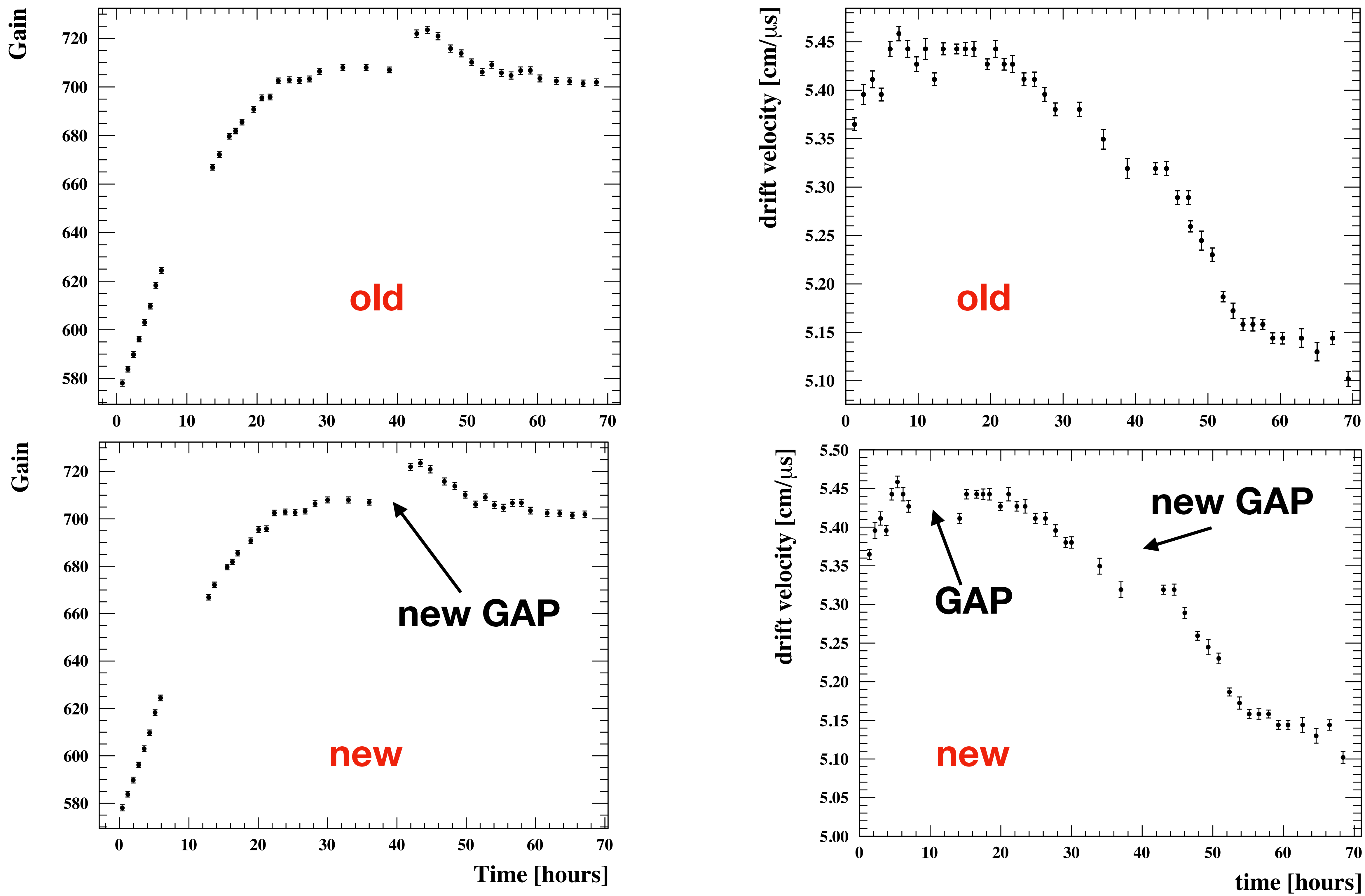
C. Jesús-Valls
cjesus@ifae.es

Group Meeting

Towards SFGD data reconstruction

We have 1 data point every 25k events



I hope to have today equivalent plot for electron attenuation length vs time and corrected dE/dx (with gain, and attenuation)

The intention of joining SFGD analysis team has been discussed with SFGD's conveners (Masashi and Yuri) —> Both were happy to hear about it.

