Improving the 3D spacepoint outputs from Pandora

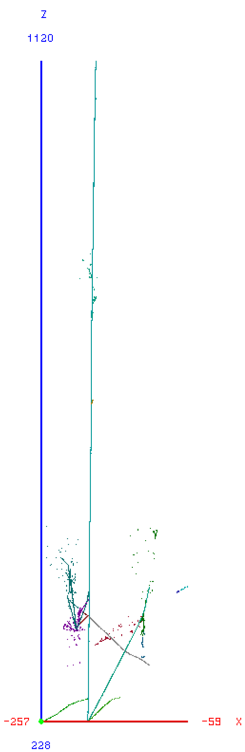
Ryan Cross

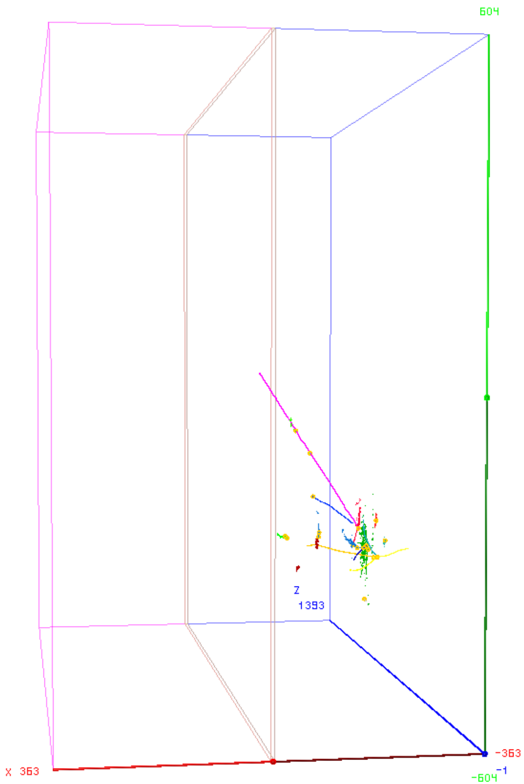
Lancaster University - 2020/01/13

What is 3D Reconstruction?

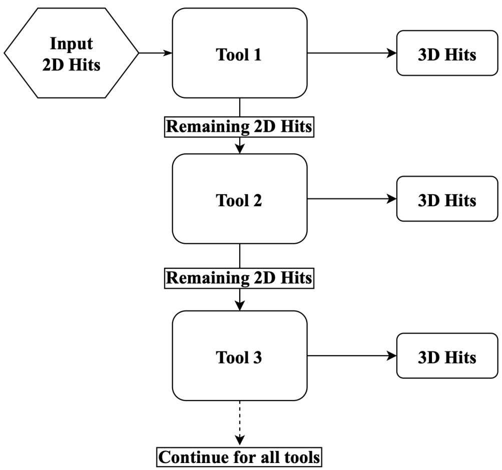
3D reconstruction, in this context, is taking the 2D particles that Pandora produces and using the hit information from each of the available views to produce a 3D reconstructed hit.

This applies to both tracks and showers, though the specifics of the implementations differ.





Current Implementation Details



The current Pandora 3D spacepoint creation works in a "waterfall" style model.

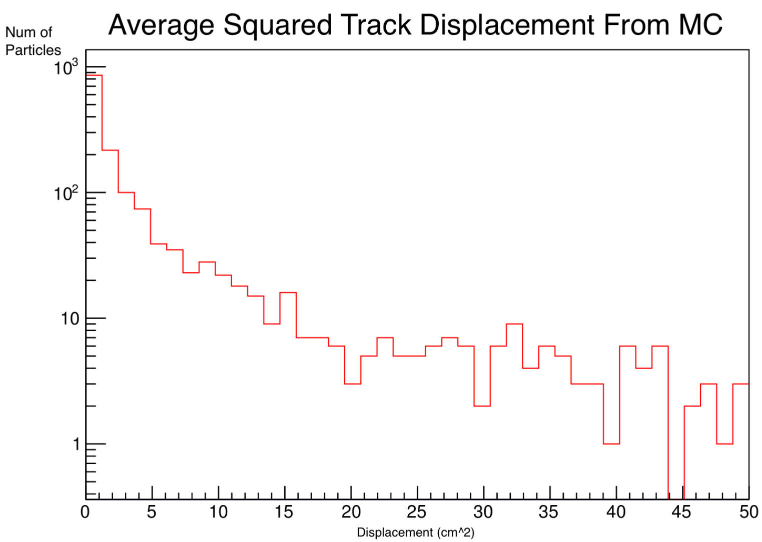
All of the 2D hits for the current particle are passed into the first tool. This tool then produces 3D hits for all of the 2D hits that it can, and associates these with the 2D hit it was made from. Any 2D hits that have no associated 3D hit, are then instead passed to the second tool to repeat the process, until all hits have a 3D hit, or there is no more tools left.

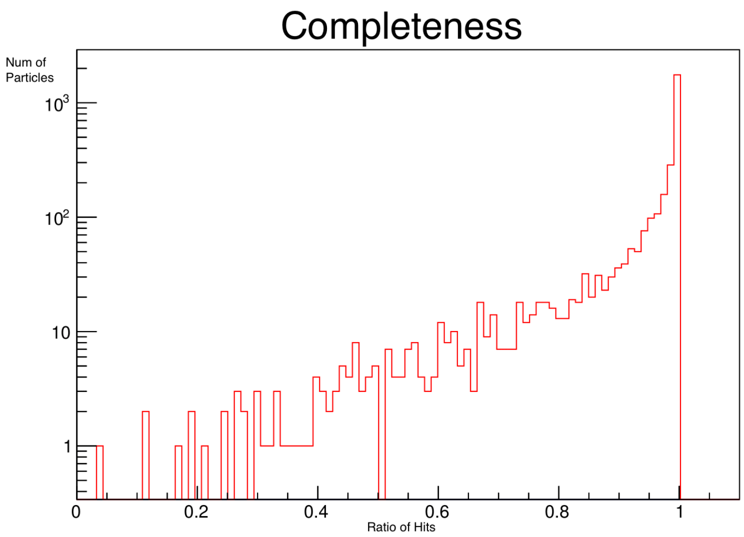
For the most part this works well enough, but there are cases where we use a tool that is inappropriate first, such that a better suited tool may never see the full event. That is, a track travelling the full length of the detector could be reconstructed first with a tool that is optimised for tracks going upwards in the detector, which could produce sub-par hits, and means those hits can not be used in the later tool(s).

Current Performance

The current Pandora performance for 3D hit reconstruction is doing very well. For the vast majority of events, we are able to produce events that are reconstructed well in both 2D and in 3D.

Most 3D hits are within a cm or so of their real position.



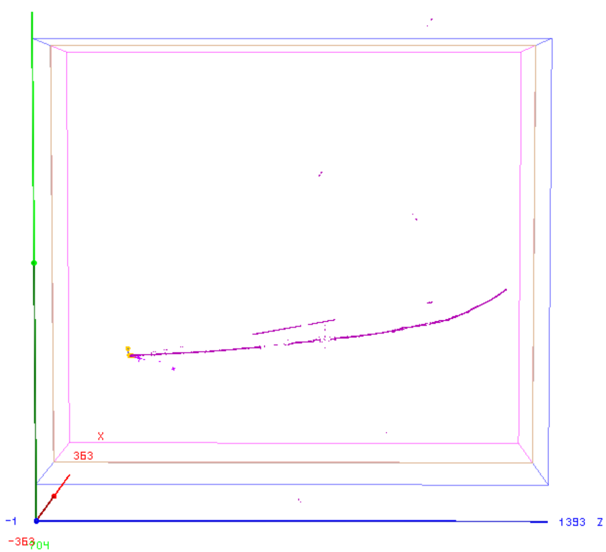


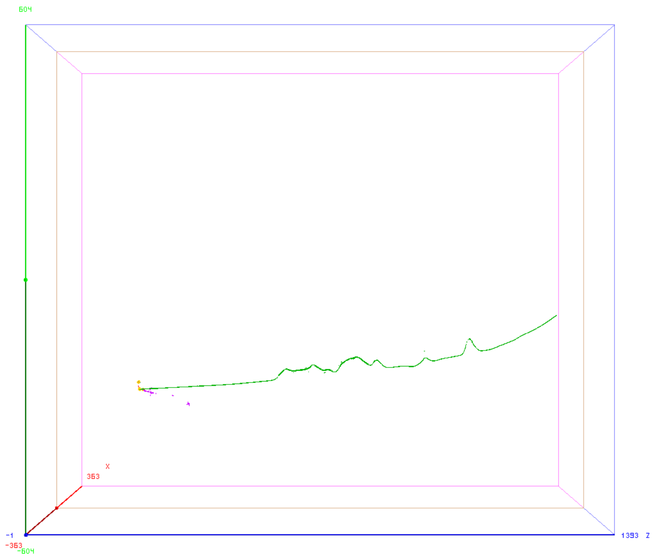
Current Issues

However, when the 3D hit creation fails, it can fail fairly dramatically. Some of the failures are simple to restrict, by using more detector knowledge, and others issues are caused by the 3D reconstruction pipeline.

The issues that need addressing (in no particular order):

- Detector geometry is not considered when making 3D hits.
- Different tool outputs can be combined, creating disjoint trajectories.
- Compounding of issues by smoothing.
- Addition of more sophisticated tools for 3D hit production.

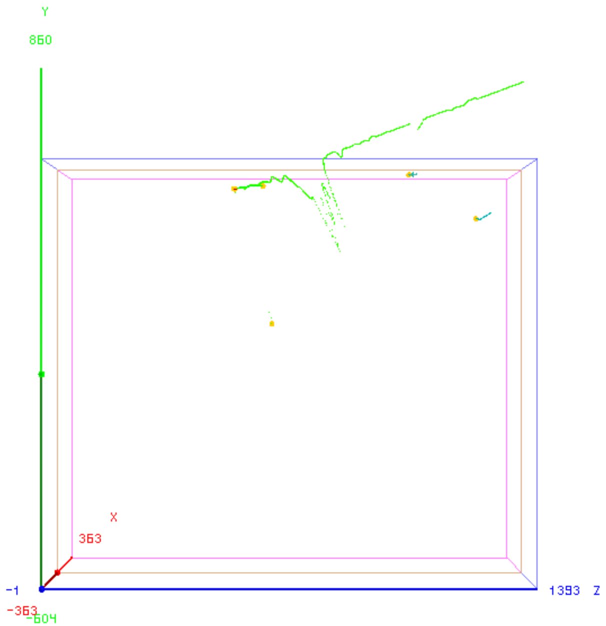


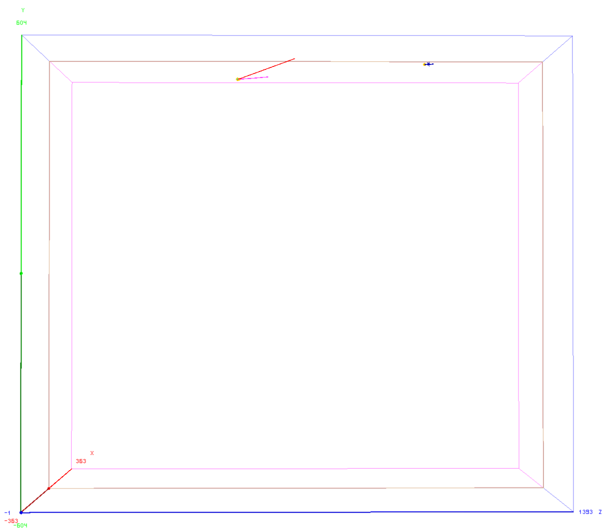


Detector Geometry Considerations

The first small change to be made was to ensure that the detector geometry was being considered during the hit creation, such that hits that fall outside the detector are made much less likely to occur.

