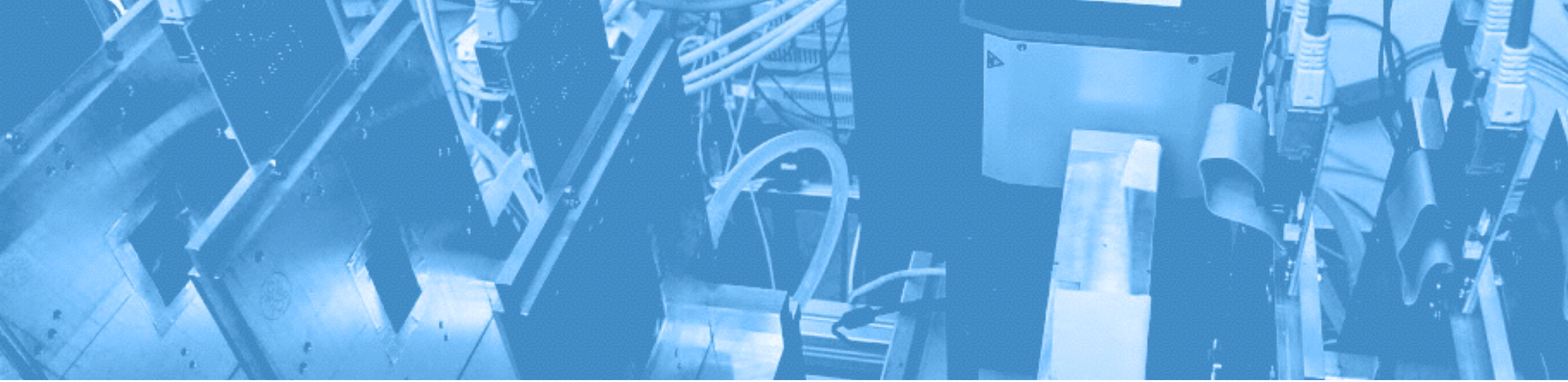


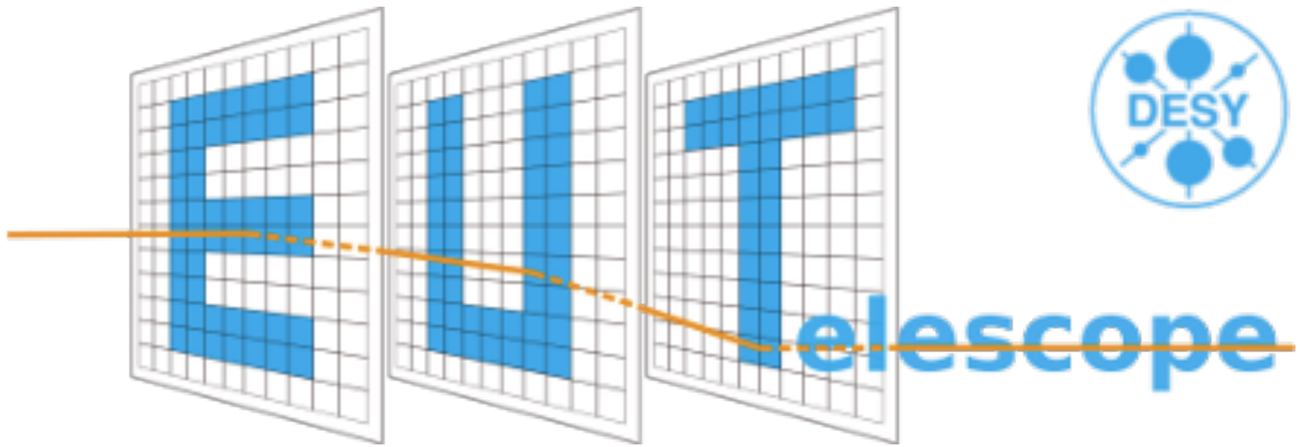
Scattering images using EUTelescope

Michaela Queitsch-Maitland
(DESY)

BTTB6
Zurich, Switzerland
18th January 2018

What is EUTelescope?

- EUTelescope is a framework for **reconstructing** and **analysing** data taken with pixel beam telescopes.
- Implemented as part of the EUDET project within the ILCsoft (ILC software) framework.
- http://eutelescope.web.cern.ch/



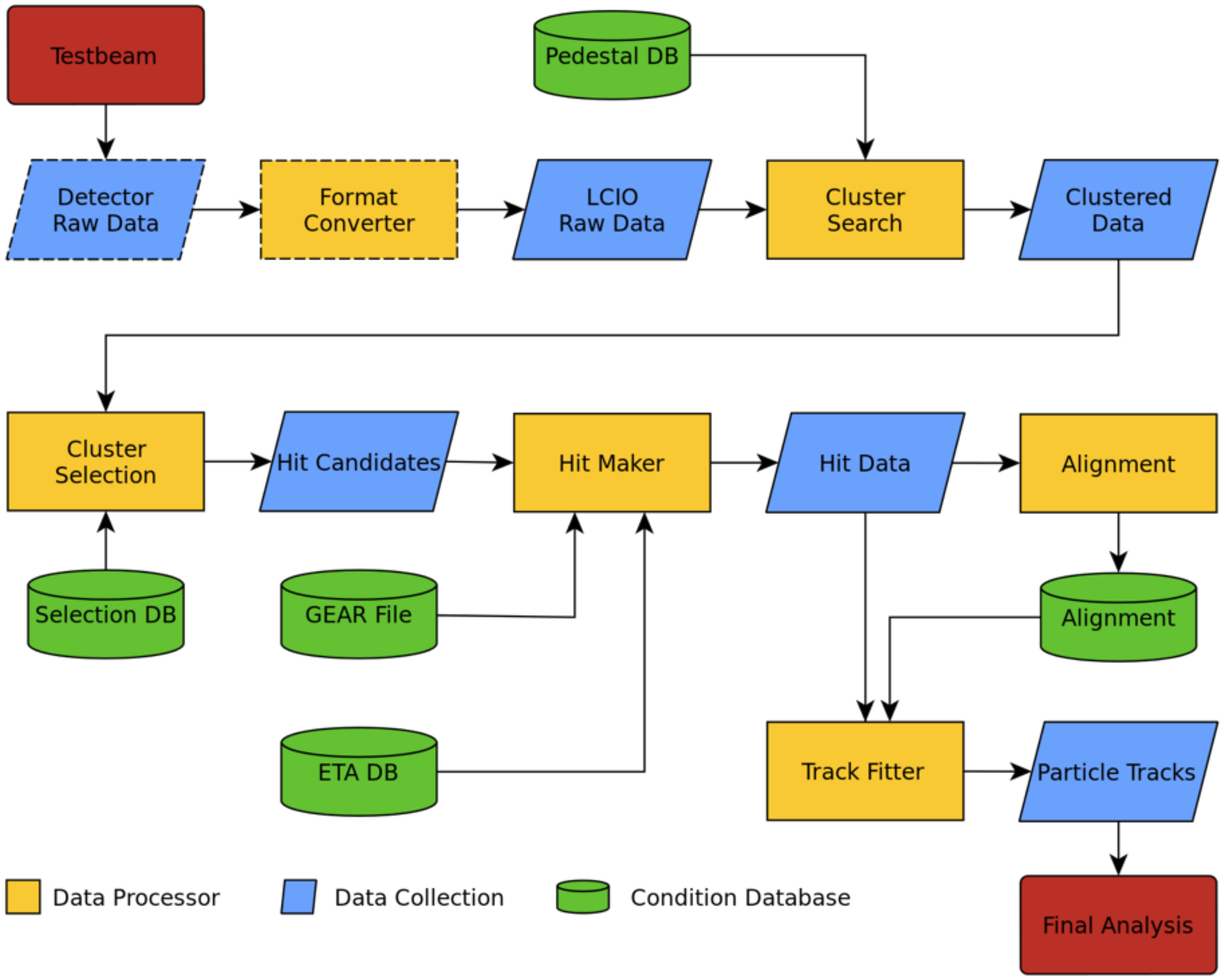
Outline of this tutorial

- **Goals**:
  - To show examples of how EUTelescope can be used to measure the scattering angle for a passive Device Under Test (DUT).
  - Hands-on experience of EUTelescope.
    - Interrupt and ask me questions at any time!

- **Caveats**:
  - All examples will use passive DUT (no data read out from DUT).
    - See other EUTelescope examples for DUT read-out.
  - I'm not an EUTelescope expert!
    - For general questions or problems, get in touch with the EUTelescope developers.
    - Git issue tracker: https://github.com/eutelescope/eutelescope/issues

Brief introduction to EUTelescope

- **EUTelescope** is a group of **Marlin** (Modular Analysis & Reconstruction for the LINear collider) *processors*.
- Embedded in the **ILCsoft** (ILC software) framework.
- Used for **reconstruction** and **analysis** of testbeam data.
- Uses the **LCIO** (Linear Collider I/O) *data model*.
- Telescope *geometry* described by **GEAR** (Geometry API for Reconstruction) markup language.

