A general analysis frame work based on slcio and stdhep

LI Gang
li.gang@mail.ihep.ac.cn
IHEPLLI LI
CEPC physics analysis working week 2015.08.20

Outline

- Motivation
- Class design/Objects
- Steering
- Not implemented yet
- Advantages and disadvantages
- Summary

Motivation

- **C++ is an object-oriented computer language**
- **The objects and procedures of physics analysis can be abstracted as some well designed classes and realized with c++**
- **Well-designing has lots of advantages:**
  - ◆ **simple**
  - ◆ **reusable**
  - ◆ **easy to steer**
  - ◆ **easy to debug**
  - ◆ **… …**
  - ◆ **You do not need coding and debugging any more …**

Overview of data-analysis

- Two stages:
  - Pre-selection and Ntuple production
  - Root script – plots and numerical results

- First state
  - Particle Objects
    - MC particles – used for MC topology comparison
    - Reconstructed particles (tracks, clusters, jets) $\rightarrow$ event
  - Combination of objects $\rightarrow$ events
  - Fill ntuples for the next stage in root …

Overview of data-analysis (cont'd)

Data ⟶ Ntuple ⟶ Plots/results

Root

Marlin

Root
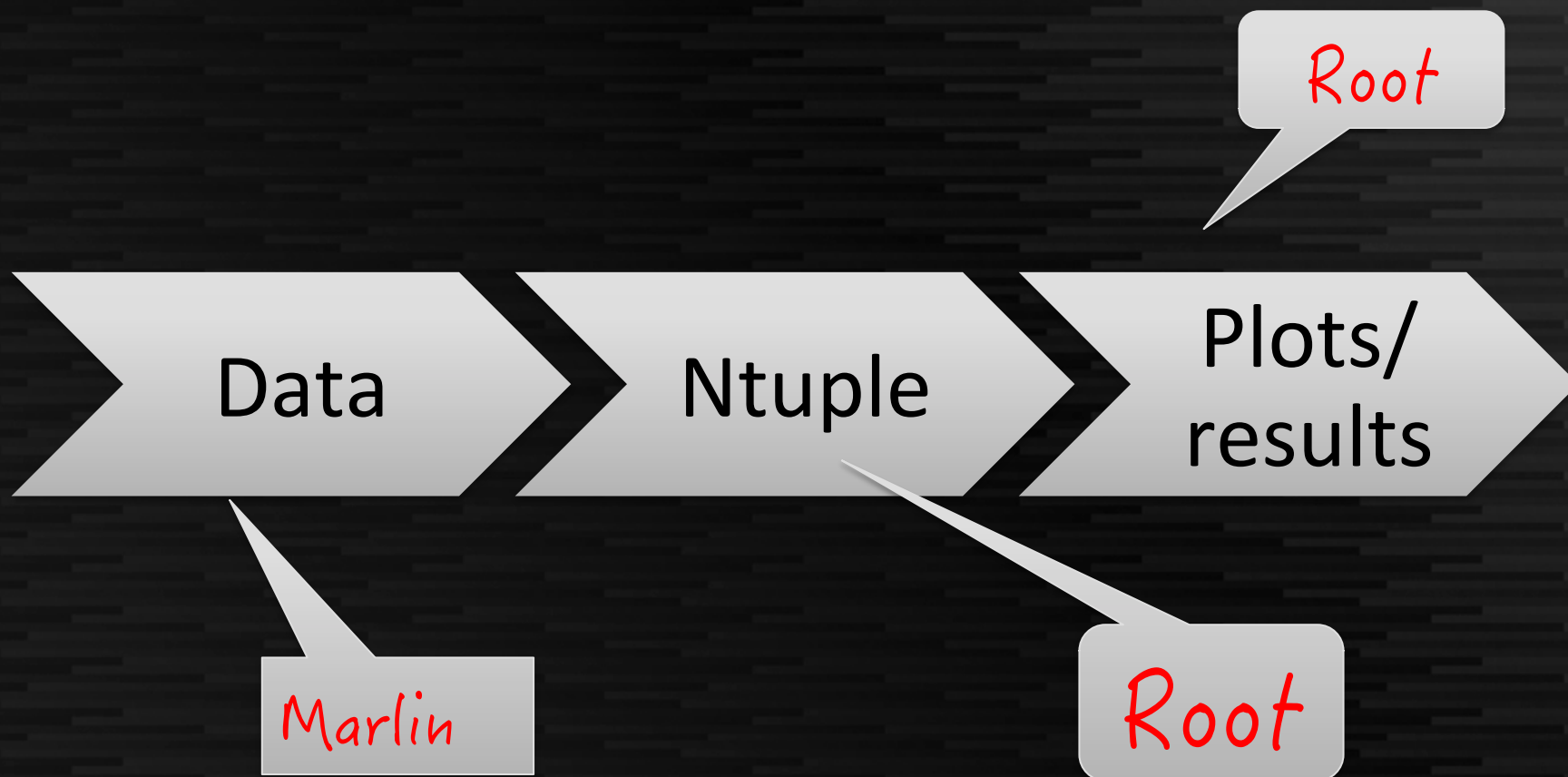
Overview of data-analysis (cont'd)