This was achieved using a chi-squared term being added to each of the hits, such that these hits were made less likely, and a cut ensured that no hits got through that were out of the detector by a large margin.



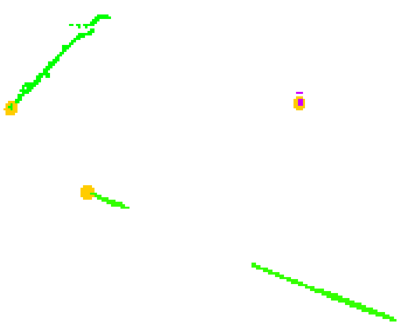


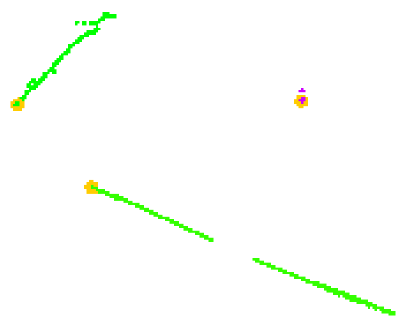
3D Hit Interpolation

A second change is the addition of 3D hit interpolation. This works by considering the 2D hits that fail to get an associated 3D hit, and attempting to project them onto a fit generated with the existing 3D hits.

This allows gaps that would have previously have been filled with a second or third tools to be filled with hits that are consistent with the existing hits, rather than hits that may not have been consistent.

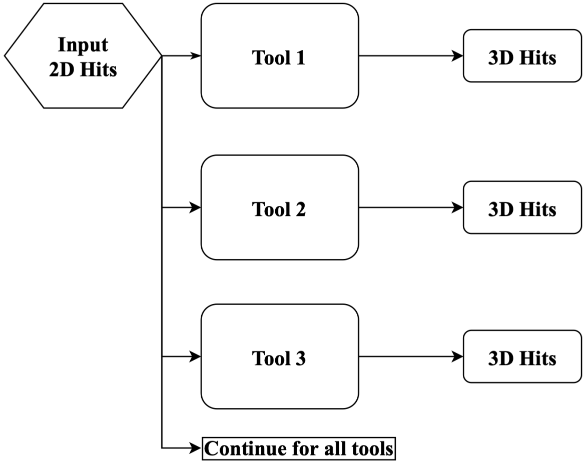
Being 2D hit first (i.e. starting with the hits missing a 3D hit) is both a good and bad thing. It means that we shouldn't interpolate over areas where there genuinely are no hits due to dead wires or similar, but does mean that if a hit has been used incorrectly, this will not fix it. (Fixing broken hits also isn't really the use case of interpolation either so that is okay...).





Pipeline Changes

The most recent change is changing the pipeline that Pandora uses to actually produce these hits.



That is, update the algorithm pipeline to allow all the reconstruction tools to produce their version of a full 3D reconstructed output, rather than just based on the subset of hits they see.

This has a few advantages over the previous approach. Firstly, in the old architecture, it was possible for a particle that was best suited to be reconstructed with tool 2 to be reconstructed with tool 1 first, such that tool 2 never got the chance to produce a full event output, which could have been better.

Secondly, when combined with interpolation, it means that a single tool can output a full output, rather than just a subset, meaning there is a larger number of available hits.

This adds an additional step at the end of selecting between the produced hits to get the most complete set of coherent hits.

Challenges this introduces

This means that an additional challenge has been added, where output hits must be picked between, but the hope is that it should allow for better 3D reconstruction overall.

The decision to be made is as follows:

- Given a choice of sets of 3D hits (that have been interpolated and smoothed, i.e. they are the best version that a single tool can produce), which hits are the best and should be used as the 3D hits for a given particle?

The difference between two outputs could be as dramatic as one failing to reproduce most/any hits, or more subtle issues, where the reconstruction works, but produces a worse, less realistic output.

How to get a single output

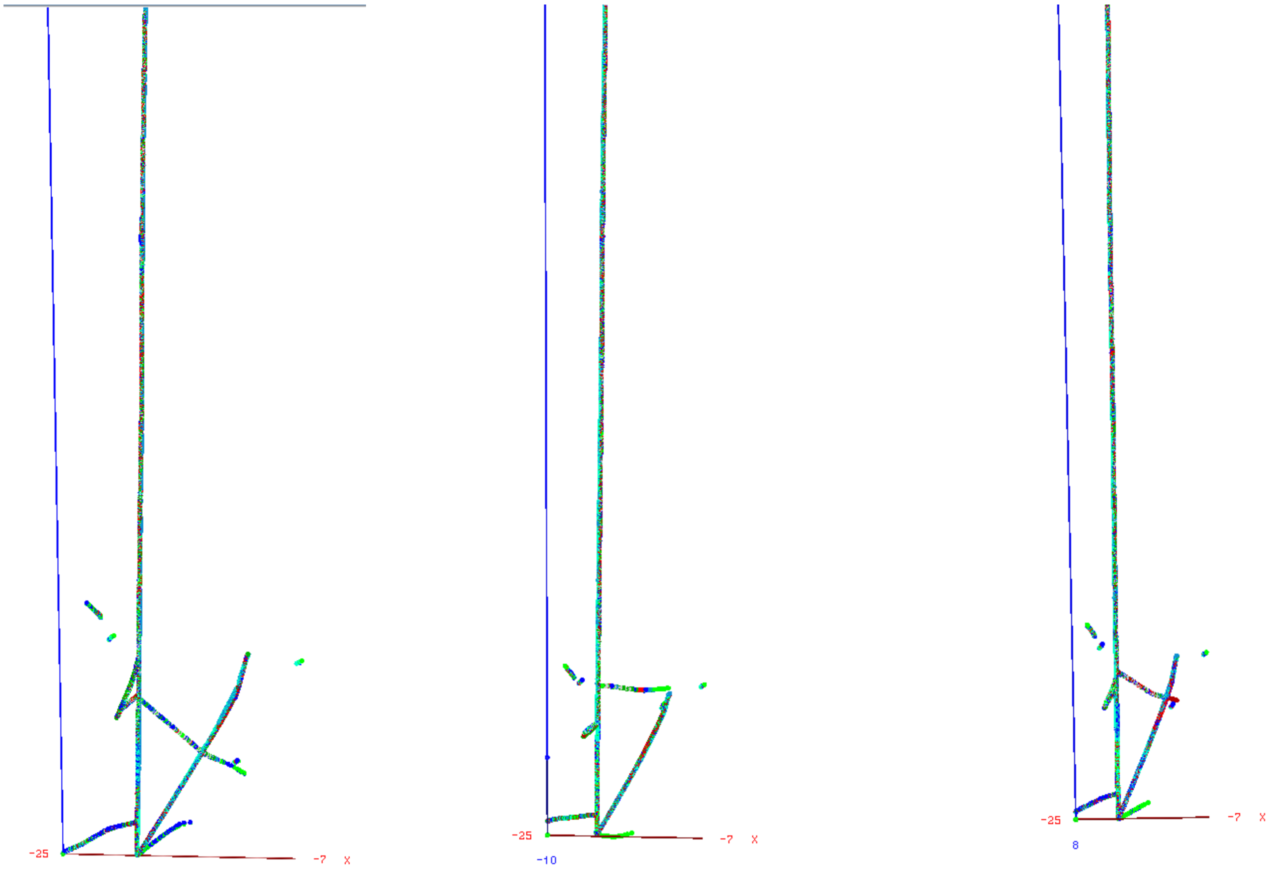
Two different options are being considered to pick between the outputs.

- Re-use the existing metrics (from the original performance check) to profile each output. The results could be taken directly, or used in an MVA. This way would then take a full single output.
- Find the coherent single trajectory through the combined set of all hits. This assumes that over all the algorithms, the correct hits have been generated, and just need to be selected out of the full set. This would then take individual hits and combine them.

Each method has its own merits, and will be checked to see how it performs. Attempting to find the coherent trajectory through the full set of all 3D hits should perform better, though could be more susceptible to the smoothing issues seen earlier.

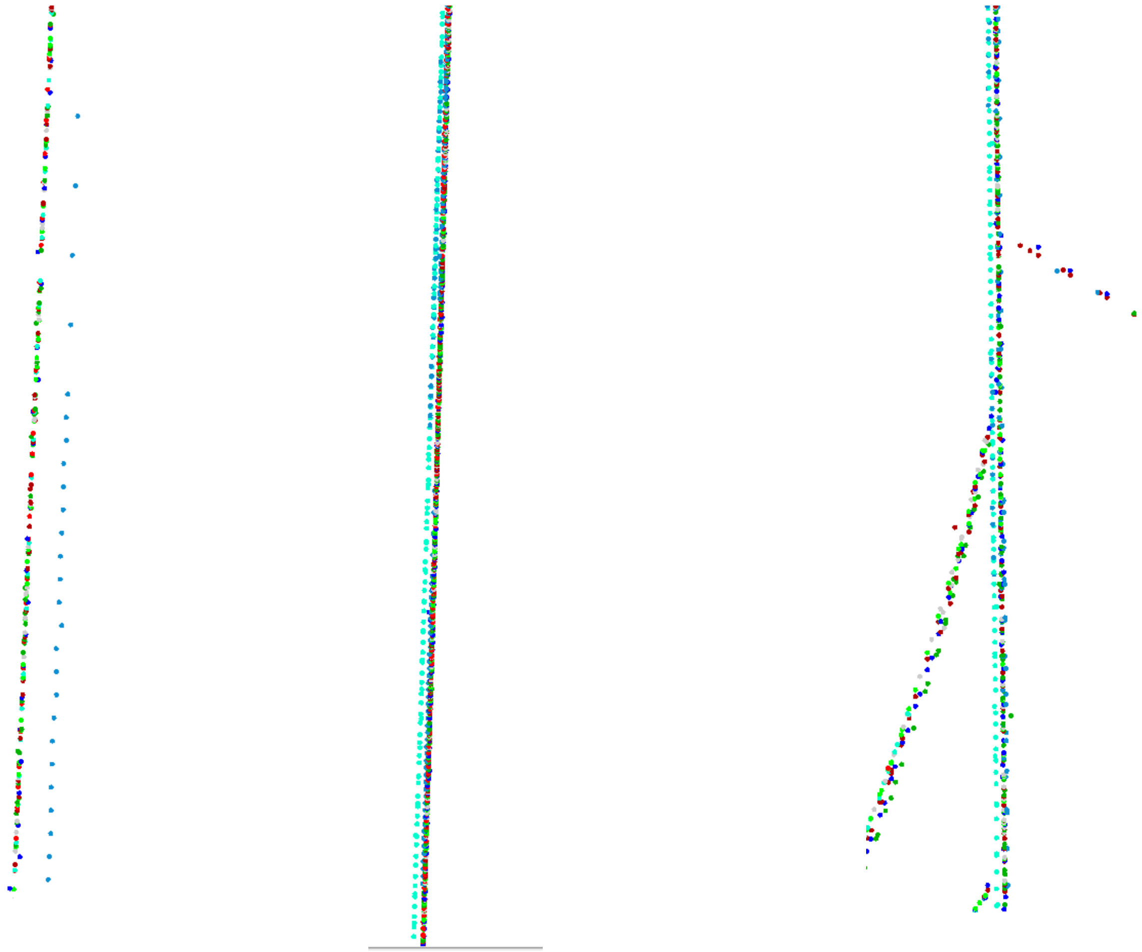
Further Techniques

Further techniques are also being added to include in either approach, such as utilising projections of the 3D hits back into 2D, to help highlight outliers.



