analysis tools
in e14lib developing version

3 Aug. 2012

Sato Kazufumi (Osaka Univ.)

CsI data analysis

data from FADC (in KEKCC:/hsm/had/koto/)

↓ convert to rootfile

conv data

pulse shape information

Feb. data (in run40/conv_data/crate[0-10])

run????_conv.root

Jun. data (in run43/fiber_out/node[0-10])

run????_conv_crate[0-11]_node[0-9].root

↓ convert (pulse shape info are lost)

sumup data

ID, energy(integratedADC) , time

Feb. data (in run40/sumup_data/)

Sum????.root

Jun. data (in run43/fiber_merged)

Sum????_fiber.root

MC simulation of KL decays

ID, energy, time

↓

cluster finding

↓

π^0 or K_L reconstruction etc..

CsI data analysis

data from FADC (in kekcc:/hsm/had/koto/)

convert to rootfile

pulse shape information of each FADC crate

Feb. data (in run40/conv_data/crate[0-10])

run????_conv.root

Jun. data (in run43/fiber_out/node[0-10])

run????_conv_crate[0-11]_node[0-9].root

convert (pulse shape info are lost)

ID, energy(integratedADC) , time

Feb. data (in run40/sumup_data/)

Sum????.root

Jun. data (in run43/fiber_merged)

Sum????_fiber.root

MC simulation of KL decays

ID, energy, time

gsim4

(geant4 based MC)

e14lib_dev

cluster finding

π^0 or K_L reconstruction etc..

today's topic

How to install e14lib dev. ver.

- **install to your own PC (for bash)**

  0, Prepare geant4, ROOT and clhep.

  1, In the directory where you want to install the library,

  svn co https://justice.hep.sci.osaka-u.ac.jp/svn/e14lib/trunk/e14

  (you'll be asked username and password. user: koto, passwd:[usual one])

  2, add follows to your .bashrc.

  export E14_TOP_DIR=**foo/bar**/e14 **(where you installed e14lib)**

  source $E14_TOP_DIR/scripts/e14.sh >& /dev/null

  export PATH=$E14_TOP_DIR/bin:$PATH

  Comment out here at first time to see error message. If any messages, follow it.

  3, compile the library.

  source ~/.bashrc

  cd $E14_TOP_DIR

  make install-headers (←Please ignore error massages in this stage.)

  make

for KEKCC users

- **for KEKCC user (bash)**

  The e14lib for common use is already prepared in

  /sw/koto/e14lib/dev/installed20120803/e14/.

  To use it, add follows to ~/Env/bashrc_post

  ```
  # geant4, root, clhep setting
    source /sw/koto/e14lib/dev/installed20120803/requiredEnvironment
  # e14lib dev. version setting
    source /sw/koto/e14lib/dev/installed20120803/setup.sh  >& /dev/null
  ```

(In the file "requiredEnvironment", proper environments of g4,root and clhep are set. Don't change those environments after this.)

* Also, if you want to prepare the private e14lib in KEKCC for some reasons, you can use the installation procedure in the previous page.

e14 analysis tools

- We already have a lot of know-hows for various analysis which were accumulated through e391a and MC study .
  ⇒ some of them are available in e14lib developing version.

source code: $E14_TOP_DIR/sources/ana/rec/
provide ...

- cluster finding → class ClusterFinder
- gamma identification → class GammaFinder
- pi0 or KL reconstruction → class Rec2g , RecKlong
- energy correction → class E14GNAnaFunction
- position correction → class E14GNAnaFunction
- shower shape cut → class ShapeChi2, ShapeANN

examples

example program : $E14_TOP_DIR/examples/e14gnana/

- e14clustering -- clustering from MC

bin/e14clustering [input file(MC data)] [output file]

- e14g2ana -- pi0 reconstruction for KLpi0nunu event

bin/e14g2ana [input file(clustering result)] [output file]

- e14g4ana -- KL reconstruction for KLpi0pi0 event

bin/e14g4ana [input file(clustering result)] [output file]

- e14g6ana -- KL reconstruction for KL3pi0 event

bin/e14g4ana [input file(clustering result)] [output file]

→**see backup slides for how to use these programs**

flow of analysis
1.cluster finding

- To measure gamma energy using CsI calorimeter, we should make "cluster".

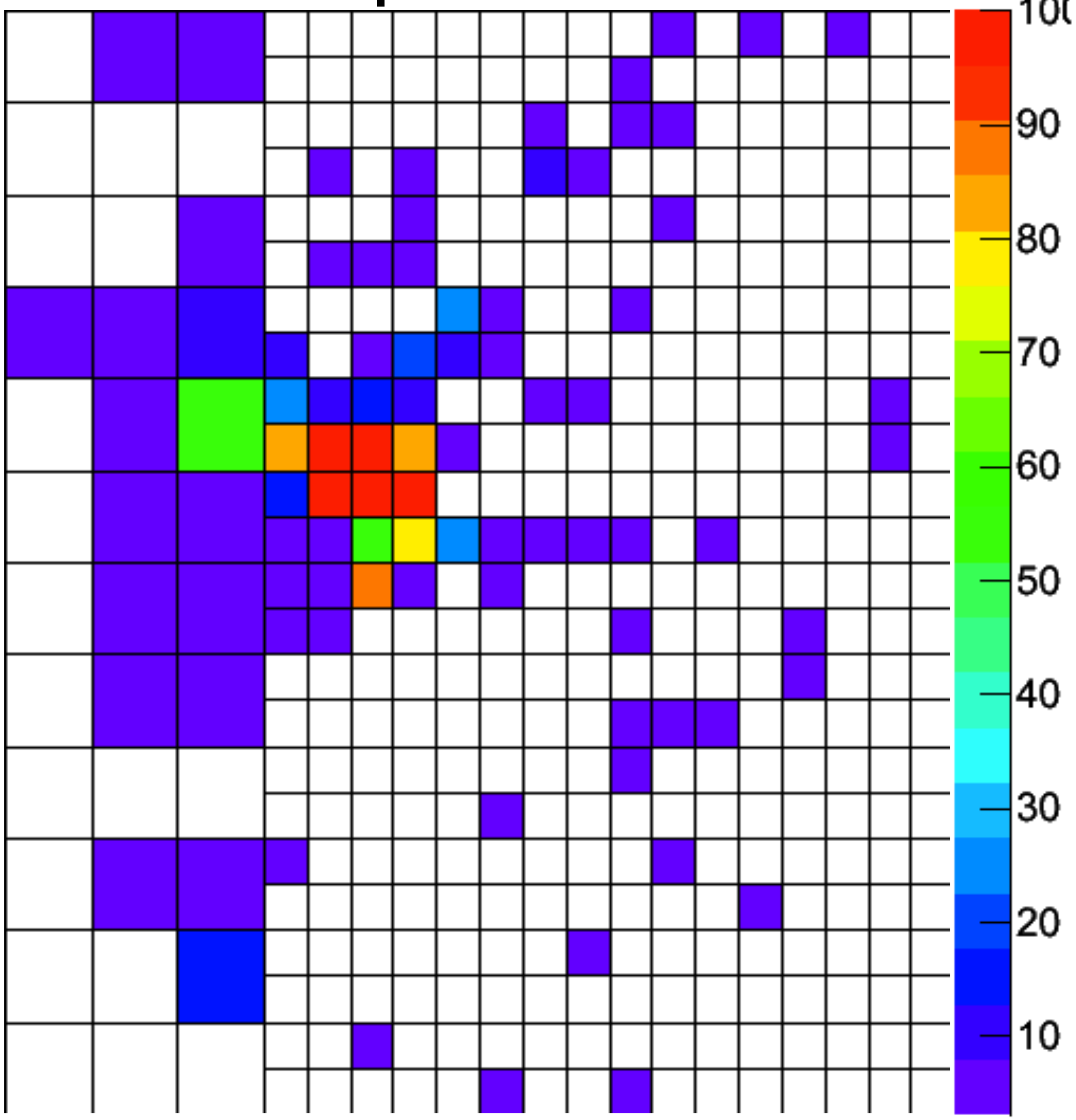
- cluster = a group of CsI crystals with energy deposit exceeding a threshold.

- class "ClusterFinder" provides the function for clustering

std::list<Cluster> findCluster(int *nCsi*,int **ID*,double **E*,double **time*)

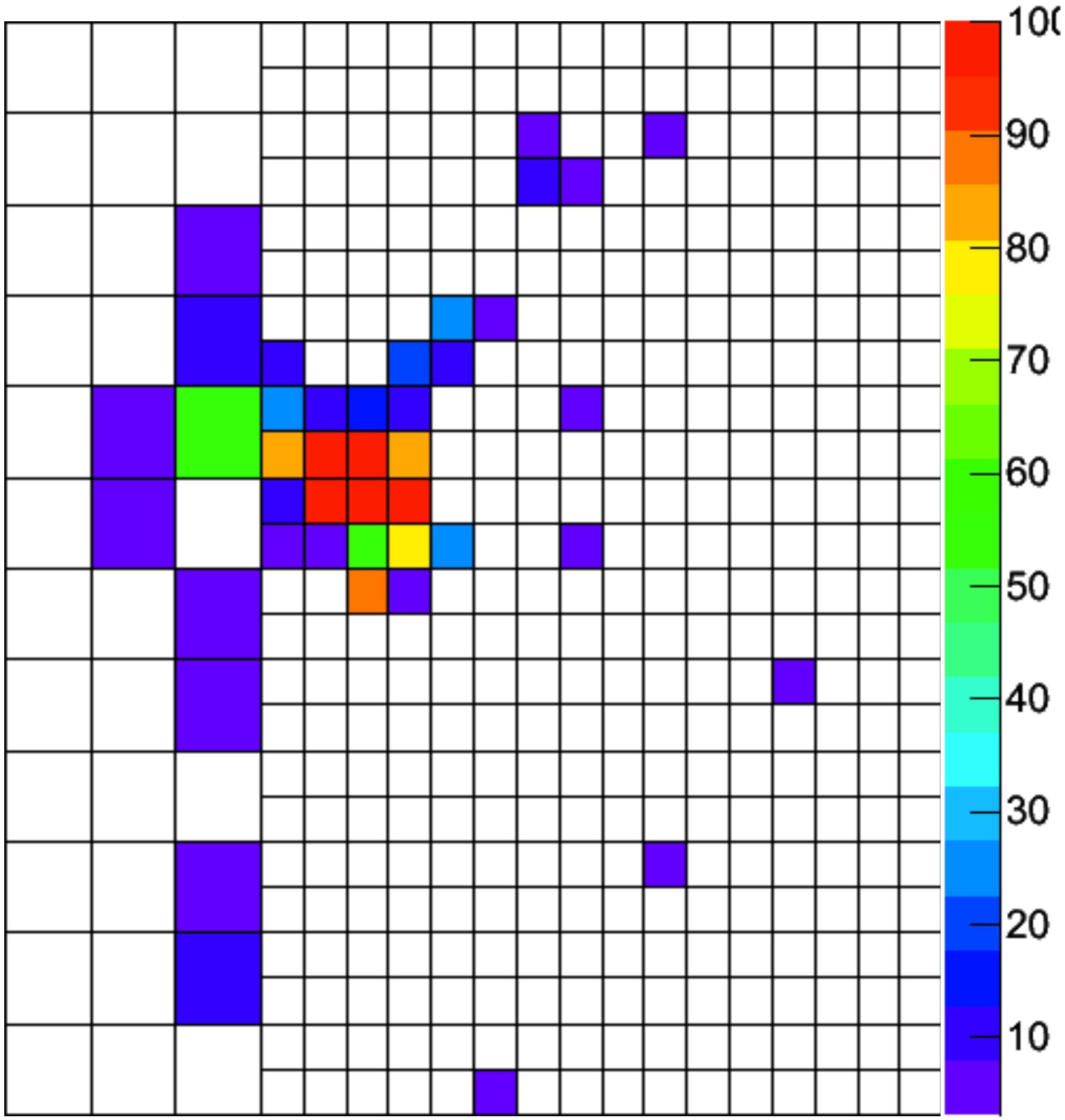
finding clusters

E deposit on CsI



consider the case like left figure.

