DirectFit event reconstruction

Method of the fit: exhaustive search

- simulate cascade events with various x,y,z,θ,φ (and fit for E,t_0), and compare them to the data event.
- optionally for each new track simulate cascades of equal energy spaced along the track and solve for best combination (next slide)
- the simulated event that looks most like the data event is the result

Advantages:

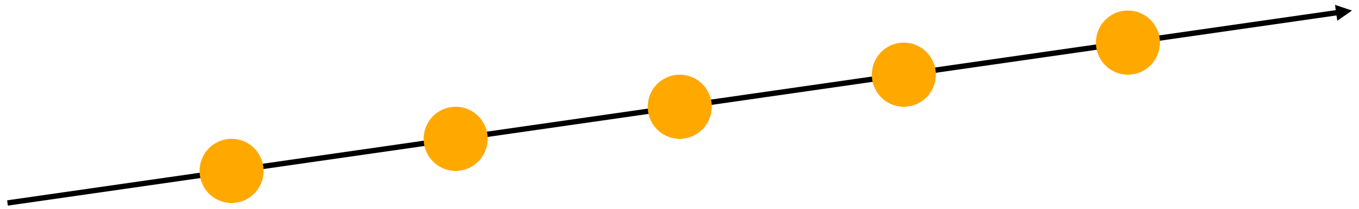
- simple and robust
- most precise description of ice can be used in reconstruction (SPICE Lea ice model: including tilt and anisotropy)

Drawbacks:

- can be very slow (~1 day/event)

Track reconstruction

For each new track given by x,y,z,θ,φ,t_0, simulate cascades of equal energy at equal intervals along the track.



Each cascade *i* creates hits in DOMs/time bins *j*: leaving charge Q_{ij}.

For the pattern of hits that we actually have in data, q_j, we can find the best representation of the event in terms of the simulated cascades by creating a weighted superposition (linear combination) of the cascades *i*.

This can be done by maximizing the limited-simulation-statistics likelihood (LSSL) against the weights, starting with an NNLS solution to $q_j=w_iQ_{ij}$.

This is done for different t_0 and the t_0 maximizing the likelihood is chosen.

Comparing a simulation event with a data event

Use the same likelihood function as in the SPICE Lea fit:

includes Poisson fluctuations in data, simulation, and a 20% allowance for non-Poisson errors (in description of ice and others).

All feature-extracted waveforms (charge histogram vs. time in a DOM) are binned in 25 ns bins, and then processed with a Bayesian blocks procedure, which combines low-count or nearly-same-charge bins.

This is the same procedure as was used in the SPICE Lea fit.

So, we are using *exactly the same* comparison procedure as was used in the *ice model* fit (here: SPICE Lea).

LSSL description

Suppose we repeat the measurement in data n_d times and in simulation n_s times. The μ_s and μ_d are the expectation mean values of counts per measurement in simulation and in data.

With the total count in the combined set of simulation and data is s + d , the conditional probability distribution function of observing s simulation and d data counts is

$$P(\mu_s, \mu_d; s, d|s+d) = \frac{(s+d)!}{s! \cdot d!} \cdot \left(\frac{n_s\mu_s}{s+d}\right)^s \cdot \left(\frac{n_d\mu_d}{s+d}\right)^d$$

There is an obvious constraint

$$n_s\mu_s + n_d\mu_d = s+d$$

which can be derived, e.g., from the normalization condition

$$\sum_{s,d} P(\mu_s, \mu_d; s, d|s+d) = \left(\frac{n_s\mu_s}{s+d} + \frac{n_d\mu_d}{s+d}\right)^{s+d} = 1$$

Two hypotheses:

If data data and simulation are unrelated and completely independent from each other, then we can maximize the likelihood for μ_s and μ_d independently, which with the above constraint yields

$$\mu_s = \frac{s}{n_s}, \quad \mu_d = \frac{d}{n_d}$$

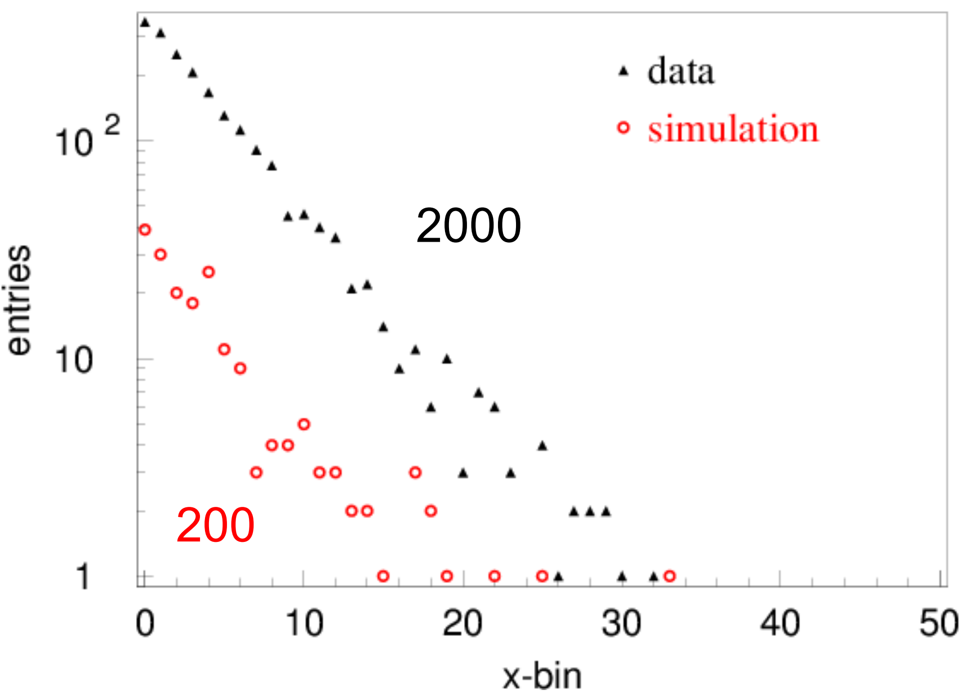
On the other hand, we can assume that data and simulation come from the same process, i.e.,

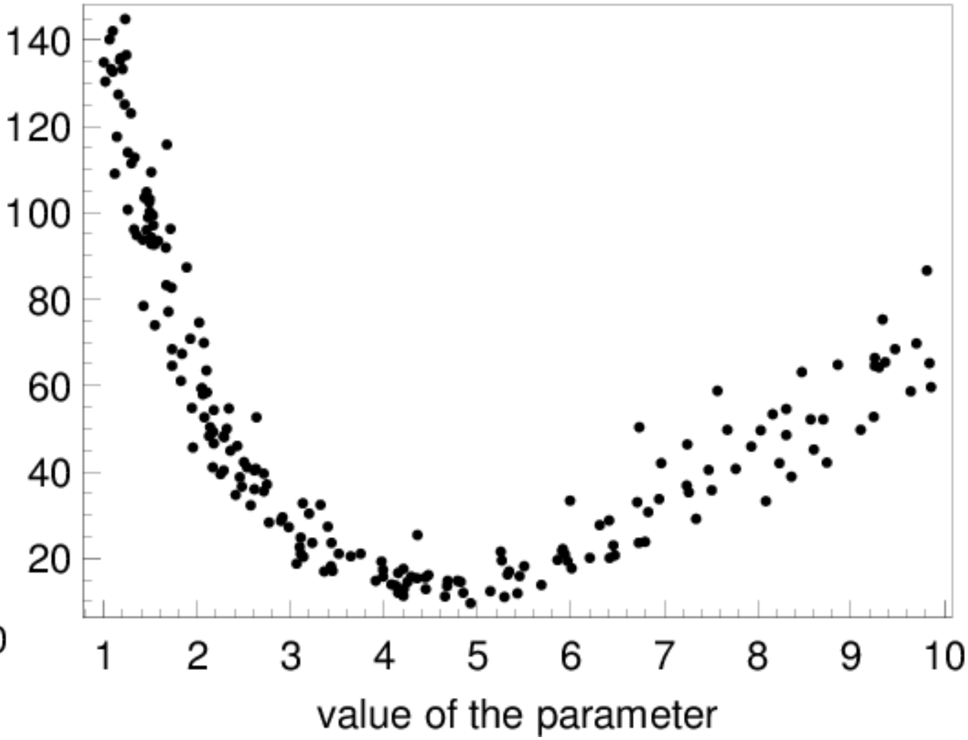
$$\mu = \mu_s = \mu_d = \frac{s+d}{n_s+n_d}$$

We can compare the two hypotheses by forming a likelihood ratio

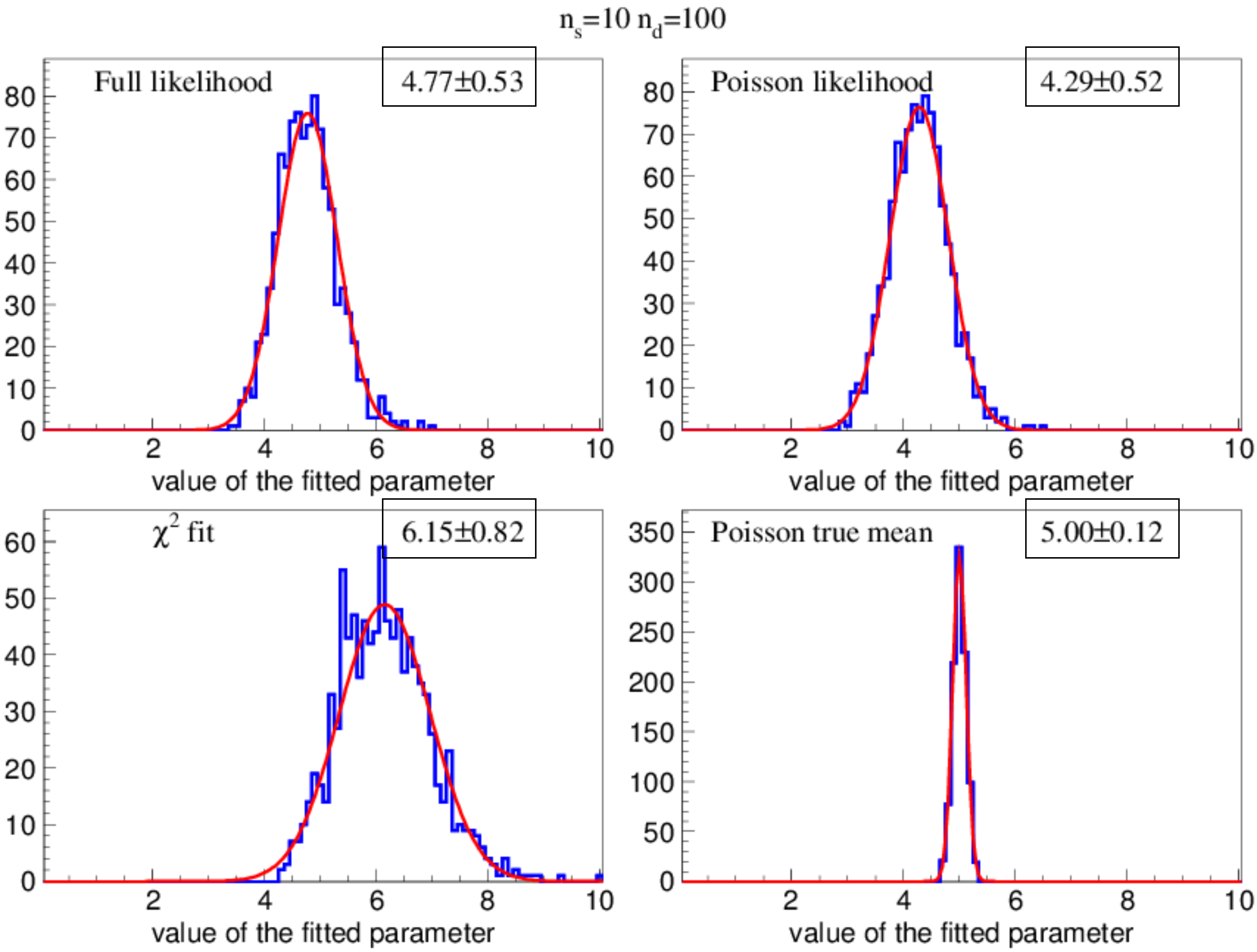
$$\frac{P(\text{same process})}{P(\text{independent processes})} = \left(\frac{n_s}{n_s+n_d} / \frac{s}{s+d}\right)^s \cdot \left(\frac{n_d}{n_s+n_d} / \frac{s}{s+d}\right)^d = \left(\frac{\mu}{s/n_s}\right)^s \cdot \left(\frac{\mu}{d/n_d}\right)^d$$

Example





To enhance the differences between the two likelihood approaches, consider that the amount of simulation is only 1/10th of that of data



Using full range of the data and simulation

Simulated exp(-x/5.0) with mean of 5.0

Search algorithm

1. Start with x,y,z of COG, θ=0, φ=0, E=10^5 GeV, t_0=0

2. Propose 25 sets of cascade parameters x,y,z,θ,φ from a gaussian distribution with rms=10 m in x,y,z and rms=30 degrees in θ,φ. Keep the values of E and t_0.

