

PPSTB meeting

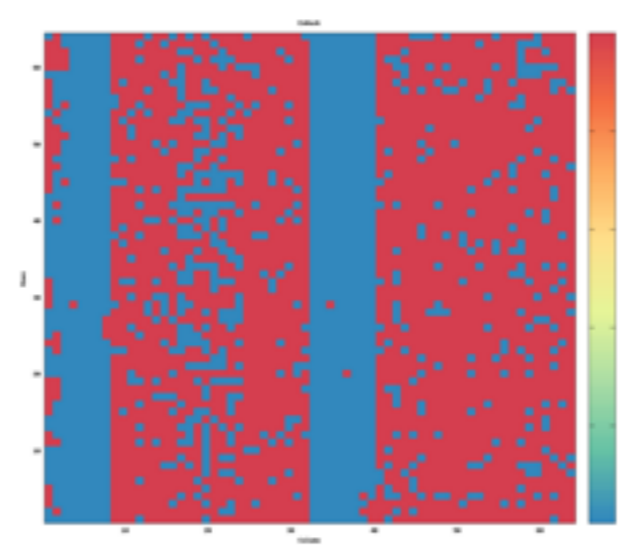
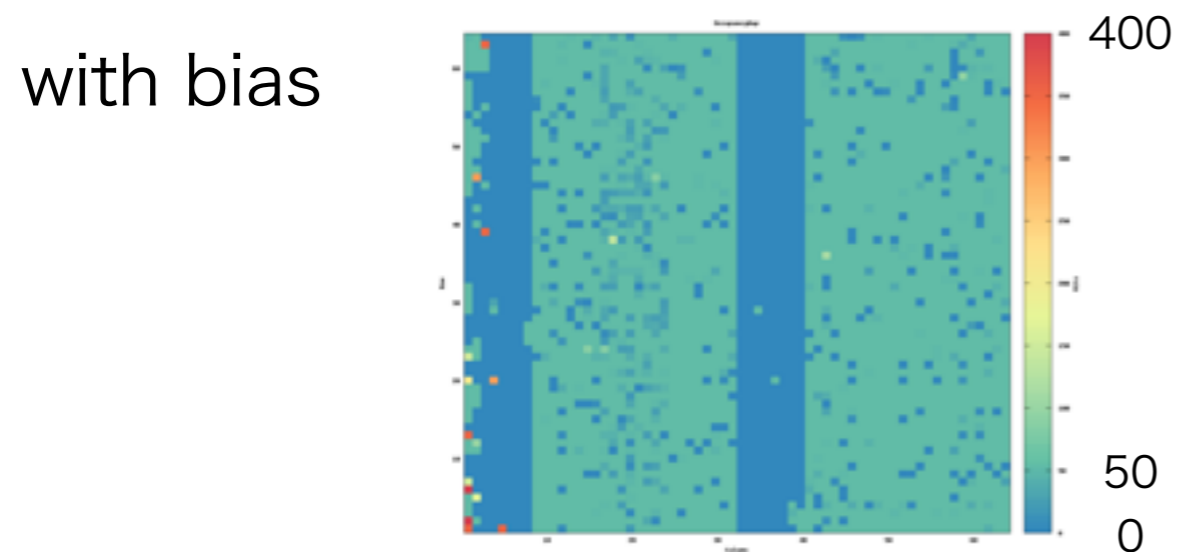
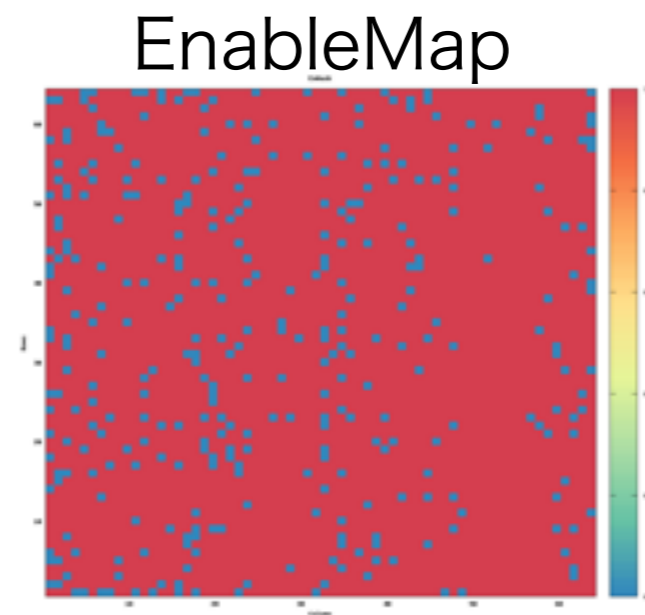
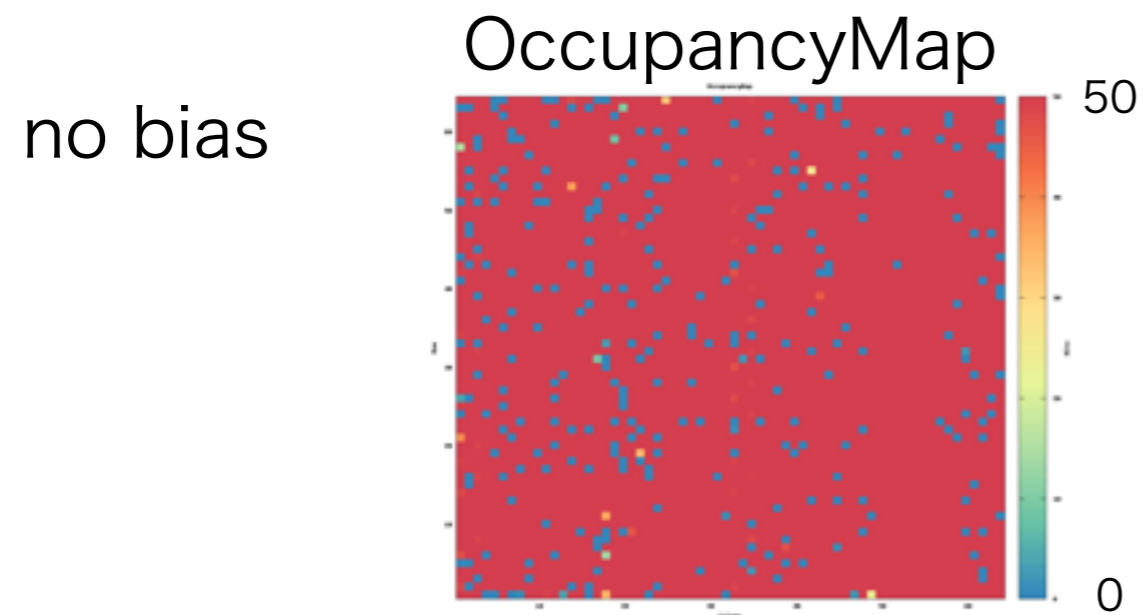
170613

Hitomi Tokutake

Analog scanによるMaskを追加

◆ Analog scanによるMaskを追加

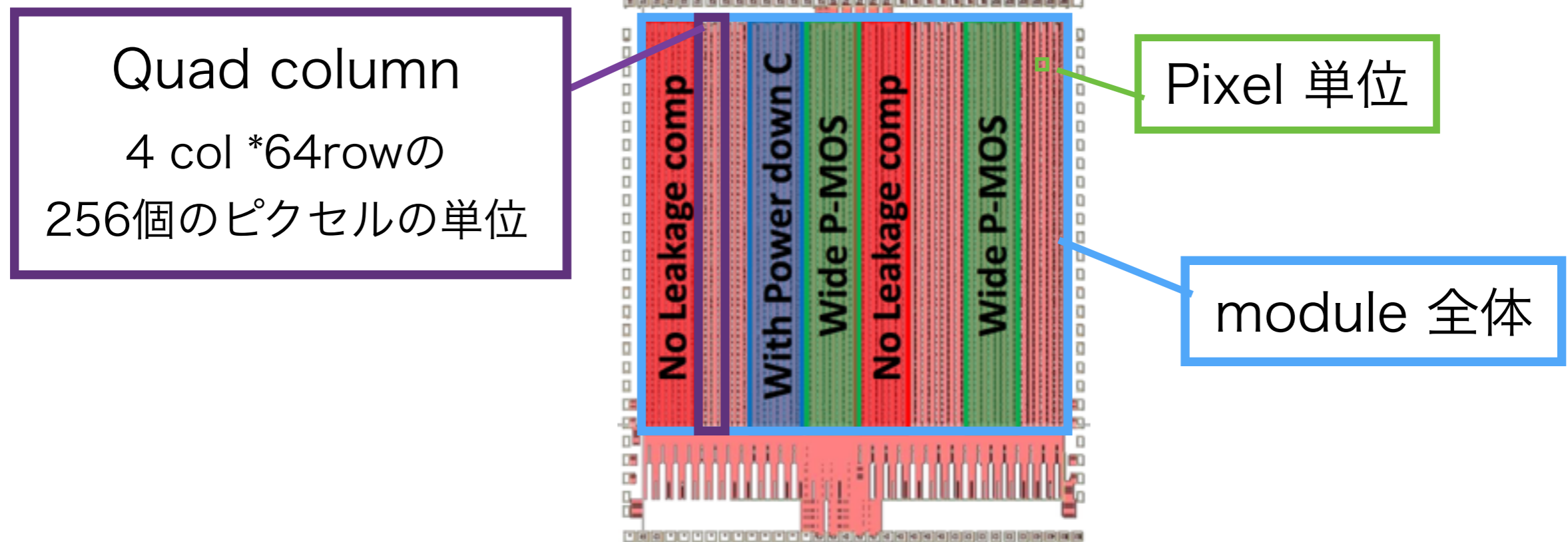
- 応答数が50/50でないピクセルはDeadPixelとして扱う



タイミング問題

◆ Dead timeの存在を仮定する

-Dead timeの発生源について3つの候補を用意した



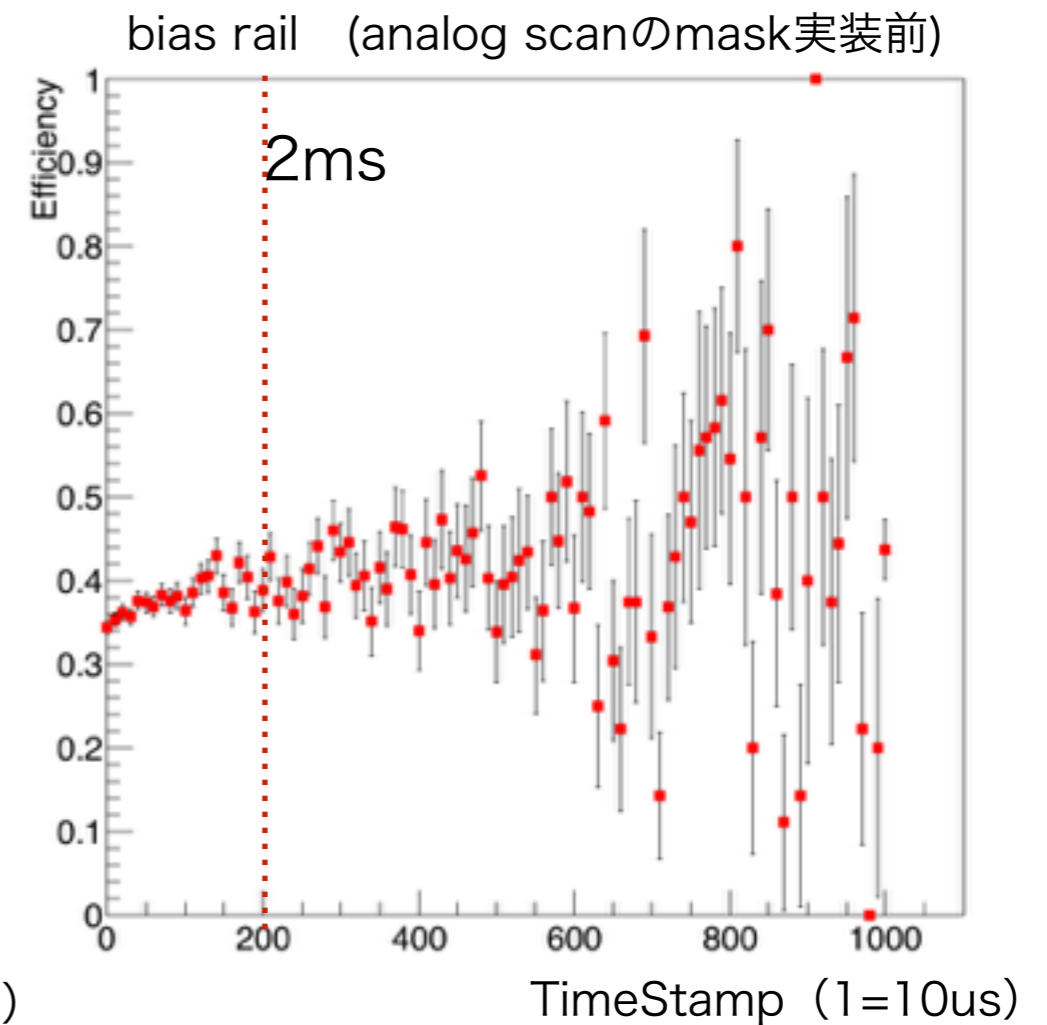
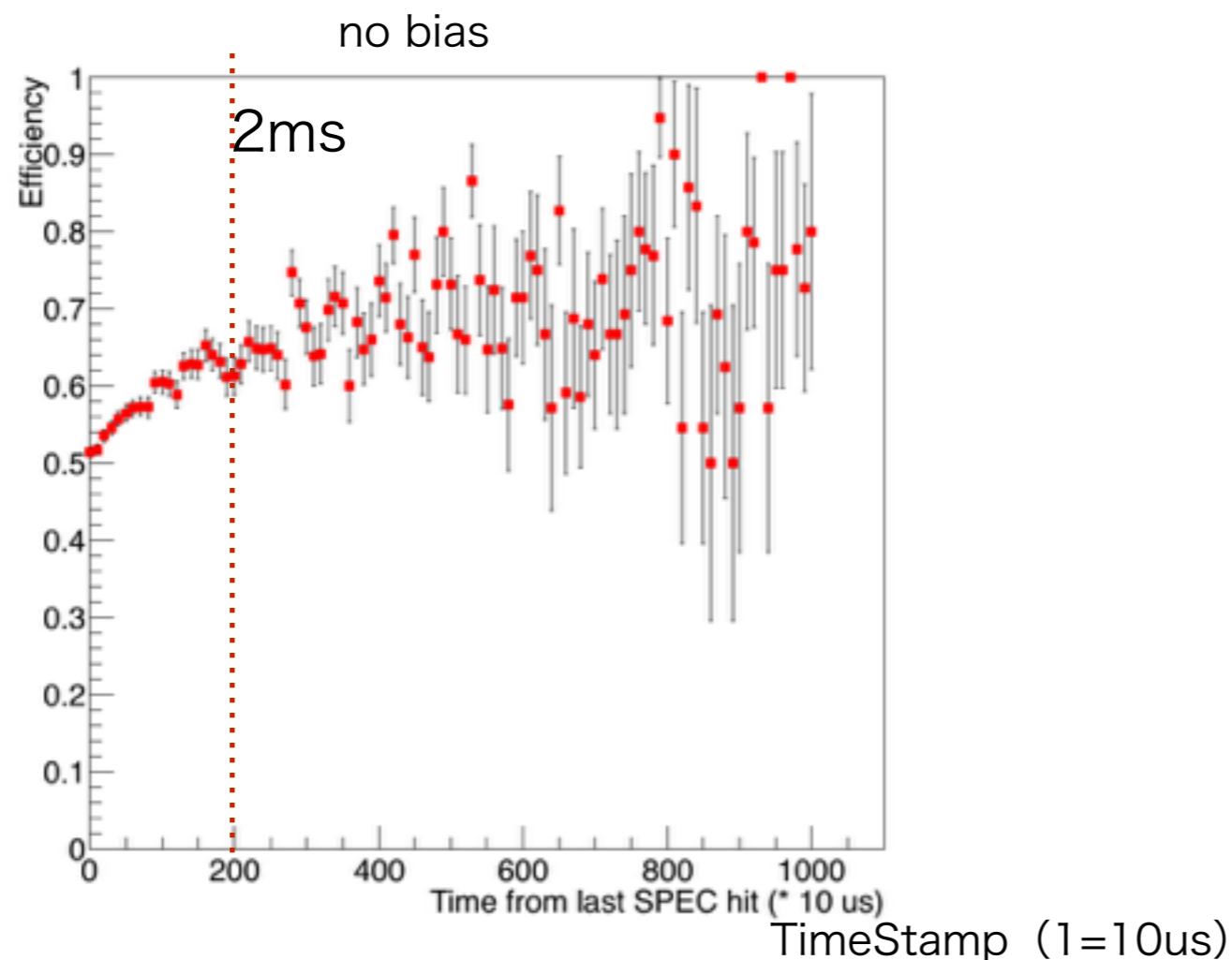
◆ 調査方法

- 各グループでの最後のヒットからの経過時間と検出効率の相関を調べる

タイミング問題

◆ Timing issue (module)

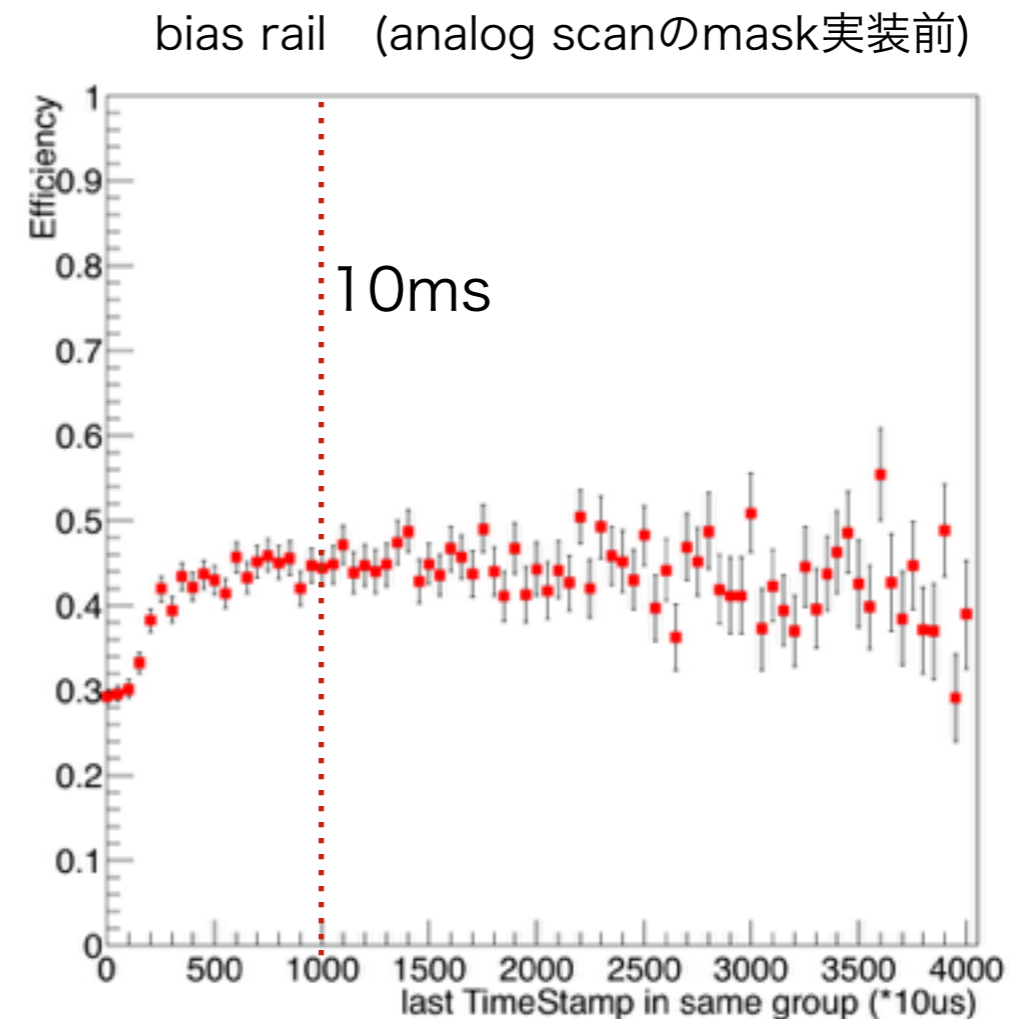
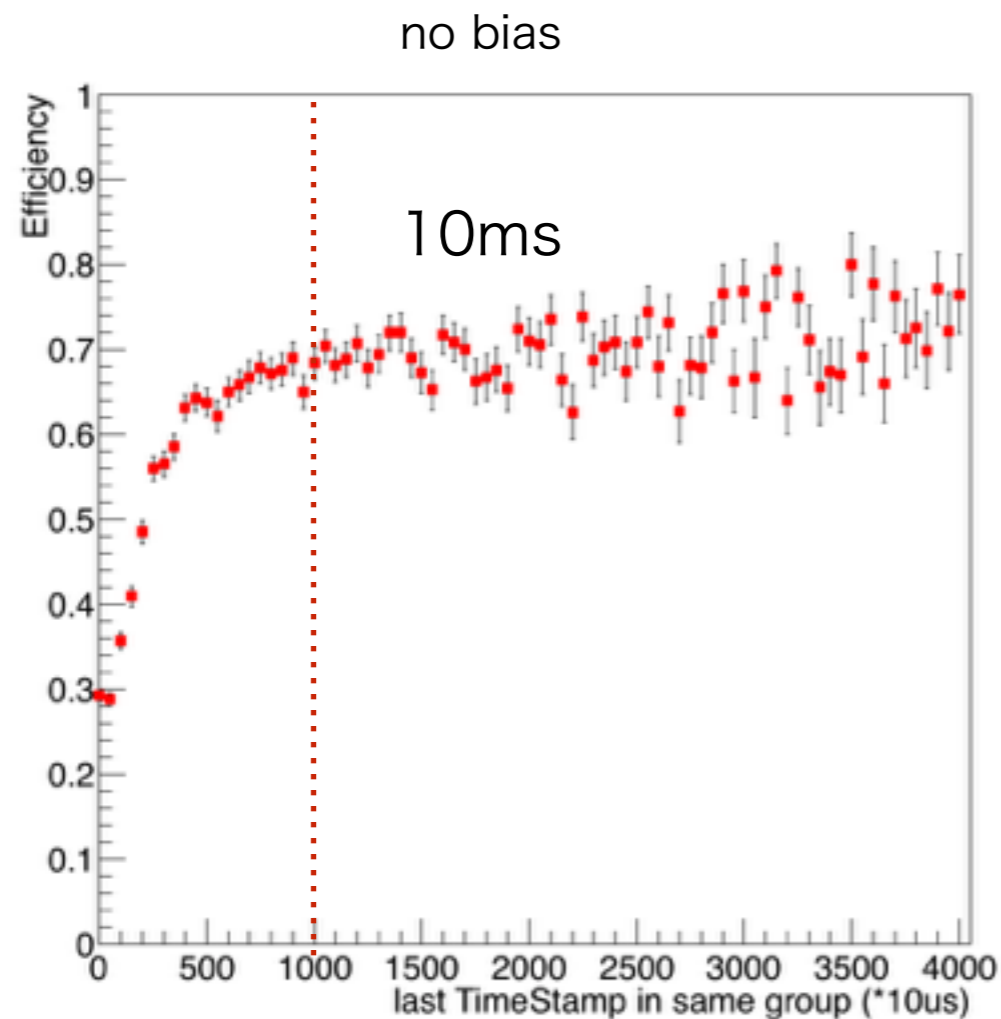
- 前の(FE65ヒットからの時間ごとに検出効率を求めた
- 2msあたりまでは時間が経つほど検出効率が上がっている
- 検出効率は70%程で頭打ちになる(他にも検出効率低下の原因がある)



