sPHENIX

Hot/Cold/Dead Channel Classifier

Jaein Hwang

Byungsik Hong

KOREA UNIVERSITY 1905

Channel Stability with Gaussian Classifier (3sigma cut)

Developed by Yuka & Jaein

- **Run list : 20867, 20868, 20869, 20885 (Zero Field run)**

Run by Run dependence
Q. Does the good channel list change run by run? Then how much?

$$\text{agreement} = \frac{\text{number of channels good in BOTH runs}}{\text{number of channels good in EITHER run}} = \frac{\text{good channel list of run A } \cup \text{ good channel list of run B}}{\text{good channel list of run A } \cap \text{ good channel list of run B}}$$

Ref) Joseph's report
(Dec. 13, 2023)

Run number	20867	20868	20869	20885
# of good channels	347045	346949	346937	348611
Ratio (# of good channels/ # of total channels)	0.931074	0.930817	0.930785	0.935276
# of events	317047	288481	550123	357343

Channel Stability with Gaussian Classifier (3sigma cut)

- **Run list : 20867, 20868, 20869, 20885 (Zero Field run)**

Run by Run dependence
Q. Does the good channel list change run by run? Then how much?

$$\text{agreement} = \frac{\text{number of channels good in BOTH runs}}{\text{number of channels good in EITHER run}}$$

From Joseph's report
(Dec. 13, 2023)

Run Number

	Agreement	20867	20868	20869	20885
Run Number	20867		0.994459	0.995107	0.989447
	20868			0.995267	0.989104
	20869				0.989349

99% identical good channels were classified by Gaussian method in several ZF runs

Channel Stability with Gaussian Classifier (3sigma cut)

- Is there any dependence on runtime in same run?

- What about the stability between 0~10,000th events, 10,000th ~20,000th events?

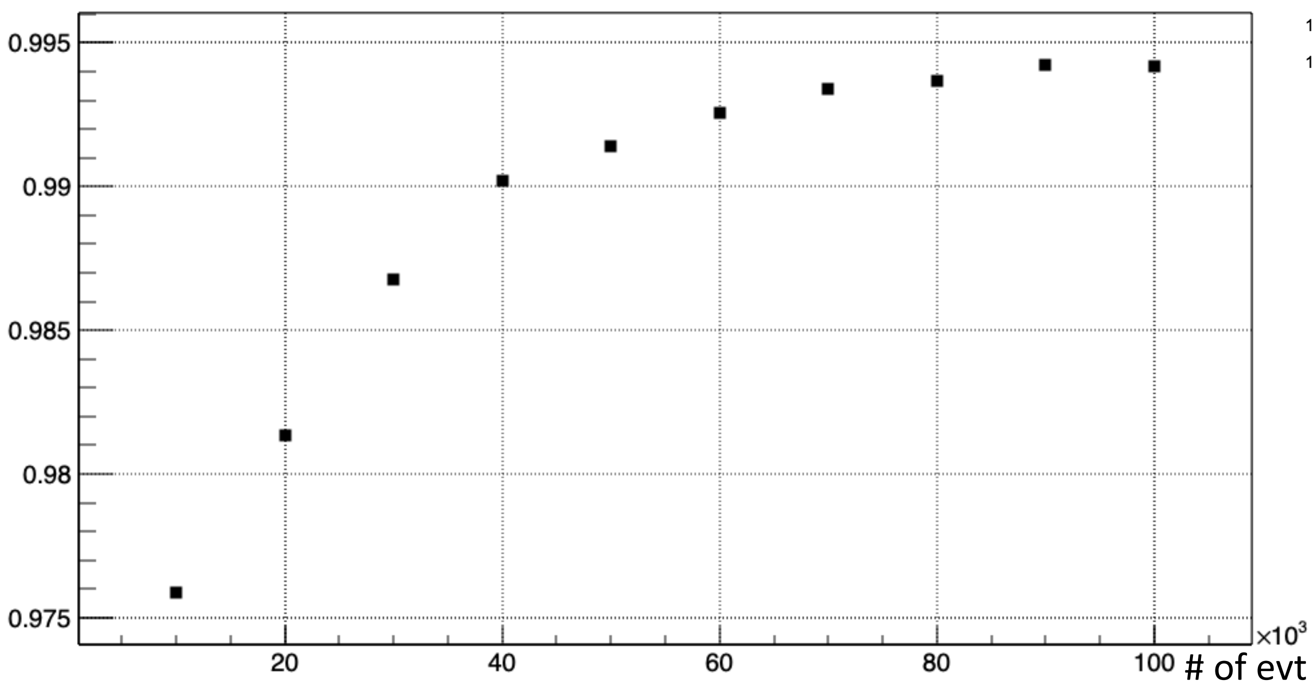
➔Starting Point : How many events would be needed to classify the channel?

- It allow us to reduce the time consuming to run through our workflow including hot channel algorithm. (question from Chris)

Effect of # of events (3sigma cut)

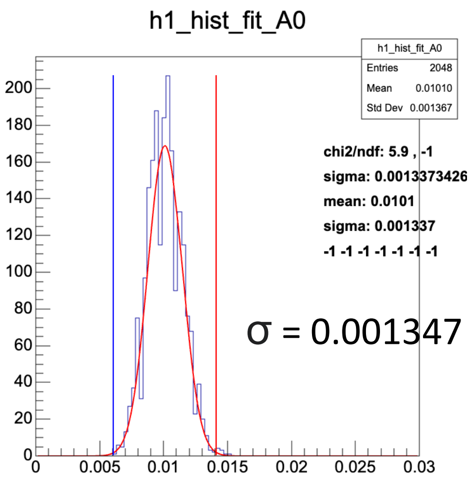
$$\text{Agreement} = \frac{\#\text{ of good channels in both 'All events used' and 'N events used'}}{\#\text{ of good channels in either 'All events used' or 'N events used'}}$$

