Corryvreckan

THE MAELSTROM FOR YOUR TEST BEAM DATA

BTTB 7, Tuesday 15th January 2019

Morag Williams, University of Glasgow and CERN
On behalf of the CLICdp collaboration
morag.williams@cern.ch

AIMS OF CORRYVRECKAN?

- **Reconstruct and analyse data from pixel assemblies**
- Modular structure - similar to Allpix-squared
- Highly flexible and configurable
- Easy to understand - written in modern C++, well documented

A little Corryvreckan demo:

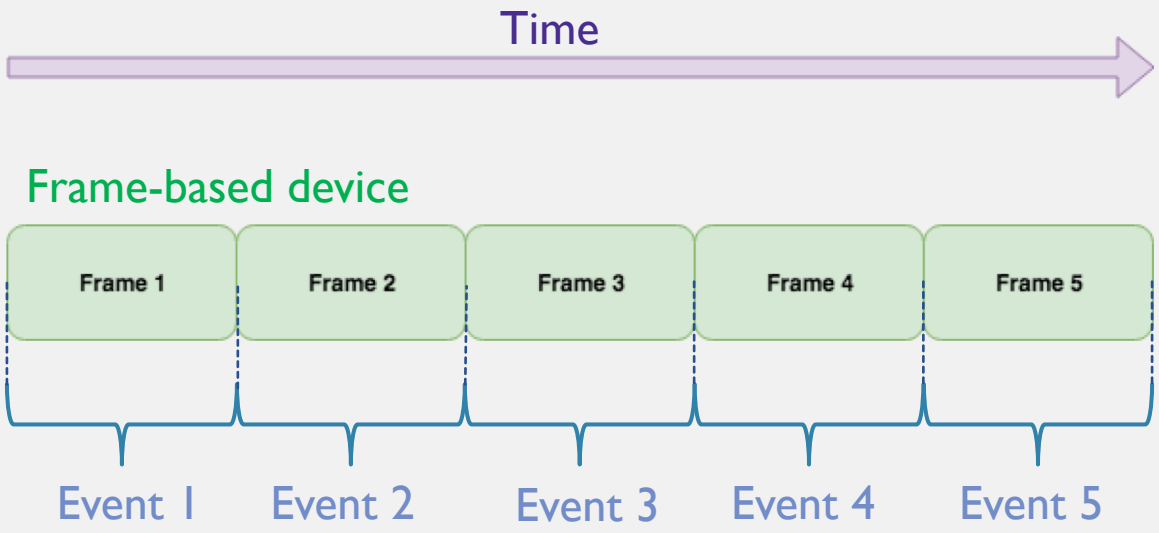
```
2019-01-07 14:40 williamm@pclcd23:corryvreckan-tests$
```

COOL FEATURES!

- 4D-tracking – Use both spatial and timing cuts to associate clusters to your track, improving track quality
- Millepede alignment – option to use millipede to align your telescope planes to high precision
- Online data monitoring – check data quality while reconstructing, very useful in test-beam environments
- Frame-based and data-driven readouts…

