

2024/01/14-15  
sPHENIX INTT フォローアップワークショップ  
立教大学池袋キャンパス

こんにちは Fun4All!  
～韓国ワークショップのつづき～  
糠塚元気 (理研)

what/who

why

where

when→now

how

# What is Fun4All?

An analysis framework originally developed for the PHENIX experiment

## Software framework

Article Talk 文 37 languages ▾

From Wikipedia, the free encyclopedia

"Framework (computer science)" redirects here. For other uses, see [Framework \(disambiguation\)](#).

In computer programming, a **software framework** is an [abstraction](#) in which [software](#), providing generic functionality, can be selectively changed by additional user-written code, thus providing application-specific software. It provides a standard way to build and deploy applications and is a universal, reusable [software environment](#) that provides particular functionality as part of a larger [software platform](#) to facilitate the development of [software applications](#), products and solutions.

## 소프트웨어 프레임워크

문서 토론 文 37개 언어 ▾

위키백과, 우리 모두의 백과사전.

컴퓨터 프로그래밍에서 **소프트웨어 프레임워크**(software framework)는 복잡한 문제를 해결하거나 서술하는 데 사용되는 기본 개념 구조이다. 간단히 빠대, 골조(骨組), 프레임워크(framework)라고도 한다. 이렇게 매우 폭넓은 정의는 이 용어를 [buzzword](#)로서, 특히 [소프트웨어](#) 환경에서 사용할 수 있게 만들어 준다.

READ  
LATER

Wikipedia

## 軟體框架

条目 讨论 汉 漢 不转换 ▾ 阅读 编辑 查看历史 工具 ▾

维基百科, 自由的百科全书



此條目没有列出任何参考或来源。 (2016年8月3日)

維基百科所有的內容都應該可供查證。请协助补充可靠来源以改善这篇条目。无法查证的内容可能会因为异议提出而被移除。

軟體框架 (software framework)，通常指的是為了實現某個業界標準或完成特定基本任務的軟體組件規範，也指為了實現某個軟體組件規範時，提供規範所要求之基礎功能的軟體產品。

框架的功能類似於基礎設施，與具體的軟體應用無關，但是提供並實現最為基礎的軟體架構和體系。[軟體開發者](#)通常依據特定的框架實現更為複雜的商業運用和業務邏輯。這樣的軟體應用可以在支持同一種框架的軟體系統中運行。

簡而言之，框架就是制定一套規範或者規則（思想），大家（程序员）在該規範或者規則（思想）下工作。或者說使用別人搭好的舞台來做編劇和表演。

## ソフトウェアフレームワーク

ページ ノート 閲覧 編集 履歴表示 ツール ▾

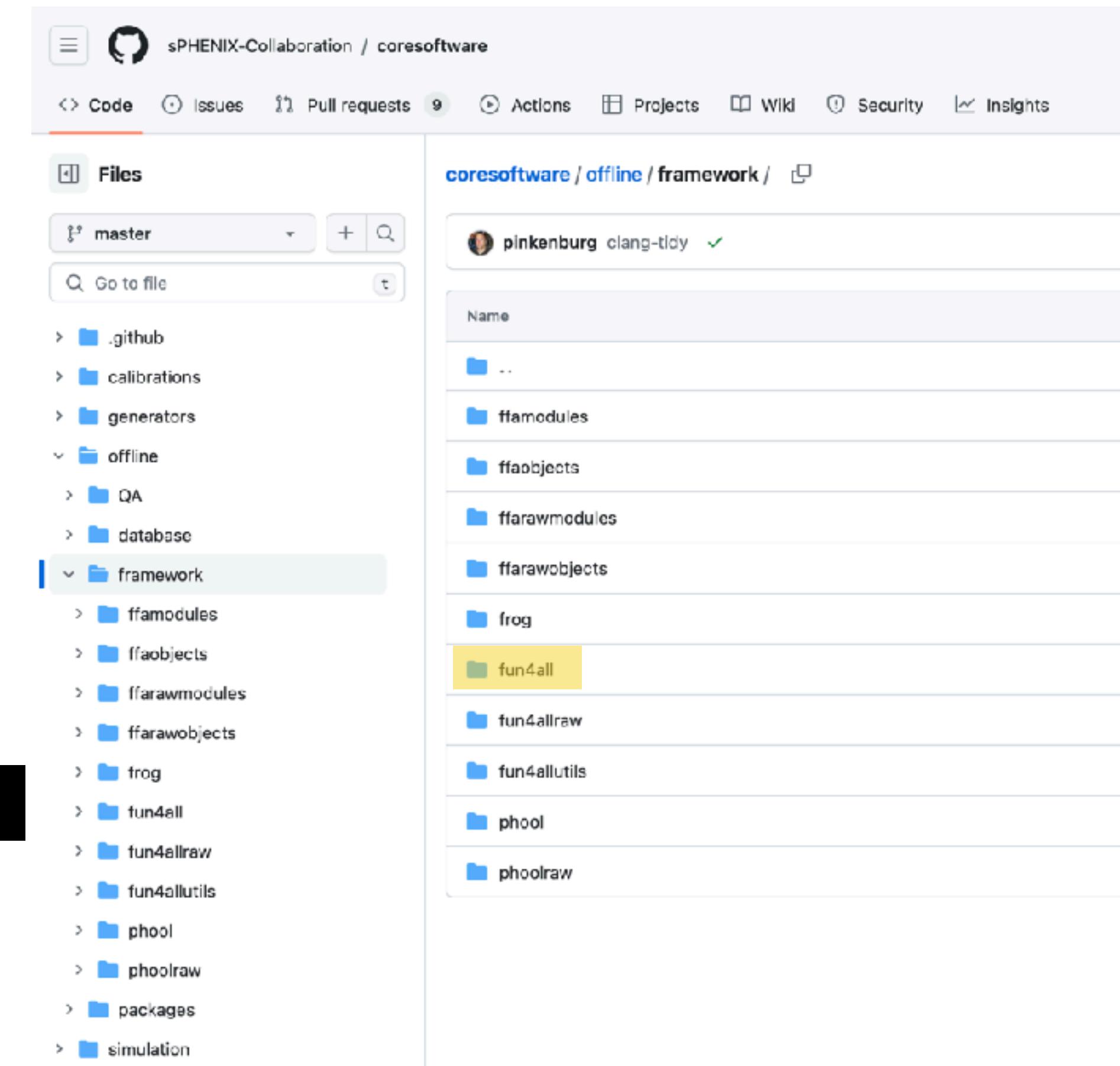
出典: フリー百科事典『[ウィキペディア \(Wikipedia\)](#)』

ソフトウェアフレームワーク (英: software framework) とは、[プログラミング](#)において、[アプリケーションソフトウェア](#)等の実装に必要となる一般的な機能や定型コードを、[ライブラリ](#)としてあらかじめ用意したものである。例えば、[Java](#)などのオブジェクト指向言語向けのクラスライブラリとして実装されている場合は、再利用可能なソフトウェア部品（[ソフトウェアコンポーネント](#)）として用意されているクラスのインスタンスを自由に組み合わせたり、基本的な機能を持つ基底クラスを継承した派生クラスをユーザーが定義し、仮想メソッドによって公開されているカスタマイズポイントを選択的に上書きしたり特化させたりする。言語によってはコールバックやデリゲートを利用するなど、他にもさまざまな形態がある。文脈から明確な場合は単に「フレームワーク」としたり、特にアプリケーションソフトウェア開発向けであることを明確にした「[アプリケーションフレームワーク](#)」など、前後に別の語をつなげた複合語を使ったりすることもある。

what/who  
why  
**where**  
when→now  
**how**

# Where is Fun4All?

- You can find it on GitHub:  
<https://github.com/sPHENIX-Collaboration/coresoftware>
- You can use it in the SDCC servers.



## Steps to set up Fun4All in SDCC

1. Log in to the SDCC gateway machine:

```
$ ssh {username}@ssh.sdcc.bnl.gov
```

2. Log in to the SDCC servers:

```
$ ssh {username}@sphnx{num}sdcc.bnl.gov  
{num}: 01 – 08
```

3. Execute the setting shell script:

```
$ source /opt/sphenix/core/bin/sphenix_setup.sh
```

# Why is Fun4All difficult?

You need

a variety of knowledge

and experience

Static  
library

Finding  
information

sPHENIX  
scripts

Compiling  
make

Fun4All

ROOT  
macro

C++

Git  
GitHub

automake  
configure

Linux

# もくじ

1. [INTT Fun4All tutorial repository in GitHub](#)
2. Sample1 (the simplest)
  - 2.1.[ROOT INCLUDE PATH](#)
  - 2.2.[LD\\_LIBRARY\\_PATH](#)
3. Sample2 (add analysis module)
  - 3.1.[SubsysReco class](#)
  - 3.2.[make and Makefile \(NWU workshop in JP\)](#)
4. About DST
  - 4.1.[Official DST](#)
5. Sample3 (InttRawHit analysis)
  - 5.1.[Reading DST file\(s\)](#)
6. [Sample4 \(TrkrCluster analysis\)](#)
  - 6.1.[including \\*.C macros](#)
  - 6.2.[宿題](#)
7. [MBD のデータはどこ？](#)
  - 7.1.[MBD データの処理手順](#)  
(糠塚の理解している範囲)
8. [Sample5 \(TrkrCluster + MbdOut\)](#)

# INTT\_Fun4All\_Tutorial repository

[https://github.com/nukazuka/INTT\\_Fun4All\\_Tutorial](https://github.com/nukazuka/INTT_Fun4All_Tutorial)

It belongs to my private account but not sPHENIX.

The screenshot shows the GitHub repository page for 'nukazuka / INTT\_Fun4All\_Tutorial'. The repository is public and has 33 commits. The main branch is 'main' with 1 branch and 1 tag. The README file contains a section titled 'Fun4All tutorial for the INTT workshops' which describes the purpose of the repository. It also lists two versions: 'ver2023' and 'ver2024', along with their descriptions and usage notes. The repository's activity section shows recent commits from 'Genki Nukazuka' and 'ver2024'. The right sidebar provides information about the repository, including its purpose (A repository to provide samples in the sPHENIX INTT analysis workshop), statistics (0 stars, 1 watching, 0 forks, 0 issues, 1 pull request, 0 releases, 0 packages), and suggested workflows (SLSA Generic generator).

[https://github.com/nukazuka/INTT\\_Fun4All\\_Tutorial/tree/main](https://github.com/nukazuka/INTT_Fun4All_Tutorial/tree/main)

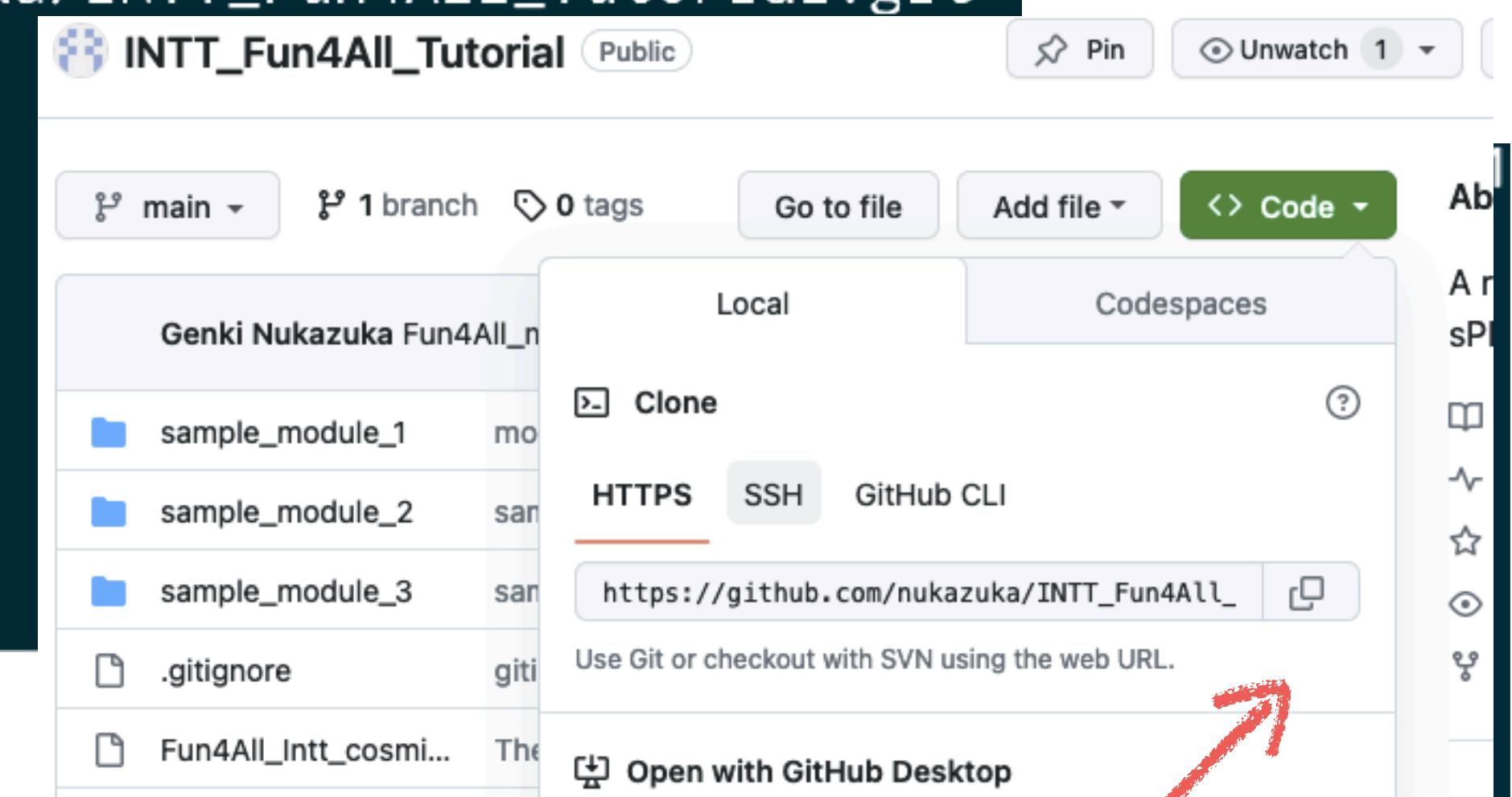
# Get the sample codes

You can get the sample codes: [https://github.com/nukazuka/INTT\\_Fun4All\\_Tutorial](https://github.com/nukazuka/INTT_Fun4All_Tutorial)

1. Make a working directory under /sphenix/tg/tg01/commissioning/INTT/work/[yours] or anywhere you like.
2. Get them by

```
$ git clone git@github.com:nukazuka/INTT_Fun4All_Tutorial.git  
in your working directory
```

```
[genki 17:55:14 fun4all_tutorial] $ git clone git@github.com:nukazuka/INTT_Fun4All_Tutorial.git  
Cloning into 'INTT_Fun4All_Tutorial'...  
X11 forwarding request failed on channel 0  
remote: Enumerating objects: 50, done.  
remote: Counting objects: 100% (50/50), done.  
remote: Compressing objects: 100% (33/33), done.  
remote: Total 50 (delta 8), reused 47 (delta 8), pack-reused 0  
Receiving objects: 100% (50/50), 443.36 KiB | 417.00 KiB/s, done.  
Resolving deltas: 100% (8/8), done.
```



[back to ToC](#)

# INTT\_Fun4All\_Tutorial repository

Structure:

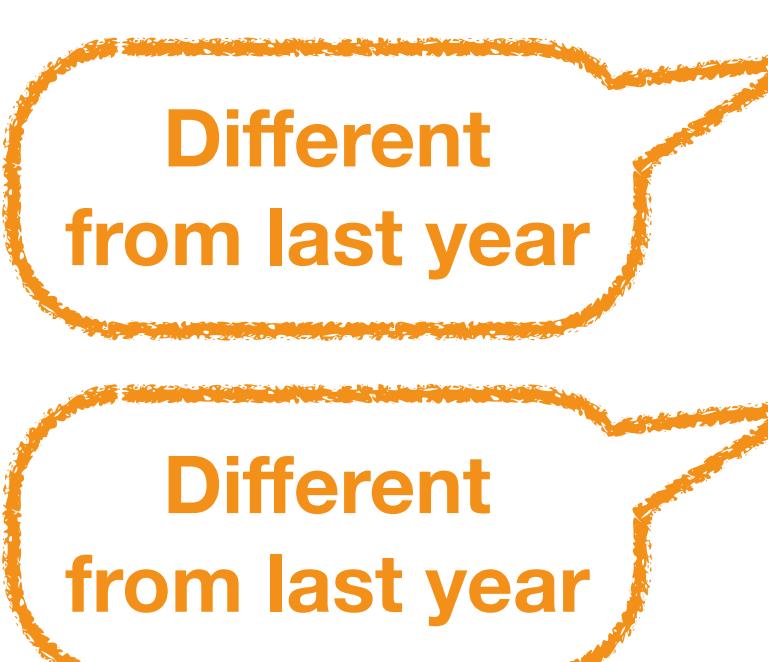
```
[nukazuka@sphnx05 08:56:08 INTT_Fun4All_Tutorial] $ tre
```

```
.---- ver2023  
|---- ver2024  
|    |---- Fun4All_samples  
|    |    |---- Fun4All_minimum.C  
|    |    |---- Fun4All_minimum_2.C  
|    |    |---- sample_module_2  
|    |    |    |---- Makefile  
|    |    |    |---- Makefile.am  
|    |    |    |---- autogen.sh  
|    |    |---- configure  
|    |    |---- configure.ac  
|    |    |---- tutorial.cc  
|    |    |---- tutorial.h  
|    |    |---- (etc...)  
|    |---- sample_module_2  
|    |---- Fun4All_minimum_3.C  
|    |---- sample_module_3  
|    |---- Fun4All_minimum_4.C  
|    |---- sample_module_4  
|    |---- Fun4All_minimum_5.C  
|    |---- sample_module_5  
|    |---- Fun4All_minimum_6.C  
|    |---- sample_module_6  
|---- cpp_basics  
---- README.md
```

[https://github.com/nukazuka/INTT\\_Fun4All\\_Tutorial](https://github.com/nukazuka/INTT_Fun4All_Tutorial)

It belongs to my private account but not sPHENIX.

# INTT\_Fun4All\_Tutorial repository

- sample1
    - Exercise for running Fun4All.
    - It contains **minimum codes** to show the simplest case. So it does not require any analysis module and **does nothing**.
    - Also, we will check the configuration of your environment. If you have no idea what to do, I'll show an example.
  - sample2
    - You can see **how to add your analysis module**.
    - The analysis module (sample\_module\_2) just prints words on your terminal.
  - sample3
    - **InttRawHit** is taken from a given **DST** and analyzed using the sample analysis module (sample\_module\_3).
  - sample4
    - **TrkrCluster** is taken from a given **DST** and analyzed using the sample analysis module (sample\_module\_4).
- 

# INTT\_Fun4All\_Tutorial repository: Read README.md

The image shows a screenshot of a GitHub repository page for 'INTT\_Fun4All\_Tutorial'. The repository has 1 branch and 1 tag. The main README file contains sections for 'Fun4All tutorial for the INTT workshops' and 'INTT\_Fun4All\_Tutorial in 2024 at Korea Univ.'. It also lists commit history for 'ver2023' and 'ver2024'. A red arrow points from the 'README' section of the main page to the 'README.md' file located in the 'app\_basics' directory. Another red arrow points from the 'app\_basics' README to the 'Samples for Fun4All tutorial' section of the main README.

