

Phase I Failure Scenarios

Tamás Álmos VÁMI¹, János KARANCSI²

¹ Wigner RCP, Budapest

² ATOMKI, Debrecen

Pixel Offline meeting



Reminder

Presentation from Janos:

https://indico.cern.ch/event/602634/contributions/2431075/attachments/1393899/2124262/2017_01_11_PixelOfflineMeeting_PixelFailureScenarios.pdf

Presentation from me earlier:

http://tvami.web.cern.ch/tvami/presentations/20160902_TrackerDPG_PixelPhase1FailureScenarios.pdf

<https://twiki.cern.ch/twiki/bin/view/CMS/SiPixelPhase1FailureScenarios>

<https://tvami.web.cern.ch/tvami/projects/Phase1FailureScenarios/>

Scenarios

Scenario 0: Perfect Detector (New cabling map validation)

Scenario 00: No Pixel Detector

Scenario 1: Random 5%

Scenario 2: 1 BPix + 1 FPix cooling loop out

Scenario 3: BmO SEC4 L1+L4 out,
Bpl SEC6 L2+L3 out,
Bml Disk1 ROG3 HV2 out
BpO Disk2 ROG1 HV1 out

Scenario 4: 1 BPix + 1 FPix FED out

Scenario 5: 1 BPix Read-Out group is out

Method

MC Simulation with 80.000 events (MinBias)

SiPixelQuality object is used to „kill” the dead parts of the detector

Occupancy is plotted

For the efficiency plots the track distribution of the given scenario with the perfect detector scenario are divided
(different MinBias jobs → can result efficiency>1)

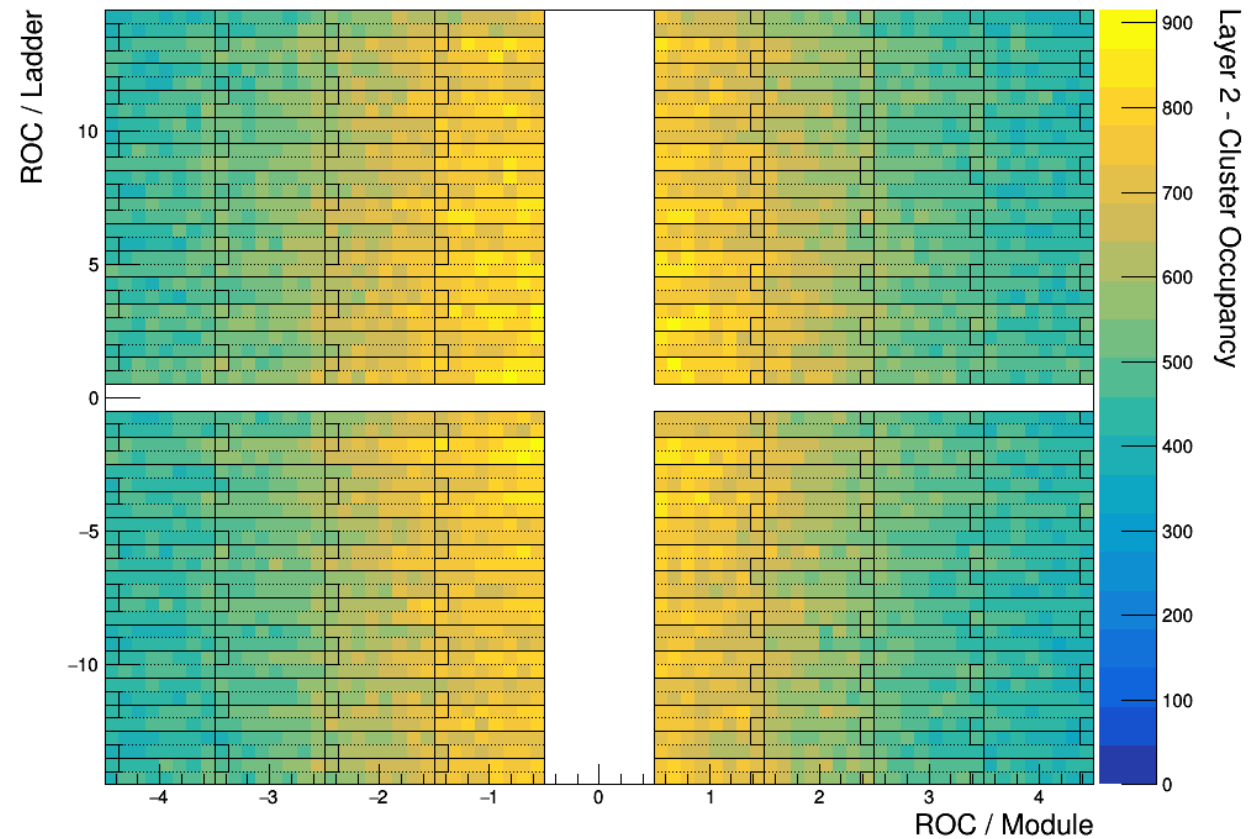
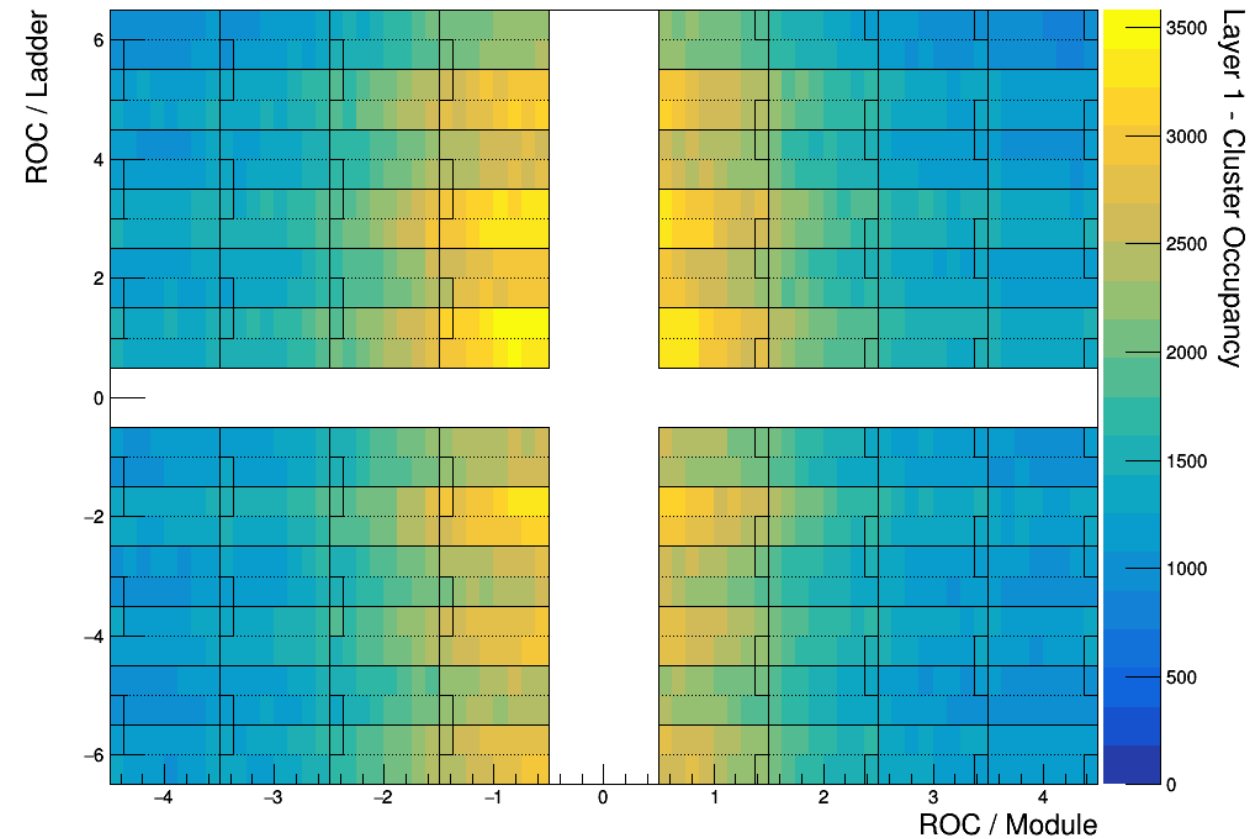
Scenario 0: Perfect detector

New cabling map

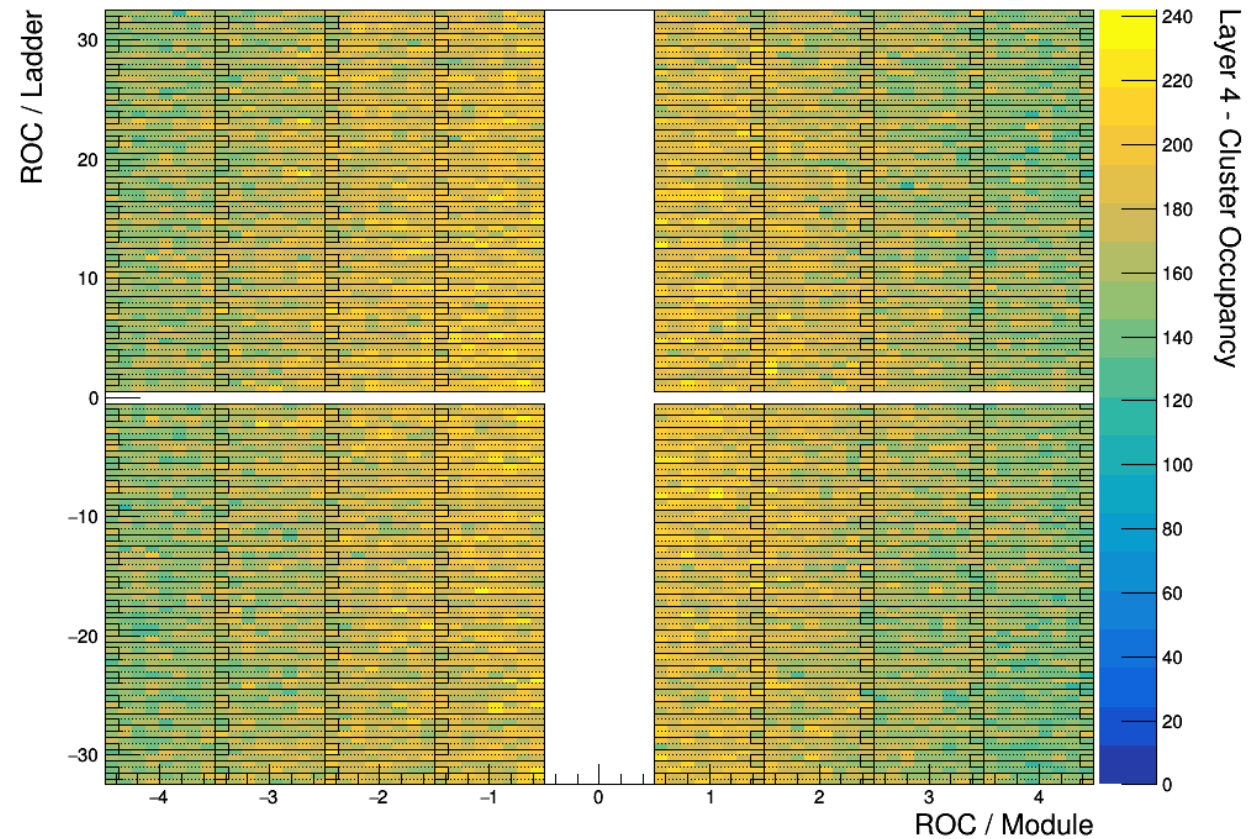
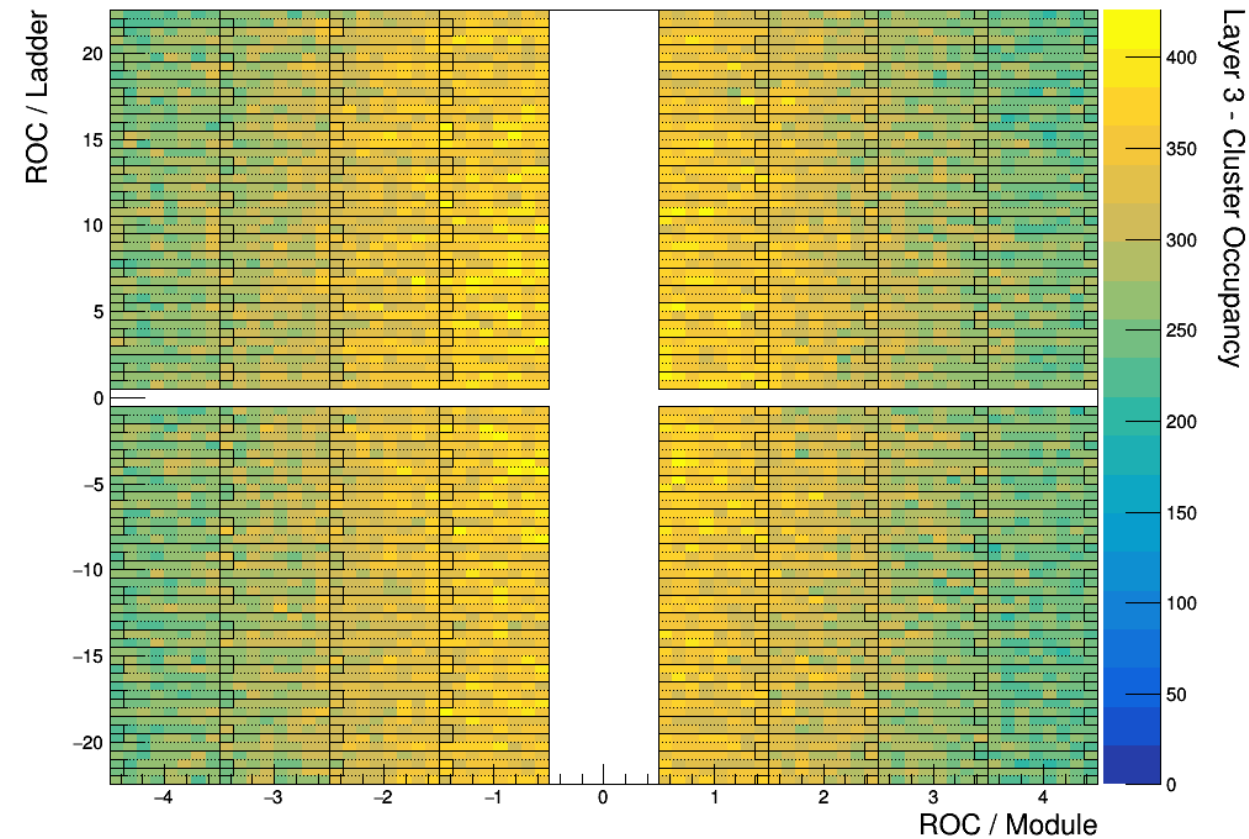
Tag name: SiPixelFedCablingMap_phase1_v6

Difference: FEDs went in the order 5,6,7,8 before but now they are in the order 8,7,6,5.

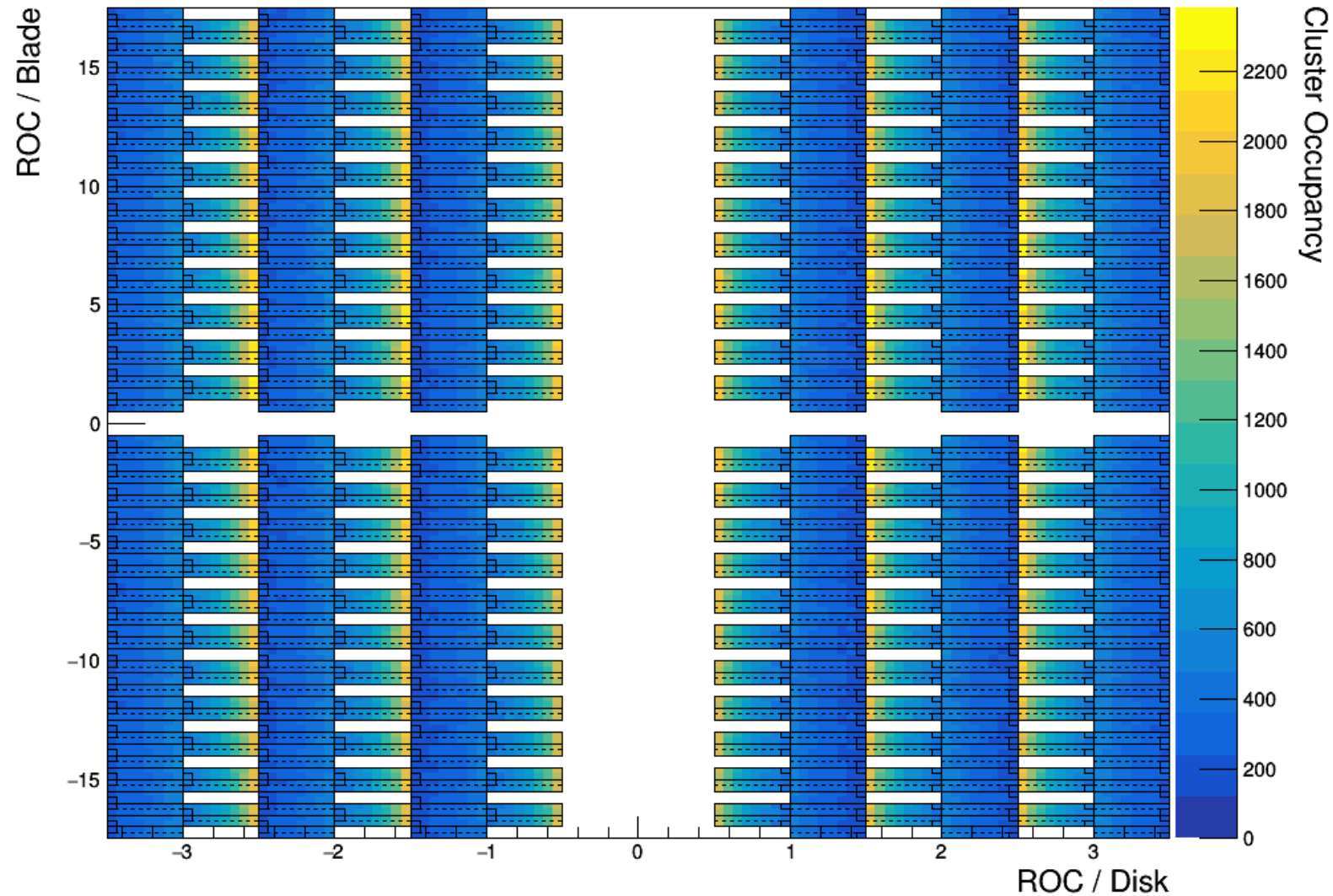
Scenario 0: Perfect detector



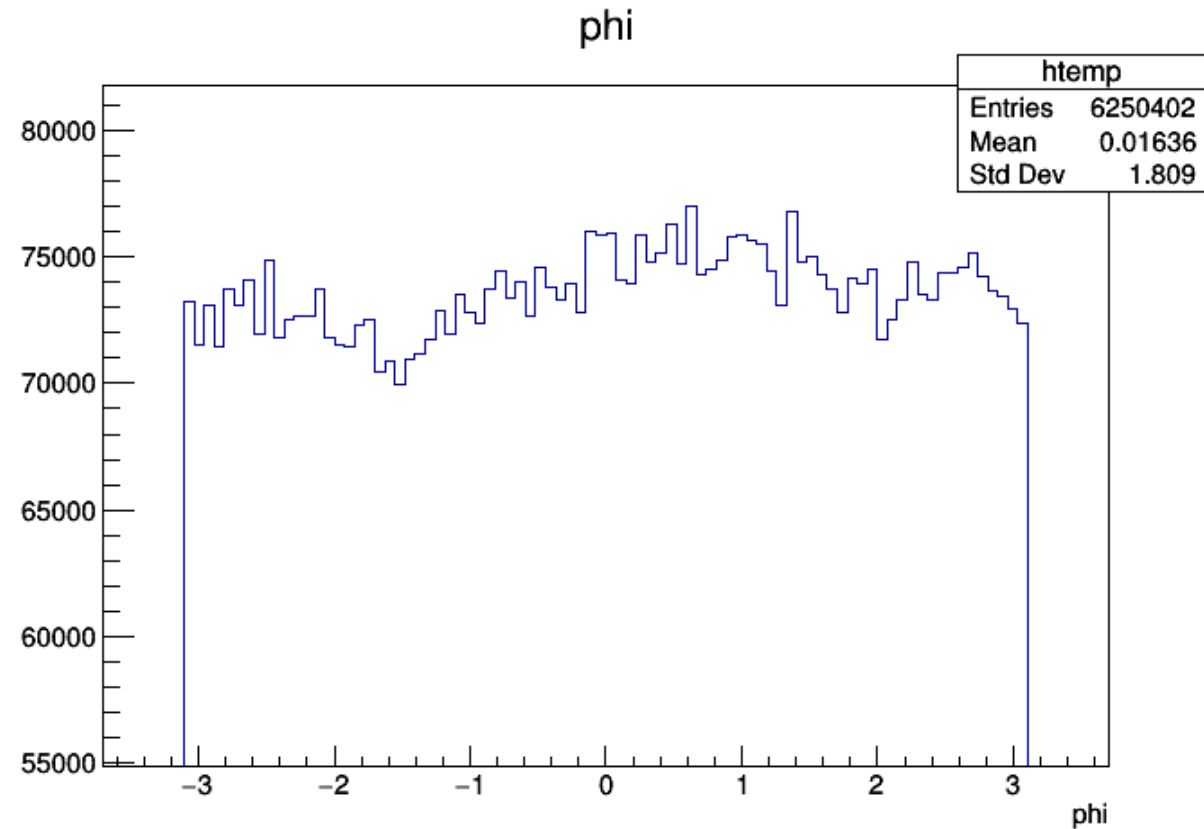
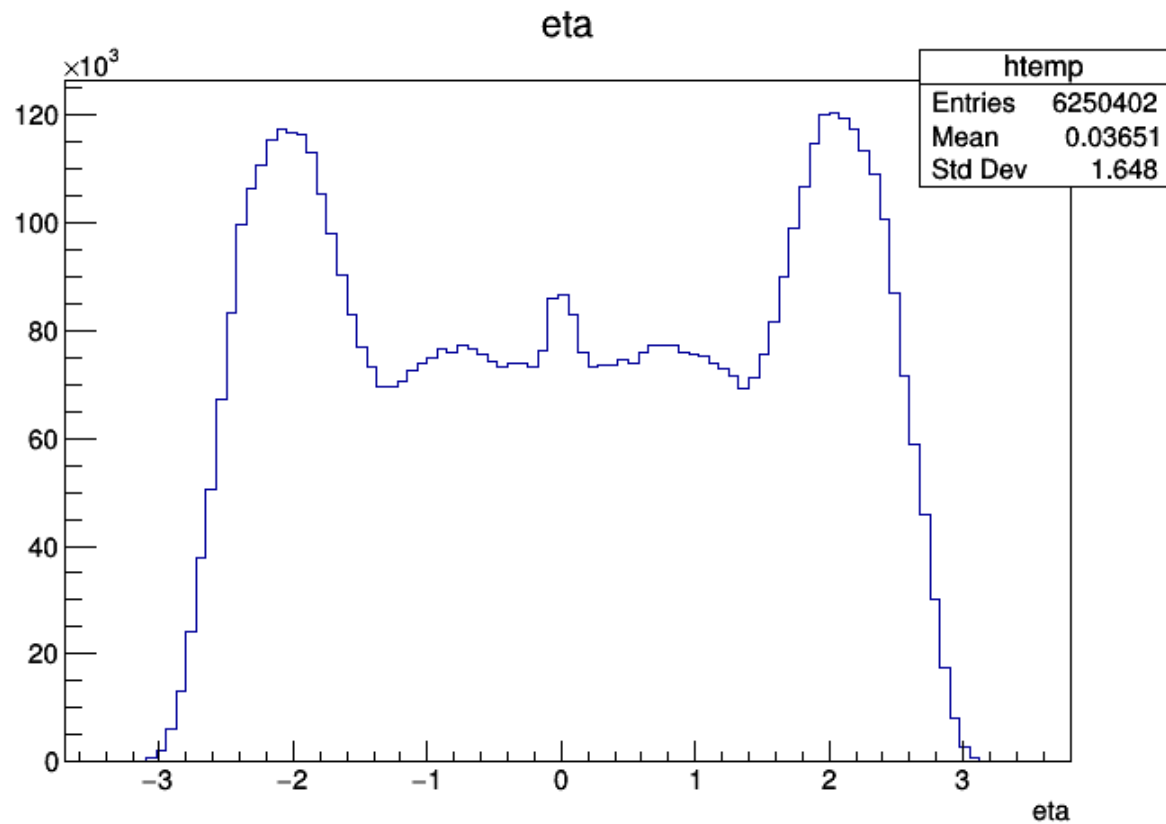
Scenario 0: Perfect detector



