

Fastsim update: EMC

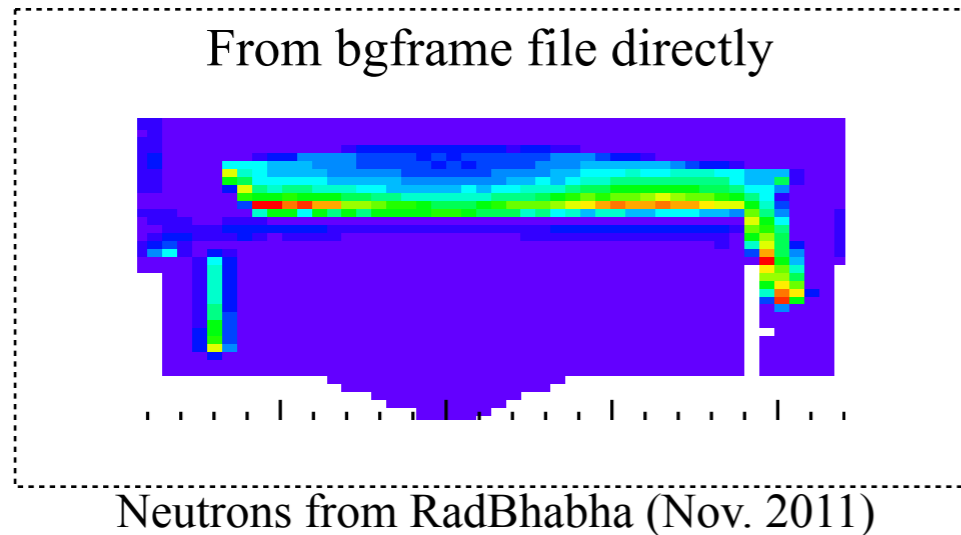
*Chih-hsiang Cheng
Caltech*

*SuperB Collaboration Meeting
Physics Tools
Frascati, Italy, 2012/03/20*

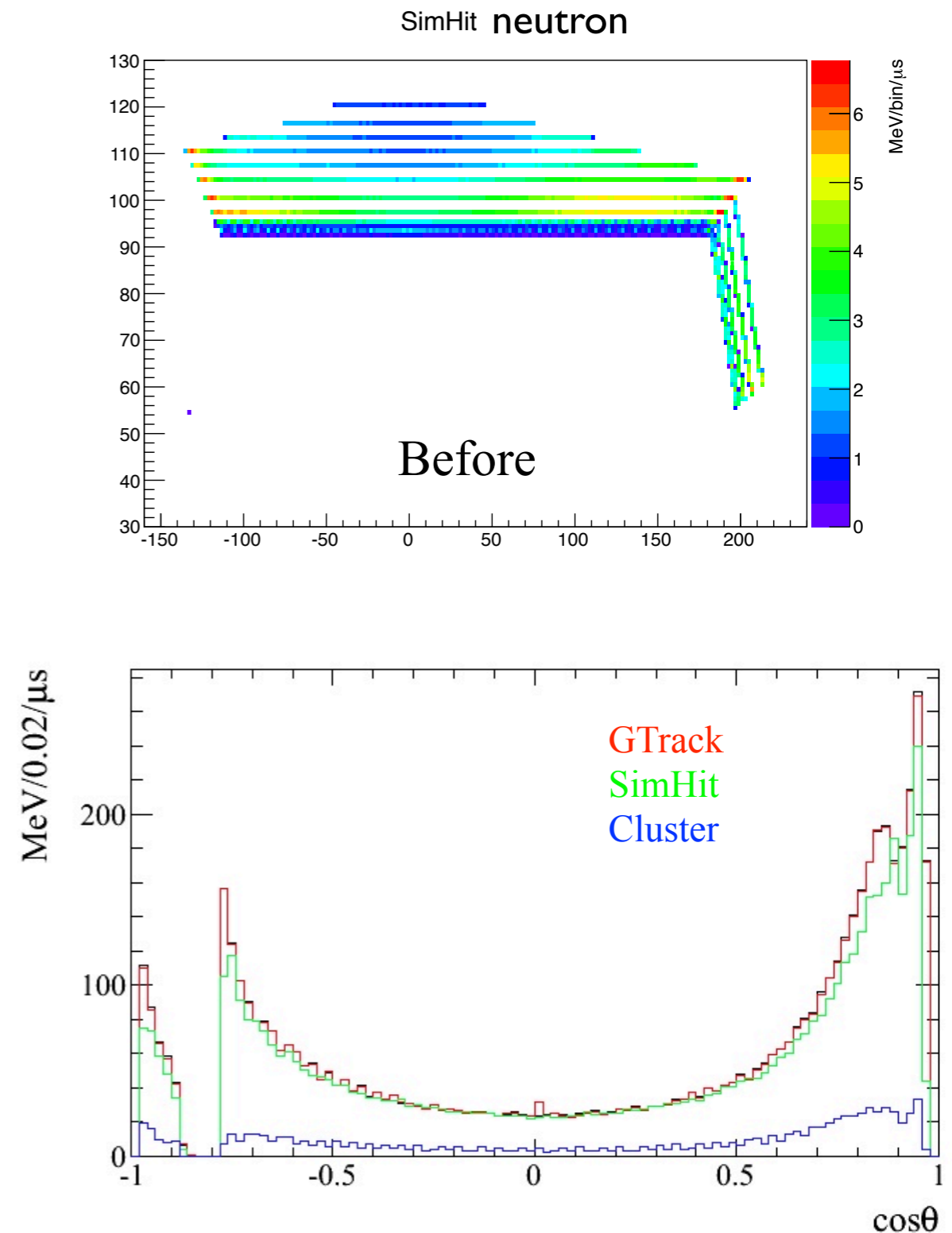
Improvement since last CM

- Bug fix for missing background clusters:
 - ▶ Backward EMC clusters were missing entirely.
 - ▶ A lot of background (especially neutrons) were lost in all regions.
- Speed improvement
 - ▶ New clustering algorithm (PacEmc)
 - ▶ TruthTools: BtaLoadMcCandidates improve speed
- Geometry improvement
 - ▶ Better match with full sim.
 - ▶ Fix voxels that cut off geometry.
 - ▶ Hybrid geometry (4 regions).
- Pulse shape look up tables, more realistic.

Missing backward background SimHit



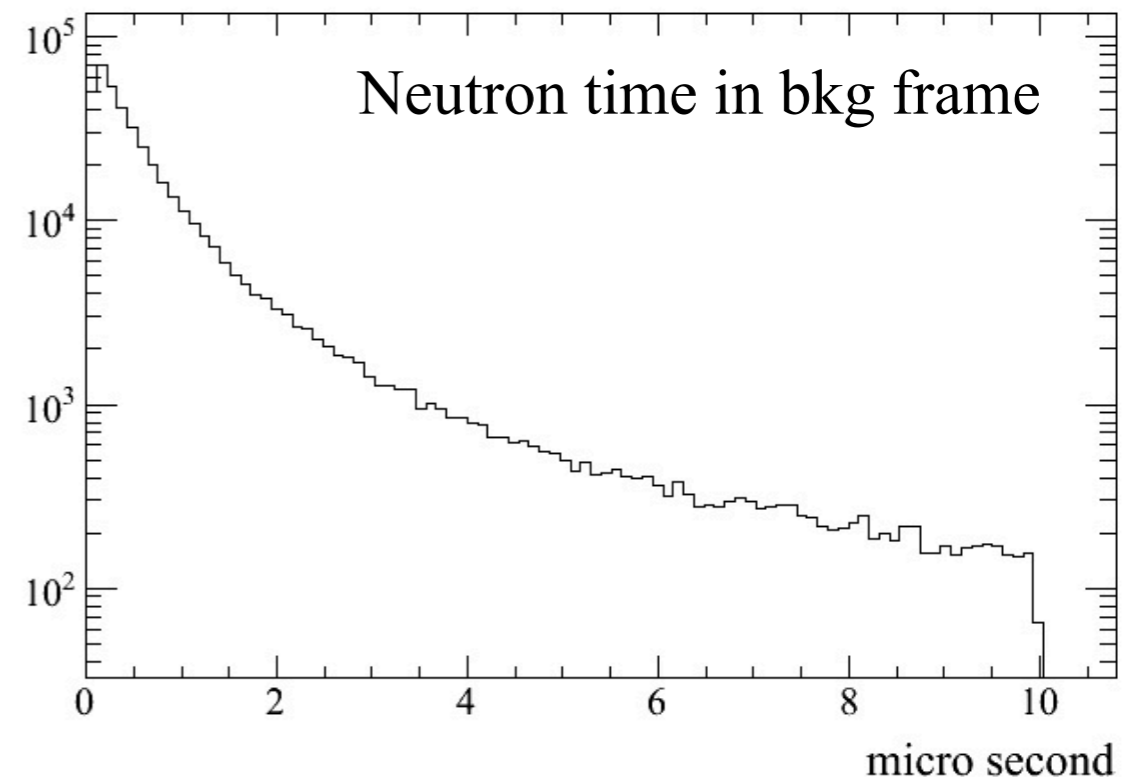
- Background SimHit at the backward EMC was completely missing. The culprit was a code number that identifies the EMC region in FullSim, in which the backward EMC uses a different code from barrel +forward.
 - ▶ Trivial fix in PacMC.
- However, only a small fraction of clusters were reconstructed (especially for neutrons).



Background particle timing

- When reading in background frame particles, we define a sensitive window, which is used to calculate the number of beam crossings to read in and starting time of bkg particles is randomized within this window.
- Each detector has its own sensitive window, using particle time (randomized starting time + travel time (FullSim+FastSim)).
- Neutron time has a long tail, falling out of the detector sensitive window.
- But tail from earlier beam crossings shouldn't be ignored.