Channel matching agreement / Run 20869



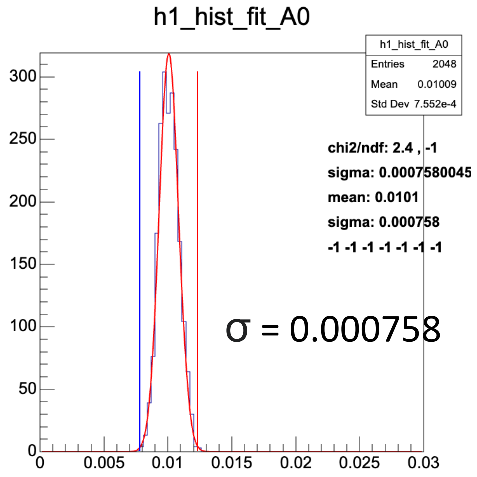
# of events used	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
Agreement	0.975889	0.981348	0.986747	0.990204	0.991387	0.99254	0.993404	0.993666	0.994215	0.994198

10k events used



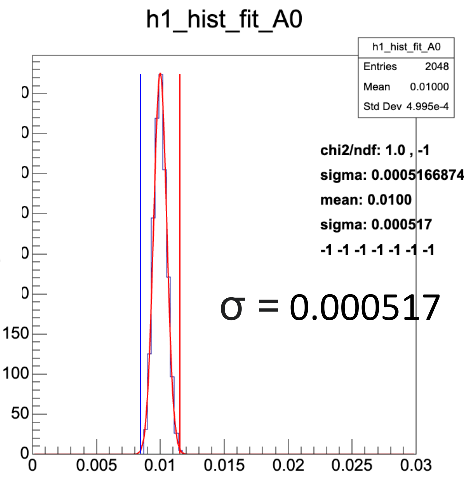
σ = 0.001347

50k events used



σ = 0.000758

All events used



σ = 0.000517

Gaussian Classifier depends on statistics due to Central Limit Theorem.

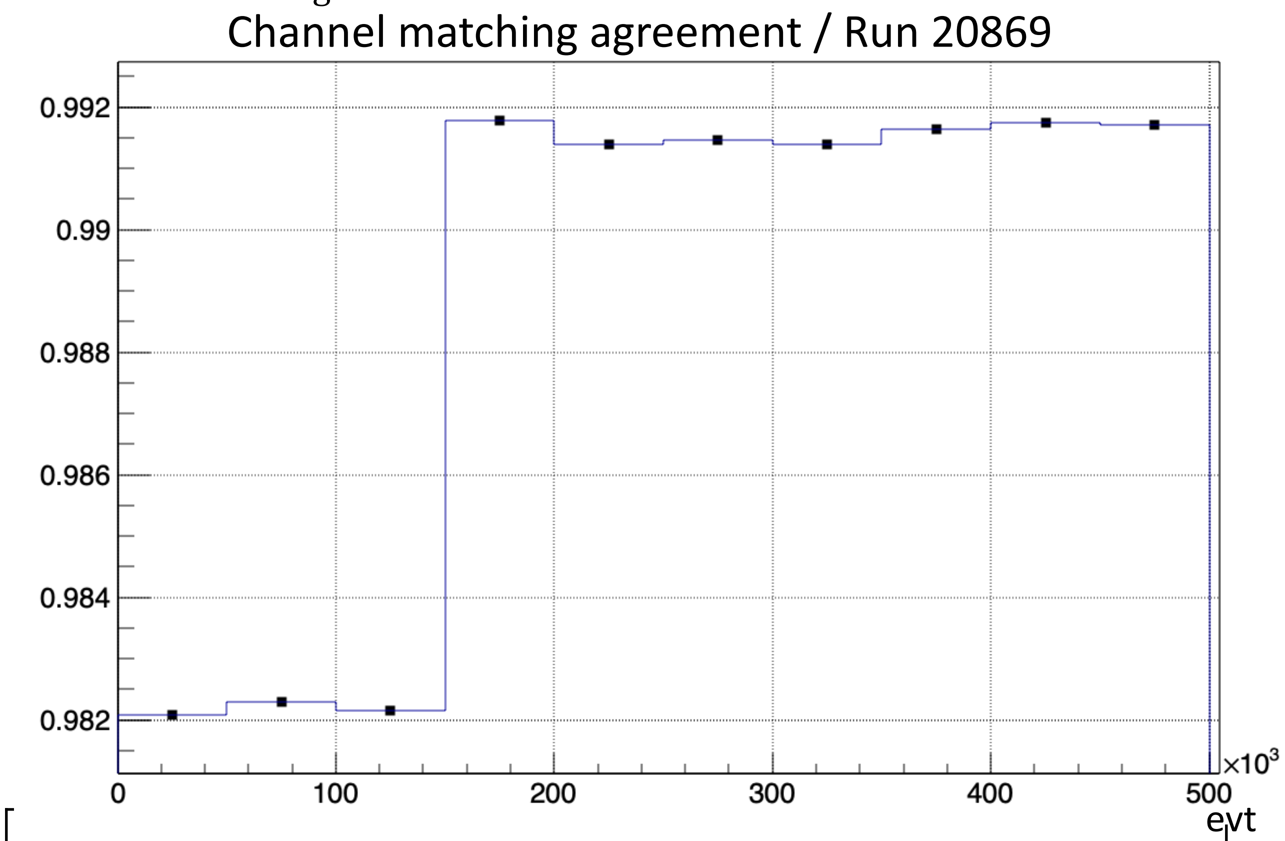
Enough statistics needed to get clear Gaussian

More events lead to more accurate classification

Let's use 50k events to do stability check.
It's already on 99% agreement.

