

Beamline for Schools 2018

Data processing and analysis

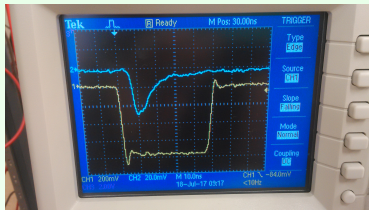
Ina Carli

Institute of Particle and Nuclear Physics
Charles University

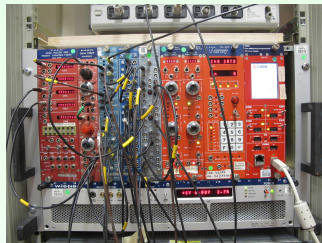


... or how to go

from



trough



to

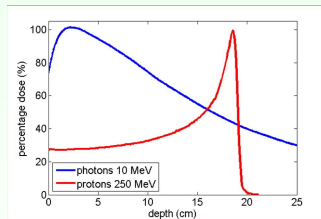
```

# Create histograms to hold the reconstructed position with the two different methods
# Cluster = ROOT::TH2D("cluster", "Hit Position with Peak Algorithm", 0, 0, 100, 100, 100, 100)
# Cluster = ROOT::TH2D("cluster", "Hit Position with Cluster Algorithm", 0, 0, 100, 100, 100, 100)

# Technically, the below should be repeated for each micrograph detector. For simplicity, here we treat only one
for event in range(1000):
    # Get the peak strip index (remember to add the offset, as explained above)
    peakStrip = event.BL4000_strip_index + event.BL4000_strip_offset
    # Convert the peak strip index into a position
    peakPosition = peakStrip * pitch - offset
    # Save it into the histogram
    cluster.Fill(peakPosition)

# Loop through the recorded strip indexes and perform weighted sum
Qsum = 0
for i in range(event.BL4000_strip):
    Qsum = Qsum + event.BL4000_strip[i] * event.BL4000_strip[i]
    if Qsum == 0:
        continue
    # Get the pseudo-range from the cluster algorithm (remember to add the offset, as explained above)
    clusterIndex = findIndex(Qsum + event.BL4000_strip_offset)
    # Convert it into the cluster position
    clusterPosition = clusterIndex * pitch - offset
    # Save it into the histogram
    cluster.Fill(clusterPosition)
  
```

and



... and to



?

Tools

- ROOT - software for data analysis used in particle physics, root.cern.ch
 - can process, store, visualise or fit data
 - uses C++ or python in pyROOT
- Jupyter - web application that allows to run and share code in browser
 - single programs - notebooks
- SWAN - CERN service to run Jupyter notebooks
swan.cern.ch
- CERNbox - cloud service with your data
cernbox.cern.ch



