Study of peak finding algorithm

Guang Zhao*, Shuiting Xin*, Siman Liu+, Linghui Wu*, Mingyi Dong*, Shengsen Sun*, Weimin Song+

zhaog@ihep.ac.cn

* Institute of High Energy Physics, + Jilin University

Cluster Counting Meeting

Nov. 25th, 2021

Outline

- **Introduction to the peak finding algorithm**
- **Parameter optimizations**
- **Summary**

Introduction

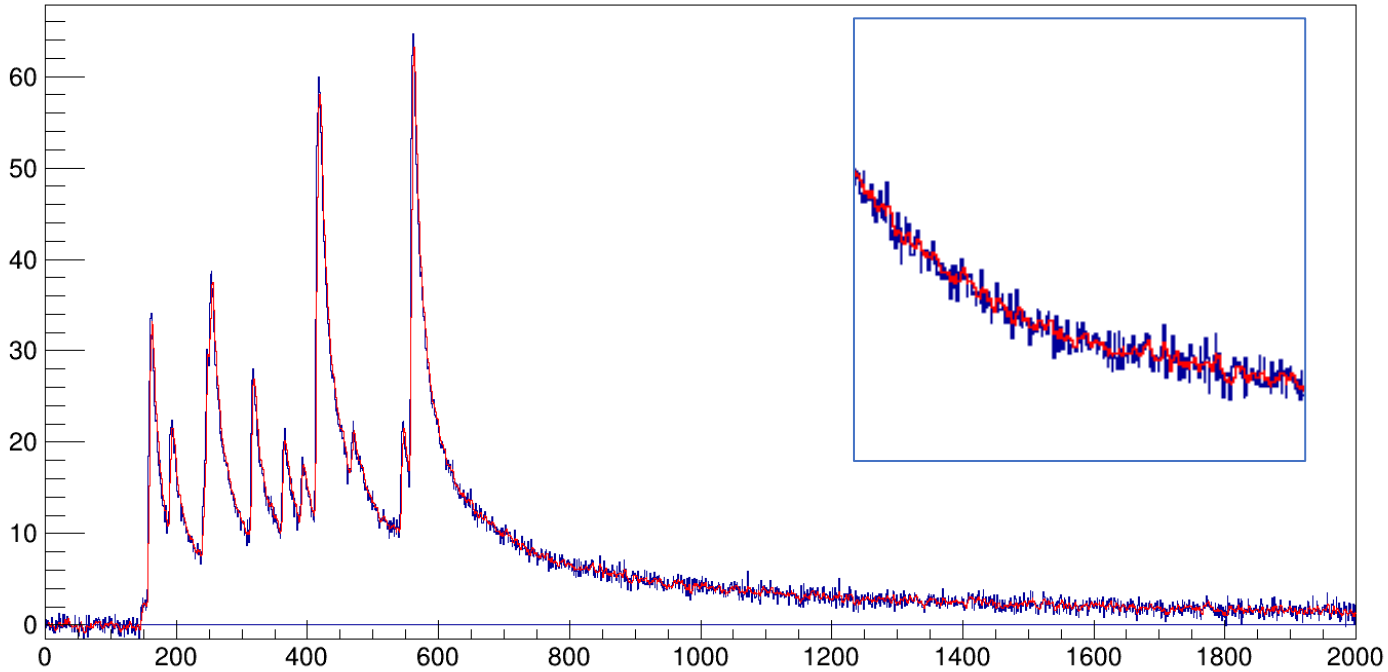
- **Peak finding is essential for ionization measurement using cluster counting technique**

- **Requirement: fast and efficient**
  - Fast: Data size of waveform is huge. Fast online algorithm at the front-end is recommended
  - Efficient: Good ability to recover pile-up. High pulse detection efficiency.

