



ECL MC-Matching

Status Update

B. Oberhof
LNF-INFN Frascati, Italy

19th March 2015

Reminder

- Many different flaws in ECL MC-Matching in old code (end 2014) which have now been disentagled:
- Analysis package related:
 - Bkg index not properly set → all clusters associated to MCParticles (and other complications) (fixed by A. Zupanc after last B2GM)
 - EclCluster quantities not dumped for charged candidates (fixed by A. Zupanc after November14 B2GM)
- ECL package related
 - Daughters of decaying particles not treated properly in ECLMCMatching module → clusters associated to decaying particle (fixed before last B2GM)
- No proper treatment of multiple-association implemented in ECLMCMatching module (still open task at last B2GM):
 - Impossible to separate bkg and physics contribution (crucial for analysis)
 - Impossible to distinguish properly overlapping MCParticles (less frequent in practice but important for code consistency)

Fixed!

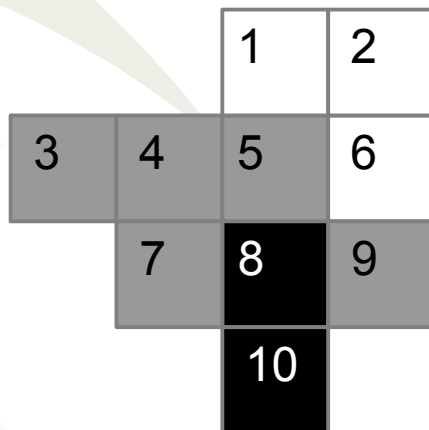
- In old eclMCMatching module:
- Association was done 1 (cluster) to 1 (MCParticle)
- The MCParticle giving highest contribution to the cluster (in terms of eclDigits) was associated with it
- A eclDigit is associated to a particle if the corresponding eclHit is, no matter what is its real contribution in terms of E
- Independently of proper multiple-matching this is conceptually wrong!

2 photons, 1 cluster

— phot. 1
— both
— phot. 2

TrueE(1)=200 MeV
TrueE(2)=300 MeV

ClusterE=680 MeV



Digit content:

1 - 100
2 - 120
3 - 300
4 - 450
5 - 1200
6 - 150
7 - 800
8 - 1000
9 - 300
10 - 300

