SBS software II
Simulation/digitization/decoding

Eric Fuchey
University of Connecticut

SBS summer '20 collaboration meeting
July 14-15, 2020

Overview of SBS software

Overview of SBS software

Main goals:

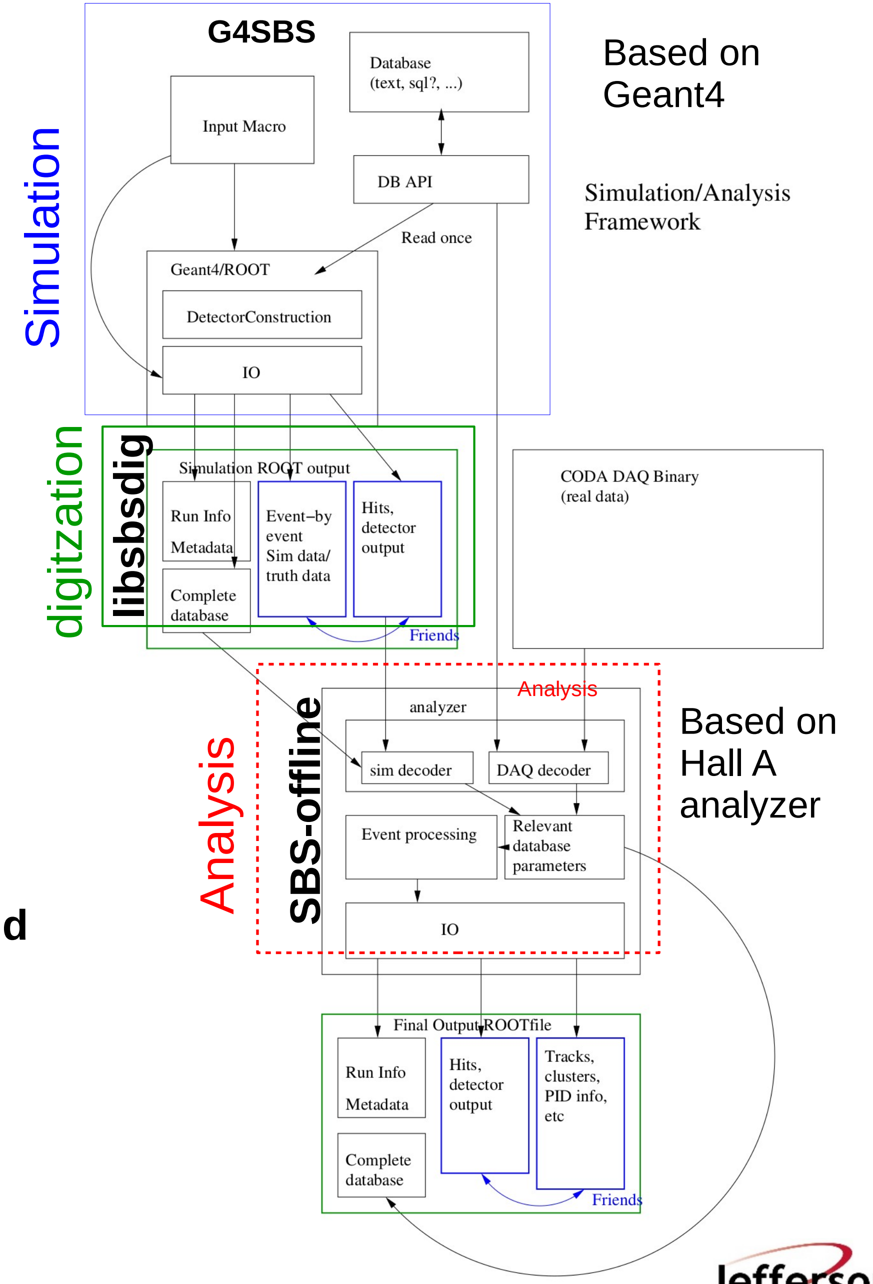
- "End-to-end" simulation:
production of pseudodata + simulation of data sizes;
- Analysis of pseudodata: test analysis chain

Requirements:

→ *modular* (ease configuration changes);
→ *accessible* (ease handling for new people);
→ *flexible* (ease inclusion of new configurations);

***** Also need:
- Well defined IO formats and standards
- Flexible database to accomodate both MC and data

Strong requirement:
Online and offline analysis to be ready and tested,
+ pseudo-data sets to be analyzed before data taking
=> critical given high luminosities / high detectors and DAQ rates.



