Cluster Counting Simulations

Jean-François Caron

LNF-INFN
Frascati

May 30, 2011

Overview

Here I present the status of cluster counting efforts from Frascati.
Emphasis is placed on the role of simulation using the Garfield program.

Essential Information

We are now able to keep all signal and cluster information from Garfield, including cluster timing. This allows us to compare cluster-counting algorithms objectively.

All simulations were done using a settings to replicate the signals from one of our single-wire tubes. The gas mixture of Helium and Isobutane at 90:10, operating voltage 1525V in a 1.5T magnetic field.

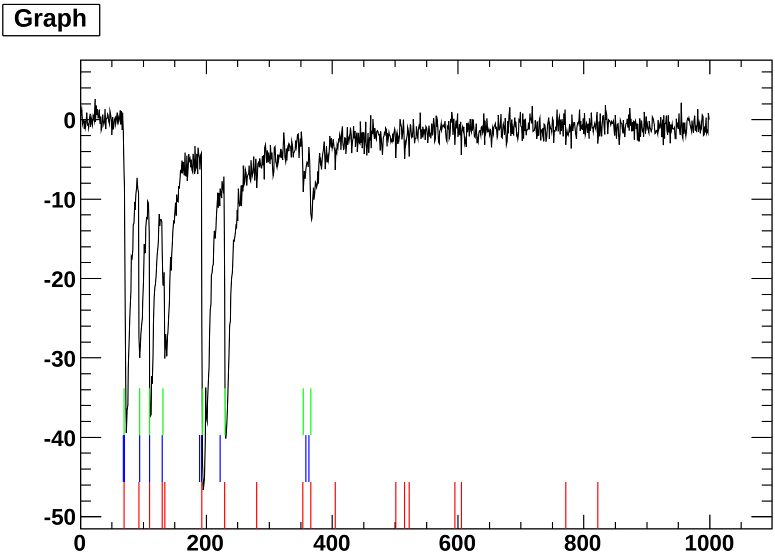
Technical Notes

Garfield gives current-at-the-wire information, so we convert the signal to voltages-at-the-oscilloscope using a convolution with a shaping function.

$$S(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}} \tag{1}$$

The result is amplified by a scaling factor to represent the electronic amplifiers. Uncorrelated (white) Gaussian noise is also added to the otherwise perfect signal

Algorithm Comparison



"Classic" Algorithm

One-way threshold crossing of the signal relative to a posterior N-frame "rolling average".

Tunable parameters: N, threshold.

This algorithm tends to miss fewer true clusters, but also find many more spurious ones. Because it tends to trigger on noise, a longer acquisition gives more clusters, even for similar events!

Typically ≈ 3.6 fake clusters per track per cell.

“Slew-Rate” Algorithm

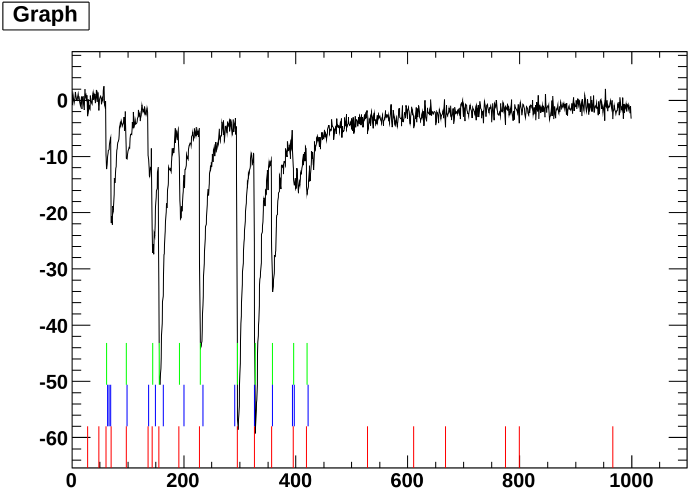
The derivative of the signal (slew rate) is required to be steeper than a threshold for 2ns. Once a cluster is identified, an artificial dead-time is applied to prevent overcounting.

Tunable parameters: Threshold, t_{dead}.

