

SiW-ECAL: Better event building

Lessons learned from TB2022-03

Jonas Kunath (LLR). 22.04.2022.



Overview

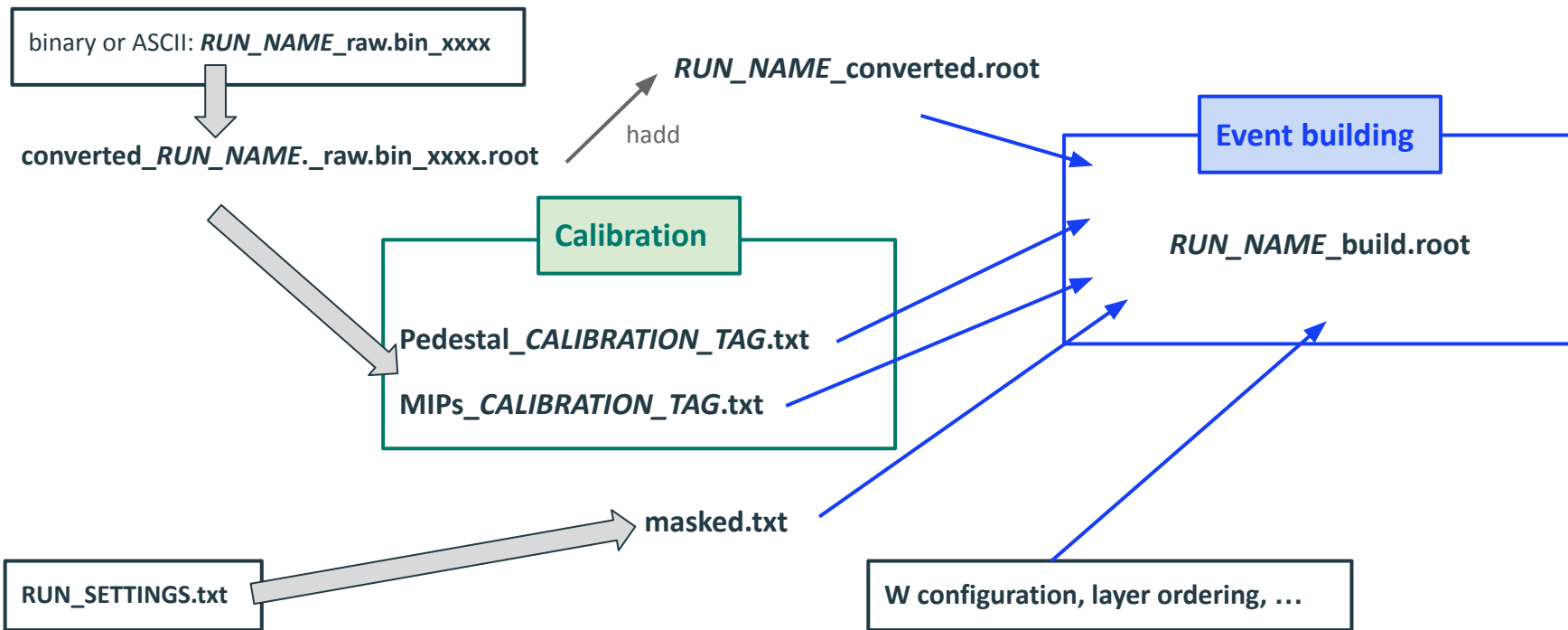
- Introduction: Event building
- Empty events
- Chip hits split



