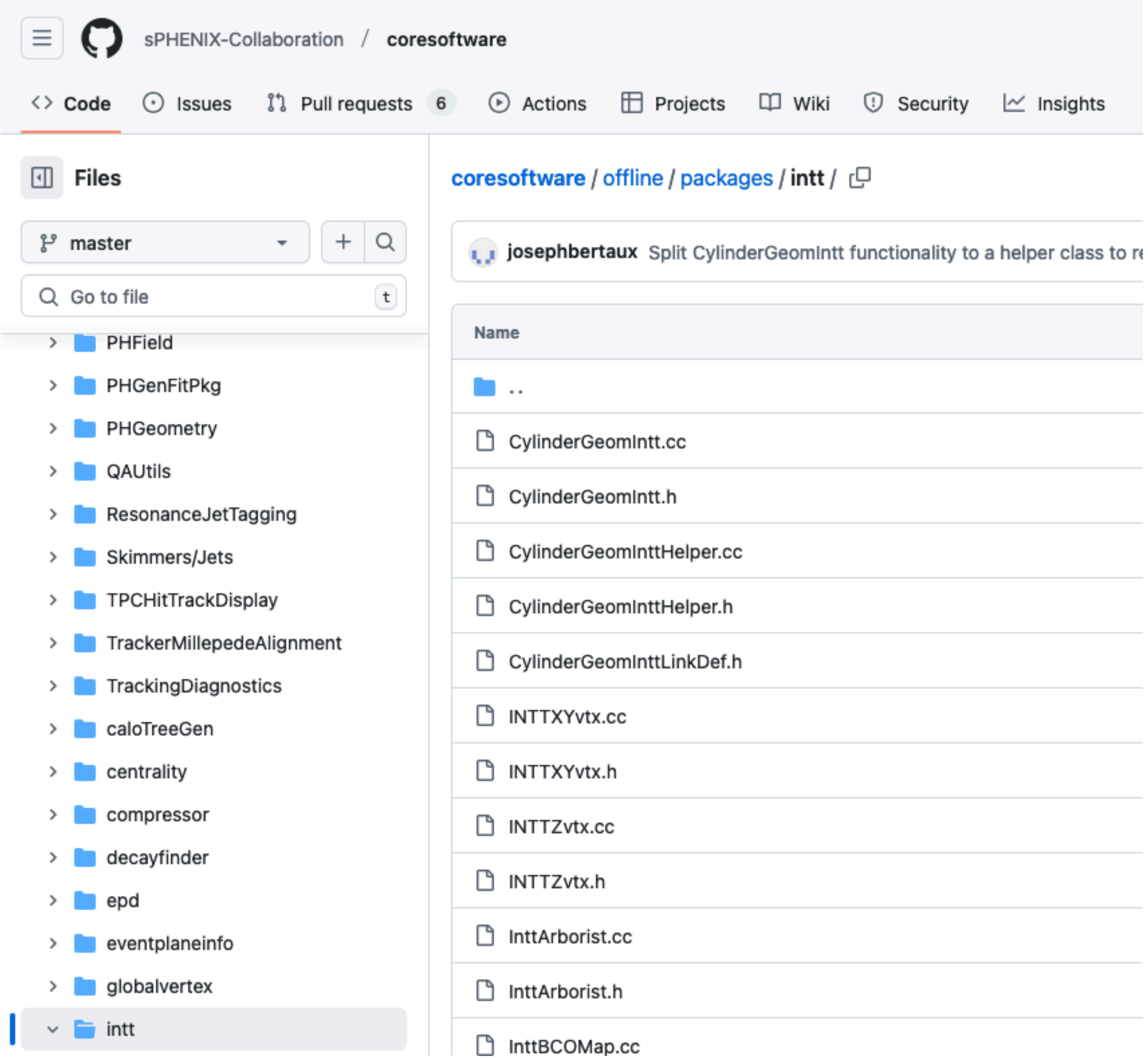
InttVertexFinder を使ってみた

糠塚元気（理研）

概要

- sPHENIX の coresoftware には InttXYVertexFinder, InttZVertexFinder があり，INTT を用いた vertex 再構成プログラムが提供されている。
- 少なくともローカルモードで取得したランで vertex 再構成を行うなら，INTT を使用しなければならない。
- 柳川くんに情報を提供するため使ってみた。
- MDC2 MC run19 (Hijing AuAu, 0_20fm(←これは何？)) のデータを用いて MC truth（真の値）, INTT 再構成, MBD 再構成それぞれの解析を行い，現状の InttVertexFinder を評価する。
- 来年の AuAu ランでも役に立つ



coresoftware / offline / packages / intt /

josephbertaux Split CylinderGeomIntt functionality to a helper class to r

- PHField
- PHGenFitPkg
- PHGeometry
- QAUtils
- ResonanceJetTagging
- Skimmers/Jets
- TPCHitTrackDisplay
- TrackerMillepedeAlignment
- TrackingDiagnostics
- caloTreeGen
- centrality
- compressor
- decayfinder
- epd
- eventplaneinfo
- globalvertex
- intt

Name
..
CylinderGeomIntt.cc
CylinderGeomIntt.h
CylinderGeomInttHelper.cc
CylinderGeomInttHelper.h
CylinderGeomInttLinkDef.h
INTTXYvtx.cc
INTTXYvtx.h
INTTZvtx.cc
INTTZvtx.h
InttArborist.cc
InttArborist.h
InttBCOMap.cc

MC データ

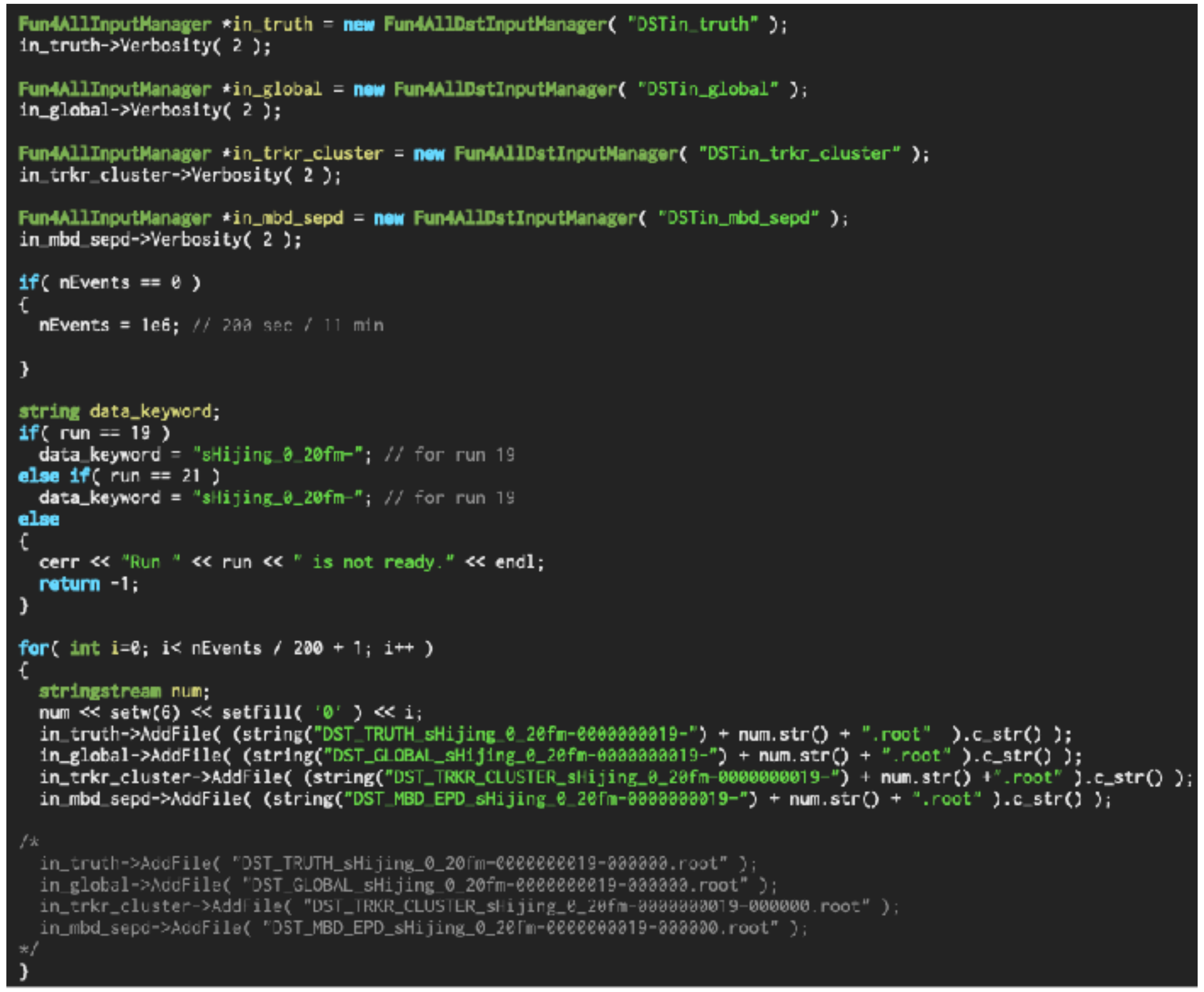
- Run19
- 金金衝突
- Hijing
- 0 — 20 fm (←これは何？)
- Pileup なし
- DST
  200 イベント / セグメント
  - MC truth: DST_TRUTH_sHijing_0_20fm-0000000019-??????.root
  - INTT: DST_TRKR_CLUSTER_sHijing_0_20fm-0000000019-??????.root
  - MBD: DST_GLOBAL_sHijing_0_20fm-0000000019-??????.root
- 6000 イベントだけ解析

