Hit Finding in the Online VETO DAQ

^{39}Ar in the Inner VETO

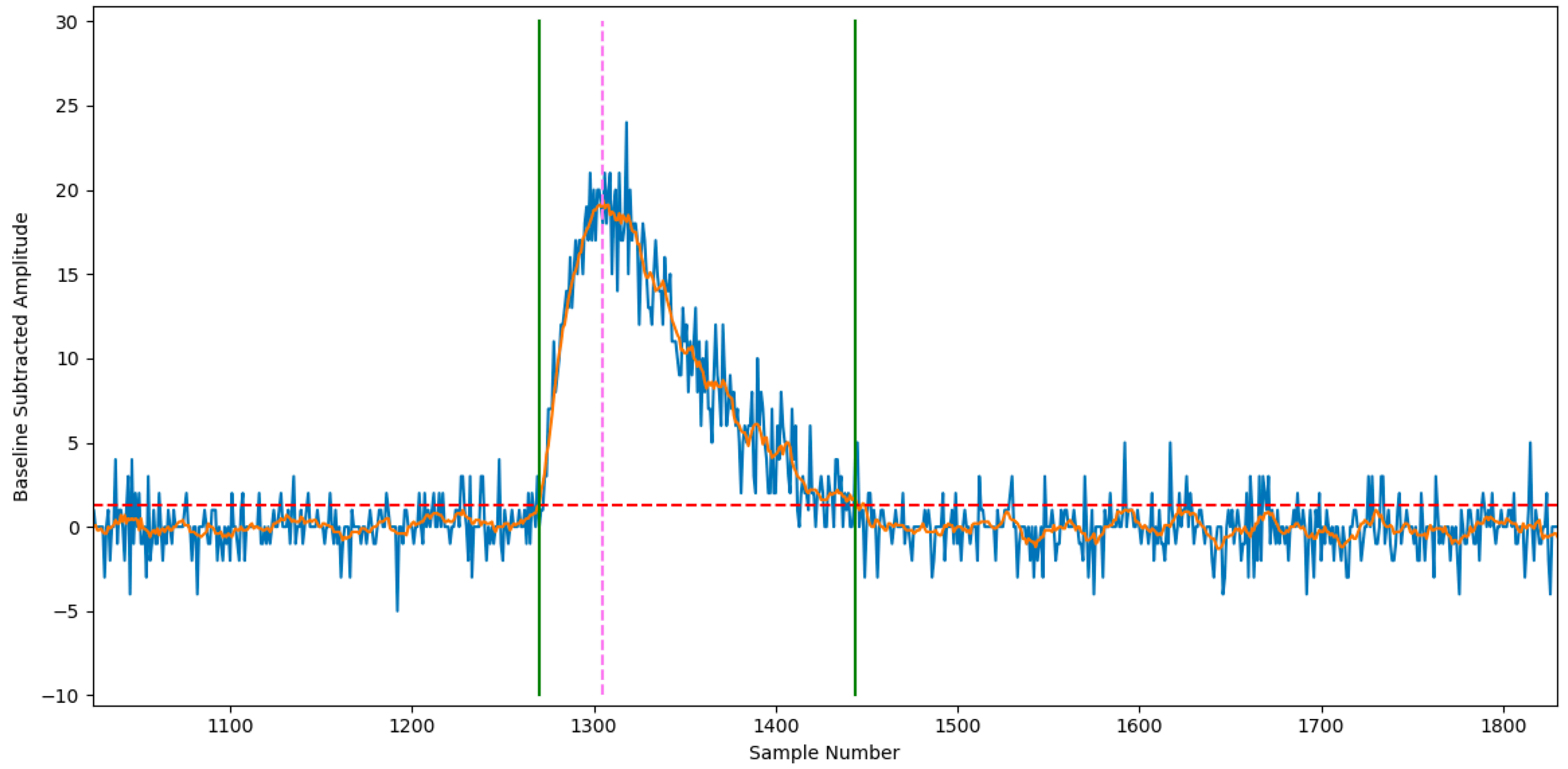
Recap: Basic hit finder used on 1 PE, ASIC waveforms in pyreco to determine fake hit rate and single PE detection efficiency in FPGA's.

Basic hit finder:

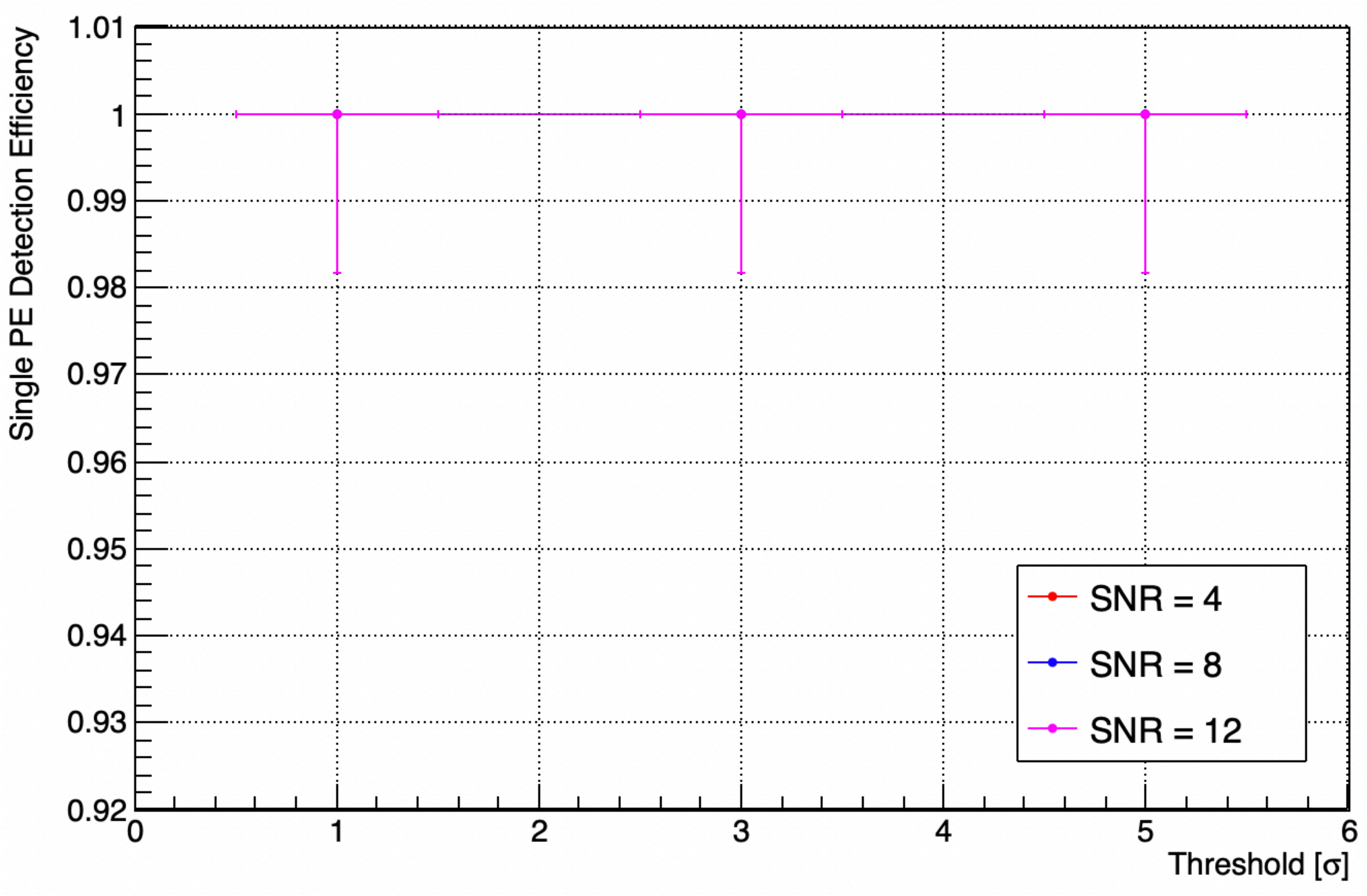
1) On RA-filtered waveform, search to see if any sample crosses a threshold, typically $\mu_{\mathrm{BL}} + x\sigma_{\mathrm{BL}}$.

