

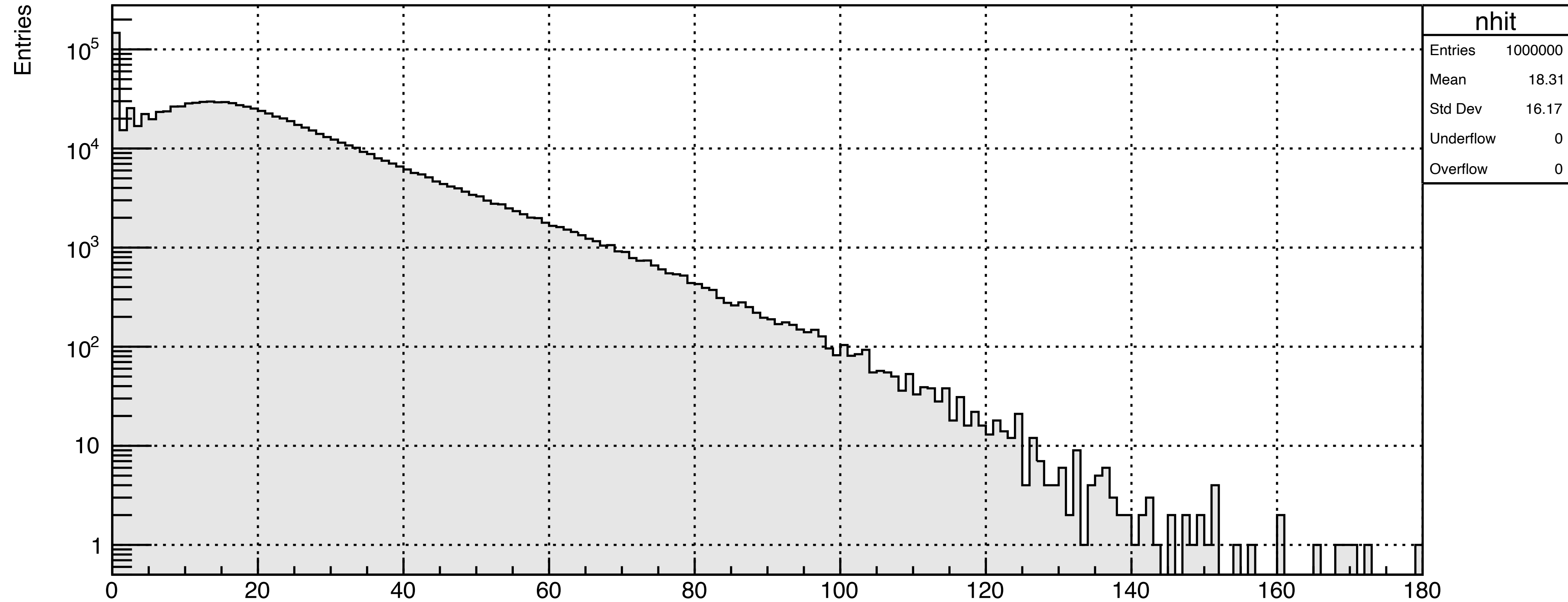
#INTT cluster distribution using MDC2 minimum bias MC data –Part 3–

Genki Nukazuka (RIKEN)

- [Part 1](#)
- [Part 2](#)

Status

~15% of minimum bias MC events have no INTT cluster



Updates: MC data

- 2 datasets were used to check #INTT cluster/event and MC truth information at the same time
 - /sphenix/lustre01/sphnxpro/mdc2/pythia8_pp_mb/trkrcluster/run0011/
 - /sphenix/lustre01/sphnxpro/mdc2/pythia8_pp_mb/nopileup/trkrhit/run0011
- 100k events were analyzed.

Updates: Analysis

- MC truth information
Fun4All analysis macro:

```
PHG4TruthInfoContainer* node_truth_;  
node_truth_ = findNode::getClass<PHG4TruthInfoContainer>(topNode, "G4TruthInfo");
```

```
PHG4TruthInfoContainer::ConstVtxRange vertex_range = node_truth_->GetVtxRange();  
vertex_num_MC_ = node_truth_->GetNumVertices();  
  
for( auto& iterator=(--vertex_range.second) ; iterator != vertex_range.first ; iterator-- )  
{  
    // primary vertex is at the end of the vertex range somehow...  
    int index = (*iterator).first;  
    PHG4VtxPoint* vertex = (*iterator).second;  
  
    // Only primary vertex needs to be checked  
    if( node_truth_->is_primary_vtx( vertex ) == false )  
        continue;  
  
    break; // there must be only 1 primary vertex, so no need to check vertices anymore  
}
```

Updates: Analysis

- Analyzing 2 DSTs at the same time
Fun4All macro:

```
Fun4AllServer *se = Fun4AllServer::instance();

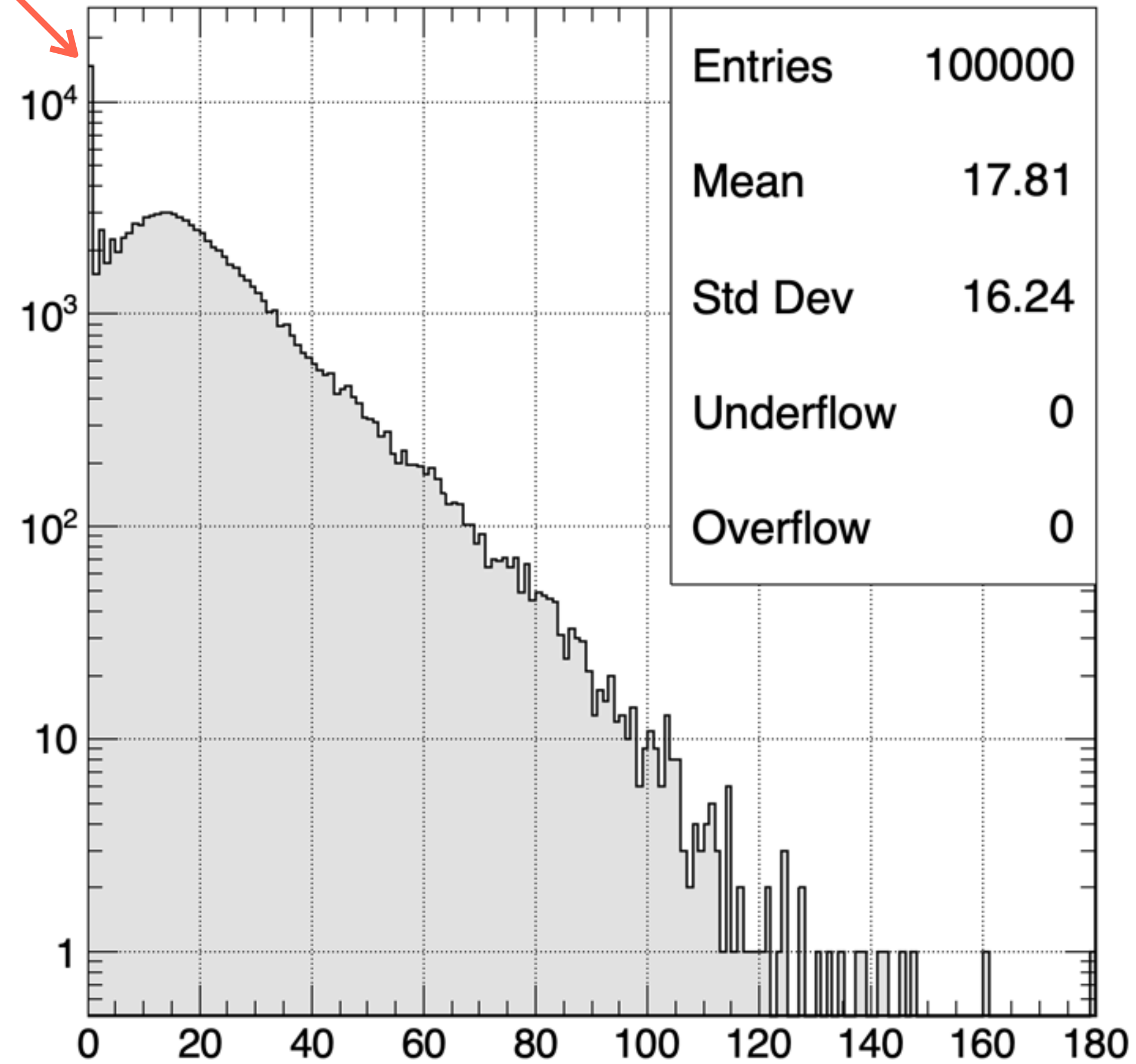
Fun4AllInputManager *in = new Fun4AllDstInputManager( "DSTin");
in->AddFile( /* something */ );
se->registerInputManager( in );

Fun4AllInputManager *in_truth = new Fun4AllDstInputManager( "DSTin_truth");
in_truth->AddFile( /* something */ );
se->registerInputManager( in_truth );
```

Results

~15%

consistent as the result shown in the last week



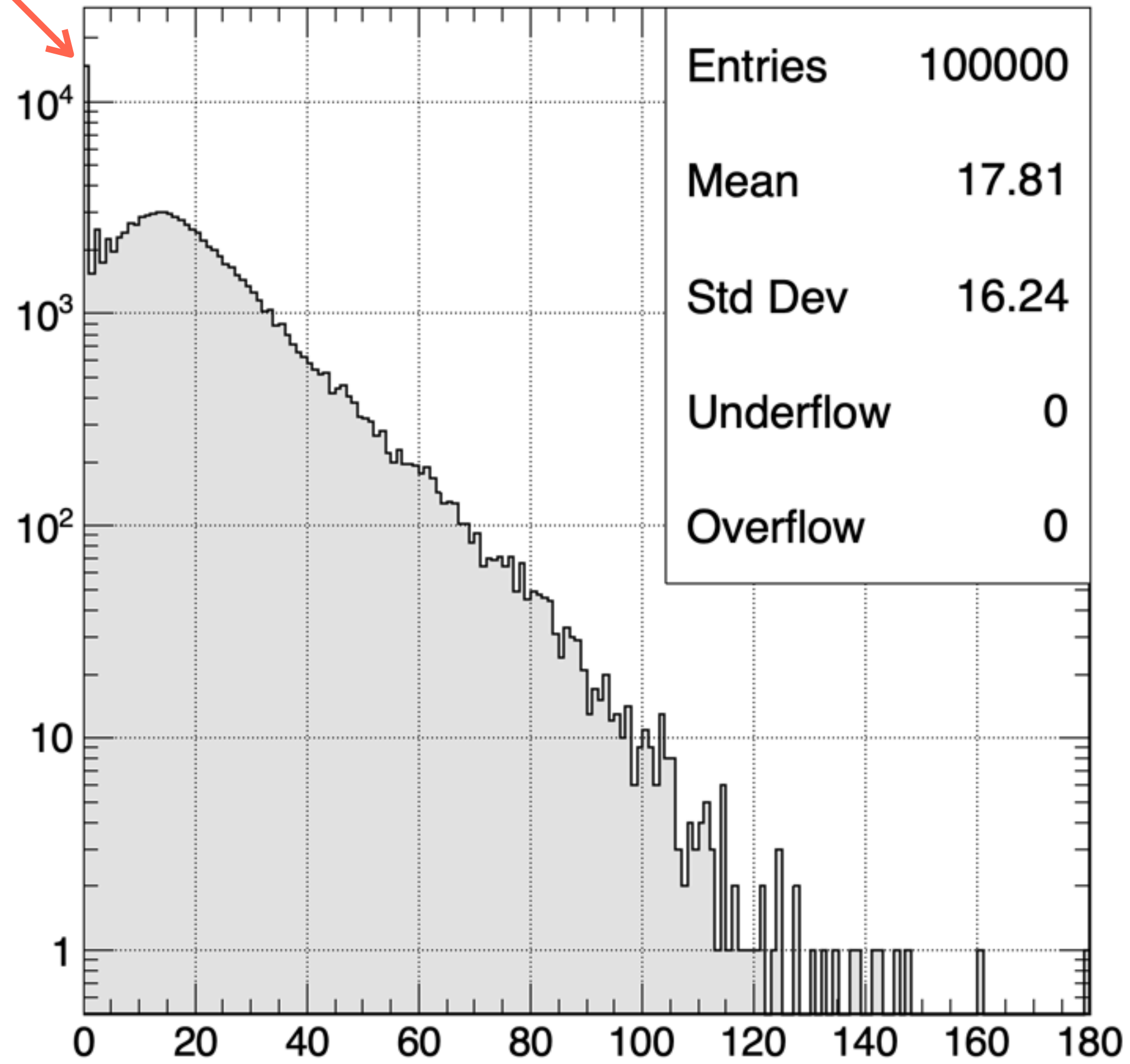
#INTT cluster/event

No cut applied.

