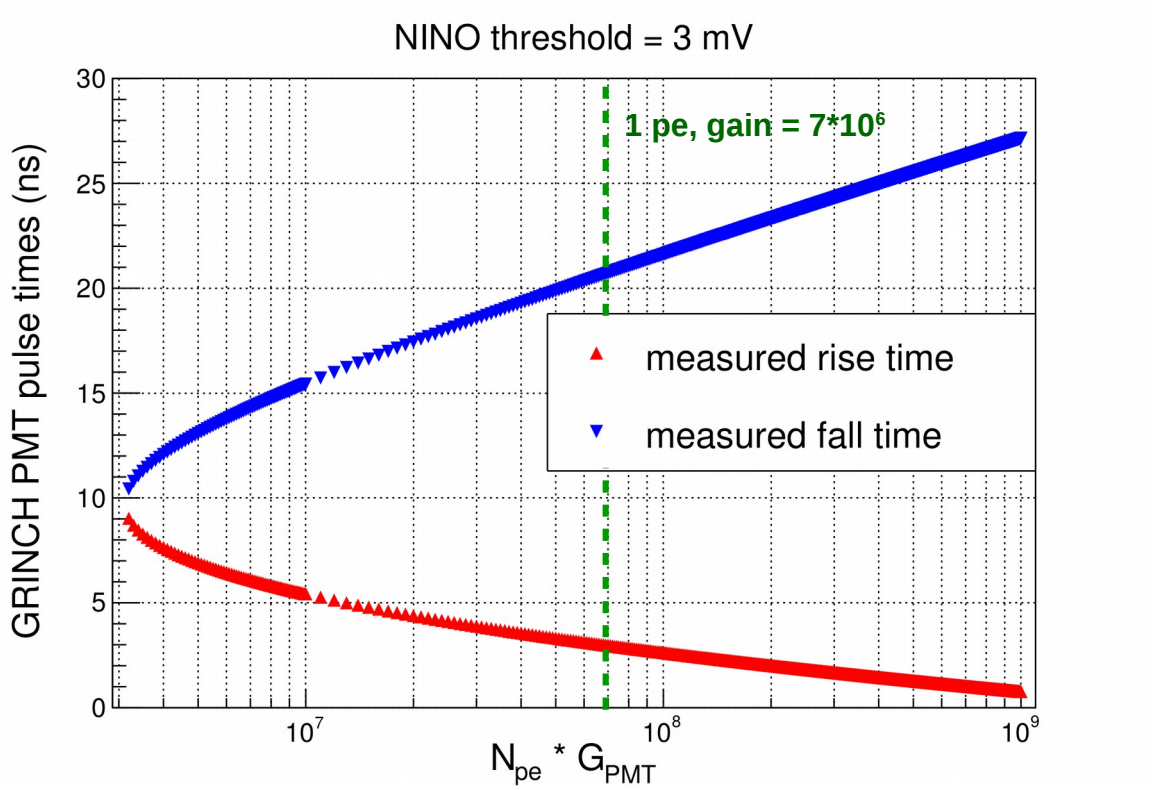
Update: digitization of Cherenkov detectors

Eric Fuchey
University of Connecticut

SBS simulation/software meeting
2017/11/29

UCONN Jefferson Lab

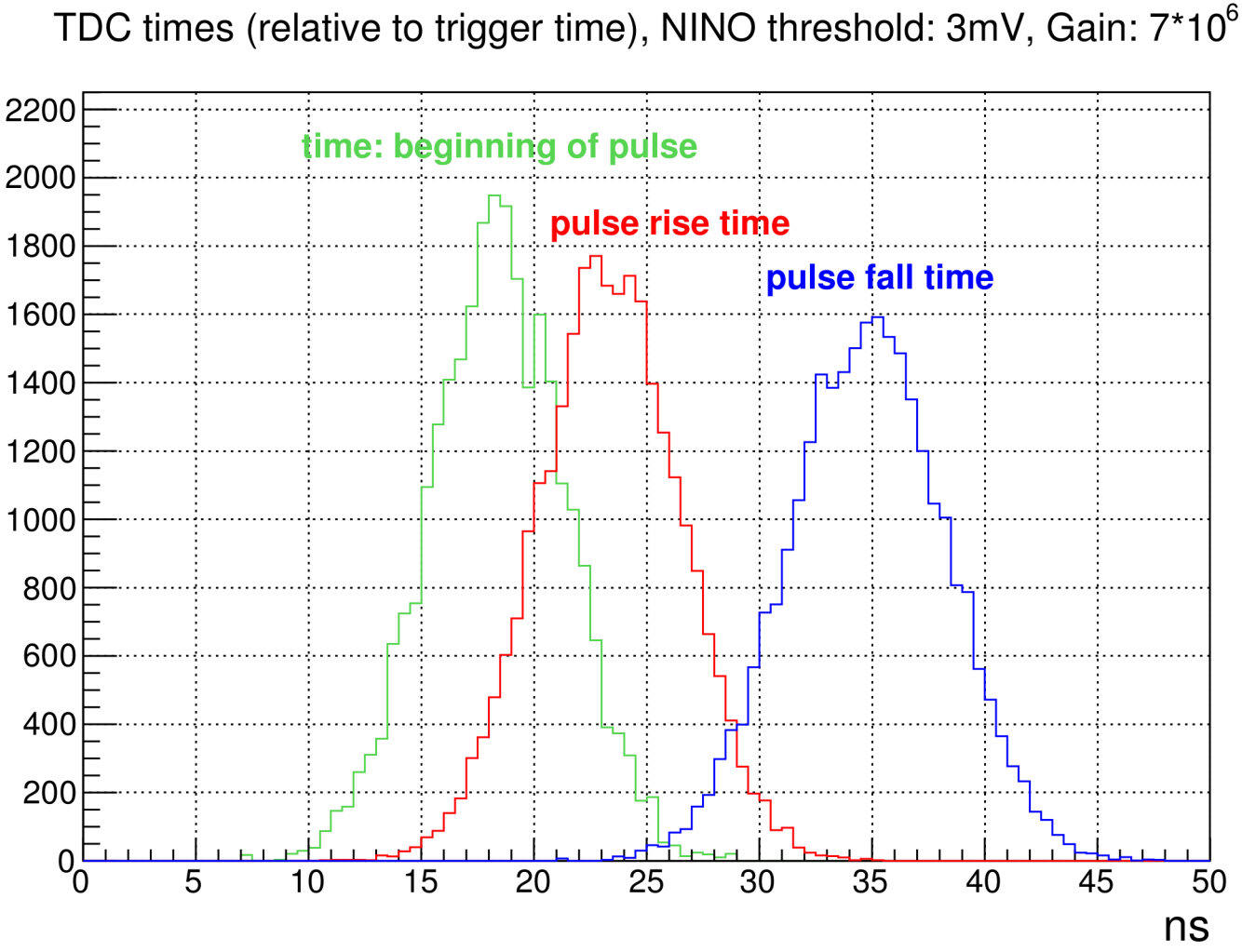
Cherenkov Digitization library:
Evaluation of PMT rise and fall times : done! (on paper)



