

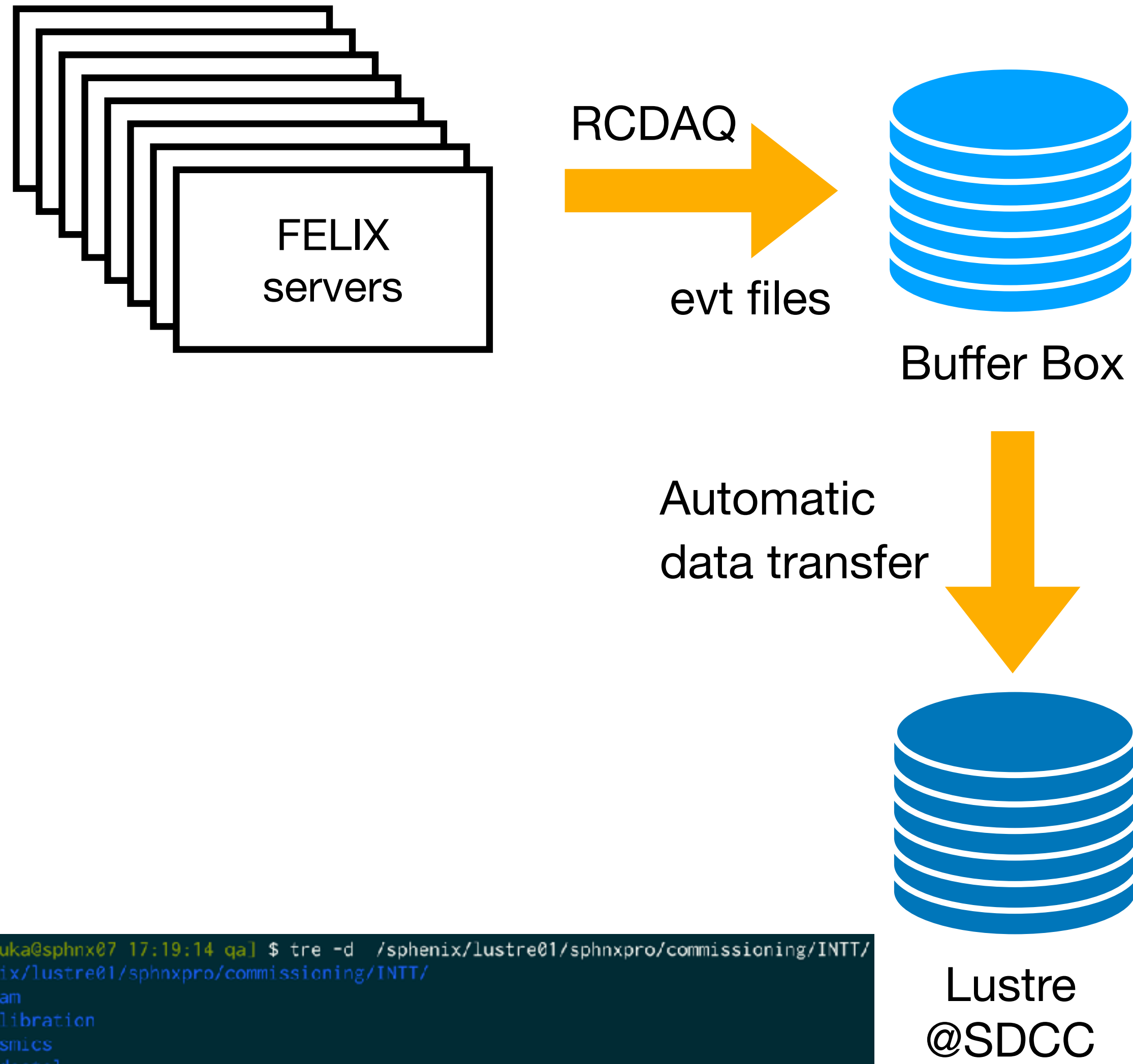
QA plan

G. Nukazuka (RIKEN/RBRC)

Quality Assurance for better data

We should check data once data is taken so we may find some problematic things.

Data Process Procedures



```
[nukazuka@sphnx07 17:19:14 qa] $ tre -d /sphenix/lustre01/sphnxpro/commissioning/INTT/  
/sphenix/lustre01/sphnxpro/commissioning/INTT/  
├── bean  
├── calibration  
├── cosmics  
├── pedestal  
├── tmp  
├── calib  
│   └── bkp  
└── from_INTTDAQ
```

A. Making a hit-base TTree using a evt file
Well-established way. The output is portable.

A single orange document icon representing a 'single evt file' is shown on the left. A large orange arrow points to the right, where a green tree icon represents a 'hit-base TTree'.

B. Making DST(s) using evt files
It became available thanks to the new decoder.
It works well for cosmic data but not for pedestal data.

Two rows of four orange document icons each represent 'some evt files'. A large orange arrow points to the right, where a green open book icon represents a 'DST'.

C. Same as B but letting sPHENIX do it automatically (official production)
It's ideal way. Some problems were found. It's too early to rely on. The production timing is out of our control.

Two rows of four orange document icons each represent 'some evt files'. A large orange arrow points to the right, where a green open book icon with the word 'OFFICIAL' written vertically on its spine represents an 'official DST'.

