

Correlations of #INTT hit of triggered/extended/streaming data

Genki Nukazuka (RIKEN)

Introduction: Confirmation of healthiness of our streaming data

We need to confirm that our streaming data is good. Checking correlations b/w parameters is one of the steps to prove it.

INTT tracklet is useful for detector performance study

We have to make sure that INTT works properly.

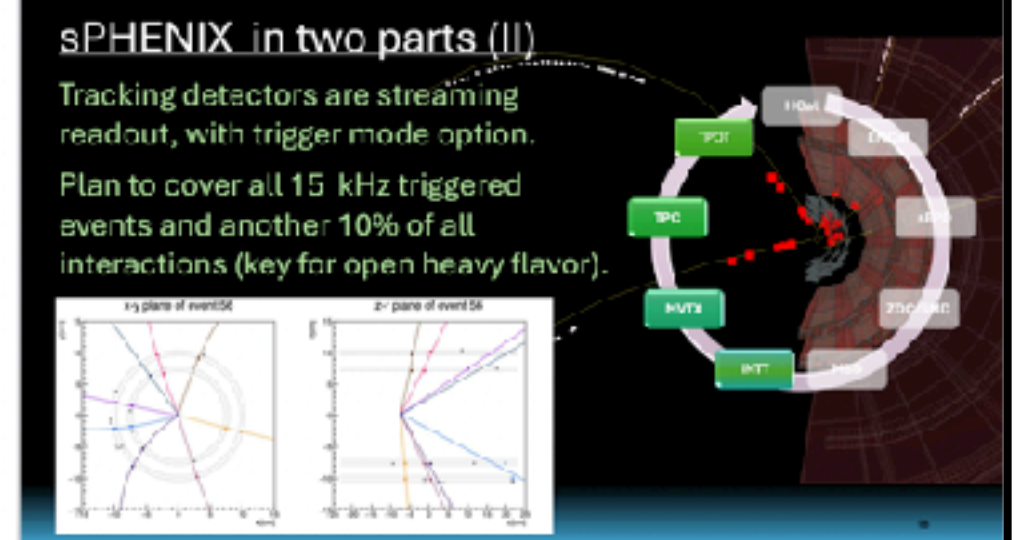
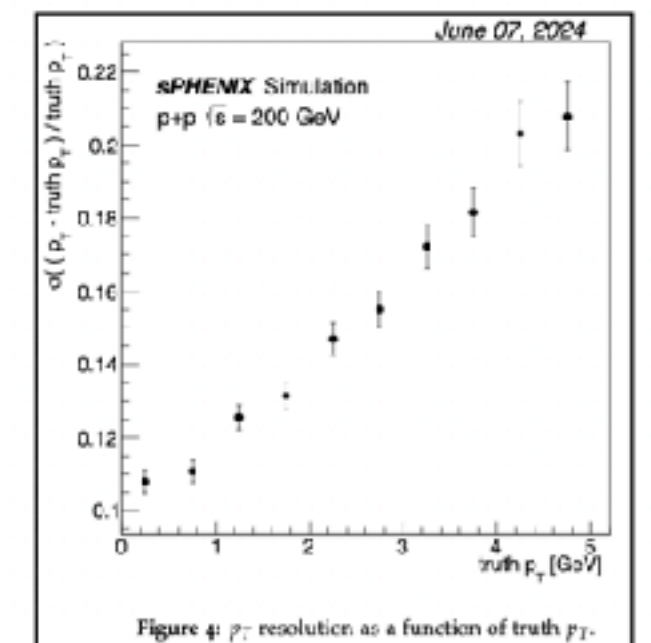
- **Looking at distributions by eye** is the easiest way. It's fine to see the detector phenomenologically but not so scientific.
- **Parameter correlations** are easy way to see, but it doesn't guarantee the normality hit by hit.
- Checking dedicated parameters (detection efficiency, MIP, etc.) is a very reliable way to see healthiness of the detector.
- Tracklet finding is the ultimate tool for tracking detectors to confirm whether the detector takes real hits.

The great demonstration has already been performed by Hinako, and the study got some preliminary plots.

We have to maintain/update this activity to check the detector performance. I inherited Hinako's codes to make tracklets using the streaming data.

Note: We have to rely on it at least for local runs as other detectors were not run. Local runs are the ones we are interested in.

[Jamie's slide](#) shown at the RHIC/AGS meeting →



Analysis

DST files were used with only INTT data. GL1 was not used.

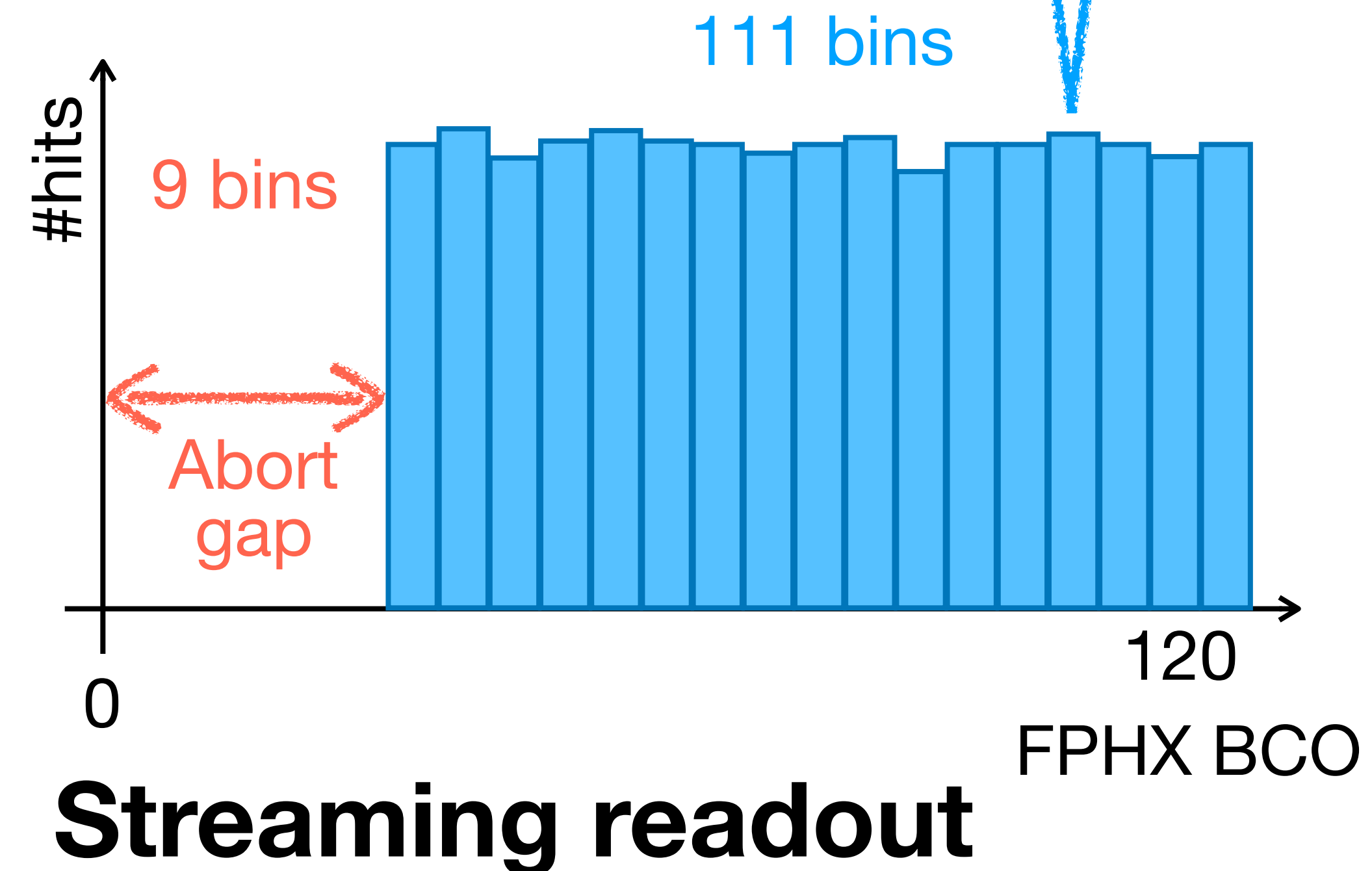
Steps:

1. Decode evt file \rightarrow $DST_{(1)}$ with InttRawHit
2. Analyze $DST_{(1)}$ \rightarrow hot channel CDB
3. Make $DST_{(2)}$ containing TrkrHit using $DST_{(1)}$ and the hot channel CDB \rightarrow $DST_{(2)}$
4. Analyze $DST_{(2)}$
 - i. Get TrkrHits
 - ii. Get InttRawHits
 - iii. Find corresponding InttRawHit to a TrkrHit
 - iv. Assign FPHX BCO of the InttRawHit to the TrkrHit
 - v. Divide TrkrHits to BCO group
 - vi. Count TrkrHits

Hits in a bin come from a single collision. An event in streaming data is **started at the beginning of the abort gap and ended at the beginning of the next abort gap (1 RHIC turn)**.

