2019 Data Reconstruction Readiness

Norman Graf (SLAC)

HPS Collaboration Meeting

November 19, 2020

Issues

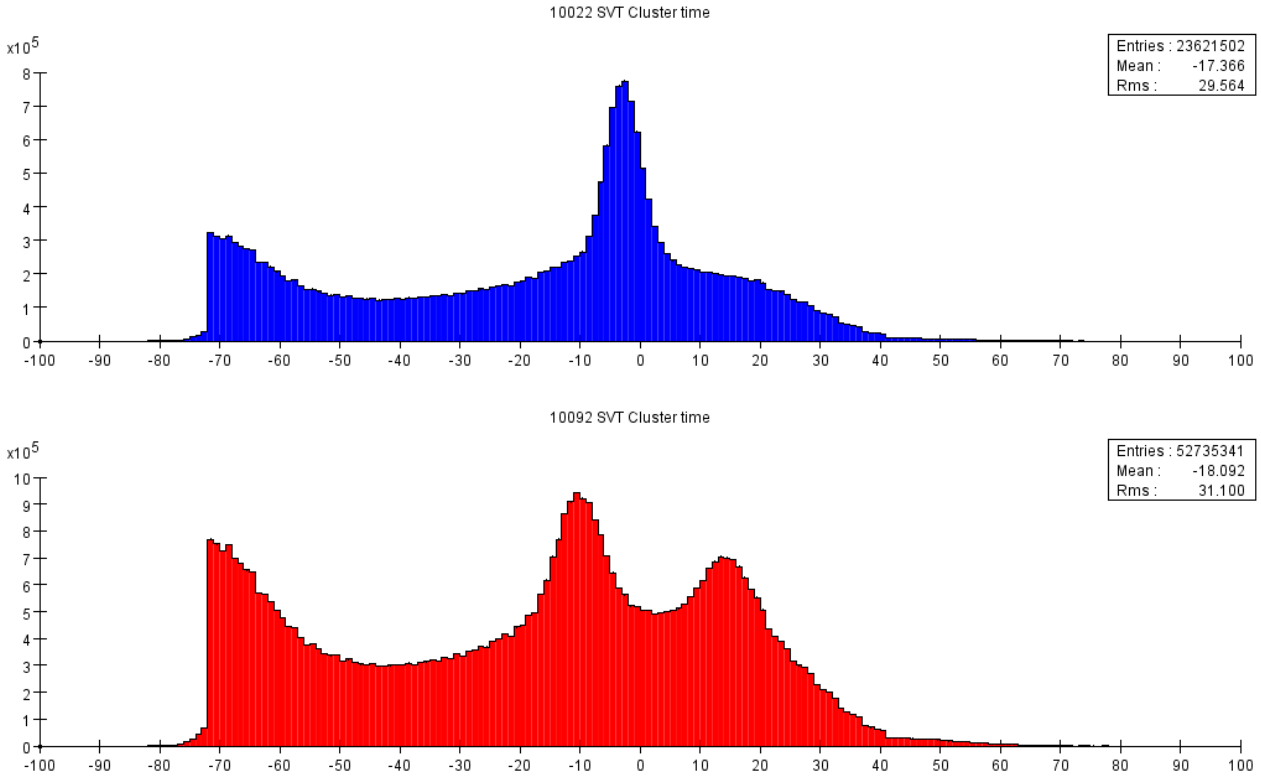
- We have promised our funding agencies that we will conduct a “timely” analysis of the 2019 data.
  - It’s been over a year since the run ended
- We have another data taking run coming up
  - Run is scheduled in less than a year
- What is needed and when is it needed to accomplish our goals?
  - Simulation (see Tongtong’s presentation)
  - Reconstruction
  - Analysis (see Cameron’s presentation)