Channel Stability in one run (3sigma cut)

$$\text{Agreement} = \frac{\#\text{ of good channels in both 'All events used' and '50k events used'}}{\#\text{ of good channels in either 'All events used' or '50k events used'}}$$



Selected events	0 ~ 50k	50k ~ 100k	100k - 150k	150k ~ 200k	200k ~ 250k	250k ~ 300k	300k ~ 350k	350k ~ 400k	400k ~ 450k	450k ~ 500k
Agreement	0.982091	0.98239	0.982157	0.991774	0.991387	0.991455	0.991387	0.991641	0.991745	0.991701

Procedure :
1. Select events with same interval (0~50k, 50k~100k, 100k~150k …up to 450k~500k)
2. Use the selected events as the input of the Gaussian Classifier.
3. Evaluate the **agreement** with original channel list Original list used all events.

Except for interval between 0~150k events, 99% identical good channels were classified

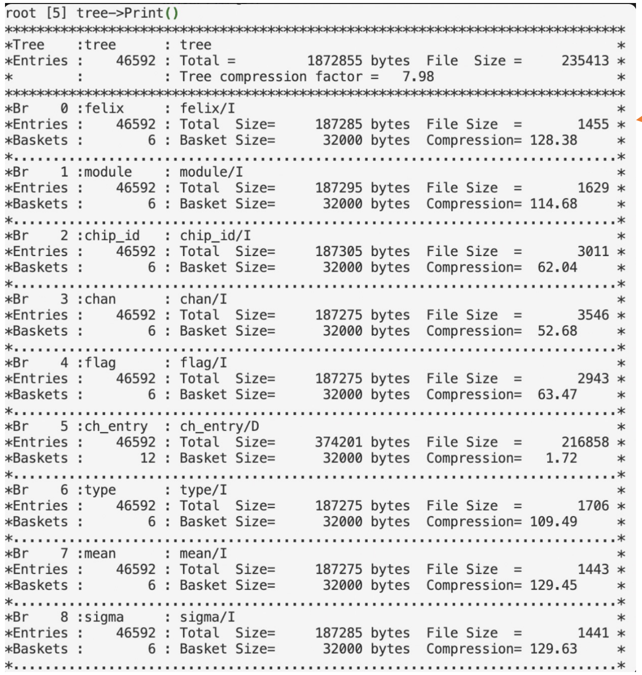
Cause of the drop at 0~150k events due to the problematic ladders Felix 2 module 11,12

From begin of the run ~ around 150k events Module 11, 12 were time-in, so some part of the channels on module 11,12 were classified as good channel.
(in original, these are classified as cold)
BCO peak of module 11,12 were moved depending on runtime(or event number)
More sophisticated study still is needed.

Status of the Hot channel classifier code

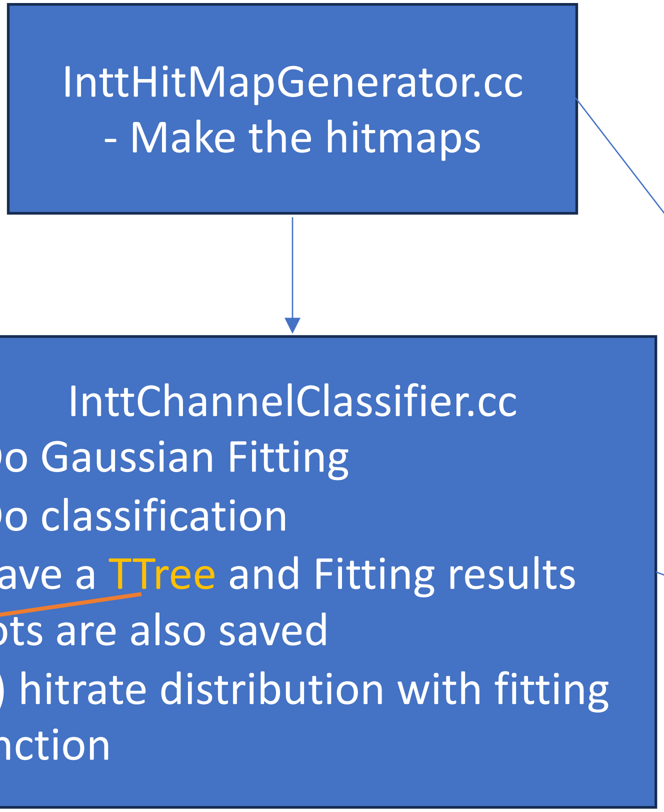
Old

One c++ code
- Make the hitmaps
- Do Gaussian Fitting
- Do classification based on Gaussian parameters



