

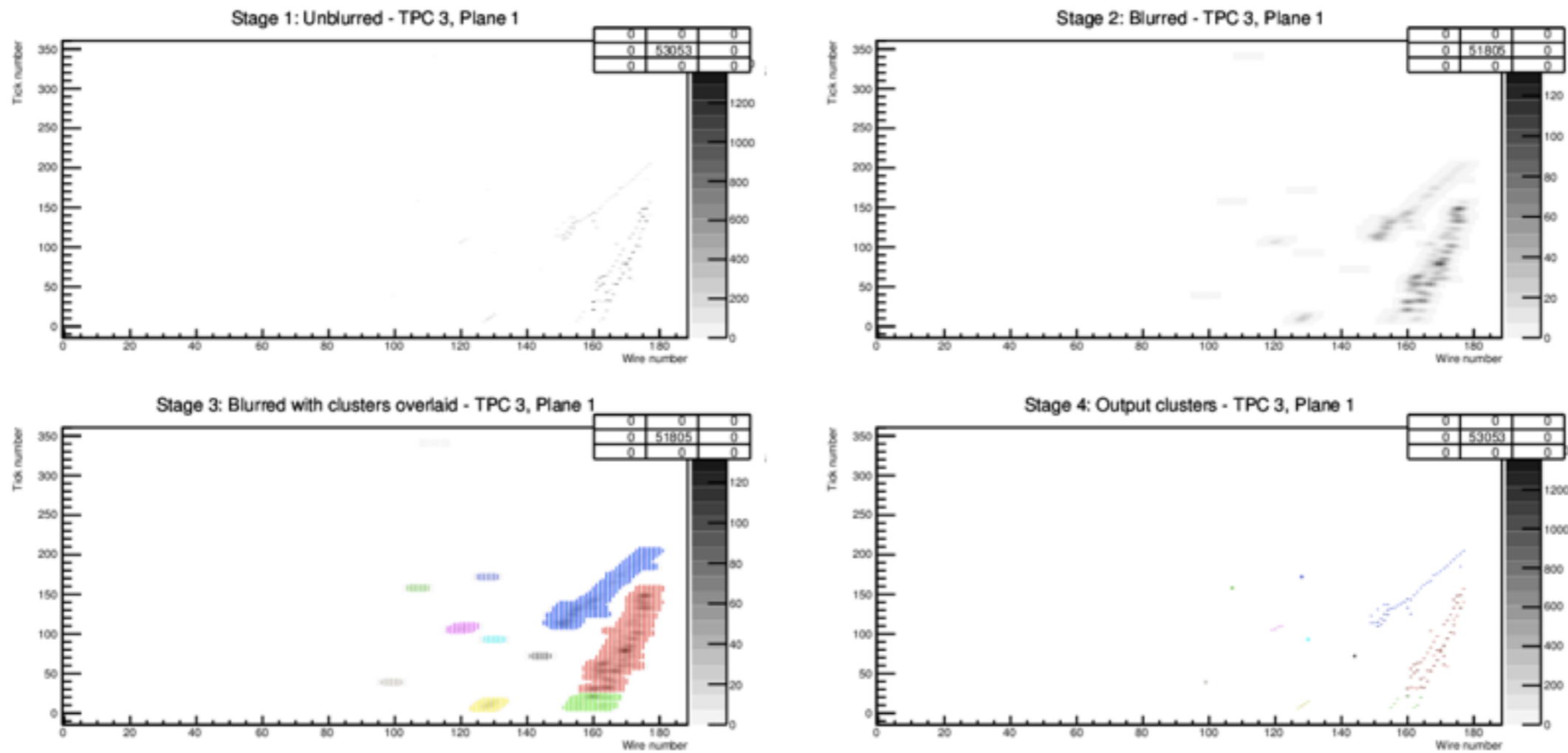
Comparing Blurred Clustering

Mike Wallbank

17/6/15

Reminder: Blurred Clustering

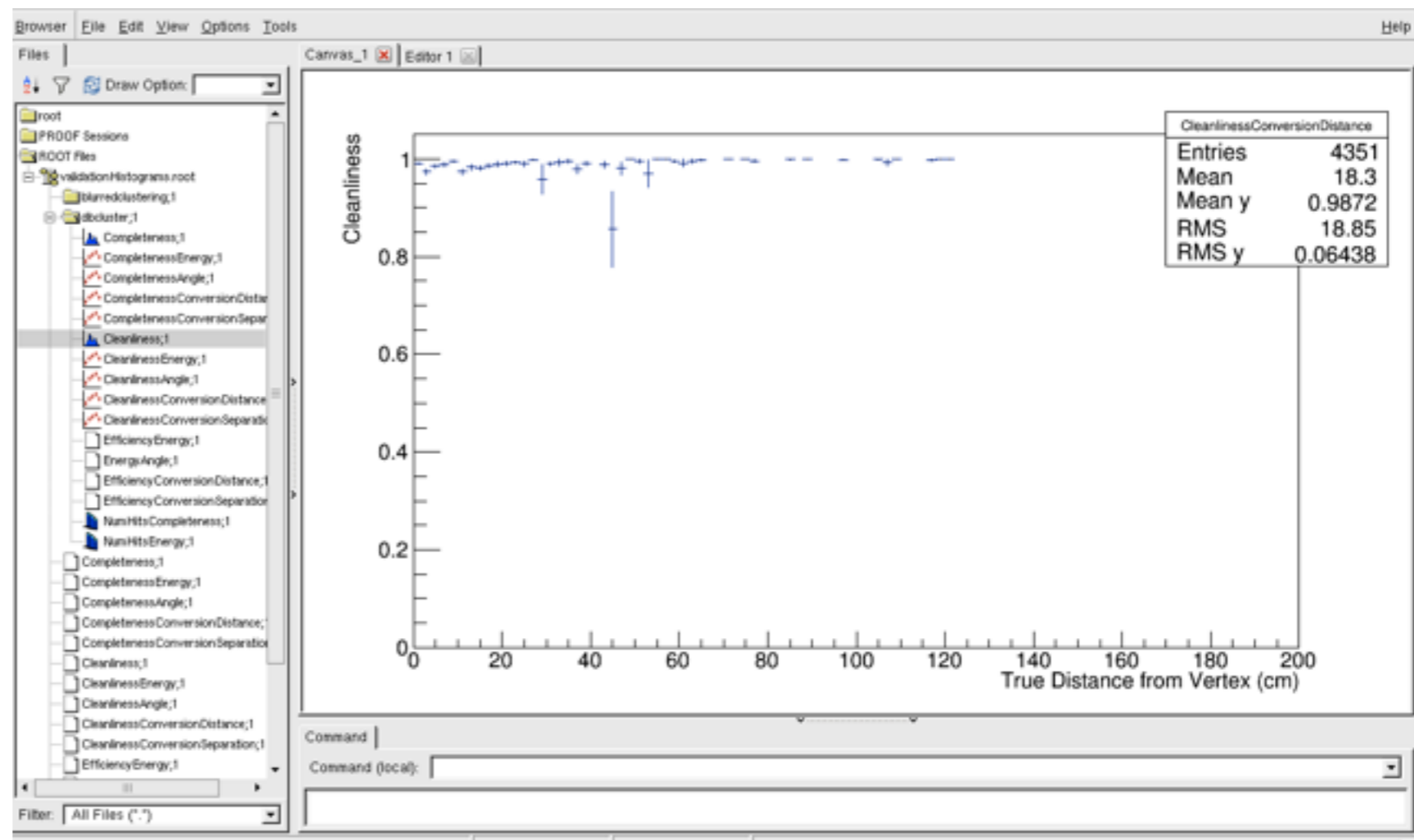
- Clustering technique which uses a Gaussian smearing to produce more full and complete clusters.