Results

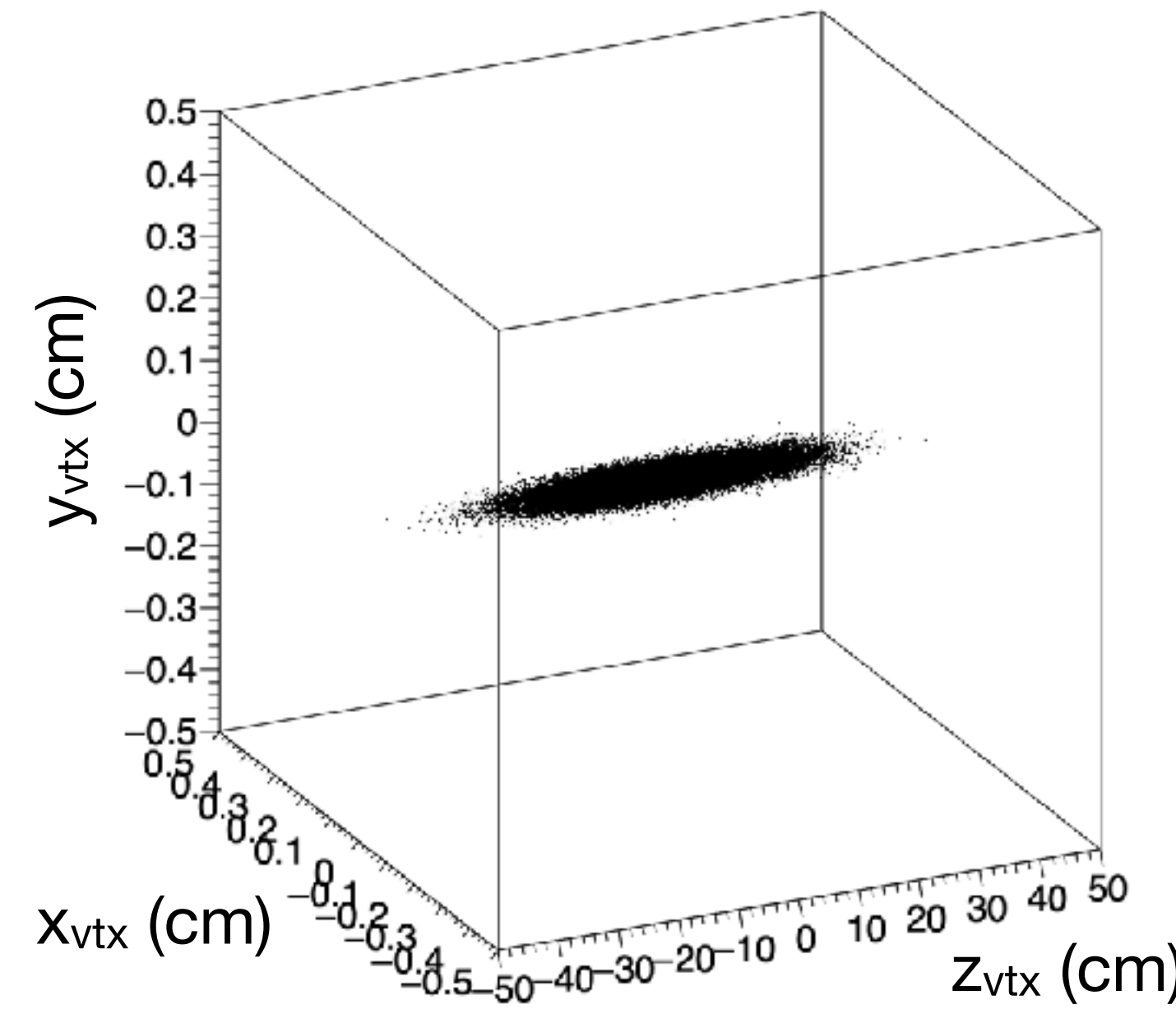
~15%

consistent as the result shown in the last week

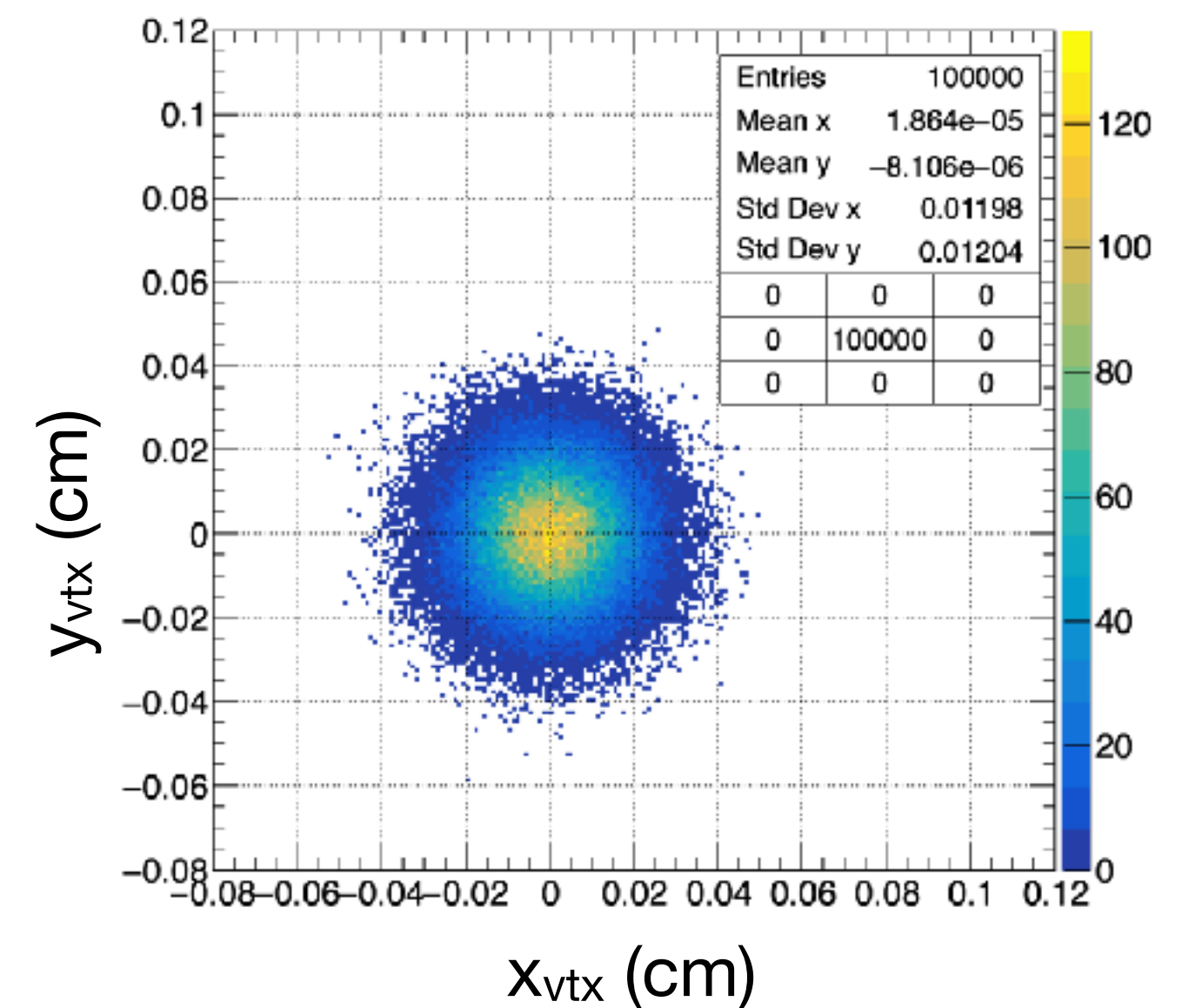
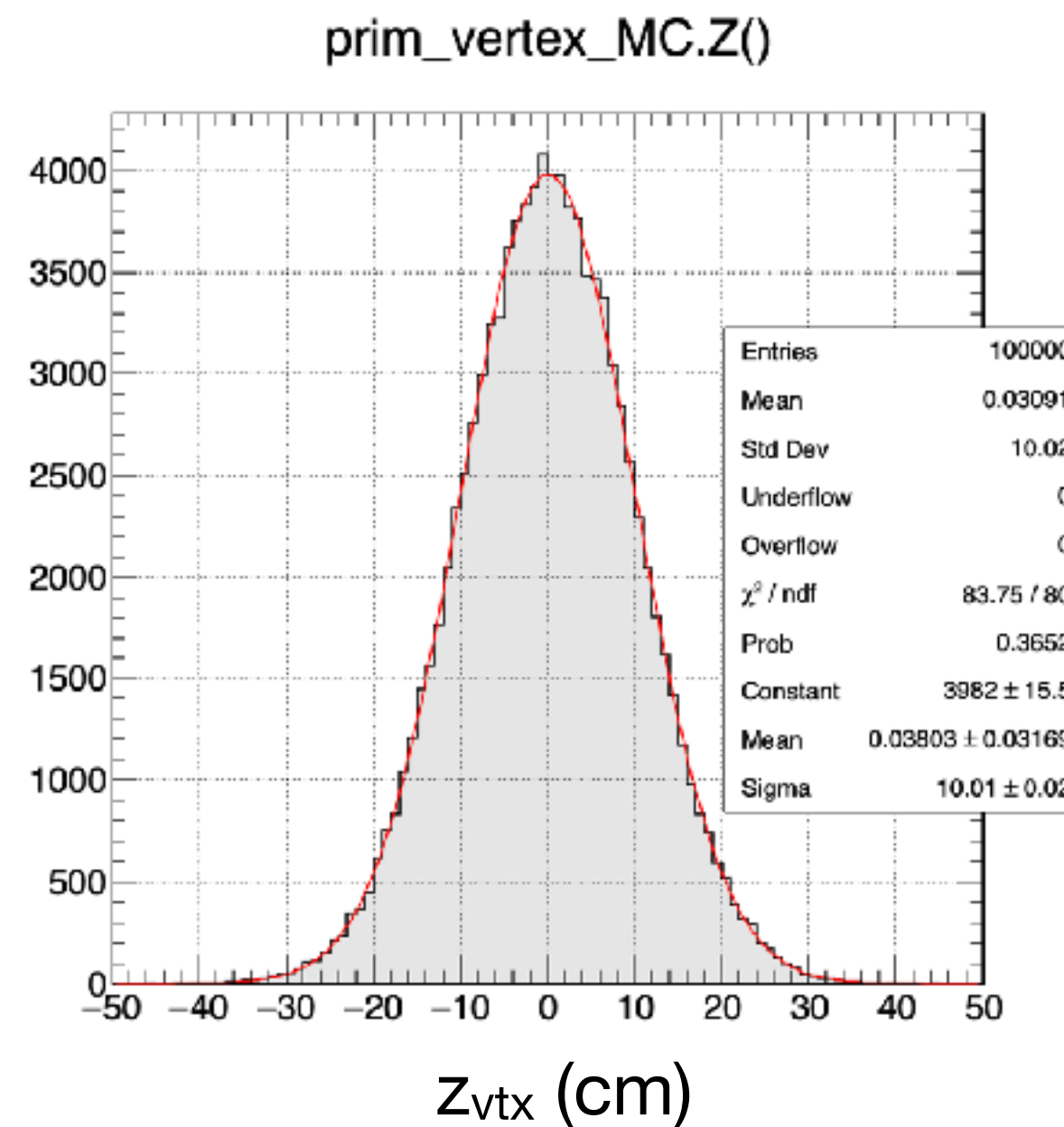


#INTT cluster/event

No cut applied.

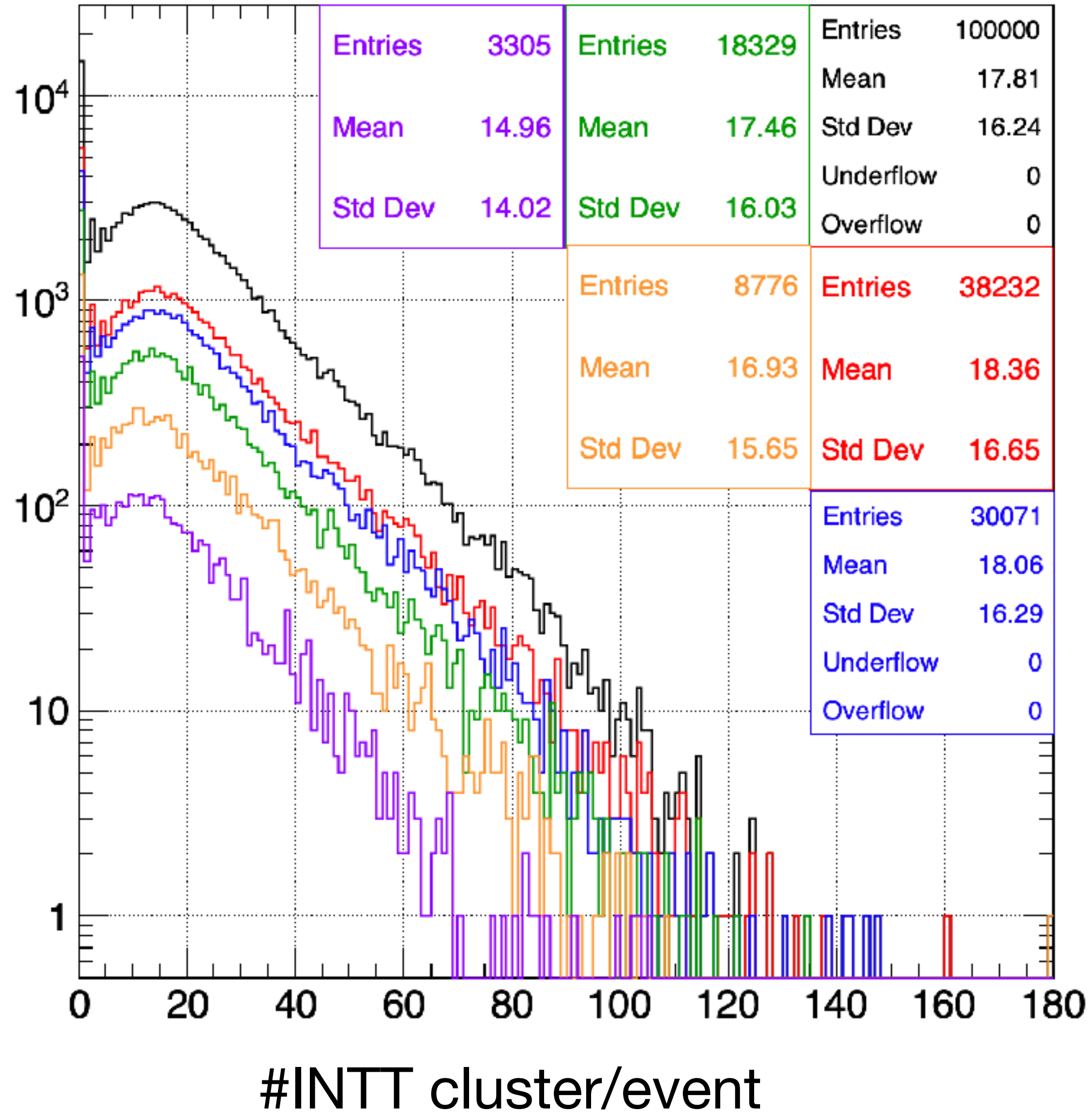


3D primary vertex (collision point) distribution

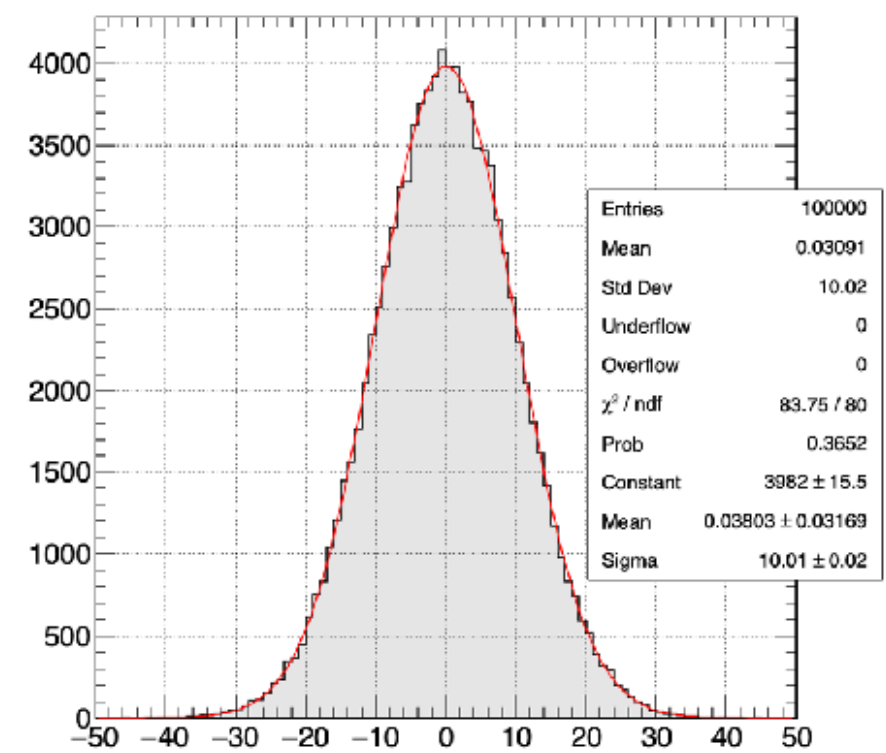


Results

- All
- $|z_{\text{vtx}}| < 5$ cm
- $5 < |z_{\text{vtx}}| < 10$ cm
- $10 < |z_{\text{vtx}}| < 15$ cm
- $15 < |z_{\text{vtx}}| < 20$ cm
- $20 < |z_{\text{vtx}}| < 25$ cm