**Fun4All tutorial for the INTT workshops**

Samples codes for the Fun4All tutorial in the INTT workshops are here. These are written by Genki Nukazuka (RIKEN/RBRC). If you find a bug, just let me know or fix it and commit it.

**Versions**

**ver2023**

ver2023 was used in [the workshop at Taiwan](#) and [the follow up workshop at NWU](#). Libraries at that time (around ana.390 (Nov/25/2023)) are good to use. The latest version at Nov/08/2024 (?) doesn't work. ana.?? (Ryota knows) is OK to use.

**ver2024**

ver2024 is used in [the workshop at Korea](#). The library version ana.444 (or the latest when the implementation was done) (should) works.

**README**

Genki Nukazuka sample2 was simplified. 3e25623 · 17 hours ago 83 Commits

ver2023 Minor updates on the C++ answers ver2024. More explain... last week

ver2024 sample2 was simplified. 17 hours ago

.gitignore gitignore updated last week

README.md README at the top level was made. last week

...  
Fun4All\_samples sample2 was simplified. 17 hours ago  
app\_basics Minor updates on the C++ answers ver2024. More ex... last week  
README.md README.md updated now

**README.md**

**Samples for Fun4All tutorial**

**Tutorial samples**

**Step1**

The minimum sample. It's just a practice to run Fun4All.

**Files**

- Fun4All\_minimum.C

**Step2**

It shows how to add your own analysis module.

**Files/Directories**

- Fun4All\_minimum\_2.C
- sample\_module\_2

**Step3**

See INTT hits from MC events.

**Files/Directories**

- Fun4All\_minimum\_3.C
- sample\_module3

**Step4**

A DST file is read, and information of INTT...

**README.md**

**Samples for Fun4All tutorial**

**Tutorial samples**

**Step1**

The minimum sample. It's just a practice to run Fun4All.

**Files**

- Fun4All\_minimum.C

**Step2**

It shows how to add your own analysis module.

**Files/Directories**

- Fun4All\_minimum\_2.C
- sample\_module\_2

**Step3**

See INTT hits from MC events.

**Files/Directories**

- Fun4All\_minimum\_3.C
- sample\_module3

**Step4**

A DST file is read, and information of INTT...

**Sample 2**

This analysis module just prints words on your terminal like:

```
[nukazuka@spinx05 09:49:10 Fun4All_samples] $ root -q -b Fun4All_minimum_2.C ; echo $?
```

```
Processing Fun4All_minimum_2.C...
tutorial::tutorial(const std::string &name) Calling ctor
tutorial::Init(PHCompositeNode *topNode) Initializing
Fun4AllServer::setRun(); run 0 uses CDB TIMESTAMP 0
tutorial::InitRun(PHCompositeNode *topNode) Initializing for Run XXX
```

List of Nodes in Fun4AllServer:

```
Node Tree under TopNode TOP
TOP (PHCompositeNode)/
  DST (PHCompositeNode)/
    RLM (PHCompositeNode)/
      PAR (PHCompositeNode)/
```

```
tutorial::process_event(PHCompositeNode *topNode) Processing Event
tutorial::ResetEvent(PHCompositeNode *topNode) Resetting internal structures, prepare for
tutorial::EndRun(const int runnumber) Ending Run for Run 0
tutorial::End(PHCompositeNode *topNode) This is the End...
tutorial::Reset(PHCompositeNode *topNode) being Reset
tutorial::~tutorial() Calling dtor
```

```
[nukazuka@spinx05 09:49:10 Fun4All_samples] $ root -q -b Fun4All_minimum_2.C ; echo $?
```

README is a commonly used text file to tell what's inside. I put README.md in each directory for the explanation for the directory. It can be your hint. Read it!

# 韓国ワークショップのおさらい

This talk presents how to run your analysis codes in the Fun4All framework. Audiences are asked to download/run/change some codes, so you need to have a BNL account.

## 実習

1. サンプルコードのダウンロード
2. 環境変数の確認
3. 最小コード (Fun4All\_minimum.C) を動かしてみる
4. 自分の解析モジュールを作ってみる
5. 環境変数を書き換え、自分の解析モジュールを使えるようにする
6. サンプルコードを動かしてみる
  - 6.1. 自分の解析モジュールを動かす (Fun4All\_minimum\_2.C)
  - 6.2. InttRawHit の解析 (Fun4All\_minimum\_3.C)
  - 6.3. TrkrCluster の解析 (Fun4All\_minimum\_4.C)
- + 宿題

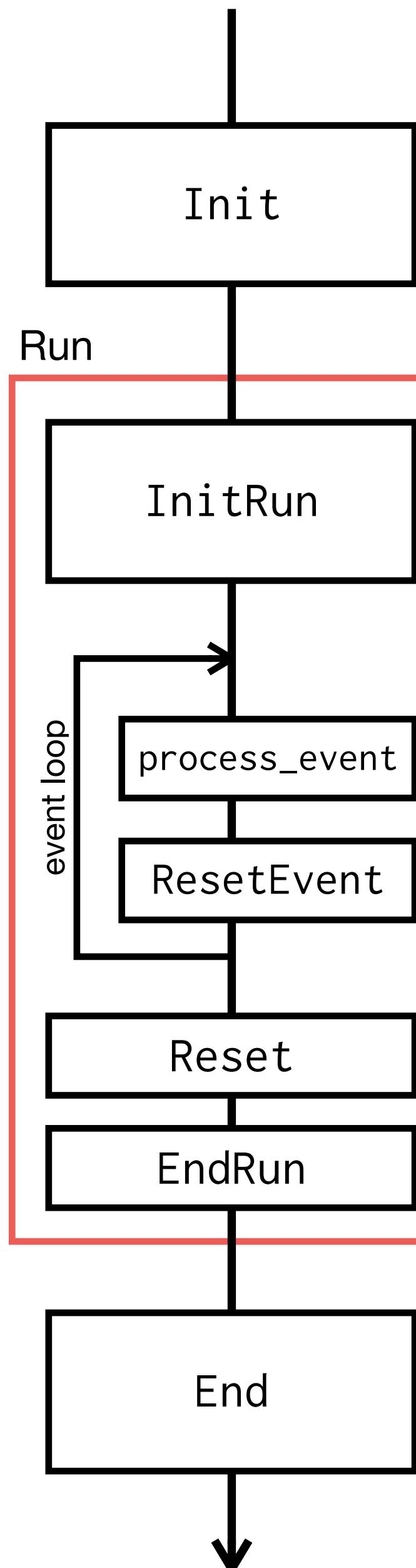
# sample 4

# Sample4: Analyzing TrkrCluster

Fun4All\_minimum\_4.C

```

29 int Fun4All_minimum_4(
30     int nEvents = 10,
31     const string &data = "/sphenix/lustre01/sphnxpro/physics/slurp/tr
32     )
33 {
34
35     Fun4AllServer *se = Fun4AllServer::instance();
36     //se->Verbosity(0);
37
38     // Read DST
39     Fun4AllInputManager *in = new Fun4AllDstInputManager("DSTin");
40     in->fileopen( data );
41     // in->AddListFile(inputfile); // to read a list of files, use it. A path to the text
42     se->registerInputManager( in );
43
44     // Flag Handler is always needed to read flags from input (if used)
45     // and update our rc flags with them. At the end it saves all flags
46     // again on the DST in the Flags node under the RUN node
47     FlagHandler *flag = new FlagHandler();
48     se->registerSubsystem(flag);
49     Enable::CDB = true;
50     // global tag
51
52     recoConsts *rc = recoConsts::instance();
53     rc->set_StringFlag("CDB_GLOBALTAG", CDB::global_tag);
54     // 64 bit timestamp
55     rc->set_uint64Flag("TIMESTAMP", CDB::timestamp);
56     rc->set_IntFlag("RUNNUMBER", 0 );
57
58     /////////////////////////////////
59     // Something depends on Acts should be below....
60     ///////////////////////////////
61     // central tracking
62     Enable::MVTX
63     Enable::TPC
64     Enable::MICROMEGAS
65     Enable::INTT
66     Enable::BLACKHOLE
67     G4MAGNET::magfield_rescale = 1.4;
68
69     // Initialize the selected subsystems
70     // G4Init();
71
72     // GEANT4 Detector description
73     // if (!Input::READHITS)
74     // {
75     //     G4Setup();
76     // }
77
78     TrackingInit(); // necessary for ActsGeometry
79 }
```

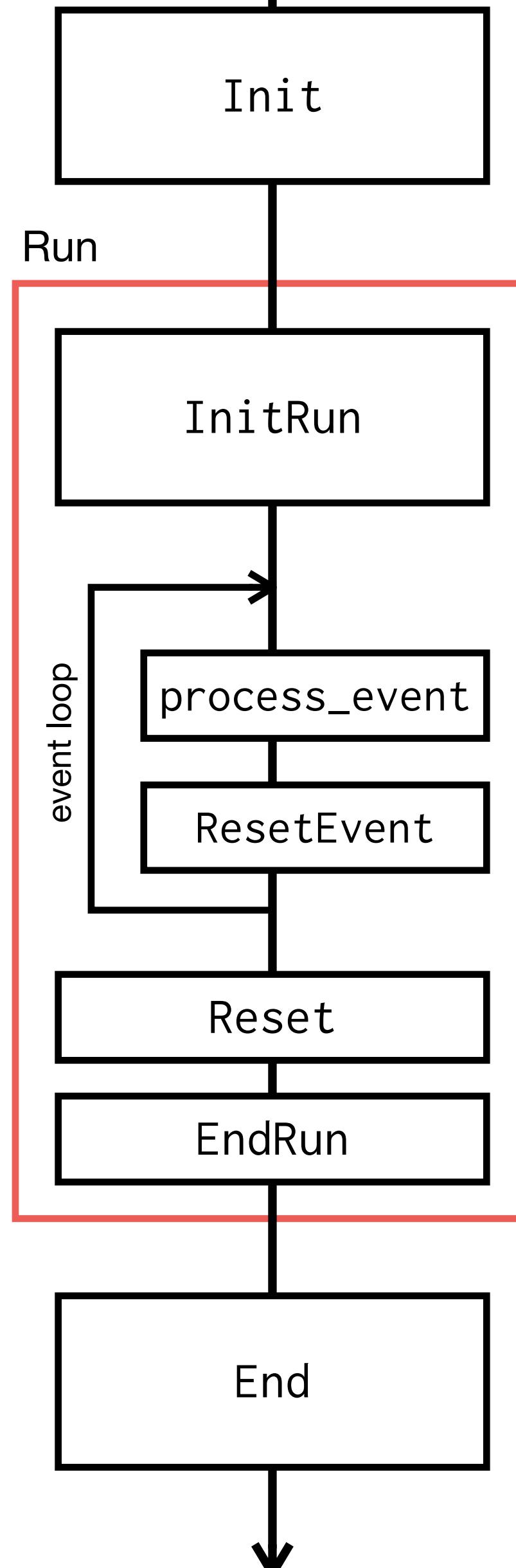


```

34 class tutorial : public SubsysReco
35 {
36     public:
37
38     tutorial(const std::string &name = "tutorial");
39
40     ~tutorial() override;
41
42     int Init(PHCompositeNode *topNode) override;
43
44     int InitRun(PHCompositeNode *topNode) override;
45
46     int process_event(PHCompositeNode *topNode) override;
47
48     /// Clean up internals after each event.
49     int ResetEvent(PHCompositeNode *topNode) override;
50
51     /// Called at the end of each run.
52     int EndRun(const int runnumber) override;
53
54     /// Called at the end of all processing.
55     int End(PHCompositeNode *topNode) override;
56
57     /// Reset
58     int Reset(PHCompositeNode * /*topNode*/) override;
59
60     void Print(const std::string &what = "ALL") const override;
61
62     //! You can set the name of the output file, otherwise it's tutorial_sample4.root
63     void SetOutputPath( std::string path ){ output_path_ = path; }
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95 }
```

## sample 4

# Sample4: Analyzing TrkrCluster: sample\_module\_4/tutorial.cc



### process\_event

```
110 // analysis codes for INTT clusters are written in the function below  
111 this->cluster_analysis( topNode, node_cluster_map, nodeActs );  
112  
113 // Fill event-base TTree at the end of event process  
114 treeEvent_->Fill();  
115 return Fun4AllReturnCodes::EVENT_OK;  
116 }
```

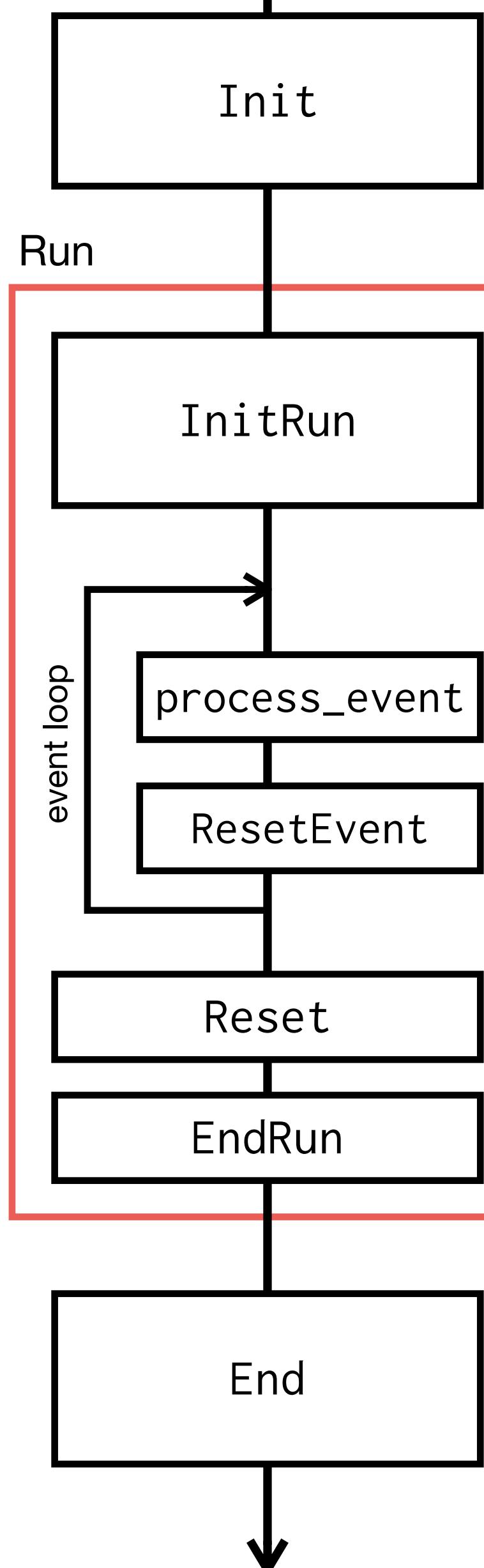
```
118 int tutorial::cluster_analysis(PHCompositeNode *topNode, TrkrClusterContainerv4* node_cluster_map, ActsGeometry* nodeActs )  
119 {  
120  
121  
122 // loop over all INTT layers (0: inner of inner barrel, 1: outer of inner, 2: inner of outer, 3: outer of outer)  
123 for (unsigned int inttlayer = 0; inttlayer < 4; inttlayer++) ← For loop over 4 INTT layers  
124 {  
125  
126 // get clusters only on the INTT layer, and loop over them  
127 for (const auto &hitsetkey : node_cluster_map->getHitSetKeys(TrkrDefs::TrkrId::inttId, inttlayer + 3) )  
128 {  
129  
130 // #cluster counters  
131 cluster_num++; // all of them  
132 cluster_num_layer_[inttlayer]++;
    } // for each layer  
133  
134 // type: std::pair<ConstIterator, ConstIterator> ConstRange  
135 // here, MMap_::const_iterator ConstIterator;  
136 auto range = node_cluster_map->getClusters(hitsetkey); TrkrClusterContainer  
137 // loop over iterators of this cluster  
138 for (auto clusIter = range.first; clusIter != range.second; ++clusIter)  
139 {  
140  
141     const auto cluskey = clusIter->first; cluster key  
142     const auto cluster = clusIter->second; TrkrCluster  
143 }
```

Annotations on the right side of the code:

- A red arrow points from the line 'this->cluster\_analysis( topNode, node\_cluster\_map, nodeActs );' to the first call to 'cluster\_analysis' in the 'tutorial' class definition.
- A red bracket labeled 'For loop over 4 INTT layers' spans the entire 'for' loop from line 123 to line 124.
- A red bracket labeled 'For loop over clusters on this INTT layer' spans the inner 'for' loop from line 127 to line 128.
- A red curly brace labeled 'Counting #cluster' spans the two lines of code that increment 'cluster\_num' and 'cluster\_num\_layer\_'.
- A red arrow points from the line 'auto range = node\_cluster\_map->getClusters(hitsetkey);' to the explanatory text '→ pair or cluster key & TrkrCluster'.

## sample 4

# Sample4: Analyzing TrkrCluster: sample\_module\_4/tutorial.cc



```
145 // Get cluster position in lab-coordinate using Acts
146 const auto globalPos = nodeActs->getGlobalPosition(cluskey, cluster);
147
148 // Set cluster position in lab-coordinate to t ↑ Getting cluster position in the lab frame by using Acts geometry.
149 cluster->setPosition(0, globalPos.x());
150 cluster->setPosition(1, globalPos.y());
151 cluster->setPosition(2, globalPos.z());
152
153 // Assign cluster parameters
154 position_[0] = cluster->getPosition( 0 ); // x
155 position_[1] = cluster->getPosition( 1 ); // y
156 position_[2] = cluster->getPosition( 2 ); // z
157 adc_ = cluster->getAdc();
158 size_phi_ = cluster->getPhiSize();
159
160 /** @TODO Calculate phi, theta, eta (pseudorapidity) by yourself
161     phi_ = 0; // (radian)
162     theta_ = 0; // (radian)
163     eta = 0; // pseudorapidity
164  */
165
166 // After getting all cluster parameters, let's fill them
167 treeCluster->Fill(); ← Don't forget to fill cluster-base TTree
168
169 // Then, reset the parameters (it's not mandatory if all parameters are available all the time. It's just in case)
170 this->ResetClusterLoop();
```

↑  
The cluster variables are used to fill the cluster-base TTree and reused. If a value cannot be obtained for a cluster, the parameter value for the previous cluster is used. It's good to assign a default value before reusing.

Getting cluster position in the lab frame by using Acts geometry.  
It's necessary because cluster key & TrkrCluster have only position in the INTT ladder.