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

Validation Module

- Have written a clustering validation module to make the validation plots — tried to keep it as flexible and generic as possible so it can be used for any clustering.
- Specify the hit label and the clusterings to validate (vector) in the configuration file and the module creates the figures for each and also comparison plots:
- Lives in larreco/
ClusterFinder/
ClusteringValidation_
module.cc

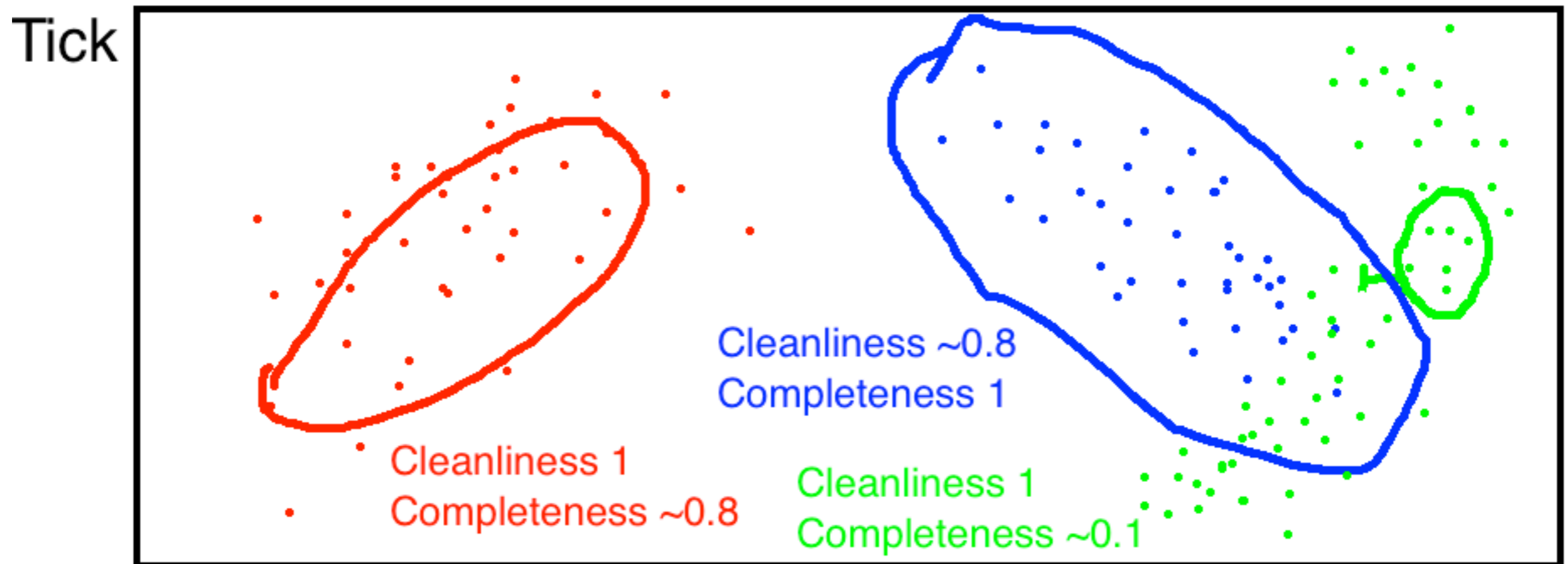


Clustering Properties

- 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);
 $\text{quantity(passes cut)} / \text{quantity(all)}$

Clustering Properties

- Wasn't going to show this again but have actually found myself using it to help visualise the clustering...