タイミング問題

◆ Timing issue (Quad column)

- 前の同じQuadColumnグループ内のヒットからの経過時間と検出効率
- 4msあたりまでは時間が経つほど検出効率が上がっている
- こちらも検出効率は70%程で頭打ちになる

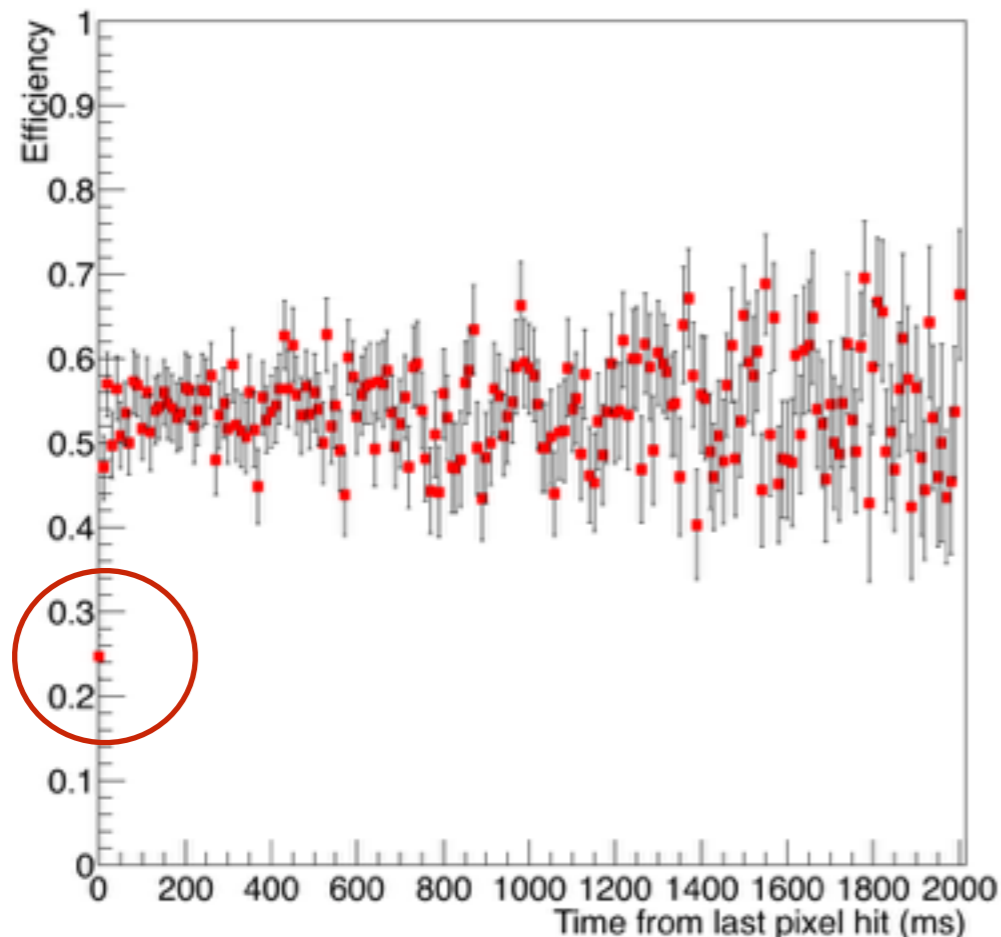


タイミング問題

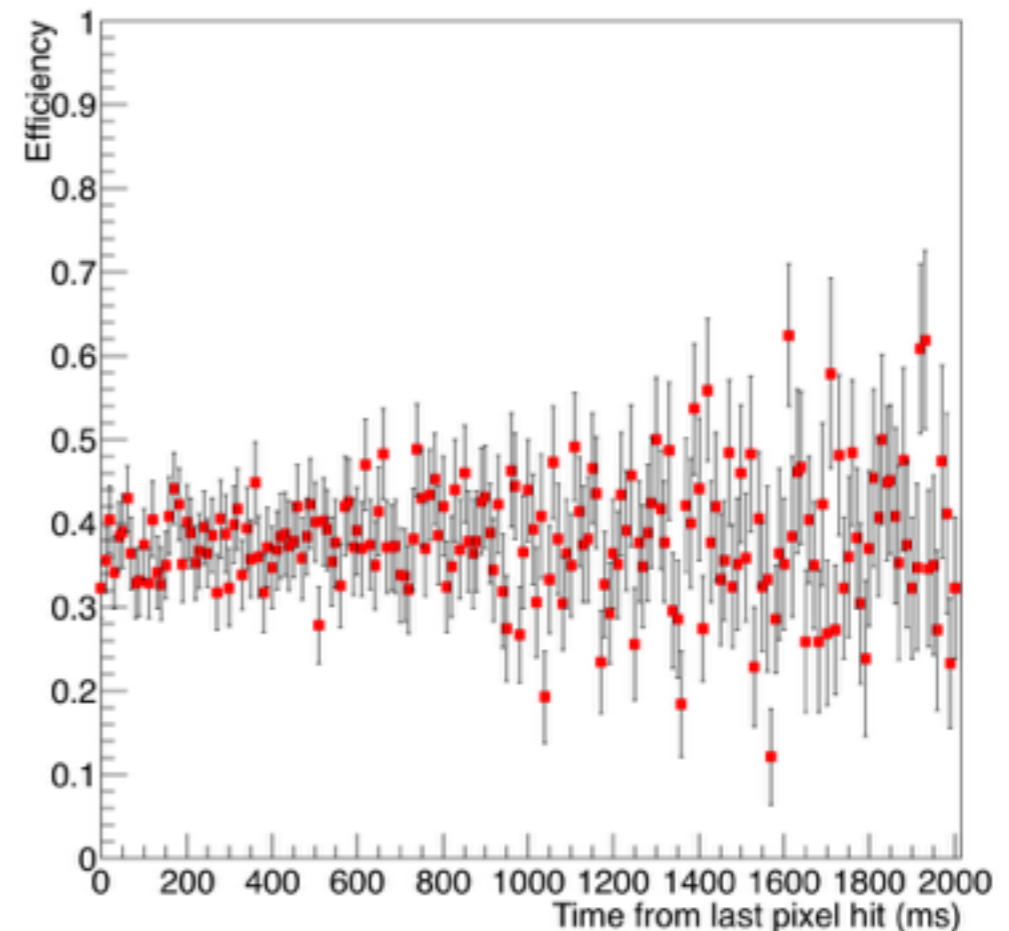
◆ Timing issue (pixel unit)

- 前回同じピクセルがなってからの時間ごとに検出効率を求めた
- trackが通ったピクセルが最後に鳴った時のTimeStampを参照した
- no biasは前回のHitから10ms以内のとき検出効率が低い
- bias railは上記の現象は見られない

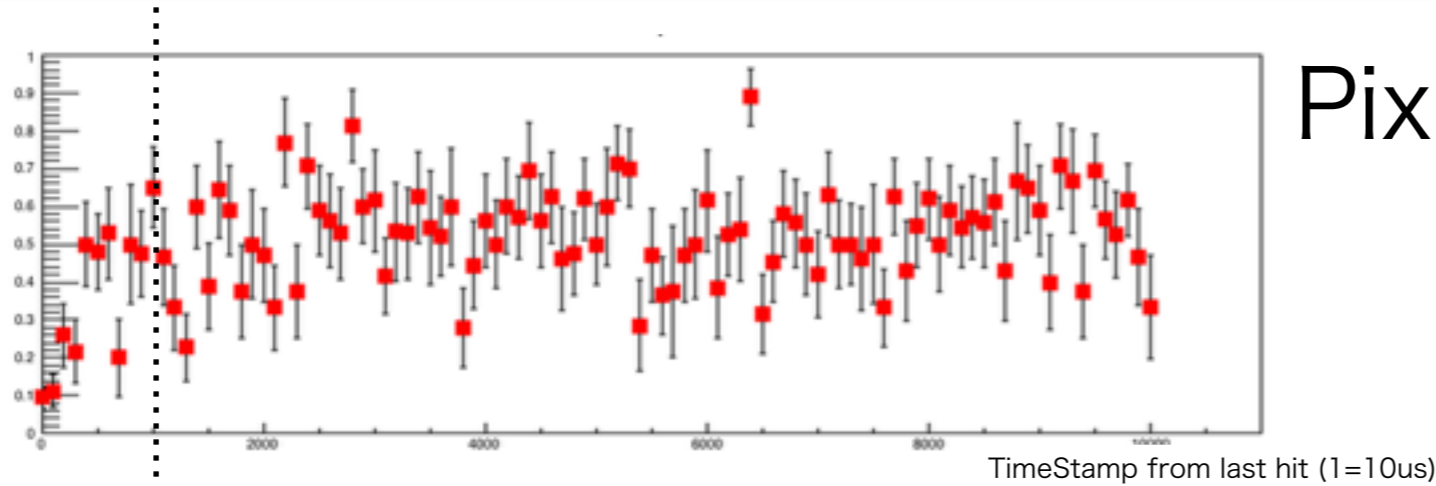
no bias



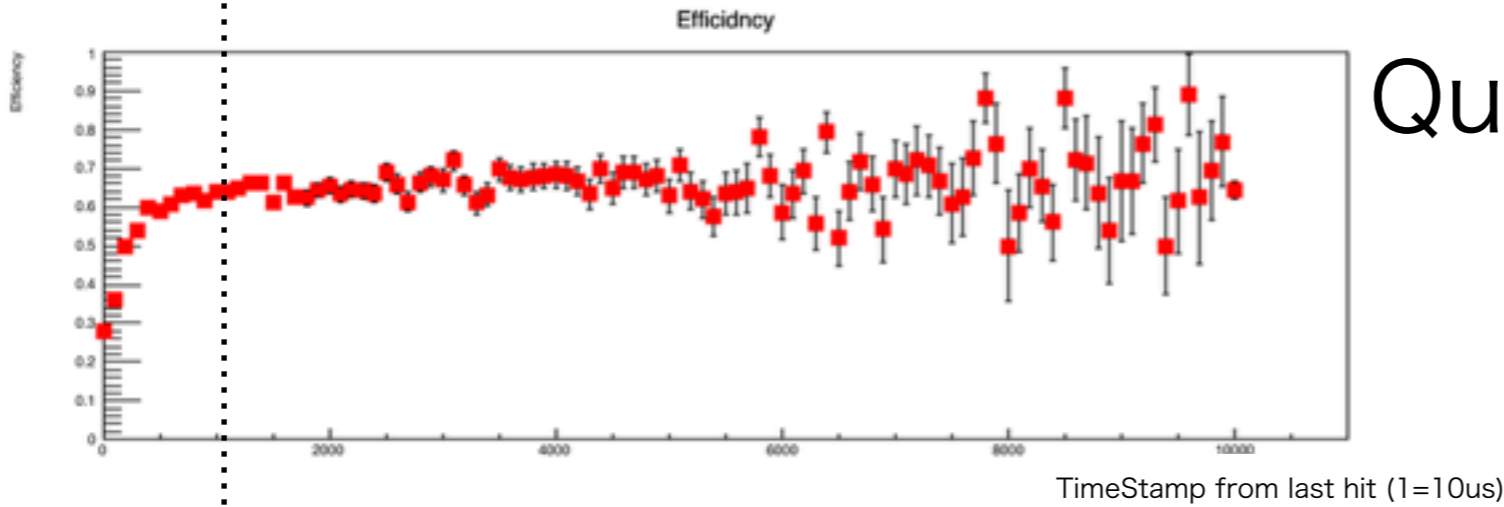
bias rail (analog scanのmask実装前)



3つの比較

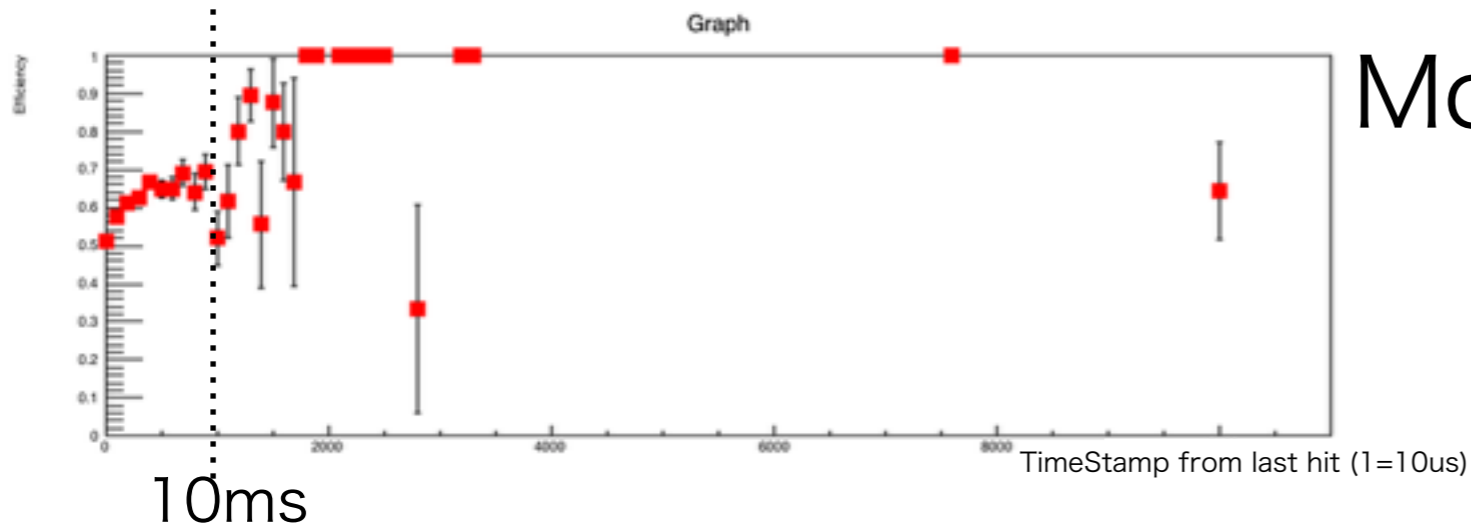


Pixel



Quad Column

→一番影響がありそう



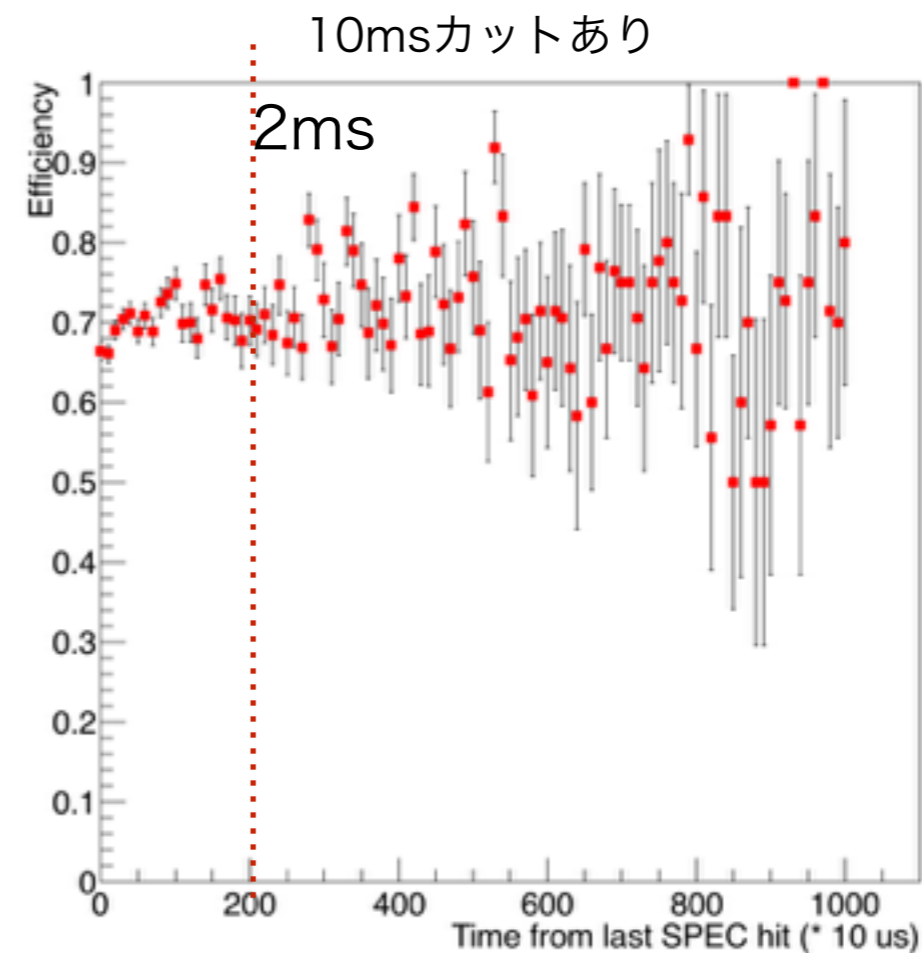
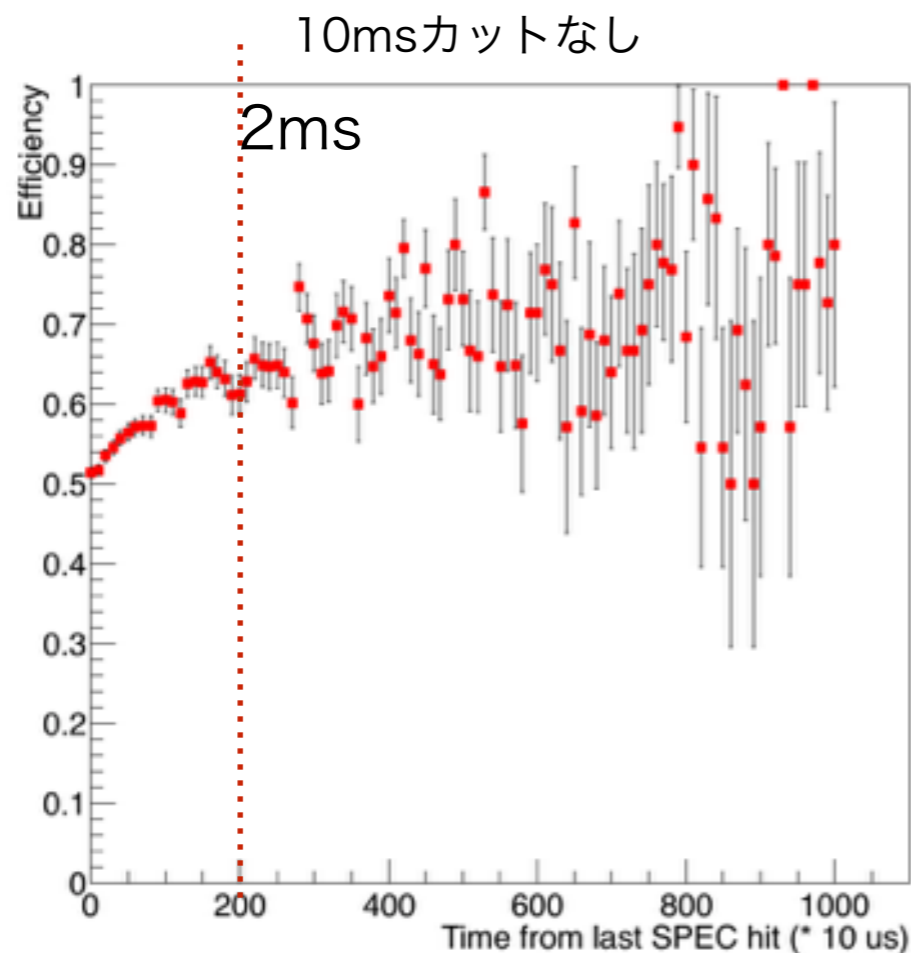
Module

3つ全てについてDeadtimeの存在を否定できない？

Moduleから生じるDeadtimeの検証

◆ Timing issue (module unit with 10ms cut)

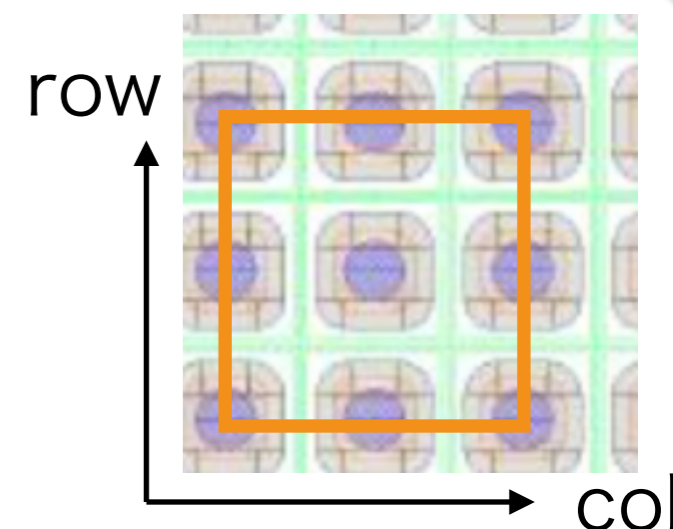
- 同じQuadColumn内での最後のヒットから10ms以上のイベントのみを
取り出して(これを10msカットと記す)Module全体のDeadtimeを検証した
- QuadColumnの影響を減らしてModuleから生じるDeadtimeの効果が見られる
- 左側(<2ms)の検出効率の落ちが大きく改善した
- 統計数は6割近く減る



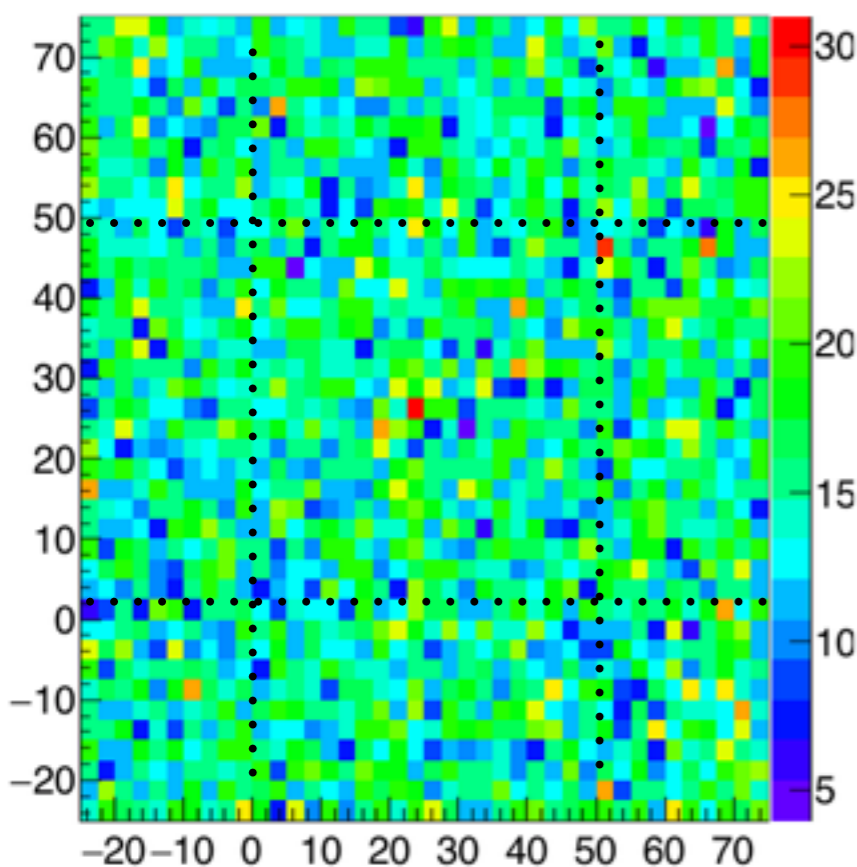
ピクセル内部の検出効率

◆ KEKFE65-6 (no bias)

