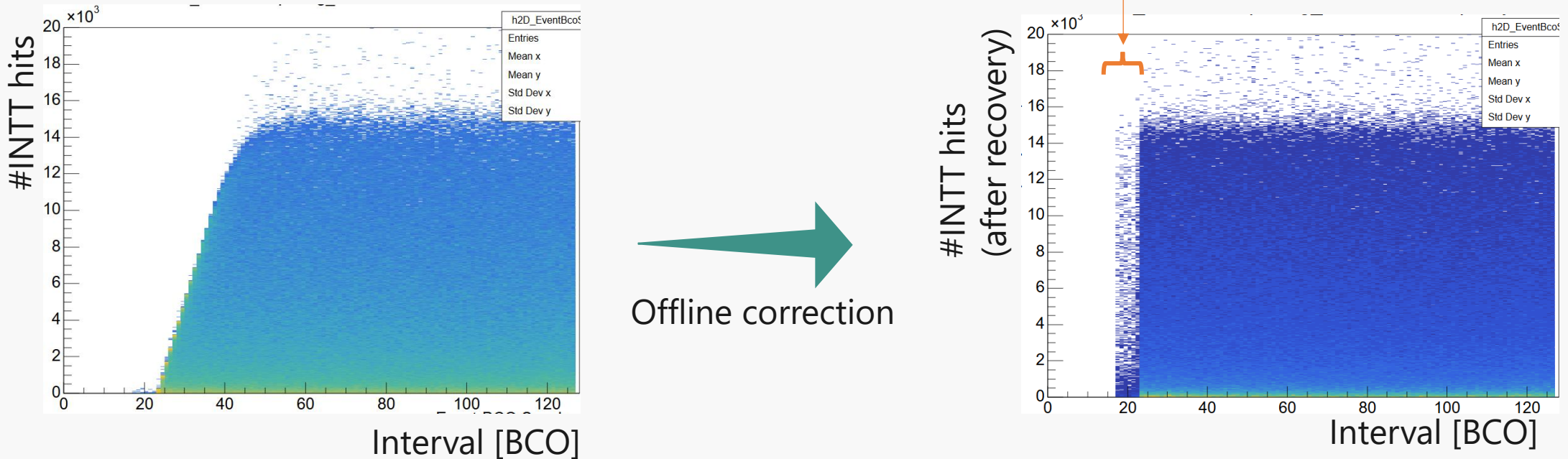
Hit Carryover: Updates on offline analysis

Ryotaro Koike
Kyoto University

Recap

- **Offline recovery procedure**
  1. Identify carryover candidates based on fphx_bco values.
     - ver. 1: the same value as previous event's mode
     - ver. 2: the value that falls in the trigger timing if bco_diff is calculated with previous bco_full
  2. Push candidates back following a criteria.
     - criteria ver. 1: should be in the first chank of the hit list
     - criteria ver. 2: no cut

- **The distinction at interval of 22**



The distinction was a fake/artifact!

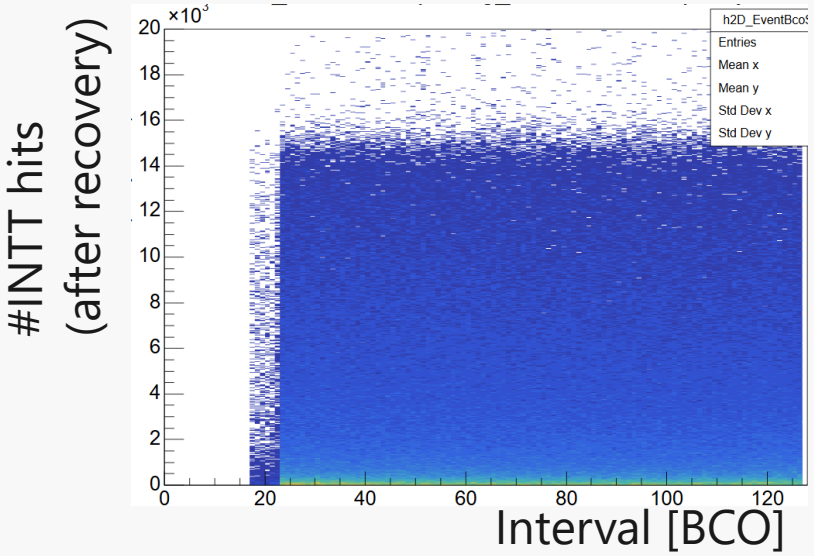
The entries drawn in the plot were those who passed some cut:

- (z_vertex < 10 cm., Next event have at least 1 hit., Next event have at least 1 carryover hit.)
- In the original event, bco_diff distribution have a peak in the trigger timing. ← This was an ill filter.

However, the peak tends to disappear especially in short interval case, resulting in cutting events that we should monitor.