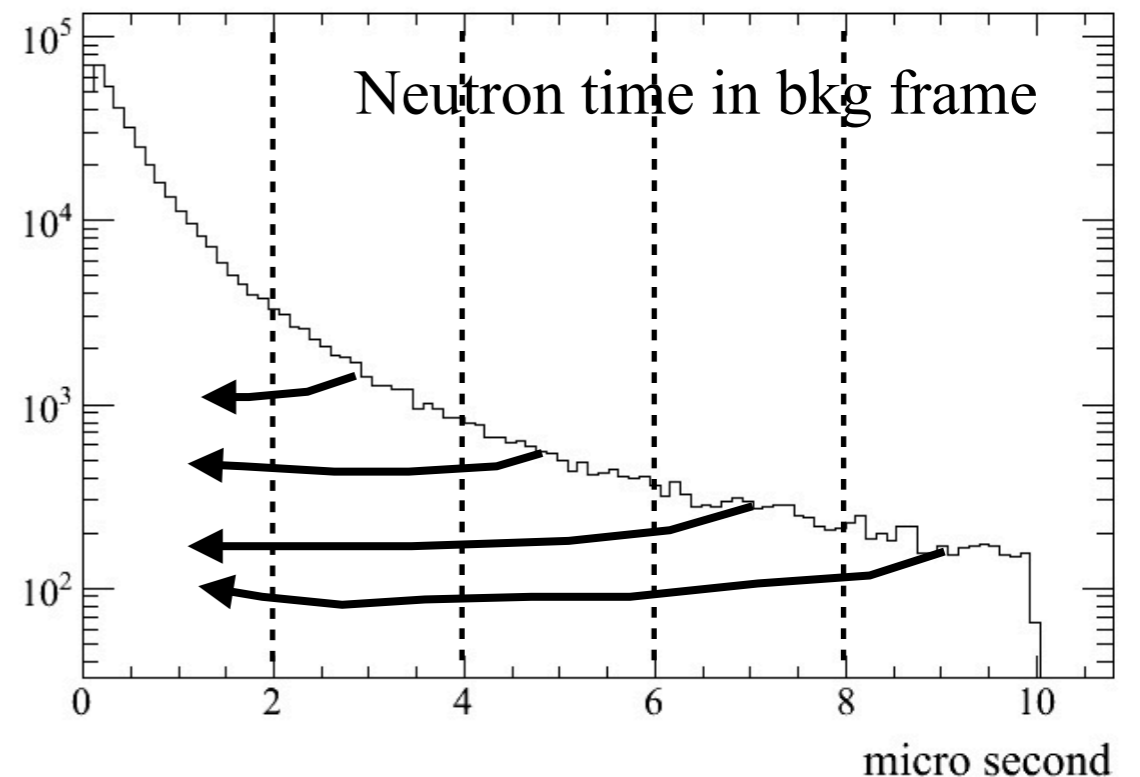
```
local(10:00)/V0.3.0 % grep -B 2 sensitiveWindow  
PacProduction/MixRadBhabhaBkg.tcl  
talkto PmcRadBhabhaBkgInput {  
  BackgroundType      set RadBhabha  
  sensitiveWindow     set 2.0e-6  
  sensitiveWindowStart set -1.75e-6
```



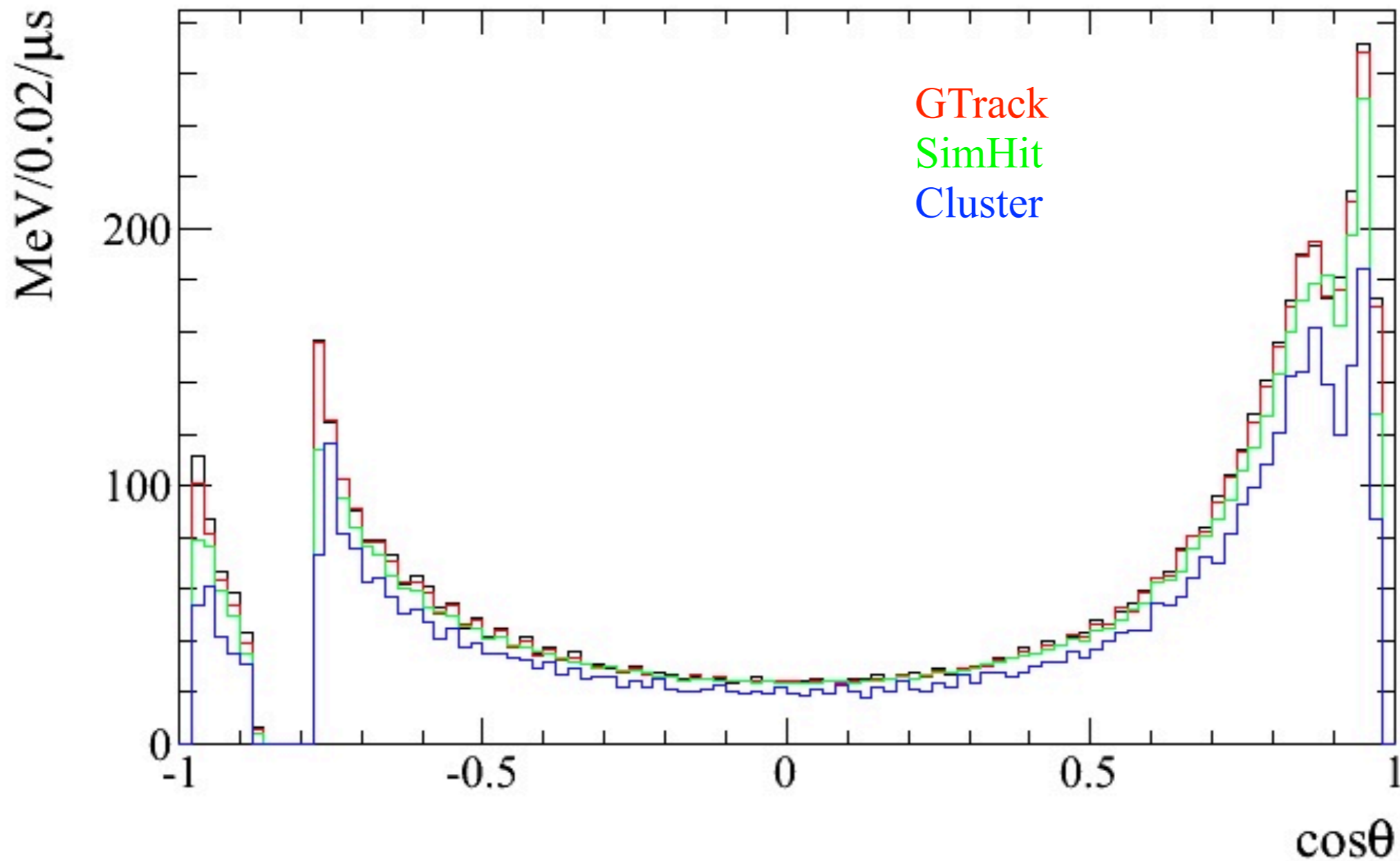
Background particle timing

- When reading in background frame particles, we define a sensitive window, which is used to calculate the number of beam crossings to read in and starting time of bkg particles is randomized within this window.
- Each detector has its own sensitive window, using particle time (randomized starting time + travel time (FullSim+FastSim)).
- Neutron time has a long tail, falling out of the detector sensitive window.
- But tail from earlier beam crossings shouldn't be ignored.
- Solution: fold the time back to the detector sensitive window.

```
local(10:00)/V0.3.0 % grep -B 2 sensitiveWindow
PacProduction/MixRadBhabhaBkg.tcl
talkto PmcRadBhabhaBkgInput {
  BackgroundType      set RadBhabha
  sensitiveWindow     set 2.0e-6
  sensitiveWindowStart set -1.75e-6
}
```



Most clusters are recovered



Slow down due to background

- We found out that, while running FastSim mixed with “normal” background level the speed is acceptable, running at 5x background is way too slow.
 - ▶ To increase background in FastSim, we increase bunch-crossing frequency.
- The time per event grows far worse than linearly as the background level increases.

