#INTT cluster distribution using MDC2 minimum bias MC data –Part 4–

Genki Nukazuka (RIKEN)

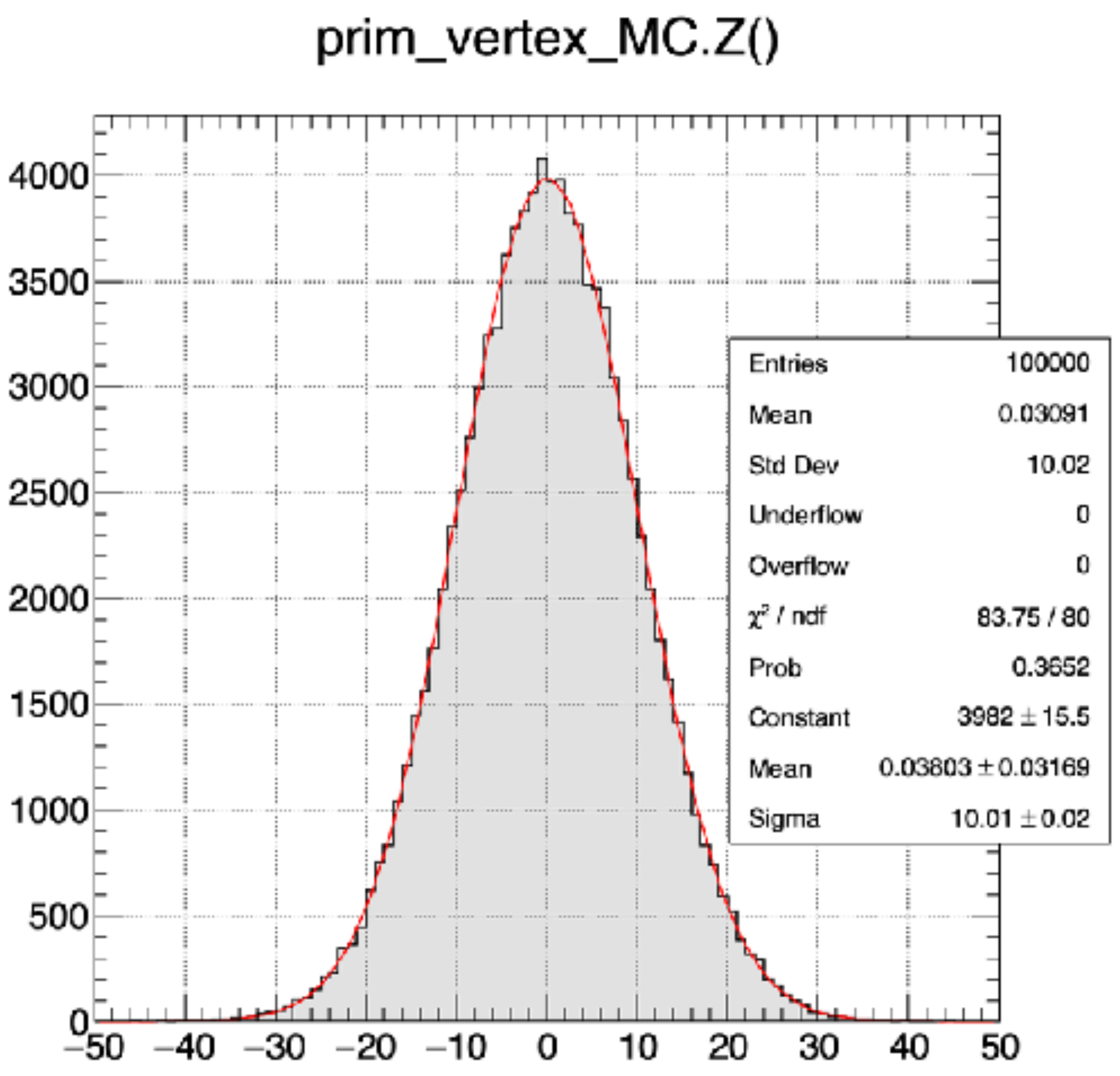
- Part 1
- Part 2
- Part 3

Status

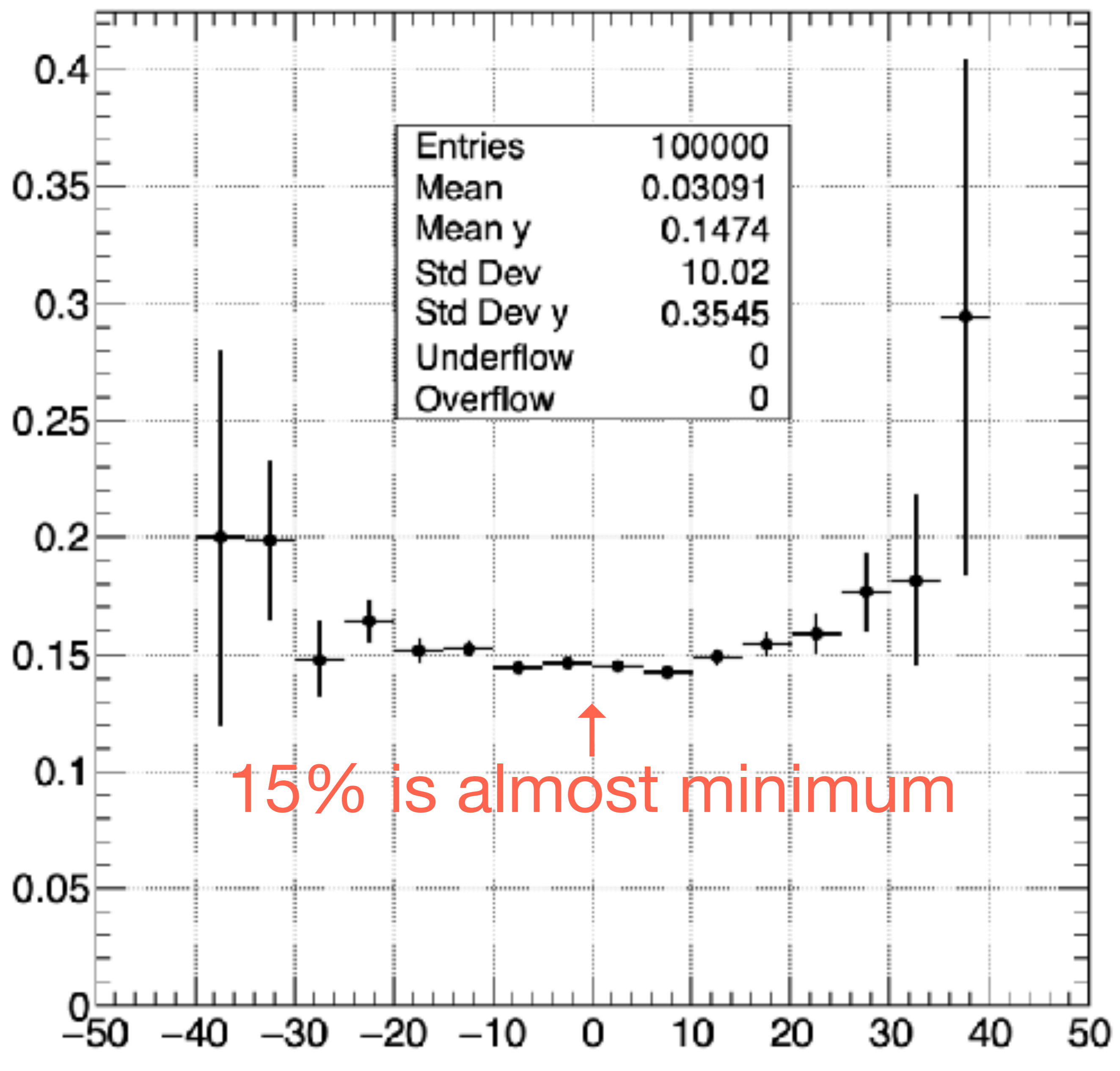
Truth z_{vtx} distribution was analyzed.

The selection of zvtx didn't greatly increase the ratio of #INTT cluster/event.

MBD trigger needs to be simulated.

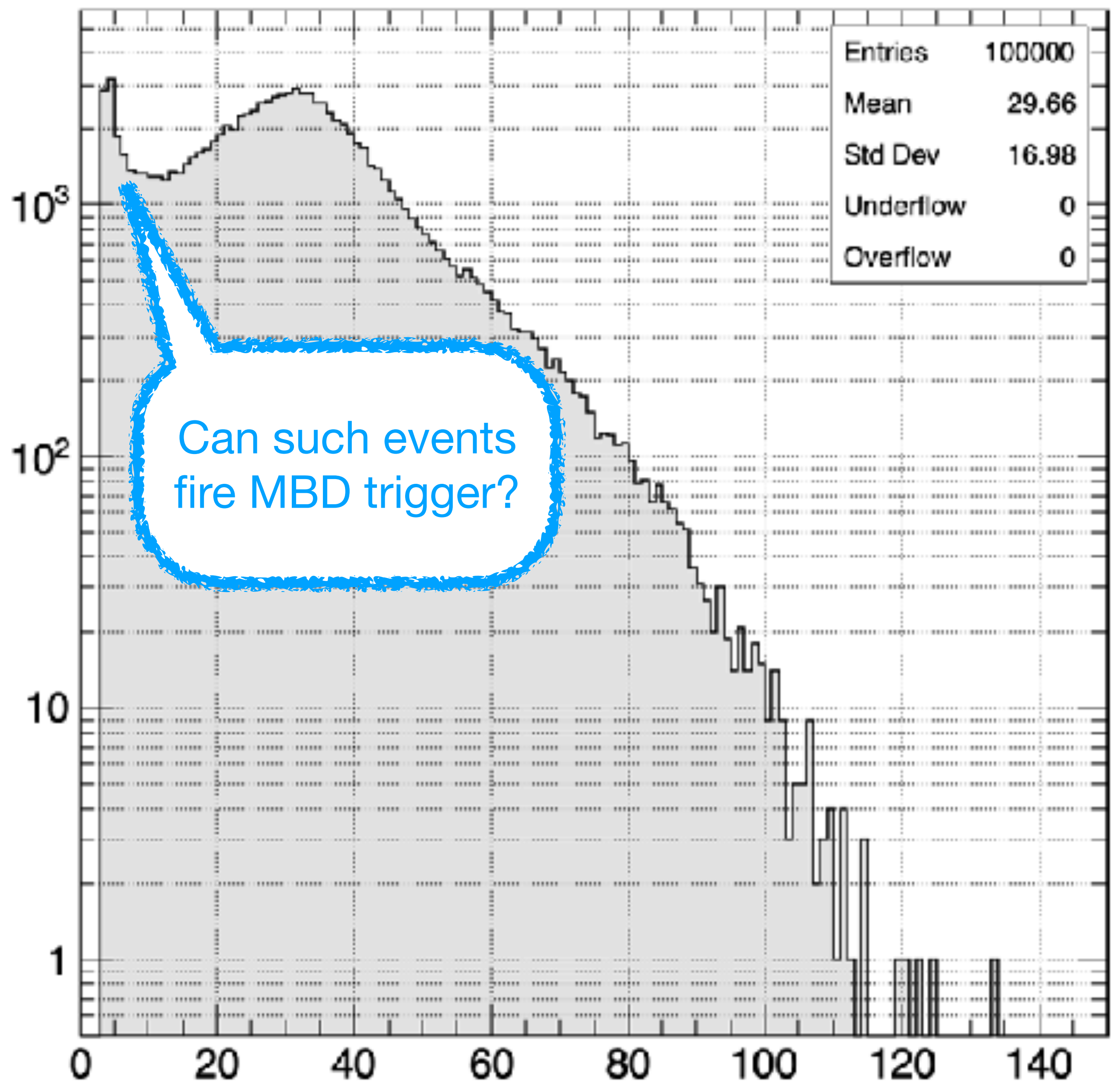


MC truth z_{vtx} (cm)



15% is almost minimum

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

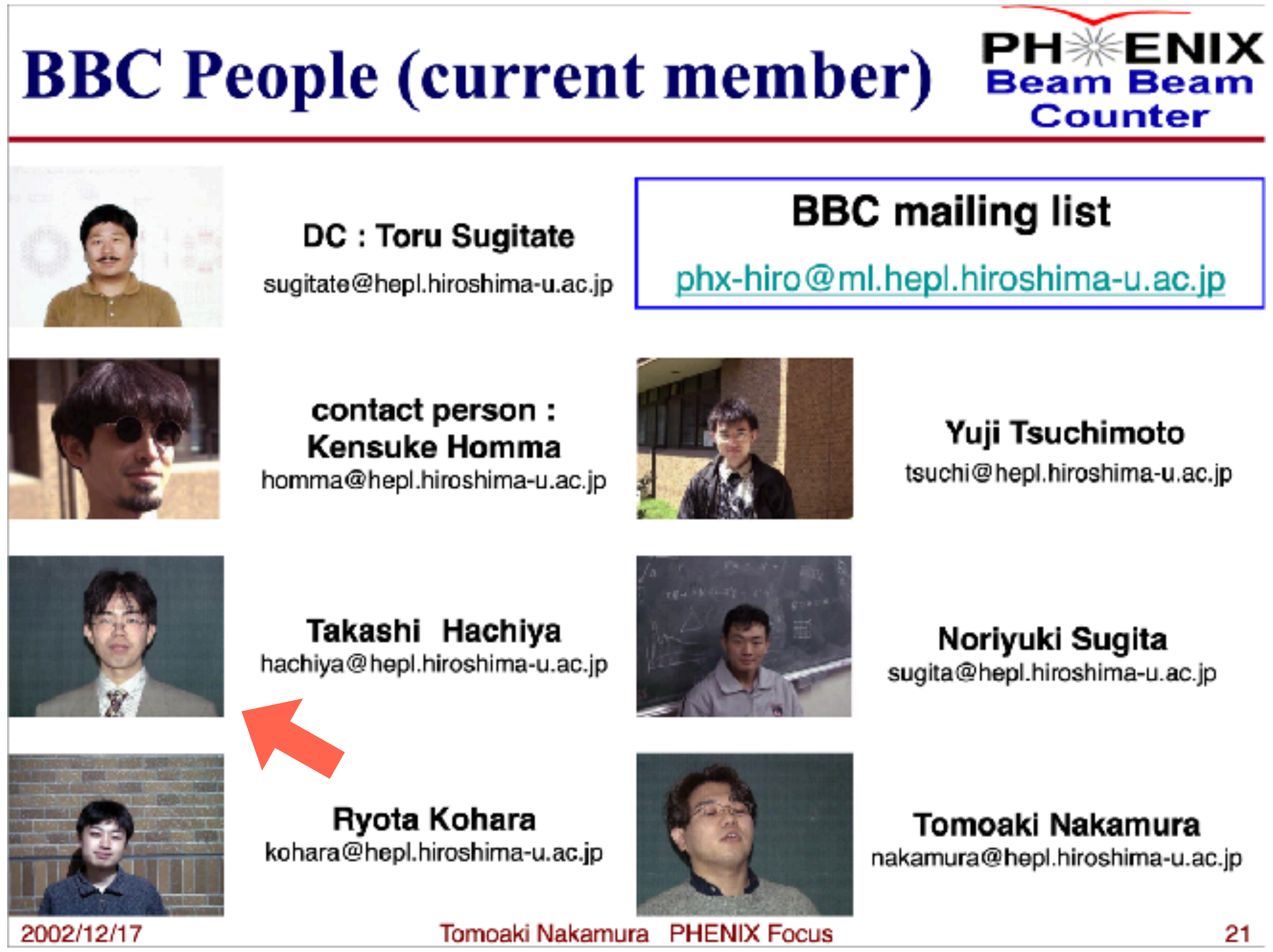


Can such events fire MBD trigger?

#primary particle / event distribution

Mimimum**B**ias**D**etector

- 2 sets of Cherenkov counter locating forward region ($3.51<|\eta|<4.61$)
  - It consists of a quartz radiator and PMTs.
  - It was used as **B**eam**B**eam**C**ounter in the PHENIX era.
  - Timing resolution: 120 ps. The difference in detection timing between the north and south gives z coordinate of collisions.
  - It provides minimum bias trigger
- Information: PHENIX web, NIM paper, MBD review



BBC People (current member)

PHENIX Beam Beam Counter

DC : Toru Sugitate
sugitate@hepl.hiroshima-u.ac.jp

BBC mailing list
phx-hiro@ml.hepl.hiroshima-u.ac.jp

contact person :
Kensuke Homma
homma@hepl.hiroshima-u.ac.jp

Yuji Tsuchimoto
tsuchi@hepl.hiroshima-u.ac.jp

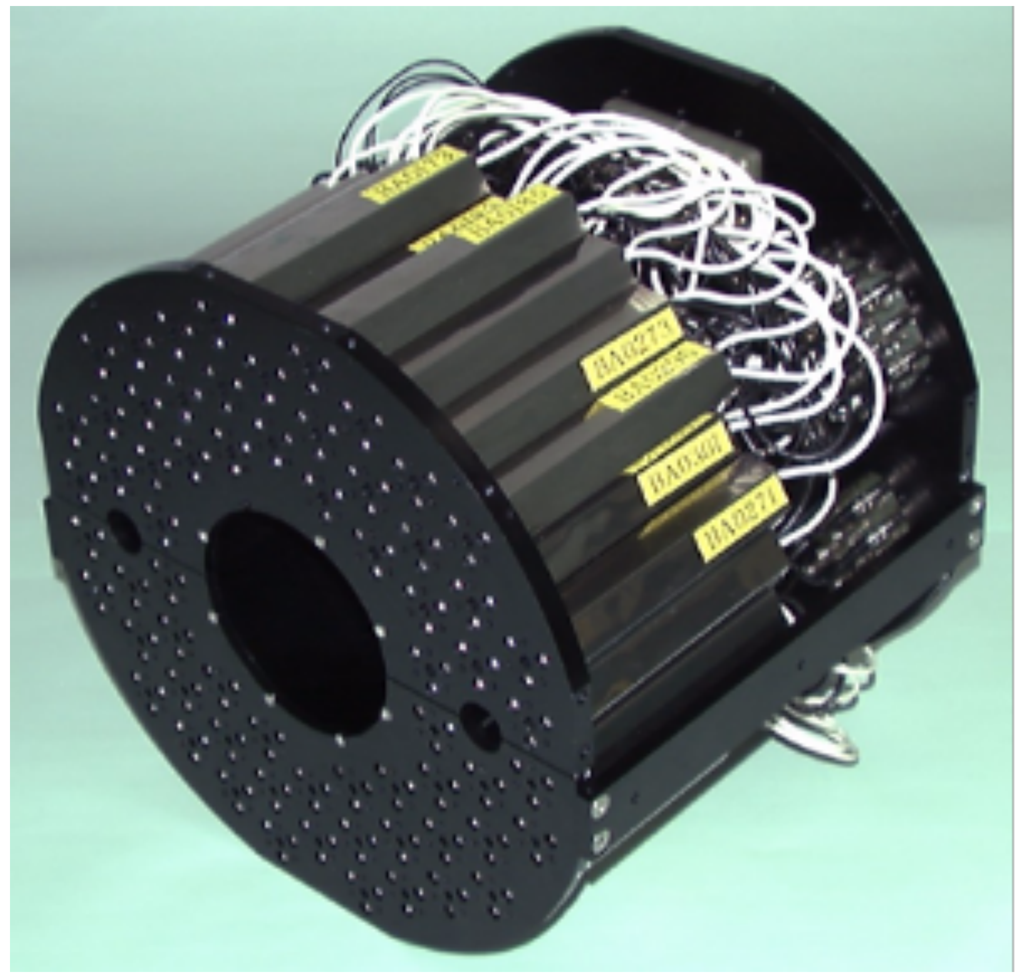
Takashi Hachiya
hachiya@hepl.hiroshima-u.ac.jp

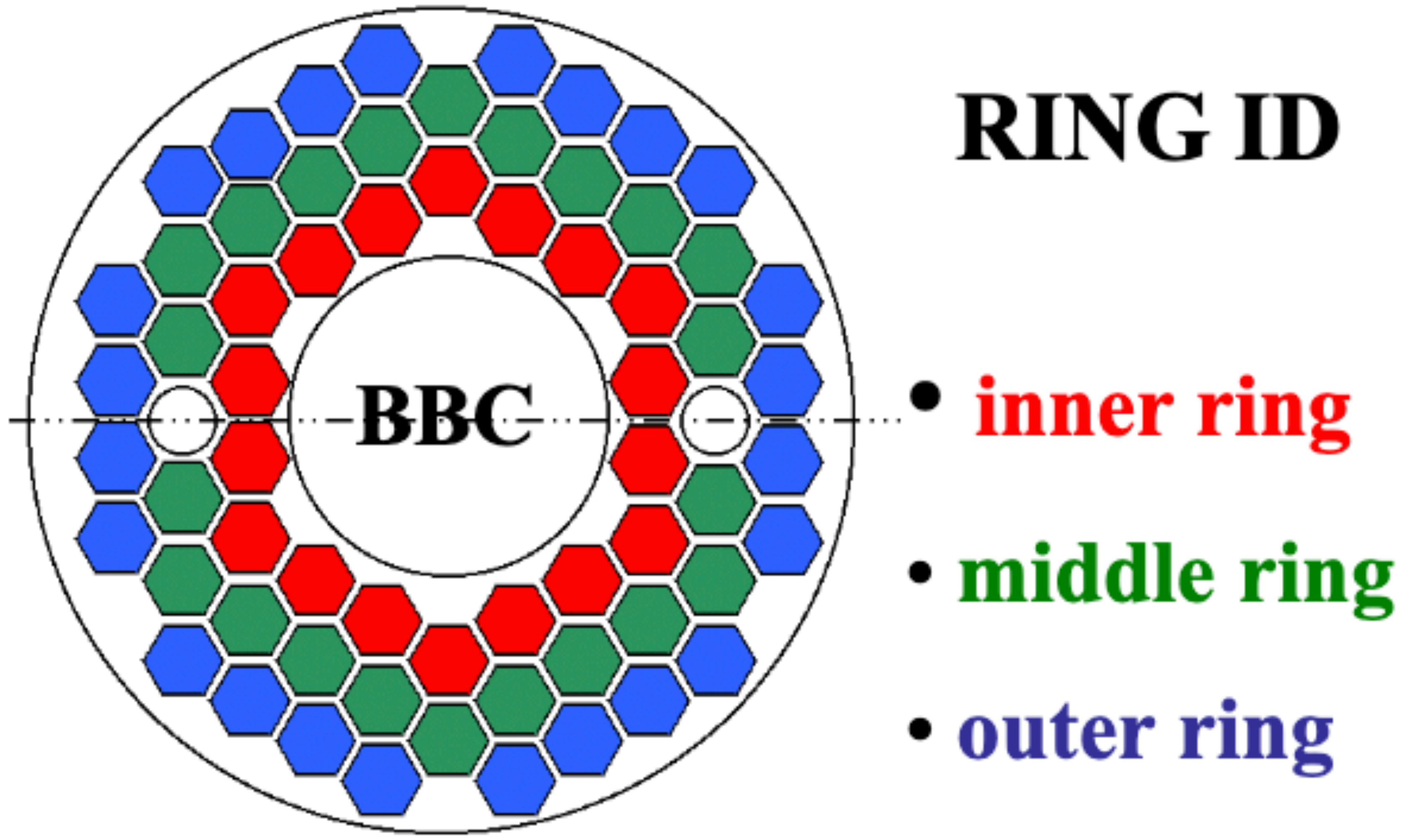
Noriyuki Sugita
sugita@hepl.hiroshima-u.ac.jp

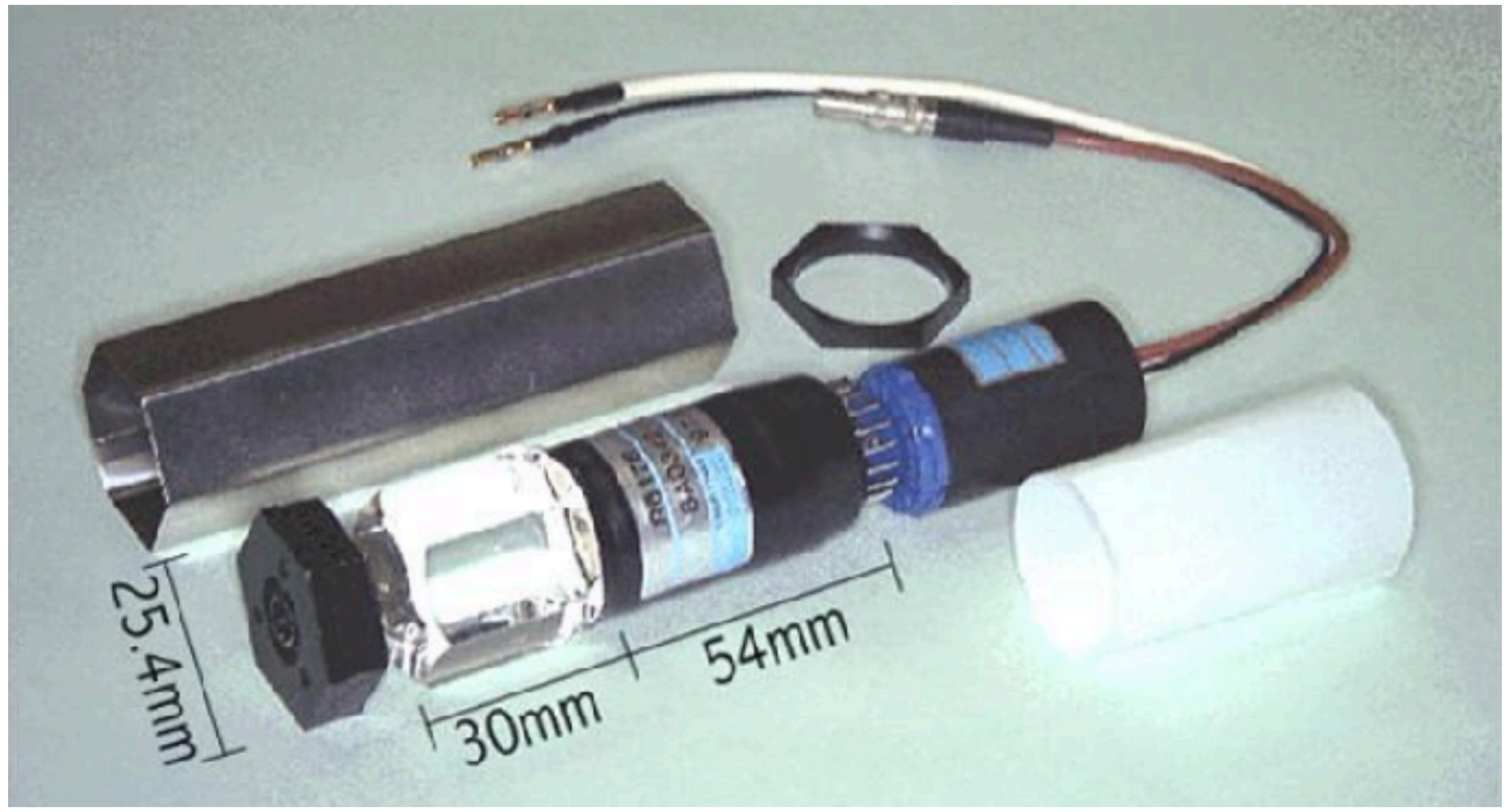
Ryota Kohara
kohara@hepl.hiroshima-u.ac.jp

Tomoaki Nakamura
nakamura@hepl.hiroshima-u.ac.jp

2002/12/17 Tomoaki Nakamura PHENIX Focus 21



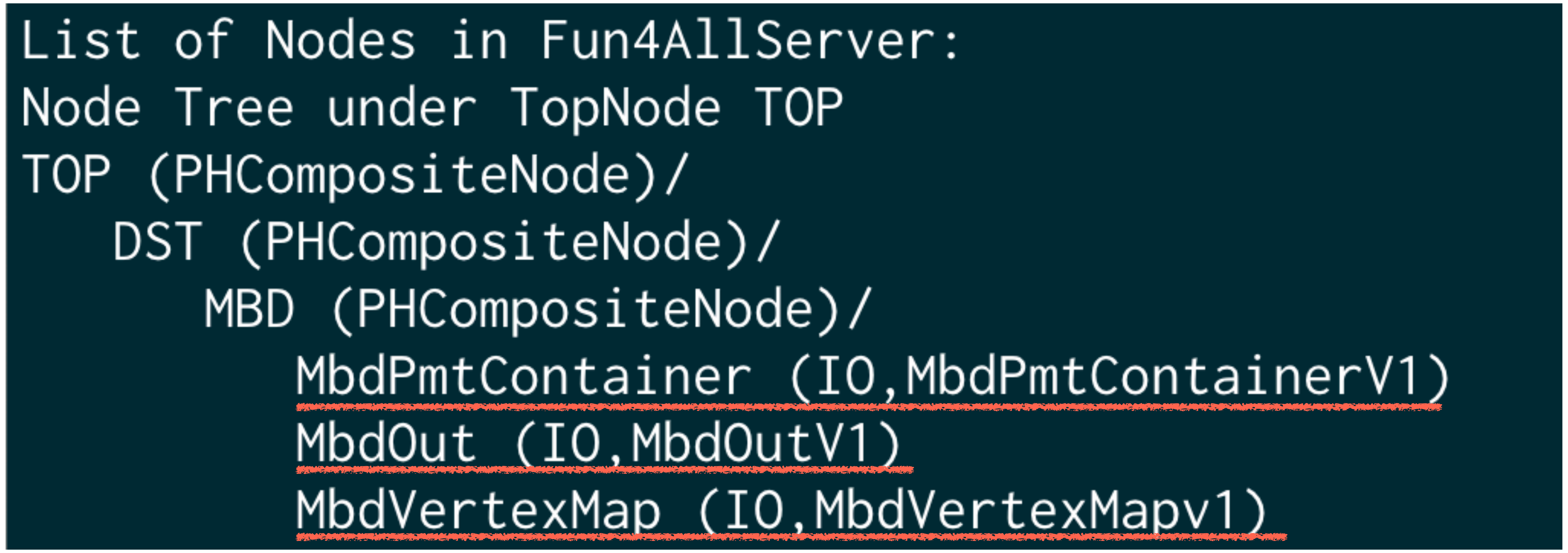




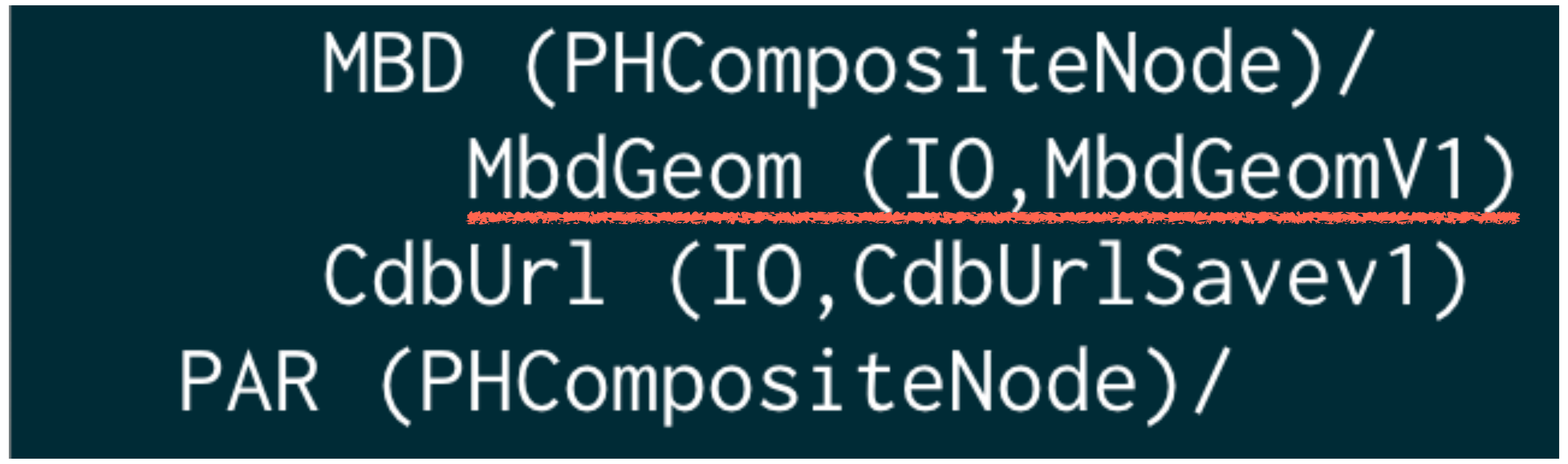
MBD data analysis

- Though I’m not quite sure how to analyze MBD data, I tried to guess.
- In the MDC2 minimum bias data, for example,
  - /sphenix/lustre01/sphnxpro/mdc2/pythia8_pp_mb/nopileup/mbdepd/run0011/DST_MBD_EPD_pythia8_pp_mb-0000000011-00000.root

  there are some nodes related to MBD



```
List of Nodes in Fun4AllServer:
Node Tree under TopNode TOP
TOP (PHCompositeNode)/
   DST (PHCompositeNode)/
      MBD (PHCompositeNode)/
         MbdPmtContainer (IO,MbdPmtContainerV1)
         MbdOut (IO,MbdOutV1)
         MbdVertexMap (IO,MbdVertexMapv1)
```

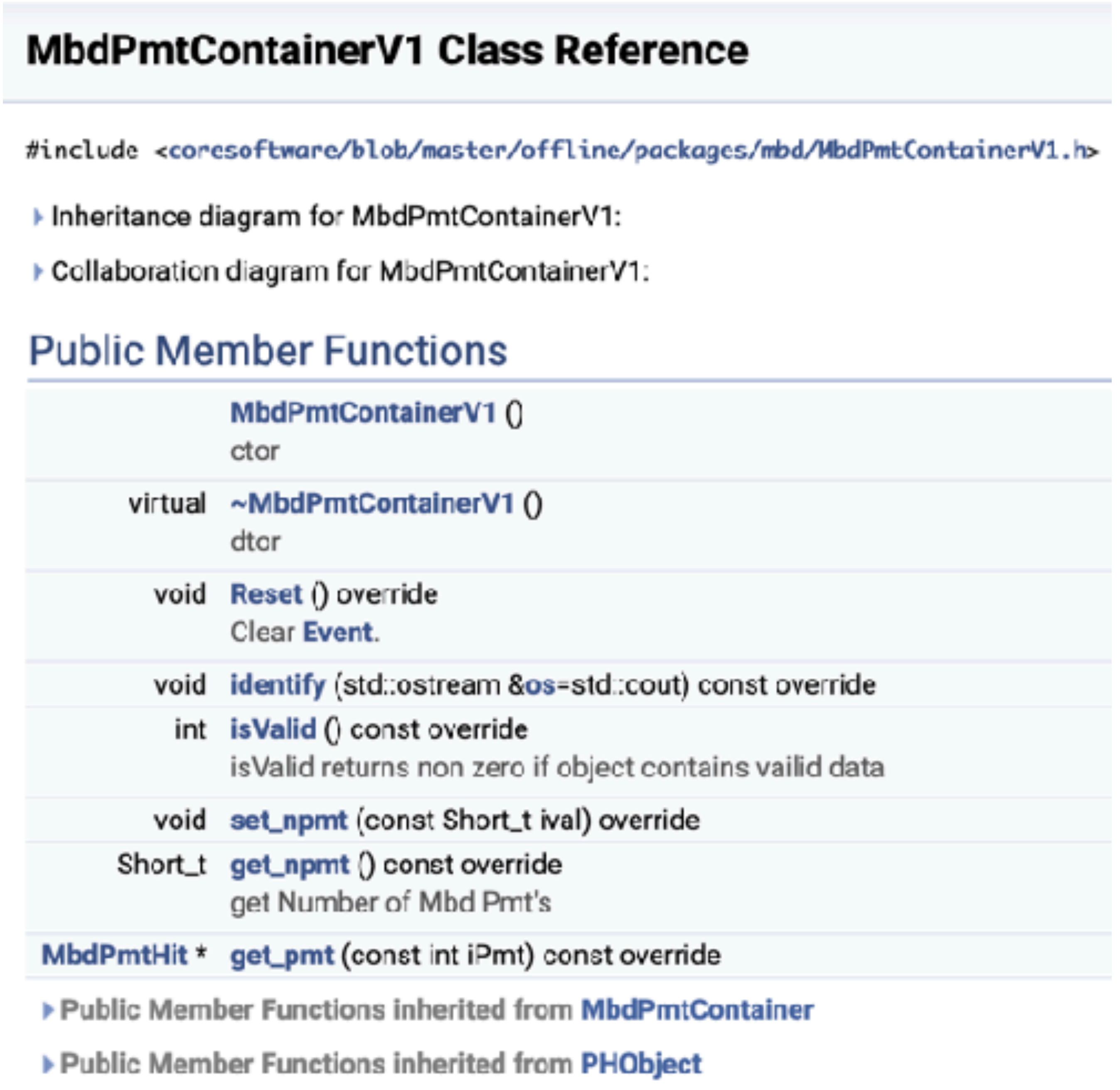


```
      MBD (PHCompositeNode)/
         MbdGeom (IO,MbdGeomV1)
      CdbUrl (IO,CdbUrlSavev1)
   PAR (PHCompositeNode)/
```

MBD data analysis:

MbdPmtContainer

- A container class to access MbdPmtHit



MbdPmtContainerV1 Class Reference

`#include <coresoftware/blob/master/offline/packages/mbd/MbdPmtContainerV1.h>`

- Inheritance diagram for MbdPmtContainerV1:
- Collaboration diagram for MbdPmtContainerV1:

Public Member Functions

	MbdPmtContainerV1 () ctor
virtual	~MbdPmtContainerV1 () dtor
void	Reset () override Clear Event.
void	identify (std::ostream &os=std::cout) const override
int	isValid () const override isValid returns non zero if object contains vailid data
void	set_npmt (const Short_t ival) override
Short_t	get_npmt () const override get Number of Mbd Pmt's
MbdPmtHit *	get_pmt (const int iPmt) const override

- Public Member Functions inherited from MbdPmtContainer
- Public Member Functions inherited from PHObject