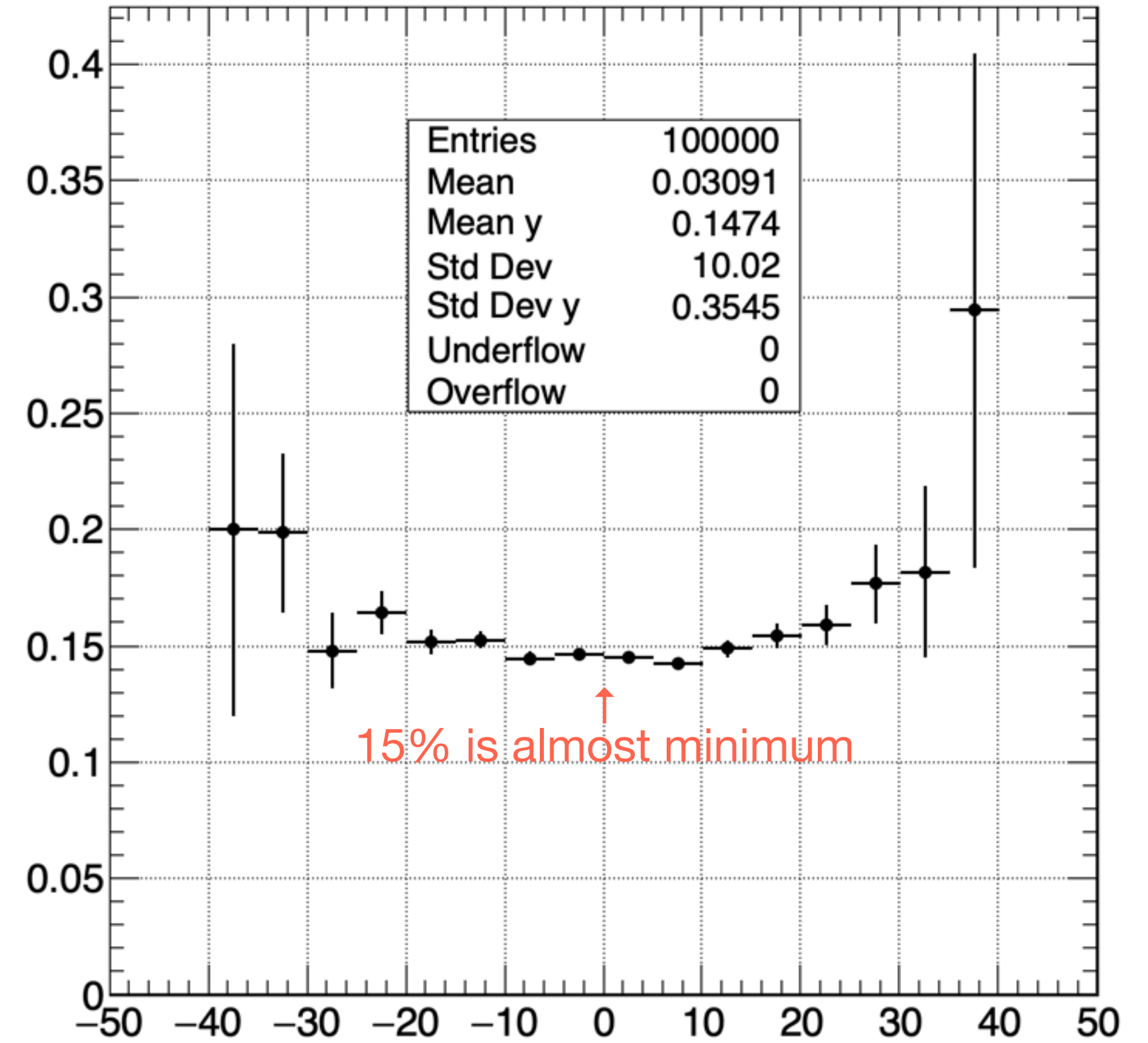
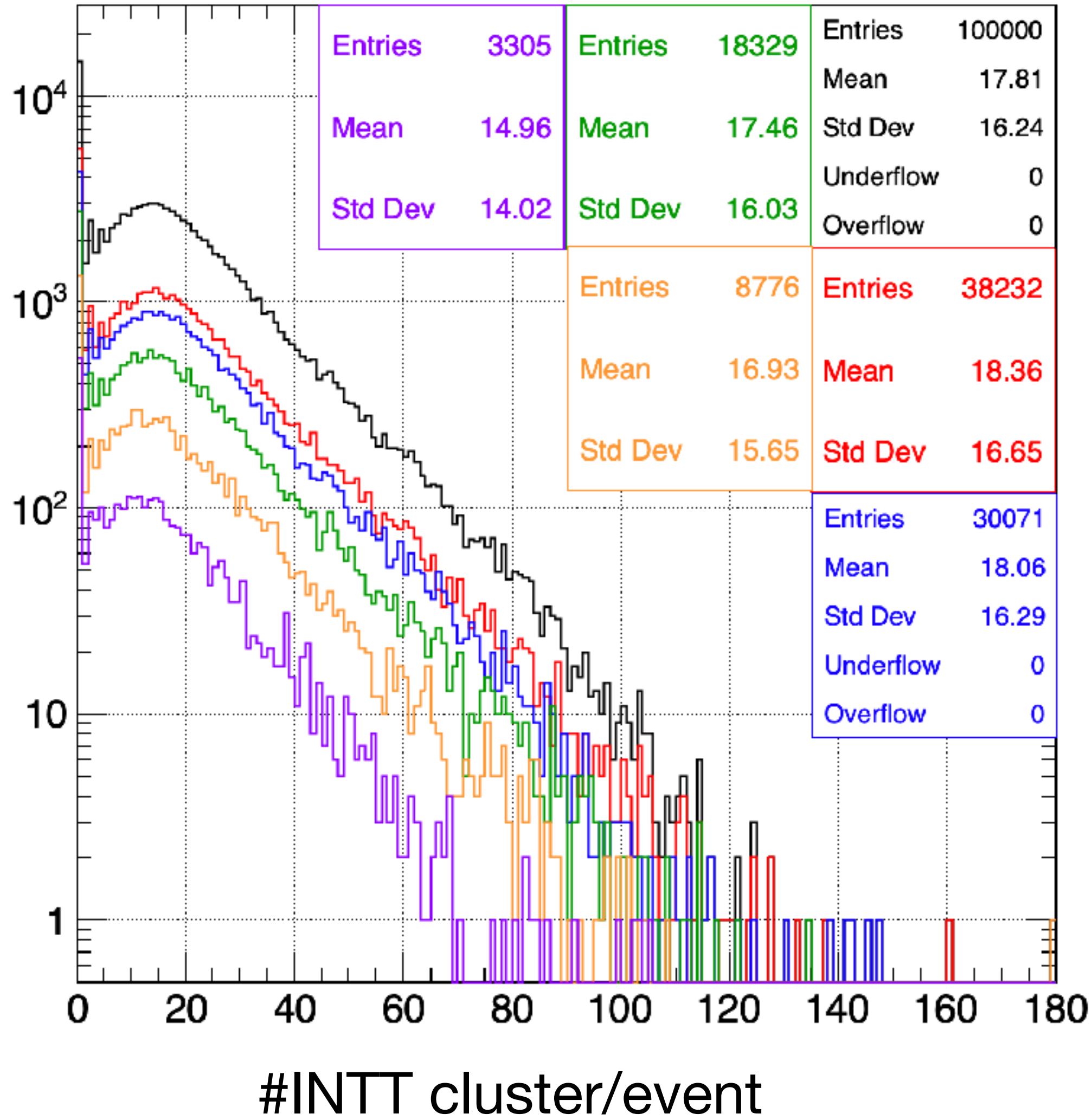


prim_vertex_MC.Z()



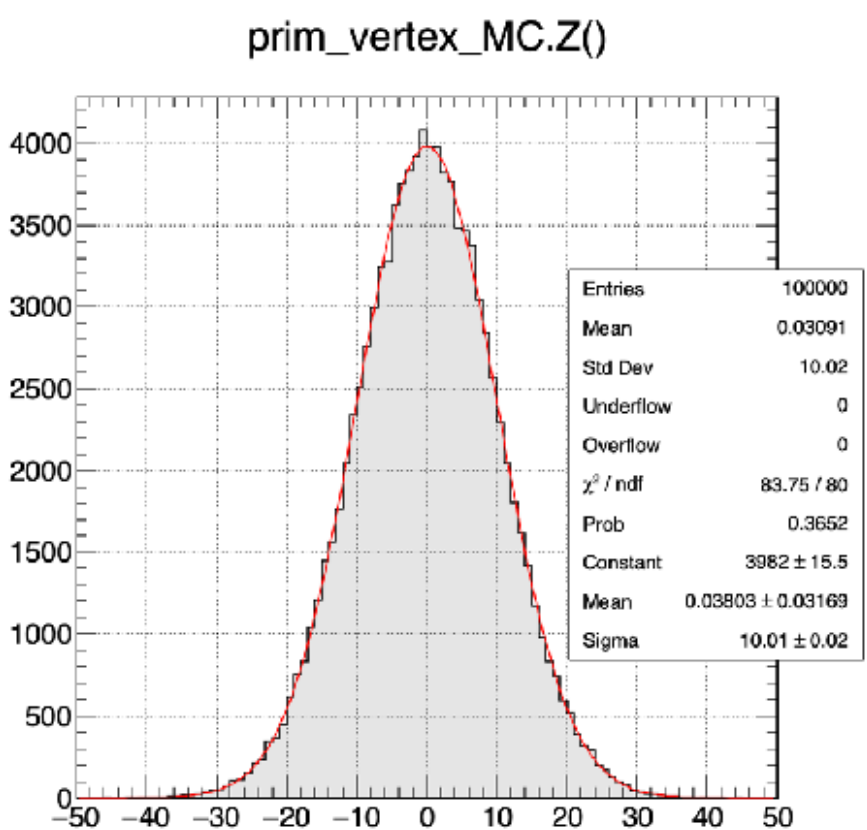
Results

- All
- $|z_{\text{vtx}}| < 5$ cm
- $5 < |z_{\text{vtx}}| < 10$ cm
- $10 < |z_{\text{vtx}}| < 15$ cm
- $15 < |z_{\text{vtx}}| < 20$ cm
- $20 < |z_{\text{vtx}}| < 25$ cm



Ratio of #INTT cluster/event = 0
as a function of z_{vtx}

The ratio of events without #INTT cluster is 15% and constant for $|z_{\text{vtx}}| < 20$ cm.



What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower

sPHENIX Analysis Software
Documentation for sPHENIX simulation software

Home page | Related Pages | Modules | Namespaces | **Classes** | Files | Examples

Class List | Class Index | Class Hierarchy | Class Members

PHG4TruthInfoContainer Class Reference

```
#include <coresoftware/blob/master/simulation/g4simulation/g4...>
```

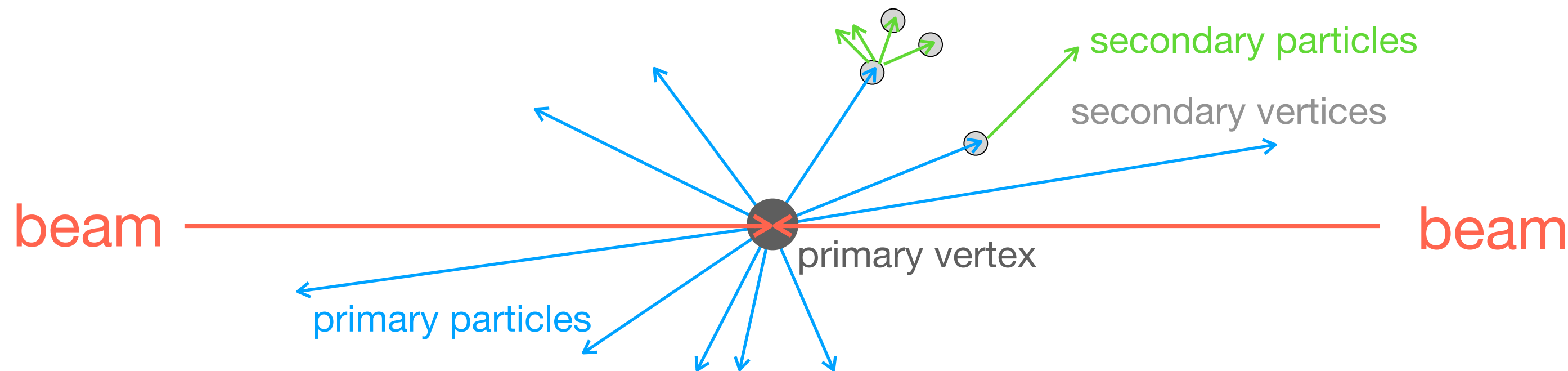
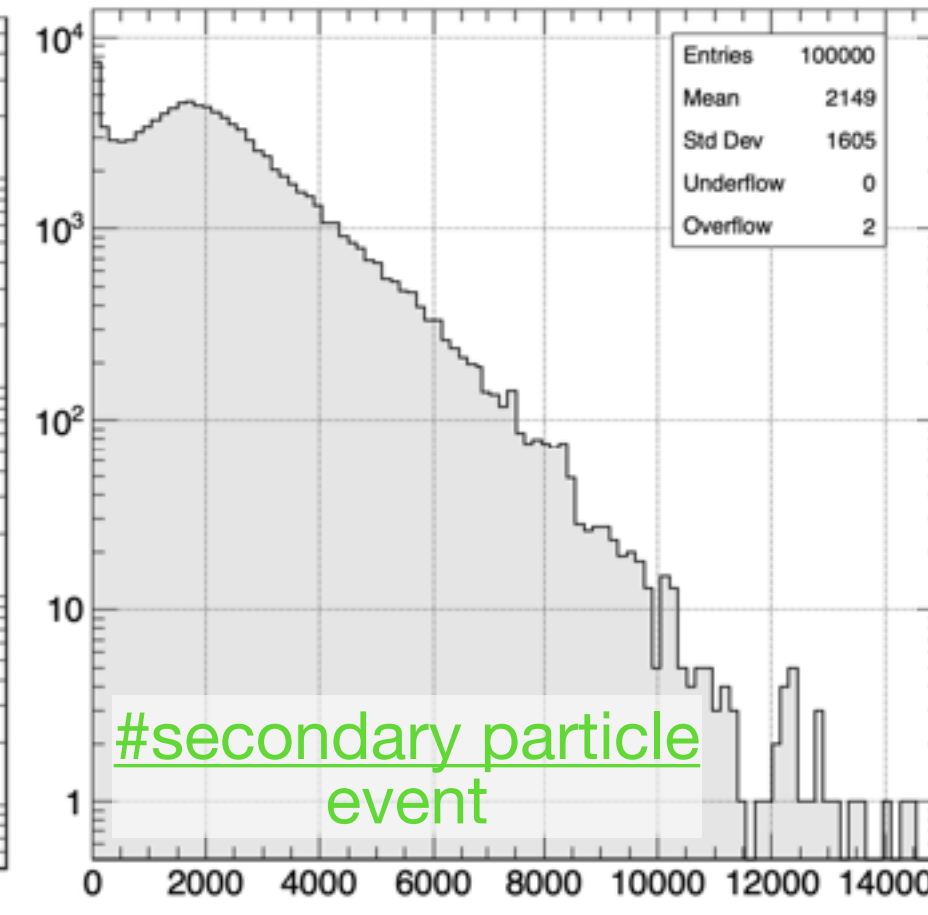
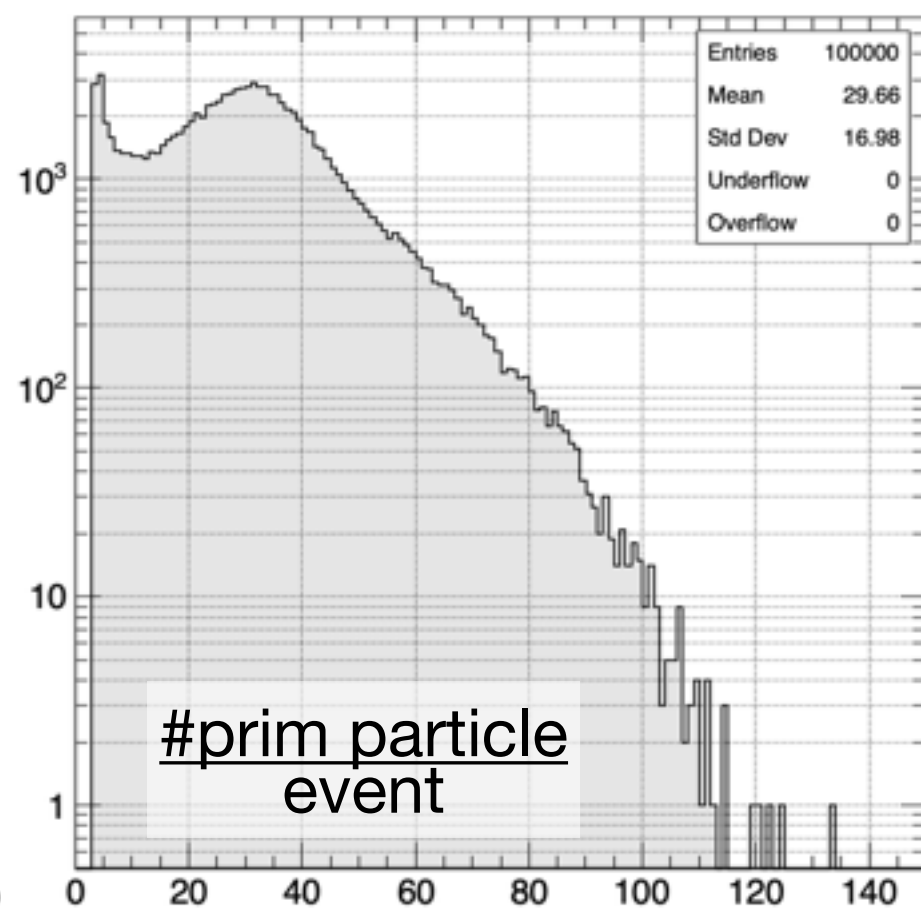
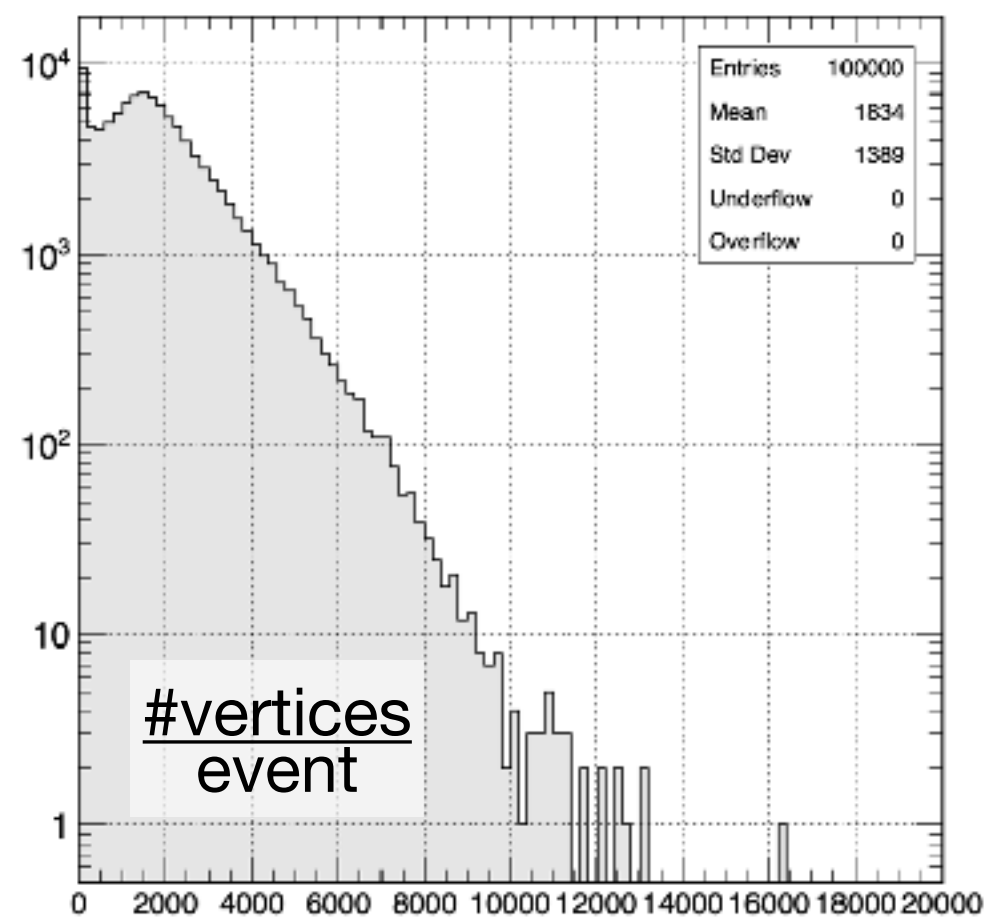
› Inheritance diagram for PHG4TruthInfoContainer:
› Collaboration diagram for PHG4TruthInfoContainer:

Public Types

typedef std::map< int, PHG4Particle * >	Map
typedef Map::iterator	Iterator
typedef Map::const_iterator	ConstIterator
typedef std::pair< Iterator, Iterator >	Range
typedef std::pair< ConstIterator, ConstIterator >	ConstRange
typedef std::map< int, PHG4VtxPoint * >	VtxMap
typedef VtxMap::iterator	VtxIterator
typedef VtxMap::const_iterator	ConstVtxIterator
typedef std::pair< VtxIterator, VtxIterator >	VtxRange
typedef std::pair< ConstVtxIterator, ConstVtxIterator >	ConstVtxRange
typedef std::map< int, PHG4Shower * >	ShowerMap
typedef ShowerMap::iterator	ShowerIterator
typedef ShowerMap::const_iterator	ConstShowerIterator
typedef std::pair< ShowerIterator, ShowerIterator >	ShowerRange
typedef std::pair< ConstShowerIterator, ConstShowerIterator >	ConstShowerRange

What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower



sPHENIX Analysis Software
Documentation for sPHENIX simulation software

Home page | Related Pages | Modules | Namespaces | **Classes** | Files | Examples

Class List | Class Index | Class Hierarchy | Class Members

GetSecondaryVtxRange
GetShower
GetShowerMap
GetShowerRange
GetVtx
GetVtxMap
GetVtxRange
Identify
IsPrimary
IsPrimaryVtx
IsEmbedded
IsEmbeddedVtx
MaxShowerIndex
MaxTrkIndex
MaxVtxIndex
MinShowerIndex
MinTrkIndex
MinVtxIndex
Reset
ShowerSize
Size
ParticleEmbedFlags
ParticleMap
ShowerMap
VertexEmbedFlags
VtxMap

PHG4TruthSubsystem
PHG4TruthTrackingAction
PHG4Utilsession
PHG4UserPrimaryParticleInformation
PHG4Utils

PHG4TruthInfoContainer Class Reference

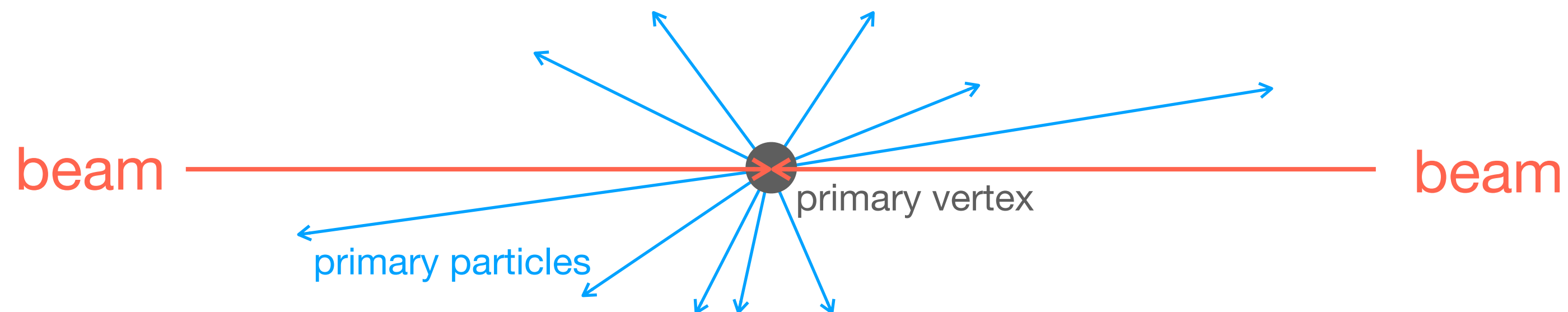
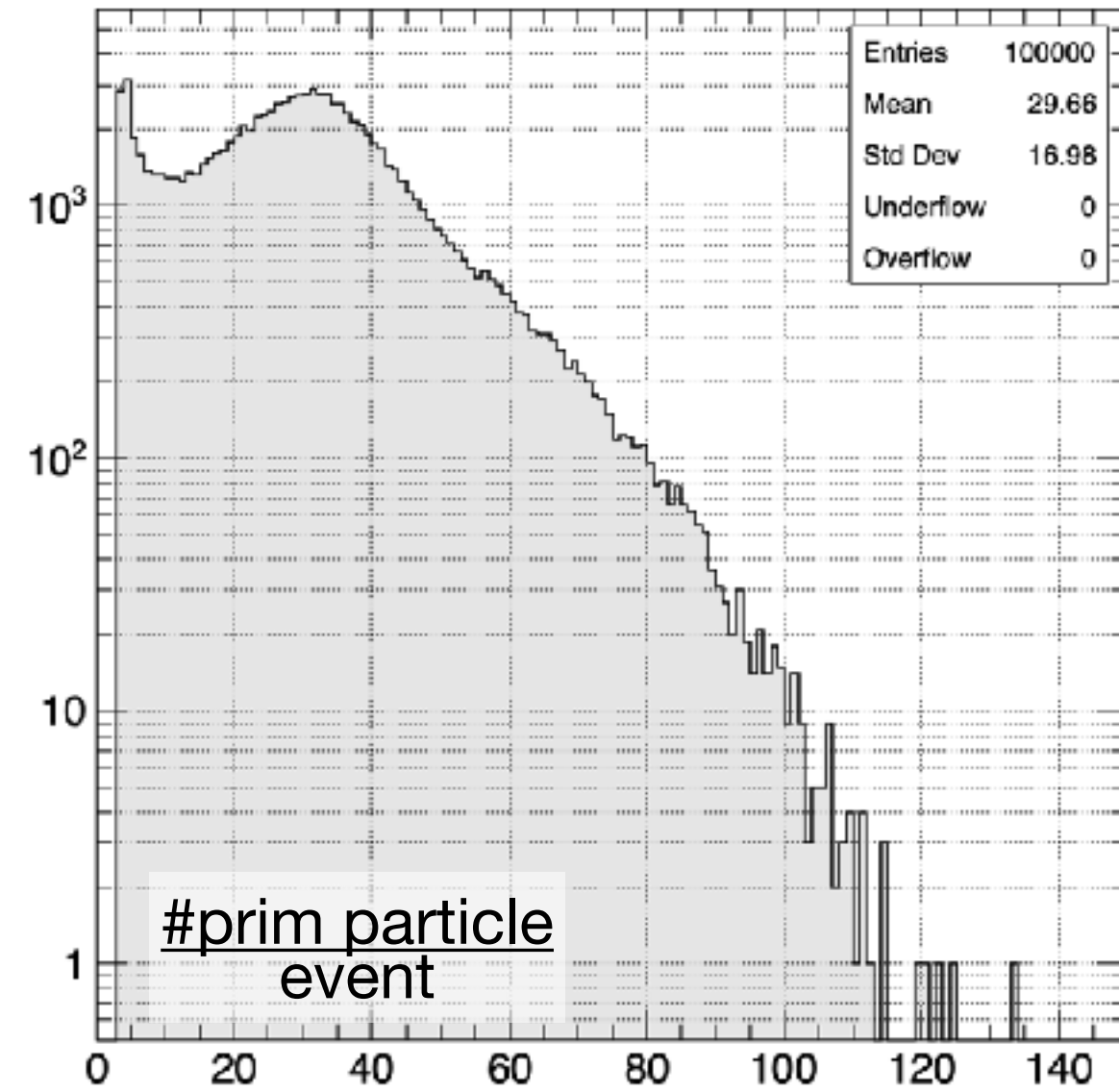
```
#include <coresoftware/blob/master/simulation/g4simulation/g4...>
Inheritance diagram for PHG4TruthInfoContainer:
Collaboration diagram for PHG4TruthInfoContainer:
```

Public Types

```
typedef std::map< int, PHG4Particle * > Map
typedef Map::iterator Iterator
typedef Map::const_iterator ConstIterator
typedef std::pair< Iterator, Iterator > Range
typedef std::pair< ConstIterator, ConstIterator > ConstRange
typedef std::map< int, PHG4VtxPoint * > VtxMap
typedef VtxMap::iterator VtxIterator
typedef VtxMap::const_iterator ConstVtxIterator
typedef std::pair< VtxIterator, VtxIterator > VtxRange
typedef std::pair< ConstVtxIterator, ConstVtxIterator > ConstVtxRange
typedef std::map< int, PHG4Shower * > ShowerMap
typedef ShowerMap::iterator ShowerIterator
typedef ShowerMap::const_iterator ConstShowerIterator
typedef std::pair< ShowerIterator, ShowerIterator > ShowerRange
typedef std::pair< ConstShowerIterator, ConstShowerIterator > ConstShowerRange
```