- **Peak finding with derivatives is a simple and classic algorithm that may satisfy the speed and performance requirements**
  - e.g., similar algorithm implemented for PANDA EMC

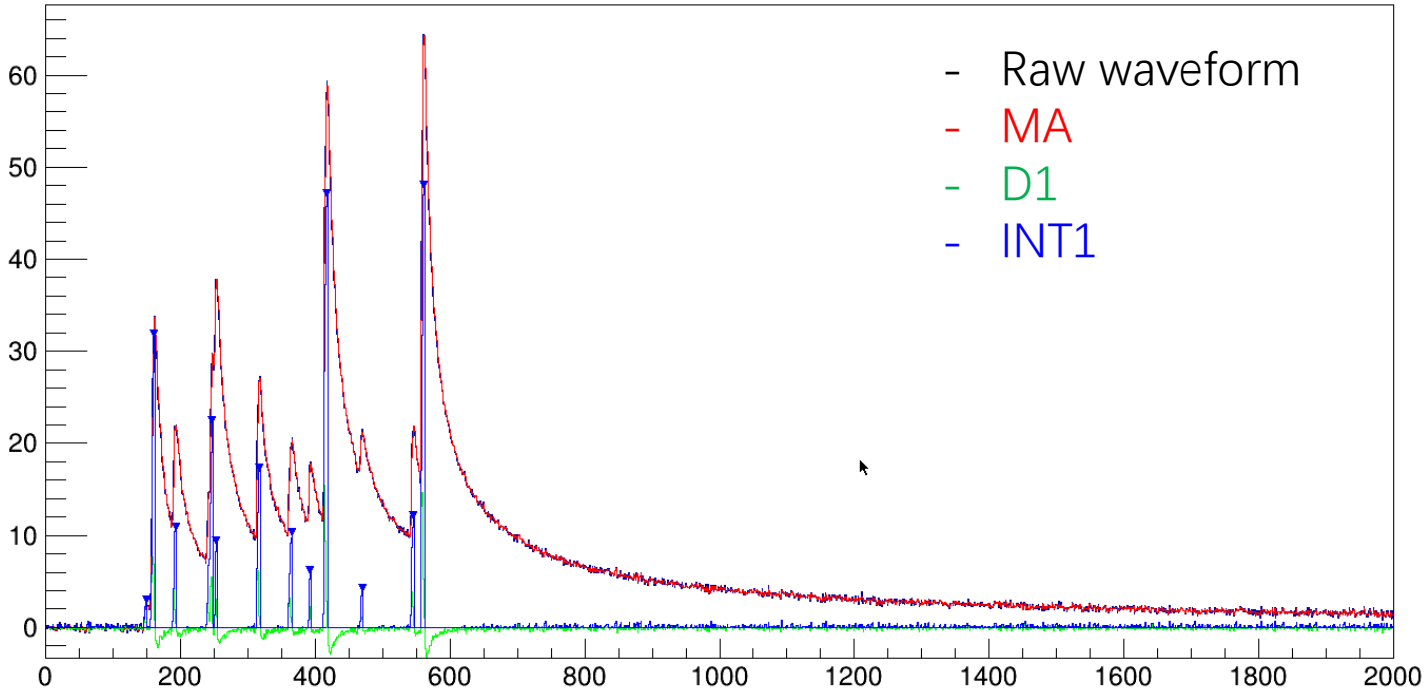
Peak finding algorithm

- **Noise reduction**
  - Filter out high frequency noises in the waveforms in order to improve the S/N ratio
  - Moving average: $\mathbf{MA[i]} = \frac{\mathbf{1}}{\boldsymbol{M}} \times \sum_{\boldsymbol{k=0}}^{\boldsymbol{K<M}} \boldsymbol{S[i-k]}$