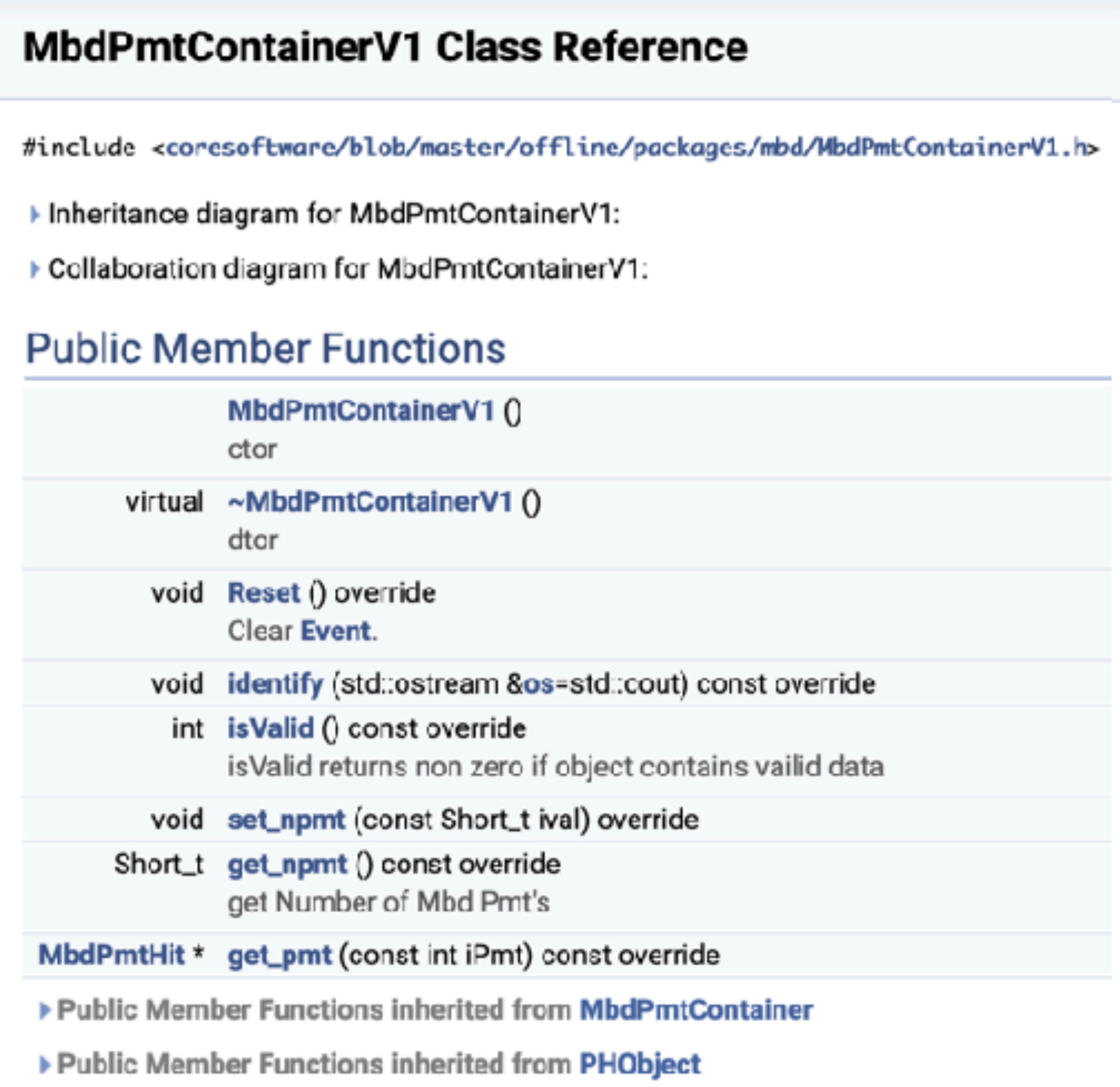
MBD data analysis:

MbdPmtContainer

- A container class to access MbdPmtHit

MbdPmtHit

- It contains charge and timing information



MbdPmtContainerV1 Class Reference

`#include <coresoftware/blob/master/offline/packages/mbd/MbdPmtContainerV1.h>`

▸ Inheritance diagram for MbdPmtContainerV1:

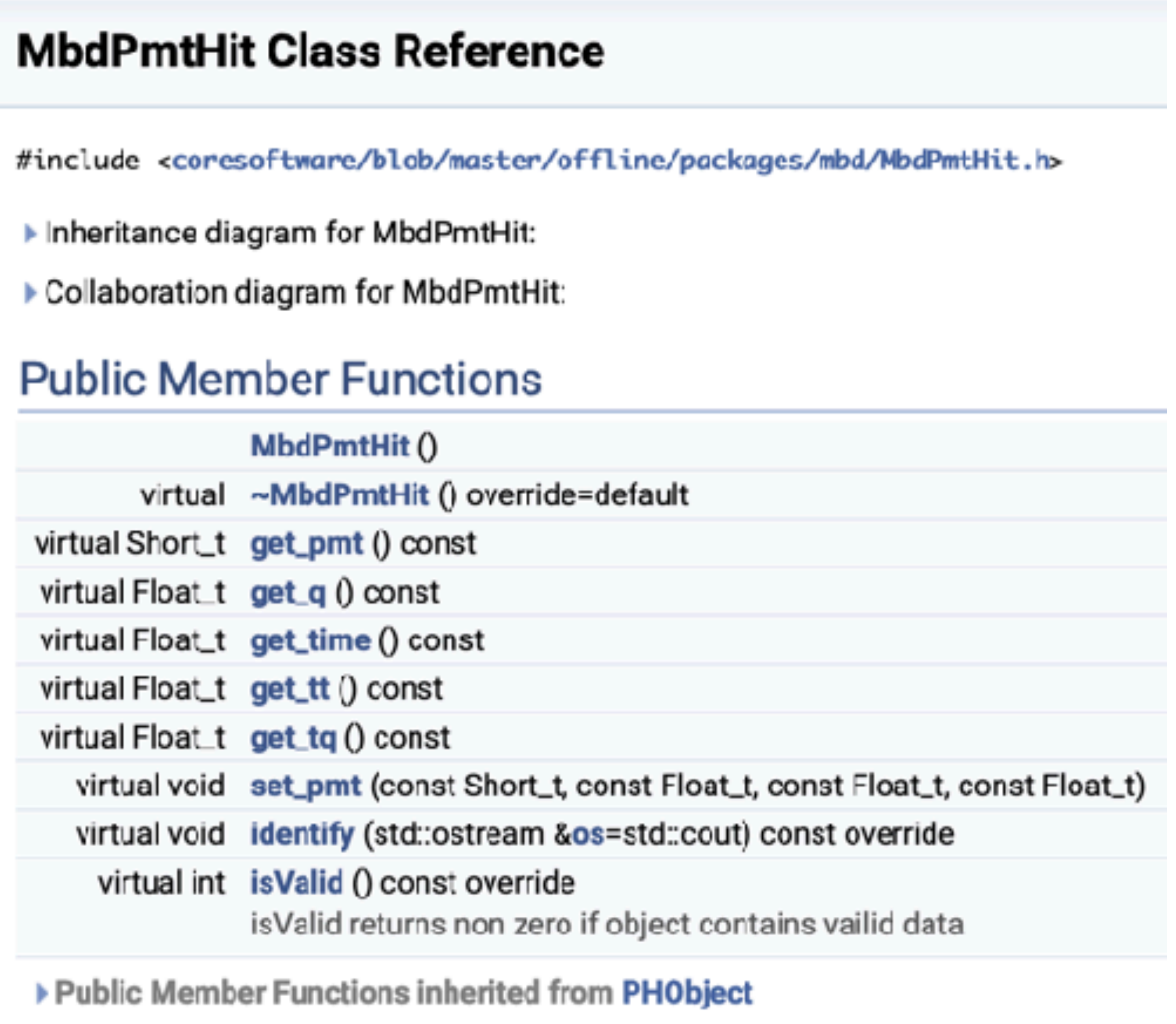
▸ Collaboration diagram for MbdPmtContainerV1:

Public Member Functions

	MbdPmtContainerV1 () ctor
virtual	~MbdPmtContainerV1 () dtor
void	Reset () override Clear Event.
void	identify (std::ostream &os=std::cout) const override
int	isValid () const override isValid returns non zero if object contains vailid data
void	set_npmt (const Short_t ival) override
Short_t	get_npmt () const override get Number of Mbd Pmt's
MbdPmtHit *	get_pmt (const int iPmt) const override

▸ Public Member Functions inherited from MbdPmtContainer

▸ Public Member Functions inherited from PHObject



MbdPmtHit Class Reference

`#include <coresoftware/blob/master/offline/packages/mbd/MbdPmtHit.h>`

▸ Inheritance diagram for MbdPmtHit:

▸ Collaboration diagram for MbdPmtHit:

Public Member Functions

	MbdPmtHit ()
virtual	~MbdPmtHit () override=default
virtual Short_t	get_pmt () const
virtual Float_t	get_q () const
virtual Float_t	get_time () const
virtual Float_t	get_tt () const
virtual Float_t	get_tq () const
virtual void	set_pmt (const Short_t, const Float_t, const Float_t, const Float_t)
virtual void	identify (std::ostream &os=std::cout) const override
virtual int	isValid () const override isValid returns non zero if object contains vailid data

▸ Public Member Functions inherited from PHObject

MBD data analysis:

MbdPmtContainer

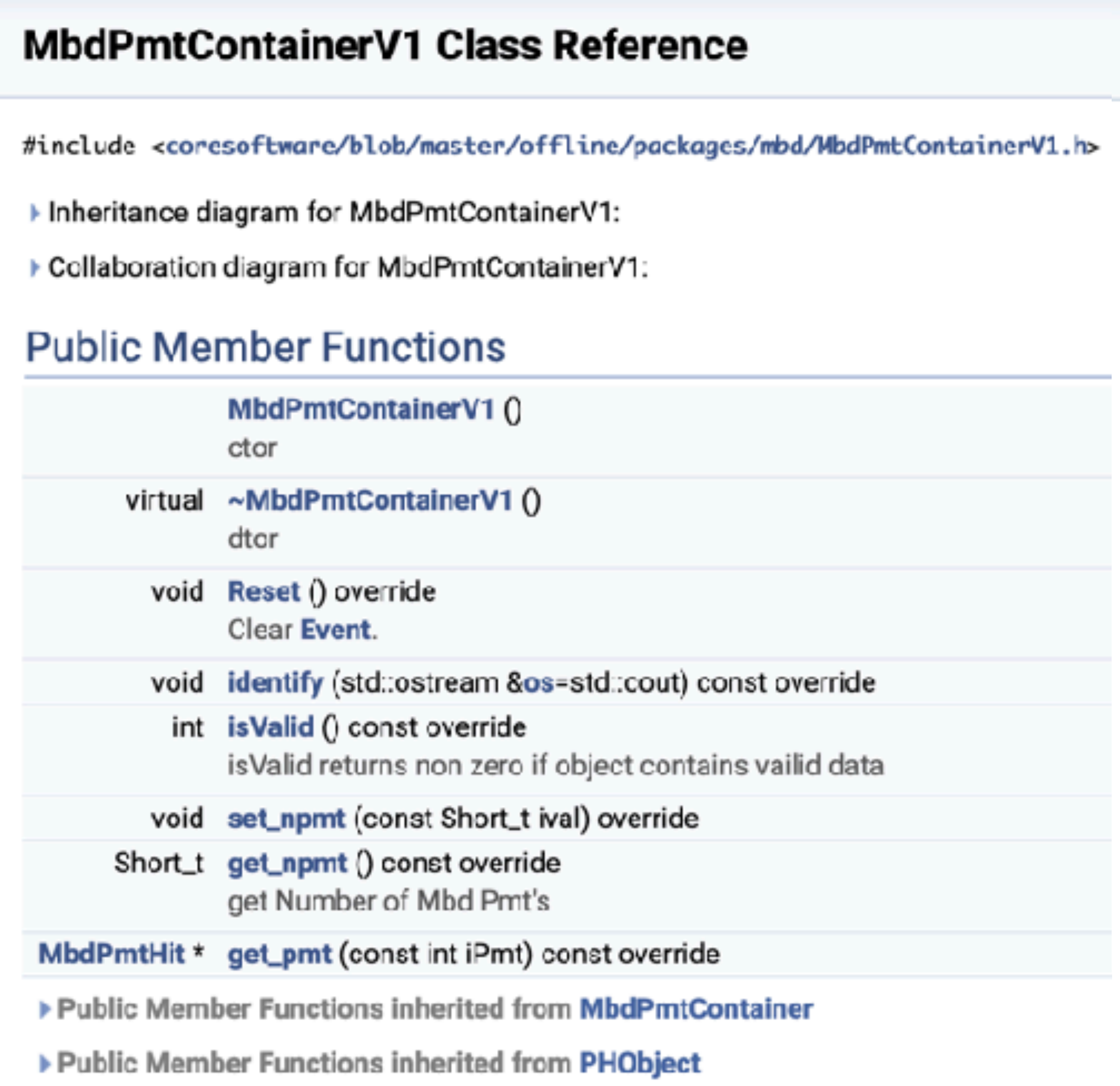
- A container class to access MbdPmtHit

MbdPmtHit

- It contains charge and timing information

MbdGeom

- We can get position information (x, y, z, r, ϕ, arm) of MBD PMT



MbdPmtContainerV1 Class Reference

#include <coresoftware/blob/master/offline/packages/mbd/MbdPmtContainerV1.h>

▸ Inheritance diagram for MbdPmtContainerV1:

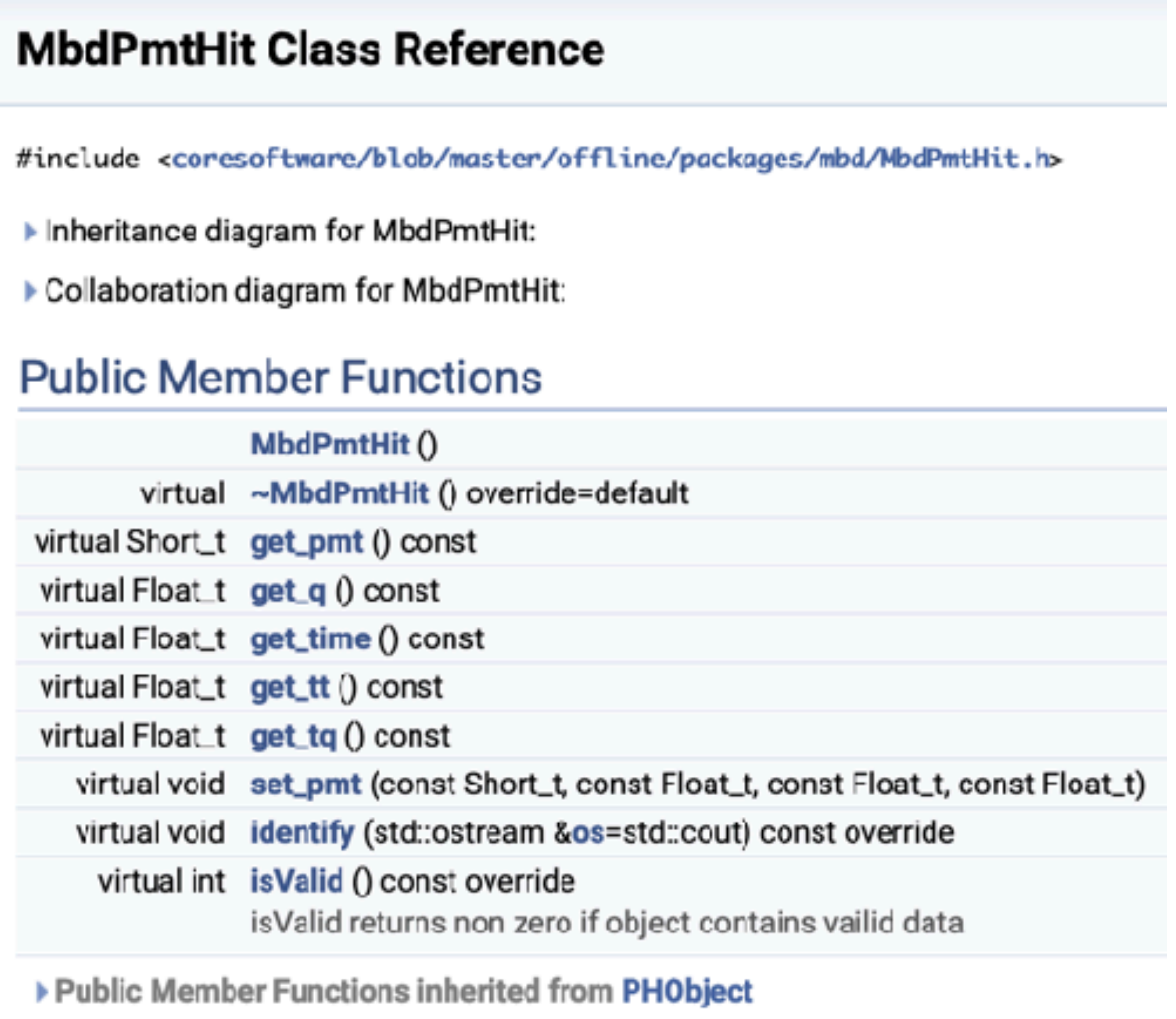
▸ Collaboration diagram for MbdPmtContainerV1:

Public Member Functions

	MbdPmtContainerV1 () ctor
virtual	~MbdPmtContainerV1 () dtor
void	Reset () override Clear Event.
void	identify (std::ostream &os=std::cout) const override
int	isValid () const override isValid returns non zero if object contains vailid data
void	set_npmt (const Short_t ival) override
Short_t	get_npmt () const override get Number of Mbd Pmt's
MbdPmtHit *	get_pmt (const int iPmt) const override

▸ Public Member Functions inherited from MbdPmtContainer

▸ Public Member Functions inherited from PHObject



MbdPmtHit Class Reference

#include <coresoftware/blob/master/offline/packages/mbd/MbdPmtHit.h>

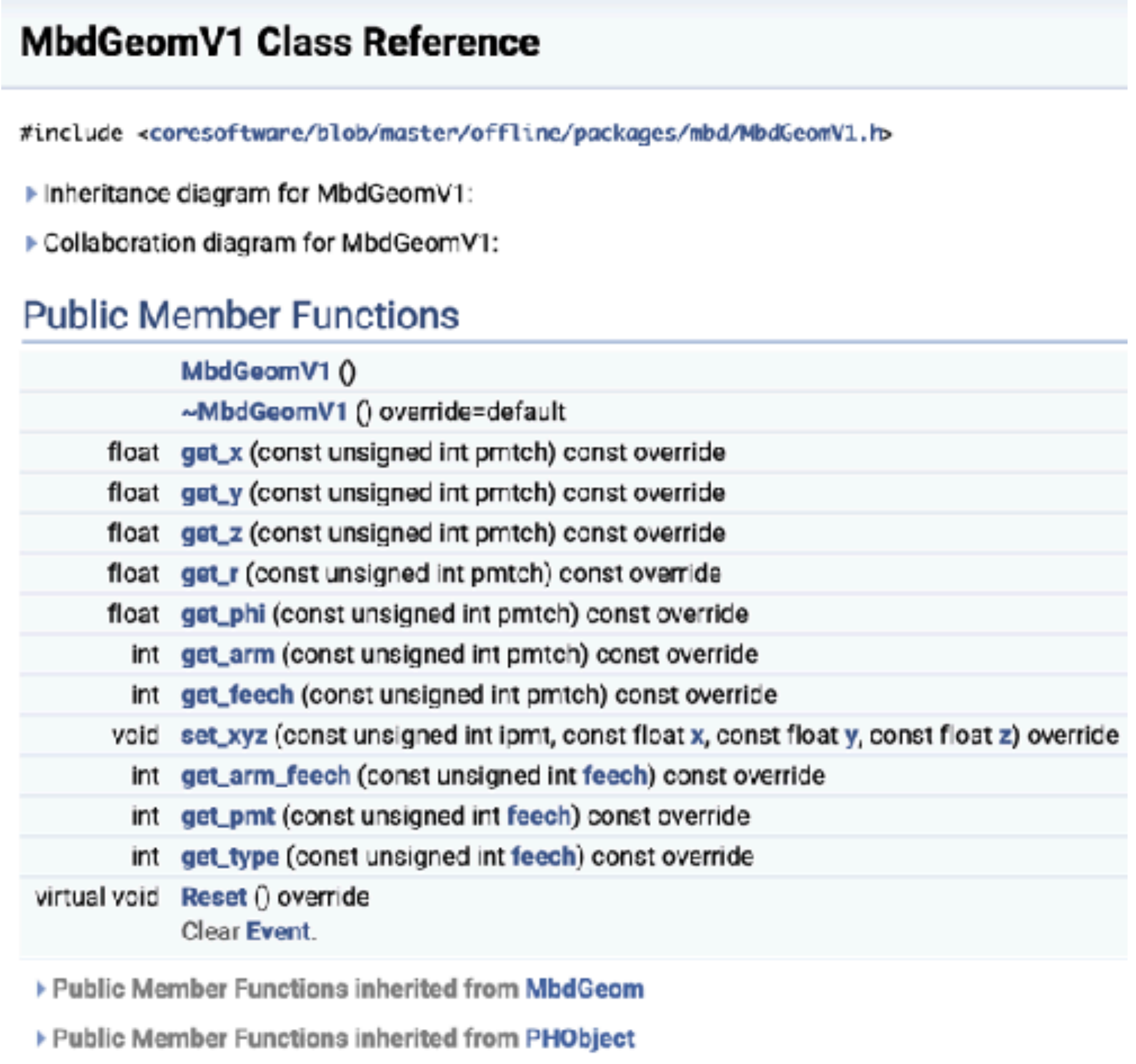
▸ Inheritance diagram for MbdPmtHit:

▸ Collaboration diagram for MbdPmtHit:

Public Member Functions

	MbdPmtHit ()
virtual	~MbdPmtHit () override=default
virtual Short_t	get_pmt () const
virtual Float_t	get_q () const
virtual Float_t	get_time () const
virtual Float_t	get_tt () const
virtual Float_t	get_tq () const
virtual void	set_pmt (const Short_t, const Float_t, const Float_t, const Float_t)
virtual void	identify (std::ostream &os=std::cout) const override
virtual int	isValid () const override isValid returns non zero if object contains vailid data

▸ Public Member Functions inherited from PHObject



MbdGeomV1 Class Reference

#include <coresoftware/blob/master/offline/packages/mbd/MbdGeomV1.h>

▸ Inheritance diagram for MbdGeomV1:

▸ Collaboration diagram for MbdGeomV1:

Public Member Functions

	MbdGeomV1 ()
	~MbdGeomV1 () override=default
float	get_x (const unsigned int pmtch) const override
float	get_y (const unsigned int pmtch) const override
float	get_z (const unsigned int pmtch) const override
float	get_r (const unsigned int pmtch) const override
float	get_phi (const unsigned int pmtch) const override
int	get_arm (const unsigned int pmtch) const override
int	get_feech (const unsigned int pmtch) const override
void	set_xyz (const unsigned int ipmt, const float x, const float y, const float z) override
int	get_arm_feech (const unsigned int feech) const override
int	get_pmt (const unsigned int feech) const override
int	get_type (const unsigned int feech) const override
virtual void	Reset () override Clear Event.

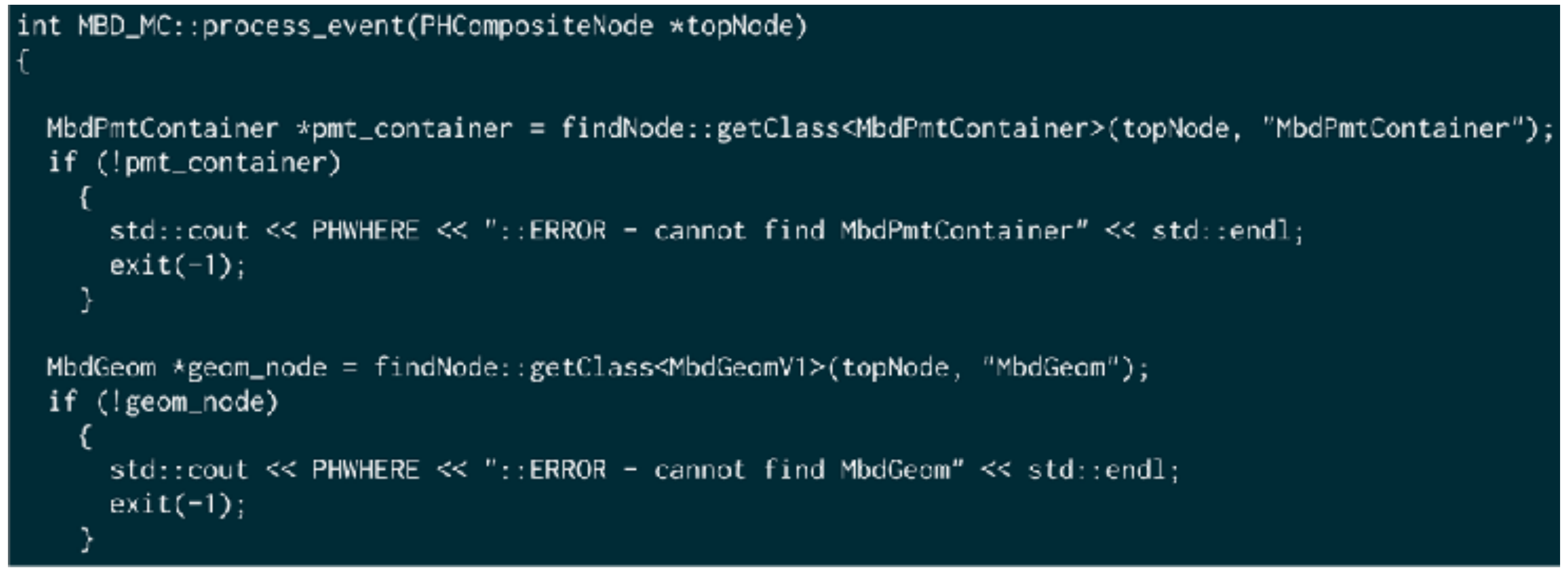
▸ Public Member Functions inherited from MbdGeom

▸ Public Member Functions inherited from PHObject

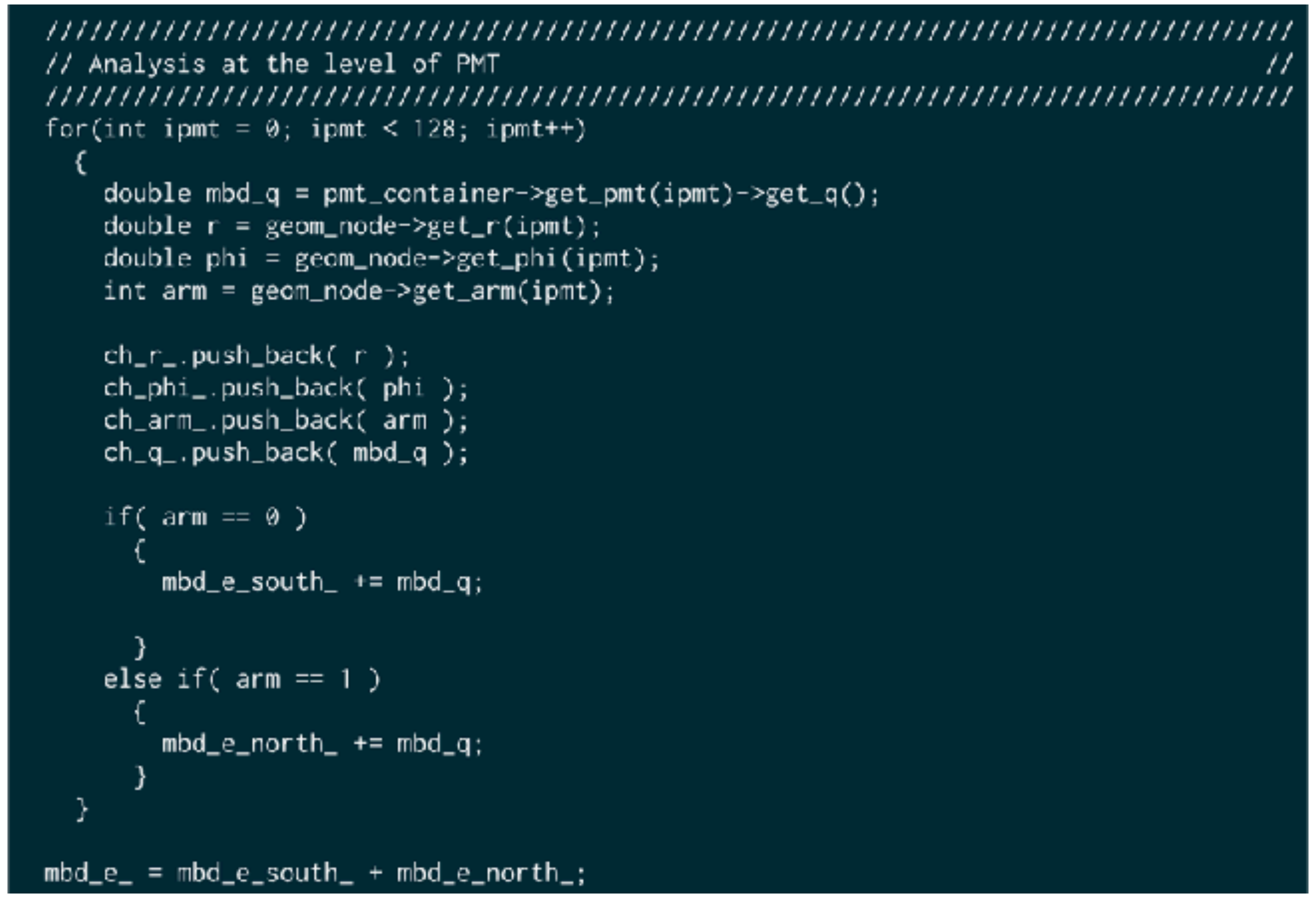
MBD data analysis:

PMT level analysis

- I learned Ejiro’s codes to use those 3 classes.
- 1M minimum bias MC events were analyzed



```
int MBD_MC::process_event(PHCompositeNode *topNode)
{

  MbdPmtContainer *pmt_container = findNode::getClass<MbdPmtContainer>(topNode, "MbdPmtContainer");
  if (!pmt_container)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdPmtContainer" << std::endl;
      exit(-1);
    }

  MbdGeom *geom_node = findNode::getClass<MbdGeomV1>(topNode, "MbdGeom");
  if (!geom_node)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdGeom" << std::endl;
      exit(-1);
    }
```

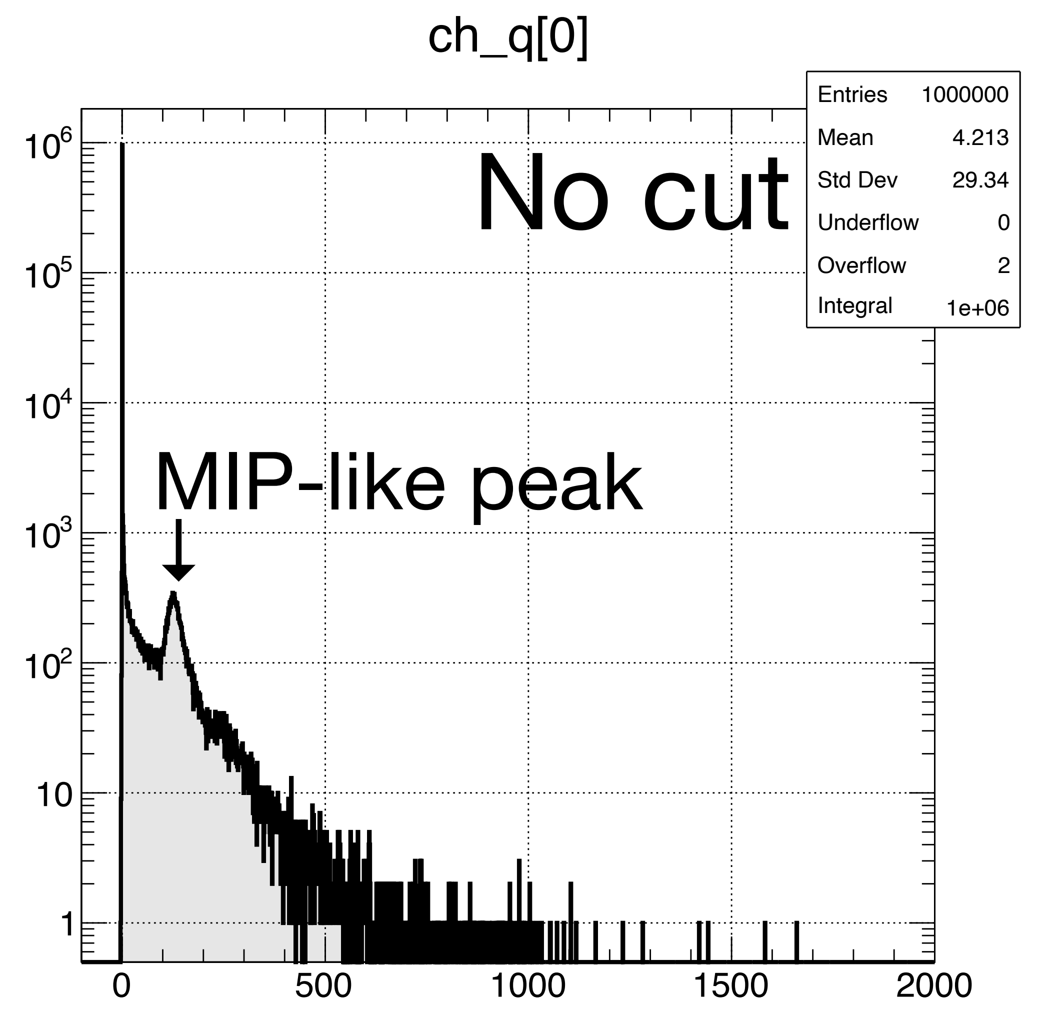


```
  /////////////////////////////////////////////////////////////////////////////////////
  // Analysis at the level of PMT                                                    //
  /////////////////////////////////////////////////////////////////////////////////////
  for(int ipmt = 0; ipmt < 128; ipmt++)
    {
      double mbd_q = pmt_container->get_pmt(ipmt)->get_q();
      double r = geom_node->get_r(ipmt);
      double phi = geom_node->get_phi(ipmt);
      int arm = geom_node->get_arm(ipmt);

      ch_r_.push_back( r );
      ch_phi_.push_back( phi );
      ch_arm_.push_back( arm );
      ch_q_.push_back( mbd_q );

      if( arm == 0 )
        {
          mbd_e_south_ += mbd_q;

        }
      else if( arm == 1 )
        {
          mbd_e_north_ += mbd_q;
        }
    }

  mbd_e_ = mbd_e_south_ + mbd_e_north_;
```

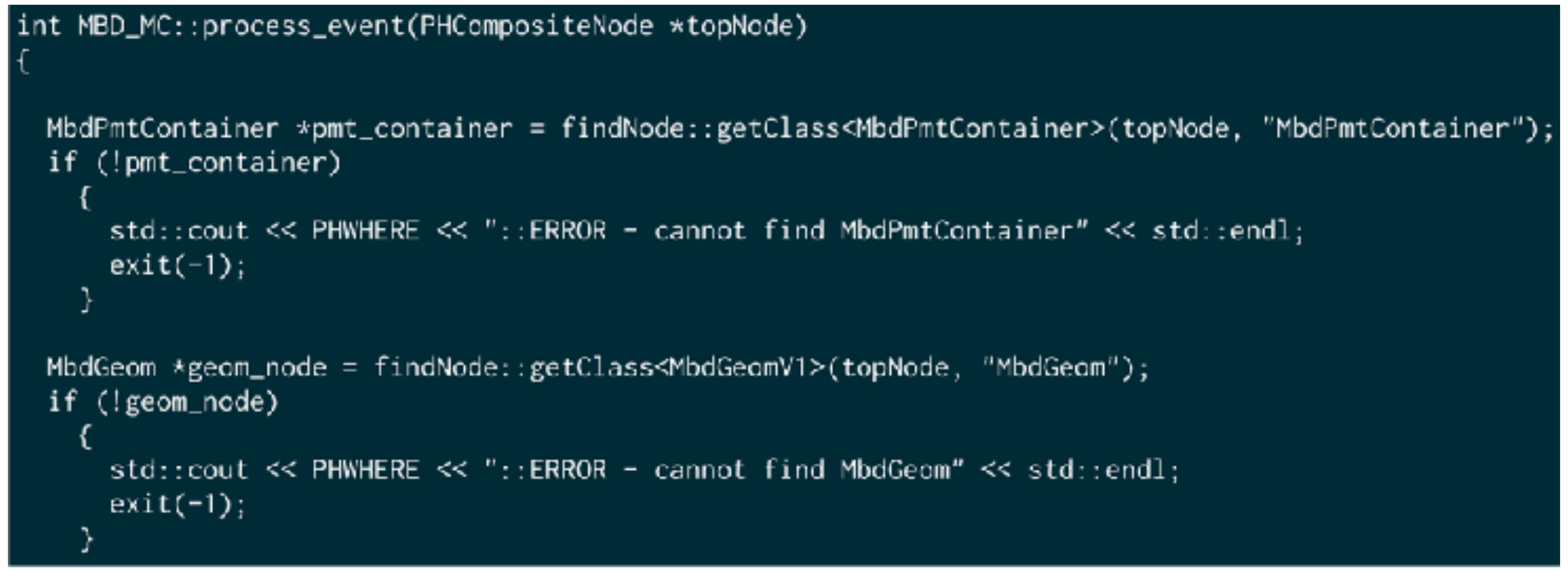
MBD data analysis:

PMT level analysis

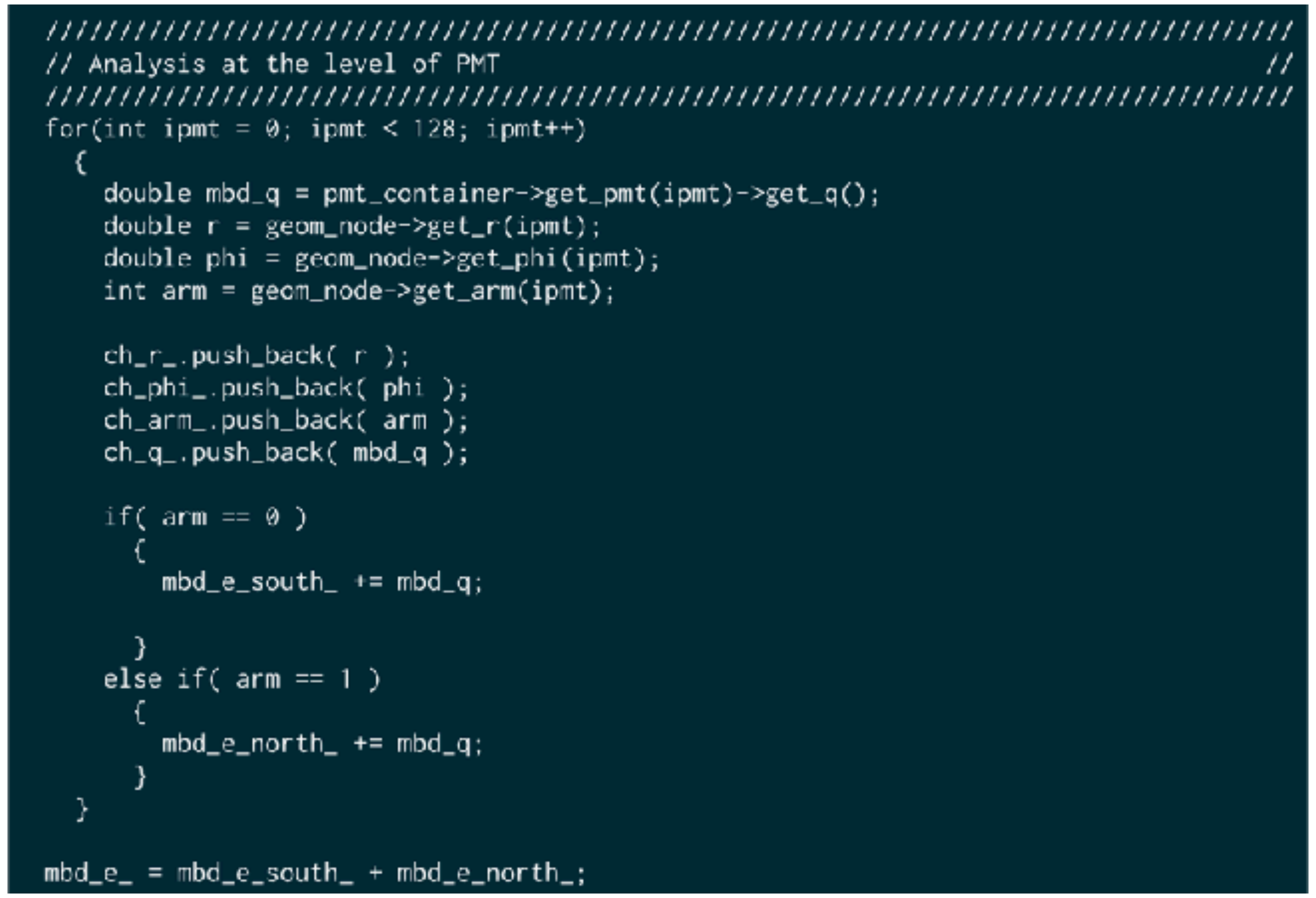
- I learned Ejiro’s codes to use those 3 classes.
- 1M minimum bias MC events were analyzed



