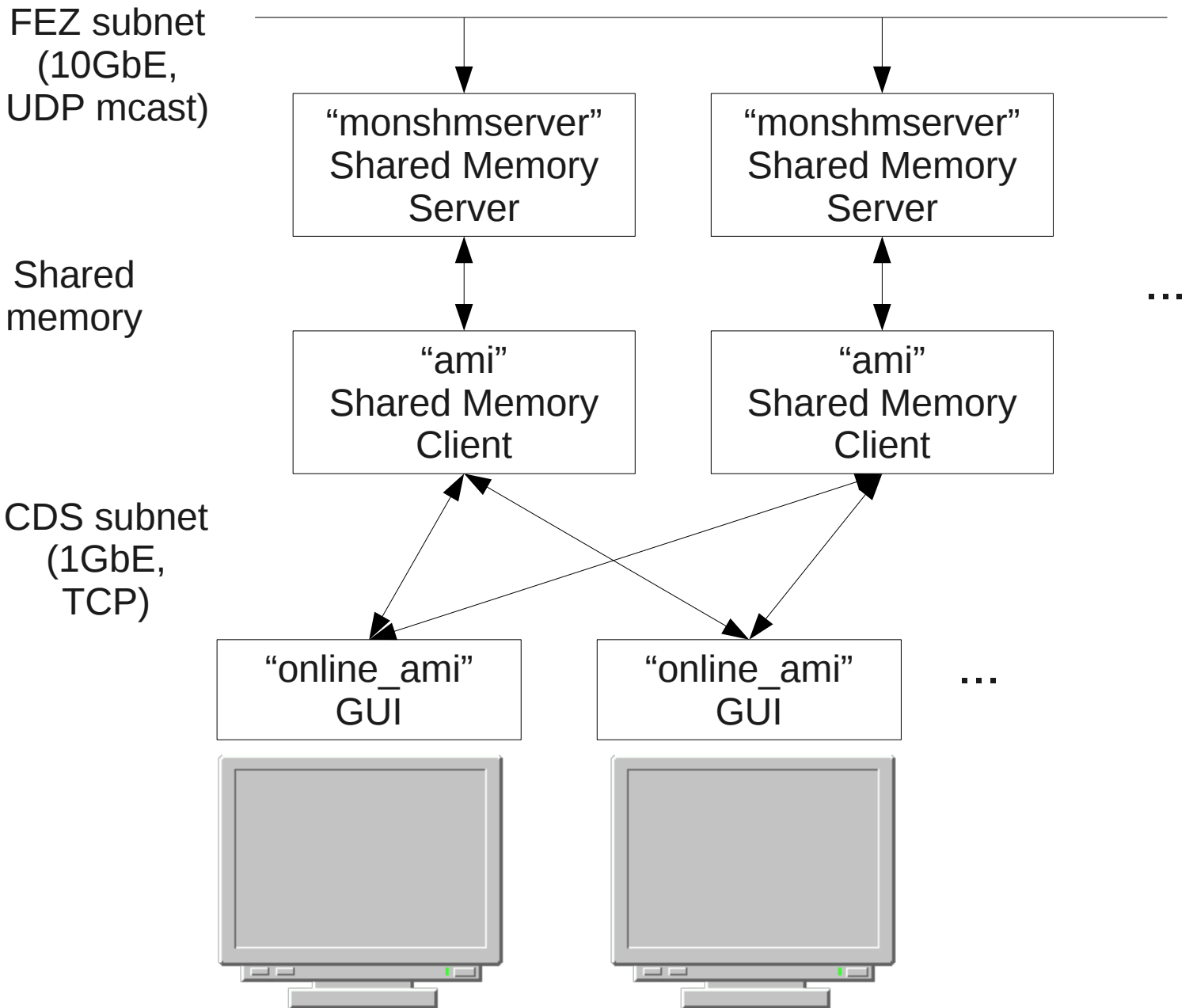
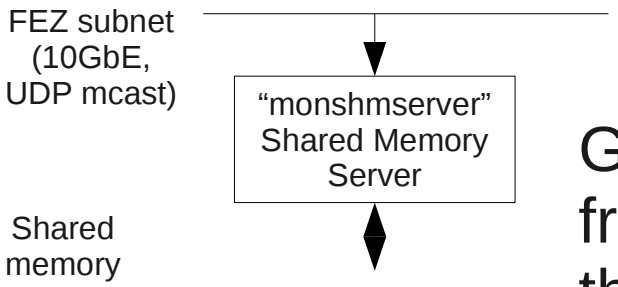
Online Analysis Design

Matt Weaver
January 17, 2012

Apps


FEZ subnet
(10GbE,
UDP mcast)
"monshmserver"
Shared Memory
Server
"monshmserver"
Shared Memory
Server
...
Shared
memory
"ami"
Shared Memory
Client
"ami"
Shared Memory
Client
CDS subnet
(1GbE,
TCP)
"online_ami"
GUI
"online_ami"
GUI
...

monshmserver



Generic application to receive datagrams from the DAQ private network and push them into shared memory.

Analysis designs are independent of DAQ network/partitioning design

Transitions are cached for late-arriving clients, so clients can learn the full current DAQ state.

L1Accepts are served promptly or dropped.

monshmserver - parameters

monshmserver -p <platform> -i <node mask> -n <buffers> -s <buffersize> -P <shm tag> -u <tag ext> -c <nclient queues>

platform distinguishes separate DAQ instances on the same network

node mask is a bit mask of recording nodes' data to monitor

nbuffers is the number of event buffers allocated in shared memory

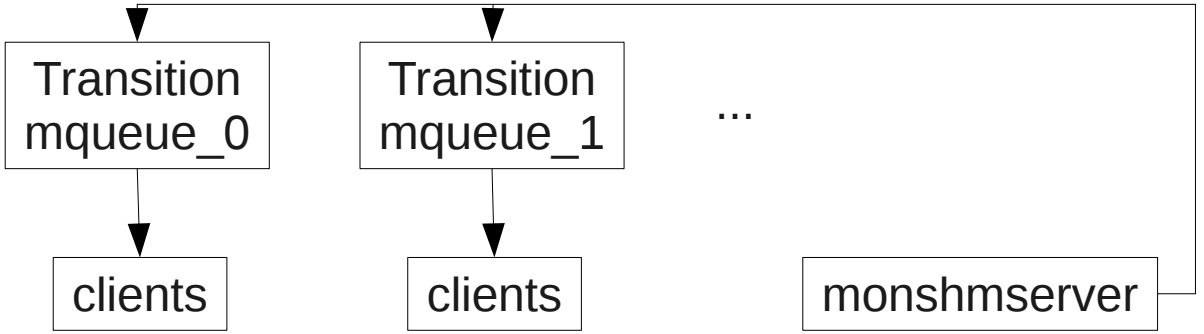
buffersize is the size of each event buffer in bytes

shm tag is a string used to identify the shared memory segment and the message queues

tag ext is merged with <*shm tag*>

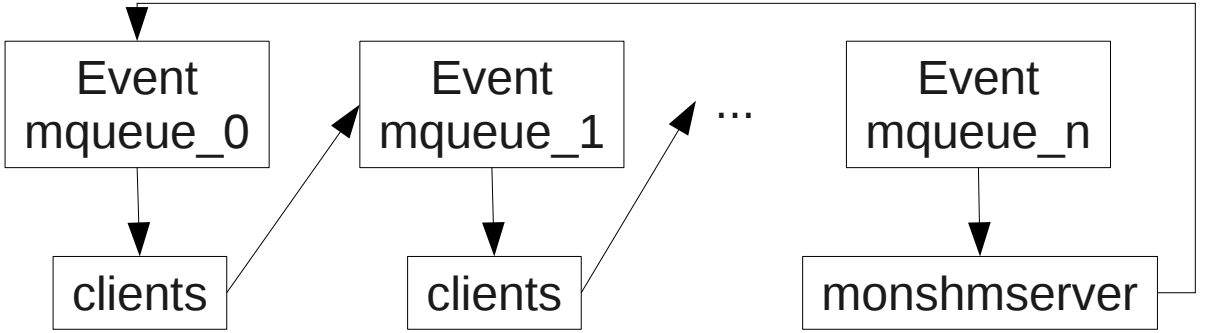
nclient queues is the number of serial message queues to setup

monshmserver - dataflow



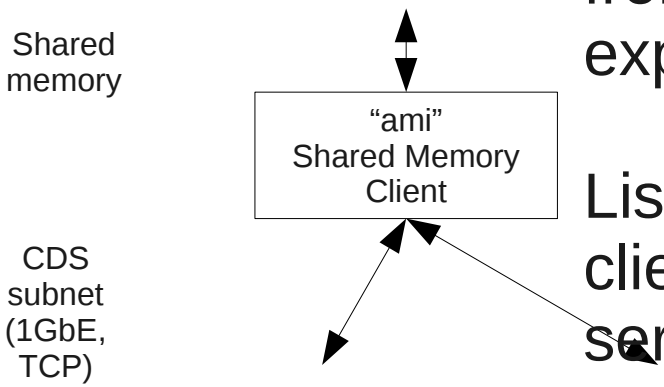
Arriving transitions are bcast to all queues.

New clients are served cached transitions (ucast) to reach the current DAQ state.



If a free event buffer is available, arriving L1Accepts are ucast to the first queue.

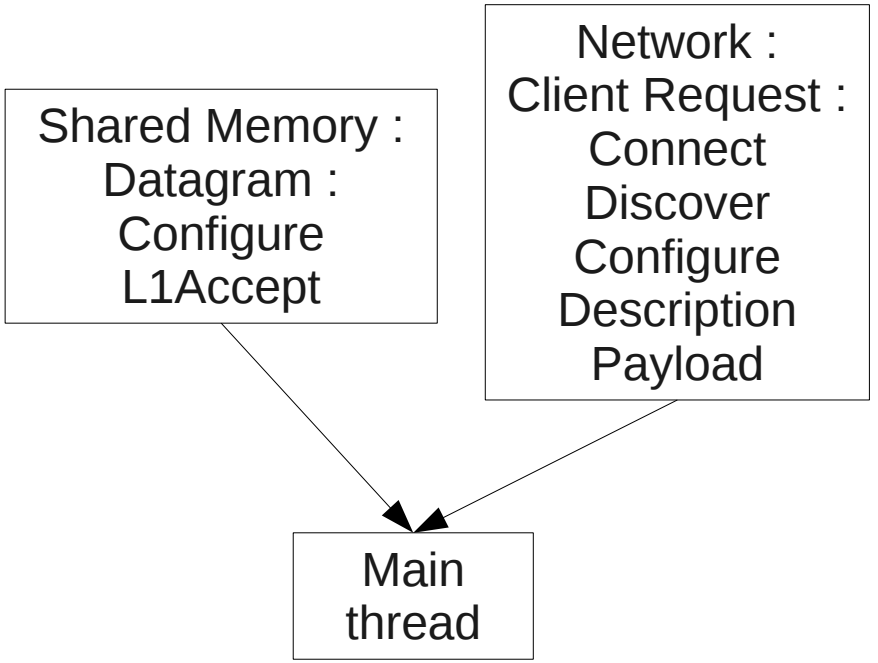
ami



Custom application to receive datagrams from shared memory, perform analyses, and export viewable data (plots).

Listens for TCP connections from analysis clients. Clients make requests to the *ami* server for analyses and updated plots.

ami - inputs



One shared memory input

Many network clients

Processing of client requests is serialized with datagram handling.

ami – handling client requests

Discover – transmit description of input event data set
list of scalars {EPICS vars, diodes, ...},
waveforms, images

Configure – setup all analyses associated with the client
Analysis : input data, event filter, operator, output data
input data : from the discovered data or
from the output of another Analysis
event filter : none, var range, logic and, logic or
operator : none, average, integral, entry math, bin math,
edge finder, peak finder, xy projection, rφ projection,
history chart, ...