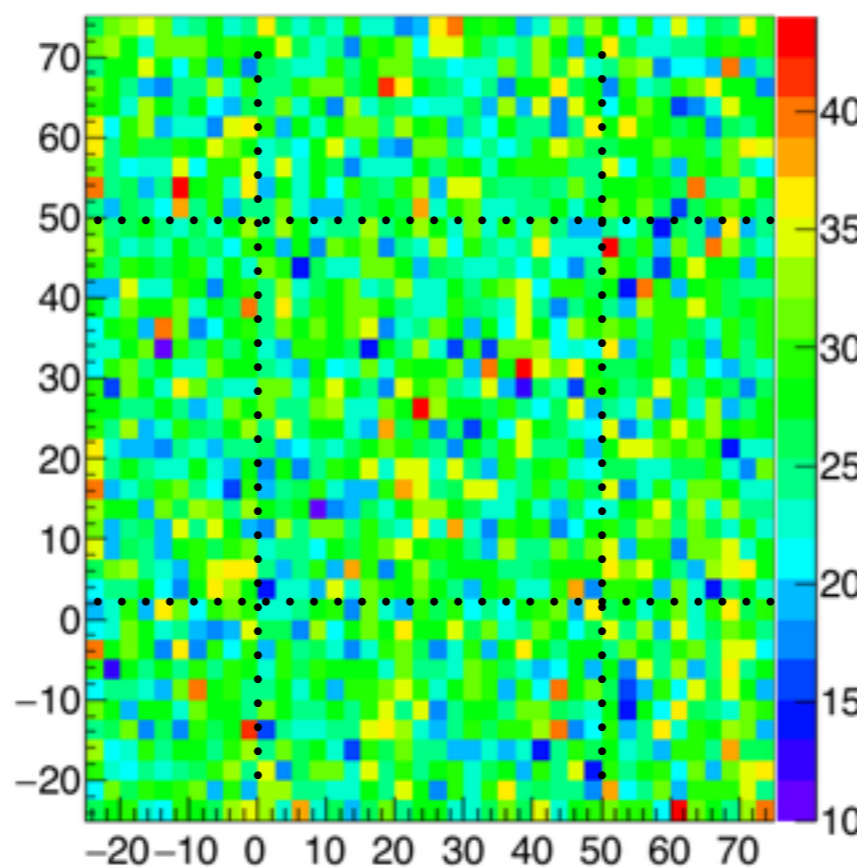
- 4ピクセル(2*2)分のEfficiency Map
- 概ね検出効率は一様



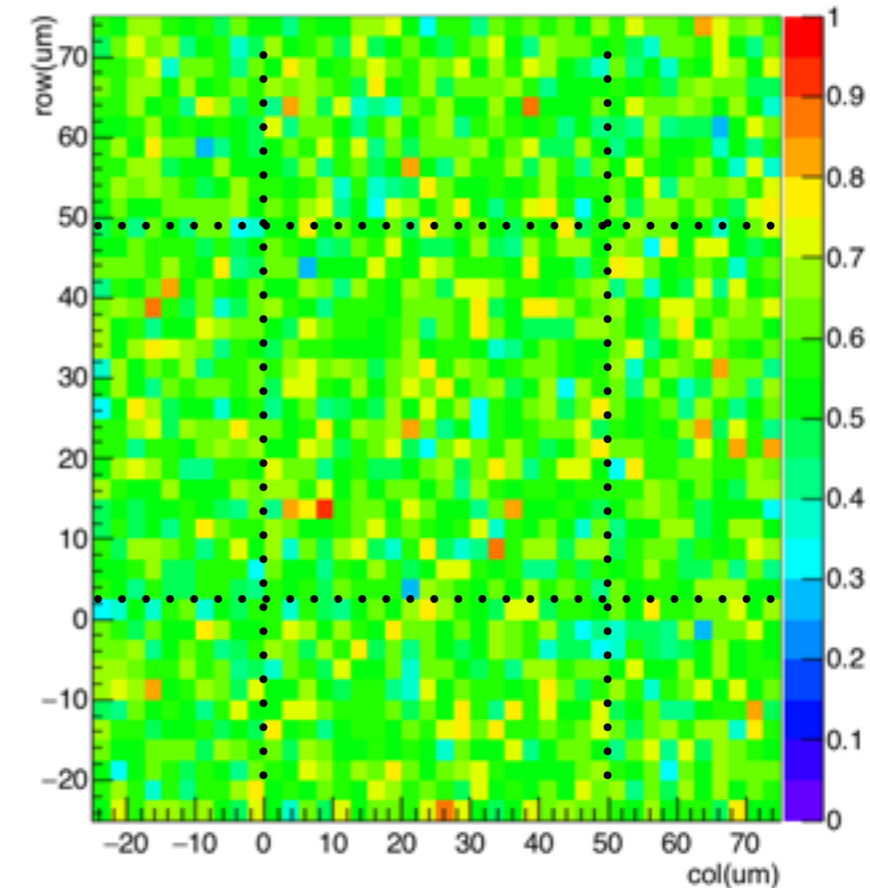
Track with Hit



Track



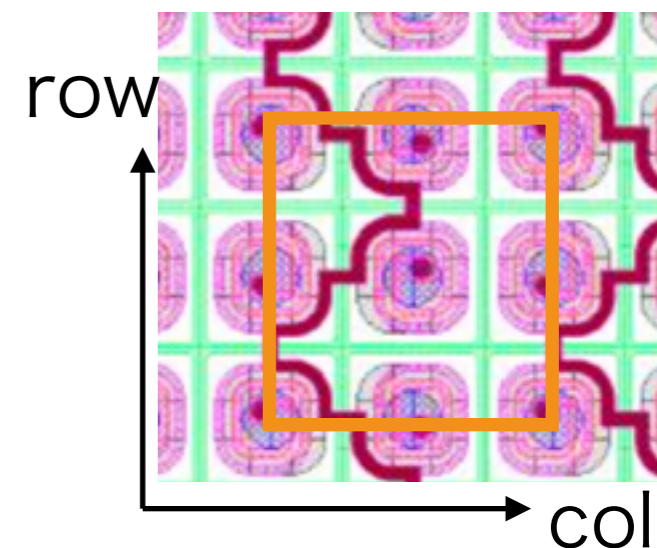
Efficiency



ピクセル内部の検出効率

◆ KEKFE65-9 (with bias rail)

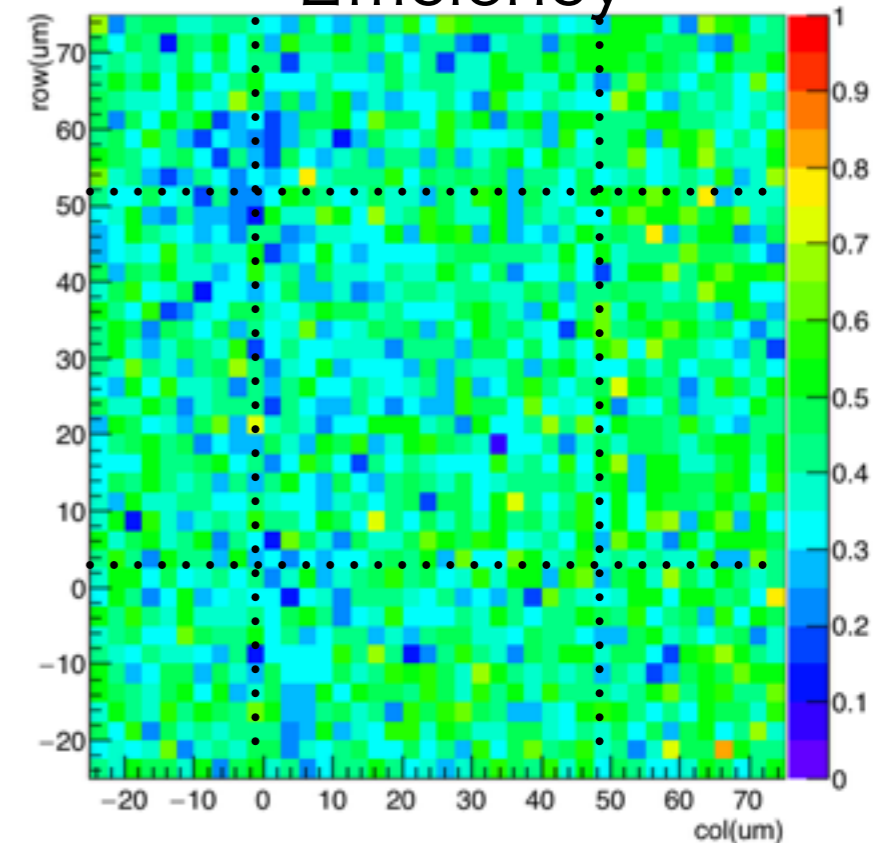
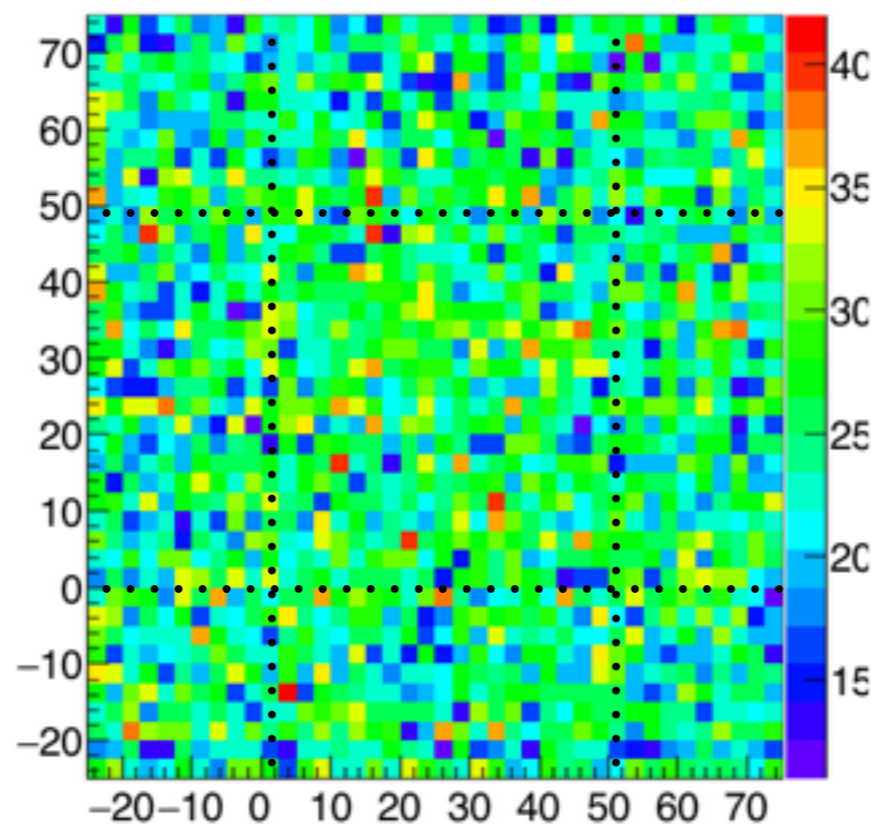
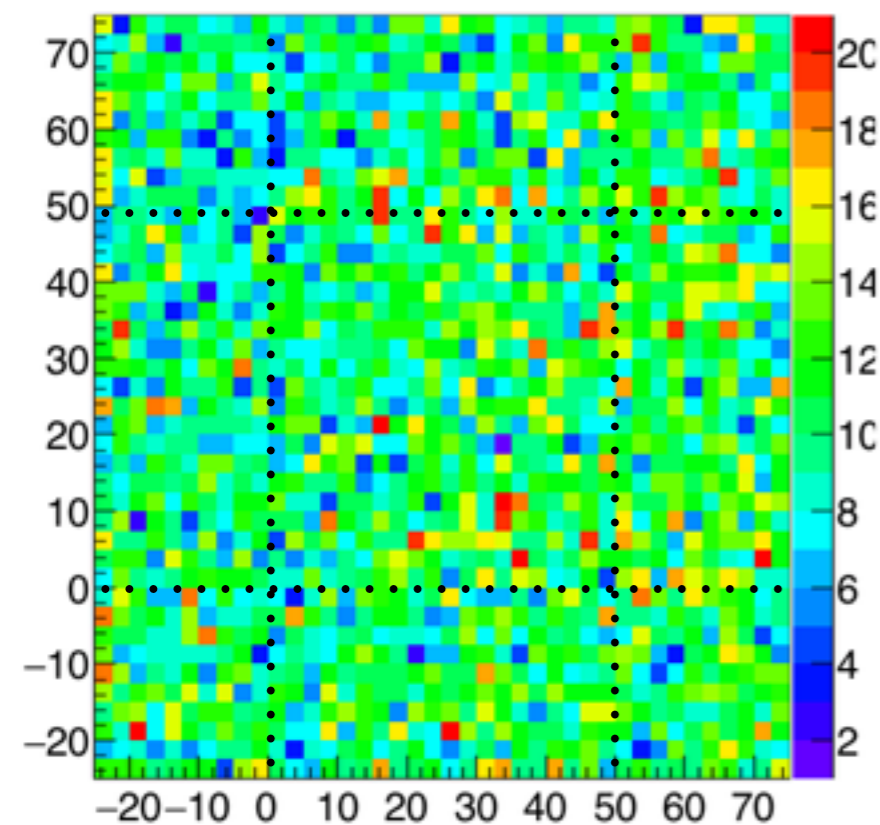
- ピクセル境界部の1つ(0,50)で検出効率が大きく低下
- $x=0$ (bias railがある方)上は全体的に検出効率が低い?
- 1つのrunだけでは統計数が足りない



Track with Hit

Track

Efficiency

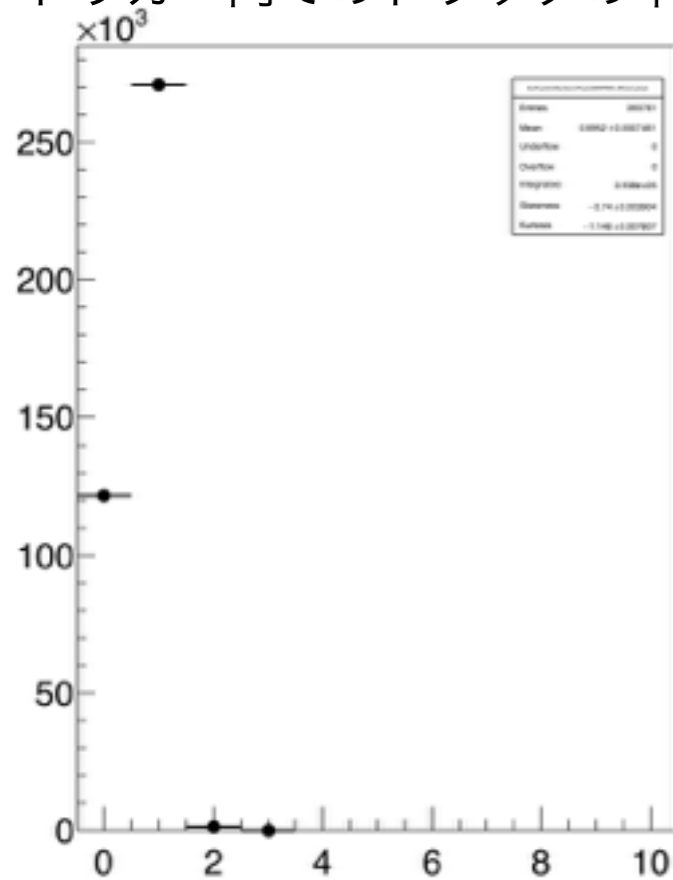


(analog scanのmask実装前)

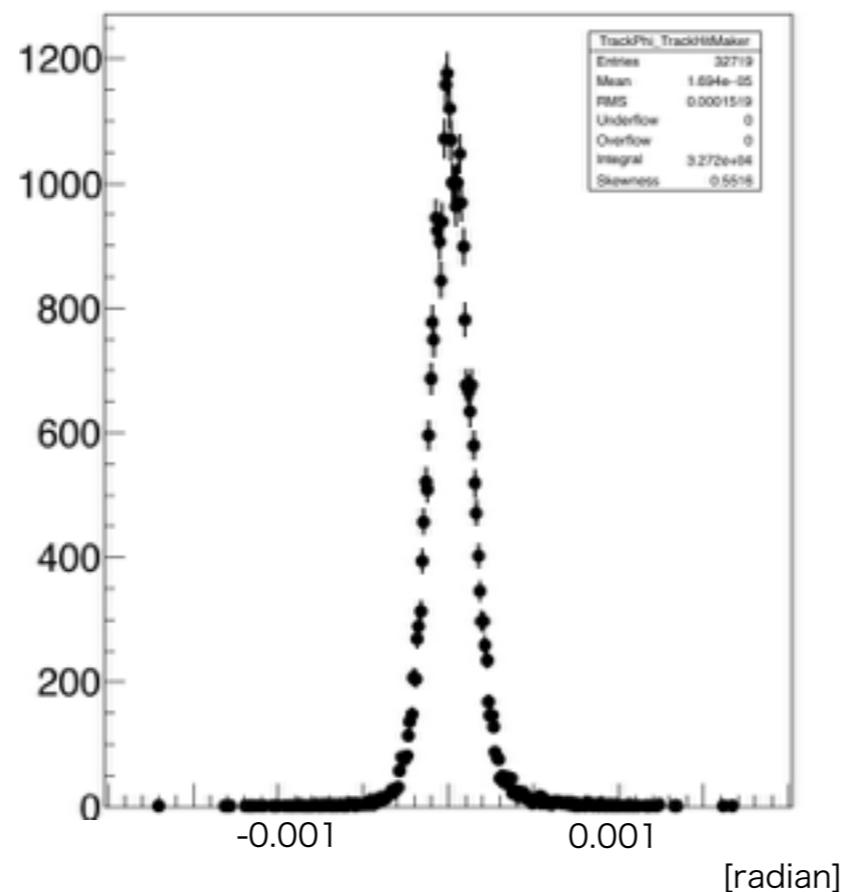
Trackのチェック

- 変なトラックをたくさん引くと検出効率が下がる
- 各トリガーでのトラックの本数、傾きをチェックした
- $\text{ChaiX} > 3$, $\text{ChaiY} < 3$ を要求

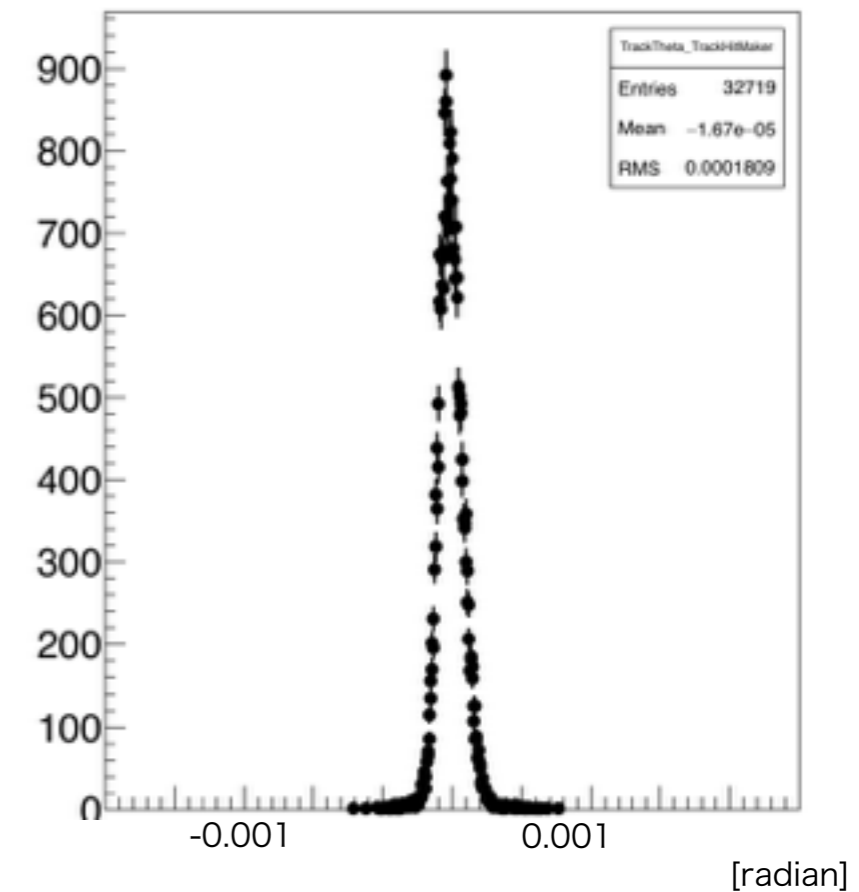
トリガー内でのトラックの本数



トラックのY方向の傾き



トラックのY方向の傾き

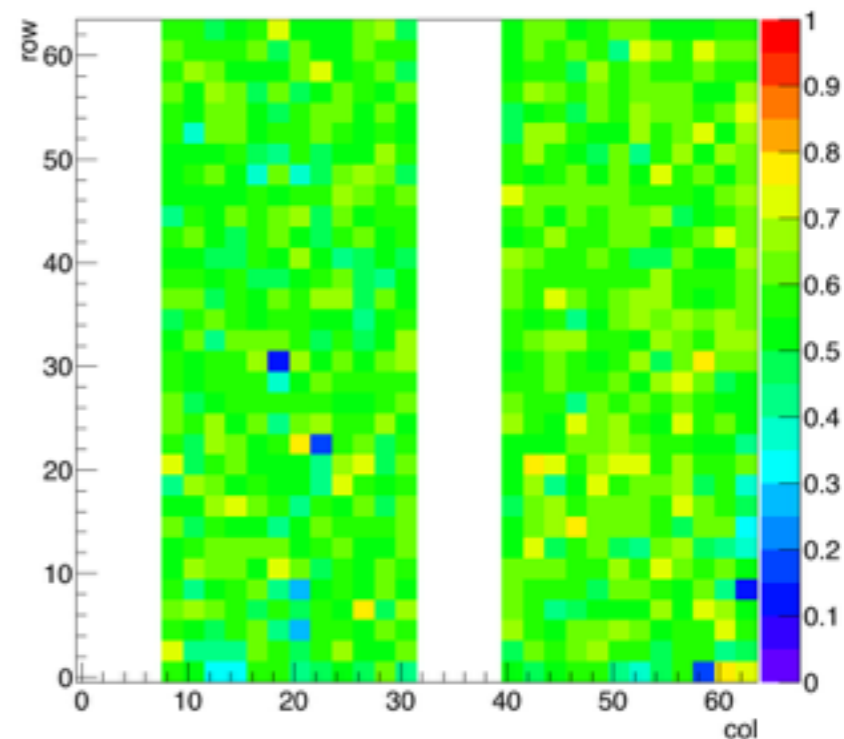
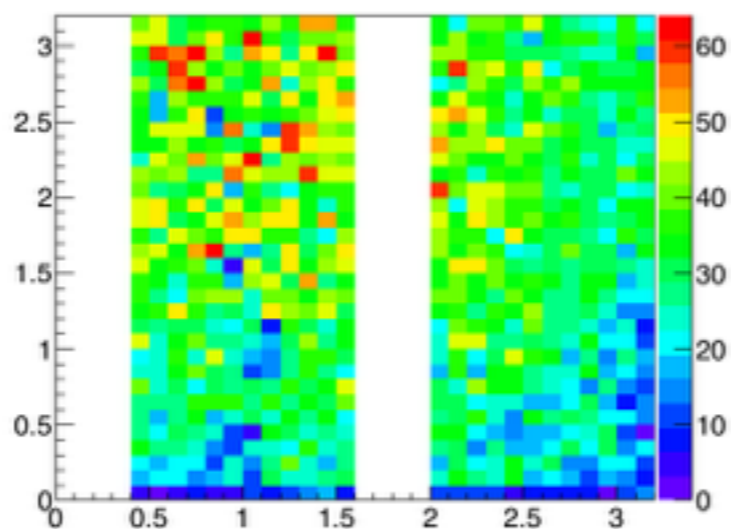
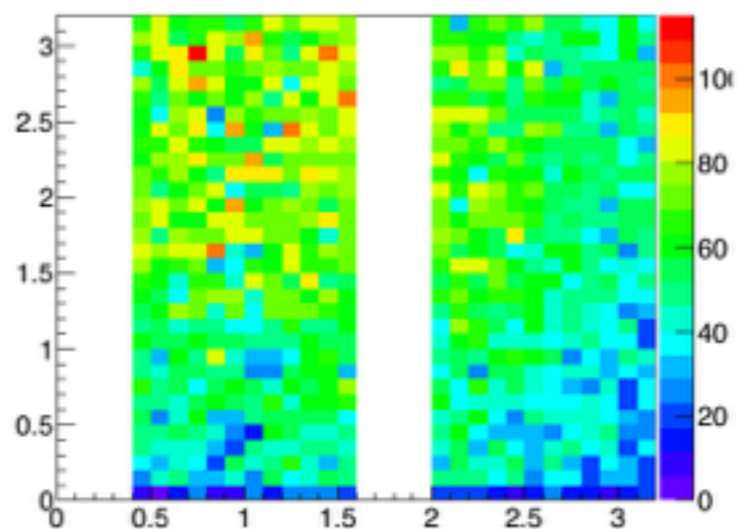