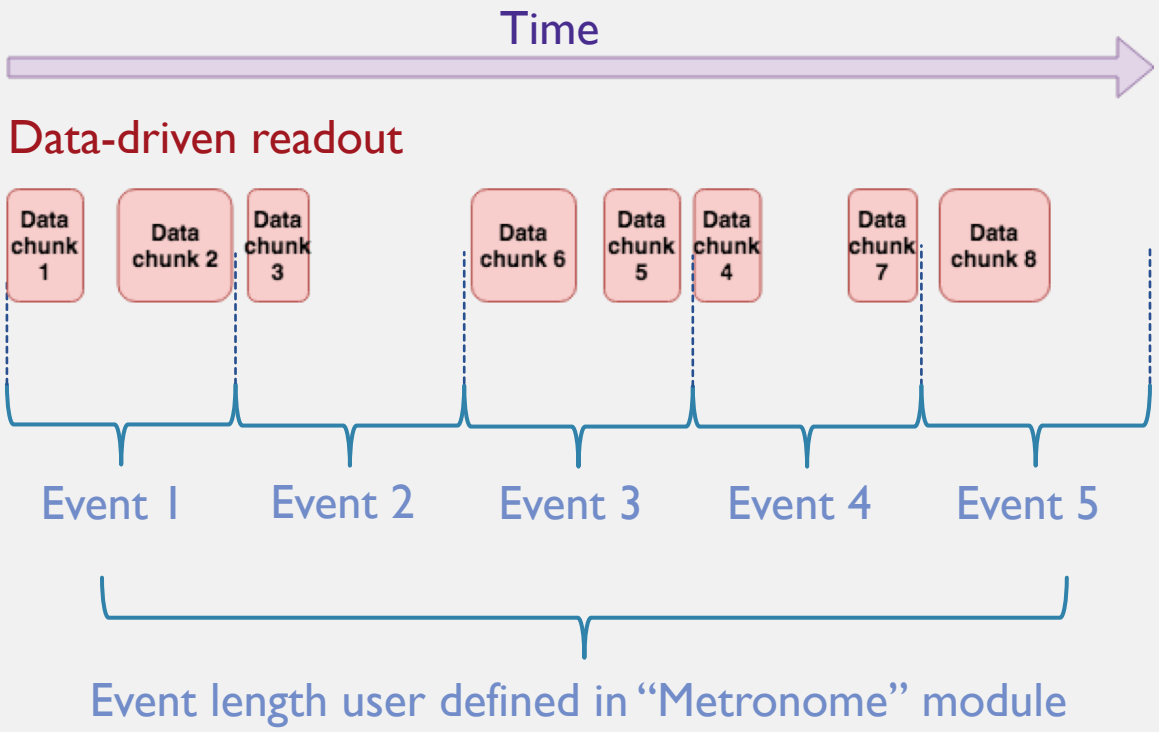
DATA-DRIVEN AND FRAME-BASED READOUT DEVICES

- Can easily use combinations of devices with frame-based and/or data-driven readouts in the same reconstruction
- Need arbitrary chunks of data to process ('events')
- Frame-based: Corryvreckan uses the start and end points of the frame to define an event