Charge measured by PMT#0
(r=13.014552, ϕ=2.8081204, south)



```
int MBD_MC::process_event(PHCompositeNode *topNode)
{

  MbdPmtContainer *pmt_container = findNode::getClass<MbdPmtContainer>(topNode, "MbdPmtContainer");
  if (!pmt_container)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdPmtContainer" << std::endl;
      exit(-1);
    }

  MbdGeom *geom_node = findNode::getClass<MbdGeomV1>(topNode, "MbdGeom");
  if (!geom_node)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdGeom" << std::endl;
      exit(-1);
    }
```

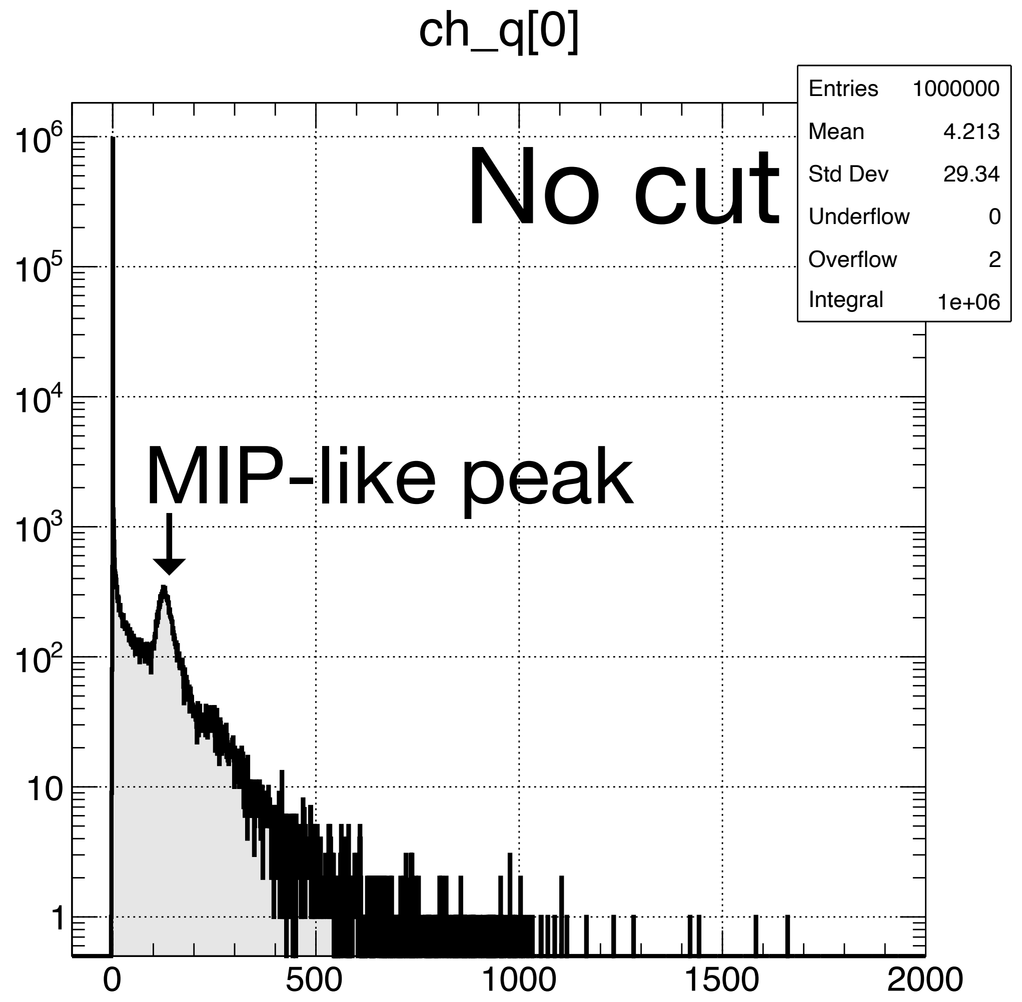


```
  /////////////////////////////////////////////////////////////////////////////////////
  // Analysis at the level of PMT                                                    //
  /////////////////////////////////////////////////////////////////////////////////////
  for(int ipmt = 0; ipmt < 128; ipmt++)
    {
      double mbd_q = pmt_container->get_pmt(ipmt)->get_q();
      double r = geom_node->get_r(ipmt);
      double phi = geom_node->get_phi(ipmt);
      int arm = geom_node->get_arm(ipmt);

      ch_r_.push_back( r );
      ch_phi_.push_back( phi );
      ch_arm_.push_back( arm );
      ch_q_.push_back( mbd_q );

      if( arm == 0 )
        {
          mbd_e_south_ += mbd_q;

        }
      else if( arm == 1 )
        {
          mbd_e_north_ += mbd_q;
        }
    }

  mbd_e_ = mbd_e_south_ + mbd_e_north_;
```

MBD data analysis:

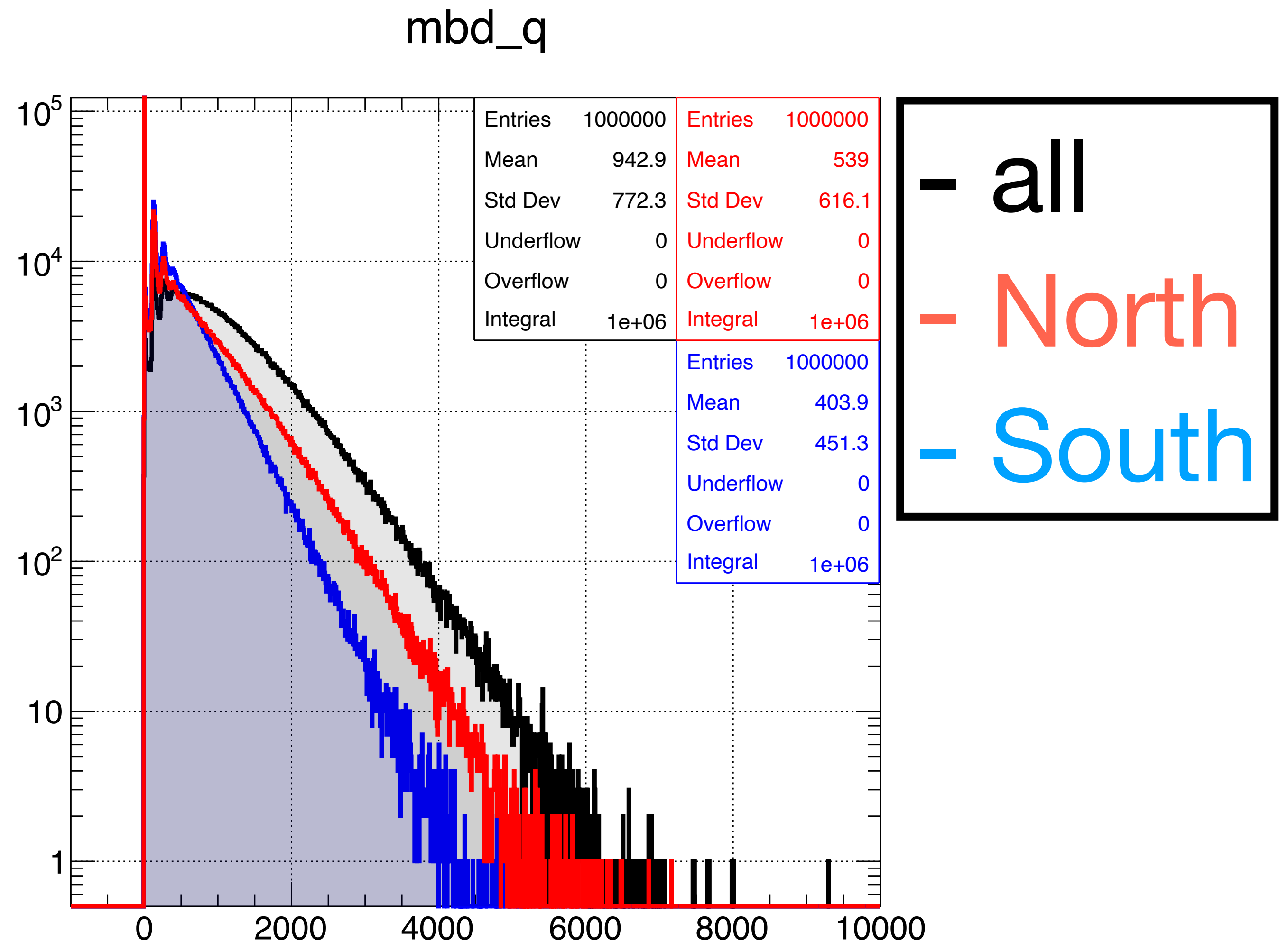
PMT level analysis

- I learned Ejiro’s codes to use those 3 classes.
- 1M minimum bias MC events were analyzed

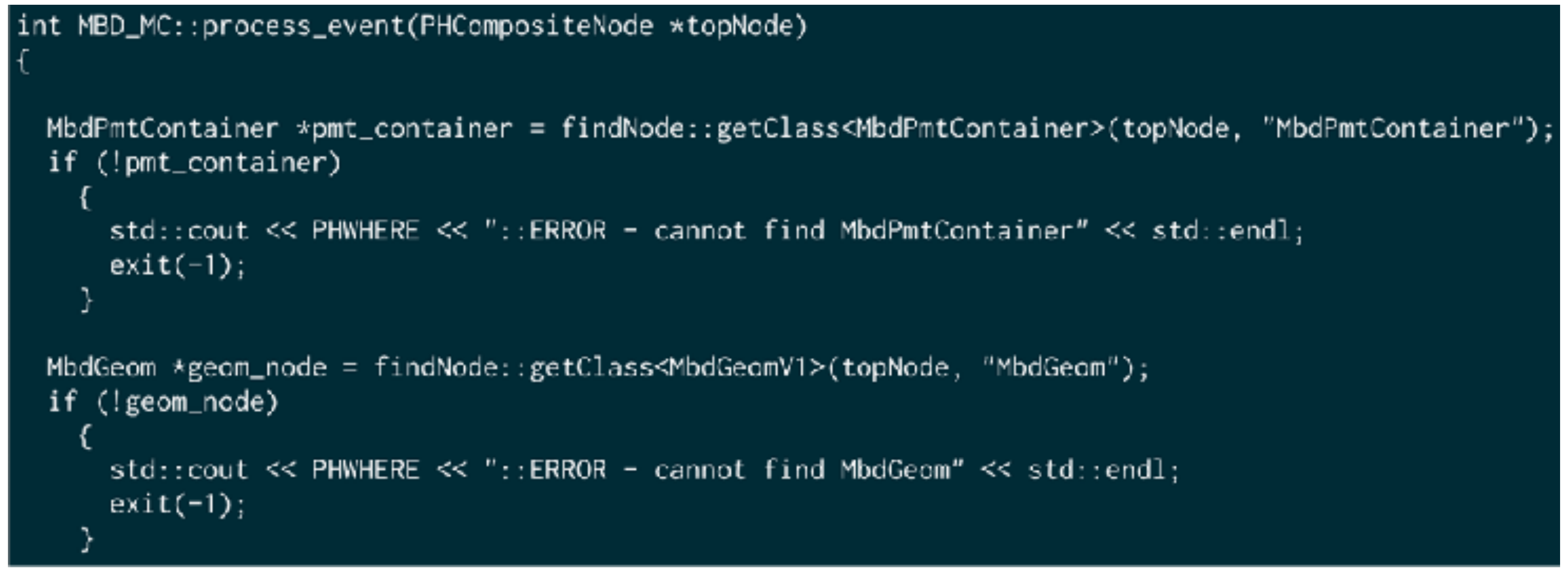


Charge measured by PMT#0

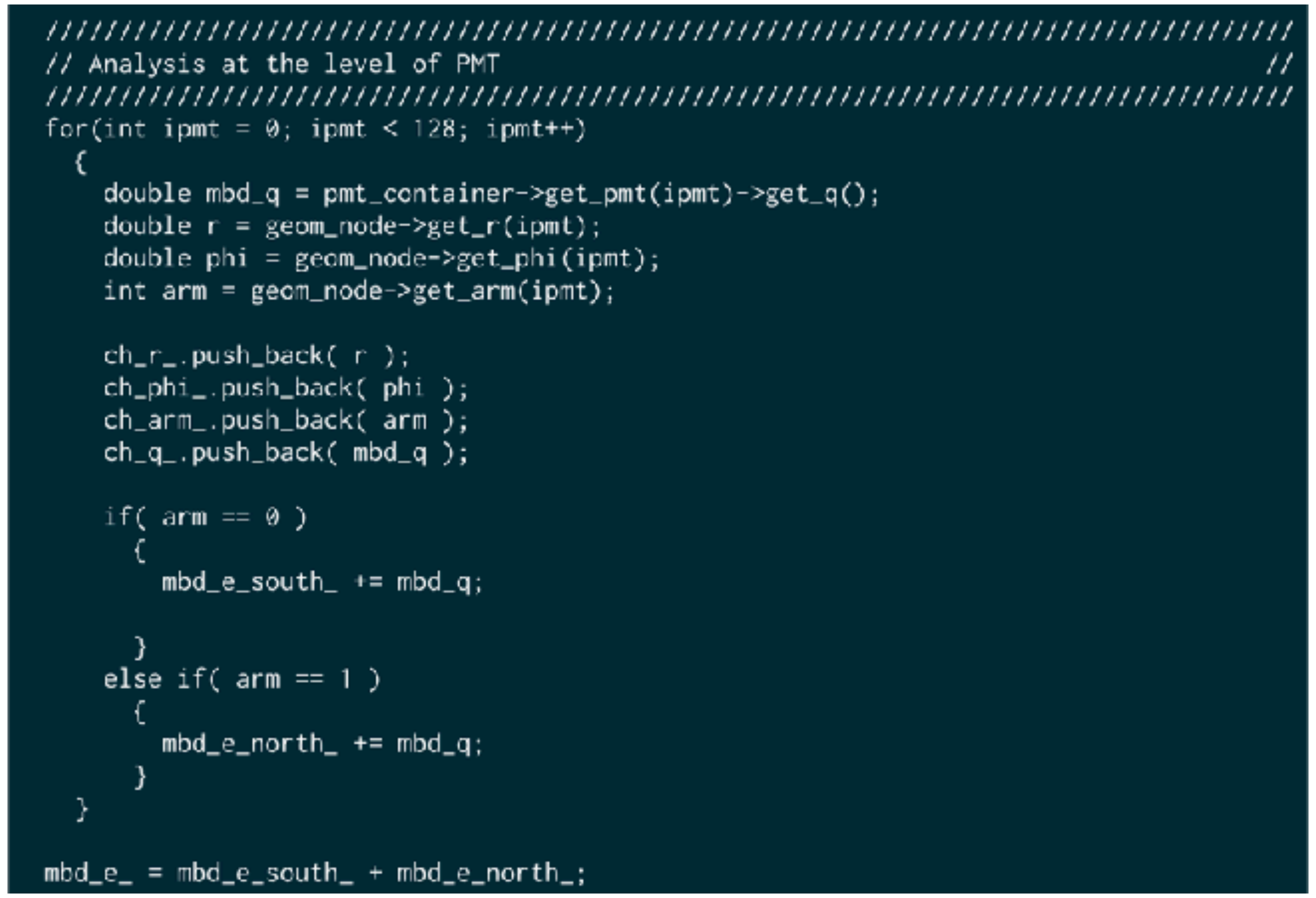
(r=13.014552, ϕ=2.8081204, south)



Charge distribution



```
int MBD_MC::process_event(PHCompositeNode *topNode)
{

  MbdPmtContainer *pmt_container = findNode::getClass<MbdPmtContainer>(topNode, "MbdPmtContainer");
  if (!pmt_container)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdPmtContainer" << std::endl;
      exit(-1);
    }

  MbdGeom *geom_node = findNode::getClass<MbdGeomV1>(topNode, "MbdGeom");
  if (!geom_node)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdGeom" << std::endl;
      exit(-1);
    }
```

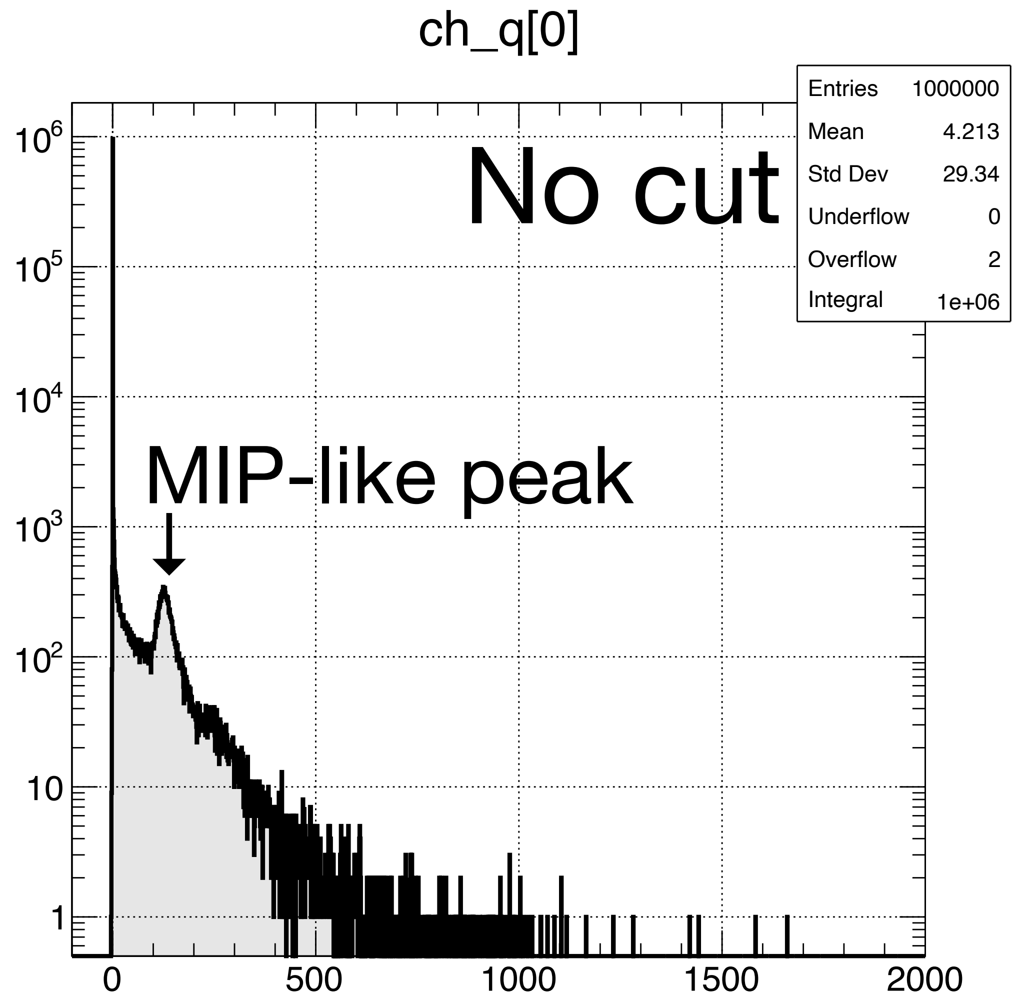


```
  ////////////////////////////////////////////////////////////////////////////////
  // Analysis at the level of PMT                                              //
  ////////////////////////////////////////////////////////////////////////////////
  for(int ipmt = 0; ipmt < 128; ipmt++)
    {
      double mbd_q = pmt_container->get_pmt(ipmt)->get_q();
      double r = geom_node->get_r(ipmt);
      double phi = geom_node->get_phi(ipmt);
      int arm = geom_node->get_arm(ipmt);

      ch_r_.push_back( r );
      ch_phi_.push_back( phi );
      ch_arm_.push_back( arm );
      ch_q_.push_back( mbd_q );

      if( arm == 0 )
        {
          mbd_e_south_ += mbd_q;

        }
      else if( arm == 1 )
        {
          mbd_e_north_ += mbd_q;
        }
    }

  mbd_e_ = mbd_e_south_ + mbd_e_north_;
```

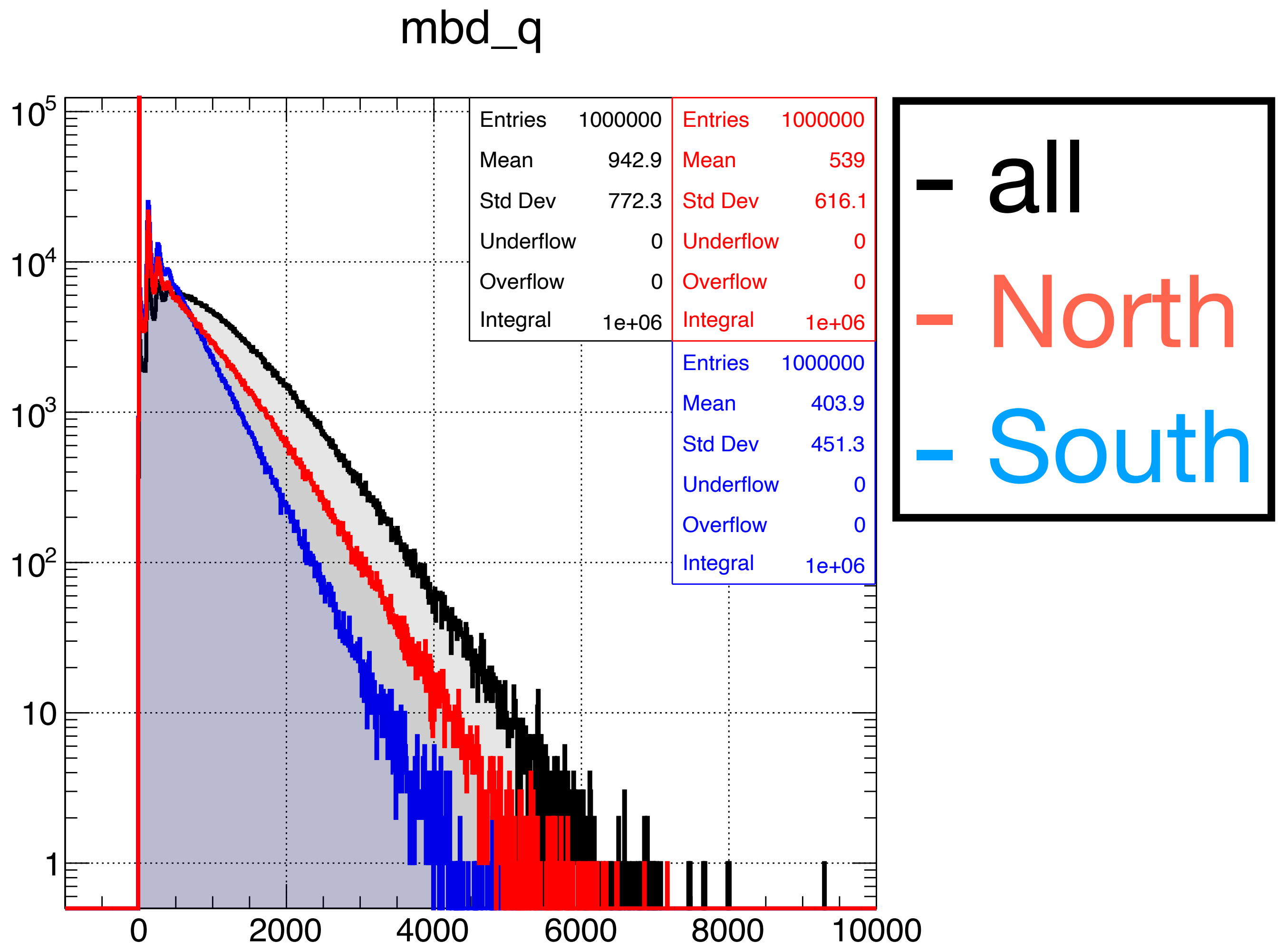
MBD data analysis:

PMT level analysis

- I learned Ejiro's codes to use those 3 classes.
- 1M minimum bias MC events were analyzed

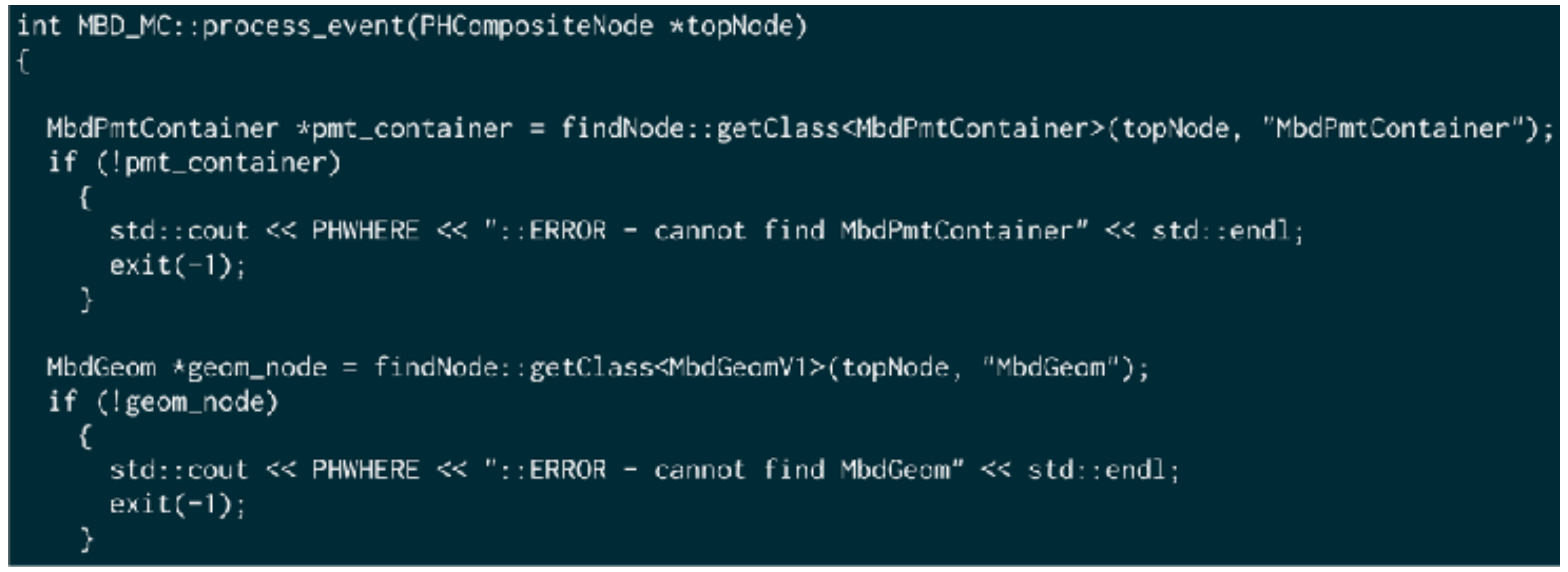


Charge measured by PMT#0
(r=13.014552, ϕ=2.8081204, south)

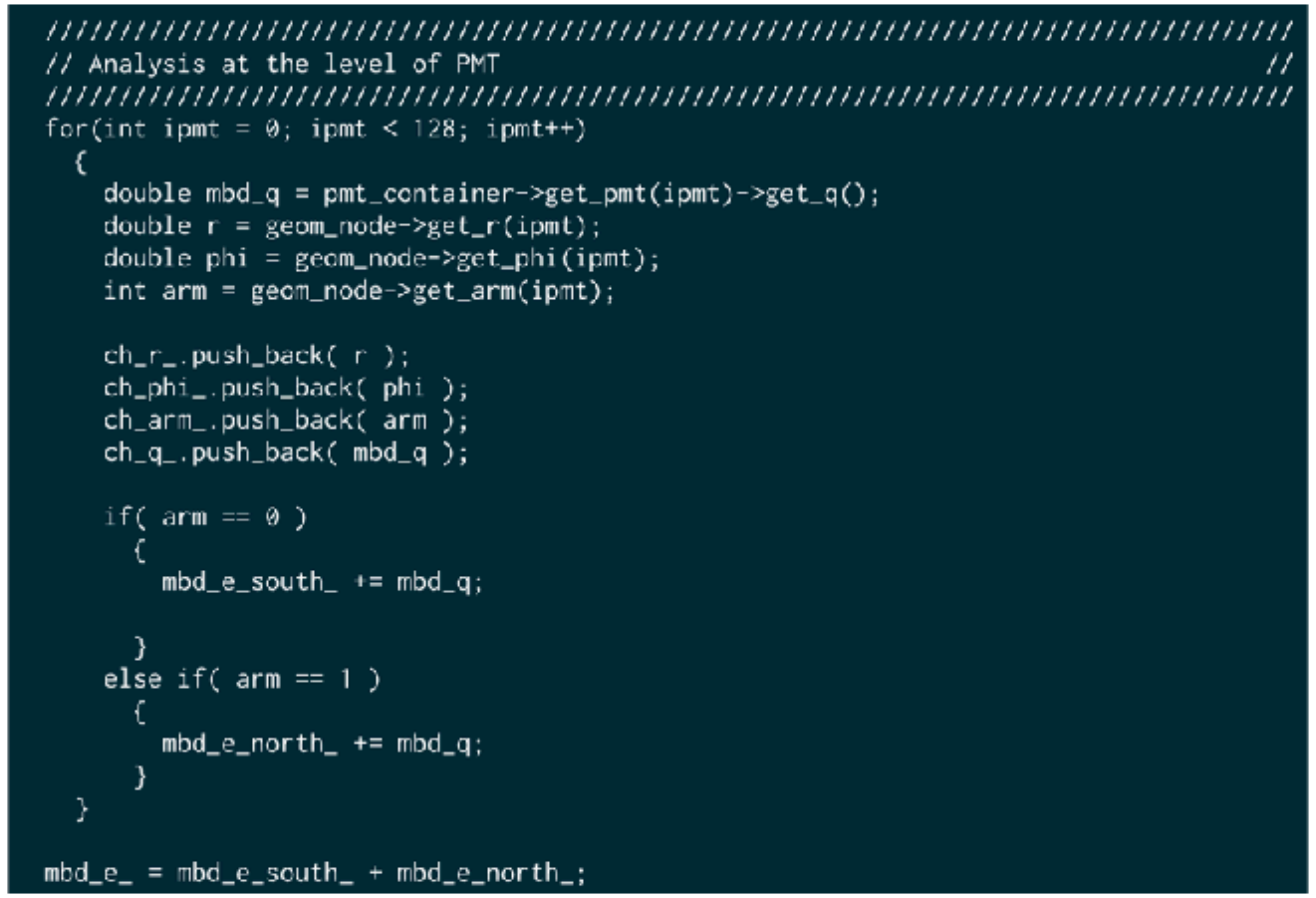


Charge distribution

OK, then, how can I simulate the MBD trigger?



```
int MBD_MC::process_event(PHCompositeNode *topNode)
{

  MbdPmtContainer *pmt_container = findNode::getClass<MbdPmtContainer>(topNode, "MbdPmtContainer");
  if (!pmt_container)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdPmtContainer" << std::endl;
      exit(-1);
    }

  MbdGeom *geom_node = findNode::getClass<MbdGeomV1>(topNode, "MbdGeom");
  if (!geom_node)
    {
      std::cout << PHWHERE << "::ERROR - cannot find MbdGeom" << std::endl;
      exit(-1);
    }
```

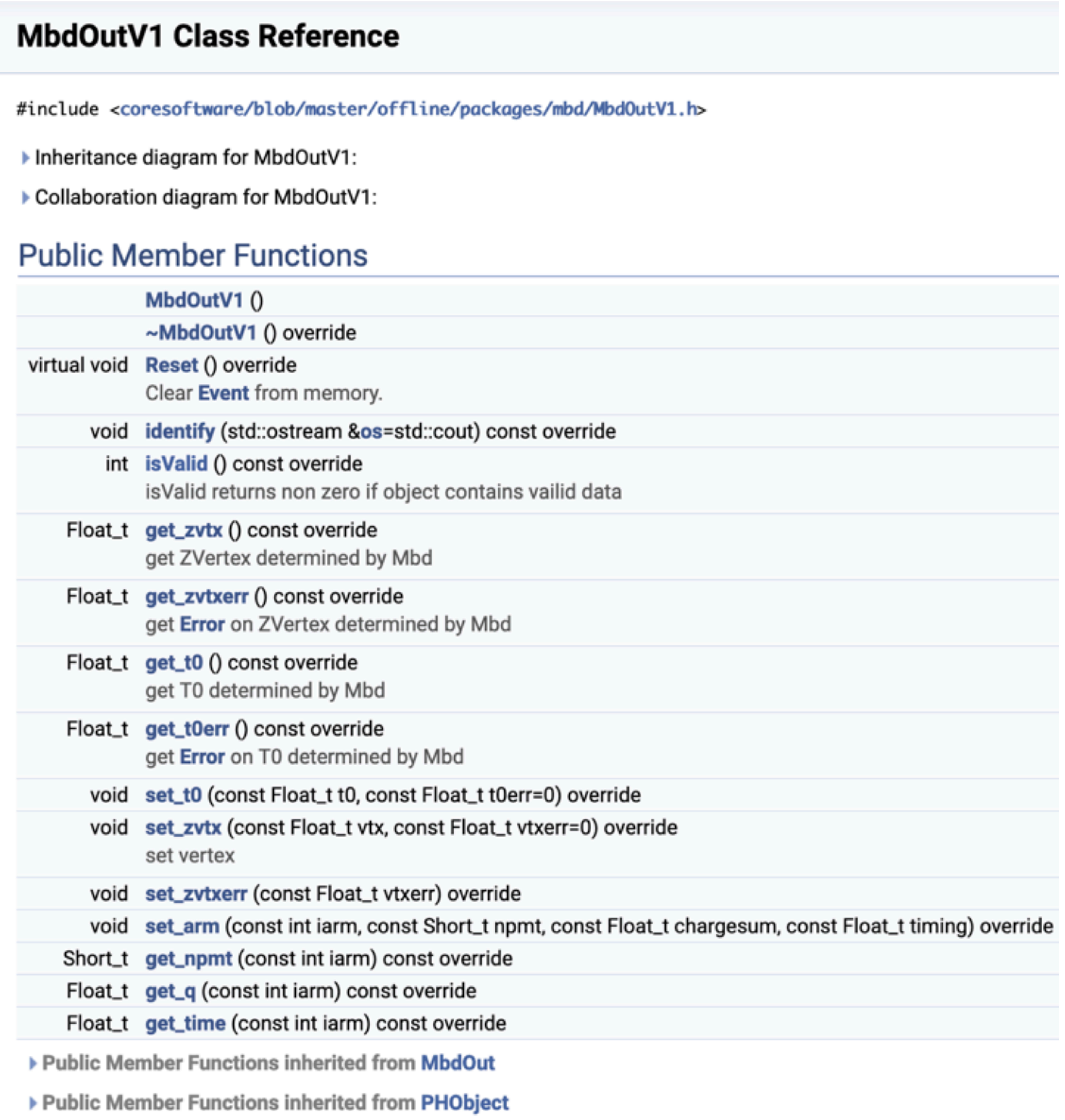


```
  ////////////////////////////////////////////////////////////////////////////////
  // Analysis at the level of PMT                                               //
  ////////////////////////////////////////////////////////////////////////////////
  for(int ipmt = 0; ipmt < 128; ipmt++)
    {
      double mbd_q = pmt_container->get_pmt(ipmt)->get_q();
      double r = geom_node->get_r(ipmt);
      double phi = geom_node->get_phi(ipmt);
      int arm = geom_node->get_arm(ipmt);

      ch_r_.push_back( r );
      ch_phi_.push_back( phi );
      ch_arm_.push_back( arm );
      ch_q_.push_back( mbd_q );

      if( arm == 0 )
        {
          mbd_e_south_ += mbd_q;

        }
      else if( arm == 1 )
        {
          mbd_e_north_ += mbd_q;
        }
    }

  mbd_e_ = mbd_e_south_ + mbd_e_north_;
```

MBD data analysis:

MbdOut

- I could find no explanation about this class.
- Judging from its member functions, this class provides ana results using MBD data. We can get
  - z_{vtx}, error of z_{vtx}
  - t_0, error of t_0
  - #fired PMT
  - charge



MbdOutV1 Class Reference

`#include <coresoftware/blob/master/offline/packages/mbd/MbdOutV1.h>`

▸ Inheritance diagram for MbdOutV1:

▸ Collaboration diagram for MbdOutV1:

Public Member Functions

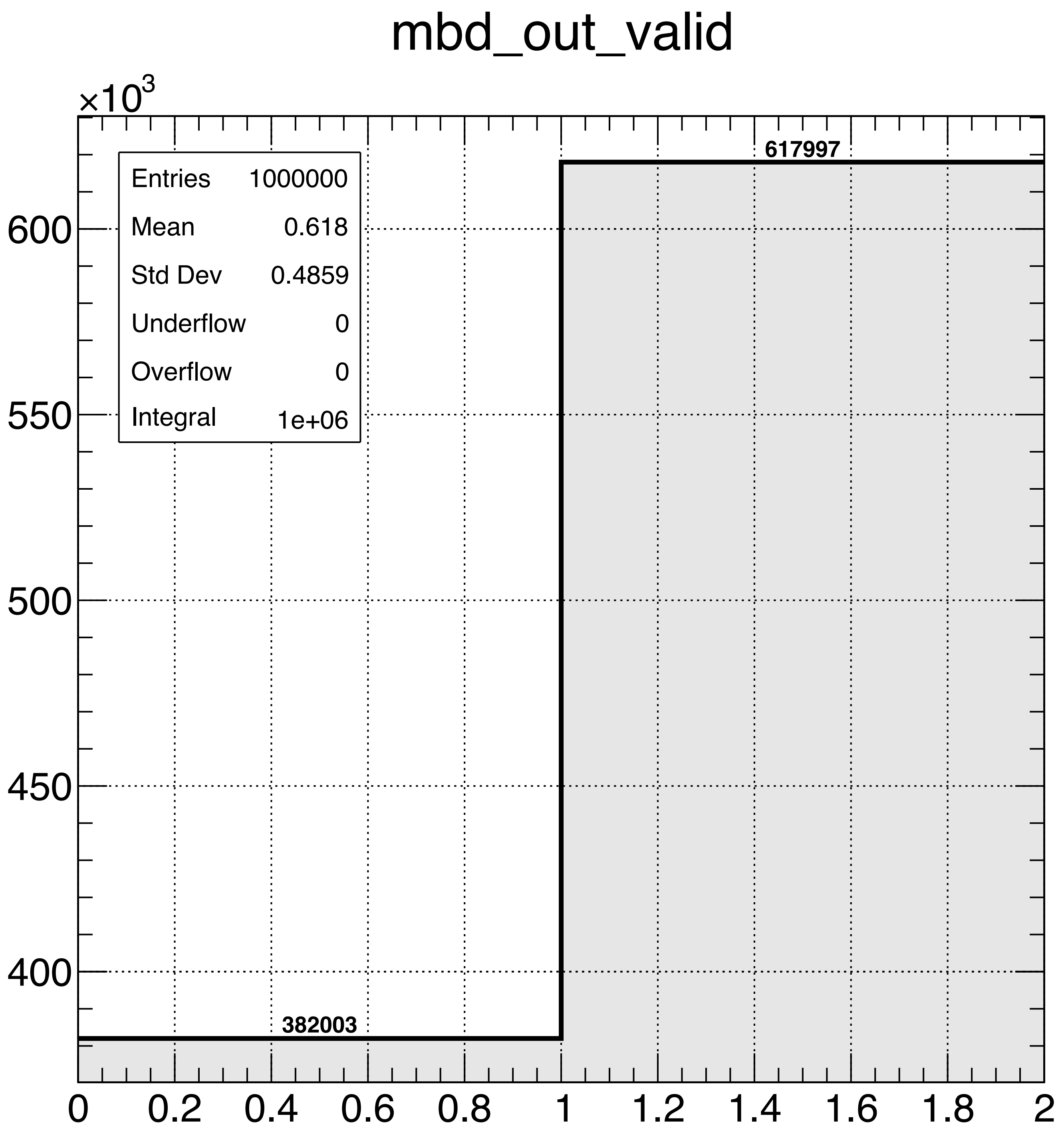
	MbdOutV1 ()
	~MbdOutV1 () override
virtual void	Reset () override Clear Event from memory.
void	identify (std::ostream &os=std::cout) const override
int	isValid () const override isValid returns non zero if object contains vailid data
Float_t	get_zvtx () const override get ZVertex determined by Mbd
Float_t	get_zvtxerr () const override get Error on ZVertex determined by Mbd
Float_t	get_t0 () const override get T0 determined by Mbd
Float_t	get_t0err () const override get Error on T0 determined by Mbd
void	set_t0 (const Float_t t0, const Float_t t0err=0) override
void	set_zvtx (const Float_t vtx, const Float_t vtxerr=0) override set vertex
void	set_zvtxerr (const Float_t vtxerr) override
void	set_arm (const int iarm, const Short_t npmt, const Float_t chargesum, const Float_t timing) override
Short_t	get_npmt (const int iarm) const override
Float_t	get_q (const int iarm) const override
Float_t	get_time (const int iarm) const override

▸ Public Member Functions inherited from MbdOut

▸ Public Member Functions inherited from PHObject

MBD data analysis:

MbdOut

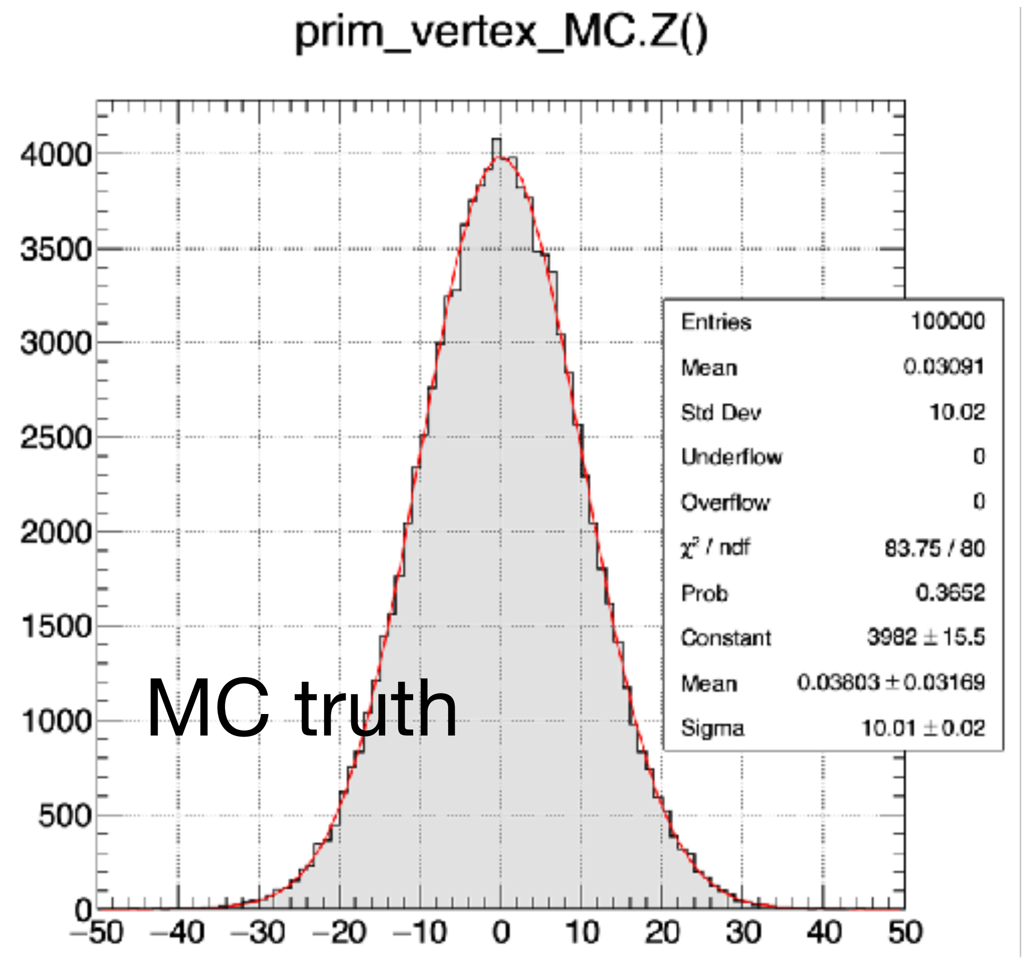


#valid events for MbdOut

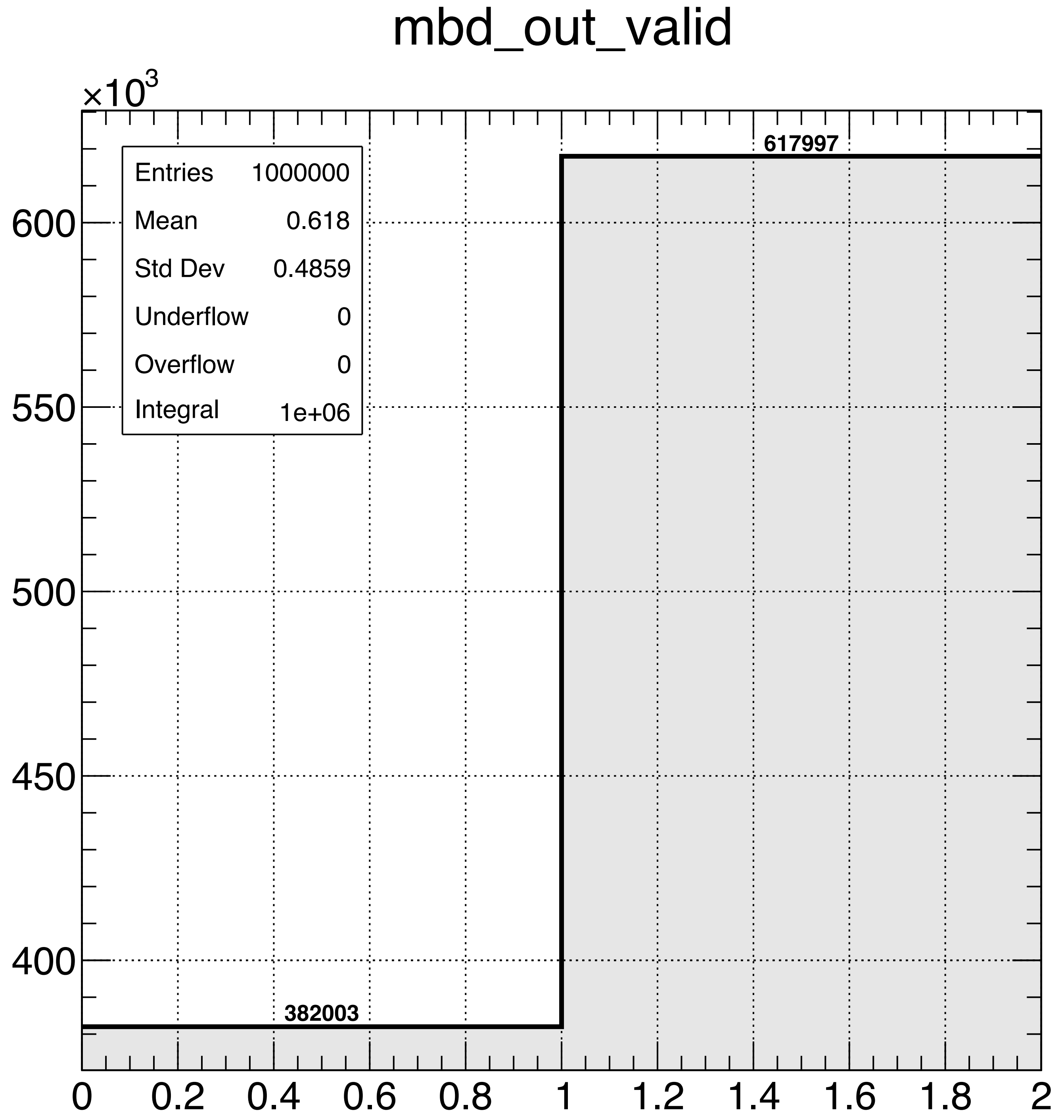
If t_0 can be defined and larger than -9999, the event is valid.

MBD data analysis:

MbdOut

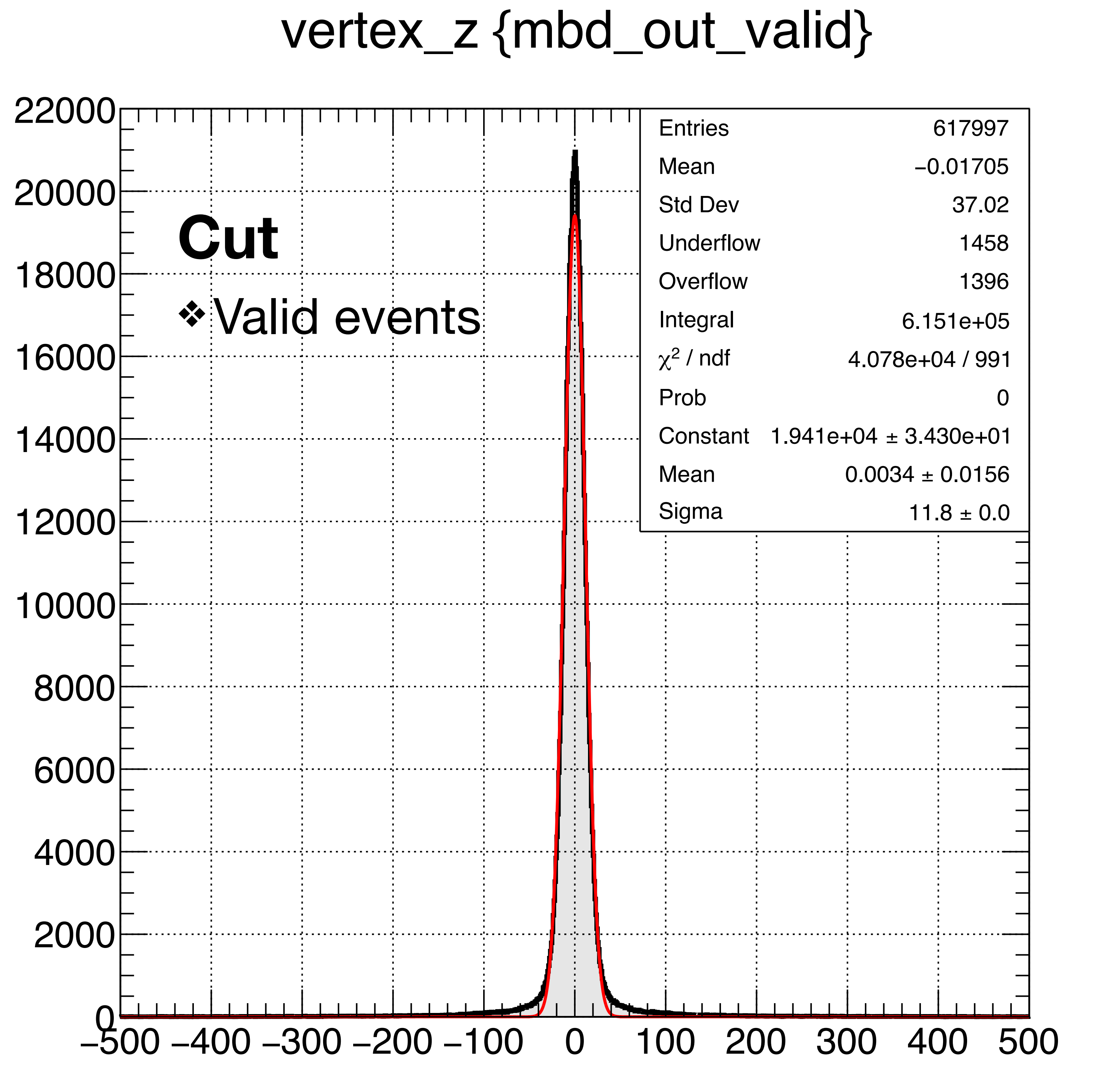


MC truth



#valid events for MbdOut

If t_0 can be defined and larger than -9999, the event is valid.



Cut

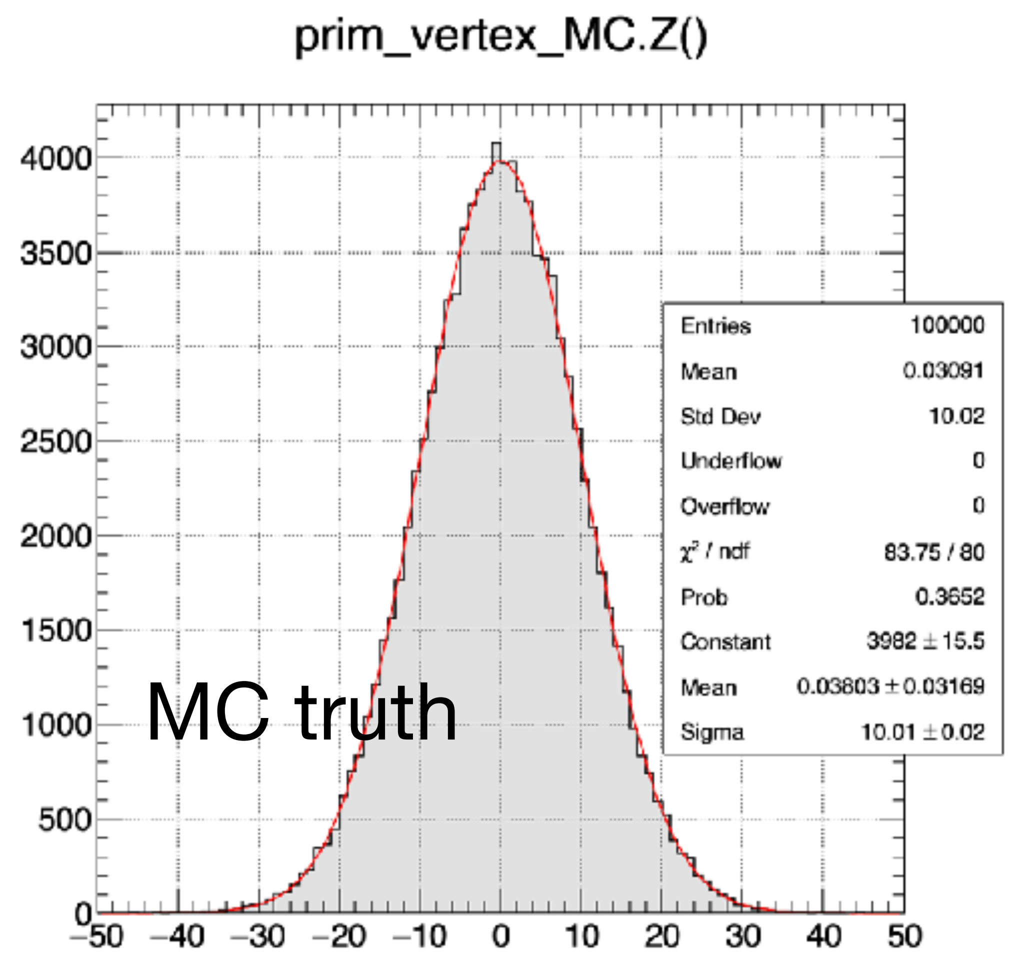
❖Valid events

z_{vtx} distribution using valid events.

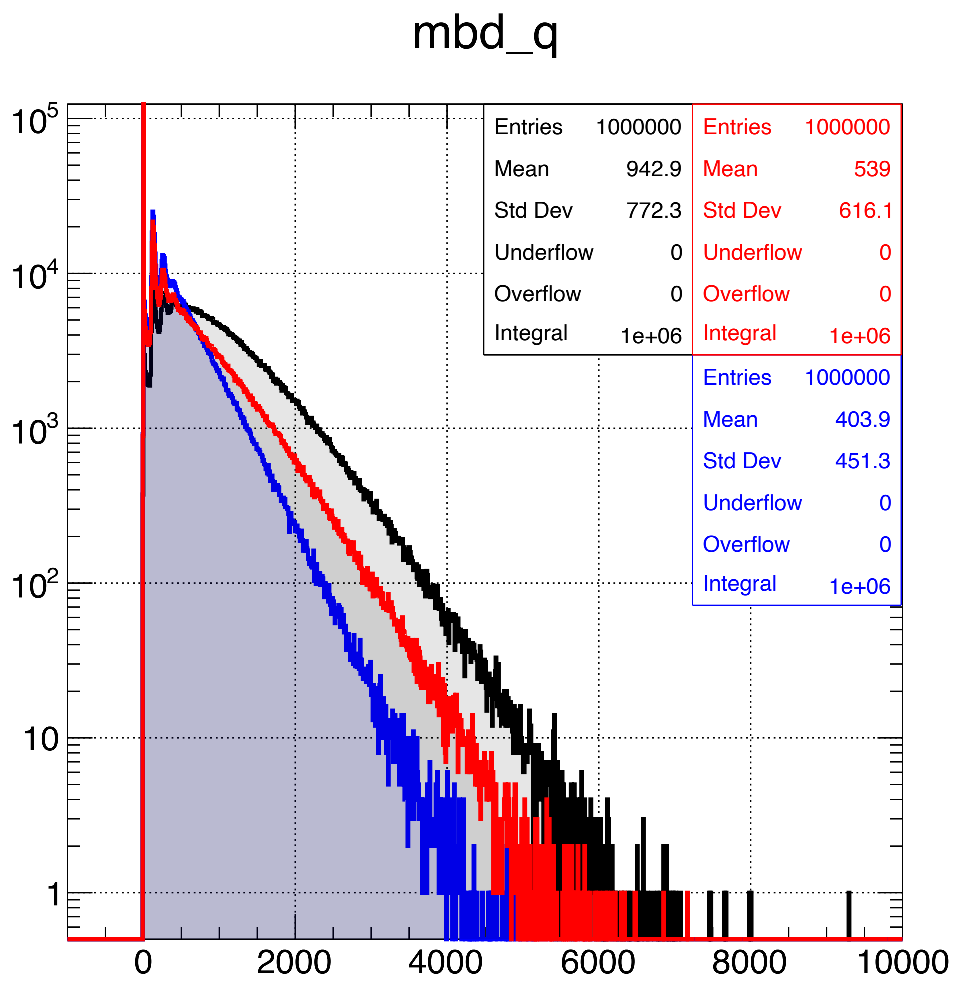
The unit of the x-axis is maybe cm.

MBD data analysis:

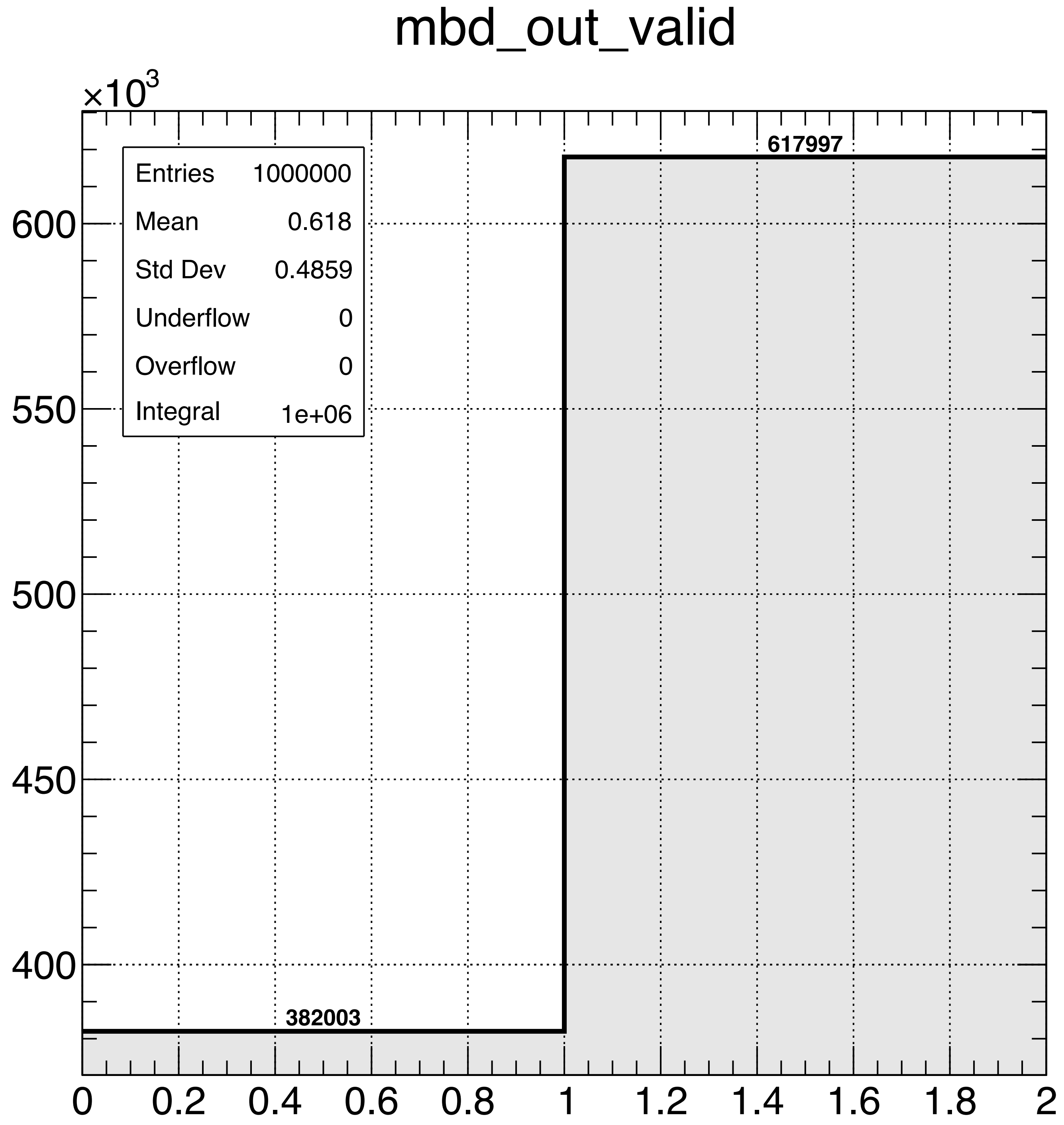
MbdOut

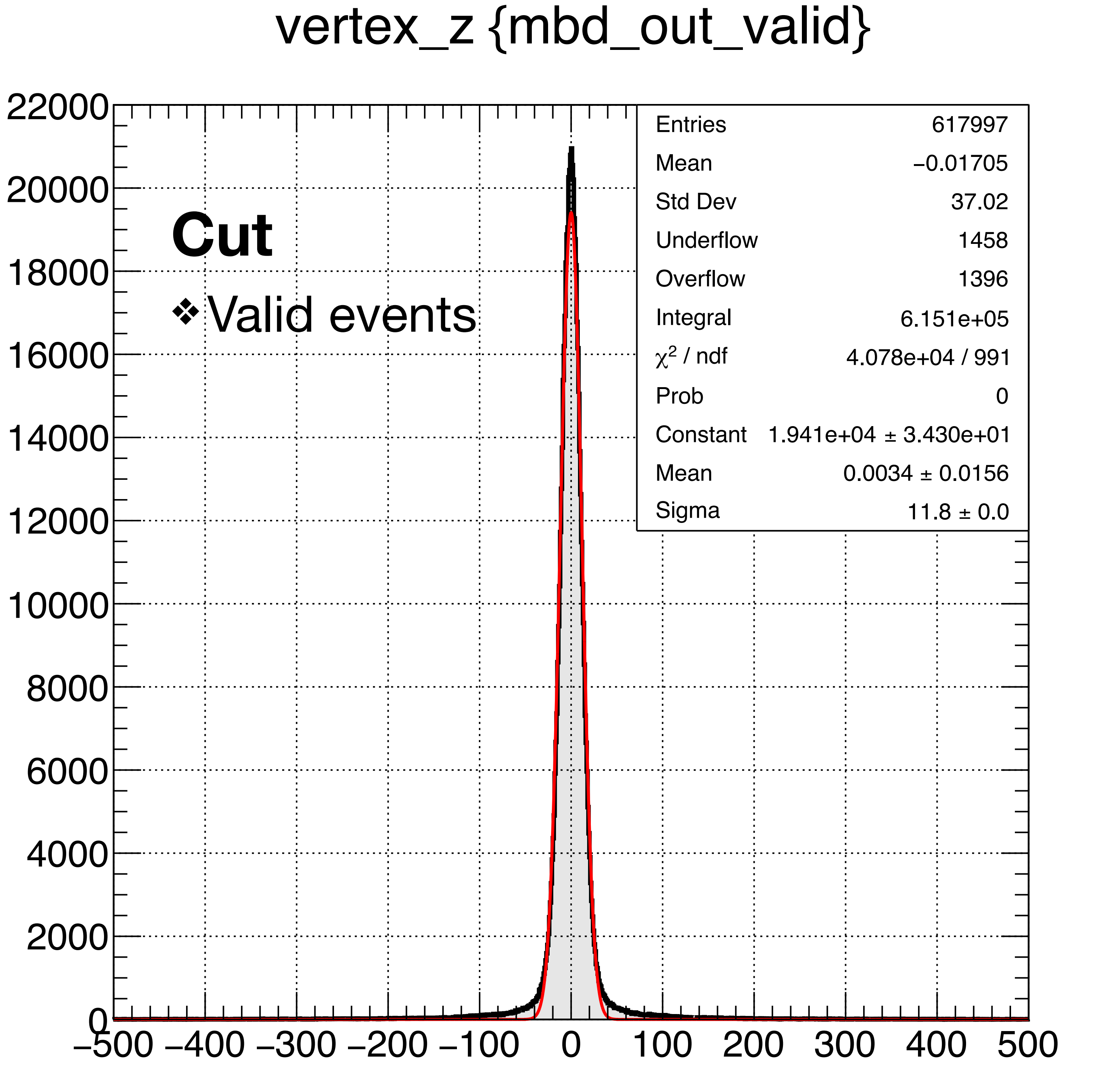


MC truth



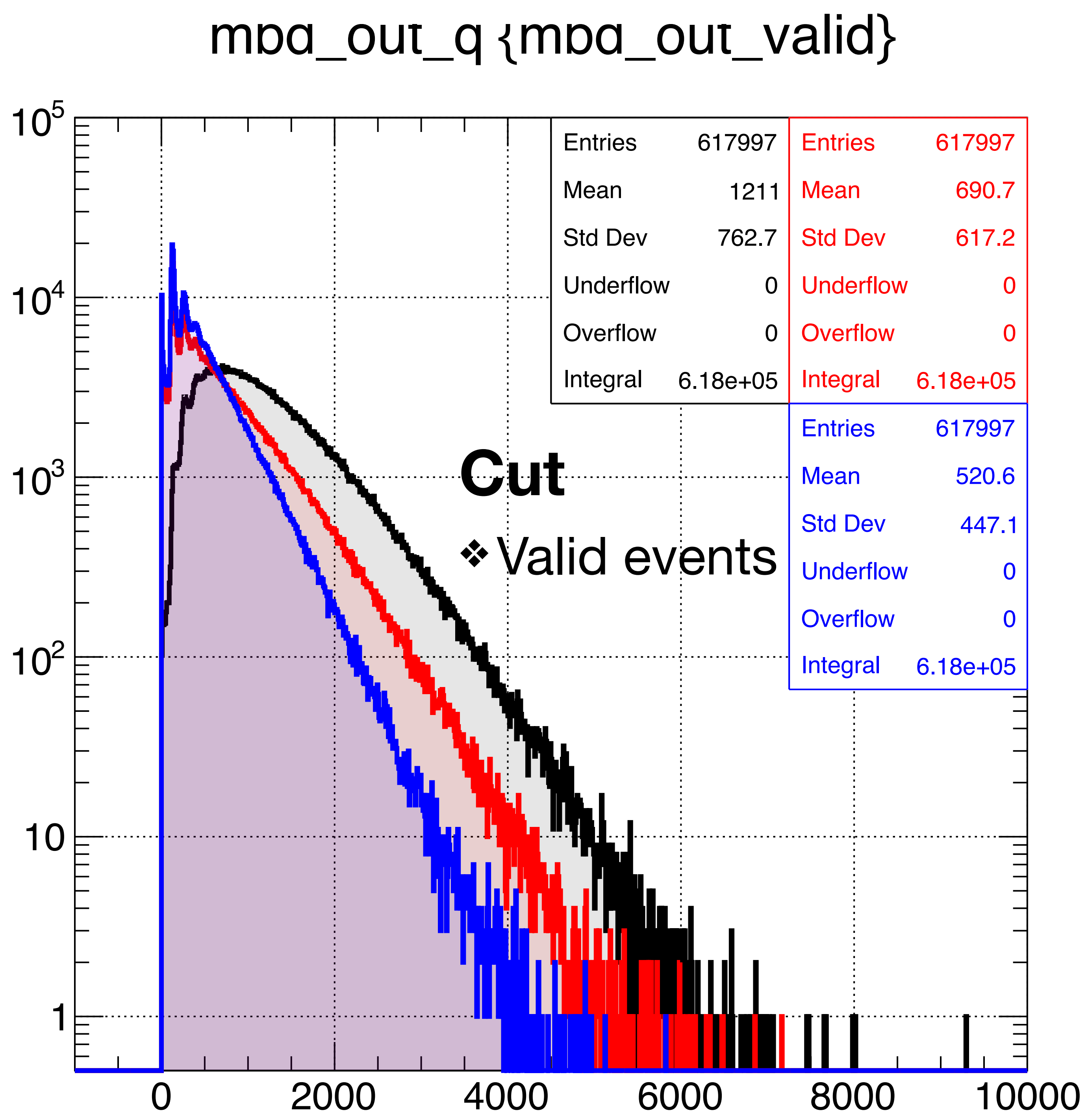
- all
- North
- South





Cut

❖ Valid events



Cut

❖ Valid events

#valid events for MbdOut

If t_0 can be defined and larger than -9999, the event is valid.

z_{vtx} distribution using valid events.

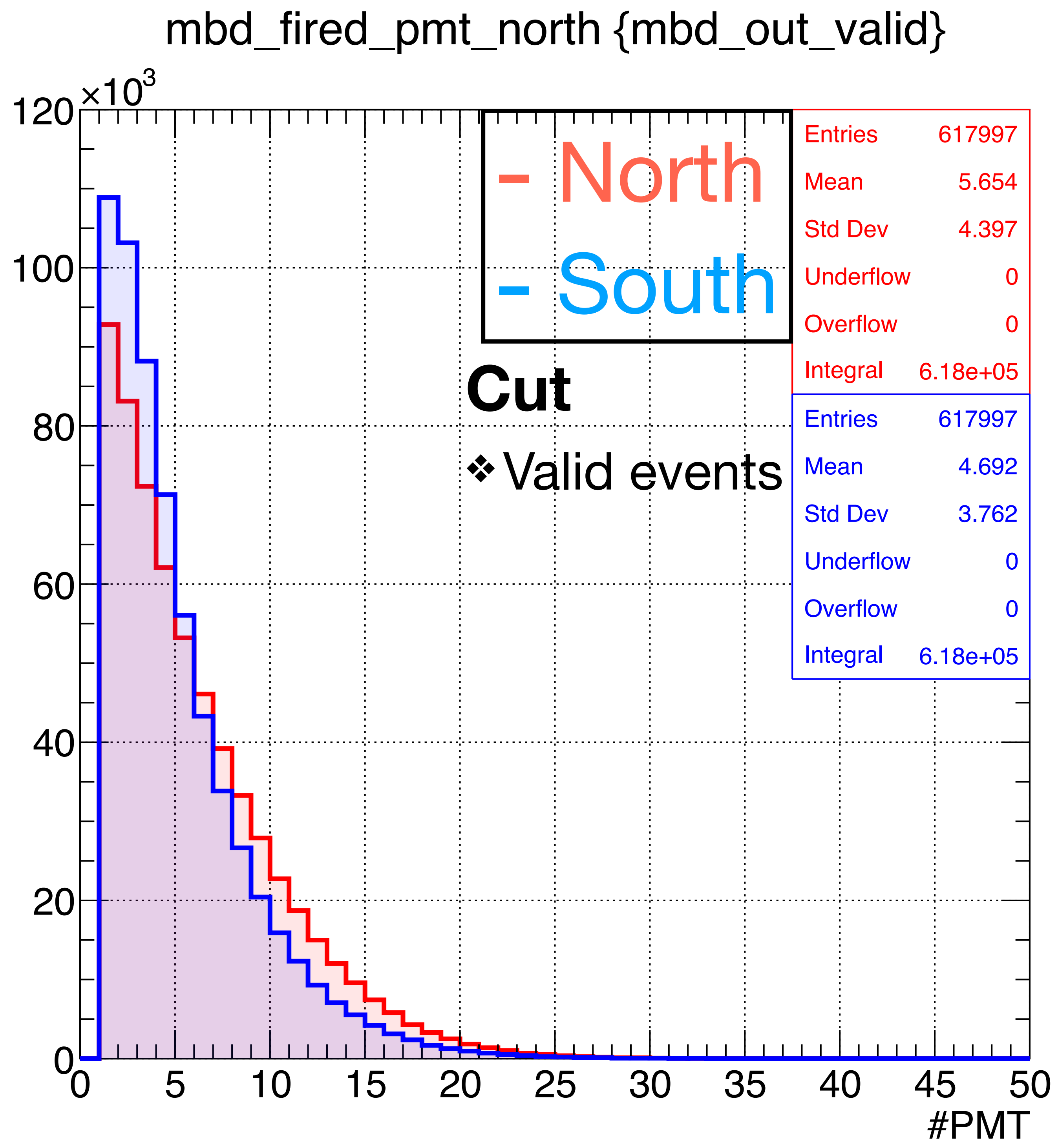
The unit of the x-axis is maybe cm.

Charge distributions.

These are not same as those made using PMT info.

MBD data analysis:

MbdOut



Cut

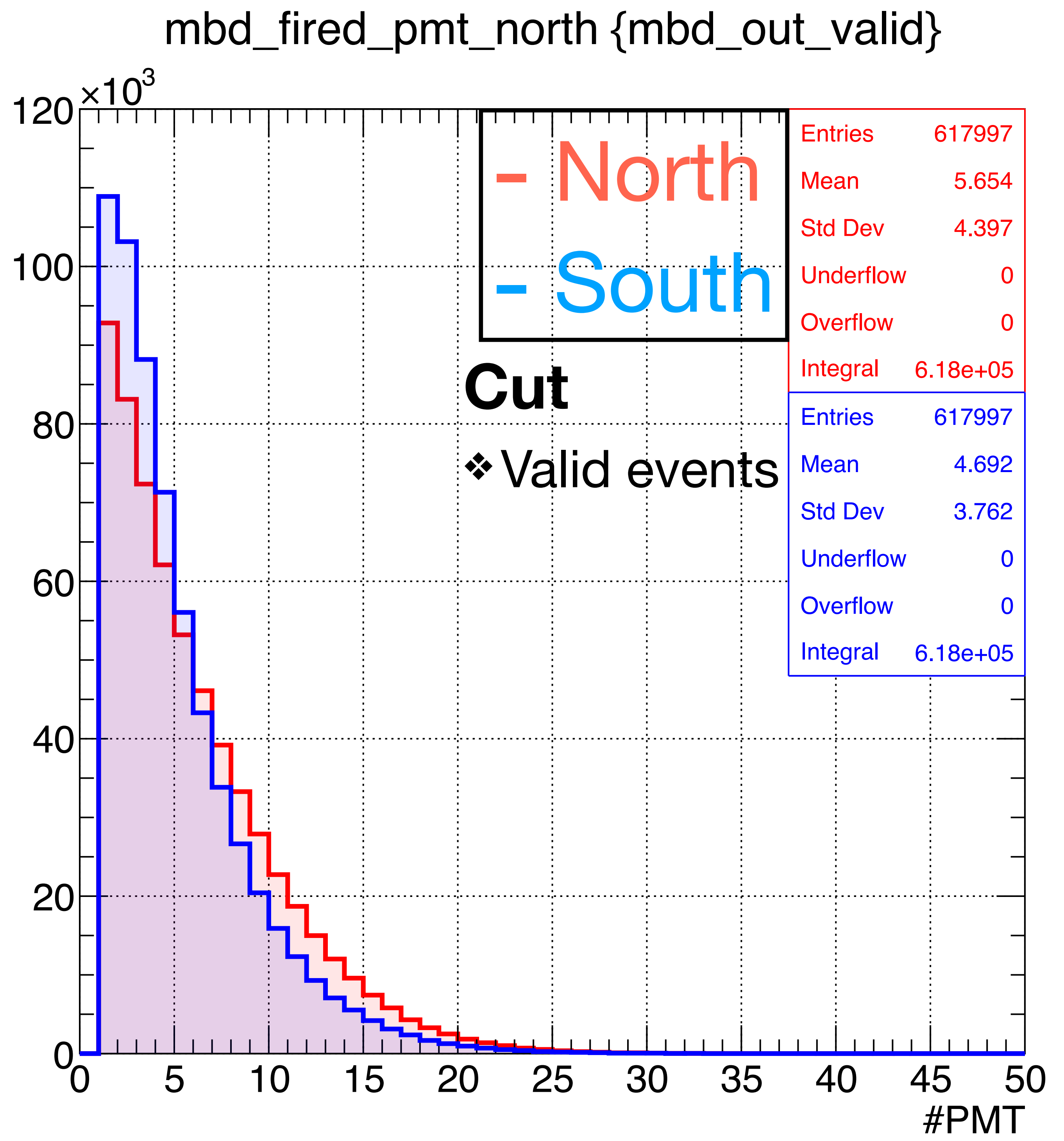
- Valid events

#fired PMT distributions.

What is the condition of “fired”?

