



# Horizontal Muon Trigger

Charlie Batchelor

FD Data Selection

November 30<sup>th</sup> 2021



# Outline

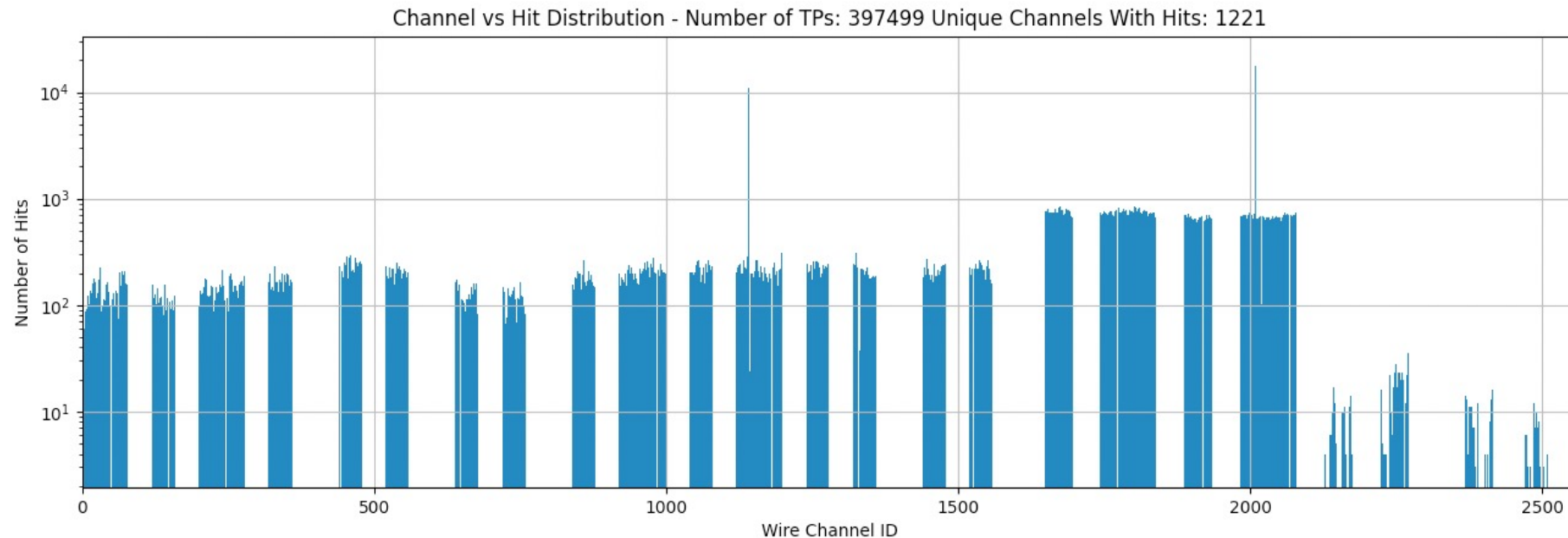
- Previous Questions
- Testing Methods
  1. faketp\_chain
  2. Electronics Emulator – minidaqapp
- Next Steps

# Previous Questions

- tps\_link\_11.txt – What is this data we've been testing on up to now?
- How do we know the algorithm is ready for 'live' DAQ on coldBox?