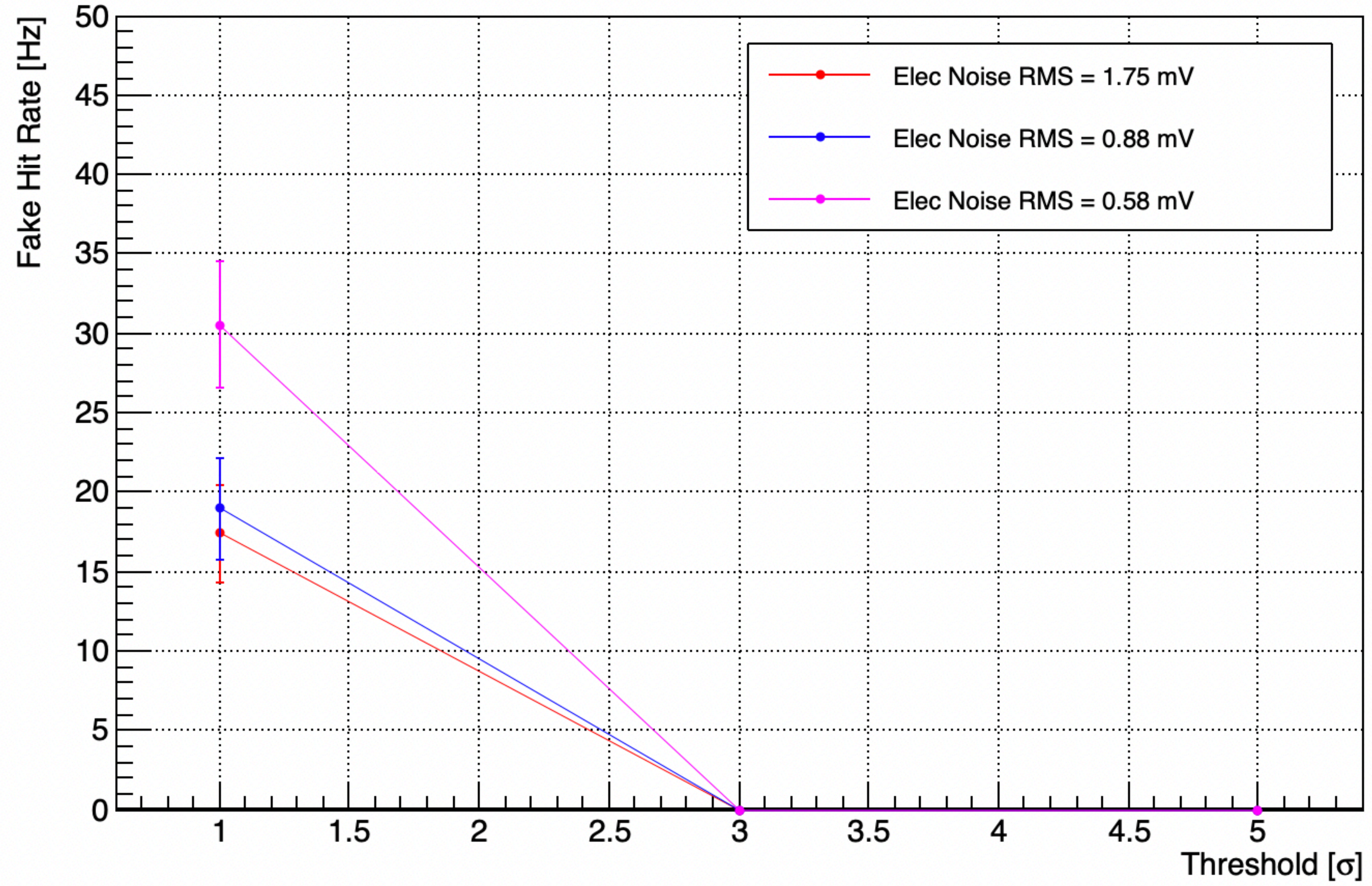
2) If sample crosses this threshold, start summing samples of waveform - continue until sample falls below threshold.

3) Check if the hit is long enough: if ToT > value, yes, otherwise do not record hit.



ToT = 50 bins [400 ns]





Hit Finding in the Online VETO DAQ

^{39}Ar in the Inner VETO

Goal: attempt to implement basic filtering and hit finding at FEP level in ds_vslice to test if there is enough CPU to perform this online, to study the average event size/ throughput, to study whether the output can be used in the TSP as a trigger for a veto-triggered data stream, and to check whether all physics information is retained with this acquisition method.

This hit finder can be used in the FPGA for identifying ZLE segments, but not for finding individual pulses in FEP.

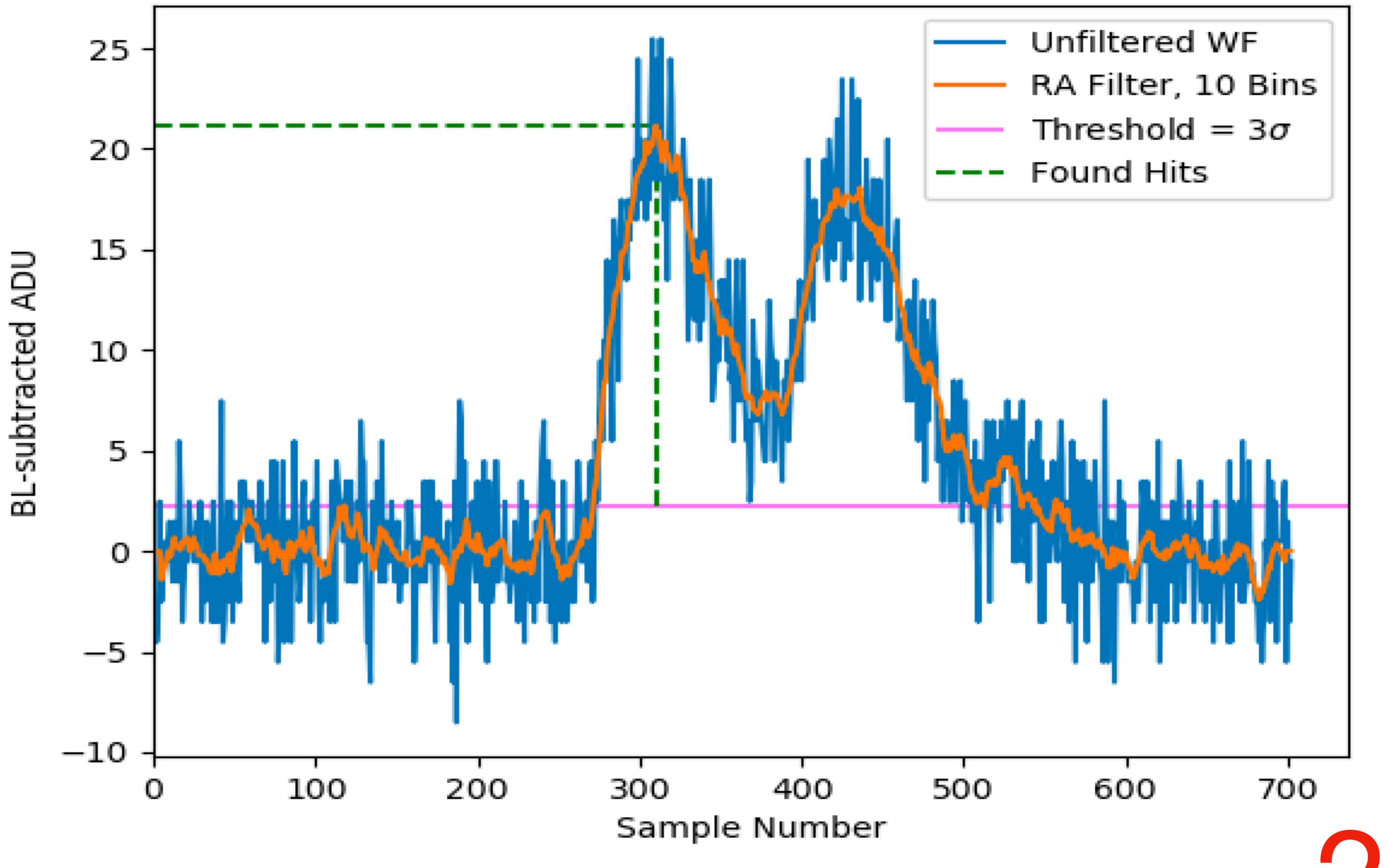
Following WF segments are generated from ^{39}Ar generated in the inner veto from pyreco with ASIC response (older Plan C geometry with 2024 channels) with ASIC response, SNR = 5.

WF segments are calculated using true PE times from G4DS. Nominal window is 7000 ns [2000 ns pre-trigger, 5000 ns post-trigger] — if another PE detected within window, window extended by further 7000 ns from subsequent PE time — WF segments roughly range between 7 - 15k ns.

