DigiSim Tutorial

Guilherme Lima
for the ILC-software group at NIU

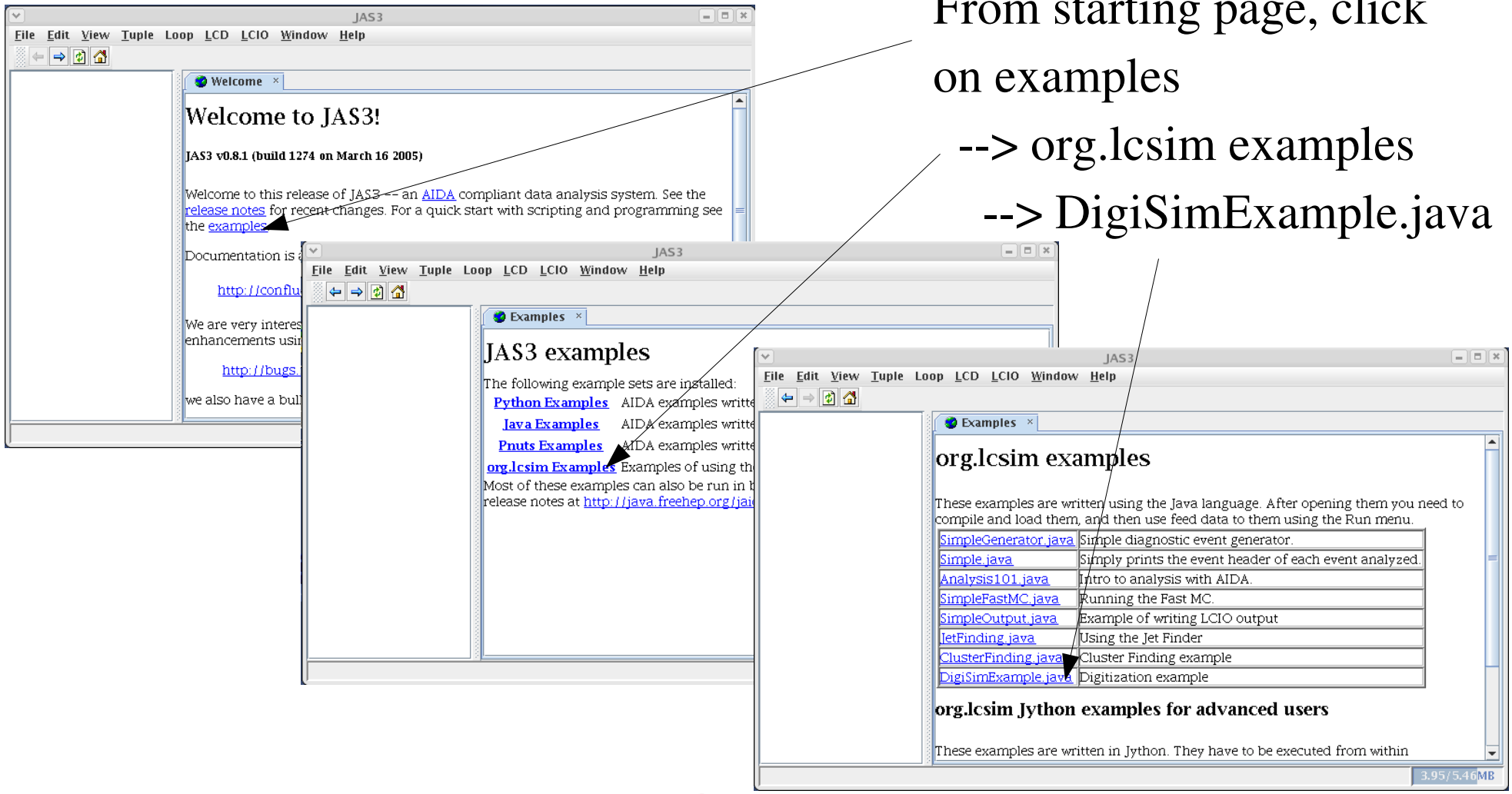
NORTHERN ILLINOIS
UNIVERSITY

ILC Simulations Workshop
Boulder, January 09-11, 2006

DigiSim usage instructions

- Download/install/build java 1.5, Maven 1.0.2, org.lcsim
  (see http://www.lcsim.org for details)
- Drivers needed: (all available from org.lcsim.digisim)
  CalHitMapDriver, DigiSimDriver and CalorimeterHitsDriver,
  plus LCIODriver (for standalone run, saving output file)
  or YourAnalysisDrivers (as an on-the-fly preprocessor)
- DigiSim configuration file stored on LCDetectors: digisim/digisim.steer
- Run it:
  - From command line: after setting the CLASSPATH (see docs for details)
    java org.lcsim.digisim.DigiSimMain <inputfile>
    an output file ./digisim.slcio will be produced, to be used for analysis or reconstruction
  - From inside JAS, as an on-the-fly preprocessor:
    DigiSimExample is available from JAS3 --> Examples --> org.lcsim examples