Changed the NINO threshold to 3mV (might still change);
* plugged these tables in the digitization library: could not manage to fit it;
* however, exponential extrapolation between two points is a reasonable approximation;
* If something else has to be changed (i.e. NINO threshold and/or pulse normalization) it is easy to redo these tables ;

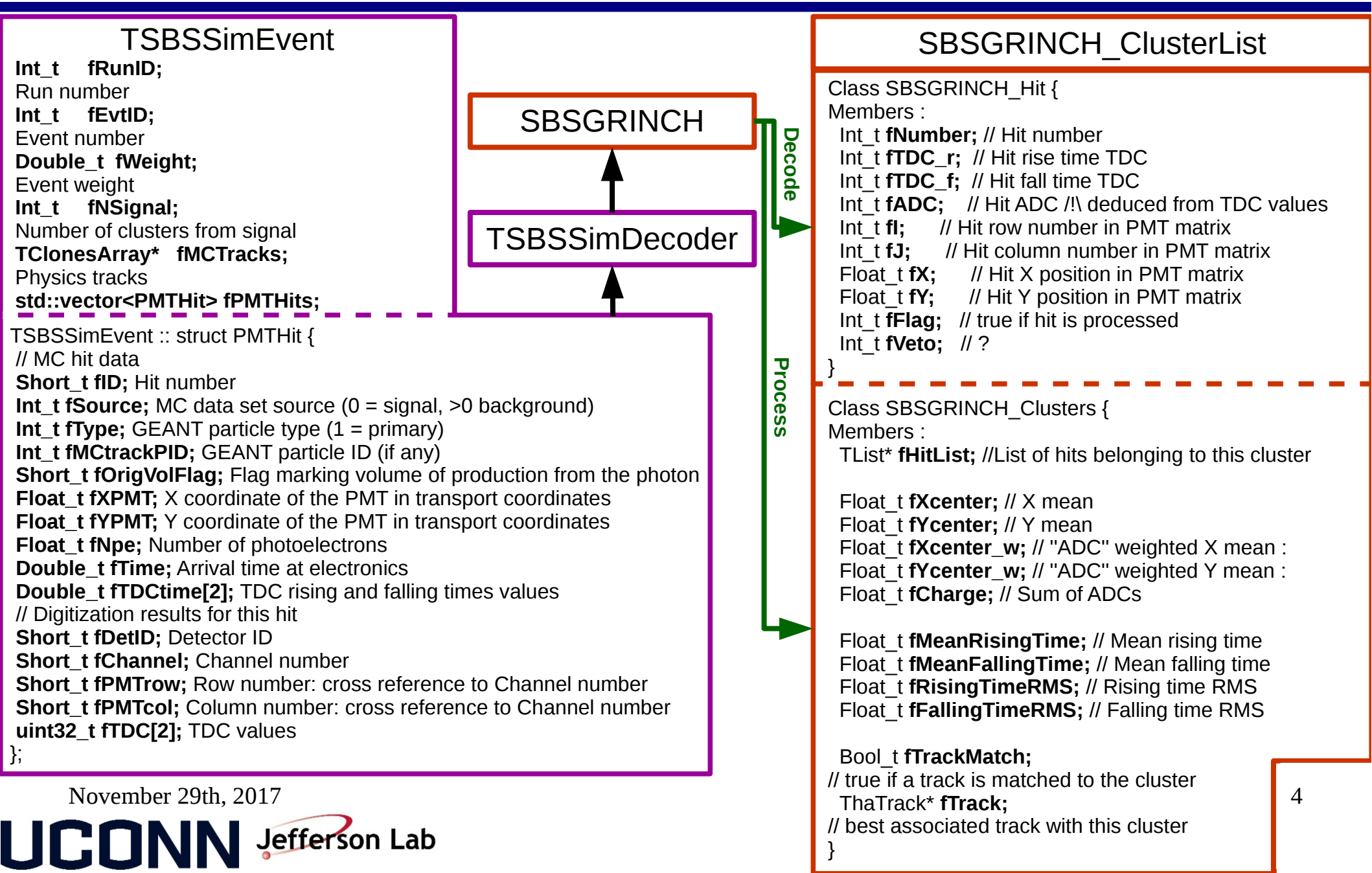
UCONN Jefferson Lab

Cherenkov Digitization library:
Evaluation of PMT rise and fall times : done! (on paper)



UCONN Jefferson Lab

Library/SBS-offline interface : Dataflow



Interface library with SBS-offline
In progress...

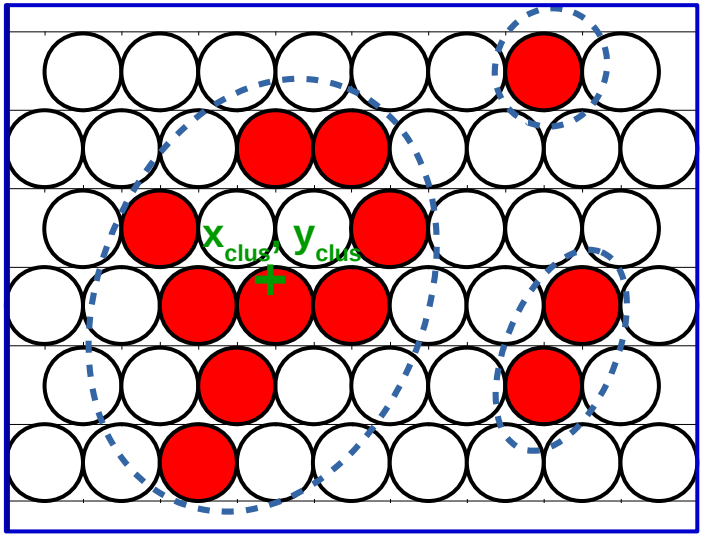
Requires *massive* rewriting SBS-offline class SBSGRINCH, for both decoding and processing:

Data analysis steps for GRINCH (under development) **:**

- Decoding (*started*) : fill the hits from the raw/digitized data ;

- Clustering (*rewritten*) : all neighboring hits PMTs are considered to be part of the same cluster (*);

- one may consider refining clustering with hit time information (*TODO*);