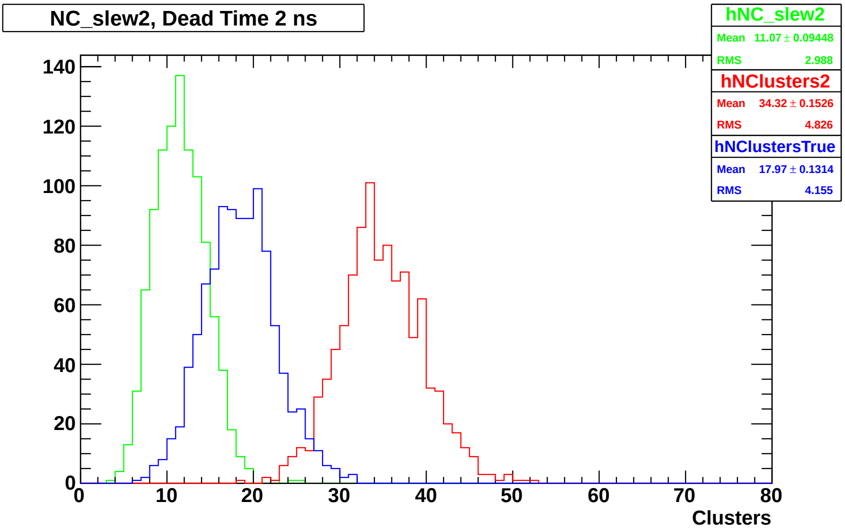
This algorithm is much less likely to find spurious clusters, but also misses a few true ones.