Running DigiSim inside JAS3

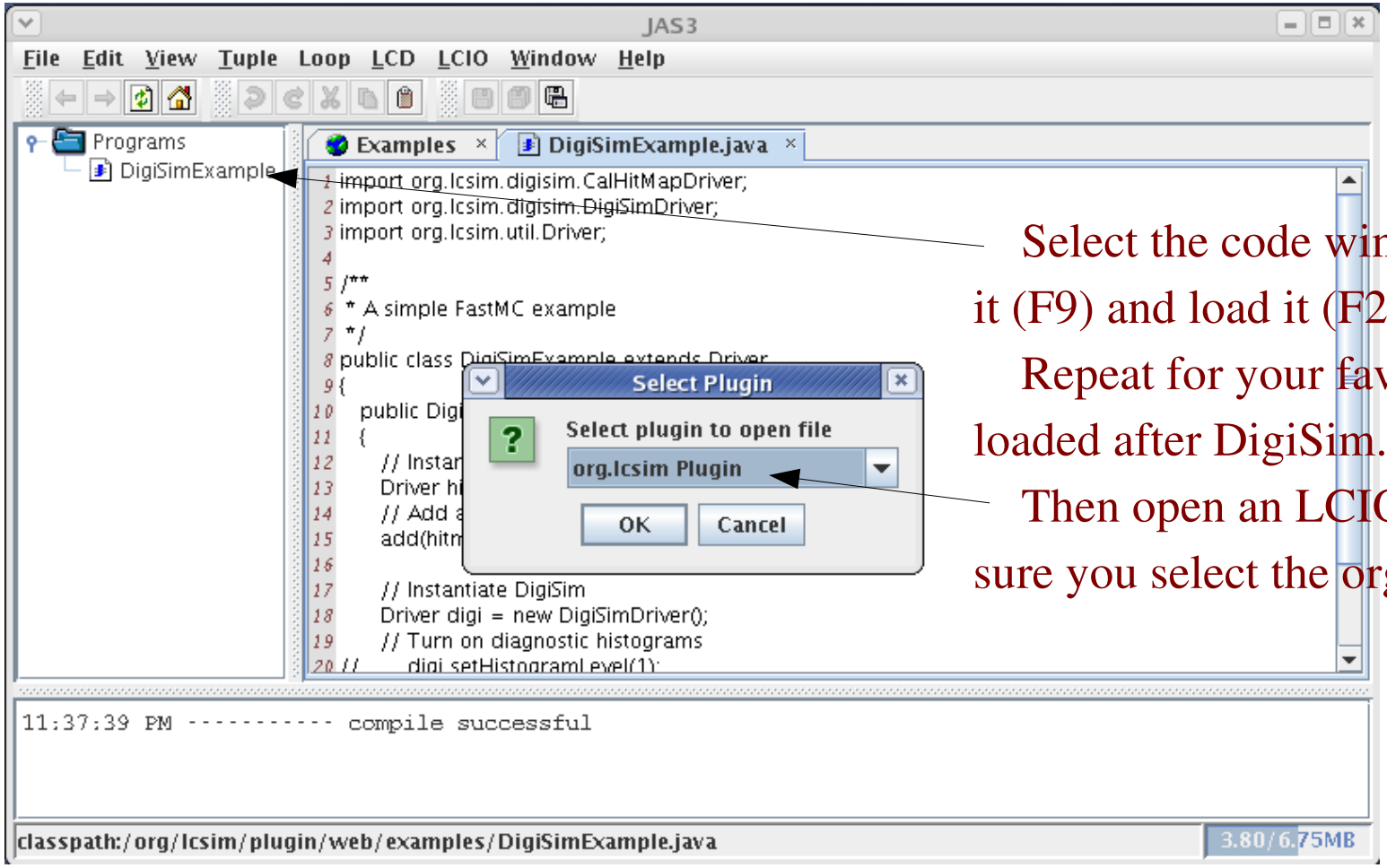


From starting page, click on examples

--> org.lcsim examples

--> DigiSimExample.java

Loading driver(s) and data

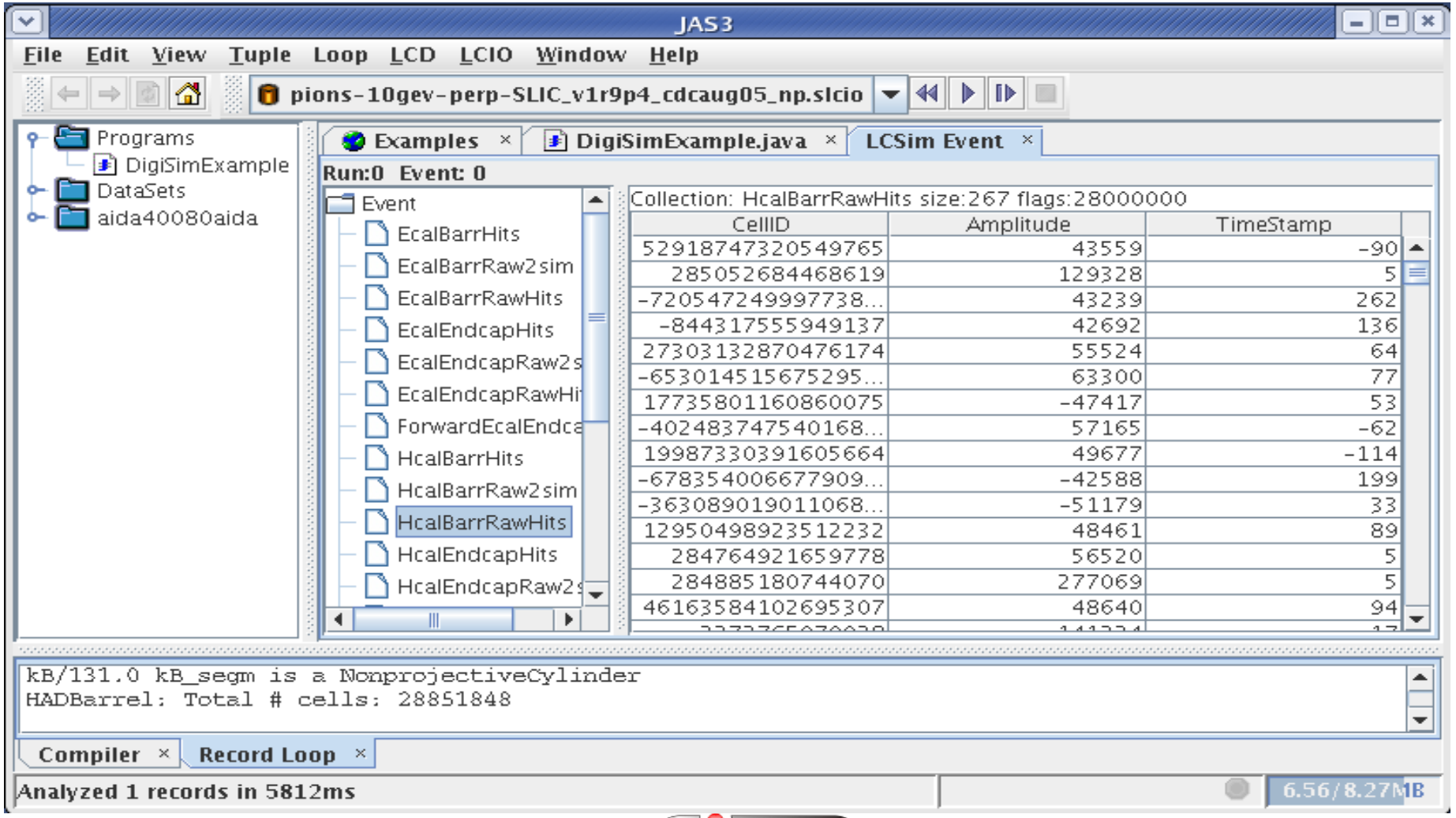


Select the code window, compile it (F9) and load it (F2).

