Subject: Question on R3BNeutronTracker2D
Posted by C. A. Douma on Thu, 07 Jul 2016 13:17:57 GMT
View Forum Message <> Reply to Message

Sometimes I get not enough neuthits in the output.

I know that the neutron tracker applies the cuts from the
calibration files at lines 267-300 and then stores the
number of tracks that have to be reconstructed in the variable
nNeut. nNeut is also filled in the histogram h_ntracks.
However, if I let the neutron tracker run
(TGeant4, Ca48 with 500 AMeV on C-target & GLAD-field on),
then neuthit.@size and h_ntracks do not match at all. @size is smaller.

Can anyone explain me why the neutron tracker sometimes has less tracks?
I did find that if I skip the beta-window test in the AdvanchedMethod()
that this problem is solved, but then I get other problems in return...

Yours Sincerely,
Christiaan Douma.

Subject: Re: Question on R3BNeutronTracker2D
Posted by Dmytro Kresan on Thu, 07 Jul 2016 15:00:38 GMT
View Forum Message <> Reply to Message

Hi Christiaan,

How often does this happen, percentage of events?

Determining nNeut is just the first step in the reconstruction. nNeut is based on calorimetric and multiplicity properties of an event. Afterwards you need to find your candidates in the hit pattern. Imagine you did 3n -> 4n misidentification, then most probably program will find only 3 best candidates, so @size < nNeut. And this can happen in 20 - 30% of events - depending on input and cuts.

How did you calculate the energy/clusters cuts for your simulation? You need to recalculate them for every different input, do not apply "standard" values.

In addition, beta cut is applied, to select neutrons with velocity close to beam. Due to finite resolution, this condition could also cut out good candidates.

You can investigate 2 things:

1) Determine your multi-neutron identification matrices, like was done for 132Sn input. In your case, you can define primary neutron by cutting on time and position of the MCTrack vertex - close to (0, 0,0,0). Then you will know your proper 2D cuts, efficiency and misidentification.

2) To study the reaction input - plot the velocity distribution of "primary" neutrons, and cut values on top of it. This will help to adjust the beta-cut for this physics case.

Best regards,
Dima

Subject: Re: Question on R3BNeutronTracker2D
Posted by C. A. Douma on Thu, 07 Jul 2016 15:20:17 GMT
View Forum Message <> Reply to Message

Dear Mr. Kresan,

This happens for nearly all events. The tracker identified either 0 or 1 track (and 1/10000 events: 2 tracks). The zeros are understandable, since not every Ca48 reacts in the target. Then it is bend away.

I re-calculated the cuts for this specific case. I also use different cuts for different number of hits in the VETO. I use the 0p cuts if no VETO fired, the 1p cuts if exactly one bar of the VETO fired (TOF<NeuLAND), and so on.
The cuts might not be completely ideal (as they would be with your method 1), but they do look quite OK to me.

If I understand both you and the code correct, then you first mark & sort the clusters based on whether they can come from elastic scattering or not. Then from the sorted & marked clusters, you simply take the starting positions as the neuthits, but only those that have not been used yet, pass the energy threshold & pass the beta test, correct? And if the elastic scattering test provides you with less marked clusters then nNeut, then you cannot reconstrcut nNeut tracks. This is the 20%-30% MisID that you were talking about. Am I correct?

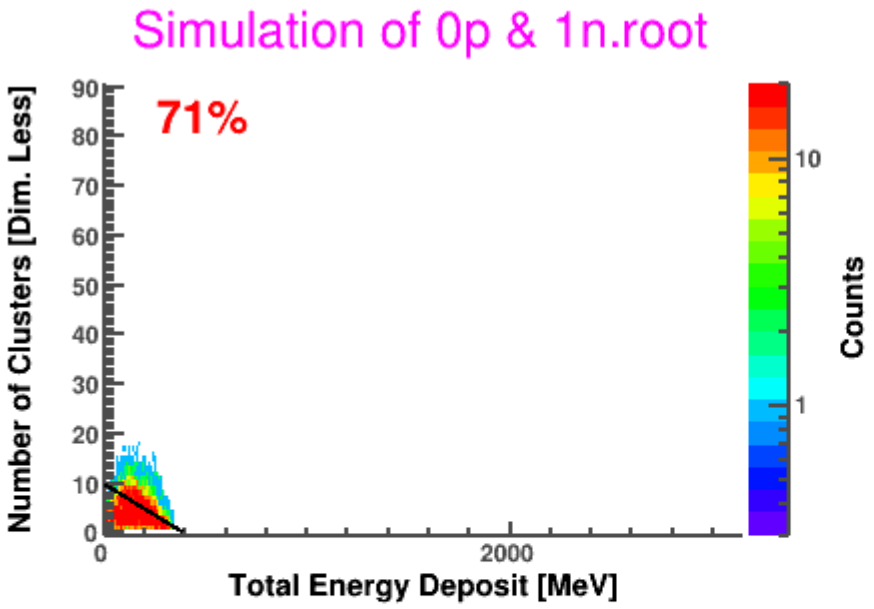
But if so, then how relevant is this beta test? In my case most particles are from background, not from the target, so their beta will not be close to the beam.