Event building



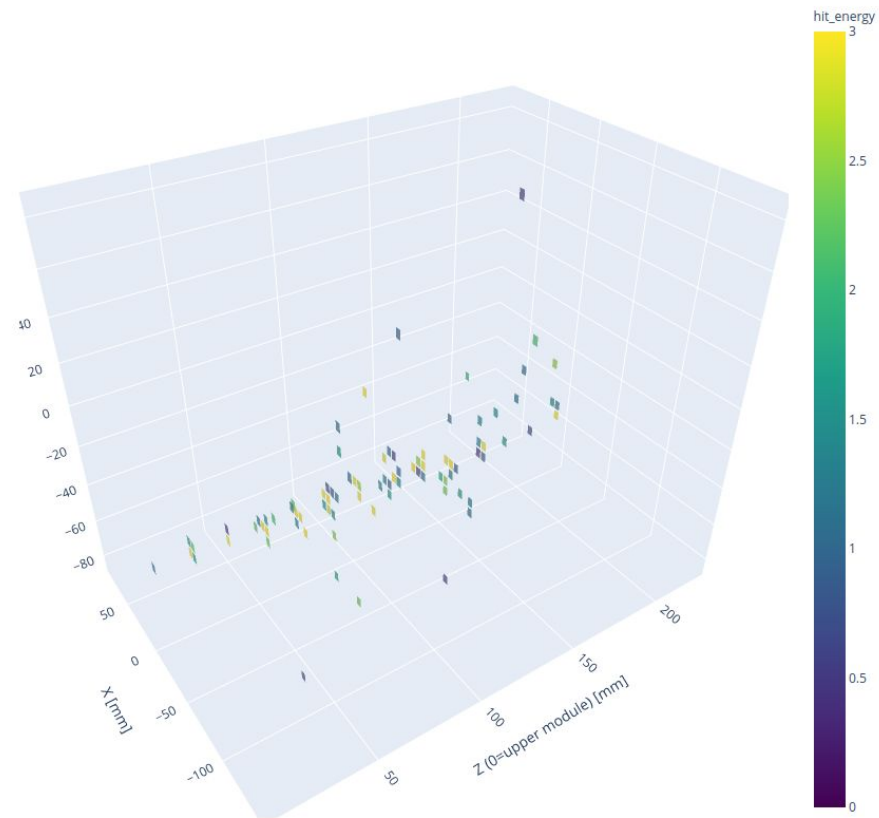
Event building pipeline



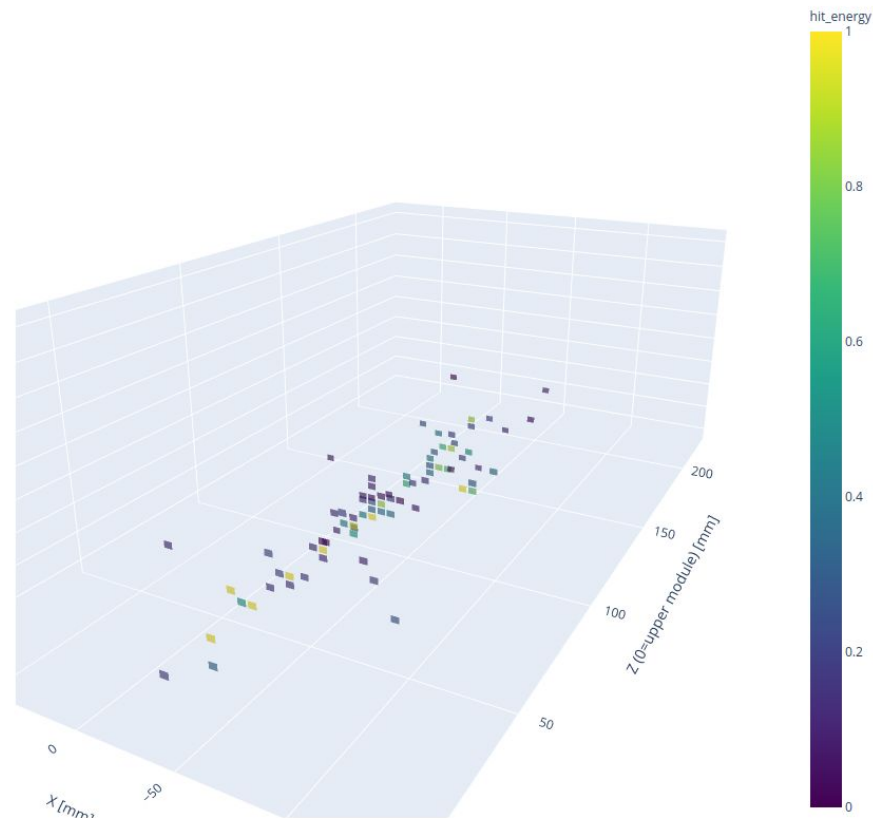
Comparing Simulation with data: build.root files

- simulation.lcio → sim_build.root written by Fabricio
- E.g. raw energy spread plots (see later)
- New monitoring scripts to create the build.root file *while the run is ongoing*
- Next slide: Event displays real data (left) & simulation @3GeV

