egraph and evalstat

To analyze raw event data of EXED and other experiments, which record time of flight (TOF) data, the tandem programs *evalstat* and *egraph* are provided here at Helmholtz-Zentrum Berlin.

evalstat

- reduces event data to 2- and 3-dimensional spectra (x,y) and (x,y,TOF) intensity
- writes these to a simple binary file format to be used by egraph
- writes raw files to be further processed with Ariel
- writes NeXus files suitable for working with Mantid

egraph

- has the GUI to provide evalstat with parameters
- displays spectra

Raw Data

Event data are expected in the format provided by the Mesydaq data acquisition software (mesytec psd+ system). An experiment comprises one to many data files with a given naming convention. All files have a header of clear text meta data, and binary chunks of event data. The meta data of the first data file, or an explicitly stated description file with the same syntactical structure, contain the experiment setup: chopper positions, detector tube positions and sizes, environment parameters,... Meta data are provided by the experiment control software CARESS at EXED. They comprise
`key = value`
text lines, grouped to sections like
`[CHO_]`
where a key `dev` in that section defines a variable cho_dev.

Binary data are time stamped chopper signals (chopper id, time), and detected neutron events (tube id, position on tube, time), where time values are from a 48 bit resolution clock of 10 MHz, relative to the wall clock time with a second resolution of the experiment start. Time stamped ADC values may be recorded, too.

Time of Flight computation

To identify time of flight (TOF) values in addition to the position data, evalstat

- does a chopper time statistic to obtain the periods of choppers
- computes tof_min, tof_max from this
- stores times of chopper A events which define neutron packet start times
- stores times of a second chopper B with lower frequency to identify the valid chopper A times among those belonging to neutron packets swallowed by the chopper cascade
- subtracts the corresponding chopper A times from neutron event times
- optionally normalizes TOF values to one distance from chopper A to detector pixel

Data selection

evalstat may select events. In TOF mode only those events with a valid TOF are used to increment spectra. You may select parts of the whole experiment time with a second resolution. For example you may read and process ADC values in a first go to identify parts, and increment spectra in a second go, suppressing all unwanted events.

Read and plot

A typical egraph session has two steps:

1. select event data files, and read them to egraph storage
2. select spectra to plot and analyze

Read Selection

Data files are expected to follow naming conventions,

`/hmi/EXED_DATA/2010/08/V15_0000001277_S000_P01.mts`

`/hmi/EXED_DATA` is the data root, the file was generated in August of 2010, and it is the first file or sub run (P01) of run 1277. You may browse files, or just enter the run number and press Return to find the first sub run of a run.
You may then specify which files of the run should be used by adding a specification.

- the keyword `all` specifies to use all sub run numbers
- 1-3,5 would select sub runs 1,2,3,5

You may combine different runs given as a blank spaced list here, too.

You may optionally specify to create Ariel or Mantid output files when reading raw event data. These files are written to a user specific output directory, and resemble the name of input files. For user `guest` this would for example be the NeXus file `V15_0000001277_S000_P01.nxs` in the directory `/hmi/guest/RAW/2010`

evalstat knows many parameters which control spectrum integration.

Print Plots

Plots may be printed to available printers, especially printed to PDF files.

You may combine more than one plot to output pages, either to save paper or visits to the printer, or to combine them in a PDF file. To do so you first store plots (buttons “Store plot” and “Print+”) and later print stored plots, where you may combine 1, 2, 4, or 8 plots to a DIN A4 page.

Plot Selection

You may select various kinds of spectrum plots, just select them from a list of available plots.

1d spectra:

- Amplitudes per tube
- Amplitude statistic
- Monitor
  shows the monitor count over the whole experiment
- Computed Monitor
  show the total count of accepted neutron events
- TOF of selected tubes
- TOF spectra of pixels
  where e.g. 48 tubes with 100 pixel each give 4800 individual pixel
- ADC
  shows measured environment values

1d spectra may be saved to text files with comma separated rows of x, intensity columns.

