

# Track-Cluster (non)matching investigation

Matt Graham

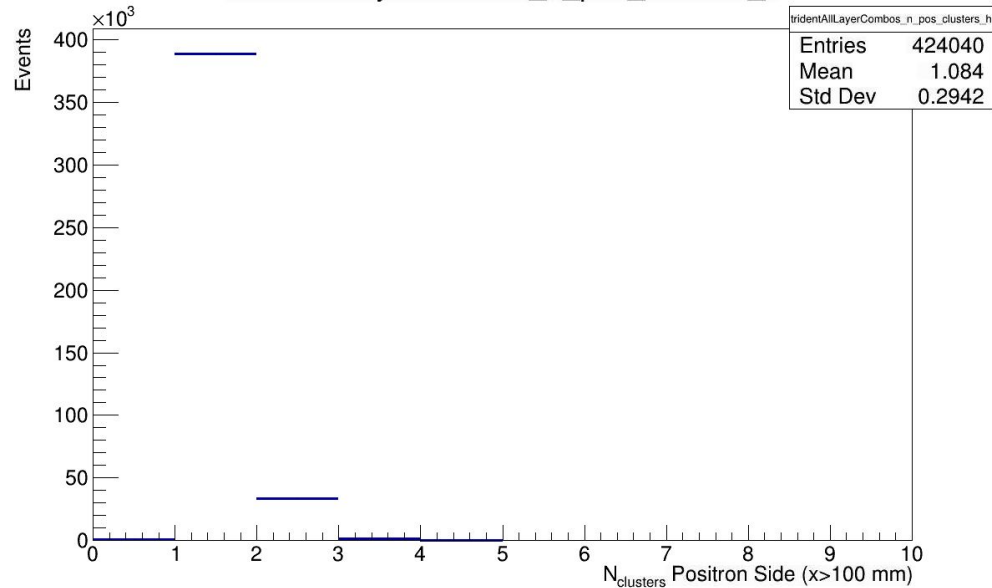
Tuesday August 22, 2023

# The issue

In [my analysis workshop talk](#), I noticed a very large number of tracks from V0s did not have a match to a cluster

- Most disturbing, this was true for positrons too
- The ratios of events for “both”, “pos-noele”, “nopus-ele”, “no-no” in data is: 1.0:0.77, 0.50, 0.41
  - a bit different from ana talk, but I tweaked some things, ran over smaller sample, no L2 requirement etc...conclusion is the same, lots of V0s with no positron cluster match
- Trigger required a cluster on the positron side, so there must be a cluster there...it's just not matching
  - I'm looking at 2021 pass0 data and we know we have issues with alignment that makes the track projection at the ECal skewed...but the requirements on matching are “wide-open”
  - Also an issue that pass0 used the wrong z position of ECal face...