Event display Run_name_1 #Hits=86 #Coincidences=15



Event display Run_name_2 #Hits=78 #Coincidences=1



SiWECAL-TB-monitoring for just-in-time event building

- steering file per SiW-ECAL setup
- Constant feedback

```
[INFO 09:28:32] Logging to file /afs/cern.ch/user/j/jokunath/public/...
[DEBUG 09:32:39] Channel masks written to /afs/cern.ch/user/j/jokunath/public/...
[INFO 09:32:39] The run has finished. Monitoring will try to catch u...
[DEBUG 09:33:42] New converted file converted_3GeV_MIPscan_eudaq_run_...
[DEBUG 09:33:59] New event file build_3GeV_MIPscan_eudaq_run_050471...
...
[DEBUG 10:03:31] A new monitoring snapshot is ready: full_run.root at /afs/c...
[INFO 10:03:31] The run has finished. The monitoring has treated all files.
```

```
monitoring.cfg
1  [monitoring]
2  max_workers = 10
3  output_parent = data
4  skip_dirty_dat = False
5  # Only used if the raw data is in raw.bin_XXXX format. -1 for no split.
6  binary_split_M = 50
7  # Needs some extra python packages, and adds some extra time. For batch
8  # finished runs, you might want to set this to 'quality_info' = False'.
9  quality_info = True
10
11 [snapshot]
12 after = 1, 10
13 every = 50
14 # Setting this to True can save some disk space for long runs.
15 delete_previous = False
16
17 # Any field in 'default_eventbuilding.cfg' can be overwritten here.
18 # That is also where you can find explanations of their meaning.
19 # (local) ./continuous_event_building/SiWECAL-TB-analysis/eventbuilding/
20 # (online) https://github.com/SiWECAL-TestBeam/SiWECAL-TB-analysis/tree/
21 [eventbuilding]
22 min_slabs_hit = 6
23 # Tungsten thickness in mm. A single number is interpreted as per layer
24 w_config = 2.8,2.8,2.8,2.8,2.8,2.8,2.8,4.2,4.2,4.2,4.2,4.2,4.2,4.2
25 asu_versions = 13,13,COB,COB,11,11,11,12,12,12,12,11,11,10,11
26 max_entries = -1
27 no_lg = False
28 zero_suppress = True
29 merge_within_chip = True
30 pedestals_file = ..... example/dummy_calibration/Pedestal_dummy_highgain.t
31 mip_calibration_file = ..... example/dummy_calibration/MIP_dummy_highgain.t
32 pedestals_lg_file = ..... example/dummy_calibration/Pedestal_dummy_lowgain.t
33 mip_calibration_lg_file = ..... example/dummy_calibration/MIP_dummy_lowgain.t
34 mapping_file = ..... continuous_event_building/SiWECAL-TB-analysis/
35 mapping_file_cob = ..... continuous_event_building/SiWECAL-TB-analysis/
```