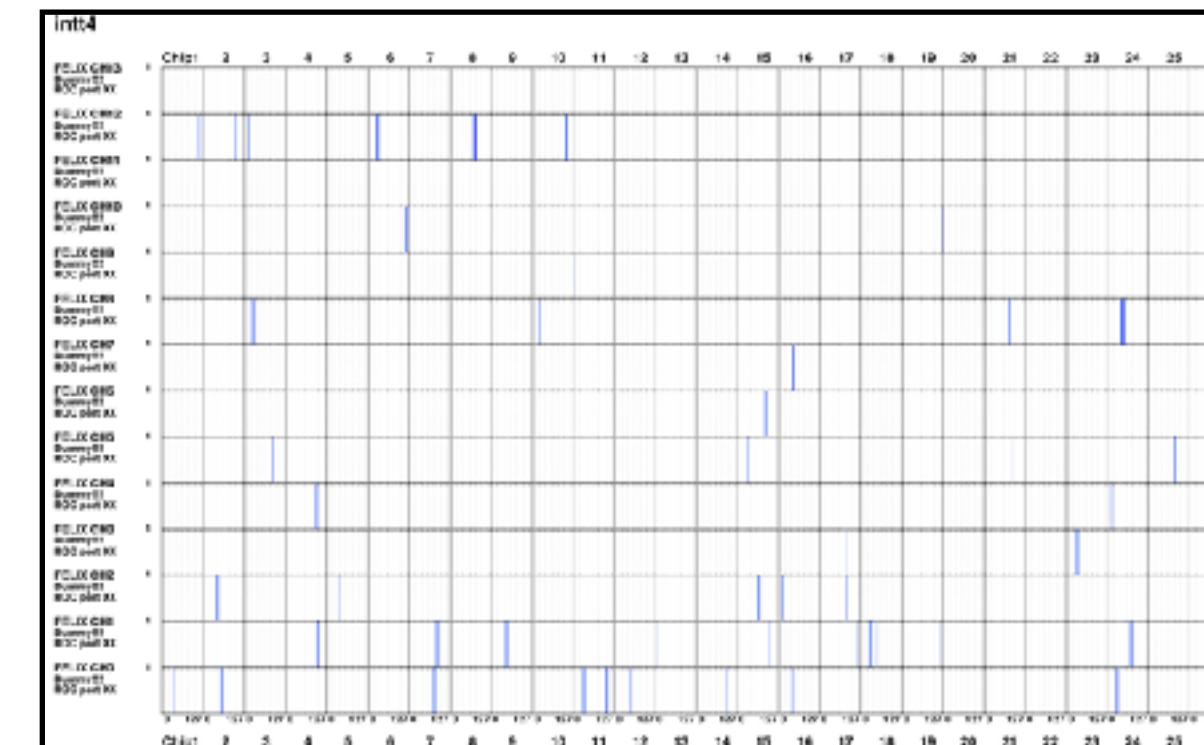
Current situation



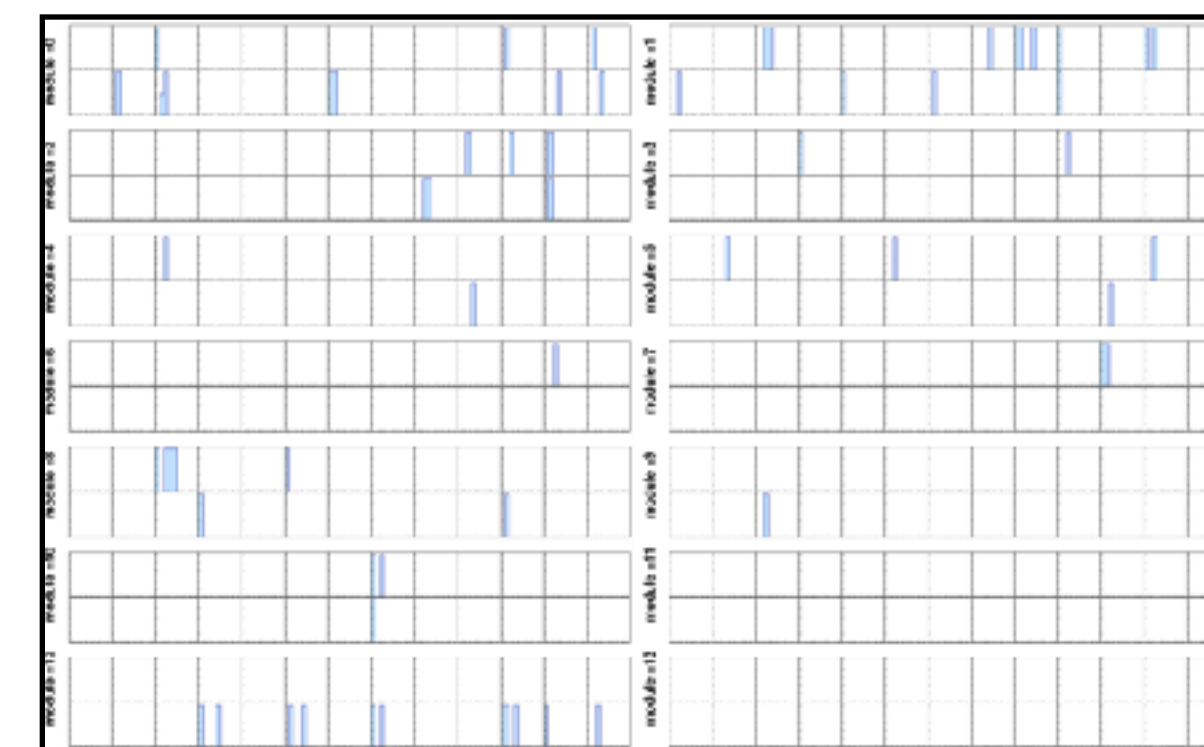
- Channel distributions (felix, half-ladder, chip, ch)
- ADC distributions (felix, half-ladder, chip)
- BCO difference distributions (felix, half-ladder)



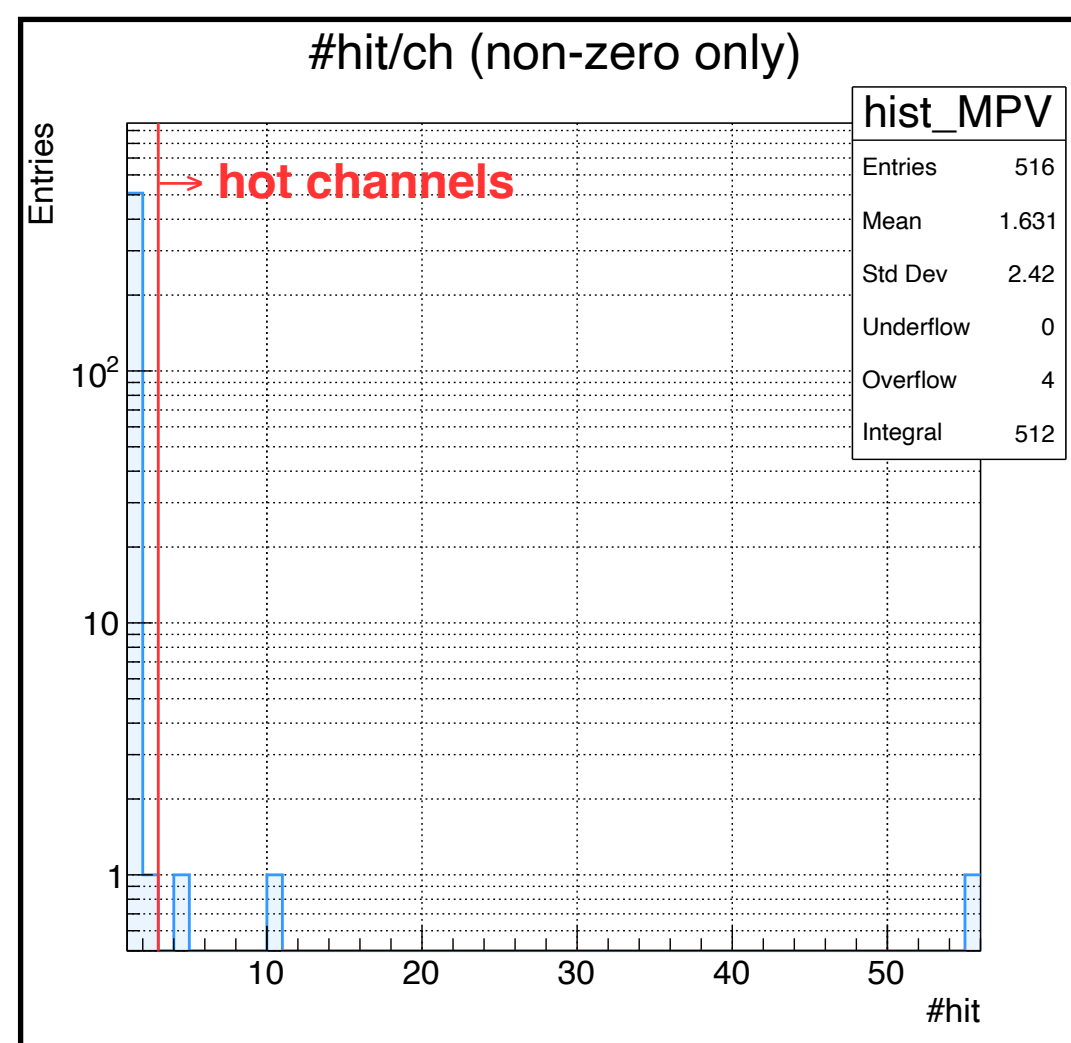
- #hit/ch for hot ch determination (cosmic runs)
- Cosmic event display (cosmic runs, WIP)



