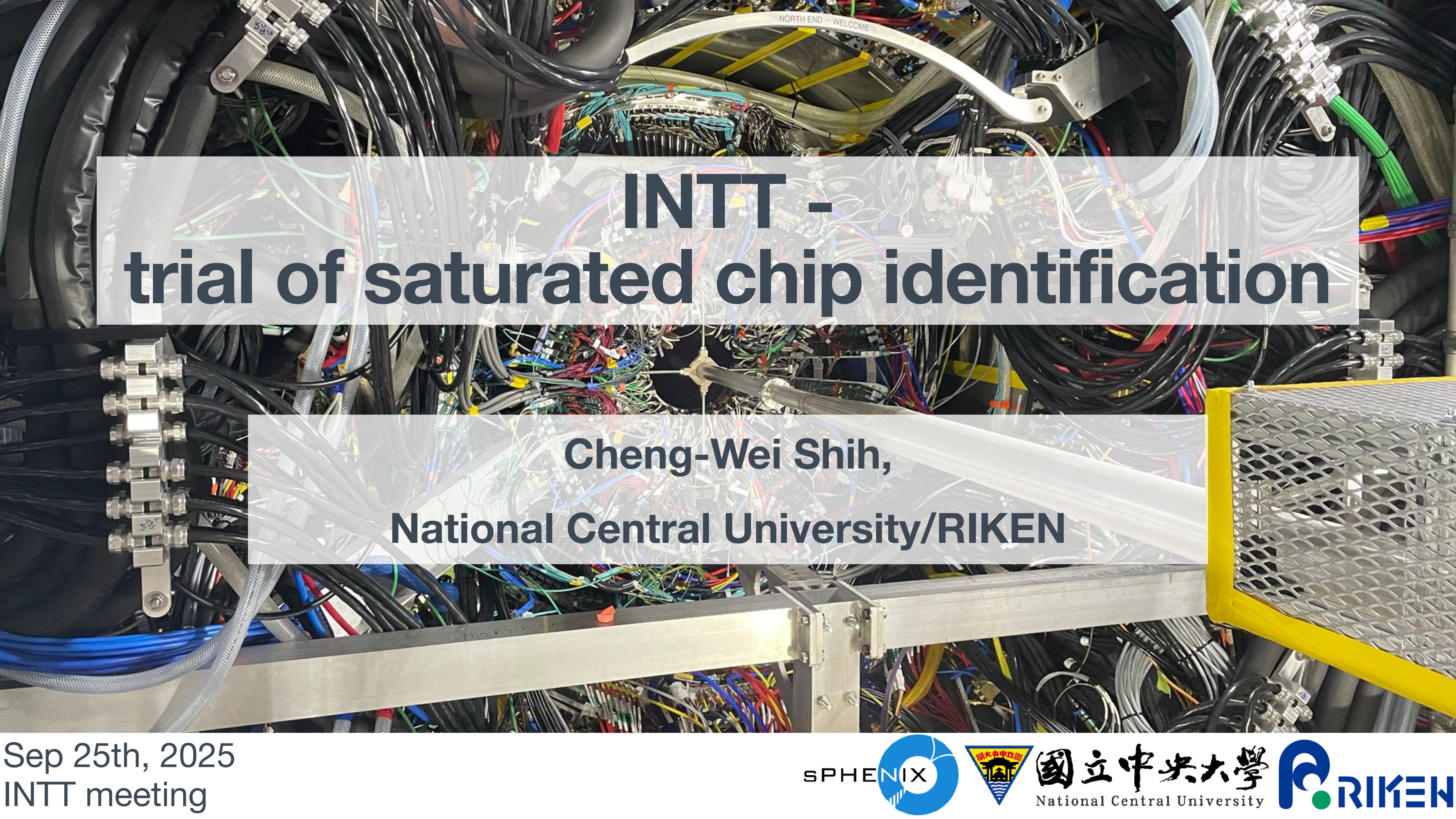




INTT - trial of saturated chip identification

Cheng-Wei Shih,
National Central University/RIKEN

Sep 25th, 2025
INTT meeting



國立中央大學
National Central University



A piece of code of InttCombinedRawDataDecoder

```
////////////////////
// bco filter
if (m_bcomap.IsBad(raw, bco_full, bco) && m_bcoFilter)
{
    // std::cout<<"bad bco removed : "<<raw.felix_server<<" "<<raw.felix_channel<<" "<<raw.chip<<" "<<raw.channel<<std::endl;
    continue;
}

hit_key = InttDefs::genHitKey(ofl.strip_y, ofl.strip_x); // col, row <trackbase/InttDefs.h>
int time_bucket = 0;
if(!m_runStandAlone) // global mode
{
    if(m_triggeredMode)
    {
        time_bucket = (intthit->get_FPHX_BCO() - (intthit->get_bco() & 0x7fU) - m_inttFeeOffset + 128) % 128;
    }
    else // streamed mode
    {
        // For triggered events with the INTT in streaming mode:
        // The BCO corresponding to a given FPHX_BCO is:
        //      intthit->get_FPHX_BCO() + intthit->get_bco() - m_inttFeeOffset
        // The bunch crossing relative to the trigger BCO is then:
        //      (intthit->get_FPHX_BCO() + intthit->get_bco() - m_inttFeeOffset) - gl1bco

        time_bucket = intthit->get_FPHX_BCO() + intthit->get_bco() - gl1bco - m_inttFeeOffset;
    }
}
```

	triggered mode		streaming mode	
	w/ bco_filter	No bco_filter	w/ bco_filter	No bco_filter
global mode	only ± 1 BCO of triggered bco kept, time_bucket = 0	Triggered BCO + extended BCOs kept (Triggered TB at 0) 128 time_bucket values	Should not have such a setting	500 time_bucket values (triggered BCO at 0)
local mode				No doable under current framework (no GL1)

The followings are the working logic of the current codes, if I understand correctly				
global mode	3 time_bucket values	128 time_bucket values	3 time_bucket values (triggered BCO = 0)	500 time_bucket values
local mode	time_bucket = 0 (regardless bco_filter)		time_bucket = 0 (regardless bco_filter)	