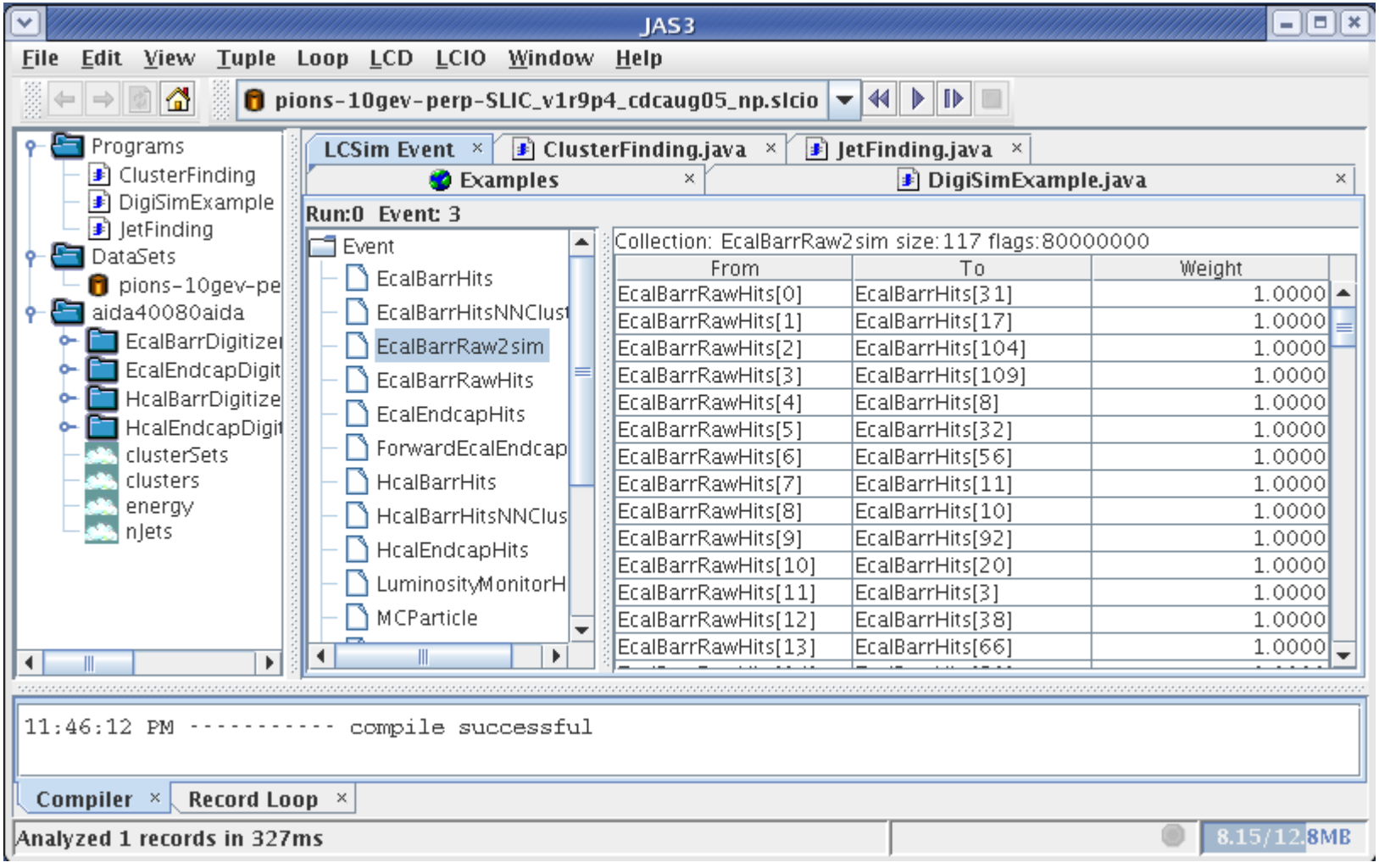
Repeat for your favourite drivers, loaded after DigiSim.

Then open an LCIO file. Make sure you select the org.lcsim plugin.

Looking at raw hits...



...and raw -> sim links



Live demo 1: DigiSim as a JAS3 preprocessor

- Launch JAS3 and load any physics sidaug05 data file
- Download http://nicadd.niu.edu/~lima/ilc-simws/plot4.py and open it from JAS3
- From the org.lcsim examples tab, open/compile/load DigiSimExample and ClusterFinding (in this order)
- At ClusterFinding.java loading, a new folder appears, like aidaXXXXaida. Update “path” variable in plot4.py accordingly
- Fire the event processing and run plot4.py right away. Observe the effect of identity transformations on clustering
- Look at some raw and digitized hit collections

Live demo 2: updating the steering file

- Checkout LCDetectors from CVS repository
  see for instance: detectors/sidaug05/digisim/digisim.steer
- Download http://nicadd.niu.edu/~lima/ilc-simws/digisim.steer
  and compare it to the existing one in the detector conditions area (LCDetectors/detectors/sidaug05/digisim/)
- Replace old with new steering file and rebuild your detector .zip file (use script LCDetectors/detectors/makeZips.sh)
- Copy the relevant .zip file to ~/.lcsim/cache
- Restart JAS3 and run again, observing the effect of energy thresholds on clustering

Live Demo 3: running DigiSim standalone

- Type this sequence of commands (on Linux)

```
> cd <some/convenient/place>
> wget  http://nicadd.niu.edu/~lima/ilc-simws/digi.tgz  # download a tarball
> tar xzvf digi.tgz          # open the tarball
> cd digi                    # move to the directory with standalone java code
> source ./addjars.sh ~/.maven/repository    # define your CLASSPATH.  This
                # requires the org.lcsim libs in the repository.  Use "maven jar:install"
> ln -sf /path/to/your/data/file inputfile   # point to the input data file
> make                                       # compile the java standalone code
> java MainLoop                              # run over data in inputfile
```

- Two files produced: digisim.slcio (digitized data) and digisim.aida

```
> jas3  digisim.aida &
```

SimCalorimeterHits or CalorimeterHits?