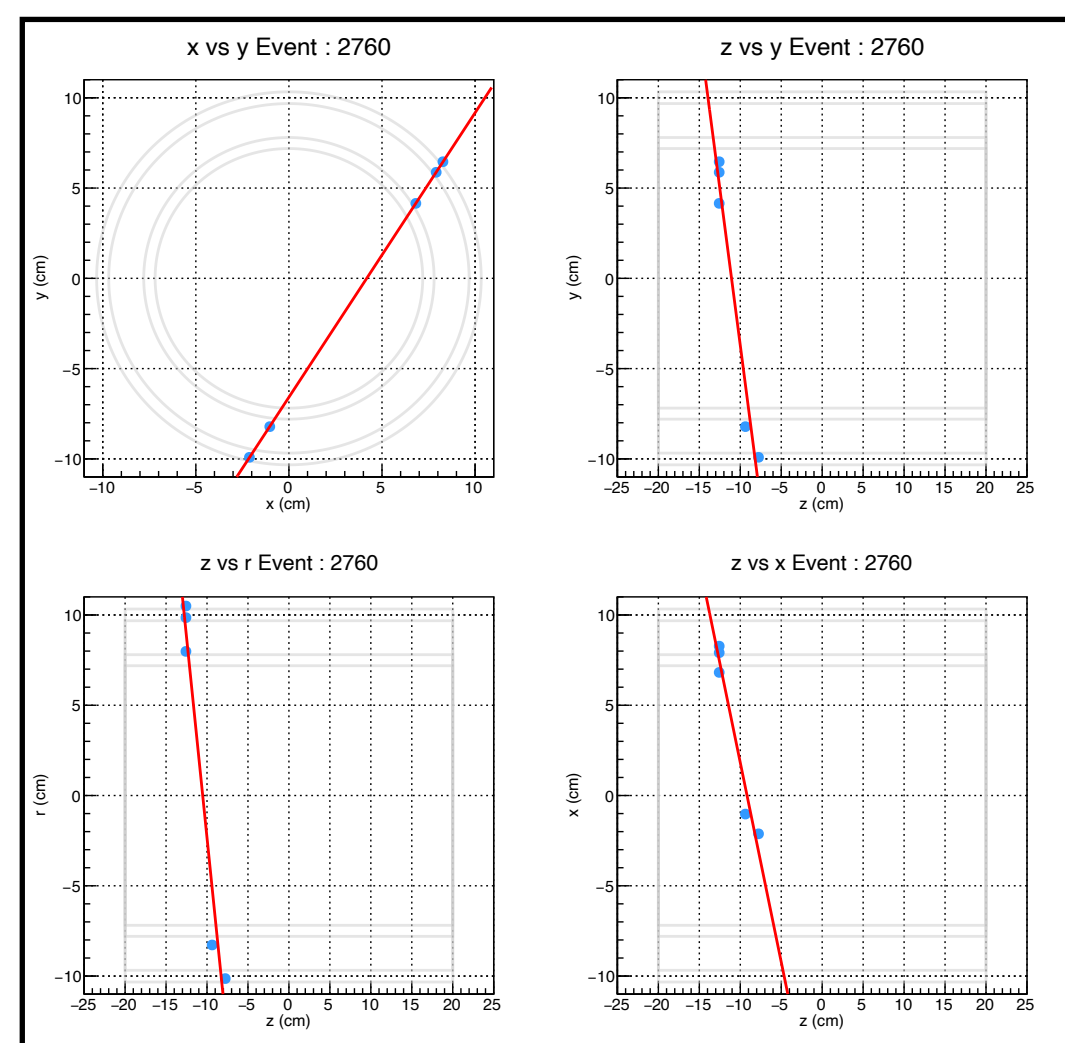
Ch dist



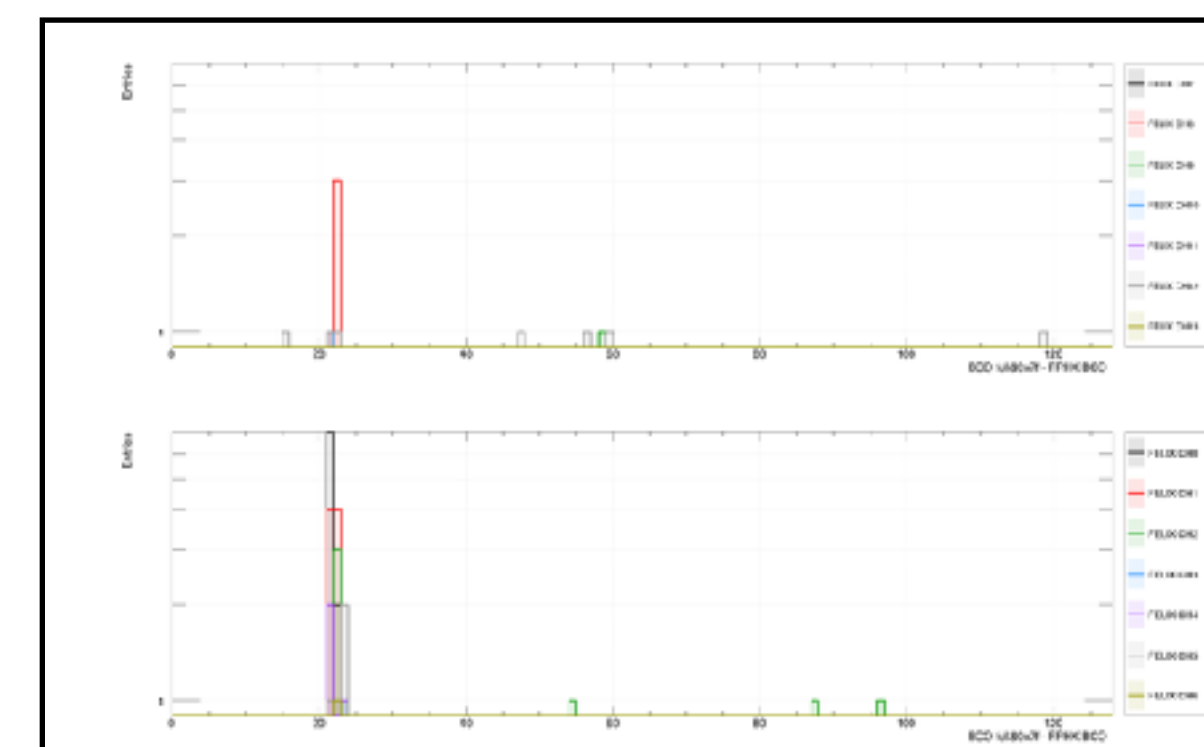
ADC dist



#hit/ch



Cosmic event display (simple ver)



BCO diff dist

Current situation

Run 39494

Go back to 2024 run list

Table of Contents for each FELIX:

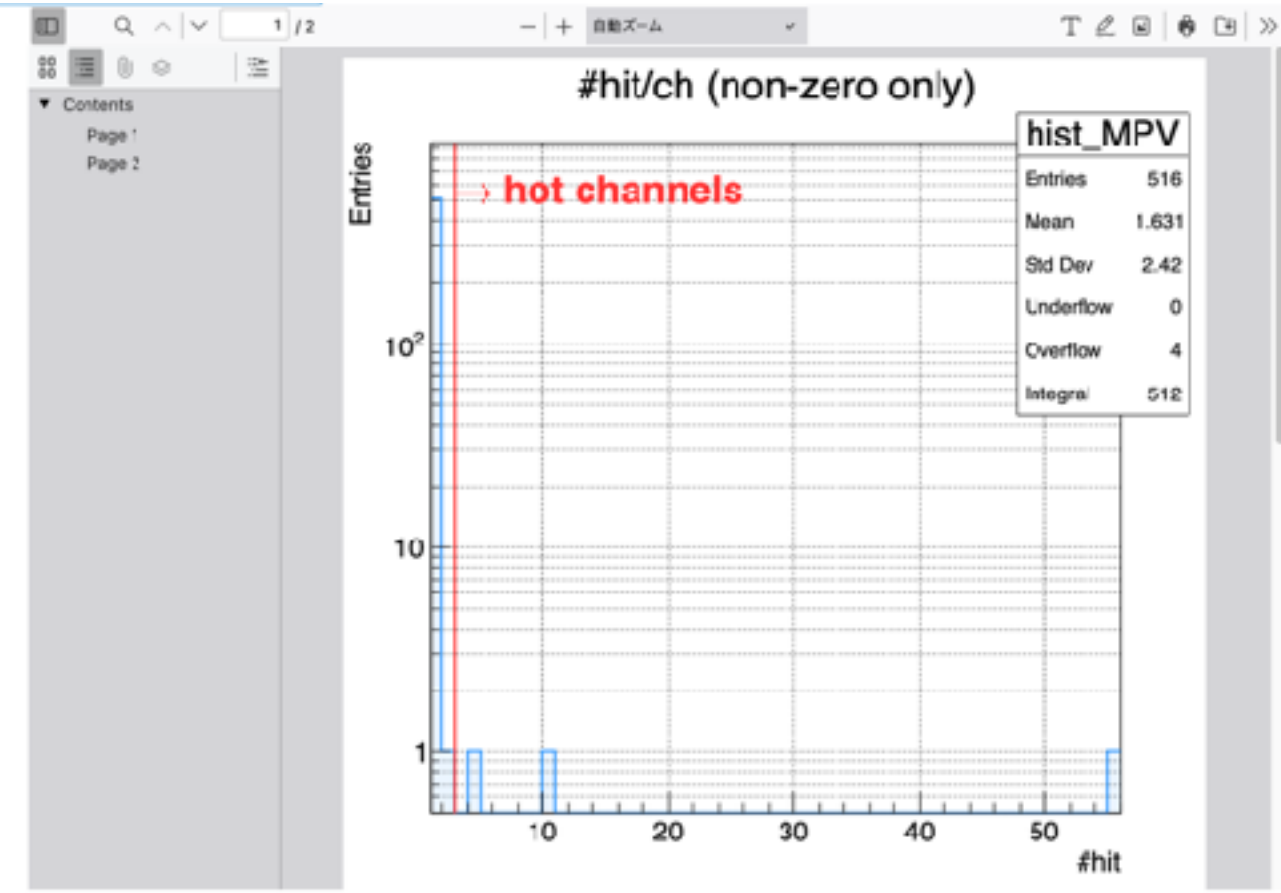
- [int0](#)
- [int1](#)
- [int2](#)
- [int3](#)
- [int4](#)
- [int5](#)
- [int6](#)
- [int7](#)