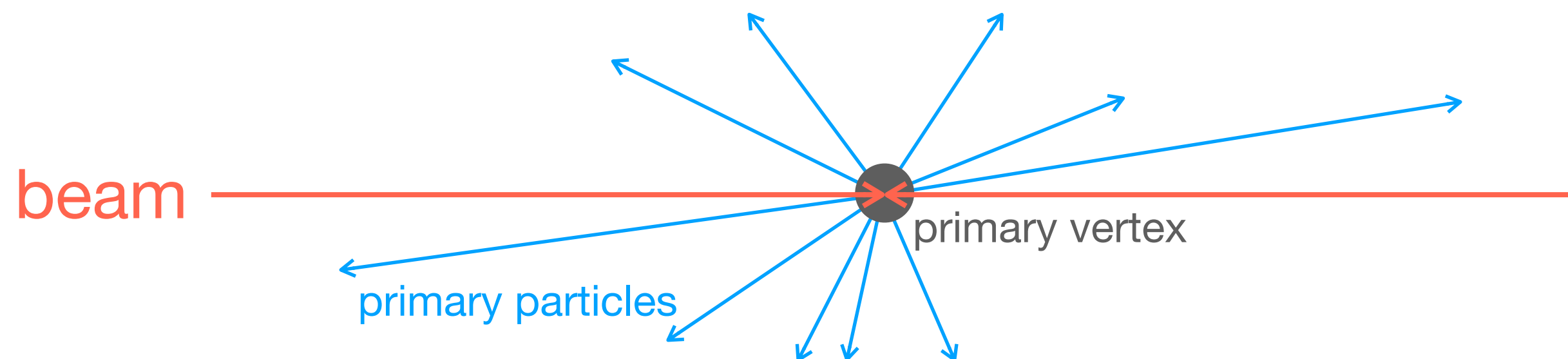
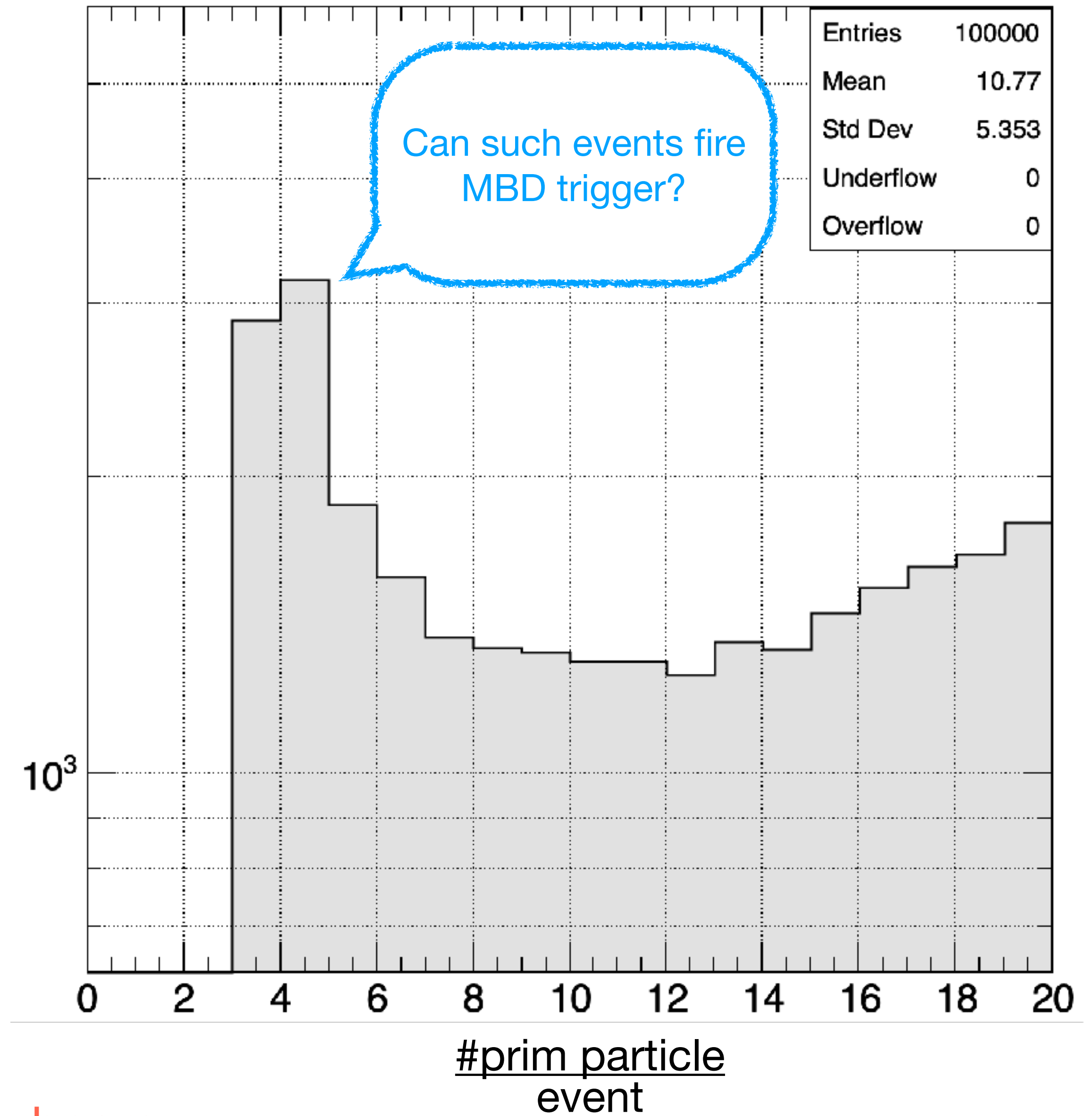
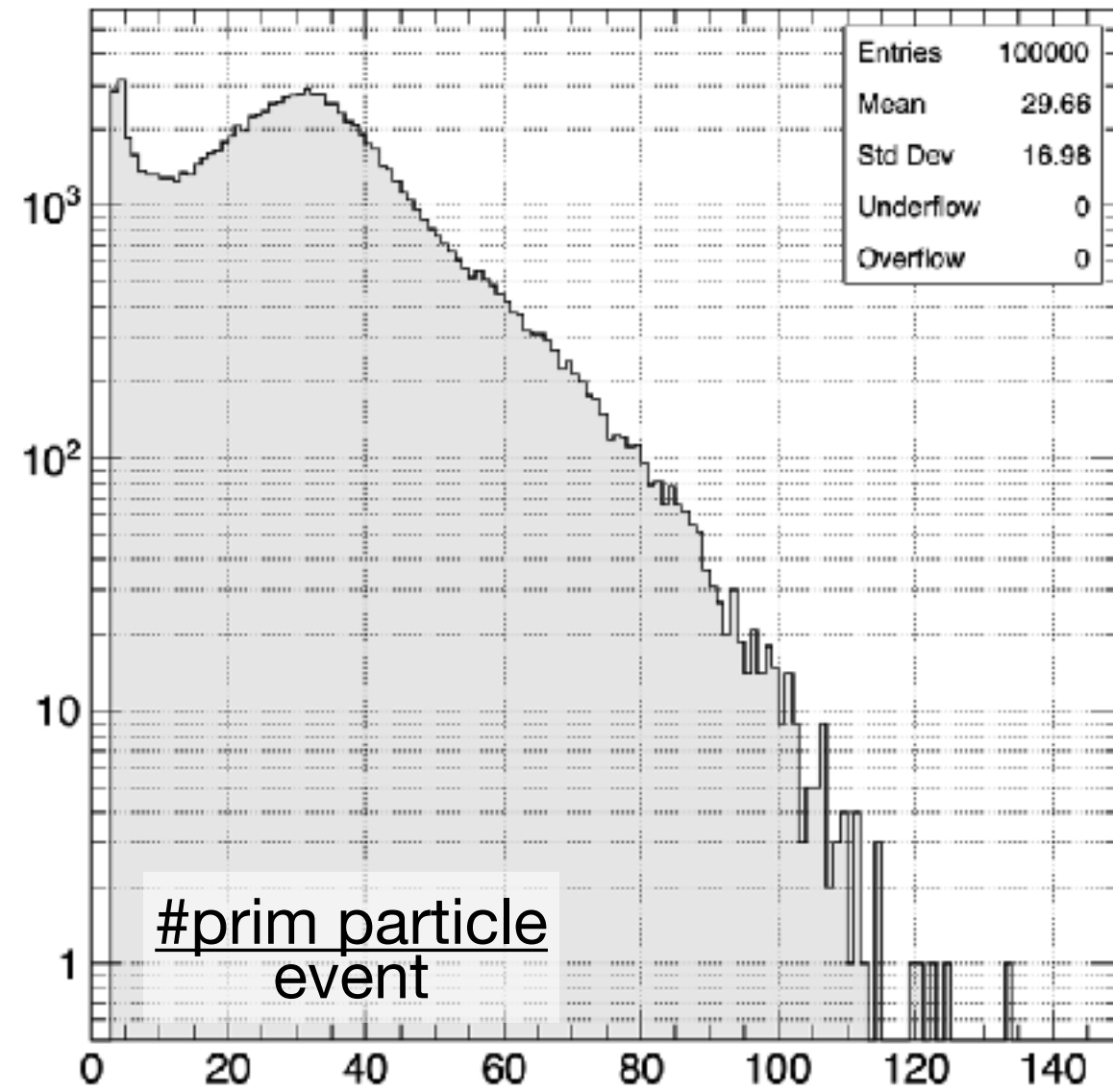
What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower



What can be done more?

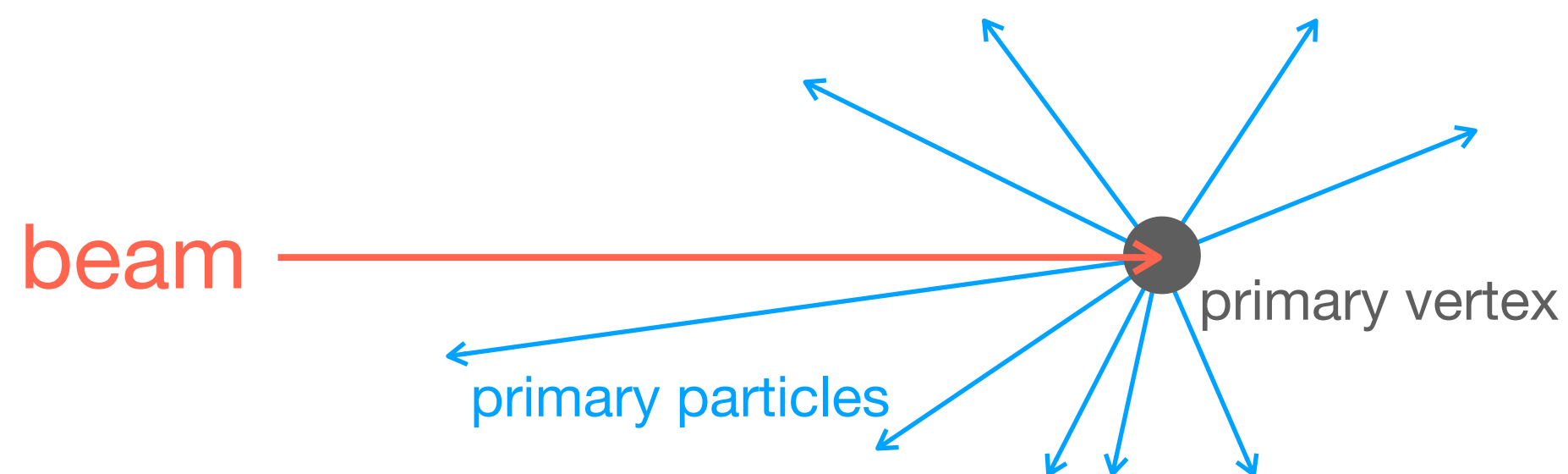
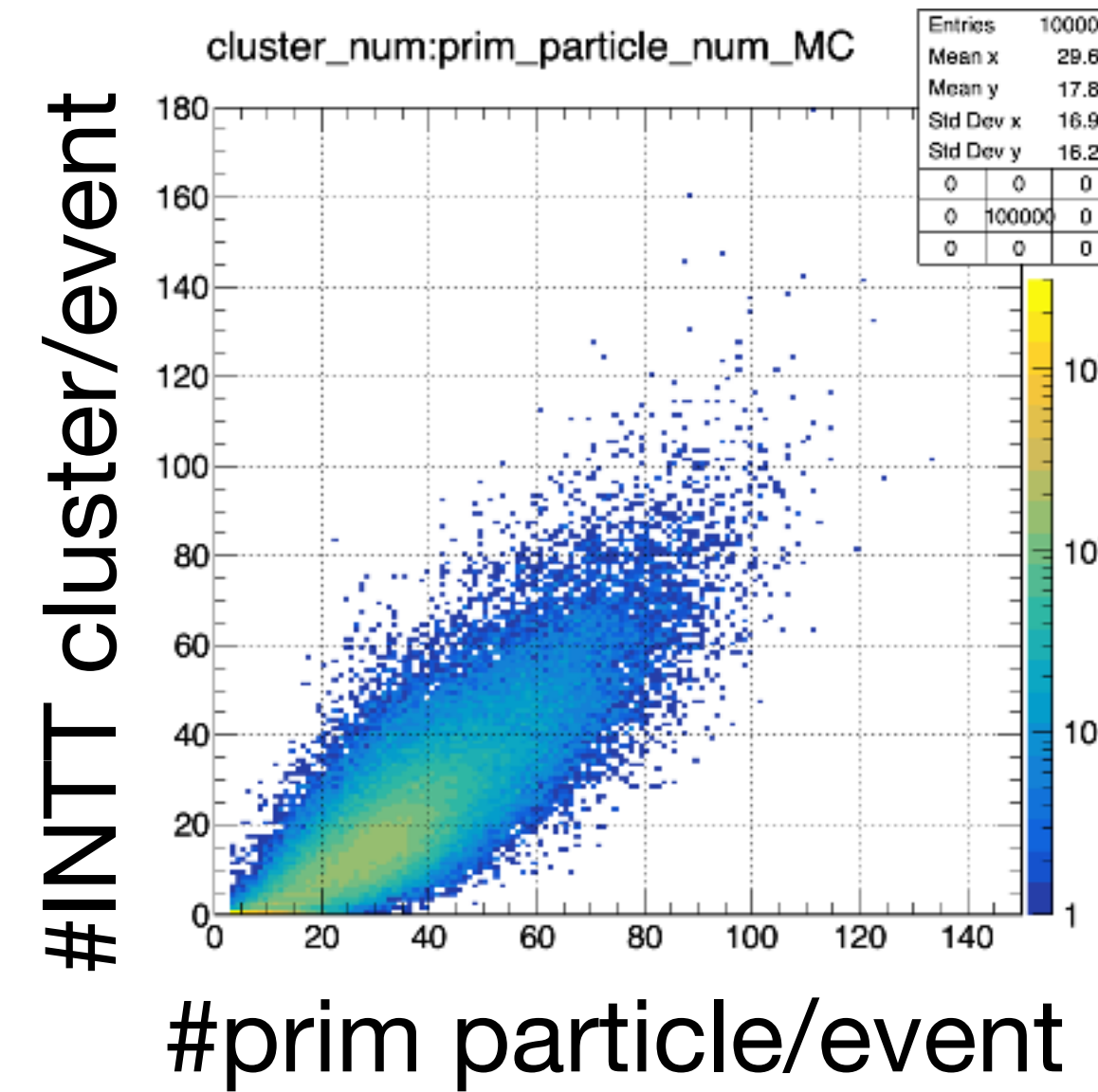
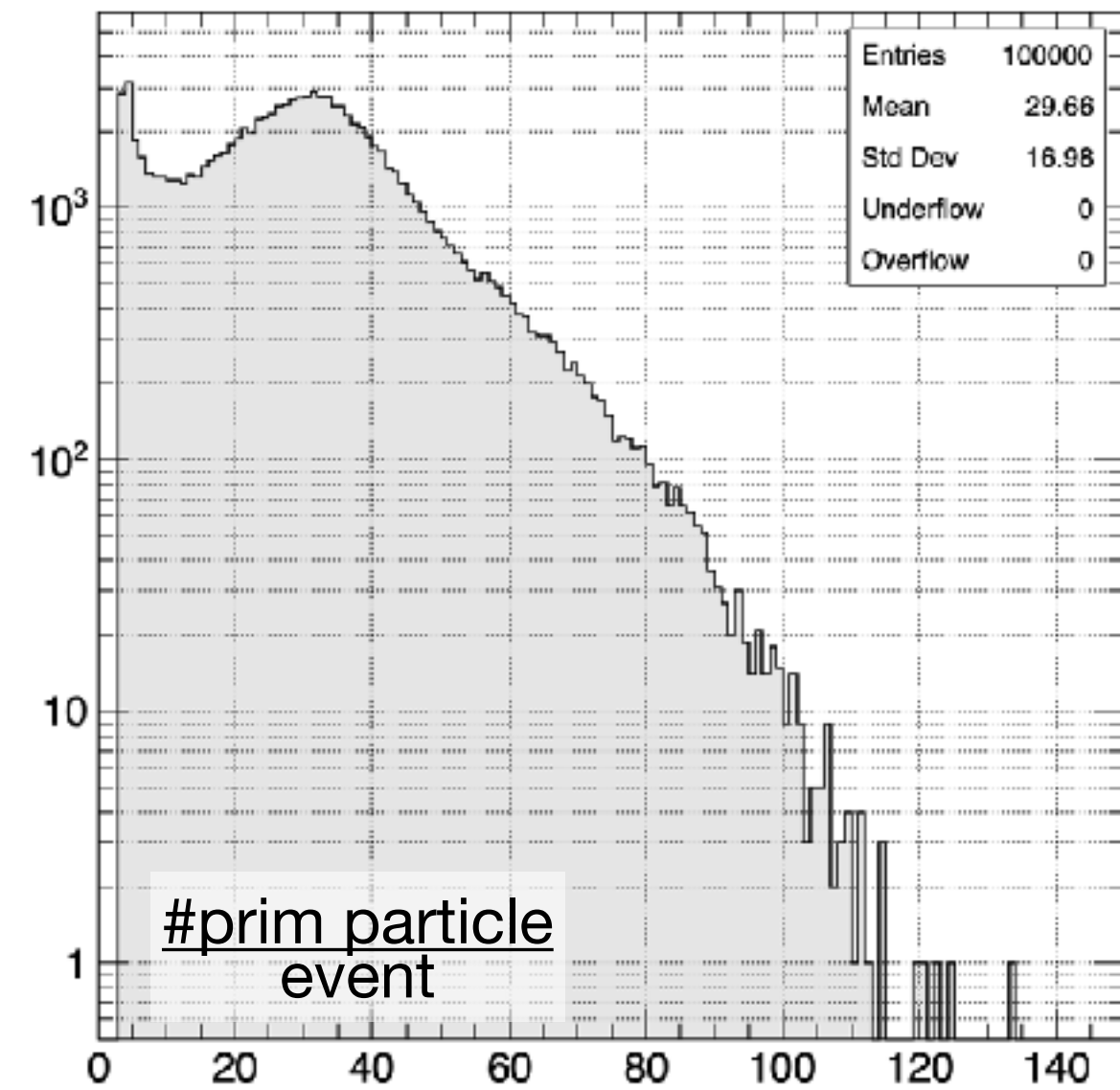
- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower



beam

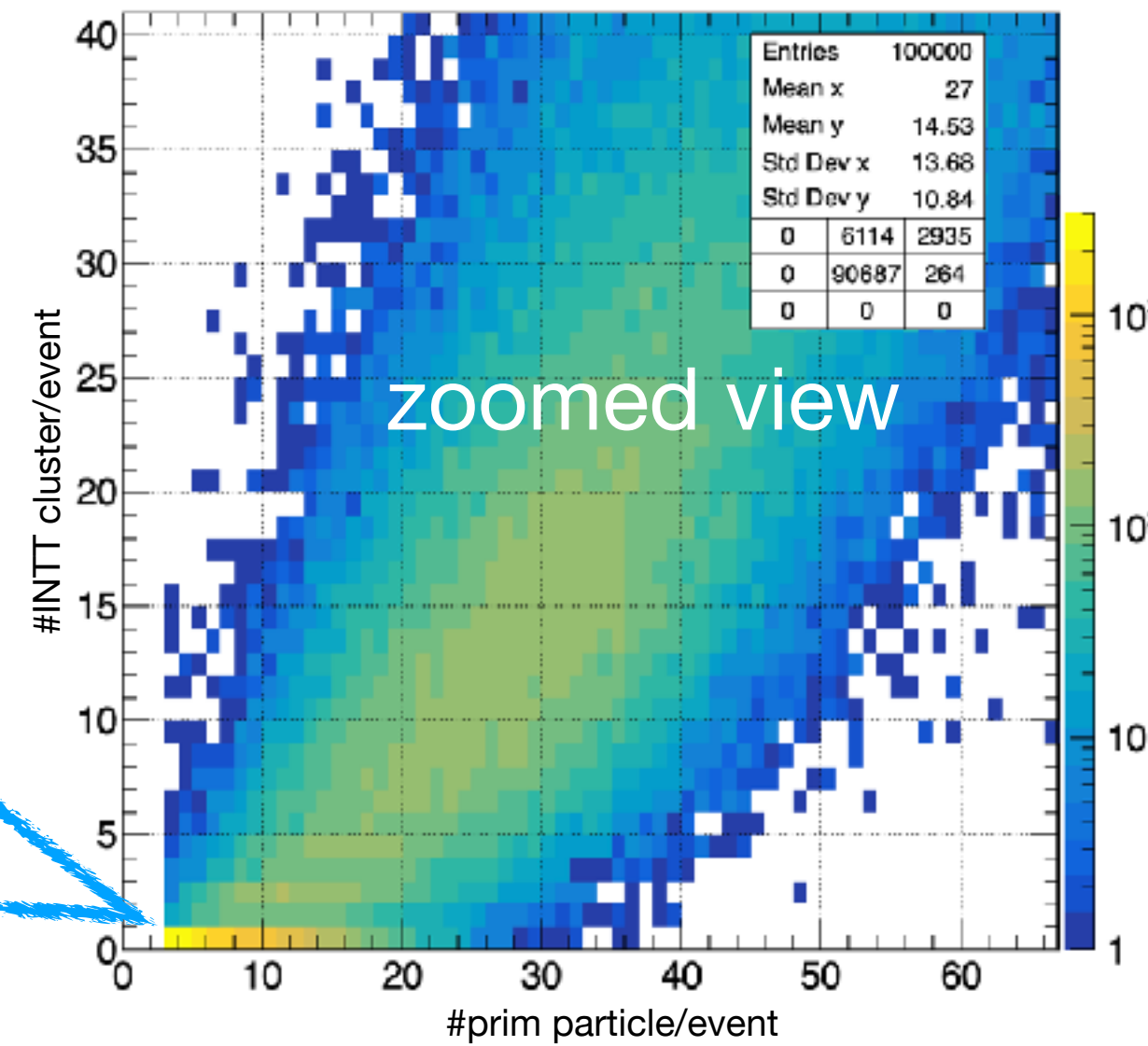
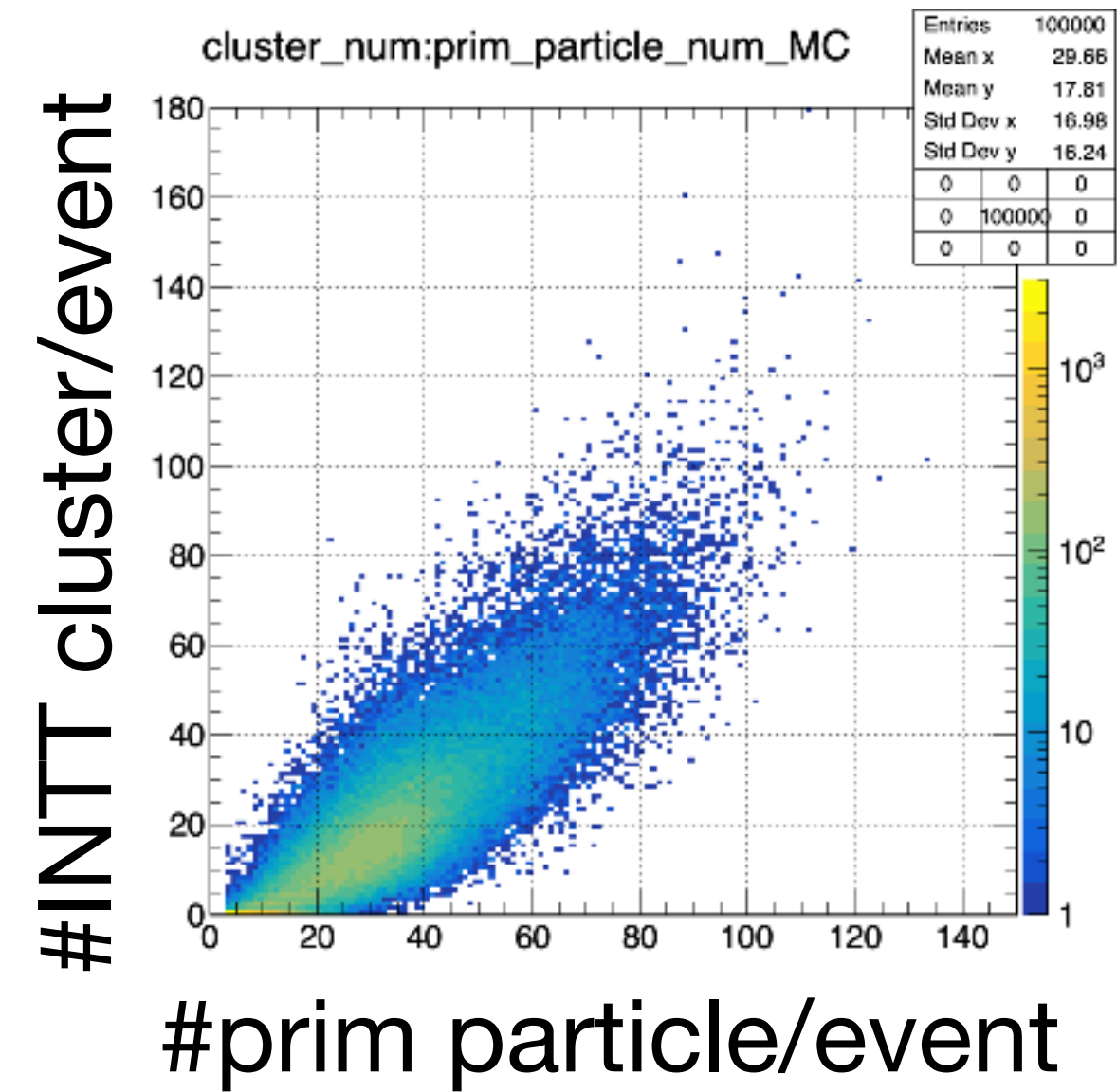
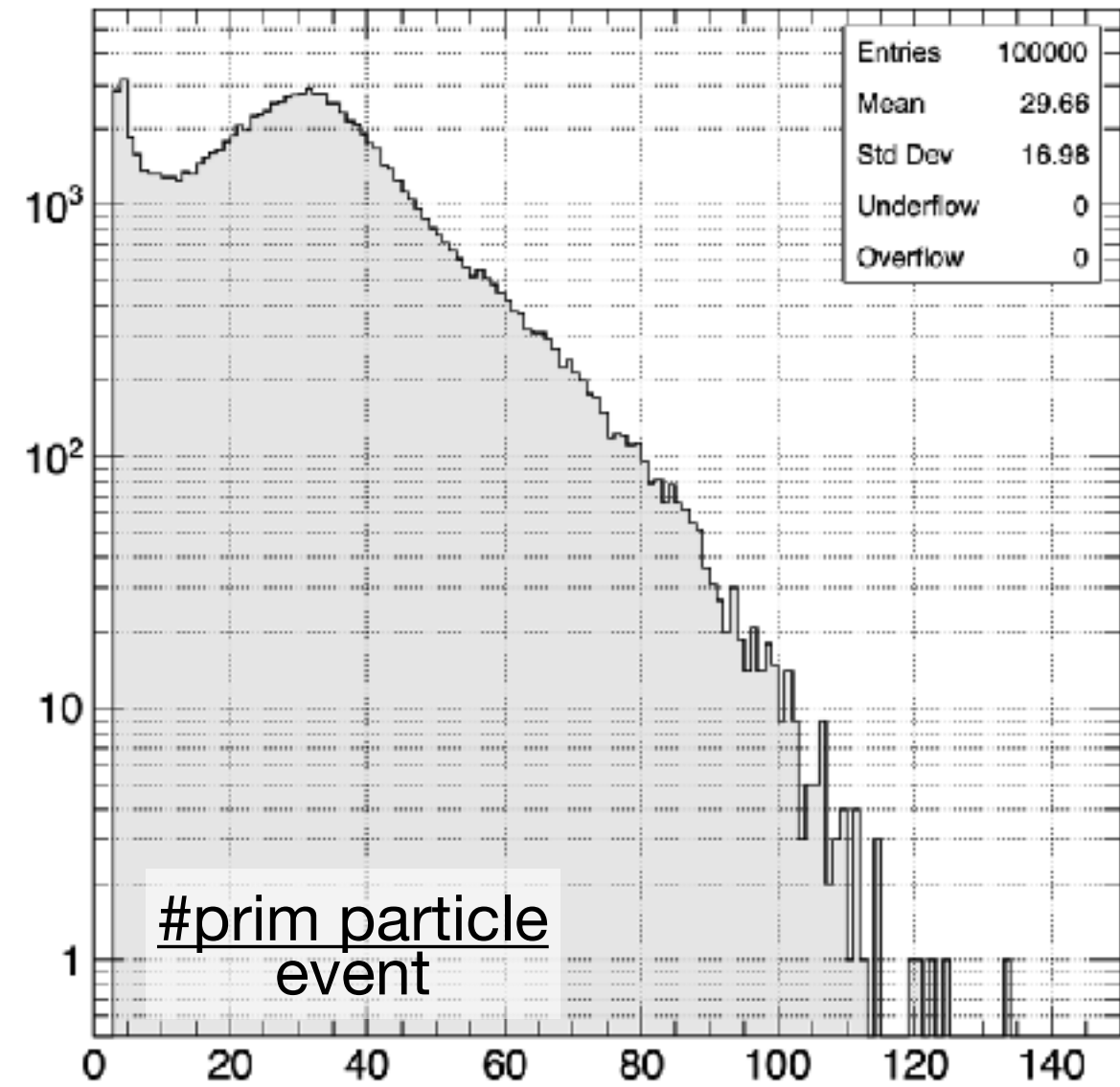
What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower

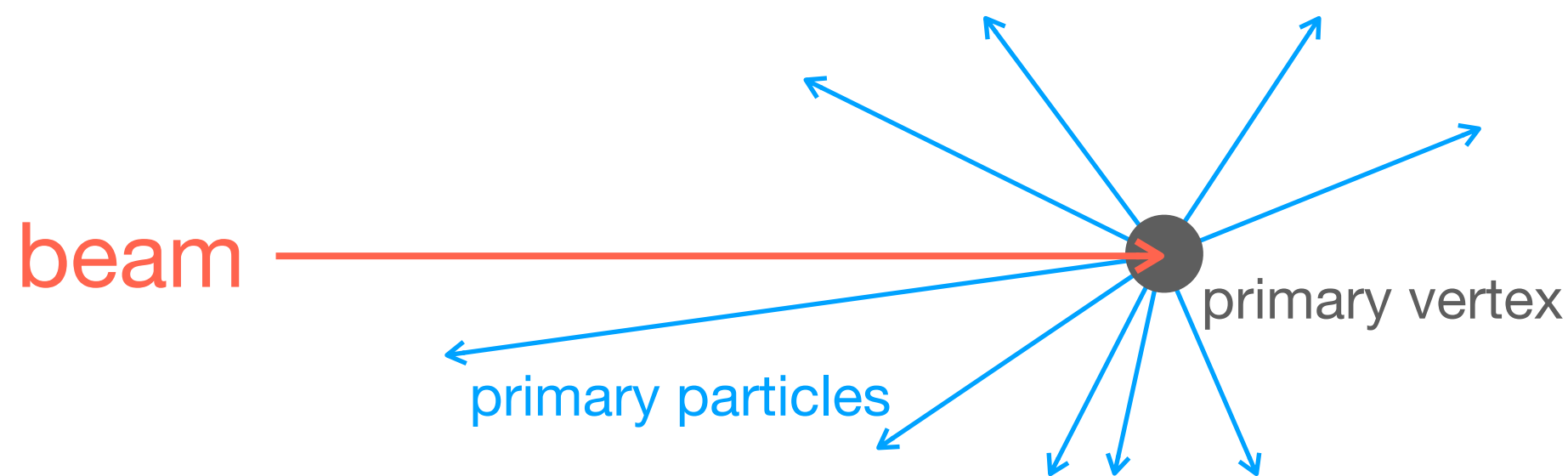


What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower

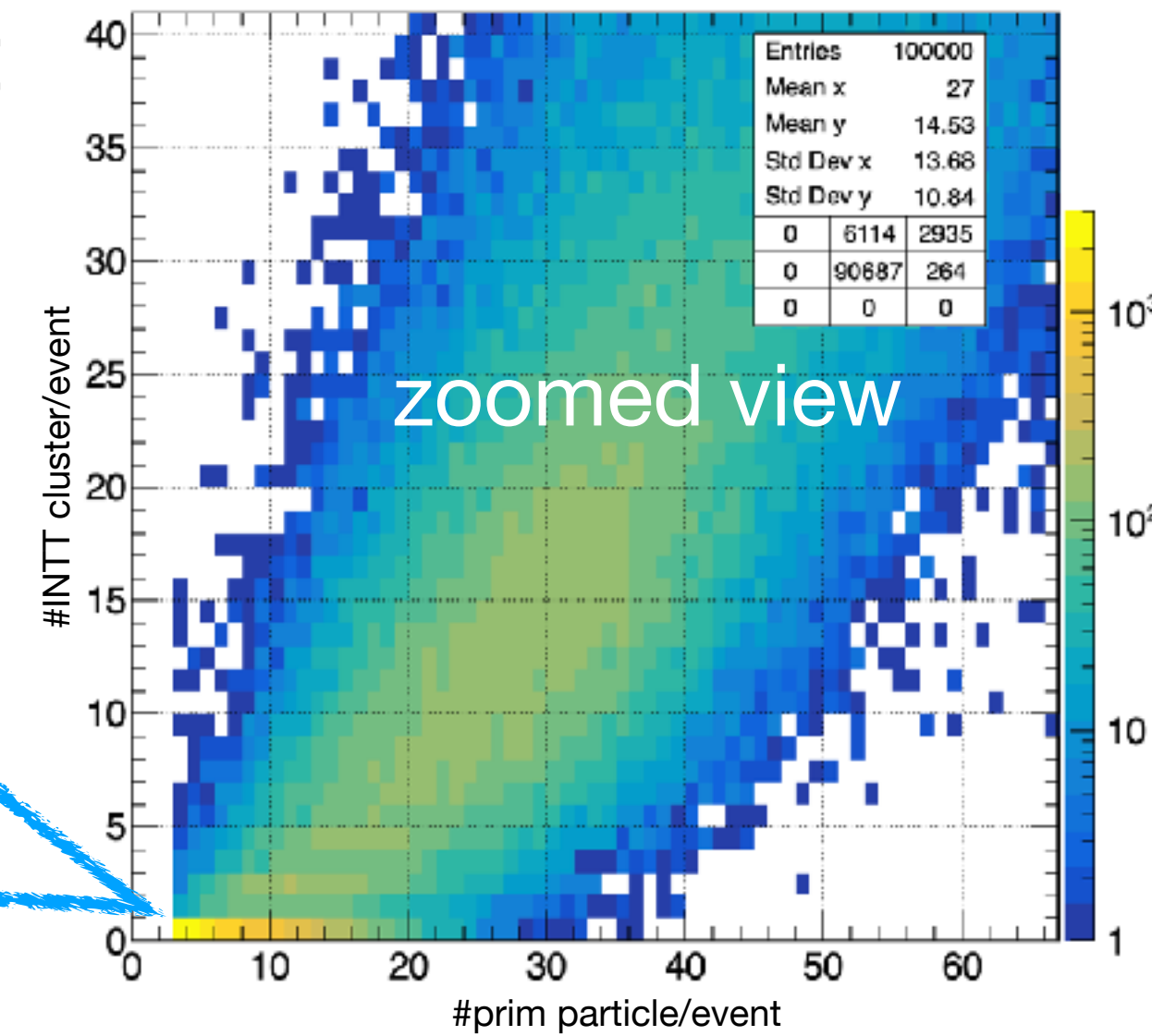
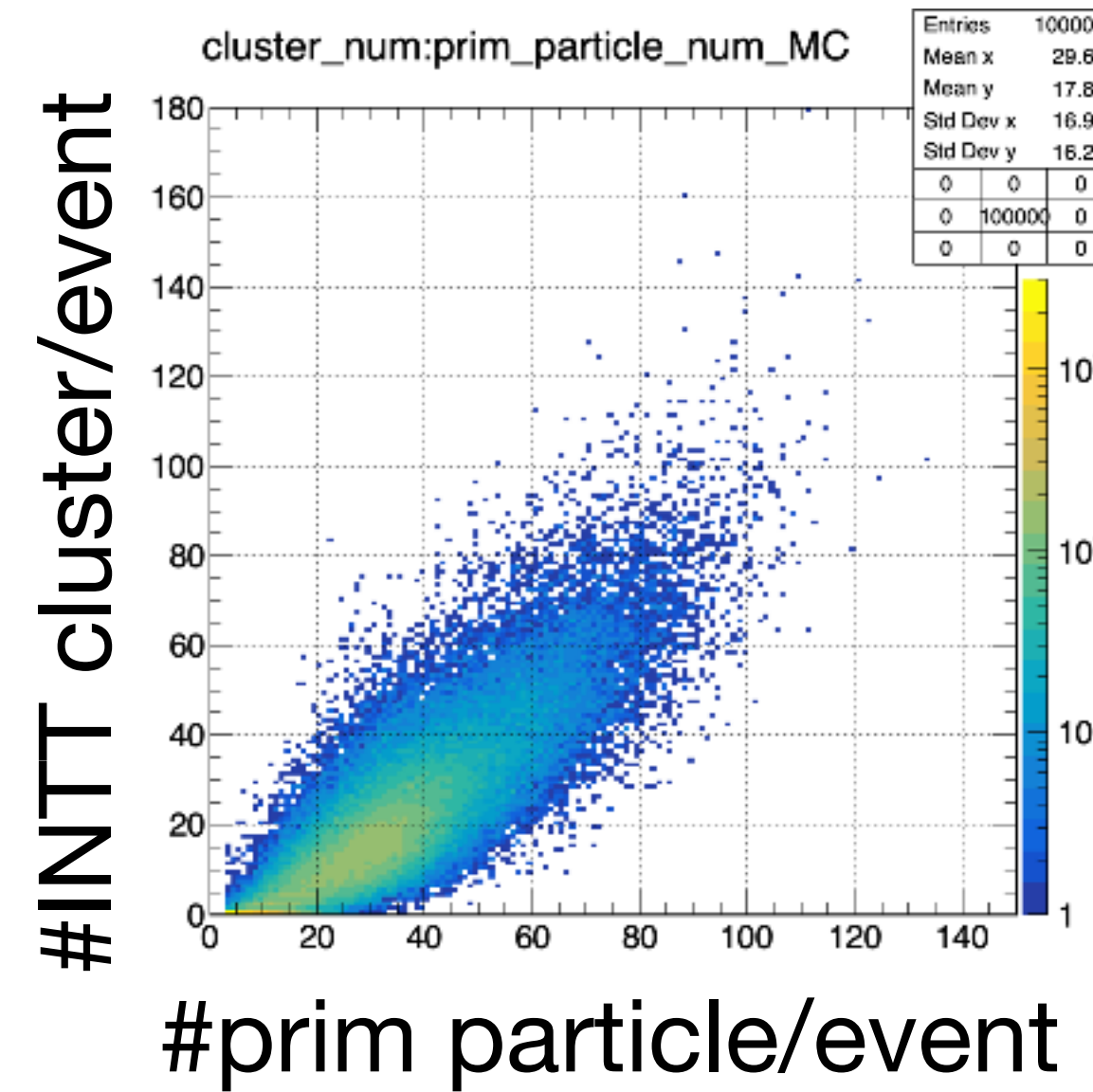
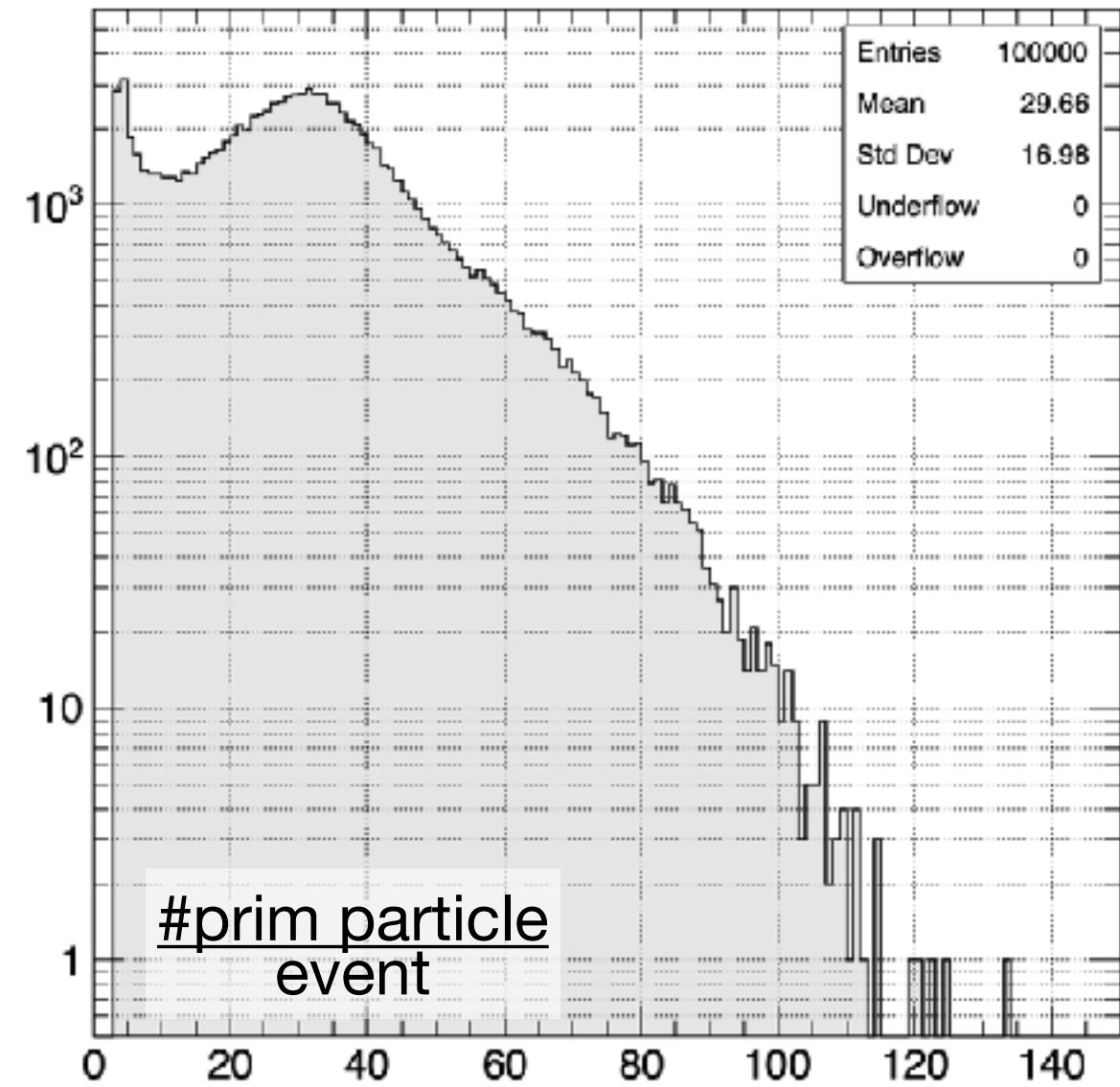


Events with #prim particle < 40 sometimes have no INTT cluster

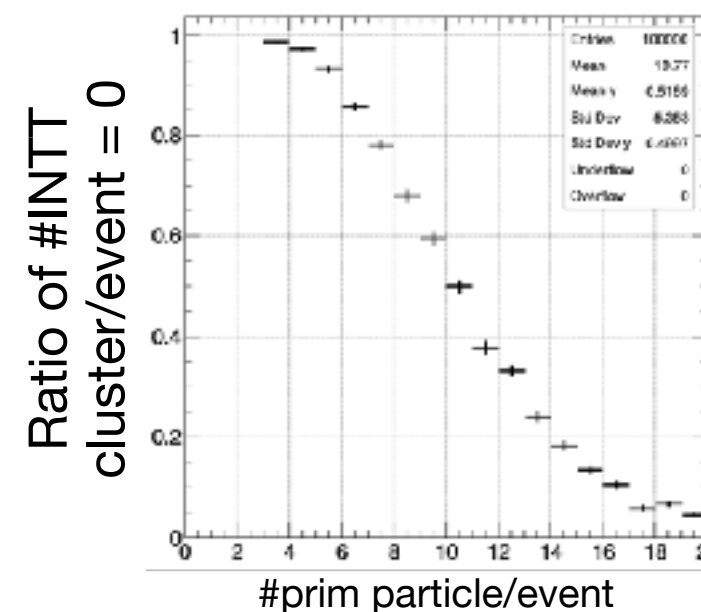
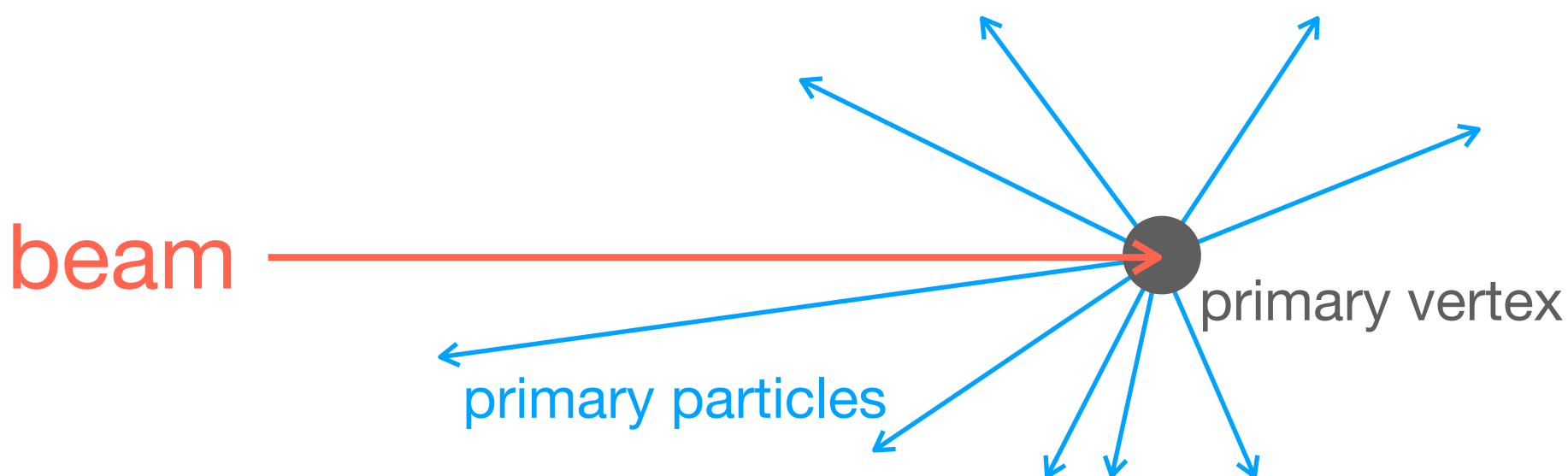


What can be done more?

