Event finding in GEM Tracker

Radoslaw Karabowicz

Introduction

- This is a continuation of my presentation from PANDA Collaboration Meeting on 26th June 2013: “GEM Tracker Status” with following:

Conclusions:

- Event-based reconstruction not enough in the time-based reality
- Some changes applied to reconstruction chain
- Improved results:
  - track finding ‘efficiency’: 57.79% increased to 87.31%
  - event reco ‘efficiency’: 80.55% increased to 95.20%
- Further improvements necessary

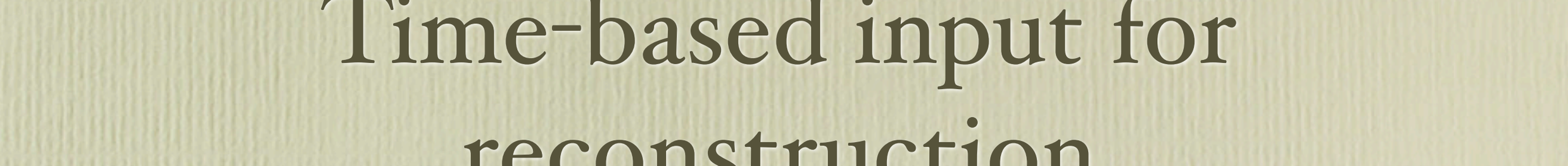
- Short summary of this talks follows:

Time-based input for reconstruction

Event-based data TTree

Ev.1 data | Ev.2 data | Ev.3 data | Ev.4 data | Ev.5 data

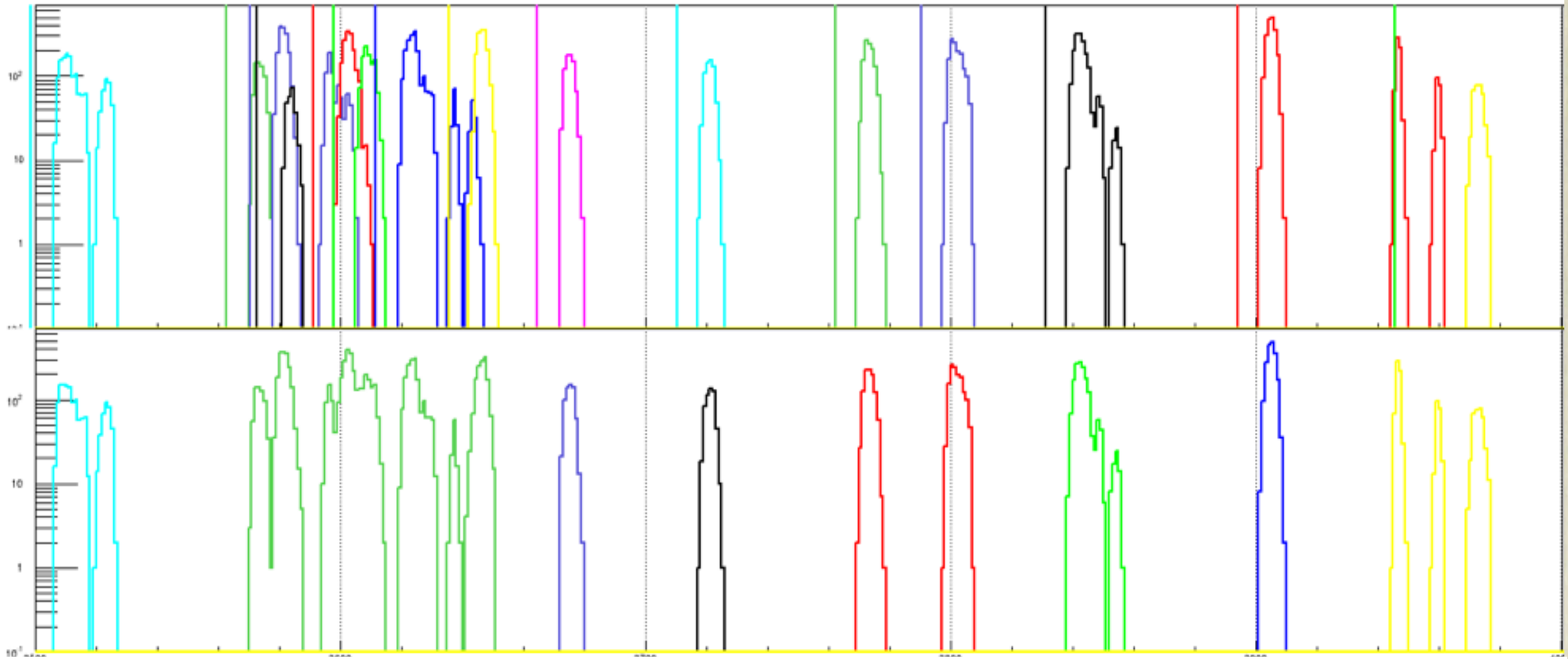
Time ordered data in FairWriteoutBuffer



Input time slices for reconstruction

GEM data's time distribution

Data yield from different events presented in different colors, vertical lines represent the begins of events.



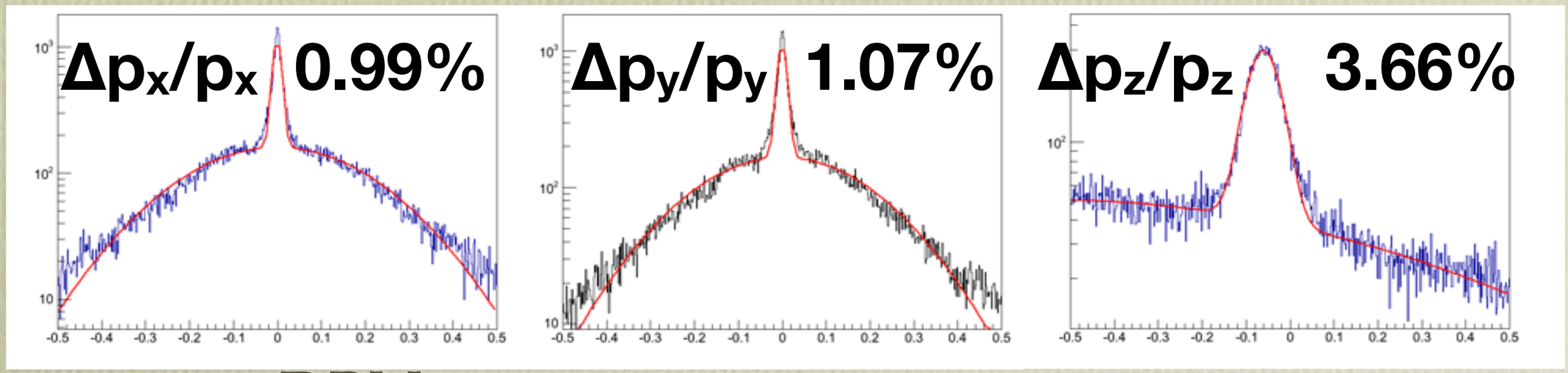
Data yield from different time slices presented in different colors (slicing by TimeGap(20ns)).

Time-based reconstruction

- Changes to code:
- very few changes to cluster finder
- hit finder: find hits on front/back pad plane as before, REQUIRE confirmation on back/front pad plane (check if relevant strips were activated in the previous 100ns - it requires storing information from previous time slices, achieved by PndGemMonitor)
- track finder: use these confirmed hits for tracking