Typically $\approx$ 0.2 fake clusters per track per cell

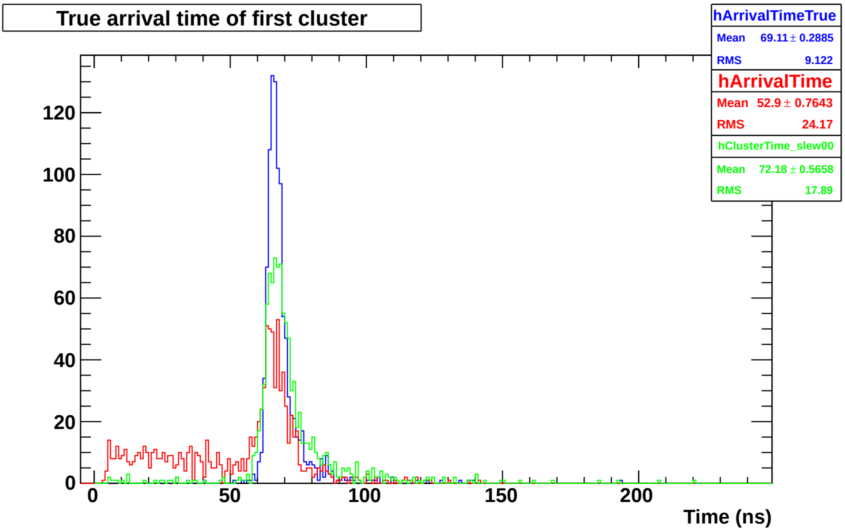
Algorithm Comparison 2



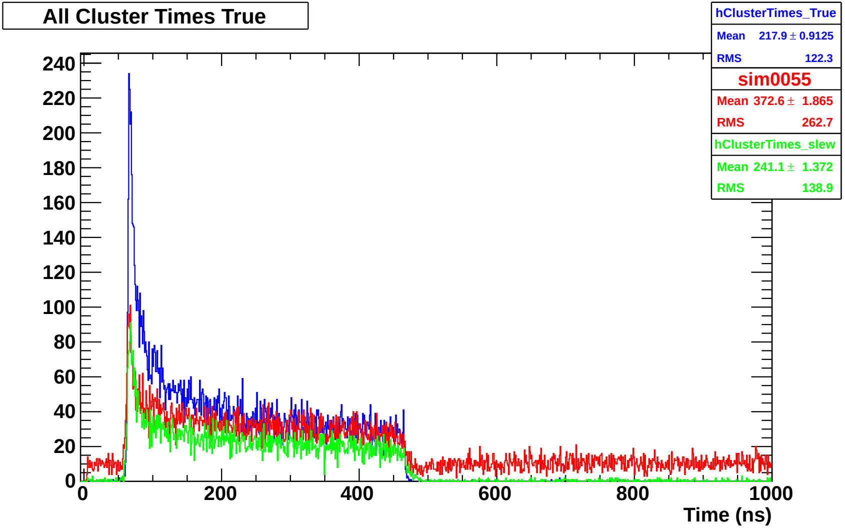
Number of Clusters Found



First Cluster Arrival Time



All Cluster Times



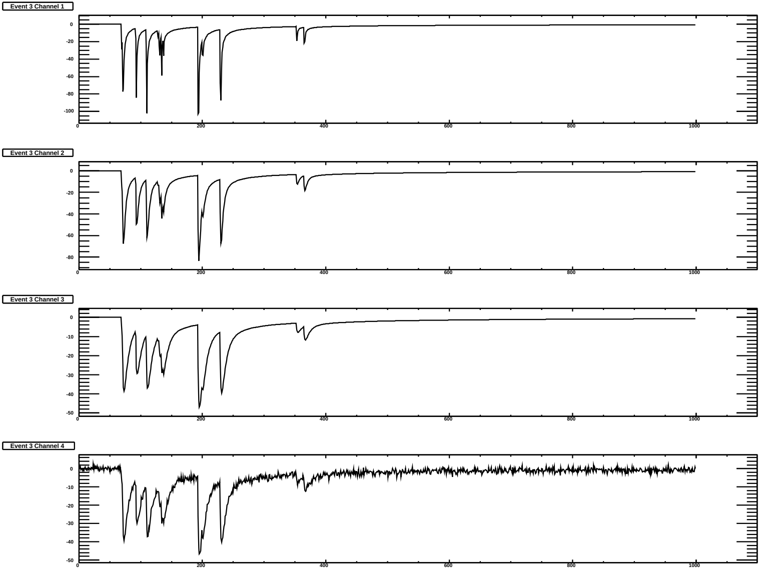
Various Parameters Chosen by Human

The time constant τ of the shaping function is chosen to be either 2 or 4.5 nanoseconds.

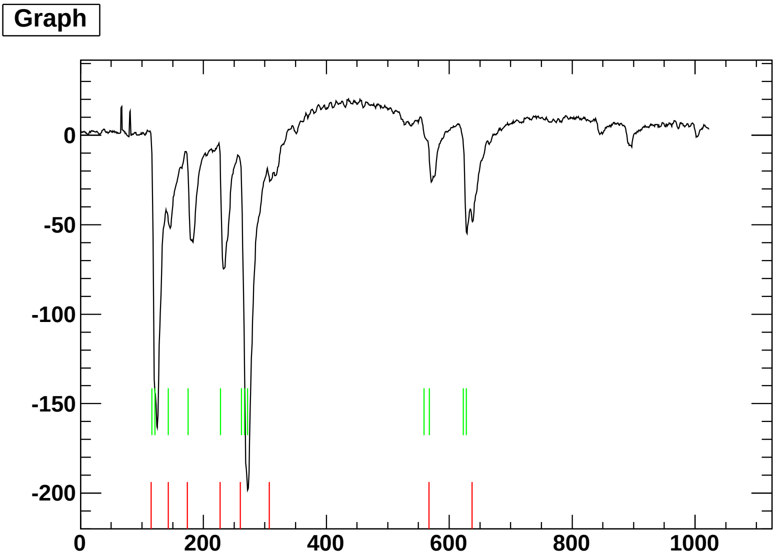
The noise added is uncorrelated and drawn from a Gaussian distribution with mean 0 and standard deviation 1 millivolt.

The amplification of all the electronics is chosen to be a factor of 300.

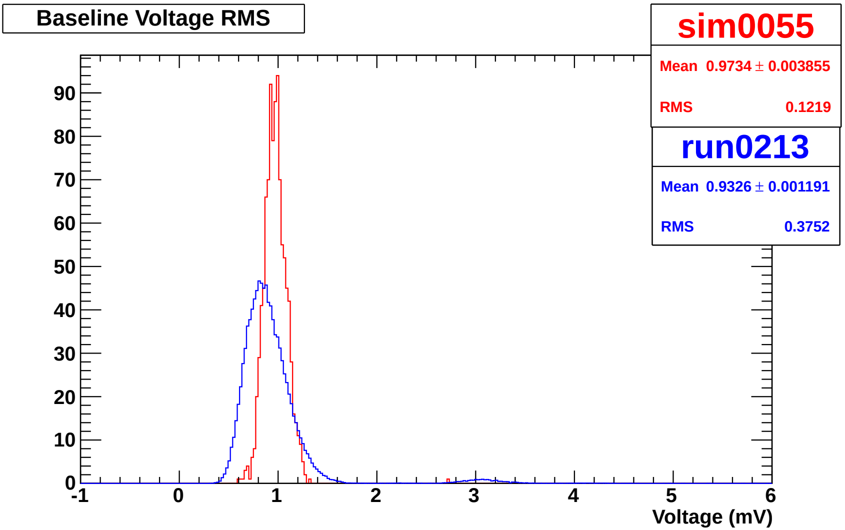
Validating τ (Simulation)