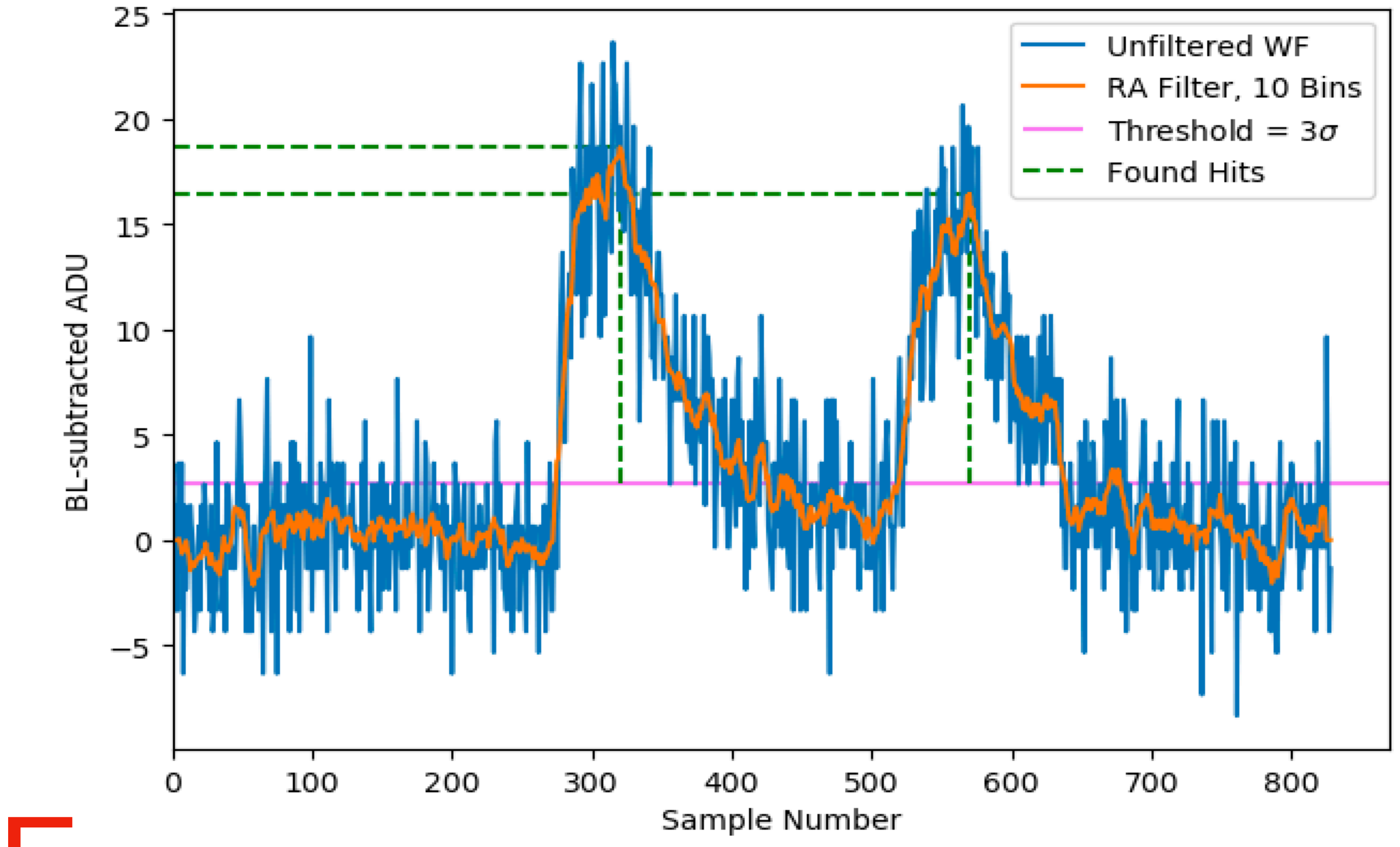
Hit Finding in the Online VETO DAQ

^{39}Ar in the Inner VETO

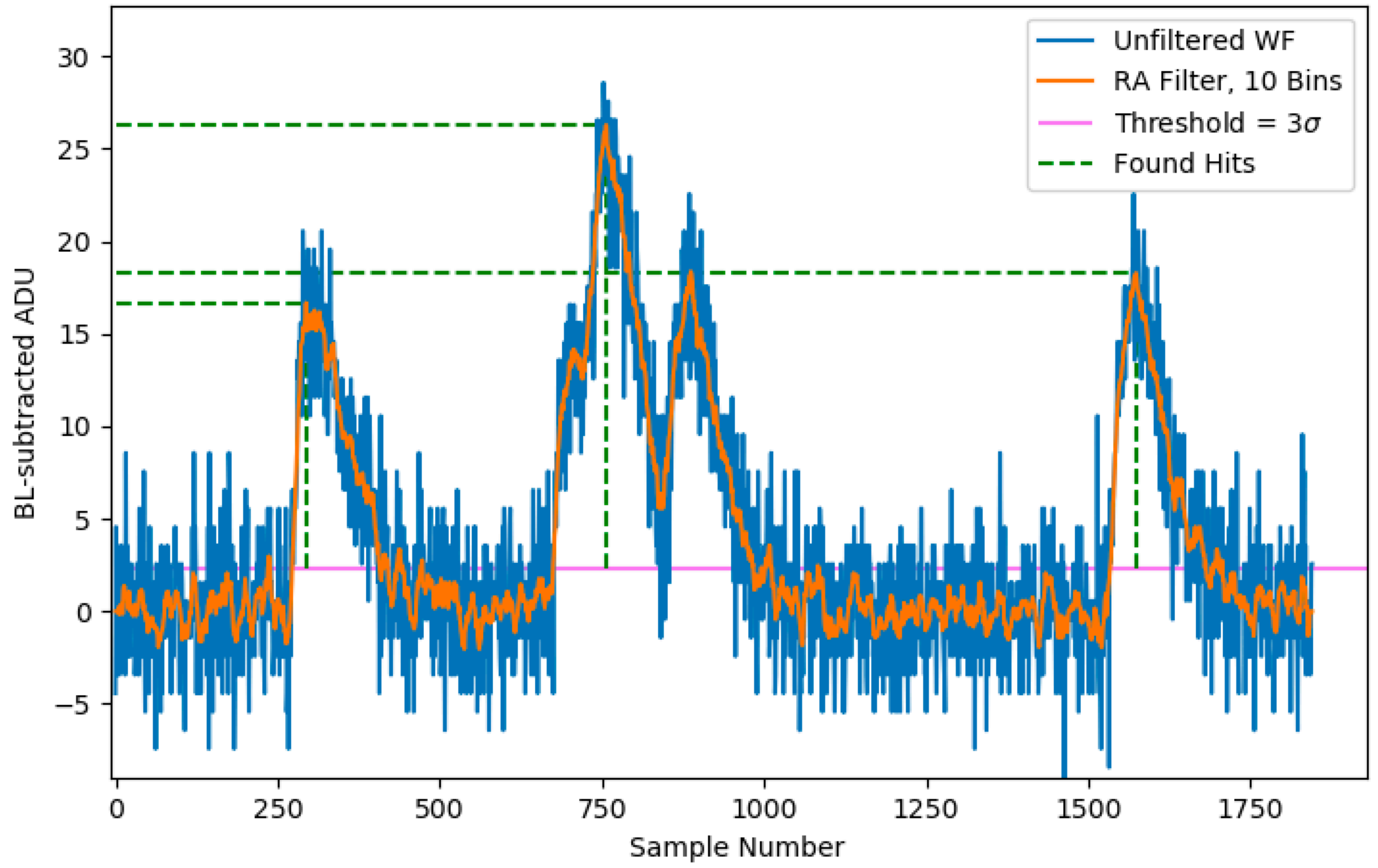
2 PE



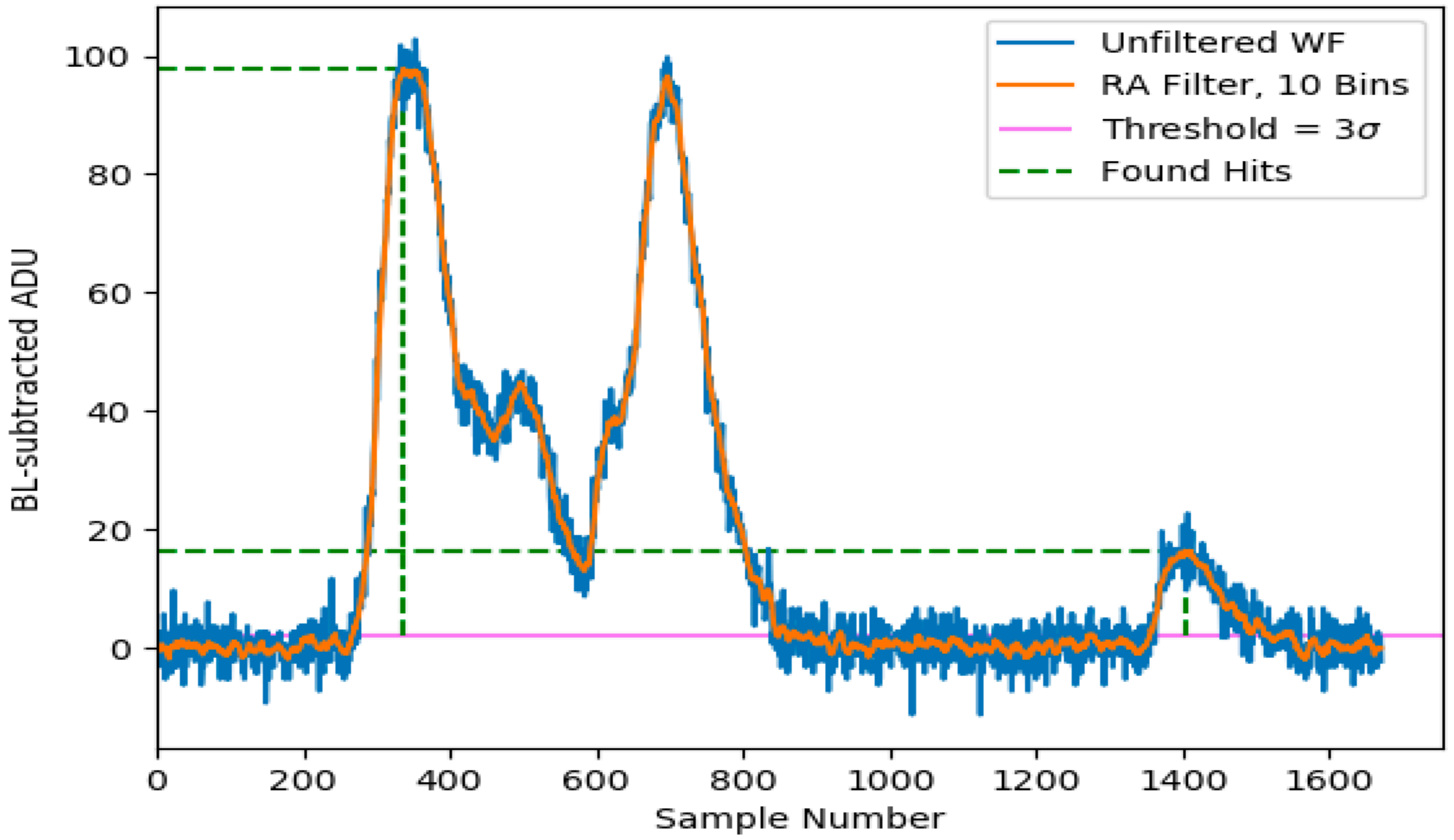
2 PE



5 PE



20 PE



Hit Finder Improvements

^{39}Ar in the Inner VETO

New hit finder method (for FEP purposes):