UCONN Jefferson Lab

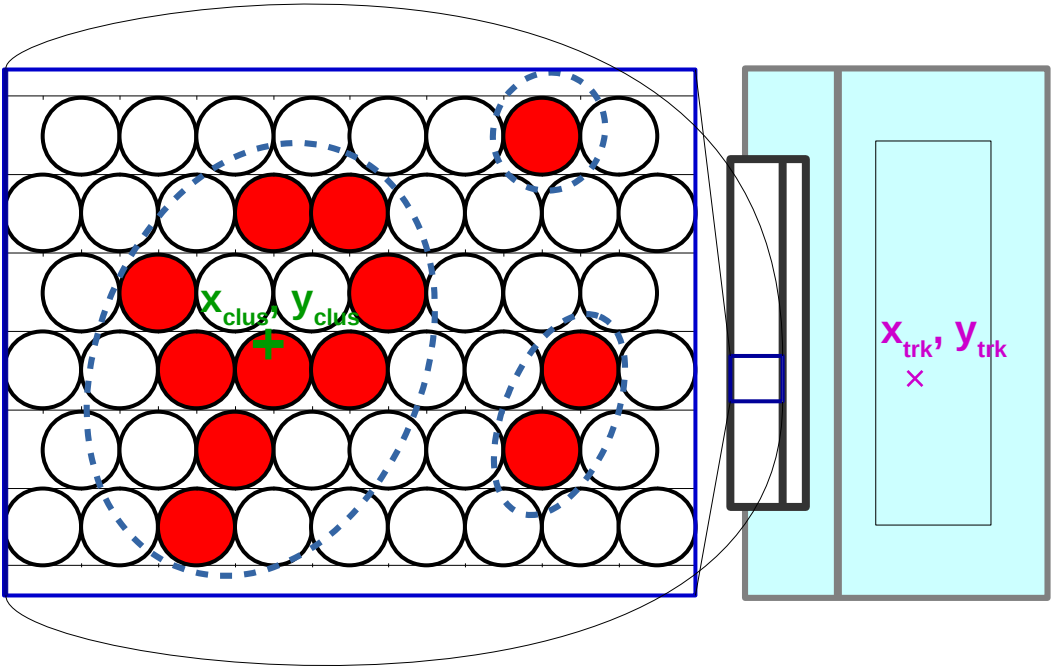
(*) this will have to be different for the RICH...

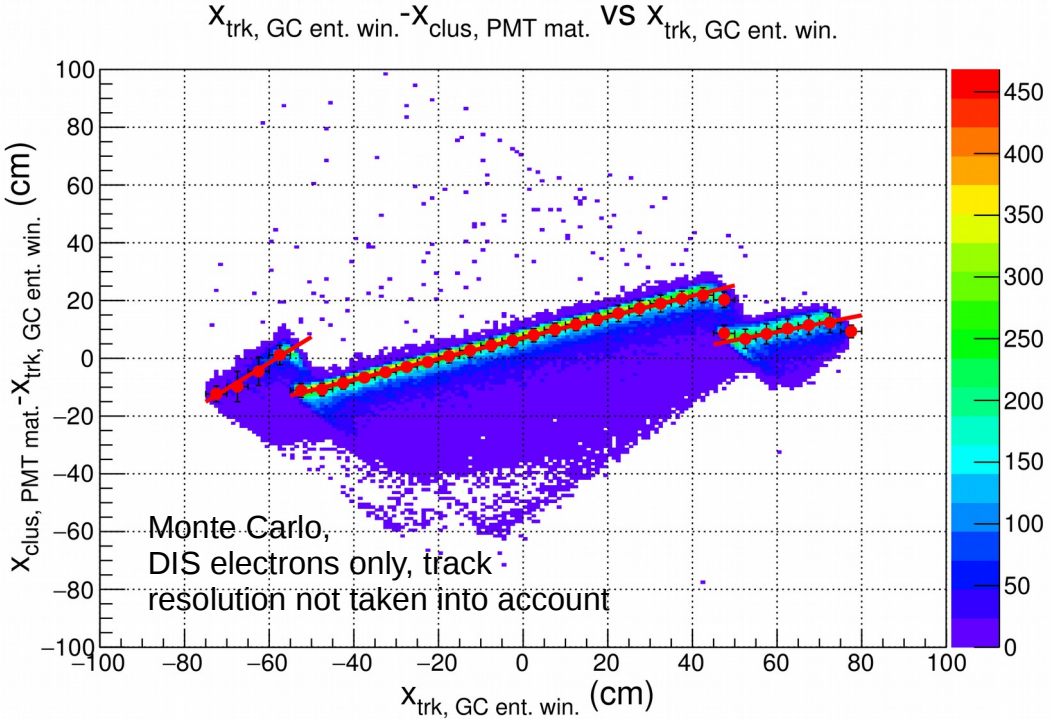
Interface library with SBS-offline
In progress...

Data analysis steps for GRINCH (under development) **:**

After clustering :

- if there are reconstructed tracks, they may be matched to the GRINCH clusters, using the exisiting correlation between a cluster position in the PMT matrix, vs the track position projected on the Cherenkov entrance window (*started*);





UCONN Jefferson Lab

Summary on digitization library:

*** The digitization part itself is pretty much done ;**
there will likely be some "minors" changes, but the frame is here, and it is functional;

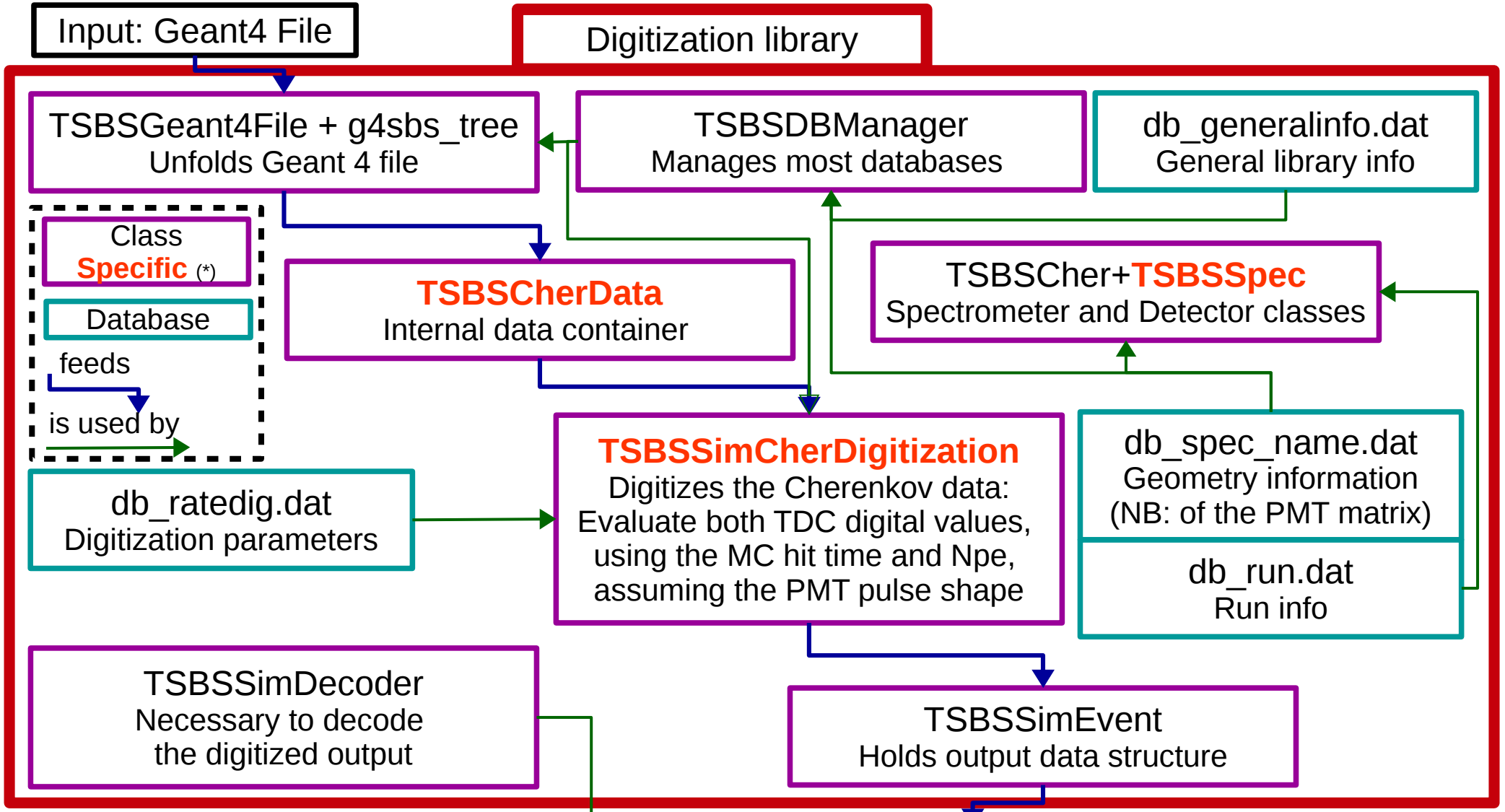
*** data decoding and processing is under development in SBS-offline;**
writing custom SimDecoder is pending on that (i.e. no real progress)...
(also, I created my own temporary git branch for that; once all changes are agreed on, this branch will be merged to the master and disappear);
I would need a DB at least to test it out (I've already asked Carlos, Evan) ;