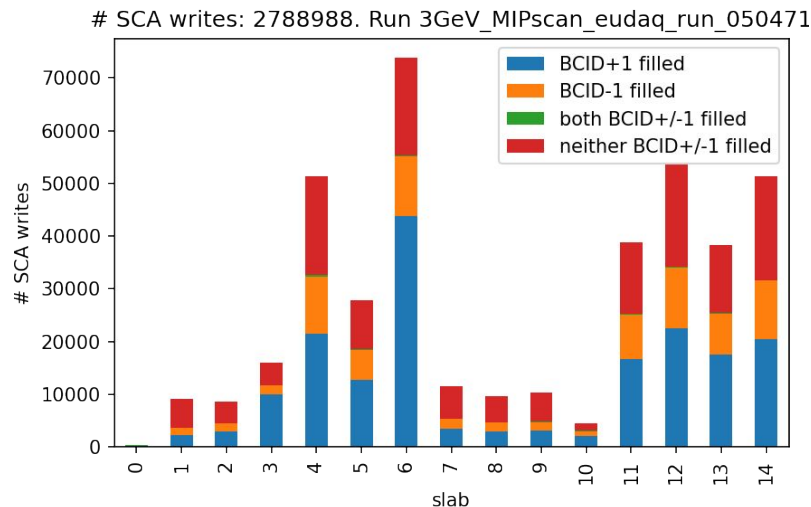


Empty SCA frames



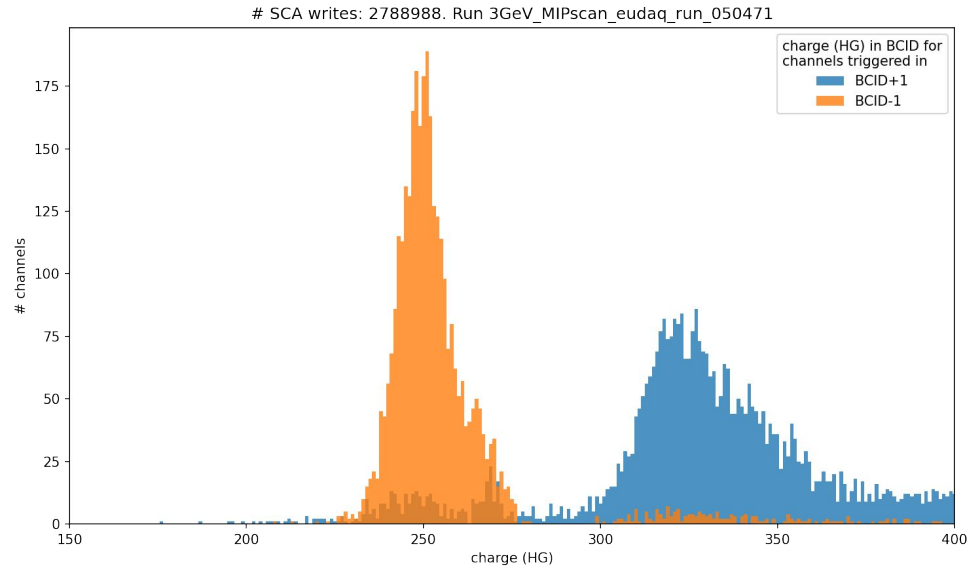
SCAs without hits (and their neighbours)

- BCID empty, BCID+1 filled
 - (analog) delay cells
- BCID empty BCID-1 filled
 - digital connection issue
 - well & long known, Stéphane can explain it better ([2016 CALICE meeting](#))
- both BCID+/-1 filled
 - most of the times probably just a retriggering train
 - sometimes not
- neither BCID+/-1 filled
 - ???
 - I found that *other chips* tend to have SCA readouts on this BCID
- Effect much smaller for Skiroc2a (vs Skiroc2)



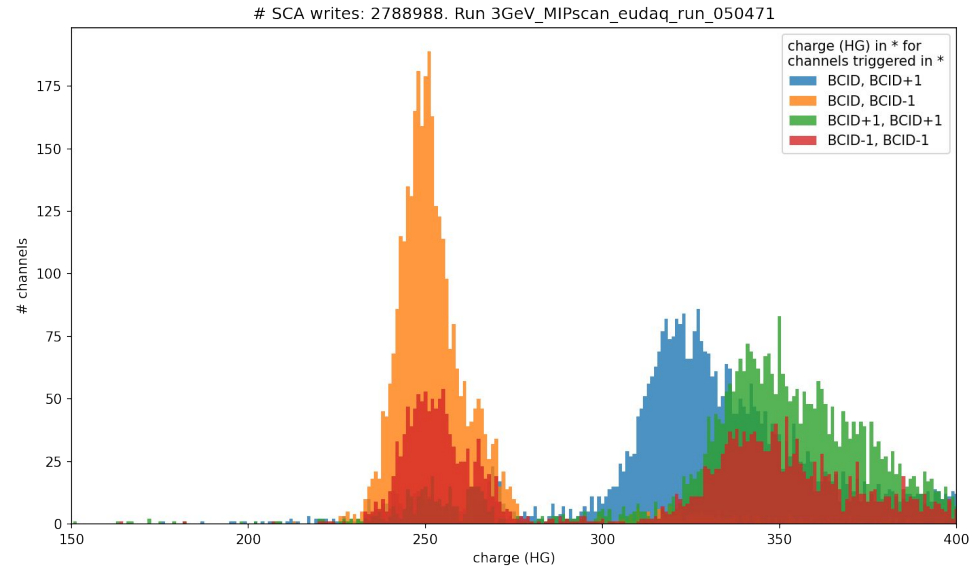
Charges in the empty events

- The following plots on layer 7 only.
- Charges from triggered channels
before/after an empty event



Charges in the empty events

- The following plots on layer 7 only.
- Charges from triggered channels **before/after** an empty event
- The charge histograms from **empty SCA** frames look less good
- Especially **triggered SCA with hits** better than **empty triggered SCA before it**



