HCal Calibrations

S.Seeds

June 25, 2024

Contents

1 Purpose of this Document

This document is intended to provide an instruction on the workflow order for all necessary in-beam calibration procedures for the SBS Hadron Calorimeter (HCal) and to give a comprehensive description of each. This is intended to be used in conjunction with calibration scripts currently hosted on github. This document is not intended to replace the more general HCal instructions containing detailed hardware descriptions, online data monitoring, and fADC/F1TDC information/configurations on overleaf. It is also not intended to provide a comprehensive history of changes to the apparatus and hardware settings located on overleaf.

2 Brief description of the SBS Hadron Calorimeter

2.a Description of Apparatus

The SBS Calorimeter consists of 288 (24 rows by 12 columns) of single energy-sampling modules. Each 15cm by 15cm by 1m module can be referred to as a separate *channel* or *block* and consists of 40 layers of interleaved steel and scintillator. The steel acts as an absorber for incoming hadrons and promotes hadronic showers which deposit energy from a single event potentially into many adjacent modules. The electromagnetic energy contained in these showers is sampled by the interleaved scintillator (PP0 2,5-Diphenyloxazole) and transduced into light. This light is channeled into a wavelength shifter (St. Gobain BC-484) and then into a waveguide which directs scintillated light into a single PMT on the end of the module (Photonis XP2262 or XP2282). The signal from each module is amplified (10x) and split with half sent to a dedicated ADC channel (fADC250) and half sent to both trigger logic and a dedicated TDC channel (F1TDC). In addition to front-end (FE) trigger logic, HCal is equipped with a pulsed LED system to monitor quantum efficiency and cosmic paddles (2x on both top and bottom of HCal). For more information, consult the detailed HCal Instructions.

2.b Configurations of HCal

HCal is not intended to undergo any internal configuration changes to the hardware. The detector is, however, moved between kinematics to optimize acceptance matching between HCal and the electron arm (BigBite). See table 1 and table 2.

2.c HCal Data

Each recorded event potentially contains the following information important for calibration procedures contained in this document:

Configuration	1	4	7	11	14	8	9
HCal Distance (m)	13.5	11.0	14.0	14.5	14.0	11.0	11.0
HCal Angle (deg)	33.5	31.9	16.1	13.3	17.3	29.4	22.0
Number of TDC Calibrations	1	1	1	1	1	1	1
Number of ADCt Calibrations	1	1	1	1	1	1	1
Number of ADC gain Calibrations	1	2	1	1	1	1	1

Table 1: GMn Kinematics

Configuration	All
HCal Distance (m)	17.0
HCal Angle (deg)	34.7

Table 2: GEn Kinematics

- All detector blocks (expert replay only, 288 channels each event)
    - ADC pedestal-subtracted integrated value
    - ADC pedestal-subtracted amplitude
    - ADC interpolated threshold crossing time
    - ADC channel index (ID)
    - TDC leading edge (le) time
    - TDC channel index
    - ADC samples (full waveforms, 4ns bins)
    - ADC sample number (default 288 channels by 40 bins, bins configurable)
- Primary cluster (all blocks)
    - Cluster Sum ADC pedestal-subtracted integrated value
    - Dispersive X integrated-ADC-weighted centroid
    - Transverse Y integrated-ADC-weighted centroid
    - Primary block ADC pedestal-subtracted integrated value
    - Primary block ADC time
    - Primary block TDC le time
- All cluster
    - Cluster Sum ADC pedestal-subtracted integrated value
    - Dispersive X integrated-ADC-weighted centroid
    - Transverse Y integrated-ADC-weighted centroid
    - Primary block ADC pedestal-subtracted integrated value
    - Primary block ADC time
    - Primary block TDC le time

- Primary cluster, all blocks
  - ADC pedestal-subtracted integrated value
  - TDC le time
  - Dispersive X
  - Transverse Y

Note that additional information exists by default on the tree which is not mentioned here. Table 3 provides additional information on HCal analysis variables. The *Array* column indicates the type of array NTuple which exists per signal/data type.

2.d General Note

HCal is designed and intended to measure energy, timing, and position of elastically scattered recoil nucleons from various targets. As a matter of course, most calibration steps require a relatively clean selection of elastic events to perform in-beam and therefor useful calibrations. This document is not intended to describe elastic kinematics or describe in detail how elastic selection is accomplished with the output root file branch structure inherent to SBS experiments via SBS-offline. Accurate calibration of HCal for new data will likely require expertise in these areas along with a working knowledge of C^{++} and *Root*. Additionally, a working build of the Hall A analyzer, SBS-offline, and *Root* (JLab environment 2.5 assumed for scripts detailed here) are needed for proper analysis of HCal data and calibrations.

3 The Database

3.a Files

All raw data is decoded and replayed via SBS-offline. Experimental configurations and calibration parameters are contained in SBS-replay. The goal of all essential calibrations is to pass constants to

db_sbs.hcal.dat

This file is located in the database directory **$SBS-replay/DB**. The purpose of these constants is to ensure that HCal branches on replayed data accurately represent physical quantities of interest.

3.b Variables and Constants

All calibrations have an associated variable name which identifies the associated constants that immediately follow them. For example, *sbs.hcal.emin 0.005* indicates that the total energy deposited in any channel in HCal must exceed 5 MeV to be considered for as a cluster element. The definitions of all variables are available in the source code for SBS-offline — adequate documentation of SBS-offline is forthcoming. Some of these constants are single-valued and represent a factor applied to all channels (for example raw TDC unit (RTU) to ns conversion; *sbs.hcal.tdc.calib*), some have multiple values which represent factors applied to multiple degrees of freedom (for example space between channels; *sbs.hcal.dxdydz*), and some have many values which represent factors applied to each channel (for example TDC time offset; *sbs.hcal.tdc.offset*). The relevant type and variable name will be indicated as necessary in the following section.

SIGNAL TYPE	DATA TYPE	SIZE	UNITS	ARRAY
All Block	Int. ADC	All Cells (288)	pC	Fixed
	ADC amp.	All Cells (288)	mV	Fixed
	ADC time	All Cells (288)	ns	Fixed
	ADC ID	All Cells (288)	idx	Fixed
	TDC le	All Cells (288)	ns	Fixed
	TDC ID	All Cells (288)	idx	Fixed
	samples	Cell x Bin (11520)	RAU	Fixed
	N samples	Cell x Bin (11520)	N	Fixed
	samples ID	Cell x Bin (11520)	idx	Fixed
Primary Cluster	Sum int. ADC	1	pC	Fixed
	X centroid	1	m	Fixed
	Y centroid	1	m	Fixed
Primary Block	Int. ADC	1	pC	Fixed
Primary Block	ADC time	1	ns	Fixed
Primary Block	TDC le	1	ns	Fixed
All Cluster	Sum int. ADC	N Cluster	pC	Variable
	X centroid	N Cluster	m	Variable
	Y centroid	N Cluster	m	Variable
Primary Block	Int. ADC	N Cluster	pC	Variable
Primary Block	ADC time	N Cluster	ns	Variable
Primary Block	TDC le	N Cluster	ns	Variable
Primary Cluster All Block	Int. ADC	N Block	pC	Variable
	TDC le	N Block	pC	Variable
	X	N Block	m	Variable
	Y	N Block	m	Variable

Table 3: Analysis Variables

3.c Timestamps

Constants located in the SBS database are organized with *timestamps* to account for changes applied to hardware or data acquisition throughout the lifetime of the detector. The proper use of these timestamps is as follows:

1. Identify the location in *db_sbs.hcal.dat* where the desired timestamp should exist in chronological order.
2. Provide a timestamp of the form (no spaces between dashes) to this location:

   - - - - - - -[YYYY-MM-DD HH:MM:SS]

3. List all relevant database variables which contain new constants and provide new constants, leaving duplicate variables with different constants after other timestamps where needed.