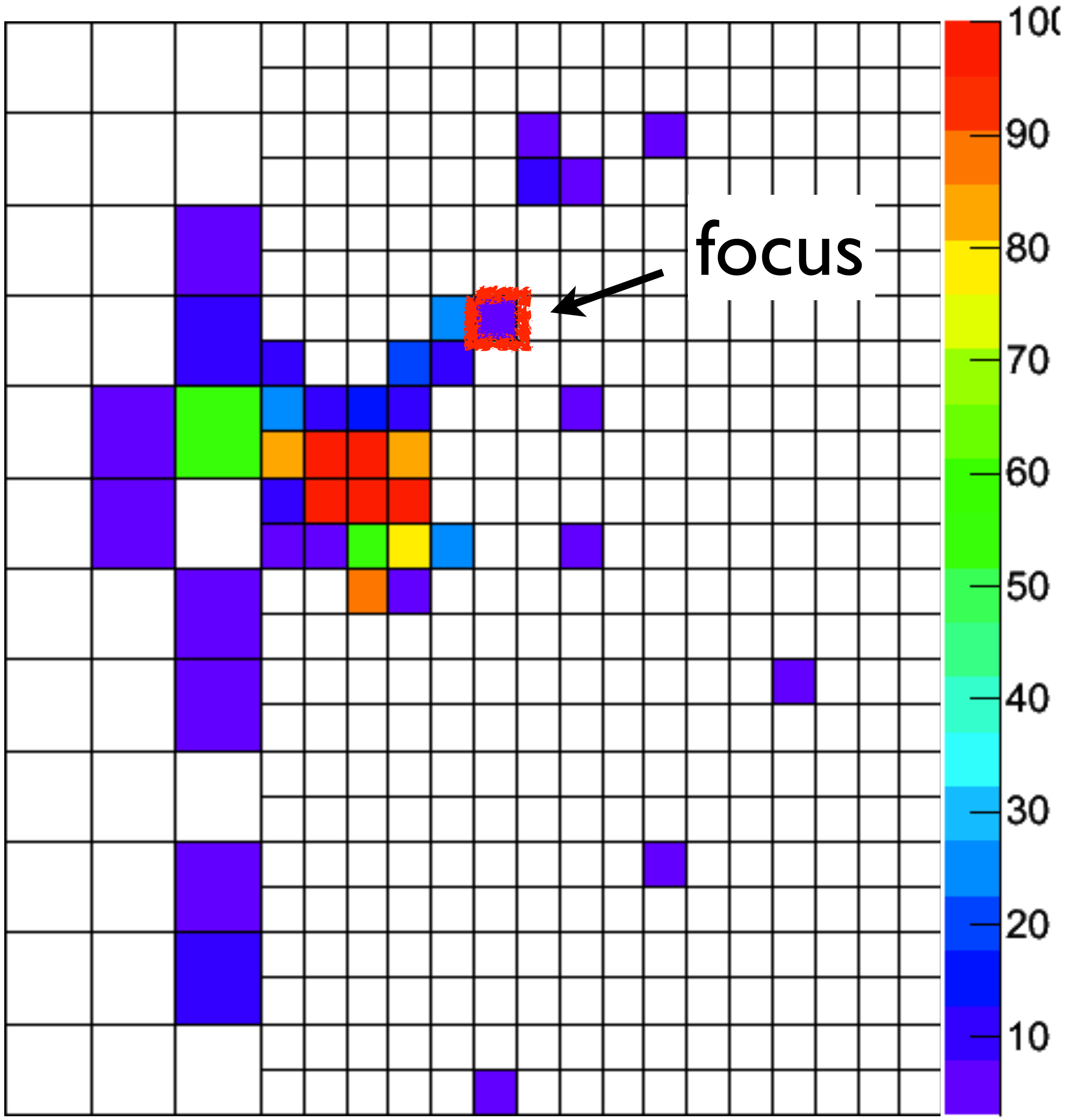
finding clusters



1,Set threshold (default = 3MeV)

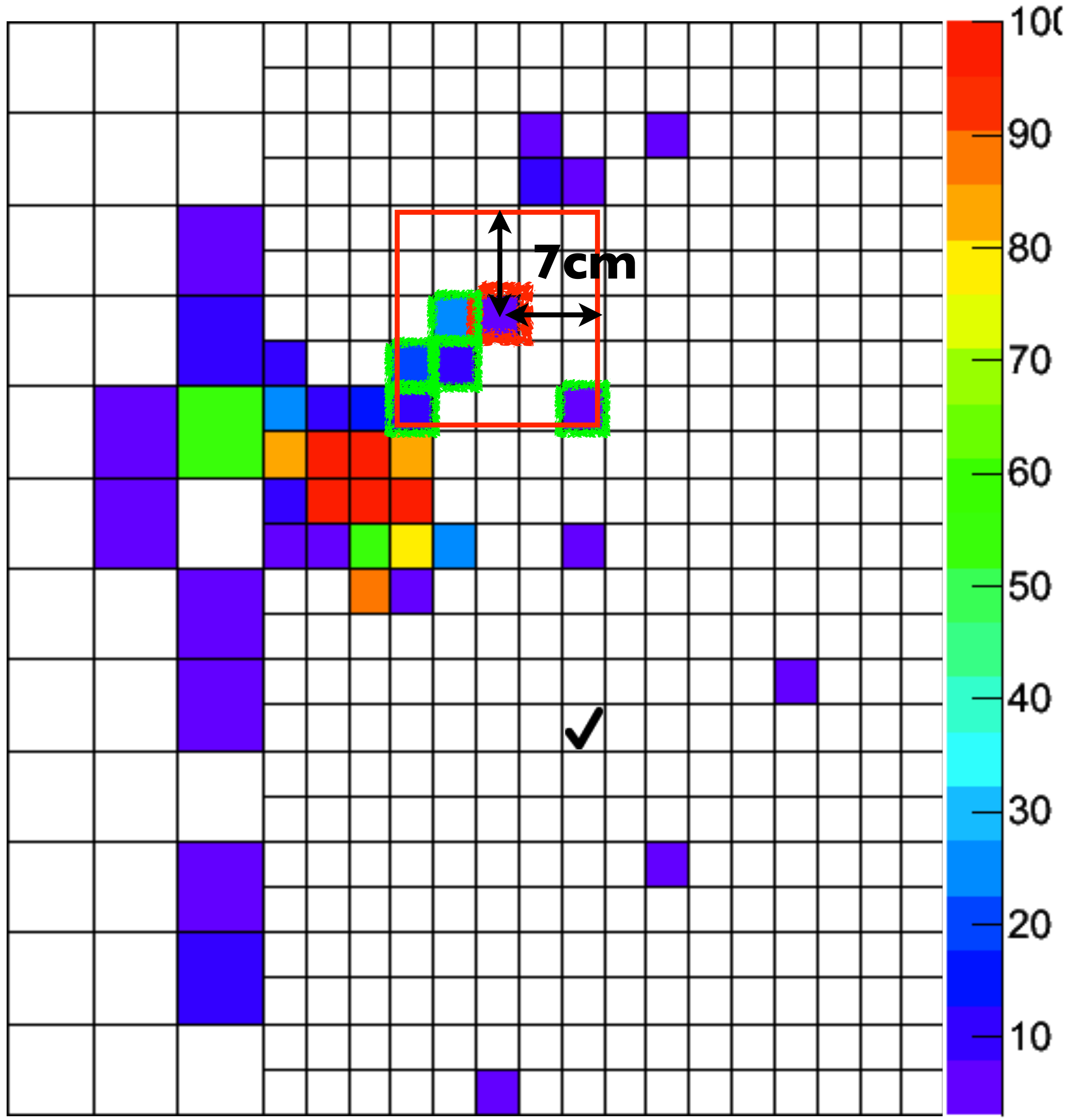
Ignore crystals whose energy deposit is less than threshold.

finding clusters



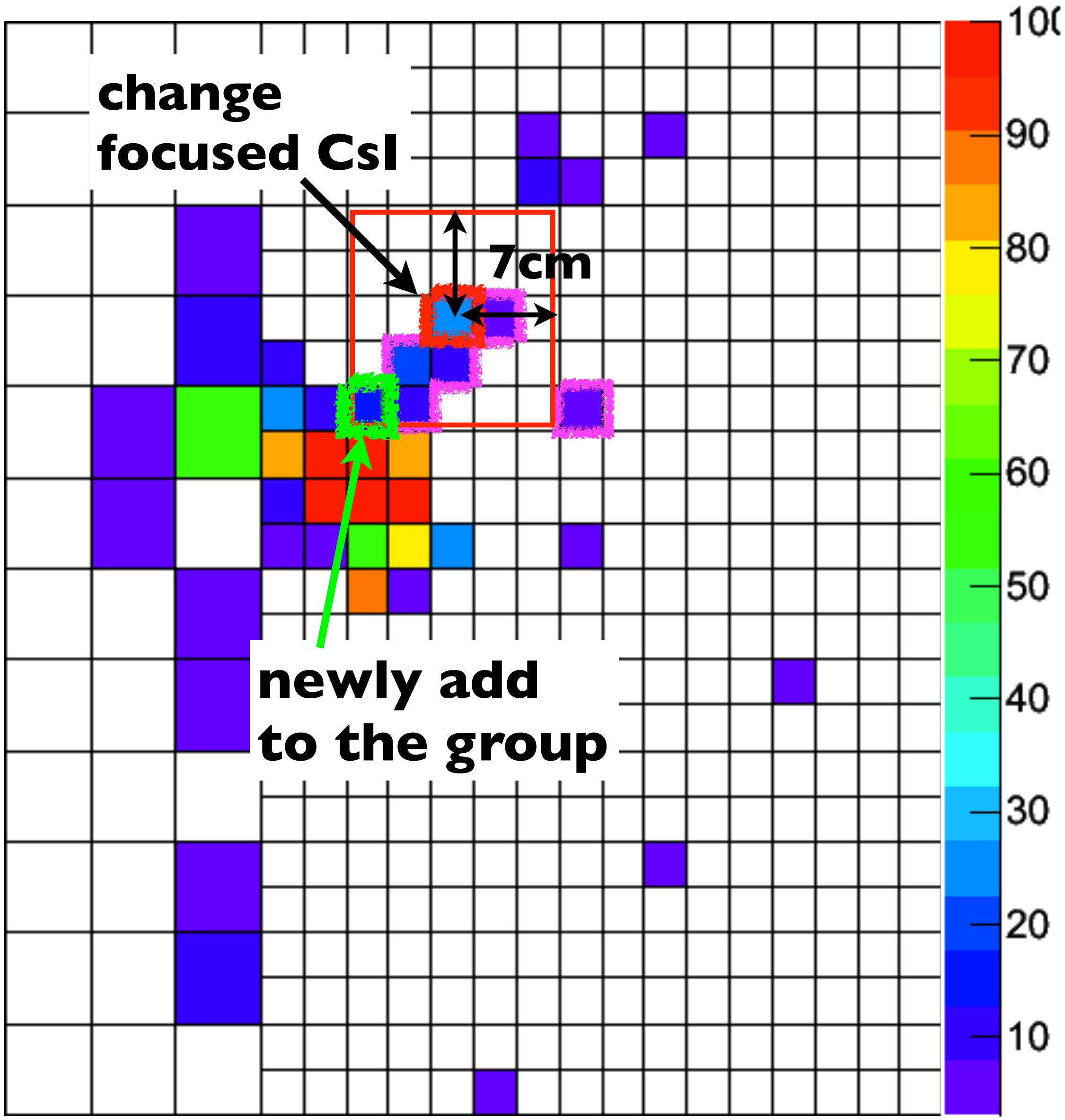
2, focus on one crystal.