3. For each proposed simulated event find the best E (by scaling the simulated event) and t_0 (by time-shifting the hits in the simulated event); calculate the likelihood L.

4. Out of these 25 event select the one with the best value of L and update the 7 cascade parameters; remember the best value of L: L^*.

5. Repeat steps 2-4 40 times. Use 20 events resulting from step 4 with the best values of L^* to update the rms in x,y,z, and rms in θ,φ, and to establish correlation between these (important since the brightest point of the cascade is some distance away from the starting point along the cascade direction; also the Cherenkov light is emitted predominantly forward).

6. Repeat steps 2-5 10 times; The final result is calculated by averaging simulated events with the best 160 values of L^*. The rms of x,y,z, and the rms in θ,φ are also computed to provide a measure of uncertainties.

Other search algorithms and uncertainties

The algorithm described on the previous slide is an optimized variant of

- *Localized random search*.

Other methods that I tried are:

- *Simultaneous perturbation stochastic approximation* with and without the estimate of the second derivative (Newton-like method).

- *Markov chain* with transitional probability defined by condition $L_{i+1}<L_i$.

Although the rms values in cascade parameters obtained in the *localized random search* and *Markov chain* methods are probably related to the uncertainties of the measurement, the well-defined values of the uncertainties should probably be calculated by applying the reconstruction to a few (dozen?) cascade events simulated with the same parameters.

Uncertainties with ABC

ABC (Approximate Bayesian Calculation) solves for an approximation to the posterior PDF when the likelihood function is not known or its calculation is intractable.

We need a distance (in this case LSSL comparing simulation sets with data) and consider steps sampled from a proposal distribution which result in the distance smaller than a pre-set upper bound. All such steps are accepted.

This is a reversible Markov Chain, with a stationary distribution being the posterior parameter PDF for events similar to the data event with LSSL<bound. This approximates the parameter PDF for the actual given data event.

Statistical sampling is possible (performed for Bert). It is unclear if the systematical uncertainties can be included in this sampling procedure (due to curse of dimensionality).

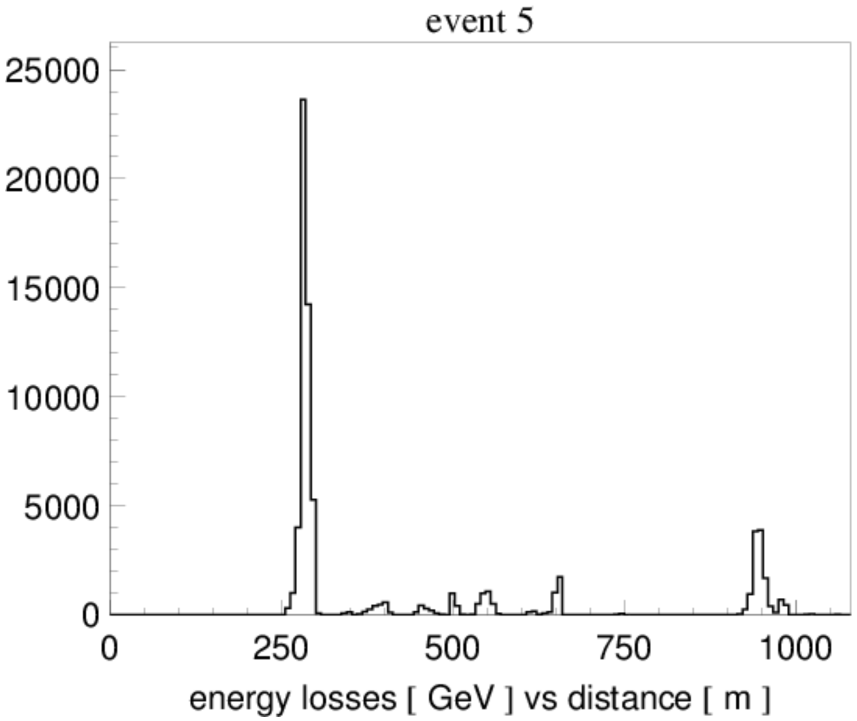
Track reconstruction in 28 HE events

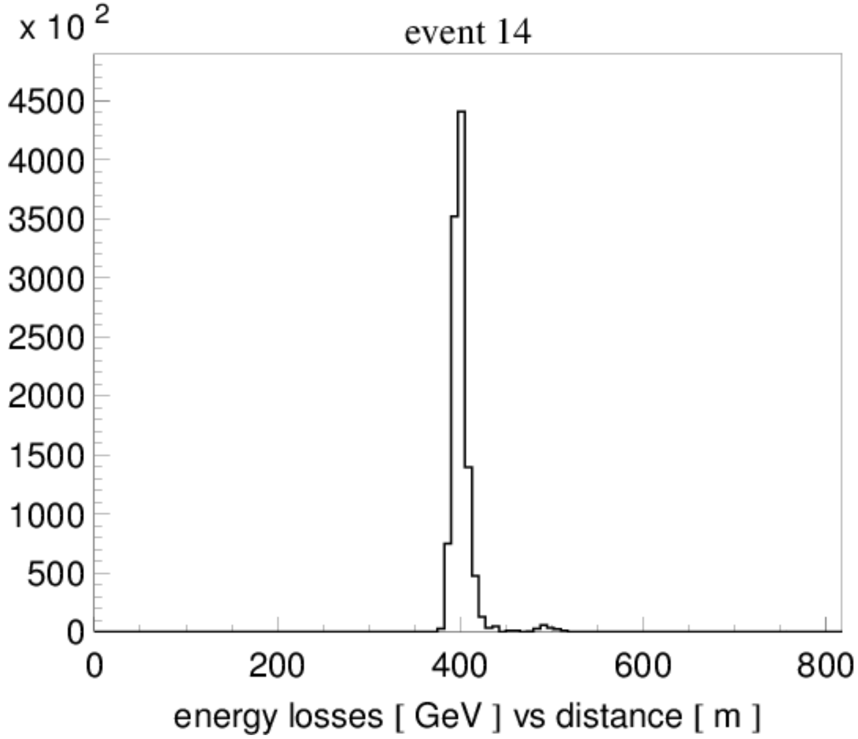
Only 1 event out of 28 was reconstructed with the reconstructed track going through the hits left by the muon.

In the other 6 events containing a track along with the interaction cascade the contribution to the likelihood from the smaller losses along the track are “washed out” by the fluctuations in the large contribution from the interaction cascade.

→ this results in track missing the smaller hits left by the muon.
→ possibly solved by over-simulating, however a factor x10 did not help (although only tried on the first 3 muon events)