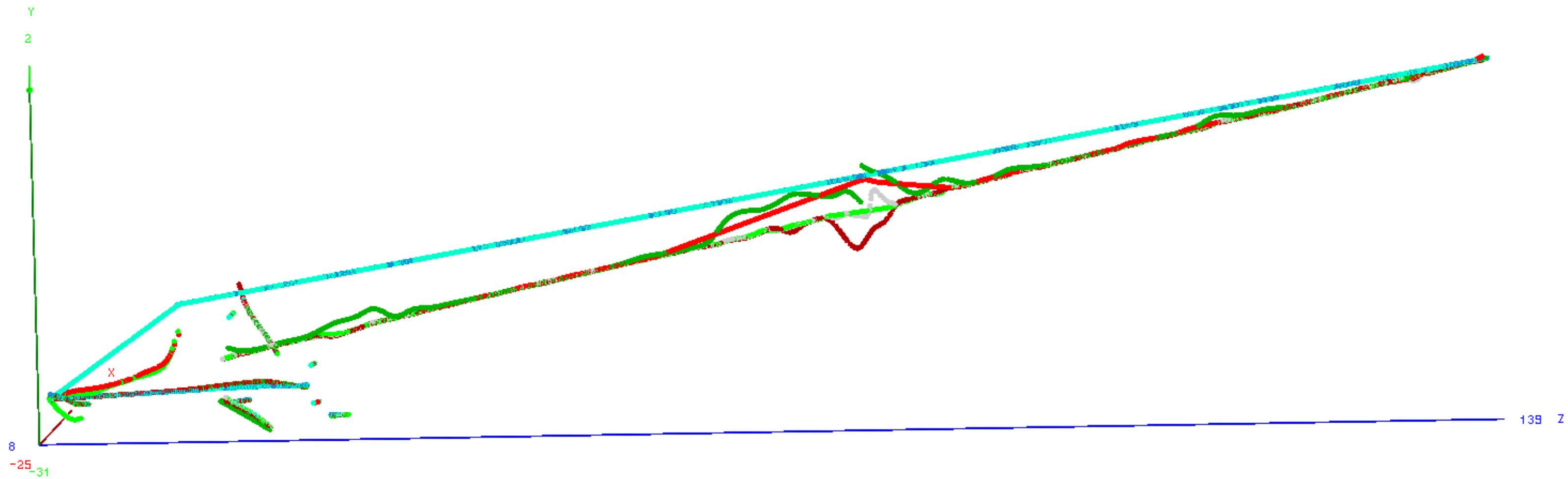
Further Techniques

Each colour is a different output. Dark blue is the original calo hits.



Further Techniques

We can also look at the same outputs superimposed in 3D, to get an idea of what is happening and see how we could help remove further outliers:



It is then very obvious why the cyan and blue points look so bad projected back into 2D, as they are very far away from the rest of the points. Combining information from both 3D hits and their projections gives us more options for picking out the correct hits.

Pulling the correct hits out

The current plan for pulling the correct hits out of the superimposed "hit cloud" is as follows:

- Remove the obvious outliers (this could be done in 2D, 3D or both).
- Over this reduced set of hits, keep the hits that look the most consistent with the other hits (again, this could be done by projecting to 2D, comparing in 3D, or both).
- With this final set of sensible, consistent hits, take 1 spacepoint per calo hit. This can be done using the chi-squared term that is calculated at the spacepoint creation to take the best hit, or to take some form of average over all the available hits.

There also needs to be some experimentation around cuts that occur earlier in the process, such that more candidate hits could be produced to be considered.

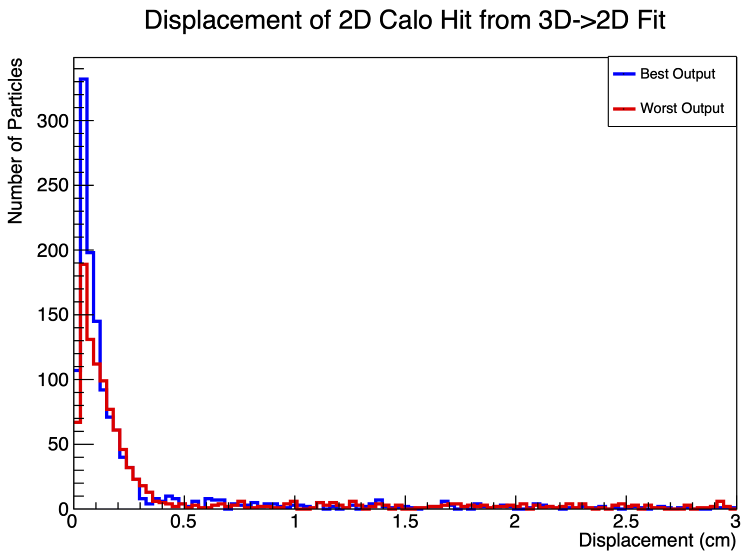
MVA options

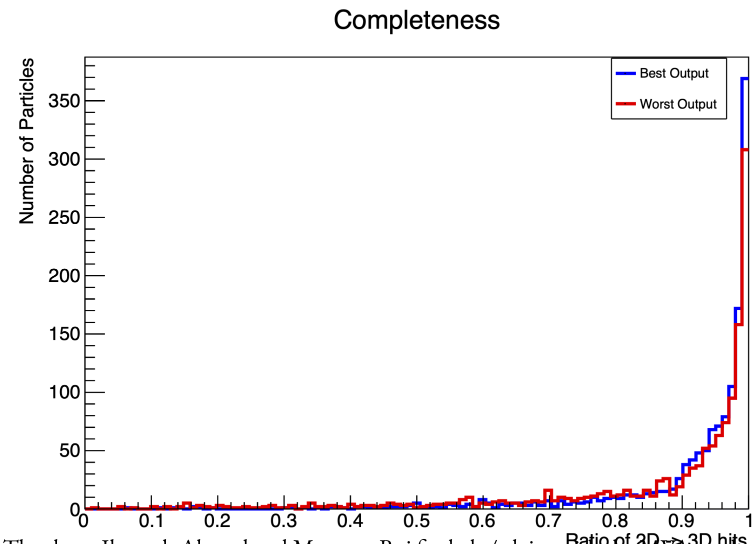
The other option is to use an MVA approach.

Metrics can be generated using the final interpolated and smoothed results of each tool.

This includes measures of completeness, "wiggle", number of interpolated hits, and measures of the previously shown 2D displacement.

Between these, a single output can be chosen, which would mean that issues of smoothing between tools would be ignored, though perhaps at the detriment of losing information from the other outputs.



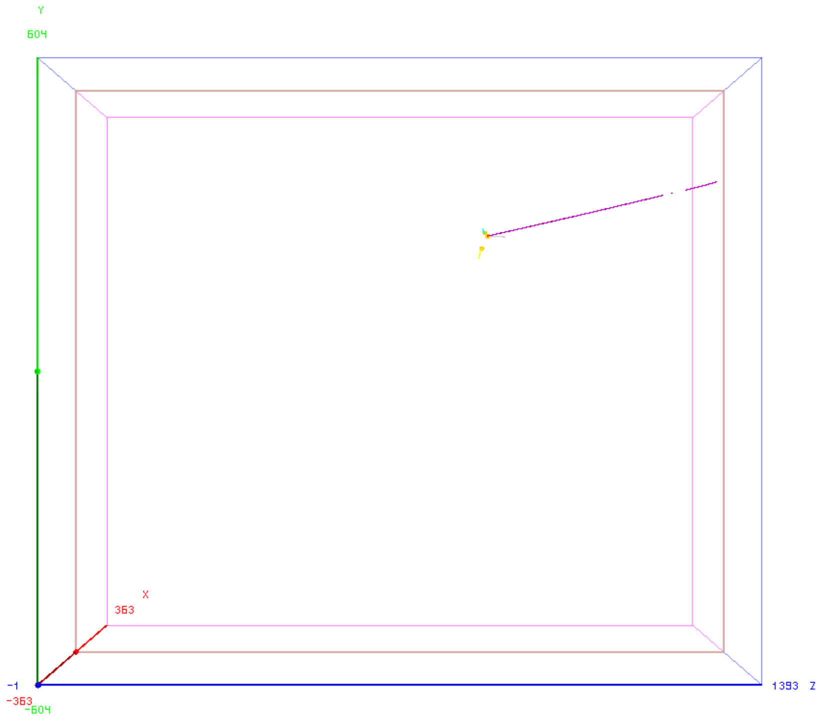


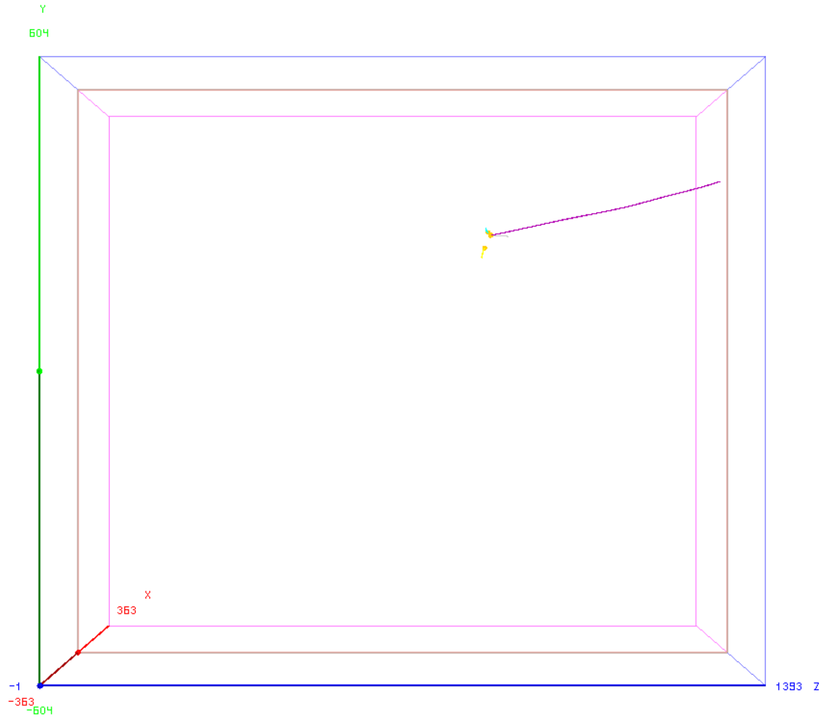
Thanks to Jhanzeb Ahmed and Mousam Rai for help/advice with the MVA work.

Current Work

Both methods are going to be considered, with the aim of producing the most consistent, realistic output that is possible, given the input hits.