finding clusters



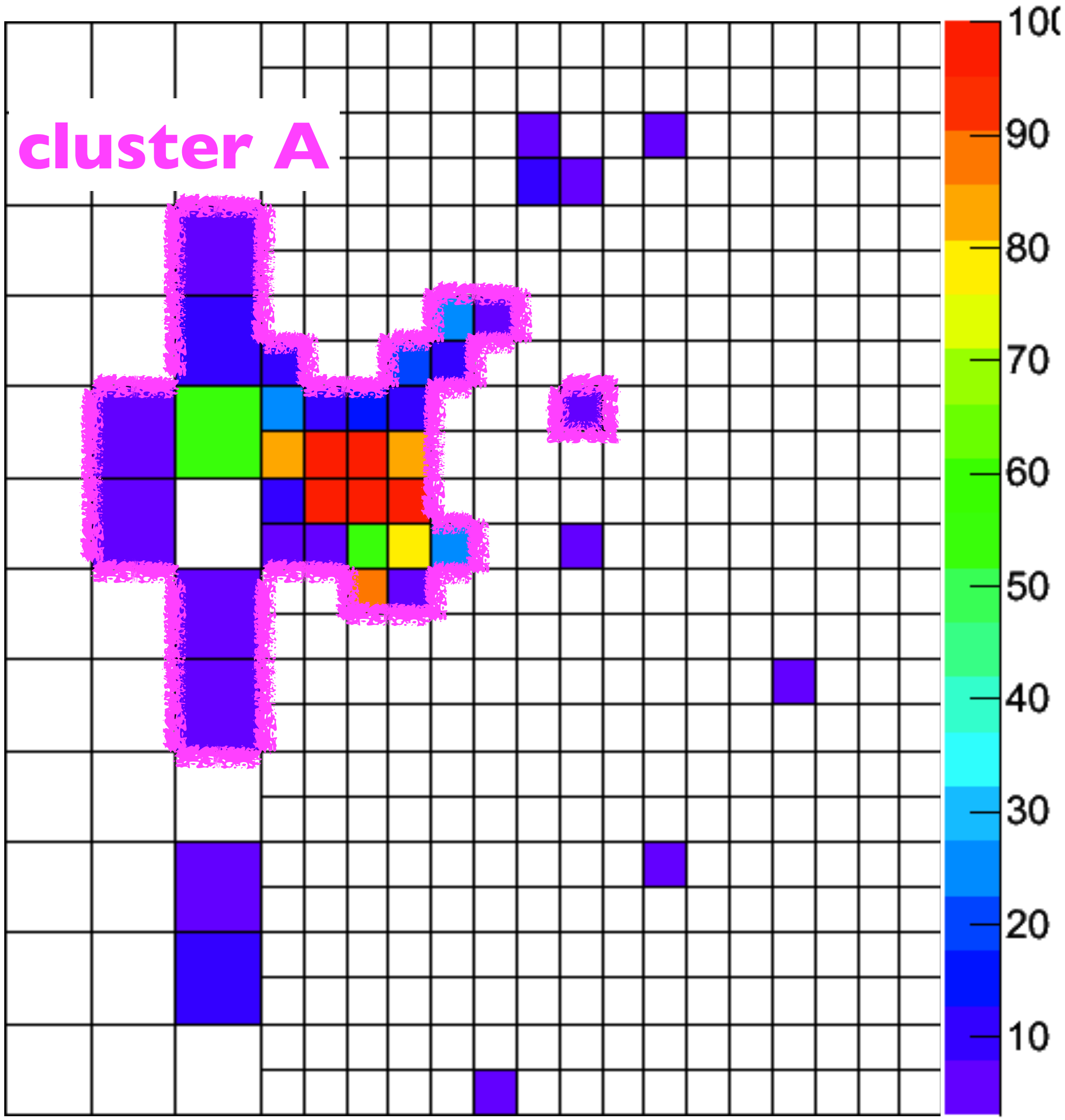
3, group crystals of which center positions are less than 7cm apart from the center of focused crystal

finding clusters



4, focus on another crystal in the same group and repeat process 3.

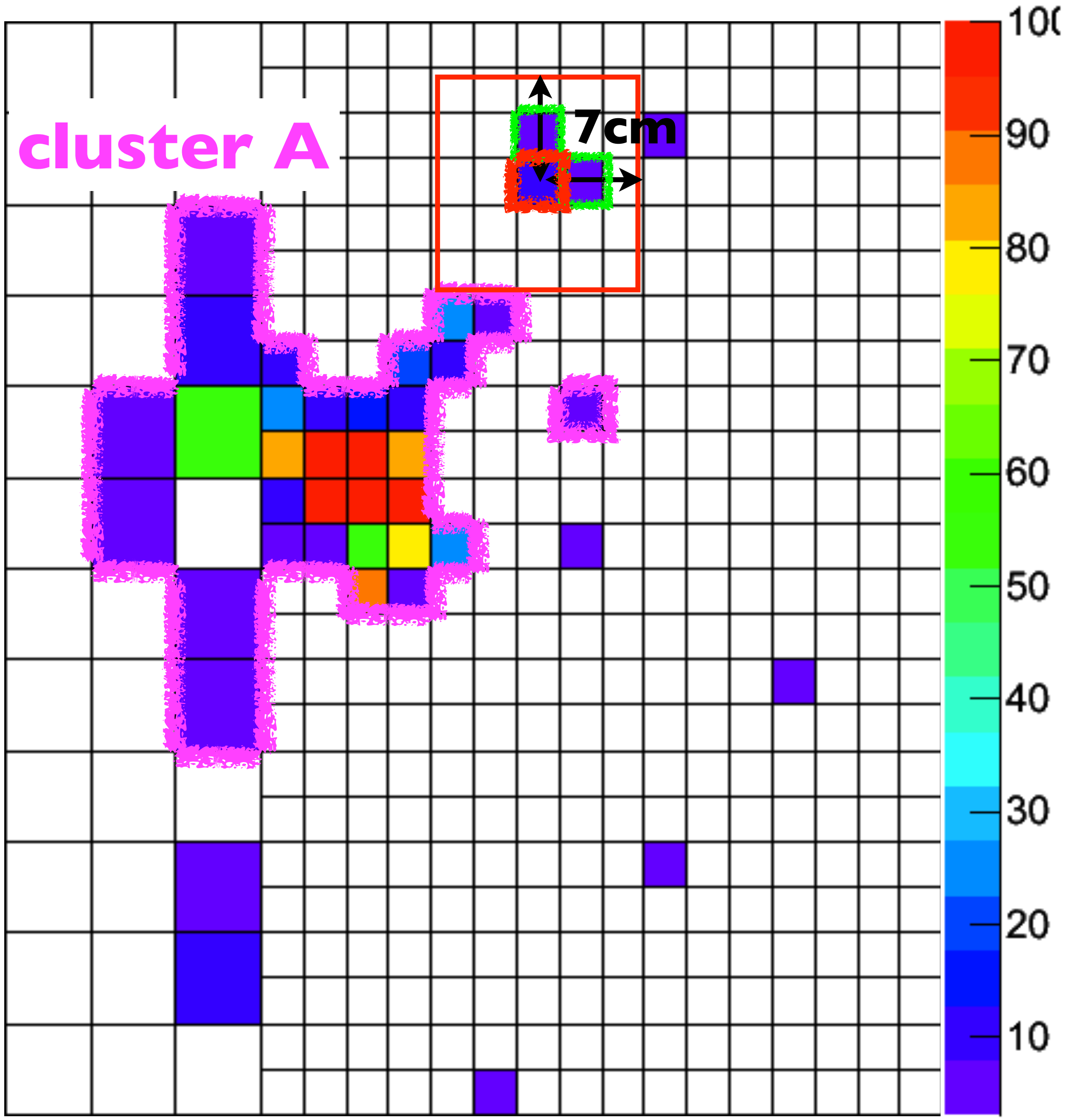
finding clusters



4, focus on another crystal in the same group and repeat process 3.

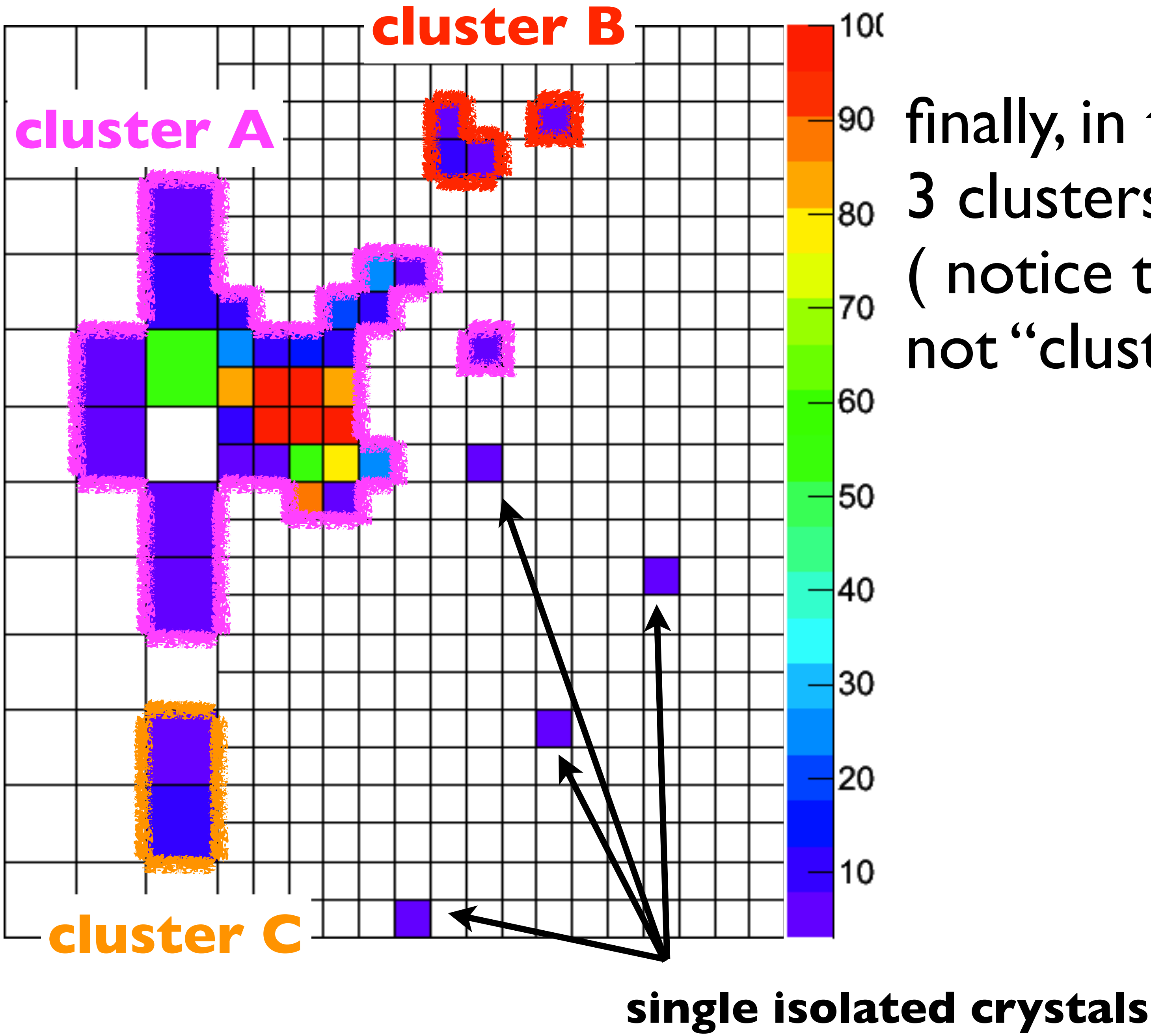
→run over until there is no crystal to add to the group.

finding clusters



5, focus on a remaining crystal and repeat the procedure above until all crystals are used.

finding clusters



finally, in this case,
3 clusters were found.
(notice that single hits are
not "cluster")

energy,position,time of cluster

following cluster informations are calculated

- sum of energy deposit
- position of center-of-energy

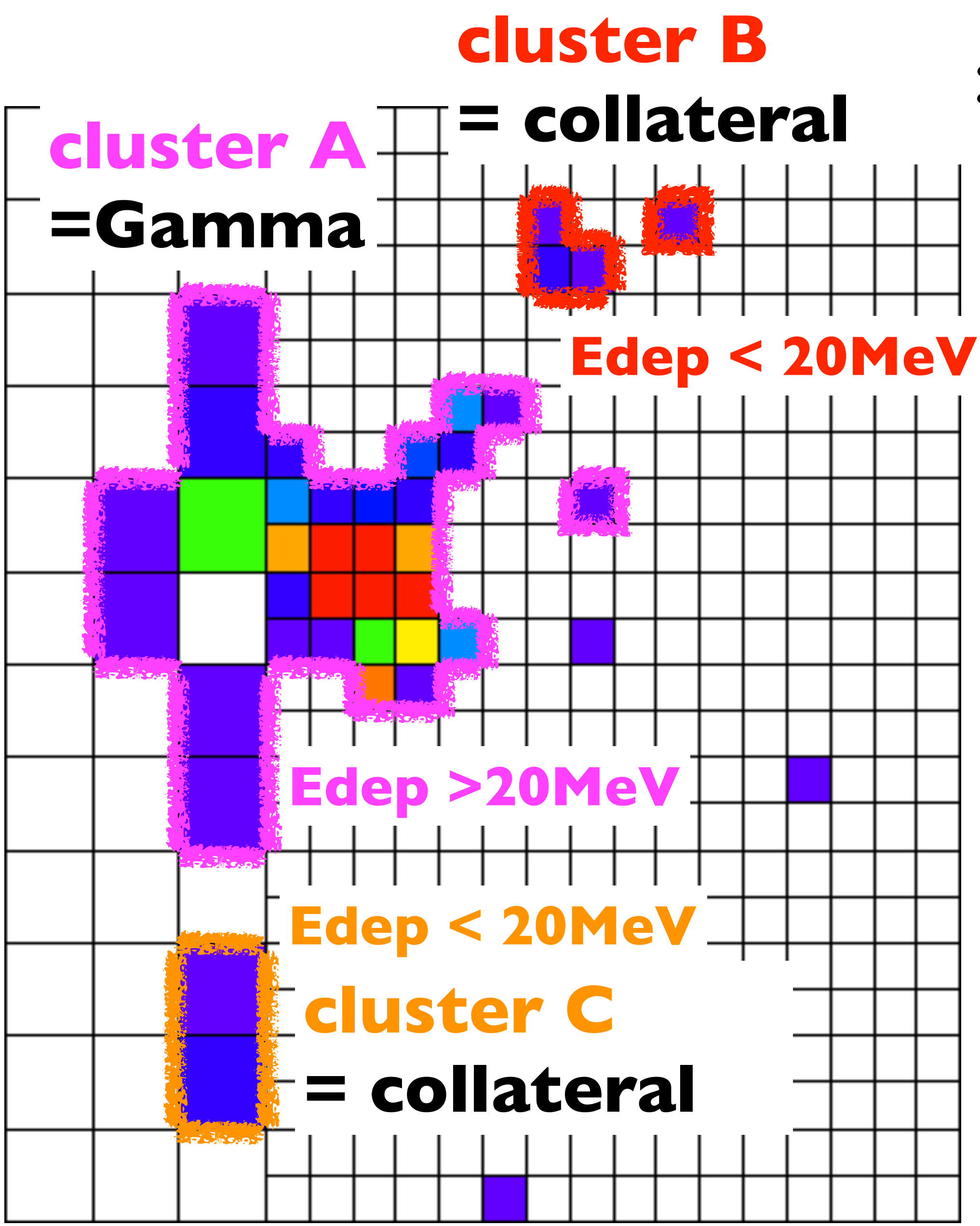
$$x_c = \sum_i^n \frac{(E_i \times x_i)}{\sum_i E_i}$$

x_i = position of each CsI
E_i = energy deposit each CsI

$$y_c = \sum_i^n \frac{(E_i \times y_i)}{\sum_i E_i}$$

- timing ← Cluster timing is not studied sufficiently. Now I temporarily defined cluster time as earliest time of CsI hits.