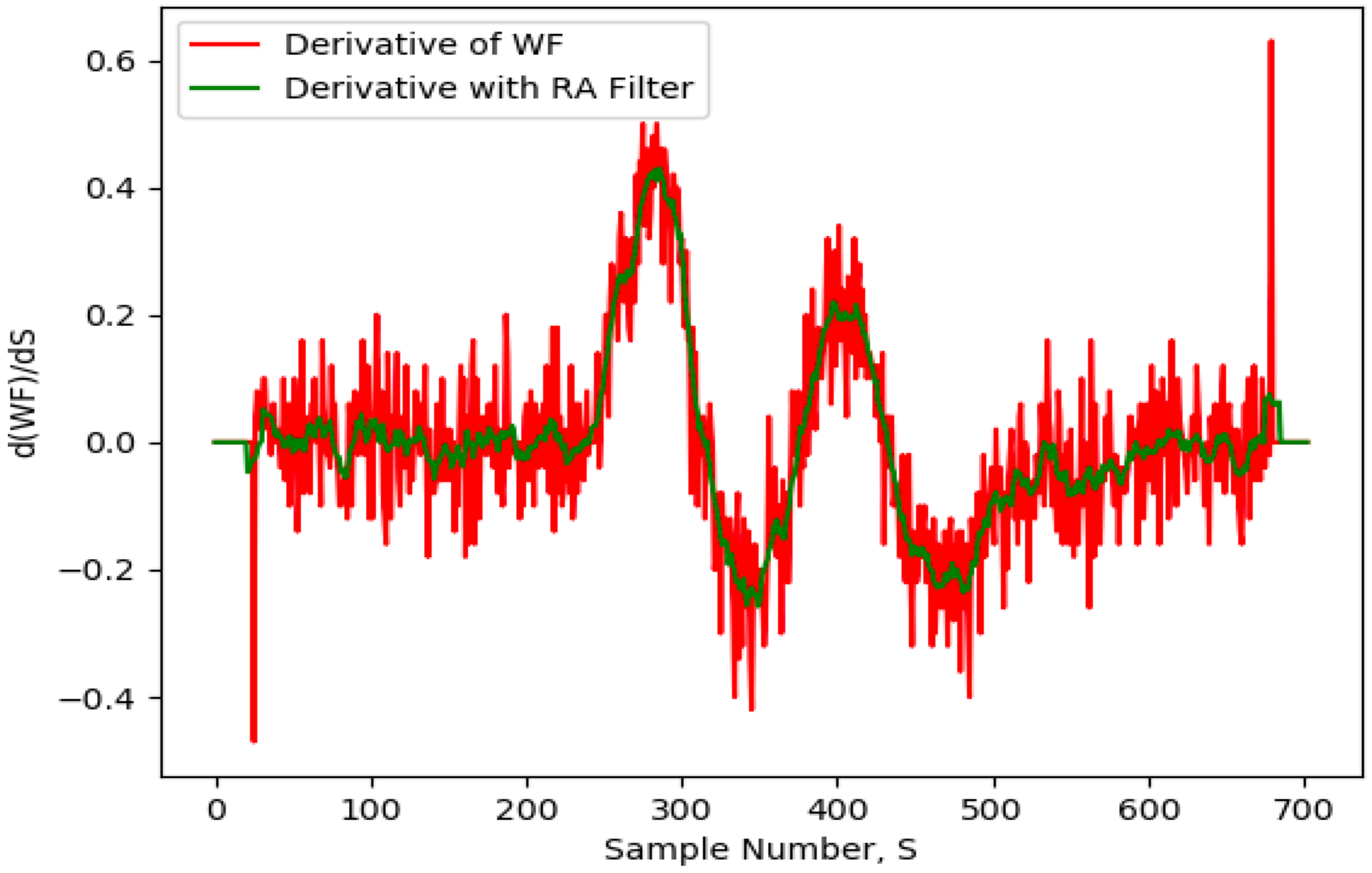
1) On RA-filtered WF, calculate the derivative of samples [i , i-1].

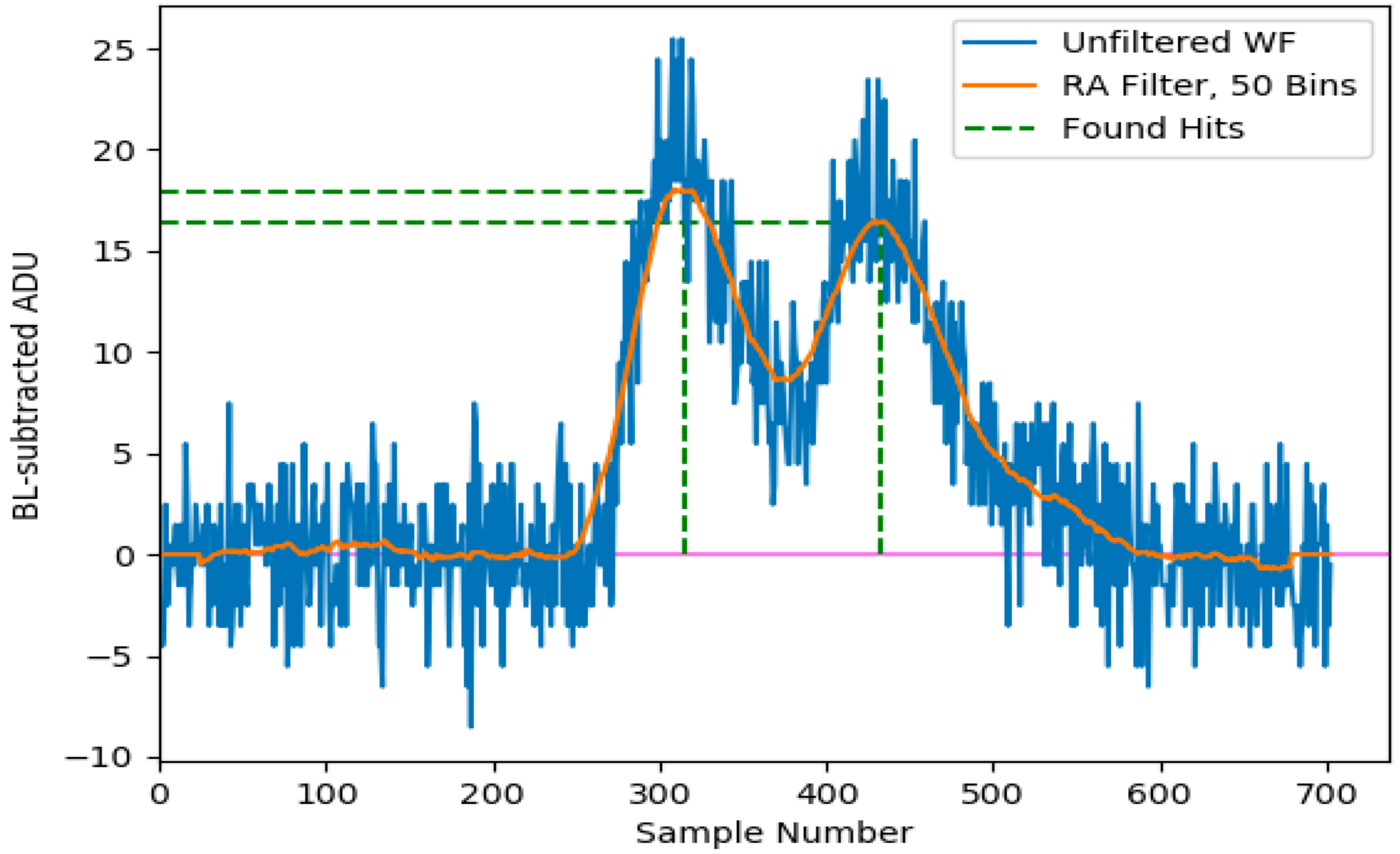
2) Apply another RA filter to smooth out waveform.

3) Find when threshold is crossed in derivative (set to 0.05) - count how many consecutive bins are above threshold:

  3.1) If ToT is at least 10 bins, find time of zero-crossing (== peak of hit, $t = i_{hit}$). Restart hit counter after $t = i_{hit}$.

  3.2) If ToT is less than 10 bins, consider this noise fluctuations and restart hit counter.