I'm focusing now on the coherent hits method, and will come back to a metric based method after.





Upcoming Work

To move on from here, there are a few steps, after finishing the on-going work.

- Finalising 3D Reconstruction improvements for tracks
  - Add new more sophisticated tools
  - Use more physical knowledge in the reconstruction

- Extending work to 3D reconstruction for showers
  - This is a different tool set, but the base is the same as the track algorithms, so benefits from the changes there.
- Changes required for Deep Learning integration
  - Could be used to implement some reconstruction changes.
- Integration of Deep Learning in existing tools/algorithms
  - Anywhere a decision is made, or images are used could be updated.
- Use real data from protoDUNE for performance checks
  - Profile performance on real data.

Questions?

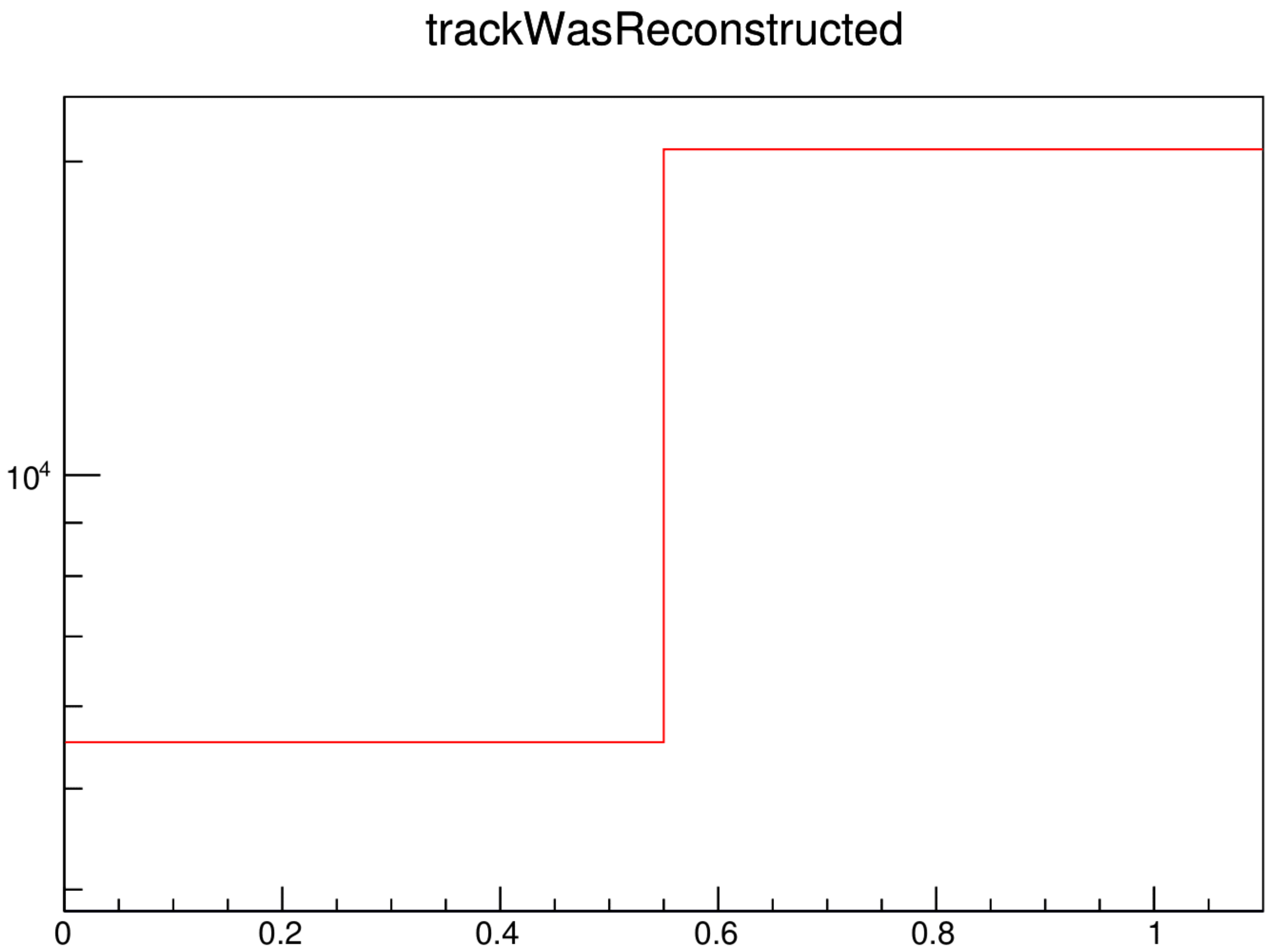
Backup Slides

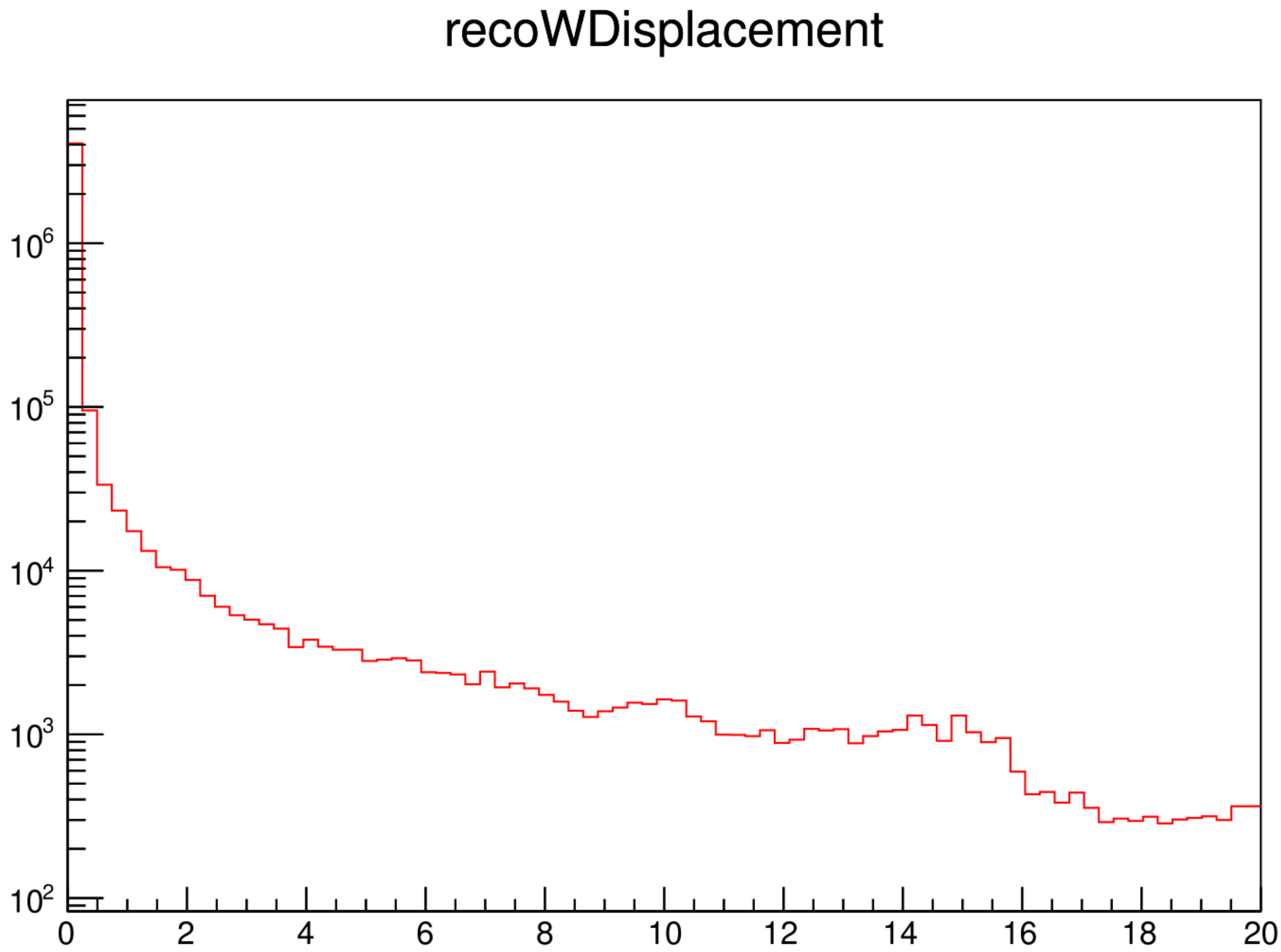
BDT Variables

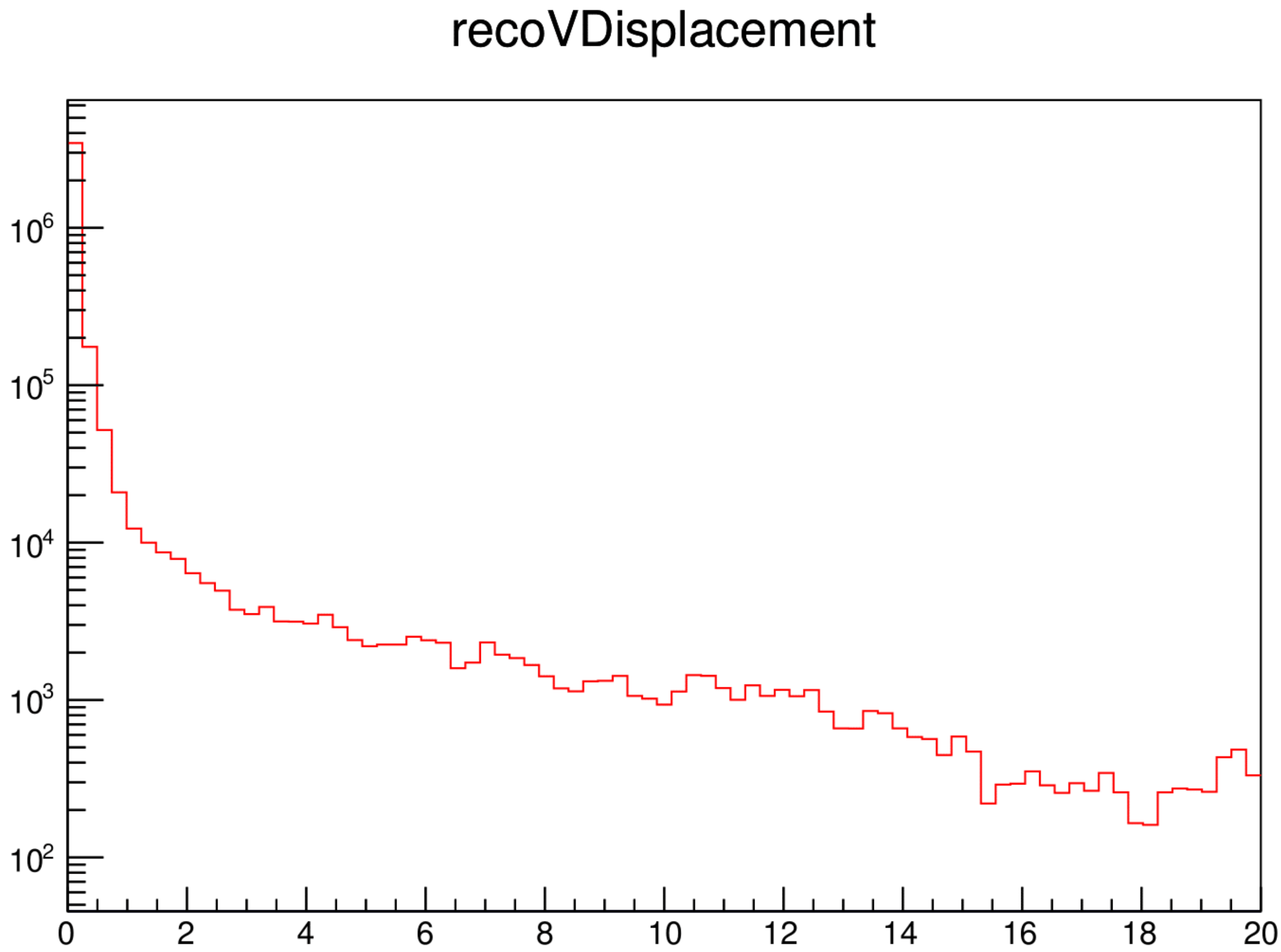
The metrics produced, which are useful for this technique at least, are as follows:

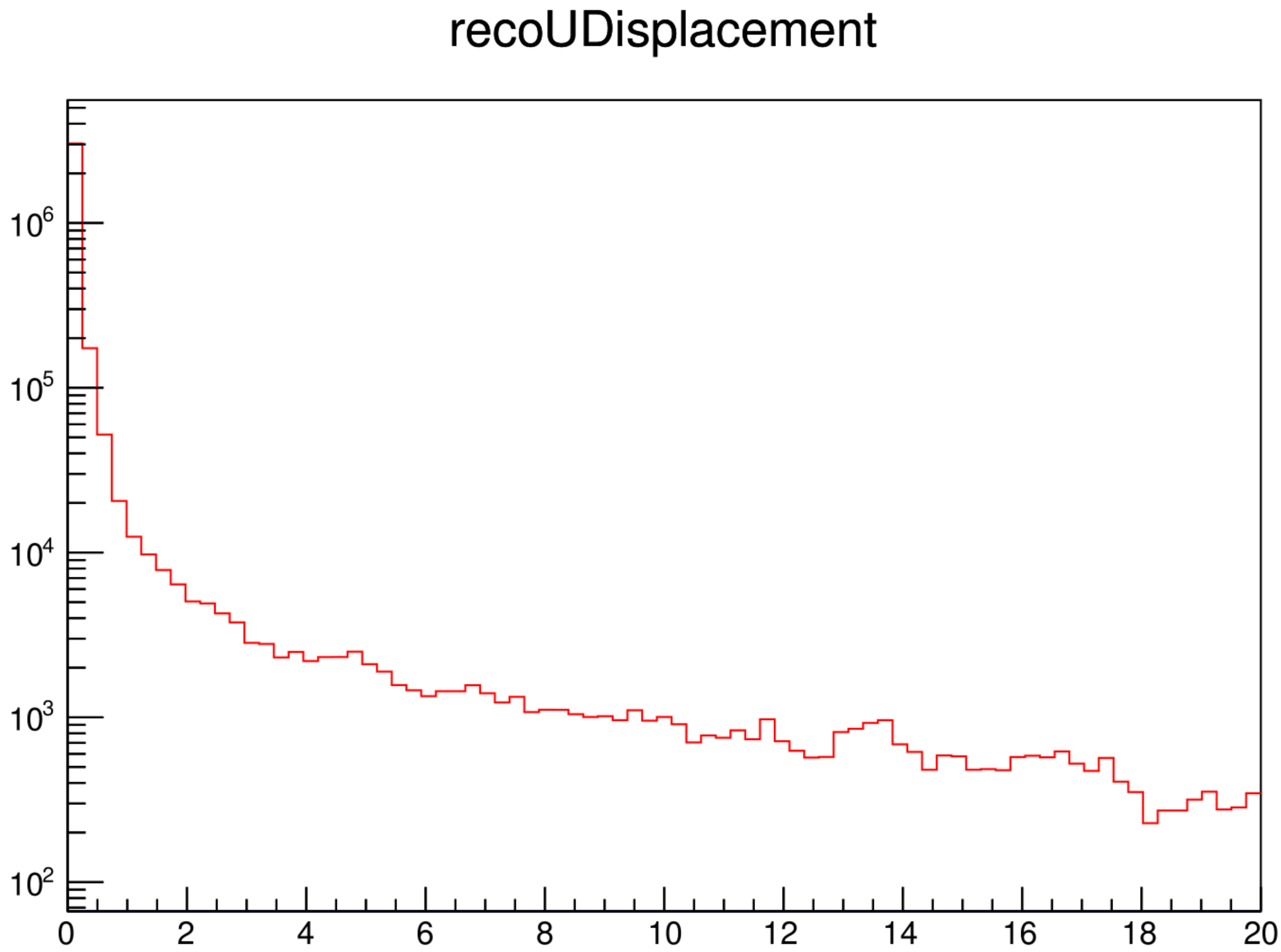
- Completeness.
- 'Wiggle' / Measure of the angle relative to a fit built along the 3D hits, done per hit.
- Displacement from a fit built along the 3D hits (i.e. a measure of outliers from the main track component).
- Ratio of interpolated to non-interpolated hits.
- A 2D versus 3D based metric.
  - Given a set of 3D hits, project every hit into 2D. Take these projected hits and build a fit over them, and see how far away the original (non-projected) 2D hits are from this fit.
  - This metric is especially important as the same approach can be taken with the MC 3D step positions, to give an MC based metric to steer the training of the BDT.

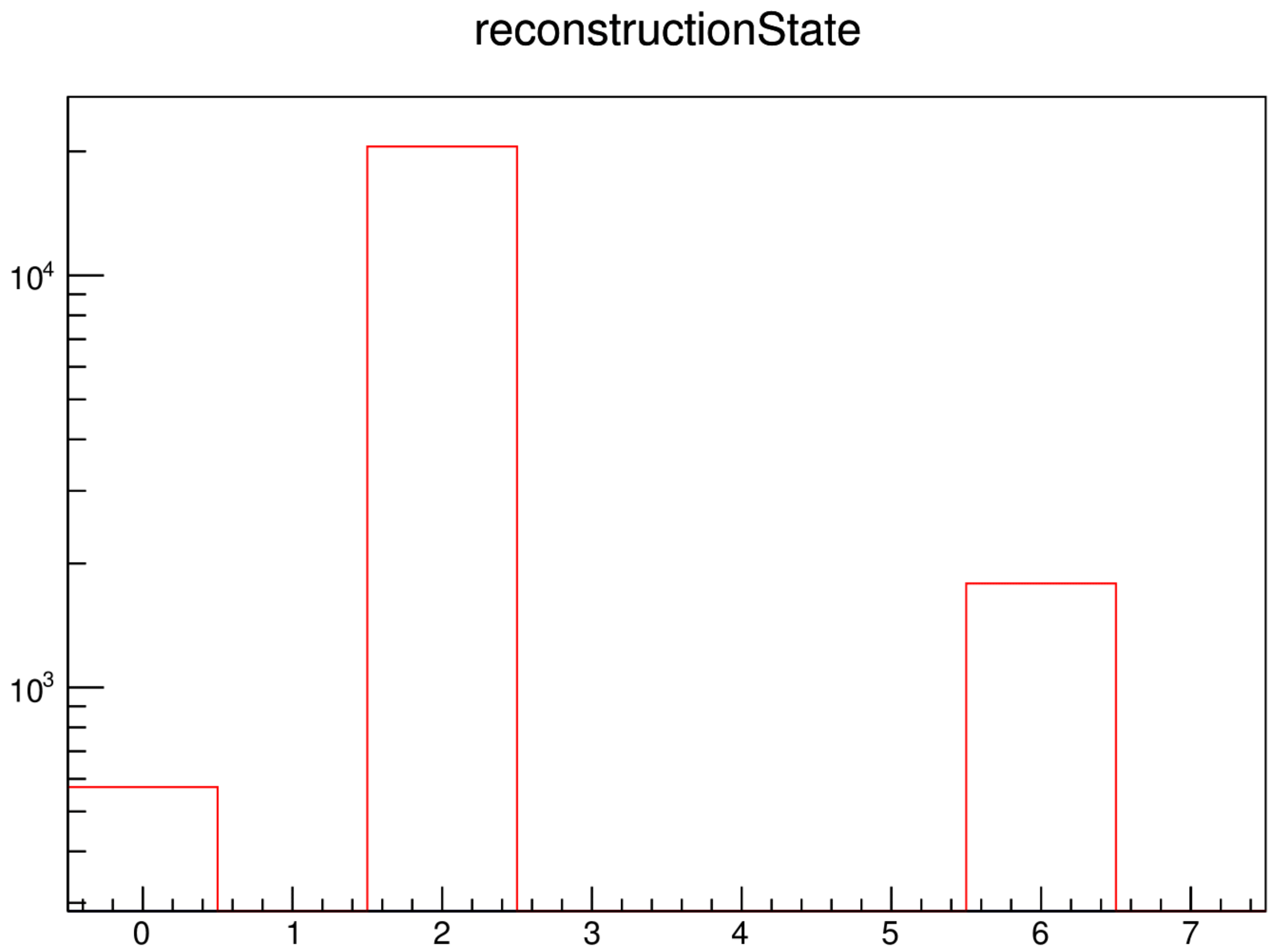
For any metric done on a per hit basis, the average was taken, before being passed as an input into a BDT (this is still ongoing).