2d spectra map intensities to colors from a palette of 256 values

- Position statistic
  just x,y intensities
- TOF
  shows a rectangle with TOF-channels * pixels color coded values
- Integrated TOF range
  like position statistic, but with events in a given [tof_min, tof_max] interval

Comparing Spectra

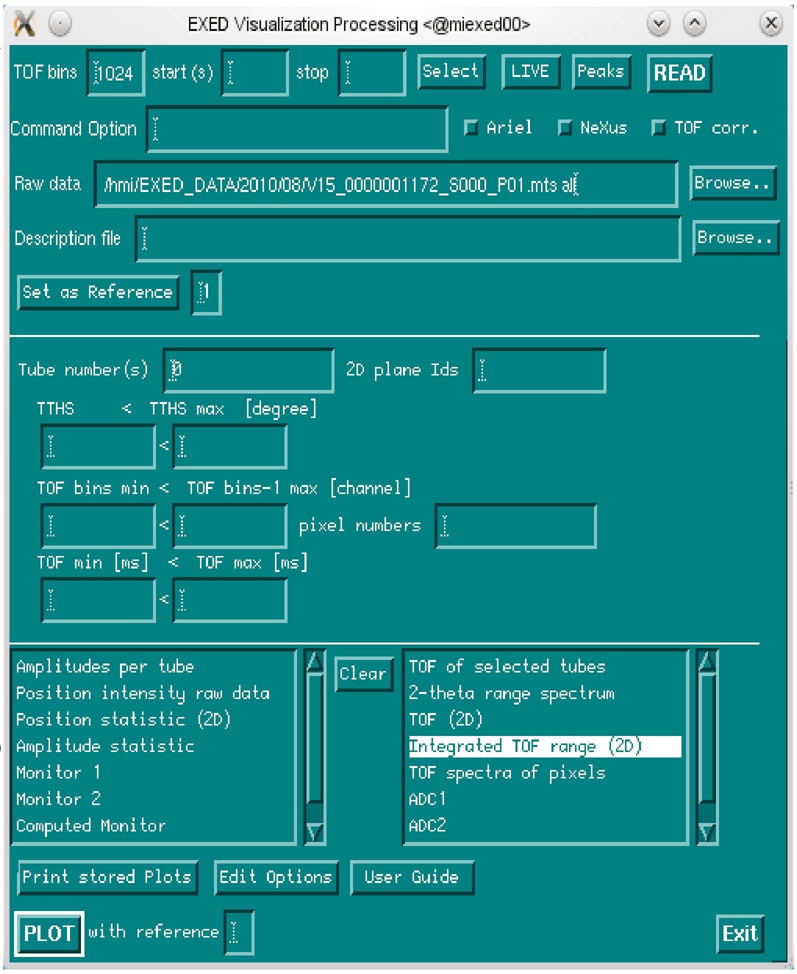
1d spectra of different runs or parameter setups may be compared by first saving a reference data set in an egraph session, then read a new data set, and specify to plot spectra with over-plots of reference data.

egraph Options

Special options are set and stored per user to a ~/.egraph file, and may be edited with the GUI.

datadir	root of raw data files, default /hmi/EXED_DATA/
outdir	if given, output files go there
printer	the preferred printer to use
printer_avail	white space separated list of available printer, especially PDF
logcolor	if 0 (default) colors are mapped linearly, 1:logarithmically, 2:exponentially
nexus_compr	if 0 (default) NeXus output files are gzip compressed, 1: run length compressed

Main GUI



After starting egraph you see the main graphical user interface (GUI).

In the example to the right the experiment run 1172 has been selected by typing “1172” in the “Raw data” text field, followed by pressing the Return key to expand the full file name, and adding “all” to use all sub runs of run 1172.

These data have been assumed to be correctly documented in the header of the mts file, so no “Description file” has been specified to replace the information in the raw data file.

No special command options had been given to evalstat in the “Commands Option” text field, and no Ariel or NeXus output file have been created when clicking the READ button to read raw data here.

You will see a clock cursor while evalstat reads raw data.

Then “Integrated TOF range (2D)” was selected, and PLOT clicked. You could select more plots to your needs.

Clicking Clear closes all plots and deselects all list items.

In the middle area you may specify the plot (and read) options. Where ever you may enter more than one value, you can give a comma separated lists of ranges, like 1-2,5-7 to select 1,2,5,6,7