TODO :
* merge libsbscer with libsbsgem (to form libsbsdig ?);

* Add and digitize background;

* Add pedestal noise, cross talk effects;

Cherenkov Digitization library:
Class structure => How to merge with libsbsgem

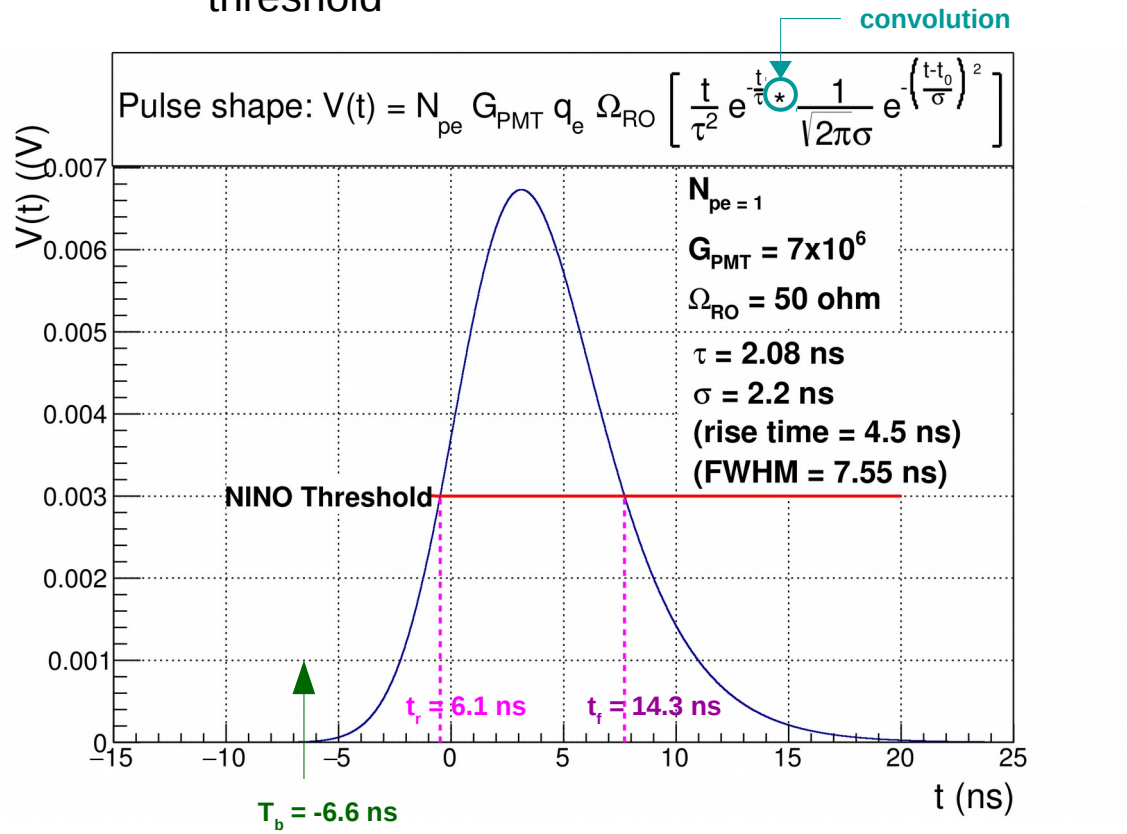


(*) : may inherit, but a template must be written

Backup

GRINCH PMT pulse shape

To evaluate the times t_r and t_f at which the pulse will go over/under the threshold, we need to know the pulse shape and its amplitude (see below);
The goal is to retrieve the GRINCH PMT timing characteristics (right panel), which has required the *convolution* of a exponentially decreasing function $\mathbf{t/\tau^2\ exp(-t/\tau)}$ with a gaussian.
Another issue is the normalization of this pulse, to evaluate at which moment it will cross the threshold