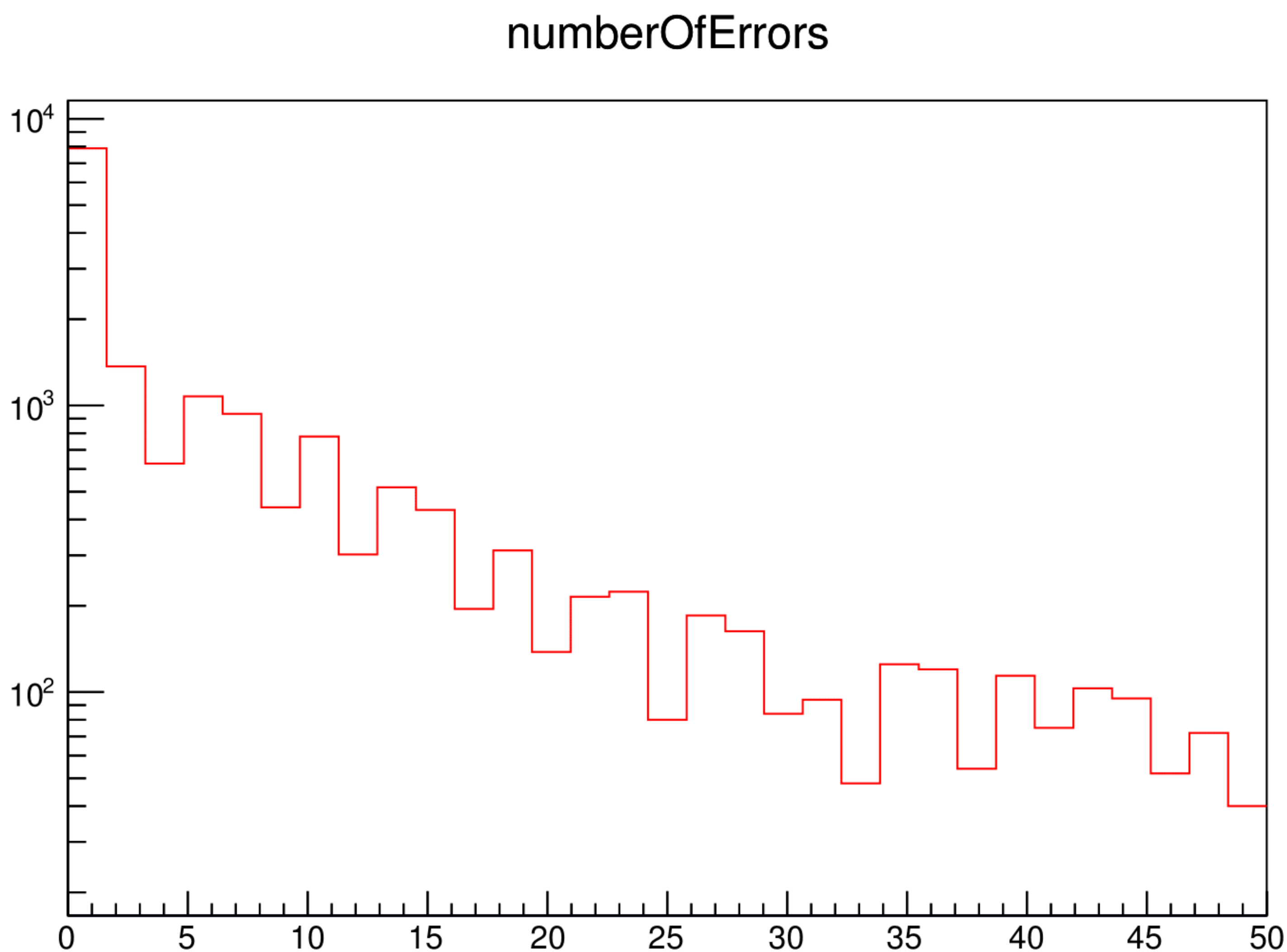

trackWasReconstructed
10^4
0
0.2
0.4
0.6
0.8
1

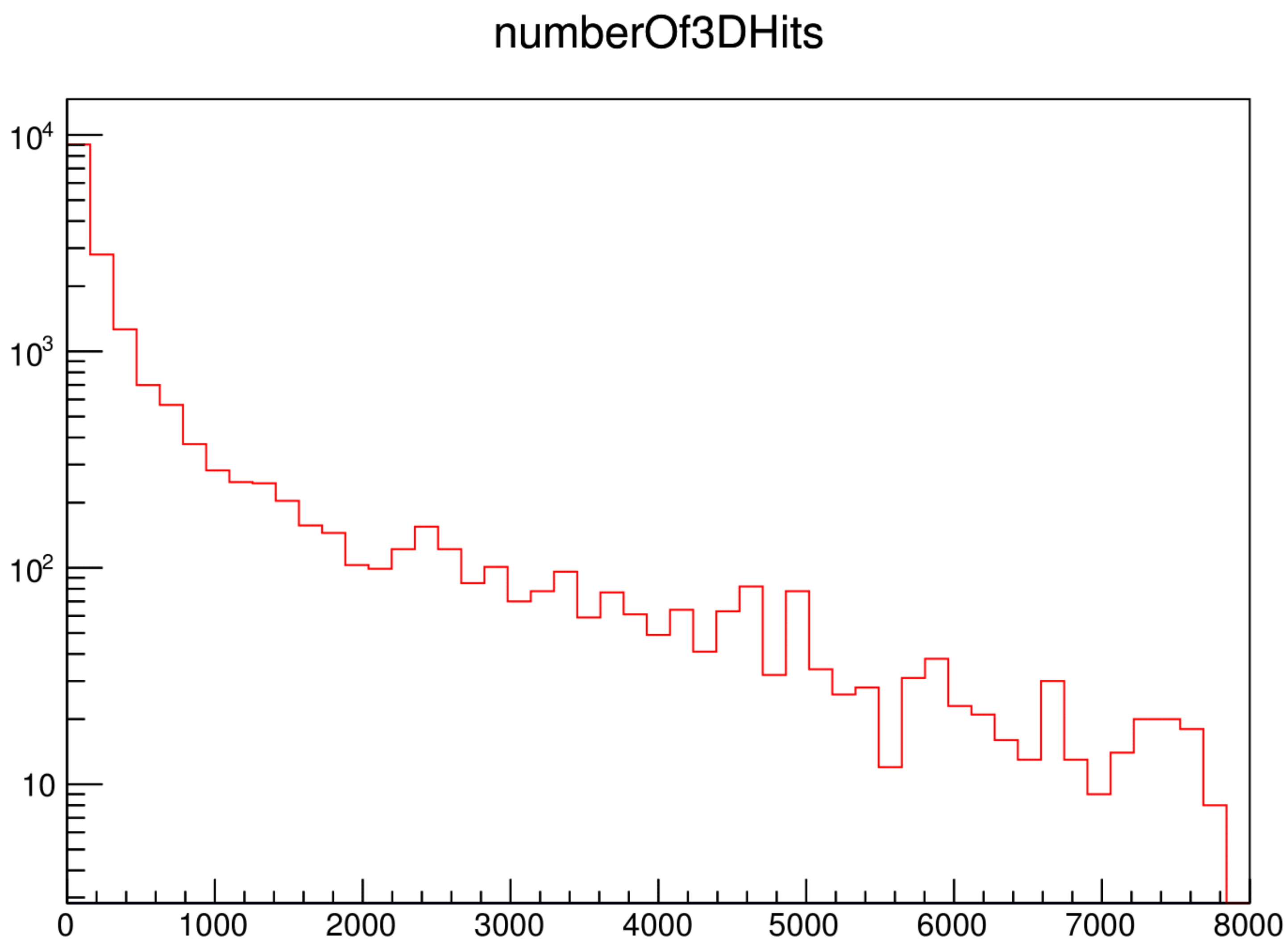

recoWDisplacement
10^6
10^5
10^4
10^3
10^2
0
2
4
6
8
10
12
14
16
18
20

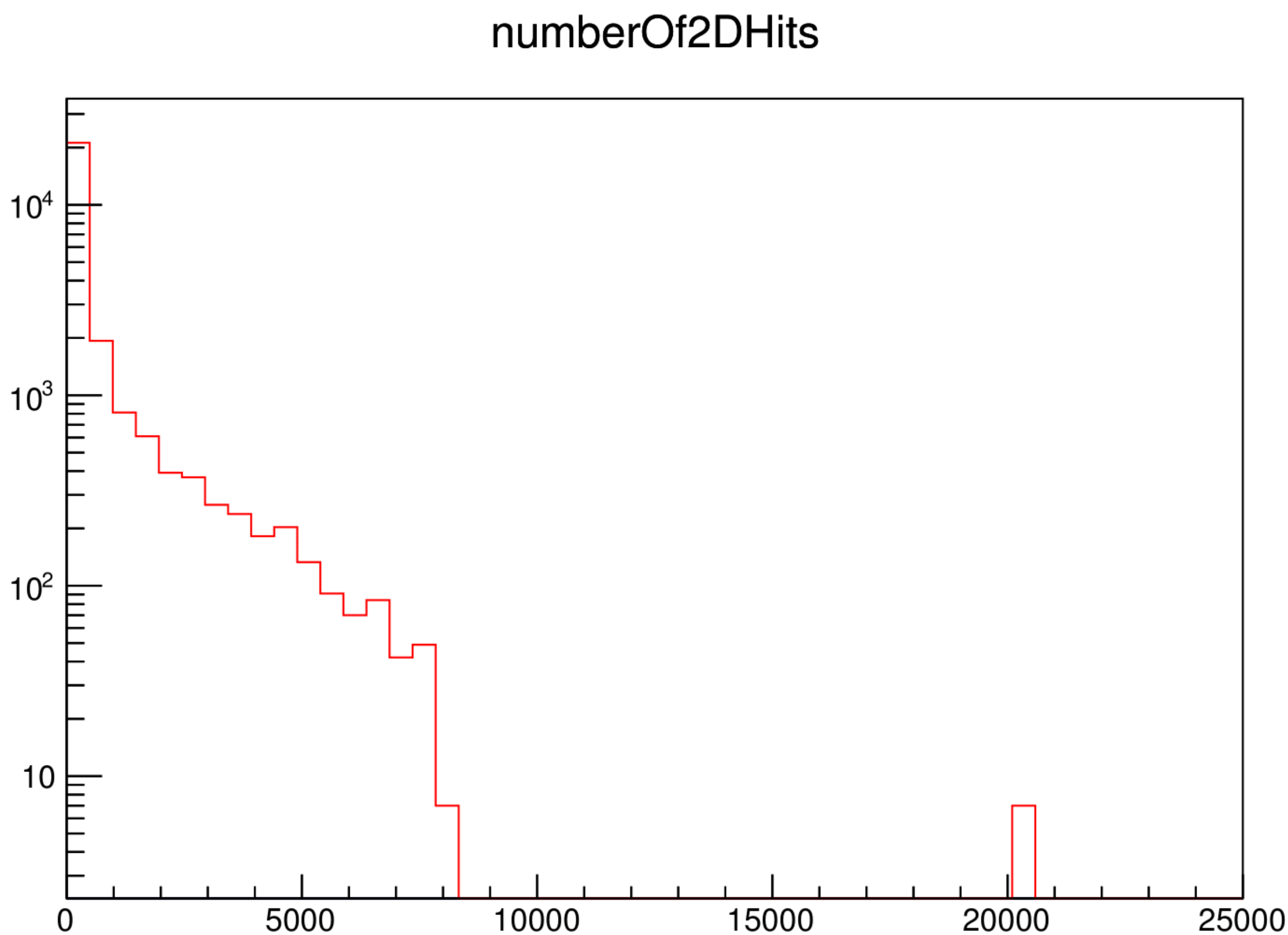

recoVDisplacement
10^6
10^5
10^4
10^3
10^2
0
2
4
6
8
10
12
14
16
18
20


recoUDisplacement
10^6
10^5
10^4
10^3
10^2
0
2
4
6
8
10
12
14
16
18
