

# TMS Timing Simulation Update Redux

## TMS Meeting

Jeffrey Kleykamp

2023-02-01



# Introduction

- DUNE/dune-tms package handles edep sim output to do TMS simulation and reco
  - Made by Clarence Wret
- With overlay/pileup option enabled, we get  $\sim 130$  interactions per spill
- With scintillator, timing resolution is very fast
  - Time slicing
- Also added timing simulation

# Reorganization of dune-tms code

Simulation:

Reconstruction:

Merge edep hits

Pedestal subtraction

Track/cluster reconstruction

# Reorganization of dune-tms code

Simulation:

Merge edep hits  
Pedestal subtraction

Reconstruction:

Merge edep hits  
Pedestal subtraction

Track/cluster reconstruction



# Added timing effects, and time slicer

## Simulation:

Merge edep hits

Pedestal subtraction

### **Optical model effects**

Changes to hit energy

### **Timing reco effects**

Changes to hit time

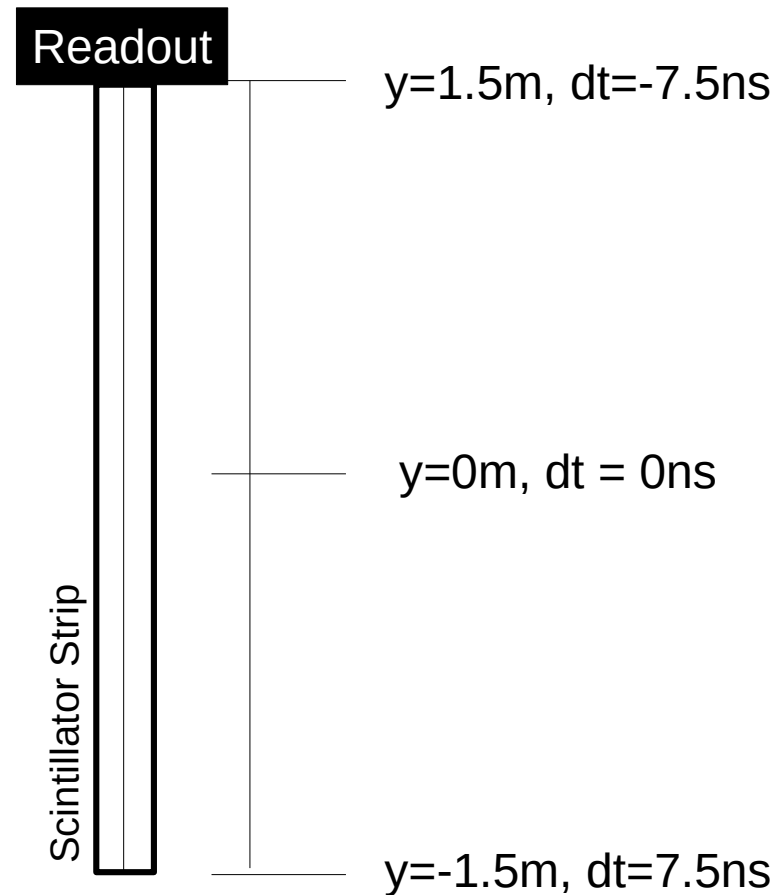
## Reconstruction:

### **Time Slicing**

Track/cluster reconstruction

# Reconstruction Effects

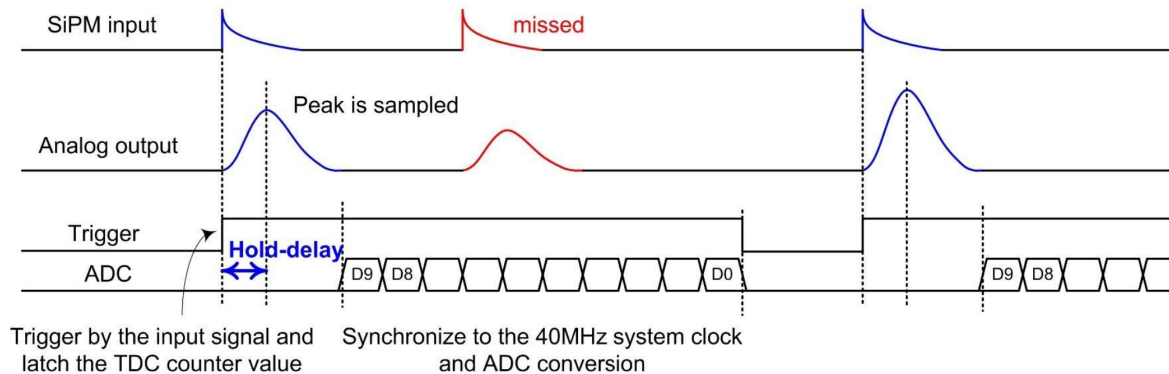
- Electronic errors ✓
  - Used  $dt = 1\text{ns}$ , seen  $dt \sim 150\text{ps}$
- Optical fiber delay lengths ✓
  - Assuming  $n=1.5$
- Time slew ✓
  - Next slide
- Deadtime simulation
  - Is  $500\text{ns}$  of deadtime correct?
- Other effects:
  - Effects from noise, cross talk, afterpulsing
- Need some help with the parameters



# Source of 500ns deadtime figure

- Slide from Sept collab meeting: [link](#)
- For testing specific ASIC alternative

## Hit digitization



### Conversion started after time-comparator fires

- Sample time stamp in TDC latches
- Configurable hold delay (~100ns) to sample peak
- ADC conversion, gain bit sampling
- Busy released after conversion is finished, ready for new hit → typical dead time ~500ns (10b ADC mode)



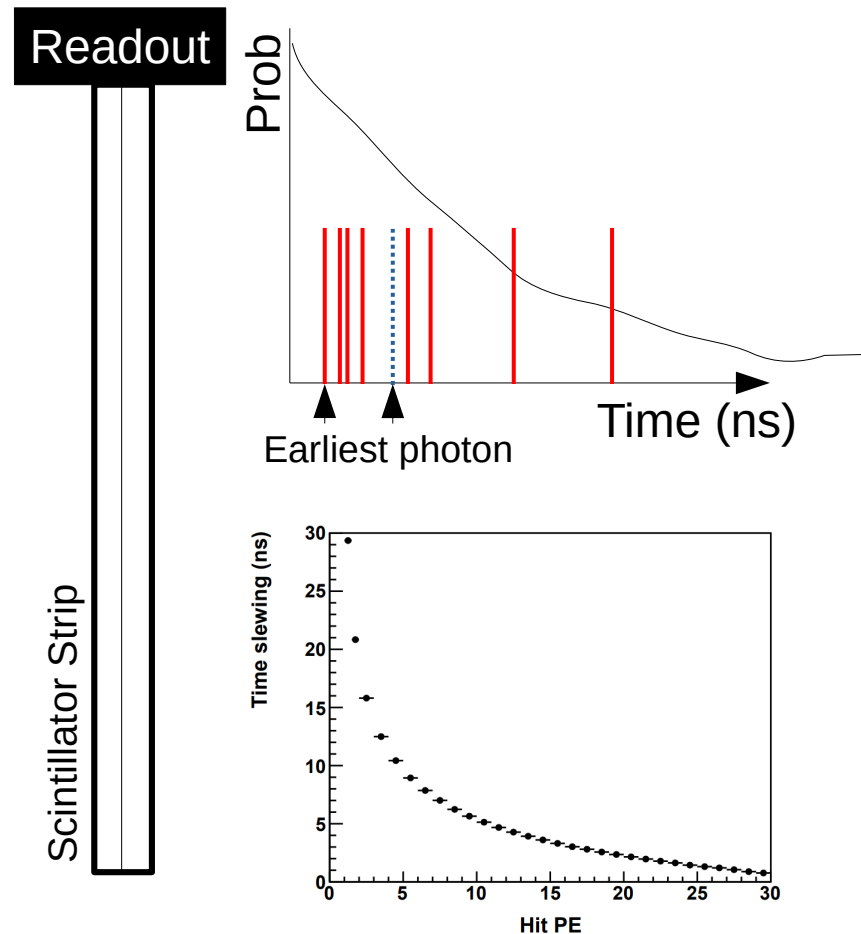
# More questions about deadtime

- How exactly would it work
- In groups of 12, 48, 64 channels?
  - ie everything connected to the same FEB
- Deadtime after
  - reading each hit
  - or only after digitizing 7-8 hits?



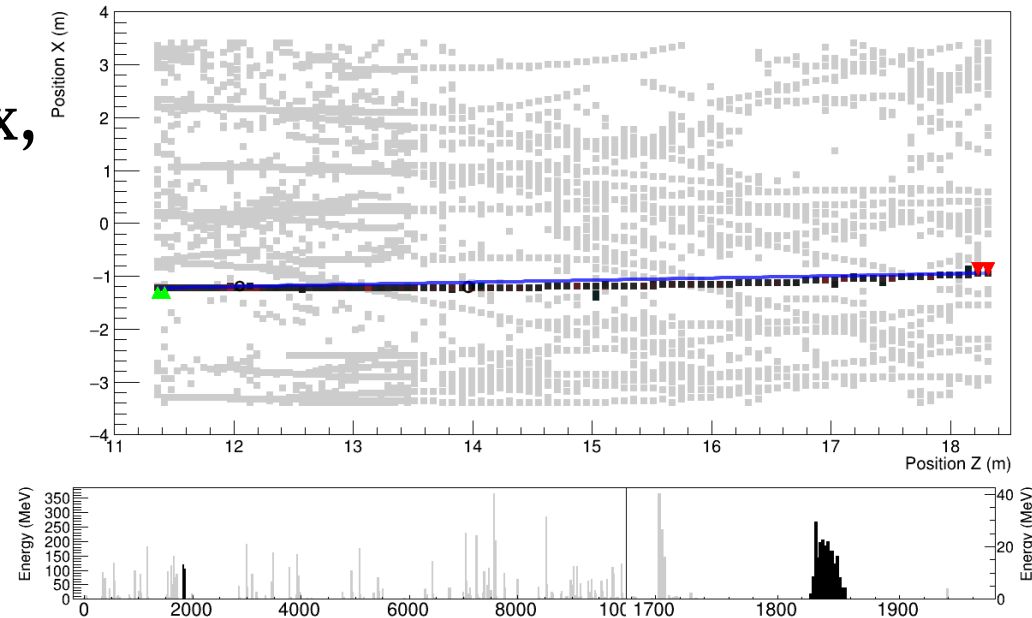
# Time Slew

- Hit time = earliest photon received
  - half go long way and bounce off end
  - Need optical model to correctly simulate PE
  - Assuming 50 PE / 1 MeV
- Assuming double exponential decay
  - WSF decay mean = 20ns (11ns?)
  - Scint decay mean = 2ns