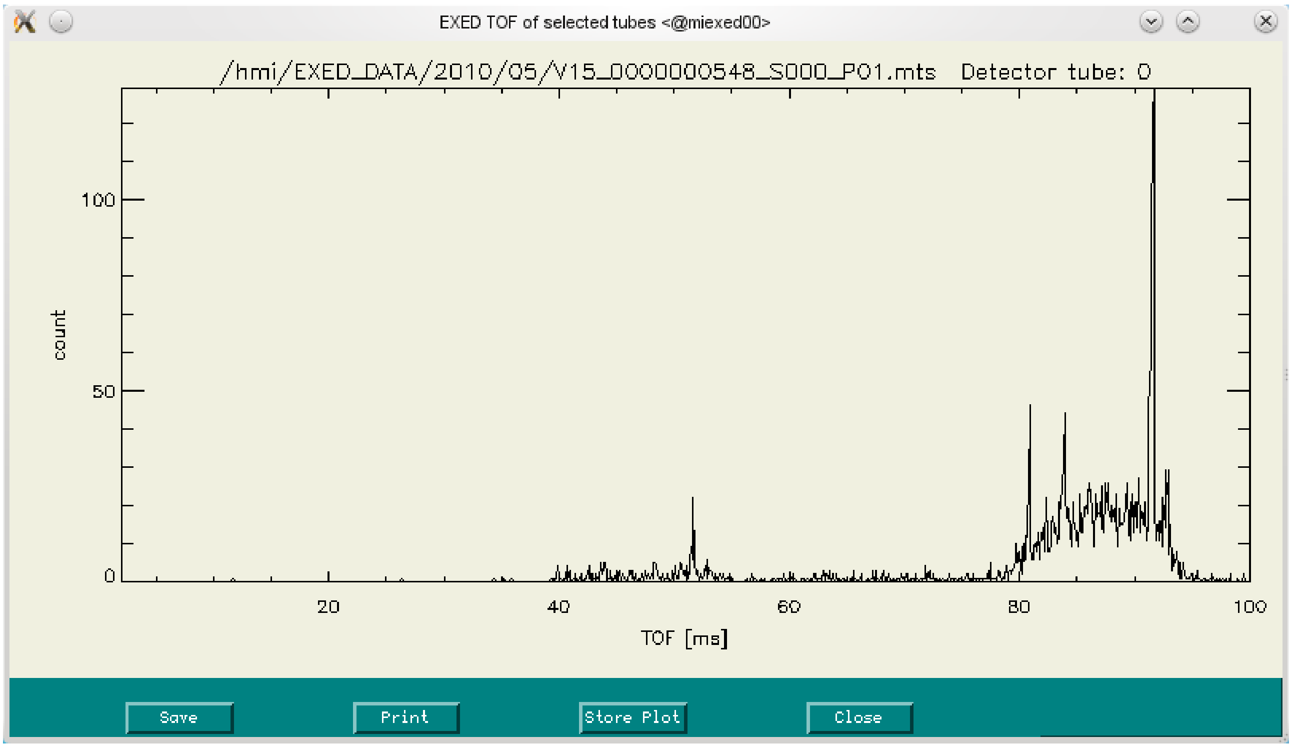
Tube number(s)	select tubes to be displayed in plot operations, e.g. for the “TOF of selected tubes” or “Position intensity raw data” plot
2D plane Ids	reduce detector tubes to those tube numbers given when reading raw data; useful if you want to concentrate on one planar detector, often tubes 0-47 of the big planar detector, or need to suppress tubes with faulty data
TOF bins	select TOF channels by number, numbers in the range 0-1023
TOF min	select TOF channels by time of flight in milliseconds used for “Integrated TOF range (2D)” plots
pixel numbers	select a list of pixels, where tube 0 has pixels 0..99, tube 13 has 1300..1399 if 100 pixels had been specified for tubes, which is the default

If you click “Set as Reference” after reading some data, this data set is stored internally as a reference with the number given. You may store references 1 to 4.

If a reference had been defined before, and if you enter a number in the “with reference” field, some 1D plots will show the reference data, in addition to the actual data set.

Hint: Normally the intermediate out files from evalstat are deleted after egraph read them. You may keep them if you specify “-O result.out” in the “Command Option” text field, and may read the file result.out later again, if you specify this file in “Raw data”. This may save time when dealing with big data sets, as evalstat may process 70 MB/s, summing up to a minute processing time if your experiment comprises 4 GB of raw data.

1D Plots



One-dimensional spectra like this TOF intensity count spectrum for tube 0 share some properties:

- You may zoom the x-y range to see details; click the left mouse button and drag a rubber band rectangle to include the range of interest, release the left mouse button. If you want to have back the full range, click the right mouse button over the plot window.
- Clicking the “Print” button will print this plot to the default printer.
- Clicking on “Store Plot” will add this plot to the plot storage.
- Click on “Save” to write a text file with two columns of data, first column is the x axis with float numbers for TOF values like in the example, second column is the integer intensity. Monitor plots use the Unix integer time format for x values, that is seconds of the epoch, which is seconds since 1.1.1970 now. Files are written to outdir specified with options, or to the directory specified in the environment variable RESULTDIR, and as last resort if not given otherwise to the home directory.