back up

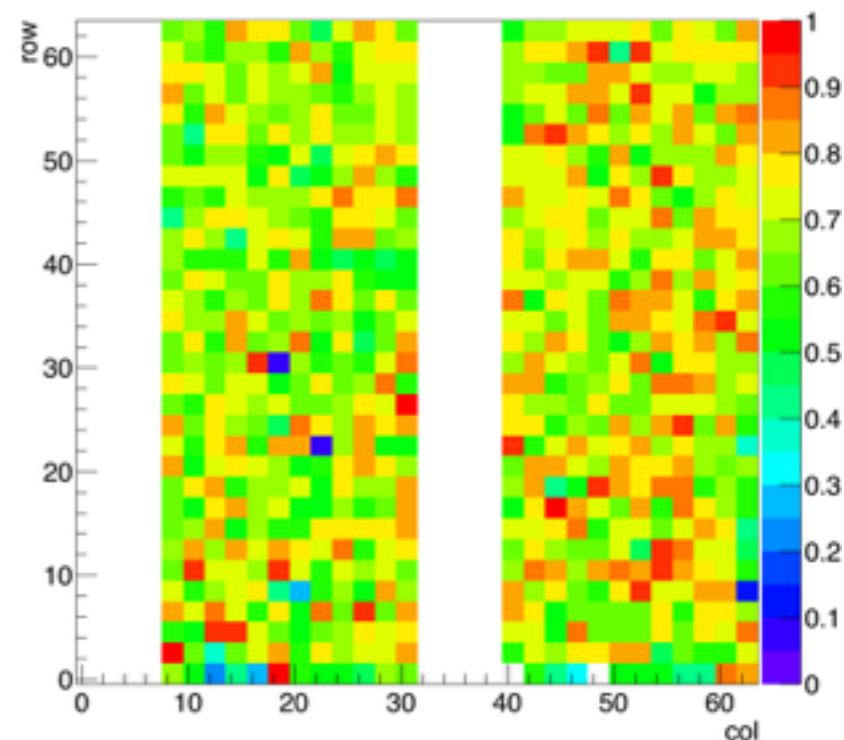
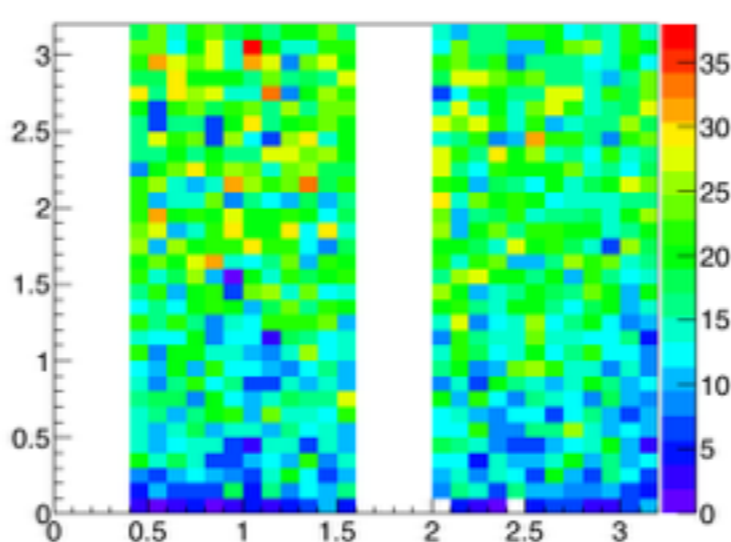
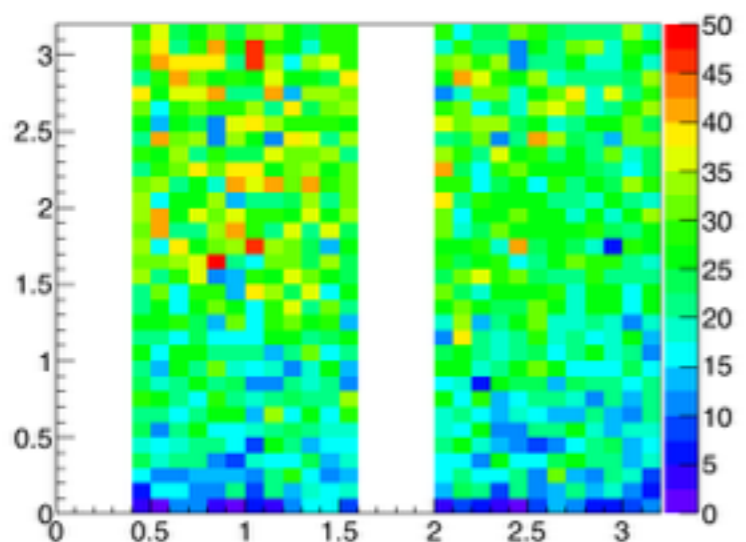
全体での検出効率

◆ KEKFE65-6 (no bias)

10msカットなし(59%)



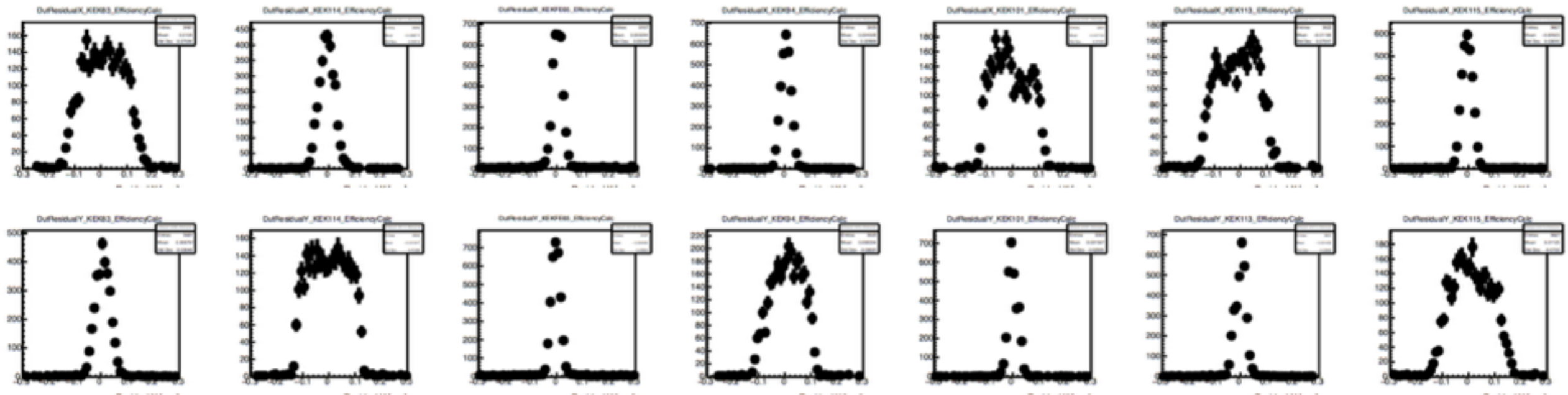
10msカットなし(72%)



アライメント

◆ Residual分布

- ピクセルごとの特性を調べる上でアライメントは非常に重要
- 鈴木くんのalignfileを借りてアライメントを確認してみた
- 各ヒット位置の誤差を正しく見積もるように変更した(一律100um \rightarrow pixelsize/sqrt(12))
- 後ろ3枚のResidualが二又にならなくなった

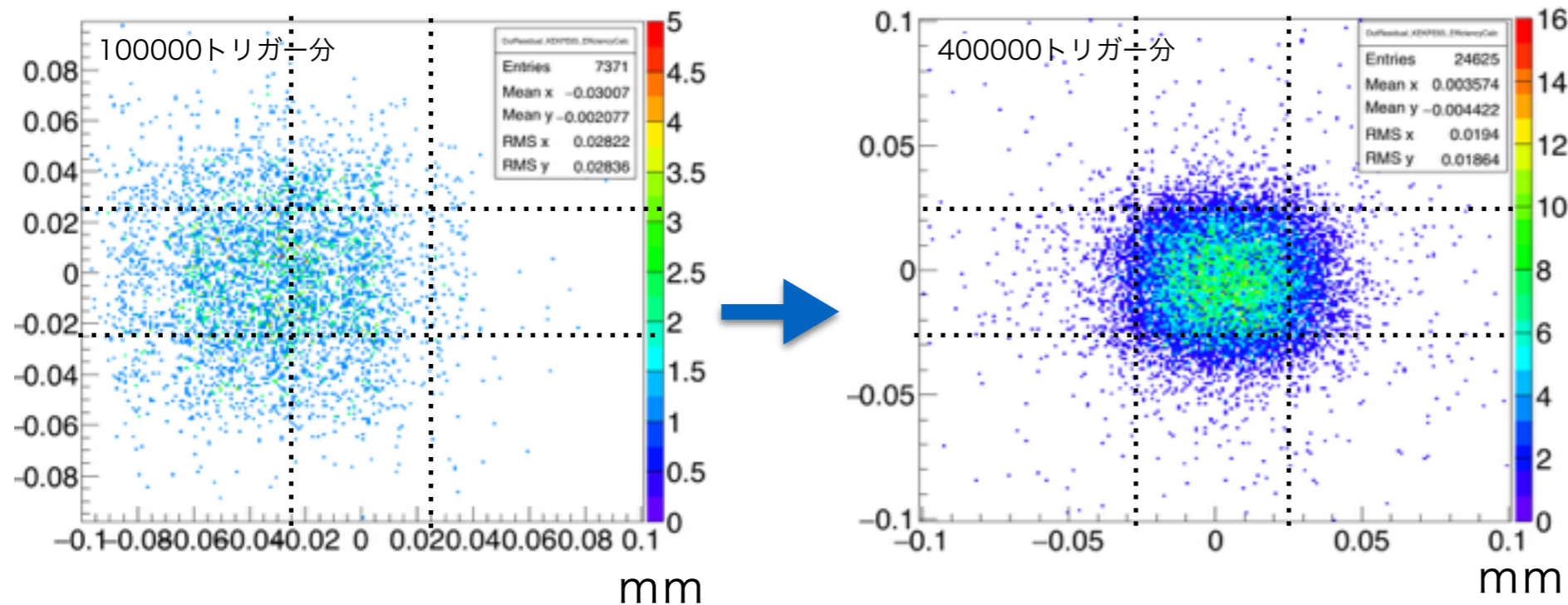


ビフォーアフター

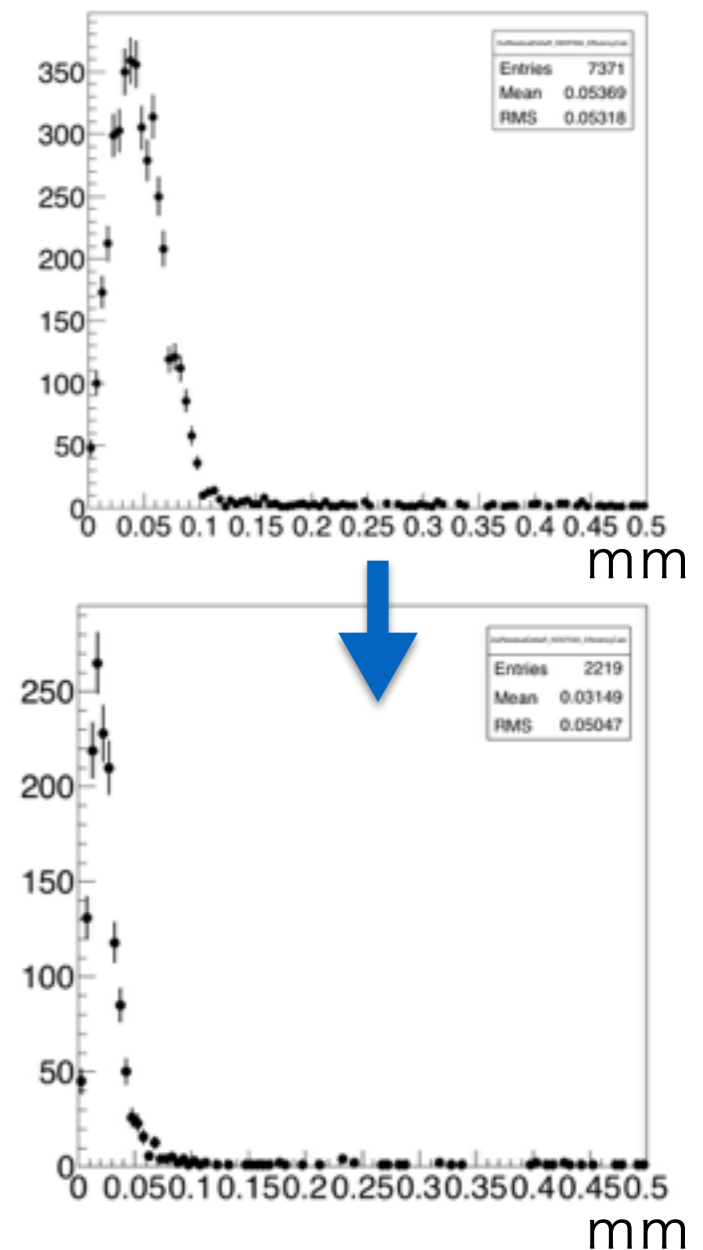
◆ TrackHitとHit

- TrackHitとHitの差を2次元、1次元(距離)で確認
- ほとんどのtrackでtrack通過位置とhit位置が同じピクセルに乗るようになった

Track hitとHit間の差(dx, dy)の分布



TrackとHit間距離の分布



検出効率の原因調査

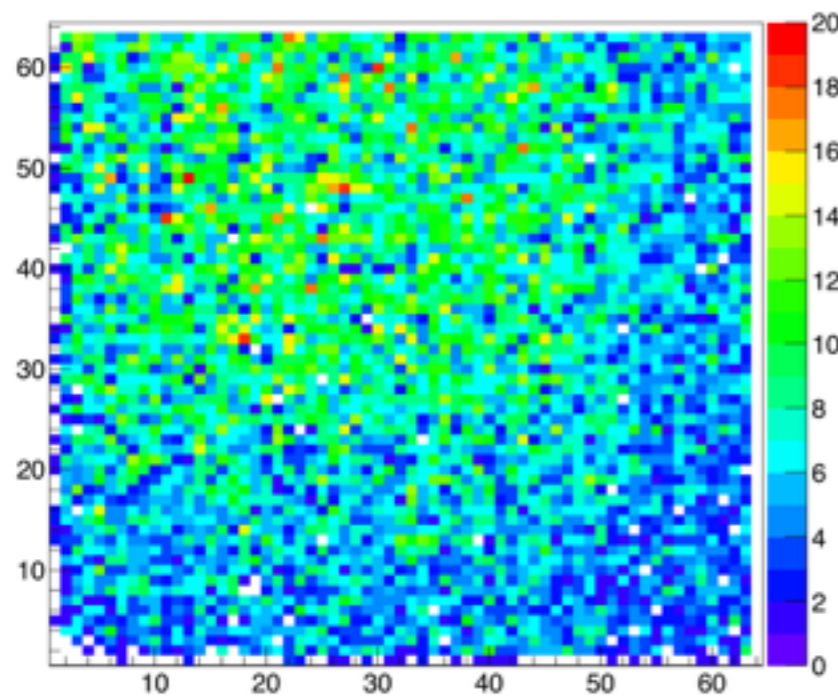
- ◆ タイミング関係
 - 各ピクセルのDeadTime
 - モジュール全体のDeadTime →多少影響あるかも？
- ◆ ~~Masked pixel~~の処理 ←あまり関係なかった
- ◆ 領域・ピクセル依存性
- ◆ HV依存性
- ◆ 照射の影響？

ToTの位置依存性

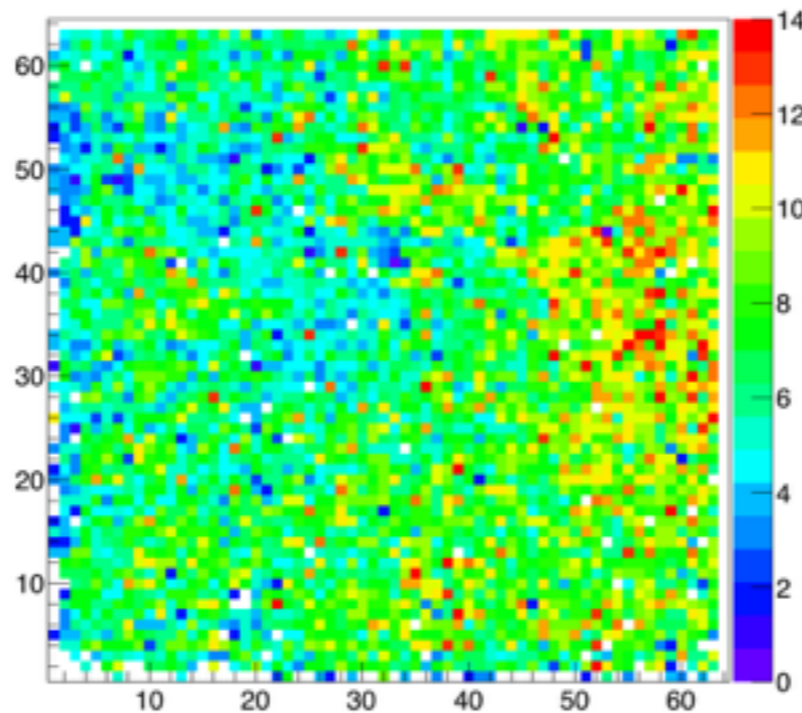
❖ KEKFE65-6(no bias)

- ToT分布の位置依存性を求めるため、各ピクセルの平均と分散(不偏分散)を求めた
- 左側にToT平均が低い領域、右側に高い領域がある
- 分散(ToTのばらつき)はほぼ一様だった
- 各ToT値の分布は次のページ

