PFA Status

Mat Charles (mcharles@slac.stanford.edu)
The University of Iowa

Algorithm quick overview

- Identify easy shower/sub-shower bits
  - Track segments, photons, dense clumps, other hits
- Remove photons, electrons [incomplete]
- Find hadronic showers, including structure
  - Separate overlapping showers
  - Extrapolate tracks from IP, match to clusters
  - Identify & assign fragments

Study pairwise combinations of tracks & clumps; apply a likelihood selector to see whether they're linked

See previous talks:

Boulder: http://www.slac.stanford.edu/~mcharles/talks/2006-01-10_boulder/

SiD: http://www.slac.stanford.edu/~mcharles/talks/2005-09-21_pflow/

Snowmass: http://www.slac.stanford.edu/~mcharles/talks/2005-08_snowmass/

News from Boulder: PFA Framework

- Agreed on a framework for PFAs in org.lcsim
- Goal is to make code-sharing feasible
- More info on the web, but basics are:
  - Modular design
  - Store hit collections as HitMaps (basically a Map<Long,CalorimeterHit>)
  - Leave input collections unchanged, upload modified collections to the Event

PFA Framework webpage

https://confluence.slac.stanford.edu/display/ilc/lcsim+PFA+guide

PFA Code status

- CVS: In the process of migrating code from contrib into org.lcsim proper
- Most of the interesting parts are there already
- ... but still working on some parts to allow a start-to-finish PFA in org.lcsim (e.g. need a ReconstructedParticle implementation)
- Unstable snapshot of PFA is in CVS at