\leftarrow It may be changed in the future.

FPHX BCO can be used to identify which bunch-crossing hit were from. Hits need to be counted for each bunch-crossing to check #hit correlation.



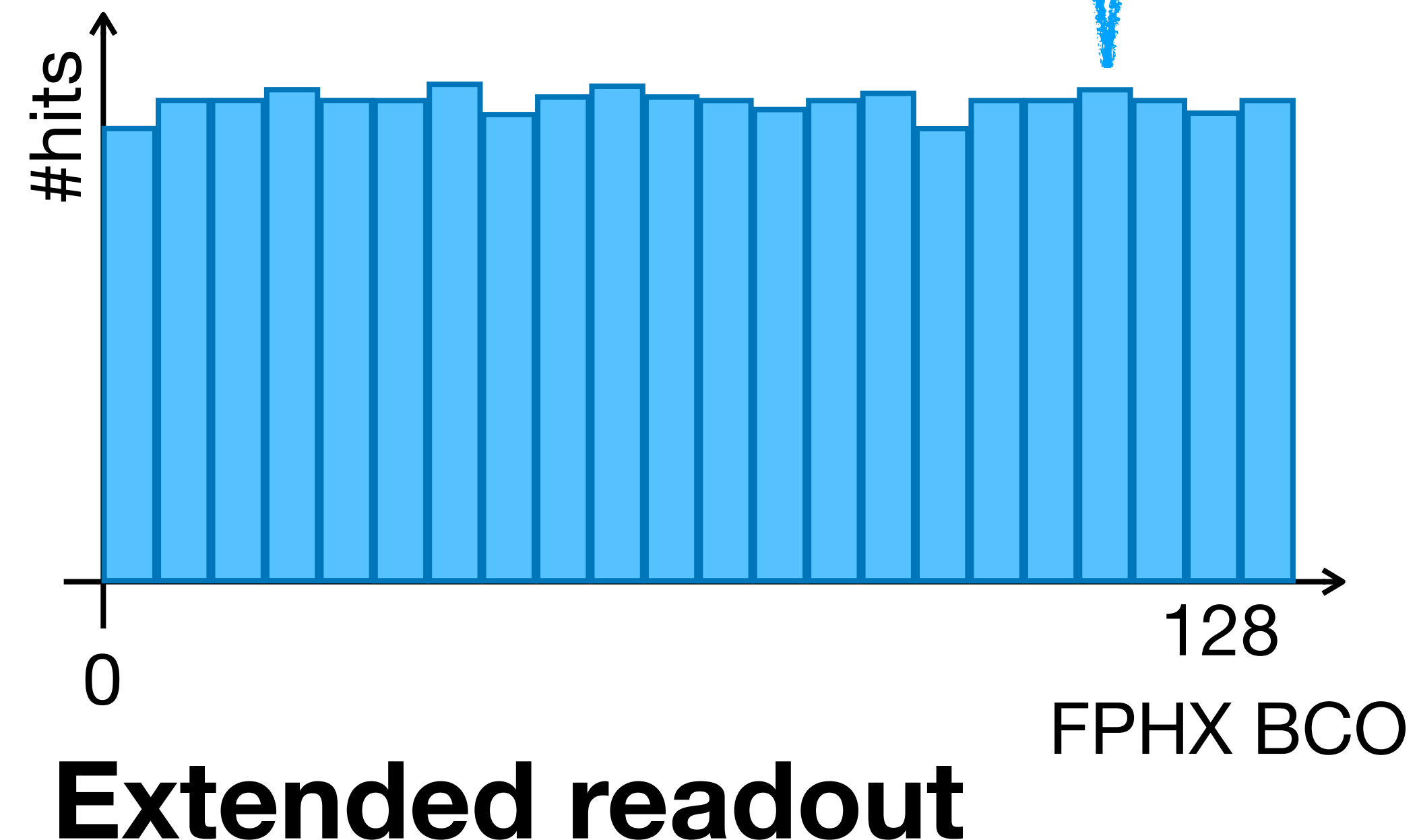
Analysis

DST files were used with only INTT data. GL1 was not used.

Steps:

1. Decode evt file \rightarrow $DST_{(1)}$ with InttRawHit
2. Analyze $DST_{(1)}$ \rightarrow hot channel CDB
3. Make $DST_{(2)}$ containing TrkrHit using $DST_{(1)}$ and the hot channel CDB \rightarrow $DST_{(2)}$
4. Analyze $DST_{(2)}$
 - i. Get TrkrHits
 - ii. Get InttRawHits
 - iii. Find corresponding InttRawHit to a TrkrHit
 - iv. Assign FPHX BCO of the InttRawHit to the TrkrHit
 - v. Divide TrkrHits to BCO group
 - vi. Count TrkrHits

BCO difference is normally used for triggered/extended data to identify triggered/extended hits. FPHX BCO distribution of them is just uniform. But **it can be used to divided them into bunch-crossing groups** though FPHX BCO doesn't tell which bunch. Grouping is necessary to see #hit correlation.



Analysis

```
1 int InttTrkrHitQA::process_event(PHCompositeNode *topNode)
2 {
3
4     auto status = this->GetNodes(topNode); // Get nodes to access data, It's written elsewhere
5
6     // get TrkrHitSet of INTT only, this is ConstRange = std::pair< ConstIterator, ConstIterator >
7     // It's written elsewhere
8     auto hits = GetHits( node_trkrhitset_map_->getHitSets() );
9
10    // FPHX BCO and event counter, It's written elsewhere
11    auto bco_event_counter_pair = this->GetBcoEventCounter(); // first: uint16_t (FPHX BCO), second: int (event counter)
12
13    int hit_num_barrel[ 2 ][ InttQa::kBco_max ] = { { 0 } }; // Hit counters, 0: inner, 1:outer
14    for( int i=0; i<hits.size(); i++ )
15        {
16            auto offline_hit = InttNameSpace::ToOffline( online_hit ); // useful structure
17            int barrel = ( offline_hit.layer <= 4 ? 0 : 1 ); // 0: inner barrel, 1: outer, maybe
18            hit_num_barrel[ barrel ][ bco_event_counter_pair[i].first ]++;
19        }
20
21    for( int bco=0; bco<InttQa::kBco_max; bco++ )
22        {
23            // skip empty hits
24            if( hit_num_barrel[ 0 ][ bco ] == 0 || hit_num_barrel[ 1 ][ bco ] == 0 )
25                continue;
26
27            hist_barrel_correlation_->Fill( hit_num_barrel[ 0 ][ bco ], hit_num_barrel[ 1 ][ bco ] );
28        }
29
30    return Fun4AllReturnCodes::EVENT_OK;
31 }
```

It's not the exact codes I used but fine to give a hint to students.

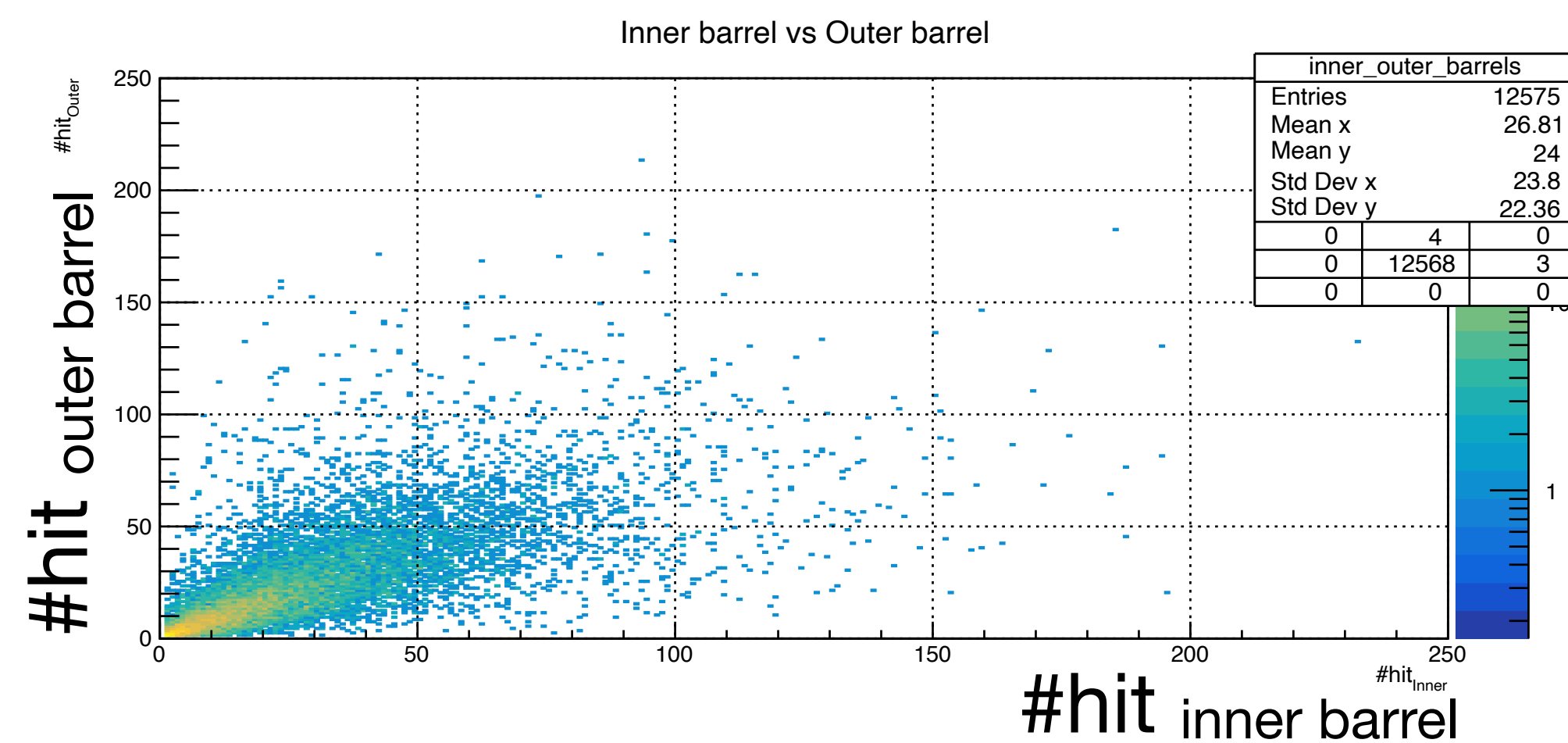
Correlation b/w the inner and the outer barrels