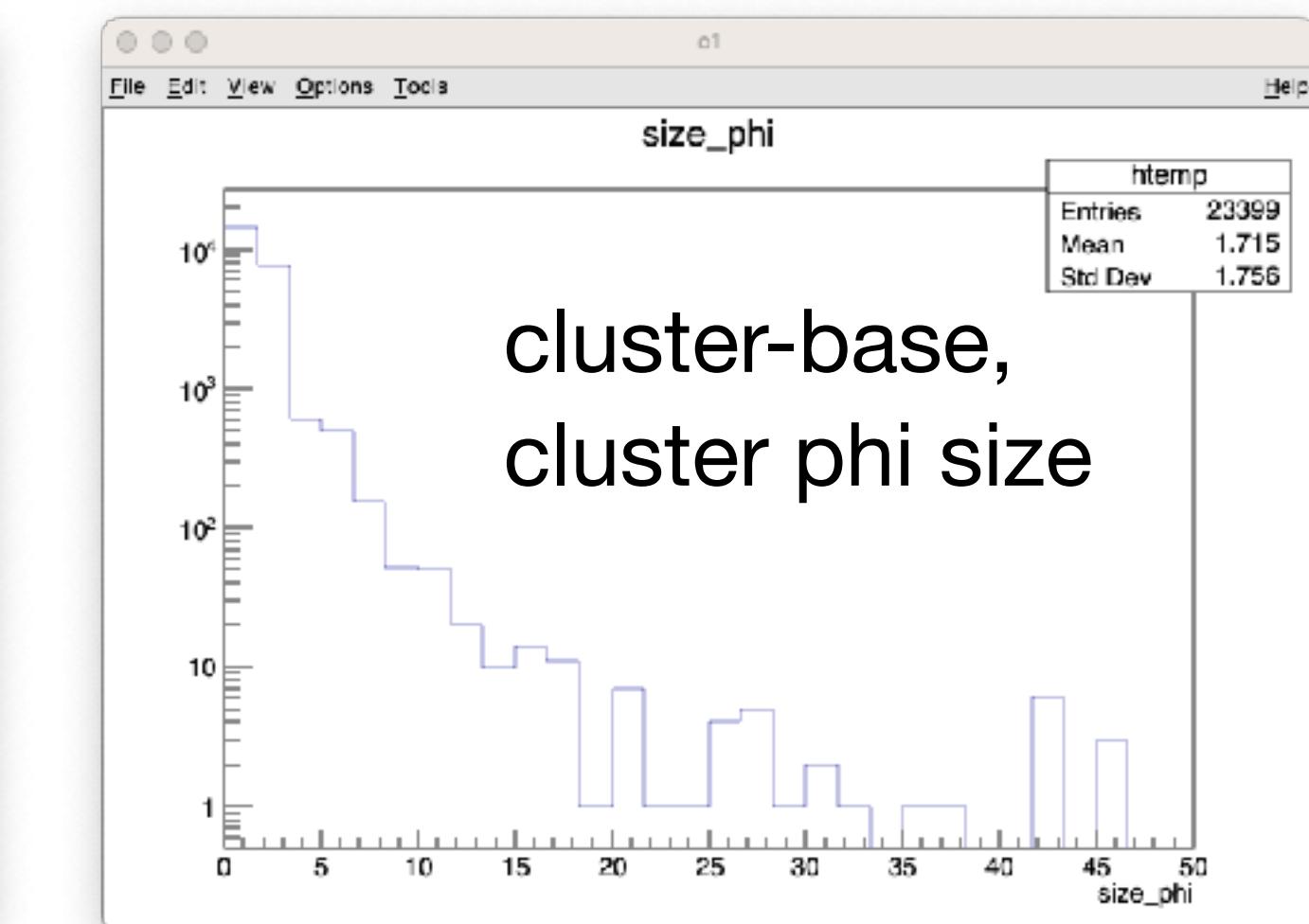
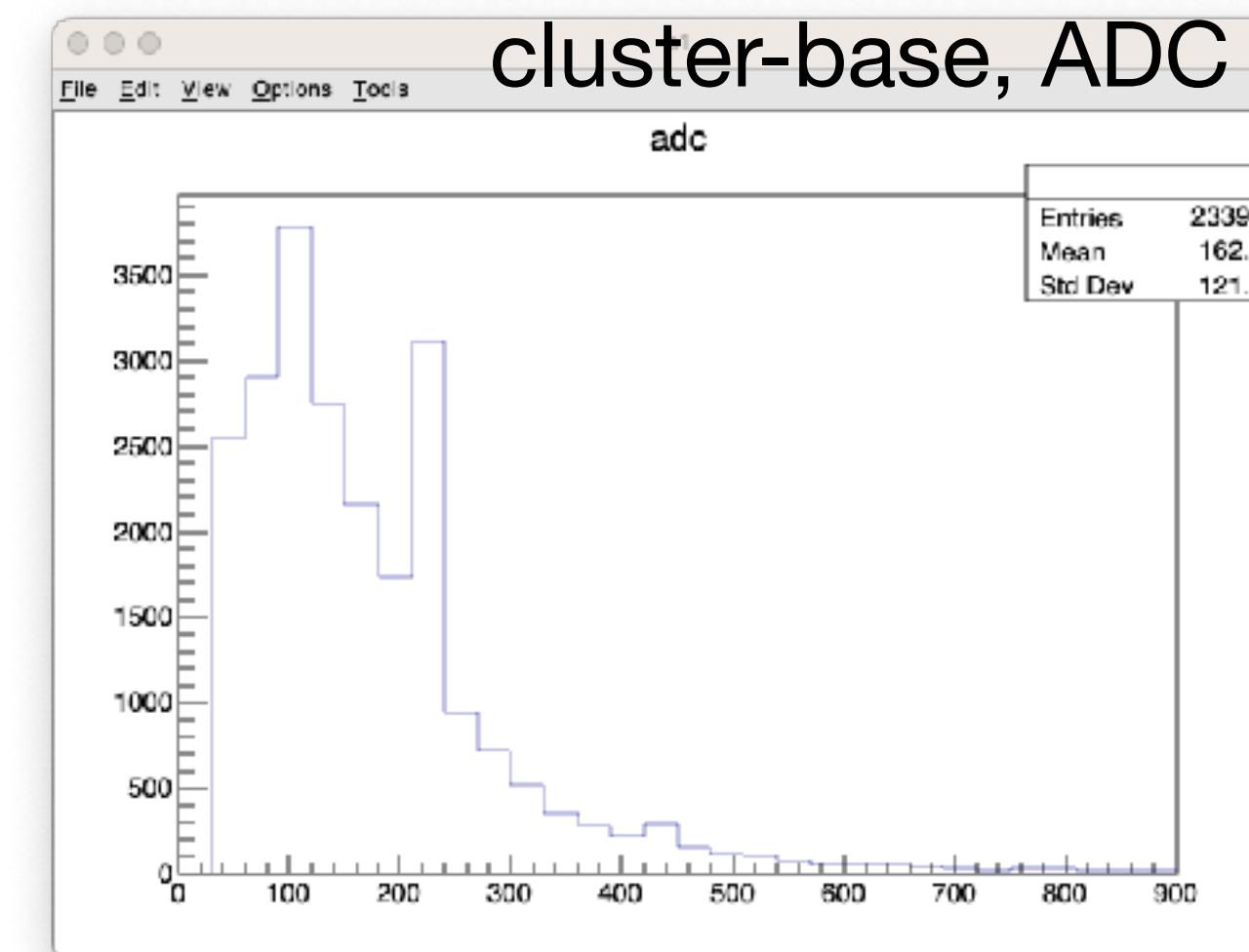
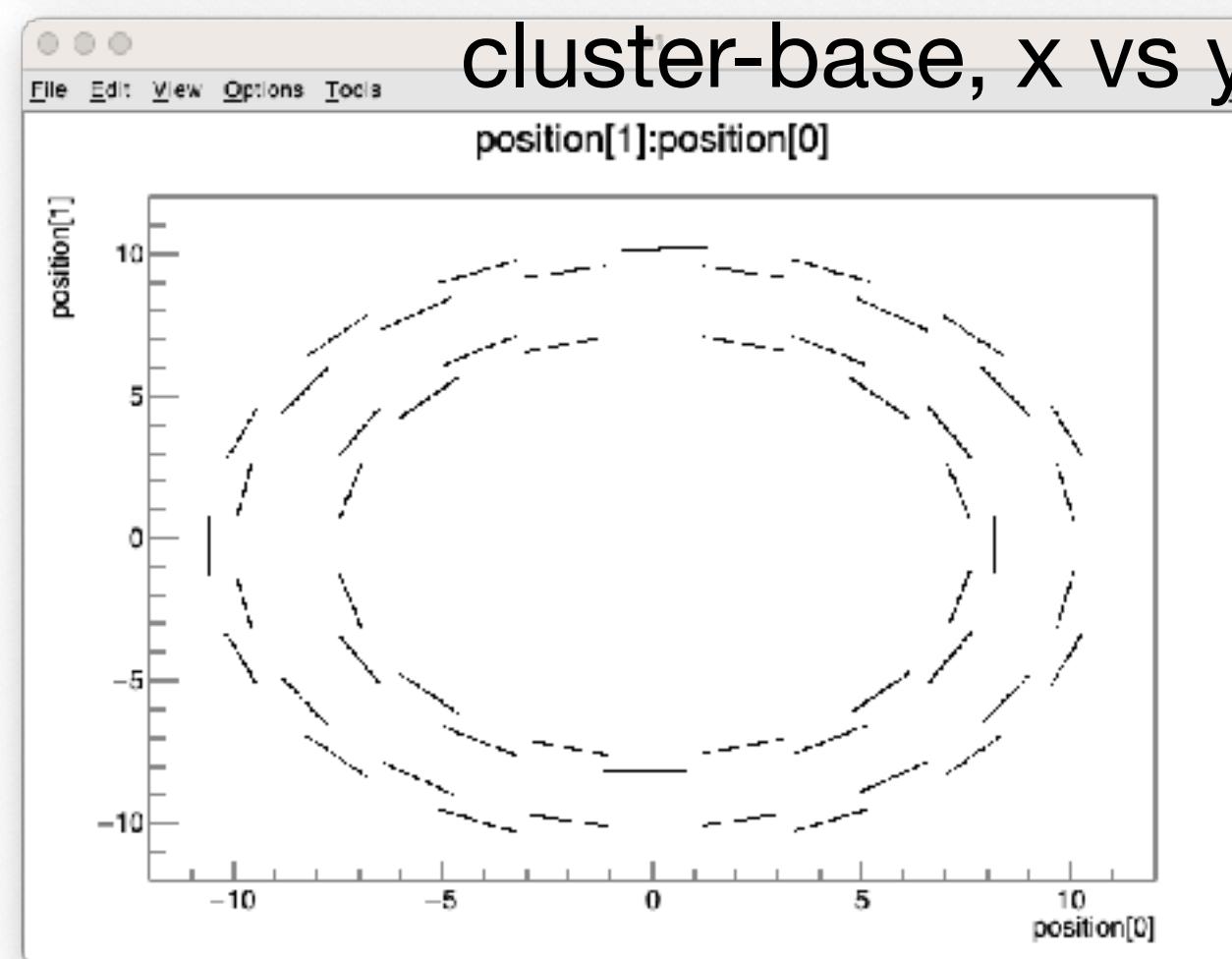
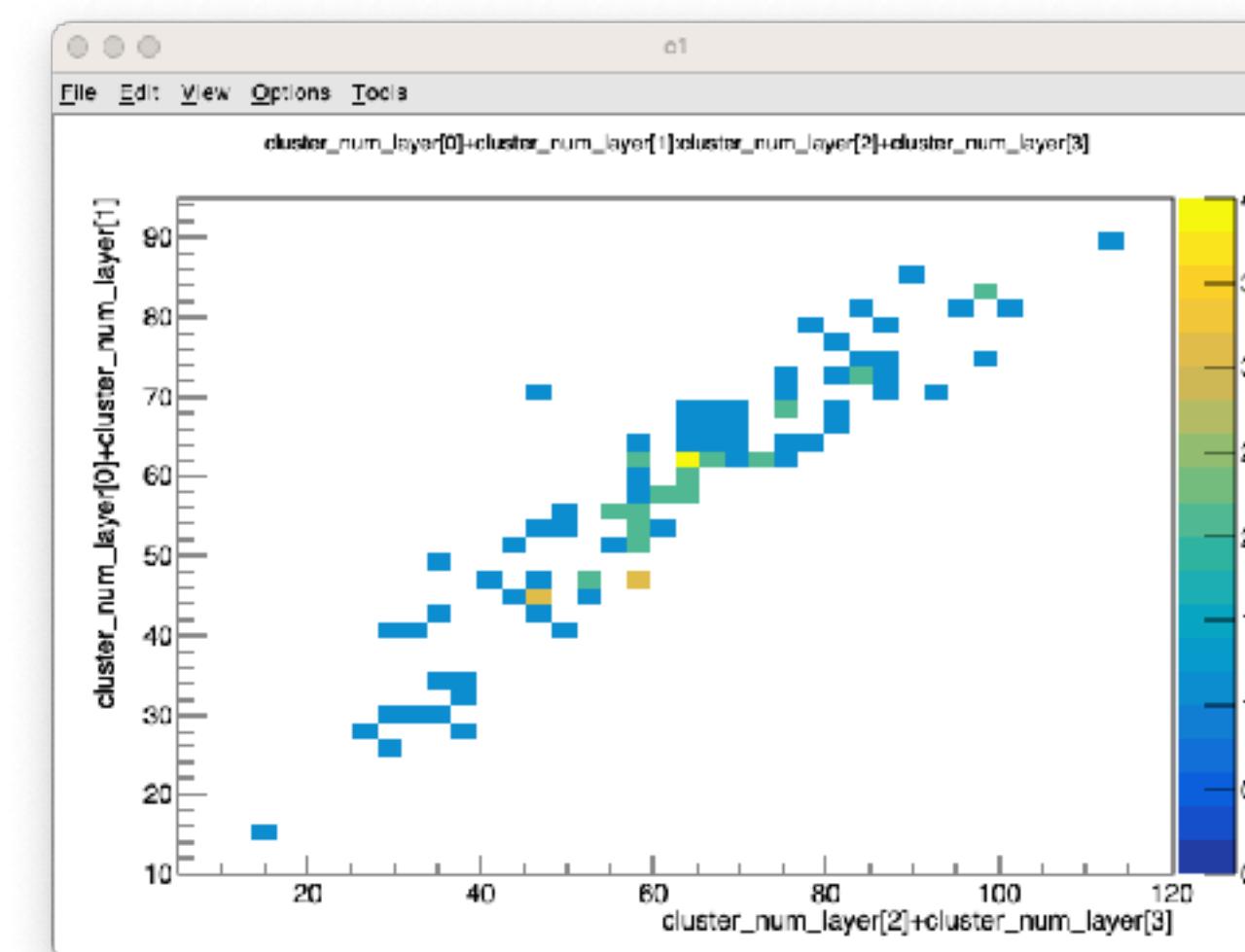
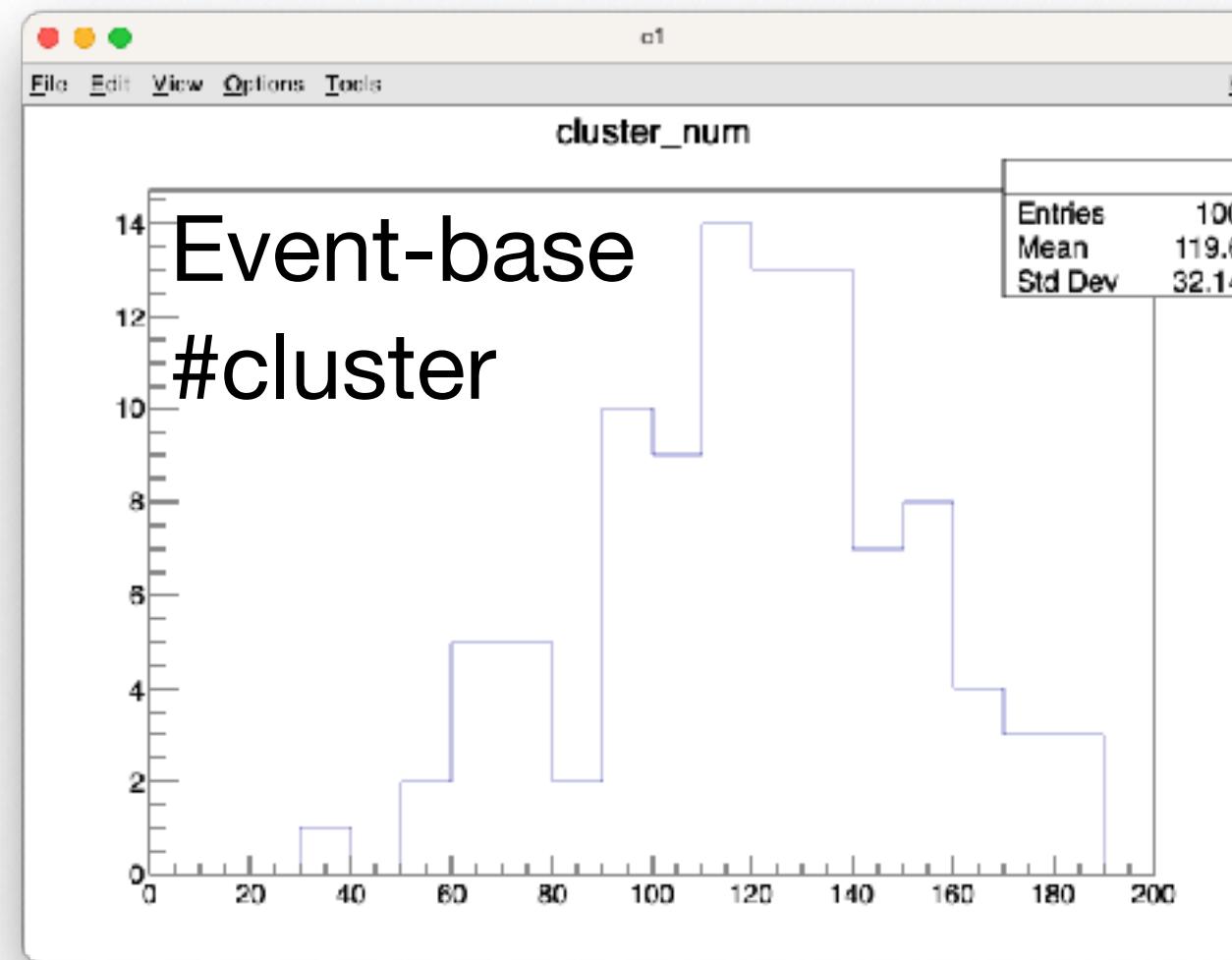
Getting cluster information and assigning them to the variables.

