

# Event Mixup

20240524

INTTJPMT

NWU M2 加納麻衣

# Event Mixup について分かっていること

- 1つ前のイベントのヒット数と相関がある
- 去年の最後にあったFelixアップグレード後にMixupの起こりやすさが変わった (1つ前のイベントのヒット数:1500以上→3000以上)
- アップグレードについて、Felixに繋がっている2つのROCに対して発するFPHXカウンターリセットタイミングを揃えた
  - ⇒ ヒットにタグ付けされるBCOの変化に変化はあり得るが  
確認されたこの現象に対して直接説明はつかない

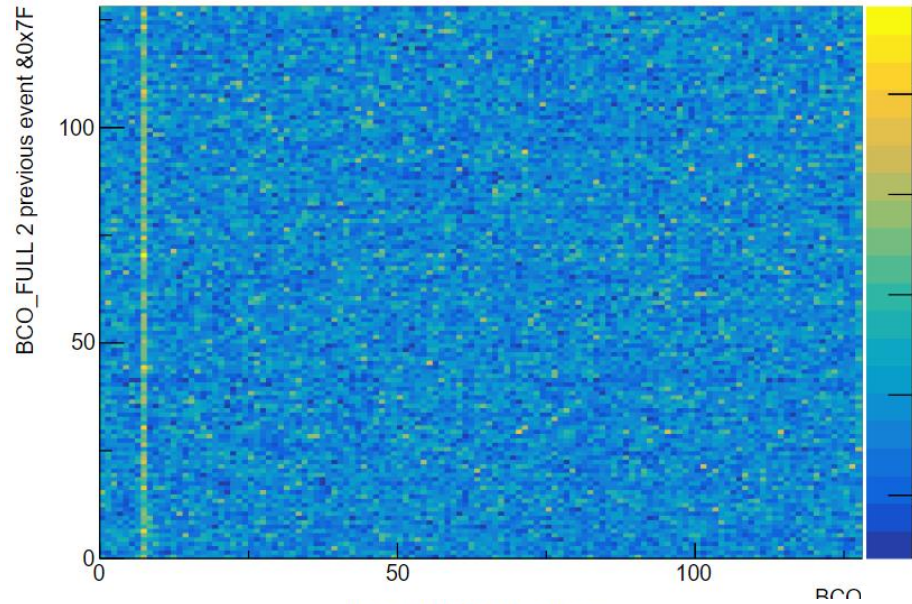
## 進捗（全てRun23のデータ）

- 2つ前のイベントからのリークがあるかどうか
- Mixupイベント及びヒットの定量化
- Mixupヒットが1つ前のイベントからのコピーであるかどうか

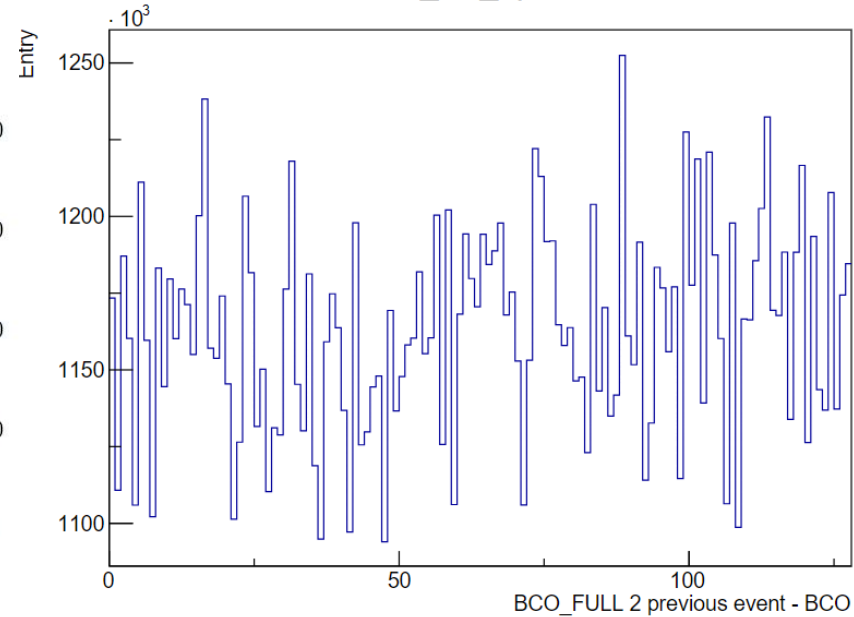
# 2つ前イベントからのリーク (アップグレード前 Run20708intt1)

2つ前のイベントのBCO\_FULLと今のイベントのBCOによるプロット

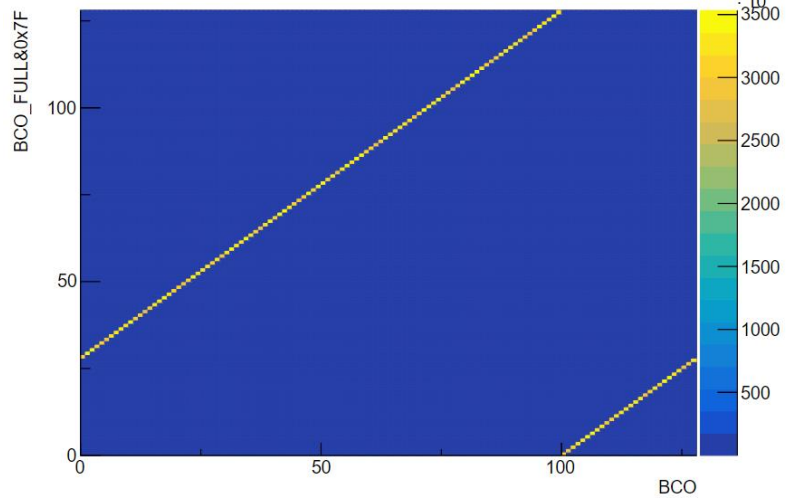
bco\_full&0x7F\_2prev vs bco 20708



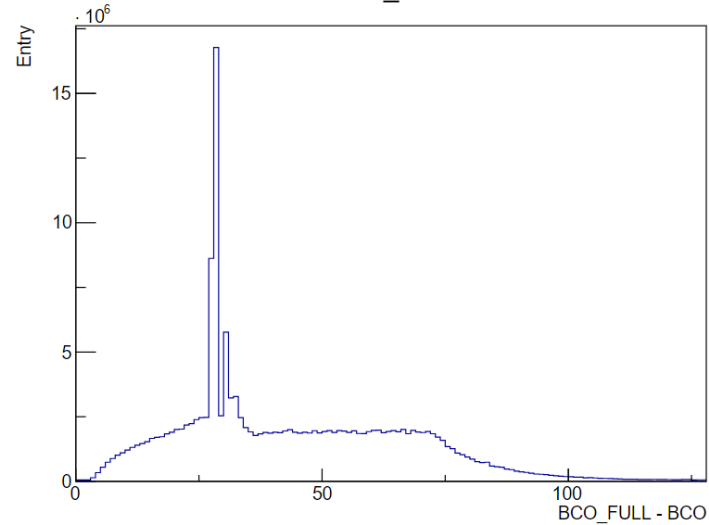
bco\_full\_2prev-bco



bco\_full&0x7F vs bco



bco\_full-bco

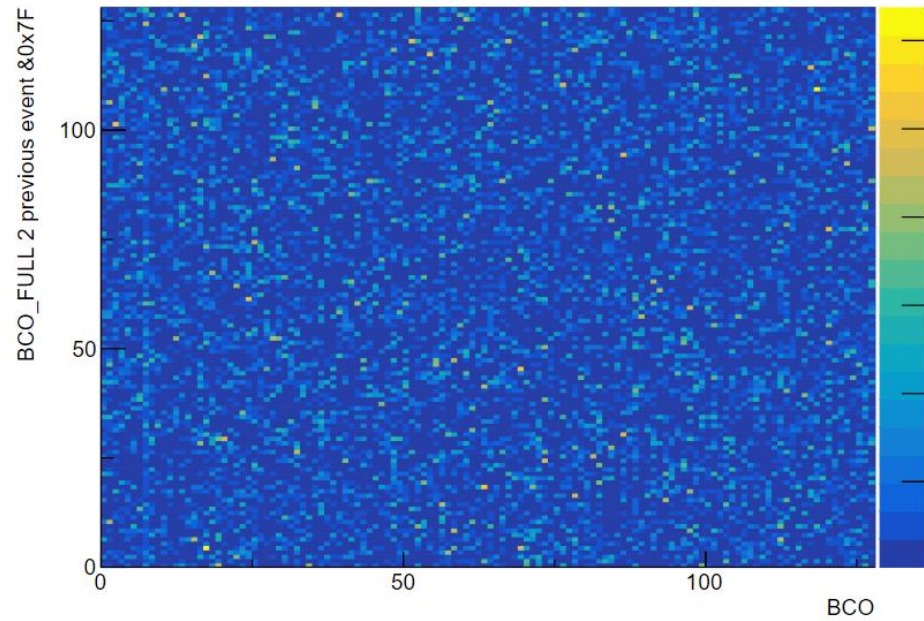


- 同じイベントのBCO\_FULL、BCOのプロットと同じ位置に相関やピークは見られない  
→リークなし

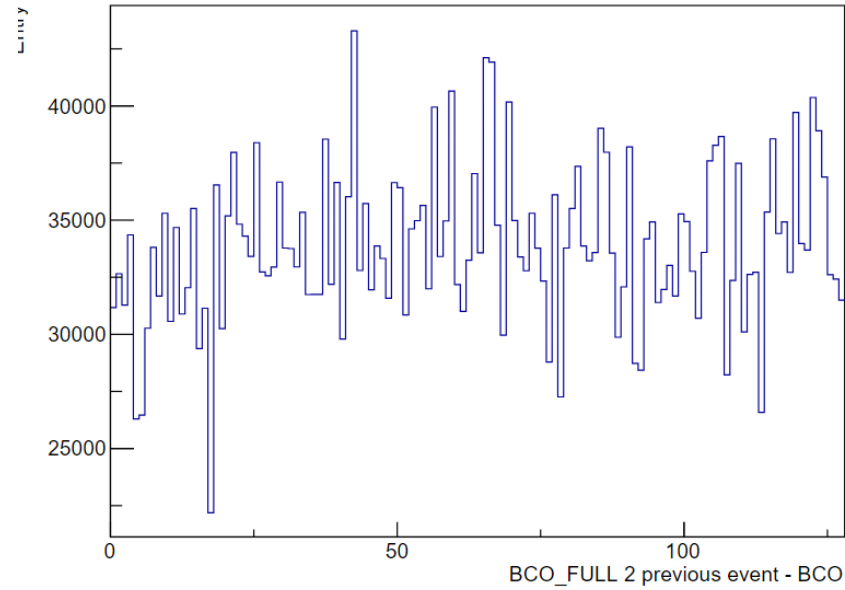
# 2つ前イベントからのリーク (アップグレード後 Run24768intt1)

2つ前のイベントのBCO\_FULLと今のイベントのBCOによるプロット

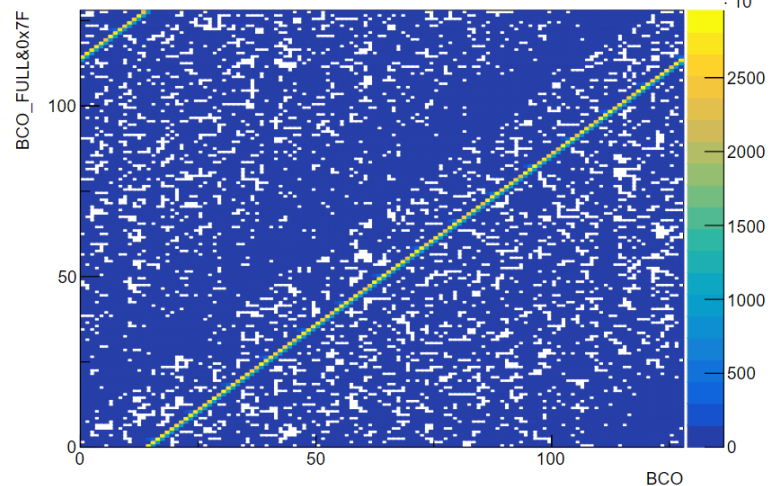
bco\_full&0x7F\_2prev vs bco 24768



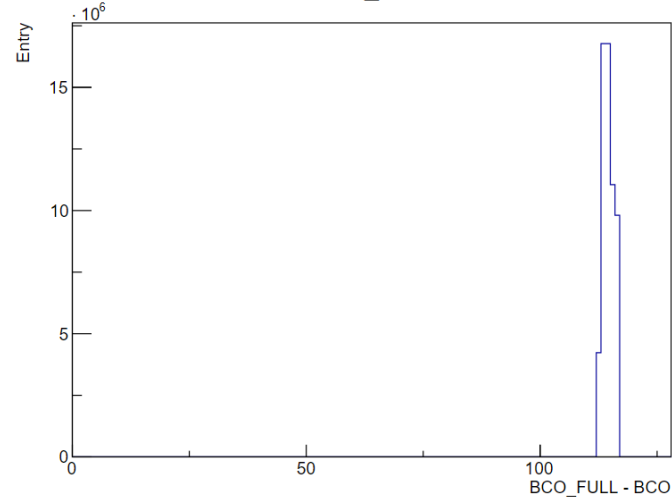
bco\_full\_2prev-bco



bco\_full&0x7F vs bco



bco\_full-bco

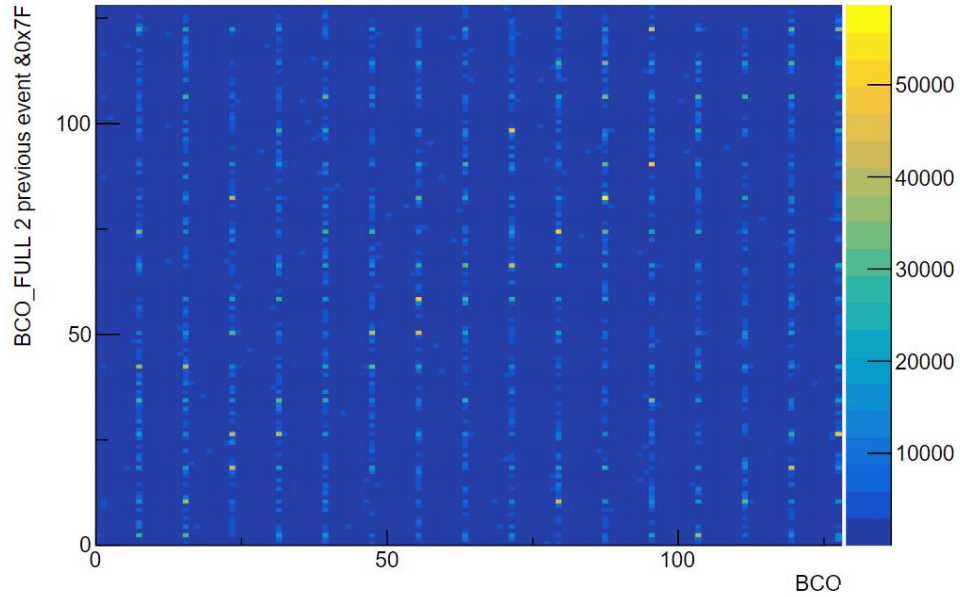


- 同じイベントのBCO\_FULL、BCOのプロットと同じ位置に相関やピークは見られない  
→リークなし

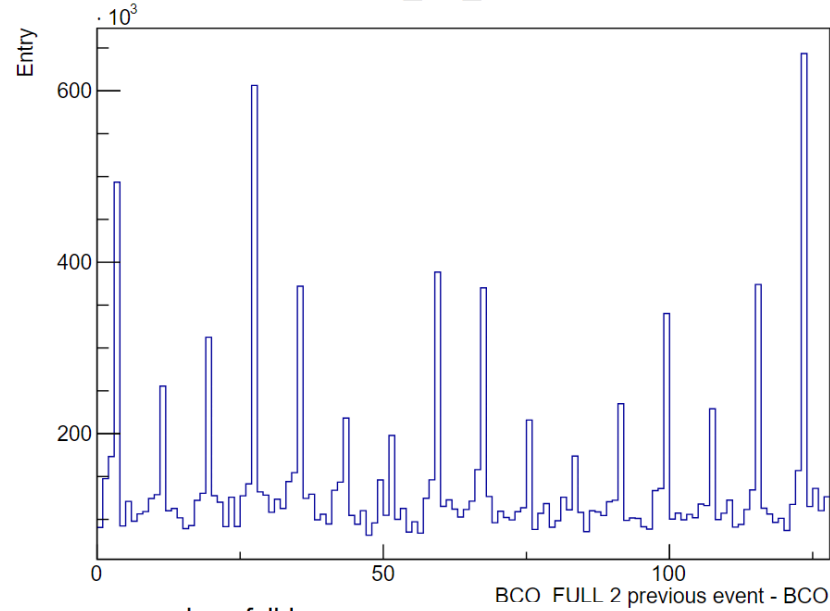
# 2つ前イベントからのリーク (アップグレード後 Run24768intt0)

2つ前のイベントのBCO\_FULLと今のイベントのBCOによるプロット

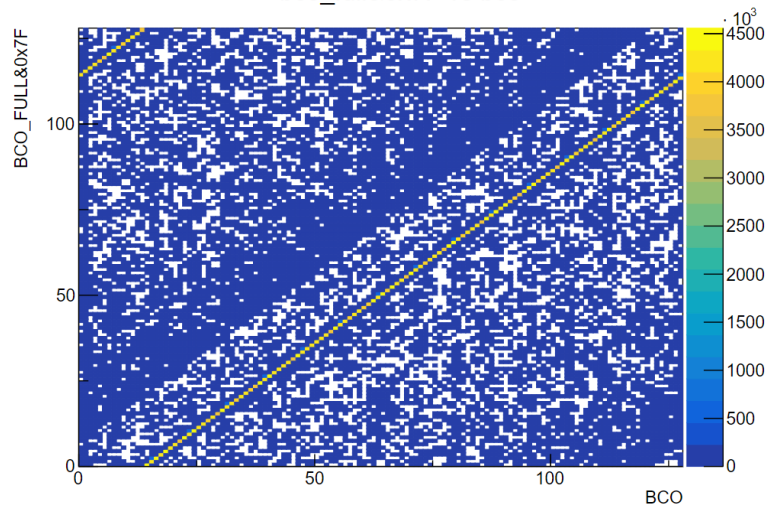
bco\_full&0x7F\_2prev vs bco 24768



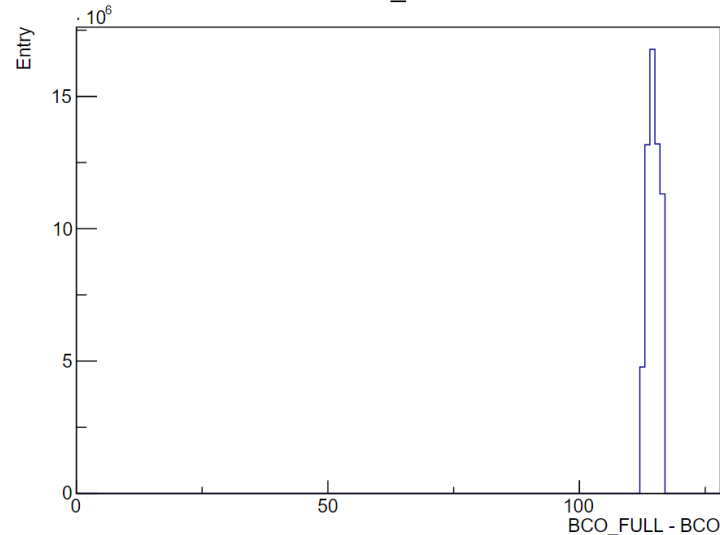
bco\_full\_2prev-bco



bco\_full&0x7F vs bco



bco\_full-bco



- ノイズの形は異なるがリークしていないように見える

# Mixupイベント及びヒットの割合の計算

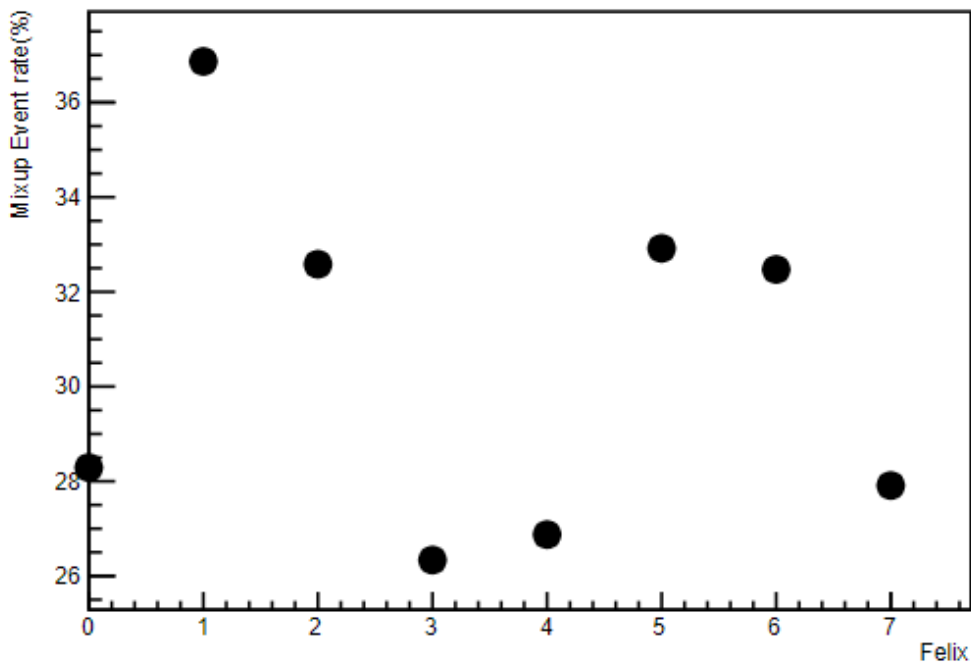
- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100  
**Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均をとった  
まだエラーバーを付けていない

Mixup hit があるイベント: Mixup hitの数 > 0 かつ 1つ前のイベントのヒット数 > 0

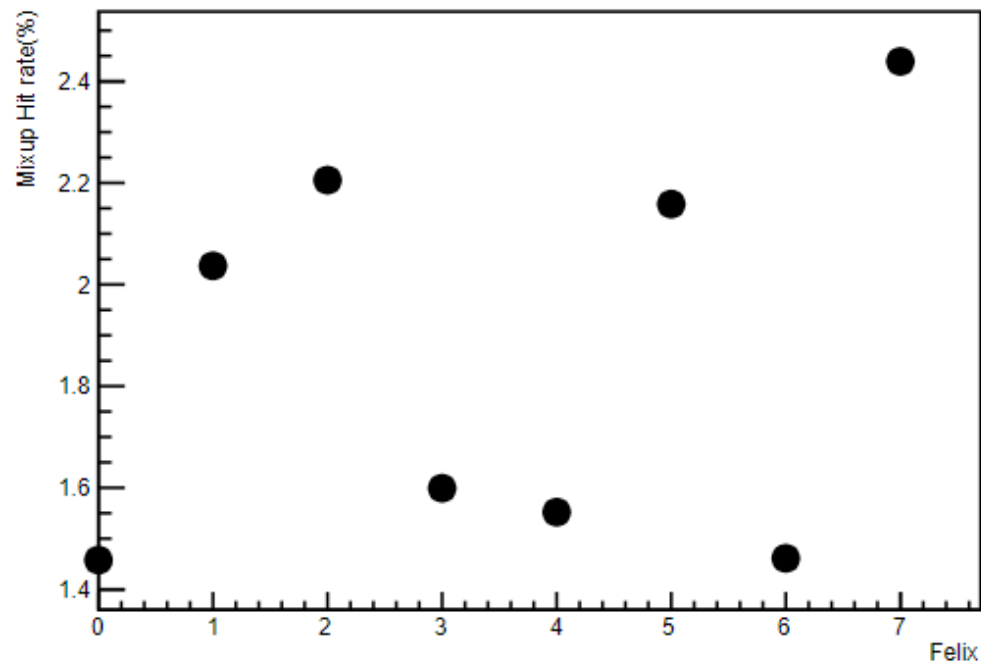
# アップグレード前 Run20444

- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100  
**Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均

Mixup event rate



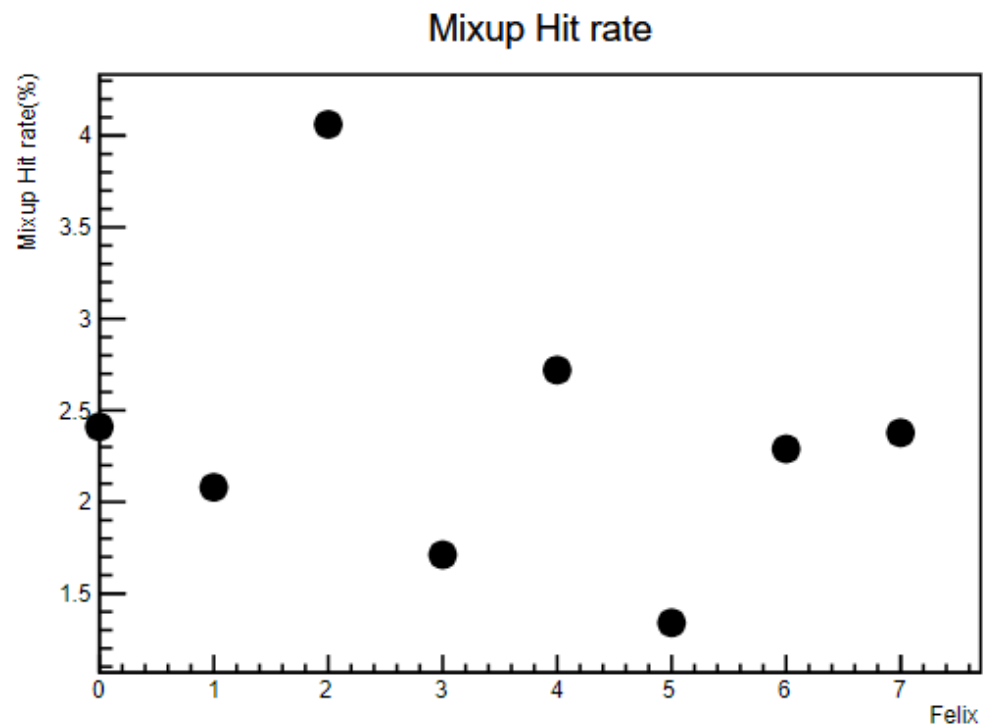
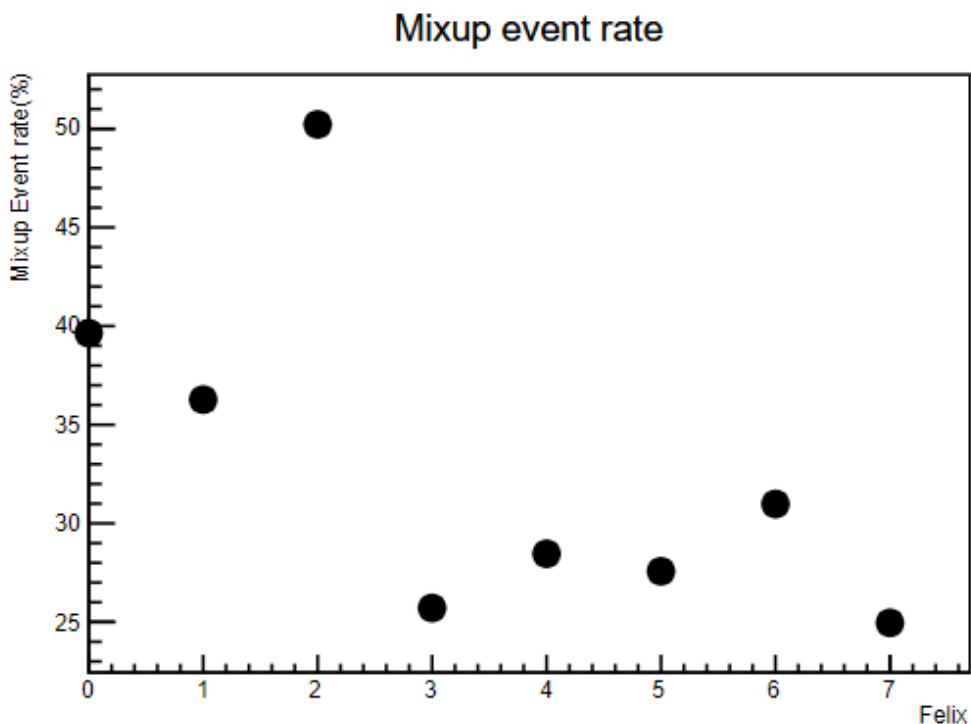
Mixup Hit rate





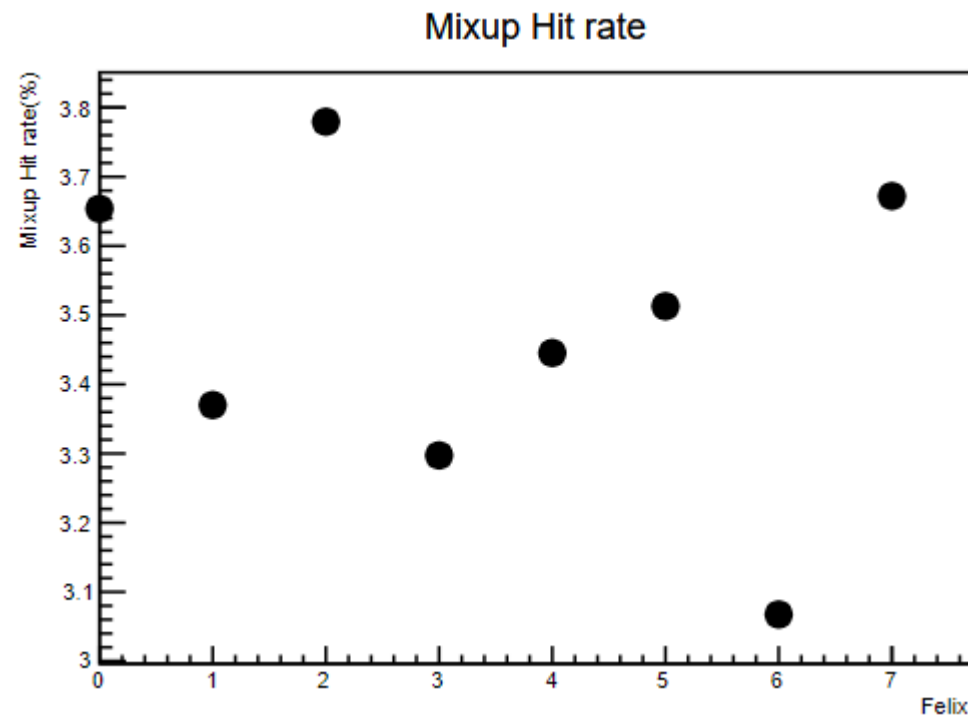
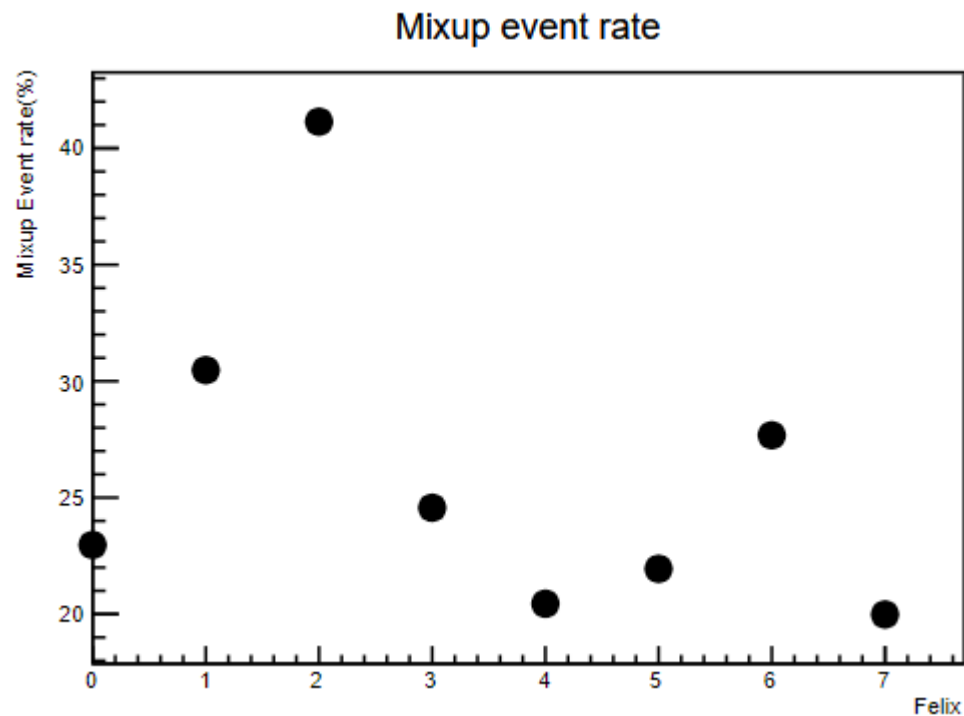
# アップグレード前 Run20708

- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100
- **Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均



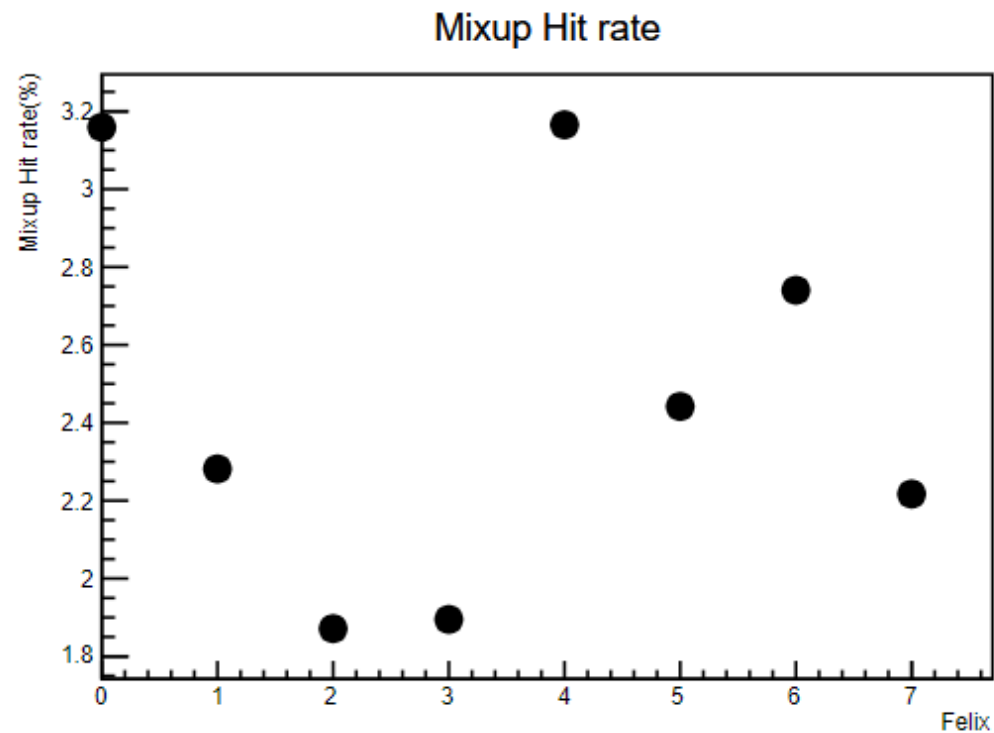
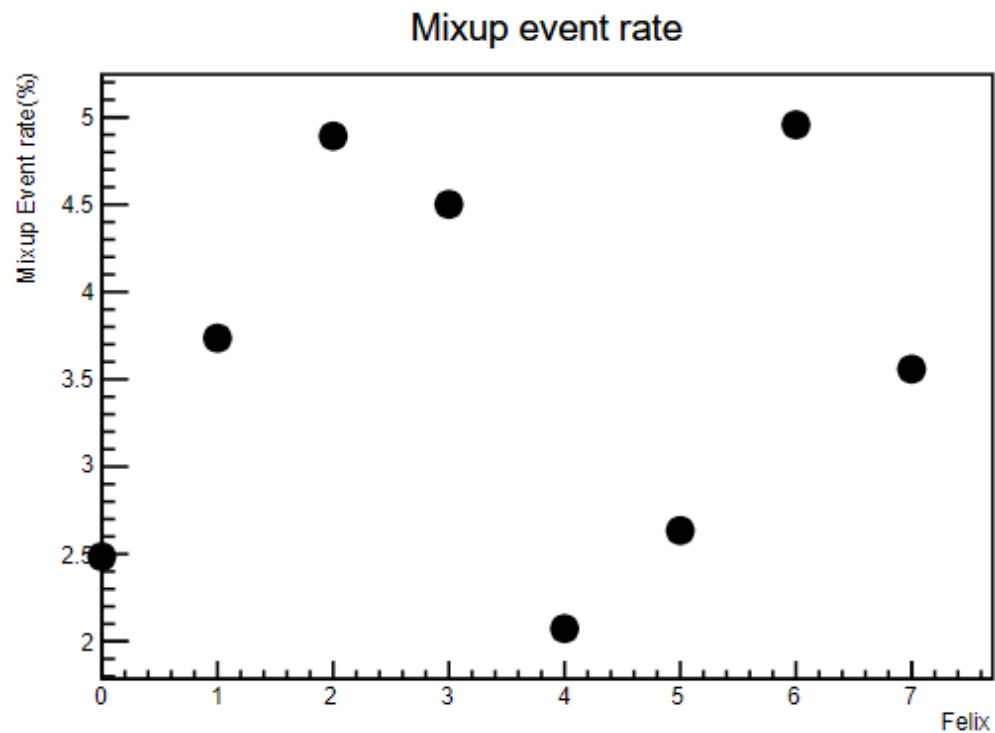
# アップグレード前 Run20869

- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100
- **Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均



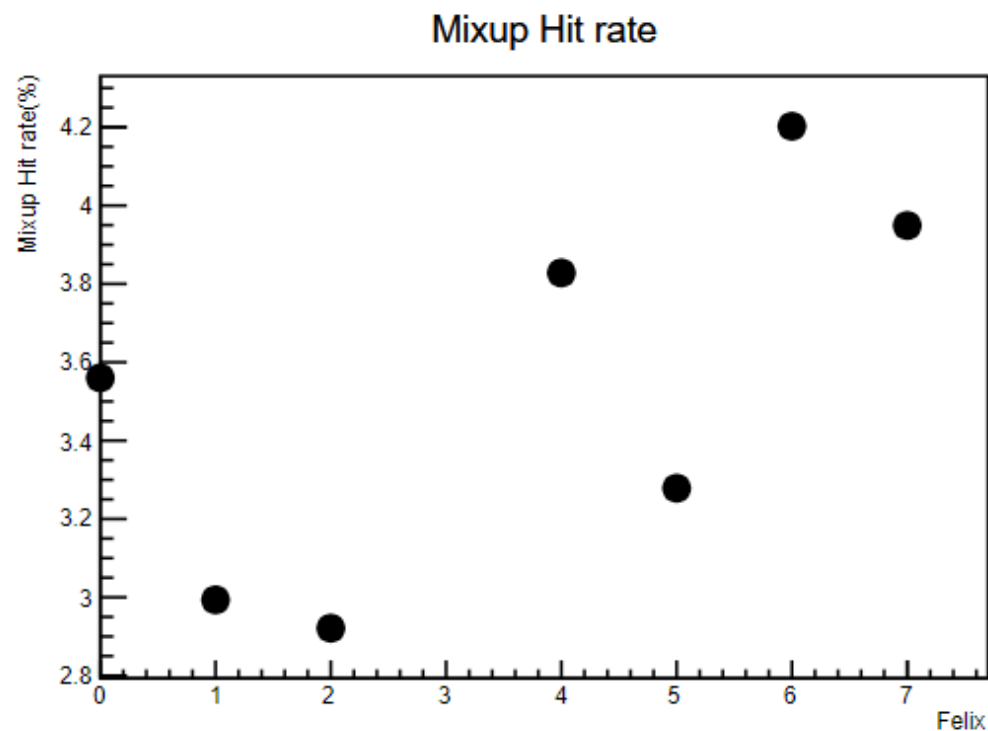
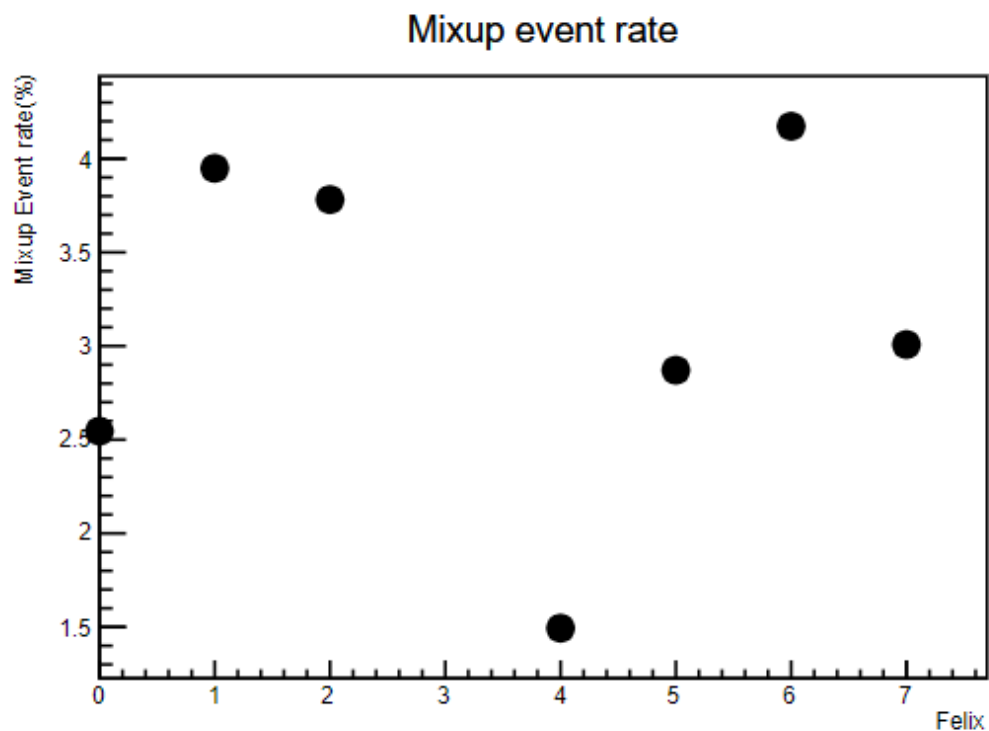
# アップグレード後 Run23947

- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100
- **Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均