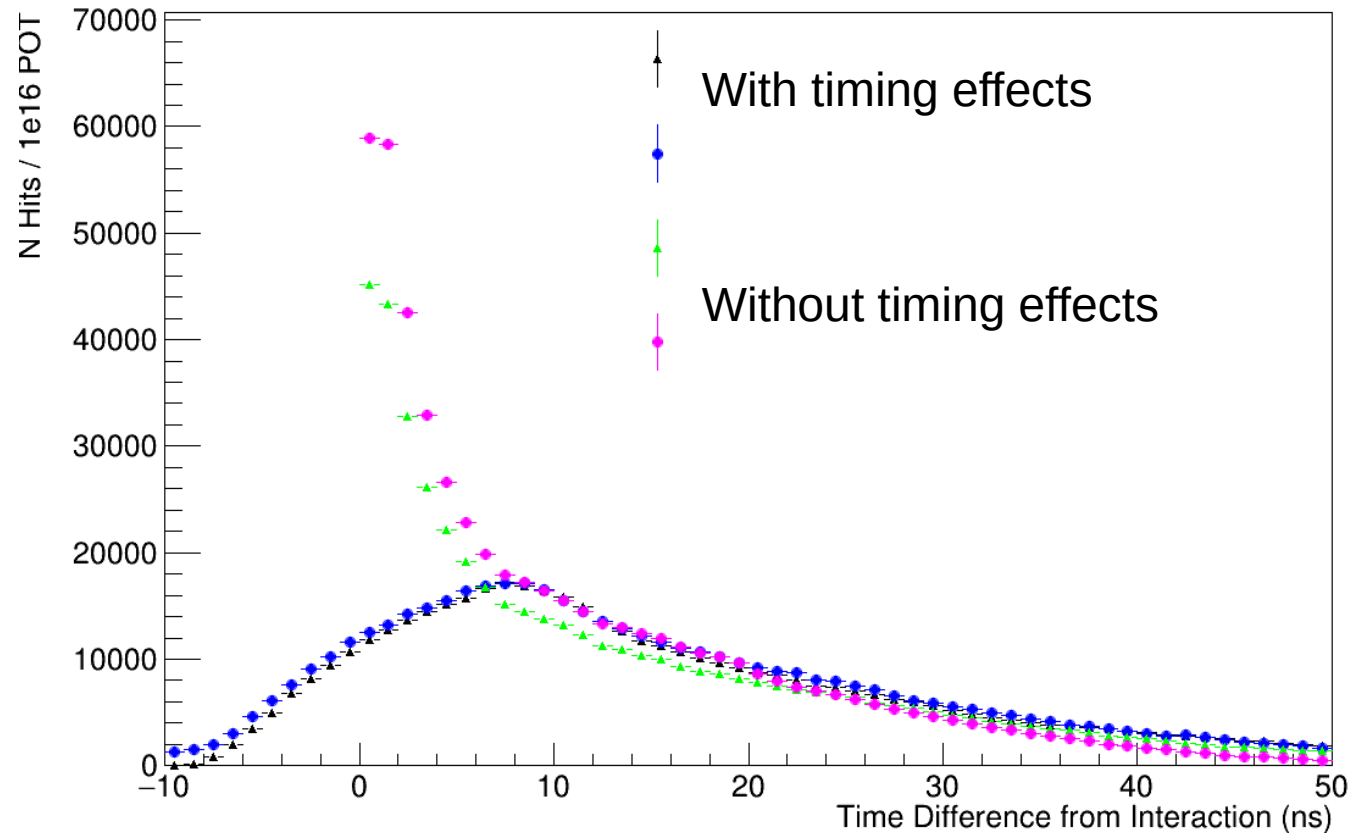


# Additional reco effect: poor calibration?

- Without a good sample of muons, it would be difficult to calibrate
- Find muons that start at the front and go to the back, and calibrate time using those
- Y position is currently in flux, but only need relative time between hits

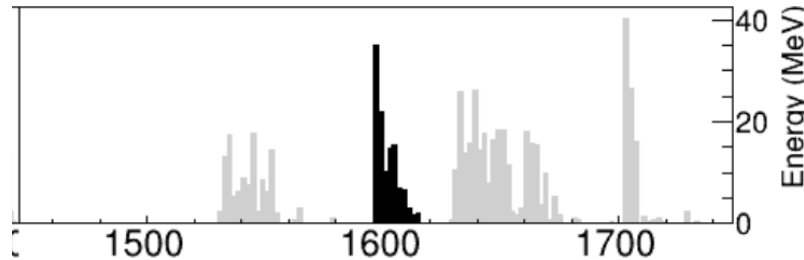


# Reco hit time – Interaction time



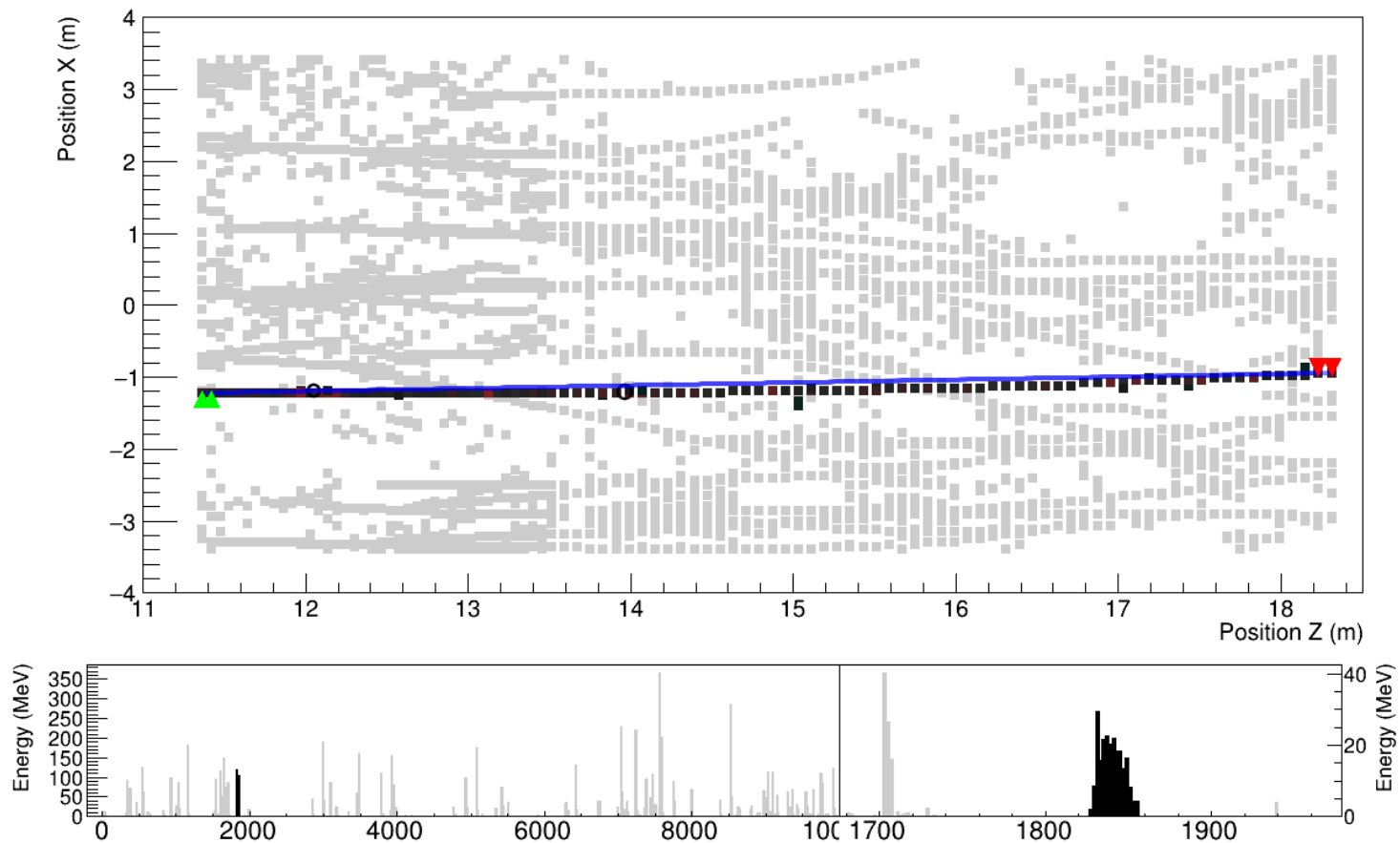
# Time Slicing Algorithm

- If energy within a small window is  $> \text{energy\_threshold\_1}$ , start adding hits into a slice
- Continue adding until energy in window  $< \text{energy\_threshold\_2}$ .

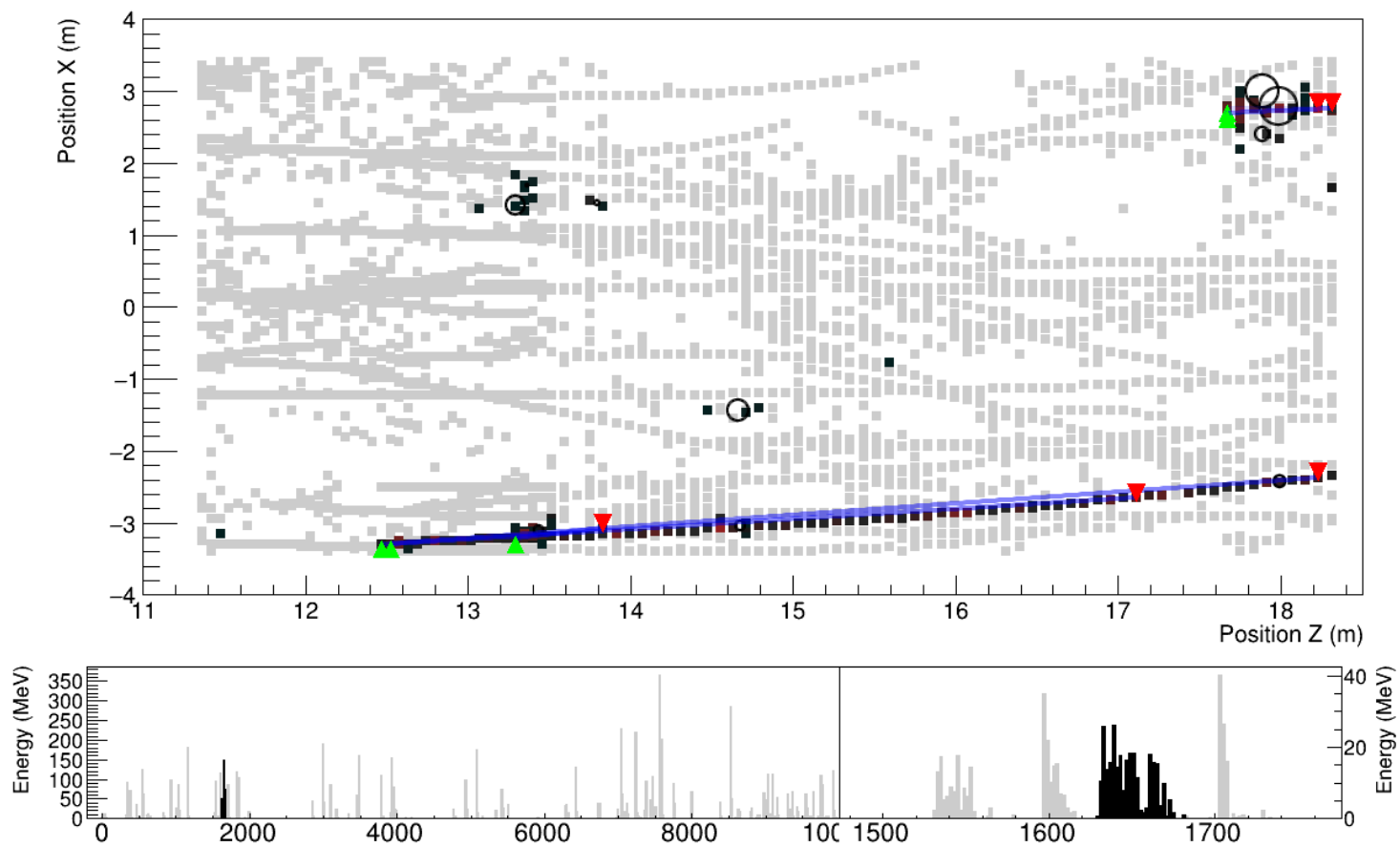


- Optimize to
  - Maximize energy fraction of single event
  - Minimize energy from secondary events
- Based on Minerva time slicing algorithm