Sum(1)=4050

Sum(2)=3420

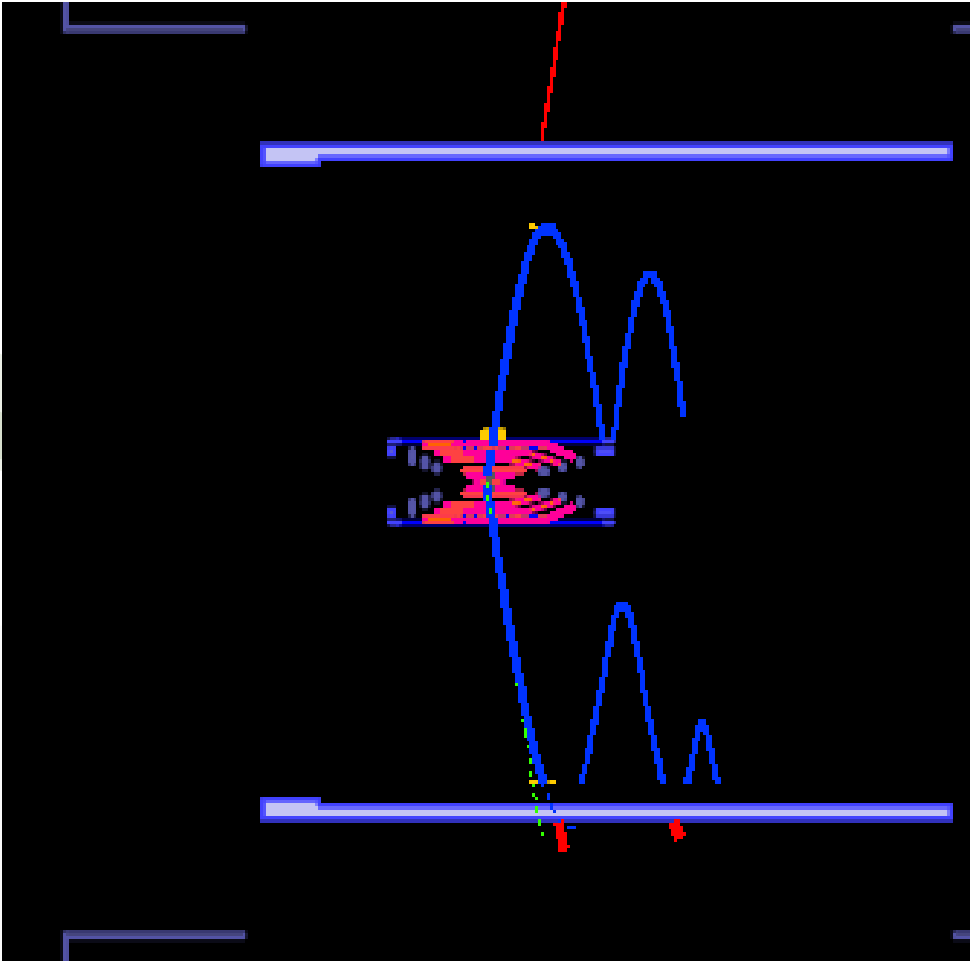
MCParticle 1
is associated
to 680 MeV

- 2 new modules in package right now:
 - ECLMCMATCH:
temporary fix requested by analysis people at last B2GM. Module handles multiple association but gives no weights. Relation between MCParticle and eclCluster is set only if energy contribution in terms of hits is $>$ than a fixed amount (50%, empirically tuned). Has been tested and appreciated by analysis people (A. Zupanc 12th March SW meeting).
 - ECLMCMatching (new version):
Tentative final version. Handles properly multiple association, weights are calculated based on SimHits for every MCParticle and stored in the relation array. Distinction between signal and bkg is based on bkgTag. Still alpha version.

- As far as I have tested ECLMCMatching module works properly BUT:
- Feedback by analysis people would be highly desirable asap
- Probably some mods to NtupleMaker (Tools) are necessary to properly handle the weights (obviously not needed for ECLMCMatch module)
- Individual bkg contributions cannot be distinguished with current module (no SimHit info available for bkg). Is this something that we want/need?
- Association is not “unitary”, i.e. the sum of contributions from eclSimHits does not equals eclClusterEnergy (minimal violation, few % level)
- Technical: being based on the old module the new one first sets relations for eclShowers and then derives them for eclClusters. Should this be avoided? (Not difficult to modify)

- ECLMCMatching seems to be fixed at least for what concerns ecl package (some work on NtupleMaker may be needed)
- Validation:
 - Some problems with validation plots (do not appear as they should, sometimes the scripts fail → under investigation)
 - Scripts using “old” ECL objects (e.g. ECLPi0s, ECLGamma, etc.) have to be substituted with actual analysis objects
 - What should be monitored by us (and what not)? (e.g. Pi0 ..)
- Software quality:
 - Till 1 week ago: dependencies: 7+7 missing/extra, cppcheck: 196 warnings/34 remarks, documentation: 212 missing
 - I started to work to address this issues, in future refer to: https://belle2.cc.kek.jp/internal/development_build/index.html
- If the latest schedule holds (only Digitizer missing): what's next?

Is our simulation reliable?