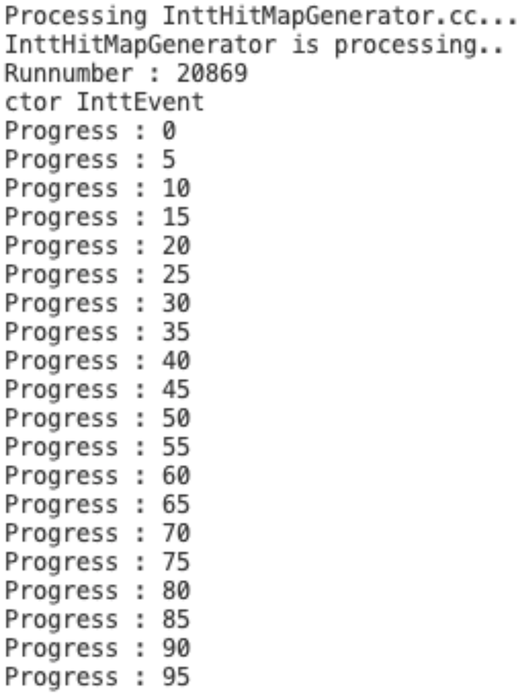
```
root [5] tree->Print()
******************************************************************************
*Tree    :tree      : tree                                                   *
*Entries :    46592 : Total =         1872855 bytes  File  Size =     235413 *
*        :          : Tree compression factor =   7.98                       *
******************************************************************************
*Br    0 :felix     : felix/I                                                *
*Entries :    46592 : Total  Size=     187285 bytes  File Size  =       1455 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression= 128.38     *
*............................................................................*
*Br    1 :module    : module/I                                               *
*Entries :    46592 : Total  Size=     187295 bytes  File Size  =       1629 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression= 114.68     *
*............................................................................*
*Br    2 :chip_id   : chip_id/I                                              *
*Entries :    46592 : Total  Size=     187305 bytes  File Size  =       3011 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression=  62.04     *
*............................................................................*
*Br    3 :chan      : chan/I                                                 *
*Entries :    46592 : Total  Size=     187275 bytes  File Size  =       3546 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression=  52.68     *
*............................................................................*
*Br    4 :flag      : flag/I                                                 *
*Entries :    46592 : Total  Size=     187275 bytes  File Size  =       2943 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression=  63.47     *
*............................................................................*
*Br    5 :ch_entry  : ch_entry/D                                             *
*Entries :    46592 : Total  Size=     374201 bytes  File Size  =     216858 *
*Baskets :       12 : Basket Size=      32000 bytes  Compression=   1.72     *
*............................................................................*
*Br    6 :type      : type/I                                                 *
*Entries :    46592 : Total  Size=     187275 bytes  File Size  =       1706 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression= 109.49     *
*............................................................................*
*Br    7 :mean      : mean/I                                                 *
*Entries :    46592 : Total  Size=     187275 bytes  File Size  =       1443 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression= 129.45     *
*............................................................................*
*Br    8 :sigma     : sigma/I                                                *
*Entries :    46592 : Total  Size=     187285 bytes  File Size  =       1441 *
*Baskets :        6 : Basket Size=      32000 bytes  Compression= 129.63     *
*............................................................................*
```

New

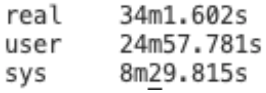


InttHitMapGenerator.cc
- Make the hitmaps

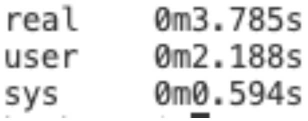
InttChannelClassifier.cc
- Do Gaussian Fitting
- Do classification
- Save a TTree and Fitting results
Plots are also saved
ex) hitrate distribution with fitting function



```
Processing InttHitMapGenerator.cc...
InttHitMapGenerator is processing..
Runnumber : 20869
ctor InttEvent
Progress : 0
Progress : 5
Progress : 10
Progress : 15
Progress : 20
Progress : 25
Progress : 30
Progress : 35
Progress : 40
Progress : 45
Progress : 50
Progress : 55
Progress : 60
Progress : 65
Progress : 70
Progress : 75
Progress : 80
Progress : 85
Progress : 90
Progress : 95
```



```
real    34m1.602s
user    24m57.781s
sys     8m29.815s
```



```
real    0m3.785s
user    0m2.188s
sys     0m0.594s
```

flag : 0 Good channel
flag : 1 Dead Channel (0 entry)
flag : 2 Cold Channel (less than mean-3sigma)
flag : 3 Hot Channel (greater than mean+3sigma
Form of TTree will be changed to fit to our framework

Short-term plans

-Writing RIKEN APR (Hot Channel Classifier)

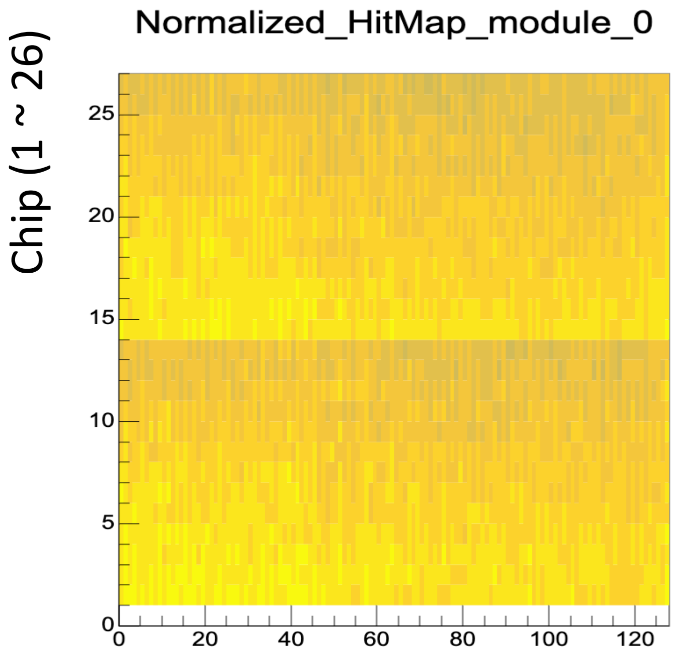
-Working with Joseph to compare channel lists with his algorithm