Show 2D

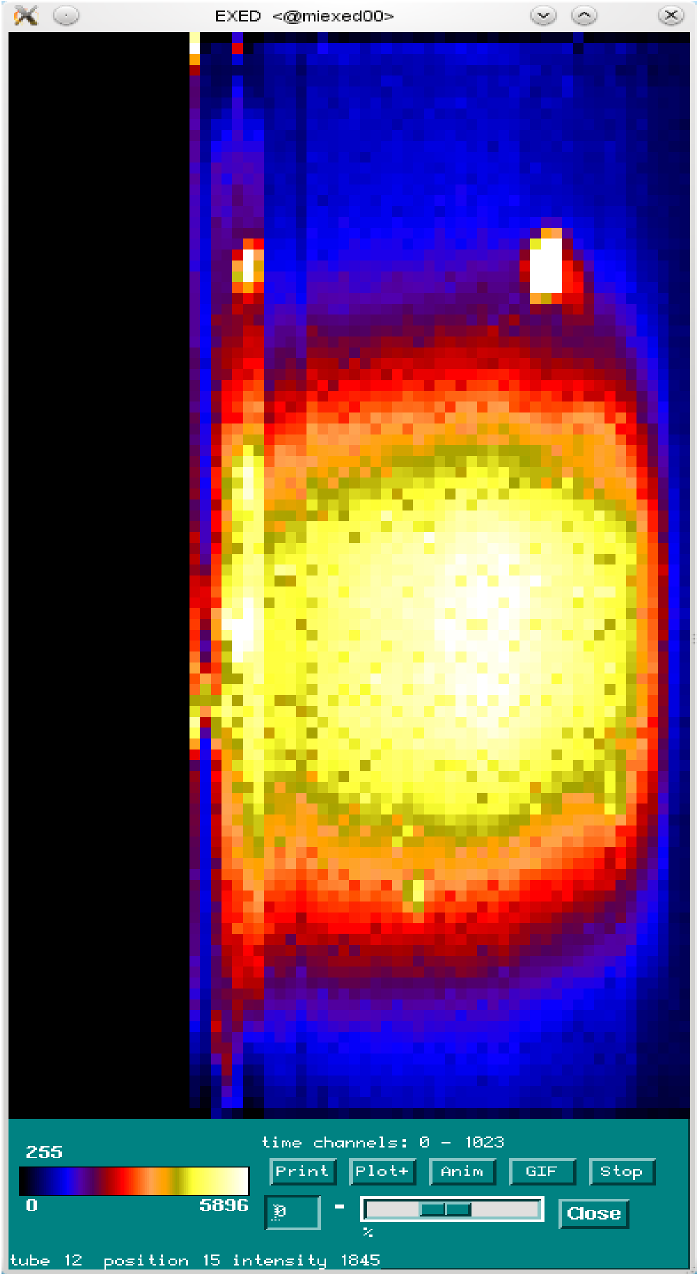
In this example you see a histogram of TOF intensities of vertical tubes, color coded with 256 colors. The intensity range has been reduced to 0 – 5096, you could change the maximum value 5096 by moving the slider, or increase the minimum value by entering something other than 0.

The annotation at the very bottom of the window shows where the cursor had been moved over the display, and what intensity had been detected at the corresponding tube and position.

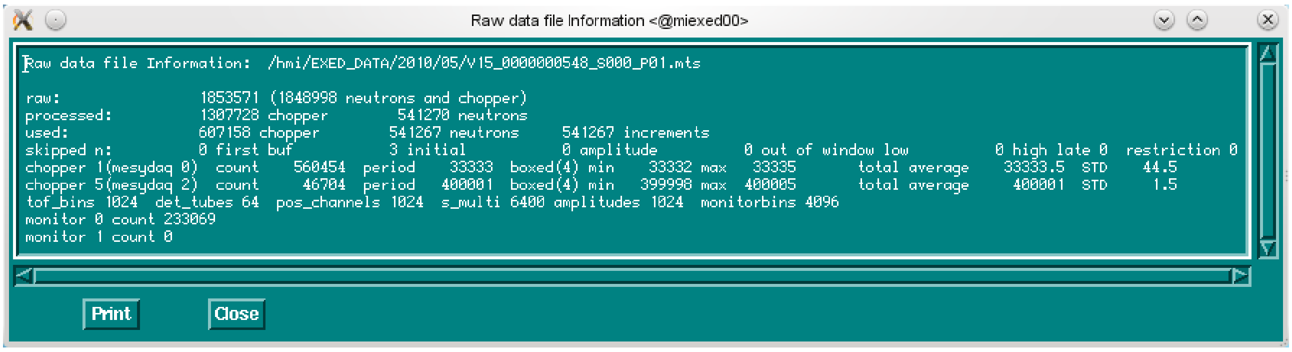
The comment “time channels: 0 – 1023” indicates that all TOF channels had been integrated per detector pixel. You could restrict this in terms of time bins or ms in the main GUI.