MBD data analysis:

MbdOut

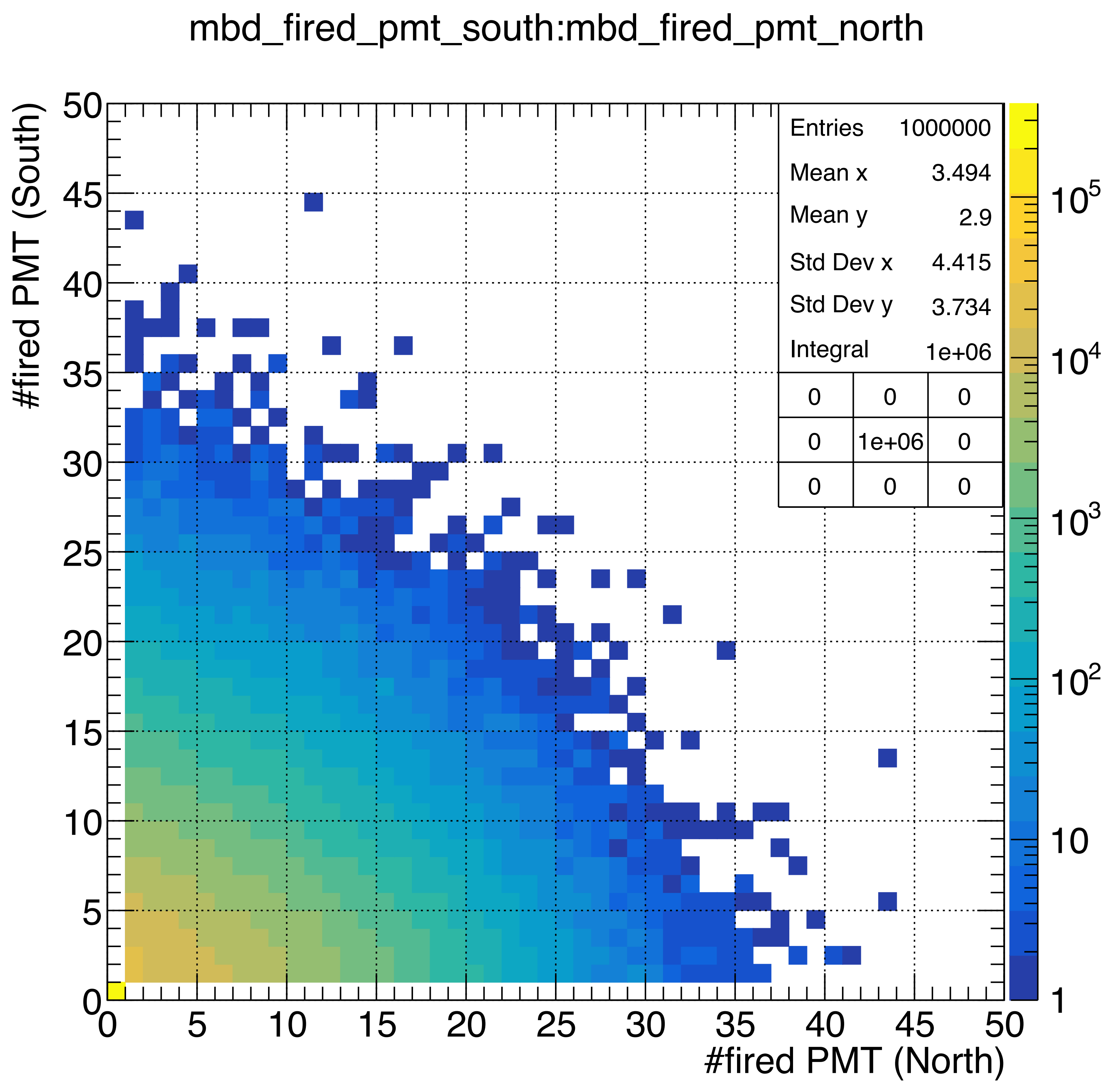


Cut

❖ Valid events

#fired PMT distributions.

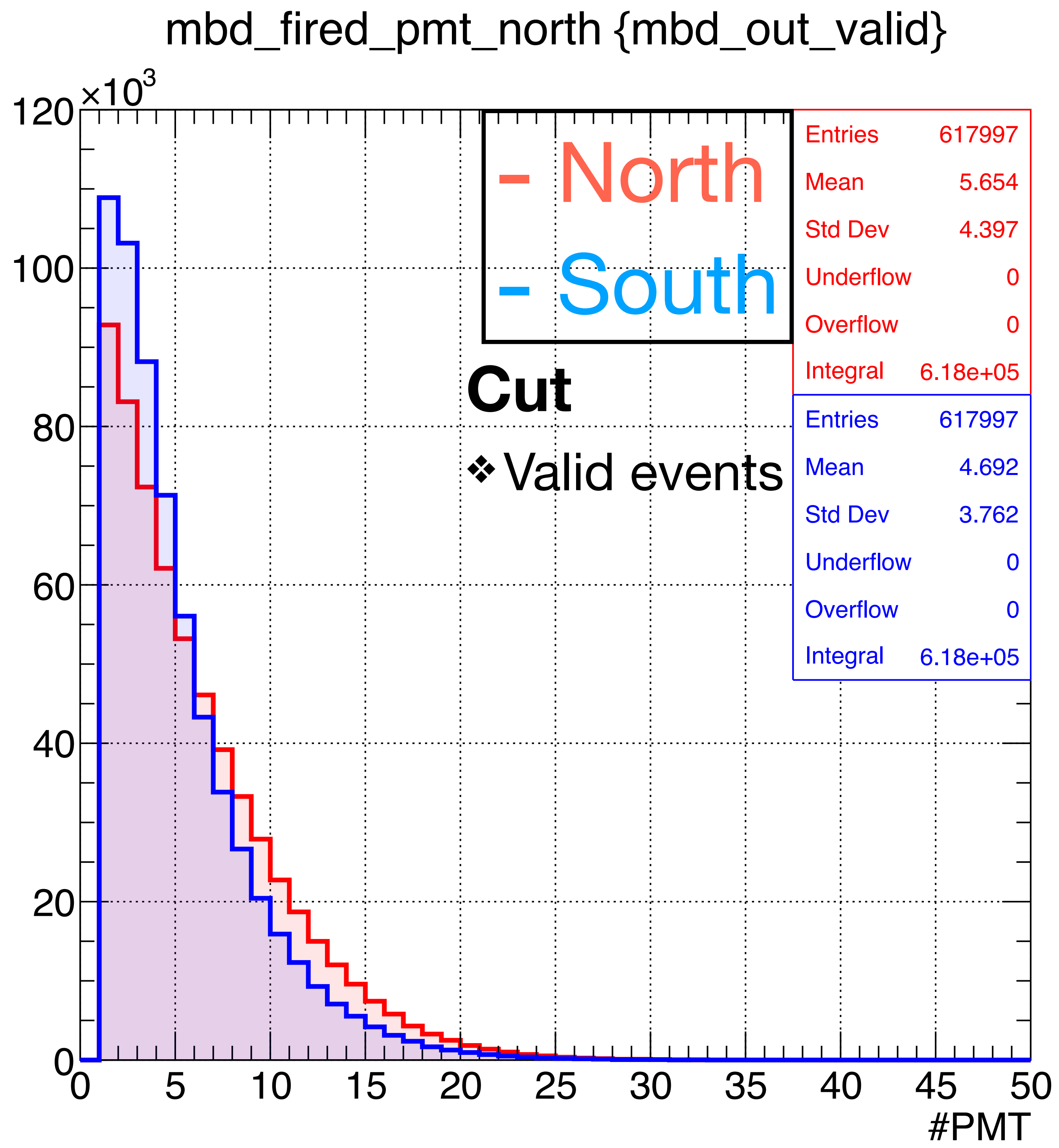
What is the condition of “fired”?



#fired PMT (North vs South) without any cuts.

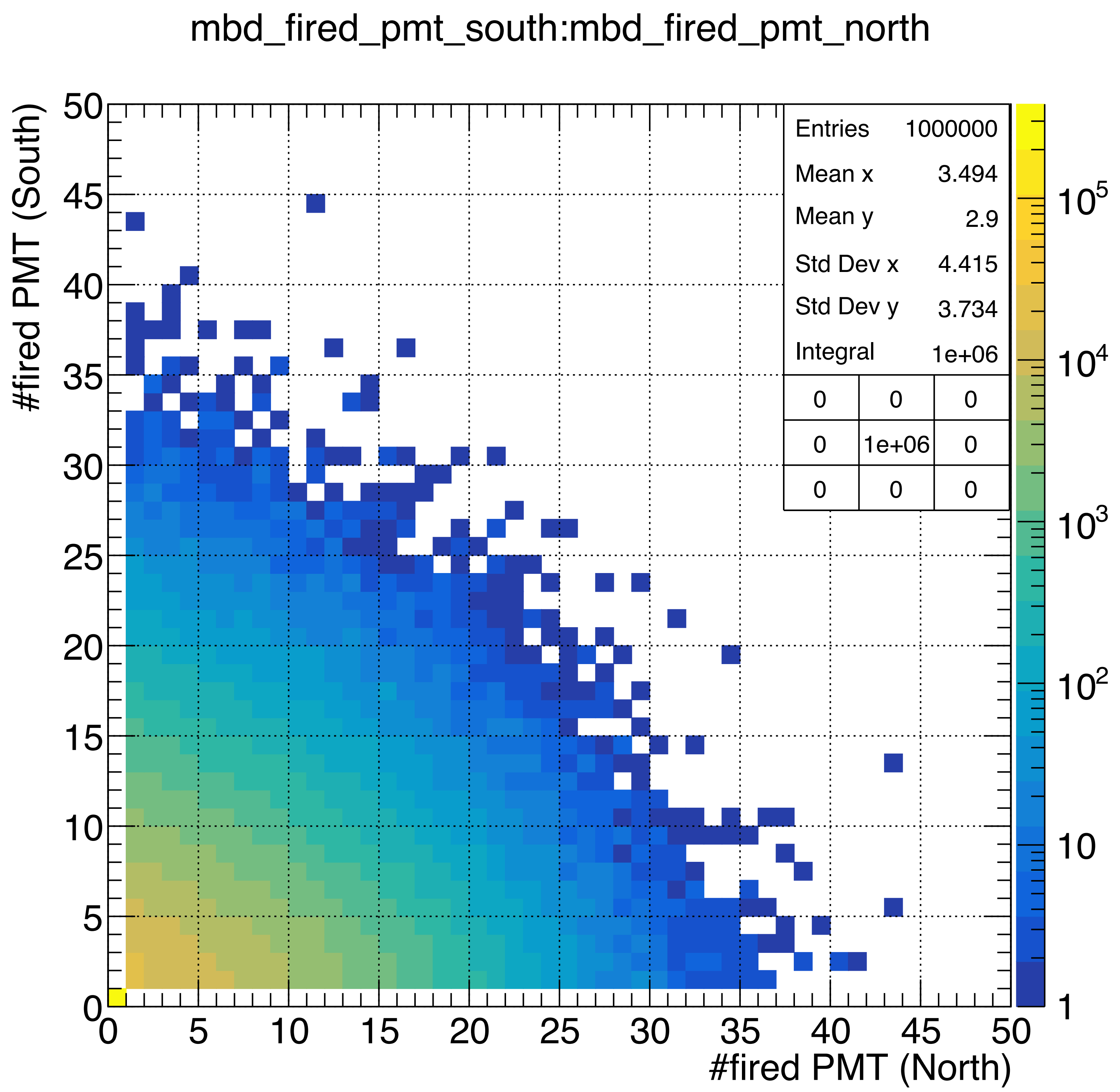
MBD data analysis:

MbdOut

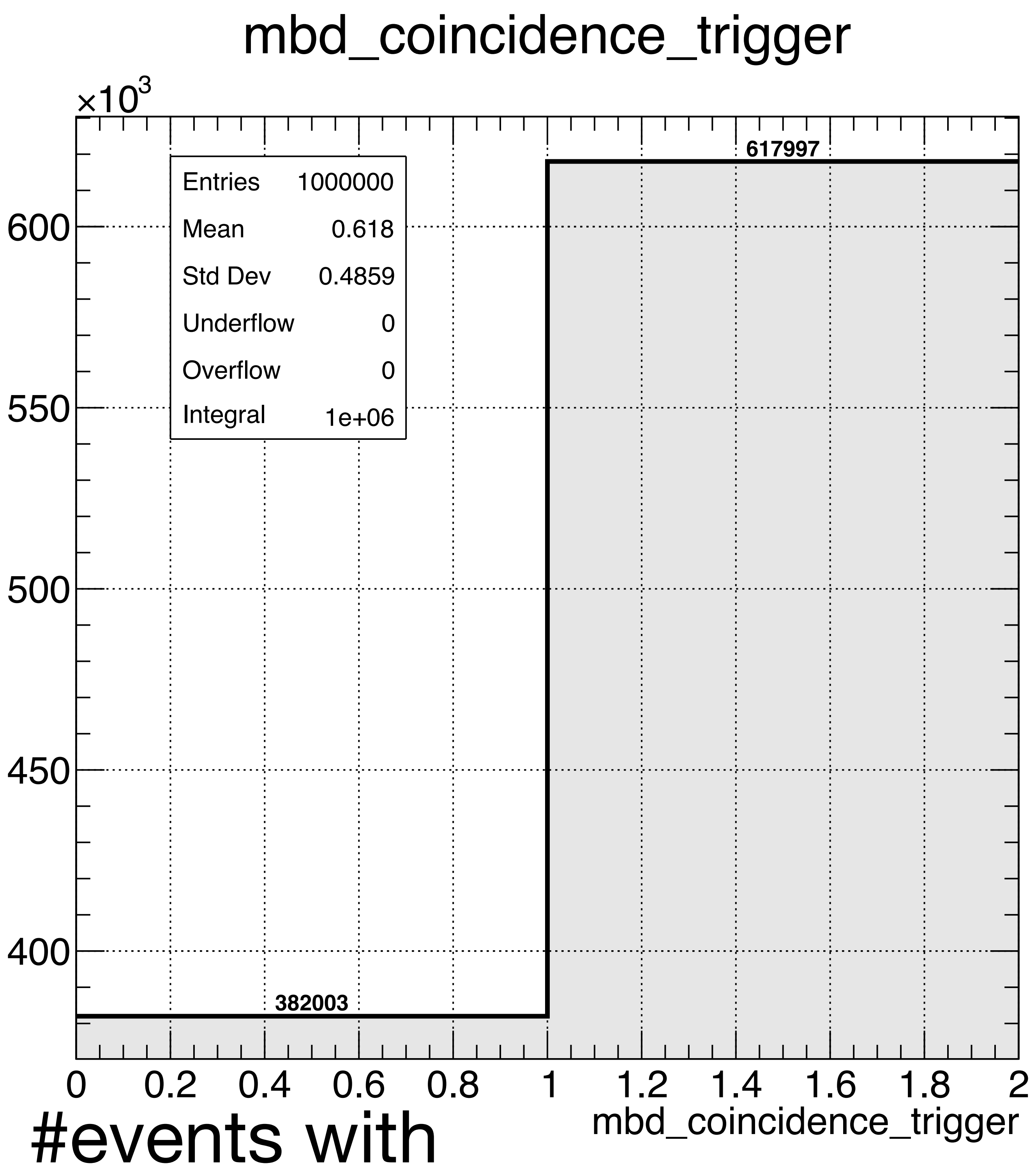


#fired PMT distributions.

What is the condition of “fired”?



#fired PMT (North vs South) without any cuts.



#events with

#fired PMT (north) > 0 AND

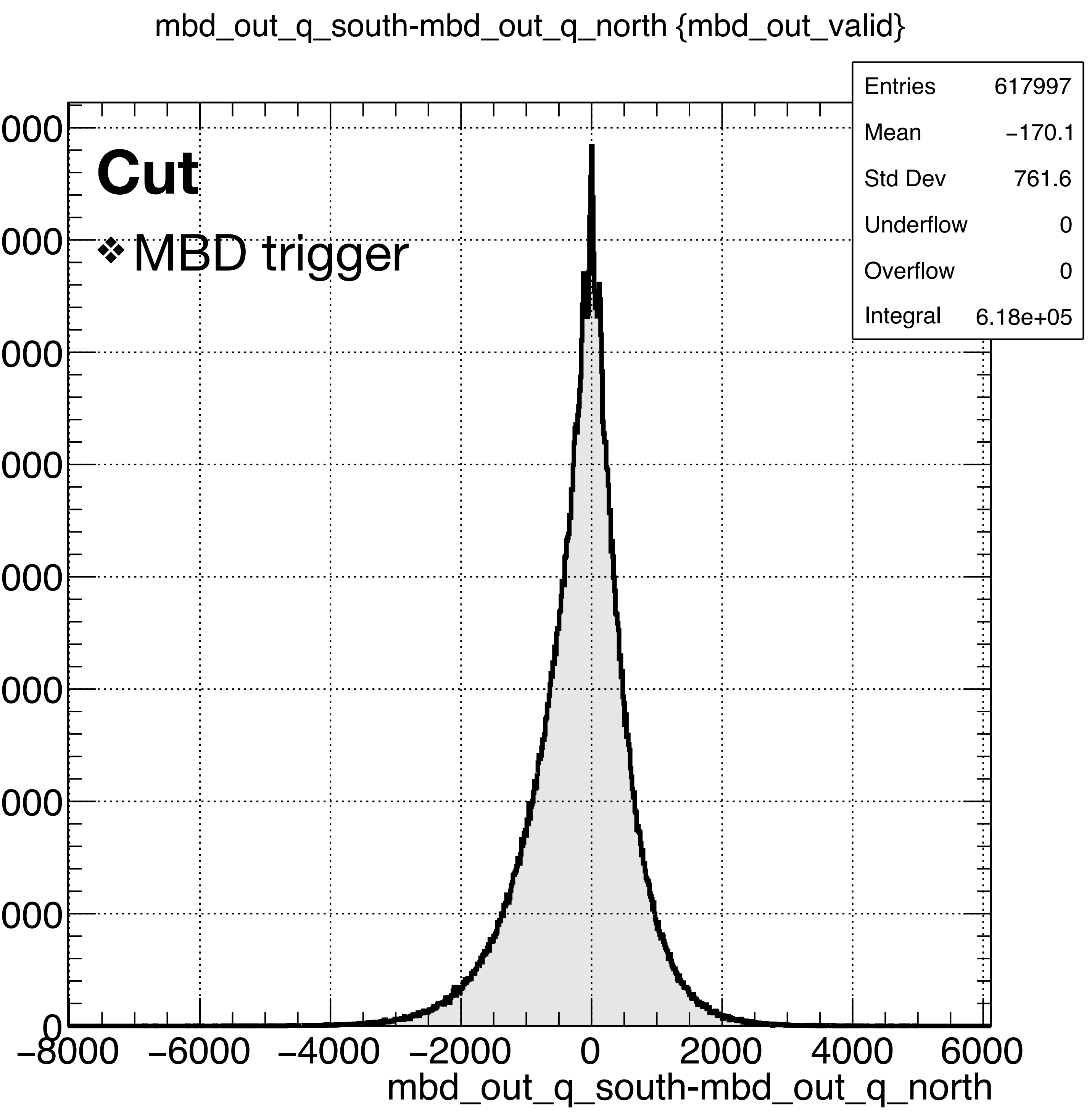
#fired PMT (south) > 0

i.e. equivalent with MBD N&S ≧ 1

~62% of minimum bias events fired MBD N&S ≧ 1 trigger

MBD data analysis:

MbdOut



Cut

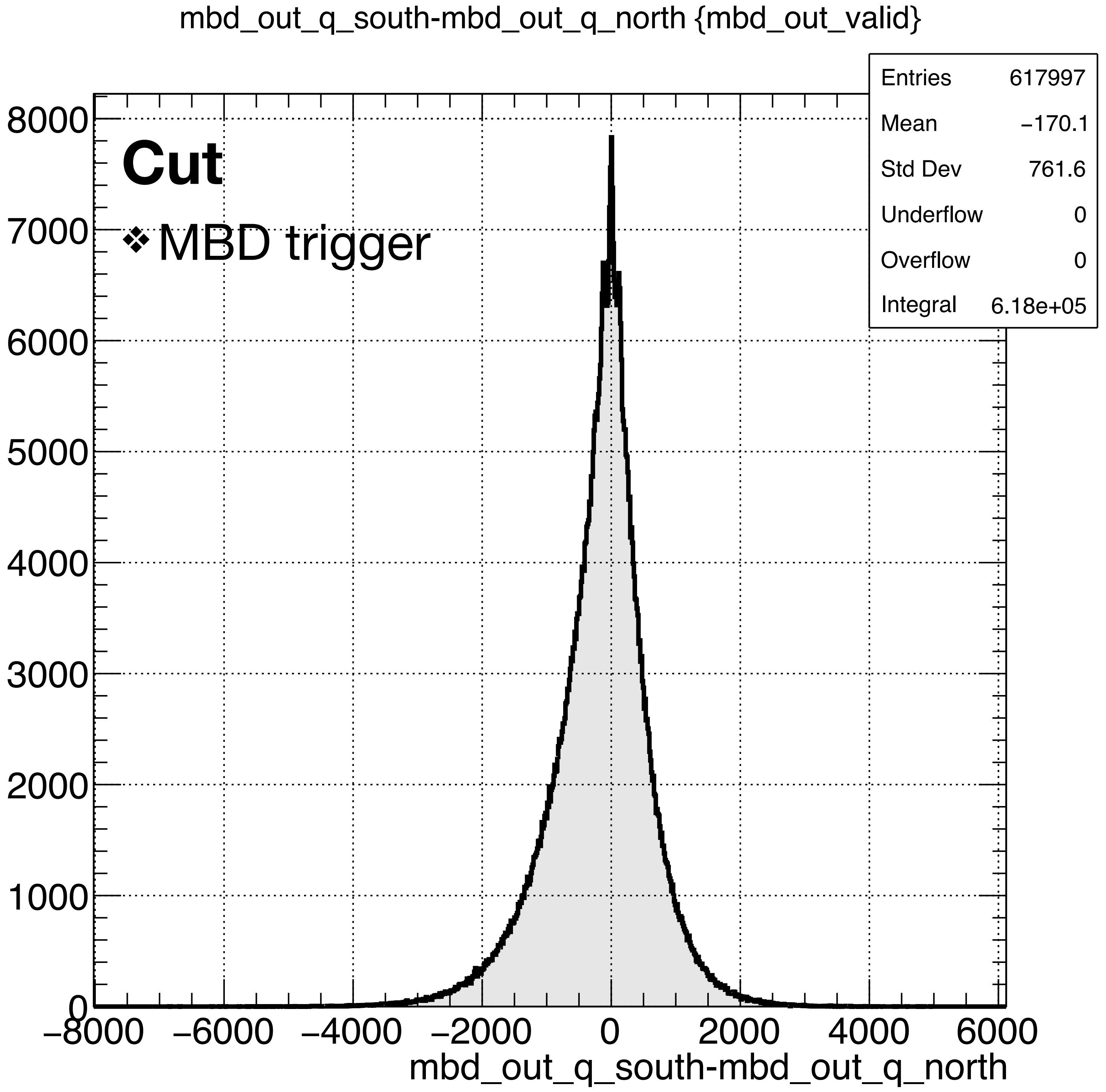
- ❖ MBD trigger

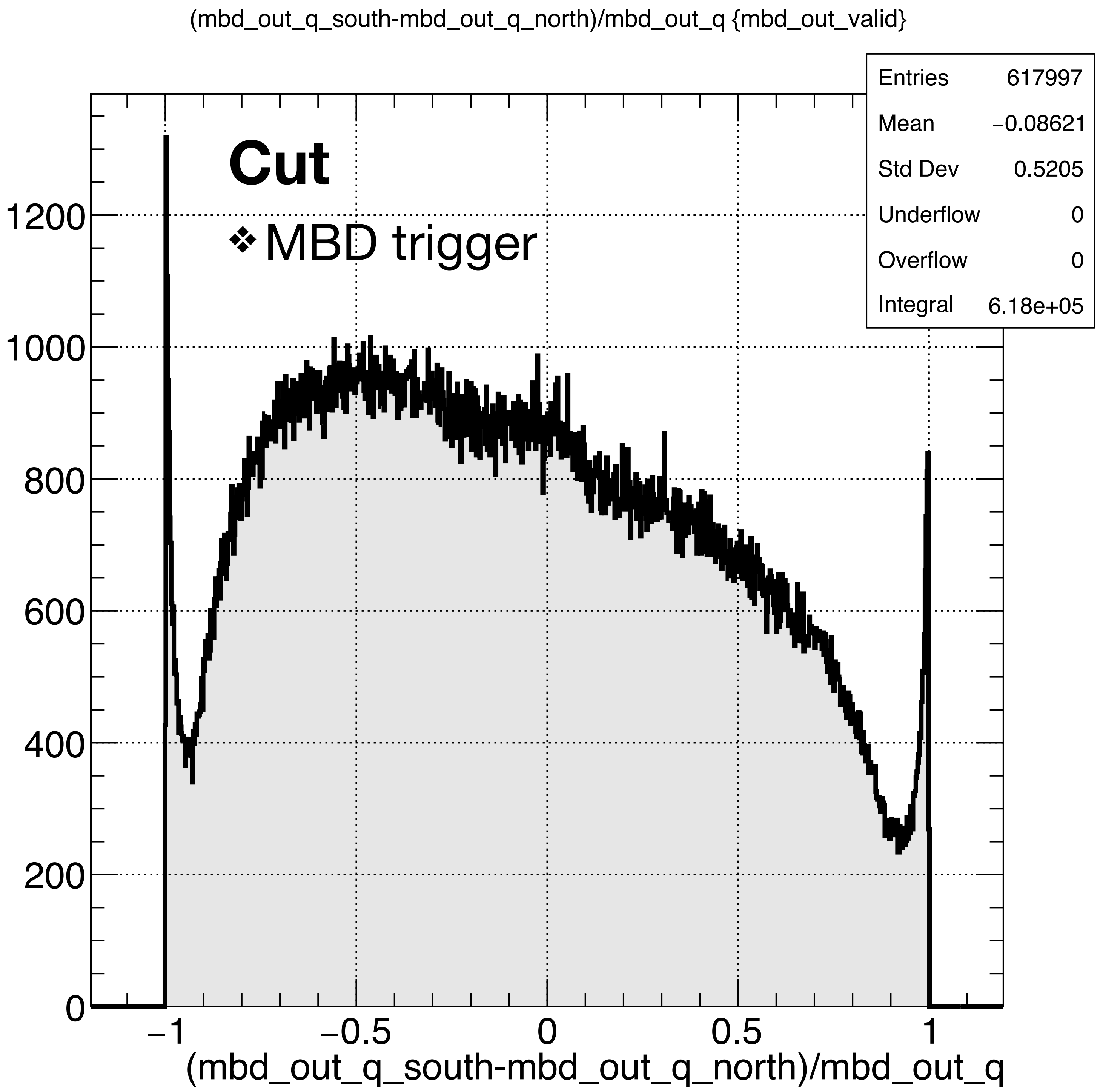
q_{south} - q_{north}

Why is the shape asymmetric?

MBD data analysis:

MbdOut





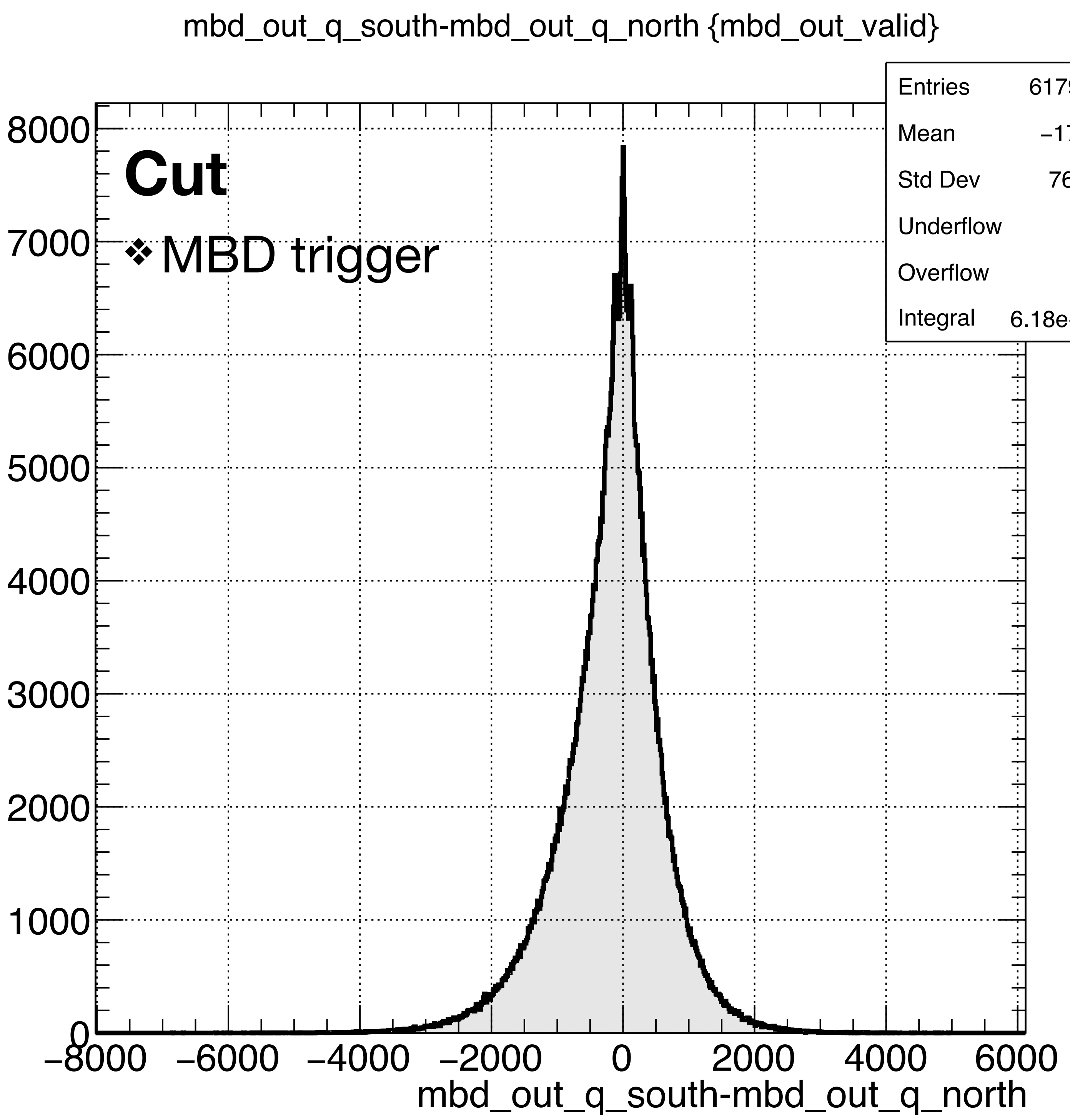
q_{south} - q_{north}

Why is the shape asymmetric?

$(q_{south} - q_{north}) / (q_{south} + q_{north})$

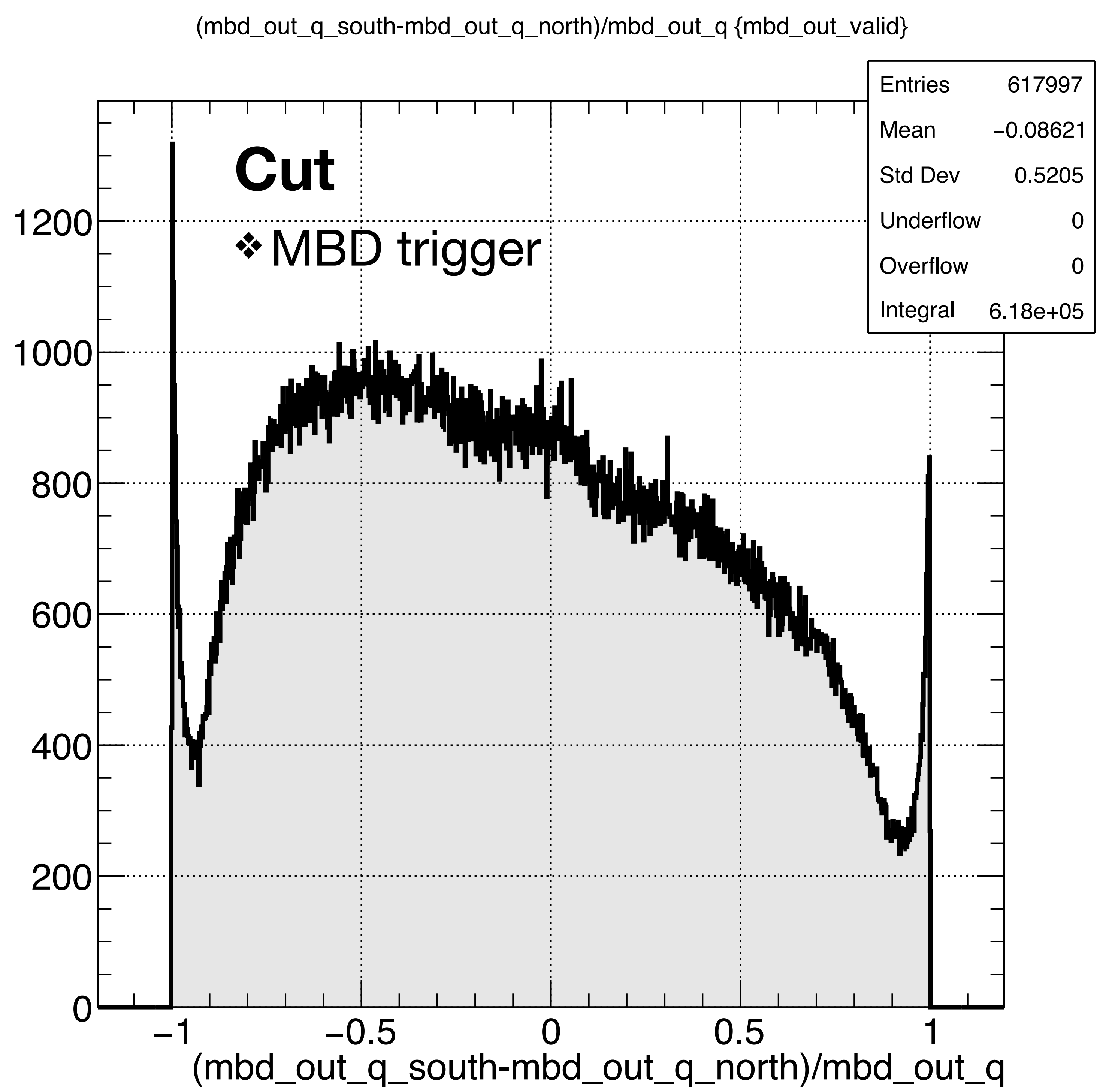
MBD data analysis:

MbdOut

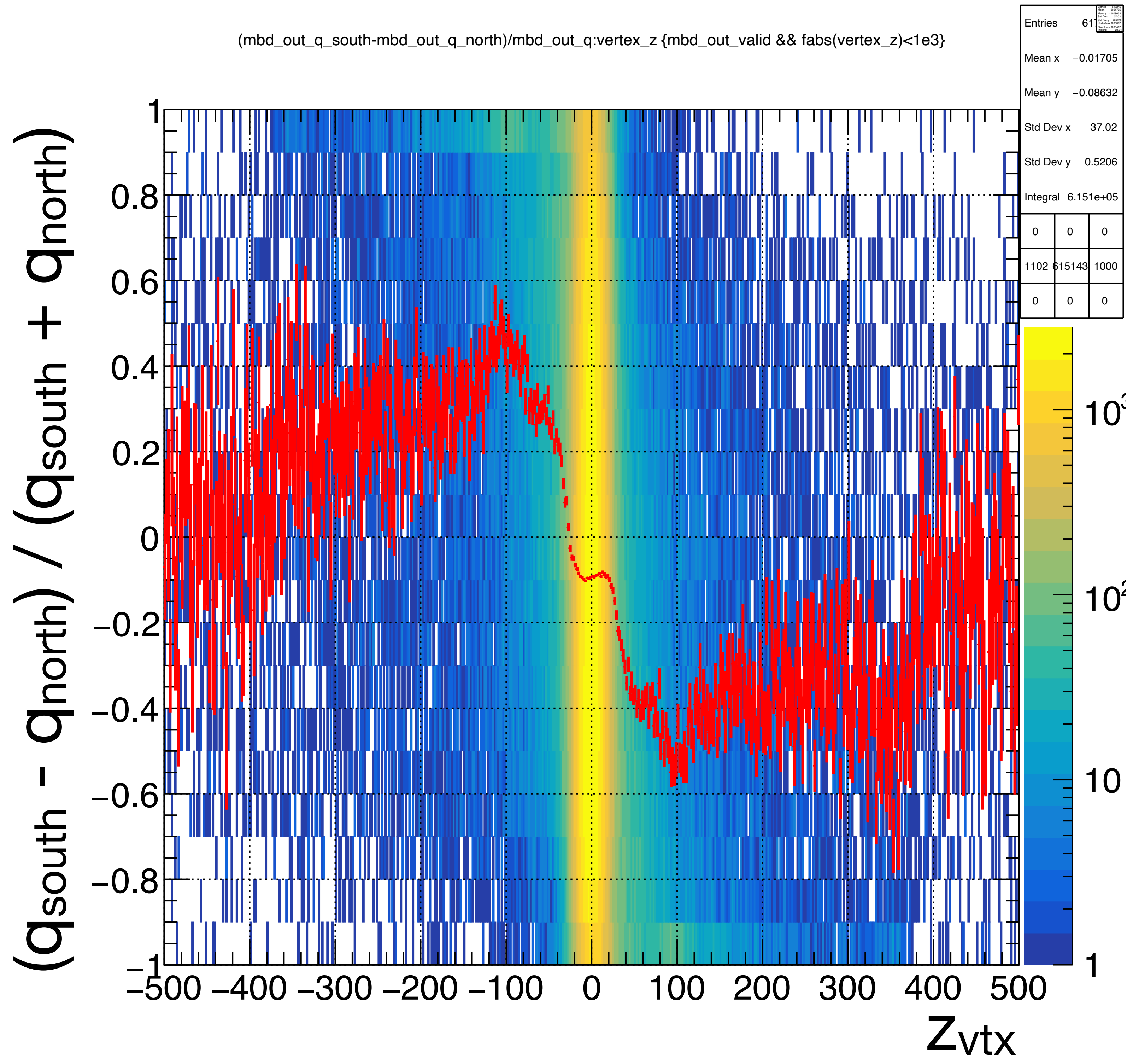


$q_{south} - q_{north}$

Why is the shape asymmetric?



