Pandora Analysis Calibration Binaries Explained

Steve Green

November 6, 2014

This document is designed to give a brief introduction on how to perform the the full calibration produced, digitisation and PandoraPFA calibration, which must be implemented during each change in detector model or reconstruction software.

This is an iterative process. Each of these scripts takes in a series of single particle root file, which must have been produced from a recent version of Pandora Analysis (revision 1668 or later). To perform the full calibration produced you need to have a series of single particle root files containing:

- Photons at a single fixed energy. Generally take to be 10 GeV, but can be changed if desired.
- KaonL either (a) single fixed energy or (b) a range of energies. If you want to implement non linearity corrections you must have a range of energies. Generally the single fixed energy is taken as 20 GeV.
- Muons at a single fixed energy, which **must** be taken as 10 GeV as this is our definition of a minimum ionising particle (MIP).

1 Digitisation

1.1 ECalDigitisation_ContainedEvents

This script is run on the root file outputs for the single fixed photon energy and aims to set CalibrECAL.

Inputs:

1. True energy of photon
2. Full path to photon root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.

4. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

After running this script on the root files you rescale CalibrECAL by:

$$\frac{\textit{True Photon Energy}}{\textit{Mean Reconstructed Calo Hit Energy}} \tag{1}$$

Iteration is now performed. By iteration I mean you must reproduced the photon single particle root files, by rerunning the photon LCIO files through Marlin with an updated version of CalibraECAL. You then rerun the script again to see if the mean reconstructed calo hit energy is within acceptable limits. In this case I have been using $\pm 2\%$ of the true photon energy as the limits for my iteration. If after rerunning the mean is outside the limits you keep reproducing the root files and updating CalibrECAL until the mean reconstructed calo hit energy is within the acceptable limits.

1.2 HCalDigitisation_ContainedEvents

This script is run on the root file outputs for the single fixed KaonL energy and aims to set CalibrHCALEndcap and CalibrHCALBarrel.

Inputs:

1. True energy of kaonL
2. Full path to kaonL root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.
4. Detector component. Options are either "Barrel" or "Endcap". This parameter is used as you perform separate digitisation for the Barrel and EndCap, but the code is identical.
5. Lower CosTheta Cut. This is the lower limit of cos theta (polar angle), which you are using to define the Barrel or EndCap.
6. Upper CosTheta Cut. This is the upper value of cos theta (polar angle), which you are using to define the Barrel or EndCap.
7. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.
8. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

I have been defining the Barrel to have cos theta values between 0.2 and 0.6.

I have been defining the Barrel to have cos theta values between 0.8 and 0.9.

After running this script on the root files you rescale CalibrHCALBarrel and CalibrHCALEncap by:

$$\frac{\textit{True KaonL Energy}}{\textit{Mean Reconstructed Calo Hit Energy (for barrel and endcap setting respectively)}} \tag{2}$$

Iteration is now performed as it was for the ECalDigitisation script. You keep iterating until the mean reconstructed calo hit energy is within acceptable limits. In this case I have been using $\pm 2\%$ of the true kaonL energy as the limits for my iteration.

1.3 HCalDigitisation_DirectionCorrectionDeistribution

This binary (and the next) is designed to address the digitisation of the HCAL ring (CalibrHCALOther). This script is run on the root file outputs for the single fixed KaonL energy.

Inputs:

1. True energy of kaonL
2. Full path to kaonL root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.

This script joins together all the histograms (from all input root files) of direction corrections applied to the cells within the HCal in the barrel, endcap and other (other is ring) for the kaonL samples. There is no iteration required for this step. The mean direction corrections for each of the three distributions (barrel, endcap, other) should be recorded and saved for the next step.

1.4 HCalDigitisation_Ring

This binary carries on with the digitisation of the HCAL ring. This script is run on the root file outputs for the single fixed muon energy.

Inputs:

1. True energy of muon
2. Full path to muon root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.

$$Calibr_{Ring} = Calibr_{Endcap} * \frac{\langle DirCorr_{Endcap} \rangle}{\langle DirCorr_{Ring} \rangle} * \frac{MIP\ Peak_{Endcap}}{MIP\ Peak_{Ring}} * \frac{Absorber\ Thickness\ Endcap}{Scintillator\ Thickness\ Endcap} * \frac{Scintillator\ Thickness\ Ring}{Absorber\ Thickness\ Ring}$$

The output for this binary is a plot of the direction corrected sim calorimeter hit energies (ADC current) for the HCal barrel, endcap and other. These plots should show a clear MIP peak in all cases.