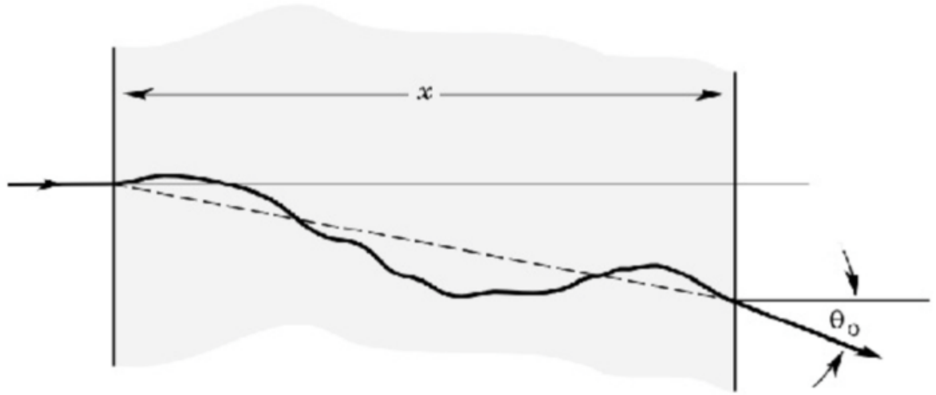


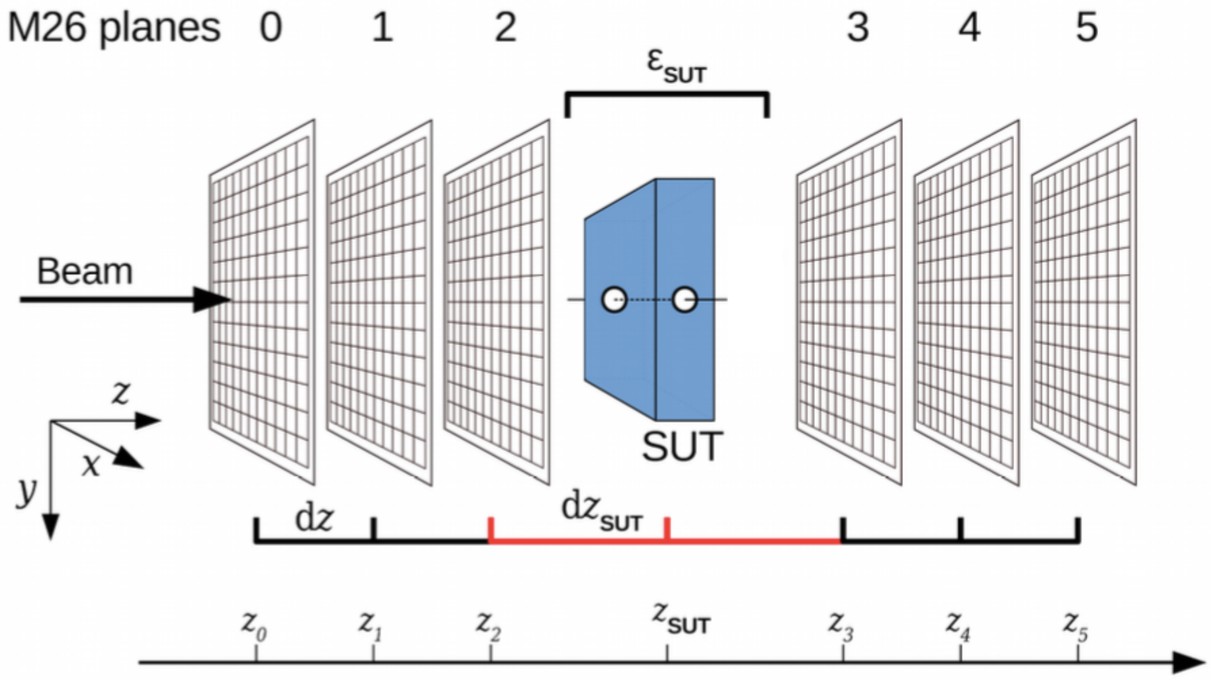
Components of an EUTelescope analysis

- The analysis is controlled by a **config** file (*.cfg), a **runlist** csv table (*.csv), and **steering file** templates (*.xml).
  - The config file specifies the configuration for the processors.
  - The csv table contains a list of runs and associated parameters.
  - Steering files control which processors/options are executed by Marlin.
- **Gear files** (*.xml) used to describe telescope geometry.

Scattering measurements

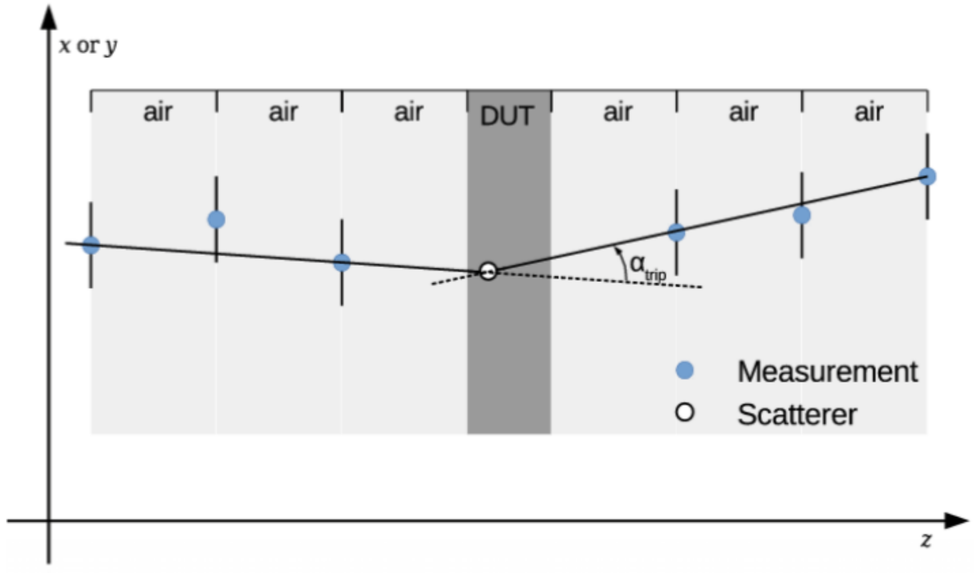
- Cover today examples of passive DUTs:
  - Measurement of scattering angles using a beam telescope.
  - Track finding and fitting performed using GBL in EUTelescope.
- For further details: see talks by Jan, Jan-Hendrik, and Hendrik.



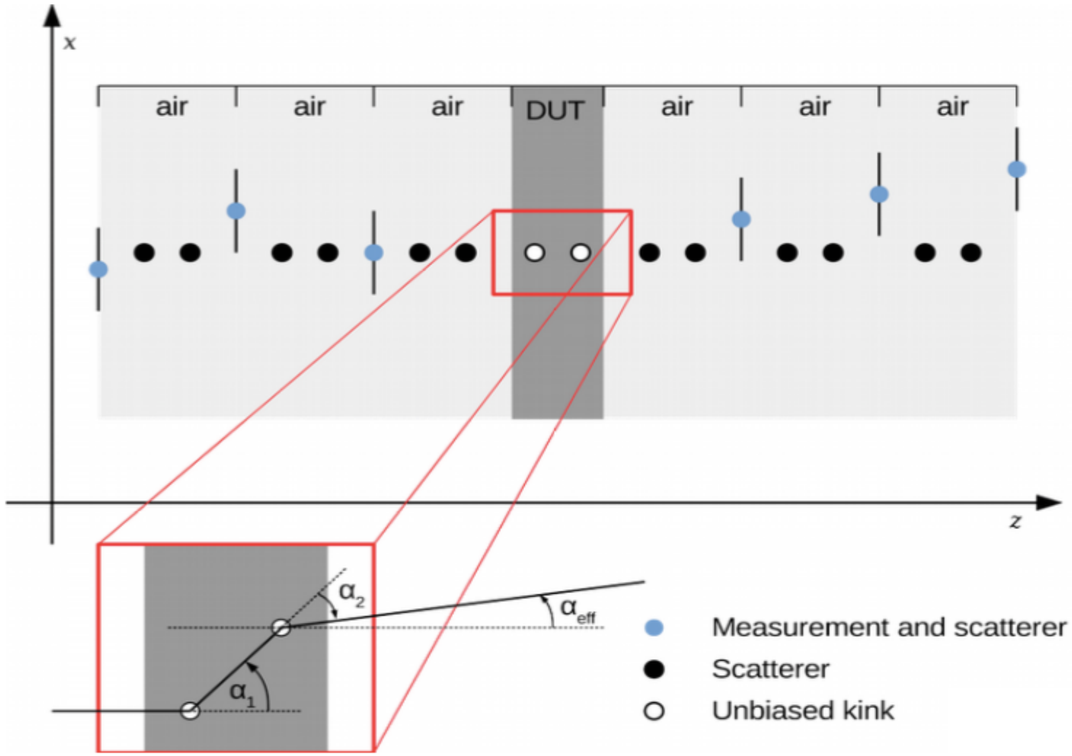


Scattering measurements

- Cover today examples of passive DUTs:
  - Measurement of scattering angles using a beam telescope.
  - Track finding and fitting performed using GBL in EUTelescope.
- For further details: see talks by Jan, Jan-Hendrik, and Hendrik.

