Subject: Using Rho without MC info
Posted by Marcel Tiemens on Wed, 05 Aug 2015 14:33:48 GMT
View Forum Message <> Reply to Message

Hello Rho experts,

In order to evaluate my clustering algorithm in the PandaRoot framework, I would like to analyse the reconstructed clusters with Rho. Basically, I want to reconstruct the parent particle(s). However, after running a macro (which is based on ana_complete.C in /macro/run/, but could very well be flawed as I am no Rho expert), I get the following error (a lot):

Error in <BuildMcCands>: MC track Array does not exist.

My interpretation of this is that it looks for MC truth information, which is not there because the current implementation of the timebased simulation for the EMC is not yet compatible with MC backprogation (Tobias is working on this I understood). The absence of this information is not such a big problem at this stage, however. To circumvent this issue, I would like Rho to just take the EmcClusters from the reco macro and basically treat these as photon candidates and then continue the analysis under this assumption. The question is, I don't really know how. I guess it should be relatively easy, but can anyone tell me?

Also, just to check, to reconstruct a neutral pion, I would need to run something like
pion.Combine(gamma,gamma);
, correct? Where "pion" and "gamma" are RhoCandLists. This part, concerning neutral particles, is not covered in the Rho tutorial, which is why I'm asking.

Thanks a lot in advance!

Subject: Re: Using Rho without MC info
Posted by StefanoSpataro on Wed, 05 Aug 2015 15:20:30 GMT
View Forum Message <> Reply to Message

Hi,
would it be maybe better to change the PndPidCorrelator so that for neutrals it takes the proper MC info from the timebased simulation code and FairLinks, and not the standard GetMcIndex? At present the interface to the analysis should be changed to use the FairLinks, but it still uses the old index structure.

Subject: Re: Using Rho without MC info
Posted by Klaus Götzen on Thu, 06 Aug 2015 05:57:13 GMT
View Forum Message <> Reply to Message

Hi Marcel,

I'm not completely sure whether I understand your problem. One thing is MC truth information, another is combining RhoCandLists, and I don't see a big connection between them. Combining neutrals exactly works like combining charged particles, so it should work like you wrote in case you have proper RhoCandLists.

If you'd like to build your own candidates (from e.g. your clusters), so that you can combine them via RhoCandList::Combine(), you can take a look how candidates are created from FairRecoCands in PndAnalysis::ReadRecoCandidates(). You would have to create FairRecoCands from your clusters beforehand. Important is to give each RhoCandidate a uid (unique id) in case you need overlap control, since the bitmarkers are set according to it. For pi0 -> gamma gamma it shouldn't play a role though.

Best,
Klaus

Subject: Re: Using Rho without MC info
Posted by Marcel Tiemens on Thu, 06 Aug 2015 09:10:30 GMT
View Forum Message <> Reply to Message

The analysis macro gives a lot of warnings and errors that it can't find certain branches. For most of them, this isn't surprising, since I'm only running with the EMC; no other subdetectors are present. This one bugs me however:

[INFO] Branch: PidNeutralCand not found in Tree
[INFO] Branch: PidNeutralCand not found in Tree
-I- PndAnalysis::ReadTCA(): No PidNeutralCand array found.

Or can the PID macro not construct a list of neutral candidates using the EMC information only? If it needs information from other subdetectors, that could be related to why it doesn't work, seeing as these aren't there.

And I don't know how much of a problem it is that this occurs:
Error in <BuildMcCands>: MC track Array does not exist.

I'm running the full simulation by the way.

Subject: Re: Using Rho without MC info
Posted by StefanoSpataro on Thu, 06 Aug 2015 18:28:11 GMT
View Forum Message <> Reply to Message

Hi,
if you check PndPidCorrelator::ConstructNeutralCandidate(), that is the function which is contructing the NeutralCand starting from the emc bumps (or clusters). You should modify it to have the analysis working with neutrals and the new code.

Subject: Re: Using Rho without MC info
Posted by Marcel Tiemens on Fri, 07 Aug 2015 14:33:40 GMT

Thank you for that suggestion. A small update... I sort of managed to get it to work by disabling the "return kERROR" statement, so that the PidCorrelator doesn't stop when it doesn't find a track array, and commenting out the part in ConstructNeutralCandidate() that tries to match tracks. If I'm honest, I don't what kind of tracks it would need. Surely, for neutrals you don't need a complete track; the starting point (=interaction vertex) and end point (=EMC cluster position) should uniquely identify the track, right? Especially in this case it doesn't matter, since there is nothing in between. Perhaps it would be possible to use a Kalman filter or something to interpolate the "track", but I am no expert. The default clustering algorithm does something with the tracks, using

```
FairMultiLinkedData hitLinks =
theDigi->GetLinksWithType(FairRootManager::Instance()->GetBranchId("EmcHit"));
    for (Int_t j = 0; j < hitLinks.GetNLinks(); j++) {
        PndEmcHit* hit = (PndEmcHit*)fHitArray->At(hitLinks.GetLink(j).GetIndex());
   if(hit) cluster->AddTracksEnteringExiting(hit->GetTrackEntering(), hit->GetTrackExiting());
 else std::cout << "-E in PndEmcMakeCluster::Exec FairLink " << hitLinks.GetLink(i) << "to
EmcHit delivers null" << std::endl;
    }
```

However, as Philipp explained in his email (you were included there, Stefano), this part is not compatible with the new timebased simulation for the EMC.