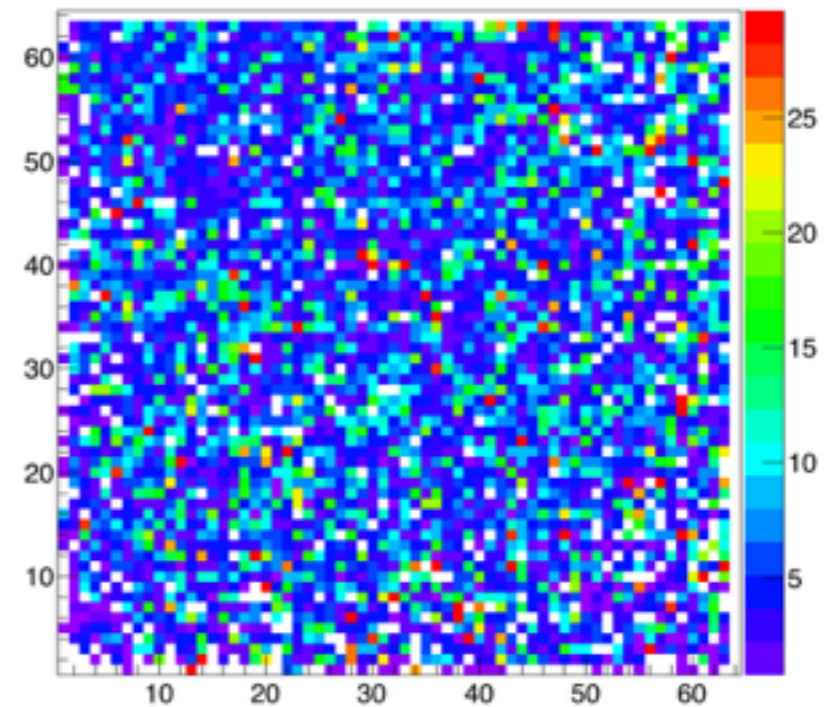
Hitmap



ToT平均



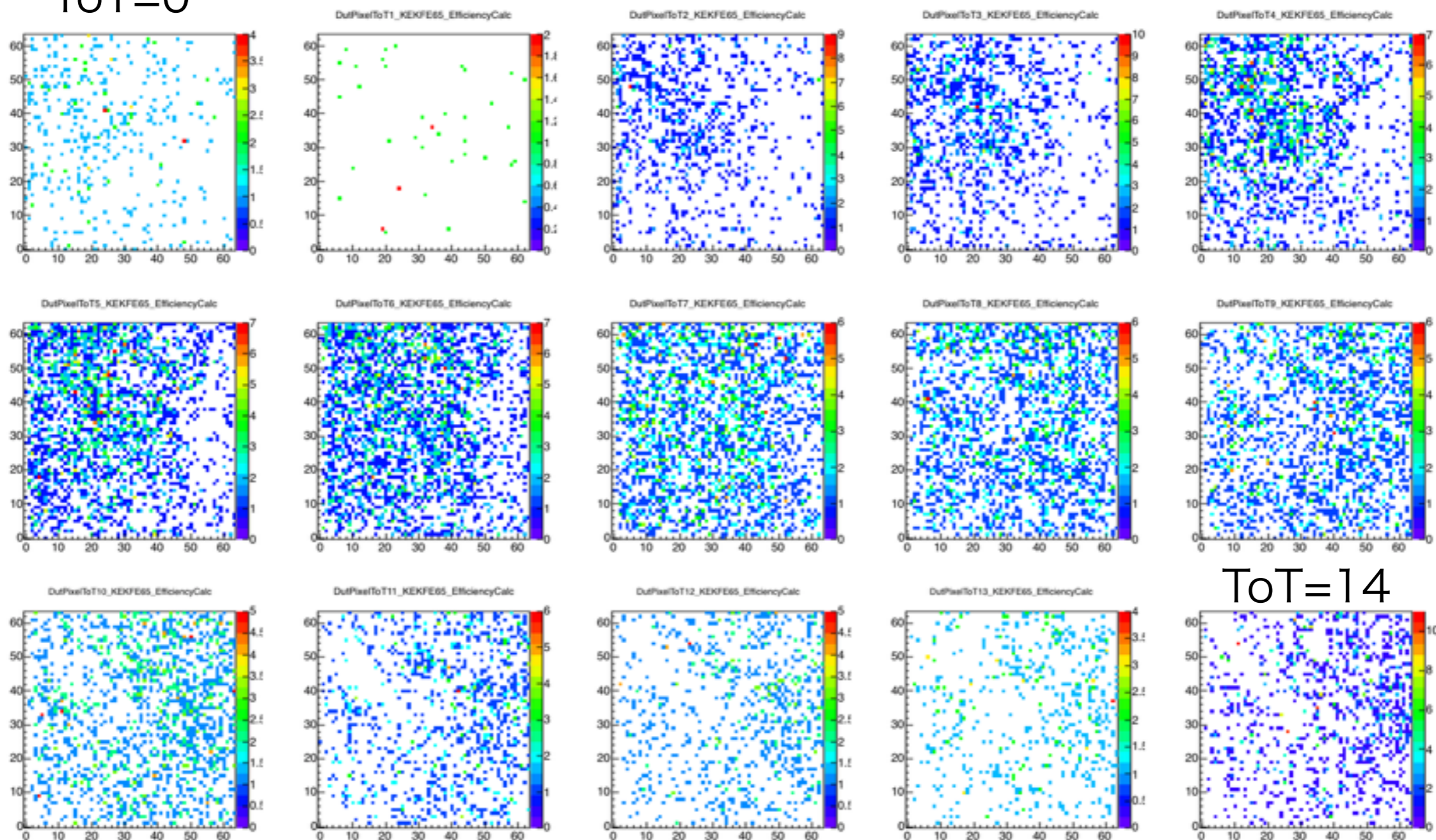
ToT分散



ピクセルごとのToT特性

❖ KEKFE65-6(no bias)

ToT=0



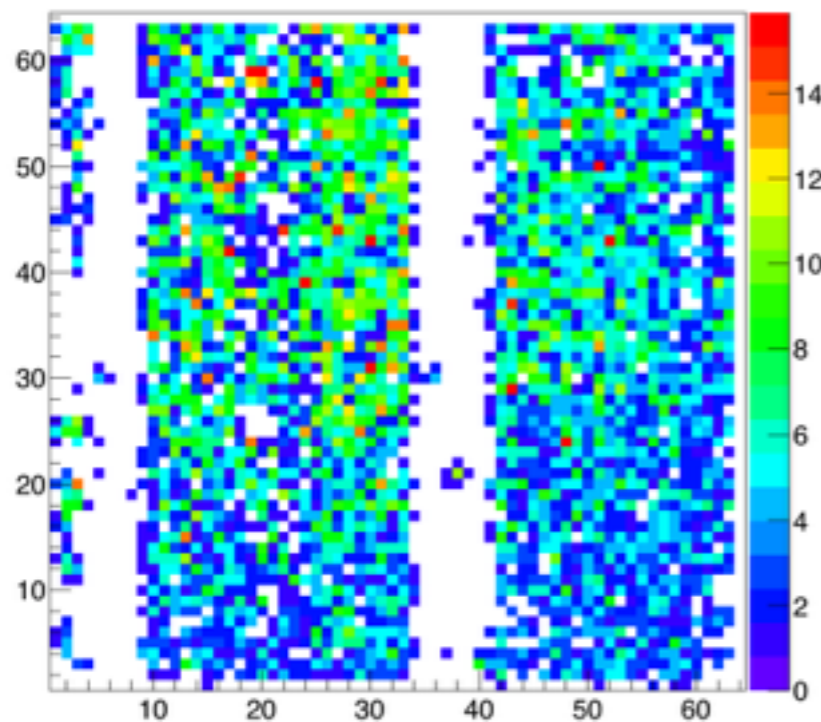
ToT=14

ToTの位置依存性

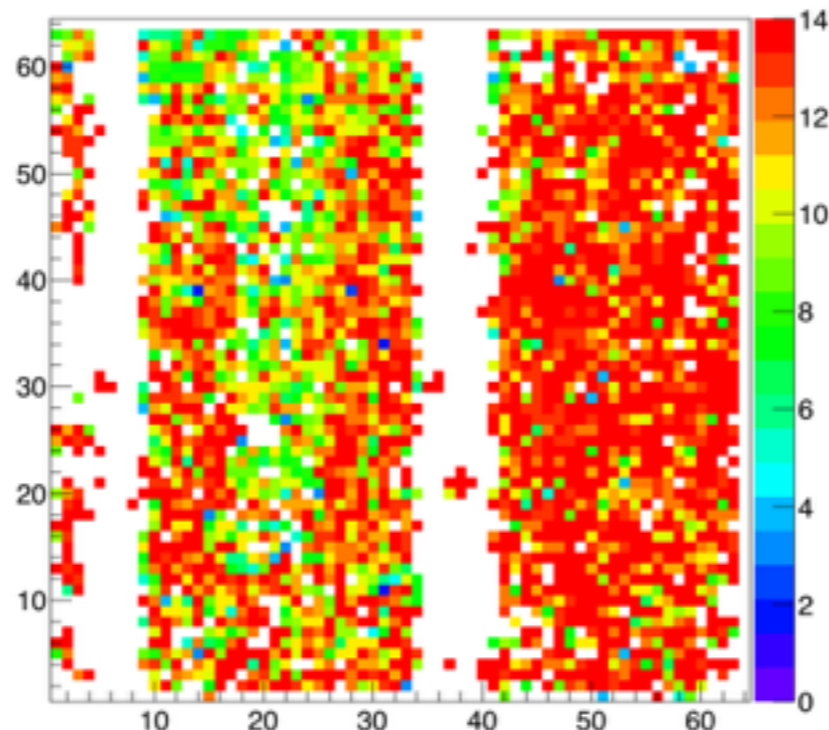
❖ KEKFE65-9(bias rail)

- ほとんどのピクセルでToT平均が12以上
- 左側に平均が低い部分がある(with Power Down C)
- こちらも分散(ToTのばらつき)はほぼ一様だった
- 各ToT値の分布は次のページ

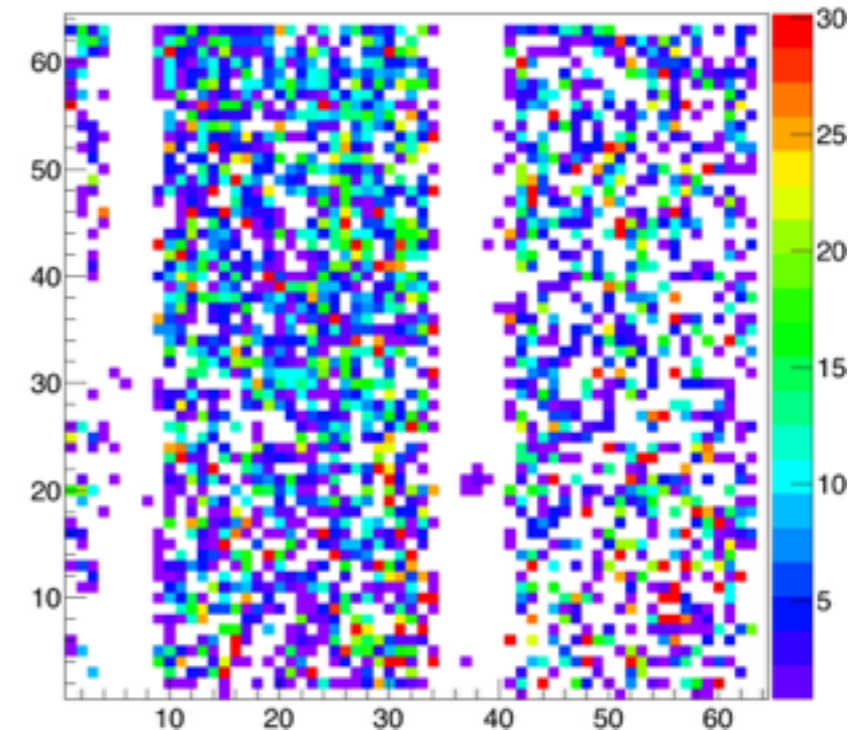
Hitmap



ToT平均



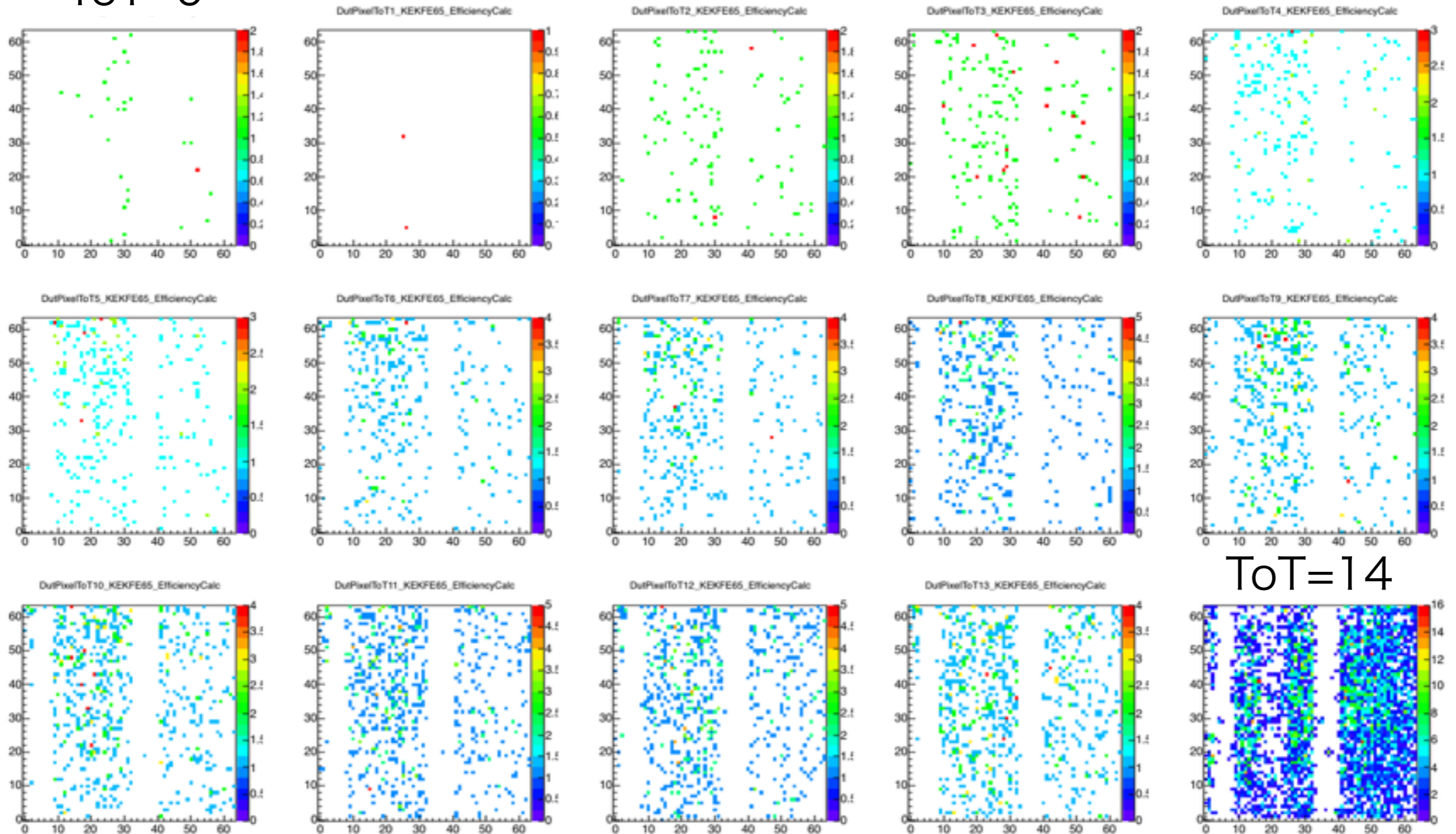
ToT分散



ピクセルごとのToT特性

❖ KEKFE65-9(bias rail)

ToT=0

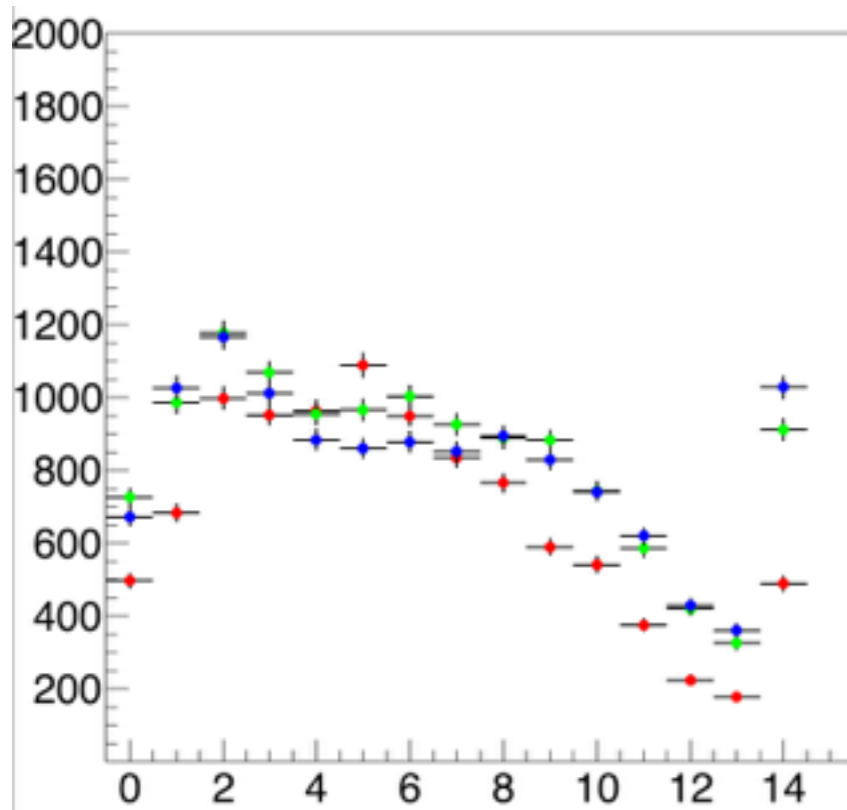


HV依存性

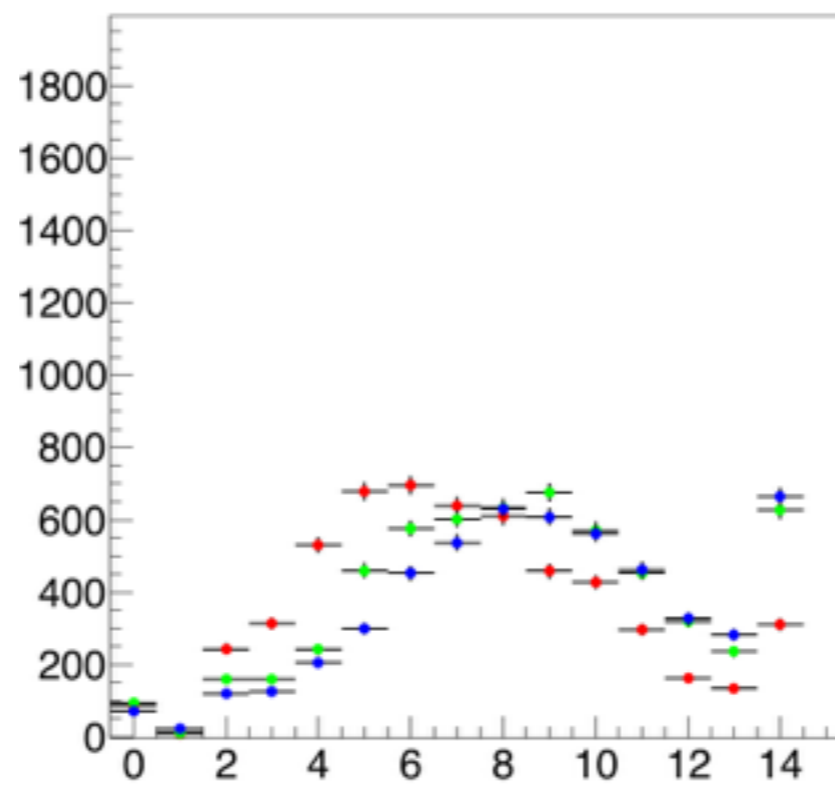
❖ KEKFE65-6(no bias)

- 低HVによって空乏層化が十分でなく、検出効率が低い可能性について考える
- ToTの山が左側で切れていないので、HVを上げることが検出効率向上につながるとは考えにくい(多少は上がるはず)

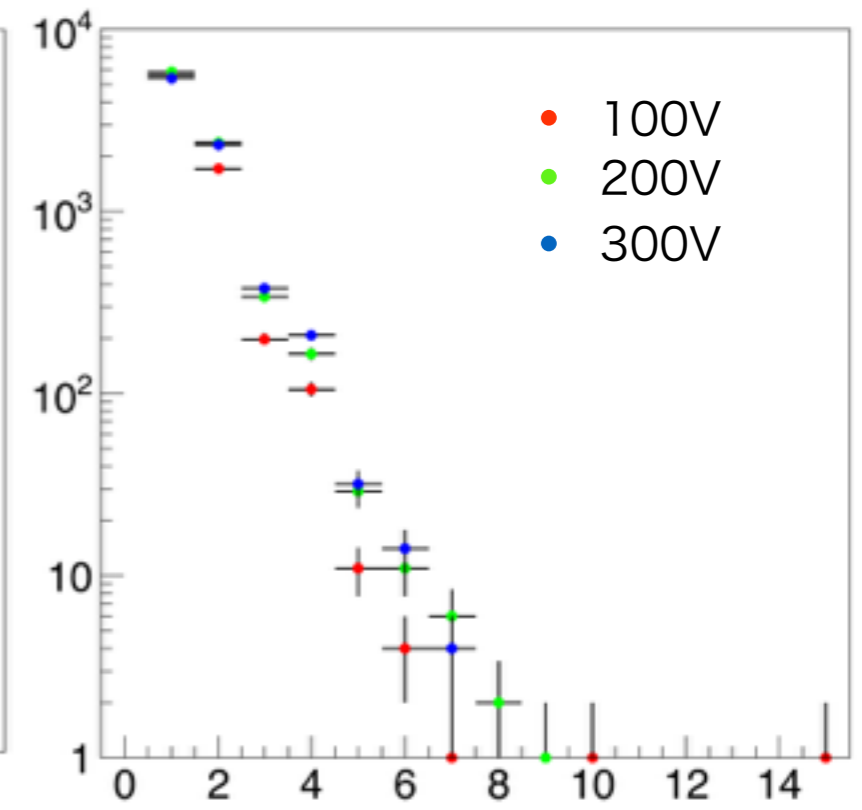
ToT分布



ToT分布(ClusterSize=1)



ClusterSize分布

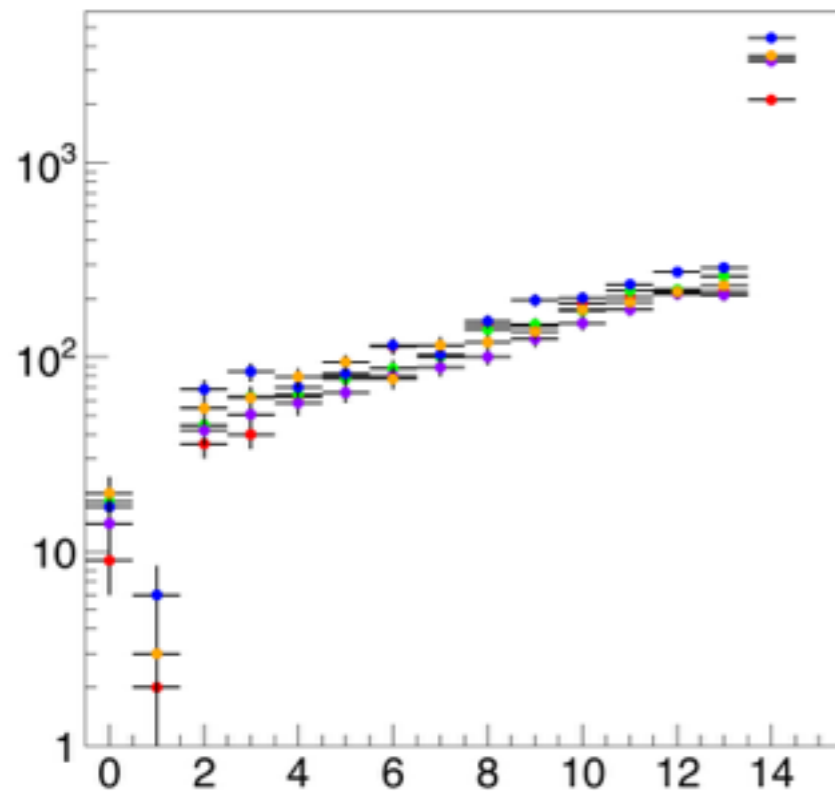


HV依存性

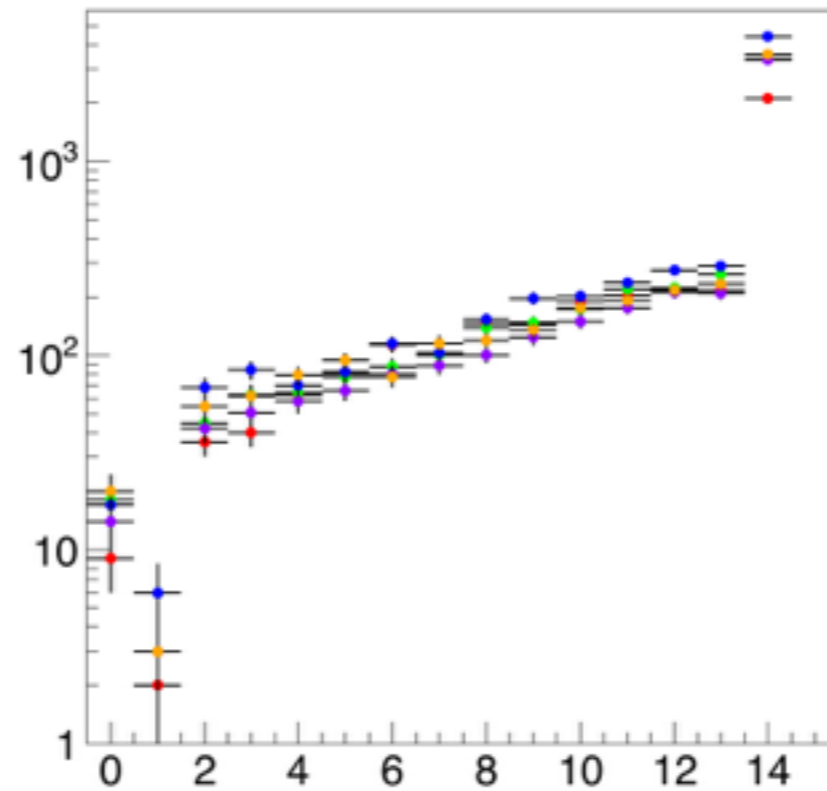
❖ KEKFE65-9(bias rail)