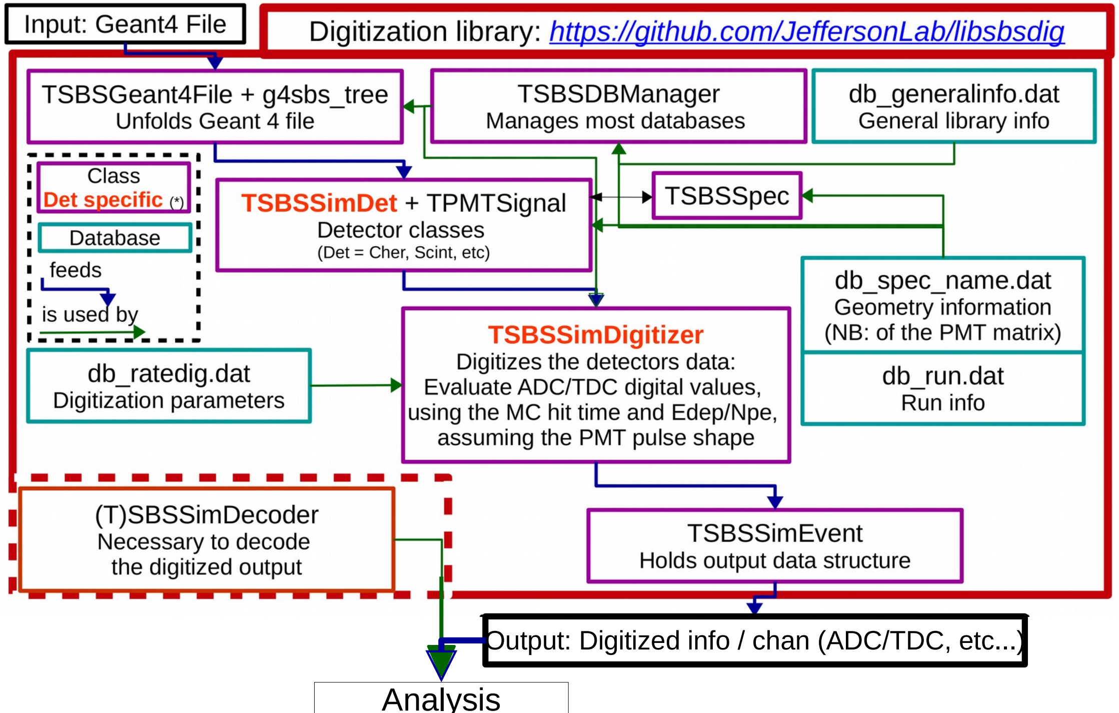
Digitization

Purpose:

* produce ADC/TDC values from G4SBS simulation, to process this infomation in SBS-offline and benchmark the analysis.

Libsbsdig:

* uses MC info (energy dep, number of p.e., time) to simulate ADC and TDC values
* adds user-defined levels of background on top of signal
* output files can be read by SBS-offline with SBSSimDecoder (merged in SBS-offline)
* now migrated under cmake
* documentation at *https://redmine.jlab.org/projects/sbs-software/wiki/Documentation_of_libsbsdig*



Input: Geant4 File

Digitization library: *https://github.com/JeffersonLab/libsbsdig*

TSBSGeant4File + g4sbs_tree
Unfolds Geant 4 file

TSBSDBManager
Manages most databases

db_generalinfo.dat
General library info

Class
Det specific (*)
Database
feeds
is used by

TSBSSimDet + TPMTSignal
Detector classes
(Det = Cher, Scint, etc)

TSBSSpec

db_spec_name.dat
Geometry information
(NB: of the PMT matrix)

db_run.dat
Run info

db_ratedig.dat
Digitization parameters

TSBSSimDigitizer
Digitizes the detectors data:
Evaluate ADC/TDC digital values,
using the MC hit time and Edep/Npe,
assuming the PMT pulse shape

(T)SBSSimDecoder
Necessary to decode
the digitized output

TSBSSimEvent
Holds output data structure

Output: Digitized info / chan (ADC/TDC, etc...)