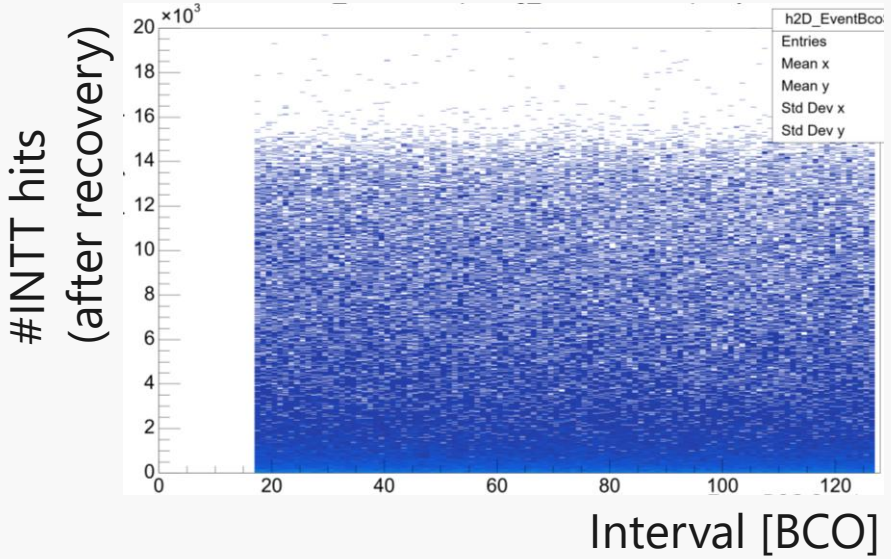
- **After removing the ill filter, the distinction disappeared.**

w/ the ill cut



w/o the ill cut

(Statistics is less than the left plot)



Tight correction

- **Offline recovery procedure**

New !

1. Pre-check that a condition for carryover is met.
2. Identify carryover candidates based on fphx_bco values.
3. Push candidates back following a criteria.

- **Reason**

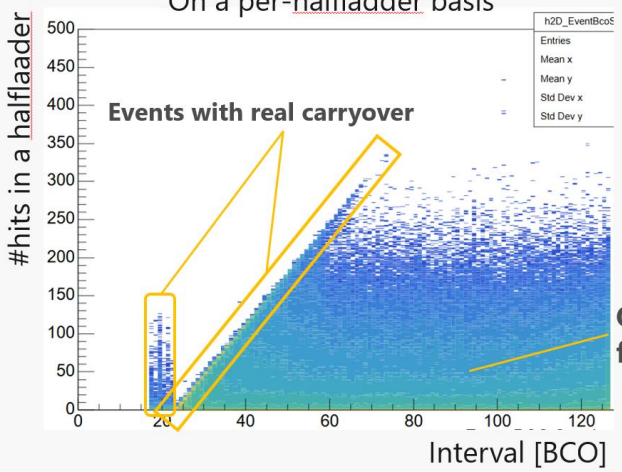
Our assumption of carryover mechanism:

- The number of hits to be processed exceeds some threshold.
- The excess will be carried over to the next event.

In this assumption, the number of hits recorded in the original event should be a certain number when carryover really occurred.

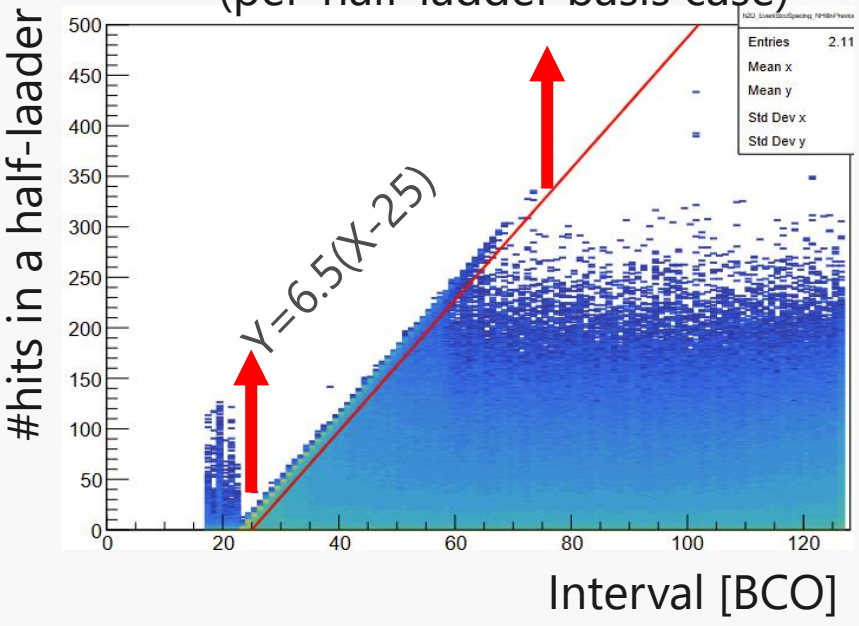
- **New requirement**
  - The recorded number of hits in some unit (half-ladder or chip) should be above a threshold.

