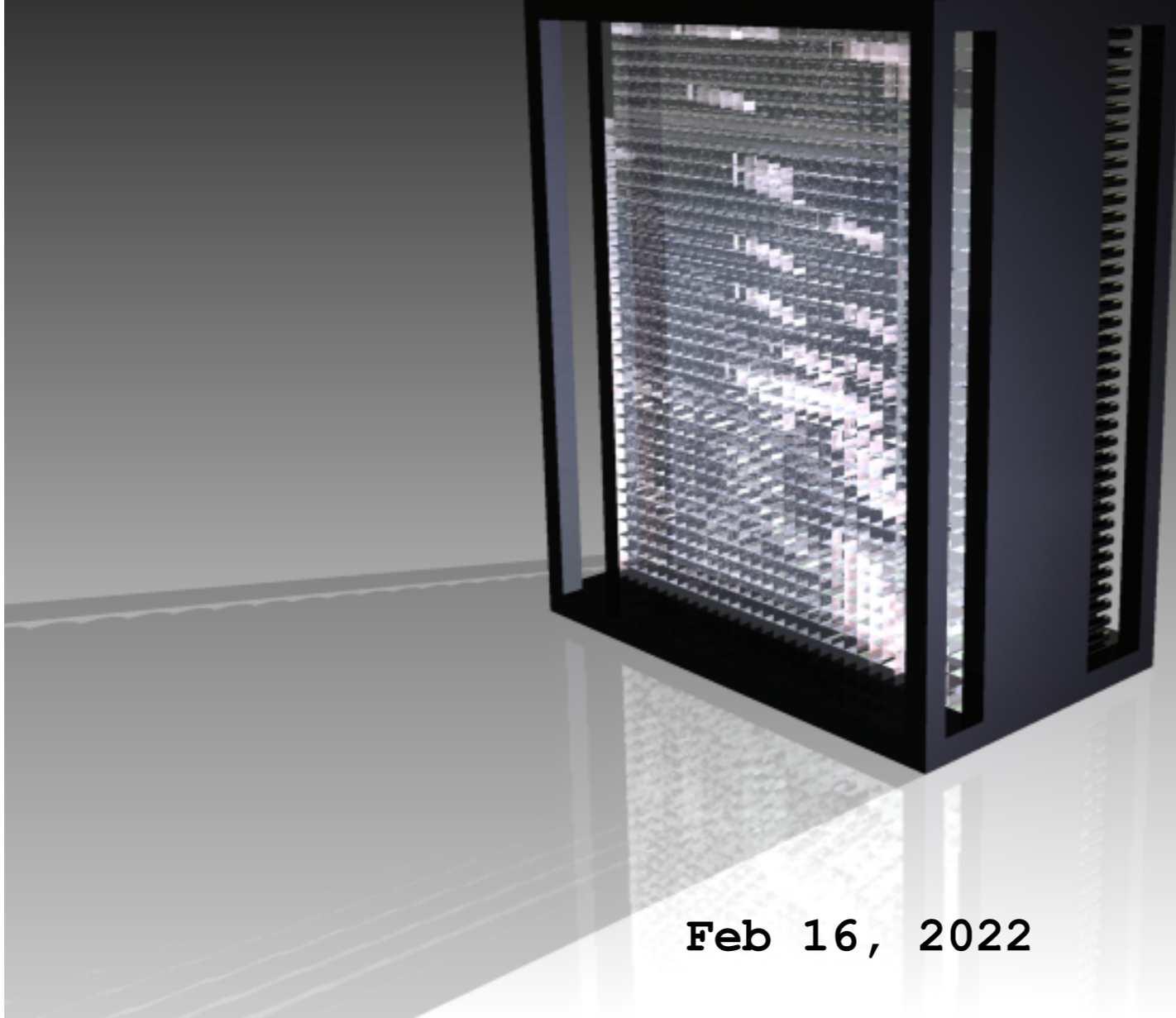


# NPS Software Analyzer Development NPS Collaboration Meeting



Feb 16, 2022

Carlos Yero  
Steve Wood



## What has been done ?

- created NPS analysis software
- tested NPS analysis software  
(See [Hall C Doc-DB NPS Software Update \(Yero 2021\)](#) )
- developed NPS calorimeter cluster algorithm
- S. Wood added code infrastructure to facilitate fADC sample (mode) data readout

## What needs to be done ?

- develop NPS calorimeter calibration code
- test the software when NPS assembly is completed (on test bench)

**Follow us on GitHub to contribute:**

<https://github.com/sawjlab/NPSApp/tree/main>

# What has been done ?

- **created NPS analysis software**
  - NPS C++ classes adapted from HCANA/podd to the NPSApp repository
    - ▶ -> THcNPSAnalyzer
    - ▶ -> THcNPSApparatus
    - ▶ -> THcNPSCalorimeter
      - ◆ -> THcNPSArray
      - ◆ -> THcNPSShowerHit

## NPSApp directory structure

DBASE  
DEF-files

MAPS  
PARAM

README.md  
SCRIPTS

TEMPLATES  
cache

db\_run.dat  
hcana

numbering.jpg src  
raw

replay & utility scripts

NPS source code



# What has been done ?

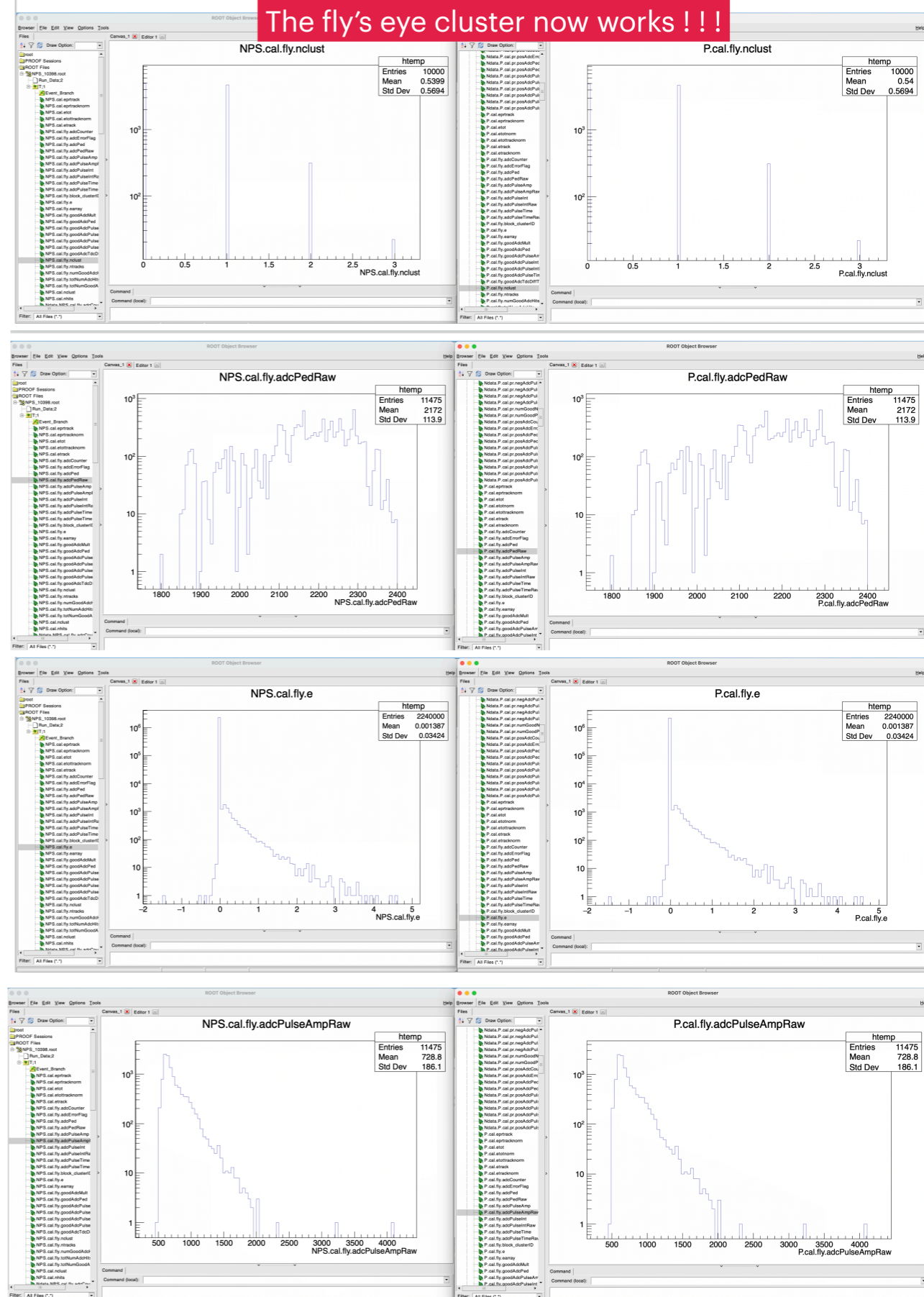
## • tested NPS analysis software

- NPS replay script adapted from `hallc_replay`
  - parameters, maps & database structure also adapted and re-named accordingly
- NPS does **NOT** have tracking detector
  - tracking removed from code
  - tested NPS source code using actual SHMS data (see right plots)
- mimic NPS geometry:
  - use *translation, reflection* maps
  - use `nps_cal_replicate` option
- simulate higher multiplicities:
  - use `SetNevMerge(N)` method