CalibrHCALOther is defined to be equal to CalibrHCALEndcap multiplied by the ratio of the MIP peaks for the endcap and other (from HCalDigitisation_Ring) and multiplied by the ratio of the absorber scintillator ratio (from gear files) and multiplied by the ratio of direction corrections (from HCalDigitisation_Ring):

This is somewhat ugly, but we have no events confined entirely within the HCal ring therefore, we infer CalibrHCALOther from CalibrHCALEndcap given the ring and endcap are of the same class in Mokka so should behave similarly. Again there is no iteration at this step.

2 PandoraPFACalibration

2.1 PandoraPFACalibration_HadronicScale_MipResponse

This binary carries dines Mip response in PandoraPFA, which is used for things such as isolation cuts. This script is run on the root file outputs for the single fixed muon energy.

Inputs:

1. True energy of muon
2. Full path to muon root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.

This script makes plots of the direction corrected calo hit energy for the HCal, ECal and Muon Chamber. The constants ECalToGeVMIP, HCalToGeVMIP and MuonToGeVMIP are then defined as the constant, which is needed to scale the observed MIP peak in these plots of the direction corrected calo hit energy plots so that the peak lies on 1 (i.e. a single mip like response). This converts energy measurements into a number of normally incident mips required to deposit the same amount of energy. Again no iteration here.

2.2 PandoraPFACalibration_EMScale

This script is run on the root file outputs for the single fixed photon energy and aims to set ECal/HCalToEMGeVCalibration.

Inputs:

1. True energy of photon.
2. Full path to photon root files. Wildcards can be used here.
3. Output path to send the calibration text file and the output plot to.
4. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

This binary makes a Gaussian fit to the PFO energy of the photons and ECalToEMGeVCalibraiton is scales until the peak in the PFO energy matches the true energy. I have been defining acceptable limits as the PFO energy falls within 0.25% of the true photon energy. Then we define HCalToEMGeVCalibration to equal ECalToEMGeVCalibraiton.

2.3 PandoraPFACalibration_HadronicScale_ChiSquaredMethod

This script is run on the root file outputs for the single fixed kaonL energy and aims to set ECal/HCalToHadGeVCalibration.

Inputs:

1. True energy of kaonL.
2. Full path to photon root files. Wildcards can be used here.
3. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.
4. Output path to send the calibration text file and the output plot to.

This forms the 2D histogram of ECalToHad energy vs HCalToHad energy. A line of best fit is applied to the data set, once a cut is applied to remove leakage effects and the intercepts of this best fit are used to rescale ECal/HCalToHadGeVCalibration until both intercepts match the available energy of the kaonL being simulated (i.e iteration required here). I have again been defining acceptable limits as the PFO energy falls within 0.25% of the true kaonL energy.

2.4 PandoraPFACalibration_HadronicScale_TotalEnergyMethod

This script is run on the root file outputs for the single fixed kaonL energy and is an alternative method for setting ECal/HCalToHadGeVCalibration.

Inputs:

1. True energy of kaonL.

2. Full path to photon root files. Wildcards can be used here.

3. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.

4. Output path to send the calibration text file and the output plot to.

Performs a Gaussian fit to total PFO energy, once both ECal/HCalToHad have been independently rescaled. The mean is fixed and the ideal rescaling constants (one for ECalToHad one for HCalToHad) minimise the standard deviation of the Gaussian fit. The scale factors produced for the ideal fit are used to rescale ECal/HCalToHadGeVCalibration. Iteration of this procedure keeps happening until the rescaling factors are within the range 1 ± 0.0025 i.e 0.25% accuracy. The rescaling factors step in 0.01, as it scales the 2D parameter scale of scaling factors, so the limit actually corresponds to finding a minimum to 1% accuracy in rescaling factors.

2.5 PandoraPFACalibration_HadronicEnergyGaussianFit

This script is run on the root file outputs for the single fixed kaonL energy and is used for applying Gaussian fits to reconstructed kaonL PFO energy, which can be used for non linearity corrections.

I have not been implementing the non linearity corrections in my optimisation studies in an attempt to keep the analysis as clear as possible.

Inputs:

1. True energy of kaonL.

2. Full path to kaonL root files. Wildcards can be used here.

3. Output path to send the calibration text file and the output plot to.

4. Non linearity constants on or off (just sets the labels for the output)

5. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.

6. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

This binary produces the Gaussian fit to and histogram of PFO energy for the kaonL data. A text file contains the mean, standard deviation and amplitude of the gaussian fit is also produced. It would be the mean and true energy used as inputs for non linearities if they were desired.