  `contrib/uiowa/template/NonTrivialPFA.java`

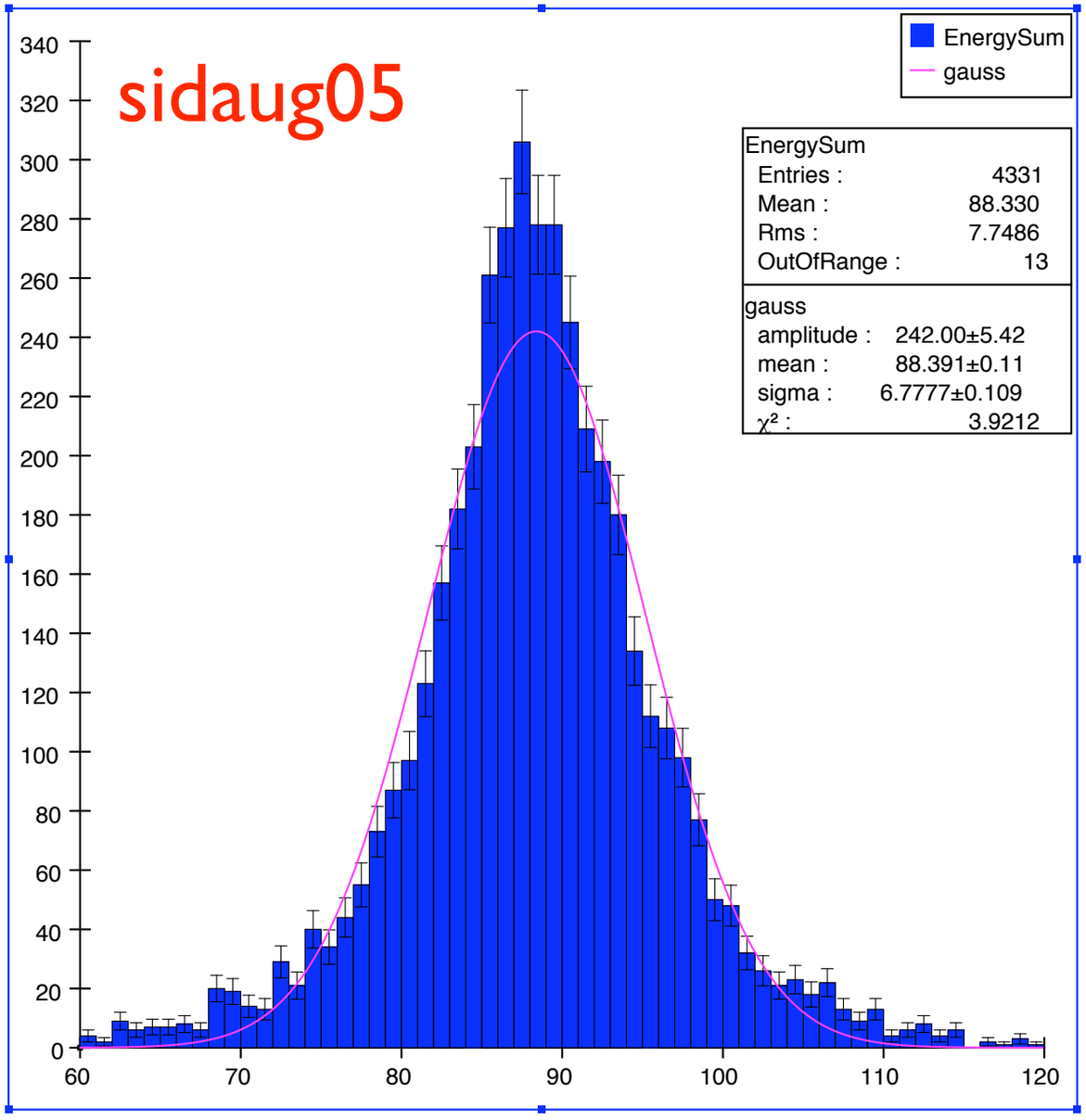
Code status: org.lcsim.xxx

`recon.cluster.structural`	Drivers and utilities for structural algorithm
`recon.cluster.structural.likelihood`	Likelihood quantities for structural algorithm
`recon.cluster.mst`	MST clusterer
`recon.cluster.mipfinder`	Drivers to find MIPs, track segments in calorimeter
`recon.cluster.clumpfinder`	Driver to find dense clumps
`util.hitmap`	HitMap class and supporting drivers
`util.decision`	Simple templated interface for accepting/rejecting objects

Results: Energy Sum

Energy sum distribution in hadronic Z-pole events, minimal* cheating:

* Cheating on photon-finding, missing energy



Single Gaussian fit

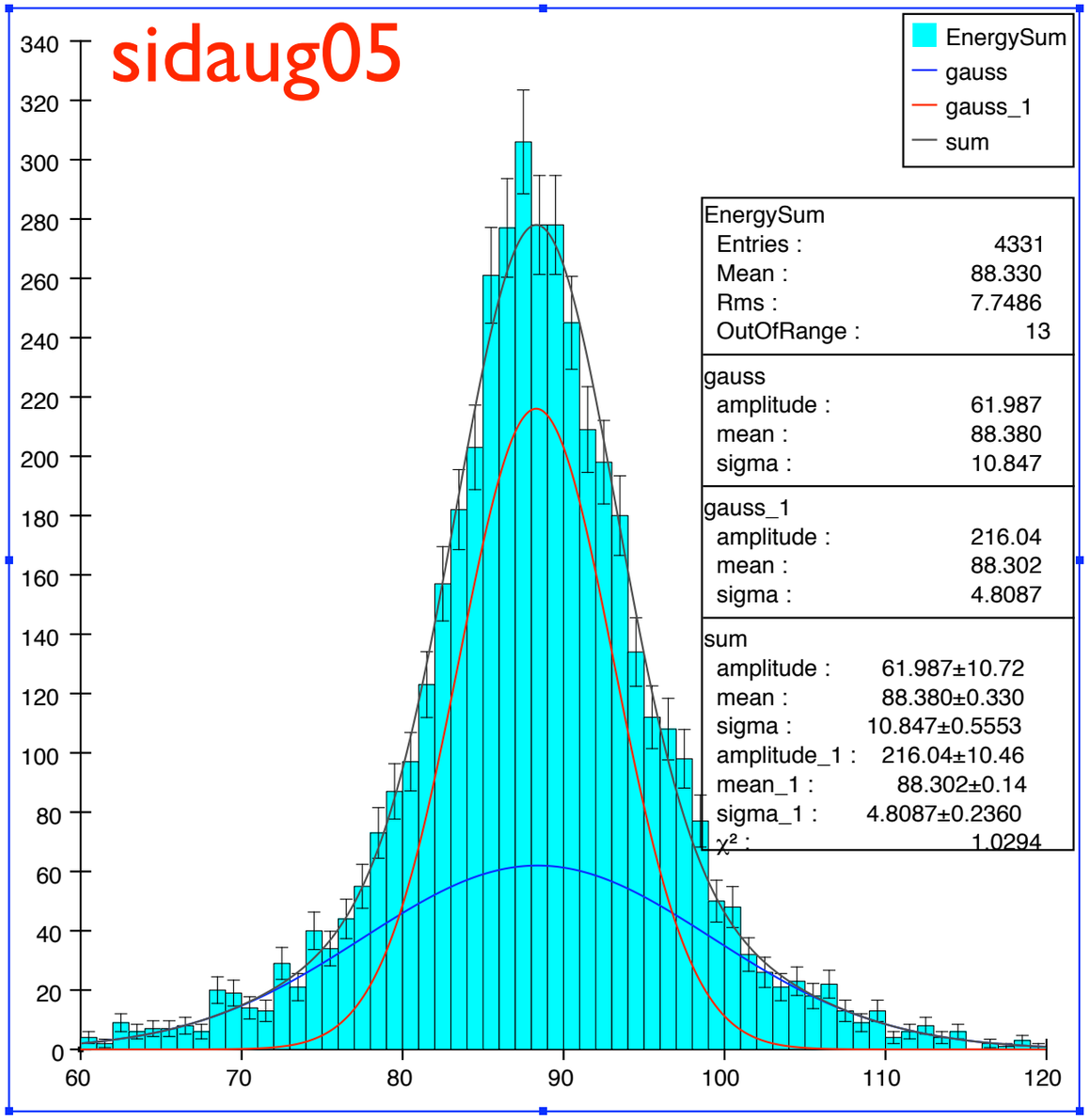
Mean	88.4 GeV
Sigma	6.8 GeV

HWHM: 8.0 GeV

... but this folds in many different effects (e.g. calibration)

Results: Energy Sum

Energy sum distribution in hadronic Z-pole events, minimal cheating:



Double Gaussian fit

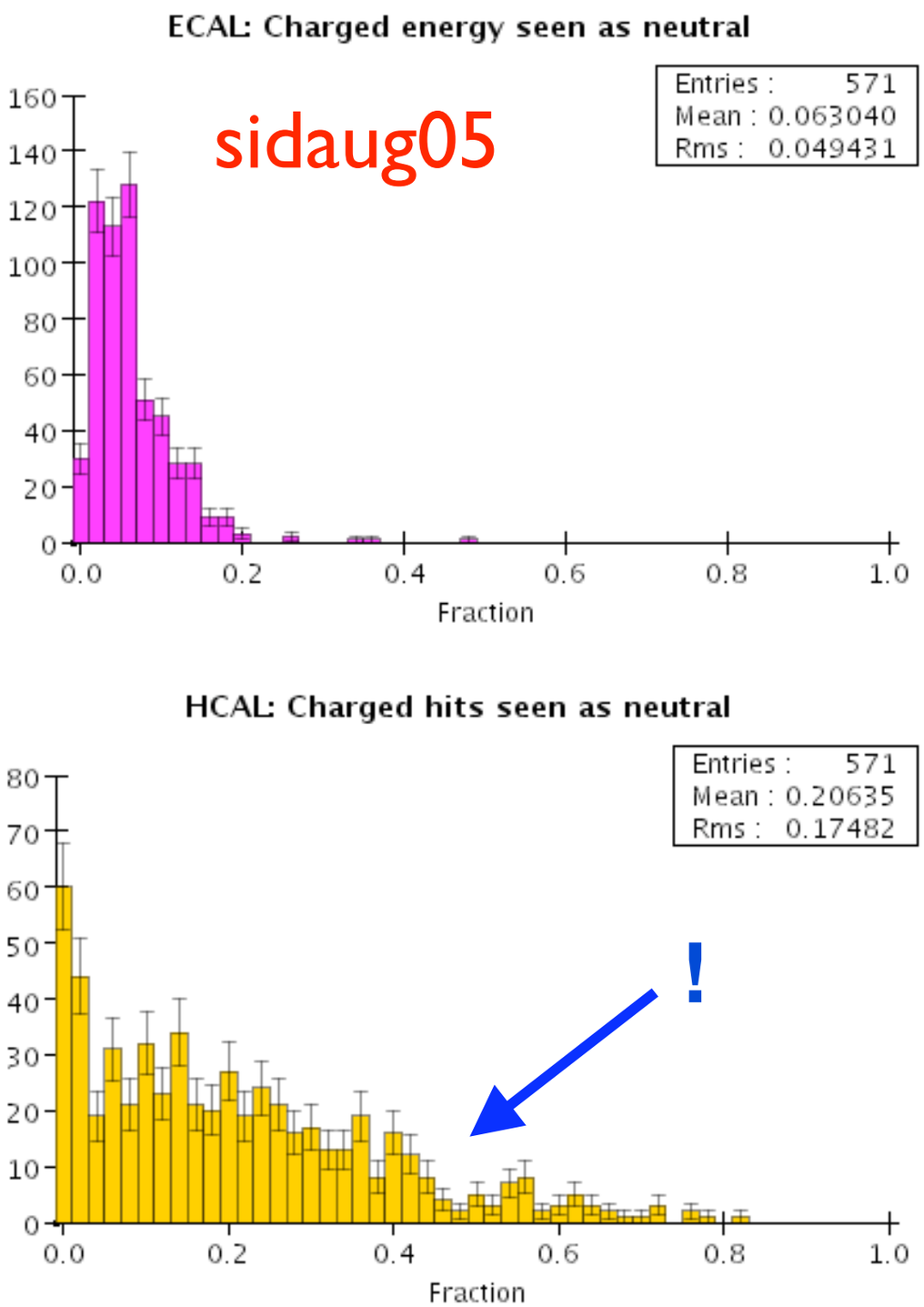
Mean	88.3 GeV	88.4 GeV
Sigma	4.8 GeV	10.8 GeV
Area	61%	39%