$(q_{south} - q_{north}) / (q_{south} + q_{north})$

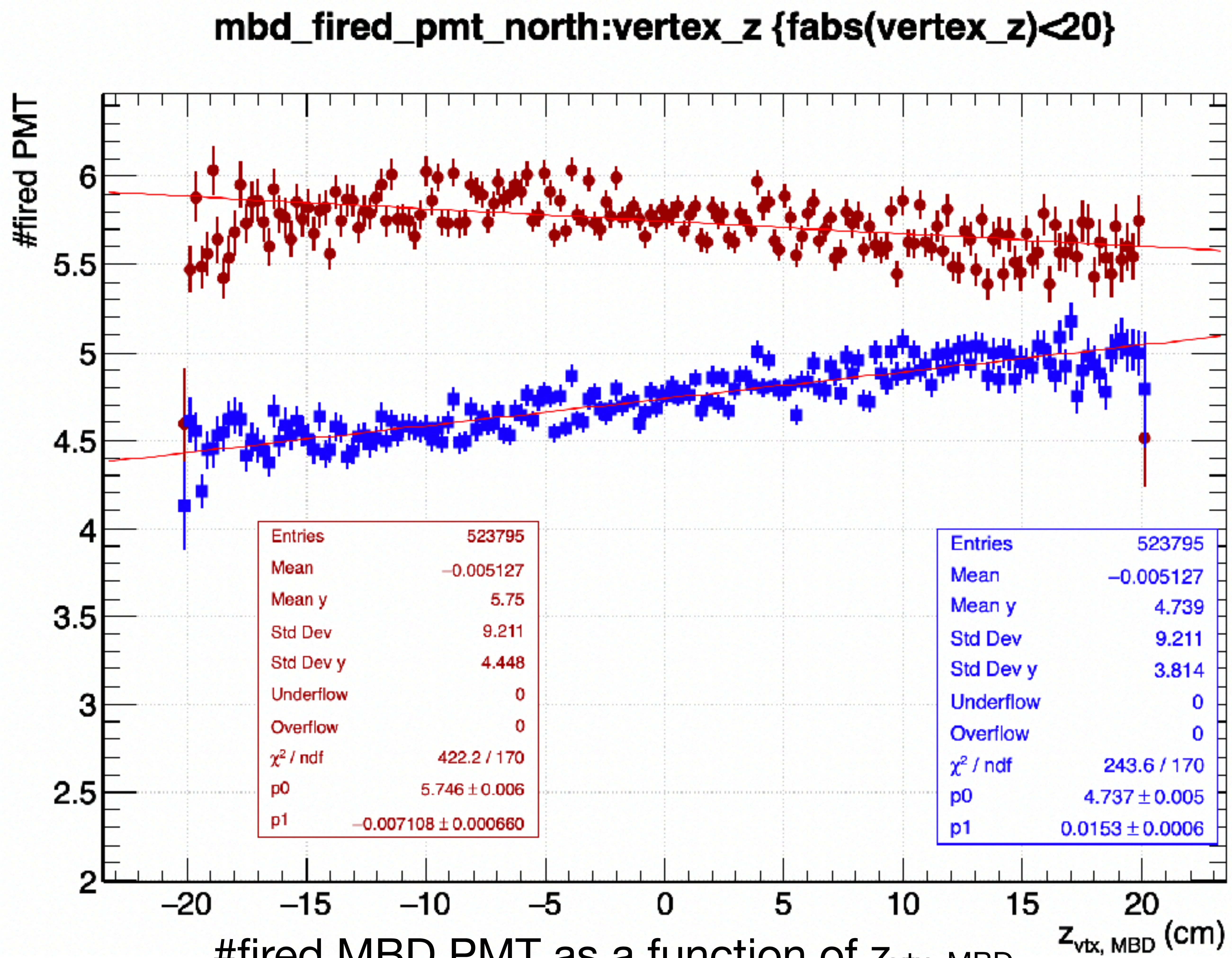


z_{vtx} vs $(q_{south} - q_{north}) / (q_{south} + q_{north})$

The asymmetry depends on z_{vtx}.

MBD data analysis:

MbdOut



Entries	523795
Mean	−0.005127
Mean y	5.75
Std Dev	9.211
Std Dev y	4.448
Underflow	0
Overflow	0
χ^2 / ndf	422.2 / 170
p0	5.746 ± 0.006
p1	−0.007108 ± 0.000660

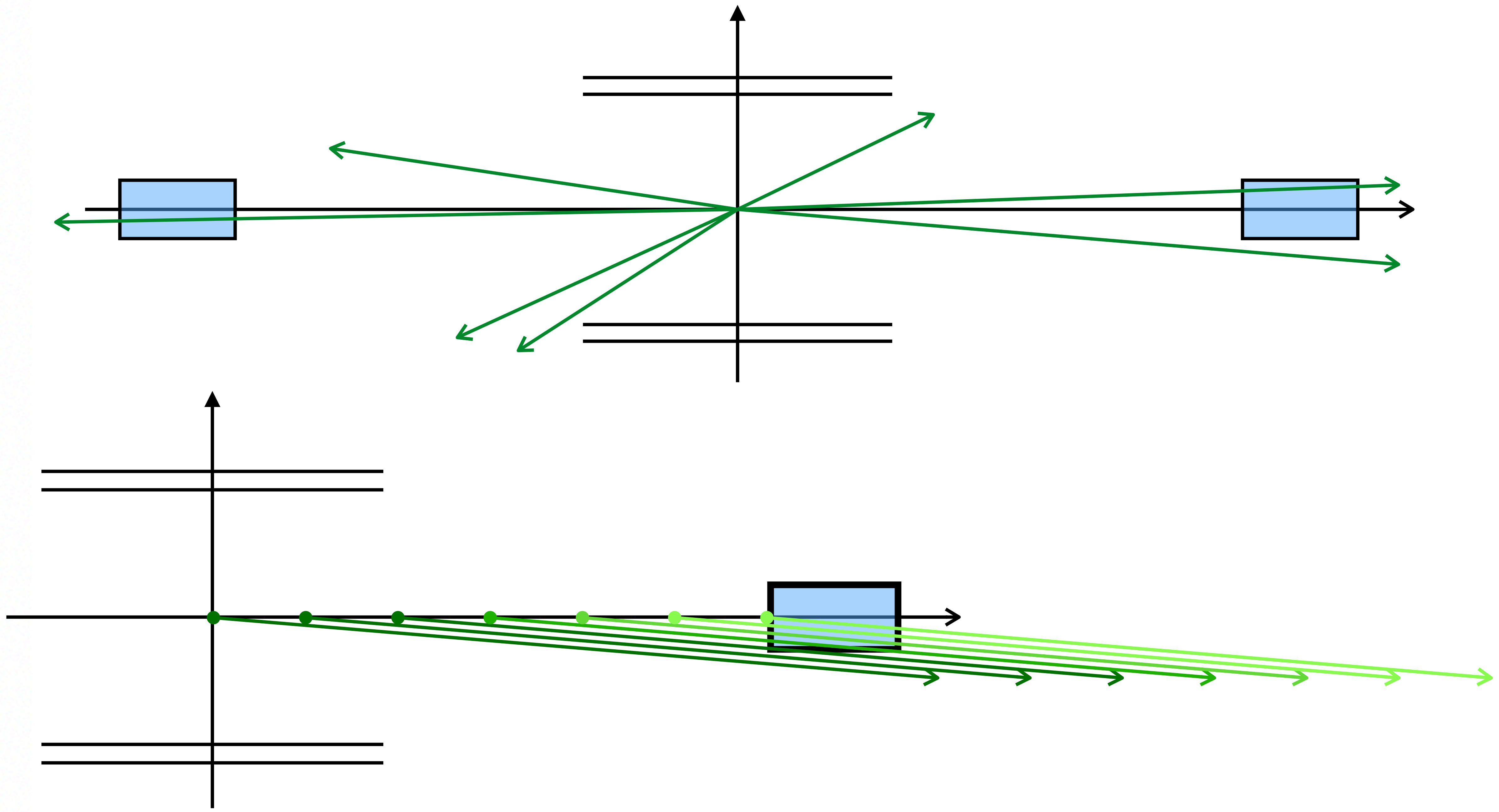
Entries	523795
Mean	−0.005127
Mean y	4.739
Std Dev	9.211
Std Dev y	3.814
Underflow	0
Overflow	0
χ^2 / ndf	243.6 / 170
p0	4.737 ± 0.005
p1	0.0153 ± 0.0006

#fired MBD PMT as a function of $z_{vtx,\ MBD}$

#fired PMT should depends on z_{vtx} due to acceptance.

Cut

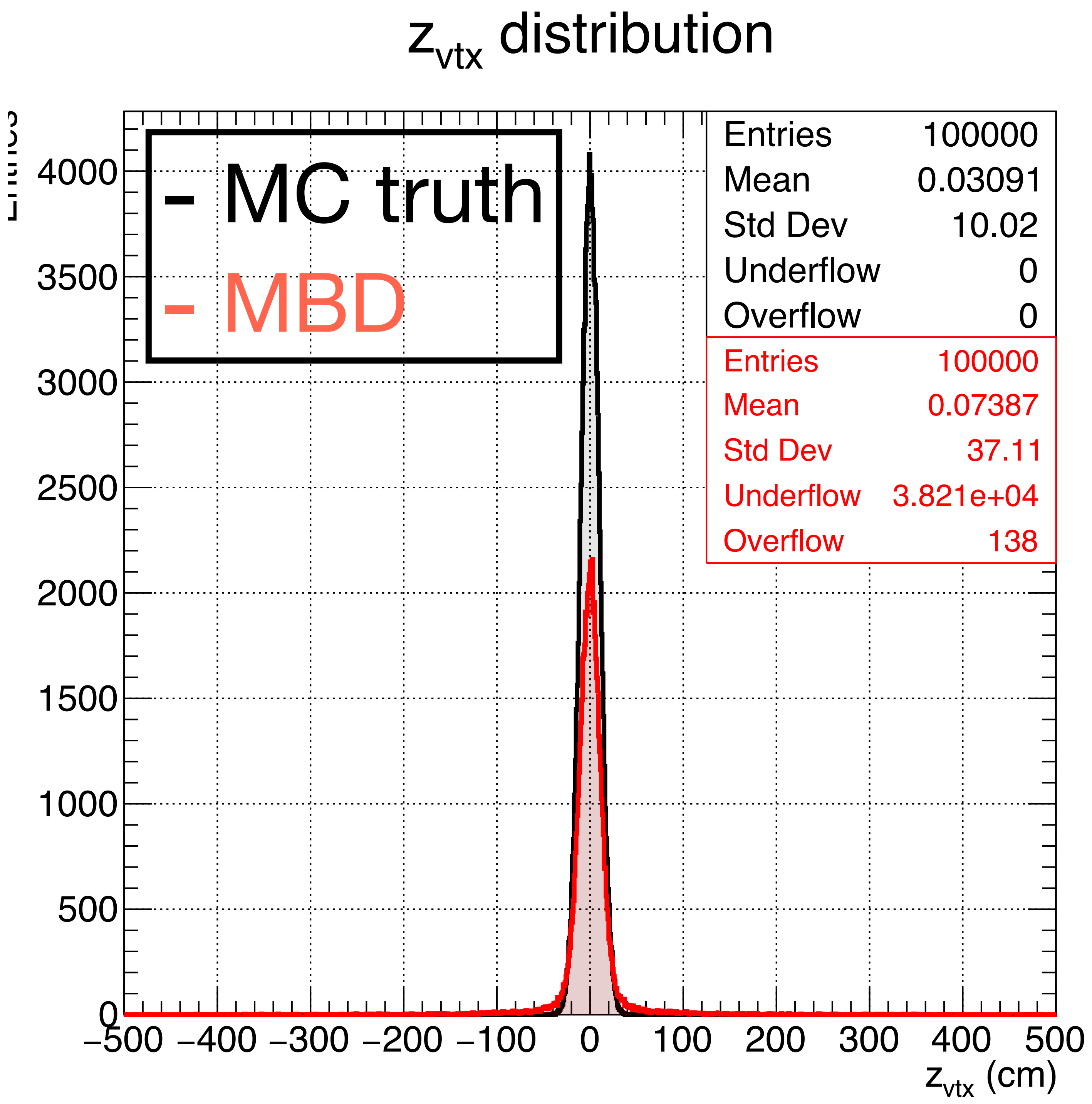
- ❖MBD trigger
- ❖$|z_{vtx,\ MBD}|$ < 20 cm: Successful reconstruction



The acceptance effect can be seen but the absolute value of the slopes is different by ×2.

MBD data analysis:

MbdOut & MC truth

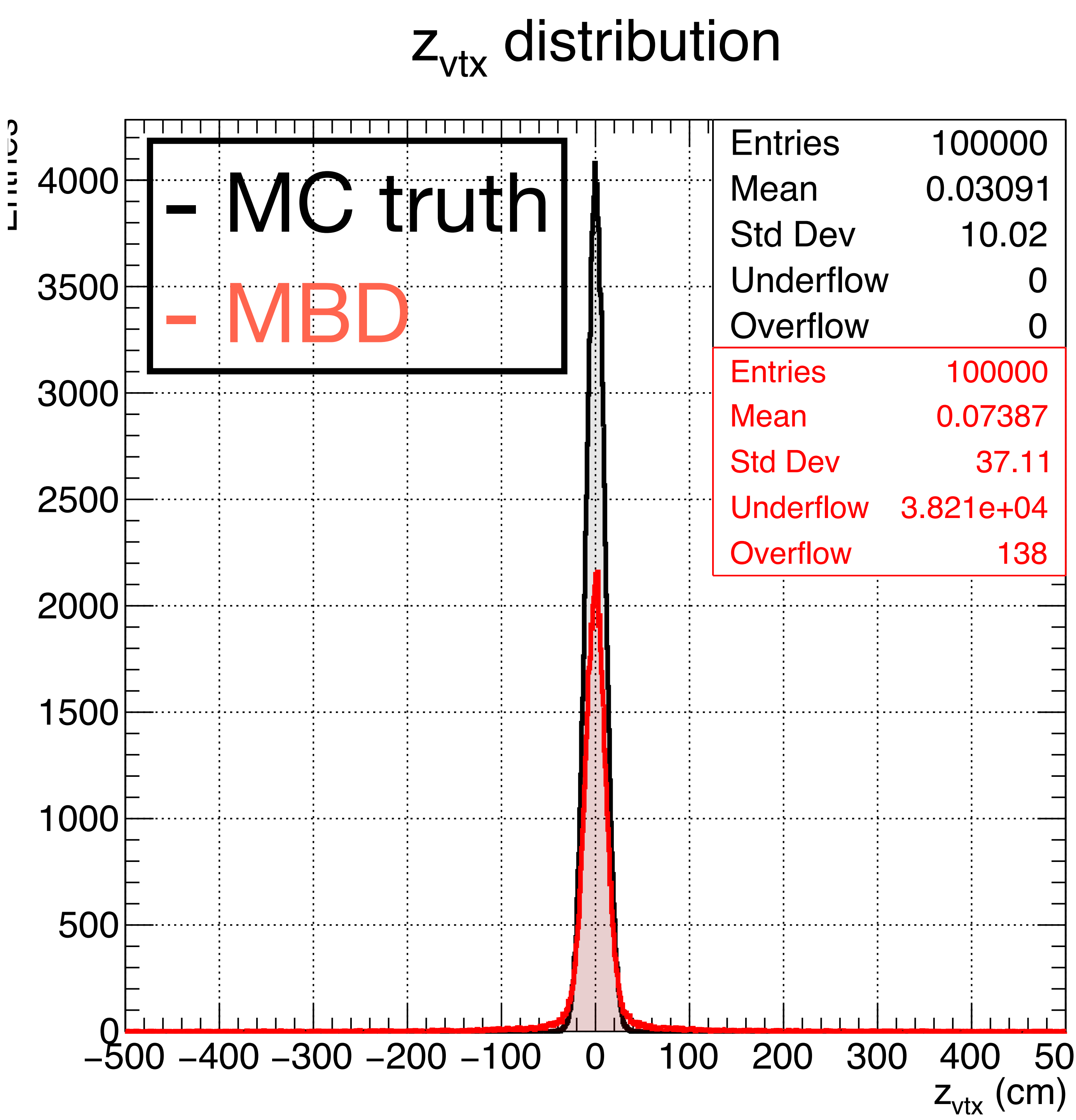


z_{vtx} distributions.

Underflow entry of MBD means invalid events.

MBD data analysis:

MbdOut & MC truth

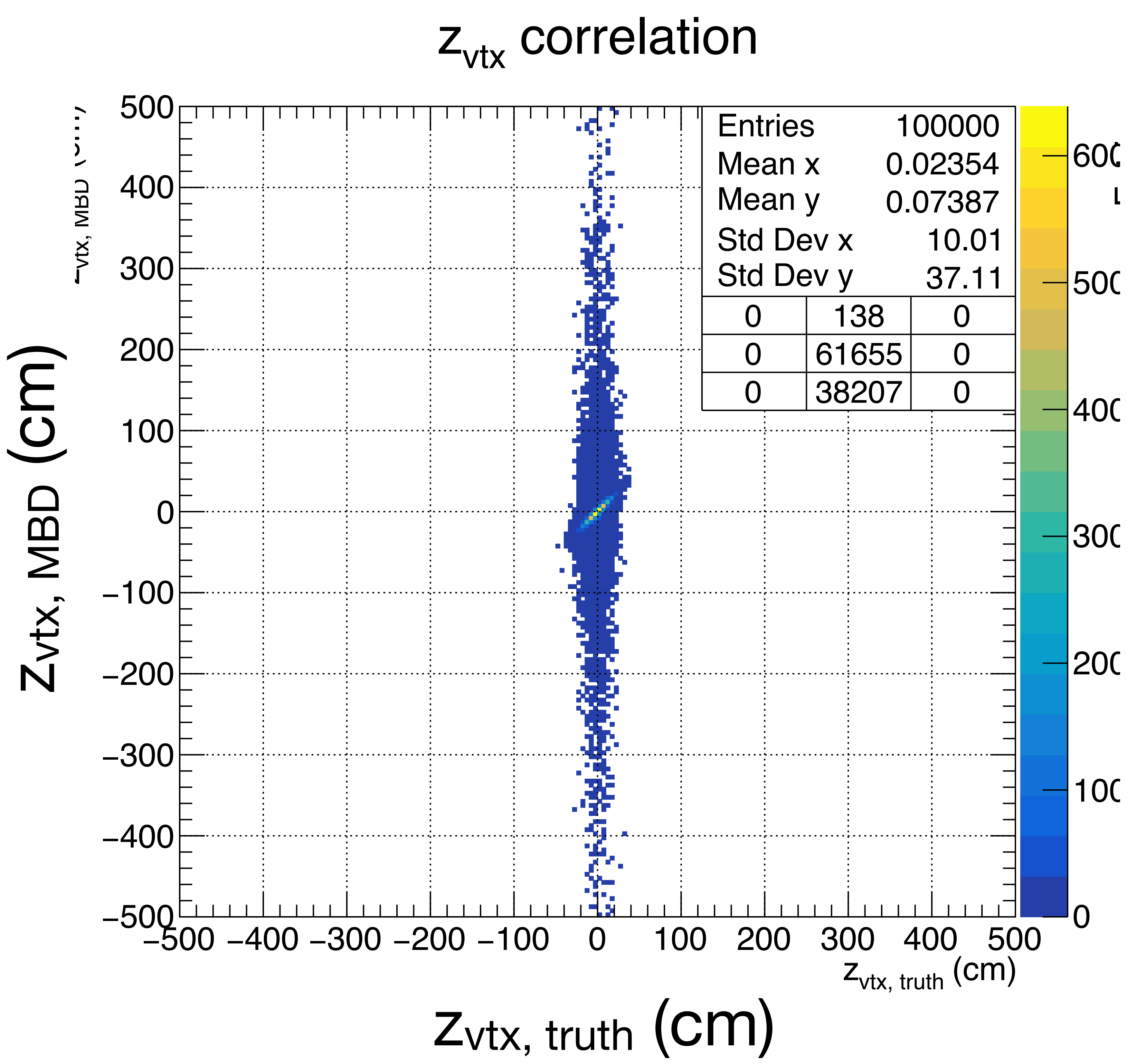


Entries	100000
Mean	0.03091
Std Dev	10.02
Underflow	0
Overflow	0

Entries	100000
Mean	0.07387
Std Dev	37.11
Underflow	3.821e+04
Overflow	138

z_{vtx} distributions.

Underflow entry of MBD means invalid events.



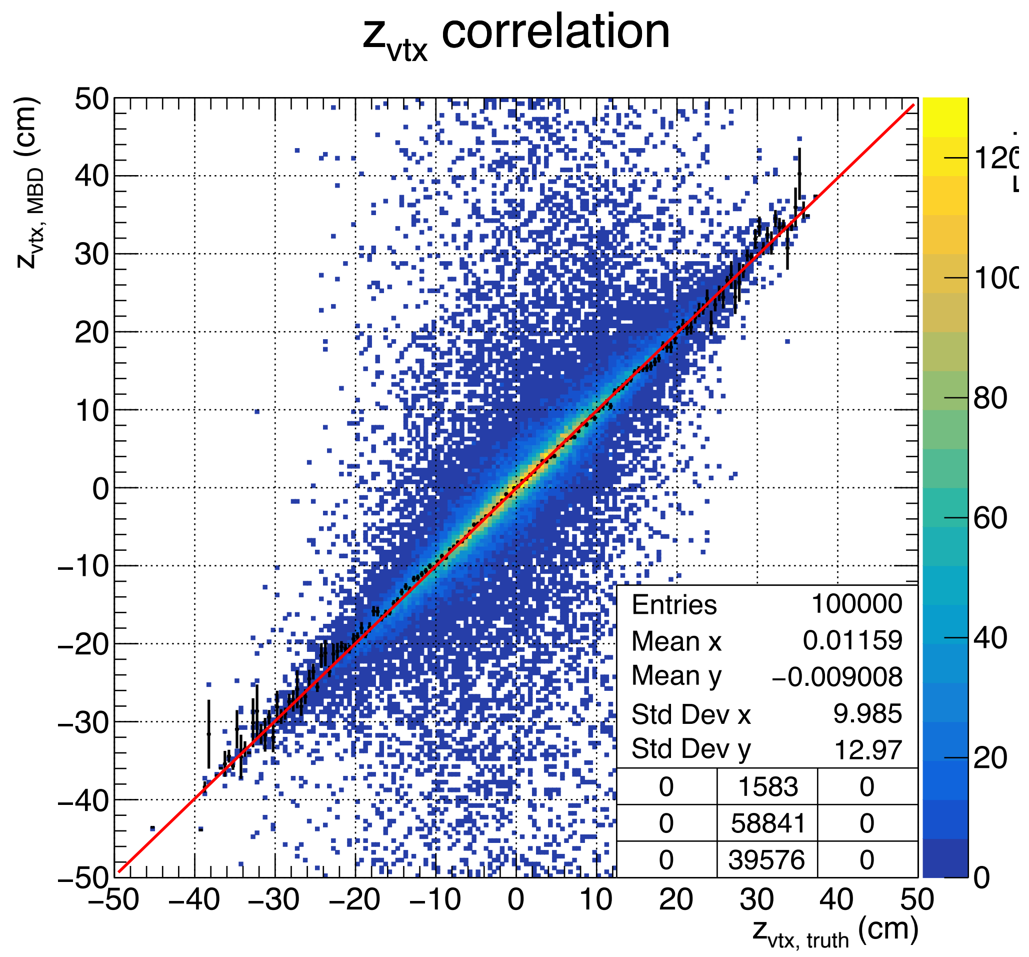
Entries	100000
Mean x	0.02354
Mean y	0.07387
Std Dev x	10.01
Std Dev y	37.11

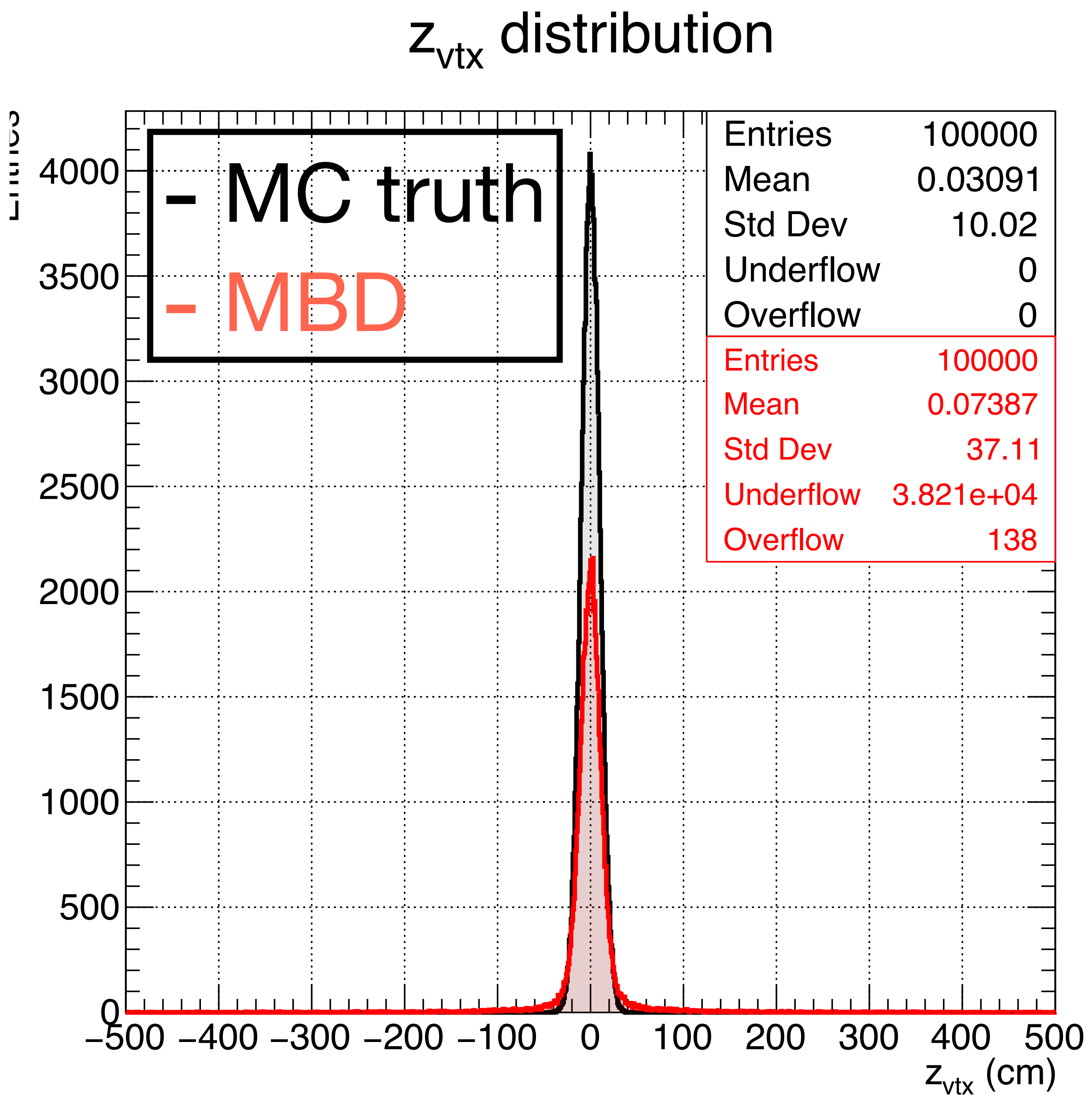
0	138	0
0	61655	0
0	38207	0

z_{vtx} correlation b/w MC truth and reconstructed by MBD.

MBD data analysis:

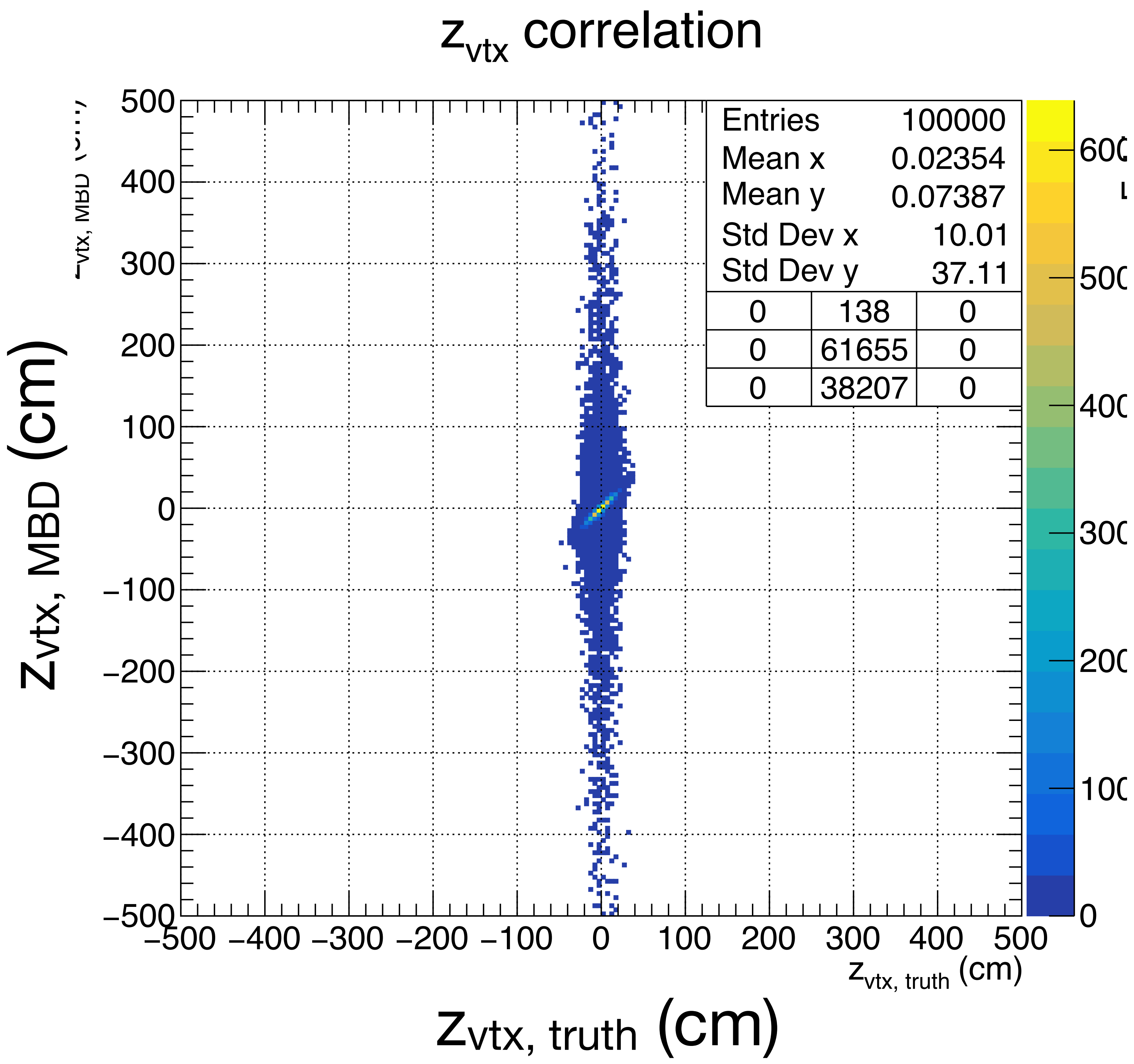
MbdOut & MC truth



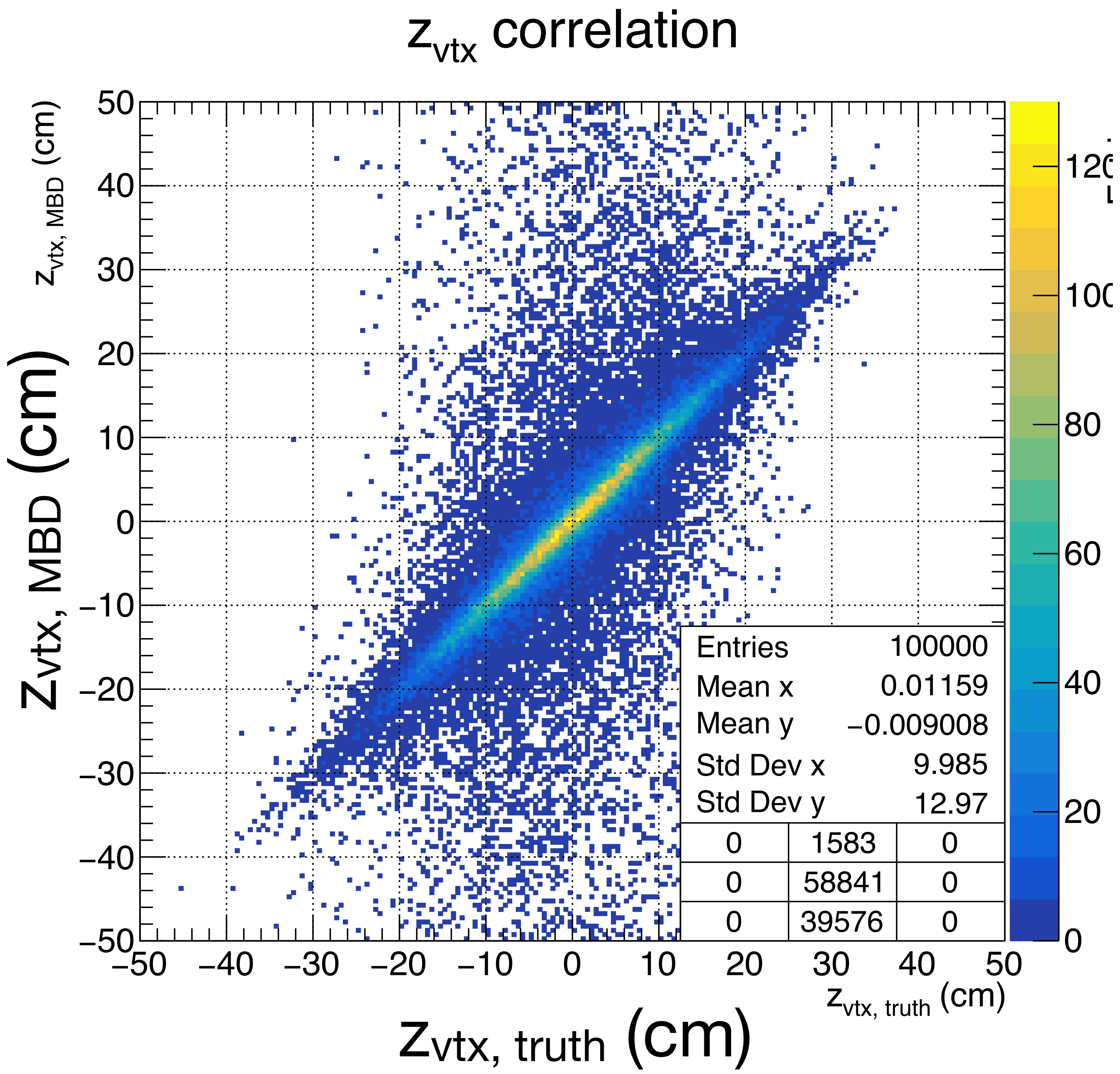


z_{vtx} distributions.

Underflow entry of MBD means invalid events.