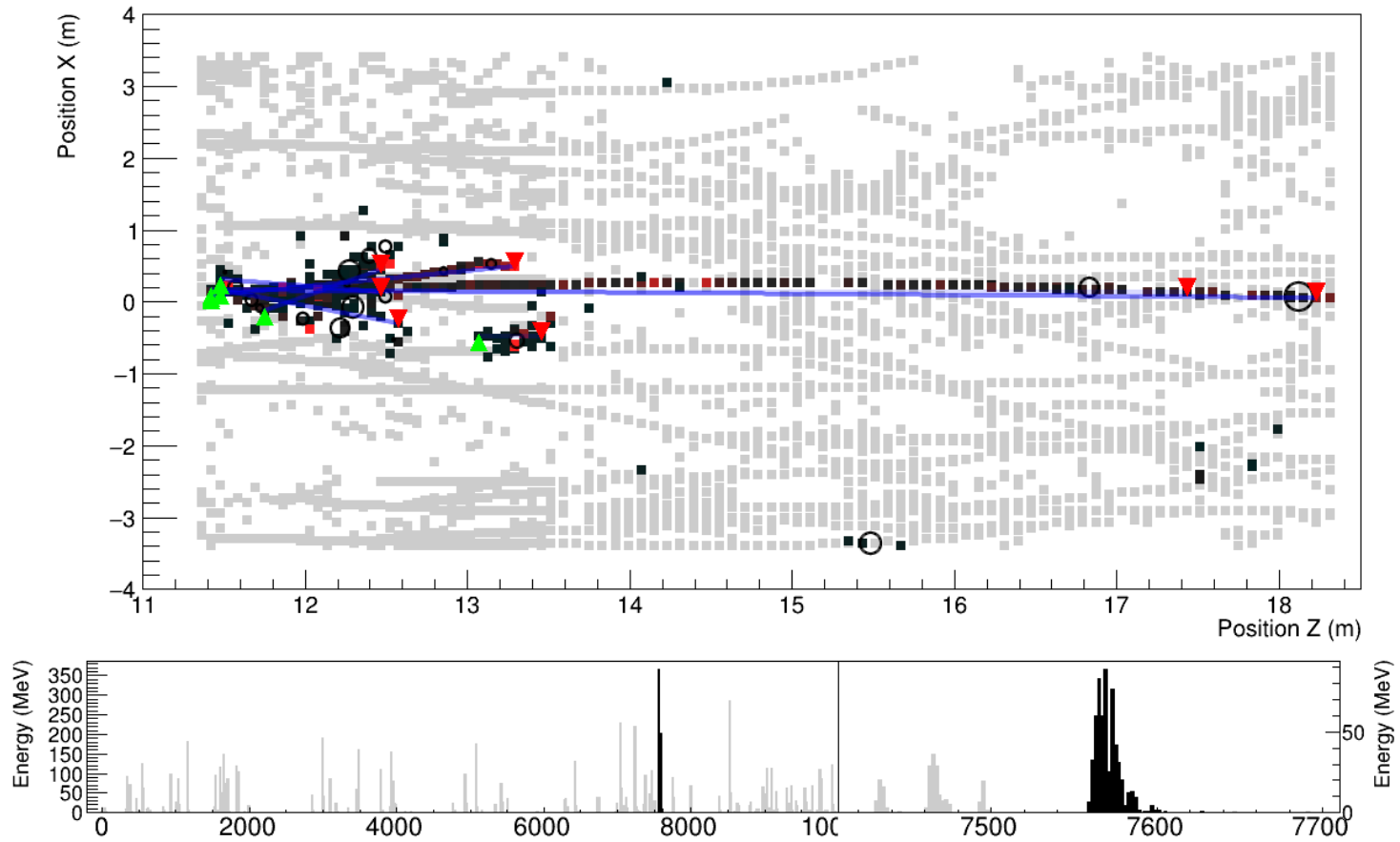
# Good separation for clear tracks



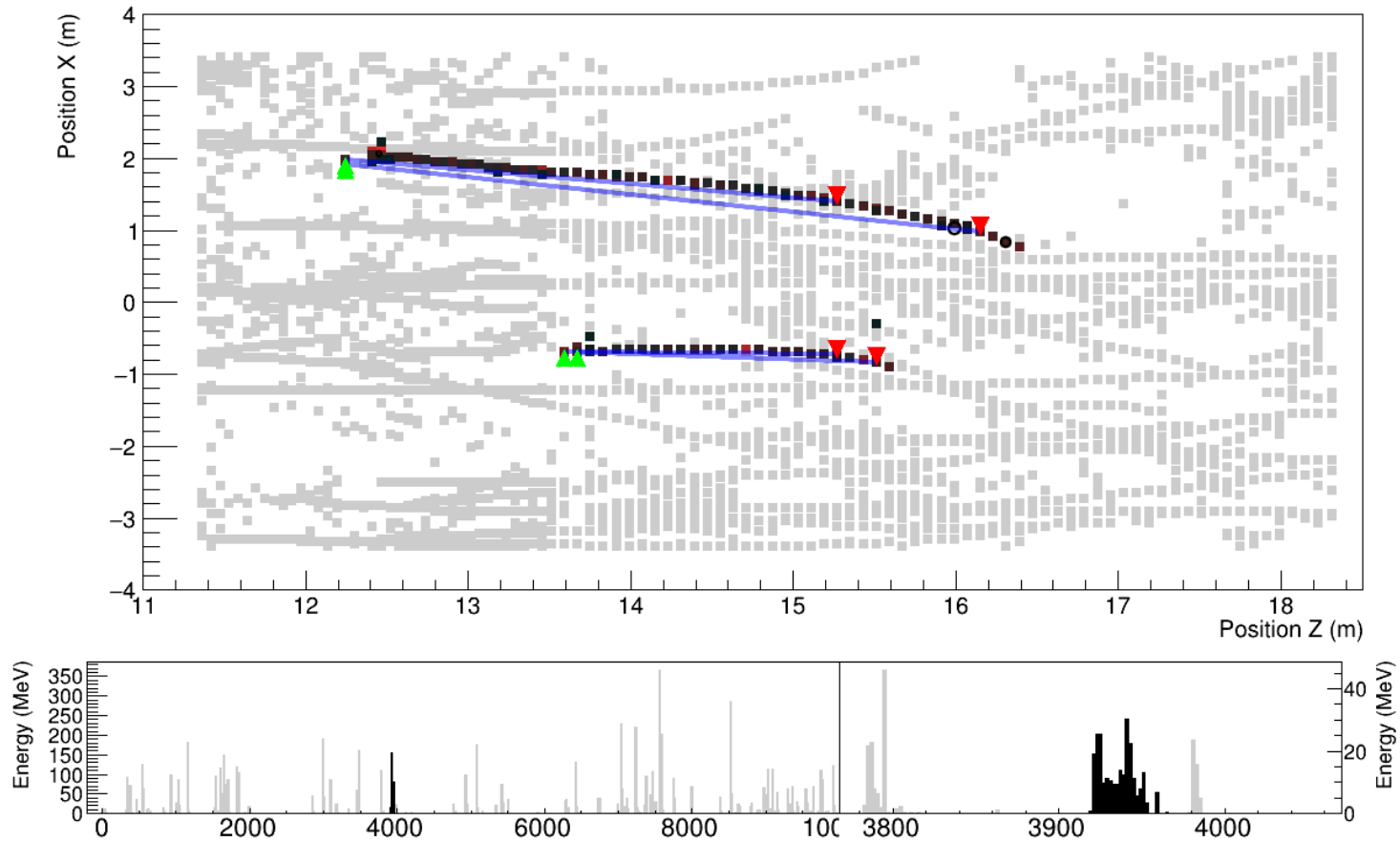
# Pileup?