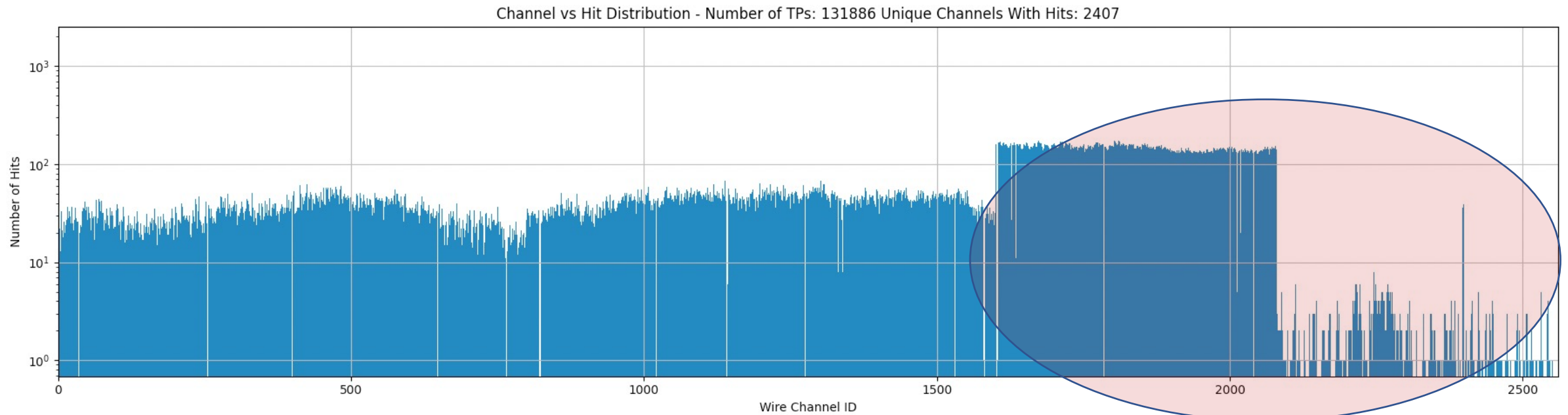
# tps\_link\_11.txt

- Up to now, tests of horizontal muon trigger have been performed on this input file. It contains a list of generated TPs from the **half** of the channels on the first APA of the ProtoDUNE SP 1 run.
- The TPs from the other half of the APA are stored in tps\_link\_5.txt. These files are obtainable at: [cernBox](#)



# tps\_link\_<11/5>\_no\_noisy.txt

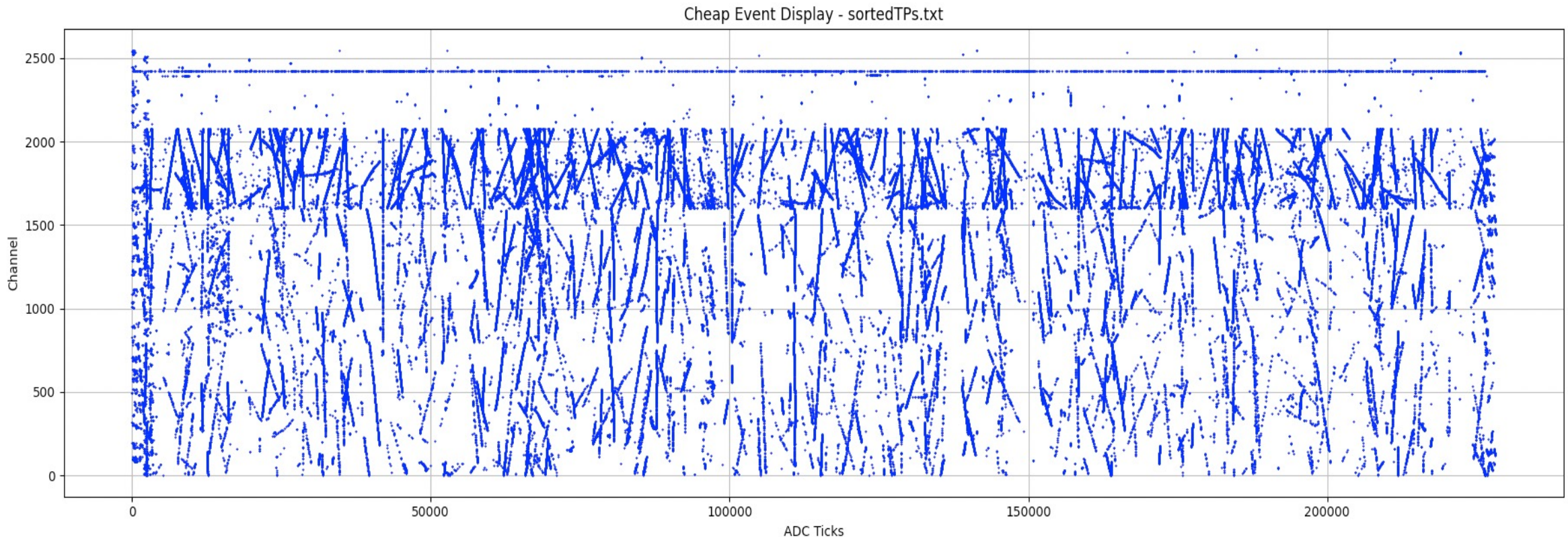
- The same data files with known noisy channels removed were merged, and the full histogram looks like this:



- There seems to be an issue (with mapping?) on the last ~500 channels.