- とにかくToT=14が多い
- こちらもToTが左側で見切れているとは考えにくい

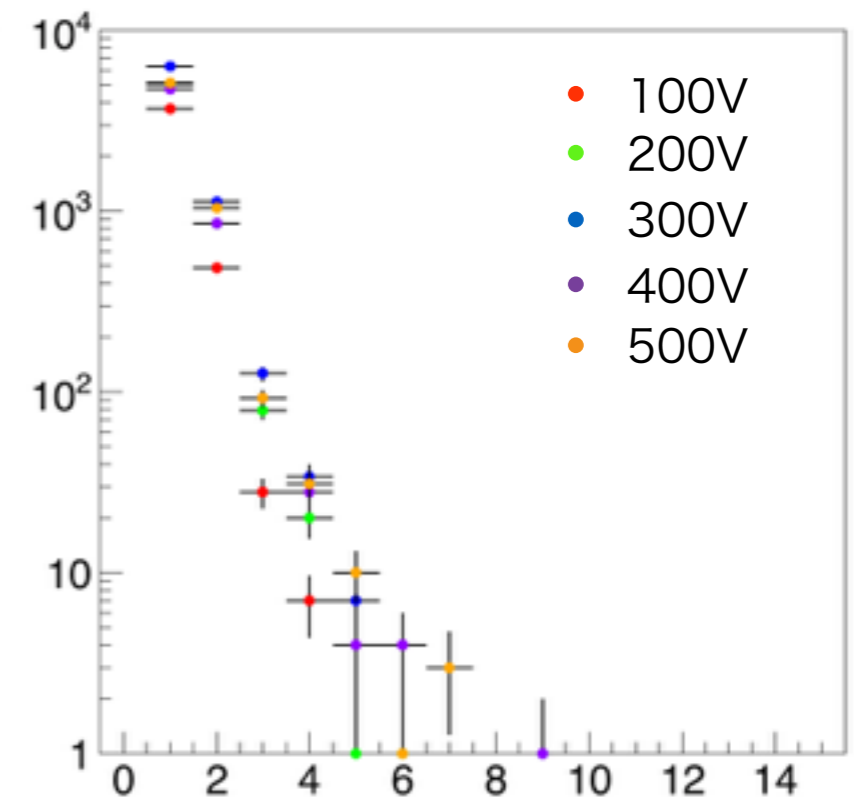
ToT分布



ToT分布(ClusterSize=1)



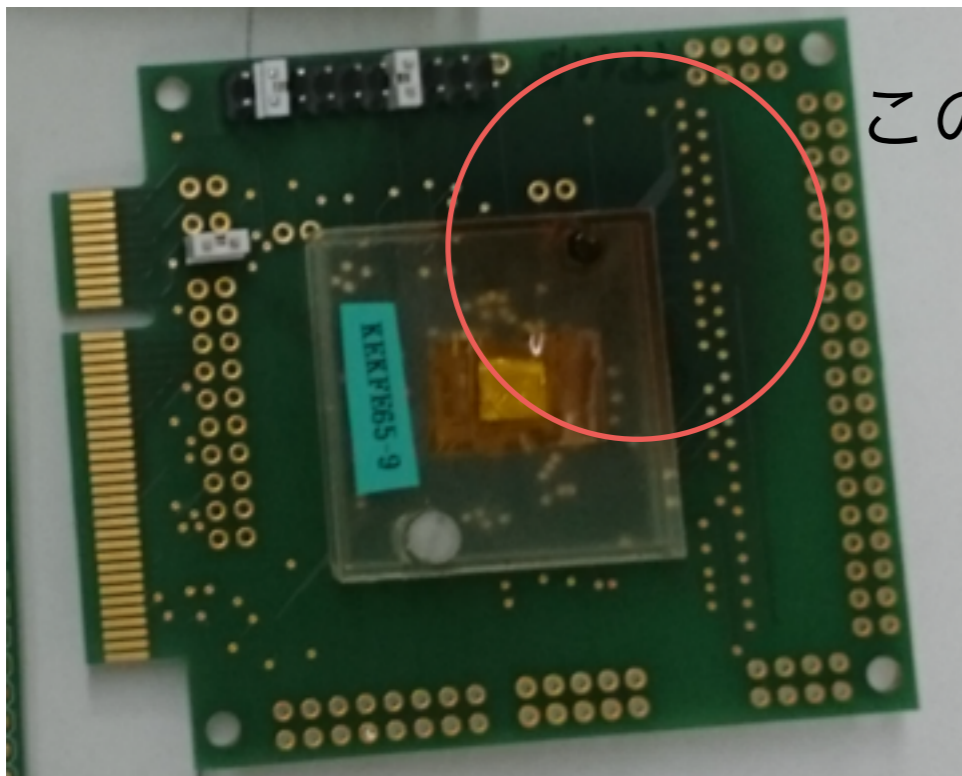
ClusterSize分布



照射の影響？

❖ 10月のCYRIC照射

- 照射Boxのトラブル？で照射位置がずれる問題が生じていた
- 全体での照射量は当初の予定の半分以下



このあたりが焦げている

| Sample | position | AI # | Target | Dose |
|-----------|----------|-------------------------------------|--------|----------|
| KEKFE65-6 | 下流 | 1,2,3(中央) 4,5,6(右上) 7,8,9(左下) | 3e+15 | 1.34e+15 |
| KEKFE65-8 | 中流 | 11,12,13 | 3e+15 | 1.18e+15 |
| KEKFE65-9 | 上流 | 14,15,16 | 3e+15 | 6.76e+14 |

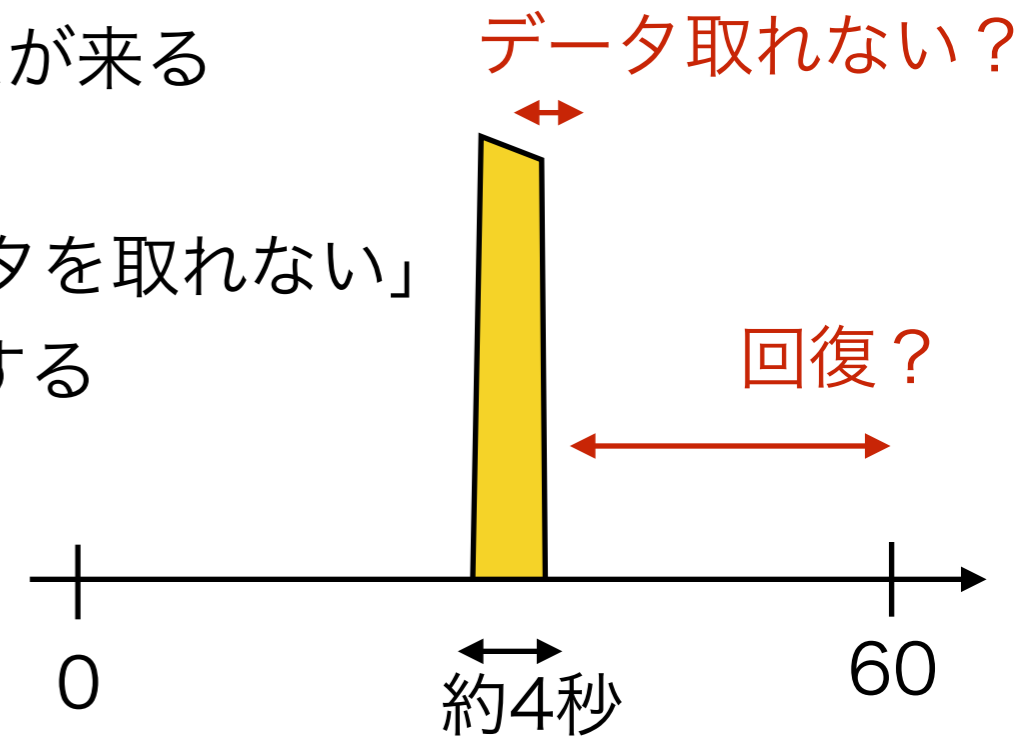
(鈴木くんのITkスライドより)

back up

スピル内での検出効率

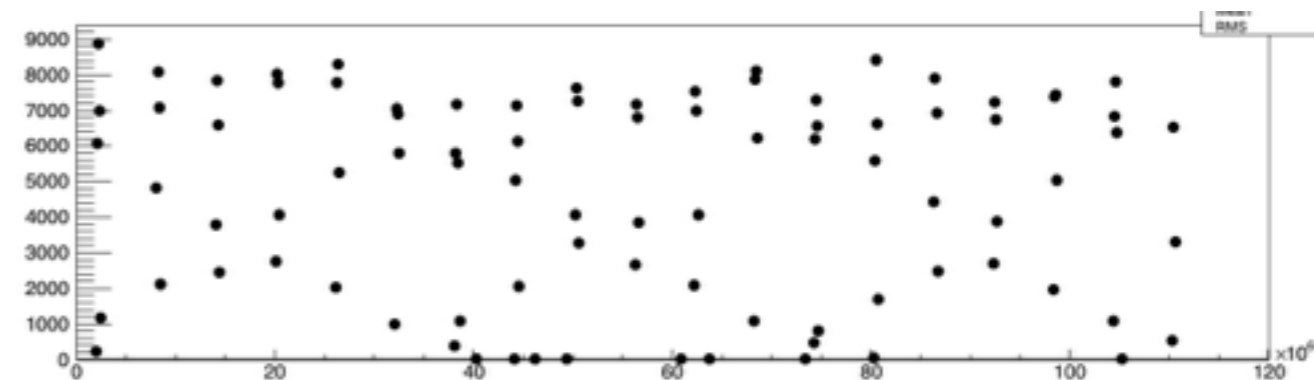
◆ スピル構造

- FNALテストビームでは、約60秒に4秒ほどビームが来る
(main injectorからビームラインに送られてくる)
- 「仮説：1度hitがあったピクセルはしばらくデータを取れない」
が正しい場合、1つのスピル内で検出効率が変わる
- 1スピルの周期は60.19秒だった



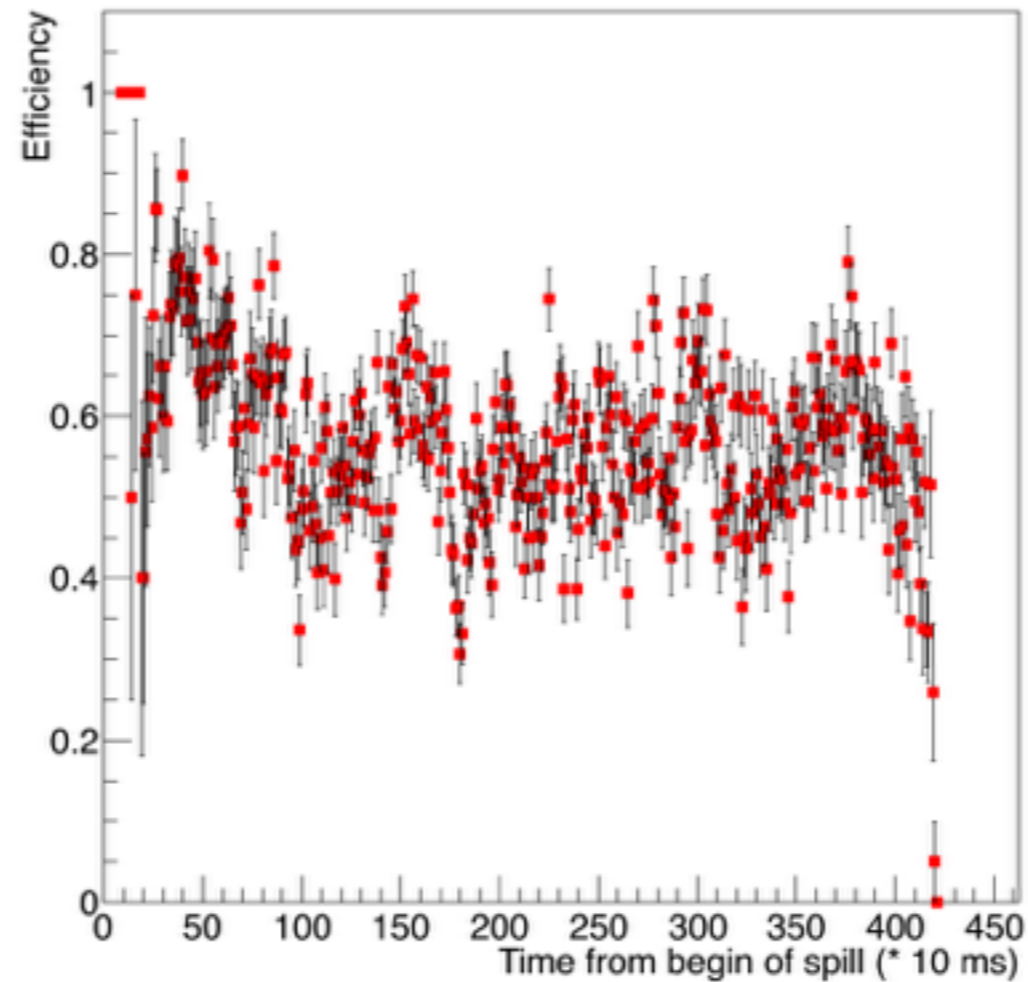
TimeStamp

- 100KHz、最大値16777215
- TLUが配っているTimeStampを用いてスピルを把握する
- TIMESTAMP(TimeStampの累計)を実装



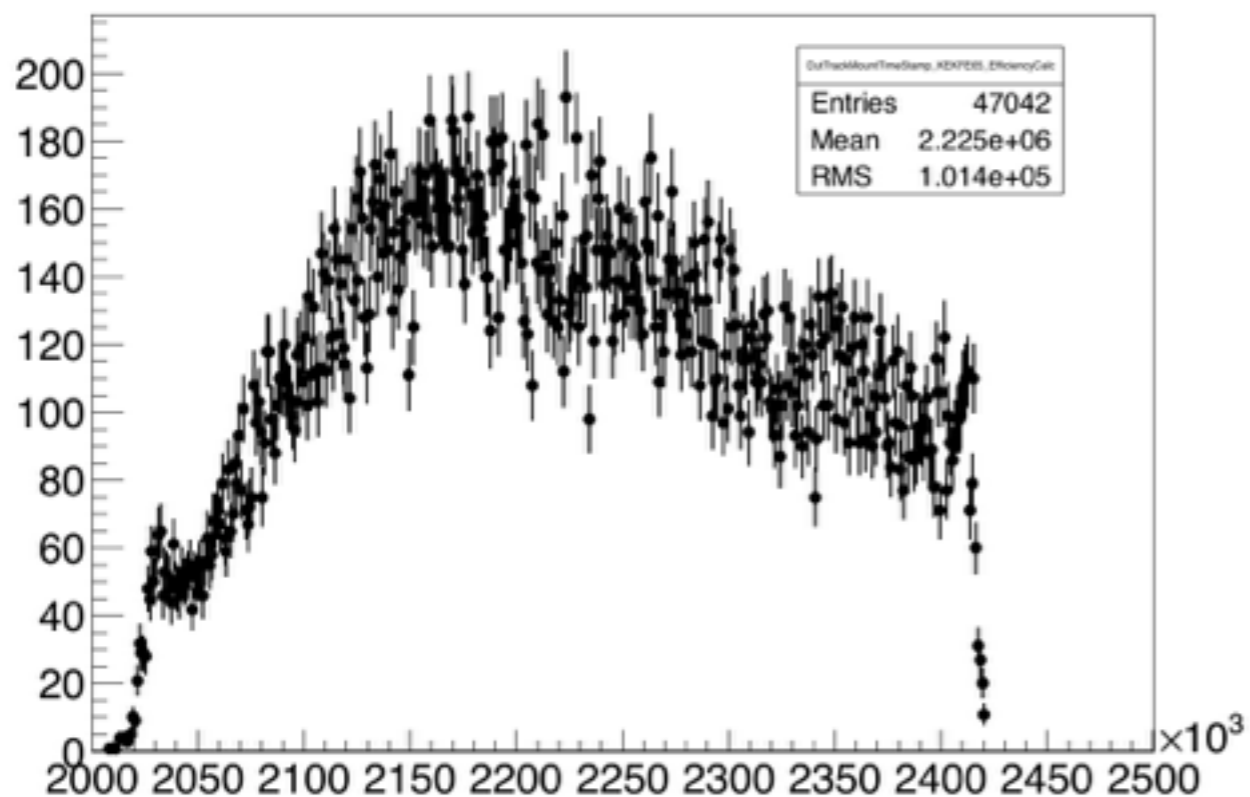
スピル内での検出効率

- 1つのスピル内での検出効率の変移を計算した
- 最初は検出効率が高いが、徐々に下がっている
- スピル初期でも検出効率は1でない？



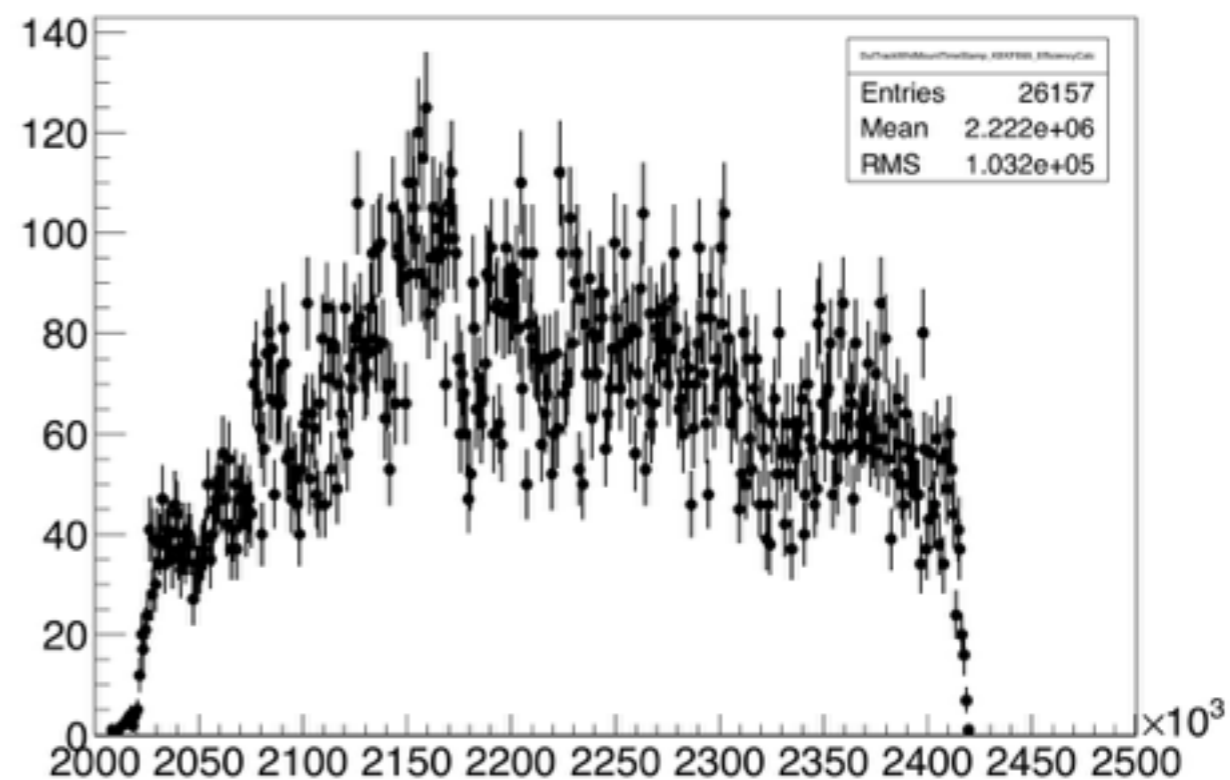
スピル内での検出効率

Track



TimeStamp (1=10us)

Hit

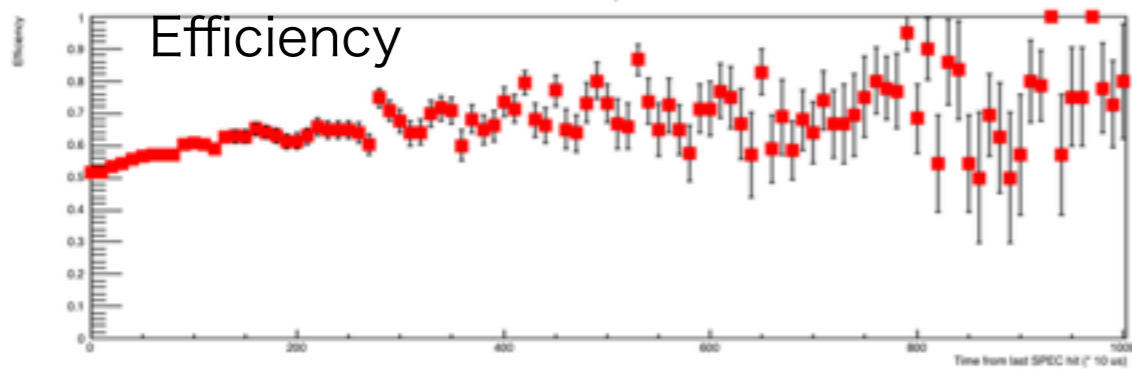


TimeStamp (1=10us)