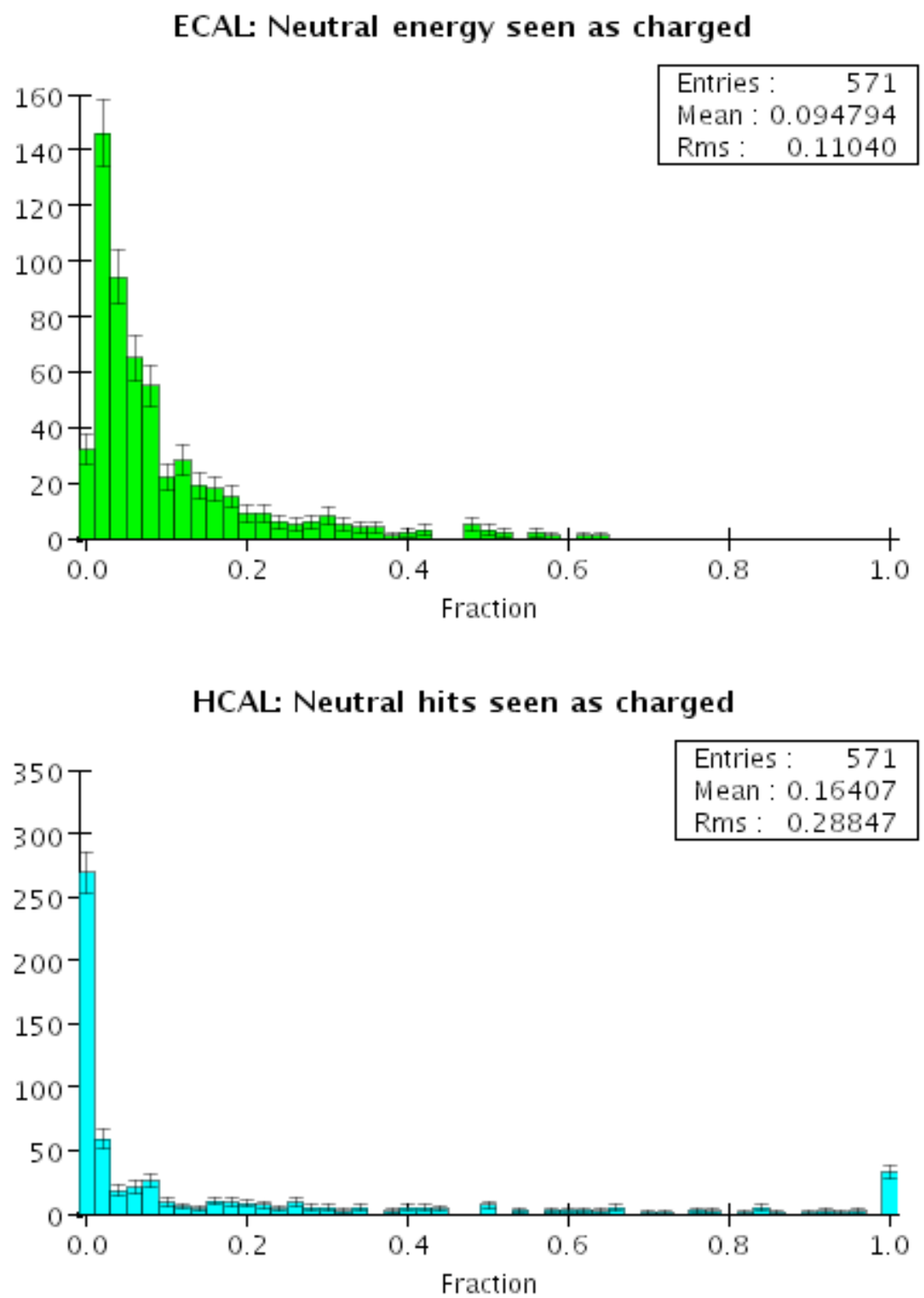
HWHM: 6.5 GeV

... but this folds in many different effects (e.g. calibration)

Results: Confusion PDFs

No cheating; simple fragment ID & assignment





Results: Toy MC

Relevant parts of resolution for PFA are:

1) Intrinsic detector resolution
2) Confusion term

sidaug05	Mean (GeV)	RMS (GeV)
No confusion	91.0	2.04
Cheating on fragment ID and fragment assignment	90.5	2.94
Cheating on fragment ID but not fragment assignment	93.2	3.88
Not cheating at all	92.9	4.71

Not cheating on cluster core reconstruction, track matching etc. to get confusion PDFs

Resolution is dominated by fragment handling

Fragment thoughts

- My current implementation is pretty dumb
- Some better solutions exist (c.f. Lei Xia)
- Some ideas to feed into a probabilistic or likelihood-based selector:

Credits: Adam Para, Usha Mallik

  - Regional information (nearby charged clusters, neutral clusters, helices)
  - E/p for nearby charged jet
  - Depth (λ_i for primaries, $\lambda_i+\lambda_i$ for secondaries)
  - Angle from parent to fragment vs parent mom.