Note that those variables and associated constants that do not follow a timestamp are considered to be active for all time before the first timestamp. If no timestamp is provided, then these variables and associated constants are for all time.

4 Cosmic Calibrations and Setting High Voltage

HCal energy calibrations can be performed from cosmic muons over many events and high voltage targets can be calculated from these data. The process, broadly speaking, is as follows:

1. **Alphas** Extract alpha parameters by PMT from LED data with a single fixed LED intensity over many HV settings. Alpha parameters are fits to exponential rise expected from gain response as a controlled function of HV alone.
2. **Cosmic Energy** Build energy spectra per PMT from cosmic muons per channel. Via many landau fits, extraction of MPV from each of these spectra can be compared with target values based on known muon energy deposition and the saturation point on each PMT.
3. **HV and Gain** Calculate target HV or extract gain coefficients from these data to shift MPV to target values.

Preliminary cosmic calibrations and commissioning of HCal requires configuration of the pulsed LED system and cosmic paddles in HCal which will not be discussed here. For a more detailed discussion of this, see HCal Instructions.

4.a Alpha Extraction

The primary measure of a PMT's gain profile is it's alpha parameter. In terms of the mean of ADC spectra vs various HV settings, it can be expressed as follows:

$$ADC_{peak} = ADC_{ref}\frac{HV_{set}}{HV_{ref}}^{\frac{1}{n_d\cdot\alpha}} \tag{4.1}$$

Here ADC_{peak} is the mean of the ADC spectra (dependent variable), HV_{set} is the HCal HV setting (independent variable), ADC_{ref} scales each spectral mean and is the ADC mean corresponding to the lowest HV setting, HV_{ref} normalizes the independent variable and is the lowest HV setting, n_d is the number of dynodes (or amplification stages) in the PMT, and α is the alpha parameter (allowed to float for the fit) which defines the gain profile for the PMT.

Alpha parameters are, in principle, unique per PMT and depend on the number of dynodes and construction parameters of each. In HCal, we employ two types of PMTs - Photonis XP2262 "CMU" 12-dynode and Photonis XP2282 "JLab" 8-dynode. Further, constraints on the dependent variable must arise from photoelectron (PE) analysis of the PMT's plateau region, or the range of HV settings which optimize the PE response from the PMT. For HCal, this analysis was performed primarily by Vanessa Brio.

The scripts to extract alpha parameters can be found on the Hall A counting house computers in the following location:

- /adaqfs/home/a-onl/sbs/hcal_devel

They are:

- alpha_extraction.C
- LED_spectral_analysis.C

These scripts are hardcoded to function only with LED runs taken before GMn commissioning, so no detailed instructions for their use are included here. That said, they are commented and the formalism can by replicated as necessary.

4.b Cosmic Calibration

The gain response from PMTs in HCal must first be obtained in the form of alpha parameters *before* cosmic analysis is performed. With these parameters in hand, the process is as follows:

1. Calculate expected energy deposition per block in HCal from known cosmic muon energies at sea level (**14 MeV**).
2. Determine the maximum expected energy deposition in HCal for the upcoming experiment. For GMn it is **700 MeV** at $Q^2 = 11\ GeV$.
3. Determine the ADC saturation point in raw ADC units (RAU). For GMn:

$$\begin{aligned} ADC_{sat} &= 4095(RAU) \cdot 1.5(\textit{dynamic range of amplifier/fADC}) \cdot 0.5(\textit{cable attenuation}) \quad (4.2) \\ &= 3071(RAU) \quad (4.3) \end{aligned}$$

4. Determine target signal maximum per channel in RAU with these data. For GMn:

$$\begin{aligned} ADC_{target} &= ADC_{sat} \cdot factor_{Emax} \quad (4.4) \\ &= 3071(RAU) \cdot \frac{14(MeV)}{700(MeV)} \quad (4.5) \\ &= 61(RAU) \quad (4.6) \end{aligned}$$

   This is RAU_{target}. This target can be converted to pC (as necessary). For GMn (with more accurate cable attenuation and saturation parameters), $pC_{target} = 8.12(pC)$.

5. Take many cosmic events with cosmic paddle trigger in HCal such that each channel is populated with a minimum of 1000 events.
6. Fit these distributions with landau (or alternatively skewed-gaussian) functions and extract the MPV (or mean) from each per channel. This is $RAU_{measured}$.
7. Grouping the number of dynodes into α and denoting with α^*, calculate the target HV as follows:

$$HV_{target} = \frac{HV_{set}}{\left[\frac{RAU_{measured}}{RAU_{target}}\right]^{\frac{1}{\alpha^*}}} \tag{4.7}$$

The script to calculate target HV from a single cosmic run and extracted alpha parameters can be found on the Hall A counting house computers in the same location as before:

- /adaqfs/home/a-onl/sbs/hcal_devel

It is:

- cosmic_gain_match.C

5 When to Calibrate

For all in-beam calibrations, expected energy and timing values are determined from elastic kinematics and as such HCal requires a significant number of elastic events in order to populate its 288 channels. For this reason alone, it is not possible to calibrate by run number or even by magnetic field setting (in many cases). However, each calibration for the detector is not expected to vary significantly based upon normal variations within an experimental configuration. For instance, a change in the SBS magnetic field will not affect scattered hadron energy or time of flight significantly. Further, monte carlo can be used to inform the expected changes between configurations. We observe a significant variance in HCal sampling fraction (the fraction of scattered hadron kinetic energy transduced into signal at the ADC) and, by extension, energy spectra.

In order to standardize the calibration workflow and remain sensitive to monte carlo expectations, **HCal is expected to be calibrated by configuration**. This entails a calibration set for ADC time offsets, ADC time additional cluster elastic cut, TDC time offsets, and ADC gain (energy) coefficients (at minimum). Additionally, **if any hardware changes exist within a configuration, two calibration sets for all variables should be generated for that configuration**. For example, some hardware changes that might necessitate recalibration are high voltage changes, ADC time latency adjustment, ADC dynamic range adjustment. A breakdown of the total number of calibration points by experiment and configuration can be found in table 1 and table 2.

For initial commissioning in an experimental hall, one should consider alpha extraction and cosmic calibration as discussed earlier.

6 Calibration Framework and Scripts

The SBS-HCal calibration framework consists of several source files and headers, but does not precompile. These files and headers reduce the overhead per calibration script and provide several tools for further analysis. A full list of these files and brief descriptions of each follow. One can review the files, and the readme, on github.