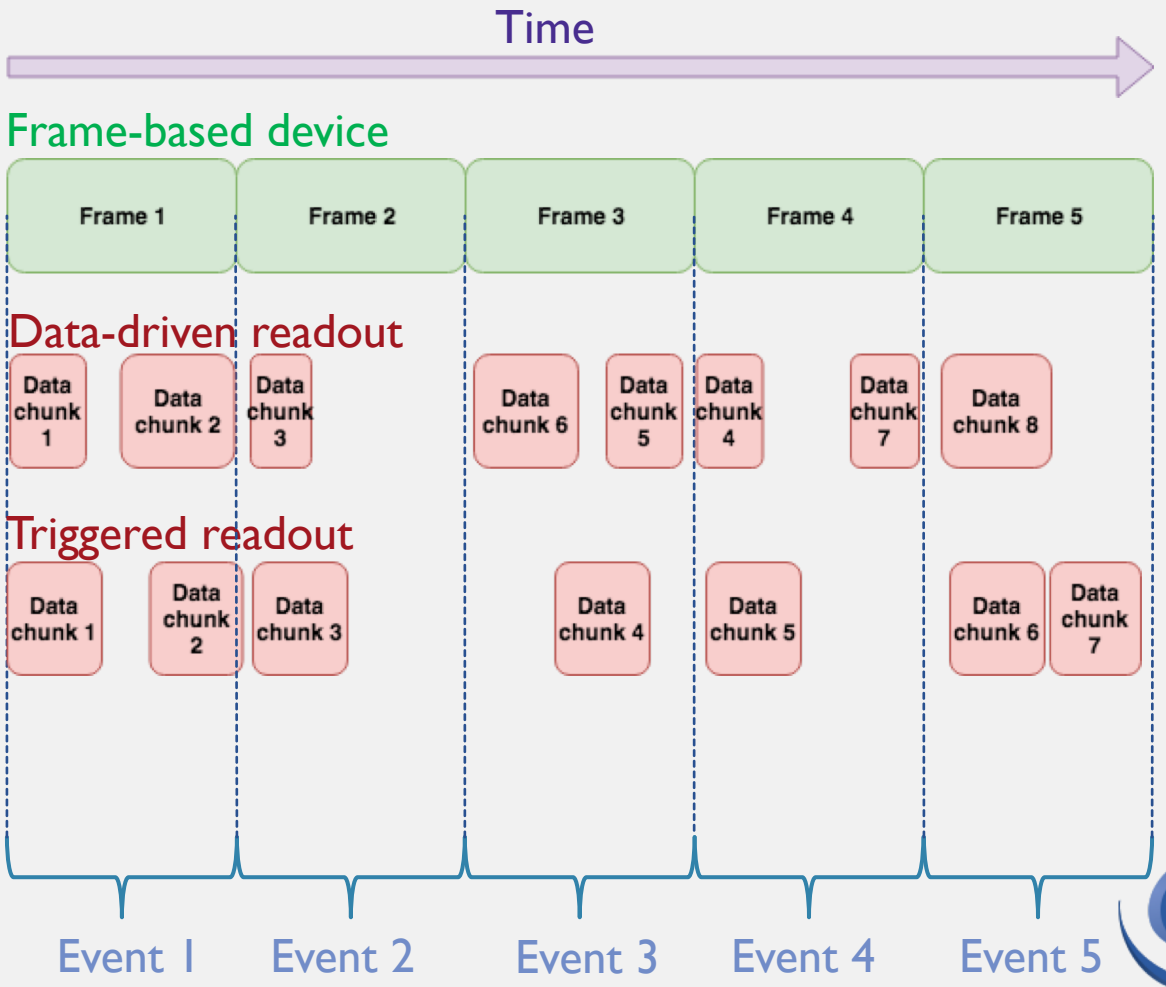
DATA-DRIVEN AND FRAME-BASED READOUT DEVICES

- Can easily use combinations of devices with frame-based and/or data-driven readouts in the same reconstruction
- Need arbitrary chunks of data to process ('events')
- Frame-based: Corryvreckan uses the start and end points of the frame to define an event
- Data-driven: either use the "Metronome" module to define the length of an event…



DATA-DRIVEN AND FRAME-BASED READOUT DEVICES

- Can easily use combinations of devices with frame-based and/or data-driven readouts in the same reconstruction
- Need arbitrary chunks of data to process ('events')
- Frame-based: Corryvreckan uses the start and end points of the frame to define an event
- Data-driven: either use the "Metronome" module to define the length of an event…or use the definition of an event from a frame-based readout device

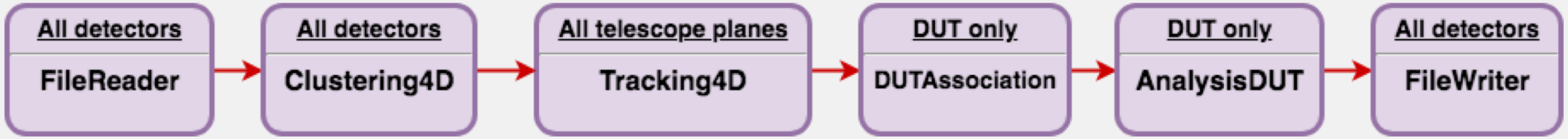


COOL FEATURES!

- 4D-tracking – Use both spatial and timing cuts to associate clusters to your track, improving track quality
- Millepede alignment – option to use millipede to align your telescope planes to high precision
- Online data monitoring – check data quality while reconstructing, very useful in test-beam environments
- Frame-based and data-driven readouts – can easily use combinations of triggered and/or trigger-less devices in the same reconstruction using the "Metronome" module
- Modular approach…

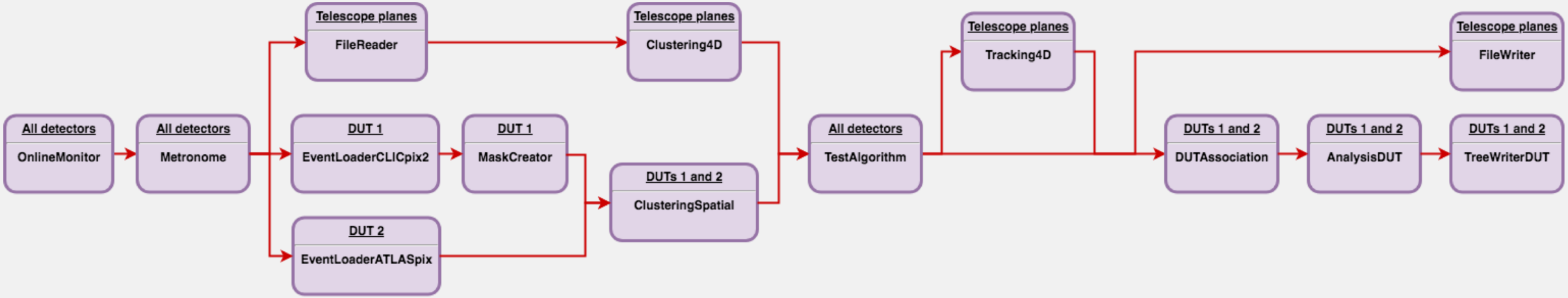
MODULAR APPROACH

- Modules = plug-and-play algorithms for specific tasks, using objects (pixels, clusters, or tracks).
- Select suitable clustering, tracking, ... modules
- Modular approach allows quick set-up, and quick configuration



MODULAR APPROACH

- Can create more involved reconstruction chains
- Apply different modules to different devices in the same reconstruction



HOW TO USE CORRY:

1. SET-UP

- Configuration file: specifies modules, sets module parameters, and paths to input and output
- Geometry file: describe geometry of set-up
- TOML style = easy to read
- Physical unit support

telescope_cp2_commandline.conf

```
[Corryvreckan]
log_level = "WARNING"
log_format = "DEFAULT"
detectors_file=../geometries/Alignment_November2018_3.geo
histogram_file=histograms_29786.root
number_of_tracks=20000

[EventLoaderCLICpix2]
discard_zero_tot = true
input_directory=/eos/experiment/clicdp/grid/ilc/user/c/clictel/CERN_TB_November_2018/clicpix2/Run29786

[EventLoaderTimepix3]
input_directory=/eos/experiment/clicdp/grid/ilc/user/c/clictel/CERN_TB_November_2018/data/Run29786

[Clustering4D]
type = "Timepix3"

[ClusteringSpatial]
type = "CLICpix2"

[TestAlgorithm]
make_correlations=true
timing_cut=2ms

[Tracking4D]
min_hits_on_track=5

[DUTAssociation]
spatial_cut = 200um 200um
timing_cut = 2ms
```

Alignment_November2018_3.geo

```
[014_CP_PS]
number_of_pixels = 128,128
orientation = -0.000445,0.000443,-0.351941
orientation_mode = "xyz"
pixel_pitch = 0.025000,0.025000
position = 1.332275,0.180374,106.000000
type = "CLICpix2"
role = "dut"

[W0013_G03]
number_of_pixels = 256,256
orientation = 8.99876deg,8.99412deg,-0.0178763deg
orientation_mode = "xyz"
pixel_pitch = 55um,55um
position = -10.202um,-11.943um,284.5mm
resolution = 4um,4um
role = "reference"
type = "Timepix3"
```

HOW TO USE CORRY: 2. THE CORRY EXECUTABLE

```
corry  -c config.cfg
```

Path to configuration file

HOW TO USE CORRY: 2. THE CORRY EXECUTABLE

Sets the global log verbosity level

module configurations - specified by adding a dot (.) between the module and the key

```
corry -c conf.cfg -l log.txt -v "INFO" -o histogram_file="hists" -o FileWriter.onlyDUT="true" -g mydut.orientation=0deg,5deg,0deg
```

Specify an additional location such as a file to forward log output to.

framework parameters

detector parameters

HOW TO USE CORRY:

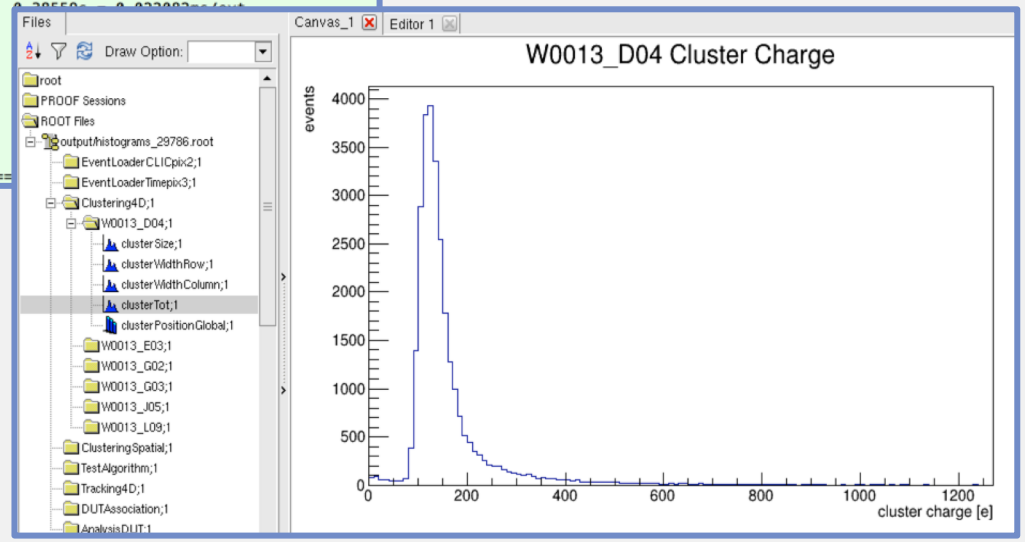
3. OUTPUT

- Clear terminal output
- Descriptive logging/error output
- Available outputs:

  ROOT files of Corryvreckan objects (pixels, clusters, tracks)