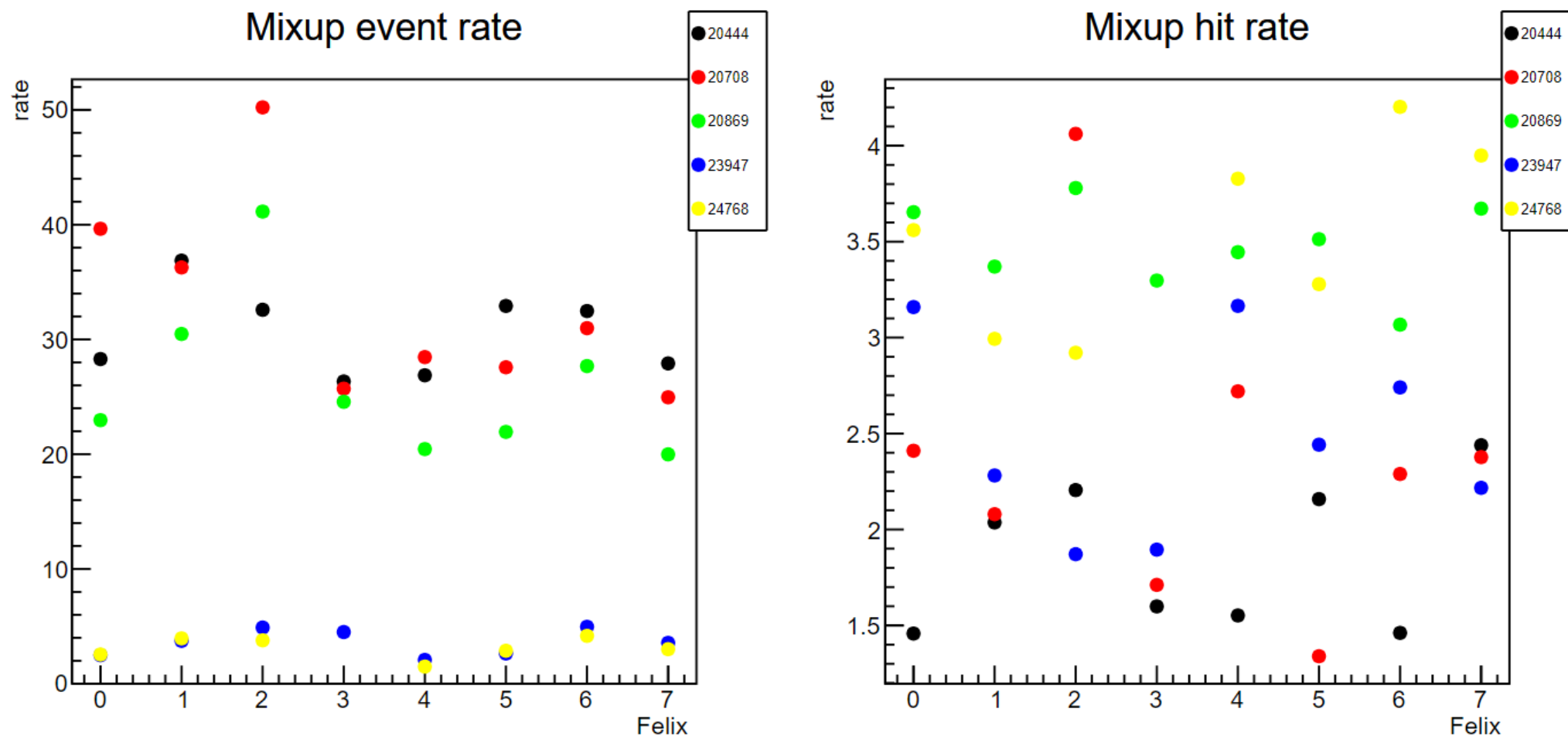


# アップグレード後 Run24768

- **Mixup event rate** = ((Mixup hitがあるイベント数)/(Entries)) \* 100
- 各イベントごと **rate** = ((Mixup hitの数)/(イベント内のヒット数+Mixup hitの数)) \* 100
- **Mixup hit rate** = (rate)/(Mixup hitがあるイベント数) 上記の平均



# 全て重ね書きしたグラフ



Event rate : アップグレード前後で変化あり (割合多→少)

Hit rate : 前後で変化なし 1~5%

# コピーヒット

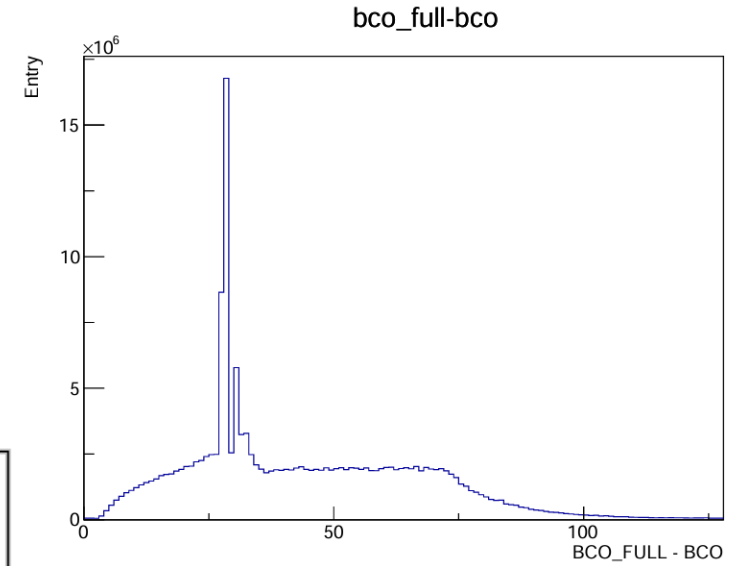
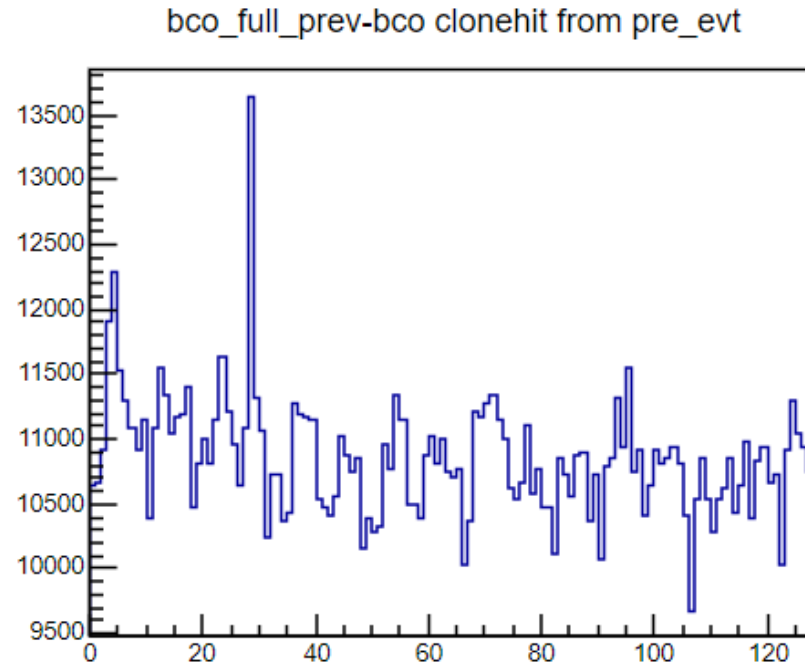
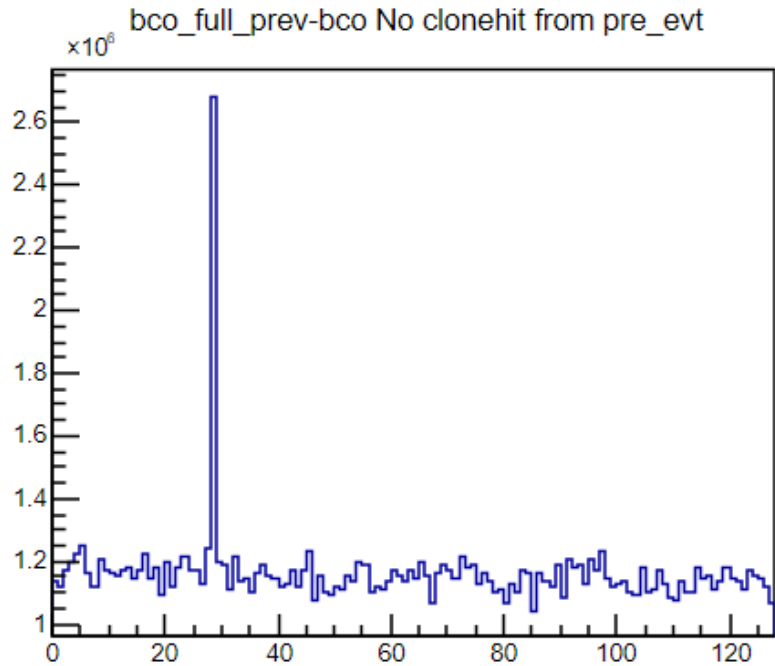
- Mixupヒットが、1つ前のイベントからのリークではなく、コピーヒットであるかどうか
- コピーヒット：module , chip\_id, chan\_id , adcが全て一致するヒット

# Run20708 intt1 アップグレード前

1つ前のイベントのBCO\_FULL - BCO

コピーヒットなし

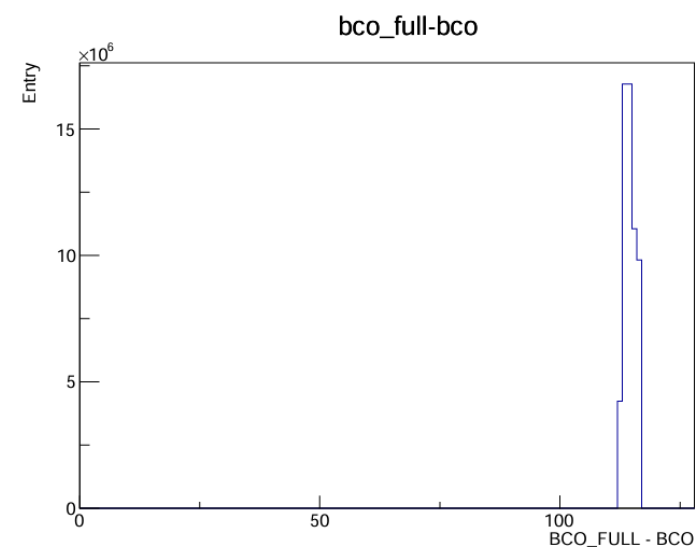
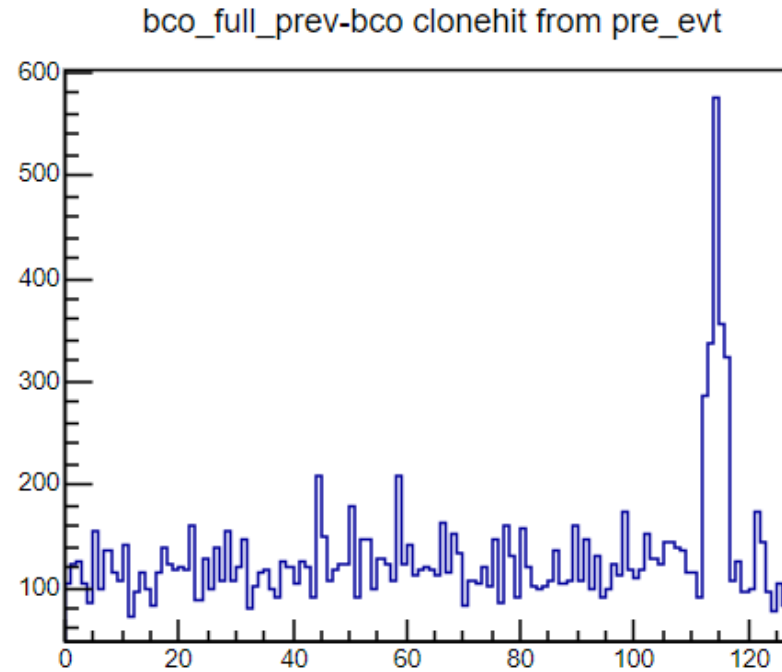
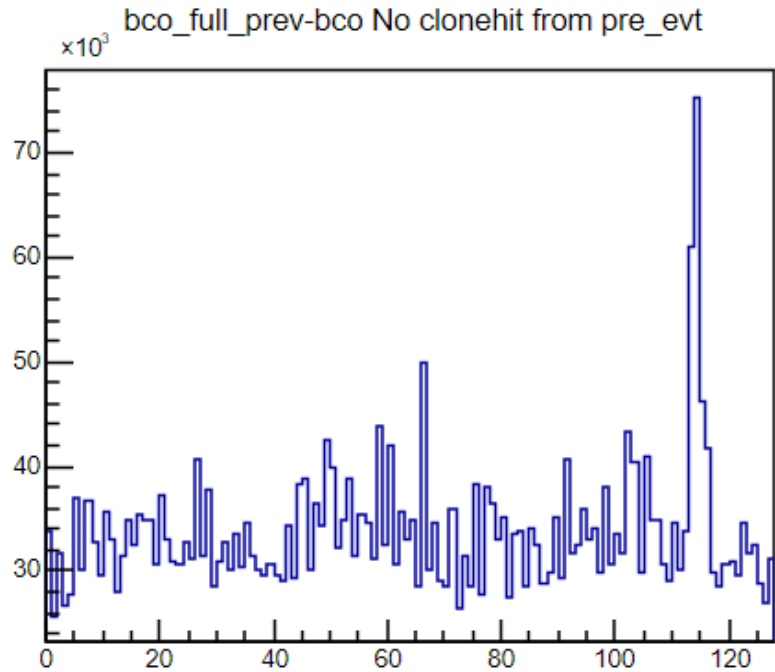
コピーヒットあり



- コピーヒットがあるときも同じ位置にピークが見られる
- Mixuoと1つ前のイベントからのコピーヒットは関係がある

# Run24768 intt1 アップグレード後

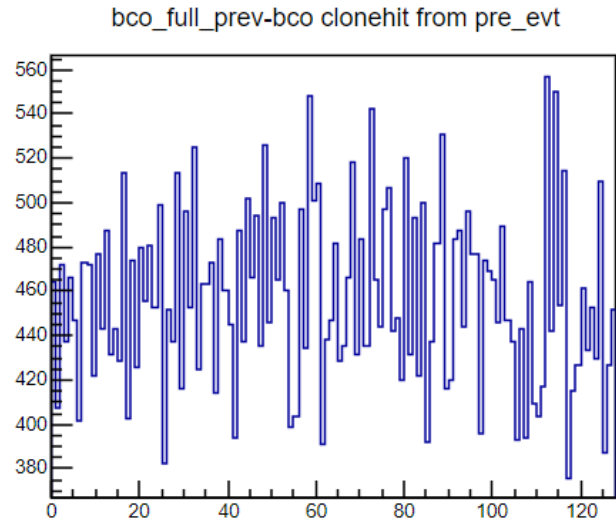
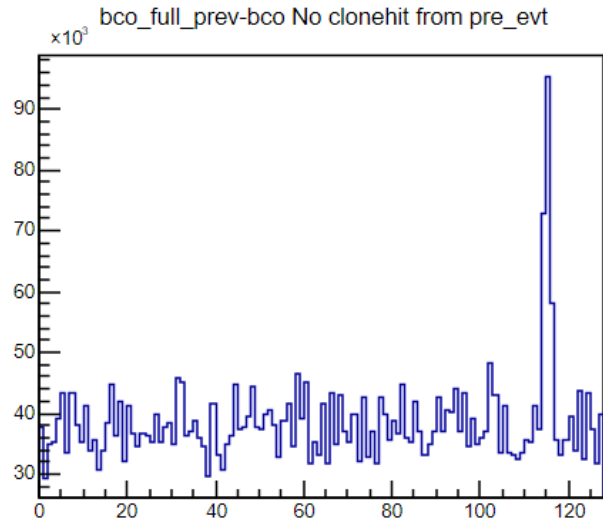
1つ前のイベントのBCO\_FULL - BCO  
コピーヒットなし                      コピーヒットあり



- コピーヒットがあるときも同じ位置にピークが見られる
- Mixuoと1つ前のイベントからのコピーヒットは関係がある

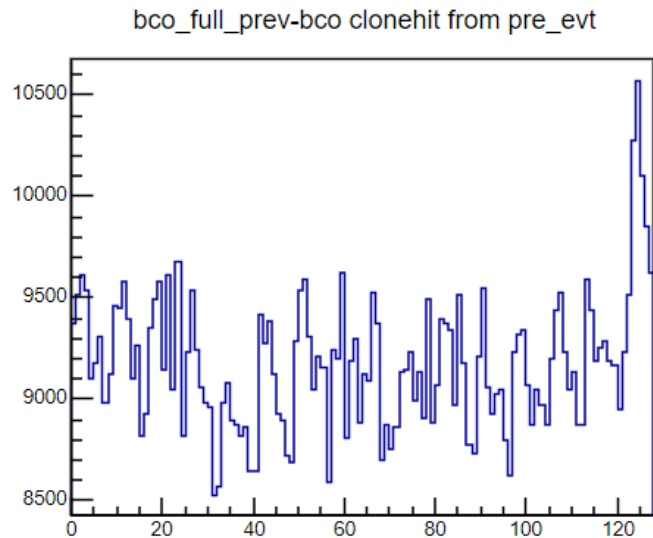
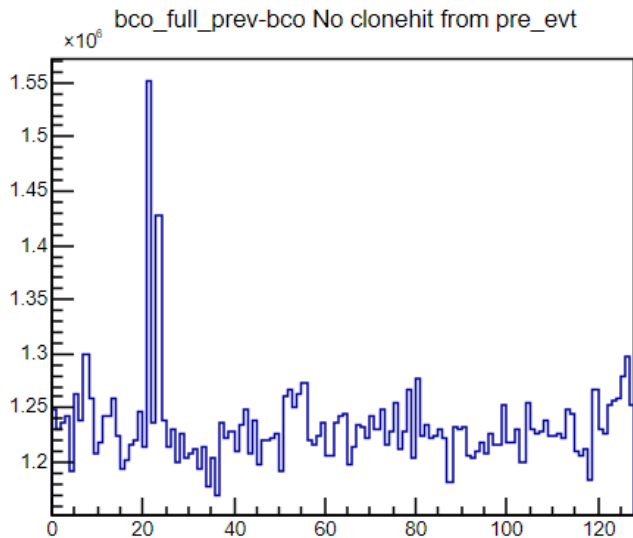


## Run24768 intt6



- Runごと、Felixごとに結果は異なり、ピーク (Mixup) が見られないものやピークが異なる位置に見えるものがある
- 相関など調査中

## Run20869 intt1

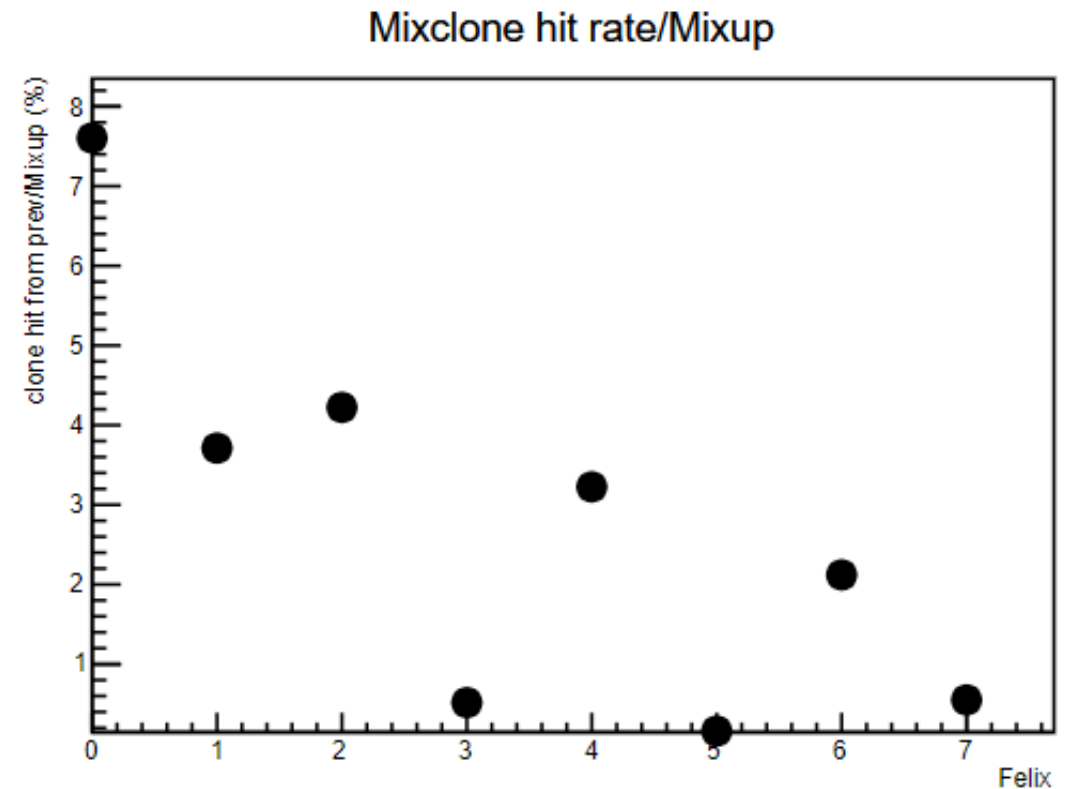
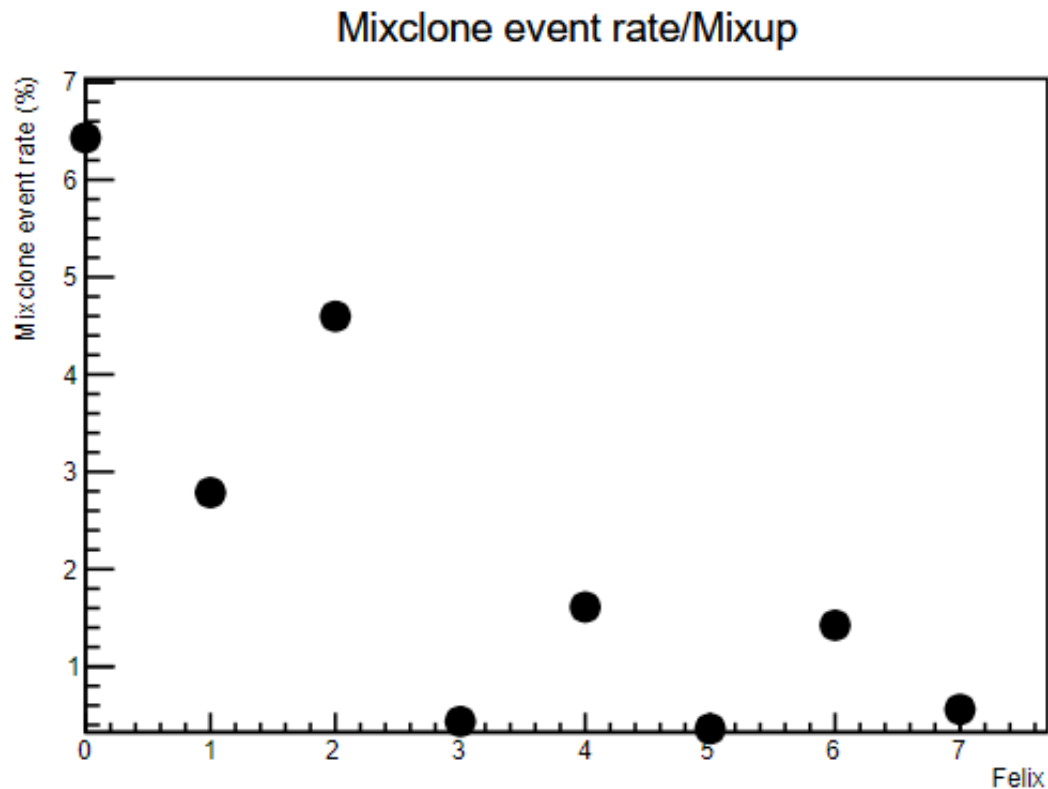


- Mixupヒットのなかでもコピーヒットの割合を調べた

# 1つ前のイベントからコピーヒット割合 20708

左：mixupかつコピーヒットを持つイベント数/全イベント数 \*100

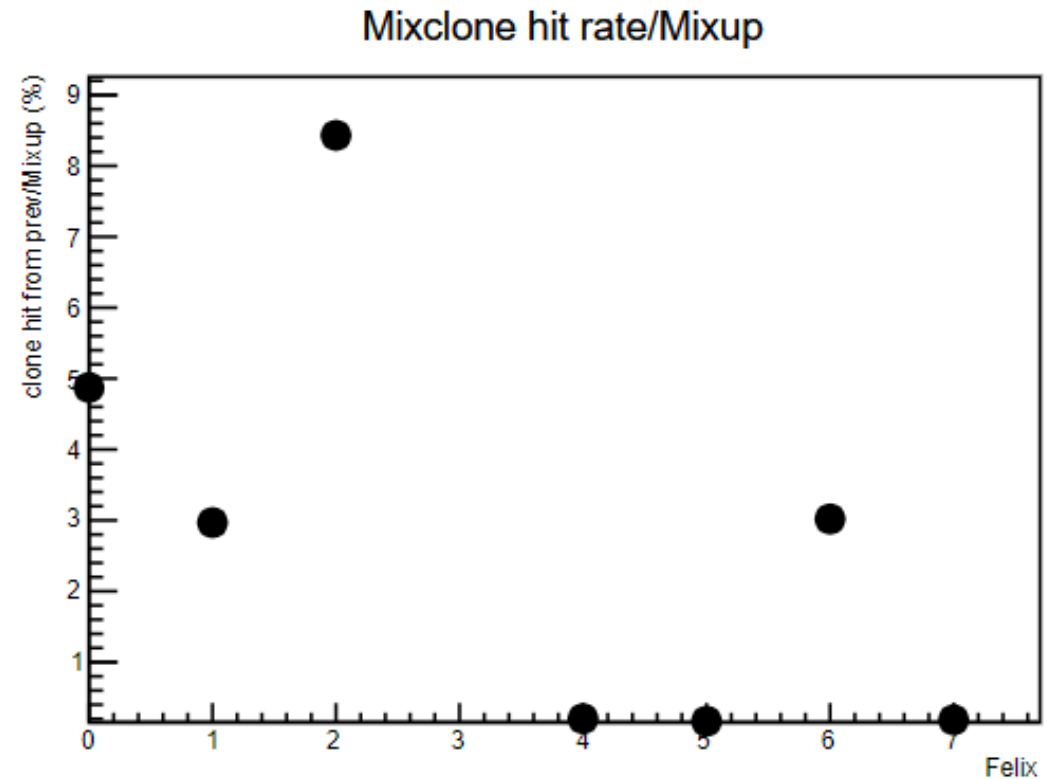
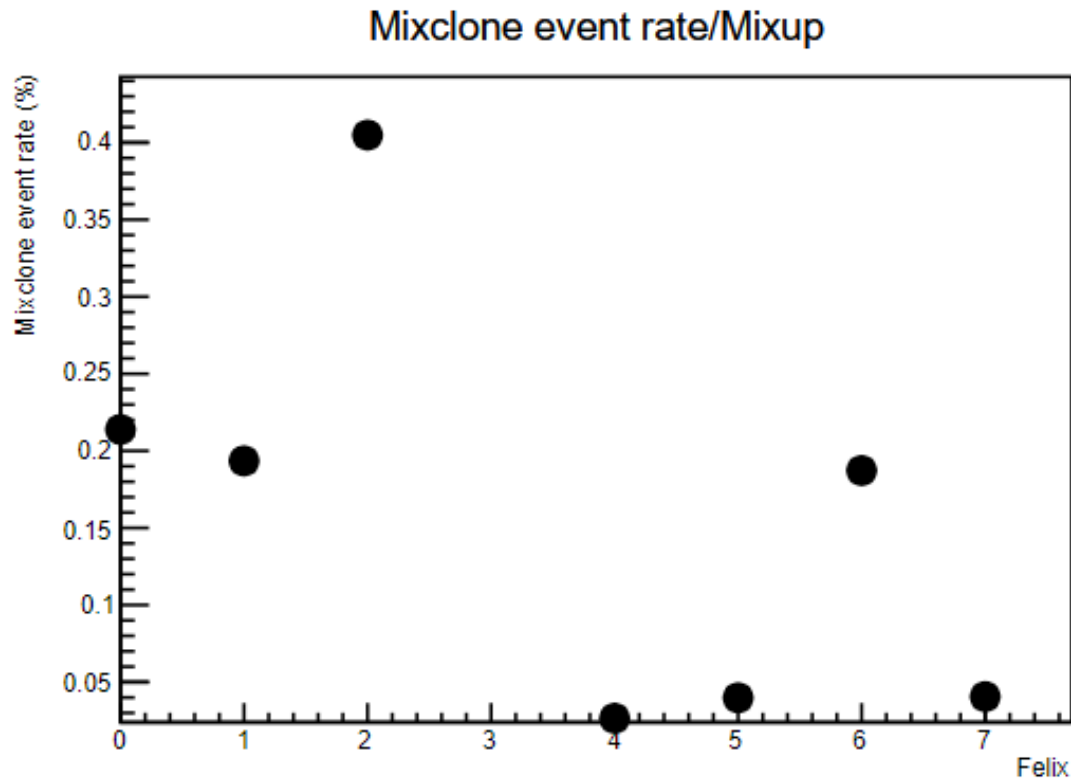
右：各イベント内のmixupかつコピーヒットのヒット数/mixupのヒット数 \*100 をmixupのあるイベント数で平均をとったもの（←間違い？）



# 1つ前のイベントからコピーヒット割合 24768

左：mixupかつコピーヒットを持つイベント数/全イベント数 \*100

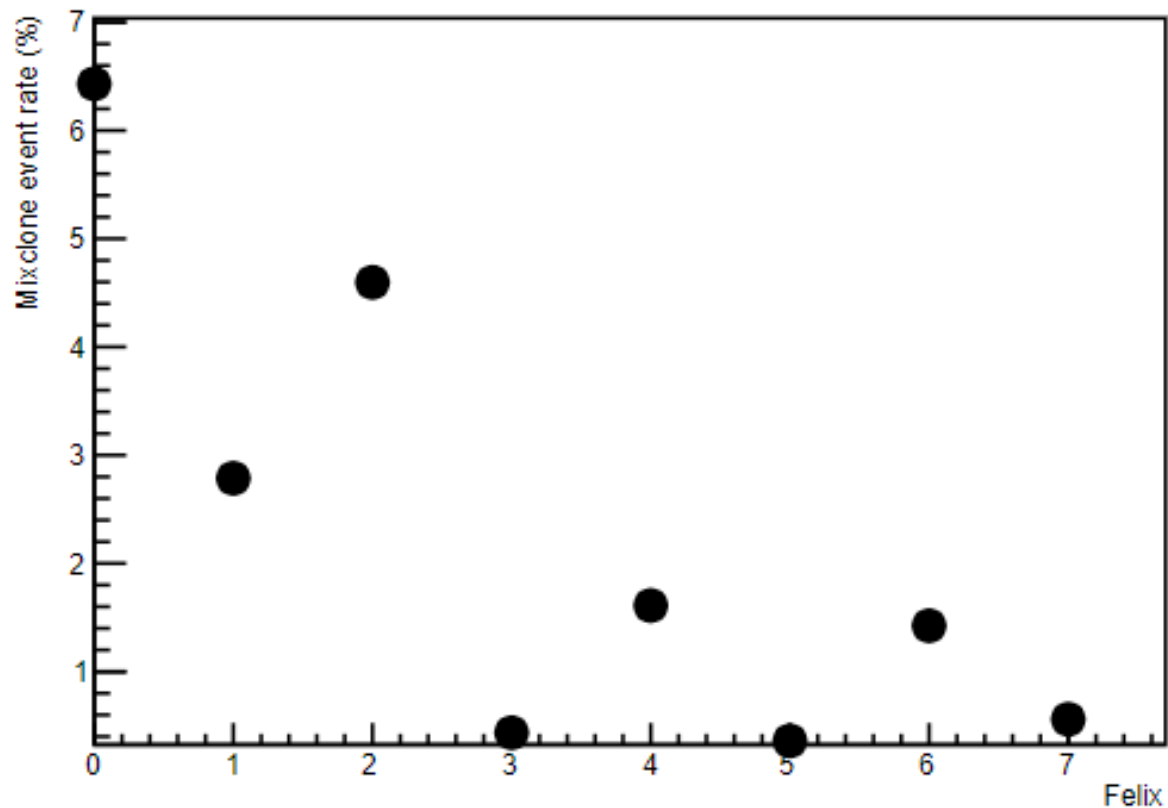
右：各イベント内のmixupかつコピーヒットのヒット数/mixupのヒット数 \*100 をmixupのあるイベント数で平均をとったもの（←間違い？）



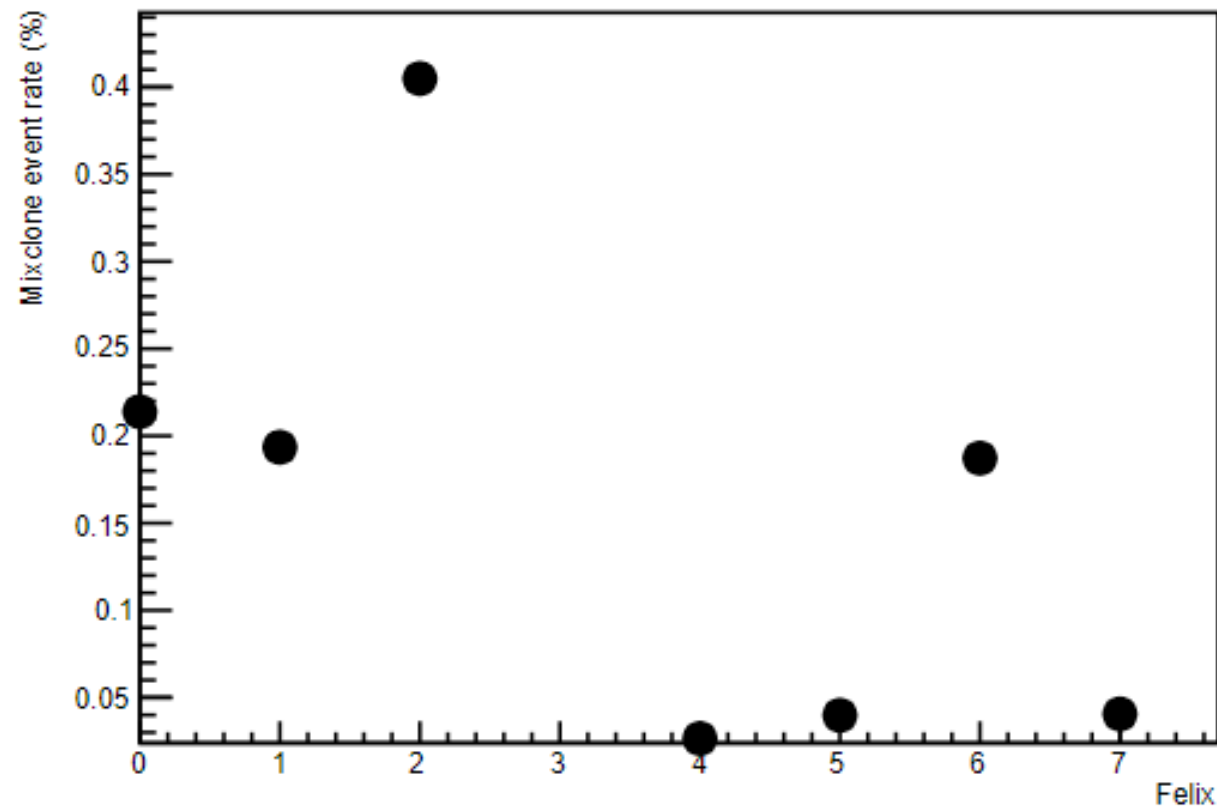
# 1つ前のイベントからコピーヒット割合

mixupかつコピーヒットを持つイベント数/全イベント数 \*100  
においてアップグレード前後で変わった (多→少)

Mixclone event rate/Mixup



Mixclone event rate/Mixup

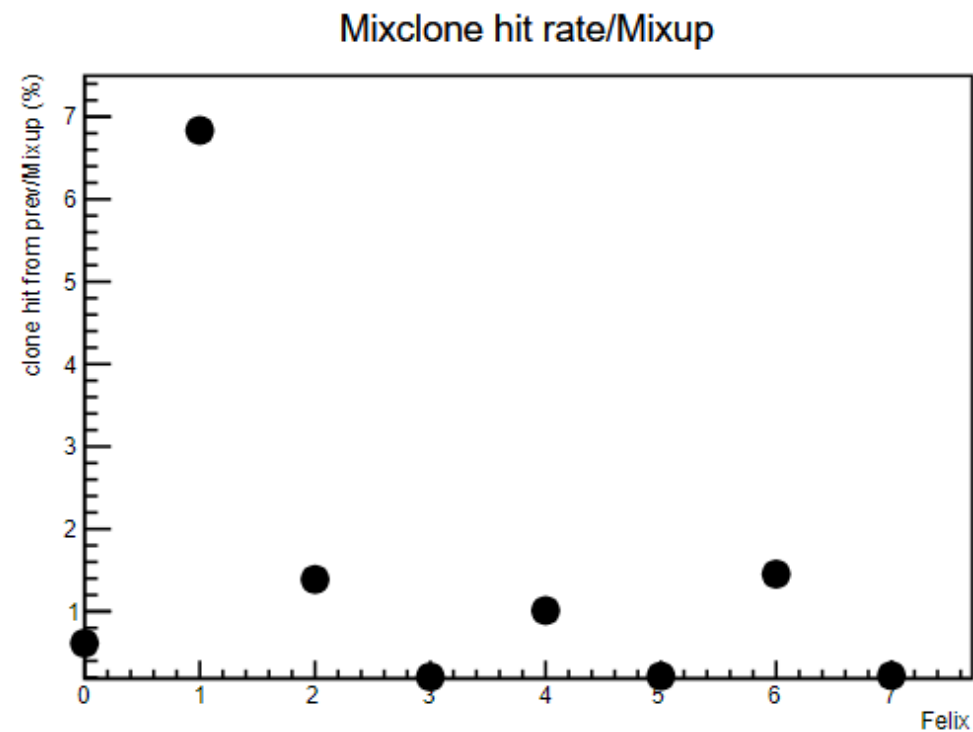
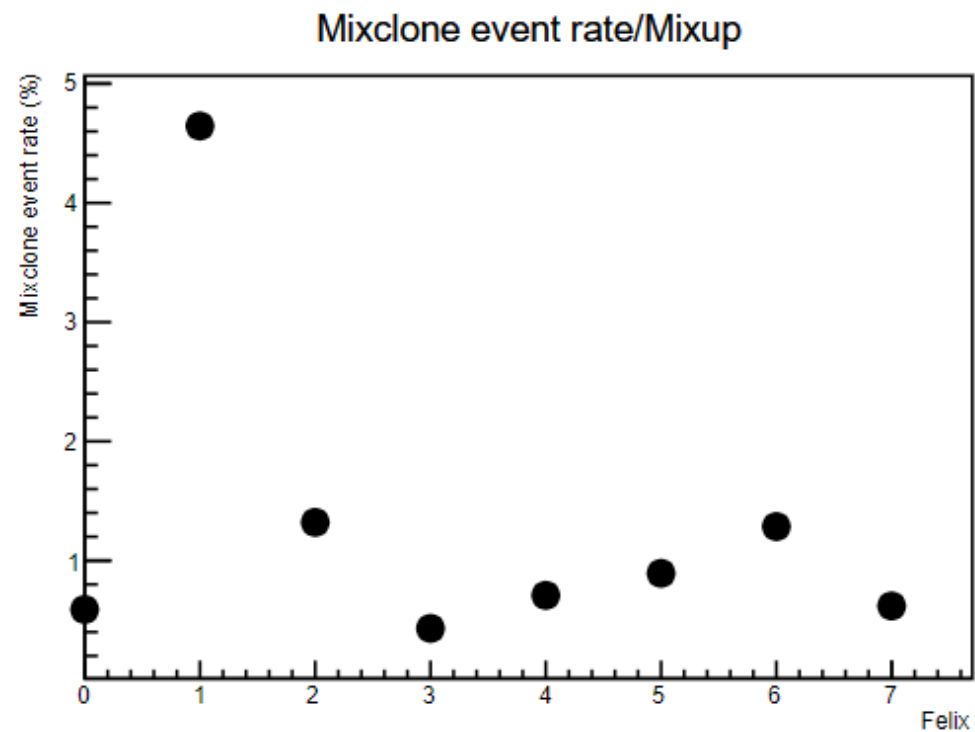


## まとめと今後の課題

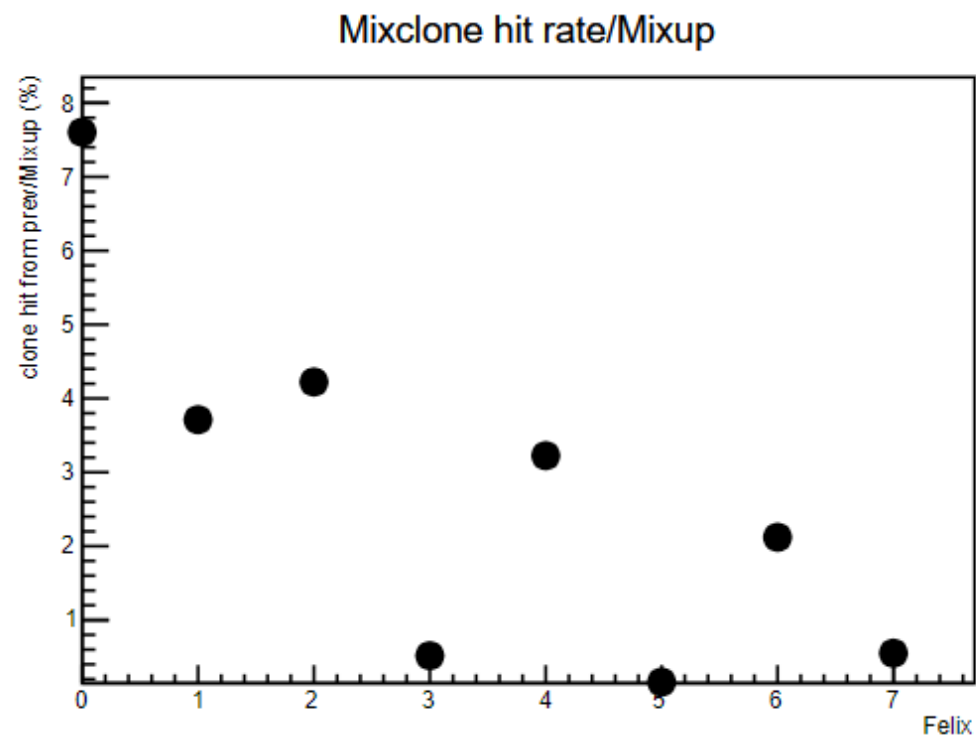
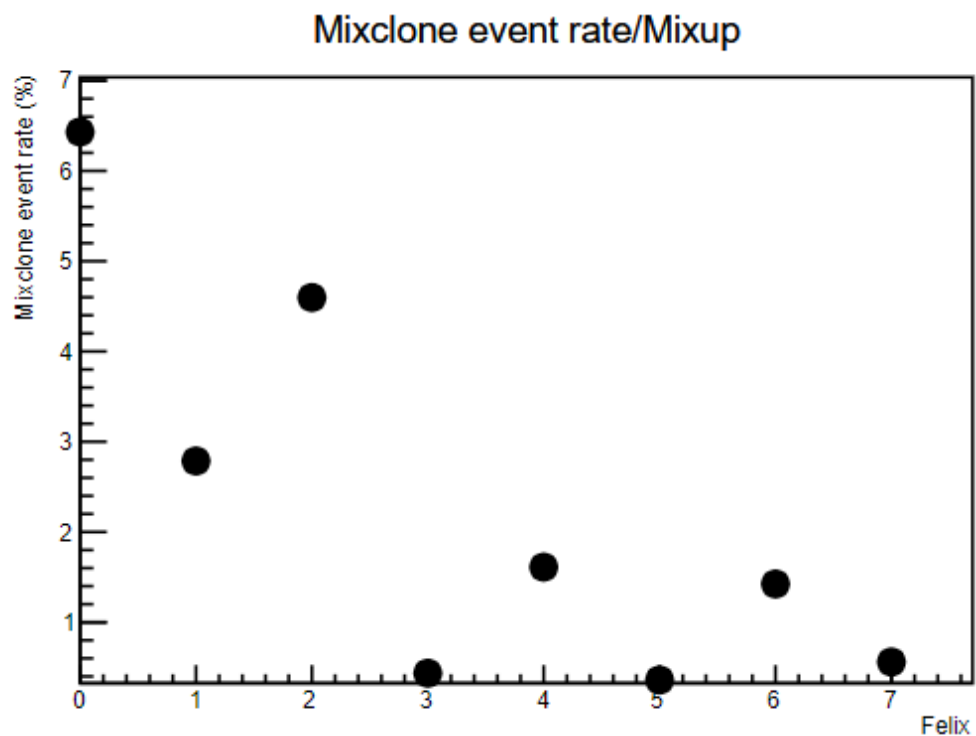
- 2つ前のイベントからのヒットのリークは無いように見える
- アップグレード前後でMixupが起こったイベントの割合減
- 1イベント内のMixupのヒットの割合は平均1~5%
- 1つ前のイベントからのコピーヒットと関係があるかもしれないが Mixup全てがコピーではない
  
- Run24のデータでも調べる (DSTを使えるように)
- Hot Channelをカットして調べる (DSTを使えるようになってから?)