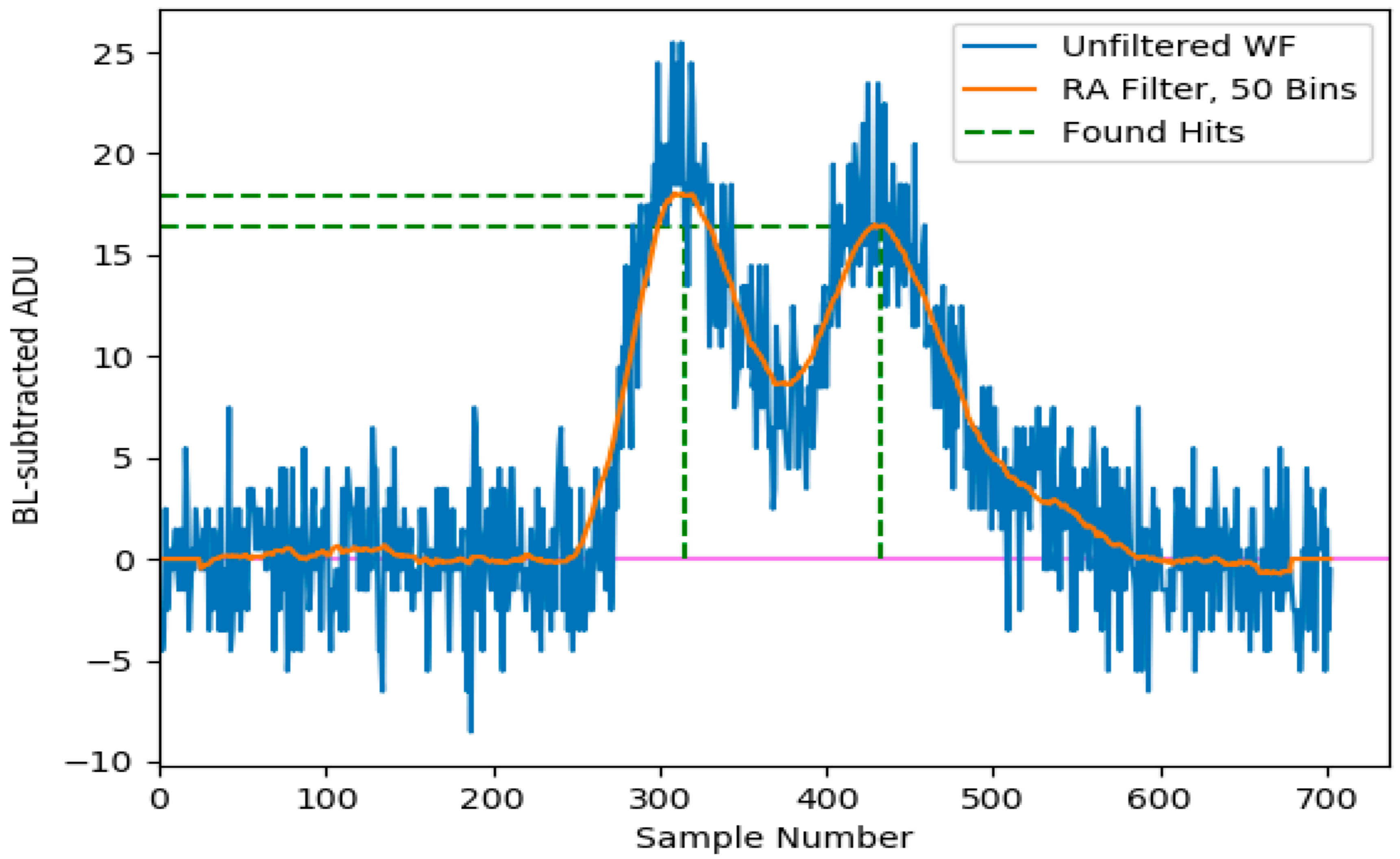
Need to properly quantify efficiency of this hit finder, but some preliminary studies, looks to be far more effective at identifying hits close to each other.

Also this is not contingent upon doing a BL-subtraction on the WF beforehand.