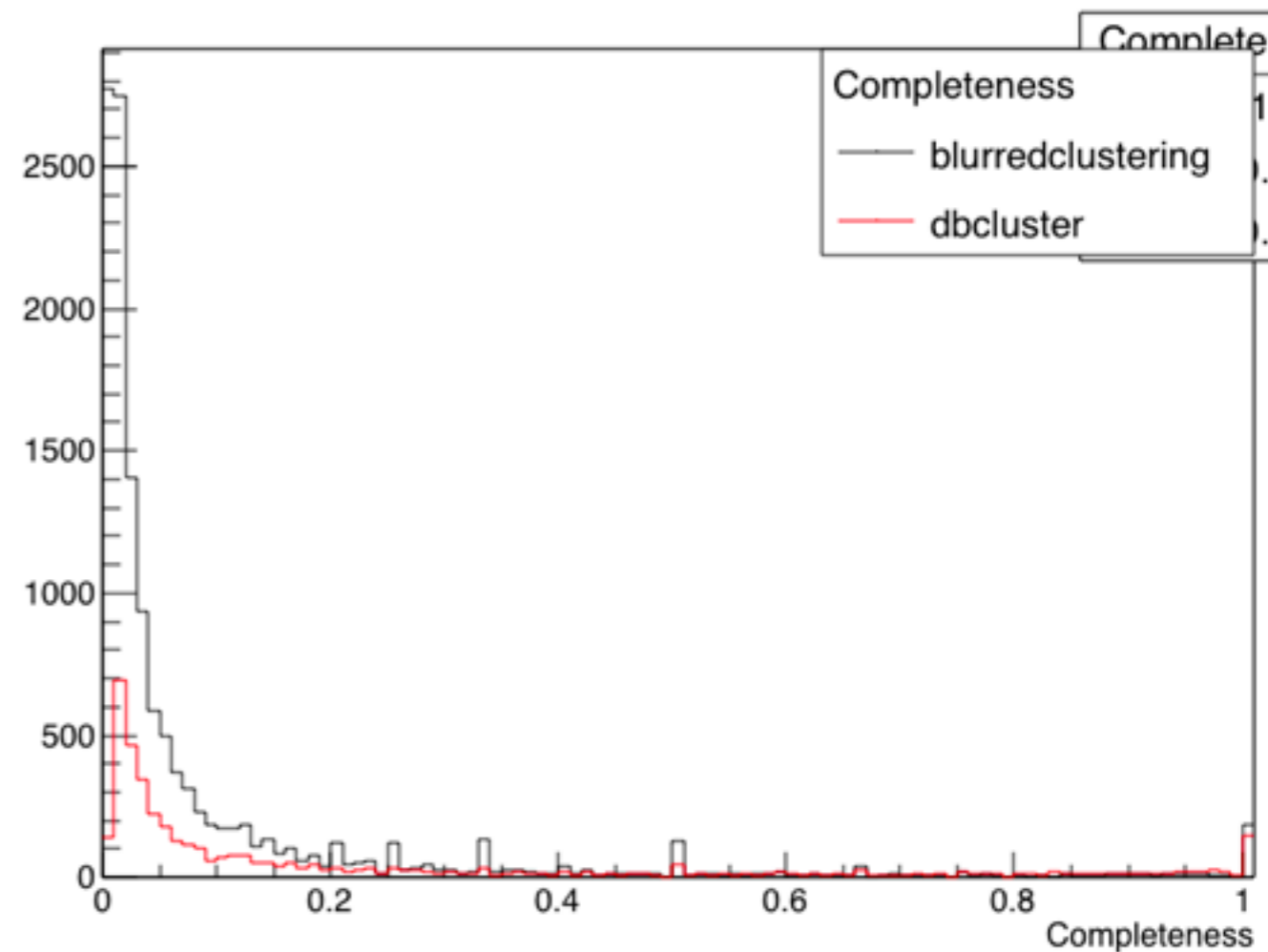
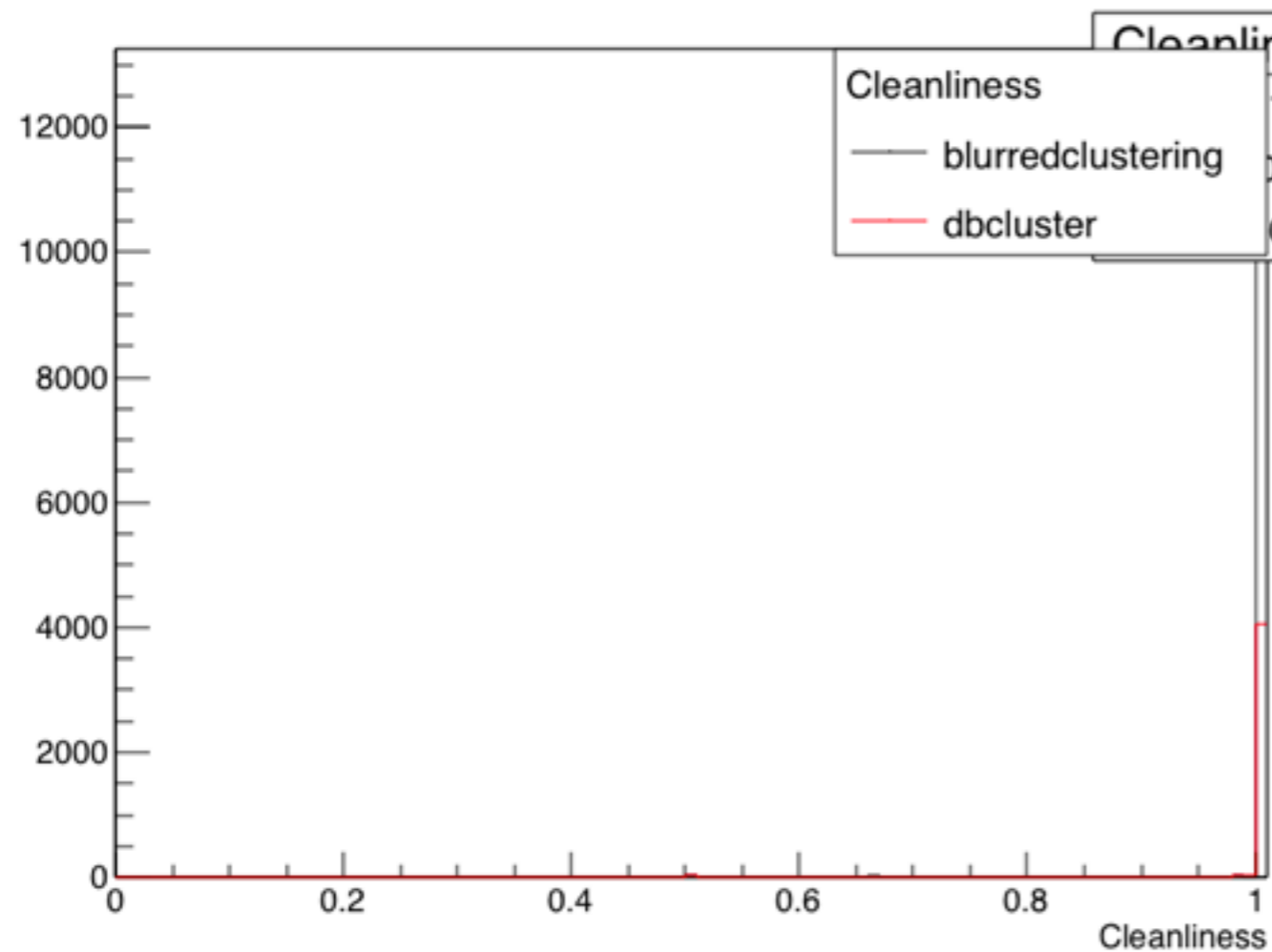
Would pass the efficiency cut!