# TMS starting event

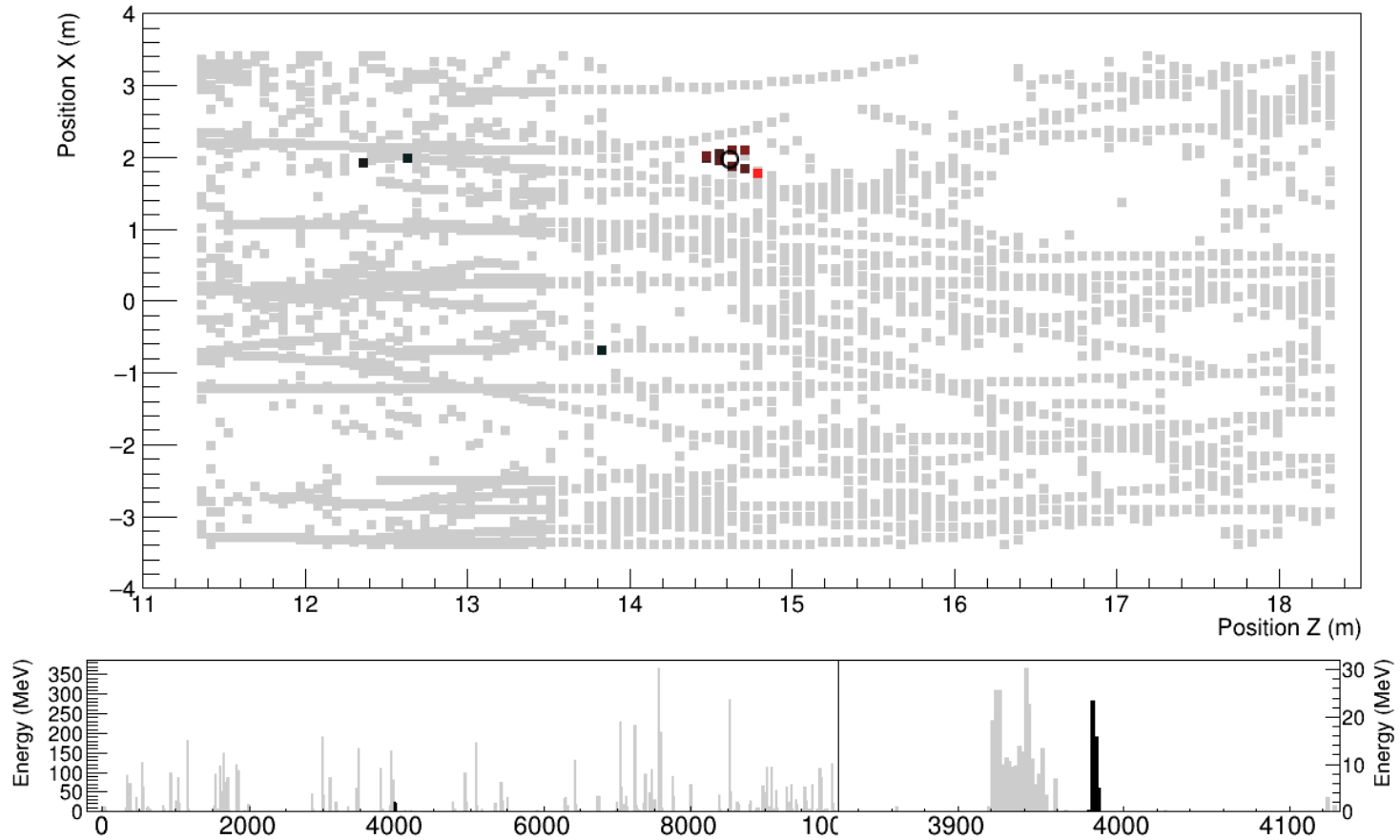


# Is this related to next? 1/2

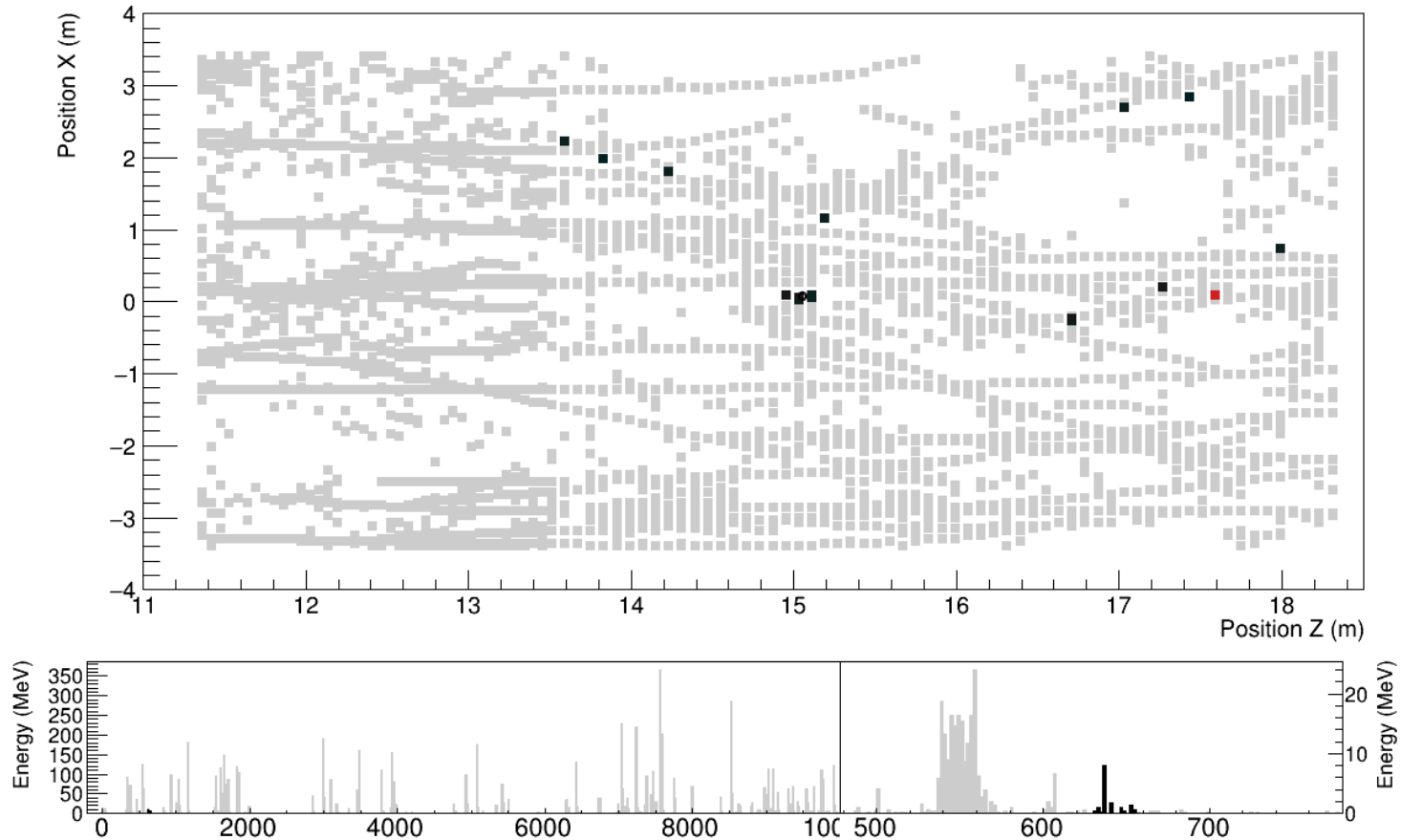




# Is this related to previous? 2/2



# What are these? - Too aggressive?

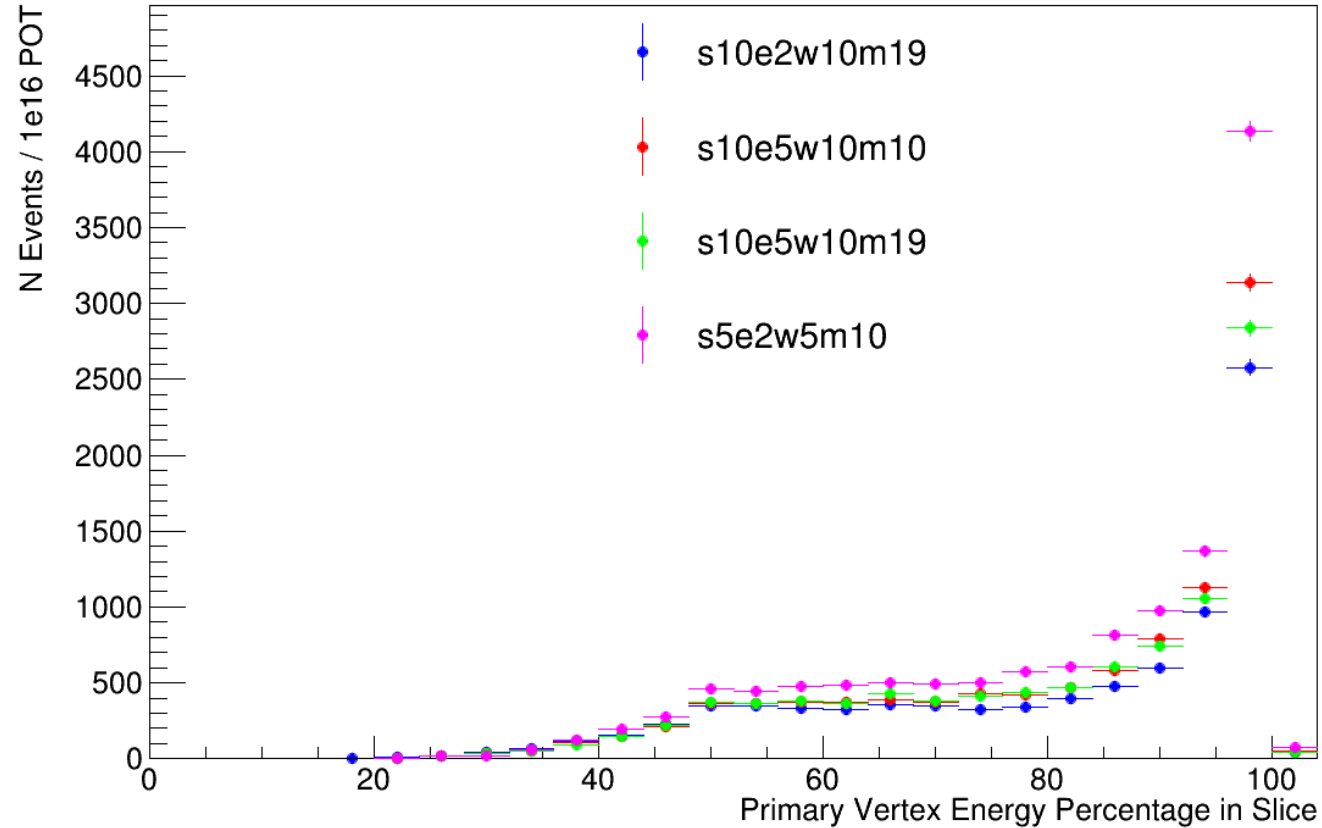


# Tuning parameters

- Starting energy threshold
- Ending energy threshold
- Window width
  - In principle there could be two, but only used a single one
- Minimum slice width
- Naming scheme
  - s10e5w10m19 = starting energy 10 MeV, ending 5 MeV, 10ns width, minimum 19 ns (distance between batches in flux)

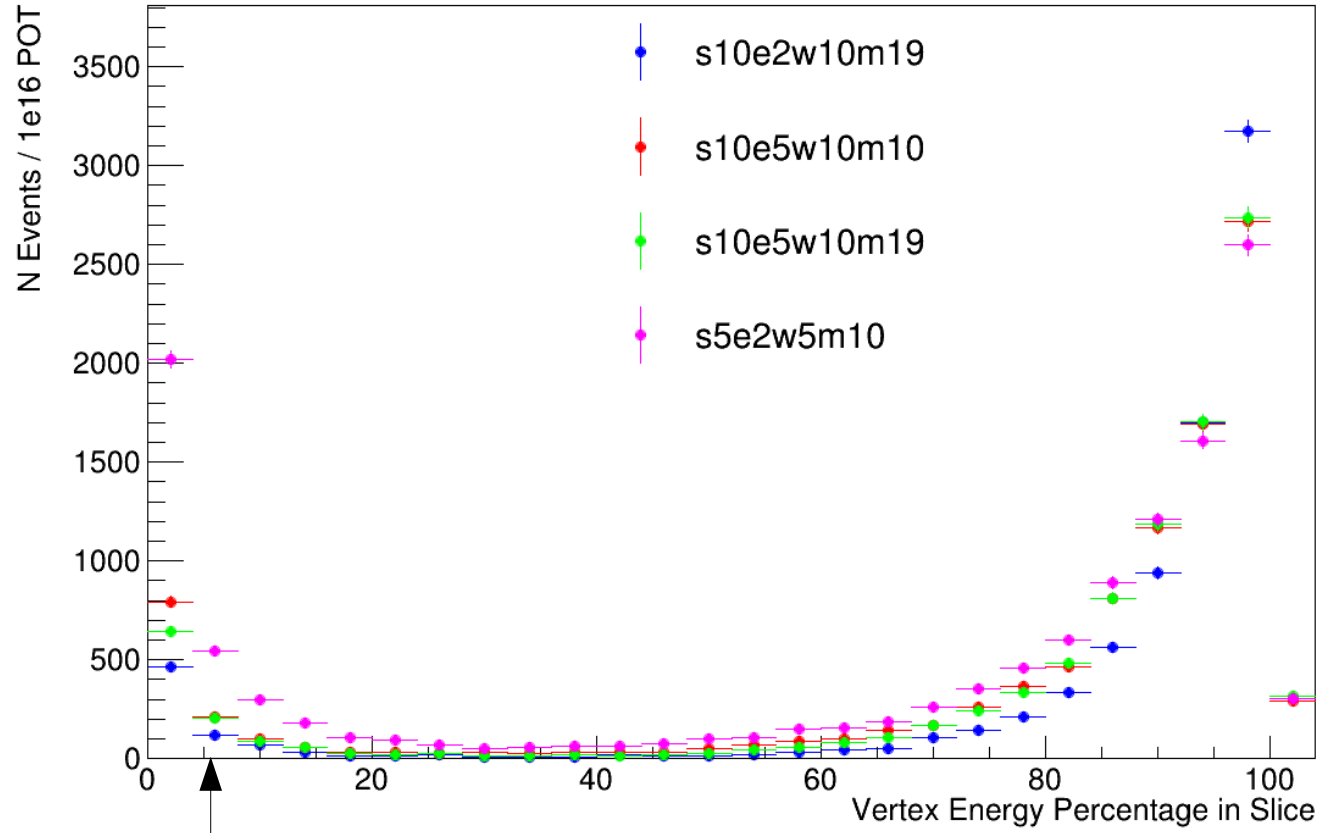
# Fraction of energy from primary vertex

ie purity



# Fraction of energy from primary in slice

ie efficiency



Cases where most of primary vertex energy was captured in another slice  
Like if split too aggressively or michel

# Todo improvements

- True primary vertex is vertex with most visible energy deposited in a slice
- Right now very little information is stored about
  - additional true vertices in a slice
  - cases where this slice's true primary vertex contributed to another slice
    - Like when you split a slice too aggressively
- Makes it difficult to measure true reco eff

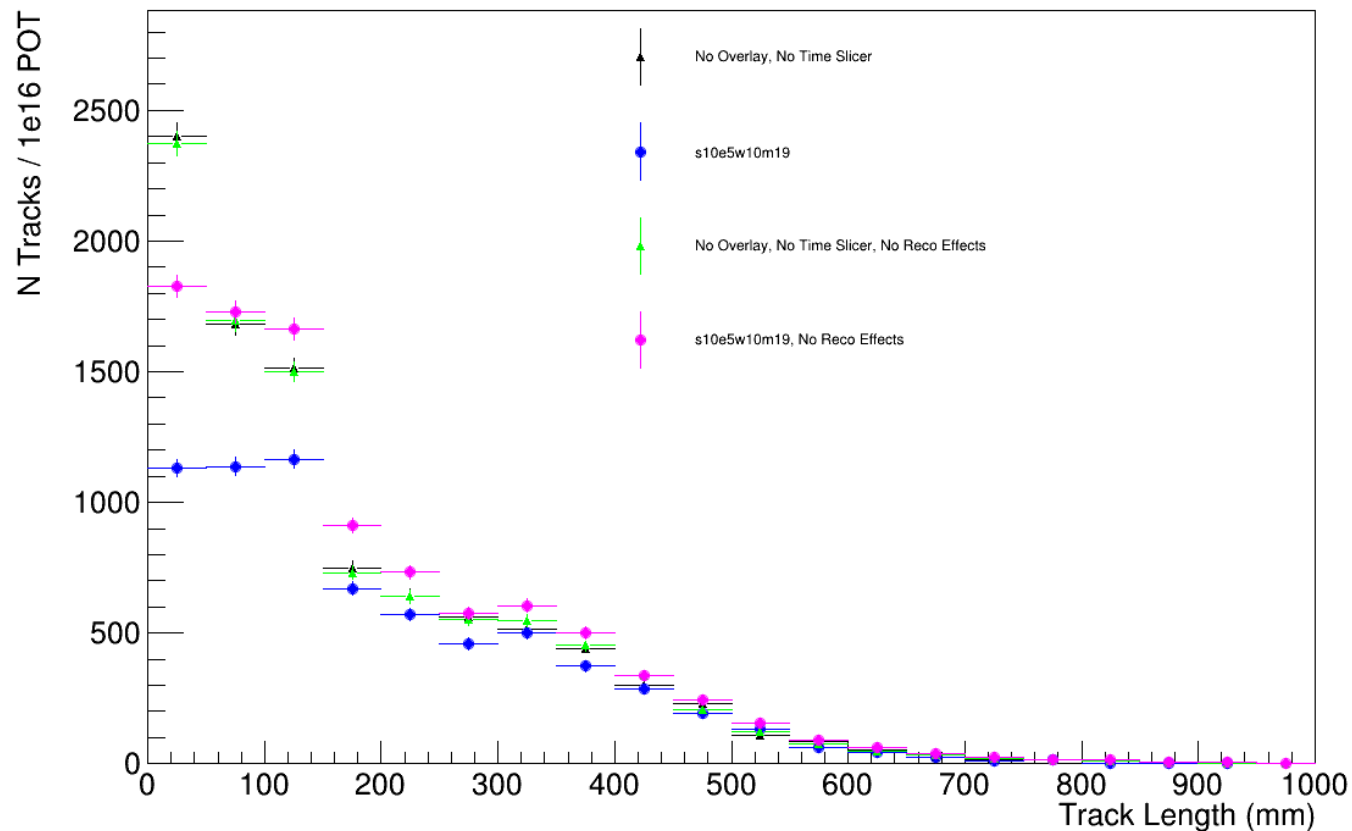
# Conclusion

- The timing reco effects have been added to simulation
  - Please let me know parameters, especially deadtime
  - Current plan is to separate detector simulation into its own program
- Time slicer added
  - Default option of s10e5w5m19 works reasonably well for now
  - Need to understand true reco cases, and various slices
  - What we're optimizing dictates choice of slicer parameters
- The next step would be to add slices to downstream CAFs