Slow down due to background

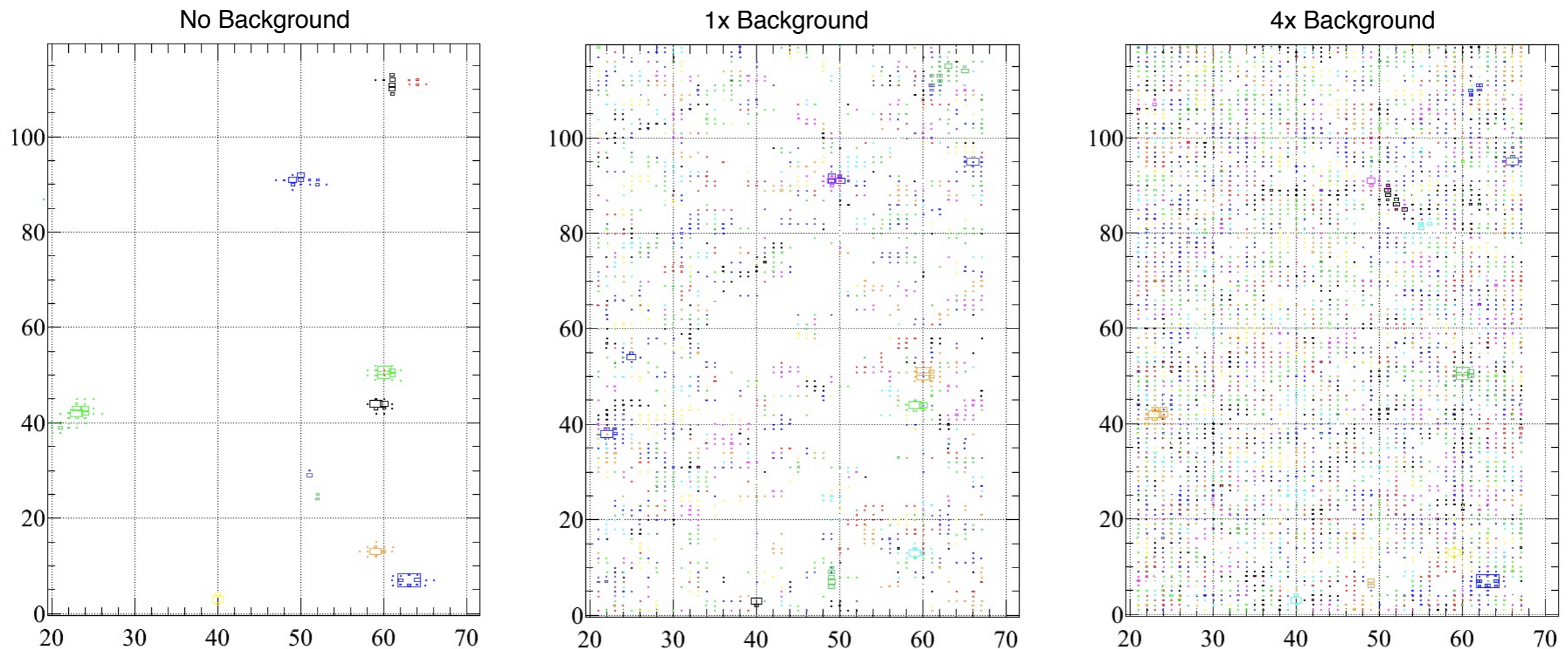
- We found out that, while running FastSim mixed with “normal” background level the speed is acceptable, running at 5x background is way too slow.
 - ▶ To increase background in FastSim, we increase bunch-crossing frequency.
- The time per event grows far worse than linearly as the background level increases.

CPU time per event (ms/event) consumed by modules (running 10 B0B0bar events)

Module	no bkg	1x	2x	3x	4x
PmcReconstruct	205	340	572	917	1191
PmcSimulate	60	176	290	522	625
BtaLoadMcCandidates	0	303	1671	8987	16147
PacCaloSplitMerge	1	190	3086	18178	54315
PmcRadBhabhaNeutronBkgInput	0	177	227	336	319
RacTestInput	6	80	158	317	366

Old algorithm

- Each SimTrack produces one cluster. It is then split to single bump clusters if necessary. All pairs of clusters are then tested to check if they can be merged to a single bump cluster.
- Pairing is $O(N^2)$. Other operations in each merging process make it worse than $O(N^2)$. It won't work if $N \sim$ hundreds.

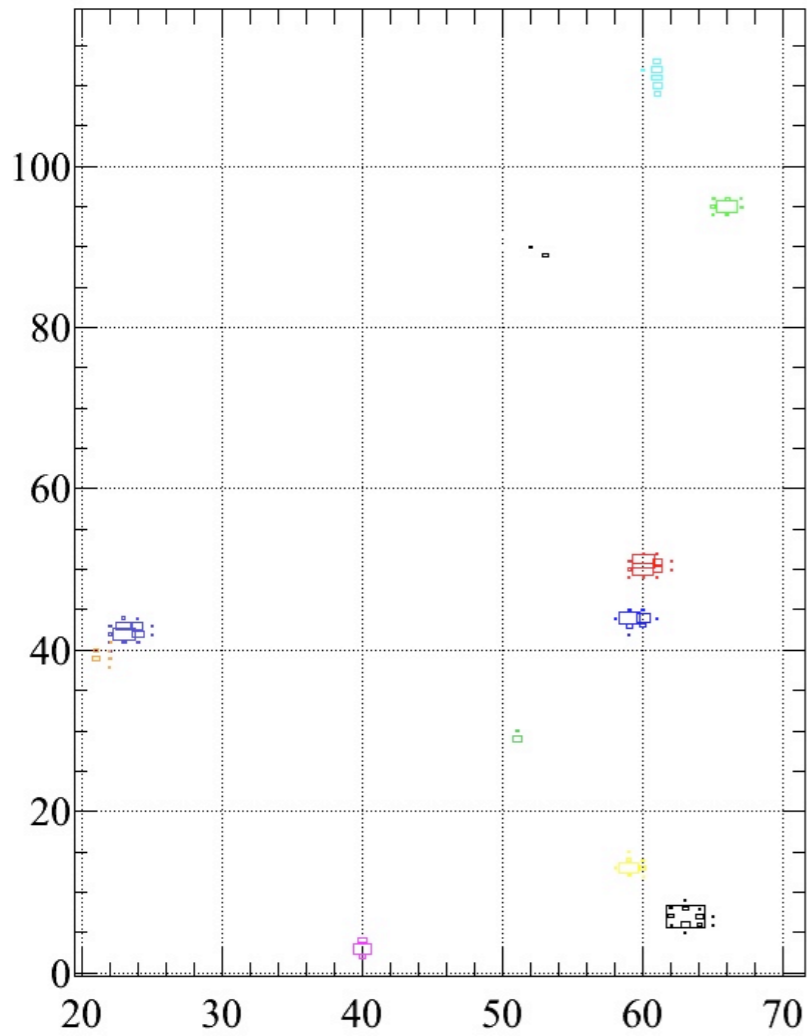


Improved algorithm

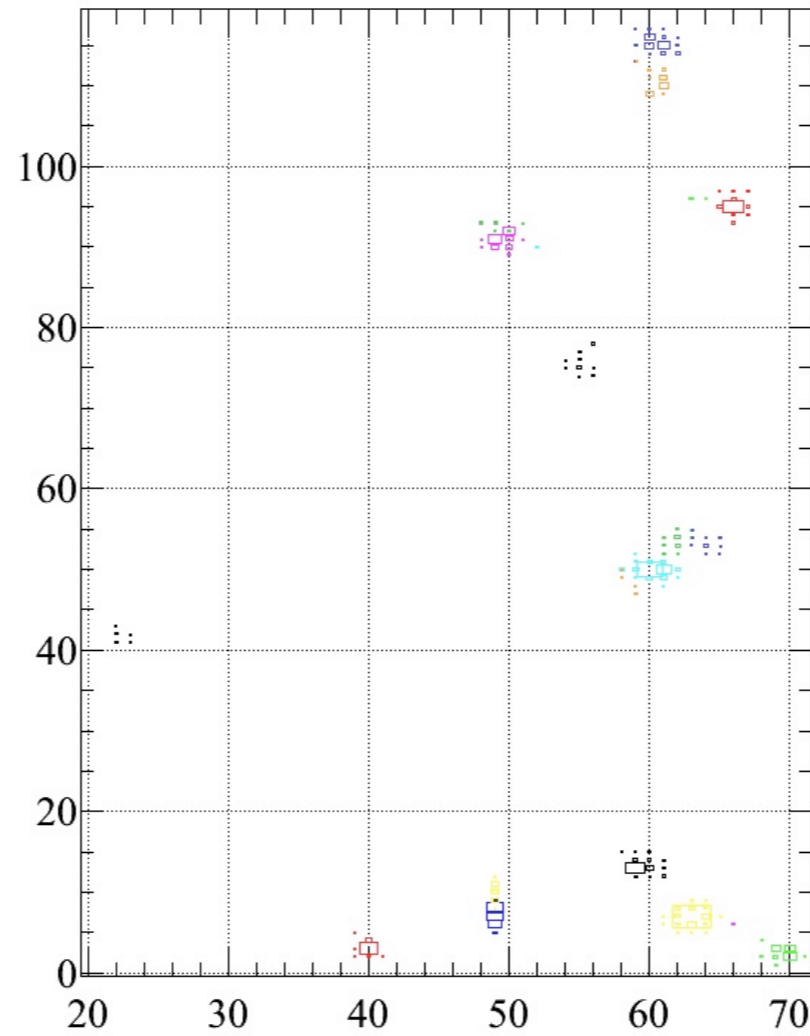
- Each SimTrack produces a cluster. Fill the energy in a map of crystals. Select seeds and build clusters, and then split them up to single bump clusters.
- One needs to stop growing clusters at some point.
- Although it may be necessary to keep every single low energy hit for background study, we want to remove as much noise as possible for physics while keeping the physics clusters as complete as possible.
- Algorithm (recursive):
 - ▶ Select seed crystals above 20 MeV; starting from the highest one.
 - ▶ Connecting adjacent crystals (once used, removed from the map).
 - ▶ If a crystal is below 5 MeV and none of its neighbors is above 10 MeV, stop (this crystal is not used).
 - ▶ Resulting clusters are then split into single bump clusters.