Loss pattern along track

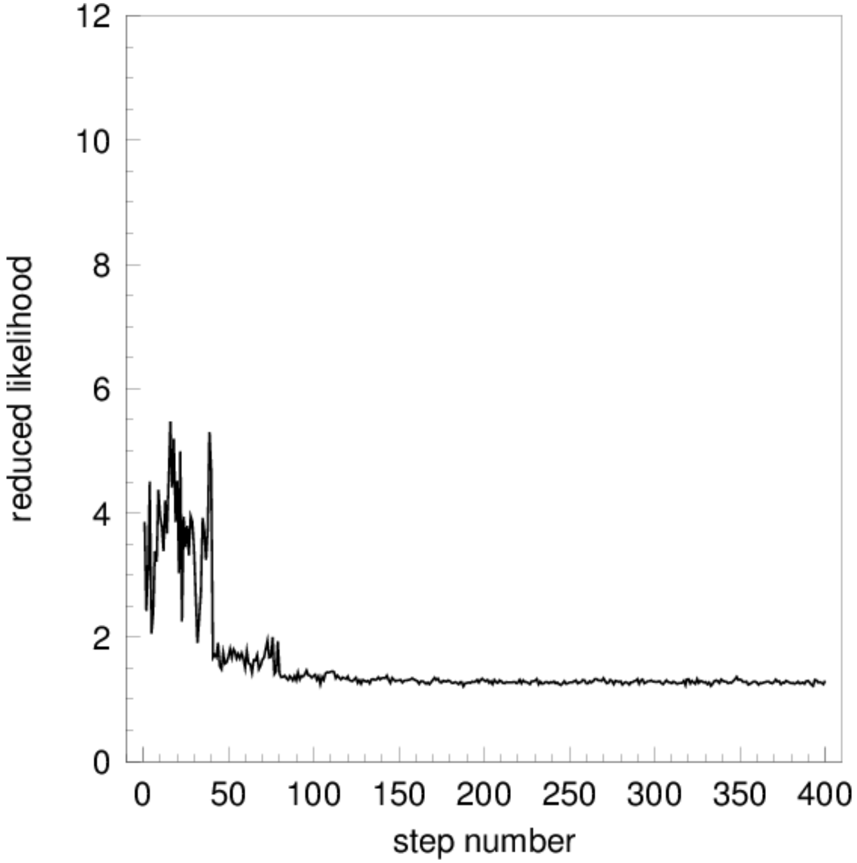




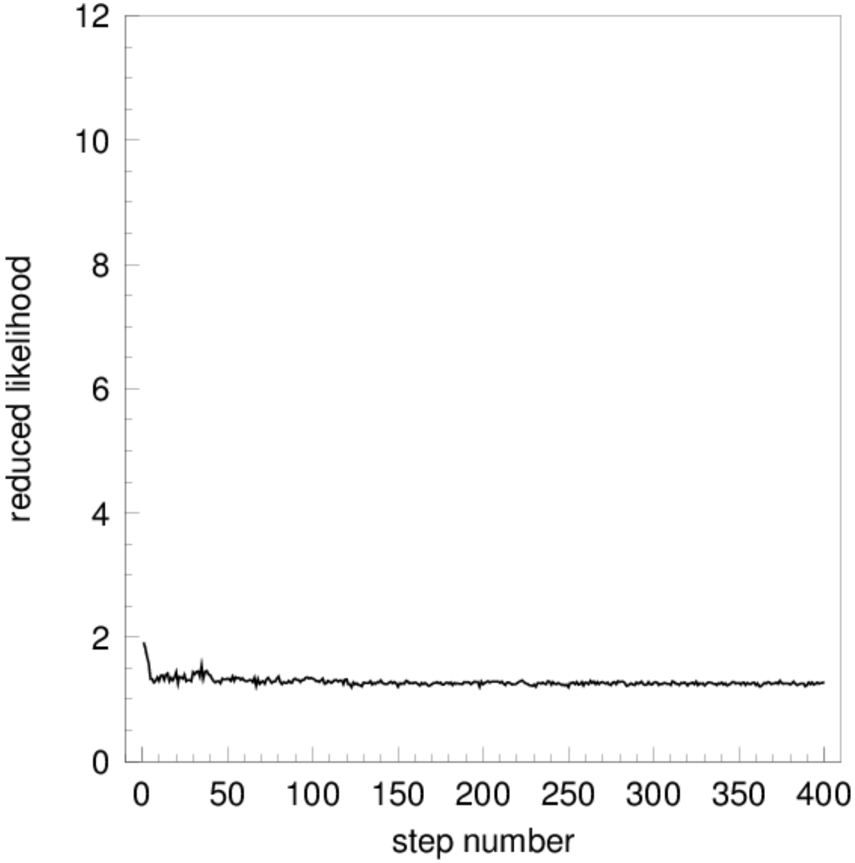
Dr. Strangepork

Bert

llh vs. step number

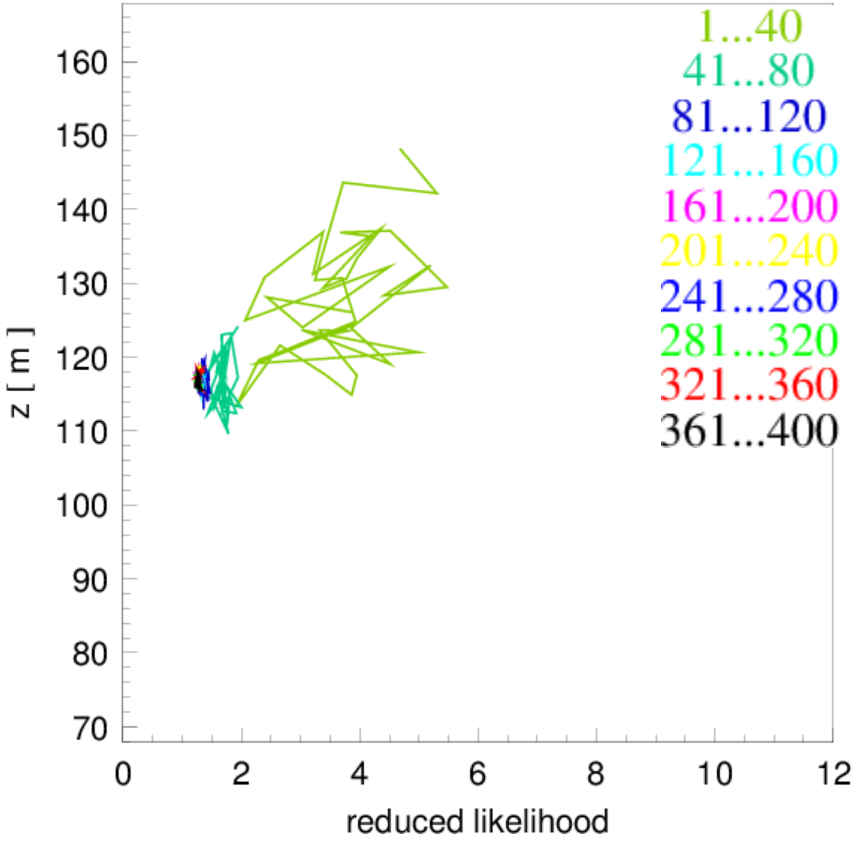


Dr. Strangepork

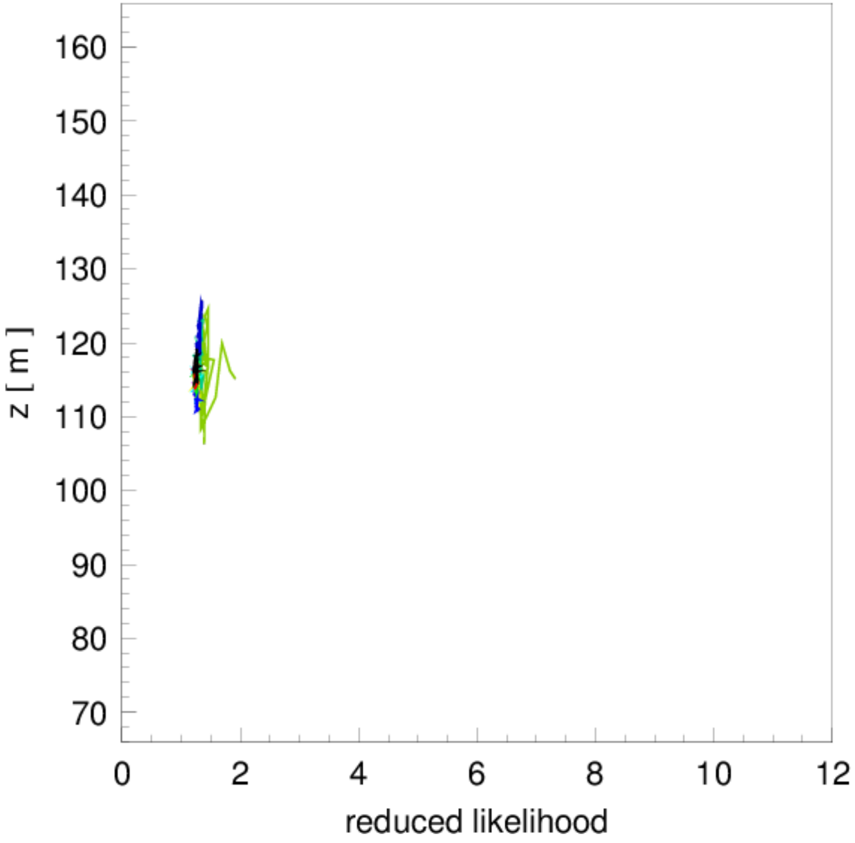


Bert

z vs. llh



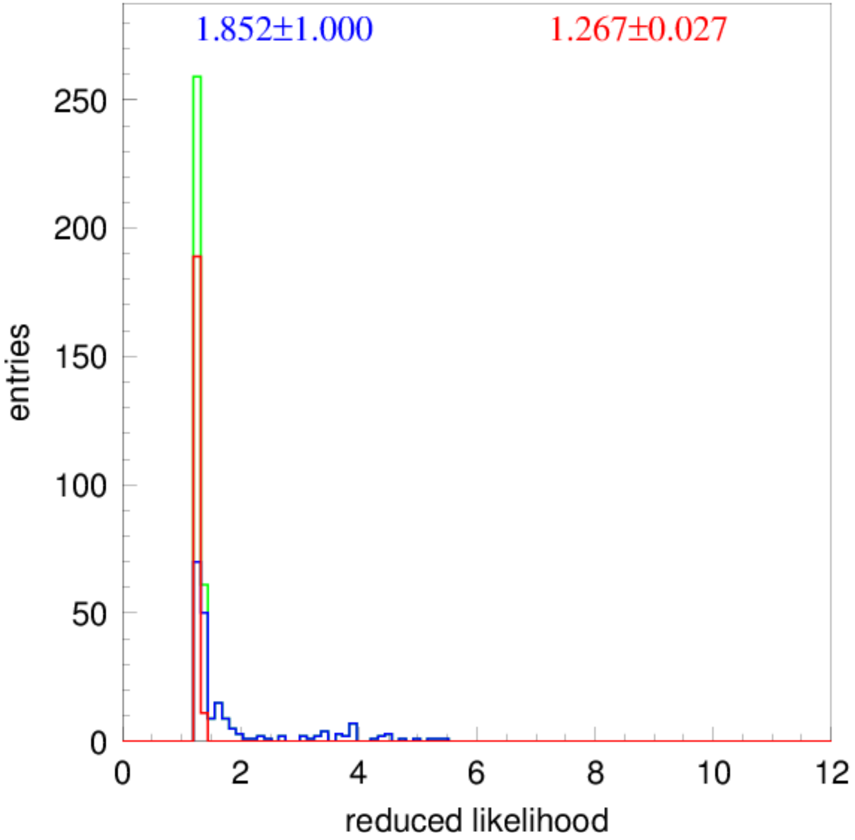
Dr. Strangepork



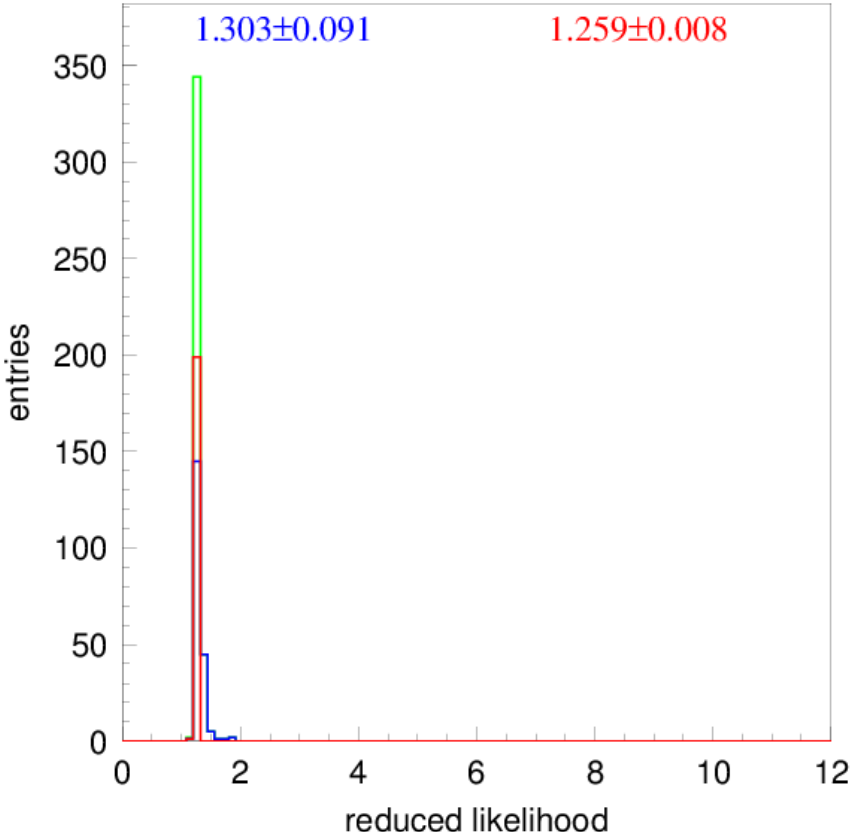
Bert

llh

1…200 all 201…400



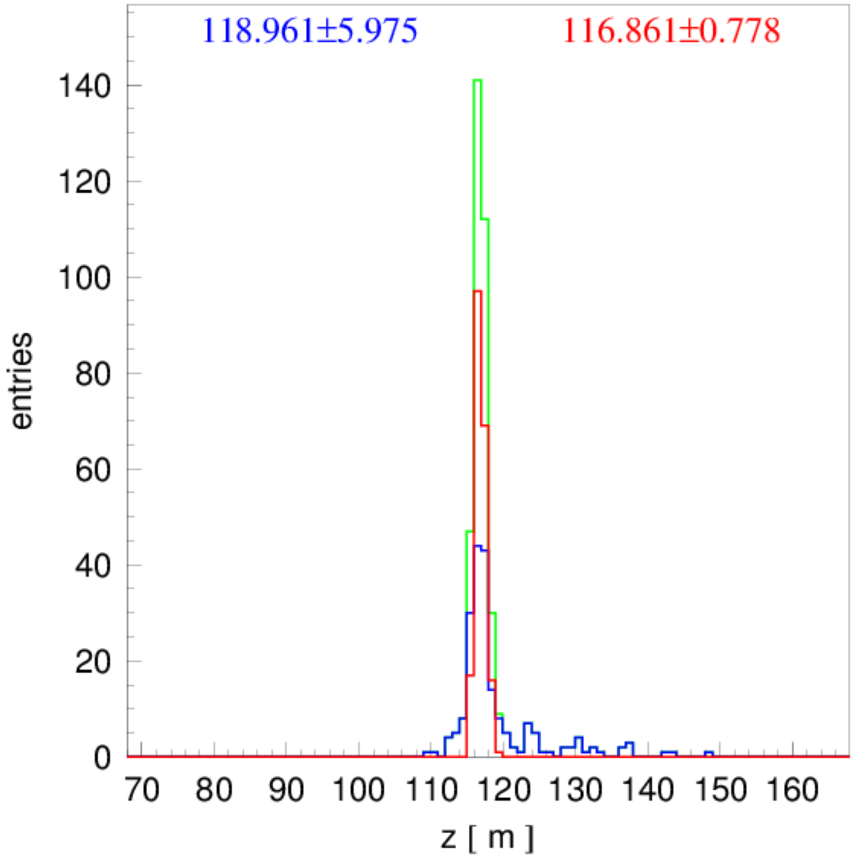
Dr. Strangepork



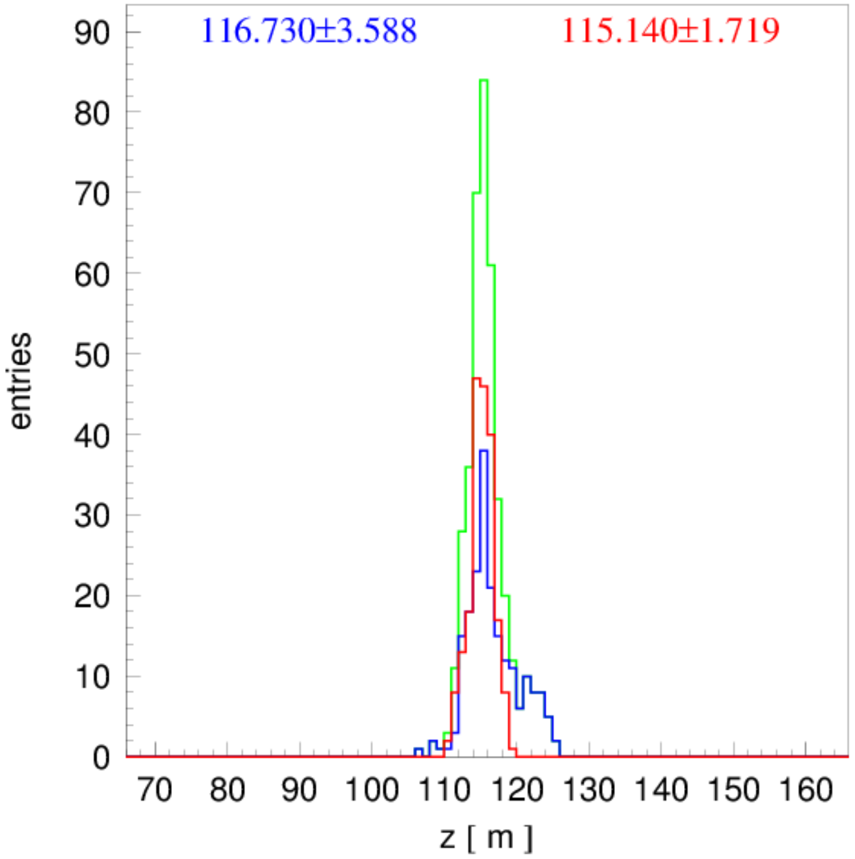
Bert

z

1…200 all 201…400

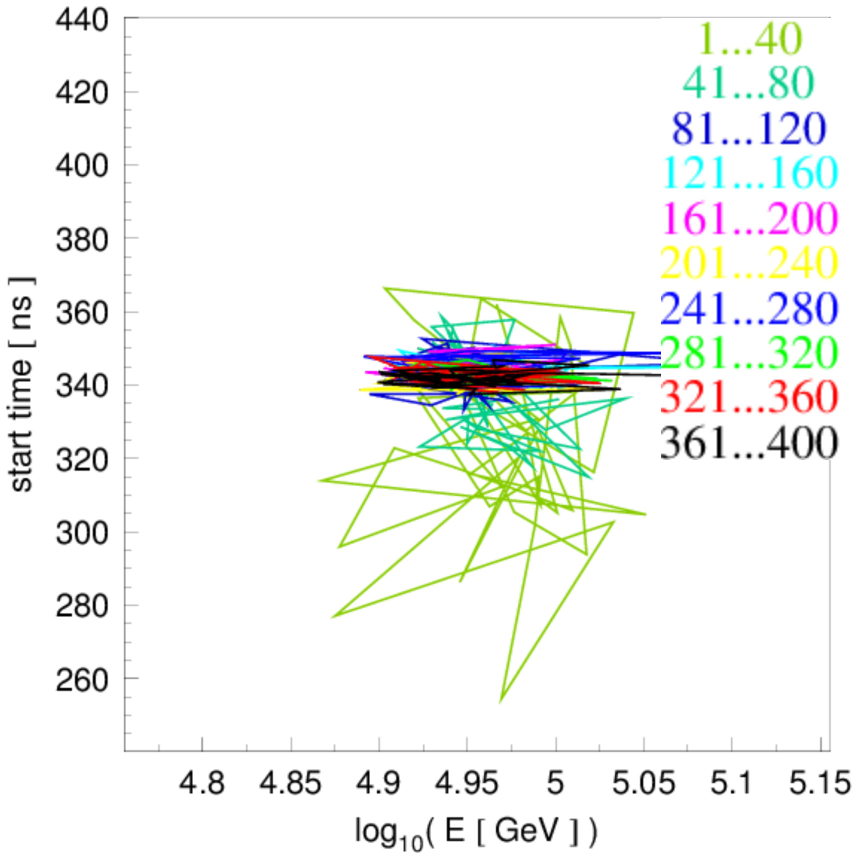


Dr. Strangepork

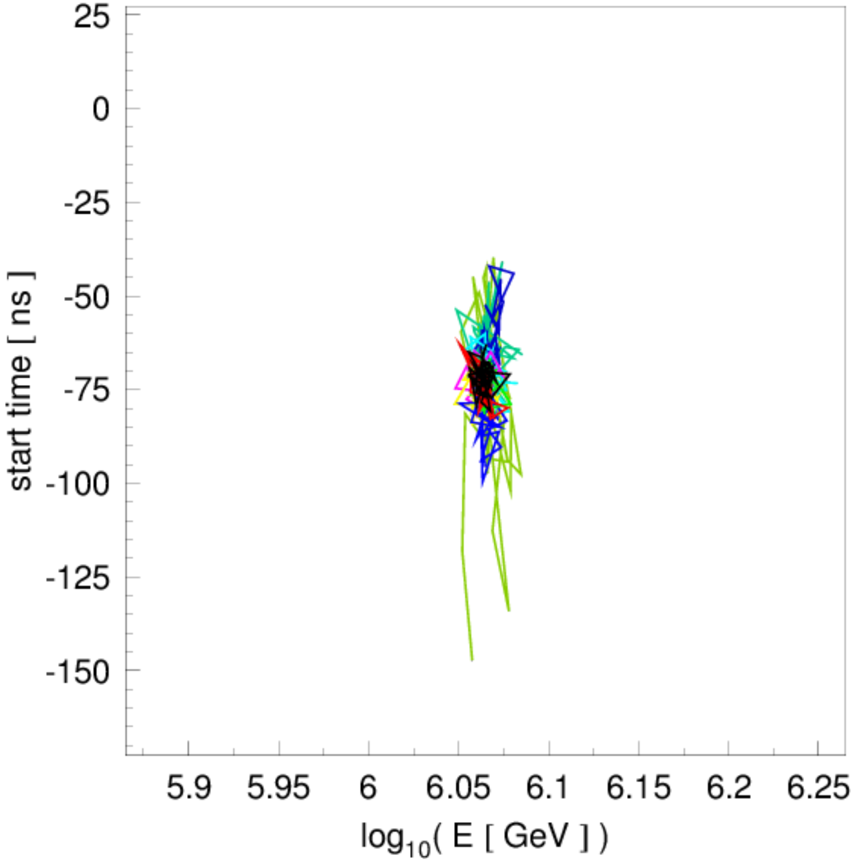


Bert

t_0 vs. E

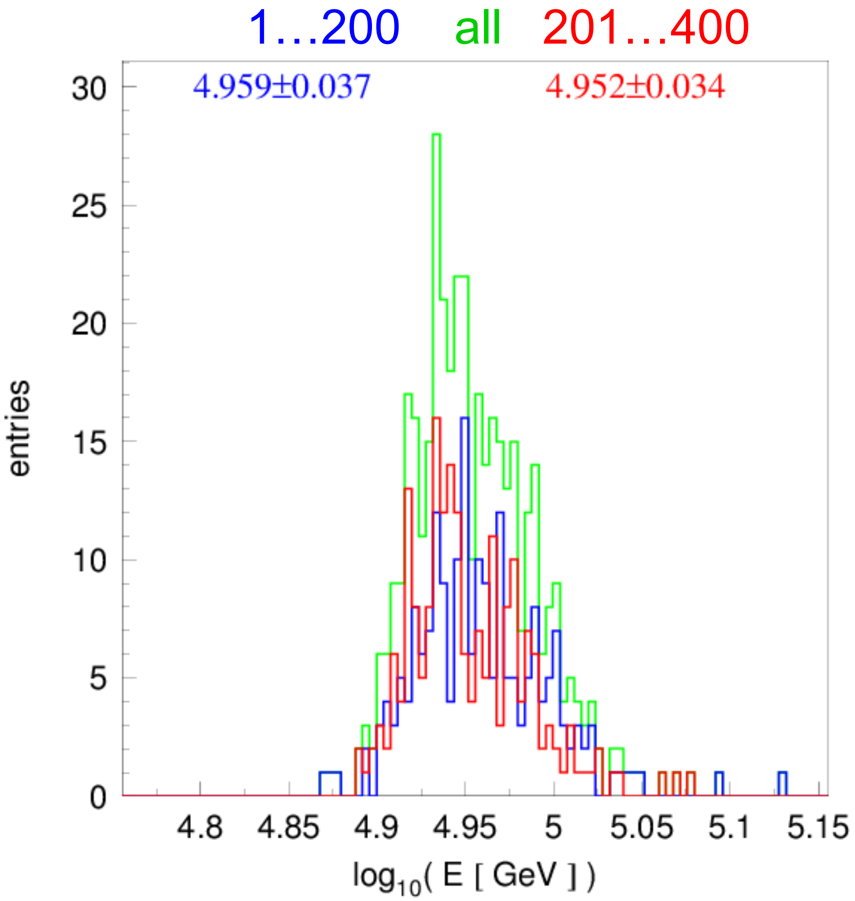