Description – transmit description of output data set
list of plots

Payload – transmit output data set

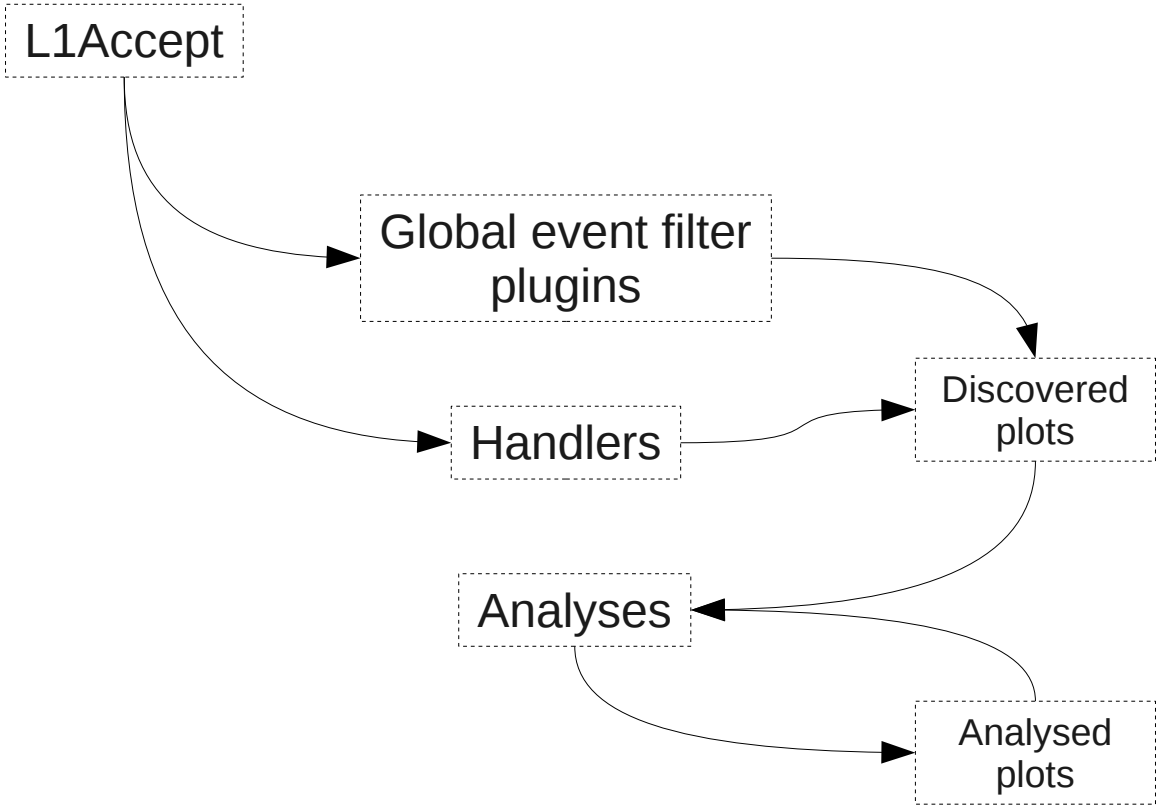
ami – handling *Configure* datagram

Destroy all previous handler/analysis setup

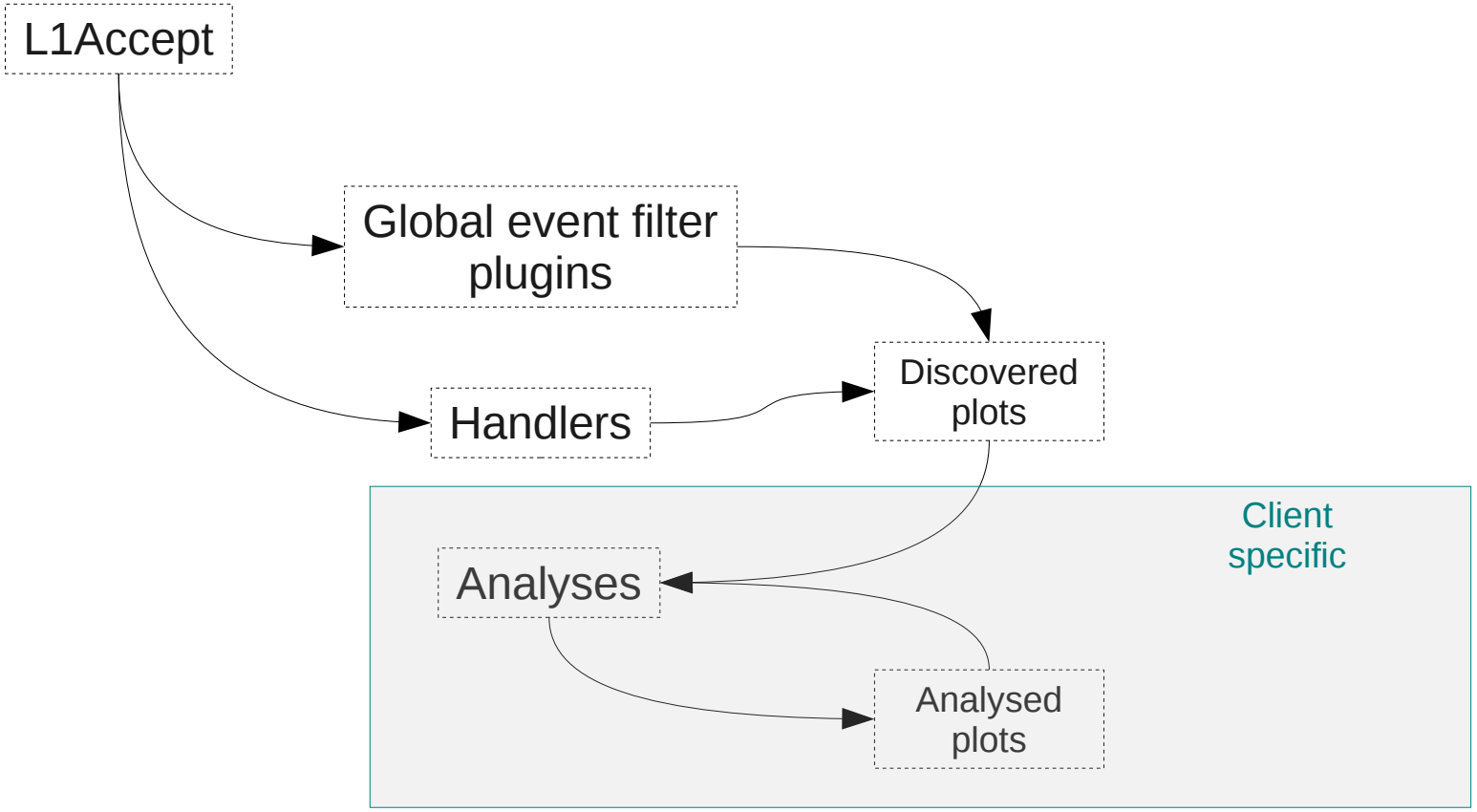
Discover data stream components {detectors,BLD}
and register handlers

Advertise discovered components (and plug-in modules) to network clients.

ami – handling *L1Accept* datagram



ami – handling *L1Accept* datagram

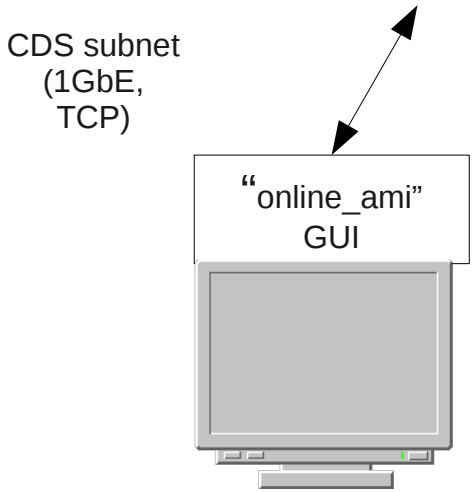


Order matters:
Each Analysis' input data must be computed beforehand.
It is up to the clients to configure the analyses in the correct dependency order.

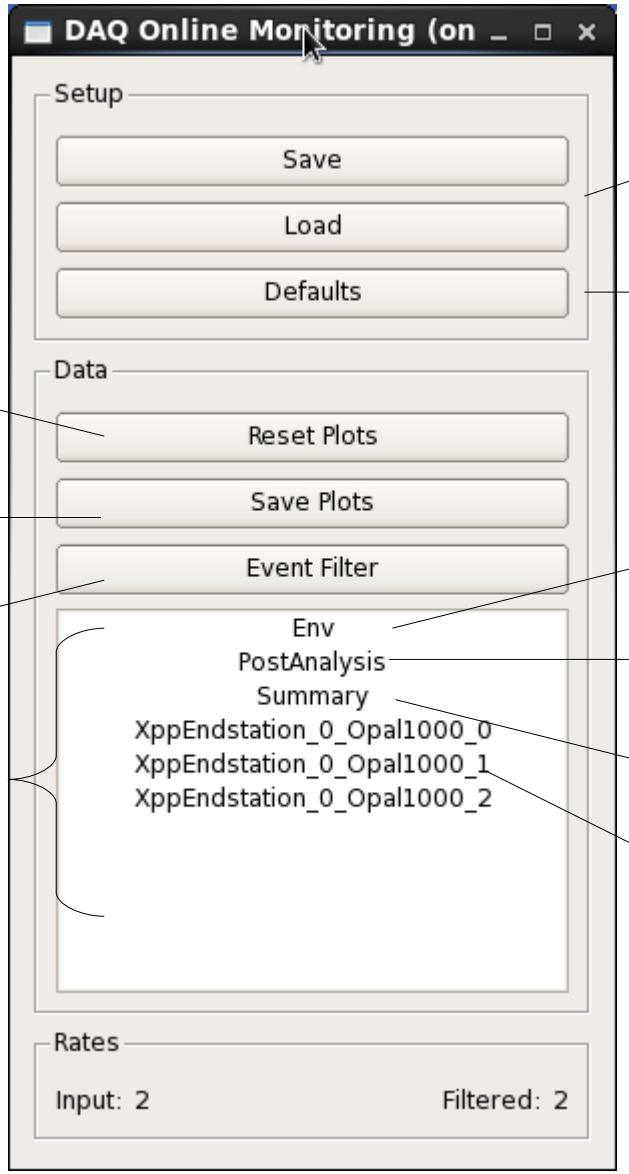
online_ami

Qt-based GUI that serves as a network client to ami.

Receives users' analysis configurations and displays resulting plots.



online_ami – main window



GUI state is stored as XML (+autosave)

Plot grid setup, ...

Clear all plots' statistics

Write each plot's contents to a text file.

Select global event filter

Create plots from input scalars

Create plots from analysis output scalars

A predefined analysis

Create plots from input camera images

Discovered input data

Each item launches a window that generates a separate Analysis chain

PostAnalysis depends on others, so SW must configure it last

online_ami – “Env”(”PostAnalysis”) display

Update displays once or continuosly

Type variable name or Select/Calculate

Define event filter

Add variable to PostAnalysis as *title*

Generate 1d histogram with specified binning

Generate history chart with prescaled time

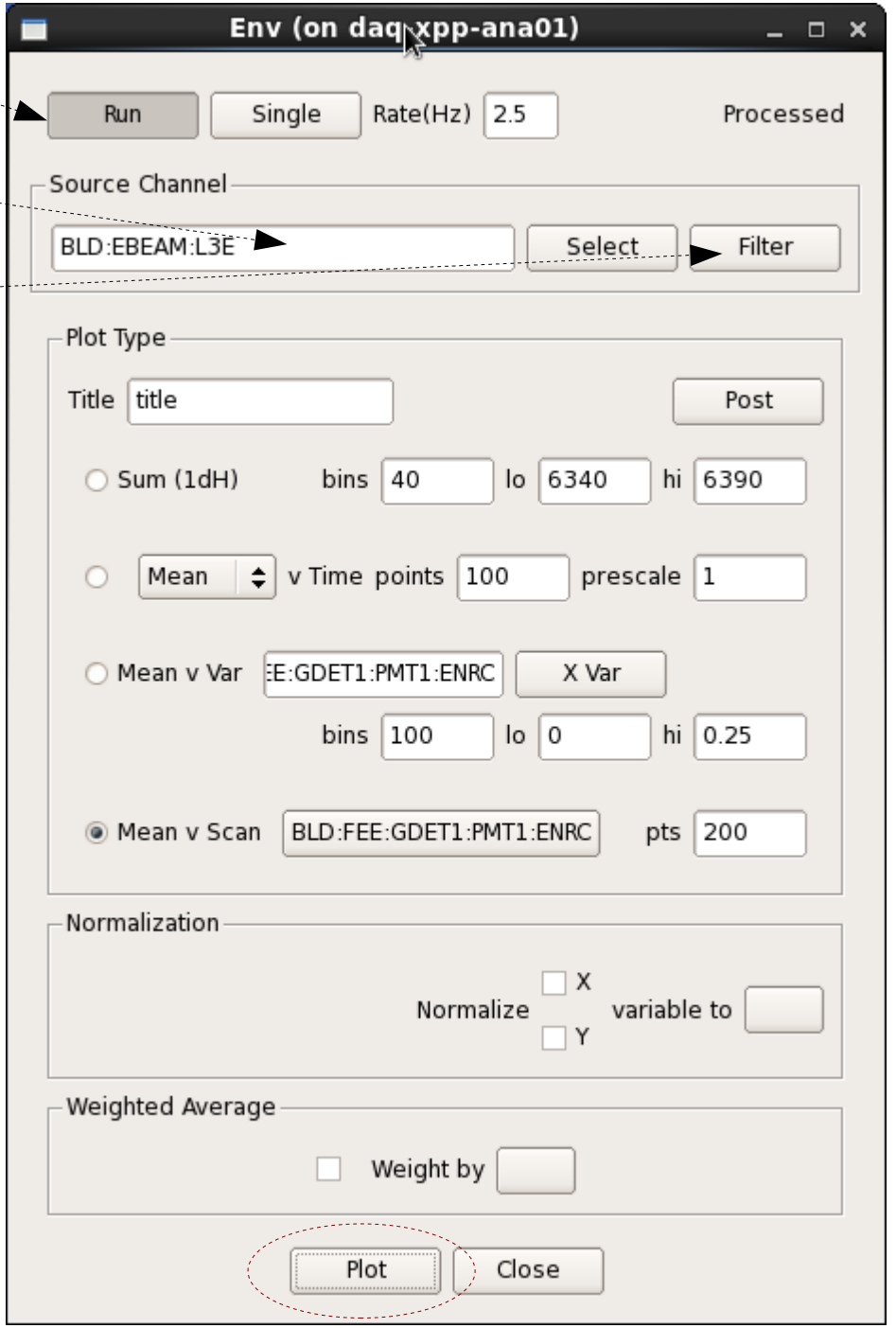
Binned 2d plot (versus Calculation)

Scatter plot (versus other variable)

Define a variable to divide into all others (flux normalization)

Weight averages by another variable

$$\frac{\sum x}{N} \rightarrow \frac{\sum w*x}{\sum w}$$



online_ami – “Env” display

Update displays once or continuosly

Type variable name or Select/Calculate

Define event filter

Add v

Generate

Generat

Binn

Scat

Define

W

binning

time

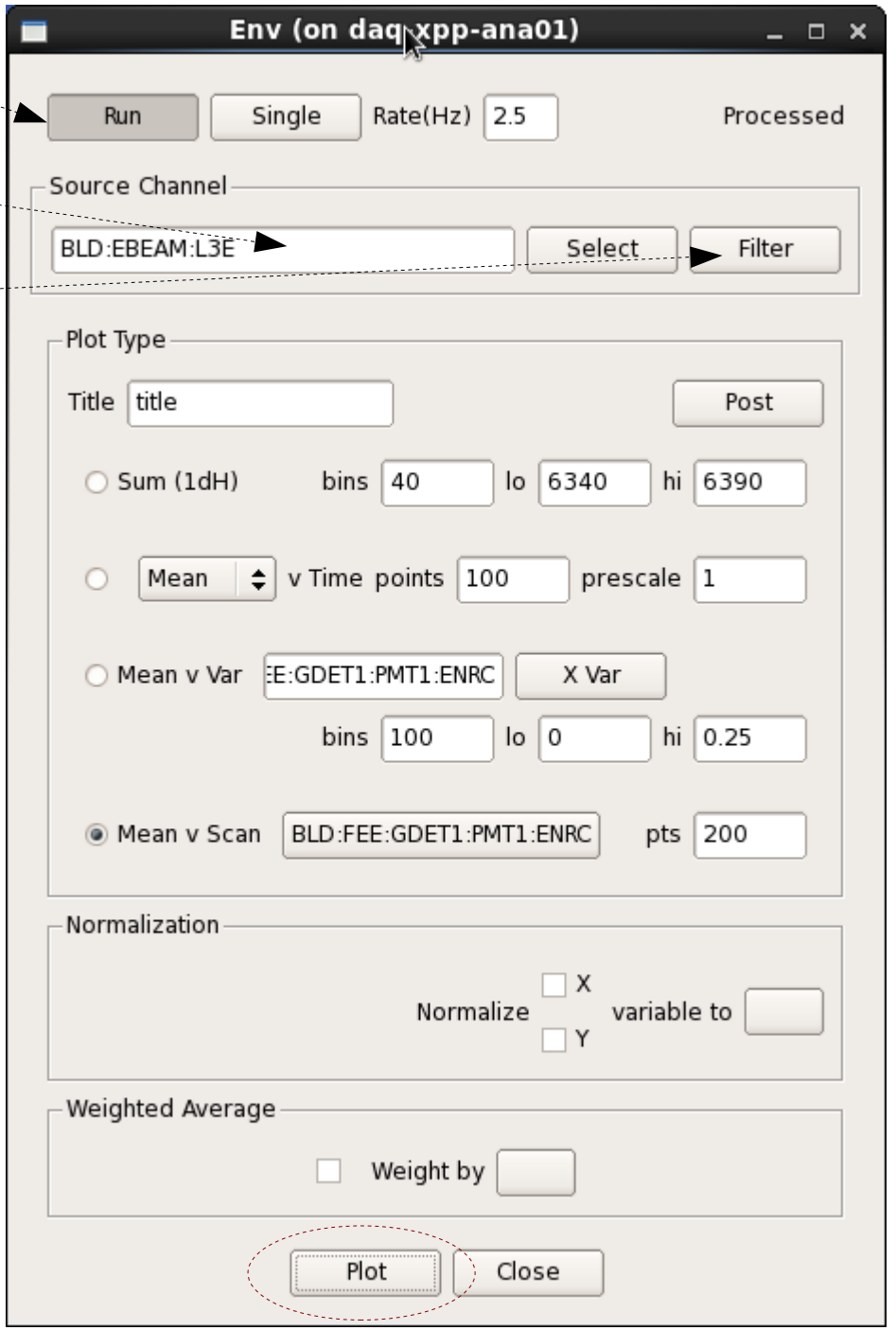
)