flow of analysis



*2. gamma finding

- gamma finding = to identify clusters as gamma cluster or not
- require energy deposit > 20 MeV as a gamma cluster

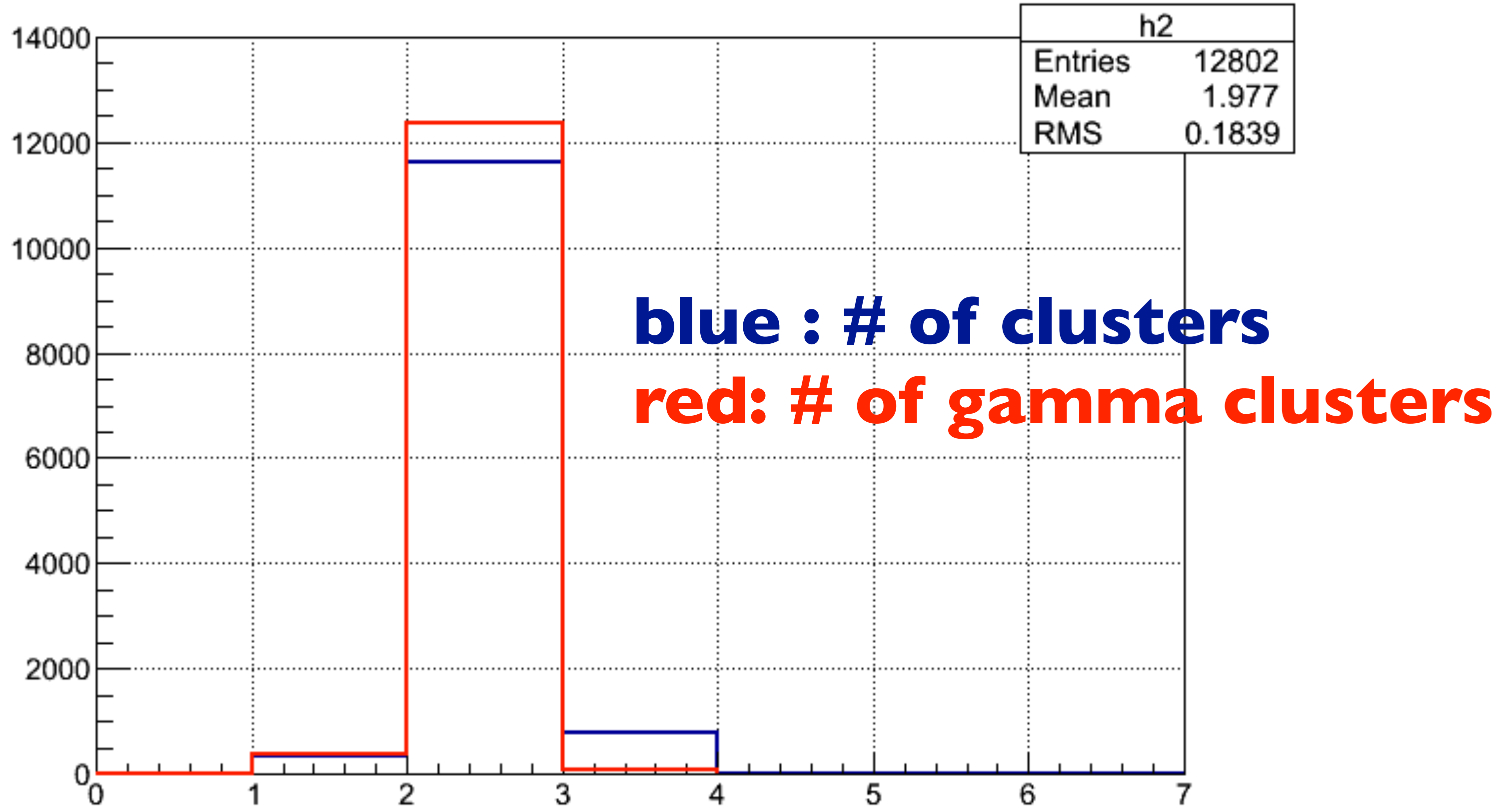
note: "gamma cluster" means "not a collateral cluster but a cluster made by particle". I don't mean that "gamma cluster" is made by gamma.

class "GammaFinder" provide the Function for this.

void findGamma(std::list<Cluster> const &*clist*, std::list<Gamma> *glist*)

KLpi0nunu MC

(events where 2 gammas hit on CsI fiducial)

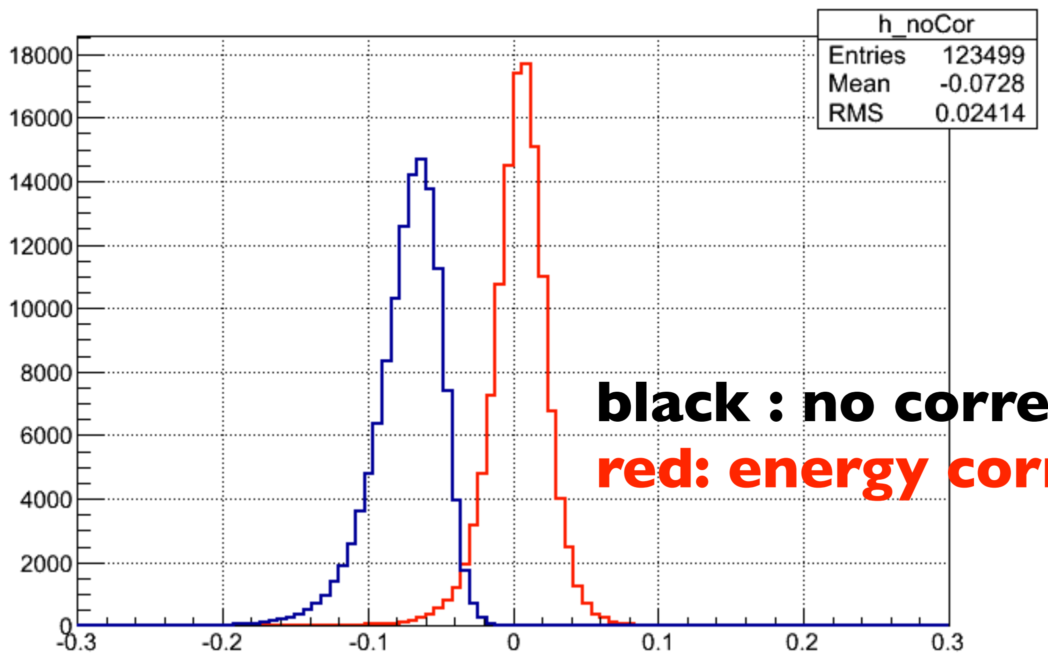


☆90.9% accepted→96.6%

Now I just require very simple condition (E deposit > 20 MeV).
One who analyzes the data should study and develop this procedure.

flow of analysis
3. energy correction (1st)

energy difference of
gamma(E_{gamma}) &cluster($E_{cluster}$)
(KLpi0nunu event MC)



black : no correction
red: energy correction

($E_{cluster}$- E_{gamma})/E_{gamma}

- correct cluster energy

(some energies of incident gamma leaked out of cluster)

$$E_{gamma} =(1+f(E_{cluster})) \times E_{cluster}$$

this process is also done in GammaFinder::findGamma()

void findGamma(std::list<Cluster> const &*clist*, std::list<Gamma> *glist*)

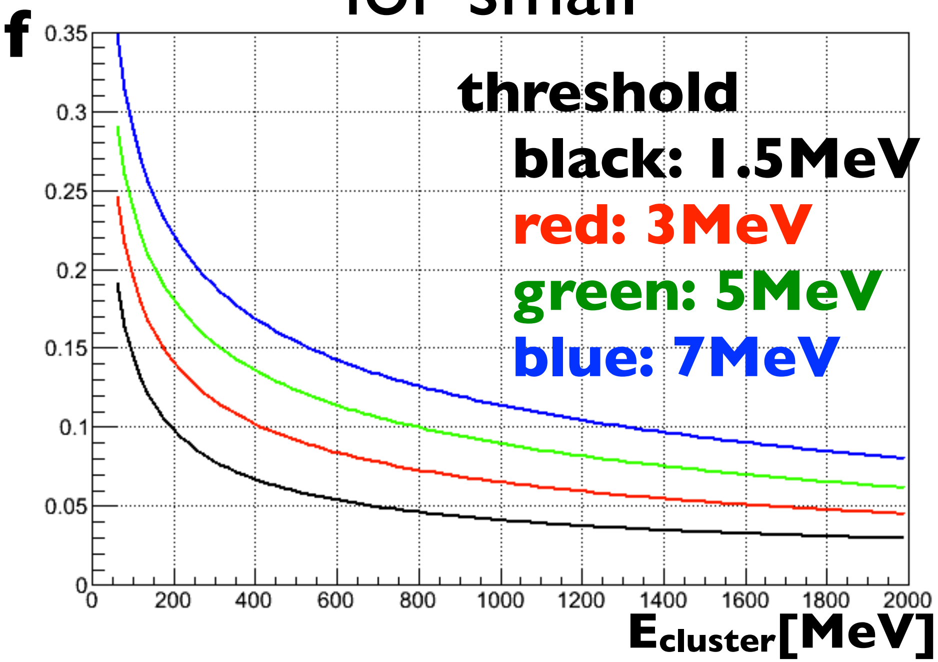
correction function

- correction function **f** was derived from KLpi0nunu MC.
- fraction of energy leakage strongly depends on energy and threshold.