What next?

- Waiting to have access to SFGD beam test data.
- Good moment to start thinking.

INPUT?

After proper data taking and decoding:

SFGD: Collection of hit cubes (x1,x2,x3) at (t1,t2,t3) with charge (p.e) Q.
HATPC: Collection of hit pixels (x1,x2) at (t1,t2) with charge Q.
ToF: Bars hit with very good time resolution. Q? if measured not with good resolution… (probably not relevant). It is posible to constrain track crossing point via timing on both sides.
Ecal: It is basically like SFGD with much larger modules.

OUTPUT?

Global clusters (across different detector modules) with orientation, and vertices.
Do we have to provide event ID???

HOW TO DO IT?

Reconstruct locally each track. (Tracks in HATPCs, in SFGD, in TPCs, in FGDs, in Ecal, psoition of hits in ToF…)
Currently is done for Ecal, TPCs, FGDs. Old way —> Is that a constrain?

Merge local tracks into global tracks. Use SFGD timing and ToF timing to set tracks direction. Associate tracks coming from a common vertex in Target, use FGDs, SFGDs positioning to reconstruct TPCs Z position. Extrapolate tracks and look for matching among different modules.

THE BEGINNING:

Having in mind the final goal is paramount to make the code smartly organized and focused from the beginning, adding as much helpful utilities as possible.

However, as initial step we want to find the best possible ML algorithm to reconstruct tracks with amazingly different shapes, with lots of backgrounds and do it fast and efficiently for SFGD.

2Million cubes mean a huge amount of input information to analyze. The best will be to have beam test data to start working, although **only** 10 thousand cube input was recorded.

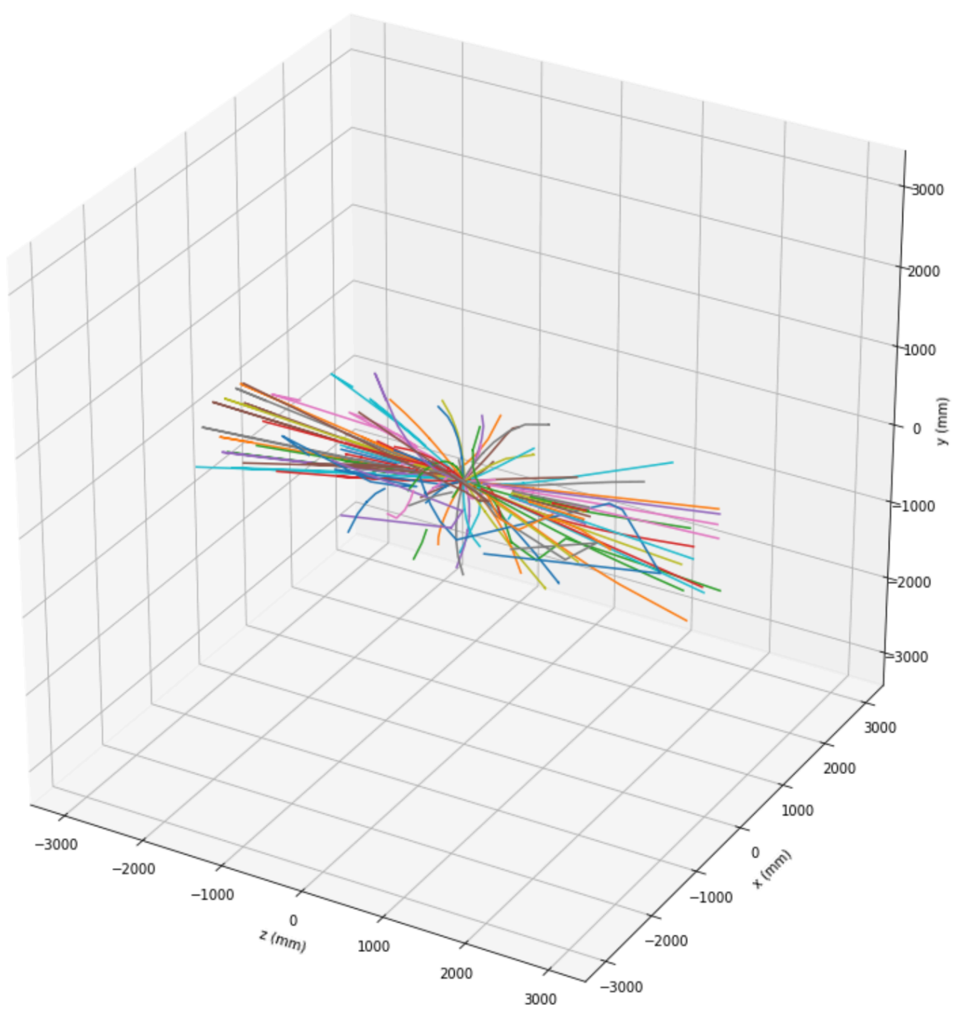
In a later stage MC information from current software can be generated to give as better training samples.