## sample 4

# Analysis module #4: Hands on

Let's try

### 5. Draw some histograms



**HANDS ON!**  
**#8**

## Analysis module #4: Hands on

Let's try

5. Draw some histograms (hint)
  - 5.1. NoMachine user: just run root command
  - 5.2. VS code user: look ROOT file in VS code
  - 5.3. terminal user: send ROOT file with scp command

Example: `$ scp sphnx03:/sphenix/u/nukazuka/work_now/tutorial_sample4.root .`

**HANDS ON!**  
**#8**

# Analysis module #4: Homework

- Learn class inheritance in C++.
- Learn polymorphism.
- Learn the environment variable LD\_LIBRARY\_PATH
- Complete cluster  $\phi$ ,  $\theta$ ,  $\eta$  calculation. If some more information is needed for the calculation, just assume the simplest case.

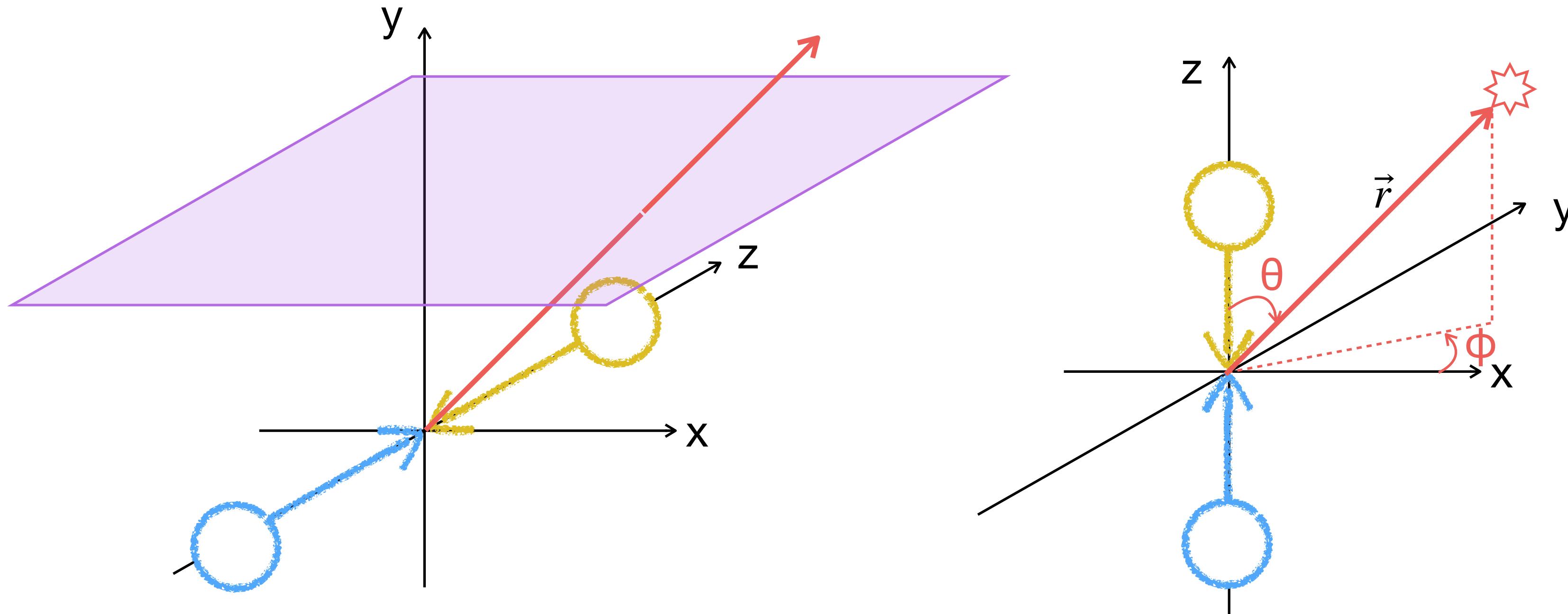
```
152
153     // Assign cluster parameters
154     position_[0] = cluster->getPosition( 0 ); // x
155     position_[1] = cluster->getPosition( 1 ); // y
156     position_[2] = cluster->getPosition( 2 ); // z
157     adc_ = cluster->getAdc();
158     size_phi_ = cluster->getPhiSize();
159
160     /** @TODO Calculate phi, theta, eta (pseudorapidity) by yourself
161      phi_ = 0; // (radian)
162      theta_ = 0; // (radian)
163      eta = 0; // pseudorapidity
164 */
165
```

# Analysis module #4: Homework

```

152
153     // Assign cluster parameters
154     position_[0] = cluster->getPosition( 0 ); // x
155     position_[1] = cluster->getPosition( 1 ); // y
156     position_[2] = cluster->getPosition( 2 ); // z
157     adc_ = cluster->getAdc();
158     size_phi_ = cluster->getPhiSize();
159
160     /** @TODO Calculate phi, theta, eta (pseudorapidity) by yourself
161      phi_ = 0; // (radian)
162      theta_ = 0; // (radian)
163      eta = 0; // pseudorapidity
164
165

```



$$\begin{cases} x = r \sin \theta \cos \phi \\ y = r \sin \theta \sin \phi \\ z = r \cos \theta \end{cases}$$

$$\vec{r} = (x, y, z)$$

$$r = |\vec{r}|$$

$$\theta = \arccos \frac{z}{r}$$

$$\phi = \arctan \frac{y}{x}$$

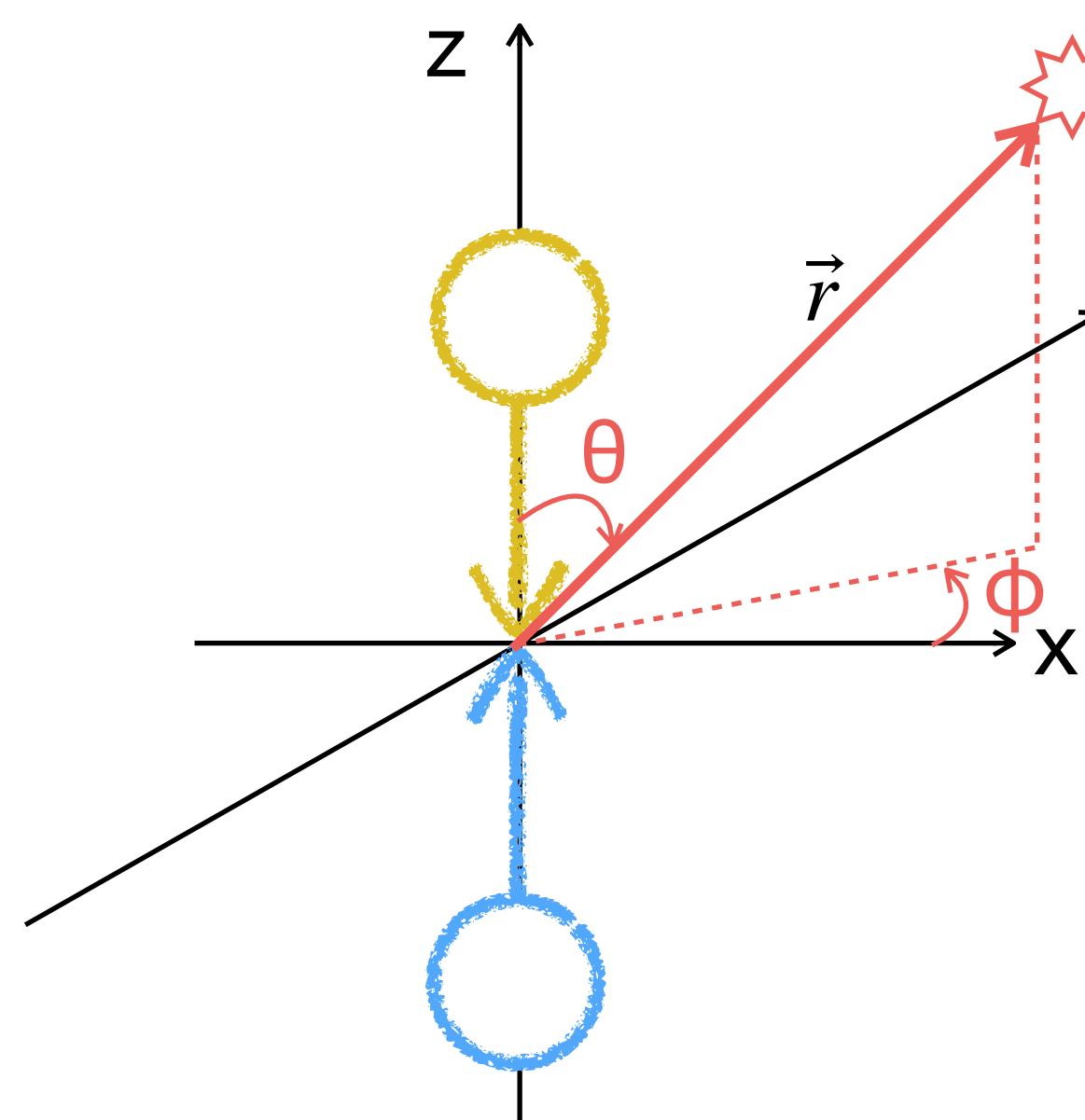
$$\eta = -\ln \left( \tan \frac{\theta}{2} \right)$$

# Analysis module #4: Homework 再考

```

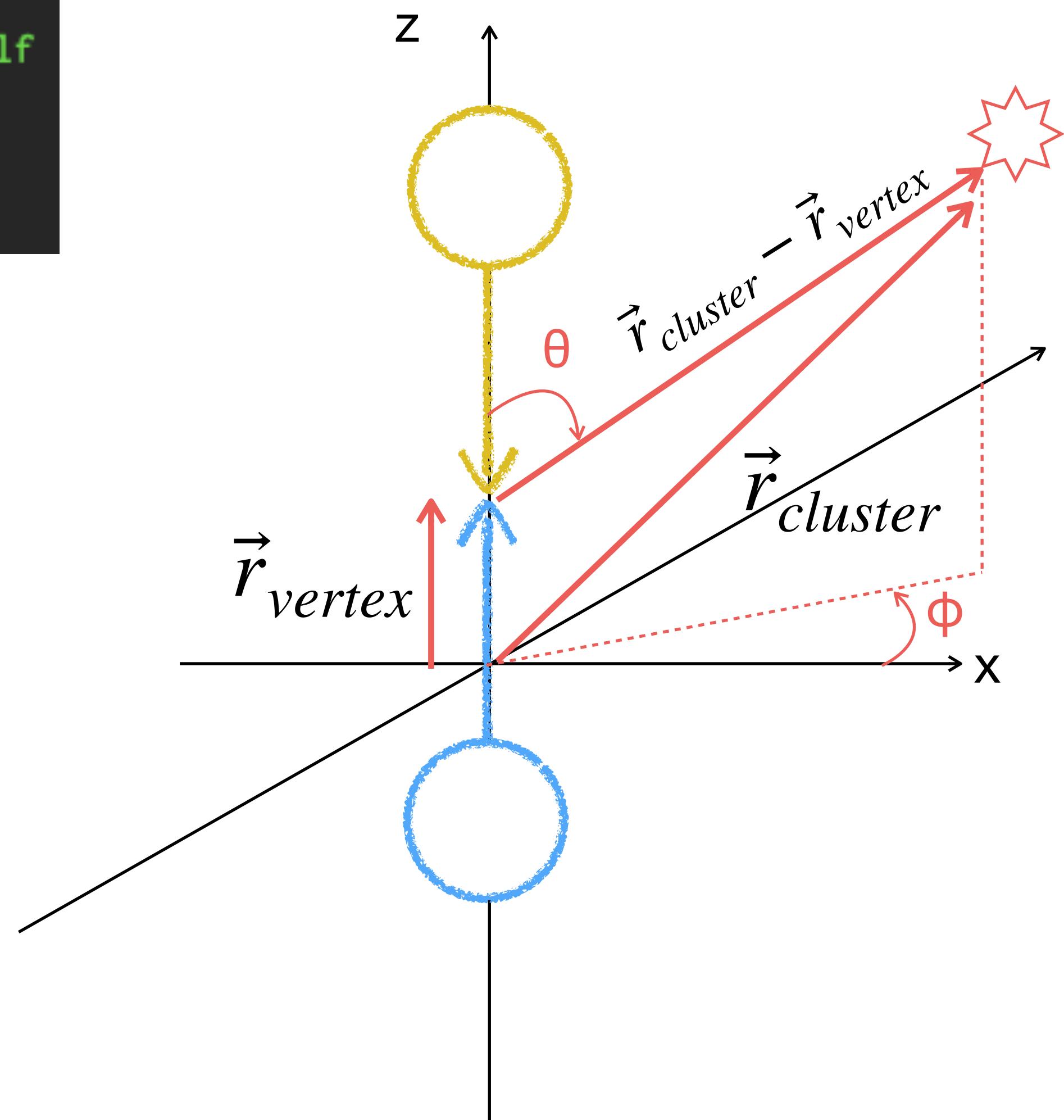
160
161      /** @TODO Calculate phi, theta, eta (pseudorapidity) by yourself
162      TVector3 vec( position_[0],position_[1], position_[2] );
163      phi_ = vec.Phi();
164      theta_ = vec.Theta();
165      eta_ = vec.Eta();

```



このような想定で計算したが

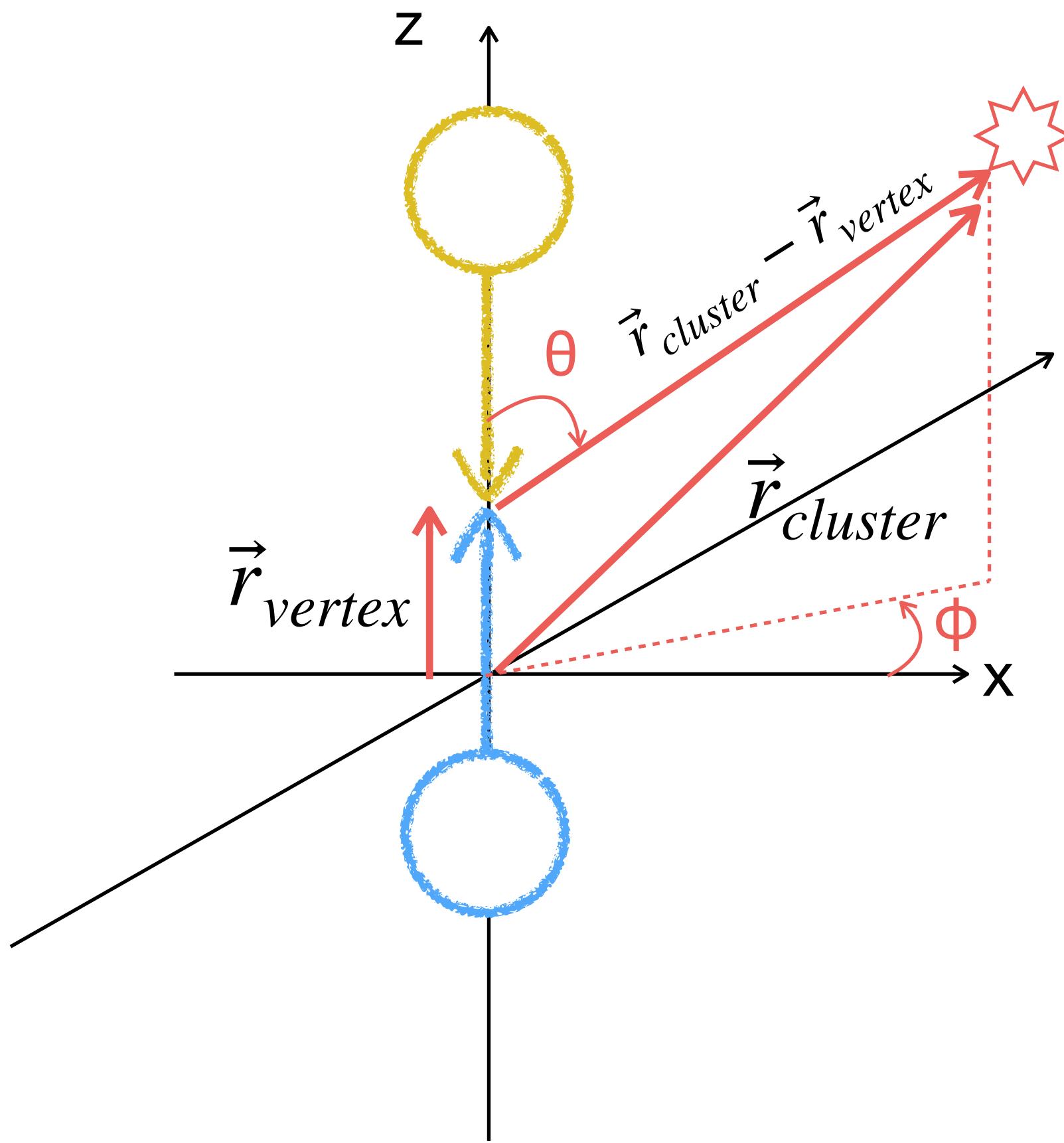
衝突は必ずしも座標原点で  
起こるとは限らない



$\vec{r}_{cluster} - \vec{r}_{vertex}$ について  
 $\theta, \phi, \eta$ 求めないと意味がない

# Analysis module #4: Homework 再考

```
160  
161      /** @TODO Calculate phi, theta, eta (pseudorapidity) by yourself  
162      TVector3 vec( position_[0],position_[1], position_[2] );  
163      phi_ = vec.Phi();  
164      theta_ = vec.Theta();  
165      eta_ = vec.Eta();
```



$\vec{r}_{vertex}$  (衝突 vertex の位置) はどうやって求める？

1. MBD を使う (z 方向のみ)
2. INTT を使う (3 次元)
3. SVTX (MVTX + INTT + ?) を使う (3 次元, 精密)
4. SVTX + MBD を使う (こういうものがあるらしい)

MBD データも合わせた解析をやってみよう

# Official DSTs

See [here](#) for more details

/sphenix/lustre01/sphnxpro/

- bbox3
- db\_bkup
- testbed
- beam
- m1
- bbox1
- mlp
- cosmics
- 1008\_backup
- **physics**
- bbox5
- bbox2
- slurp
- calib
- slurptest
- mdc2
- tchou
- HPSS\_Status
- run1auau\_pi0\_calib
- t1044
- bbox4
- production
- liumigrate
- bbox0
- **commissioning**
- scratch
- sdcc

→ /sphenix/lustre01/sphnxpro/physics

- tpc
- INTT
- GL1
- run2pp
- mbd
- ZDC
- MVTX
- **slurp**
- online\_monitoring
- run2auau
- mbdcalib
- TPOT
- LL1
- emcal
- HCal
- TEST

[SLURP](#) (sPHENIX Lightweight Utilities for Realtime Production )

/sphenix/lustre01/sphnxpro/physics/slurp/

- calocosmics
- jetproduction
- cosmics
- caloy2test
- tpccosmics
- caloy2calib
- **streaming**
- cosmics
- physics
- fast
- fast\_tracking
- calobeam
- calophysics
- **tracking**
- run\_00050900\_00051000
- run\_...
- caloy2fitting
- tpccalib
- tpcbeam
- junkdrawer
- TEST

Data taken in the early phase were processed, and their DSTs are put here.