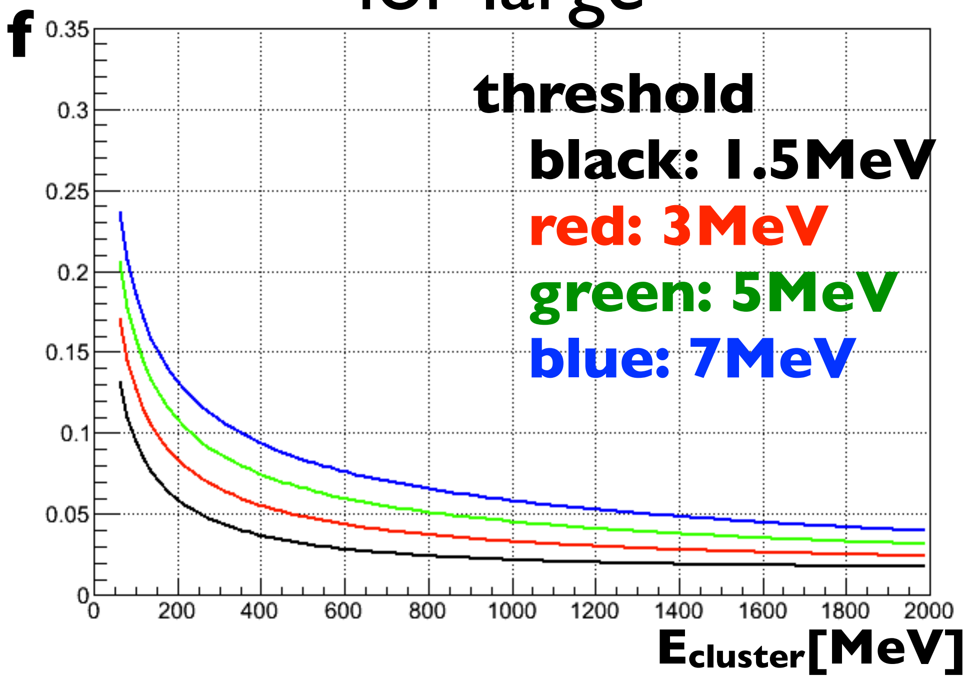
→ f = f($E_{cluster}$, threshold)

- f are prepared for small and large crystals respectively. Energy weighted mean of 2 functions are used for clusters which include both small and large crystals
- f = 0.014+0.052/sqrt(E[GeV])-0.0074 log(E[GeV]) [Small, thre=3MeV]

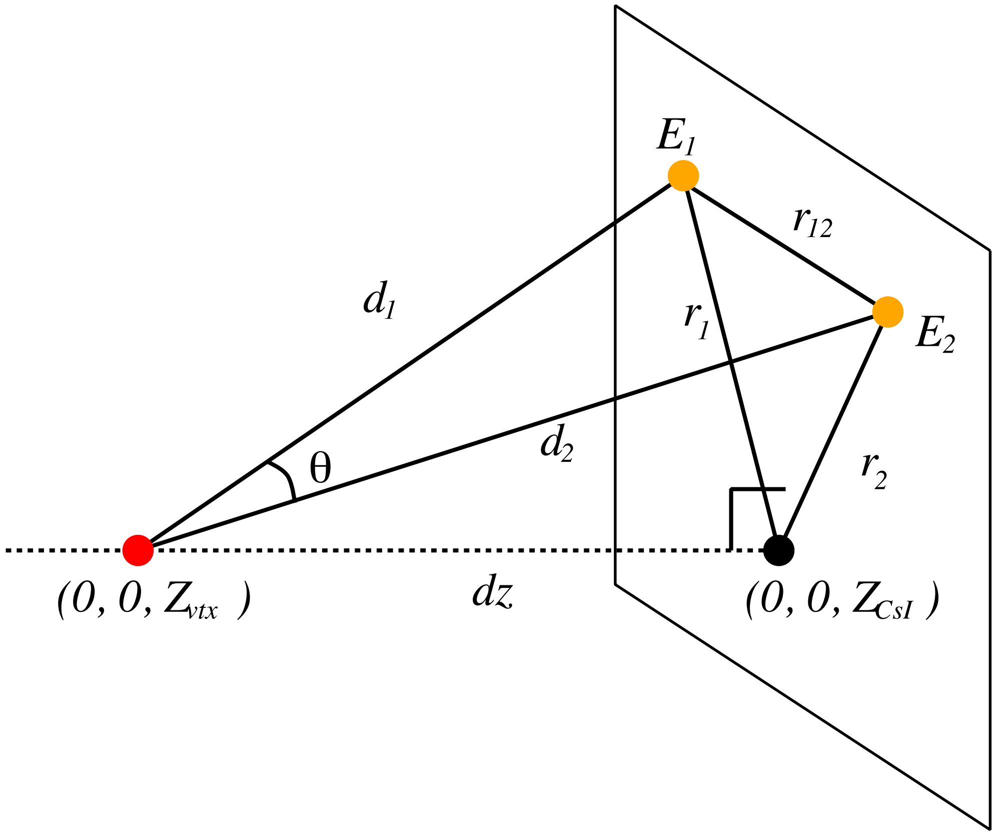
for small



for large



flow of analysis
4-1. reconstruction pi0



- for the events where 2 gamma clusters on CsI calorimeter.

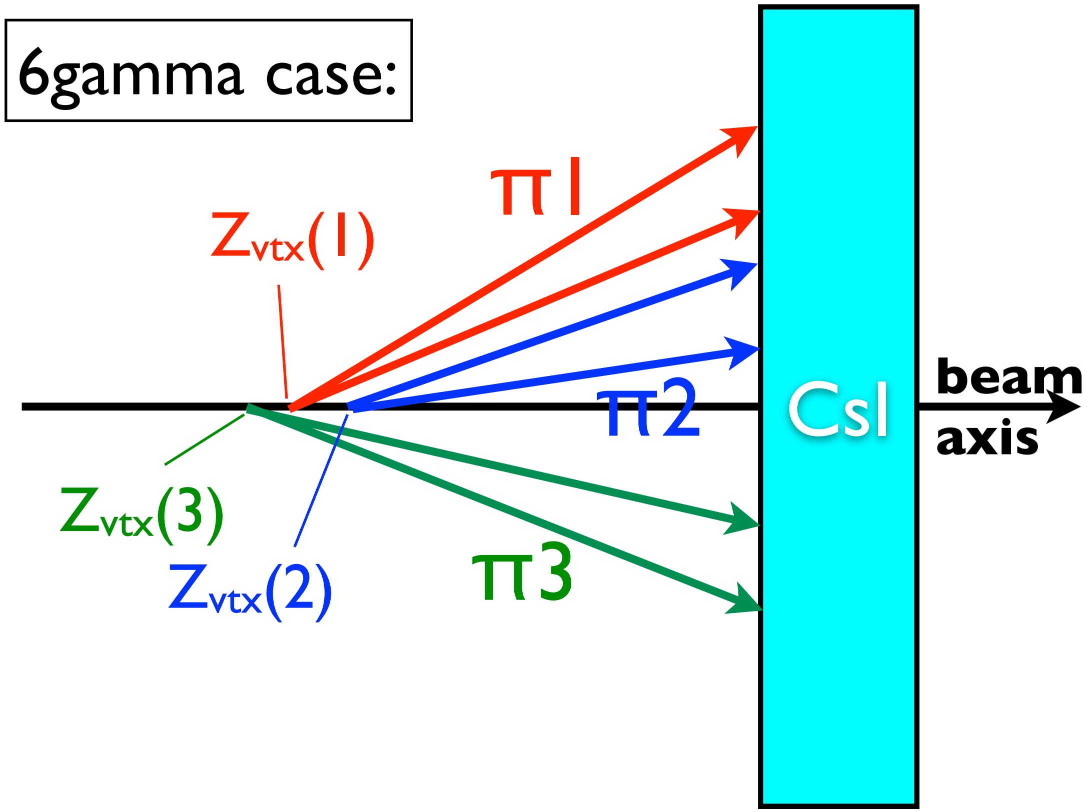
- assume that 2 gamma on CsI were generated from pi0
 then,

$$cos\theta = 1 - \frac{M_{\pi^0}^2}{2E_1E_2}\ .$$

- assume the vertex is on beam axis
→ can calculate Z_{vtx} as left figure

class Rec2g provide the function for pi0 reconstruction
std::list<Pi0> recPi0withConstM(std::list<Gamma> glist, const double M=Mpi0)

flow of analysis
4-2. reconstruction KL



- for the events where 4 or 6 gamma clusters on CsI.
- pions are reconstructed using a method in previous page.
  → Z_{KL} = mean of Z_π

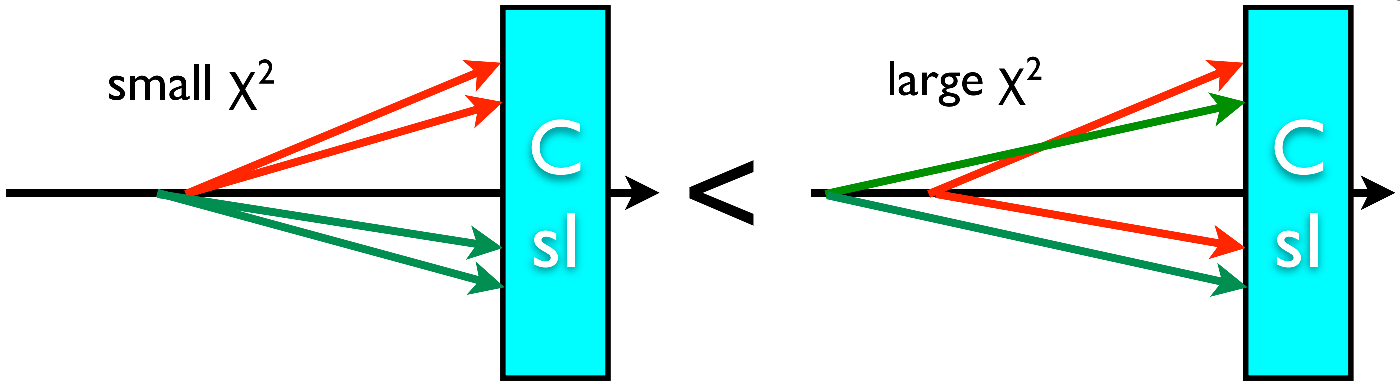
$$Z_{K_L} = \frac{\sum_{i=1}^{3} [Z_{\mathrm{vtx}}(i)/\sigma_{\mathrm{vtx}}(i)^2]}{\sum_{i=1}^{3} [1/\sigma_{\mathrm{vtx}}(i)^2]}$$

- KL energy = sum of gamma energies

$$E_{K_L} = \sum_{i=1}^{6} E_\gamma(i)$$

- combination of gammas is decided by chi square

$$\chi^2 = \sum_{i=1}^{3} \frac{[Z_{K_L} - Z_{\rm vtx}(i)]^2}{\sigma_{\rm vtx}(i)}$$



- KL vertex x,y
= center-of-gravity of gammas

$$\begin{pmatrix} X_{K_L} \\ Y_{K_L} \\ Z_{K_L} \end{pmatrix} = \begin{pmatrix} s \times \sum_{i=1}^{6} E_\gamma X_\gamma(i)/E_{K_L} \\ s \times \sum_{i=1}^{6} E_\gamma Y_\gamma(i)/E_{K_L} \\ Z_{K_L} \end{pmatrix}$$

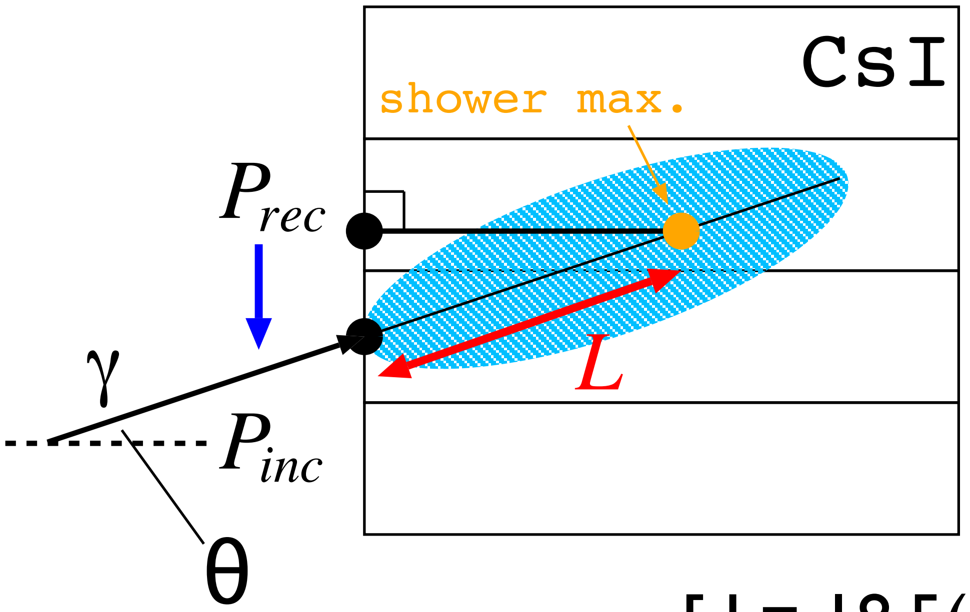
$$s = \frac{Z_{K_L} - Z_{target}}{Z_{CsI} - Z_{target}},$$

class RecKlong provide the function for KL reconstruction

std::vector<Klong> recK2pi0(const std::list<Pi0>& pi0list)

std::vector<Klong> recK3pi0(const std::list<Pi0>& pi0list)

flow of analysis
5. position correction



- correct cluster position
- center-of-energy position (P_{rec}) is different from gamma incident position (P_{inc}) due to shower length

$$P_{inc} = P_{rec}\ (1\text{-}L\sin\theta)$$

[L= 18.5(6.49+0.992log(E[GeV]))mm from MC]