This needs to move forward

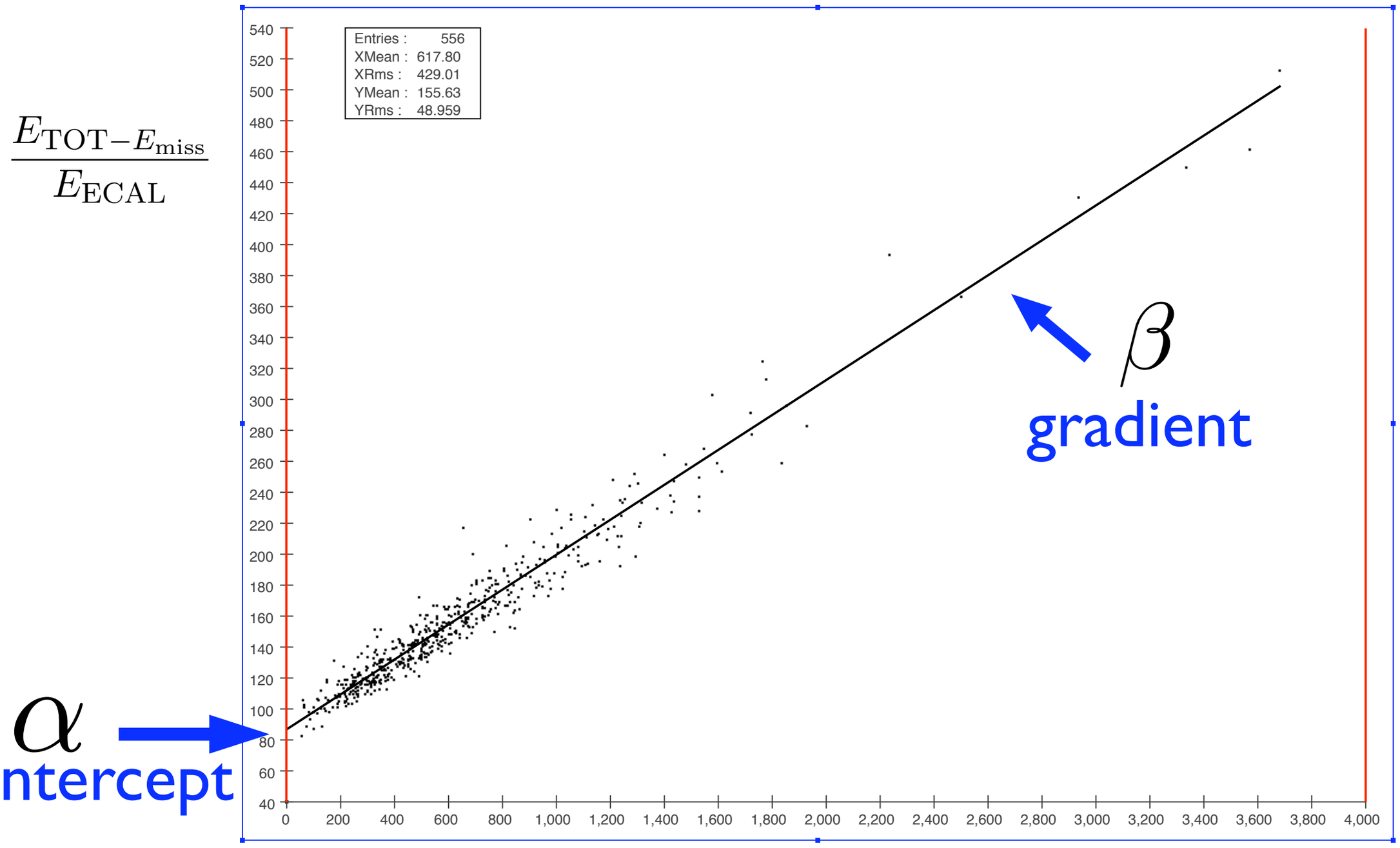
Future plans

- Finish migration to org.lcsim tree
- Better fragment handling
- Second pass / iterative approach
  - Currently only using local information to make decisions... should use regional too
- Pop in a real photon finder
- Look at harder events (> Z-pole)