# MBD データはどこ？

トラッキング検出器: MVTX, INTT, TPC, TPOT

カロリメータ（系）: EMCAL, HCal, MBD, sEPD, ZDC

\* MBD, sEPD はカロリメータではないが、カロリメータに分類されている

```
/sphenix/lustre01/sphnxpro/physics/slurp
└── calobeam          : Run type が beam のカロリメータの raw hit (?) データ
└── calocosmics        : カロリメータの宇宙線データ
└── calophysics        : Run type が physics のカロリメータの raw hit (?) データ
└── caloy2calib         :
└── caloy2fitting       :
└── cosmics             : 宇宙線データ
└── jetproduction        : Jet 解析のためのデータ
└── single_streaming    : トラッキング検出器ごとの raw hit データ
└── streaming            : 全トラッキング検出器をまとめた raw hit データ
└── tpccosmics          : TPC の宇宙線データ
└── tracking             : 全トラッキング検出器の TrkrHit, TrkrCluster, seed, etc.
```

# MBD データはどこ？

トラッキング検出器: MVTX, INTT, TPC, TPOT

カロリメータ（系）: EMCAL, HCal, MBD, sEPD, ZDC

\* MBD, sEPD はカロリメータではないが、カロリメータに分類されている

```
/sphenix/lustre01/sphnxpro/physics/slurp
└── streaming : 全トラッキング検出器をまとめた raw hit データ
```

```
/sphenix/lustre01/sphnxpro/physics/slurp/streaming
├── cosmics
│   ├── ana441_2024p007
│   └── ana427_2024p005
└── physics
    ├── ana441_2024p007 (2057 ラン, 235915 ファイル)
    │   ├── run_00050900_00051000
    │   ├── ...
    │   └── new_2024p007 (362 ラン, 87469 ファイル)
    │       ├── run_00054200_00054300
    │       ├── ...
    └── ...
```

\* セグメント番号 00000 を必ず含んでいるとは限らない

```
DST: DST_STREAMING_EVENT_run2auau_new_2024p007-00054279-00000.root
List of Nodes in Fun4AllServer:
Node Tree under TopNode TOP
TOP (PHCompositeNode)/
    DST (PHCompositeNode)/
        GL1 (PHCompositeNode)/
            GL1RAWHIT (IO,GL1Packetv2)
        INTT (PHCompositeNode)/
            INTTRAWHIT (IO,InttRawHitContainerv2)
        MVTX (PHCompositeNode)/
            MVTXRAWEVTHEADER (IO,MvtxRawEvtHeaderv2)
            MVTXRAWHIT (IO,MvtxRawHitContainerv1)
        TPC (PHCompositeNode)/
            TPCRAWHIT (IO,TpcRawHitContainerv2)
        MICROMEGAS (PHCompositeNode)/
            MICROMEGASRAWHIT (IO,MicromegasRawHitContainerv1)
        Sync (IO,SyncObjectv1)
        EventHeader (IO,EventHeaderv1)
    RUN (PHCompositeNode)/
        RunHeader (IO,RunHeaderv1)
        Flags (IO,FlagSavev1)
```

# MBD データはどこ？

トラッキング検出器: MVTX, INTT, TPC, TPOT

カロリメータ（系）: EMCAL, HCal, MBD, sEPD, ZDC

\* MBD, sEPD はカロリメータではないが、カロリメータに分類されている

```
/sphenix/lustre01/sphnxpro/physics/slurp
```

```
└── tracking
```

: 全トラッキング検出器の TrkrHit, TrkrCluster, seed, etc.

```
/sphenix/lustre01/sphnxpro/physics/slurp/tracking/
├── ana441_2024p007          (1032ラン, 17609 ファイル)
│   ├── run_00051500_00051600
│   ├── ...
│   └── ana439_2024p007          (36ラン, 48 ファイル)
│       ├── run_00053700_00053800
│       ├── ...
│       └── ana449_2024p008          (0 ラン, 0 ファイル)
│           ├── run_00051500_00051600
│           ├── ...
```

```
DST: DST_TRKR_CLUSTER_run2pp_ana441_2024p007-00051512-0008.root
List of Nodes in Fun4AllServer:
Node Tree under TopNode TOP
TOP (PHCompositeNode)/
    DST (PHCompositeNode)/
        Sync (IO,SyncObjectv1)
        EventHeader (IO,EventHeaderv1)
    TRKR (PHCompositeNode)/
        TRKR_CLUSTER (IO,PHObject)
        TRKR_CLUSTERCROSSINGASSOC (IO,PHObject)
        LaserEventInfo (IO,PHObject)
        LASER_CLUSTER (IO,PHObject)
    RUN (PHCompositeNode)/
        RunHeader (IO,RunHeaderv1)
        Flags (IO,FlagSavev1)
        CYLINDERGEOM_MVTX (IO,PHObject)
        CYLINDERGEOM_INTT (IO,PHObject)
        CYLINDERCELLGEOM_SVTX (IO,PHObject)
        CYLINDERGEOM_MICROMEAS_FULL (IO,PHObject)
        GEOMETRY_IO (IO,PHObject)
        CdbUrl (IO,CdbUrlSavev1)
    PAR (PHCompositeNode)/
```

\* セグメント番号 00000 を必ず含んでいるとは限らない

# 注意：オフィシャル DST の様子は韓国ワークショップから大きく変わった

保存容量不足のため削除

- Path: /sphenix/lustre01/sphnxpro/physics/slurp/streaming/physics/inttonlyrun\_\*
  - file name: DST\_INTT\_EVENT\_run2pp\_new\_2024p002-XXXXXXX-YYYYY.root
  - Contents:
- Path:/sphenix/lustre01/sphnxpro/physics/slurp/streaming/physics/run\_\*
  - file name: DST\_STREAMING\_EVENT\_run2pp\_new\_2024p002-XXXXXXX-YYYYY.root
  - Contents:  
new\_2024p002 は削除, ana441\_2024p007 と new\_2024p007 が存在
- Path:/sphenix/lustre01/sphnxpro/physics/slurp/tracking/new\_2024p007/run\_\*
  - file name: DST\_TRKR\_HIT\_run2pp\_new\_2024p007-XXXXXXX-YYYYY.root
  - Contents:  
new\_2024p007 は削除,  
ana439\_2024p007 ana441\_2024p007 ana449\_2024p008 が存在
- Path:/sphenix/lustre01/sphnxpro/physics/slurp/tracking/new\_2024p007/run\_\*
  - file name: DST\_TRKR\_CLUSTER\_run2pp\_new\_2024p007-XXXXXXX-YYYYY.root
  - Contents:
- Path:/sphenix/lustre01/sphnxpro/physics/slurp/tracking/run\_\*  
- File name: DST\_TRKR\_SEED\_run2pp\_new\_2024p007-XXXXXXX-YYYYY.root
  - Contents:  
前はこうだった？

# MBD データはどこ？

トラッキング検出器: MVTX, INTT, TPC, TPOT

カロリメータ（系）: EMCAL, HCal, MBD, sEPD, ZDC

\* MBD, sEPD はカロリメータではないが、カロリメータに分類されている

```
/sphenix/lustre01/sphnxpro/physics/slurp
```

```
  |-- calophysics : Run type が physics のカロリメータの raw hit (?) データ
```

```
/sphenix/lustre01/sphnxpro/physics/slurp/calophysics/
  |-- run_00044700_00044800
  |-- ana430_2024p007      : 0 ラン, 0 ファイル
    |-- run_00050900_00051000
    |-- ...
  |-- new_2024p003         : 97 ラン, 8315 ファイル
    |-- run_00051300_00051400
    |-- ...
    |-- run_00051100_00051200
  |-- new_2024p007         : 1 ラン, 1578 ファイル
    |-- run_00051100_00051200
```

\* セグメント番号 00000 を必ず含んでいるとは限らない

```
DST: DST_TRIGGERED_EVENT_run2pp_new_2024p003-00050508-00000.root
List of Nodes in Fun4AllServer:
Node Tree under TopNode TOP
TOP (PHCompositeNode)/
  DST (PHCompositeNode)/
    GL1 (PHCompositeNode)/
      GL1Packet (IO,GL1Packetv2)
    MBD (PHCompositeNode)/
      MBDPackets (IO,CaloPacketContainerv1) ← これ?
    ZDC (PHCompositeNode)/
      ZDCPackets (IO,CaloPacketContainerv1)
    SEPD (PHCompositeNode)/
      SEPD_packets (IO,CaloPacketContainerv1)
    HCAL (PHCompositeNode)/
      HCALPackets (IO,CaloPacketContainerv1)
    CEMC (PHCompositeNode)/
      CEMCPackets (IO,CaloPacketContainerv1)
    Sync (IO,SyncObjectv1)
    EventHeader (IO,EventHeaderv1)
  RUN (PHCompositeNode)/
    RunHeader (IO,RunHeaderv1)
    Flags (IO,FlagSavev1)
  PAR (PHCompositeNode)/
```

# MBD データはどこ？

```
/sphenix/lustre01/sphnxpro/physics/slurp/calophysics/
├── run_00044700_00044800
├── ana430_2024p007      : 0 ラン, 0 ファイル
│   ├── run_00050900_00051000
│   ├── ...
└── new_2024p003          : 97 ラン, 8315 ファイル
    ├── run_00051300_00051400
    ├── ...
    └── run_00051100_00051200 } ←
└── new_2024p007          : 1 ラン, 1578 ファイル
    └── run_00051100_00051200
```

```
/sphenix/lustre01/sphnxpro/physics/slurp/tracking/
├── ana441_2024p007      (1032ラン, 17609 ファイル)
│   ├── run_00051500_00051600
│   ├── ...
│   └── run_00051800_00051900
└── ana439_2024p007      (36ラン, 48 ファイル)
    ├── run_00053700_00053800
    ├── ...
    └── ana449_2024p008      (0 ラン, 0 ファイル)
        ├── run_00051500_00051600
        ├── ...
        └── ...
```

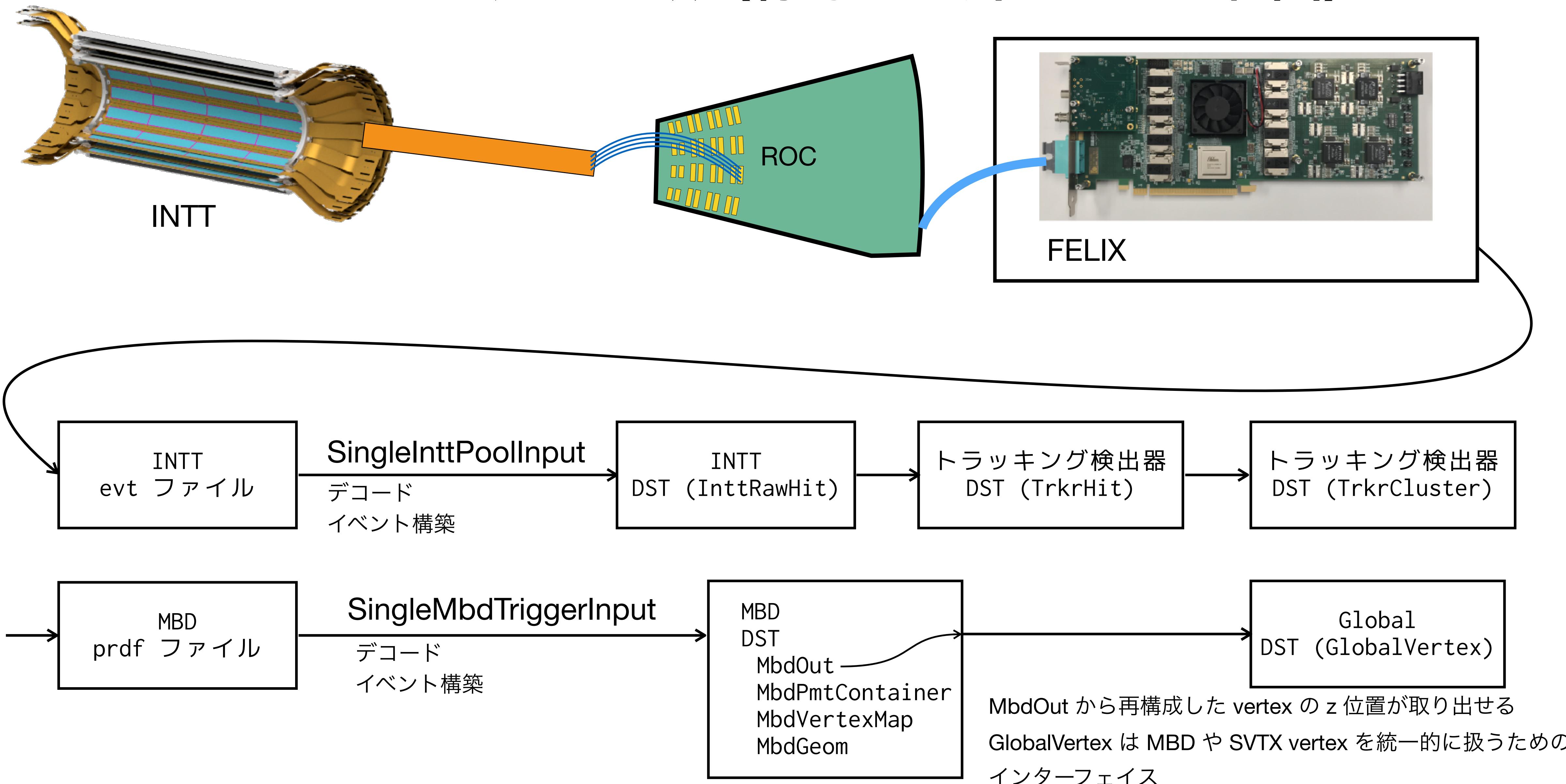
Official DST の tracking/ana441\_2024p007 と calophysics/ana430\_2024p007 は  
共通するラン番号が今のところ存在しない 😱

解1：プロダクショングループに DST 生成を要請する ←これが正しい手順だが、時間がかかる

解2：INTT か MBD の raw hit レベルの処理から自分で行う ← 非エキスパートはあまりやらないほうが安全

解3：今回は教育目的なので MC データで同じようなことをしてごまかす ← 仕方がないのでこれ

# MBD データの処理手順 (糠塚の理解している範囲)



sample 5

## INTT + MBD でデータ解析

INTT + MBD とはどういう意味か？

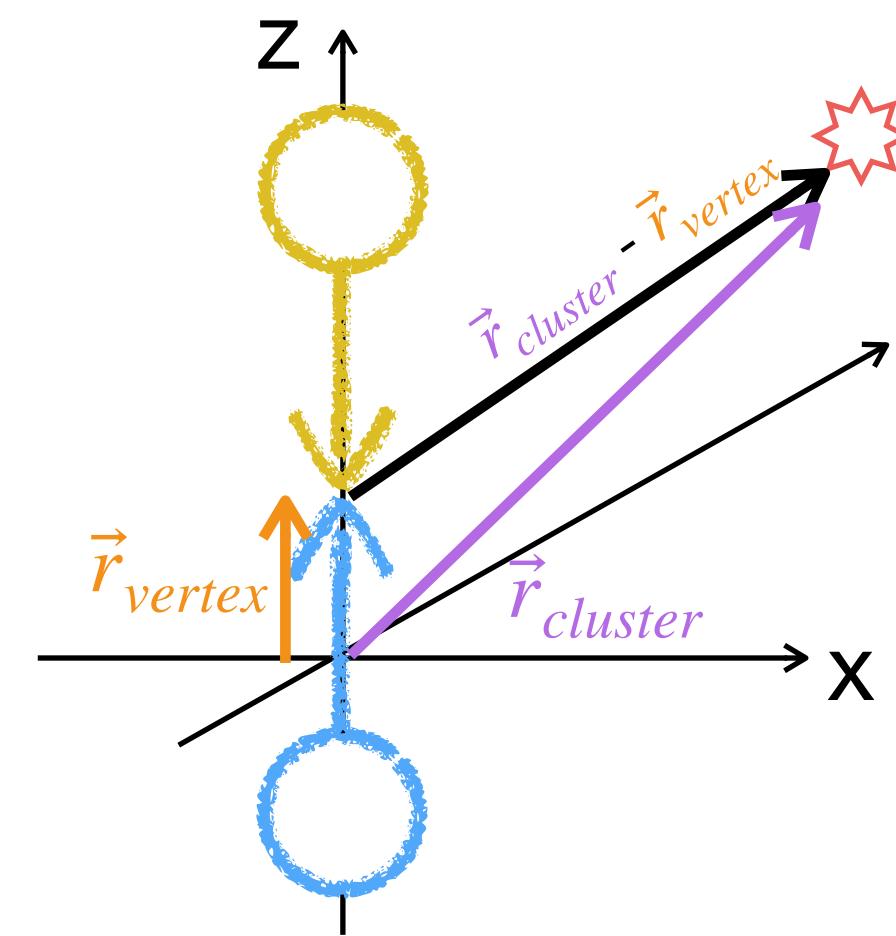
## sample 5

# INTT + MBD でデータ解析

INTT + MBD とはどういう意味か？

INTT 単体解析

INTT のデータのみを扱う



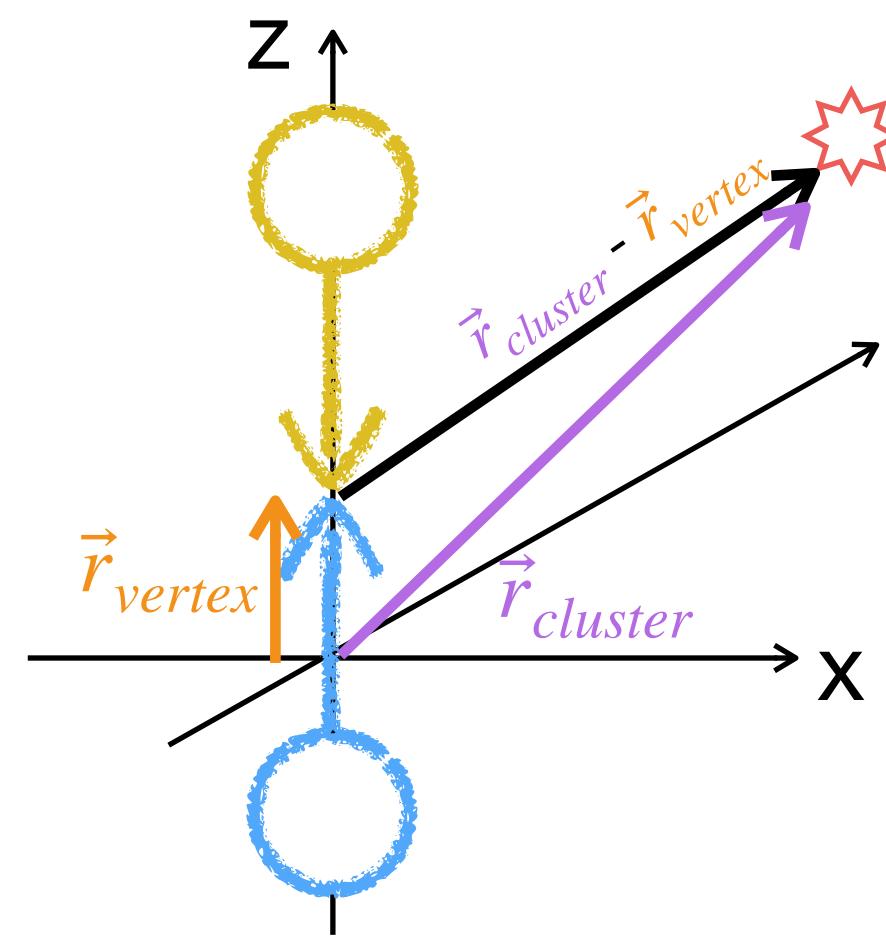
```
TrkrCluster_DST/  
└── TTree  
    ├── event0  
    ├── event1  
    ├── event2  
    ├── event3  
    ├── event4  
    └── event5
```