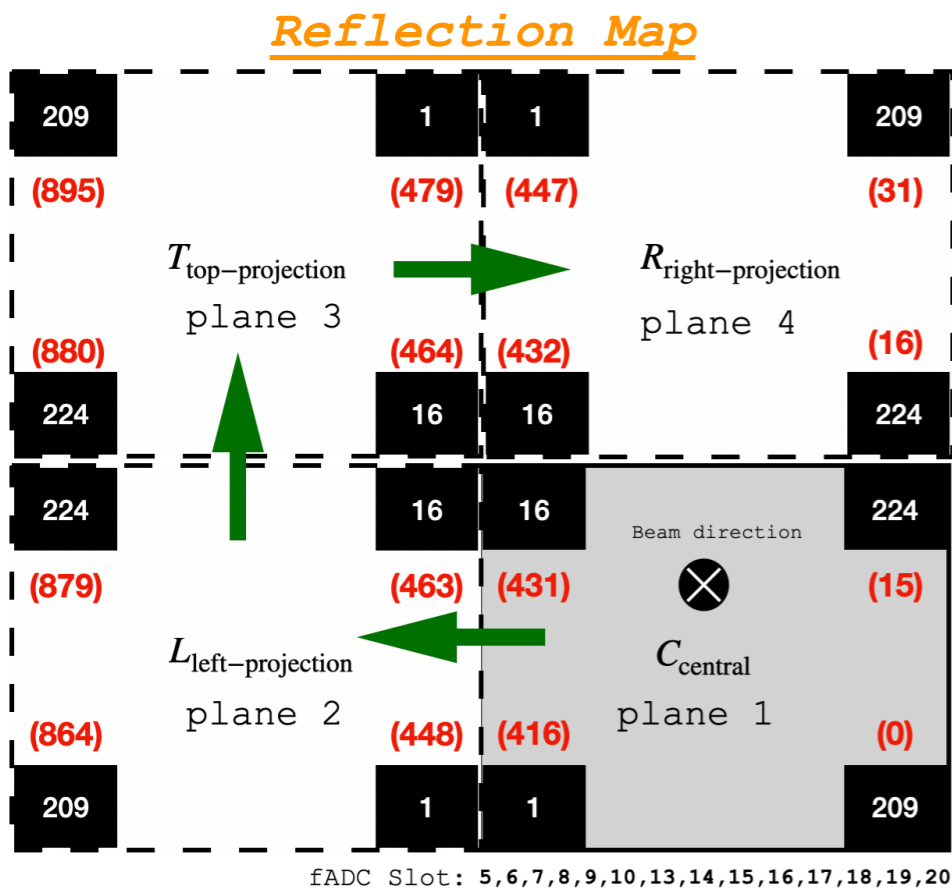
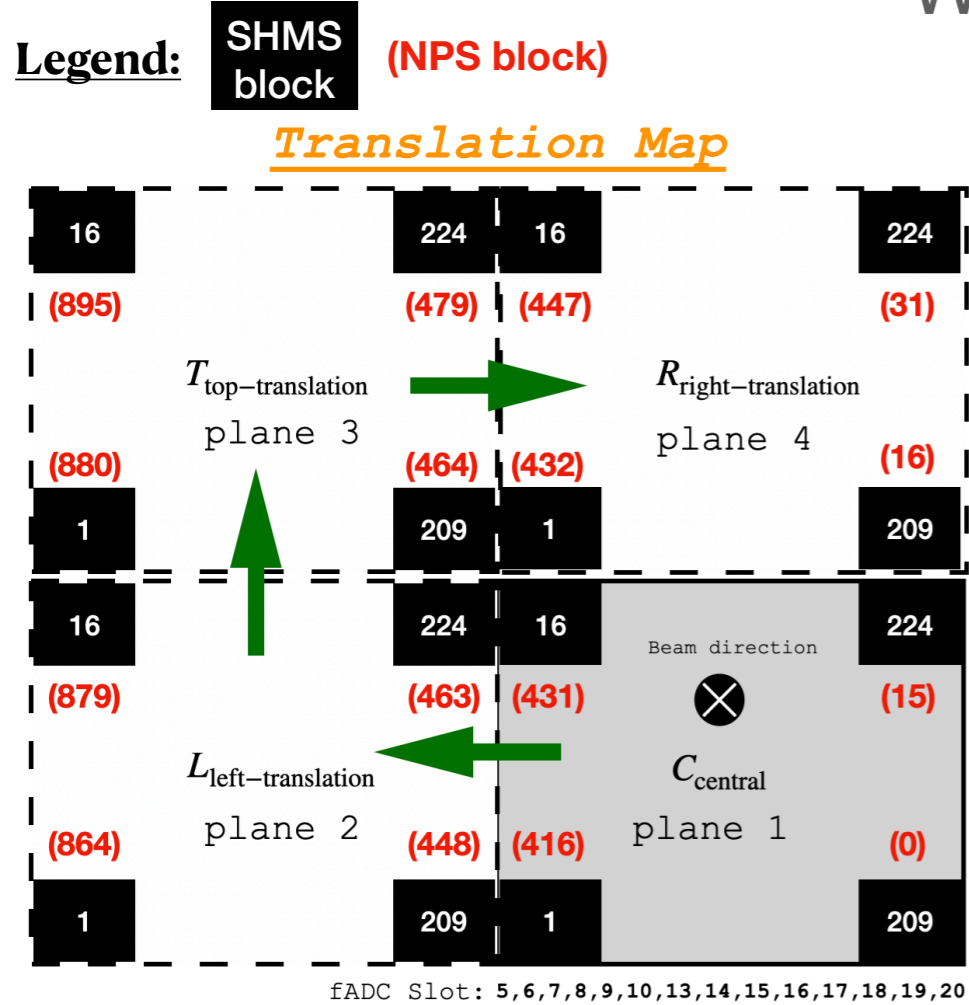
### NPS

### SHMS

The fly's eye cluster now works !!!

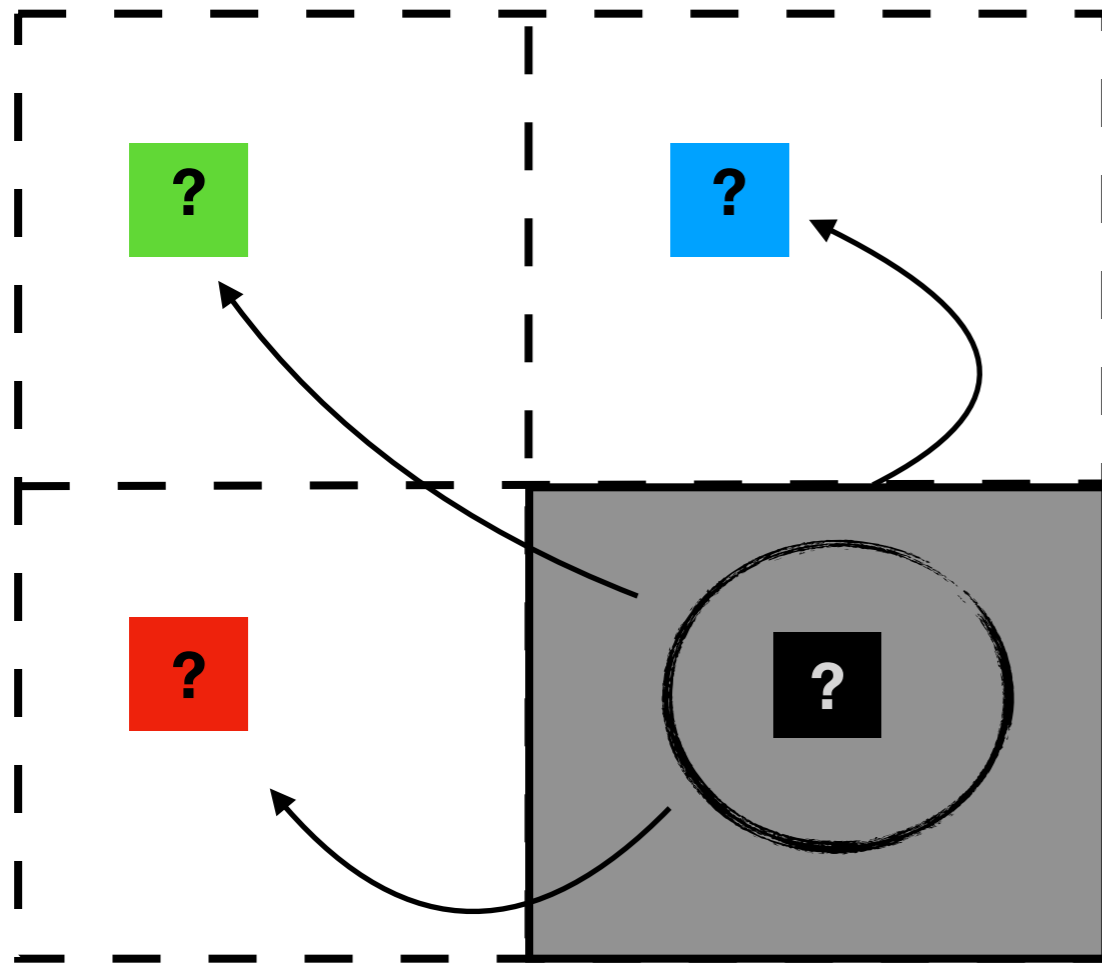


# What has been done ?

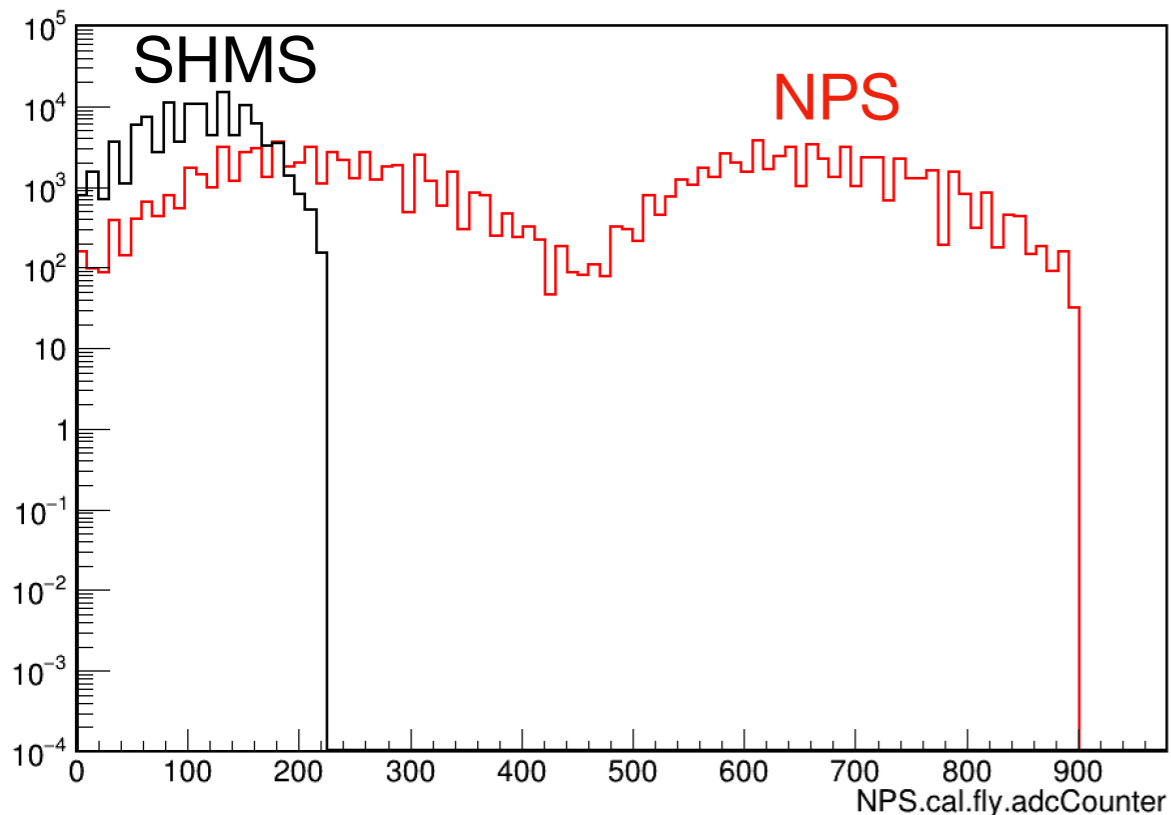


- mimic NPS geometry:
  - use *translation, reflection* maps
- **What was done ?**
  1. modify geometry/signal maps to include the 3 “fake” calorimeter planes
  2. calibration/cuts parameters were copied (x3) to generate 896 parameters “fake” NPS
  3. events *translated* or *reflected* onto 3 remaining “fake” planes (generate higher x4 multiplicity)