Triggered hits

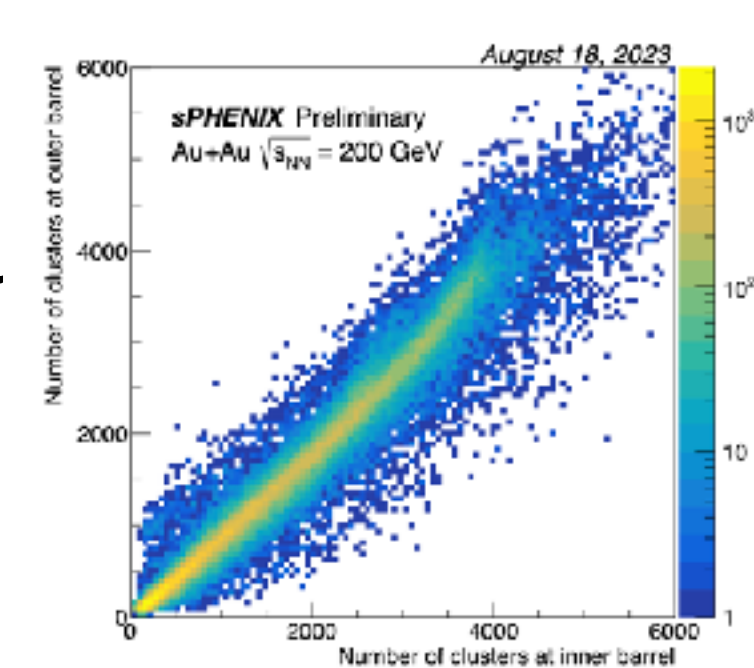
Run 41981 (May/10)

13k events analyzed

Hot channel rejection applied



AuAu
data



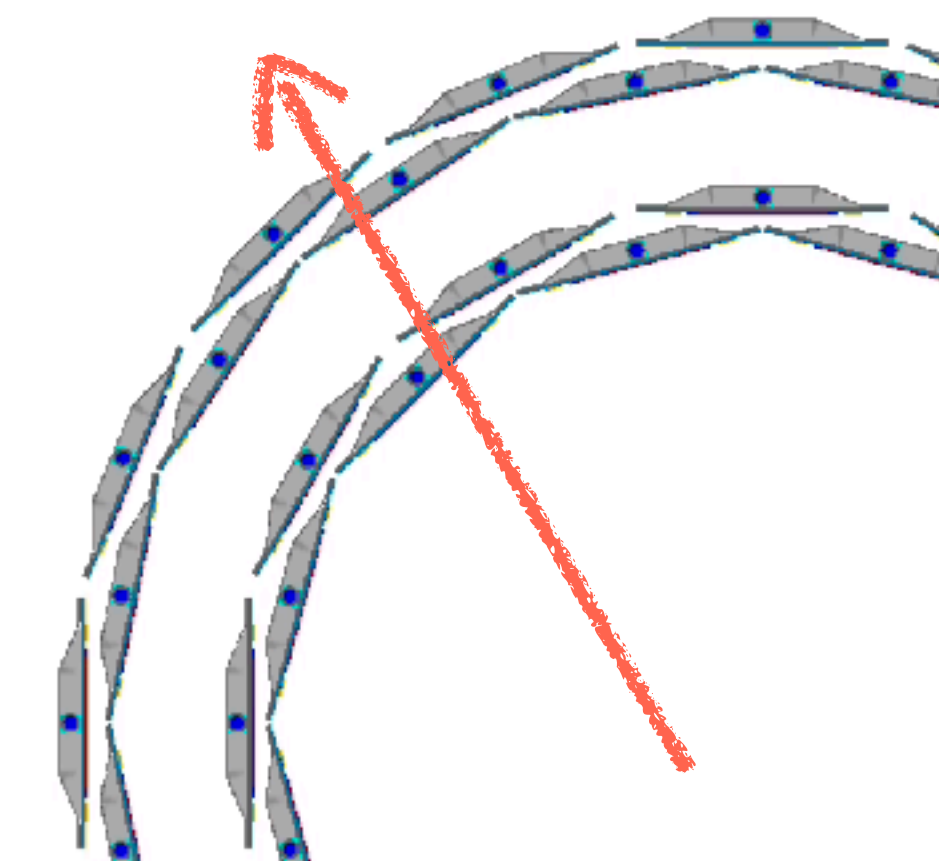
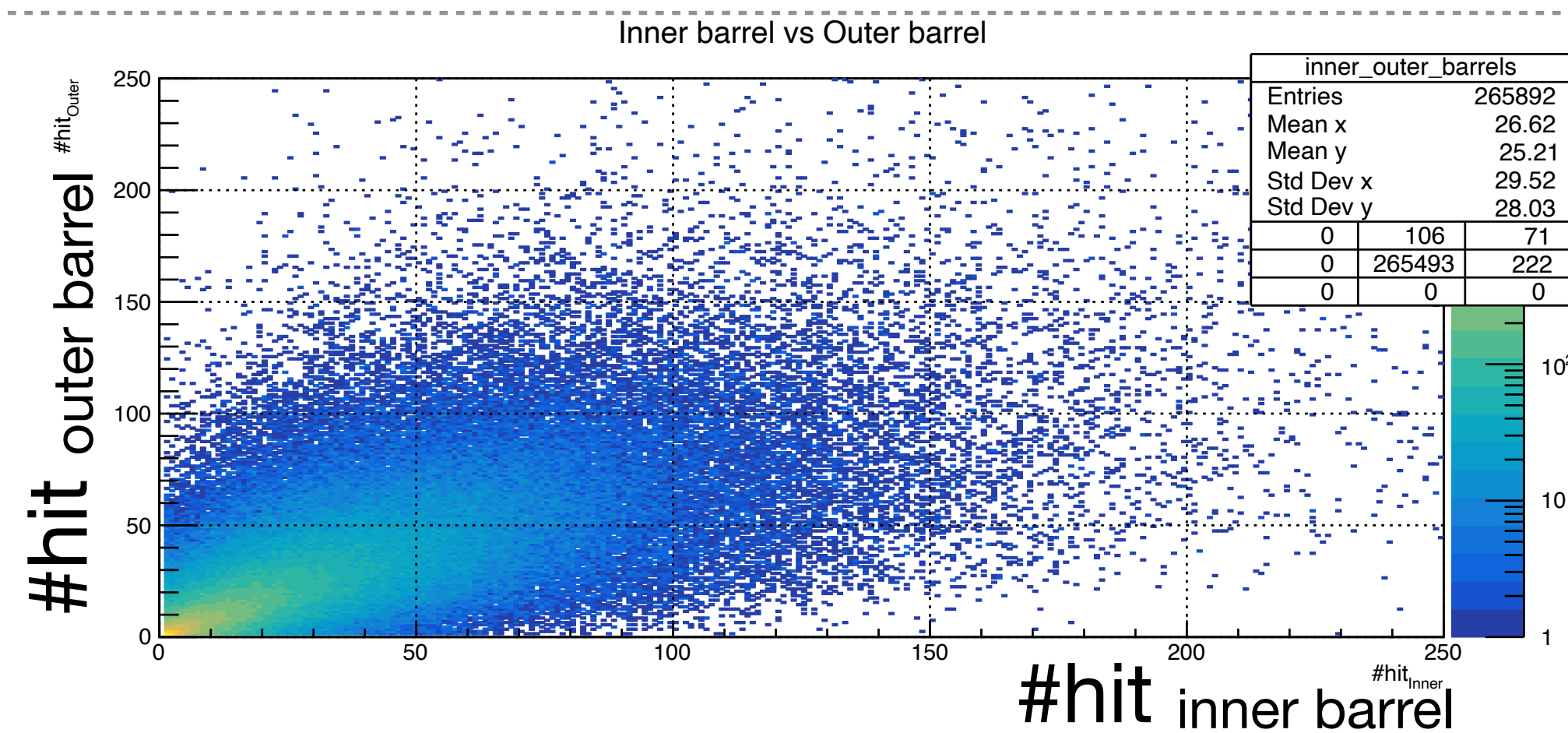
#hit on the inner and the outer should be correlated