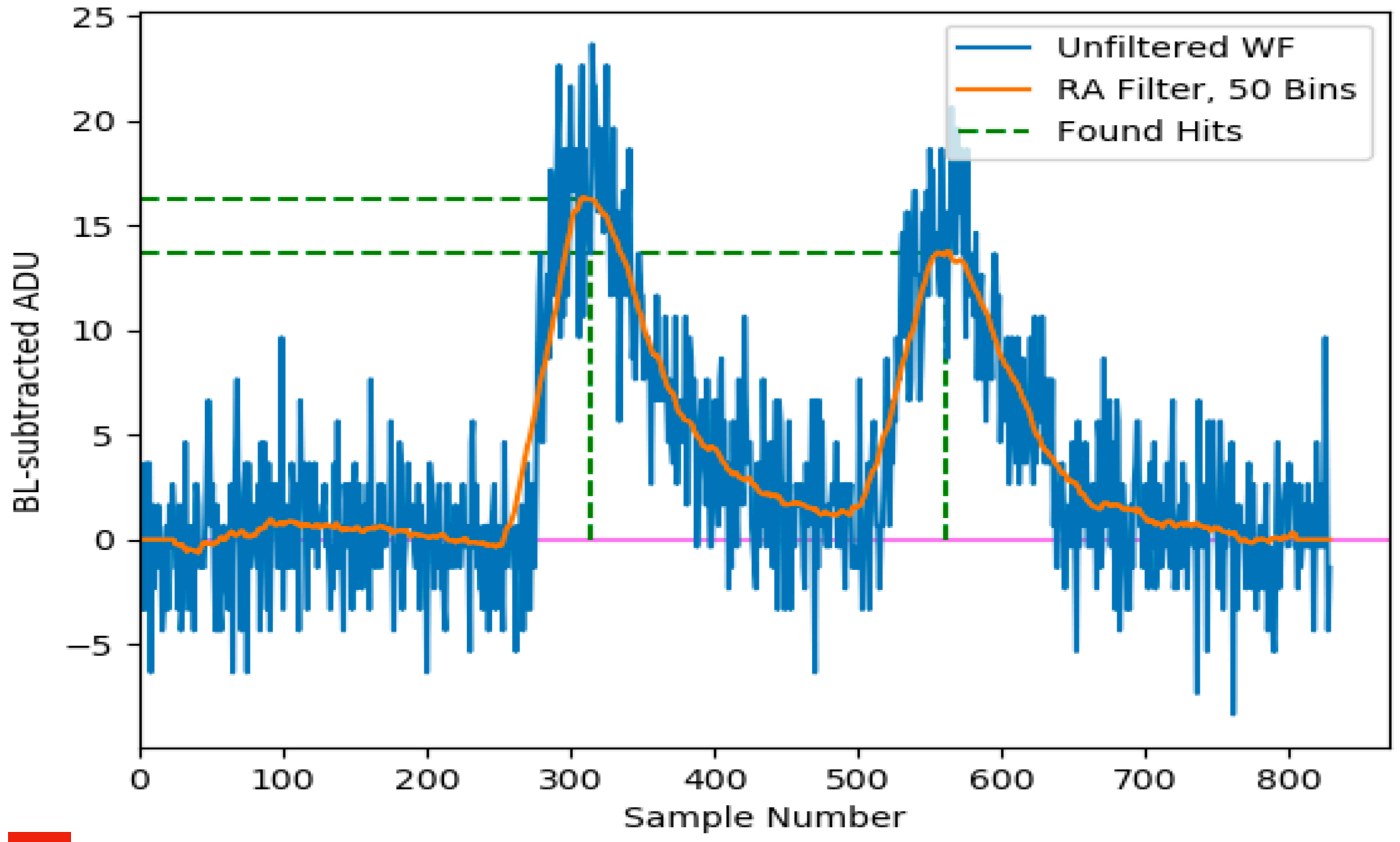
Hit Finder Improvements

^{39}Ar in the Inner VETO

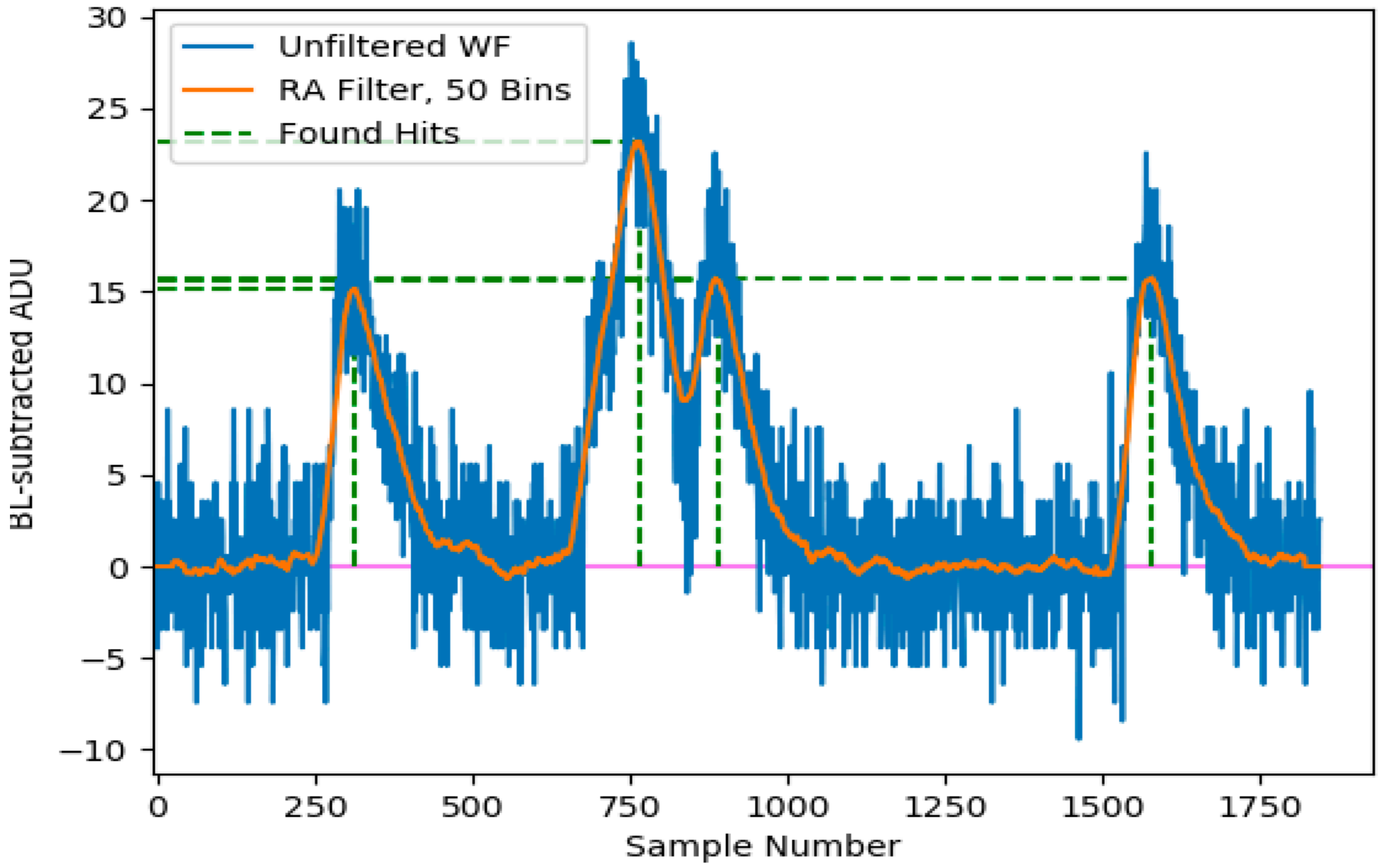
2 PE



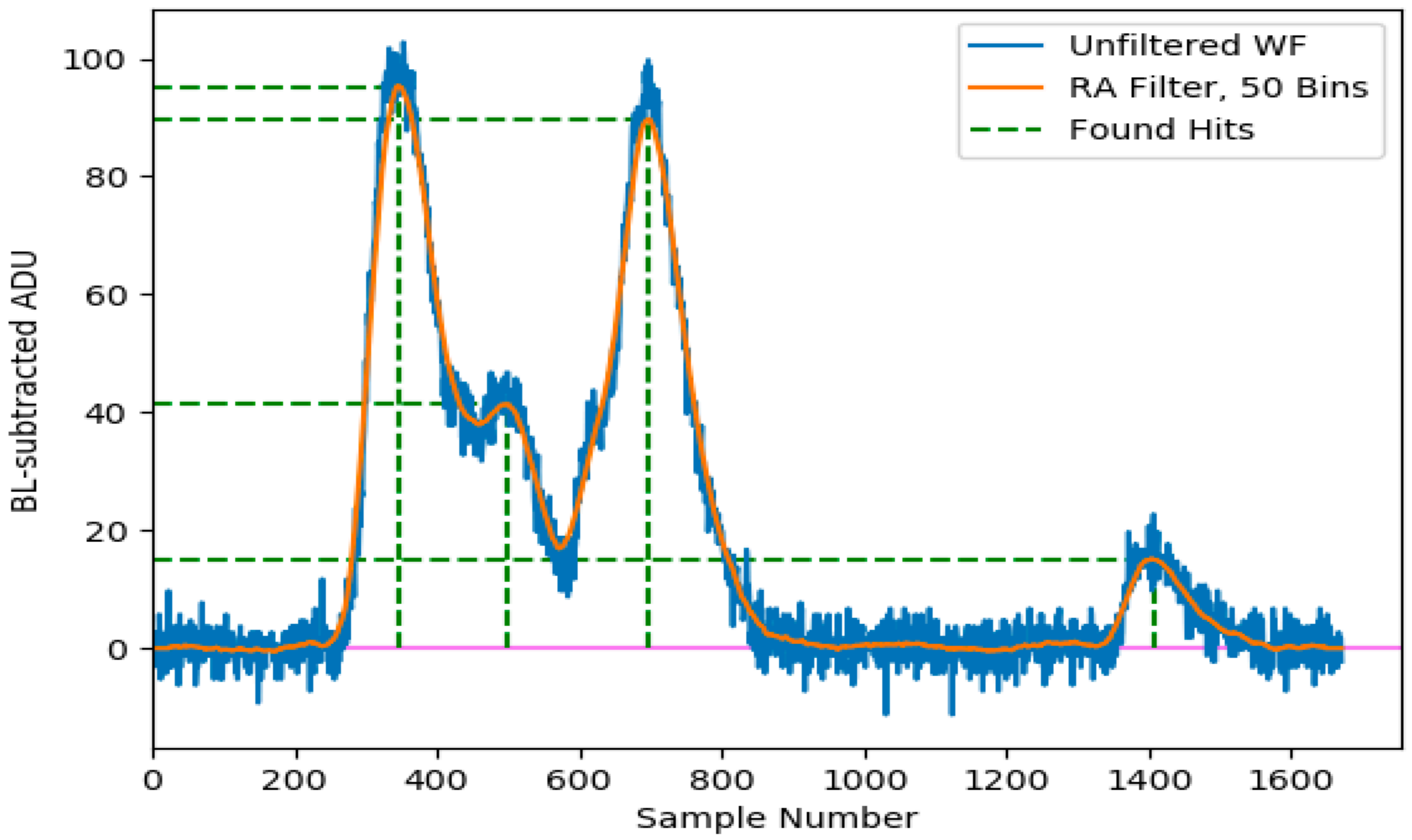
2 PE



5 PE



20 PE



Hit Finder Improvements

^{39}Ar in the Inner VETO

Construct $N_{PE,rec}$ vs $N_{PE,true.}$ [Right]

For 100, 1 PE pulses, average prominence from hit finder of RA-filtered WF is calculated. This is used to calculate $N_{PE,rec}$ *by* dividing prominence of each found hit identified by hit finder in each WF by 1 PE prom. Std of 1 PE prominence is propagated in error on $N_{PE,rec.}$

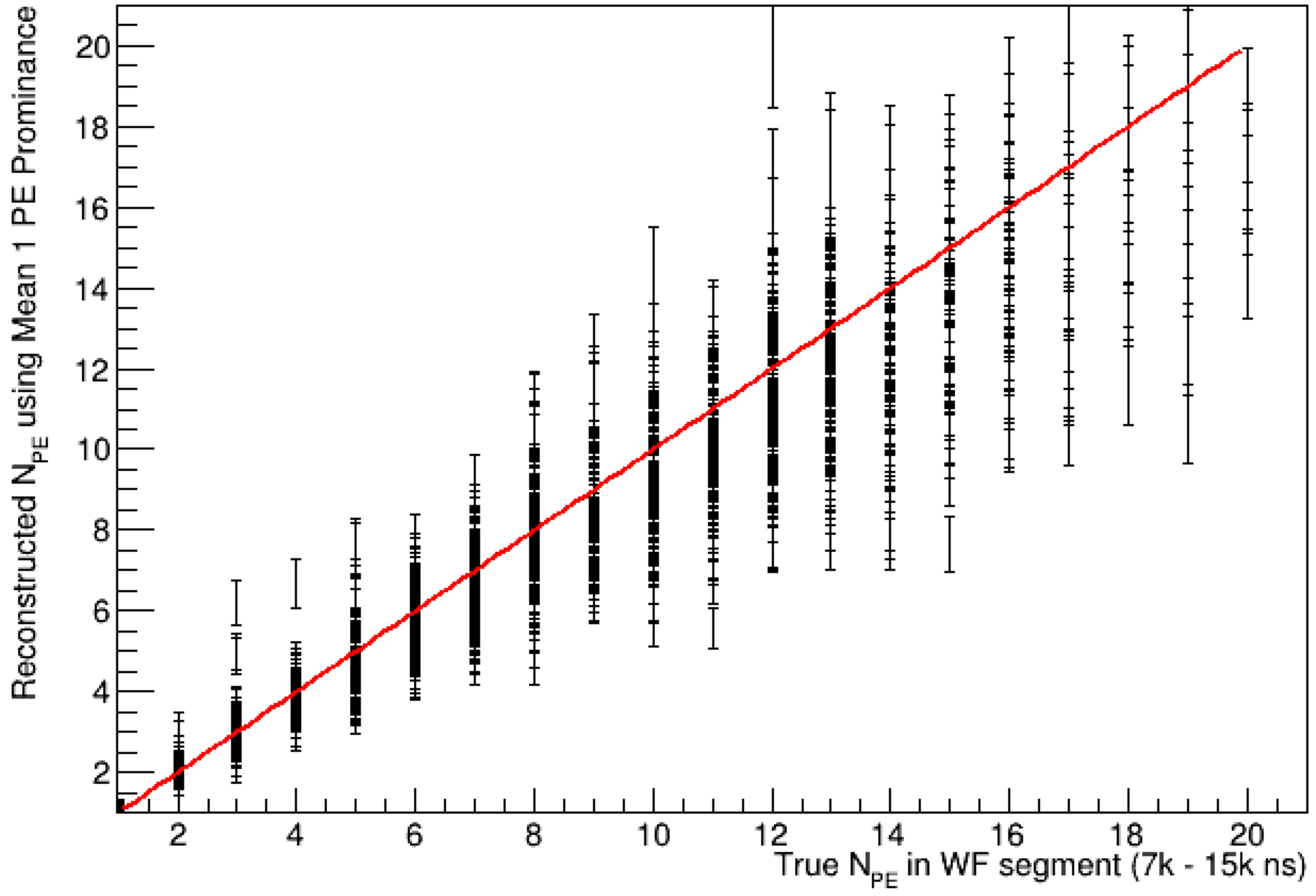
As expected, higher occupancy results in a reduction in $N_{PE,rec.}$