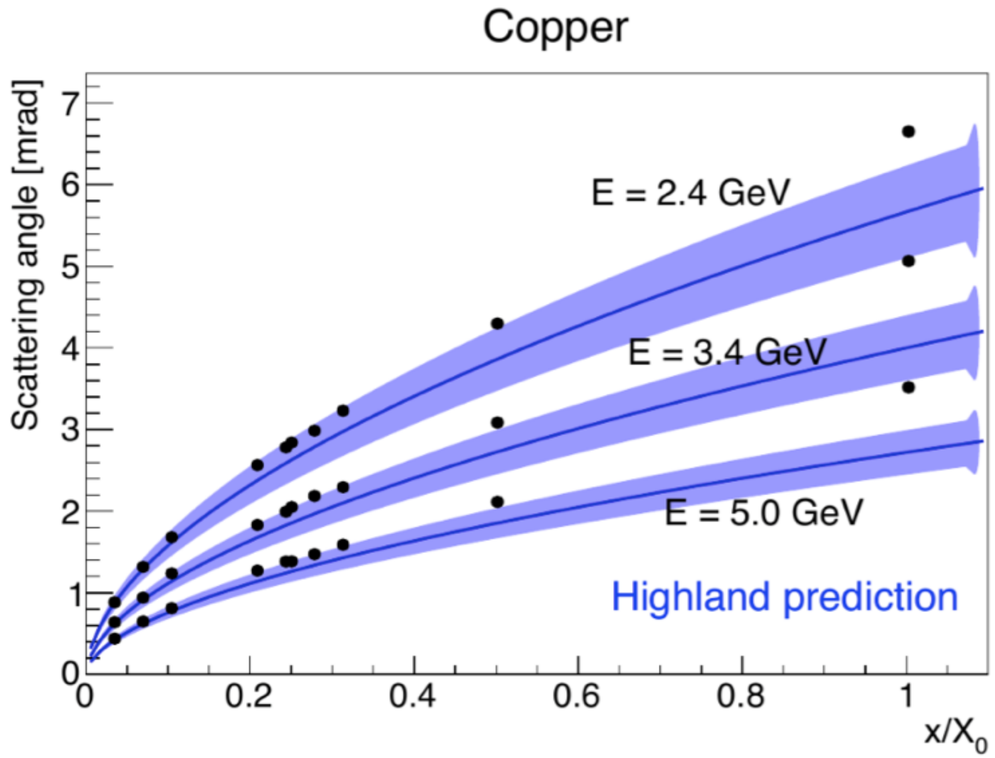


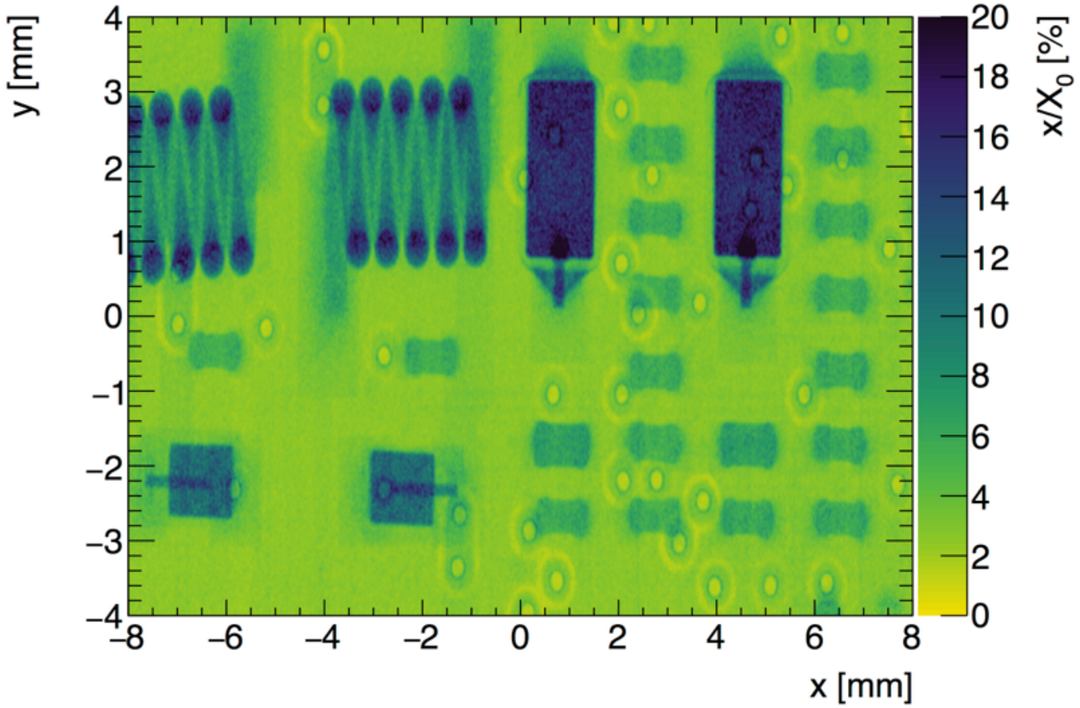
GBL



Scattering measurements

- Cover today examples of passive DUTs:
  - Measurement of scattering angles using a beam telescope.
  - Track finding and fitting performed using GBL in EUTelescope.
- For further details: see talks by Jan, Jan-Hendrik, and Hendrik.





Tutorial: hands-on

- Requires recent build of ILCSoft + EUTelescope.
  - http://eutelescope.web.cern.ch/content/installation

- Build available on the DESY AFS, setup using:

```
source /afs/desy.de/user/m/michaela/public/ILCSoft/v01-19-02/Eutelescope/master/build_env.sh
```

  - Can run from lxplus or NAF (might be slow!)
- Data can be found here:

```
/afs/desy.de/user/m/michaela/public/Data
```

Tutorial: examples

1) **gbl_local**: Example EUTelescope analysis for empty telescope using GBL processors
   https://github.com/eutelescope/eutelescope/tree/master/jobsub/examples/gbl_local

2) **gblkinkestimator**: Example EUTelescope analysis to produce scattering images
   `/afs/desy.de/user/m/michaela/public/ScatteringTutorial`

- More examples can be found here:
  - https://github.com/eutelescope/eutelescope/tree/master/jobsub/examples

Example 1: empty telescope

- mkdir my_analysis; cd my_analysis
- source /afs/desy.de/user/m/michaela/public/ILCSoft/v01-19-02/Eutelescope/master/build_env.sh
- cp -r $EUTELESCOPE/jobsub/examples/gbl_local/* .

- In config.cfg change:
  - BasePath: path/to/my_analysis
  - NativePath: /afs/desy.de/user/m/michaela/public/Data

Example 1: empty telescope

1) Setup the EUTelescope environment:

```
source $EUTELESCOPE/build_env.sh
```

2) Create the directory structure for the output files:

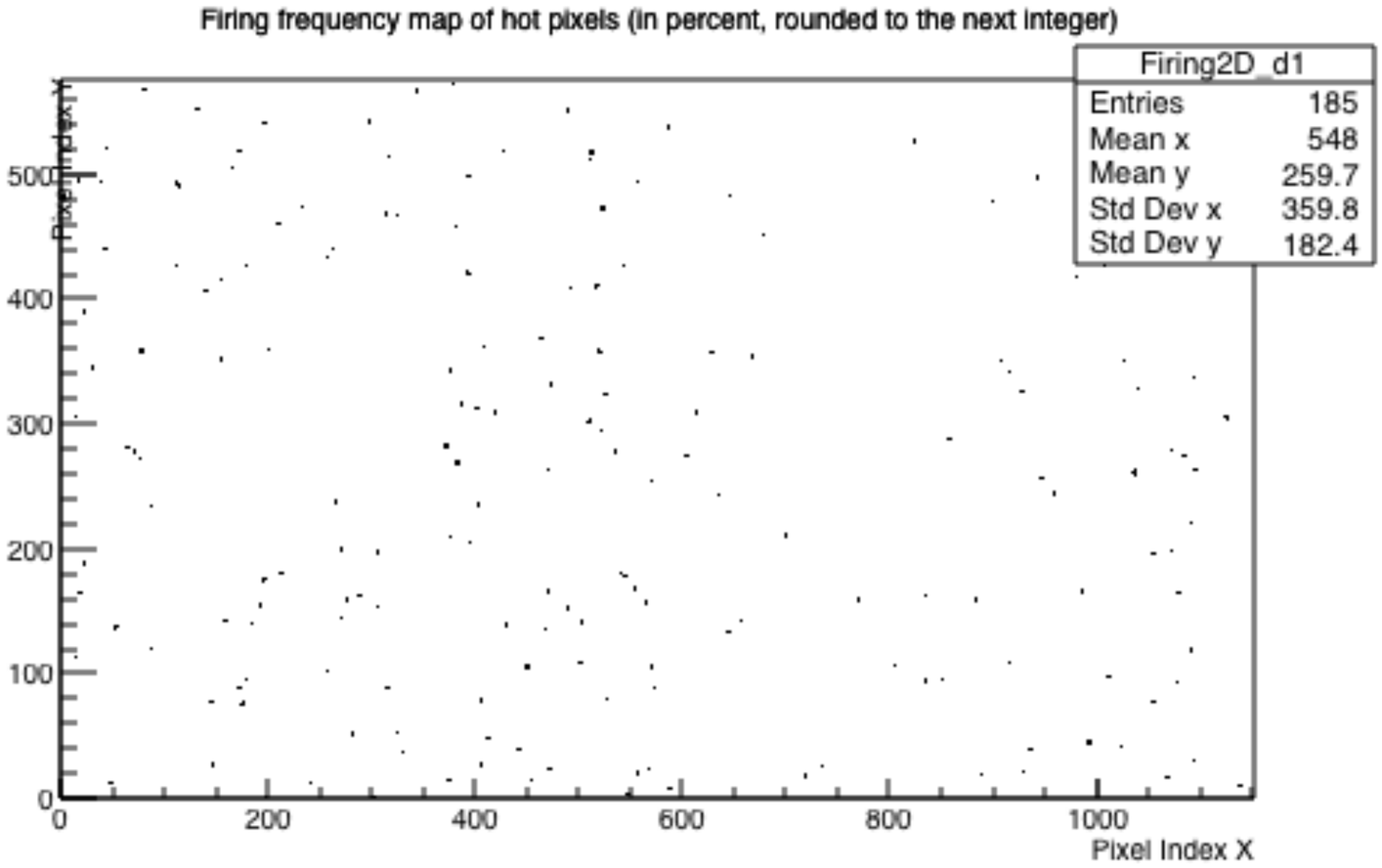
```
mkdir -p ./output/histograms && mkdir -p ./output/database && mkdir -p ./output/logs
&& mkdir -p ./output/lcio
```