結果

## sample 5

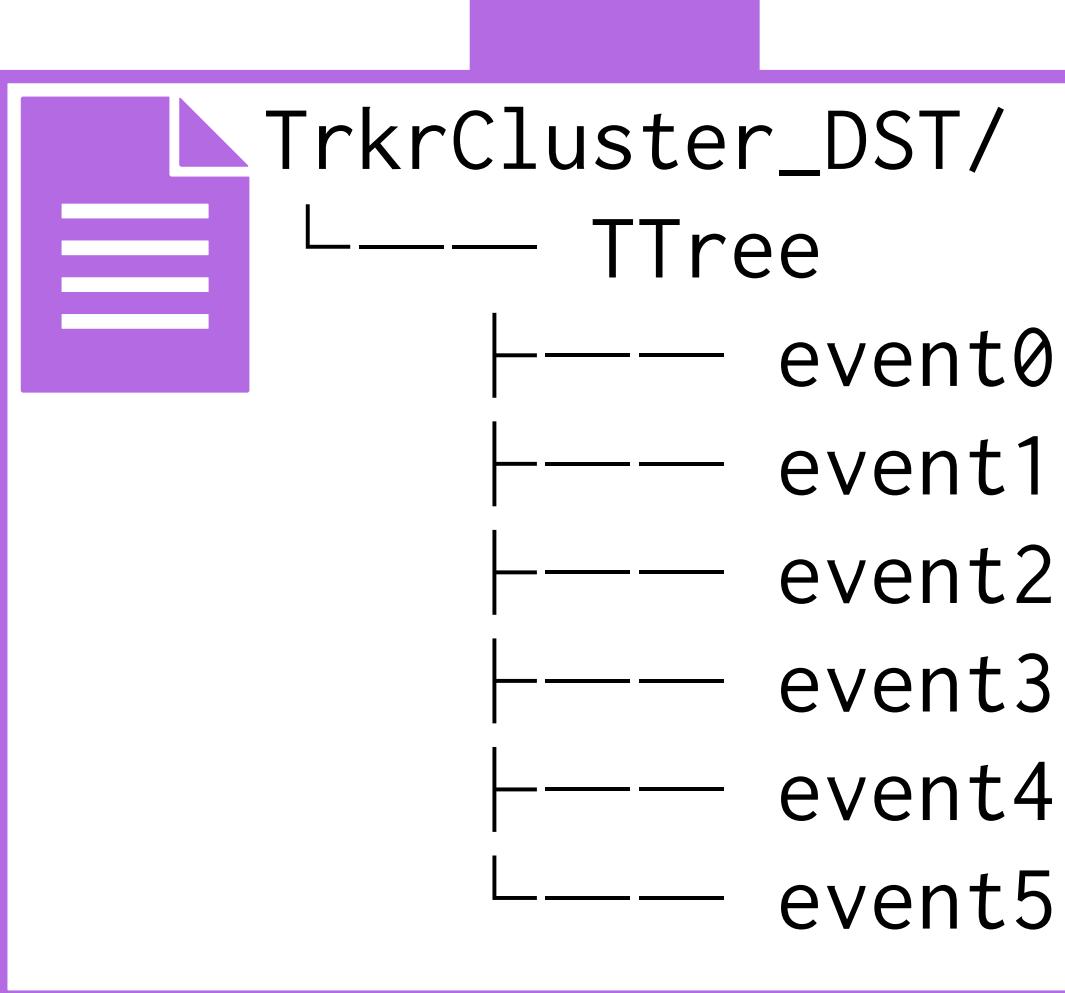
# INTT + MBD でデータ解析

INTT + MBD とはどういう意味か？



### INTT 単体解析

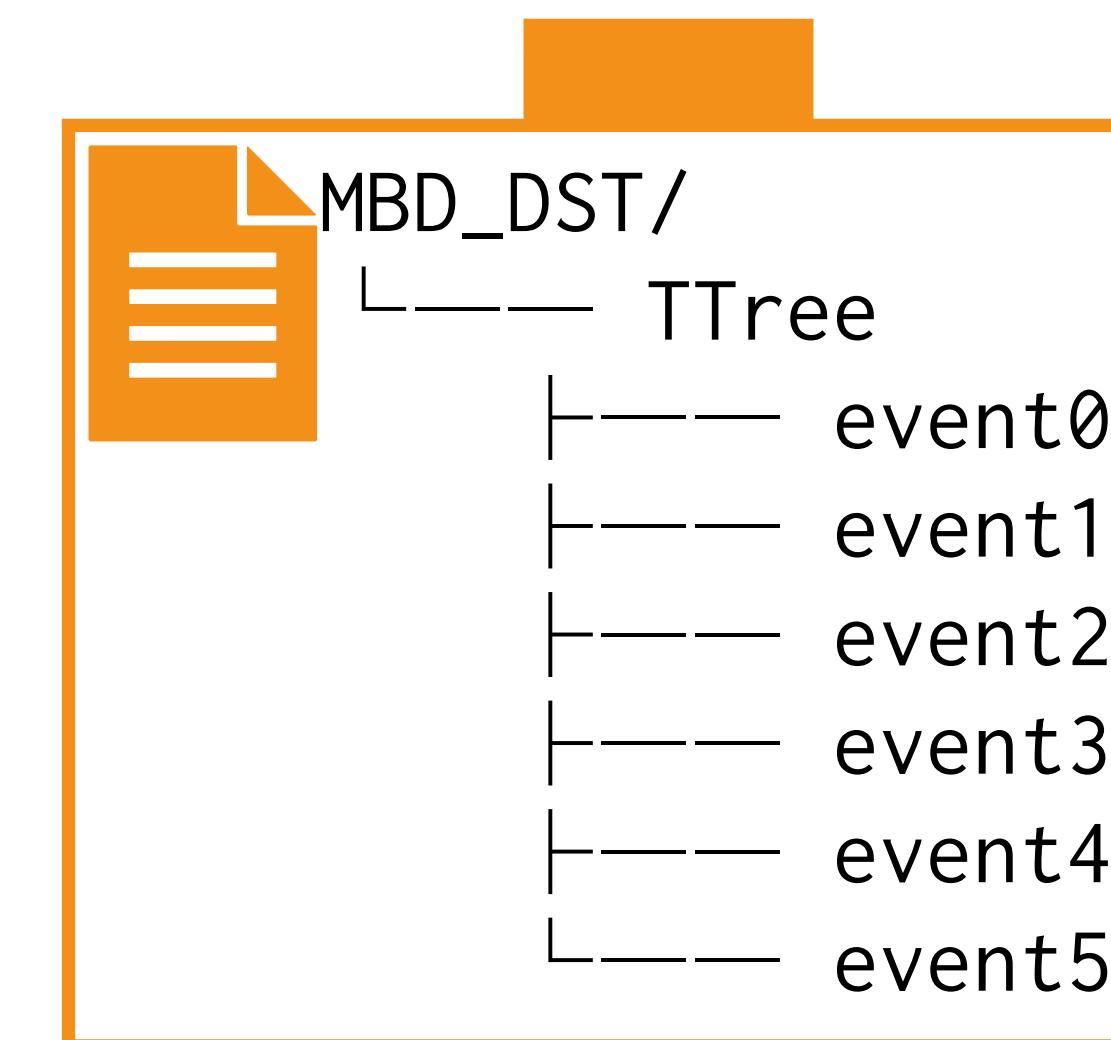
INTT のデータのみを扱う



結果

### MBD 単体解析

MBD のデータのみを扱う



結果

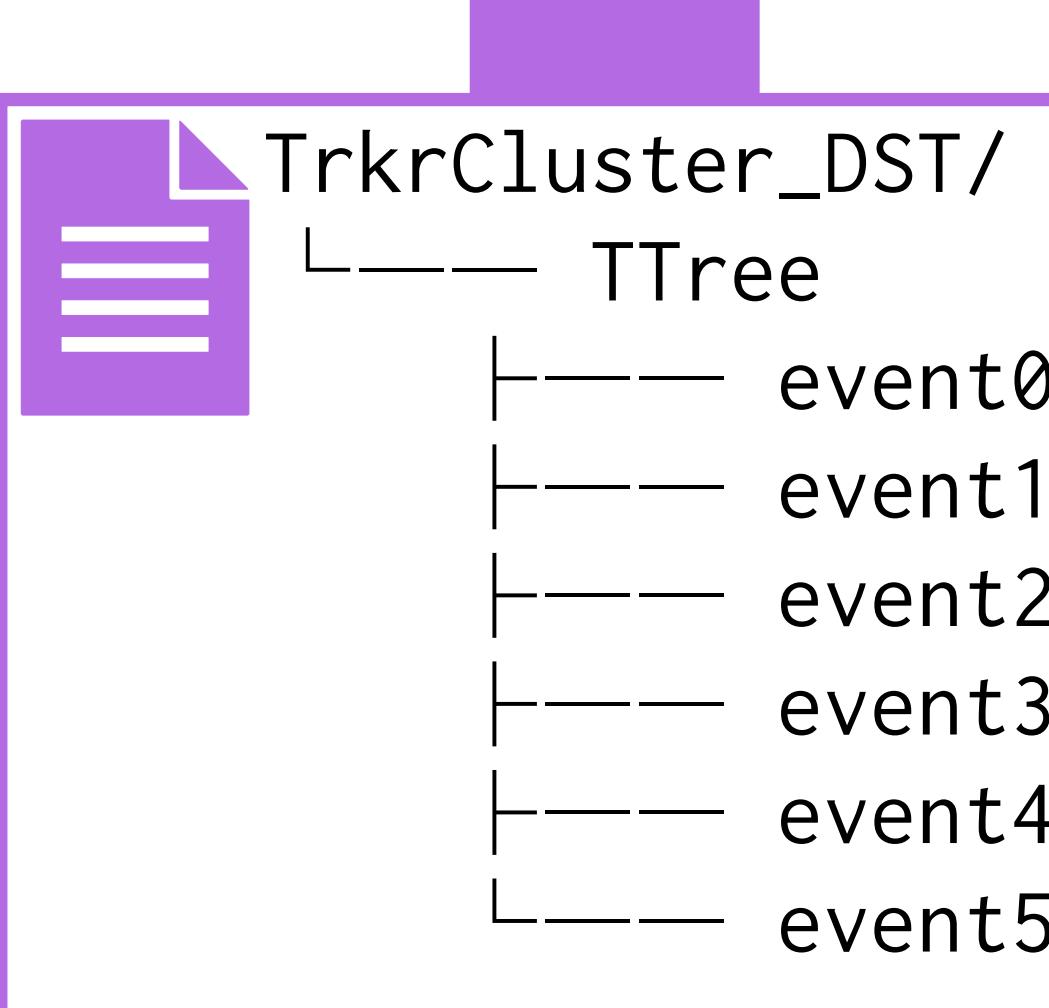
## sample 5

# INTT + MBD でデータ解析

INTT + MBD とはどういう意味か？

### INTT 単体解析

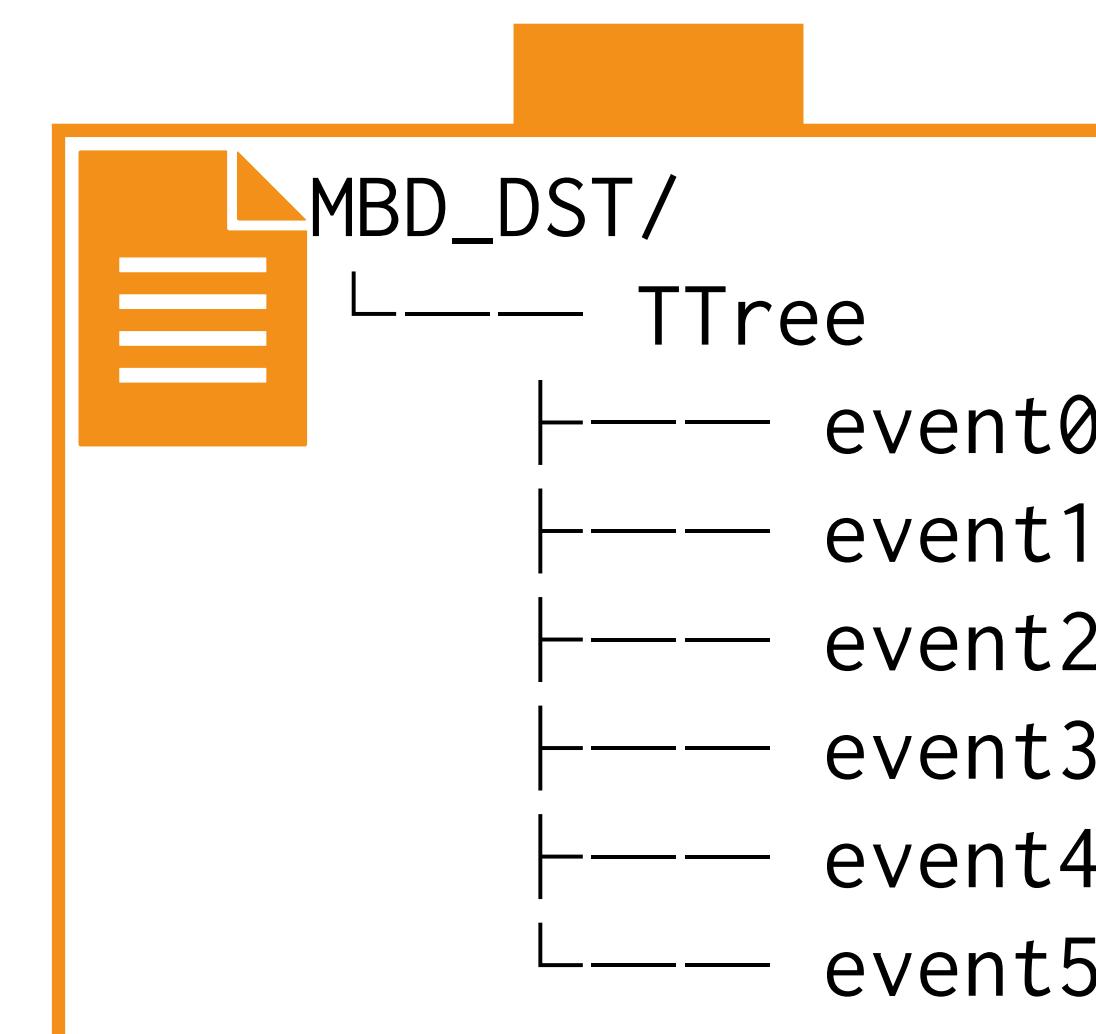
INTT のデータのみを扱う



結果

### MBD 単体解析

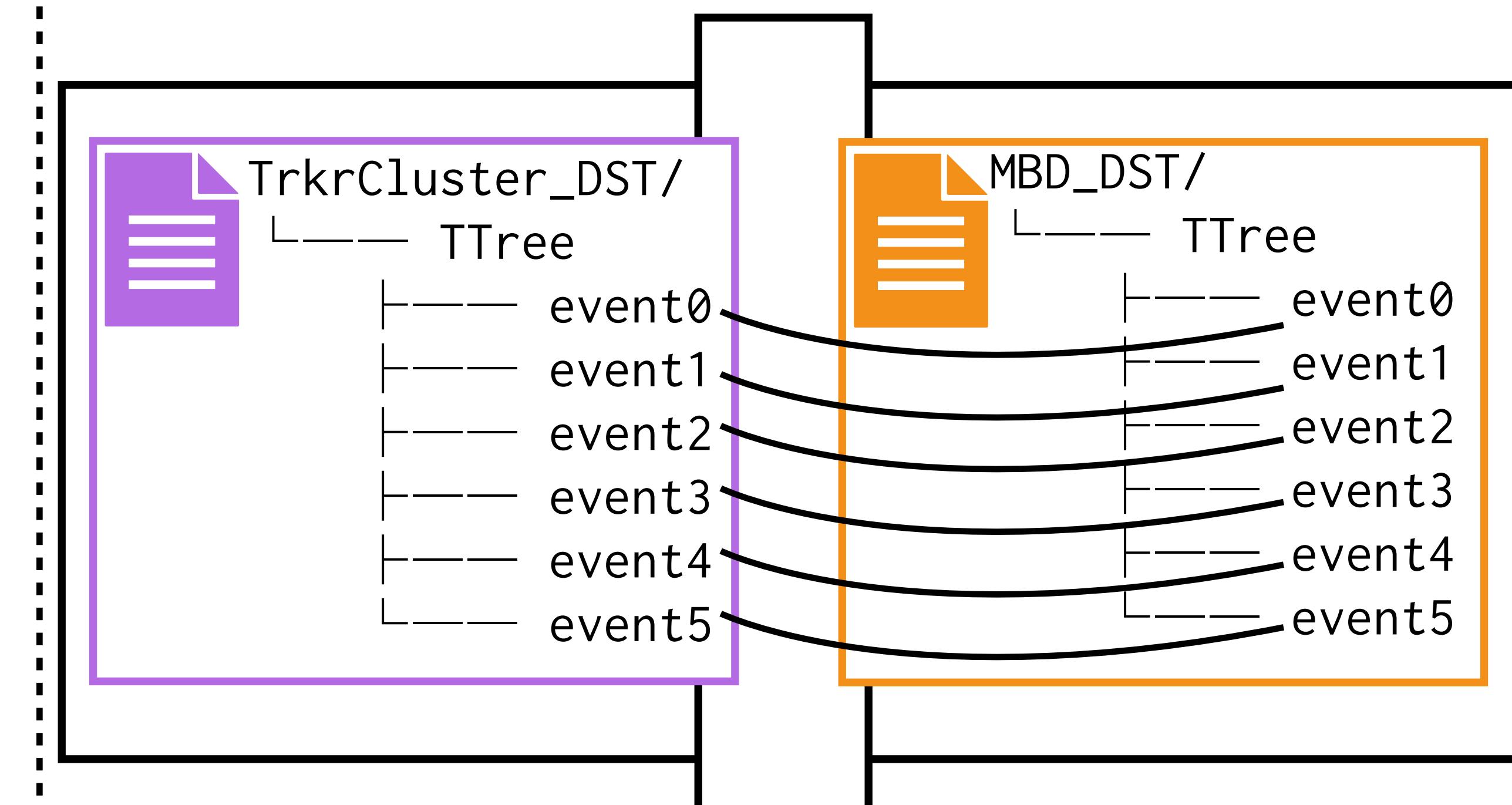
MBD のデータのみを扱う



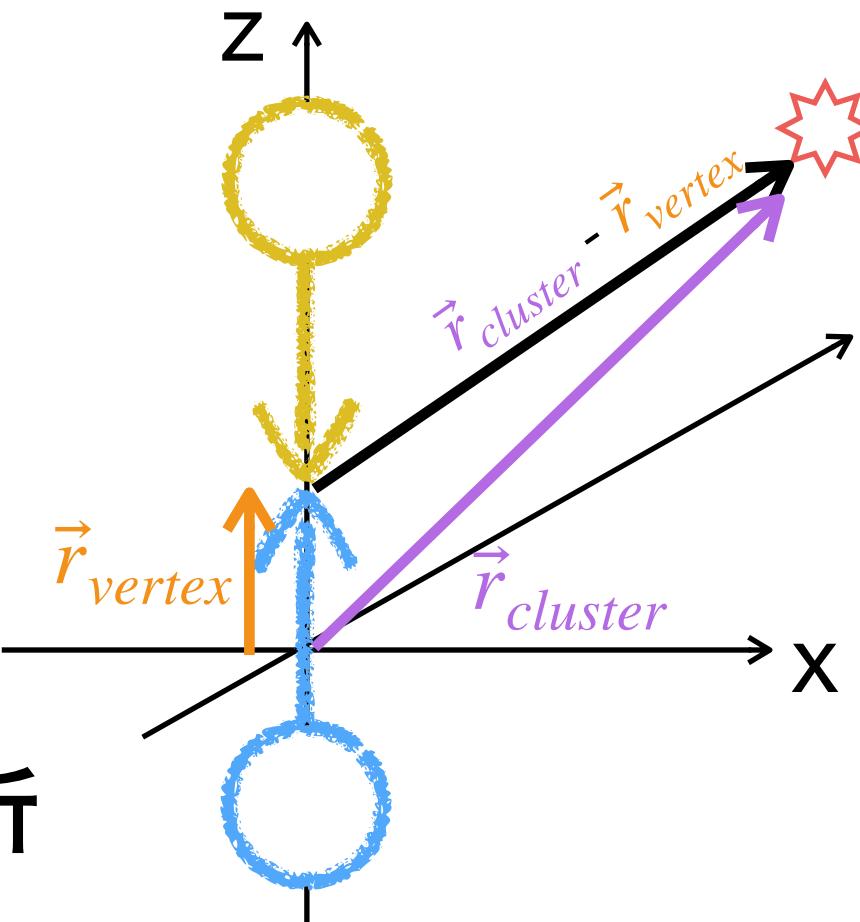
結果

### INTT + MBD 解析

INTT と MBD の両方のデータを扱う。  
イベントごとに両検出器のパラメータを操作



結果



## sample 5

### Fun4All\_minimum\_5.C

```
1 // Fun4All headers
2 #include <fun4all/Fun4AllServer.h>
3 #include <fun4all/Fun4AllDstInputManager.h>
4
5 // Some general header macros
6 #include <GlobalVariables.C>
7 #include <G4Setup_sPHENIX.C>
8 #include <G4_Input.C>
9
10 // Trkr headers
11 #include <Trkr_RecoInit.C>
12 #include <Trkr_Clustering.C>
13 #include <G4_ActsGeom.C>
14
15 // something else
16 #include <ffamodules/FlagHandler.h>
17 #include <ffamodules/HeadReco.h>
18 #include <ffamodules/SyncReco.h>
19 #include <ffamodules/CDBInterface.h>
20
21 #include <phool/PHRandomSeed.h>
22 #include <phool/recoConsts.h>
23
24 #include <fun4allraw/SingleMbdTriggerInput.h>
25 #include <fun4allraw/InputManagerType.h>
26
27 R__LOAD_LIBRARY(libfun4all.so)
28
29 #include <tutorial.h>
30 R__LOAD_LIBRARY( libtutorial.so )
31
32 int Fun4All_minimum_5(
33     int nEvents = 100
34     // const string &data = "/sphenix/lustre01/sphnxpro/physics/slurp/track
35     // n2pp_ana441_2024p007-00051428-00000.root"
36     )
37
38     Fun4AllServer *se = Fun4AllServer::instance();
39     //se->Verbosity(0);
40
41     // Data input
42     // Data input
43
44     // Trkr cluster
45     Fun4AllInputManager *in_cluster = new Fun4AllDstInputManager( "DST_input_cluster" );
46     // string data_cluster = "DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00000.root";
47     // in_cluster->fileopen( data_cluster );
48     in_cluster->AddListFile( "list_MDC2_AuAu_run19_TrkrCluster.txt" ); // to read a list of files
49     se->registerInputManager( in_cluster );
50
51     // MBD
52     Fun4AllInputManager *in_mbd = new Fun4AllDstInputManager( "DST_input_mbd" );
53     // string data_mbd = "DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00000.root";
54     // in_mbd->fileopen( data_mbd );
55     in_mbd->AddListFile( "list_MDC2_AuAu_run19_MBD.txt" ); // to read a list of files, use it
56
57     se->registerInputManager( in_mbd );
```

## INTT + MBD でデータ解析

[INTT\\_Fun4All\\_Tutorial](#) を更新し、サンプルをダウンロードしよう。

サンプルのあるディレクトリで

\$ git pull

を実行する。

サンプルコード：

- Fun4All マクロ : Fun4All\_minimum\_3.C
- 解析コード : sample\_module\_5

2 つの Fun4AllInputManager インスタンスを作り、各タイプごとの DST を渡す

TrkrCluster DST を読み込む

MBD DST を読み込む

\* 同一 Fun4AllInputManager に全 DST を渡すとイベントごとのデータの結合をやってくれない。

```
[nukazuka@sphnxuser03 06:03:55 Fun4All_samples] $ cat list_MDC2_AuAu_run19_TrkrCluster.txt
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00000.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00001.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00002.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00003.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00004.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00005.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00006.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00007.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00008.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00009.root
DST_TRKR_CLUSTER_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00010.root

[nukazuka@sphnxuser03 06:37:22 Fun4All_samples] $ cat list_MDC2_AuAu_run19_MBD.txt
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00000.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00001.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00002.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00003.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00004.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00005.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00006.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00007.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00008.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00009.root
DST_MBD_EPD_sHijing_0_20fm_50kHz_bkg_0_20fm-0000000019-00010.root
```

今回は MDC2 の AuAu minimum bias MC データを触ってみます

## sample 5

# INTT + MBD でデータ解析

### Fun4All\_minimum\_5.C

```
59 // Flag Handler is always needed to read flags from input (if used)
60 // and update our rc flags with them. At the end it saves all flags
61 // again on the DST in the Flags node under the RUN node
62 FlagHandler *flag = new FlagHandler();
63 se->registerSubsystem(flag);
64 Enable::CDB = true;
65 // global tag
66
67 recoConsts *rc = recoConsts::instance();
68 rc->set_StringFlag("CDB_GLOBALTAG", CDB::global_tag);
69 // 64 bit timestamp
70 rc->set_uint64Flag("TIMESTAMP", CDB::timestamp);
71 rc->set_IntFlag("RUNNUMBER", 0 );
72
73 /////////////////////////////////
74 // Something depends on Acts should be below....
75 /////////////////////////////////
76 // central tracking
77 Enable::MVTX = true;
78 Enable::TPC = true;
79 Enable::MICROMEGAS = true;
80 Enable::INTT = true;
81 Enable::BLACKHOLE = true;
82 G4MAGNET::magfield_rescale = 1.0;
83
84 // Initialize the selected subsystems
85 // G4Init();
86
87 // GEANT4 Detector description
88 // if (!Input::READHITS)
89 // {
90 //   G4Setup();
91 // }
92
93 TrackingInit(); // necessary for ActsGeometry
94
95 /////////////////////
96 // Your analysis module //
97 ///////////////
98 tutorial* analysis_module = new tutorial( "name" );
99 analysis_module->Verbosity( 1 ); // 0: minimum(default), 1: event by event info, 2: hit by hit info
100 se->registerSubsystem( analysis_module );
101
102 //se->skip(skip);
103 se->run(nEvents);
104 se->End();
105
106 delete se;
107
108 gSystem->Exit(0);
109 return 0;
110 }
```

[INTT\\_Fun4All\\_Tutorial](#) を更新し、サンプルをダウンロードしよう。

サンプルのあるディレクトリで

\$ git pull

を実行する。

サンプルコード：

- Fun4All マクロ : Fun4All\_minimum\_3.C
- 解析コード : sample\_module\_5

解析モジュールは sample\_module\_5 に入っている  
tutorial をコンパイルして使う

```

39 class tutorial : public SubsysReco
40 {
41     public:
42
43     tutorial(const std::string &name = "tutorial");
44
45     ~tutorial() override;
46
47     int Init(PHCompositeNode *topNode) override;
48
49     int InitRun(PHCompositeNode *topNode) override;
50
51     int process_event(PHCompositeNode *topNode) override;
52
53     /// Clean up internals after each event.
54     int ResetEvent(PHCompositeNode *topNode) override;
55
56     /// Called at the end of each run.
57     int EndRun(const int runnumber) override;
58
59     /// Called at the end of all processing.
60     int End(PHCompositeNode *topNode) override;
61
62     /// Reset
63     int Reset(PHCompositeNode * /*topNode*/) override;
64
65     void Print(const std::string &what = "ALL") const override;
66
67     //! You can set the name of the output file, otherwise it's tutorial_sample4.root
68     void SetOutputPath( std::string path ){ output_path_ = path; }
69
70 private:
71     //! A function for the analysis of INTT clusters
72     int cluster_analysis(PHCompositeNode *topNode, TrkrClusterContainerv4* node_cluster_map, ActsGeometry* nodeActs );
73     int mbd_analysis(PHCompositeNode *topNode, MbdOut* mbdout_node );
74
75     //! Reset function for cluster parameters
76     int ResetClusterLoop();
77
78     std::string output_path_ = "tutorial_sample5.root";
79     TFile* output_;      //! I/O of output ROOT file
80     TTree* tree_event_; //! Tree for event information