Vertex (z のみ) 再構成

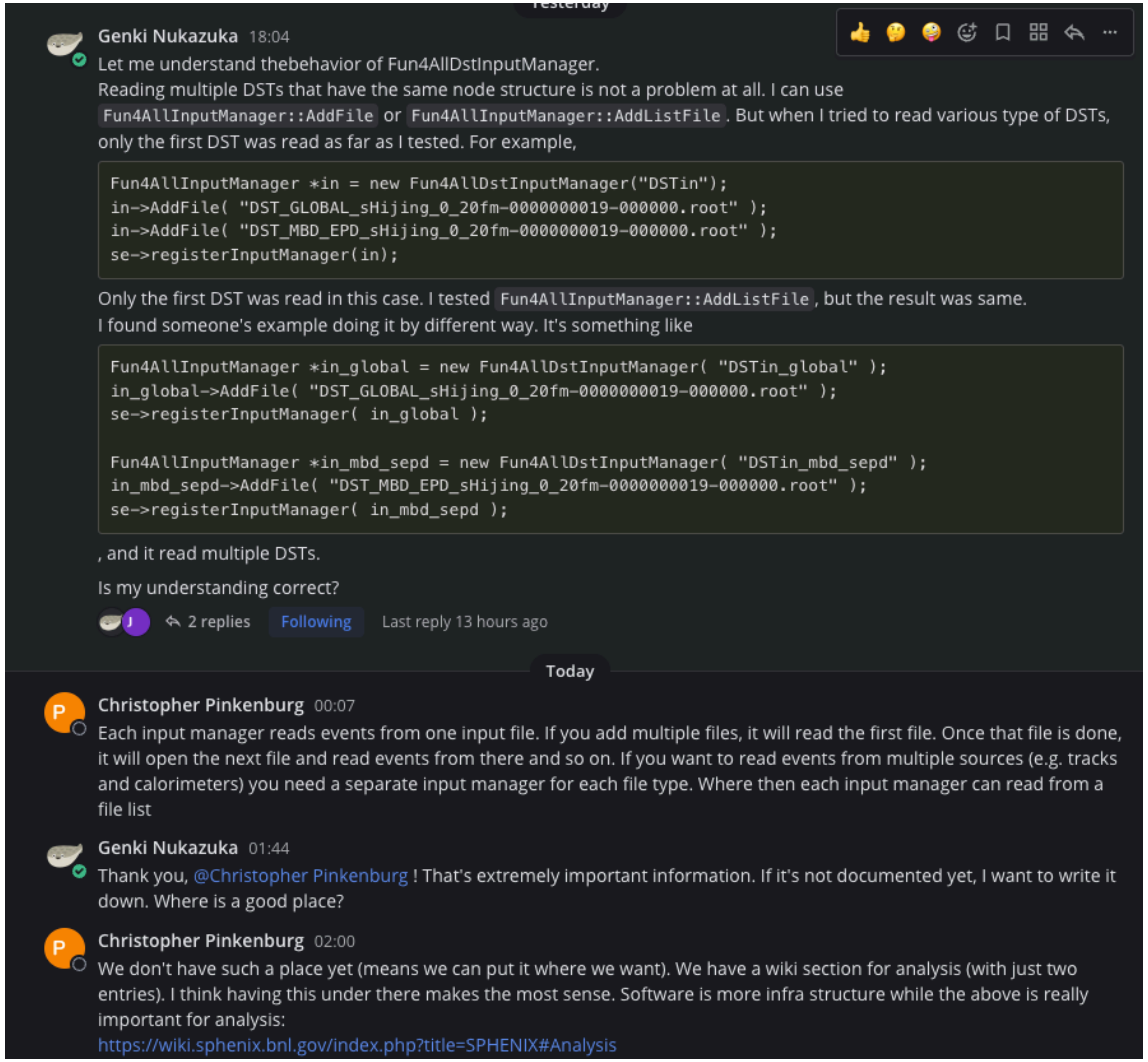
リンク 1

リンク2

- 複数の DST を並列して読み込むためには，DST 事に Fun4AllInputManager を作らなければ行けないらしい



```
Fun4AllInputManager *in_truth = new Fun4AllDstInputManager( "DSTin_truth" );
in_truth->Verbosity( 2 );

Fun4AllInputManager *in_global = new Fun4AllDstInputManager( "DSTin_global" );
in_global->Verbosity( 2 );

Fun4AllInputManager *in_trkr_cluster = new Fun4AllDstInputManager( "DSTin_trkr_cluster" );
in_trkr_cluster->Verbosity( 2 );

Fun4AllInputManager *in_mbd_sepd = new Fun4AllDstInputManager( "DSTin_mbd_sepd" );
in_mbd_sepd->Verbosity( 2 );

if( nEvents == 0 )
{
  nEvents = 1e6; // 200 sec / 11 min

}

string data_keyword;
if( run == 19 )
  data_keyword = "sHijing_0_20fm-"; // for run 19
else if( run == 21 )
  data_keyword = "sHijing_0_20fm-"; // for run 19
else
{
  cerr << "Run " << run << " is not ready." << endl;
  return -1;
}

for( int i=0; i< nEvents / 200 + 1; i++ )
{
  stringstream num;
  num << setw(6) << setfill( '0' ) << i;
  in_truth->AddFile( (string("DST_TRUTH_sHijing_0_20fm-0000000019-") + num.str() + ".root"  ).c_str() );
  in_global->AddFile( (string("DST_GLOBAL_sHijing_0_20fm-0000000019-") + num.str() + ".root" ).c_str() );
  in_trkr_cluster->AddFile( (string("DST_TRKR_CLUSTER_sHijing_0_20fm-0000000019-") + num.str() +".root" ).c_str() );
  in_mbd_sepd->AddFile( (string("DST_MBD_EPD_sHijing_0_20fm-0000000019-") + num.str() + ".root" ).c_str() );

/*
  in_truth->AddFile( "DST_TRUTH_sHijing_0_20fm-0000000019-000000.root" );
  in_global->AddFile( "DST_GLOBAL_sHijing_0_20fm-0000000019-000000.root" );
  in_trkr_cluster->AddFile( "DST_TRKR_CLUSTER_sHijing_0_20fm-0000000019-000000.root" );
  in_mbd_sepd->AddFile( "DST_MBD_EPD_sHijing_0_20fm-0000000019-000000.root" );
*/
}
```



Yesterday

Genki Nukazuka 18:04

Let me understand thebehavior of Fun4AllDstInputManager.
Reading multiple DSTs that have the same node structure is not a problem at all. I can use `Fun4AllInputManager::AddFile` or `Fun4AllInputManager::AddListFile`. But when I tried to read various type of DSTs, only the first DST was read as far as I tested. For example,

```
Fun4AllInputManager *in = new Fun4AllDstInputManager("DSTin");
in->AddFile( "DST_GLOBAL_sHijing_0_20fm-0000000019-000000.root" );
in->AddFile( "DST_MBD_EPD_sHijing_0_20fm-0000000019-000000.root" );
se->registerInputManager(in);
```

Only the first DST was read in this case. I tested `Fun4AllInputManager::AddListFile`, but the result was same.
I found someone's example doing it by different way. It's something like

```
Fun4AllInputManager *in_global = new Fun4AllDstInputManager( "DSTin_global" );
in_global->AddFile( "DST_GLOBAL_sHijing_0_20fm-0000000019-000000.root" );
se->registerInputManager( in_global );

Fun4AllInputManager *in_mbd_sepd = new Fun4AllDstInputManager( "DSTin_mbd_sepd" );
in_mbd_sepd->AddFile( "DST_MBD_EPD_sHijing_0_20fm-0000000019-000000.root" );
se->registerInputManager( in_mbd_sepd );
```

, and it read multiple DSTs.

Is my understanding correct?

2 replies Following Last reply 13 hours ago

Today

Christopher Pinkenburg 00:07

Each input manager reads events from one input file. If you add multiple files, it will read the first file. Once that file is done, it will open the next file and read events from there and so on. If you want to read events from multiple sources (e.g. tracks and calorimeters) you need a separate input manager for each file type. Where then each input manager can read from a file list

Genki Nukazuka 01:44

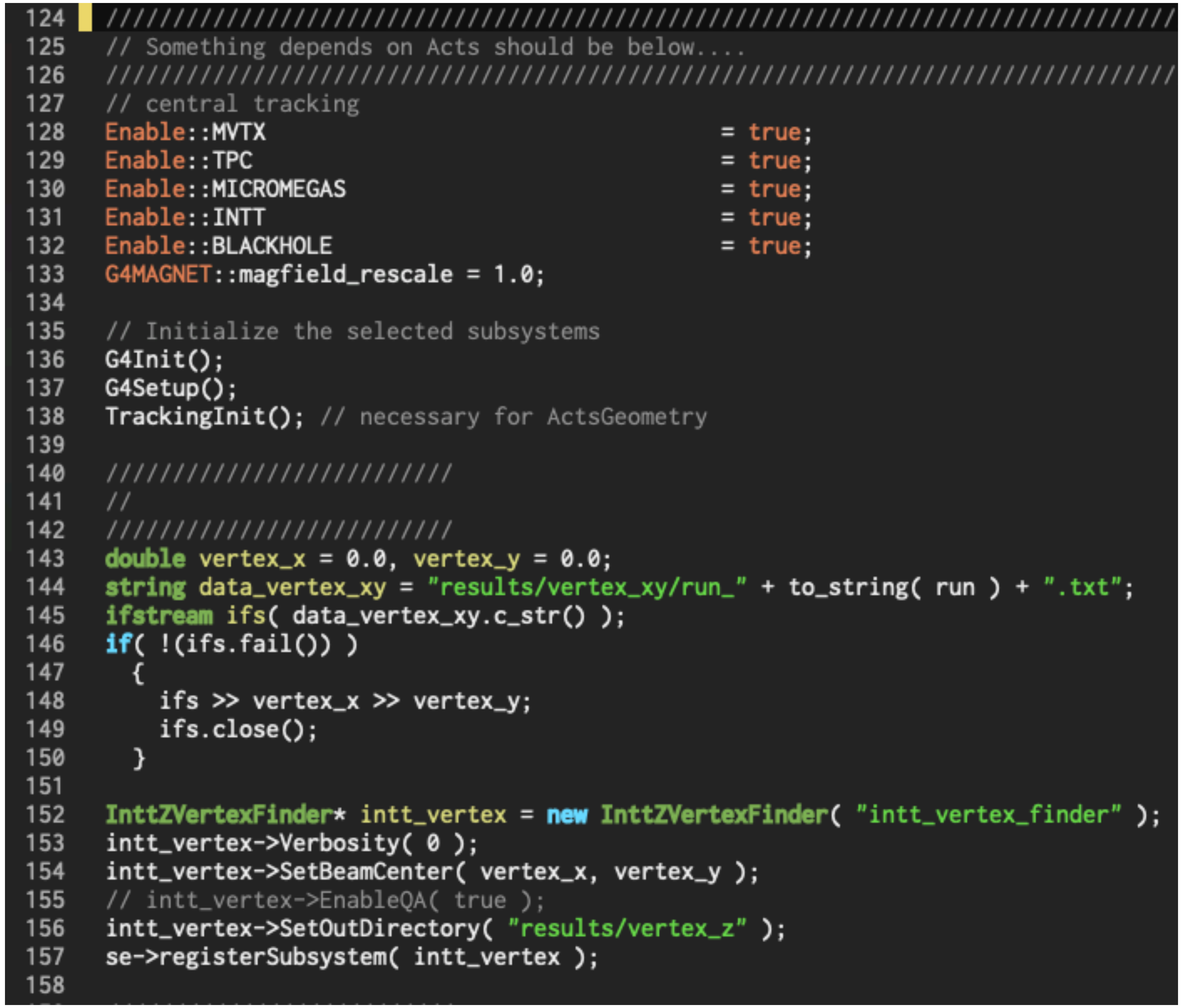
Thank you, @Christopher Pinkenburg ! That's extremely important information. If it's not documented yet, I want to write it down. Where is a good place?