The “Anim” button starts an animation of the time of flight in the given range, and displays e.g. growing Debye-Scherrer rings for reflection of powder probes. You may stop a animation, or produce a GIF file of the animation.

“Plot+” means to store this plot to be printed later.



The Information Window



After reading data, evalstat outputs some overall values of interest. These number are shown in the information window.

Albeit you will ignore most of the numbers here, they give important information on the correctness of the data and especially meta data.

- If you compare the number of total chopper and neutron events to those which are skipped or used for increments in the end, you may detect errors, when e.g. chopper definitions are poor.
- Here you see the full overall statistic of chopper A and chopper B, and that they had stable periods.
- You may request even more numbers, if you use the evalstat option --fullStatistic 1
- If you give the command option -v you will see more verbose output, especially the time correction for chopper times, and the identified chopper time cascade. This option helps to identify TOF-related problems.

Live Display

egraph may watch a running experiment, and look for new data for a given run in a constant interval (5 s). In that mode evalstat runs in daemon mode watching input files, and changes the output files if new data are found. egraph watches the binary out-file, re-displaying plots if anything changed.

tisane Mode

In tisane mode the probe is stimulated by an alternating external field with relative stable period. The external field is monitored, e.g. by the monitor 2. First you determine the field's period in Hz like

/opt/eval/tisane_mon.pl /hmi/EXED_DATA/2010/09/V15_0000001407_S000_P01.mts -v

and hand the option `--tisane` frequency

to evalstat then. Events are binned to channels time modulo period then; chopper times are ignored.

Select Mode

Some experiments observe neutron scattering relative to environment parameters like temperature or magnetic field. Because scattering events are recorded with absolute time values alongside with analog values recorded as ADC values, it is possible to select scattering events matching special patterns of ADC values.

To do so you press the “Select” button, and follow 3 steps in the “Select Time Segments of the Experiment” window.

1. extract ADC values to a special binary file to be used later
2. create a time selection file, usually by formula
3. read sources under selection

Because the select mode is of interest with high event rates, and measurement times of weeks, steps 1 and 3 may last longer. In that case you may use the “nowait” buttons, where files are prepared in the background.

The time selection file has as many bytes as the experiment had seconds, and, guess what, a 0 byte means to skip a second, a 1 byte selects this second as valid.

If you plot the ADC values, you may specify valid seconds by a formula, and examine in the plot, which seconds you specified by this.

To know what happens in detail, you may restrict the time plot to a time range (s)

In the formulas

`a1` denotes ADC 1 values in the range 0 to 100

`a2` is for ADC 2

`secs` the seconds of the experiment.

De facto a1,a2, and secs are array variables, processed by PV-Wave. In the formula you may use

arithmetic operators	`+ - * / ^`
PV-Wave functions	like `SQRT SIN EXP ALOG`
comparison operators	`LT LE EQ NE GE GT`
Boolean operators	`AND OR NOT`
parentheses	`( )`

Names and operators are case-insensitive, stating “a1 lt a2” does the same as “A1 lT a2”.
A special function `stable` helps to identify times where values are stable.

stable(a1,10)	seconds, where a1 is constant over a range of at least 10 consecutive values
stable(a1,il=100, delta=1)	ranges, where consecutive values are +- 1; additionally: take only the first 100 seconds or these ranges

The parameter delta has a default of 1.e-5, il is 1000000L (L for integer*4).