Scenario 0: Perfect detector



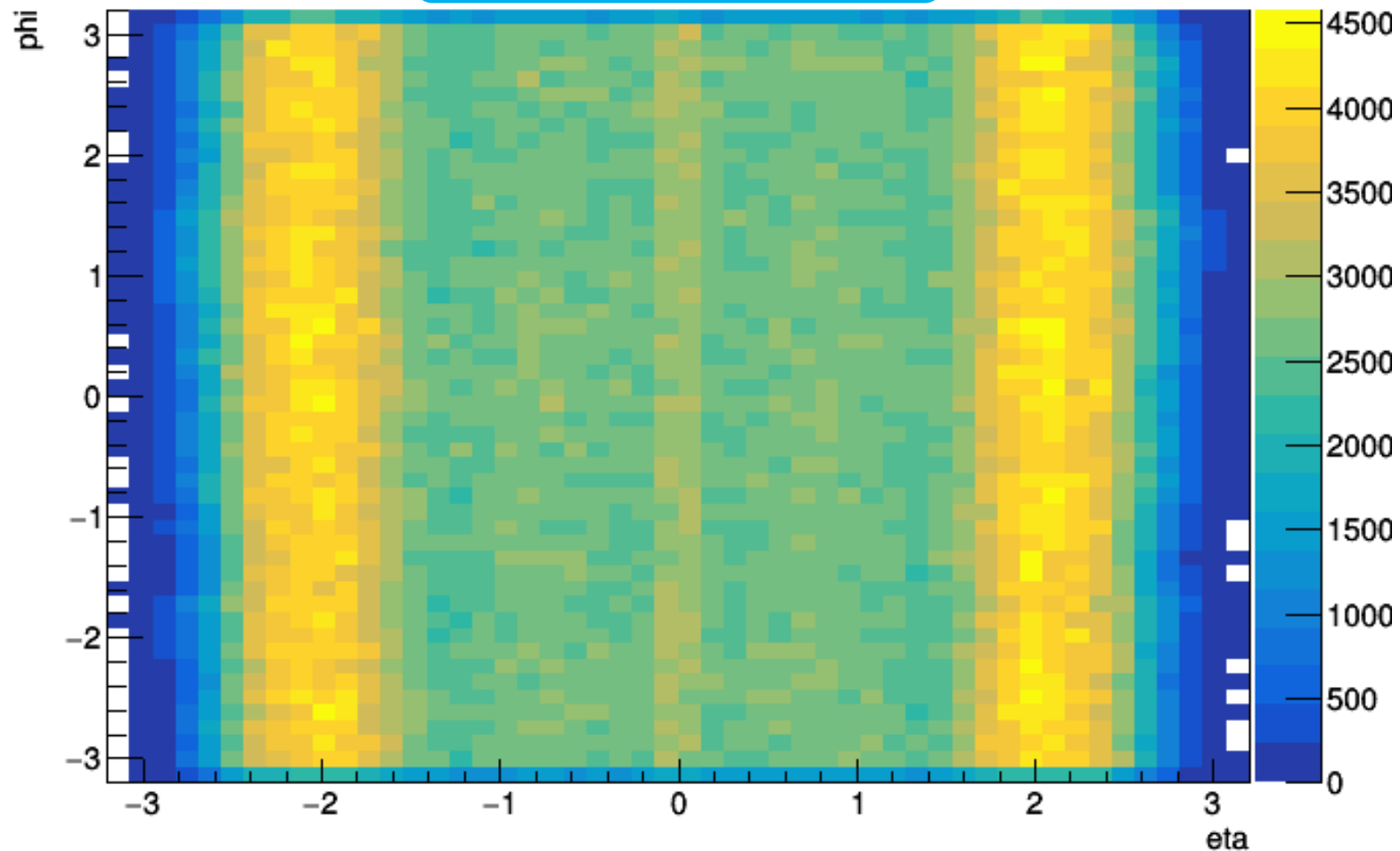
Scenario 0: Perfect detector



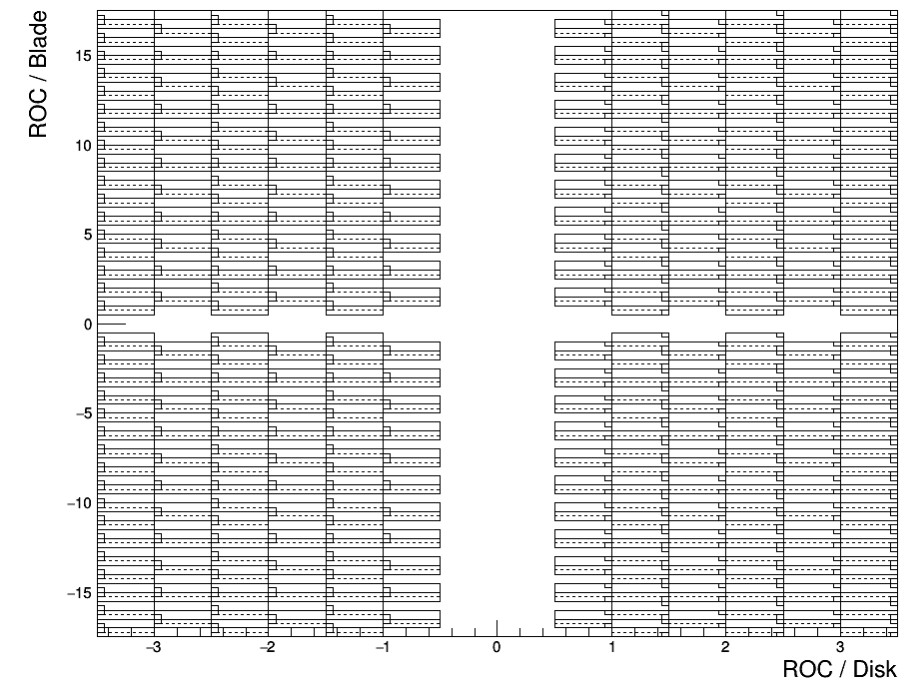
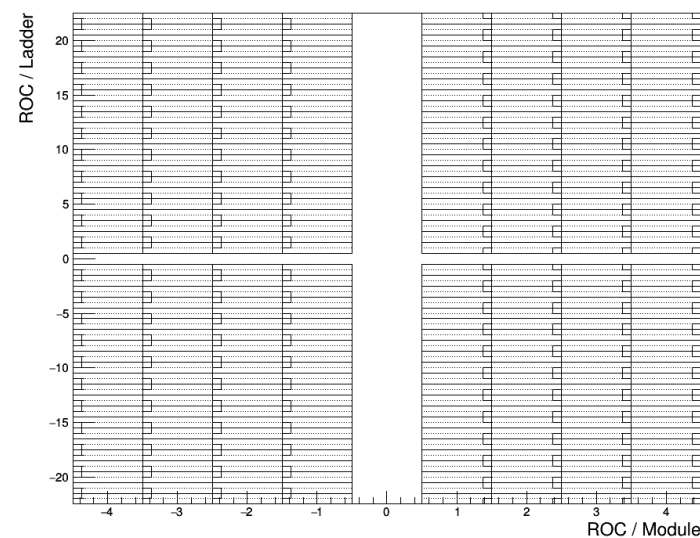
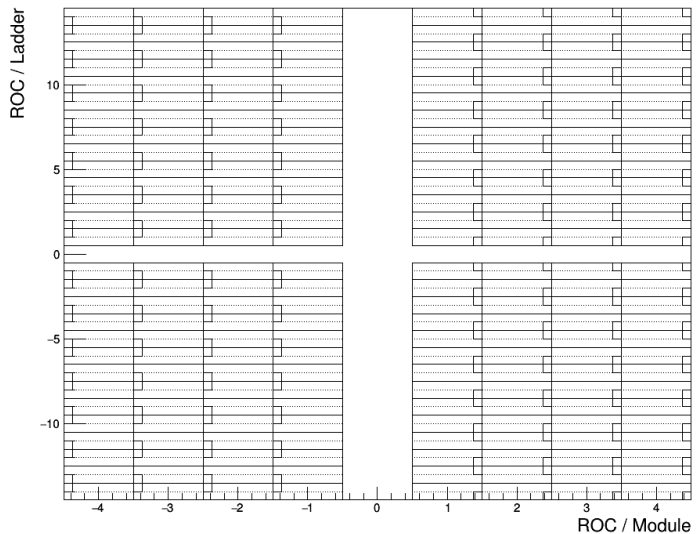
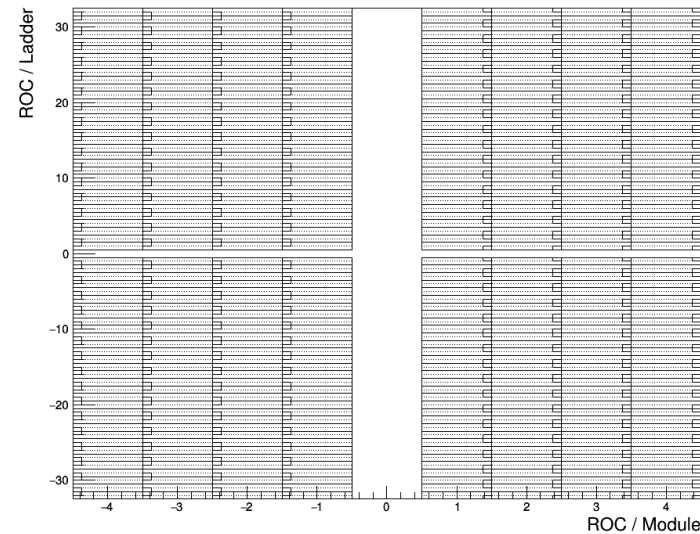
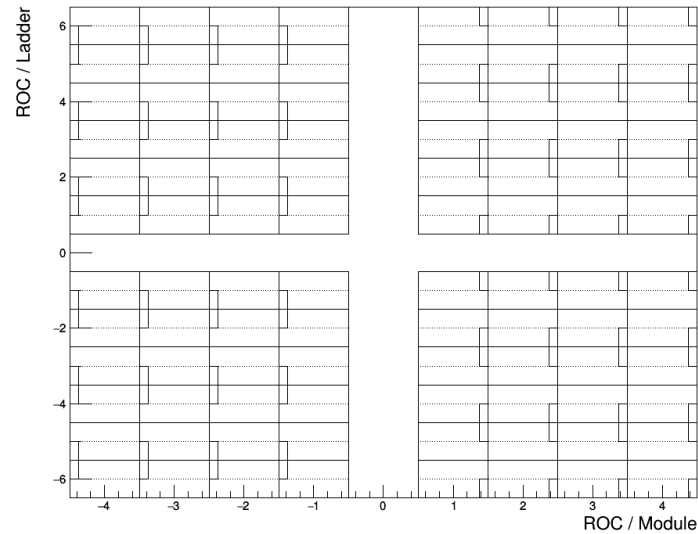
Hit distributions

Scenario 0: Perfect detector

Hit distribution



Scenario 00: No pixel detector



Strip only tracking
can be tested

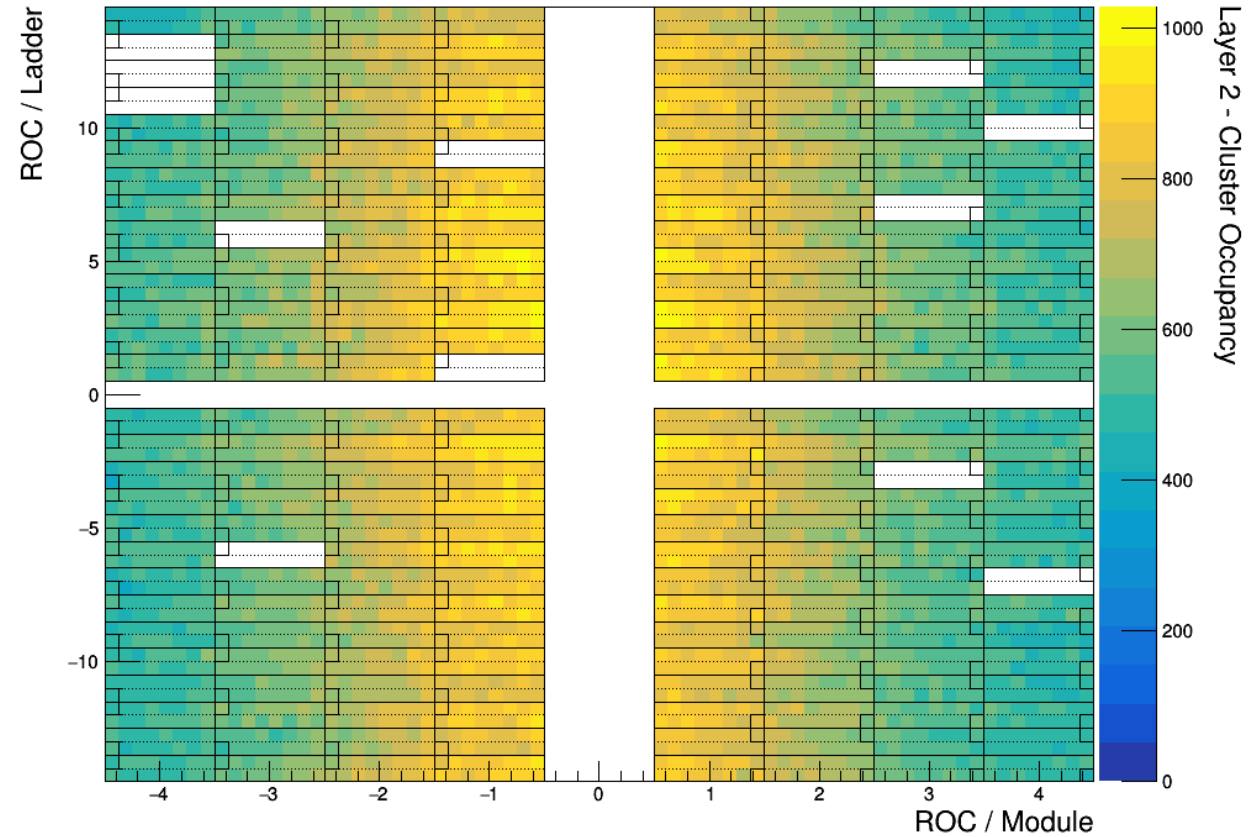
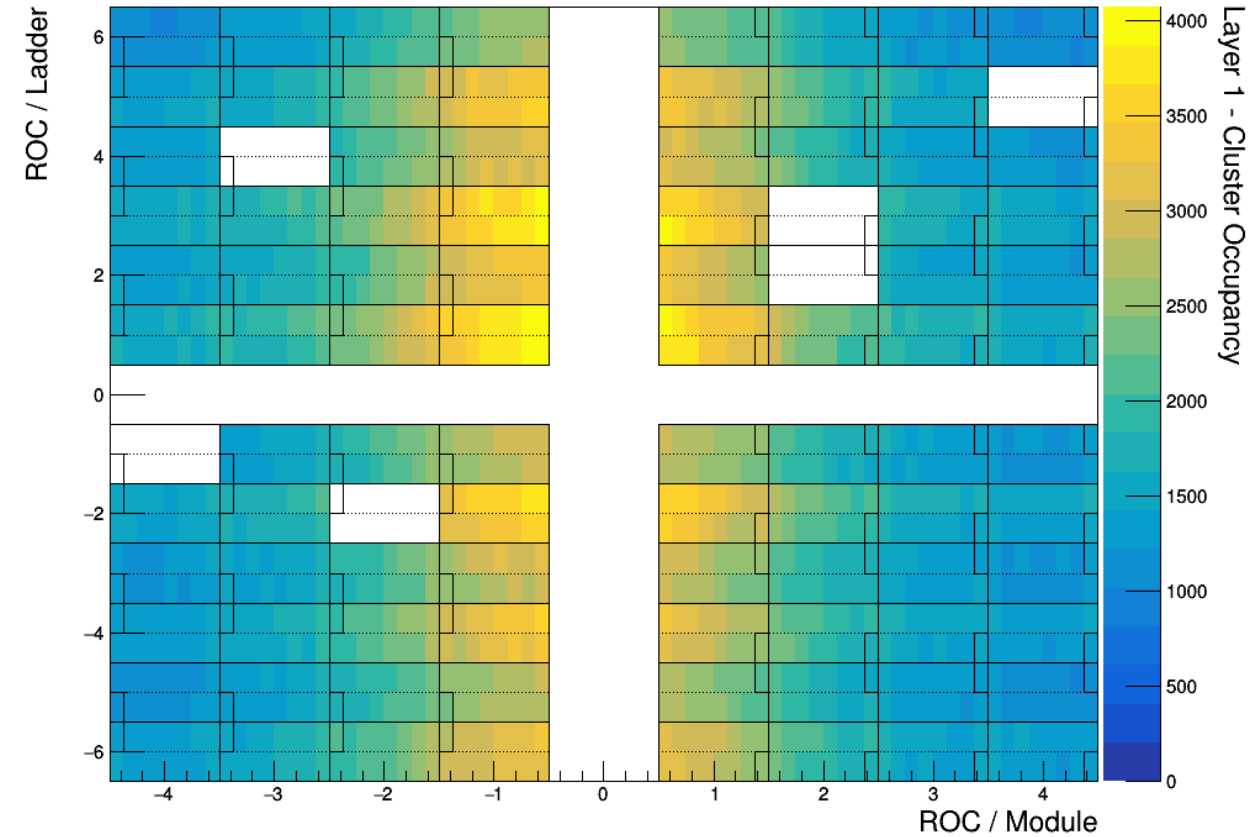
Scenario 1: Random 5%

Print all of the detids

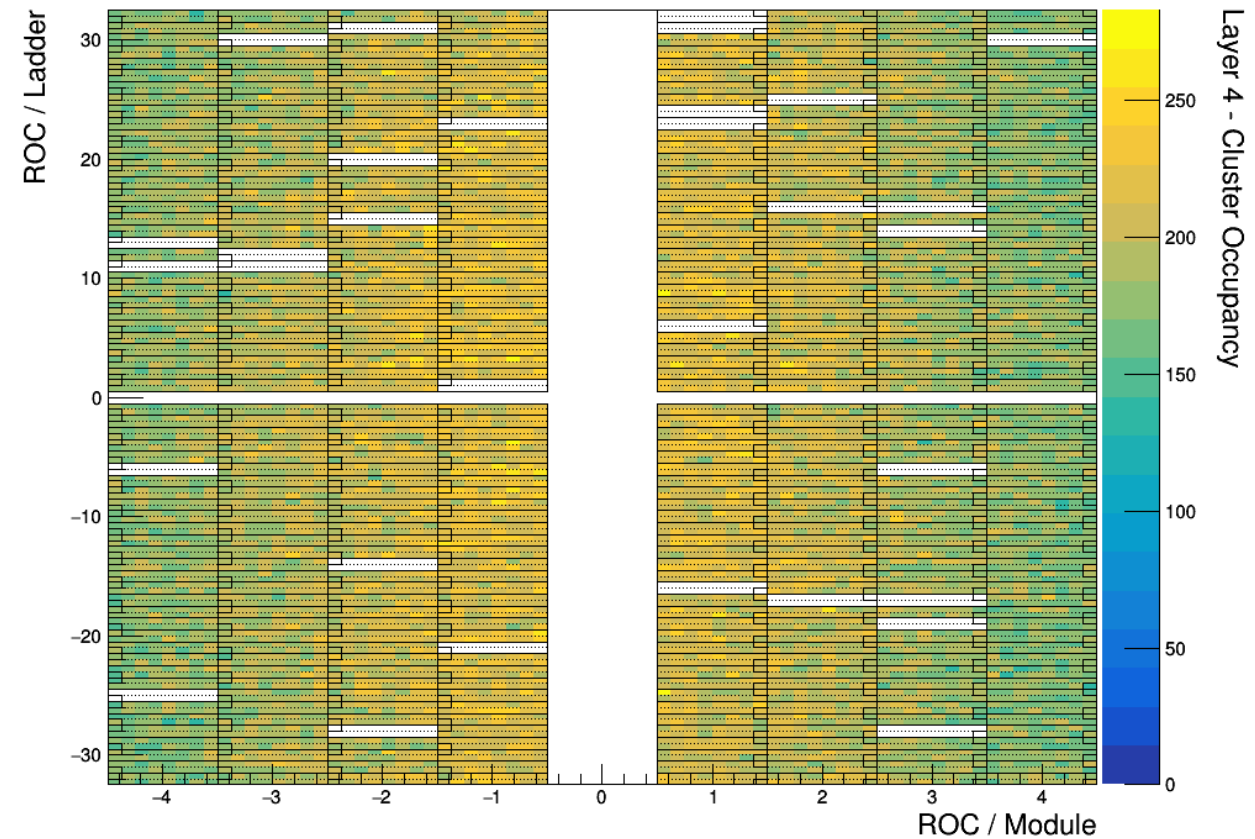
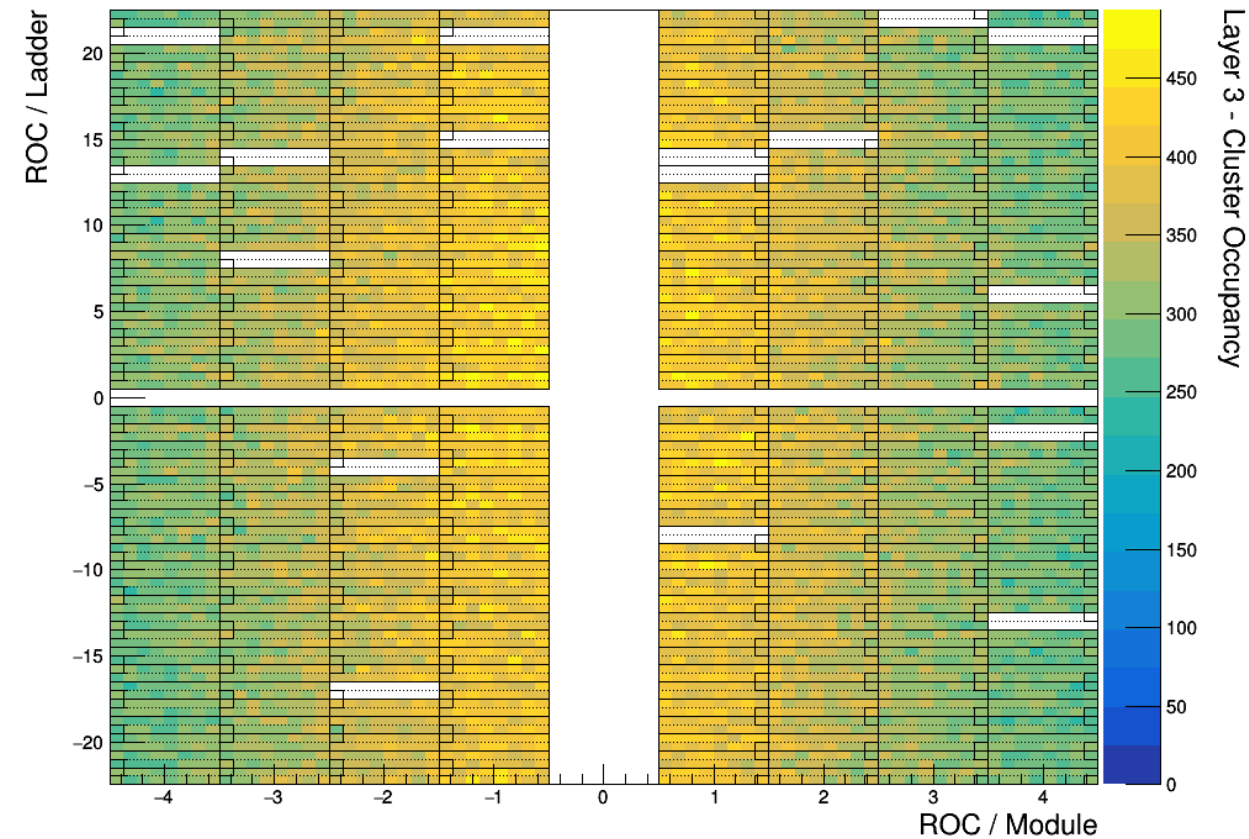
Randomize them with Python's random algorithm