## What has been done ?



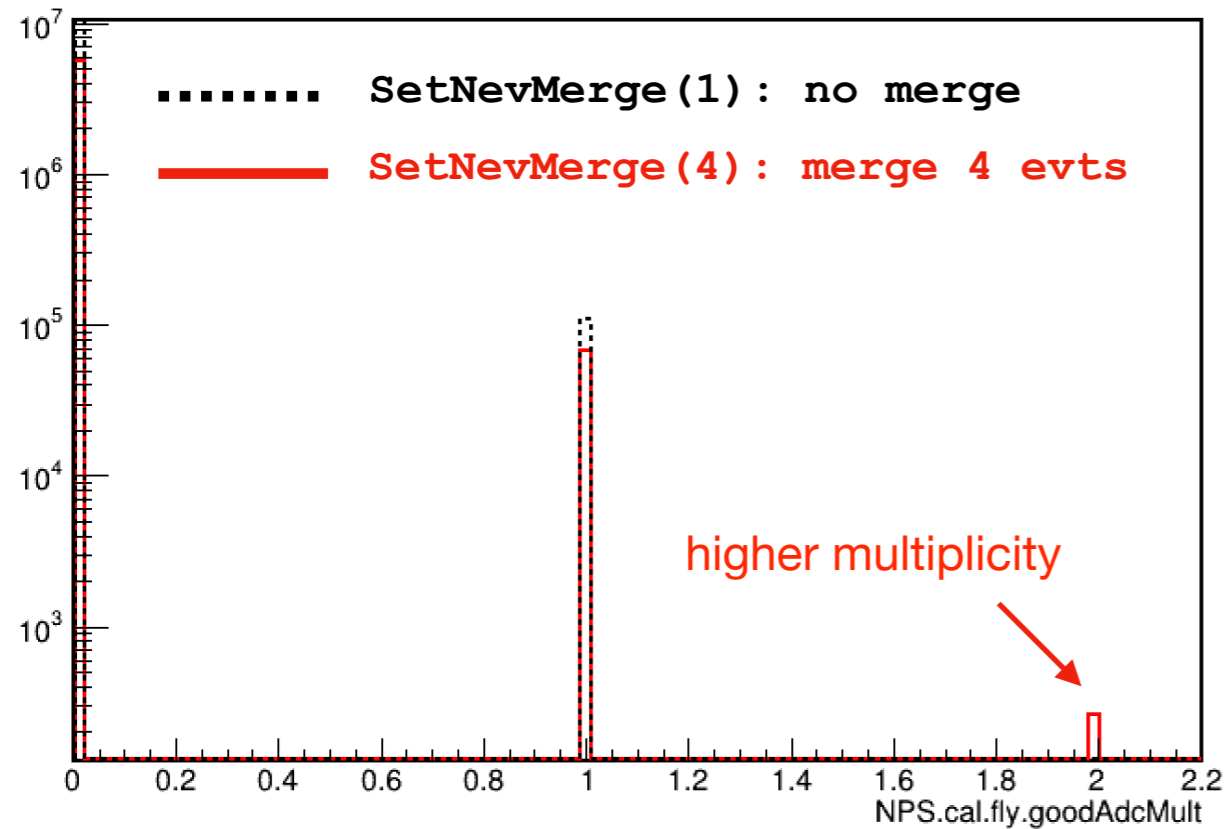
NPS.cal.fly.adcCounter



- **mimic NPS geometry (improved):**
  - use `nps_cal_replicate` option
- **How does it work ?**
  1. per trigger (physics event), select a quadrant at random and move the event to that quadrant by *translation* (`nps_cal_replicate=1`) or *reflection* (`nps_cal_replicate=2`)
  2. requires ONLY signal map from standard SHMS calorimeter since mapping done internally (in `NPSApp` source code)
  3. still need 896 elements (param), as these will be used depending on which quadrant the event lands in

# What has been done ?

NPS.cal.fly.goodAdcMult



- simulate higher multiplicities:
  - ▶ use `SetNevMerge(N)` method  
(call `THcNPSAnalyzer::SetNevMerge(N)` in replay script)
  
- How does it work ?
  - ▶ “merge” hits every  $N$  events into hit list
  - ▶ at  $N$ th event, all hits from previous events are analyzed as if they were from the  $N$ th event
  - ▶ hit list cleared before analysis of next  $N$  events

# What has been done ?

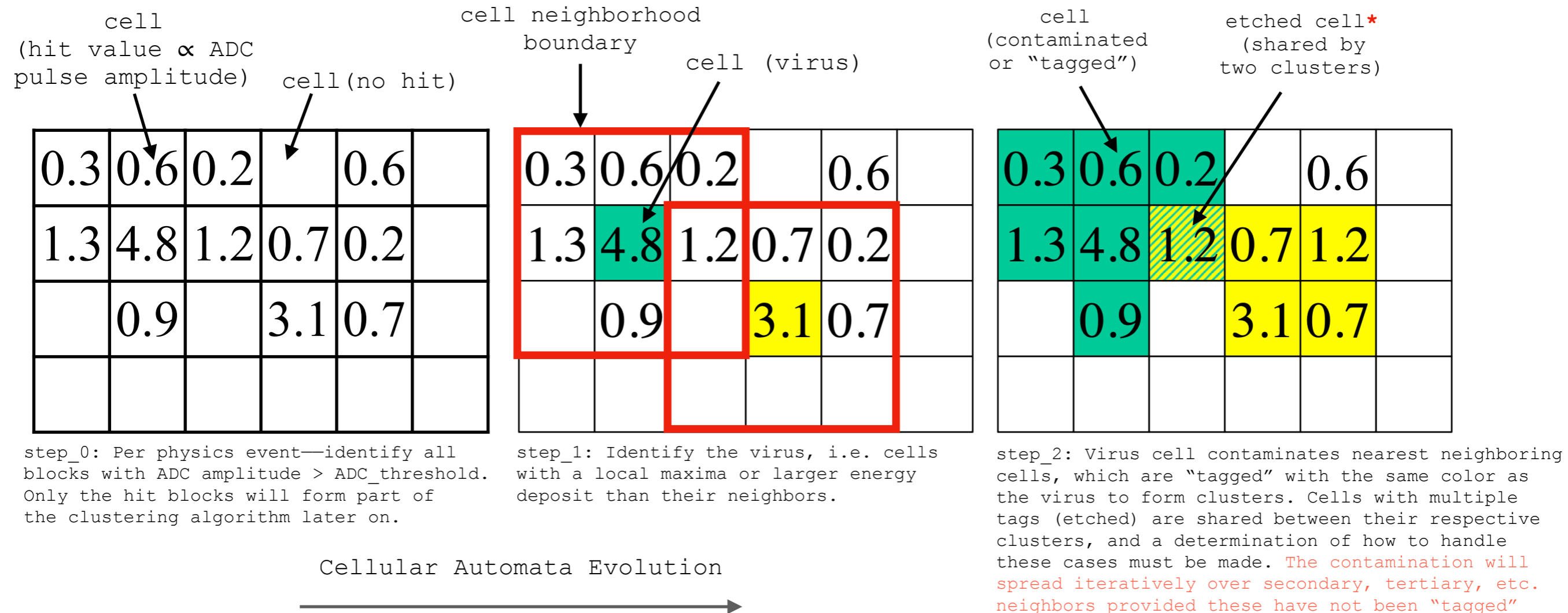
- developed NPS calorimeter cluster algorithm (“Cellular Automata”)
  - rules may be flexible (e.g., may need to re-define what constitutes neighboring cells)
  - How to treat cells with shared clusters ?
    - Do we apply rule 3? or determine how to share energy between two clusters ?

[Y. Roblin et al. \(1995\). Nucl.Instrum.Meth. A362: 478-486](#)

Rule-I A given cell is only sensitive to its eight neighbours. A cell is a virus if its value is higher than the value of each of its neighbours.

Rule-II At a given step, a cell will take the value of its highest energy neighbour.

Rule-III A cell already contaminated in an earlier stage by a virus is immunized against any other virus (restriction to Rule II)





# What has been done ?

○ Currently, both HCANA and NPS “Good Hit” Selection Criteria is as follows:

1. ALL ADC Pulses for ALL calorimeter blocks hit are analyzed per event. (Includes blocks with **Multiple Hits\*\*** per event)
2. To be considered a **‘Good ADC Pulse Hit’** (per block):
  - ◆ Require Pulse Amplitude [mV] > 0 && Raw Pulse Integral [Chan] > 0
  - ◆ Require Pulse Time Difference [ns] (ADC Pulse Time - **StartTime\*\***) to be within an upper and lower bound (set by external per-block parameters)
  - ◆ Require Raw Pulse Integral [Chan] > Threshold (based on pedestal peak)

\*\*If there are still multiple pulses per block per event that passed These conditions, then the 1st hit within the Pulse Time Difference Window will be chosen as the **‘Good ADC Pulse Hit’**
3. The **‘Good ADC Pulse Hits’** are added to a HitSet (to be discussed) which is used by both clustering algorithms to make clusters of hits.

**\*\*Multiple Hits:** I have not yet fully developed the handling of multiple hits. So far, I have made 2D vectors to store the multiple hits per block, however, it is not clear yet how to re-distribute the multiple hits per block in the clustering algorithm. Maybe we can simply add up the pulse integral of multiple hits per block, and consider that when forming clusters?

**\*\*StartTime:** This definition used the hodoscope information in the standard HCANA code however, in the NPS, the StartTime needs to be re-defined, as there are no hodoscopes in the spectrometer. Maybe we can use the HMS reference time (trigger), or the accelerator RF time instead ?