# tps\_link\_<11/5>\_no\_noisy.txt – Event Display

- A cheap event display was made, so we know what we're tuning the algorithm on.
- Lots of muons! Good for tuning the algorithm.

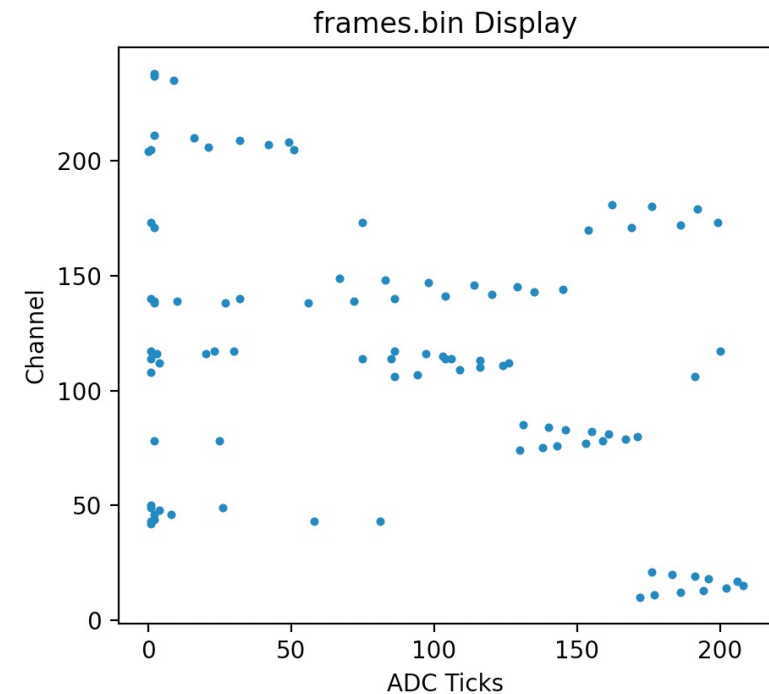
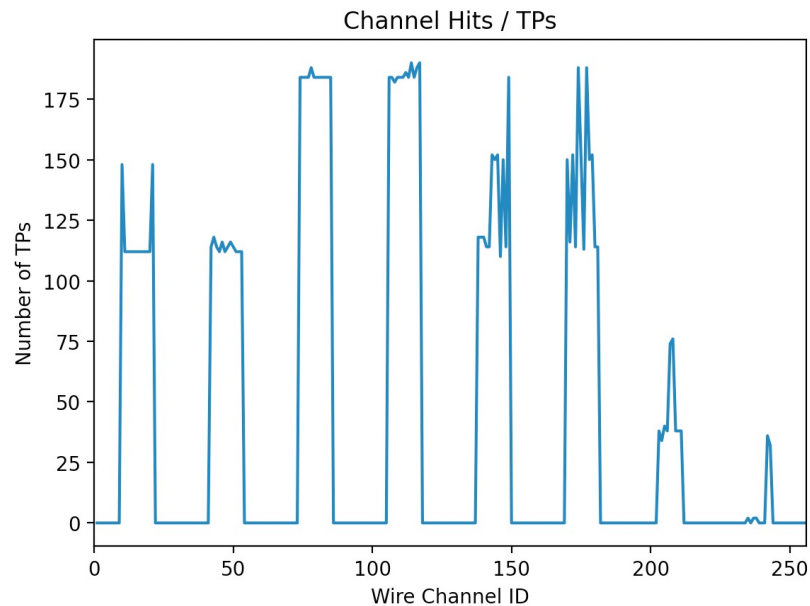


# How do we know if the code is ready for 'live' DAQ?

- Up to now, we have been testing the algorithm using the [faketp chain](#) in the trigger area of DUNEDAQ. This is a good way to test the algorithm in isolation from other DUNEDAQ modules/code.
- The next natural step then is to test on the minidaqapp configurations, which include emulated electronics. These read in the [frames.bin](#) file and generate the TPs to then feed the algorithm.
- Let's take a look at what that file looks like too first...