QA

Hot/Dead channel

List of hot channels

FELIX	FELIX ch	Chip	Channel
1	2	14	0
1	2	16	0
1	10	17	83
1	10	17	86
2	1	7	0
3	13	18	0
6	10	11	0



Plots for each FELIX

int0

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

int1

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

int2

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

int3

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

int4

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

int5

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

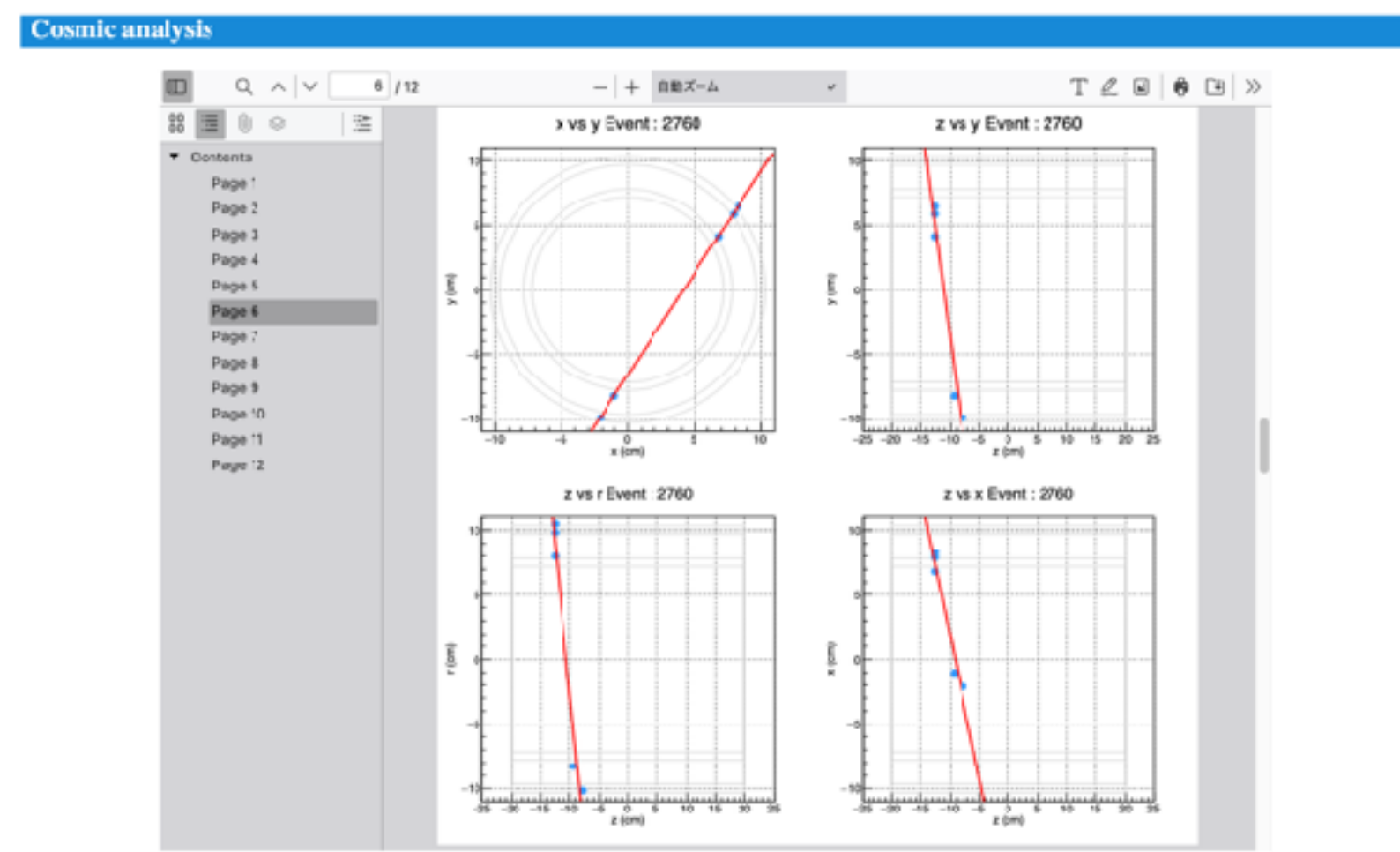
int6

cosmics ADC distribution chunk 0000

cosmics Unknown type chunk 0000

cosmics Hist distributions chunk 0000

Analysis results are shown in a homepage.
It's OK to keep using this method.

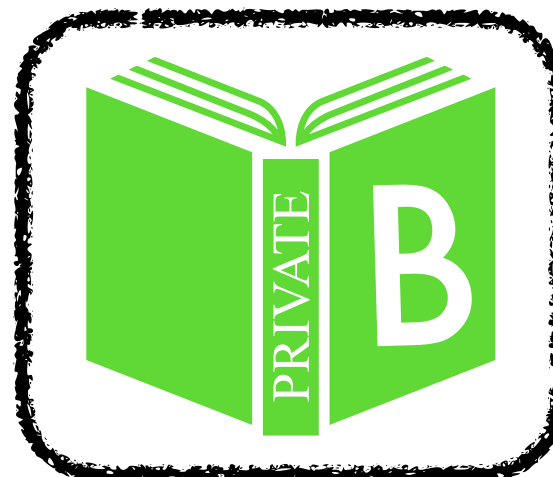


What can we do?



The usual hit-base TTree is available:

```
KEY: TTree          tree;2
KEY: TTree          tree;1
*Br    0 :pid        : pid/I
*Br    1 :adc        : adc/I
*Br    2 :ampl       : ampl/I
*Br    3 :chip_id   : chip_id/I
*Br    4 :module    : module/I
*Br    5 :chan_id   : chan_id/I
*Br    6 :bco       : bco/I
*Br    7 :bco_full  : bco_full/L
*Br    8 :evt       : evt/I
*Br    9 :roc       : roc/I
*Br   10 :barrel    : barrel/I
*Br   11 :layer     : layer/I
*Br   12 :ladder    : ladder/I
*Br   13 :arm       : arm/I
*Br   14 :full_fphx : full_fphx/I
*Br   15 :full_roc  : full_roc/I
```



Fun4All objects are available:

- INTTRAWHIT
 - same as our hit information at testbenches.
 - clone hits are included.
- TrkrHit
 - Information of corresponding INTTRAWHIT is accessible.
 - Index of layer, z, and phi are available.
 - x, y, and z coordinates available?
- TrkrCluster
 - x, y, and z coordinates available
 - Cluster size
 - Combined ADC information

The official DST also includes data of other tracking detectors except TPC.

➔ Output: ROOT file, PDF, text files to our QA directory

The tracking group also plan to do QA of TrkrCluster of INTT: We can directory contribute to their plan.