Extended hits

Run 49737 (July/30)

10k events analyzed

Hot channel rejection applied



y x

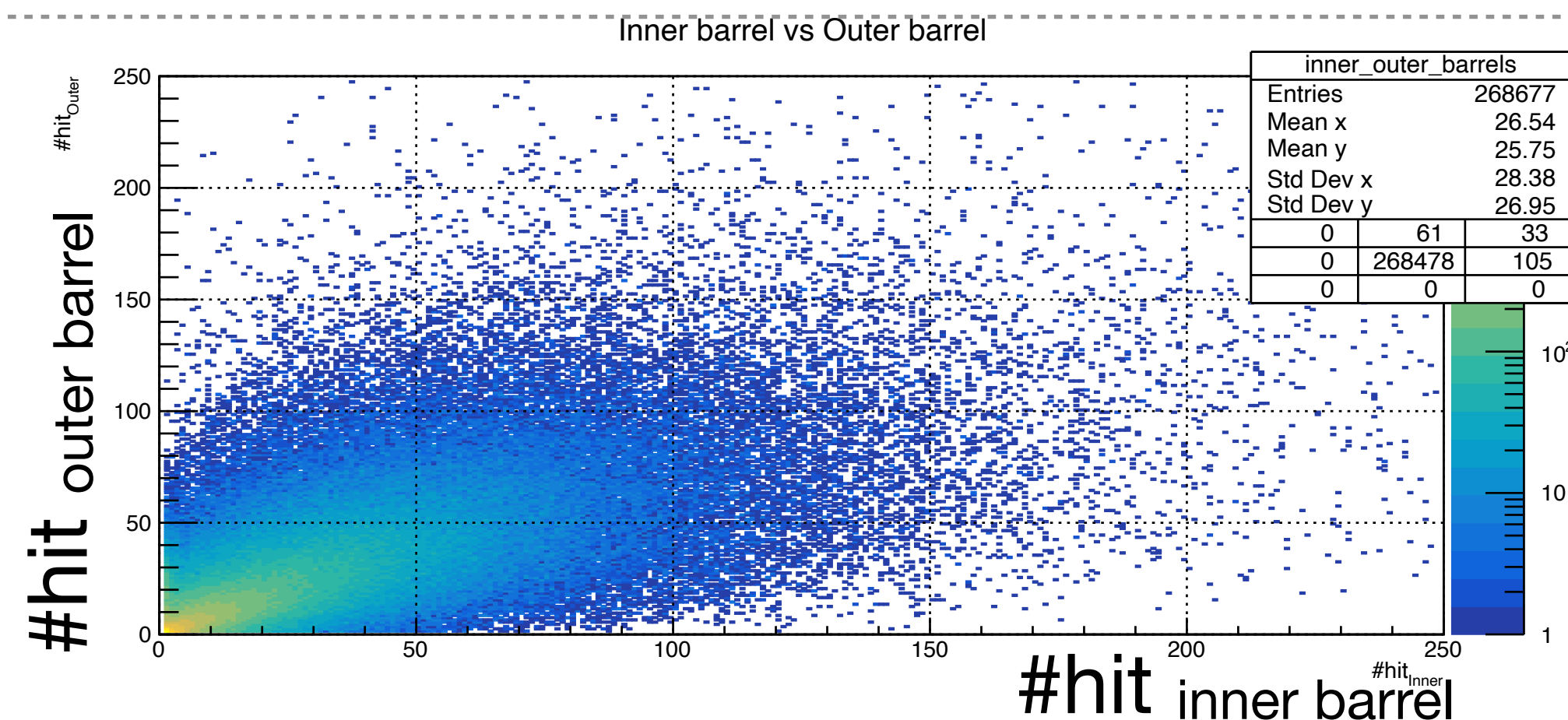
Streaming hits

Run 49743 (July/30)

10k events analyzed

Hot channel rejection applied

2 noisy half-ladders were masked at that time



Yes! We can see correlations in all cases.