Data Reconstruction Software Issues

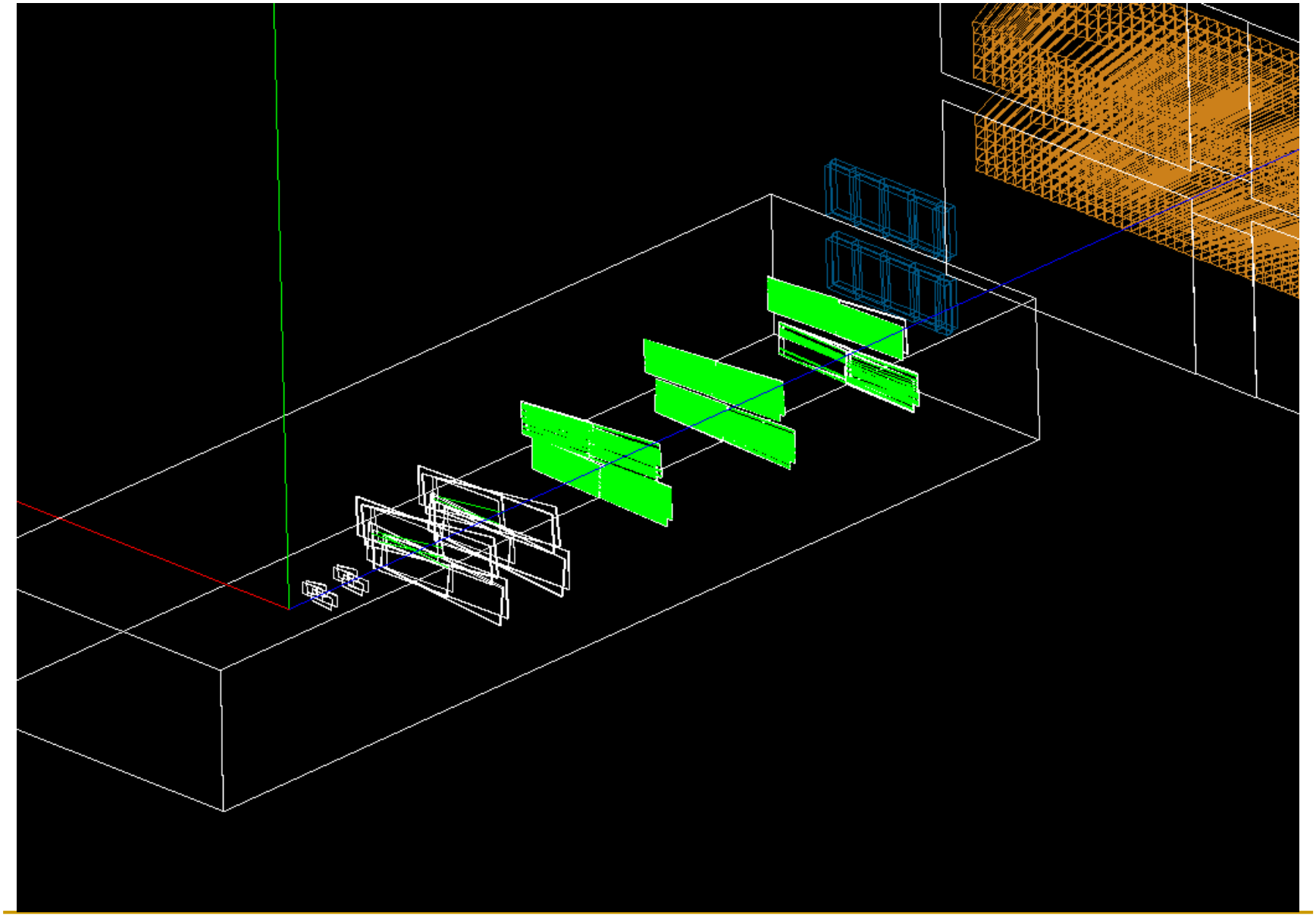
- ECal has finished up gains, sampling fractions, cluster position corrections and timing work ongoing.
- Hodoscope software OK.
- SVT trigger phase needs to be fixed for certain runs.
- Need a 2019 Event Flag Filter to remove obviously bad events
  - flag wrong SVT position, wrong SVT voltage, etc.
  - identify recoverable “monster” SVT events
  - identify, skip and drop truly “monster” SVT events
- SVT APV25 waveform fitting
  - Is the current fitting sufficient for our track timing?
  - Replacing simplex with migrad improves fitting, gives uncertainties, takes much more time.
    - Need to study this ASAP, as we plan to drop raw data from output.
  - Can it be improved and/or sped up?
- SVT actively working on alignment/calibration
  - See PF’s talk yesterday.
  - Will we need more than one alignment?
- Tracking group actively improving CPU performance
  - See PF’s talk today.
  - Need more work to optimize track-finding strategies
- Need characterization and performance evaluation of tracking software
  - Track-finding efficiencies
  - Momentum scale & resolution
- Output lcio files are bloated with extraneous data.
  - Remove extraneous Drivers
  - Need to prune our data tree and remove unnecessary collections from lcio output
- Memory footprint needs to be below 1GB to be efficient at JLab.

SVT Trigger Phase

- For certain runs, SVT locked on to the wrong trigger phase.
- Runs have been identified, just need to fix this.