Select 5% of them

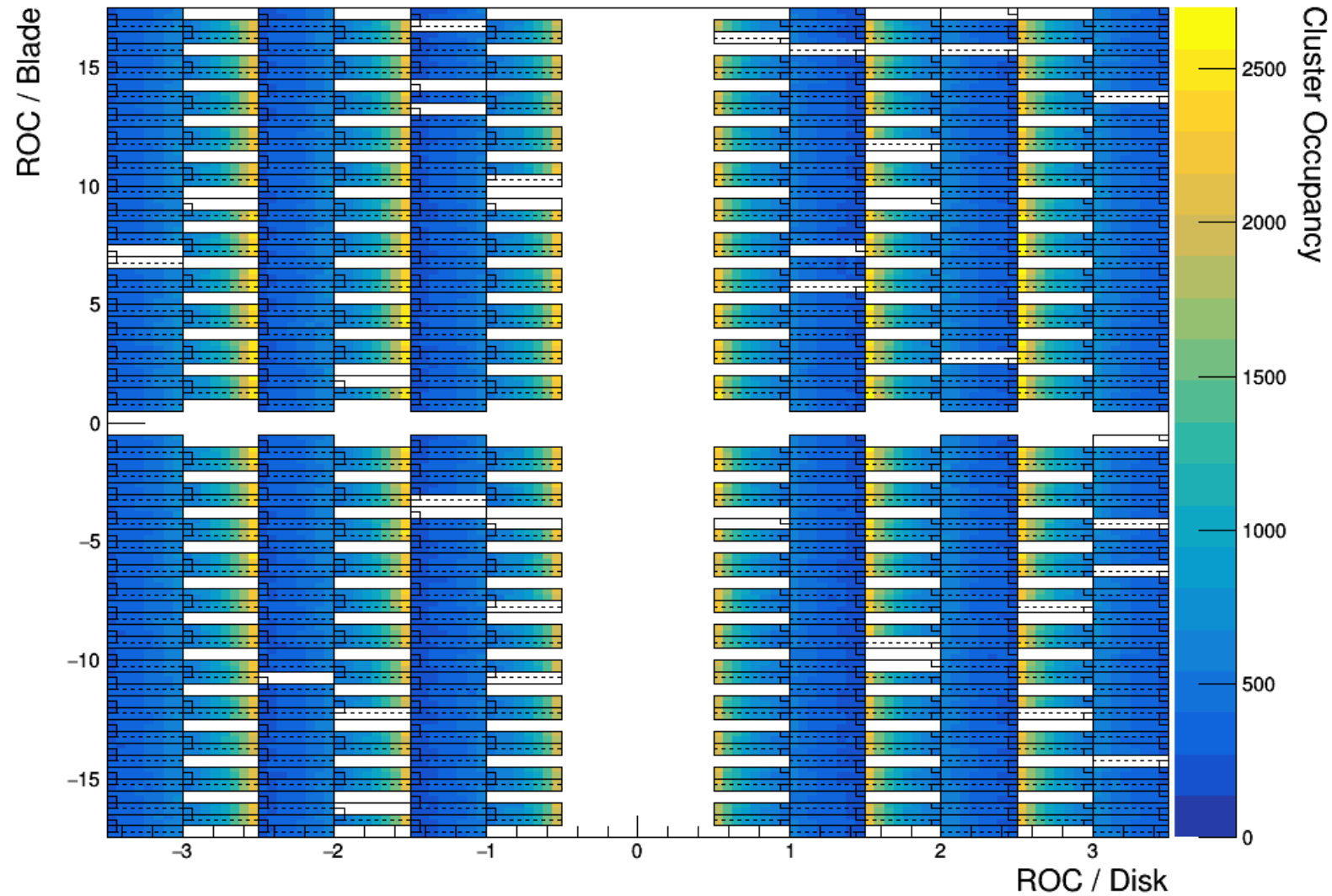
Scenario 1: Random 5%



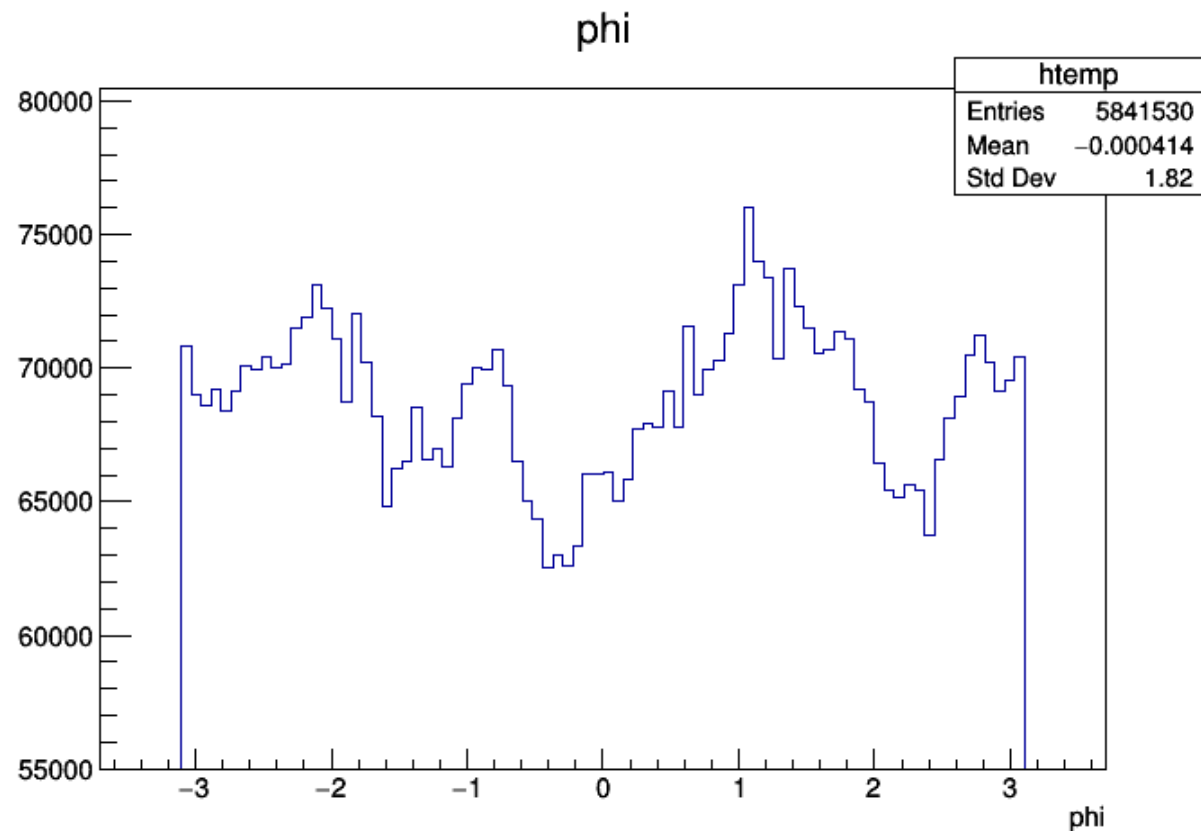
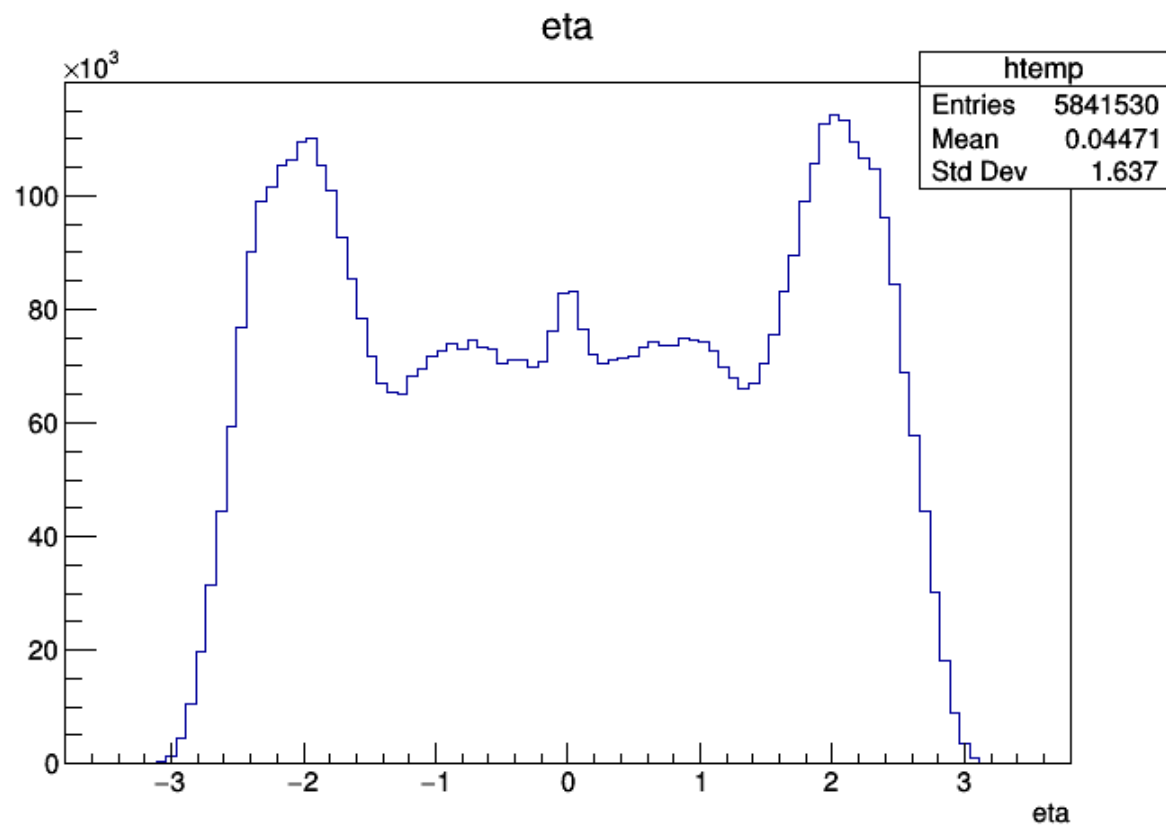
Scenario 1: Random 5%



Scenario 1: Random 5%



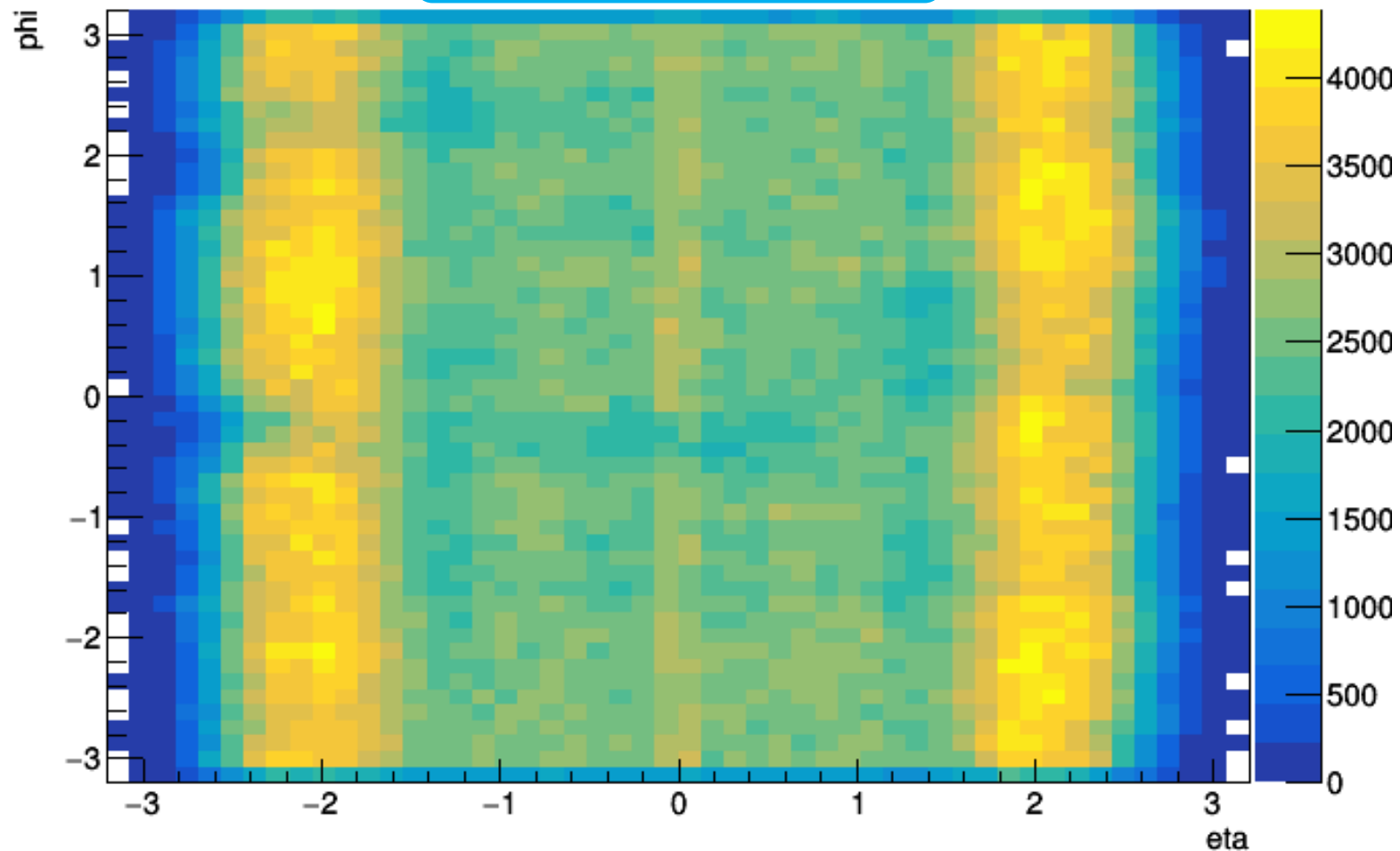
Scenario 1: Random 5%



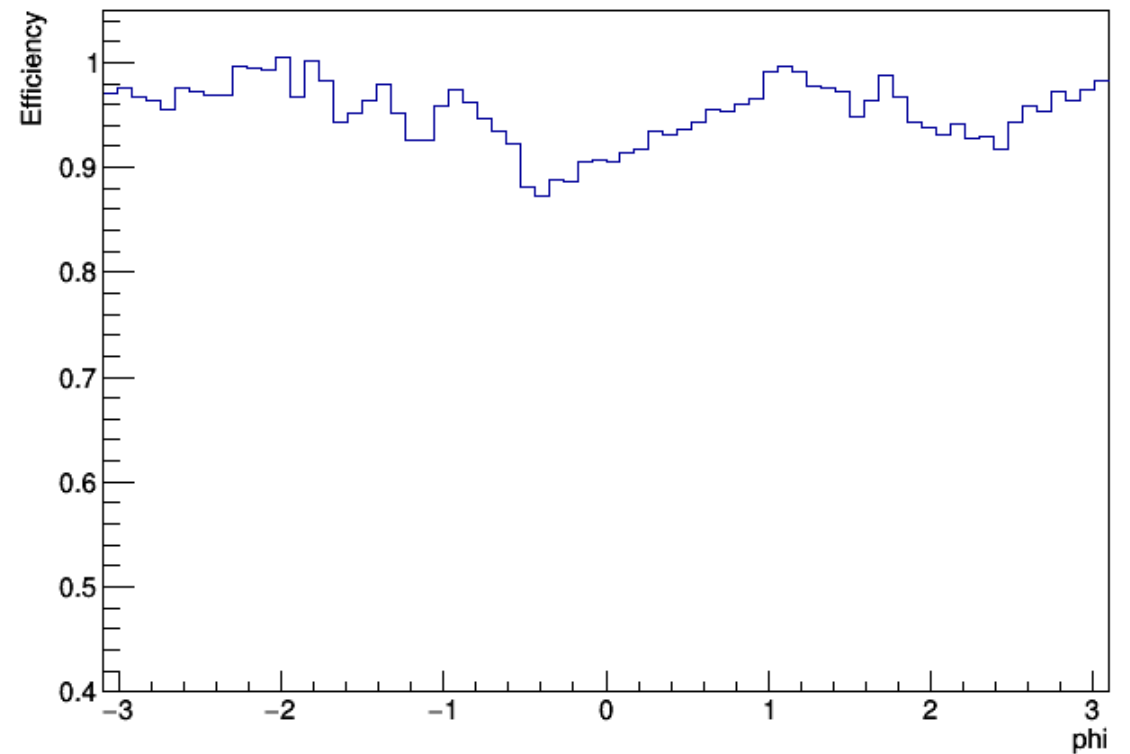
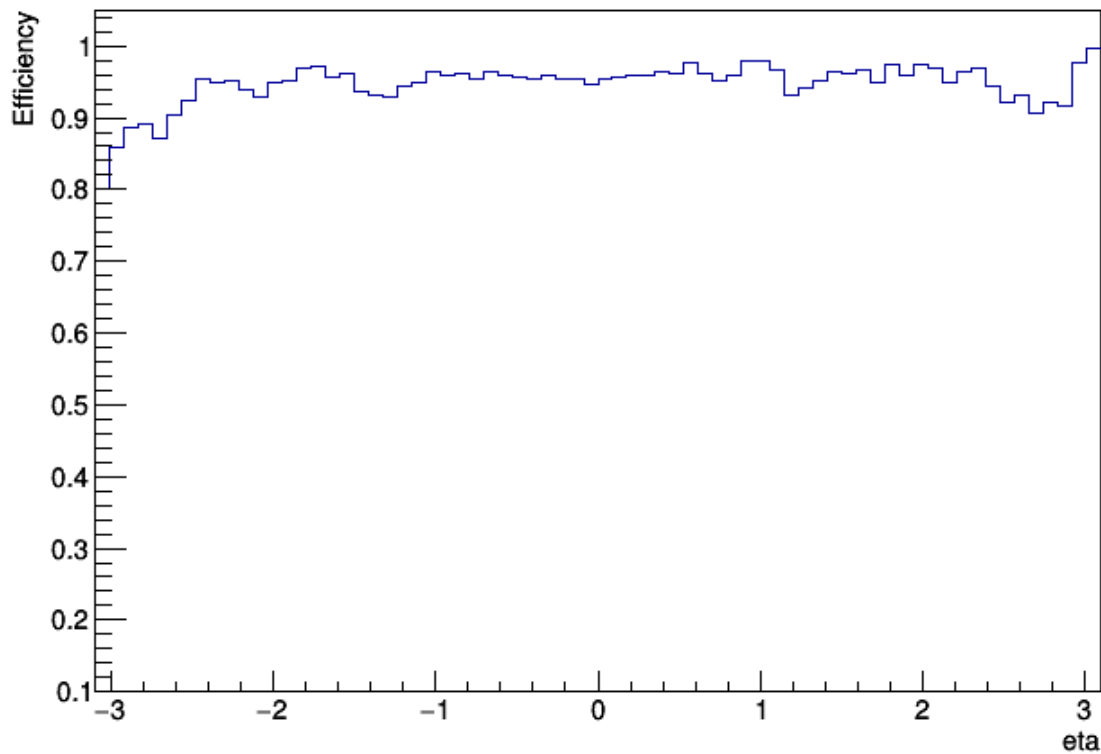
Hit distributions

Scenario 1: Random 5%

Hit distribution



Scenario 1: Random 5%

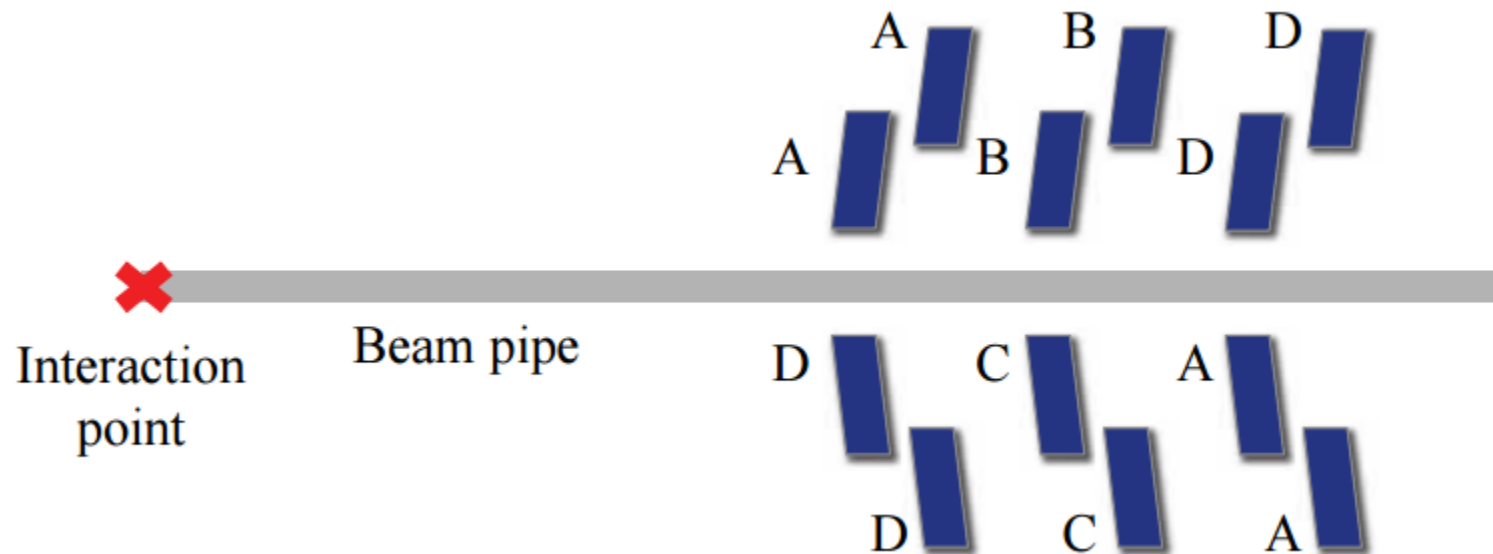


Tracking efficiencies

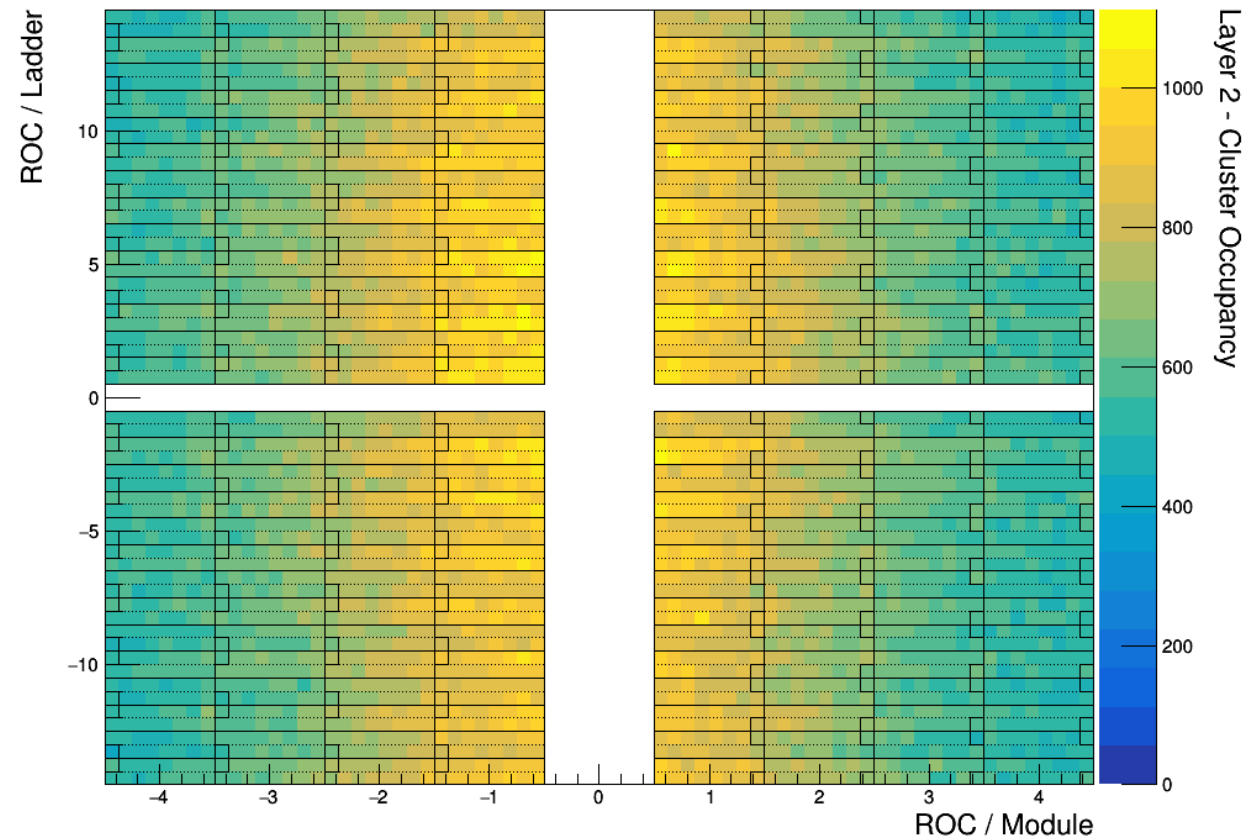
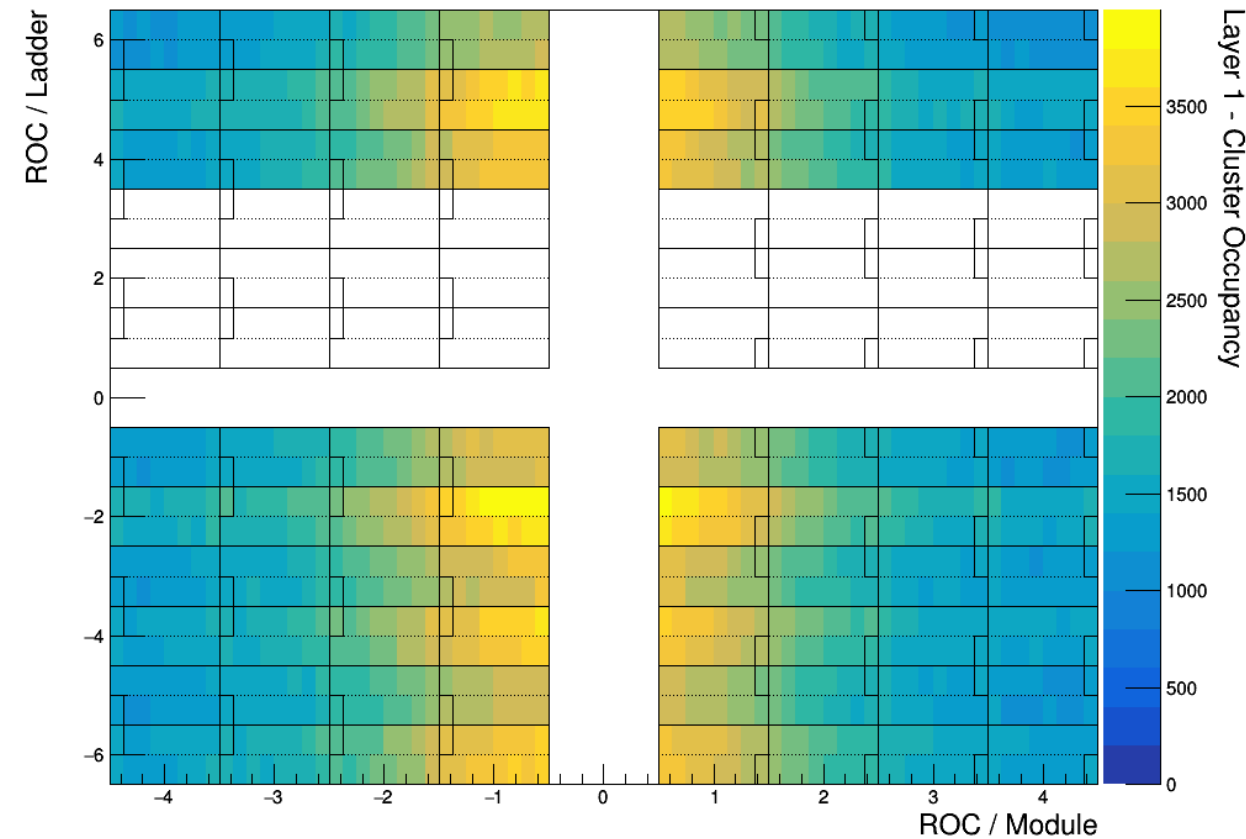
Scenario 2: 1 BPix + 1 FPix cooling loop

