BL4S 2020
Data Analysis

How to make the most of your Test Beam data

Cristóvão B. da Cruz e Silva
& Paul Schütze

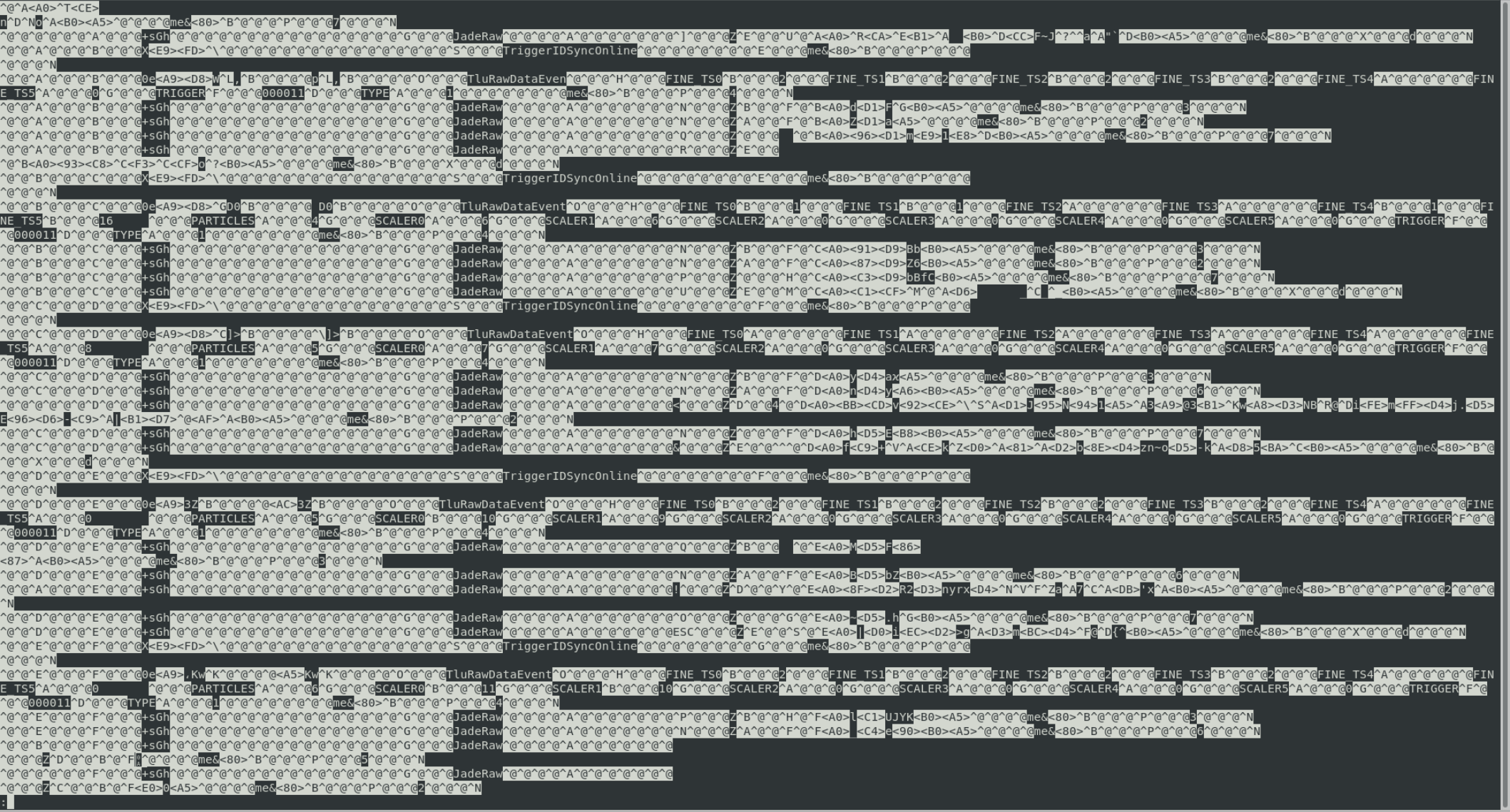
27.09.2020

Caveat

- We’ll try to give you an insight into what to consider for your test beam data analysis
- There is no recipe for **the** data analysis – you’ll have data from different detectors and want to derive different insights from that

  ➔ You will have to use different strategies

Origin: Raw data



Origin: Raw data

- The *raw data* is what is written to disk by the DAQ software
- It is usually not human readable and contains the compressed output of our detectors grouped in *Events* → bits and bytes

