
: -----  
: 1 : vtxmasspc    : 1.097e-01  
: 2 : vtxmom1_jete : 6.706e-02  
: 3 : d0cprob      : 5.563e-02  
: 4 : jprobz       : 5.447e-02  
: 5 : vtxdirang1_jete : 5.411e-02  
: 6 : vtxmass1     : 5.097e-02  
: 7 : nelectron    : 4.765e-02  
: 8 : trk2d0sig    : 4.670e-02  
: 9 : trkmass      : 4.409e-02  
: 10 : z0qprob      : 4.311e-02  
: 11 : vtxprob      : 3.891e-02  
: 12 : z0cprob      : 3.808e-02  
: 13 : jprobr       : 3.798e-02  
: 14 : vtxmult1     : 3.765e-02  
: 15 : vtxsig1_jete : 3.604e-02  
: 16 : d0qprob      : 3.434e-02  
: 17 : trk1pt_jete  : 3.023e-02  
: 18 : trk2pt_jete  : 2.916e-02  
: 19 : trk2z0sig    : 2.736e-02  
: 20 : nmuon        : 2.729e-02  
: 21 : trk1z0sig    : 2.268e-02  
: 22 : trk1d0sig    : 2.131e-02  
: 23 : z0bprob      : 1.722e-02  
: 24 : vtxlen1_jete : 1.421e-02  
: 25 : d0bprob      : 1.408e-02  
: -----
```

```

-rw-r--r-- 1 lig physics 5.6M Apr 1 21:31 zpole_v00_c0_bdt.weights.xml
-rw-r--r-- 1 lig physics 523K Apr 1 21:31 zpole_v00_c0_bdt.class.C
-rw-r--r-- 1 lig physics 243M Apr 1 23:02 zpole_v00_c0.root
-rw-r--r-- 1 lig physics 5.6M Apr 1 23:56 zpole_v00_c1_bdt.weights.xml
-rw-r--r-- 1 lig physics 525K Apr 1 23:56 zpole_v00_c1_bdt.class.C
