

Neutron Detector Array NAND Lesson 2

Ambar Chatterjee



Series of video lectures about ROOT
analysis of pre-fission neutrons from the
NAND Array at IUAC, New Delhi

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

WRITING STRUCTURED ROOT PROGRAMS

Creating around 600 neutron spectra in coincidence with fission

I assume you are familiar with ROOT. Otherwise first see my ROOT Channel

<https://www.youtube.com/@CERN-ROOT-ambar>

We will work with a sample NAND data root file called **IUAC_NAND_data.001**. In your analysis there will be several files corresponding to a projectile-target system and beam energy. All these files would be added into a TChain.

Since actual data files are very big I am providing only one small sample file with 73 neutron detectors for demonstration.

The program source code files are packaged in nand02.tgz

The files can be downloaded from my website.

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



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 	Click
Neutron Array 	Click

Videos

NAND Lesson 1	MWPC Slicing, Angle Calculations	Click
NAND Lesson 2	MWPC Slicing, Angle Calculations	Click

Support Files

NAND Lesson 1	nand01.tgz	
NAND Lesson 2	IUAC_NAND_data.001	nand02.tgz

First task: find the name of the tree and leaves.

You know TBrowser or from CLI `.ls` and `TreeName->Print();`

The tree is RoseNIAS (use the last version)

Here we demonstrate a program that saves the list of leaves in a file

Prog001.cpp

```
g++ -o Prog001 Prog001.cpp `root-config --cflags --libs`  
Creates file ParaList.txt
```