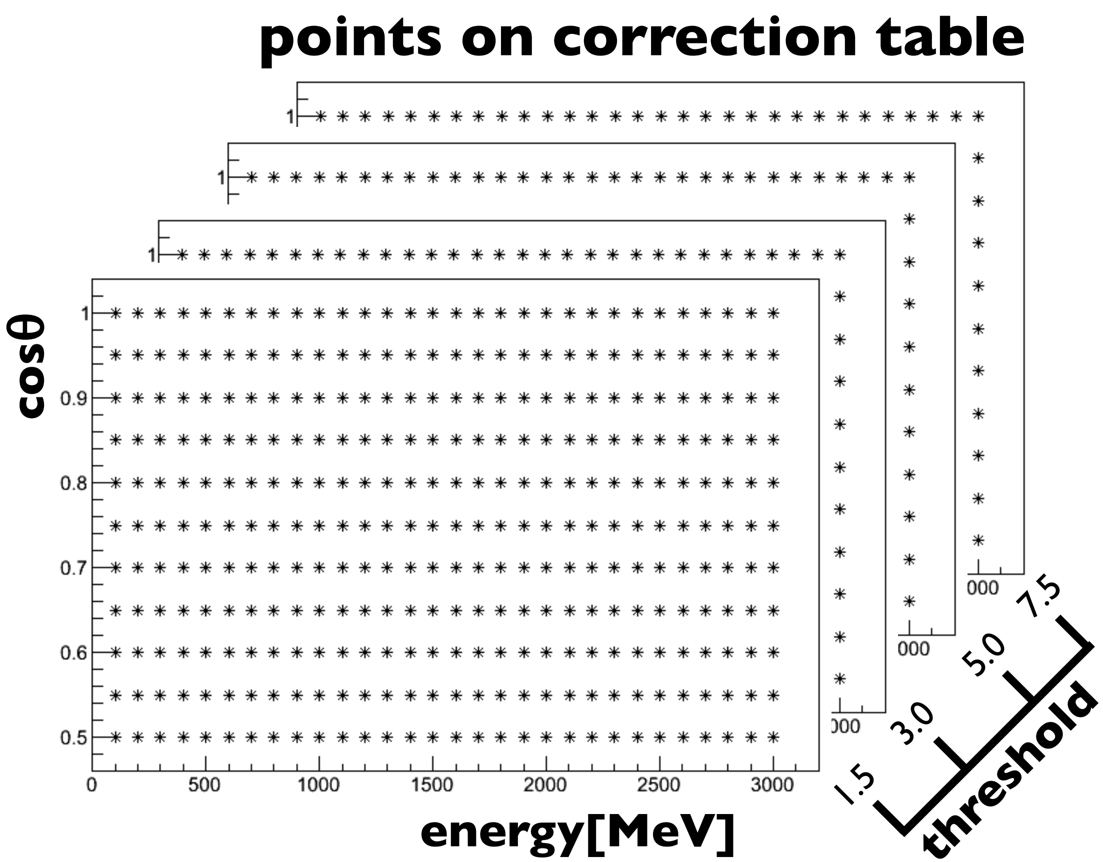
the function is defined in class E14GNAnaFunction

void correctPosition(Gamma& gamma)

Note: this can't be done before vertex reconstruction because angle of gamma is needed.

flow of analysis
6. energy correction (2nd)

- energy correction including angle dependence.



f = f(E,cosθ,threshold)
is derived from correction table

☜ each point have value f_i estimated by MC

To calculate f from these point,

- spline interpolation for E
- linear interpolation for cosθ and threshold

the function is defined in class E14GNAnaFunction

void correctEnergyWithAngle(Gamma& gamma)

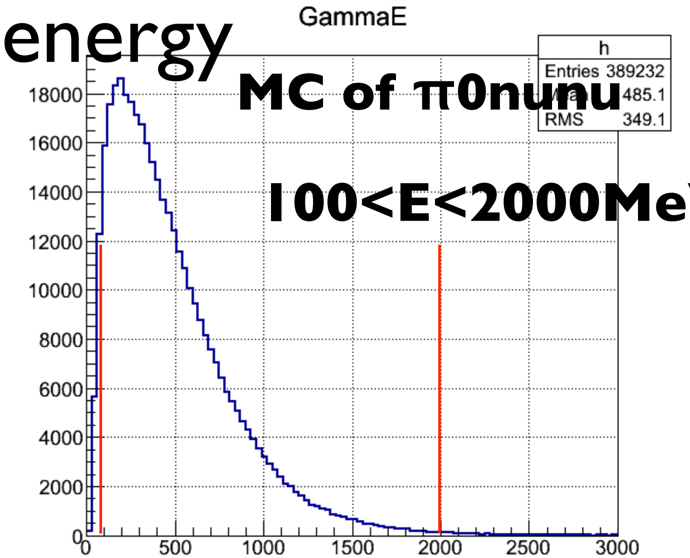
flow of analysis
7. re-reconstruction

- repeat $\pi^0(K_L)$ reconstruction again with corrected positions and energies.

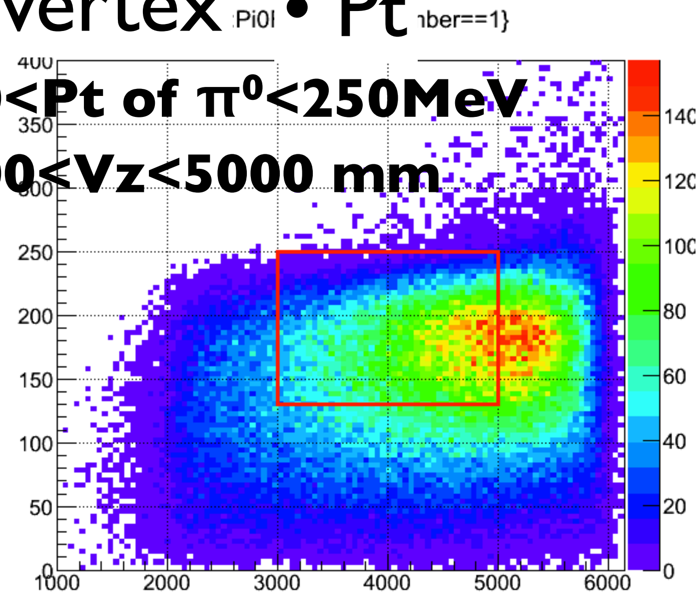
you can find some examples in
$E14_TOP_DIR/examples/e14gnana/e14g?ana/src/user_rec.cc

flow of analysis
*8. kinetic cut

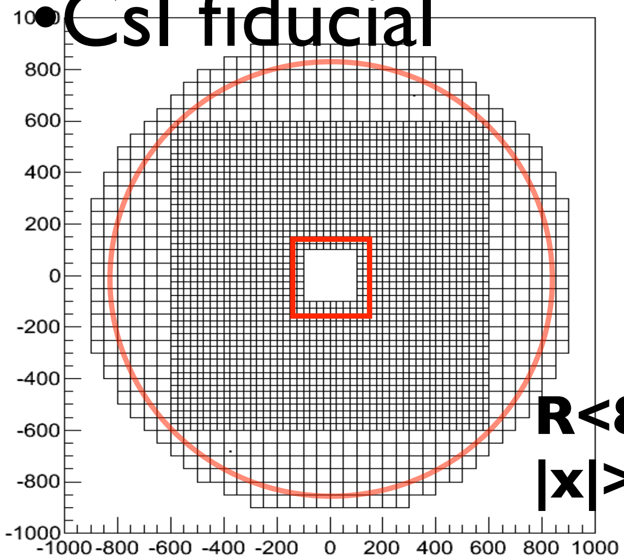
- kinetic cuts for KLpi0nunu (see KOTO proposal for detail)
- energy