First step: Converting data

- The raw data can be converted into file formats that are more easy to read …
- This first analysis step will be performed by the shift crew
  - The converted data is then copied to computing clusters at DESY and CERN
- Two tools are used to convert the data from TDAQ and EUDAQ into similar data formats (ROOT TTrees)
  - You’ll get instructions at your shifts …

ROOT TTrees

- In a TTree, the measured values of our detectors are grouped
- It’s hard to put in words, but that’s roughly what it looks like …

- PixelHits
  - Responding pixel along **x** in Alpide_50
    - Event 1
      - 150
    - Event 2
      - 231, 232
    - …
  - Responding pixel along **y** in Alpide_50
    - Event 1
      - 322
    - Event 2
      - 431, 431
    - …
  - Responding pixel along x in Alpide_51
    - ...

ROOT TTrees

- In a TTree, the measured values of our detectors are grouped
- It's hard to put in words, but that's roughly what it looks like …

- From this data we can …
  a) quickly generate histograms/graphs
  b) look at details on an event-by-event basis

- PixelHits
  - Responding pixel along **x** in Alpide_50
    - Event 1
      - 150
    - Event 2
      - 231, 232
    - …
  - Responding pixel along **y** in Alpide_50
    - Event 1
      - 322
    - Event 2
      - 431, 431
    - …
  - Responding pixel along x in Alpide_51
    - ...

ROOT & Jupyter Notebooks

- To read the data we have to use the ROOT framework
- Jupyter notebooks are a nice environment to start with the analysis and/or programming itself
- CERN and DESY host *JupyterHub*s that have access to our converted data of one of your experiments each
  - They work pretty much the same way

  https://naf-jhub.desy.de https://swan.cern.ch/

- How to read ROOT Trees from Jupyter notebooks…?

Computing Accounts

- All of you get a CERN or DESY account that you can use for a few months
  - DESY account names: schoolXY (you may change the password)
  - CERN accounts: You have to change the password

- With these you can access the BL4S data and the Jupyter notebooks
- ✗ DESY: Please do not remove files or folders except for in your own *home directory*
- If you have doubts what you can do and what not, please ask

Where to go from TTrees ...

- For your analysis you will need to look at the signals from individual particles
  - Evaluating your data on an *event-by-event* basis can be useful (but it is usually slower)
  - You will need this to e.g. find tracks

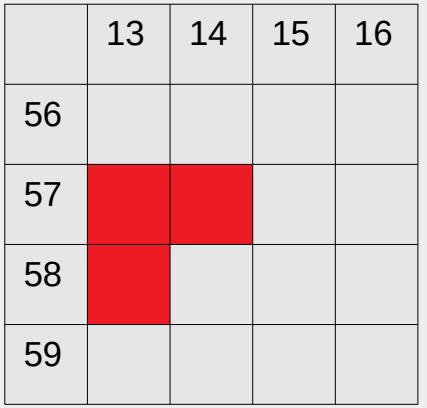
Where to go from TTrees ...

- For your analysis you will need to look at the signals from individual particles
  - Evaluating your data on an *event-by-event* basis can be useful (but it is usually slower)
  - You will need this to e.g. find tracks

- There are some typical *steps* that have to be performed in an analysis of this kind of data …

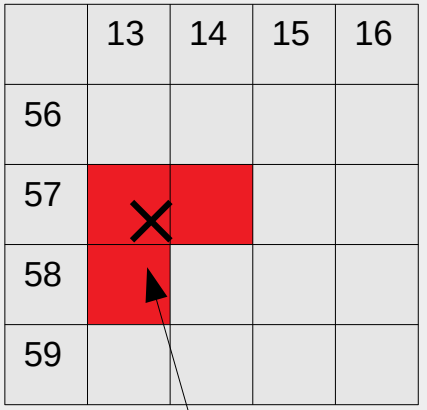
Clustering

- A single particle can leave a signal in more than one strip or pixel → *Cluster*



Clustering

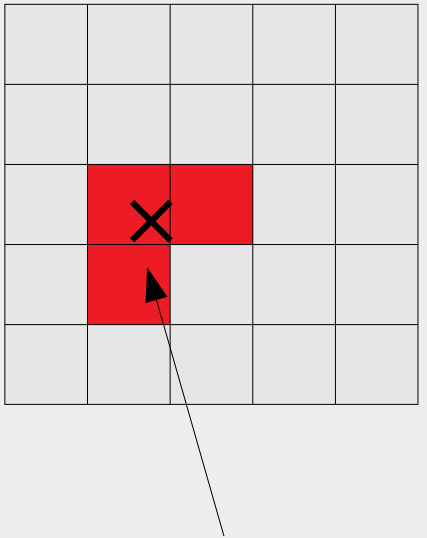
- A single particle can leave a signal in more than one strip or pixel → *Cluster*
- Task:
  - Find adjacent pixels/strips that contain a signal in the same event
  - Calculate the center of gravity
  - ➔ This is the most likely position of the particle traversal (in pixel coordinates)