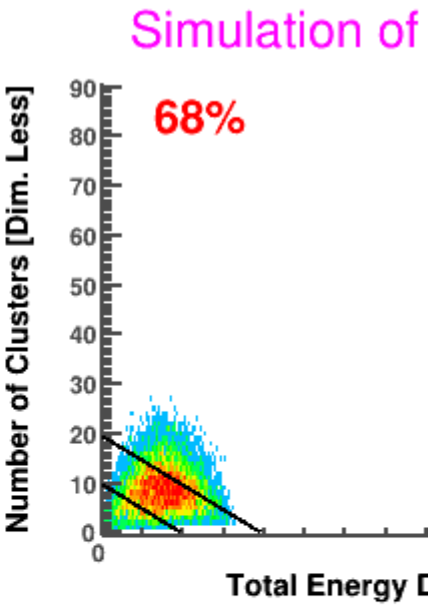
What I also do not understand is, why does the code (line 509) read:
if(TMath::Abs(beta-beamBeta) > (0.05*600./beamEnergy) && ic > 0) {
I would expect the beta-test to be a '<', not '>'.
And if I would have the beta's for my specific case to be for example between (from your suggestion 2) 0.7 and 0.75, then how would this translate in a better beta cut?

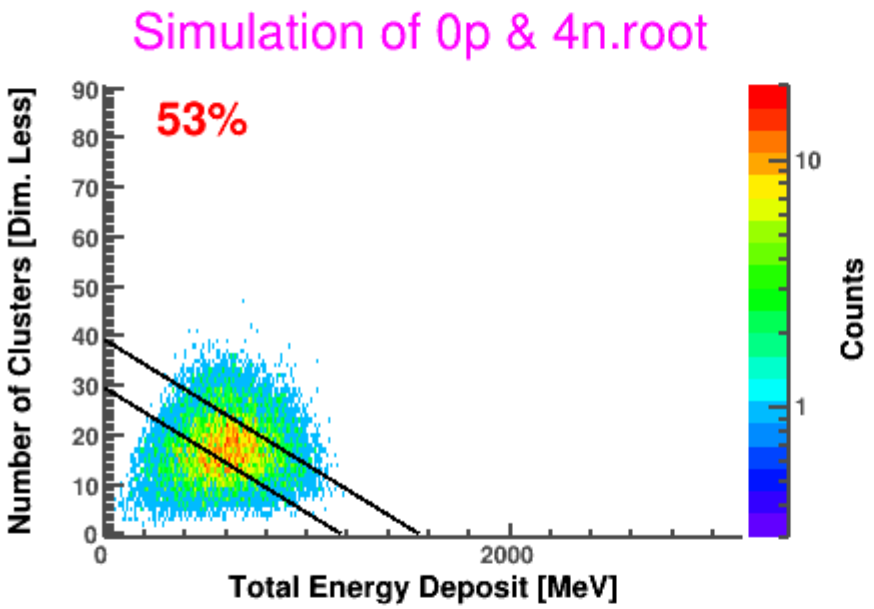
Christiaan.

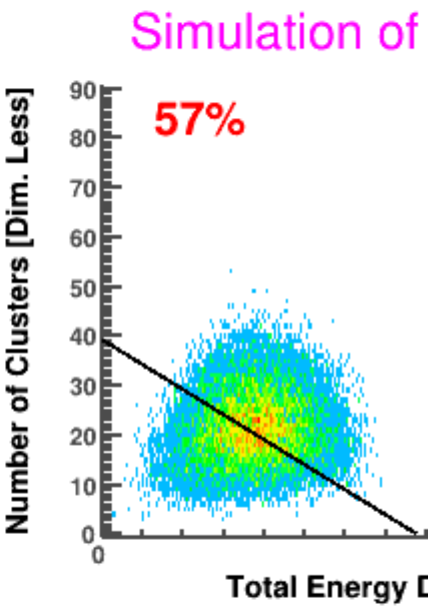
File Attachments

1) NeutronCalibration_0p.png, downloaded 997 times

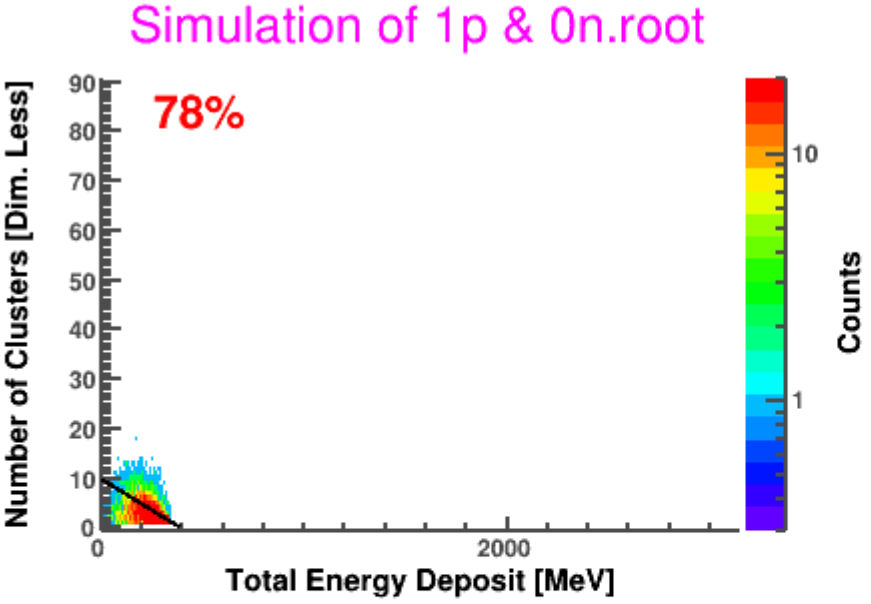

Simulation of 0p & 1n.root
71%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts
0
2000
10
1