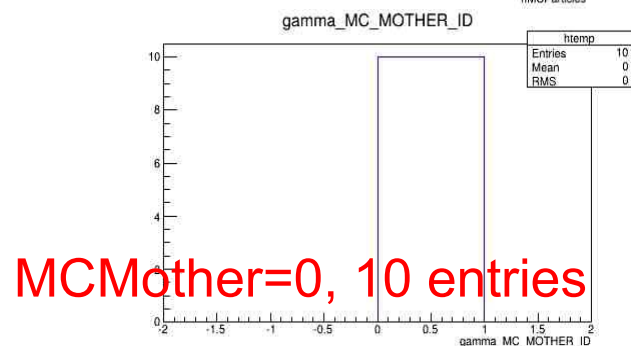
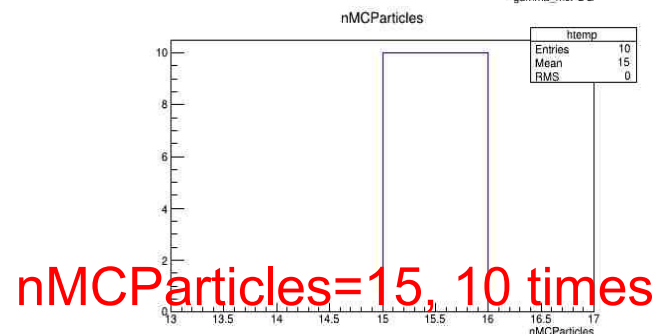
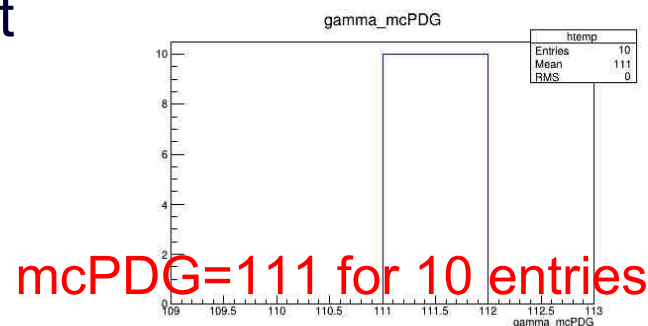
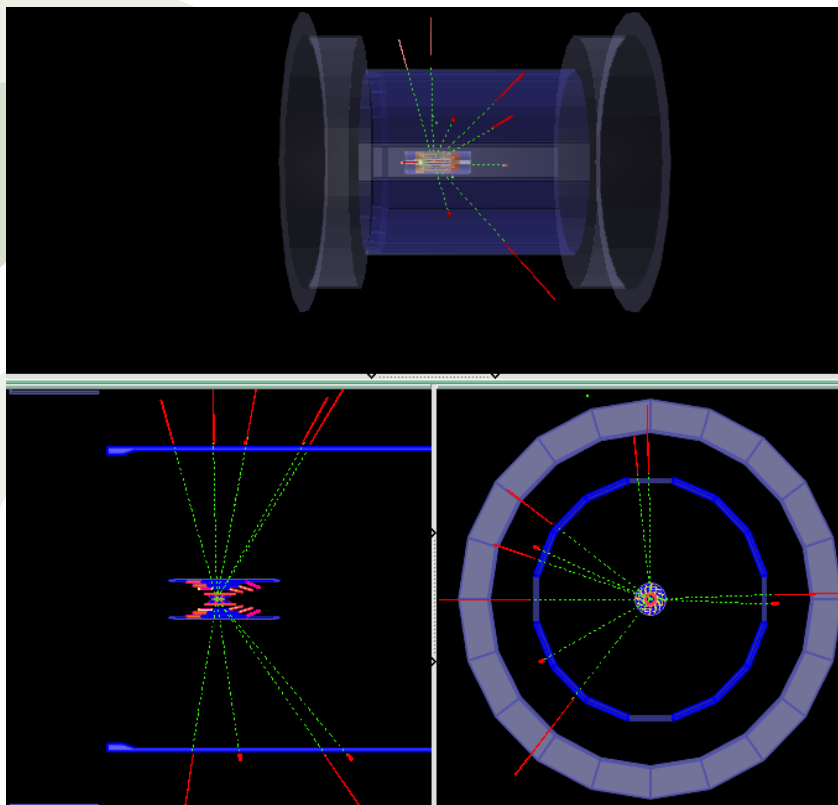