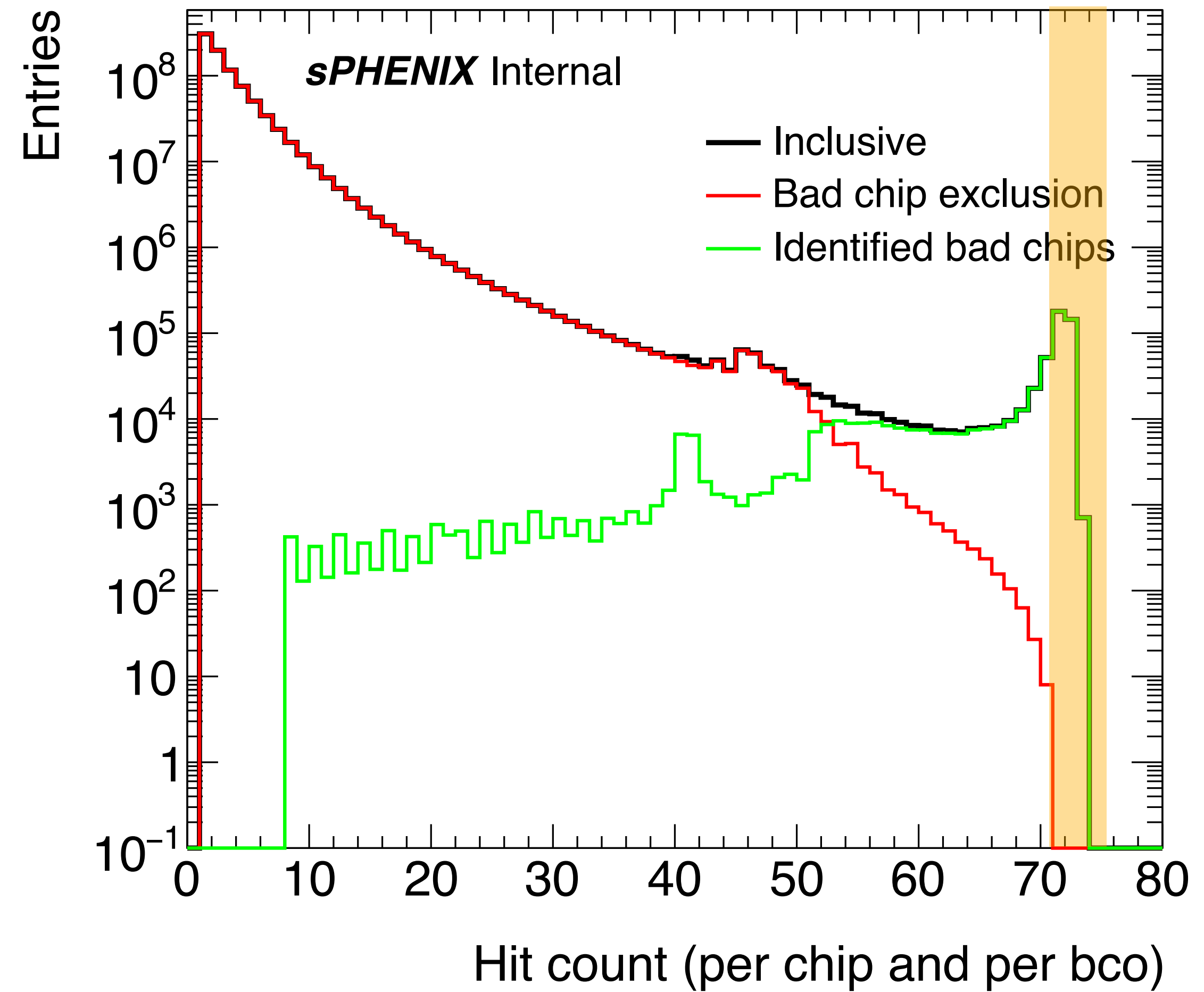
sPHENIX default

The proposed modification: `m_runStandAlone` x `m_triggeredMode` x `m_bcoFilter` = 8 options in total

```
288 //////////////////////////////////////////////////
289 // bco filter
290 if (m_bcomap.IsBad(raw, bco_full, bco) && m_bcoFilter)
291 {
292     // std::cout<<"bad bco removed : "<<raw.felix_server<<" "<<raw.felix_channel<<" "<<raw.chip<<" "<<raw.channel<<std::endl;
293     continue;
294 }
295
296 hit_key = InttDefs::genHitKey(ofl.strip_y, ofl.strip_x); // col, row <trackbase/InttDefs.h>
297 int time_bucket = 0;
298
316 // Note: Case 1: Local, Triggered, With BCO filter
317 if ( m_runStandAlone && m_triggeredMode && m_bcoFilter ) {time_bucket = 0;}
318
319 // Note: Case 2: Local, Triggered, No BCO filter
320 else if ( m_runStandAlone && m_triggeredMode && !m_bcoFilter ) {time_bucket = (intthit->get_FPHX_BCO() - (intthit->get_bco() & 0x7fU) - m_inttFeeOffset + 128) % 128;}
321
322 // Note: Case 3: Local, Streaming, With BCO filter
323 else if ( m_runStandAlone && !m_triggeredMode && m_bcoFilter ) {
324     std::cout<< PHWHERE << "\n" << "You selected INTT local mode, streaming mode, and BCO_filter, which is not supported. Exiting."<< std::endl;
325     gSystem->Exit(1);
326     exit(1);
327 }
328
329 // Note: Case 4: Local, Streaming, No BCO filter
330 else if ( m_runStandAlone && !m_triggeredMode && !m_bcoFilter ) {
331     std::cout<< PHWHERE << "\n" << "You selected INTT local mode, streaming mode, and WITHOUT BCO_filter, but GL1 is not available for time-bucket calculation. Exiting."<< std::endl;
332     gSystem->Exit(1);
333     exit(1);
334 }
335
336 // Note: Case 5: Global, Triggered, With BCO filter
337 else if ( !m_runStandAlone && m_triggeredMode && m_bcoFilter ) {time_bucket = 0;}
338
339 // Note: Case 6: Global, Triggered, No BCO filter
340 else if ( !m_runStandAlone && m_triggeredMode && !m_bcoFilter ) {time_bucket = (intthit->get_FPHX_BCO() - (intthit->get_bco() & 0x7fU) - m_inttFeeOffset + 128) % 128;}
341
342 // Note: Case 7: Global, Streaming, With BCO filter
343 else if ( !m_runStandAlone && !m_triggeredMode && m_bcoFilter ) {
344     std::cout<< PHWHERE << "\n" << "You selected INTT global mode, streaming mode, and BCO_filter, which is not supported. Exiting."<< std::endl;
345     gSystem->Exit(1);
346     exit(1);
347 }
348
349 // Note: Case 8: Global, Streaming, No BCO filter
350 else if ( !m_runStandAlone && !m_triggeredMode && !m_bcoFilter ) {time_bucket = intthit->get_FPHX_BCO() + intthit->get_bco() - gl1bco - m_inttFeeOffset;}
```

For discarding the saturated chips

As discussed last week, we can first try to remove chips with nchip hit > 71



Original structure of the InttCombinedRawDataDecoder

```
int process_event(PHCompositeNode* topNode){
```

Initialization:

```
TrkrHitSetContainer  
InttRawHitContainer  
Gl1Packet  
INTTEVENTHEADER
```

```
for (unsigned int i = 0; i < inttcont -> get_nhits(); i++) {  
    BadChannel Rejection  
    BCO filter  
  
    Define time_bucket  
  
    hit_key & hit_set_key preparation  
    DAC value calibration  
    Add the hit  
}  
}
```

New structure of the InttCombinedRawDataDecoder

```
int process_event(PHCompositeNode* topNode){
```

Initialization:

```
TrkrHitSetContainer  
InttRawHitContainer  
Gl1Packet  
INTTEVENTHEADER
```

```
int loop_start = (m_Saturated_rm)? 0 : 1;  
for (int loop = loop_start; loop < 2; loop++){  
    for (unsigned int i = 0; i < inttcont -> get_nhits(); i++) {  
        BadChannel Rejection  
        BCO filter  
        Define time_bucket  
  
        if (loop == 0){  
            chip_hit_counting;  
            continue;  
        }  
        if (loop == 1 && m_Saturated_rm && chip_hit >= 71) {continue;}  
  
        hit_key & hit_set_key preparation  
        DAC value calibration  
        Add the hit  
    }  
}  
}
```

New structure of the InttCombinedRawDataDecoder

If you set the `m_Saturated_rm` to be `false`

`loop_start = 1`

Then it's the same as before

```
int process_event(PHCompositeNode* topNode){
```

Initialization:

```
TrkrHitSetContainer  
InttRawHitContainer  
Gl1Packet  
INTTEVENTHEADER
```

```
int loop_start = (m_Saturated_rm)? 0 : 1;  
for (int loop = loop_start; loop < 2; loop++){  
    for (unsigned int i = 0; i < inttcont -> get_nhits(); i++) {  
        BadChannel Rejection  
        BCO filter  
        Define time_bucket
```

```
        if (loop == 0){  
            chip_hit_counting;  
            continue;  
        }  
        if (loop == 1 && m_Saturated_rm && chip_hit >= 71) {continue;}
```

```
        hit_key & hit_set_key preparation  
        DAC value calibration  
        Add the hit
```

```
    }  
}  
}
```

New structure of the InttCombinedRawDataDecoder

If you set the `m_Saturated_rm` to be `true`

`loop_start = 0`

When `loop = 0`

```
int process_event(PHCompositeNode* topNode){
```

Initialization:

```
TrkrHitSetContainer  
InttRawHitContainer  
Gl1Packet  
INTTEVENTHEADER
```

```
int loop_start = (m_Saturated_rm)? 0 : 1;
```

```
for (int loop = loop_start; loop < 2; loop++){
```

```
    for (unsigned int i = 0; i < inttcont -> get_nhits(); i++) {
```

```
        BadChannel Rejection
```

```
        BCO filter
```

```
        Define time_bucket
```

```
        if (loop == 0){
```

```
            chip_hit_counting;
```

```
            continue;
```

```
        }
```

```
        if (loop == 1 && m_Saturated_rm && chip_hit >= 71) {continue;}
```

```
        hit_key & hit_set_key preparation
```

```
        DAC value calibration
```

```
        Add the hit
```

```
    }
```

```
}
```

```
}
```

New structure of the InttCombinedRawDataDecoder

If you set the `m_Saturated_rm` to be `true`

`loop_start = 0`

When `loop = 1`

```
int process_event(PHCompositeNode* topNode){
```

Initialization:

```
TrkrHitSetContainer  
InttRawHitContainer  
Gl1Packet  
INTTEVENTHEADER
```

```
int loop_start = (m_Saturated_rm)? 0 : 1;
```

```
for (int loop = loop_start; loop < 2; loop++){
```

```
    for (unsigned int i = 0; i < inttcont -> get_nhits(); i++) {
```

```
        BadChannel Rejection
```

```
        BCO filter
```

```
        Define time_bucket
```

```
        if (loop == 0){  
            chip_hit_counting;  
            continue;  
        }
```

```
        if (loop == 1 && m_Saturated_rm && chip_hit >= 71) {continue;}
```

```
        hit_key & hit_set_key preparation
```

```
        DAC value calibration
```

```
        Add the hit
```

```
    }
```

```
}
```

```
}
```