Feed all types of particle object to the combination engine for further processing

Jets
• tracks
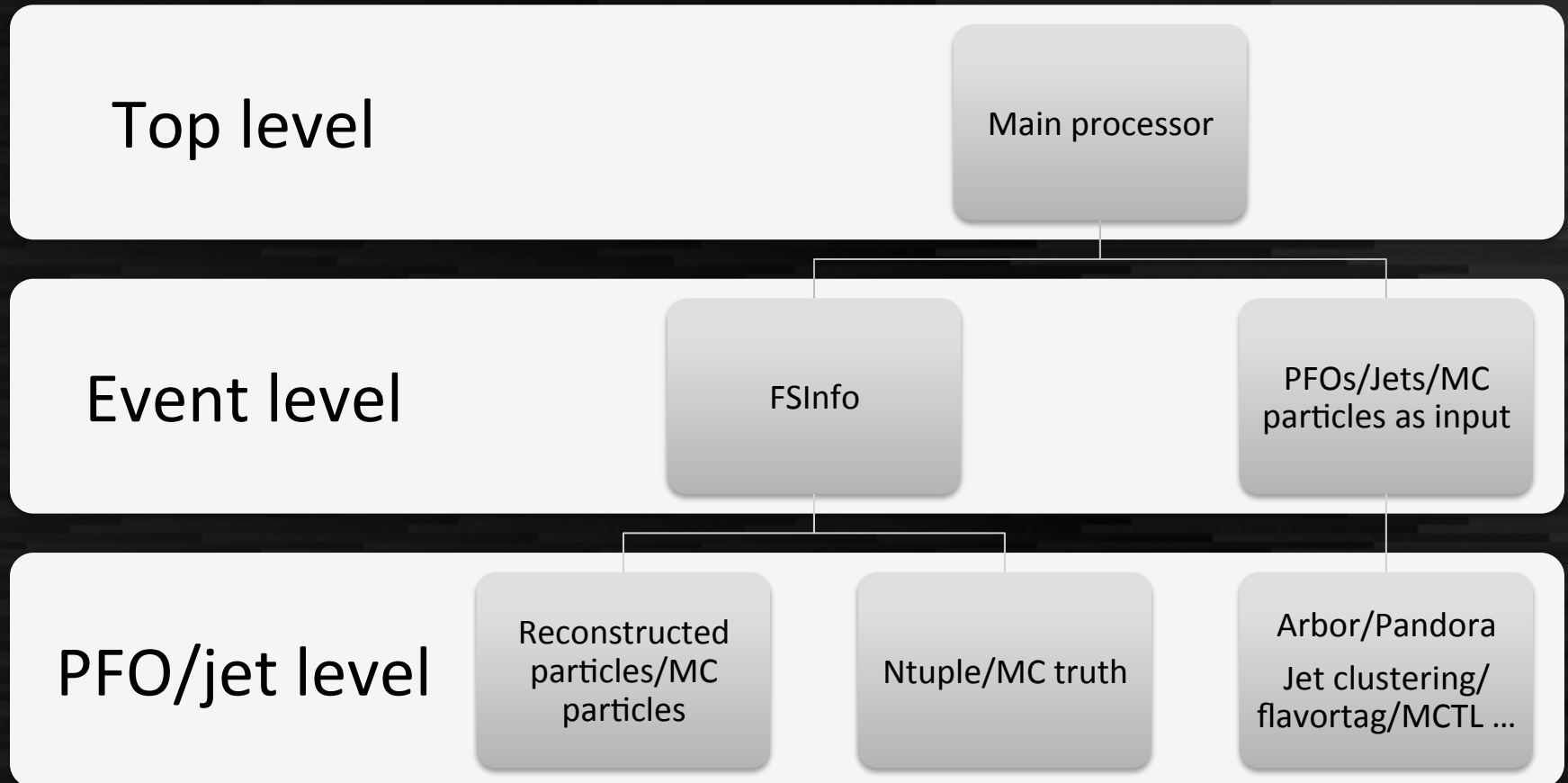• clusters

leptons
• e/μ/τ