Christopher Pinkenburg 02:00

We don't have such a place yet (means we can put it where we want). We have a wiki section for analysis (with just two entries). I think having this under there makes the most sense. Software is more infra structure while the above is really important for analysis:
https://wiki.sphenix.bnl.gov/index.php?title=SPHENIX#Analysis

Vertex (z のみ) 再構成

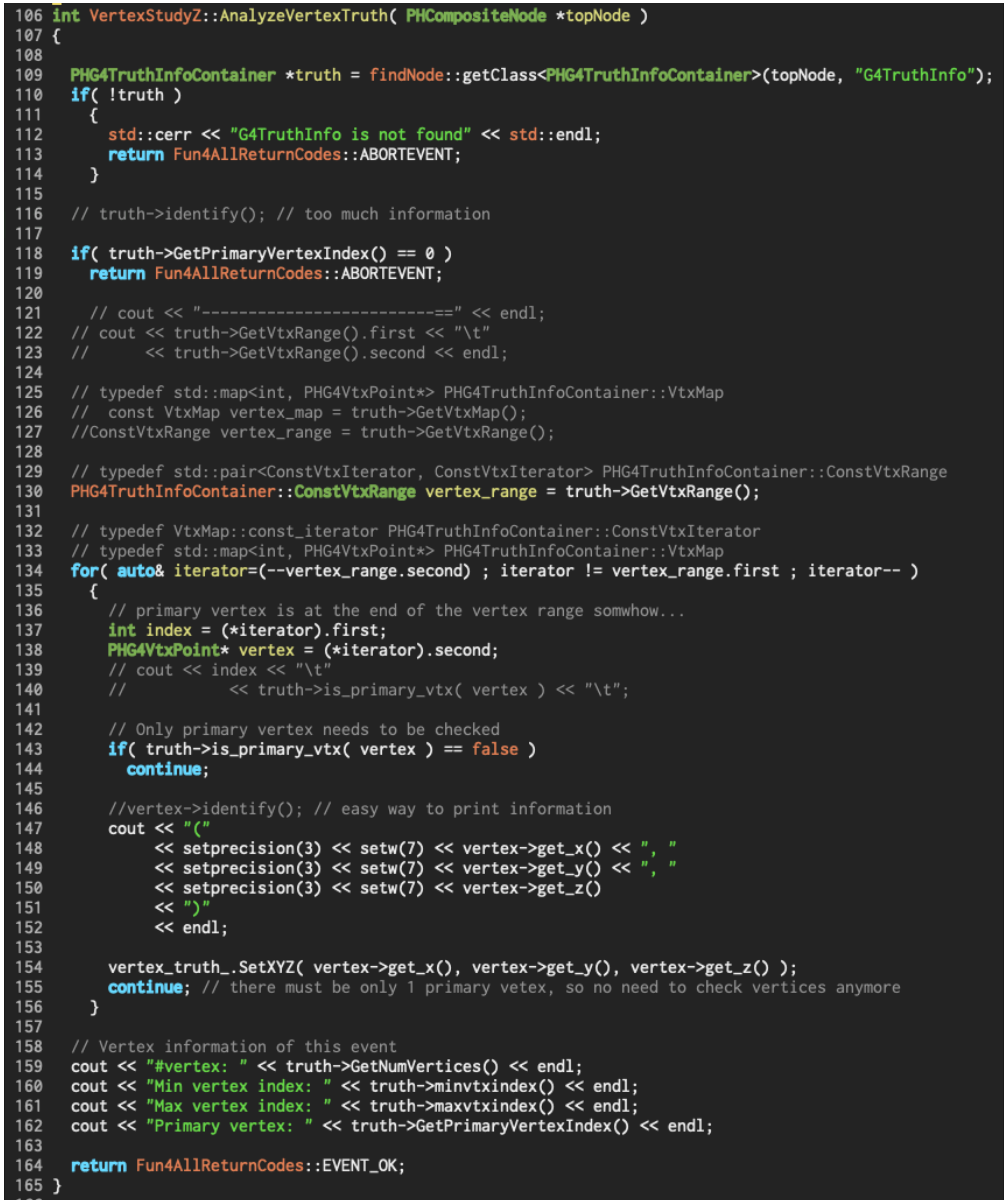
- MC truth, MBD 再構成の結果はすでに DST に入っている
- INTT 再構成は自分で行う必要がある
- MC の解析では簡単のため，ビーム衝突点の x, y 座標を 0 cm とした
- 柳川くんに提供した INTT 再構成では，一応 InttXTVertexFinder を動かしてランごとにビーム衝突点の x, y 座標を出したが，使い方が正しいかよくわからない



```
////////////////////////////////////////////////////////////////////////////////
// Something depends on Acts should be below....
////////////////////////////////////////////////////////////////////////////////
// central tracking
Enable::MVTX                                  = true;
Enable::TPC                                   = true;
Enable::MICROMEGAS                            = true;
Enable::INTT                                  = true;
Enable::BLACKHOLE                             = true;
G4MAGNET::magfield_rescale = 1.0;

// Initialize the selected subsystems
G4Init();
G4Setup();
TrackingInit(); // necessary for ActsGeometry

//////////////////////////
//
//////////////////////////
double vertex_x = 0.0, vertex_y = 0.0;
string data_vertex_xy = "results/vertex_xy/run_" + to_string( run ) + ".txt";
ifstream ifs( data_vertex_xy.c_str() );
if( !(ifs.fail()) )
  {
    ifs >> vertex_x >> vertex_y;
    ifs.close();
  }

InttZVertexFinder* intt_vertex = new InttZVertexFinder( "intt_vertex_finder" );
intt_vertex->Verbosity( 0 );
intt_vertex->SetBeamCenter( vertex_x, vertex_y );
// intt_vertex->EnableQA( true );
intt_vertex->SetOutDirectory( "results/vertex_z" );
se->registerSubsystem( intt_vertex );

```

Vertex (z のみ) 再構成: 解析モジュール MC truth

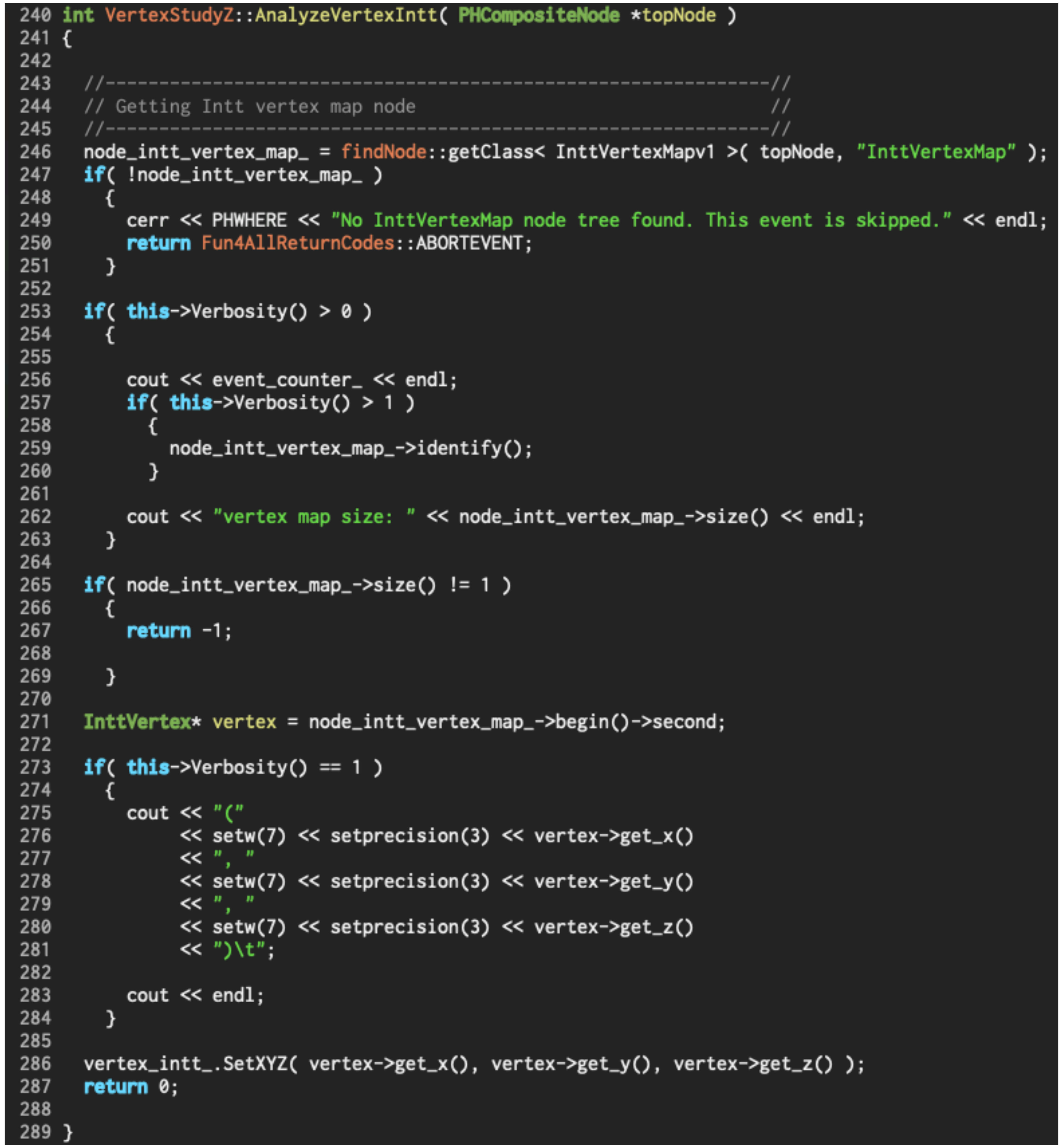
- 2024/07~08 に行った MBD MC データ解析のコードを流用



```
int VertexStudyZ::AnalyzeVertexTruth( PHCompositeNode *topNode )
{

  PHG4TruthInfoContainer *truth = findNode::getClass<PHG4TruthInfoContainer>(topNode, "G4TruthInfo");
  if( !truth )
    {
      std::cerr << "G4TruthInfo is not found" << std::endl;
      return Fun4AllReturnCodes::ABORTEVENT;
    }

  // truth->identify(); // too much information

  if( truth->GetPrimaryVertexIndex() == 0 )
    return Fun4AllReturnCodes::ABORTEVENT;

    // cout << "------------------------==" << endl;
  // cout << truth->GetVtxRange().first << "\t"
  //      << truth->GetVtxRange().second << endl;

  // typedef std::map<int, PHG4VtxPoint*> PHG4TruthInfoContainer::VtxMap
  //  const VtxMap vertex_map = truth->GetVtxMap();
  //ConstVtxRange vertex_range = truth->GetVtxRange();

  // typedef std::pair<ConstVtxIterator, ConstVtxIterator> PHG4TruthInfoContainer::ConstVtxRange
  PHG4TruthInfoContainer::ConstVtxRange vertex_range = truth->GetVtxRange();

  // typedef VtxMap::const_iterator PHG4TruthInfoContainer::ConstVtxIterator
  // typedef std::map<int, PHG4VtxPoint*> PHG4TruthInfoContainer::VtxMap
  for( auto& iterator=(--vertex_range.second) ; iterator != vertex_range.first ; iterator-- )
    {
      // primary vertex is at the end of the vertex range somwhow...
      int index = (*iterator).first;
      PHG4VtxPoint* vertex = (*iterator).second;
      // cout << index << "\t"
      //           << truth->is_primary_vtx( vertex ) << "\t";

      // Only primary vertex needs to be checked
      if( truth->is_primary_vtx( vertex ) == false )
        continue;

      //vertex->identify(); // easy way to print information
      cout << "("
           << setprecision(3) << setw(7) << vertex->get_x() << ", "
           << setprecision(3) << setw(7) << vertex->get_y() << ", "
           << setprecision(3) << setw(7) << vertex->get_z()
           << ")"
           << endl;

      vertex_truth_.SetXYZ( vertex->get_x(), vertex->get_y(), vertex->get_z() );
      continue; // there must be only 1 primary vetex, so no need to check vertices anymore
    }

  // Vertex information of this event
  cout << "#vertex: " << truth->GetNumVertices() << endl;
  cout << "Min vertex index: " << truth->minvtxindex() << endl;
  cout << "Max vertex index: " << truth->maxvtxindex() << endl;
  cout << "Primary vertex: " << truth->GetPrimaryVertexIndex() << endl;

  return Fun4AllReturnCodes::EVENT_OK;
}
```