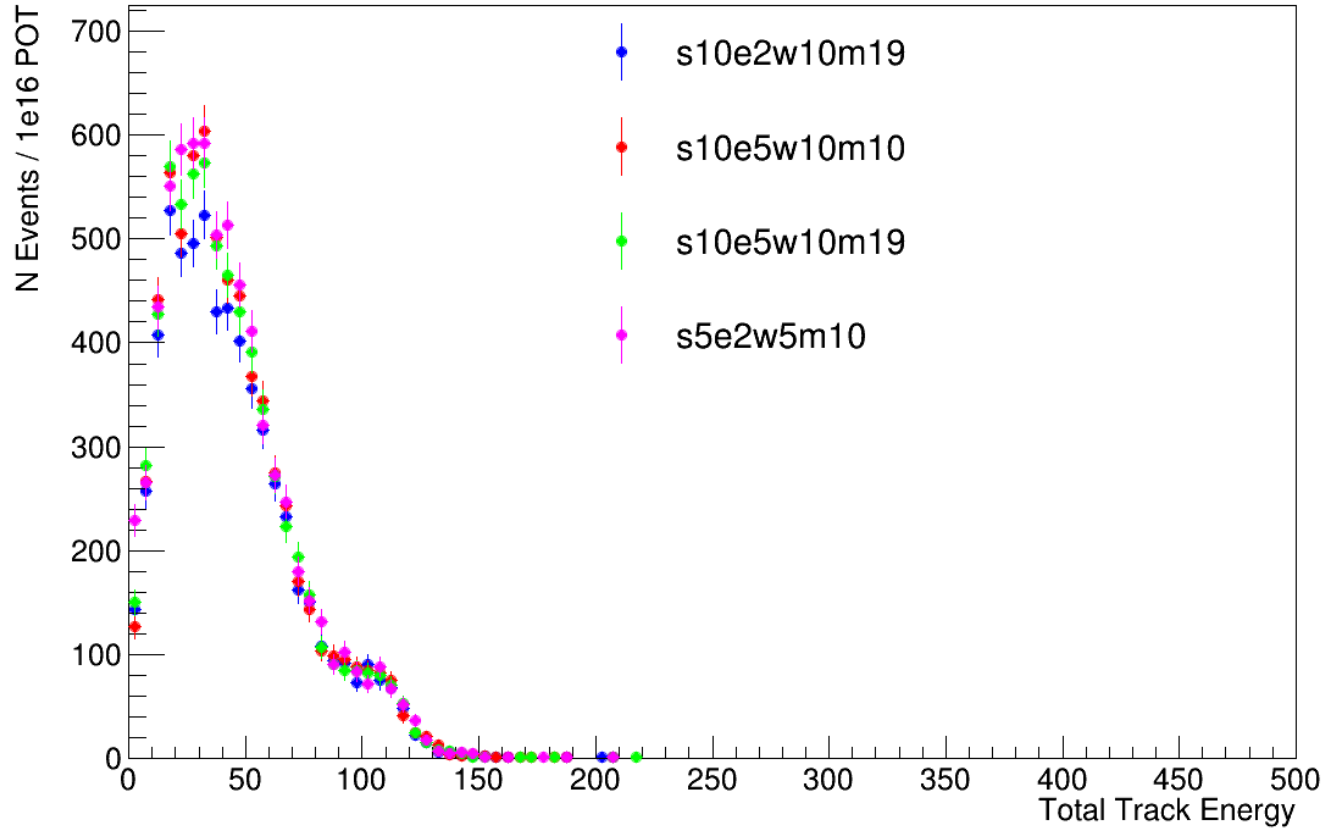
# Backup



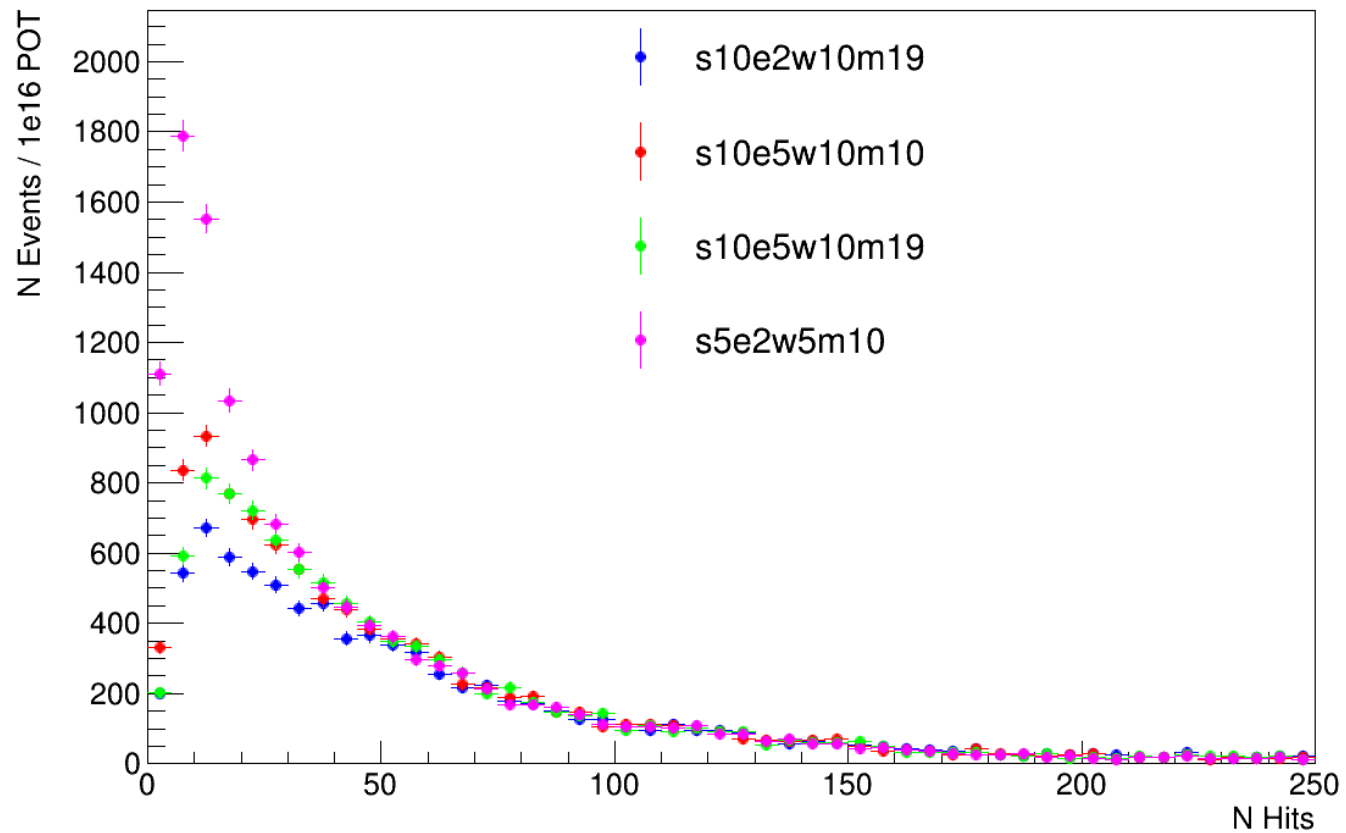
# With and without timing reco effects



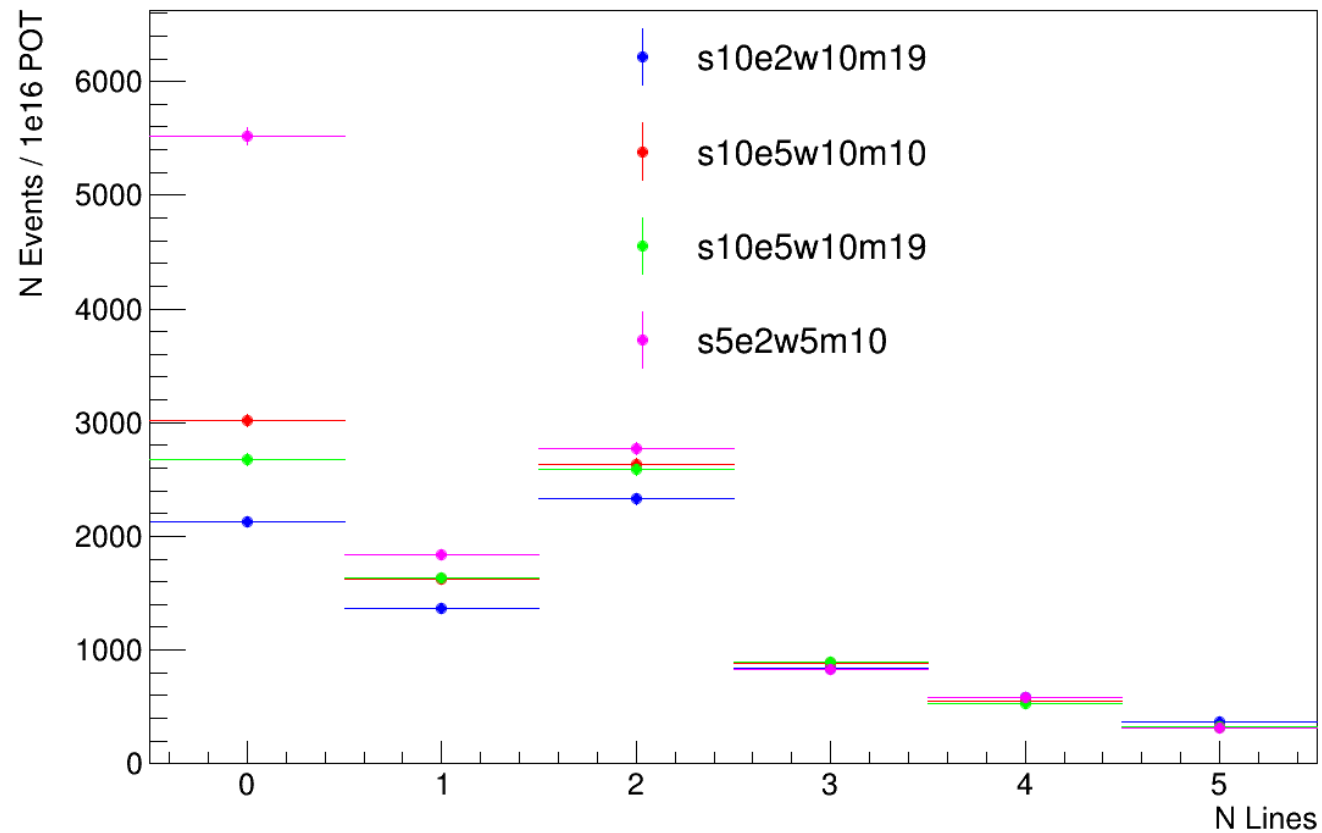
# Total Track Energy (MeV)