• τ finder

photons

Missing particles
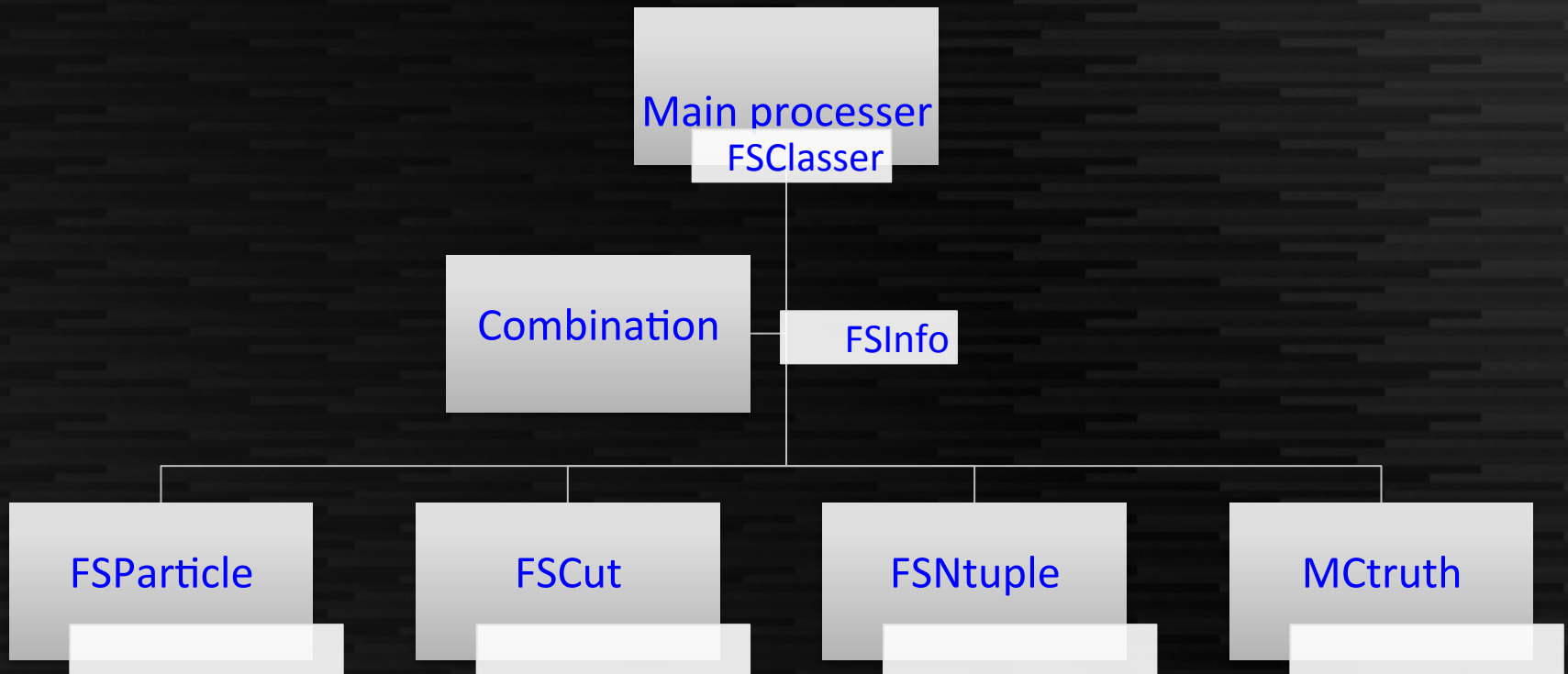
ee+X, μμ+X, jj+ee, jj+μμ ···
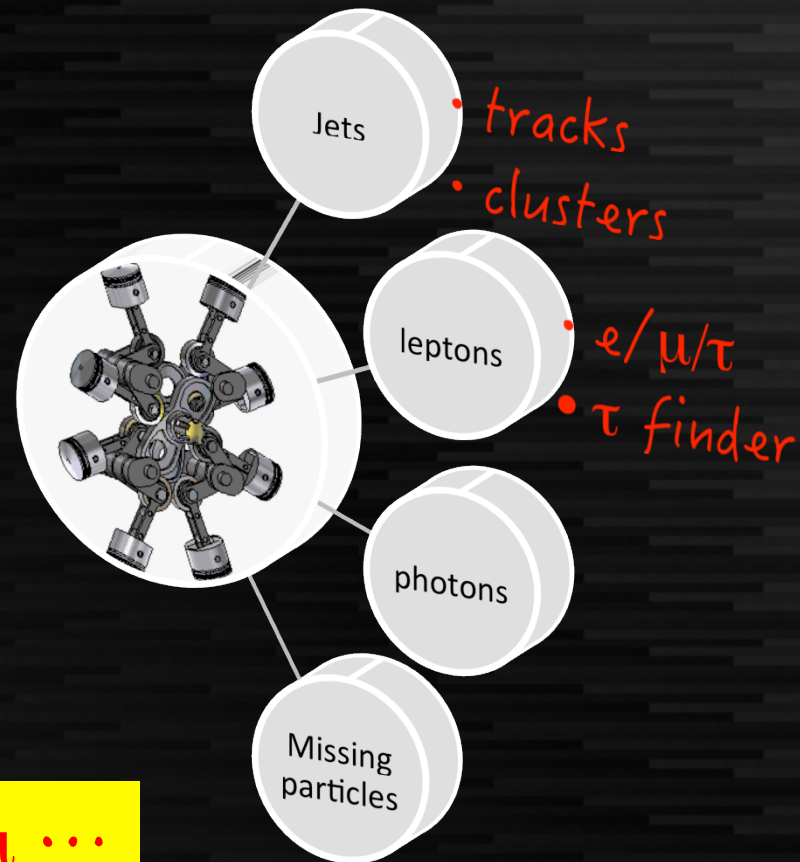
Abstract of tasks