# NPSApp Calorimeter Clustering Algorithm: *Cellular Automata* Approach Animation

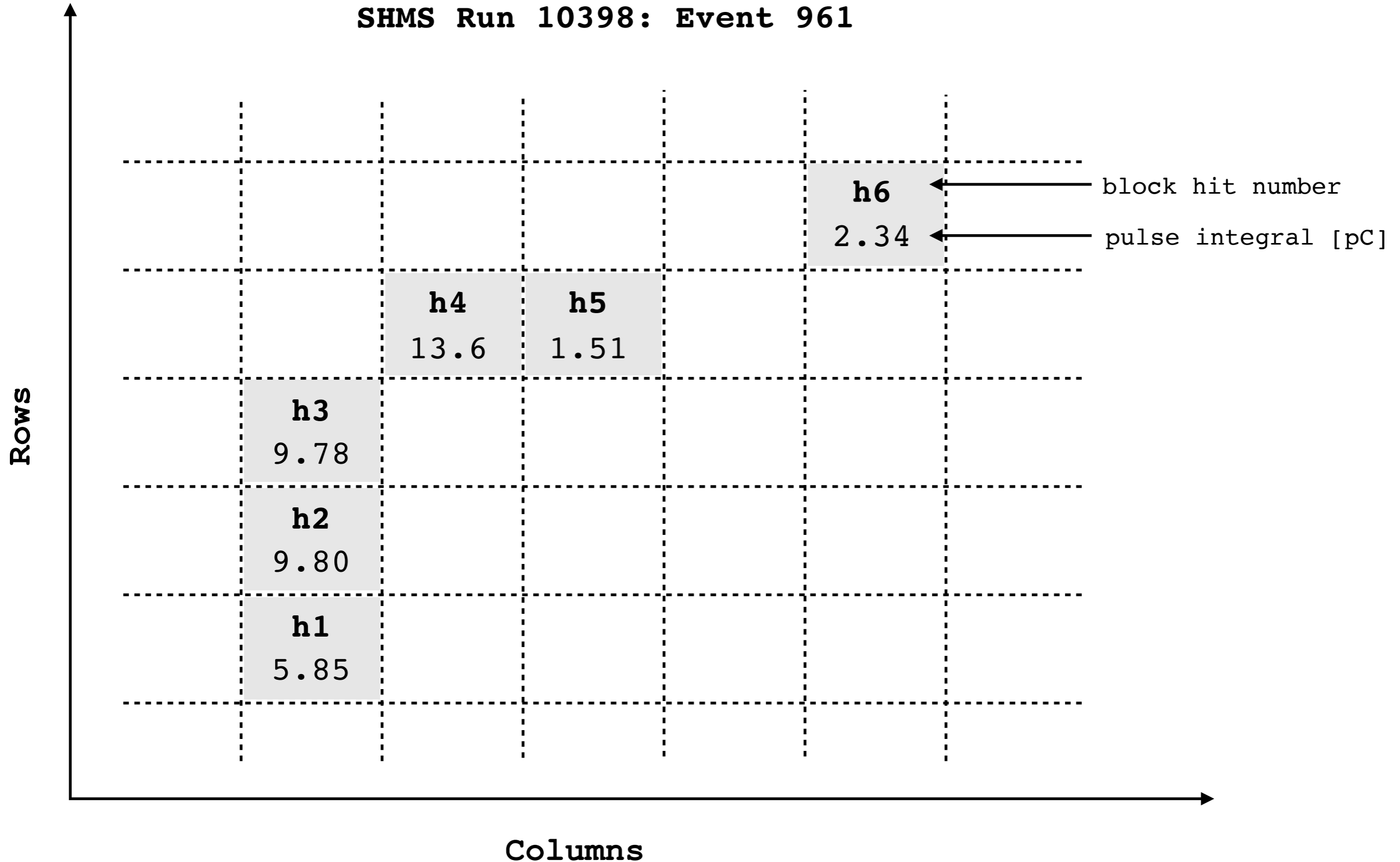
- The Hall C Neutral Particle Spectrometer (NPS) clustering algorithm is in development

Initial results are presented and compared to the **HCANA approach** using SHMS calorimeter data.

# Step 1

- Identify virus blocks. These are blocks that have the largest pulse integral compared to their nearest neighbors\*\*.

SHMS Run 10398: Event 961



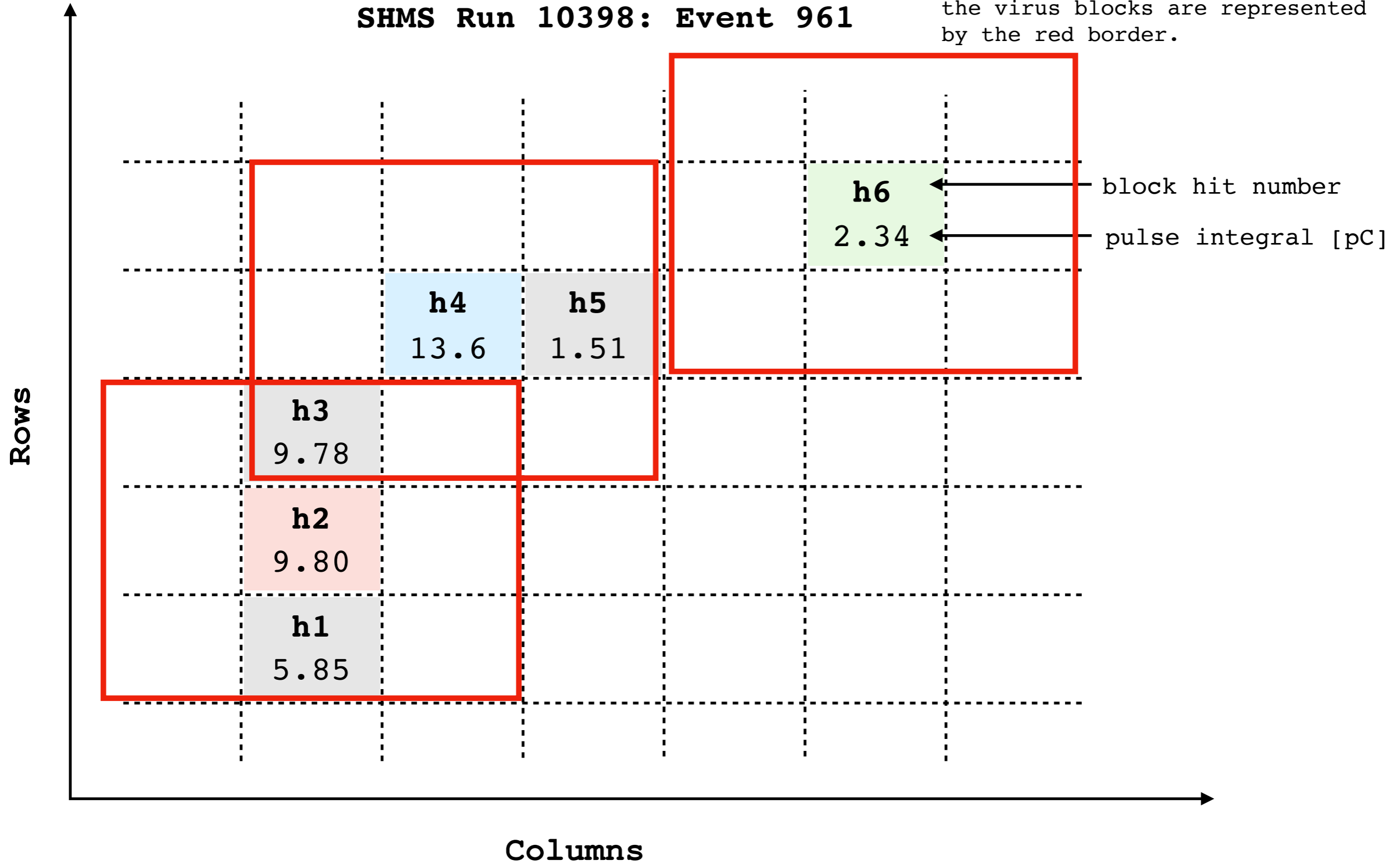
# Step 1

○ The virus blocks identified are shaded in red, blue and green

**\*\*NOTE:**

The nearest by neighbors are identified as the adjacent blocks to the central block in question. The neighborhood of the virus blocks are represented by the red border.