Track finder results TB - momentum resolution

- Fitted function: sum of two gaussians
- Resolution: sigma of the thin gaussian

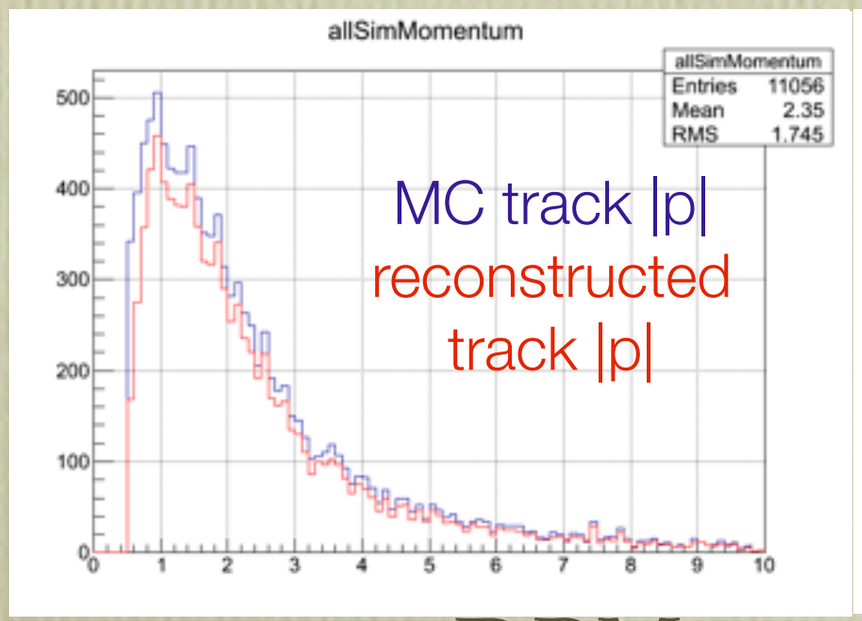


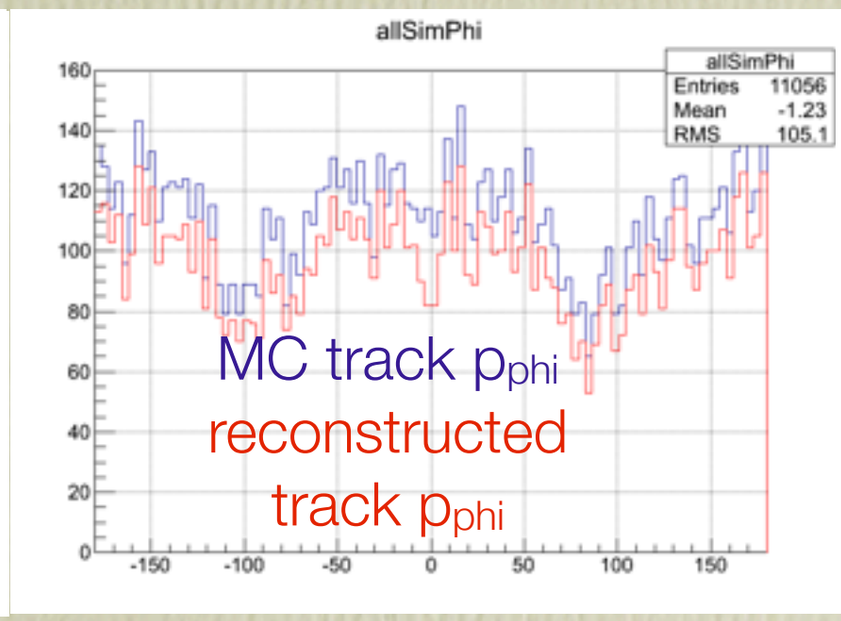
10000 DPM events

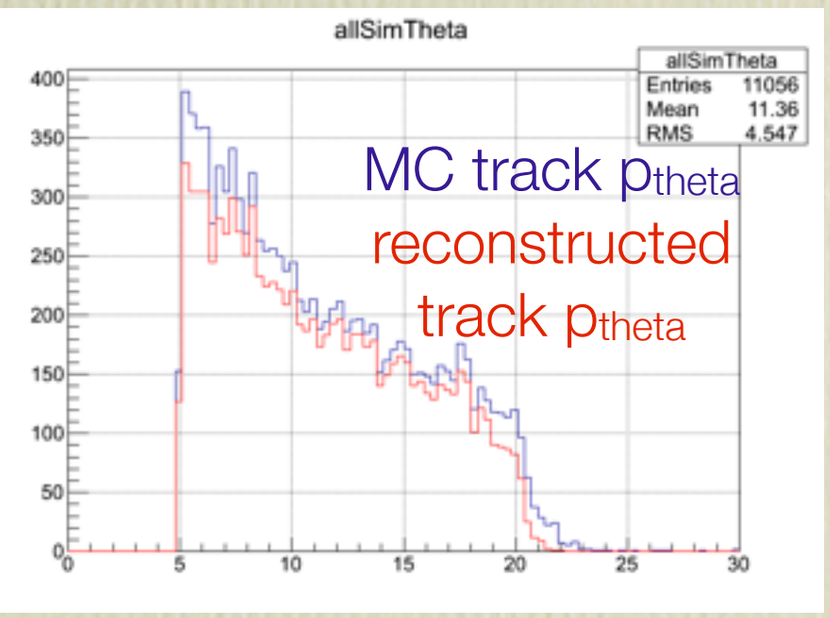
Track finder results TB - track finding efficiency

87% for primaries with |p|>1GeV/c,

compared to ~95% in event-based reconstruction, improved from 58%, when EB code used in TB mode.





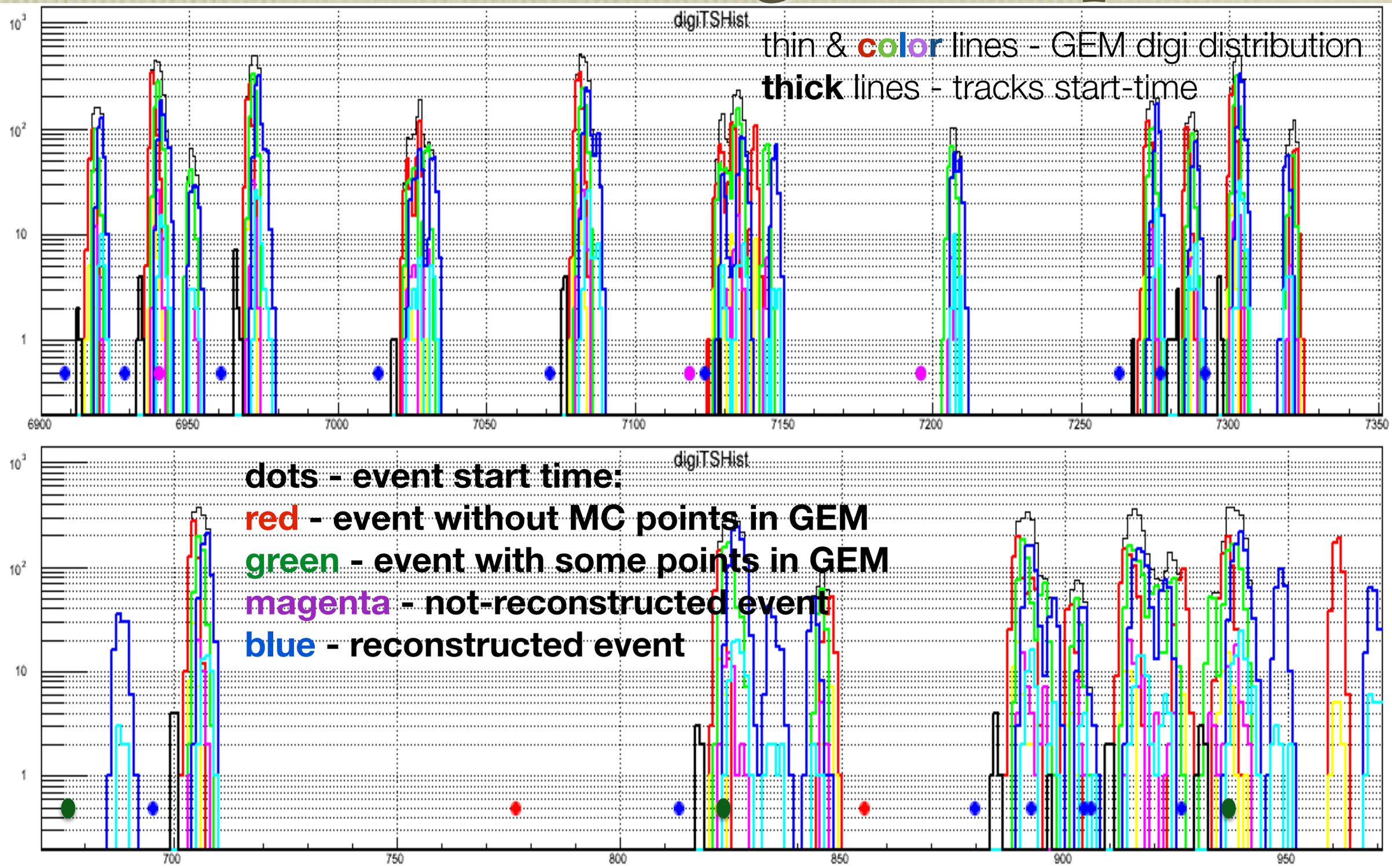


10000 DPM events

Event finding

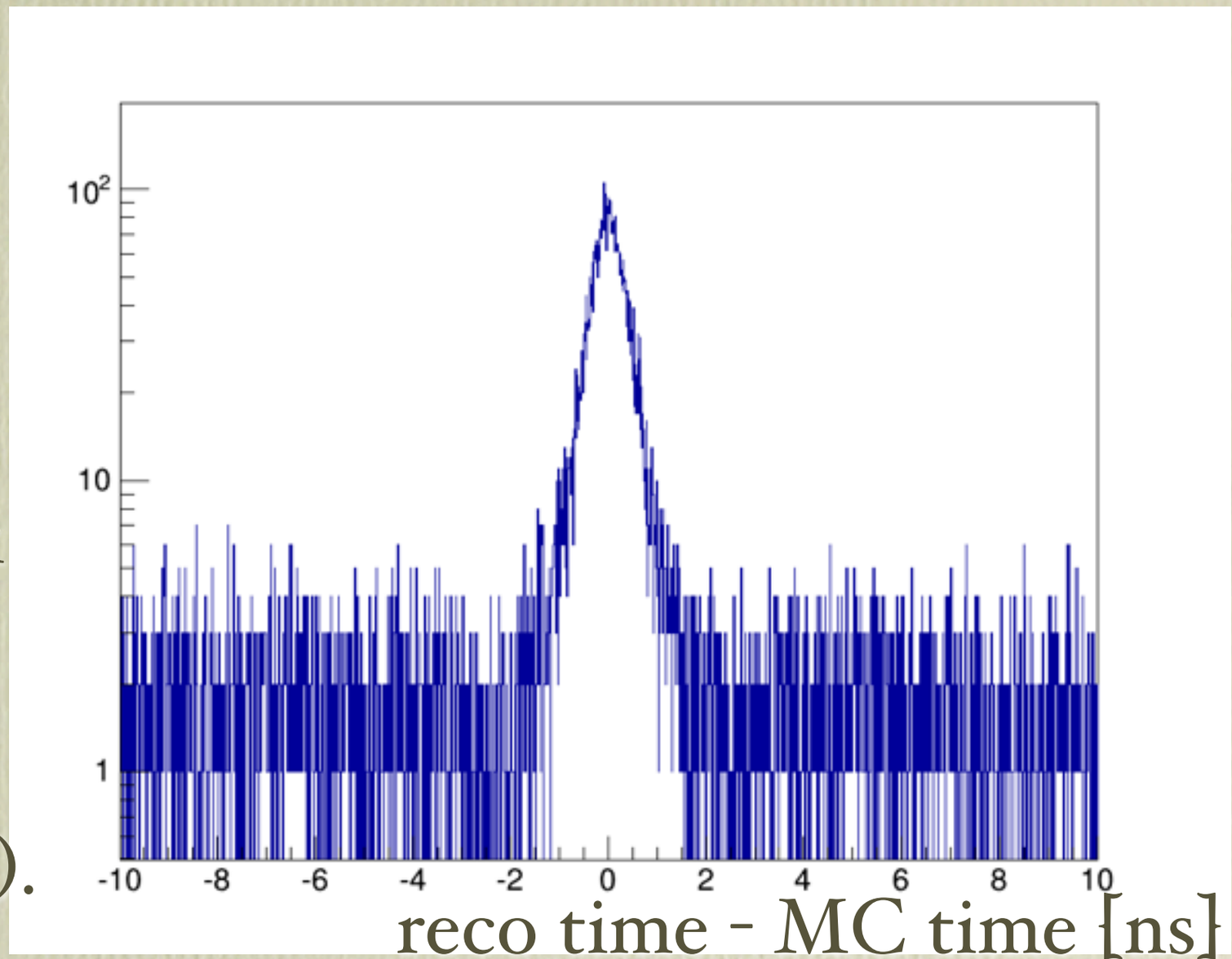
- Use the tracks' start-time
- Tracks with start-time closer than 3 ns end up in one event (3ns come from the tracks start-times correlation), with event time set to a mean of constituent tracks' start times
- Even one track can form event
- Compare reconstructed event times with MC event times, if the difference is smaller than 5 ns the MC event is considered to be reconstructed