Back up

# 1つ前のイベントからコピーヒット割合 20444

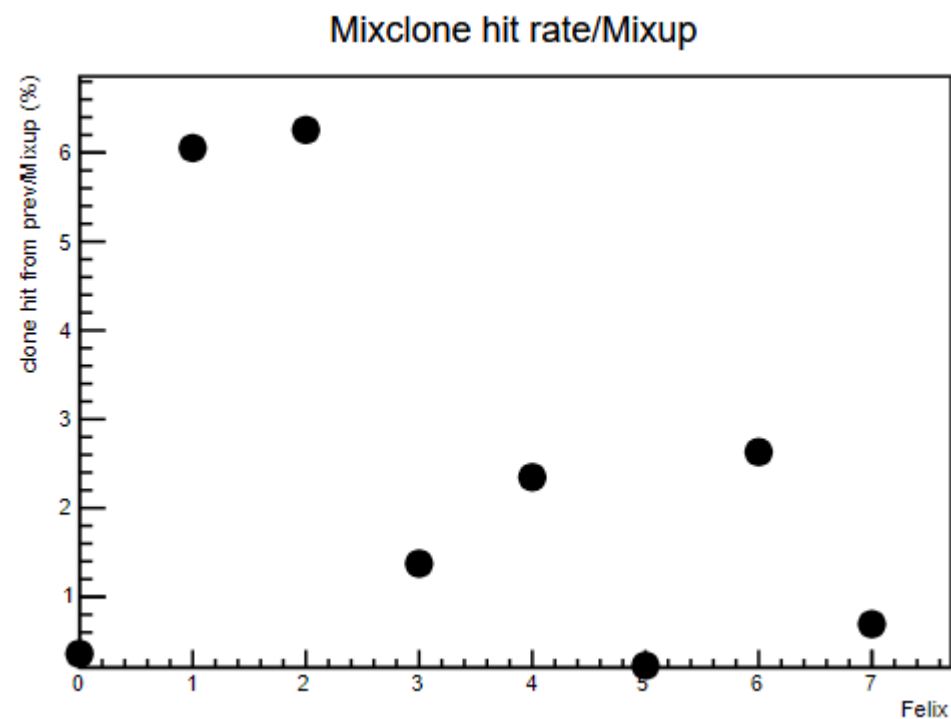
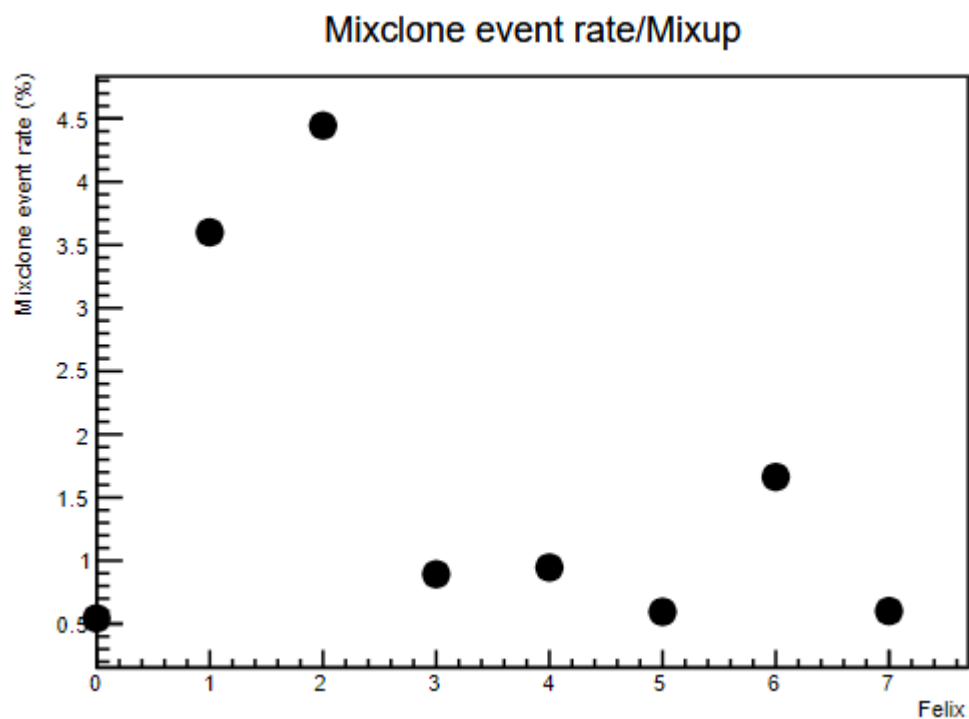


# 1つ前のイベントからコピーヒット割合 20708

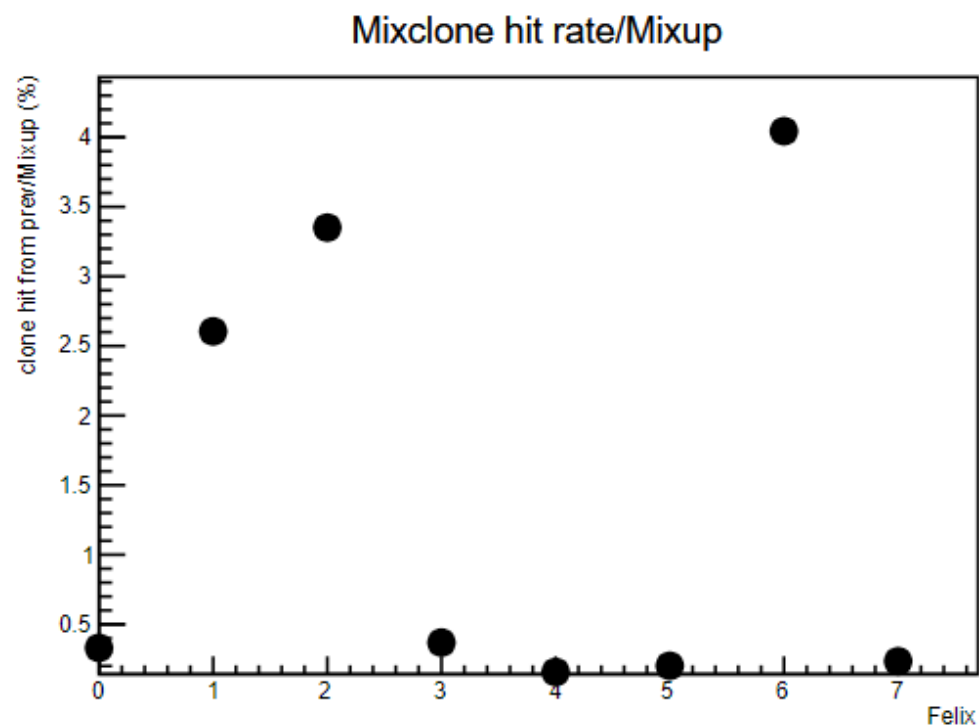
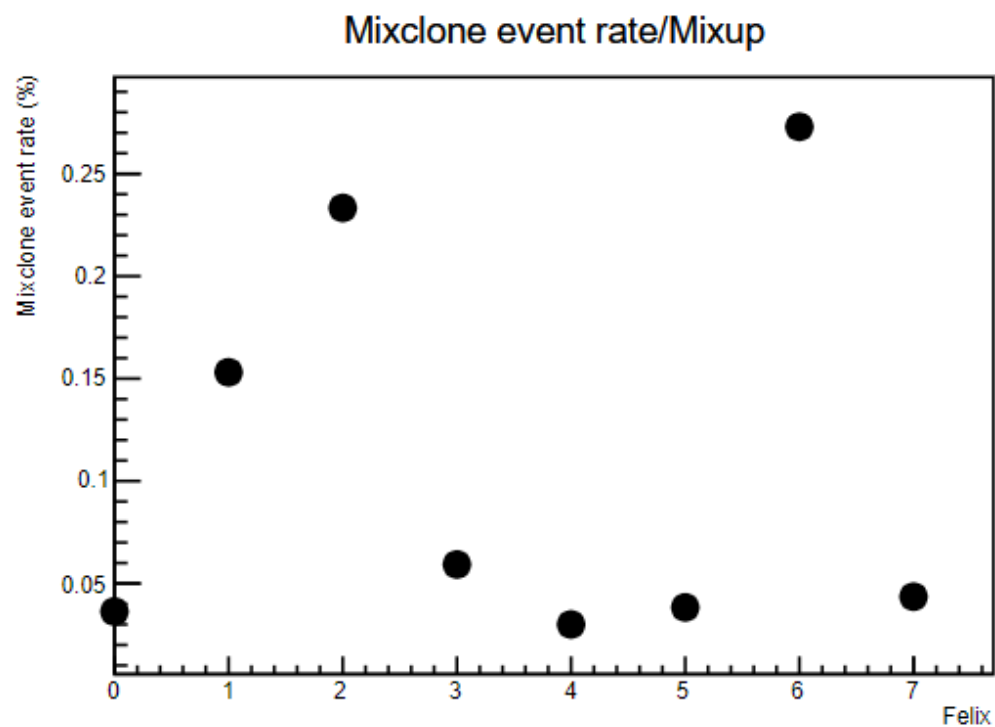




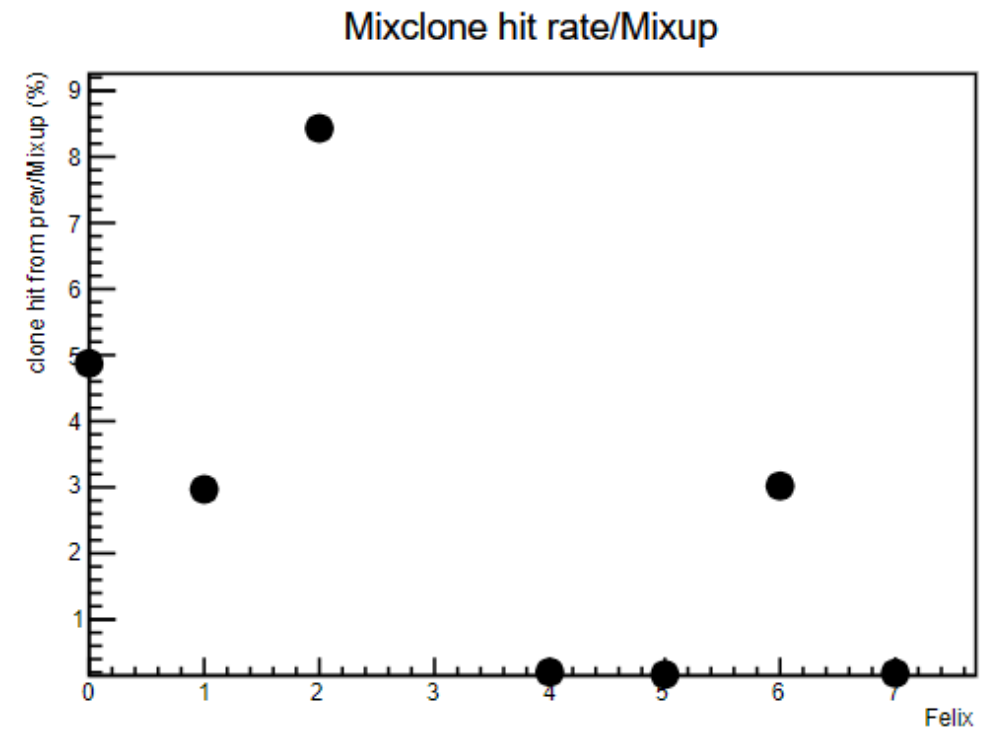
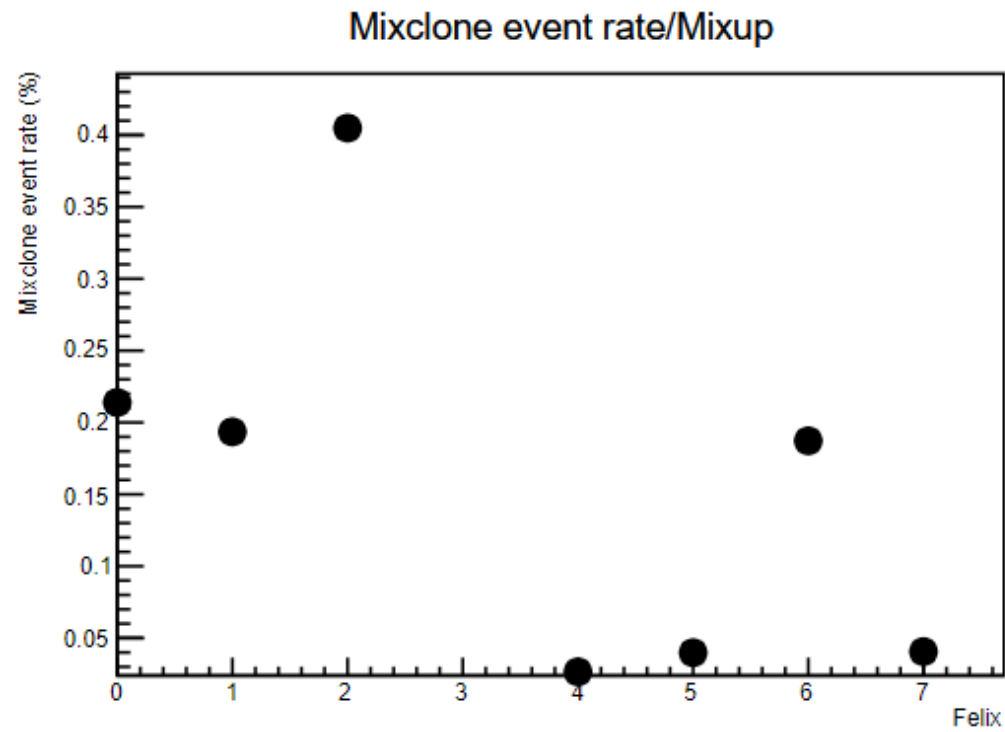
# 1つ前のイベントからコピーヒット割合 20869



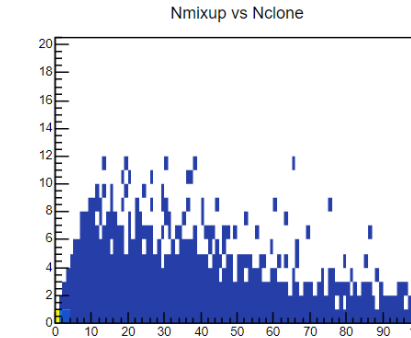
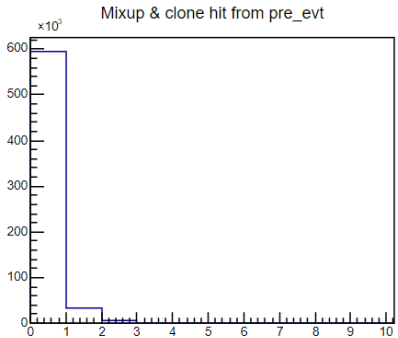
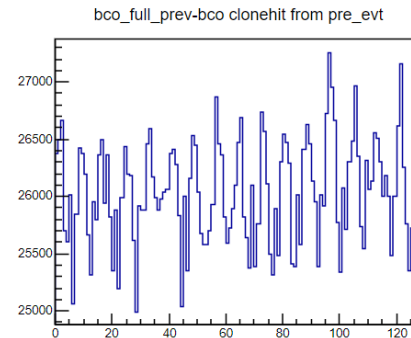
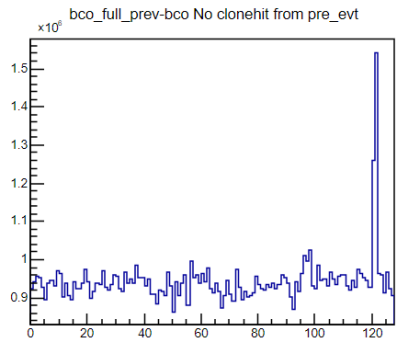
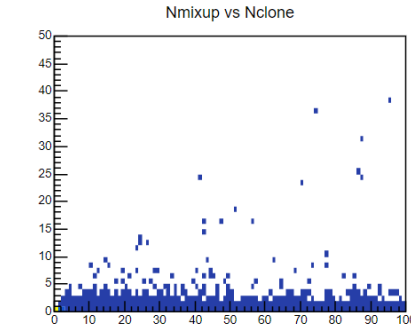
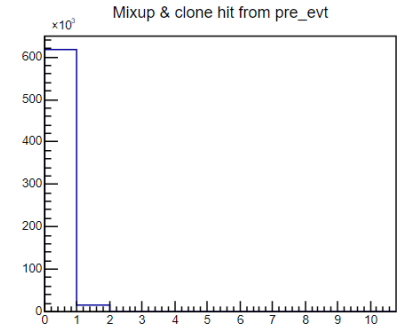
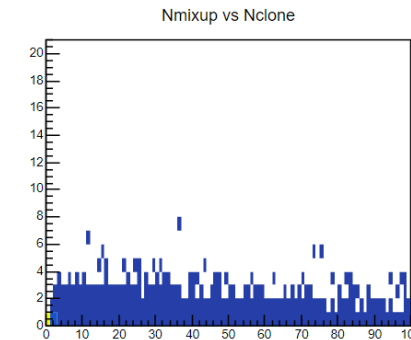
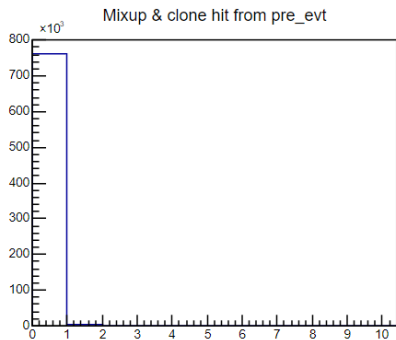
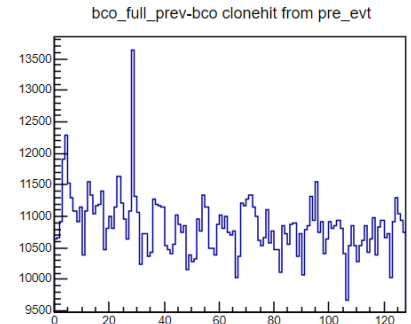
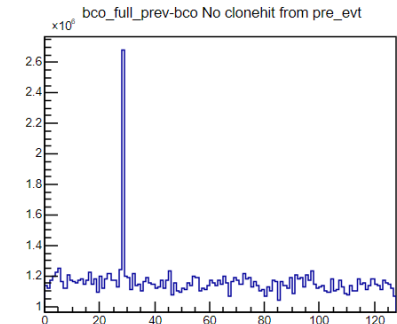
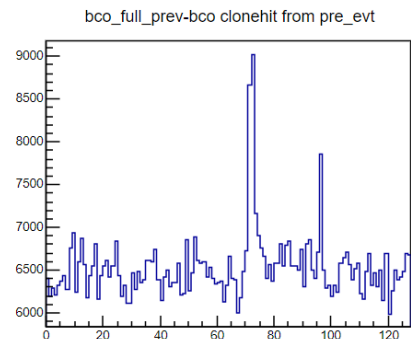
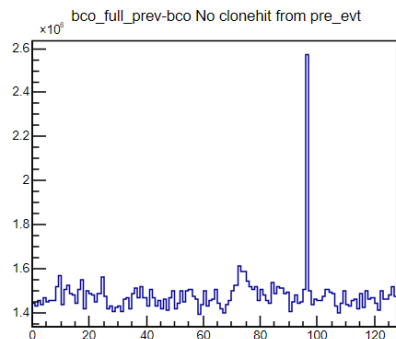
# 1つ前のイベントからコピーヒット割合 23947



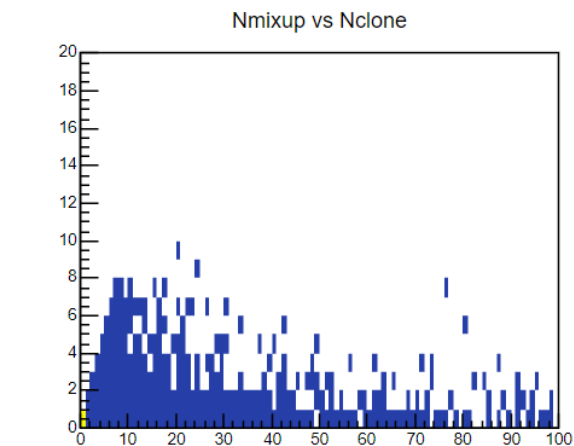
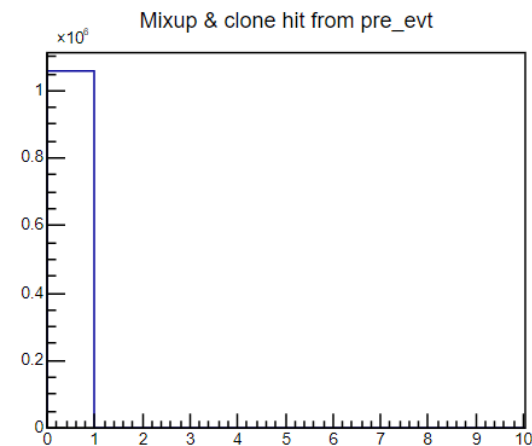
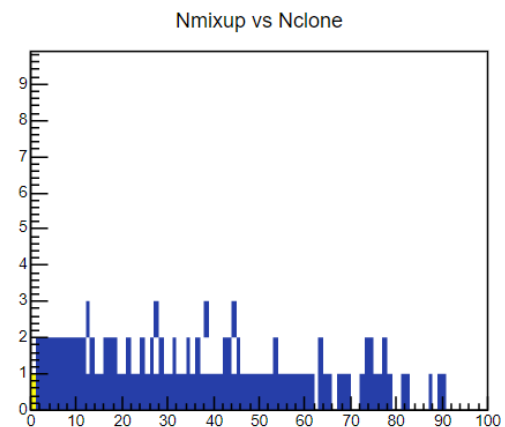
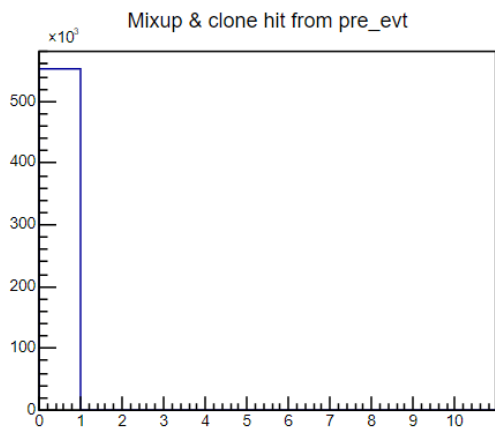
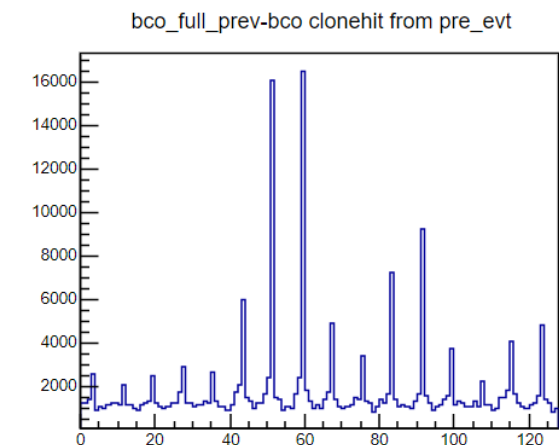
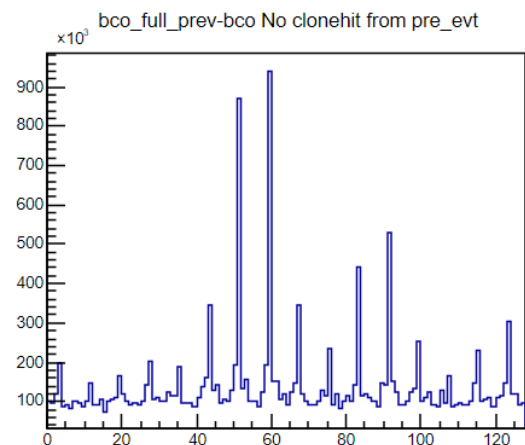
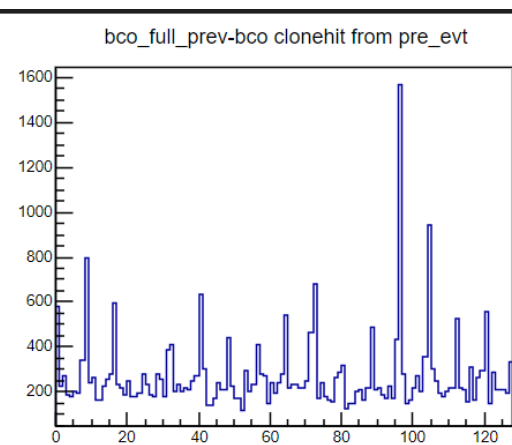
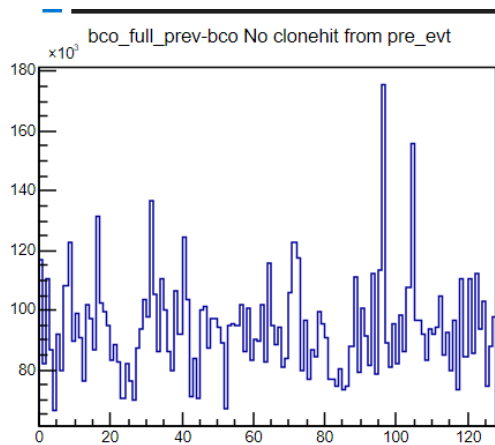
# 1つ前のイベントからコピーヒット割合 24768



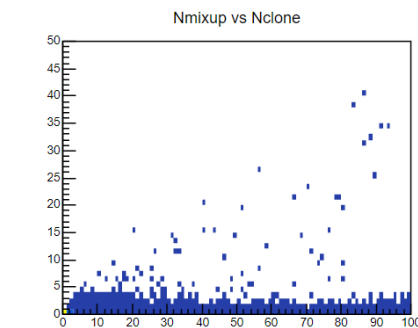
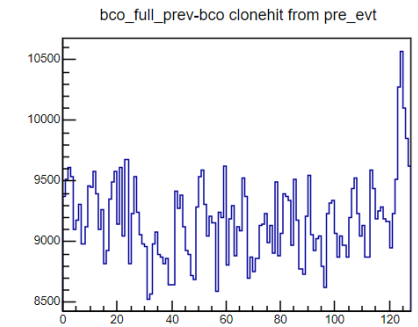
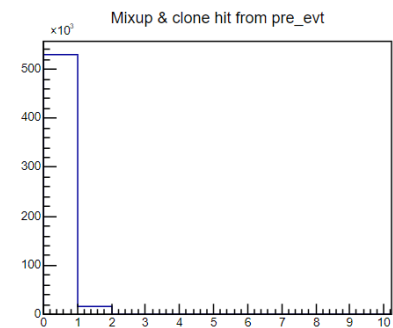
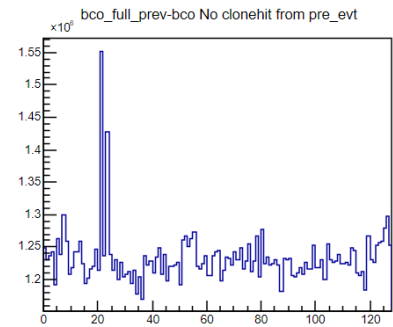
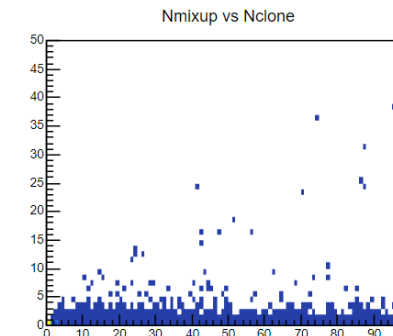
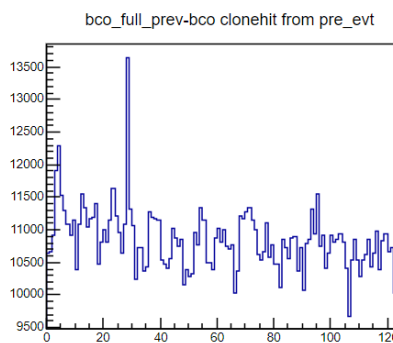
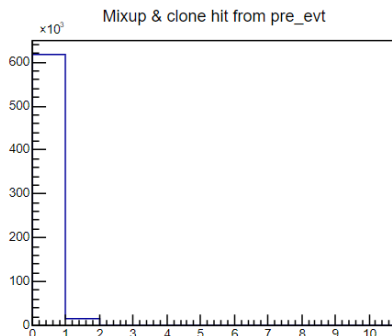
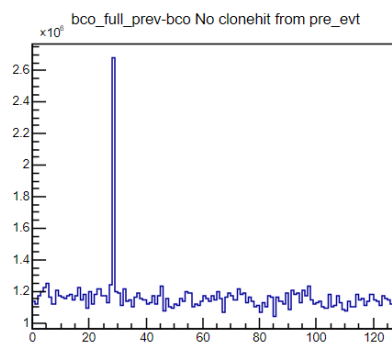
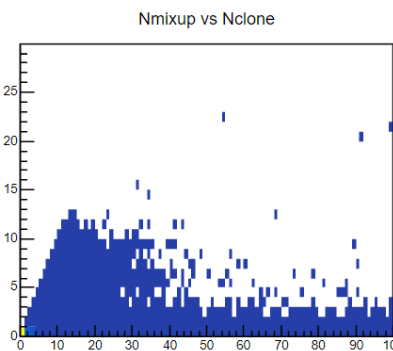
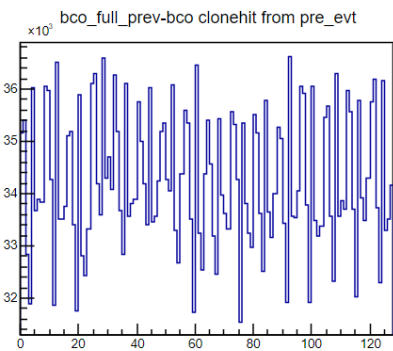
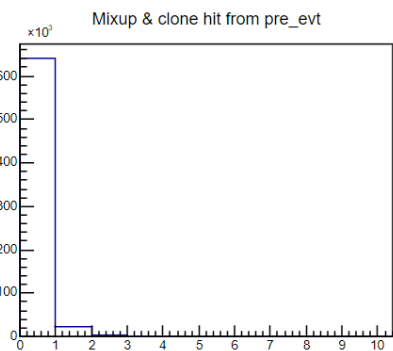
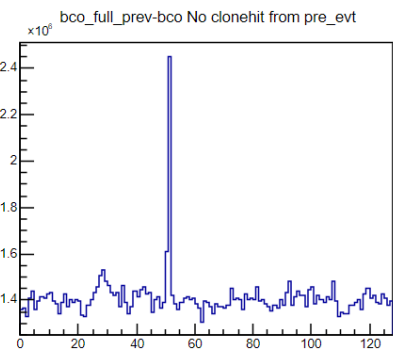
20444 20708 20869 intt0



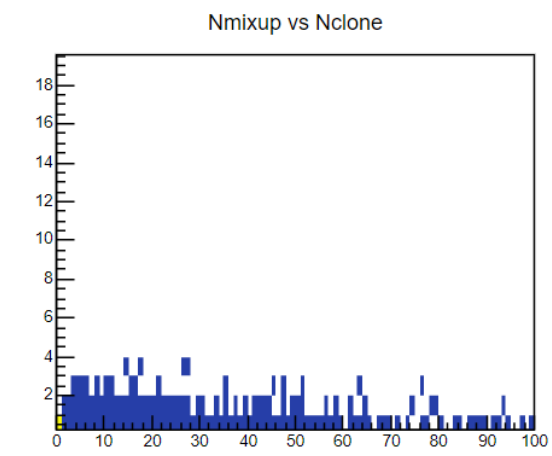
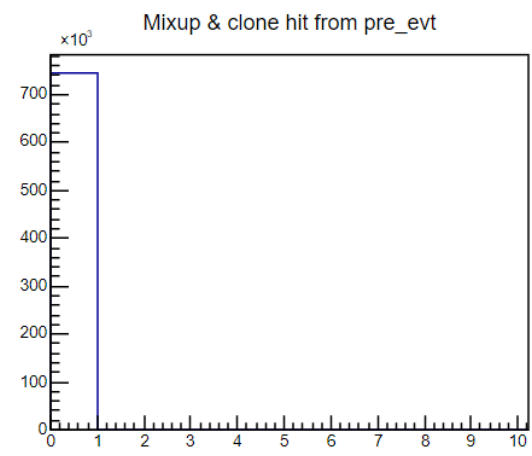
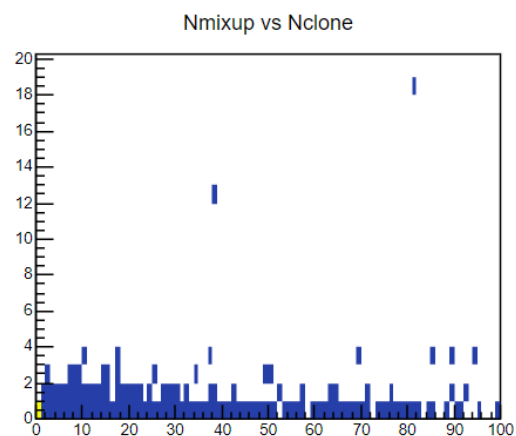
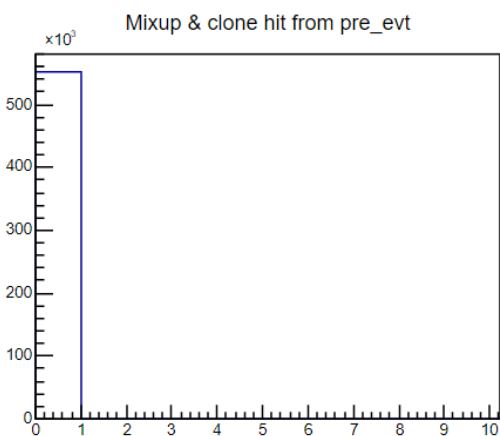
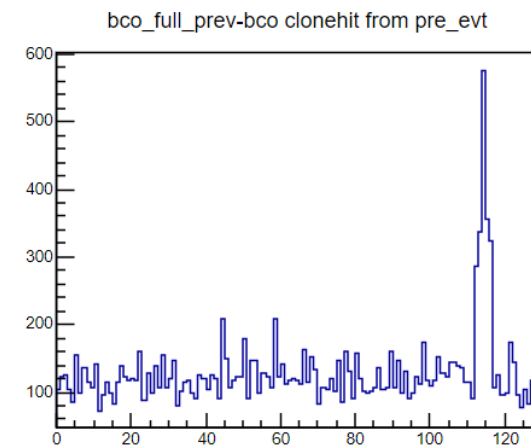
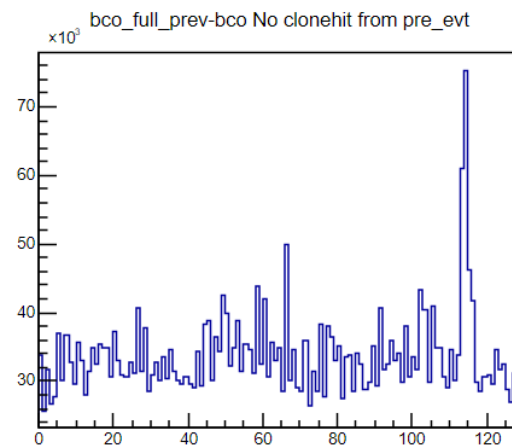
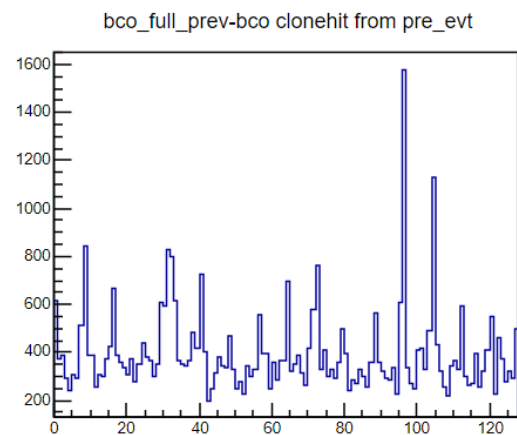
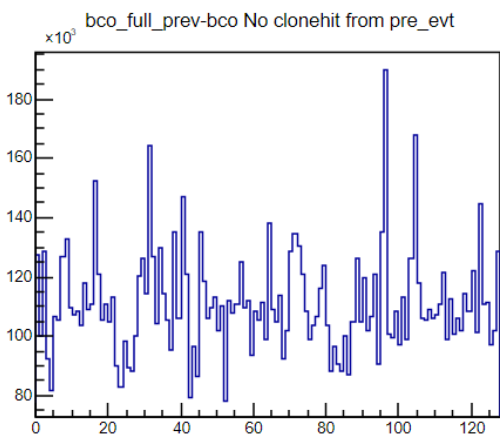
# 23947 24768 intt0



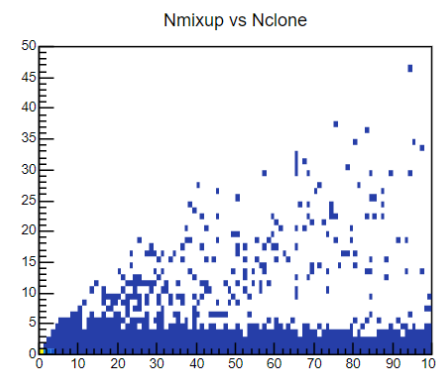
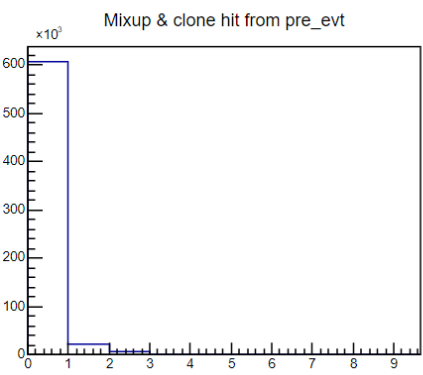
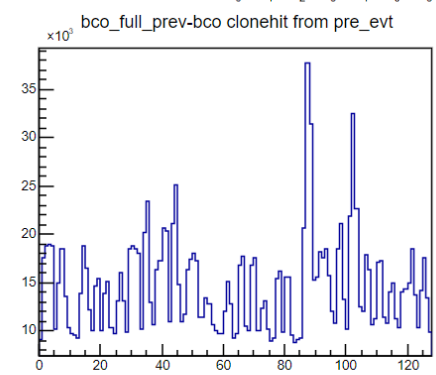
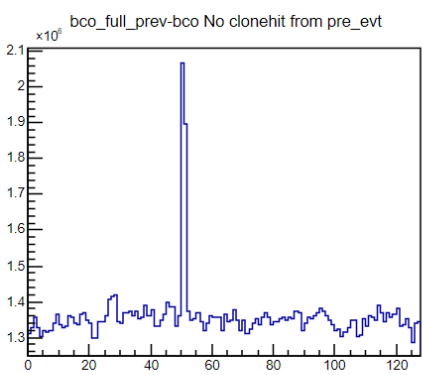
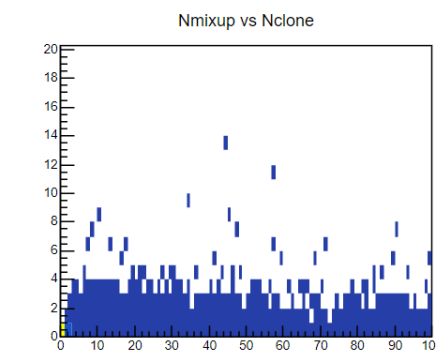
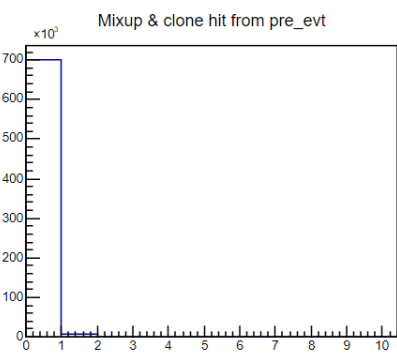
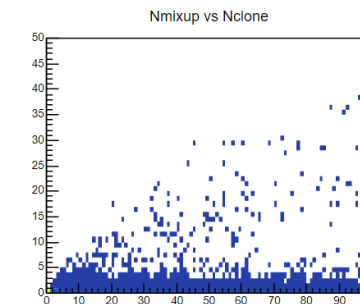
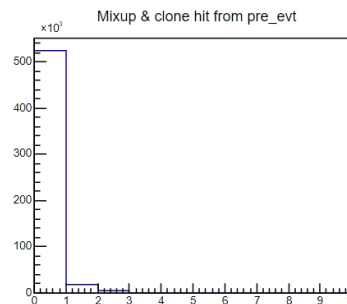
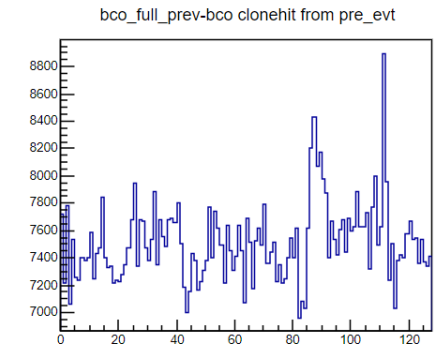
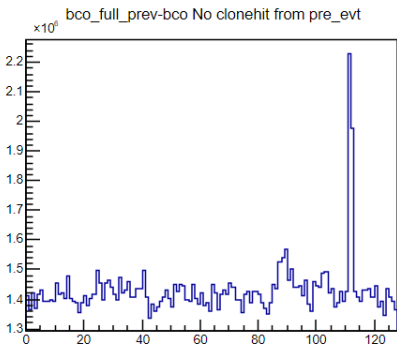
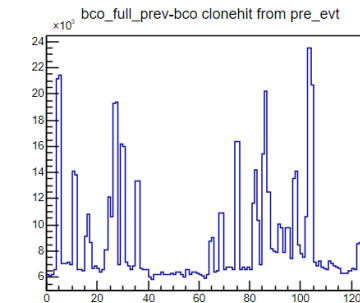
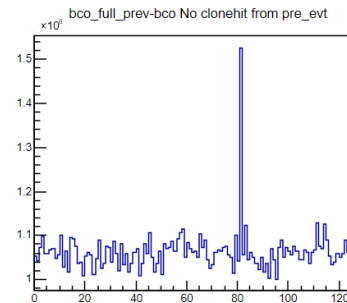
# 20444 20708 20869 intt1



# 23947 24768 intt1

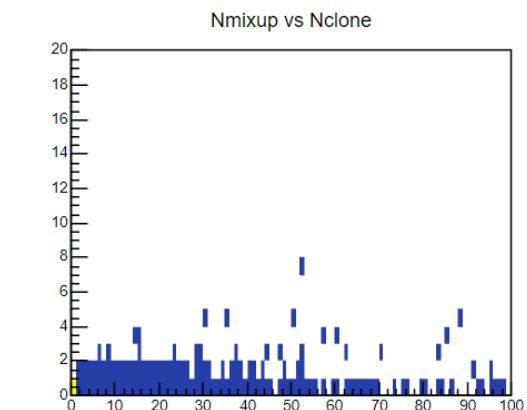
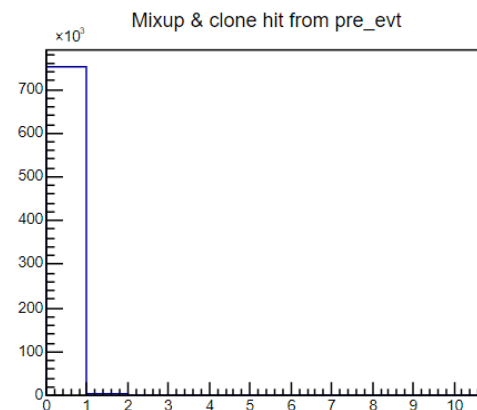
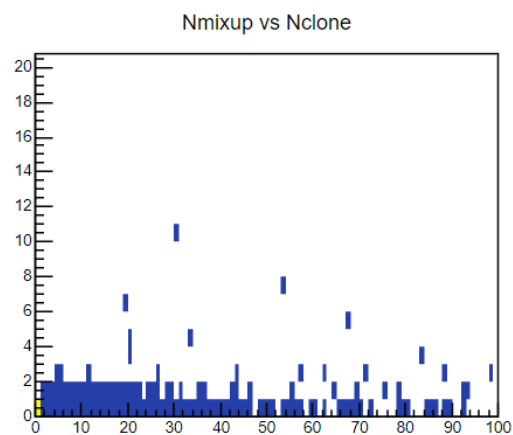
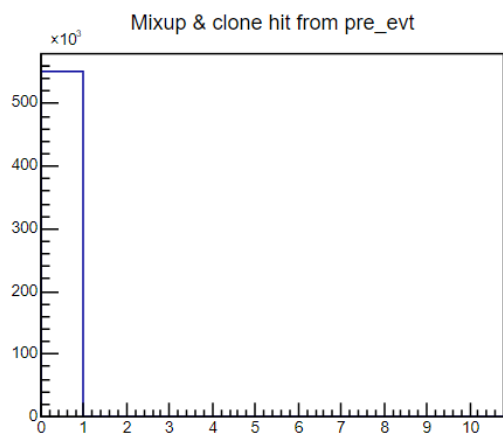
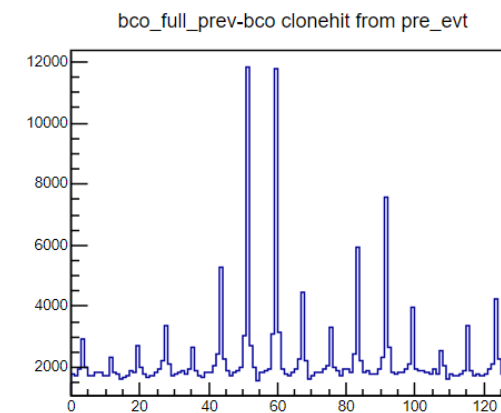
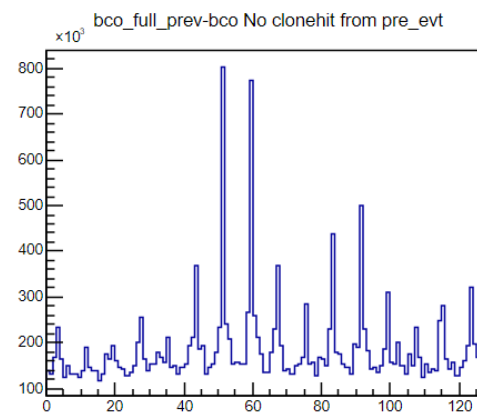
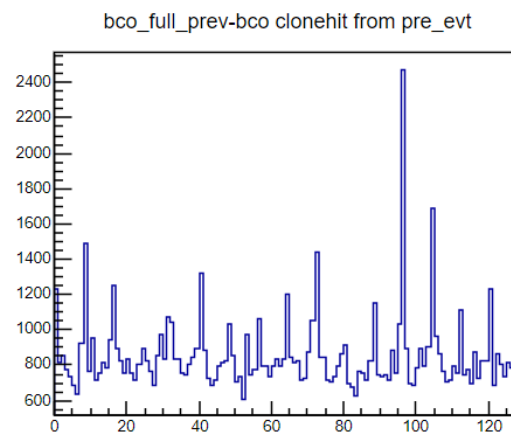
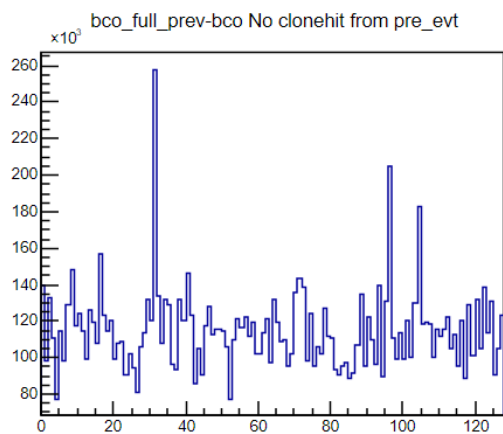