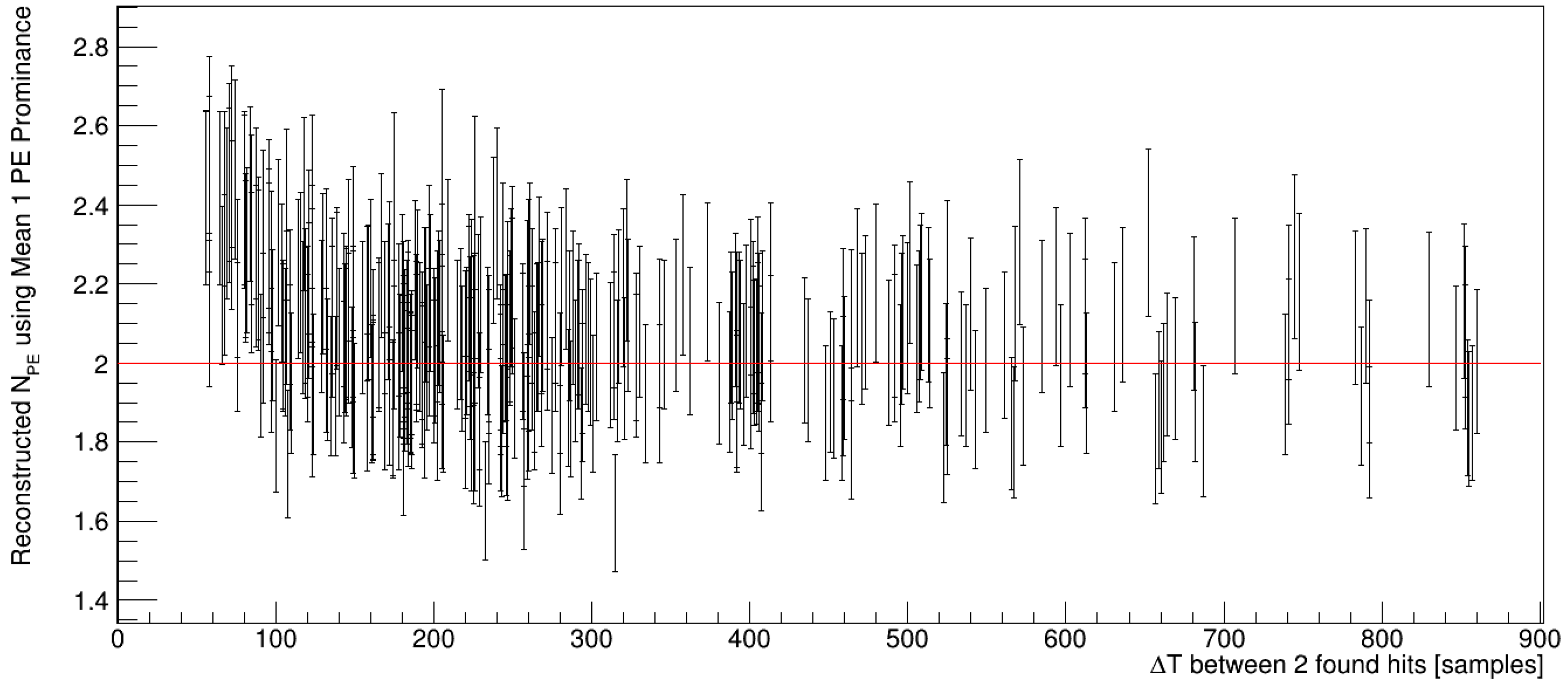
For 2 PE WFs, calculate $N_{PE,rec}$ vs ΔT of two hit found by hit finder. [Lower left]

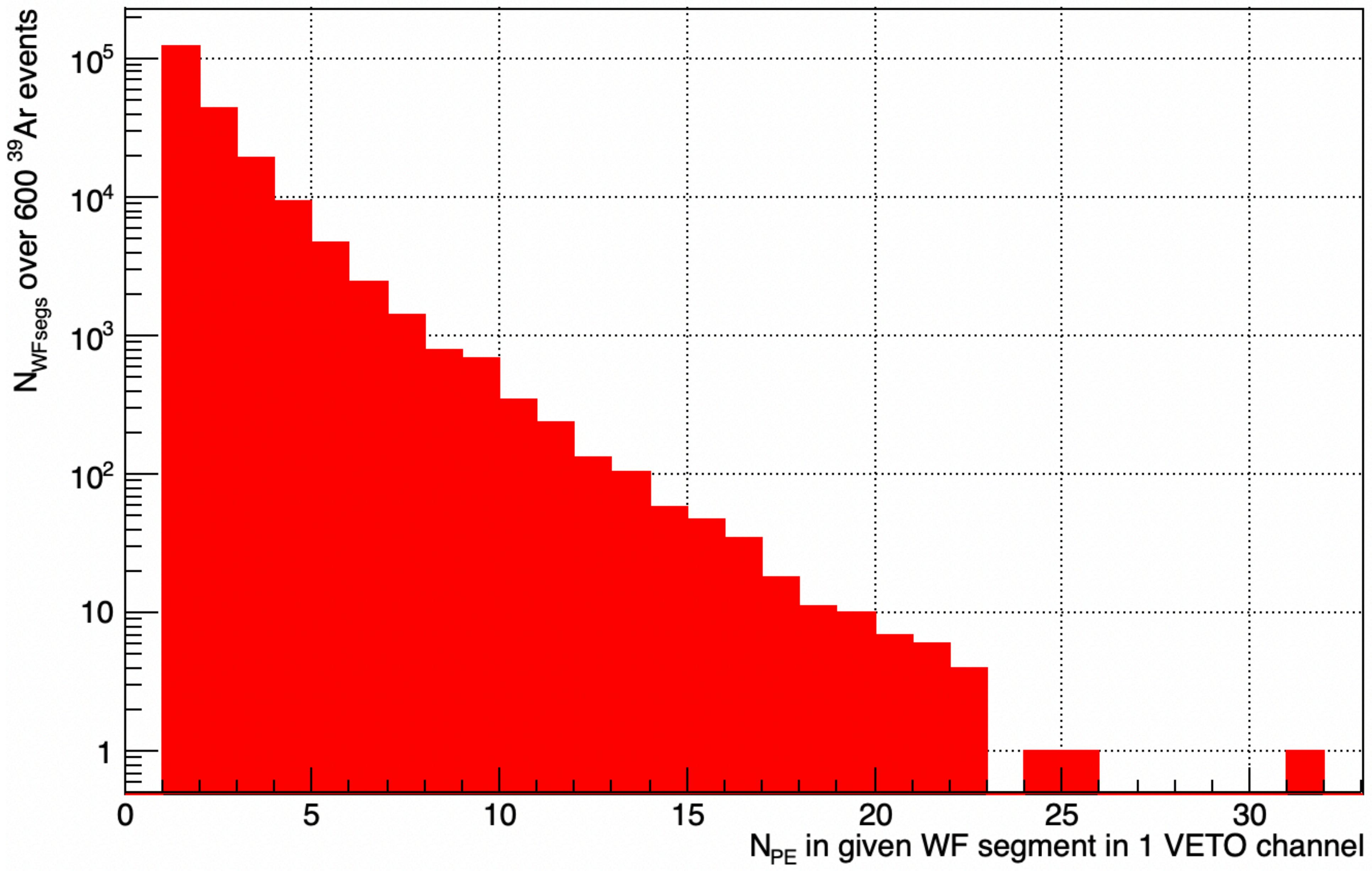
At lower ΔT, $N_{PE,rec}$ tends to increase, and we seem to lose sensitivity to distinguish 2 individual pulses at ~50 samples (400 ns).

Channel occupancy [Lower right].

With no channel summation, 60% of WF segments as defined on slide 2 are single PE.







VETO rates/throughput using ds_vslice

^{39}Ar in the Inner VETO

Using WF segments from ^{39}Ar generated in the inner veto from pyreco (older Plan C geometry with 2024 channels - as shown on previous slides), can implement a realistic data throughput using ds_vslice at TRIUMF.

^{39}Ar rate = 26 Hz, DCR/ch = 24 Hz - no other sources included so far.

Two scenarios considered:

1) In first case, no filtering/hit finding is performed on WF segments at FEP, raw WFs sent from FEPs to TSPs.

2) In second case, RA-filtered is first performed on WF segments, then hit finder described before is applied. MultiHit object is sent through FEP only, consisting of a std::map of prominances and times of each hit instead of WFs.