New clustering

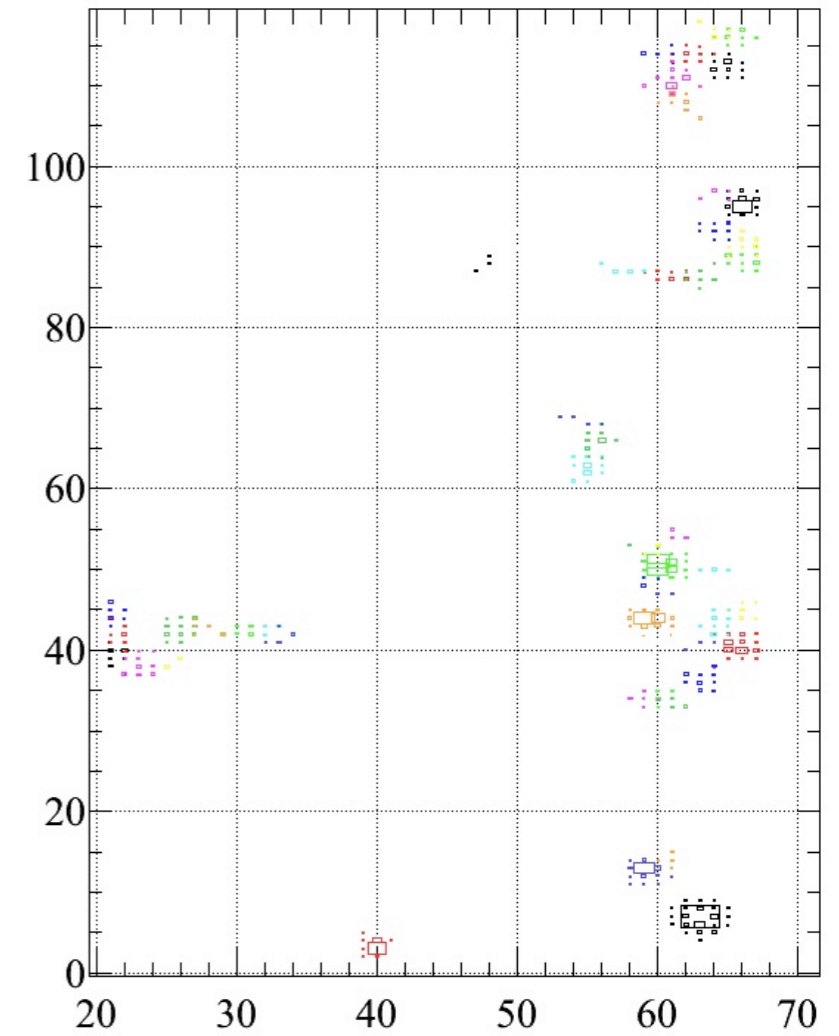
No Background



2x Background



5x Background



* Emc signal pulse shape uses a simple analytical function. More realistic shape from circuit simulation has a much larger tail; background will be much higher than shown here.

Improving BtaLoadMcCandidates

- BtaLoadMcCandidates builds MC BtaCandidates from GTracks, including genealogy.
- It is very inefficient in searching for daughters of each GTracks. It loops over the entire GTrack list in search for each given GTrack, including those from background, which don't have daughters (great majority of them).
- Simply checking whether the GTrack has any daughters before looping over the list saves a lot of time.
 - ▶ It's a one-line fix.

Improved speed

- Time consumption is greatly improved.

CPU time per event (ms/event) consumed by modules (running 10 B0B0bar events)

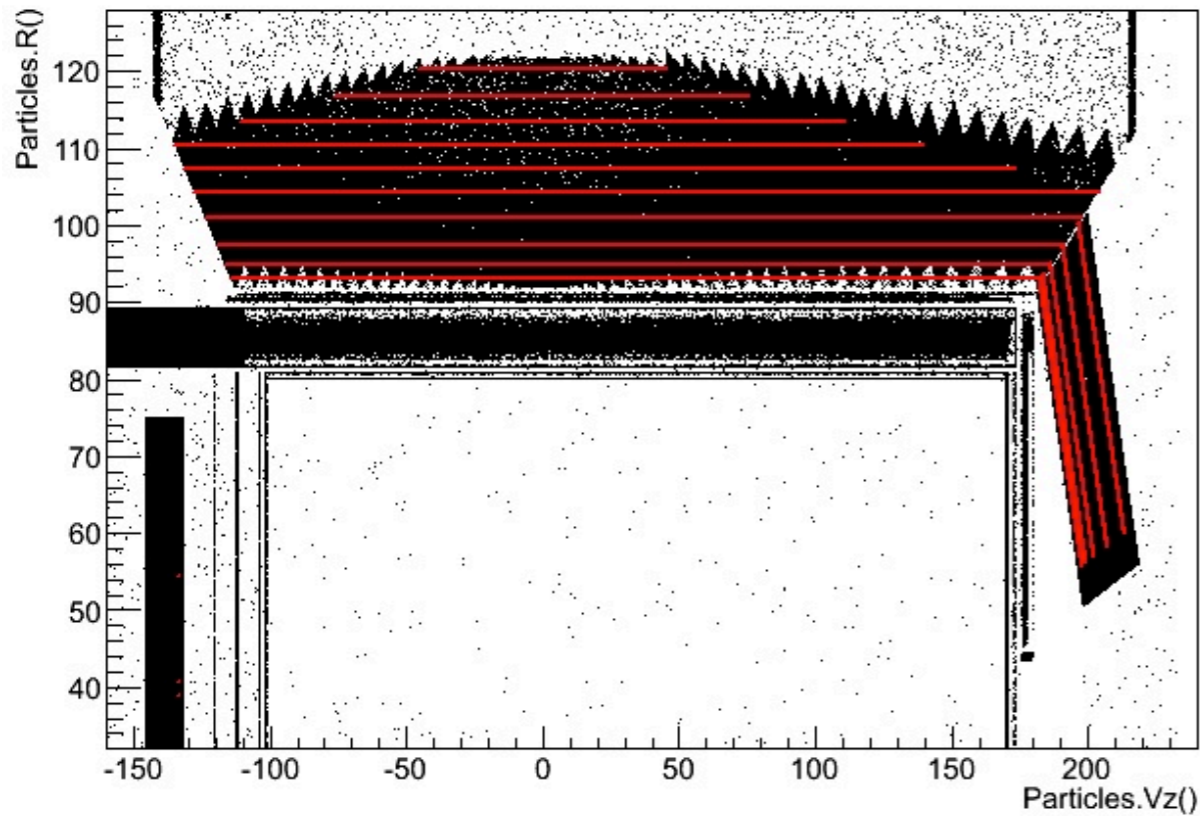
Module	no bkg	1x	2x	3x	4x
PmcReconstruct	279	355	705	1035	1442
PmcSimulate	82	192	376	550	696
BtaLoadMcCandidates	0	34	130	243	450
PacEmcReclustering	1	4	11	15	21
PacCaloSplitMerge	0	1	1	4	11
PmcRadBhabhaNeutronBkgInput	0	171	260	291	299
RacTestInput	9	73	190	299	393

Known issues

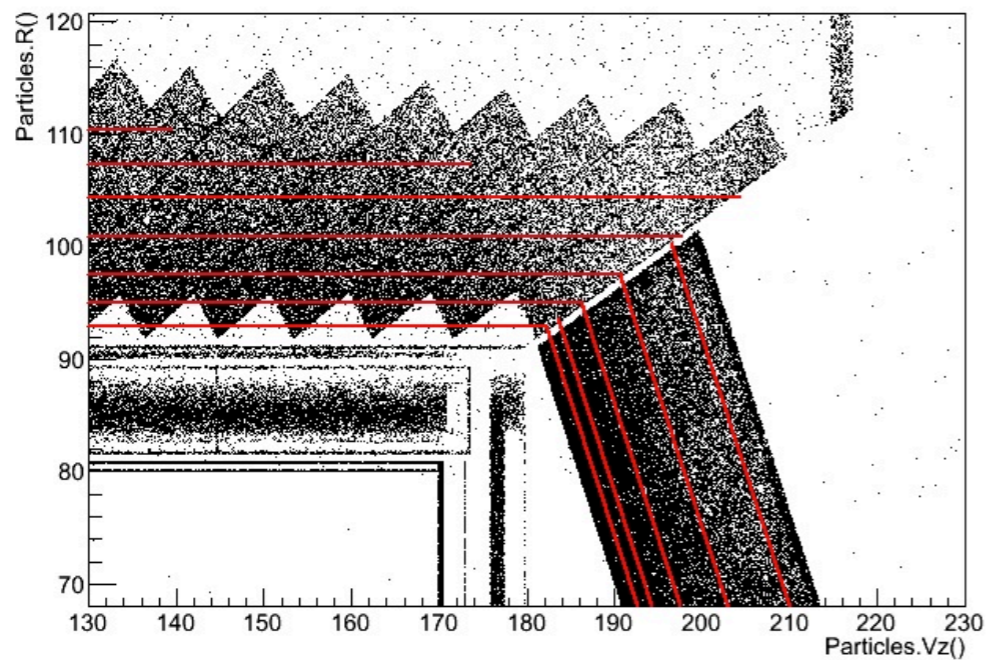
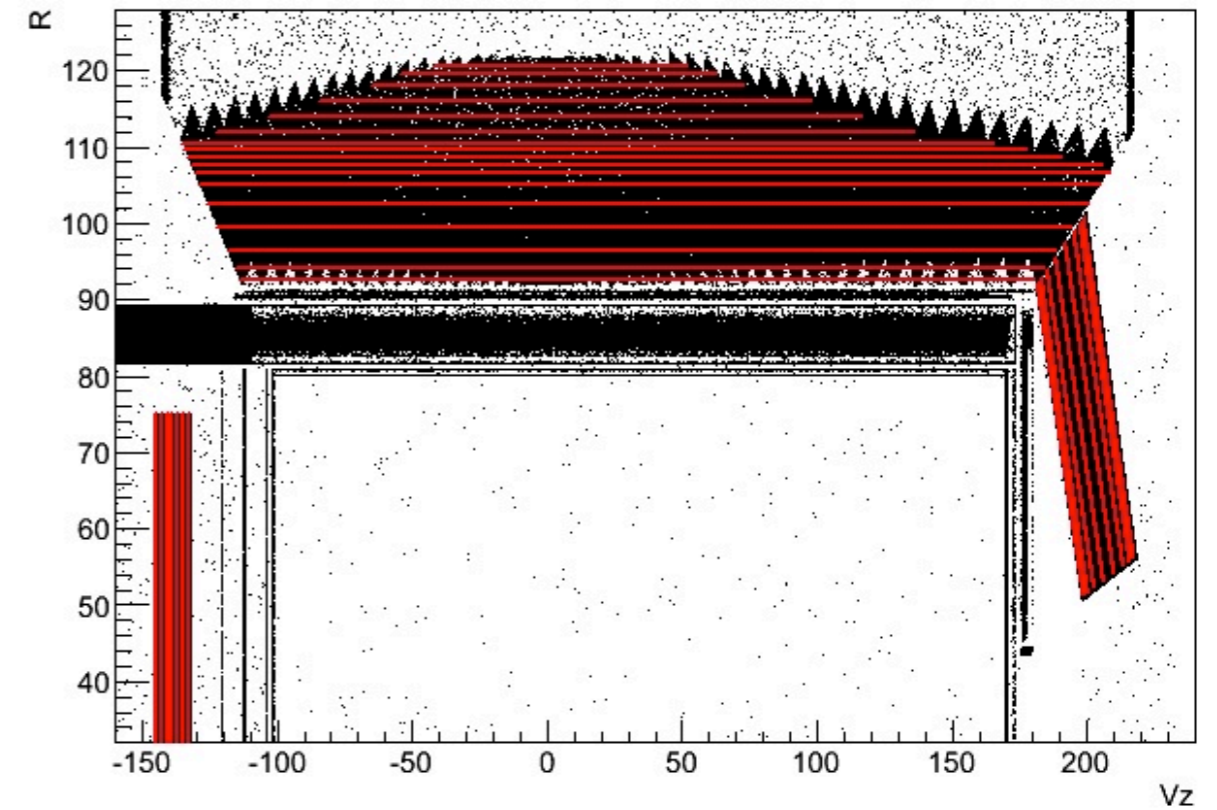
- In larger scale tests (thanks to Elisa and Daniel), there are still difficulties running with 5x background (e.g., seg-fault at semi-random events).
- Several memory leaks from PacEmc were fixed, but problems remain.
- Memory usage grows slowly but can reach more than 1GB.
- Valgrind shows large “possibly lost” from `PmcBkgInput::AddTCtoArray` (PacMC package).