On a per-halfladder basis



The new requirement

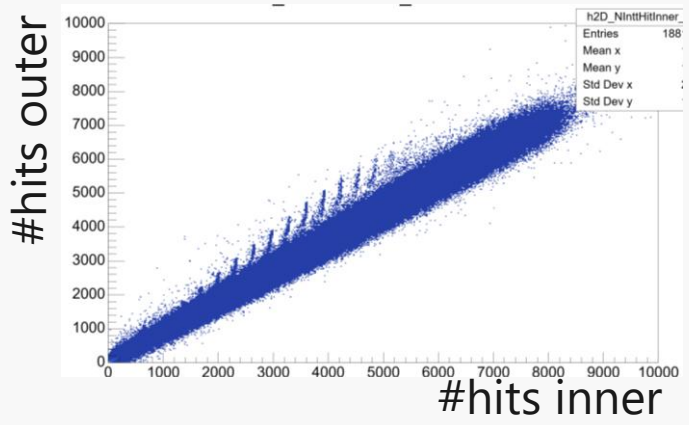
(per-half-ladder basis case)

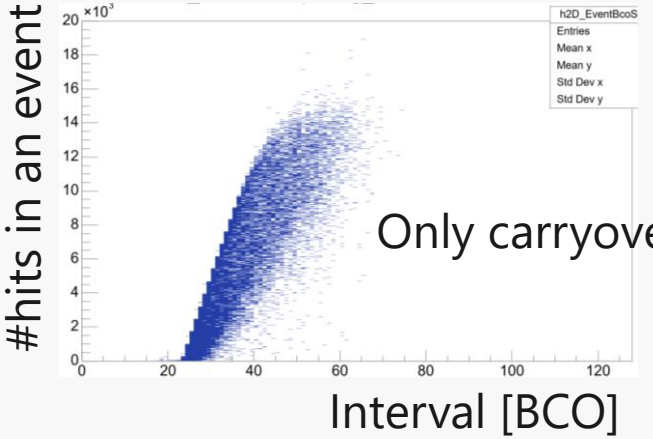


Tight correction

- **Results (per-half-ladder basis case)**
  Reasonable.