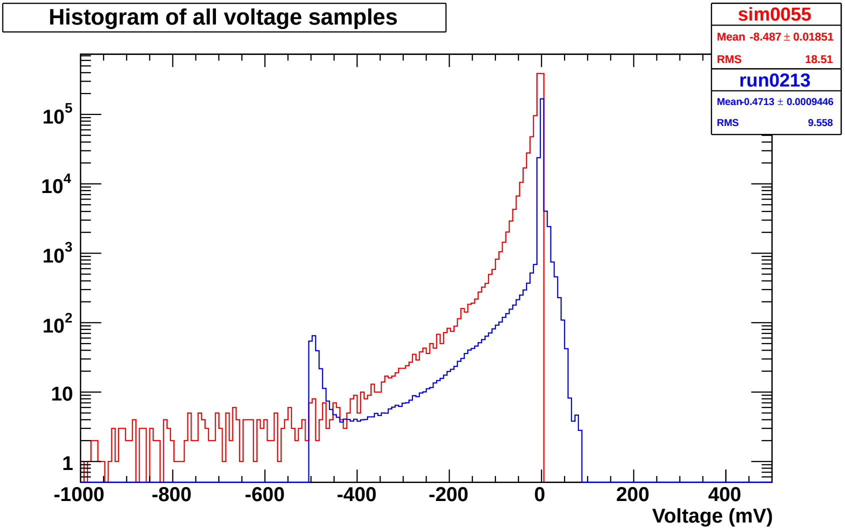
Validating τ (Data)



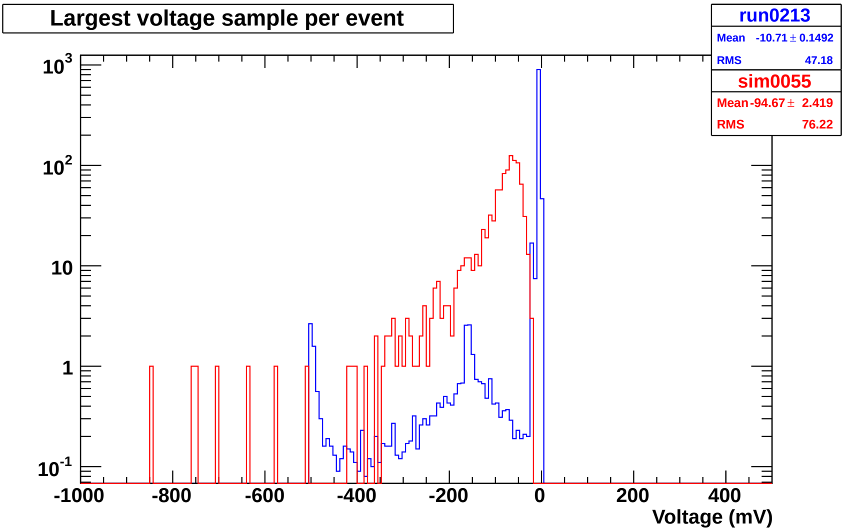
Validating Noise



Validating Amplification



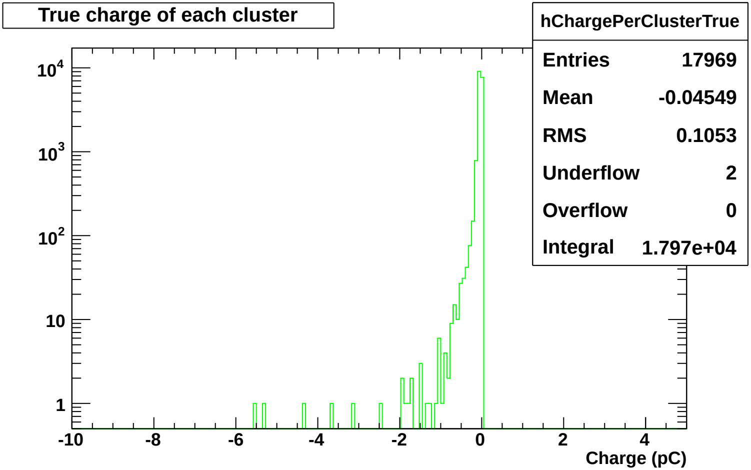
Validating Amplification (2)