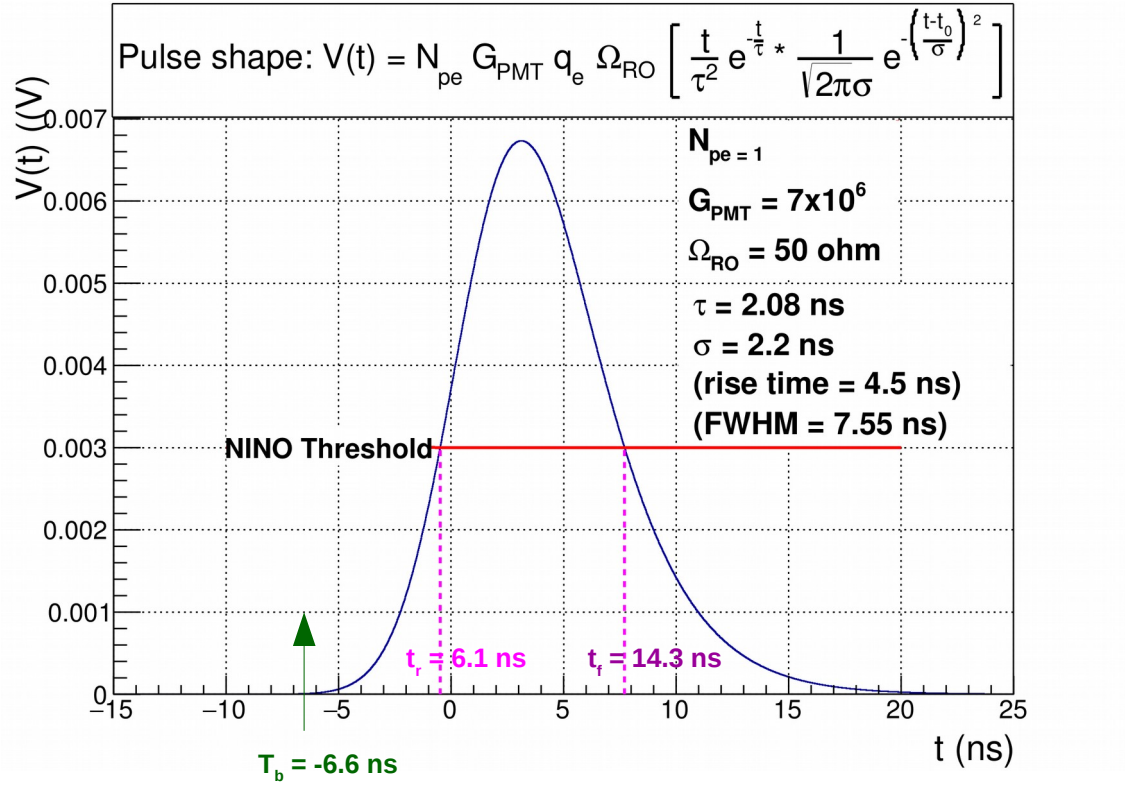
6 characteristics

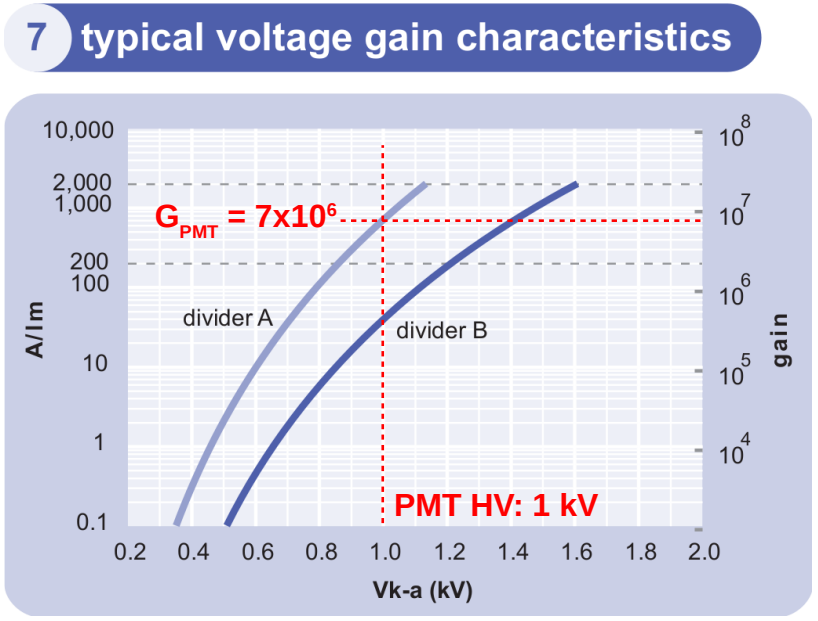
	unit	min	typ	max
timing:				
single electron rise time	ns		4.5	
single electron (fwhm)	ns		7.5	
single electron jitter (fwhm)	ns		4	
transit time	ns		33	

UCONN Jefferson Lab

GRINCH PMT pulse normalization

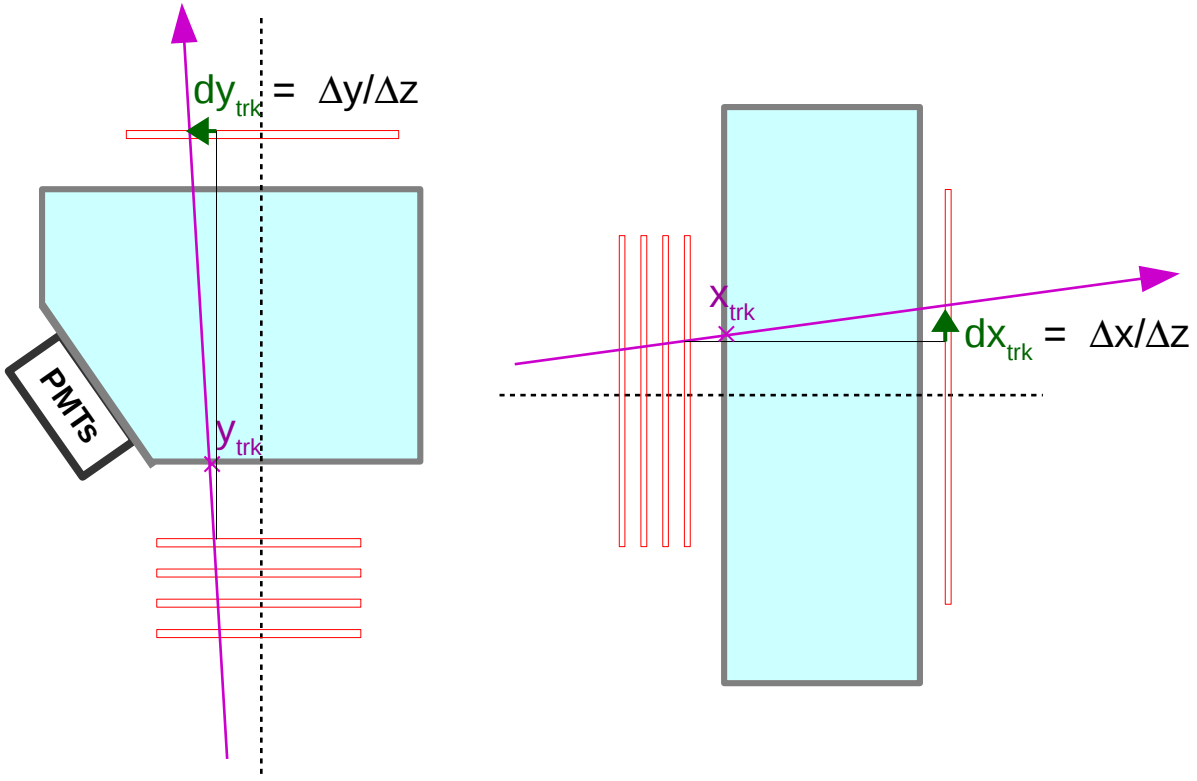
The pulse integral contains N_{pe} G_{PMT} electrons, or N_{pe} G_{PMT} q_e coulombs.
Hence, the pulse function shall be an intensity function I(t), which just needs to be multiplied by the readout impedance Ω_{RO} to obtain a voltage function V(t).
PS : yes, my scale is in ns, and I have taken this into account in the normalization.
I get a similar pulse height if I use scale in seconds (instead of ns).

