

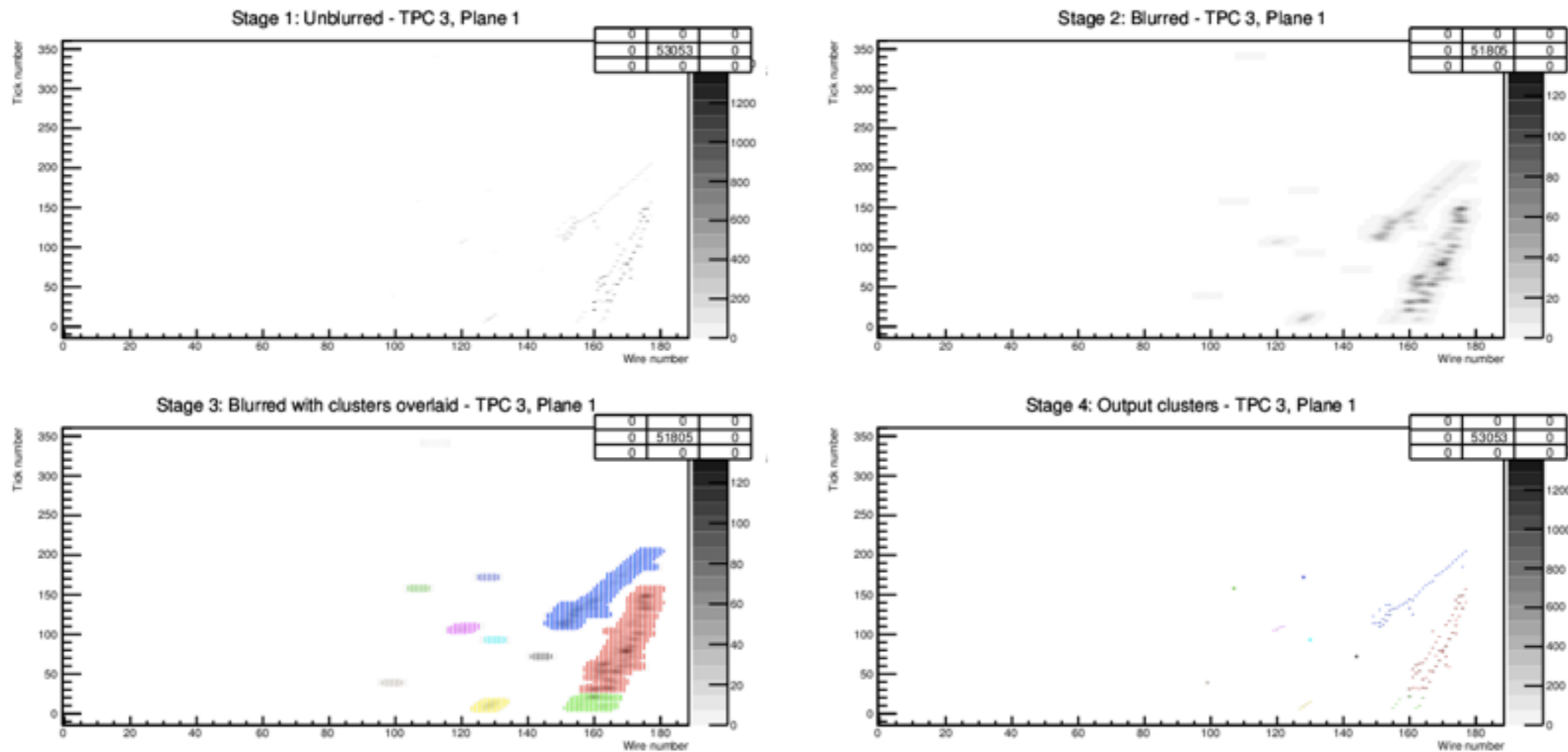
Blurred Clustering: Dynamic Blurring

Mike Wallbank

24/6/15

Reminder: Blurred Clustering

- Clustering technique which uses a Gaussian smearing to produce more full and complete clusters.



- Blurs the hit map and then clusters neighbouring hits before removing the 'fake hits'.

Characterising the Clustering

- To facilitate a evaluation of the effectiveness of the various clustering methods, a few quantities are used:
- **Completeness:** How complete a cluster is, i.e.
 $\text{hits in cluster} / \text{hits left by particle}$
- **Cleanliness:** How clean a cluster is, i.e.
 $\text{hits associated with particle} / \text{all hits in cluster}$
- **Efficiency:** Uses a cut (currently requires a view to have at least two clusters each at least 50% complete);
 $\text{quantity(passes cut)} / \text{quantity(all)}$

Changes...

- Before this week, the blurring had the same effect in each direction. Doesn't make much sense, considering the scales and resolution of wire and tick are uncorrelated.
- This has now been changed so that the blurring is separate in each direction, and can be set separately.

Free Parameters

- There are **many** free parameters in the algorithm!
- The blurring uses a Gaussian kernel...

$$\frac{1}{\sqrt{2\pi}\sigma_{wire}} \frac{1}{\sqrt{2\pi}\sigma_{tick}} e^{-\frac{r_{wire}^2}{2\sigma_{wire}^2}} e^{-\frac{r_{tick}^2}{2\sigma_{tick}^2}}$$

where r determines how large the kernel is (how far to blur) and sigma is the standard deviation.

- Right now $\sigma_{tick} = 1.5\sigma_{wire}$

Free Parameters

- BlurWire — blur radius in wire direction
- BlurTick — blur radius in tick direction
- BlurSigma — sigma for Gaussian kernel
- ClusterWireDistance — how far to cluster in wire direction
- ClusterTickDistance — how far to cluster in tick direction
- NeighboursThreshold — number of neighbours needed to cluster a hit
- MinNeighbours — min number of neighbours a hit must have to keep it in cluster
- MinSize — minimum number of hits a cluster must contain to keep
- MinSeed — the minimum charge needed before any clustering occurs in a view
- TimeThreshold — time threshold to add hit to cluster
- ChargeThreshold — minimum charge before adding hit to cluster

Very little
difference!