BPix: From Lea: [Phase1_BPIX_CoolingLoop-To-Ladder.xlsx](#)
I choose L1D1PN cooling loop located on Layer 1

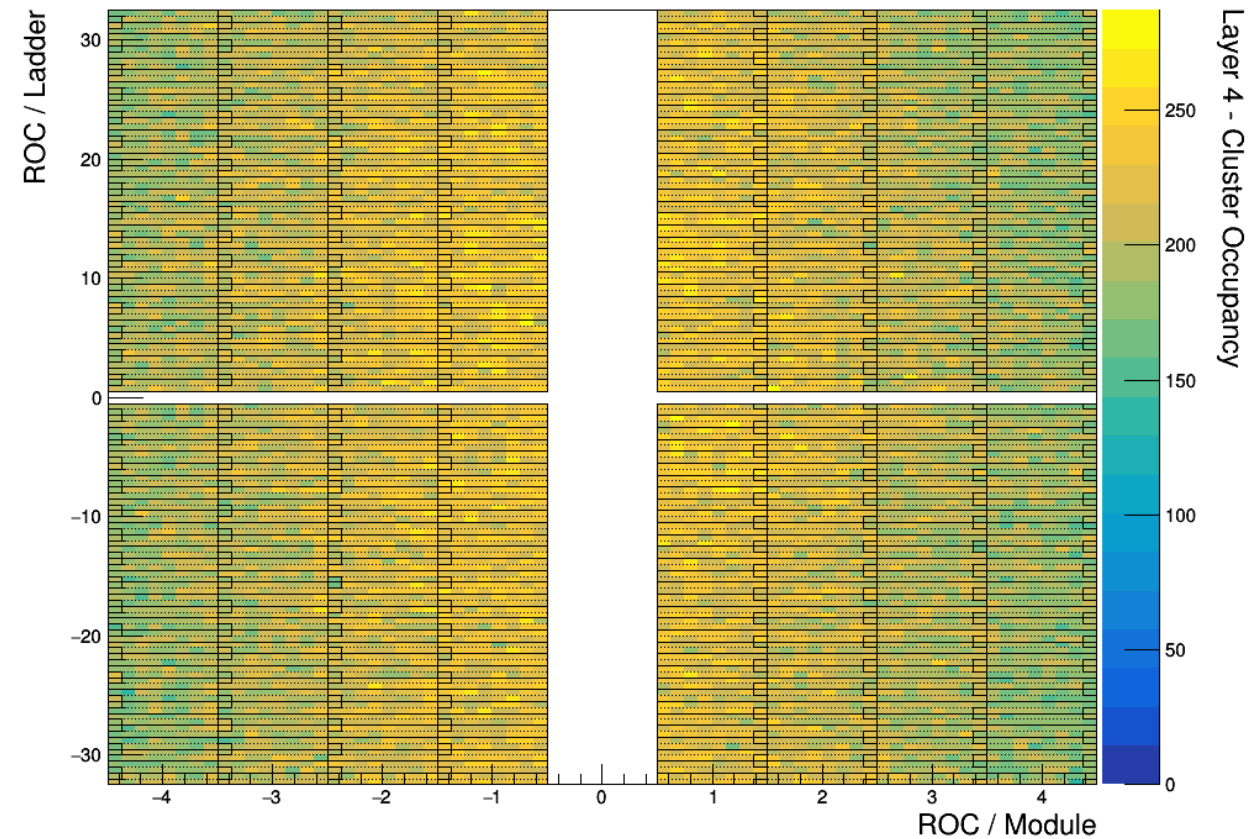
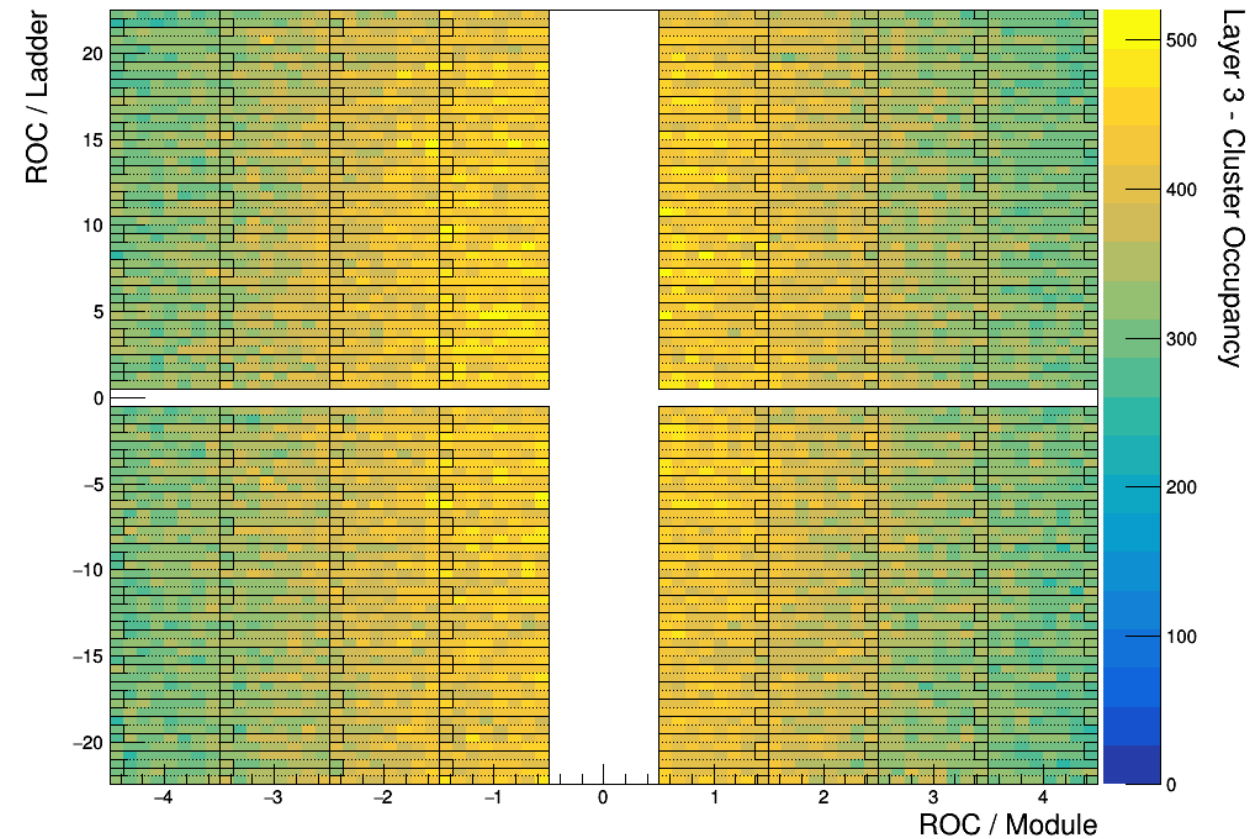
FPix: From Marco (Verzocchi): [CoolingFailures.pdf](#)
I choose Loop A „half disk 1 of HC1 and half disk 3 of HC2”



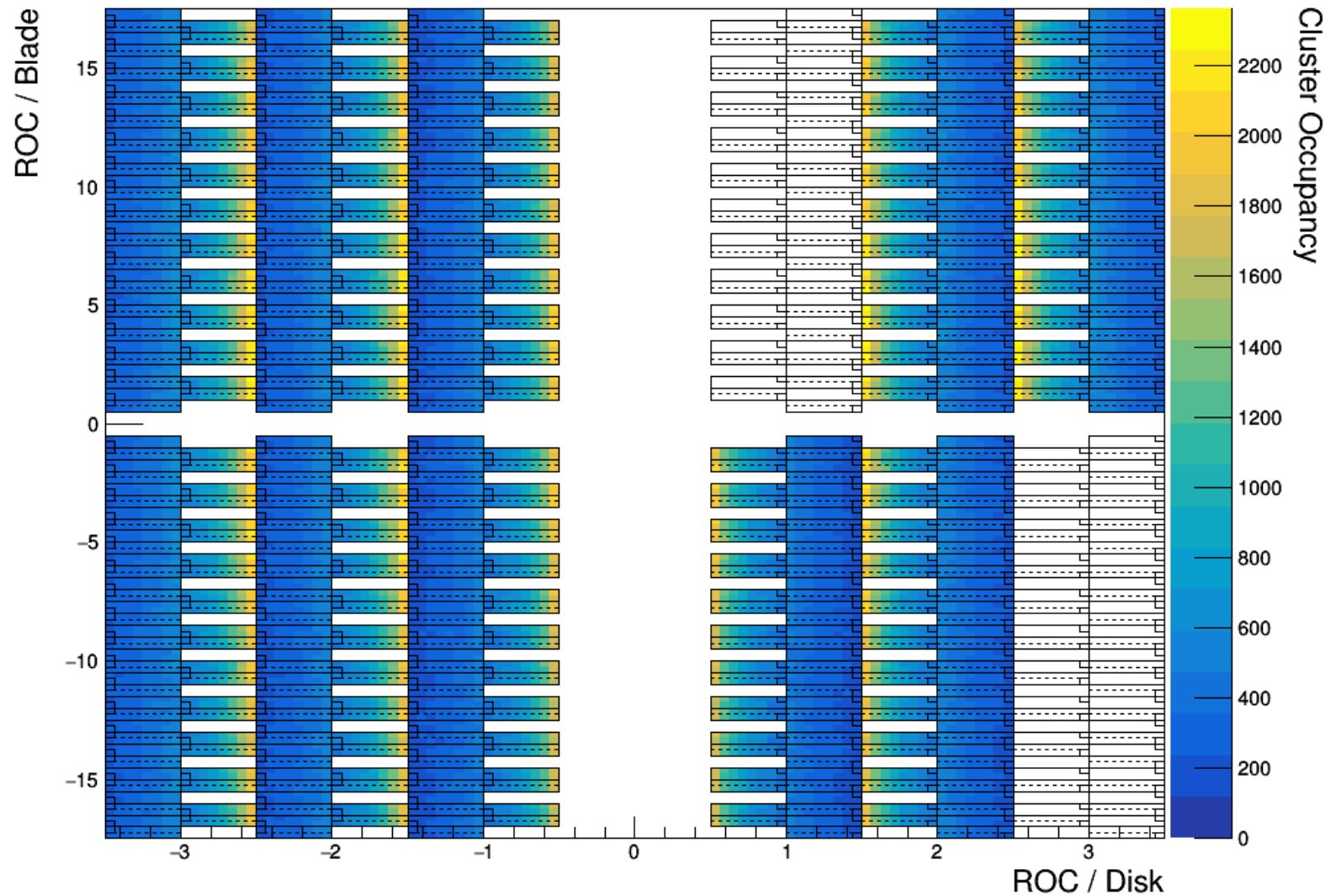
Scenario 2: 1 BPix + 1 FPix cooling loop



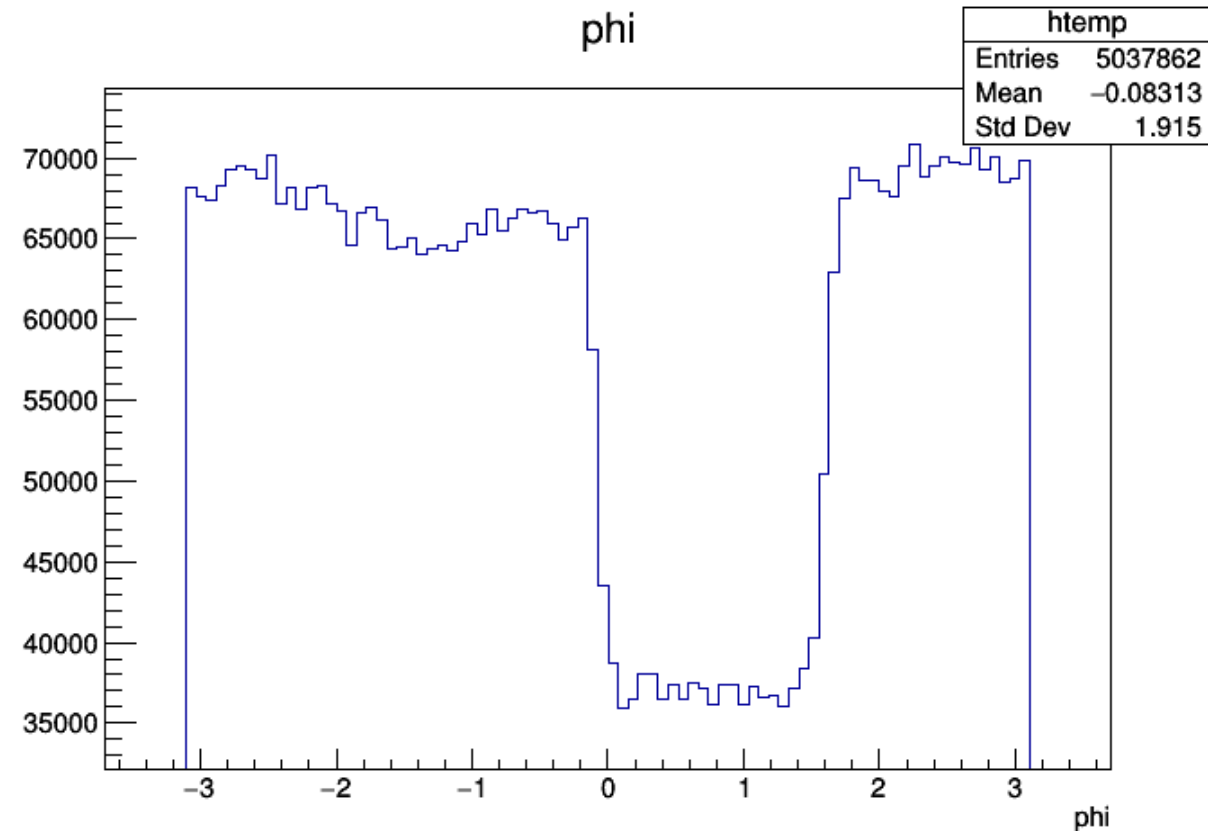
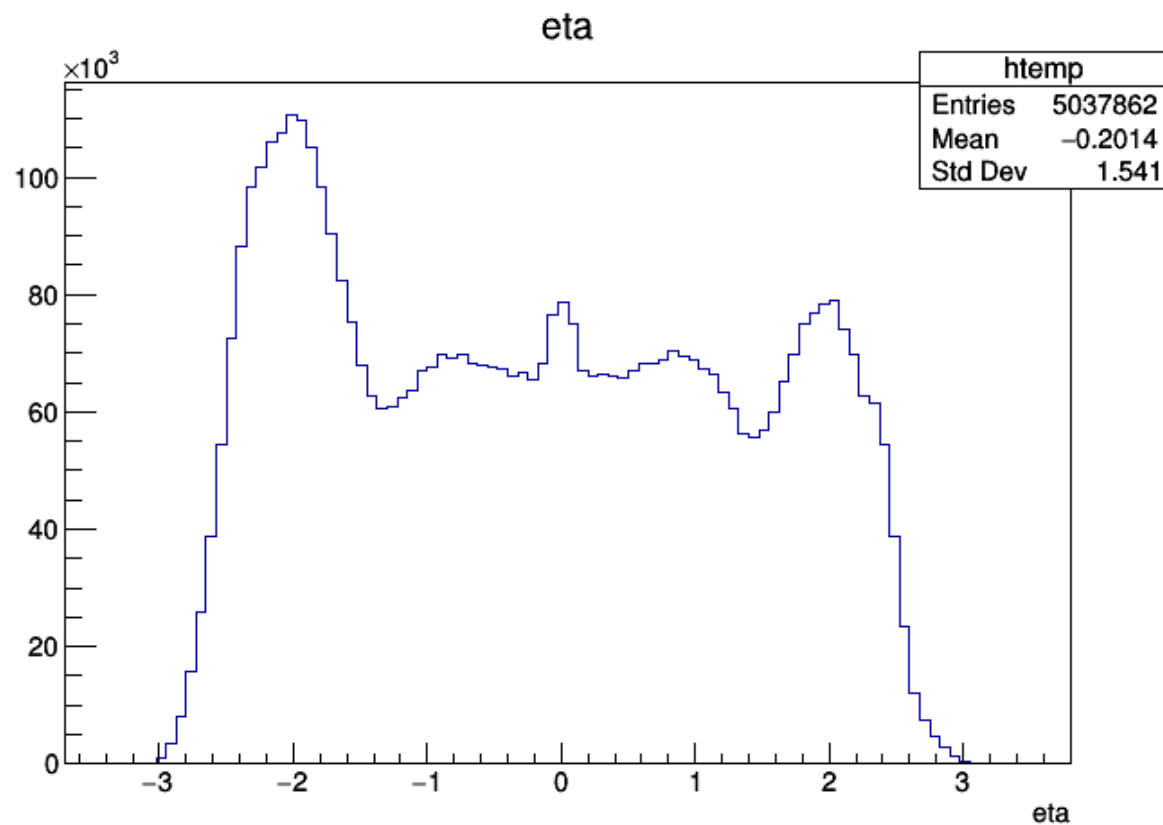
Scenario 2: 1 BPix + 1 FPix cooling loop



Scenario 2: 1 BPix + 1 FPix cooling loop



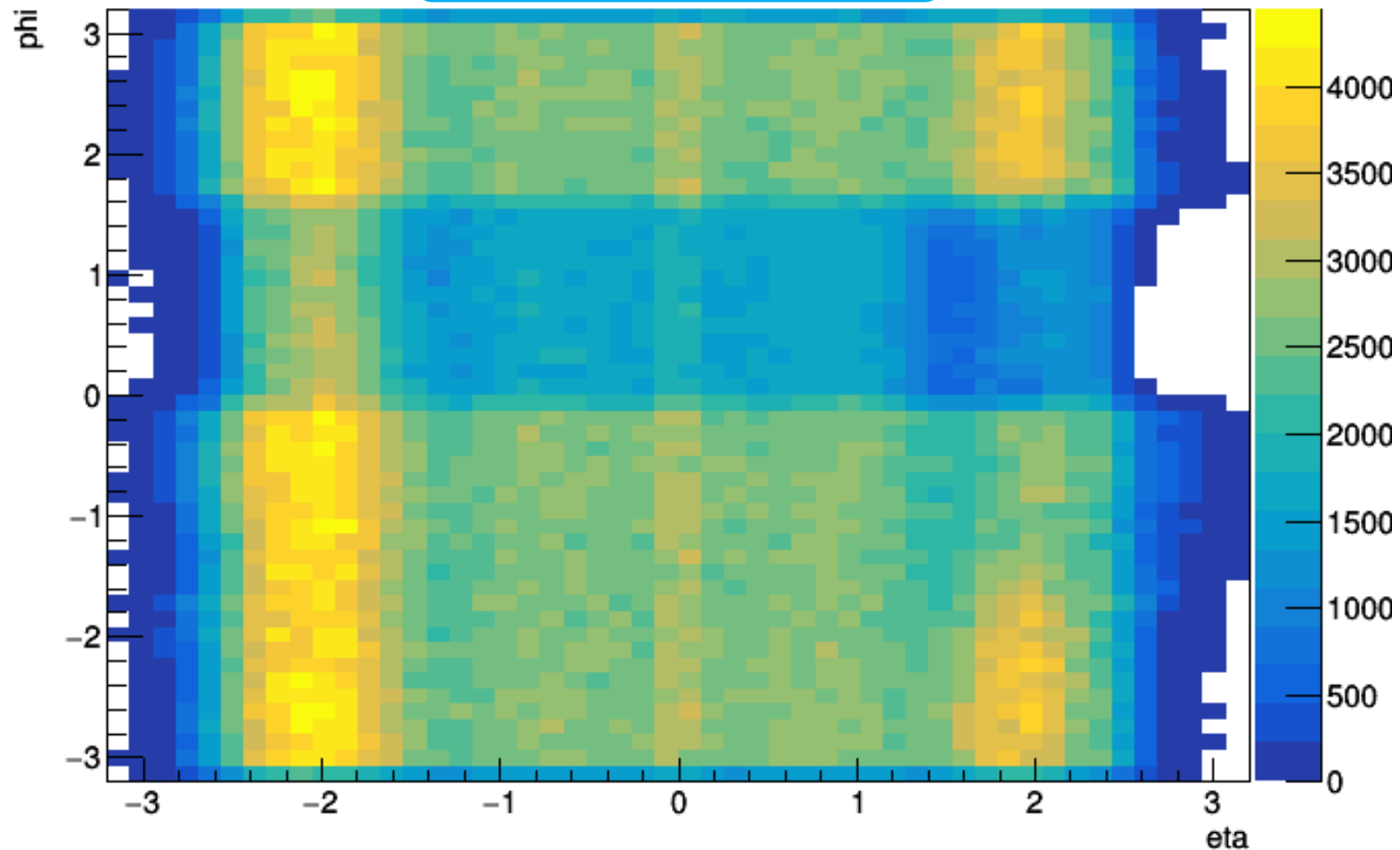
Scenario 2: 1 BPix + 1 FPix cooling loop