)

thers

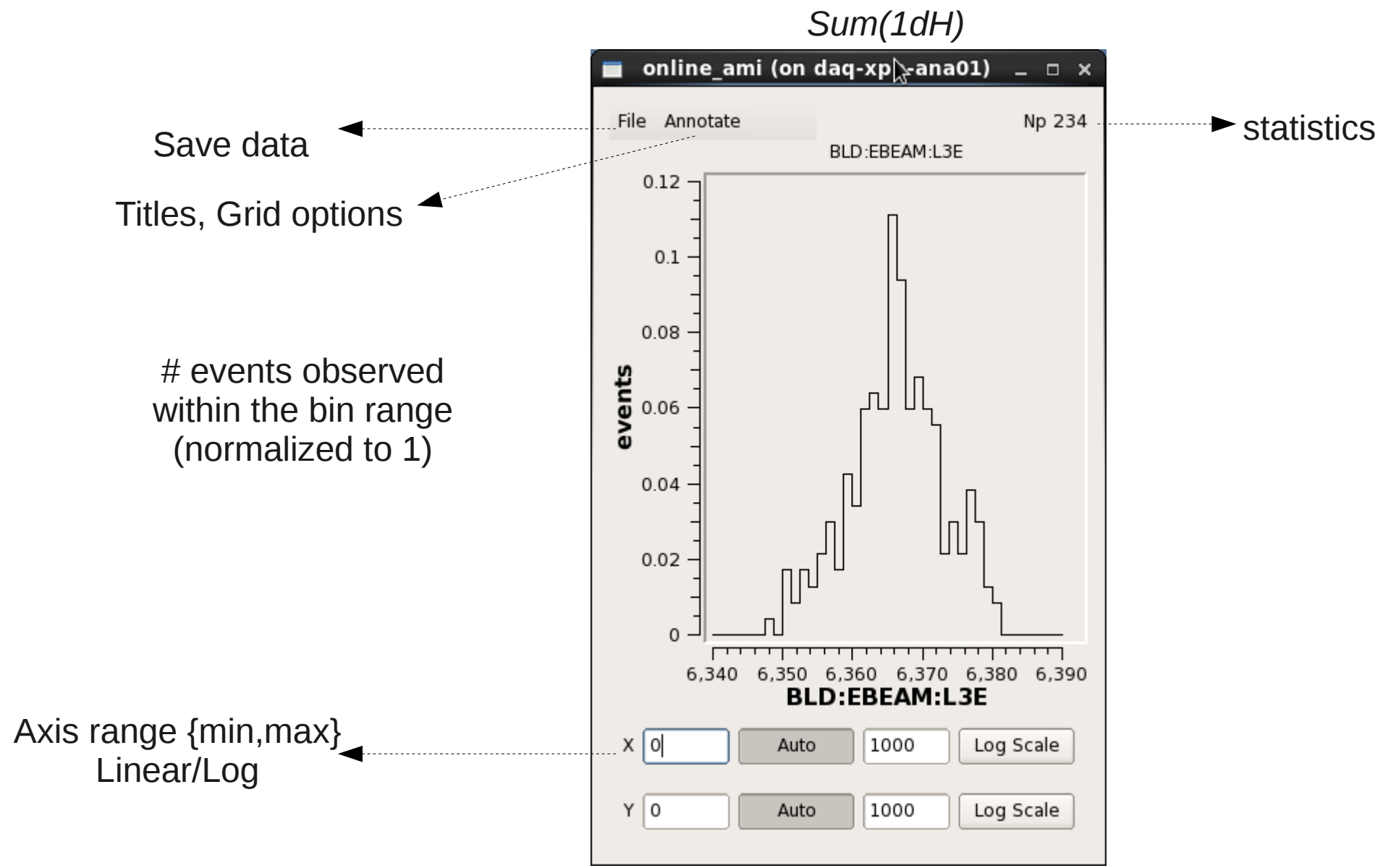
riable

Plot interface reused everywhere a scalar variable is produced



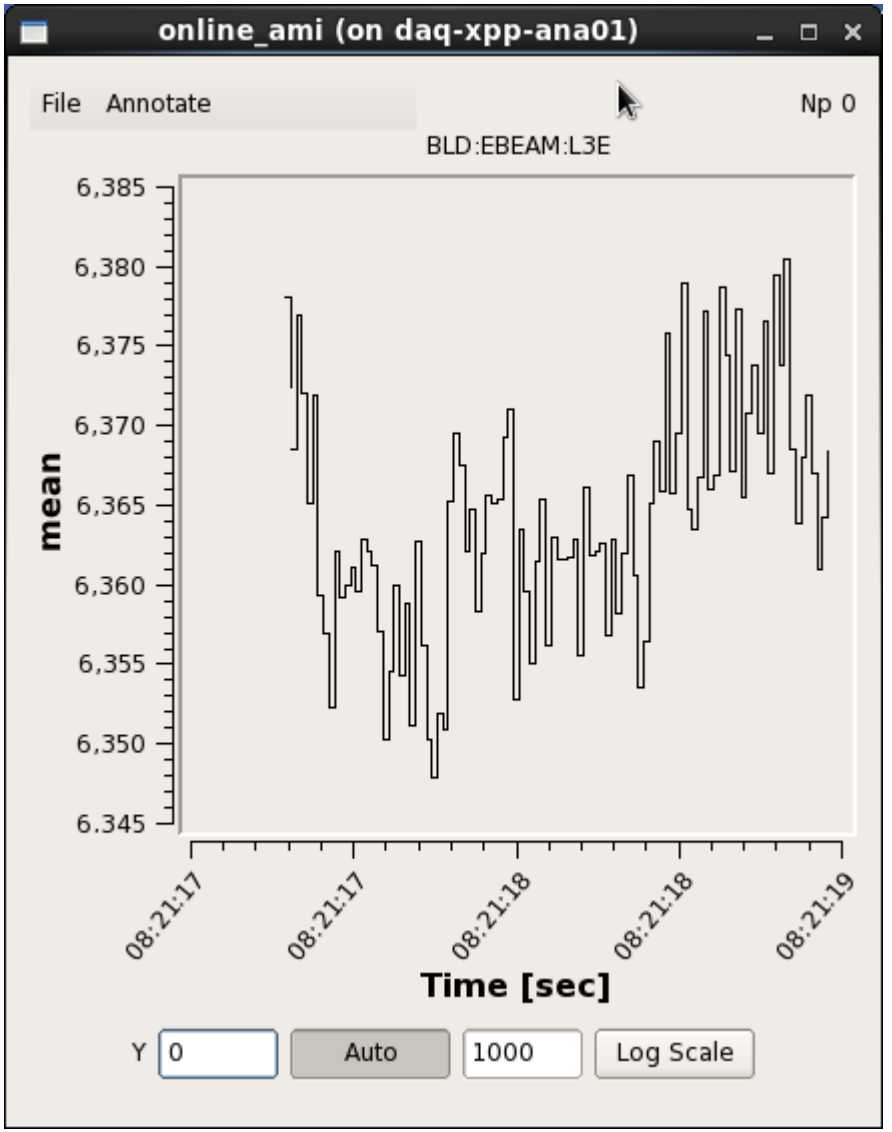
$N \quad \sum w$

online_ami – “Env” display



online_ami – “Env” display

Mean v Time

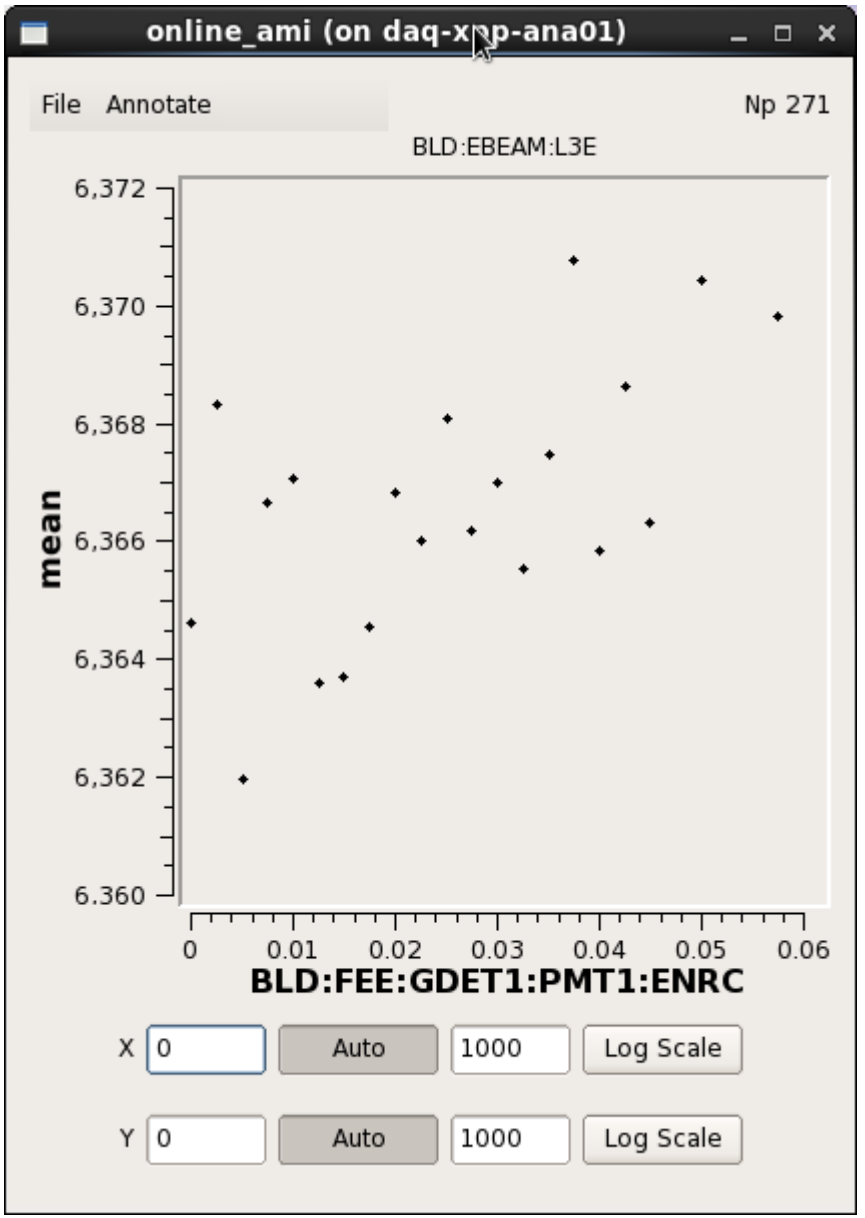


Each point is the mean of *variable* in events observed between plot updates

Time from most recent event's timestamp

online_ami – “Env” display

Mean v Var

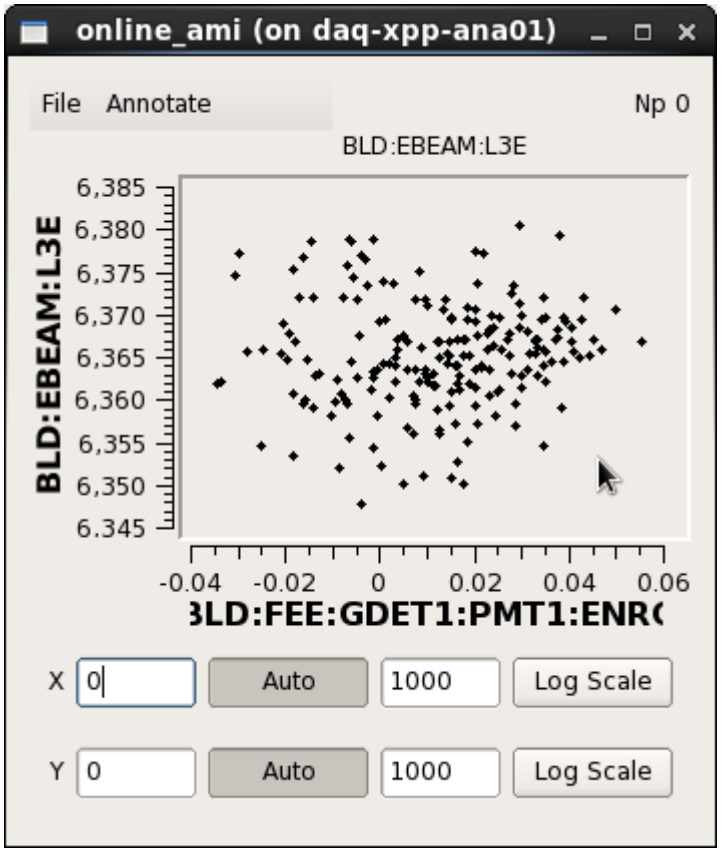


Y-axis is mean of *variable* for all events where *x-var* is within bin ranges

X-axis is equidistant bins

online_ami – “Env” display

Mean v Scan

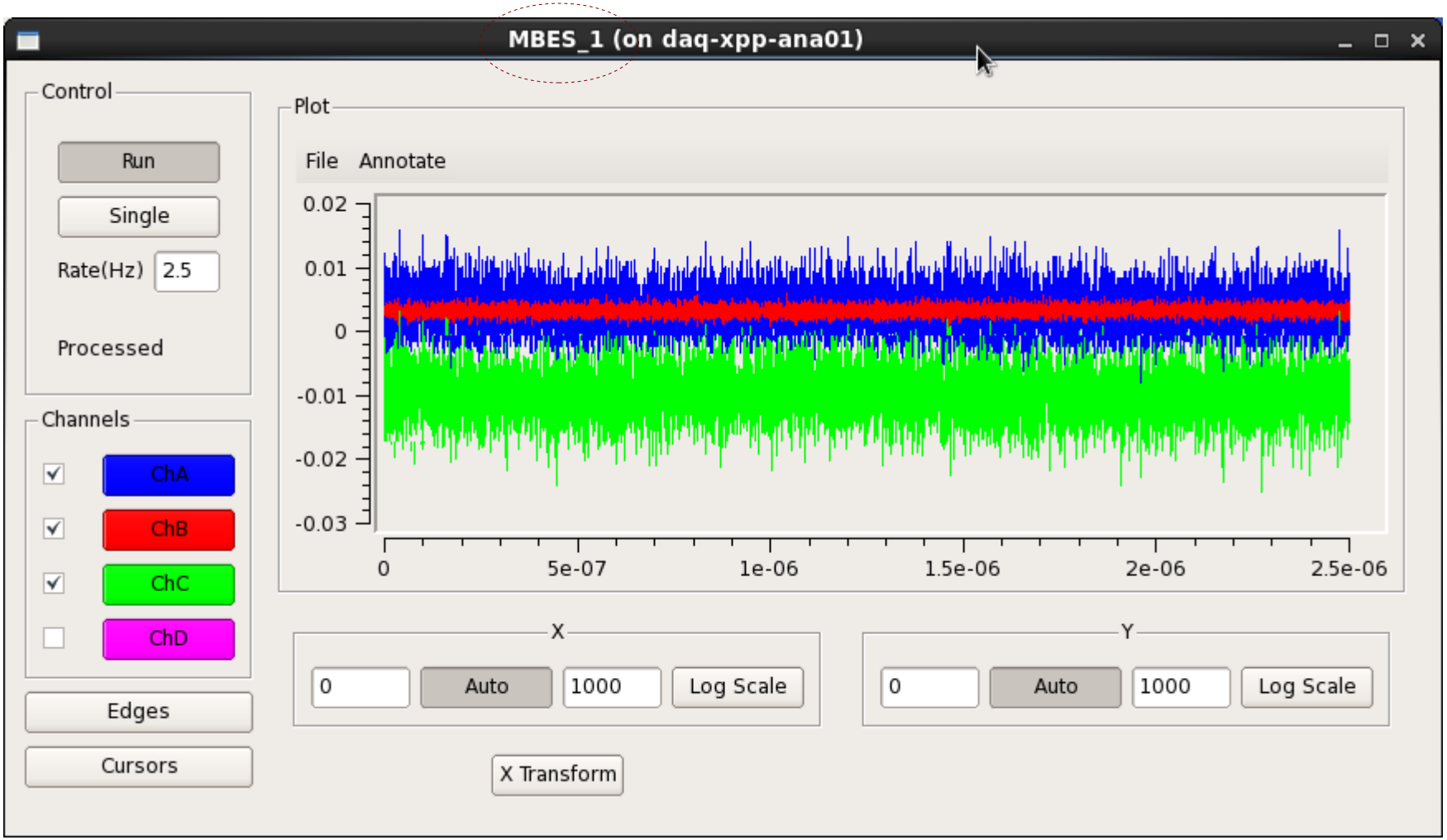


