

Oh no,

now I'm more confused than before. The GetMcTruth() seems to be too difficult for me. I simulated 2000 events and used the following code:

```
PndAnalysis* theAnalysis = new PndAnalysis("SttMvdGemGenTrack","FtsIdealGenTrack");
theAnalysis->FillList(piplus, "PionLoosePlus",pidalgos);
theAnalysis->FillList(mcpiplus, "PionLoosePlus");
for (Int_t iplus=0;iplus<piplus.GetLength();++iplus){
    if(piplus[iplus]->GetMcTruth()) {
        if(!theAnalysis->McTruthMatch(piplus[iplus]))
            //(1) pointer set, but no McTruthMatch
    } else
        //(2) pointer not even set
}
for (Int_t iplus=0;iplus<mcpiplus.GetLength();++iplus){
    if(mcpiplus[iplus]->GetMcTruth()) {
        if(!theAnalysis->McTruthMatch(mcpiplus[iplus]))
            //(3) pointer set, but no McTruthMatch
    } else
        //(4) pointer not even set
}
```

On every marker, I implemented a counter, to check, what is working in which way. But the result was again confusing.

| has not even a truth pointer | has a truth pointer but McTruthMatch failed | number of particles

PID	15		2809		9031	
MC	28		847		9558	// in this case:

MC=PidAlgoIdealCharged

I understand, that with a pid algorithm the McTruthMatch failed so often. I mean, the actual pid is crap (or "not the best one"). As I understood right, the pointer is set after track reconstruction, and just the particles are identified wrong. Even that some of them didn't have a pointer makes sense to me. But why are in the PidAlgoIdealCharged-list so many candidates without a pointed and, that's what I was really wondering about, didn't pass the McTruthMatch!?

This time, I'm using trunk, rev 21245!!!

Additionally: What are the criteria for setting the truth pointer? And shall I change theAnalysis->FillList(mcpiplus, "PionLoosePlus") to something like theAnalysis->FillList(mcpiplus, "PionAllPlus");

```
for(int i=0;mcpiplus.GetLength(),i++) {
    if(theAnalysis->McTruthMatch(mcpiplus[i]))
        mcpiplustrue.Add(mcpiplus[i]); // mcpiplustrue as a new RhoCandList
```

}

Thanks in advance
Simon