Vertex (z のみ) 再構成: 解析モジュール
INTT vertex

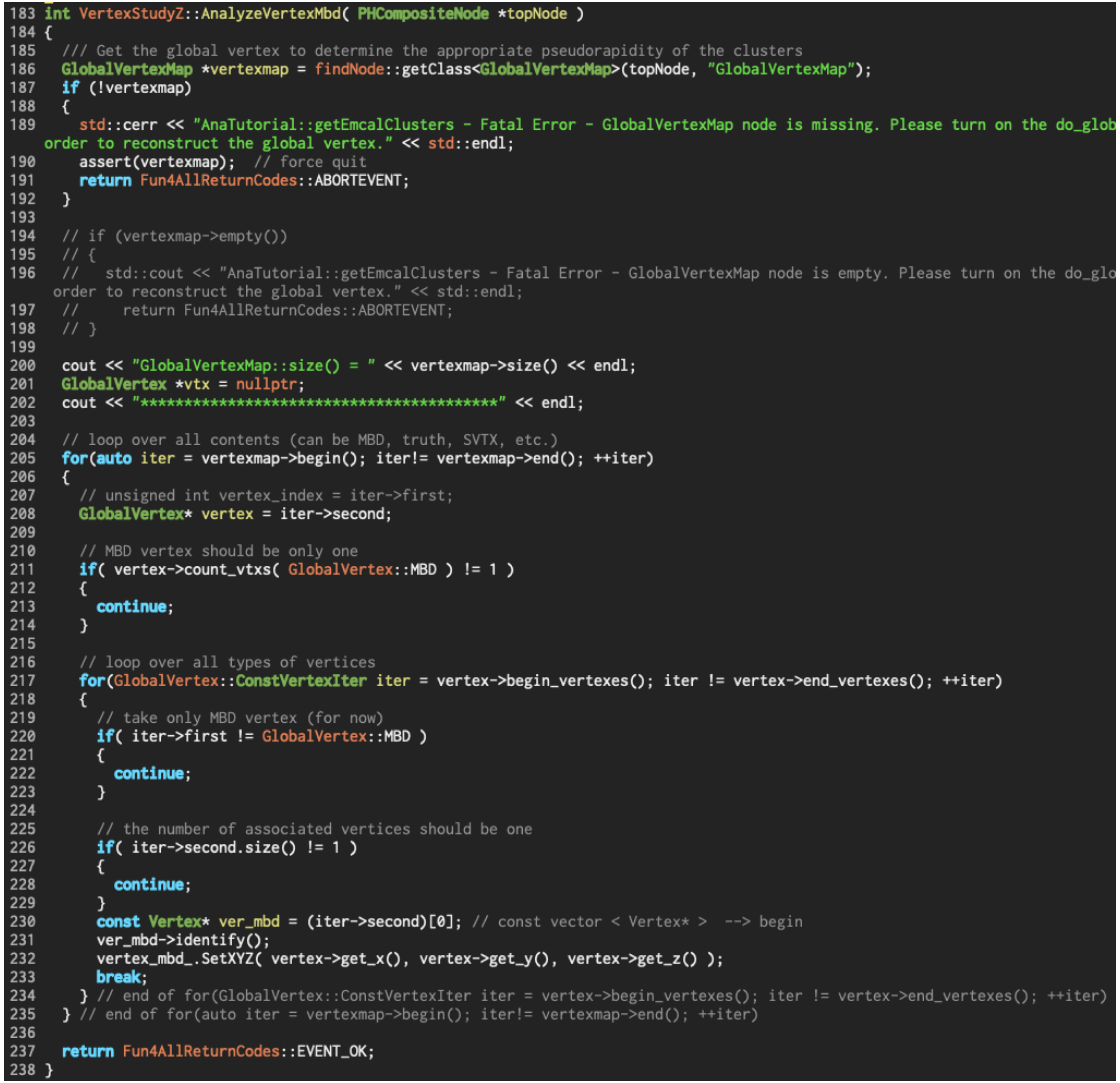
- 2024/07~08 に行った MBD MC データ解析のコードを流用



```cpp
int VertexStudyZ::AnalyzeVertexIntt( PHCompositeNode *topNode )
{

  //--------------------------------------------------------------//
  // Getting Intt vertex map node                                 //
  //--------------------------------------------------------------//
  node_intt_vertex_map_ = findNode::getClass< InttVertexMapv1 >( topNode, "InttVertexMap" );
  if( !node_intt_vertex_map_ )
    {
      cerr << PHWHERE << "No InttVertexMap node tree found. This event is skipped." << endl;
      return Fun4AllReturnCodes::ABORTEVENT;
    }

  if( this->Verbosity() > 0 )
    {

      cout << event_counter_ << endl;
      if( this->Verbosity() > 1 )
        {
          node_intt_vertex_map_->identify();
        }

      cout << "vertex map size: " << node_intt_vertex_map_->size() << endl;
    }

  if( node_intt_vertex_map_->size() != 1 )
    {
      return -1;

    }

  InttVertex* vertex = node_intt_vertex_map_->begin()->second;

  if( this->Verbosity() == 1 )
    {
      cout << "("
           << setw(7) << setprecision(3) << vertex->get_x()
           << ", "
           << setw(7) << setprecision(3) << vertex->get_y()
           << ", "
           << setw(7) << setprecision(3) << vertex->get_z()
           << ")\t";

      cout << endl;
    }

  vertex_intt_.SetXYZ( vertex->get_x(), vertex->get_y(), vertex->get_z() );
  return 0;

}
```