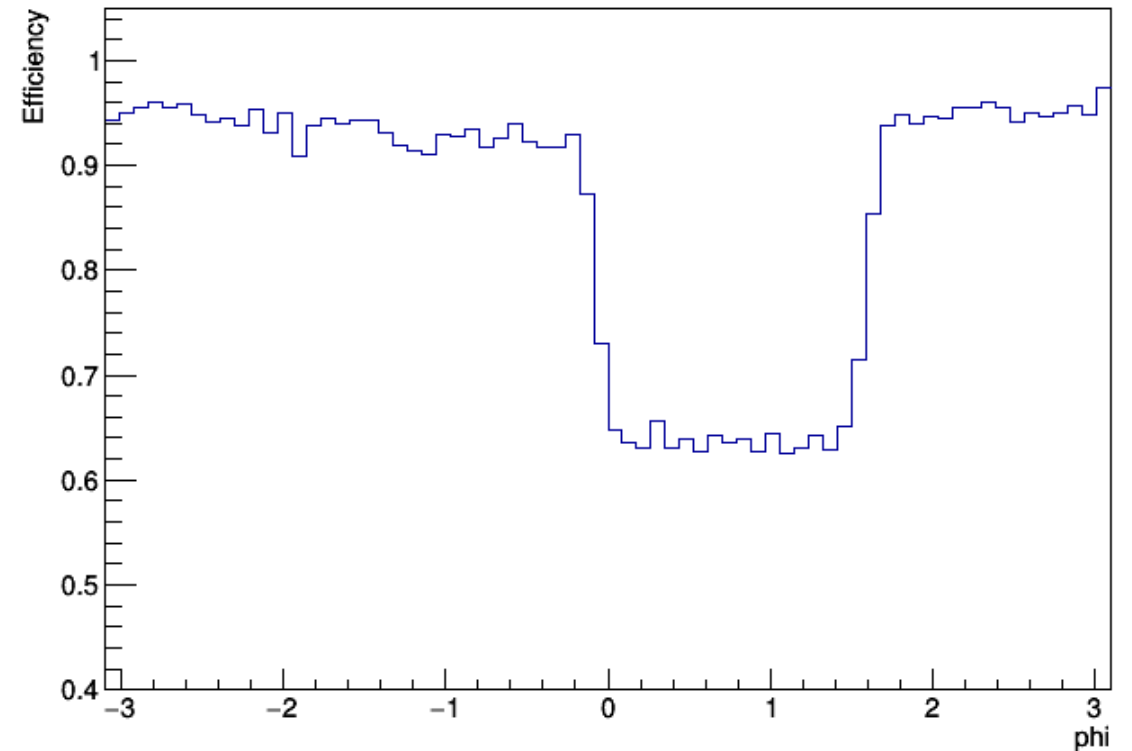
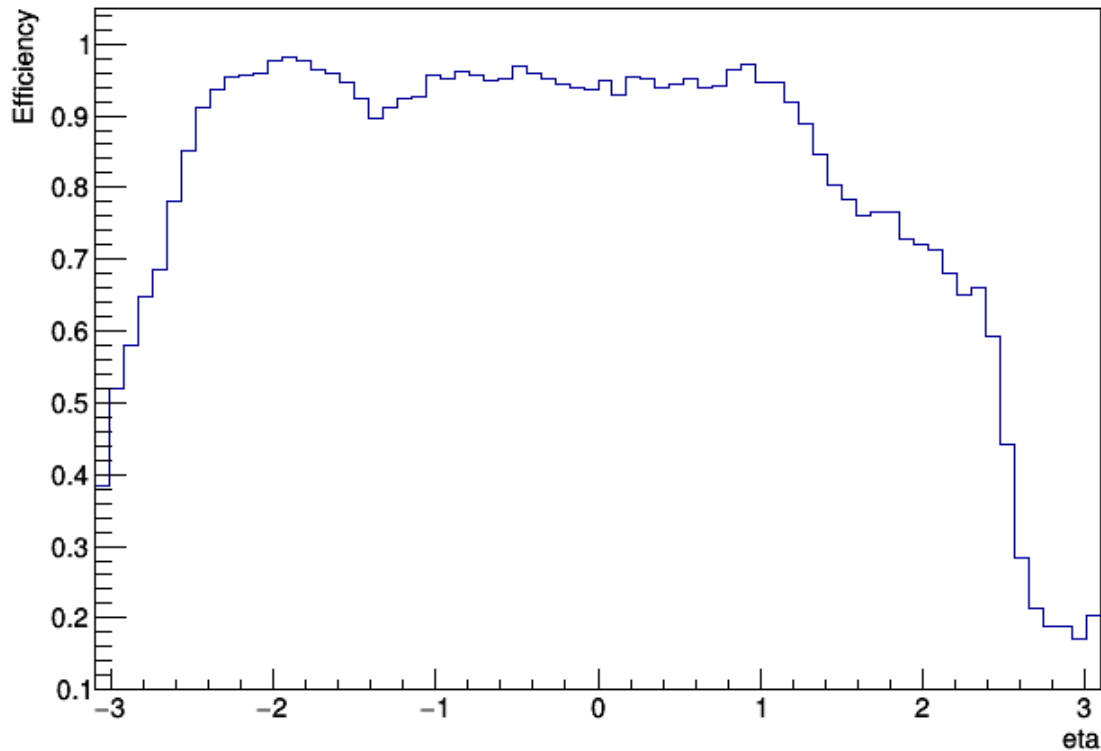
Hit distributions

Scenario 2: 1 BPix + 1 FPix cooling loop

Hit distribution



Scenario 2: 1 BPix + 1 FPix cooling loop



Tracking efficiencies

Scenario 3: HV group out

BPix: From Lea: [Phase1 BPIX modules per Sector.xlsx](#)

I choose BmO SEC4 L1+L4 out
+ BpI SEC6 L2+L3 out

FPix: From Stefanos: [Cabling Map for FPix Phase 1 Detector.pdf](#)

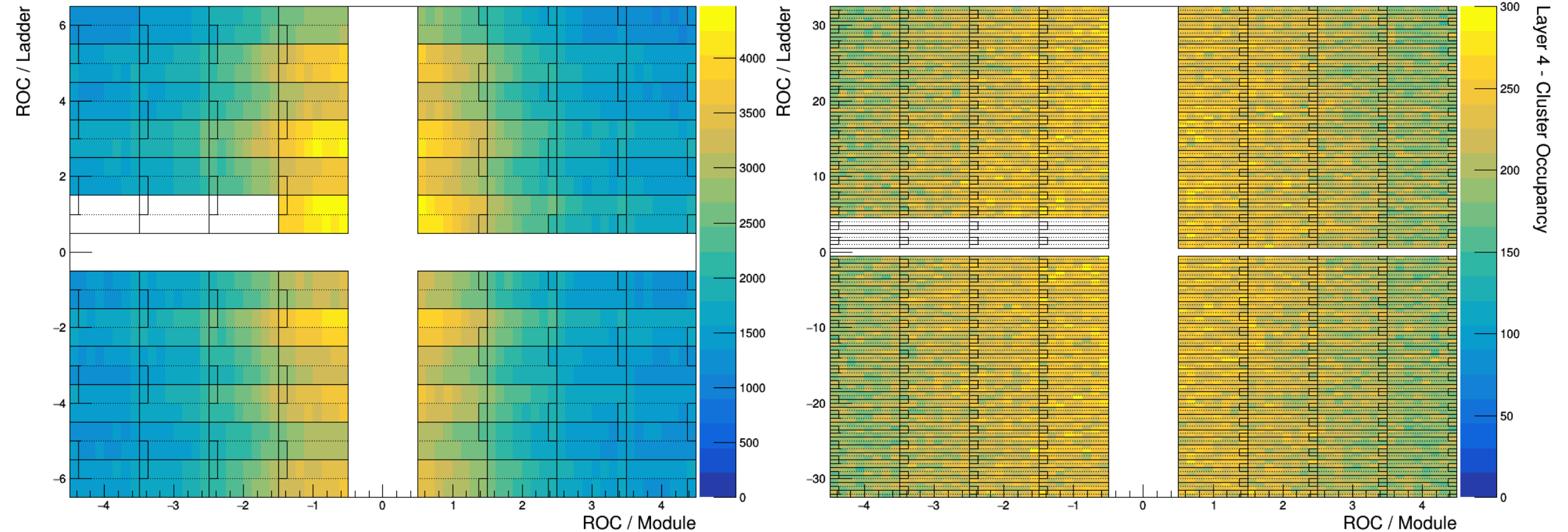
BmI Disk1 ROG3 HV2: Ring 1 Blade 6-7-8 Panel 1-2

BpO Disk2 ROG1 HV1: Ring 2 Blade 10-11-12-13 Panel 1-2

.

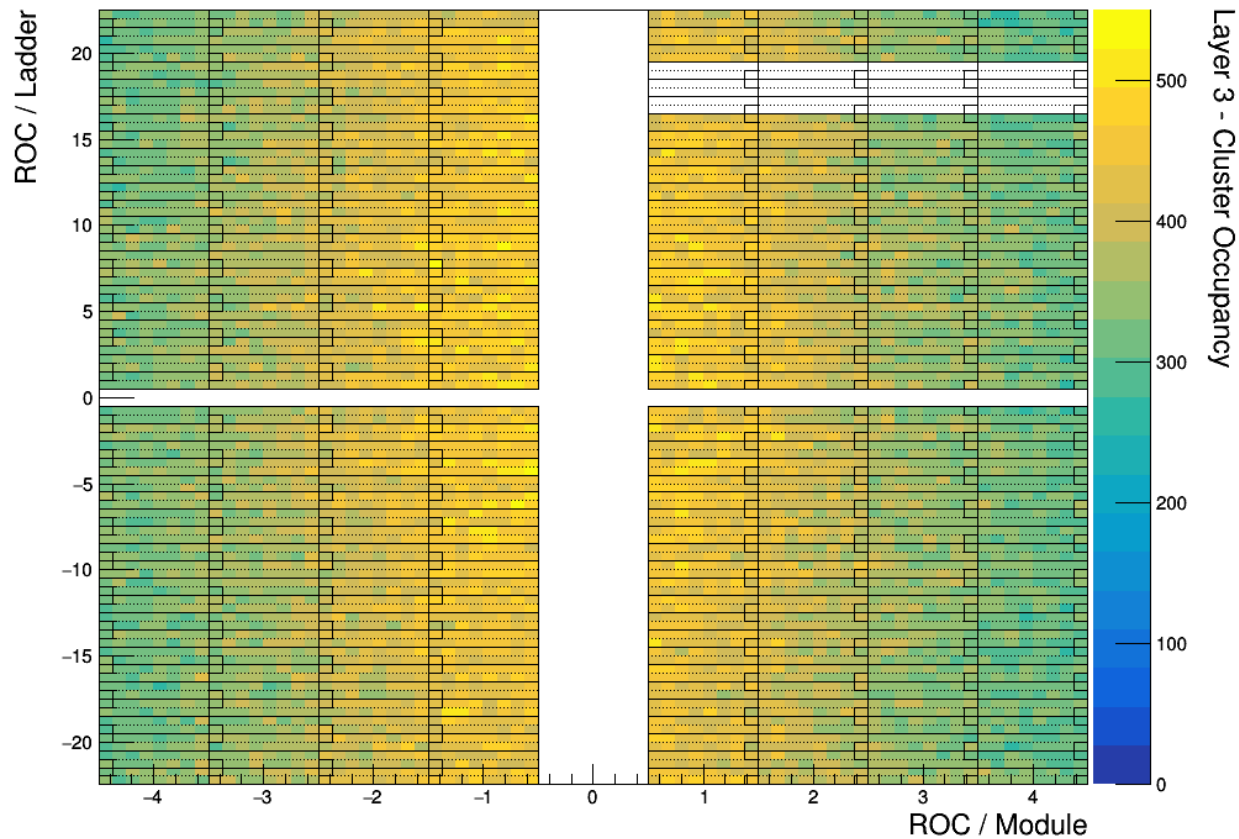
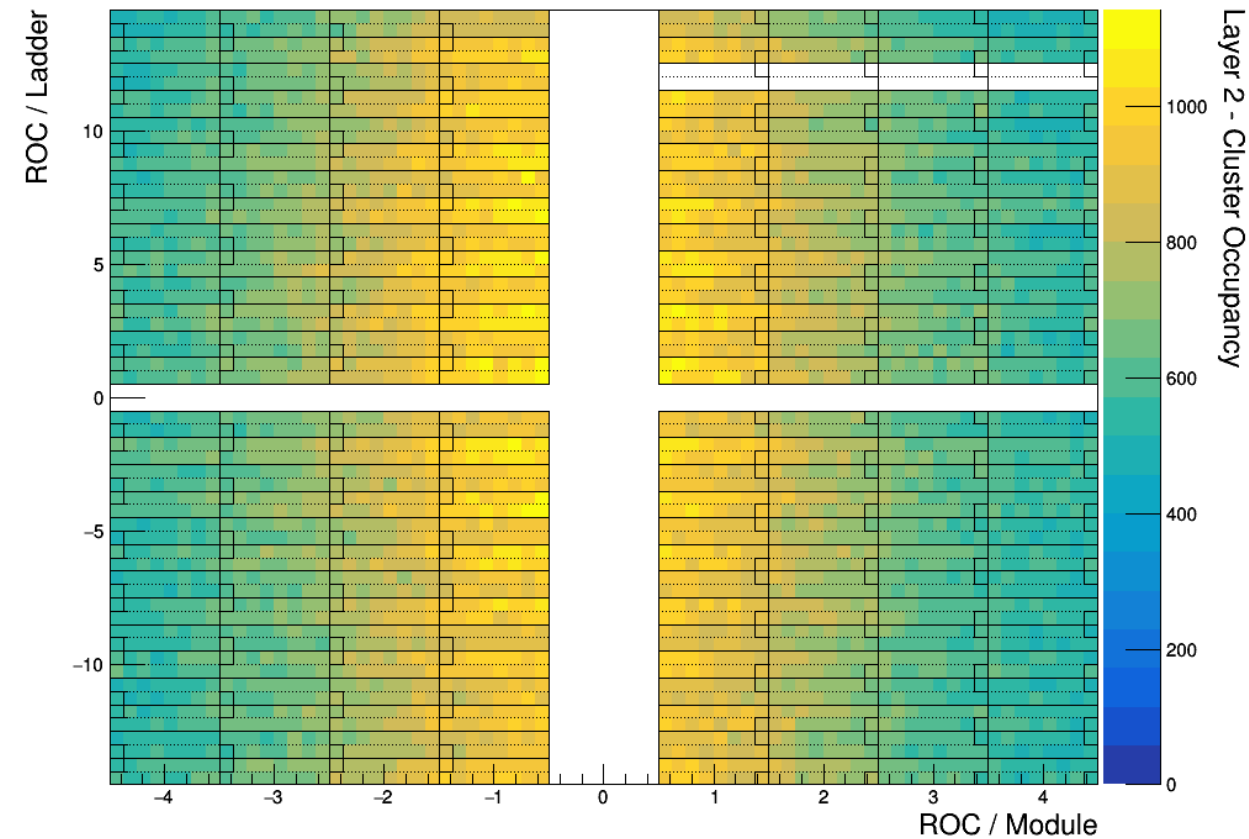
Scenario 3: HV group out

BmO SEC4 L1+L4 out

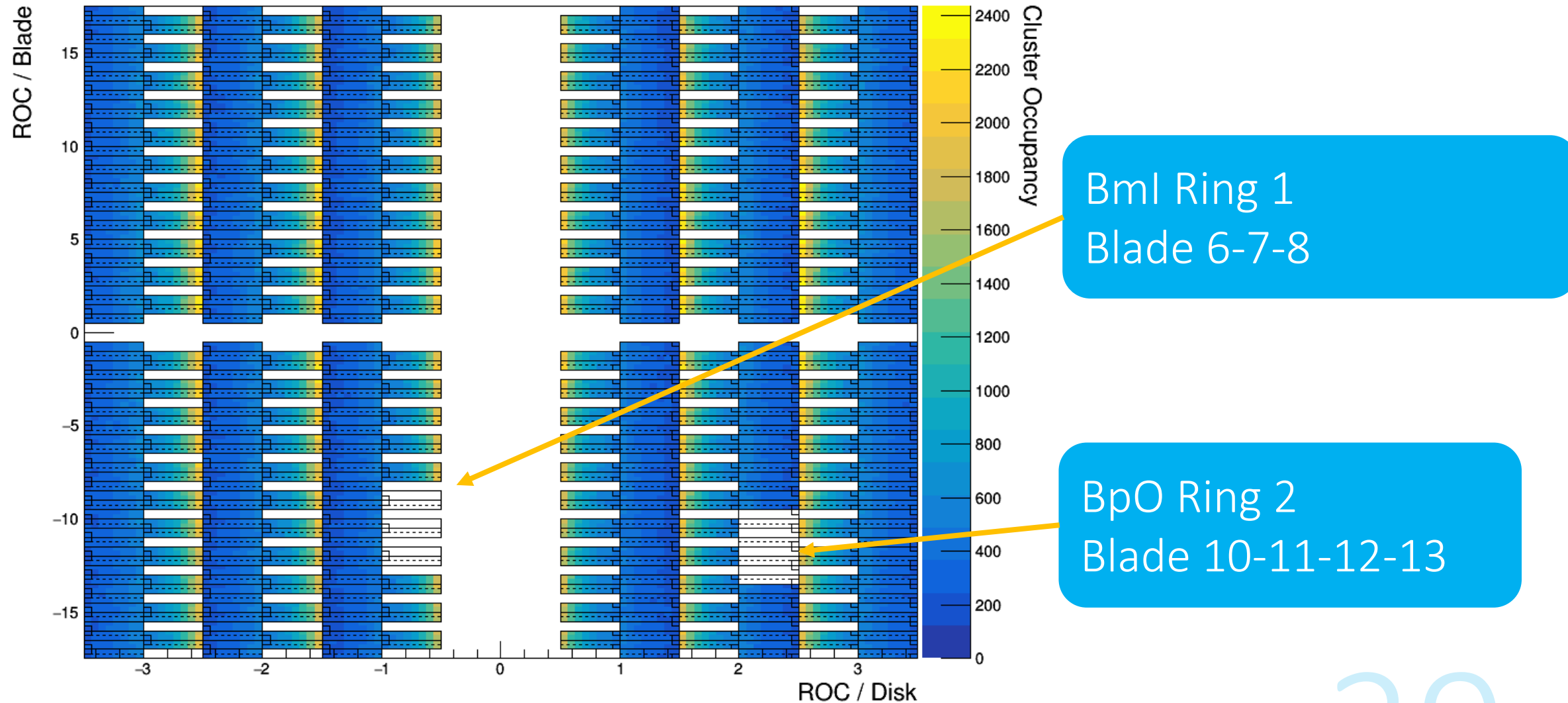


Scenario 3: HV group out

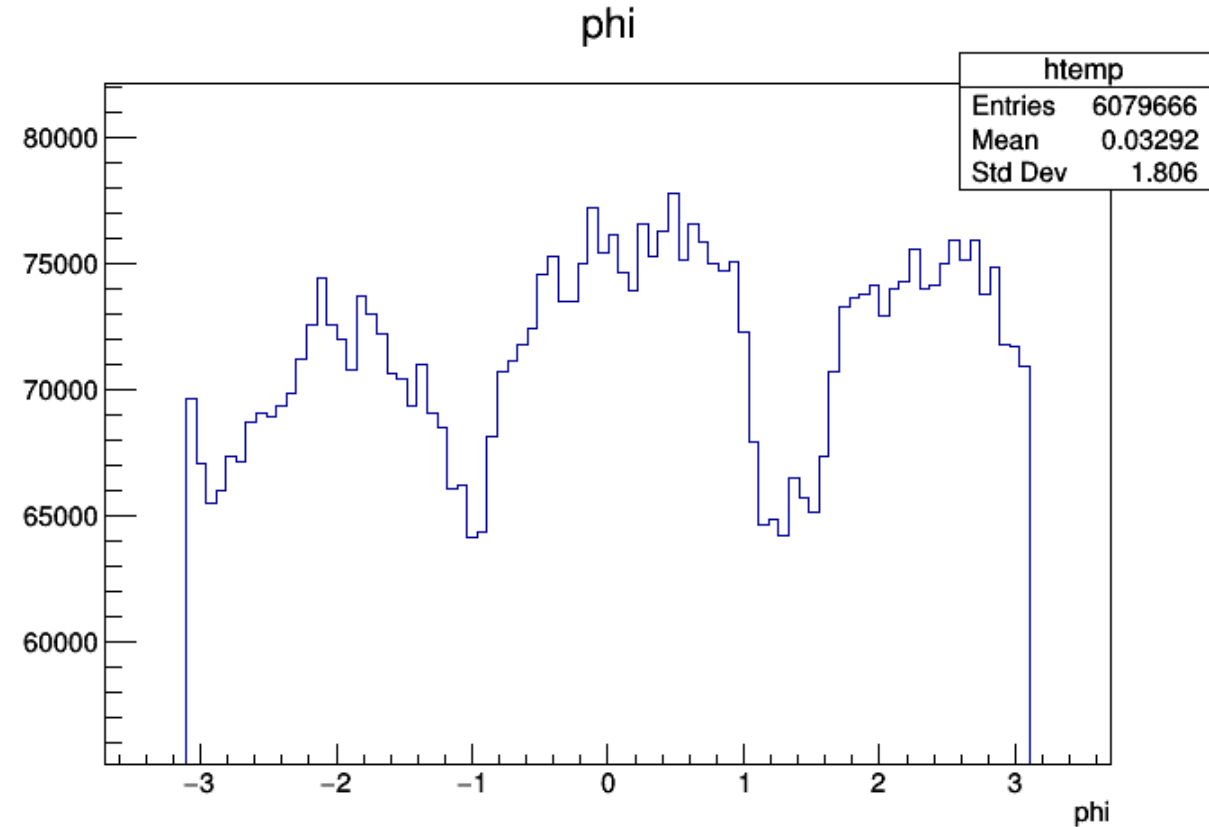
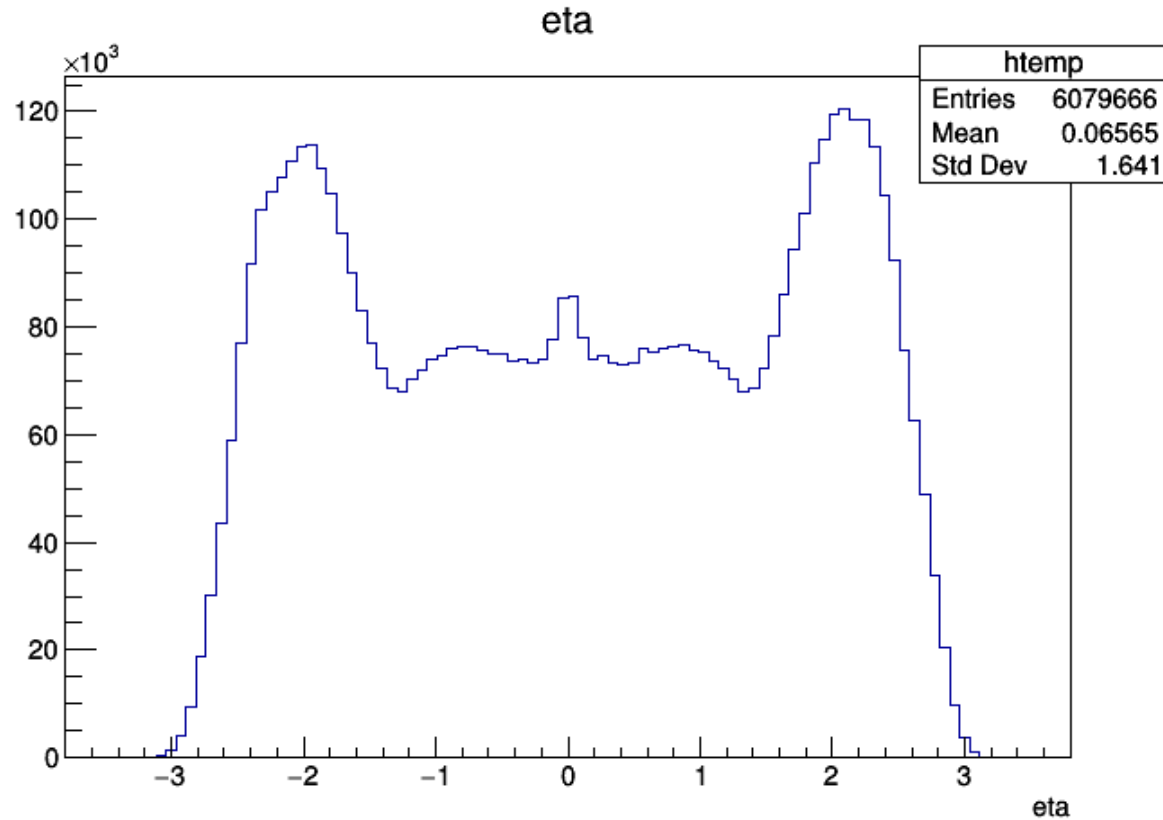
BpI SEC6 L2+L3 out



