Calo calibrations data-taking and run tasks

Introduction

Documented here are an important set of tasks related to the data-taking, commissioning, and important time-sensitive aspects of the reconstruction. Timely progress and quick response on these tasks will be necessary and should be taken into account when signing up. There are a subset of people who have been working very hard doing some or all of these things; the goal of this signup sheet is to have more clearly defined roles, spread out the workload, and allow for less expert people to develop skills in a specific area so they can make a concrete contribution to the calorimeter effort. If you are a relatively experienced sphenix analyzer but have mainly worked on MC-based or physics-type analysis, this may be a good opportunity to get involved with ensuring the basic calorimeter data is of the highest quality, which will make your life easier in the future.

Long-term roles

These roles will require routine consistent work throughout the run. The roles can switch from person-to-person.

Online Monitoring
EMCal: Vincent Andrieux
HCal: Shuhang Li

HCAL cosmics: Hangpu Jaing, Blair Seidlitz
Coordinate cosmics data-taking, reconstruction, shift instructions.

HCAL Tower slope: Justin Bryan, Blair Seidlitz
Coordinate automated running of calibration code and application of results to calibration

LEDs: (EMCal) (HCal)
Work closely with the DAQ group to optimize running of LEDs. Update shifter instructions. DB recording of the means and widths.

Hot/cold towers: Anthony Hodges (EMCal) (HCal)
Run hot/cold tower finding code and early in the run contact relevant experts to attempt to fix the hot/cold towers.
https://github.com/sPHENIX-Collaboration/coresoftware/tree/master/calibrations/calorimeter/calo_emc_noisy_tower

EMCal calibration: Blair Seidlitz, Justin Frautz, Sijan Regmi

Offline QA and good run list: Greg Mattson (EMCal) (HCal)
Coordinate the running and publishing of run-by-run offline QA as well as making good run lists.

Details: histograms will be made by the CaloValid module. Then the histograms will be plotted and published by the QAhtml package, for which I need to make a new CEMC subdirectory and create CemcDraw.cc and .h following the example .

Projects

These are projects which have a foreseeable or atlast once a stable state is reached less effort will be required.

Pedestal and noise checks: Pedro Marin

Determine pedestal fluctuations and noise at ADC/interface board level vs eta,phi level vs xmit or crate or rack level. Compare the pedestal from samples in the waveform to the pedestal extracted by the template fit. Compare noise levels for beam vs no beam.
This uses basic sphenix calo code/skills.

Waveform template:

Derive a new template with 2024 data and compare it to 2023. Compare different ADC ranges for input waveforms and how that affects the chi2.
This analysis creates templates from combine prdfs:
https://github.com/sPHENIX-Collaboration/analysis/tree/master/SimpWaveformFit

Offline Bit-flip recovery:

If the bit flips are recovered, the bit shifts are prevented, and the timing cuts removed, all towers could be good in all events leading to only a dead/hot mask greatly simplifying the downstream reconstruction. One would think a simple difference between a sample and its neighbors could detect this. The first question would be how many false positives would there be? The second question would be, if we can easily remove the sample from the fit or take the average of the neighbors.
This requires basic calo coding/skills and then will require the editing of
https://github.com/sPHENIX-Collaboration/coresoftware/blob/master/offline/packages/CaloReco/CaloWaveformFitting.cc for implementation of the recovery algorithm

Event alignment, combining and sustained readout:

How do we know every packet is full in every event? How can we test alignment with the new DST structure?
This will require a more diverse set of software and some knowledge of the production framework but basic checks can be achieved with standard software.

Trigger:
Tower energy distributions for various triggers, compared to pythia
Occupancy (non-zero suppressed rate) for various triggers, compared to pythia
This requires basic sphenix calo software

Zero suppression:
Optimize and monitor zero-suppression thresholds (fixed ADC or fixed energy?).
This requires basic sphenix calo software