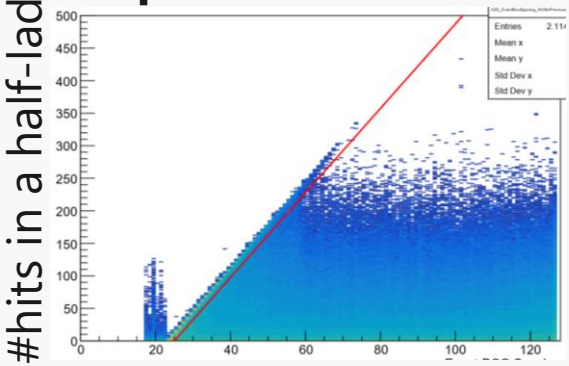
Before correction



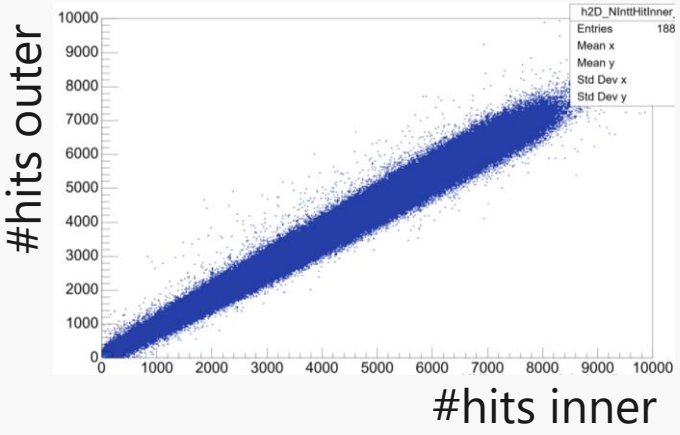


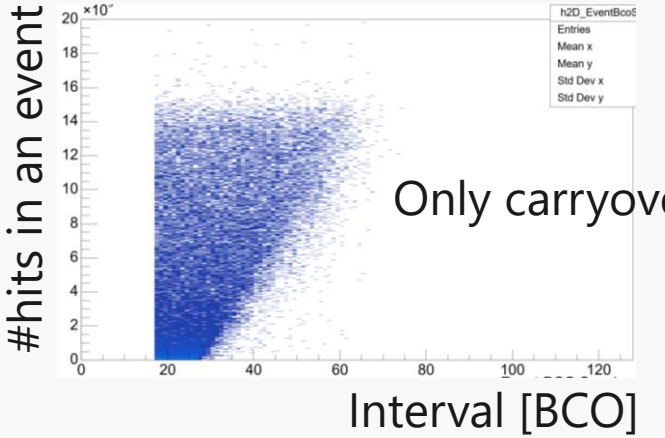
Only carryover event drawn

Requirement



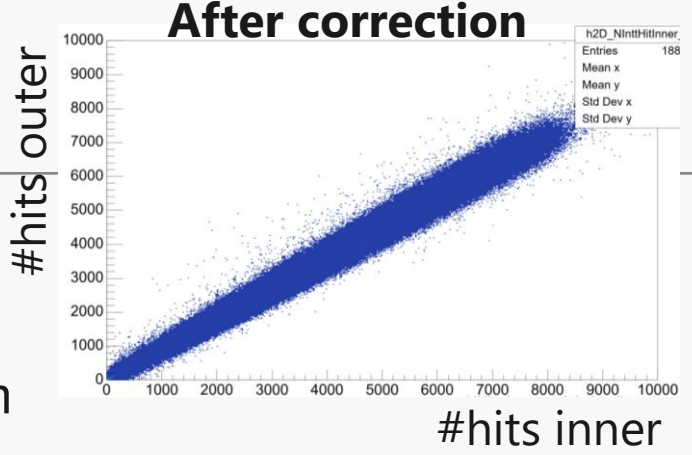
After correction



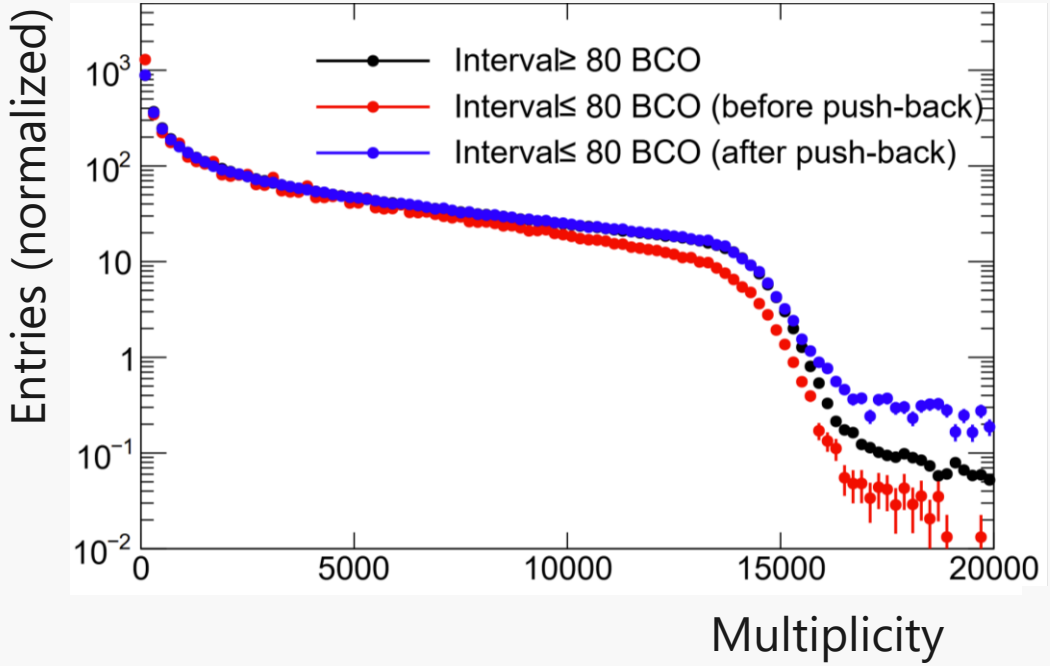


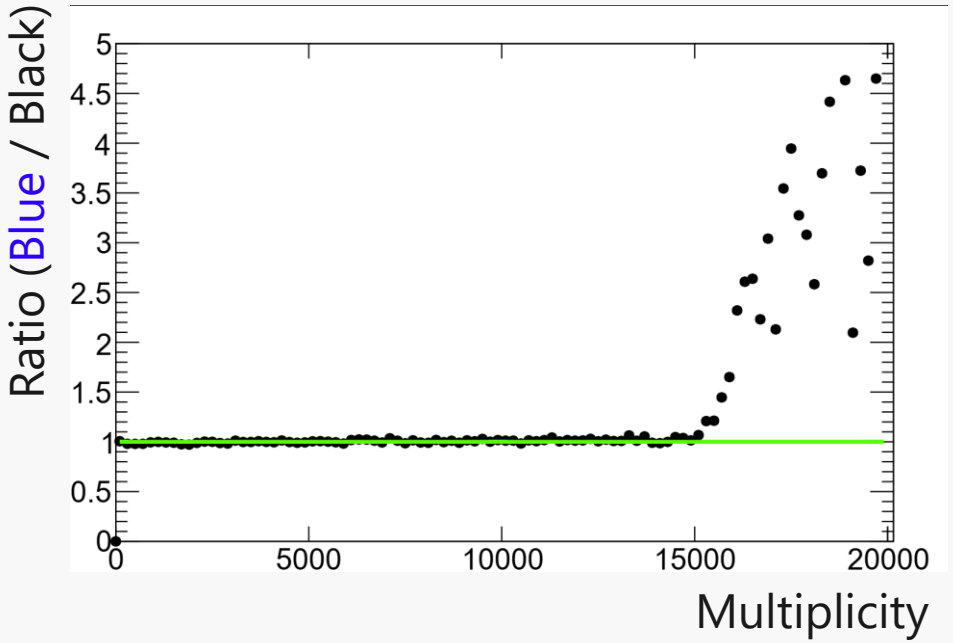
Only carryover event drawn

Tight correction



- **Results (per-half-ladder basis case)**
  - Corrected multiplicity looks reasonable.
  - Although the increase of the ratio in high multiplicity region may imply too much push-back.