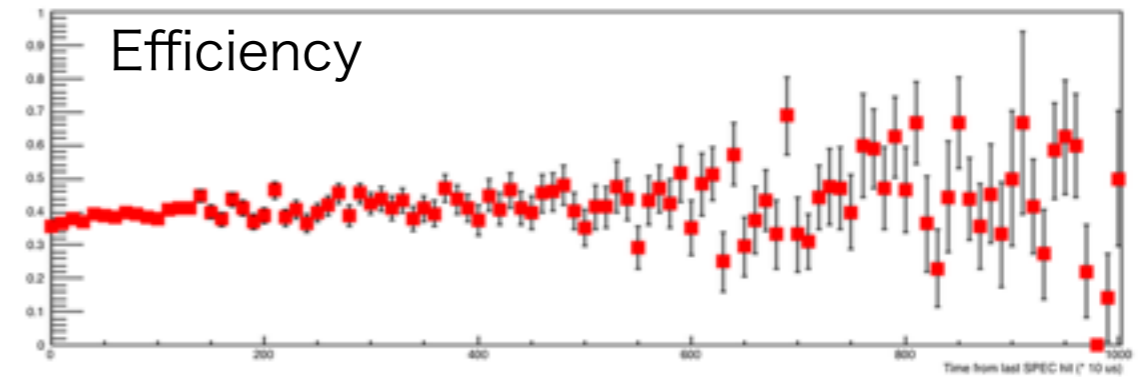
Investigations of Cause

◆ Timing issue (module unit)

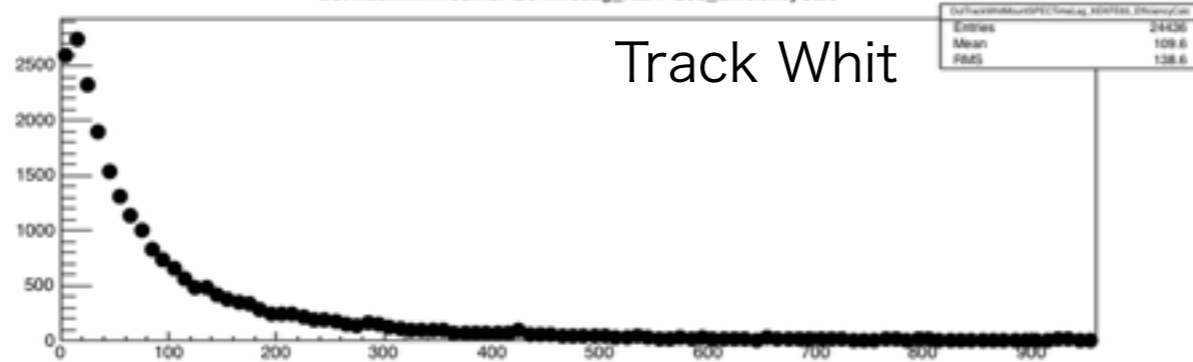
no bias



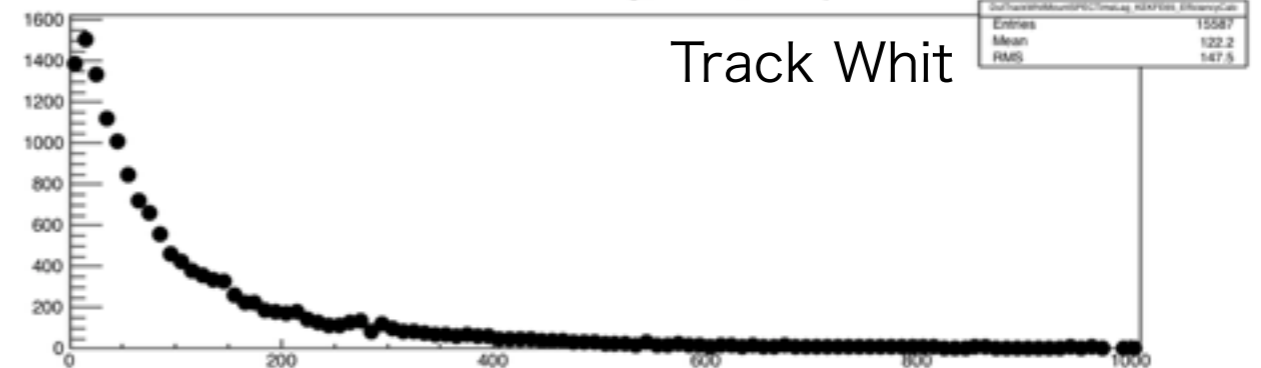
bias rail



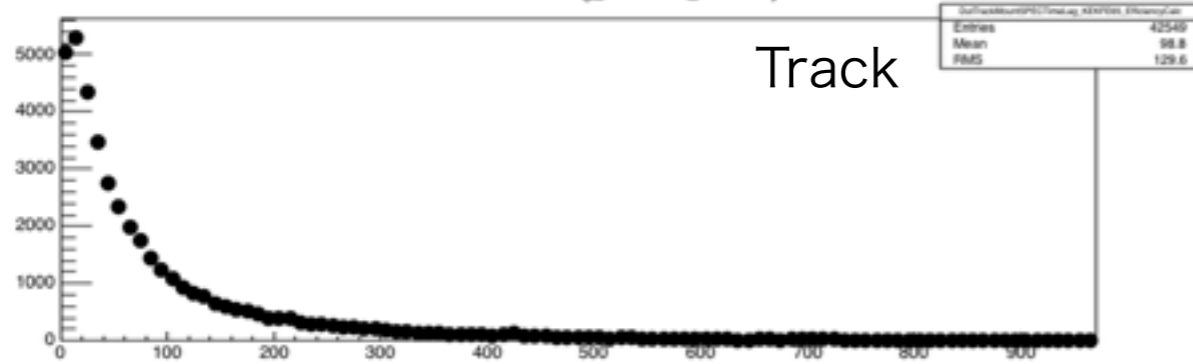
DutTrackWhitMountSPECTimeLag_KEKFE65_EfficiencyCalc



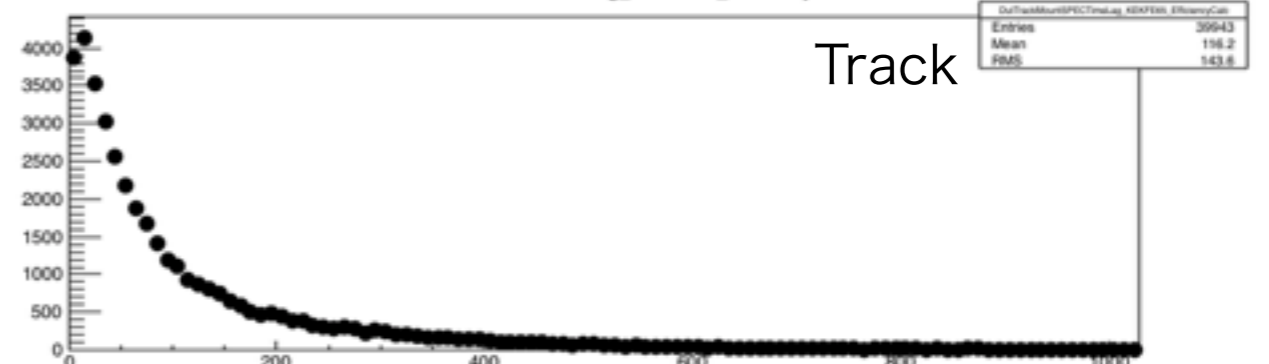
DutTrackWhitMountSPECTimeLag_KEKFE65_EfficiencyCalc



DutTrackMountSPECTimeLag_KEKFE65_EfficiencyCalc



DutTrackMountSPECTimeLag_KEKFE65_EfficiencyCalc



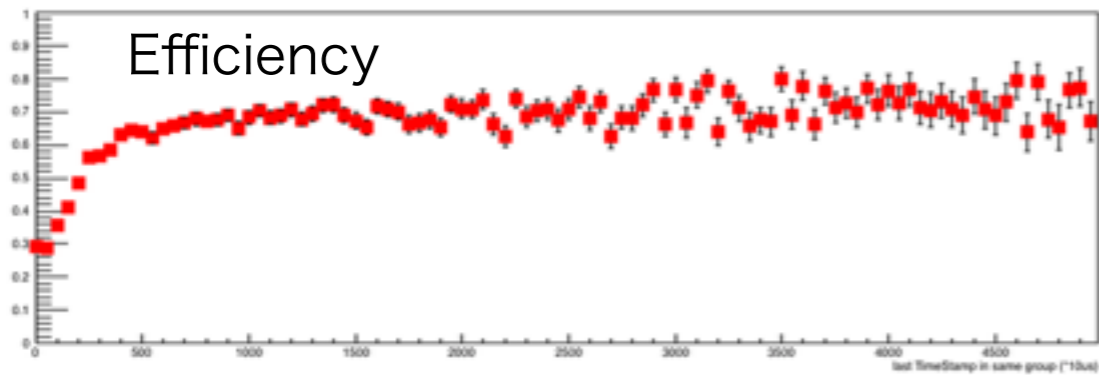
TimeStamp(*10us)

TimeStamp(*10us)

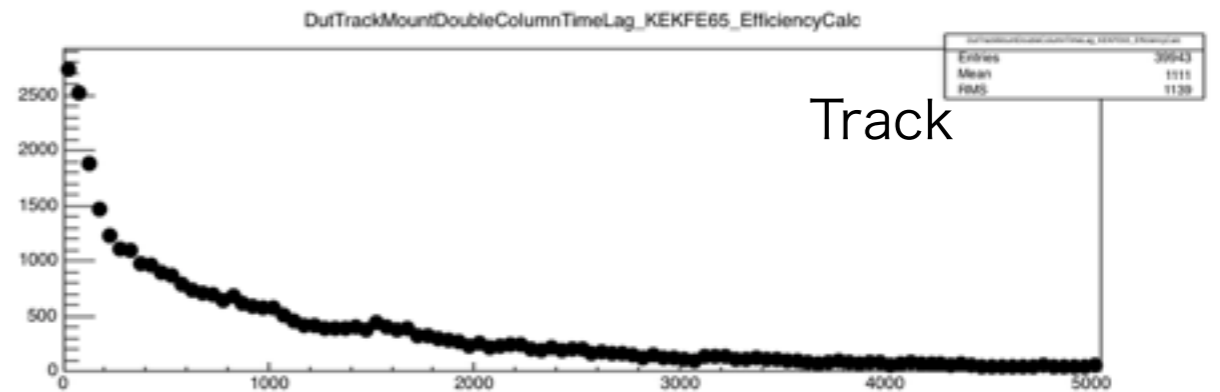
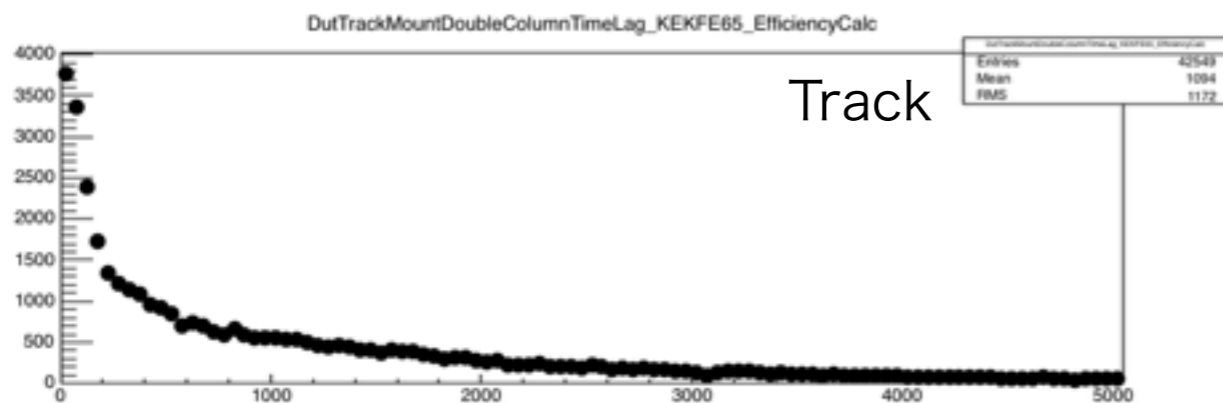
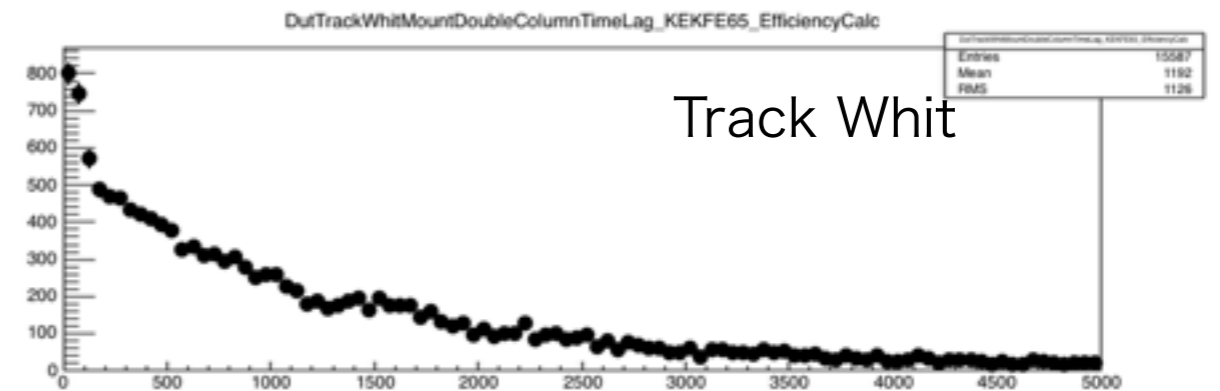
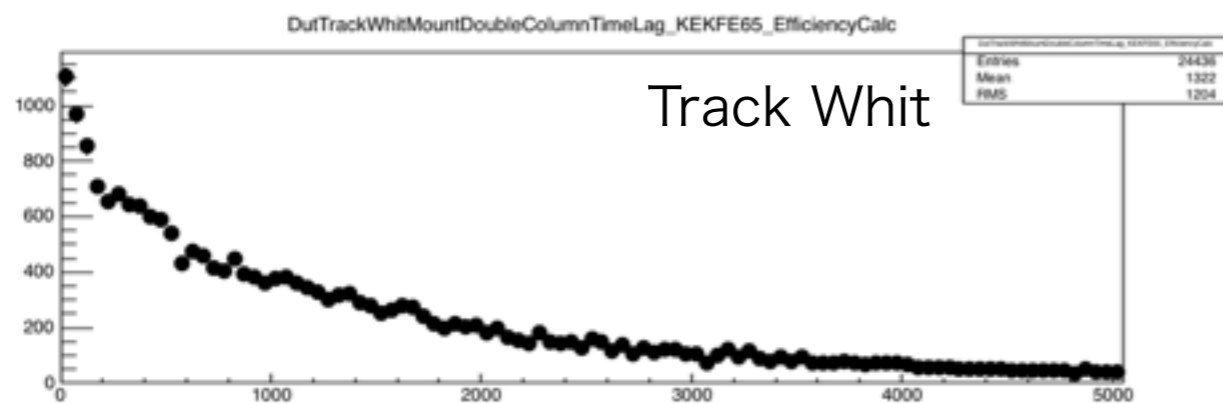
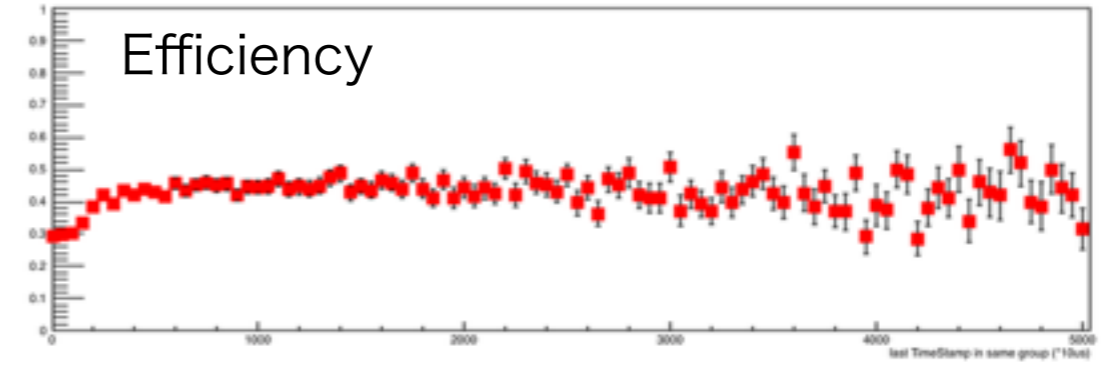
Investigations of Cause

- ◆ Timing issue (Quad column unit)

no bias



bias rail



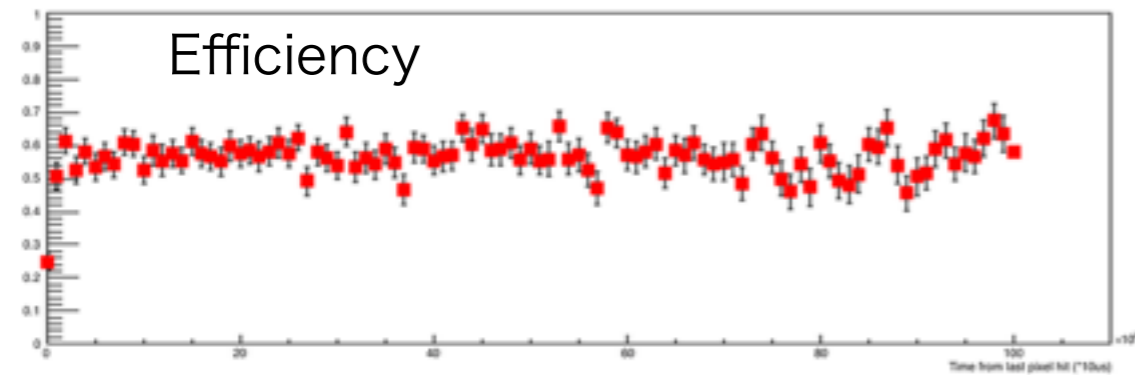
TimeStamp(*10us)

TimeStamp(*10us)

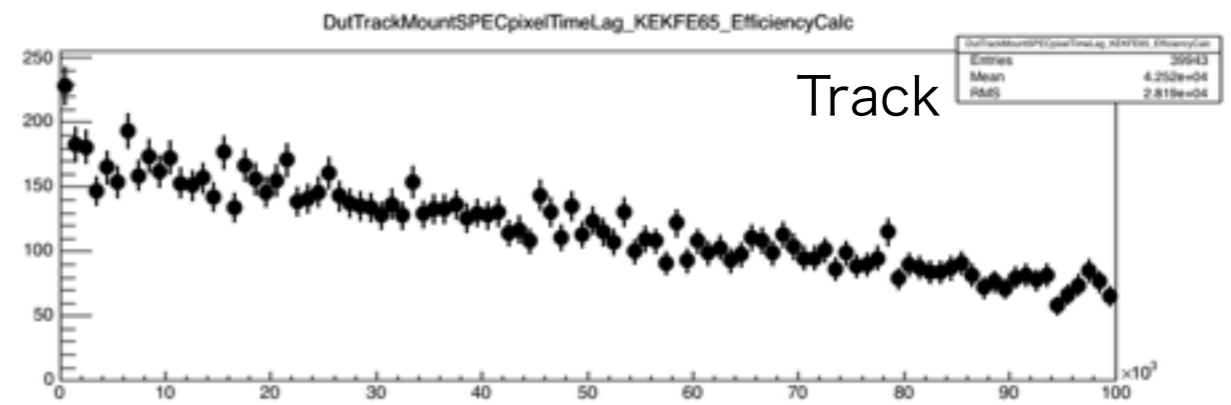
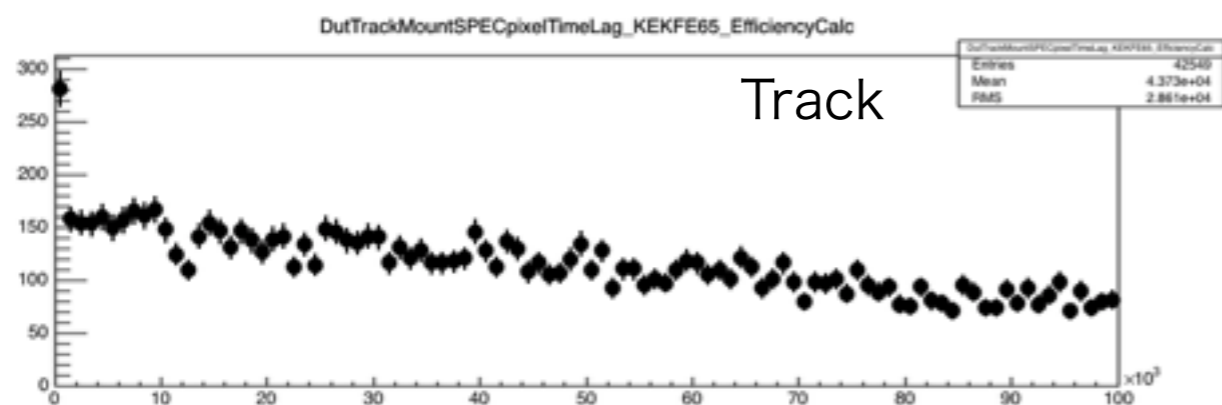
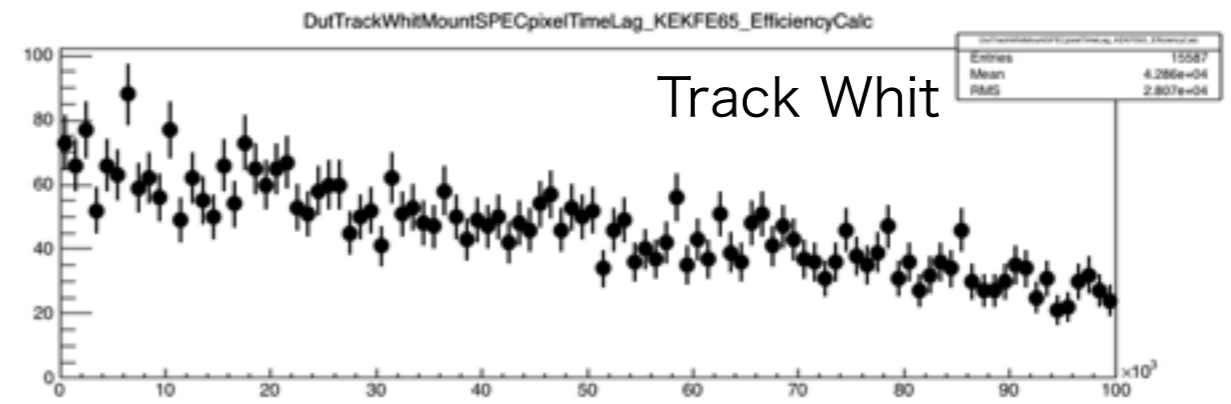
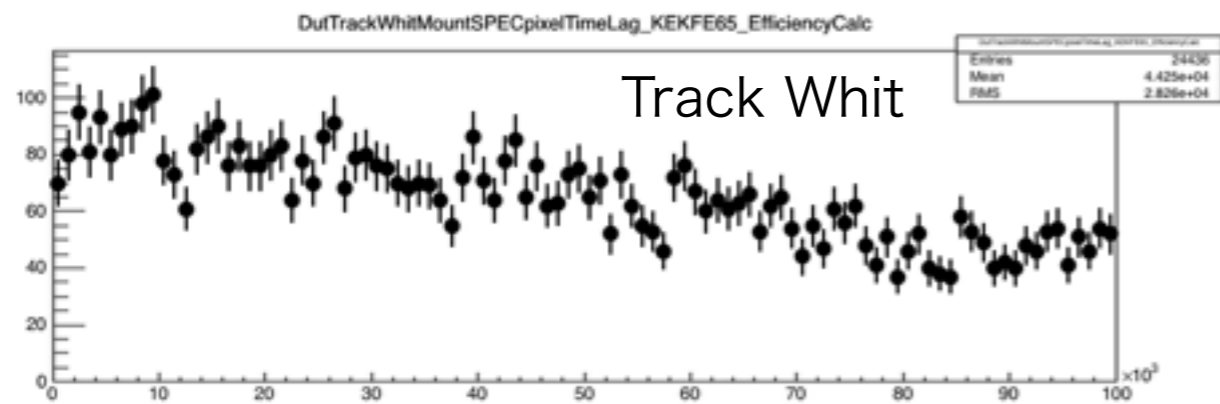
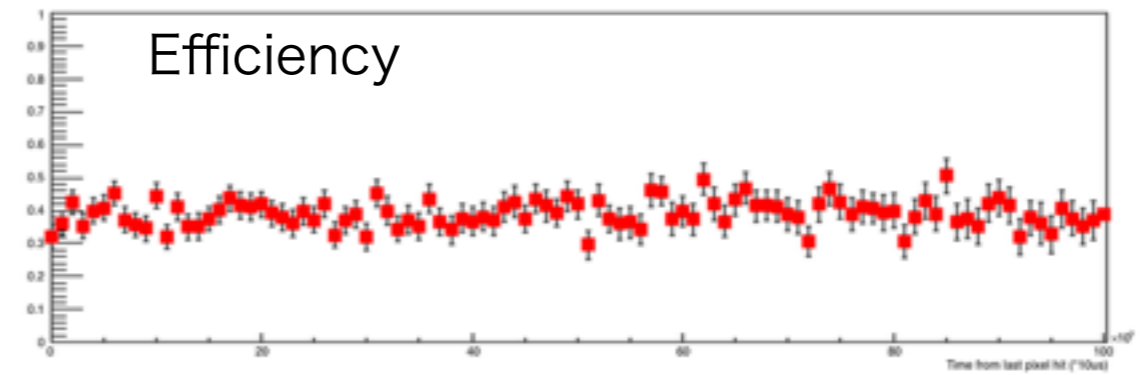
Investigations of Cause