backup

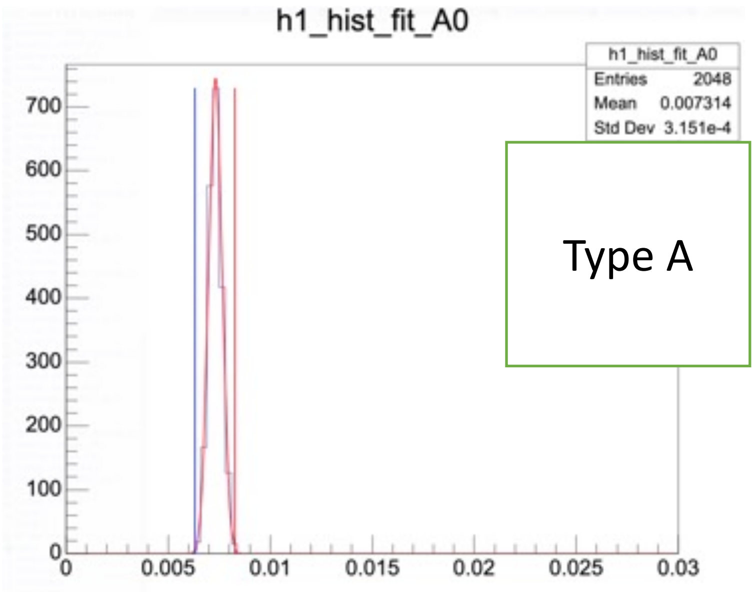
Hot Channel algorithm

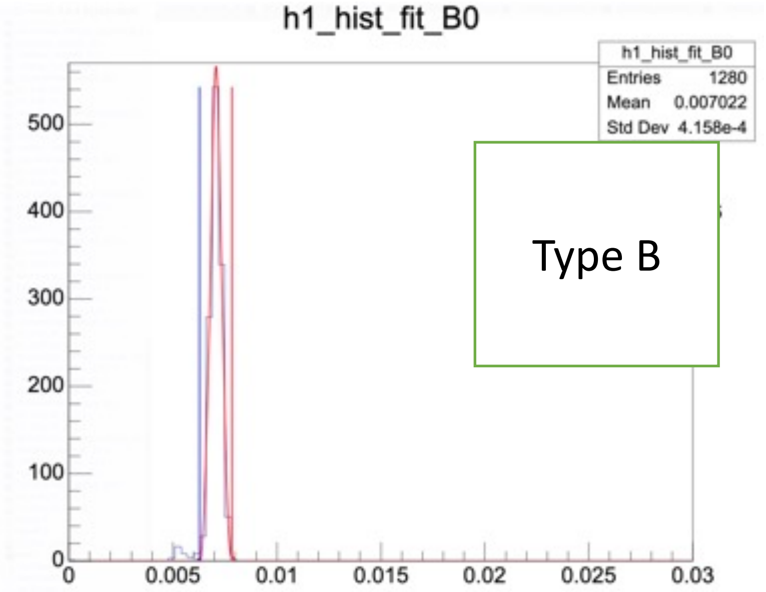
Reminder : Procedure of hot channel classifier

1. Make a normalized hit map distribution
normalized by :
- number of event
- Acceptance difference depending on the chip type (Type A and Type B) and layer (inner and outer)



2. Draw Hit rate distribution for every half ladders.
Each half ladder have two hit rate distributions, one is distribution of Type A, the other is for Type B.