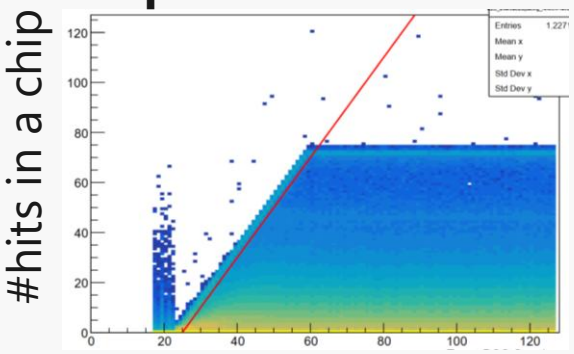


Total number of events ~ 2000 event/file * 4646 files = 9.292 M events
The number of events with interval > 80 = 9.292 M * 93 % = 8.642 M
The number of events with interval < 80 = 9.292 M * 7 % = 650 k

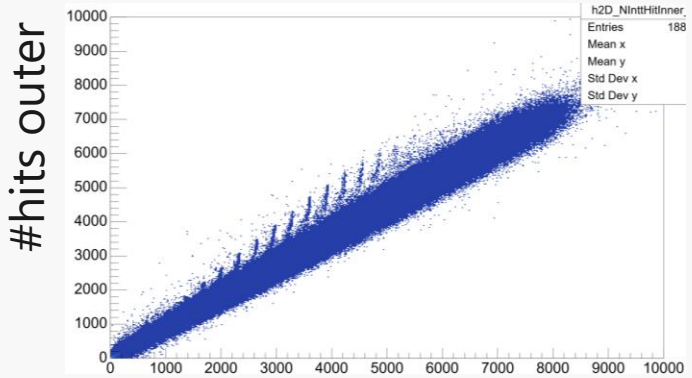
Tight correction

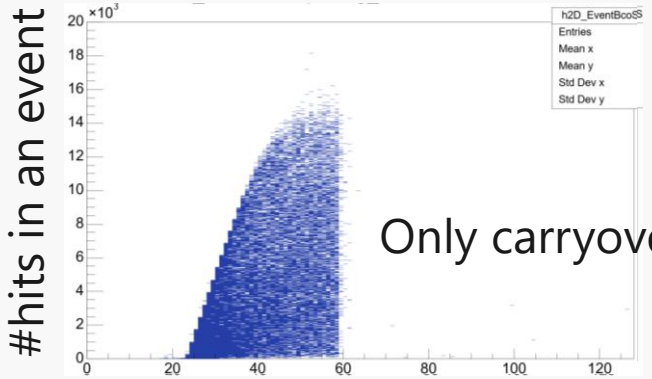
- **Results (per-chip basis case)**
  - Issue **not** recovered, which leads to a conclusion that carryover is not occurring chip by chip (but rather half-ladder by half-ladder!)