Y-axis is mean of *variable* for all events where *x-var* is the same

X-axis is most recent *N* unique values

online_ami – waveform display

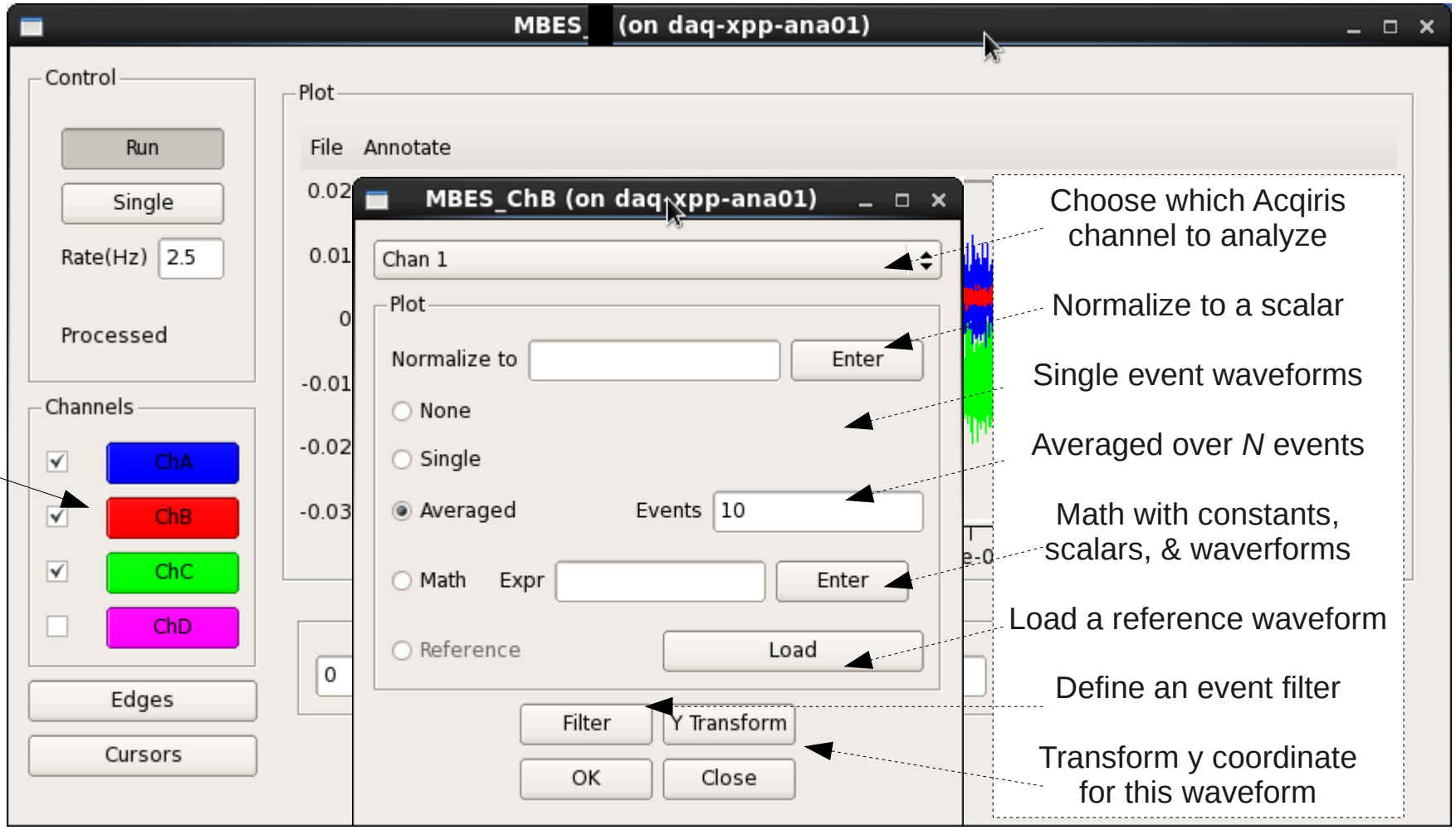
MBES - generate analysis derived from any MBES channel
MBES_1 – generate analysis derived from only MBES channel 1



online_ami – waveform display

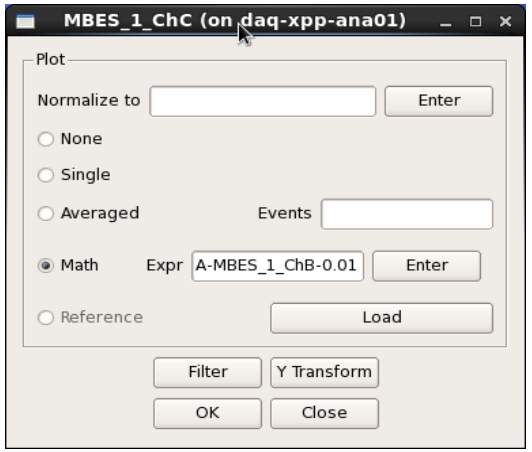
MBES - generate analysis derived from any MBES channel
MBES_1 – generate analysis derived from only MBES channel 1

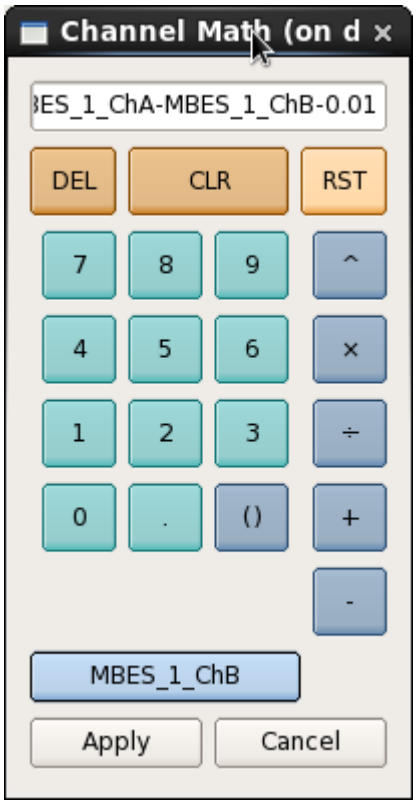
Setup initial analysis of MBES waveform

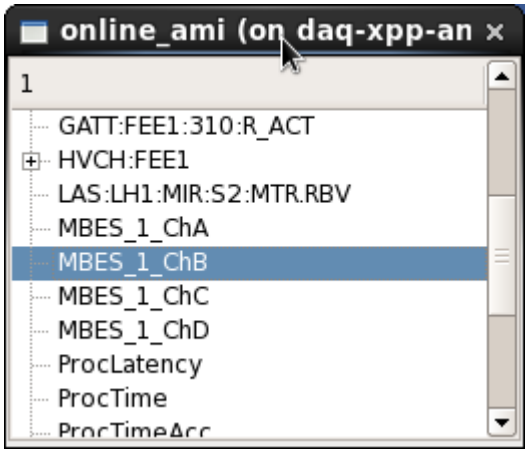


online_ami – waveform display

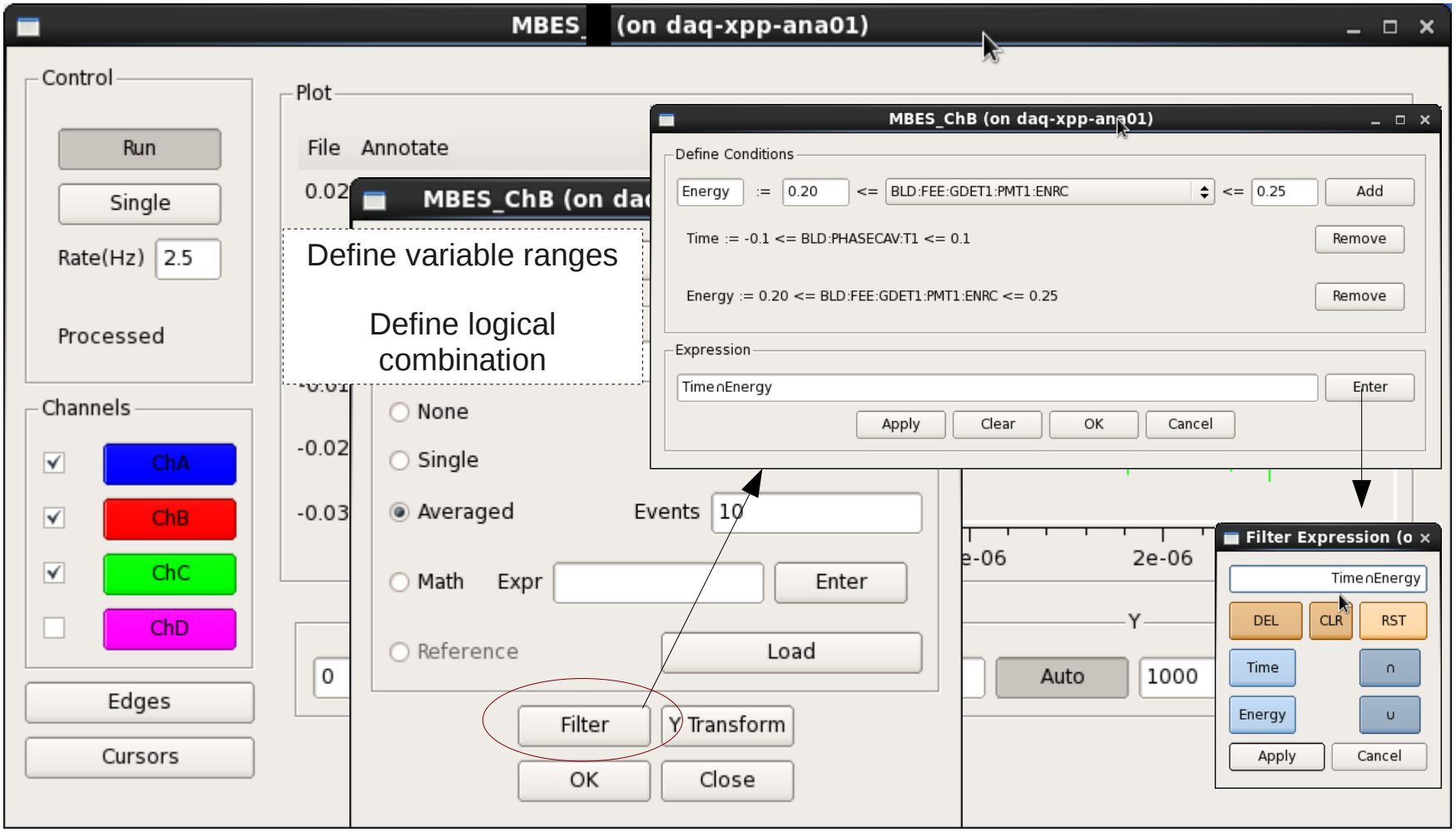
Channel math example:
ChA – ChB - 0.01(V)