Event building - example

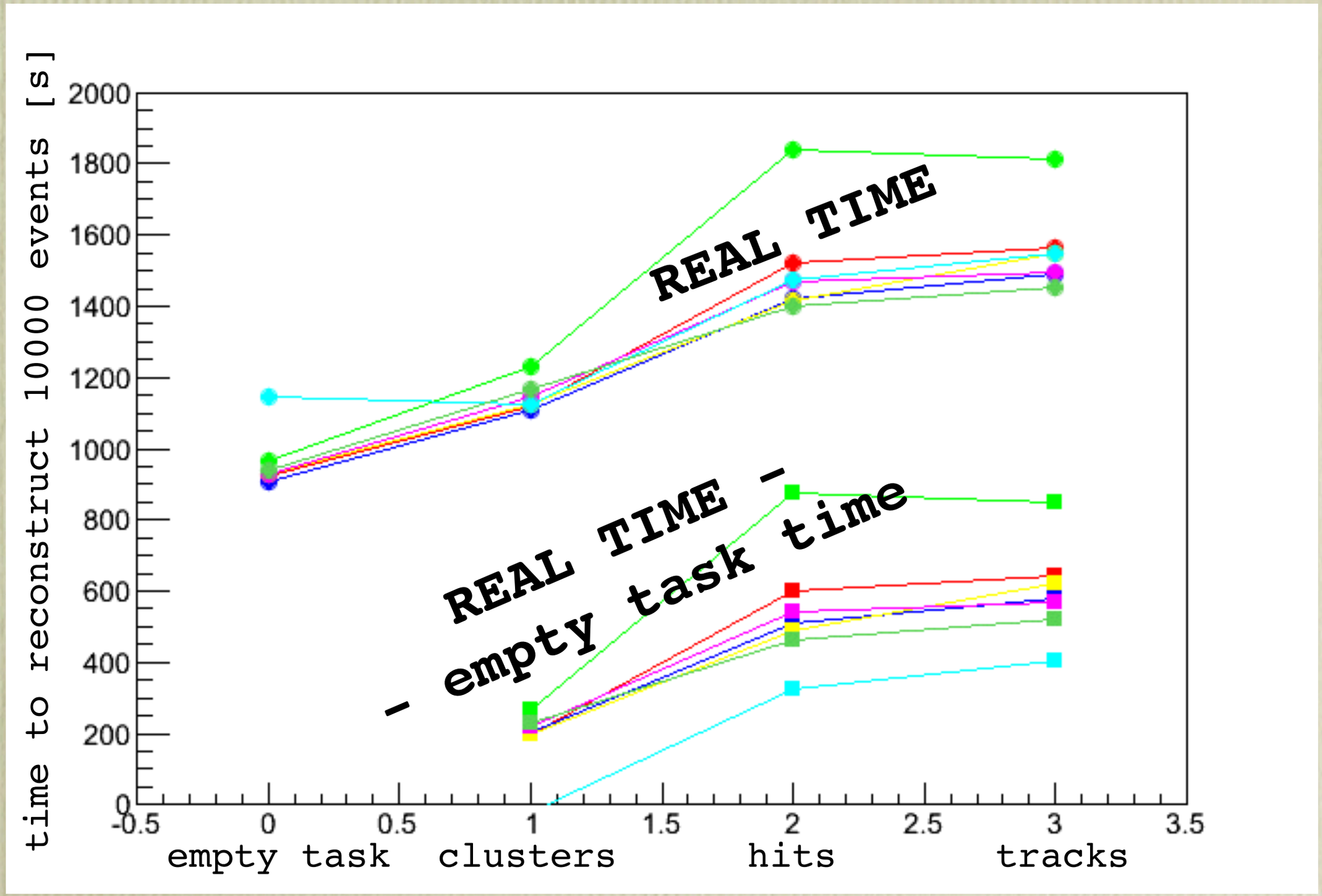


Results - event finding

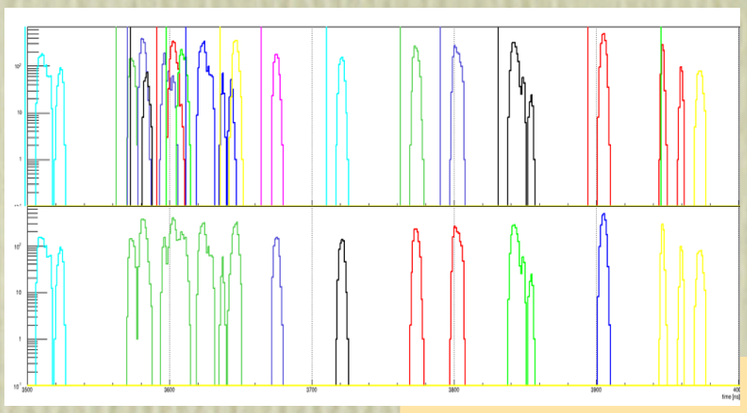
- 10000 DPM events simulated.
- 8165 events with reconstructable track in GEM tracker.
- 7536 events reconstructed (92.3%).
- 139 ghost events (1.8%).



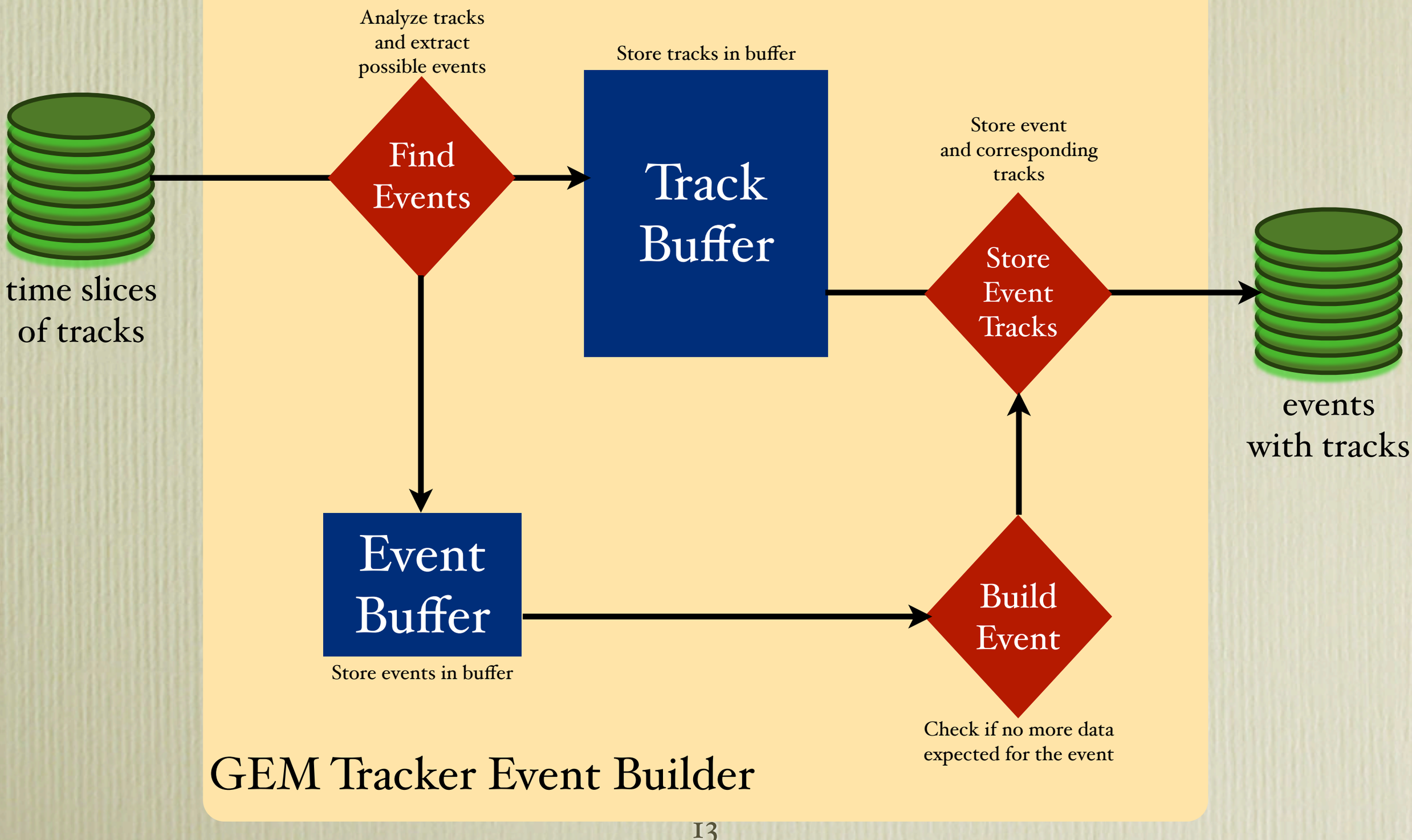
Reconstruction speed



Event building



Event building



GEM Tracker Event Builder

Generalize the idea

- Several different ways to extract event start time possible
- Some of these tasks will not need to store any data
- Some tasks will only be storing data to already-found events (they are not able to reconstruct events)
- Few tasks will require event start time (t0) to reconstruct data