Wire

Comparing Blurred & DB

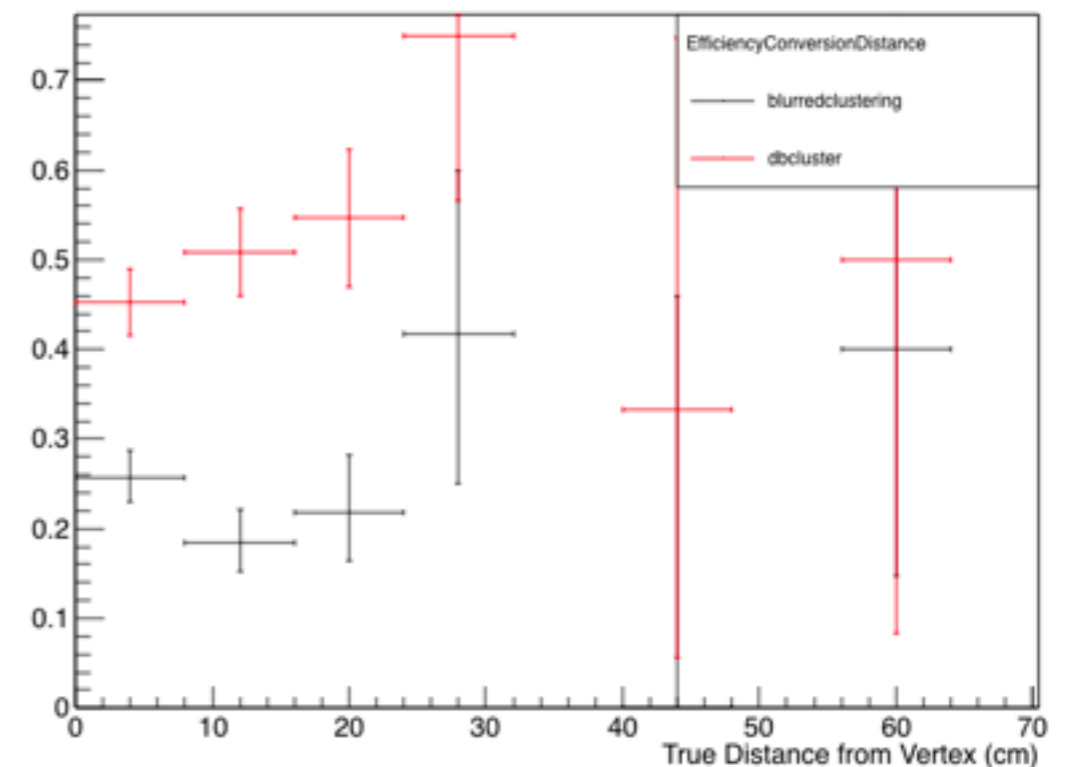
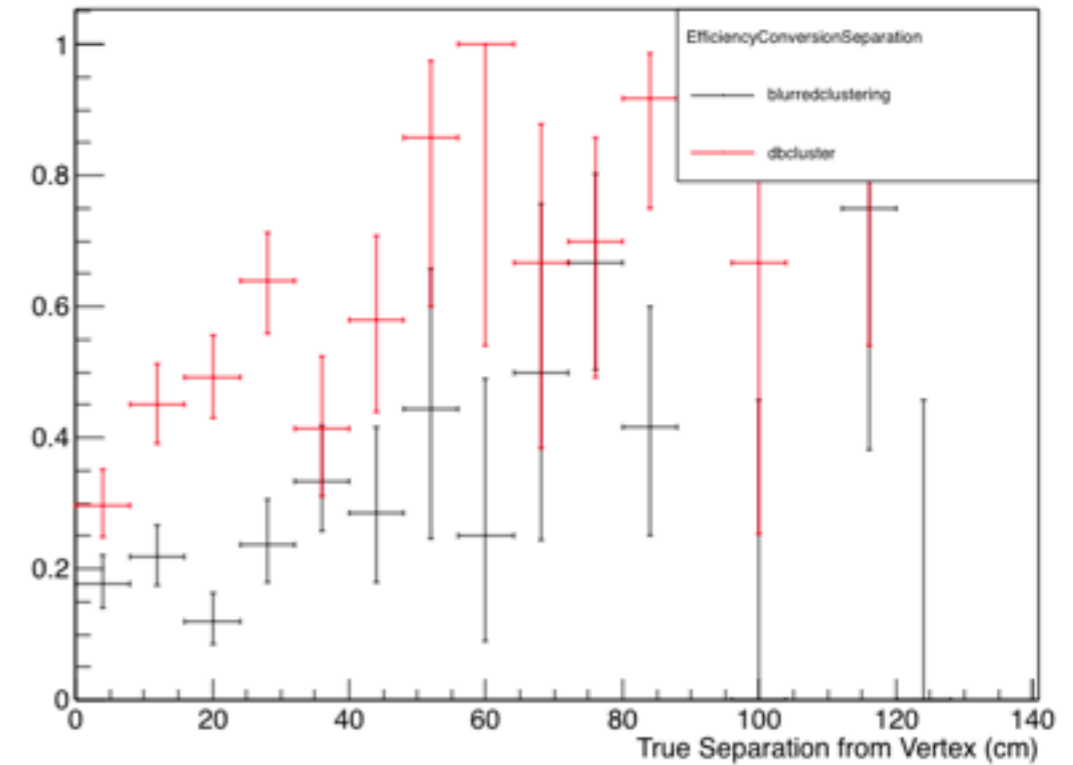
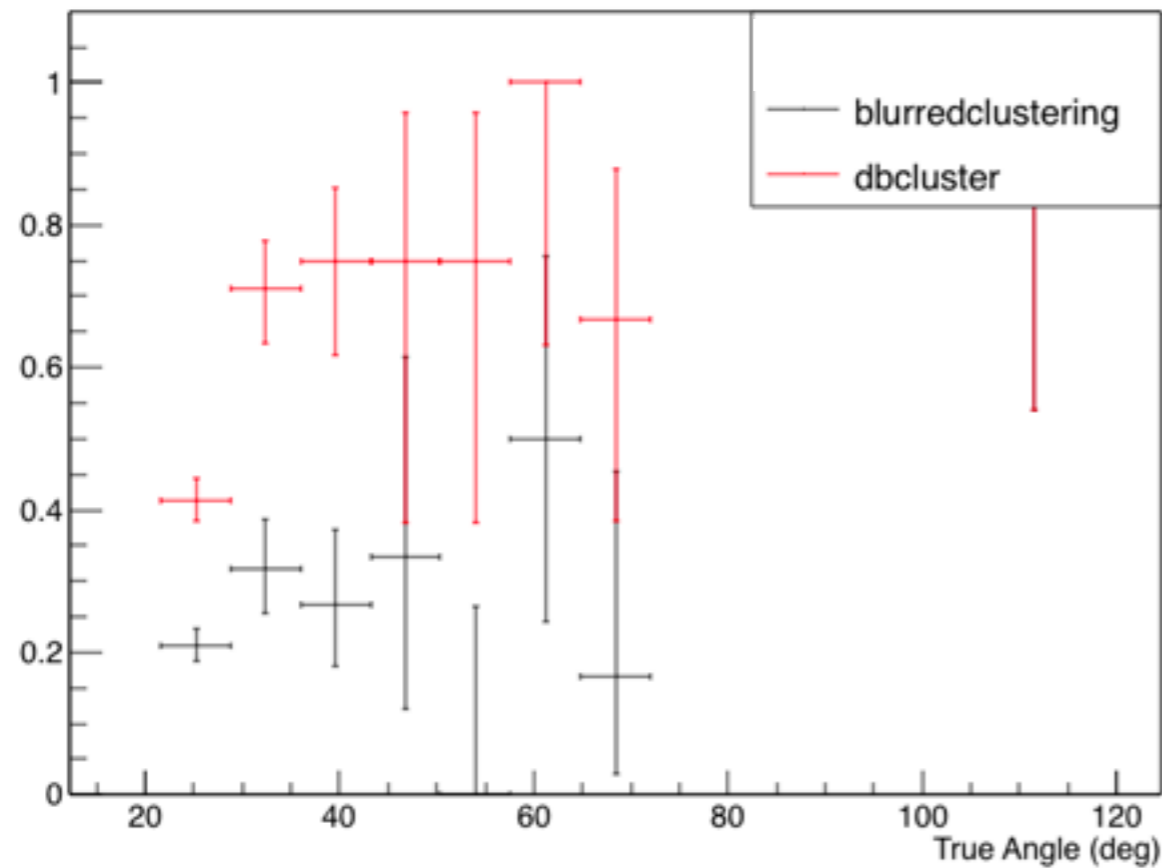
- Using the validation module, comparisons have been made between Blurred Clustering and the current clustering benchmark, DBCluster.
- Compared many different properties to get an idea of which performs better (spoiler: DBCluster right now is much better!)
- Some examples:

Cleanliness/Completeness



- Both very clean (blurred makes many many more clusters from same sample though).
- DBCluster makes much more complete clusters.