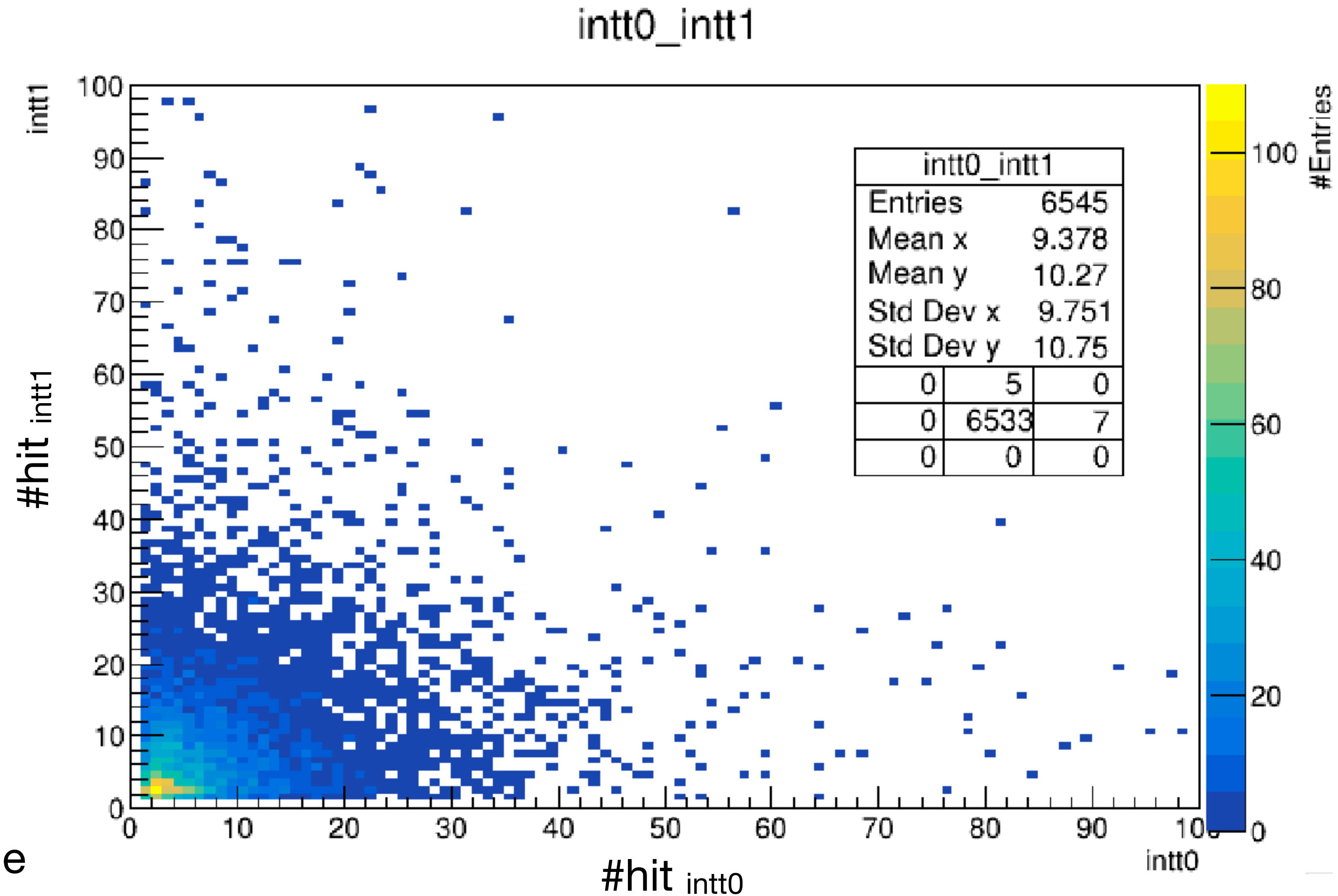
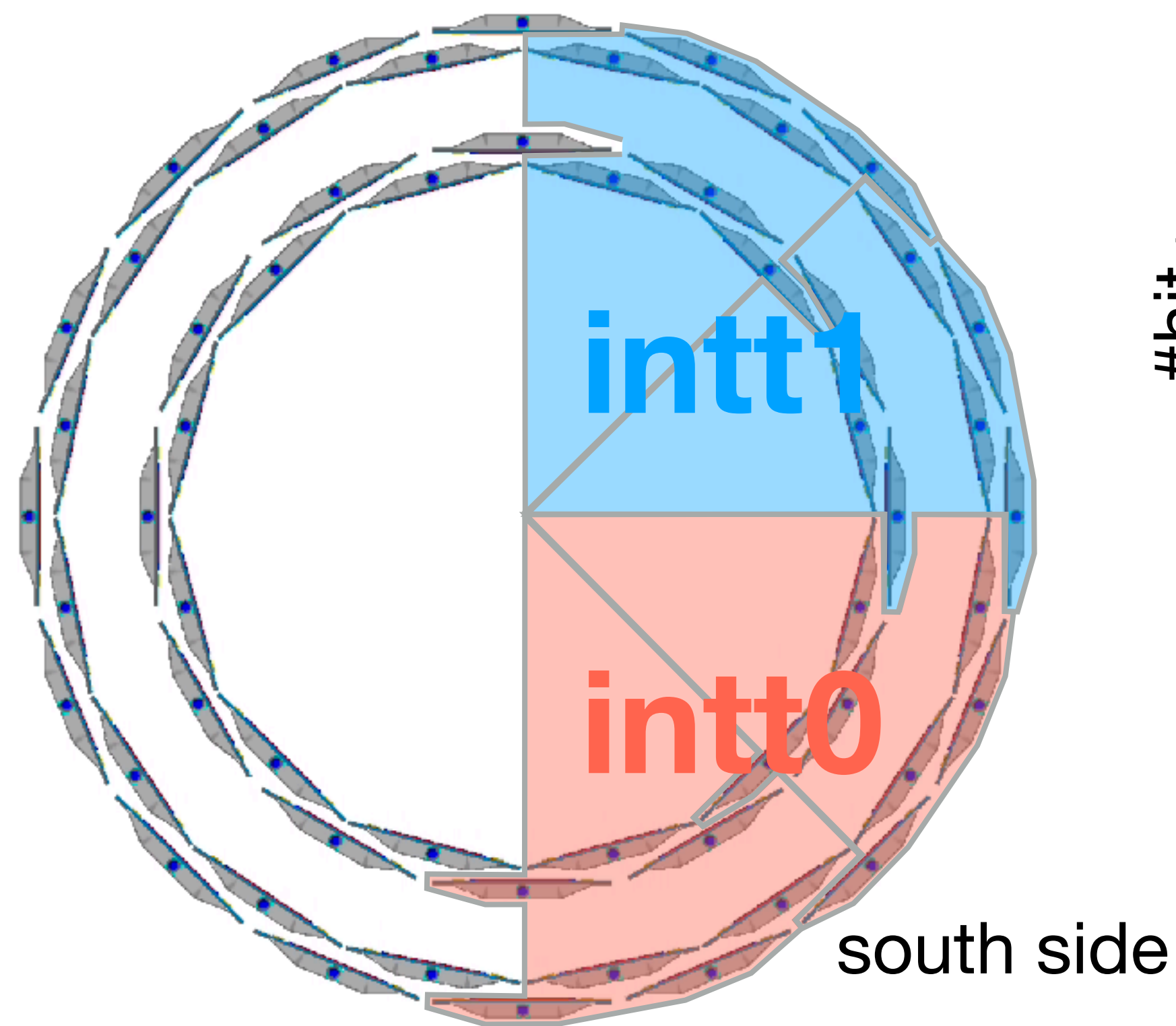
Correlation b/w FELIXes

Triggered hits

Run 41981 (May/10)

13k events analyzed

Hot channel rejection applied



No strong correlation can be seen b/w FELIXes. Probably, multiplicity is not high enough to make hits on INTT uniformly.

Correlation b/w FELIXes

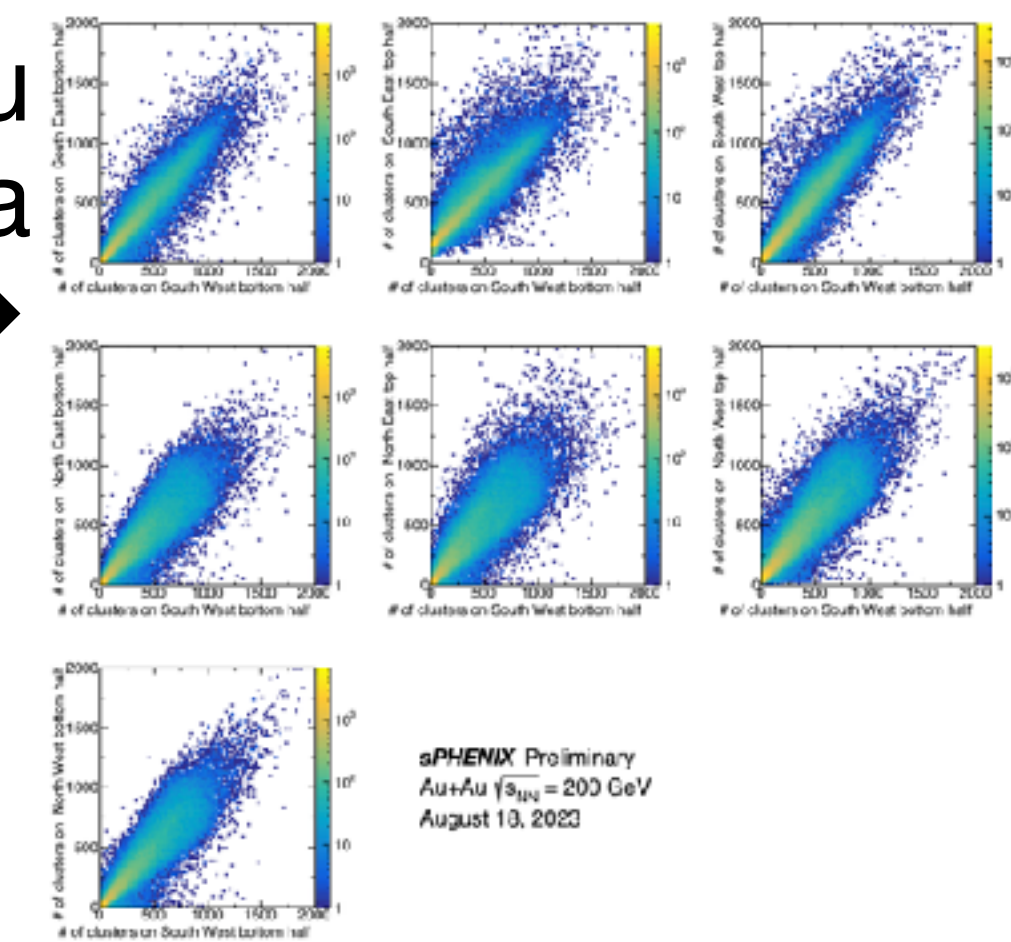
Triggered hits

Run 41981 (May/10)

13k events analyzed

Hot channel rejection applied

AuAu
data



y-axis

intt1

intt2

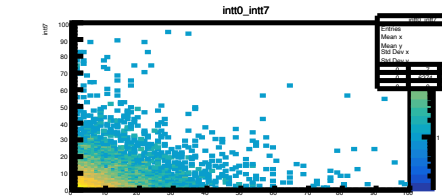
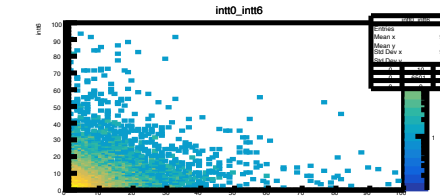
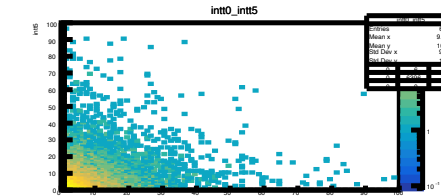
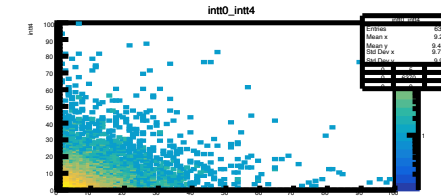
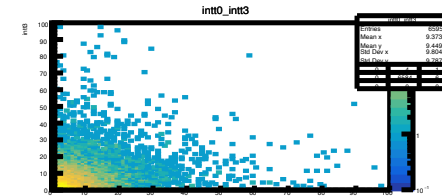
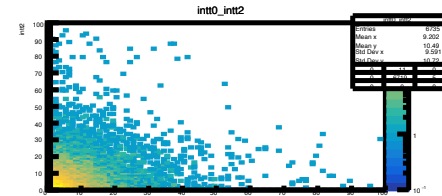
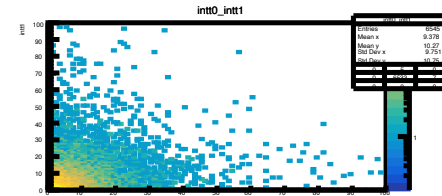
intt3