-rw-r--r-- 1 lig physics 204M Apr 2 00:42 zpole_v00_c1.root
-rw-r--r-- 1 lig physics 5.6M Apr 2 00:56 zpole_v00_c2_bdt.weights.xml
-rw-r--r-- 1 lig physics 527K Apr 2 00:56 zpole_v00_c2_bdt.class.C
-rw-r--r-- 1 lig physics 68M Apr 2 01:03 zpole_v00_c2.root
-rw-r--r-- 1 lig physics 5.4M Apr 2 01:18 zpole_v00_c3_bdt.weights.xml
-rw-r--r-- 1 lig physics 515K Apr 2 01:18 zpole_v00_c3_bdt.class.C
-rw-r--r-- 1 lig physics 52M Apr 2 01:26 zpole_v00_c3.root

```

```
root [1] .ls
```

```

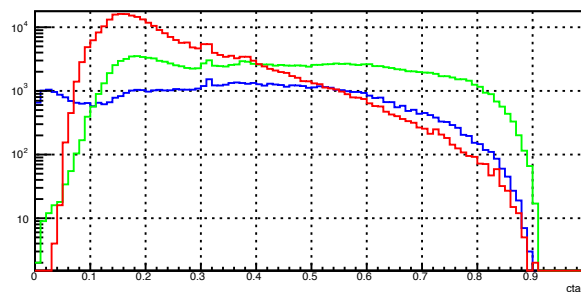
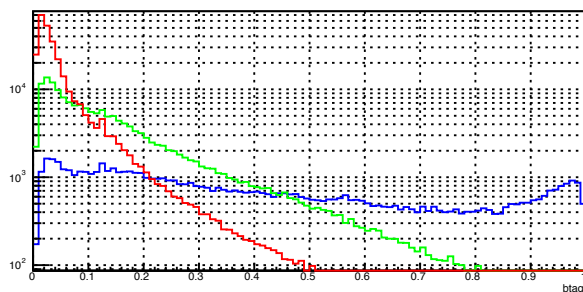
TFile**      zpole_v00_c0.root
TFile*       zpole_v00_c0.root