Example: GEMEventBuilder

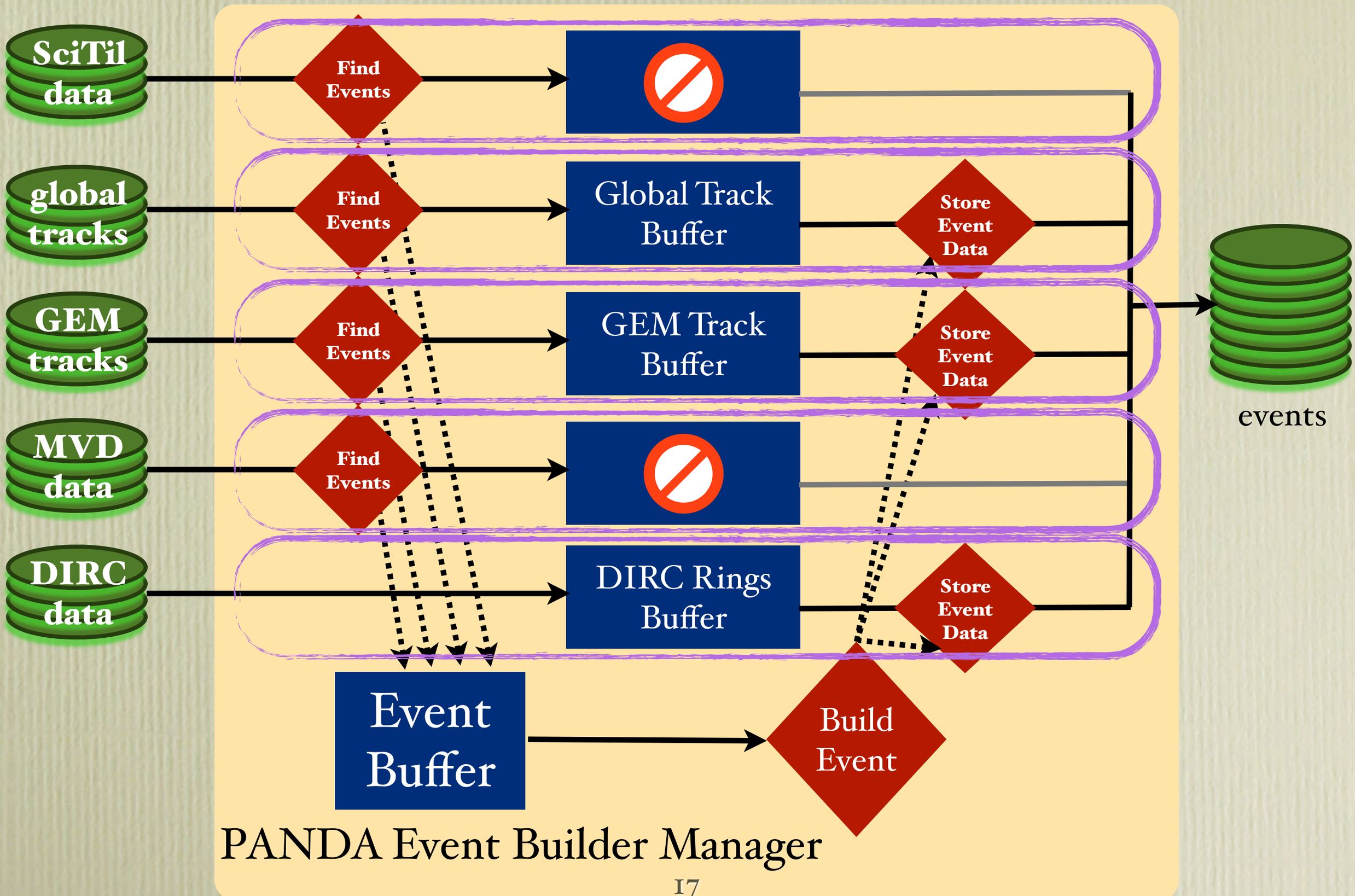
- vector<FairRecoEventHeader*> FindEvents()
  this function looks into input data (GEMTracks) and tries to find possible events. The data is stored in the internal buffer (thanks to derivation from FairWriteoutBuffer), found events are return;ed.

- void StoreEventData(FairRecoEventHeader* recoEvent)
  the function looks into data in the buffer and writes the data corresponding to recoEvent to the output TClonesArray.

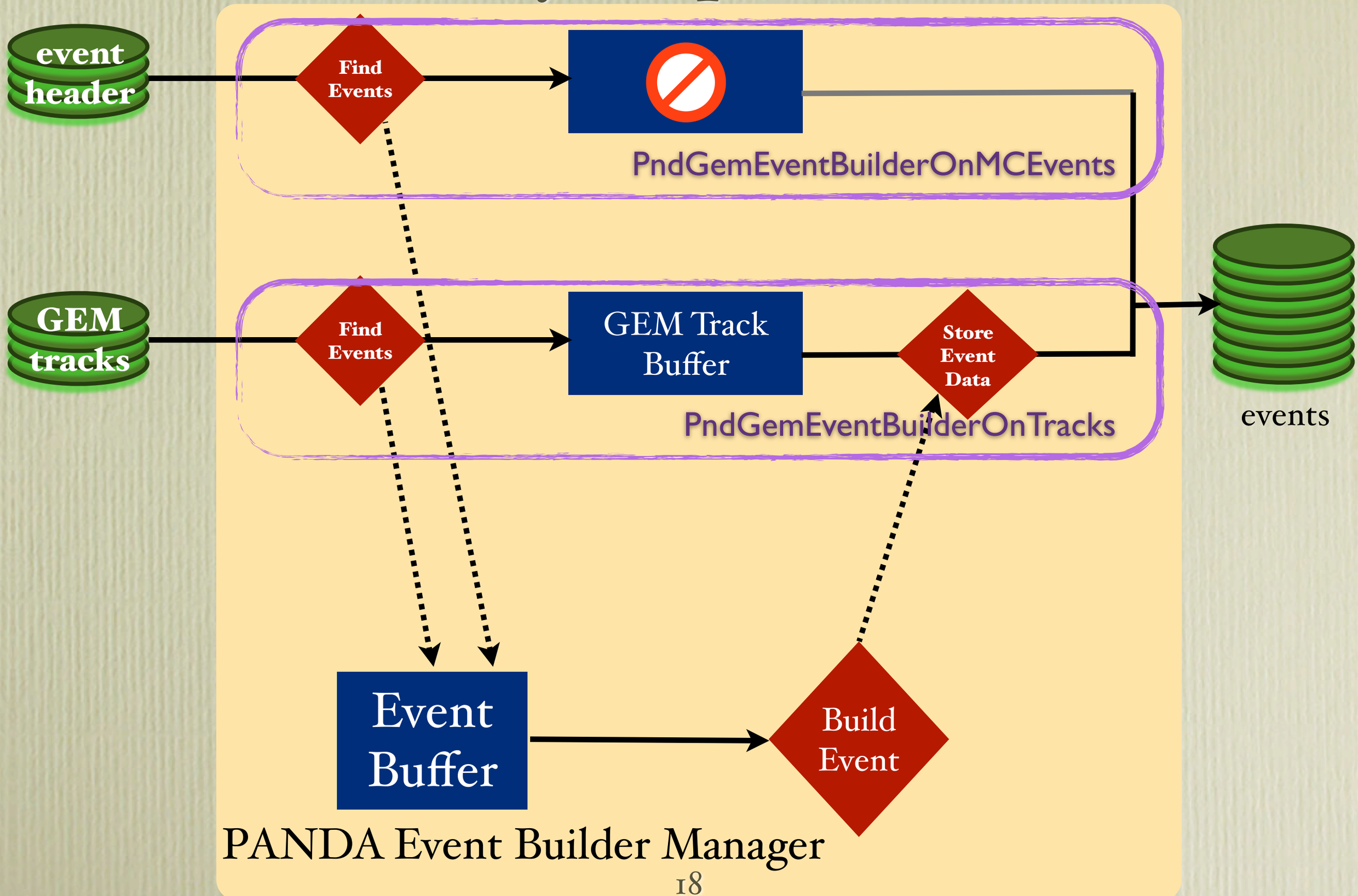
Event Builder Manager

- Internally has a vector of event builders
- Contains buffers with reconstructed events
- In the Exec(), it:
  - loops over event builders and calls FindEvents() function
  - analyses reconstructed event buffers and creates events
  - for each created event it calls StoreEventData() function

General idea



Already implemented



Usage

- The necessary changes to PandaRoot can be taken from: `svn/pandaroot/development/karabowi/eventBuilderDec2013`
- The code will be available in the trunk soon.

- Macro:

```
// -----   Event Builder   --------------------------------------
FairEventBuilder* eventBuilder = new FairEventBuilder("Event Builder", 0);//verboseLevel);
fRun->AddTask(eventBuilder);

PndGemEventBuilderOnMCEvents* gemEBOnMCEvents = new PndGemEventBuilderOnMCEvents("GemEBOnMCEvents",0);
eventBuilder->AddEventBuffer (gemEBOnMCEvents);
PndGemEventBuilderOnTracks*   gemEBOnTracks   = new PndGemEventBuilderOnTracks  ("GemEBOnTracks",0);
eventBuilder->AddEventBuffer (gemEBOnTracks);
// ------------------------------------------------------------------

// -----   Intialise and run   ------------------------------------------
fRun->Init();
fRun->RunEventReco(0,nEvents);
```

Few comments

- The output is organised as follows, to mimic as closely as possible the original structure of the data:
  - RecoEventHeader. Each entry in the TTree has an object of FairRecoEventHeader, where currently:
    - fEventTime,
    - fEventTimeError and
    - fIdentifier

    are stored;
  - TClonesArrays with the data.
- fIdentifier - to easily identify, which event builder was responsible for the event creation.