This program uses `Tr -> GetListOfLeaves`

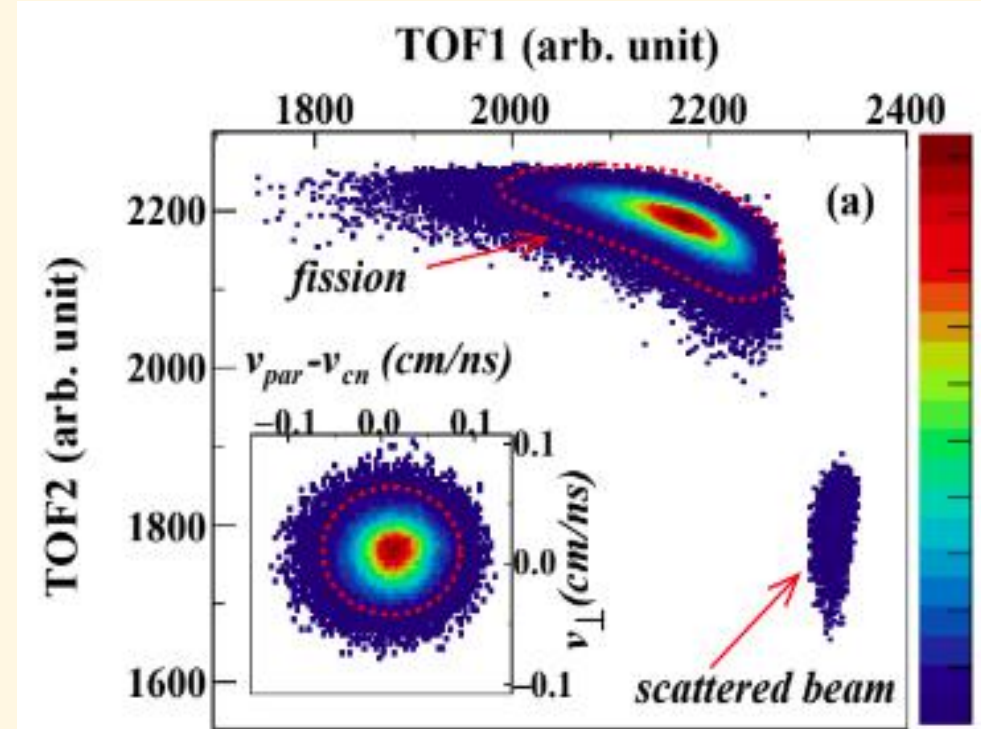
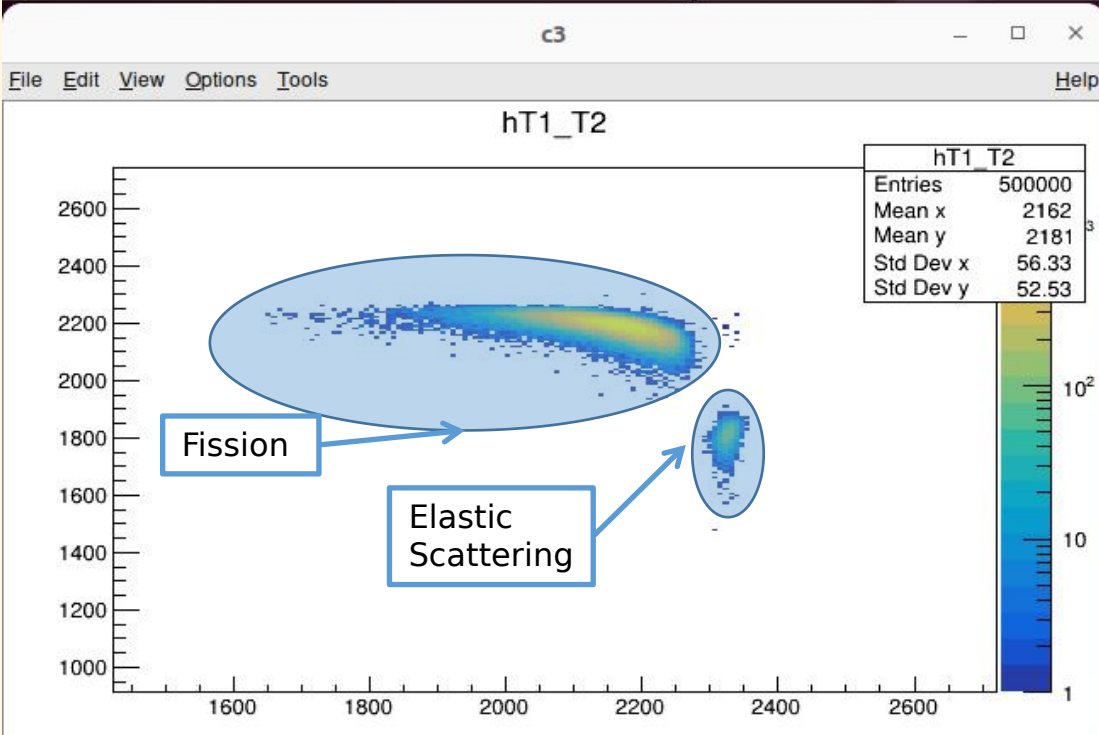
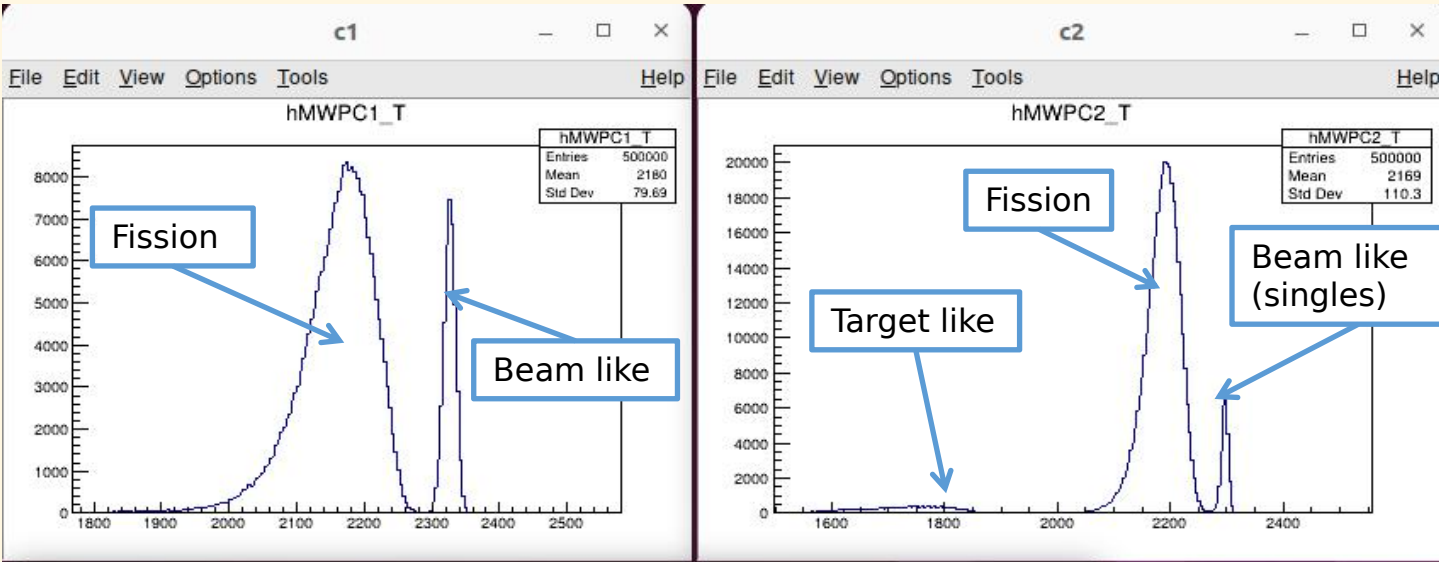
The elements of the list can be accessed by `->At(i)`

and the leaf names by `->GetName()`

After ParaList.txt is created we can refer to parameters by number. This is our first step for adding structure to our codes.