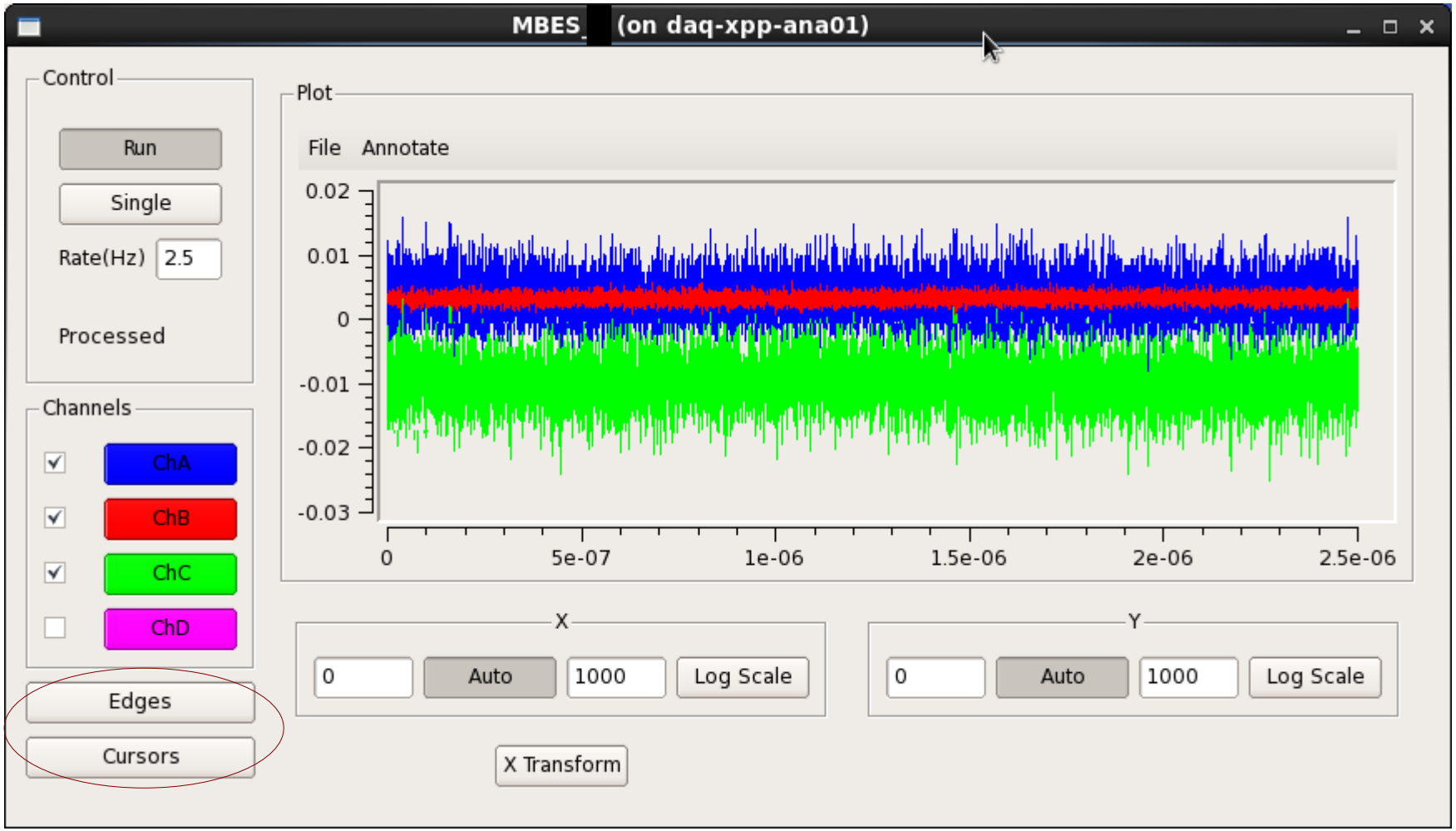


online_ami – waveform display

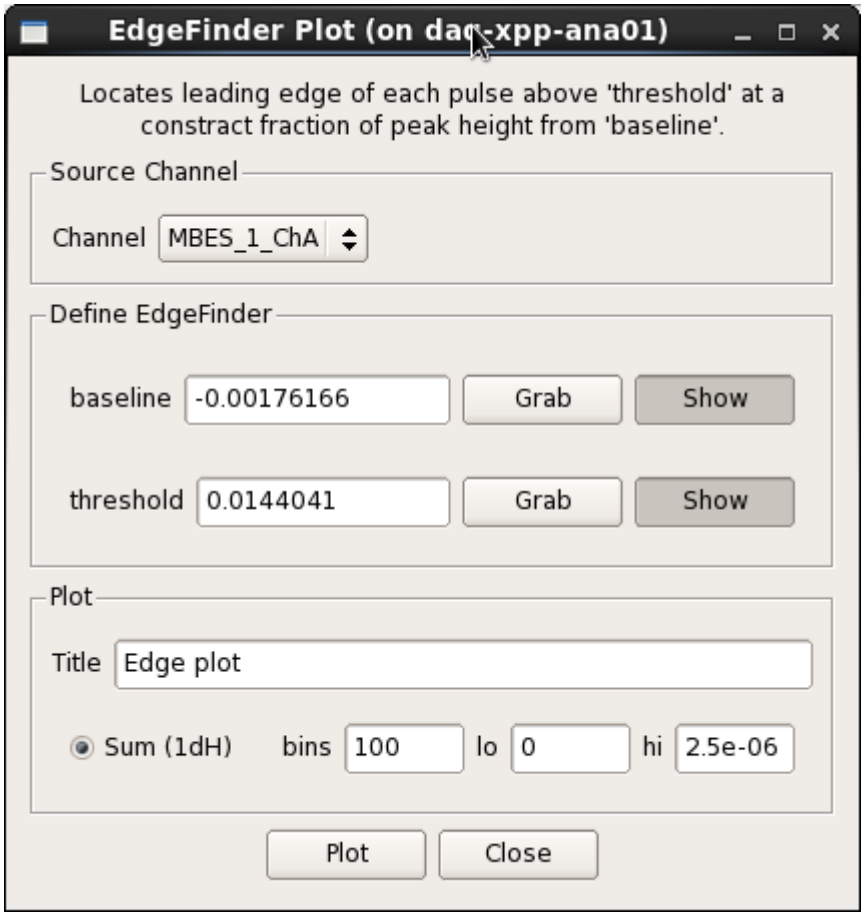


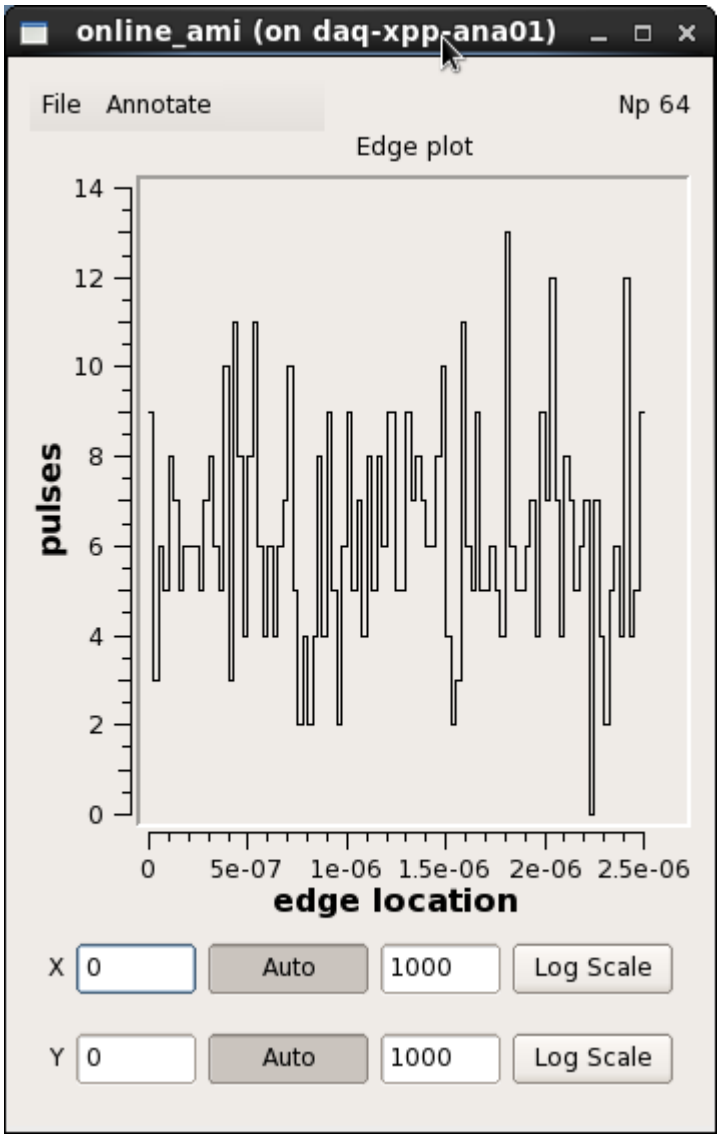
online_ami – waveform display

Additional processing

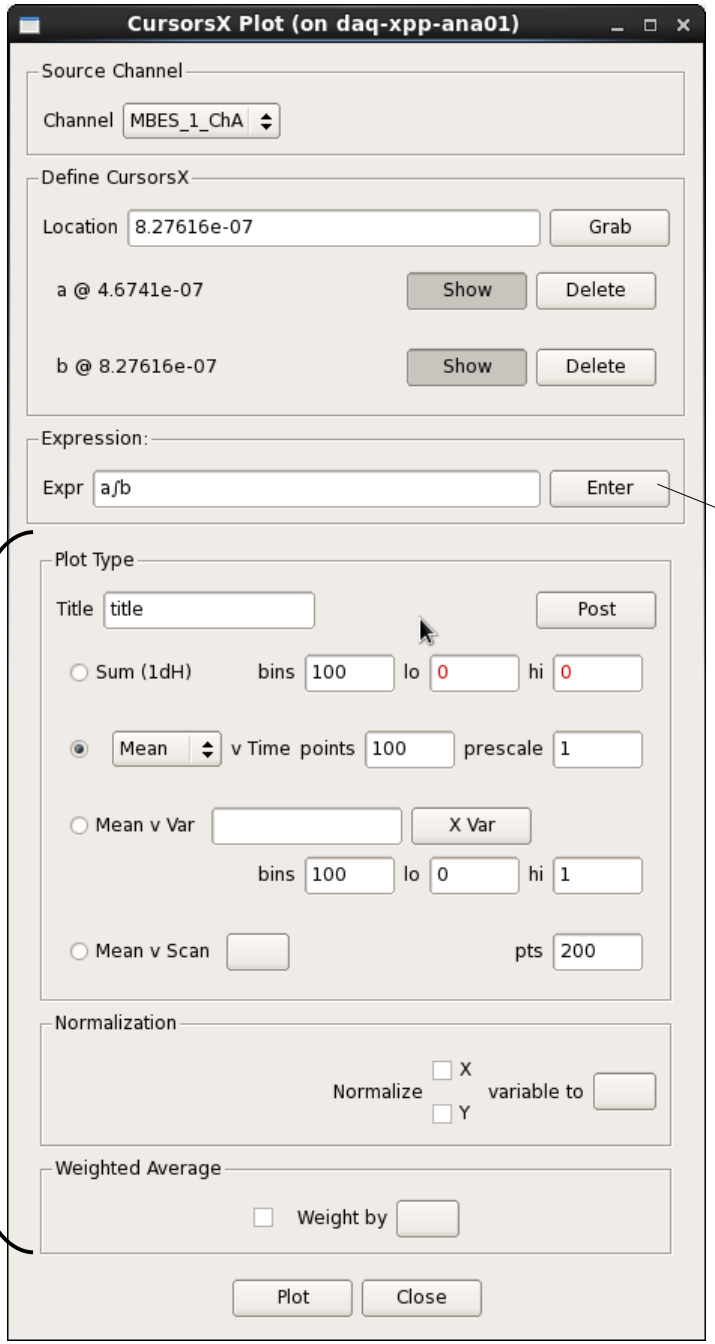


online_ami – waveform edge finder





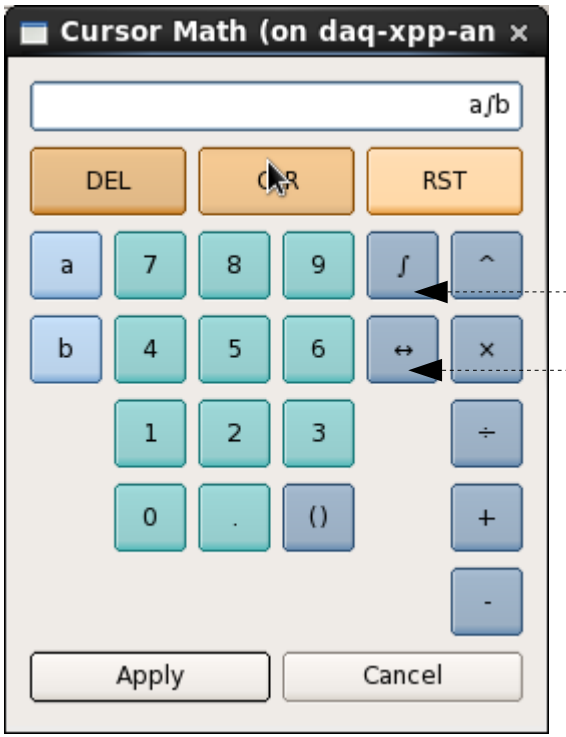
online_ami – waveform cursors



Analyze waveform contents

Define locations on waveform

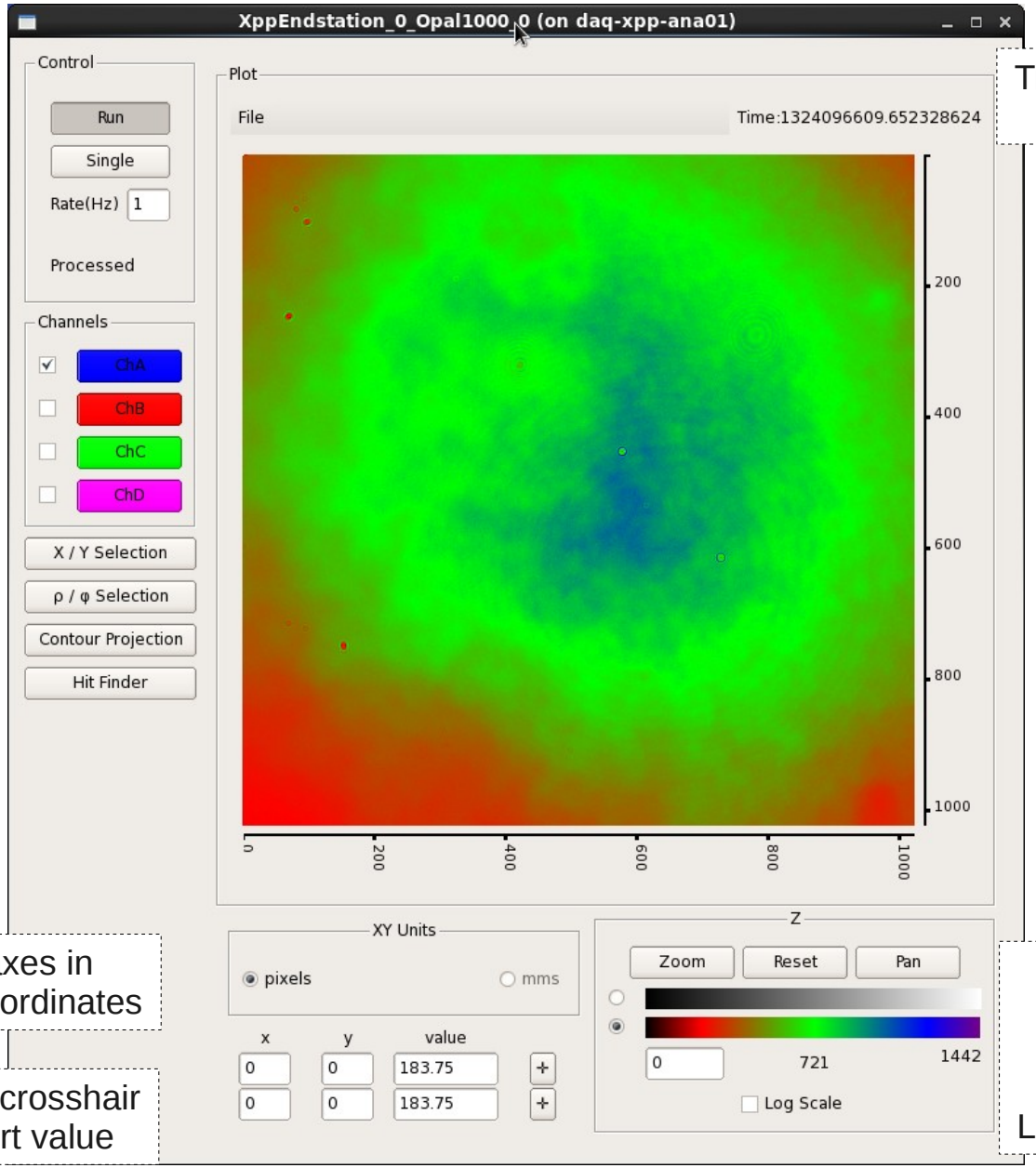
Enter calculations with those locations



Integrate btw cursors

samples btw cursors

online_ami – image display



Familiar

Timestamp of most recent L1Accept