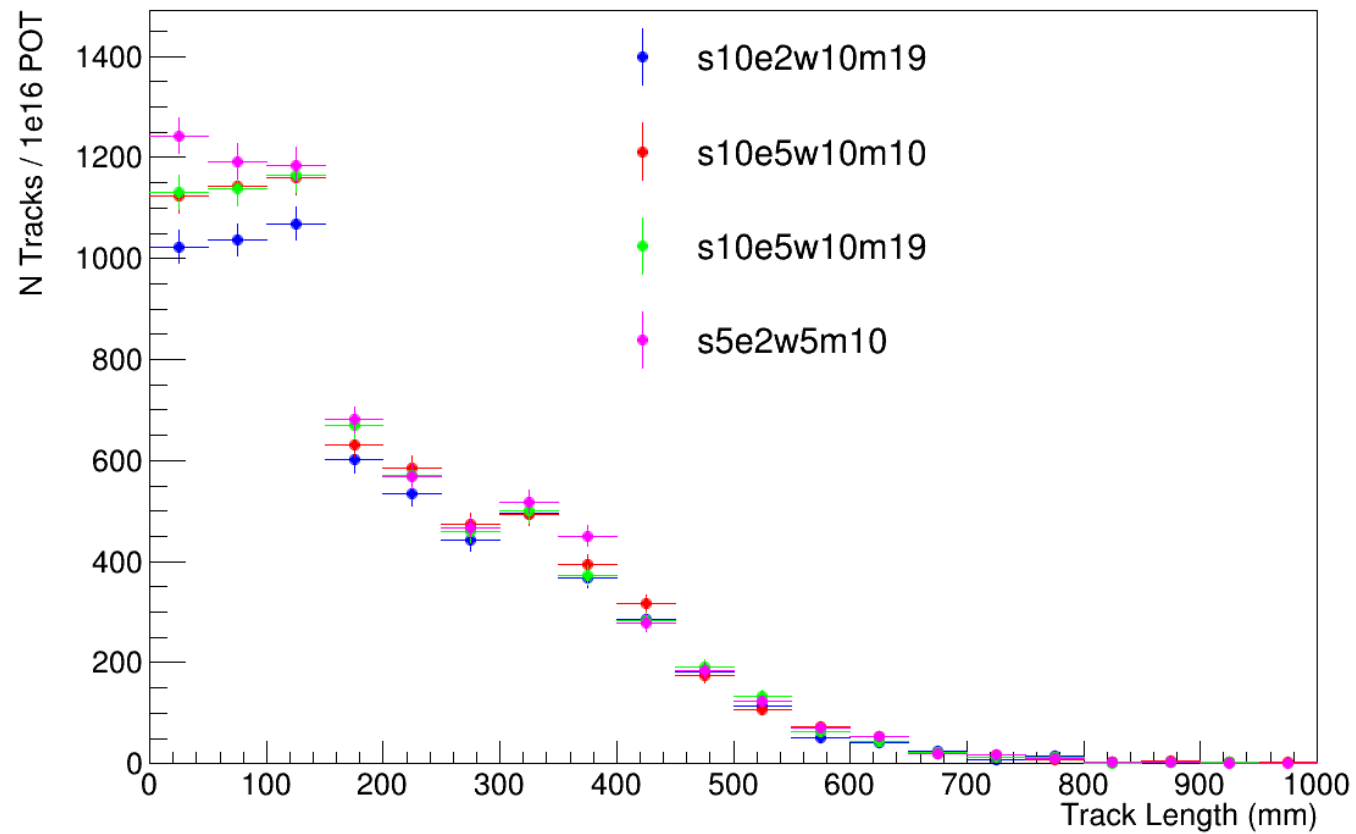


# N Hits

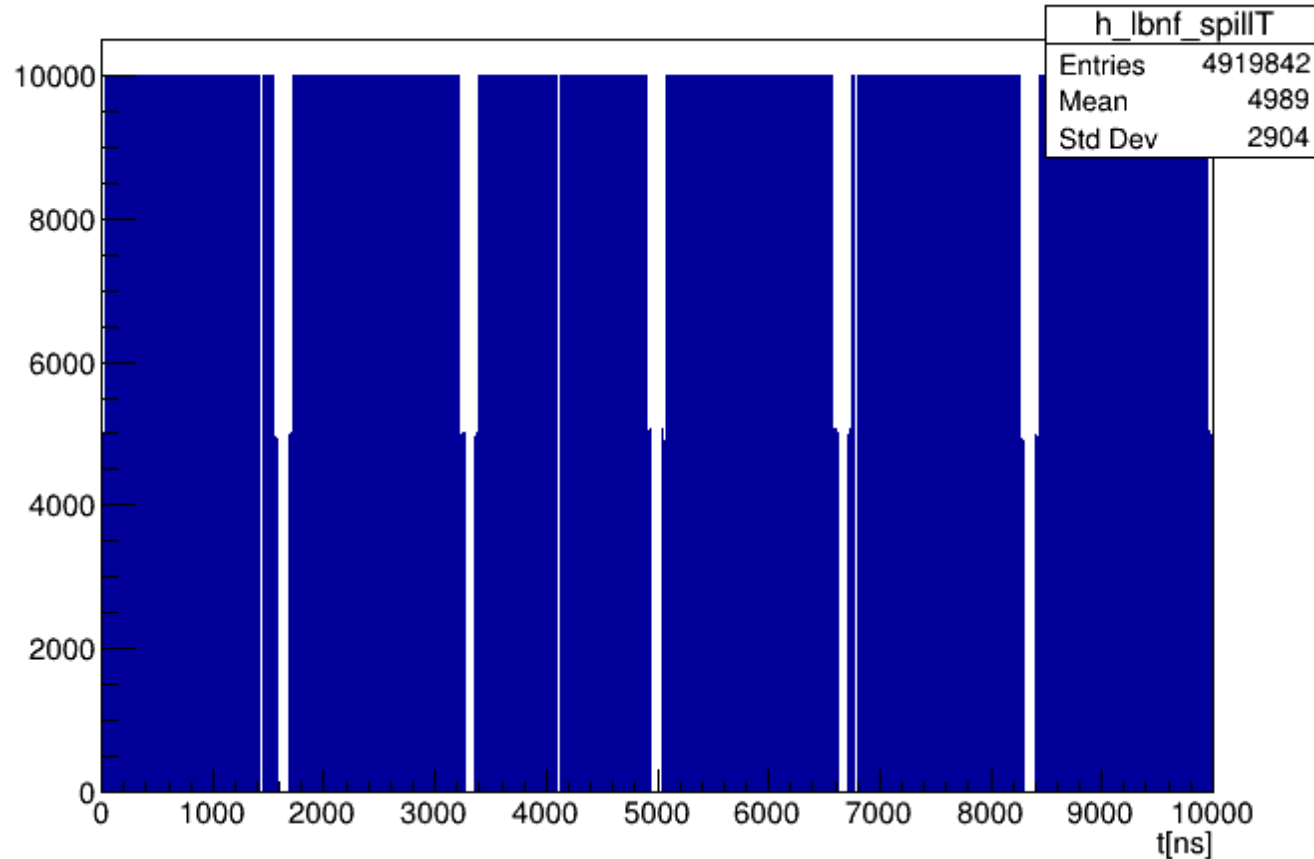


# N Lines

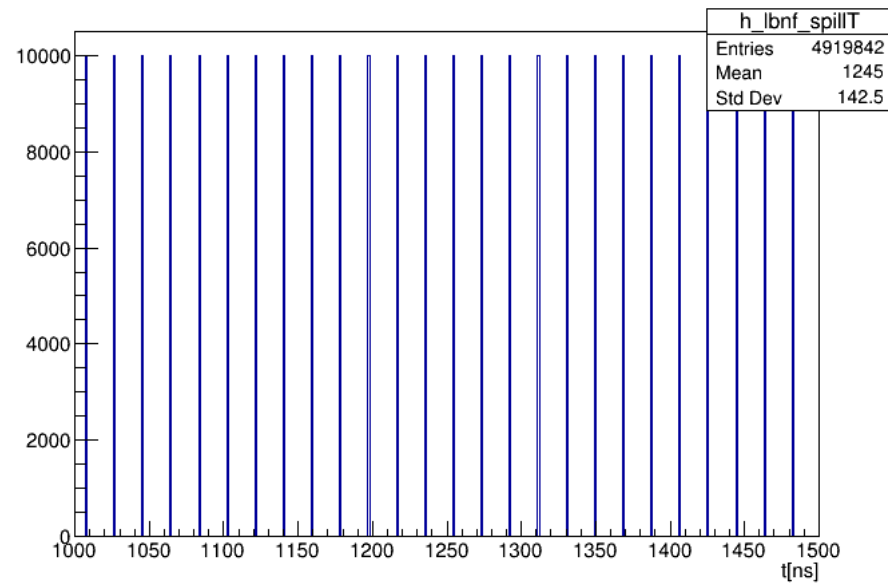
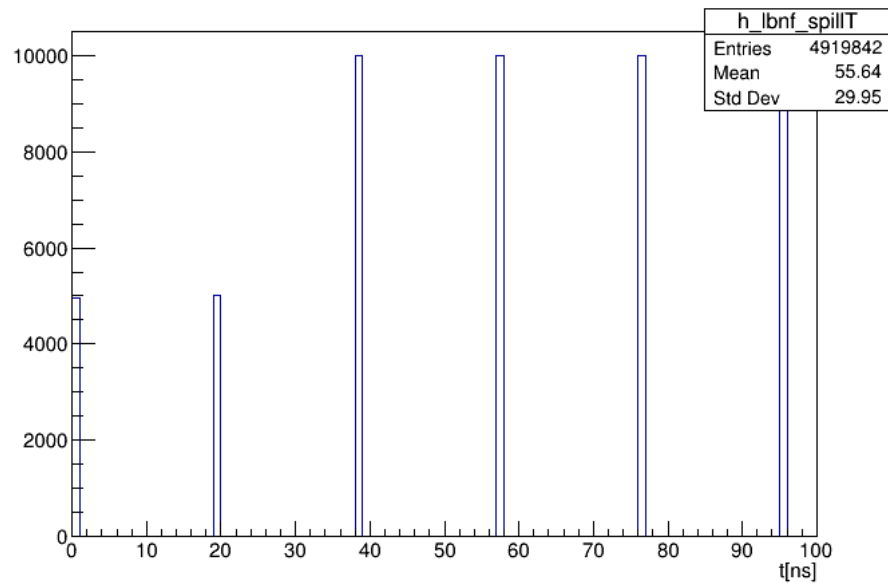




# Beam Spill Structure



# Beam Substructure



# EDepSimSpillTimeFactory.cc

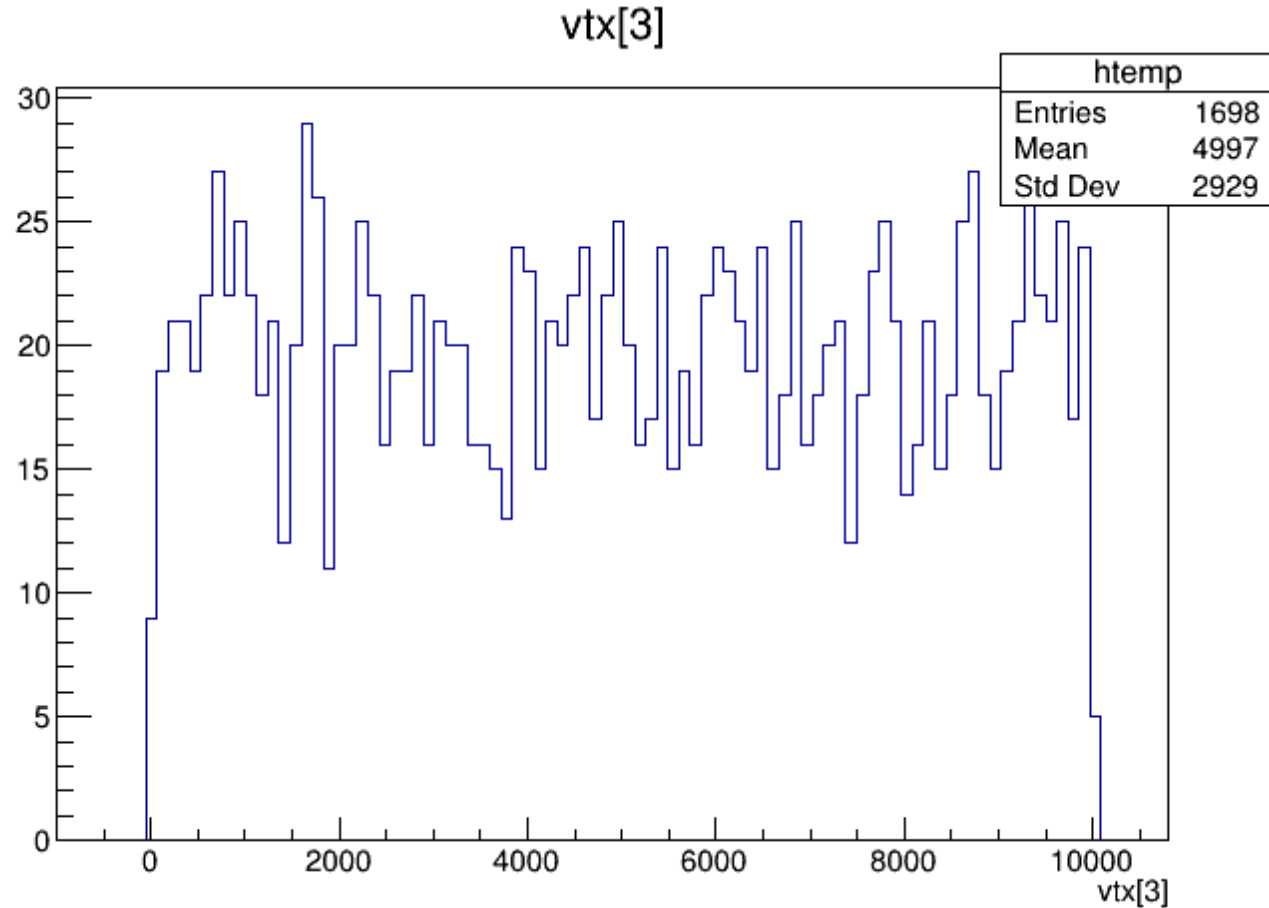
- Edep sim has “factories” that allow you to event rates and event times via config file
- For pileup simulation, we setup the “mean” factory with a mean number of events per
  - By default all event times are 1ns
- But we can change to “spill” factory, which randomly picks from a spill structure

- In theory we can set the individual bunch powers but do we really want 1000 different config lines to match exactly?