intt4

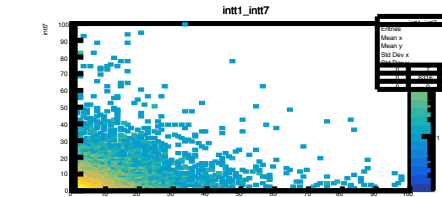
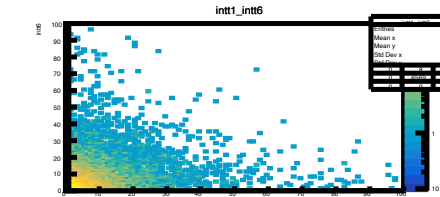
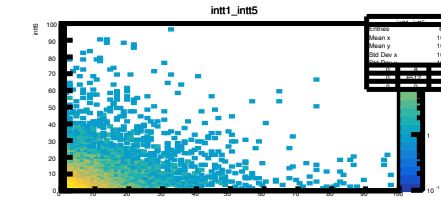
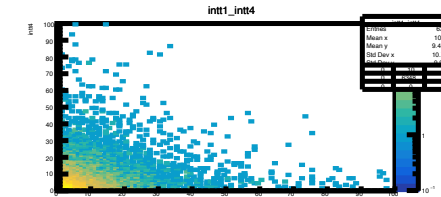
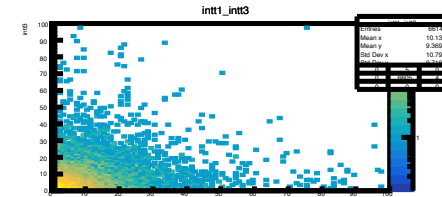
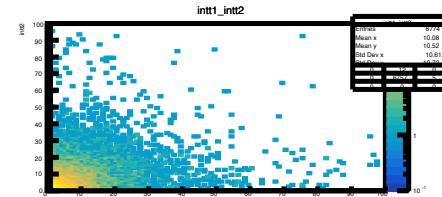
intt5

intt6

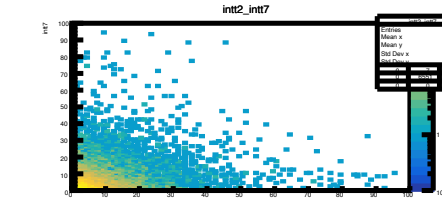
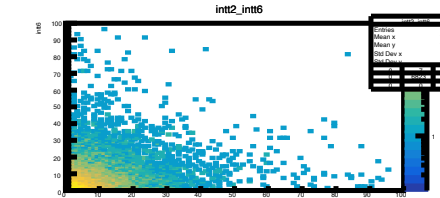
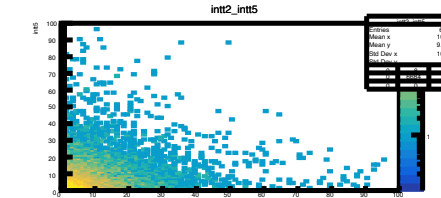
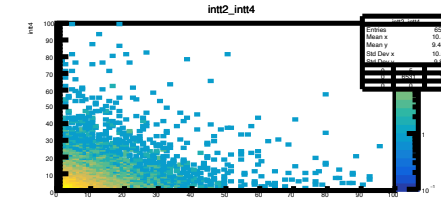
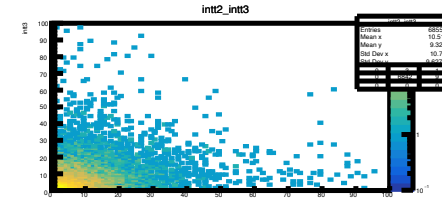
intt7



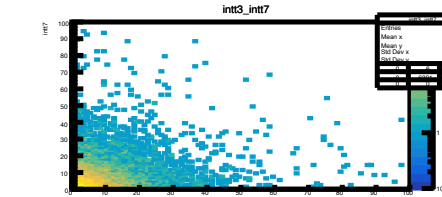
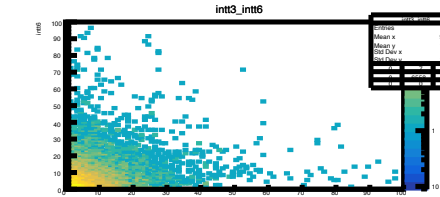
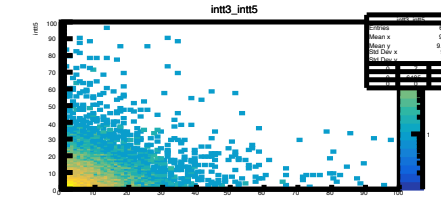
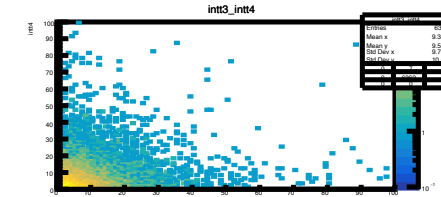
intt0



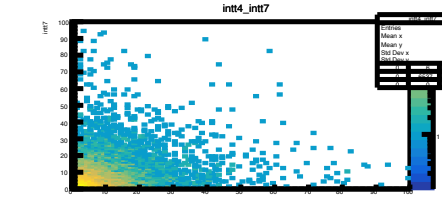
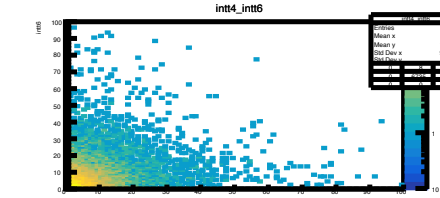
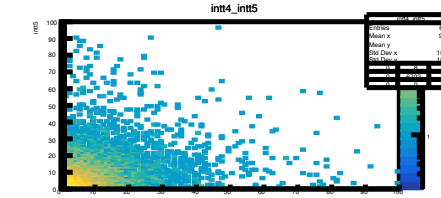
intt1



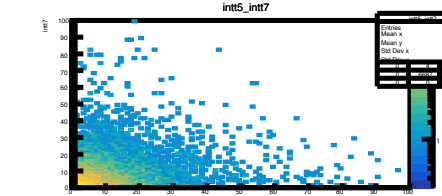
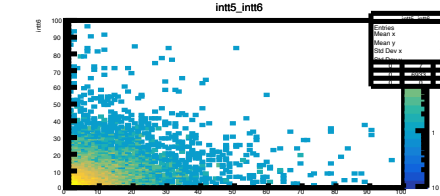
intt2



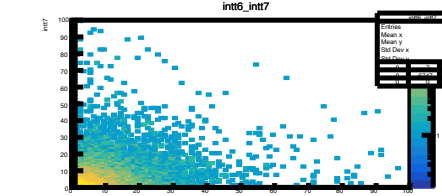
intt3



intt4



intt5



intt6

All correlations look similar.

x-axis

Correlation b/w FELIXes

Extended hits

Run 49737 (July/30)