3) Run each of the processors in order:

```
jobsub -c config.cfg -csv runlist.csv -g converter 117
jobsub -c config.cfg -csv runlist.csv -g clustering 117
jobsub -c config.cfg -csv runlist.csv -g hitmaker 117
jobsub -c config.cfg -csv runlist.csv -g aligngbl 117
jobsub -c config.cfg -csv runlist.csv -g aligngbl2 117
jobsub -c config.cfg -csv runlist.csv -g aligngbl3 117
jobsub -c config.cfg -csv runlist.csv -g trackgbltriplet 117
```

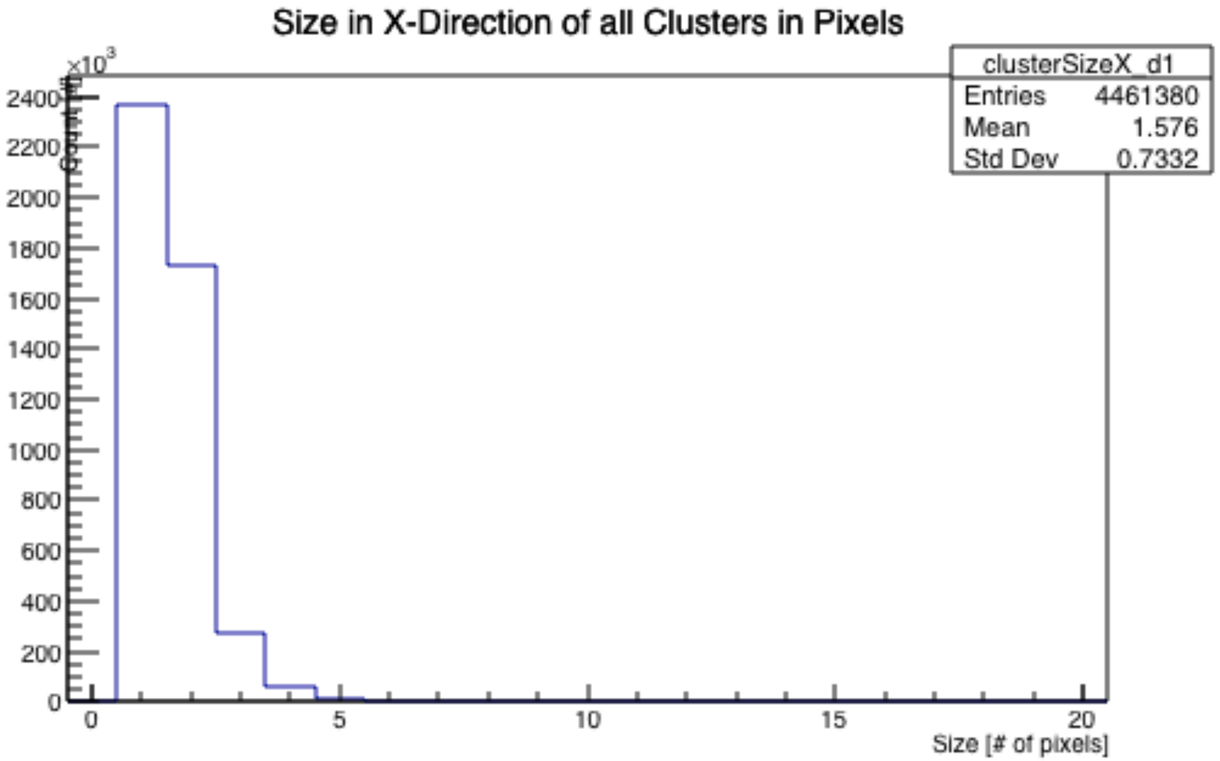
Step-by-step: converter

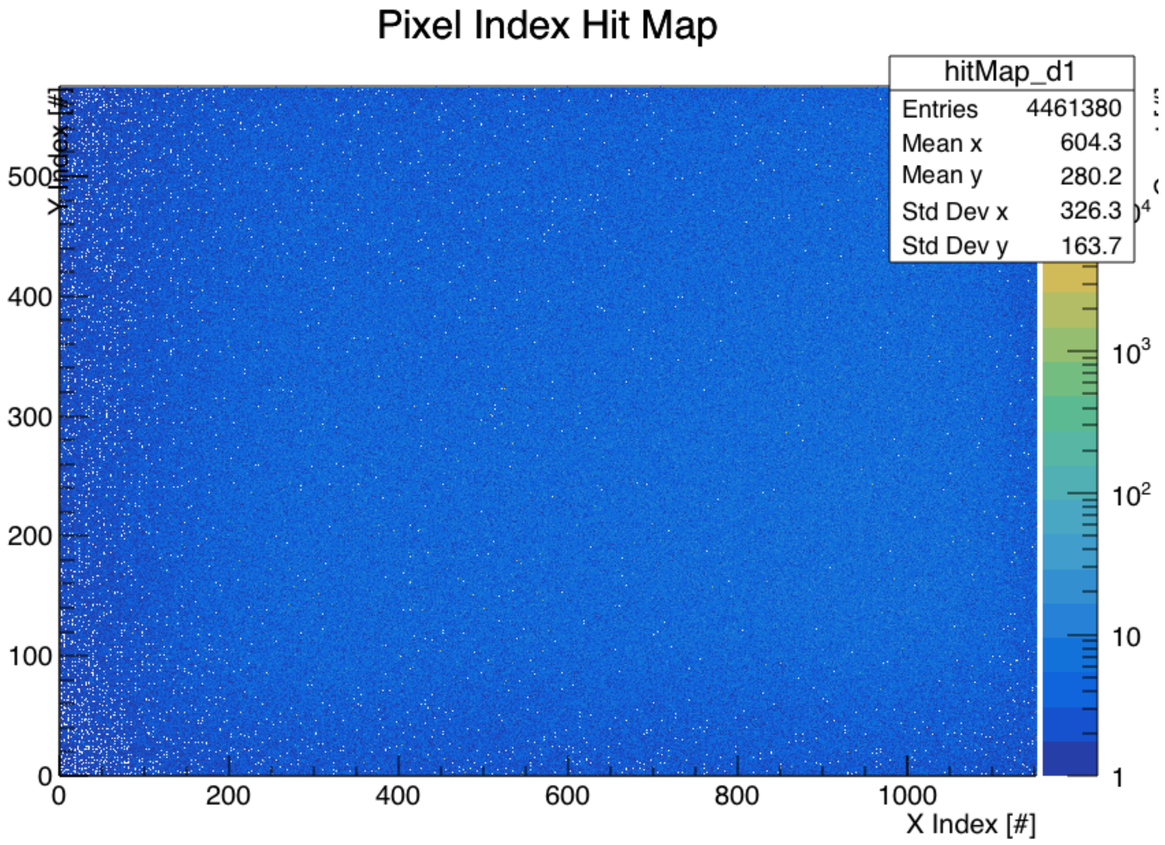
- **Converter**: Convert data from RAW to LCIO format, using EUDAQ's Native Reader. Noisy pixels are also identified.