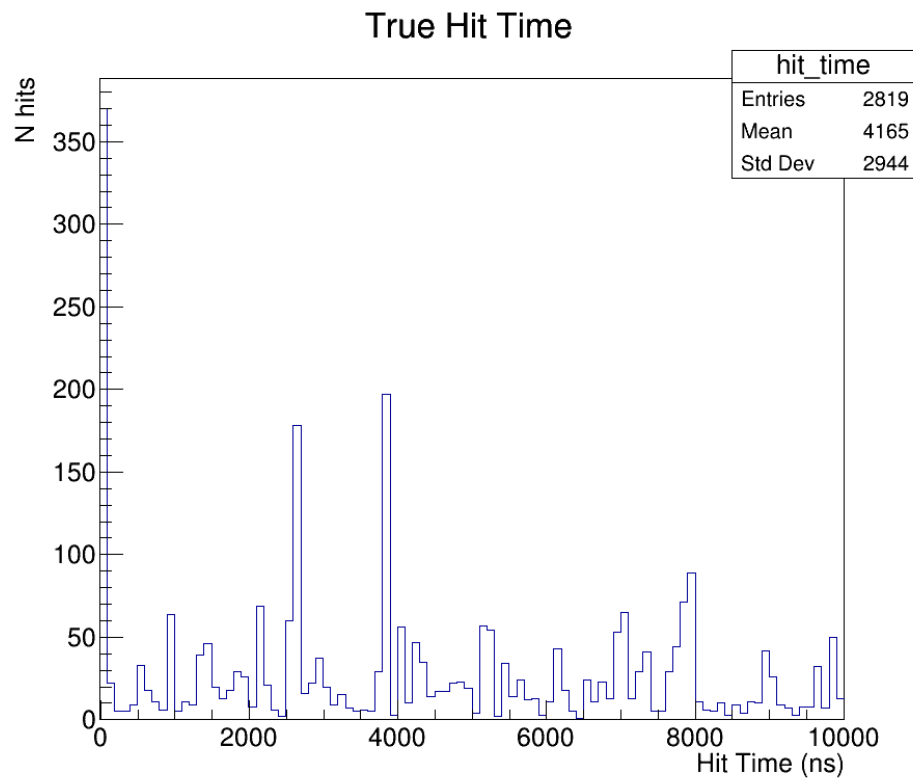
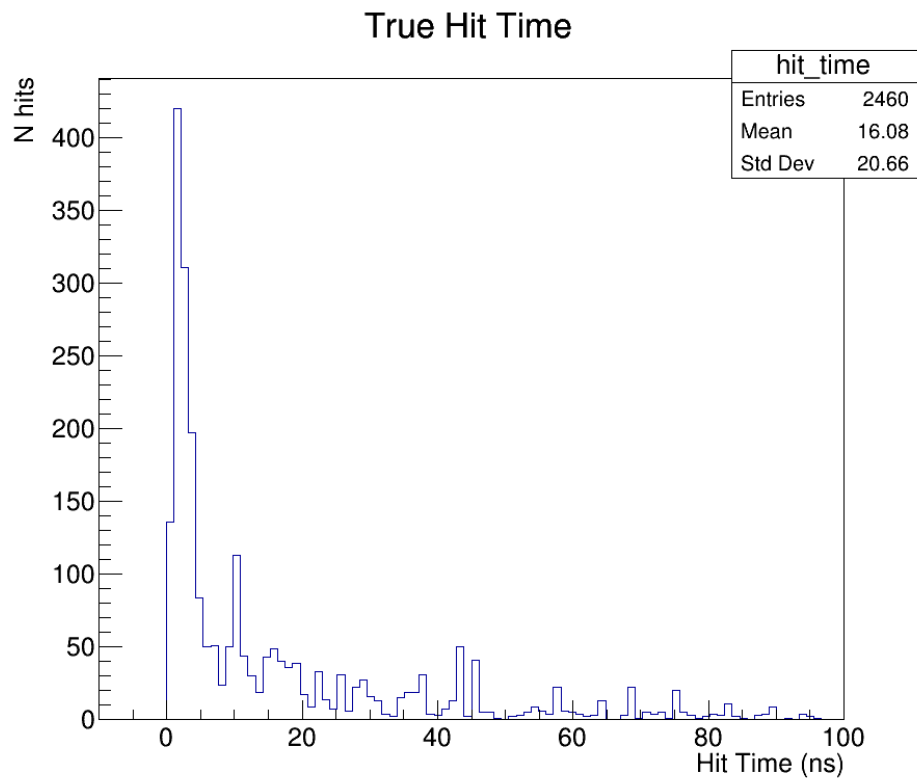
In edep sim's config file,  
/generator/time/spill/start 0 ns  
/generator/time/spill/bunchSep 10 ns  
/generator/time/spill/bunchLength 5 ns  
/generator/time/spill/bunchCount 1000  
/generator/time/spill/bunchPower 1 0.5  
/generator/time/spill/bunchPower 2 0.5  
/generator/time/set spill



# Example Event Rate

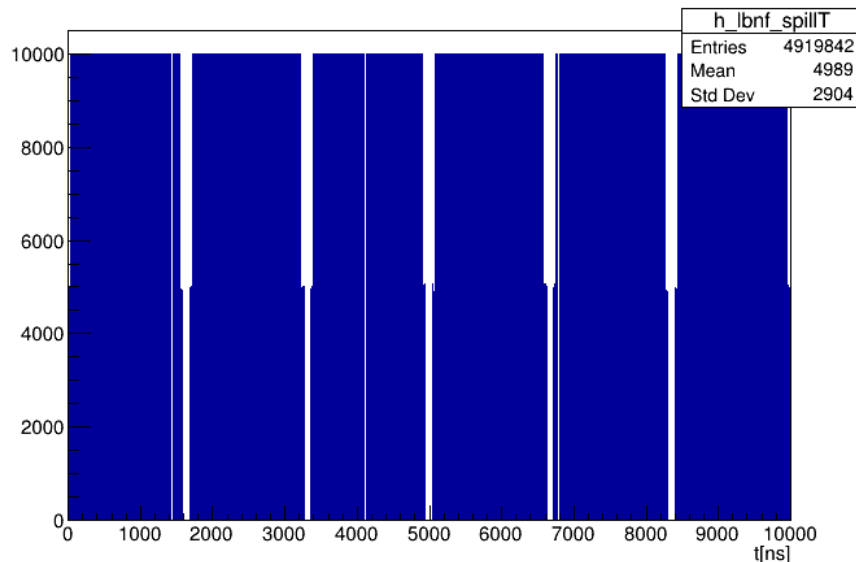


# Before/After



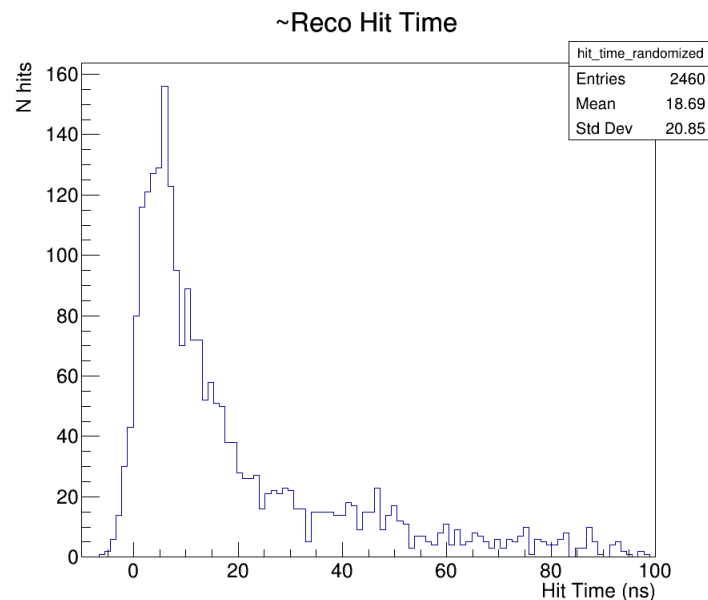
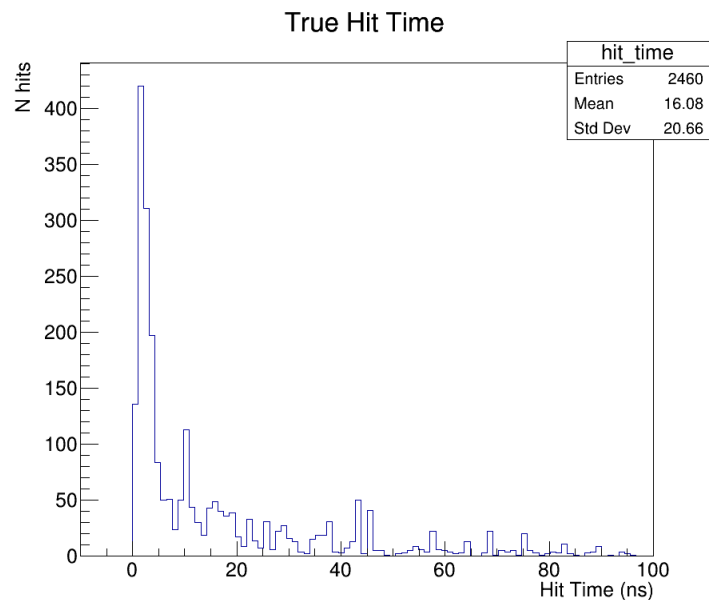
# Kevin Wood's Code

- I didn't know about this code
  - [https://github.com/krwood/ND\\_Production/blob/feature\\_spillsFromEdepSim/geant4/overlaySinglesIntoSpills.C](https://github.com/krwood/ND_Production/blob/feature_spillsFromEdepSim/geant4/overlaySinglesIntoSpills.C)
- Does a similar thing to edep sim's time factory but as a second program
  - But also the rest of pileup simulation
  - I like that edep sim separates N events and event times into two separate steps
    - We can simulate 1 event with random spill time, or 10 events, instead of just Poisson random
- In theory I can implement this
- One advantage is that the hit time is chosen randomly from the hit time histogram, automatically giving us the correct power as function of time.
  - Ideally I'd like this function to be integrated into edep sim instead