# 20444 20708 20869 intt2

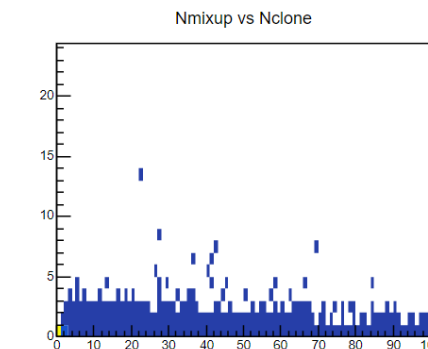
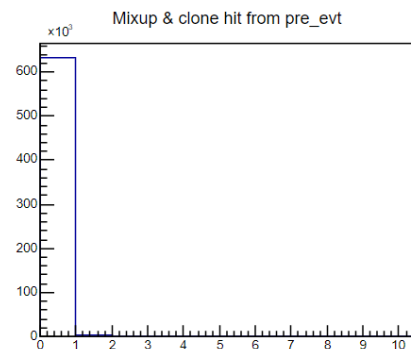
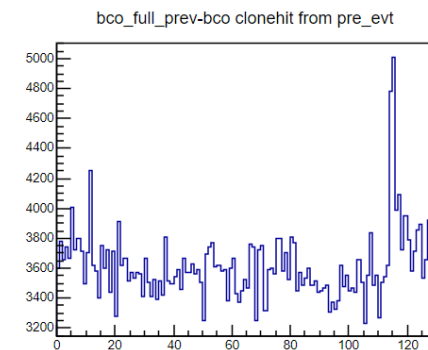
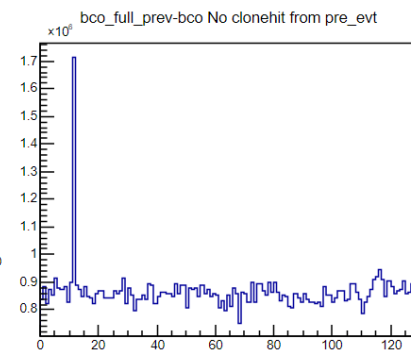
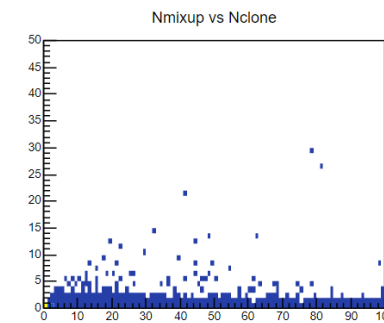
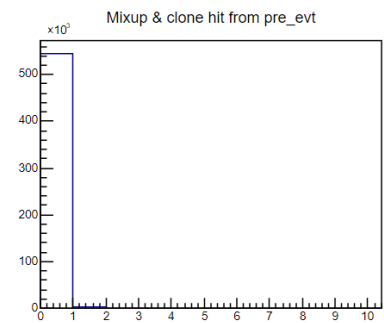
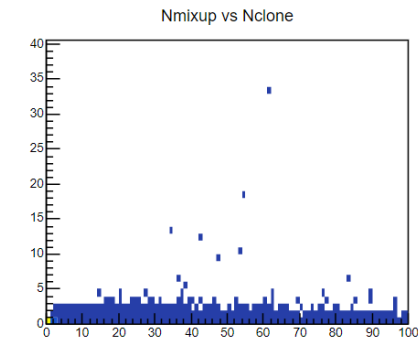
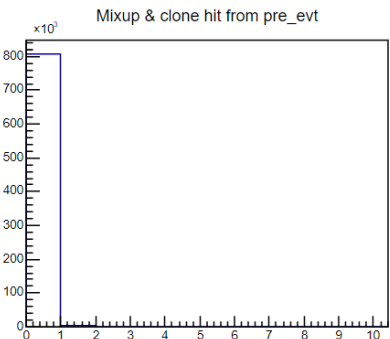
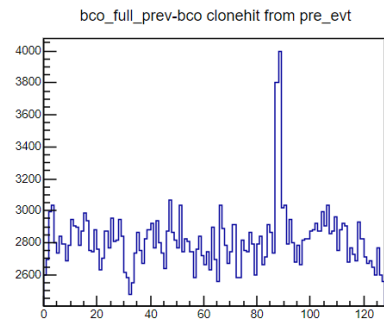
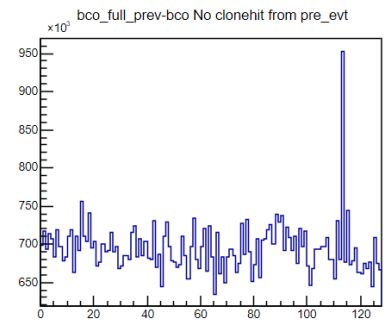
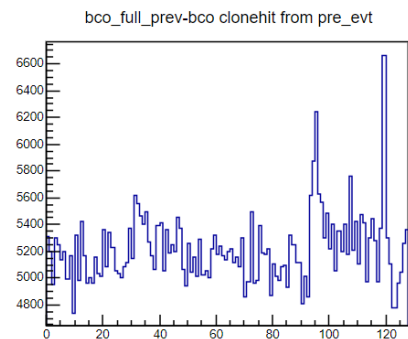
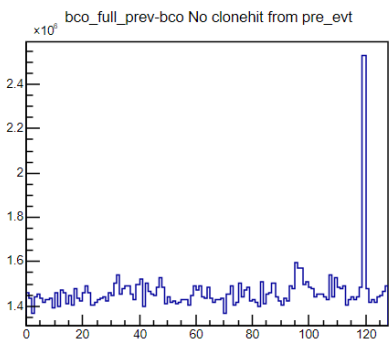




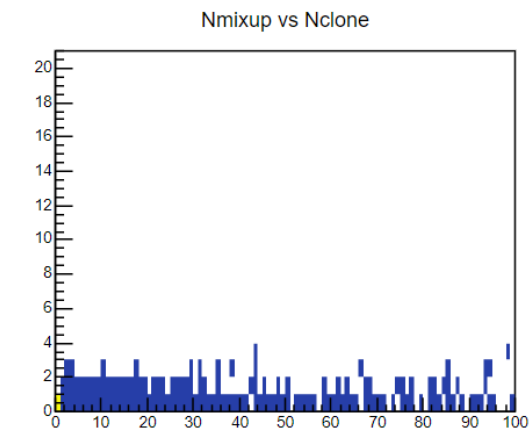
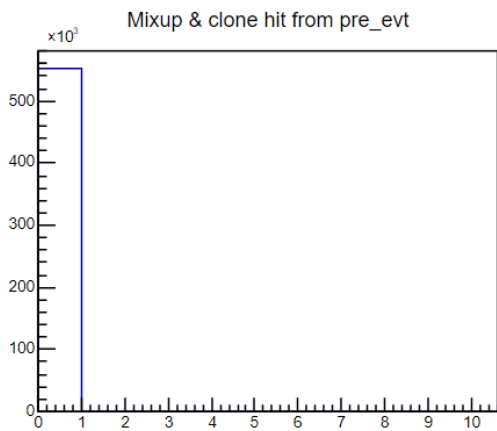
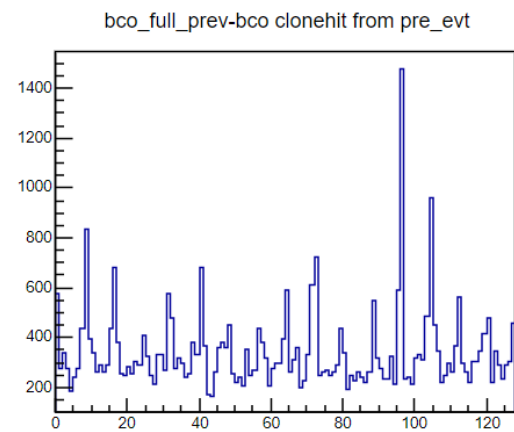
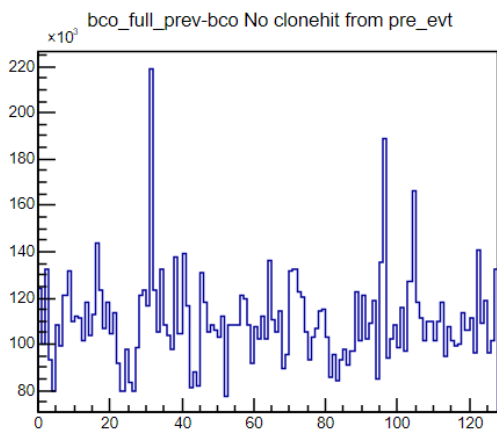
# 23947 24768 intt2



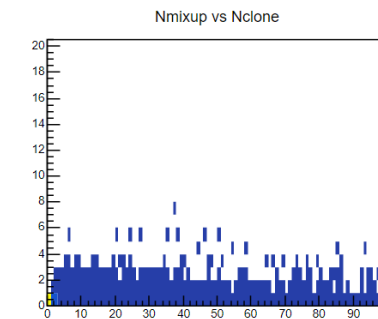
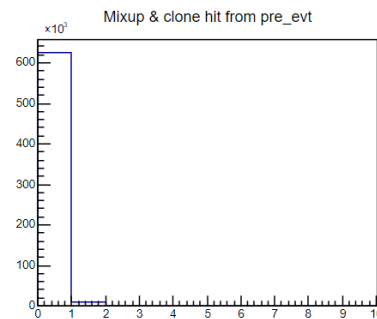
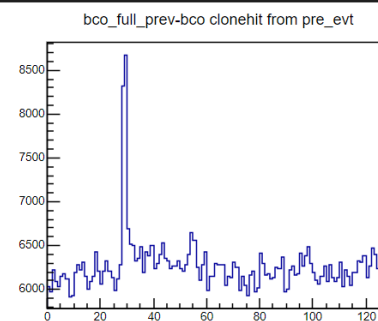
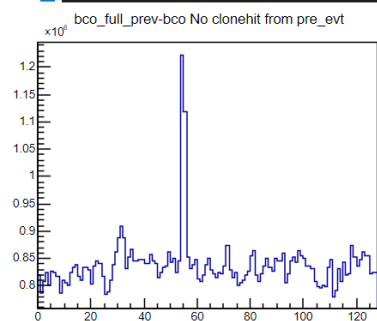
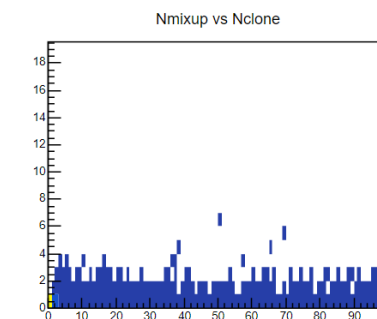
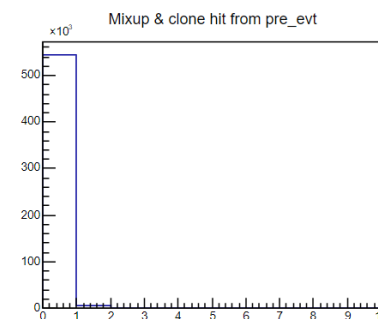
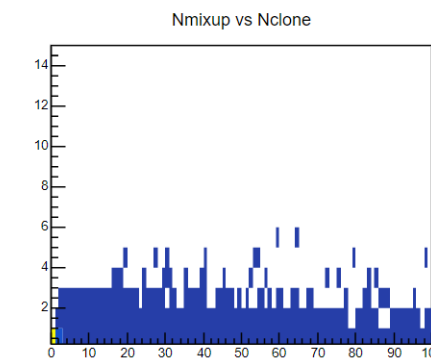
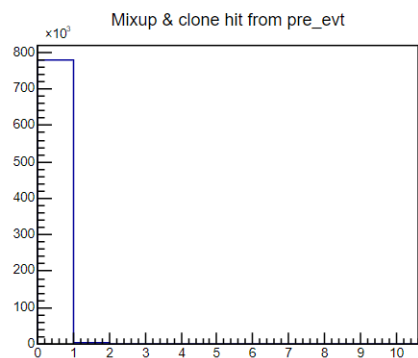
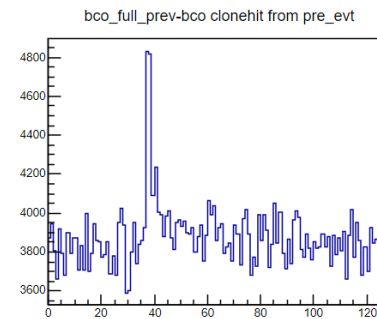
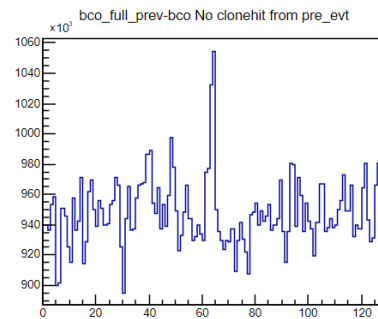
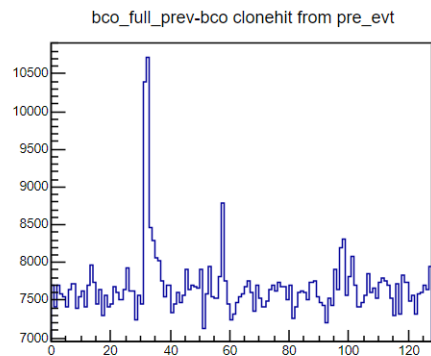
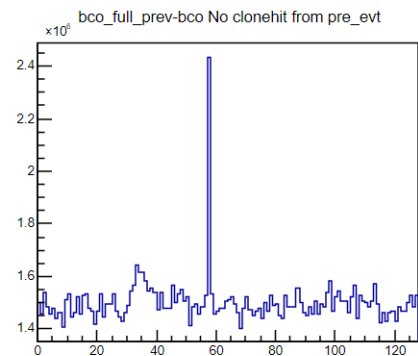
# 20444 20708 20869 intt3



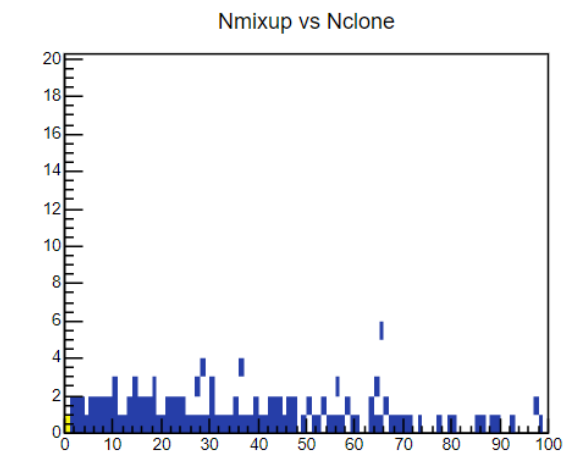
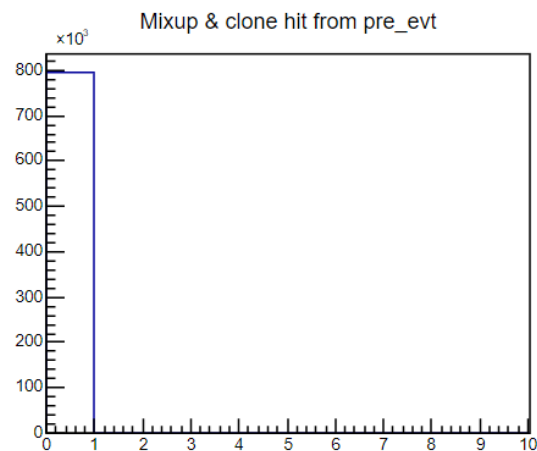
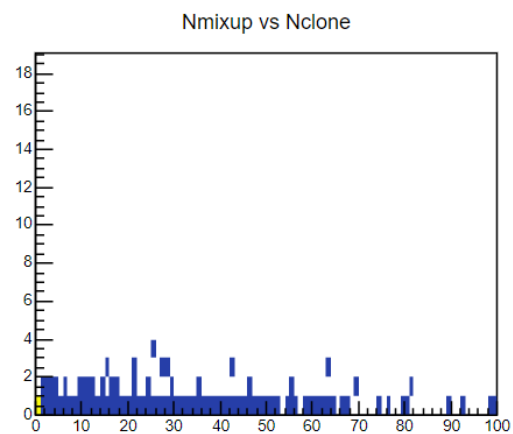
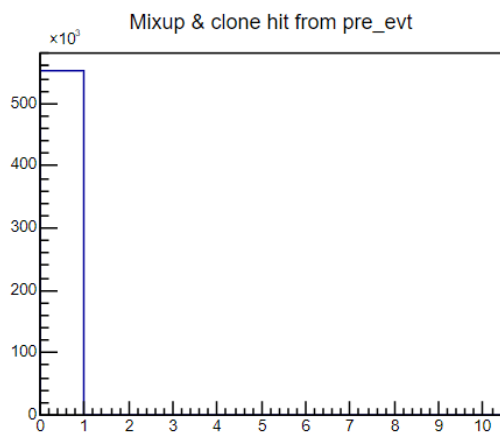
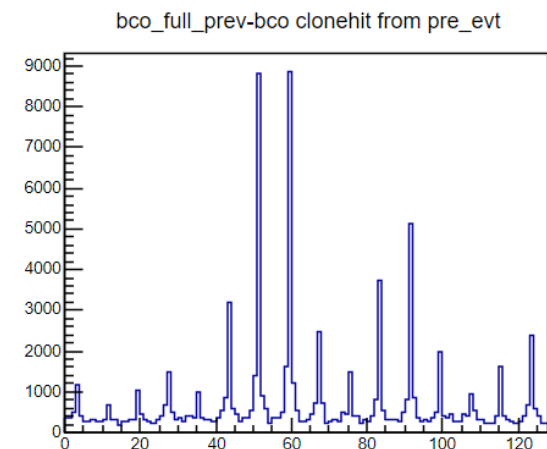
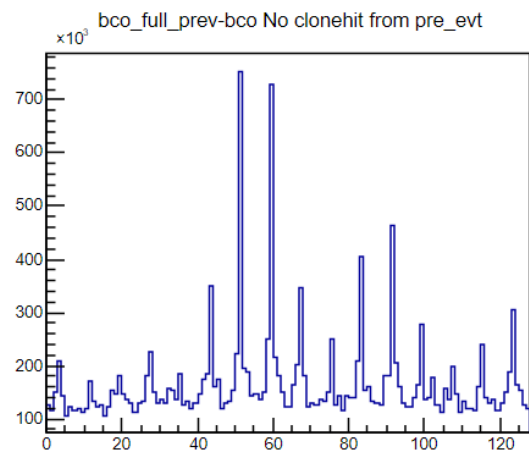
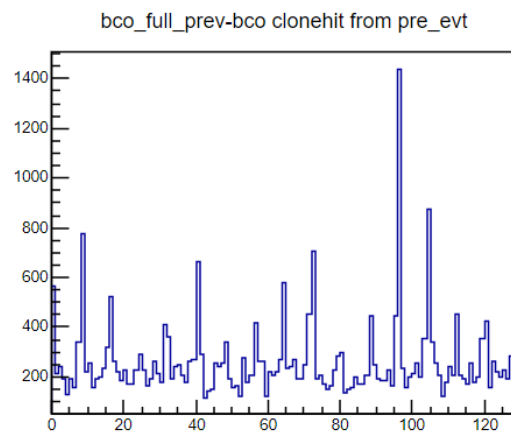
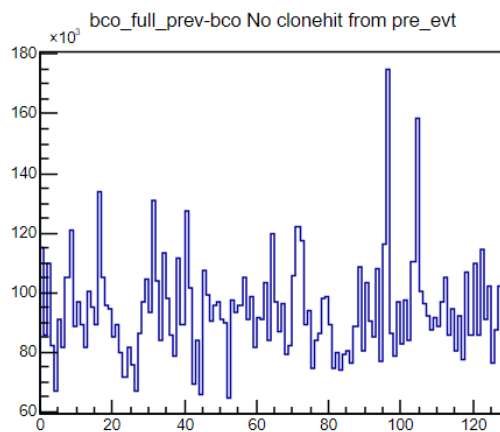
# 23947 24768 intt3



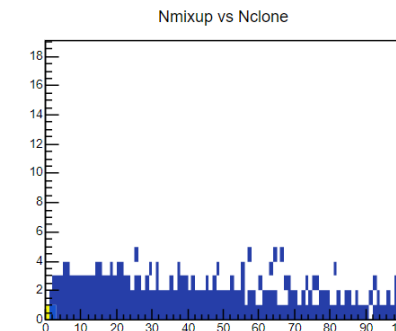
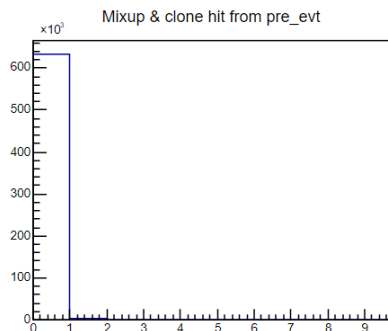
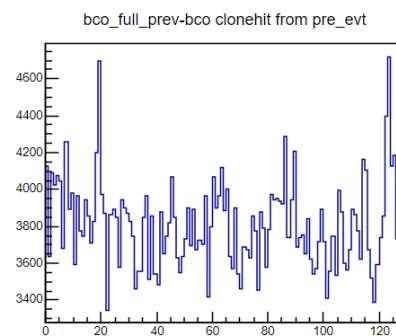
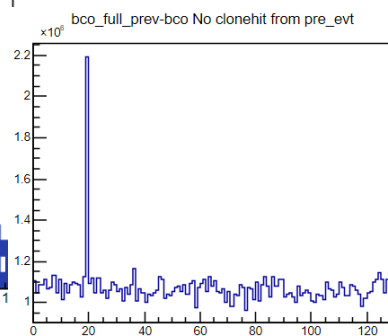
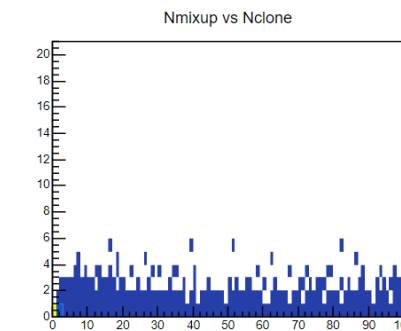
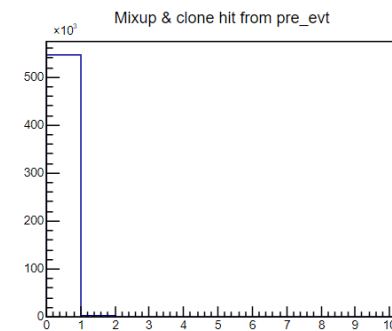
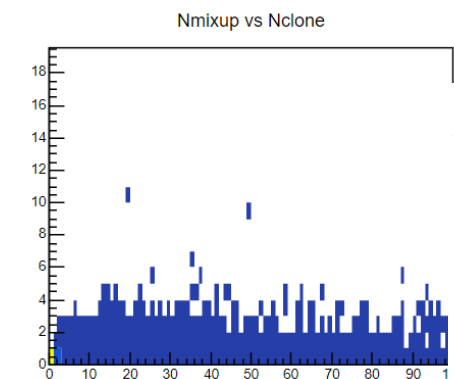
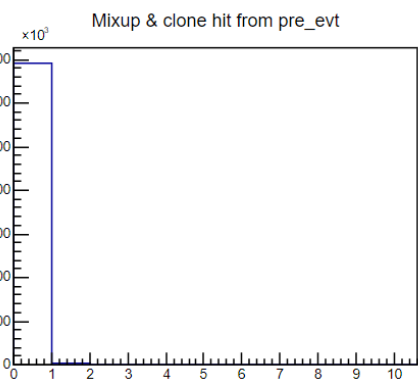
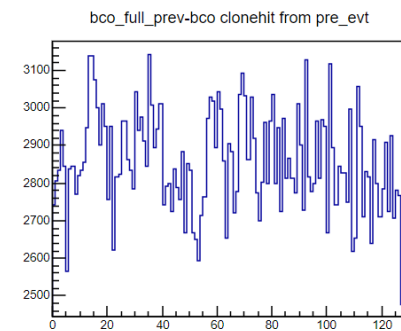
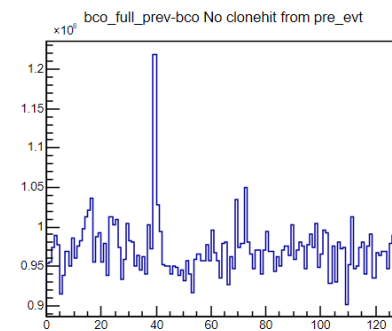
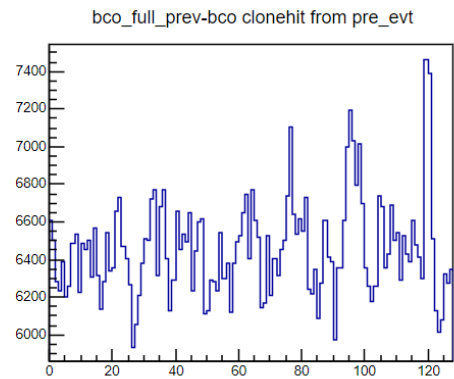
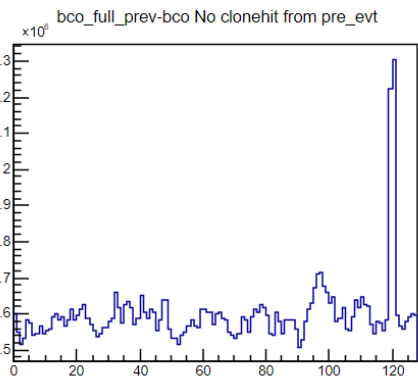
# 20444 20708 20869 intt4



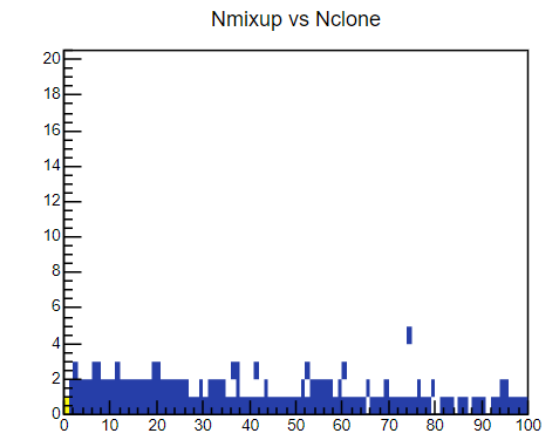
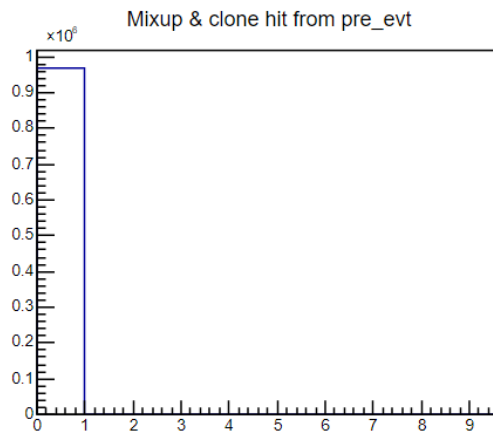
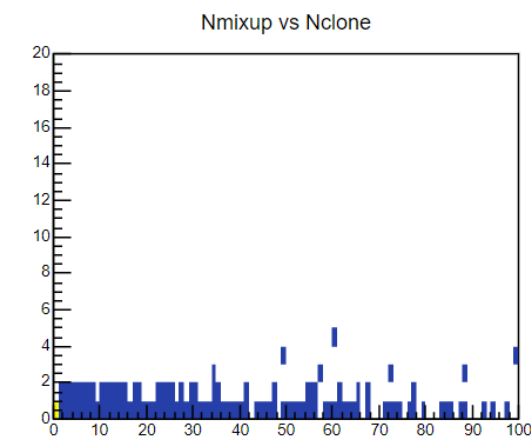
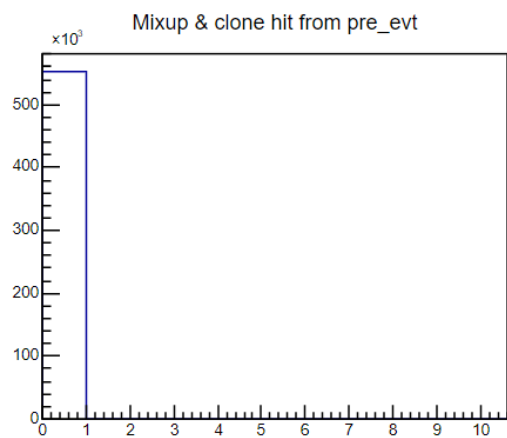
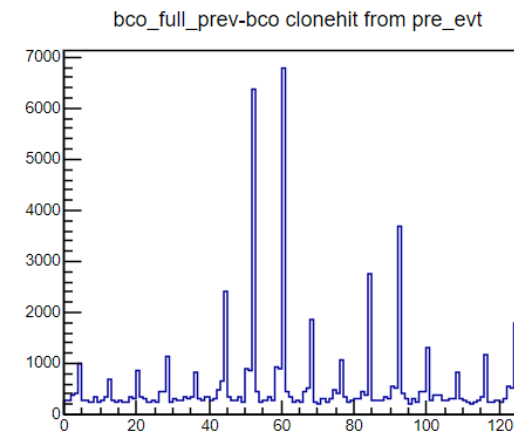
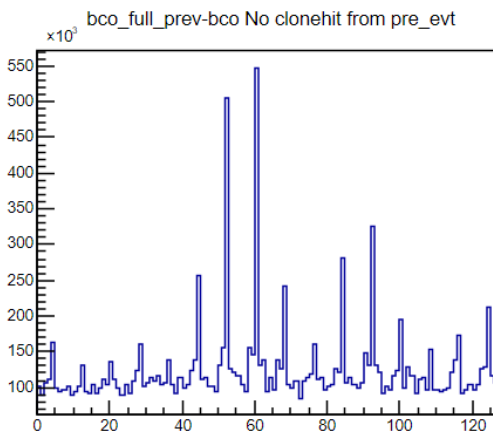
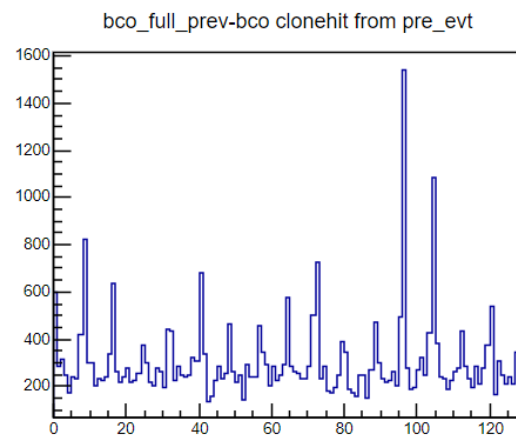
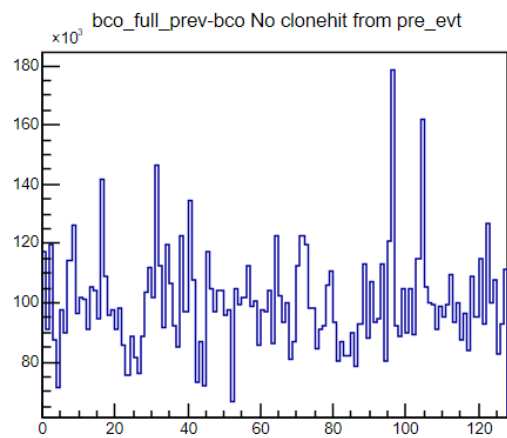
# 23947 24768 intt4



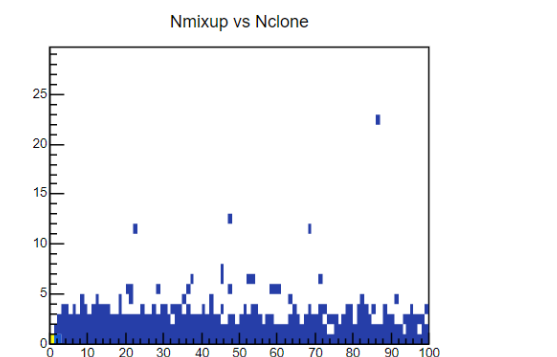
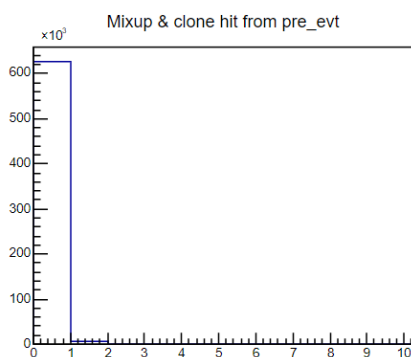
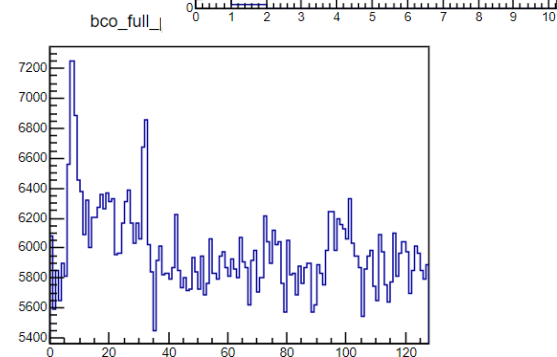
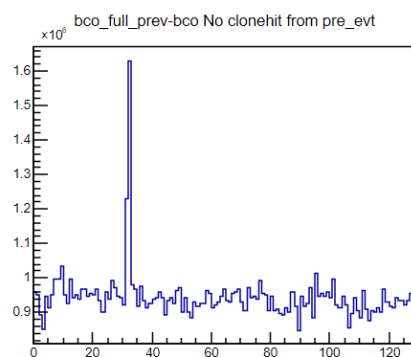
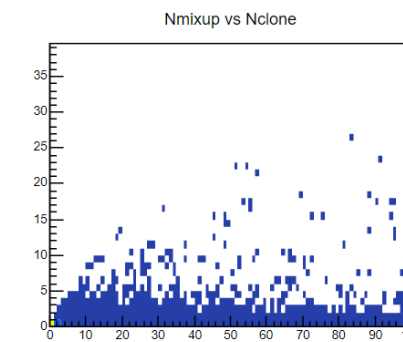
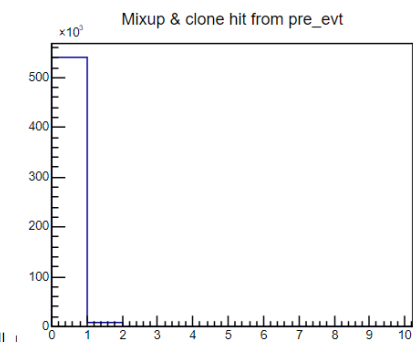
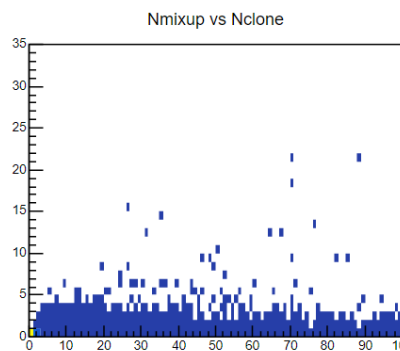
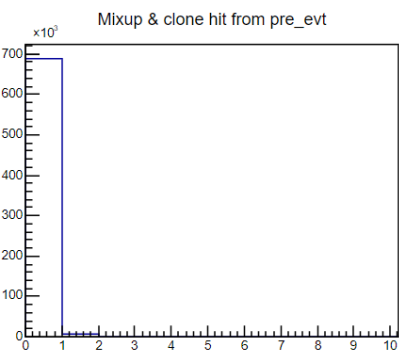
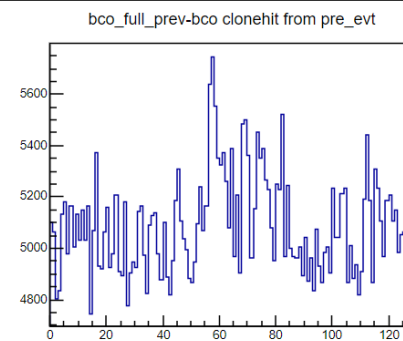
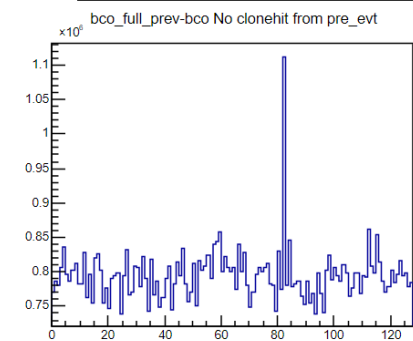
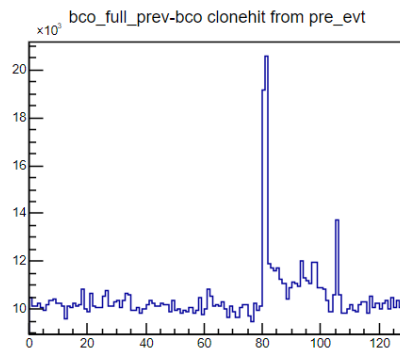
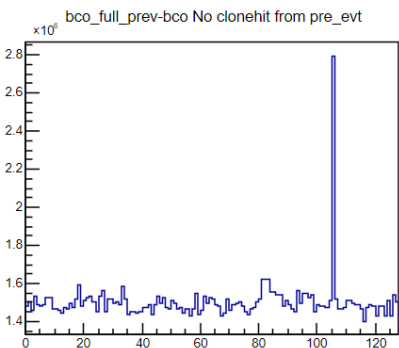
# 20444 20708 20869 intt5



# 23947 24768 intt5

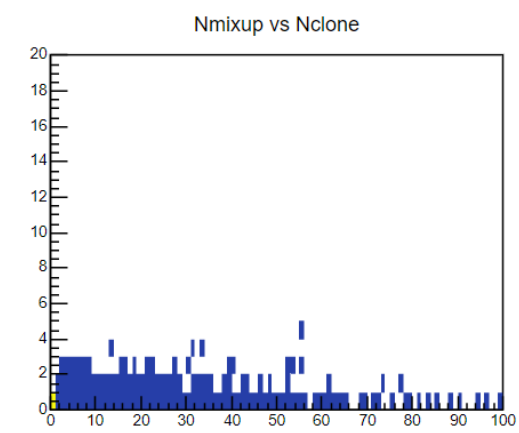
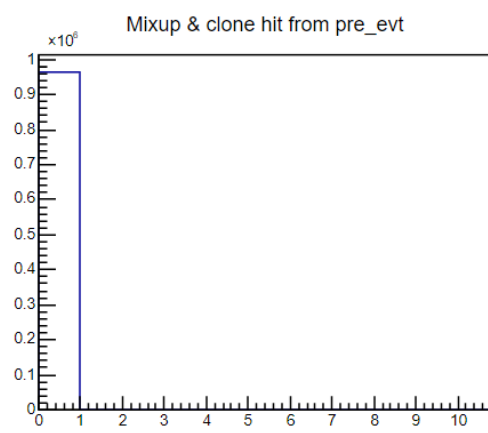
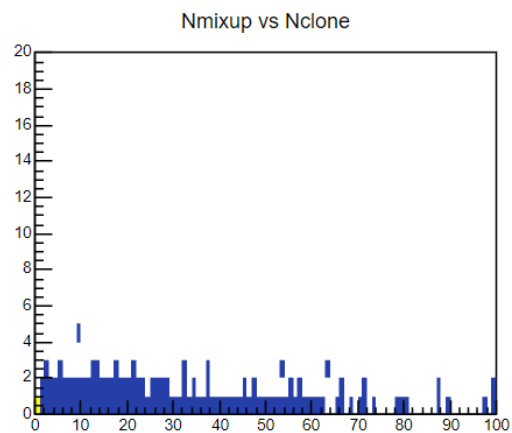
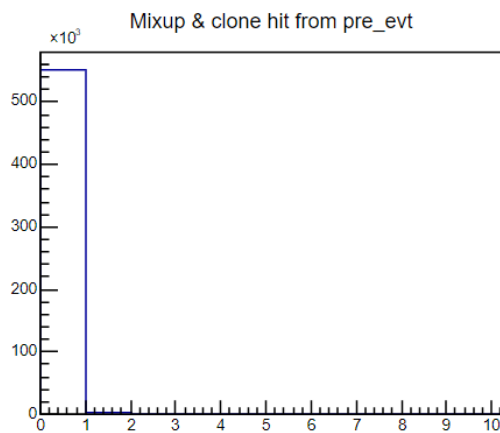
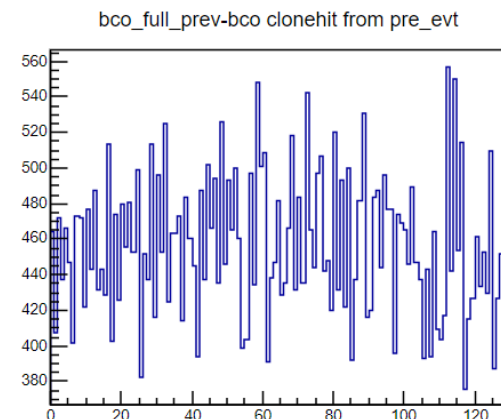
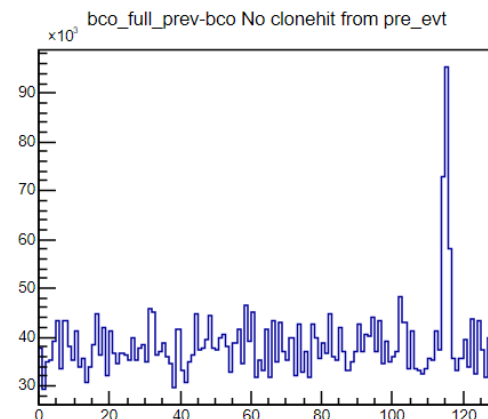
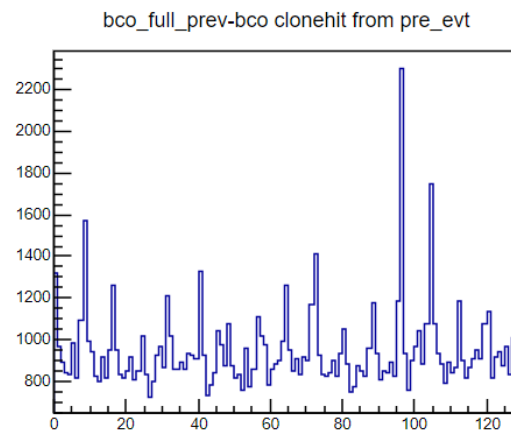
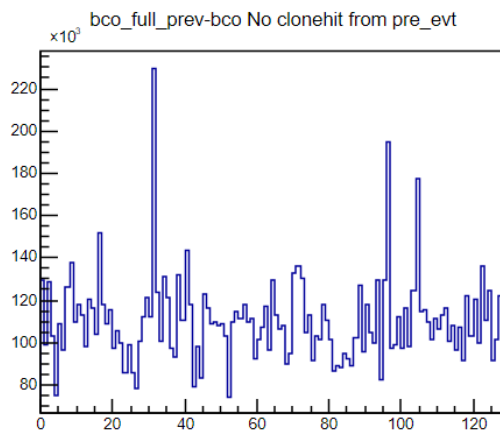