- Header Files (located under *hcalCalibration/SBS/include/*): These files serve as a declaration for all necessary class objects, structs, and functions.
    - **sbs.h**: This header includes all necessary scripts and headers to perform HCal calibrations and can be sourced in lieu of each individually.
    - **MC.h**: This header is an auto-generated class component for raw G4SBS (SBS monte carlo) output which enables intuitive variable links for all active branches.
    - **hcal.h**: This header contains the namespace *hcal::* which includes many static detector parameters. Additionally, this header establishes the target and configuration class objects which return static experimental configuration and target parameters by experiment/configuration and target respectively.
    - **struct.h**: This header establishes several structs which are used to store run-specific variables, such as beam charge, and to hold calibration constants obtained from database and generated during calibrations, such as tdc offset parameters.
    - **util.h**: This header declares many utility functions used widely throughout calibrations. More details will follow in the *scripts* subsection.
- Configuration Files (located under *hcalCalibration/SBS/config/*): These are broken down by experiment and consist of two *.csv* files.
    - **runlist**: This spreadsheet is editable and establishes beam quantities and calibration timestamps per run for further analysis. The expected format is $< experiment > runs\_pass < replay - pass >$*.csv*.
    - **cutlist**: This spreadsheet is editable and establishes elastic cuts across many subsystems for wide elastic cuts. Tight elastic cuts are expected to be passed by calibration script. The expected format is $< experiment > cuts\_pass < replay - pass >$*.csv*.
- Framework Scripts (located under *hcalCalibration/SBS/src/*): These files include the algorithms necessary to perform utility tasks and to make classes dynamic.
    - **MC.C**: This script provides command line functionality for the *MC.h* class and is auto-generated by root.
    - **hcal.C**: This script includes all target-specific and configuration-specific experimental parameters which are accessed with *hcal.h* classes. See table 4 and table 5 for details.
    - **util.C**: This script provides several utility functions which are used in the calibration process. See table 6 for details.
- Calibration Scripts (located under *hcalCalibration/SBS/energy/* and *hcalCalibration/SBS/timing/*): These scripts are used to generate database parameters pursuant to calibration of HCal. Details can be found in the following sections.
    - **ecal.C**: This script generates **sbs.hcal.adc.gain** parameters by calibration set.
    - **tdc_align.C**: This script generates **sbs.hcal.tdc.offset** parameters by calibration set.

Variable	Switch	Description
ebeam	experiment/configuration	Average beam energy in GeV
bbtheta	experiment/configuration	Angle θ BigBite (electron) arm wrt exit beamline (deg)
bbdist	experiment/configuration	Distance from the target to the BigBite magnet in m
sbstheta	experiment/configuration	Angle θ SBS magnet (hadron arm) wrt exit beamline (deg)
sbsdist	experiment/configuration	Distance from target to the SBS magnet in m
hcaldist	experiment/configuration	Distance from target to the hadron calorimeter in m
hcaltheta	experiment/configuration	Angle θ HCal wrt exit beamline (deg)
sbsts	experiment/configuration	Timestamp preceding start of each configuration

Table 4: SBSconfig Class Variables

Variable	Switch	Description
target	target	Target
tidx	target	Target index (1:lh2,2:ld2,3:he3)
ltgt	target	Length of the target (m)
tarrho	target	Target density (g/cc)
crho	target	Density of {Al,Al,glass}
cdiam	target	Cell diameter (m)
cdEdx	target	Cell wall collisional stopping power (GeV*m/g)
cthick	target	Target cell thickness (m)
uwallthick	target	Upstream wall thickness (cm)
dwallthick	target	Downstream wall thickness (cm)
dEdx	target	Target collisional stopping power (GeV*m/g)
M_t	target	Average Target Mass (GeV)

Table 5: SBStarget Class Variables

  - **adct_align.C**: This script generates **sbs.hcal.adc.timeoffset** parameters and **sbs.hcal.adc.tmax** parameters by calibration set.
  - **tdc_tw.C**: This script generates **sbs.hcal.tdc.tw** parameters by calibration set.

Note that the environment following environment variables should be configured on user environment:

- **$OUT_DIR**: Path to output files. Scripts expect the following structure:
  - Timing calibration output: $OUT_DIR/hcal_calibrations/pass<replay-pass>/timing
  - Energy calibration output: $OUT_DIR/hcal_calibrations/pass<replay-pass>/energy
  - Energy MC calibration output: $OUT_DIR/hcal_calibrations/MC
- **$DB_DIR**: Path to database files. Structure should follow current database format in SBS-replay.

FUNCTION	TYPE	DESCRIPTION
getDate()	string	Returns today's date and time in mm/dd/yyyy format
readParam()	void	Reads a set of parameters into a vector from a txt file at input path. Assumes endl delimiter
countSets()	int	Counts the number of calibration sets for a given calibration parameter
readDB()	void	Reads database-style file for 288 (hcal::maxHCalChan) elements after a given moniker (type). Accepts 2 overloads
tsCompare()	void	Compares two timestamps and returns the one that is later in time. Format must be similar
ReadRunList()	void	Reads run list .csv file
ReadCutList()	void	Reads cut list .csv file
sethcalaxes	void	Sets HCal axes by kinematic (from hall coordinates)
getxyhcalexpect	void	Gets expected location of scattered nucleon assuming straight line projections from BB track
checkPID	void	checks particle id with information from observed dx distributions.

Table 6: Util.C Functions

7 Calibration Workflow

Calibrations of HCal should proceed in the order described in flowchart 1. The following subsections detail each of the five calibration categories. Note that the top three are all necessary calibrations for GMn.

8 Essential Calibration Procedures

8.a Timing Alignments/Cut-Calibrations

As a preliminary correction to all HCal timing, one should evaluate the raw TDC and ADC time values as a difference between the tree variable and the Hodoscope cluster mean time. This eliminates spread to timing distributions which result from trigger jitter where the reference for both Hodoscope and HCal is the primary trigger for the experiment. Any such jitter will cancel on the difference. Calibrations can improve this correction, but only those associated with Hodoscope and, as such, are outside of the scope of this document.

8.a.1 TDC

Precision F1TDC time is used for clean elastic selection and can be used for particle identification (PID) on some kinematics. Raw TDC values in HCal are a time difference in ns between the leading edge of an analog signal threshold crossing and the HCal TDC reference channel. The F1TDC records multiple hits on each event such that it is necessary to pass a good timing cut around which the best le time is selected per event. This reference channel is set to the primary experimental trigger. For more information, see HCal Instructions.