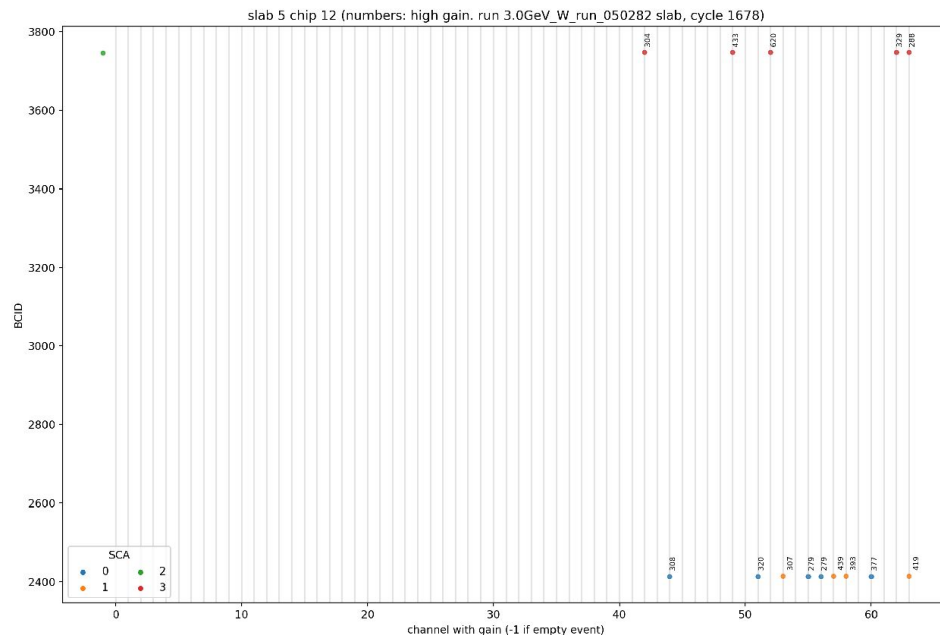


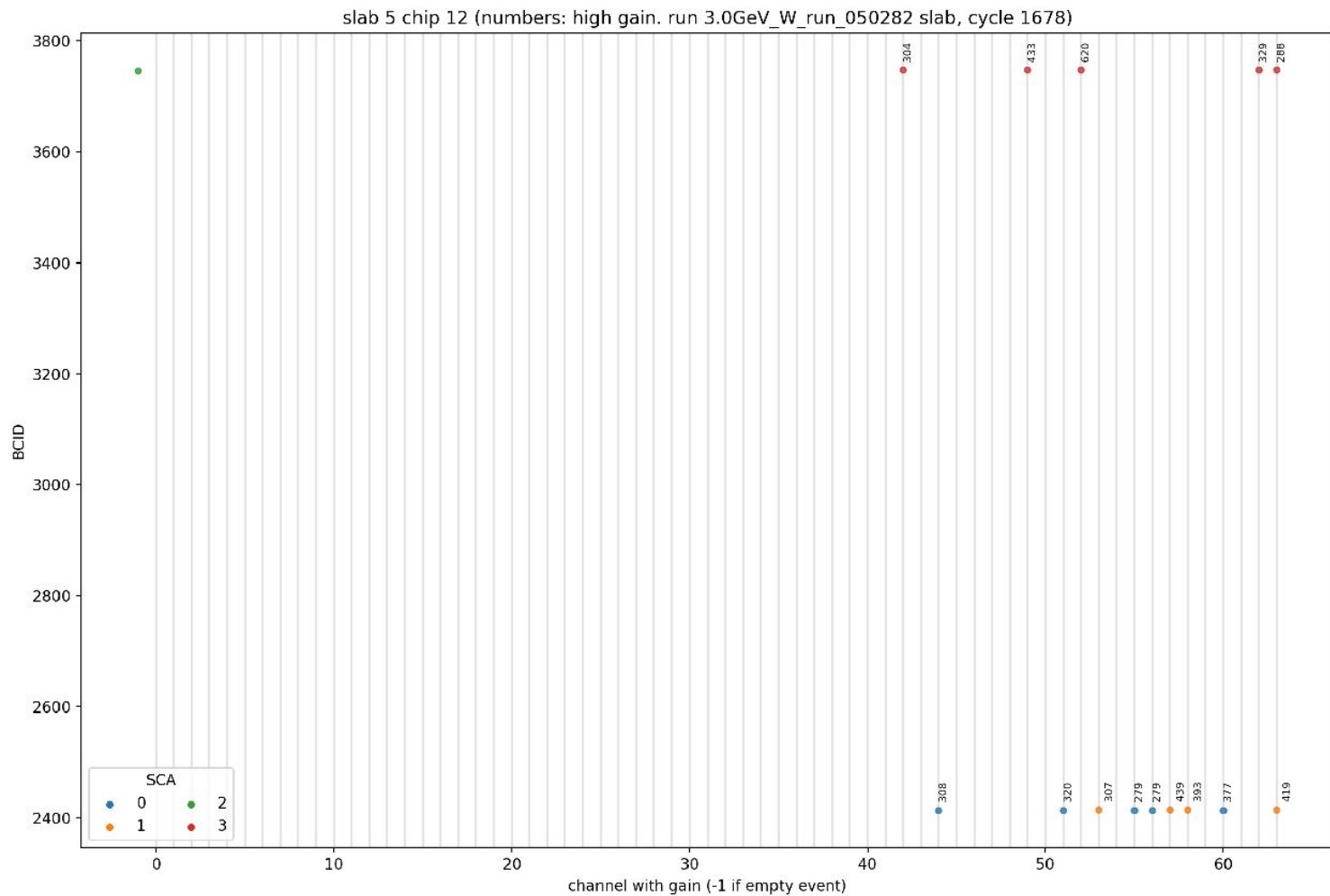
Chip hits split



Chip hits split over multiple BCIDs/SCAs