**SHMS Run 10398: Event 961**



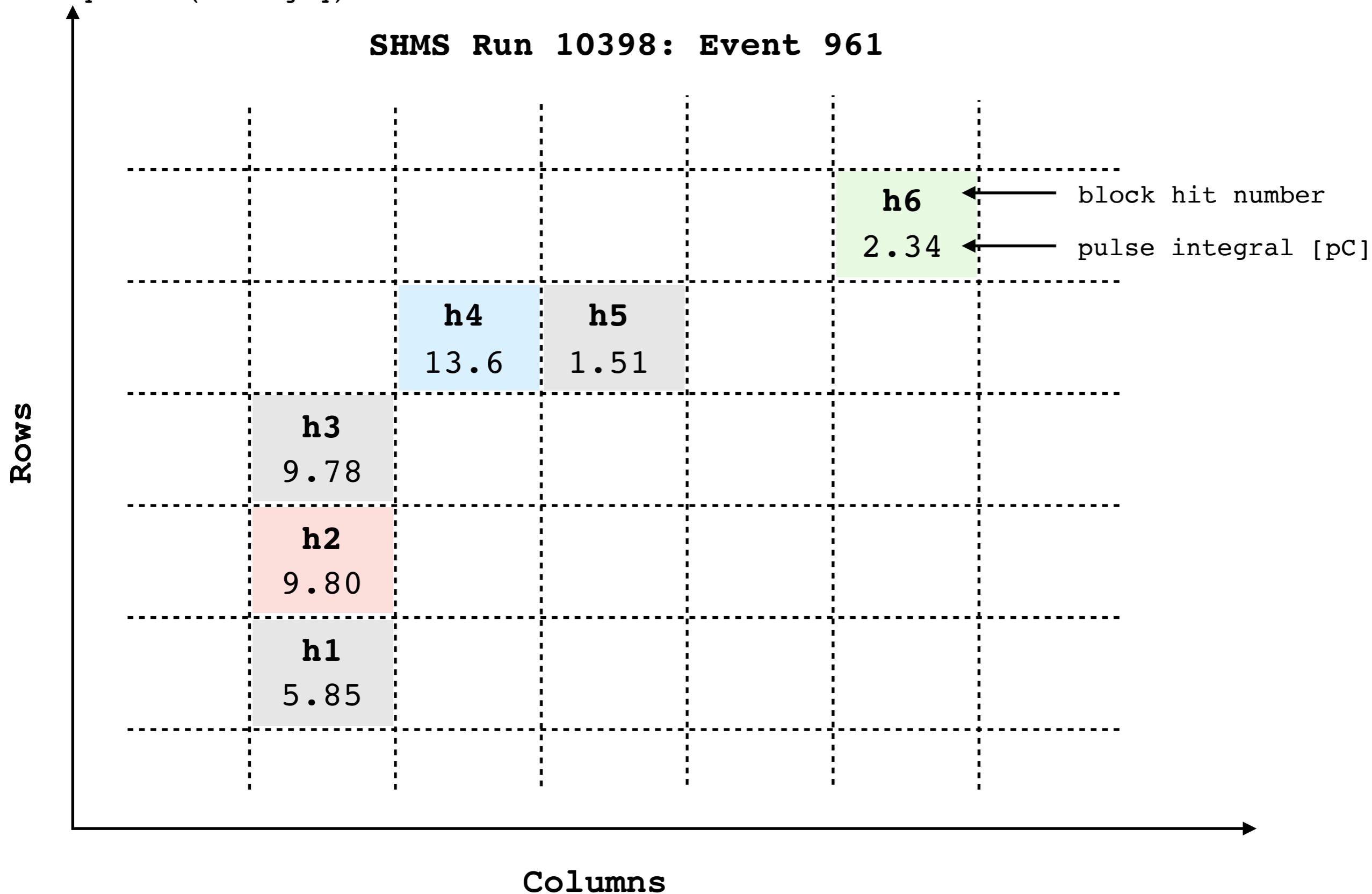
← block hit number

← pulse integral [pC]

# Step 2

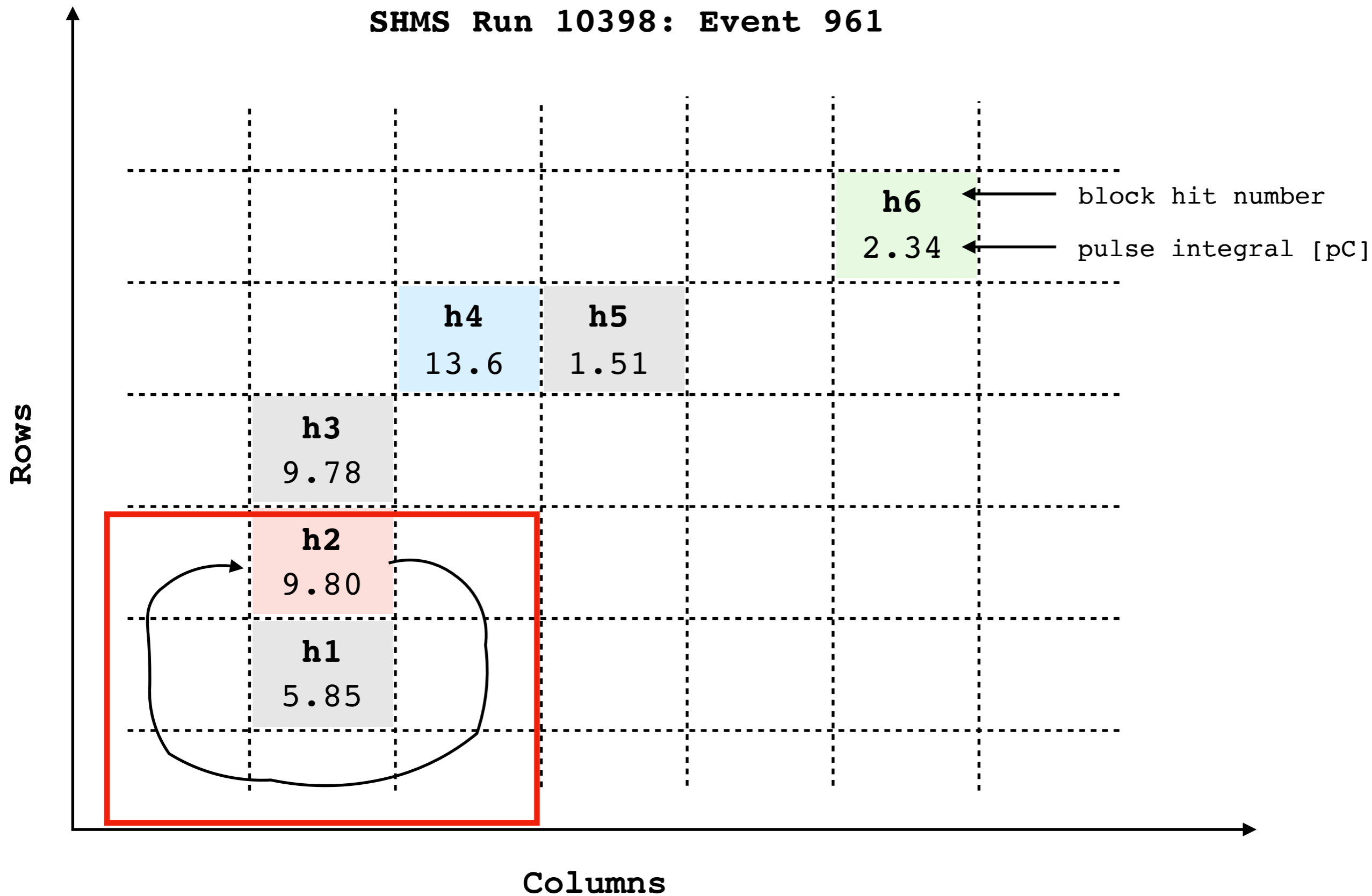
- For each of the non-virus blocks (shaded grey), examine the nearest neighbors.
- Determine the nearest neighbor with the highest pulse integral. If this neighbor is a virus (shaded red, blue or green), then it will "contaminate" the block in question (shaded grey) and make it ill.

## SHMS Run 10398: Event 961



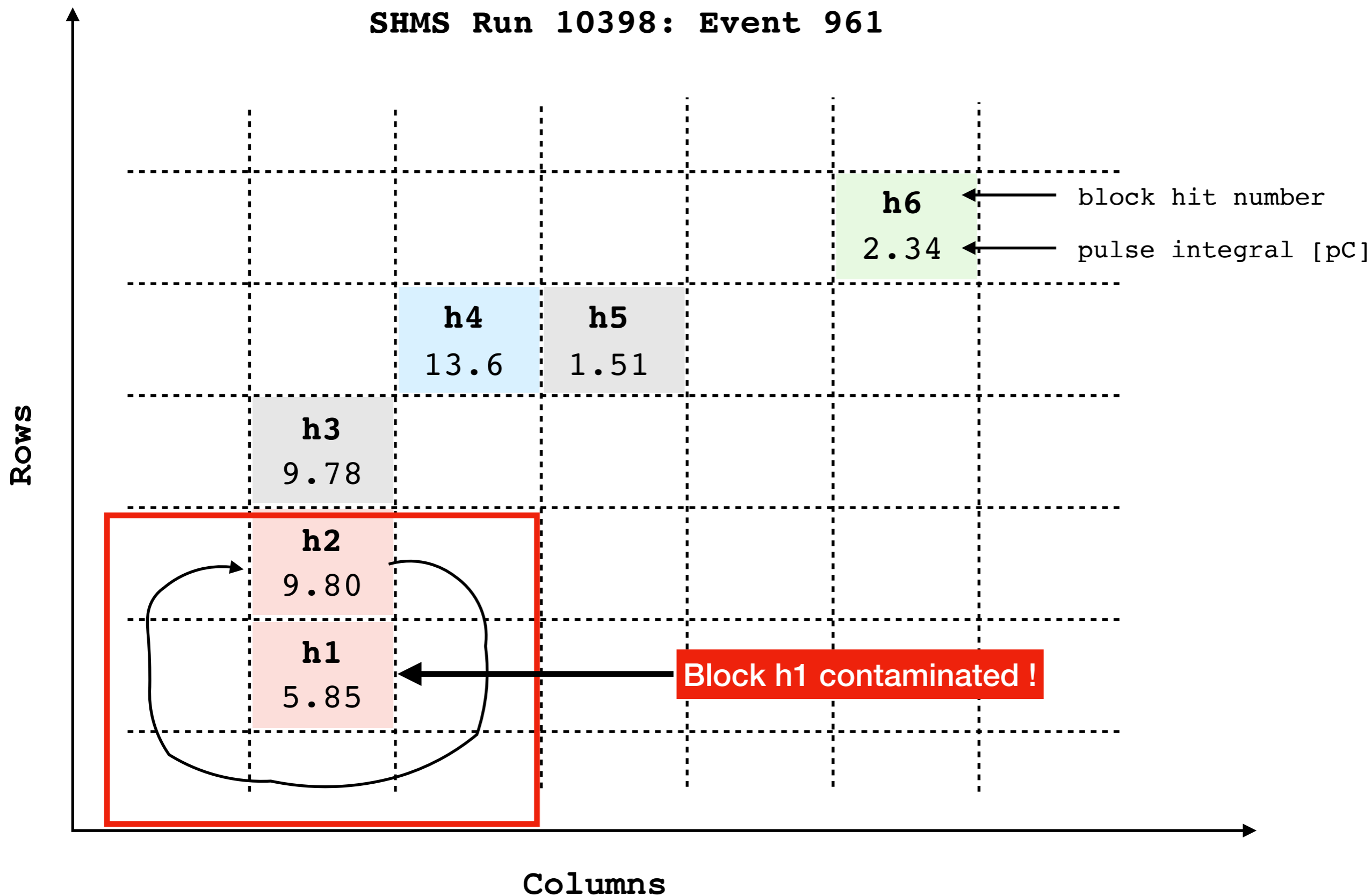
# Step 2

- Examine block h1: Found block h2 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h1 and make it ill.