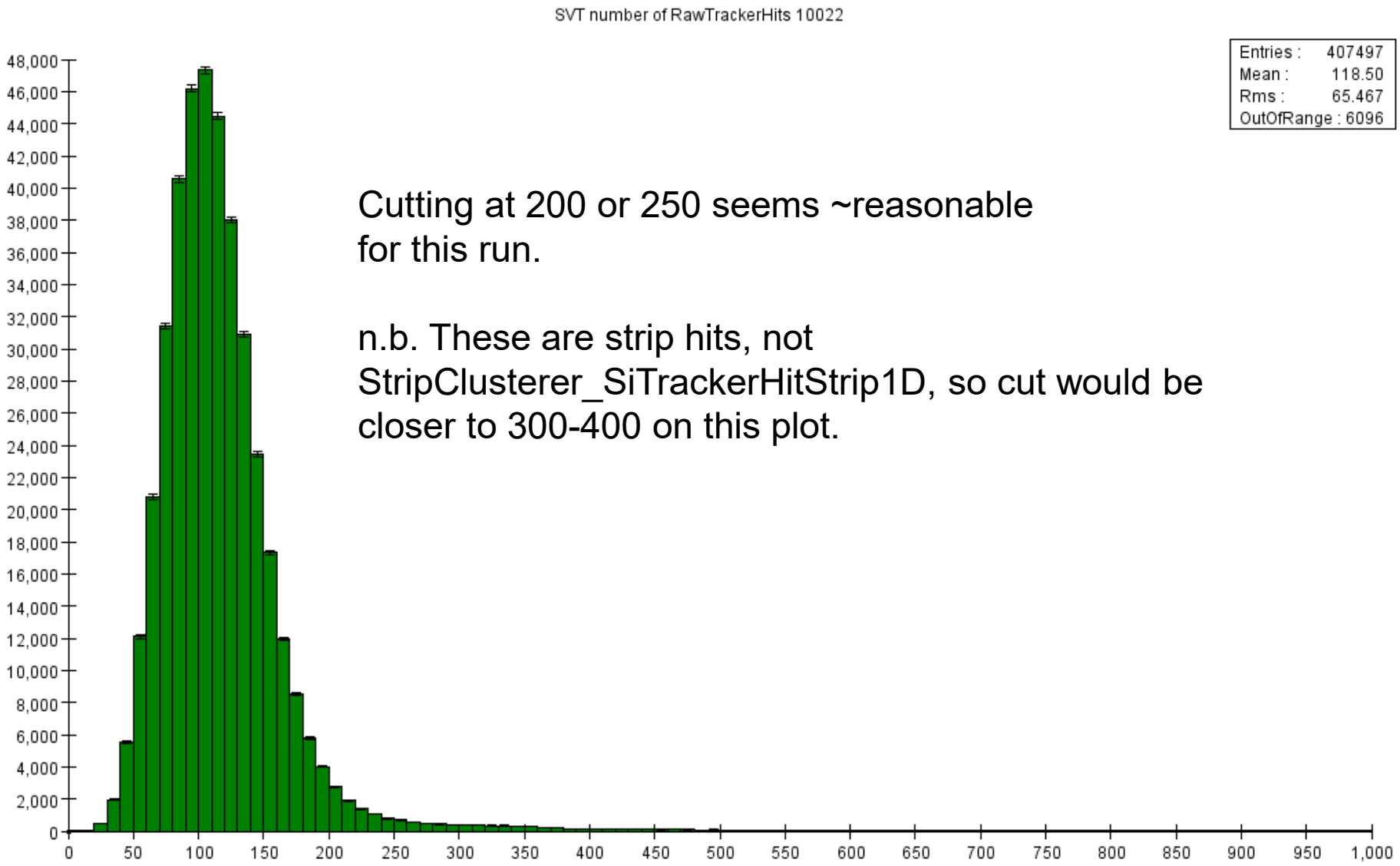
SVT “Monster” Event



SVT "Monster" Events

- A number of *ad hoc* solutions have been implemented to deal with these events.
  - Matt cuts at less than 250 SVT clusters for SeedTracker
  - Robert cuts at fewer than 200 SVT clusters in Kalman Filter
  - PF cuts at less than 200 SVT clusters in DataTrackerHitDriver
- All of these were based on global number of hits
  - May work for some situations, but clearly fails for processing all our runs.
- Need to identify specific types of "monster" events and handle them individually.
- Need to do this early on (before fitting of waveforms)
- Truly "monster" events should be abandoned and not even processed further nor written out.