Geometry, better match with FullSim

Old

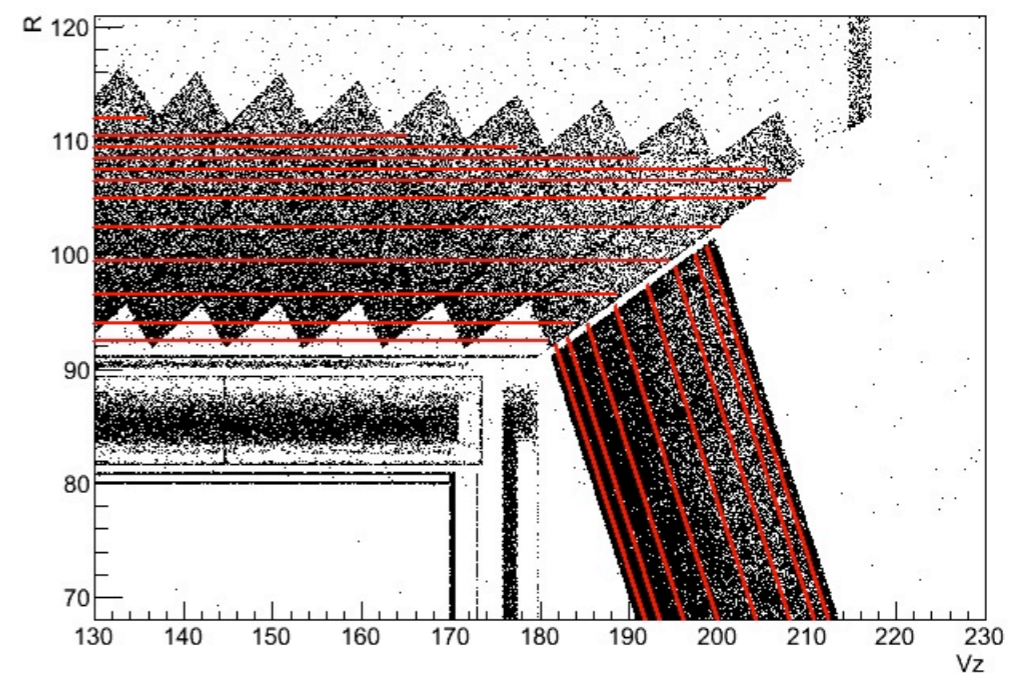


New



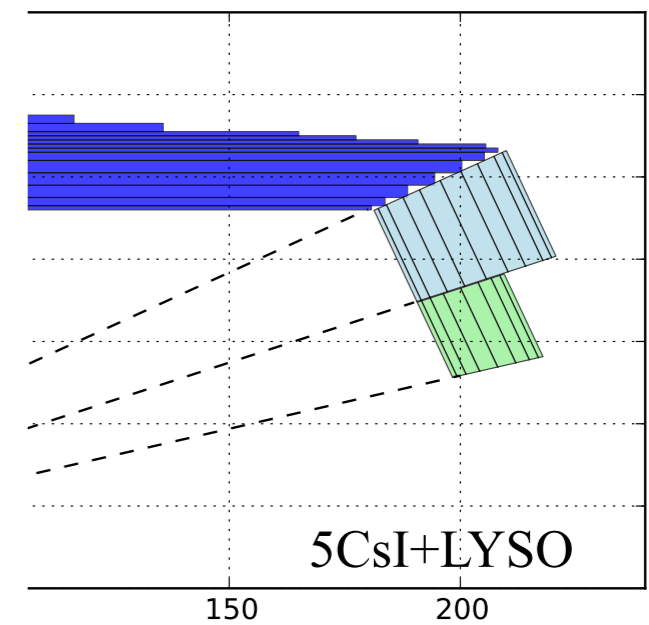
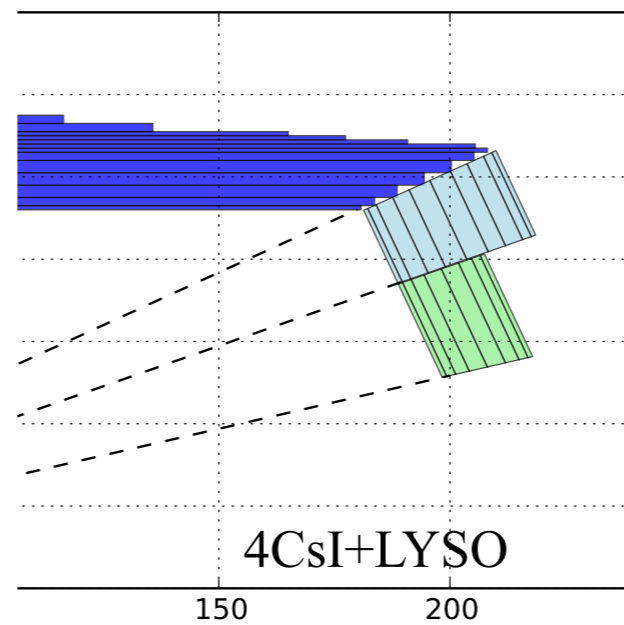
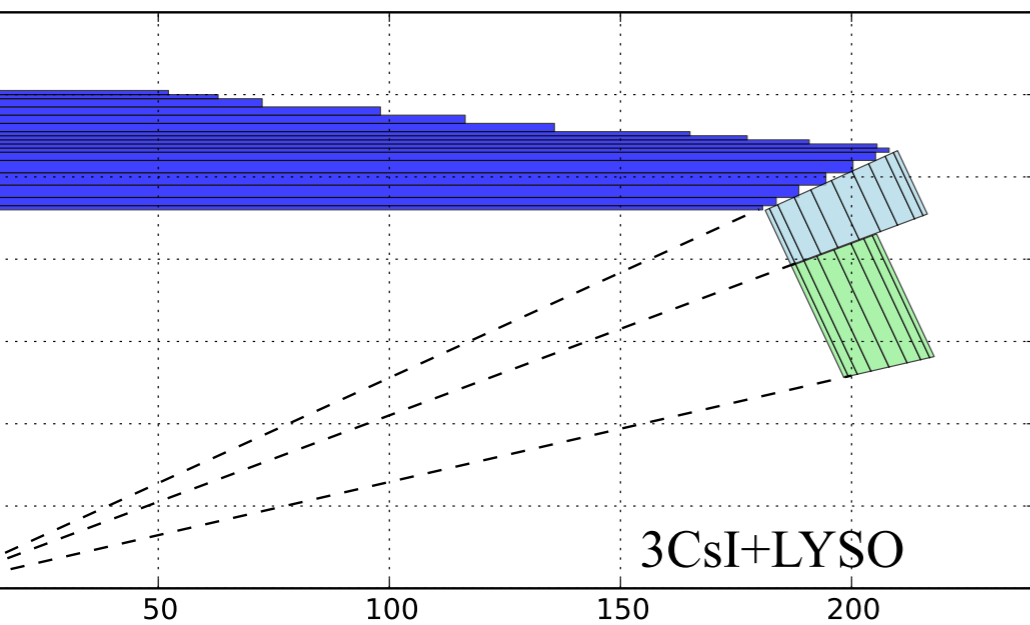
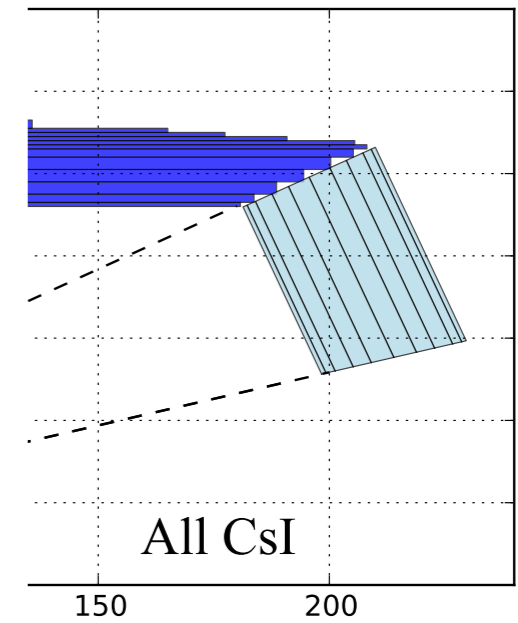
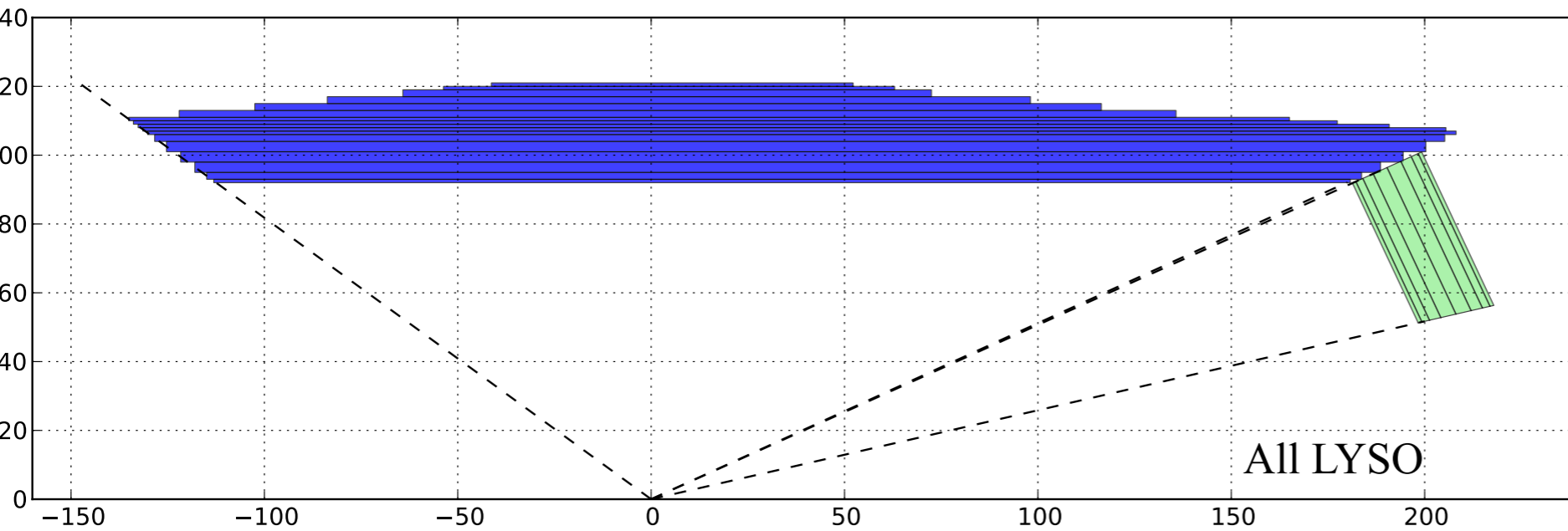
black:
bkgframe
neutron
interaction