The modification, please take a look

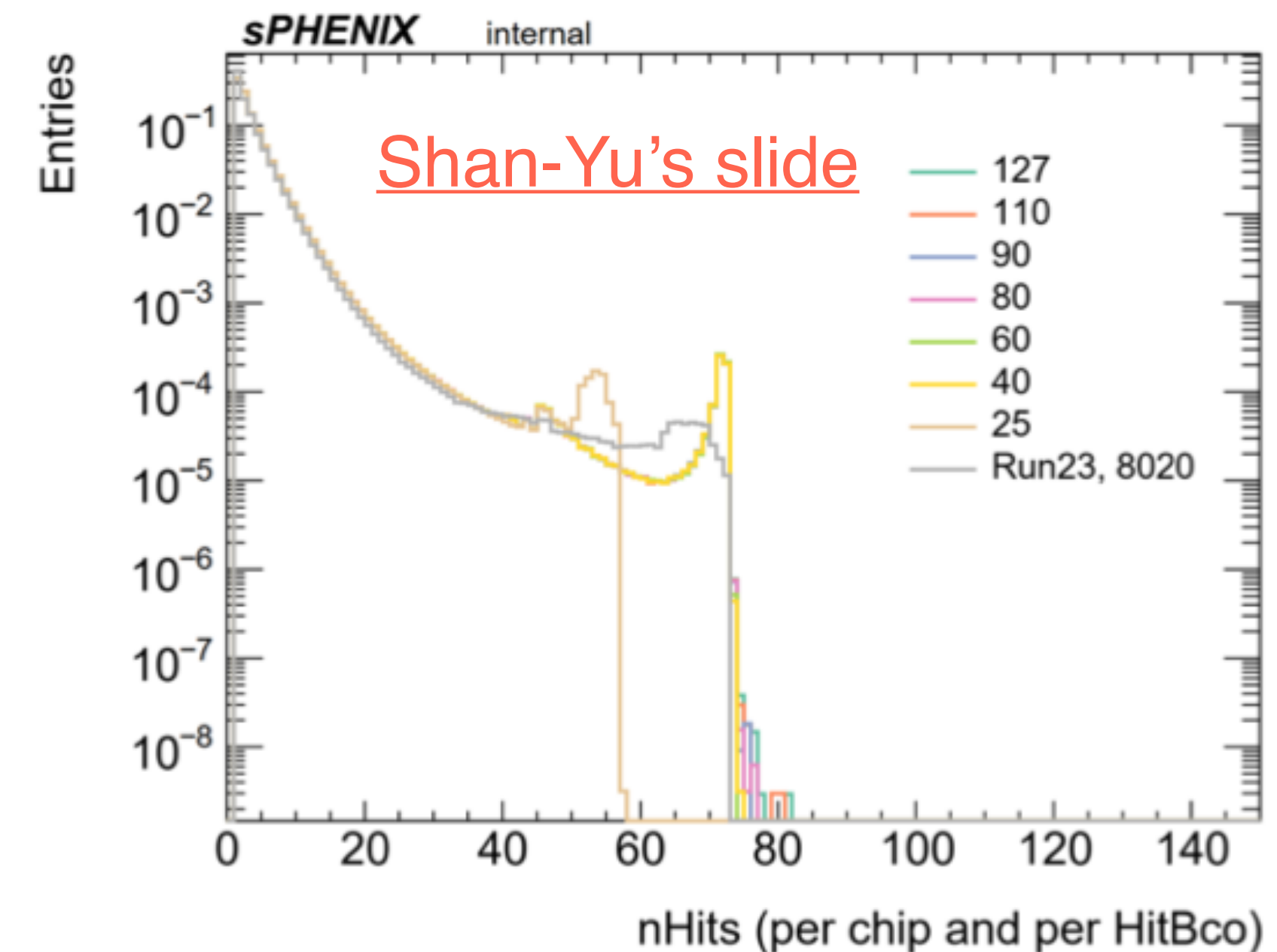
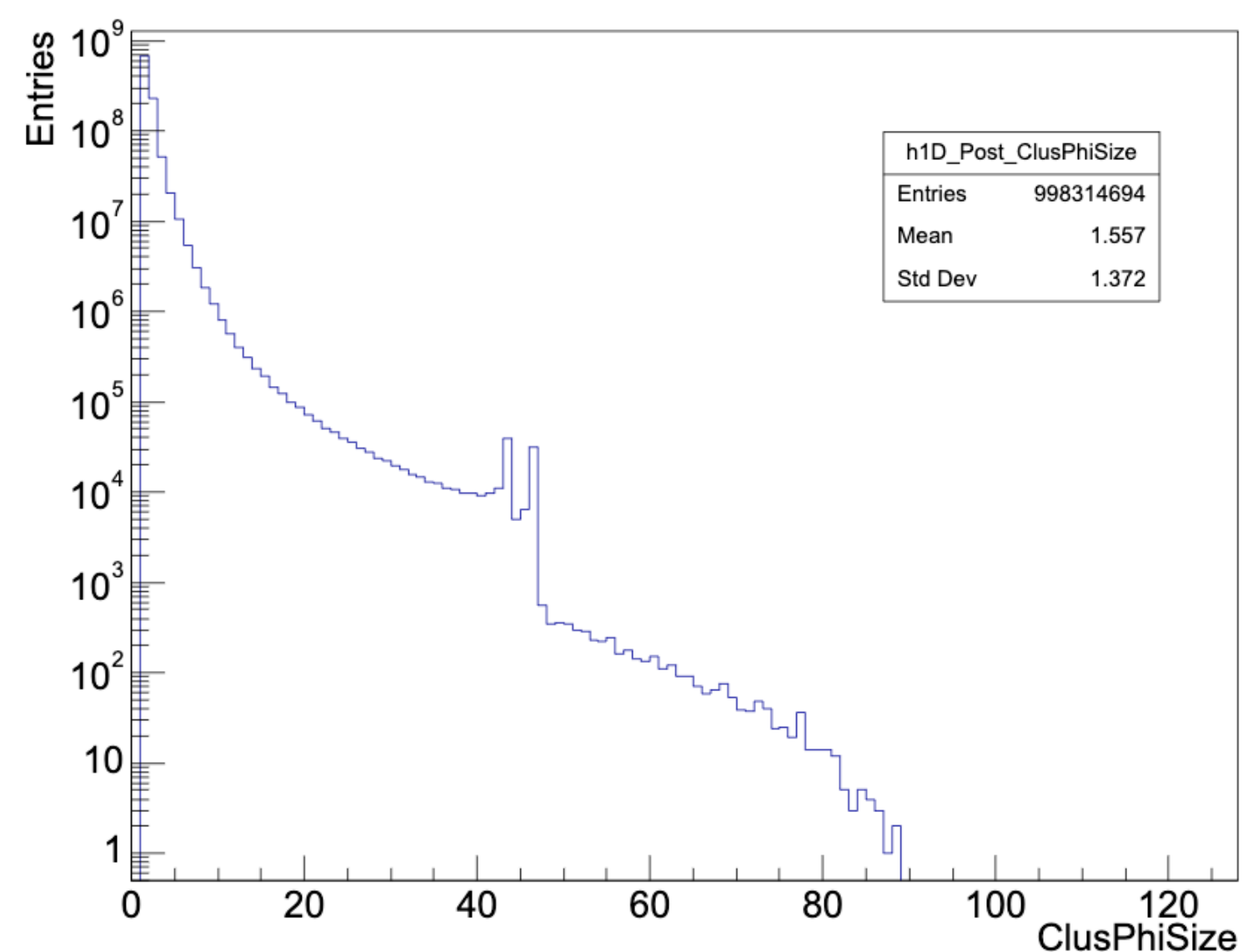
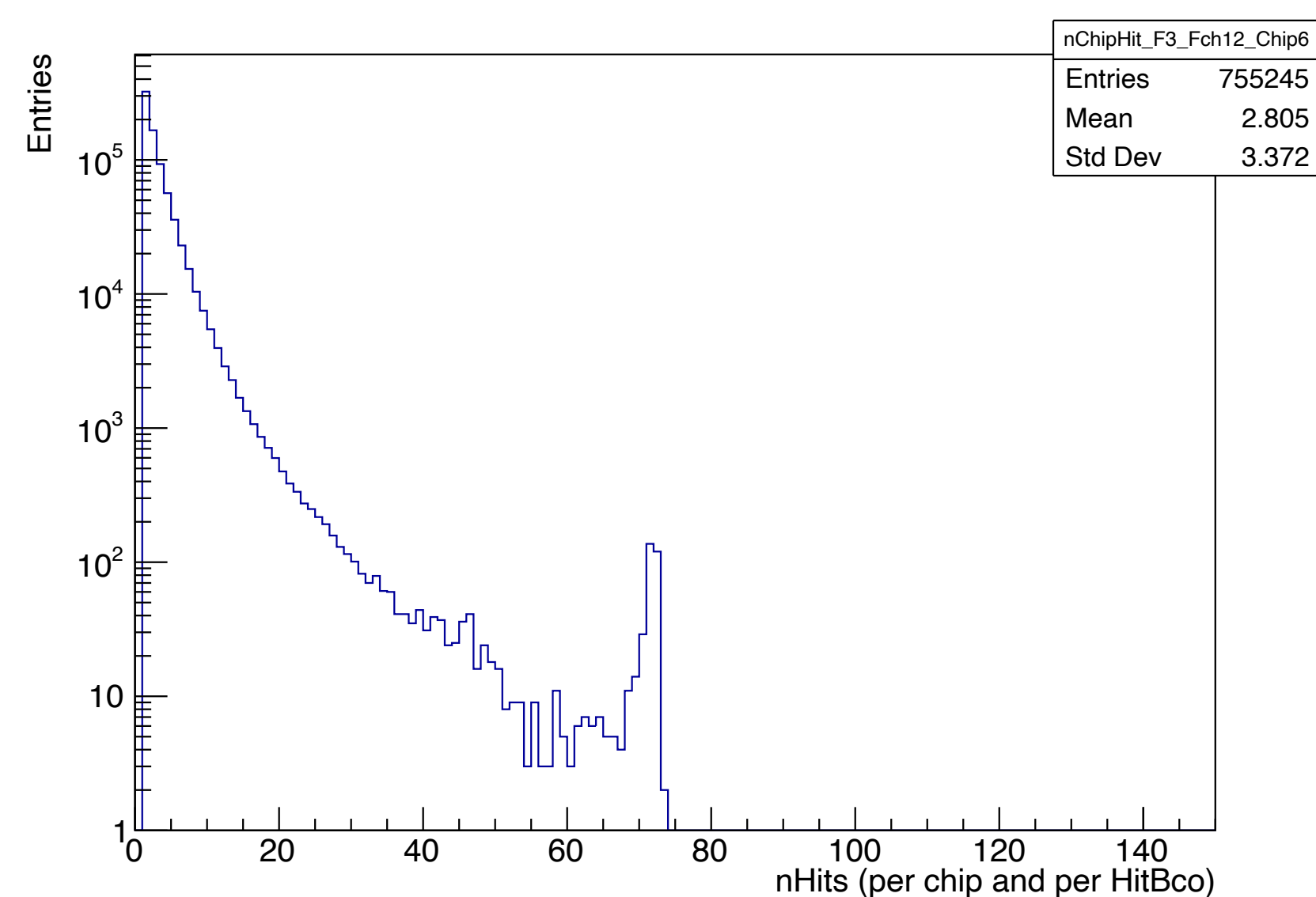


- https://github.com/sPHENIX-Collaboration/coresoftware/compare/master...ChengWeiShih:sPH_coresoftware:master
- The code has been tested running OK (with ana.520)

Back up

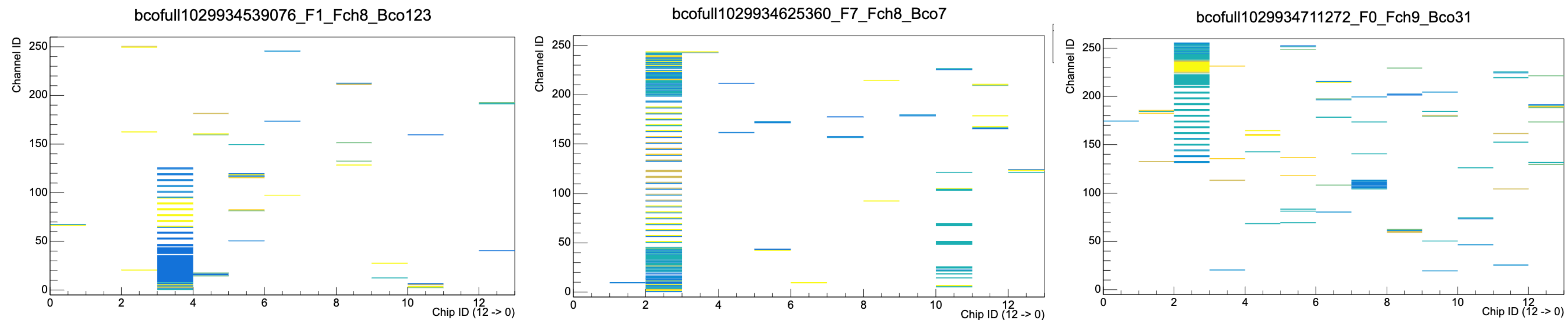
Recap - chip saturation

- A cutoff in chip-hit distribution and a two-spike structure in cluster-phi-size distribution were observed
 - Indication of chip saturation & the two issues are correlated
 - It's a chip-level problem (cannot be resolved by extending the open_time, studied by Shan-Yu)
 - The chip-overflow-flag cannot identify the chip saturation
 - The meaning of the chip-/ROC-overflow-flag is still unclear
 - **It seems NOT a major issue**
 - Ratio of saturated chips < 1% & it's independent of the event activity