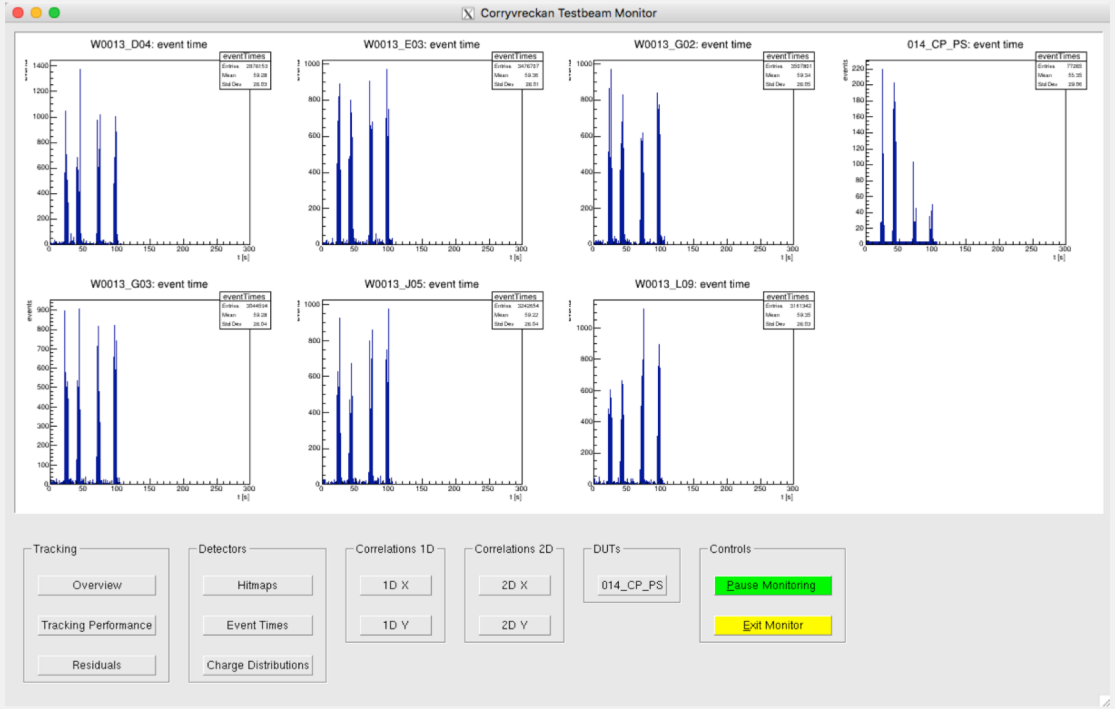
  ROOT file of analysis plots from each module used
- Online monitoring functionality…