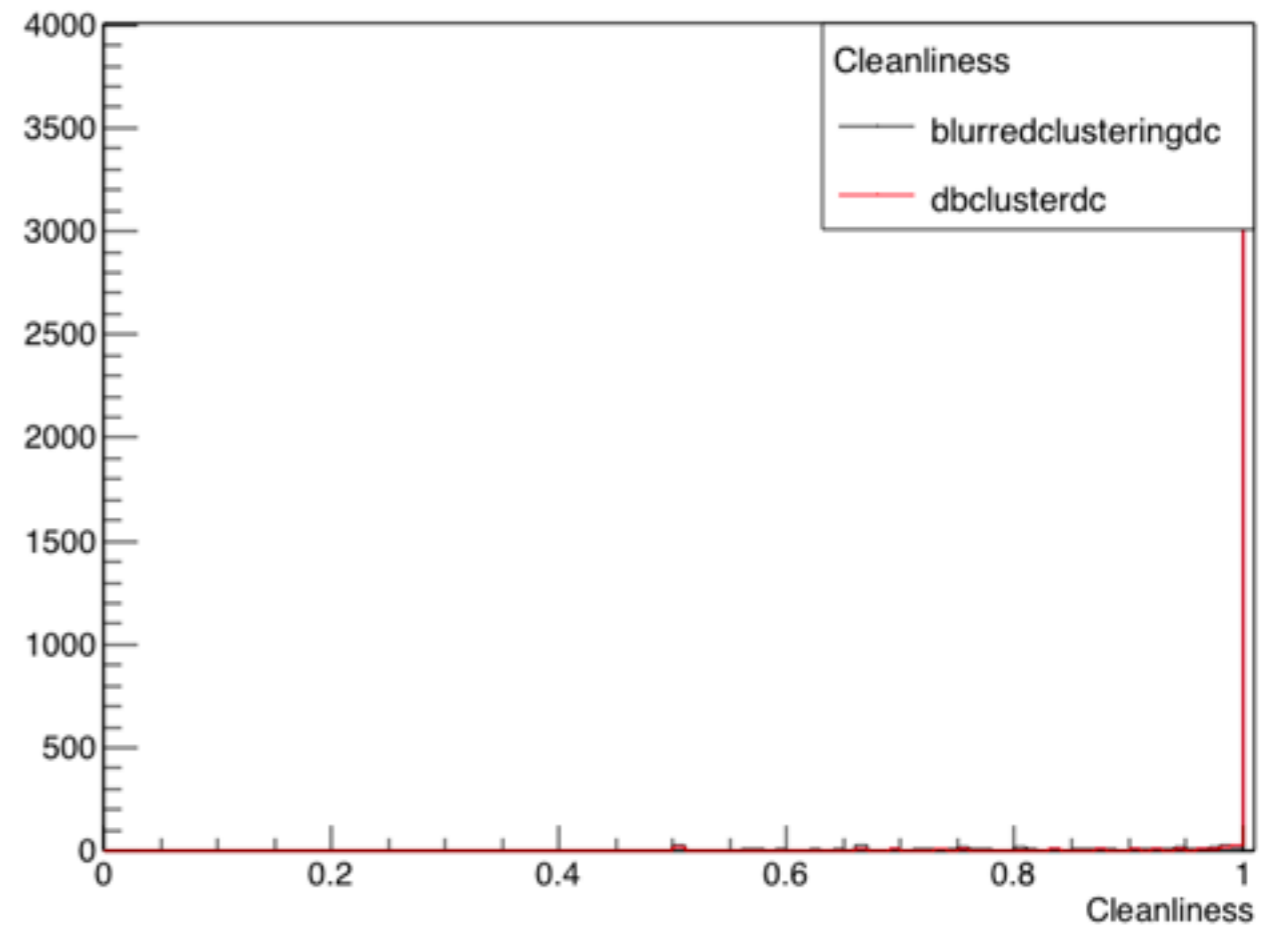
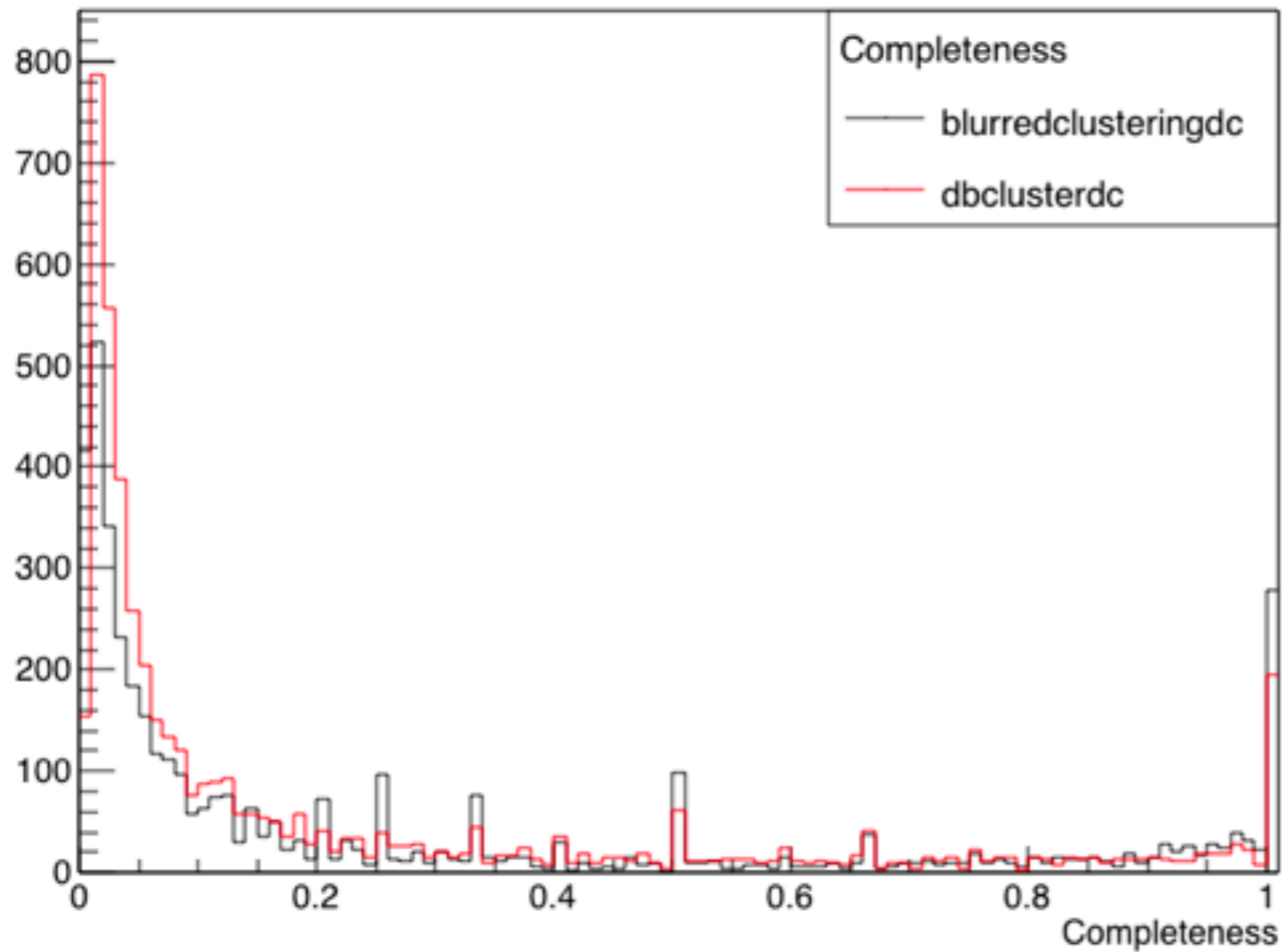
Tuning the Parameters