Display axes in pixel/size coordinates

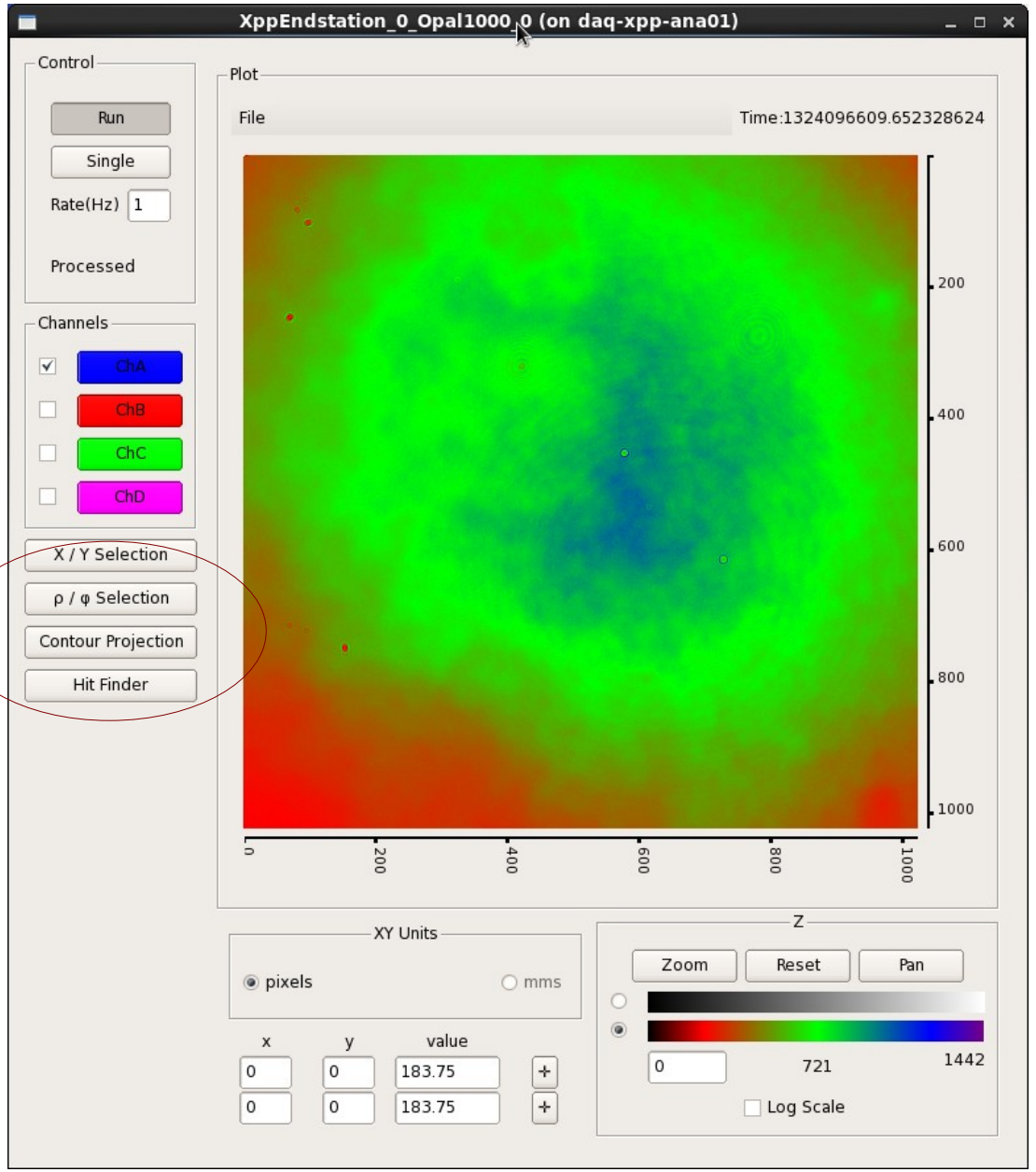
Position a crosshair and report value

Intensity scale

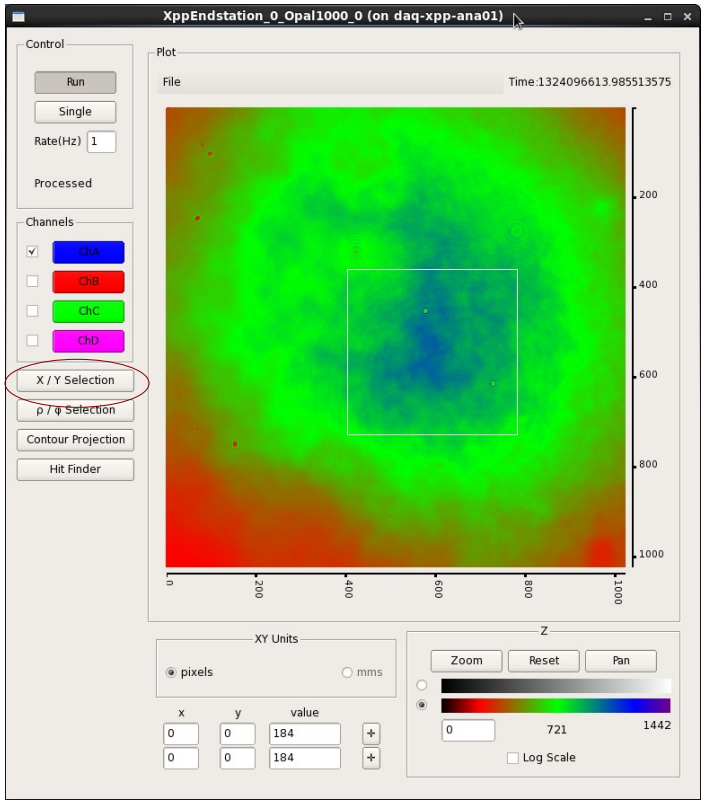
False color map
Scale offset
Log/Linear scaling

online_ami – image display

Additional processing



online_ami – image xy projection

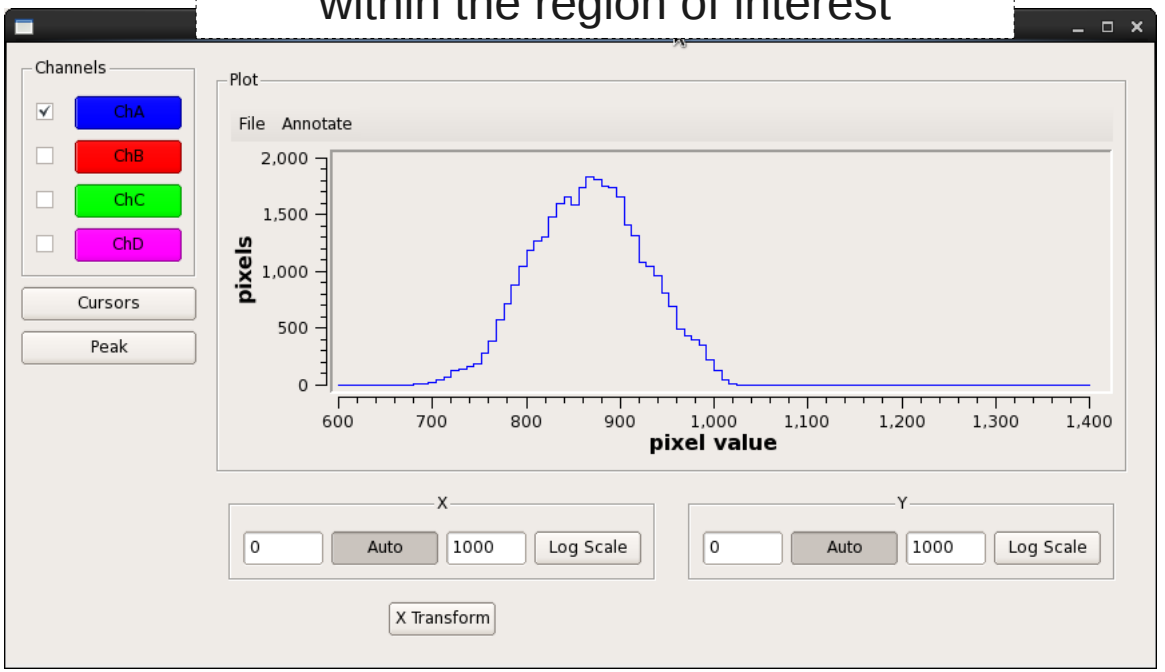


Enter or grab the region of interest

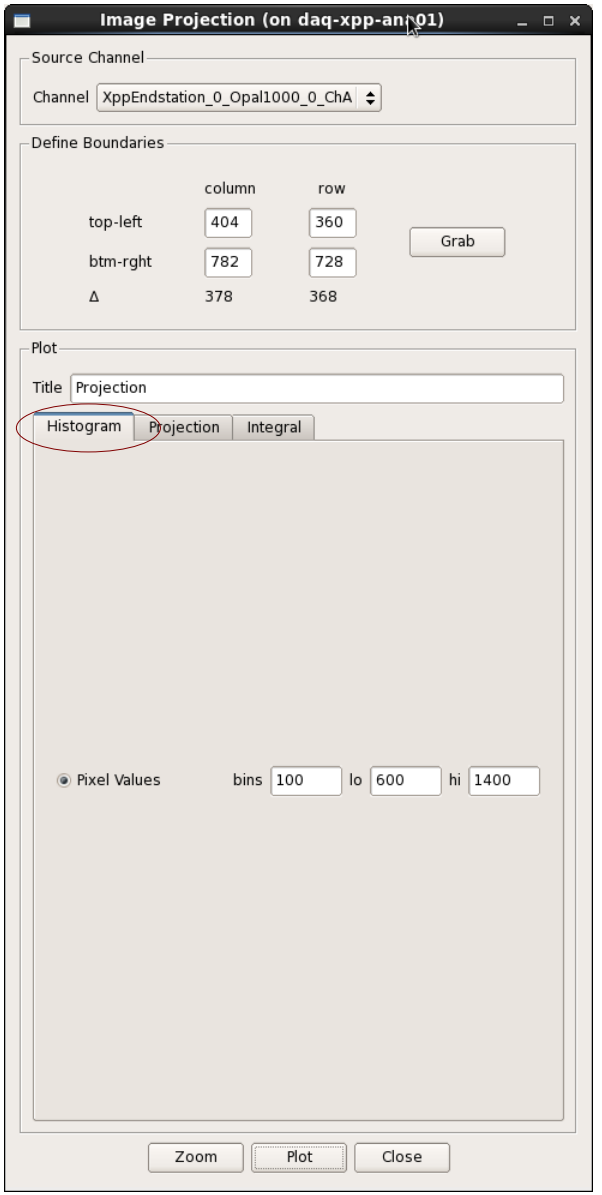
Select the processing on that region

online_ami – image xy projection

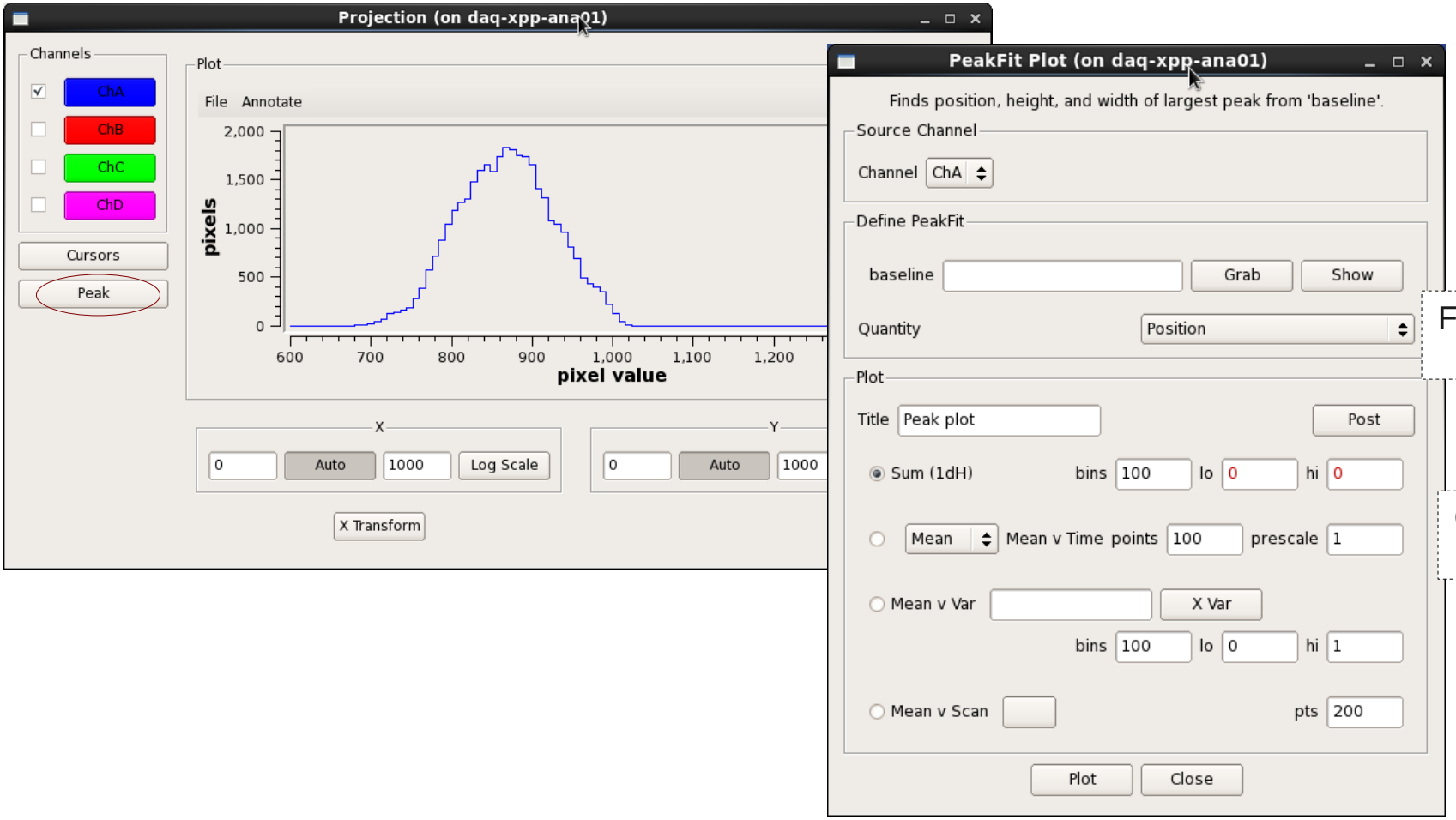
Histogram distribution of pixel values within the region of interest