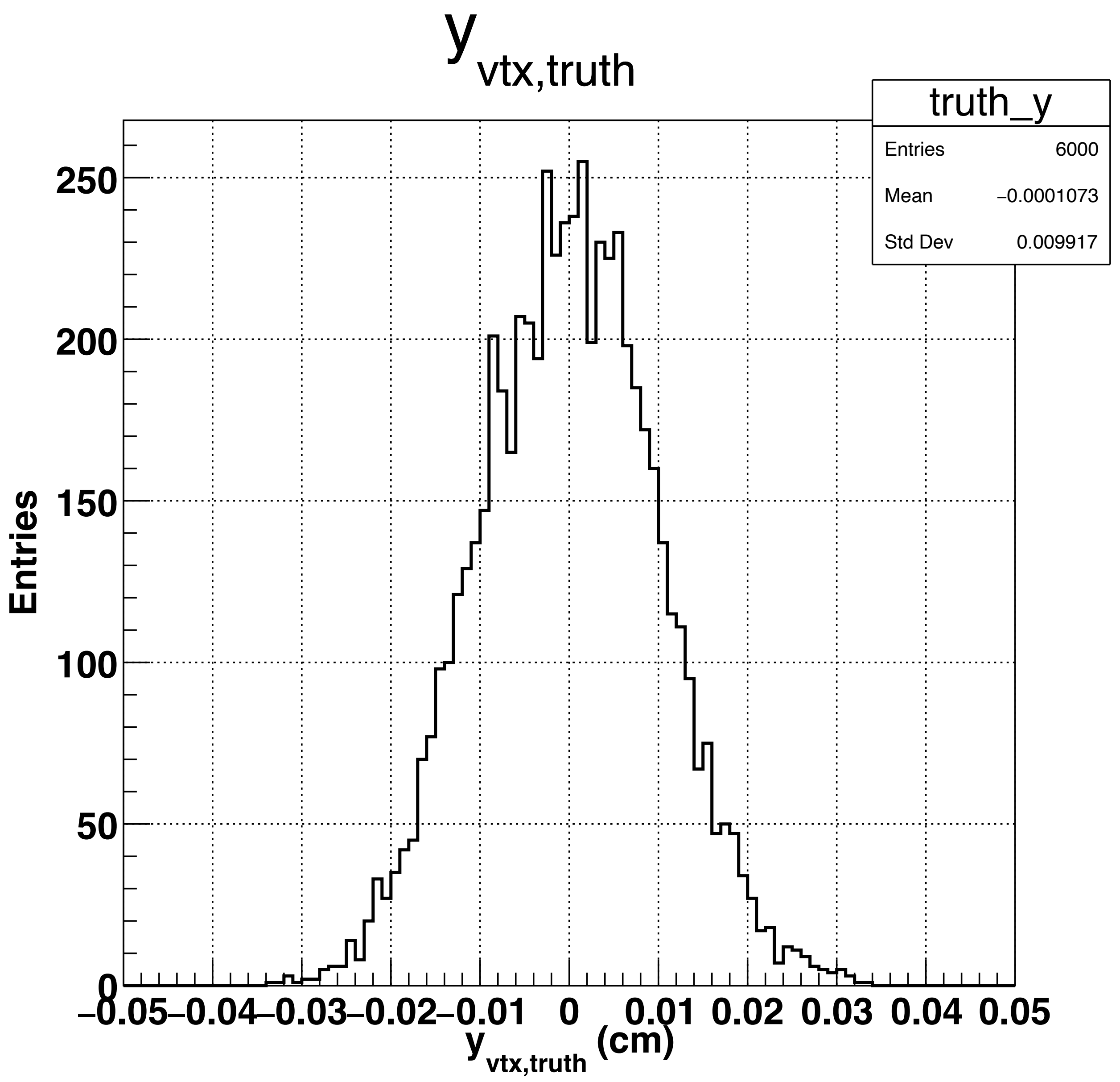
Vertex (z のみ) 再構成: 解析モジュール MBD vertex

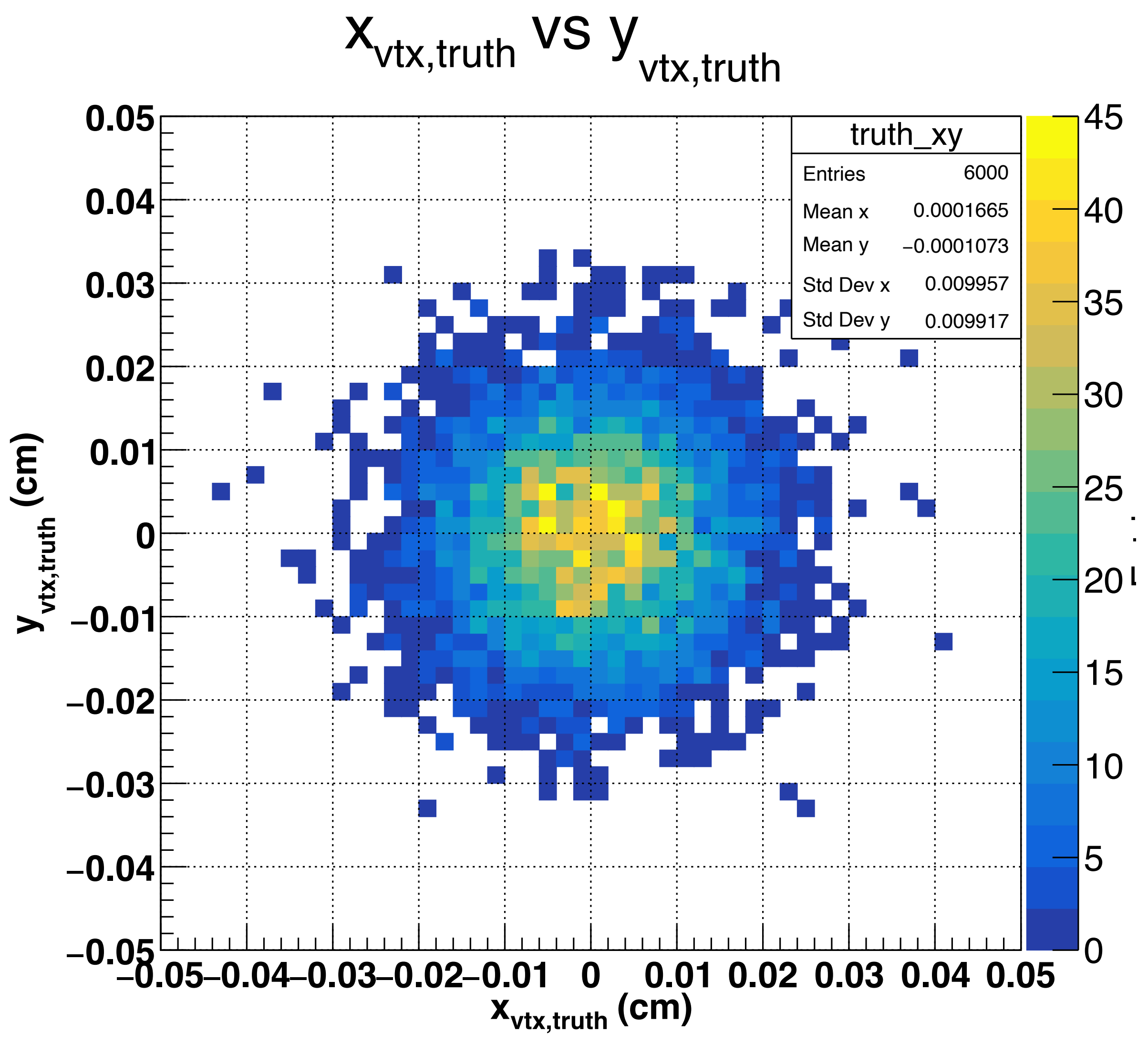
- GlobalVertex の使い方に苦戦したが，多分合っているはず
- GlobalVertex::identify を参考にした

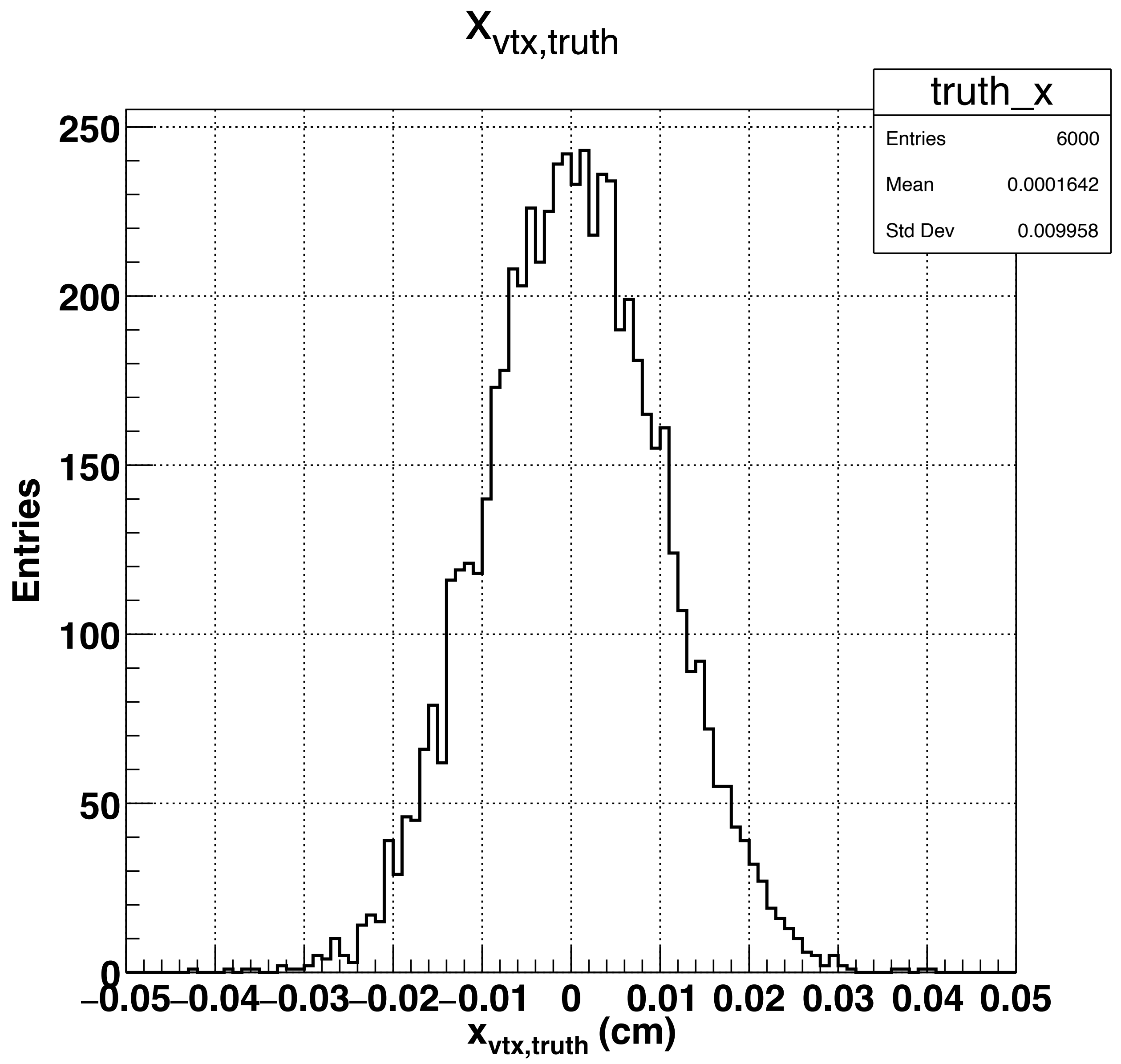


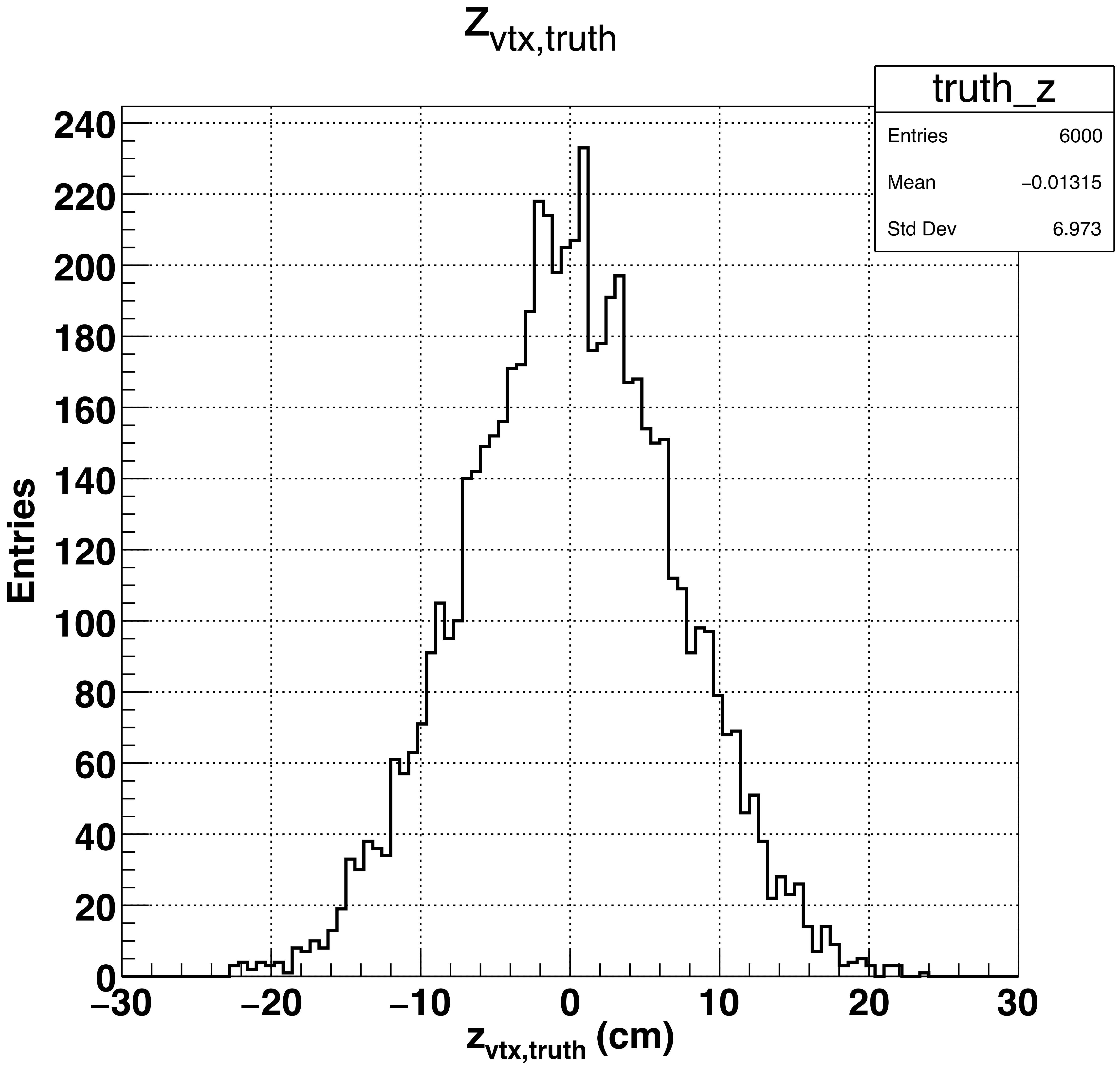
```cpp
int VertexStudyZ::AnalyzeVertexMbd( PHCompositeNode *topNode )
{
  /// Get the global vertex to determine the appropriate pseudorapidity of the clusters
  GlobalVertexMap *vertexmap = findNode::getClass<GlobalVertexMap>(topNode, "GlobalVertexMap");
  if (!vertexmap)
  {
    std::cerr << "AnaTutorial::getEmcalClusters - Fatal Error - GlobalVertexMap node is missing. Please turn on the do_glob
order to reconstruct the global vertex." << std::endl;
    assert(vertexmap);  // force quit
    return Fun4AllReturnCodes::ABORTEVENT;
  }

  // if (vertexmap->empty())
  // {
  //   std::cout << "AnaTutorial::getEmcalClusters - Fatal Error - GlobalVertexMap node is empty. Please turn on the do_glo
 order to reconstruct the global vertex." << std::endl;
  //     return Fun4AllReturnCodes::ABORTEVENT;
  // }

  cout << "GlobalVertexMap::size() = " << vertexmap->size() << endl;
  GlobalVertex *vtx = nullptr;
  cout << "*****************************************" << endl;

  // loop over all contents (can be MBD, truth, SVTX, etc.)
  for(auto iter = vertexmap->begin(); iter!= vertexmap->end(); ++iter)
  {
    // unsigned int vertex_index = iter->first;
    GlobalVertex* vertex = iter->second;

    // MBD vertex should be only one
    if( vertex->count_vtxs( GlobalVertex::MBD ) != 1 )
    {
      continue;
    }

    // loop over all types of vertices
    for(GlobalVertex::ConstVertexIter iter = vertex->begin_vertexes(); iter != vertex->end_vertexes(); ++iter)
    {
      // take only MBD vertex (for now)
      if( iter->first != GlobalVertex::MBD )
      {
        continue;
      }

      // the number of associated vertices should be one
      if( iter->second.size() != 1 )
      {
        continue;
      }
      const Vertex* ver_mbd = (iter->second)[0]; // const vector < Vertex* >  --> begin
      ver_mbd->identify();
      vertex_mbd_.SetXYZ( vertex->get_x(), vertex->get_y(), vertex->get_z() );
      break;
    } // end of for(GlobalVertex::ConstVertexIter iter = vertex->begin_vertexes(); iter != vertex->end_vertexes(); ++iter)
  } // end of for(auto iter = vertexmap->begin(); iter!= vertexmap->end(); ++iter)

  return Fun4AllReturnCodes::EVENT_OK;
}
```