To effectively select elastic events with coincidence time in HCal with a single cut, alignment of all signal peaks is needed. The essential steps are as follows:

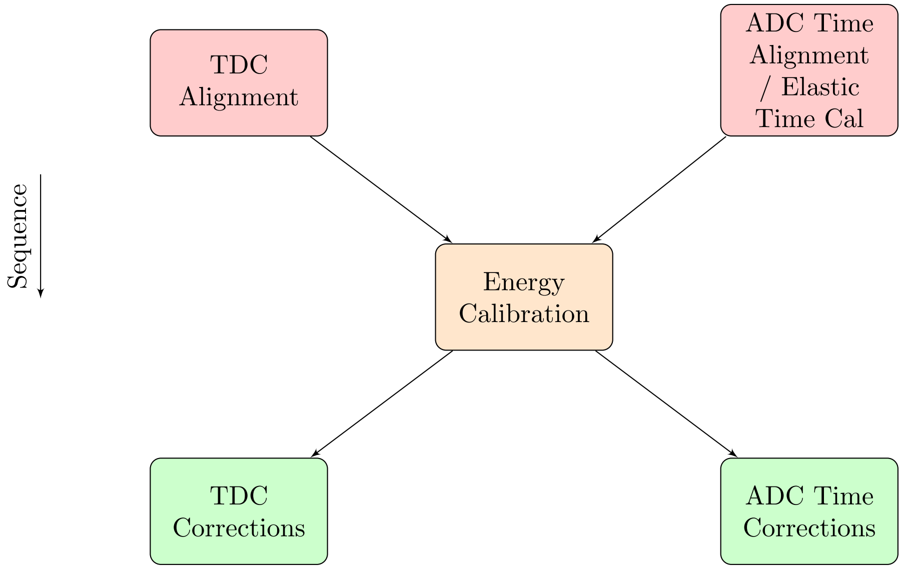


Figure 1: Calibrations flowchart. Sequence reflects dependence of later calibration procedures on output of previous calibration procedures.

1. Obtain a wide selection on elastic events with enough data to populate each channel in HCal with roughly 1000 events. Global cuts and physics cuts are already applied in the script for GMn along with run selection. All of these parameters are established by the configuration .csv files detailed earlier and entail elastic selection analysis by the user to populate new cut lists.
2. Loop over all events and obtain a TDC distribution per channel.
3. Fit the elastic peak in each distribution with a Gaussian and extract the mean.
4. Divide each of these mean values by the TDC calibration constant (located in database file with variable name *sbs.hcal.tdc.calib*) to convert to RTU.
5. Obtain previous TDC offsets per channel (*sbs.hcal.tdc.offset*, offsets list in order by channel in database file). These value are in RTU.
6. Add the previous TDC offset to the extracted mean to remove previous offset per channel.
7. Choose a target elastic peak location centered in the TDC window and divide by the TDC calibration constant to convert to RTU.
8. Obtain the difference between uncorrected TDC mean and target peak location per channel in RTU (TDC mean - target).
9. Provide a timestamp where necessary and update the database with these parameters in RTU after variable name *sbs.hcal.tdc.offset*.

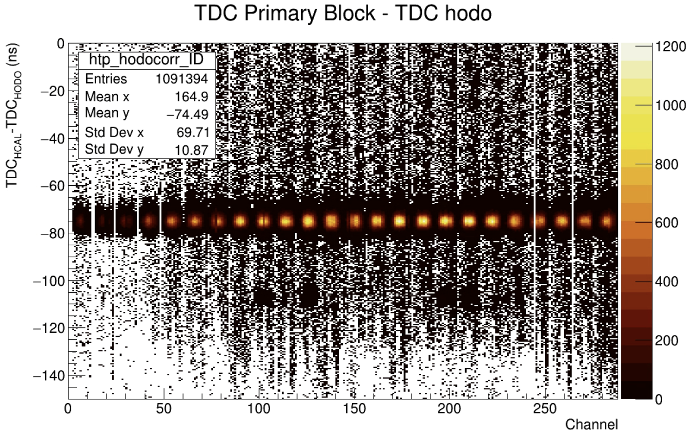


Figure 2: Hodoscope-corrected primary-block HCal TDC vs HCal ID. All data aligned.

The primary script used to calibrate and align TDC data during GMn is **tdc_align.C** located on github. This script has wide elastic selection per kinematic configured for GMn and all relevant data paths on the JLab farm ready for analysis. To run the script:

1. Do: *root ‘tdc_align.C(<(string)experiment>,<(int)configuration>,<(bool)quasi-replay>,<(int)pass>)’* where *<kinematic>* refers to available GMn kinematics and *<quasi-replay>*= *false*. One can also use the shell script for convenience. Do (spaces between inputs):
   ./run_tdc_align.sh <(string)experiment> <(int)configuration> <(bool)quasi-replay> <(int)pass>

   > Note: Additional configurations can be produced for use with this script. The SBSconfig and SBStarget classes must be updated by configuration and the runlist and cutlist spreadsheets must be filled by run.

2. Output quality plots can be found in the aforementioned output directory. TDC offsets can be found from this working directory under
   /parameters/tdcoffsets_class_<experiment>_conf<configuration>_pass<replay-pass>.txt.

Note that shell scripts exist in this working directory to submit these alignment jobs to the farm where applicable. Some changes to accommodate user environment are necessary.

To verify that new offsets are effective, one can run this script again in sequence with <qreplay>= *true*. This will quasi-replay the data with new TDC offsets and produce quality plots in the previously identified location. Properly aligned TDC signals should demonstrate a form similar to figure 2.

Note that these alignment offsets require good fits of TDC distributions by channel. As such, tuning and modification of **tdc_align.C** may be necessary.

8.a.2 ADC Time

HCal is equipped with flash analog to digital converters (fADC250) whose primary purpose is to provide integrated pulses in pC for measurement of energy in GeV. Each sample taken by the ADC has a 4ns width, however, and event timing can be extracted from a clean signal in the ADC. A linear interpolation method is used in SBS-offline to extract a corrected ADC time from the first bin that crosses threshold with respect the trigger. This value is written to the replayed data tree and can be used to improve clean elastic selection via coincidence time with the primary experimental trigger, especially where TDC data is not available.

Pursuant to this elastic selection, it is necessary to align all ADC time in a similar fashion to TDC described above. The essential steps are as follows:

1. Obtain a wide selection on elastic events with enough data to populate each channel in HCal with roughly 1000 events. As before, global cuts and physics cuts are already applied in the script for GMn along with run selection. Also as before, all of these parameters are established by configuration and cut .csv files detailed earlier.
2. Loop over all events and obtain an ADC time distribution per channel.
3. Fit the elastic peak in each distribution with a Gaussian and extract the mean.
4. Obtain previous ADC time offsets per channel (*sbs.hcal.adc.timeoffset*, offsets list in order by channel in database file).
5. Add the previous ADC time offset to the extracted mean to remove previous offset per channel.
6. Choose a target elastic peak location in the **second quarter** of the ADC time window. This location will allow time in the window for ADC waveforms to return to baseline.
7. Obtain the difference between uncorrected ADC time mean and target peak location per channel in (ADC time mean - target).
8. Provide a timestamp where necessary and update the database with these parameters in RTU after variable name *sbs.hcal.adc.timeoffset*.

The primary script used to calibrate ADC time during GMn is **adct_align.C** located in the aforementioned location. Running the script with *<qreplay>*$= false$ produces quality plots and offsets for both TDC data and ADC time data. All plots are located in the same directory and file as described for TDC alignments.

Running this script again in sequence with *<qreplay>*$= true$ produces quasi-replayed plots to verify alignments. Properly aligned TDC signals should demonstrate a form similar to figure 3. Note that, similar to TDC alignments, extraction of these alignment offsets require good fits of ADC time distributions by channel. As such, tuning and modification of **adct_align.C** may be necessary.