# frames.bin – Event Display & Histogram

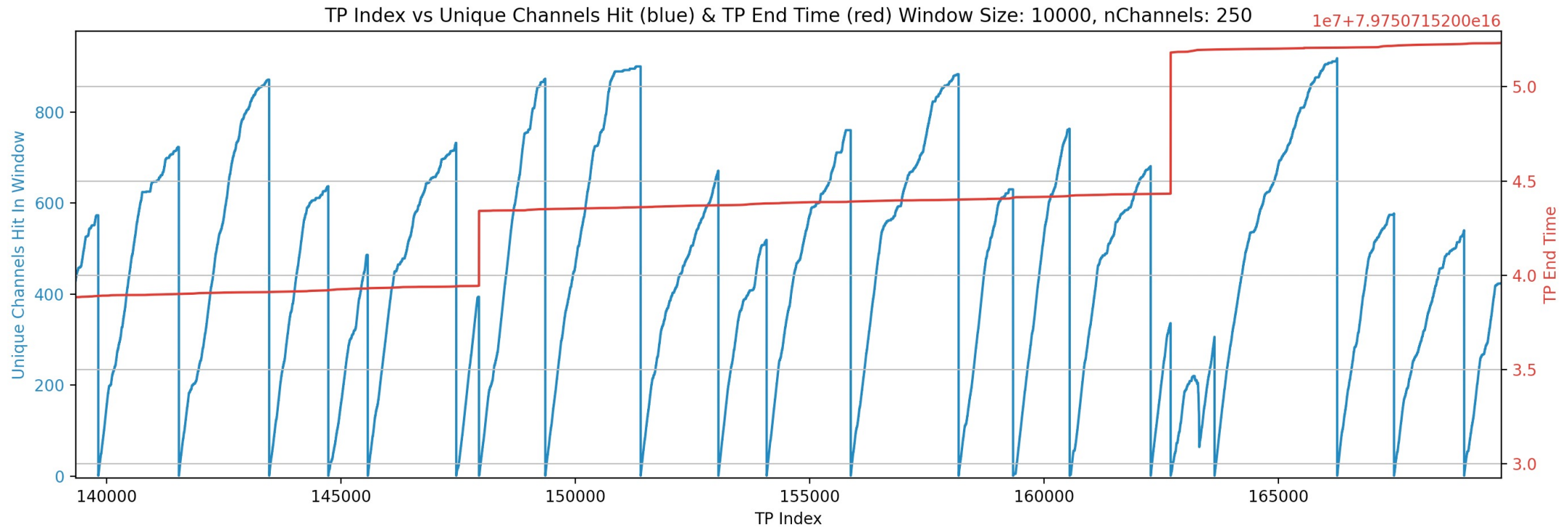
- frames.bin contains raw binary from a single link (1 10<sup>th</sup> of an APA) corresponding to 256 channels.
- We are seeing hits on only ~1/3 of the channels, think these are the collection channels only.