結果: MC truth

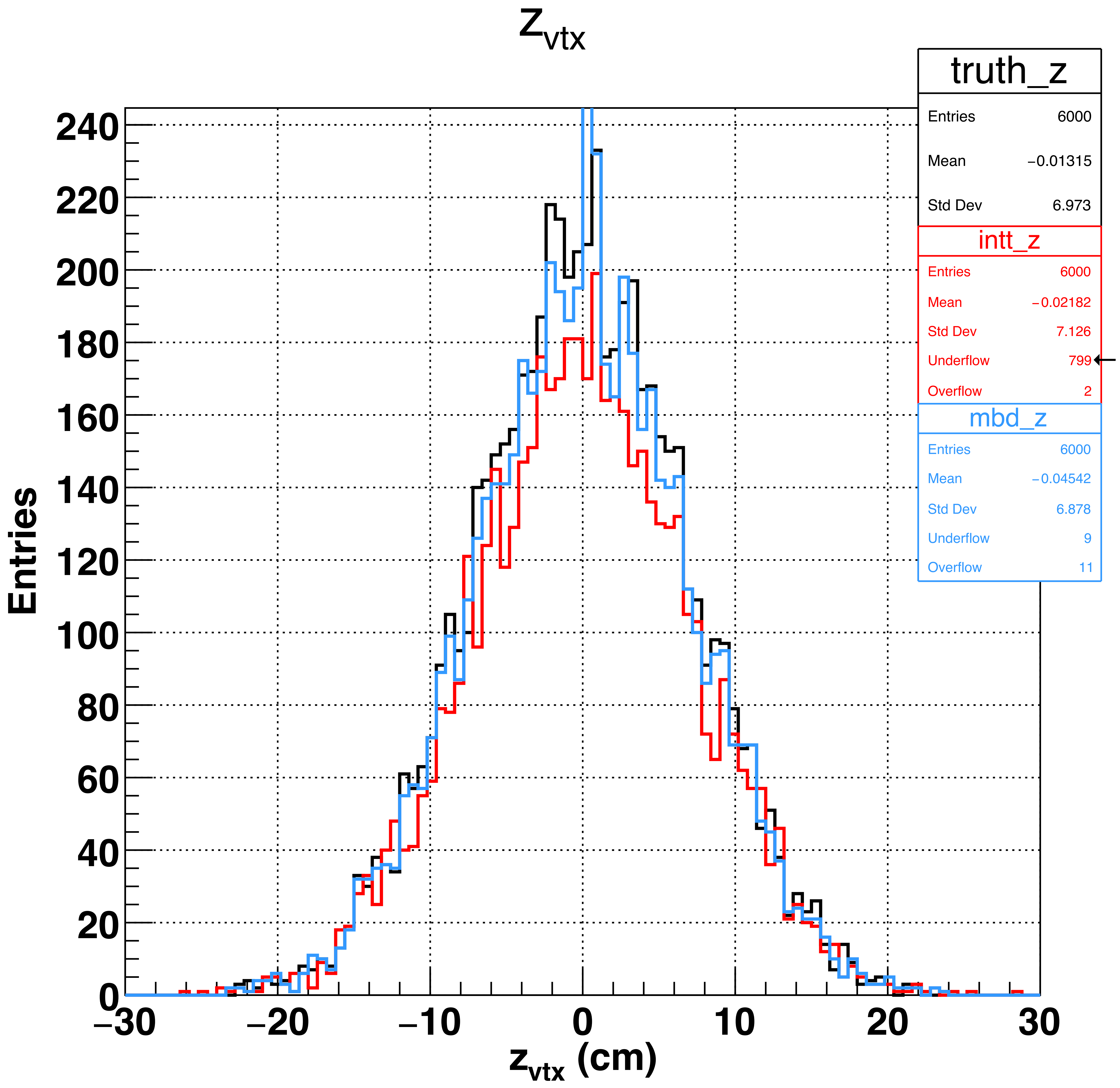

$y_{vtx,truth}$
truth_y
Entries 6000
Mean −0.0001073
Std Dev 0.009917
Entries
0
50
100
150
200
250
−0.05 −0.04 −0.03 −0.02 −0.01 0 0.01 0.02 0.03 0.04 0.05
$y_{vtx,truth}$ (cm)


$x_{vtx,truth}$ vs $y_{vtx,truth}$
truth_xy
Entries 6000
Mean x 0.0001665
Mean y −0.0001073
Std Dev x 0.009957
Std Dev y 0.009917
$y_{vtx,truth}$ (cm)
0.05
0.04
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.04
−0.05
−0.05 −0.04 −0.03 −0.02 −0.01 0 0.01 0.02 0.03 0.04 0.05
$x_{vtx,truth}$ (cm)
45
40
35
30
25
20
15
10
5
0


$x_{vtx,truth}$
truth_x
Entries 6000
Mean 0.0001642
Std Dev 0.009958
Entries
0
50
100
150
200
250
−0.05 −0.04 −0.03 −0.02 −0.01 0 0.01 0.02 0.03 0.04 0.05
$x_{vtx,truth}$ (cm)

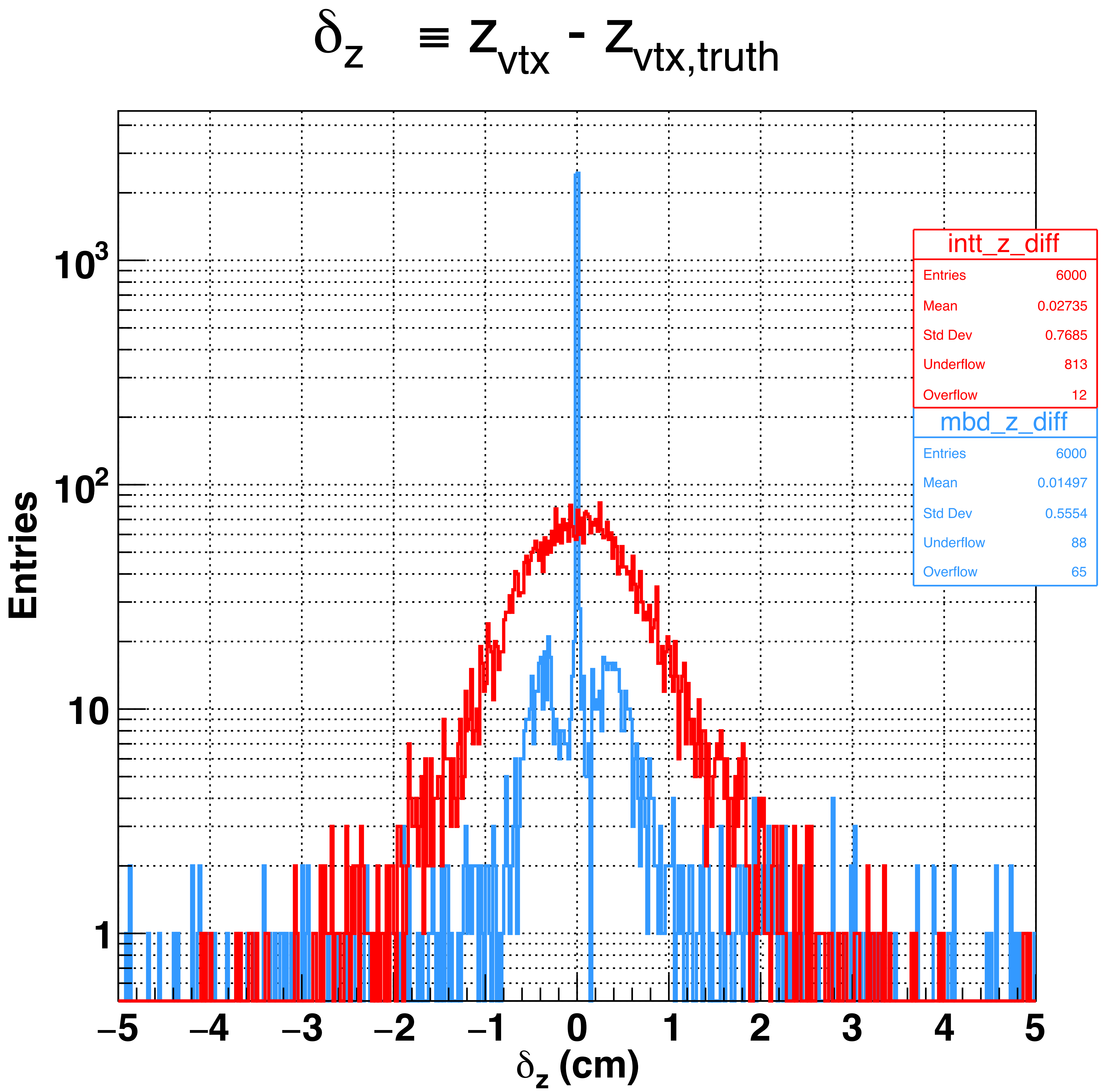

$z_{vtx,truth}$
truth_z
Entries 6000
Mean −0.01315
Std Dev 6.973
Entries
0
20
40
60
80
100
120
140
160
180
200
220
240
−30 −20 −10 0 10 20 30
$z_{vtx,truth}$ (cm)