- Consider moving your reconstruction algorithms to use CalorimeterHits instead of SimCalorimeterHits
- How to do this:
  - All non-MC calls to SimCalorimeterHits (energy, position, time) can be transparently replaced with equivalent calls to CalorimeterHit objects.
    For MC-related methods, use (same) cellid as a key to access SimCalorimeterHits.
  - Detector conditions system is used to select the correct DigiSim configuration file.
  - Configuration files exist for most Snowmass detectors
    - All RPC-based have identity configurations
    - All scintillator-based: SDJan03, sidaug05_scint, cdcaug05* have non-identity configurations (see later)
  - Identity DigiSim config files are available for helping people to get started with DigiSim output

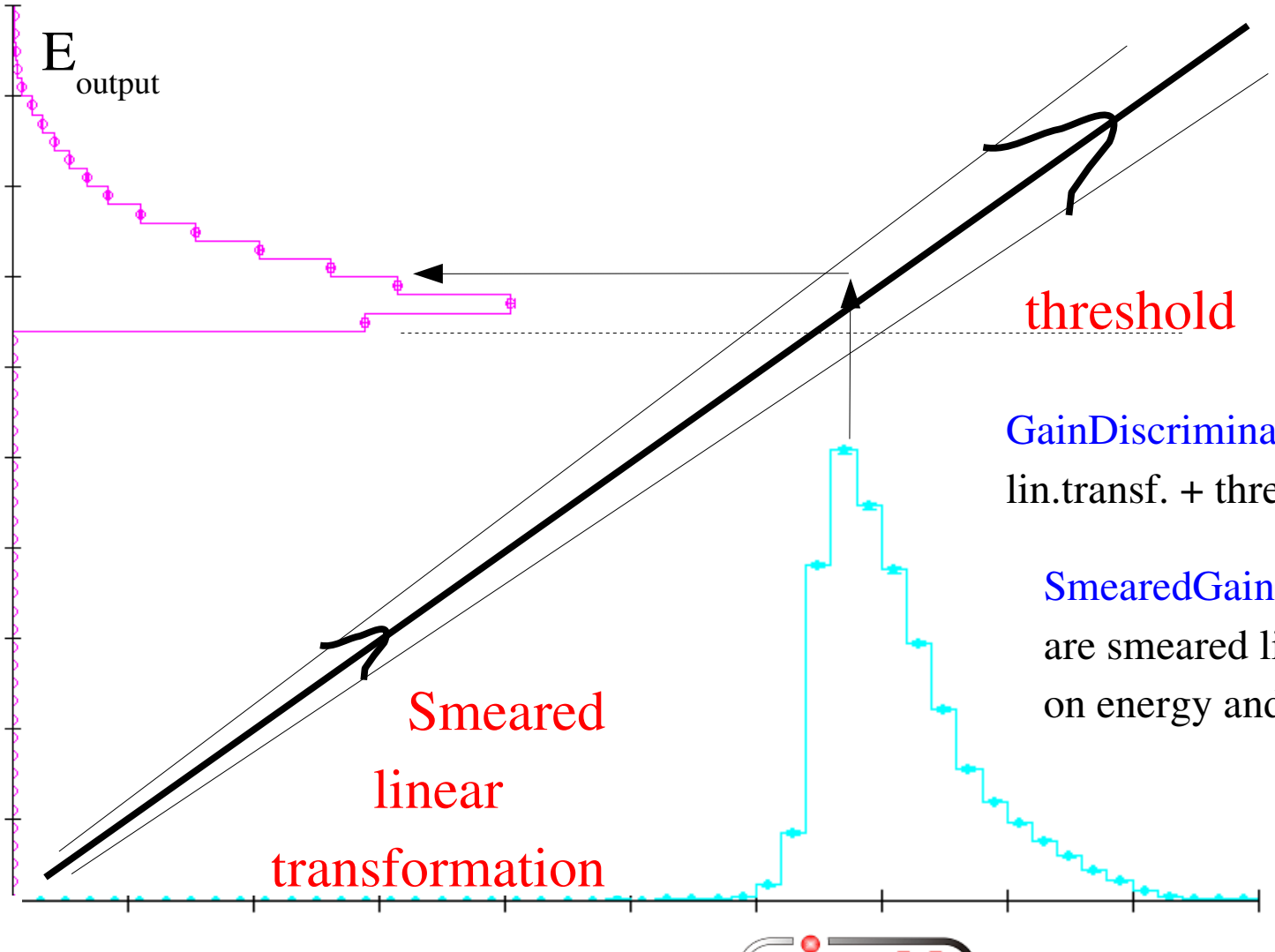
Setting up DigiSim modifiers

- Modifiers' role is to *tweak* hit energy and timing
- Important: It is helpful to interpret the "energy" field according to the process to be modeled:

  Energy (GeV) --> (light yield) --> # produced photons --> (photon collection) --> # photons collected --> (Quantum effic) --> # photoelectrons --> (Gain) --> uAmp --> (signal integration) --> charge collected --> (Digitization) --> ADC counts

- DigiSim modifiers are just factors (or more generally, functions) which represent each step along the digitization process

A common transformation



GainDiscrimination is a smeared lin.transf. + threshold on energy

SmearedGain and SmearedTiming are smeared linear transformations on energy and time respectively

Setting up a DigiSim configuration file

```
##############################################
#  Example steering file for DigiSim
##############################################
.begin Global  ----------------------------------------
# specify one or more input files (in one or more lines)
LCIOInputFiles inputfile

# the active processors that are called in the given order
ActiveProcessors CalHitMapProcessor
ActiveProcessors EcalBarrelDigitizer
ActiveProcessors EcalEndcapDigitizer
ActiveProcessors HcalBarrelDigitizer
ActiveProcessors HcalEndcapDigitizer
ActiveProcessors OutputProcessor

# limit the number of processed records (run+evt):
MaxRecordNumber 500
.end Global  ------------------------------------------
#############################################################
.begin EcalBarrDigitizer
ProcessorType DigiSimProcessor

InputCollection        EcalBarrHits
OutputCollection       EcalBarrRawHits
Raw2SimLinksCollection EcalBarrRaw2sim

ModifierNames    EMBDigiIdentity
# modifierName    Type                  Parameters (floats)
EMBDigiIdentity    SmearedGain          100000000       0
.end ---------------------------------------------
```