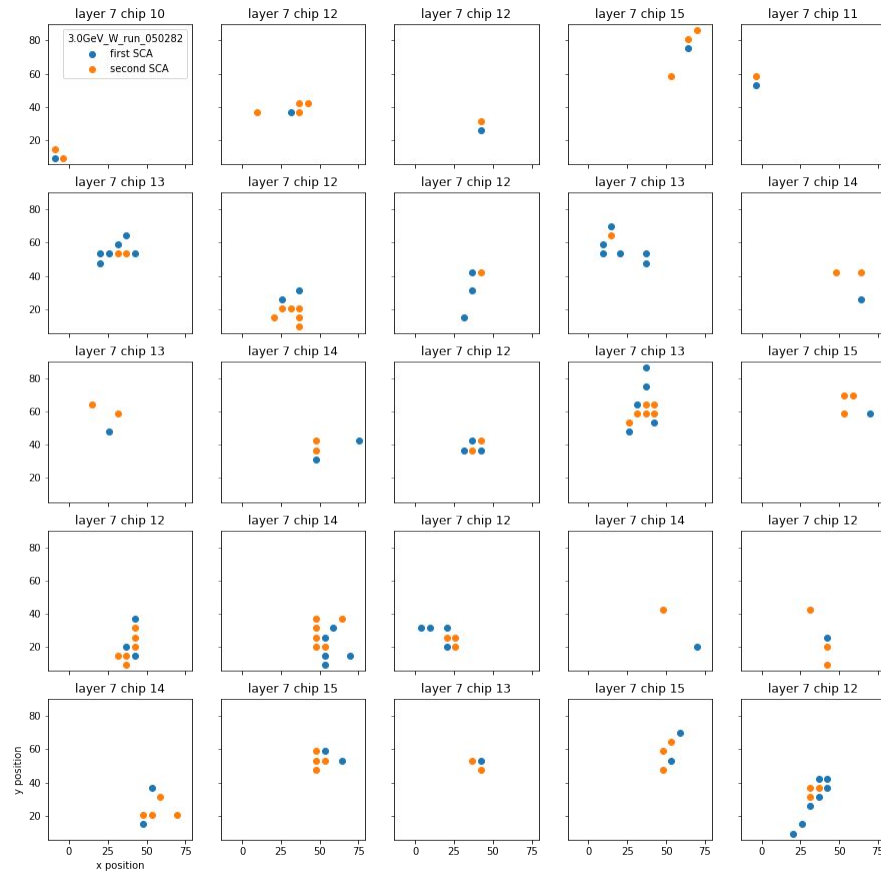
- Triggers on *different* channels in consecutive BCIDs
- If more than 2 SCAs in a row, probably retiggering effects
- Due to Delay box?