INTTRAWHIT

InttRawHit Class Reference

#include <coresoftware/blob/master/offline/framework/ffarawobjects/InttRawHit.h>

► Inheritance diagram for InttRawHit:

► Collaboration diagram for InttRawHit:

Public Member Functions

	InttRawHit ()=default
virtual	~InttRawHit ()=default
virtual uint64_t	get_bco () const
virtual void	set_bco (const uint64_t)
virtual int32_t	get_packetid () const
virtual void	set_packetid (const int32_t)
virtual uint32_t	get_word () const
virtual void	set_word (uint32_t)
virtual uint16_t	get_fee () const
virtual void	set_fee (uint16_t)
virtual uint16_t	get_channel_id () const
virtual void	set_channel_id (uint16_t)
virtual uint16_t	get_chip_id () const
virtual void	set_chip_id (uint16_t)
virtual uint16_t	get_adc () const
virtual void	set_adc (uint16_t)
virtual uint16_t	get_FPHX_BCO () const
virtual void	set_FPHX_BCO (uint16_t)
virtual uint16_t	get_full_FPHX () const
virtual void	set_full_FPHX (uint16_t)
virtual uint16_t	get_full_ROC () const
virtual void	set_full_ROC (uint16_t)
virtual uint16_t	get_amplitude () const
virtual void	set_amplitude (uint16_t)

► Public Member Functions inherited from **PHObject**

TrkrHit

TrkrHit Class Reference

Base class for hit object. [More...](#)

```
#include <coresoftware/blob/master/offline/packages/trackbase/TrkrHit.h>
```

► Inheritance diagram for TrkrHit:

► Collaboration diagram for TrkrHit:

Public Member Functions

	<code>~TrkrHit ()</code> override dtor
void	<code>identify (std::ostream &os=std::cout) const</code> override
void	<code>Reset ()</code> override Clear Event .
int	<code>isValid ()</code> const override isValid returns non zero if object contains valid data
virtual void	<code>addEnergy (const double)</code>
virtual double	<code>getEnergy ()</code>
virtual void	<code>setAdc (const unsigned int)</code>
virtual unsigned int	<code>getAdc ()</code>

[Dogygen: TrkrHit](#)

TrkrHitSet Class Reference

Public Types | Pu

Container for storing **TrkrHit**'s. [More...](#)

```
#include <coresoftware/blob/master/offline/packages/trackbase/TrkrHitSet.h>
```

► Inheritance diagram for TrkrHitSet:

► Collaboration diagram for TrkrHitSet:

Public Types

```
using Map = std::map< TrkrDefs::hitkey, TrkrHit * >  
using ConstIterator = Map::const_iterator  
using ConstRange = std::pair< ConstIterator, ConstIterator >
```

Public Member Functions

	void <code>identify (std::ostream &=std::cout) const</code> override Object functions.
	void <code>Reset ()</code> override Clear Event .
virtual void	<code>setHitSetKey (const TrkrDefs::hitsetkey)</code> Set the key for this object.
virtual TrkrDefs::hitsetkey	<code>getHitSetKey ()</code> const Get the key for this object.
virtual ConstIterator	<code>addHitSpecificKey (const TrkrDefs::hitkey, TrkrHit *)</code> Add a hit to this container using a specific key.
virtual void	<code>removeHit (TrkrDefs::hitkey)</code> Remove a hit using its key.
virtual TrkrHit *	<code>getHit (const TrkrDefs::hitkey) const</code> Get a specific hit based on its index.
virtual ConstRange	<code>getHits ()</code> const Get all hits.
virtual unsigned int	<code>size ()</code> const Get the number of hits stored.

[Dogygen: TrkrHitSet](#)

InttDefs Namespace Reference

Funcio

Utility functions for **INTT**. [More...](#)

Functions

static const unsigned int	<code>kBitShiftTimeBucketIdOffset</code>	<code>__attribute__((unused))=0</code> = 10
uint8_t	<code>getLadderZid (TrkrDefs::hitsetkey key)</code>	Get the ladder id from hitsetkey.
uint8_t	<code>getLadderZid (TrkrDefs::cluskey key)</code>	Get the ladder id from cluskey.
uint8_t	<code>getLadderPhid (TrkrDefs::hitsetkey key)</code>	Get the sensor id from hitsetkey.
uint8_t	<code>getLadderPhid (TrkrDefs::cluskey key)</code>	Get the sensor id from cluskey.
int	<code>getTimeBucketId (TrkrDefs::hitsetkey key)</code>	Generate a hitkey from a strip id.
int	<code>getTimeBucketId (TrkrDefs::cluskey key)</code>	Get the time bucket id from the hitsetkey.
uint16_t	<code>getCol (TrkrDefs::hitkey key)</code>	Get the time bucket id from the cluskey.
uint16_t	<code>getRow (TrkrDefs::hitkey key)</code>	Get the row index from hitkey.
TrkrDefs::hitkey	<code>genHitKey (const uint16_t col, const uint16_t row)</code>	
TrkrDefs::hitsetkey	<code>genHitSetKey (const uint8_t lyr, const uint8_t ladder_z_index, const uint8_t ladder_phi_index, const int time_bucket)</code>	Generate a hitsetkey for the Intt.
TrkrDefs::cluskey	<code>genClusKey (const uint8_t lyr, const uint8_t ladder_z_index, const uint8_t ladder_phi_index, const int crossing, const uint32_t clusid)</code>	Generate a cluster key from indeces.
TrkrDefs::hitsetkey	<code>resetCrossingHitSetKey (const TrkrDefs::hitsetkey hitsetkey)</code>	Zero the crossing bits in a copy of the hitsetkey.

[Dogygen: InttDefs](#)

TrkrCluster

TrkrCluster Class Reference

Base class for cluster object. [More...](#)

```
#include <coresoftware/blob/master/offline/packages/trackbase/TrkrCluster.h>
```

► Inheritance diagram for TrkrCluster:

► Collaboration diagram for TrkrCluster:

Public Member Functions

	<code>~TrkrCluster ()</code> override=default dtor
void	<code>Identify (std::ostream &os=std::cout)</code> const override
void	<code>Reset ()</code> override Clear Event .
int	<code>IsValid ()</code> const override IsValid returns non zero if object contains valid data
virtual void	<code>CopyFrom (const TrkrCluster &)</code> copy content from base class
virtual void	<code>CopyFrom (TrkrCluster *)</code> copy content from base class
virtual float	<code>getLocalX ()</code> const
virtual void	<code>setLocalX (float)</code>
virtual float	<code>getLocalY ()</code> const
virtual void	<code>setLocalY (float)</code>
virtual void	<code>setAdc (unsigned int)</code>
virtual unsigned int	<code>getAdc ()</code> const
virtual void	<code>setMaxAdc (uint16_t)</code>
virtual unsigned int	<code>getMaxAdc ()</code> const
virtual char	<code>getOverlap ()</code> const
virtual void	<code>setOverlap (char)</code>
virtual char	<code>getEdge ()</code> const
virtual void	<code>setEdge (char)</code>
virtual void	<code>setTime (const float)</code>
virtual float	<code>getTime ()</code> const
virtual char	<code>getSize ()</code> const
virtual float	<code>getPhiSize ()</code> const
virtual float	<code>getZSize ()</code> const
virtual float	<code>getPhiError ()</code> const
virtual float	<code>getRPhiError ()</code> const
virtual float	<code>getZError ()</code> const

virtual void	<code>setActsLocalError (unsigned int, unsigned int, float)</code> Acts functions, for Acts modules use only.
virtual float	<code>getActsLocalError (unsigned int, unsigned int)</code> const
virtual TrkrDefs::subsurfkey	<code>getSubSurfKey ()</code> const
virtual void	<code>setSubSurfKey (TrkrDefs::subsurfkey)</code>
virtual float	<code>getX ()</code> const
virtual void	<code>setX (float)</code>
virtual float	<code>getY ()</code> const
virtual void	<code>setY (float)</code>
virtual float	<code>getZ ()</code> const
virtual void	<code>setZ (float)</code>
virtual float	<code>getPosition (int)</code> const
virtual void	<code>setPosition (int, float)</code>
virtual void	<code>setGlobal ()</code>
virtual void	<code>setLocal ()</code>
virtual bool	<code>isGlobal ()</code> const
virtual float	<code>getError (unsigned int, unsigned int)</code> const
virtual void	<code>setError (unsigned int, unsigned int, float)</code>
virtual float	<code>getSize (unsigned int, unsigned int)</code> const
virtual void	<code>setSize (unsigned int, unsigned int, float)</code>

[Dxygen: TrkrCluster](#)

What do we monitor?