20444 20708 20869 intt6

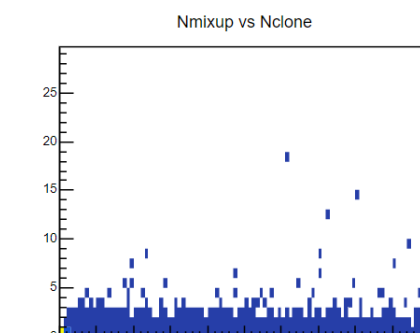
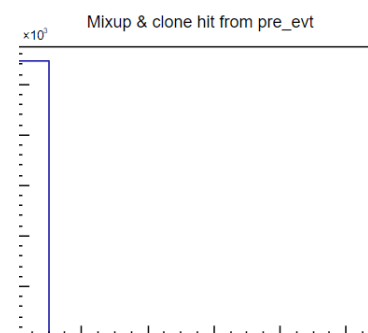
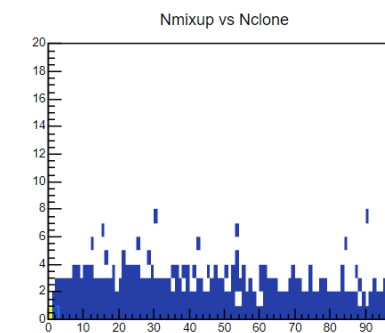
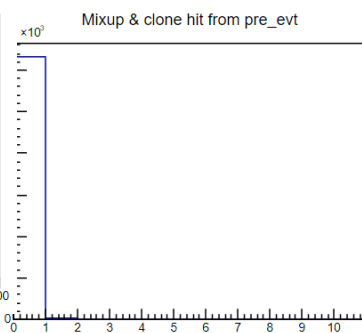
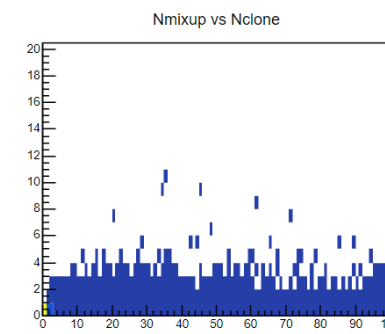
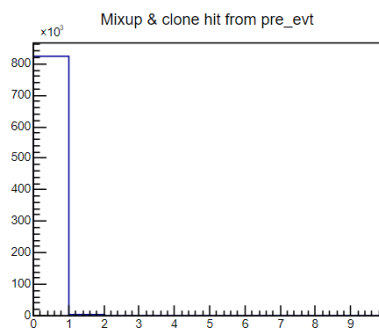
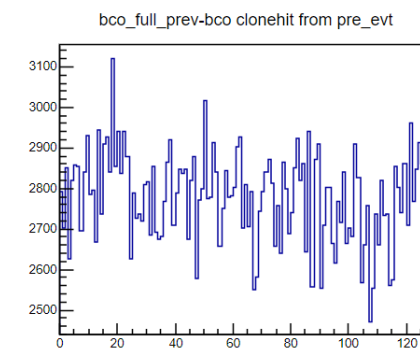
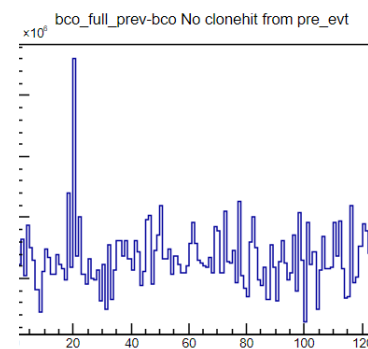
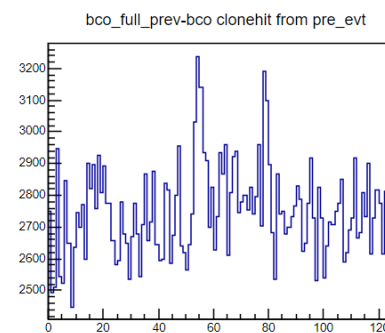
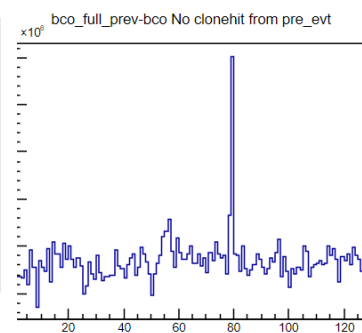
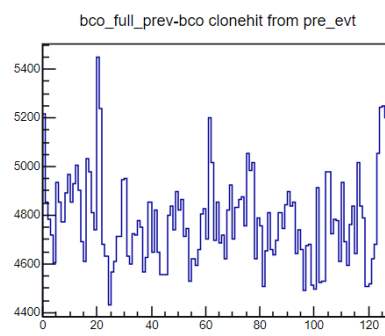
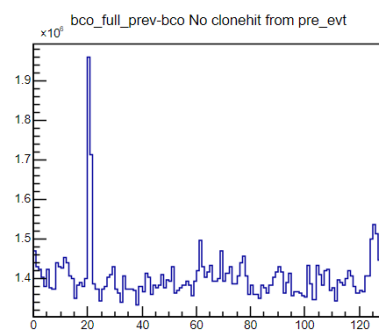




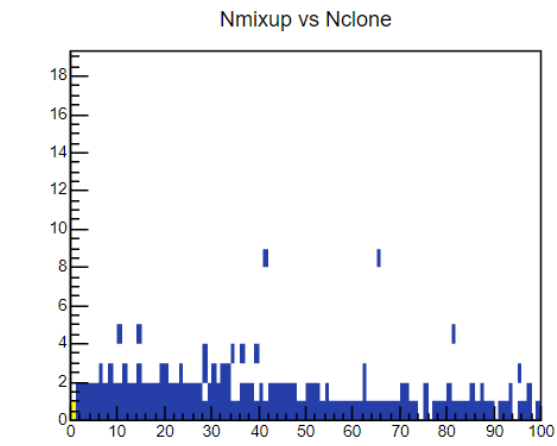
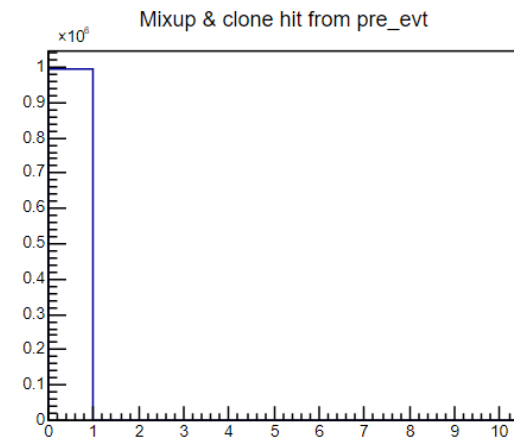
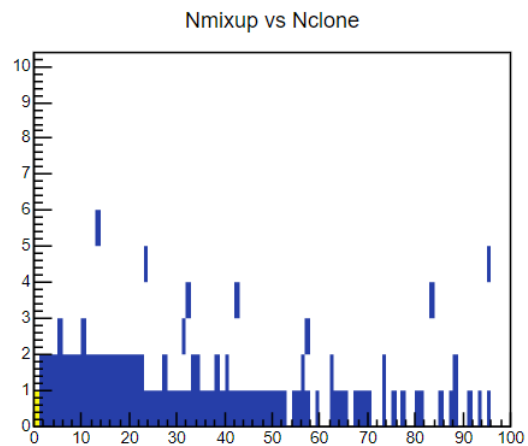
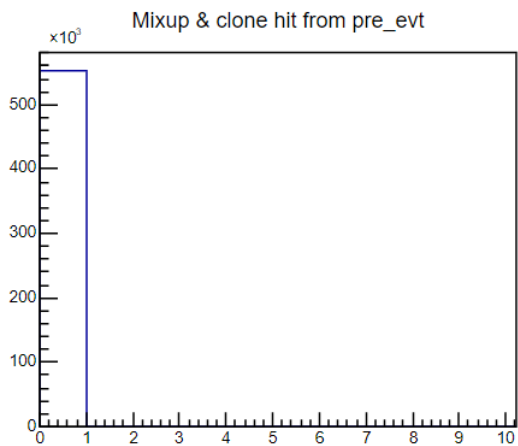
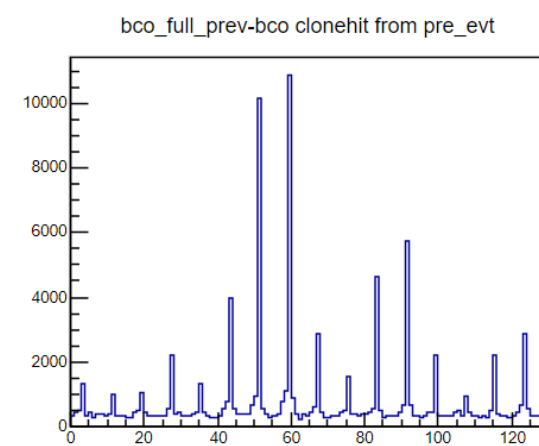
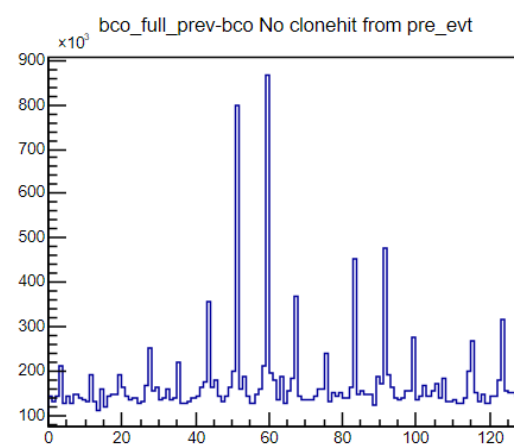
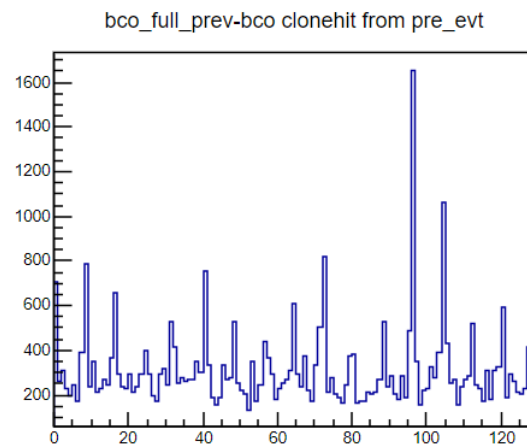
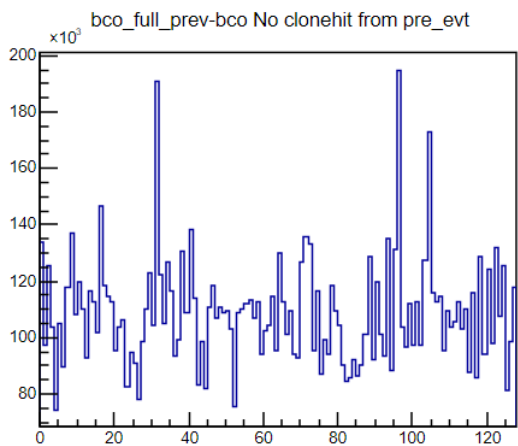
# 23947 24768 intt6



20444 20708 20869 intt7

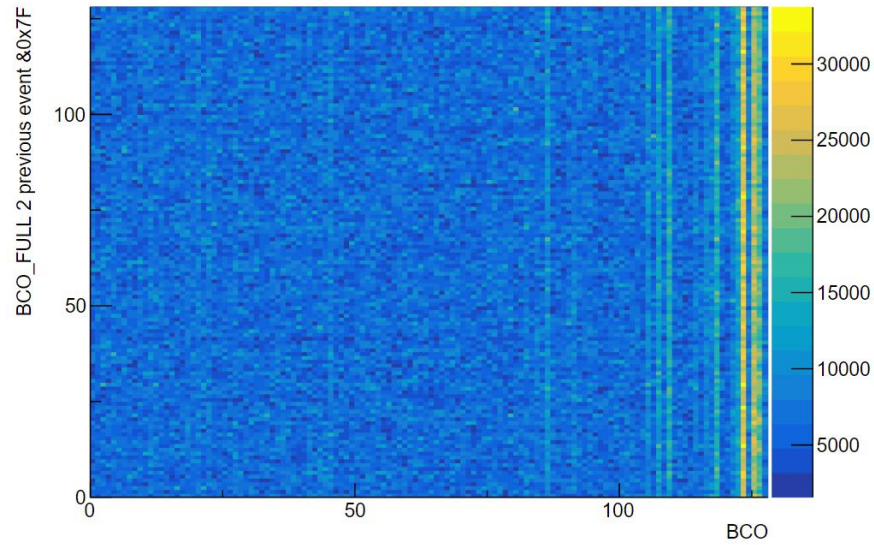


# 23947 24768 intt7

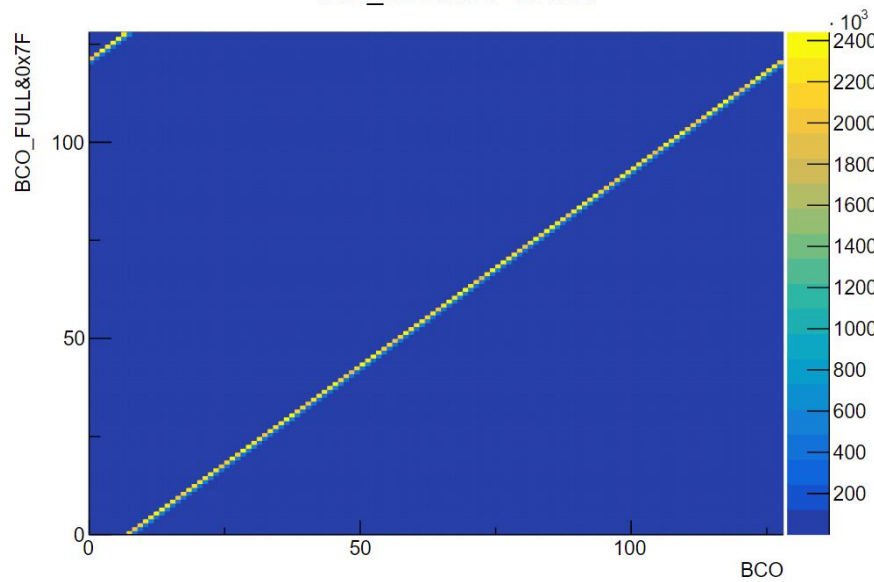


# 2 previous event 20708 intt0

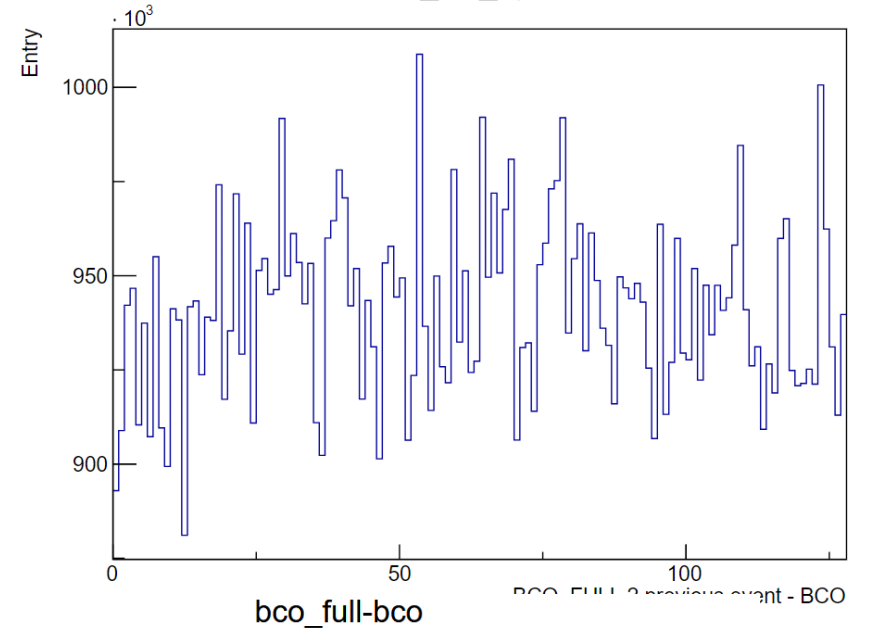
bco\_full&0x7F\_2prev vs bco 20708



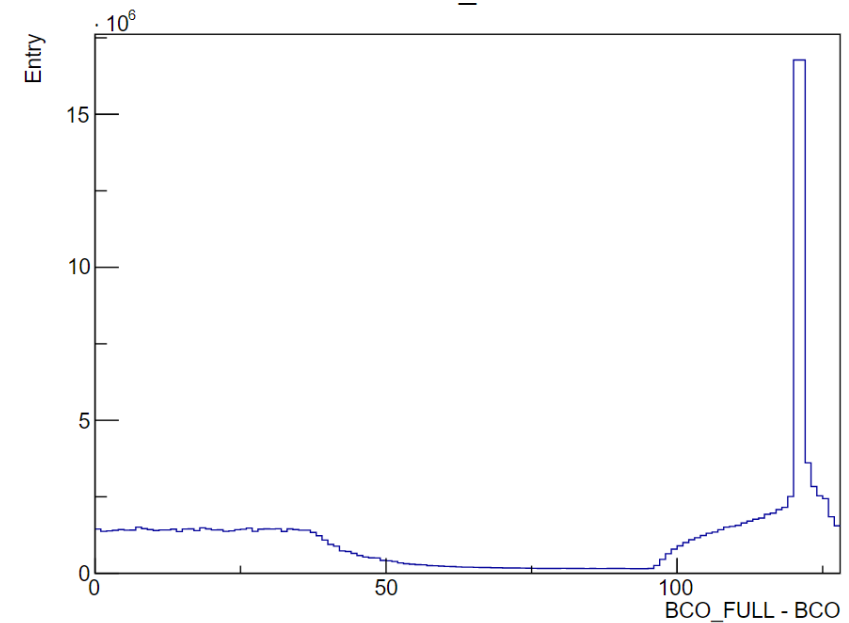
bco\_full&0x7F vs bco



bco\_full\_2prev-bco

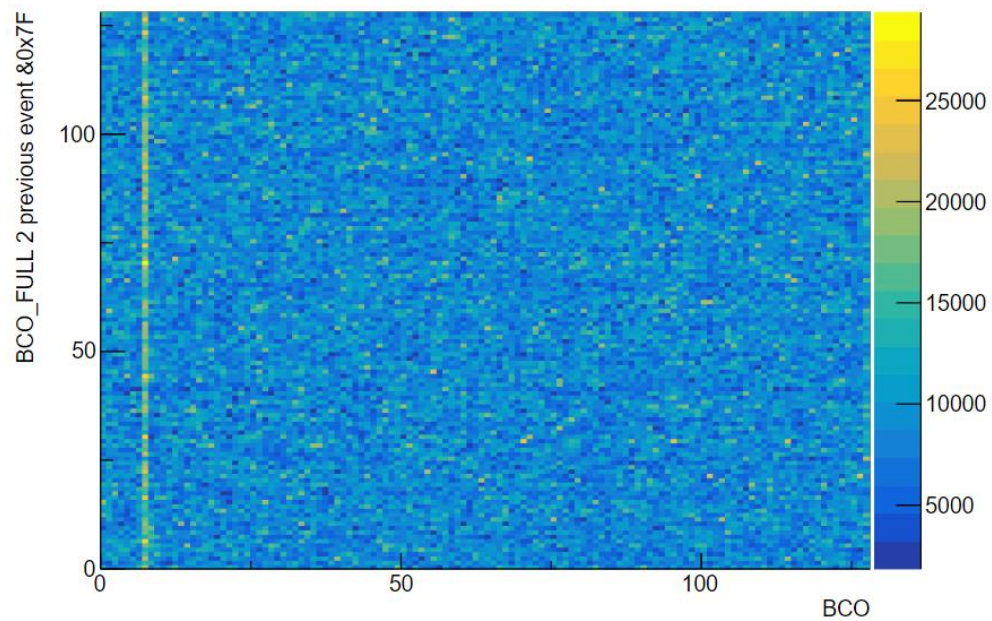


bco\_full-bco

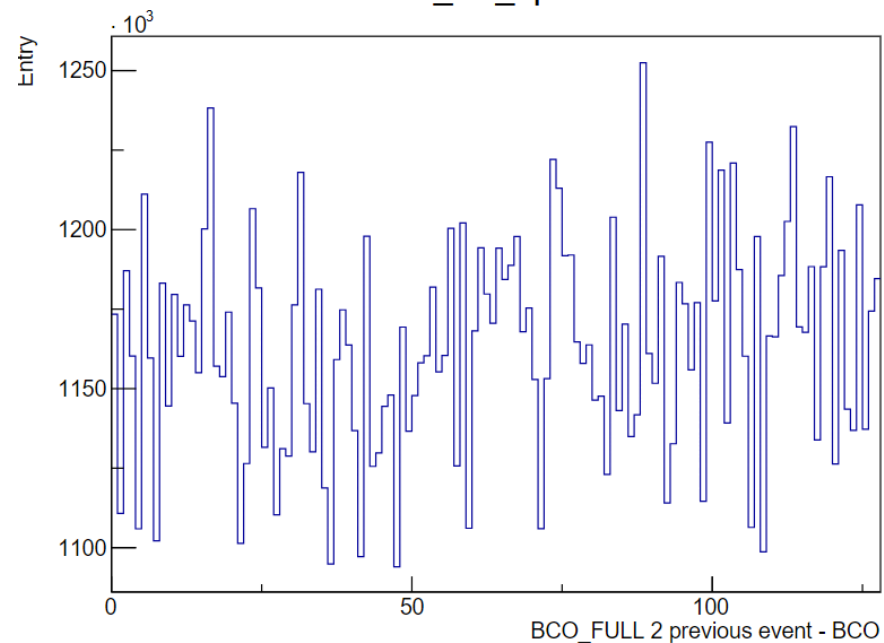


# 2 previous event 20708 intt1

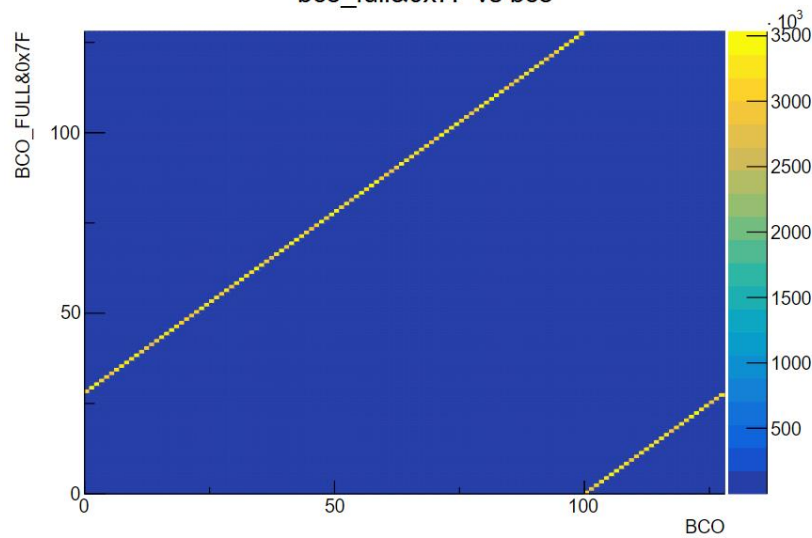
bco\_full&0x7F\_2prev vs bco 20708



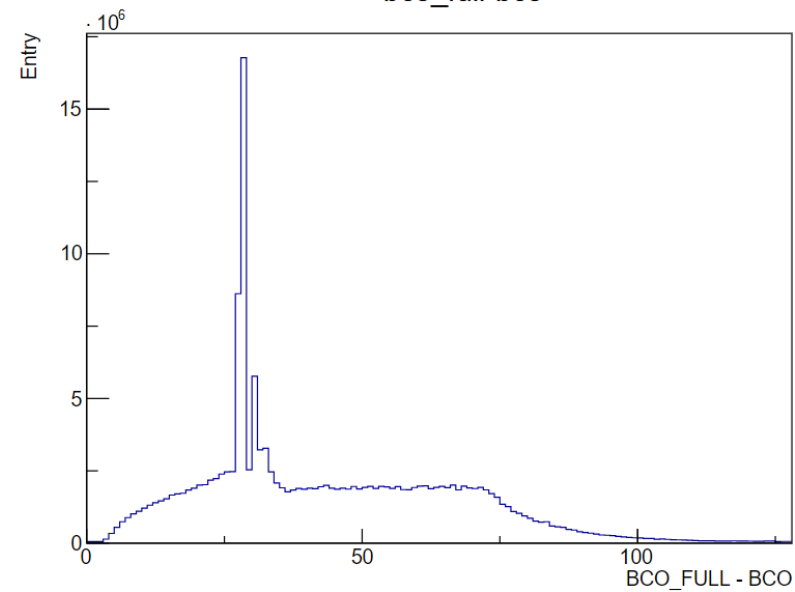
bco\_full\_2prev-bco



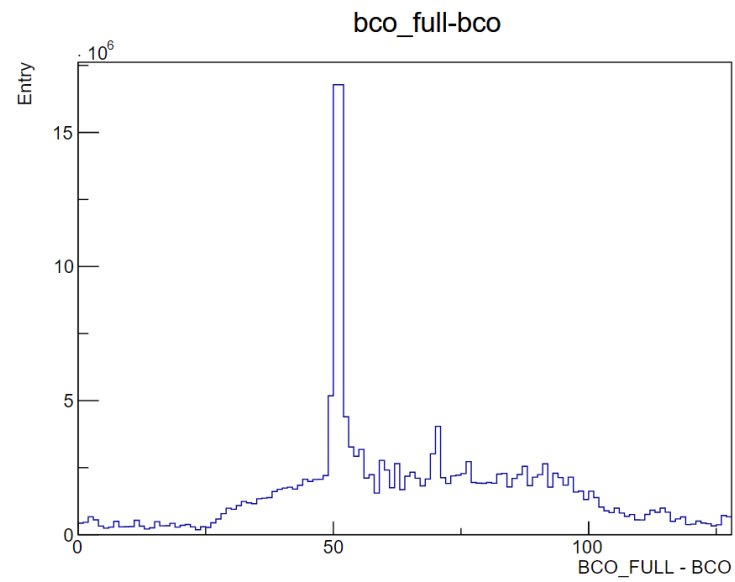
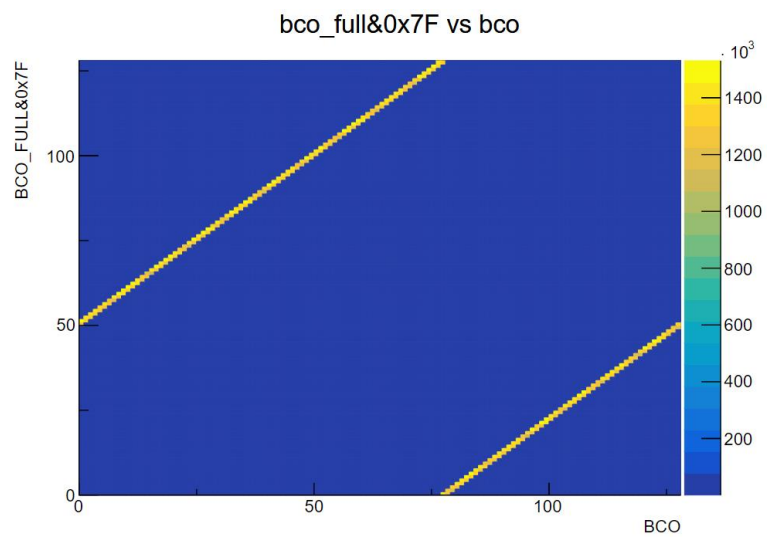
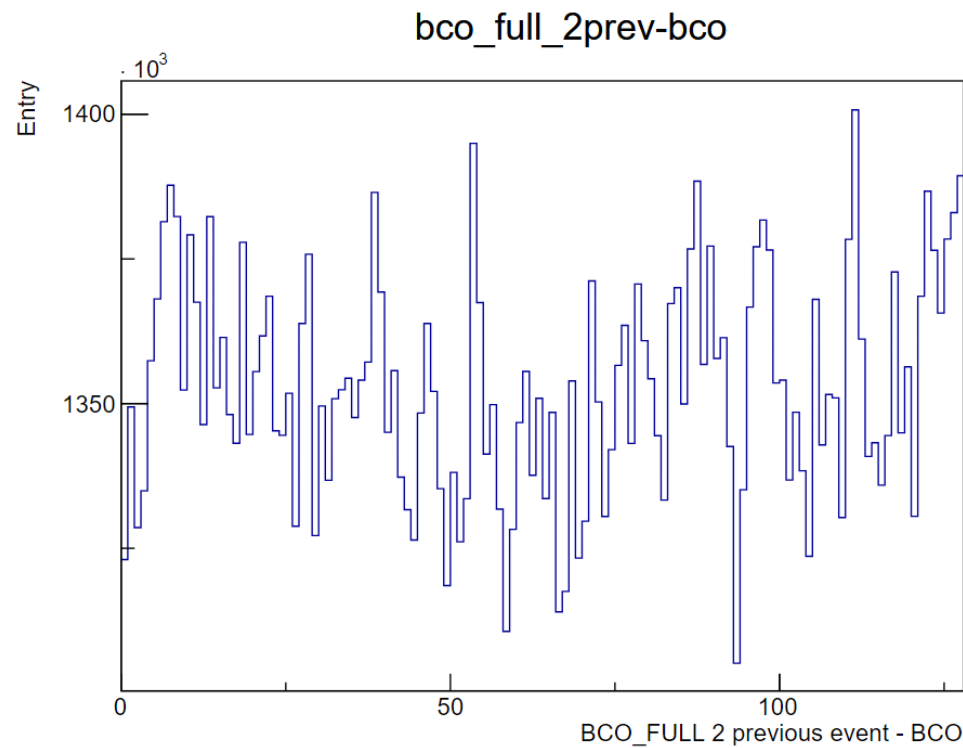
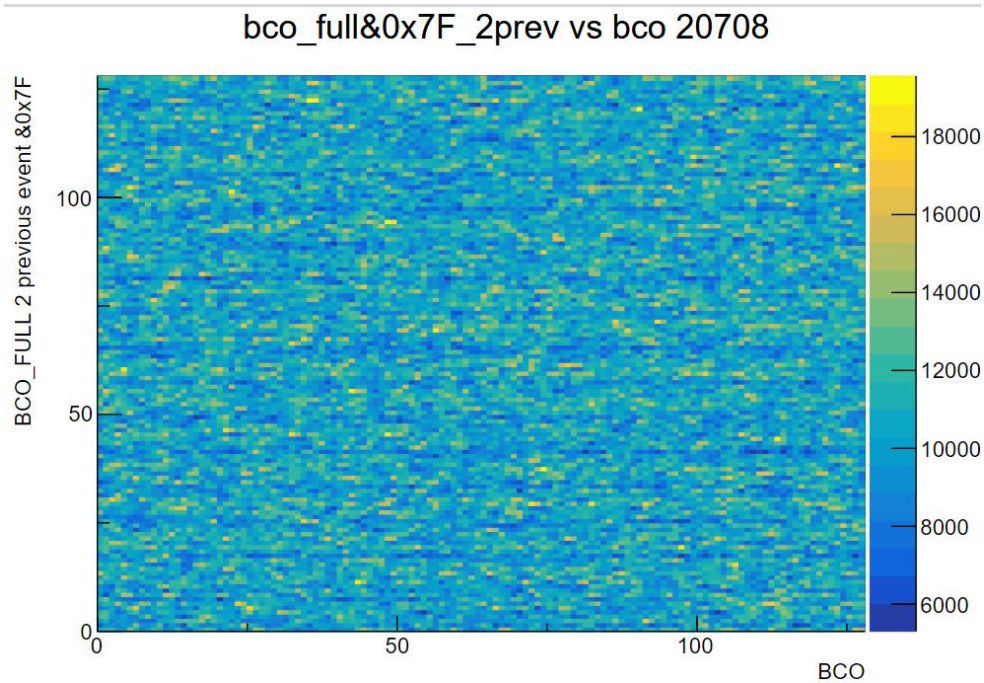
bco\_full&0x7F vs bco



bco\_full-bco



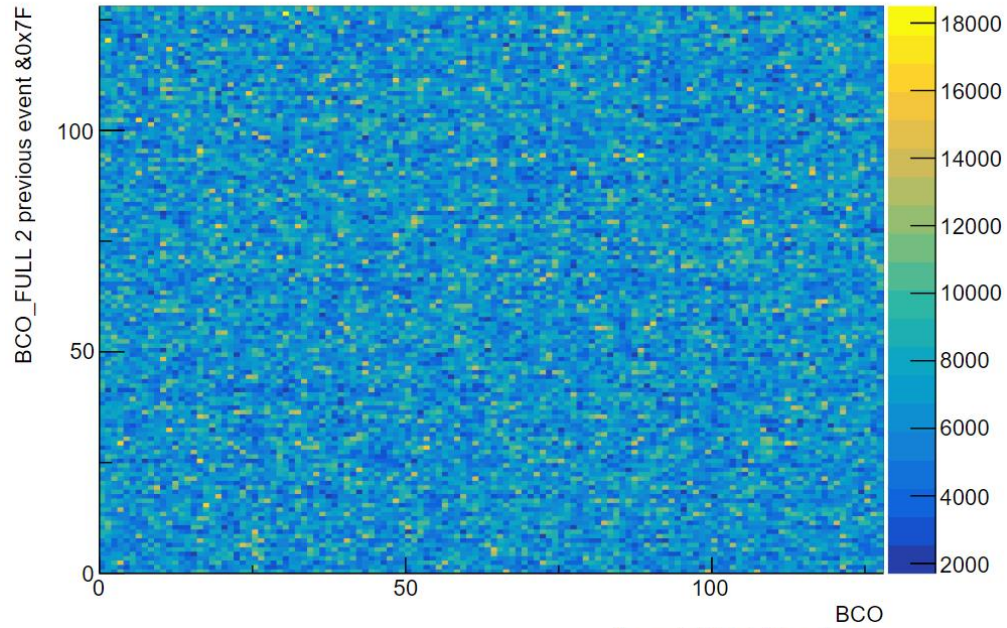
# 2 previous event 20708 intt2



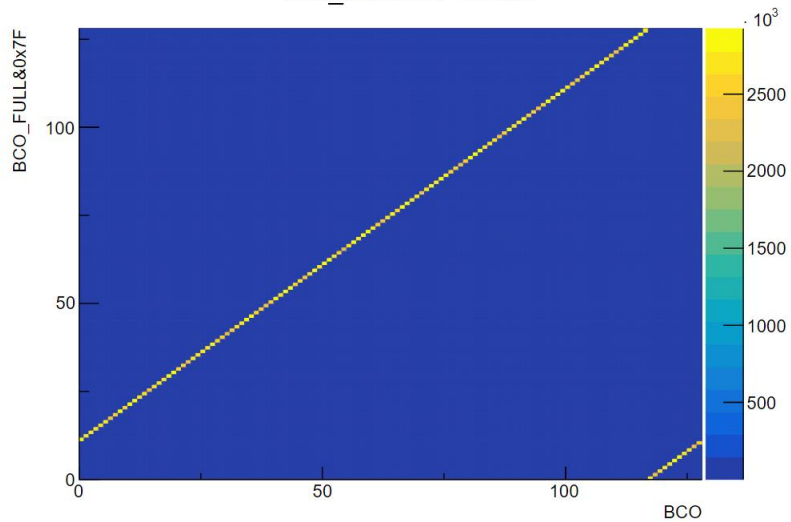
2024/5/24

# 2 previous event 20708 intt3

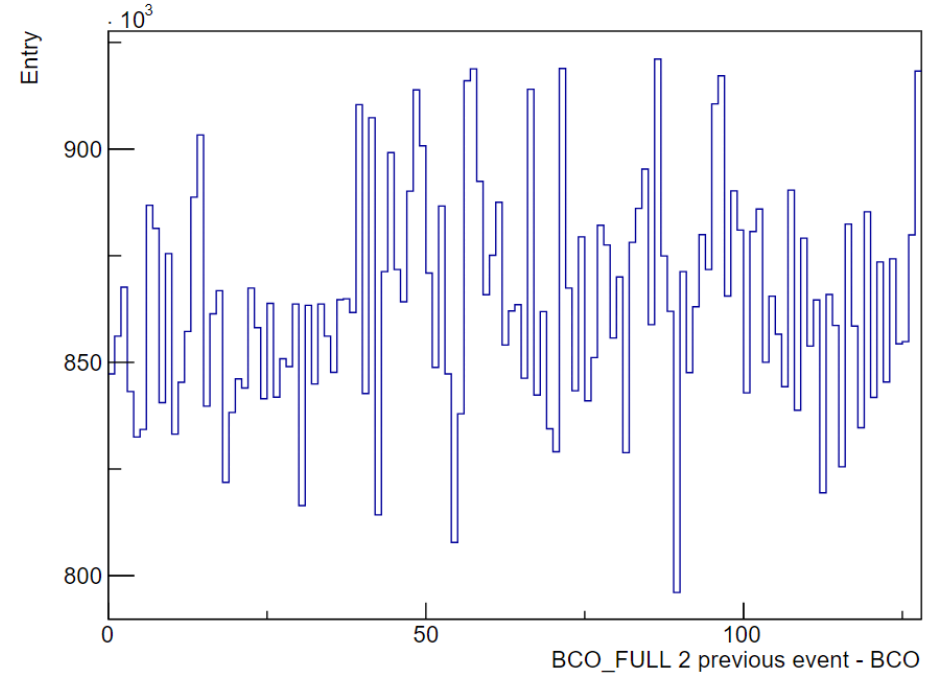
bco\_full&0x7F\_2prev vs bco 20708



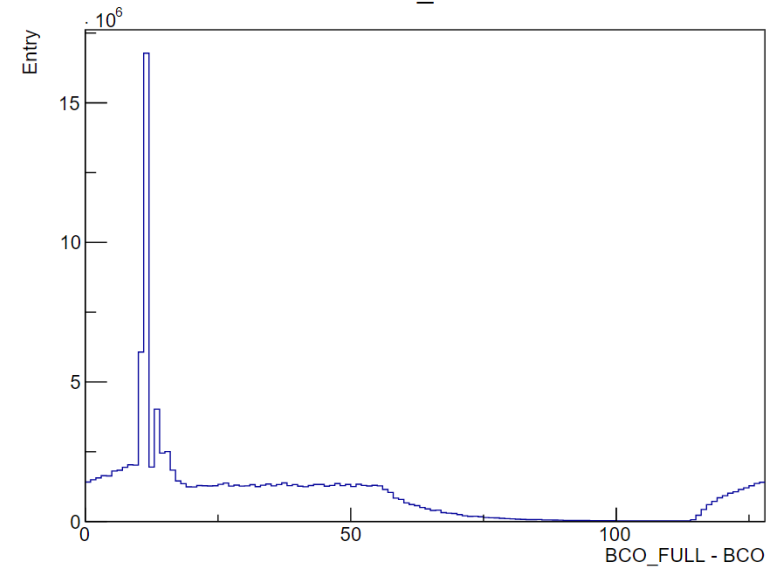
bco\_full&0x7F vs bco



bco\_full\_2prev-bco

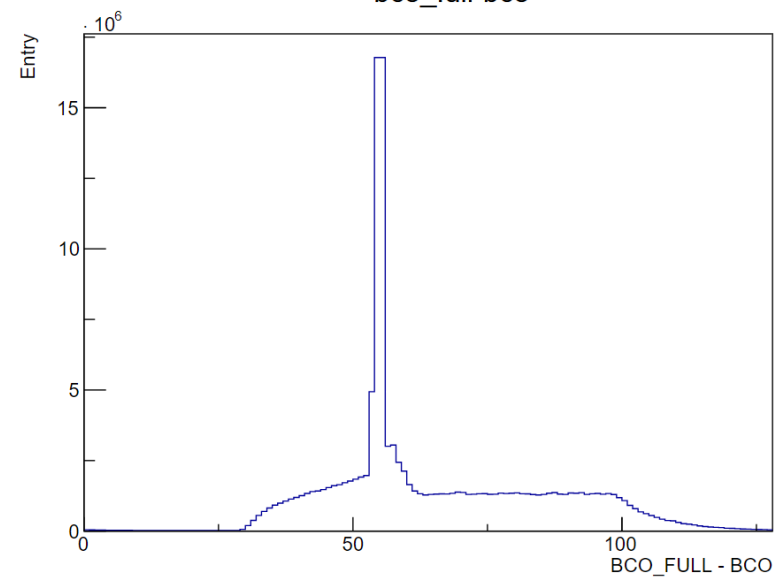
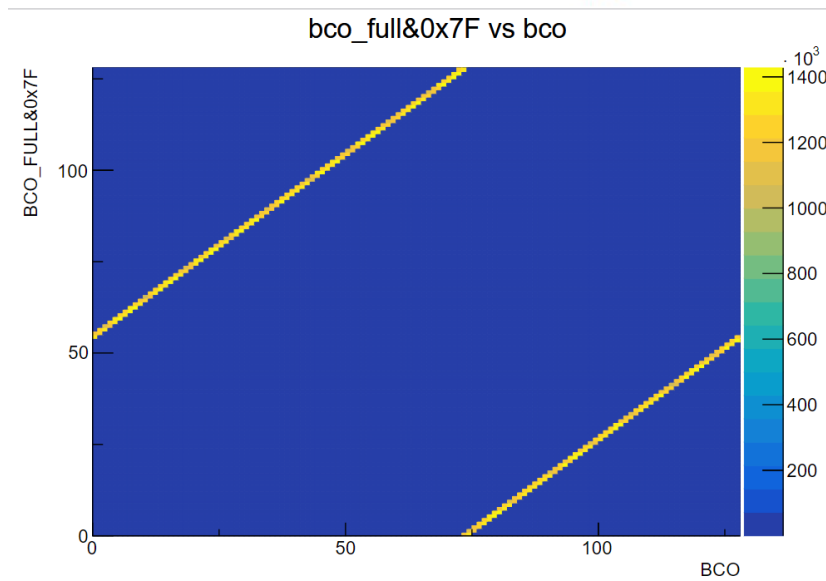
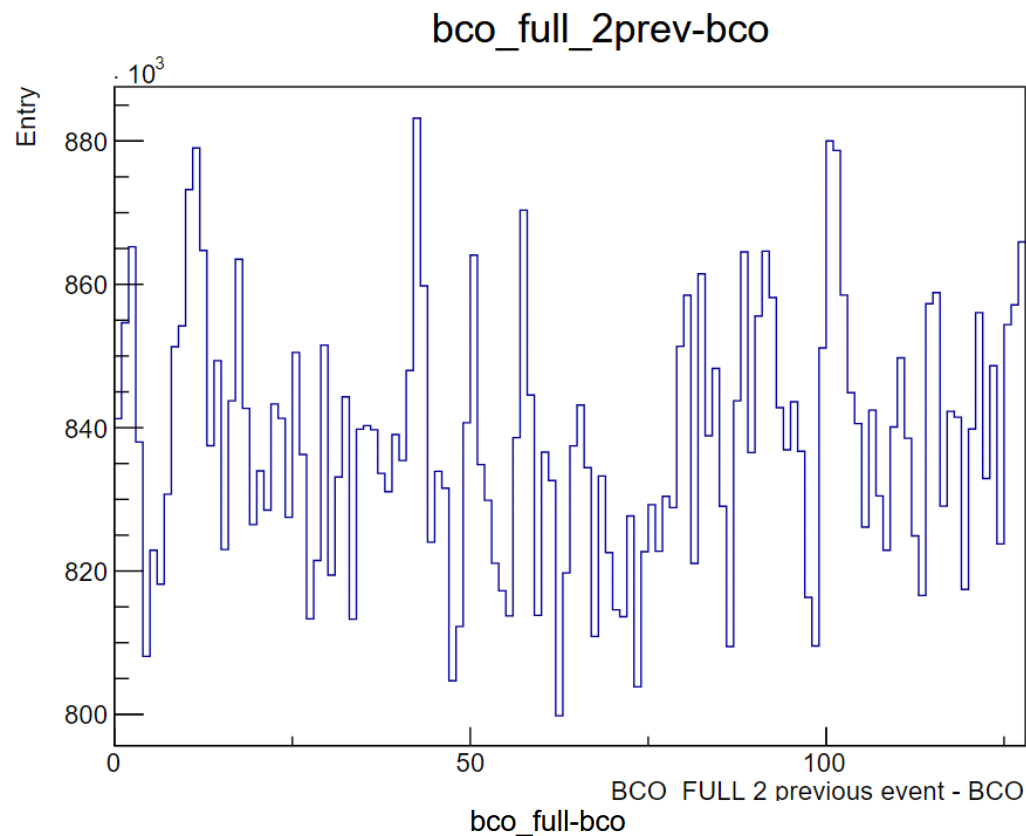
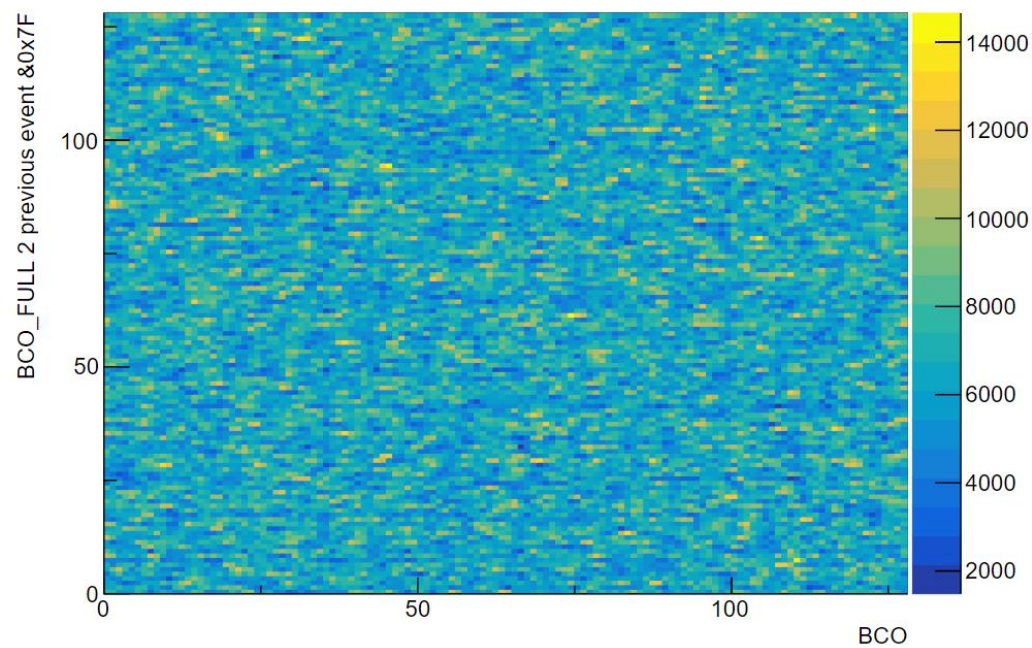