10k events analyzed

Hot channel rejection applied

y-axis

intt1

intt2

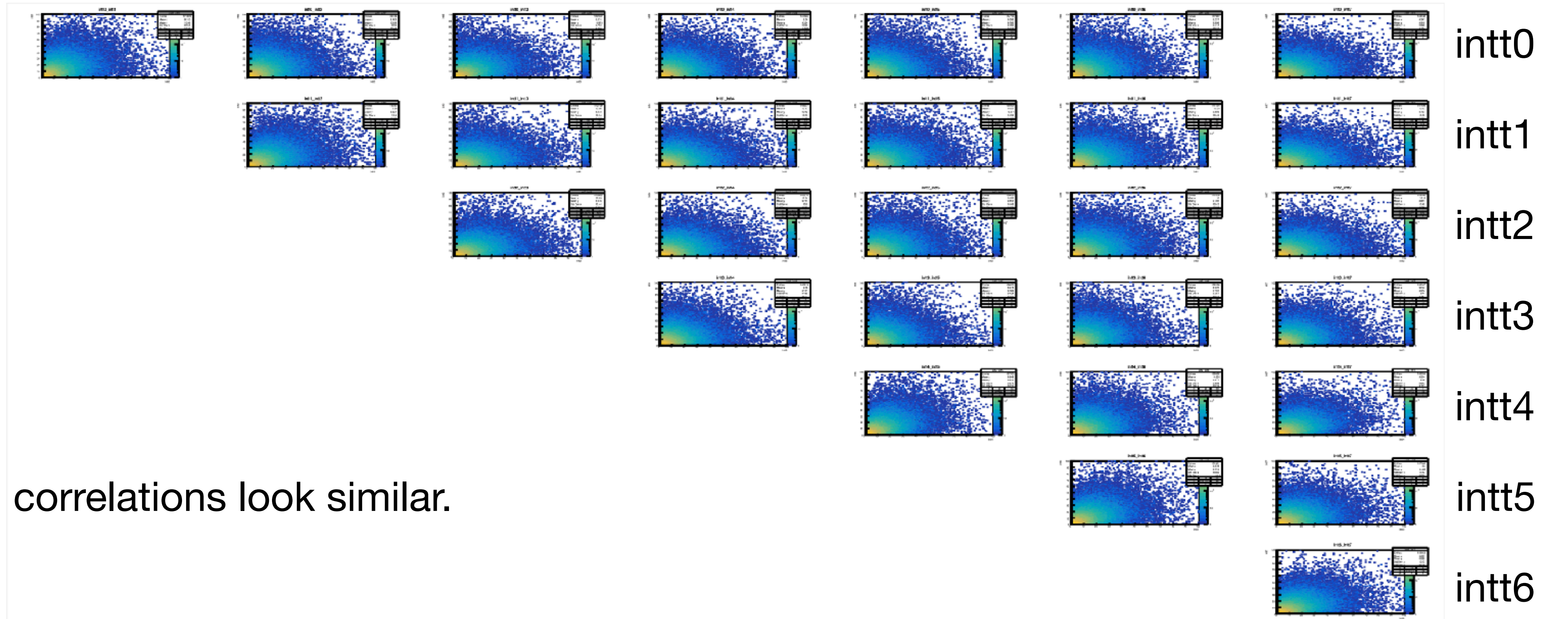
intt3

intt4

intt5

intt6

intt7



intt0

intt1

intt2

intt3

intt4

intt5

intt6

x-axis

Correlation b/w FELIXes

Streaming hits

Run 49743 (July/30)

10k events analyzed

Hot channel rejection applied

2 noisy half-ladders were masked at that time

y-axis

intt1

intt2

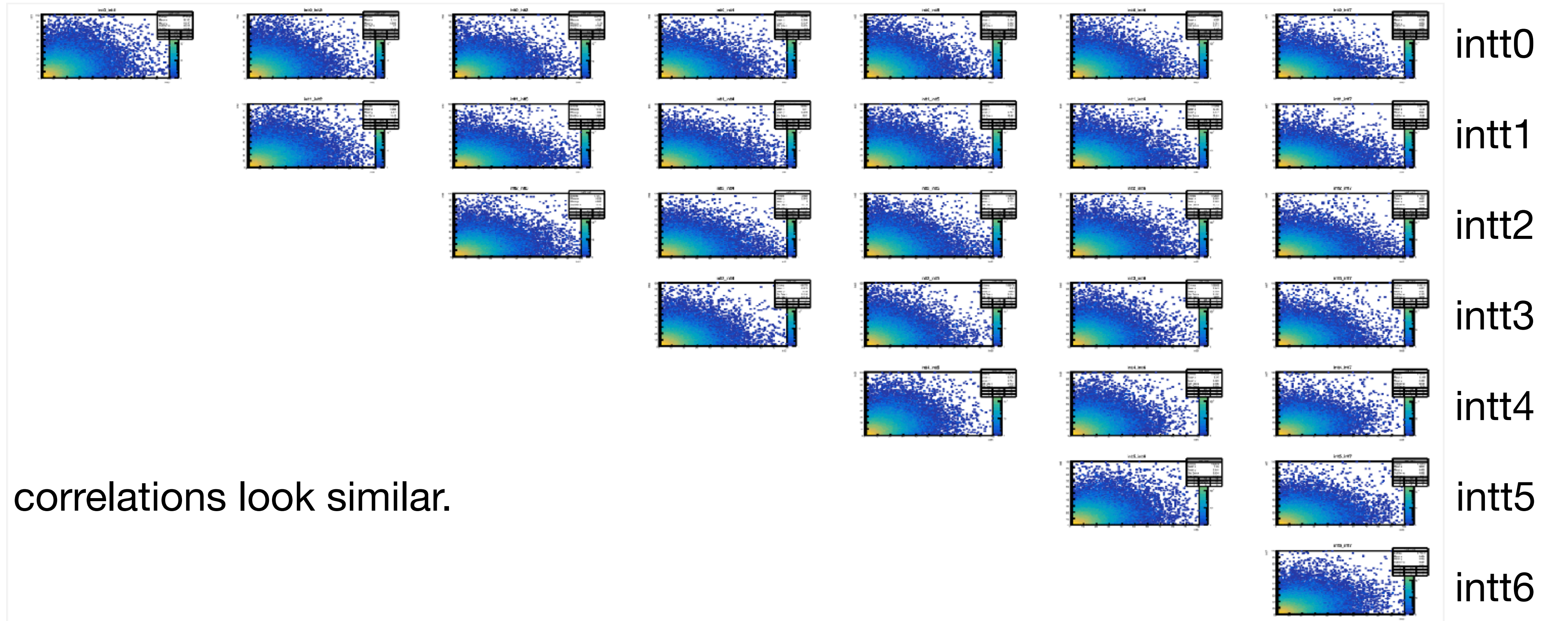
intt3

intt4

intt5

intt6

intt7



All correlations look similar.