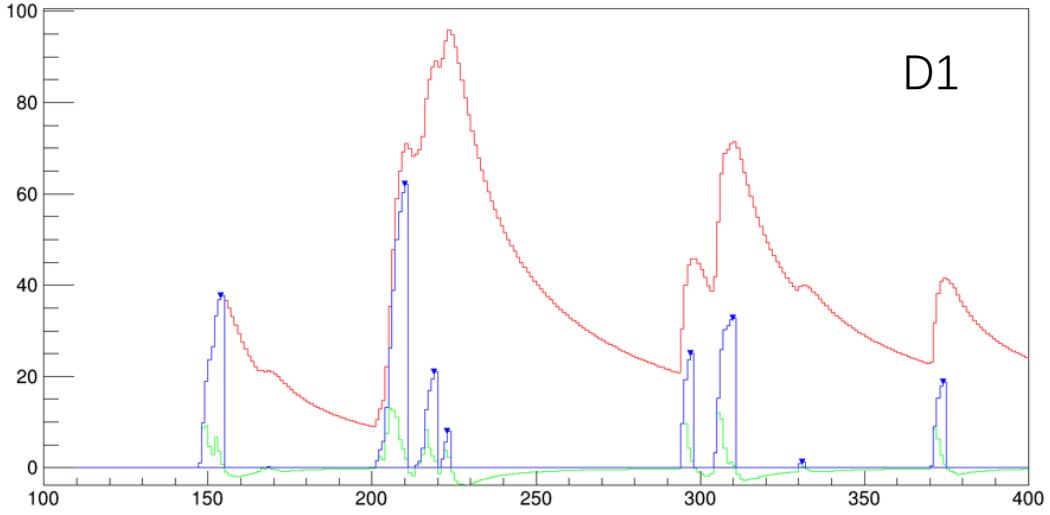
Cluster counting algorithm (II)

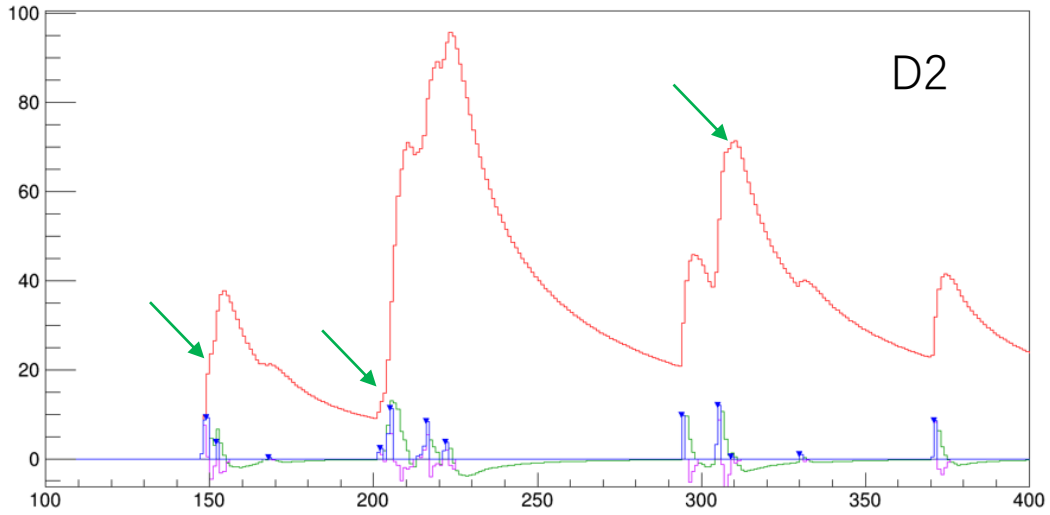
- **First derivative and integration**
  - First derivative (**D1**): D1[i] = MA[i] – MA[i – 1]
  - Integration on the positive D1 (**INT1**): recover the rising edge and removing falling edge
  - Hit detection: Passing a threshold



Cluster counting algorithm (III)

- **Second derivative and integration: recover pile-up peaks on the rising edge**
  - Second derivative (**D2**): D2[i] = INT1[i] – INT1[i – 1]
  - Integration on the positive D2 (**INT2**)
  - Hit detection: Passing a threshold