Step-by-step: clustering

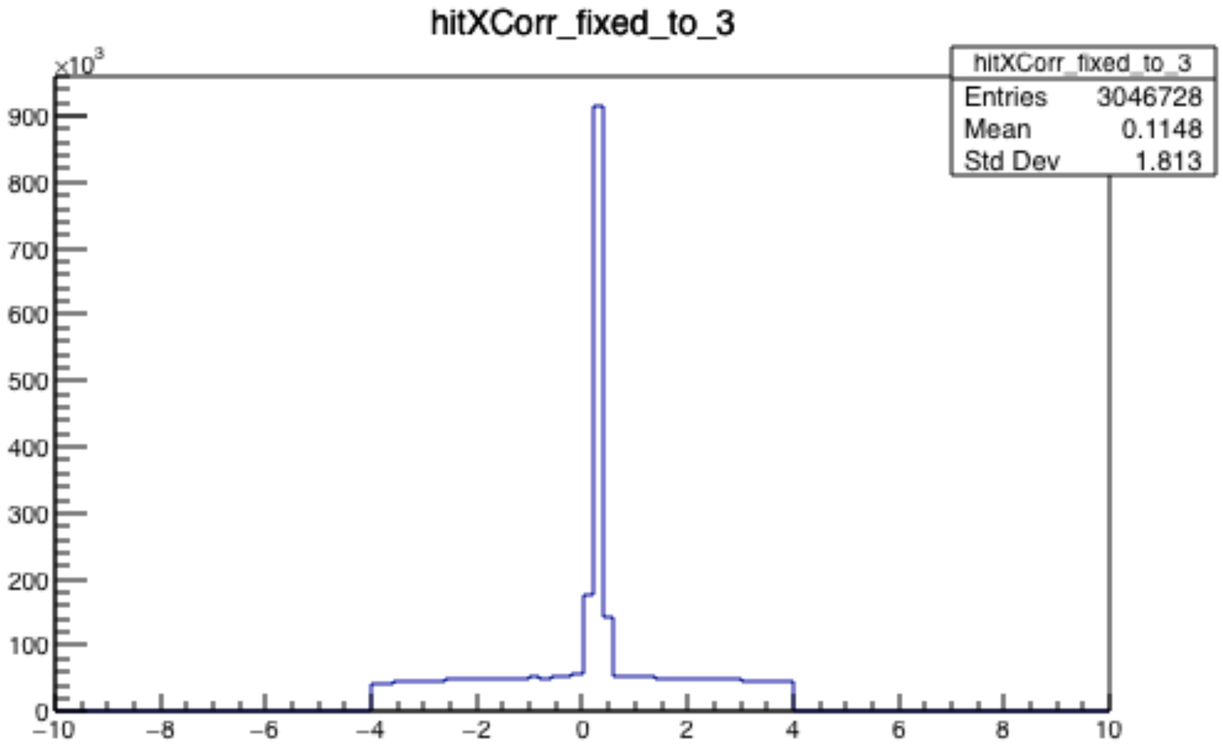
- **Converter**: Clusters in MIMOSA16 sensors are identified from adjacent hits. Noisy pixels are masked during this step.

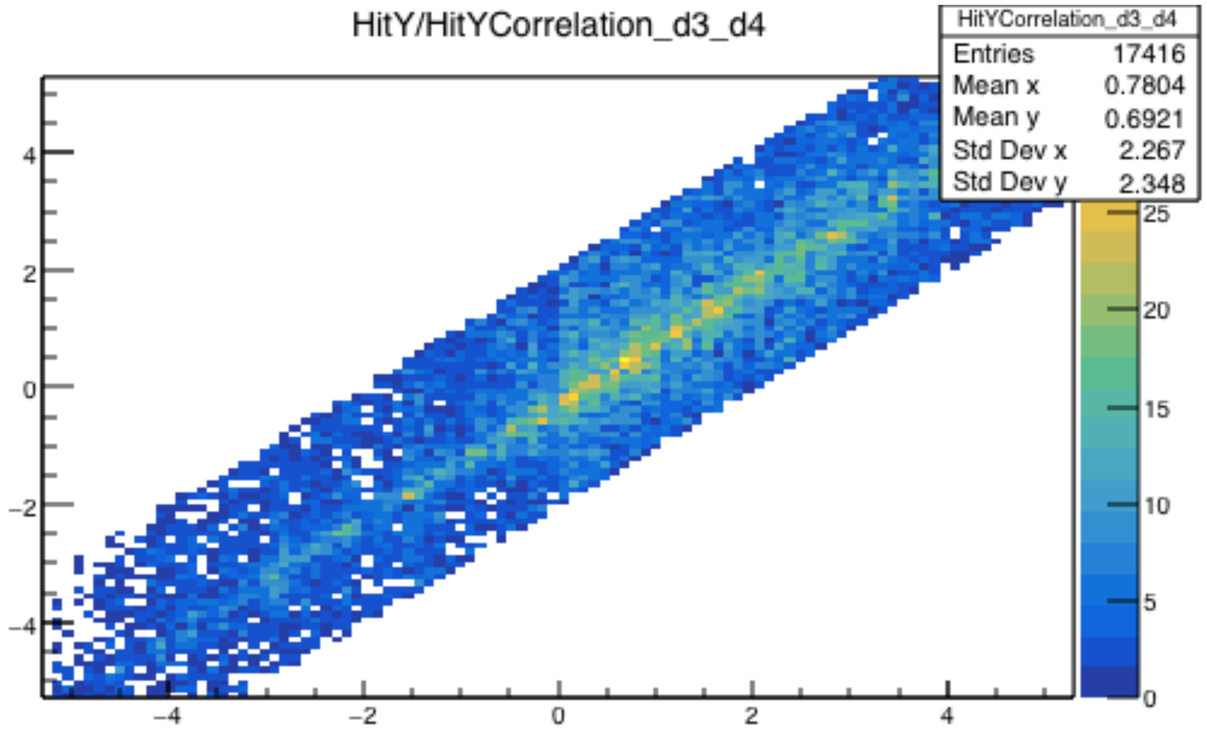




Step-by-step: hitmaker

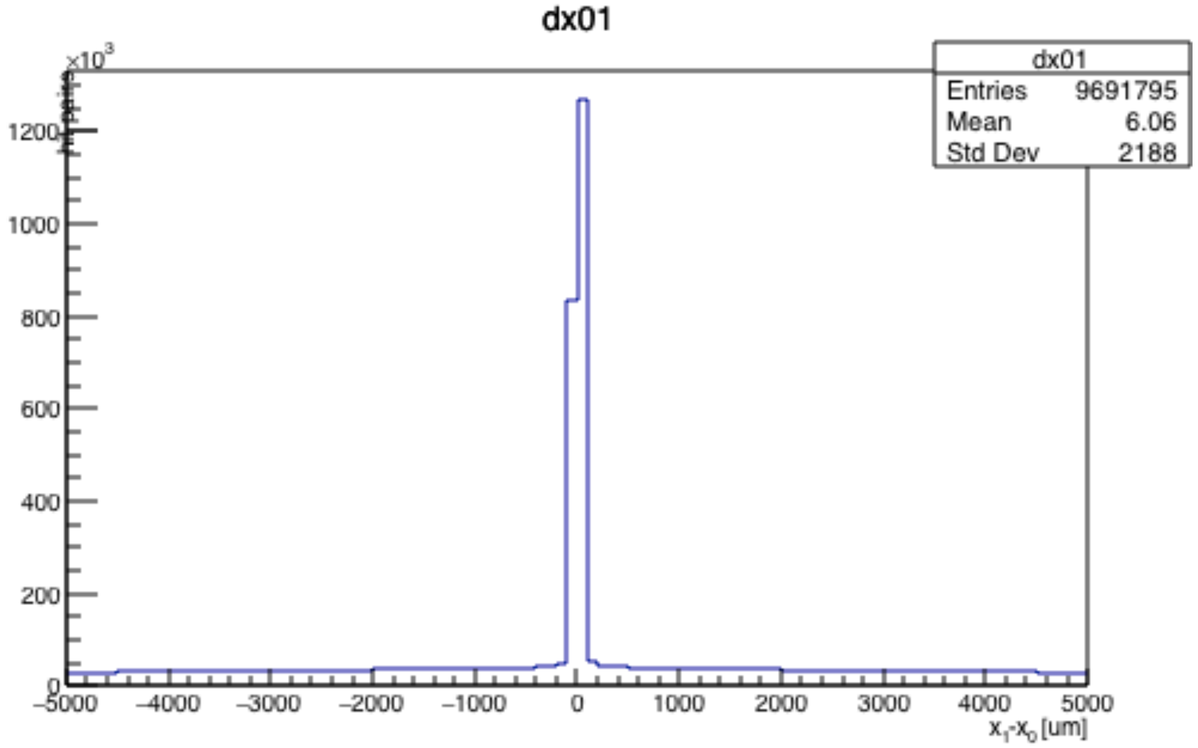
- **Hitmaker**: Creates collection of hits in the local coordinate system, evaluates correlations between telescope planes, performs pre-alignment.