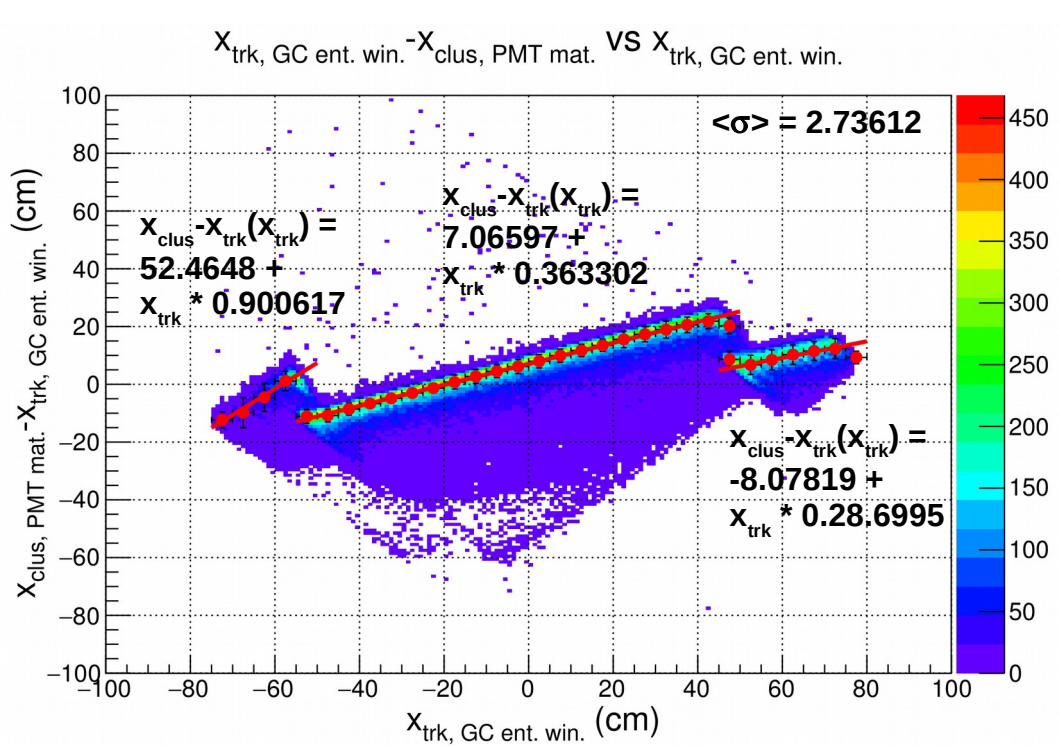


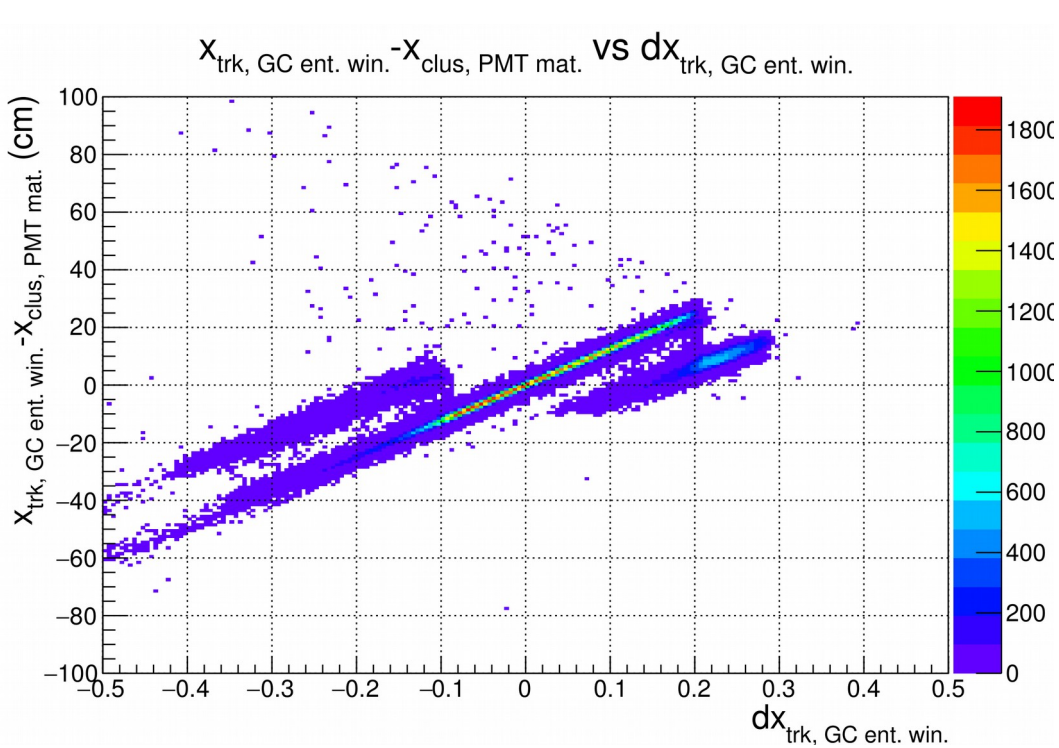


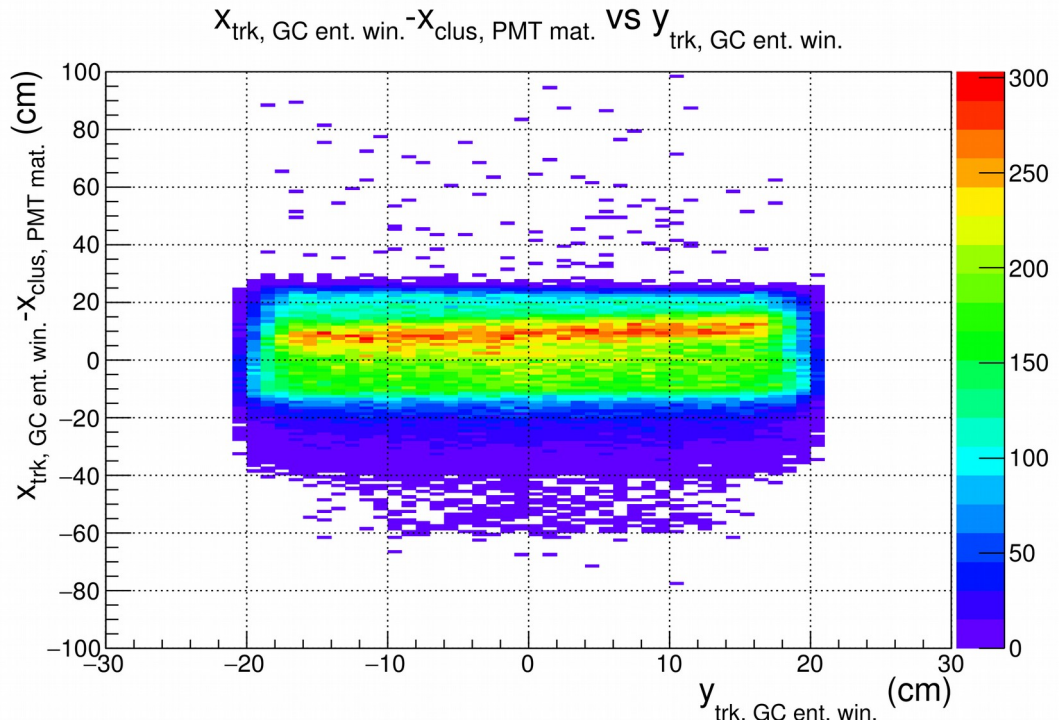
If this is normalization is correct, then it means the threshold should be lower (2-3 mV).