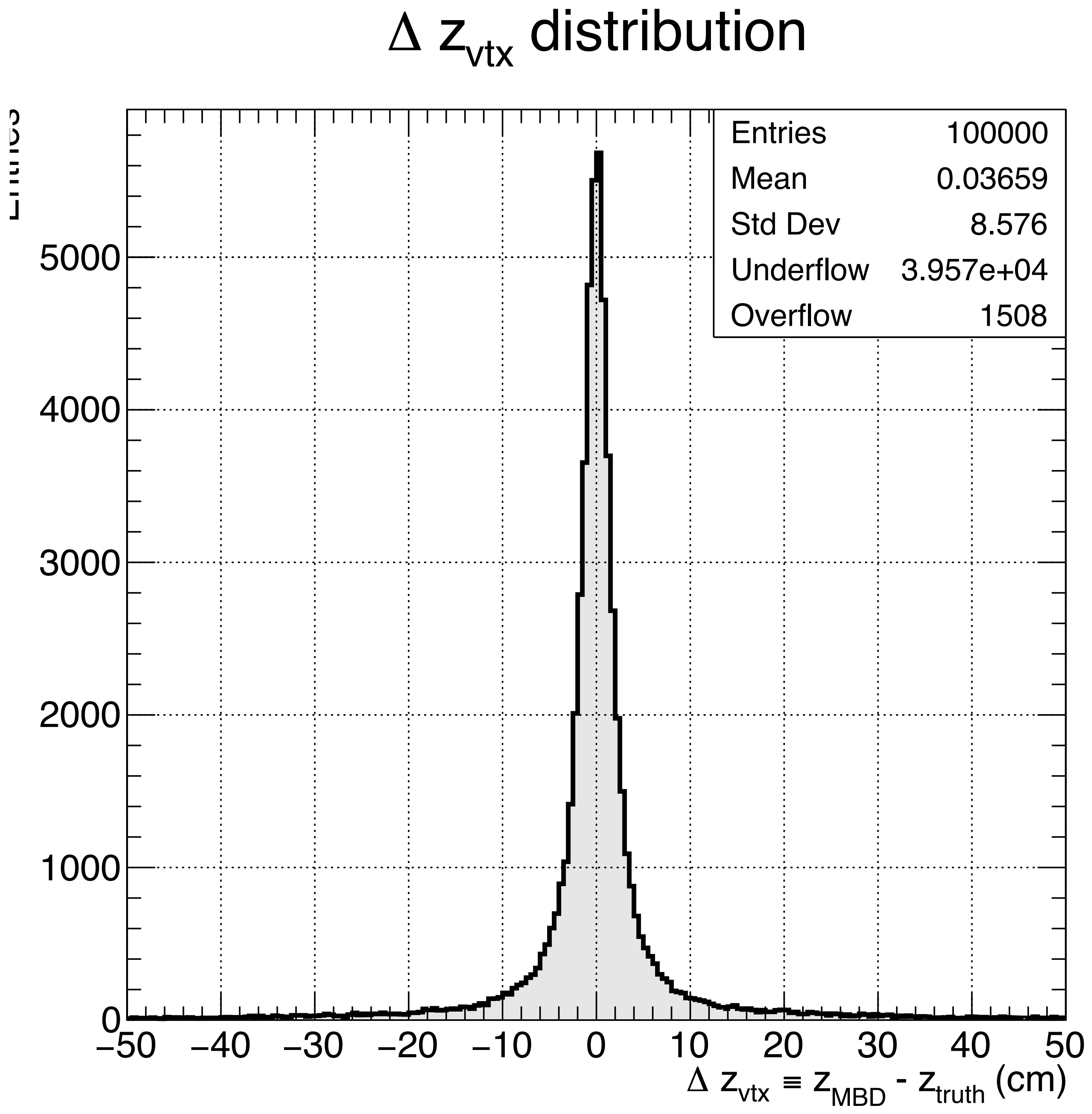
z_{vtx} correlation b/w MC truth and reconstructed by MBD.



z_{vtx} correlation b/w MC truth and reconstructed by MBD. (zoomed view)

MBD data analysis:

MbdOut & MC truth

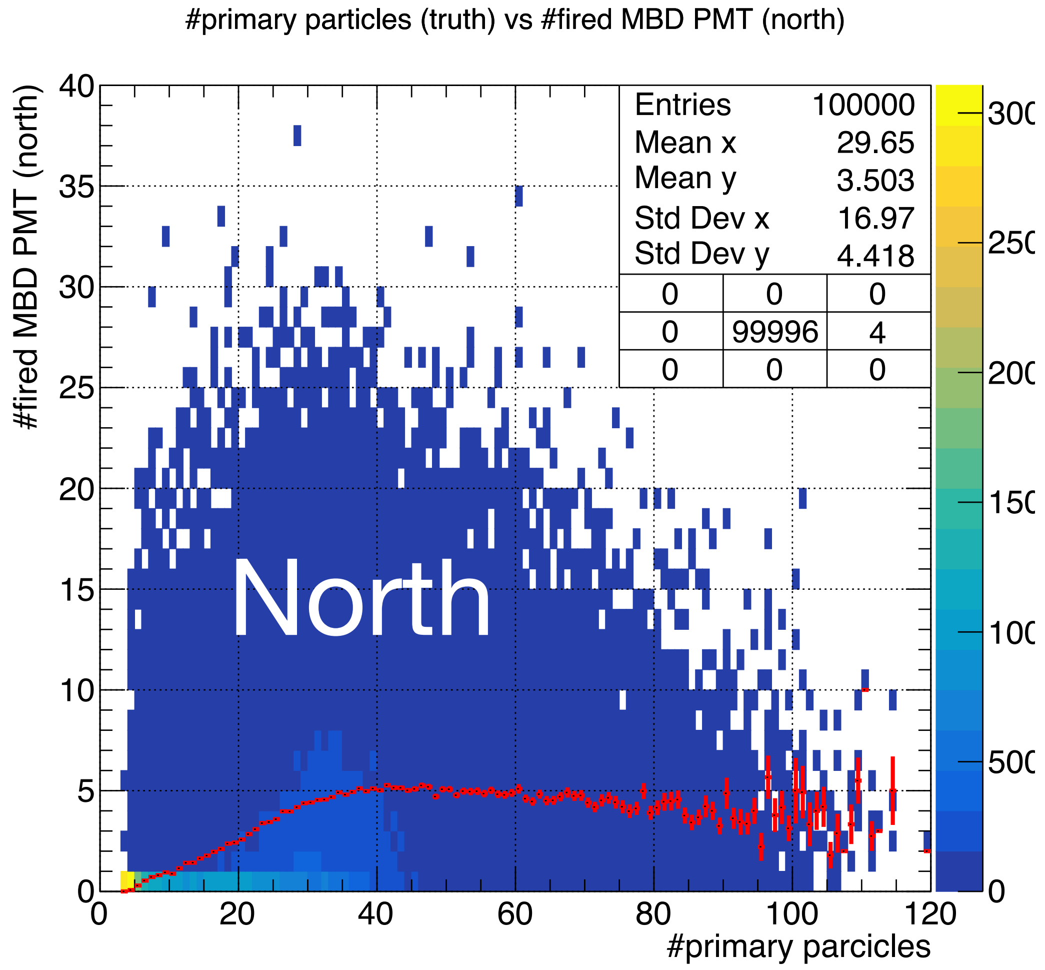


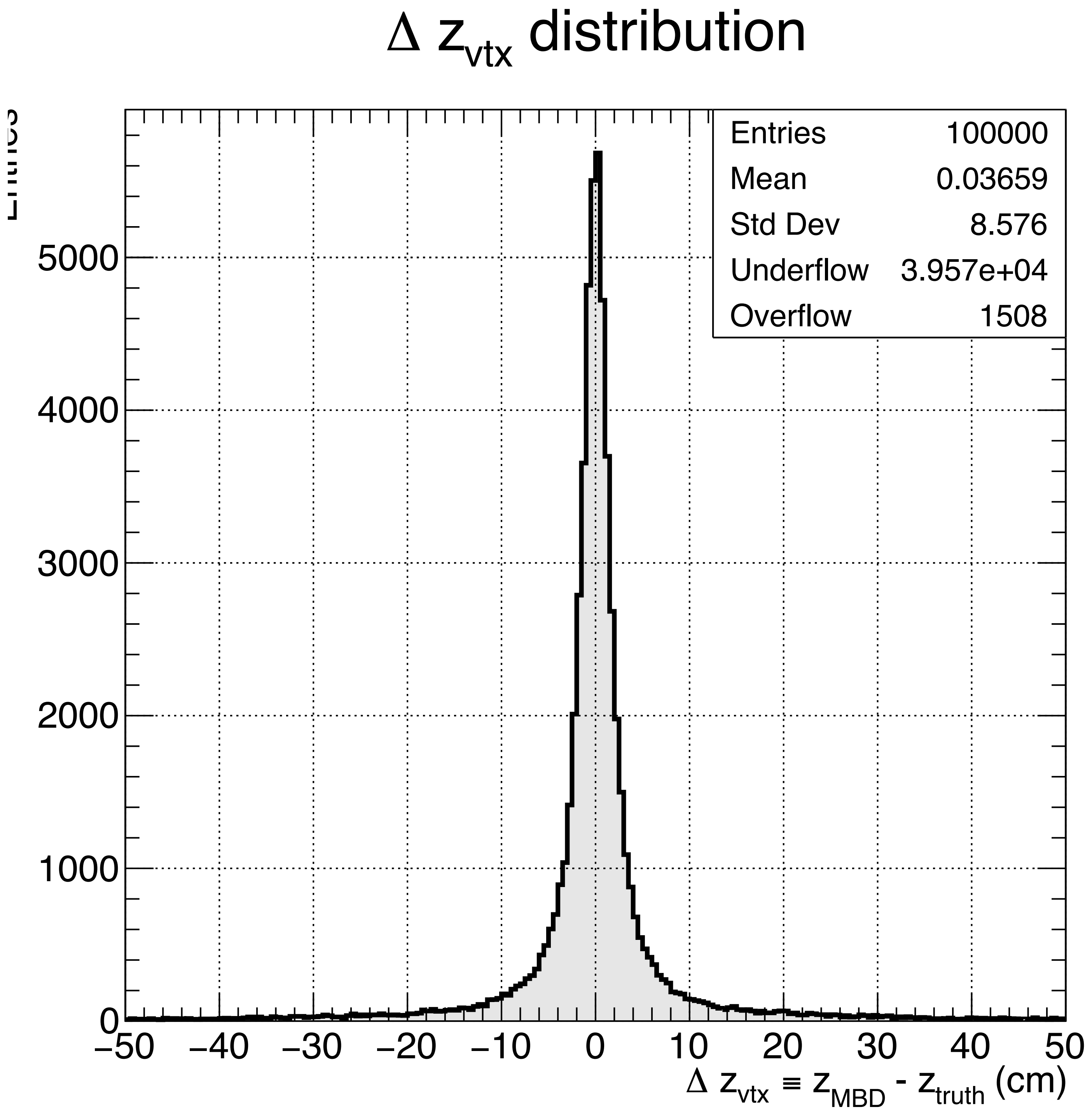
$z_{\mathrm{vtx,\ MBD}}$ - $z_{\mathrm{vtx,\ MC\ truth}}$ distribution.

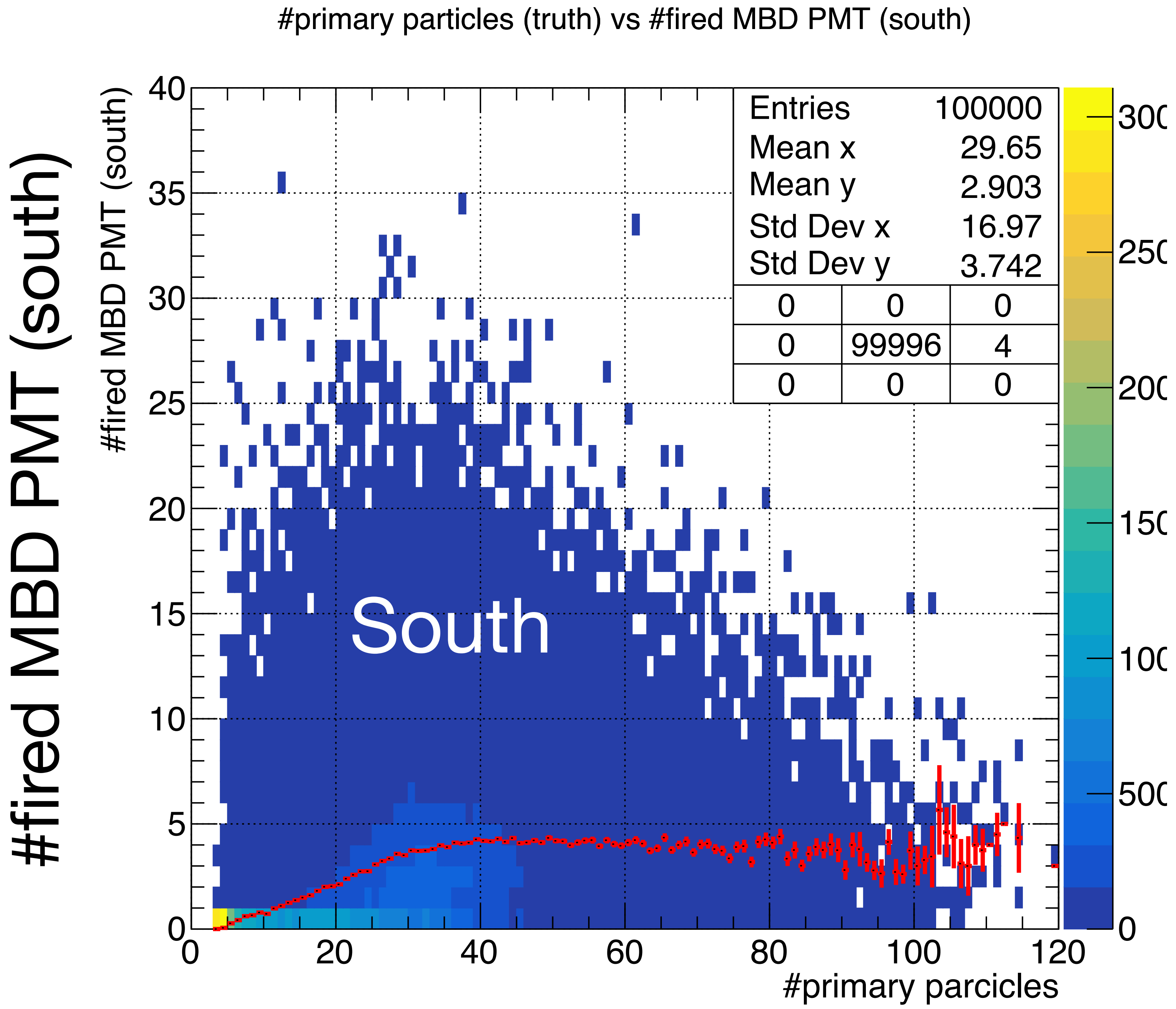
Underflow entry of MBD means invalid events.

MBD data analysis:

MbdOut & MC truth







$z_{vtx,\ MBD}$ - $z_{vtx,\ MC\ truth}$ distribution.

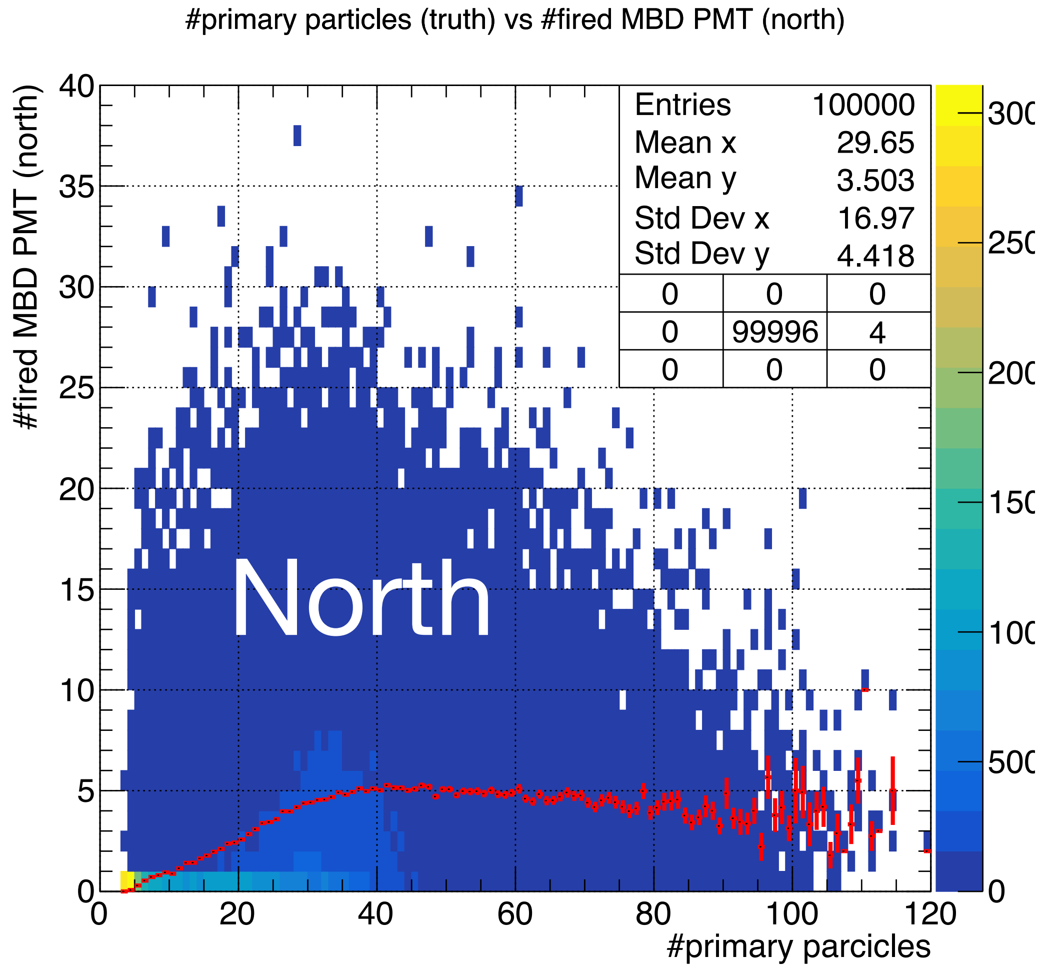
Underflow entry of MBD means invalid events.

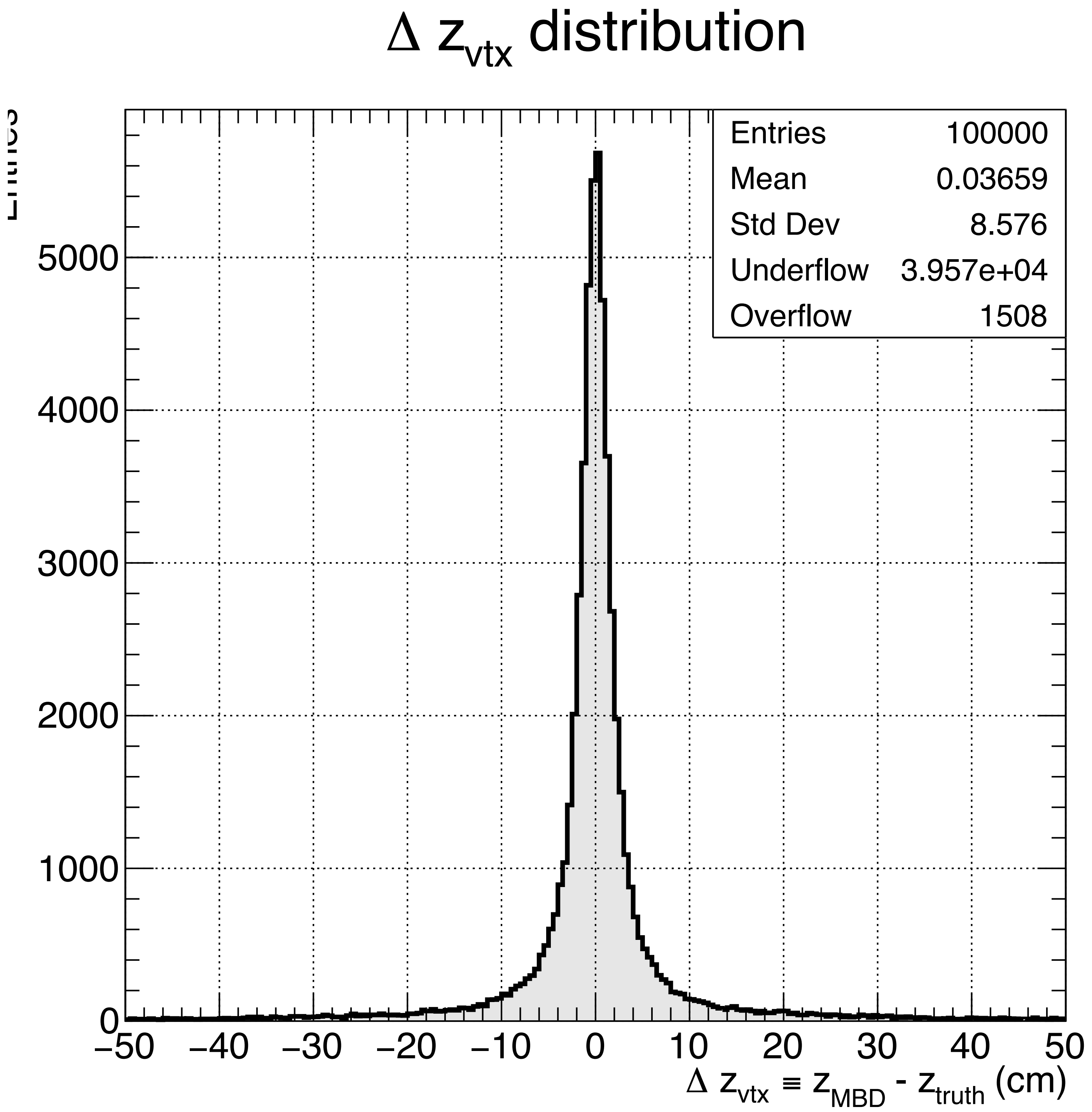
#primary particle vs #fired MBD PMT (south).

The more particles are generated, the more particles goes into MBD

MBD data analysis:

MbdOut & MC truth

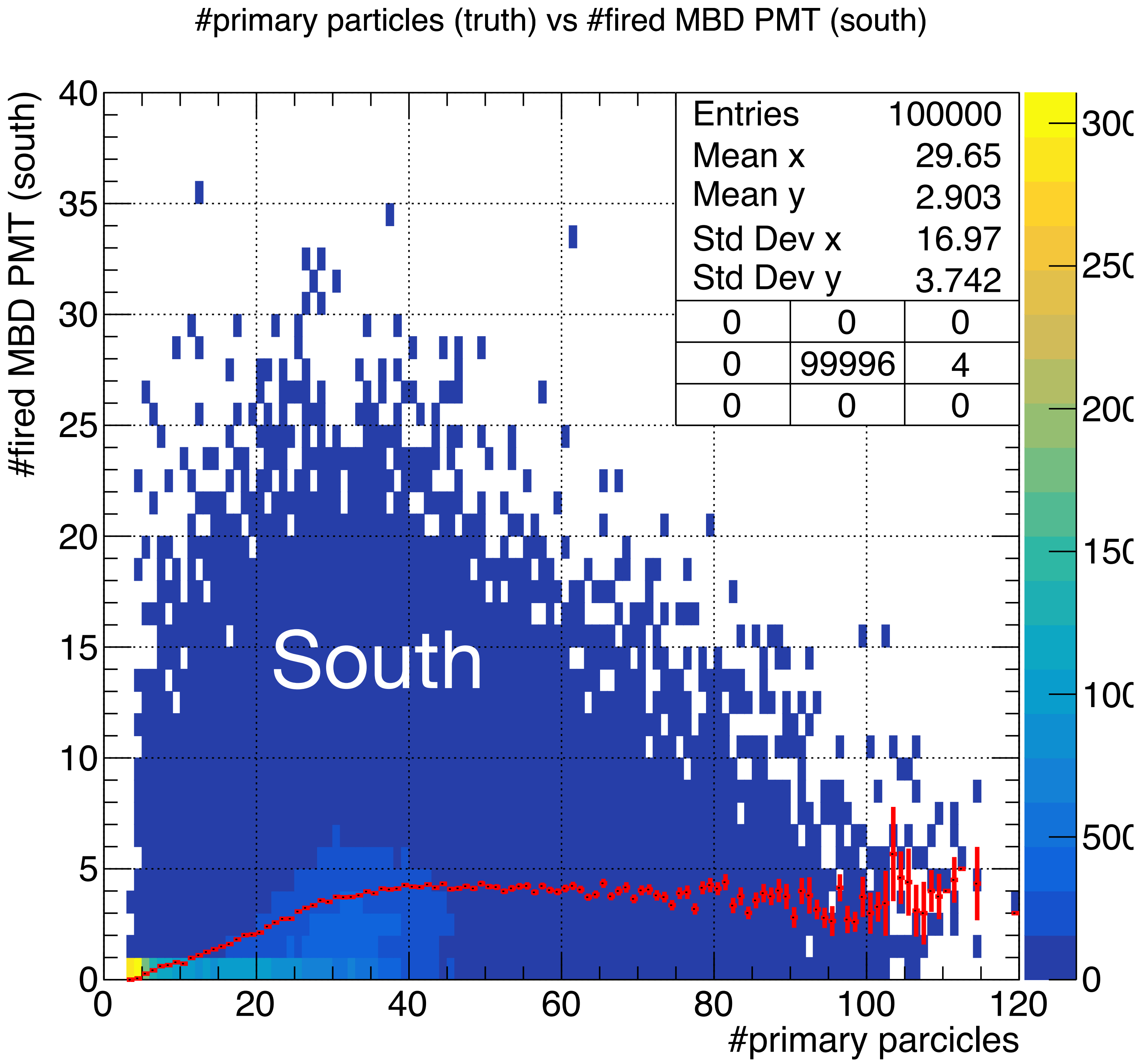




$z_{\mathrm{vtx,\ MBD}}$ - $z_{\mathrm{vtx,\ MC\ truth}}$ distribution.

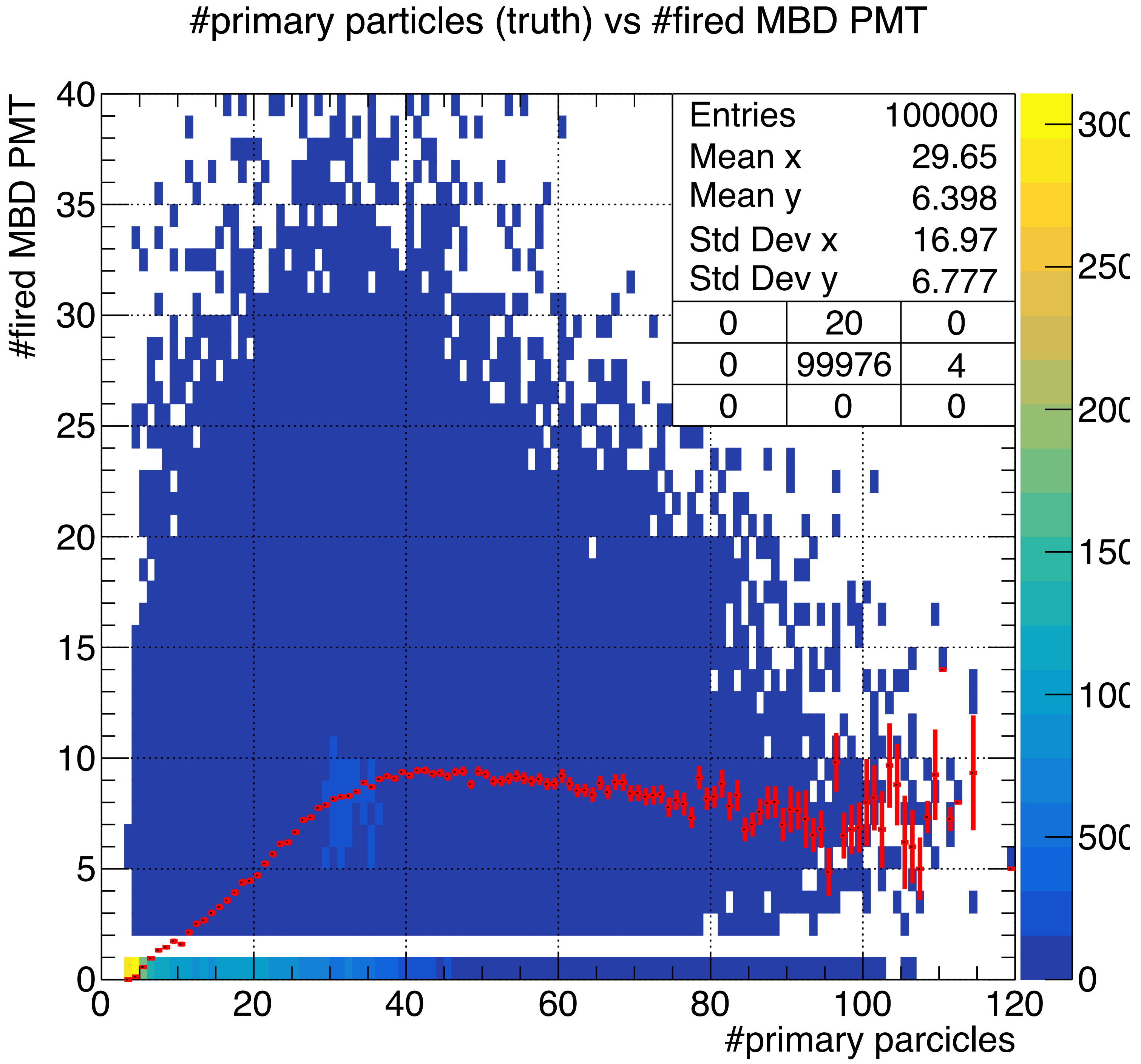
Underflow entry of MBD means invalid events.

#fired MBD PMT (south)



#primary particle vs #fired MBD PMT (south).

The more particles are generated, the more particles goes into MBD



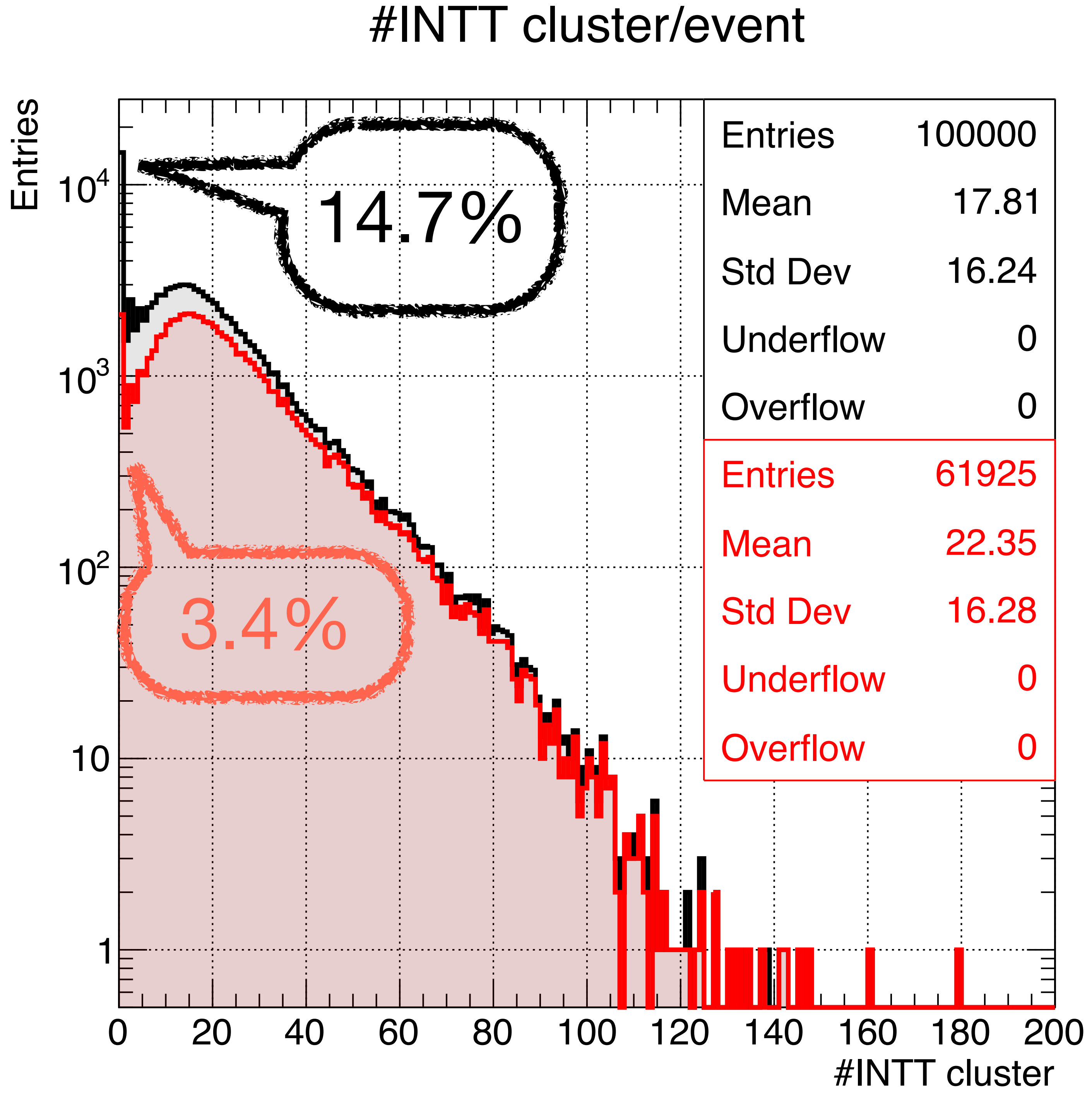
#primary particle vs #fired MBD PMT (north + south).

MBD data analysis:

MbdOut & INTT cluster

- No cut

- Simulated MBD trigger



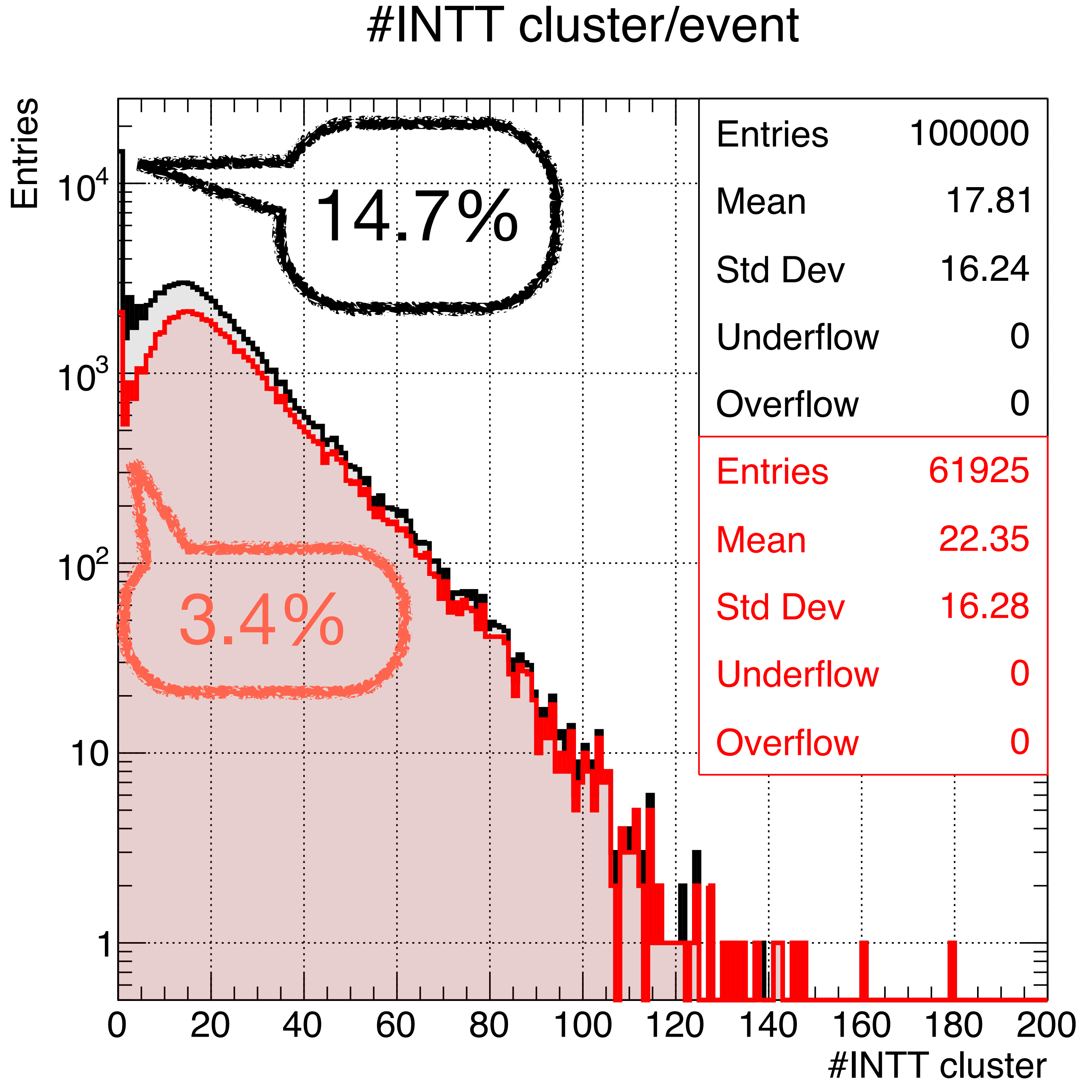
Entries	100000
Mean	17.81
Std Dev	16.24
Underflow	0
Overflow	0
Entries	61925
Mean	22.35
Std Dev	16.28
Underflow	0
Overflow	0

#INTT cluster/event.

MBD data analysis:

MbdOut & INTT cluster

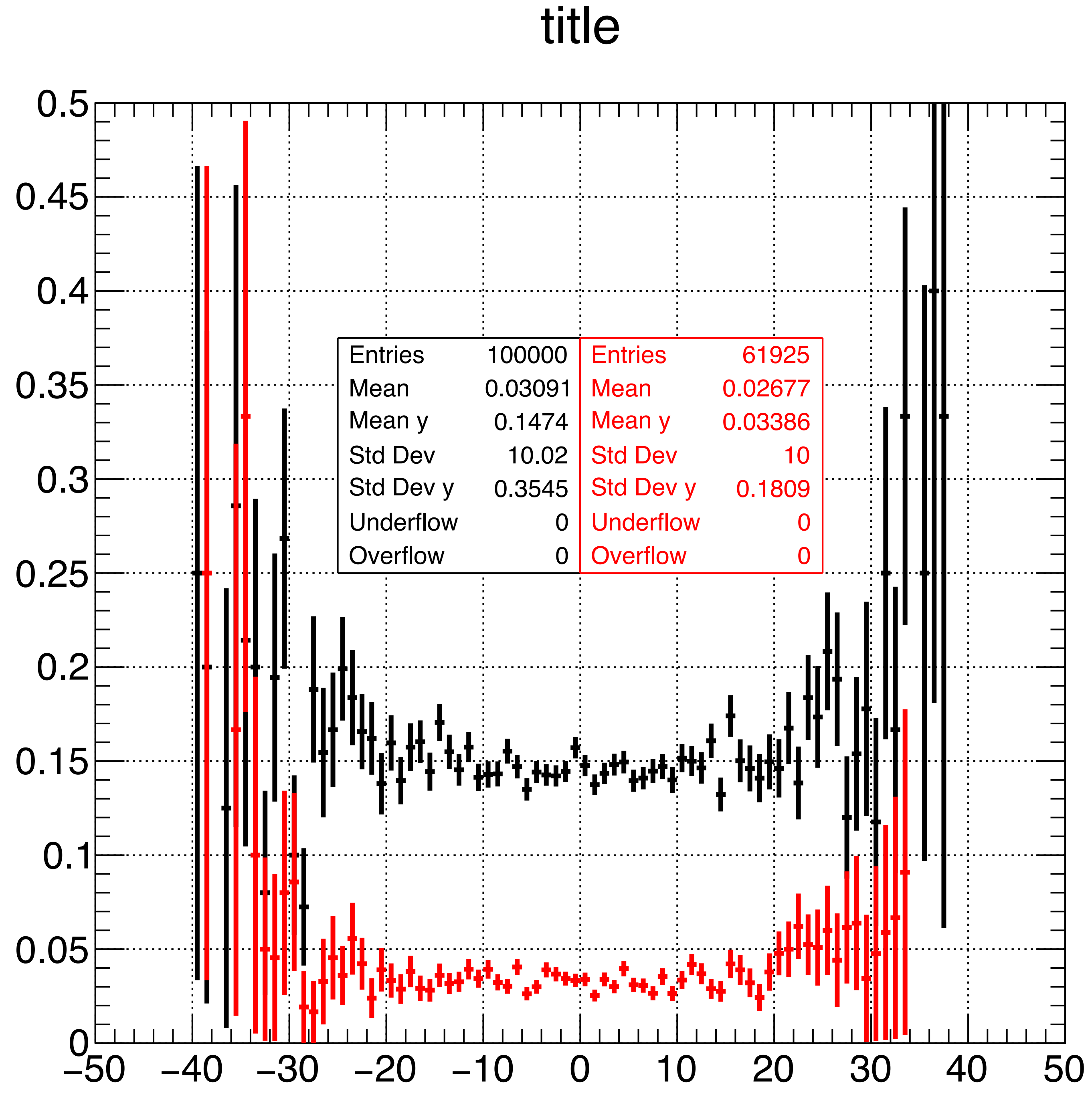
- No cut
- Simulated MBD trigger



Entries	100000
Mean	17.81
Std Dev	16.24
Underflow	0
Overflow	0
Entries	61925
Mean	22.35
Std Dev	16.28
Underflow	0
Overflow	0

#INTT cluster/event.