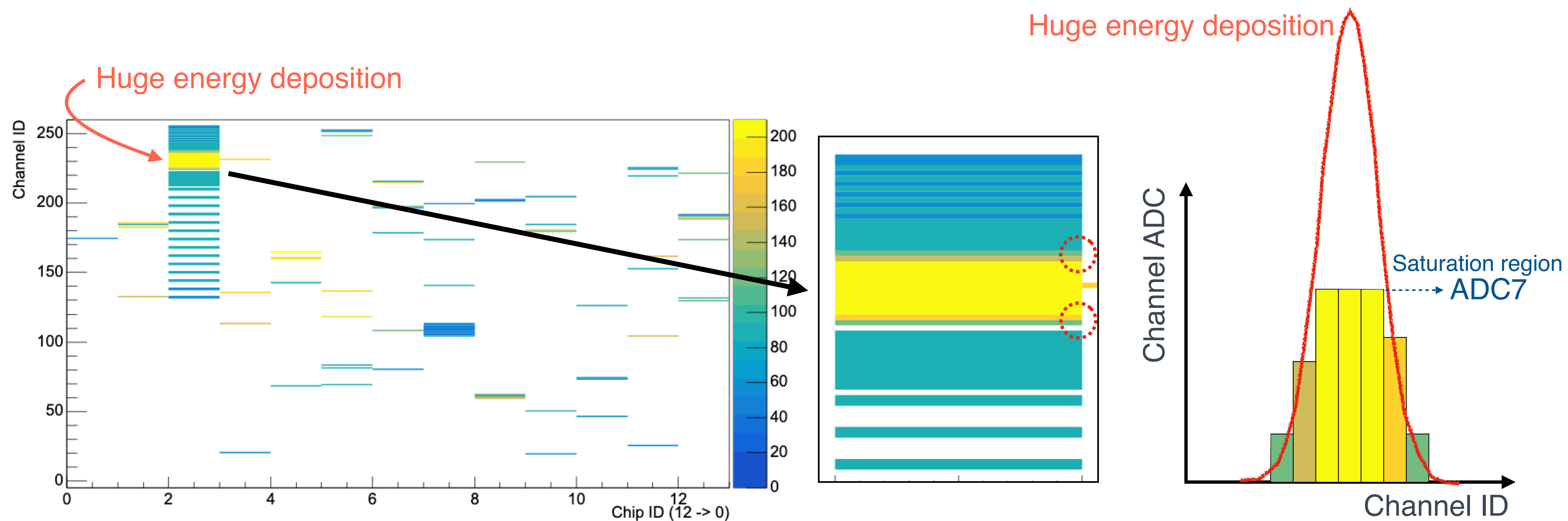
Recap - chip saturation

- Remaining questions
 - ~~Would the chip saturation issue be relevant to the previous chip activity?~~
 - Correlation with multiplicity in p+p collisions?
 - Are hits/clusters in the saturated chips real or fake?
 - Any treatments for those hits and clusters?
 - If so, how can those chips be identified?
 - Their patterns could be very diverse (even if disabling the hit adc info.)



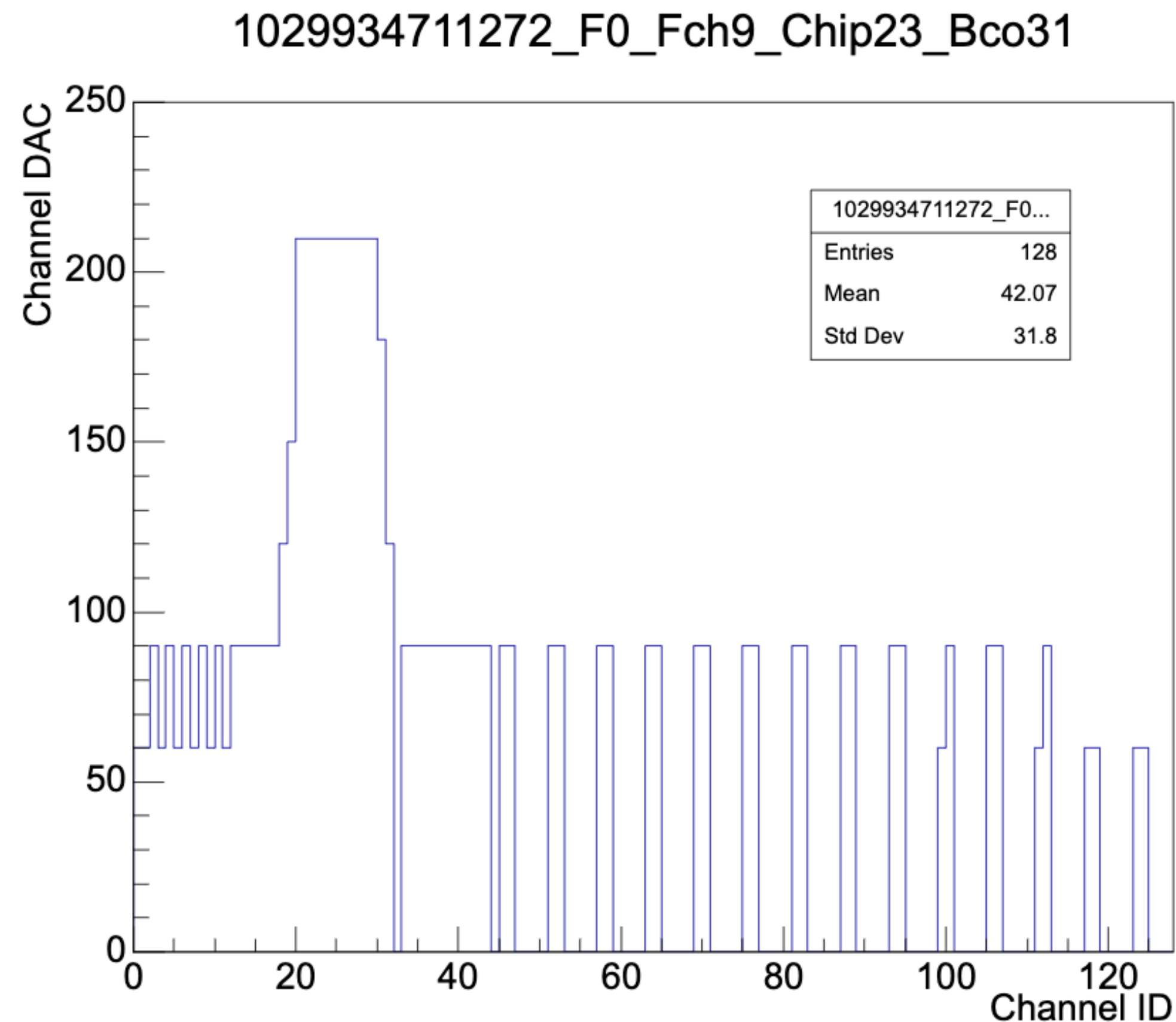
Can those hits be real?

- Pro side:
 - Huge energy deposition leads to a charge cloud large enough to cover the full chip
 - Not a step function when transitioning to the plateau region

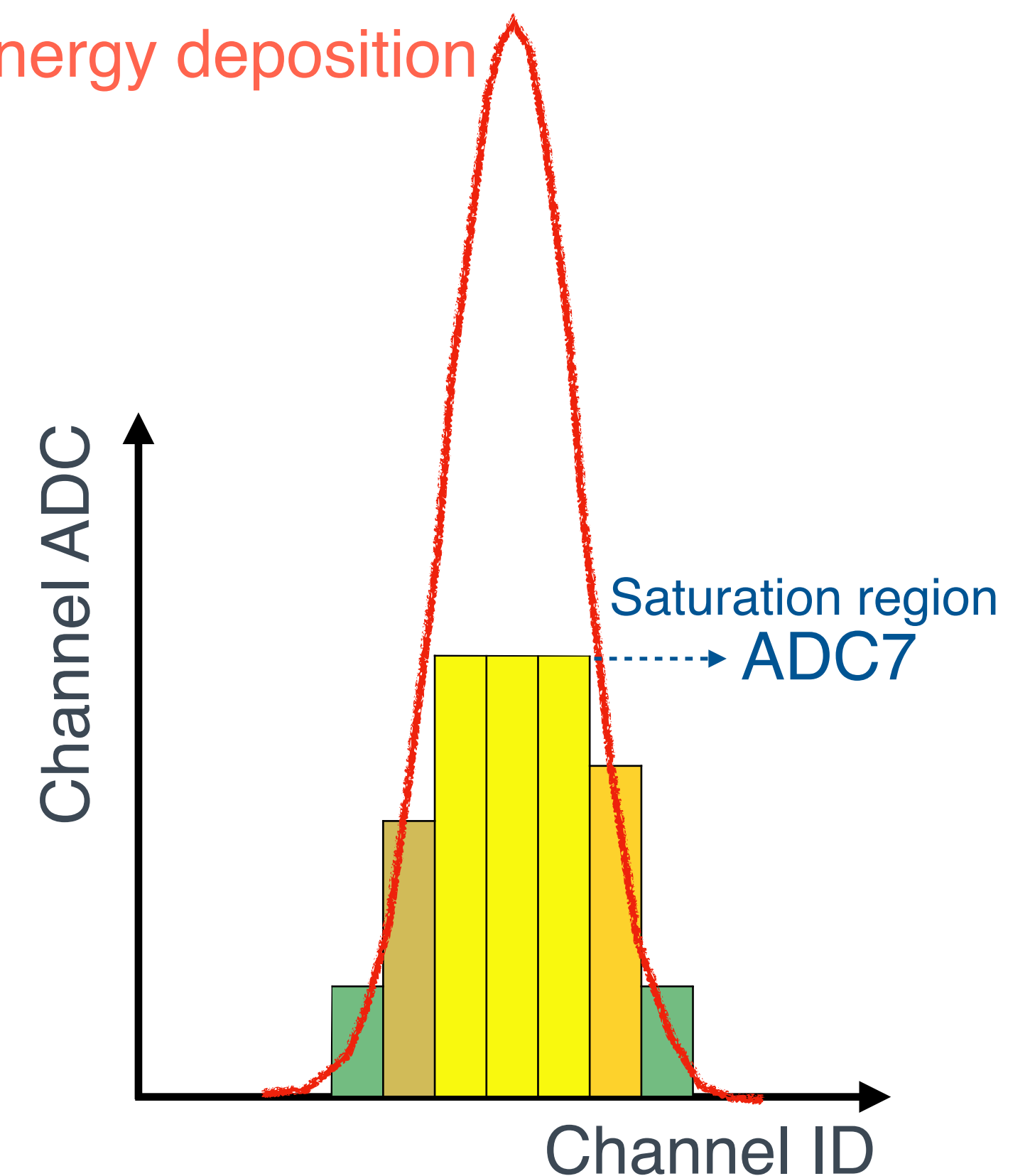


Can those hits be real?