Analysis

Digitization library

Issues with libsbsdig:

* Overkill for its mission!

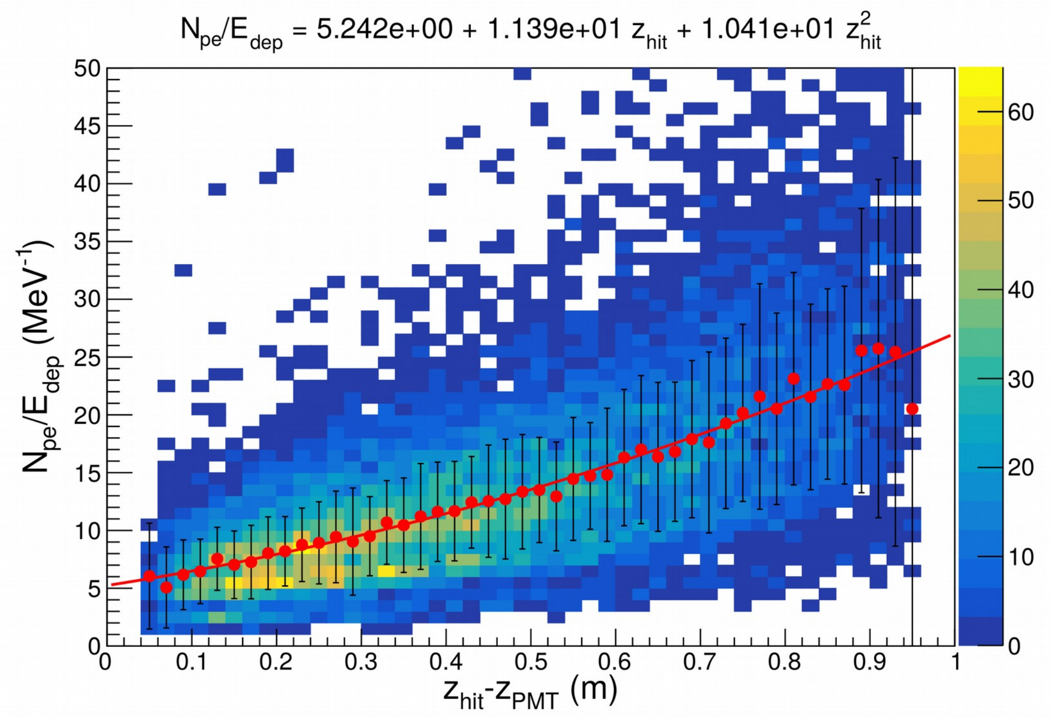
- * very complex – and cumbersome;
- * not intuitive to use;
- * as of now: **way too slow!**

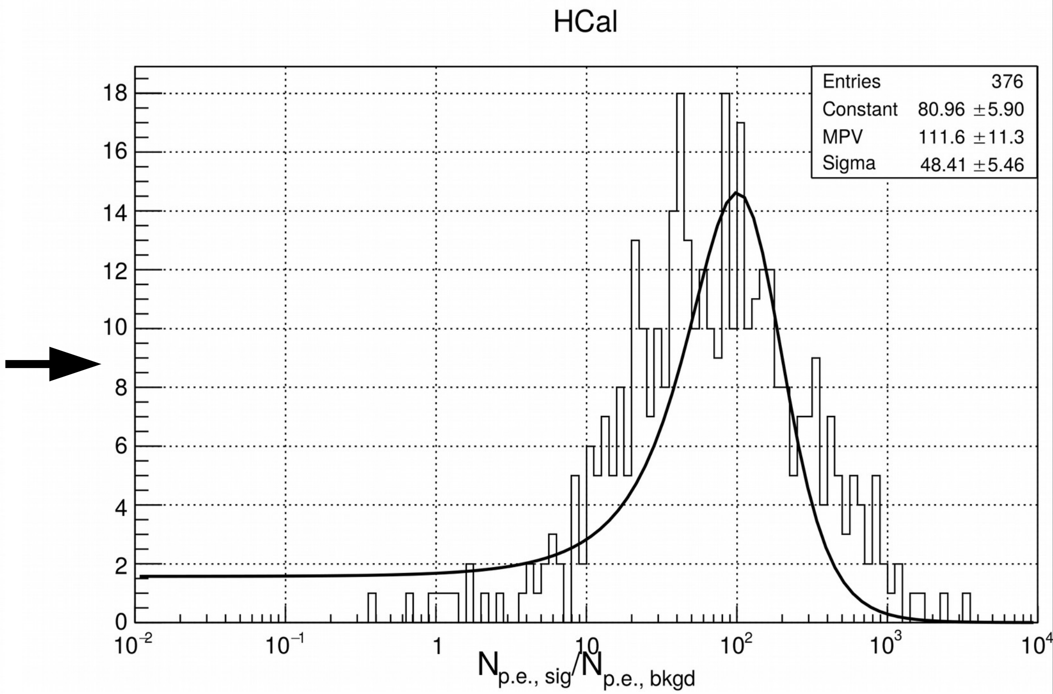
Plan of action:
* Give ourselves no more than a couple weeks to try to at least address the speed issue which is a big showstopper
* if unsuccessful: grab the valuable pieces of code out of it and turn it into something lightweight and easy to use