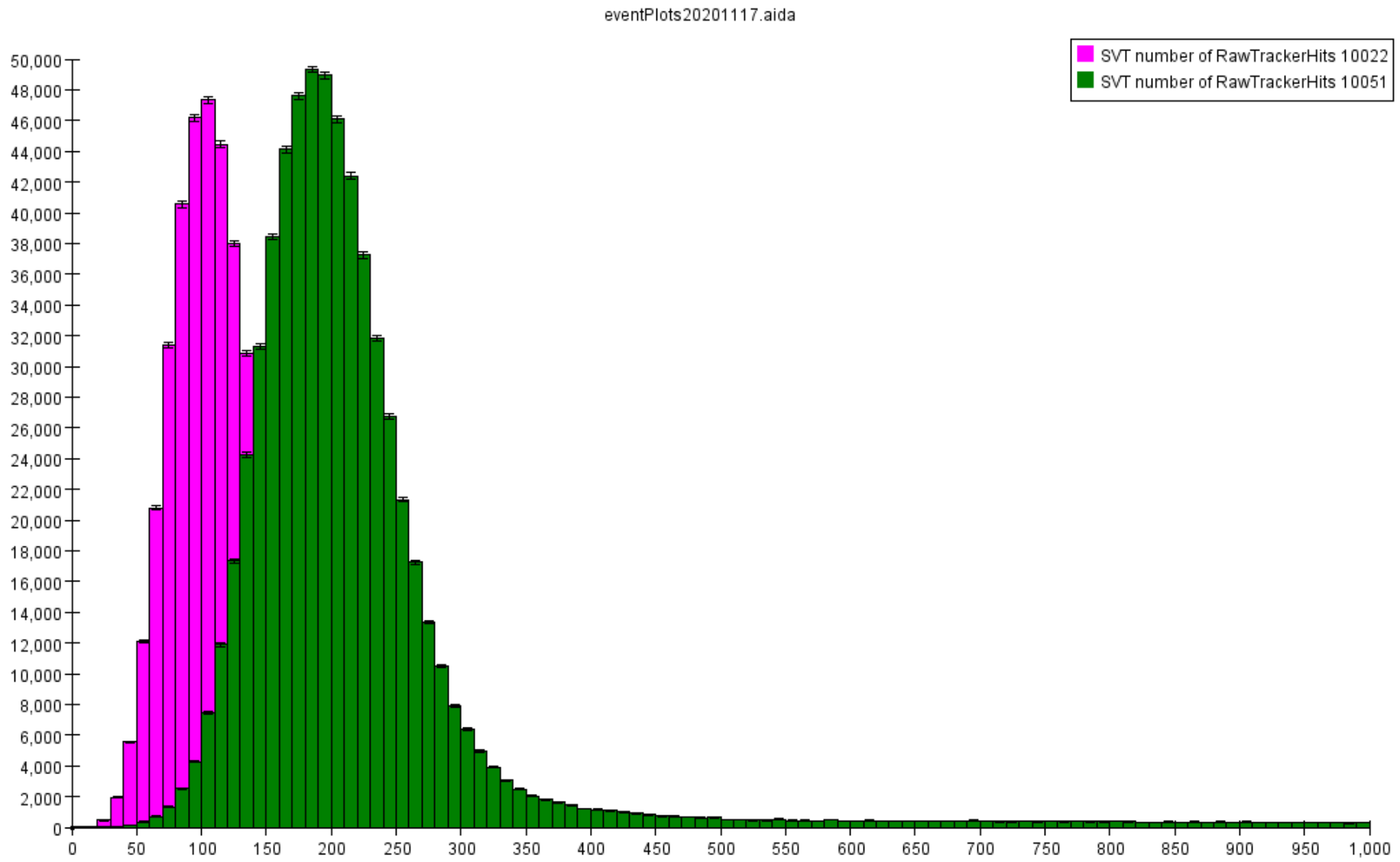
SVT Number of Raw Tracker Hits