8.a.3 ADC Time Elastic Cut on Added Cluster Blocks

Using the ADC time available on the tree, one can obtain a better elastic selection by passing the mean and standard deviation of the observed ADC time peak to database. This will reject all primary clusters whose primary block arrives outside of three sigma about the mean elastic ADC time peak location in time. Additionally, one can also reject all additional blocks added to cluster which arrive outside of a three sigma difference in time with the primary block. The former of these is not yet implemented in SBS-offline, but the latter is.

The above ADC time script will aggregate all blocks and build a spectrum of ADC time values across the entire detector, fit this distribution with a Gaussian, and include this plot in the output

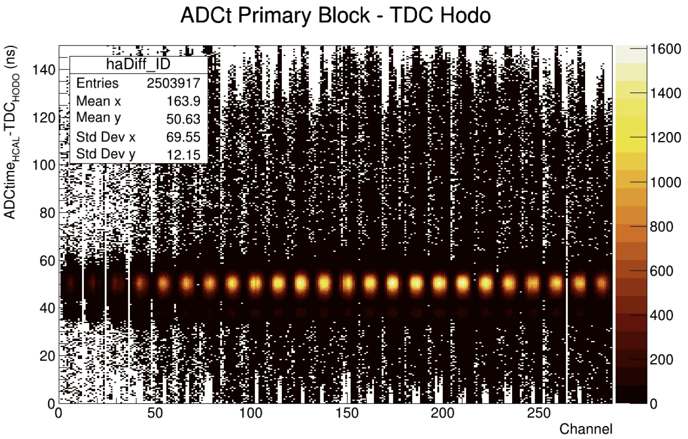


Figure 3: Hodoscope-corrected primary-block HCal ADCt vs HCal ID. All data aligned.

when run with *qreplay==true*. In the standard parameters location, a file will exist with the timestamp, precise database parameter, and value needed for this calibration. For reference, the parameter is *sbs.hcal.adc.tmax*.

8.b Energy

8.b.1 Monte Carlo

In order to provide a baseline for expected signals in HCal, monte carlo simulation results must be run by experiment and configuration. See G4SBS documentation for details on how to run the simulation and obtain viable MC output. To analyze the output data, a script exists to produce benchmark sampling fraction and energy results. This can be found on github. The steps to run the script are as follows:

1. Do: *root ‘hcalE_mc.C(<(string)experiment>,<(int)configuration>’* where *<configuration>* refers to available experimental configuration. One can also use the shell script for convenience. Do (spaces between inputs):
   ./run_hcalE_mc.sh <(string)experiment> <(int)configuration>

8.b.2 Primary Calibration

The essential aim of energy calibrations is to convert signal from ADC channels (in pC) to energy (in GeV) as accurately as possible. Making a tight selection on elastic events, elastic kinematics are used to calculate the expected kinetic energy (KE, ν) of elastically scattered hadrons resulting from e’ tracks reconstructed in BigBite. To obtain a tight selection on elastic events with BigBite, use of all active subsystems should be employed which include, but are not limited to:

- BBCal preshower pion cut (typically greater than 0.2 GeV)
- BBCal total energy cut (by kinematic)
- Grinch primary cluster cut where available (PID, typically N block greater than 1)

- Minimum GEM plane hits cut
- Primary e' track cut
- HCal coincidence time cut, applied to additional blocks added to cluster in addition to rejection of primary clusters
- W^2 cut on elastic peak

Cutting out all HCal event clusters whose primary block exists on the periphery of the acceptance (active area cut), the expected KE is summed over all blocks in the primary cluster. Summing all of these clusters over all events, we build a system of linear equations and minimize chi-squared to extract the most probable value of the gain coefficient per block which best reproduces the ADC response given expected KE per event.

$$\chi^2 = \sum_{i=1}^{n} \left(E_i - \sum_j c_j A_j^i \right)^2 / \sigma_E^2 \tag{8.1}$$

$$\sigma_E^2 \approx E_i \tag{8.2}$$

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}c_k} = 2 \sum_{i=1}^{n} \frac{1}{E_i} \left(E_i - \sum_j c_j A_j \right) A_k = 0 \tag{8.3}$$

$$\sum_i A_k = \sum_i \sum_j A_j c_j \frac{A_k}{E_i} \tag{8.4}$$

This first-pass approach follows the Hall C example for calibrating the HMS calorimeter. For HCal, subtle exceptions exist with this approach however. Foremost, the expected KE per event must be mitigated by the HCal sampling fraction (S_f) apprehended via monte-carlo simulations (MC, G4SBS). This factor, usually between 5%-10%, reduces the expected energy normalization per event. Second, the above scheme and formula is more accurately applied to electromagnetic calorimeters (EMCals) where the sampling fraction is much higher - nearly 100% - and $\sigma_E^2 \approx E$ is a fair approximation. In reality for HCal, the σ_E^2/E_i ratio is not nearly unity and must be evaluated per kinematic — ideally per channel. To account for the energy dependence of the response, one typically finds of the energy resolution the following quadrature sum:

$$\frac{\sigma}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c \tag{8.5}$$

Here a is the stochastic term that reflects the number of photoelectrons generated over events, b is the noise term that reflects electronics effects including pile-up, and c reflects dimensional variations and non-uniformity from detector geometry. While the stochastic term drops out of the chi-squared minimization after differentiation, the other terms, in principle, do not. To approximate the effects of both terms (noise and constant), a single factor R is introduced to modify S_f.

A more accurate way to extract this factor is via empirical fits to the energy spectra per channel (no calibration necessary for the σ_E^2/E ratio), however sufficient data to extract this ratio accurately is unlikely to exist on all channels. With the expected energy resolution of HCal between 30% and 70%, it is sufficient to estimate the energy spectra over all channels with MC and tune a single R

factor per kinematic (all channels) to match MC. Checks to verify this match are presented at the end of this section. The resulting scheme is as follows:

$$\chi^2 = \sum_{i=1}^{n} \left(E_i - \sum_j c_j A_j^i \right)^2 / \sigma_E^2 \tag{8.6}$$

$$S_f^* = S_f * R \tag{8.7}$$

$$E_i = \nu_i * S_f^* \tag{8.8}$$

$$\sigma_E^2 \approx \nu_i * S_f^* \tag{8.9}$$

$$\frac{\mathrm{d}\chi^2}{\mathrm{d}c_k} = 2 \sum_{i=1}^{n} \frac{1}{\nu_i * S_f^*} \left(\nu_i * S_f^* - \sum_j c_j A_j \right) A_k = 0 \tag{8.10}$$

$$\sum_i A_k = \sum_i \sum_j A_j c_j \frac{A_k}{\nu_i * S_f^*} \tag{8.11}$$

Where ν_i is calculated from the elastic e' track per event, S_f is evaluated from MC, R is allowed to float to match calibration results to MC energy spectra, and A is ADC response summed over all block indices j, k and then over all events i. Equation (8.10) follows from minimizing the difference in equation (8.11).

The essential steps of this analysis are as follows:

1. Obtain a tight selection on elastic events with enough data to populate each channel in HCal with roughly 1000 events. Global cuts and physics cuts are already applied in the script for GMn along with run selection. All of these parameters can be found on github. These setup files can be used as templates for new SBS configurations.