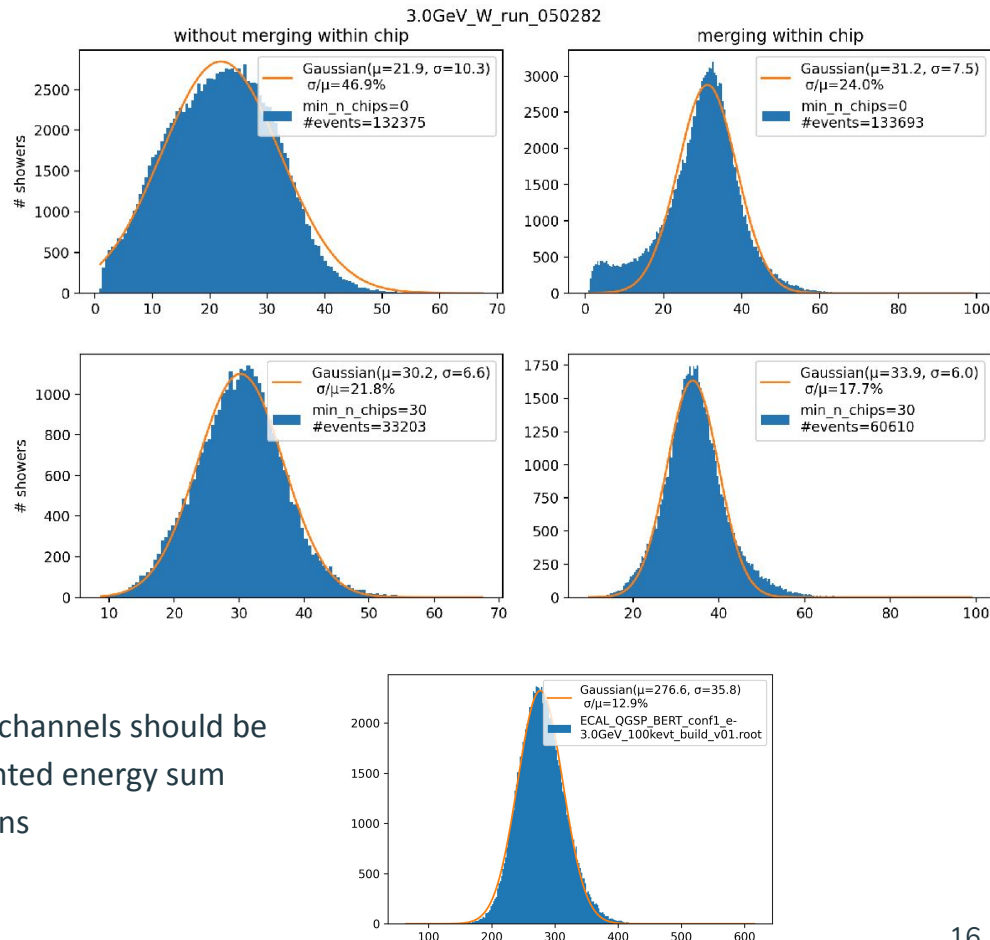
Chip split in x-y

- No obvious pattern for first/second SCA of the splitted event in x vs y



Energy spread

- Left: before within chip merging
- Clearly, the merging is necessary
- Some chips not great: require ***min_n_chips*** for improvements
- Lowest: Comparison with Simulation @Fabricio (here: before any digitisation, masking, ...)
- Caveats
 - At these low energies, counting hit channels should be better than sampling-fraction weighted energy sum
 - Preliminary MIP & Pedestal calibrations
 - No correction for masked cells here



Summary

1. **Empty SCAs:** (most of the time, at least for event building): just take the BCID-neighbouring, non-empty SCA
2. **Hits splitting** on same chip: if exactly 2 consecutive SCAs: take the union of triggered channels
3. New **JIT-build-monitoring** tools
4. **SIM-vs-data** comparison tools in place