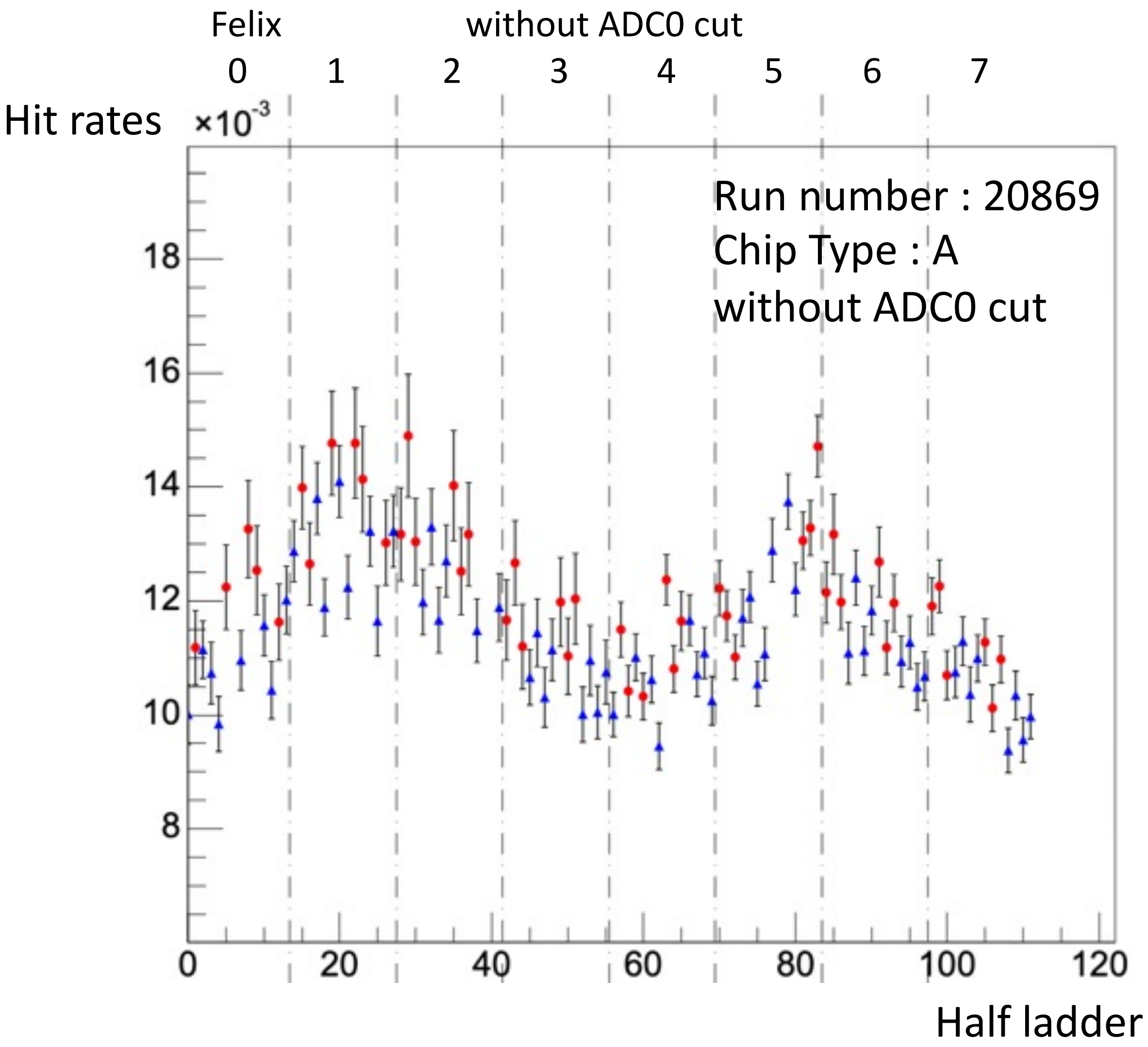
3. Definition of channel

Hot Channel : mean + 3sigma

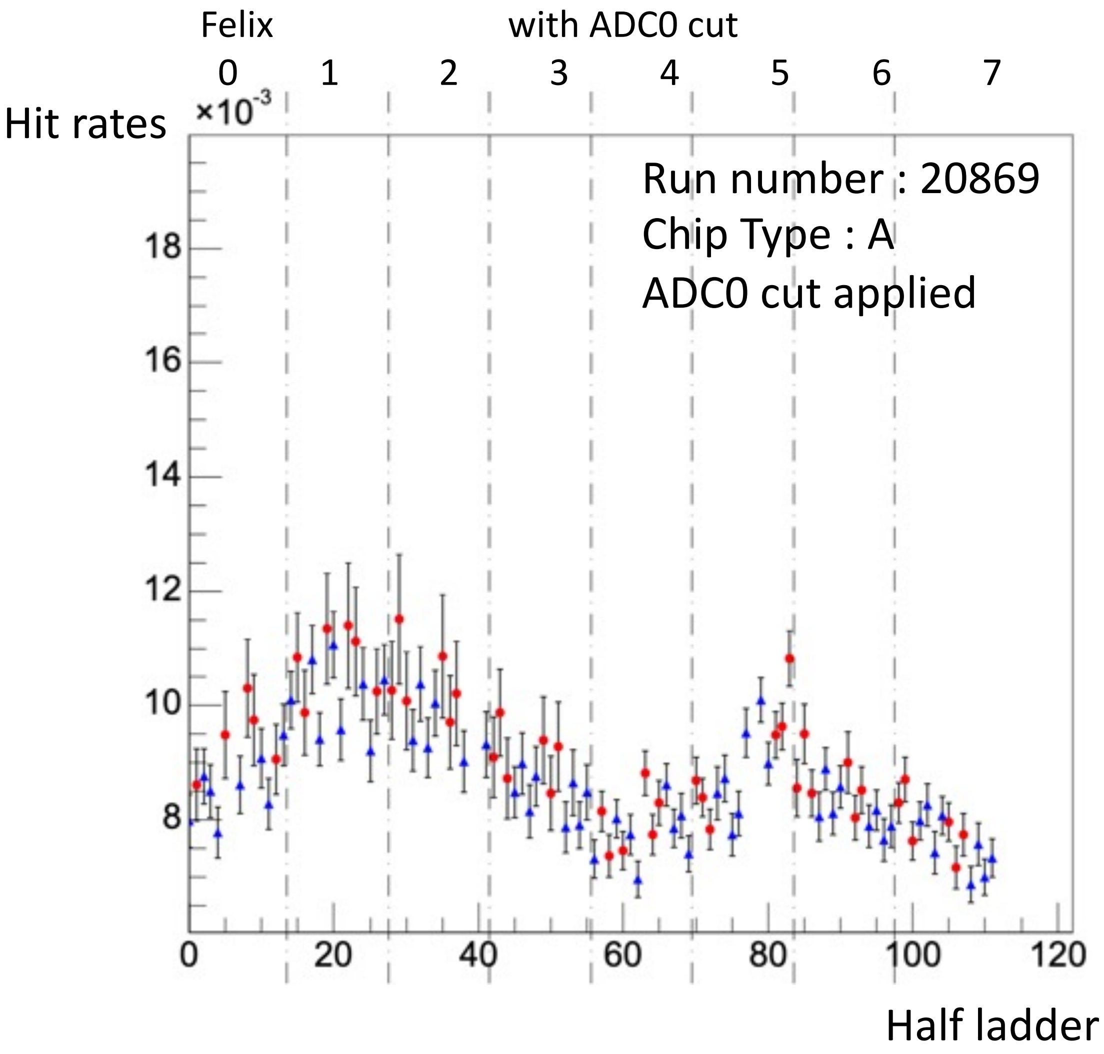
Cold Channel : mean – 3sigma

Dead Channel : 0 hit

Effect of ADC0 cut



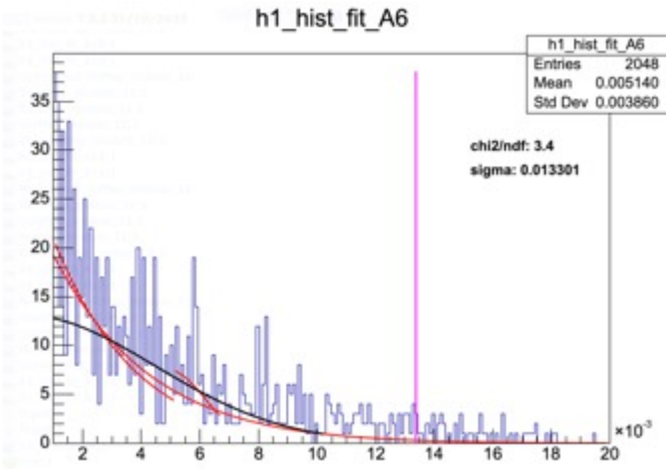
Run number : 20869
Chip Type : A
without ADC0 cut