One digitizer per subdetector

"Identity" factor 10^8 avoids precision loss in the conversion double -> int -> double

Configuring processors and modifiers

```
##########################################################
# A subdetector digitizer.  It instantiates one or more calorimeter hit
#  "modifiers", which together represent the full digitization process
.begin HcalBarrDigitizer

ProcessorType DigiSimProcessor

InputCollection  HcalBarrHits
OutputCollection HcalBarrRawHits
Raw2SimLinksCollection HcalBarrRaw2sim

ModifierNames HBlightYield HBcrosstalk HBlightCollEff HBPDQuEffic HBExpoNoise  HBGaussNoise
HBdiscrim HBGain

# Parameters:
# modifierName    Type                 gainNom  gainSig  thresh   thrSig
HBlightYield     GainDiscrimination  10000000        0       1        0

# Crosstalk                             mean    sigma
HBcrosstalk      Crosstalk               0.020     0.005

# Smeared gain parameters:              gain   gainSigma  thresh   thrSig
HBlightCollEff   GainDiscrimination  0.0111     0.0029       1        0
HBPDQuEffic      GainDiscrimination    0.15          0       1        0

(...Truncated... see file cdcaug05_np.steer in DigiSim area)
.end ------------------------------------------------------------
```

(As many as needed)

Simple example: configuration for the tile HCal

- Scintillation: 100 eV / photon , or 10^{+4} photons/MeV

  Ex: a MIP at normal incidence on 0.5cm-thick scintillator deposits ~ 0.9MeV, or 9000 photons

  ==> use GainDiscrimination modifier with 10^{+7} photons/GeV and a threshold at 1 photon

```
#modifierName   type                  gainNom  gainSig  thresh   thrSig
HBlightYield   GainDiscrimination  10000000        0       1        0
```

- Light crosstalk:

  first neighbors: 1.5% to 2% --> (2.0 +/- 0.5) %

```
HBcrosstalk      Crosstalk                0.020     0.005
```

- Photon collection efficiency (QE~15%):

  9000 scint.photons/MIP --> 15 PE/MIP detected (cosmics measurements at NICADD)

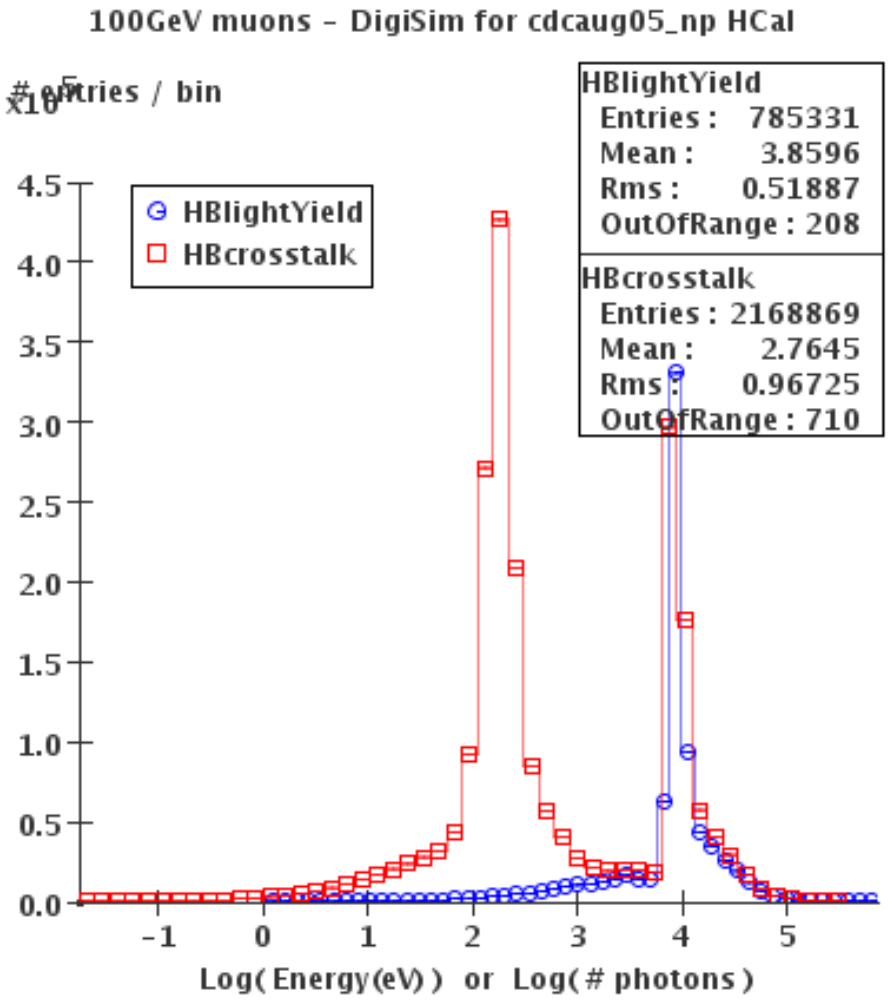
  15 / 0.15 = 100 incident photons => Eff_{coll} = 100 inc. / 9000 tot.scint. = 0.0111