- ✓ Channel distributions (felix, chip, ch)
- ✓ ADC distributions (felix, chip)
- ✓ BCO difference distributions (felix, chip)

Nothing changed, nothing added as much as possible.

• INTTRAWHIT

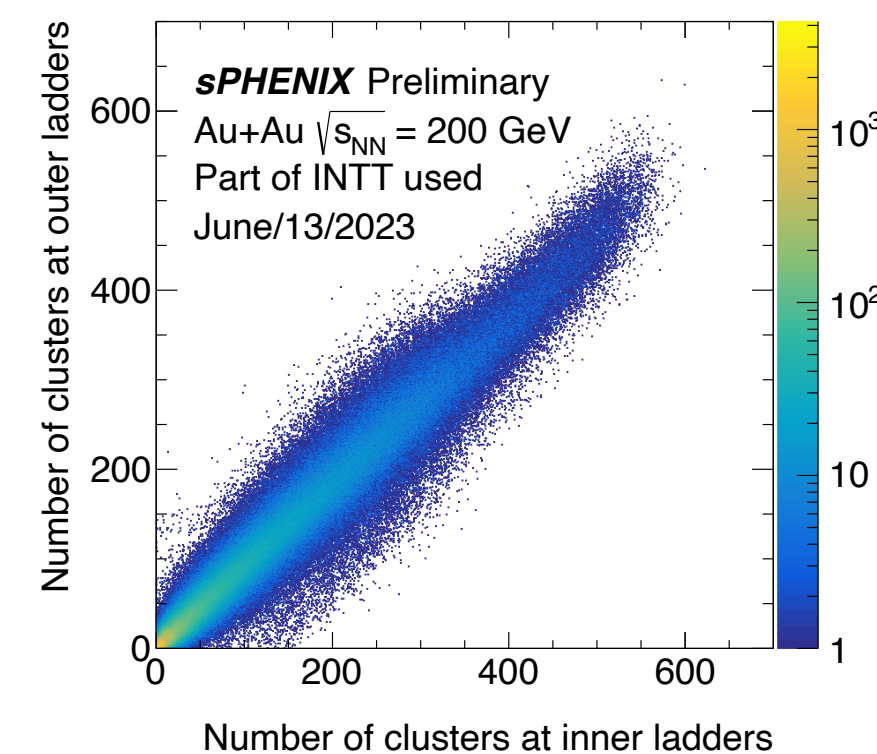
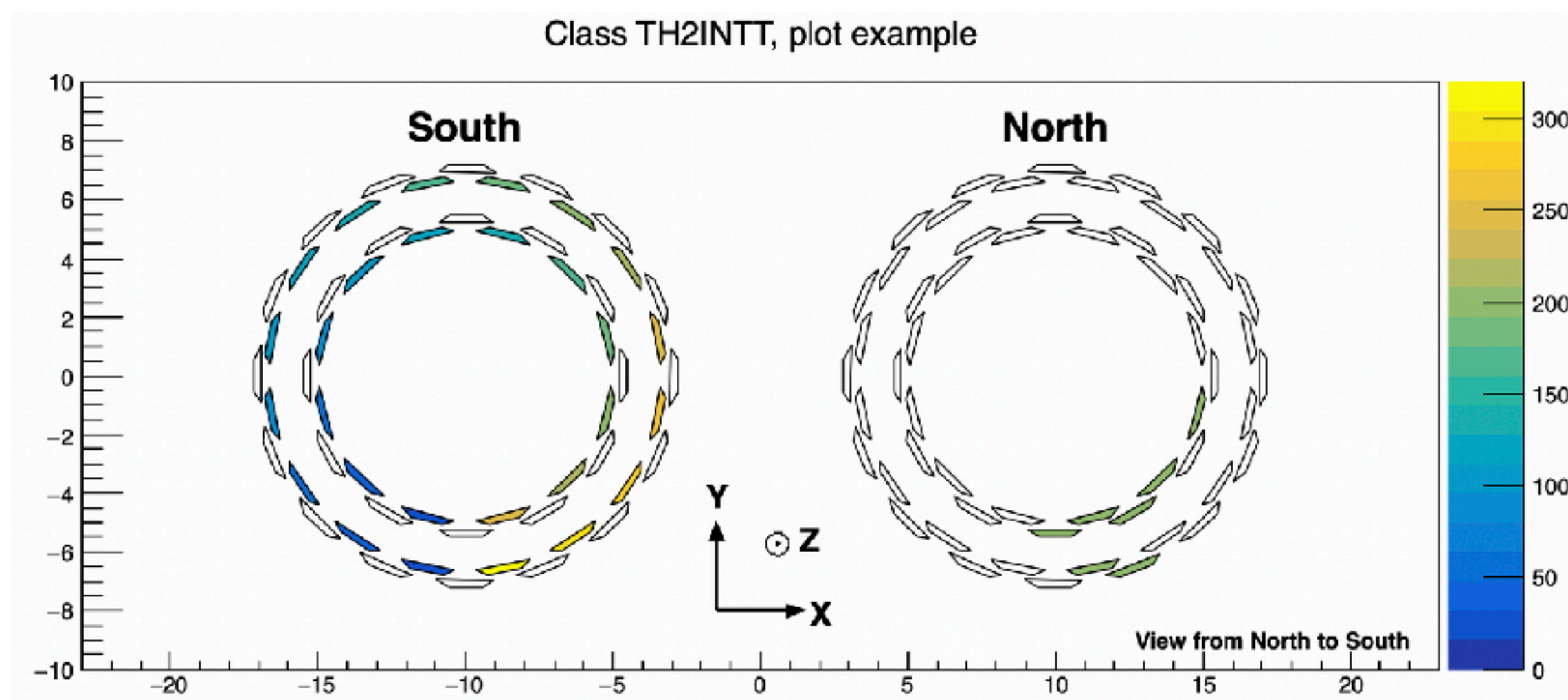
- ✓ the number of hits per event
- ✓ the number of hits per FELIX (per event)
- ✓ BCO
- ✓ BCO full
- BCO difference
- ✓ ADC
- ✓ hitmap (felix, half-ladder)
- Occupancy?: how much ratio of a chip (half-ladder?) occupied in an event

• TrkrHit

- the same distributions of INTTRAWHIT
- Cross-section hitmap (using TH2INT)
- Correlation of #hit/event
 - outer barrel vs inner barrel
 - top half vs bottom half
 - FELIX vs FELIX
- Clone hit information?