Digitization

 * For calorimeters and scintillators, the digitization library only uses energy deposits recorded by g4sbs.

 * For scintillators based detectors (HCal (below), hodoscope), p.e. yield more or less proportional to energy deposit, till very low energy deposits.



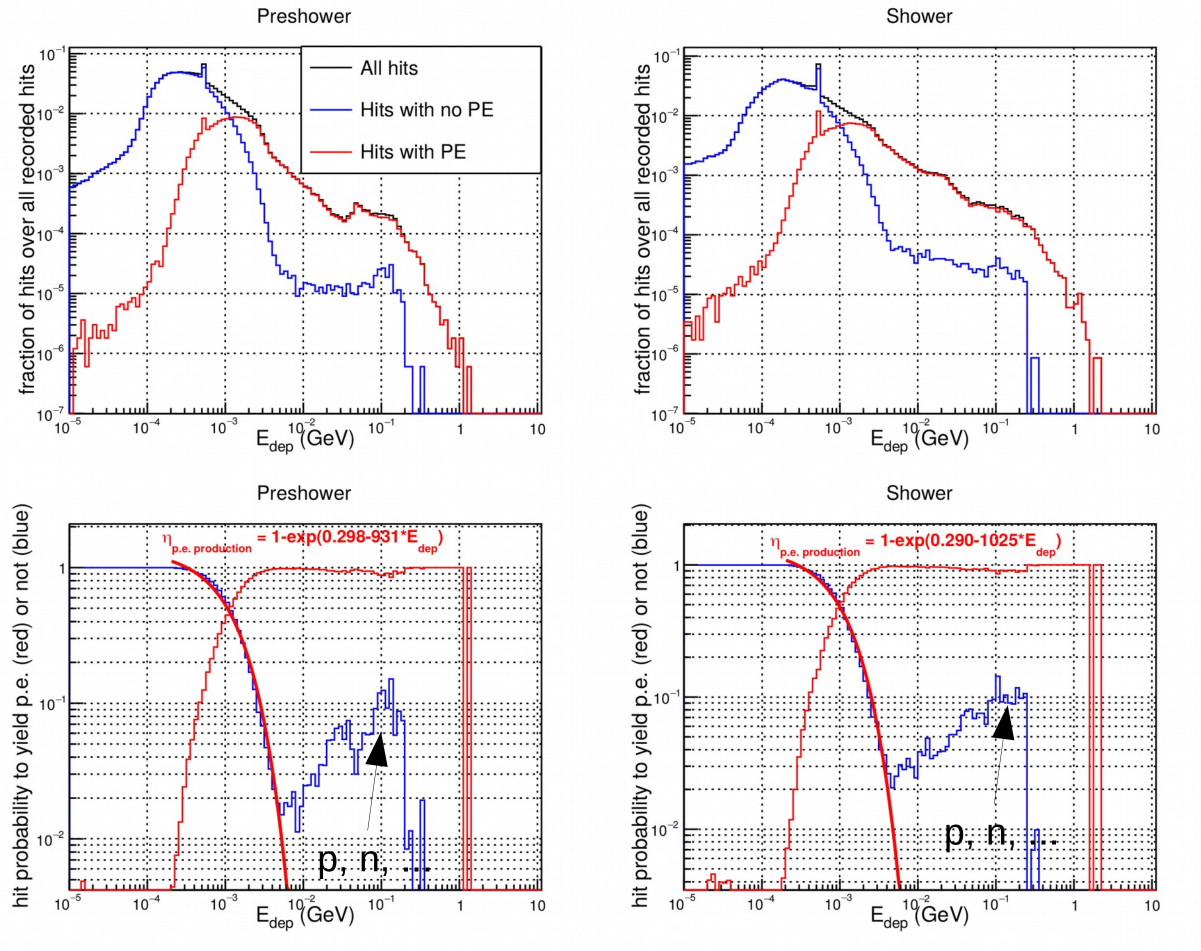


Digitization

* For lead glass detectors, things are slightly more complicated, since at very low energy, the hits may not give signal

* Apply on each hit below ~10 MeV a probability function to give some photoelectrons or not.

