

# MIPピークへの放射線損傷の影響

2025/2/4 NWU 石垣優衣

@INTT JPMT

# MIPピーク解析について

[テーマ]ppデータを用いたMIPピーク位置の時間変化の研究

放射線ダメージがシリコンセンサーを完全空乏化するために必要なバイアス電圧を増やす。Run24のpp衝突では、INTTのバイアス電圧は一定で稼働(100V)

放射線ダメージが大きければMIPピーク位置が下がるはず。

[現状の進捗]

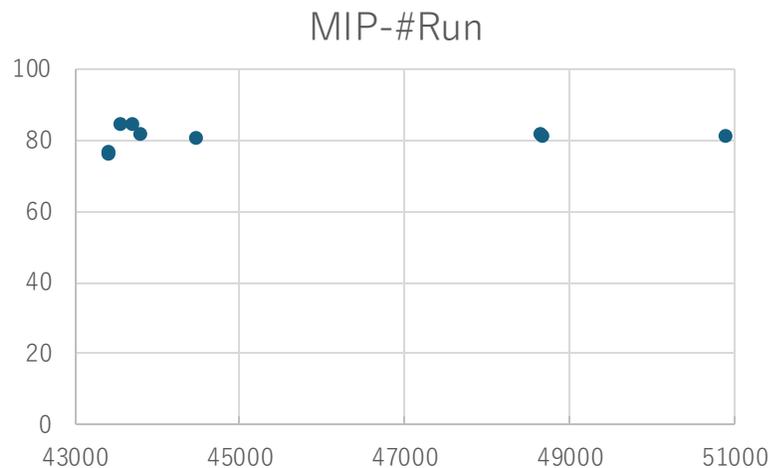
2024年4月～8月までのデータを用い、放射線によってFPHXチップの劣化が起きているのかを調べている。

ワークショップでの結果では、RUNの数も9つ程と少なく、またexcelを用いた結果だった。今回rootを用いてプロットを作成・アップデートして、RUNの数も290ランほど増やした。(その後イベント選定を行い結果的には164ラン残った)

# 前回までの進捗

## 扱っているデータ

- track  $\theta = 90 \pm 10$
- $-10 < z_{vtx} < 10$
- トリガーモード
- エントリー数1000個以下のイベントを除外
- z\_vtx\_finder



2024.5.21

2024.8.9

MIPの位置が 75-85 でふらついている。ピーク位置の大きい変化はない。

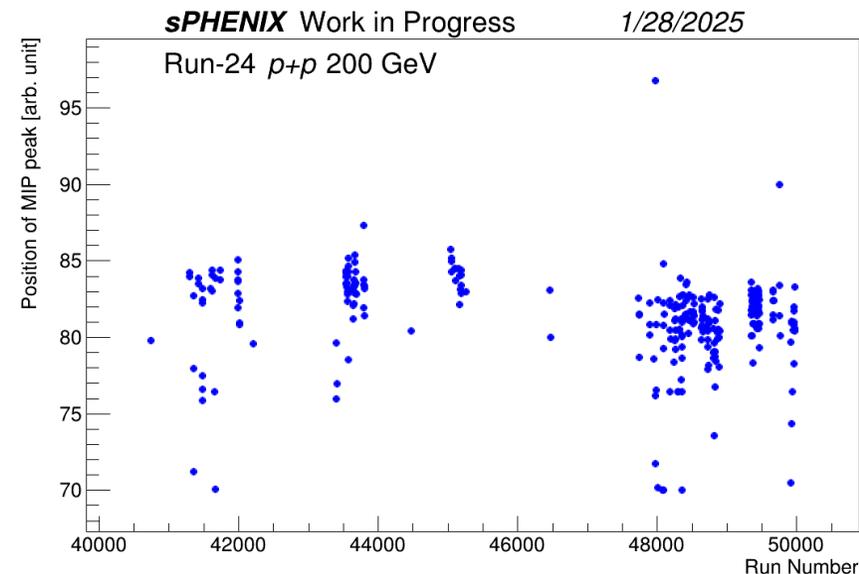
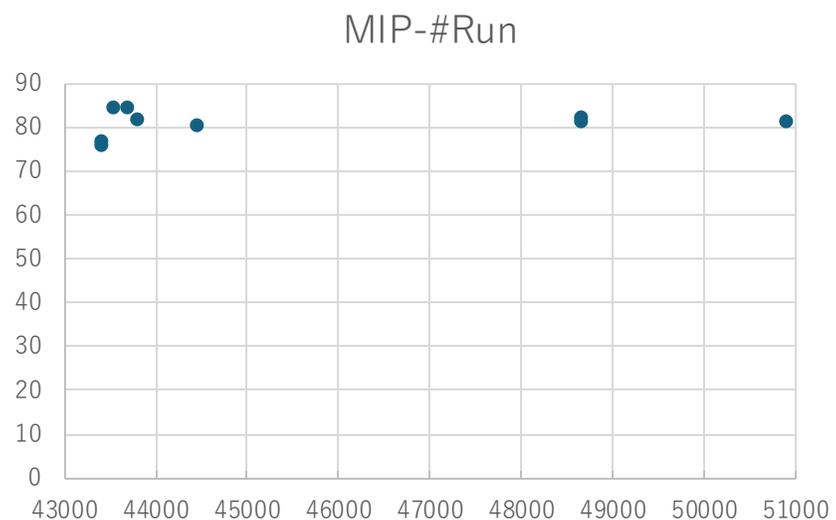
# 変更点・試したカット

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$

# 変更点・試したカット

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$

# データ点を増加

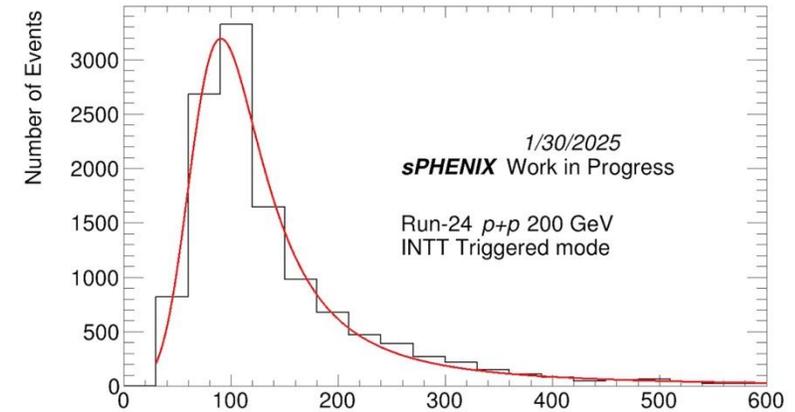
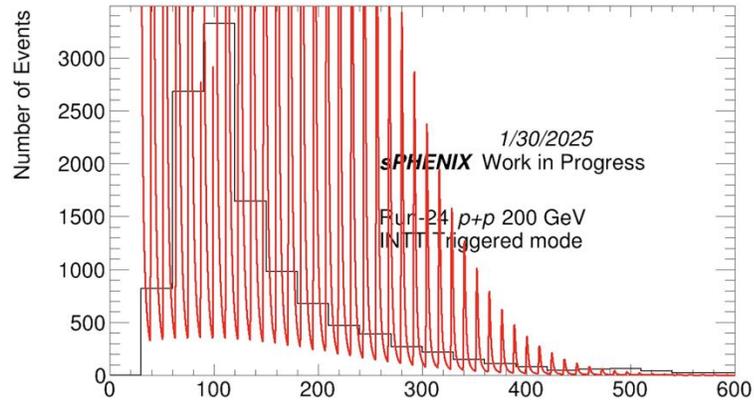


- Rootでプロットを作成して、sPHENIXフォーマットを取り入れた
- 4月中(40953~42207)から18ラン、5月以降から270ラン分増やした。
- ばらつきが大きい
- MIP=70のデータが多い

# 変更点・試したカット

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$

# Fitの見直し



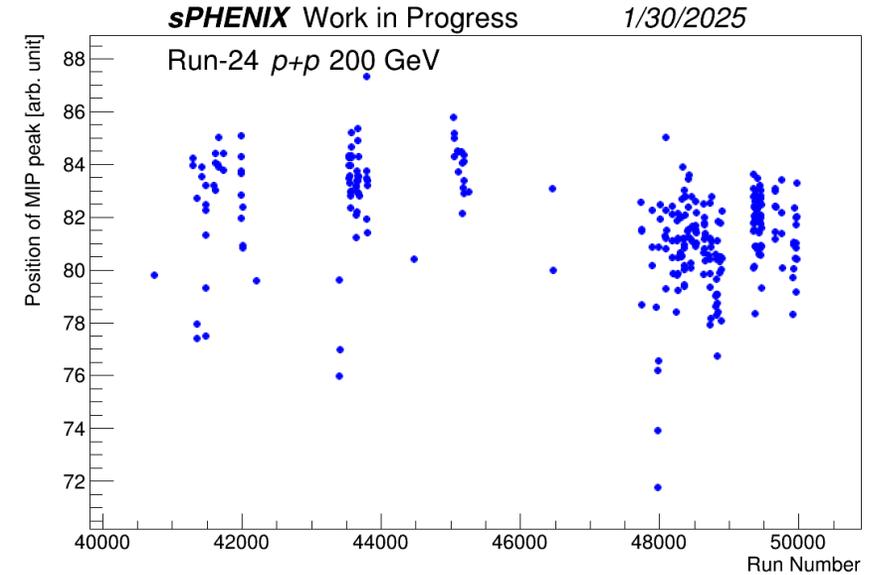
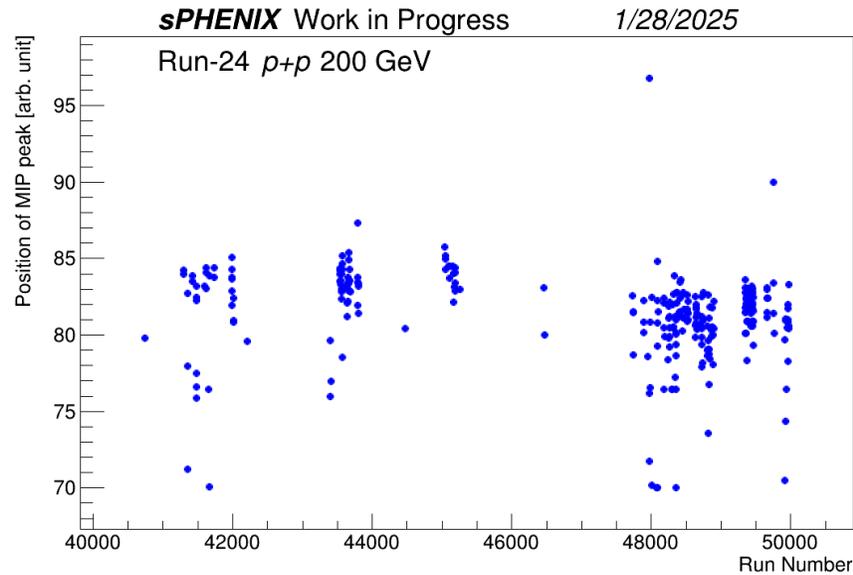
MIP=84

```
Fitting...
FCN=1897.14 FROM MIGRAD STATUS=CONVERGED 1175 CALLS 1176 TOTAL
EDM=2.04373e-07 STRATEGY= 1 ERROR MATRIX ACCURATE
EXT PARAMETER STEP FIRST
NO. NAME VALUE ERROR SIZE DERIVATIVE
1 Width 5.08905e-01 4.08326e-03 3.20743e-05 2.28949e-01
2 MP 7.00000e+01 3.00451e-02 1.32319e-03** at limit **
3 Area 1.09084e+06 2.20906e+04 2.20343e-06 -5.53534e-01
4 GSigma 1.20124e+02 7.83765e-03 1.80586e-06 -5.63372e+00
Fitting done
```