Scenario 3: HV group out



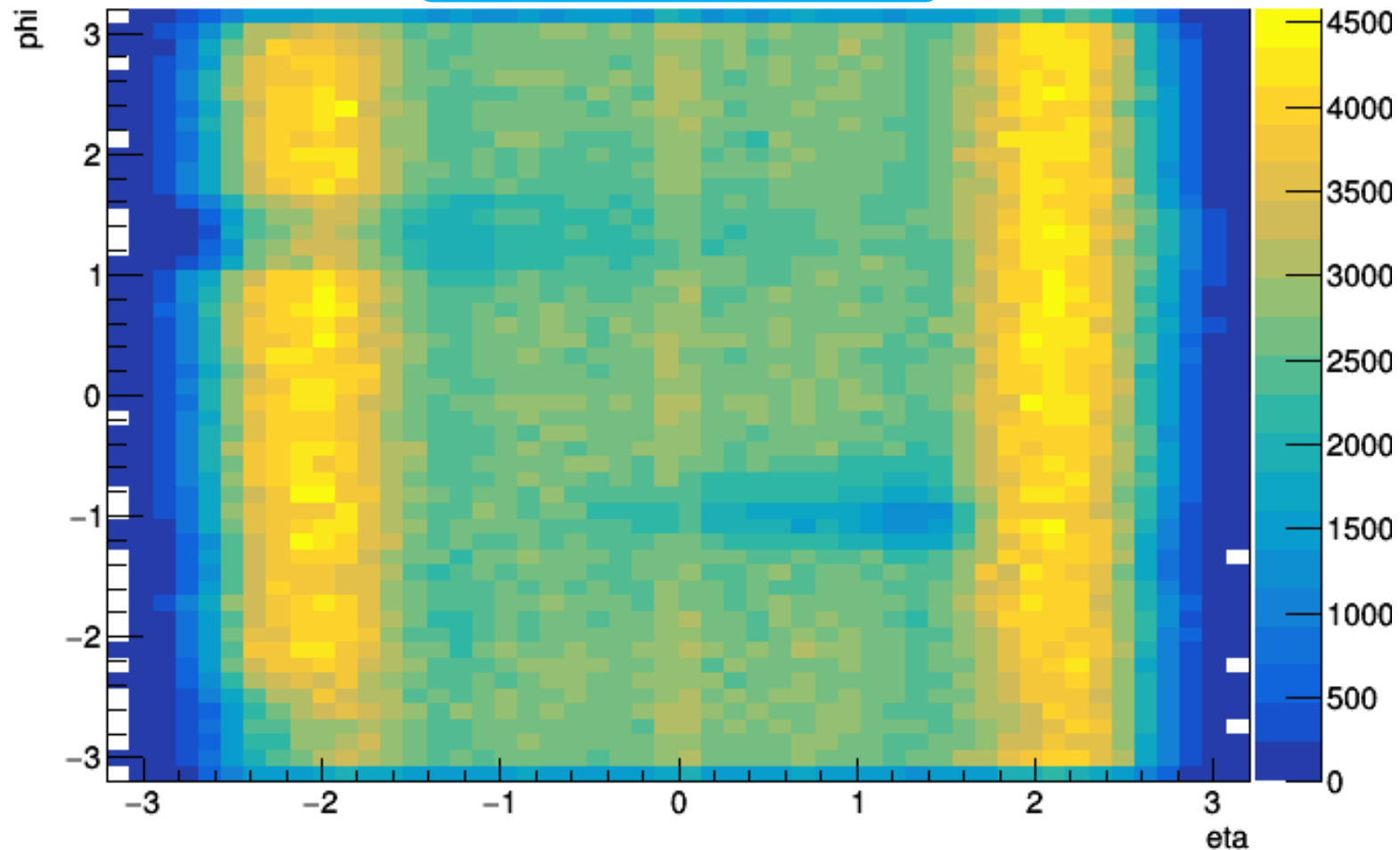
Scenario 3: HV group out



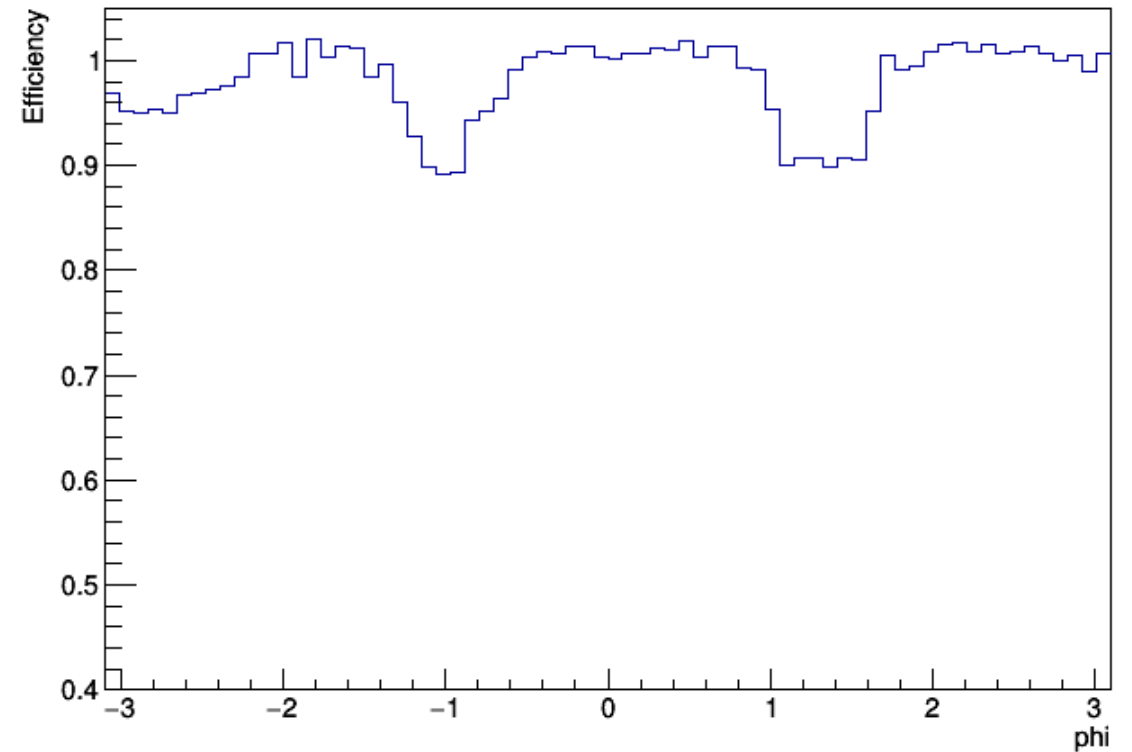
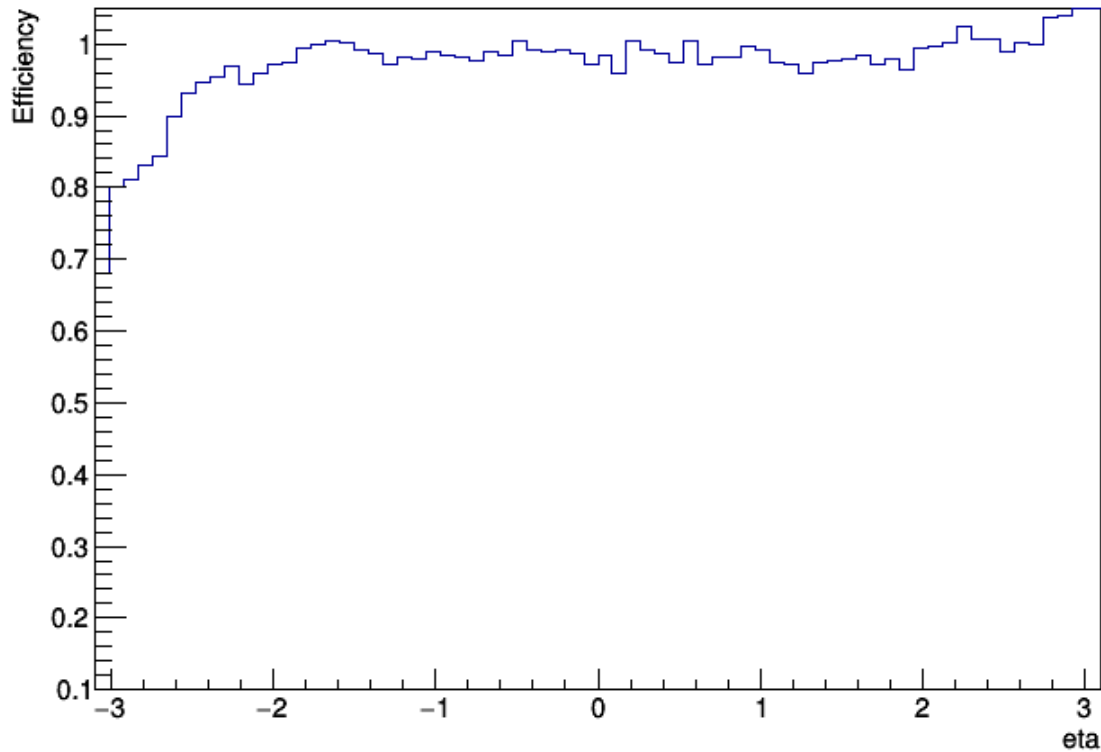
Hit distributions

Scenario 3: HV group out

Hit distribution



Scenario 3: HV group out



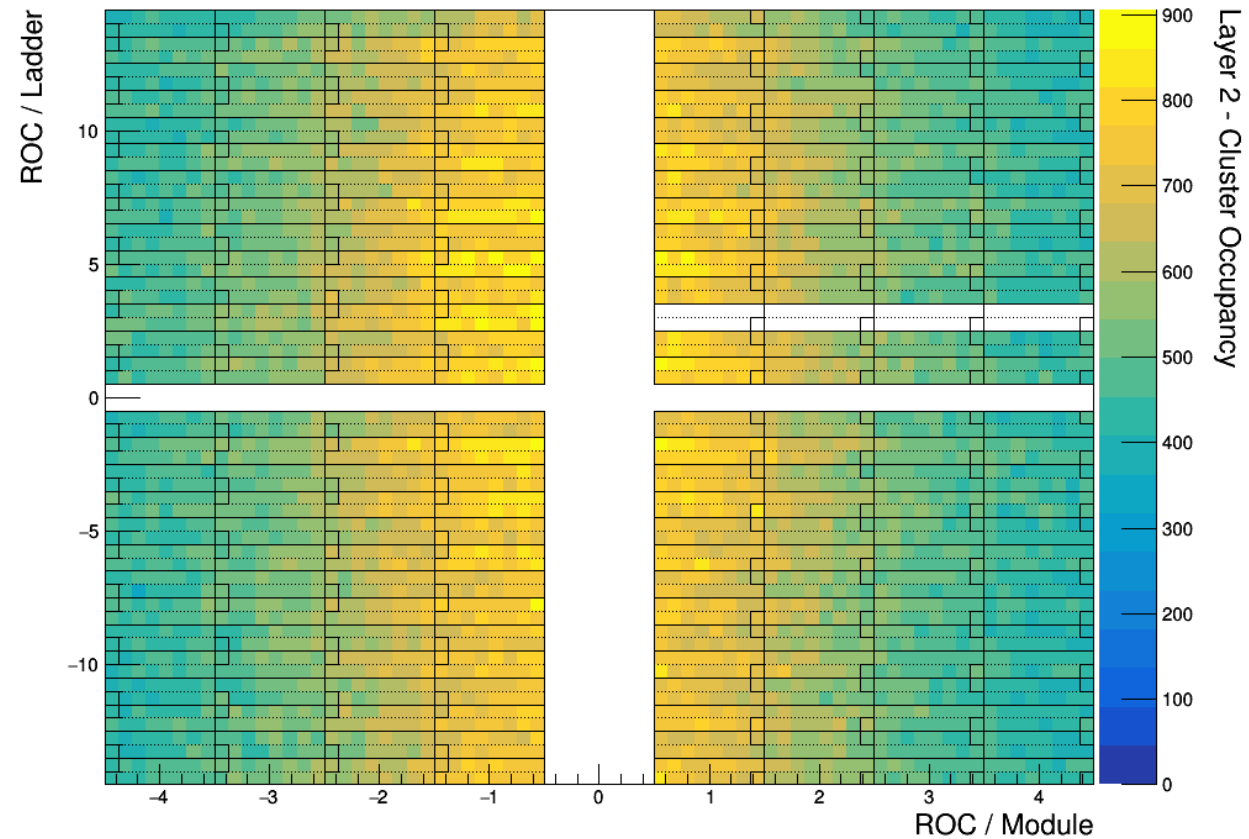
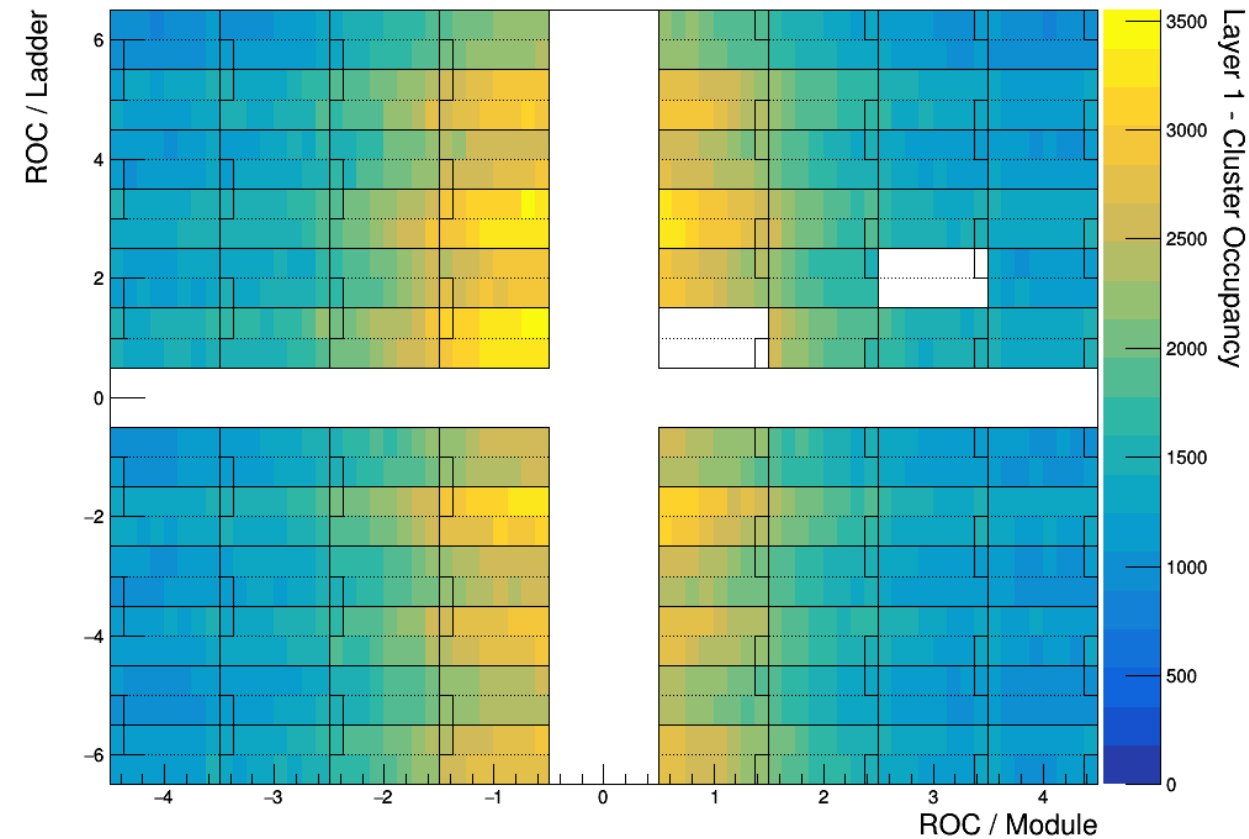
Tracking efficiencies

Scenario 4: 1 BPix + 1 FPix FED out

BPix: From Danek: [SiPixelFedCablingMap_phase1_v6](#)
FED 1203 is chosen (on multiple layers by design)

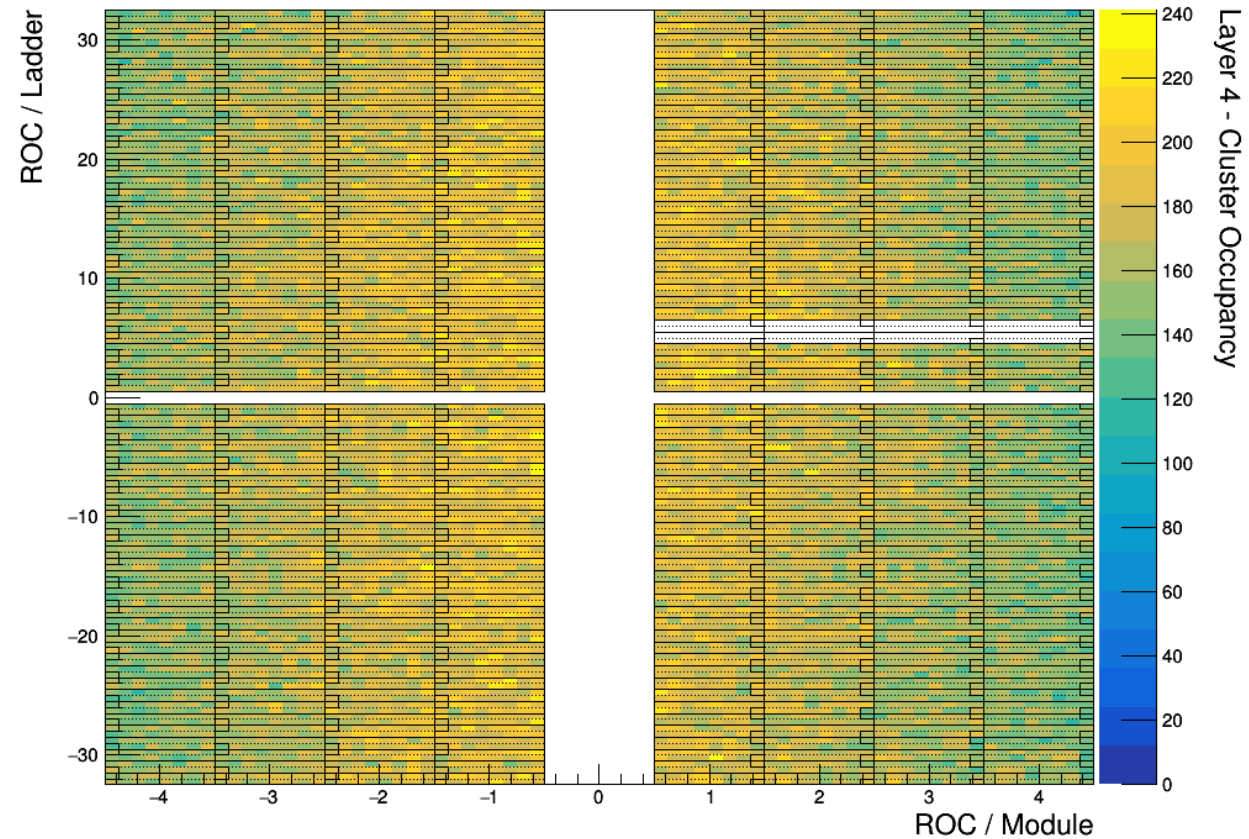
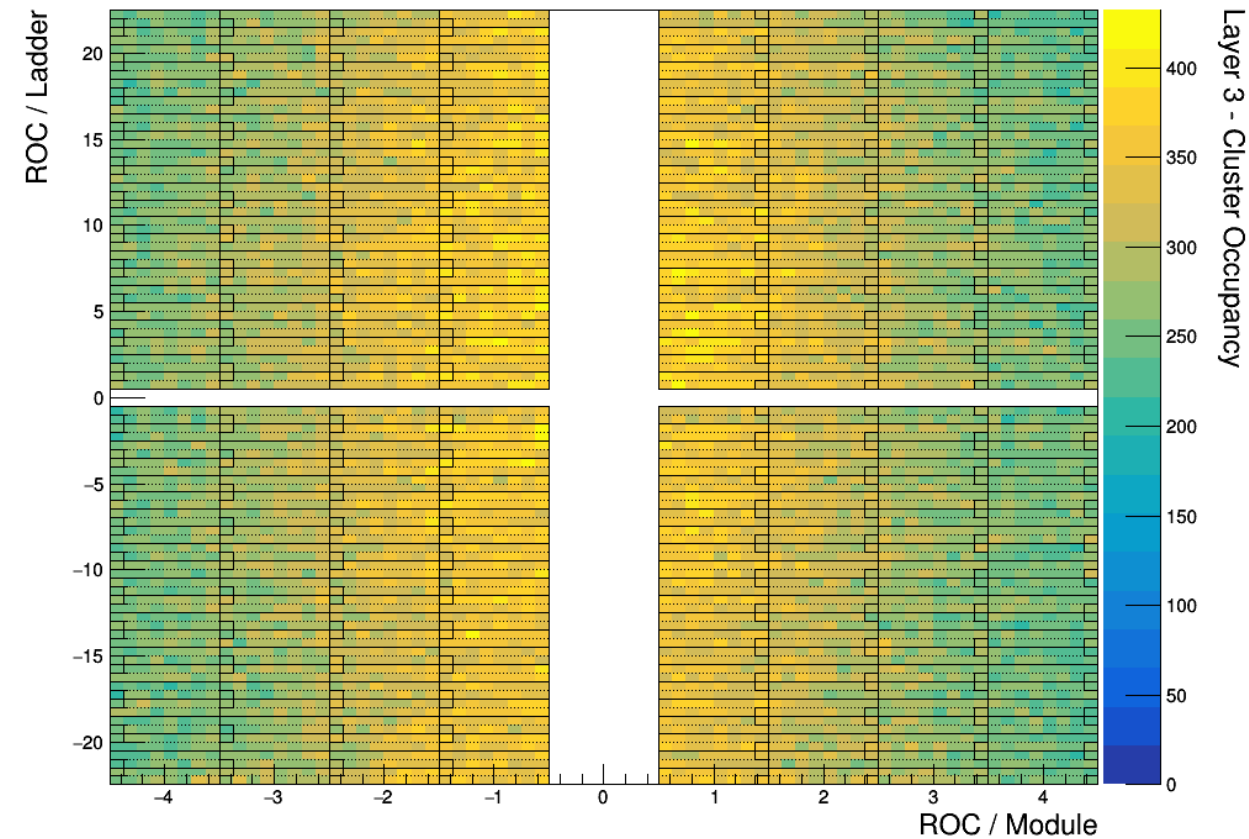
FPix: From Danek: SiPixelFedCablingMap_phase1_v6.db
FED 1333 is chosen