Efficiencies



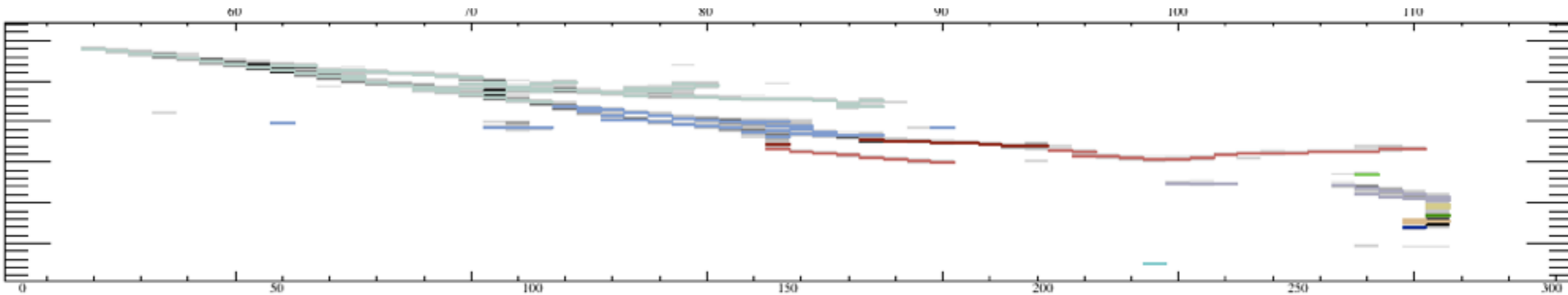
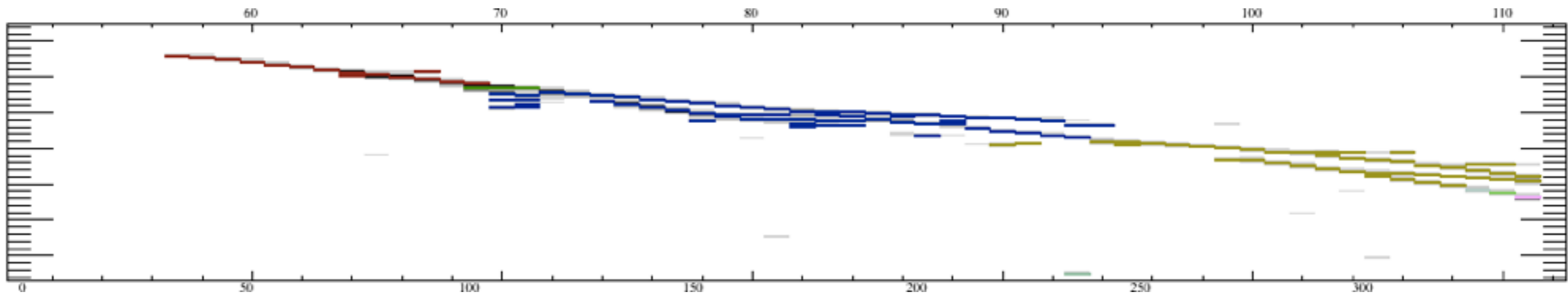
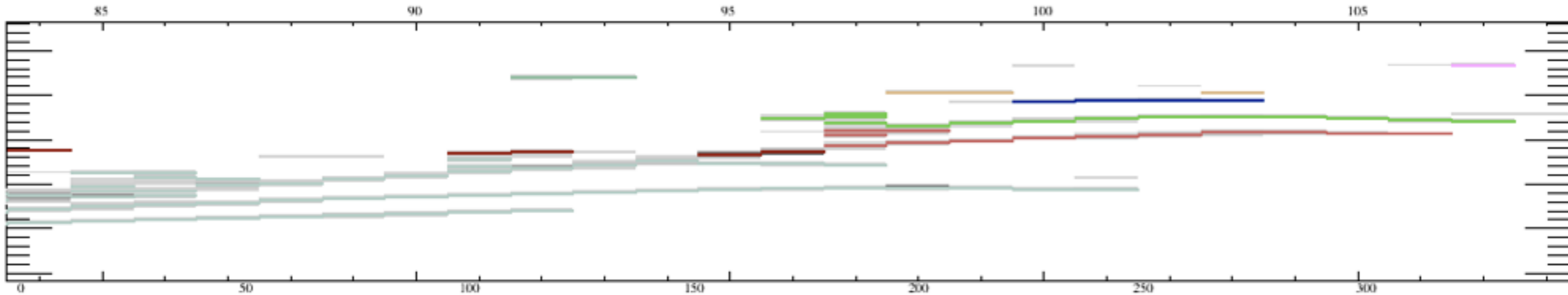
DBCluster is much more efficient!

Blurred gets to around ~50% but DB always performs better

Problems!...

- Robert and Dorota noticed some strange issues when using the Blurred Clustering.
- In general the clustering was ok and made tracks / showers as expected, but there are many examples which show strange behaviour!
- Demonstrated on next slide...