- [PHG4TruthInfoContainer](#) allows us to access
 - Primary/secondary vertices
 - Primary/secondary particles
 - Shower



Events with #prim particle < 40 sometimes have no INTT cluster

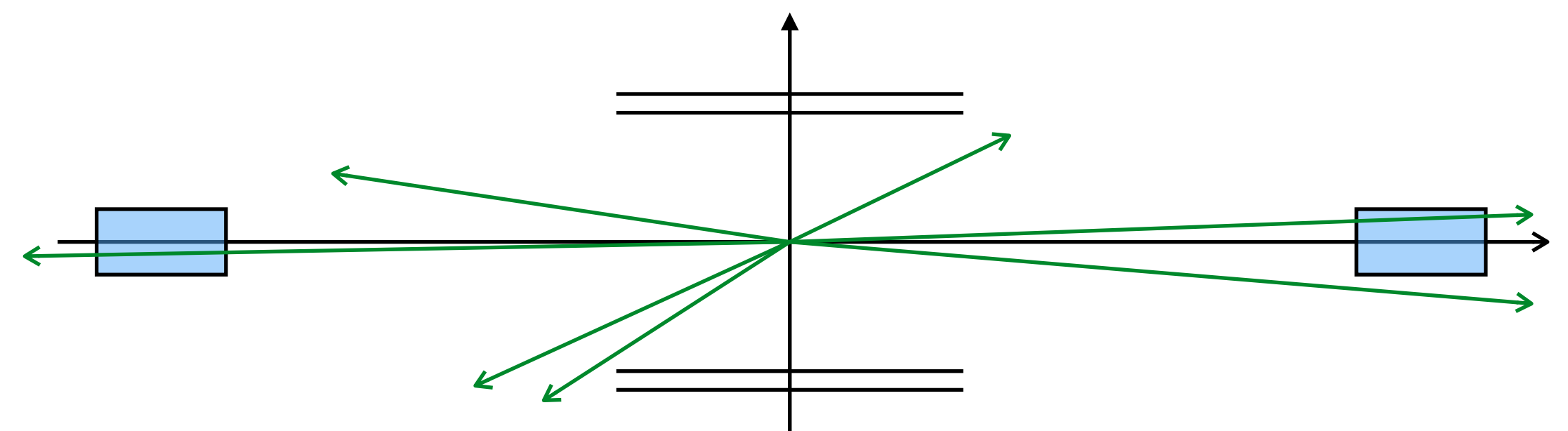
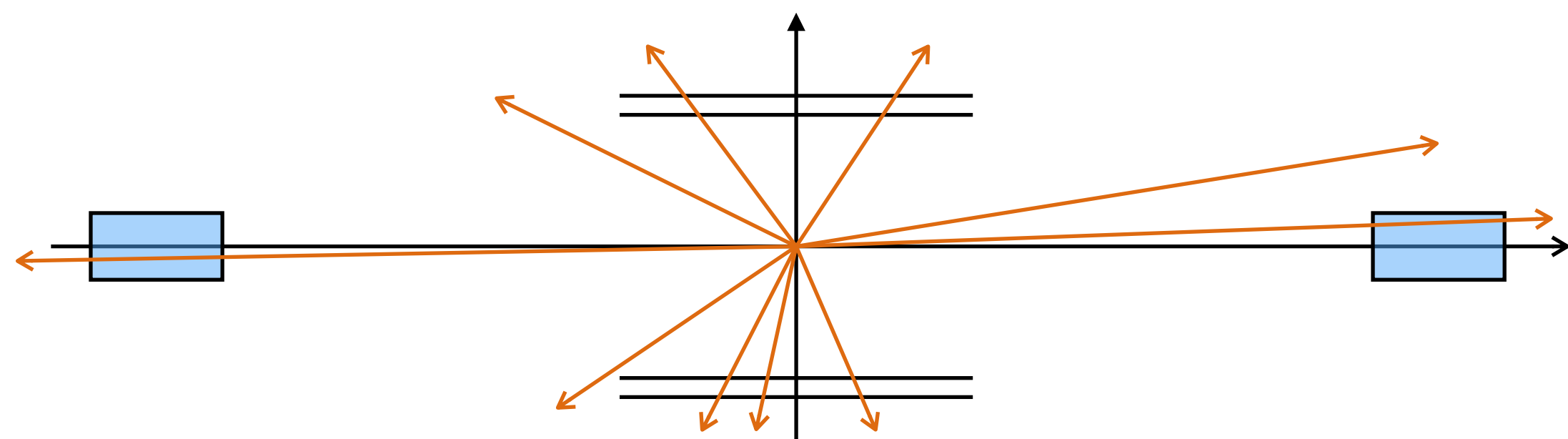
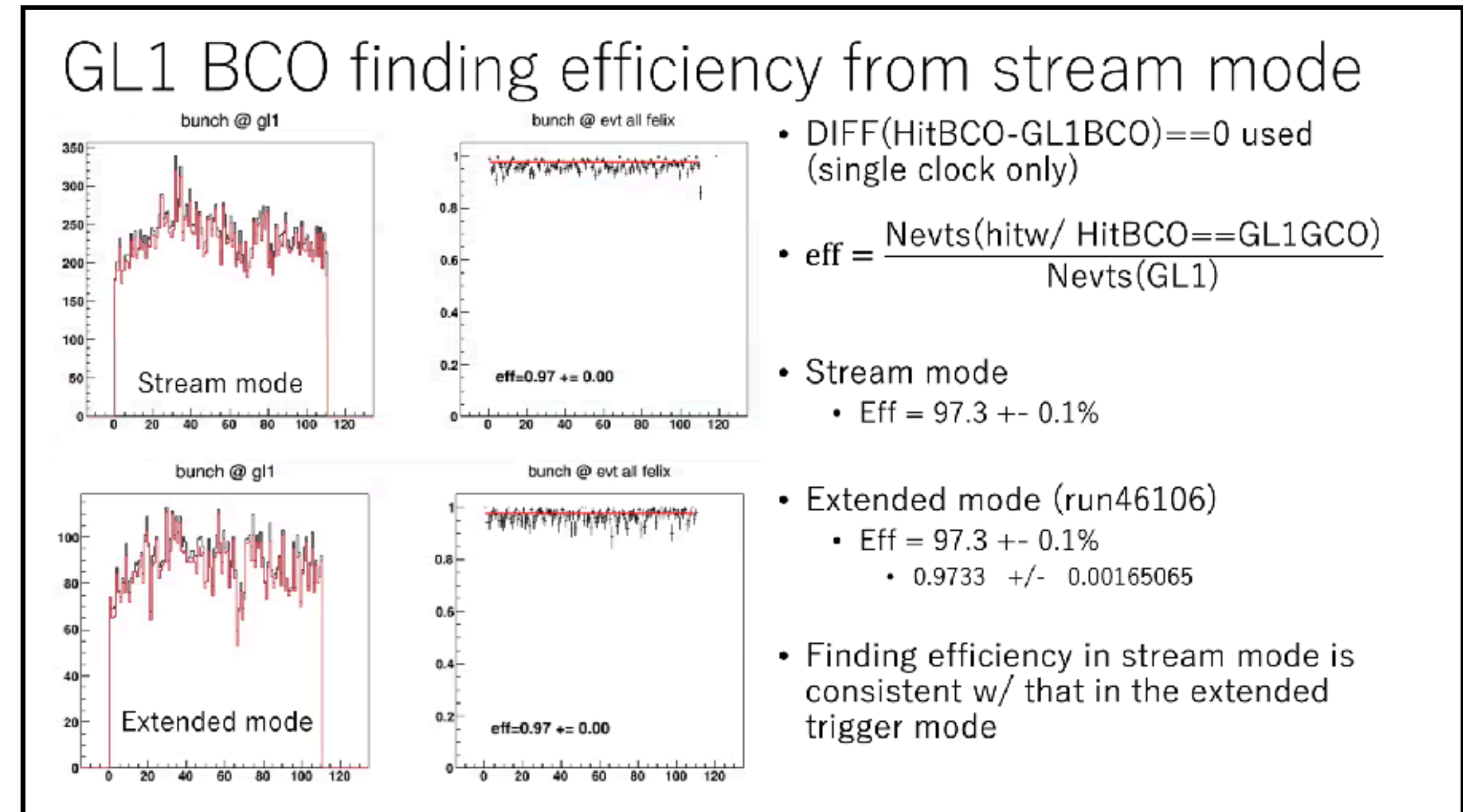


Events with few primary particles are the main contribution of #INTT cluster=0/event.
Can such events fire MBD trigger? ← Next step...

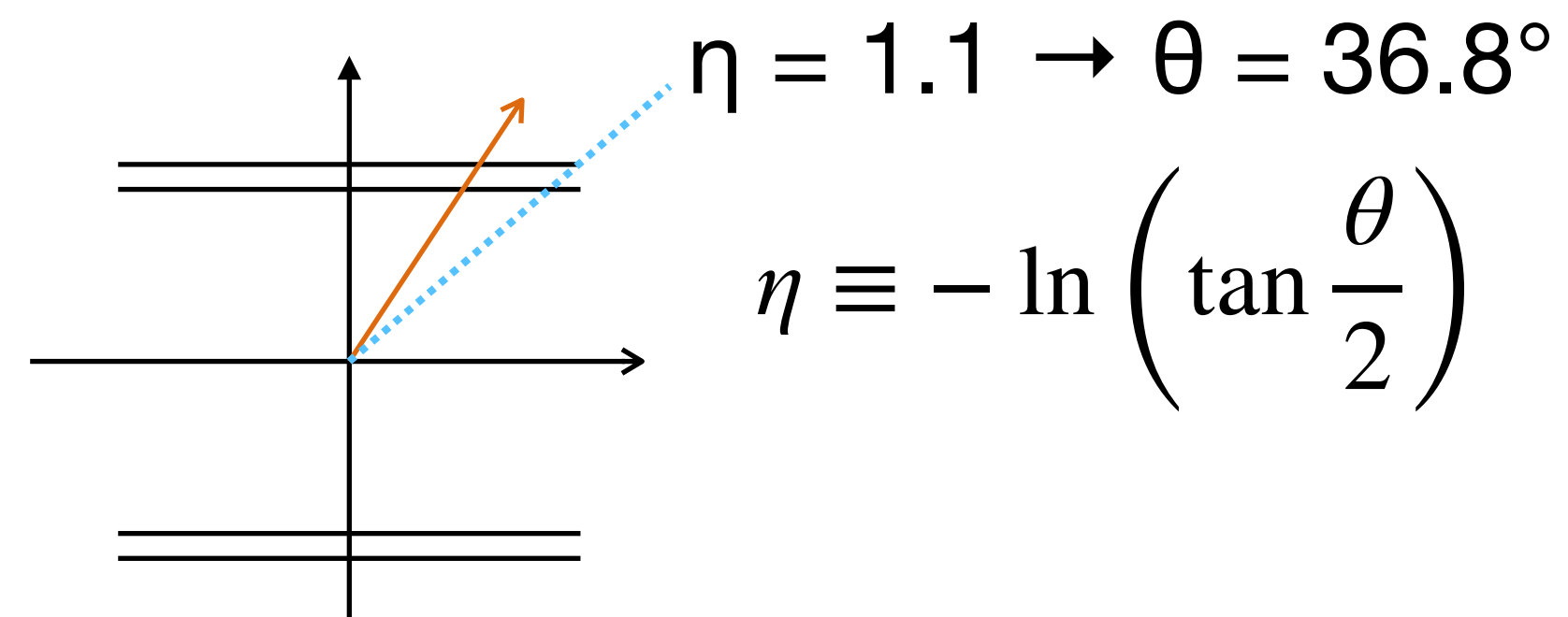
backup

Why does INTT take 97% of minimum bias trigger events?

- Takashi showed GL1 BCO finding efficiency for the trigger+extended readout data and the streaming readout data. Both were 97.3%.
- It's good to check MC data whether (a part of) the inefficiency comes from physics.



Easy estimation

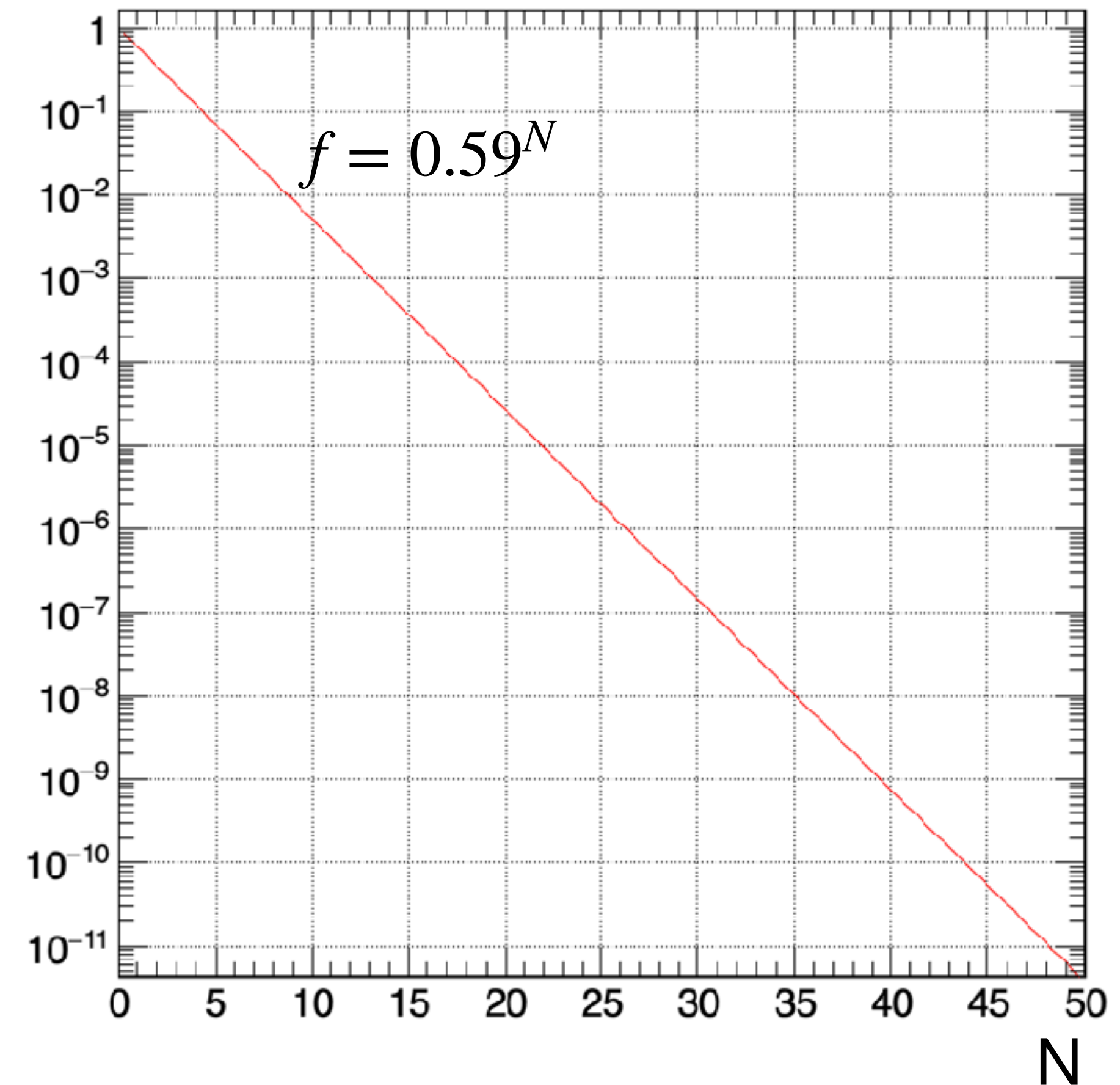


Probability of a particle outside INTT acceptance

$$1 - 36.8/90 = 0.59$$

Probability of uncorrelated N particles outside INTT acceptance

$$0.59^N$$



The ratio of events without INTT cluster with various selection of #primary particles

- All
- #prim part. > 5
- #prim part. > 10
- #prim part. > 15
- #prim part. > 20