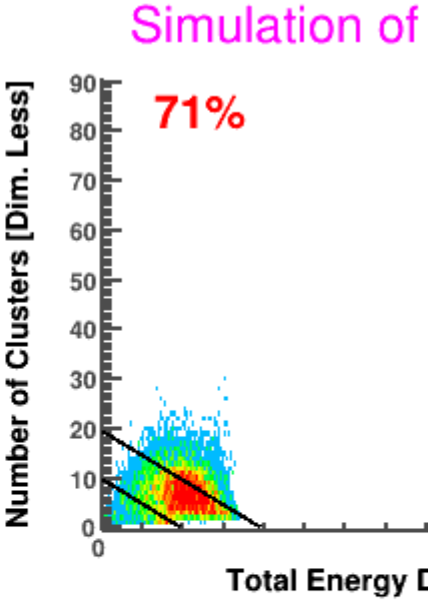

Simulation of
68%
Number of Clusters [Dim. Less]
Total Energy D

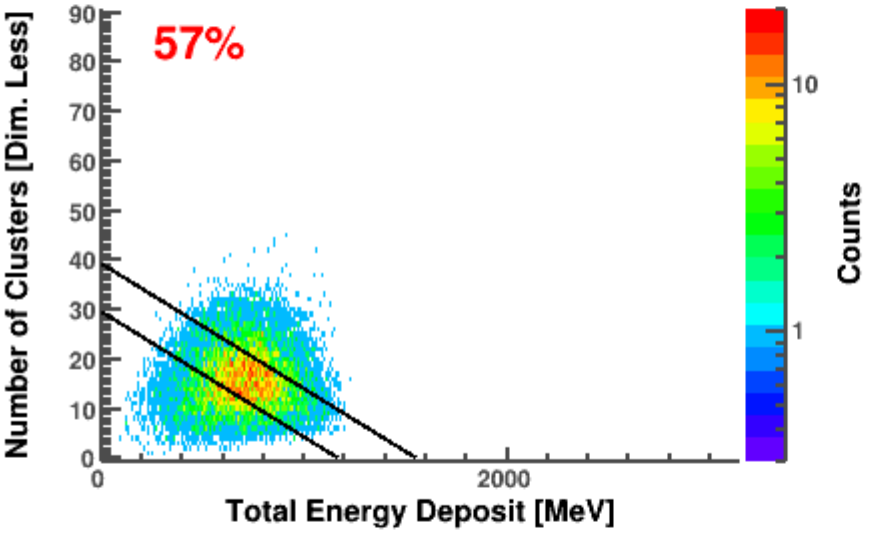

Simulation of 0p & 4n.root
53%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts
0
2000
10
1

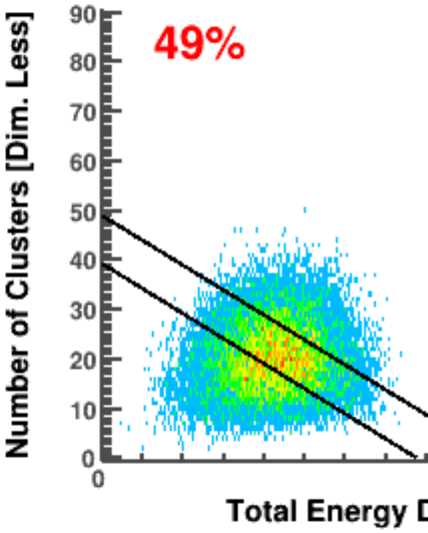

Simulation of
57%
Number of Clusters [Dim. Less]
Total Energy D

2) NeutronCalibration_1p.png, downloaded 936 times


Simulation of 1p & 0n.root
78%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts
0
2000
10
1

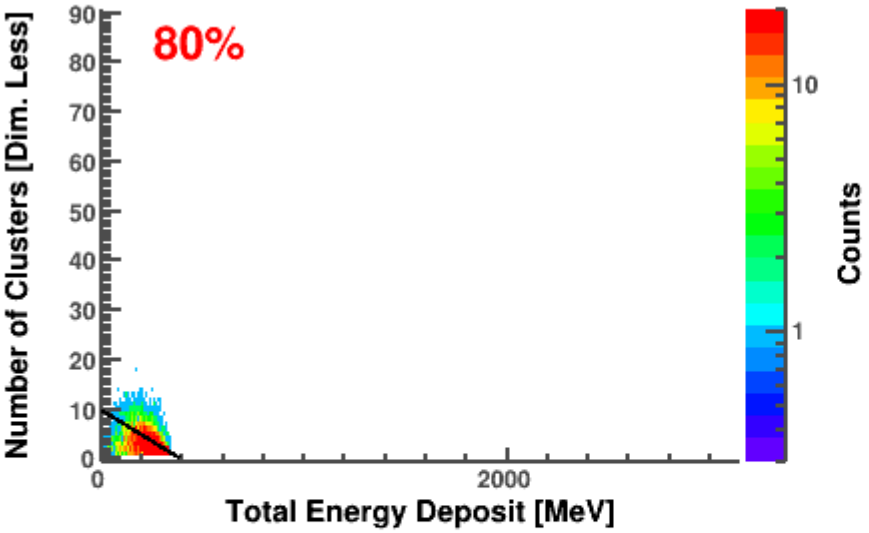

Simulation of
71%
Number of Clusters [Dim. Less]
Total Energy D


Simulation of 1p & 3n.root
57%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts
0
2000
10
1