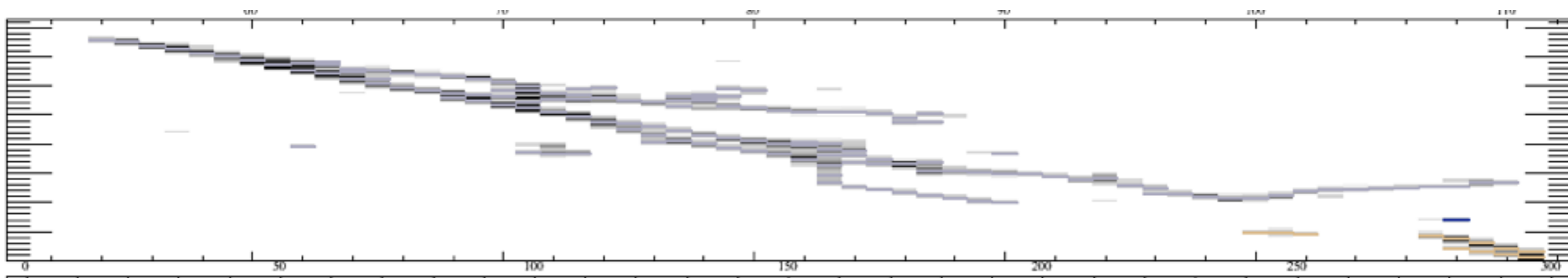
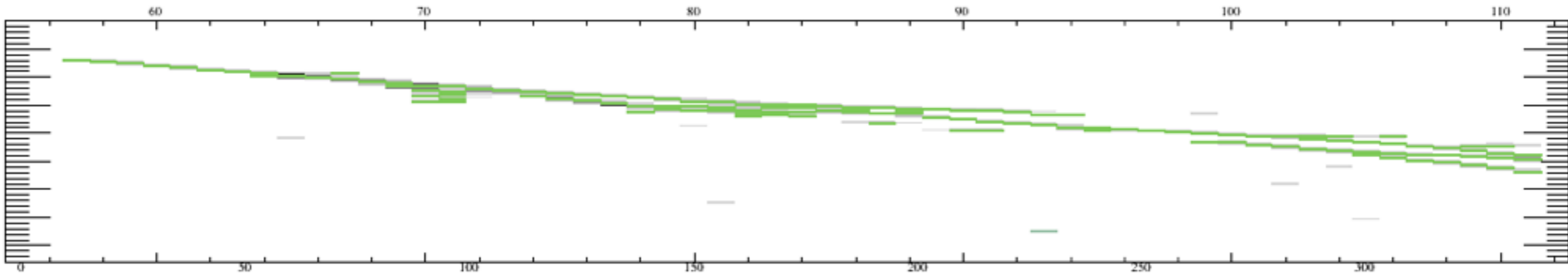
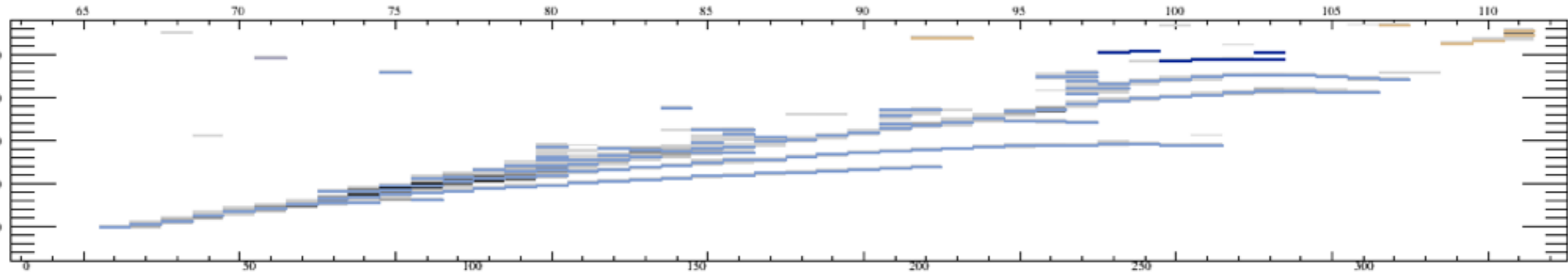
Problems



Fixes?

- There are many free parameters in the clustering and *still* trying to get a handle on them!
- Will talk about this more next week.
