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 CORRYSRECKAN?

- 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:
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 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
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…

Event length user defined in “Metronome” module
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:

I. 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
HOW TO USE CORRY: 2. THE CORRY EXECUTABLE

corry -c config.cfg

Path to configuration file
HOW TO USE CORRY: 2. THE CORRY EXECUTABLE

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

Sets the global log verbosity level

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

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

Detector parameters

Framework 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…
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 chi²
  - 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

- **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
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
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](https://gitlab.cern.ch/corryvreckan/corryvreckan).