20

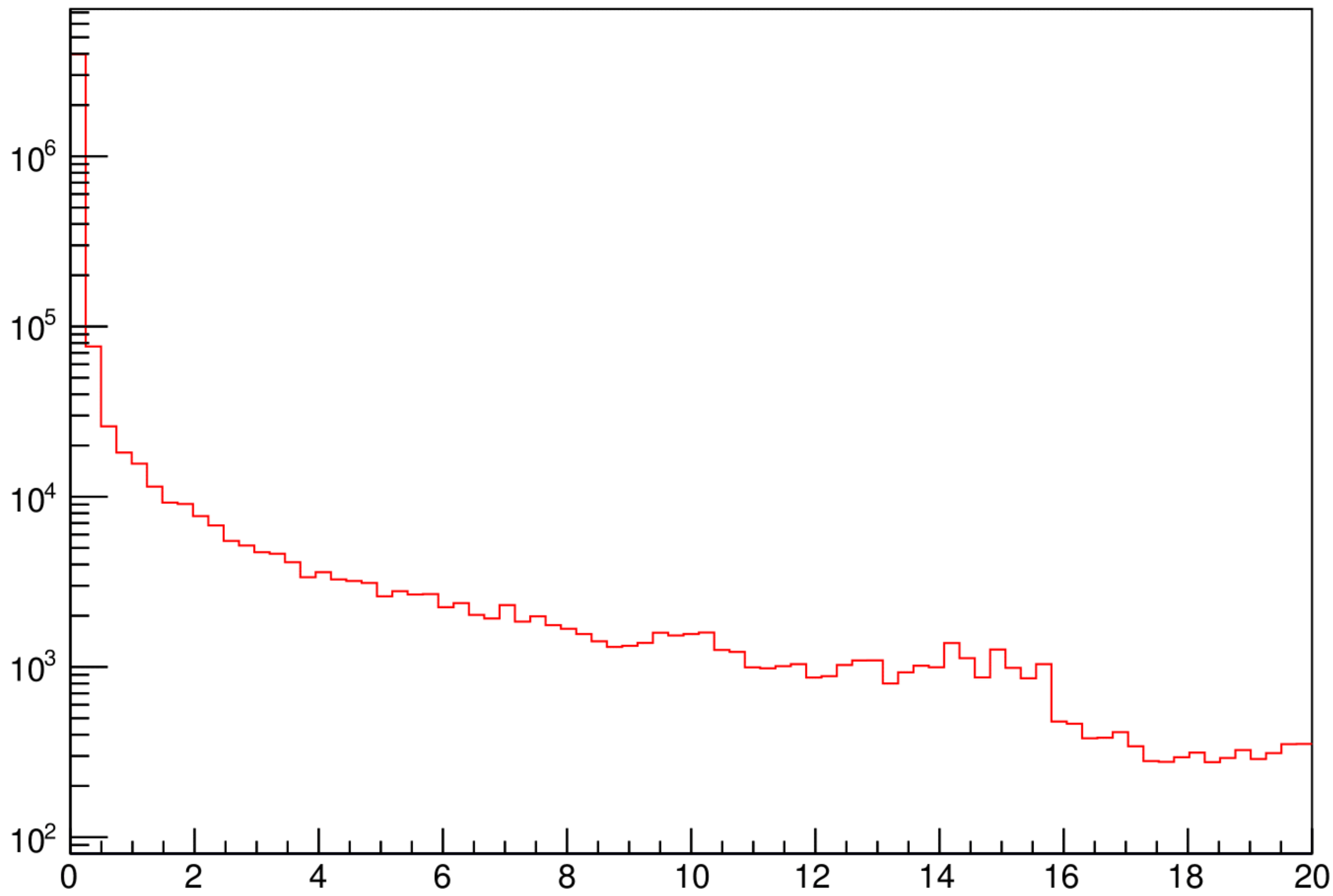

reconstructionState
10^4
10^3
0
1
2
3
4
5
6
7

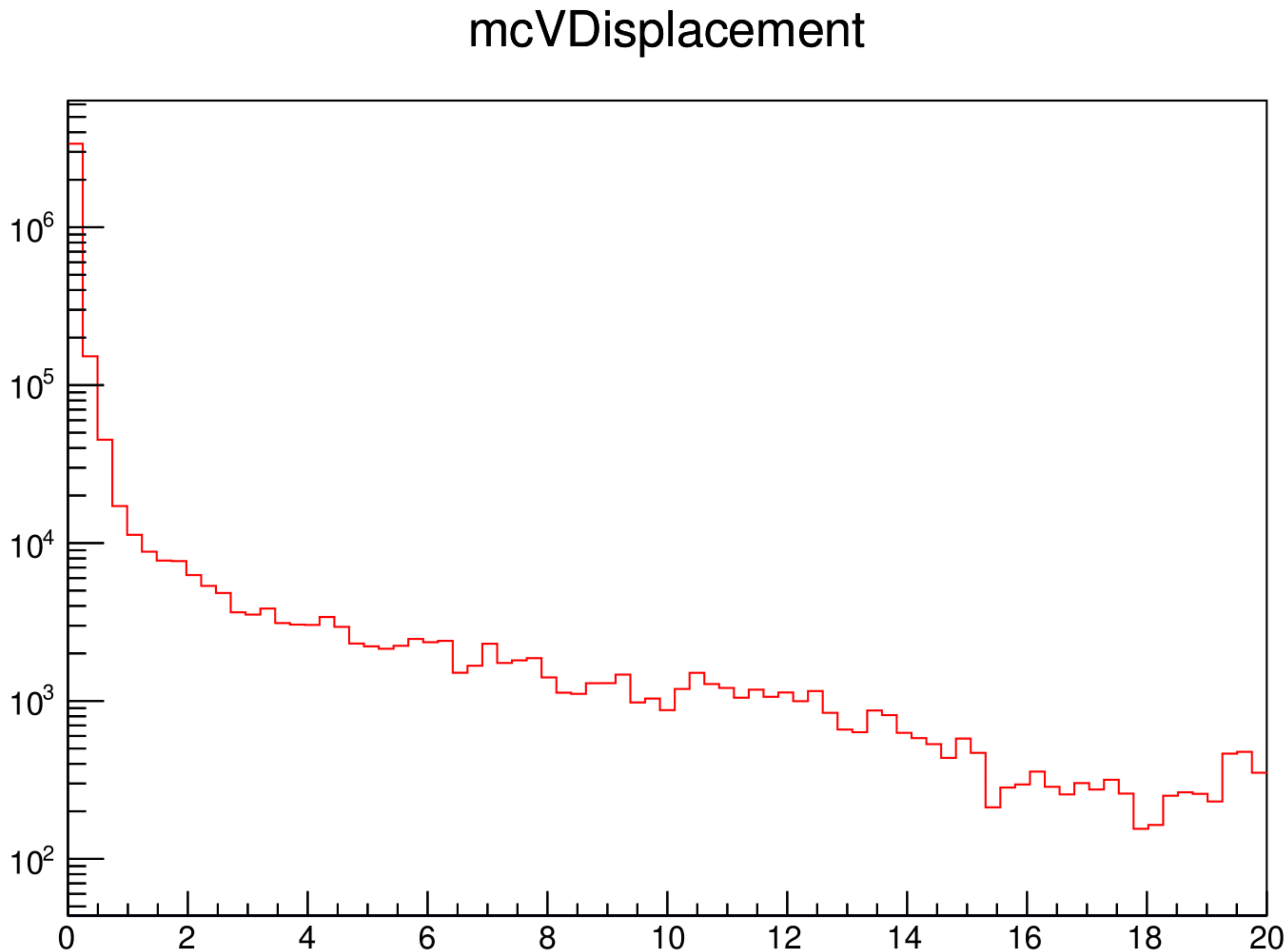

numberOfErrors
10^4
10^3
10^2
0
5
10
15
20
25
30
35
40
45
50


numberOf3DHits
10^4
10^3
10^2
10
0
1000
2000
3000
4000
5000
6000
7000
8000

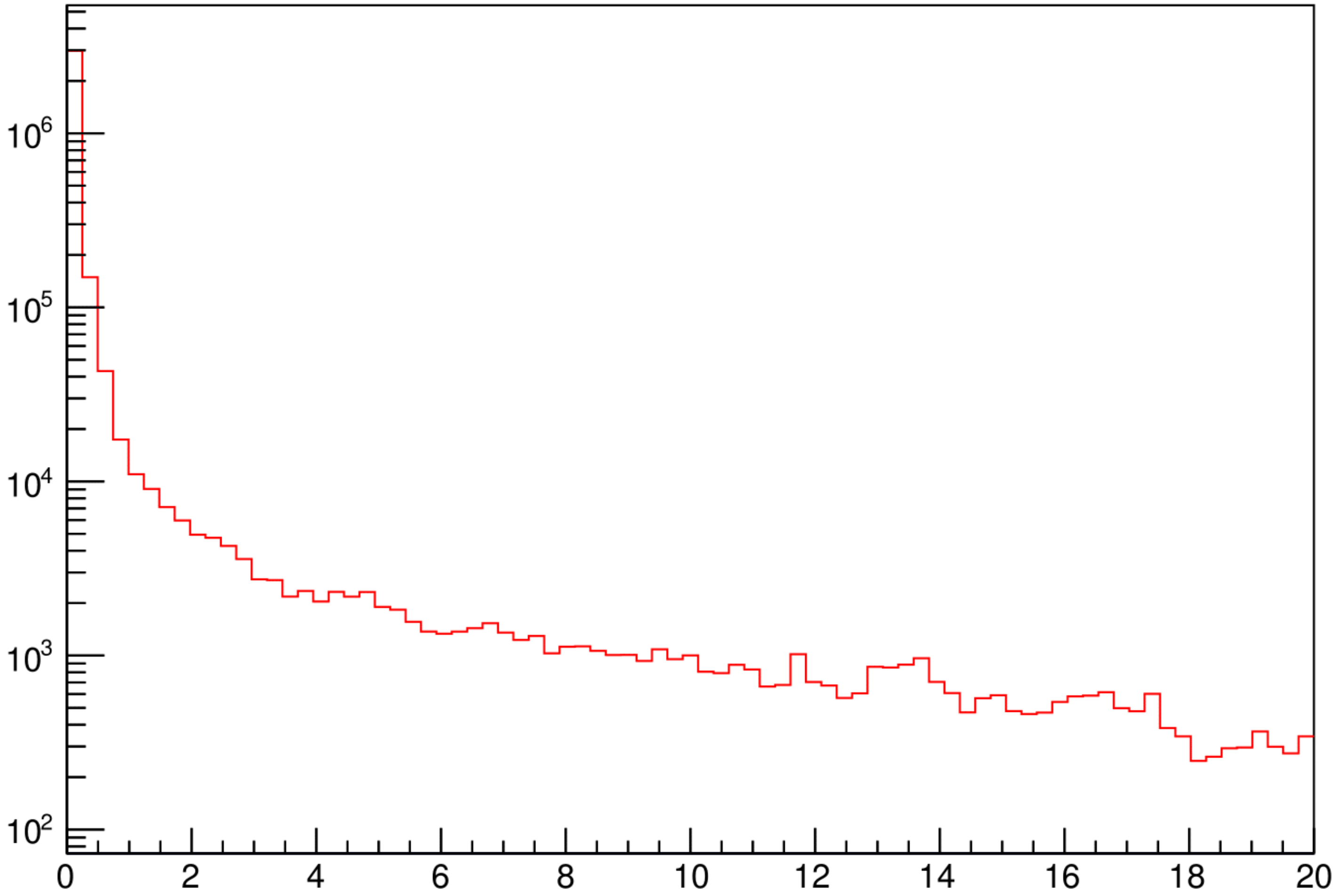

numberOf2DHits
10^4
10^3
10^2
10
0
5000
10000
15000
20000
25000