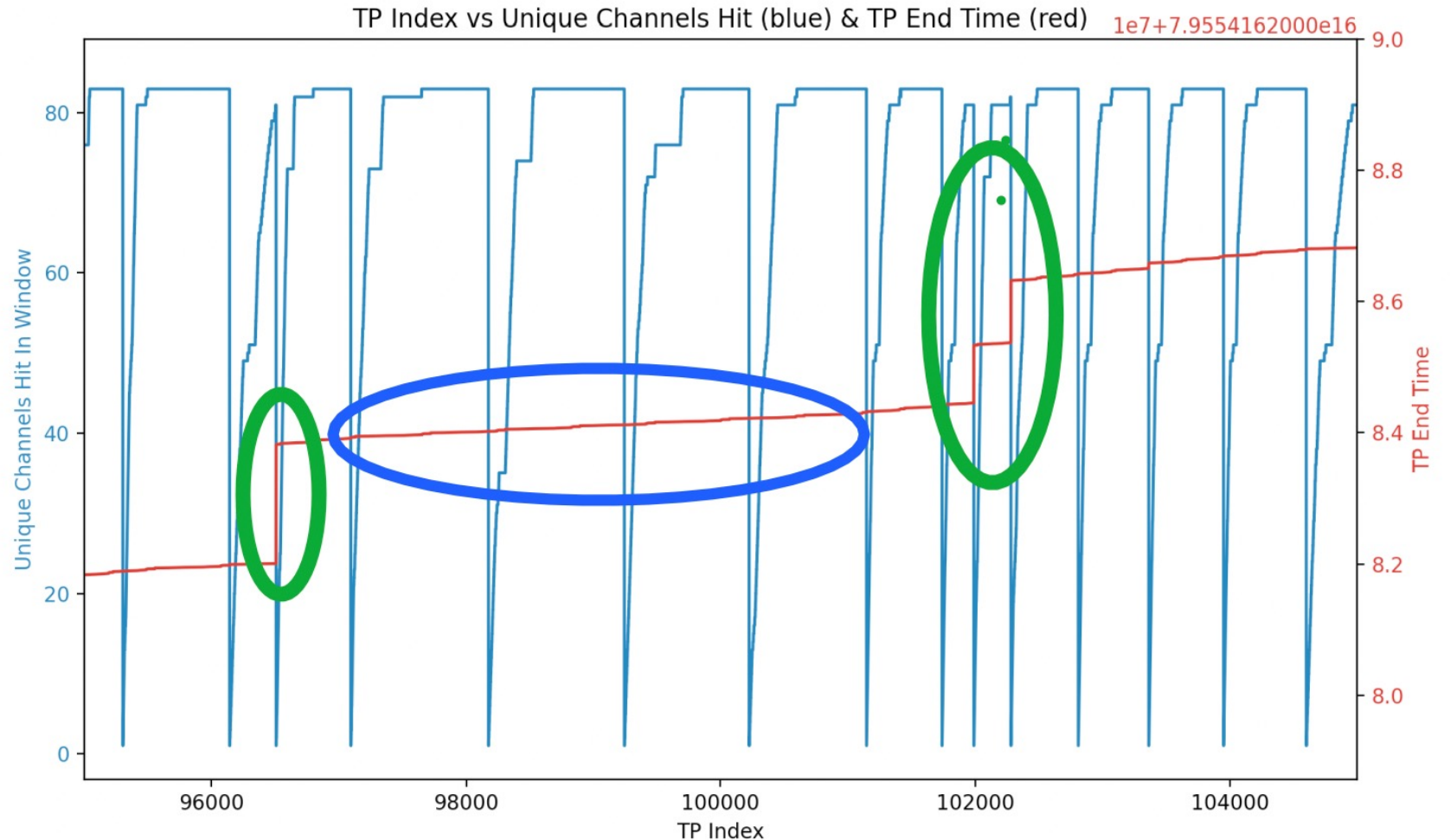
# Checks : tps\_link (faketp\_chain)

- We can see trigger activities being generated when windows contain over 250 unique channels with hits.



# frames.bin – Testing

- Green areas show large time gaps in TP start time, corresponding to the window being reset.
- Blue areas are where window is reset due to TAs being created!



# Ready for live DAQ?

- The minidaqapp tests show that the current algorithm can keep up with the rate of incoming TPs and other DUNEDAQ modules. This is done by removing the 'slow down factor' -s in the config generation.
- Steps to replicate this are given at the back of these slides.
- Current Plan:
  1. Test 'next to simplest' trigger on ColdBox – ADCSimpleWindow, from which the Horizontal Muon code is based.
  2. Clean up horizontal muon code and get this pushed up so we can trigger on some muons!

# Next Steps

- Trigger Candidate Maker Logic – This is in progress, TCMaker will see TPs from multiple APAs, not just a single one like cold box.
- Implementing an Adjacency Check. Currently generating TAs based on the number of unique channels hit within a window. (I'm working on this now...)
- Tune the algorithm on tps\_link files – lots of activity!
- Push up code -> Test on ColdBox! We should get the code tested to check we can trigger using the n\_channels\_threshold variable first

# Summary

- Algorithm has been tested in isolation using tps\_link files.
- It has also been tested in a full data stream simulation among other DUNEDAQ modules (using nanorc) which is a good check that it can keep up with the rate of TPs and rest of DAQ.
- We will continue to tune and implement more advanced trigger types such as adjacency checks for future cold box runs.

# Do Try This At Home

- Get a copy of the frames.bin file by clicking [here](#).
- The configuration to test the trigger algorithm may be generated using the below command within your DUNEDAQ work environment:

```
python -m minidaqapp.nanorc.mdapp_multiru_gen -s 10 --enable-software-tpg -d
./frames.bin -o . --trigger-activity-plugin
TriggerActivityMakerHorizontalMuonPlugin --trigger-candidate-plugin
TriggerCandidateMakerHorizontalMuonPlugin --trigger-activity-config
"dict(n_channels_threshold=60)" horizontalMuon_emulator
```

```
nanorc horizontalMuon_emulator boot init conf start 1 resume wait 30 stop
```