Dr. Strangepork

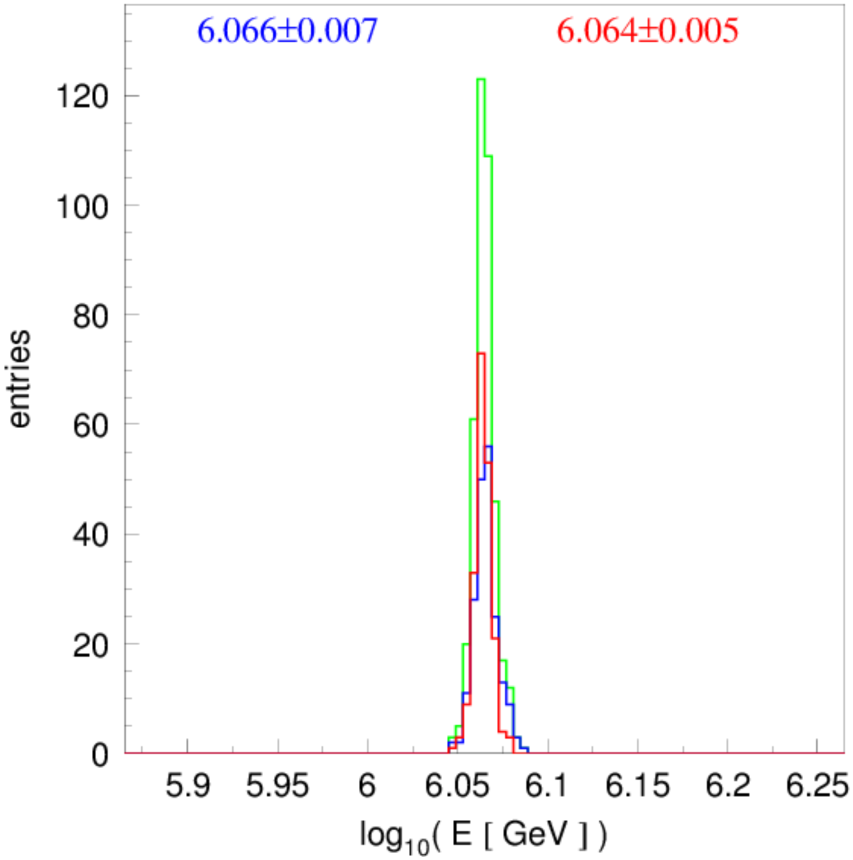


Bert

E



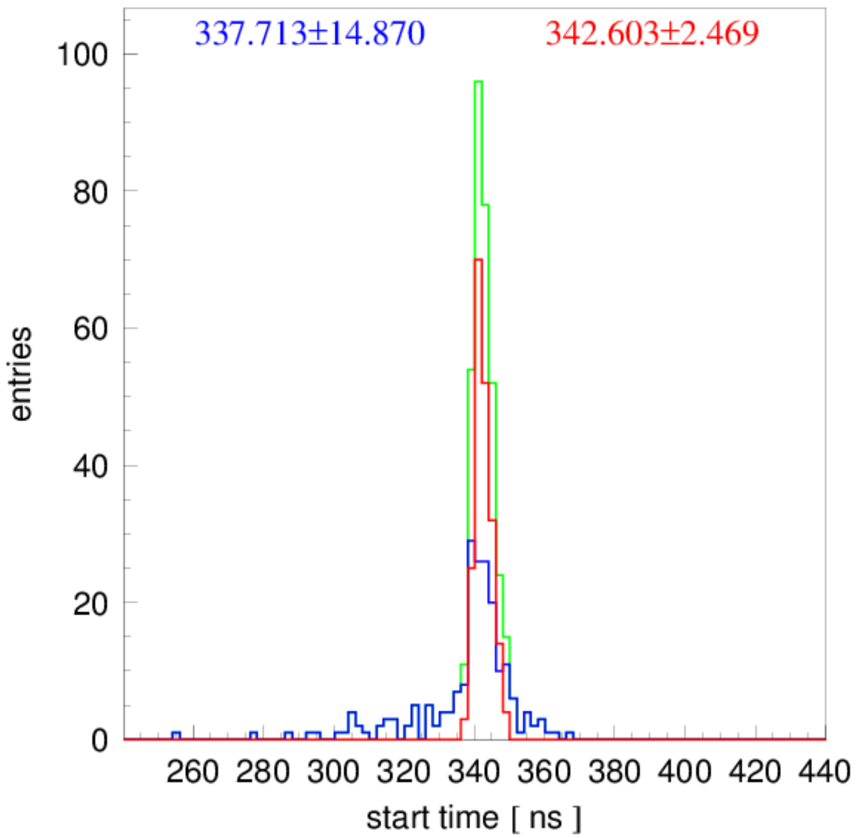
Dr. Strangepork



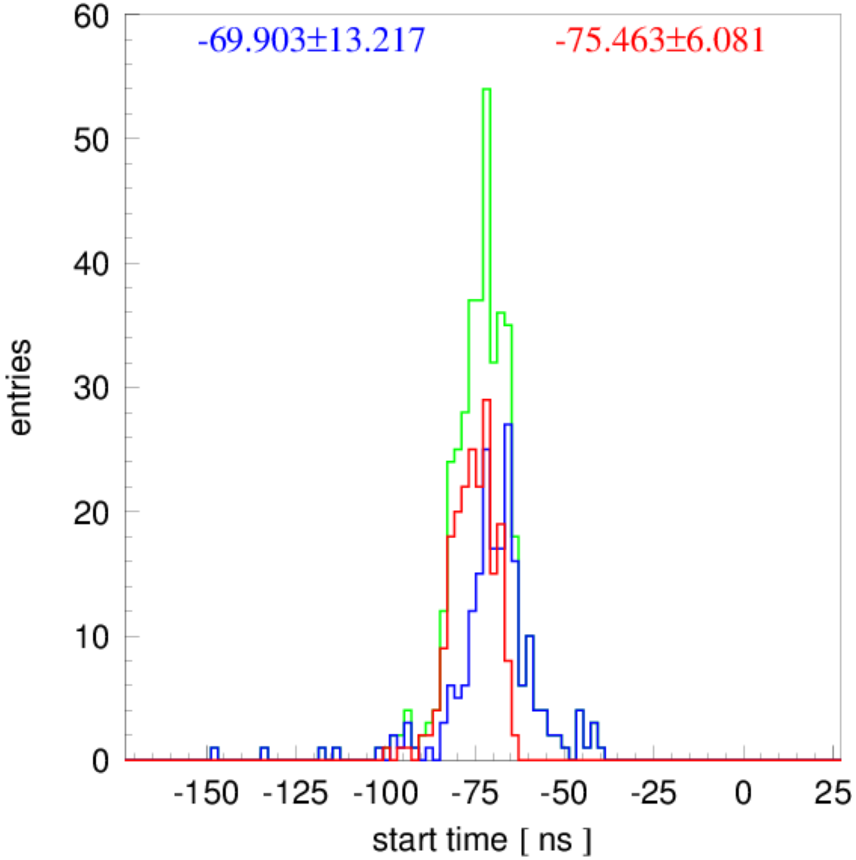
Bert

t_0

1…200 all 201…400

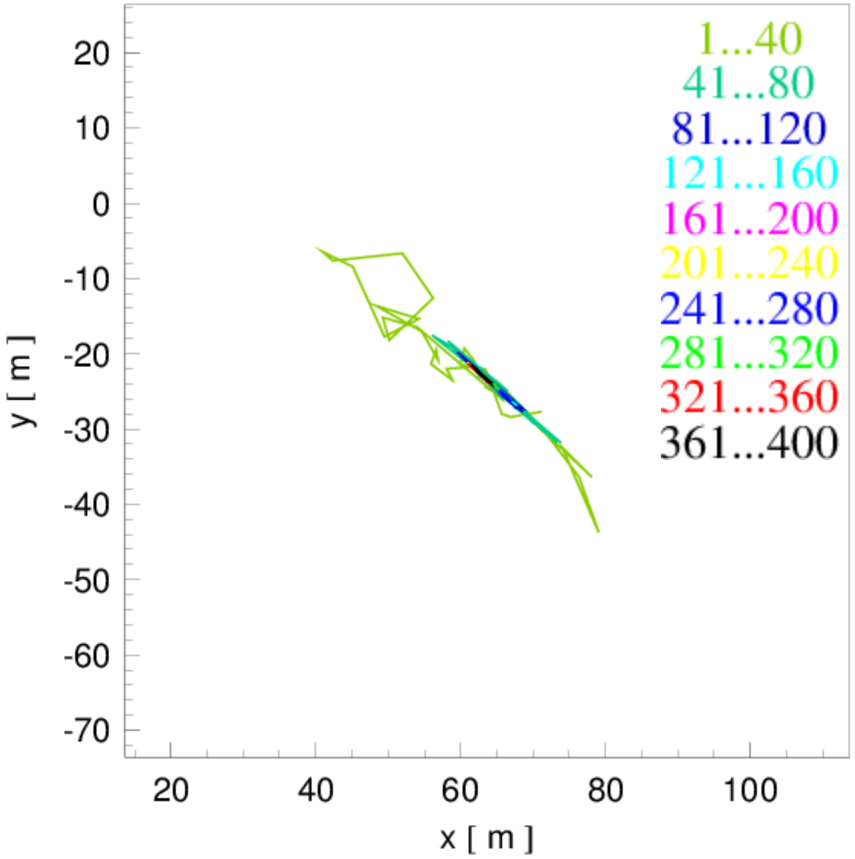


Dr. Strangepork

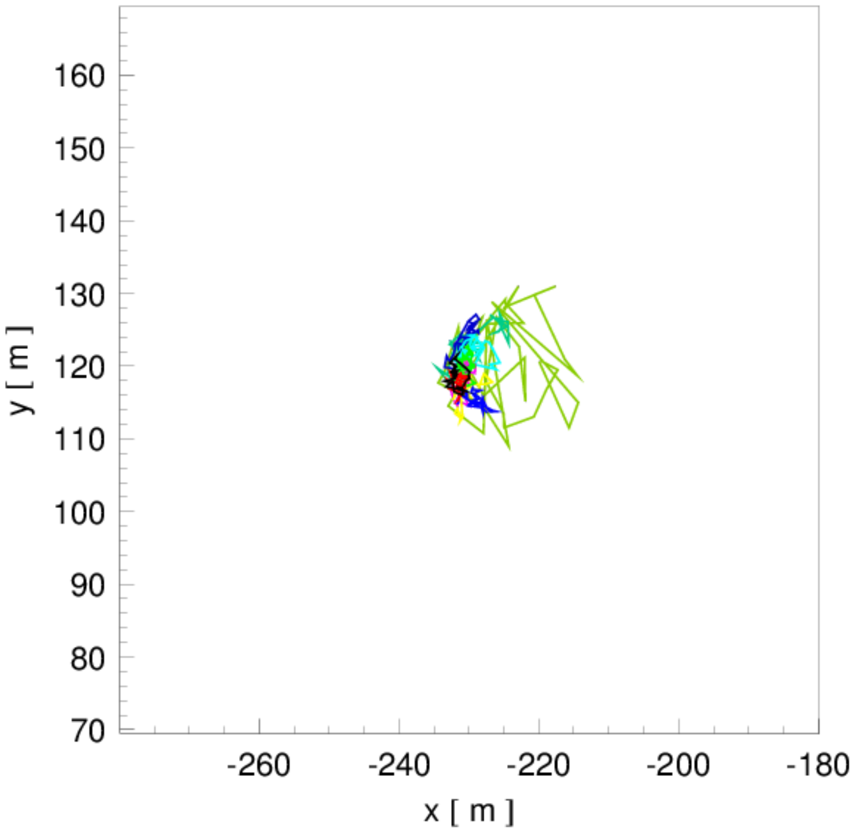


Bert

y vs. x

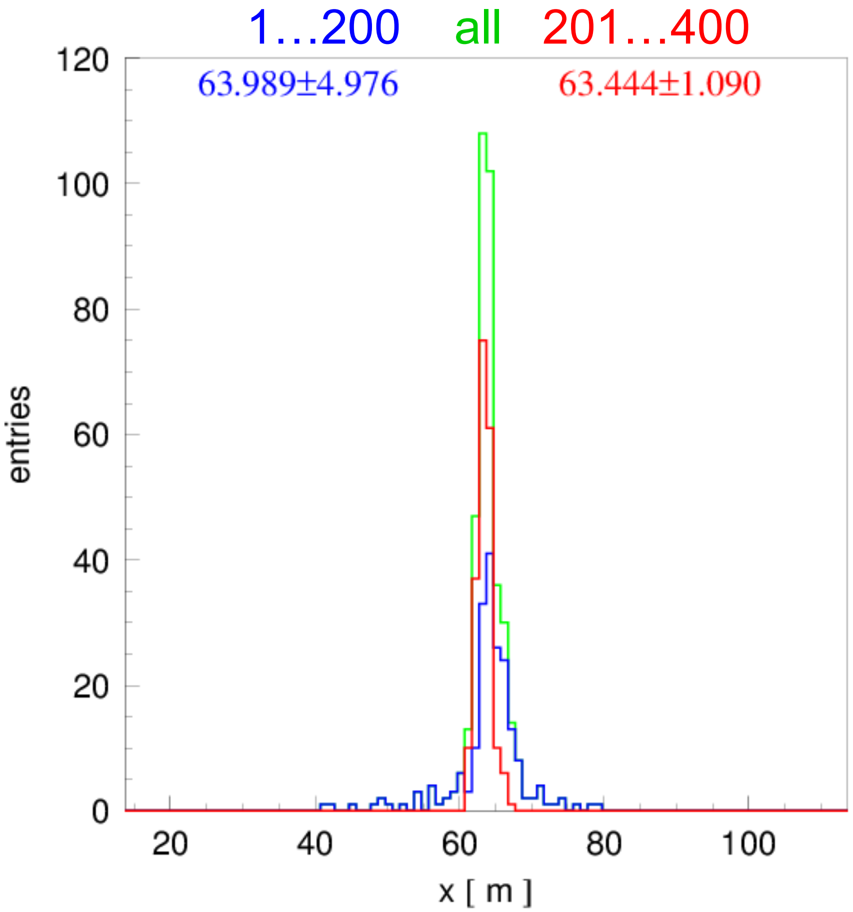


Dr. Strangepork

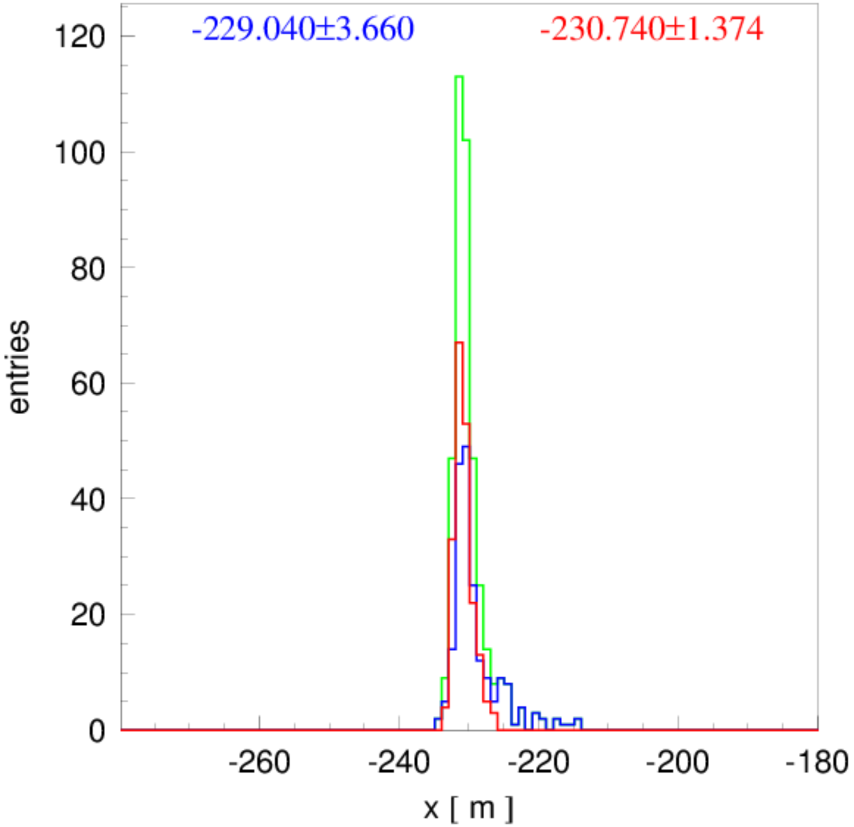


Bert

x



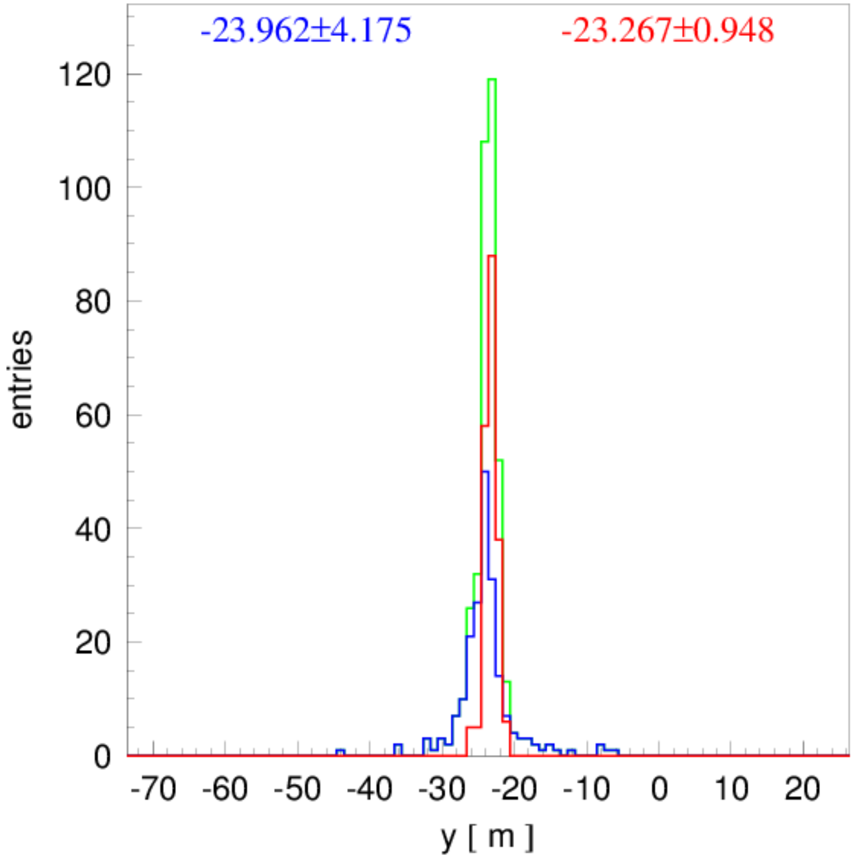
Dr. Strangepork



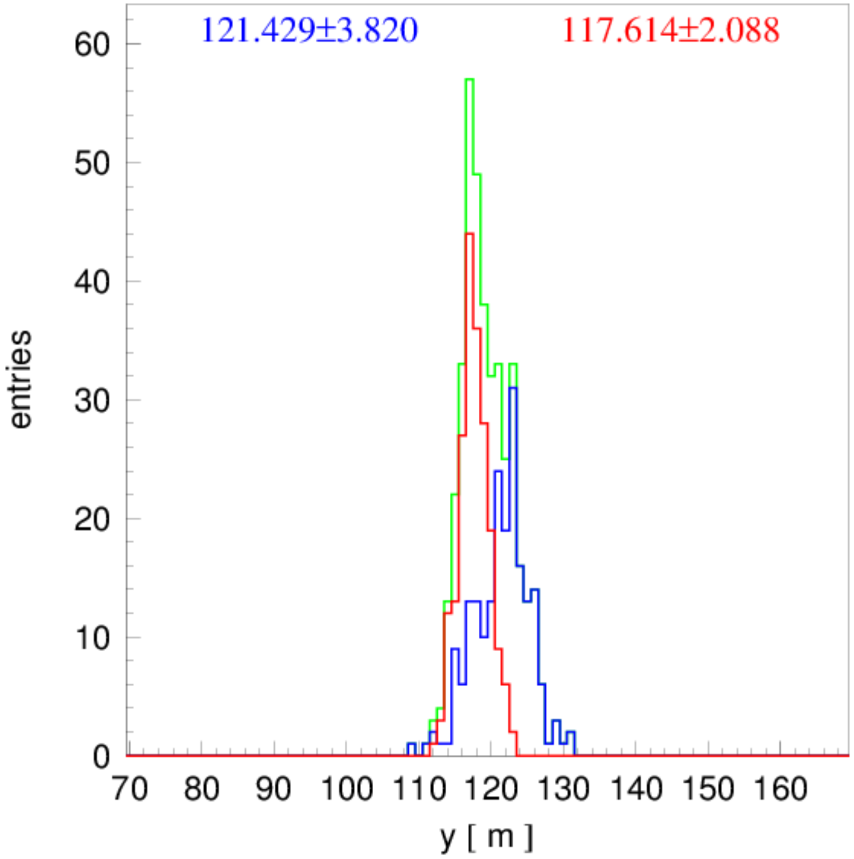
Bert

y

1…200 all 201…400

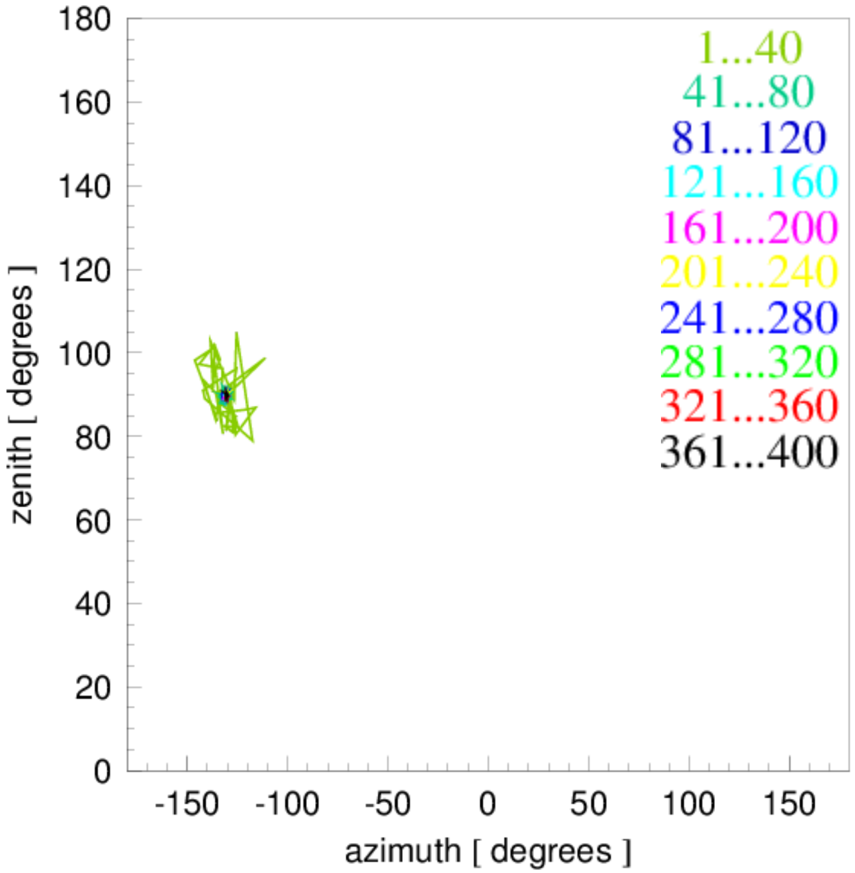


Dr. Strangepork

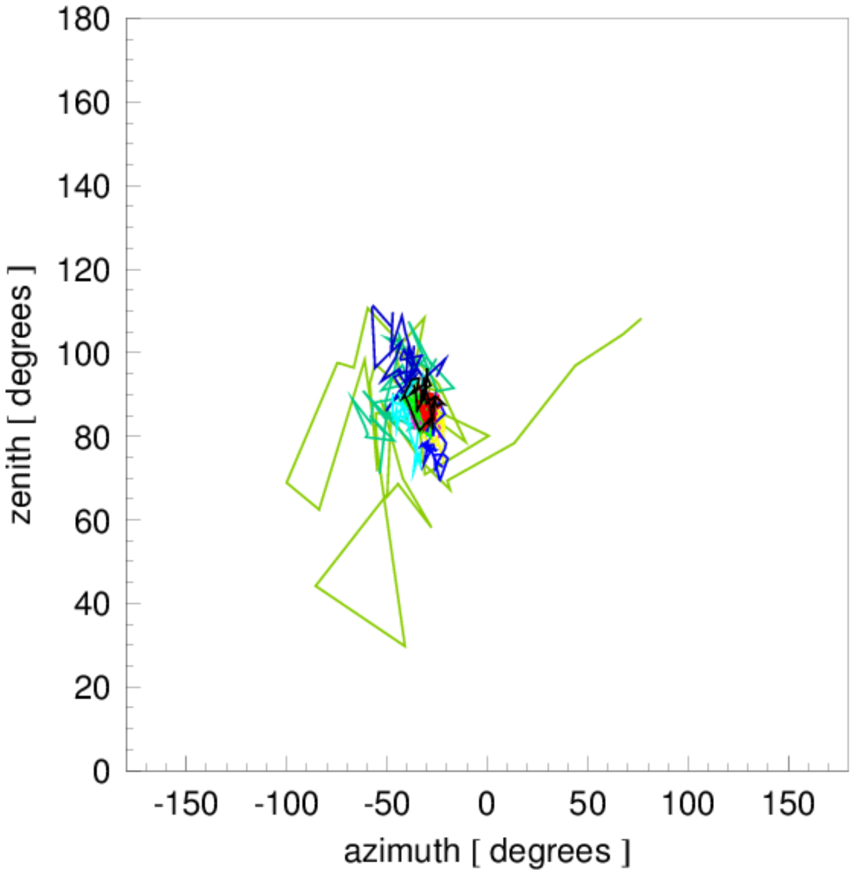


Bert

θ vs. φ