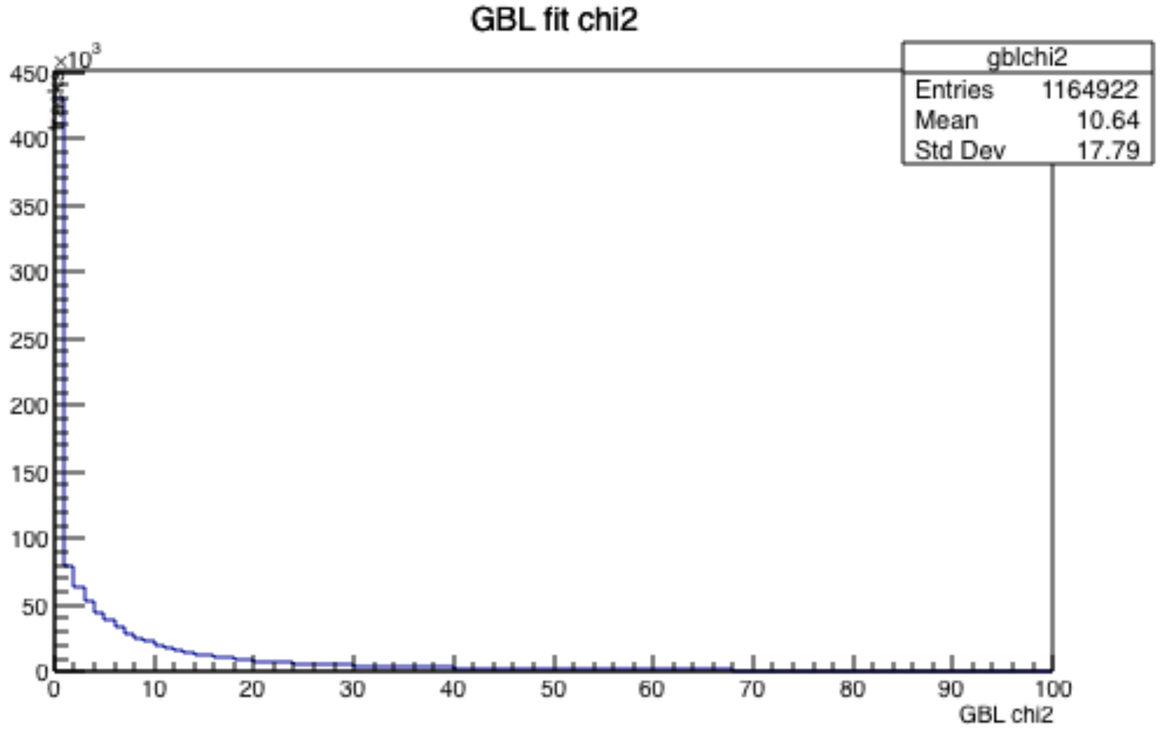




Step-by-step: alignment

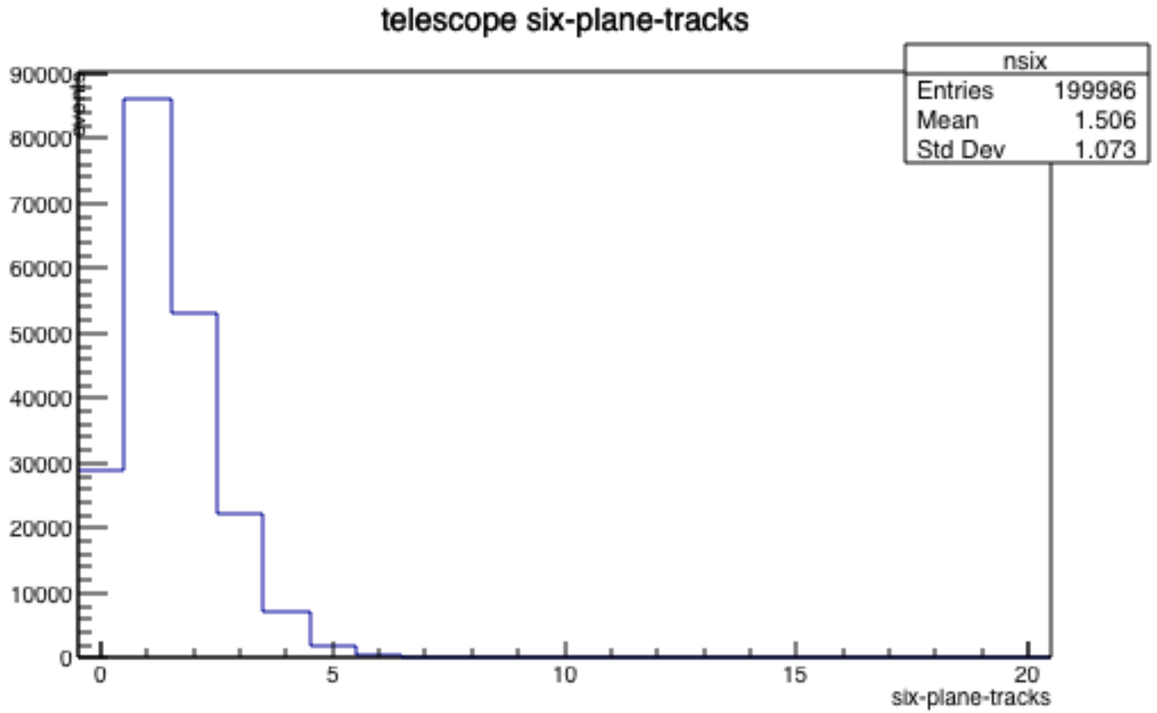
- **Alignment**: Performed using GBL and MillepedeII, iterative procedure with updated gear file after each step.

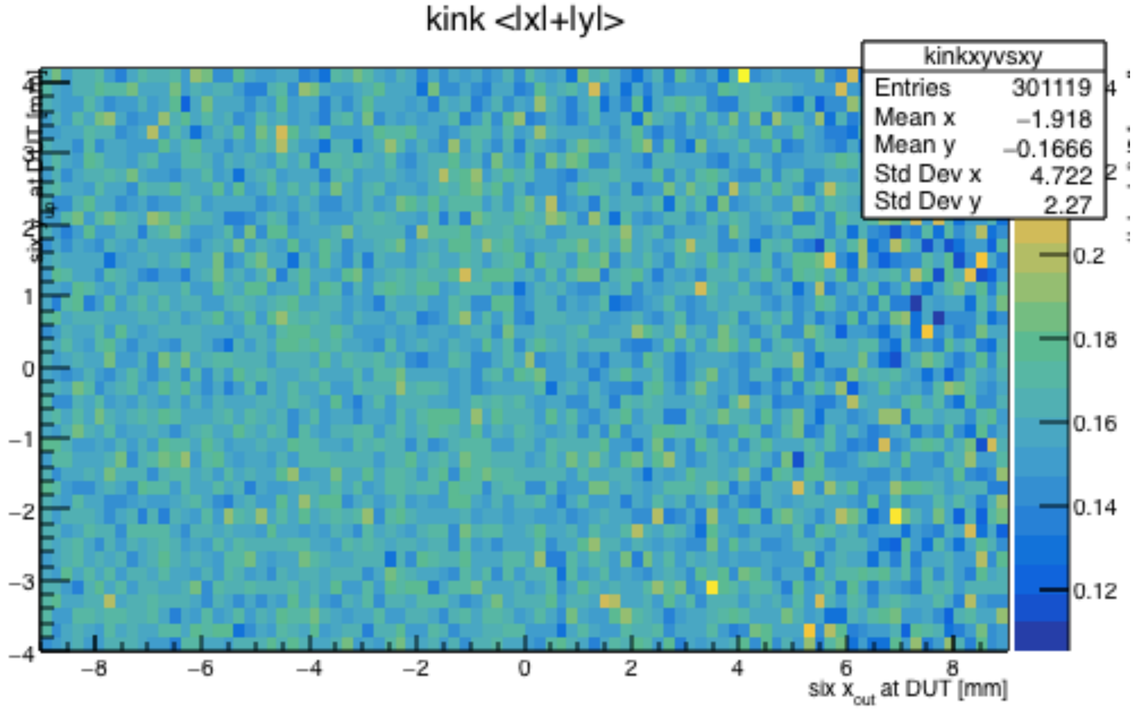




Step-by-step: trackgbltriplet

- **Track GBL triplet**: perform track fitting using GBL, evaluate track kink angle at centre of telescope.
  - Histograms stored in FitterXX/GBL and FitterXX/GBLUtility





Example 2: scattering images

1) Create a new analysis directory.
2) Setup the EUTelescope environment:

```
source $EUTELESCOPE/build_env.sh
```

3) Copy the analysis files:

```
cp -r /afs/desy.de/user/m/michaela/public/ScatteringTutorial/*
```

4) Change BasePath in config.cfg to your analysis directory.
5) Create the directory structure for the output files:

```
mkdir -p ./output/histograms && mkdir -p ./output/database && mkdir -p ./output/logs &&
mkdir -p ./output/lcio
```

6) Run each of the processors in order:

```
jobsub -c config.cfg -csv runlist.csv -g converter 110
jobsub -c config.cfg -csv runlist.csv -g clustering 110
jobsub -c config.cfg -csv runlist.csv -g hitmaker 110
jobsub -c config.cfg -csv runlist.csv -g aligngbl 110
jobsub -c config.cfg -csv runlist.csv -g aligngbl2 110
jobsub -c config.cfg -csv runlist.csv -g aligngbl3 110
jobsub -c config.cfg -csv runlist.csv -g gblkinkestimator 110
```