# Is there a positron cluster in these events?



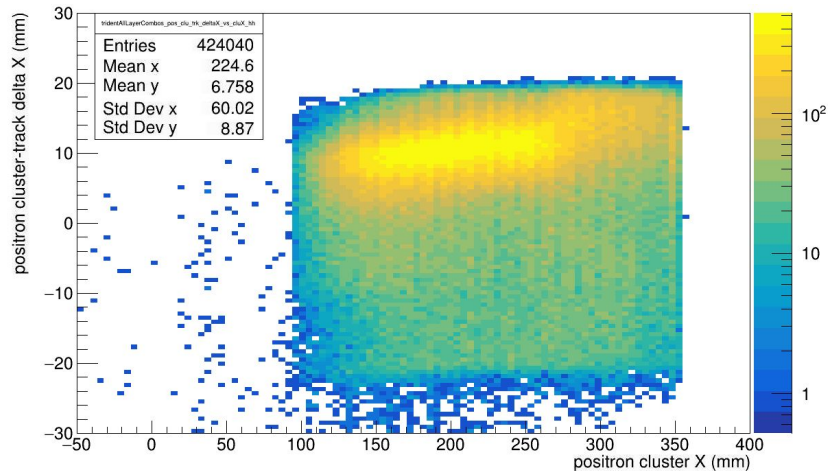
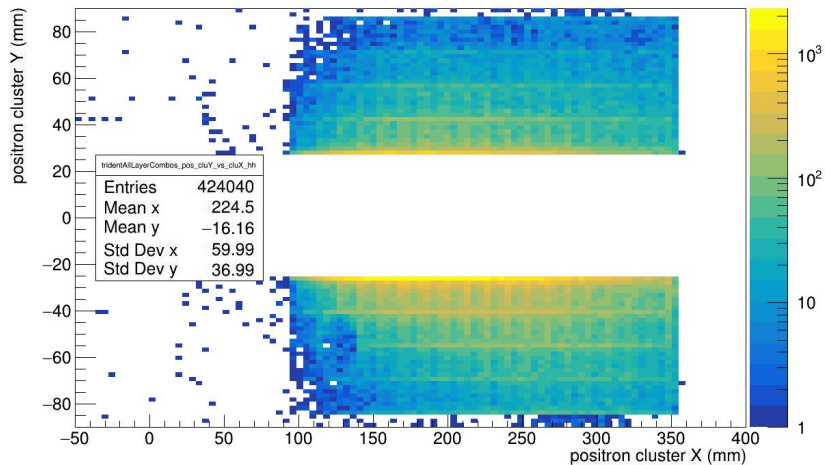
The requirements here:

pass singles2 or 3  
has at least 1 reconned V0 (with loose timing/p/pSum etc. cuts)

Define a “positron cluster” as cluster with  $X > 100\text{mm}$  at eCal

So, yes, pretty much all of these events have positron clusters in them.

# Is track-cluster matching “wide-open”



...not really; looks like there is a cut at  $\sim \text{delta} \pm 20\text{mm}$  (but not exactly and asymmetric?)  
deltaX offset by  $\sim 10\text{mm}$  with positive, so that's going to cut off a bunch of lower-p tail events

# Some code:

We use `TrackClusterMatcherMinDistance.java` to match tracks to clusters, and this uses cuts defined in `StandardCuts.java`. There's a 20mm cut on both the X and Y maximum match distance. Also a cut on delta-t, but that is set to 40ns (big) in the particle driver.

Looking at the production of pass0. Options used in [HpsReconParticleDriver.java](#) pertinent for cluster matching:

```
<maxMatchDt>40</maxMatchDt>
<trackClusterTimeOffset>40</trackClusterTimeOffset>
```

```
<useCorrectedClusterPositionsForMatching>false</useCorrectedClusterPositionsForMatching>
```

```
<applyClusterCorrections>true</applyClusterCorrections>
```

```
<useTrackPositionForClusterCorrection>true</useTrackPositionForClusterCorrection>
```

But there are `StandardCuts` used:

`hps-java/record-util/src/main/java/org/hps/record/StandardCuts.java`

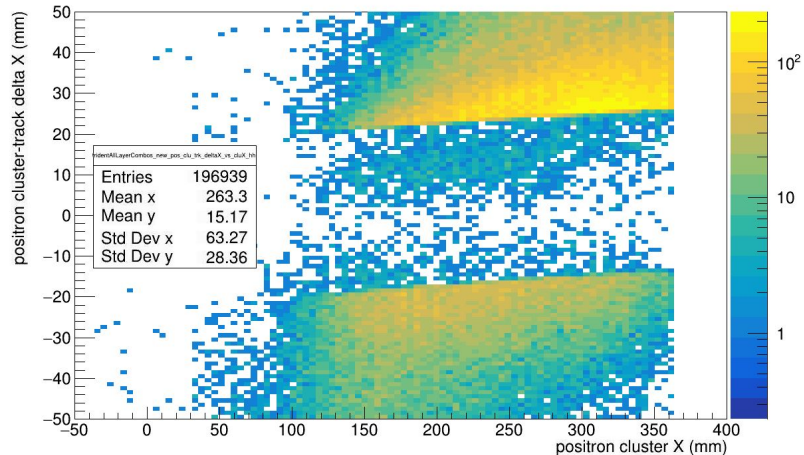
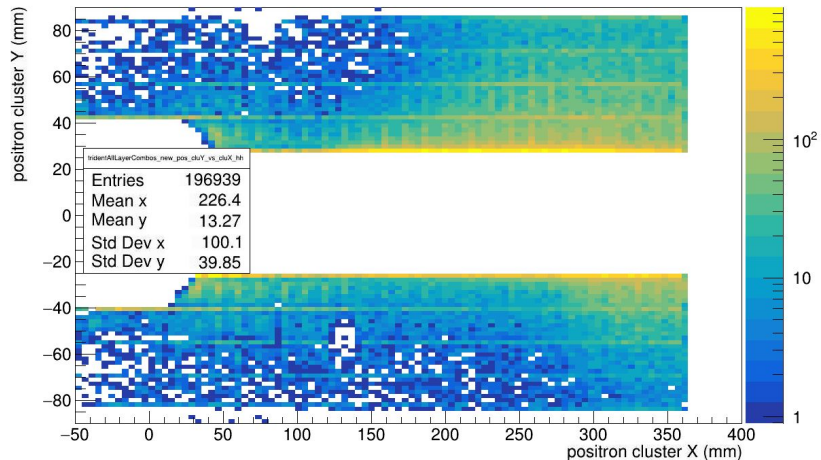
```
public StandardCuts(double ebeam) {
    maxSharedHitsPerTrack = 5;
    maxMatchChisq = 10.0;
    maxMatchDt = 6.0;
    maxMatchDx = 20.0;
    maxMatchDy = 20.0;
    maxVertexClusterDt = 2.0;
    minVertexChisqProb = 0.00001;
    minMollerChisqProb = 0.00001;
    maxTrackChisqProb = 0.00001;

    maxElectronPset = false;
    minMollerPset = false;
    maxMollerPset = false;
    maxVertexPset = false;
    OffsetSet = false;

    maxTrackChisq = new HashMap<Integer, Double>();
    maxTrackChisq.put(5, new
ChiSquaredDistribution(5).inverseCumulativeProbability(1.0-maxTrackChisqProb));
    maxTrackChisq.put(7, new
ChiSquaredDistribution(7).inverseCumulativeProbability(1.0-maxTrackChisqProb));
```

```
    maxElectronP = 0.75*ebeam;
    minMollerP = 0.8*ebeam;
    maxMollerP = 1.2*ebeam;
    maxVertexP = 1.2*ebeam;
    if (ebeam < 2)
        trackClusterTimeOffset=43;
```

# Really opening up the matching



In hpstr, I did my own matching algorithm:

- track & cluster must point to same half
- $|\text{track-cluster time}| < 10\text{ns}$