Few comments cont'd

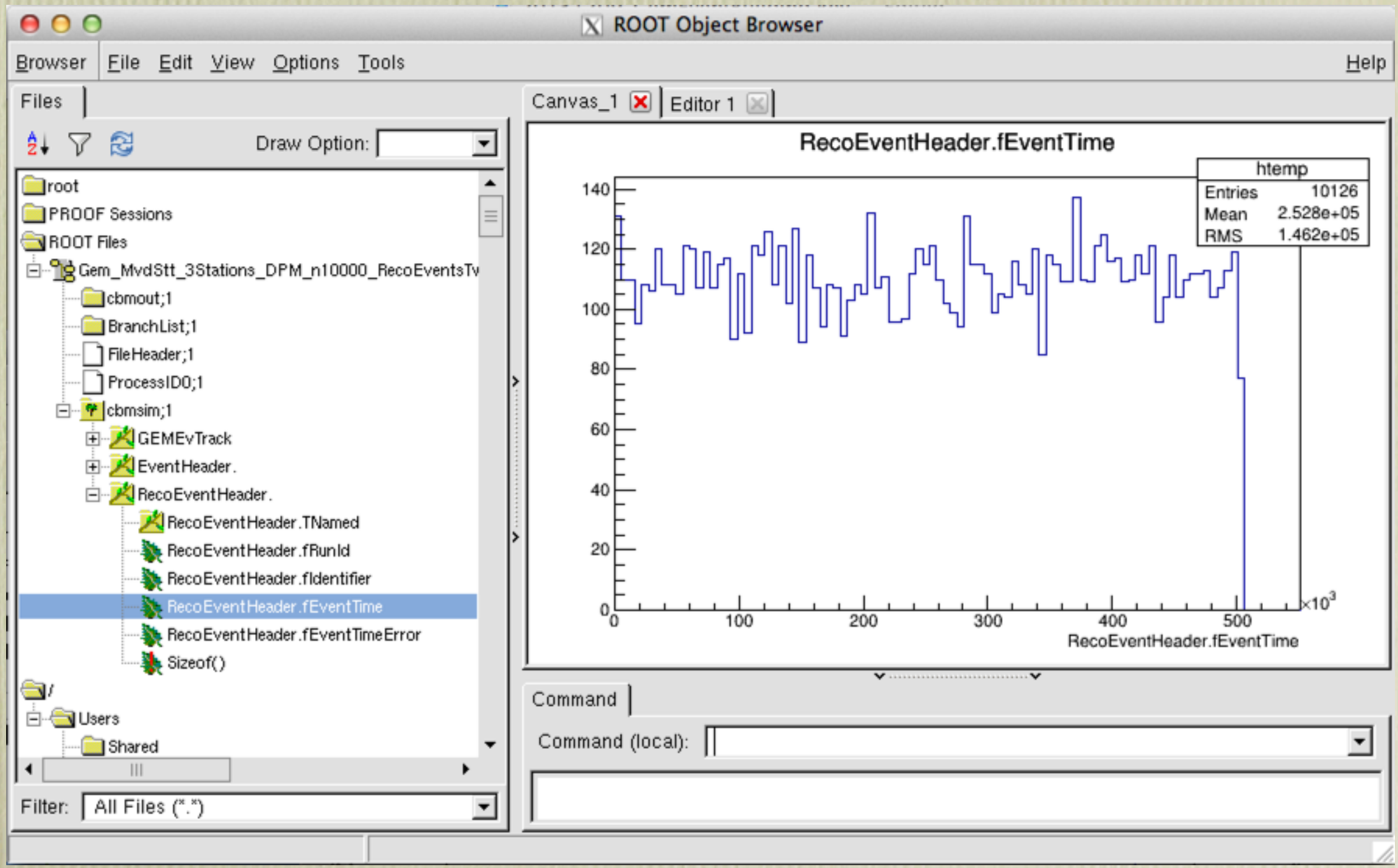
- PndGemEventBuilderOnMCEvents has nothing to do with GEM, I just developed it in gem/. The name should be changed.

- Naming problem:

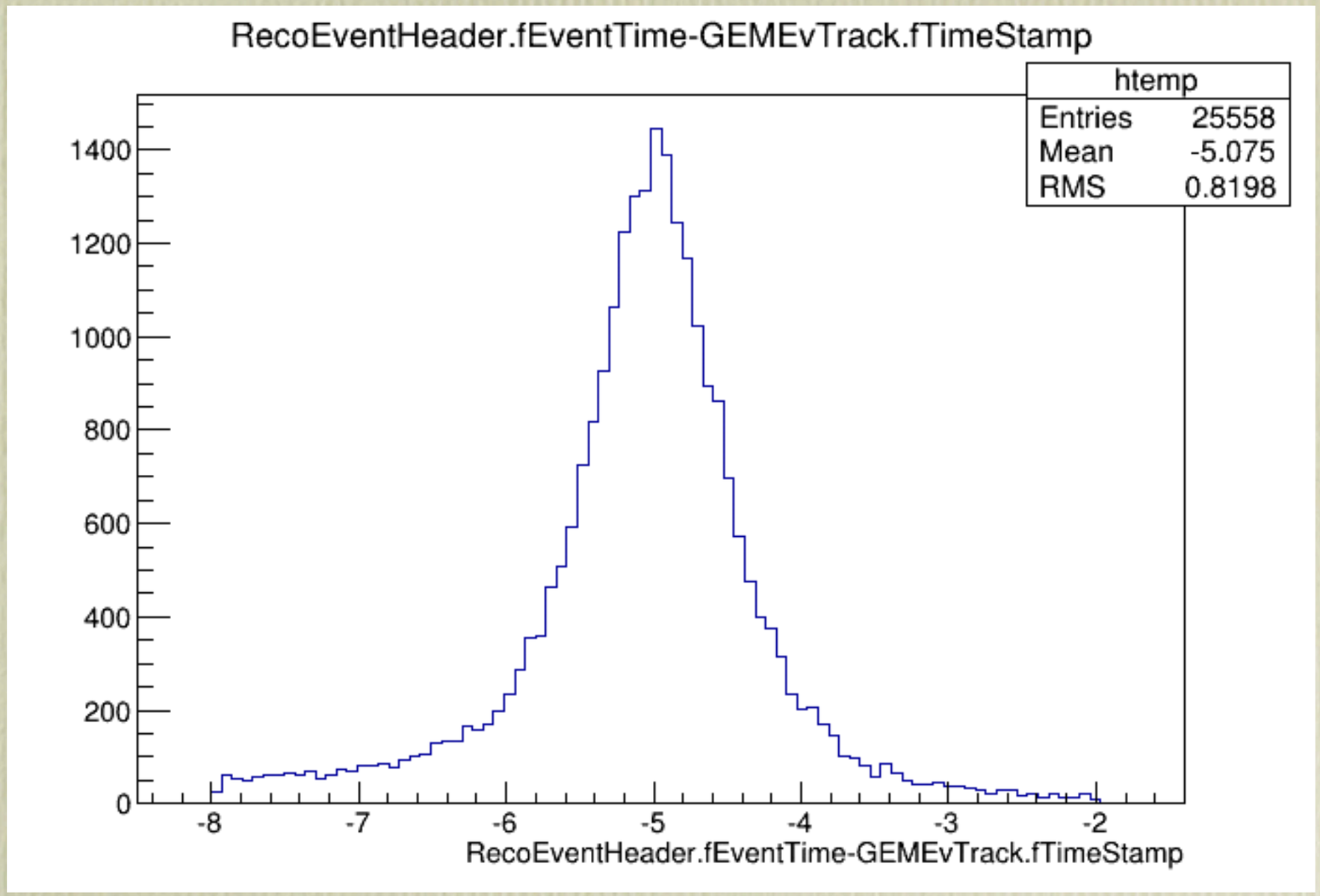
  - currently the EventBuilderManager is FairEventBuilder. Maybe it should be renamed to FairEventBuilderManager. There should be different implementations for different experiments, that will differ in analysis and extraction of events.

  - the task's event builders derive from FairEventBuffer. They are actually responsible for event finding, storing task specific information in buffer, and writing them to the output TClonesArrays. It's a lot of responsibility but why to split it? Maybe eventually FairEventBuffer will have to be renamed to FairEventBuilder.