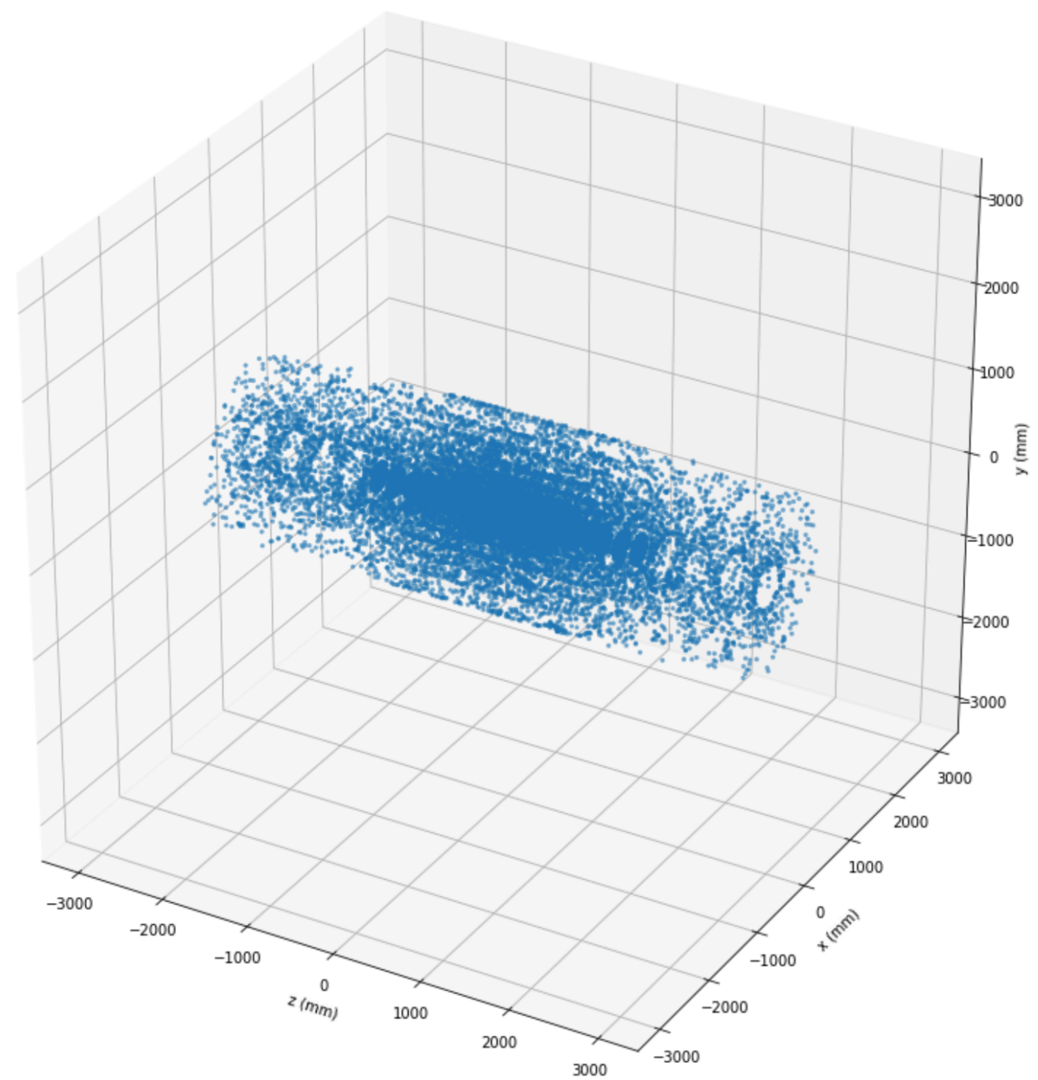
Kaggle

A nice possibility also is to look inside the ***TrackML Particle Tracking Challenge*** to get some initial ideas.

What is in there?

I have currently downloaded the stuff provided by the finished competition and is quite simple to download and run.





Input data contains truth data, and hits data.

In truth data you have all hits belonging to a track. The point were it starts, and the different modules and layers were it has been detected.
In hits information you only have the hits, were they have been detected, but no information about what is the real track that has generate it.
The challenge consist in to associate as efficiently as possible hits to a cluster matching a real track.

track

	hit_id	particle_id	tx	ty	tz	tpx	tpy	tpz	weight
0	1	324262265547128832	-69.290901	-0.794919	-1502.5	-0.511341	0.019336	-11.0690	0.000009
1236	1237	324262265547128832	-69.059402	-0.803144	-1497.5	-0.513453	0.017717	-11.0694	0.000008
1259	1260	324262265547128832	-60.033699	-1.070330	-1302.5	-0.511872	0.012358	-11.0696	0.000007
2656	2657	324262265547128832	-59.802502	-1.075990	-1297.5	-0.511703	0.012523	-11.0702	0.000005