- I've been looking at these parameters closely over the last few weeks to find their optimum values.
 - Very difficult — there are so many and some are complimentary (e.g. blurring and clustering radii).
- To just get some idea, I ran a huge job (~4000 iterations) to vary many of the parameters over quite a coarse range
 - Didn't vary most of the parameters near the bottom of the list!
 - Learnt very little... more blurring and clustering preferred!
- Very crudely took an average of the completeness and cleanliness of all the clusters made with each combination and used the metric $\text{completeness}^2 * \text{cleanliness}$ to pick the best...

Preferred Parameters

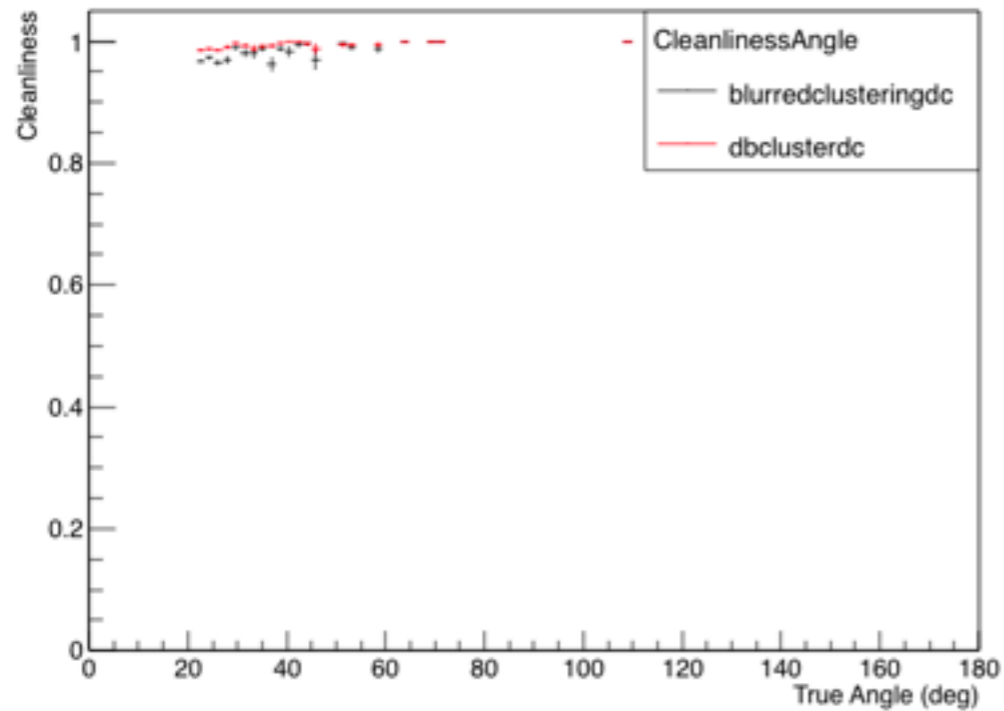
- BlurWire — 8
 - BlurTick — 6
 - BlurSigma — 4
 - ClusterWireDistance — 20
 - ClusterTickDistance — 40
 - NeighboursThreshold — 0
 - MinNeighbours — 0
 - MinSize — 2
 - MinSeed — 0.1
 - TimeThreshold — 20
 - ChargeThreshold — 0.07
- Using these parameters, I ran the reconstruction over 100 events and compared to dbcluster
 - (Same as what I showed last week, see talk from then)
 - Some comparisons...