- Most likely this problem is due to the method not yet being fully optimised — I will only accept DBCluster is superior after it's been properly looked at!
- Right now I've trying clustering further distances from the seed and this works much better...

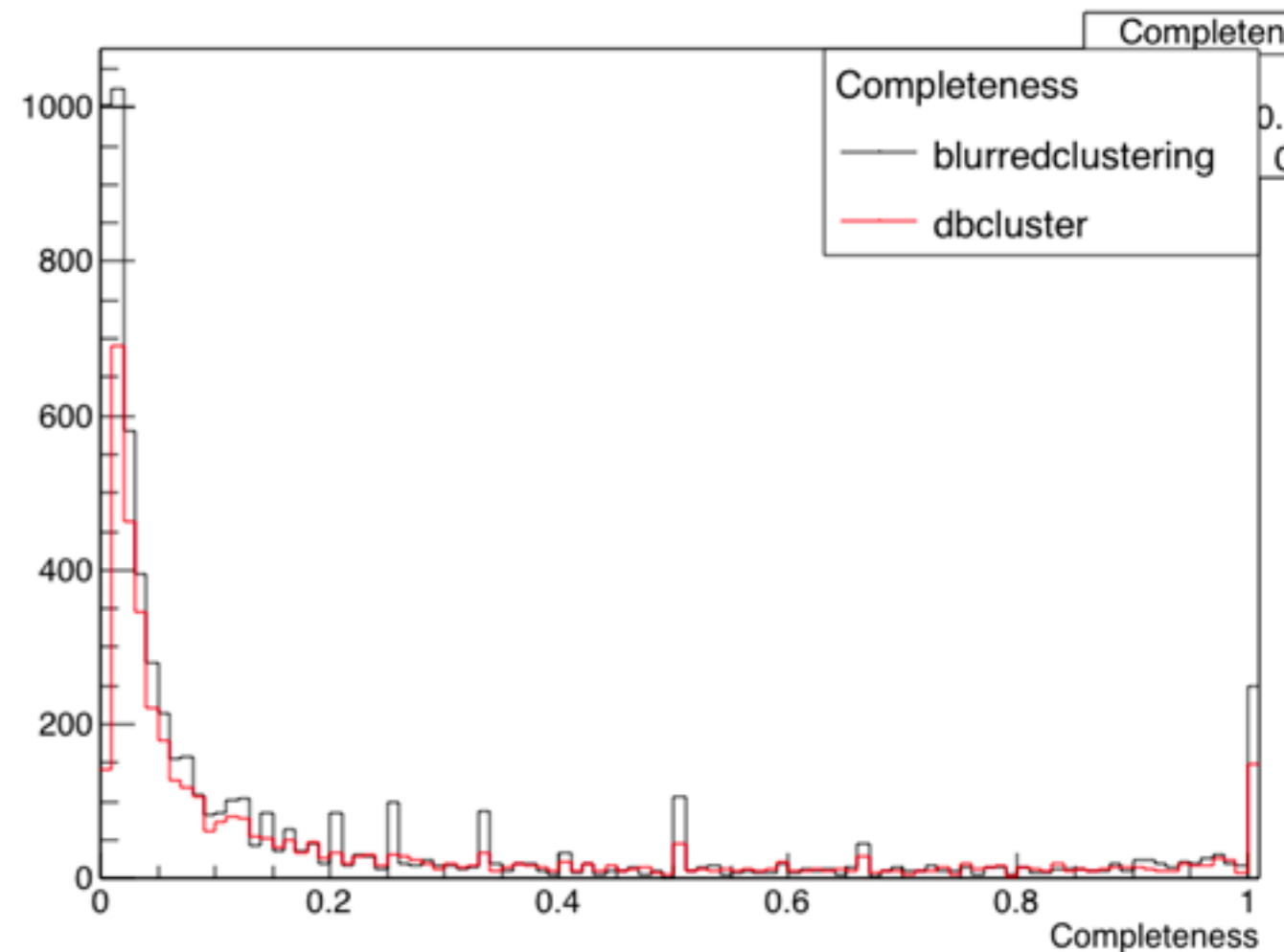
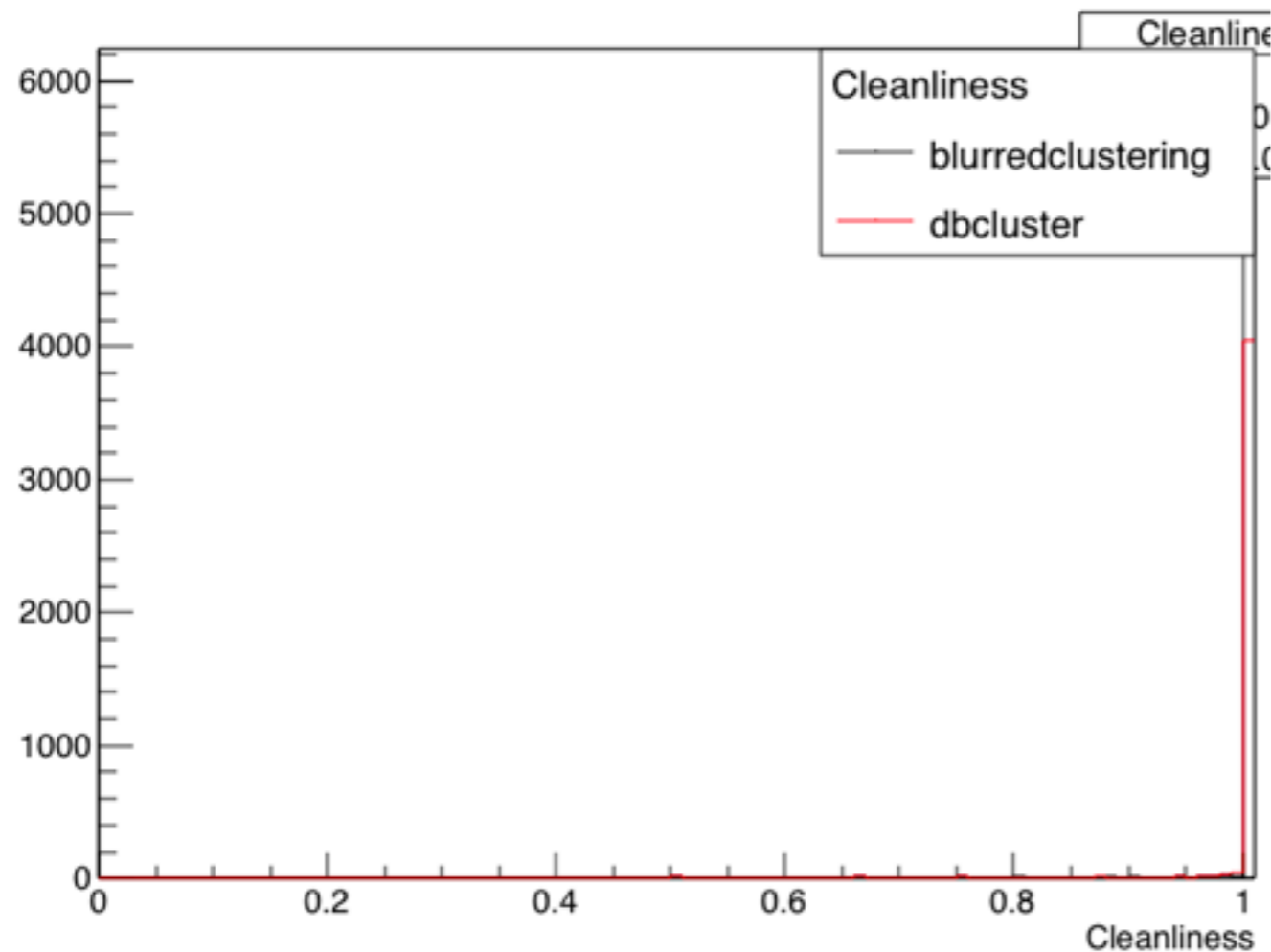
Fixed problem!