Requirement



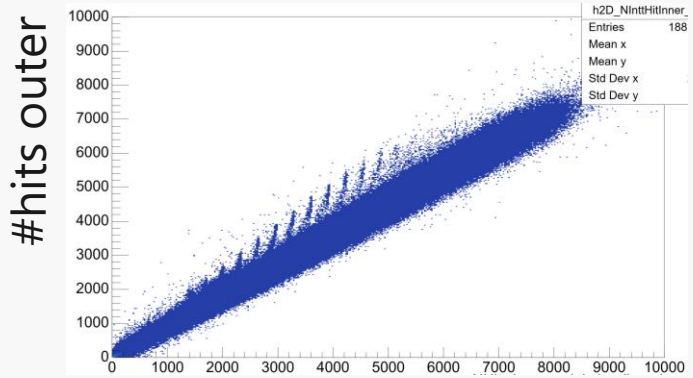
Before correction

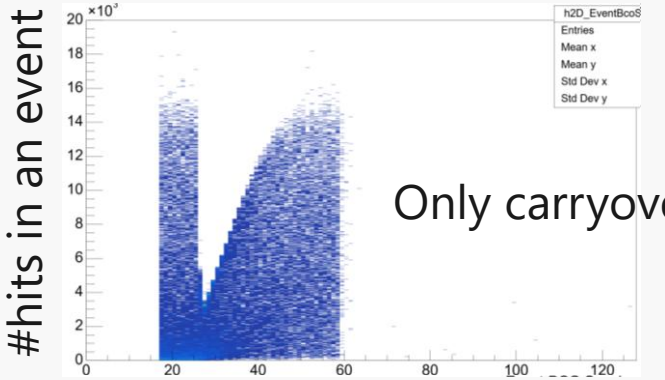




Only carryover event drawn

After correction





Only carryover event drawn

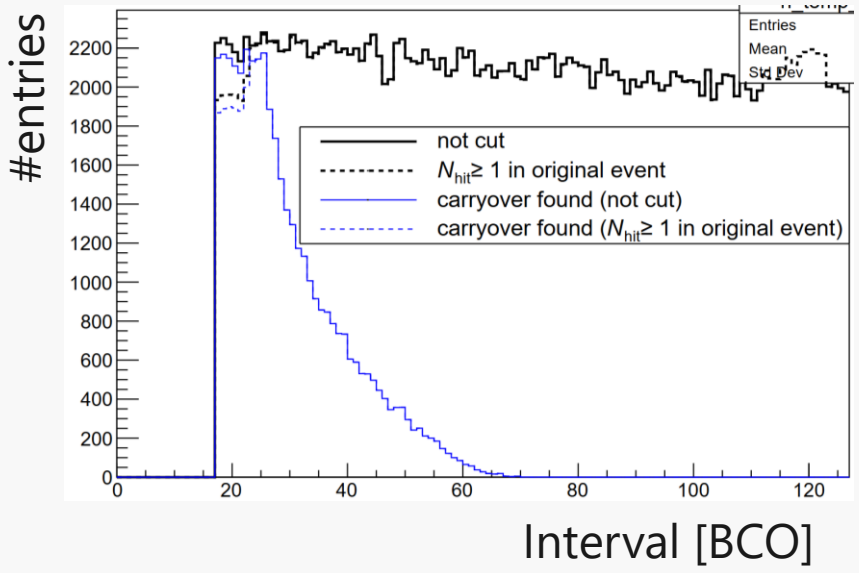
Fraction of carryover event

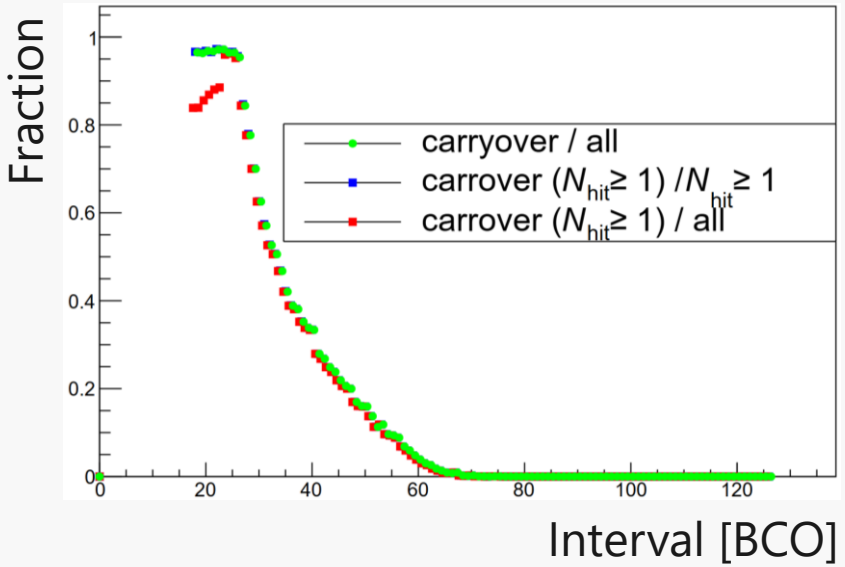
- **The fraction was calculated as a function of interval to the next event.**
  - Carryover event = an event in which at least 1 carryover hit was identified.
  - Denominator = #(carryover events)
  - Numerator = #(all events), or #(events with at least 1 INTT hit)