Completeness & Cleanliness

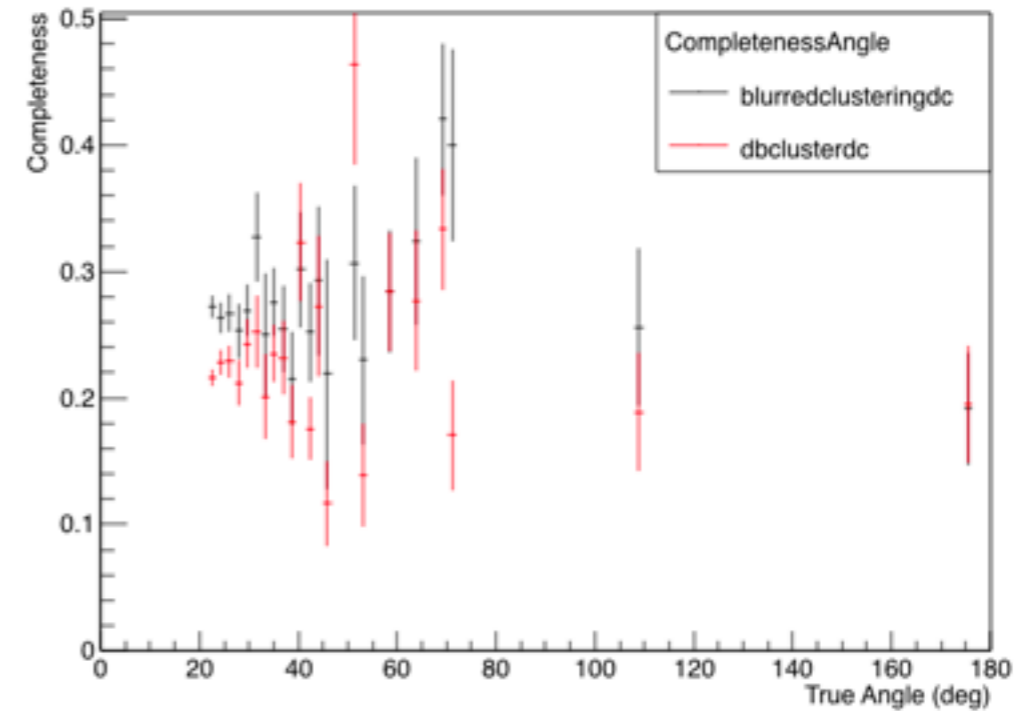


- Now outperforms dbcluster!

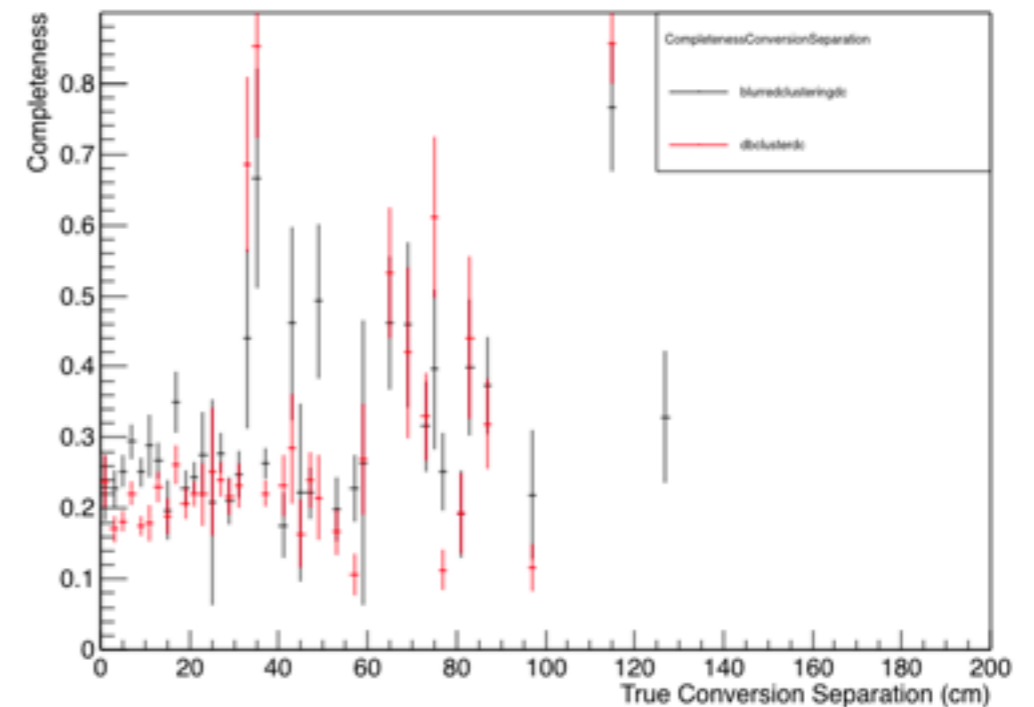
Completeness & Cleanliness



Cleanliness against
 π^0 decay angle
(always pretty clean!)