2. Loop over all events and calculate W^2 and the coincidence time (coin) for that event (now significantly more convenient with ADC and TDC signals aligned). Skip all events which fail global elastic cuts on subsystems and whose W^2 and coin values lie outside of the elastic mean $\pm 3\sigma$. Apply additional elastic cuts as necessary.

3. On each event, calculate the expected elastic recoil nucleon HCal location (X_{exp} and Y_{exp}) and expected energy ν from the e' track. Skip all events whose expected positions do not lie within one block of the periphery on HCal.

4. On each event, calculate the expected event energy normalization (from above, $\nu * S_f / R$).

5. On each event, divide each ADC value from the primary block with its previous gain factor from the database. These values are one-to-one with HCal channels and should be available for all blocks on all events. This returns the raw pC value of the data.

6. On each event, add ADC values to the A_k vector (LHS above, 288 elements) where they exist in the primary cluster.

7. On each event, add coupled ADC values to the $A_j * A_k$ matrix (RHS above, 288 x 288 elements, $M_{j,k}$) and normalize each value in the matrix by expected event energy normalization.

8. After the loop over events, check each element of the A_k vector to ensure sufficient events exist to calibrate (ideally, 100) and that the ratio of diagonal matrix elements $M_{k,k}$ to the A_k vector element is reasonable (a good lower limit is 5%). For each element which fails one of these tests:
   - Set $M_{k,k} = 1.0$.
   - Set $A_k = 1.0$.
   - Set $M_{k,j} = 0.0$ for all j.
   - Set $M_{j,k} = 0.0$ for all j.
9. Invert the matrix and calculate the coefficients: $M^{-1}_{j,k} * A_k = c_j$.
10. Provide a timestamp where necessary and update the database with these parameters (in GeV/pC) after variable name *sbs.hcal.adc.gain*.

The script used to calibrate GMn using this approach exists on github. The steps to run the script are as follows:

1. Do: *root 'ecal.C(<(string)experiment>,<(int)configuration>,<(bool)quasi-replay>,<(int)pass>)'* where *<configuration>* refers to available experimental configuration and *<quasi-replay>*$=$ $false$. One can also use the shell script for convenience. Do (spaces between inputs): *./run_ecal.sh <(string)experiment> <(int)configuration> <(bool)quasi-replay> <(int)pass>*

   > Note: Additional configurations can be produced for use with this script. The SBSconfig and SBStarget classes must be updated by configuration and the runlist and cutlist spreadsheets must be filled by run.

2. Output quality plots can be found in the output directory detailed earlier for energy calibrations. ADC gain parameters can be found from this working directory under */parameters/adcgaincoeff_<experiment>_conf<configuration>_pass<replay-pass>.txt*.

To verify that new gain coefficients are accurate, one can run this script again in sequence with $qreplay == true$. This will quasi-replay the data with new ADC gain coefficients and produce quality plots in the previously identified location.

Properly calibrated ADC gains should be verified in the following ways (at minimum).

- Sampling fraction should be roughly uniform across the dispersive (rows) and transverse (columns) directions of HCal. See figure 4.
- Aggregate sampling fraction should agree well with MC. See figure 5.
- Aggregate energy spectra should agree well with MC. See figure 6.

9 Additional Calibrations

Calibrations in this section rely primarily on accurate energy calibrations and are generally essential to evaluate the performance of HCal during experiments. Where timing resolution in HCal is of heightened import (to distinguish between neutrons and photons via time of flight (TOF), for example) or where position offsets are of heightened import (to evaluate fiducial cuts on proton-neutron cross section ratios, for example), the following can be used.