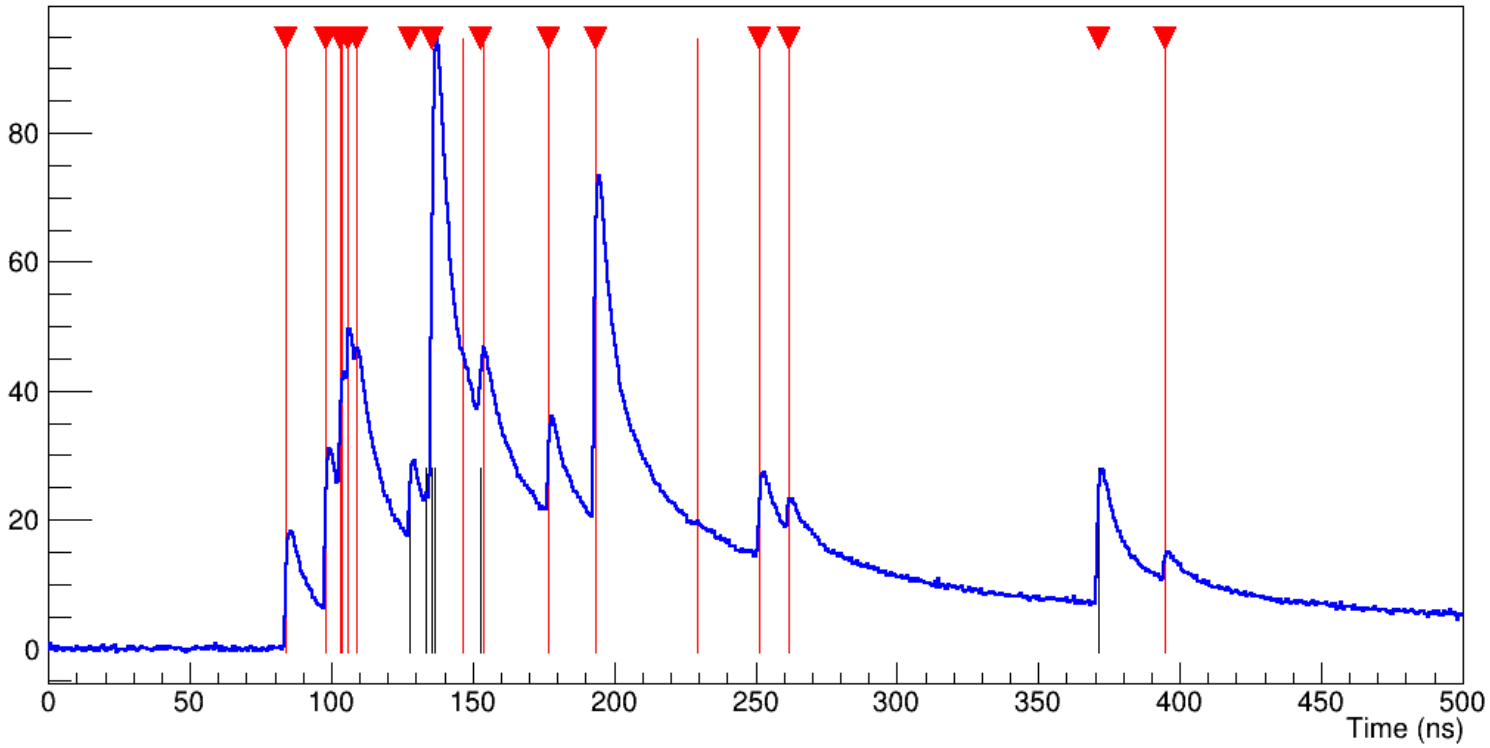


Pile-up on the falling edge is easier to recover. However, it is not the case for pile-up on the rising edge.