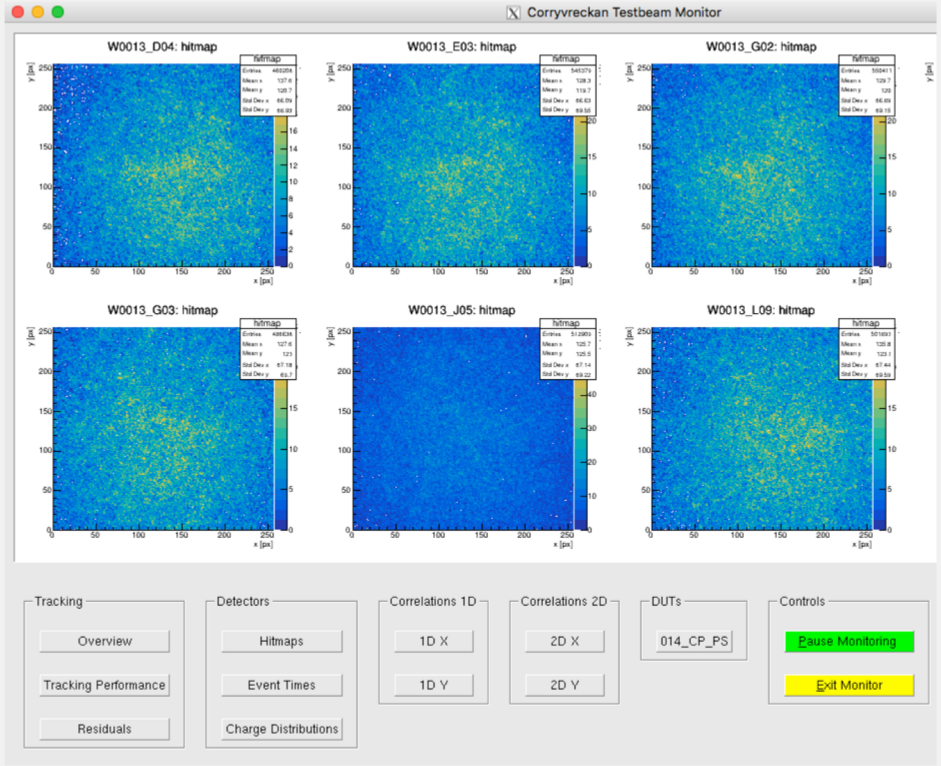
```
2019-01-08 11:21 williamm@pclcd23:corryvreckan-tests$ corry -c configs/telescope_cp2_commandline.conf
|11:21:48.957|  (STATUS) Welcome to Corryvreckan v0.9.3
|11:21:48.958|  (STATUS) Loaded 7 detectors
|11:21:49.523|  (STATUS) Loaded 24 module instances
|11:21:49.523|  (STATUS) =================| Initialising modules |=================
|11:21:50.038|  (STATUS) [I:AnalysisDUT:014_CP_PS] Initialising "AnalysisDUT:014_CP_PS"

|11:21:50.113|  (STATUS) =======================| Event loop |=======================
|11:21:59.785|  (STATUS) Ev: 17.4k Tr: 17.9k (1.03/ev) t = 22.751s
|11:22:00.155|  (STATUS) ===================| Finalising modules |===================
|11:22:02.910|  (STATUS) Wrote histogram output file to /afs/cern.ch/work/w/williamm/corryvreckan-tests
/configs/histograms_29786.root
|11:22:02.910|  (STATUS) ================| Wall-clock timing (seconds) |================
|11:22:02.910|  (STATUS)  EventLoaderCLICpix2 :  014_CP_PS  --  5.27746s = 0.302226ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_D04  --  0.17145s = 0.009819ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_E03  --  0.11291s = 0.006466ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_G02  --  0.12782s = 0.007320ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_G03  --  0.12887s = 0.007380ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_J05  --  0.11337s = 0.006493ms/evt
|11:22:02.910|  (STATUS)  EventLoaderTimepix3 :  W0013_L09  --  0.10317s = 0.005908ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_D04  --  0.10084s = 0.005775ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_E03  --  0.13371s = 0.007657ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_G02  --  0.12005s = 0.006875ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_G03  --  0.08286s = 0.004745ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_J05  --  0.10069s = 0.005766ms/evt
|11:22:02.910|  (STATUS)         Clustering4D :  W0013_L09  --  0.07790s = 0.004461ms/evt
|11:22:02.910|  (STATUS)    ClusteringSpatial :  014_CP_PS  --  0.05172s = 0.002962ms/evt
|11:22:02.910|  (STATUS)        TestAlgorithm :  W0013_D04  --  0.37547s = 0.021502ms/evt
|11:22:02.910|  (STATUS)        TestAlgorithm :  W0013_E03  --
|11:22:02.910|  (STATUS)        TestAlgorithm :  W0013_G02  --
|11:22:02.910|  (STATUS)        TestAlgorithm :  014_CP_PS  --
|11:22:02.910|  (STATUS)        TestAlgorithm :  W0013_G03  --
|11:22:02.910|  (STATUS)        TestAlgorithm :  W0013_J05  --
|11:22:02.911|  (STATUS)        TestAlgorithm :  W0013_L09  --
|11:22:02.911|  (STATUS)           Tracking4D               --
|11:22:02.911|  (STATUS)       DUTAssociation :  014_CP_PS  --
|11:22:02.911|  (STATUS)          AnalysisDUT :  014_CP_PS  --
|11:22:02.911|  (STATUS) ======================================
```



ONLINE MONITORING

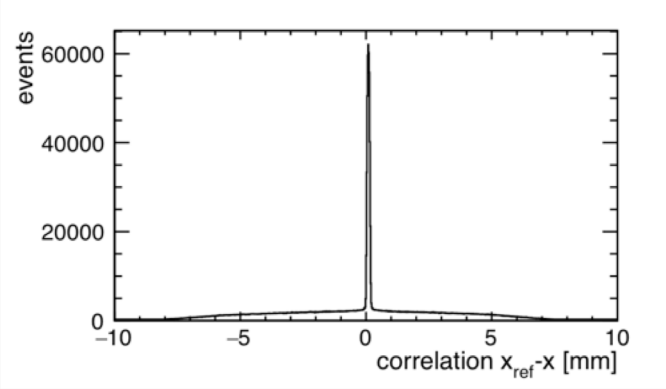
- check data quality while reconstructing
- very useful in test-beam environments