x = 13.33
y = 57.33

Calculate the hit position

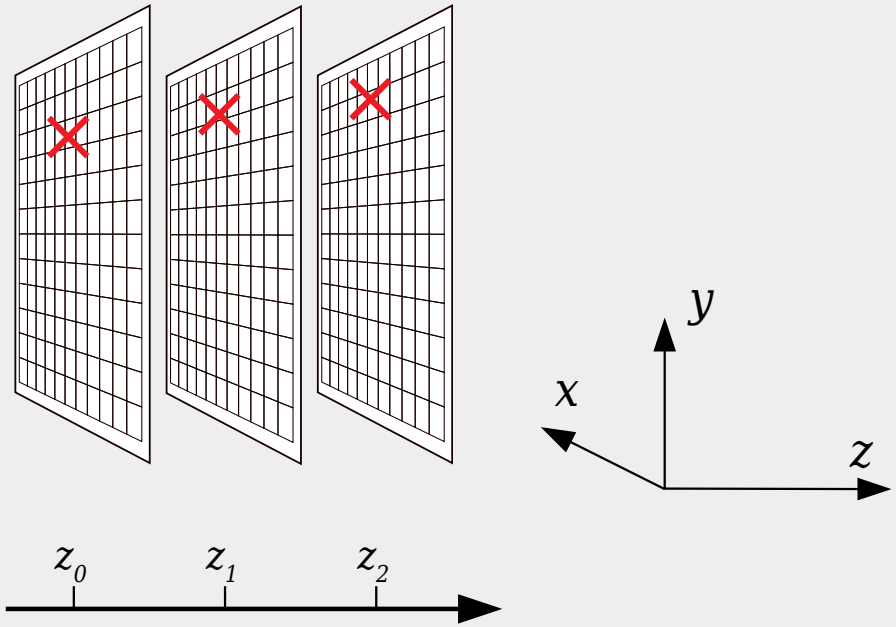
- From the known geometry and position of the sensor, calculate the actual position of the *particle hit*