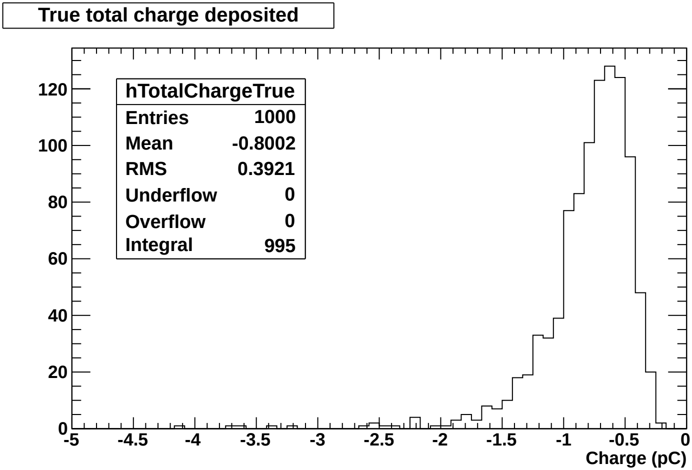
Replacing the Truncated-Mean

The distribution of charge deposited in a cell is given by a Landau distribution, which has no moments; i.e. the average of many samples diverges, even the median too! What has been done so far is to reject a fraction of all samples with the largest charges. This provides a convergent mean, but a less wasteful method would be desireable.
With cluster-wise charge information, we can instead discard the largest few clusters, but keep information from every event.

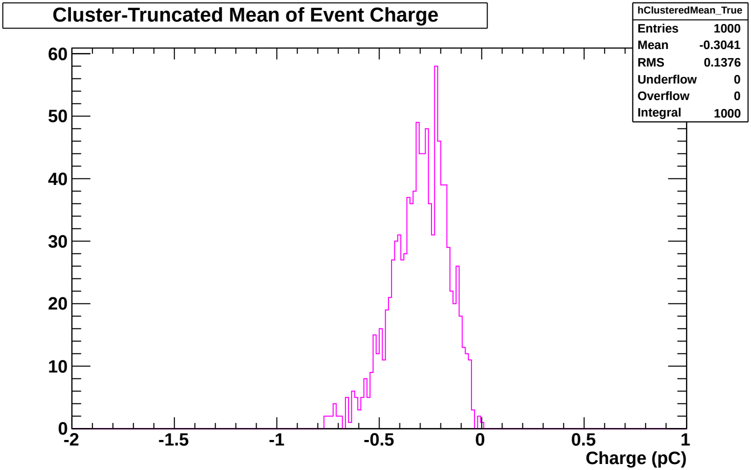
Cluster-Wise Charges (No Amplification



Total Charge w/o Discarding Clusters



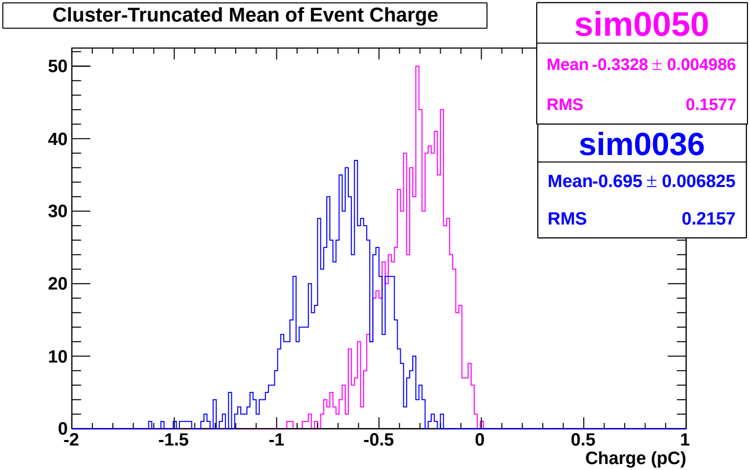
Total Charge w/5 Discards ($\approx$ 28% of clusters)



Pion-Kaon Separation

This is one area where cluster counting is hoped to dramatically improve results. The ionization behavior goes like $\frac{1}{m}$ whereas the actual tracks are nearly identical since they are relativistic. Even cluster counting which is not perfectly efficient could distinguish them.

Pion(Blue)-Kaon(Purple) Separation



Next Tasks

Our noise model is clearly incomplete. We plan to use empty data events (e.g. random trigger) to populate the simulated signals, rather than using white Gaussian noise.

Instead of using only the arrival time of the first cluster, it is hoped that by combining the timing information of the first few clusters, we can obtain a biased-but-more-accurate estimator for the impact parameter.

The codebase has become a bit bloated due to me being a physicist and not a professional programmer. An overhaul and restructuring of the analysis programs will make future work less error-prone and more extensible. As I am more familiar with Python, the plan is to re-write the analysis program using PyROOT.

In principle the best cluster information can be got from data by individually fitting each cluster as it is found, then subtracting it. This "uberfit" has been briefly explored, but more work is needed. It seems doable.