2711	2712	324262265547128832	-50.804798	-1.245940	-1102.5	-0.510417	0.006837	-11.0703	0.000005
4429	4430	324262265547128832	-50.573898	-1.248620	-1097.5	-0.511414	0.005587	-11.0708	0.000005
4501	4502	324262265547128832	-44.333099	-1.294840	-962.5	-0.511562	0.001840	-11.0708	0.000005
6492	6493	324262265547128832	-44.101799	-1.295060	-957.5	-0.512197	0.000183	-11.0713	0.000006
6568	6569	324262265547128832	-37.857399	-1.273930	-822.5	-0.511894	-0.003675	-11.0714	0.000007
6624	6625	324262265547128832	-37.649601	-1.272390	-818.0	-0.510081	-0.003528	-11.0721	0.000008
8927	8928	324262265547128832	-37.626598	-1.272210	-817.5	-0.510332	-0.003844	-11.0722	0.000010
9006	9007	324262265547128832	-32.332401	-1.209780	-702.5	-0.509577	-0.007655	-11.0723	0.000012
9024	9025	324262265547128832	-32.125500	-1.206550	-698.0	-0.508518	-0.008908	-11.0729	0.000013
11361	11362	324262265547128832	-32.102600	-1.206140	-697.5	-0.508141	-0.009176	-11.0729	0.000015

	hit_id	x	y	z	volume_id	layer_id	module_id
0	1	-64.409897	-7.163700	-1502.5	7	2	1
1	2	-55.336102	0.635342	-1502.5	7	2	1
2	3	-83.830498	-1.143010	-1502.5	7	2	1
3	4	-96.109100	-8.241030	-1502.5	7	2	1
4	5	-62.673599	-9.371200	-1502.5	7	2	1

5 hits