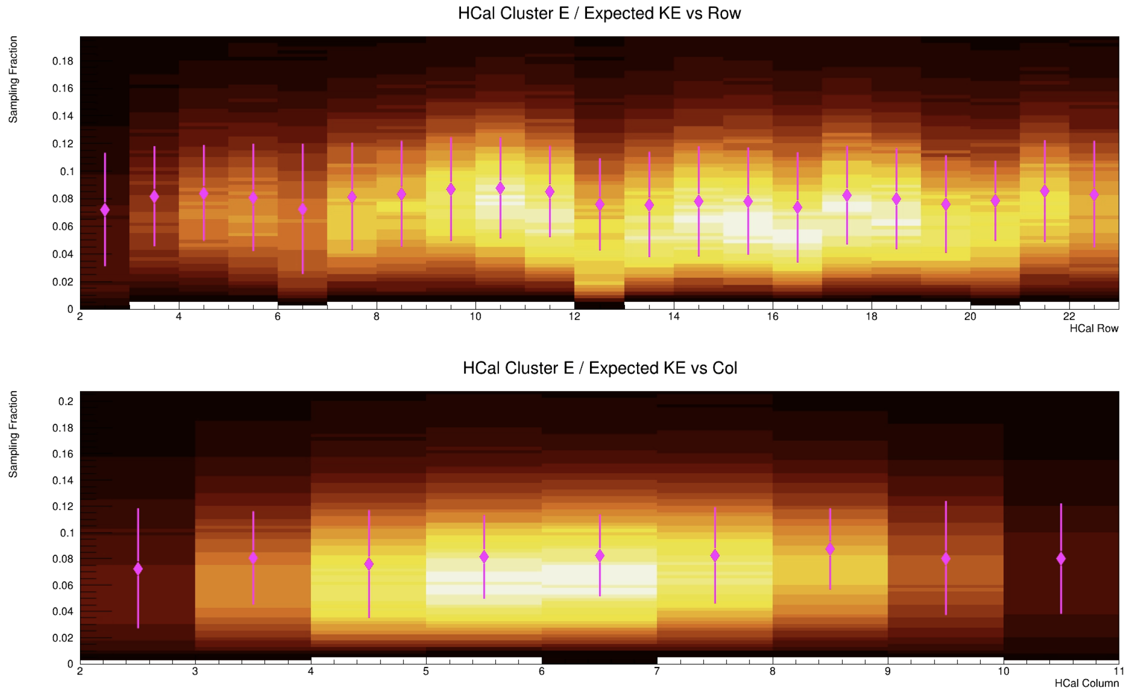


Figure 4: HCal sampling fraction vs HCal ID, uniformity check

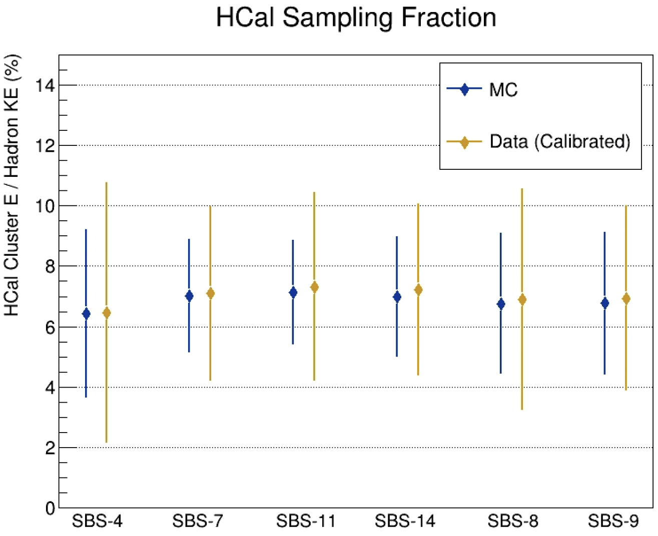


Figure 5: HCal sampling fraction vs GMn kinematic, MC/data comparison

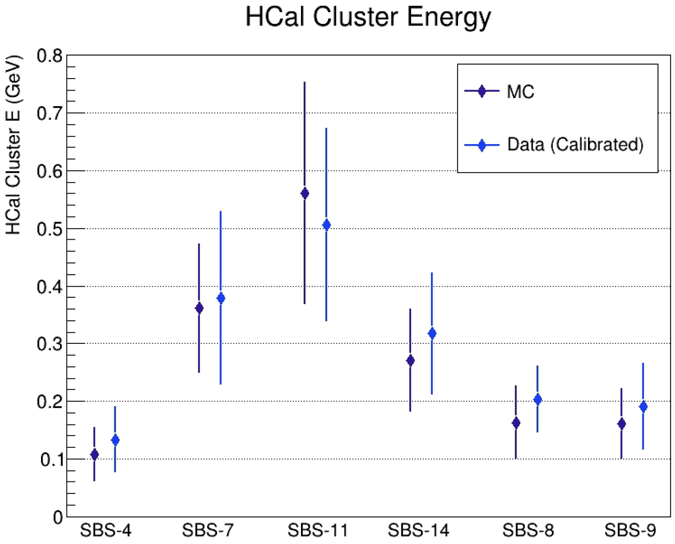


Figure 6: HCal energy spectra vs GMn kinematic, MC/data comparison

9.a Position

While several methods exist to provide general offsets to all channels in HCal, the preferred method is to pass 288 offsets (one per channel) to the variable which governs the transverse position (**sbs.hcal.ypos**) and the variable which governs the dispersive position (**sbs.hcal.xpos**).

Ideally, the engineering drawings (CAD) should be used in conjunction with the hall survey by kinematic to set these offsets. As of GMn, the vertical offset of HCal from beam height is **75 cm** (center of HCal above beam height). For the transverse direction, CAD is sufficient and the current set of offsets is (and should remain) accurate. However, due to effects from BigBite optics as impacted by fringe SBS magnetic fields, the difference between expected locations of elastic hadrons in HCal from straight line projections and neutron primary cluster centroids do not result in peak positions at the precise height of the beamline. For this reason, fits to neutron peaks per kinematic and (ideally) per magnetic field setting can also provide the offset. **The choice for vertical offset is not yet clear for all configurations** and must be chosen for each experiment/configuration combination. The steps for this calibration are as follows:

1. Obtain a tight selection on elastic events from deuterium or helium-3 with enough data to populate each channel in HCal with roughly 1000 events.
2. Fill a distribution on "delta-X," or the difference between the expected location of an elastic recoil nucleon and the energy-weighted centroid of the primary cluster in HCal with deuterium data.
3. Identify and fit the neutron peak with a Gaussian. Note that charged particles will be bent by the SBS field and enable PID. Get the peak location in m.
4. It is sufficient to subtract this value from all offsets following **sbs.hcal.xpos** in the database to complete the correction.

It may become necessary to determine the current dispersive offset from the database. If this is the case:

1. Take care to correctly determine the relevant dispersive position offset used for replay of analyzed data from the database, which may depend on timestamp. First, note the following:
   - The top row positions (lowest values) are V_t and represent the X position of the center of all blocks there.
   - The bottom row positions (highest values) are V_b and represent the X position of the center of all blocks there.
2. The X position offset (X_O) in m can be calculated as follows:

$$X_O = \frac{V_b - V_t}{2} - V_b \tag{9.1}$$

A similar procedure can be devised to extract transverse offsets. Scripts exist to easily generate these parameters with user-given corrections or total offsets. They can be found on github.

9.b Timing

As already mentioned, the timing information from TDC and ADC channels can be quickly corrected with the hodoscope cluster mean time per event to account for any timing jitter arising from the primary trigger. While timing offsets are configurable for replays with SBS-offline, additional corrections to HCal timing are not currently possible with constants passed to the database. As a result, the following sections detail methods for improving timing resolution with data post-replay. It remains a forthcoming project to add the possibility to calibrate these corrections via SBS-offline.

9.b.1 TOF

Time of flight corrections begin with MC. Simulations of elastic protons and neutrons from deuterium must be performed with relevant experimental configuration parameters such as HCal angle, HCal distance, and beam energy. For information in Geant for SBS see G4SBS documentation. From these data, two methods have been investigated to address TOF.

- **TOF vs ID**: A set of raw time corrections can be passed by block location for a fixed central momentum of elastic recoil *protons* at a given SBS configuration. Similarly, a set of raw time corrections can be passed by block location for a fixed central momentum of elastic recoil *neutrons* at a given SBS configuration. Proton and neutron TOF will vary due the presence of the SBS magnetic field. This method entails 288 (channels) x 2 (hadron type) parameters per SBS configuration, requires accurate identification of the recoil nucleon (PID), and fails to address the effect of recoil nucleon momentum on TOF.

- **TOF vs Nucleon Momentum**: A parameterized functional correction can be passed for the entire detector which depends only on the recoil nucleon momentum (Np) and is calculated per event. This method depends on the functional form of the fit to TOF vs Np for a given SBS configuration, but is typically adequate with 4 parameters. This method addresses the dependence of TOF on Np and where the transverse position of elastic nucleons is correlated with Np, it partially addresses the position dependence of TOF. This method fails to address the dispersive position dependence of TOF.

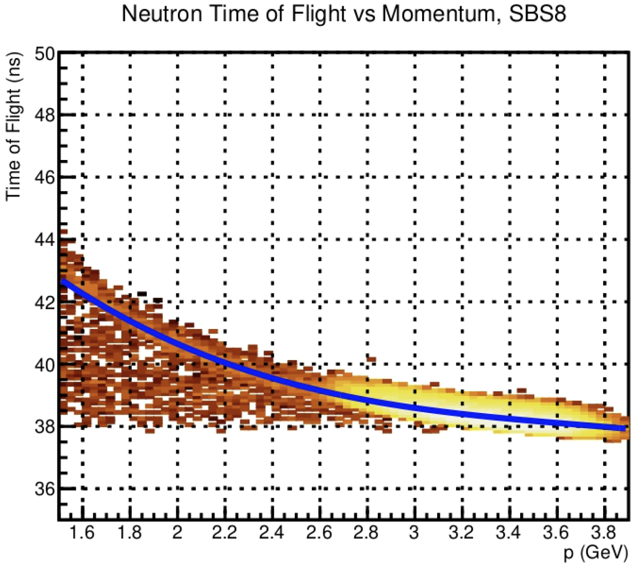


Figure 7: MC TOF vs Np, SBS4, example TOF poly4 fit

- **TOF vs Nucleon Momentum with Dispersive Correction**: This method combines the previous method with a position dependent correction by row and nucleon type. This method requires accurate PID to be fully effective and entails 4 (TOF vs Np fit) + 24 (TOF vs X position) * 2 (hadron type) parameters.