Backups

MCMatching bugs

From last B2GM

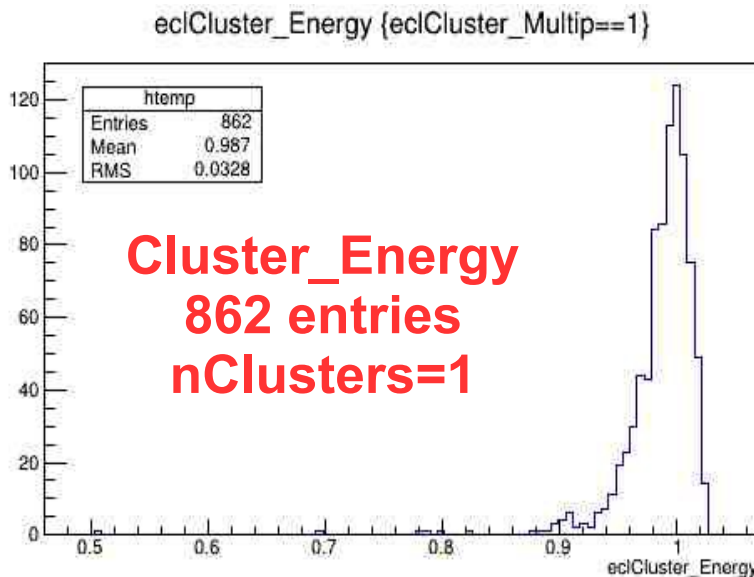
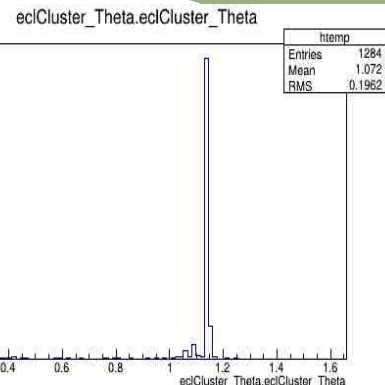
- Many flaws in ECL related MC matching
- e.g. 5 neutral pions, 1 GeV each, 1 evt
- Build-2014-09-01, default Sim&Reco,
- Flat ntuple using NtupleMaker



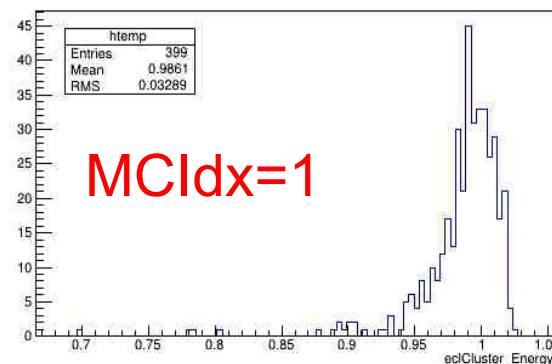
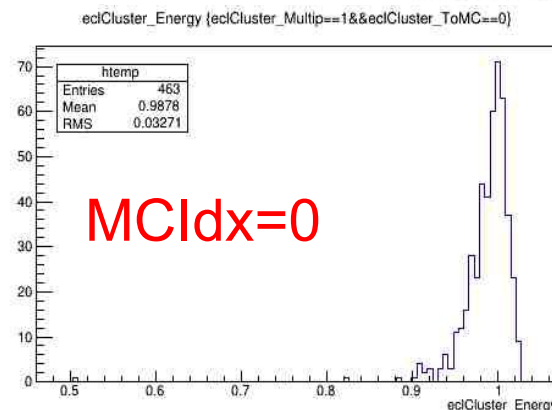
Multiple matching

From last B2GMC

- Two “nearby” photons $\Delta\theta < 1^\circ$, $\Delta\phi < 1^\circ$
- 500 MeV each, single cluster 1 GeV
- Cluster info is not shared between the MC cand