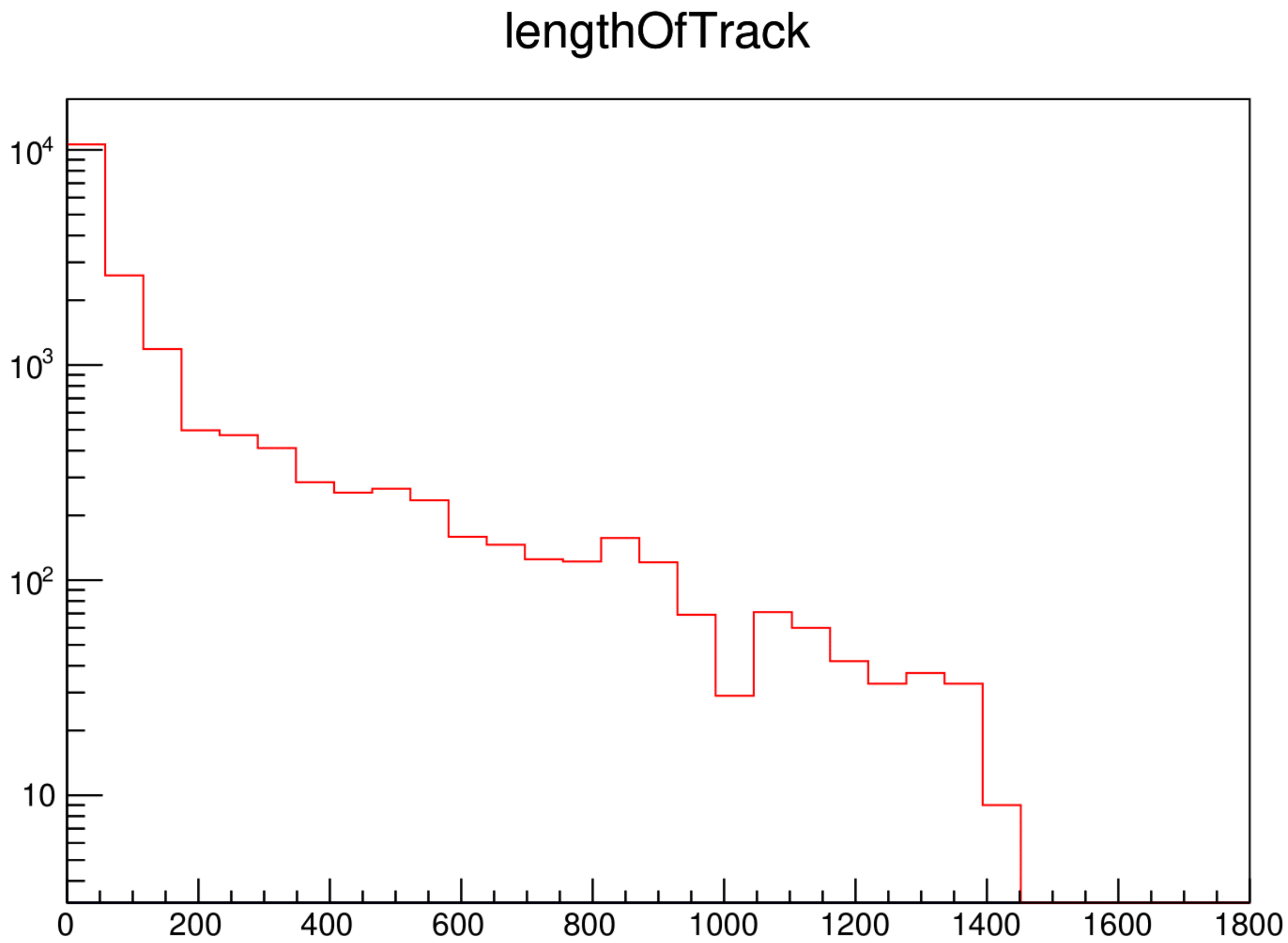
mcWDisplacement



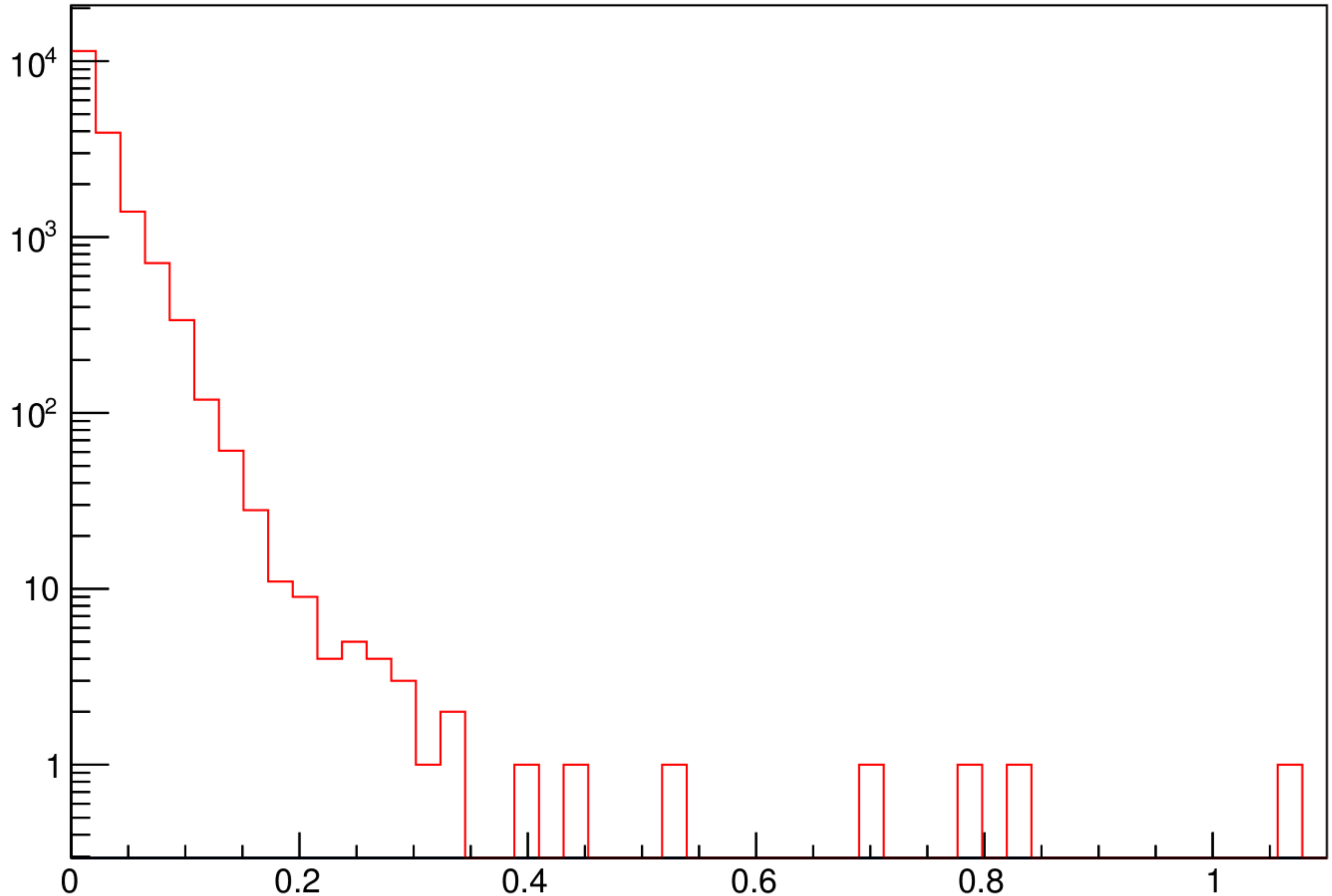

mcVDisplacement
10^6
10^5
10^4
10^3
10^2
0
2
4
6
8
10
12
14
16
18
20

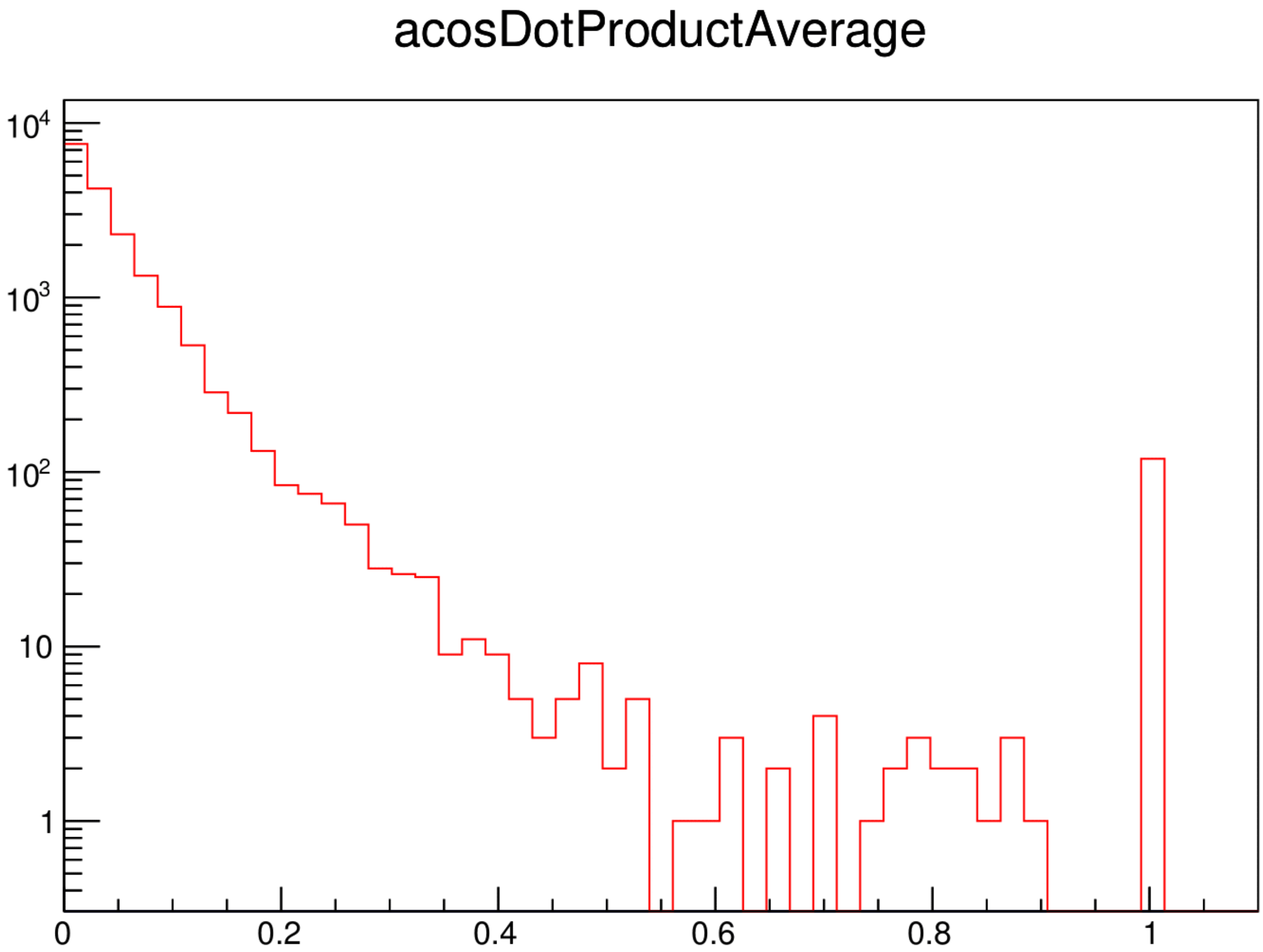
mcUDisplacement




lengthOfTrack
10^4
10^3
10^2
10
0
200
400
600
800
1000
1200
1400
1600
1800

distanceToFitAverage




acosDotProductAverage
10^4
10^3
10^2
10
1
0
0.2
0.4
0.6
0.8
1