# Step 2

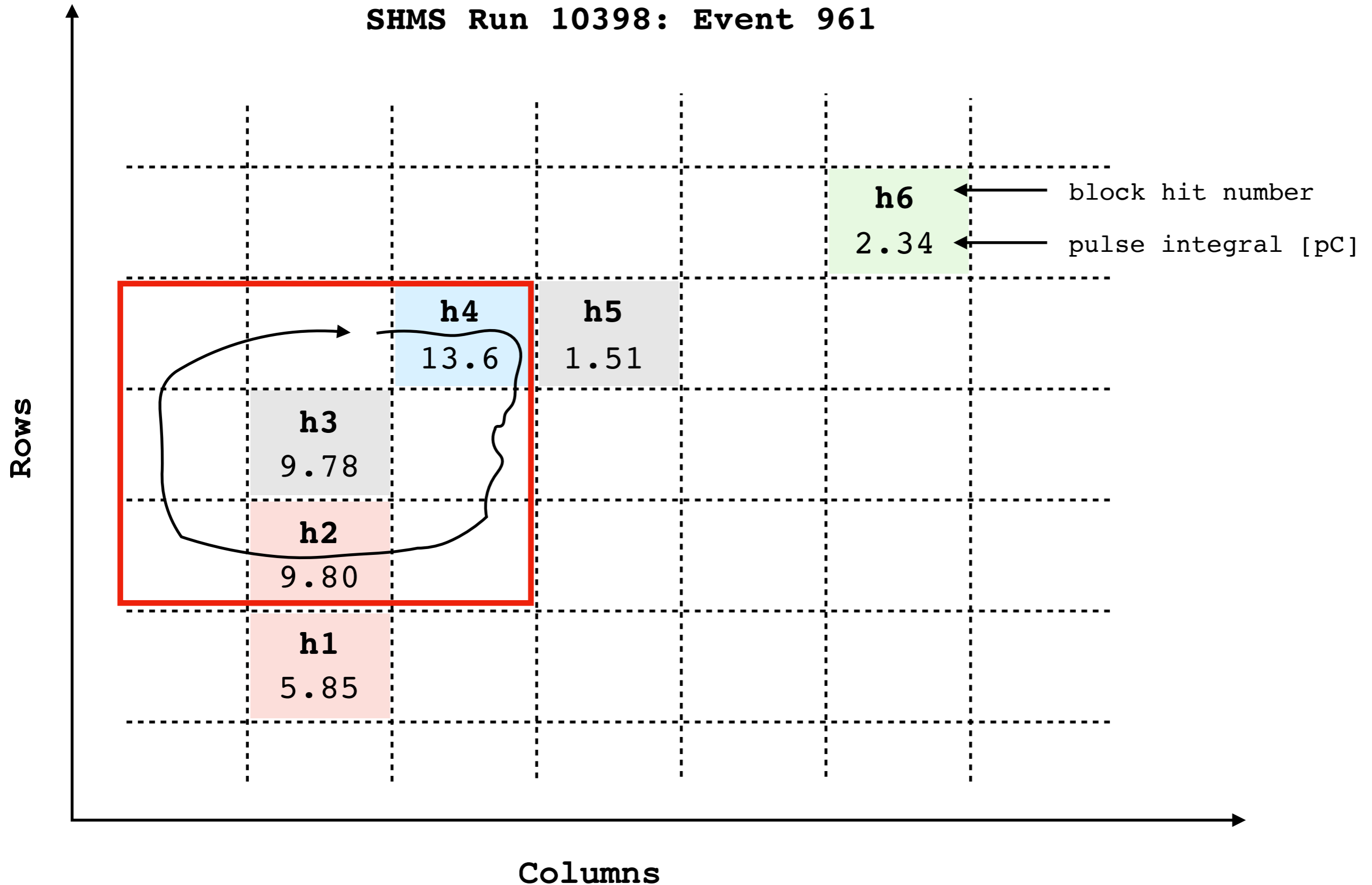
- Examine block h1: Found block h2 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h1 and make it ill.



# Step 2

- Examine block h3: Found block h4 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h3 and make it ill.

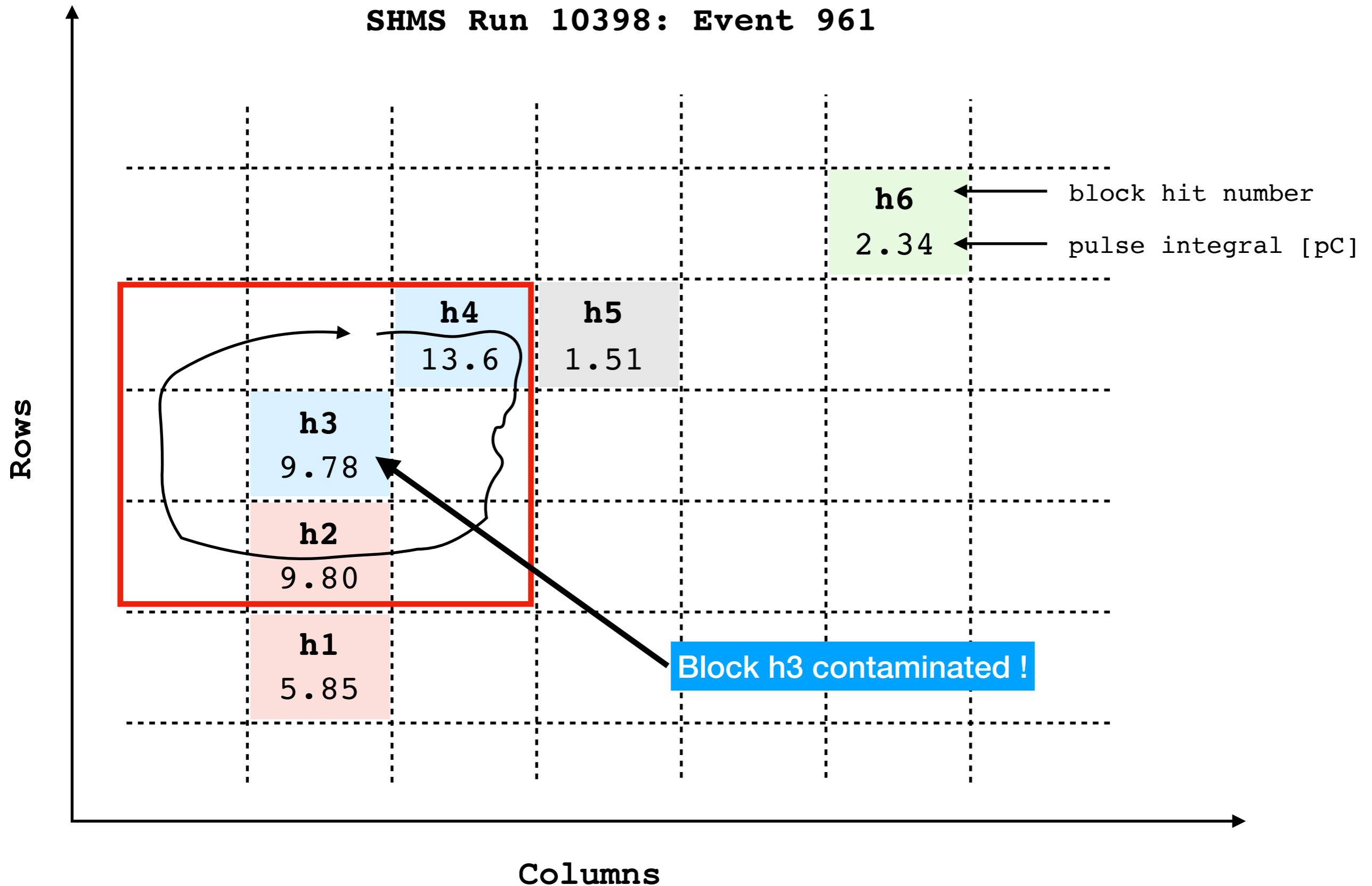
## SHMS Run 10398: Event 961





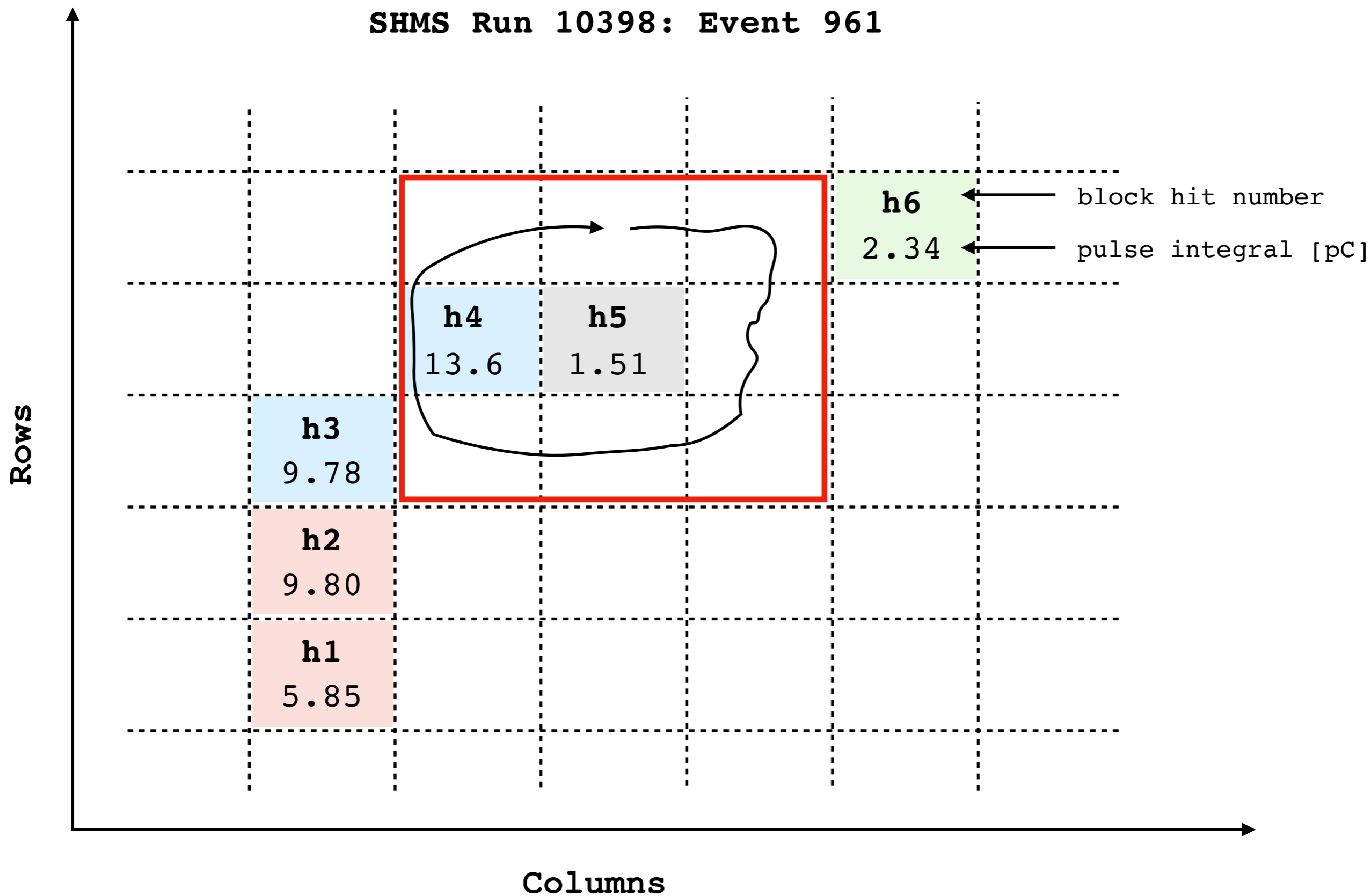
# Step 2

- **Examine block h3:** Found block h4 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h3 and make it ill.