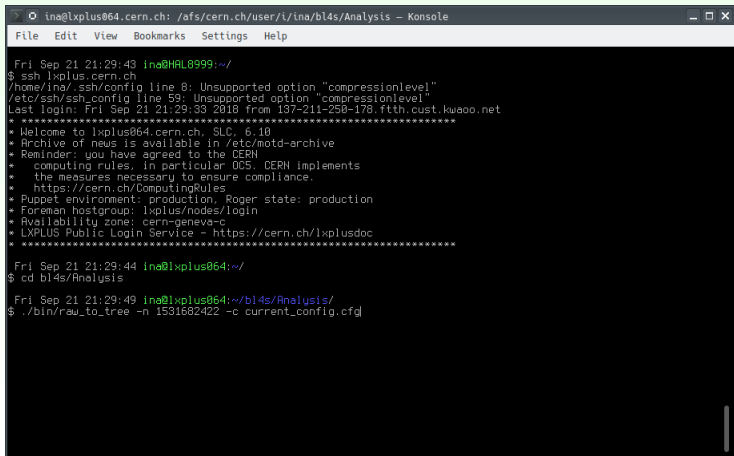
Data flow

Data flow: very raw data

```
1234CCCC 00000004 00000002 00000188 EE1234EE 00000009 03010000 00510054
5B48ABC4 00000001 00000001 00000000 00000000 00510004 00000400 00000010
00000003 00000000 00000003 00000002 00000002 00000000 00000000 00000000
00000034 0000000A 00000015 00000023 00000000 00000000 00000000 0002B115
00510001 00000300 00000022 FA012000 F8004071 F810408C F80141BD F8114072
F802423C F8124075 F803407C F8134086 F804408C F814407C F805407B F8154083
F8064091 F81640A3 F807406D F8174084 F8084094 F818408F F809407F F8194070
F80A4083 F81A408F F80B4090 F81B407A F80C4079 F81C4063 F80D4092 F81D4084
F80E4087 F81E406E F80F4092 F81F4096 FC04F3A9 00510003 00000100 4000003F
08001F44 000020B1 00402A2F 00A0230C 00C0263B 00202882 0060300A 0080269D
00E02780 00C06AAB 00E06C81 1800100C 09001F44 19001002 8000021F 00510005
00000100 4000003F 08001E47 18001002 09001E47 19001002 800000DF 00000000
00000000 00000004 00000052 00000001 1234CCCC 00000004 00000003 00000180
EE1234EE 00000009 03010000 00510054 5B48ABC4 00000002 00000002 00000000
00000000 00510004 00000400 00000010 00000004 00000000 00000004 00000003
00000002 00000000 00000000 00000000 00000040 0000000B 00000018 00000023
00000000 00000000 00000000 0002B1D6 00510001 00000300 00000022 FA012000
F8004072 F810408C F8014297 F8114073 F80241FA F8124075 F803407C F8134086
F804408E F814407D F805407A F8154082 F8064092 F81640A2 F807406D F8174085
F8084094 F818408E F809407E F8194070 F80A4084 F81A408F F80B4091 F81B407A
F80C4078 F81C4063 F80D4092 F81D4084 F80E4087 F81E406E F80F4092 F81F4097
```

Data flow: processing

- no need to read it!
- we have prepared scripts to produce .root files
- **TODO** on shift: process data that you collect



```
ina@lxplus064.cern.ch: /afs/cern.ch/user/i/ina/bl4s/Analysis – Konsole
File Edit View Bookmarks Settings Help

Fri Sep 21 21:29:43 ina@HAL0999:~/
$ ssh lxplus.cern.ch
/home/ina/.ssh/config line 8: Unsupported option "compressionlevel"
/etc/ssh/ssh_config line 59: Unsupported option "compressionlevel"
Last login: Fri Sep 21 21:29:33 2018 from 137-211-250-178.ftth.cust.kwaoo.net
*****
* Welcome to lxplus064.cern.ch, SLC, 6.10
* Archive of news is available in /etc/motd-archive
* Reminder: you have agreed to the CERN
*   computing rules, in particular OCS, CERN implements
*   the measures necessary to ensure compliance.
*   https://cern.ch/ComputingRules
* Puppet environment: production, Roger state: production
* Foreman hostgroup: lxplus/nodes/login
* Availability zone: cern-geneva-c
* LXPLUS Public Login Service - https://cern.ch/lxplusdoc
*****
Fri Sep 21 21:29:44 ina@lxplus064:~/
$ cd bl4s/Analysis
Fri Sep 21 21:29:49 ina@lxplus064:~/bl4s/Analysis/
$ ./bin/raw_to_tree -n 1531582422 -c current_config.cfg
```

Data flow: raw data

For each recorded event we have variables from all detectors:

- time of signal (TDC), integrated charge (QDC), counter

	t0	t1	t2	t3	q0	q1	q2	q3	c0	c1
1	624	660	439	615	810	412	551	369	532	53
1	727	317	355	797	313	819	477	621	864	332
2	835	-749	611	718	720	642	679	806	79	868
3	645	345	792	-316	260	361	553	523	847	769
4	302	561	-583	176	811	-319	598	703	938	203
5	940	200	242	492	895	552	825	159	466	596
6	711	739	698	590	641	799	673	-690	787	847
7	738	357	-819	855	55	-424	58	729	912	768
8	109	392	774	776	669	808	746	804	241	134
9	429	569	633	225	512	918	134	419	106	226
10	420	391	388	824	238	75	407	223	-902	566
...										

What do the numbers mean?