KEY: TTree   ntp;3    flavor tagging data
KEY: TH2F    CorrelationMatrixjetB;1 Correlation Matrix (jetB)
KEY: TH2F    CorrelationMatrixjetC;1 Correlation Matrix (jetC)
KEY: TH2F    CorrelationMatrixjet0;1 Correlation Matrix (jet0)
KEY: TDirectoryFile  InputVariables_Id;1      InputVariables_Id
KEY: TDirectoryFile  InputVariables_Deco;1    InputVariables_Deco
KEY: TDirectoryFile  InputVariables_PCA;1     InputVariables_PCA
KEY: TDirectoryFile  InputVariables_Gauss_Deco;1      InputVariables_Gauss_Deco
KEY: TDirectoryFile  Method_BDT;1      Directory for all BDT methods
KEY: TTree   TestTree;1      TestTree
KEY: TTree   TrainTree;1     TrainTree

```

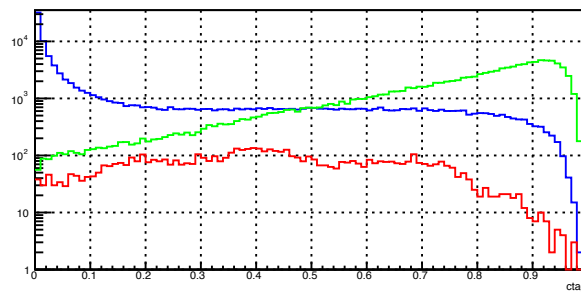
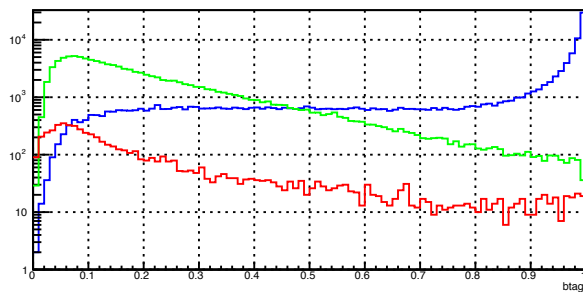
```
root [2]
```

Step by step: Check the performance and apply in analysis

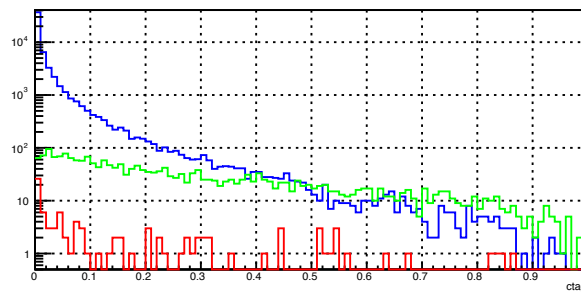
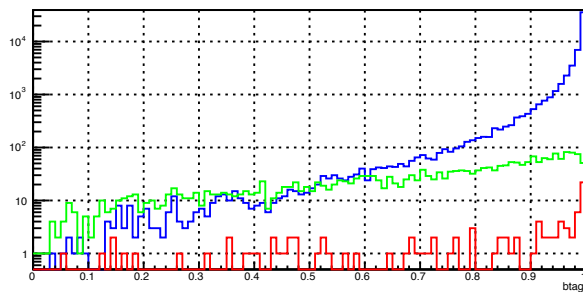
$N_{vtx}=0$



$N_{vtx}=1$

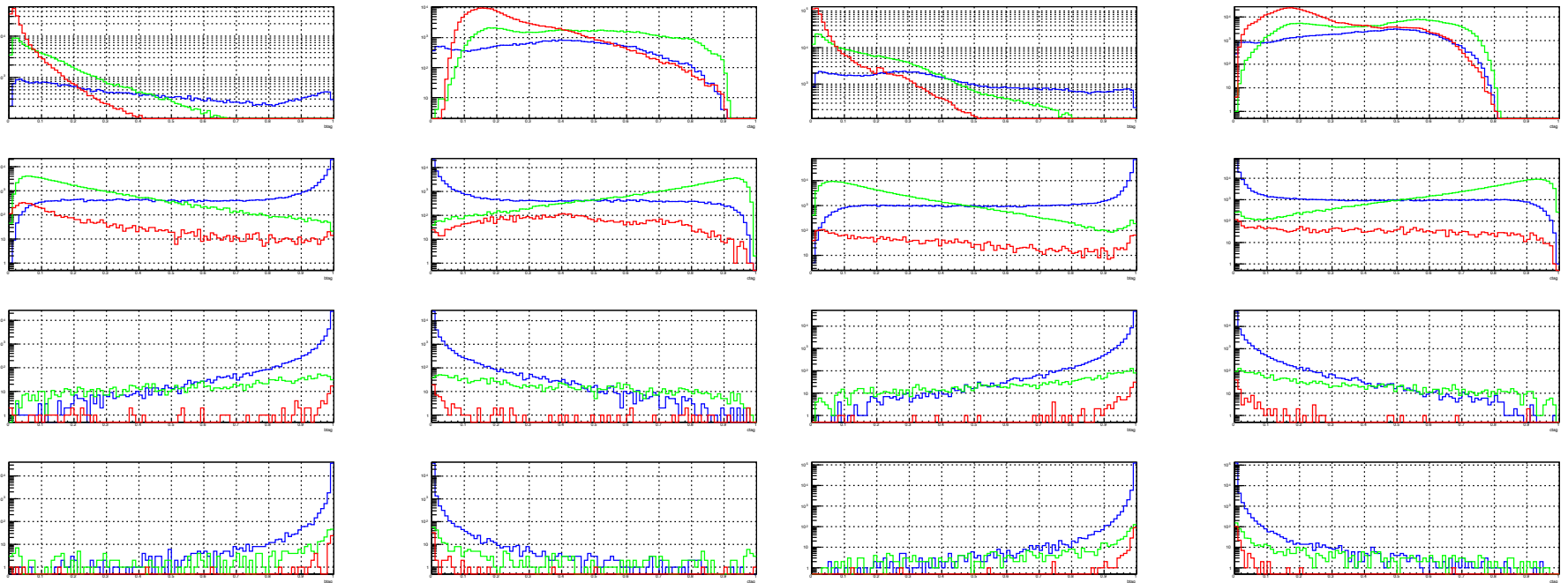


$N_{vtx}>1$

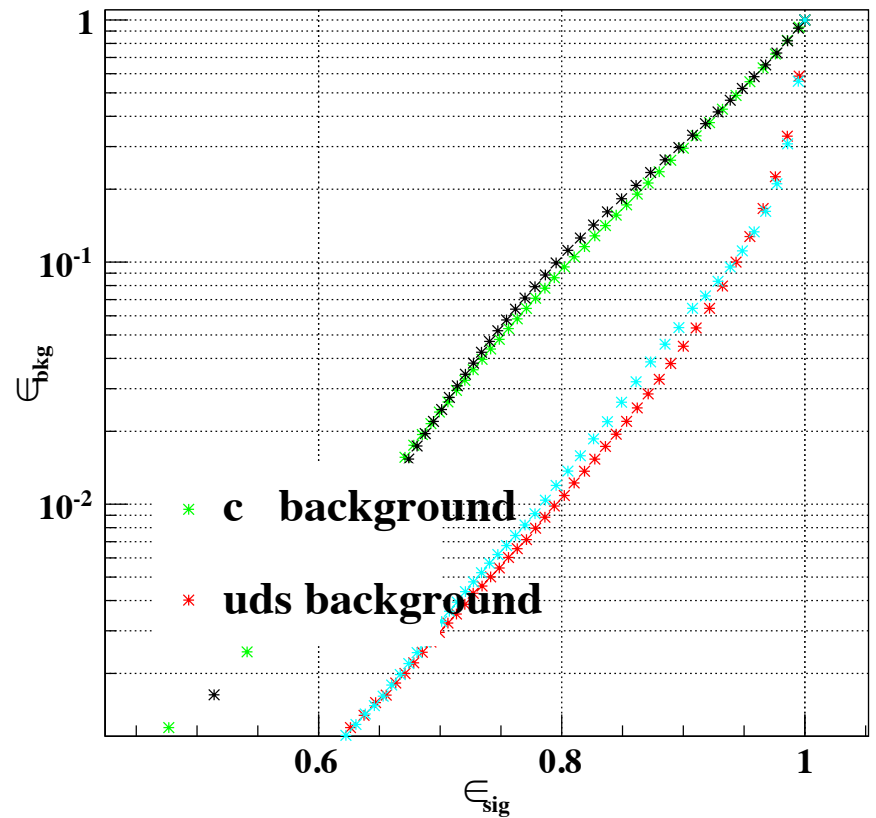
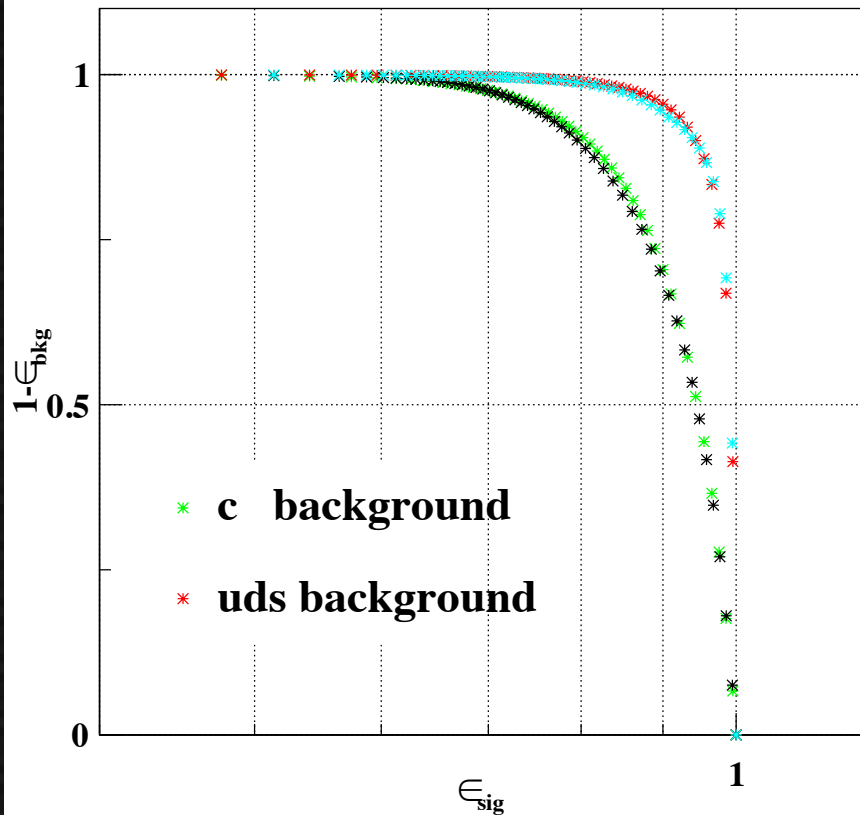


Outputs

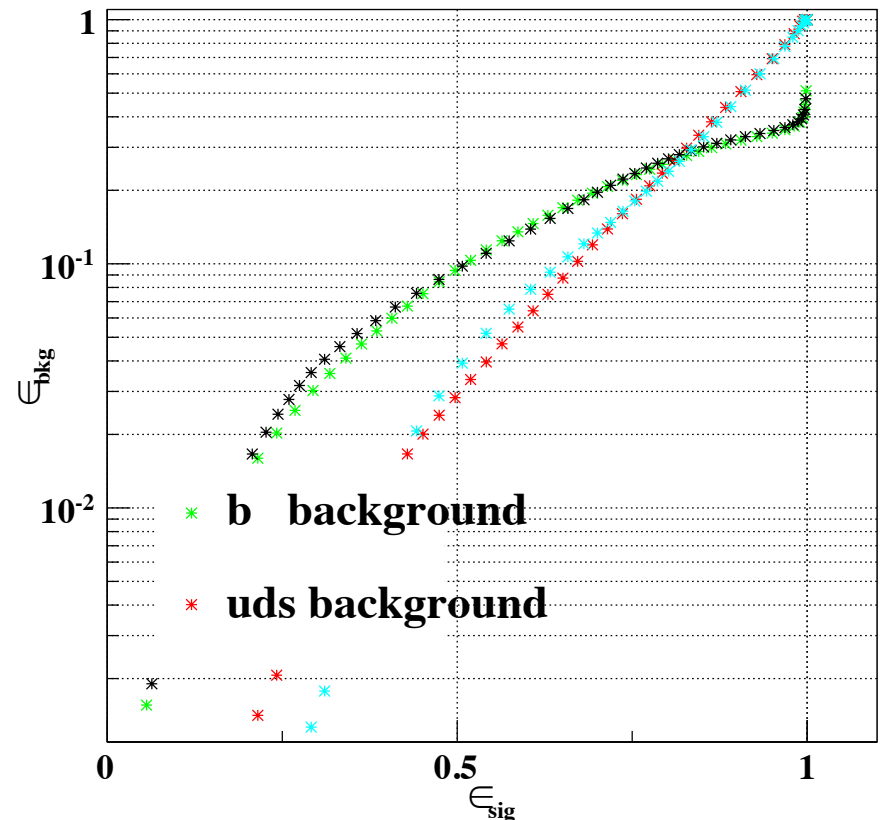
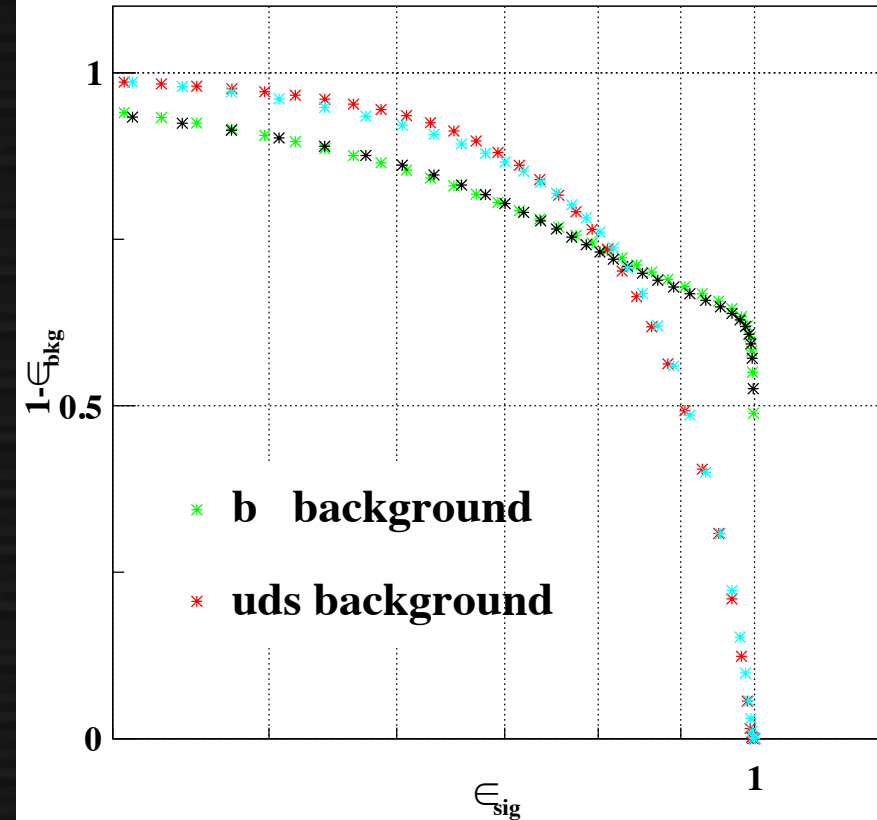
Arbor vs. ILD DBD



Btag



Ctag



Excercise