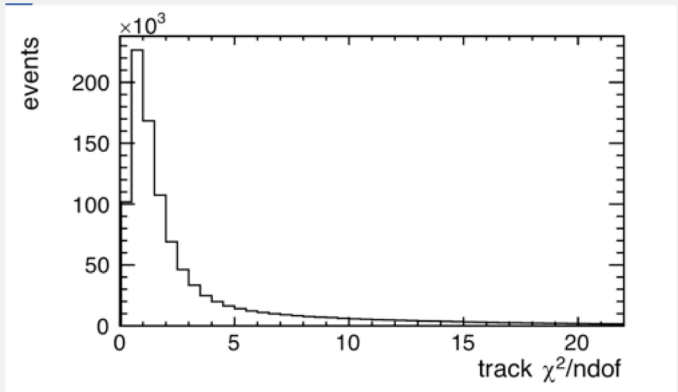


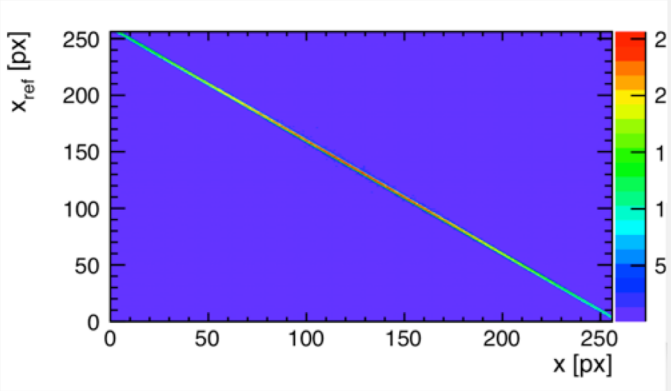


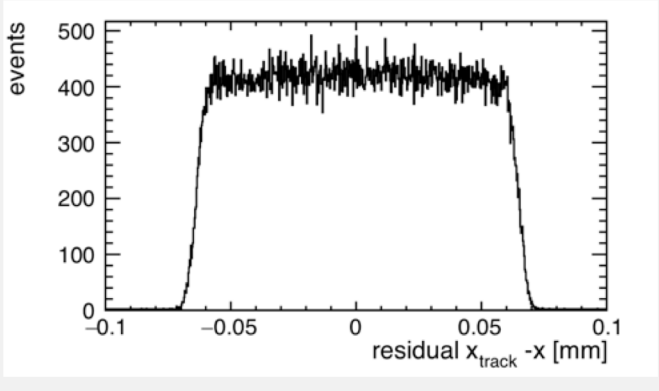
EXAMPLE RESULTS

- DUT analysis plots from ATLASpix device:
  - 1D correlation in X
  - Track chi^2
  - 2D correlation in X
  - DUT residual in X
  - DUT residual in Y
- List of currently implemented devices:
  - ATLASpix, CLICpix, CLICpix2, EUDAQ, Timepix1, Timepix3