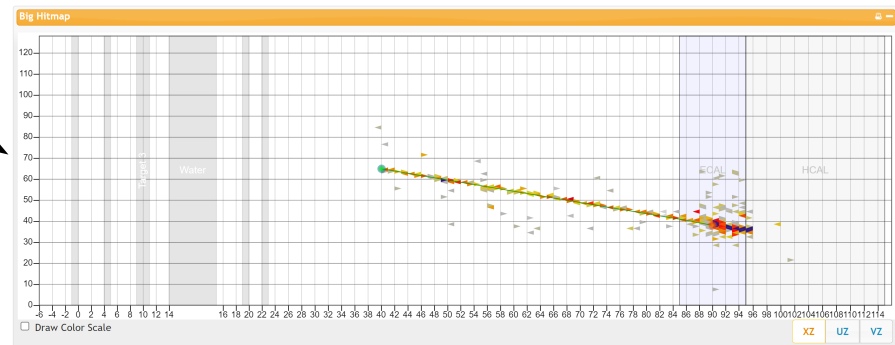
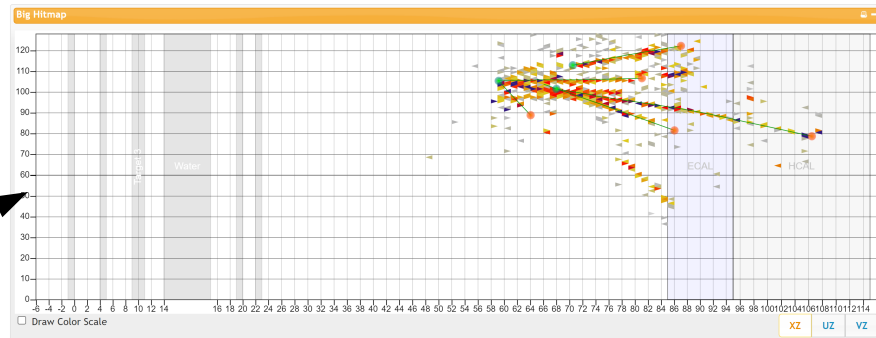
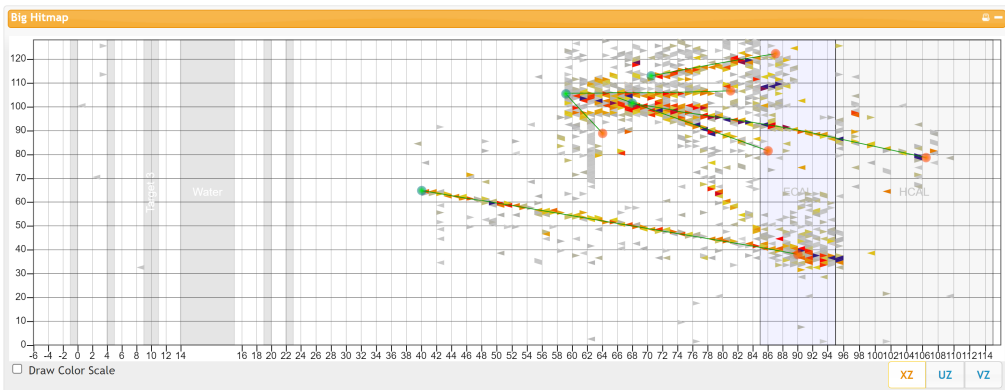
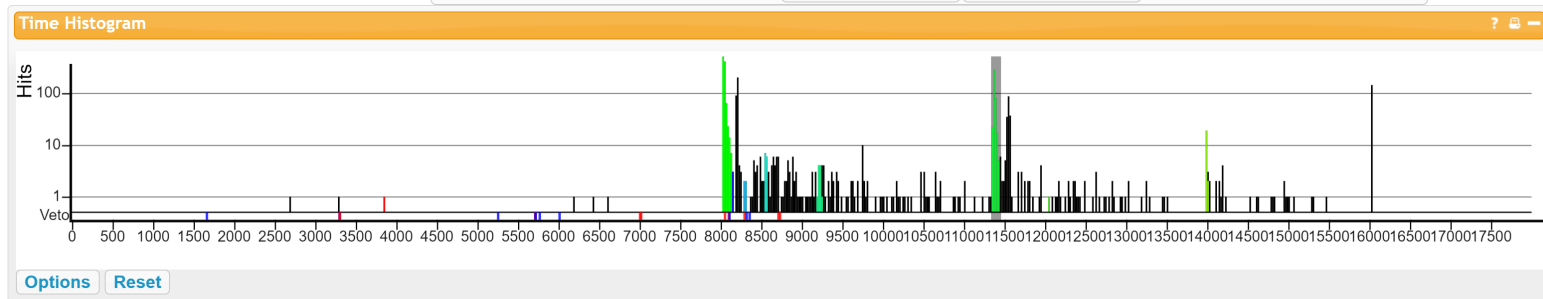


# Spill simulation conclusions

- Now spill times are randomized
- This is also the case for the liquid argon section
- The next step is setting up a time slicer for TMS
- Eventually this will require reco effects

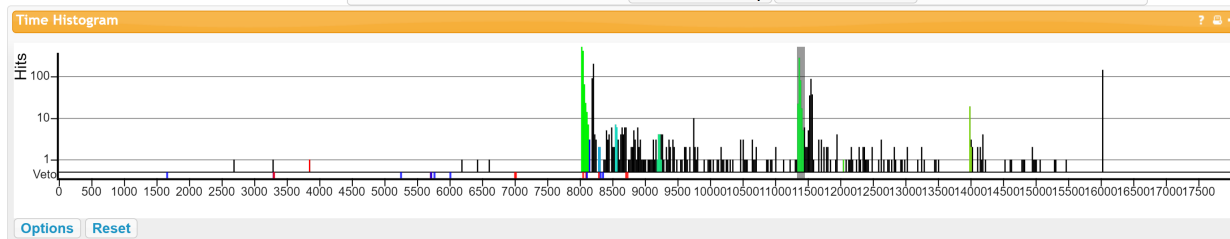


# Using Time to Figure Out Pileup



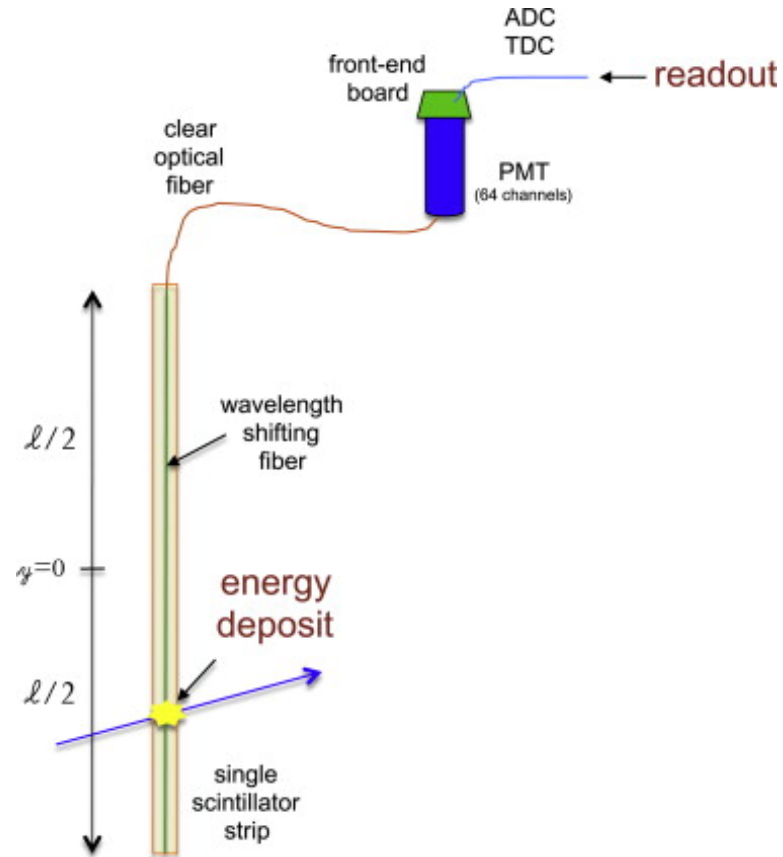
# Time Slicer Algorithm

- I'm familiar with Minerva's time slicer
- If energy within a small window is  $> \text{energy\_threshold\_1}$ , start adding hits into a slice
- Continue adding until energy in window  $< \text{energy\_threshold\_2}$ .
- Optimize to
  - Maximize energy fraction of single event
  - Minimize energy from secondary events



# Backup

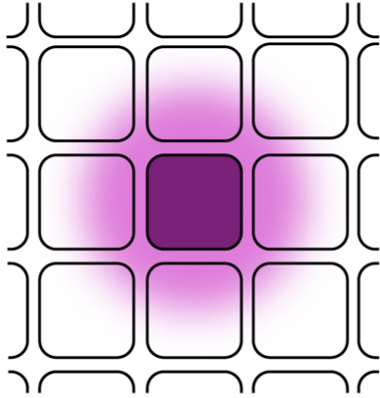
# Optical Model





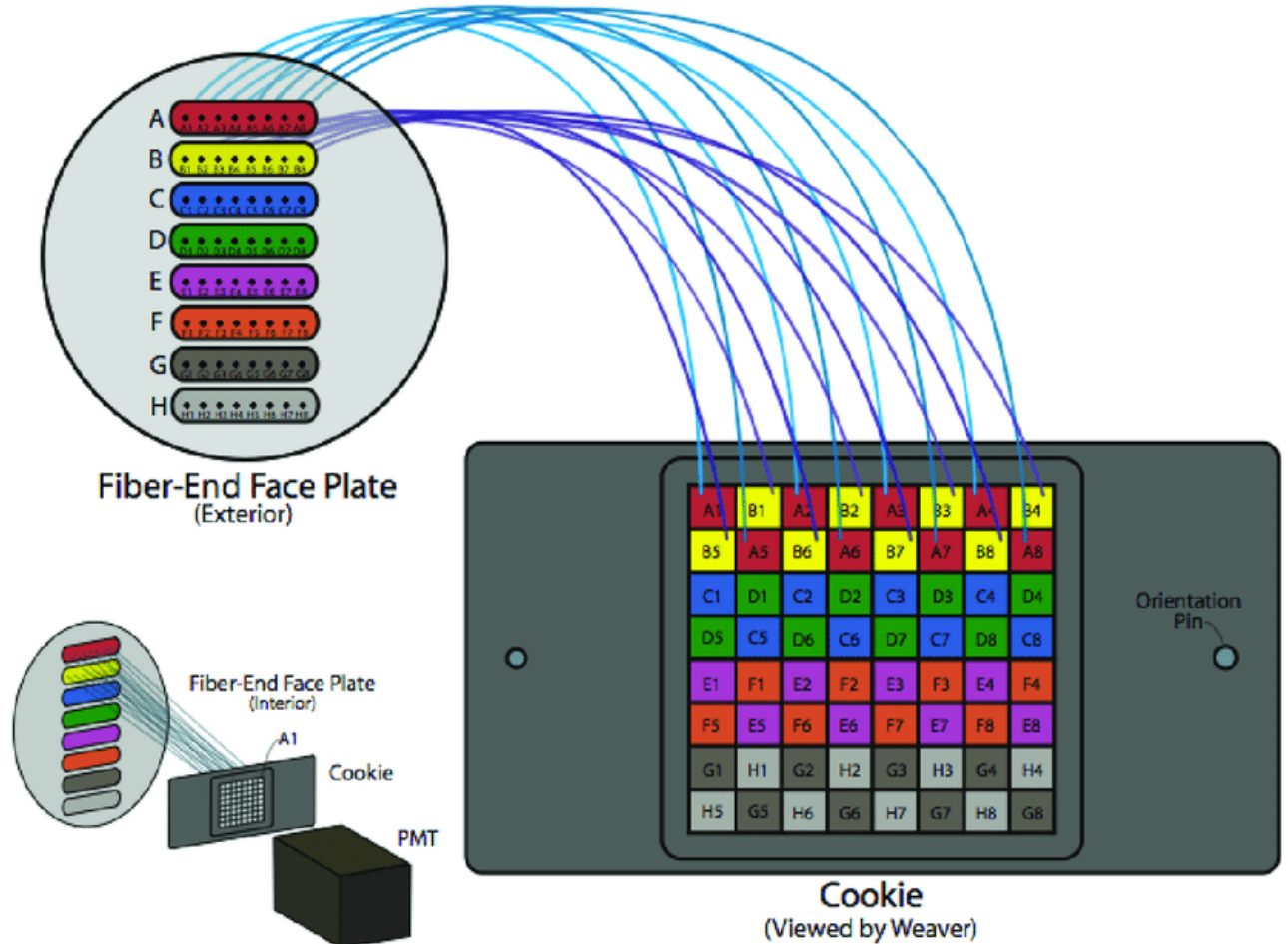
# Cross Talk

High cross-talk



Shared well walls

3-5% effect



# NuMI beam batch structure