Anyway, the analysis macro crashes after leaving my computer unresponsive for some 15 minutes with the message:

```
[INFO   ] The number of entries in the tree is 1000
[INFO   ] FairRootManager::ReadEvent(1): The tree has 1000 entries
evt 2
evt 3
evt 4
evt 5
evt 6
evt 7
evt 8
evt 9
evt 10
terminate called after throwing an instance of 'std::bad_alloc'
  what():  std::bad_alloc
```

It is of course very possible that I am at fault here; I'm only just learning this Rho stuff. I have both pi0's and eta's, which both decay to photons, but it should be possible to distinguish the two? Or do I need to do something special for that? Perhaps it is better if I share the file I use for the analysis, so you can take a look and tell me if I'm doing something stupid... Or if is has to do with the tracks part. ... Okay, maybe it wasn't that small of an update

File Attachments

1) emc_anatest.C, downloaded 532 times

Subject: Re: Using Rho without MC info

Posted by Marcel Tiemens on Wed, 16 Sep 2015 13:11:31 GMT
View Forum Message <> Reply to Message

Did anyone have a chance to take a look at this?

Subject: Re: Using Rho without MC info
Posted by Ralf Kliemt on Wed, 16 Sep 2015 14:37:53 GMT
View Forum Message <> Reply to Message

Hi Marcel,

As far as the Analysis & Rho side is concerned, try the patch in the spoiler below. Basically now it tries to create MC Cands even if we have no MC Track array. This results in a message for each event. With the patch that message won't appear.

As the pi0 combinations go:

```
RhoCandList gammas, pi0s;
while (theAnalysis->GetEvent() && i++<nevts){
  theAnalysis->FillList(gammas, "Neutral");
  pi0s.Combine(gammas,gammas);
  ...
}
```

Cheers!
Ralf

Toggle Spoiler

```
Index: PndAnalysis.cxx
===================================================================
--- PndAnalysis.cxx     (revision 28526)
+++ PndAnalysis.cxx     (working copy)
@@ -143,13 +143,13 @@

     if ( ! fMcTracks && fVerbose ) {
       std::cout << "-W- PndAnalysis::Init(): No \"MCTrack\" array found. No MC info available."
<< std::endl;
-    }
+    } else {
+      fBuildMcCands = true;
+      fMcCands = new TClonesArray ( "RhoCandidate" );
+      // next line commented by KG, 07/2012
+      fRootManager->Register ( "PndMcTracks","PndMcTracksFolder", fMcCands, kFALSE );
+       }

-    fMcCands =new TClonesArray ( "RhoCandidate" );
-
-    // next line commented by KG, 07/2012
-    fRootManager->Register ( "PndMcTracks","PndMcTracksFolder", fMcCands, kFALSE );
-    fBuildMcCands = true;
   }
```

```
   fChainEntries = ( fRootManager->GetInChain() )->GetEntries();
@@ -440,18 +440,17 @@
 {
   int i;
   // Make Monte-carlo truth candidates by the reconstructed particles up to the initial state (if
available)
-  if ( !fMcCands ){
-    Warning("PndAnalysis::BuildMcCands","No array to store candidates...");
-    return;
-  }
   if ( !fBuildMcCands ) {
     if(fVerbose) Info("PndAnalysis::BuildMcCands","No mc to build...");
     return;
   }
+  if ( !fMcCands ){
+    Warning("PndAnalysis::BuildMcCands","No array to store candidates...");
+    return;
+  }
   if ( fMcCands->GetEntriesFast() != 0 ) {
     fMcCands->Delete();
   }
-
   if ( fMcTracks == 0 ) {
     Error ( "BuildMcCands","MC track Array does not exist." );
     return;
```

Subject: Re: Using Rho without MC info
Posted by Marcel Tiemens on Fri, 18 Sep 2015 11:10:11 GMT
View Forum Message <> Reply to Message

Hi Ralf,

Thanks for your suggestion. I tried it out, and indeed there are no more error messages. However, when using your patch in conjunction with the hack I proposed earlier (in PndPidCorrelator::ConstructNeutralCandidate()), the macro doesn't finish properly... It finishes without giving any errors, but also doesn't print the lines "Macro finished successfully." etc, which the according to the macro clearly should happen. The resulting root file also doesn't contain a tree. Running it without the hack I mentioned earlier causes the crash in my previous message:

terminate called after throwing an instance of 'std::bad_alloc'
what(): std::bad_alloc