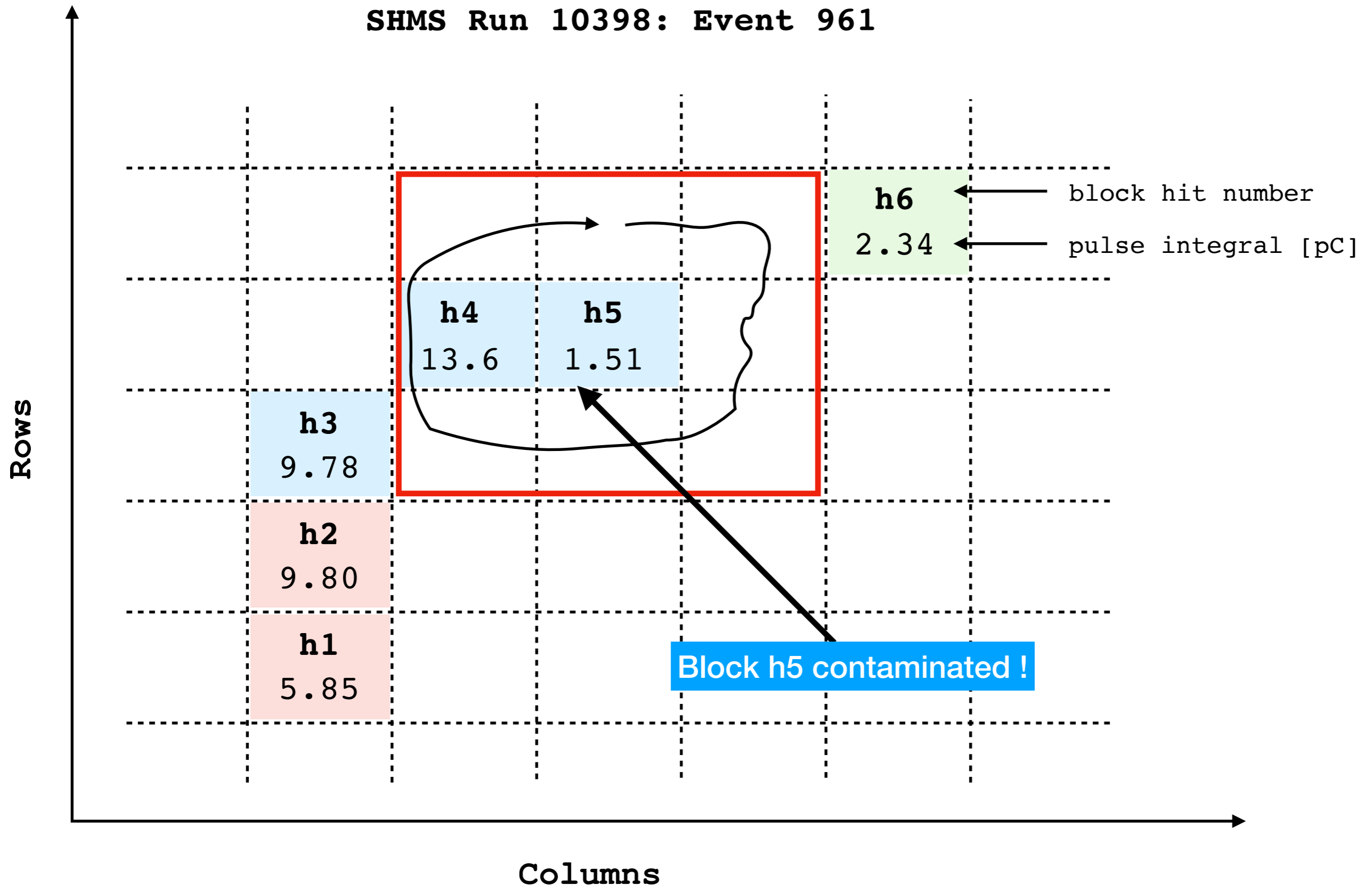
# Step 2

- Examine block h5: Found block h4 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h5 and make it ill.



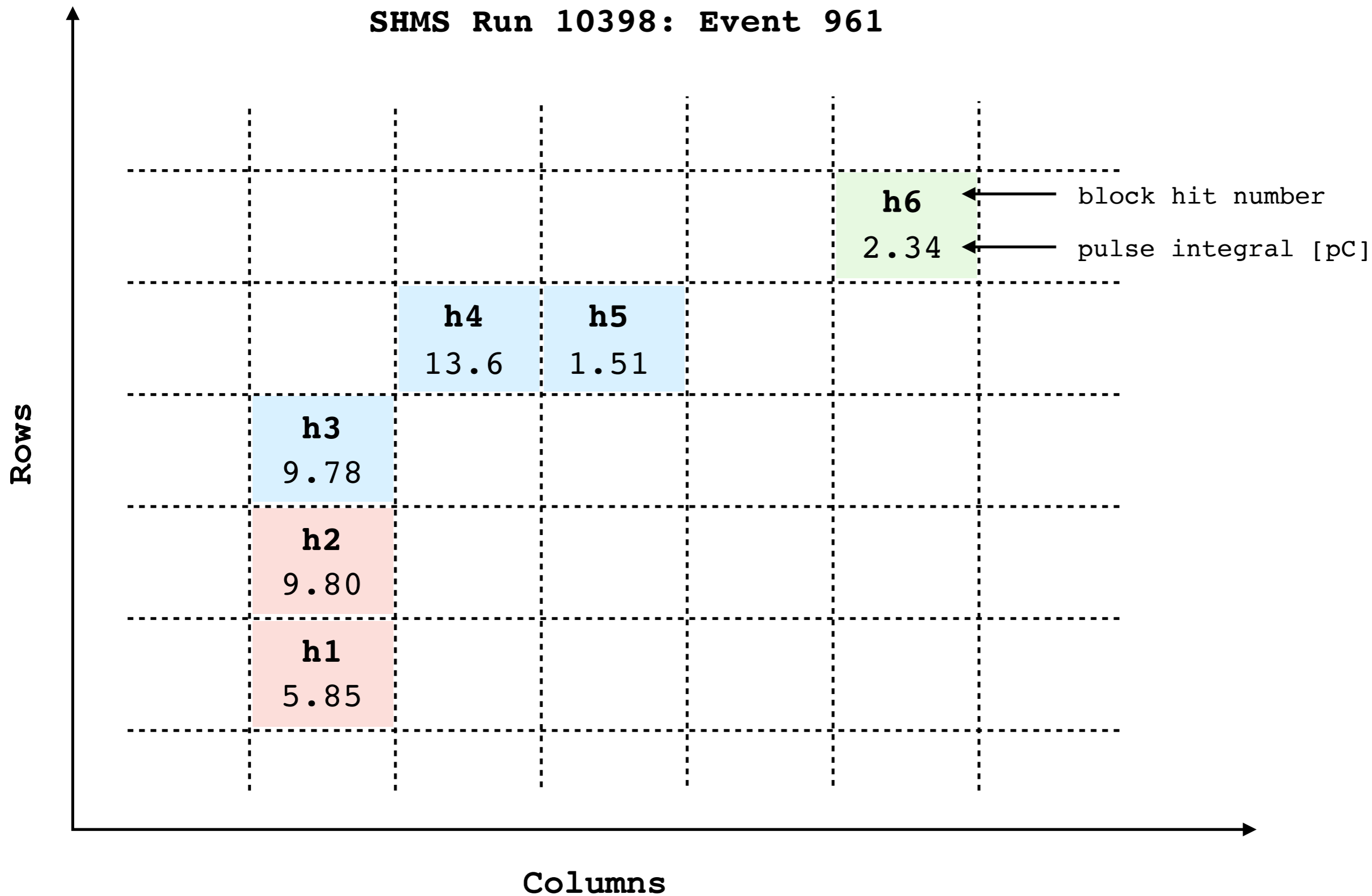
# Step 2

- **Examine block h5:** Found block h4 to be the highest pulse integral and identified as a virus, so it will "contaminate" block h3 and make it ill.