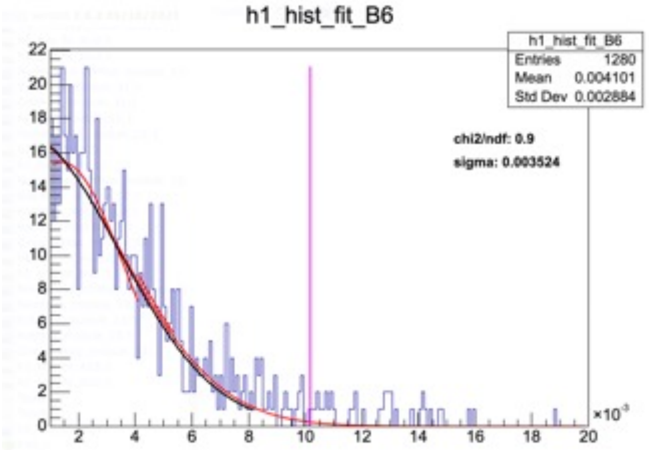


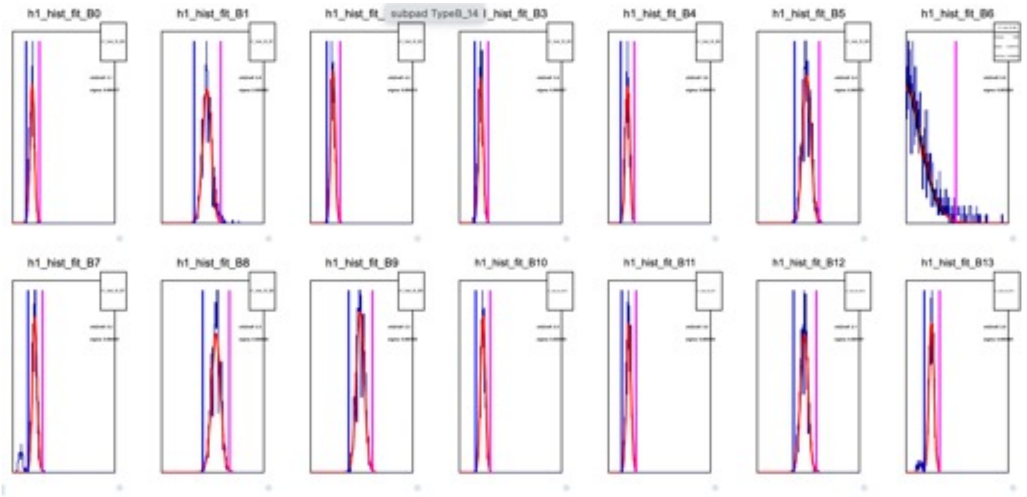
Run number : 20869
Chip Type : A
ADC0 cut applied

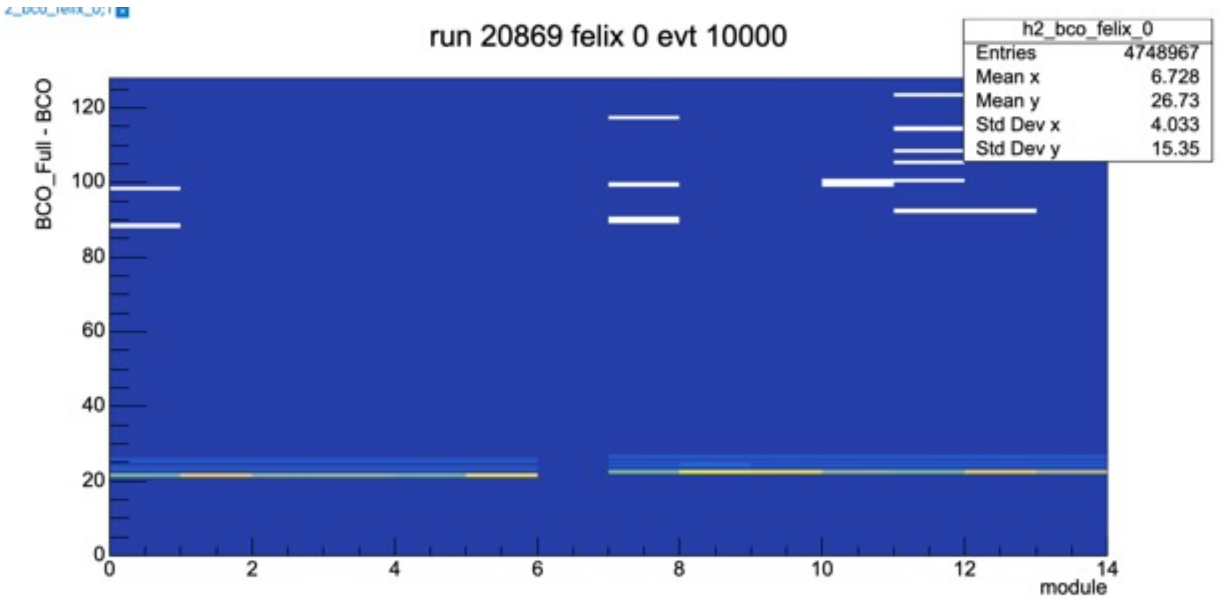
Problematic ladders(Felix 0 module 6)

Hit rate distribution before BCO cut – No clear time peak in **BCO distribution**