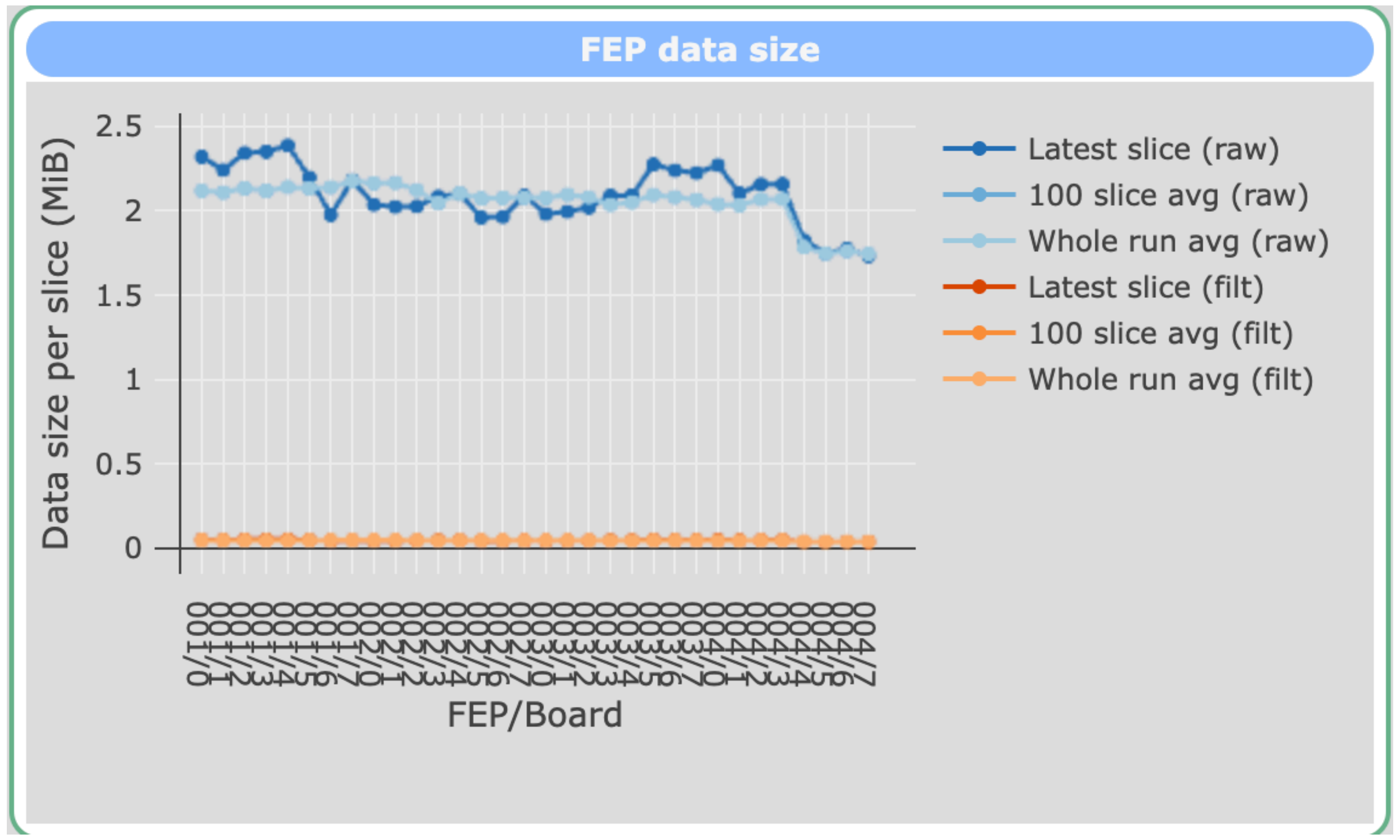
VETO rates/throughput using ds_vslice

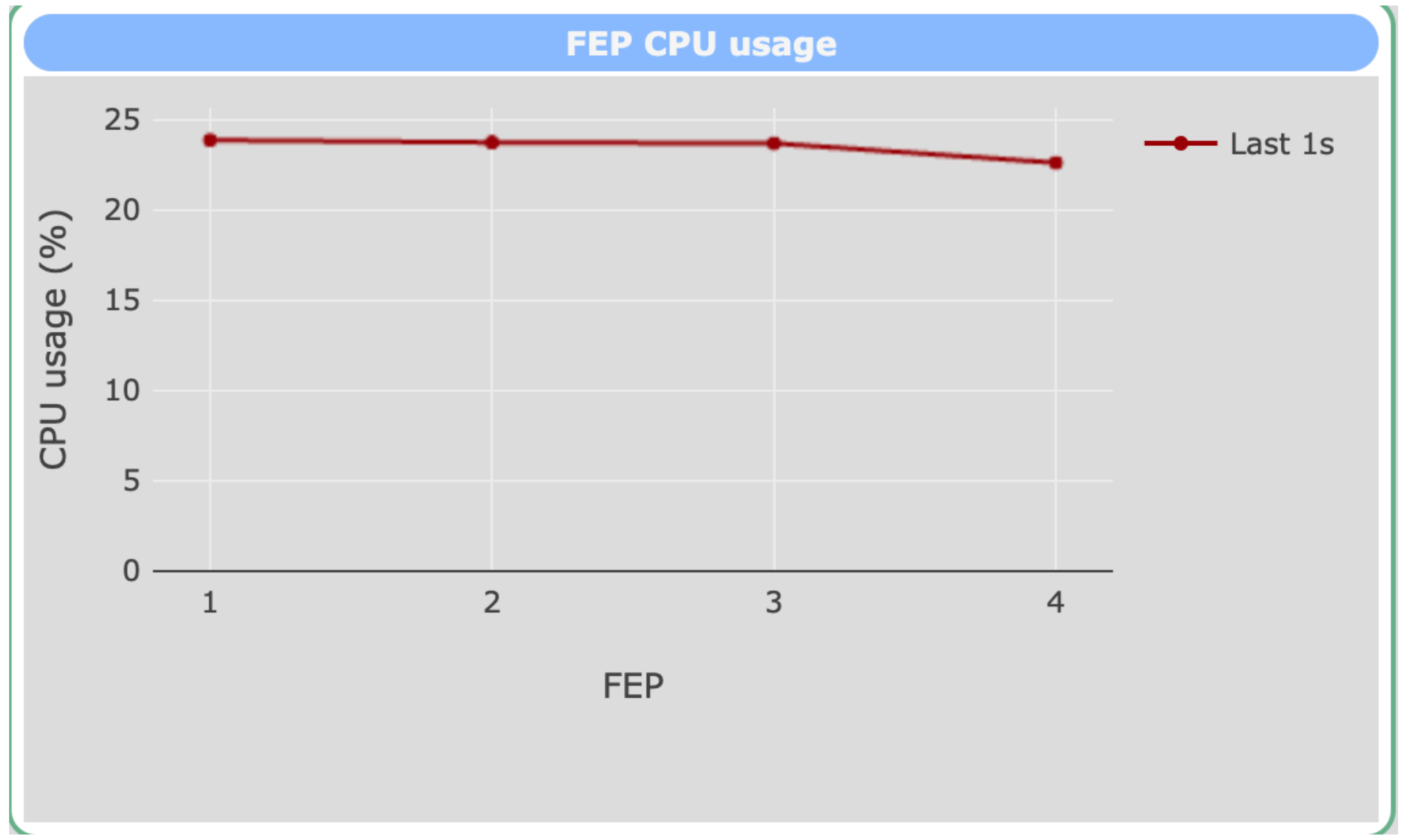
^{39}Ar in the Inner VETO - Case 1 vs Case 2

Sending MultiHit object instead of full WF reduces throughout of FEP by factor of 40.

Performing RA-filtered + hit finding on FEP does not seem to FEP overwhelm CPU usage (LR = Filtering + hit finding)

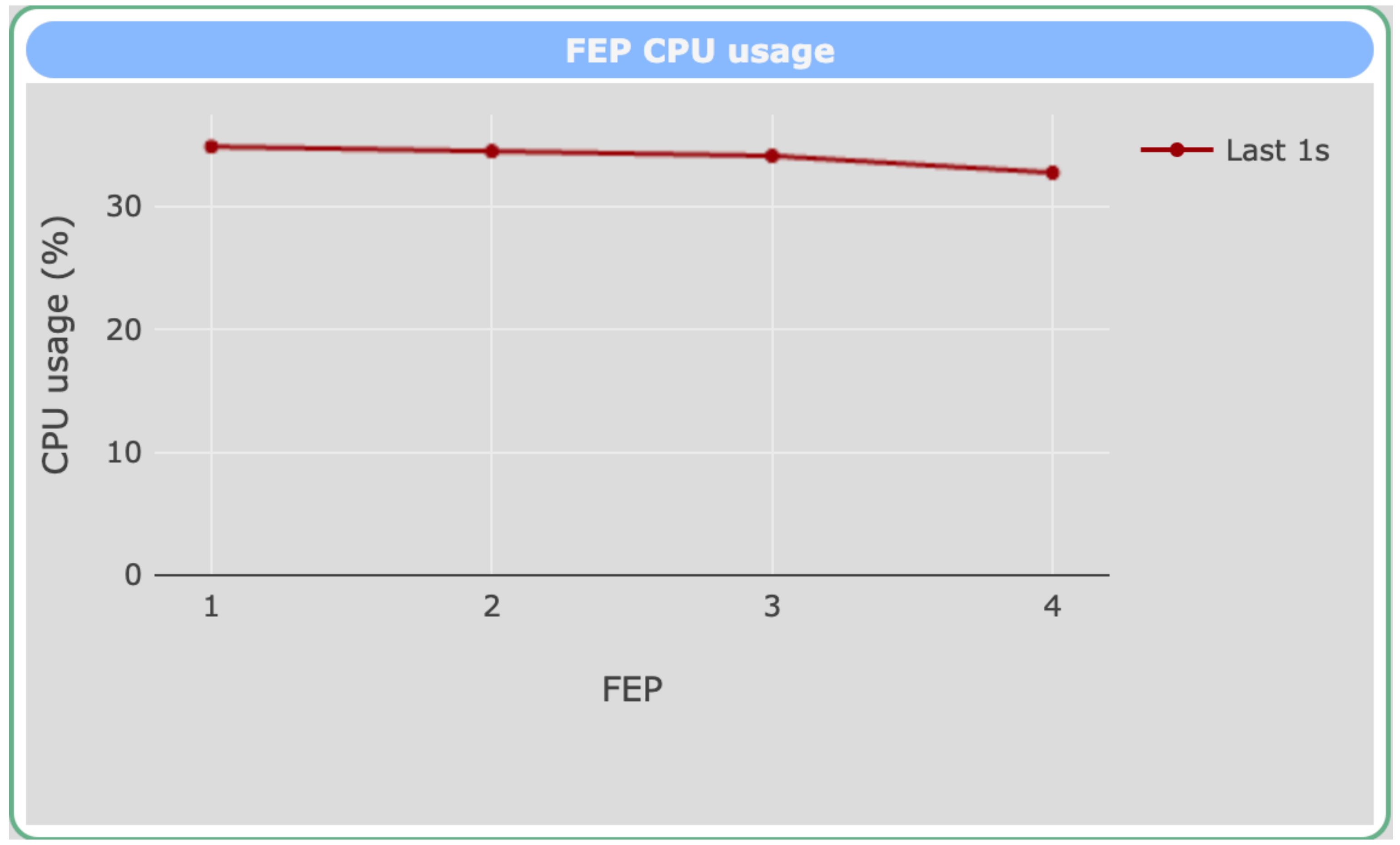
Need to add in gamma sources, but currently no worry about data throughput overwhelming DAQ system for **inner** VETO system - most likely a different story for outer!





Need to sanity check output of hit finding in ds_vslice.

Raw output has been verified, but not yet able to study output from ds_vslice due to potential encoding bug of MultiHit object - under investigation.



Next Steps.

Verify output of MultiHit in ds_vslice - ensure it is working as expected.

Include gamma contributions.

Update to newest MC/geometry when MC is available.

VETO inefficiency/deadtime for this hit finding algorithm.

Repeat above work for 2/4-summed cases.

Inclusion of outer VETO into DAQ studies.