GEAR file with 6 telescope planes only

GEAR file with 6 telescope planes + DUT

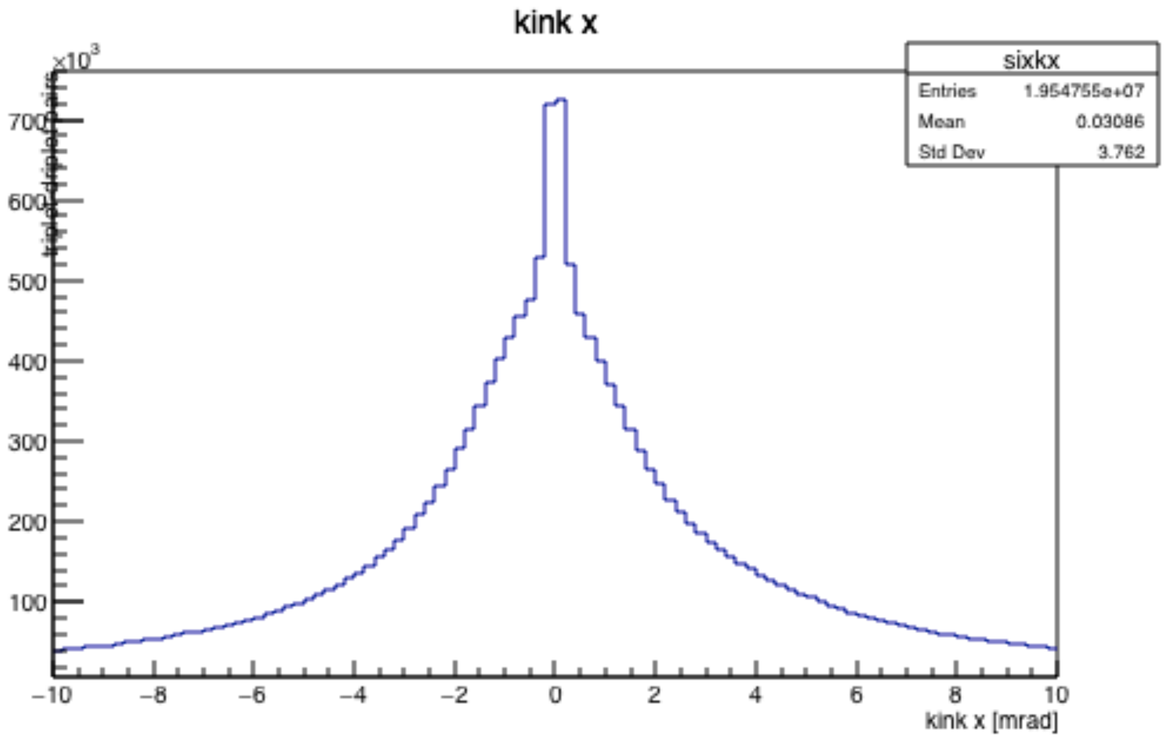
Step-by-step: post-alignment

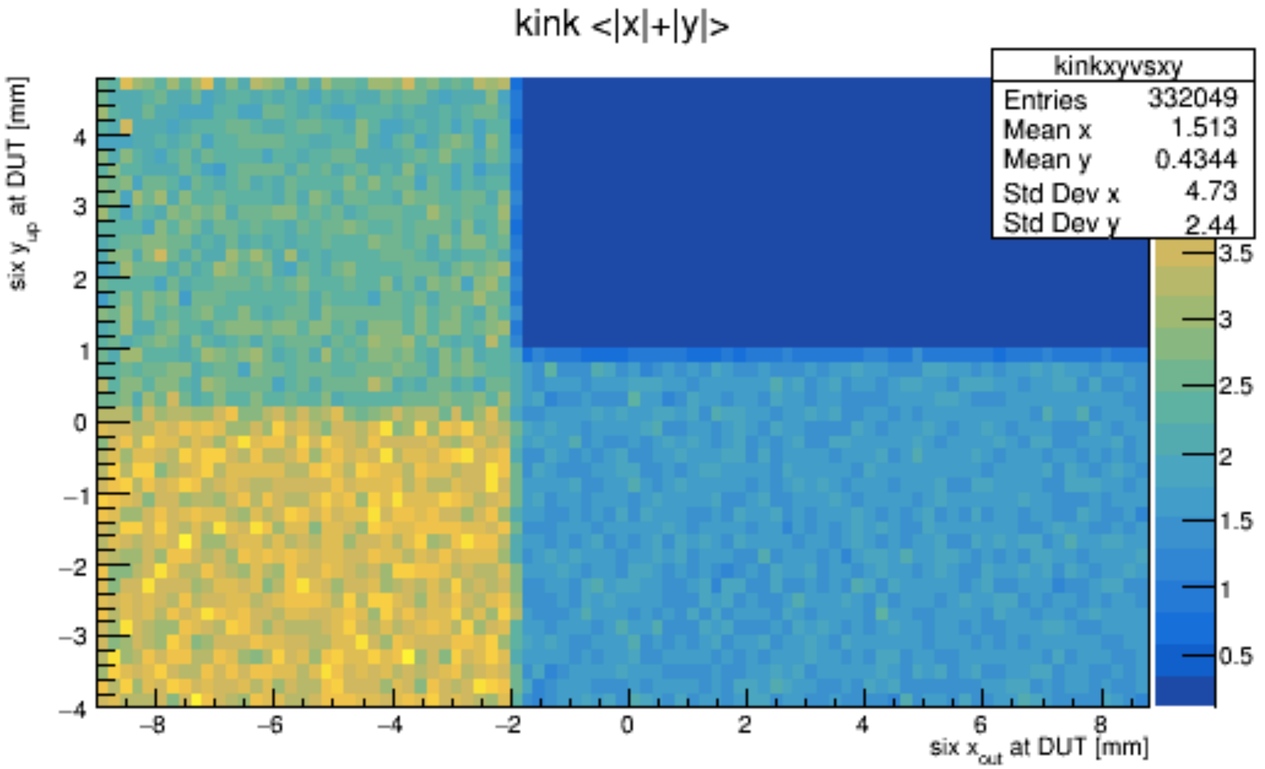
- After you've aligned the telescope sensors, make a copy of your aligned gear file (*_pre_gbl1_gbl2_gbl3.xml) and rename to *_aligned.
- Then insert a plane for the scatterer between telescope planes: **1 2 3 DUT 4 5 6**.

```
<!--DUT Plane-->
<layer>
  <ladder       ID="100"
                positionX="0.00"        positionY="0.00"        positionZ="72.5"
                rotationZY="0.00"       rotationZX="0.0"        rotationXY="0.0"
                sizeX="50."             sizeY="50."             thickness="0.01"
                radLength="89."
                />
  <sensitive    ID="100"
                positionX="0.00"        positionY="0.00"        positionZ="72.5"
                sizeX="50."             sizeY="50."             thickness="0.01"
                npixelX="1  "           npixelY="1"
                pitchX="50"             pitchY="50"             resolution="1e10"
                rotation1="-1.0"        rotation2="0.0"
                rotation3="0.0"         rotation4="-1.0"
                radLength="89."
                />
</layer>
```

Step-by-step: gblkinkestimator

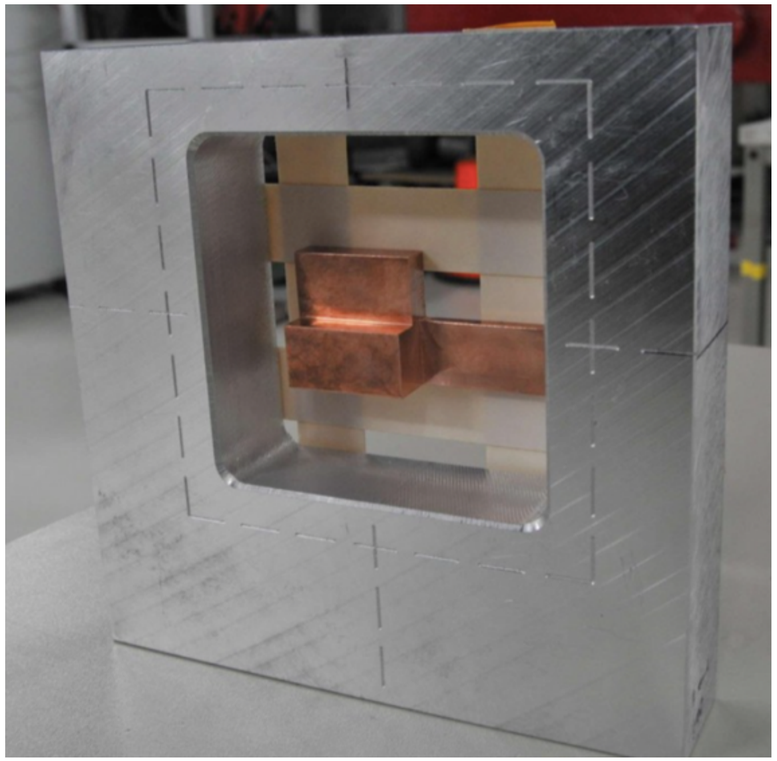
- **GBL kink estimator**: perform track fitting using GBL, with unbiased estimate of kink angle of track at DUT.
  - Histograms stored in Fitter06/GBLUtility