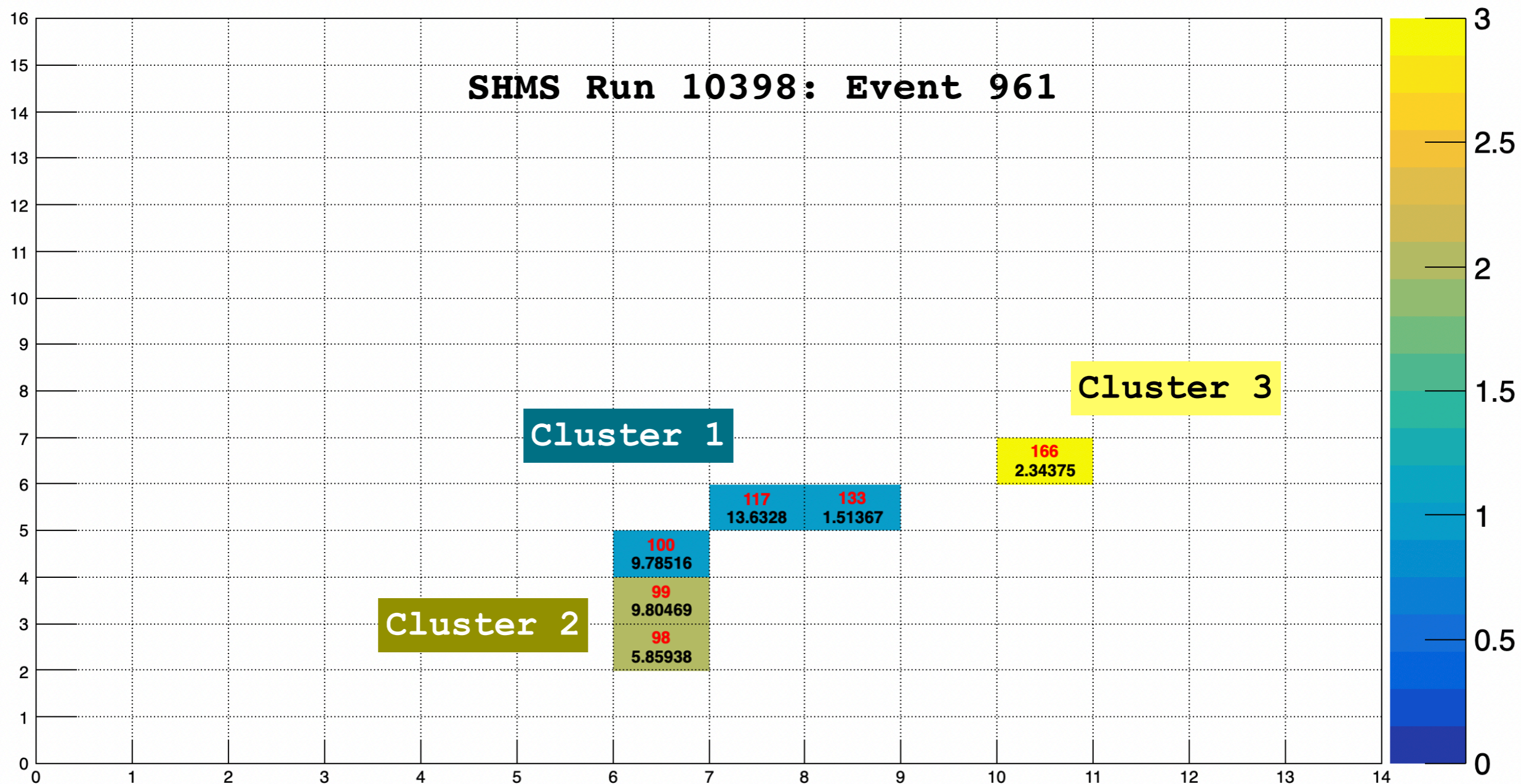
# Step 2

- Overall, there were 3 clusters found, as compared to 2 clusters in the HCANA method presented earlier.



## 2D Grid of the Actual Data Clusters Used in the Example

### SHMS (CellularAutomata) Calorimeter Map (front)



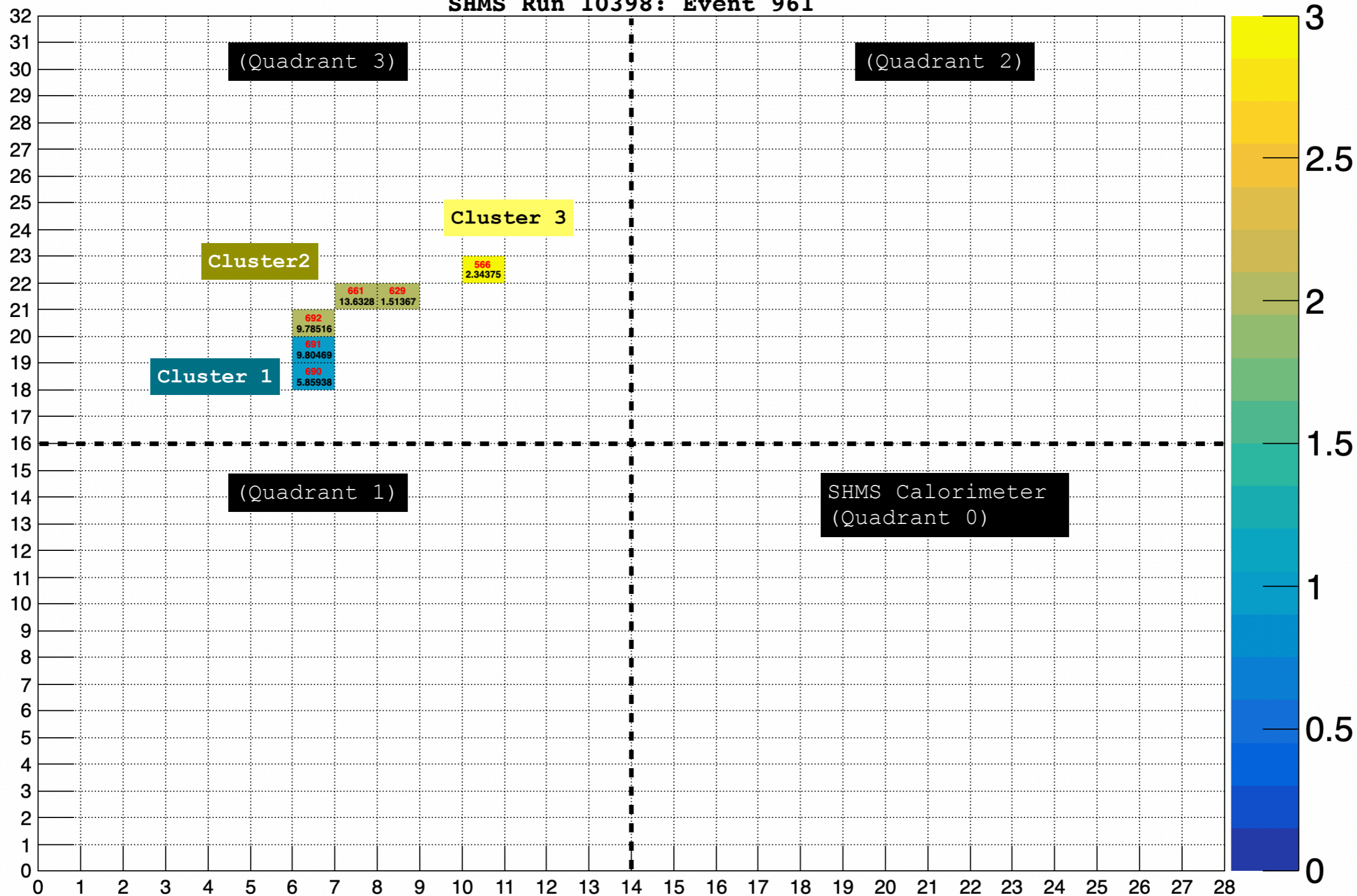
**Legend:**

**Block ID**  
**Pulse Integral [pC]**

\*\*Note: The clusters are color-coded, in integer values of 1 (for cluster 1), 2 (for cluster 2), and so on.

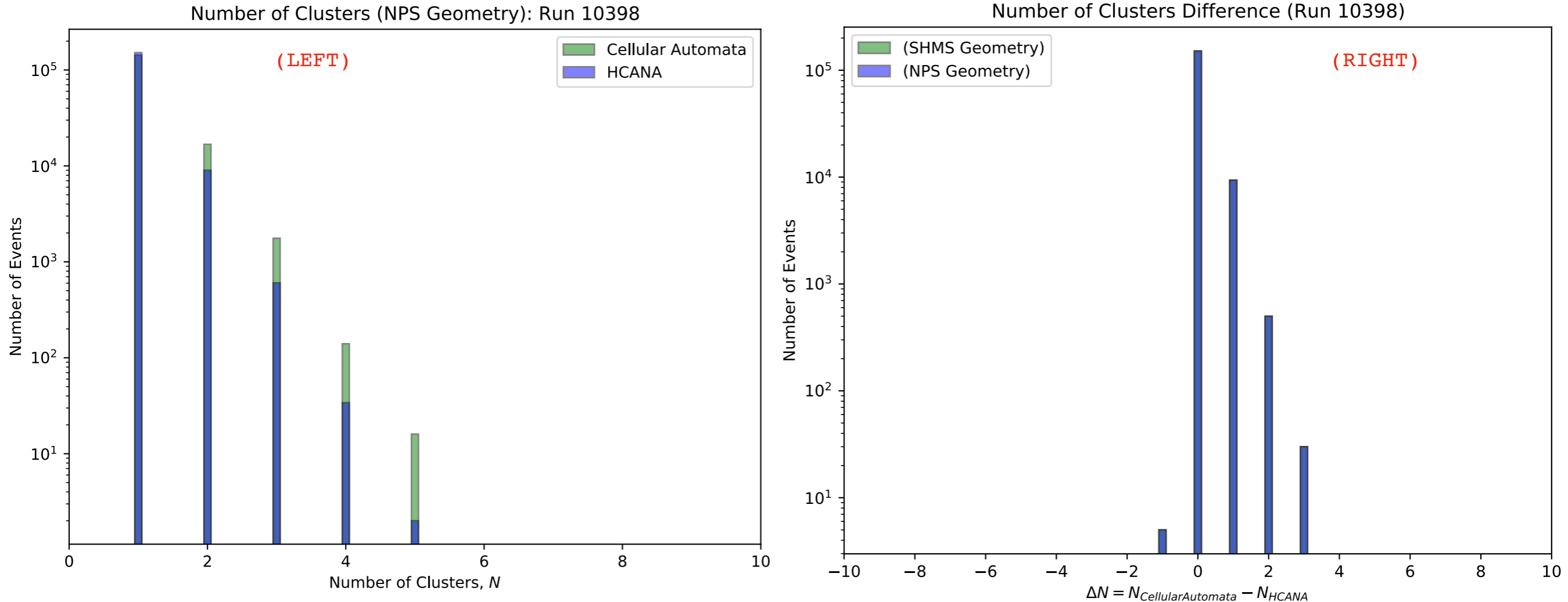
# NPS (CellularAutomata) Calorimeter Map (front)

SHMS Run 10398: Event 961



- Same event shown in previous slide, but using approximate NPS geometry which replicates the SHMS calorimeter geometry (14x16) into three other quadrants.
- The number of clusters is the same, but has been projected at random (during the NPS analysis replay) into one of the four quadrants to simulate the NPS geometry under more realistic conditions.

# Total Number of Clusters Formed with the Different Algorithms



- **(LEFT)** Total number of clusters formed using **Cellular Automata** or standard **HCANA** clustering algorithm.
- **(RIGHT)** Difference (on a per-event basis) between the number of clusters formed using “Cellular Automata” or standard “HCANA” clustering algorithm. A difference of  $\Delta N = 0$  indicates that the two algorithms formed the same number of clusters for a certain number of events, whereas a difference of  $\Delta N = n$  ( $n > 0$ ) indicates that the Cellular Automata formed  $n$  more clusters as compared to the standard HCANA approach.

# Summary

- created and tested NPS analysis software using existing SHMS data
- developed NPS calorimeter cluster algorithm based on “Cellular Automata”
- S. Wood added code infrastructure to facilitate fADC sample (mode) data readout

## What needs to be done ?

- develop NPS calorimeter calibration code (maybe can use SHMS as starting point?)
- NPS software and cluster algorithm will need fine tuning once NPS is fully assembled and on a test bench

**NPS SOFTWARE CONTRIBUTORS NEEDED ! ! !**

Follow us on GitHub to contribute:

<https://github.com/sawjlab/NPSApp/tree/main>