note: there are several definition of carryover hit

note: event intervals are calculated from GL1 info

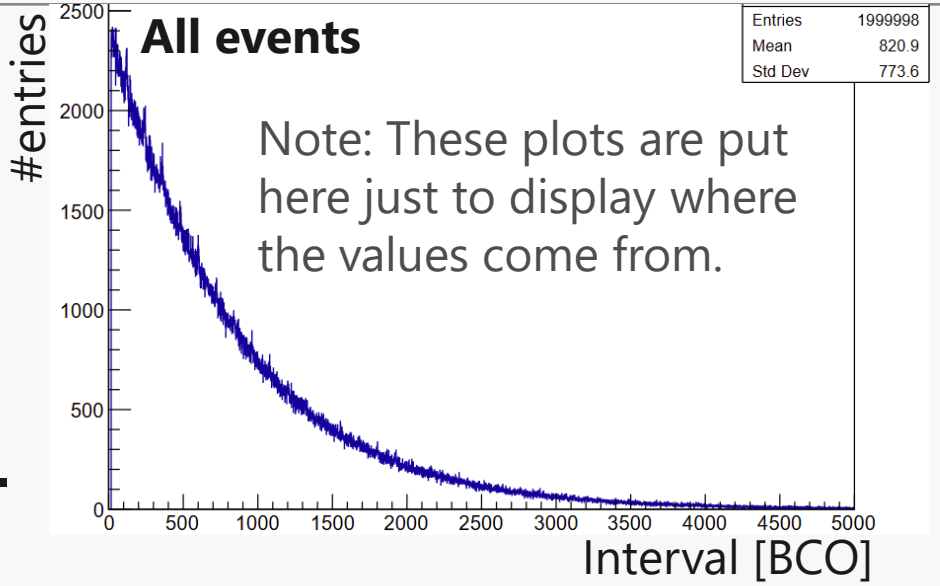
- **Results are reasonable**
  - The fraction reduces as interval is long.
  - The green line is a result of latest method.