bco\_full-bco



# 2 previous event 20708 intt4

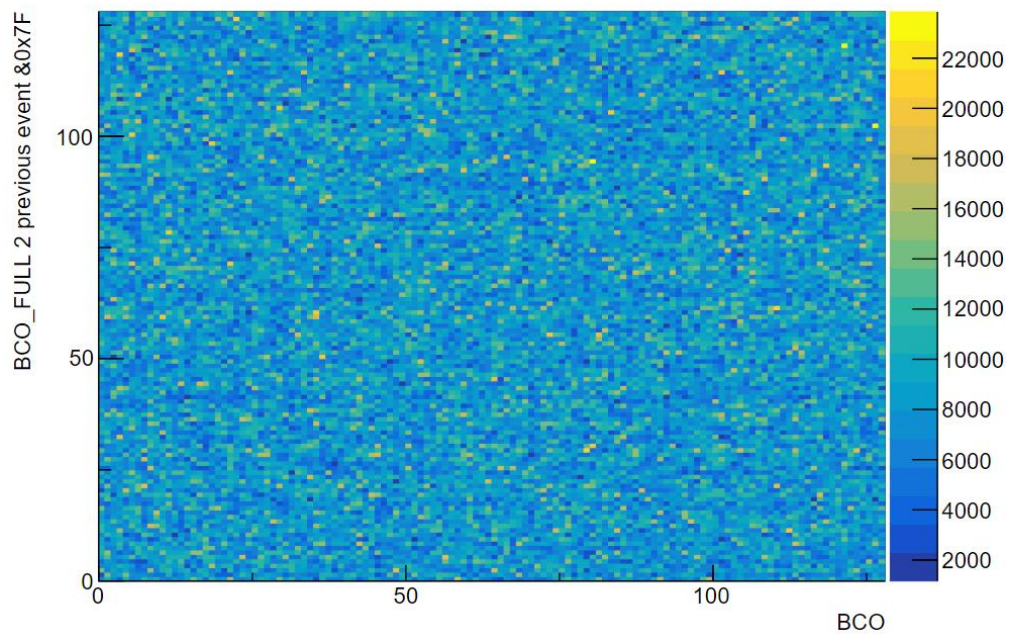
bco\_full&0x7F\_2prev vs bco 20708



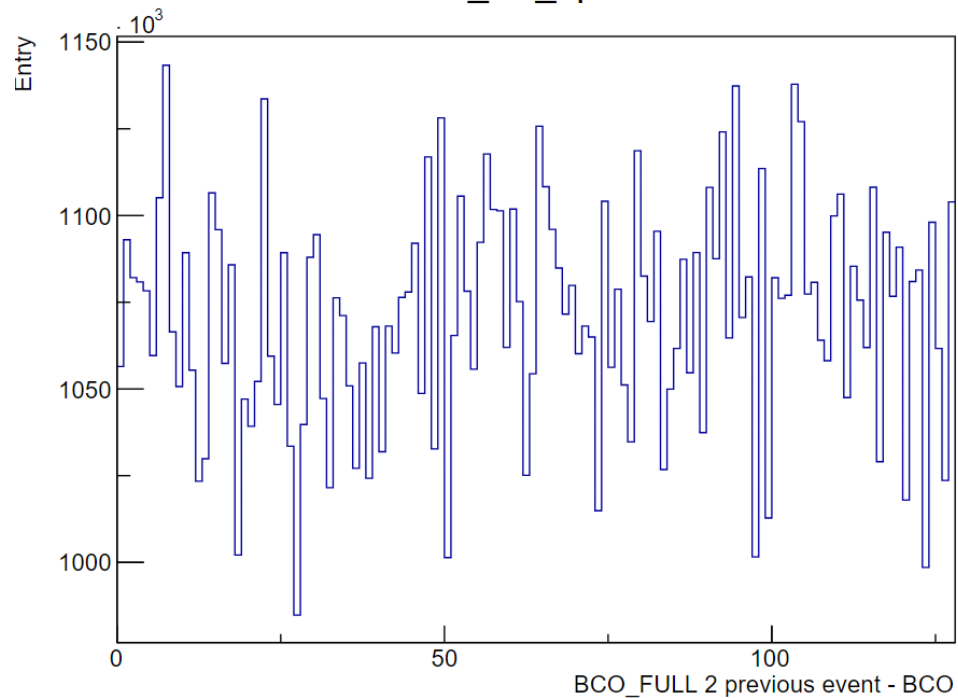


# 2 previous event 20708 intt5

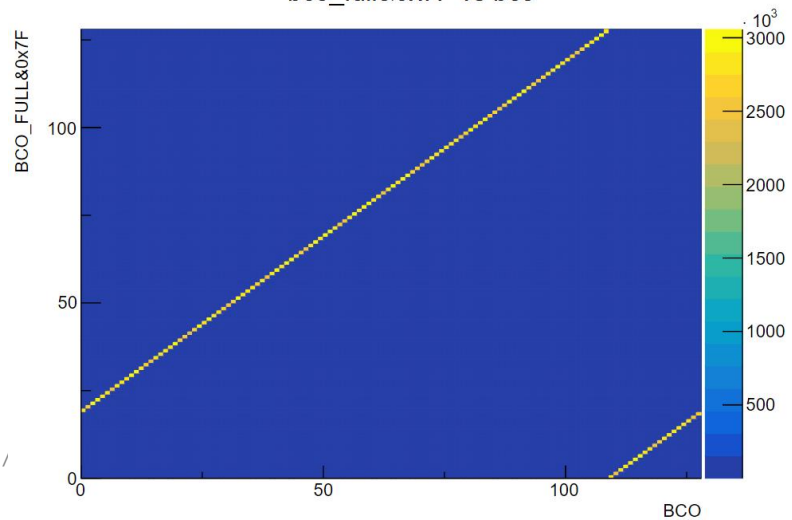
bco\_full&0x7F\_2prev vs bco 20708



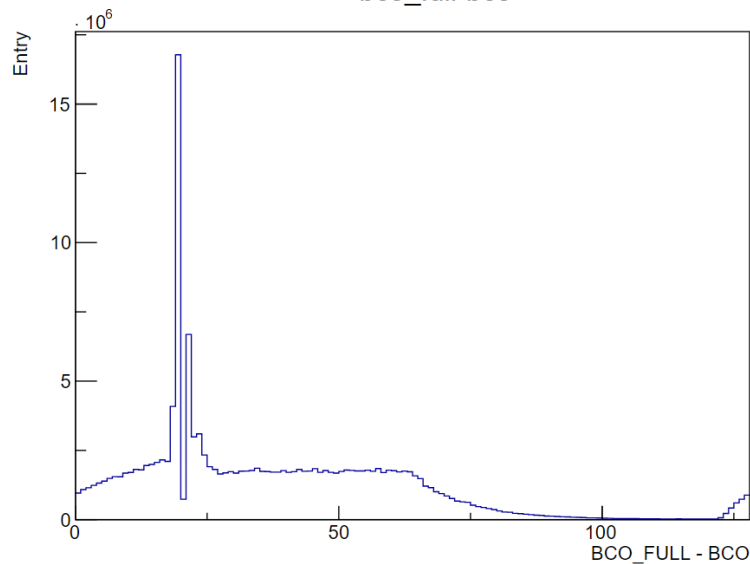
bco\_full\_2prev-bco



bco\_full&0x7F vs bco

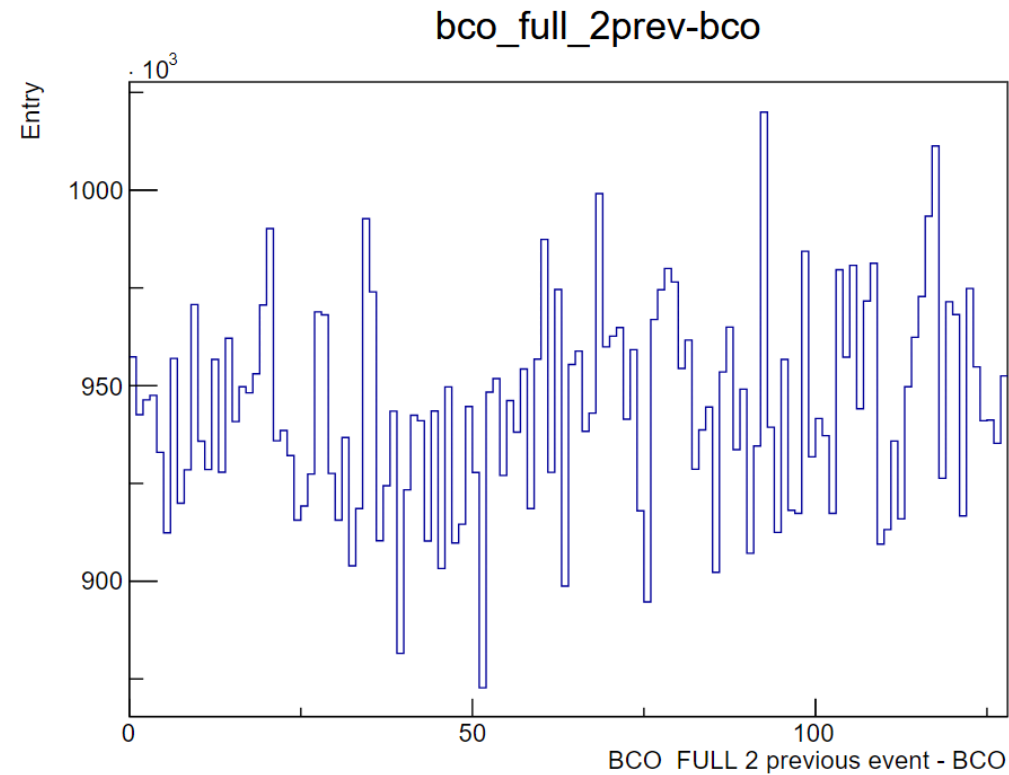
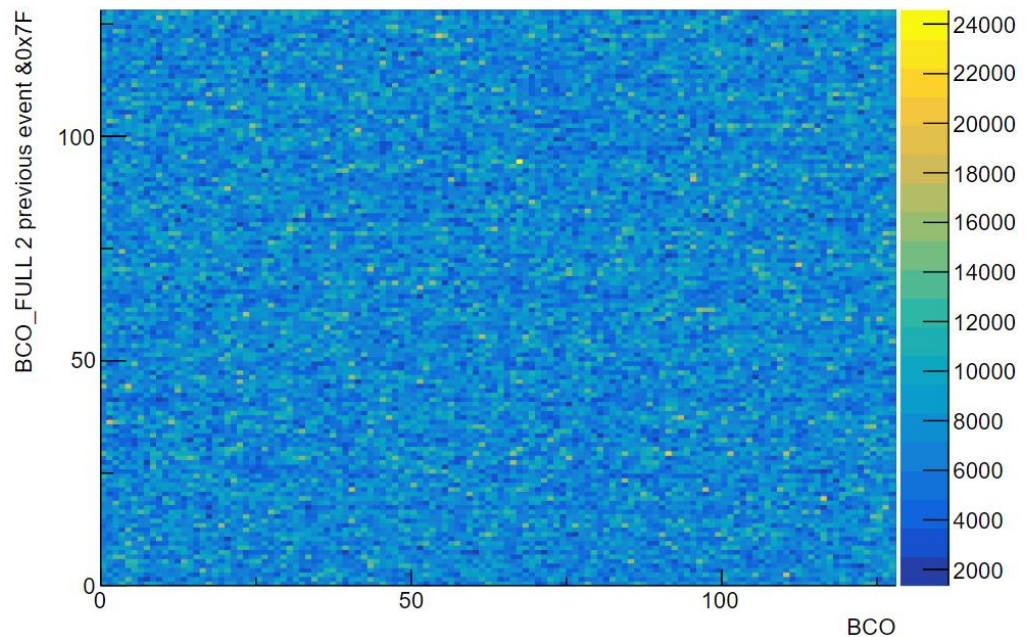


bco\_full-bco

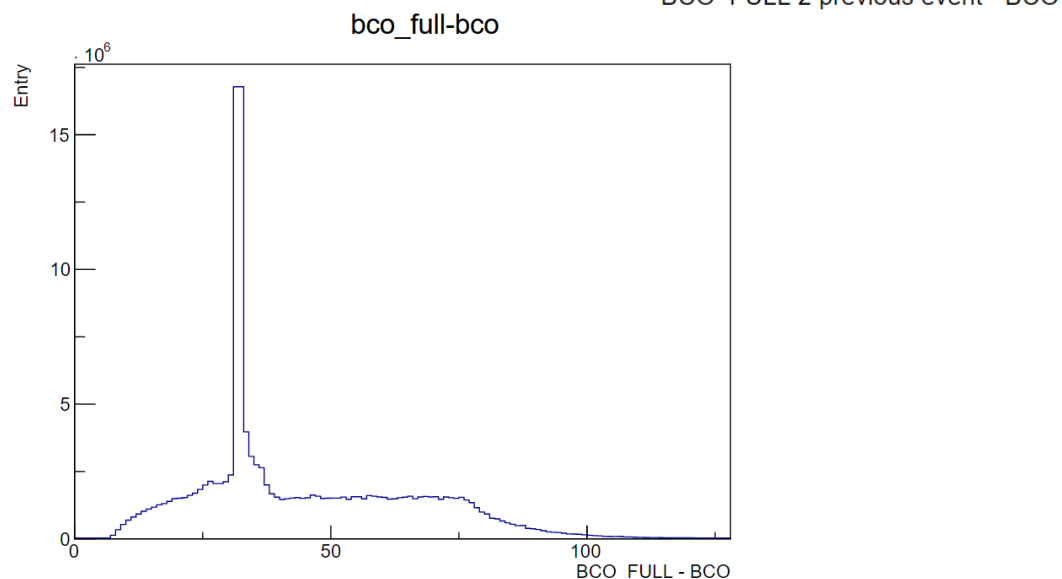
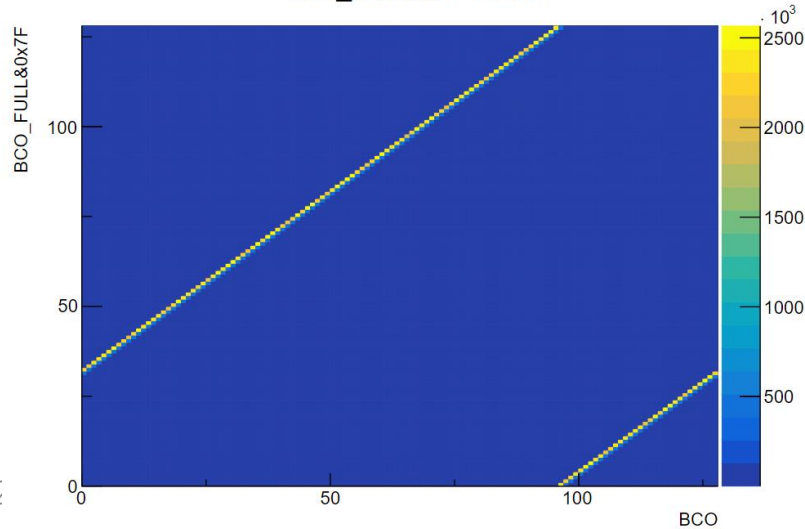


# 2 previous event 20708 intt6

bco\_full&0x7F\_2prev vs bco 20708

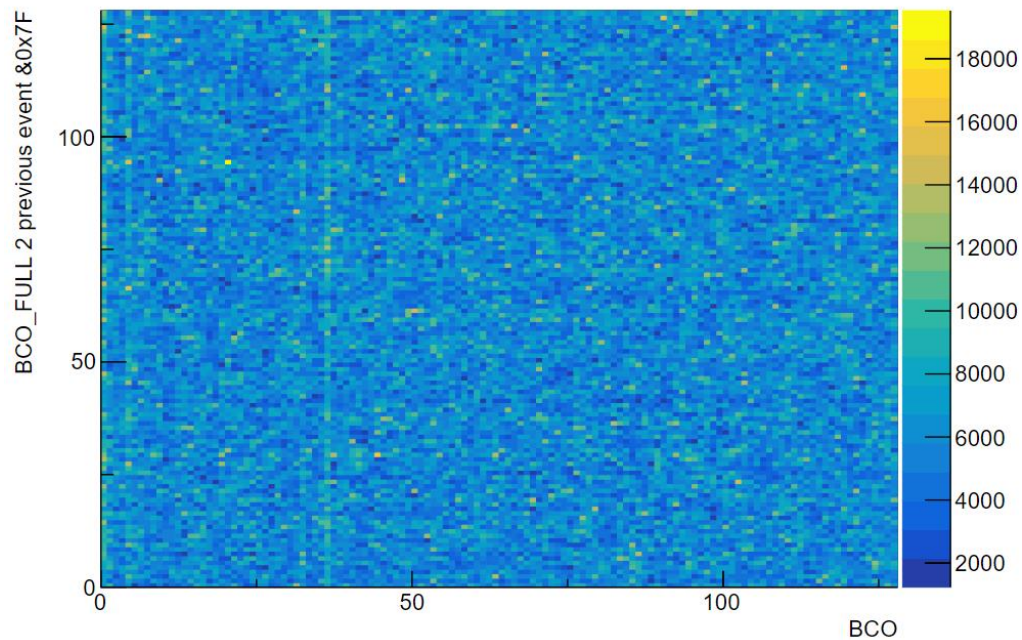


bco\_full&0x7F vs bco

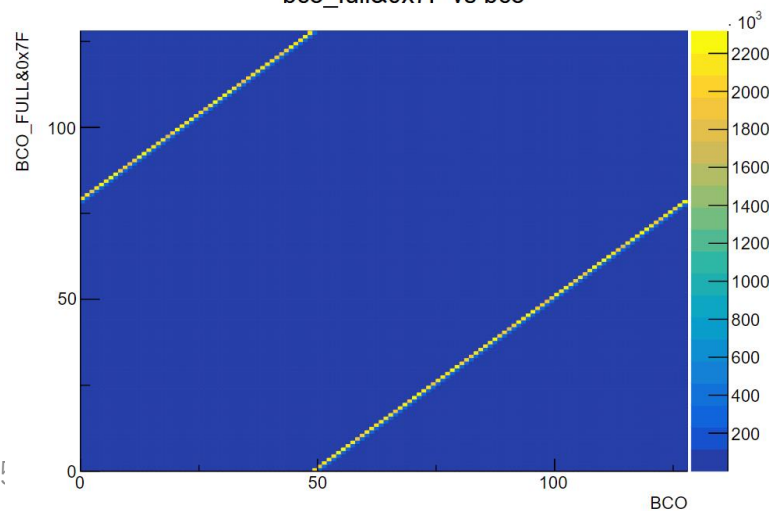


# 2 previous event 20708 intt7

bco\_full&0x7F\_2prev vs bco 20708

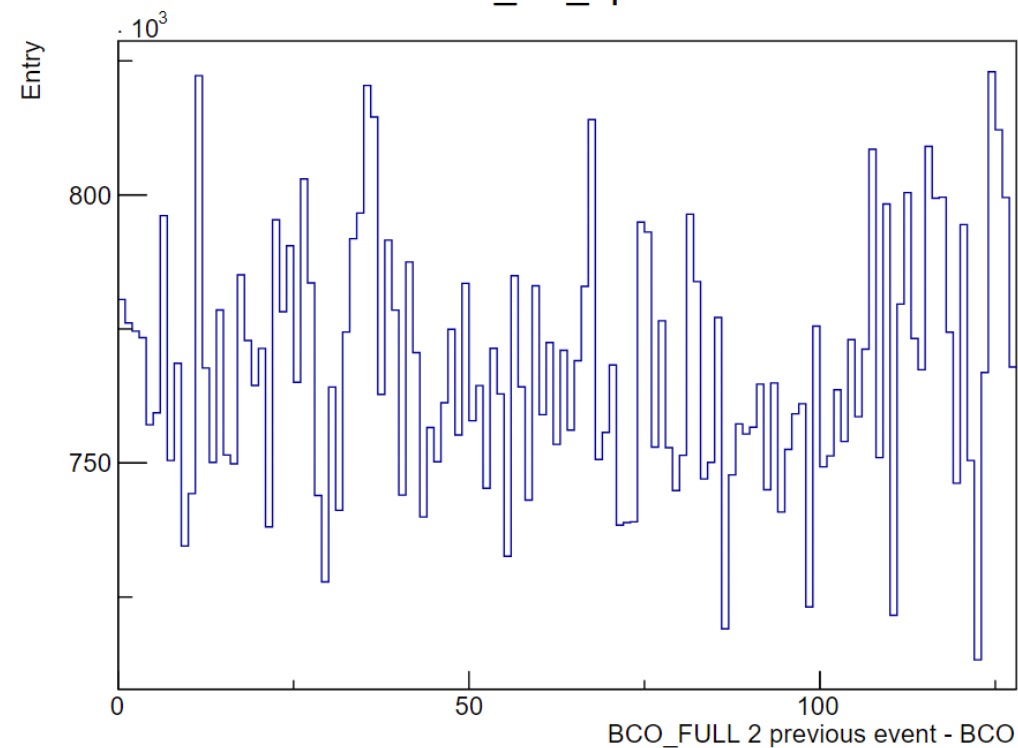


bco\_full&0x7F vs bco

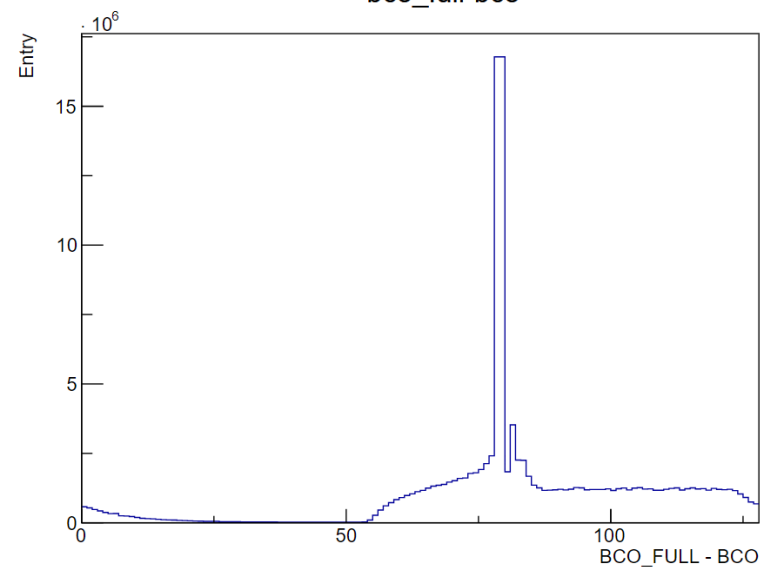


2024/!

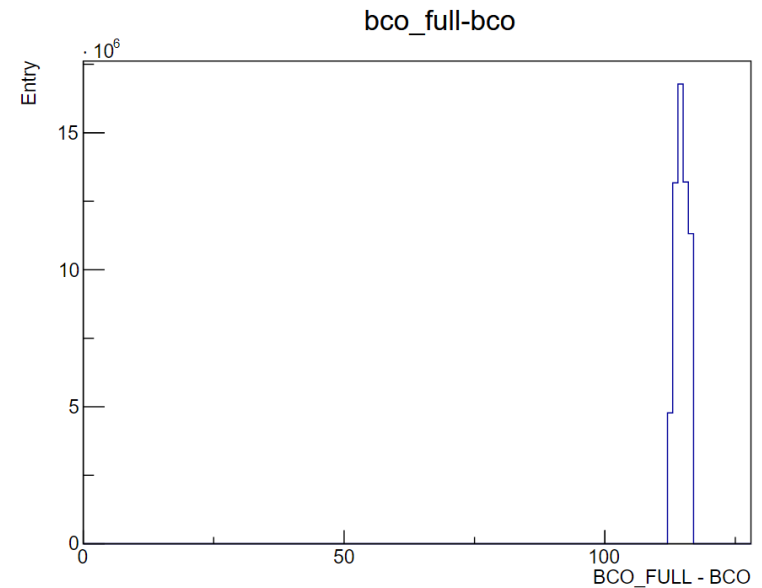
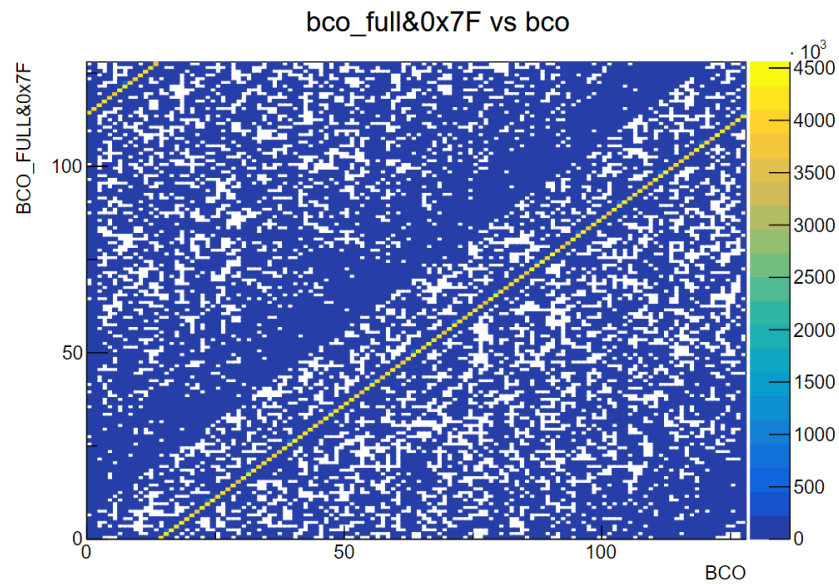
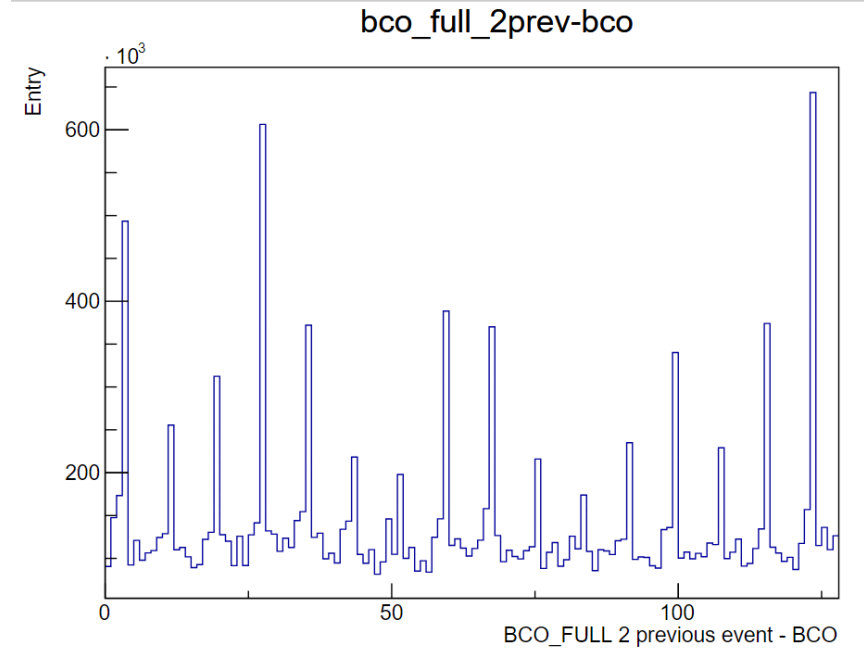
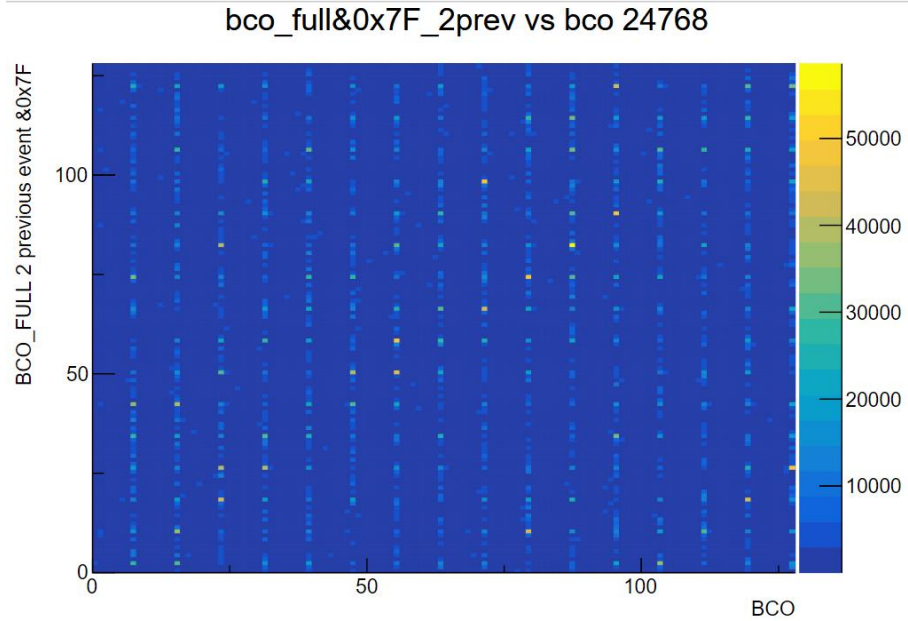
bco\_full\_2prev-bco