Scripts to extract the TOF from direct MC output can be found on github. **simTOF.C** is the primary script to obtain relevant TOF distributions there. For best results, TOF vs Np should first be fit by slices in Np, then a functional fourth order polynomial fit to the means of the these slices should be parameterized. Exclusion limits below and above which a fit to these mean values cannot be trusted should be set and observed for all practical corrections. See figure 7 for an example.

With these fit parameters, **tcal.C** can be used to apply them. After processing iteration 2, one can run iteration 3 apply TOF corrections and generate new alignment parameters, then run iteration 4 to apply new corrections and new alignments. This will generate new TDC and ADC time distributions by channel. Note that **tcal.C** is currently configured for GMn, but can be reconfigured for new data with appropriate config files (see section 5.a.1) and appropriate TOF fit parameters obtained in the manner previously mentioned. Output quality check plots can be found in the usual place; from the **tcal.C** working directory, */reporting/timing/allCorrReplay_sbs*<kinematic>*.root*. TOF fit parameters for the most common GMn configurations are included in the preamble to **tcal.C**. See figure 9 for examples of these fits applied.

9.b.2 Timewalk

On determination of TDC and ADC time signals, there exists a dependence of the signal on amplitude of the raw analog waveform. This arises from the method used to extract TDC and ADC

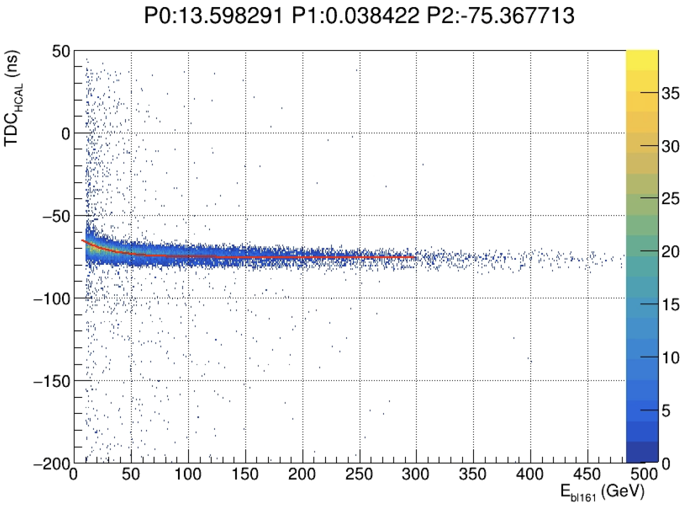


Figure 8: HCal TDC vs E, block 161, example TW expo fit

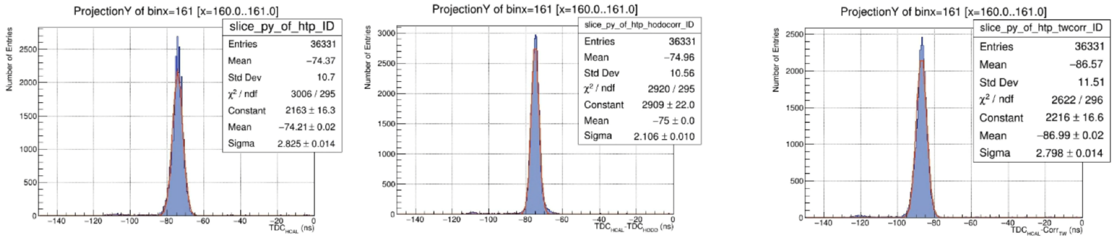
time, both dependent on analog threshold crossing and the finite rise time of analog waveforms. This is timewalk.

To account for this, it is enough to fit time vs E distributions and extract fit parameters by channel for all configurations, then apply them by event to correct for timewalk. The process is as follows:

1. Obtain a wide selection on elastic events with enough data to populate each channel in HCal with roughly 1000 events.
2. Populate 2D histograms from the primary cluster of TDC vs primary block energy and ADC time vs primary block energy.
3. Fit these distributions. A dying exponential fit works well, but one could use a polynomial fit also.
4. Obtain fit parameters by block. For a dying exponential, this entails 288 (channels) x 2 (fit) parameters. Overall offsets to these data can be safely ignored for the correction.
5. Loop over all data again and calculate the correction for the primary block in the primary cluster. Apply these corrections to TDC and ADC time values.

See figure 8 for an example where the fit parameters are reported above the histogram. Note that the linear interpolate method used to calculate the ADC time which already exists in SBS-offline largely mitigates the emergence of timewalk. This correction is primarily useful for correcting TDC times in HCal.

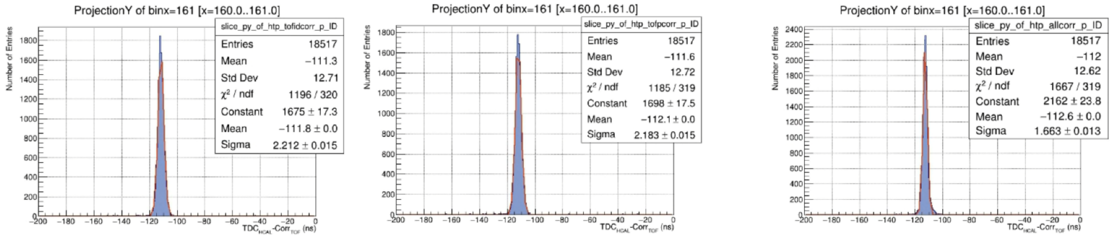
tdc_tw.C can be used to apply dying exponential fit parameters to correct for timewalk. The process to run this script is identical to that in the previous section. Output quality plots are located in the same file as described there. See figure 9 for an example of this fit applied.



tdc primary block

tdc primary block, hodo corrected

tdc primary block, TW corrected



tdc primary block, TOF vs ID corrected

tdc primary block, TOF vs p corrected

tdc primary block, all corrections

Figure 9: HCal TDC, block 161, many time corrections. No corrections applied, top left; all corrections applied, bottom right.

10 Final Remarks

Several adjustments/corrections are possible which may improve timing, energy, and position resolution and accuracy beyond what have been mentioned thus far in this document. Those that are left out of this document are done so due to scope, but some include:

- Best cluster selection using BigBite (e-arm) variables
- ADC time extraction via corrected Landau/Gaussian convolution fits to full waveforms
- TDC signal recovery via primary cluster block searches or non-primary cluster searches
- Update sampling fractions which account for HCal geometric effects and noise per channel to improve calibrated energy spectra

Additionally, cosmic calibrations and alpha parameter extraction are discussed in this document, but the scripts are not written for general use and must be updated for any future initial commissioning. Further, details of HCal PMT plateau region evaluation are left out entirely.

Finally, HCal is a new detector for SBS experiments and its full capabilities have yet to be fully understood and appreciated. Some of the methods detailed here will likely prove to be the best possible method for extracting optimal performance from the apparatus. Many will not. New methods will need to be devised, researched, and applied in conjunction with continued improvements to other subsystems to best take advantage of what HCal is capable of. This document should be considered a good start on an ongoing effort, not only to minimally calibrate the apparatus, but to optimize it for the remainder of its tenure in Hall A.