Prog002.cpp short structured code to display spectra

Fission gating from MWPC1 and MWPC2



Saneesh et al
Impact of multichance fission on fragment-neutron correlations in ²²⁷Pa
Phys. Rev. C 108 (2023) 034609

Drawing and Saving CutT1_T2

To do this we will run Prog002.cpp again not as a program but as a Macro

```
$ root Prog002.cpp or
```

```
$ root
```

```
root[0] .x Prog002.cpp
```

Draw the CUTG for fission. Then execute the following from the command line

```
CUTG -> SetName("CutT1_T2");
```

```
CutT1_T2 -> SetVarX("MWPC1_T");
```

```
CutT1_T2 -> SetVarY("MWPC2_T");
```

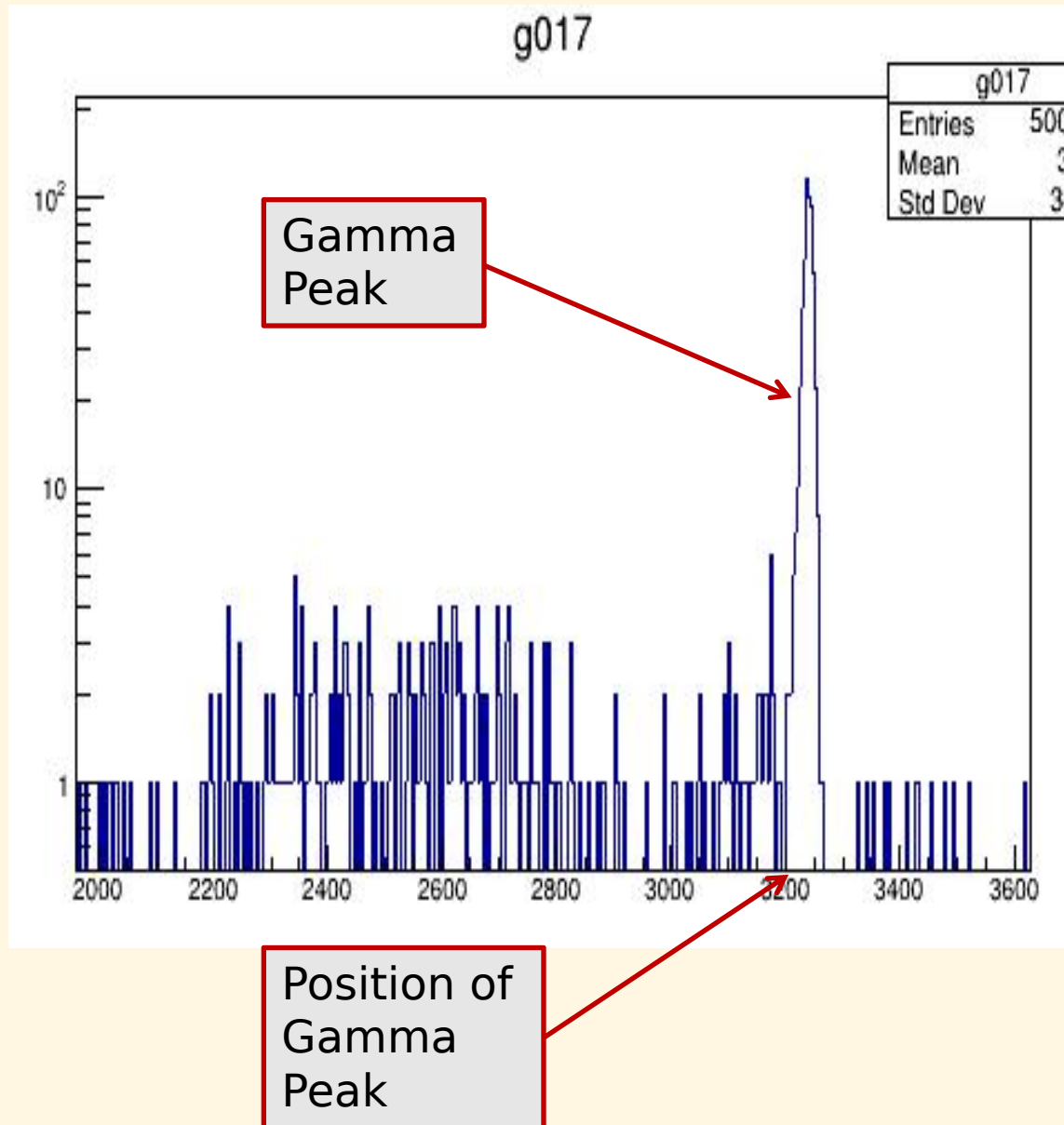
```
Tf = new TFile("Work.root", "recreate");
```

```
CutT1_T2 -> Write();
```

```
Tf -> Close();
```