MIP=70と極端に小さいものを調べると、MIPがlimitに達しているものが見つかったのでfitの際のMIPの初期値を70→60に変更、畳み込みのガウスの初期値も0→3に変更(ガウスの幅が小さすぎると振動するから)

→MIP=70でlimitが来ていたデータがなくなった

# Fitの見直し(結果)

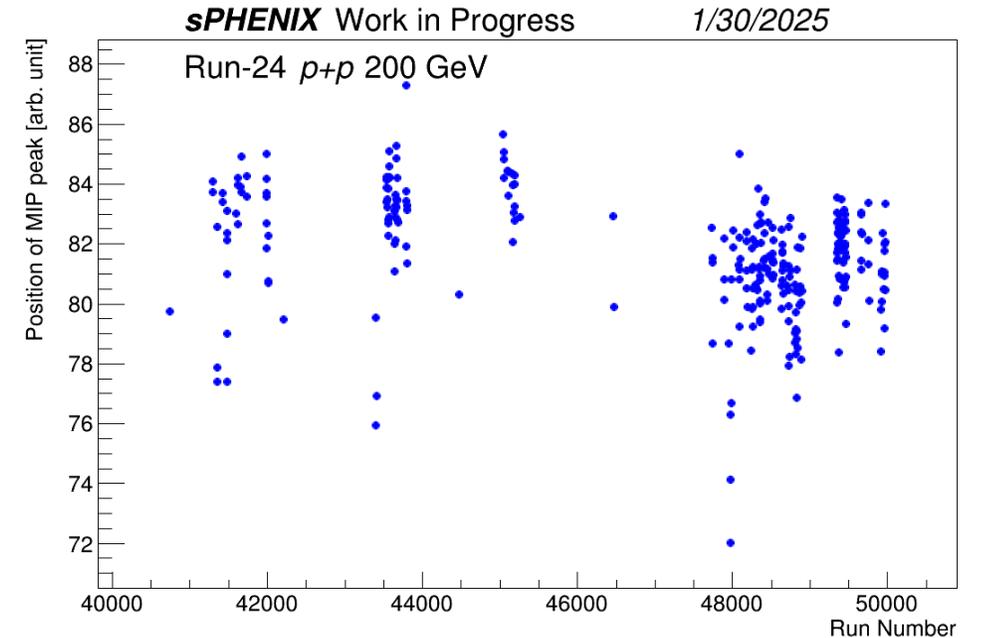
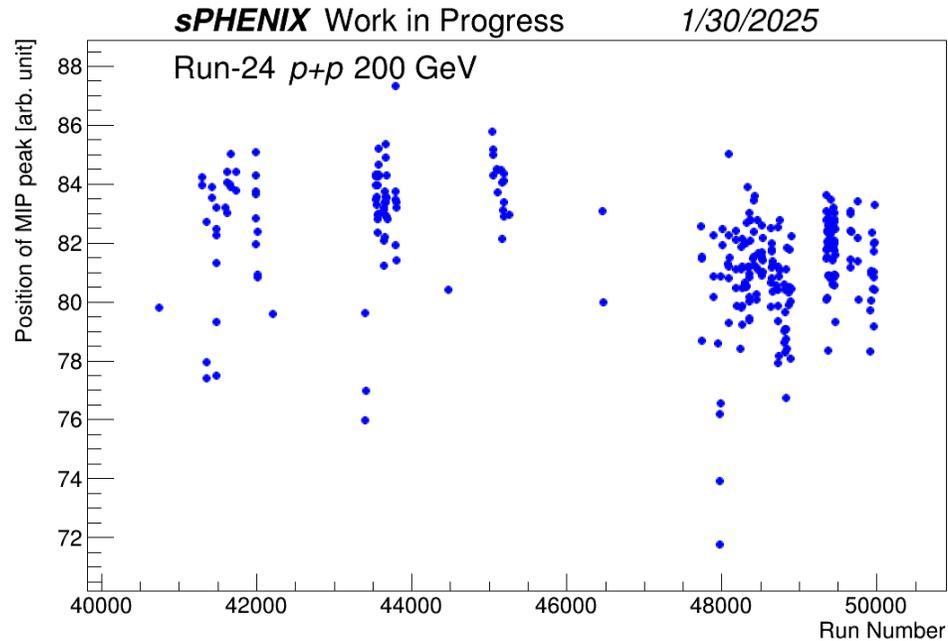


- MIP=70でlimitしていたデータ点が70付近から消えた(嬉しい)
- 依然としてデータのばらつきが気になる

# 変更点・試したカット

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$

# Vertexの範囲の変更



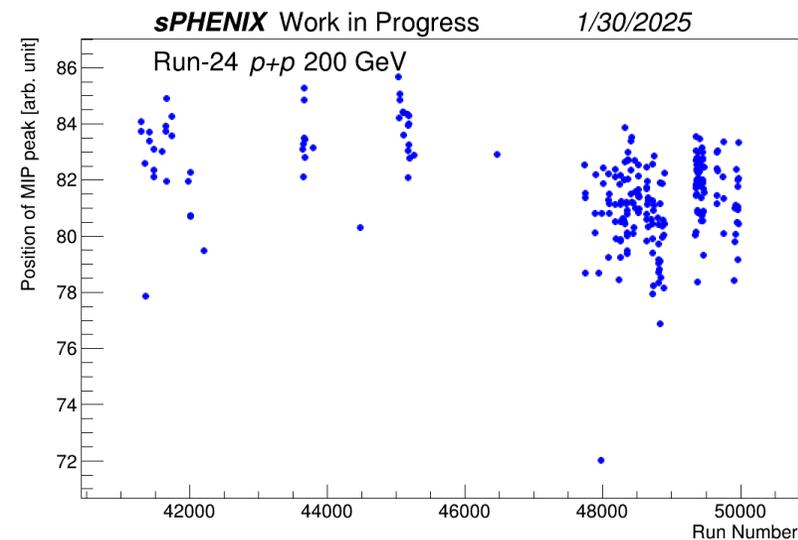
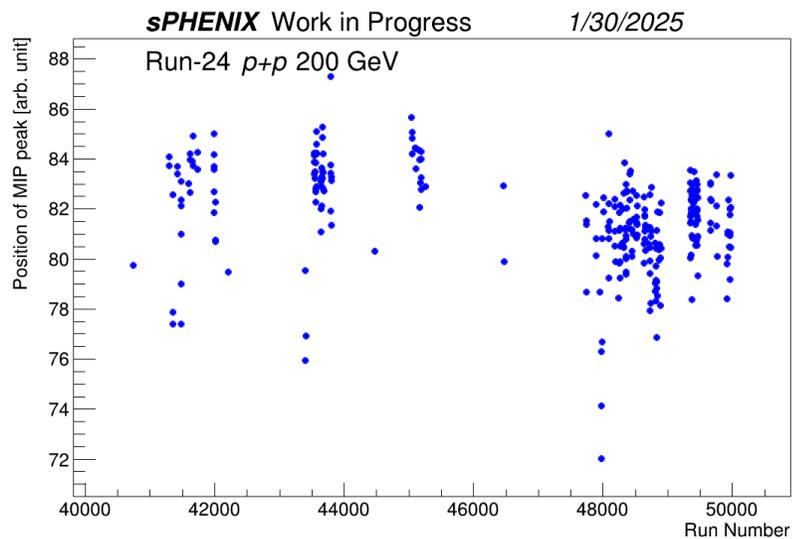
- $Z_{\text{vertx}}$ の取り扱いが微妙なので、範囲のカットをかけてみた
- $(-10 < z_{\text{vtx}} < 10)$
- 大きな変化はなかった。

# 変更点・試したカット

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$



# イベント数1000個以下のランを除去(結果)

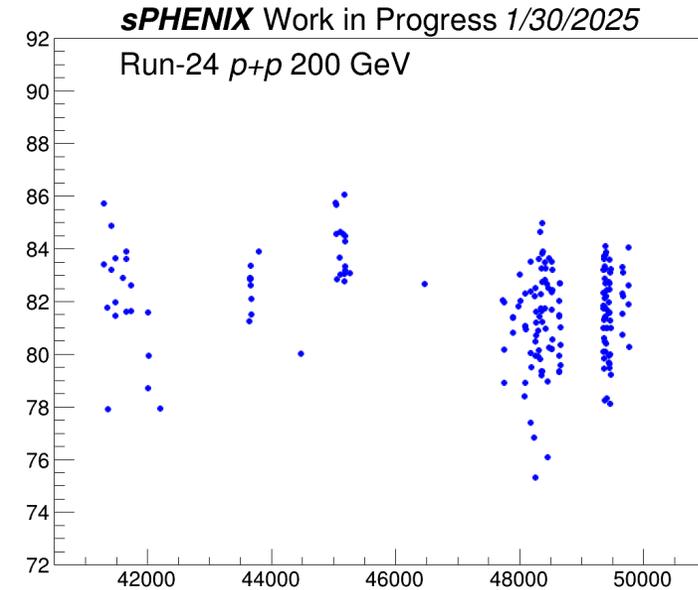
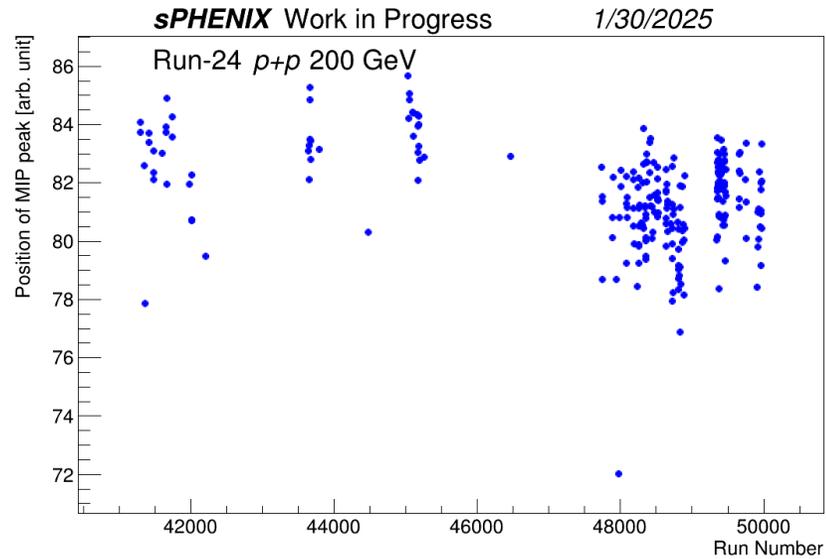