- Pro side:
 - Huge energy deposition leads to a charge cloud large enough to cover the full chip
 - Not a step function when transitioning to the plateau region

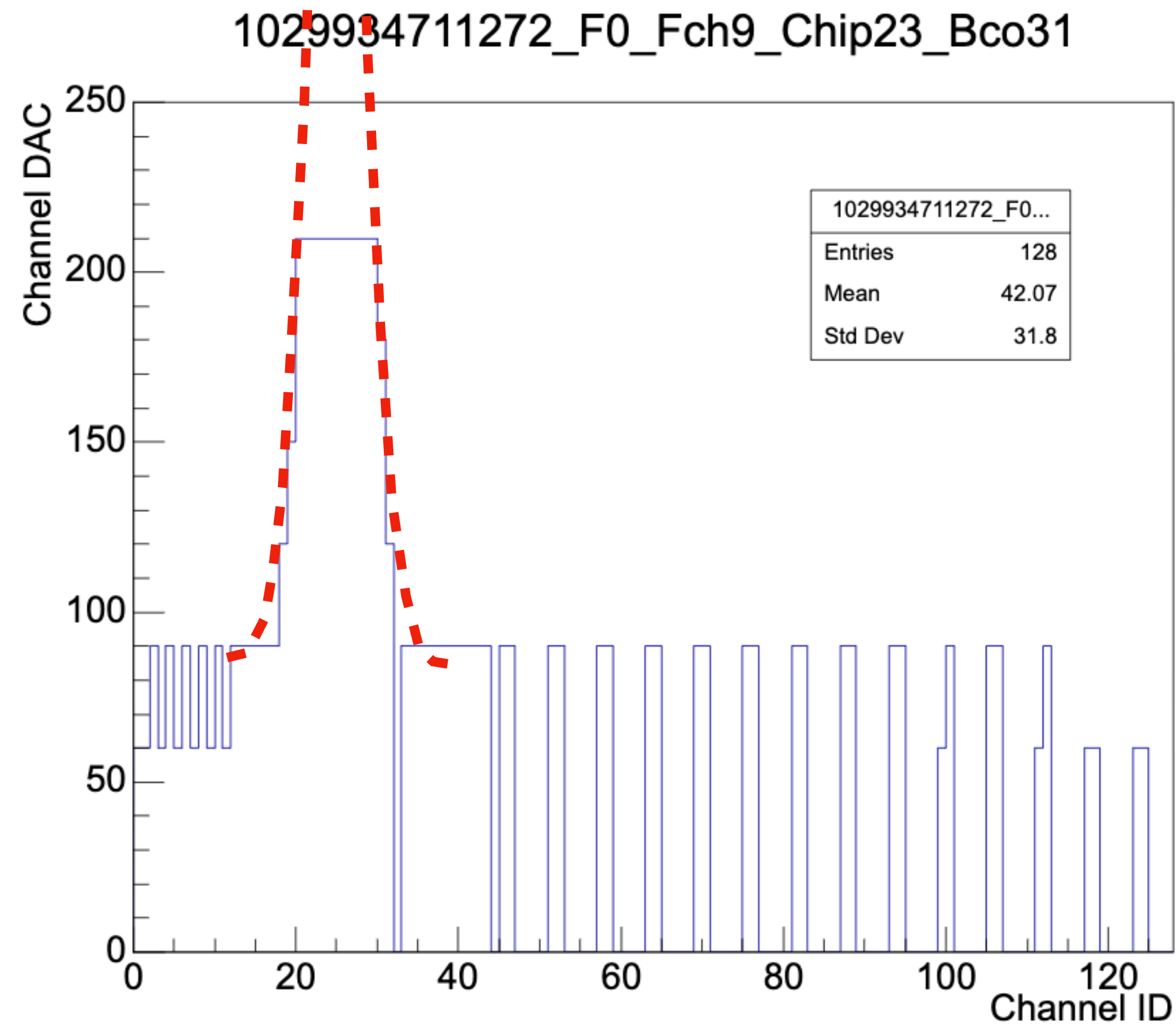


Huge energy deposition

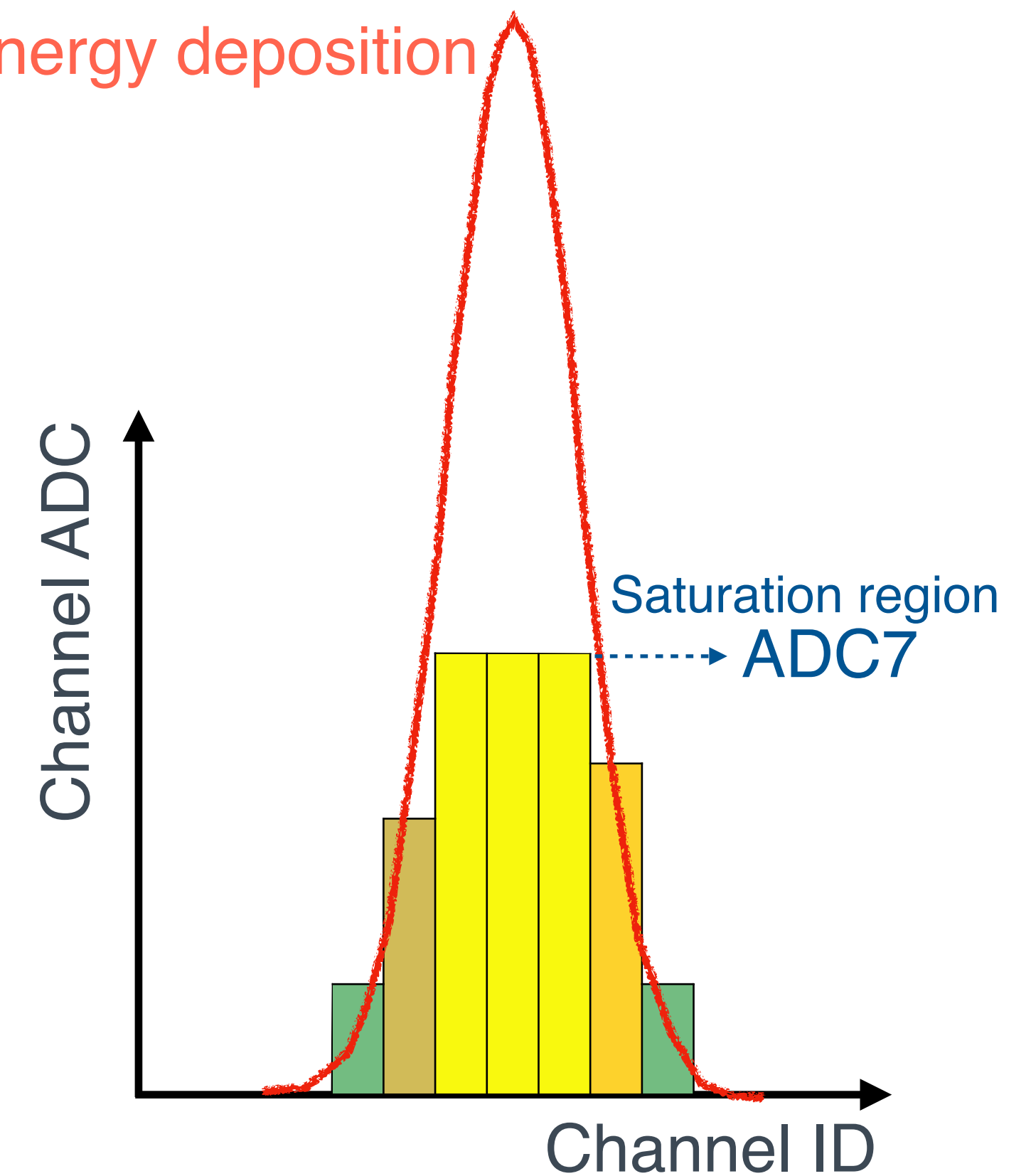


Can those hits be real?

- Pro side:
 - Huge energy deposition leads to a charge cloud large enough to cover the full chip
 - Not a step function when transitioning to the plateau region