81     TTree* tree_cluster_; //! Tree for cluster information
82
83     //variables for tree_event_
84     int run_num_ = 0;           //! run number
85     int event_id_ = 0;          //! event number in this run
86     int cluster_num_ = 0;        //! the number of clusters on INTT
87     int cluster_num_layer_[4] = { 0 }; //! the number of clusters on each INTT layer (0-3)
88
89     // variables for tree_cluster_
90     TVector3 vec_cluster_;      //! position vector of a TrkrCluster
91     float position_[3];         //! cluster position in the lab-frame in cm
92     int layer_ = 0;              //! INTT layer ID for this cluster
93     float adc_ = 0;              //! ADC of this cluster (not 0, 1, ..., 7 but DAC value)
94     float size_phi_ = 0;         //! cluster size in phi direction
95     float phi_ = 0;              //! phi position of this cluster (radian)
96     float theta_ = 0;             //! theta position of this cluster (radian)
97     float eta_ = 0;              //! pseudorapidity of this cluster;
98     float eta_diff_ = 0;          //! Difference of eta and corrected eta by MBD vertex
99
100    // variables for MBDout
101    TVector3 vec_mbd_;          //! Position vector of a vertex reconstructed by MbdOut
102    bool is_valid_mbd_out_ = false; //! A flag to judge whether the vertex reconstruction was successful or not
103    float position_mbd_[3] = { 0.0 }; //! x, y, z of MBD vertex for TTree
104    float position_error_mbd_[3] = { 0.0 }; //! Error of x, y, z of MBD vertex for TTree
105    float mbd_q_part_[2] = { 0.0 }; //! MBD charge, 0: South, 1: North
106    float mbd_q_ = { 0.0 };        //! Total MBD charg
107 };

```

# INTT + MBD でデータ解析

## 解析モジュール

sample 5

中身は sample\_module\_4 とだいたい同じ。変更点は

- MbdOut の解析を追加
- MbdOut から得た vertex の位置をクラスター  $\phi, \theta, \eta$  の計算に使用

MbdOut の解析をする関数を新たに追加

MBD 関係の変数を新たに追加

# INTT + MBD でデータ解析

## 解析モジュール

sample 5

中身は sample\_module\_4 とだいたい同じ。変更点は

- MbdOut の解析を追加
- MbdOut から得た vertex の位置をクラスター  $\phi, \theta, \eta$  の計算に使用

```

73 int tutorial::process_event(PHCompositeNode *topNode)
74 {
75
76     // Get nodes
77     // -----
78
79     // -----
80     // Getting TrkrClusterContainer node
81     // TRKR_CLUSTER node: Information of TrkrCluster
82     // -----
83     std::string node_name_trkr_cluster = "TRKR_CLUSTER";
84     TrkrClusterContainerV4* node_cluster_map =
85         findNode::getClass<TrkrClusterContainerV4>(topNode, node_name_trkr_cluster);
86
87     if(!node_cluster_map)
88     {
89         std::cerr << PHWHERE << node_name_trkr_cluster << " node is missing." << std::endl;
90         return Fun4AllReturnCodes::ABORTEVENT;
91     }
92
93     // -----
94     // Getting Acts node to assign (x, y, z) coordinate to clusters
95     // ActsGeometry node: for the global coordinate
96     // -----
97     ActsGeometry *nodeActs = findNode::getClass<ActsGeometry>(topNode, "ActsGeometry");
98     if( !nodeActs )
99     {
100        std::cout << PHWHERE << "No ActsGeometry on node tree. Bailing." << std::endl;
101        return Fun4AllReturnCodes::ABORTEVENT;
102    }
103
104    // -----
105    // Getting EventHeader to get event info
106    // -----
107    EventHeaderV1* node_event_header = findNode::getClass<EventHeaderV1>( topNode, "EventHeader" );
108    if( !node_event_header )
109    {
110        std::cout << PHWHERE << "No EventHeader on node tree. Skip this event." << std::endl;
111        return Fun4AllReturnCodes::ABORTEVENT;
112    }
113
114    MbdOut *mbdout_node = findNode::getClass<MbdOutV2>(topNode, "MbdOut");
115    if( !mbdout_node )
116    {
117        std::cout << PHWHERE << "::ERROR - cannot find MbdOut" << std::endl;
118        return Fun4AllReturnCodes::ABORTEVENT;
119    }
120
121
122    // Analysis
123    // -----
124    run_num_ = node_event_header->get_RunNumber();
125    event_id_ = node_event_header->get_EvtSequence();
126
127
128    // If user wants to see event infomation, do it
129    if( this->Verbosity() > 0 )
130    {
131        std::cout << "Run " << run_num_ << "\t"
132            << "Event " << std::setw(10) << event_id_
133            << std::endl;
134    }
135
136    // analysis codes for INTT clusters are written in the function below
137    this->mbd_analysis( topNode, mbdout_node );
138    this->cluster_analysis( topNode, node_cluster_map, nodeActs );
139
140    // Fill event-case tree at the end of event process
141    tree_event_->Fill();
142    return Fun4AllReturnCodes::EVENT_OK;
143 }
144

```

MbdOut node の取得

MbdOut 解析関数をクラスター解析関数の前で実行

クラスター解析で MBD vertex の情報を使うので、MbdOut 解析関数を先に実行しなければならない。

# INTT + MBD でデータ解析 解析モジュール

sample 5

```
73 int tutorial::process_event(PHCompositeNode *topNode)
74 {
75     //////////////////////////////////////////////////////////////////
76     // Get nodes
77     //////////////////////////////////////////////////////////////////
78
79
80     //////////////////////////////////////////////////////////////////
81     // Getting TrkrClusterContainer node
82     // TRKR_CLUSTER node: Information of TrkrCluster
83     //////////////////////////////////////////////////////////////////
84     std::string node_name_trkr_cluster = "TRKR_CLUSTER";
85     TrkrClusterContainerV4* node_cluster_map =
86         findNode::getClass<TrkrClusterContainerV4>(topNode, node_name_trkr_cluster);
87
88     if(!node_cluster_map)
89     {
90         std::cerr << PHWHERE << node_name_trkr_cluster << " node is missing." << std::endl;
91         return Fun4AllReturnCodes::ABORTEVENT;
92     }
93
94     //////////////////////////////////////////////////////////////////
95     // Getting Acts node to assign (x, y, z) coordinate to clusters
96     // ActsGeometry node: for the global coordinate
97     //////////////////////////////////////////////////////////////////
98     ActsGeometry *nodeActs = findNode::getClass<ActsGeometry>(topNode, "ActsGeometry");
99     if( !nodeActs )
100    {
101        std::cout << PHWHERE << "No ActsGeometry on node tree. Bailing." << std::endl;
102        return Fun4AllReturnCodes::ABORTEVENT;
103    }
104
105    //////////////////////////////////////////////////////////////////
106    // Getting EventHeader to get event info
107    //////////////////////////////////////////////////////////////////
108    EventHeaderV1* nodeEvent_header = findNode::getClass<EventHeaderV1>( topNode, "EventHeader" );
109    if( !nodeEvent_header )
110    {
111        std::cout << PHWHERE << "No EventHeader on node tree. Skip this event." << std::endl;
112        return Fun4AllReturnCodes::ABORTEVENT;
113    }
114
115    MbdOut *mbdout_node = findNode::getClass<MbdOutV2>(topNode, "MbdOut");
116    if( !mbdout_node )
117    {
118        std::cout << PHWHERE << "ERROR - cannot find MbdOut" << std::endl;
119        return Fun4AllReturnCodes::ABORTEVENT;
120    }
121
122    //////////////////////////////////////////////////////////////////
123    // Analysis
124    //////////////////////////////////////////////////////////////////
125    run_num_ = nodeEvent_header->get_RunNumber();
126    event_id_ = nodeEvent_header->get_EvtSequence();
127
128    // If user wants to see event information, do it
129    if( this->Verbosity() > 0 )
130    {
131        std::cout << "Run " << run_num_ << "\n"
132            << "Event " << std::setw(10) << event_id_
133            << std::endl;
134    }
135
136    // analysis codes for INTT clusters are written in the function below
137    this->mbd_analysis( topNode, mbdout_node );
138    this->cluster_analysis( topNode, node_cluster_map, nodeActs );
139
140    // Fill event-case TTree at the end of event process
141    treeEvent->Fill();
142
143    return Fun4AllReturnCodes::EVENT_OK;
144 }
```