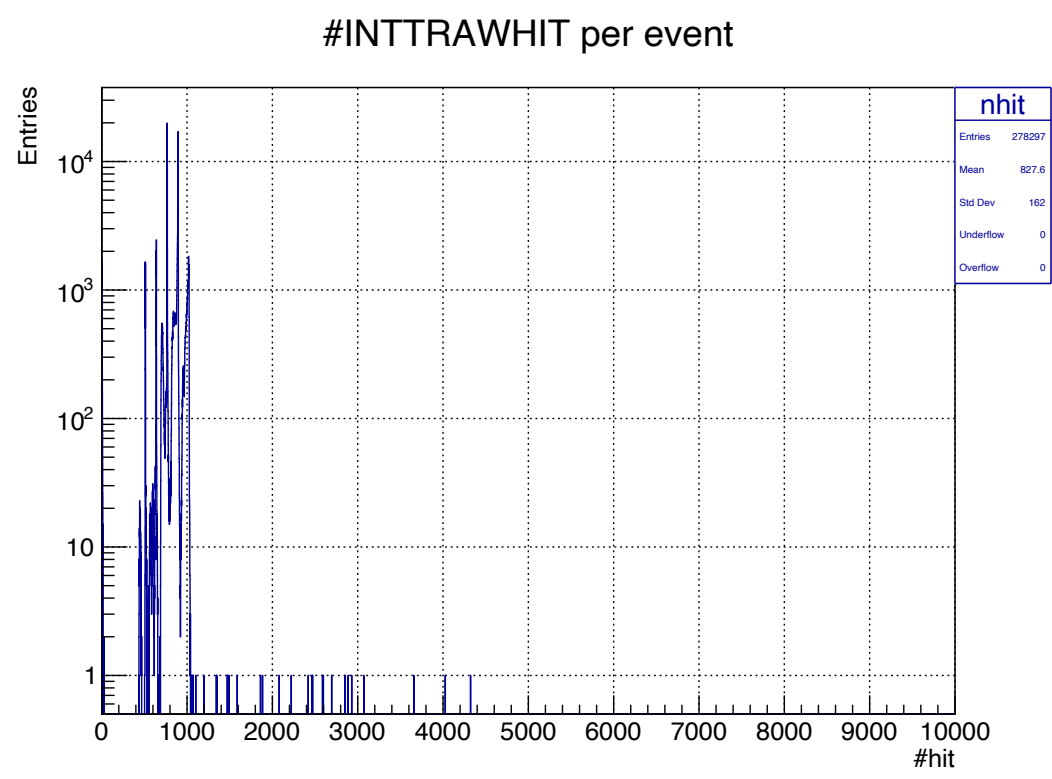
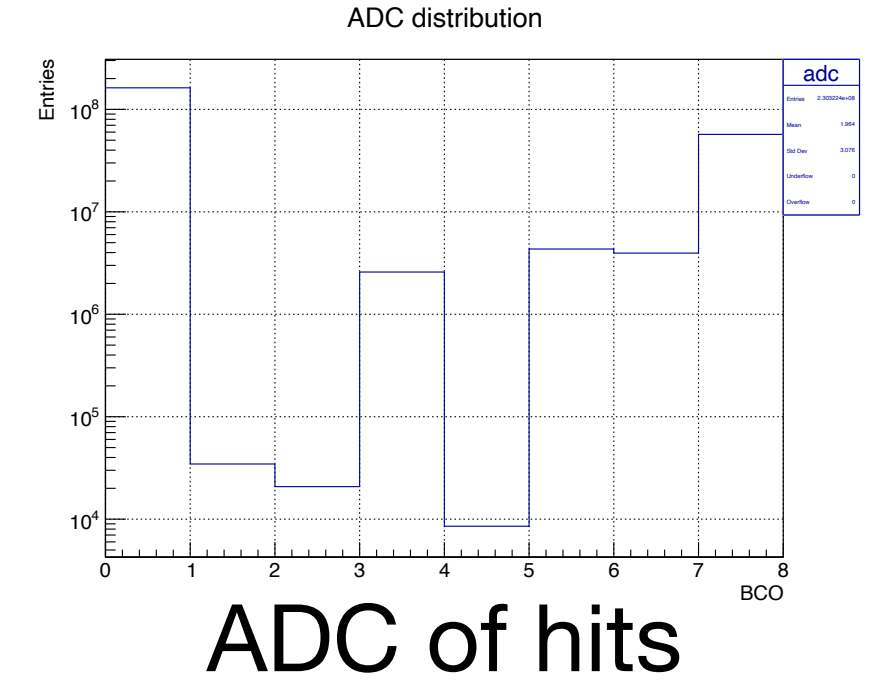
• TrkrCluster

- the same distributions of TrkrHit
- Visualization of x, y, z distributions
- Cluster size

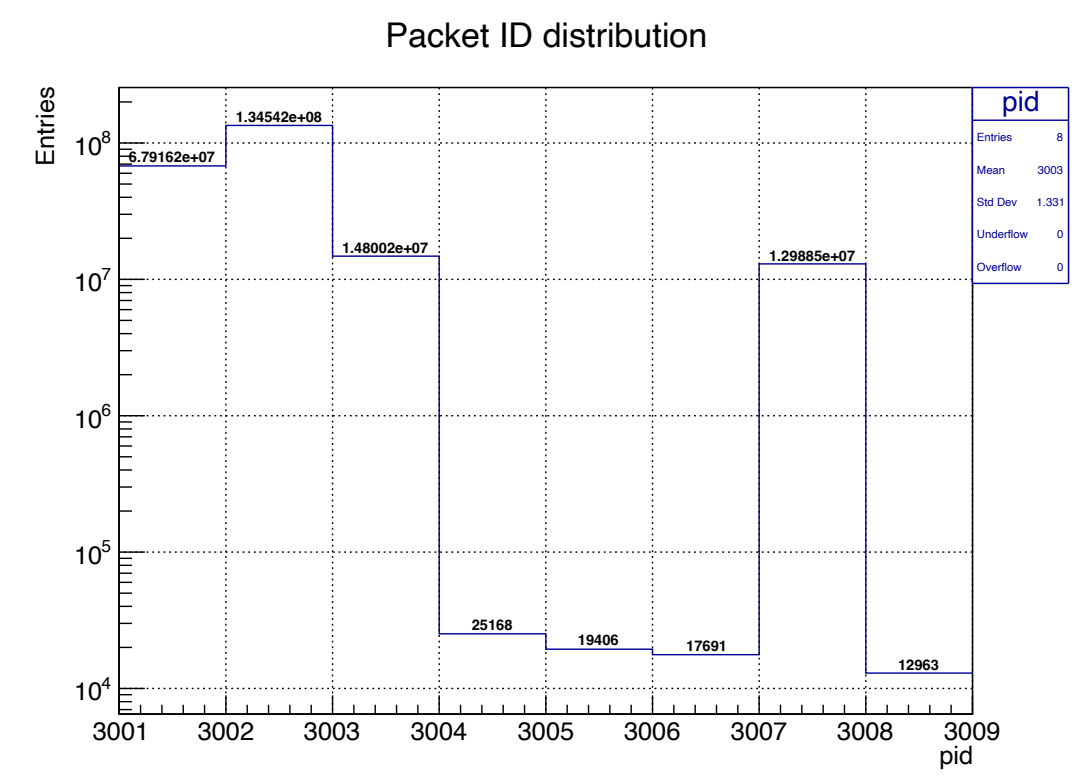


Current status of INTTRAWHIT QA

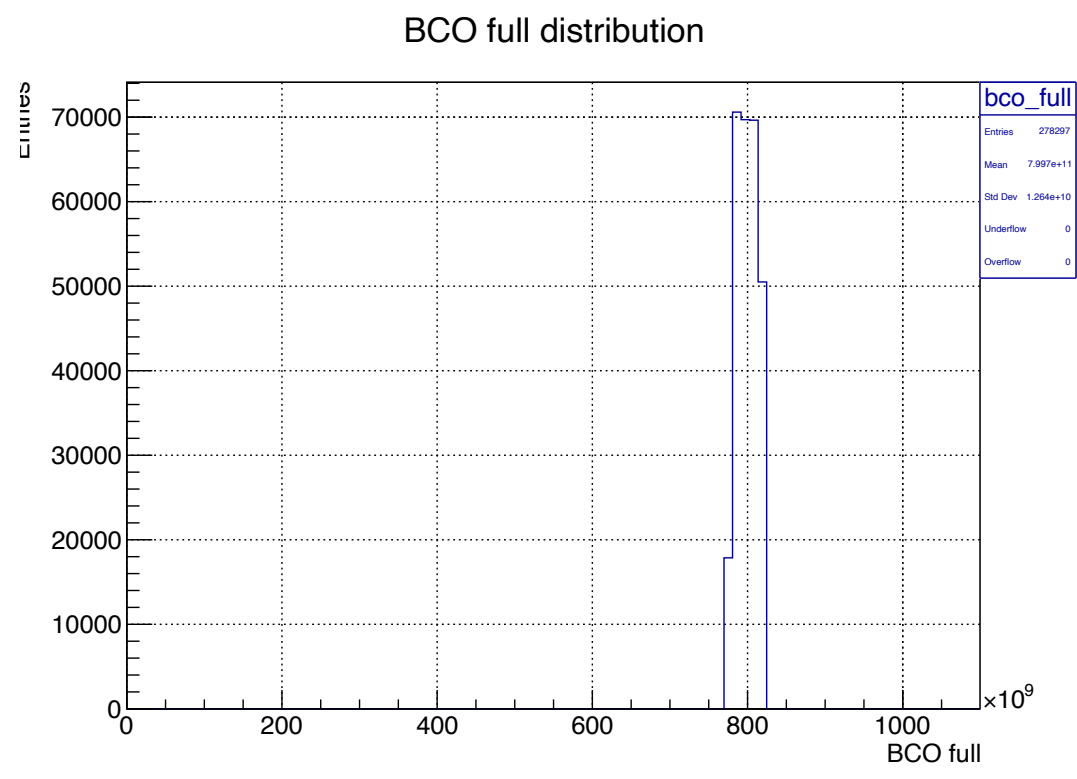
GitHub: https://github.com/sPHENIX-Collaboration/INTT/tree/main/QA_codes