Optimization

- **Detection threshold**
  - Minimize the fake rate
- **Moving average size**
  - Maximize the counting efficiency
- **Simulation setup**
  - Gas: 90% He + 10% iC_4H_{10}
  - Particle: 10 GeV/c pions
  - Time constant: 1 ns
  - Noise level: 2%

Waveform with “MC truth” times



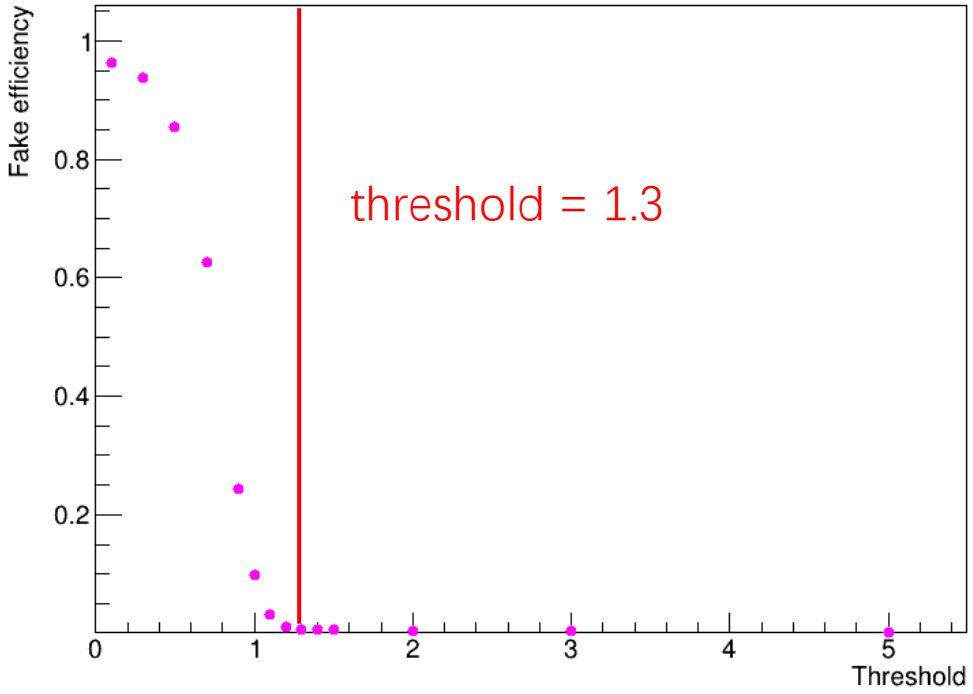
- Red line: primaries
- Black line: secondaries
- ➢ Triangle: detected

The MC truth matching algorithm

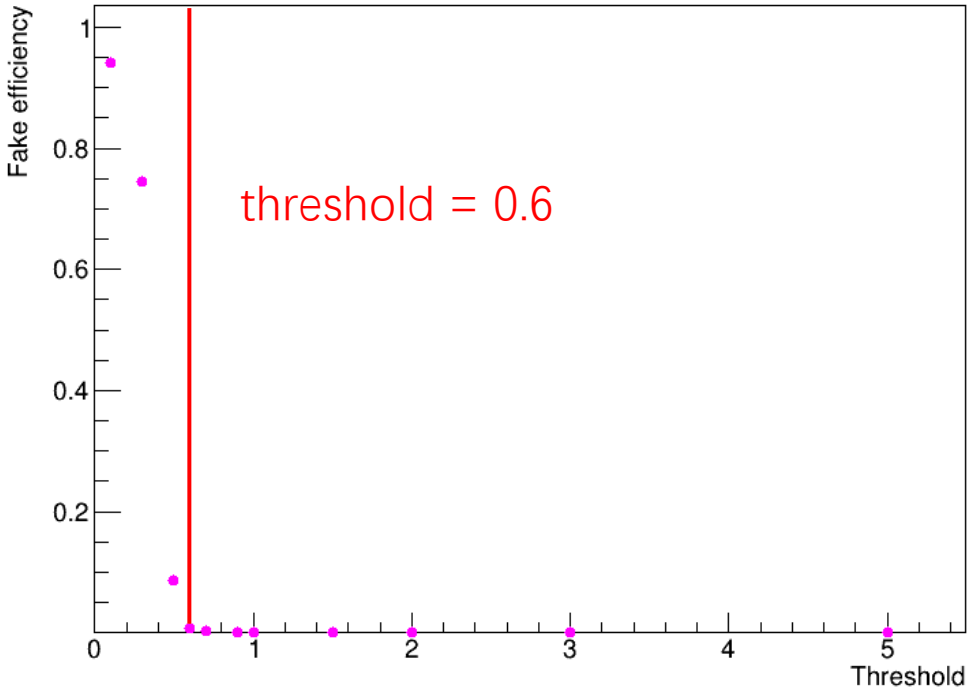
- **A detected time is matched to a truth time if**
  - $|T_{det} - T_{truth}| < 2$ ns
- **If the matched truth time is from a primary electron, the detection is defined as a "primary"**
- **If the matched truth time is from a secondary electron, the detection is defined as a "secondary"**
- **If a detection is both a primary and secondary, "primary" is set**
- **Otherwise, "fake" is set**