Tuning

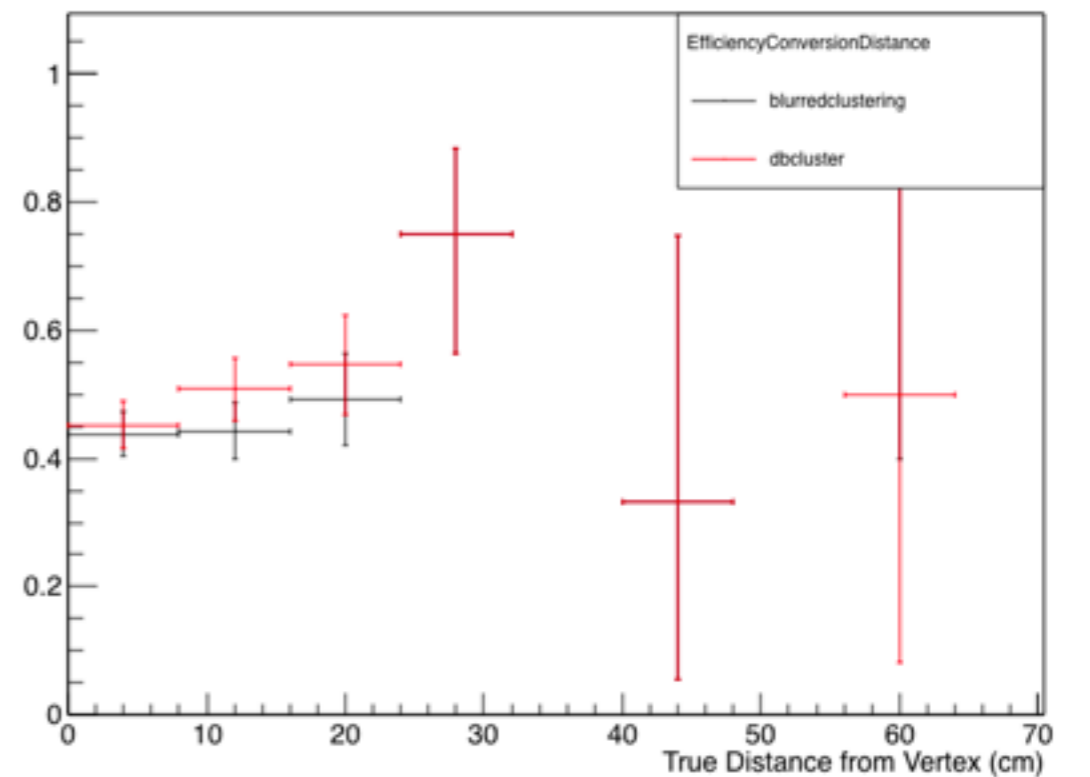
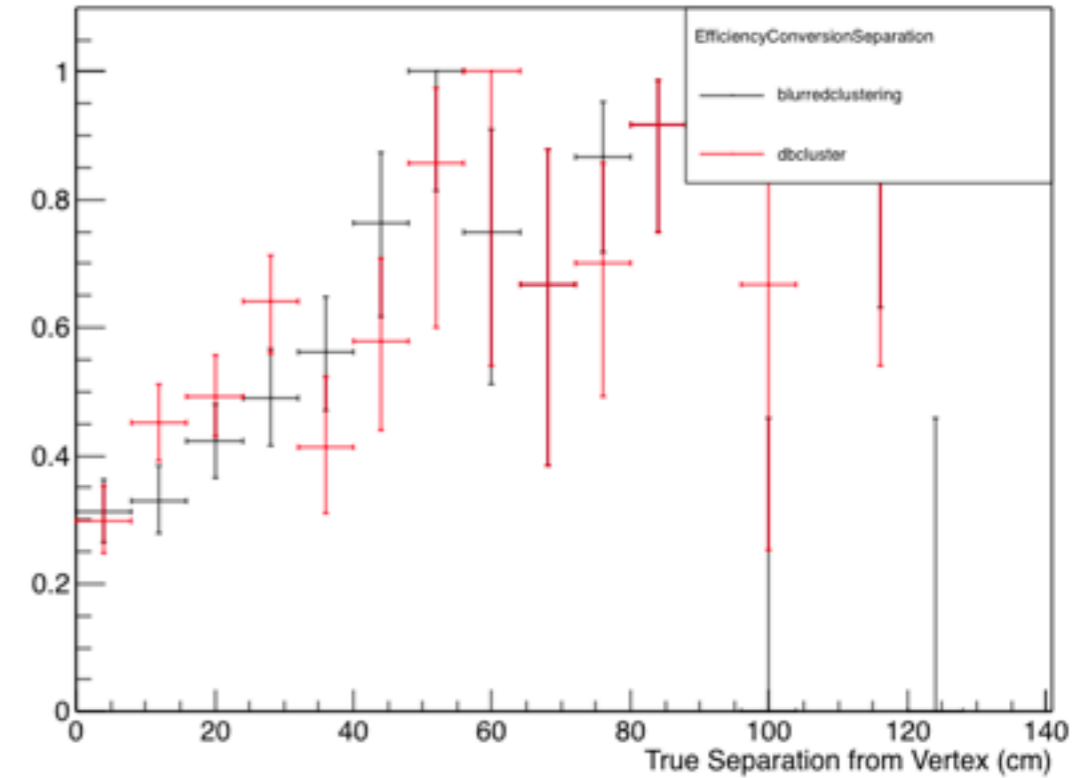
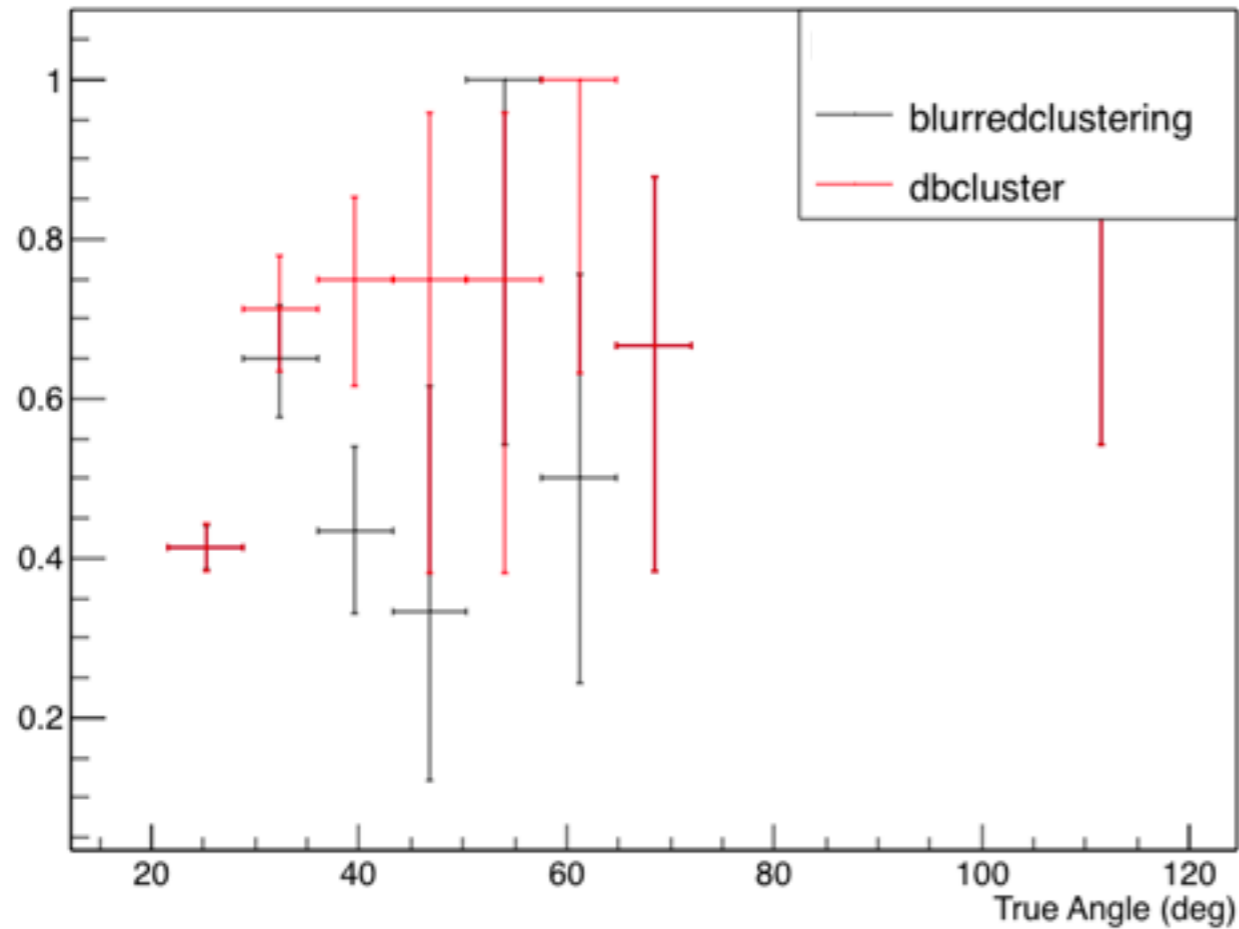
- This demonstrates how much this algorithm needs to be optimised to work properly in LAr.
- The free parameters make a huge difference to the effect of the clustering and so this needs to be seriously looked at!
 - Going to do this properly this week.
- This clustering *could* be a very useful clustering method in LArSoft once it is fully understood and the parameters tuned correctly!

Improved Completeness/Cleanliness



- Still very clean.
- Much much better completeness! (Actually outperforms DBCluster for fully complete clusters)

Improved Efficiencies



- Now very comparable to DBCluster.

Summary

- Have the framework in place to validate and compare different clustering methods.
- Whilst fixing a problem with the clustering, realised how little I actually know what's happening!
- Will spend a lot of time this week understanding everything more and improving the clustering for use in LArSoft.
- Showing more promise than last time!