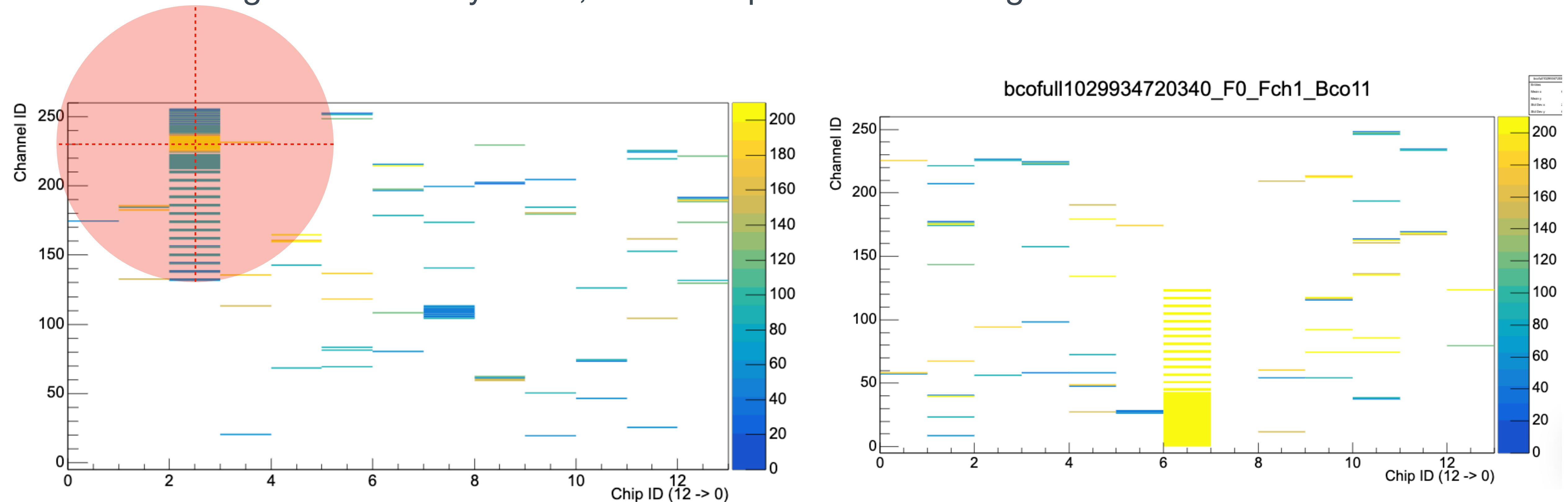


Huge energy deposition



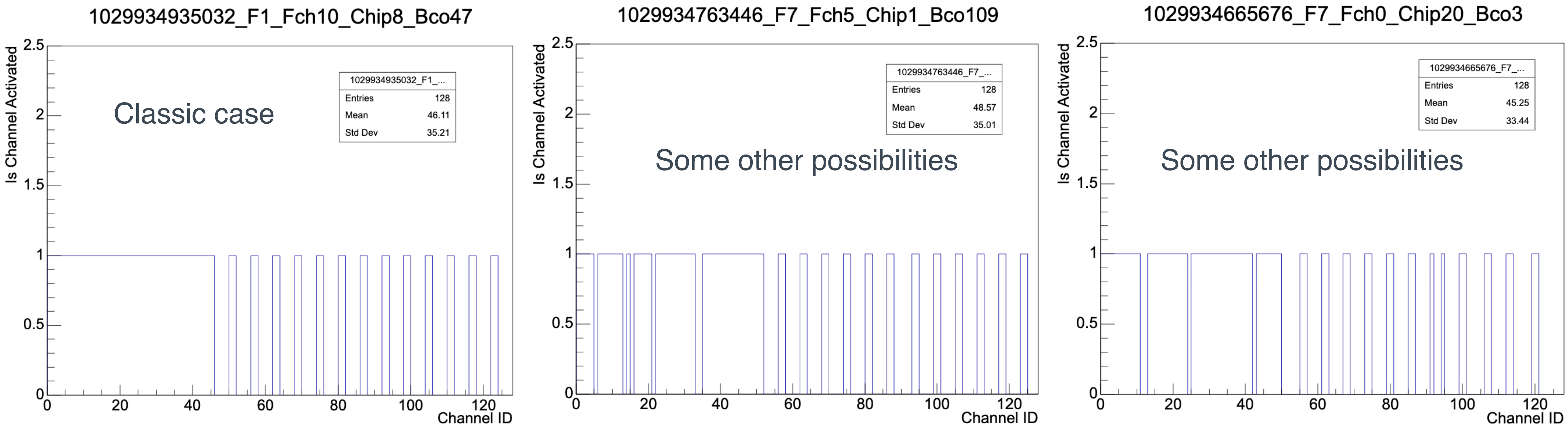
Can those hits be real?

- Con side
 - If the charge-cloud theory holds, how come it (almost) never has charge shared to the adjacent columns?
 - If the charge-cloud theory holds, how come it (almost) extends to the end of one chip (channel 127)?
 - If the charge-cloud theory holds, how to explain the following case?



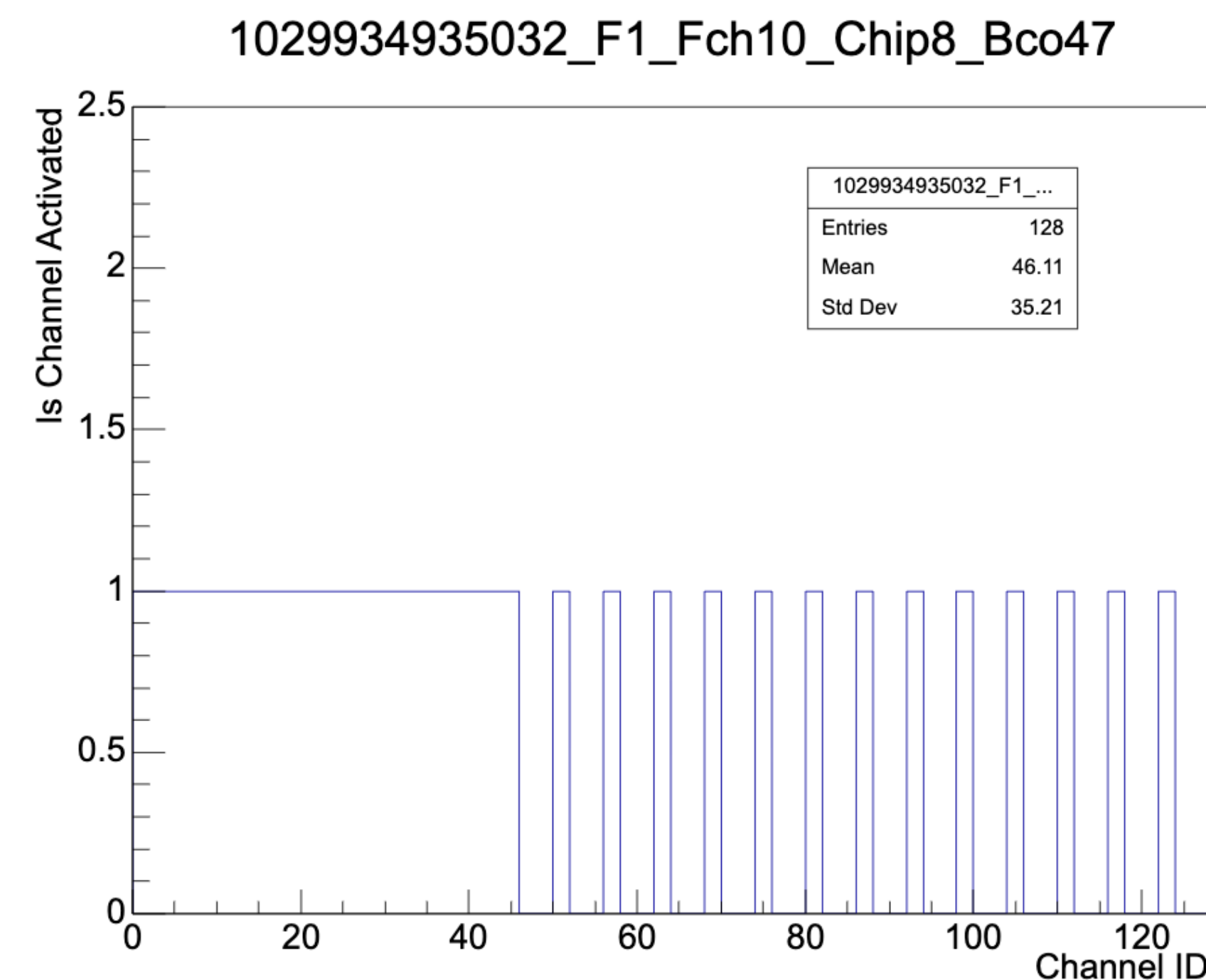
Attempt to identify the saturated chips

- The chip hit maps of saturated chips are so diverse
 - > 80k different patterns in 420k saturated chips (identified based on the ChipHit cut), even if disabling the hit ADC information (only checking whether the channel is activated or not)