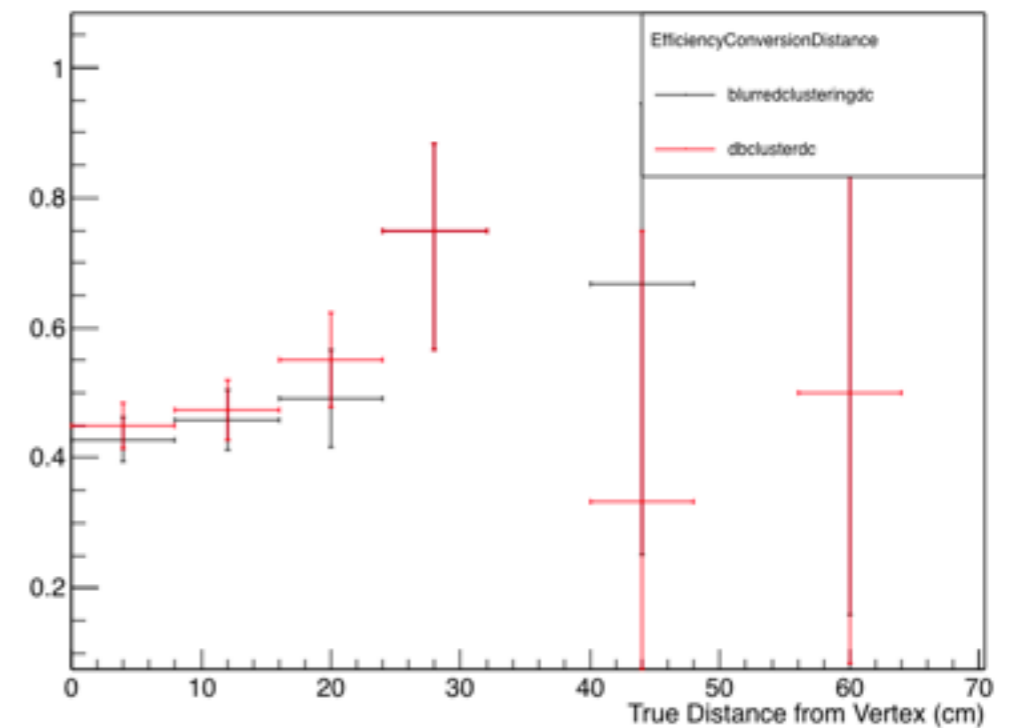
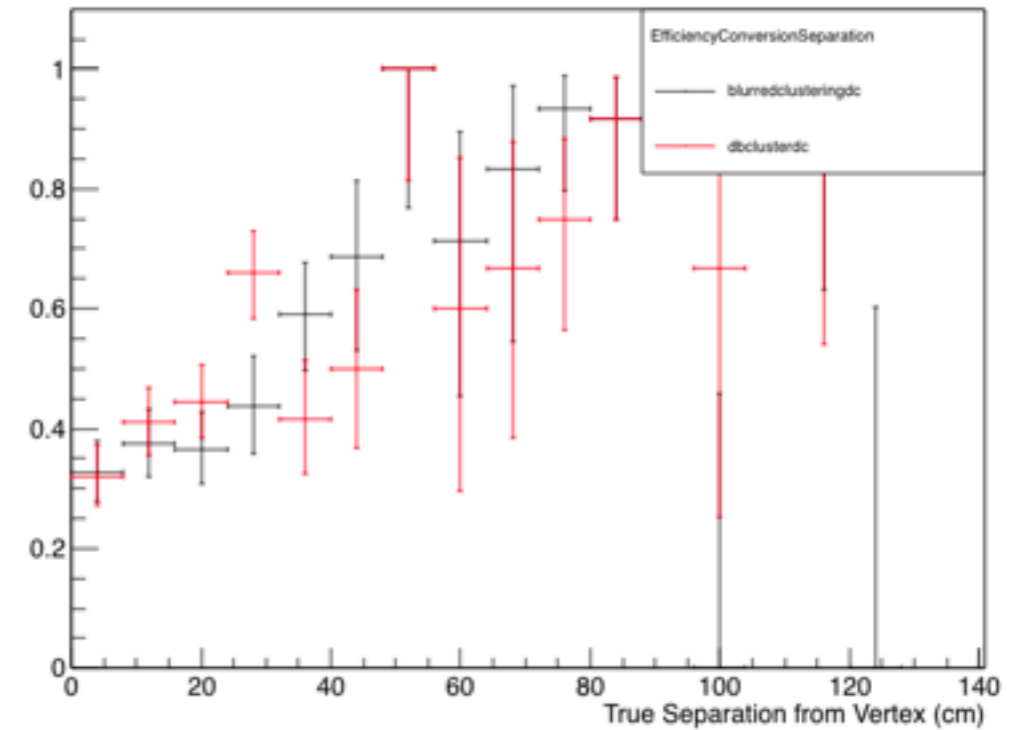
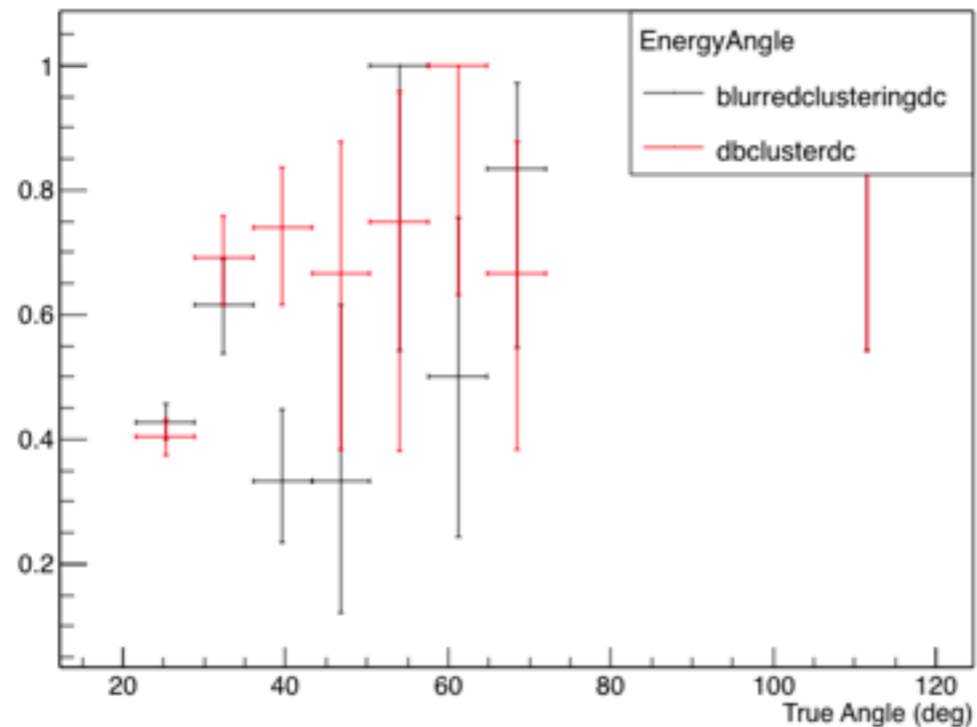


Completeness as function of angle
(top) and photon conversion
separation (bottom).
Very comparable