- Class FSParticle → all types of (reconstructed) particles
- Class FSinfo → all kinds of combination
- Class NTupleHelper → service of ntuple
- Class MCTruthHelper → service of MC truth
- Class FSCut → preliminary cuts

Final state classification

Top level — Main processor

Event level — FSInfo · PFOs/Jets/MC particles as input

PFO/jet level — Reconstructed particles/MC particles · Ntuple/MC truth · Arbor/Pandora Jet clustering/flavortag/MCTL …

Structure of code

Main processer
FSClasser

Combination

FSInfo

FSParticle

FSCut

FSNtuple

MCtruth

Structure of main program

Prepare particle lists

e, μ, γ	Jet, τ

↓

Combination the particle list as the request

Events, Ntuple,	MC truth

↓

Loop over all combinations: calculate variables and fill ntuple

Simple cuts	Kinematic constrain

↓

Fill ntuples
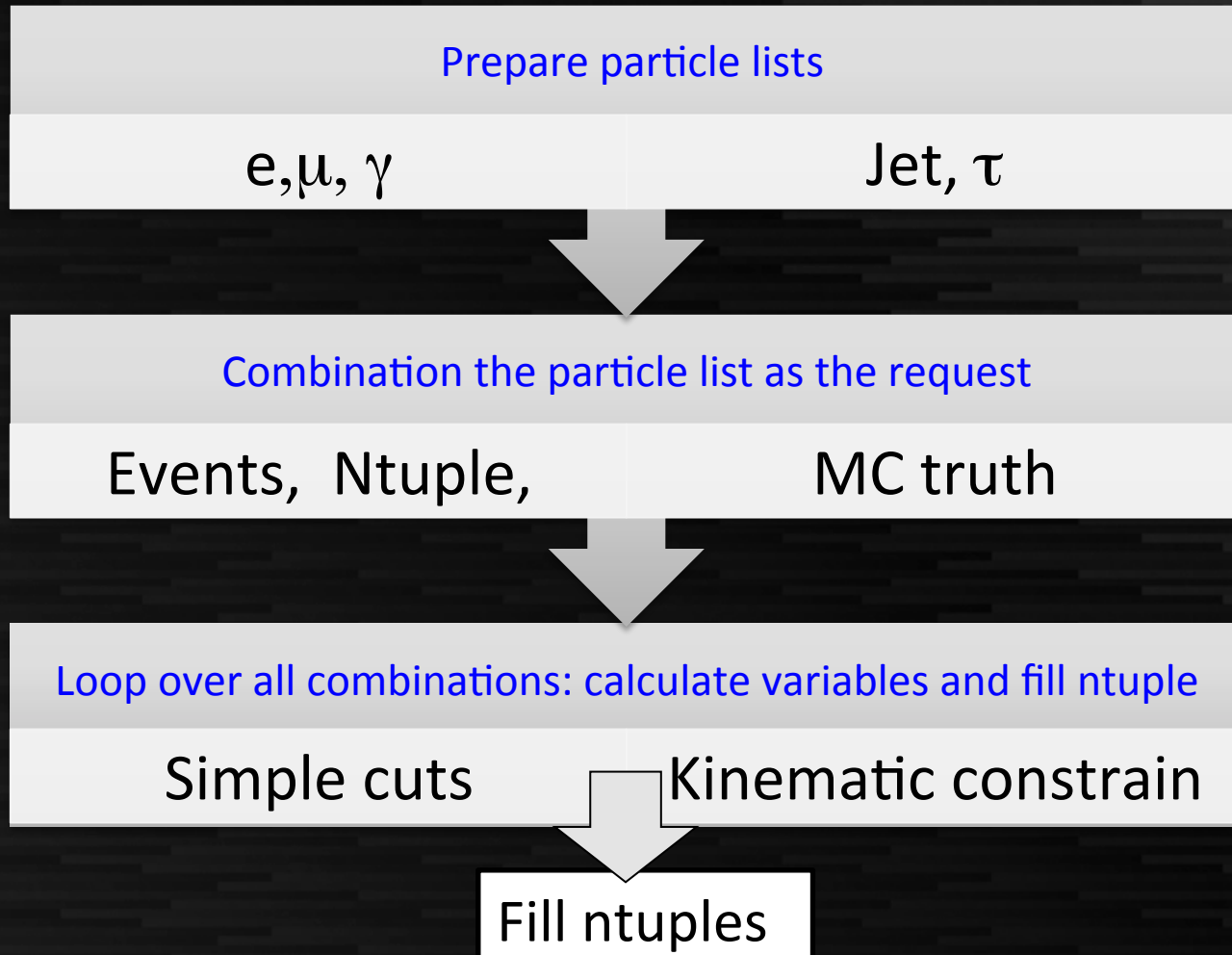
Class FSPaticle

- Data
  - PID/Mass/charge/ 4-momentum, p, pT ...
  - Flavor/vertex
  - Matched MC object

```
ReconstructedParticle *          m_pfo;
MCParticle *                     m_mcp;
//JetFitObject*                        m_JetFitObject;


string      m_name;
int         m_type;
int         m_pdgid;
bool        m_missed;
bool        m_fast;
double      m_mass;