Data flow: root data

- know your detectors, units, direction (positive x?)
- select only events which have valid information
- compute physics variables: position, particle identification
- physics data - ready for analysis

	x0	y0	t0	x1	x2	t2	q0	q1	c0	e ⁻
1	183	33	119	101	744	268	74	705	936	0
1	841	260	522	757	39	942	545	220	881	0
2	979	64	422	613	319	928	723	963	127	0
3	922	366	234	876	452	928	548	585	403	1
4	981	891	64	957	709	450	694	335	190	1
5	335	765	516	237	110	78	350	74	280	1
6	380	459	465	997	998	78	368	746	705	0
7	261	790	239	784	104	416	165	964	620	0
8	212	529	403	292	115	411	83	876	369	0
9	130	796	11	14	475	663	290	776	202	1
10	893	743	270	688	66	85	5	286	296	1

...

ROOT analysis

ROOT analysis

- tree (class TTree) - object holding data
- ntuple - sequence of ordered values = table

```
root file.root
```

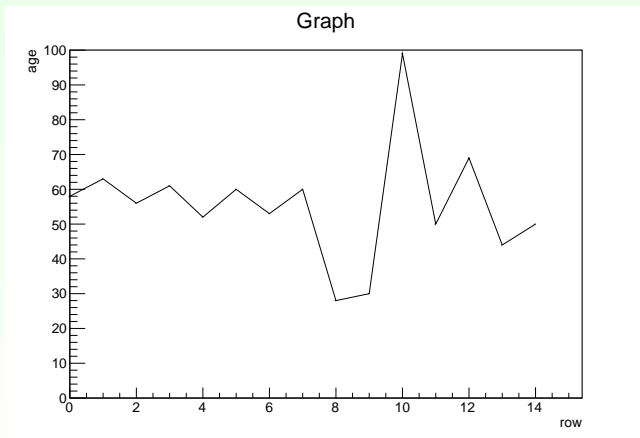
```
root [] T->Scan("Cost:Age:Children")
```

```
*****  
*      Row      *      Cost *      Age *      Children *  
*****  
*          0 *      11975 *      58 *          0 *  
*          1 *      10228 *      63 *          0 *  
*          2 *      10730 *      56 *          2 *  
*          3 *      9311 *      61 *          0 *  
*          4 *      9966 *      52 *          2 *  
*          5 *      7599 *      60 *          0 *  
*          6 *      9868 *      53 *          1 *  
*          7 *      8012 *      60 *          1 *  
*          . . .
```

Want to know more about age? Try to plot it!

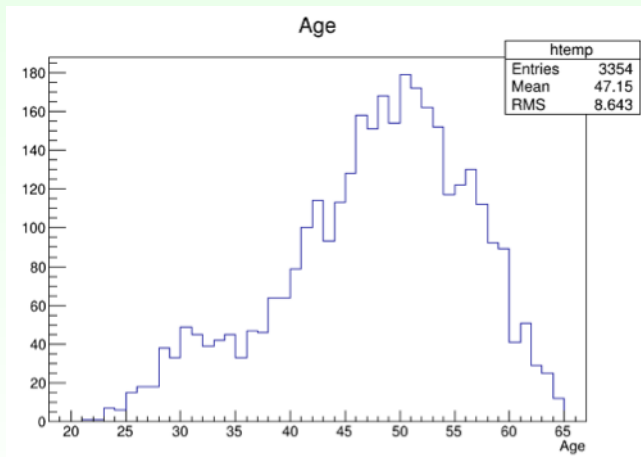
ROOT analysis

... function?

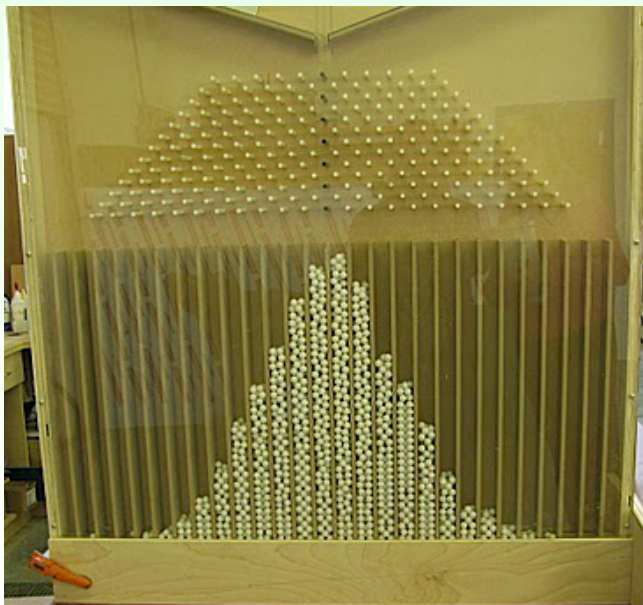


ROOT analysis

... 1D histogram!