Fake rate vs. threshold

MA Size = 1

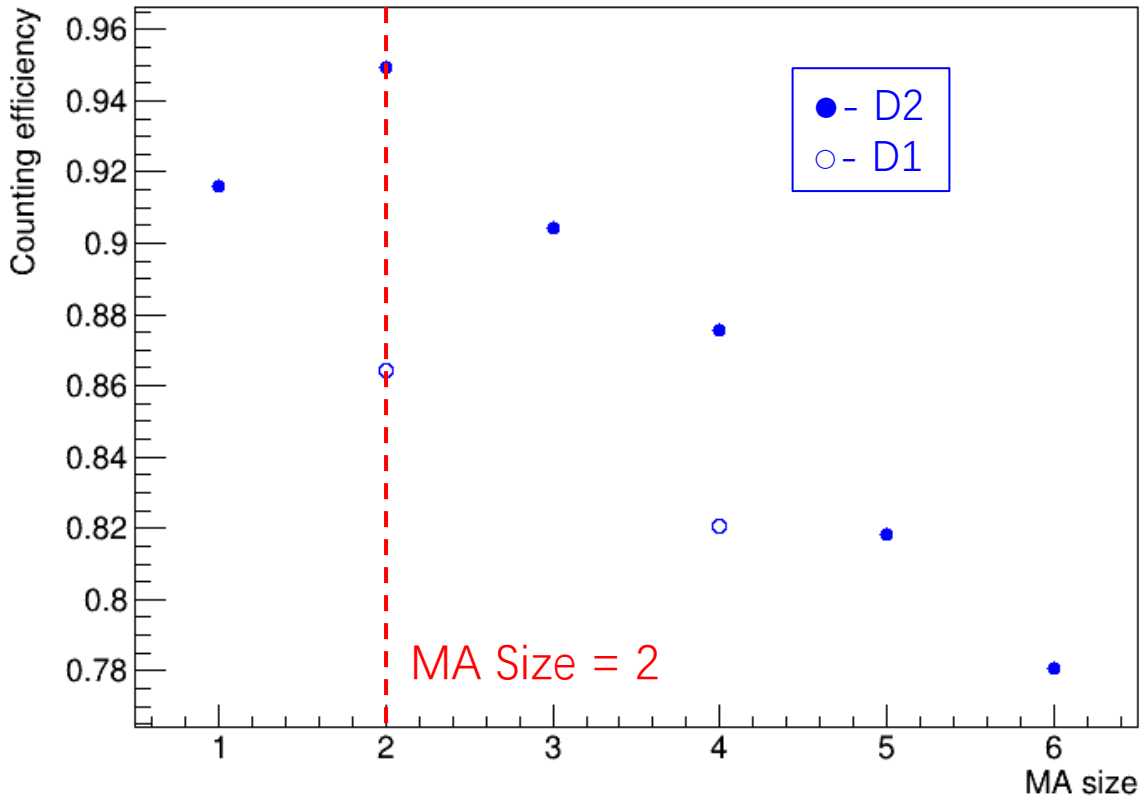


MA Size = 2



Efficiency vs. MA size