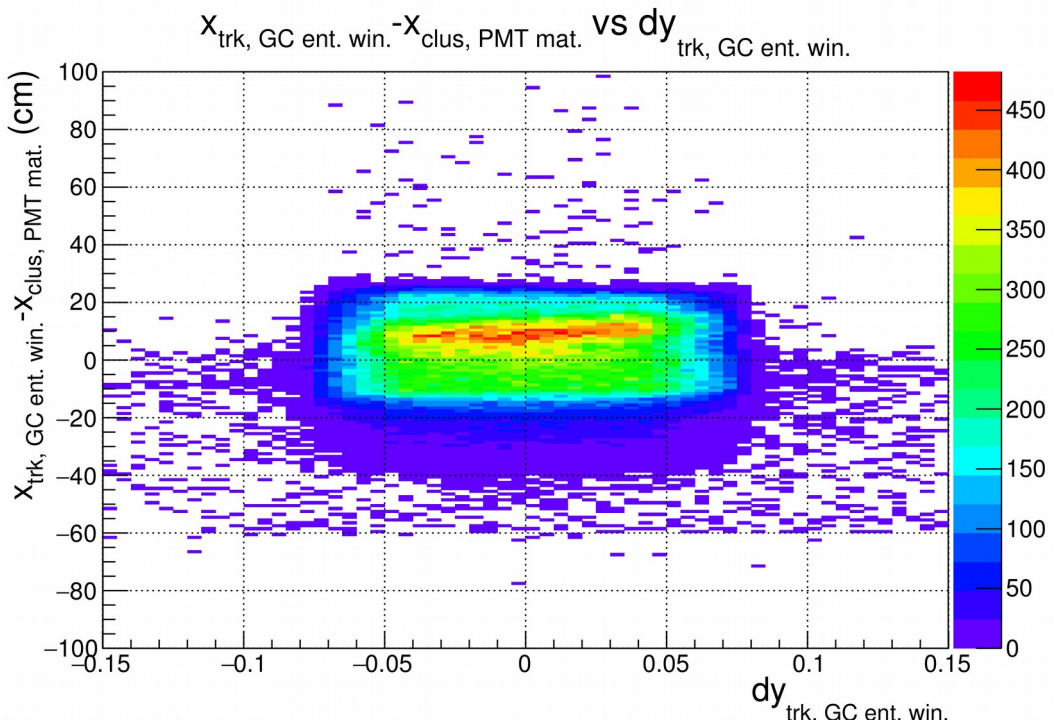
Definition of x_{trk}, y_{trk}, dx_{trk}, dy_{trk}

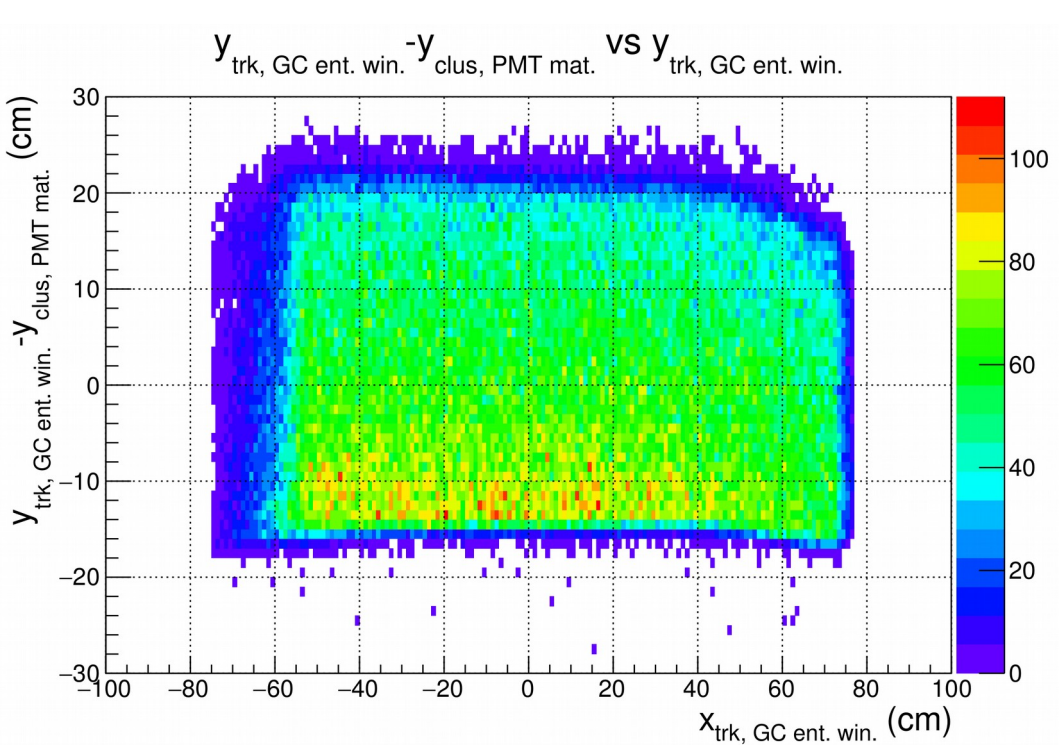


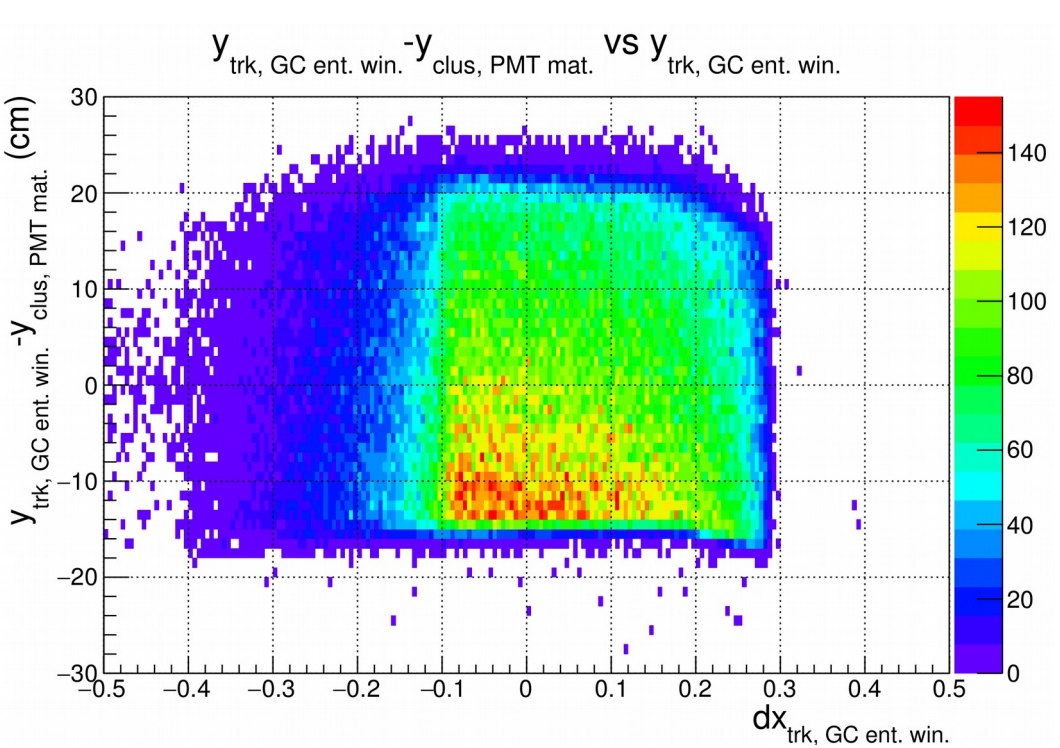

x_trk, GC ent. win.-x_clus, PMT mat. vs x_trk, GC ent. win.
<σ> = 2.73612
x_clus-x_trk(x_trk) = 52.4648 + x_trk * 0.900617
x_clus-x_trk(x_trk) = 7.06597 + x_trk * 0.363302
x_clus-x_trk(x_trk) = -8.07819 + x_trk * 0.28.6995
x_clus, PMT mat.-x_trk, GC ent. win. (cm)
x_trk, GC ent. win. (cm)