Generates a 1d projection from which further processing can be done



online_ami – image xy projection

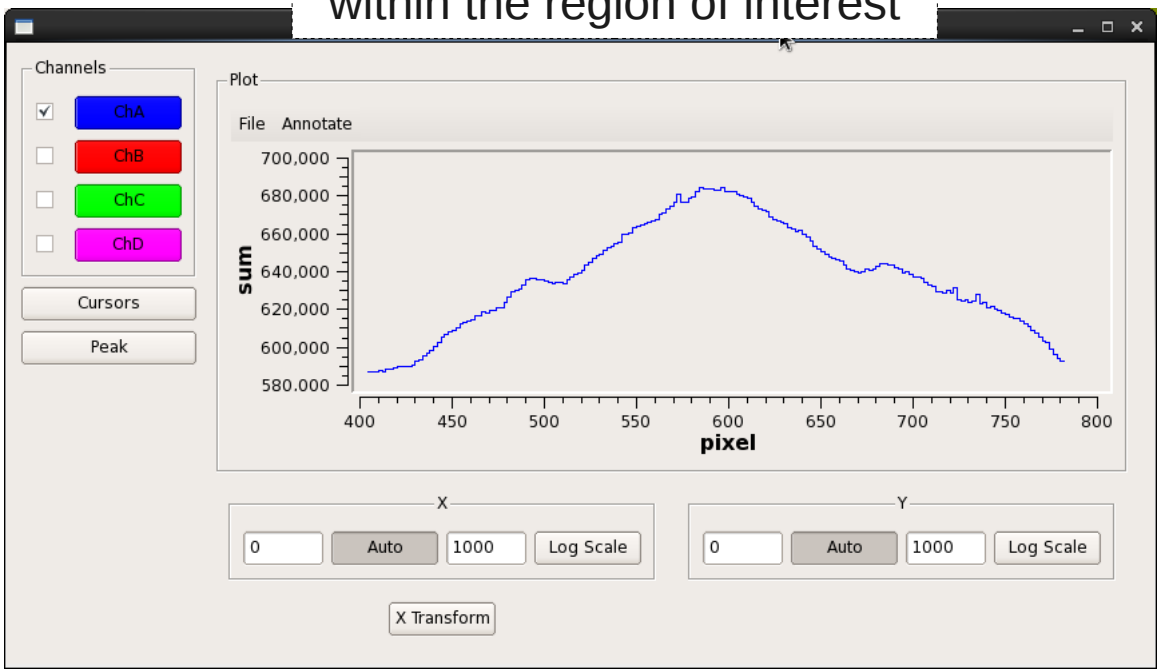


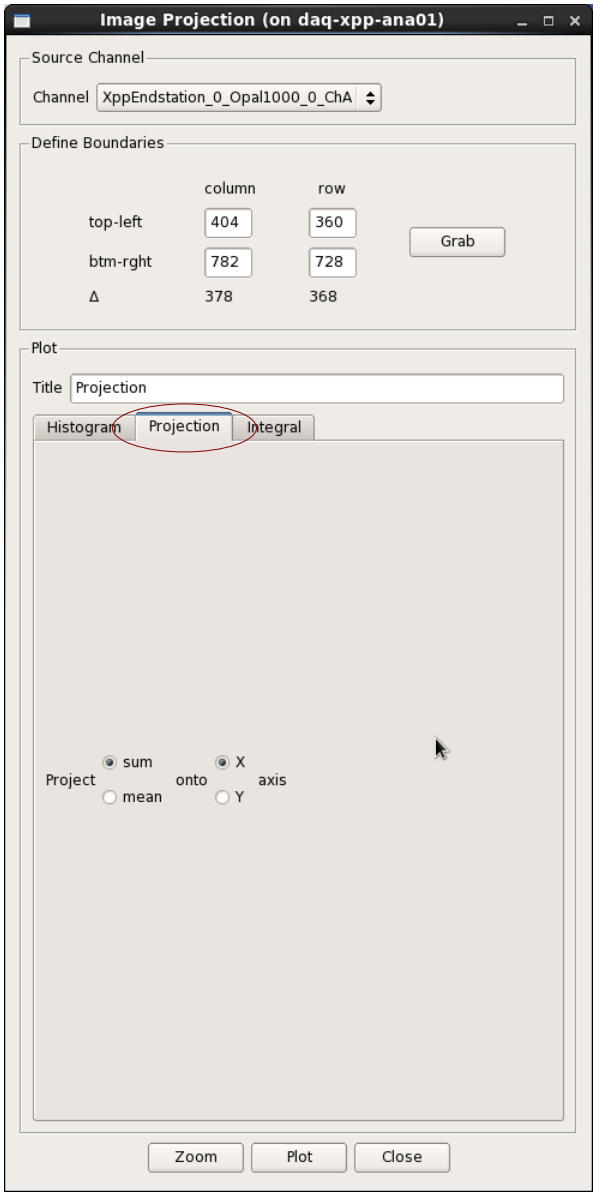
Find peak position, height, or width

Generate a plot from that result

online_ami – xy image projection

Integrate pixel rows/columns within the region of interest



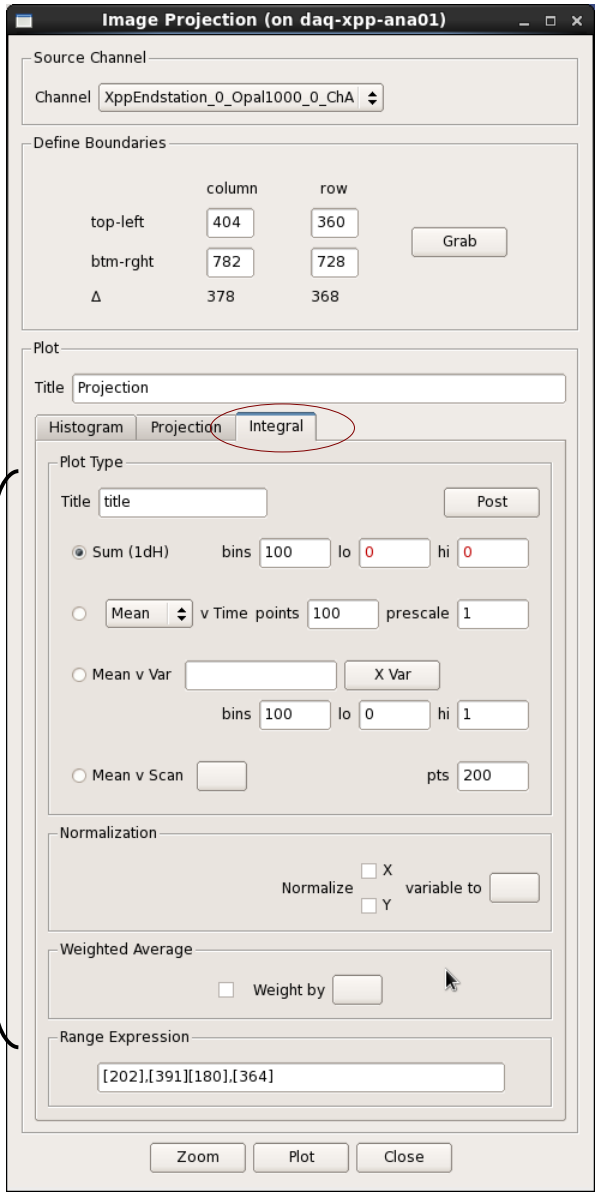


Generates a 1d projection from which further processing can be done

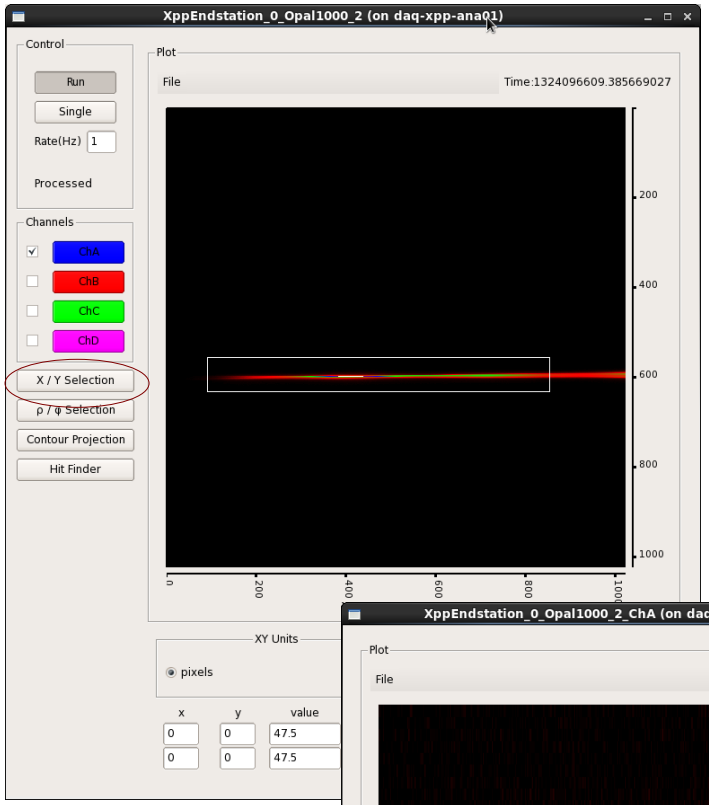
online_ami – xy image projection

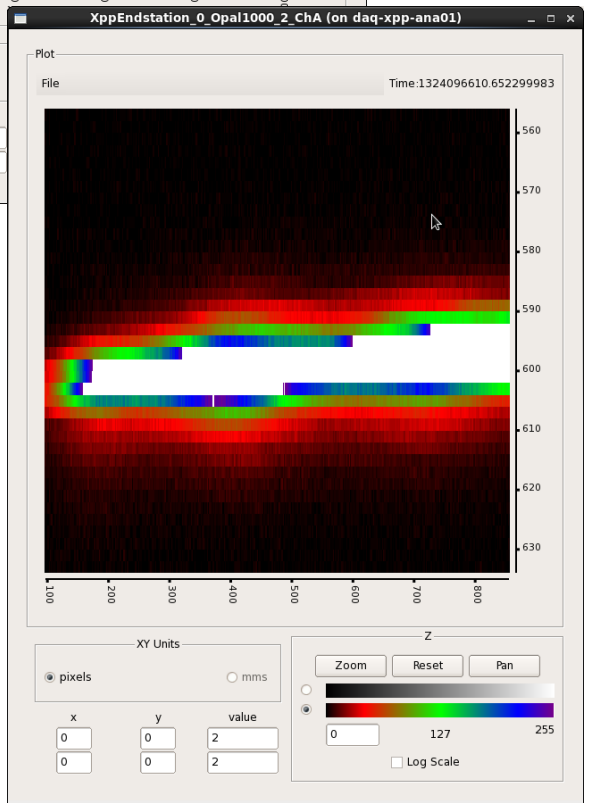
Integrate all pixel values within the region of interest