- 4月5月付近のデータが消えた
- MIPピークの値が小さい点が消えた

# 変更点

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- イベント数1000個以下のイベントを除外
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) < 0.45$

# z\_vertex finderの適用



- z\_vertex\_finderを用いたデータ処理を糠塚さんをお願いした結果、RUNの総数は183になった。
- ありがとうございました。

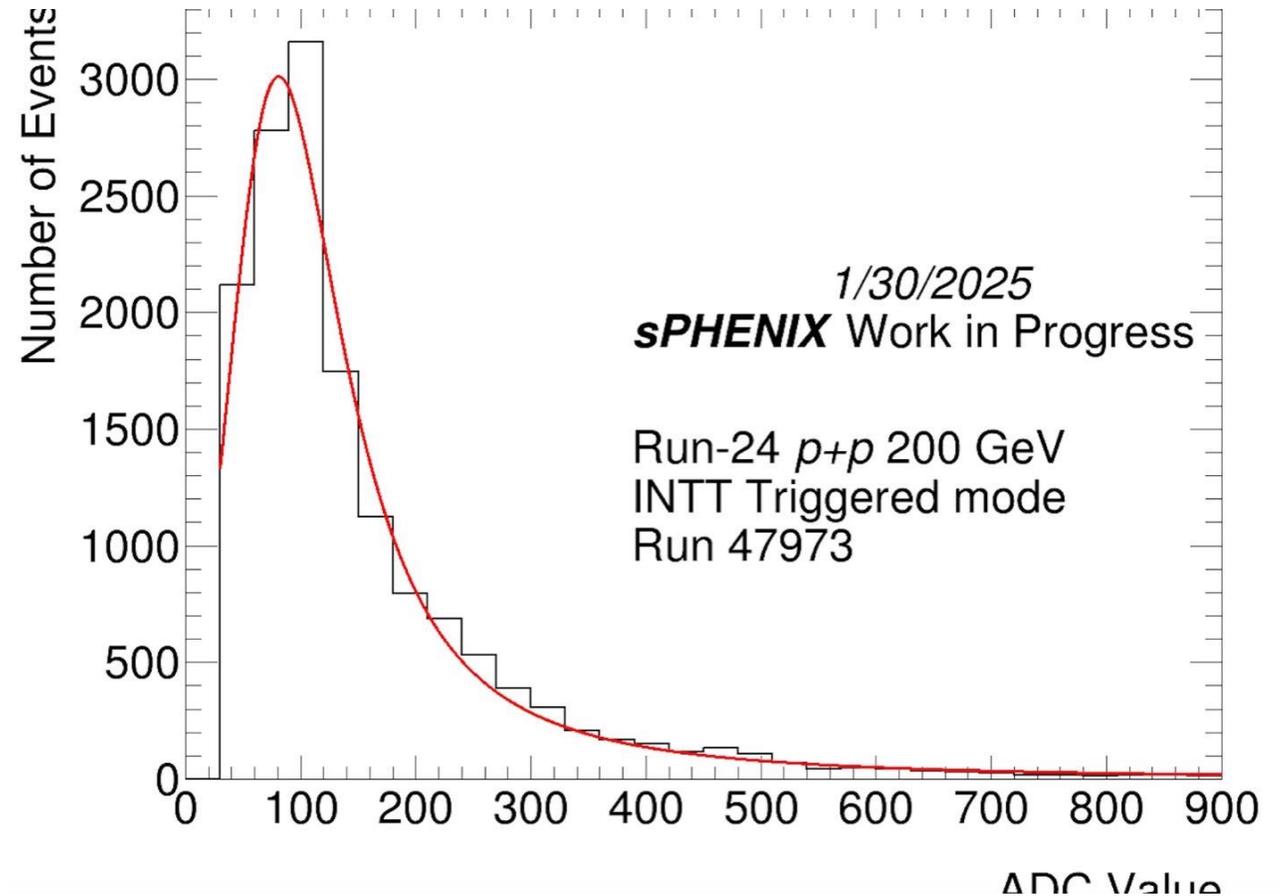
# 変更点

- データ点を増加させた(40953~42207の中から18ラン、それ以外から270ラン分)
- Fitに用いるパラメータの見直し
- Vertexの範囲の変更： $-10 < z_{\text{vtx}} < 10$
- トリガーモードだけのランを用いた
- イベント数1000個以下のイベントを除外
- `z_vtx_finder`を使用：`z_vertex`の決定精度向上
- $(1\text{bin})/(3\text{bin}) > 0.45$ のデータをカット

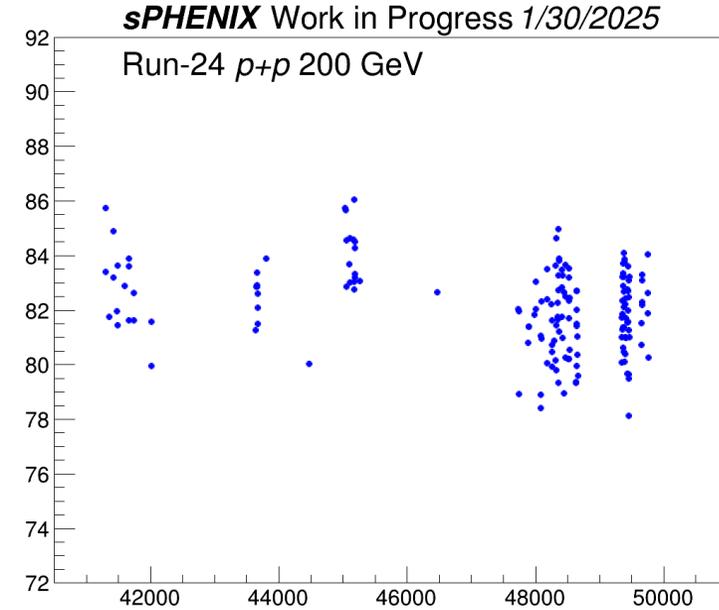
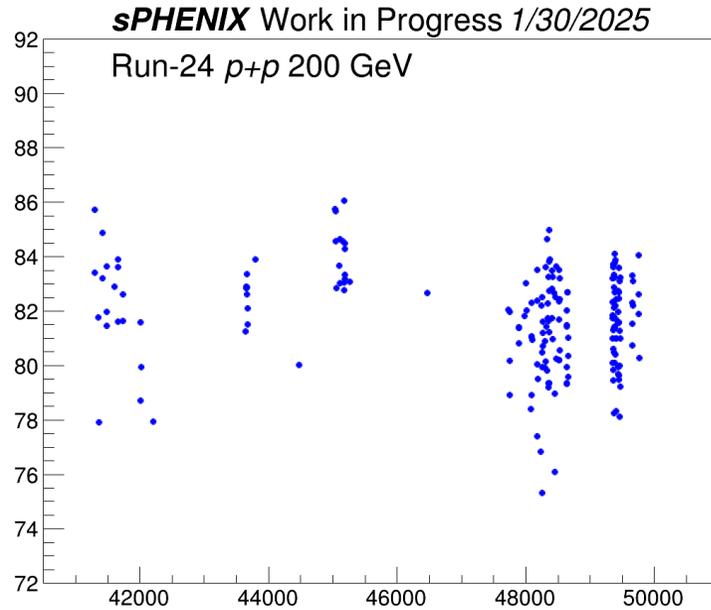
$$(1\text{bin})/(3\text{bin}) < 0.45$$

- フィットがうまくいってない
- 1bin目のノイズが高くて引っ張られている

→1bin目のノイズが高いものを検出器からのノイズが多いランとしてカットした



# (1bin)/(3bin) < 0.45 (結果)



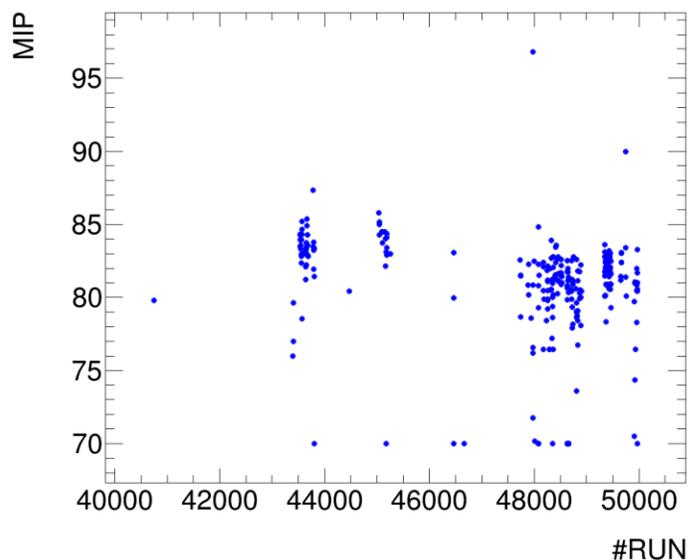
- RUNの総数183→164(結構減った)

# 結果

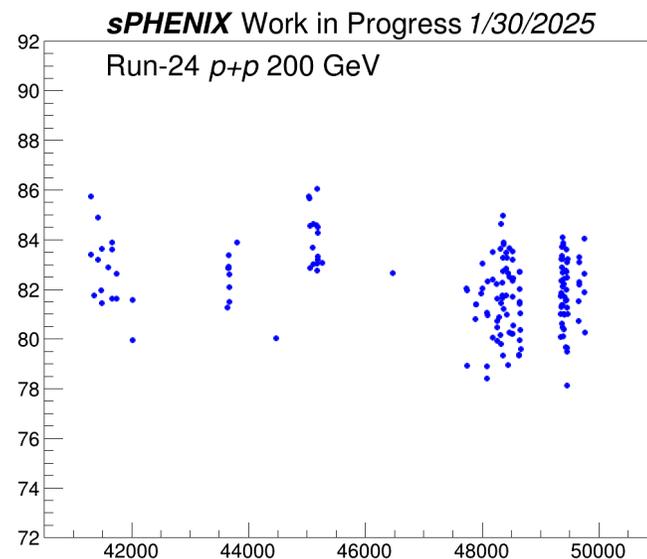
今まではMIP値がばらけていて70-85まで分布が広がっていた。より綺麗に分布を見るために、ランの選定やフィットの見直しを行いつつ、RUNの数を増やして分布を見た。

## 扱っているデータ

- track  $\theta = 90 \pm 10$
- $-10 < z_{\text{vtx}} < 10$
- トリガーモード
- エントリー数1000個以下のイベントを除外
- z\_vtx\_finder



修正前



修正後

# まとめと課題

## [まとめ]

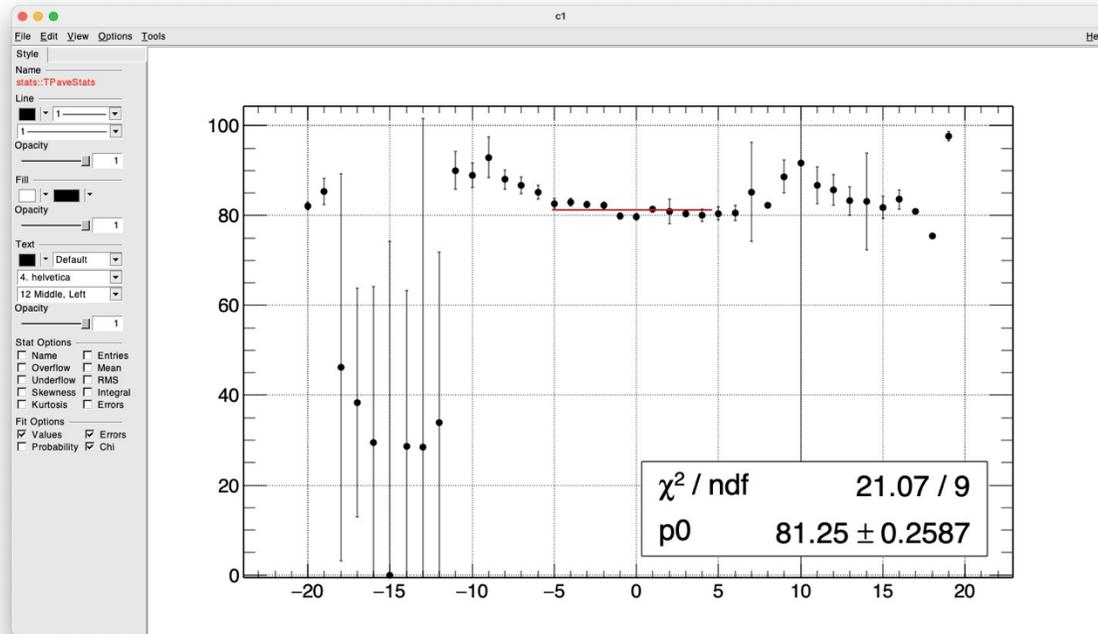
- MIPピークのばらつきが改善された(主にADCの80から85の範囲に分布) 依然としてばらつきが気になるが、 $\pm 2\%$ の範囲で安定した。
- ラン46000以降、平均ピークがやや低下したように見えるが、MIPピーク位置に大きな変化は見られなかった。
- 放射線損傷がINTTによる粒子測定に対して大きな影響を及ぼしていないことを示唆している。

## [今後の課題]

- データQAの最適化や他のカット、フィットの改善(フィットの安定性の評価)
- $\theta$ の大きい範囲でMIPピーク的位置を確認するとノイズの影響を受けにくいので試す。
- トラッキングのカットを加えなかった時のADC分布を確認してノイズ量を確認する。

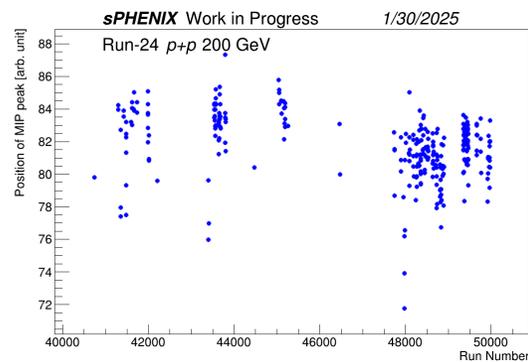
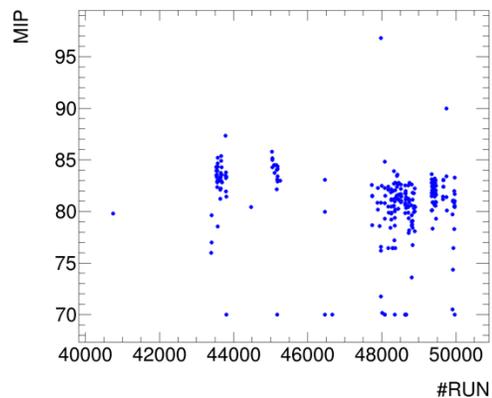
BACK UP

# z\_vertexの範囲選択

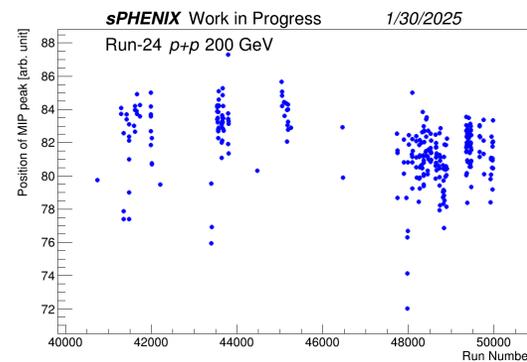


- $30 \leq \text{adc\_in} \ \&\& \ \text{adc\_in} \neq 210$
- `is_associated_in`
- $|\theta| > 80$
  
- Z\_vertexの取り扱いが微妙なので、-10から10までの範囲の結果だけを用いた( $-10 < z_{\text{vtx}} < 10$ )

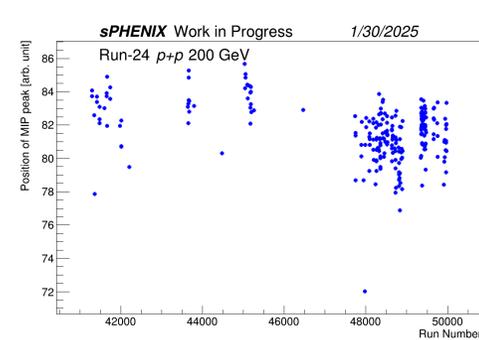
# BACK UP



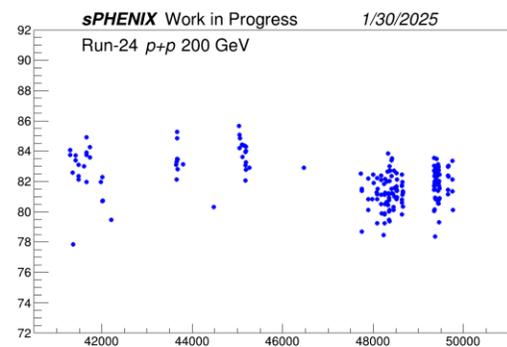
Fit improve



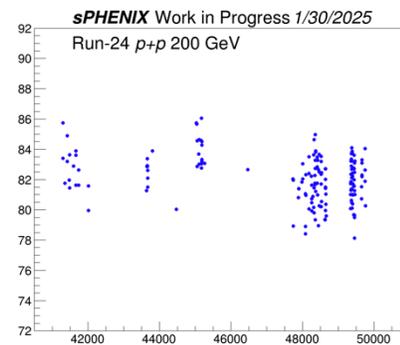
z\_vtx cut



Good run



No streaming  
z\_vtx\_finder



ADC\_ratio

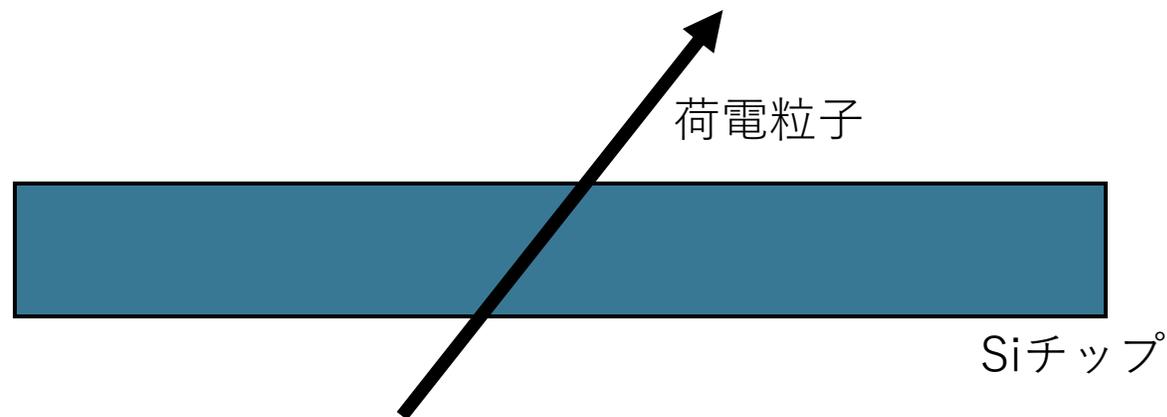
# 1/15 発表議事録

- MIP位置の角度依存性は、簡単な手計算で求められるので2つのフィットモデルの結果に加えて理論計算による予想もプロットに加える。
- フィットの範囲を変化させて、フィットの安定性を確認する。
- Run24を通してゲインの劣化は見られなかったという結果が出ているので、INTT英語ミーティングで発表する。

# BACK UP (MIP基礎知識)

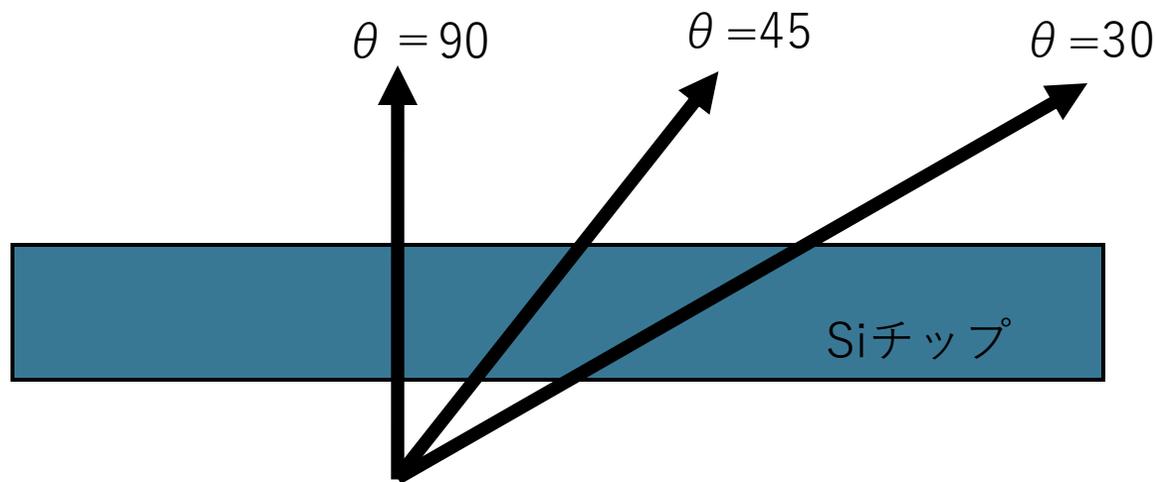
# MIPとは

- Minimum Ionizing Particle (Ionizationではない)
- 荷電粒子が通過した時のエネルギー損失量が最小になる粒子
- $\frac{\Delta E}{\Delta x}$  (エネルギー損失)
- $dE$ :失ったエネルギー、 $dx$ :単位長さ (=飛行距離)

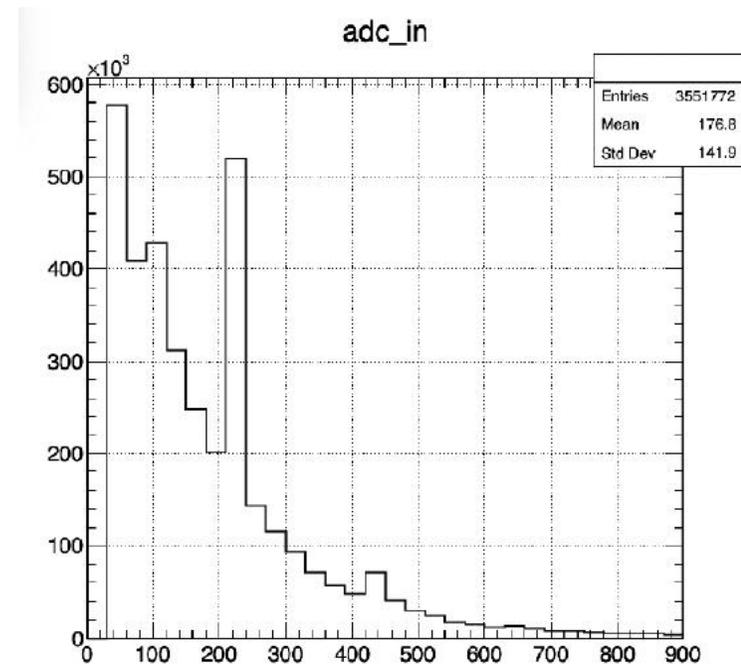


高エネルギーを持つ荷電粒子が物質中を通過するときに失うエネルギーは小さく、ほぼ一定の値

# トラックを引いた後の $\theta$ でカットをかける



エネルギー損失量( $\frac{\Delta E}{\Delta x}$ )は単位長さ辺りなので、飛行距離が変わればエネルギー損失量も変わる



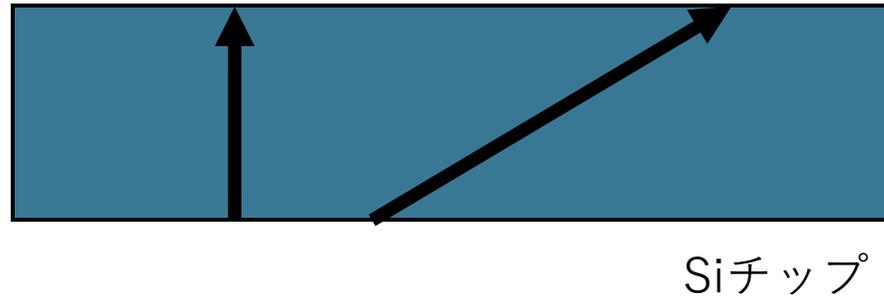
$\theta$ でカットをかけていないADC分布

**MIP** ピークがよく見えない

いろいろな角度の粒子によるエネルギー損失が混ざって、分布を汚くしている

→単位長さ辺りに揃えなくてはいけないので、チップへの入射角度でカットをかける

# トラックを引いた後の $\theta$ でカットをかける理由

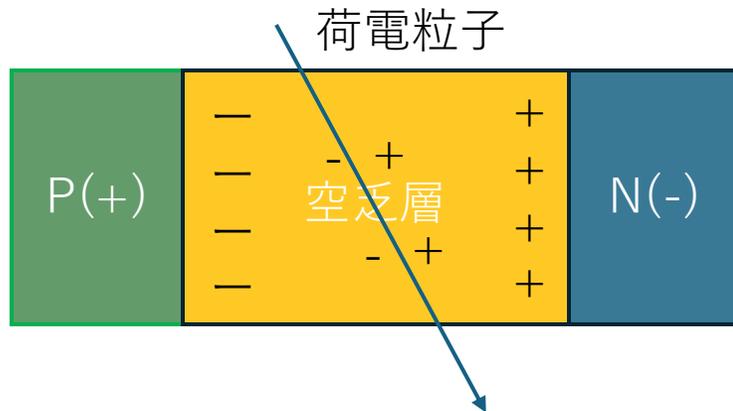


エネルギー損失量( $\frac{\Delta E}{\Delta x}$ )は単位長さ辺りなので、飛行距離が変わればエネルギー損失量も変わる

単位長さを揃えなくてはいけないので、チップへの入射角度でカットをかける

# 半導体検出器の原理

- P型の半導体とN型の半導体が接合(ダイオード)



ダイオードに逆電圧を流すと、マイナス電極側に+が集まり、プラス電極側に-が集まるので空乏層(キャリアの存在しない部分)が出来上がる  
荷電粒子が入射すると、電離作用によって電子正対が生成される。

# ADC

- アナログ信号をデジタル化した時の1信号
- ADC分布=イベントーADCのグラフは、ADCの大きさごとに積算されていくヒストグラムとしてみれる
- DAC値 = エネルギー損失(eV)の値を電圧値(mV)に変換したもの
- INTTではそのDAC値を閾値で8分割している

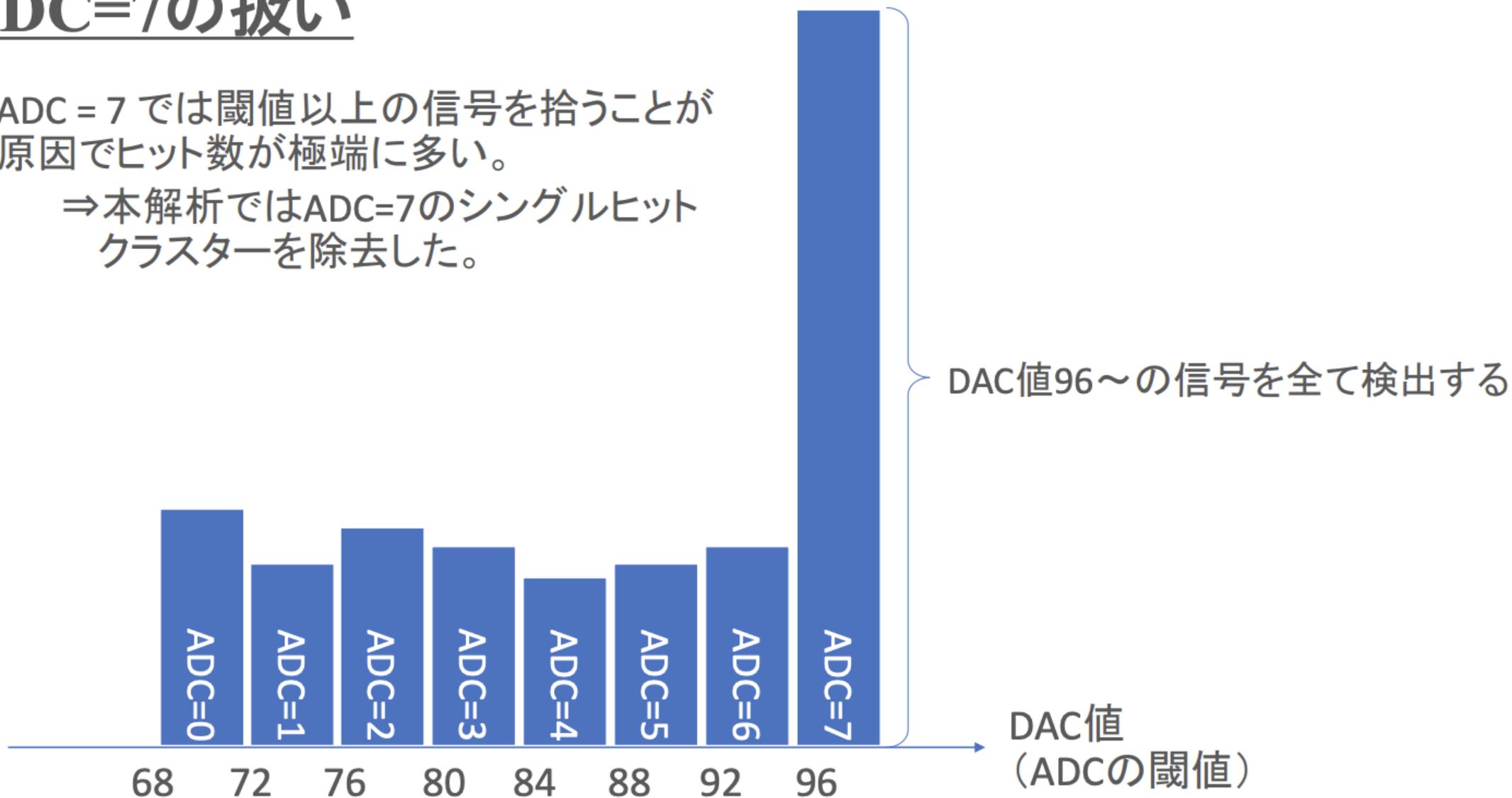
ADC 0 ~ 7

DAC 0 ~ 255

# ADC=7の扱い

ADC = 7 では閾値以上の信号を拾うことが原因でヒット数が極端に多い。

⇒本解析ではADC=7のシングルヒットクラスターを除去した。



# BACK UP (研究背景)

# 畳み込みする物理的な理由(参考:チェンウェイが送った論文)

## 1. エネルギー損失の分布の特徴

荷電粒子が物質を通過する際のエネルギー損失分布は統計的性質を持つ。実験結果では理論値よりも広がりが大きくなる。

## 2. 電子の束縛エネルギーの影響

Landau分布は自由電子を仮定しているが、固体中の電子は束縛されている。この影響がエネルギー損失分布を変化させる。

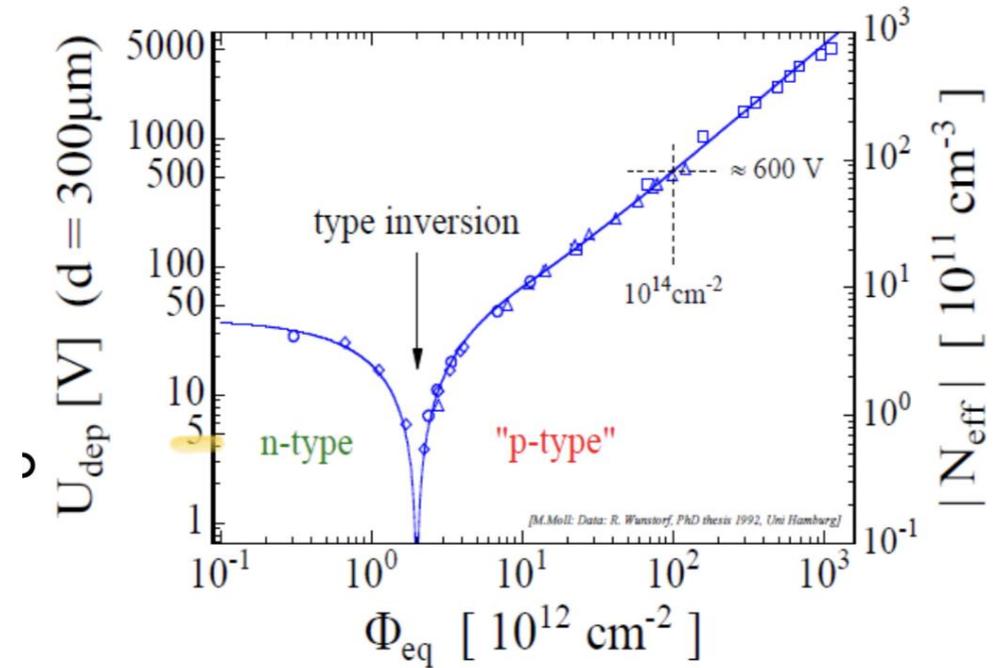
## 3. 検出器のノイズ(熱雑音)をgaussで近似

※電子の束縛エネルギーとは→原子核や結晶格子内で特定の状態に束縛されている電子が、自由になるために必要なエネルギー。電子が物質内の異なるエネルギー準位間を移動する際のエネルギーギャップや、電子が完全に物質を脱出するのに必要なエネルギーとして現れる。

※熱雑音→内部で流れる電流や電圧が熱運動している時の雑音

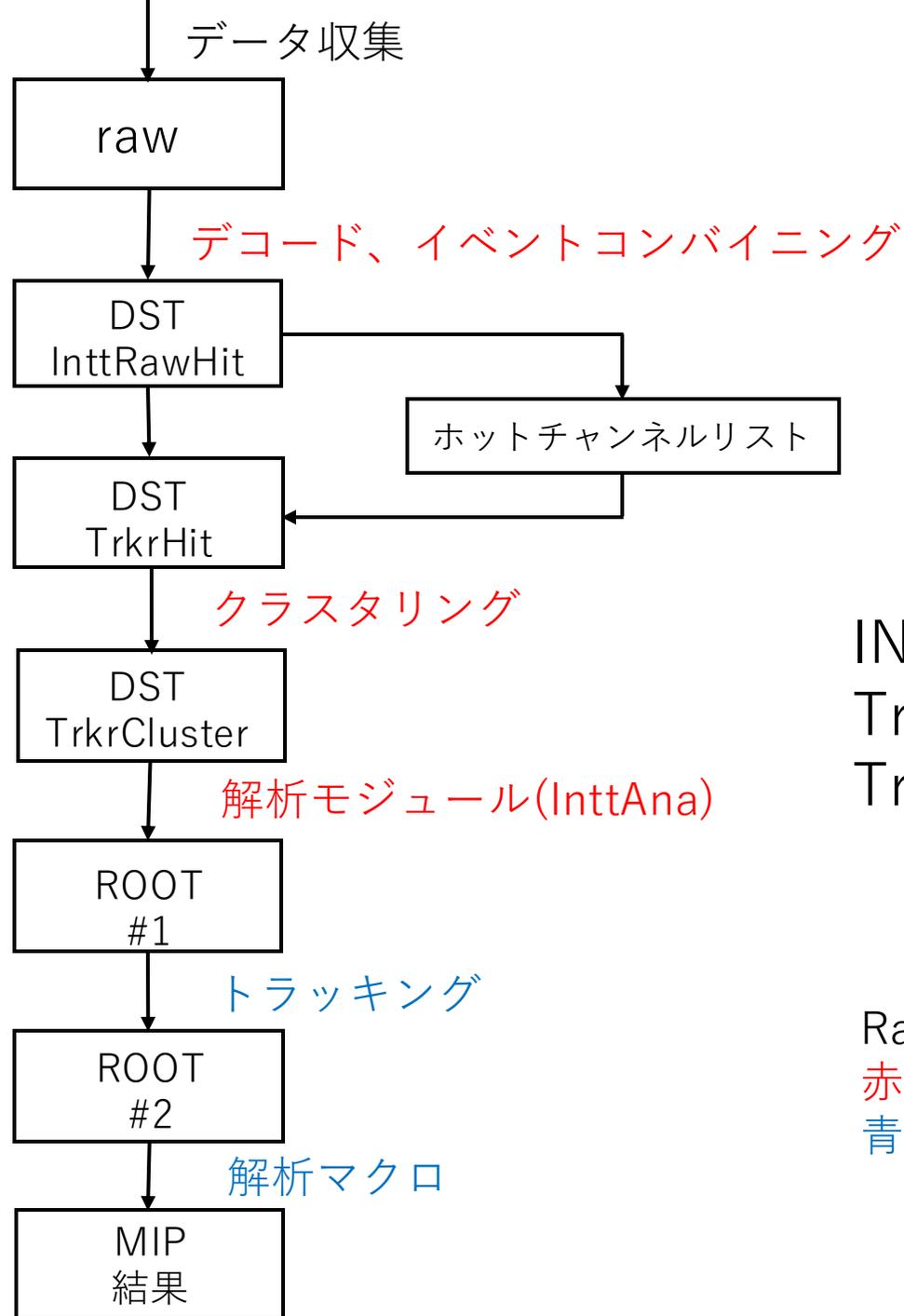
# 放射線損傷:バルク損傷

- シリコンセンサーに放射線が入射すると、電子を励起、反射したり、シリコン原子と衝突を繰り返してエネルギーを失っていく。
- バルク損傷を起こし、n型のバルクはp型に型変換する。
- 型変換すると空乏層の広がり逆になるので、読み出しには全空乏化が必要。
- 右の図から、全空乏化にはより大きい全空乏電圧が必要になる。



BACK UP (clus\_tree)

# MIPピーク解析の流れ



INTTRawHit:INTTT用のヒットフォーマット  
TrkrHit:トラッキング検出器で共通のフォーマット  
TrkrCluster:クラスタリングしたヒット

Rawデータを処理して段々と扱いやすいデータの形に変えていく  
赤: Fun 4 Allによる処理  
青: rootによる処理

# ROOT ファイル#2: clus\_treeの中身

| ブランチ名                           | :型              | :説明  |
|---------------------------------|-----------------|--|
| *Tree :clus_tree                | :clus_tree      |  |
| *Br 0:evt_clus                  | :evt_clus/I     | :クラスター数  |
| *Br 1:x_in                      | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターのx座標 (cm)                           |
| *Br 2:y_in                      | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターのy座標 (cm)                           |
| *Br 3:z_in                      | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターのz座標 (cm)                           |
| *Br 4:r_in                      | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターのr座標 (cm)                           |
| *Br 5:size_in                   | :vector<int>    | :内側バレルの <i>i</i> 番目のクラスターの大きさ                                |
| *Br 6:phi_in                    | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターの $\phi$ (degree, $-\pi \sim \pi$ ) |
| *Br 7:theta_in                  | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターの $\theta$ (degree)                 |
| *Br 8:adc_in                    | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターのADC                                |
| *Br 9:is_associated_in          | :vector<bool>   | :内側バレルの <i>i</i> 番目のクラスターがトラッキングに使用されたかどうか                   |
| *Br 10:track_incoming_theta_in  | :vector<double> | :内側バレルの <i>i</i> 番目のクラスターを使ったトラックの $\theta$                  |
| *Br 11:x_out                    | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターのx座標 (cm)                           |
| *Br 12:y_out                    | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターのy座標 (cm)                           |
| *Br 13:z_out                    | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターのz座標 (cm)                           |
| *Br 14:r_out                    | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターのr座標 (cm)                           |
| *Br 15:size_out                 | :vector<int>    | :外側バレルの <i>i</i> 番目のクラスターの大きさ                                |
| *Br 16:phi_out                  | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターの $\phi$ (degree)                   |
| *Br 17:theta_out                | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターの $\theta$ (degree)                 |
| *Br 18:adc_out                  | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターのADC                                |
| *Br 19:is_associated_out        | :vector<bool>   | :外側バレルの <i>i</i> 番目のクラスターがトラッキングに使用されたかどうか                   |
| *Br 20:track_incoming_theta_out | :vector<double> | :外側バレルの <i>i</i> 番目のクラスターを使ったトラックの $\theta$ (degree)         |
| *Br 21:z_vertex                 | :z_vertex/D     | :vertexのz座標 (cm)   |

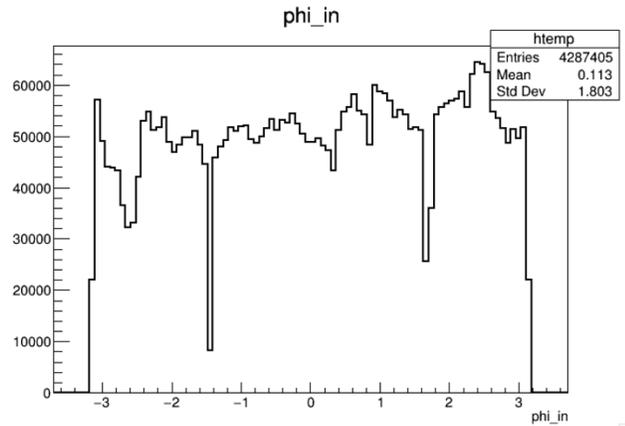
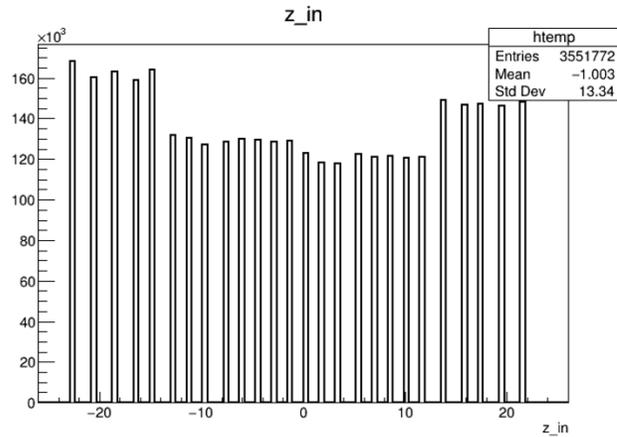
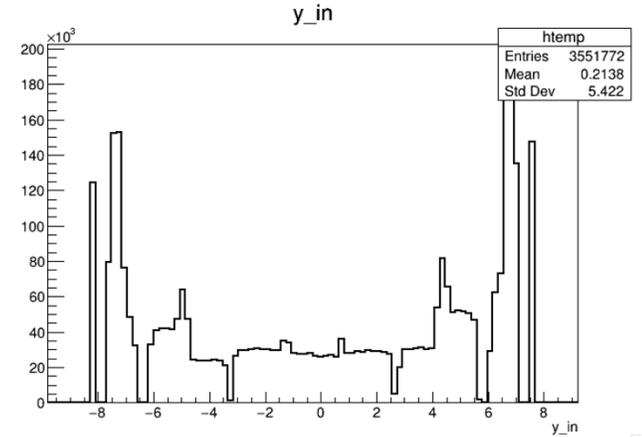
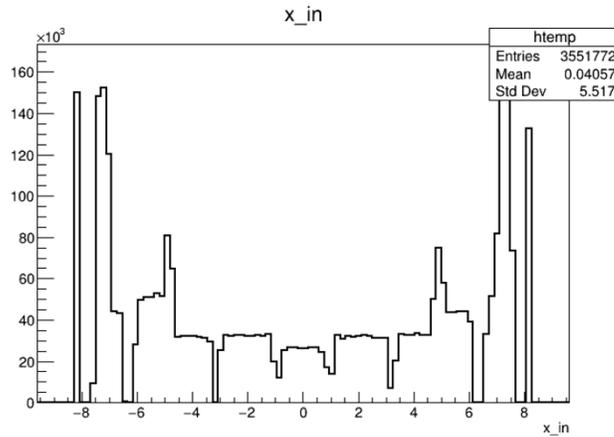
clus\_treeの中身。枝の種類と説明

# ROOT ファイル#2: clus\_treeの中身

clus\_tree->Print()

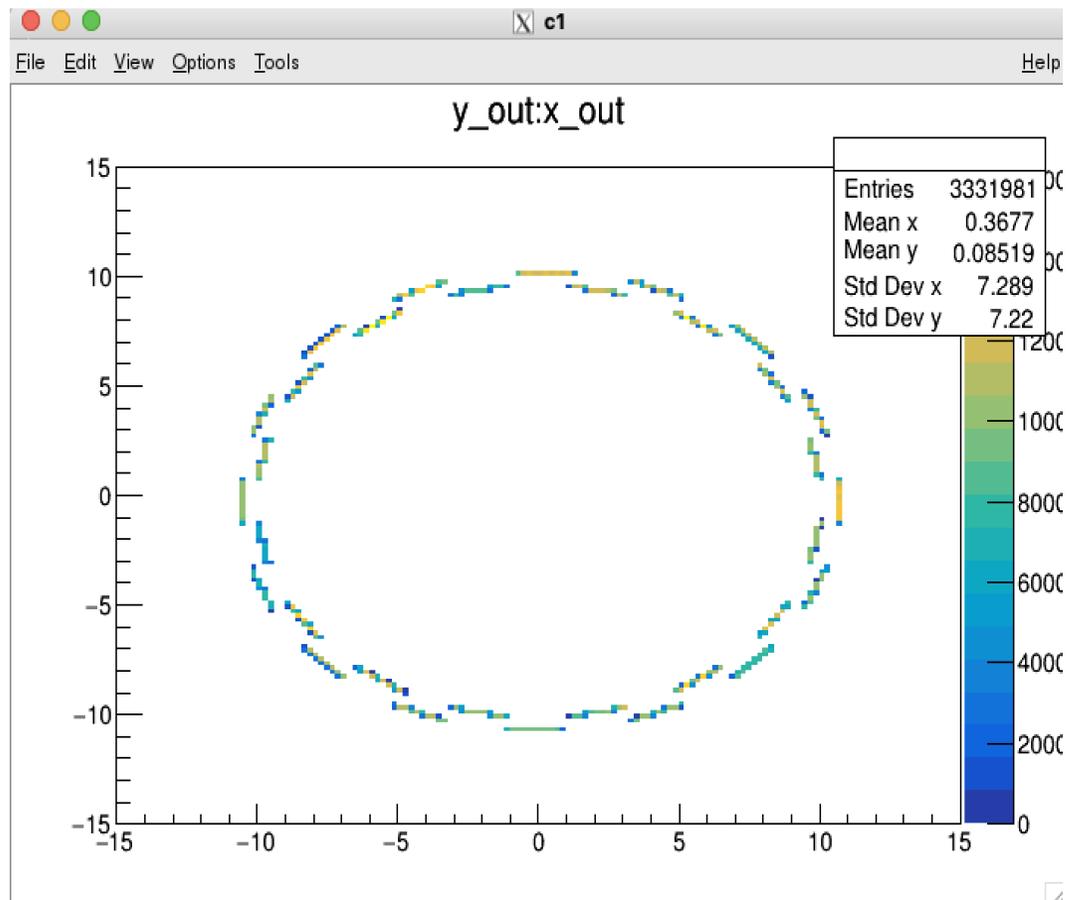
```
*****
*Tree :clus_tree : clus_tree
*Entries : 366403 : Total = 590093093 bytes File Size = 250584882 *
* : : Tree compression factor = 2.35 *
*****
*Br 0 :evt_clus : evt_clus/I *
*Entries : 366403 : Total Size= 1477029 bytes File Size = 524671 *
*Baskets : 111 : Basket Size= 32000 bytes Compression= 2.81 *
*.....*
*Br 1 :x_in : vector<double> *
*Entries : 366403 : Total Size= 33664555 bytes File Size = 18727278 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 1.80 *
*.....*
*Br 2 :y_in : vector<double> *
*Entries : 366403 : Total Size= 33664555 bytes File Size = 18794908 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 1.79 *
*.....*
*Br 3 :z_in : vector<double> *
*Entries : 366403 : Total Size= 33664555 bytes File Size = 7532590 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 4.47 *
*.....*
*Br 4 :r_in : vector<double> *
*Entries : 366403 : Total Size= 33664555 bytes File Size = 26632869 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 1.26 *
*.....*
*Br 5 :size_in : vector<int> *
*Entries : 366403 : Total Size= 19413231 bytes File Size = 4790315 *
*Baskets : 712 : Basket Size= 32000 bytes Compression= 4.05 *
*.....*
*Br 6 :phi_in : vector<double> *
*Entries : 366403 : Total Size= 39570917 bytes File Size = 29190603 *
*Baskets : 1337 : Basket Size= 32000 bytes Compression= 1.35 *
*.....*
*Br 7 :theta_in : vector<double> *
*Entries : 366403 : Total Size= 33669203 bytes File Size = 2700409 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 12.46 *
*.....*
*Br 8 :adc_in : vector<double> *
*Entries : 366403 : Total Size= 33666879 bytes File Size = 7869557 *
*Baskets : 1158 : Basket Size= 32000 bytes Compression= 4.28 *
*.....*
*Br 9 :is_associated_in : vector<bool> *
*Entries : 366403 : Total Size= 8725151 bytes File Size = 2578446 *
*Baskets : 374 : Basket Size= 32000 bytes Compression= 3.38 *
*.....*
*Br 10 :track_incoming_theta_in : vector<double> *
*Entries : 366403 : Total Size= 33686756 bytes File Size = 8746406 *
*Baskets : 1159 : Basket Size= 32000 bytes Compression= 3.85 *
*.....*
```

clus\_tree->Draw("x\_in")



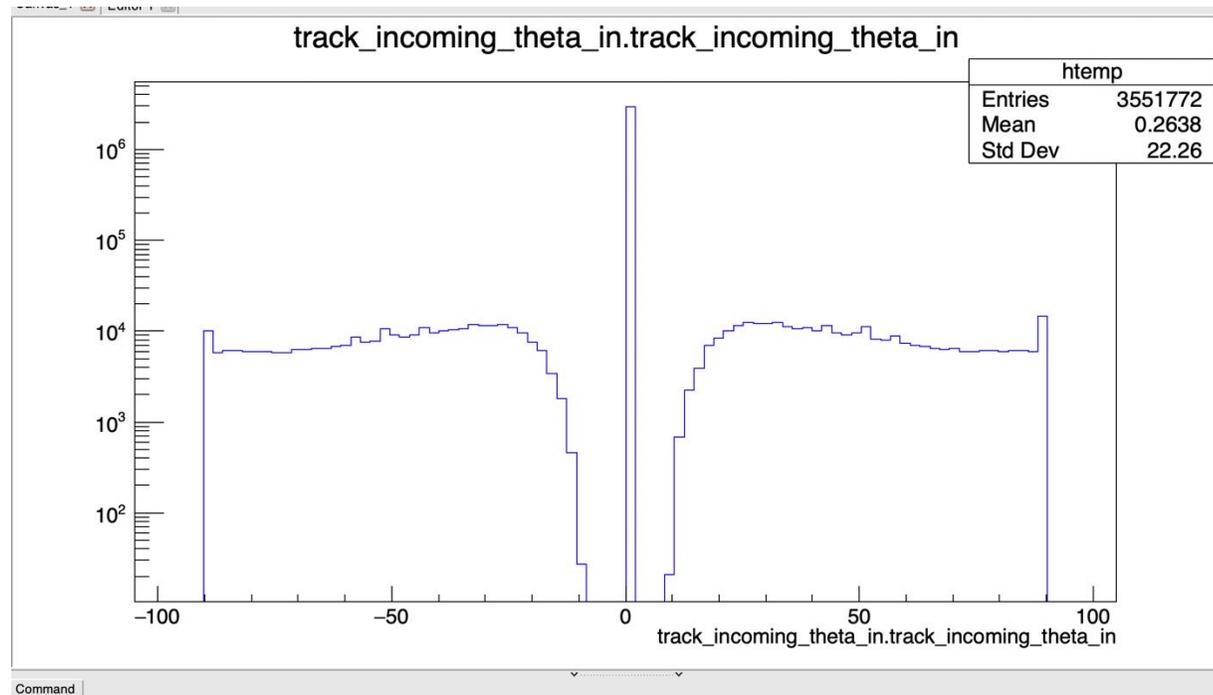
# ROOT ファイル#2: clus\_treeの中身

```
clus_tree->Draw("y_out:x_out>>(150, -15, 15, 150, -15, 15)", "", "colz")
```



クラスター位置を書いてみた

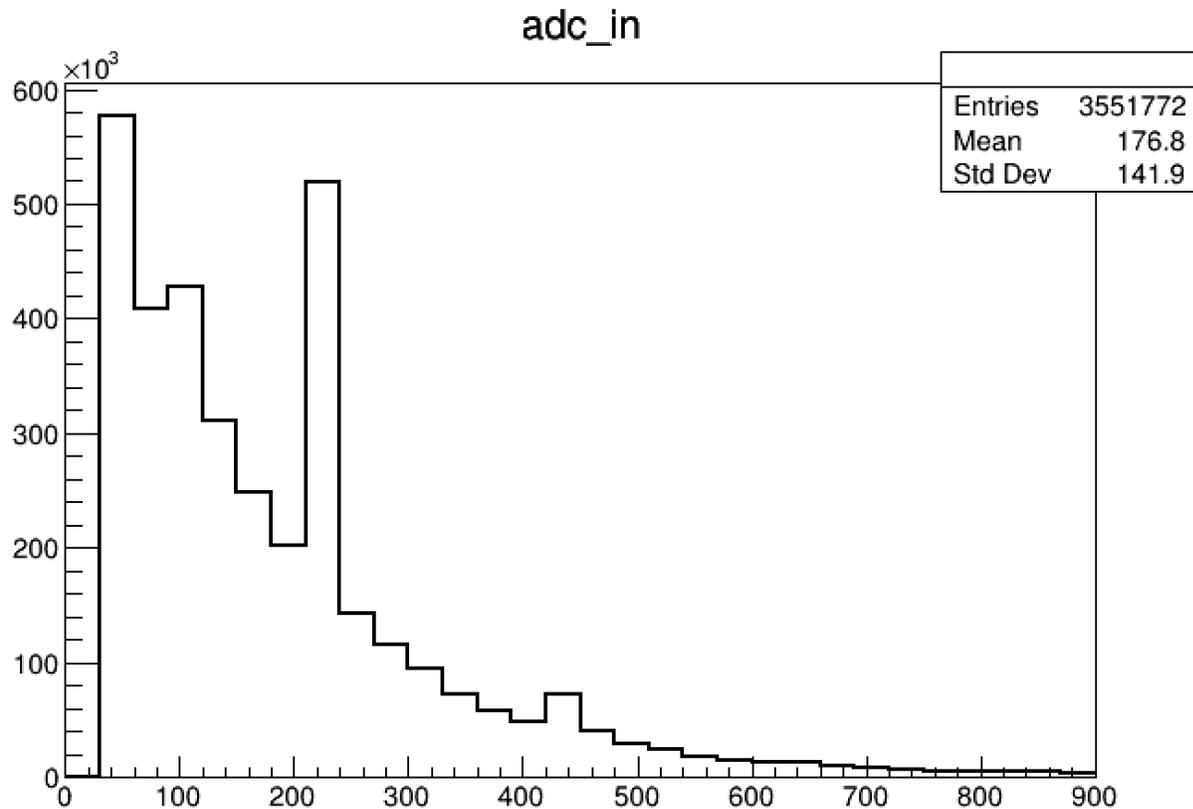
```
clus_tree->Draw("track_incoming_theta_in")
```



シータの分布

# ADC分布を見ている

```
clus_tree->Draw( "adc_in>>(30, 0, 900)", "", "" )
```



ピークよく見えない

いろいろな角度の粒子によるエネルギー損失が混ざって、分布を汚くしている

→角度 $\theta$ でカットをかける

# クラスツリーについて

- ツリーの中身はイベント毎に置かれている。
- イベントごとのクラスター数はイベントごとに違うので、vectorを用いている。
- インナーとアウターの値に対応関係はない。

```
(long long) 25
root [7] clus_tree->Scan( "evt_clus: x_in: x_out" )
*****
+-----+
| Row  * Instance * evt_clus *  x_in *  x_out *  |
+-----+
| 0 * 0 * + 0 * 7.4118075 * * |
| 1 * 0 * 1 * 6.1039886 * 7.7431697 * |
| 1 * 1 * 1 * -5.279232 * 5.4970717 * |
| 1 * 2 * 1 * 7.1218271 * 7.9710173 * |
| 1 * 3 * 1 * -1.839832 * 9.8751277 * |
| 1 * 4 * 1 * -1.614294 * 8.6378821 * |
| 1 * 5 * 1 * -1.029330 * -5.737440 * |
| 1 * 6 * 1 * 4.5460696 * -5.980092 * |
| 1 * 7 * 1 * 8.1572790 * 1.4814559 * |
| 1 * 8 * 1 * 7.0139341 * -2.606016 * |
| 1 * 9 * 1 * -0.034971 * -2.602193 * |
| 1 * 10 * 1 * 0.2224116 * -7.468428 * |
| 1 * 11 * 1 * -4.204267 * -0.717451 * |
| 1 * 12 * 1 * -4.315352 * -0.397666 * |
| 1 * 13 * 1 * -4.763060 * 0.4134963 * |
| 1 * 14 * 1 * 1.0764548 * -0.005838 * |
| 1 * 15 * 1 * 0.6864808 * 10.065373 * |
| 1 * 16 * 1 * -0.116865 * 0.4153557 * |
| 1 * 17 * 1 * -6.729800 * * |
| 1 * 18 * 1 * 7.729800 * * |
| 2 * 0 * 2 * 4.9215331 * 8.6254121 * |
| 2 * 1 * 2 * 5.9447646 * -9.764591 * |
| 2 * 2 * 2 * -7.344849 * -1.456239 * |
| 2 * 3 * 2 * -1.997325 * -2.492608 * |
| 2 * 4 * 2 * 7.4476065 * 1.3681385 * |
| 2 * 5 * 2 * 7.5122432 * 8.3466396 * |
| 2 * 6 * 2 * 7.2596621 * 10.012333 * |
| 2 * 7 * 2 * 7.5132379 * 6.4407577 * |
| 2 * 8 * 2 * 7.4275784 * -5.226254 * |
| 2 * 9 * 2 * 7.3725428 * -5.857148 * |
| 2 * 10 * 2 * 7.3399295 * -6.051269 * |
| 2 * 11 * 2 * 7.3215847 * -7.906499 * |
| 2 * 12 * 2 * 7.2961053 * -8.208580 * |
| 2 * 13 * 2 * 2.7421927 * 1.51589 * |
| 2 * 14 * 2 * -7.162256 * -8.419674 * |
| 2 * 15 * 2 * -5.791365 * -9.813656 * |
+-----+
type <CR> to continue or q to quit ==>
```

1イベント

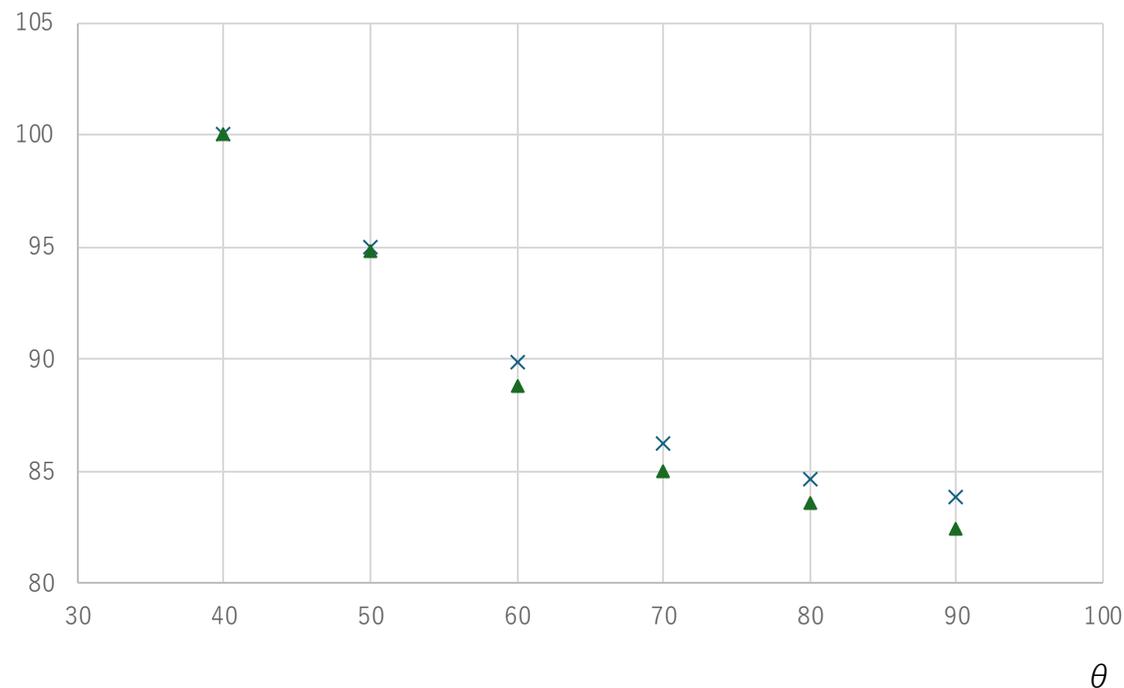
??  
ハップオシスル

BACK UP (fitting)

# MIPピーク位置の角度依存性

MIPピーク

①MPV- $\theta$  グラフ



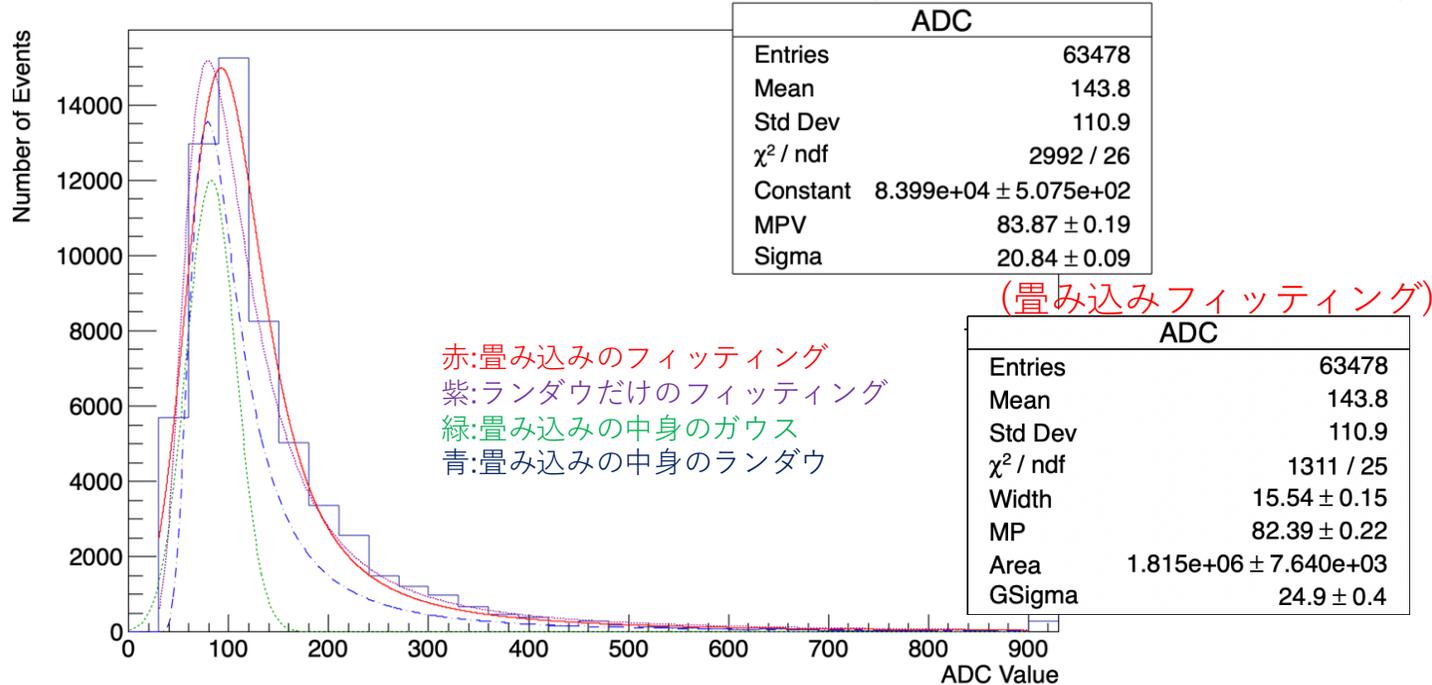
x MPV\_landau    ▲ MPV\_langau

|                    | landau | lan+gau |
|--------------------|--------|---------|
| $30 < \theta < 40$ | 100.03 | 100     |
| $40 < \theta < 50$ | 94.95  | 94.82   |
| $50 < \theta < 60$ | 89.83  | 88.78   |
| $60 < \theta < 70$ | 86.23  | 85.03   |
| $70 < \theta < 80$ | 84.61  | 83.54   |
| $80 < \theta < 90$ | 83.87  | 82.39   |

- ・ MIPは荷電粒子が通過した時のエネルギー損失量が最小になる粒子のこと。
- ・ 単位長さあたりのエネルギー損失 $\Delta E/\Delta x$ は、トラッキングに用いた $\theta$ が小さい程、飛行距離が長く、MIP値が大きい。
- ・ 結果として $\theta$ が小さい時MIP値が大きくなる様子が見られた。

# 2-1.フィッティングの改善

ADC Distribution (80 < theta < 90) (ランダウだけのフィッティング)



[ワークショップでの課題]

Roofitのコードを用いてフィッティングを行ったが、 $\chi^2/\text{ndf}$ の値をうまく算出できず、フィット精度の比較が難しかった。

→rootが提供しているチュートリアルコードを用いてフィットをやり直した。コードの中でランダウ、ガウスの関数を定義し、畳み込みのフィッティングをした。

[結果]

ランダウのみのフィットとの比較を行った。畳み込みのフィットはピークが右にシフトしている。