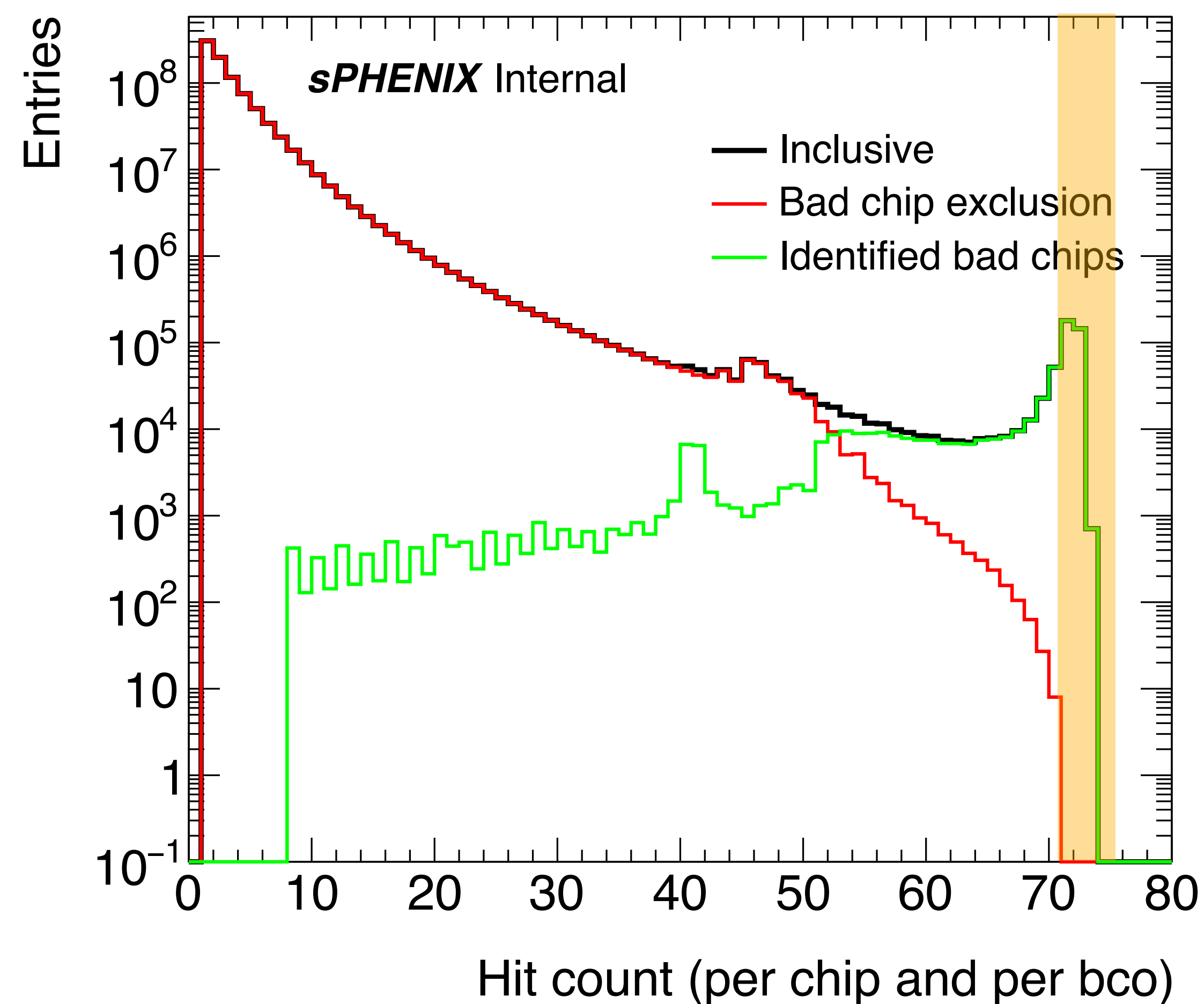
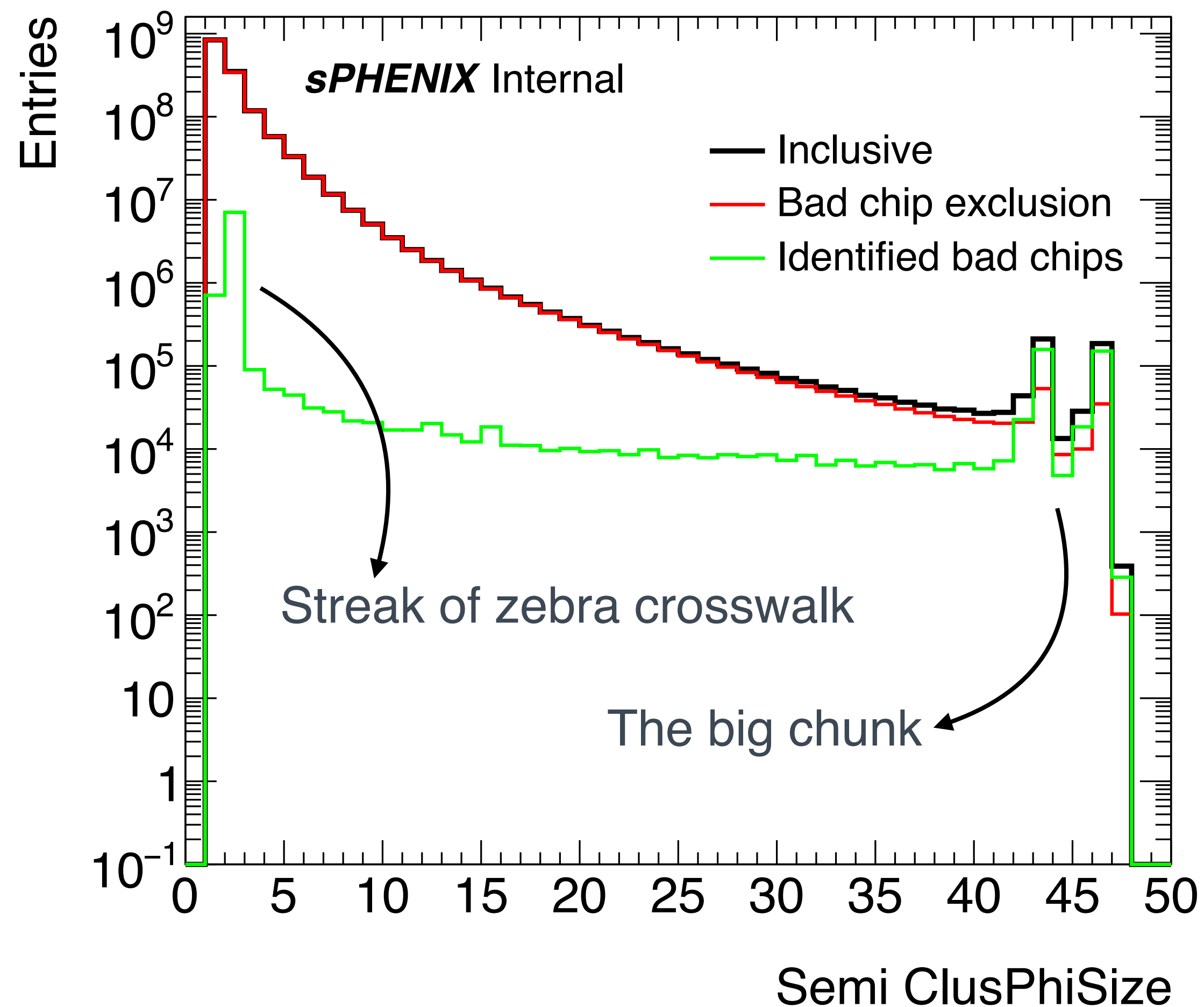


Attempt to identify the saturated chips

- Approach: more like focusing on the zebra-crosswalk region
 - (Largest cluster size > 25 && number of sized-2 clusters ≥ 5) or
 - One of the following patterns is found in the chip hit map
 - Pattern1: 11000011000011000011 (span b/w streaks: 4 channels)
 - Pattern2: 110000011000011000011 (allow one span b/w streaks to be 5 channels)
 - Pattern3: 110000110000011000011
 - Pattern4: 110000110000110000011
- Pattern4: 110000000110000110000011