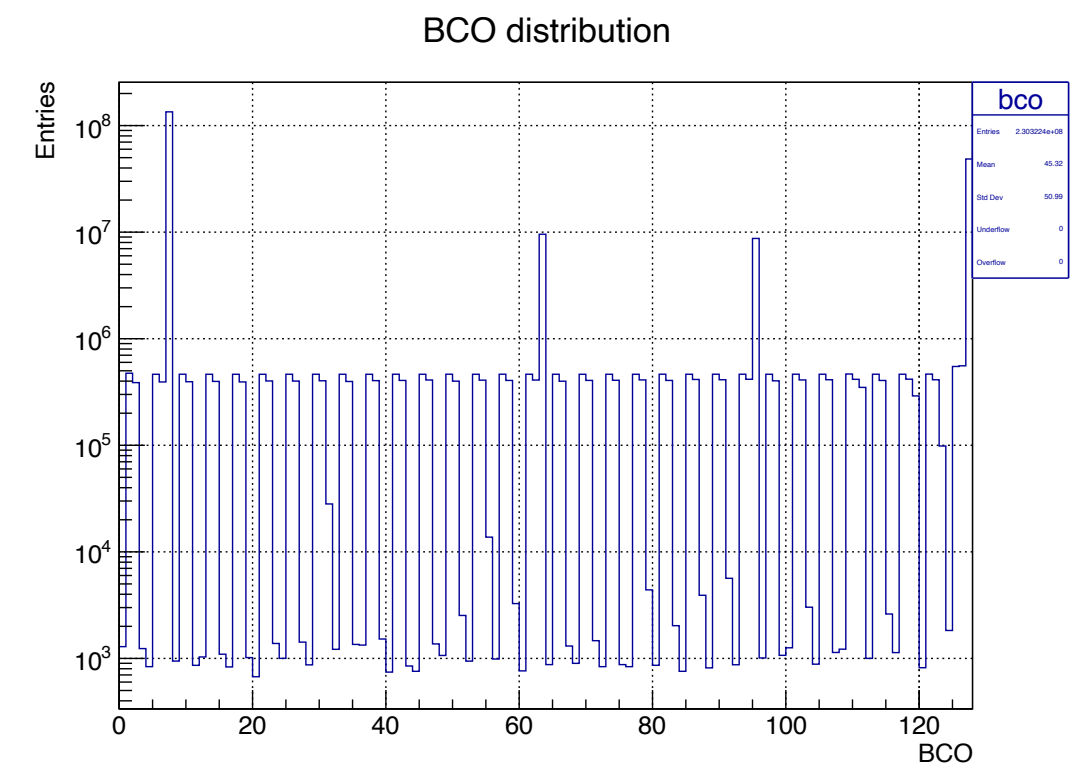
#hit/event
It's OK to check anyway.



#hit/felix
We can see whether all FELIX worked or not.



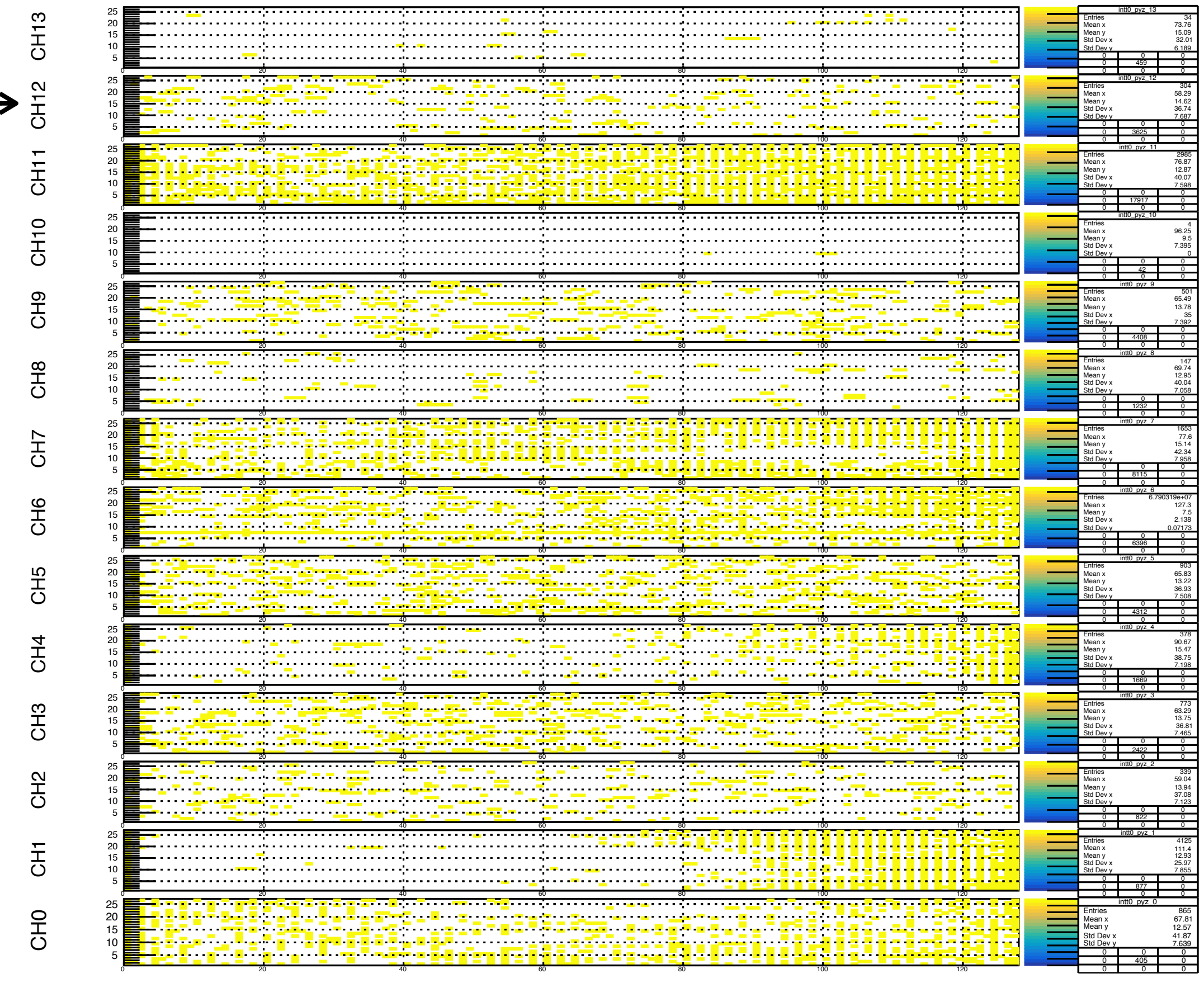
BCO of hits



BCO full of events

each half-ladder →

chip ↑



Hitmap (felix, half-ladder)