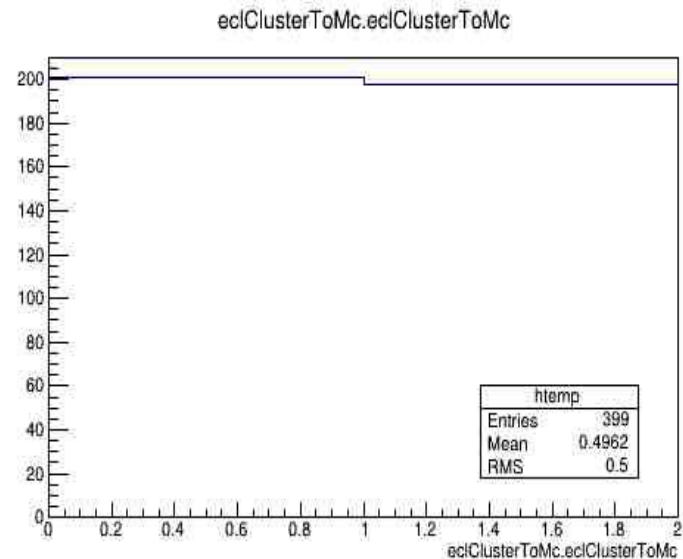
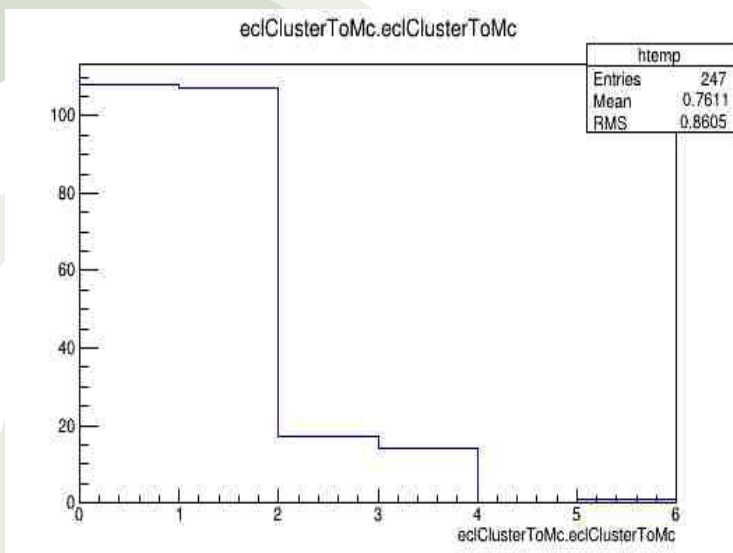
MCCandidate_Idx associated to cluster



- Distinguish individual energy releases crucial when bkg is added
- We are currently at work on that: weights will be introduced to handle multiple association and bkg

Matching fixes

- Previous bugs were related to improper treatment of unstable particles and their decay products, example:
- 1st case: 2 photons 1 GeV each, 100 evt → resulting clusters are correctly associated to primary photons
- 2nd case: 2 π^0 , 500 MeV each, 200 evt → all resulting clusters are associated to the primary particles

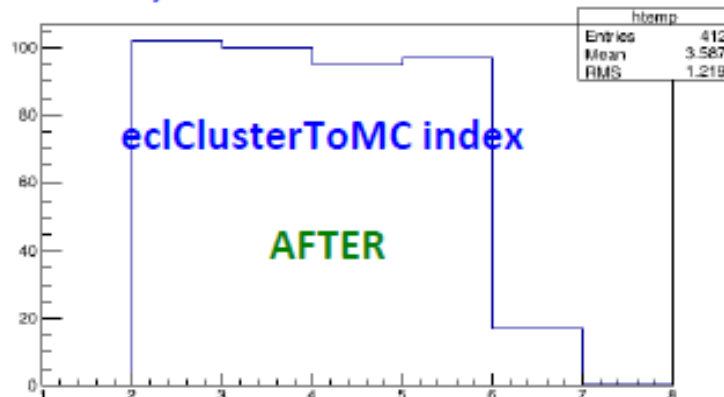
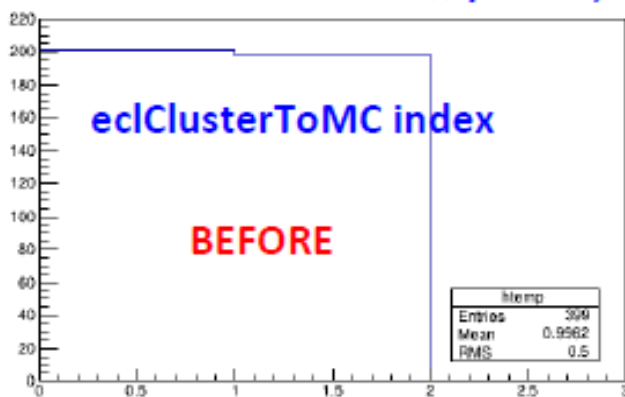