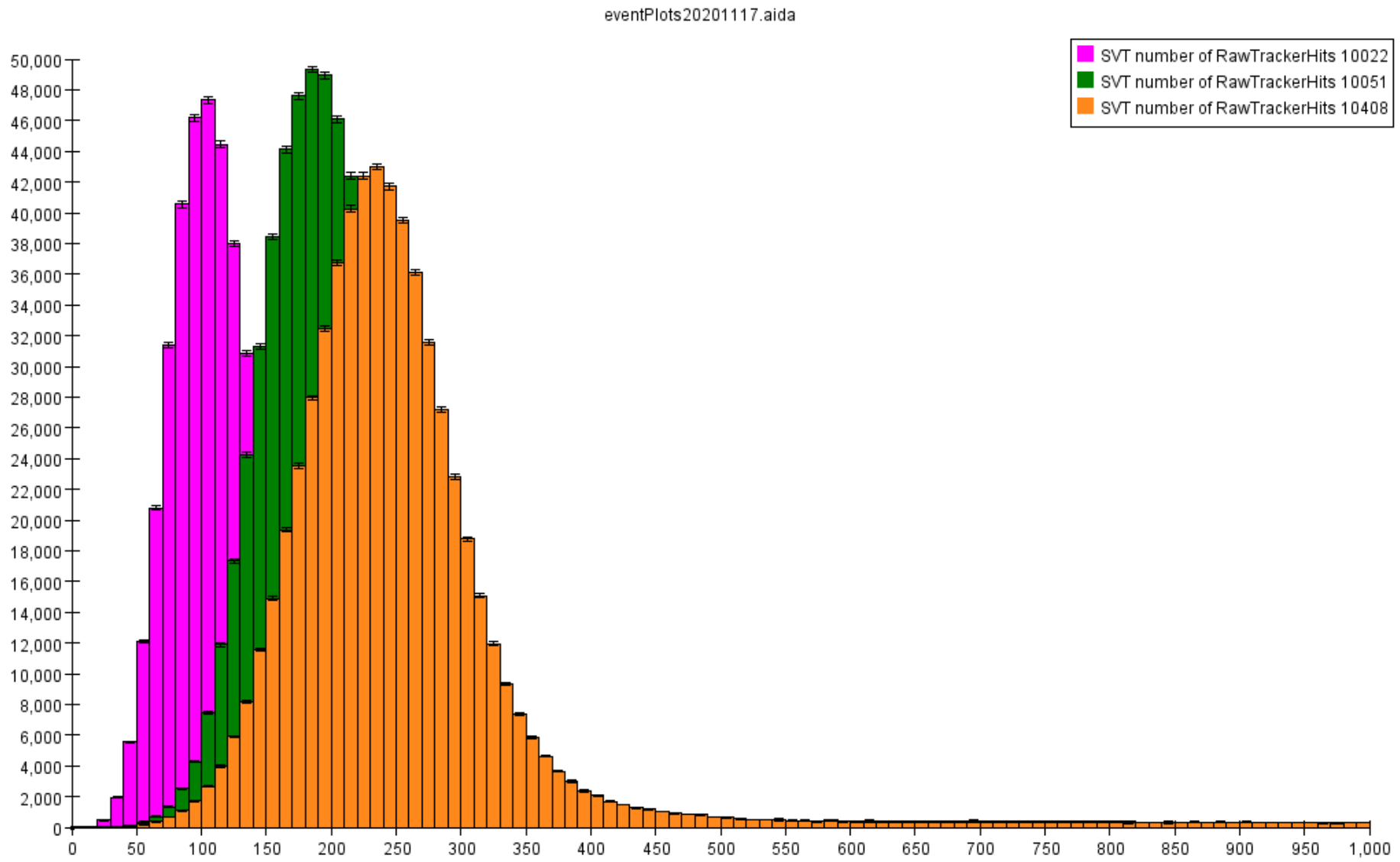
Cutting at 200 or 250 seems ~reasonable for this run.