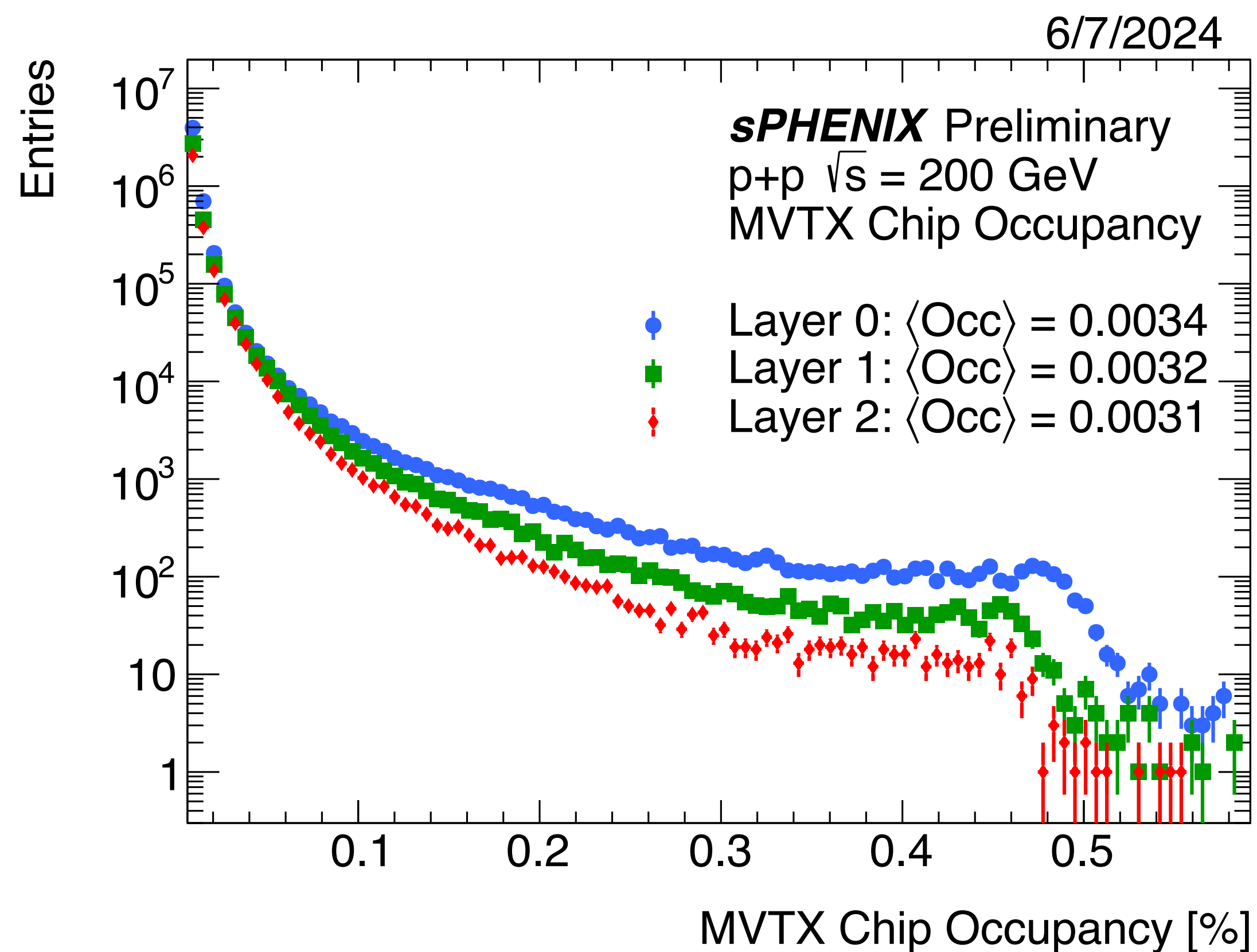
Attempt to identify the saturated chips



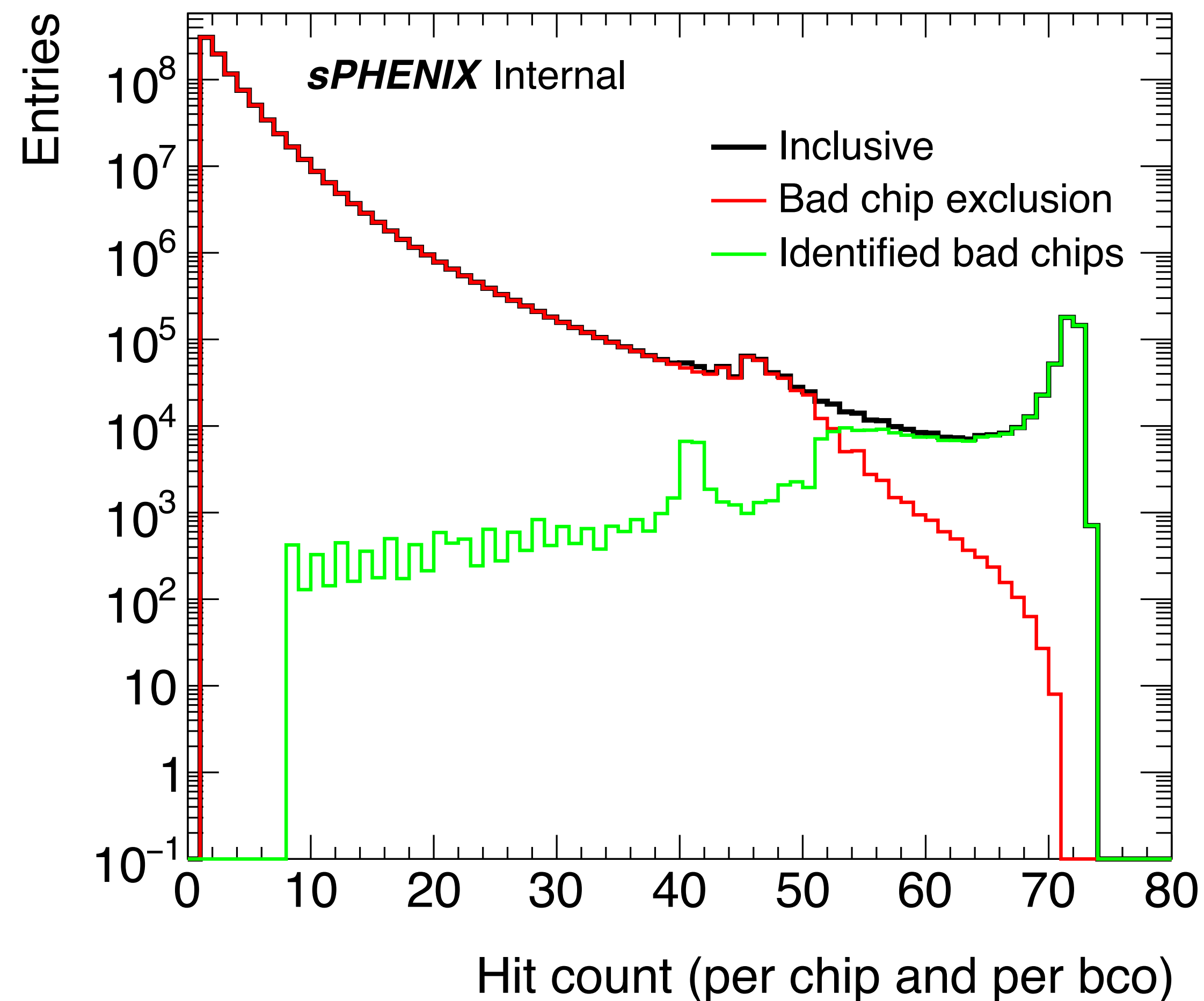
Two spikes still visible → not all the saturated chips can be identified using the current algorithm
But the sizes of the two spikes are reduced by ~ 80% already

Attempt to identify the saturated chips

MVTX chip occupancy in **p+p** collisions

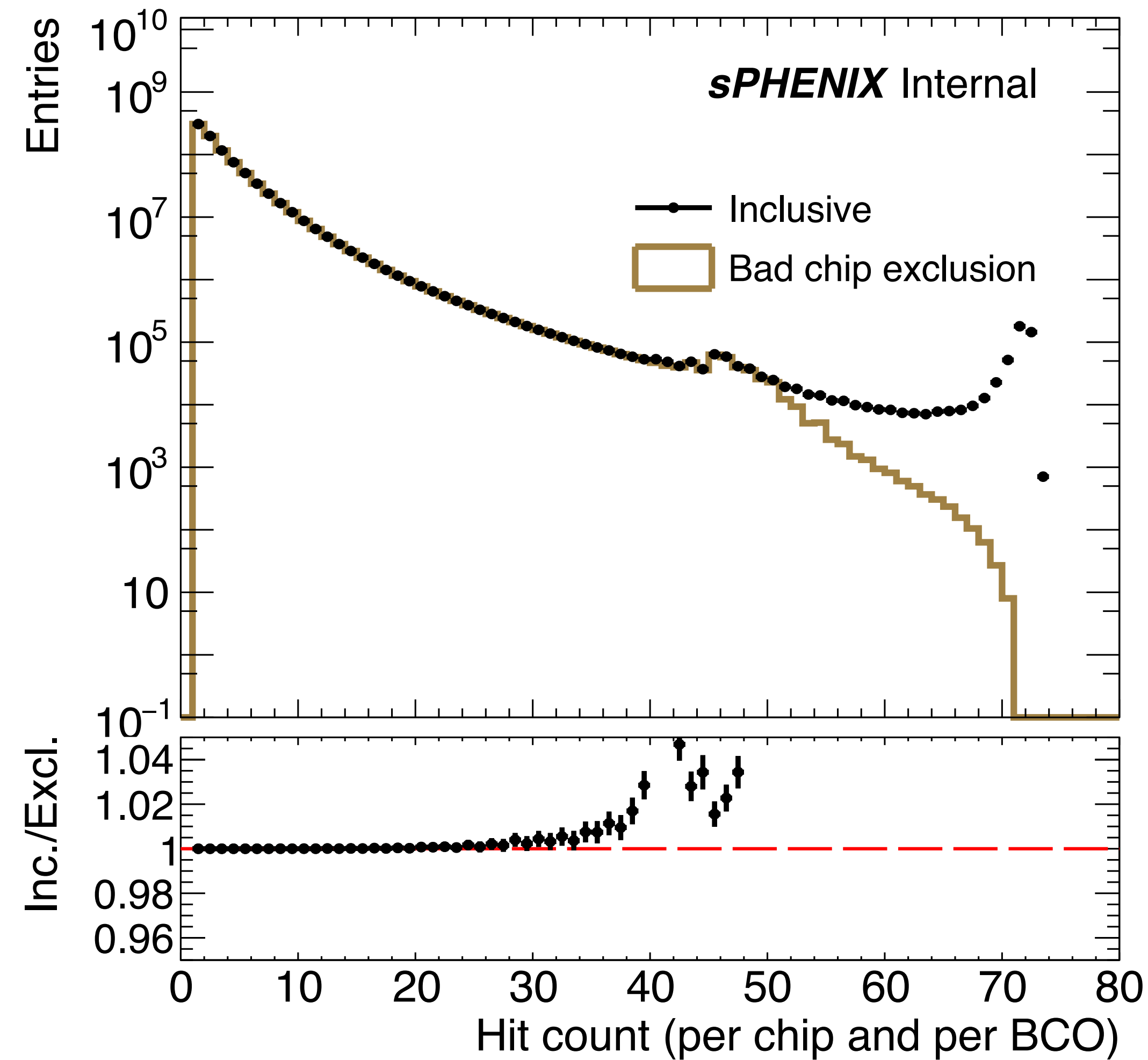
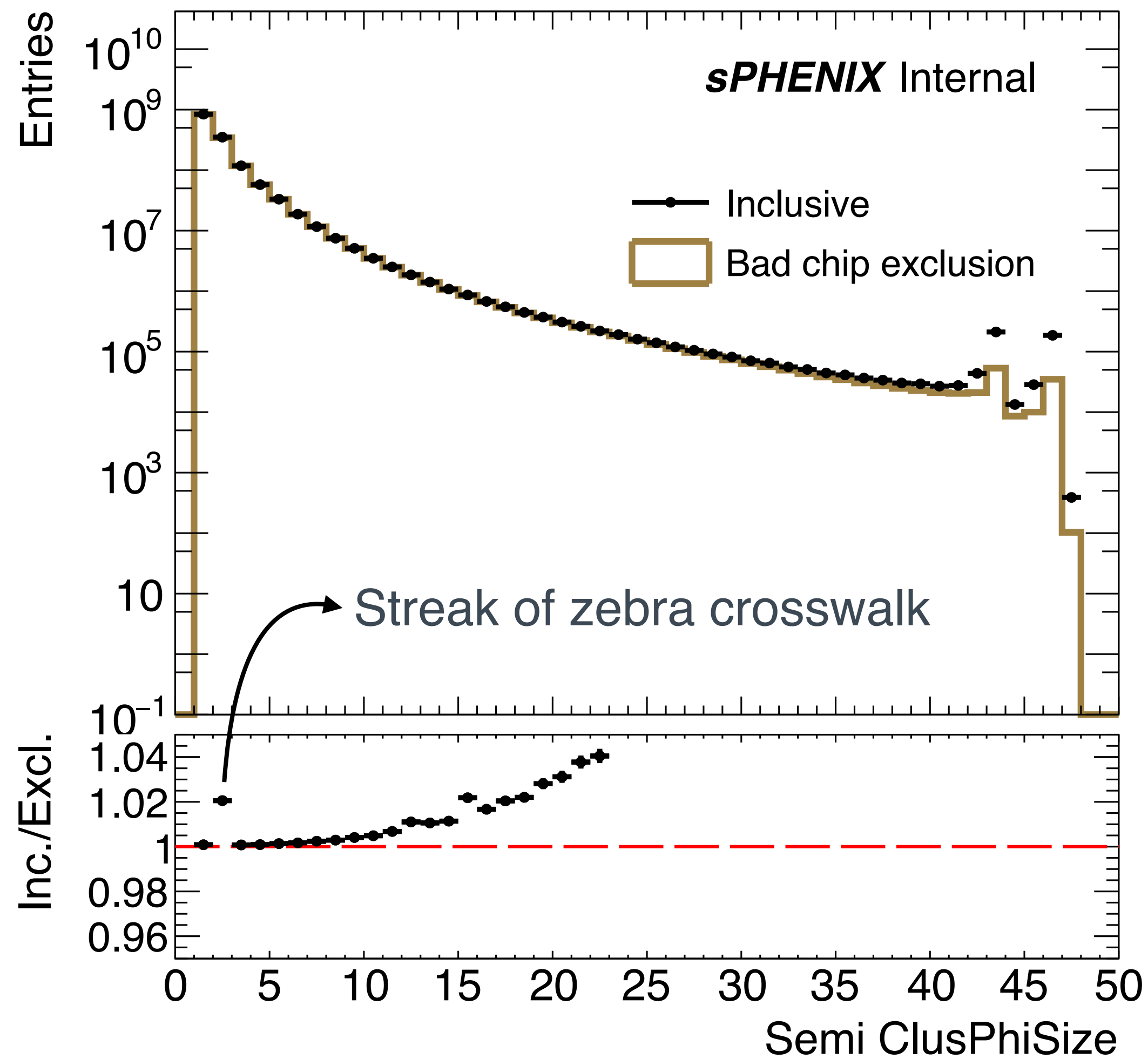


INTT chip occupancy in **Au+Au** collisions



Similar shape, perhaps a good sign?

Attempt to identify the saturated chips



With the semi clustering algorithm (grouping hits chip-by-chip): clusters in the bad (saturated) chips ($8.72013e+06$) / all clusters ($1.45917e+09$) = 0.597% → below 1%, not a major issue