x-axis

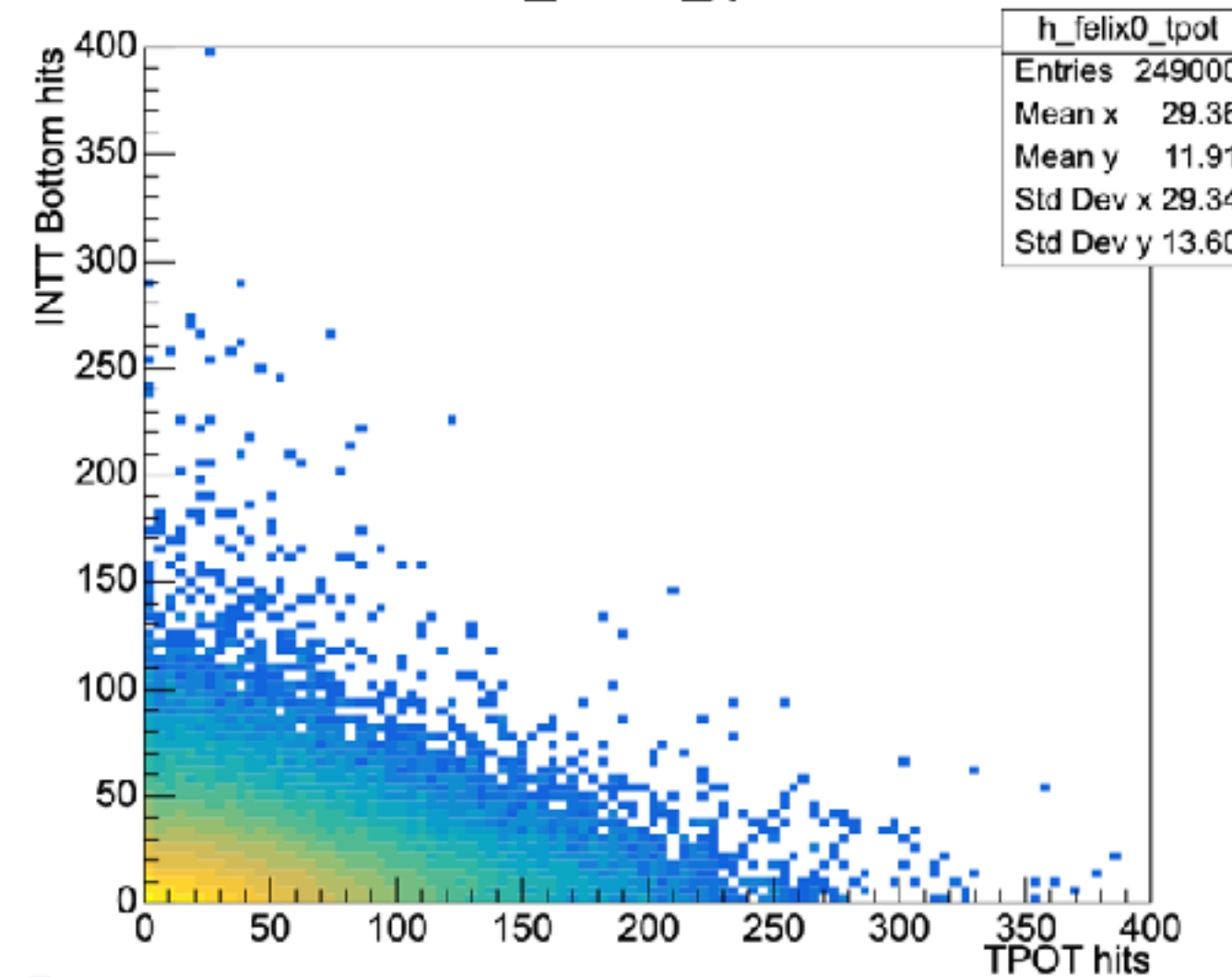
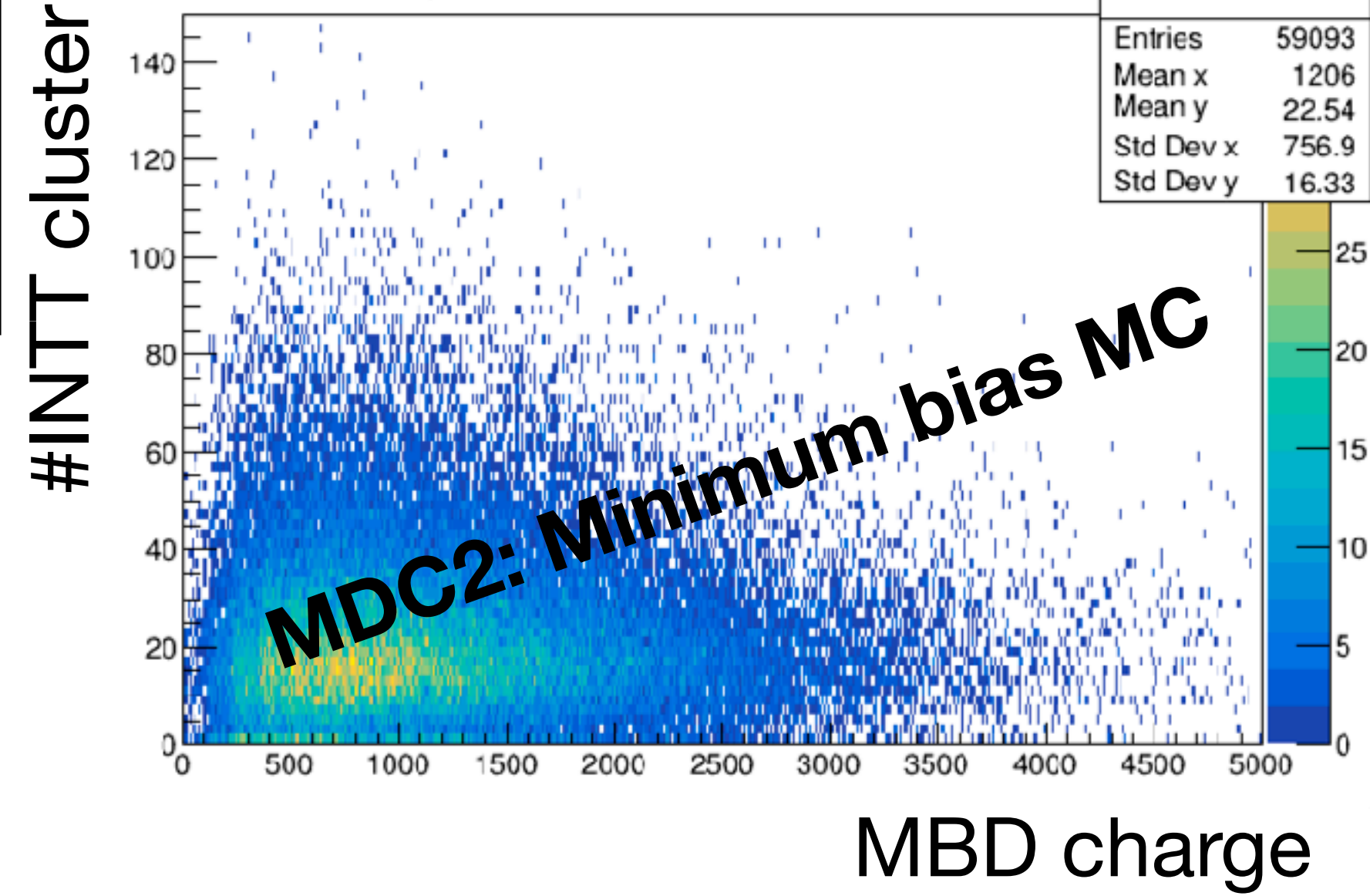
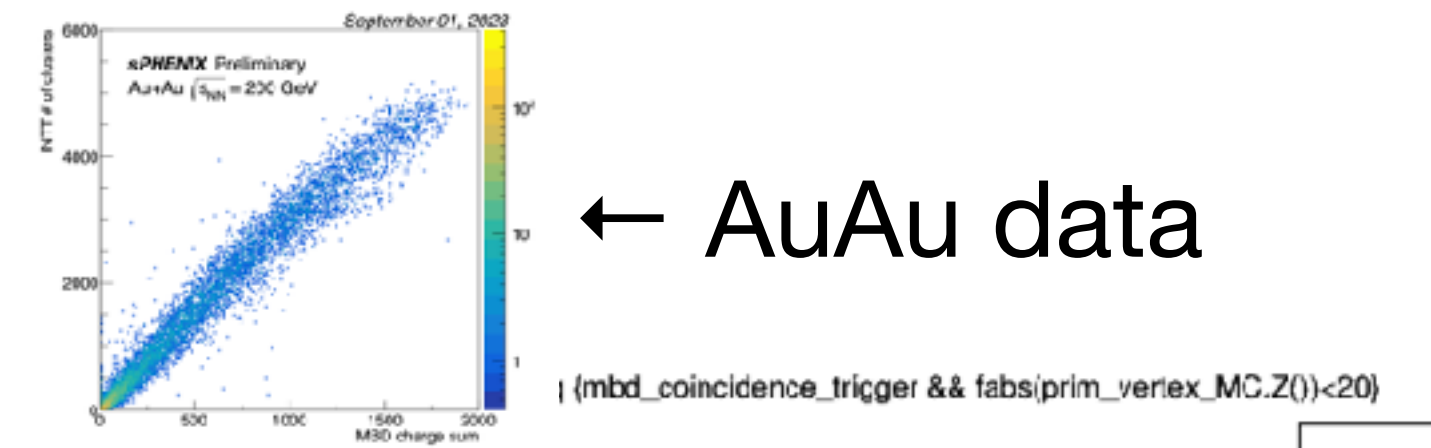
Next steps

Road Map to Establish Good Streamreadout Data
 STEPS to prove the stream-readout data are not collapsed and useful for the future analysis

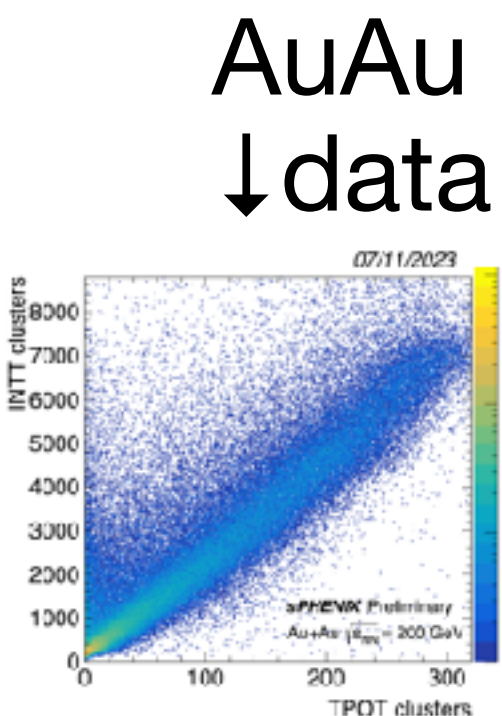
Itaru proposed a plan for this

To verify the healthiness of our streaming data,

- Checking increment check of BCO full ← Who's working on it?
- Seeing correlations of parameters of INTT
 - ✓ Inner barrel vs Outer barrel
 - ✓ FELIXes
 - South vs North
 - Top vs Bottom, ..., etc
- Looking at correlations with other subsystems
 - MBD: I checked MC data to have inspiration. I don't plan to analyze it.
 - TPOT: Jaein is working on it
 - MVTX: Ryota is interested in it. Jaein is working on it.
- Tracklet finding
 - I'm working on it.



Jaein's ongoing work for correlation with TPOT



Next steps to other direction

For coming meetings (HP2024, JPS meeting), we can ask preliminary for correlation plots as performance plots. What do we want to show?

- #hit: Inner vs Outer barrels (triggered + extended data)
- #hit: Inner vs Outer barrels (streaming data)
- #cluster: Inner vs Outer barrels (triggered data)
- #cluster: Inner vs Outer barrels (triggered + extended data) if possible
- #cluster: Inner vs Outer barrels (streaming data) if possible
- ADC distribution???
- MIP peak (not DAC scan data)
- Timing performance
- Hot channel condition?
- any idea?