Results

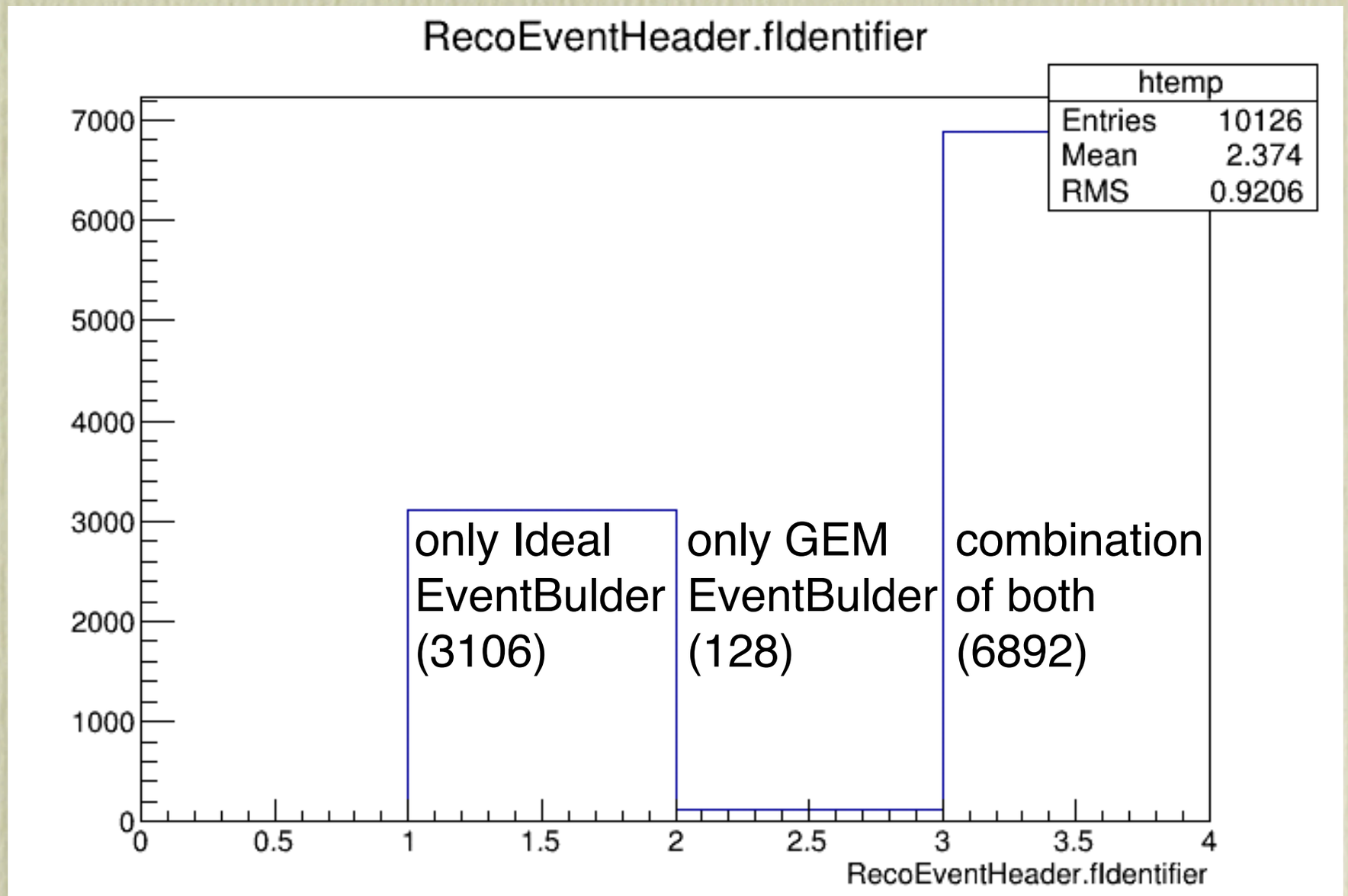

ROOT Object Browser
Browser
File
Edit
View
Options
Tools
Help
Files
Draw Option:
root
PROOF Sessions
ROOT Files
Gem_MvdStt_3Stations_DPM_n10000_RecoEventsTv
cbmout;1
BranchList;1
FileHeader;1
ProcessID0;1
cbmsim;1
GEMEvTrack
EventHeader.
RecoEventHeader.
RecoEventHeader.TNamed
RecoEventHeader.fRunId
RecoEventHeader.fIdentifier
RecoEventHeader.fEventTime
RecoEventHeader.fEventTimeError
Sizeof()
/
Users
Shared
Filter:
All Files (*.*)
Canvas_1
Editor 1
RecoEventHeader.fEventTime
htemp
Entries 10126
Mean 2.528e+05
RMS 1.462e+05
140
120
100
80
60
40
20
0
0
100
200
300
400
500
×10^3
RecoEventHeader.fEventTime
Command
Command (local):

Results cont'd



- Difference between reconstructed event time and GEM track time stamp

Results cont'd



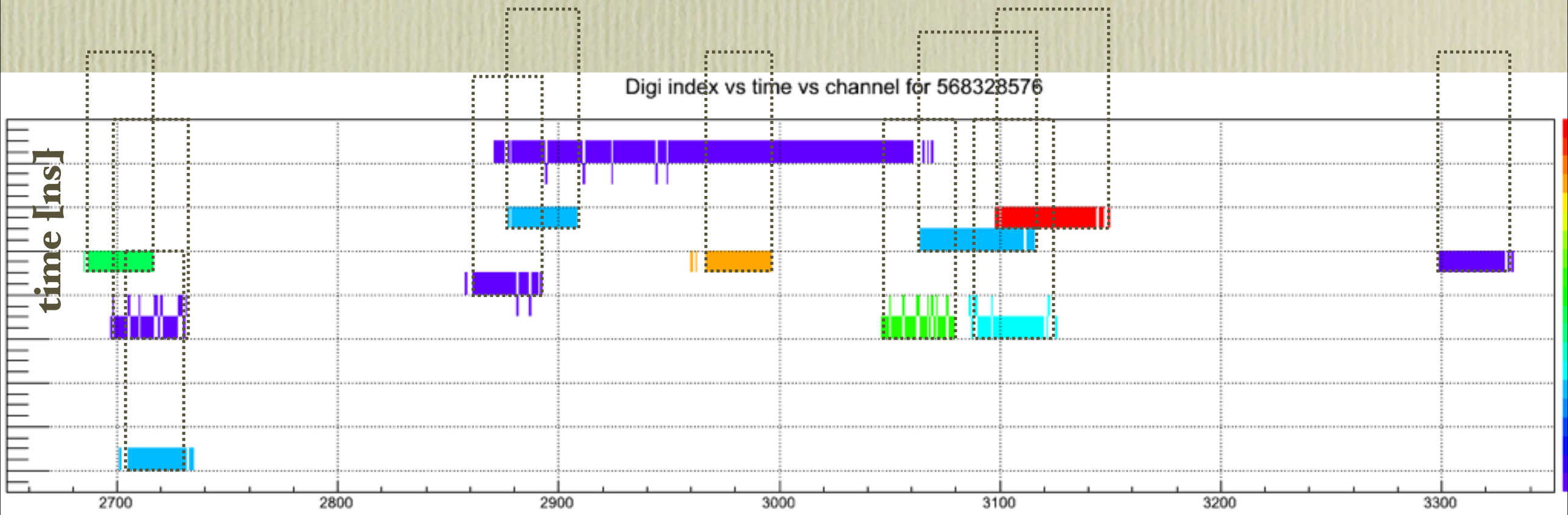
Summary

- A proposal of the Event Builder task structure is available.
- It can be easily extended with more event builders.
- Discussion needed to include more use cases.

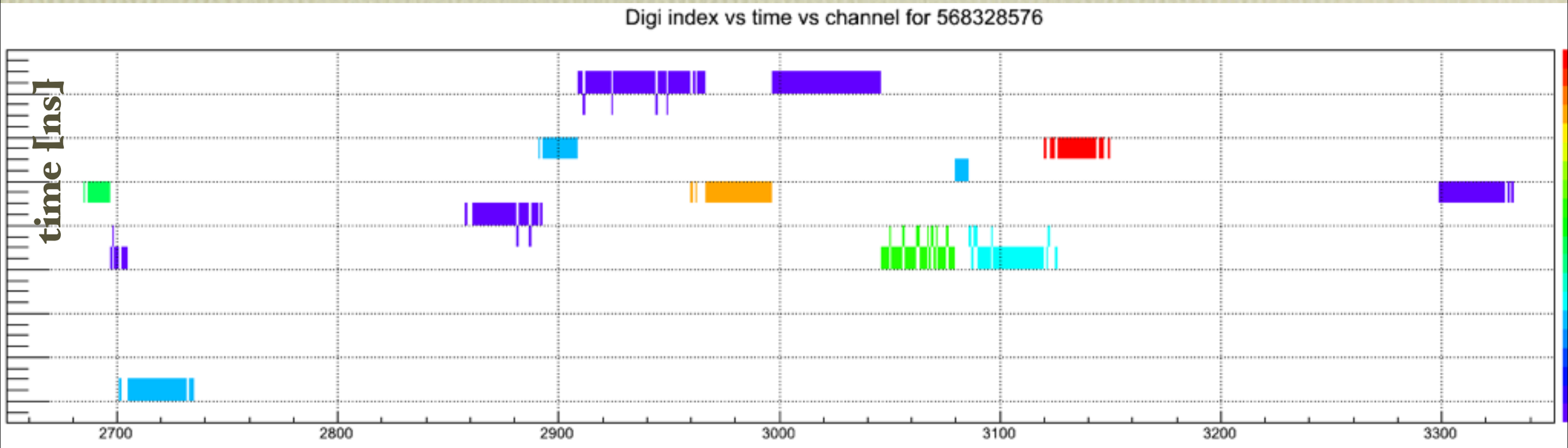
time [ns]