## mbd\_analysis

```
241 int tutorial::mbd_analysis( PHCompositeNode *topNode, MbdOut* mbdout_node )
242 {
243
244     //////////////////////////////////////////////////////////////////
245     // Analysis at the level of MbdOut
246     //////////////////////////////////////////////////////////////////
247     // Condition of valid: t0 needs to be finite and larger than -9999
248     is_valid_mbd_out_ = mbdout_node->isValid();
249     if( is_valid_mbd_out_ == false )
250         return Fun4AllReturnCodes::EVENT_OK;
251
252     position_mbd_[2] = mbdout_node->get_zvtx();
253     vec_mbd_.SetZ( position_mbd_[2] );
254     position_error_mbd_[2] = mbdout_node->get_zvtxerr();
255
256     for( int i=0; i<2; i++ )
257     {
258         mbd_q_part_[i] = mbdout_node->get_q( i );
259     }
260
261     mbd_q_ = mbd_q_part_[0] += mbd_q_part_[1];
262
263     return Fun4AllReturnCodes::EVENT_OK;
264 }
```

MbdOut の解析は至極簡単

再構成した vertex の z 座標を取得したり、 MBD で測定した電荷量を取得するだけ。

# sample\_module\_5/tutorial.cc, cluster\_analysis

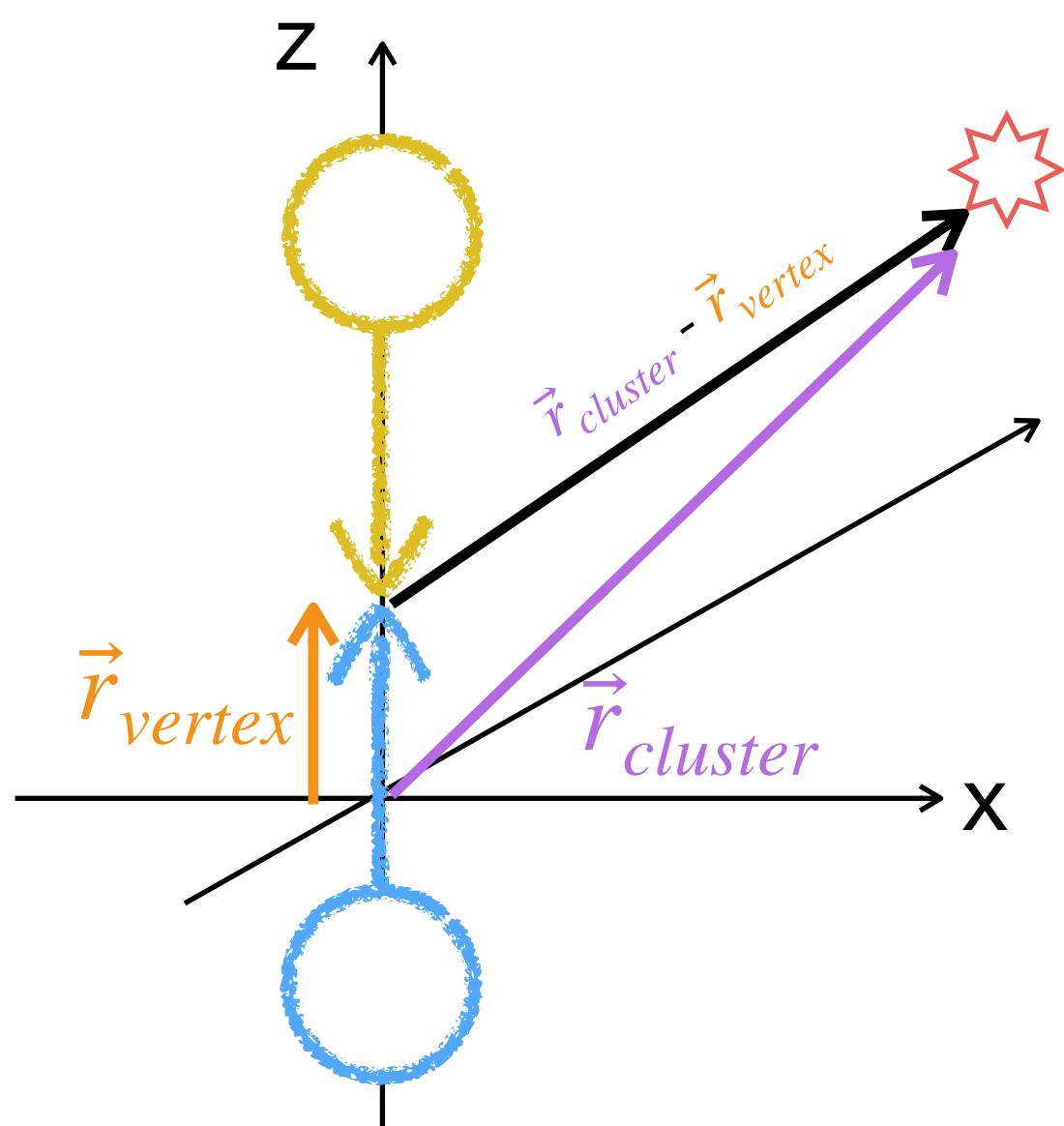
```

145 int tutorial::cluster_analysis(PHCompositeNode *topNode, TrkrClusterContainerv4* node_cluster_map, ActsGeometry* nodeActs )
146 {
147
148
149 // loop over all INTT layers (0: inner of inner barrel, 1: outer of inner, 2: inner of outer, 3: outer of outer)
150 for (unsigned int inttlayer = 0; inttlayer < 4; inttlayer++)
151 {
152
153 // get clusters only on the INTT layer, and loop over them
154 for (const auto &hitsetkey : node_cluster_map->getHitSetKeys(TrkrDefs::TrkrId::inttId, inttlayer + 3) )
155 {
156
157 // type: std::pair<ConstIterator, ConstIterator> ConstRange
158 // here, MMap_::const_iterator ConstIterator;
159 auto range = node_cluster_map->getClusters(hitsetkey);
160
161 // loop over iterators of this cluster
162 for (auto clusIter = range.first; clusIter != range.second; ++clusIter)
163 {
164
165 const auto cluskey = clusIter->first;
166 const auto cluster = clusIter->second;
167
168 // Get cluster position in lab-coordinate using Acts
169 const auto globalPos = nodeActs->getGlobalPosition(cluskey, cluster);
170
171 // Set cluster position in lab-coordinate to this cluster object
172 cluster->setPosition(0, globalPos.x());
173 cluster->setPosition(1, globalPos.y());
174 cluster->setPosition(2, globalPos.z());
175
176 // Assign cluster parameters
177 position_[0] = cluster->getPosition( 0 ); // x
178 position_[1] = cluster->getPosition( 1 ); // y
179 position_[2] = cluster->getPosition( 2 ); // z
180 adc_ = cluster->getAdc();
181 size_phi_ = cluster->getPhiSize();
182 layer_ = inttlayer;
183
184 // calc for some parameters
185 vec_cluster_ = TVector3( position_[0],position_[1], position_[2] );
186
187 TVector3 vec_cluster_vertex = vec_cluster_ - vec_mbd_;
188 phi_ = vec_cluster_vertex.Phi();
189 theta_ = vec_cluster_vertex.Theta();
190 eta_ = vec_cluster_vertex.Eta();
191
192 eta_diff_ = eta_ - vec_cluster_.Eta(); // to see the effect by vertex modification
193
194 // After getting all cluster parameters, let's fill them
195 tree_cluster_->Fill();
196
197 // #cluster counters
198 cluster_num_++; // all of them
199 cluster_num_layer_[ inttlayer ]++; // for each layer
200
201 // Then, reset the parameters (it's not mandatory if all parameters are available all the time. It's just in case)
202 this->ResetClusterLoop();
203

```

# INTT + MBD でデータ解析 sample 5 解析モジュール

MBD vertex の z 座標を用いた  
クラスター  $\phi, \theta, \eta$  の計算部分



## 手順

1. レポジトリをアップデートしてサンプルをダウンロードする

- 1.1. \$ mv {your place} # 実行場所は各自で異なる
- 1.2. \$ git pull

※ たぶんここで問題が続出する

2. sample\_module\_5 をコンパイルする

- 2.1. cd sample\_module\_5
- 2.2. mkdir build
- 2.3. mkdir install
- 2.4. cd build
- 2.5. ../autogen.sh –prefix=\$PWD/../install

3. Fun4All\_minimum\_5.C を実行する

- 3.1. \$ root -q -b ‘Fun4All\_minimum\_5.C( 100 )’

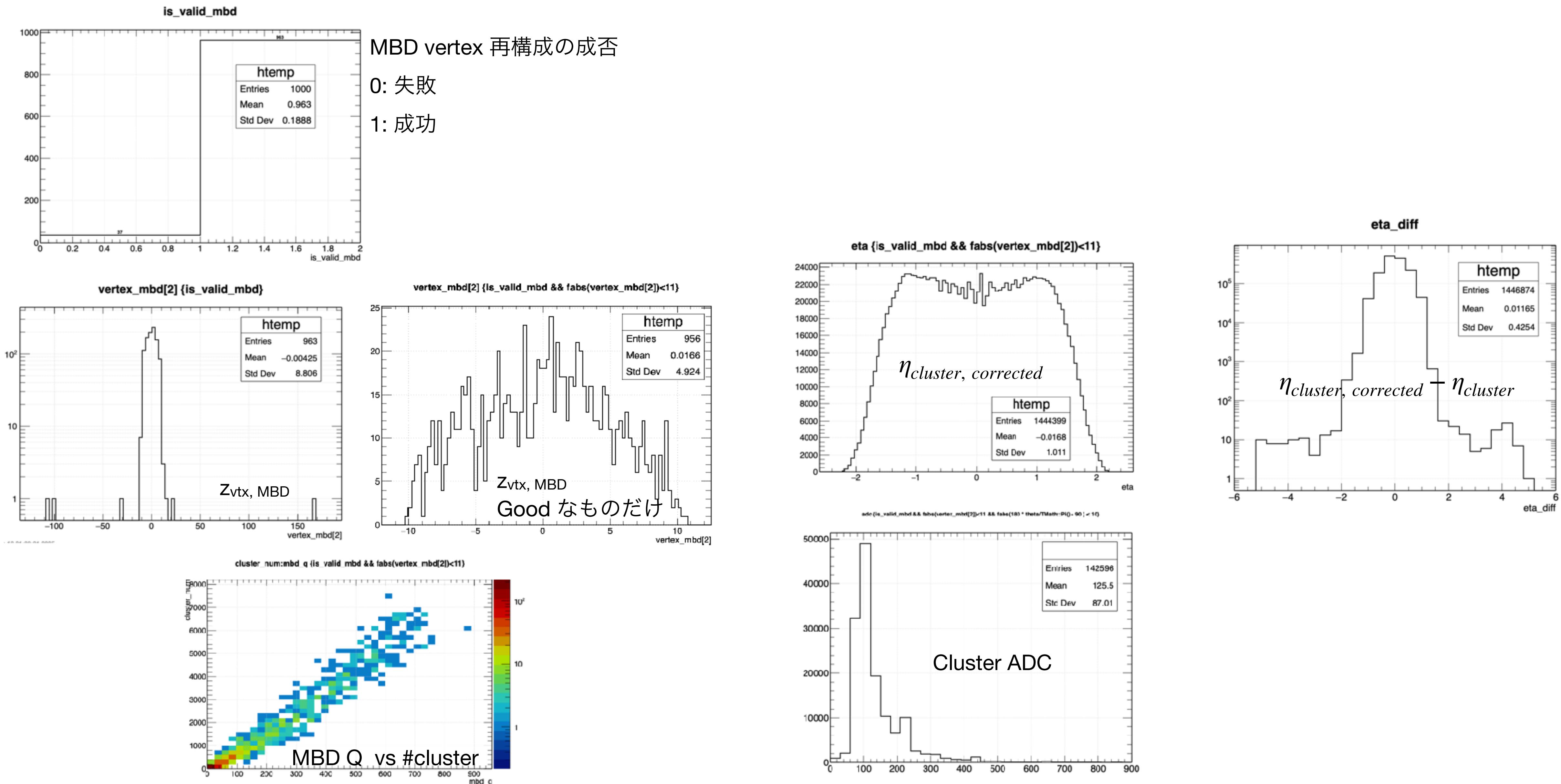
4. 生成された tutorial\_sample5.root の中身を確認する

- ・どのパラメータを見たら面白いか考えてみる

実習ポイント！

sample 5

# INTT + MBD でデータ解析：チェックするパラメータ例



# おわりに

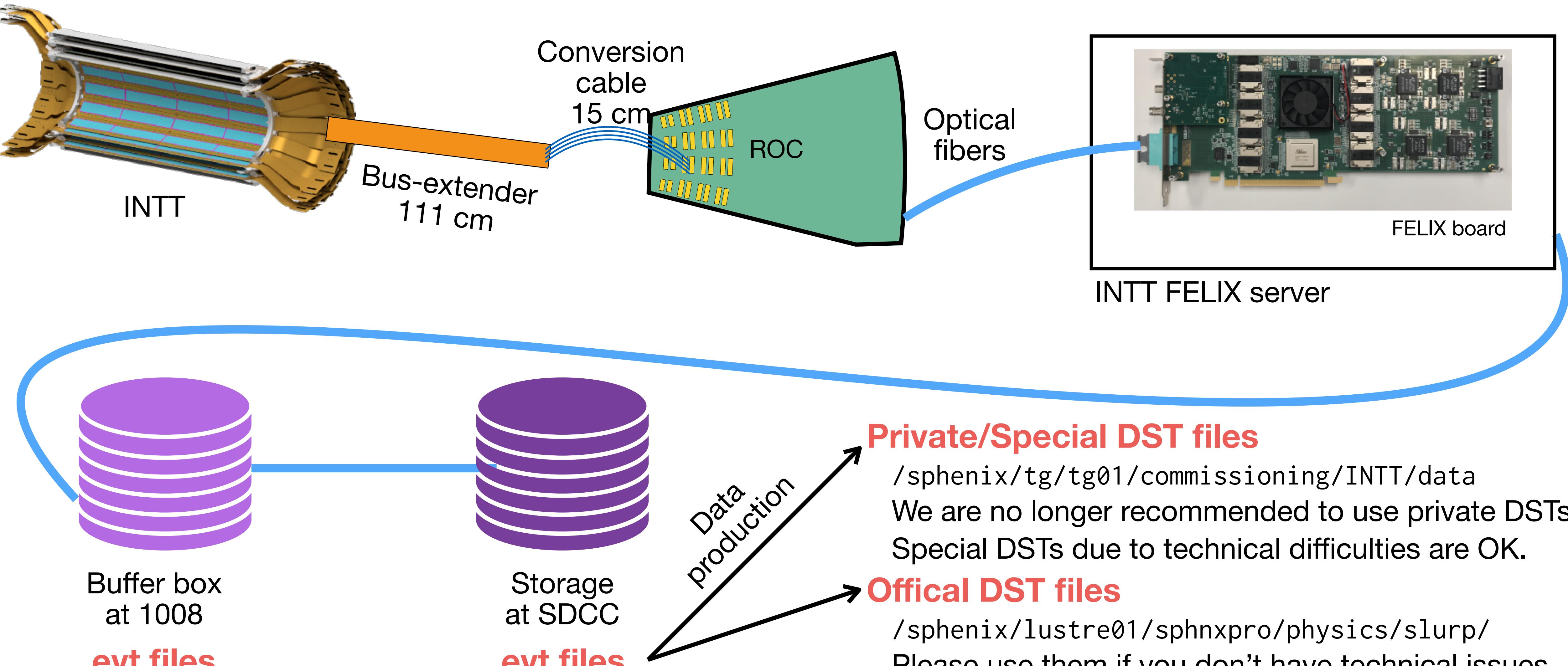
韓国ワークショップの続きとして INTT + MBD の解析を行いました。  
基本的に他の検出器や解析オブジェクト (トラッキングのシードや jet, その他諸々)  
も似たような手法で取り扱います。つまり

- [データ入力] 解析したいノードの入った DST を読み込む
- [データ入力] macro レポジトリにあるマクロの関数を実行し, よく使われる解析モジュールを取り込む  
(...Reco とか, カロリーメータタワーを取り扱うものとか・・・)
- [解析] 解析モジュールでノードを取得し, 情報を得る

バックアップ

DST

# Data process flow and data type



## sample 3

# Reading a DST file

You need to read DST(s) in your Fun4All macro using Fun4AllInputManager.  
You need to include some header files.

Including header files:

```
#include <fun4all/Fun4AllInputManager.h>
#include <fun4all/Fun4AllDstInputManager.h>
```

Single file:

```
Fun4AllInputManager *in = new Fun4AllDstInputManager("DSTin");
in->AddFile( data );
se->registerInputManager(in);
```

Single file (old method but event skipping works):

```
Fun4AllInputManager *in = new Fun4AllDstInputManager("DSTin");
in->fileopen( data );
se->registerInputManager(in);
```

Multiple files:

```
Fun4AllInputManager *in = new Fun4AllDstInputManager("DST");
in->AddListFile( list );
```

The screenshot shows a dark-themed Mattermost-style chat interface. At the top right are standard message controls: thumbs up, thumbs down, reply, forward, and more. The conversation starts with a message from Genki Nukazuka asking about functions to read DST files, with three options listed: Fun4AllInputManager::AddFile, Fun4AllInputManager::AddListFile, and Fun4AllDstInputManager::fileopen. Anthony Hedges responds that AddFile is used for a single DST and AddListFile for a file list. Christopher Pinkenburg notes that fileopen is an older API for skipping events. Genki Nukazuka thanks them and asks if fileopen actually opens the DST file, to which Christopher responds that it does and that's why AddFile is preferred. Genki then asks if fileopen is the correct method for multiple files, and Christopher confirms it shares the underlying machinery with AddListFile. Genki concludes by thanking Christopher.

Saved

Genki Nukazuka 3 months ago  
Hi, which function should I use to read DST files?

- Fun4AllInputManager::AddFile
- Fun4AllInputManager::AddListFile
- Fun4AllDstInputManager::fileopen

7 replies

Anthony Hedges 3 months ago  
In a given Fun4AllMacro, you'll probably instantiate a Fun4AllInputManager, which uses AddFile for a single DST, and AddListFile for a file list containing a single column of DST files, such as that generated by CreateDstList.pl

Christopher Pinkenburg 3 months ago  
::fileopen is an older API, but you need it when you want to skip events

Genki Nukazuka 3 months ago  
Thank you, @Anthony Hedges and @Christopher Pinkenburg !  
So, if I want to skip some events, I shouldn't use AddFile and AddListFile. Is that right?

Christopher Pinkenburg 3 months ago  
yes - that only works with fileopen() - the reason is that fileopen actually opens the DST and only then Fun4All can set up the skipping. Fixing this is a long standing item on my list but it's not that straightforward so that'll be a while

Genki Nukazuka 3 months ago  
Thank you for the explanation. OK, that's good to know. So, the recommendation is

- for a single file: Fun4AllDstInputManager::fileopen
- for multiple files: Fun4AllInputManager::AddListFile

Is it right?

Christopher Pinkenburg 3 months ago  
yes (where if you don't need the skipping, I would go for the AddFile() - it shares the underlying machinery with AddListFile() so if you go from single file to filelist it'll go through the same channels)

Genki Nukazuka 3 months ago  
Great! Thanks a lot!

[Discussion in Mattermost](#)  
[back to ToC](#)