We will use Work.root to save all our cuts and histograms

Positions of gamma peaks in all the neutron detectors



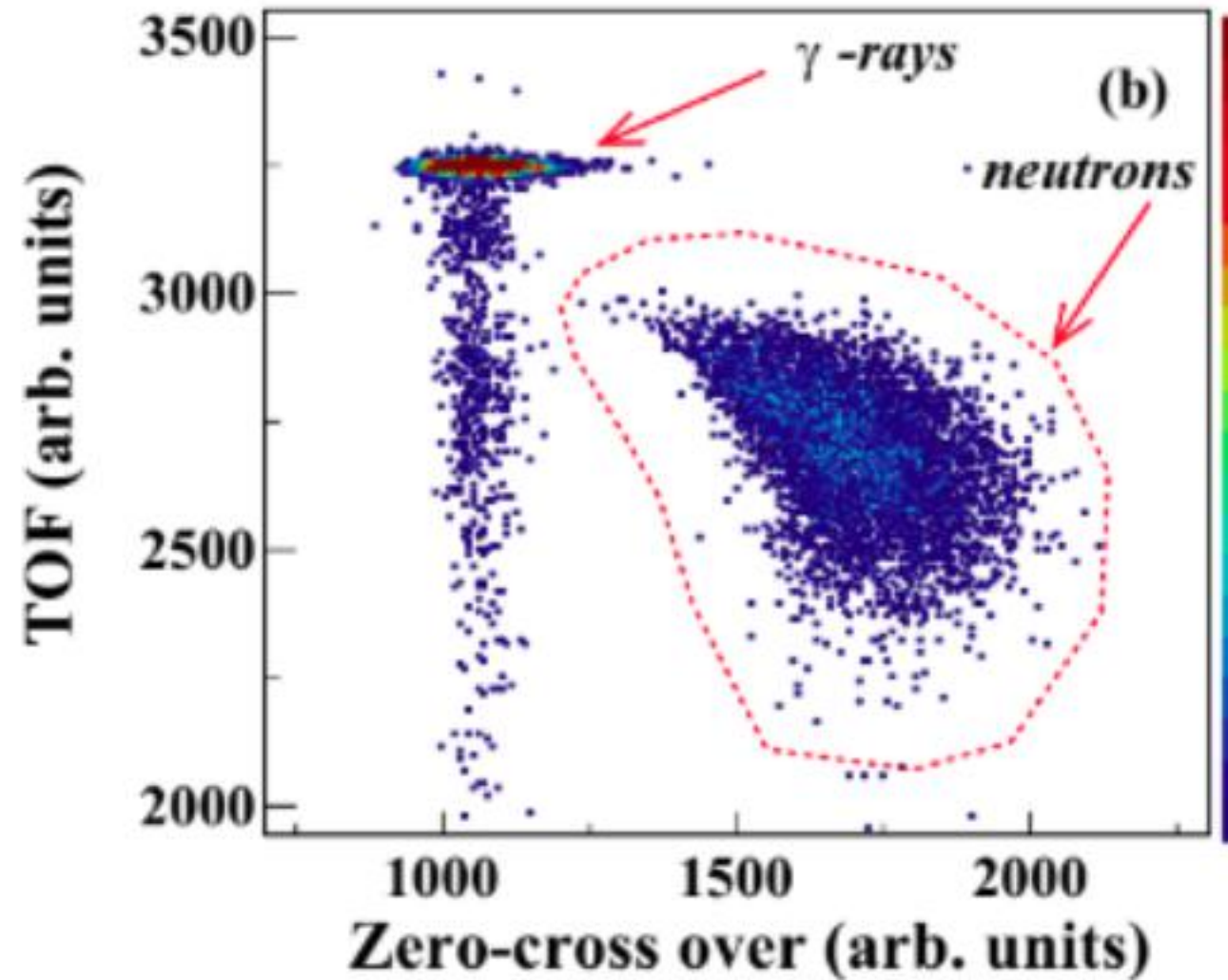
To find the neutron time of flight we need the position of the gamma peak in all the TOF spectra.

The TOF spectra looks like the picture at left.

Prog003.cpp will bulid all the TOF spectra, determine the gamma peak position and save the results in a file gPos.txt

Neutron Selection

Neutrons are to be selected by a TCutG in PSD vs TOF for all the neutron detectors



Saneesh et al
Impact of multichance fission on fragment-neutron correlations in ^{227}Pa
Phys. Rev. C 108 (2023) 034609

Build and Save all PSD vs TOF

Prog004.cpp

Structured program, saves all NDet
two-dimensional PSD vs TOF spectra
inside Work.root

Drawing and saving all the NDet TCuts into Work.root

This is a labourious process and will take time

It can also be error prone

We will do this work from the ROOT command line

```
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 detectors  
Tf->Close()  
.q
```

Look forward to the next Lesson