x = -6.940 mm
y = -5.682 mm
z = 81 mm

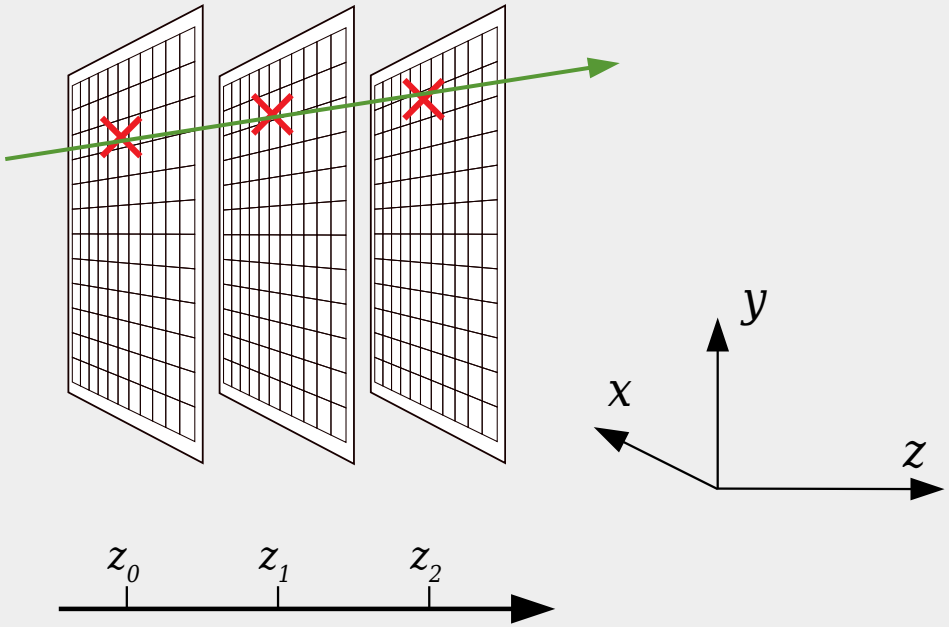
Tracking

- Look at the clusters from different sensors



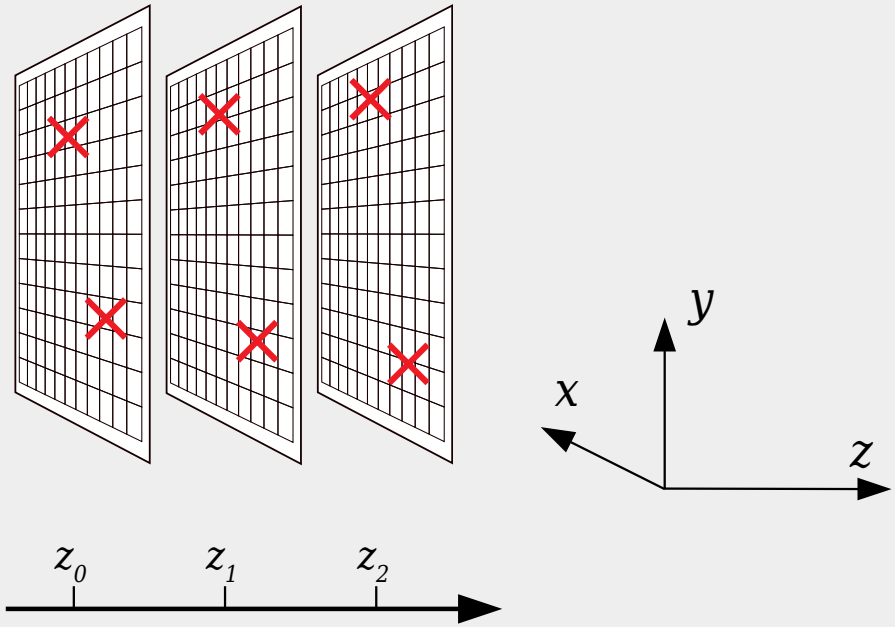
Tracking

- Look at the clusters from different sensors
- For most applications here, a fit to a straight line is a good approximation of the actual particle track



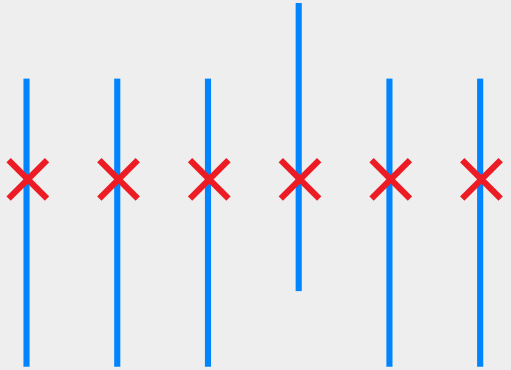
Tracking

- Look at the clusters from different sensors
- For most applications here, a fit to a straight line is a good approximation of the actual particle track
- What happens if there’s more than one particle in one event?



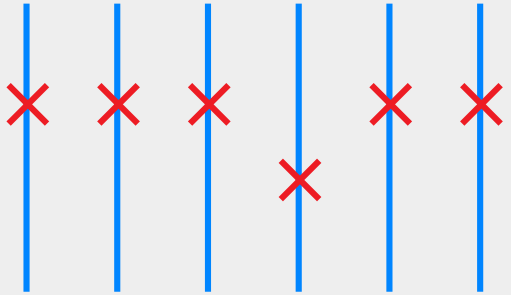
Alignment

- What if your sensor is not where you expect it to be?



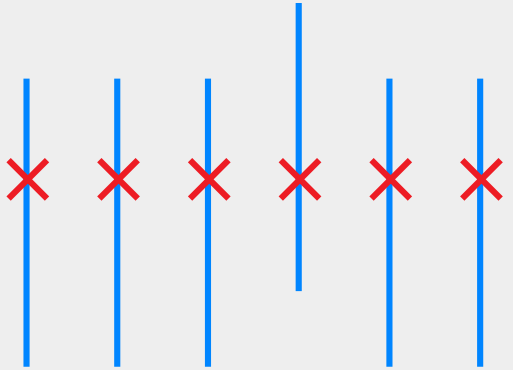
Alignment

- What if your sensor is not where you expect it to be?



Alignment

- What if your sensor is not where you expect it to be?
- You can correct for this in the data analysis → *Alignment*
- One approach:
  - Fix two detectors and calculate the line through them
  - Check the *residuals* of the other detectors
    - Where are the particle hits located with respect to the line?



There’s help

- You don’t have to do this on your own: We will guide you through your data analysis
- Work together & help each other out
- Many CERNies and DESYans volunteered for assisting you – they have experience in data analysis, ROOT, python, Jupyter (mostly…)
  - They will be in the analysis room or S’Cool lab most of the time
  - Feel free to ask them any question (analysis, physics, Hamburg, wisdoms)