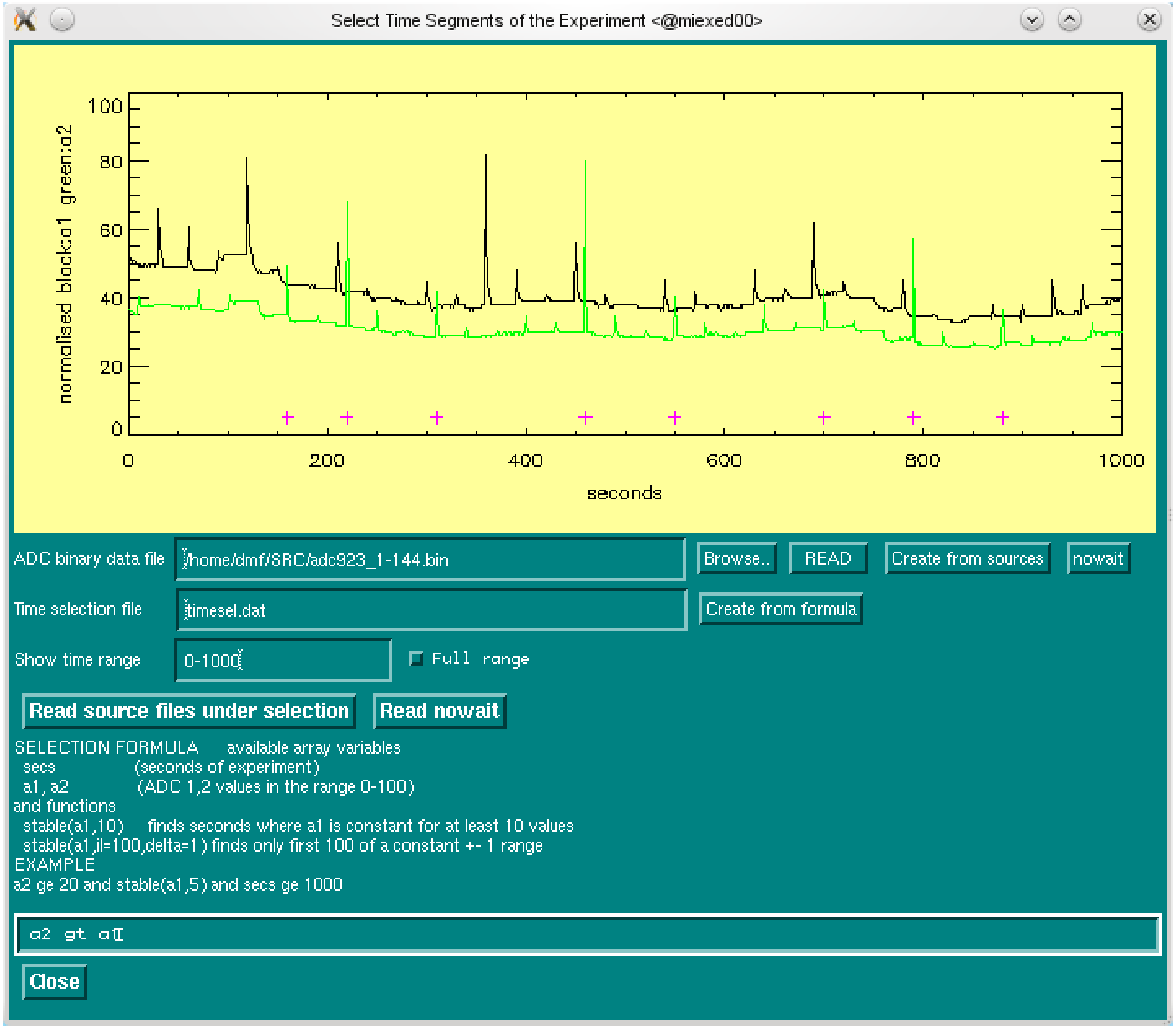
Examples:

a1 gt 0 and a2 gt 0

select only seconds, where both analog inputs were greater 0

a1^2 ge a2^2

restrict to times, where the squared ADC 1 value is at least ADC 2*ADC 2



In this **example** an ADC binary file had been created and read.

Because it was a long lasting experiment, we chose to have a closer look to the first 1000 seconds, where we see values of ADC1 in black, and those of ADC 2 in green.