double      m_recmass;
double      m_charge;
double      m_pT;
double      m_pZ;
double      m_Energy;
double      m_Rapidity;
double      m_CosTheta;
double      m_btag;
double      m_ctag;
double      m_bctag;
double      m_flavor;

TLorentzVector        m_rawFourMomentum;
TLorentzVector        m_fitFourMomentum;

vector<int>           m_trackId;
vector<int>           m_showerId;
```

FSInfo

- Data
  - Combination of a list of particles/jets
  - the associated MC truth/ Ntuple
  - Cuts
  - Steers

```cpp
private:
    string m_FSName;
    vector<string> m_particleNames;
    vector<int>    m_particleStatus;
    int            m_nChargedParticles;
    int            m_nMissingParticles;
    NTupleHelper*  m_NT;
    NTupleHelper*  m_NTGen;

    int m_decayCode1;
    int m_decayCode2;

    bool    m_fast;

    bool    m_Constrain4Mom ;
    bool    m_missingMassFit;
    double m_missingMassValue;
    string m_missedParticle;

    vector< vector<unsigned int> >& submodeIndices(const string& submodeN

    vector<FSCut*>                                m_FSCuts;
    vector<vector<FSParticle*> >                  m_particleCombinations;
    map<string, vector< vector <unsigned int> > > m_submodeIndices;
```