* Calculate relative p.e. yield of the hit as: $N_{pe} = 300\ pe/GeV \left(\frac{1-1/(n\beta)^2}{1-1/n^2} \right)$ with n=1.68 and $\beta = \sqrt{((m_e + e_{dep})^2 - m_e^2)/(m_e + e_{dep})}$



Digitization

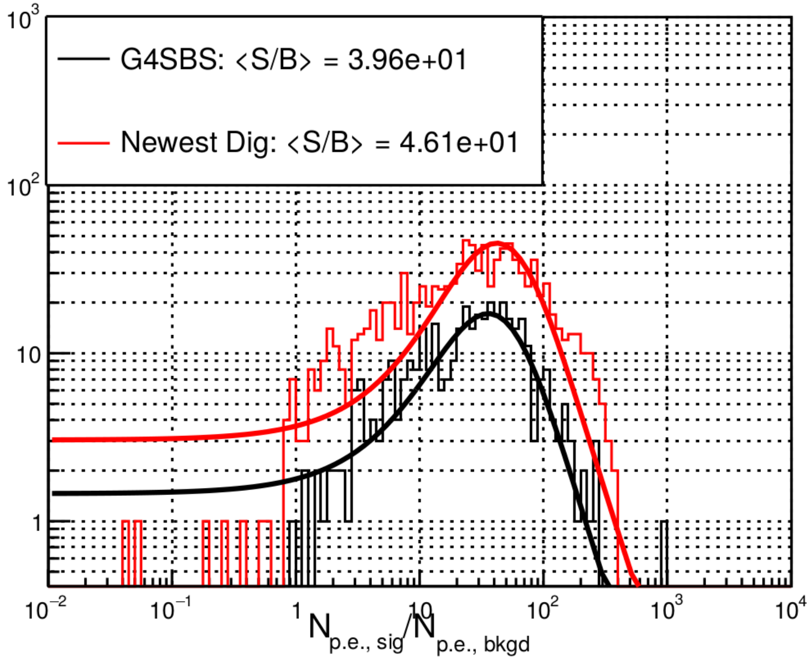
* For lead glass detectors, things are slightly more complicated, since at very low energy, the hits may not give signal

* Apply on each hit below ~10 MeV a probability function to give some photoelectrons or not.

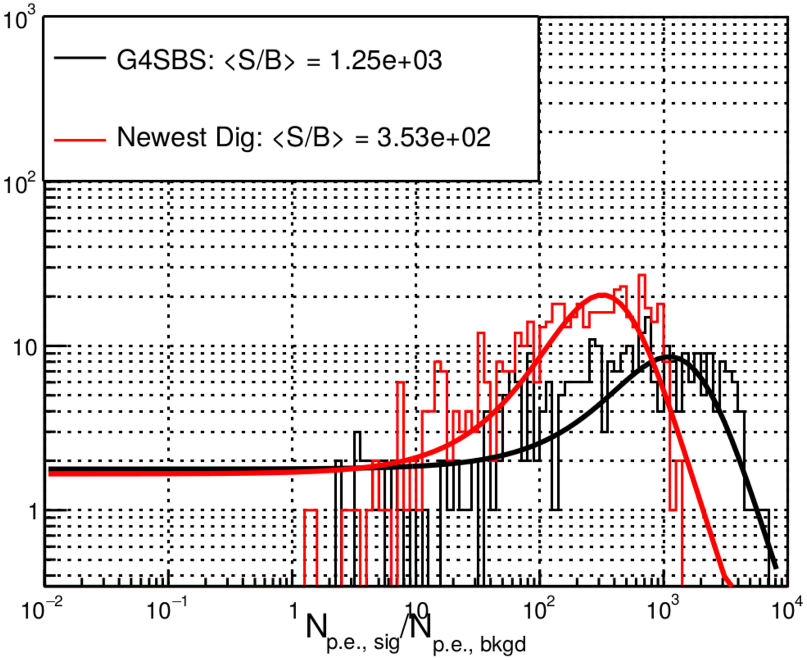
* Calculate relative p.e. yield of the hit as: $N_{pe}=300\, pe/GeV\left(\frac{1-1/(n\beta)^2}{1-1/n^2}\right)$ with n=1.68 and $\beta=\sqrt{((m_e+e_{dep})^2-m_e^2)/(m_e+e_{dep})}$