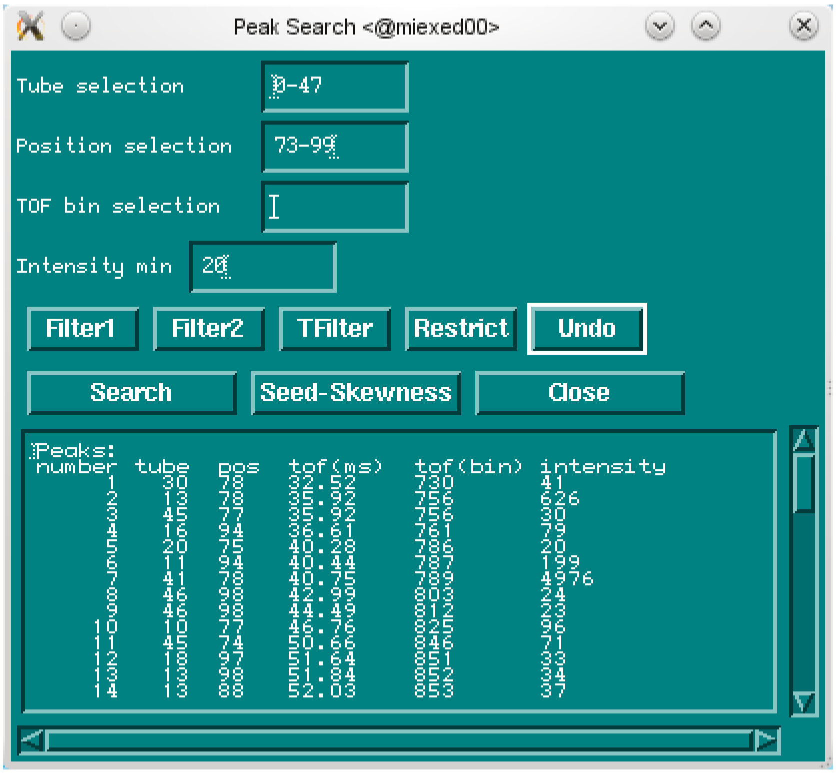
To show the capabilities here, we entered the formula “a2 gt a1” and pressed the Return key (could also click “Create from formula”). As a result egraph showed, with violet cross markers, where this equation holds.

We did not specify a “Time selection file” name, egraph provided the default timesel.dat.

If we would click “Read source files under selection” then – assuming the raw data file range has bee entered in the main GUI – egraph would start a long lasting run of evalstat in the foreground, where egraph just shows the clock icon, unable to serve GUI events. In the case of big data sets “Read nowait” is the preferred method, where egraph starts a background job and tells where to find the output file later.

Peak Search

You may search local maxima in the 3 dimensional x,y,TOF space.



It helps to reduce the space by tube, position, or TOF bin selection.

You may filter data before searching peaks.

“Restrict” just restricts data to the cube specified. All filters first restrict and then somewhat improve data to find peaks.

You may observe the result of filter operations in the show 2D plot window, and if the filter makes no sense, restore the data you had before with “Undo”.

TFilter does a one-dimensional smoothing of time data for individual pixels.

Filter 1 is a one-dimensional, Filter 2 is a two-dimensional Markov chain filter (Multidimensional peak searching algorithm for low-statistics nuclear spectra by Miroslav Morhac DOI 10.1016/j.nima.207.08.145)

“Search” starts a heuristic peak search which lists its findings like here in the example.

“Seed-Skewness” starts a batch job to find peaks with the HIPPO algorithm, adopted here to fortran 90 from sources provided by J. Peters (HIPPO_V1: Adaptation for instrument V1 at HMI, Berlin)

evalstat options

evalstat does the donkey work and may be called from the command line. Just call

```
/opt/eval/evalstat -h
```

to see what parameters are possible for a given version. In general it expects one or more data files, simple -X options, and more --X options.

```
evalstat {datafile} [-t] [-option value] [--option value]
```

either specify one or more data files, or a "from,to" time range

either specify a description file, or the first data file must contain the description.

evalstat does an initial chopper statistic to obtain chopper periods, used to get correction offsets which depend on periods, then derives a time cascade of chopper opening times, which reflects the lambda setup of the chopper control system which is not available otherwise. From this evalstat computes the expected time offset between chopper A and B. The initial chopper time statistic must be possible with the first data file.

In the following TOF computations evalstat just records corrected chopper A and B times, where chopper B times, together with the initially computed time offset, indicate which chopper A times are to be used as start times for neutron flight times.

Simple Options

-t	test description only, then bail out
-a adcfile	do only statistics on ADC values
-b bins	channel bins for TOF spectrum, default 1024
-B	compute the beam center and nothing more
-d descr	description file with experiment settings, default none means take first data file
-D datdir	directory for experiment data, default none
-e end	end date, default today
	date: Unix seconds or
	DD.MM.YYYY:hh:mm:ss
-f start	start date, Unix seconds or date string, default today
-g filename	save Ariel data to filename, default <sourcefilename>.raw give /dev/null to write none
-o specfile	save a text result with TOF, intensity columns to specfile
-O filename	save a binary output file used by egraph
-m monpixels	compute monitor count rate for detector pixels
	specify ranges like 0-1599 or 50-60:150-160, or all
-M monbins	number of monitor bins per monitor, default 4096
-n filename	save result to NeXus file, default none

-p seconds	poll interval seconds, run evalstat as background daemon specify a datafile, and the -O or -g option also
-r restrfile	use only those seconds of the experiment which are valid from restrfile
-S match	source file expression, like V_+.mts:1:3 to be expanded to V_1.mts V_3.mts

Option+Value Pairs

In these pairs b is a Boolean value (0 or 1), n is an integer value, and v a float value.