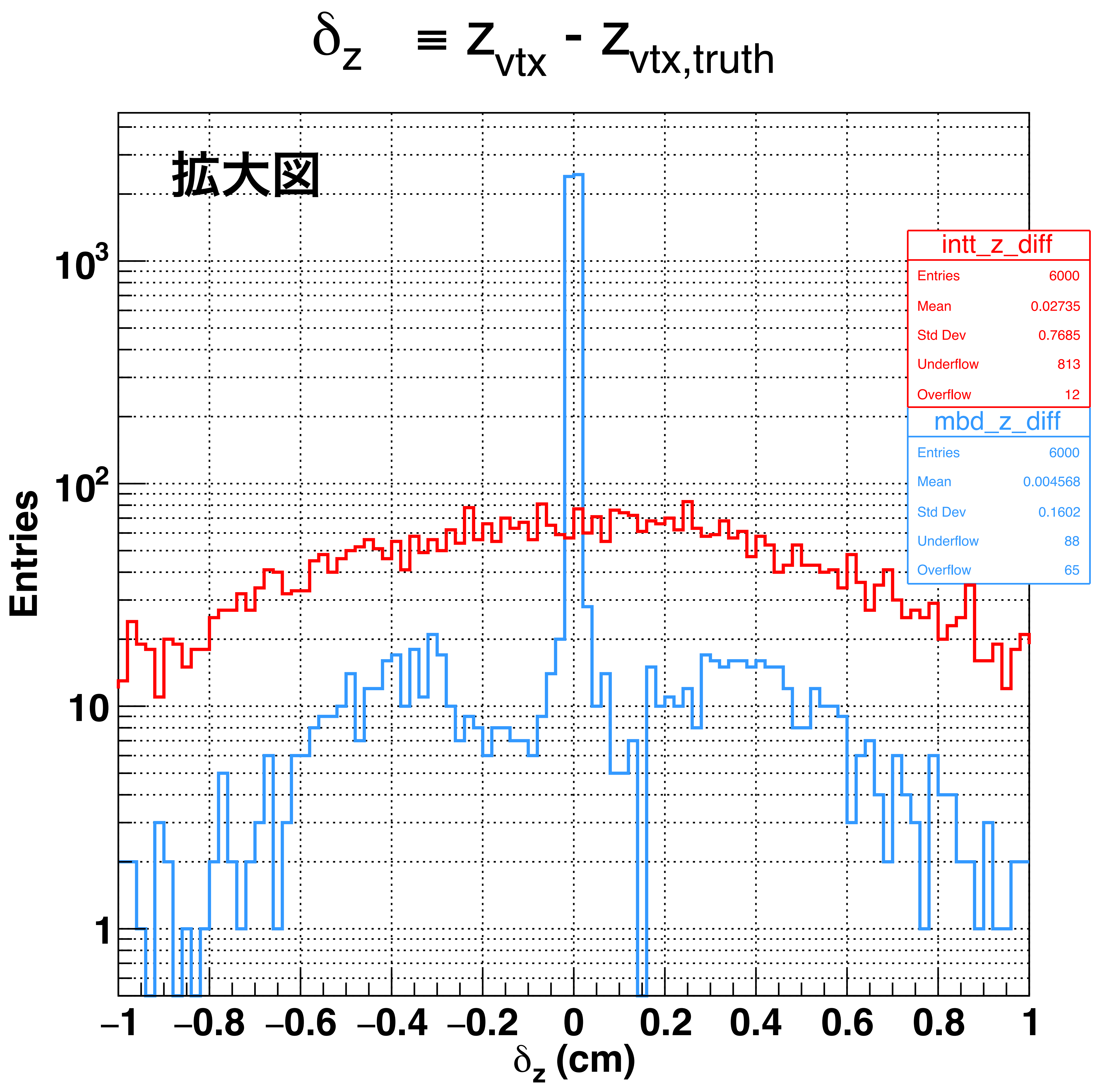
結果: z_{vtx}



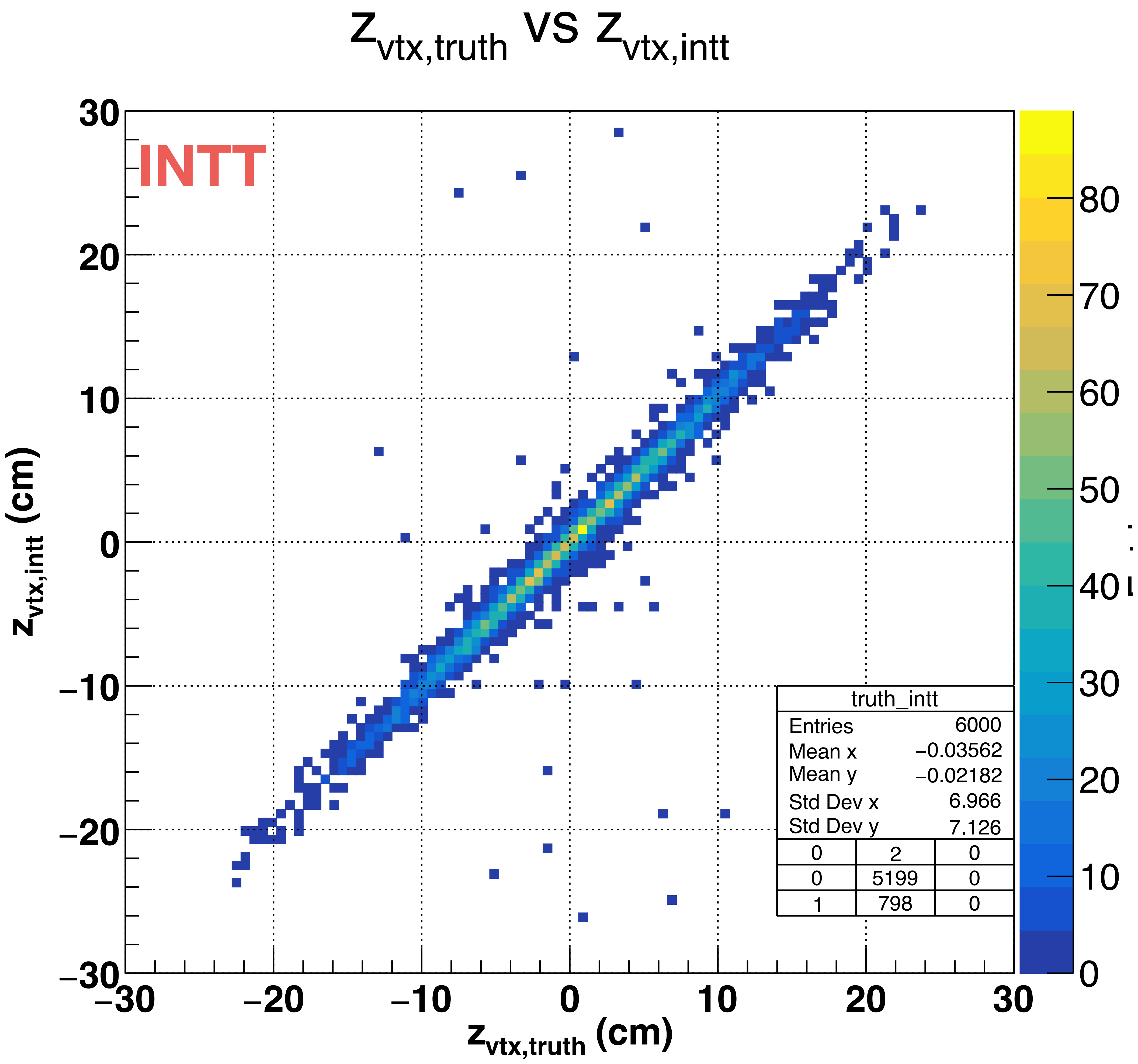
←INTT で再構成に失敗した数

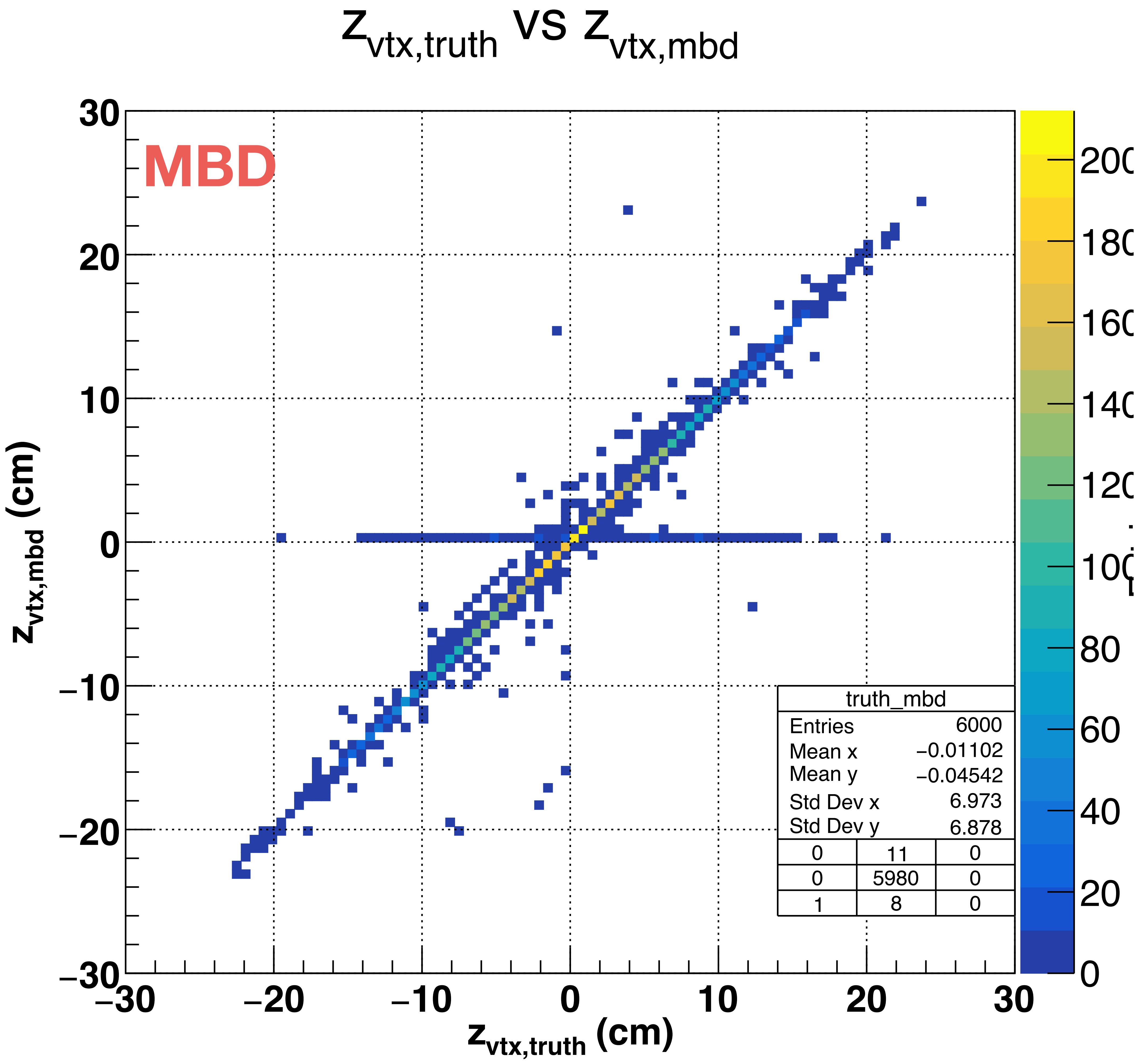
結果: $\delta_z = z_{vtx} - z_{vtx,\ truth}$



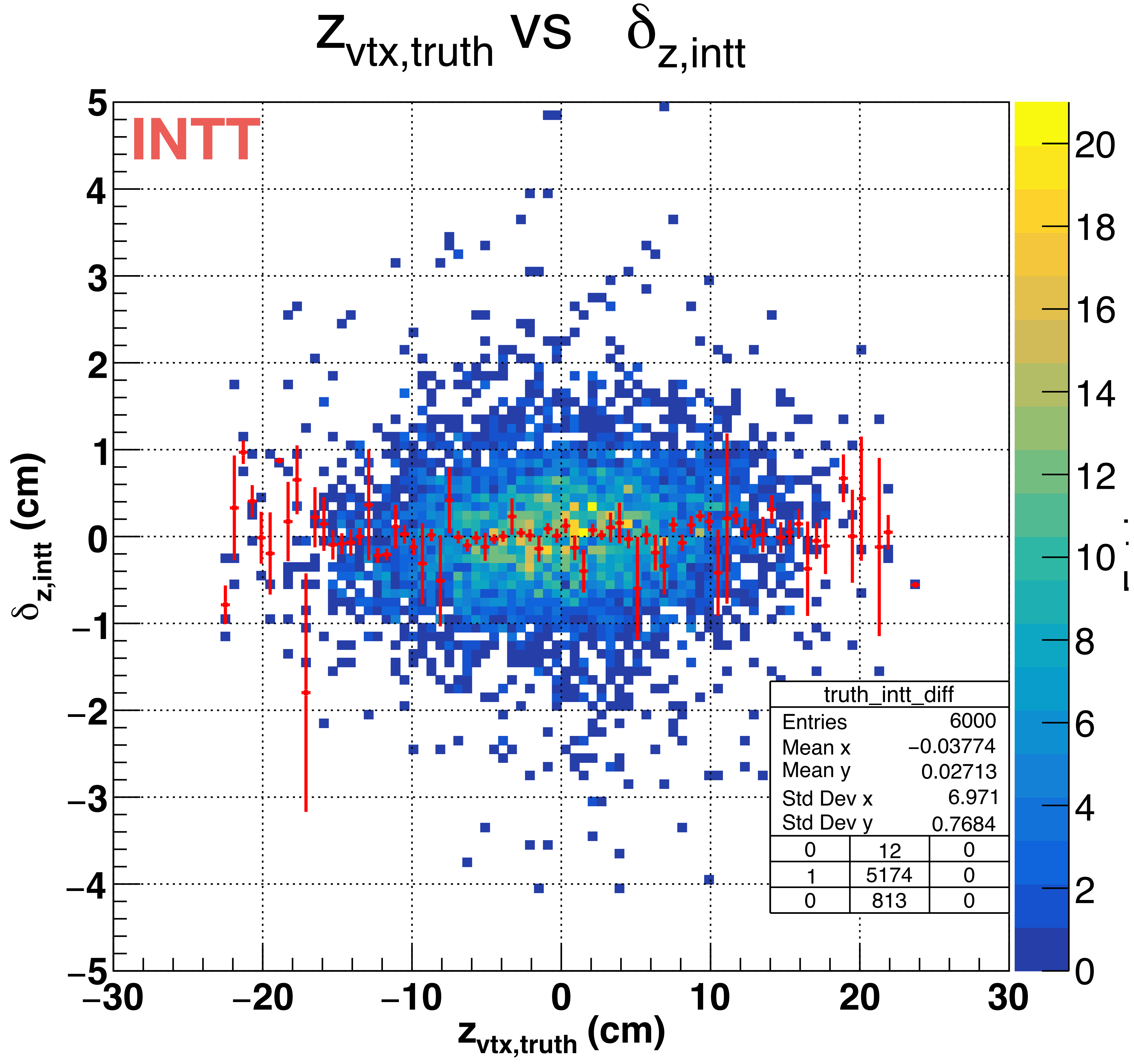


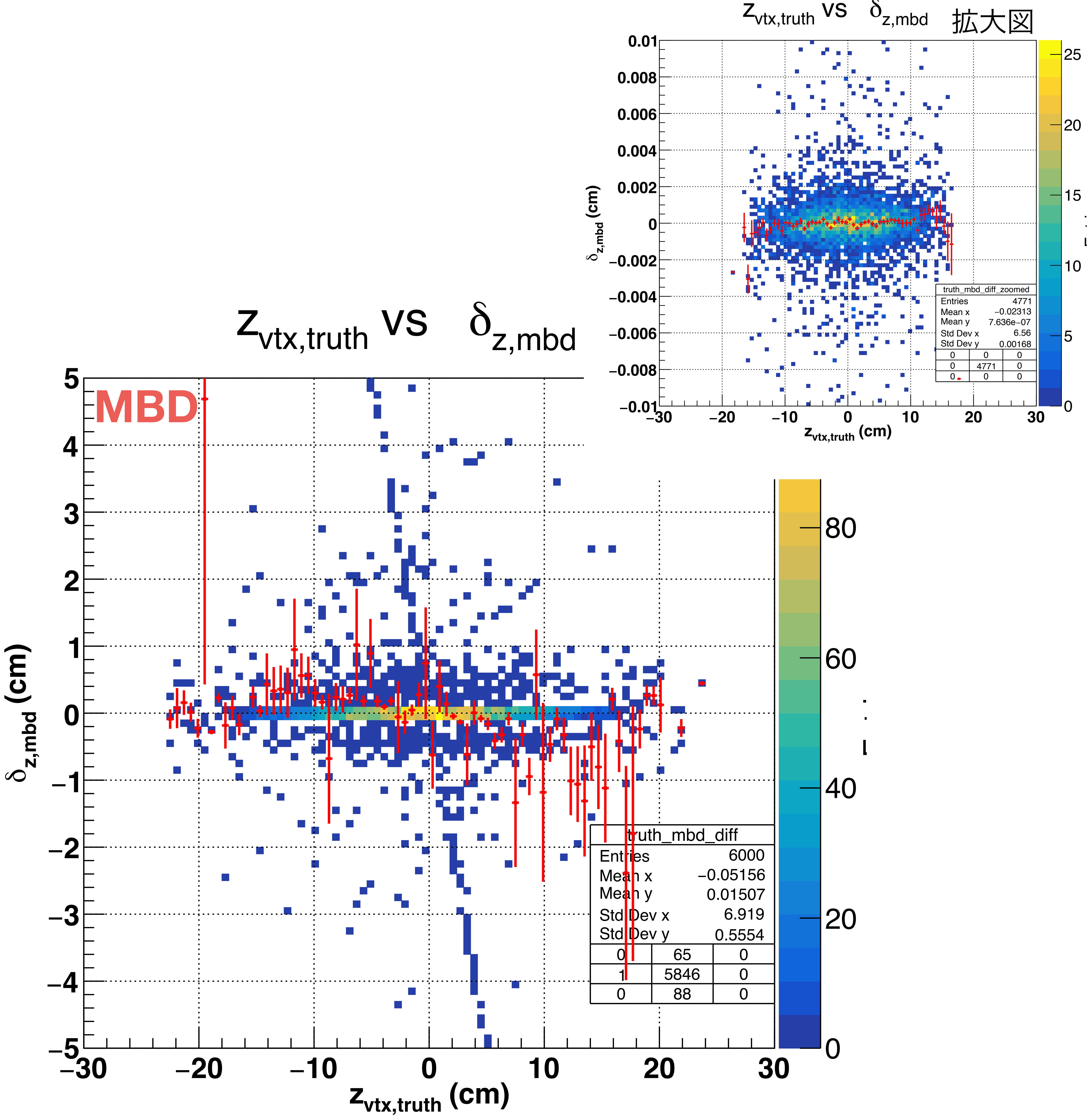
結果: $z_{vtx, truth}$ 対 z_{vtx}





結果: $z_{vtx,\,truth}$ 対 δ_z


$z_{vtx,truth}$ vs $\delta_{z,intt}$
INTT
$\delta_{z,intt}$ (cm)
$z_{vtx,truth}$ (cm)
truth_intt_diff
Entries 6000
Mean x −0.03774
Mean y 0.02713
Std Dev x 6.971
Std Dev y 0.7684


$z_{vtx,truth}$ vs $\delta_{z,mbd}$
MBD
$\delta_{z,mbd}$ (cm)
$z_{vtx,truth}$ (cm)
truth_mbd_diff
Entries 6000
Mean x −0.05156
Mean y 0.01507
Std Dev x 6.919
Std Dev y 0.5554
$z_{vtx,truth}$ vs $\delta_{z,mbd}$ 拡大図
$\delta_{z,mbd}$ (cm)
$z_{vtx,truth}$ (cm)
truth_mbd_diff_zoomed
Entries 4771
Mean x −0.02313
Mean y 7.636e−07
Std Dev x 6.56
Std Dev y 0.00168