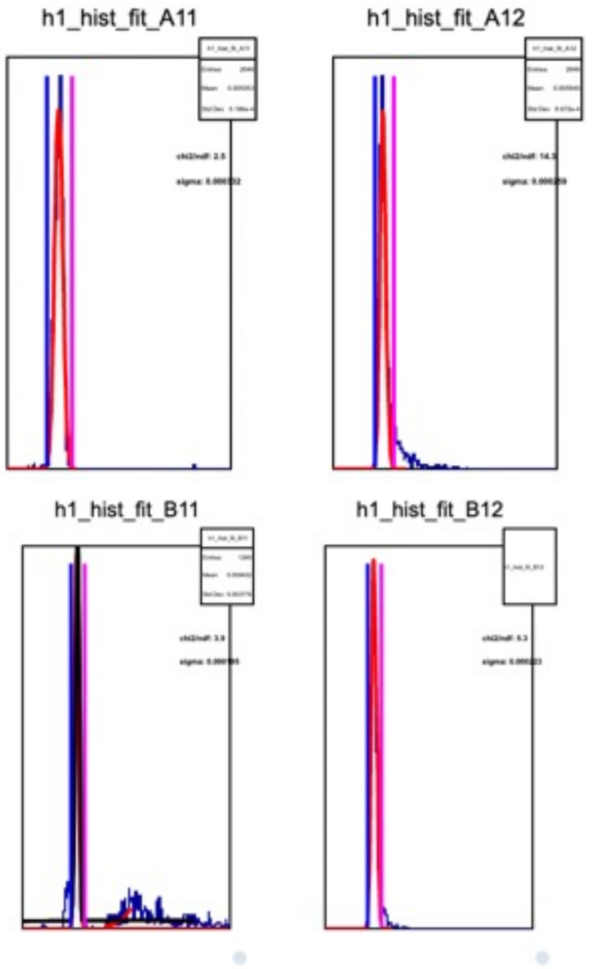


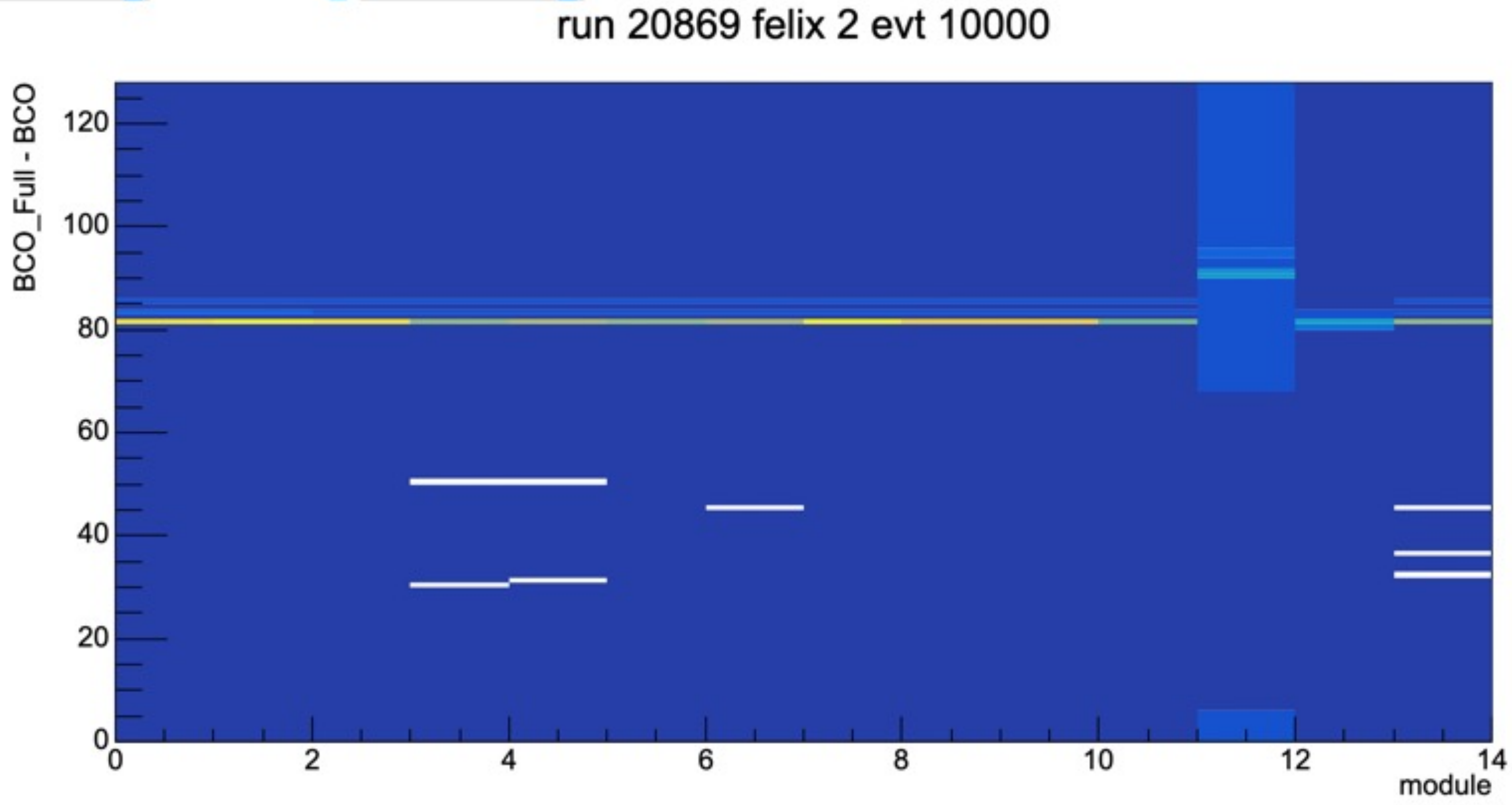




Problematic ladders(Felix 2 module 11,12)

Hit rate distribution before BCO cut – Weird shape of BCO distribution


h1_hist_fit_A11
chi2/ndf: 2.8
h1_hist_fit_A12
chi2/ndf: 14.3
h1_hist_fit_B11
chi2/ndf: 3.9
h1_hist_fit_B12
chi2/ndf: 5.3


run 20869 felix 2 evt 10000
BCO_Full - BCO
module