Post-script

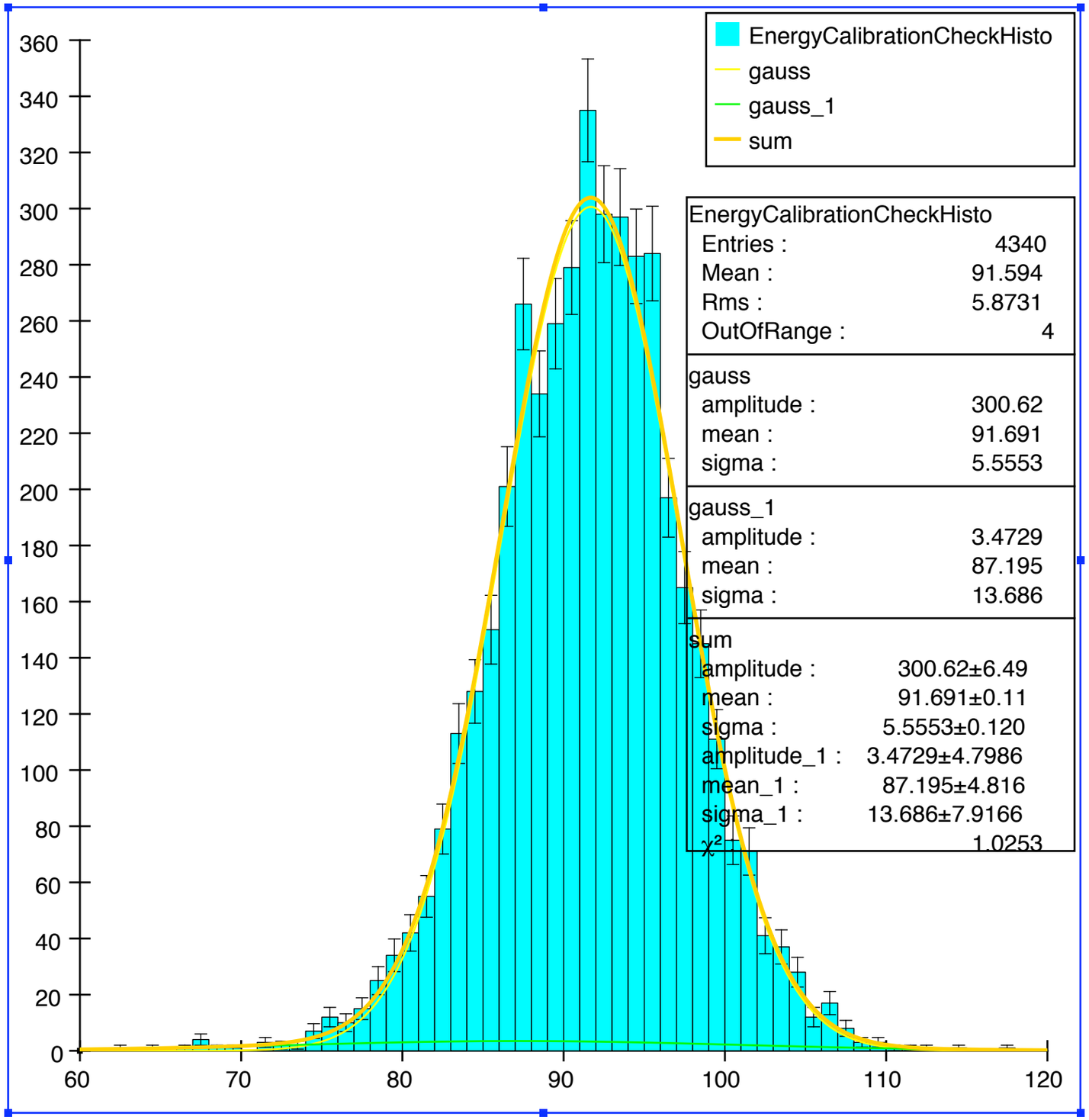
Comparison to a non-PFA straw man algorithm

$$\alpha E_{\mathrm{ECAL}} + \beta E_{\mathrm{HCAL}} + E_{\mathrm{miss}} = E_{\mathrm{TOT}} = 91.0\mathrm{GeV}$$



Post-script

Comparison to a non-PFA straw man algorithm



Double Gaussian fit

Mean	91.7 GeV	87.2 GeV
Sigma	5.6 GeV	13.7 GeV
Area	97%	3%

HWHM: 6.6 GeV