compiling
cd build; HFcmake; make install

```
下午]: /cefs/tmp_storage/lig$ /bin/cp -rp /besfs/groups/higgs/users/lig/analysis/FSClasser_v16 .
下午]: /cefs/tmp_storage/lig$ cd FSClasser_v16/build/
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$ /bin/rm -fr *
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$ alias HFcmake='cmake -C ${ILCSOFT}/ILCSoft.cmake ..'
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$ HFcmake > cmake.txt
g initial cache file /afs/ihep.ac.cn/users/m/manqi/Software/ilcsoft/v01-16/ILCSoft.cmake
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$ make > make.txt
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc: In constructor 'FSParticle::FSParticle(EVENT::ReconstructedParticle*,
ollection*, std::vector<EVENT::MCParticle*, std::allocator<EVENT::MCParticle*> >, std::string, double, double, doubl
:
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc:120: 警告: 未使用的变量 'errtheta'
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc:121: 警告: 未使用的变量 'errphi'
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc: In constructor 'FSParticle::FSParticle(EVENT::MCParticle*, std::strir
Vector)':
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc:151: 警告: 未使用的变量 'errtheta'
tmp_storage/lig/FSClasser_v16/src/FSHelper.cc:152: 警告: 未使用的变量 'errphi'
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$ make install
Built target FSClasser
l the project...
tall configuration: "RelWithDebInfo"
talling: /cefs/tmp_storage/lig/FSClasser_v16/lib/libFSClasser.so.0.1.0
to-date: /cefs/tmp_storage/lig/FSClasser_v16/lib/libFSClasser.so.0.1
to-date: /cefs/tmp_storage/lig/FSClasser_v16/lib/libFSClasser.so
runtime path of "/cefs/tmp_storage/lig/FSClasser_v16/lib/libFSClasser.so.0.1.0" to "/cefs/tmp_storage/lig/FSClasser
fs/ihep.ac.cn/users/m/manqi/Software/ilcsoft/v01-16/Marlin/v01-04/lib:/afs/ihep.ac.cn/users/m/manqi/Software/ilcsoft
o/v02-03-01/lib:/afs/ihep.ac.cn/users/m/manqi/Software/ilcsoft/v01-16/mysql/usr/lib64:/afs/ihep.ac.cn/users/m/manqi/
csoft/v01-16/gear/v01-02-02/lib:/afs/ihep.ac.cn/users/m/manqi/Software/ilcsoft/v01-16/CLHEP/2.1.1.0/lib:/afs/ihep.ac
/manqi/Software/ilcsoft/v01-16/ilcutil/v01-00/lib"
下午]: /cefs/tmp_storage/lig/FSClasser_v16/build$
```