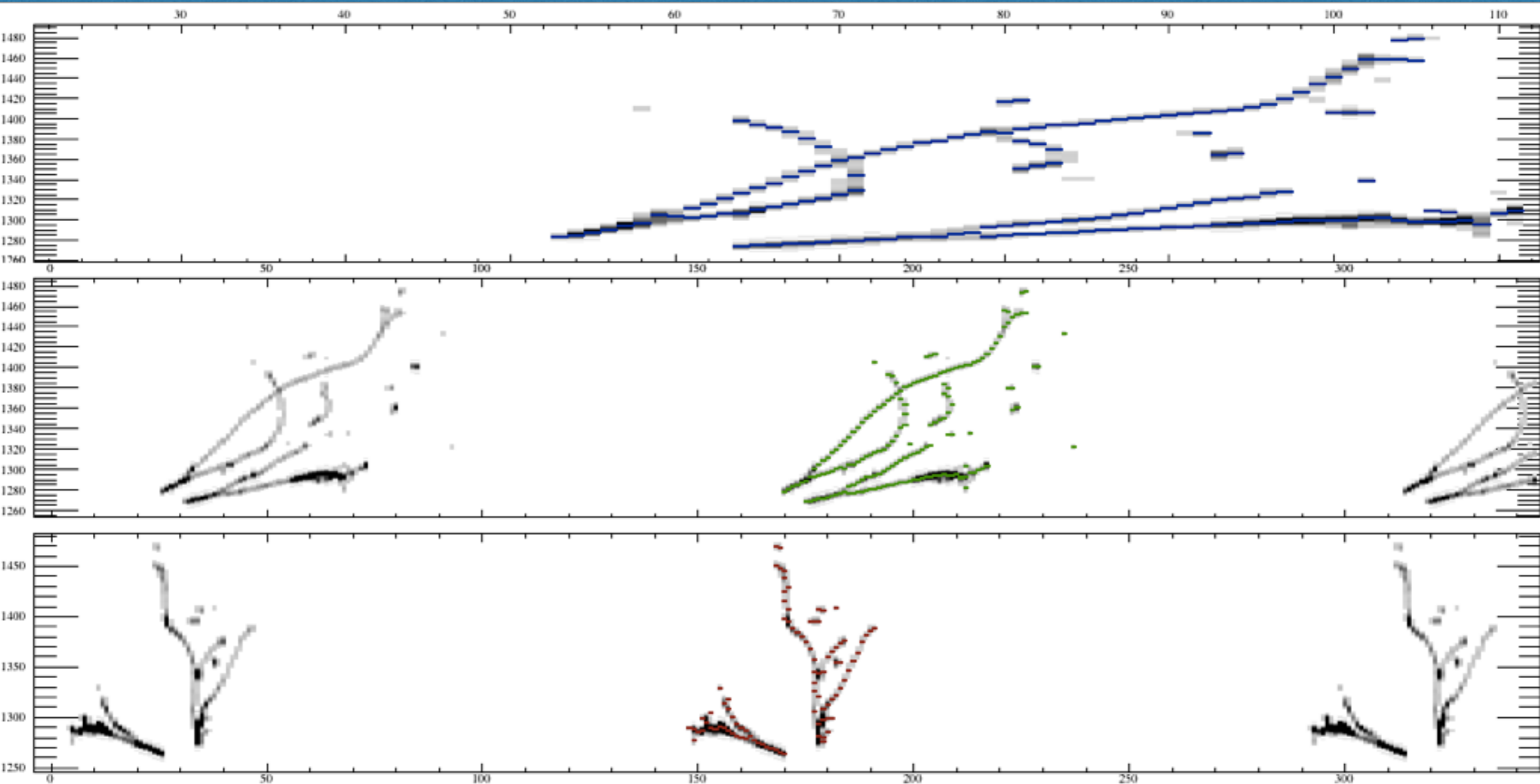


Efficiencies

Efficiencies as function of angle (below), distance from vertex (top right) and photon conversion separation (bottom right)



Next Problem



Tracks tend to travel in the similar direction and so are easily blurred and clustered together as one object...