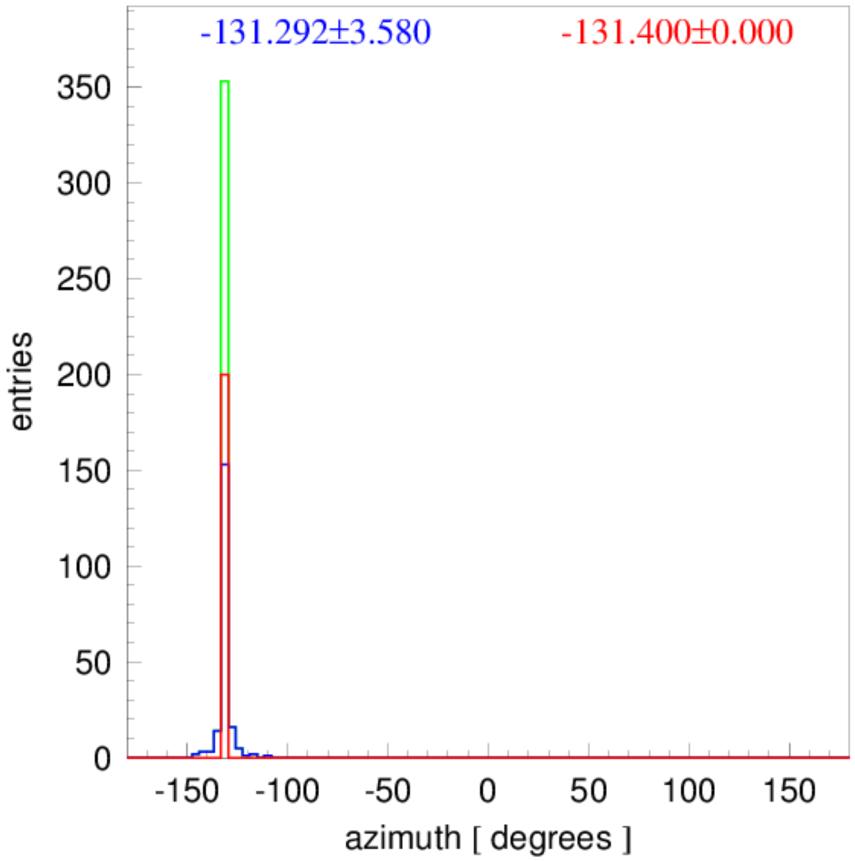
Dr. Strangepork



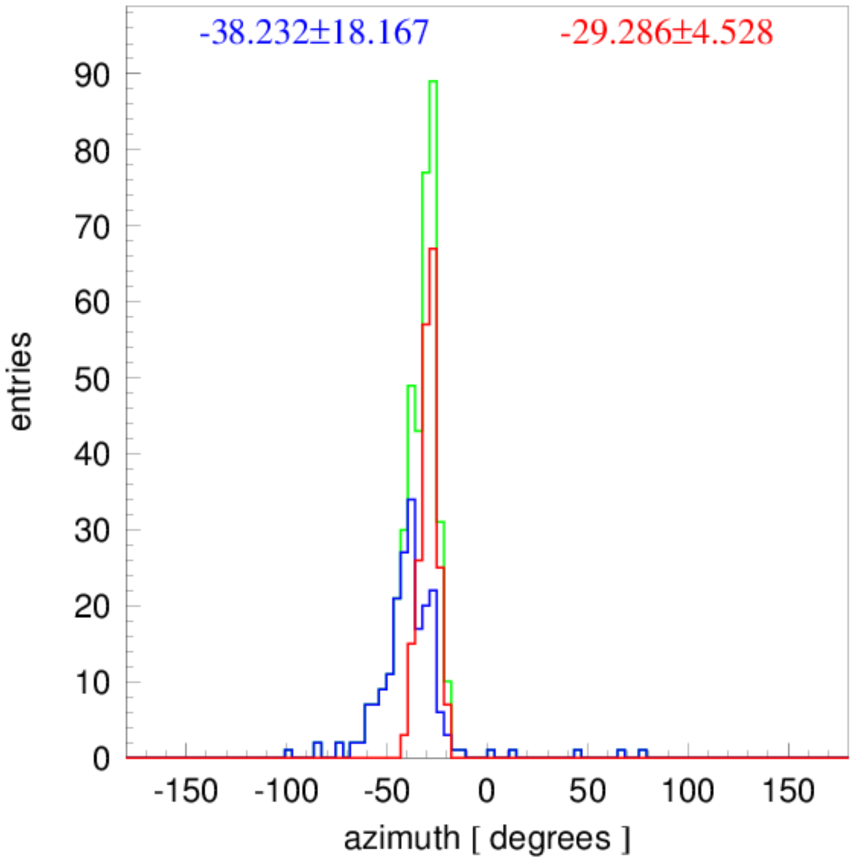
Bert

φ

1…200 all 201…400



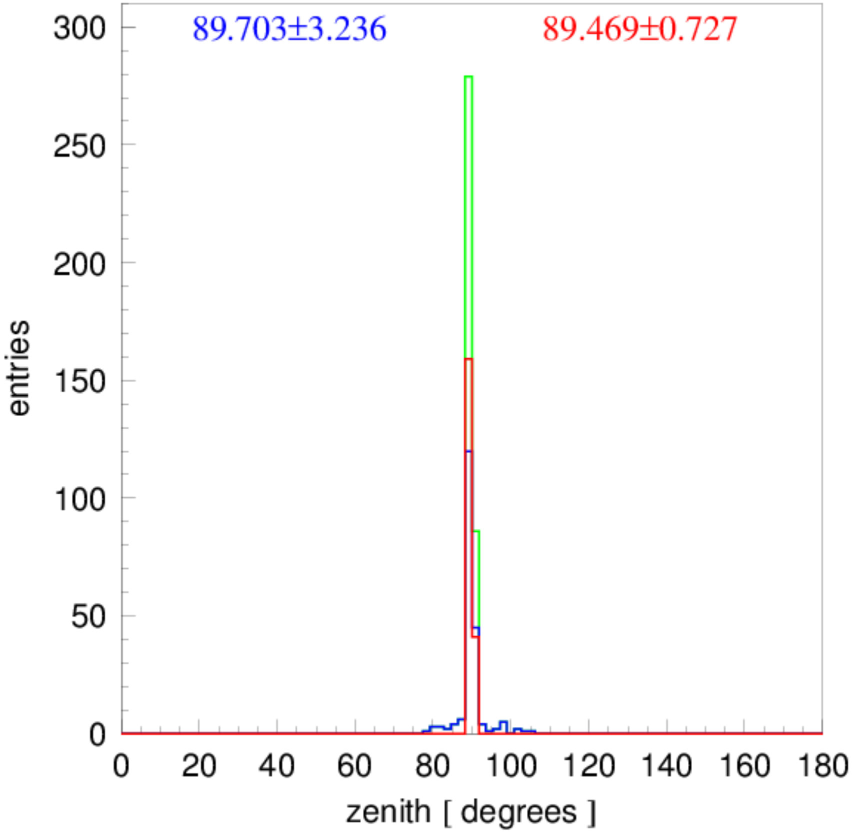
Dr. Strangepork



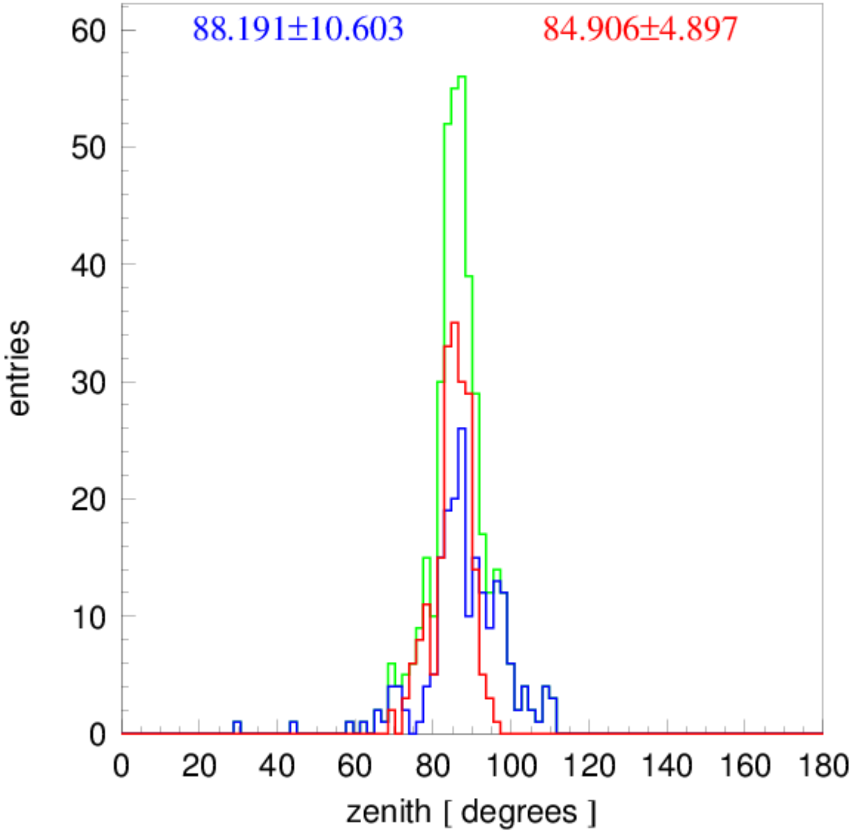
Bert

θ

1…200 all 201…400



Dr. Strangepork



Bert

References

Likelihood:

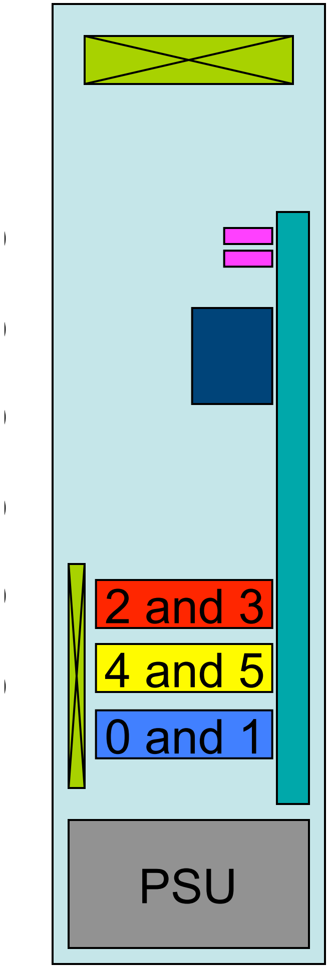
- Likelihood description for comparing data with simulation (LSSL: limited simulation statistics likelihood), arXiv:1304.0735
- DirectFit (and updated ice model): arXiv:1309.7010 (ICRC, Rio)

DirectFit experimental code:

http://icecube.wisc.edu/~dima/work/WISC/ppc/bkp/llh.tgz
http://code.icecube.wisc.edu/svn/projects/ppc/trunk/private/ppc/llh/

```
nvidia-smi -lsa

GPU 0:
          Product Name                       : GeForce GTX 295
          Serial                             : 1803836293359
          PCI ID                             : 5eb10de
          Temperature          : 87 C
GPU 1:
          Product Name                       : GeForce GTX 295
          Serial                             : 2497590956570
          PCI ID                             : 5eb10de
          Temperature          : 90 C
GPU 2:
          Product Name                       : GeForce GTX 295
          Serial                             : 1247671583504
          PCI ID                             : 5eb10de
          Temperature          : 100 C
GPU 3:
          Product Name                       : GeForce GTX 295
          Serial                             : 2353575330598
          PCI ID                             : 5eb10de
          Temperature          : 105 C
GPU 4:
          Product Name                       : GeForce GTX 295
          Serial                             : 1939228426794
          PCI ID                             : 5eb10de
          Temperature          : 100 C
GPU 5:
          Product Name                       : GeForce GTX 295
          Serial                             : 2347233542940
          PCI ID                             : 5eb10de
          Temperature          : 103 C
```