Resulting S/B ratios for PS/SH

PS: S/B, largest energy deposit block



SH: S/B, largest energy deposit block



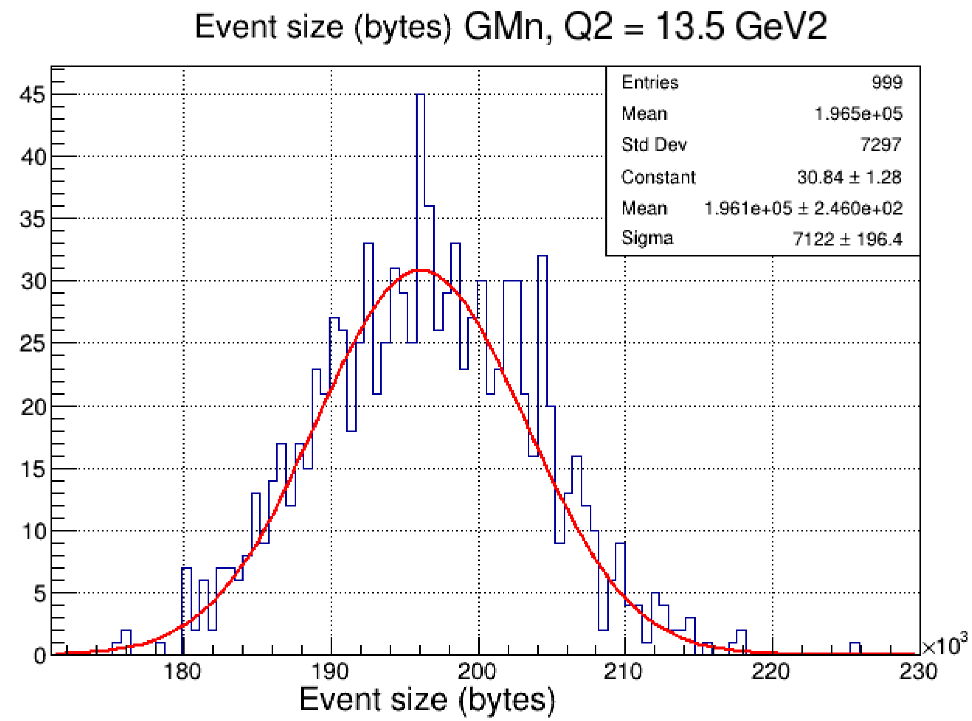
Digitization applications

* Estimation of event sizes using the digitized simulation

Using

$S_{evt} = (N_{hits}\ \mathrm{PS} + N_{hits}\ \mathrm{SH} + N_{hits}\ \mathrm{GRINCH} + N_{hits}\ \mathrm{Hodo} + N_{hits}\ \mathrm{GEM} \times 3 + N_{hits}\ \mathrm{HCal} \times 10) \times 4$

∼70 % (1.38×10^5 bytes = 135 kB) of this size is due to the front 4 INFN GEMs alone, which are confronted to a specific flux of between 100 and 130 kHz/cm^2;
∼ 12 % (2.4 × 10 5 bytes = 23.4 kB) of this size is due to the back GEM, which is only confronted to a specific flux of ∼40 kHz/cm 2, but is also wider (2 × 0.6 m^2 instead of 1.5 × 0.4 m^2).