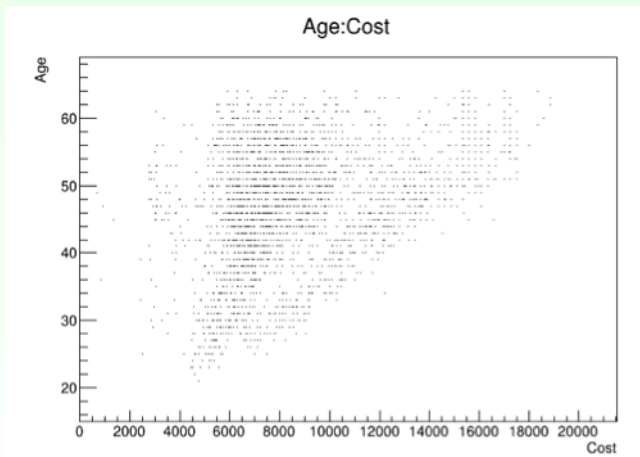


ROOT analysis



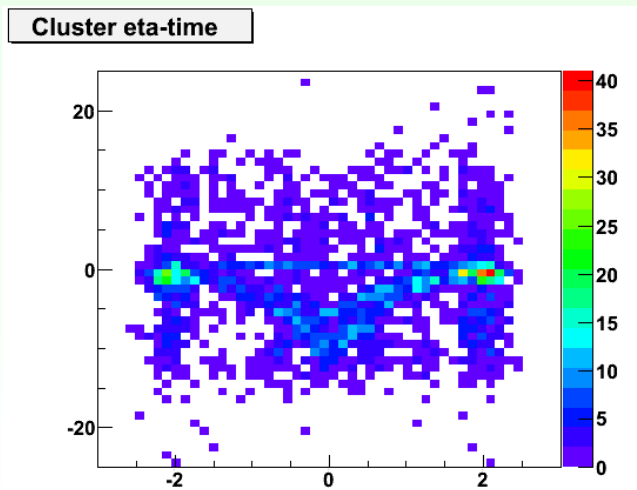
ROOT analysis

... 2D histogram!



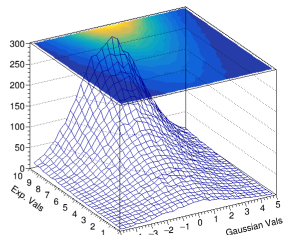
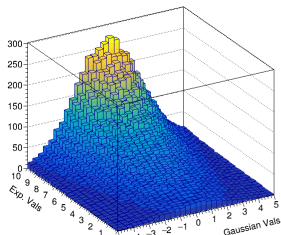
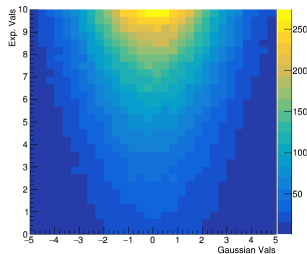
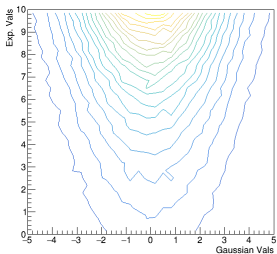
ROOT analysis

... fancy 2D histogram!



ROOT analysis

... other 2D histogram!

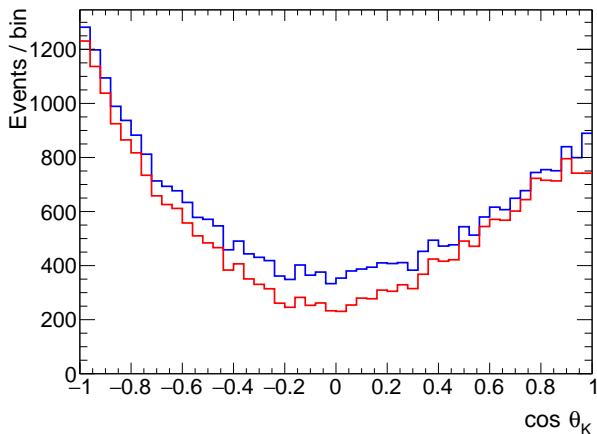


ROOT analysis

... apply cuts (=selection criteria):

blue - original histogram

red - fill only when other variable larger than x



ROOT analysis

Understand your plots and histograms:

- number of entries
- mean
- spread
- does it make sense?
-
- signal? background?
- error bars

In practice

- write program in Jupyter notebook on SWAN
- share code/results with the team, understand what you see in data

```
In [4]: import ROOT # load ROOT bindings. All ROOT classes are available e.g. as ROOT.TFile
        ROOT.enableJVis() # make plots interactive. You want that ;)

In [13]: import ROOT

        # Open ROOT file
        #f = ROOT.TFile("../hunt/15068576360dc_monitor.root");
        #run = "1506857636";

        f = ROOT.TFile("../hunt/QDC_hunt.root");
        run = "hunt";

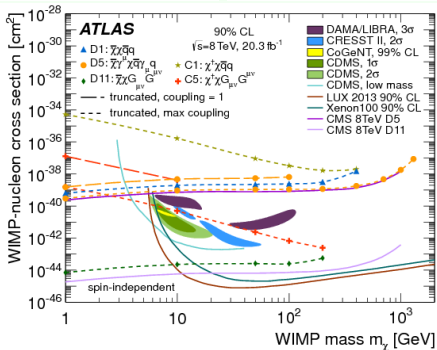
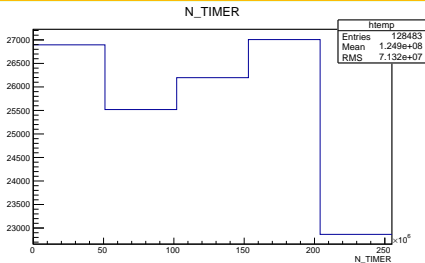
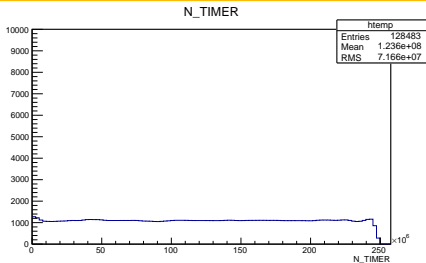
        fcpd1 = f.Get("FCPD1_corr");
        fcpd2 = f.Get("FCPD2_corr");

        c1 = ROOT.TCanvas()
        fcpd1.Draw()
        c1.Draw() # this actually displays the image
        c1.SaveAs("../plots/"+run+"_FCPD1.pdf")

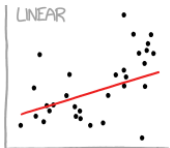
        c2 = ROOT.TCanvas()
        fcpd2.Draw()
        c2.Draw()
        c2.SaveAs("../plots/"+run+"_FCPD2.pdf")

Info in <TCanvas::Print>: pdf file ../plots/hunt_FCPD1.pdf has been created
Info in <TCanvas::Print>: pdf file ../plots/hunt_FCPD2.pdf has been created
```

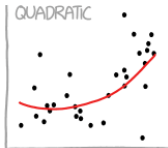
Making plots in an art



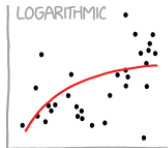
CURVE-FITTING METHODS AND THE MESSAGES THEY SEND



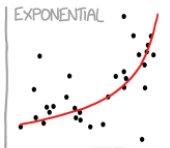
"HEY, I DID A
REGRESSION."



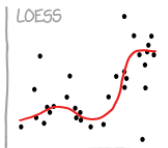
"I WANTED A CURVED
LINE, SO I MADE ONE
WITH MATH."



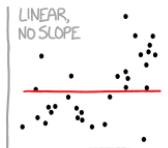
"LOOK, IT'S
TAPERING OFF!"



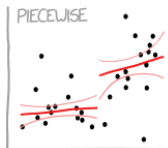
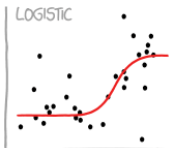
"LOOK, IT'S GROWING
UNCONTROLLABLY!"



"I'M SOPHISTICATED, NOT
LIKE THOSE BUMBLING
POLYNOMIAL PEOPLE."



"I'M MAKING A
SCATTER PLOT BUT
I DON'T WANT TO."



Final advice

- be curious and ask questions
- collaborate

And most of all, have fun!