Neutron Detector Array NAND Lesson 5

Ambar Chatterjee

Time-shifted TDCs

The time reference of the pulsed beam can change from run to run. It can also shift during the course of a run itself. Than can complicate the analysis.

All the files used here are available at
http://www.ambar-chatterjee.com
Questions?
DrAmbar@gmail.com

Foreword

After the release of the previous 4 videos, I received responses from **Hardev Singh** and **Vafiya Thalsim** where they mentioned about the timing reference shift making it necessary to analyse the files one by one with different gates combining the final E_n spectra for different runs (rather than as a TChain).

At first it appears that the most time consuming part of drawing the TCuts on neutron TOF vs PSD have to be repeated for every file.

In this video we present a shorter way to correct the time shifts

Looking at the Videos

The first time you study a lesson, the video is the best way to understand things. But later, when you want to refer to the material presented, the power point and the source codes packed in the tgz will usually be enough. So do keep a copy of these files in a separate folder.

Getting the files you need

Website: http://www.ambar-chatterjee.com

I am not providing any root data files in this lesson. The new programs written and this powerpoint are packed in nand05.tgz which you can download

Ambar Chatterjee

Work:

- Bhabha Atomic Research Centre, Mumbai (Retd.)
- MS Univ of Baroda
- Inter University Accelerator Centre, New Delhi

Websites:

- LAMPS Data Acquisition
- AIWCF Chess Server

Download lamps_offline	Click	
Chess Diagrams	App	Online
ROOT Videos YouTube	Click	
Neutron Array YouTube	Click	

Videos

NAND Lesson 1	MWPC Slicing, Angle Calculations	Click
NAND Lesson 2	Structured ROOT programs	Click
NAND Lesson 3	Building Neutron Spectra	Click

Support Files

NAND Lesson 1	nand01.tgz	
NAND Lesson 2	IUAC_NAND_data.001	nand02.tgz
NAND Lesson 3	nand03.tgz	
NAND Lesson 4	nand04.tgz	
NAND Lesson 5	nand05.tgz	

Example data for today's lesson

Files:
Data01.001, Data02.001, Data03.001
Each about 100 MB (not provided in website)
There were 50 neutron detectors (but 8351 and 6135 not working)

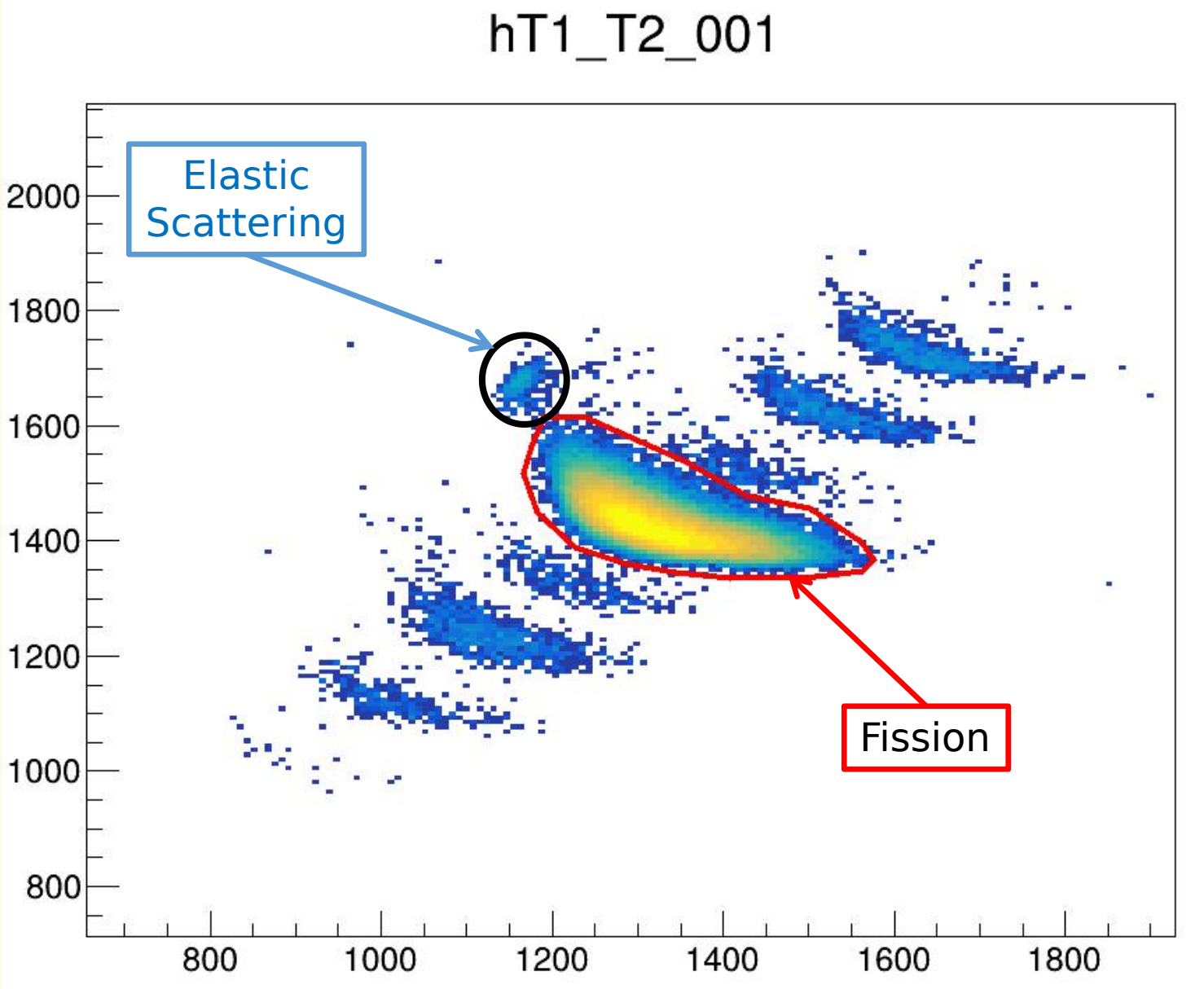
Tree name is RoseNIAS

The parameters are named a bit differently and they are in a different order. There are also many unused parameters.

After generating ParaList.txt using Prog001.cpp we will delete the unused parameters and reorder them in ParaList.txt so as to make it similar to what we have been using so far.

In the first file (Data01.001) the time reference is continuously shifting. Many different time references can be seen.

Observing the time reference shift in MW1_T vs MW2_T



In Data01.001
many bands are seen
The time reference shift is continuously shifting throughout the data

On examining the data, we find that the jumps from one time reference to another happens randomly throughout the run. So we cannot analyse by dividing the file into parts.

More about time reference shift

During data taking, if we observe continuous time reference shifts we should stop taking data and request beam re-tuning.
In practical cases, data is taken in non-ideal conditions.
In this example most of the data is in the main band, so we can analyse only this band

If the data statistics is distributed in many bands, it is possible to draw multiple gates and analyse.

At the end of this Lesson we will show how this can be done without much trouble.

The same time shifts will be in the ungated neutron time of flight.

Understanding the shifts

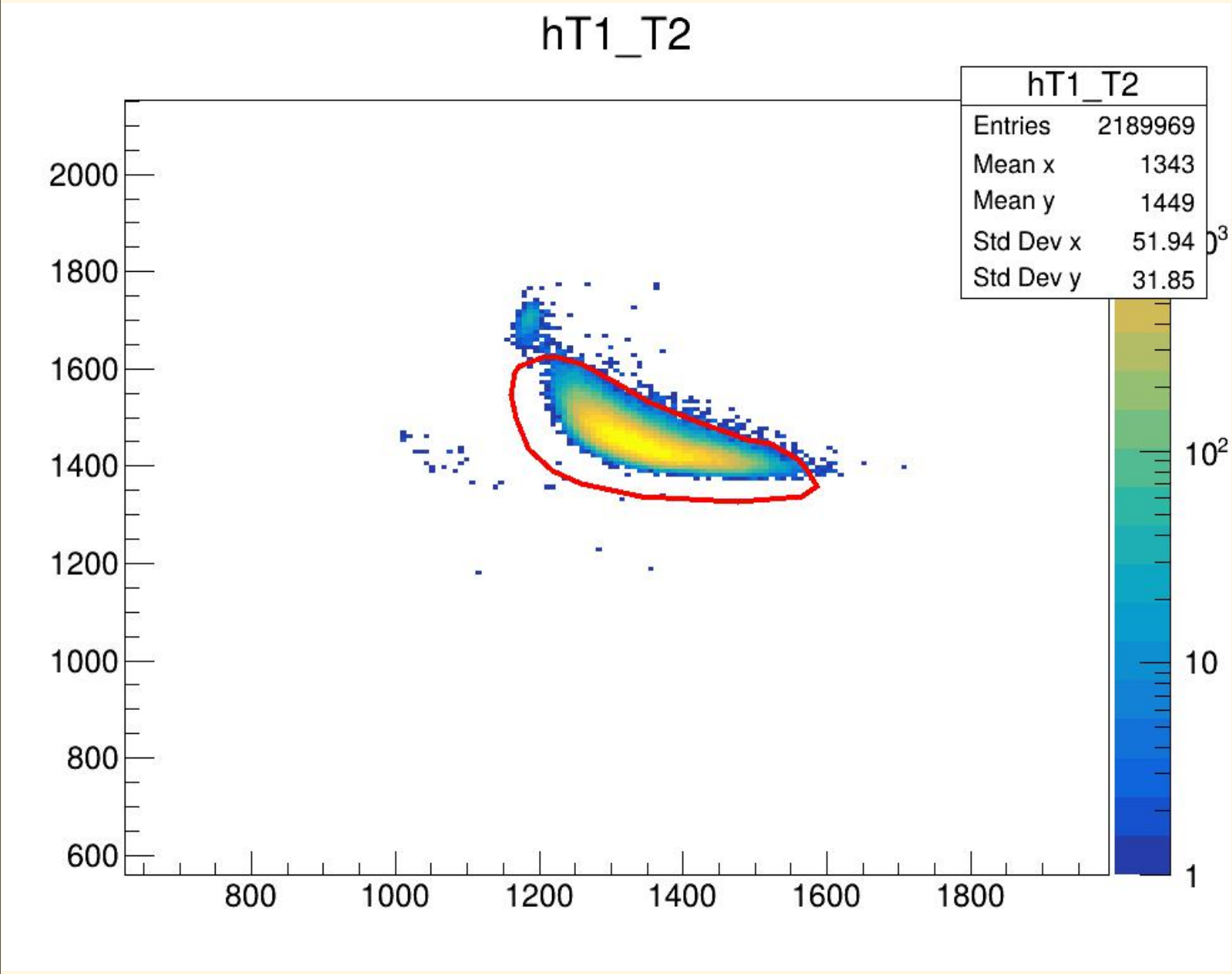
This will affect all the TDCs (where the COMMON START/STOP is derived from the beam pulsing system) in exactly the same way. The amount of shift in ns will be exactly the same for all the TDC parameters. However, in terms of channel numbers, it will vary, depending on the time calibration of each TDC input.

If you have completed the offline time calibration of all the TDC inputs then you can check this carefully - this is recommended.

Notice that in the T1_T2 plot the shift goes at an angle of approximately 45° (why?)

Similarly different shifts will appear in the gPos values in gPos.txt when we compute them using *Prog003.cpp* on the three data files. The shifts will differ slightly from detector to detector because of differences in the TDC calibrations.

Time reference shift in the next file



In the next file Data02.001 the beam has been tuned. The phase lock is now more stable. We see mostly one band.

But the T1_T2 cut made for previous file, is slightly shifted. You may think of going ahead, but in your data the shift could be more. In this Lesson we demonstrate the correct procedure to deal with these time reference shifts.

I have restructerd all the Program files. They are now names Analysis instead of Program
Use these new files for analysis where there is a possibility of time reference shifts.

Step 1 Analysis001.cpp

TFile *Tf=new TFile("Data01.001"); //Change this as required
Compile: g++ -o Analysis001 Analysis001.cpp `root-config --cflags --libs`
Run: ./Analysis001 Type Cntl-C to exit
This will produce file ParaList.txt
xEdit to change the order and remove unused parameters.
Do not change parameter names.
It would have been better to remove P8351,T8351,P6135,T6135
and take NDet = 48 instead of 50 (but I forgot to do that)

Step 2

Prepare FileList.txt
I have Data01.001, Data02.001 and Data03.001

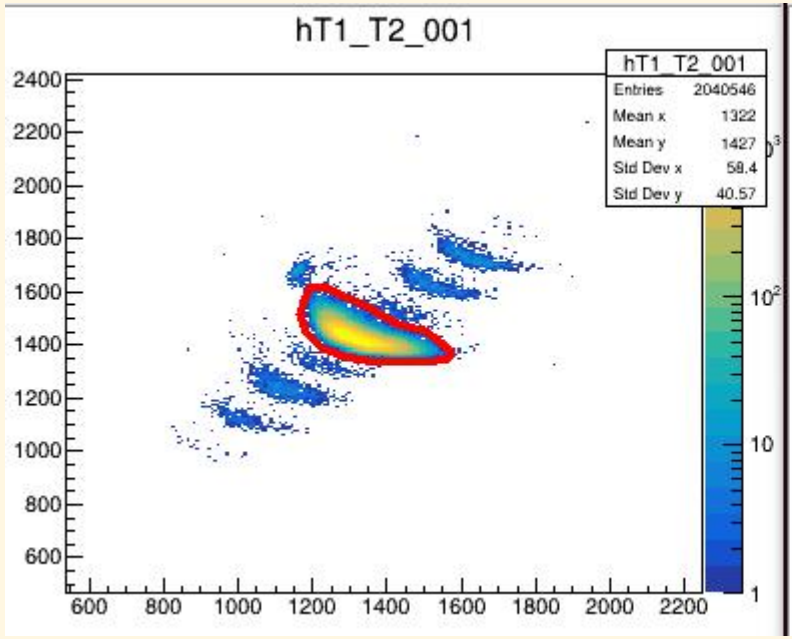
Analysis002.cpp (run as macro, do not compile)
$root Analysis002.cpp
Files are not in a chain. T1_T2 plots for each file in separate canvas
For each canvas draw and save into Work.root
Type on the command line:

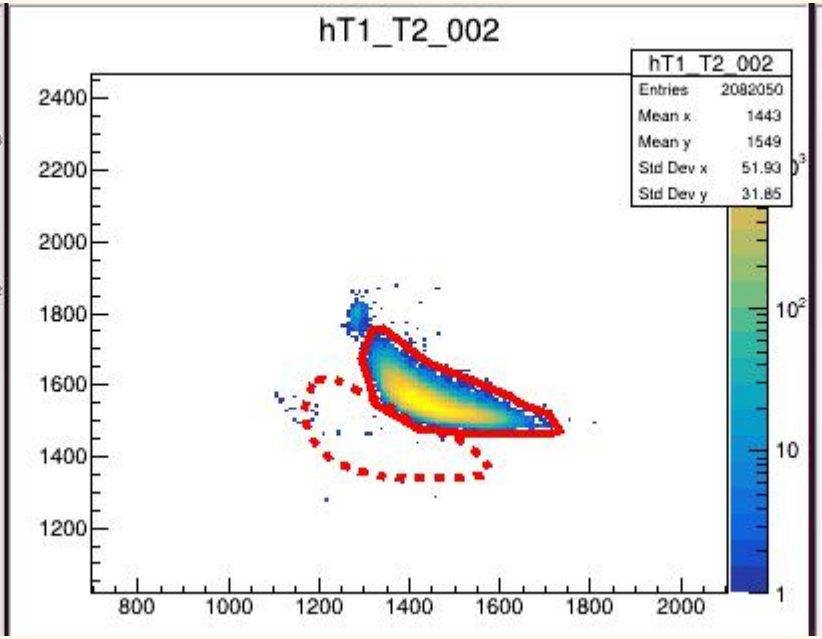
```
TFile *Tf = new TFile("Work.root","recreate");
//recreate because it is first time with Work.root
//Now draw a TCut on the first canvas
CUTG->SetName("CutT1_T2_001"); CutT1_T2_001->Write();
//Then repeat for each canvas
Tf->Close();
```

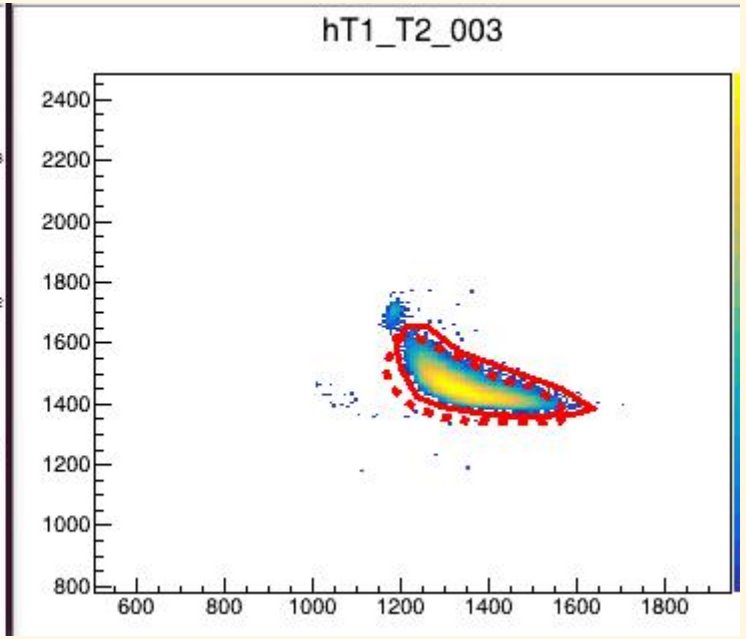
You can check how much the time reference has shifted

Checking the TCut shifts

Plot CutT1_T2_001 on the other 2 canvasses:
c002->cd(); CutT1_T2_001->Draw();
c003->cd(); CutT1_T2_001->Draw();
Right click on TCut SetLineAttributes to get dashed line







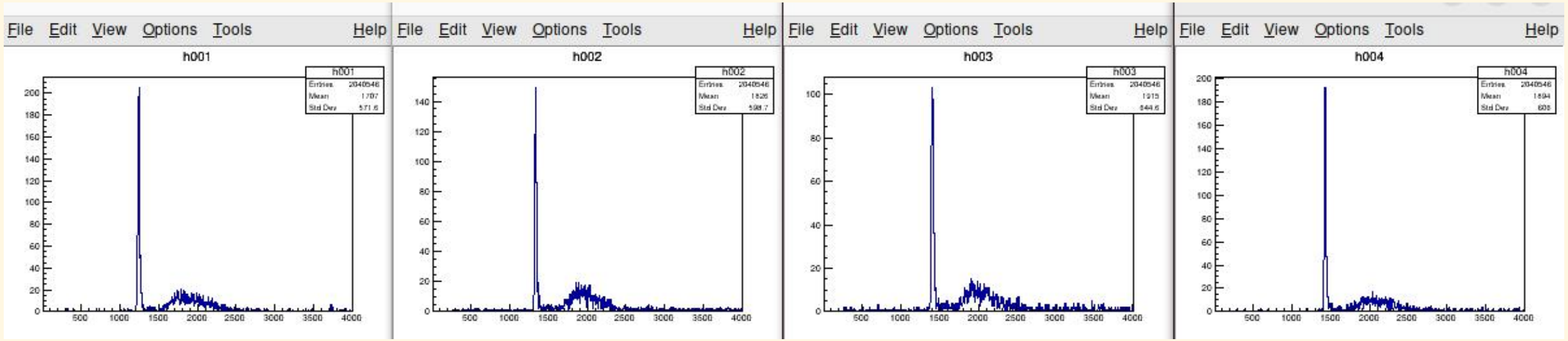
First file

Second file:
Large shift in time reference

Third file:
Small shift

Step 2a Analysis02a.cpp

This step is only to help you visualise
Analysis02a.cpp
TFile *Tf=new TFile("Data01.001"); //Change as required
Change the file name and run again to examine all the Data files
Compile: g++ -o Analysis02a Analysis02a.cpp `root-config --cflags --libs`
Run: ./Analysis02a



Step 3 Analysis003.cpp

Determine the gamma peak position of all the detectors
for each Data file separately (no chain of files)

Compile: g++ -o Analysis003 Analysis003.cpp `root-config --cflags --libs`
Run: ./Analysis003

This will result in files gPos01.txt, gPos02.txt, gPos03.txt
corresponding to Data01.001, Data02.001, Data03.001
Each file will contain NDet lines giving the positions of gamma peaks
Earlier we had only one gPos.txt file, but now the time reference is shifting from file to file

Step 4 Analysis004.cpp

Compile: g++ -o Analysis004 Analysis004.cpp `root-config --cflags --libs`
Run: ./Analysis004
Builds and saves all NDet two-dimensional PSD vs TOF spectra in Work.root
taking data from all the data files, but not in a chain
The time shifts are corrected by reading the GPos files

```
y=P[2*iDet+11]-(GPos[iDet][iFile]-GPos[iDet][0]);
```

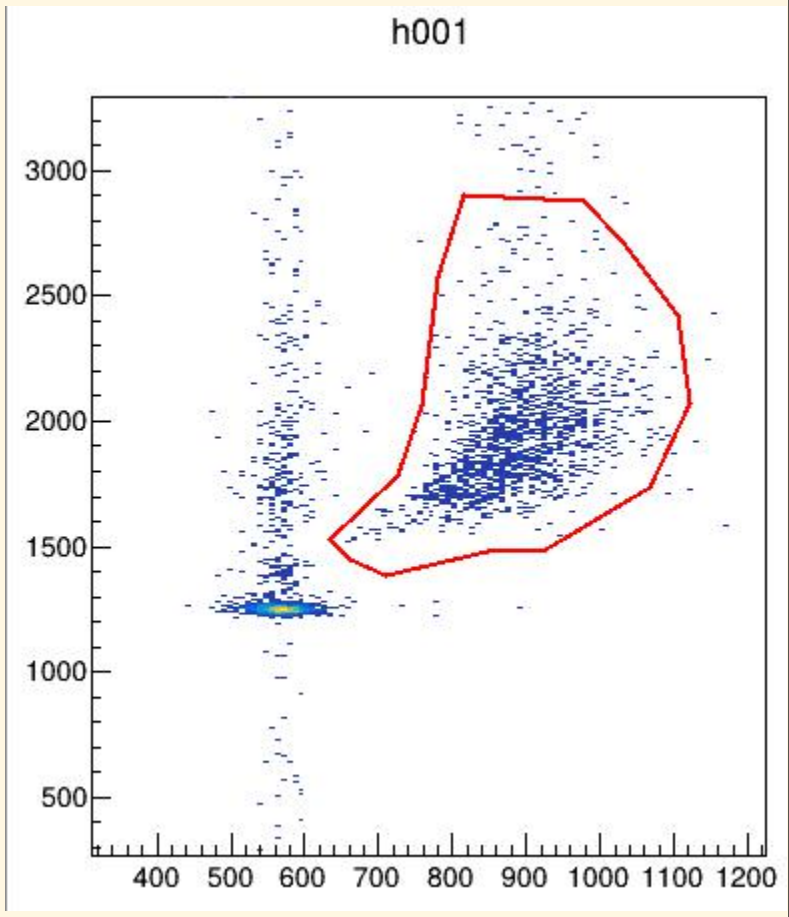
Thus we can draw the neutron selection cuts for the combined data and not file by file
Note that filtering with CutT1_T2 is applied (which we had not done before)

- Clean the data with CutT1_T2 to avoid randoms
- Combine data from all the Data files so as to have good statistics
- ***Drawing the TCut carefully is important for low energy as well as high energy neutrons, leading to better moving source fits***

Step 5 Manual operations from root prompt

Draw and save all the NDet TCuts into Work.root
This is the most time consuming step. Thanks to our programatic shifts we dont have to darw new TCuts for each file.
We will do this from the ROOT command line

```
$root
  TFile *Tf = new TFile("Work.root","update");
  TH2S *h=(TH2S *)Tf->Get("h001"); h->Draw( )
        //Now draw the TCut on the canvas
CUTG->SetName("CutN001"); CutN001->Write( );
          //Then repeat for all the NDet detectors
At the end: Tf->Close()
.q
```



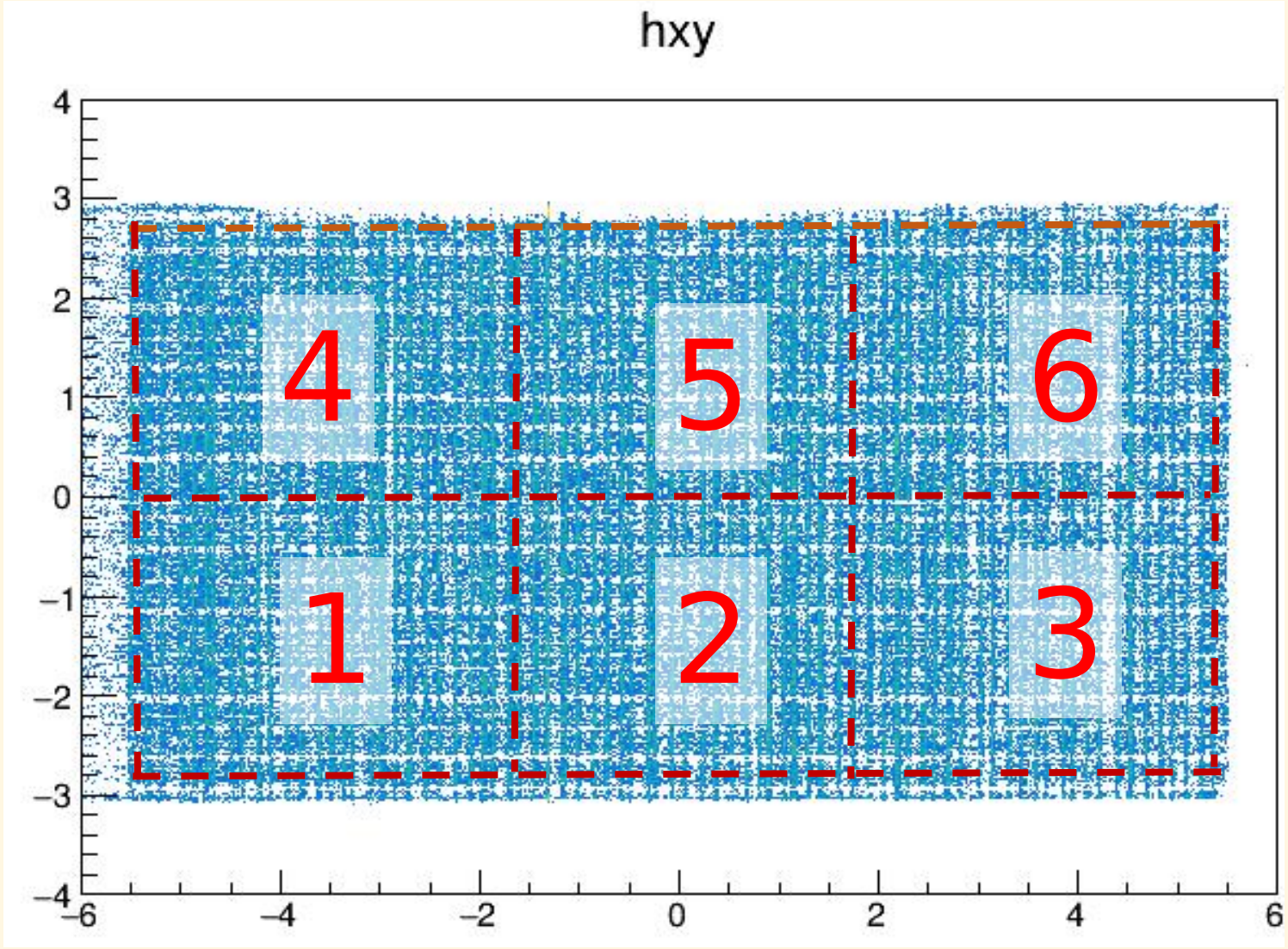
Step 6 Analysis005.cpp

Run as compiled program
Changed from earlier. Now instead of chain we process individual data files in a loop so that we can apply different TCutT1_T2 for each file

Please change the MWPC calibrations according to your electronics and the size of the MWPC

This will produce the plot in the next slide

Divide into 6 slices 3 x 2



Slice data for 6 slices

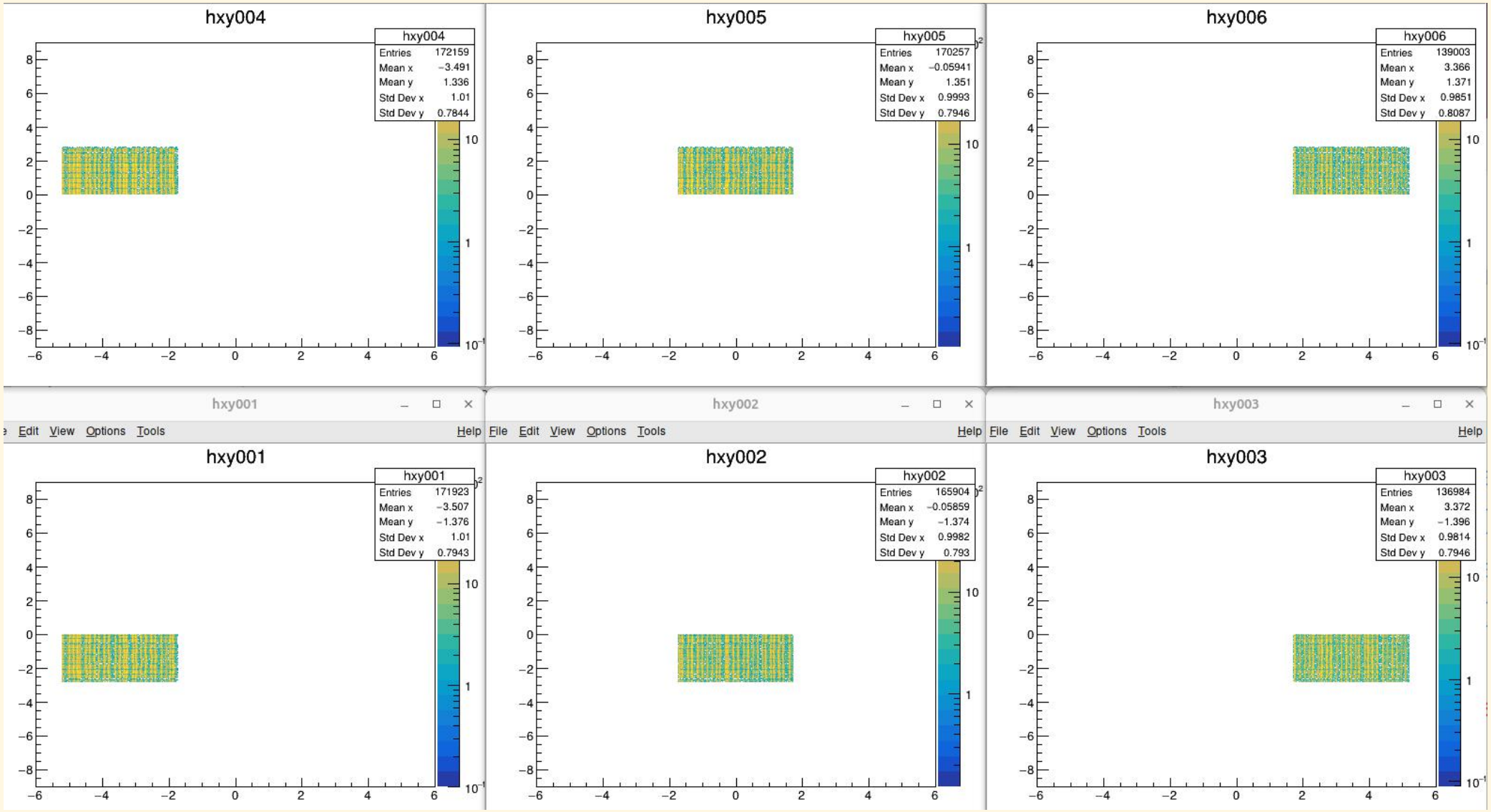
```
Slice 1:   -5.20<x<-1.73, -2.8<y<0
Slice 2:   -1.73<x<1.73,  -2.8<y<0
Slice 3:    1.73<x<5.20,  -2.8<y<0

Slice 4:   -5.20<x<-1.73, 0<y<2.8
Slice 5:   -1.73<x<1.73,  0<y<2.8
Slice 6:    1.73<x<5.20,  0<y<2.8
```

Put these numbers in SliceData.txt

Step 7 Analysis006.cpp

Prepare SliceData.txt. Compile and run Analysis006.cpp. It has been restructered to run file by file changing the CutT1_T2. The diaplay below will be seen. This step is only for visualisation.



Step 8 Analysis007.cpp

Final step.
Neutron Energy spectra for each detector coresponding to each slice.
NDet x NSlices spectra saved to ascii file for moving source fit

Analysis007.cpp restructred from earlier Program007.cpp
Does not chain the files, but processes files in a loop
You will need to change tCal.txt (calibration of TOF channels)
Different cuts TCutT1_T2 applied separately for each file
Only one set of neutron selection TCuts (CutN001, CutN002) but time corrected TOF values (different shifts for each file) while testing the gate.
Neutron Energy calculated in the same way

This time I have also placed all the final neutron energy spectra as root histograms in NeutronSpectra.root in addition to NeutronSpectra.txt so that we can examine them.

Summary

The analysis of NAND data where the time reference is shiting from file to file does not make it more difficult to analyse

- All the programs are changed to run over a loop over data files instead of a chain
- The T1_T2 cuts are first drawn separately for each file (this is the only extra labour)
- The positions of the gamma ray peaks are saved for each data file separately. This does not require any extra effort as the program has been modified.
- The labour intensive work of drawing cuts for neutron selection is to be done only once.
- For building neutron spectra, the program automatically calculates the time reference shift and applies it while testing the neutron selection TCuts.

At the end

It may happen that a run contains several time reference shifts. If only one band contains most of the data, then we have seen here how to go ahead with just that one dominant band.
But what if data is distributed in two or mode

But what if data is distributed over two or more bands.
We can still analyse all the data without having to modify any of our root programs.
Simply repeat the same data file again in FileList.txt but draw a new T1_T2 gate for it!