bco\_full-bco

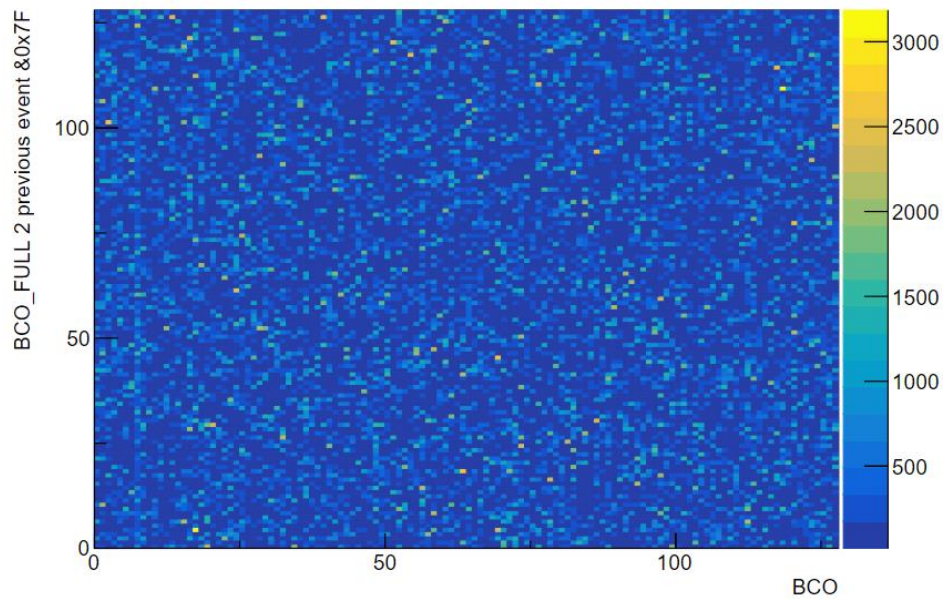


# 2 previous event 24768 intt0

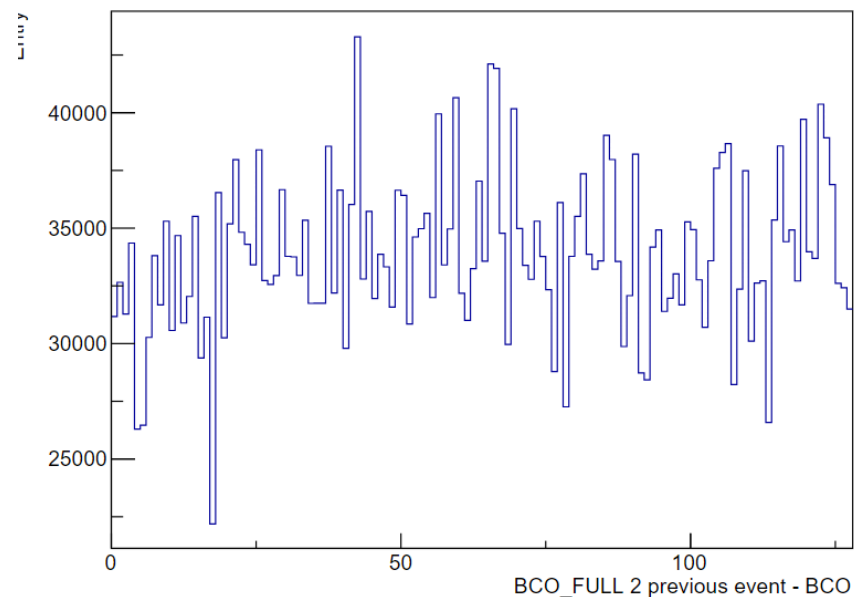


# 2 previous event 24768 intt1

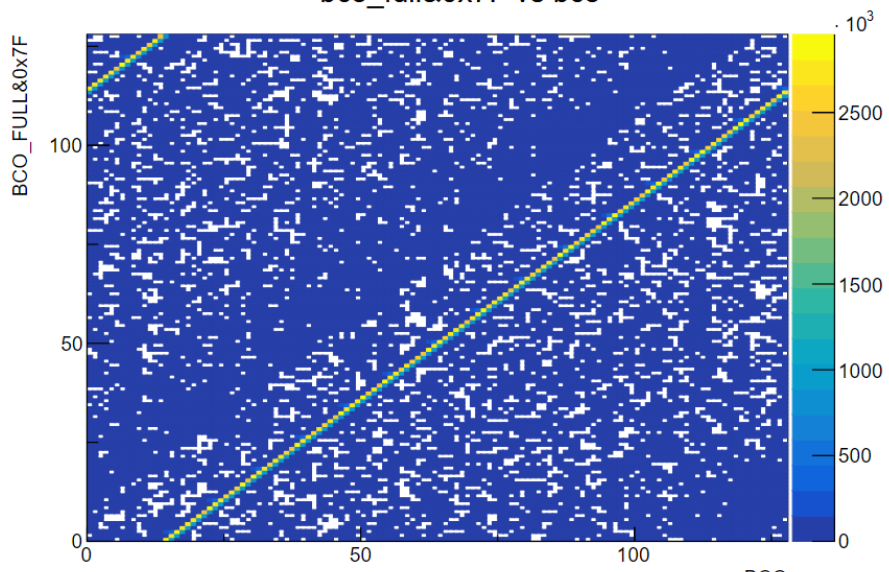
bco\_full\_2prev vs bco 24768



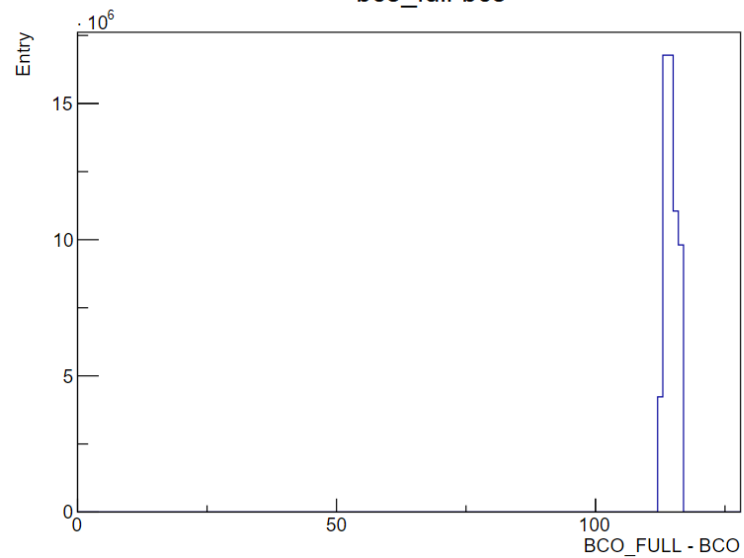
bco\_full\_2prev-bco



bco\_full vs bco

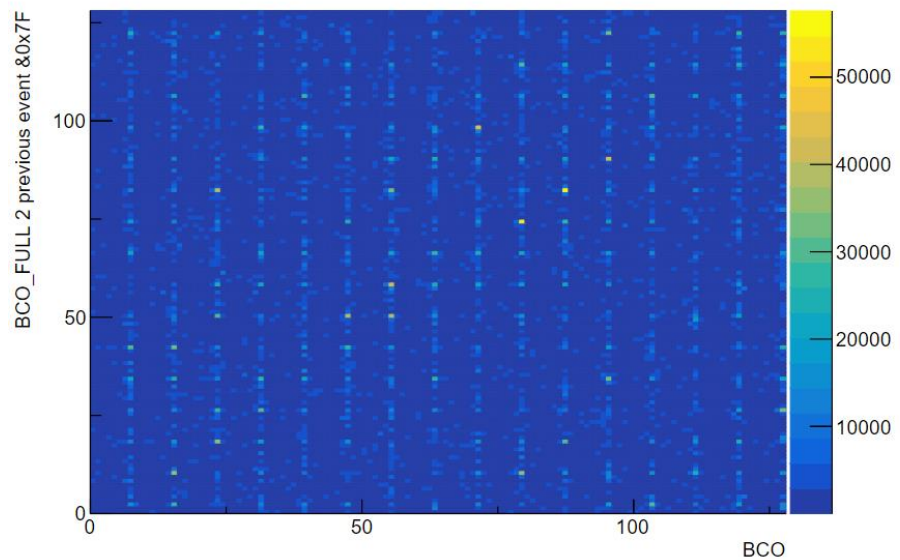


bco\_full-bco

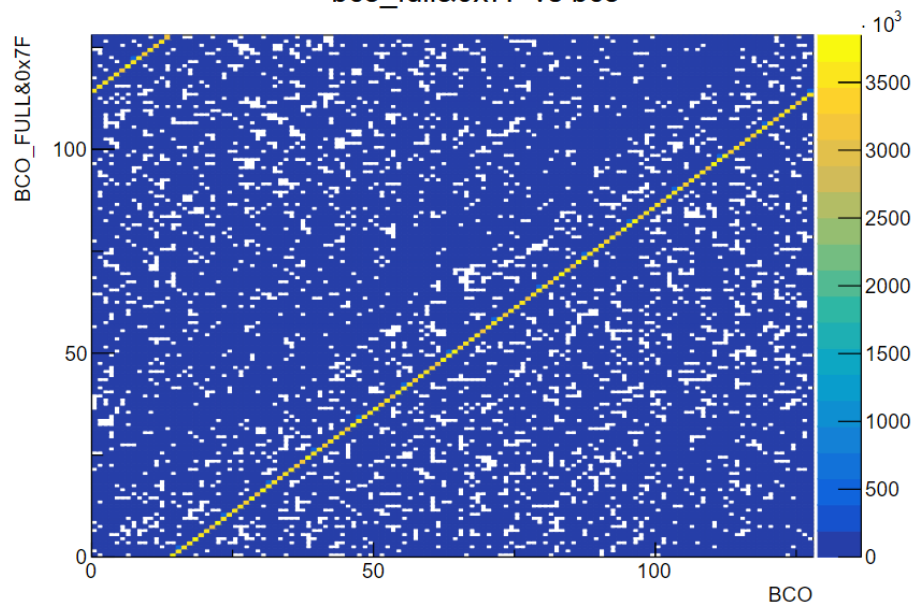


# 2 previous event 24768 intt2

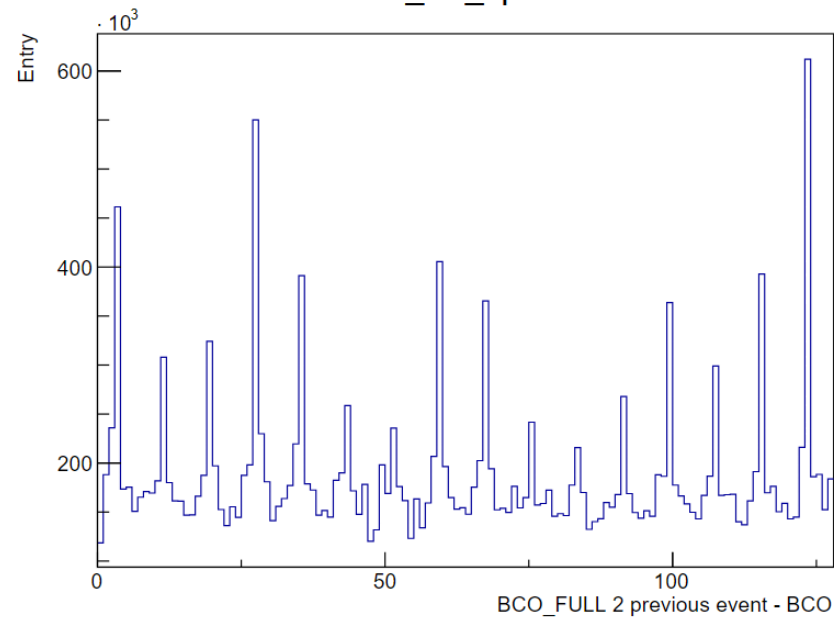
bco\_full&0x7F\_2prev vs bco 24768



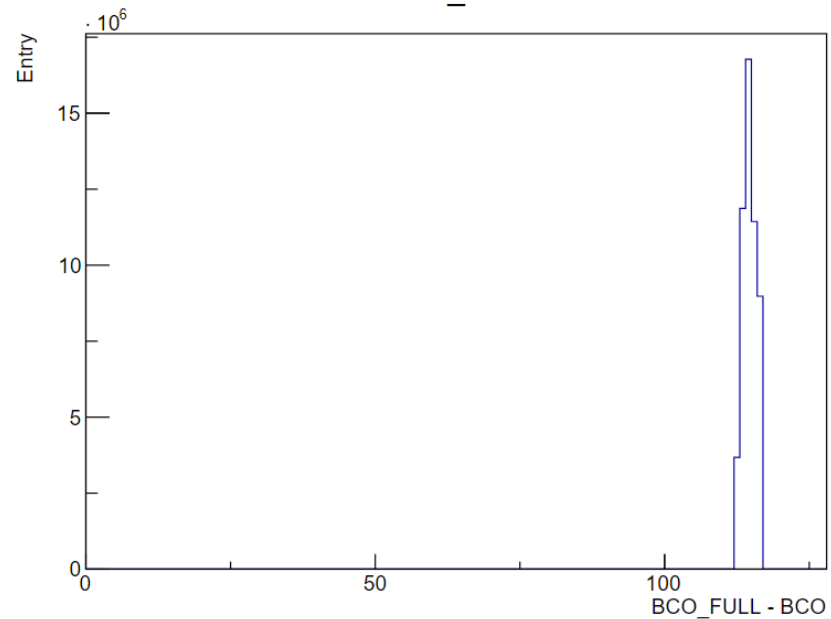
bco\_full&0x7F vs bco



bco\_full\_2prev-bco



bco\_full-bco

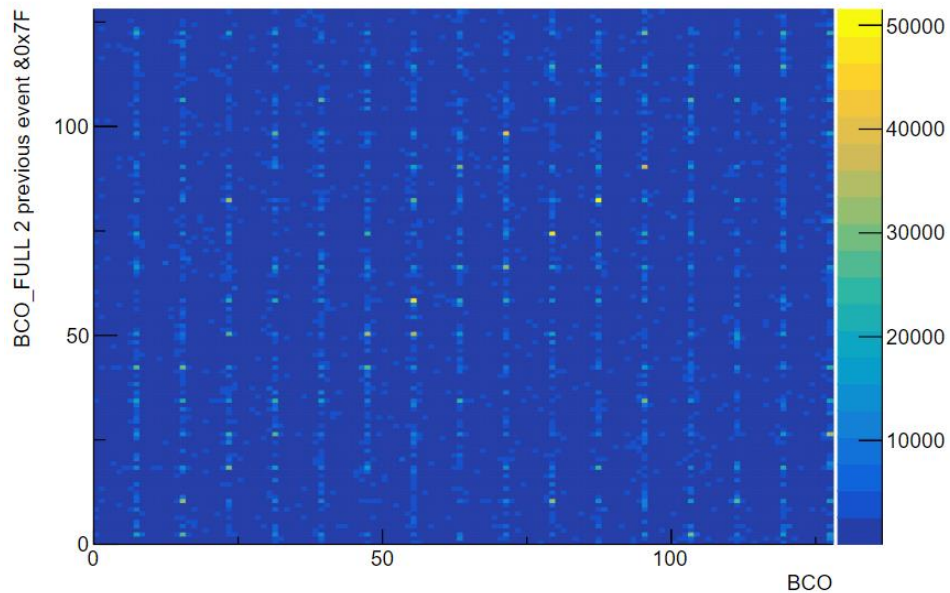


2 previous event 24768 intt3

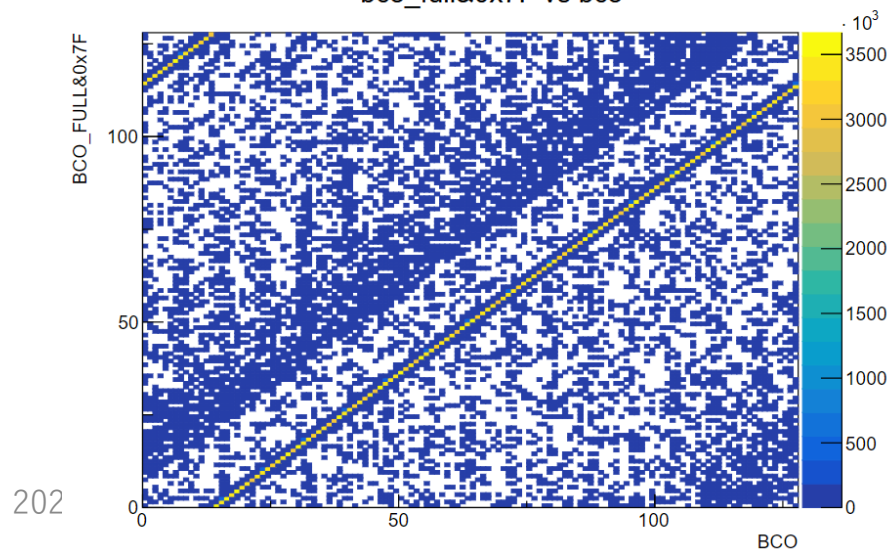
- なし

# 2 previous event 24768 intt4

bco\_full&0x7F\_2prev vs bco 24768

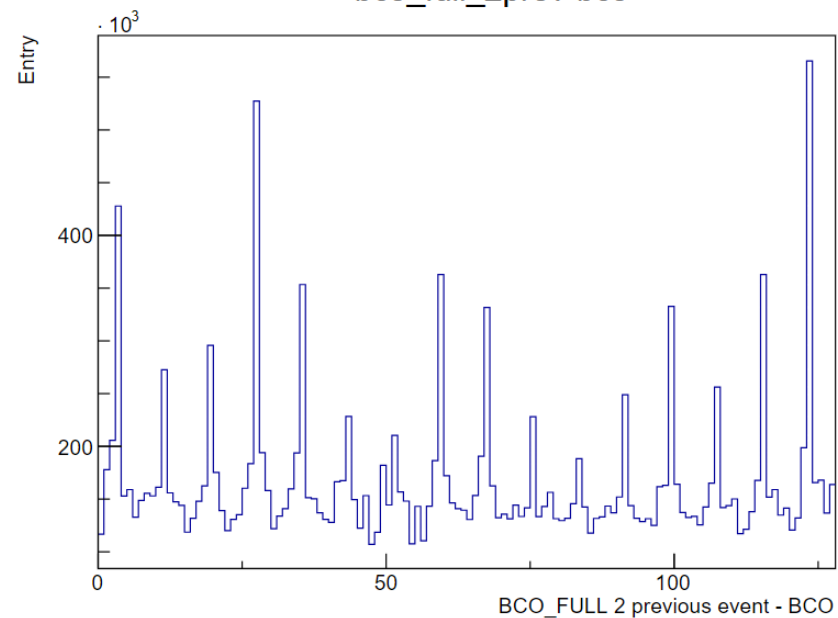


bco\_full&0x7F vs bco

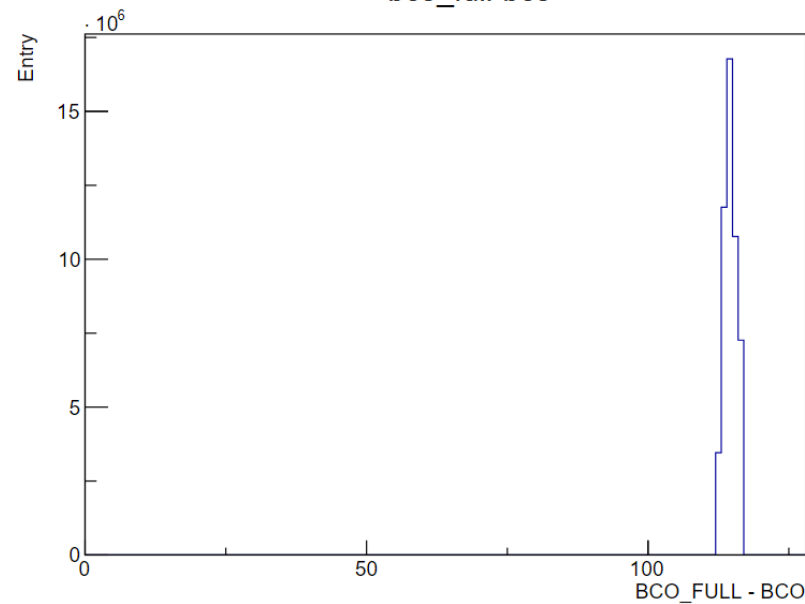


202

bco\_full\_2prev-bco



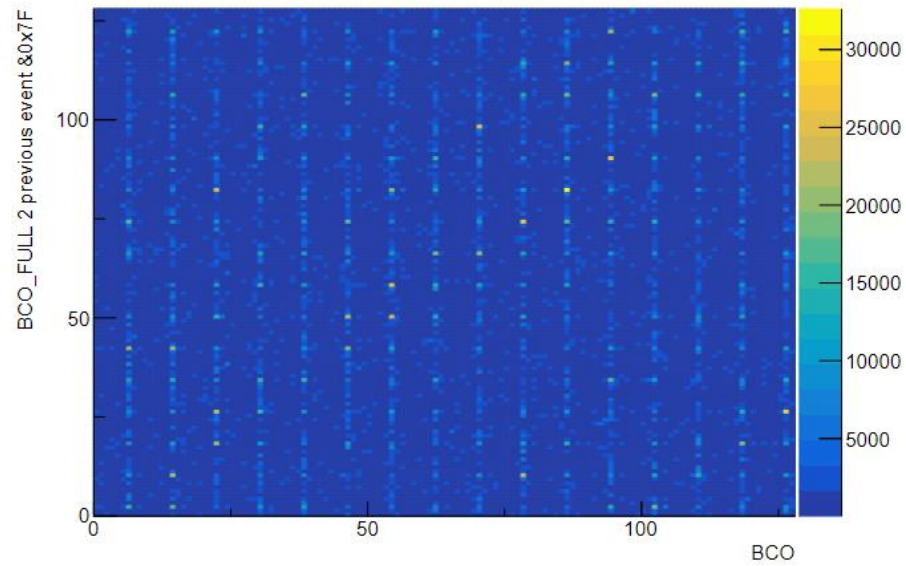
bco\_full-bco



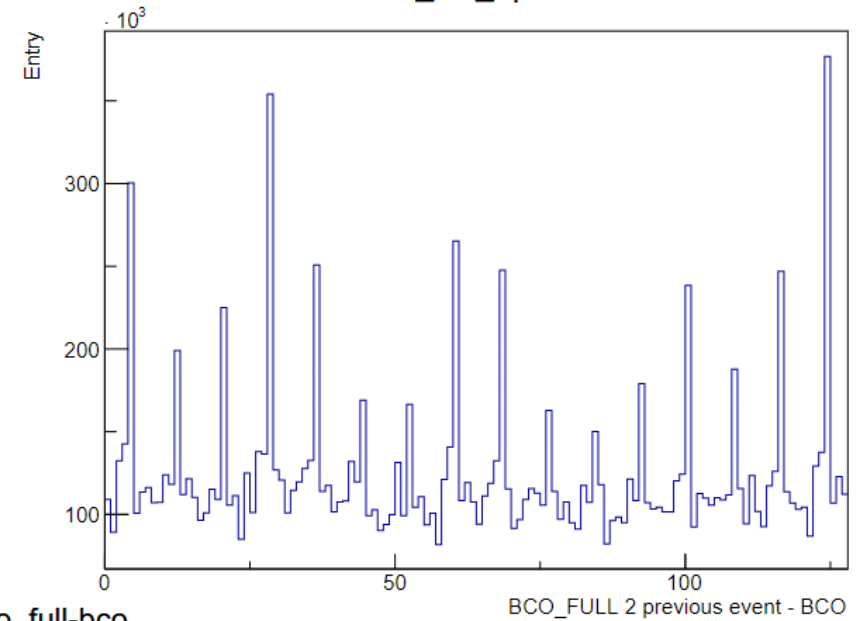


# 2 previous event 24768 intt5

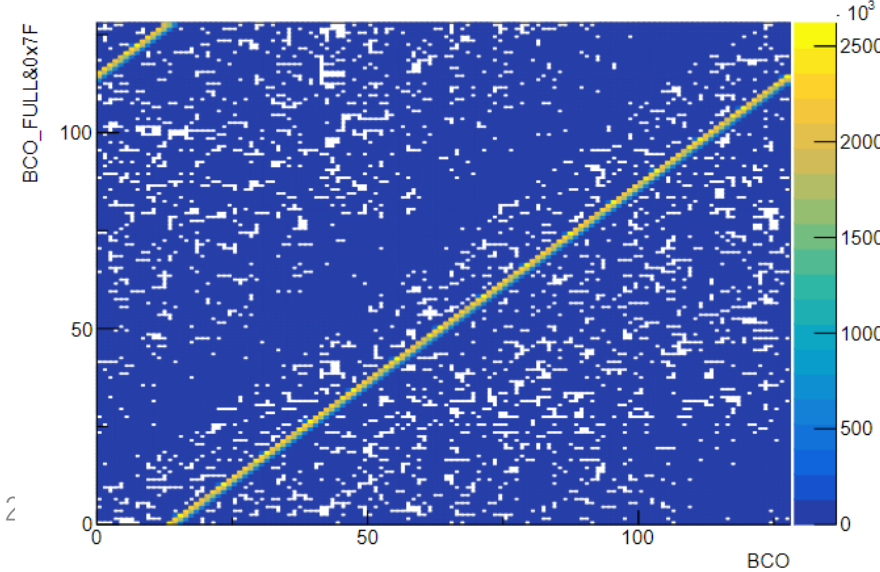
bco\_full\_2prev vs bco 24768



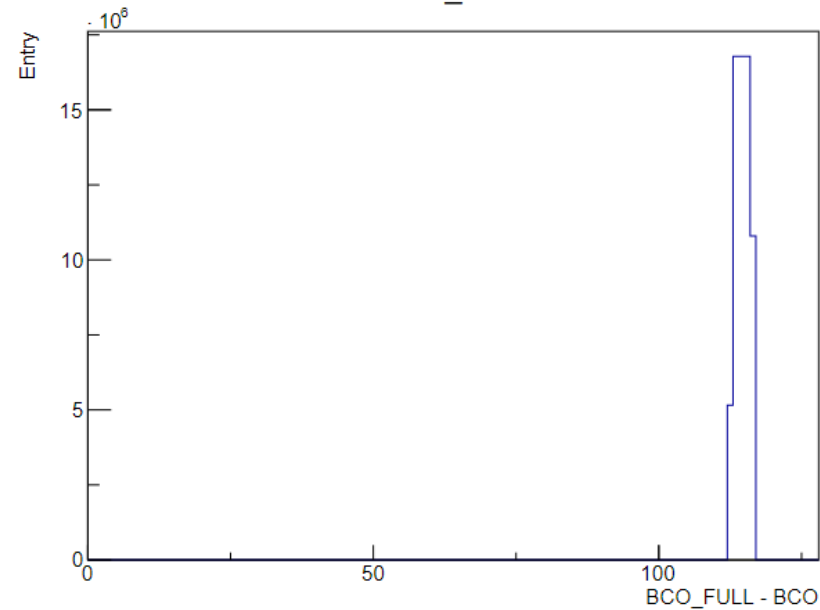
bco\_full\_2prev-bco



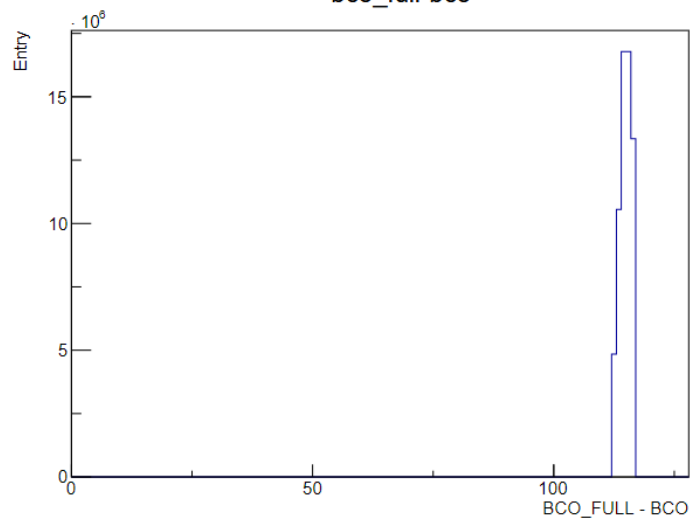
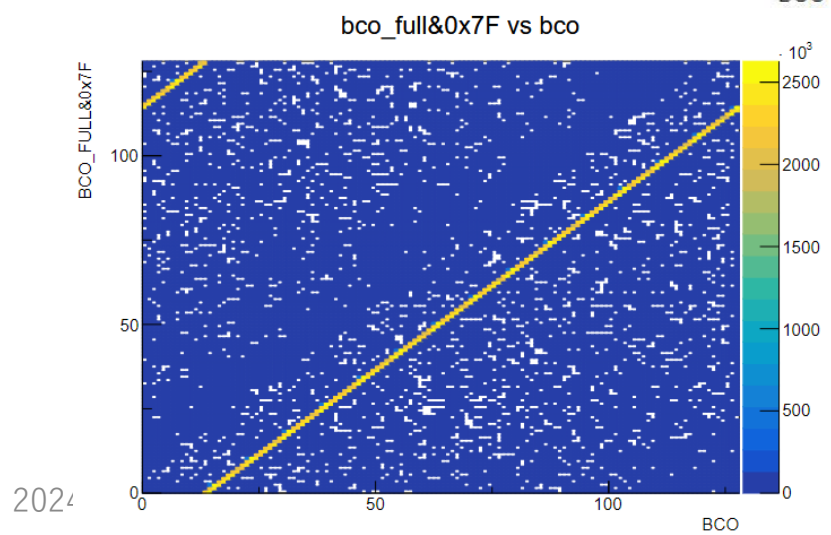
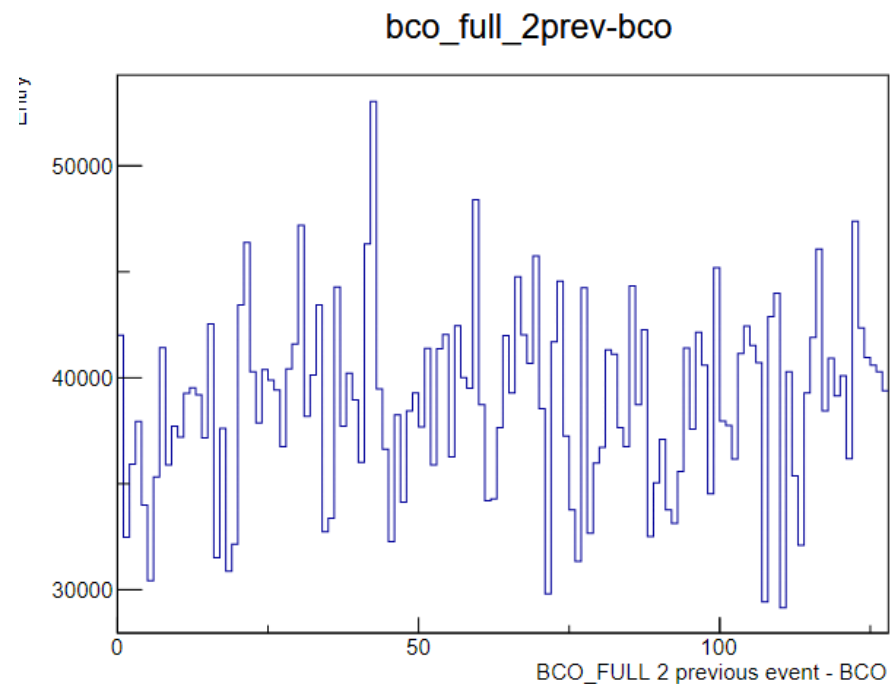
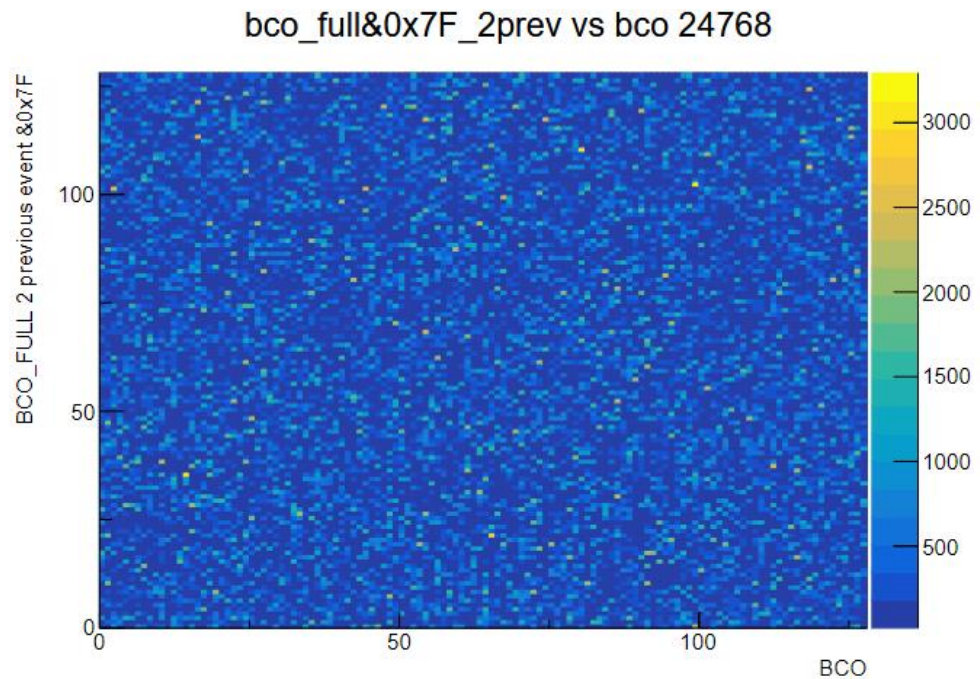
bco\_full vs bco



bco\_full-bco

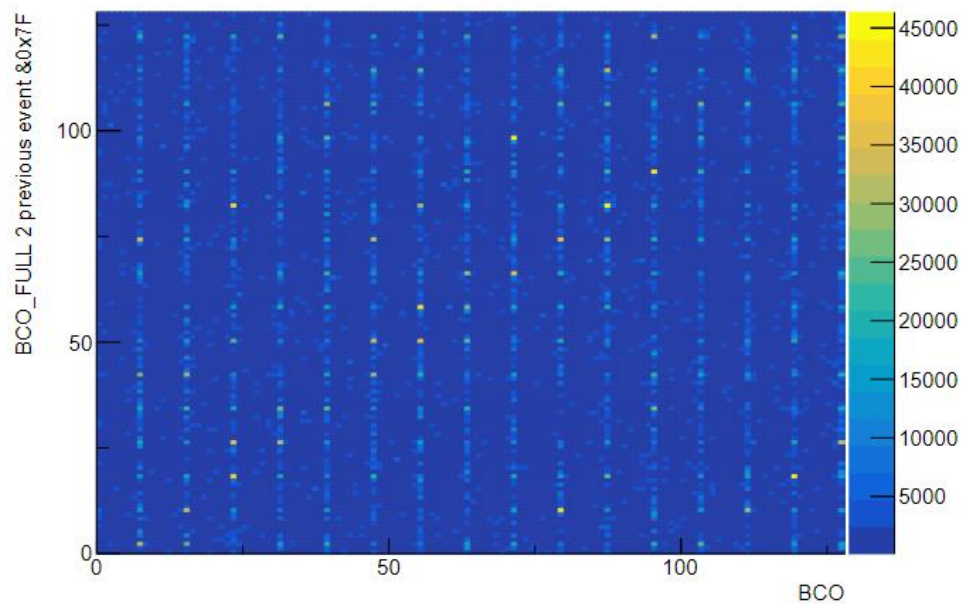


# 2 previous event 24768 intt6

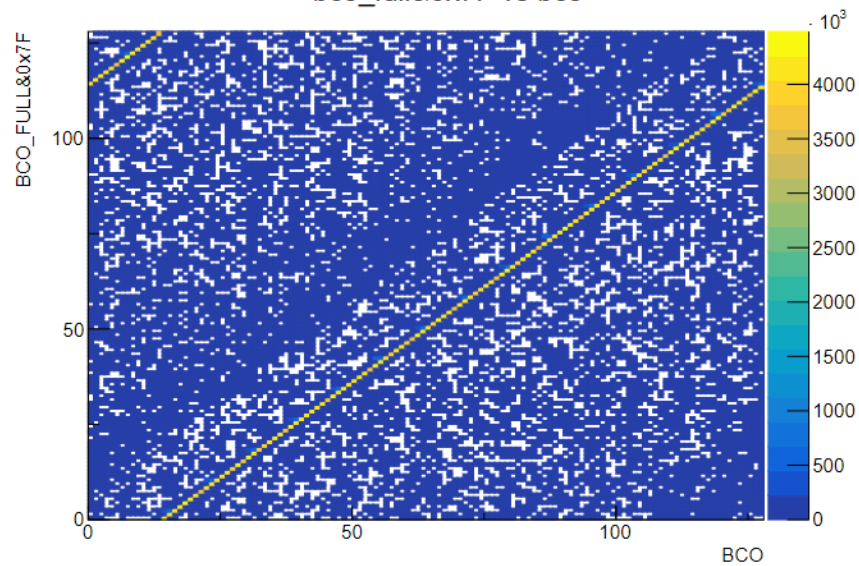


# 2 previous event 24768 intt7

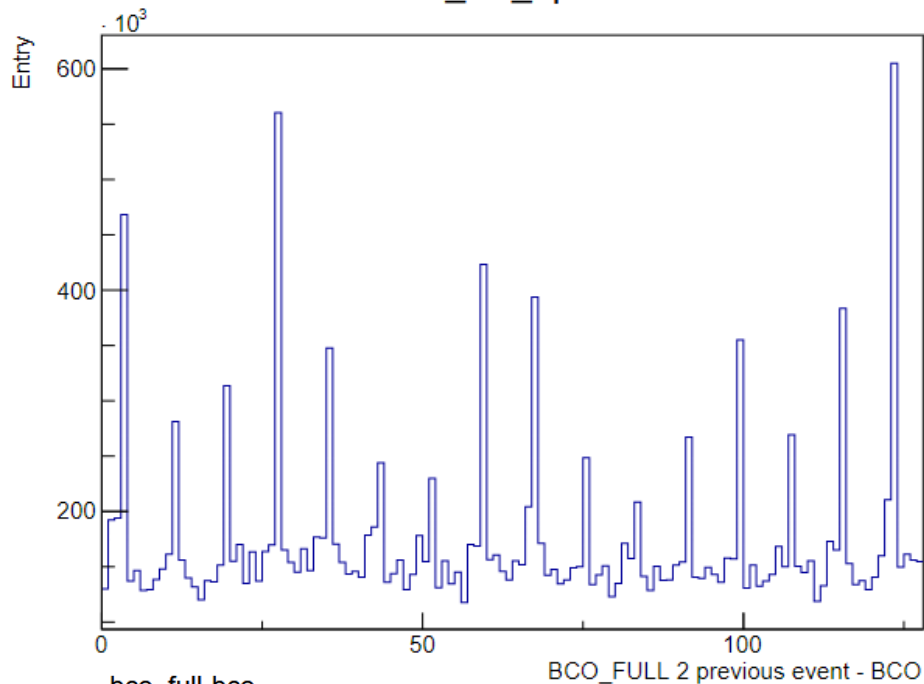
bco\_full\_2prev vs bco 24768



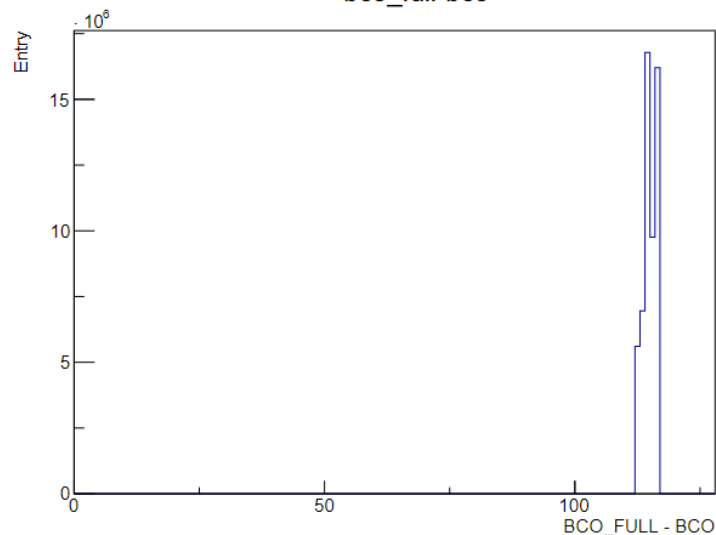
bco\_full vs bco



bco\_full\_2prev-bco

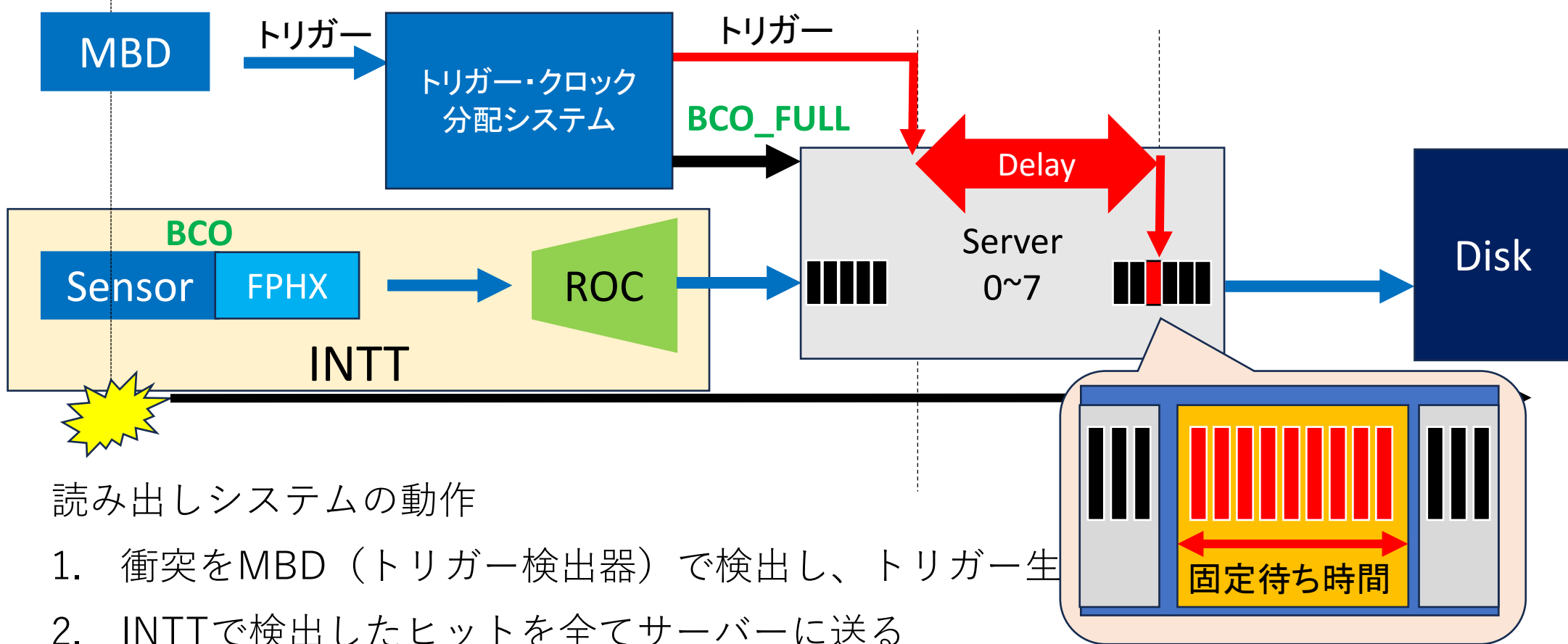


bco\_full-bco



# INTT のデータ読み出し

BCO(Beam clock counter) : INTTのビームクロック 7bit  
BCO\_FULL : sPHENIXのビームクロック 40bit



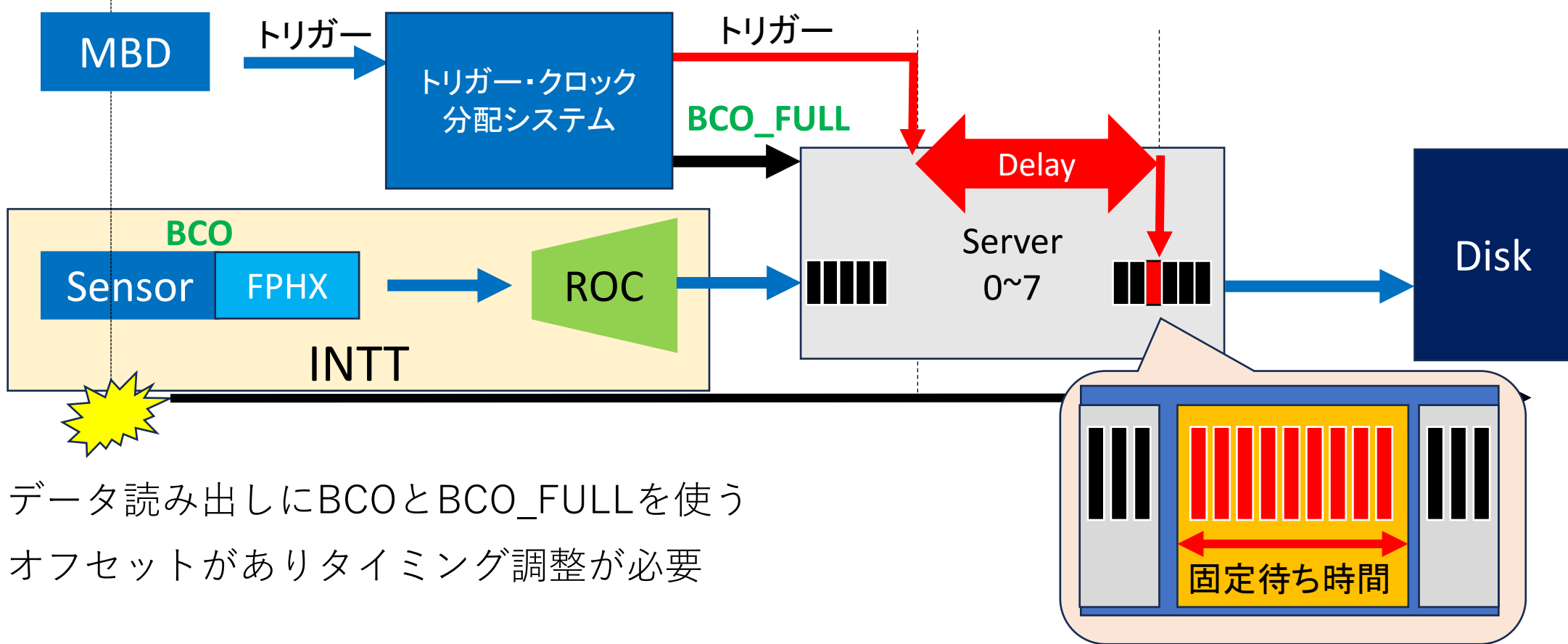
## 読み出しシステムの動作

1. 衝突をMBD (トリガー検出器) で検出し、トリガー発生
2. INTTで検出したヒットを全てサーバーに送る
3. トリガーをサーバーで受信し、そのタイミングのヒットを選びイベントとする
4. Diskにデータを保存

INTTのデータ読み出しはストリーミング法をとっている

# INTT のデータ読み出し

BCO(Beam clock counter) : INTTのビームクロック 7bit  
BCO\_FULL : sPHENIXのビームクロック 40bit



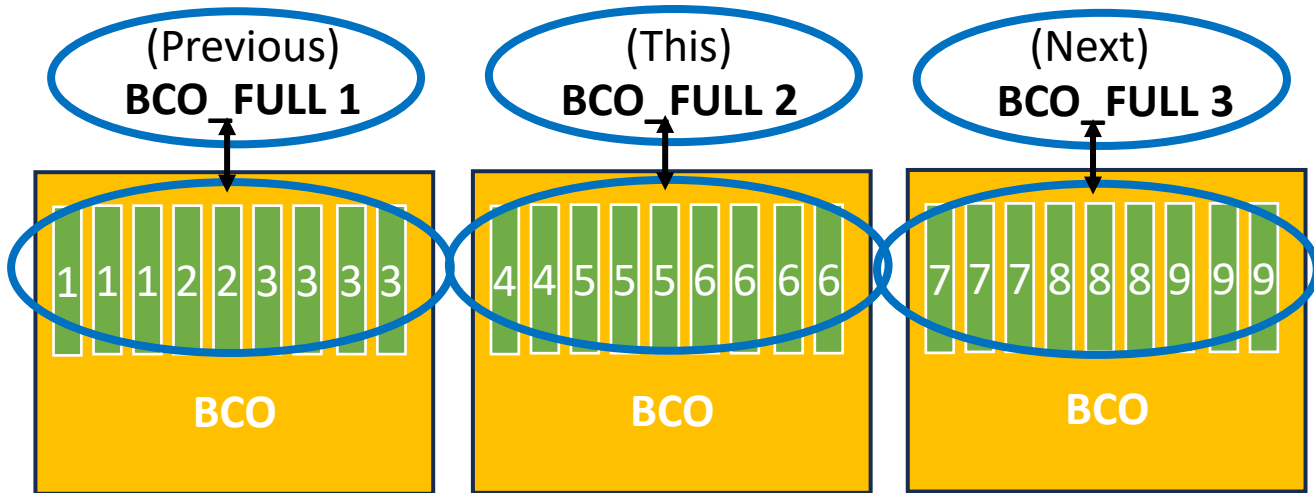
データ読み出しにBCOとBCO\_FULLを使う  
オフセットがありタイミング調整が必要

## タイミング調整

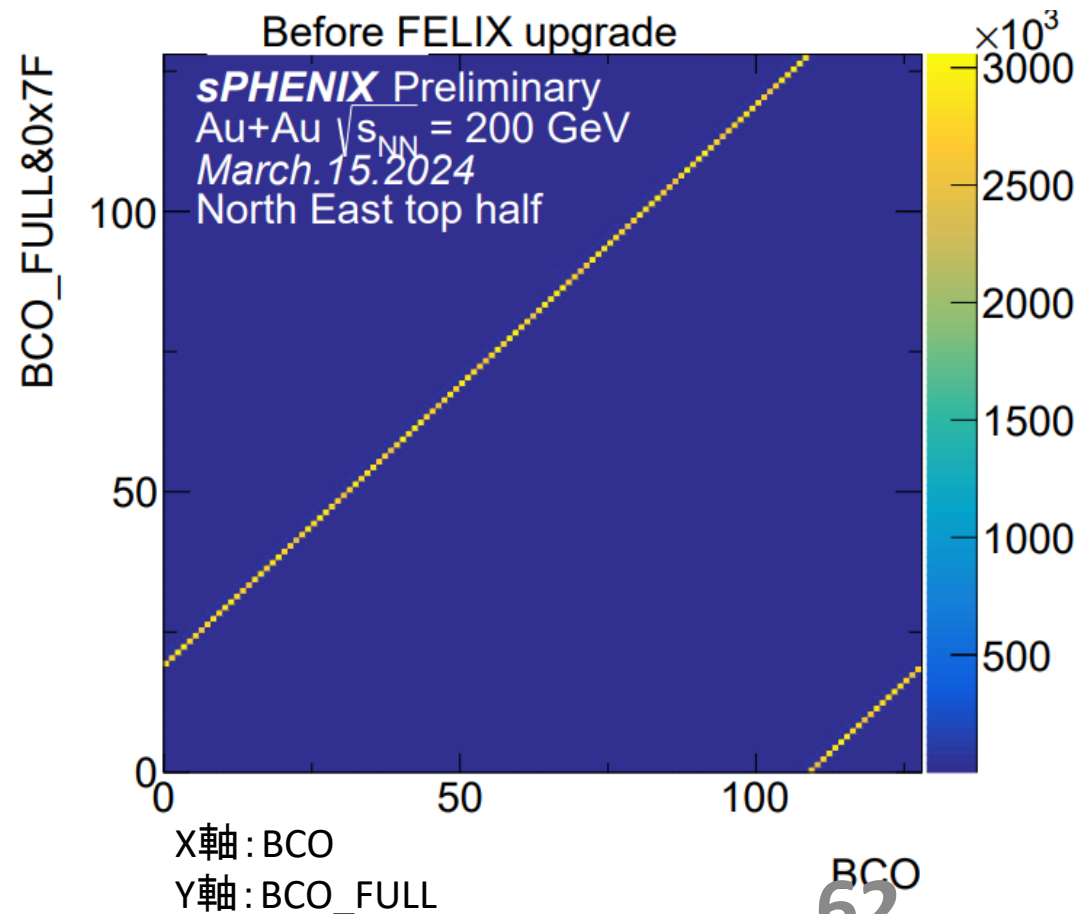
- Delay: TriggerとINTTの読み出しタイミングのずれ補正
- 固定待ち時間長: 同一時刻とみなされるデータ収集時間

# データ読み出しの現状

INTTのヒットデータでは同一の衝突によるヒットのグループを1つのイベントとしている



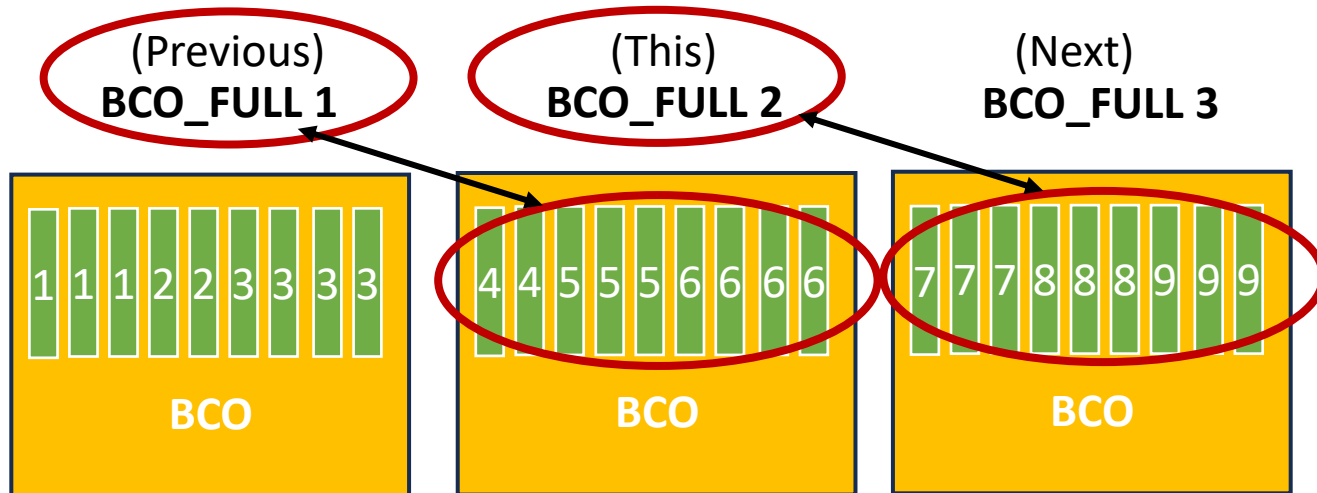
同一のイベント内の  
BCOとBCO\_FULLの相関図



同一のイベント内のBCOとBCO\_FULLの相関図で明確な相関

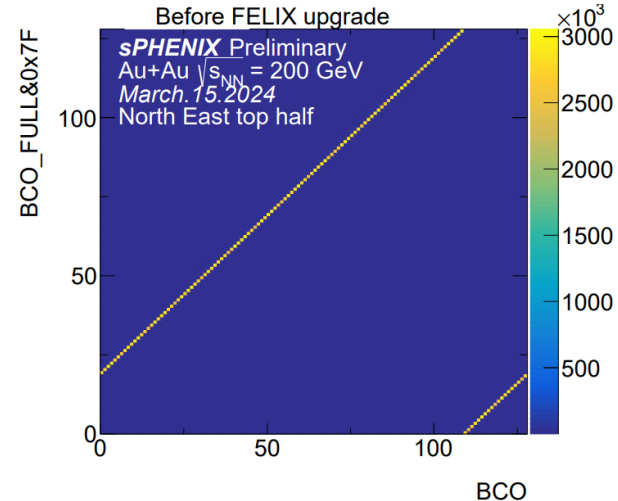
⇒ほぼ単一のビーム交差で各ヒットデータの  
BCOのタグ付けの処理が完了できている