Generates a scalar input to familiar set of plots

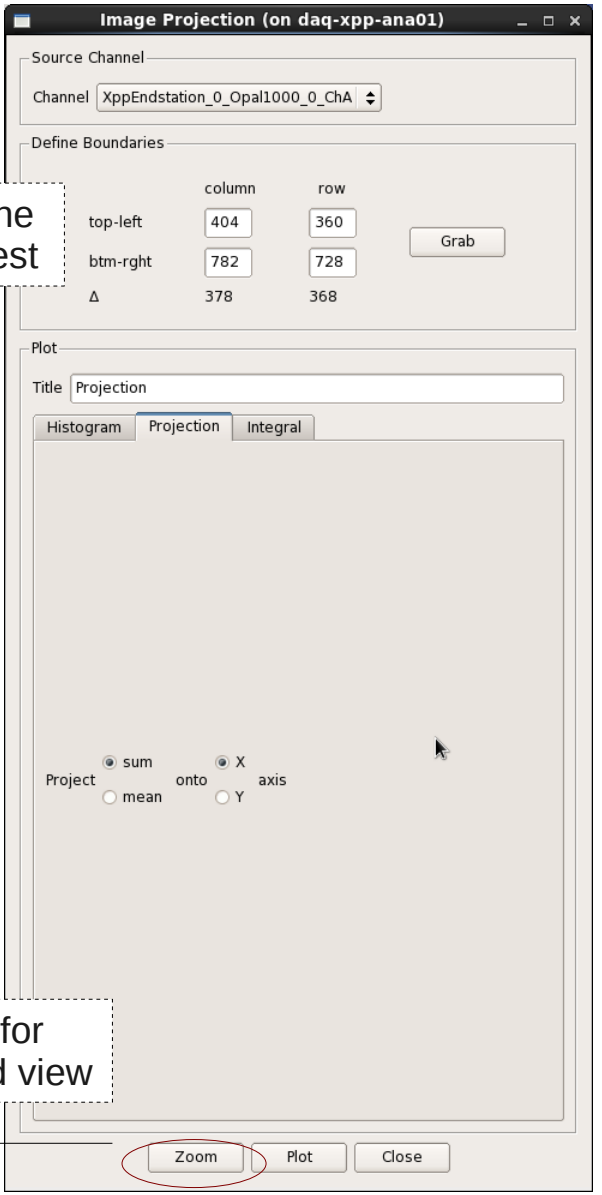


online_ami – image xy zoom



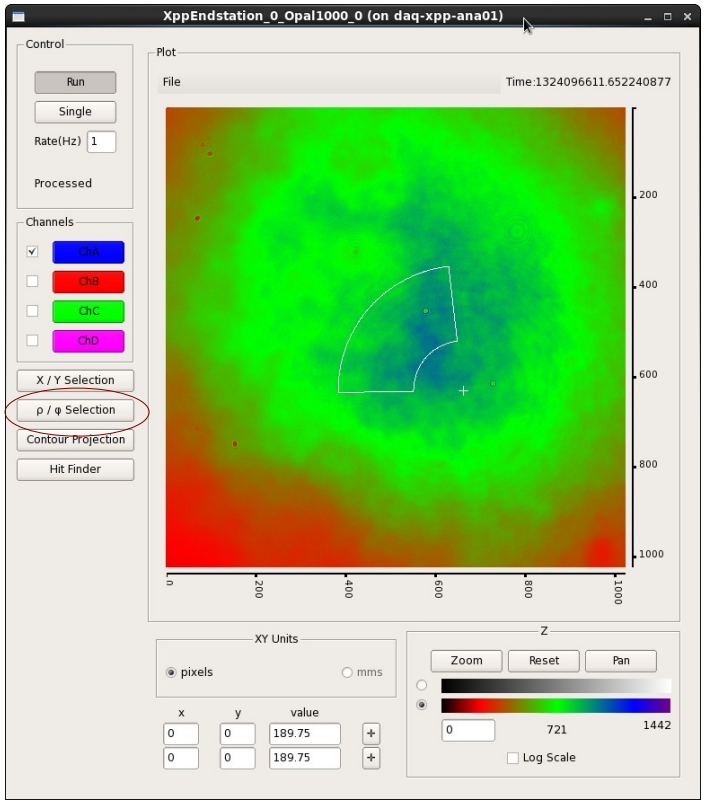


Enter or grab the region of interest



Zoom for magnified view

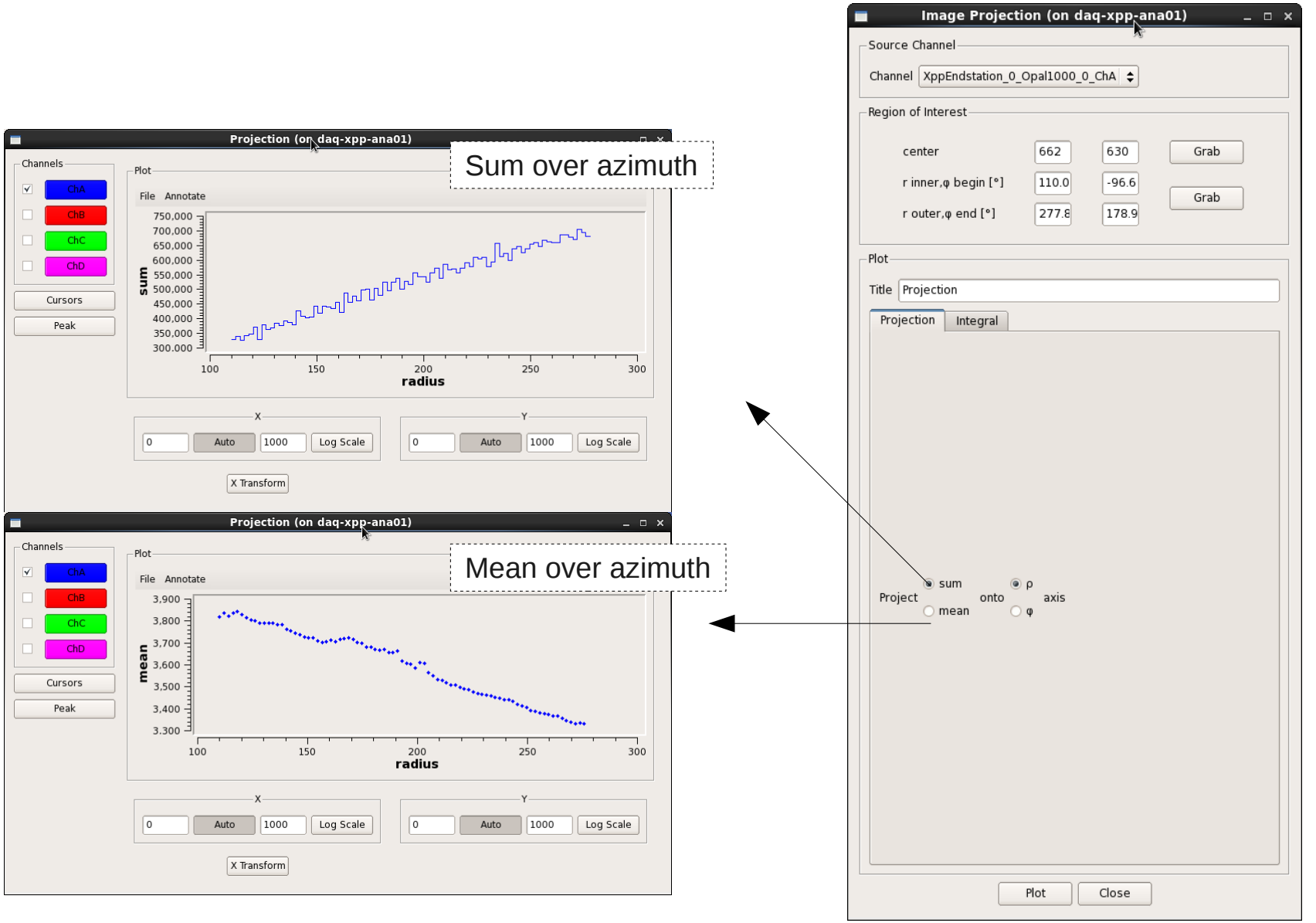
online_ami – image rφ projection



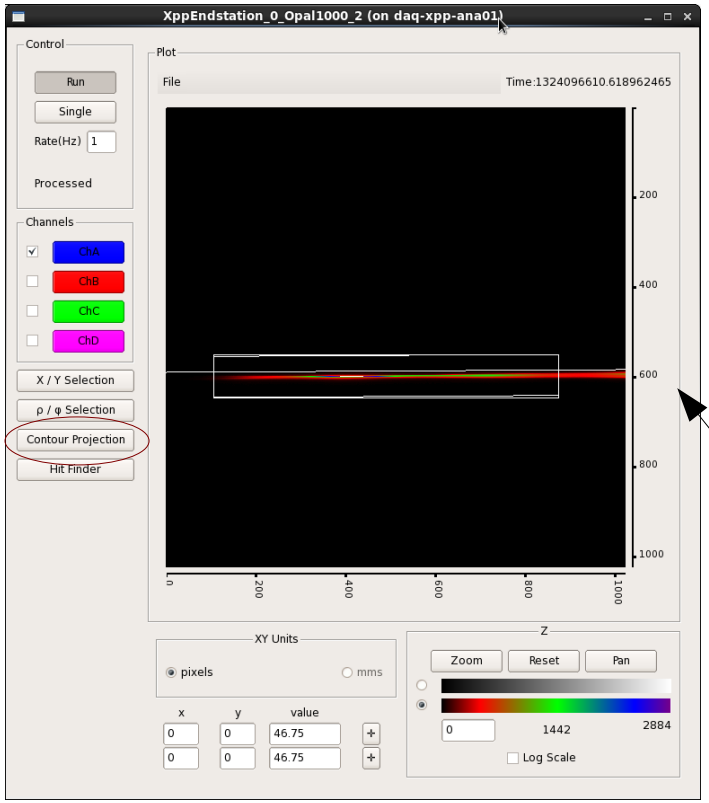
Enter or grab the region of interest

Select the processing on that region

online_ami – image rφ projection

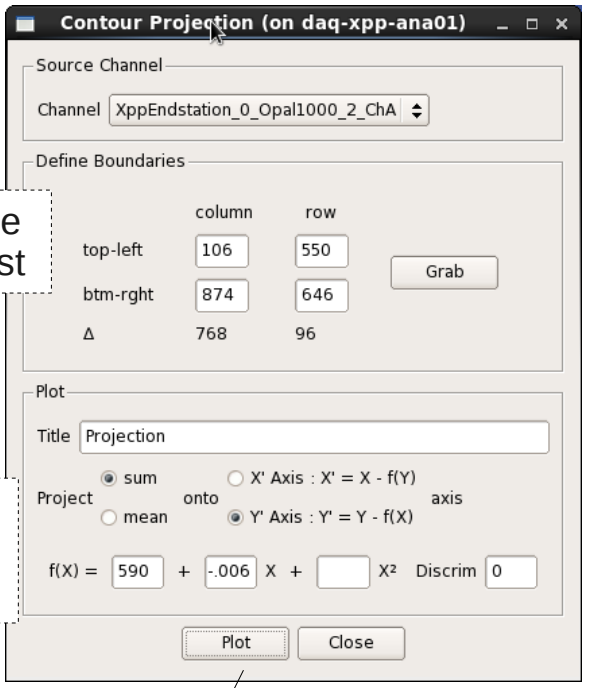


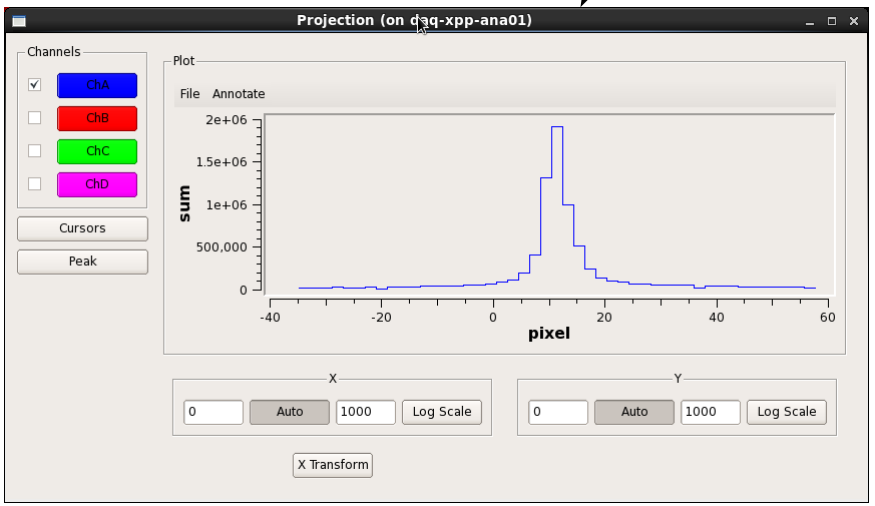
online_ami – image contour projection



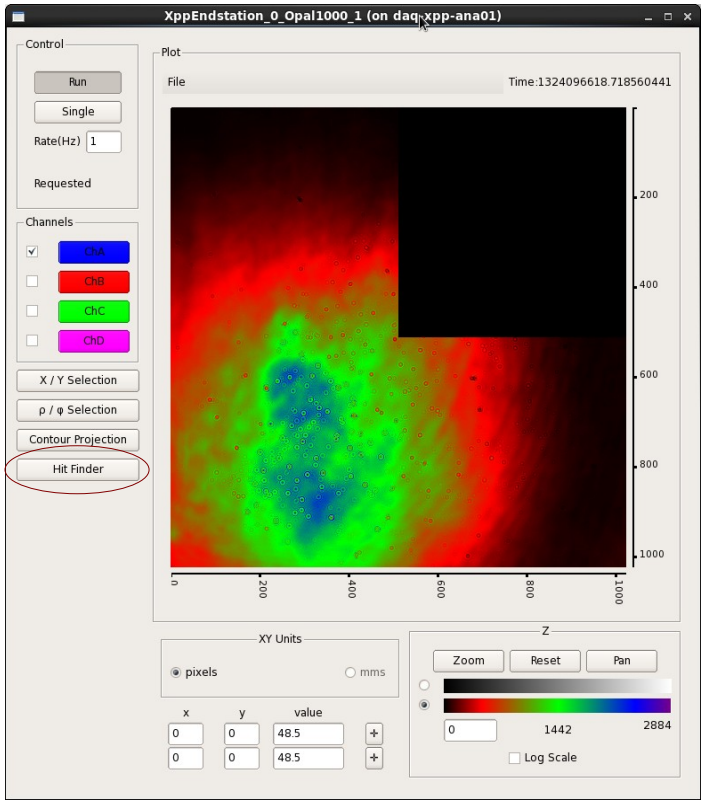
Enter or grab the region of interest

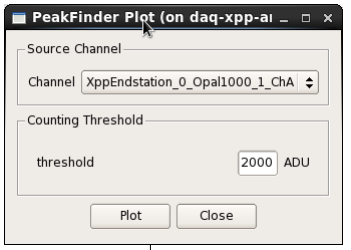
Enter zero-axis function (note white line)



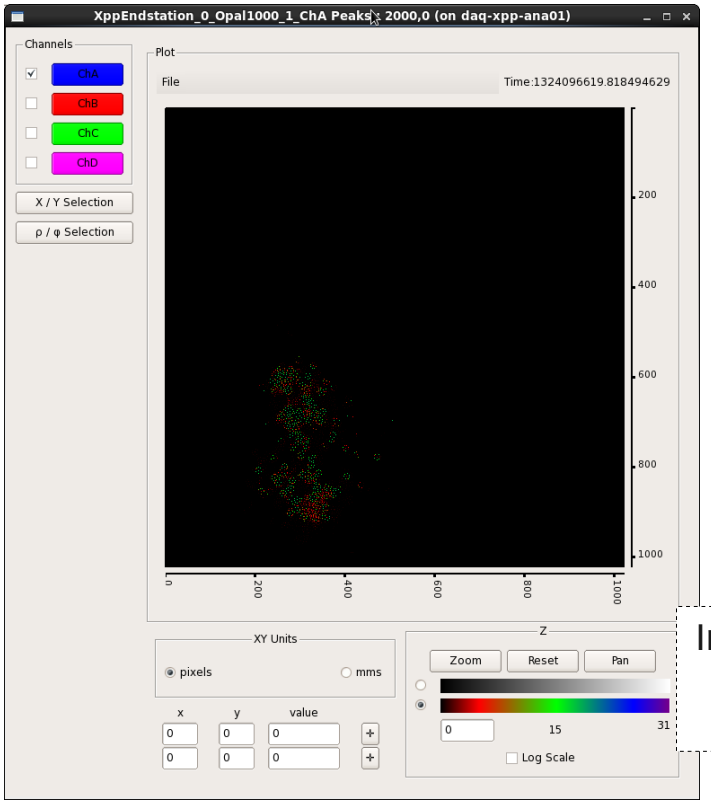


online_ami – image hit finder





Map locations of local maxima above *threshold*



Intensity scale is hits found (cumulative)

ami development

Adding new detectors

Handlers
class event/*Detector*Handler

Instanciated in XtcClient
during *configure* transition

On L1Accept
generates prototype *plot* from
xtc data

Adding new plot types

Plot Types
enum data/DescEntry::Type
{Scalar, TH1F, TH2F, Prof,
Image, Waveform, Scan, Ref}

class data/Entry*Type*

Deserialized by data/EntryFactory

Operations
enum data/AbsOperator::Type
{Single, Average, XYProjection,
EdgeFinder, EntryMath, BinMath, ...}

class data/*Operation*

Deserialized by data/OperatorFactory

Adding new processing

Analysis Inputs
class qt/*Operation*

Online Analysis Design – *ami* plugins

Use dynamicly linked library to extend server functions at runtime
- hutch specific
- reproducible layout
- user controlled code

Plug-in interface includes
- direct access to XTC contents (callbacks for each container)
- global event filter decision
- add scalar variables to global list (“Env”)
- generate plots with a described layout

Difficulties
- too much investment for visitors
- base code evolution

Online Analysis Design – *pyami*

Python library for configuring *ami* server

User scripts replace GUI (manual inputs)

User scripts responsible for plot display

Complementary to *pydaq* – python library for DAQ control

fin