◆ Timing issue (pixel unit)

no bias



bias rail



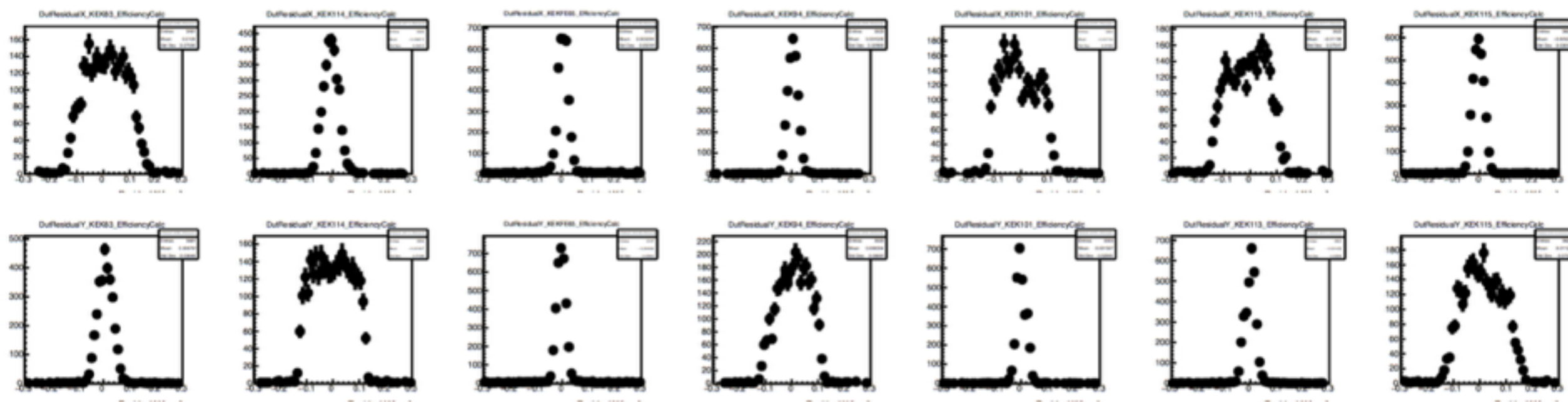
TimeStamp(*10us)

TimeStamp(*10us)

アライメント

◆ Residual分布

- ピクセルごとの特性を調べる上でアライメントは非常に重要
- 鈴木くんのalignfileを借りてアライメントを確認してみた
- 各ヒット位置の誤差を正しく見積もるように変更した(一律100um \rightarrow pixelsize/sqrt(12))
- 後ろ3枚のResidualが二又にならなくなった

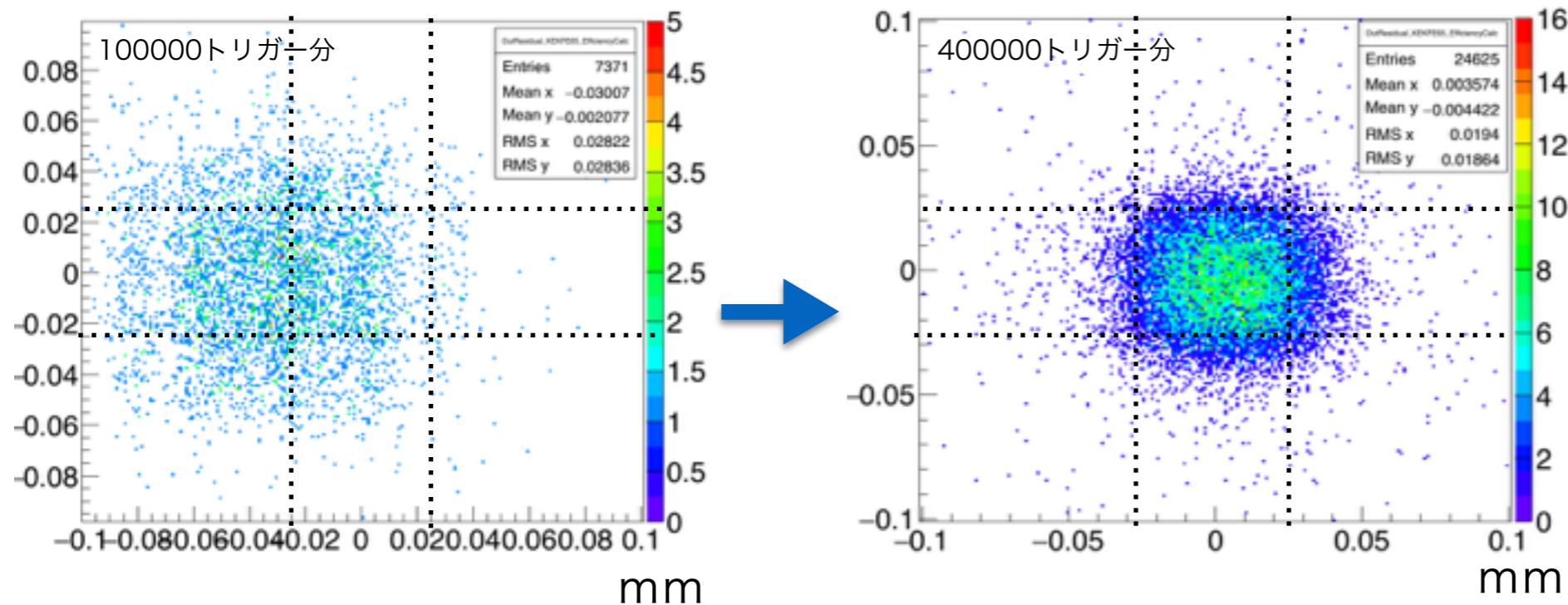


ビフォーアフター

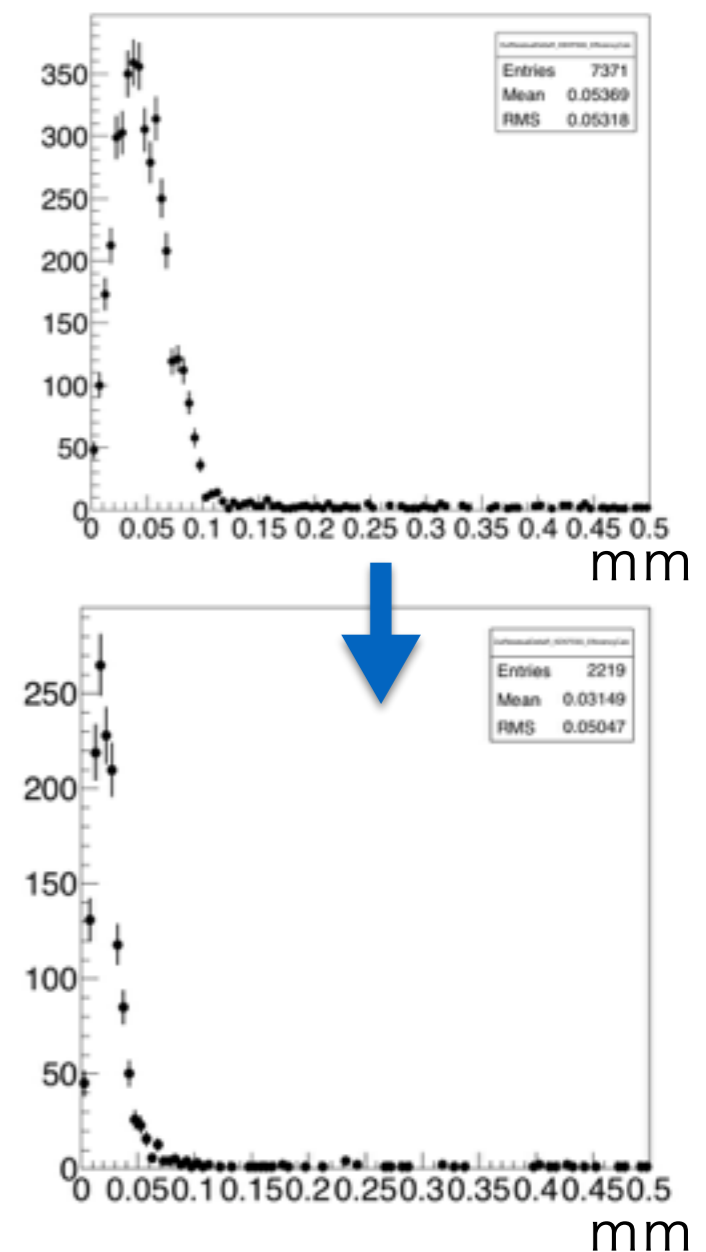
◆ TrackHitとHit

- TrackHitとHitの差を2次元、1次元(距離)で確認
- ほとんどのtrackでtrack通過位置とhit位置が同じピクセルに乗るようになった

Track hitとHit間の差(dx, dy)の分布



TrackとHit間距離の分布



ビームテスト

- ◆ 2017年2月@フェルミ国立加速器研究所
- ◆ 120 GeVの陽子ビームを照射
- ◆ センサーの検出効率を測定し、構造による性能を評価



ビーム軸

ビーム軸

Detector Under Test(DUT)

性能評価する検出器

テレスコープ

飛跡再構成用ピクセル検出器

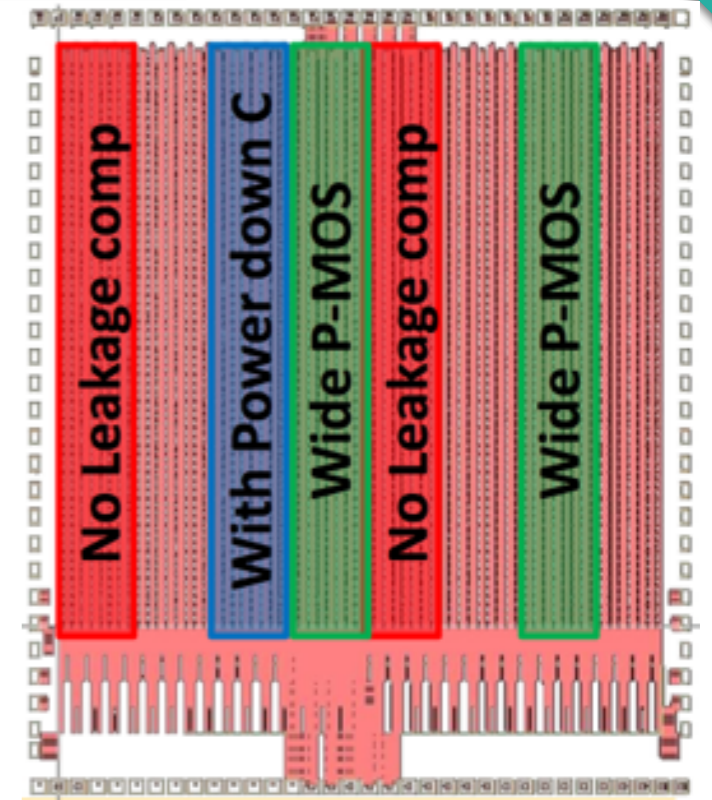
Trigger

シンチレータからトリガーを受け取り、DUT、テレスコープへ発行

FE65-p2

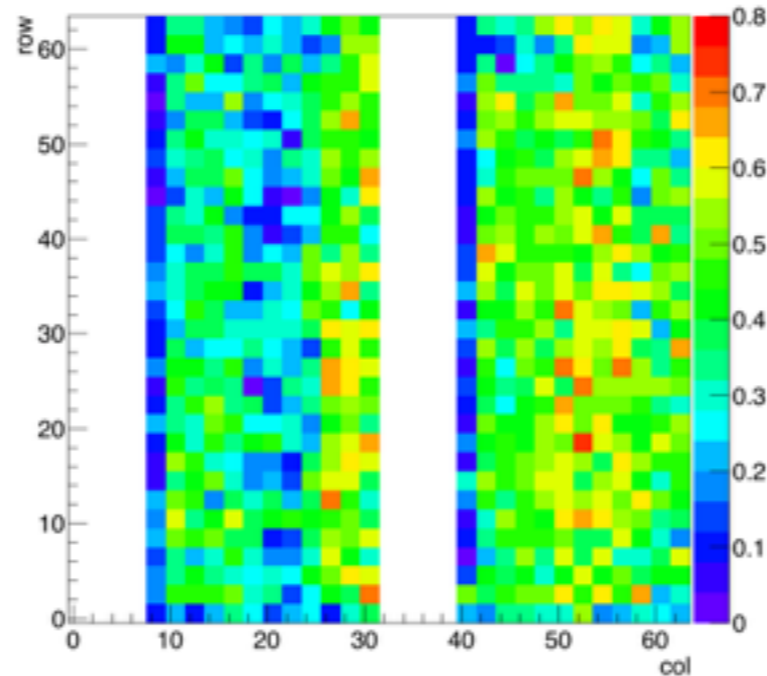
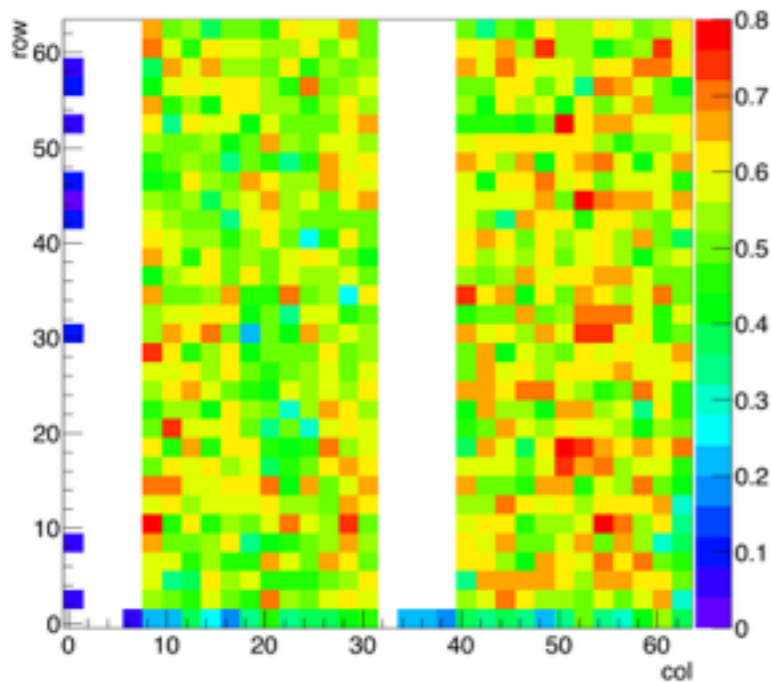
◆ FE65-p2

FE65-p2にはさまざまなテスト機能が搭載されている

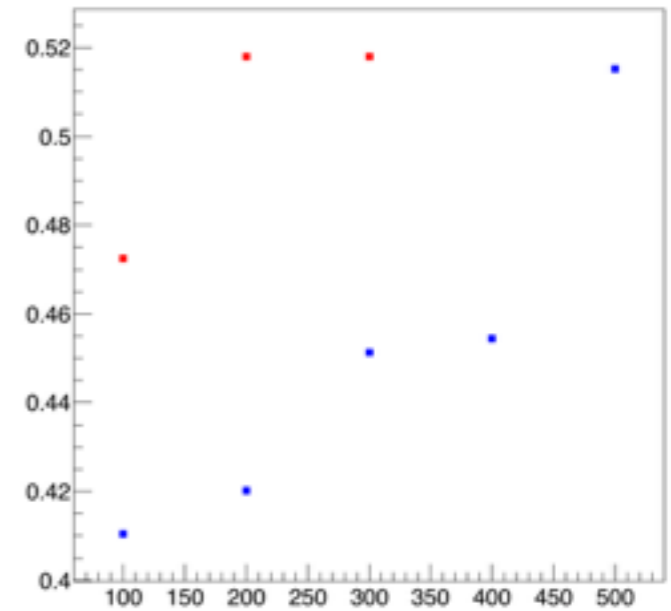


no bias

bias rail



Voltage vs Efficiency

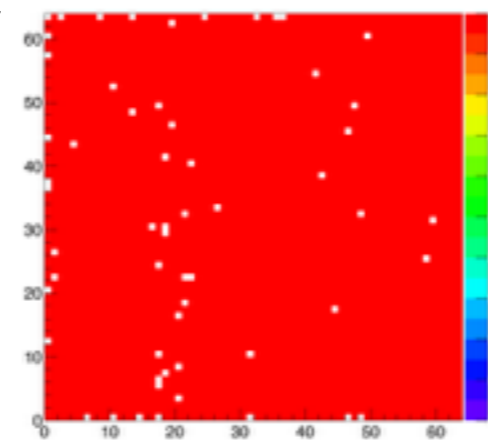


データ

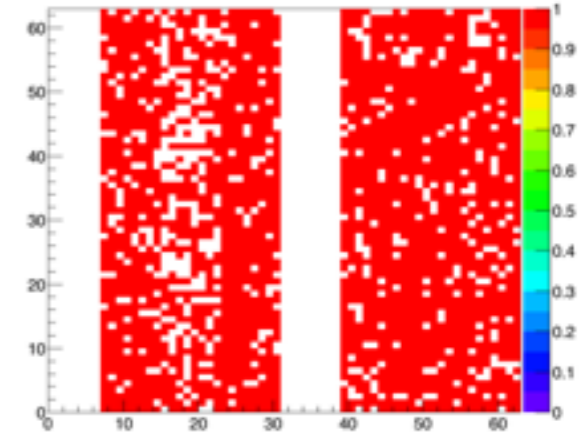
◆ Enable pixel

- Tuningの際にノイズが多かったピクセルはマスクしてデータを取らないようにする
- Bias rail有りのセンサは一左端、中央にノイズが多い領域があり、マスクした

no bias



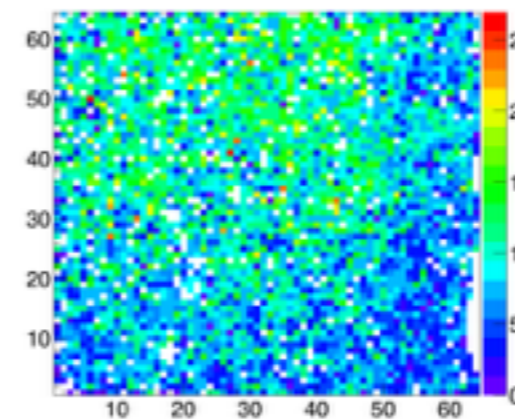
bias rail



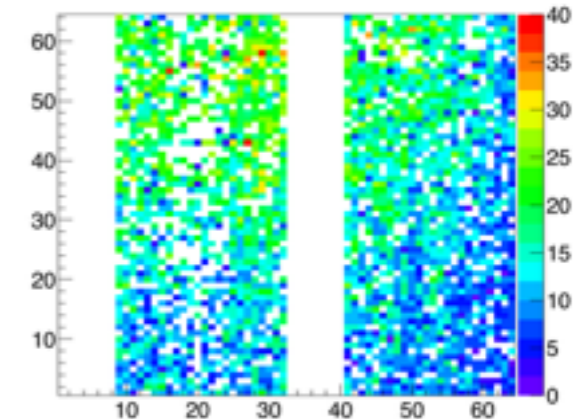
◆ Hitmap

- 2枚とも正常に動作していた

no bias



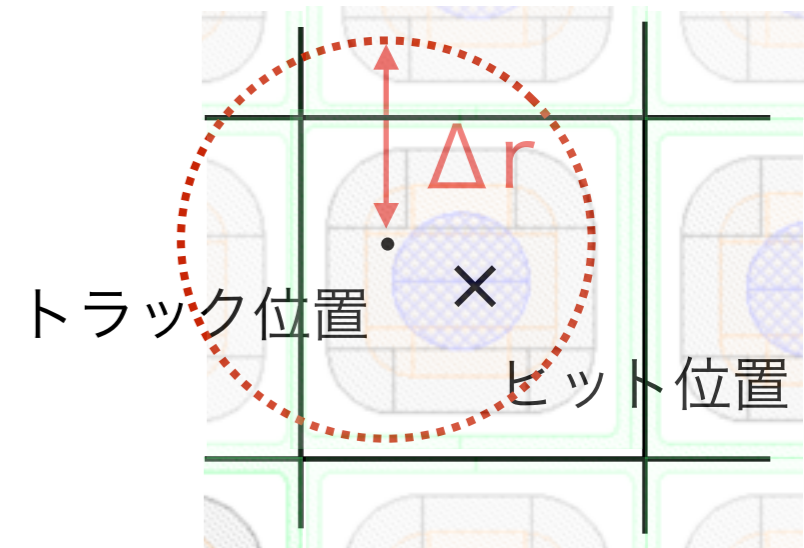
bias rail



解析2：検出効率

◆ 飛跡に対応するヒット

- テレスコープのみで再構成したトラック位置を
- 中心とした $200\mu\text{m}$ の範囲内でヒットを探す
→ ヒットがあれば飛跡に対応するヒットとみなす



◆ 検出効率の定義

$$\text{検出効率} = \frac{\text{センサー内に対応するヒットがある飛跡数}}{\text{再構成した飛跡数}}$$

Bias railあり300Vのrunで検出効率を算出した→48%

(原因)・ノイズマスクが反映できていない

・300Vでは検出器が全空乏化されていない

今後この点を改善し、検出効率を求めていきたい