Usage: running

- Marlin FS_example.xml

```
[ VERBOSE "FSClasserProcessor"] FSClasser:   Initializing Final State INC2_0000000
[ VERBOSE "FSClasserProcessor"] FSClasser:   Checking the Final State INC2_0000000
FSClasser:        jet:   normal
FSClasser:        jet:   normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:   Initializing Final State INC0_0001100
[ VERBOSE "FSClasserProcessor"] FSClasser:   Checking the Final State INC0_0001100
FSClasser:        mu+:   normal
FSClasser:        mu-:   normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:   Initializing Final State EXC2_0001100
[ VERBOSE "FSClasserProcessor"] FSClasser:   Checking the Final State EXC2_0001100
FSClasser:        jet:   normal
FSClasser:        jet:   normal
FSClasser:        mu+:   normal
FSClasser:        mu-:   normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:   Initializing Final State EXC0_2001100
[ VERBOSE "FSClasserProcessor"] FSClasser:   Checking the Final State EXC0_2001100
FSClasser:      gamma:   normal
FSClasser:      gamma:   normal
FSClasser:        mu+:   normal
FSClasser:        mu-:   normal
Channel    0: INC2_0000000
```

```xml
<execute>
   <!--processor name="MyAIDAProcessor"/-->
   <processor name="RootFileProcessor"/>
   <processor name="MyStdHepReader"/>
   <!--processor name="MyPFAFastSimu"/-->
   <processor name="MyLGFastMCProcessor"/>
   <processor name="MyLGFastJetClustering"/>
   <processor name="FSClasserProcessor"/>
   <!--processor name="MyLCIOOutputProcessor"/-->
</execute>

<global>
   <parameter name="LCIOInputFiles" type="string">
   </parameter>
   <!-- limit the number of processed records (run+evt): -->
   <parameter name="MaxRecordNumber" value="0" />
   <parameter name="SkipNEvents" value="0" />
   <parameter name="SupressCheck" value="false" />
   <!--parameter name="RandomSeed" value="1234567890" /-->
   <!--parameter name="GearXMLFile">./gear_ILD_o2_v06.xml </parameter-->
   <parameter name="Verbosity" options="DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT"> SILENT  </parameter>
</global>

<processor name="RootFileProcessor" type="RootFileProcessor">
  <parameter name="OutputRootFile" type="string">
  FS_example.root
  </parameter>
</processor>
```

```xml
<processor name="MyStdHepReader" type="LGStdHepReader">
    <!--Reads StdHep files as input and creates LCIO events with MCParticle collections.
        Make sure to not specify any LCIOInputFiles in the steering in order to read StdHep files.-->
    <!-- include here the stdhep file path -->
    <parameter name="LGStdHepFileName" type="string">
    /home/ihep/FSClasser/fastsim/E250-CDR_ws.Pe2e2h.eU.pU.001.stdhep
    /home/ihep/FSClasser/fastsim/E250-CDR_ws.Pe2e2h.eU.pU.002.stdhep
    </parameter>
    <parameter name="Verbosity" type="string"> MESSAGE </parameter>
</processor>


<processor name="FSClasserProcessor" type="FSClasserProcessor">
    <!--Name of the MCParticle collection-->
    <parameter name="InputMCParticlesCollection" type="string" lcioInType="MCParticle"> MCParticle </parameter>
    <parameter name="InputMCTruthLinkCollection" type="string" lcioInType="LCRelation"> RecoMCTruthLink </parameter>
    <parameter name="InputIsoLepsCollection"    type="string" lcioInType="ReconstructedParticle"> ArborPFOs </parameter>
    <parameter name="InputPandoraPFOsCollection" type="string" lcioInType="ReconstructedParticle"> ArborPFOs </parameter>
    <parameter name="InputJetsCollection" type="string" lcioInType="ReconstructedParticle"> RefinedJets </parameter>
    <!-- -->
    <!-- -->
    <parameter name="FS130"    type="string">     INC2_0000000 </parameter>
    <parameter name="FS131"    type="string">     INC0_0001100 </parameter>
    <parameter name="FS132"    type="string">     EXC2_0001100 </parameter>
    <parameter name="FS133"    type="string">     EXC0_2001100 </parameter>
    <!-- -->
    <parameter name="FastOrFull"    type="int"   >  0          </parameter>
    <parameter name="ShowMC"        type="int"   >  0          </parameter>
    <!-- -->
    <parameter name="Verbosity"     type="string">  4          </parameter>
    <parameter name="DEBUG"         type="string">  1          </parameter>
    <parameter name="Luxury"        type="string">  1          </parameter>
    <parameter name="MatchMC"       type="string">  1          </parameter>
    <parameter name="TagFlavor"     type="string">  0          </parameter>
    <parameter name="kmfit"         type="string">  1          </parameter>
    <parameter name="Kappa"         type="string">  1.0        </parameter>
    <parameter name="ECM"           type="string">  250.0      </parameter>
</processor>
```
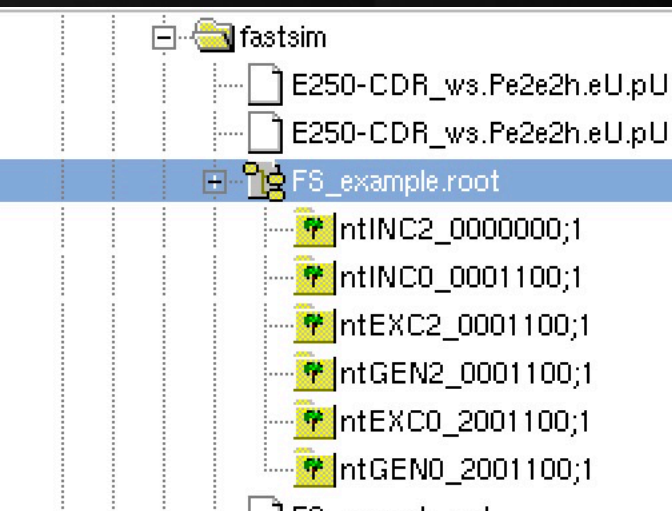
Ntuple