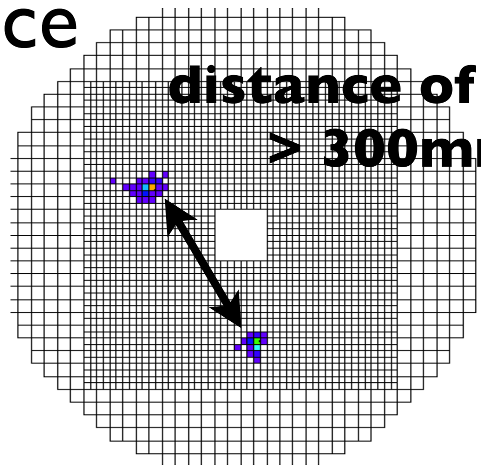
- vertex • Pt



- CsI fiducial

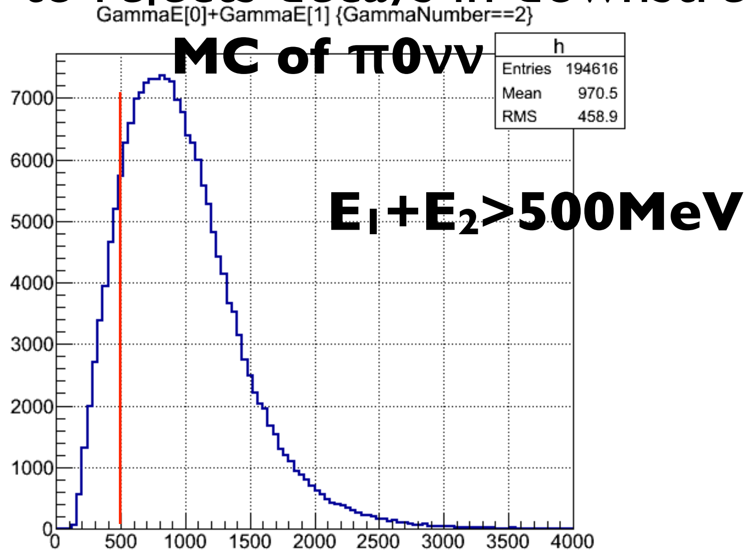


- distance

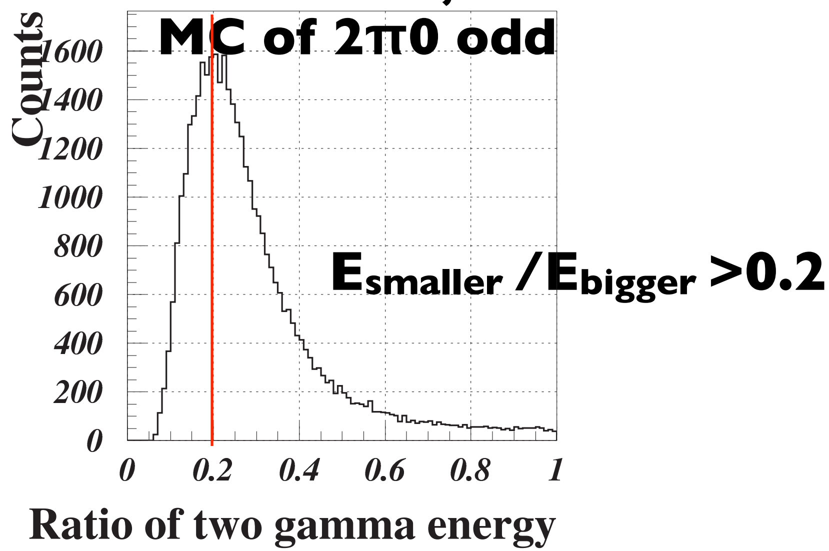


•sum of energy

to rejects decays in downstream

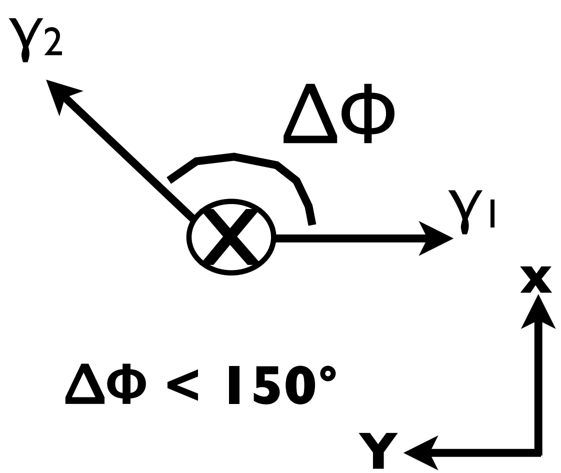


• E ratio to reject $2\pi^0$ odd

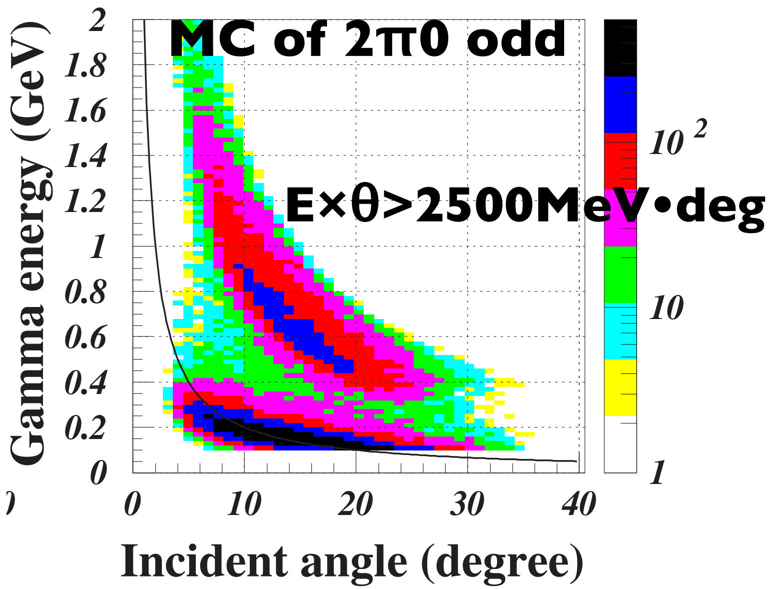


•collinear

to reject $K \rightarrow 2\gamma$

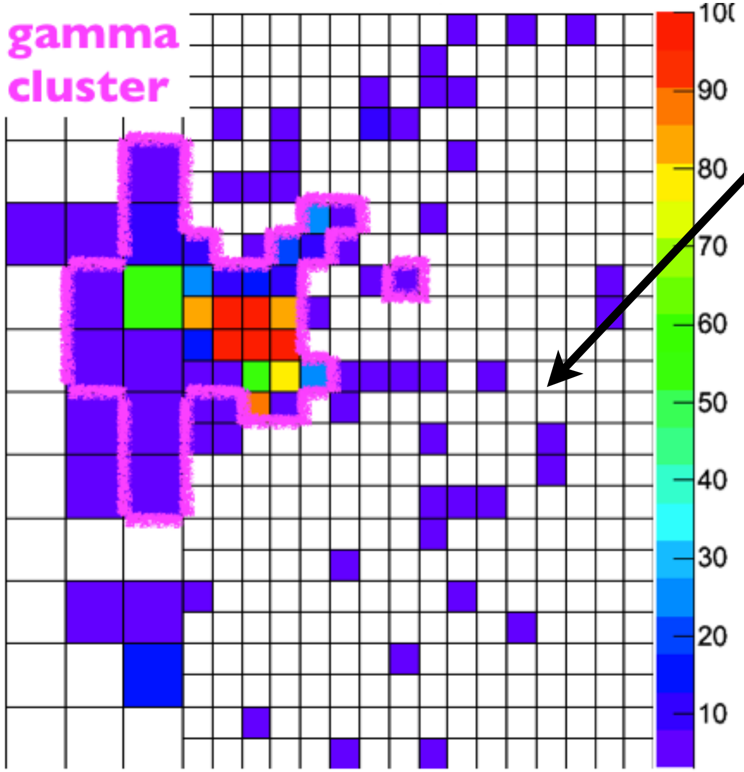


•E-θ to reject $2\pi^0$ odd

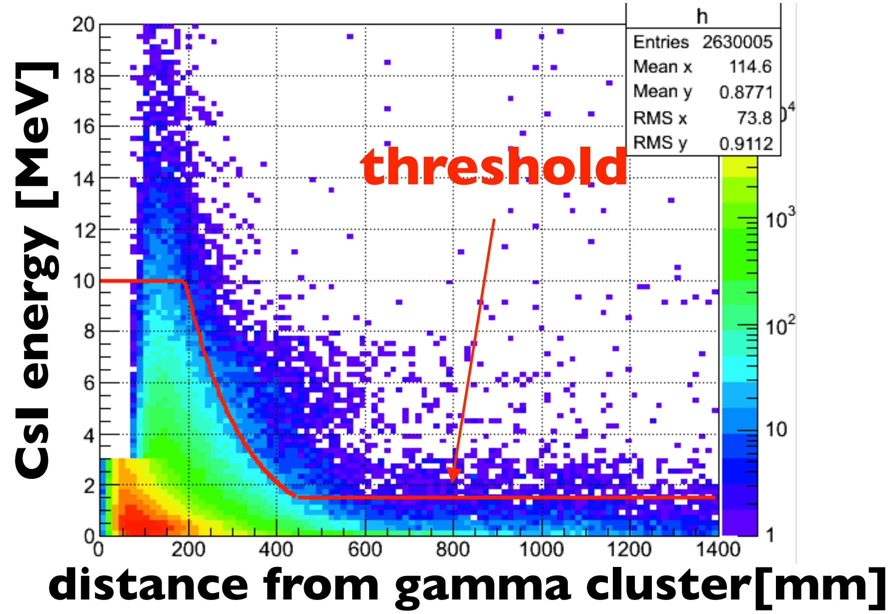


you can find a example of how to apply these cuts in
$E14_TOP_DIR/examples/e14gnana/e14g2ana/src/user_cut.cc

flow of analysis
*9. CsI veto cut



CsI hits that were not included in gamma clusters should have energy less than a threshold.



threshold is a function of distance from gamma cluster (derived from MC, signal acceptance loss = 10%)

you can find some example of how to apply this cut in
$E14_TOP_DIR/examples/e14gnana/e14g2ana/src/user_cut.cc

flow of analysis
*10. veto cut

* not prepared standard cut values for veto detector.
It's responsible for the one who analyze data.

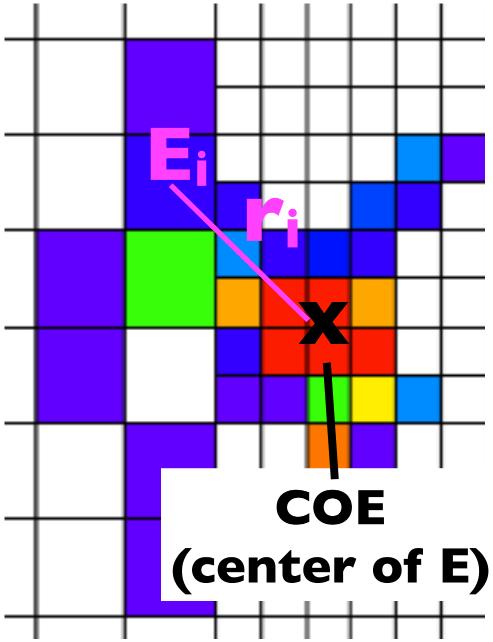
you can find some example in
$E14_TOP_DIR/examples/e14gnana/e14g2ana/src/user_cut.cc

flow of analysis

*11. shower shape cut

- qualify gamma cluster from its shape.

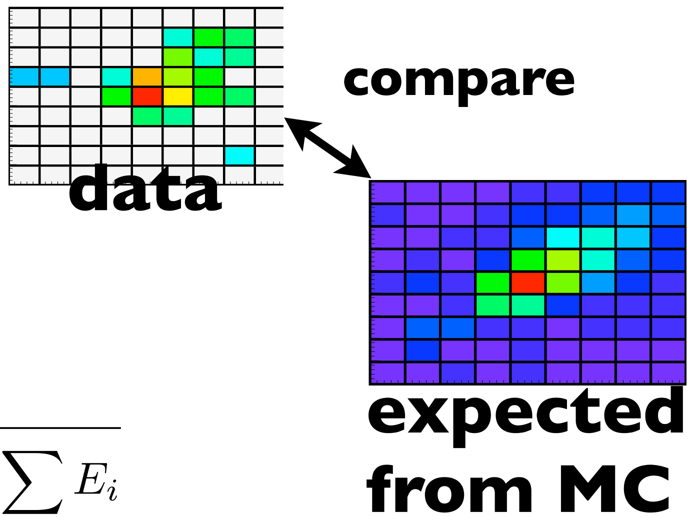
- RMS



$$RMS = \sqrt{\sum_i (E_i \times (r_i - COE)^2 / \sum_i E_i}$$

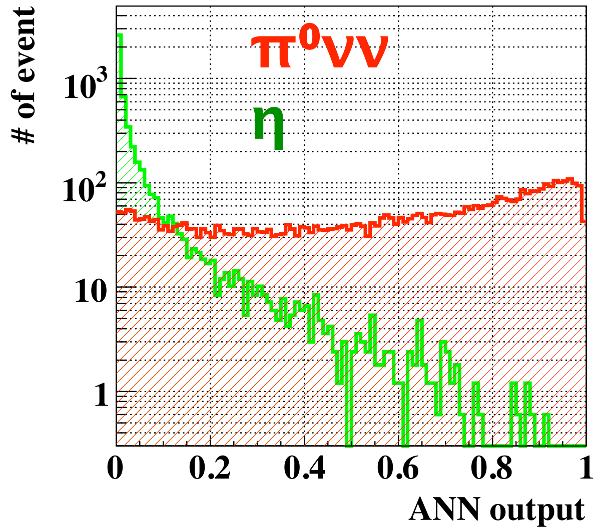
- shapeχ^2

compare with MC expection



- ANN

neural net to distinguish η events



RMS : automatically calculated in cluster finding

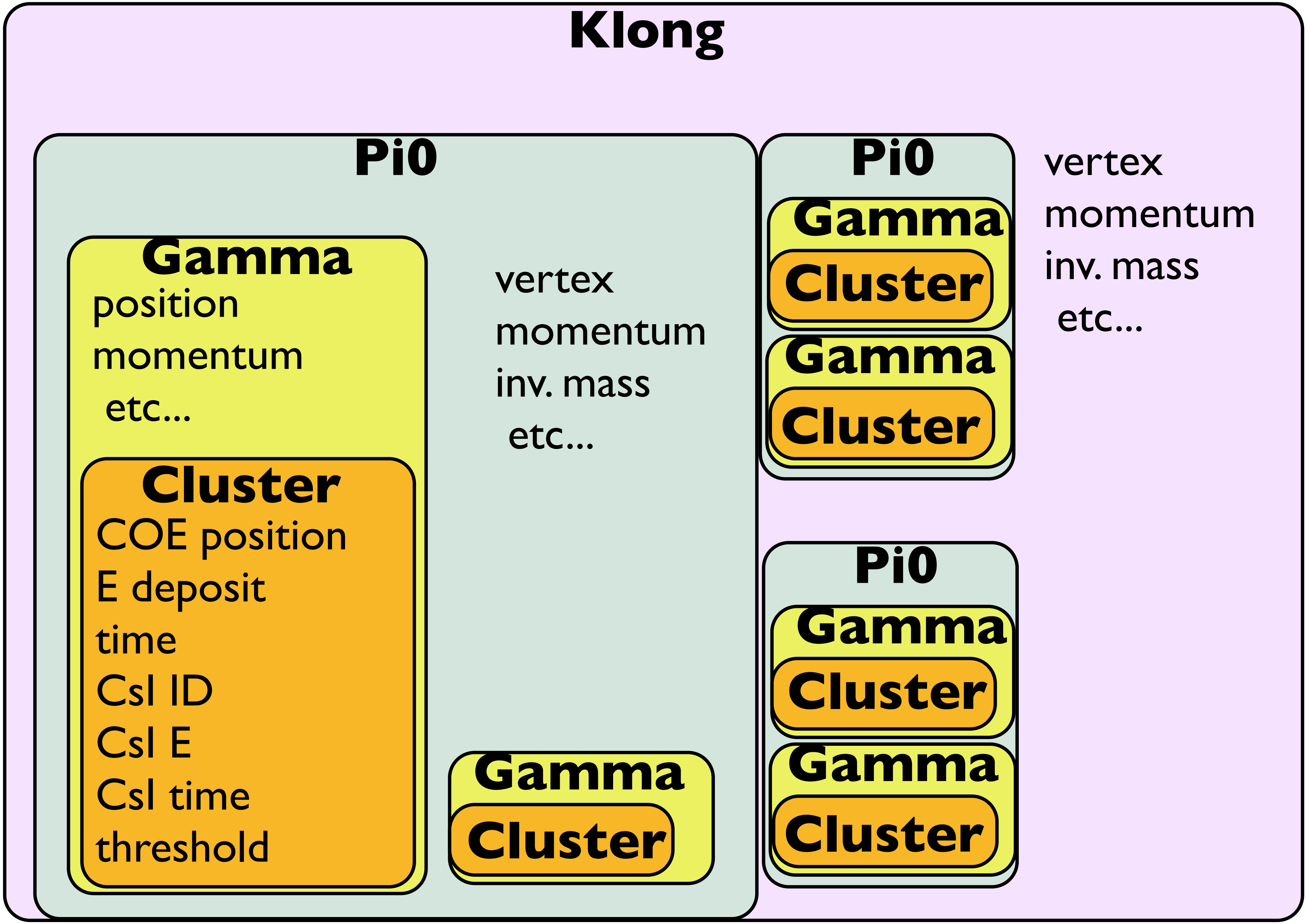
shapeχ2 : class ShapeChi2 provides. void shapeChi2(Gamma& gamma)

shapeANN : class ShapeANN provides. void shapeANN(Gamma& gamma)

data container class

☆classes of data container for analysis

- Cluster (libcluster) : $E14_TOP_DIR/sources/ana/rec/cluster
  - cluster informations
- Gamma (libgamma) : $E14_TOP_DIR/sources/ana/rec/gamma
  - gamma-cluster informations
- Pi0 (libpi0) : $E14_TOP_DIR/sources/ana/rec/pi0
  - reconstructed $\pi 0$ informations
- Klong (libklong) : $E14_TOP_DIR/sources/ana/klong
  - reconstructed K_L informations


Klong
Pi0
Gamma
position
momentum
etc...
Cluster
COE position
E deposit
time
CsI ID
CsI E
CsI time
threshold
vertex
momentum
inv. mass
etc...
Gamma
Cluster
Pi0
Gamma
Cluster
Gamma
Cluster
vertex
momentum
inv. mass
etc...
Pi0
Gamma
Cluster
Gamma
Cluster

summary

- E14 library developing version has various functions for ...
  - clustering
  - gamma identification
  - pi0 or KL reconstruction
  - energy correction
  - position correction
  - shower shape cut

  ...etc

I summarized the functions of e14lib along standard analysis flow.