red:
FastSim
SimHit



More flexible regions

- Motivation: study “hybrid forward endcap options”.



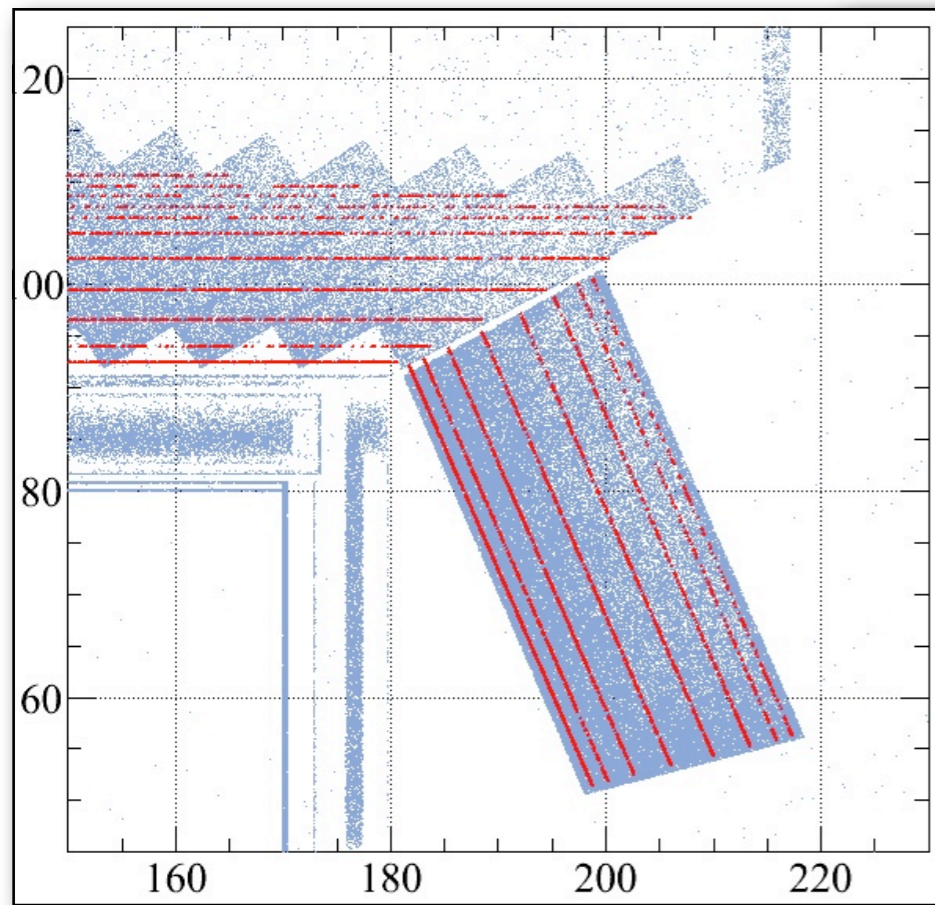
More flexible regions

- PacEmc assumed 3 regions in EMC: “forward cone”, “barrel cylinder”, “backward disk”, hard coded in C++.
- Now it allows 4 regions (code can be easily modified for more), and the geometry types and segmentations are specified in xml files.
 - ▶ It degenerates to fewer than 4 regions by setting nRings to zero for that region. See e.g., `PacEmc/EmcGeomOption_LYSO_NewStructure_SuperB.xml`
- Setting different structure in `PacEmc/PacEmcGeom_SuperB.xml`

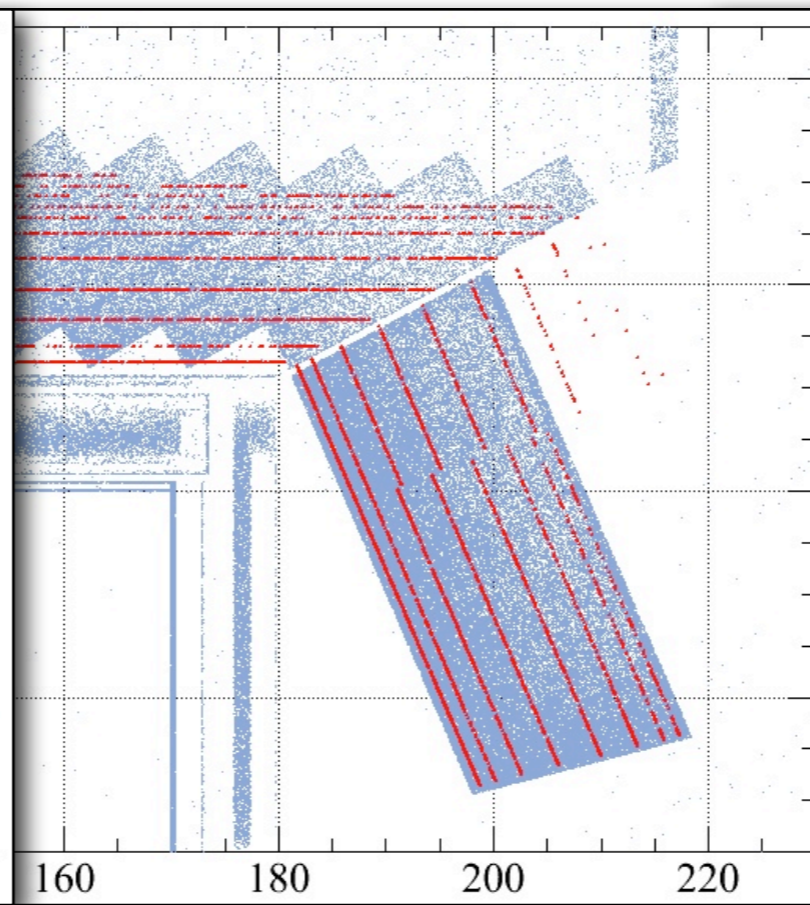
```
<!-- Geometry naming and segmentation-->
<include file="PacEmc/EmcGeomOption_LYSO_NewStructure_SuperB.xml" />
<!--include file="PacEmc/EmcGeomOption_LYSO_OldStructure_SuperB.xml" /-->
<!--include file="PacEmc/EmcGeomOption_LYSOCsI3_SuperB.xml" /-->

<!-- Other parameters -->
<include file="PacEmc/PacEmcGeomDetail_SuperB.xml" />
```