- As far as I have tested ECLMCMatching module works properly BUT:
- Extensive test by analysis people would be highly desirable
- Probably some mods to NtupleMaker (Tools) are necessary to properly handle the weights (obviously not needed for ECLMCMatch module)
- Bkg contribution cannot be separated with current module (no SimHit info available for bkg), is it something that we want/need?
- Association is not “unitary”, i.e. the sum of contributions from eclSimHits does not equals eclClusterEnergy (minimal violation, few % level)
- Technical: being based on the old module the new one first sets relations for eclShowers and then derives them for eclClusters. Should this be avoided? (Not difficult to modify)

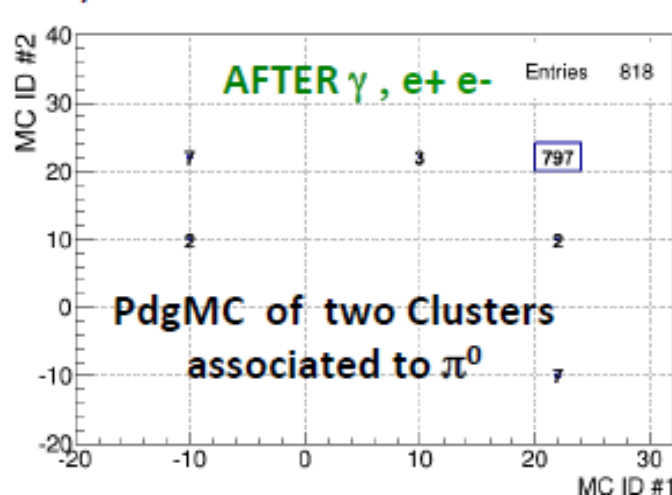
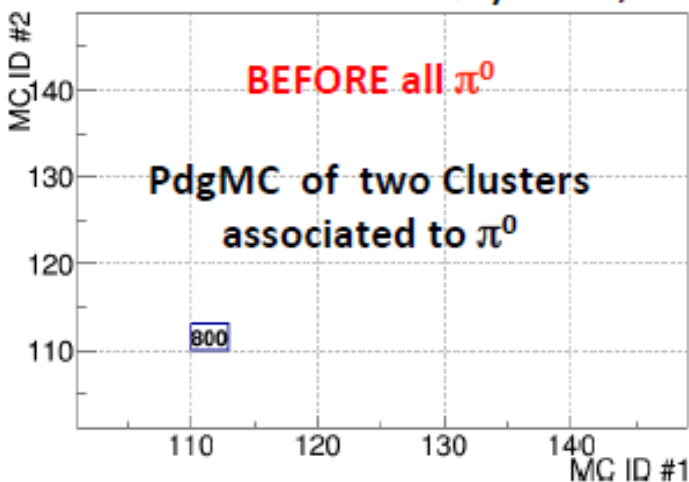
Matching fixes (2)

- Previous flaws in MC matching have now been fixed
- Fixes will be included in next integration build