Apply thresholds determined in the previous slide

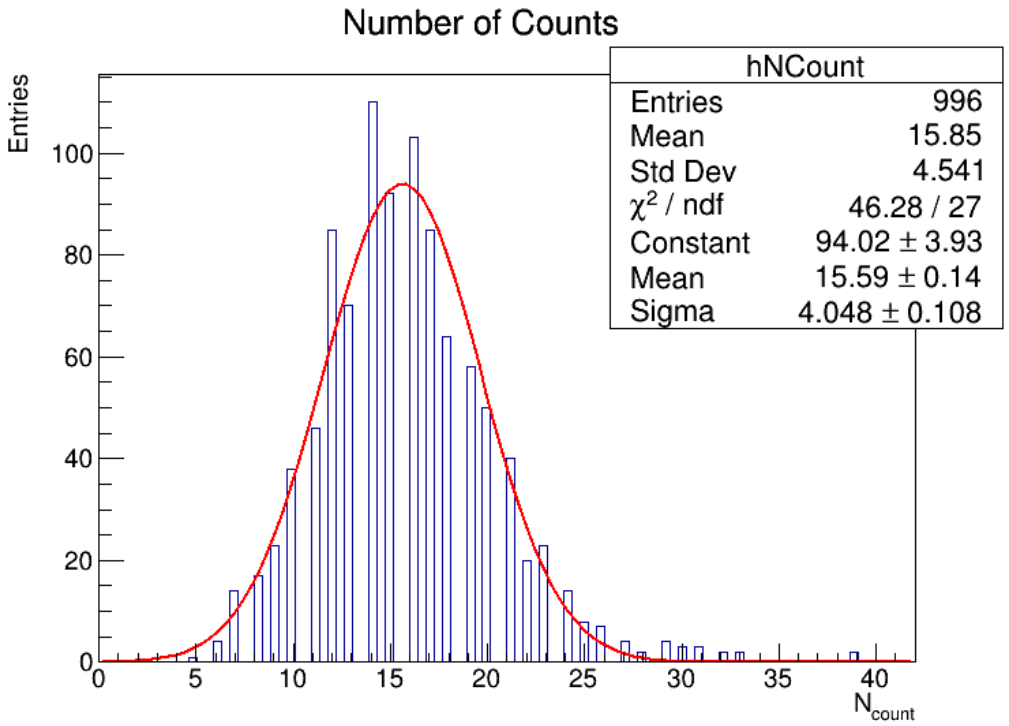


Optimized parameters:

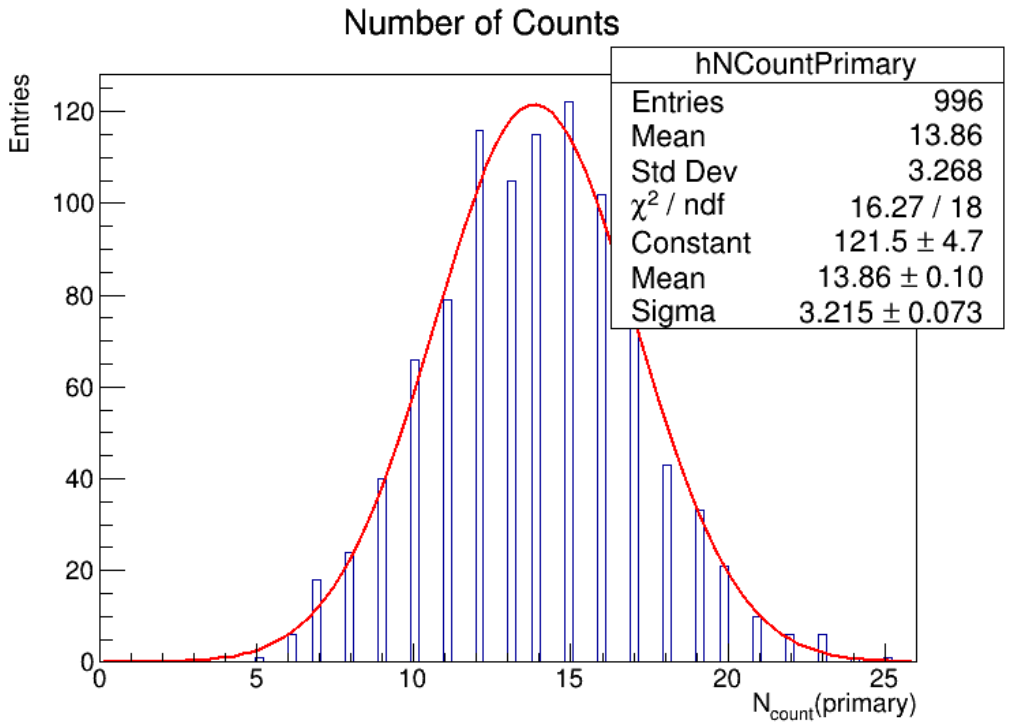
- M = 2
- Threshold = 0.6

$N_{cluster}$ detected by the algorithm

of detected pulses



of detected pulses (matched to primaries)



- The dN/dx distribution is very like a Gaussian shape
- Only a small portion of counting is related to the secondaries. If this rate is stable, there will be little harm for PID. Need further checks.

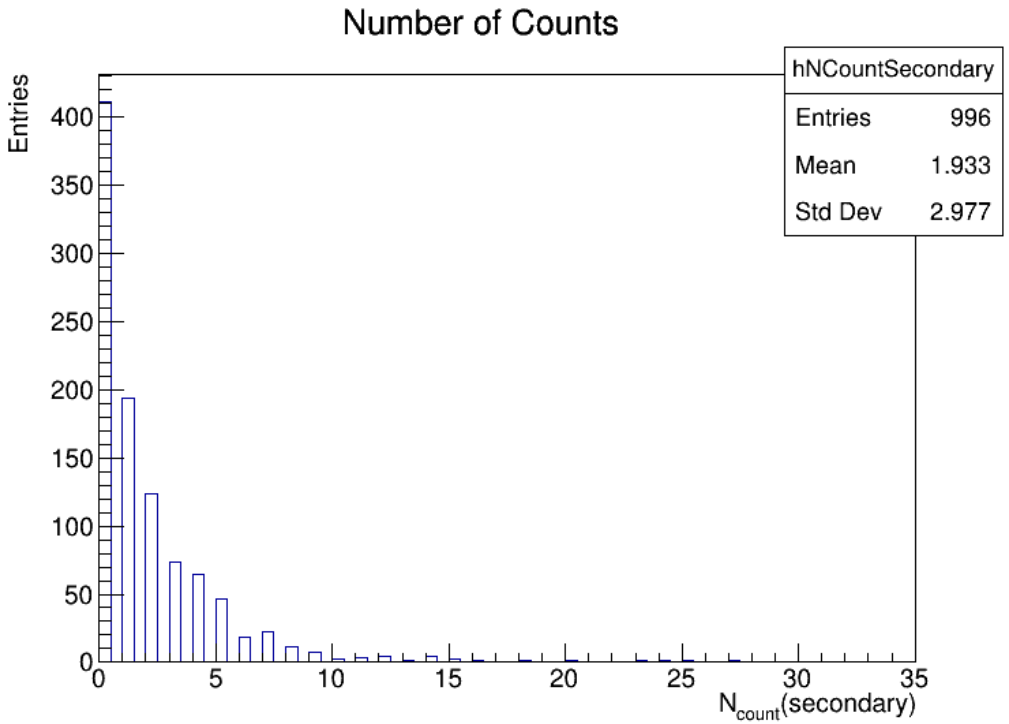
Conclusion

- **A “smoothing + derivatives” algorithm for cluster counting is developed**
- **Parameters have been optimized**
- **Next to do**
  - Develop a better low pass filter with better stopband attenuation
  - Try and optimize the algorithm based on the beam test data

Backup

More plots with the optimized algorithm

of detected pulses (matched to secondaries)



of fake pulses (match to nothing)