Scenario 4: 1 BPix + 1 FPix FED out



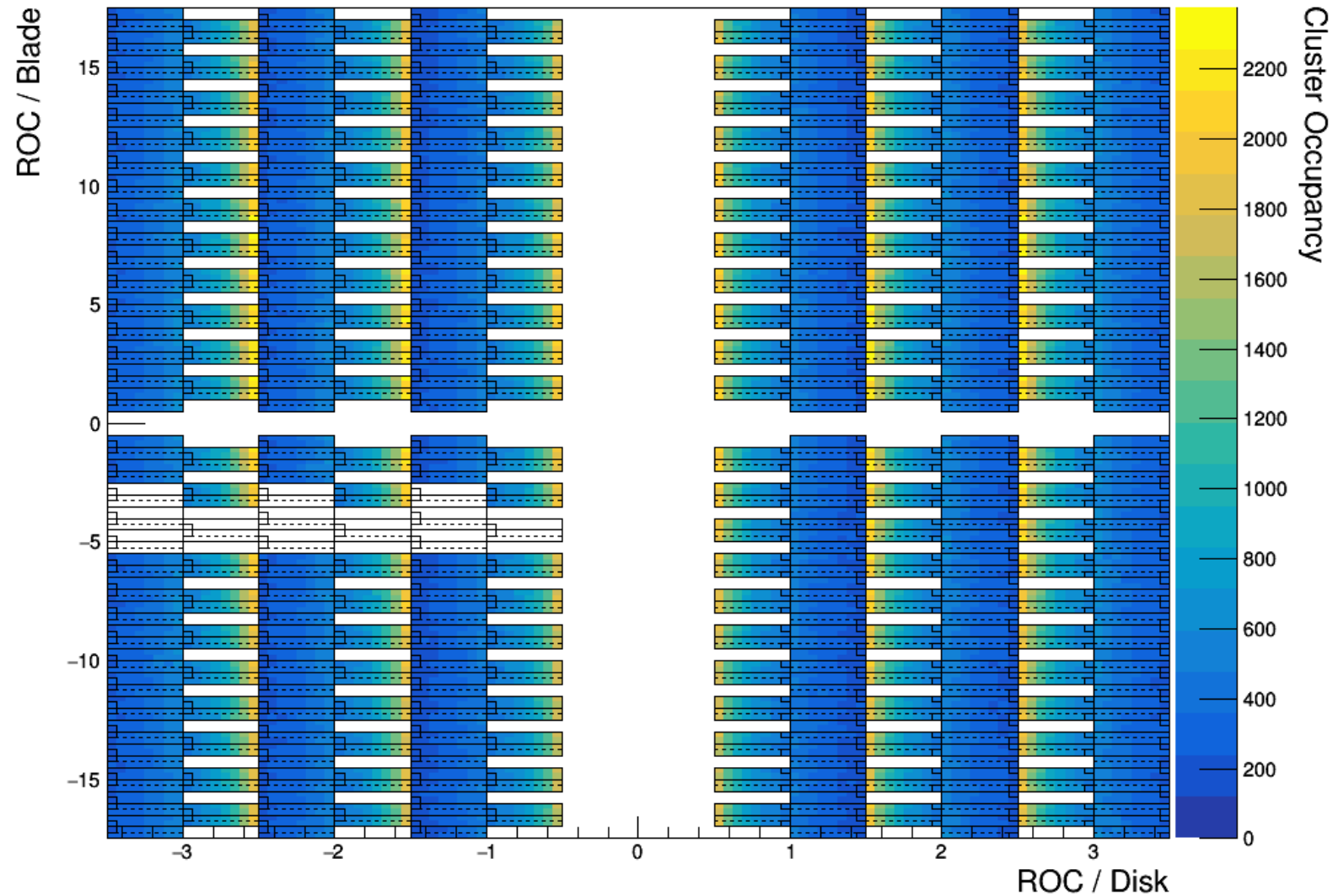
FED 1203 is out

Scenario 4: 1 BPix + 1 FPix FED out



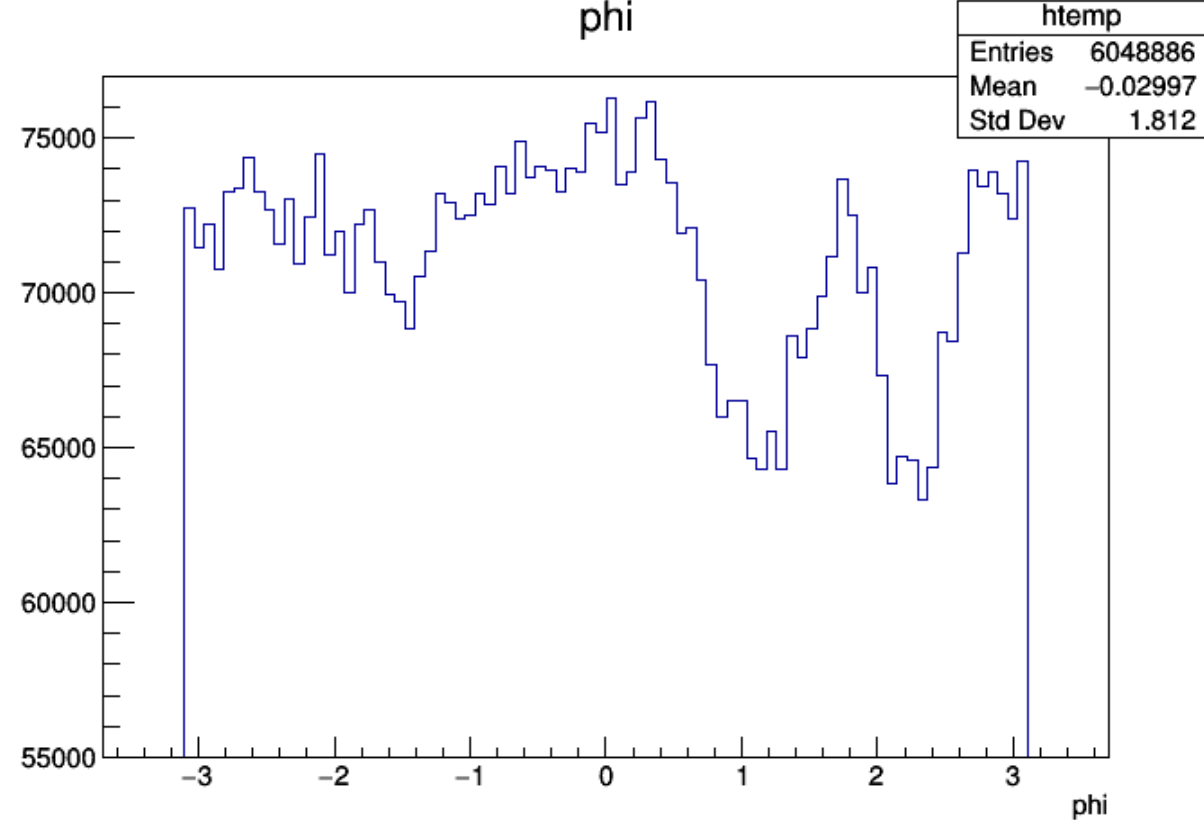
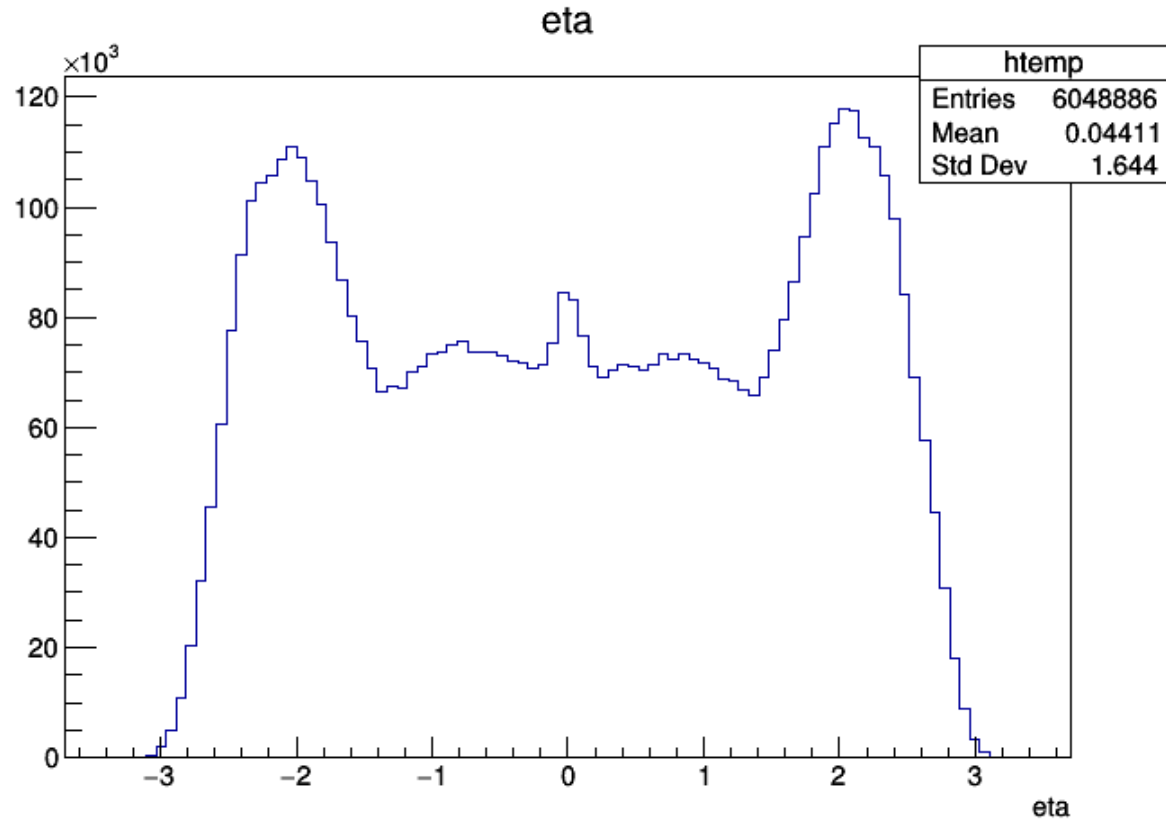
FED 1203 is out

Scenario 4: 1 BPix + 1 FPix FED out



FED 1333 is out

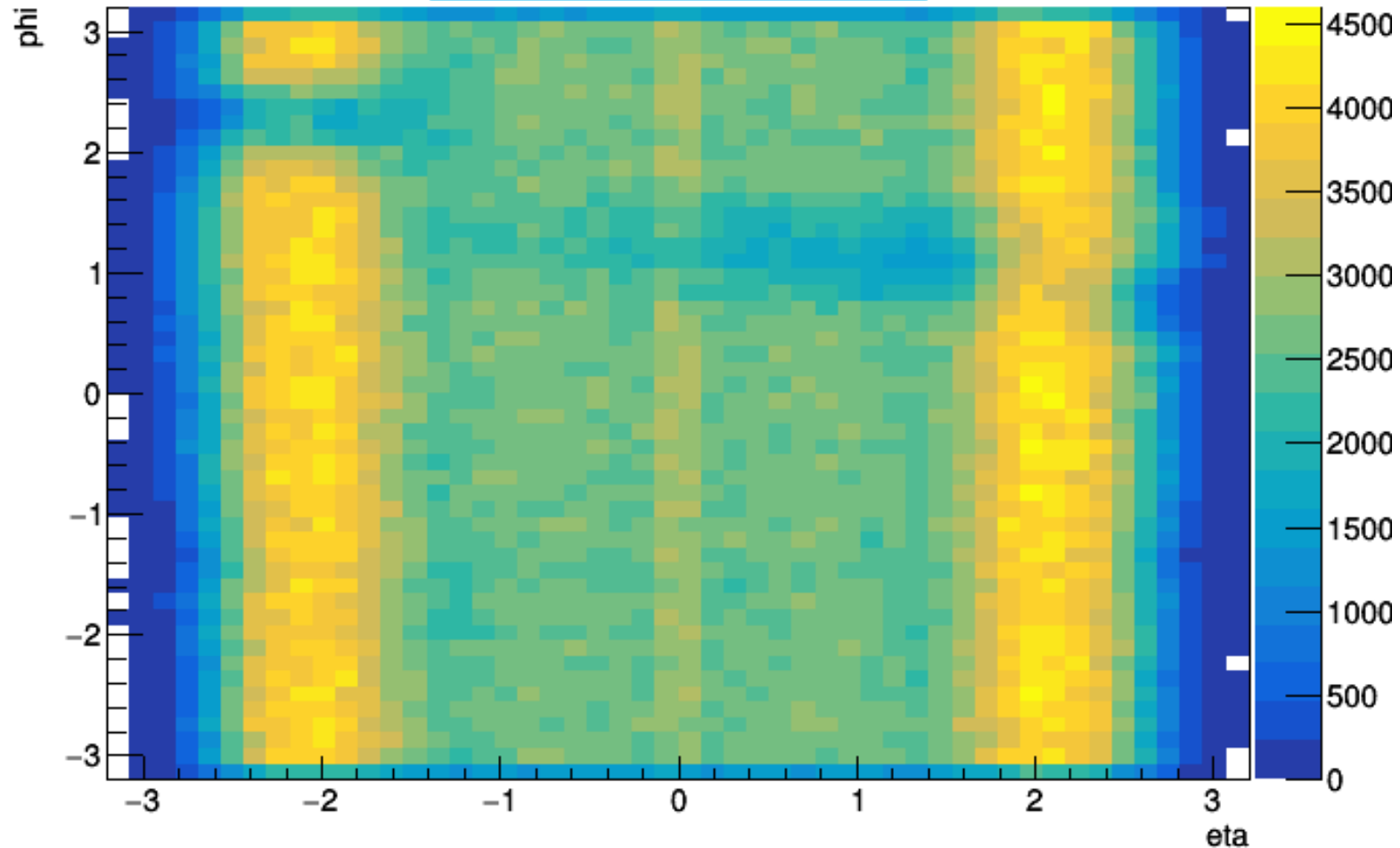
Scenario 4: 1 BPix + 1 FPix FED out



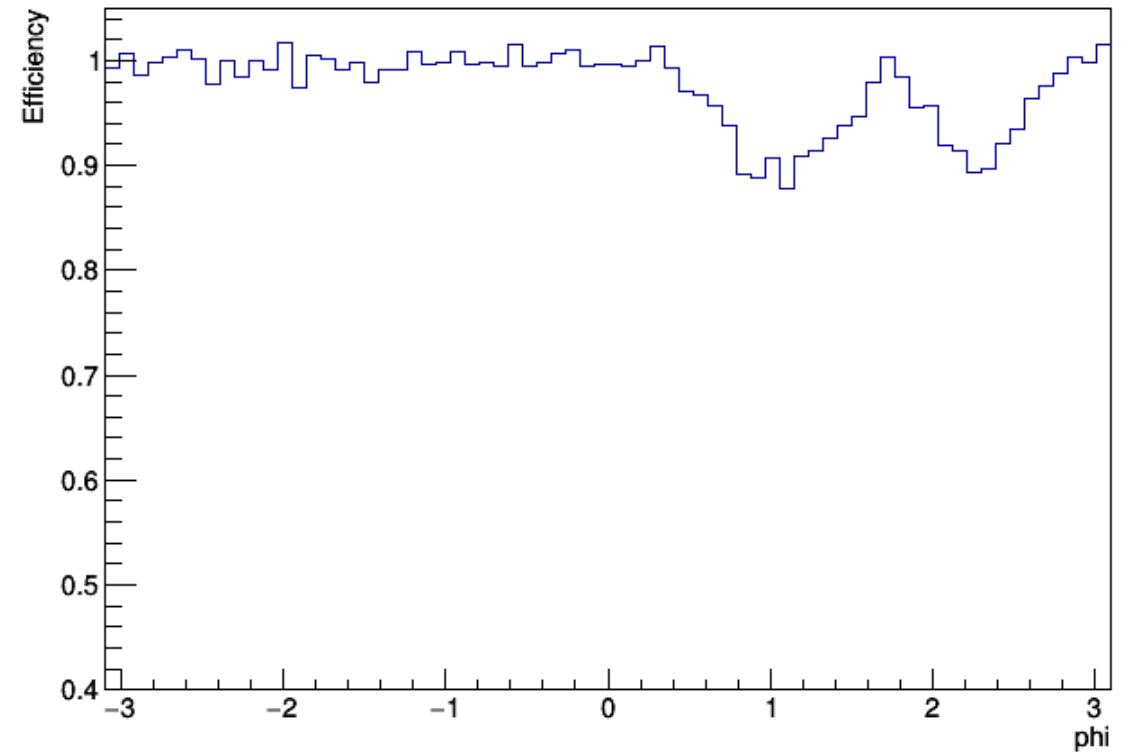
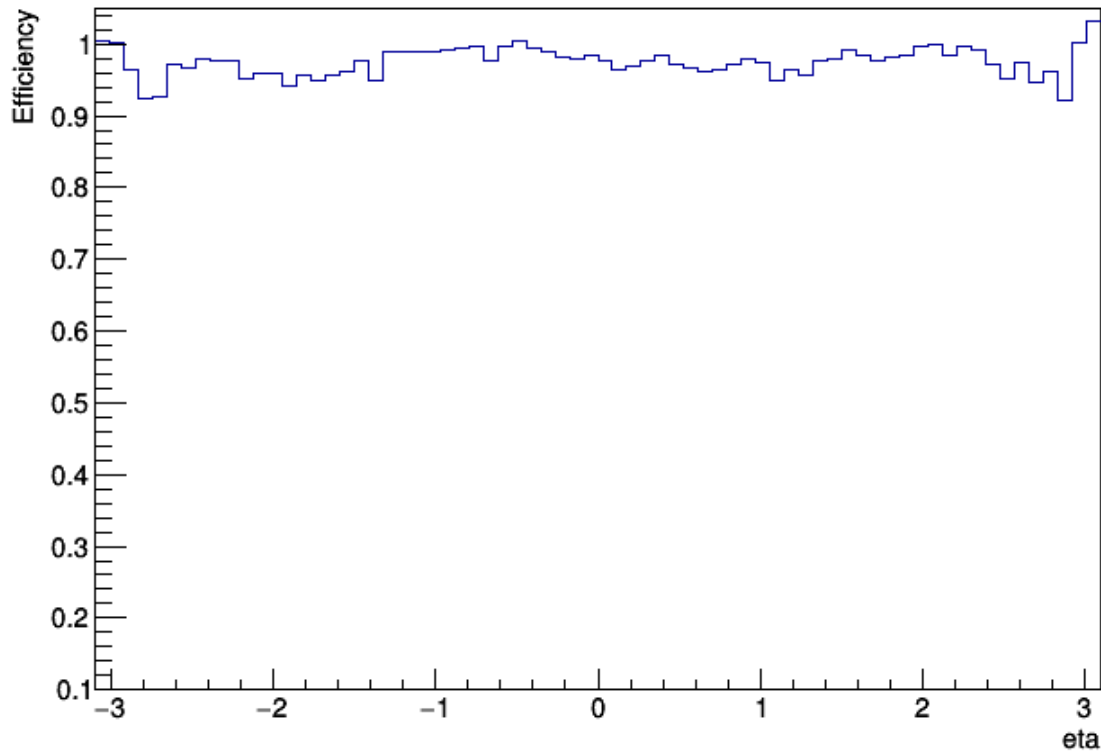
Hit distributions

Scenario 4: 1 BPix + 1 FPix FED out

Hit distribution



Scenario 4: 1 BPix + 1 FPix FED out

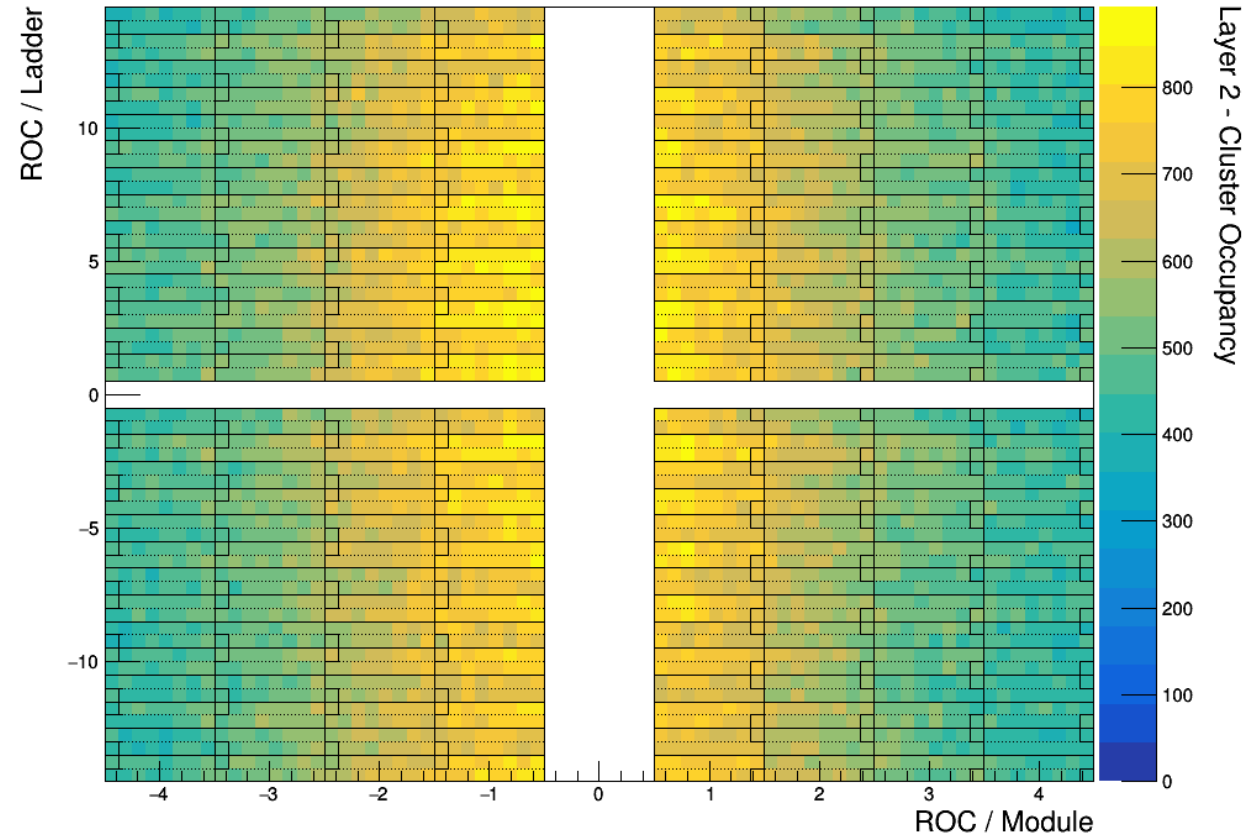
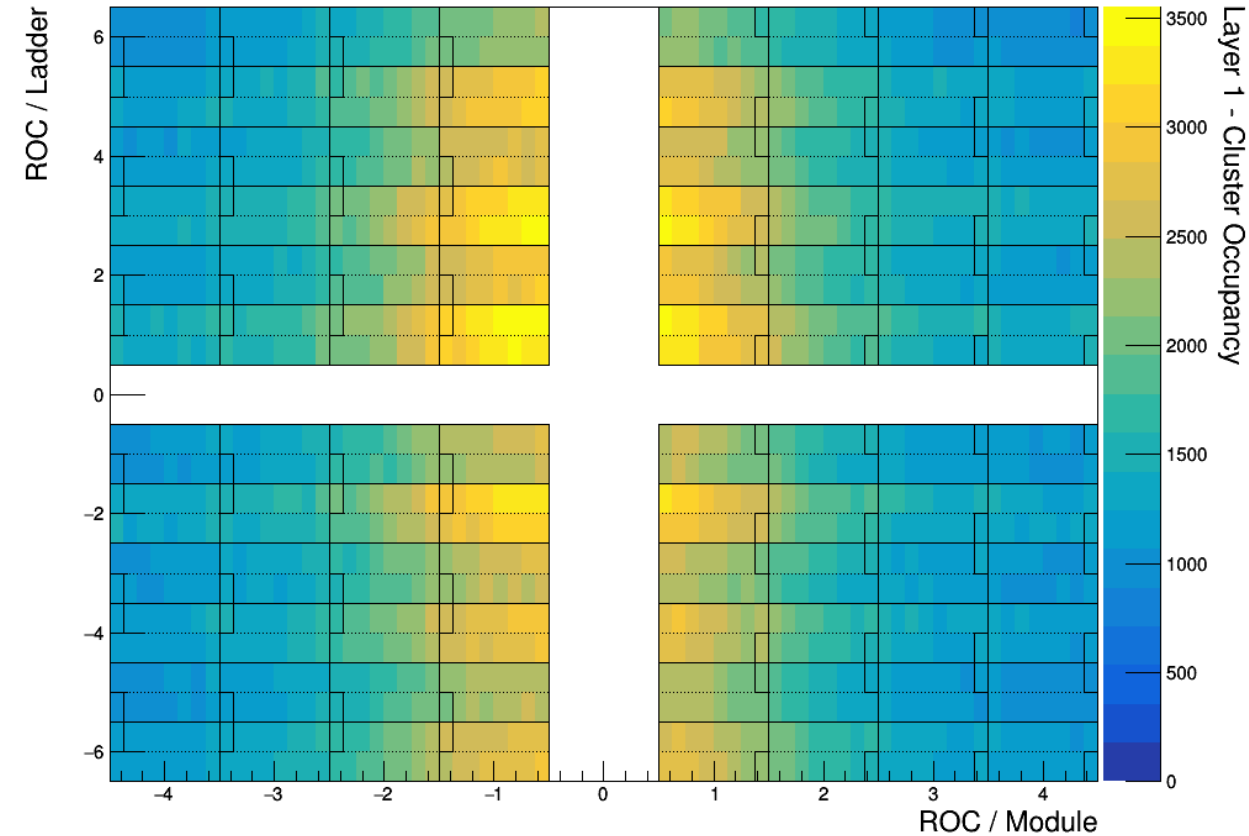