channel number

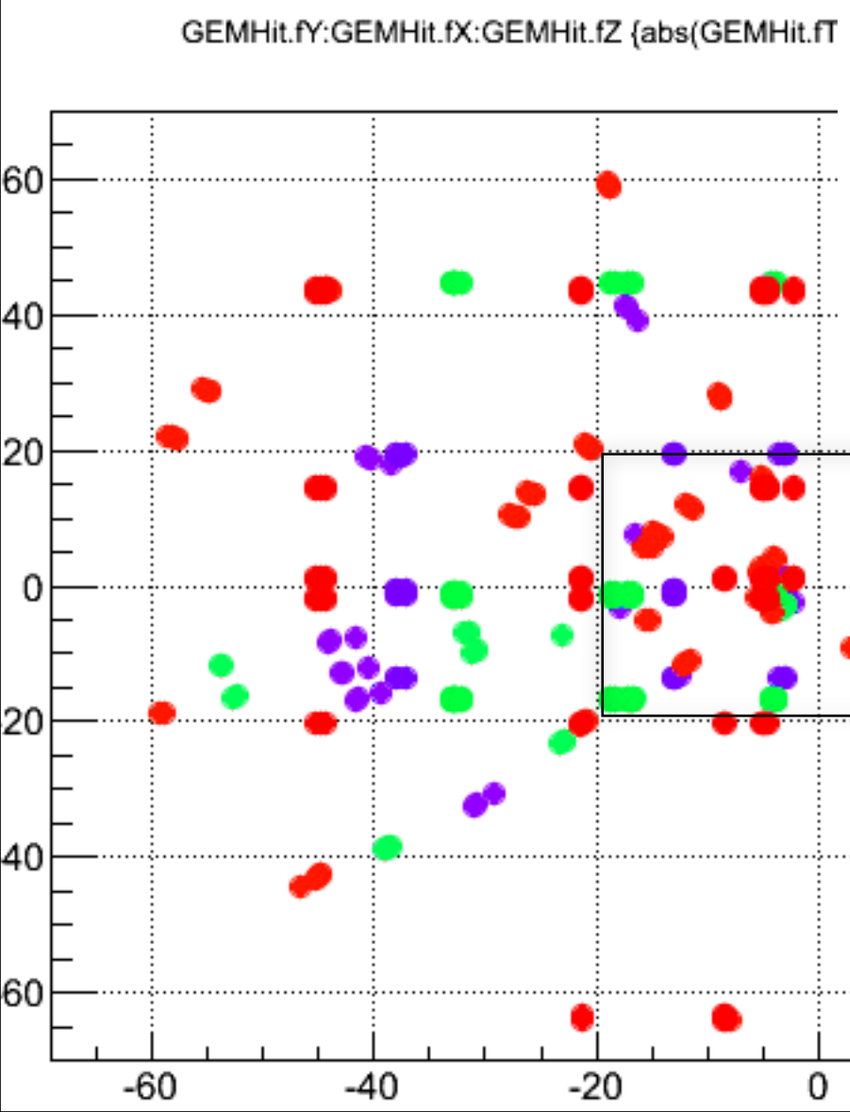
Without deadtime

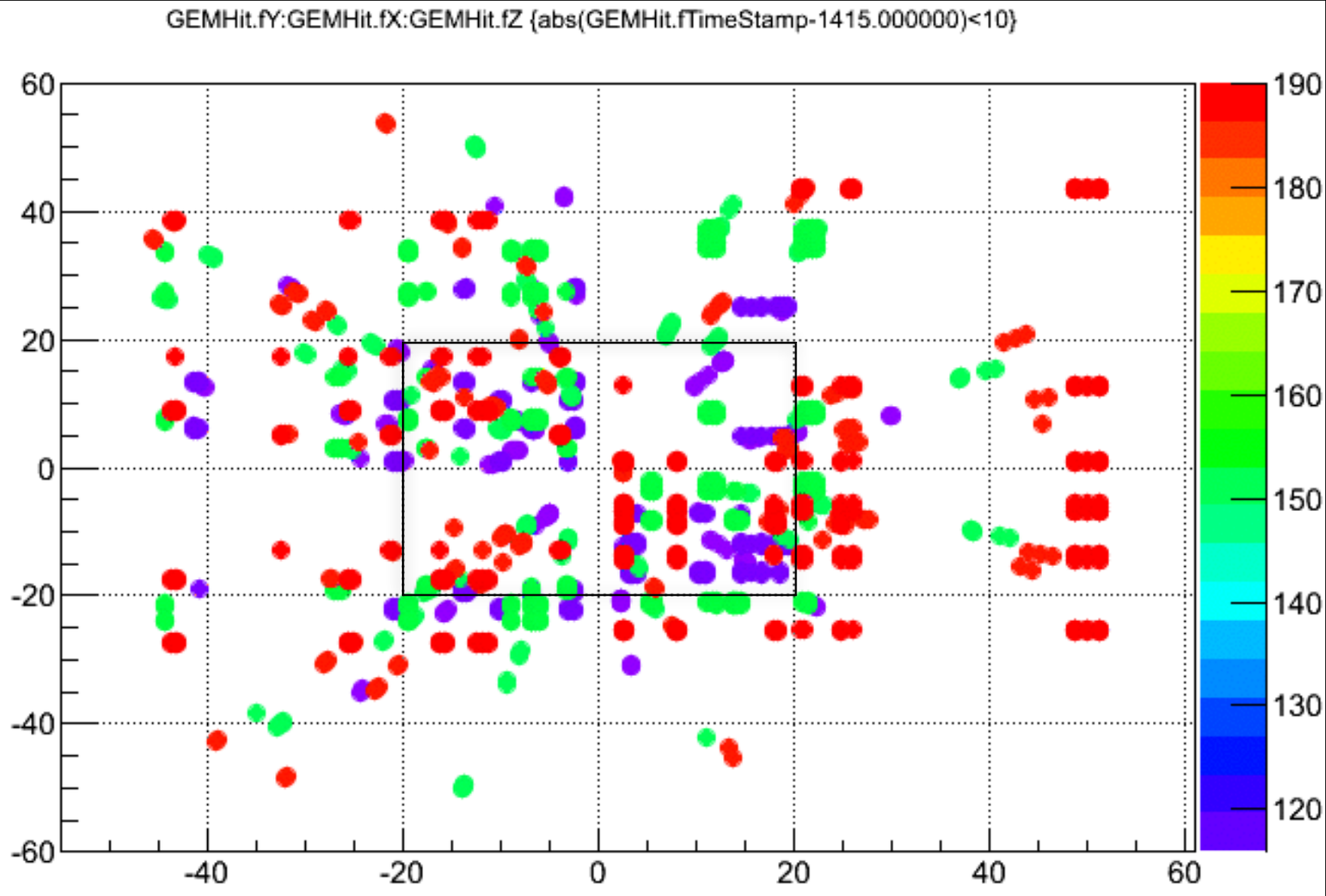


With 100ns deadtime



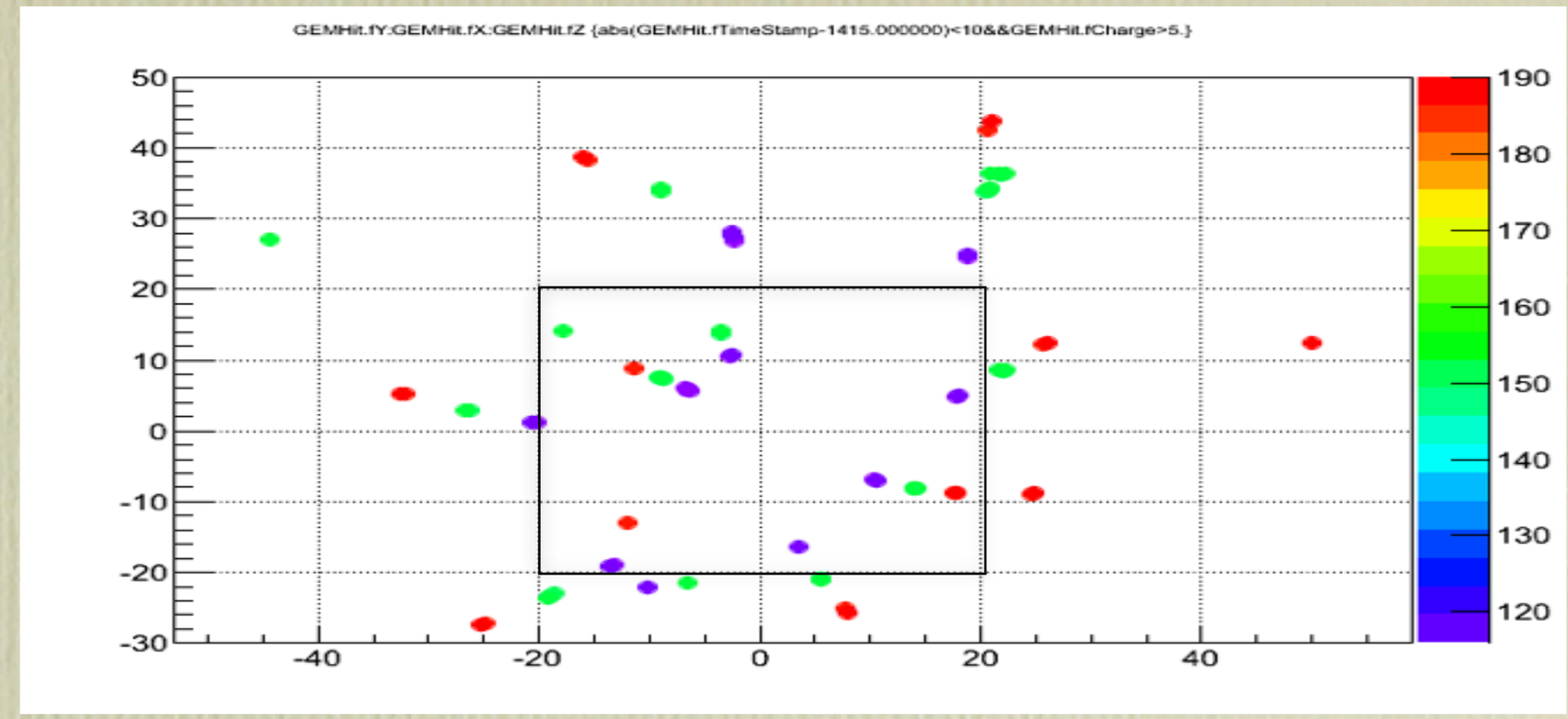
original hit finding:

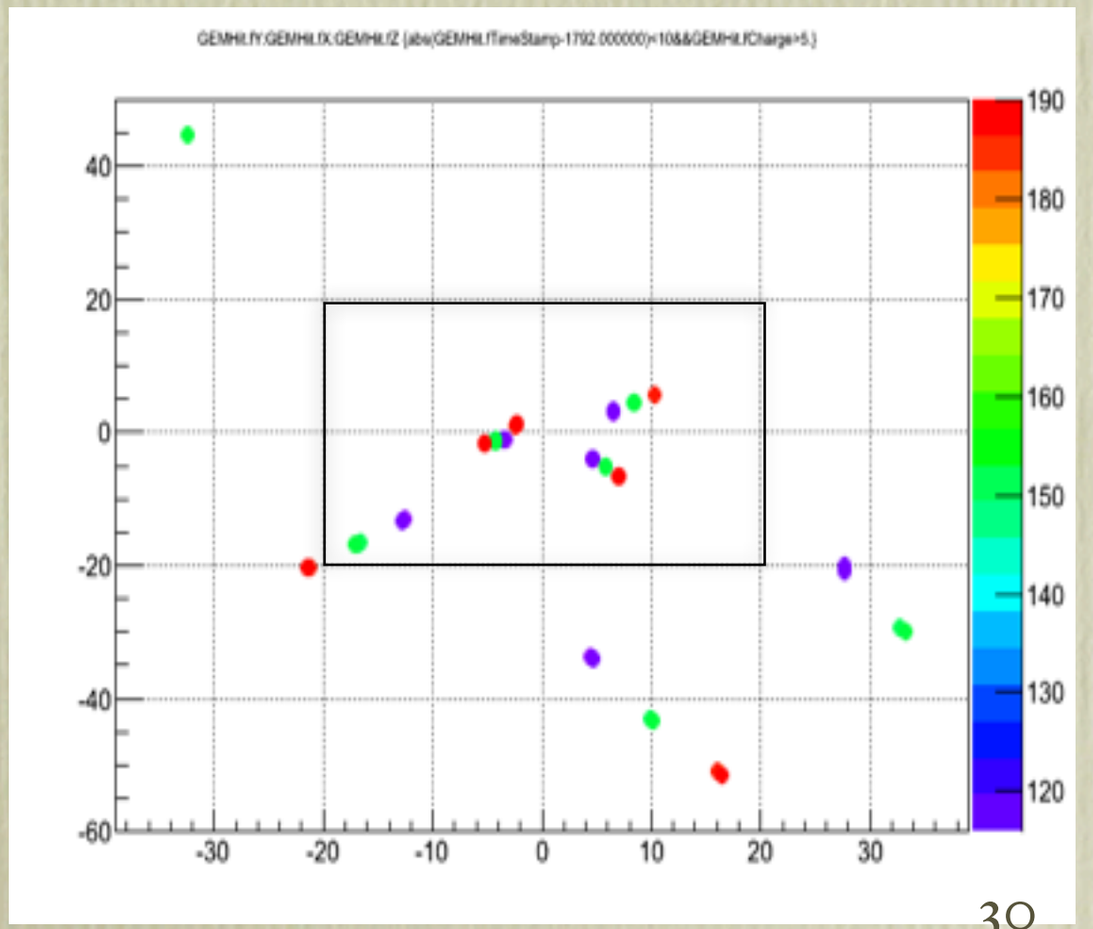




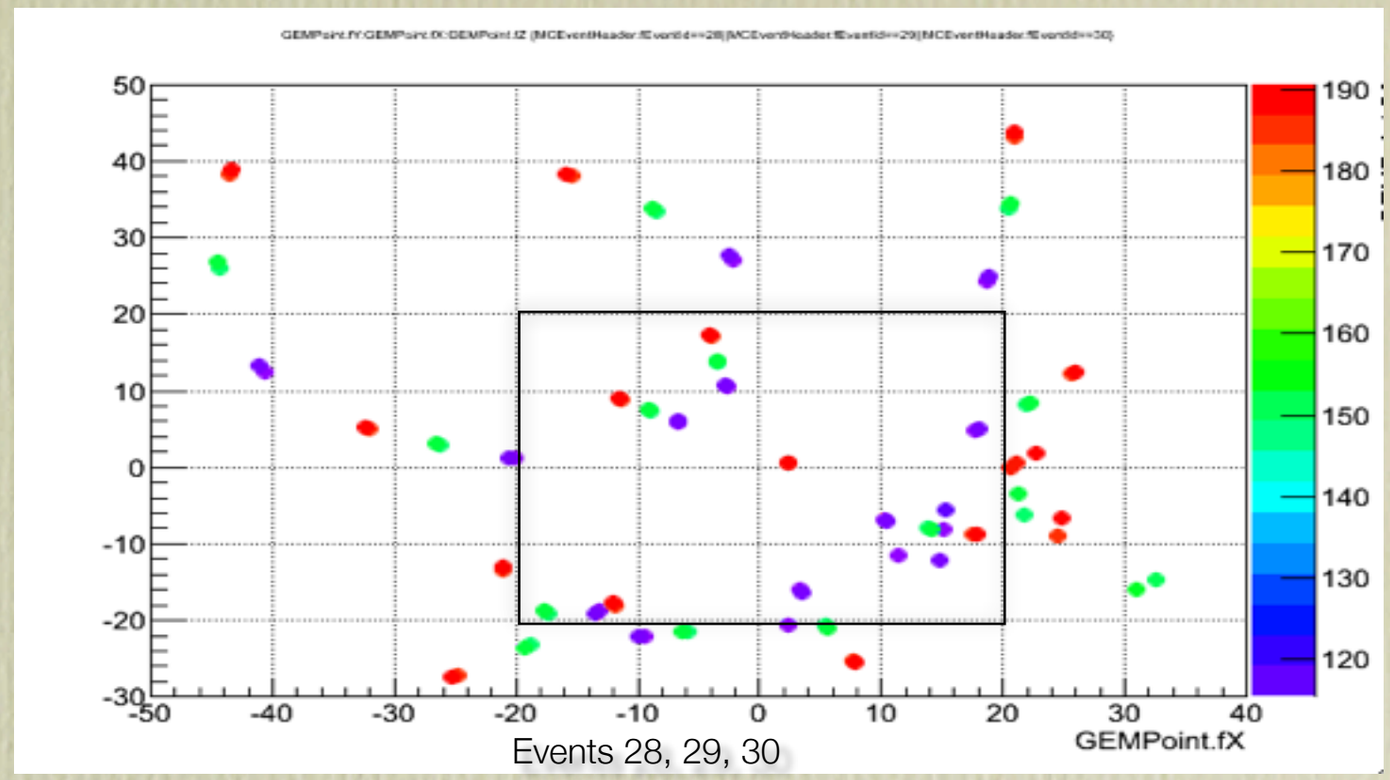
20ns time frames

hit finding after changes for TB

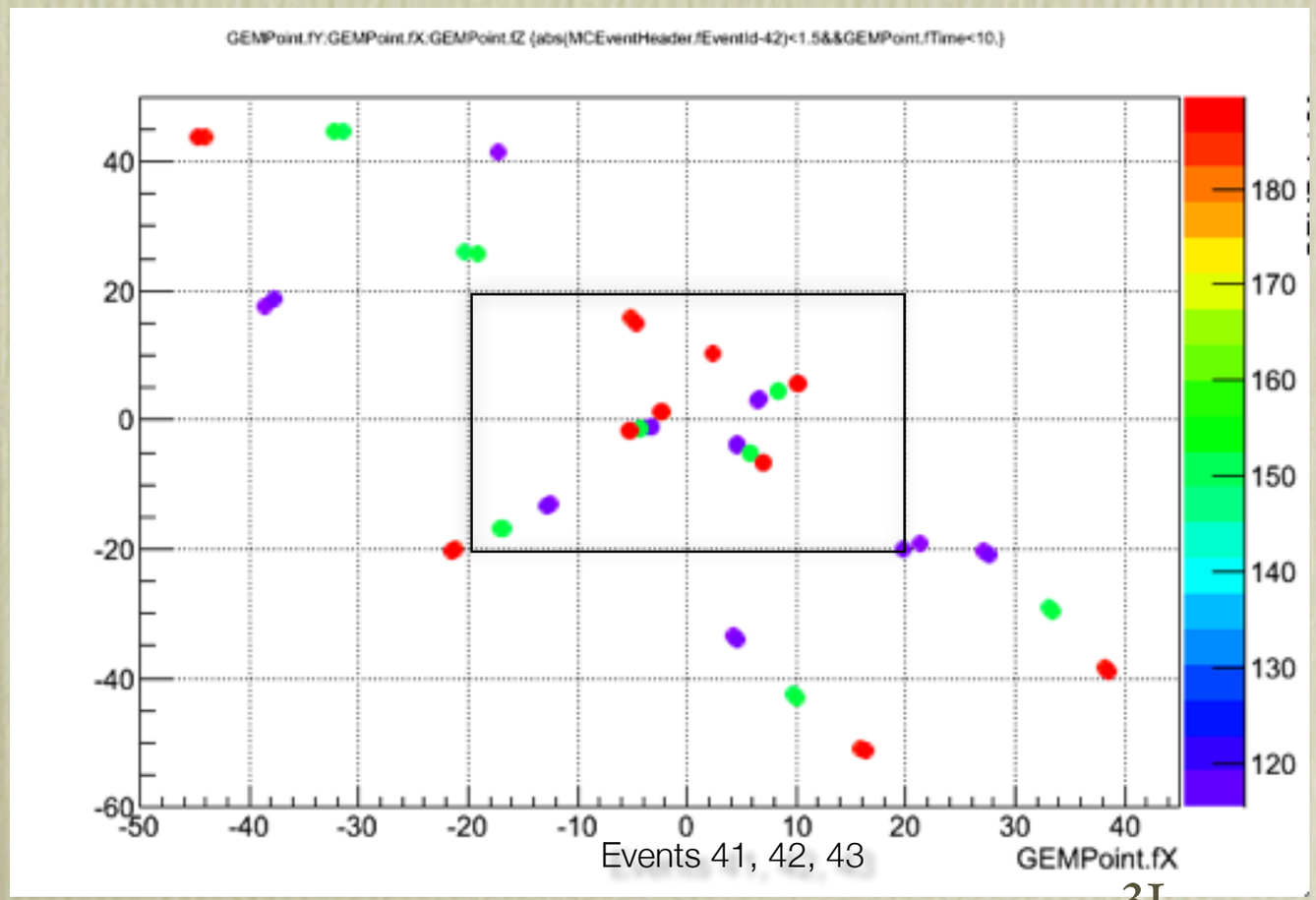




For comparison: MC points from corresponding MC events



Events 28, 29, 30



Events 41, 42, 43

PndGemTrackFinderOnHitsTB

```
FindTrackSegments()
  Double loop over hits to match close hits from
  different stations. Assuming start vertex as
  the third point, track parameters (momentum,
  theta, phi) are calculated and track segments
  are formed.
```

```
MatchTrackSegments()
  Match track segments with similar parameters
  to find track candidates.
```

```
RemoveCloneTracks()
  Check track candidates for bad tracks.
  Criteria for bad tracks are:
  - track segments too different,
  - small number of track segments,
  - many hits shared with other candidates.
  Remove bad track candidates.
```

```
CreateTracks()
  Surviving candidates form output array of
  tracks.
```