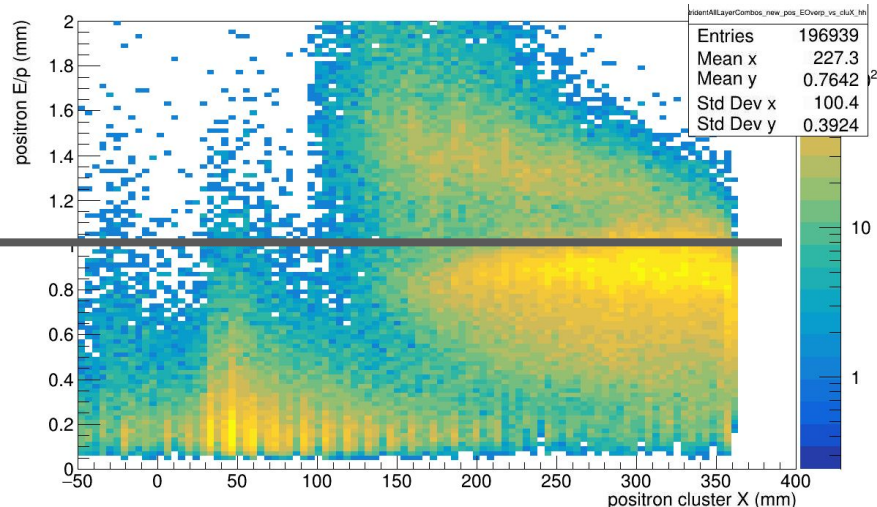
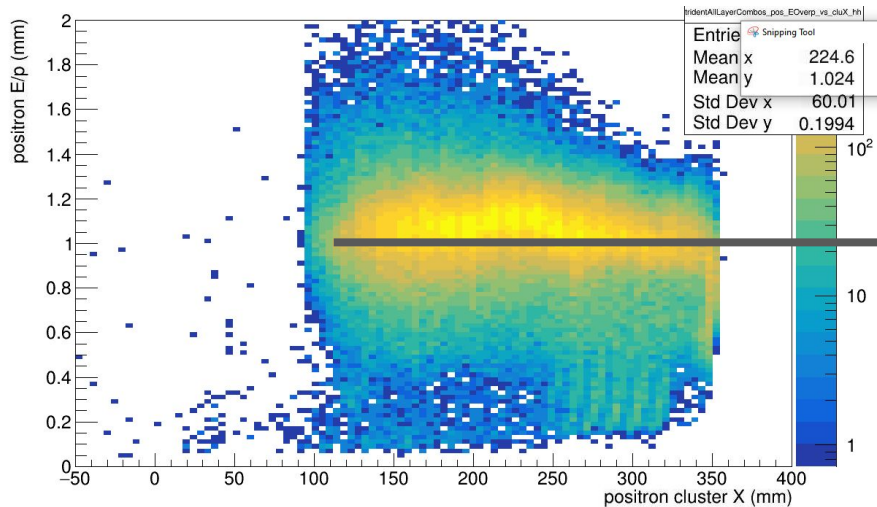
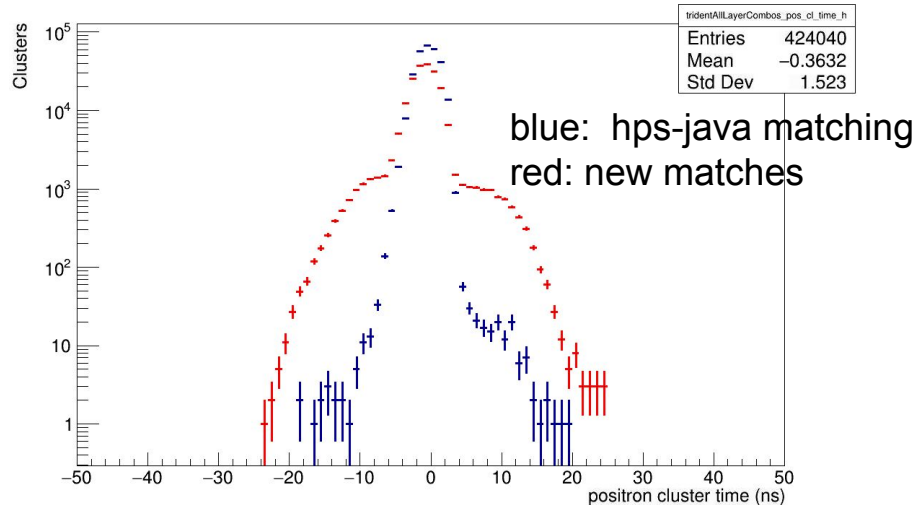
...and that's it.

It looks like this finds the matches at higher x (~lower p)

Also a lot of what looks like junk...

# Positron Match Quality

Most of these new matches look like try matches but, as inferred from previous plot, there are clearly some mis-matches here...I'm not trying to replace track-matching algorithm.



# Conclusions

- My unmatched clusters in pass0 are due to a combination of matching cuts in hps-java and poor track extrapolation (due to wrong ECal face and alignment).
- This is good, it will get better as we run on the newer detectors