The Solution?: Dynamic Blurring

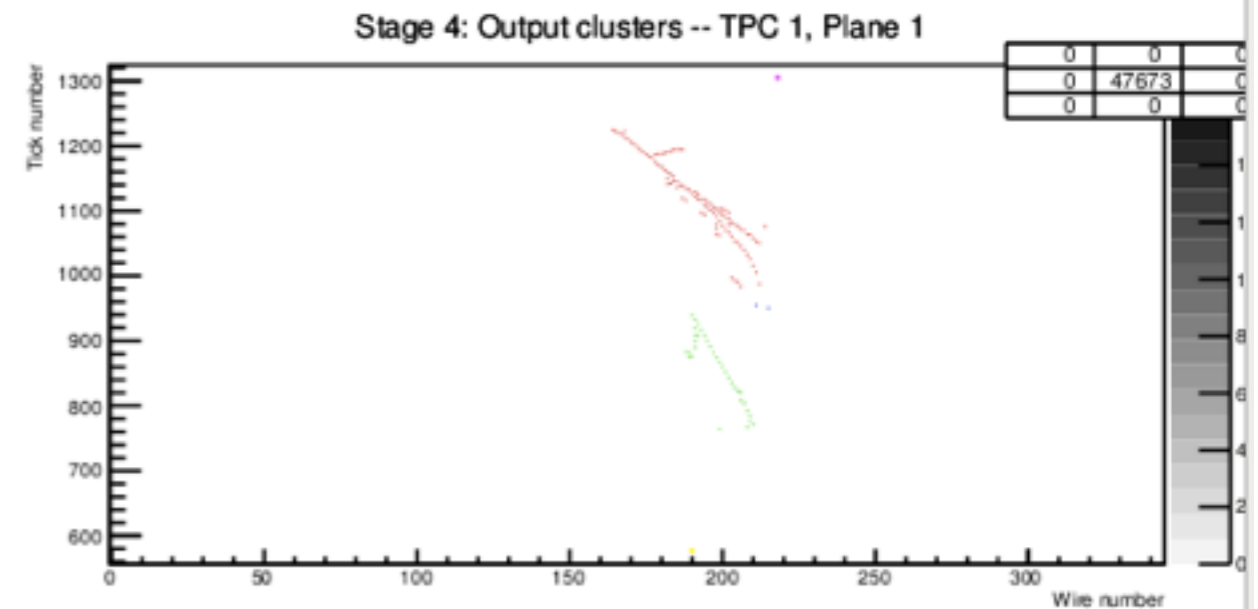
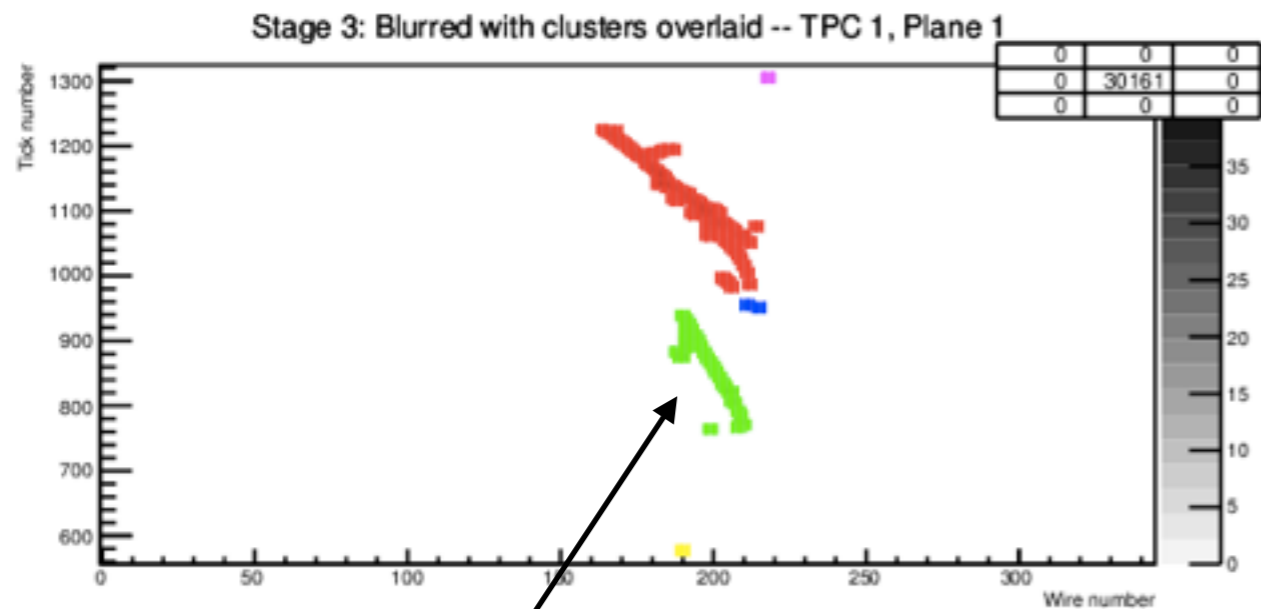
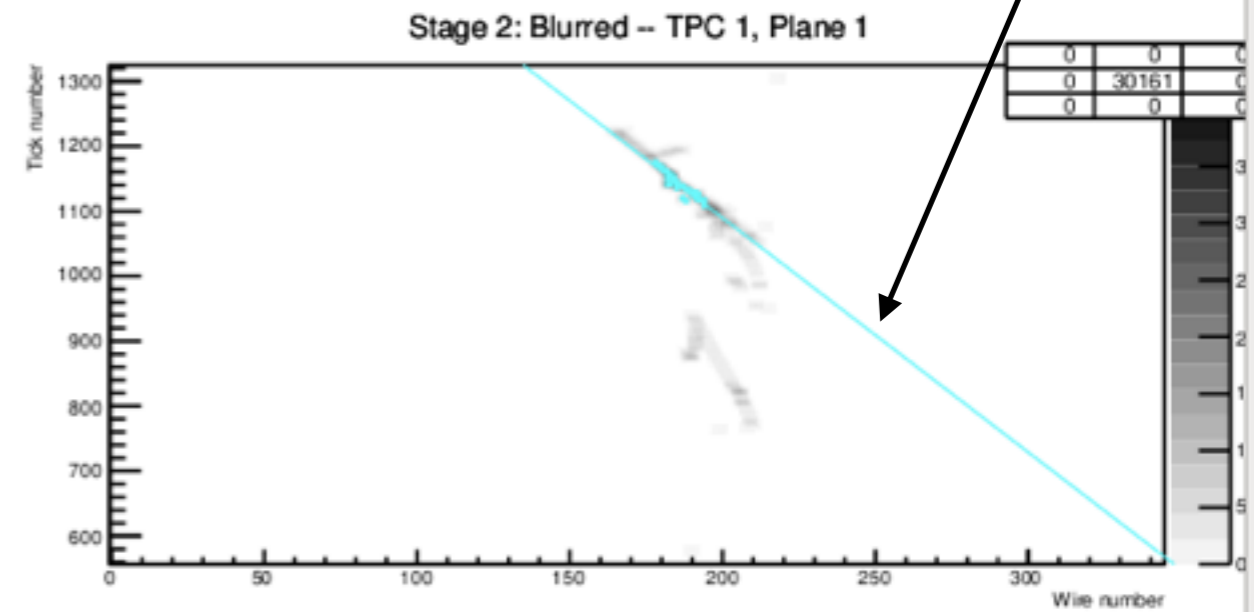
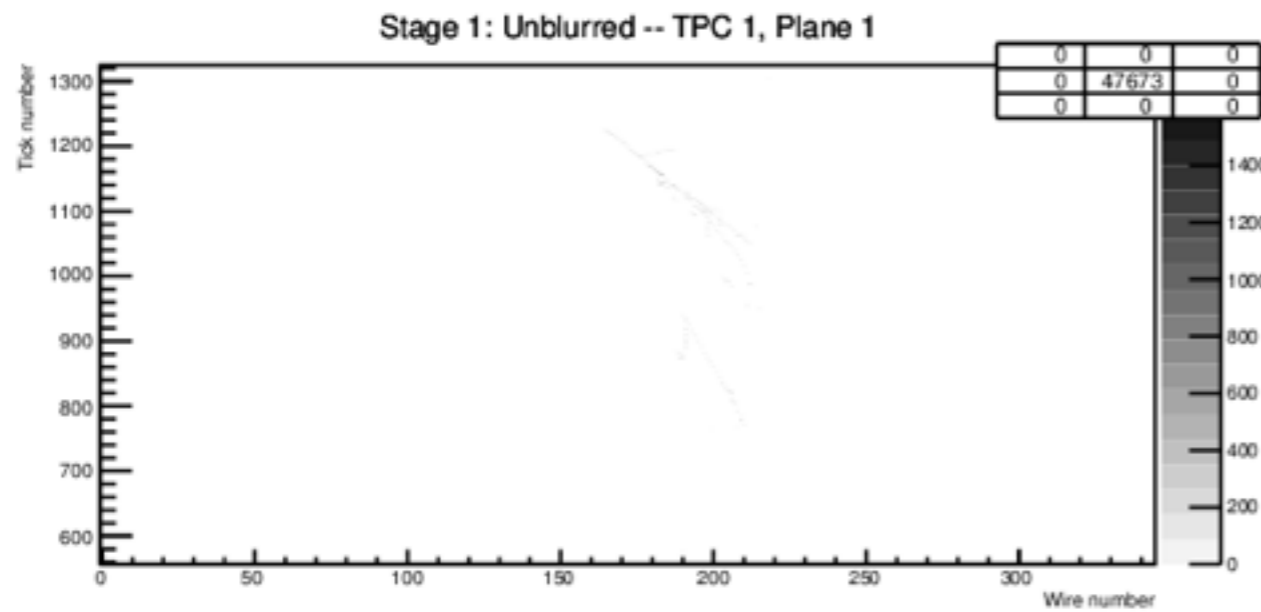
- The idea: get some idea of the direction of the tracks/showers before doing anything.
- Use this information to apply the blurring to the image as appropriate (i.e. as parallel to the tracks as possible).
- Will hopefully allow to resolve two objects which are almost parallel and would have been clustered together previously.