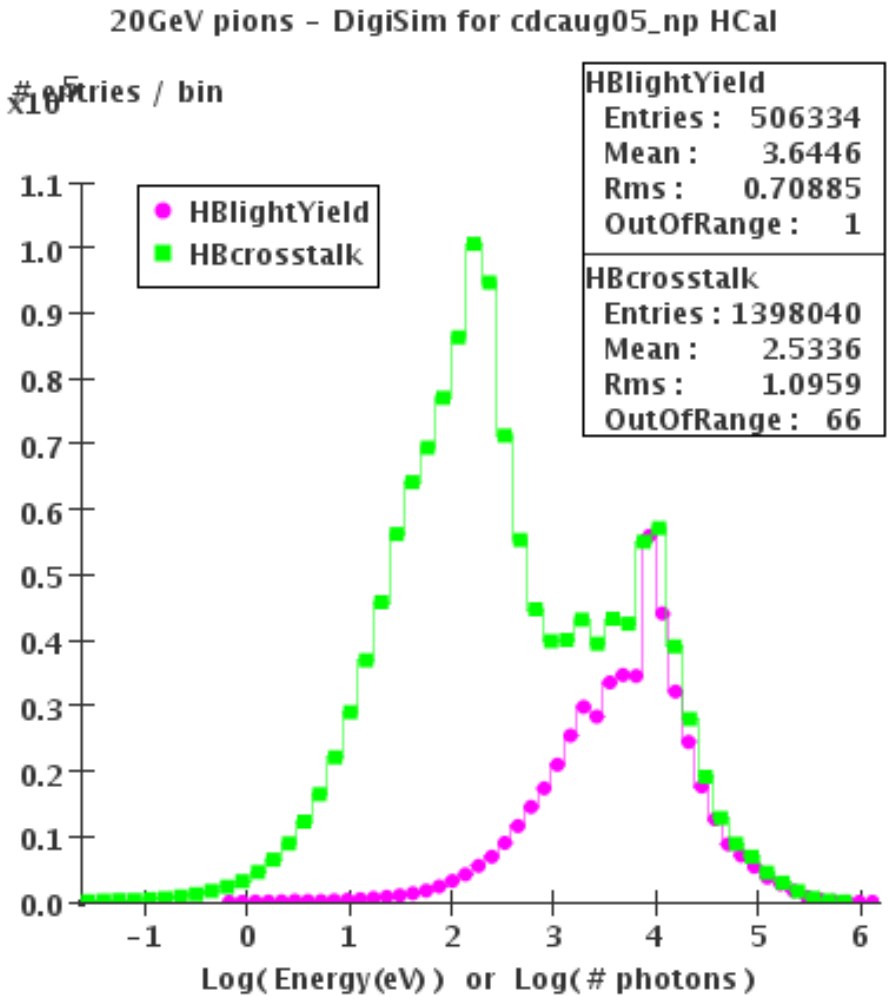
  Variance (Poisson): $sigma_N^2 = \langle N \rangle \rightarrow$ for $\langle N_{PE} \rangle = 15$, $(sigma_N / N) \sim 26\%$

  Therefore: $Eff_{coll} = 0.0111 \pm 0.0029$ => use GainDiscrimination with smearing

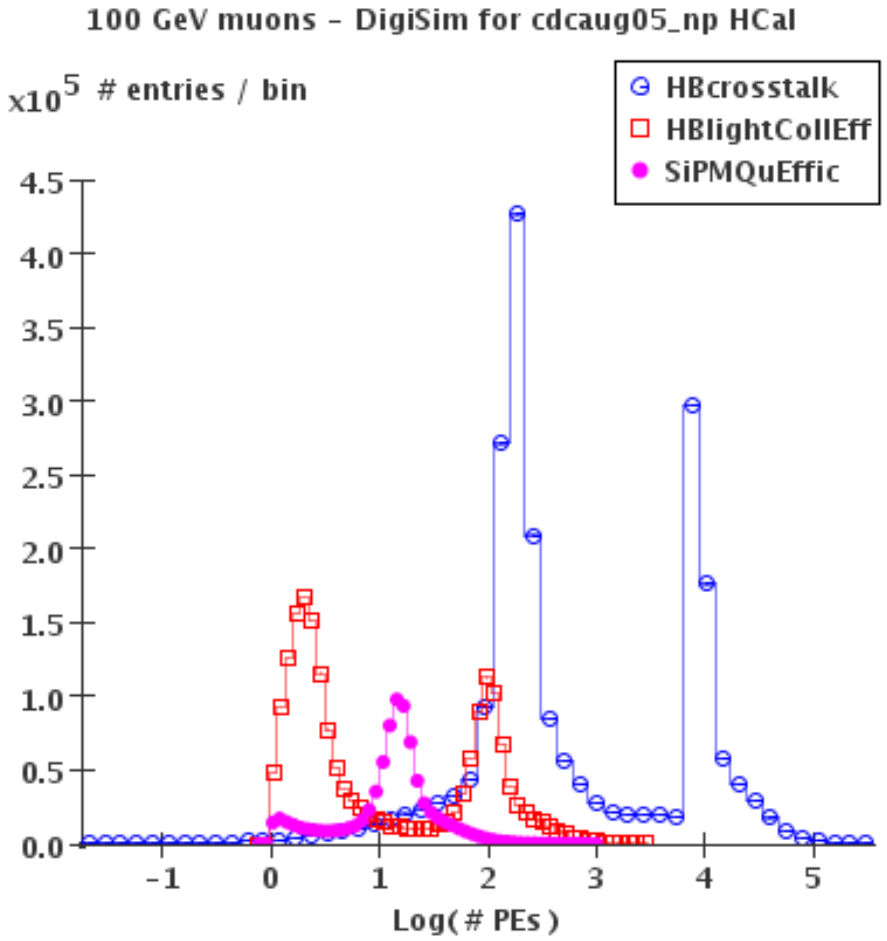
```
HBlightCollEff GainDiscrimination    0.0111   0.0029       1        0
```