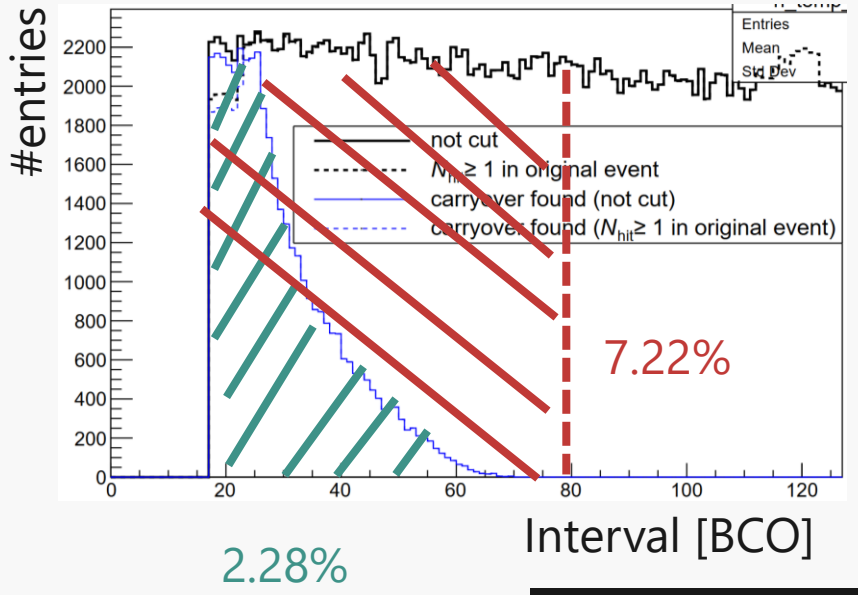


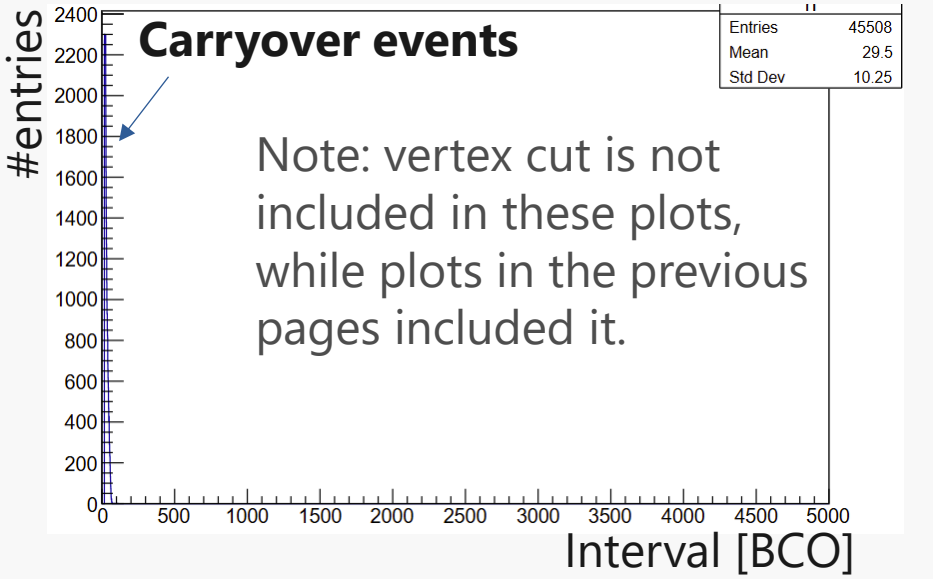


Fraction of carryover event

- **The overall fraction of carryover events**
  $45508/1999998 \approx 2.28\%$
- **If we cut carryover events by simply rejecting event with interval < 80, we will lose $144345/1999998 \approx 7.22\%$ statistics.**








```
root [7] tree->Draw("interval_from_previous >> h(5000,80,5000)", "interval_from_previous > 80")
(long long) 1853393
```

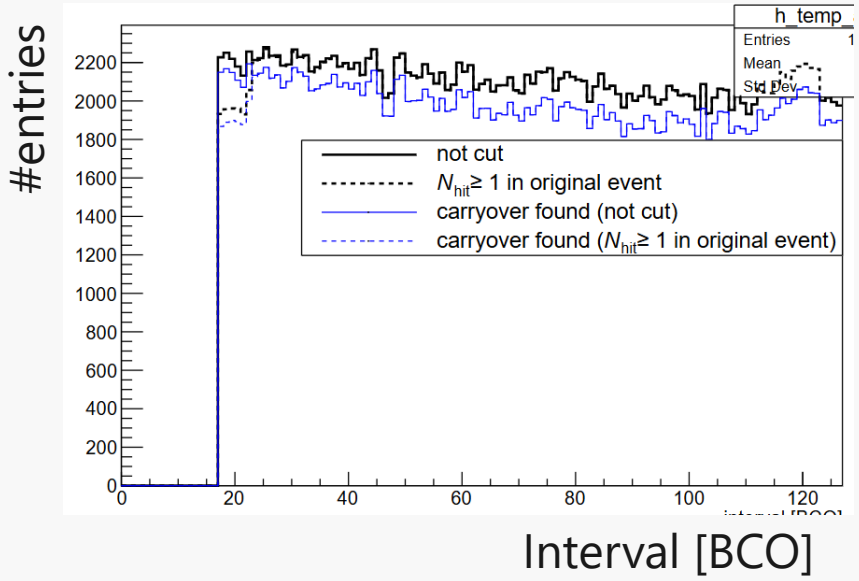
Fraction of carryover event

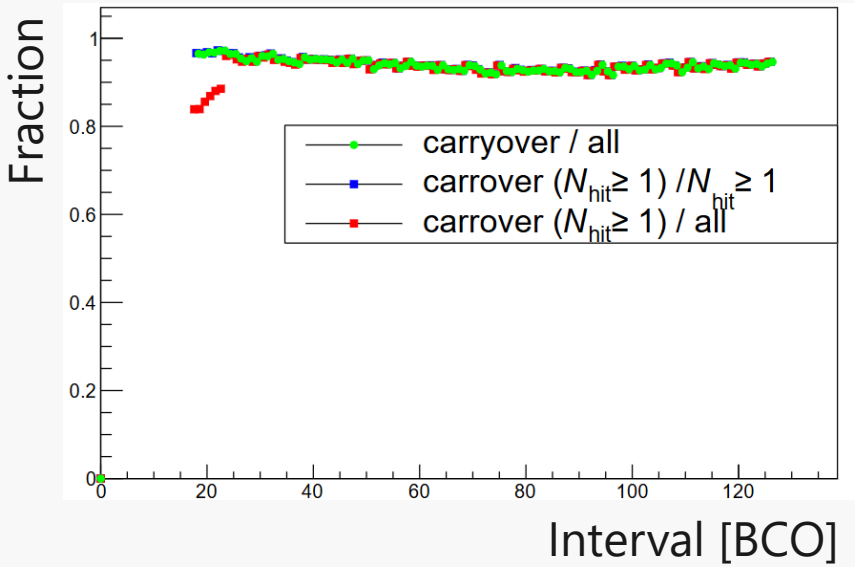
- **What if the pre-check is not applied?**

Offline recovery procedure (New !)

1. Pre-check that a condition for carryover is met.
2. Identify carryover candidates based on fphx_bco values.
3. Push candidates back following a criteria.

- **Too many carryover events.**
- **The pre-check significantly reduces fake carryover.**





Conclusion

- **The mystery of the distinction at interval 22 was resolved.**
  - It was caused by an ill filter when filling a histogram.
- **Method for identification of carryover hits are almost finalized.**

Offline recovery procedure

New !

1. Pre-check that a condition for carryover is met.
2. Identify carryover candidates based on fphx_bco values.
3. Push candidates back following a criteria.

- **The probability of having carryover event is ~2.3%.**
  The offline correction can save 7.22% of statistics.