# データ読み出しの現状

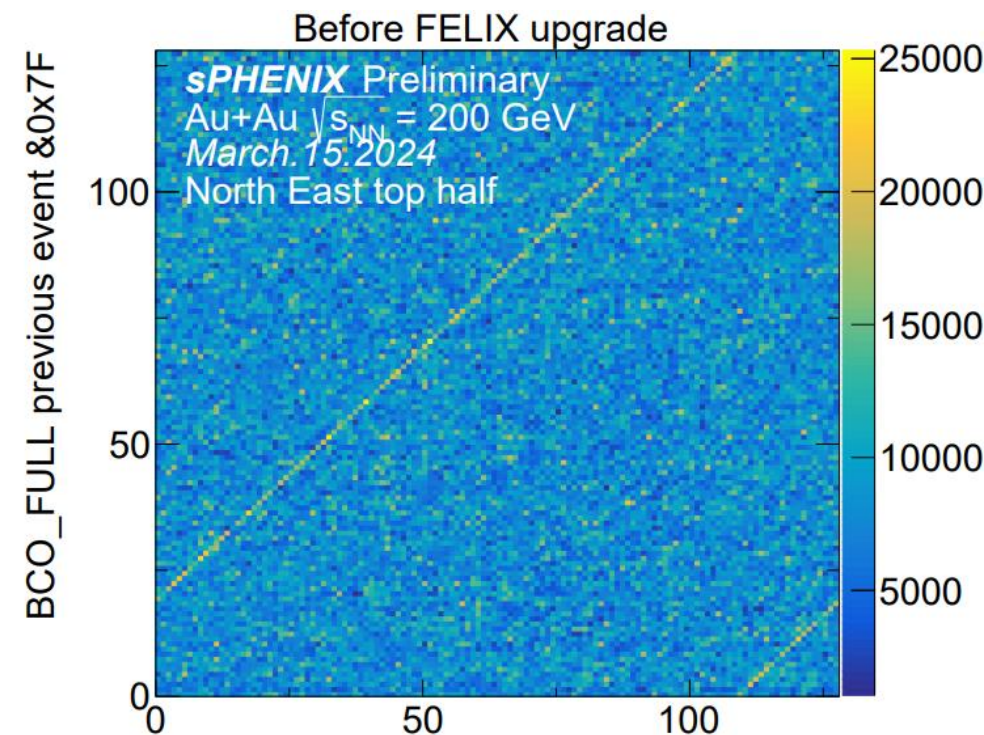


本来関連のない値同士のあるイベントのBCOと  
1つ前のイベントのBCO\_FULLには相関がないはず  
しかし同一イベント内の相関図と同じ位置に相関を確認

⇒1つ前のイベントのヒットが現在のイベントに混在



BCOと1つ前のイベントのBCO\_FULLの相関図

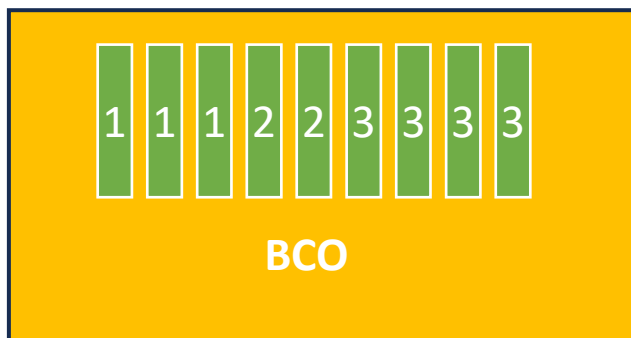


X軸: BCO  
Y軸: 1つ前のイベントのBCO\_FULL

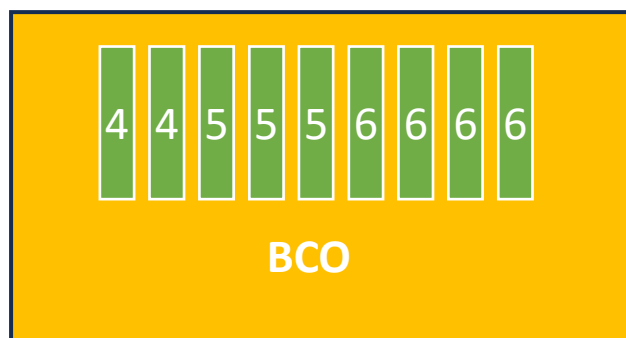
# 研究目的

正常な状態

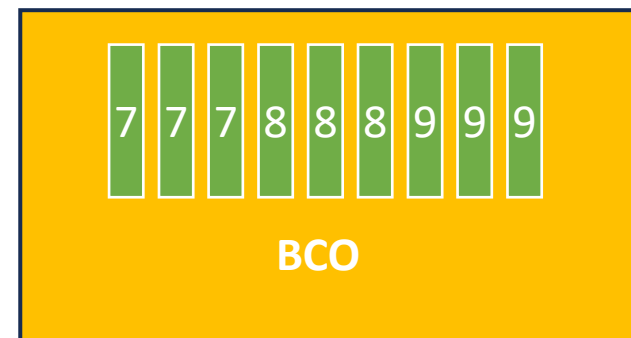
BCO\_FULL 1



BCO\_FULL 2



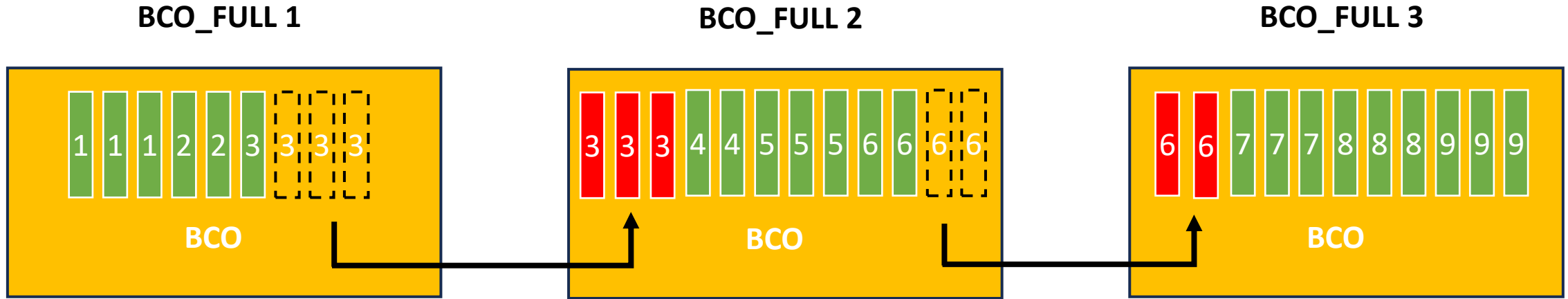
BCO\_FULL 3





# 研究目的

実際起きている状態



2つの異なるビーム衝突事象のデータが混ざってしまう現象が起きている

この現象をEvent Mixupと呼ぶこととする

Event MixupはINTTの性能を悪化させるため解決が必須

本研究では解決のため原因を調べた

# Mixupヒットの判定方法

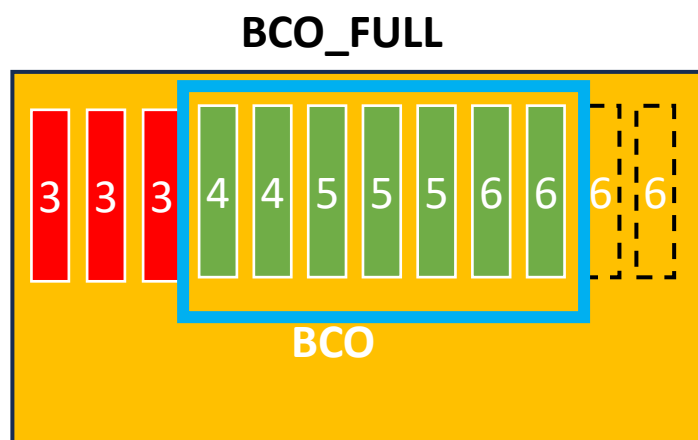
- BCO\_FULL - BCO を用いて判定

① BCO\_FULL (現在) - BCO (現在)

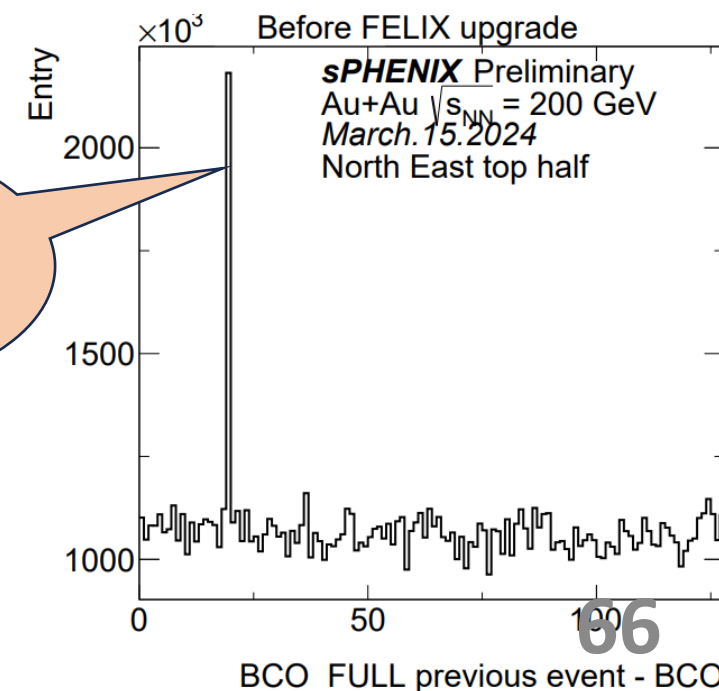
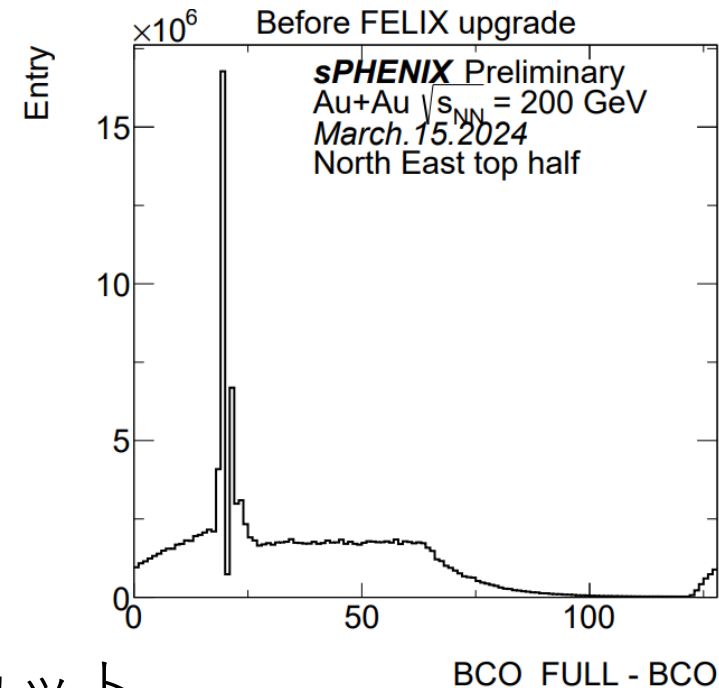
② BCO\_FULL (前) - BCO (現在)

同じ位置にあるピークをMixupと定義

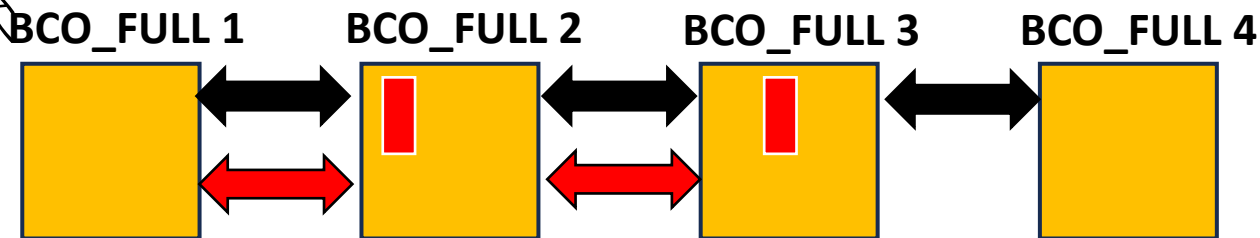
- ②に含まれる現在のイベントの衝突によるヒットをカット



Event Mixup



# ビーム衝突間隔との関係



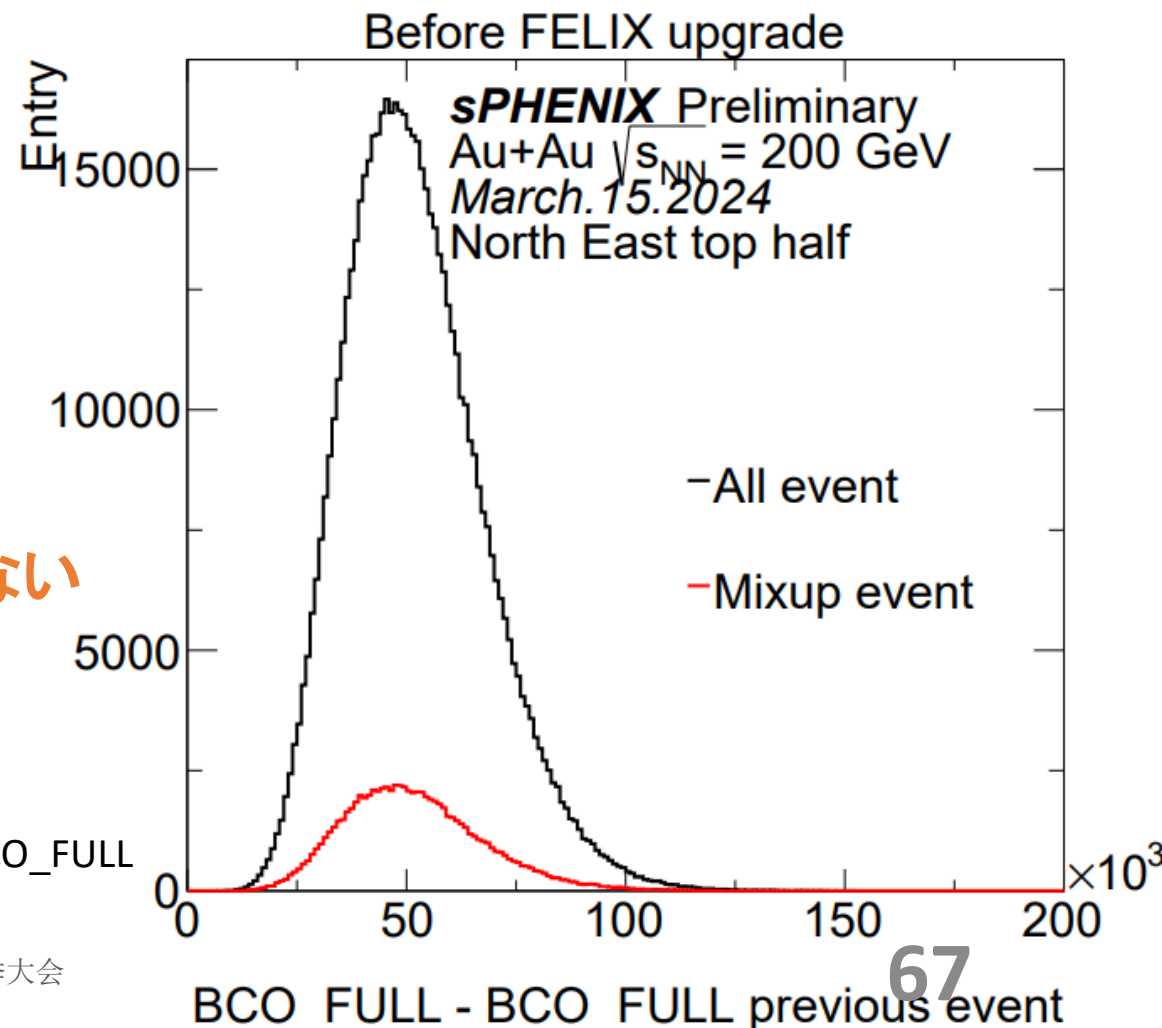
ビーム衝突の間隔との関係を調べた

右図はBCO\_FULL(現在) - BCO\_FULL(前)  
黒は全イベント、赤はMixupのヒットを持つイベント

結果: 全イベントの時とMixupのヒットを持つイベントの  
ビーム衝突の間隔の分布は特に変化なし

⇒ Event Mixupとビーム衝突の間隔に関係性はない

X軸: BCO\_FULL - 1つ前のBCO\_FULL  
Y軸: イベント数

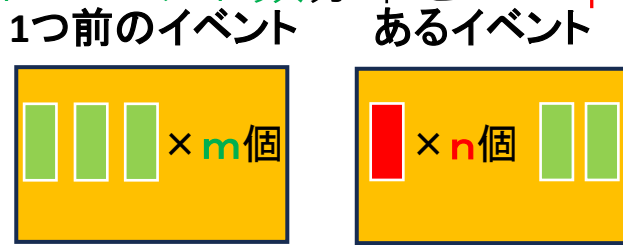


# 1つ前のイベントのヒット数との関係

X軸: 前のイベント内のヒット数  
Y軸: イベント数

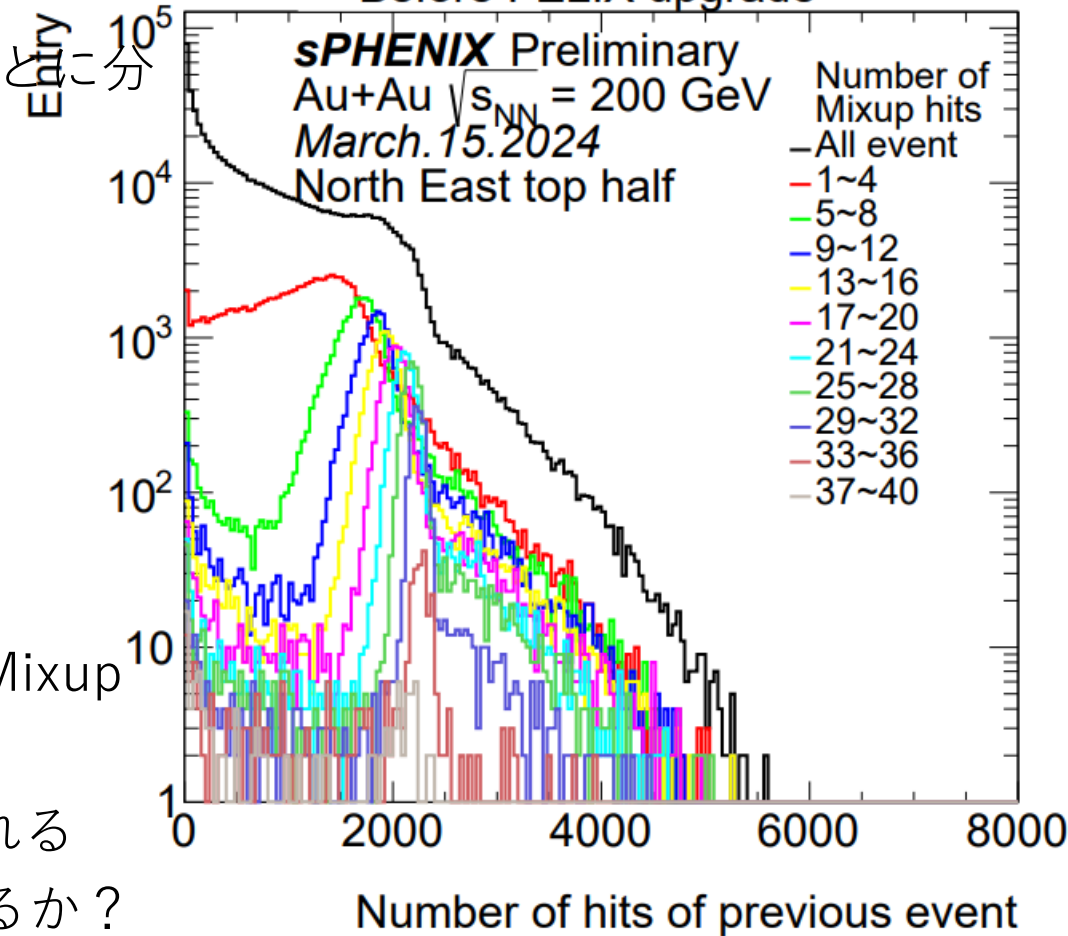
Mixupヒットがあるときの1つ前のイベントのヒット数の関係を調べる

右図はイベントのヒット数分布をMixupヒット数 (1~40) ごとに分けた



結果

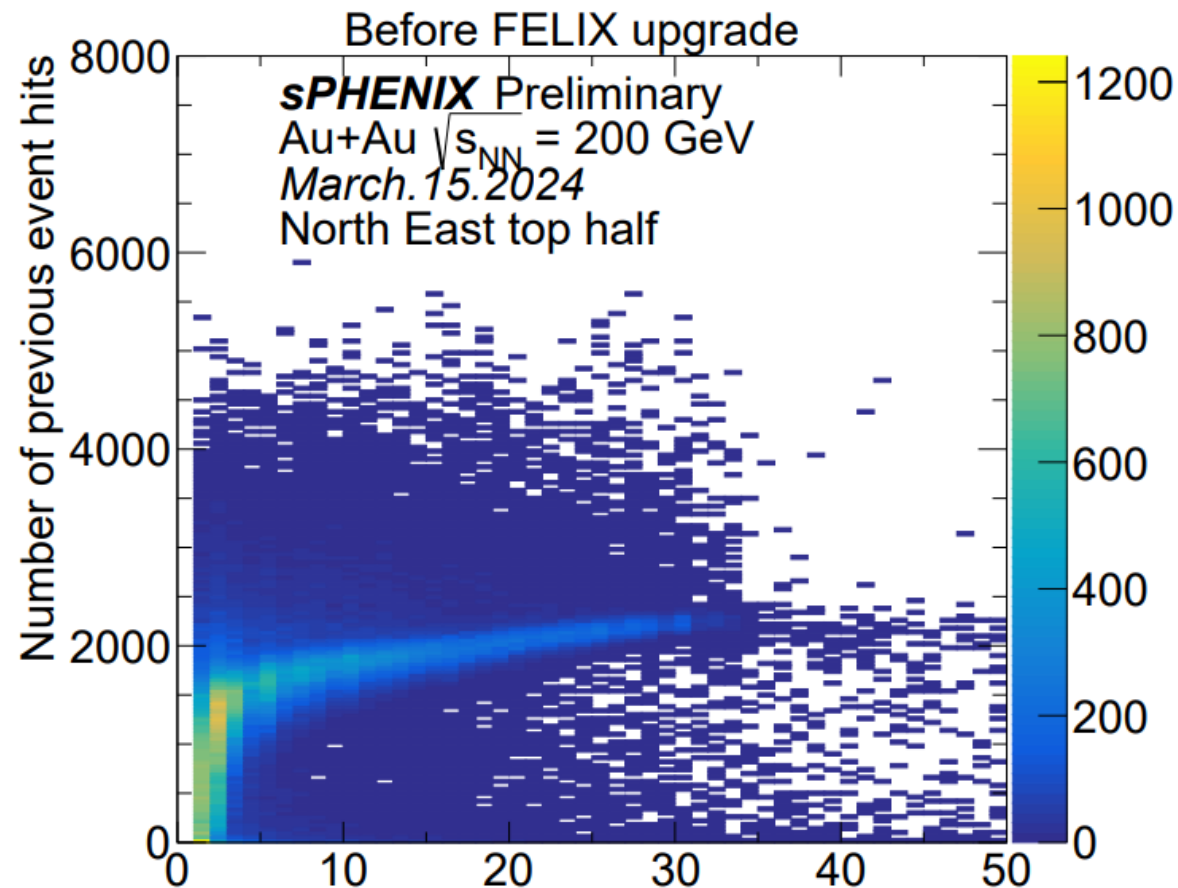
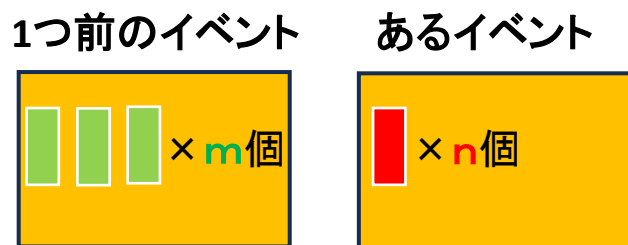
- Mixupヒットの元のイベント (1つ前のイベント) は1500以上のヒット数を持っているものが多い  
⇒ 1 イベントの内ヒット数がおおよそ1500以上でEvent Mixupは起こりやすい
- Mixupのヒット数が増えるとピークの位置も大きい方にずれる  
⇒ Mixupのヒット数とイベント内のヒット数に相関があるか？



# Mixupヒットの数と1つ前のイベントのヒット数の相関関係

前頁の結果から相関関係を調べた  
右図はMixupヒットと1つ前のイベントのヒット数の相関図

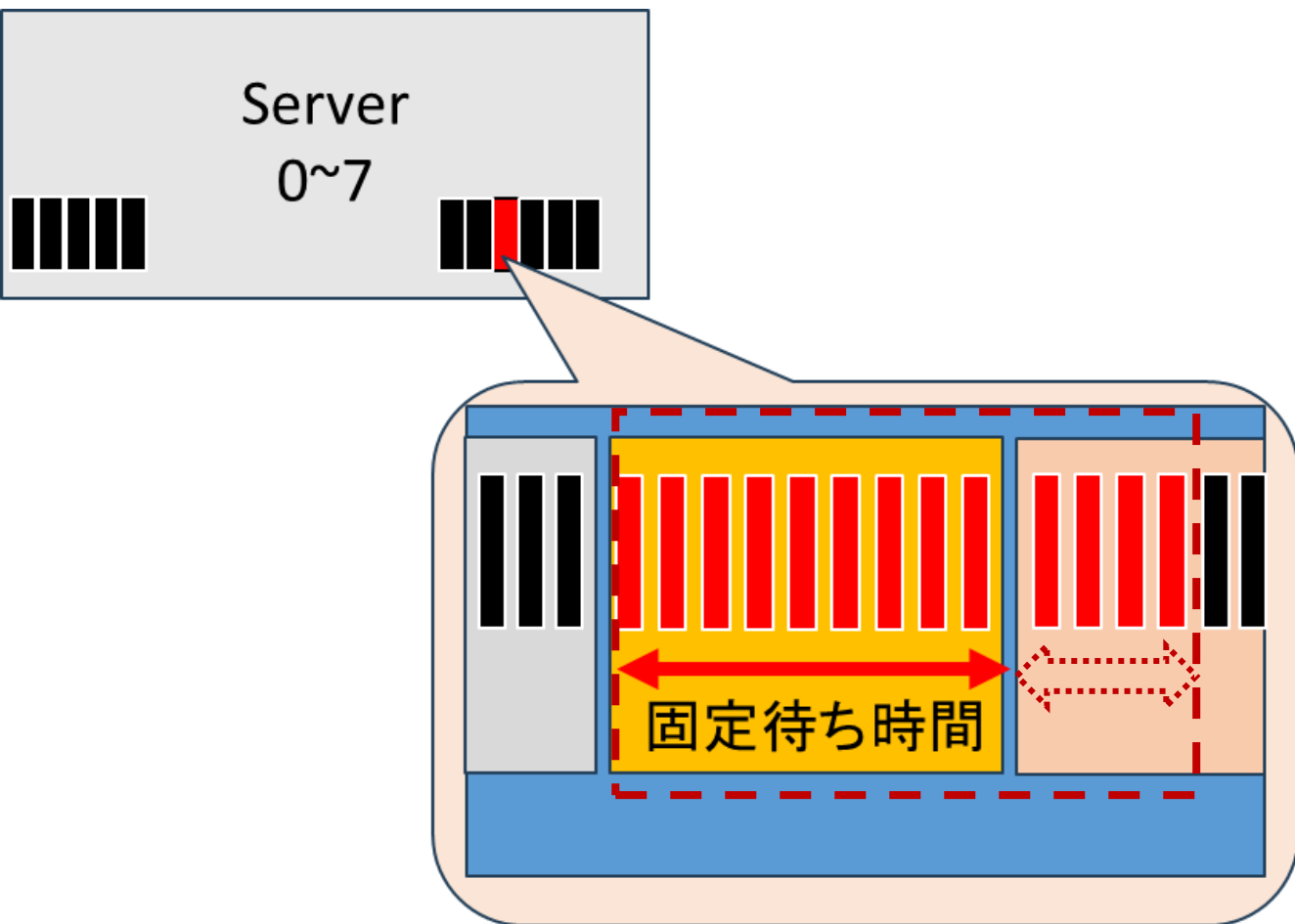
1つ前のイベントのヒット数が1500以上の範囲で  
これらに相関関係があることを確認



1500以上で1つ前のイベントのヒット数が多いほど  
Mixupヒット数は増加する

Number of Mixup hits  
X軸: Mixupヒット数  
Y軸: 前のイベント内のヒット数

## 考えられる原因

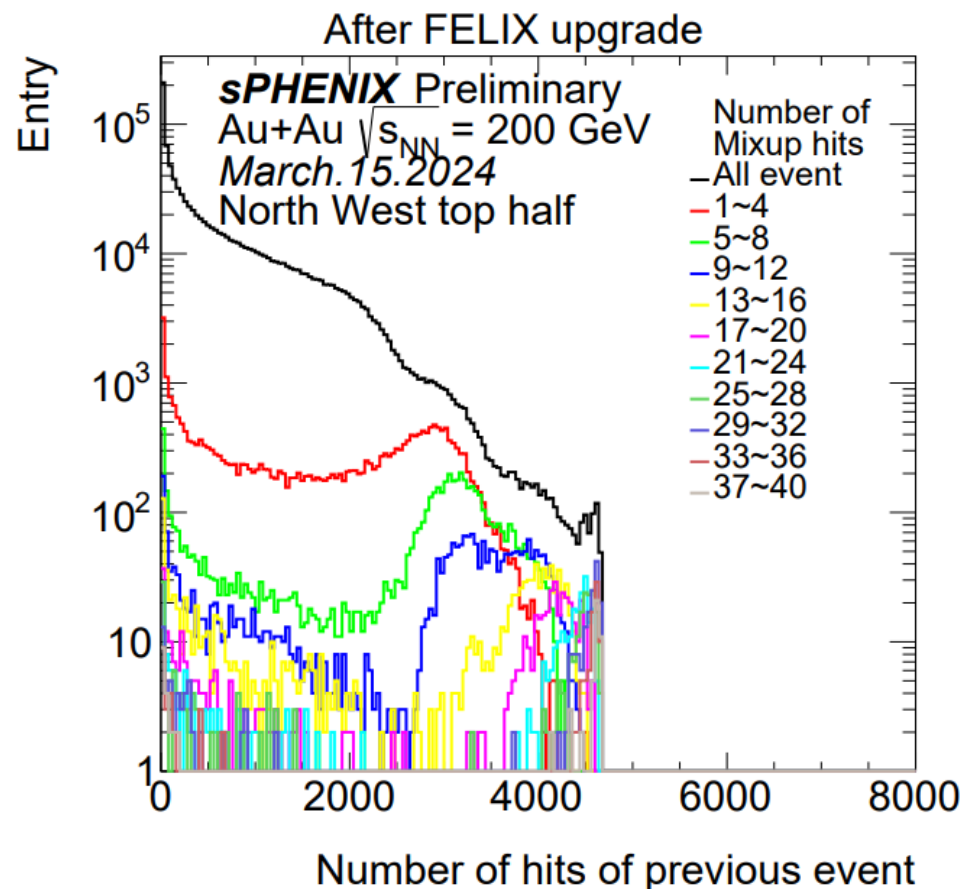
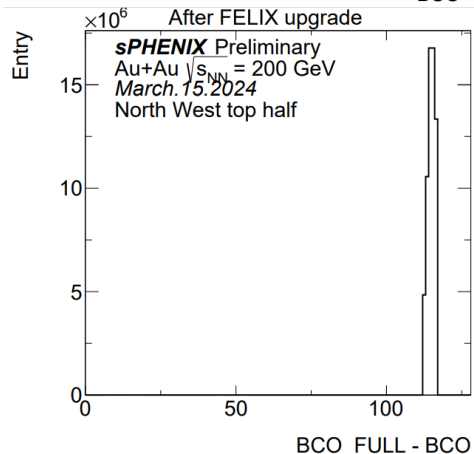
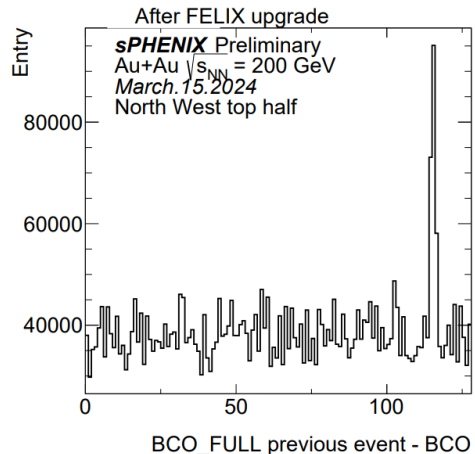
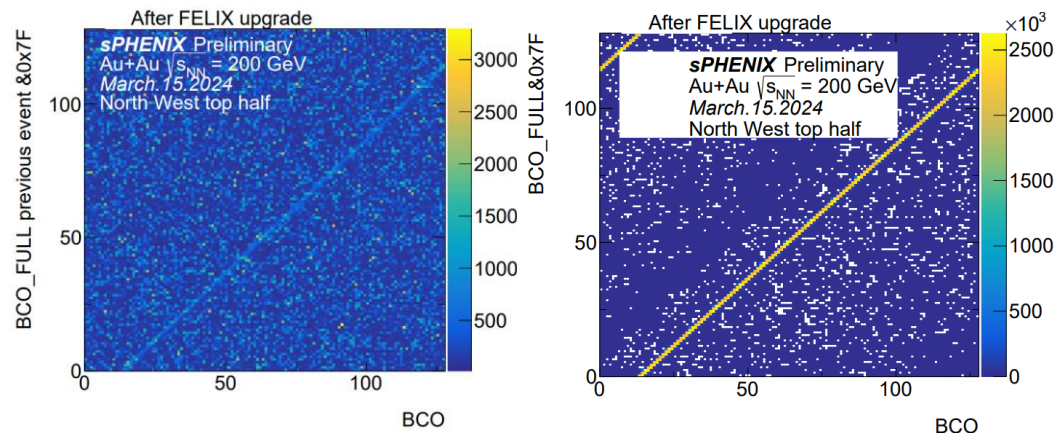


- **固定待ち時間**を超える量のヒットが来ることによってデータが漏れ出していることが考えられる
- 解析結果より**約1500以上**のヒット数で**固定待ち時間**を超える
- よってこの現象はServerでのデータ処理において発生していると考える

# サーバーアップグレード後の変化

ServerのファームウェアのアップグレードがRun2023の後半に行われた  
前頁までの結果はアップグレード前のデータを使用

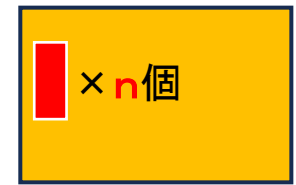
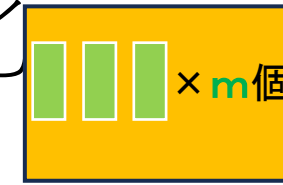
アップグレード後のデータ解析結果、Event Mixupが起きていることが確認されたが分布に変化が見られた



# サーバーアップグレード後の変化

1つ前のイベント

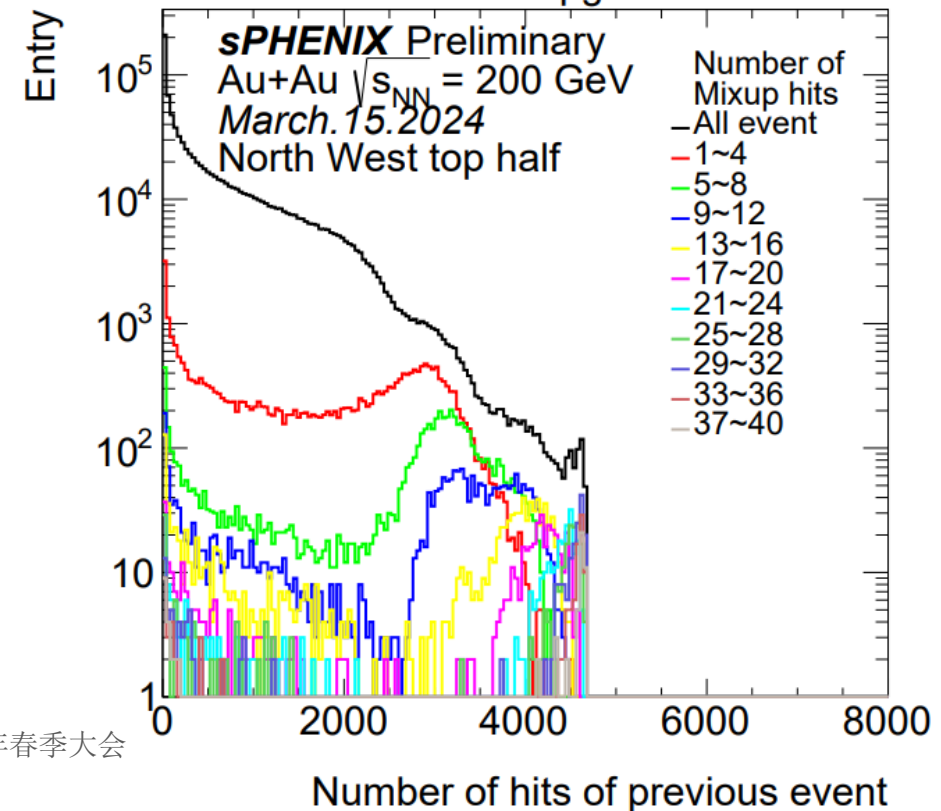
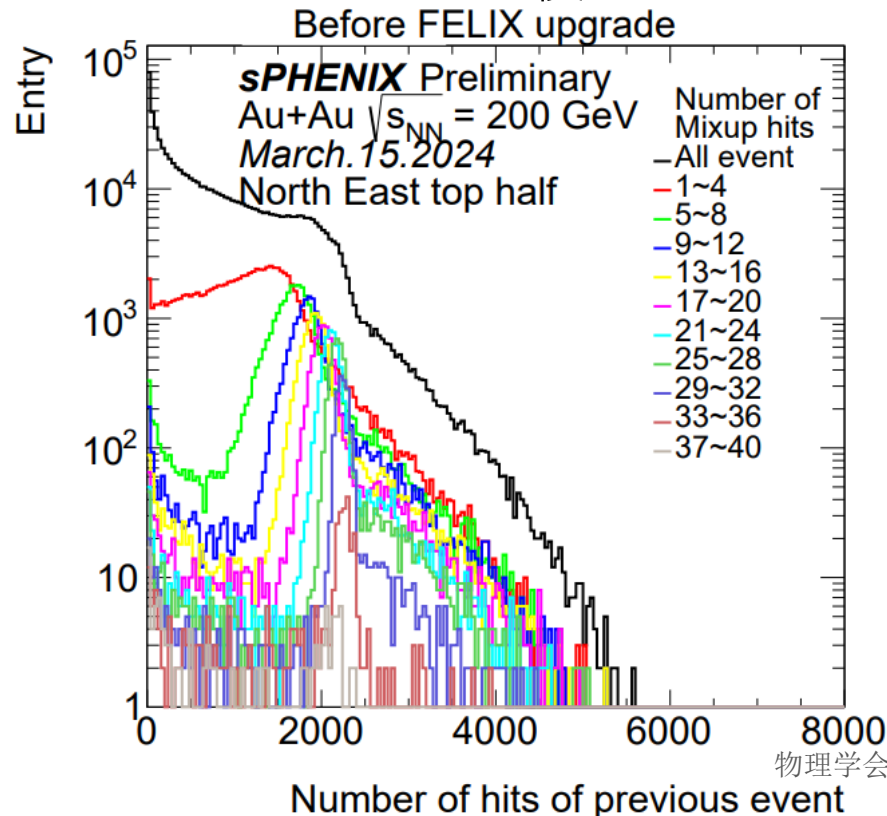
あるイベント



イベントのヒット数分布をMixupヒット数ごとに分けた分布において、ピーク的位置は大きい側にシフト

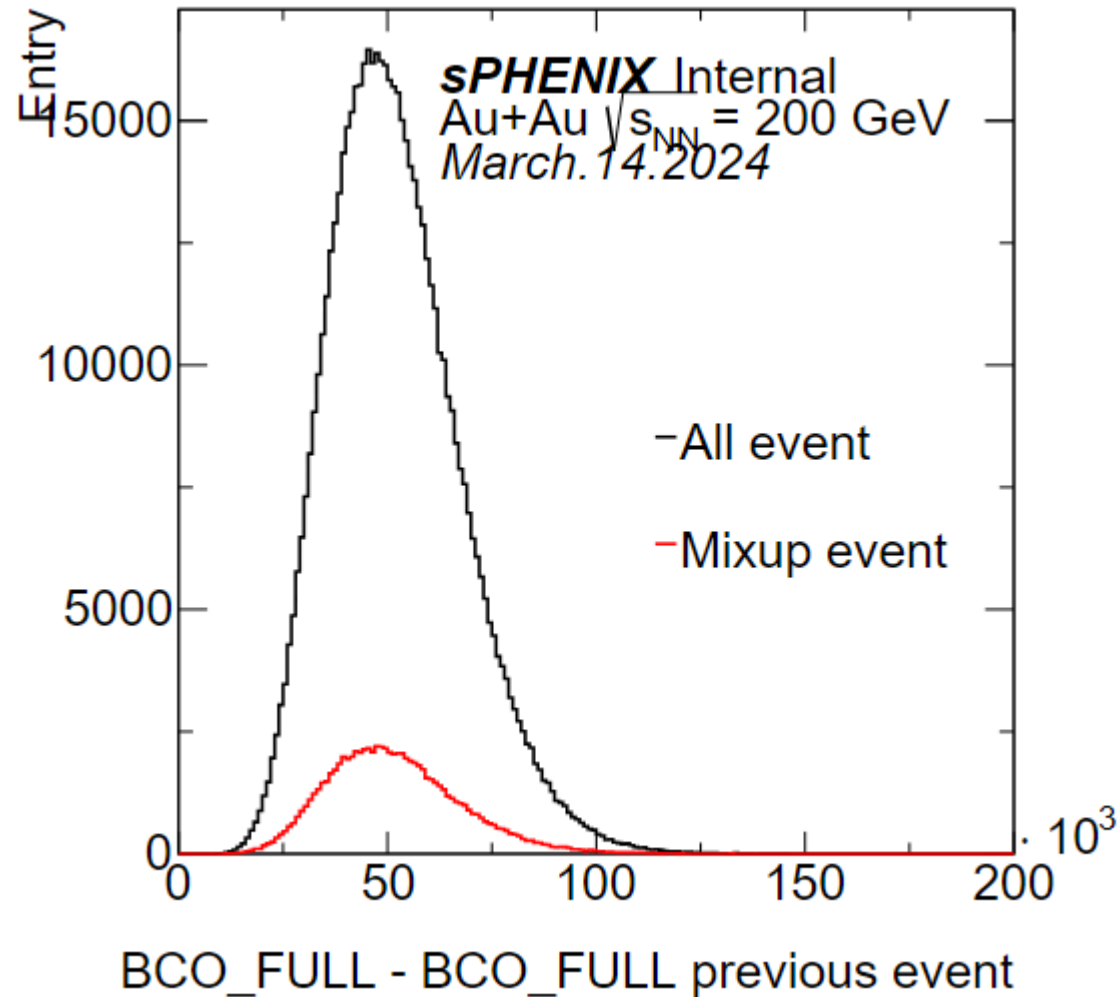
Mixupが発生しやすい1イベントの内ヒット数は1500以上から3000以上に変化した

アップグレード後にEvent Mixupが発生しにくくなったことを示唆





# Collision interval All event and Mixup event (Before Felix upgrade)



- To examine collision interval dependence, The interval distribution of the Mixup Event and the previous Event was checked.
- Results showed no change in peak position between Mixed-up events and all events. Mixed-up does not change with interval width.
- This suggests that no correlation between Event Mixup and collision interval.