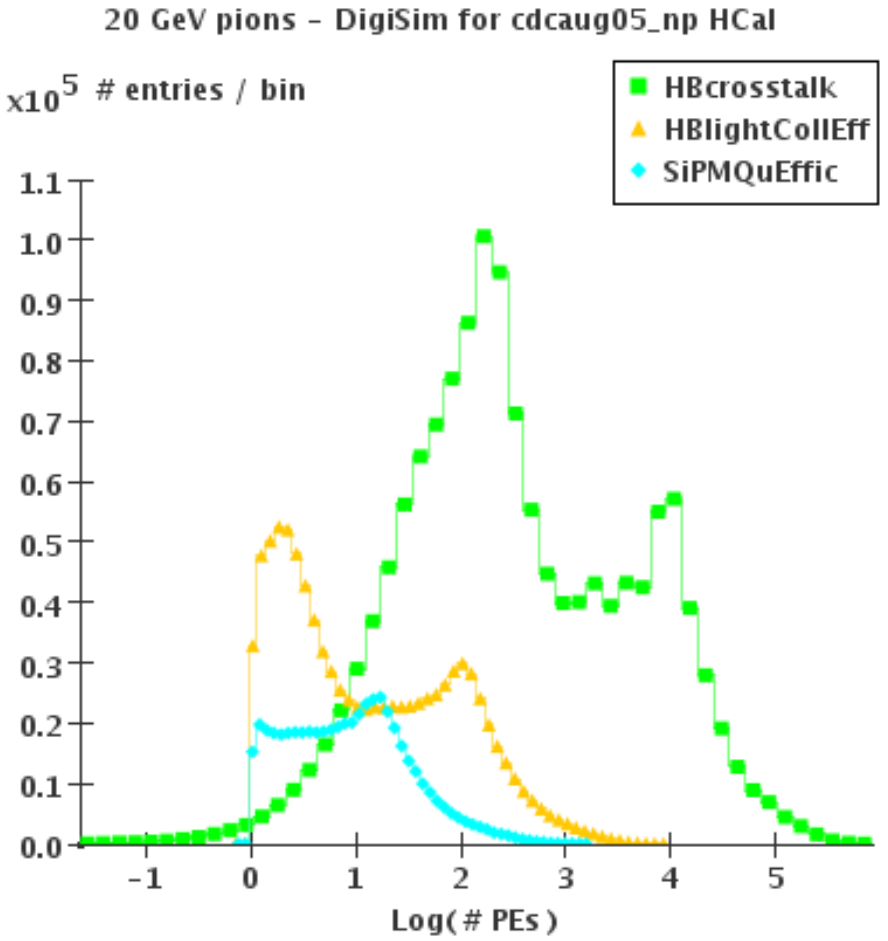
Light cross-talk





Photodetection





Simple example: parameters from the tile HCal

- Photosensor detection efficiency: QE ~ 15 %

```
HBPDQuEffic      GainDiscrimination    0.15          0       1        0
```

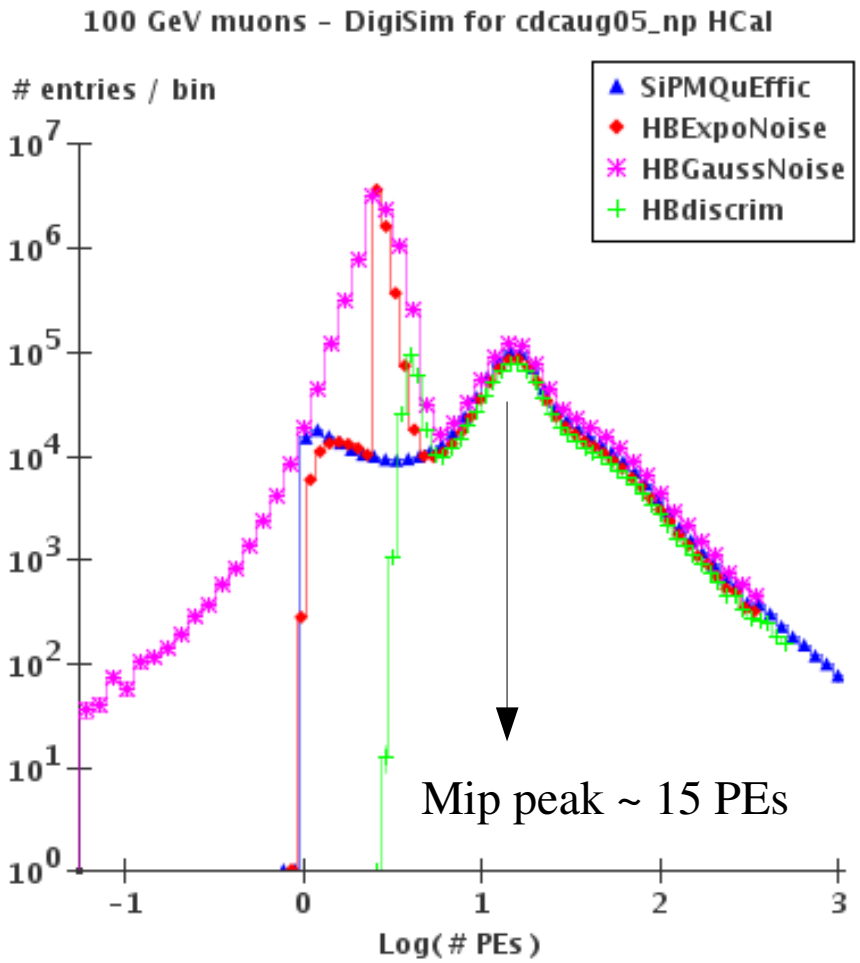
- Noise simulation:
  - Photosensor noise: exponential distribution (guess: mean 0.6)
  - Electronics noise: gaussian distribution (guess: mean 0, sigma 1.6, keep +/- tails)