n.b. These are strip hits, not StripClusterer_SiTrackerHitStrip1D, so cut would be closer to 300-400 on this plot.

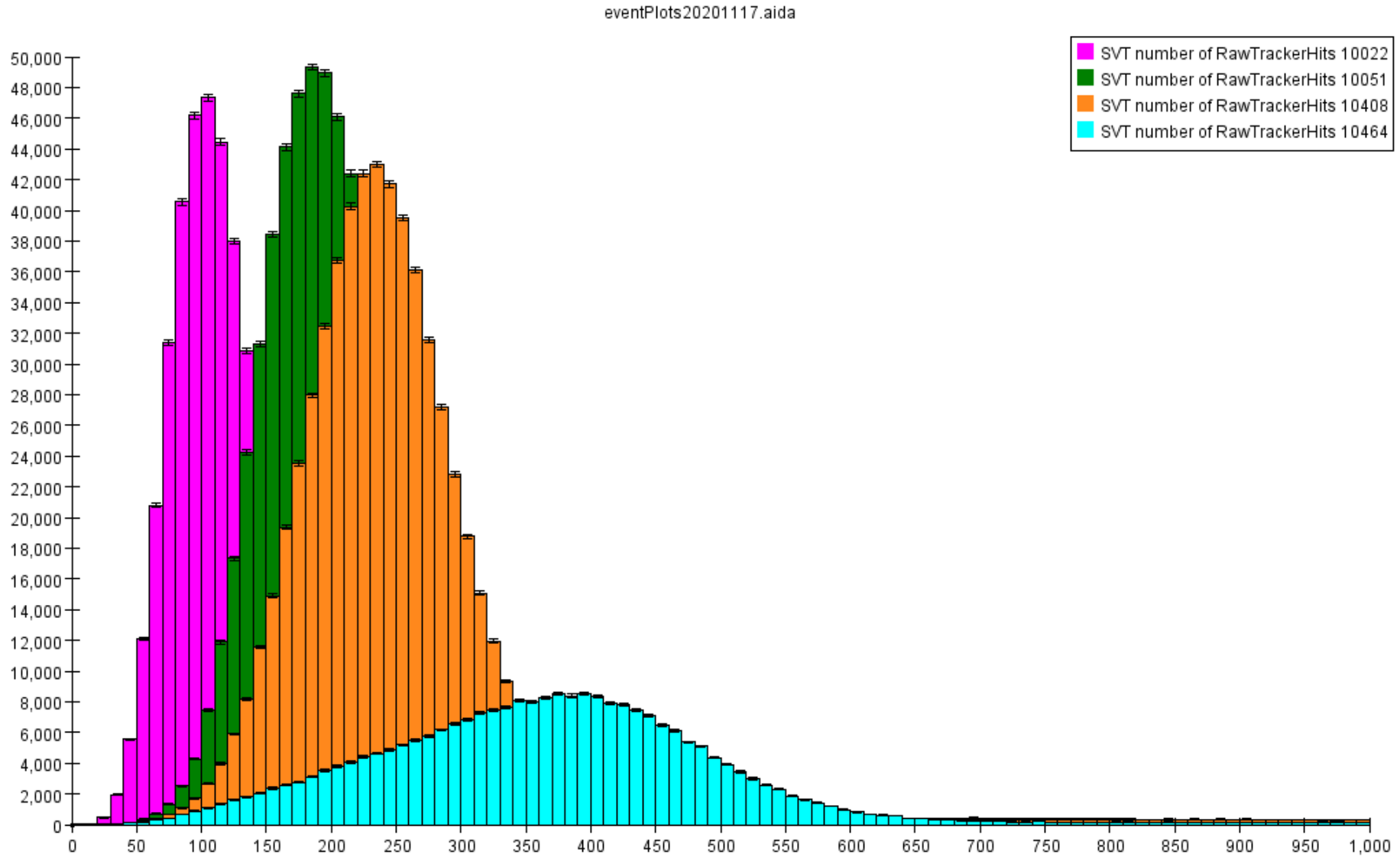
SVT Number of Raw Tracker Hits



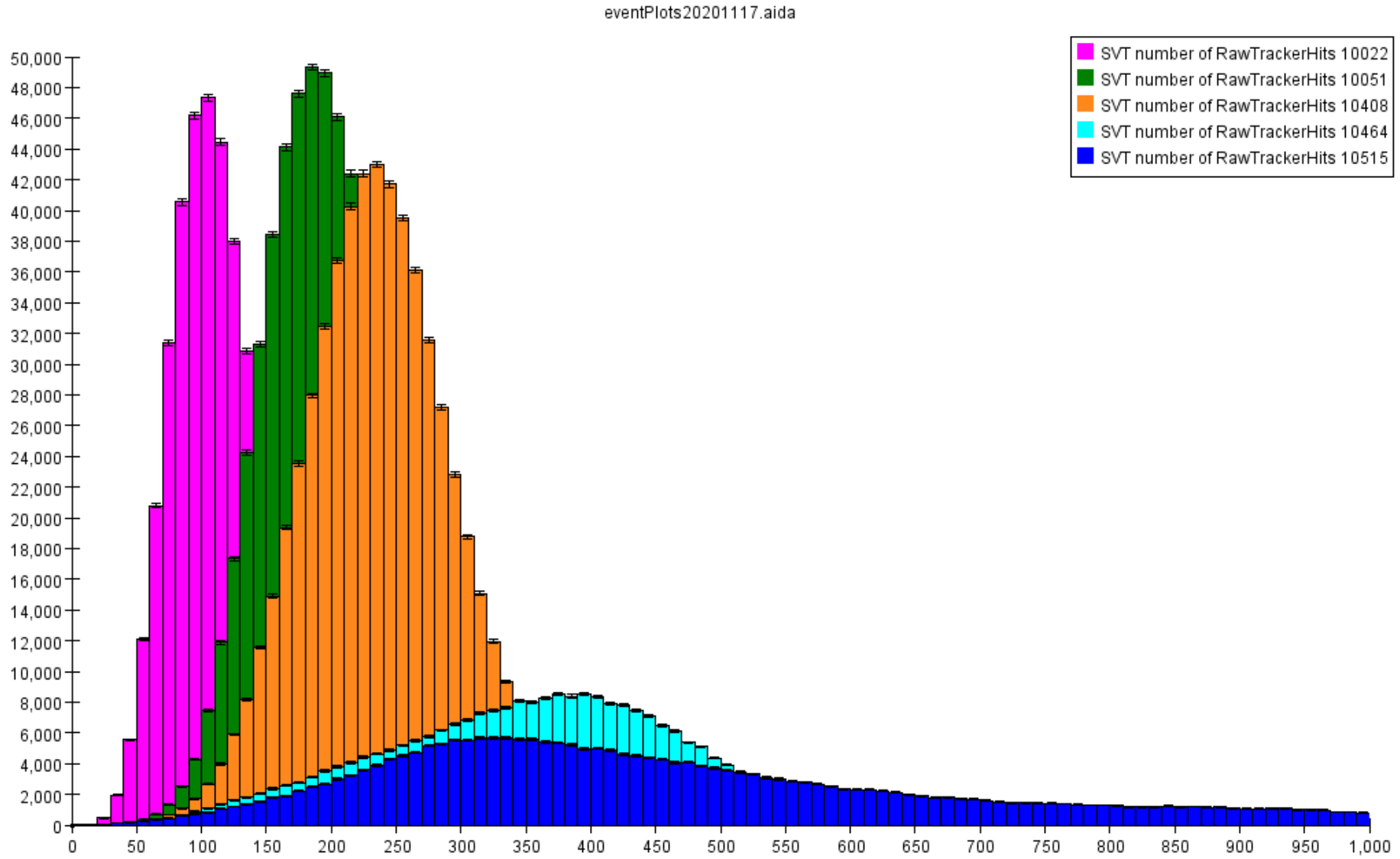
SVT Number of Raw Tracker Hits



SVT Number of Raw Tracker Hits



SVT Number of Raw Tracker Hits

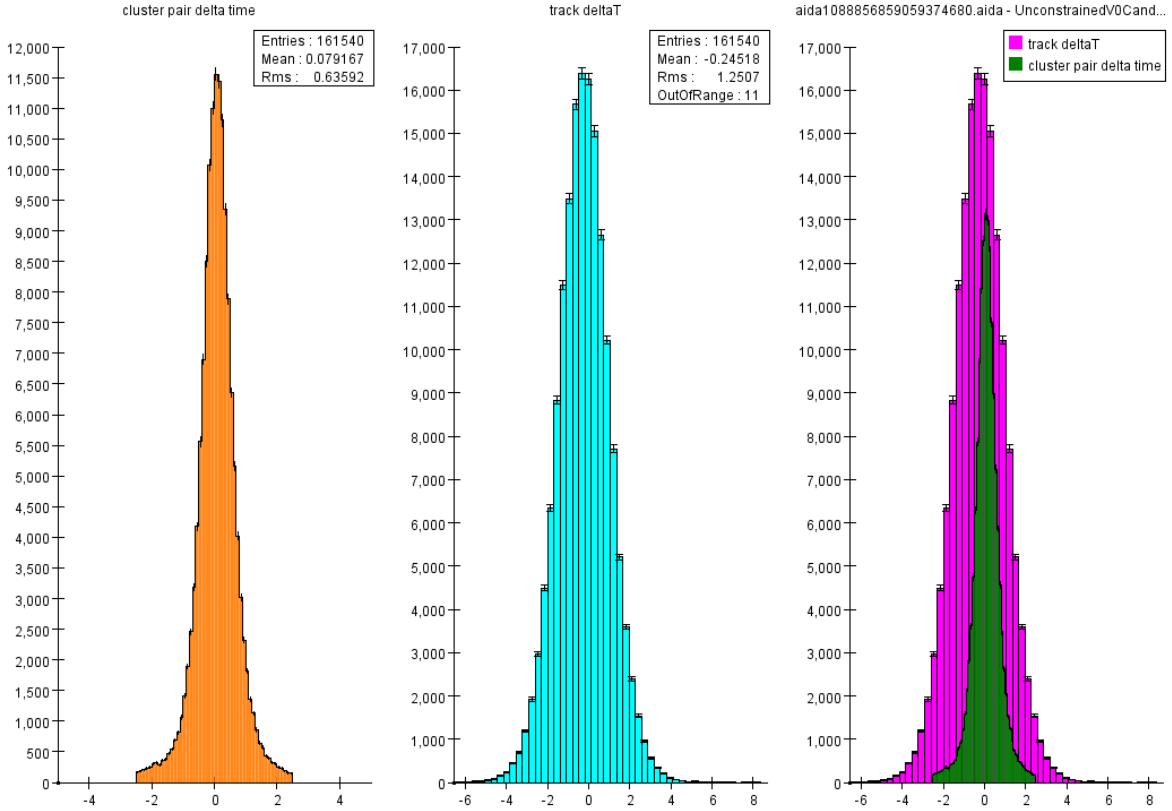


SVT "Monster" Events

- Plan is to identify and then skip both processing and writing out the event.
  - This is new behavior as in the past we simply flagged such events.
- A skim of events containing more than 250 SVTRawTrackerHits is available to characterize the issues, develop the algorithms and test the efficiency of the cuts.
- git issue iss731 addresses this.

Track Timing vs Ecal Timing

- Select V0 candidates with Ecal Clusters associated with each track.
- Track timing resolution a factor of two worse.
- Good enough?



Track Finding

- Currently running both SeedTracker and Kalman Filter for track finding.
- Need to characterize the performance and optimize track-finding efficiency and CPU time.
- May need condition-dependent strategies
  - e.g. no need to include a “dead” layer in either the seed or confirm stage of pattern recognition.
- Characterizing track-finding efficiencies
  - ECal FEEs $\rightarrow$ clean sample of high-energy electrons
  - ECal WABs $\rightarrow$ clean sample of lower-energy electrons
  - Positron trigger (coincidence of hodoscope hit with calorimeter cluster) provides clean sample of positrons

Logistics

- We need good estimates of our CPU needs to process the full 2019 “good” data sample
  - ~50 Billion events
  - Goal is better than 10Hz with a memory footprint of less than 1 GB
- We need good estimates of the amount of computing power we can rely on.
  - will be competing with other experiments for processing resources
- We need good estimates of our storage needs
  - ~600TB of evio data
  - How large is the recon output?
- Will inform the overall HPS data processing plan