--noTOF b	if set to 1, just count events and do no time of flight computation
--tisane v	see Tisane Mode
--startSecond n	skip first n seconds of measurement
--stopSecond n	stop after second n has been processed
--skipSeconds list	omit seconds, specify e.g. 10-20,27,1000-10000 Hint: no white space in list!
--intervalSeconds n	run evalstat in daemon mode, check each n seconds for new data
--fullStatistic b	report a more detailed statistic on chopper times if set to 1
--finalWaitTime n	daemon mode: if no new data are found for n (default 60) s, then terminate
--incMask mask	a restriction mask for spectrum increments default is all set: table_det_posM=1, tof_countM=2, table_yM=4, table_y_detM=8
--noGap b	do not reset chopper and neutron storage buffers with each file, default 0
--nxRLE v	compression of NeXus output files, default 0:gzip 1:RLE
--Monitor n	end if n monitor events have been processed, default process all events
--amplitudeBins n	number of bins for amplitude statistic, default 1024
--minAmplitude v	minimal value to be mapped to 0
--maxAmplitude v	maximal value to be mapped to amplitudeBins-1
--amplitudesPerTube b	if set to 0, no amplitude per tube statistics are done
--sansMode 1	simpler initial statistics, strict usage of description parameters
--sansFile filename	augment a sans file, used by v2bersans.pl
--sansYScale v	scale factor used to rebin positions for sans files, default is 0.108833772
--sansScreen v	if > 0 it's a radius of pixels around the beam center to be ignored

--triggerDt v	time period after external trigger events, in clock ticks
--triggerDtN n	time bins after external trigger events
--triggerId n	id of an external trigger
--tofRanges 1	compute individual TOF ranges for all detector and monitor pixels, where the length of the tmin, tmax ranges is kept, but the middle value (tmin+tmax)/2 is corrected to reflect the real distance from the first chopper to the pixel.

	With this option the overall range of the time axis will increase, as some pixels are nearer, and some are farther away than the reference pixel.
refPixDist distance	by default the reference pixel is computed to be the distance of the detector pixel which is nearest to the probe.
--tofCorrection b	0 (default) or 1 to do TOF correction
--sPosChans n	number of positions per detector tube, default n=100
--skipFirstNeutrons b	evalstat normally exits with a warning message, if internal buffers for neutron times are insufficient; if you give 1 here, some neutron events in the beginning may be lost silently
--allChopperTimes b	if set to 1, only chopper A times are important, and all define valid packet start times; normally chopper B times are used to ennoble the valid chopper A times

Environment Variables

evalstat respects some environment variables which control behavior specific to the experiment, by now for EXED or VSANS.

USER	controls where to put output data, like /hmi/guest for user guest
EXPERIMENTNAME	used in output strings
HOST	if this contains “sans”, automatically –sansMode 1 is set

Helper Applications

dumper filename

does a text dump of mts raw data files. If dumper fails to read a file, something was wrong with the mesydaq software or setup.
Albeit slow, dumper helps to produce special statistics, and is used in Perl scripts like

tisane_mon.pl filename

does a monitor 2 statistic

dump2ascii.pl

bariel.pl batch processing to produce series of ARIEL files, see bariel.pl -h

makeCorrTable.pl

is a script to produce a binary correction file for detector data from a calibration measurement

make_description.pl

generates a XML description file of the experiment from the description read in data headers; used by egraph

patchMesy.pl patches some mts raw data files with a wrong offset in the

DATA=offset

line, assuming binary data just behind that line guillotine.pl dumps the text header of mts raw data files

make_vsans_description.pl

make_48_back_description.pl

is a template script to generate detector descriptions for planar detectors, individual tube data are computed from few geometry data

perm.pl

chopstat.pl do chopper statistics on mts files, see chopstat.pl -h

hasTOF.pl checks if mts raw files contain chopper events

parseTextToSGML.pl

generates a XML file from the description read in data headers;

used by egraph to help Mantid export

fixps.pl used by egraph to fix errors in postscript output files

hipcall.pl used by egraph to batch process peak searches

hip_exed_s is an adapted HIPPO f90 program to search 2d peaks, used by egraph