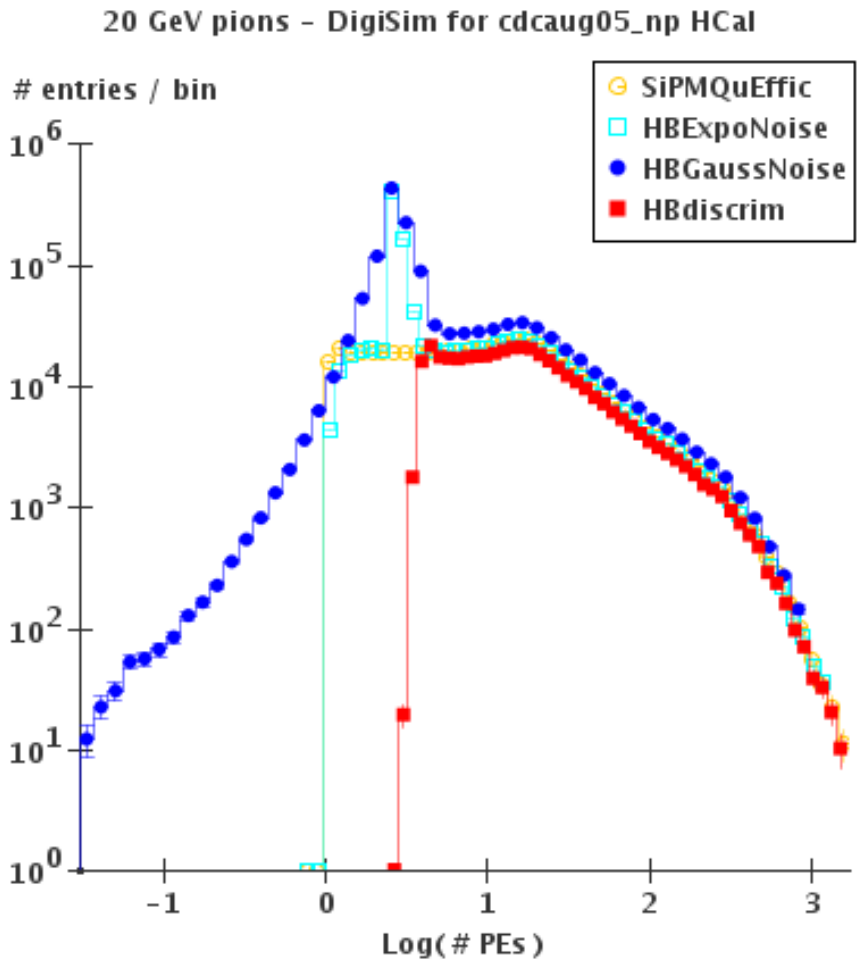
```
# GaussNoise parameters:        sys   be    Ecut   TimeNom  TSig   Mean  Sigma
# Note: sigma<0 means that threshold acts on absolute value only
HBGaussNoise   GaussianNoise      3     0    2.5       100   100    0.0  -0.58

# ExponentialNoise parameters:  sys    be   Ecut   TimeNom  TSig   Mean
HBExpoNoise    ExponentialNoise   3     0    2.5       100   100   0.23
```

- Discrimination: ¼ MIP cut ~ 4 PEs: threshold at 4 +/- 0.25 (on abs value)

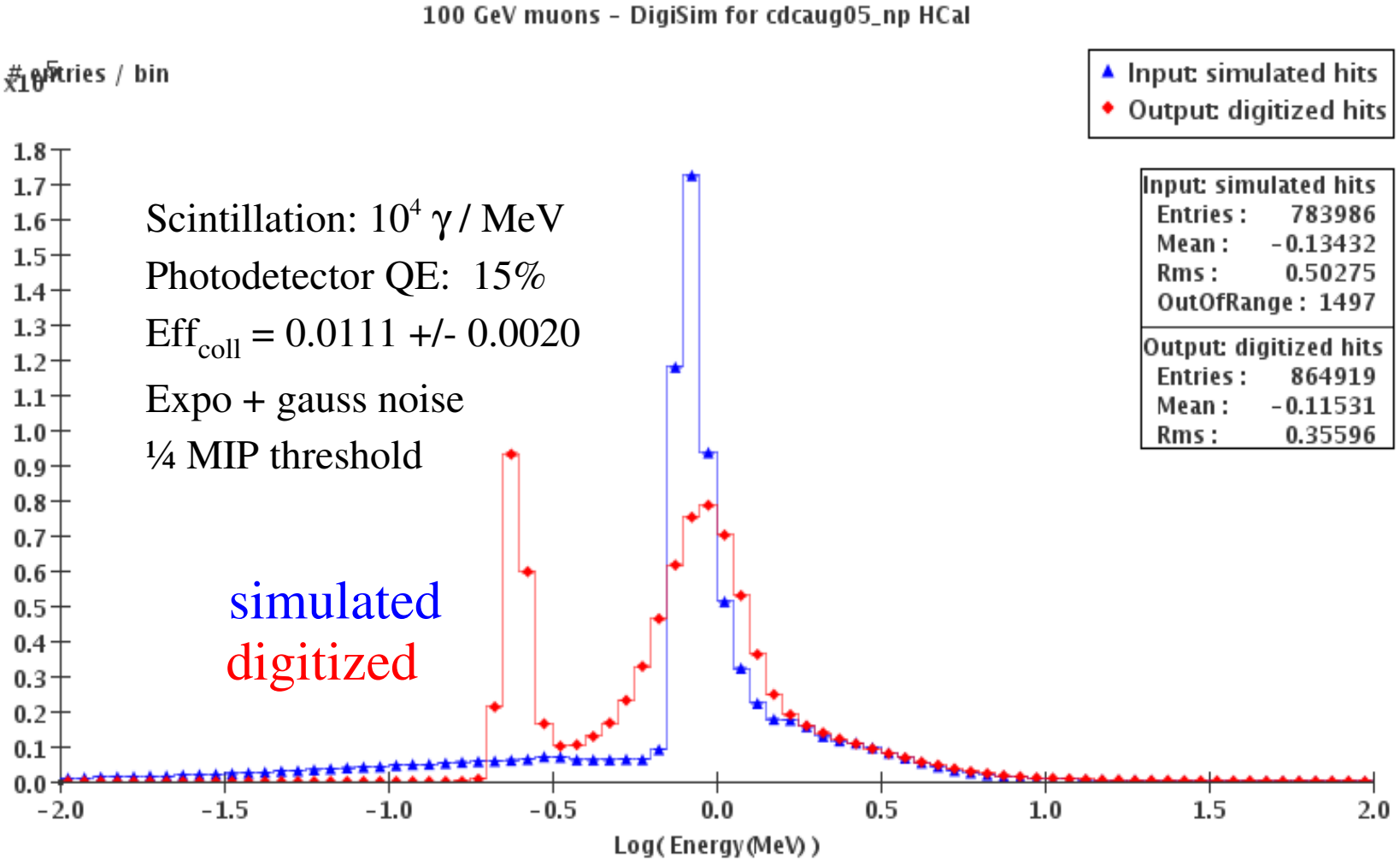
```
# Discrimination                                threshold      sigma
HBdiscrim    AbsValueDiscrimination                   4       0.25
###HBdiscrim GainDiscrimination       1       0       4       0.25
```

Noise and discrimination





HCal scintillator digitization (preliminary)



HCal scintillator digitization (preliminary)