

MC Topology Analysis

Shu-Xian Du

September 15, 2012



Why

- Inclusive Monte Carlo sample is one of helpful methods for background study.



Why

- Inclusive Monte Carlo sample is one of helpful methods for background study.
- It tells us which process contaminates the signal.



Why

- Inclusive Monte Carlo sample is one of helpful methods for background study.
- It tells us which process contaminates the signal.
- We can optimize our selection to suppress the background.



How — based on ROOT file

- save the Monte Carlo information into the ROOT file



How — based on ROOT file

- save the Monte Carlo information into the ROOT file
- category of topology



How — based on ROOT file

- save the Monte Carlo information into the ROOT file
- category of topology
- output the results



Save MC information

- in the head file of your package, add the following variables:

```
NTuple::Item<int>    m_idxmc;  
NTuple::Array<int>   m_pdgid;  
NTuple::Array<int>   m_motheridx;
```

- in the source file of your package, add:

```
status = m_tuple0->addItem("indexmc",          m_idxmc, 0, 100);  
status = m_tuple0->addIndexedItem("pdgid",      m_idxmc, m_pdgid);  
status = m_tuple0->addIndexedItem("motheridx",  m_idxmc, m_motheridx);
```



Save MC information (cont.)

- in the source file of your package, add:

```
SmartDataPtr<Event::EventHeader> eventHeader(eventSvc(), "/Event/EventHeader");
m_run = eventHeader->runNumber();
m_rec = eventHeader->eventNumber();
if (eventHeader->runNumber() < 0) {
    SmartDataPtr<Event::McParticleCol> mcParticleCol(eventSvc(), "/Event/MC/McParticleCol");
    int m_numParticle = 0;
    Event::McParticleCol::iterator iter_mc = mcParticleCol->begin();
    for (; iter_mc != mcParticleCol->end(); iter_mc++) {
        .....
        m_pdgid[m_numParticle] = pdgid;
        m_motheridx[m_numParticle] = mcidx;
        m_numParticle += 1;
    }
    m_idxmc = m_numParticle;
}
```

- run the job, get one ROOT file.



Category of Topology

- prepare one card named “topo.cards”

```
#tree name
gghh
#root file name
path/topo1.root
path/topo2.root
INCLUSIVE PARTICLE
-3122
TOPO START
#pi+ pi- pi0
0 100443 -1
1 111 0
2 -211 0
3 211 0
4 22 1
5 22 1
```

note:

- the line includes something as the following is ignored: blank line; begin with “#” or white space.
- don not change the order of tree name, the file name and the topology.
- the number of ROOT fille may be more than one
- set the PID for inclusive decay, otherwise set the PID to 0
- TOPO START must be there.
- pay attention to the order of the decay chain of the signal.
- the topology of the signals may be one or more.

- execute the command under the current directory:
topology



The Results

Two files are generated: “mclist.txt” and “topo.root”. In the first file, you will see something like:

```
topology: 0           Events: 476

      Event listing (summary)
I      particle/jet      KF      orig
0      psi(2S)           100443    -1
1      pi0               111       0
2      pi-               -211      0
3      pi+               211       0
4      gamma             22        1
5      gamma             22        1

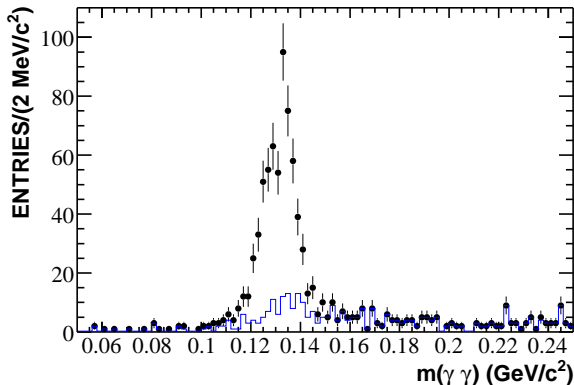
topology: 1           Events: 148
.....
```

This means there are 476 events of $\psi(2S) \rightarrow \pi^+ \pi^- \pi^0$.



The Results (cont.)

In the file “topo.root”, you can see the background level when you execute the macro “draw.cxx” under ROOT environment.



One Example

You can see the example under

`/ihepbatch/bes/dusx/common/topo`

Material for head file and source file can be found in “head.h”
and “source.cxx”.

Any problem please tell me!

Thank You!