Tracking efficiencies

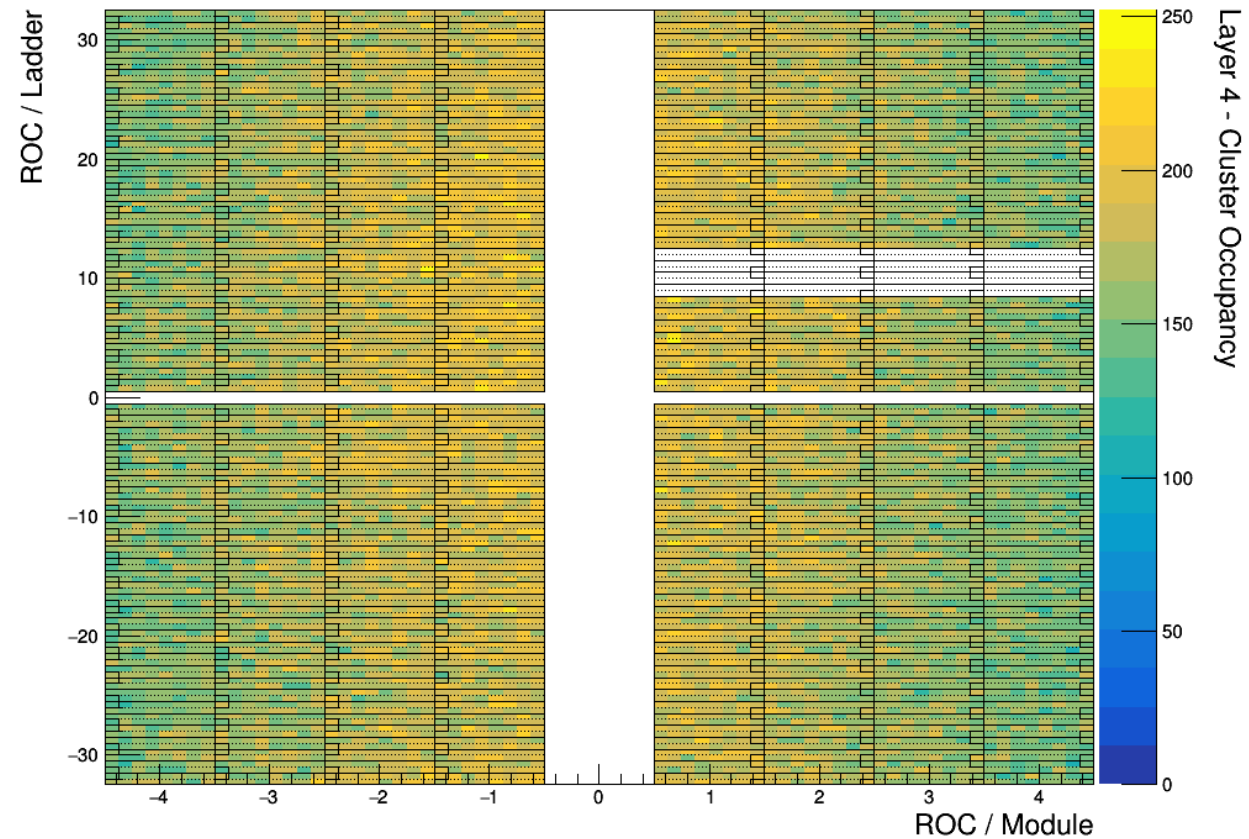
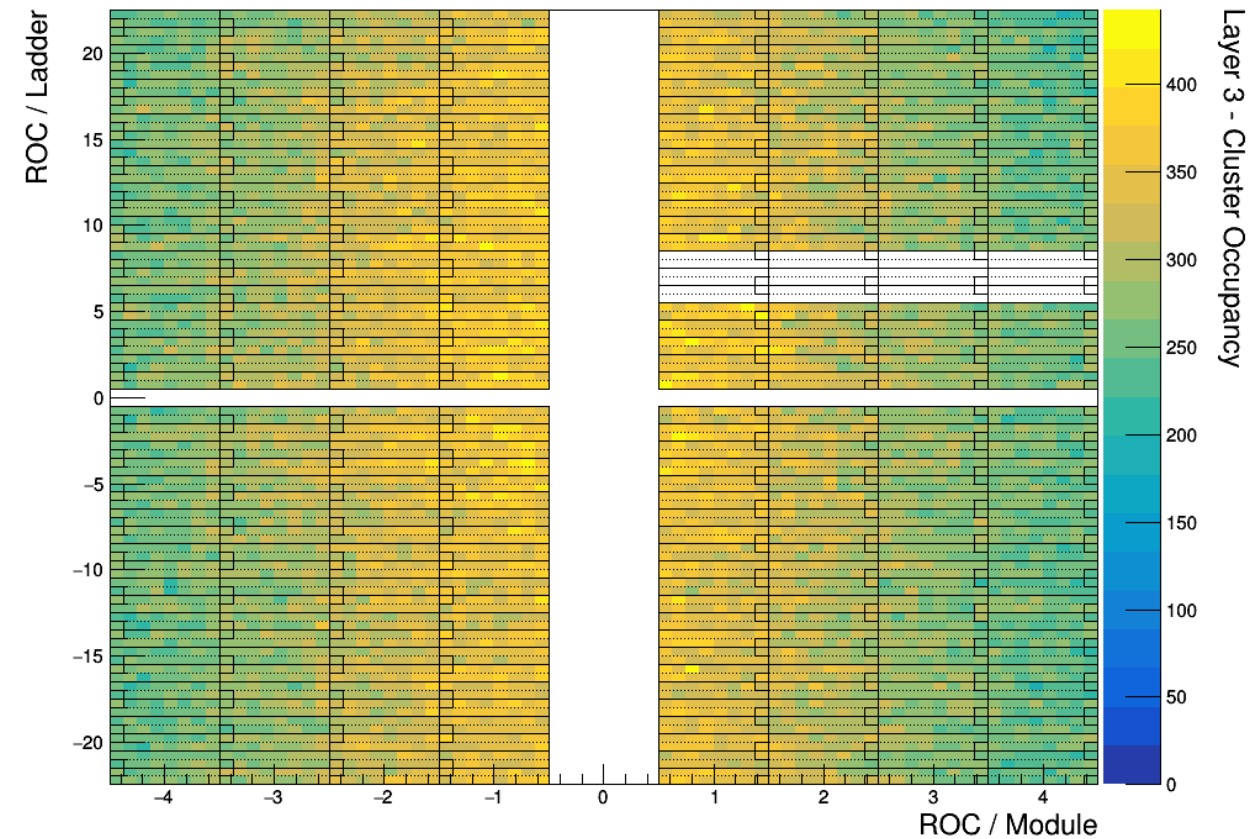
Scenario 5: Read-Out group out

BPix: From Lea: [Phase1 BPIX modules per Sector.xlsx](#)
I choose Bpl SEC3 L3+L4 out

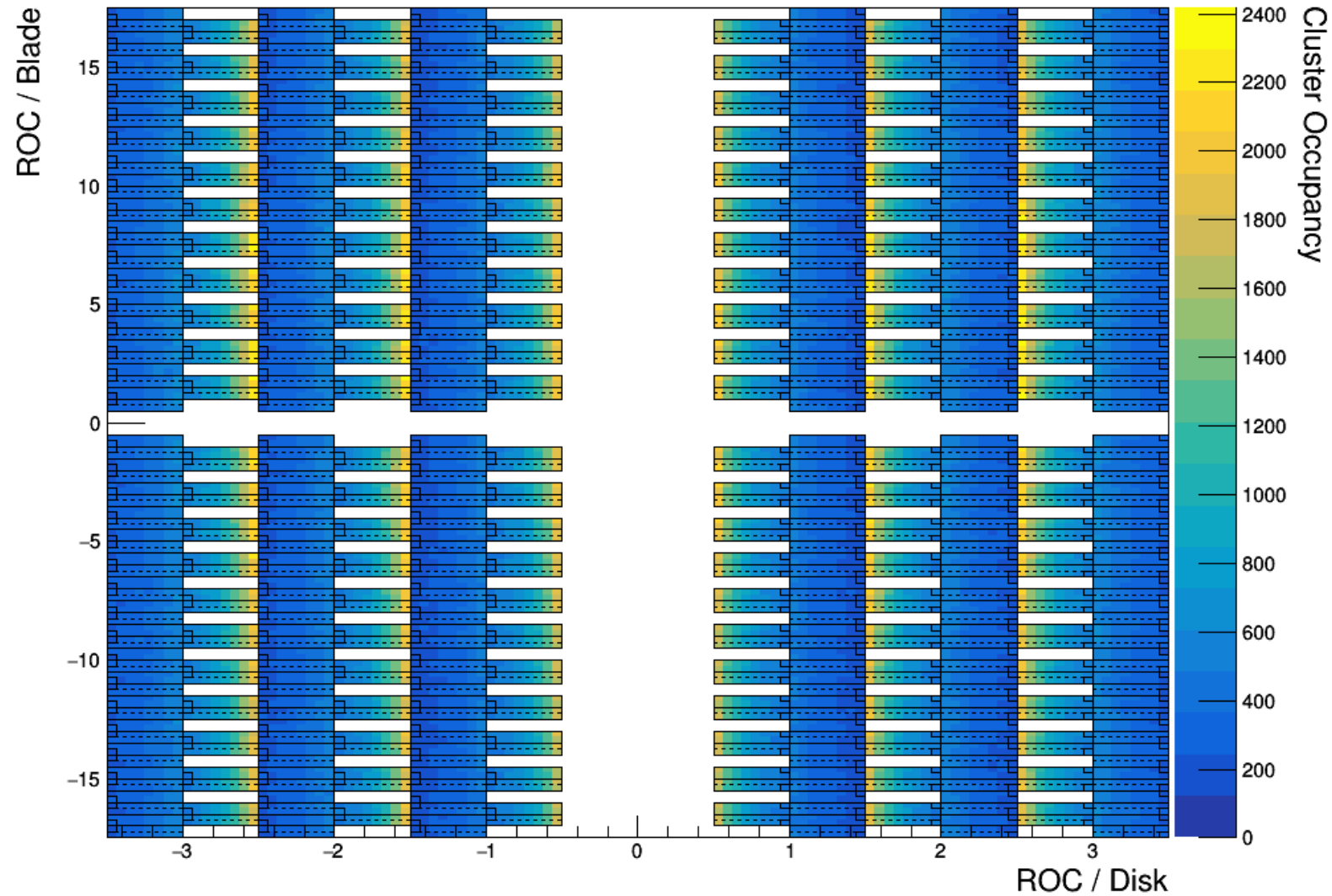
Scenario 5: Read-Out group out



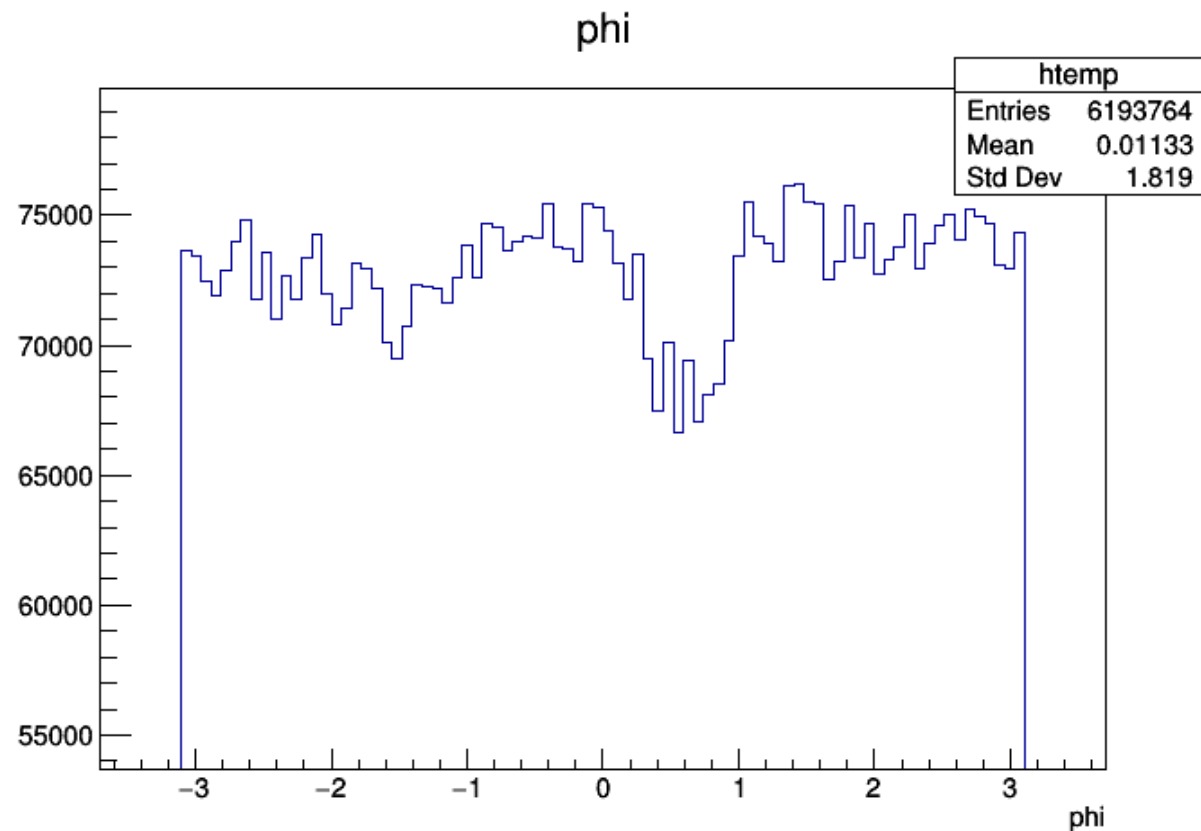
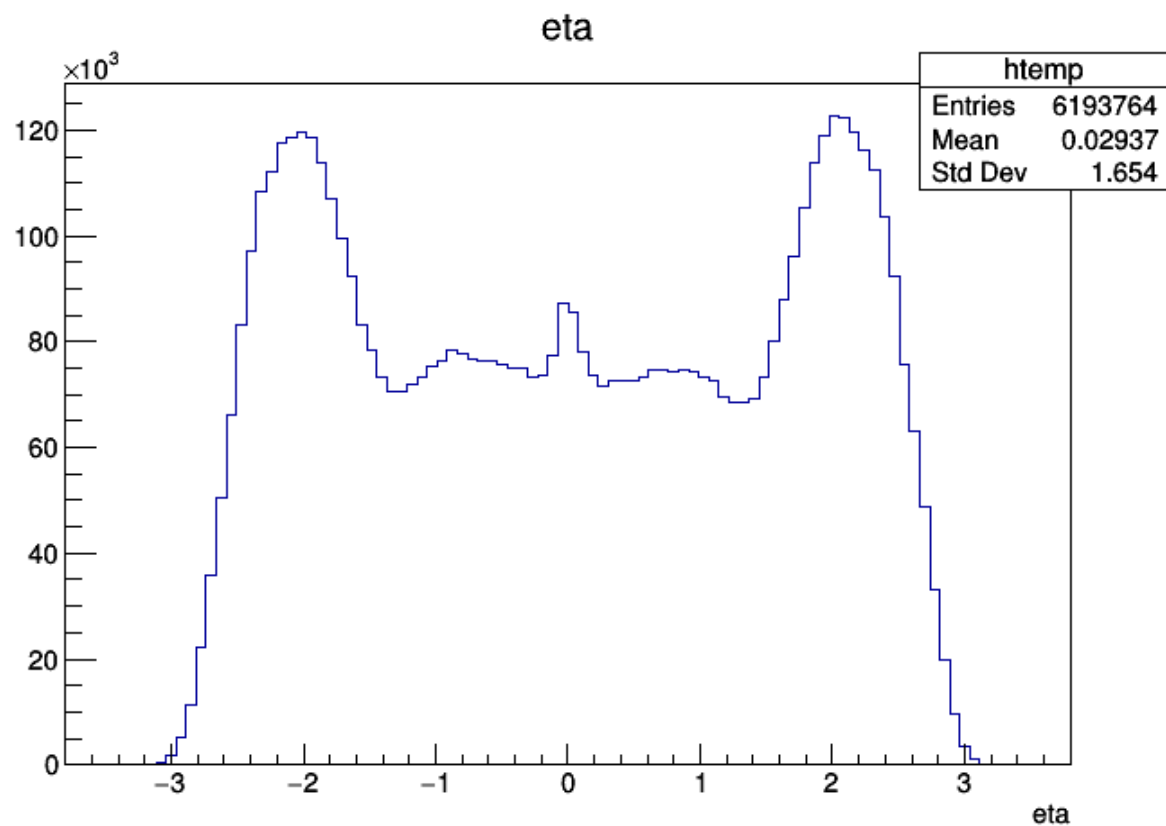
Scenario 5: Read-Out group out



Scenario 5: Read-Out group out



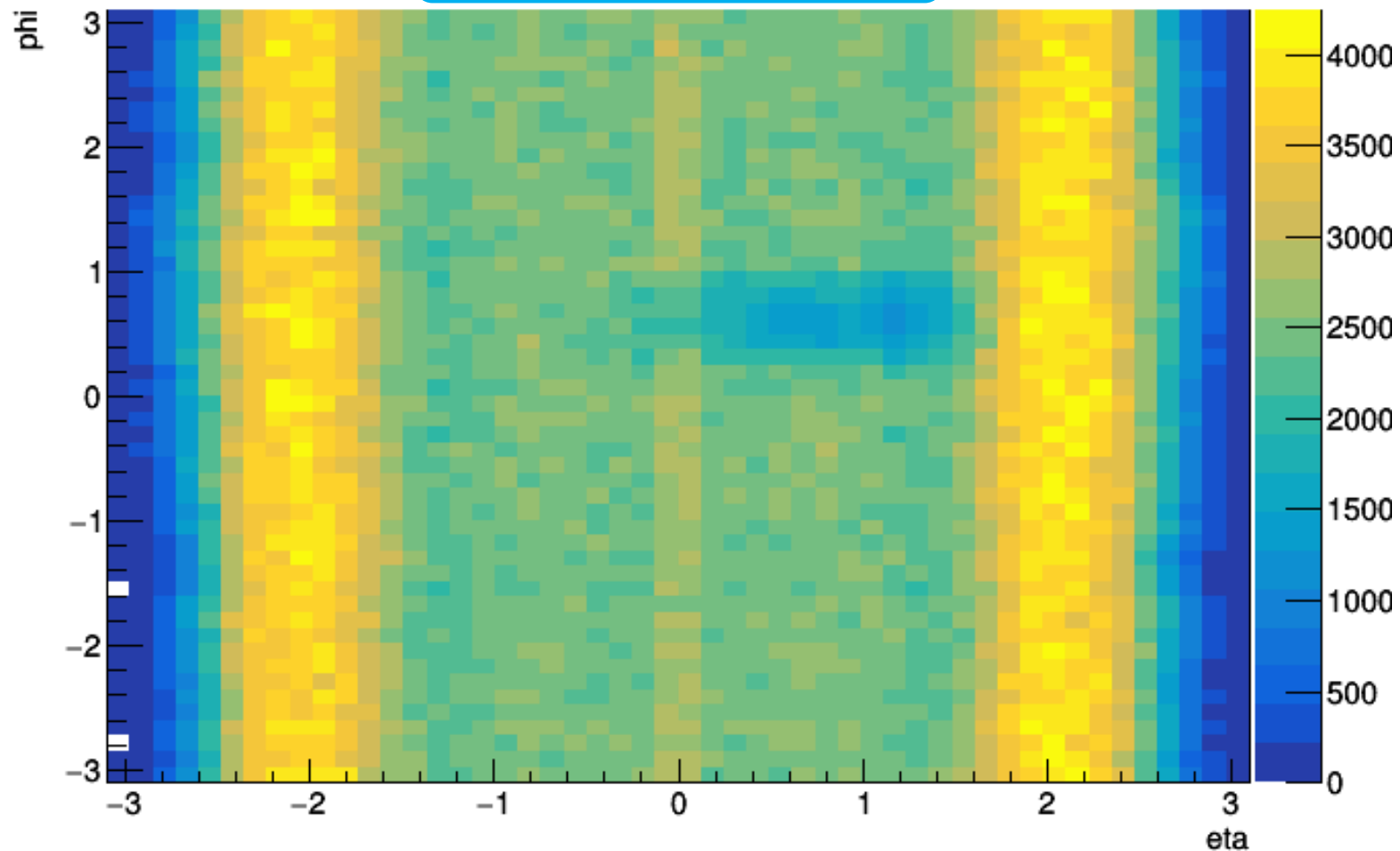
Scenario 5: Read-Out group out



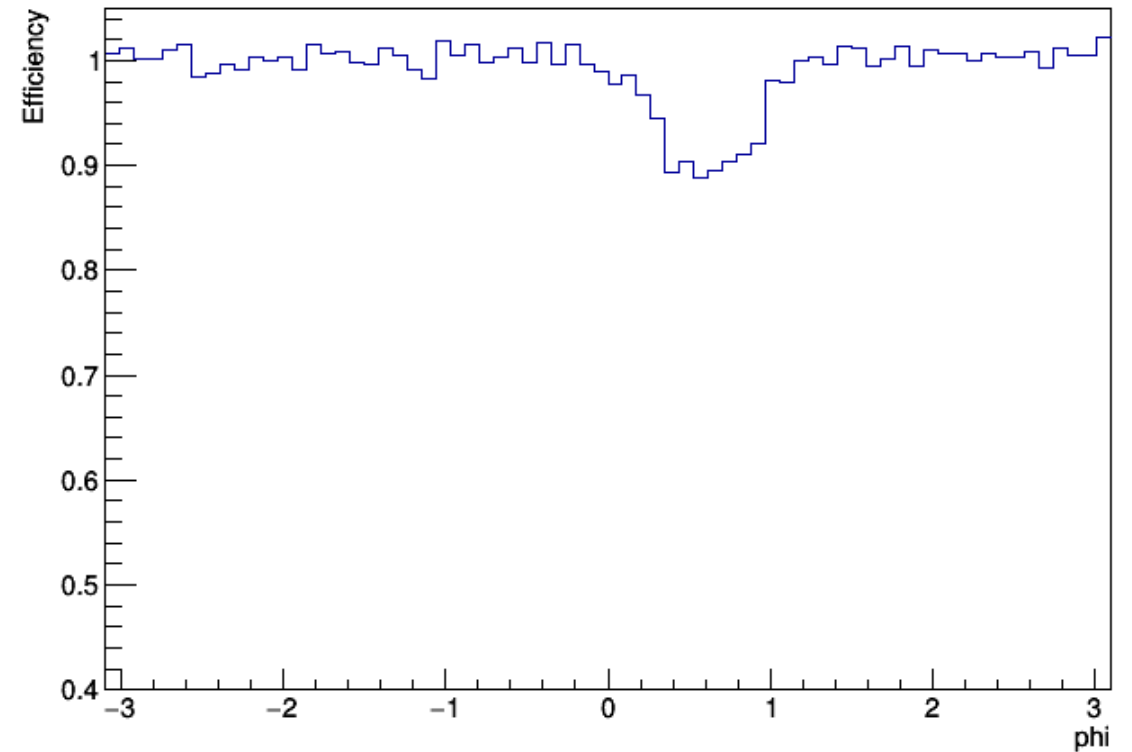
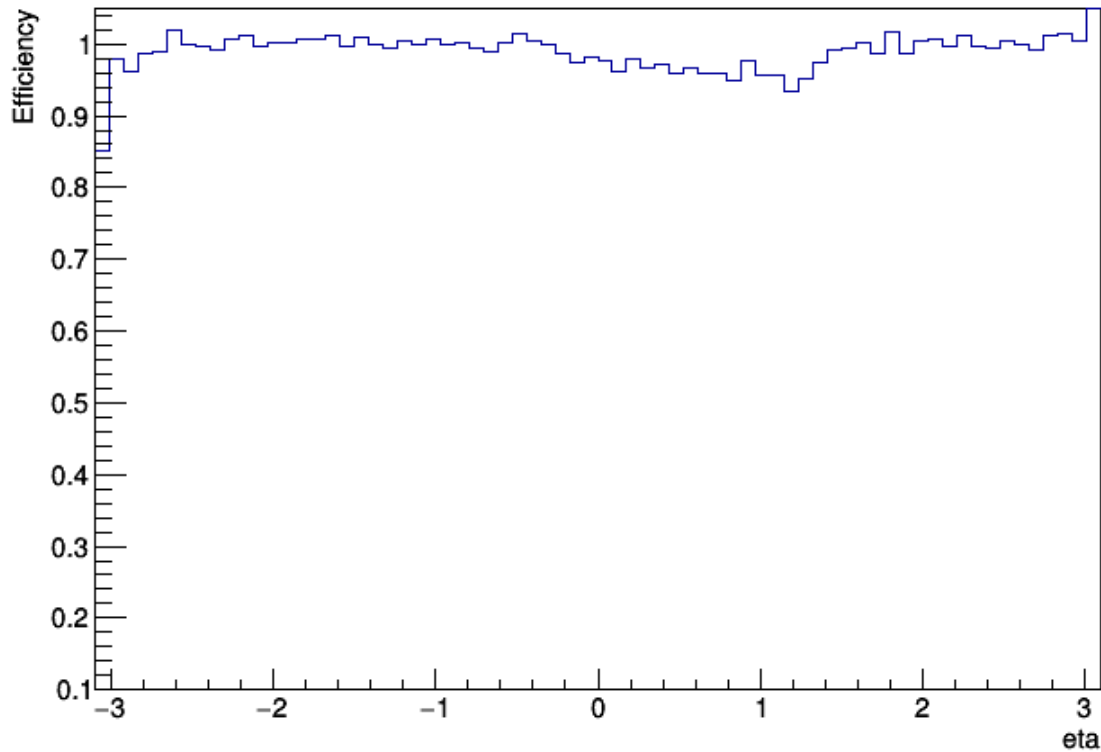
Hit distributions

Scenario 5: Read-Out group out

Hit distribution



Scenario 5: Read-Out group out



Tracking efficiencies

How to reach the payloads

```
from CondCore.CondDB.CondDB_cfi import *
process.SiPixelQualityDBReader = cms.ESSource("PoolDBESSource",
    BlobStreamerName = cms.untracked.string('TBufferBlobStreamingService'),
    DBParameters = cms.PSet(
        messageLevel = cms.untracked.int32(0),
        authenticationPath = cms.untracked.string("")
    ),
    connect = cms.string('frontier://FrontierProd/CMS_CONDITIONS'),
    toGet = cms.VPSet(
        cms.PSet(
            record = cms.string('SiPixelQualityFromDbRcd'),
            # uncomment the favored scenario
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v00'), #NoPixelDet
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v1'), #FS1: Random
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v2'), #FS2: CoolingLoop
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v3'), #FS3: HV_out
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v4'), #FS4: FED_out
            #tag = cms.string('SiPixelQuality_Phase1FailureScenario_v5'), #FS5: ReadOut_out
        )
    )
)
process.es_prefer_Quality = cms.ESPrefer("PoolDBESSource","SiPixelQualityDBReader")
```

How to reach the payloads

For the subscenarios please use

...

```
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v2a'), #FS2: BPix CoolingLoop  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v2b'), #FS2: FPix CoolingLoop  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v3a'), #FS3: BmO SEC4 L1+L4 HV_out  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v3b'), #FS3: BpI SEC6 L2+L3 HV_out  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v3c'), #FS3: BmI Disk1 ROG3 HV2 HV_out  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v3d'), #FS3: BpO Disk2 ROG1 HV1 HV_out  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v4a'), #FS4: BPix FED_out  
#tag = cms.string('SiPixelQuality_Phase1FailureScenario_v4b'), #FS4: FPix FED_out
```

...