- No cut
- Simulated MBD trigger

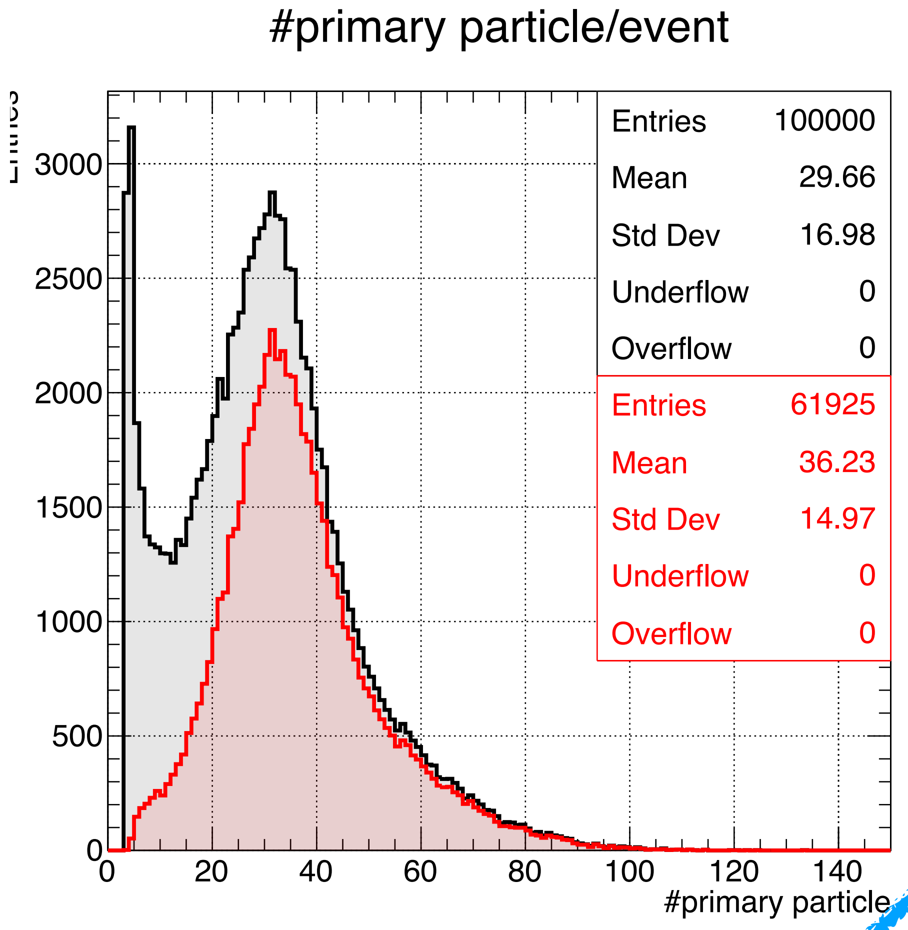


Entries	100000	Entries	61925
Mean	0.03091	Mean	0.02677
Mean y	0.1474	Mean y	0.03386
Std Dev	10.02	Std Dev	10
Std Dev y	0.3545	Std Dev y	0.1809
Underflow	0	Underflow	0
Overflow	0	Overflow	0

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

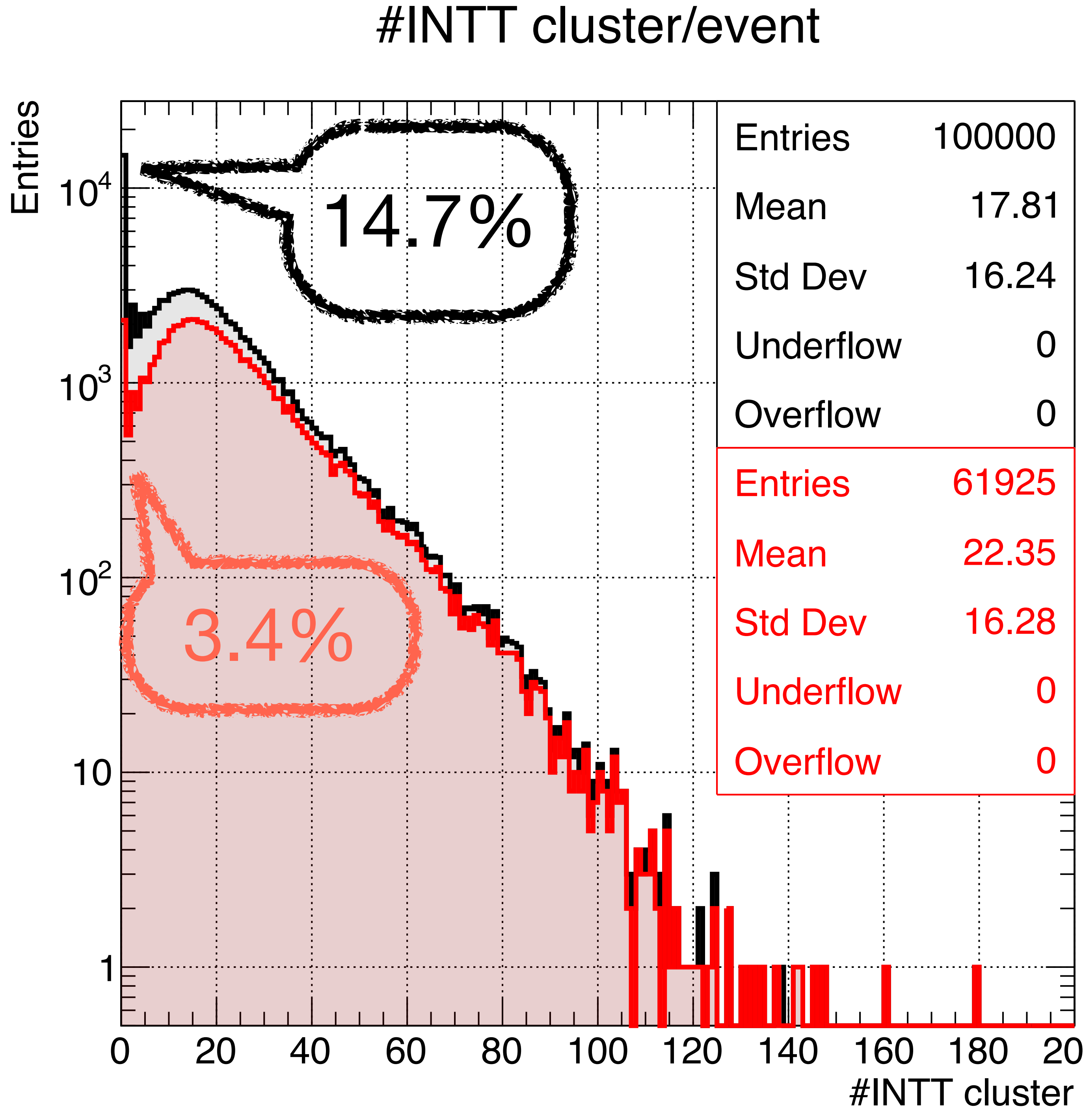
MBD data analysis:

MbdOut & INTT cluster



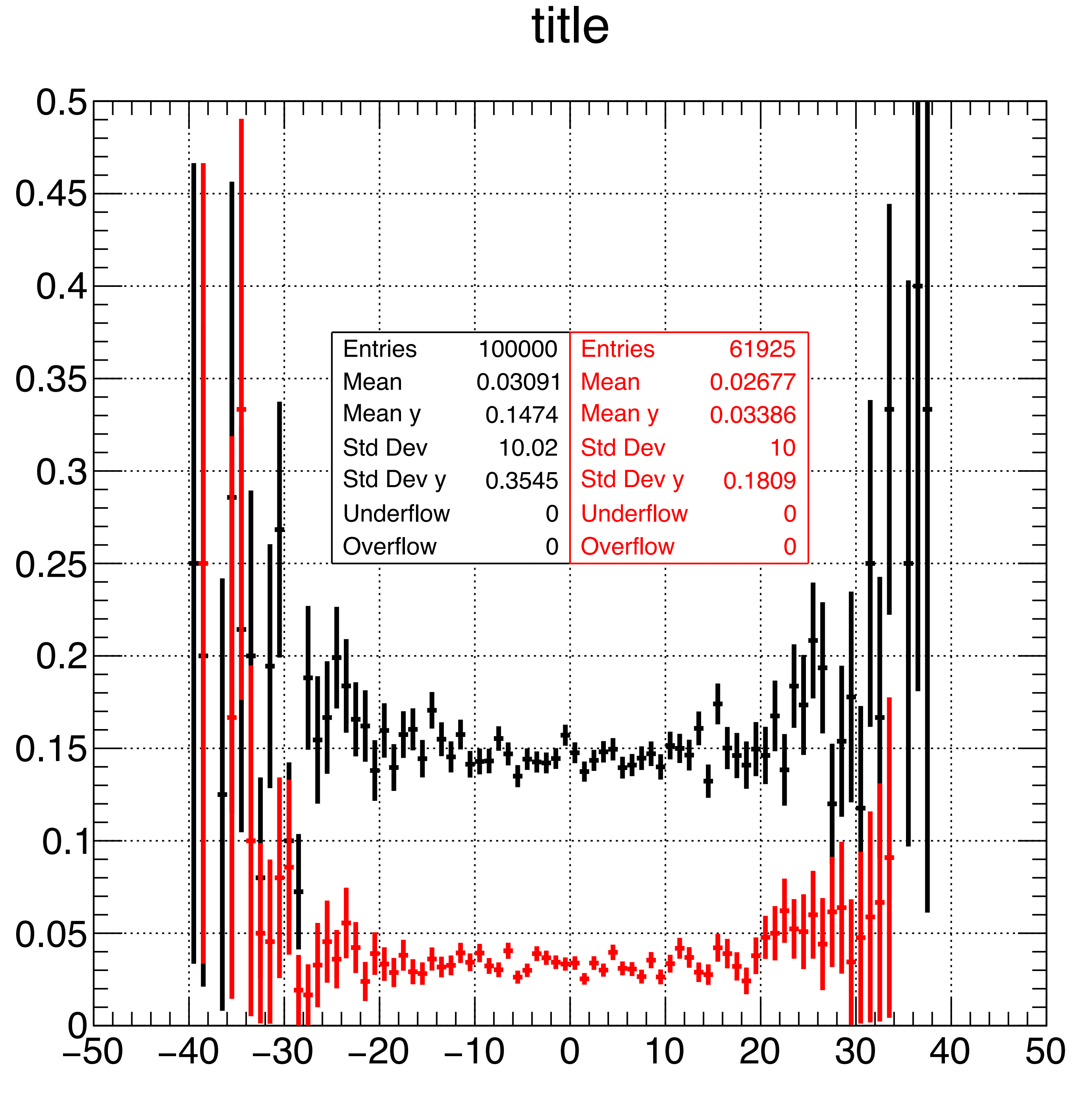
Can such events fire MBD trigger?

- No cut
- Simulated MBD trigger

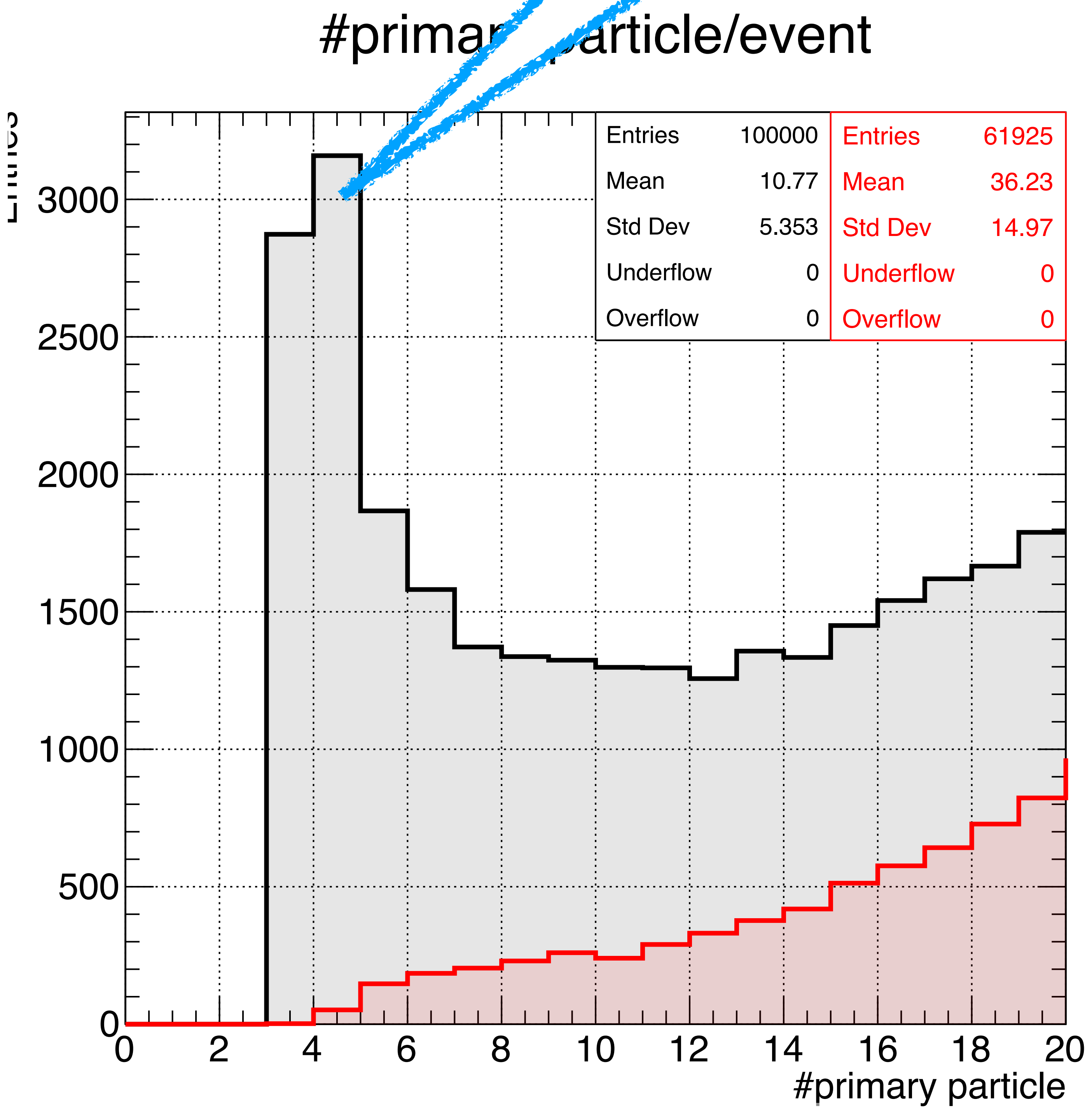


#INTT cluster/event.

- No cut
- Simulated MBD trigger



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

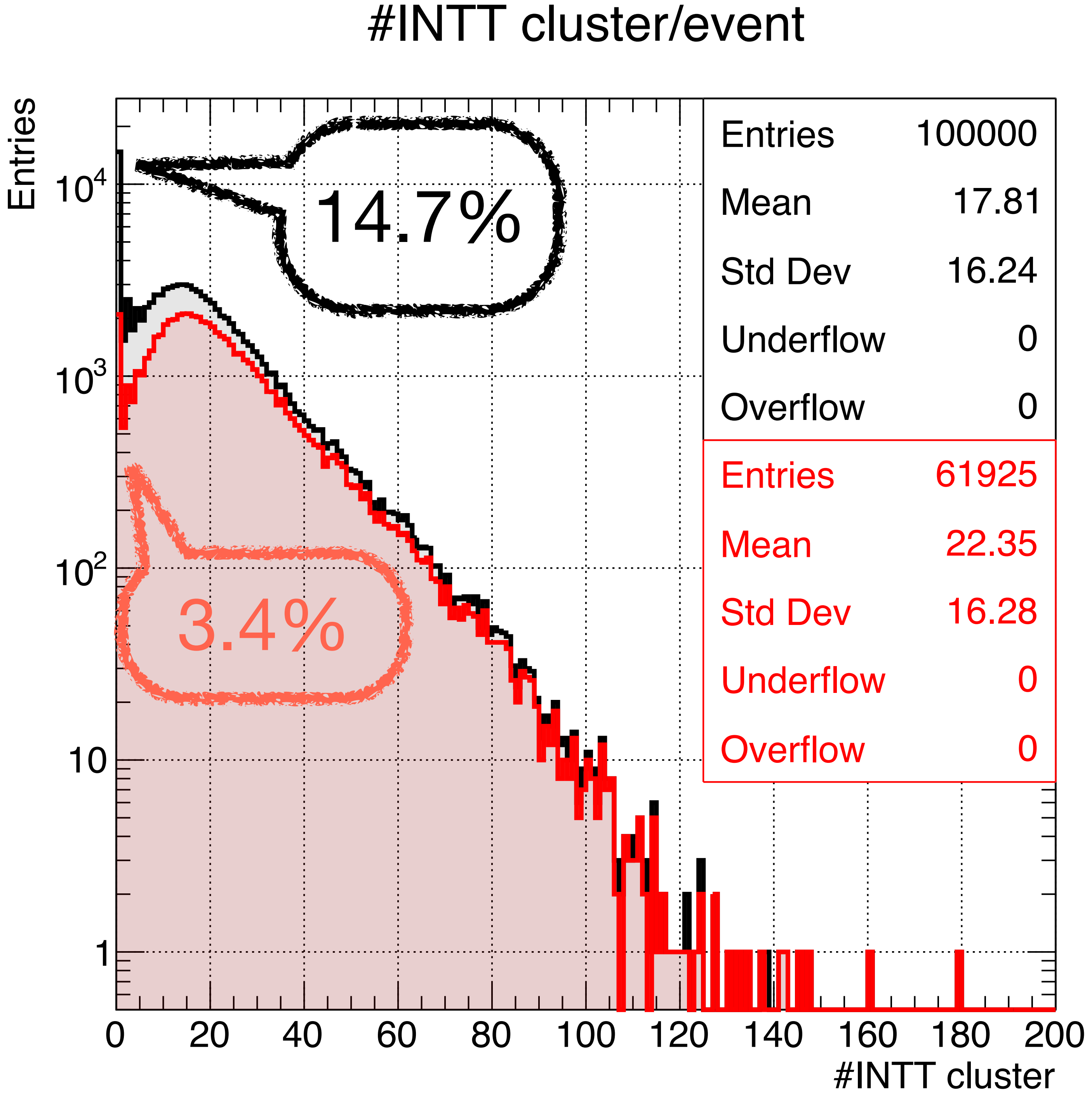


$\frac{\text{\#prim particle}}{\text{event}}$

MBD data analysis:

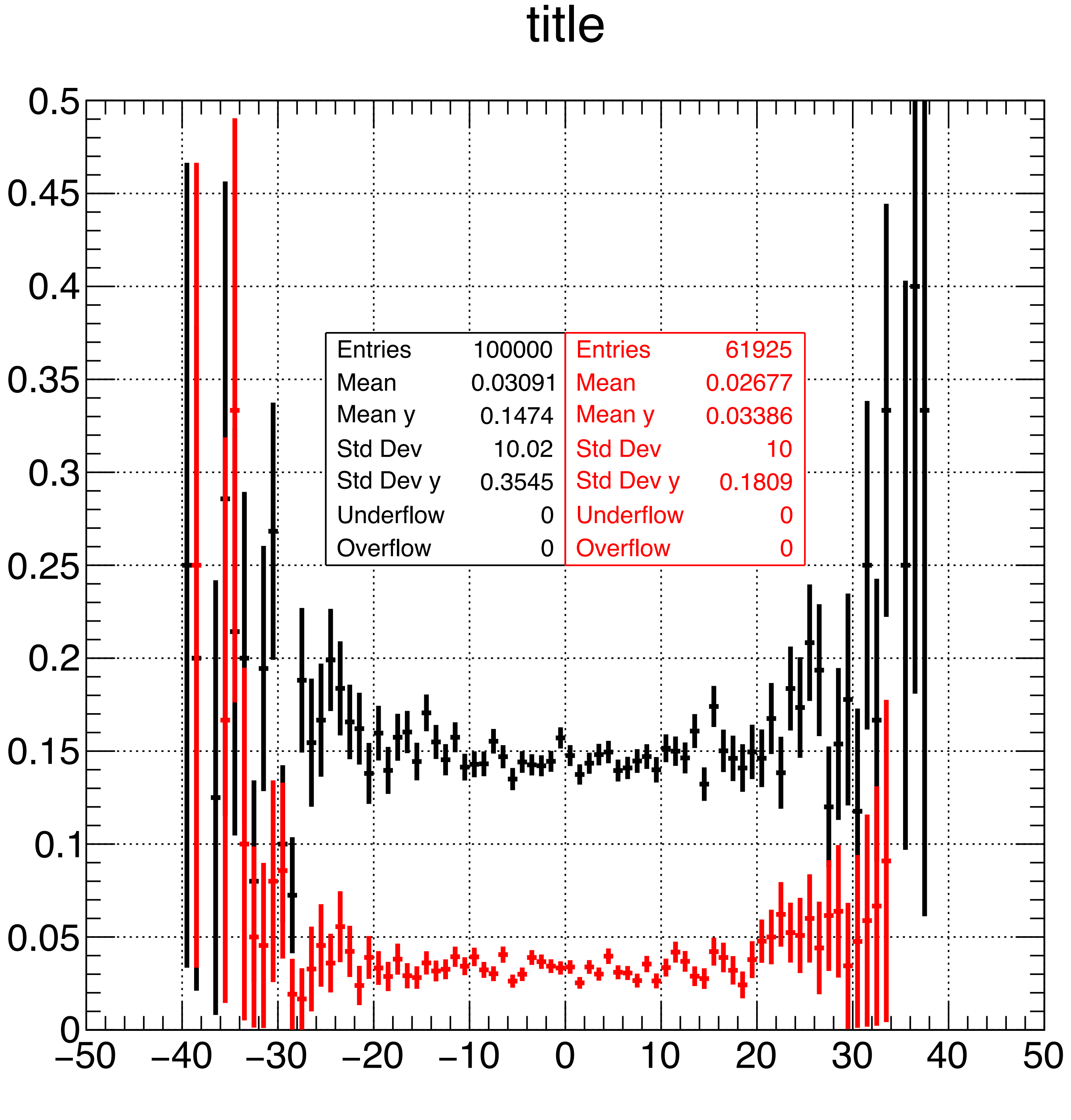
MbdOut & INTT cluster

- No cut
- Simulated MBD trigger

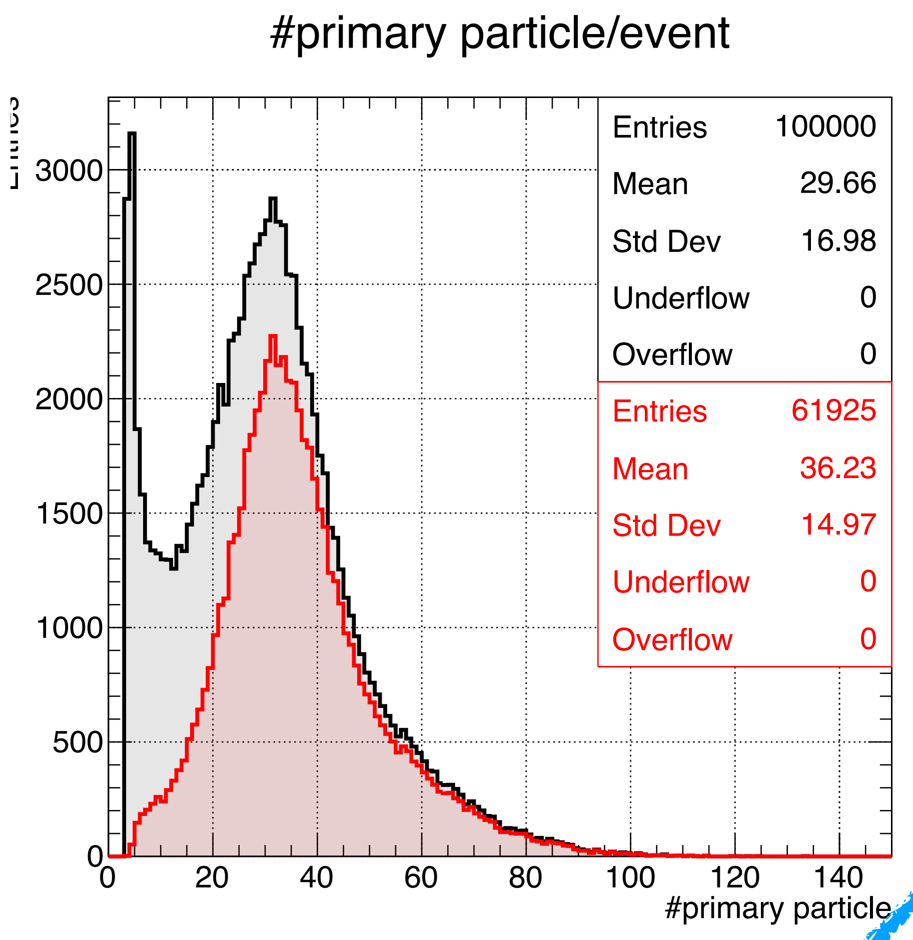


#INTT cluster/event.

- No cut
- Simulated MBD trigger

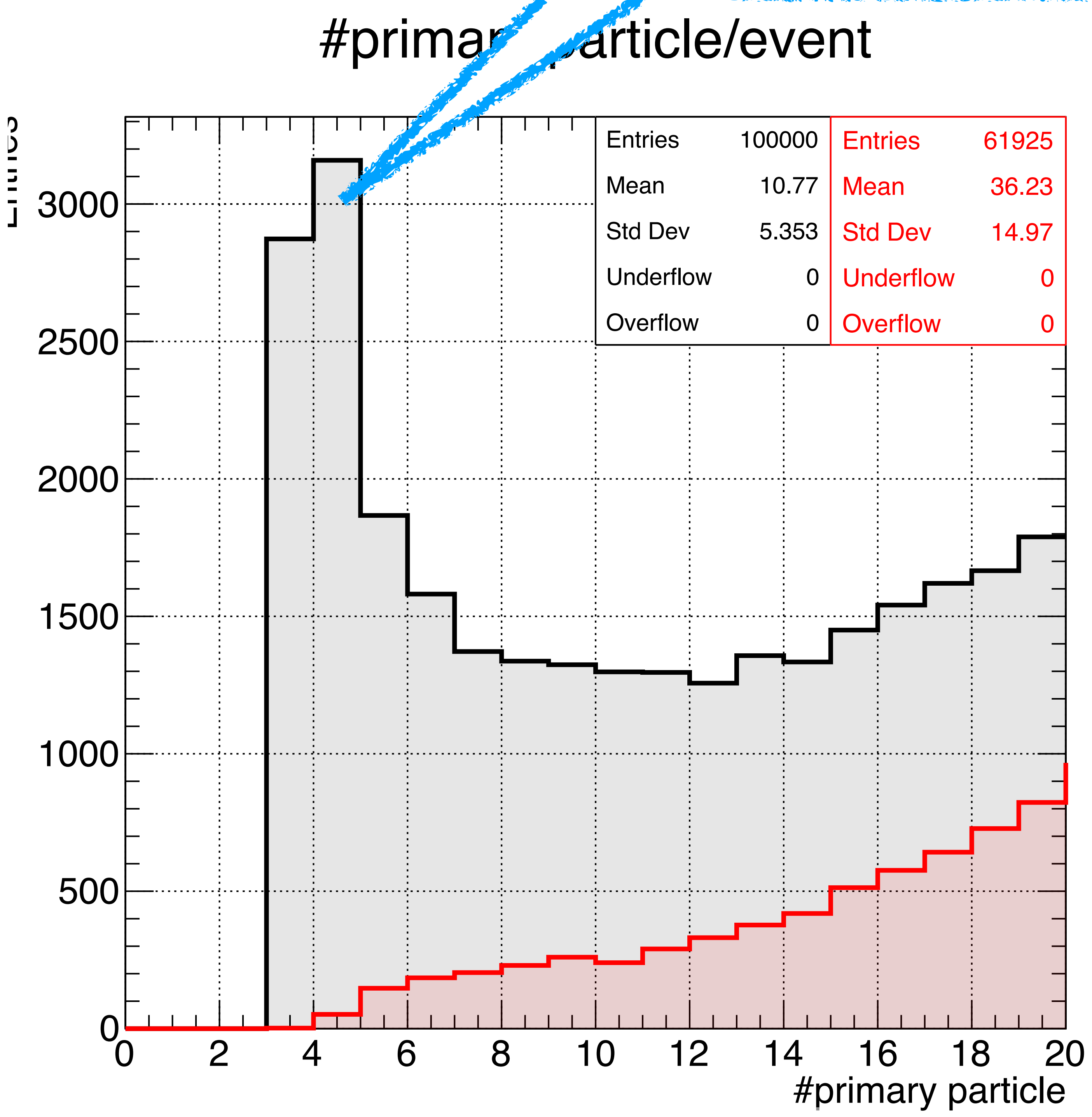


Ratio of #INTT cluster/event = 0 as a function of truth z_{vtx}



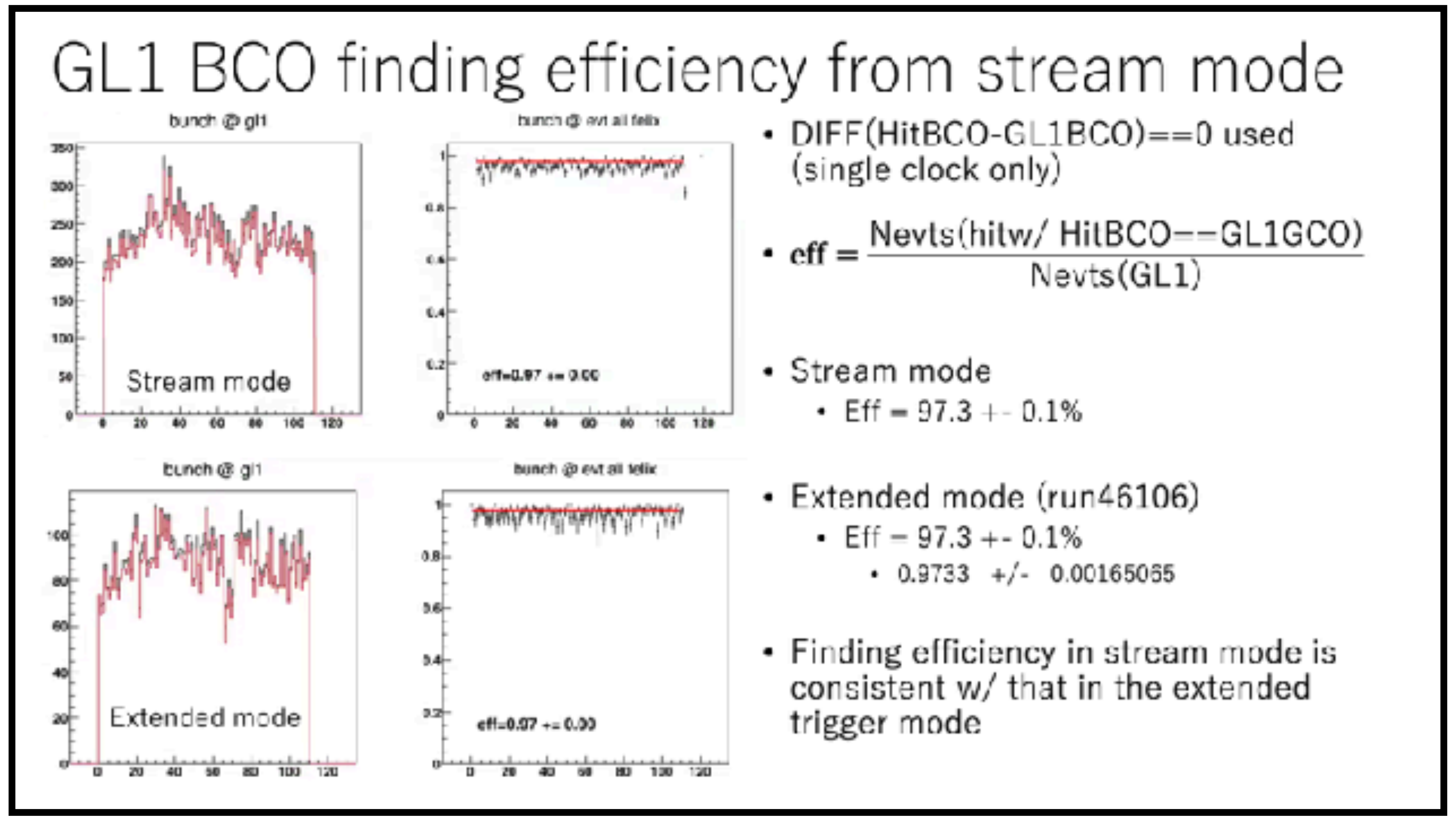
Can such events fire MBD trigger?

NO



#prim particle
event

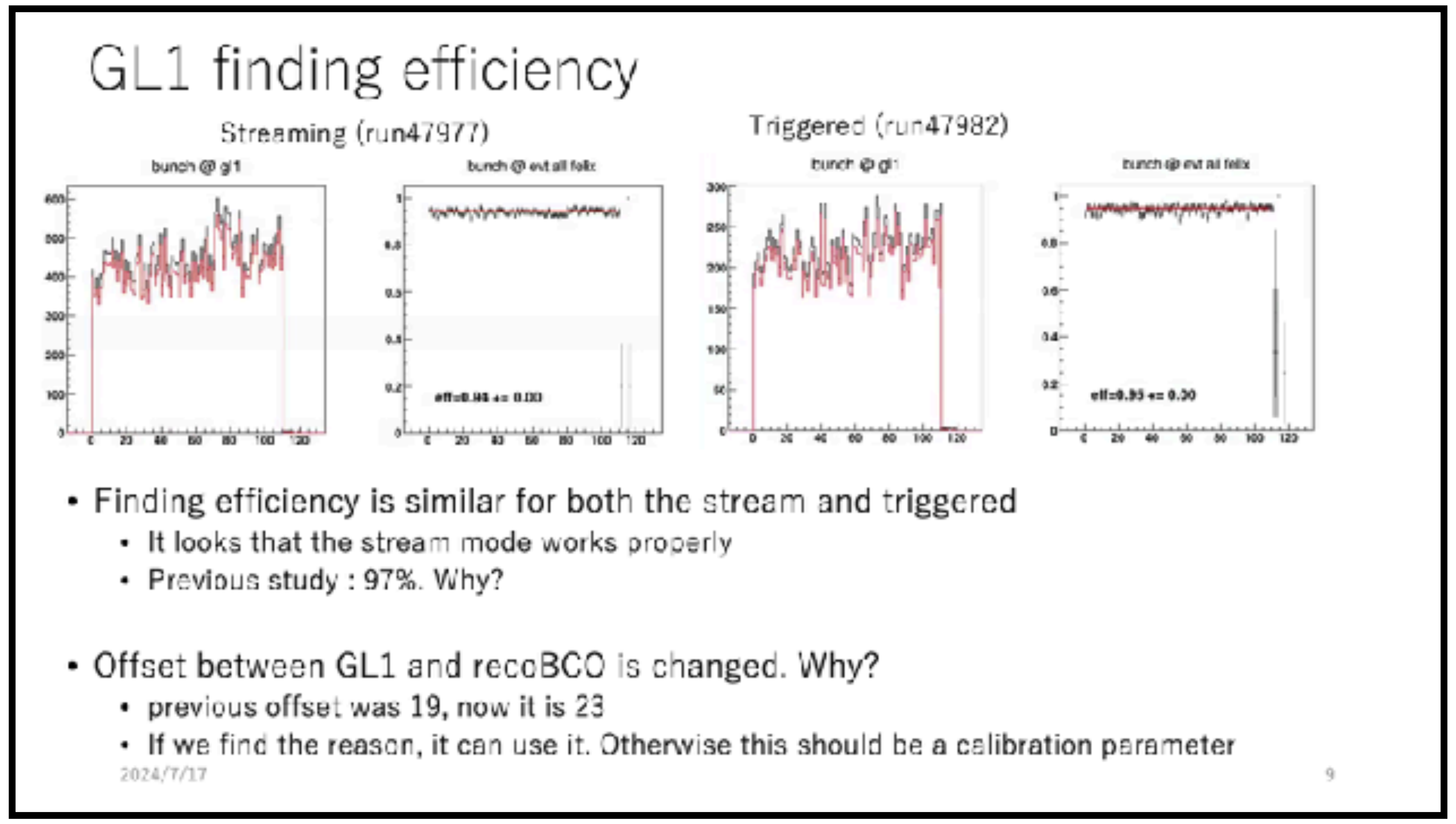
Summary



GL1 BCO finding efficiency from stream mode

- DIFF(HitBCO-GL1BCO)==0 used (single clock only)
- $\text{eff} = \frac{\text{Nevts(hitw/ HitBCO==GL1GCO)}}{\text{Nevts(GL1)}}$
- Stream mode
  - Eff = 97.3 +- 0.1%
- Extended mode (run46106)
  - Eff = 97.3 +- 0.1%
    - 0.9733 +/- 0.00165065
- Finding efficiency in stream mode is consistent w/ that in the extended trigger mode

Takashi's first report

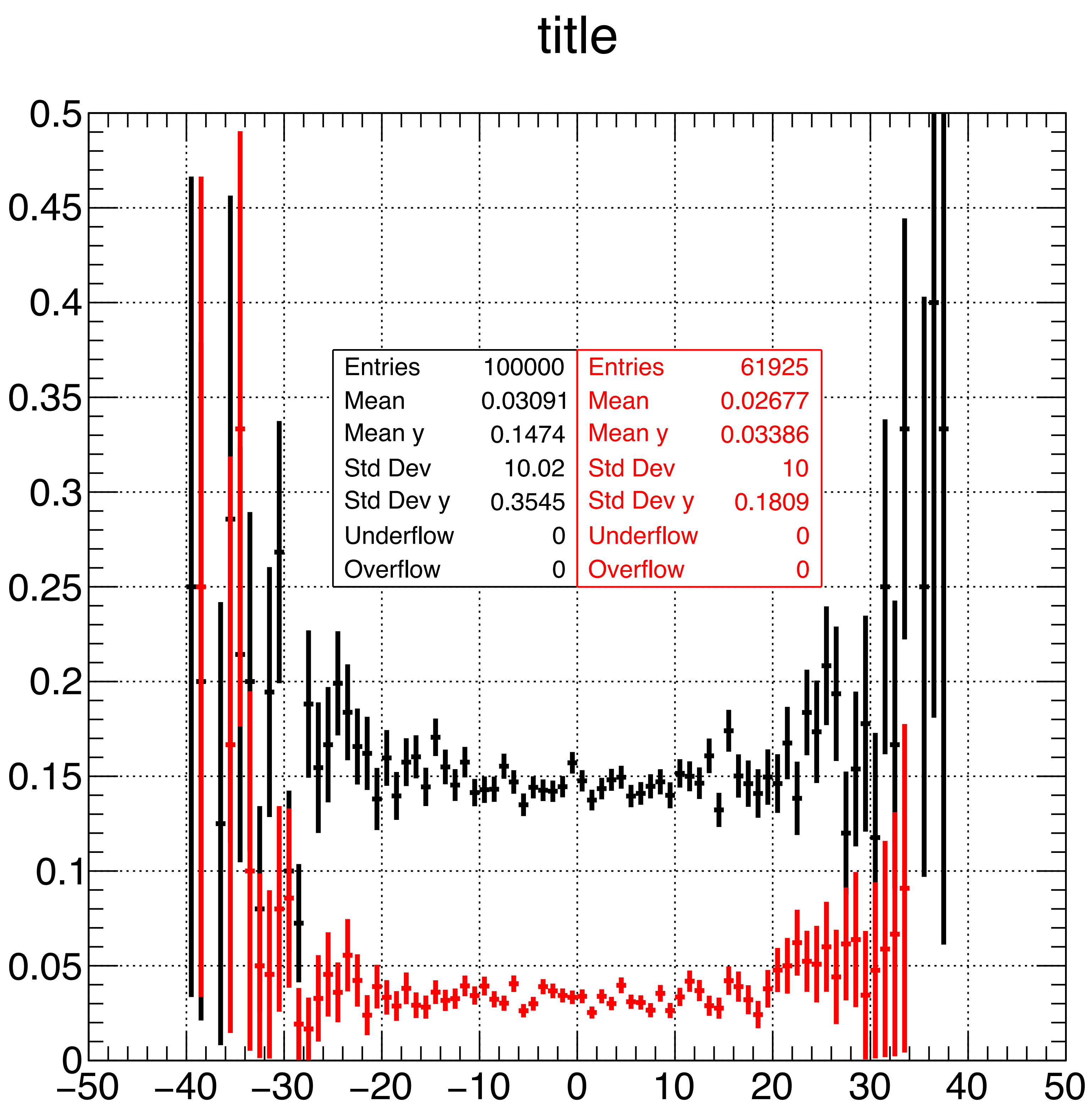


GL1 finding efficiency

- Finding efficiency is similar for both the stream and triggered
  - It looks that the stream mode works properly
  - Previous study : 97%. Why?
- Offset between GL1 and recoBCO is changed. Why?
  - previous offset was 19, now it is 23
  - If we find the reason, it can use it. Otherwise this should be a calibration parameter

2024/7/17 9

Takashi's latest report



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

- No cut
- Simulated MBD trigger

GL1 finding efficiency

- Takashi's first report: 97.3%
- Takashi's latest: 94 — 95%
- Suggested efficiency from MC study: 96 — 97%

Note: MBD trigger mimic needs to be checked by the MBD group.