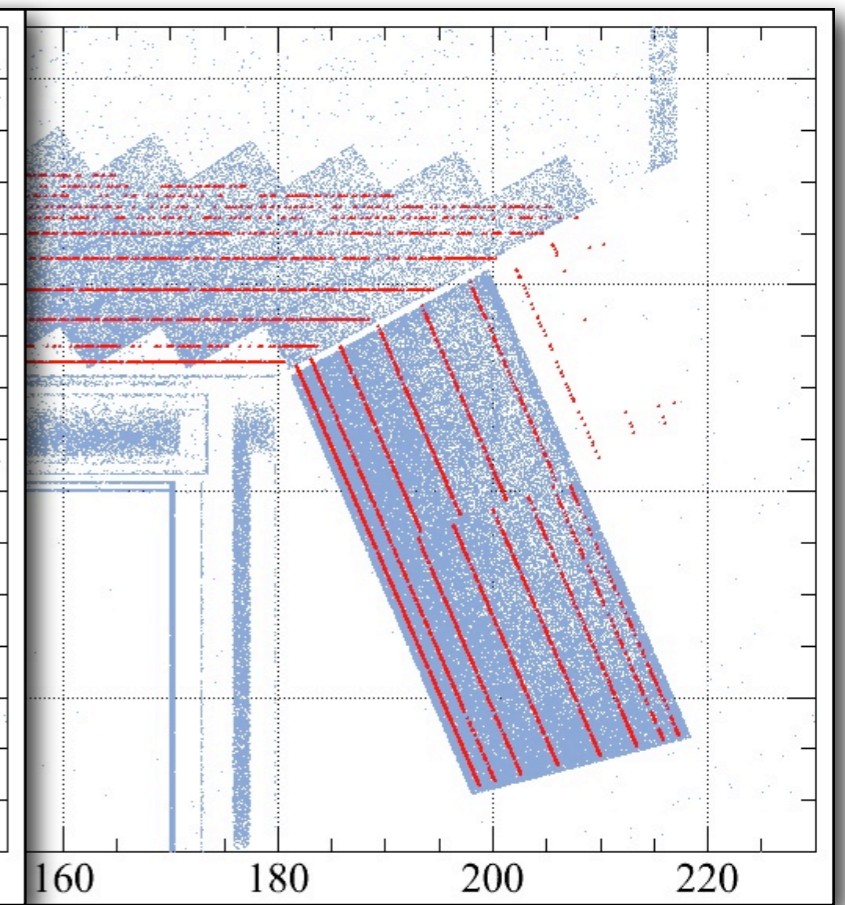
All LYSO



3CsI+LYSO



4CsI+LYSO

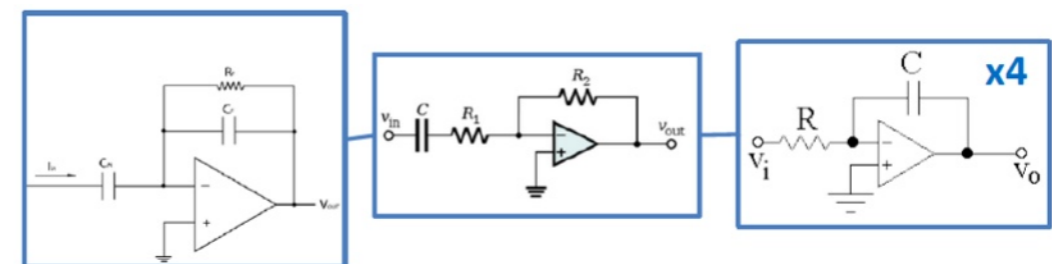
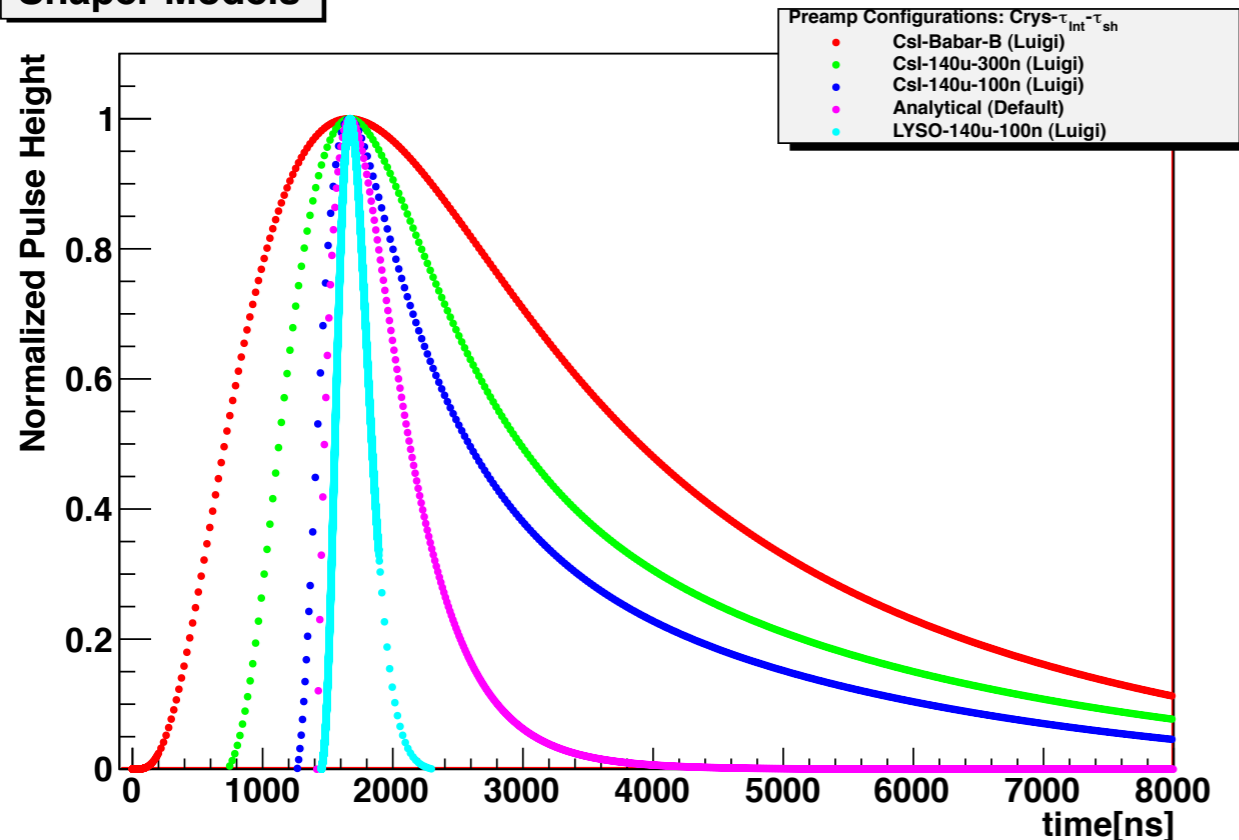


Pulse shape model

- Signal pulse shape was modeled with a simple CR-RC filter.
- Now uses more realistic numerical simulation for a more complicated circuit
- E.g., in PacEmc/PacEmcGeomDetail_SuperB.xml