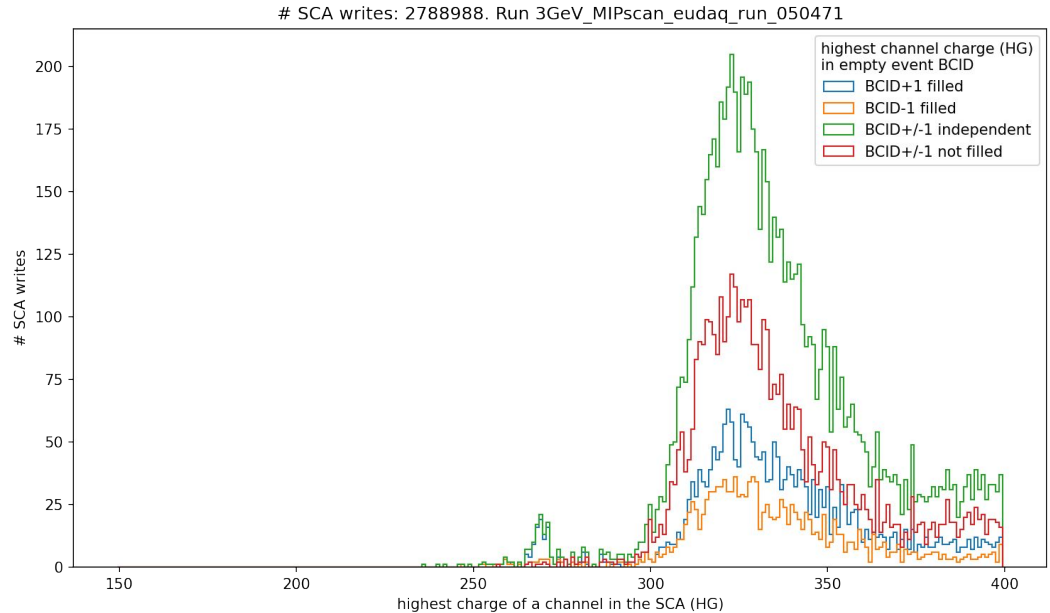


Backup

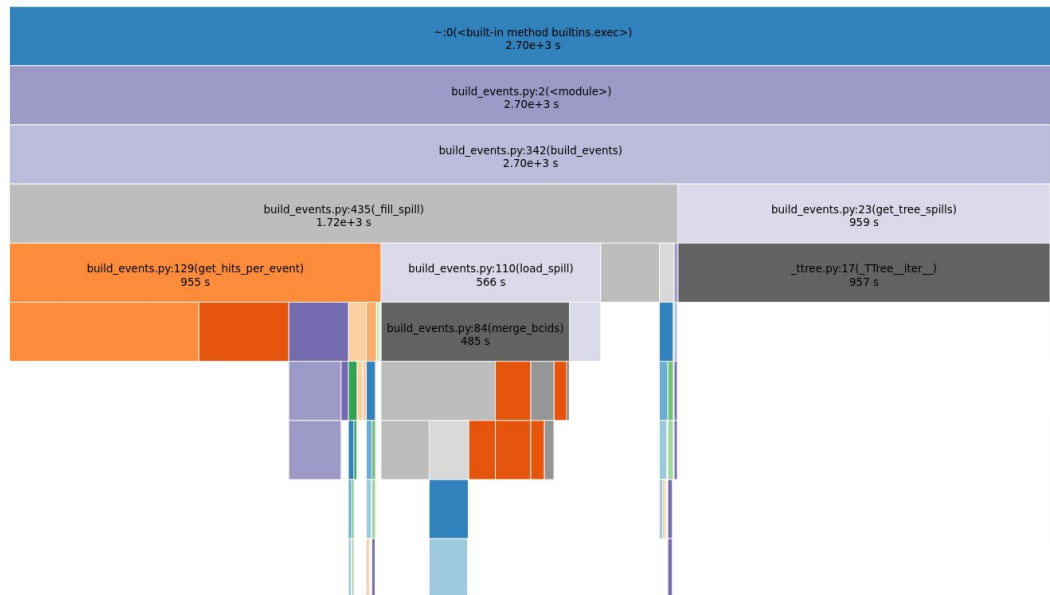


Highest charge per readout

- Here, plotted only highest charge per SCA
- All histograms look rather similar



Is this fast enough?



Profiling shows:

- No obvious bottlenecks left
- $> \frac{1}{3}$ of time spend in `get_tree_spills` (loading data to memory?)
 - I do not think this can be sped up
 - Even pure ROOT/C++ should not be faster here
- Upfront cost usually < 8 s
 - Loading imports
 - Loading & improving calibration
 - Writing calibration histograms
- `merge_bcids` could potentially be faster
 - But why bother?

Why buildfiles

- event
- spill
- bcid
- prev_bcid
- next_bcid
- nhit_slab
- nhit_chip
- nhit_chan
- nhit_len
- sum_hg
- sum_energy

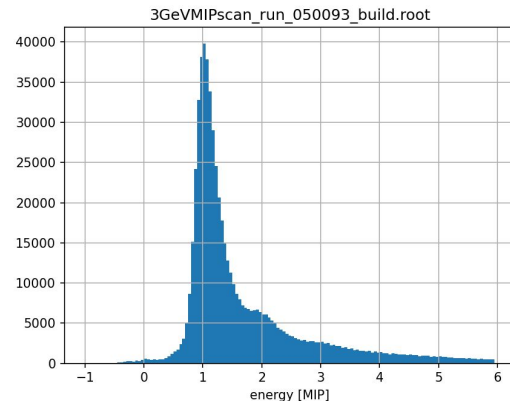
Event level information

- Useful for selection
- e.g. “nhit_slab > 10”

- hit_slab
- hit_chip
- hit_chan
- hit_sca
- hit_x
- hit_y
- hit_z
- hit_hg
- hit_lg
- hit_energy
- hit_n_scas_filled
- hit_isHit
- hit_isMasked
- hit_isCommissioned

hit arrays

- Know which hits belong together
- hit_energy: calibration applied. ==1 for average MIP deposit



```
[1]: import awkward as ak
import matplotlib.pyplot as plt
%matplotlib widget
import numpy as np
import uproot

file = "3GeV MIP scan run 050093 build.root"
ar = uproot.open(file)["ecal"].arrays(filter_name="hit_*")
n_slabs = uproot.open(file)["ecal"]["nhit_slab"].array()

[2]: hits = ar[n_slabs > 10]
hits = hits[hits.hit_isHit == 1]

energy = ak.flatten(hits.hit_energy).to_numpy()
fig, ax = plt.subplots()
ax.hist(energy, bins=np.arange(-1, 6, 0.05))
ax.set_xlabel("energy [MIP]")
ax.set_title(file)
ax.grid()
```