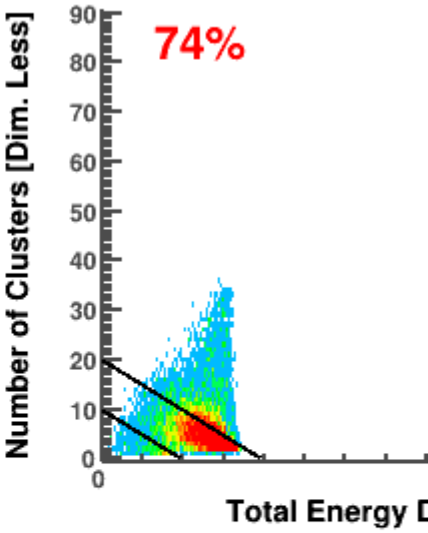

Simulation of
49%
Number of Clusters [Dim. Less]
Total Energy D

3) NeutronCalibration_2p.png, downloaded 908 times

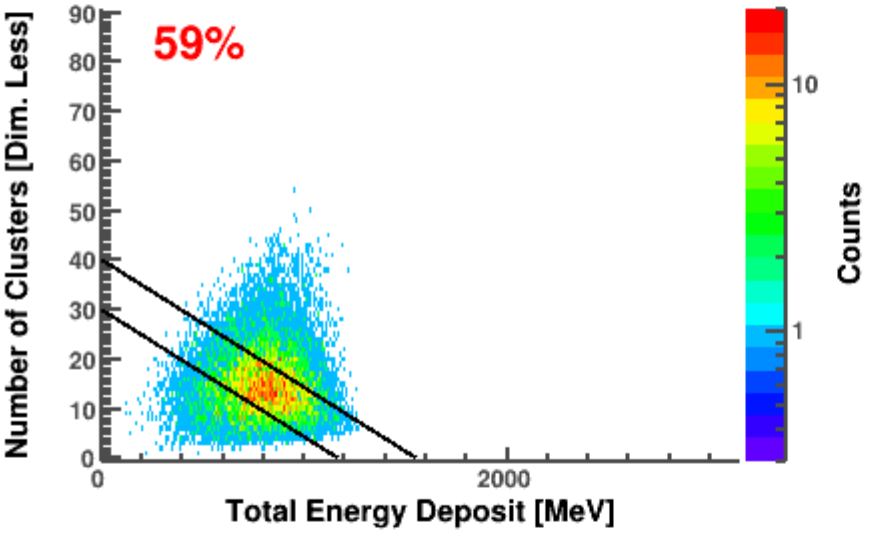
Simulation of 1p & 0n.root



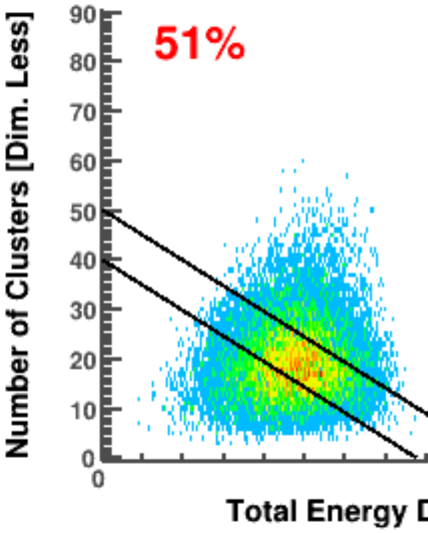
Simulation of



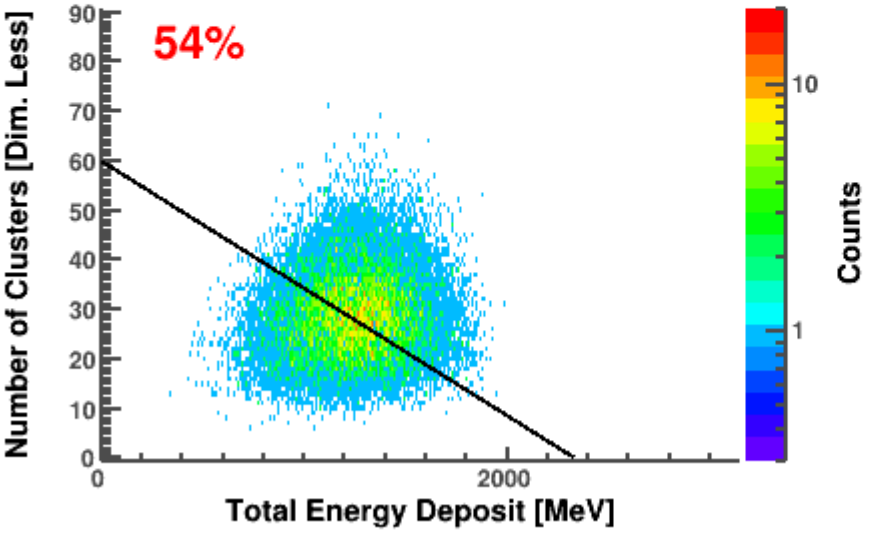
Simulation of 2p & 2n.root



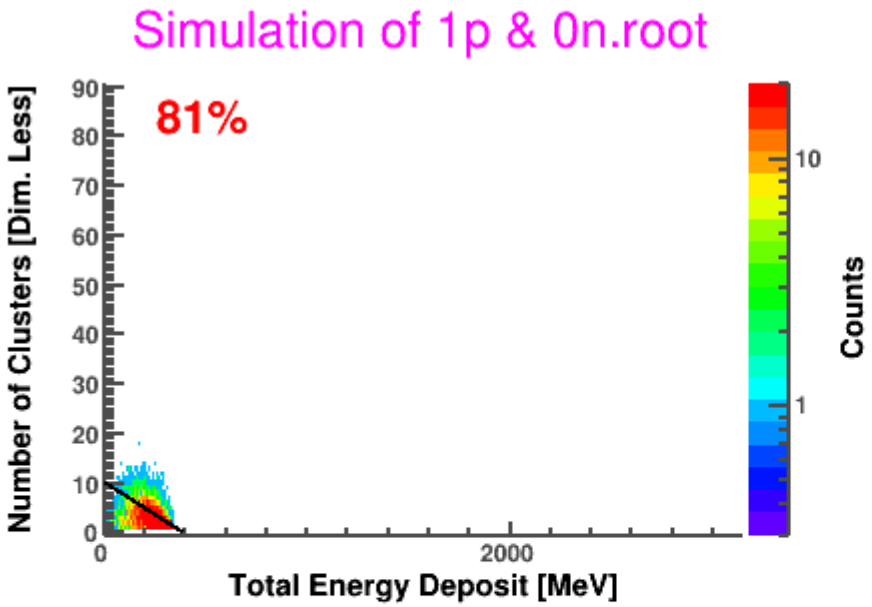
Simulation of

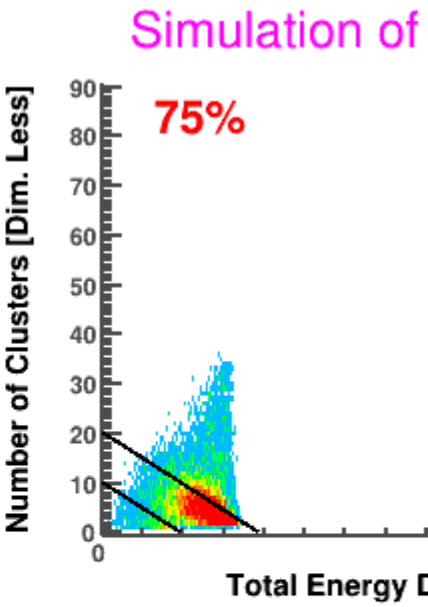


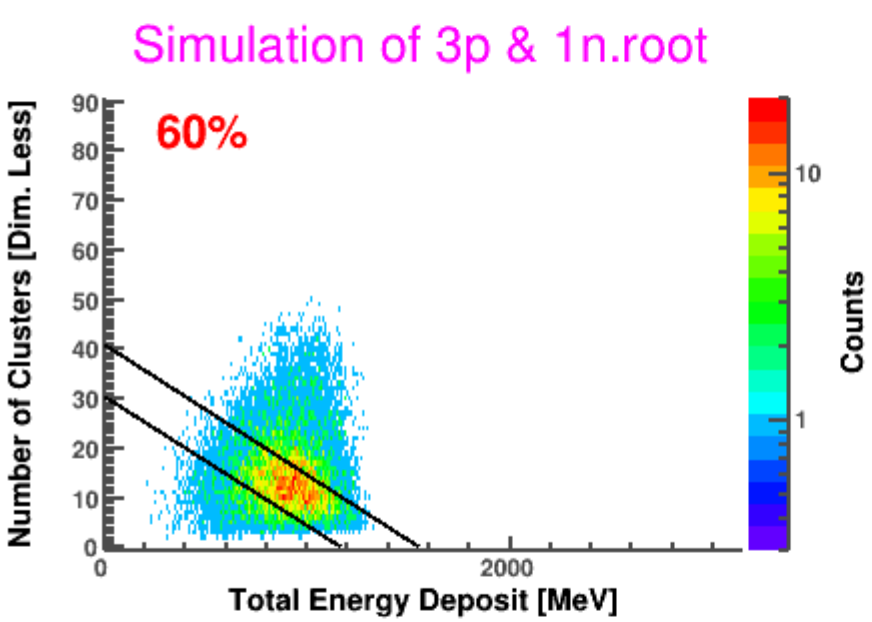
Simulation of 2p & 5n.root

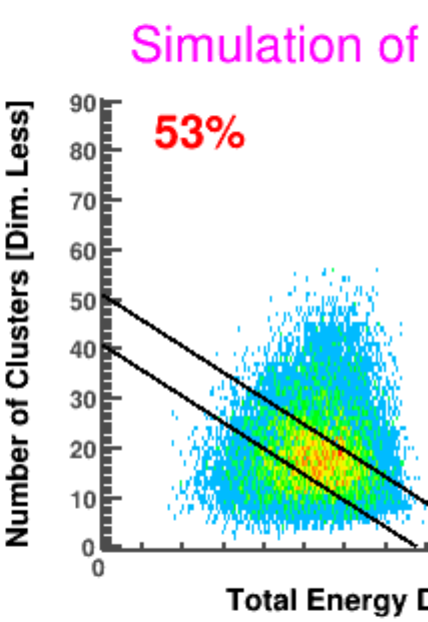


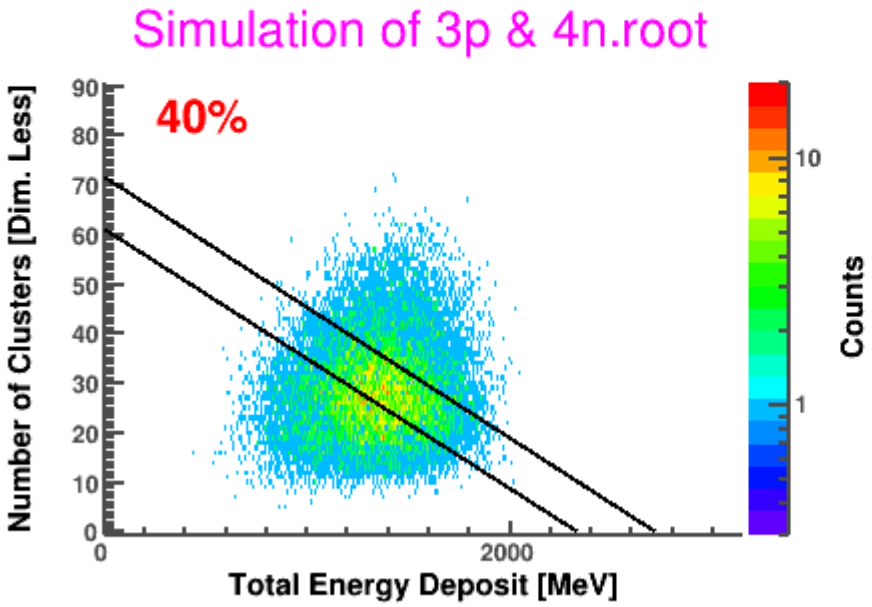
4) NeutronCalibration_3p.png, downloaded 938 times

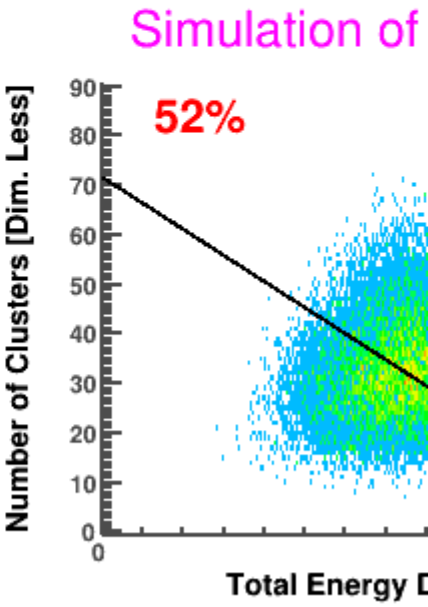

Simulation of 1p & 0n.root
81%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts


Simulation of
75%
Number of Clusters [Dim. Less]
Total Energy D

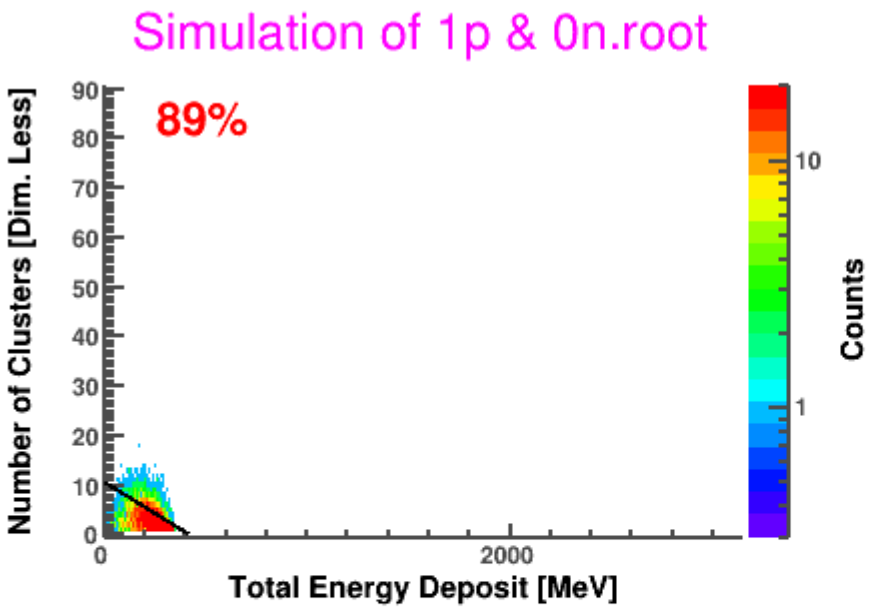

Simulation of 3p & 1n.root
60%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts

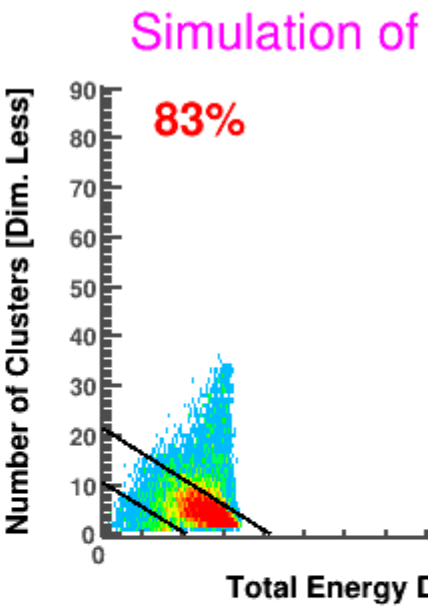

Simulation of
53%
Number of Clusters [Dim. Less]
Total Energy D

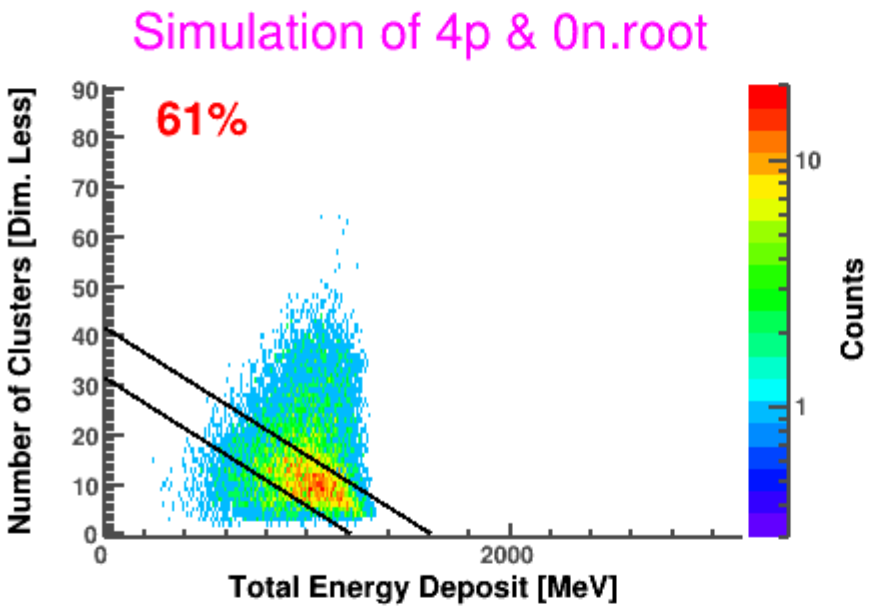

Simulation of 3p & 4n.root
40%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts

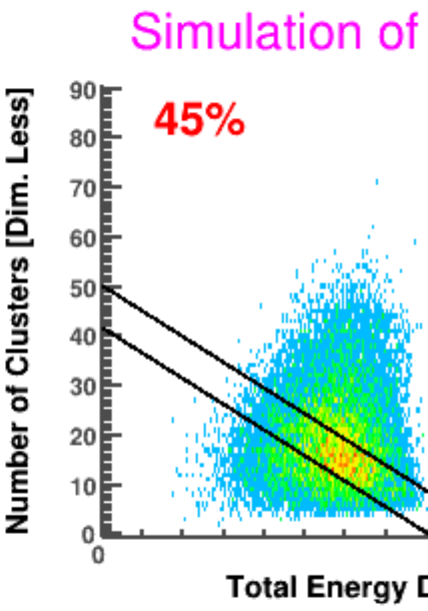

Simulation of
52%
Number of Clusters [Dim. Less]
Total Energy D

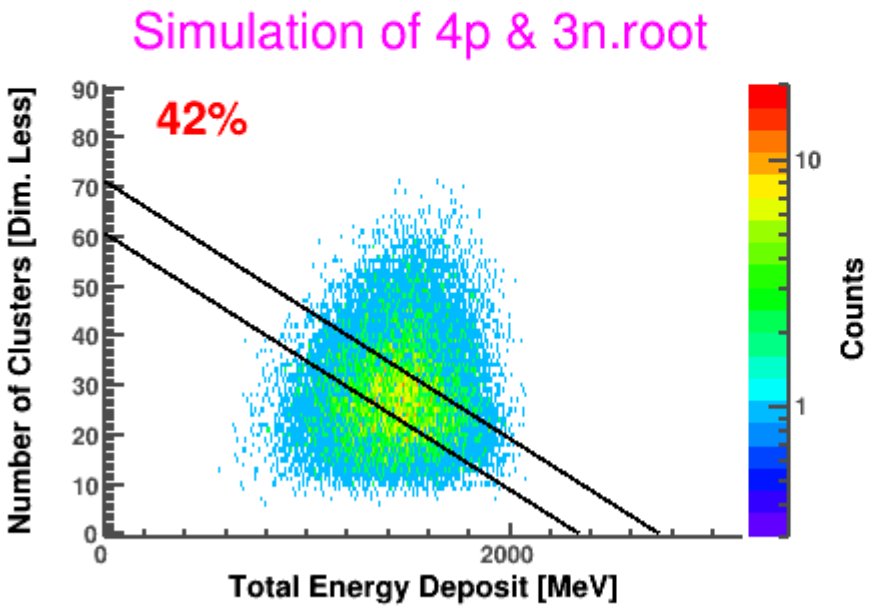
5) NeutronCalibration_4p.png, downloaded 909 times

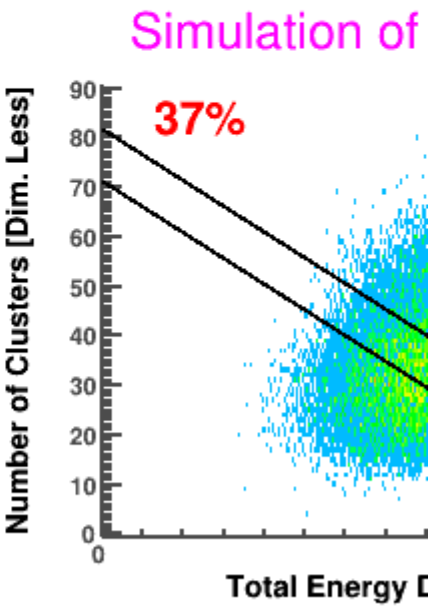

Simulation of 1p & 0n.root
89%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts


Simulation of
83%
Number of Clusters [Dim. Less]
Total Energy D


Simulation of 4p & 0n.root
61%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts


Simulation of
45%
Number of Clusters [Dim. Less]
Total Energy D


Simulation of 4p & 3n.root
42%
Number of Clusters [Dim. Less]
Total Energy Deposit [MeV]
Counts


Simulation of
37%
Number of Clusters [Dim. Less]
Total Energy D

Subject: Re: Question on R3BNeutronTracker2D
Posted by Dmytro Kresan on Thu, 07 Jul 2016 15:42:26 GMT
View Forum Message <> Reply to Message

1) Yes, you understood me and code correctly. This is what I meant with misidentification.

2) if(TMath::Abs(beta-beamBeta) > (0.05*600./beamEnergy) && ic > 0) {
continue;
}
This is inverted condition - if deviation is larger -> skip the cluster.

3) Quote:In my case most particles are from background, not from the target, so their beta will not be close to the beam.
This is exactly the problem. R3BNeutronTracker2D was written to reconstruct signal neutrons. What about if you add to every event with Ca ion, interacting with target and producing mostly background, also "signal" neutron(s) from a particle gun (or ASCII). Then you can study how your Ca background affects detectors performance.

Best regards,
Dima

Subject: Re: Question on R3BNeutronTracker2D
Posted by C. A. Douma on Thu, 07 Jul 2016 15:51:34 GMT
View Forum Message <> Reply to Message

If I manually add signal neutrons, I cannot observe the realistic case.
Both the probability for a reaction and the probability for background from the unreacted beam are small and
of comparable size. This is why the background causes trouble. If I manually add signal neutrons
to the unreacted beam, the suddenly the signals are so much stronger then the background that the VETO does
more harm then good: if I have for example 0.5% background and 1.5% of the neutrons is detected by the VETO, the VETO will
never help us. But if I have 0.5% background and 0.5% neutrons, then the VETO only blocks 1.5% of 0.5%. Then it might work.

This is also the reason why I asked you about the TGeant4 physics list. I need to really produce the neutrons from a simulated
reaction and I need to take the unreacted beam along to produce the background. And according to Ken Miki, TGeant3 cannot
do this target collisions.

But when I do add these signal neutrons, the tracker works fine, also with the beta test.

Christiaan.

Subject: Re: Question on R3BNeutronTracker2D

Posted by C. A. Douma on Mon, 11 Jul 2016 11:26:37 GMT
View Forum Message <> Reply to Message

Dear Mr. Kresan,

I found the source of the problem concerning the beta test. The beta test works fine exactly as you designed it.
However, in my specific case the beam starts 4 meters behind the target. This causes that the collision at the target
does not happen at t=0. When beta is then computed as

beta = cluster->GetZ()/(cluster->GetT()*c);

the time requested by cluster->GetT() is the TOF between the target and NeuLAND plus the time between the beam starting point
and the target. But beta should of course be computed as

beta = cluster->GetZ()/(TOF*c);

where TOF is only the time between the cluster startpoint and the target collision.

I modified this in the neutron tracker. I also changed the beta definitions (and other similar pieces of code) to:

beta = TMath::Sqrt(cluster->GetZ()*cluster->GetZ() + cluster->GetY()*cluster->GetY() + cluster->GetX()*cluster->GetX())/(TOF*c);

so that the neutron tracker is now fully invariant under rotations etc. of the detector (relevant to simulate the EOS experiment of Igor, for example).

NOTE: the io-system of the tracker that I added here is still designed to work only for my own simulations.

Christiaan.

PS: I also noticed that the tracker still does not work with the R3B Neuland-class, only with the land-class. Is there a possibility to fix this?

File Attachments

1) R3BNeutronTracker2D.cxx, downloaded 434 times
2) R3BNeutronTracker2D.h, downloaded 439 times

Subject: Re: Question on R3BNeutronTracker2D
Posted by Dmytro Kresan on Mon, 11 Jul 2016 12:07:08 GMT
View Forum Message <> Reply to Message

Hi Christiaan,

perfect, that you have found solution for your problem with beta cut. When do you plan to commit this fix?

About status of Tracker in combination with NeuLAND detector class we have to ask Jan Mayer.

Best regards,
Dima

Subject: Re: Question on R3BNeutronTracker2D
Posted by C. A. Douma on Mon, 11 Jul 2016 13:57:53 GMT
View Forum Message <> Reply to Message

How am I supposed to commit this fix? I have sent the code to you. I though that that was the procedure to contribute code to R3BRoot.

Besides: the tracker I have send you does more then fixing this. It is also capable of applying different sets of cuts depending on how many bars fired in the VETO. However, this procedure does not work for general R3BRoot, since it requires the veto_seg-class (and corresponding digitizer) that do not exist today. I always add those classes myself by duplicating NeuLAND. I even have a script to do that.

So then how do we add the new tracker to R3BRoot? (Same question for the new clusterfinder and for my adapted version of the Digitzer as long as Jan does't have had time to write it more elegantly. That software however does not need the veto class.)

Christiaan.

Subject: Re: Question on R3BNeutronTracker2D
Posted by Dmytro Kresan on Tue, 12 Jul 2016 08:47:33 GMT
View Forum Message <> Reply to Message

We have git repository on GitHub, through which the commits should be done. Follow the link to learn about the workflow:

https://www.r3broot.gsi.de/git-workflow

You can create "veto" subdirectory in R3BRoot and put all your relevant classes there. In addition create also the "macros/r3b/veto" directory.

Best regards,
Dima

Subject: Re: Question on R3BNeutronTracker2D
Posted by C. A. Douma on Wed, 13 Jul 2016 12:32:11 GMT
View Forum Message <> Reply to Message

Dear Mr. Kresan,

Thank you for sending me the documentation on the GitHub repository. However, in a few days
I will go on holiday for 4 weeks. I will not have time to set up this git environment before I leave. May I therefore ask you to commit my changes? I have adapted the software to work without the need of a VETO class, etc, so it can be committed right away.

I tested the compilation on the Dev Branch of 26 May, and it compiles without errors. After testing the code on my own simulations, all I did were some changes in how to collect inputs in the Init()-functions. It should therefore work during runtime as well.

The changes I made comprise the following:

1) Digitizer: coordinate transformations between local and global frames work OK now. The user can also adapt energy thresholds & time resultions if he/she desires.
2) Clusterfinder: Clusters will now be merged if they touch/overlap. Cluster radius in both space and time can also be set by the user now.
3) NeutronTracker: The beta-test (& the rest of the tracker) is now invariant under global/local differences. The user can smanually set the beam start position and the target position, so that Time Of Flight will now be computed only between the target and NeuLAND (as it should be).
   In addition, the tracker now automatically takes any number of energy cuts specified in the calibration file. You are no longer bound to exactly 5 cuts.
   Furthermore, depending on the number of hits in the VETO, the tracker can also use a set of cuts for 1-5 neutrons, or a set for 1 proton + 0-5 neutrons, or a set of 2 protons + 0-5 neutrons, etc. Multiple calibration files will be read automatically and the right file is selected in each event based on the number of hits in the VETO. The user has to specify an upper bound for the number of protons (it is set to 0 by default) and calibration files up to this upper bound should be provided. If the VETO digitizer ouput is not specified by and AddFriend-call, the tracker will automatically revert back to the 0 proton case for every event.

Thanks in advance!
Christiaan.

File Attachments

1) R3BRoot.tar.gz, downloaded 593 times