- Some example executables are prepared in $E14_TOP_DIR/examples/e14gnana. They will help you to learn how to use e14lib for analysis.

e14lib web page

- e14lib developing version is often updated.
  You can get the information of e14lib from the web page
  https://justice.hep.sci.osaka-u.ac.jp/trac/e14lib
  (username: koto, password:[usual one]).
  - from "timeline" tab, you can see the changes in recent updates.

  * some information in this web page is a bit old. Please ignore "installation" part.

backup

flow chart
of example programs

e14clustering

output from gsim4

↓ **(GsimDigiData)**

data conversion to simple array form
DigiReader::getCsiDigi(n,ID[],E[],Time[])

↓ **(array (n,ID[],E[],Time[]))**

clustering
ClusterFinder::findCluster(n,ID[],E[],Time[])

↓ **(std::list<Cluster>)**

data conversion to simple array form
E14GNAnaDataContainer::setData(std::list<Cluster>)

↓ **(array)**

output to root file

e14g2ana

output from e14clusteing

↓ **(array)**

data conversion to std::list<Cluster>
E14GNAnaDataContainer::setData(std::list<Cluster>&)

↓ **std::list<Cluster>**

gamma finding & 1st energy correction
GammaFinder::findGamma(std::list<Cluster>,std::list<Gamma>&)

↓ **std::list<Gamma>**

pi0 reconstruction
Rec2g::recPi0WithConstM(std::list<Gamma>)

↓ **std::list<Pi0>**

position correction & 2nd energy correction → reconstruction
E14GNAnafunction::correctPosition(Pi0&)
E14GNAnafunction::correctEnergyWithAngle(Pi0&)
Rec2g::recPi0WithConstM(std::list<Gamma>)

↓ **std::list<Pi0>** ← apply kinetic cuts, veto cuts,shape cut

data conversion to simple array form
E14GNAnaDataContainer::setData(std::list<Pi0>)

→ output to root file

e14g4(6)ana

output from e14clusteing

↓ **(array)**

data conversion to std::list<Cluster>

E14GNAnaDataContainer::setData(std::list<Cluster>&)

↓ **std::list<Cluster>**

gamma finding & 1st energy correction

GammaFinder::findGamma(std::list<Cluster>,std::list<Gamma>&)

↓ **std::list<Gamma>**

KL reconstruction

RecKlong::recK2pi0(std::list<Gamma>)

↓ **std::vector<Klong>**

position correction & 2nd energy correction → reconstruction

E14GNAnafunction::correctPosition(Klong&)

E14GNAnafunction::correctEnergyWithAngle(Klong&)

RecKlong::recK2pi0(std::list<Gamma>)

std::vector<Klong> ↓ ← apply kinetic cuts, veto cuts,shape cut

data conversion to simple array form

E14GNAnaDataContainer::setData(std::vector<Klong>) → output to root file

demonstration (1)
estimate # of $K_L \rightarrow \pi^0 \nu\nu$
events using MC

(how to use analysis tools)

generate simulation

Terminal — ssh — 80×24

```
[sato@xenys example]$
[sato@xenys example]$ mkdir KLpi0nunuAna
[sato@xenys example]$ cd KLpi0nunuAna/
[sato@xenys KLpi0nunuAna]$ mkdir gsimoutput
[sato@xenys KLpi0nunuAna]$ cp $E14_TOP_DIR/examples/gsim4test/e14.mac .
[sato@xenys KLpi0nunuAna]$ emacs e14.mac &
[1] 24782
[sato@xenys KLpi0nunuAna]$ $E14_TOP_DIR/bin/gsim4test e14.mac gsimoutput/KLpi0nu
nu.root 100000 0 0

*************************************************************
 Geant4 version Name: geant4-09-00    (29-June-2007)
                      Copyright : Geant4 Collaboration
                      Reference : NIM A 506 (2003), 250-303
                            WWW : http://cern.ch/geant4
*************************************************************


 illegal code for meson  PDG code=10000130
Particle KLpi0nunu has a strange PDGEncoding
 illegal code for meson  PDG code=100130
Particle KLpienu has a strange PDGEncoding
 illegal code for meson  PDG code=200130
Particle KLpimunu has a strange PDGEncoding
 illegal code for meson  PDG code=300130
```

make working directory

copy a macro file and edit it
(→see next page)

output file

#event

random seed

start MC

edit the macro file

* configuration of MC are written in a macrofile.

examples of macrofile are found in $E14_TOP_DIR/examples/gsim4test

e14.mac – emacs@xenys.hep.sci.osaka-u.ac.jp

- edit e14.mac as follows

```
/GsimPhysicsListFactory/QGSP_BERT/buildAndRegister
/GsimPrimaryGeneratorActionFactory/GsimGeneralParticleSource/buildAndRegister gps
#/gps/particle KLpi0pi0
/gps/particle KLpi0nunu
/gps/energy 1 GeV
/gps/position 0 0 -1 m
/gps/direction 0 0 1
#/control/execute oglsini.mac
#/GsimEventAction/setVisualizationMode 1
#/GsimEventAction/setAccumulationNumber -1
/GsimTrackingAction/setBriefTrackStore true
```

change a kind of particle to KLpi0nunu

comment out viewer settings

• • • • • •

```
#/GsimSpectrumFactory/GsimE14Spectrum/buildAndRegister
#/GsimSpectrum/GsimE14Spectrum/addSpectrum Special
/GsimSpectrumFactory/GsimE14SurveySpectrum/buildAndRegister
/GsimSpectrum/GsimE14SurveySpectrum/addSpectrum Special

/GsimDetectorManager/update
/run/initialize
#/run/beamOn 20
```

change momentum spectrum from E14Spctrum (a bit old) to E14SurveySpectrum.

clustering

Terminal — ssh — 80×24

```
[info    ] [Thu Aug  2 12:24:27 2012]  [MemSize =    68242 | RES =    57681 | SHR
 =      7347] ### 100 % of events processed.
[info    ] [Thu Aug  2 12:24:27 2012]  [MemSize =    68242 | RES =    57684 | SHR
 =      7350] number of event = 100000 User=4850.18s Real=4857.25s Sys=6.15s

Graphics systems deleted.
Visualization Manager deleting...
[sato@xenys KLpi0nunuAna]$ mkdir clusteringoutput
[sato@xenys KLpi0nunuAna]$ $E14_TOP_DIR/bin/e14clustering gsimoutput/KLpi0nunu.r
oot clusteringoutput/KLpi0nunu.root
input file: gsimoutput/KLpi0nunu.root
output file: clusteringoutput/KLpi0nunu.root
[info    ] DigiReader::addDetector:  ID 13:os.
[info    ] read /net/fs/powval3/koto/e14lib/installed20120731/e14/share/csimap/cs
i_dettree.root as CsiMap-data file.
[info    ] CsiMap:: # of crystals == 2716
Error in <TTree::SetBranchStatus>: unknown branch -> CC01.
Error in <TTree::SetBranchStatus>: unknown branch -> CC02.
Error in <TTree::SetBranchStatus>: unknown branch -> os.
# of entry in input tree ==100000

 start loop analysis for 100000 events...
0%
1%
```

MC

←make cluster from gsim4 output file

start clustering program

g2ana (reconstruct pi0)

Terminal — ssh — 80×24

```
97%
98%
99%
finish!
[sato@xenys KLpi0nunuAna]$ mkdir g2anaoutput
[sato@xenys KLpi0nunuAna]$ $E14_TOP_DIR/bin/e14g2ana clusteringoutput/KLpi0nunu.
root g2anaoutput/KLpi0nunu.root
input file: clusteringoutput/KLpi0nunu.root
output file: g2anaoutput/KLpi0nunu.root
start loop analysis
# of entry : 100000
0%
[info    ] CsiMap:: # of crystals == 2716
read /net/fs/powval3/koto/e14lib/installed20120731/e14/share/gamma/mapFineCsi_f.
dat
```

e14clustering

←reconstruct pi0 from e14clustering output file

start reconstruction program

. . . . . .

```
60%
70%
80%
90%
[sato@xenys KLpi0nunuAna]$ █
```

reconstruction program

result

Terminal — ssh — 80×24

```
[sato@xenys KLpi0nunuAna]$ root g2anaoutput/KLpi0nunu.root

 starting root... processing rootlogon.C

root [0]
Attaching file g2anaoutput/KLpi0nunu.root as _file0...
root [1] Tree->Draw("Pi0Pt[0]:Pi0Pos[0][2]","(CutCondition)==0","colz")
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
(Long64_t)222
root [2] 222./1e5*1e7*3600*24*365*3/3/3.3*3e-11
(const double)6.36453818181818320e+00
```

require kinetic cut

in signal box

plot the events in signal box

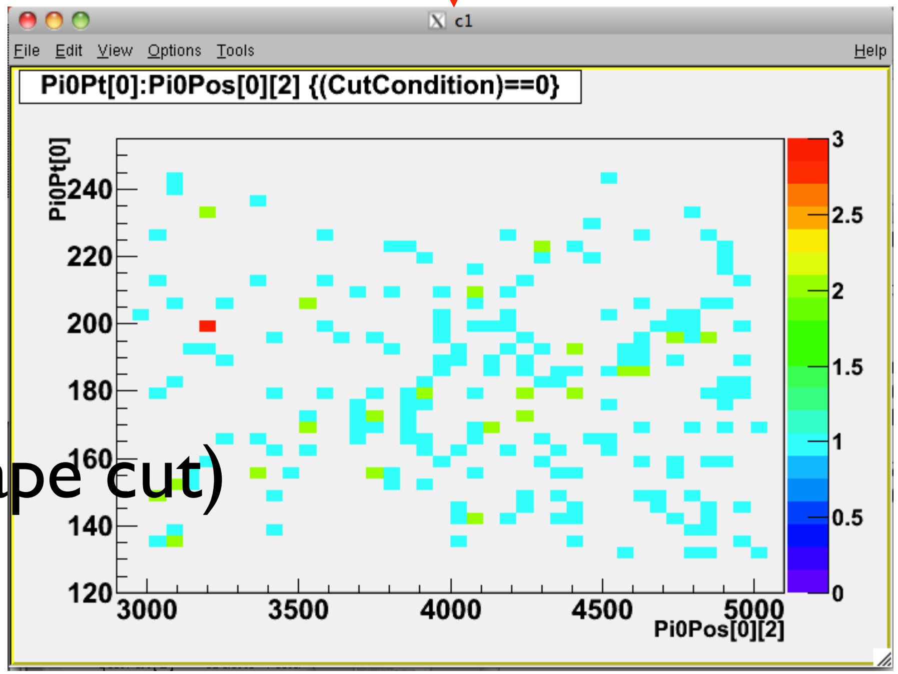
#of KL generated

#of KL/spill

3 snowmass years

spill cycle[sec]

Br



~ 6 events are expected

(w/o veto, accidental loss, shower shape cut)

demonstration (2)
compile your code
with e14library

consider you want to compile test.cc that use class Cluster.

```
$ mkdir test
$ cd test/
$ emacs test.cc&
$ cp $E14_TOP_DIR/examples/e14gnana/e14clustering/REQUIREMENTS .
$ cp $E14_TOP_DIR/examples/e14gnana/e14clustering/Makefile .
$ emacs Makefile
$ make
Making dependency for file test.cc ...
Compiling test.cc ...
Using granular libraries ...
Linking test ...
... Done!
$ bin/test
hello, cluster
```

make working directory

edit test.cc

copy Makefile and REQUIREMENT

edit Makefile

execute

work as expected

test.cc

```
#include <iostream>
#include <cluster/Cluster.h>

int main(){
  Cluster cluster;
  std::cout<<"hello, cluster"<<std::endl;
}
```

REQUIREMENT

```
#config
gnana
csimap
cluster
gamma
pi0
klong
rec2g
E14Fsim
GsimData
GsimPersistency
```

name of libraries are defined in REQUIREMENT

Makefile

```
# -*- Makefile -*-

#MAINS = e14clustering
MAINS = test
PACKAGE = $(shell basename `pwd`)
INSTALL_LIBS = no
#INSTALL_BINS = yes
INSTALL_BINS = no

include $(E14_CONFIG_DIR)/Makefile.g4
```

←edit MAINS

←change yes⇒no