Dynamic Blurring

- Implementation (very crude right now!):
 - find the highest charged hit and start with this
 - put all hits within some defined surrounding window (two more parameters!) into a TGraph
 - find the best fit and take the gradient as a rough estimation of the direction of the particles
 - use this information to find appropriate blurring radii
- Assumptions:
 - highest charge hit is associated with photon (good I think)
 - the two photons are vaguely parallel (again good I reckon)

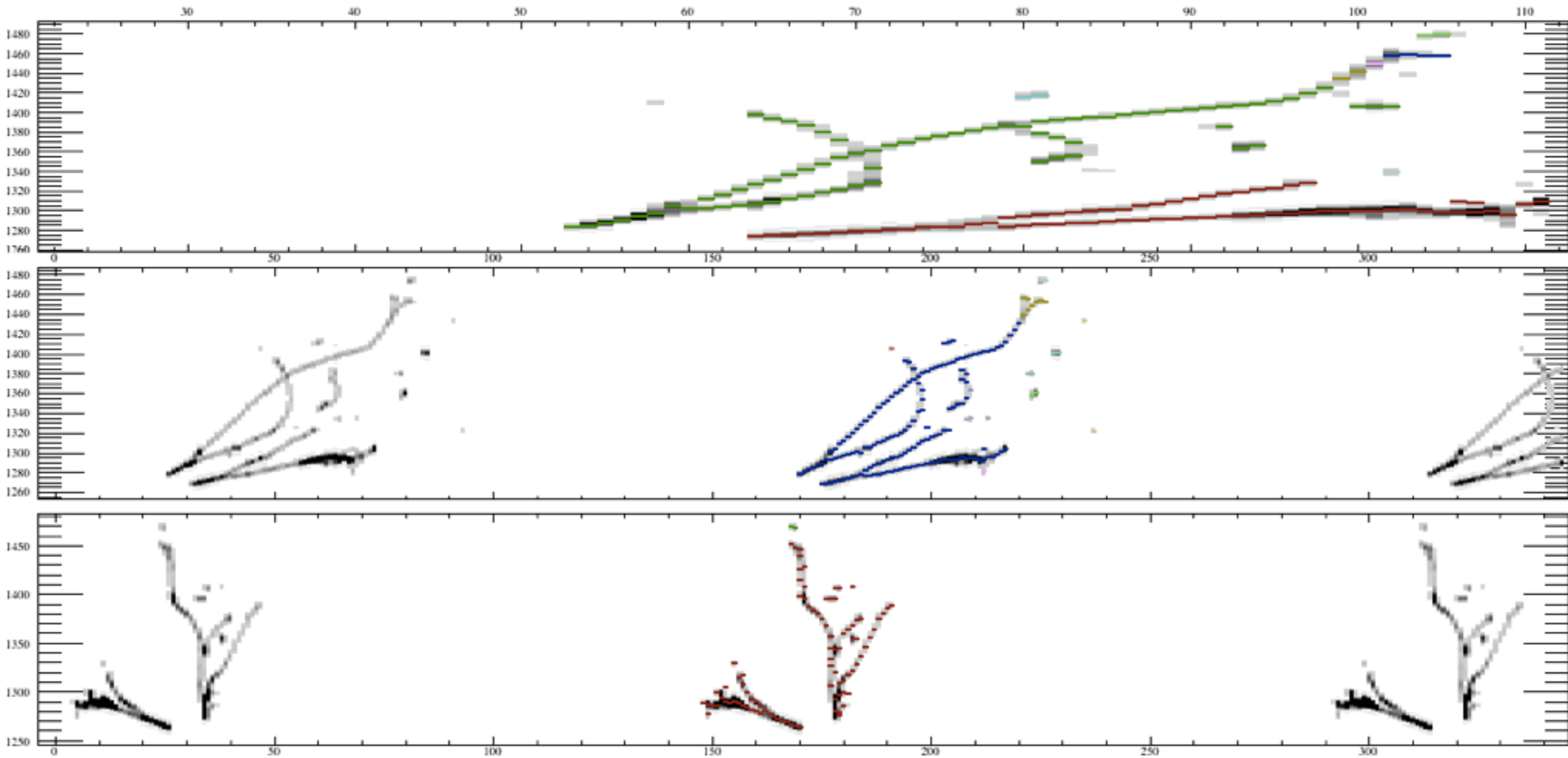
Dynamic Blurring

Line shows the rough blurring direction



Has helped to distinguish between these tracks...

The Problem from Before...



Summary

- After some tuning of the parameters, the Blurred Clustering technique now forms more complete clusters than dbcluster
 - Much more playing around with the parameters to be done!
- Still having a problem with close tracks being blurred and clustered together
 - Dynamic Blurring looks like it could help...
 - Still a lot to be looked at with this method but I feel with some better understanding it will hugely improve reconstruction!