- fastsim
  - E250-CDR_ws.Pe2e2h.eU.pU.
  - E250-CDR_ws.Pe2e2h.eU.pU.
  - FS_example.root
    - ntINC2_0000000;1
    - ntINC0_0001100;1
    - ntEXC2_0001100;1
    - ntGEN2_0001100;1
    - ntEXC0_2001100;1
    - ntGEN0_2001100;1
  - FS_example.xml

ntINC2_0000000;1
- Run
- Event
- Weight
- ntrks
- nclus
- nPFOs
- ntrks_Pandora
- nclus_Pandora
- nPFOs_Pandora
- Pmax
- Emax
- njets
- ntaus
- nElec
- nMuon
- nGamma
- VisEn
- VisPx
- VisPy
- VisPz
- RawAllMass
- y12
- y23
- y34
- y45
- y56
- y67
- nhfs
- MissingMass2
- TotalP
- TotalEnergy

- y56
- y67
- nhfs
- MissingMass2
- TotalP
- TotalEnergy
- TotalPx
- TotalPy
- TotalPz
- JetntrkP1
- JetncluP1
- JetchargeP1
- JetnPFOP1
- JetmassP1
- JetEnP1
- JetPxP1
- JetPyP1
- JetPzP1
- JetPtP1
- JetPtotP1
- JetRapidityP1
- JetcosThetaP1
- JetSphericityP1
- JetPDGIDP1
- JetMcPxP1
- JetMcPyP1
- JetMcPzP1
- JetMcEnP1
- JetAngleRecMcP1
- JetntrkP2
- JetncluP2
- JetchargeP2
- JetnPFOP2

```
if( m_debug>=1 ) printf( ... ... ... ); //e;
NT->fillEvent(evt);
NT->fillDouble("ntrks",        ntrks);
NT->fillDouble("nclus",        nclus);
NT->fillDouble("nPFOs",        nPFOs);
NT->fillDouble("njets",        numberJets);
NT->fillDouble("nElec",        epList.size() + emList.size());
NT->fillDouble("nMuon",        mupList.size()+mumList.size());
NT->fillDouble("nGamma",       photonList.size());
NT->fillDouble("VisEn",        pVis.E());
NT->fillDouble("VisPx",        pVis.X());
NT->fillDouble("VisPy",        pVis.Y());
NT->fillDouble("VisPz",        pVis.Z());
NT->fillDouble("RawAllMass",   pVis.M());
//
NT->fillDouble("yMinus" ,      yMinus  );
NT->fillDouble("yPlus"  ,      yPlus   );
NT->fillDouble("yMinus4",      yMinus4 );
NT->fillDouble("yPlus4" ,      yPlus4  );
```

```cpp
if( rawp4list.size()>1 ){
    for(unsigned int ki=0; ki<rawp4list.size()-1; ki++){
        for(unsigned int kj=ki+1; kj<rawp4list.size(); kj++){
            sprintf(index ,"RMass%d%d",ki+1,kj+1);
            NT->fillDouble((string)index,(rawp4list[ki]+rawp4list[kj]).M());
            if (  m_kmfit>0  ) {
                sprintf(index ,"KMass%d%d",ki+1,kj+1);
                NT->fillDouble((string)index,(kmfp4list[ki]+kmfp4list[kj]).M());
            }
            sprintf(index ,"Rreco%d%d",ki+1,kj+1);
            NT->fillDouble((string)index,(m_ecms-rawp4list[ki]-rawp4list[kj]).M());
            if ( m_kmfit>0 && kmfp4list.size()==rawp4list.size()) {
                sprintf(index ,"Kreco%d%d",ki+1,kj+1);
                NT->fillDouble((string)index,(m_ecms-kmfp4list[ki]-kmfp4list[kj]).M2());
            }
        }
    }
}
```

Summary and discussion

- An usable analysis frame work of full simulation and fast simulation(smearing) is ready

- It is easy to use and you can get ntuple without coding any more

- But the users have to prepare a more sophisticated root script instead

- To be improved

  – More on particle list selection

  – Reduce the Ntuples

  – More on MC truth

  – Something else user requests

  – Any suggestions and contribution are warmly welcome

Printing MC information

Example 1: fast simulation based on stdhep

Skimming data

Make simple plots

μμ H – inclusive analysis

μμ H(jj) – exclusive analysis

jet objects and flavor tag

Done many analysis simultaneously