SBSSimDecoder

Main purpose : interface between libsbsdig and SBS-offline.

 * takes the information stored from the libsbsdig output file, decodes it, and stores the information into standard podd objects (Decoder::Module)

 * requires detector map (and channel map when relevent) from the detectors databases

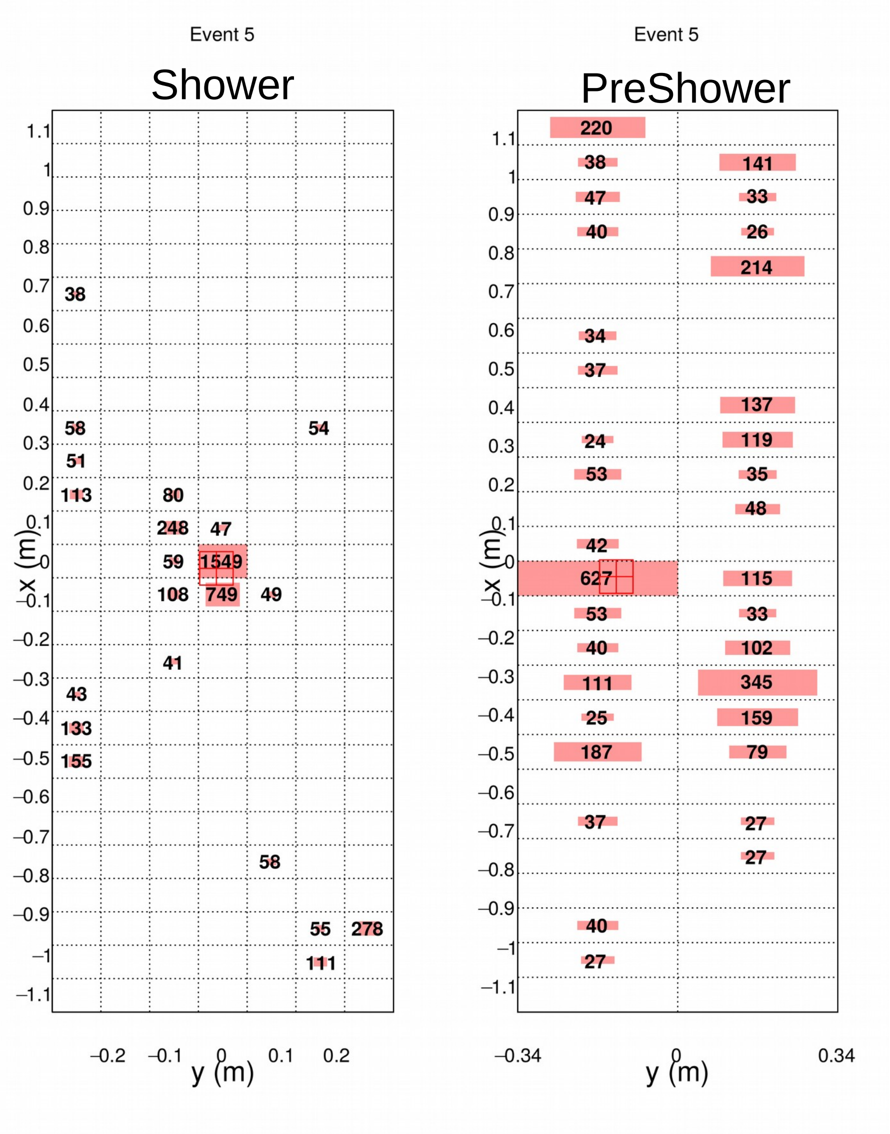
 * currently implemented in SBS-offline

 * may need revision if major revision of libsbsdig

SBSSimDecoder

Interface with SBS-offline

Example of analyzed digitized event display for G_M^n subsystems **full background**



Conclusion

*** Need *IMMEDIATE* focus on improving libsbsdig speed!**

- the code is overcomplex for its own scope;
- wouldn't be a problem if it wasn't *slow*
- most likely due to underoptimal ancillary classes (digitization itself seems to be fine)
  - * good news is that the speed may be improved by a lot;
  - * bad news is that the whole structure of the code needs to be revisited; at best simplified, perhaps rewritten;
- **plan of corrective action already setup**

* Following this, SBS-offline SBSsim decoder may have to be revised as well: