

Coincidence Tool Debug

• Coincidence をかける方法

1. RAW data から TrigT1TGC を用いて D3PD product
2. 既存の D3PD に対して Tool を使って Coincidence

1. RAW data から D3PD product

- 正しく coincidence がかかっていそう
- coincidence 結果は `TGC_coin_*` に対して適用。(`trig_L1_mu_*` には適用されない)
- `TGC_coin_*` には current bunch 以外の情報なども含まれていて逆に eta, phi 情報は含まれないため、単純に `TGC_coin_*` と `trig_L1_mu_*` との対応が取りづらい。そのため operation mode の study には現状使えていない
- exclusive OR (kimeuchi) の数字が出せない

2. Tool で Coincidence

- coincidence がおかしい
- 稲丸さんが過去に MC で作った event については TrigT1TGC の結果と consistent だったらしい
- いくつか定義してある変数が分からないので稲丸さんに質問を投げている
- 完成させれば `trig_L1_mu_*` に対して coincidence 結果が適用されるので、すぐに operation mode study ができる

- TrigT1TGC の方が正しいと思っている根拠

正しく CW を適用出来れば田代さんの Coincidence よりも improve するはず

TrigT1TGC の方は improve していて、Tool の方は逆に悪くなっている

<https://indico.cern.ch/getFile.py/access?contribId=4&resId=0&materialId=slides&confId=284601>

TGC_coin_pt と trig_L1_mu_thrNumber の一部

```
*****
*      Row      * Instance * TGC_coin_ * trig_L1_m *
*****
*          0 *         0 *         0 *          *
*          0 *         1 *         0 *          *
*          0 *         2 *         0 *          *
*          0 *         3 *         0 *          *
*          1 *         0 *         0 *          *
*          1 *         1 *         0 *          *
*          1 *         2 *         0 *          *
*          1 *         3 *         0 *          *
*          1 *         4 *         0 *          *
*          2 *         0 *         0 *          *
*          2 *         1 *         0 *          *
*          2 *         2 *         0 *          *
*          2 *         3 *         0 *          *
*          2 *         4 *         0 *          *
*          2 *         5 *         0 *          *
*          2 *         6 *         0 *          *
*          2 *         7 *         0 *          *
*          2 *         8 *         0 *          *
*          ~          ~          ~          ~
*          81 *         0 *         0 *          *
*          81 *         1 *         0 *          *
*          81 *         2 *         0 *          *
*          81 *         3 *         0 *          *
*          81 *         4 *         0 *          *
*          81 *         5 *         0 *          *
*          81 *         6 *         0 *          *
*          81 *         7 *         0 *          *
*          82 *         0 *         0 *          3 *
*          82 *         1 *         0 *          2 *
*          82 *         2 *         0 *          *
*          82 *         3 *         0 *          *
*          82 *         4 *         0 *          *
*          82 *         5 *         0 *          *
*          82 *         6 *         0 *          *
*          82 *         7 *         0 *          *
```

Tool

- 場所

<svn+ssh://tkunigo@svn.cern.ch/repos/atlas-tkunigo/tkunigo/TILE/EIFIcoin>

<https://svnweb.cern.ch/cern/wsvn/atlas-tkunigo/tkunigo/TILE/EIFIcoin/>?

- 比較対象

1. TrigT1TGC の InnerVETO をかけた sample の TGC_coin_pt
/gpfs/fs2001/tkunigo/TILE/EIFI/fromRAW_216416_setK/data_*.root

2. かけていないsample

(/gpfs/fs2001/tkunigo/TILE/EIFI/TrigT1TGC_noInnerVETO_216416_skim/data_*.root)



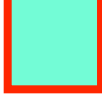
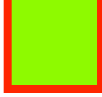
に Tool を適用した結果の trig_L1_mu_thrNumber

Tool : CW 読み込み

最初に InnerCoincidenceMap.setK._12.db という TrigT1TGC で使用したのと同じ dbFile から CW を読み込む (EIFI:read_CW)

dbFile は次のように 2行が 1 set で情報を表している

#	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1
	0	2	9	4	4	0	0	0							

-  Sector: 48 通り (0 ~ 47)
 -  SSC : 19 通り (0 ~ 18)
 -  not used
 -  Rol: SSC 中の 8 Rol に対して Inner Coincidence を要求するかどうかを 0 or 1 で表現 (要求するなら 1)
- 1 行目**

つまりこの行では Sector 0 の SSC 0 にある Rol 1 ~ 3, 5 ~ 8 (Rol 4 以外) について Inner Coincidence を要求することを示している

Tool : CW 読み込み

0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 1
0 29440 0 0 2 行目

4つの数字は SL Board に来ている EI/FI 0 ~ 3 の4つに対応する
その数字は2進数で考えて16bitで、例えば29440という数字は0111 0011
0000 0000ですが、これはMCの結果本物のmuonならこうhitがあるべきだ
ということだと思います

FI wire				FI strip				EI wire			EI strip				
0-7	8-15	16-23	24-31	0-7	8-15	16-23	24-31	NC	0-7	8-15	16-24	0-7	8-15	16-23	24-31
0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0

2行をまとめて読むと Sector:0, SSC:0 にある RoI には RoI 4 以外
に Inner Coincidence を要求し、EI/FI は EI/FI opt 1 に対して上の
bit のような bit を指定している(他は0)

これを全ての Sector, SSC に対して繰り返す
以下ではここで読み込んだ CW を元に Coincidence 判定を行う

Tool : DoCoincidence

- PT5 or PT6 の Endcap trigger を select して、その trigger に対して Coincidence 判定を行う
- 判定自体は EIFI:DoCoincidence という中に入っていて、accept すべきであれば true を返す function である

• DoCoincidence の中身

- 1.Inner Coincidence を要求する Rol かどうかの確認 (CW に基づいて trigger sector, SSC, Rol number(0 ~ 7) から判定)
- 2.EIFI:SetHitPattern
- 3.EIFI:TuneHitPattern
- 4.EIFI:Declustering
- 5.EIFI:SetTriggerBit
- 6.EIFI:MatchHitWithICW

Tool: SetHitPattern

- trigger sector を 2 個 1 set にして SlotID という変数にする（よく見る Octant の絵の Sector 2 個分に対応していると思います）
- TGC_prd_* の情報を用いて EI / FI の wire, strip について hit した channel に関する情報を見ている（HitPattern という構造体に対して hit した channel 情報を記録している）
- この際に TGC_prd_channel の情報から EIFI:GetHitChannel を用いて HitChannel という情報に変換している。これはHitChannelは、chamber 毎に 2 層の OR の取り方や、 ϕ 方向と channel の向きが異なっているため、それをTrigT1TGCでの結果を元に考慮したものらしい。計算式は理解出来ていないです。
- GetHitChannel の if 文がおかしいと思うので指摘したがこれで合っていると返答があった

Tool: Tune, Decluster, SetTriggerBit

- TuneHitPattern

64bit の HitChannel の情報を bit 積 bit 和を使って少し変更しているが何をしているのかが分かりません。

- Declustering

Declustering をしているのだと思うのですがこの計算があっているか分かりません

- SetTriggerBit

64bit の情報から 4bit の情報にする。4bit の情報というのは channel 数の違い以外は次のようなものになっているはず

FI wire			
0-7	8-15	16-23	24-31

Tool: MatchHitWithICW

- CW から読み込んだ EI/FI の BitPattern と trigger 情報の BitPattern から Coincidence が取れたかを判定

FI wire				FI strip				EI wire				EI strip			
0-7	8-15	16-23	24-31	0-7	8-15	16-23	24-31	NC	0-7	8-15	16-24	0-7	8-15	16-23	24-31
0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0

CW から読み込んだ BitPattern

FI wire				FI strip				EI wire				EI strip			
0-7	8-15	16-23	24-31	0-7	8-15	16-23	24-31	NC	0-7	8-15	16-24	0-7	8-15	16-23	24-31
0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0

trigger 情報から BitPattern

これを足し合わせると、MC と一致するところだけ 2 になるので、それで MC と一致するような Hit なのかどうかの判断をしている

- $(EIWire \ \&\& \ EIStrip) \ || \ (FIWire \ \&\& \ FIStrip)$ で最終結果である Inner Coincidence が取れたかどうか true or false で返す。この点については想定通りです。

疑っているところ

- HitChannel という変数が一番怪しいと思っています。
- SetHitPattern の if 文はやっぱり何かおかしいのではないかなと思っています、ただ HitChannel 周りの計算式が理解出来ていないので、ここを直すといいというのが分かりません。