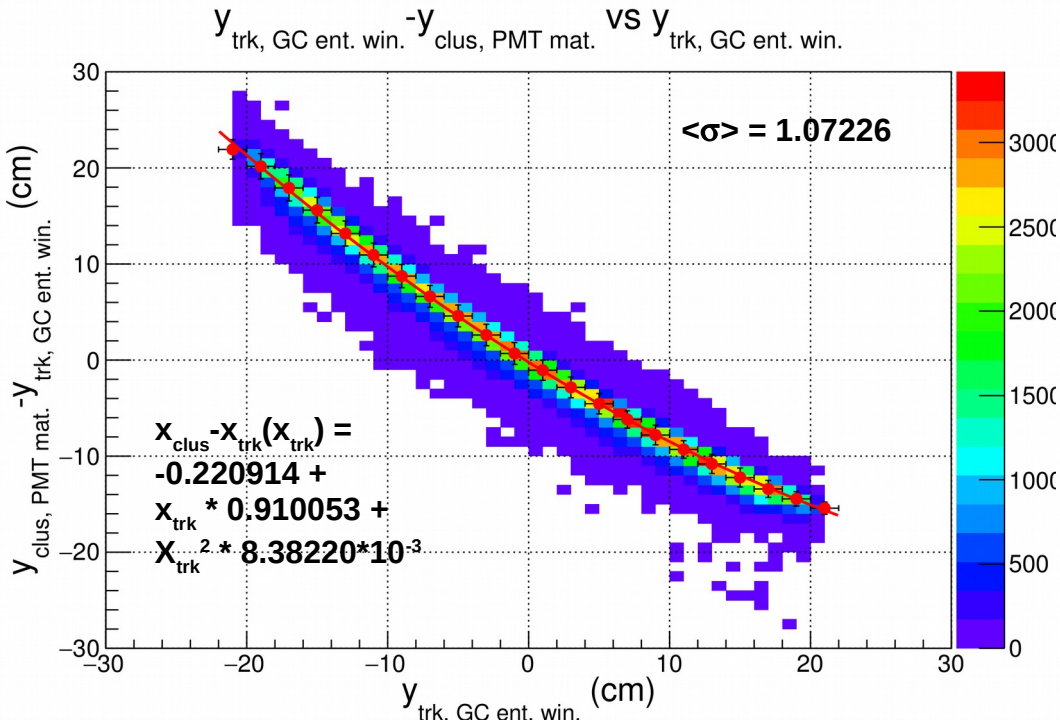

x_trk, GC ent. win.-x_clus, PMT mat. vs dx_trk, GC ent. win.
x_trk, GC ent. win.-x_clus, PMT mat. (cm)
dx_trk, GC ent. win.

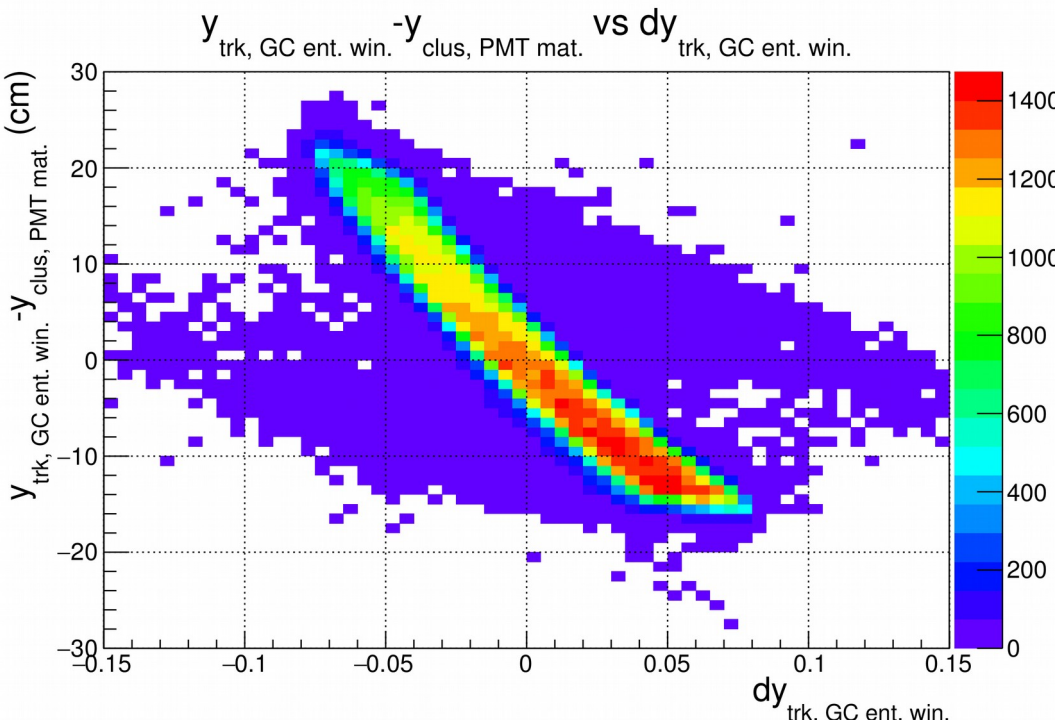

x_trk, GC ent. win.-x_clus, PMT mat. vs y_trk, GC ent. win.
x_trk, GC ent. win.-x_clus, PMT mat. (cm)
y_trk, GC ent. win. (cm)


x_trk, GC ent. win.-x_clus, PMT mat. vs dy_trk, GC ent. win.
x_trk, GC ent. win.-x_clus, PMT mat. (cm)
dy_trk, GC ent. win.


y trk, GC ent. win. -y clus, PMT mat. vs y trk, GC ent. win.
y trk, GC ent. win. -y clus, PMT mat. (cm)
x trk, GC ent. win. (cm)


y trk, GC ent. win. -y clus, PMT mat. vs y trk, GC ent. win.
y trk, GC ent. win. -y clus, PMT mat. (cm)
dx trk, GC ent. win.


y trk, GC ent. win. -y clus, PMT mat. vs y trk, GC ent. win.
<σ> = 1.07226
x clus -x trk (x trk) =
-0.220914 +
x trk * 0.910053 +
X trk 2 * 8.38220*10-3
y clus, PMT mat. -y trk, GC ent. win. (cm)
y trk, GC ent. win. (cm)


y trk, GC ent. win. -y clus, PMT mat. vs dy trk, GC ent. win.
y trk, GC ent. win. -y clus, PMT mat. (cm)
dy trk, GC ent. win.