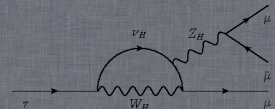
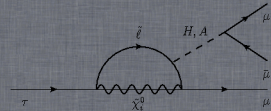
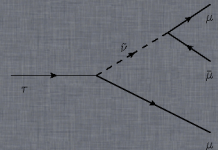
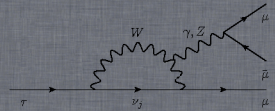


Software development

Marcin Chrząszcz

Institute of Nuclear Physics,
Polish Academy of Science,
and INFN sezione di Pisa

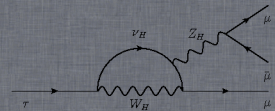
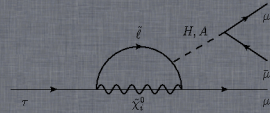
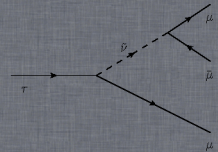
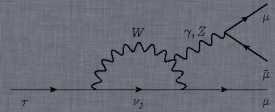
23th January 2013



Given task

Cod structure

Tests

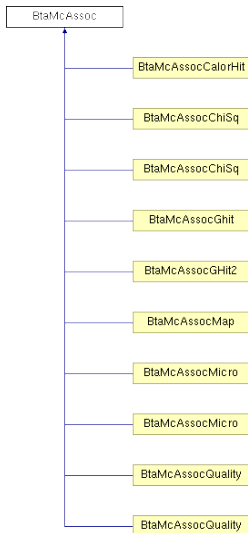


Given task

- 1 Truth Matching Ignores Final State Radiation(FSR).
- 2 Abi example: $D^0 \rightarrow \pi K$ will not give MCtruth because the decay on list is $D^0 \rightarrow \pi K \gamma$
- 3 Task: correct this problem, with a on/off switch.

Cod structure

- The Top class is BtaMCAssoc, however the function is virtual =(
- It's also defined in BtaMcAssocGHit, BtaMcAssocMicro, BtaMcAssocQuality, BtaMcAssocQuality, BtaMcAssocGHit
- They are distributed over packages: BetaCoreTools, BetaMC and BetaMicroTruth.
- Call mcFromReco is from BtaMcAssocMicro. Than it's checked that particle is a composite and if it is than mcFromRecoComposite is called.

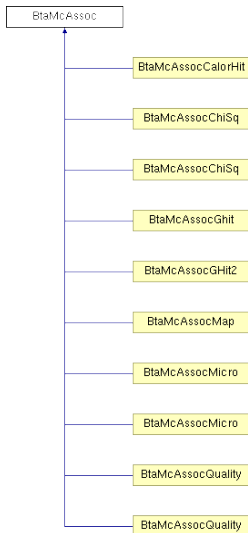


Cod structure

Listing 1: Original declaration:

```
// Abstract interfaces
virtual BtaCandidate* recoFromMC(
    BtaCandidate const *mc, int
    which = 0, bool checkMassHypo=
    false) const = 0 ;

virtual BtaCandidate*
    mcFromRecoComposite(BtaCandidate
    const *reco, int which =0, bool
    checkMassHypo=false) const;
```



Cod structure

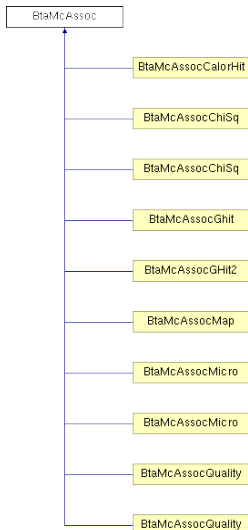
Listing 2: Implement switch:

```
// Abstract interfaces
virtual BtaCandidate* recoFromMC(
    BtaCandidate const *mc, int
    which = 0, bool checkMassHypo=
    false, bool FSR=false) const =
    0 ;

virtual BtaCandidate*
    mcFromRecoComposite(BtaCandidate
    const *reco, int which =0, bool
    checkMassHypo=false, bool FSR=
    false) const;
```

Definition of those functions have to be change in every inheritance class.

There is a default argument set to false so old code written by others will not see the difference.



Real code

Listing 3: Implement switch:

```
if(aMother != 0 && aMother->nDaughters() != listOfDaughters.
    size()) {
    // the mother has more daughters than on the list
// something not impotent ...
    if(FSR)
        {
            if( dau->pdtEntry() != 0 )
                {
                    string name( dau->pdtEntry()->name() );
                    if( name.find( "gamma" ) != string::npos){
                        std::cout<<"SUCCESSSSSS!!!!!!!!!!, we found
                            gamma"<<std::endl; // to be commented
                                out
                                    continue;
                                        }
                                            } // if we have pdt entry!
                                                } // end of FSR
```

Tests

I am lazy => For The $D^0 \rightarrow \pi K$ there exists a working code in the tutorial:

LINK

Before the modifications

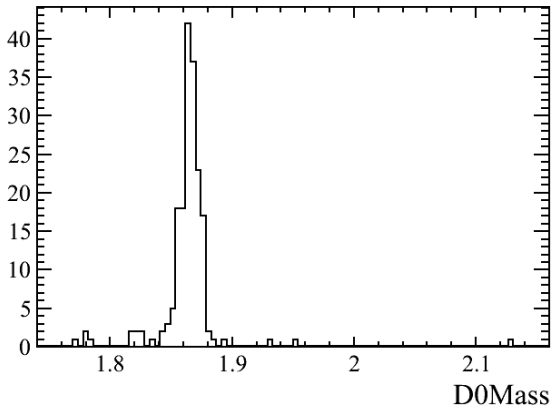


Figure: 183 events in data sample

After modifications and `FSR==true`

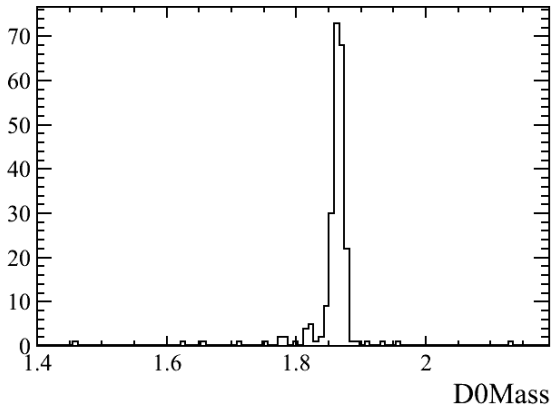
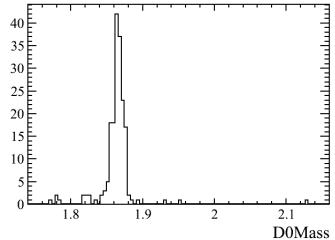
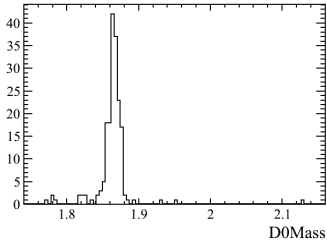


Figure: 230 events in data sample

For now looks ok.

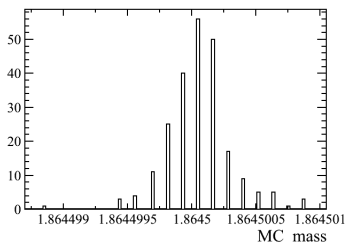
Consistency test old vs new with FSR == false



Spot difference =P. They have the same number of events: 183.

Consistency test 2

Do we really get truth D0 in ntuple?



I don't know different particle which mass so close to D^0 so I think I can declare it that it works. (I also worked in decay tree =)