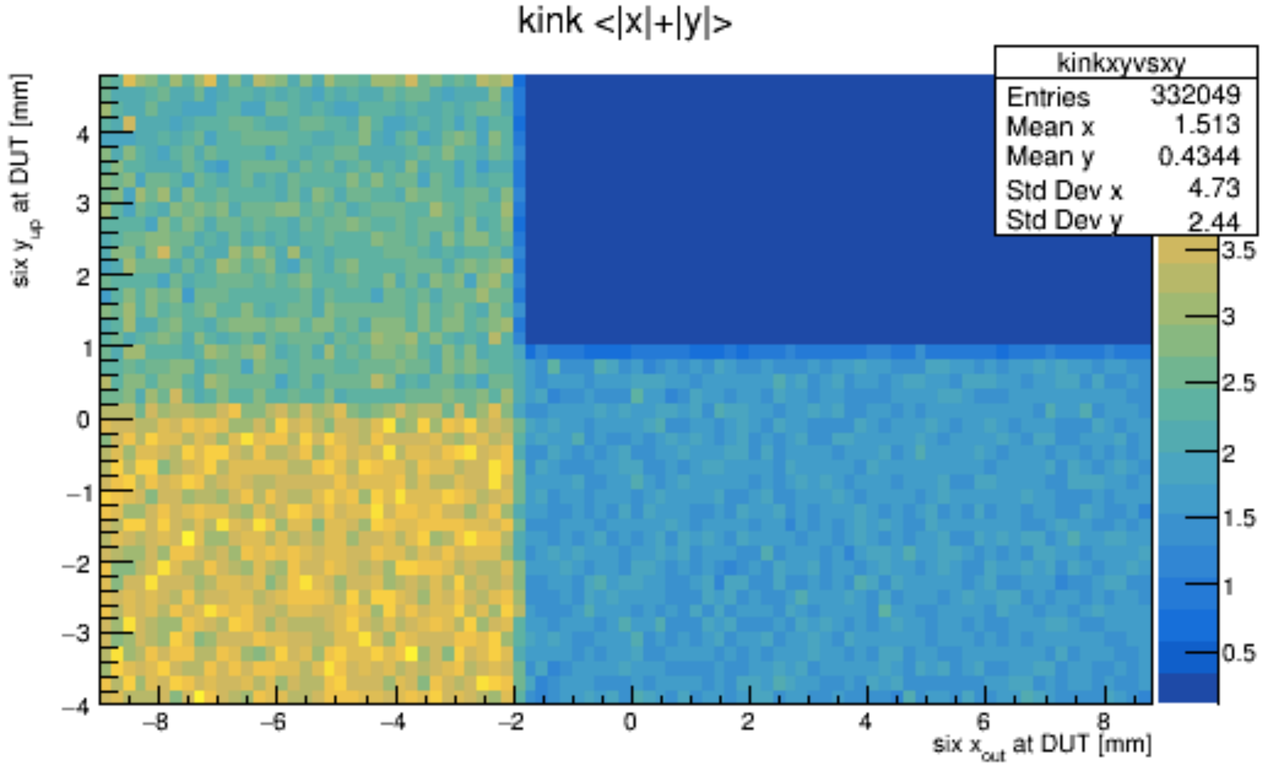




Step-by-step: gblkinkestimator

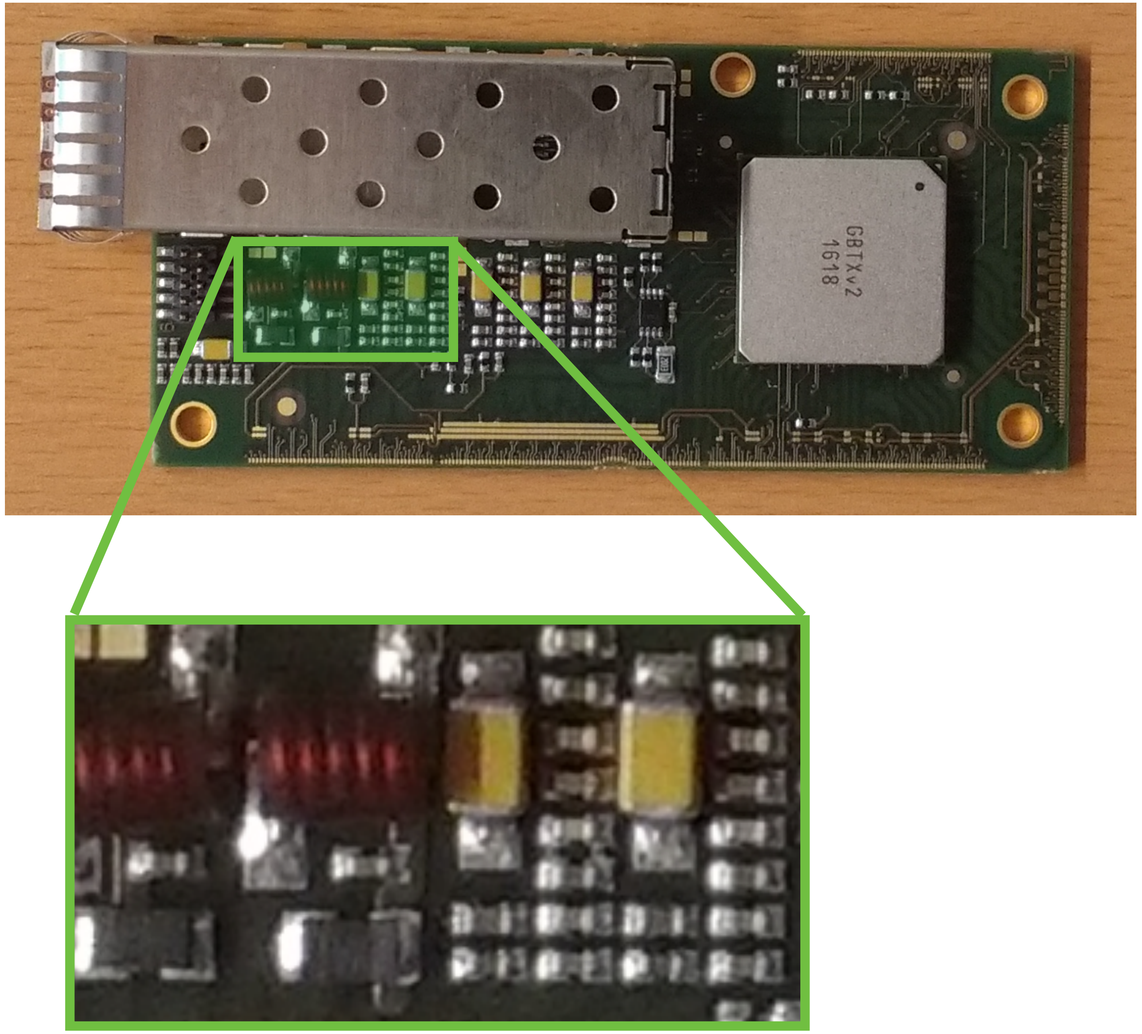
- **GBL kink estimator**: perform track fitting using GBL, with unbiased estimate of kink angle of track at DUT.
  - Histograms stored in Fitter06/GBLUtility

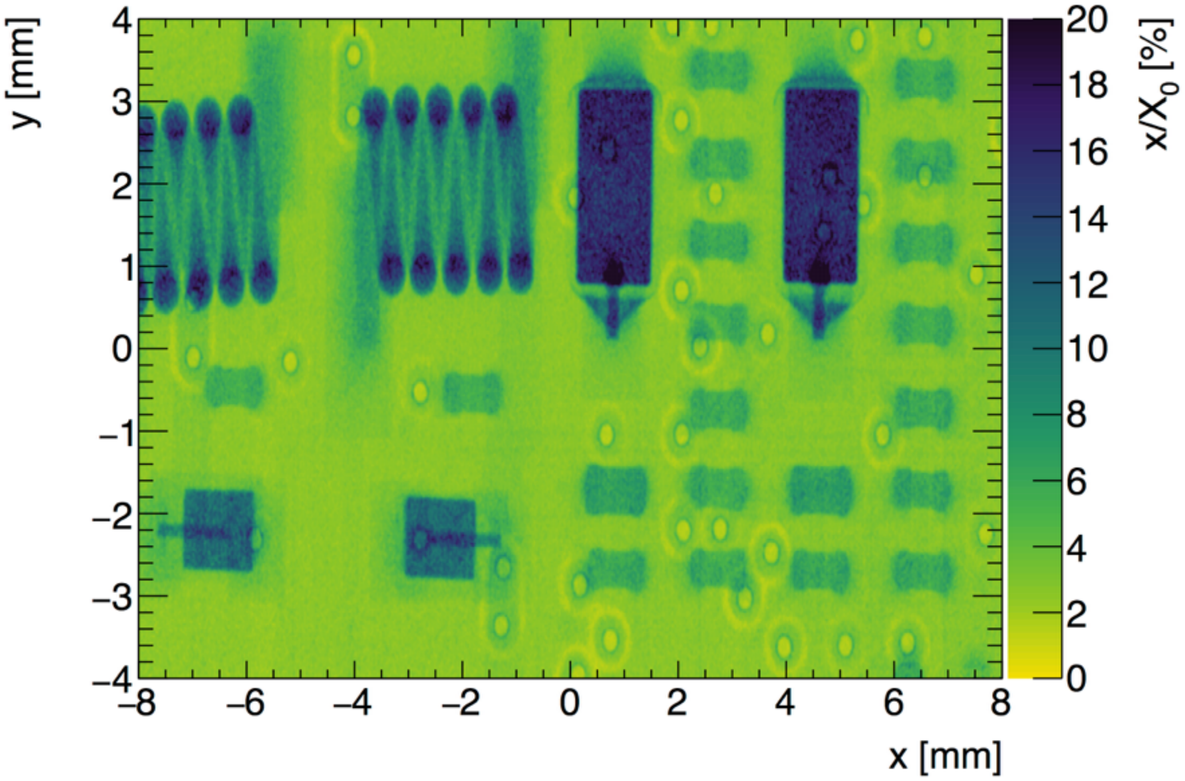




More example results

- With more time and much more data…





More example results

- With more time and much more data…