As fast as 900 CPU cores

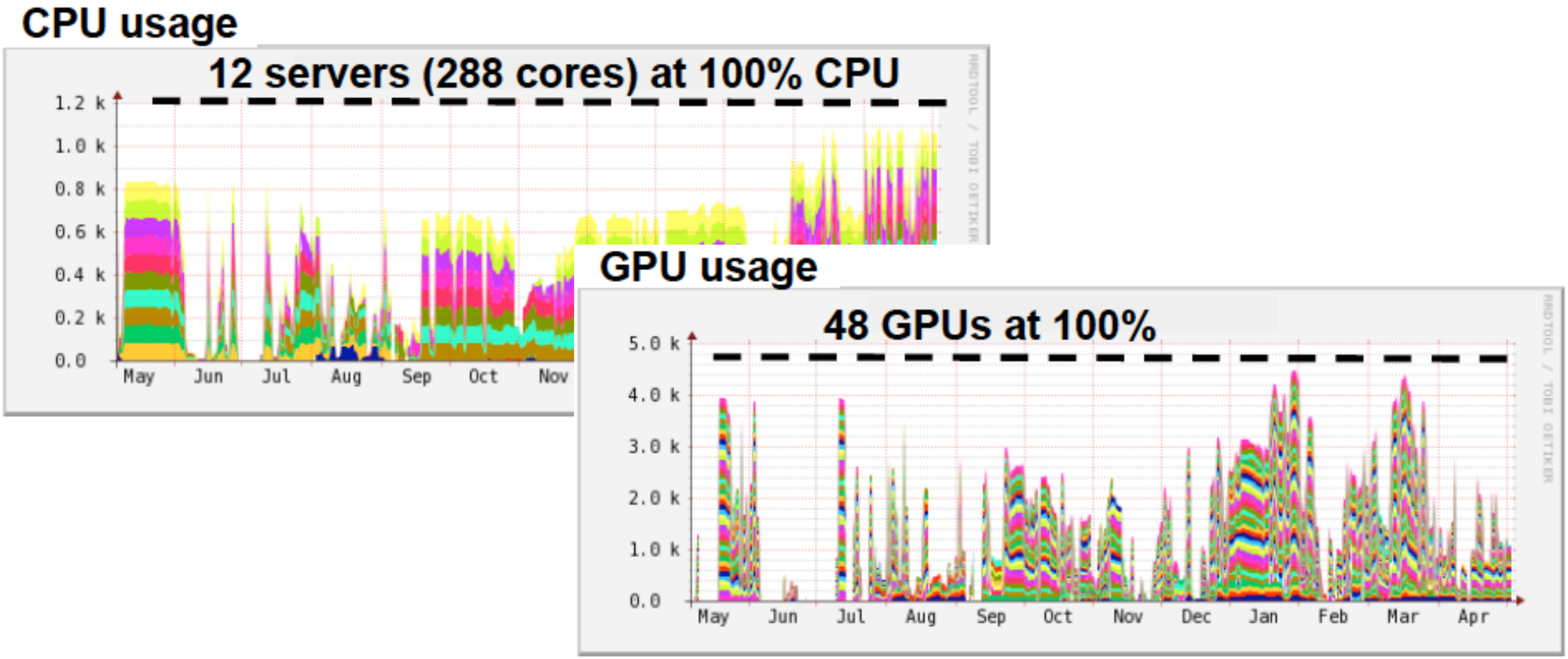
GZK9000 GPU Cluster

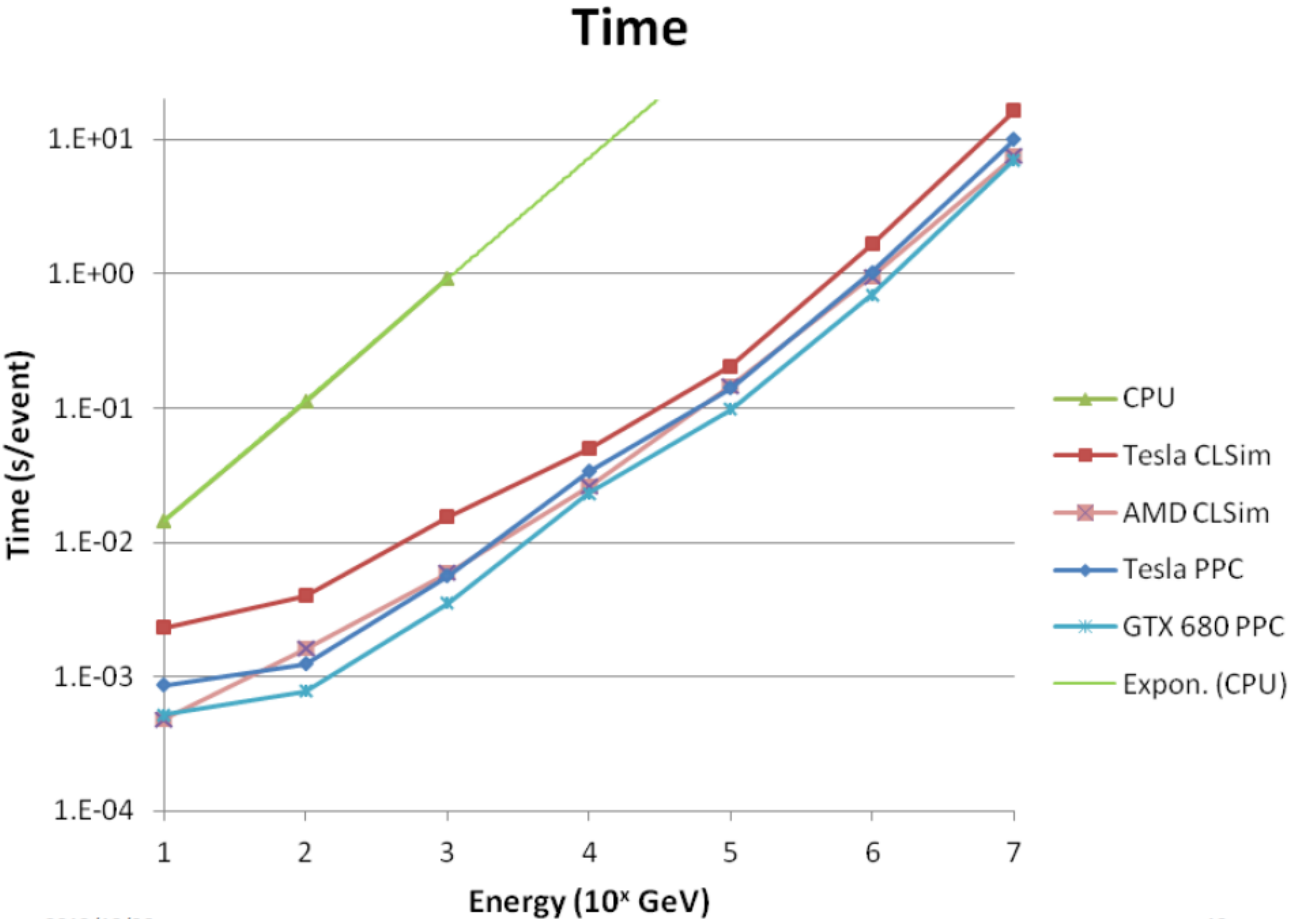
Deployed in 2012 at the WID/MIR datacenter (shared with CHTC @ 30%)

12 servers, each with

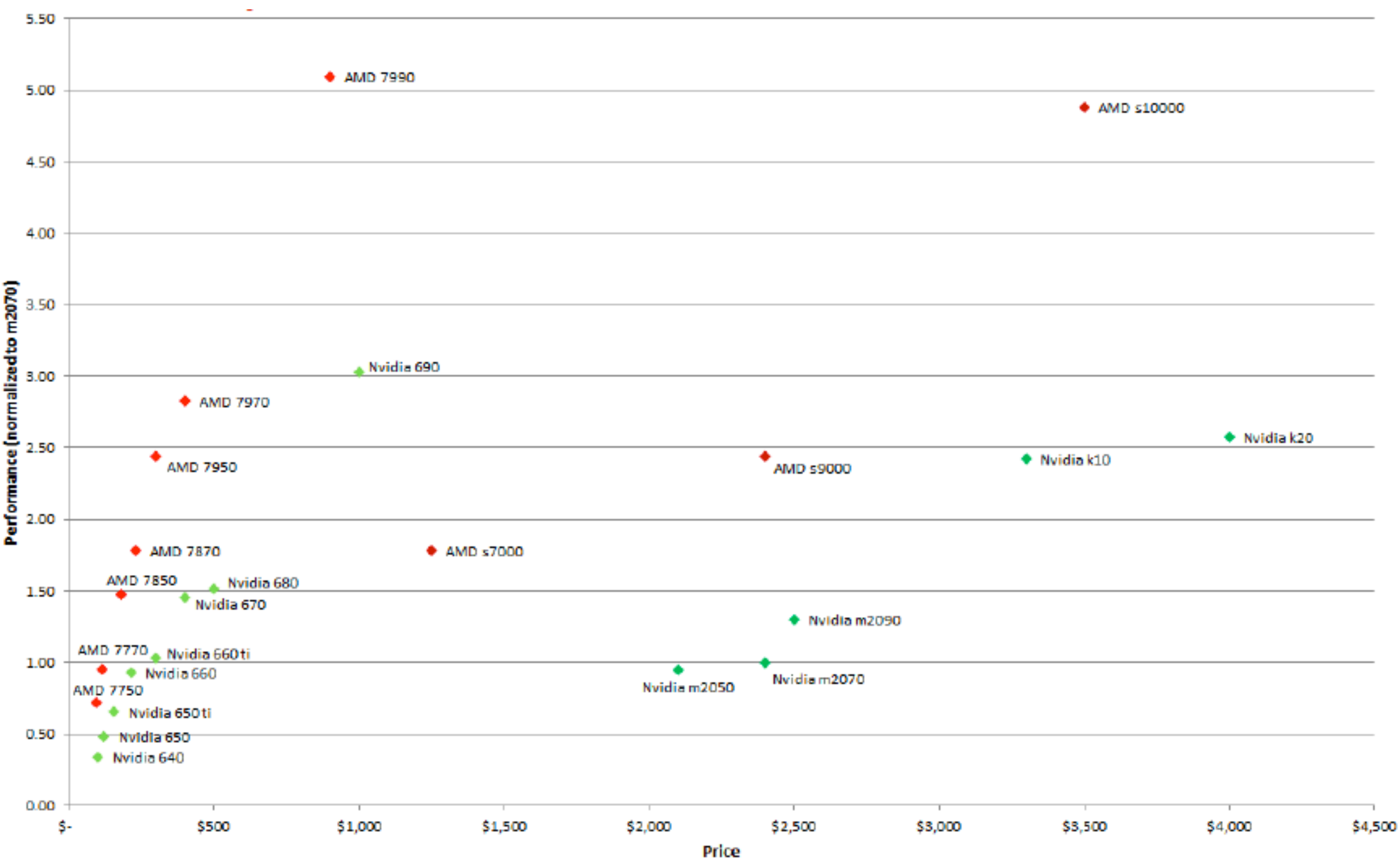
2x AMD 6176 (12 cores/CPU)

4x GPUs Nvidia Tesla M2070 (448 CUDA cores/GPU)




Time
1.E+01
1.E+00
1.E-01
1.E-02
1.E-03
1.E-04
Time (s/event)
1
2
3
4
5
6
7
Energy (10ˣ GeV)
CPU
Tesla CLSim
AMD CLSim
Tesla PPC
GTX 680 PPC
Expon. (CPU)

Performance vs. Price



GPU Cluster expansion

During 2012, purchased several evaluation systems based on consumer-grade GPUs. Benchmarked running simulation production.

See: https://wiki.icecube.wisc.edu/index.php/GPU_Resources

From IC86, the main simulation production has moved into GPU-based photon propagation.

Several months experience running on GZK9000 cluster.

=> 300 GPUs need estimated for real time simulation.

Agreed to provision 50% of this resource at UW-Madison. Plan to bring this new resource online during Summer 2013.

Plan forward: keep a constant budget for GPU expansion in the next years foreseeing a general increasing need for GPU power.

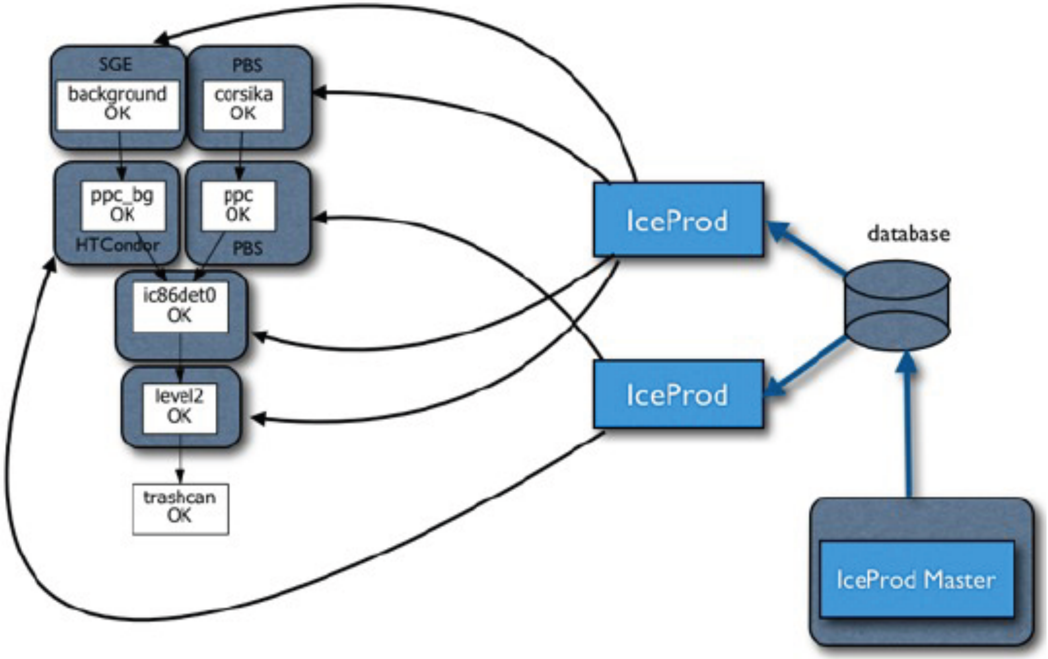
New GPU resources on NPX4

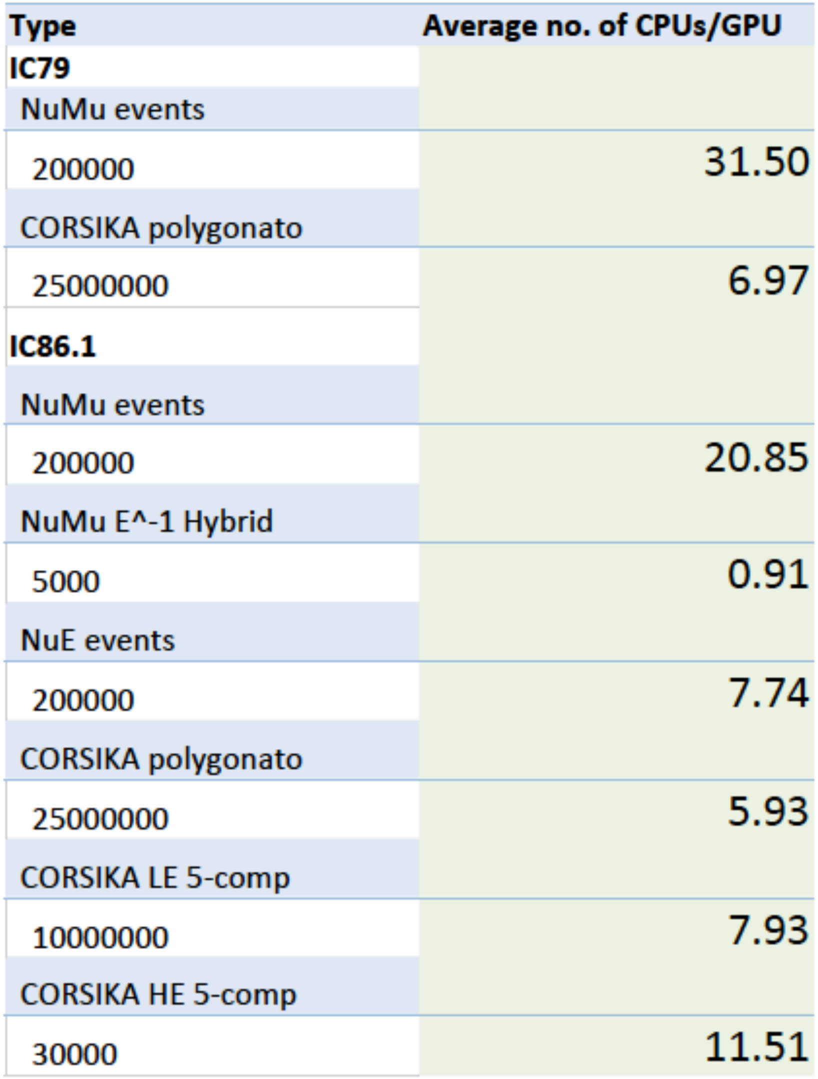
16 servers recently purchased and deployed at UW-Madison

- CPU: 2x Intel Xeon E5-2670 (16 cores/server, 4GB RAM/core)
- GPU:
  - 8 servers with 4x Nvidia GTX 690 cards
  - 8 servers with 4x AMD Radeon 7970 cards

Just made available to the collaboration as additional NPX cluster slots.

Production optimizations





Type	Average no. of CPUs/GPU
IC79	
NuMu events	
200000	31.50
CORSIKA polygonato	
25000000	6.97
IC86.1	
NuMu events	
200000	20.85
NuMu E^-1 Hybrid	
5000	0.91
NuE events	
200000	7.74
CORSIKA polygonato	
25000000	5.93
CORSIKA LE 5-comp	
10000000	7.93
CORSIKA HE 5-comp	
30000	11.51