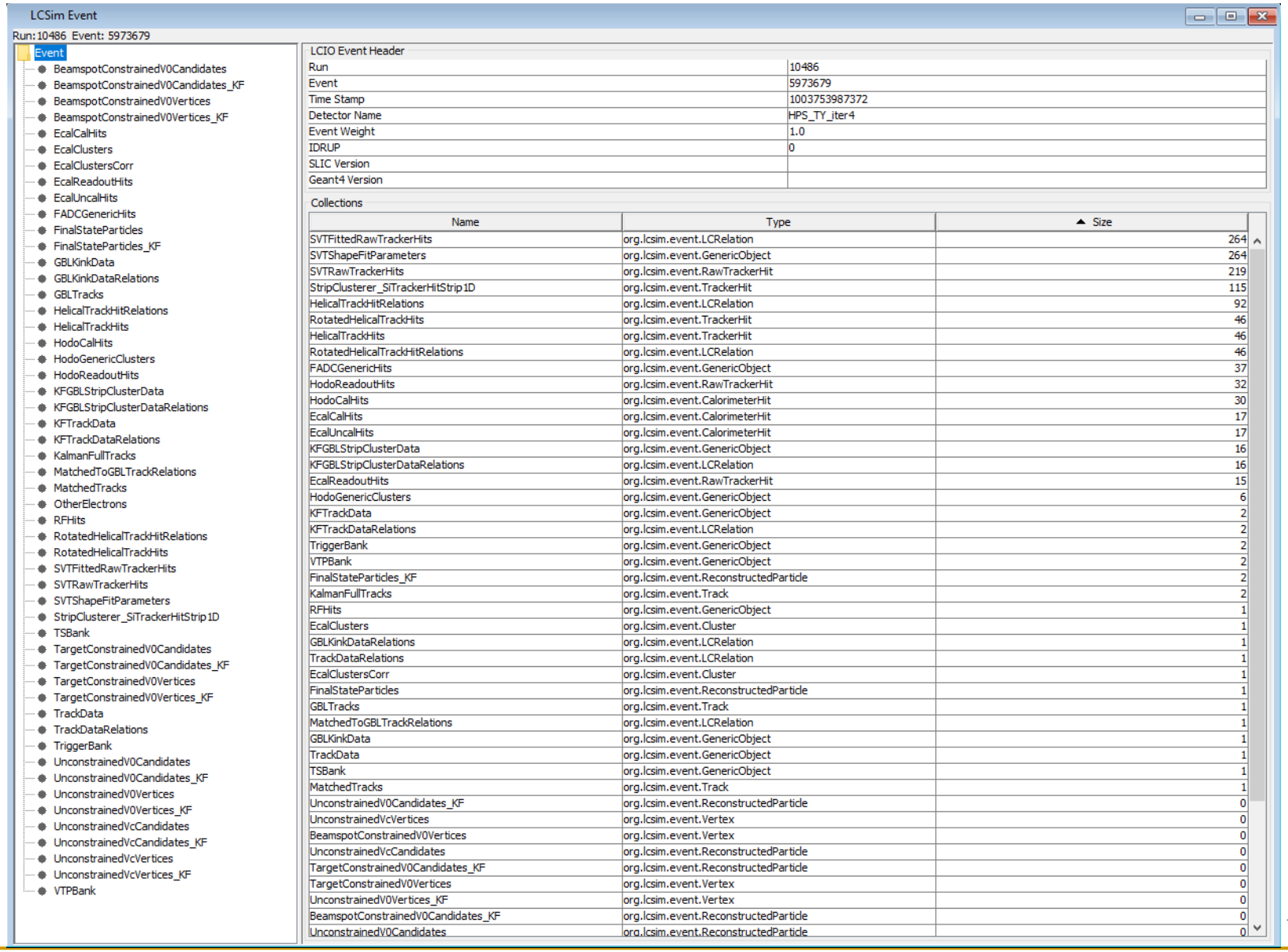
Sample Partitions

- Essential to test on a faithful subset of the full 2019 data run.
- Partitions sampled ('.*04[12]$') from the list of 282 “good” runs:
  - Roughly 150M events (142937602)
  - Roughly 3‰ of the 2019 “good” data (857 / 276339)
- Using:
  - EvioToLcio
  - pass1-dev_fix + iss732-refactor
  - PhysicsRun2019_pass0_recon.lcsim
  - HPS_TY_iter4

Output File Size Status

- Status of the reconstruction is still in flux.
- Little (no?) effort has been devoted to limiting content or file size.
- Effort concentrated on understanding efficiency, resolution, etc. i.e.”physics” performance.
- Production Reconstruction will differ substantially.
  - Latest “pass0” steering file includes both SeedTracker/GBL & Kalman Filter to enable comparison of tracking.
- Nevertheless…

Recon Output



Recon Output

- So, it's clear that there is a LOT of extra data included in this file.
  - For instance, we won't have both SeedTracker/GBL and Kalman Filter tracks and ReconstructedParticles.
- Won't try to analyze every collection here, but it's clear that we need to survey what's going into the output and justify what's there.

What is the role of the recon file?

- Historically we have kept all of the data, including the raw data, to enable re-reconstruction from the lcio files.
- At this point, we should be able to drop the raw waveforms and only save the fitted t0 and pulse area.
- Obviously we need to include all the information for subsequent “physics” analyses.
- But, do we need to save individual SVT readout channels, or can we live with just StripClusterer_SiTrackerHitStrip1D?

Output Data Size Reduction

- A number of strategies can gain us a substantial reduction in the size of our recon output files.
- Dropping the “raw” waveforms is easiest.
  - Are we satisfied with our current pulse fitting?
- Not running the SeedTracker is straightforward
  - Need to validate Kalman Filter.
    - More, better tracks faster is a requirement.
- Pruning un-needed collections is next.
- Can consider DST set of collections which is optimized for “physics” analysis.
  - Just ReconstructedParticles?

Software Issues

- Need a major effort to merge all of the extant git branches back into master
  - both hps-java and lcsim
  - Must retain backwards-compatibility for 2016/2019
- Need to tag and make a library release before production processing.

Summary

- This list of topics is long, but not exhaustive.
- There is still a LOT of work to be done before we can start the full production processing of the 2019 data.
- Current focus is still on characterizing the detector and “physics” performance
  - alignment, calibration, track-finding efficiency, energy & momentum scale and resolution, etc.
- Much work has gone into speeding up the code along the way, more will most likely need to be done.
- Memory requirements may be OK
  - may be due to dropping high-multiplicity events.
- ~No effort expended on reducing output file size
- Need more involvement and feedback from other members of the collaboration!

This means YOU!