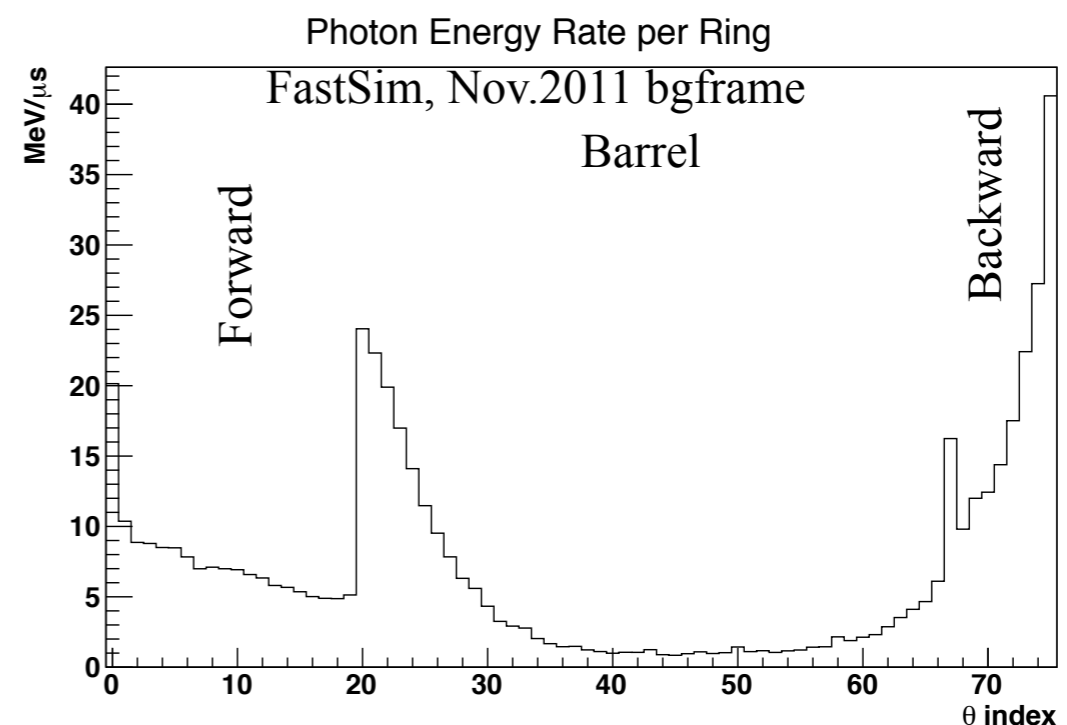
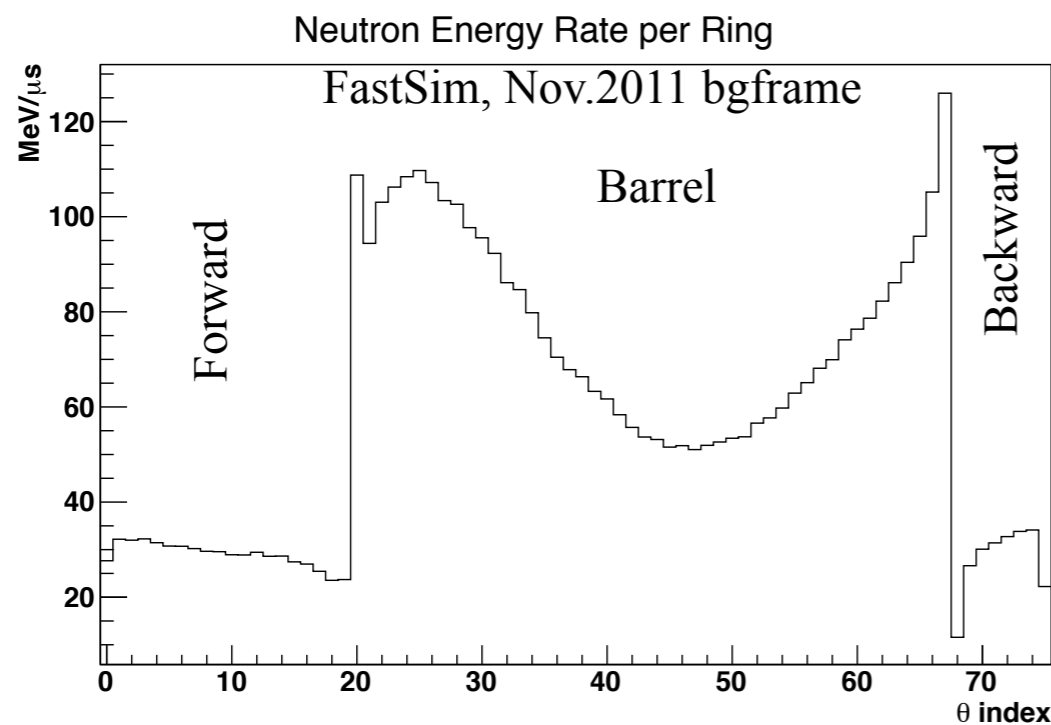
```
<param name="lookupTable" type="vector_string">PacEmc/preamp-models/  
LYSO-140u-100n-Luigi.txt, PacEmc/preamp-models/CsI-140u-300n-Luigi.txt,  
PacEmc/preamp-models/CsI-140u-300n-Luigi.txt, </param>
```

Shaper Models



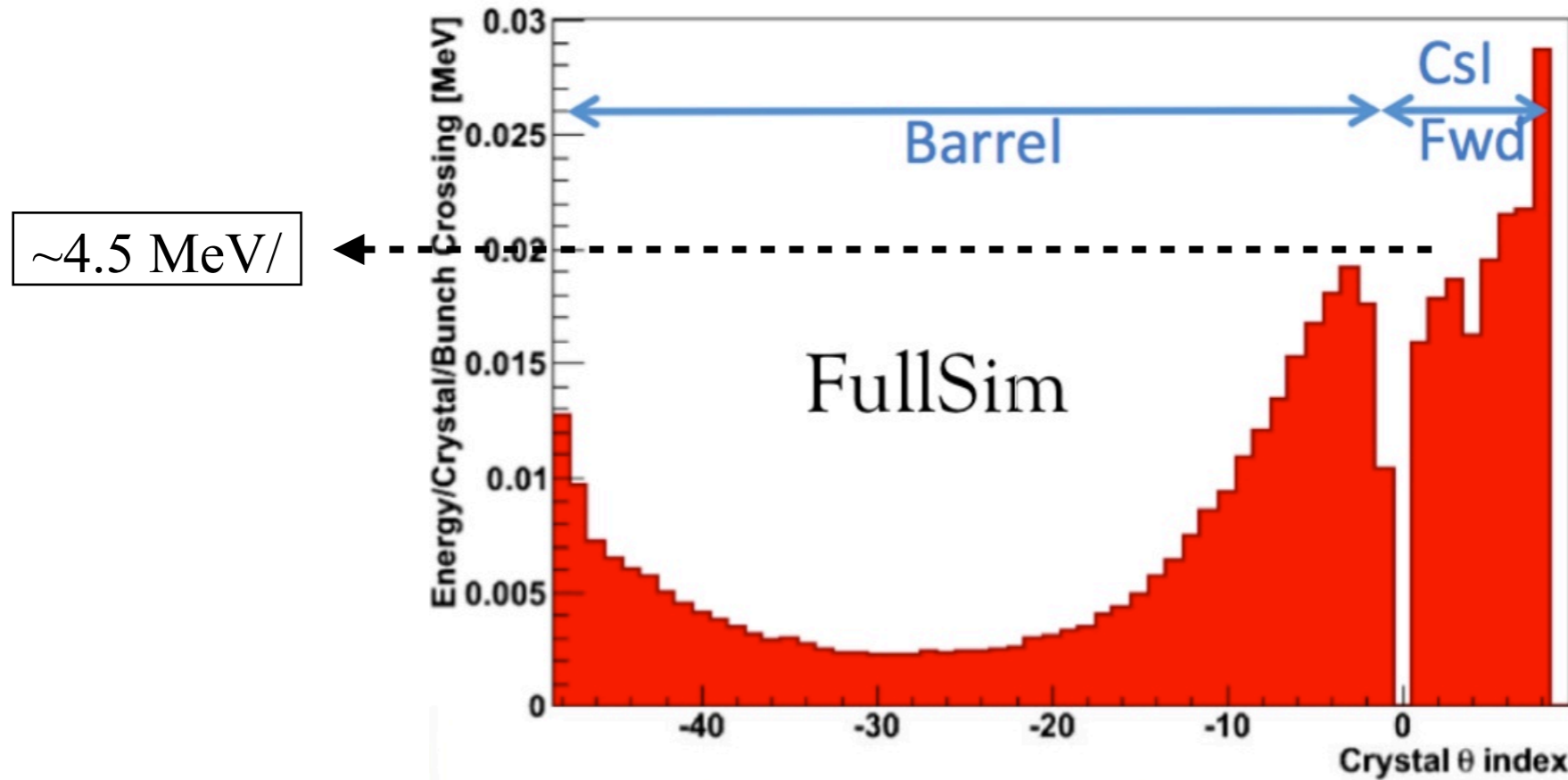
Emc energy rate from background frame

- From *Radiative Bhabha* background frames (November 2011).
 - ▶ Separate photon and neutron contributions.
 - ▶ **Photon** (neutron) energy cut off = **8 MeV** (a few keV).
- Use FastSim to record incident particle energy distribution in EMC space, as well as energy distribution after showering/clustering.



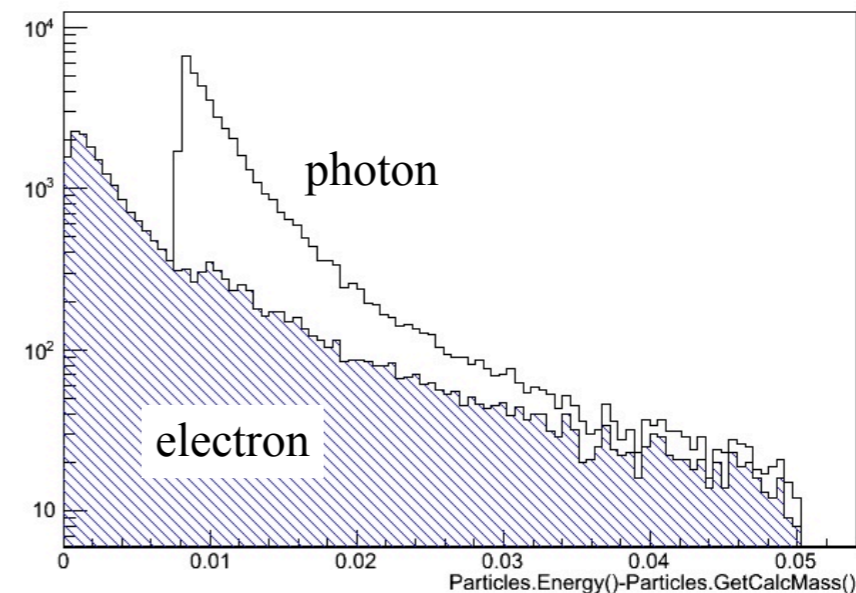
~ 1 MeV/ μ s/crystal at forward/backward ends of the barrel. *A factor of 4-5 lower than FullSim.*

4-5x lower than FullSim



* indexing system is different (reverse order)

Probably due to the 8 MeV γ energy cutoff in bgframe production.
Need new bgframe production to verify.



Summary

- Bug fix in background particle input module and introduction of background particle time folding to account for all background particles.
- Speed improvement with new clustering algorithm, and a trivial fix in BtaLoadMcCandidate.
- Difficulty running at high background remains. One killer is probably the memory leak in PmcBkgInput.
- Improve geometry, and allow more regions (hybrid forward endcap)
- More realistic pulse model.
- Need background frame production with a lower photon energy threshold.