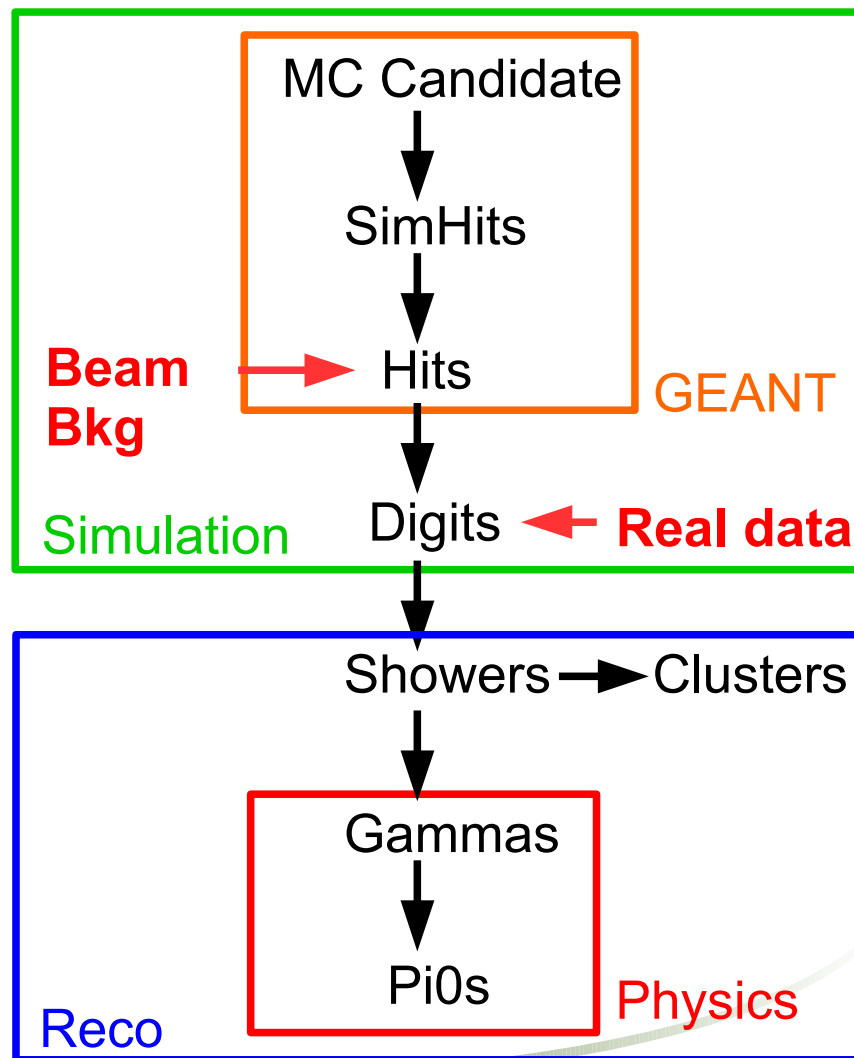
$2 \pi^0/\text{event}$, $E=0.5 \text{ GeV}$, 100 events



π^0/event , $E=0.5 \text{ GeV}$, 1000 events



- Proper multi-particle treatment is still missing
- MCMatching for reconstructed quantities is done in ECLMCMatching module
- In ECLMCMatching information about MCParticles is inherited from EclHits and propagated
- Matching for Hits is done at Simulation stage (SensitiveDetector.cc)
- Idea is to use HitEnergyDep as weights at Hits level and then propagate them to the final analysis objects



- All fixes have been introduced since November:
 - new bkgMixer to handle new time window for ECL and PXD
 - new bkg files for ECL and PXD
 - ECLMCMatching fixes (unstable particles)
- Current MC 4.5 based simulations does not include this fixes!
- Some flaws mentioned during physics session (wrong mcldx for bkg) are puzzling (same Sim and Reco (same DST) different way ntuple is dumped) → needs further study
- We still have to implement proper treatment for multi-particle relations and background
- The work is scheduled to be finished by March, for the moment I don't see any major showstopper

Summary

- MC-Matching issues have for unstable particles and/or daughter particles have been fixed
- I still need to implement proper treatment for multi-particle relations and background
- I think the easiest and most reliable way to do it is to use Hits EnergyDep to give weights to individual contributions
- Some coding conventions in ECLMCMatching seem to be outdated and need clean-up
- The work is scheduled to be finished by the end of the month, for the moment I don't see any major showstopper