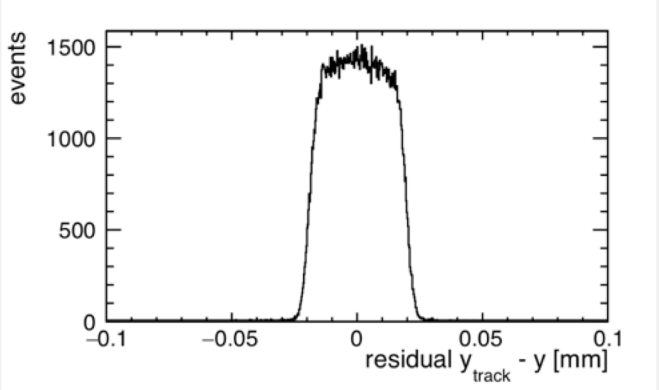




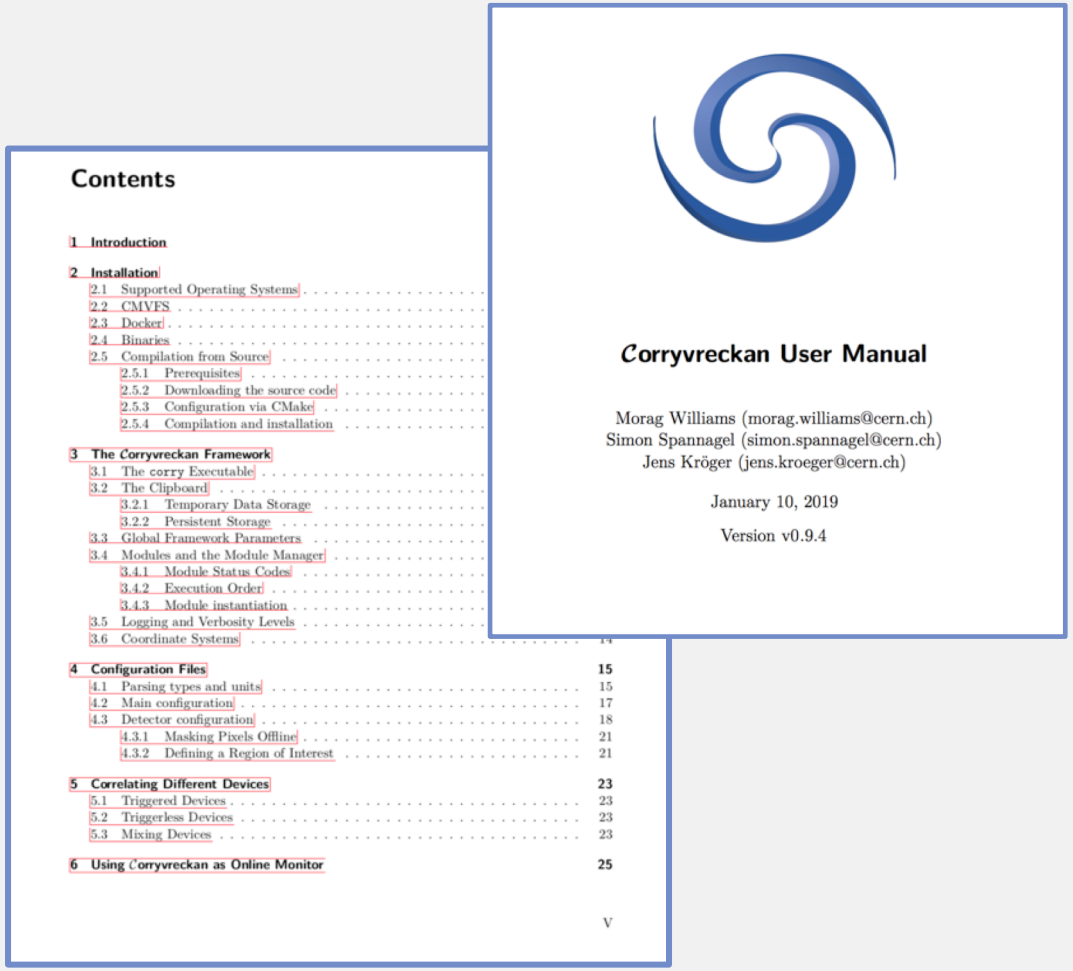




Note: more CLICdp ATLASpix results in Jen's talk on Thursday (https://indico.cern.ch/event/731649/contributions/3237243/).



CORRYVRECKAN MANUAL



Contents

V

Corryvreckan User Manual

Morag Williams (morag.williams@cern.ch)
Simon Spannagel (simon.spannagel@cern.ch)
Jens Kröger (jens.kroeger@cern.ch)

January 10, 2019

Version v0.9.4

- User manual:
  - Updated with new Corryvreckan changes
  - Autogenerated during building
  - Installation instructions, FAQ, ‘getting started’
  - Full descriptions of module functionality
  - Usage examples
- Note module descriptions can also be found as readme text files in the module’s src directory
- Can be downloaded here.

RESOURCES

- Repository:

  https://gitlab.cern.ch/corryvreckan/corryvreckan

  Contains the source code, issue tracker, user manual.

- *User manual:*

  Can be downloaded from the gitlab page, or autogenerated during corry building

  Includes installation instructions, 'getting started' guide, FAQS, and full descriptions of the framework and all modules.

- *Email for questions:*

  corryvreckan.info@cern.ch

CORRYVRECKAN

- Reconstruction software developed in CLICdp collaboration
- Highly flexible and configurable, easy to understand
- Algorithms split into ‘plug-and-play’ modules
- Several interesting features: 4D-tracking, Millepede alignment, Online data monitoring, Triggered and trigger-less devices.
- Well documented and continually maintained

TUTORIAL INFO

Today (Tuesday 15th Jan '19) at 1.30pm

Come with a working copy of Corryvreckan

To do this, look at the Corryvreckan installation slide in the back-up of this presentation

BACKUP: HOW TO INSTALL CORRYVRECKAN

There are four options:

1. Compile and install Corryvreckan locally, with local ROOT version - please follow the installation instructions in the user manual.
2. Use the Docker images - please refer to the user manual
3. Use Corryvreckan on LXPLUS using the centrally provided version on CVMFS. For this, you only need to source the appropriate script and you are ready to go:

   For CERN CentOS7:

   source /cvmfs/clicdp.cern.ch/software/corryvreckan/<version>/x86_64-centos7-gcc7-opt/setup.sh

   For CERN Scientific Linux 6:

   source /cvmfs/clicdp.cern.ch/software/corryvreckan/<version>/x86_64-slc6-gcc7-opt/setup.sh

4. Compile and install Corryvreckan locally or on LXPLUS, while using CVMFS version of ROOT - this works only for SLC6 and CentOS7 systems - install the CERN CVMFS daemon and source appropriate ROOT version using its .sh-script. Then compile Corryvreckan.

- **For all options including dependencies from CVMFS:** It might take a while until the CVMFS cache is populated with the necessary libraries when starting the program for the first time.
- More detailed instructions can be found in the "Installation" chapter of the Corryvreckan: *https://gitlab.cern.ch/corryvreckan/corryvreckan*.