A short guide to the use of the event_file tool in Lamp

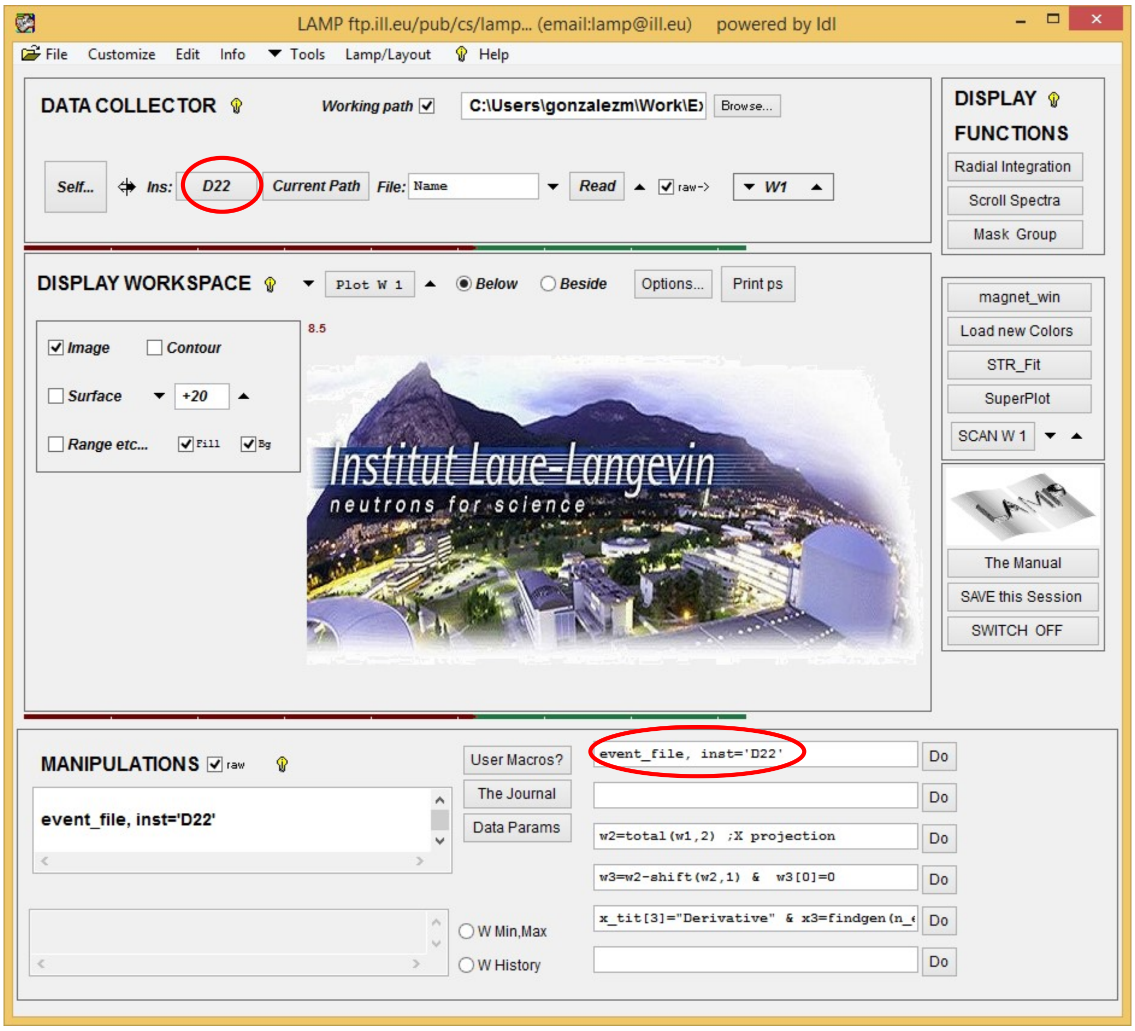
Miguel A. Gonzalez (gonzalezm@ill.fr)

ILL, 1st December 2018 (beta version)

1. Starting

Launch Lamp and use the command *event_file, inst='D22'* in any of the Do boxes (see Fig. 1) to launch the interface window to read and manipulate event files. At present, only data measured on D22 with the new format (used since cycle 183) can be read. List mode data recorded in D11, D22, and Salsa using the old format can be still read using *event_file_old, inst=INST*, where *INST* can be either *'D11', 'D22'* or *'Salsa'.* If the instrument name is not given, the program will assume that it is the same instrument as the one selected on the main Lamp window (Fig. 1).

The current version is still a beta version, only minimally tested. Please send any comments, suggestions or bug reports to gonzalezm@ill.fr.



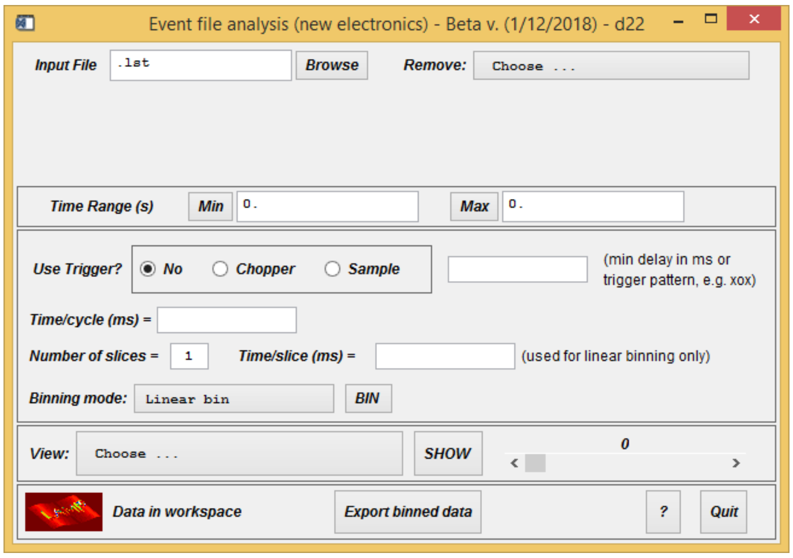
```
event_file, inst='D22'
```

```
event_file, inst='D22'
w2=total(w1,2) ;X projection
w3=w2-shift(w2,1) & w3[0]=0
x_tit[3]="Derivative" & x3=findgen(n_(
```

Fig. 1: Lamp interface and command to start the event file interface.

2. Event file analysis interface

Fig. 2 shows the empty interface that appears after calling *event_file*. Use the browse button or give a list mode file in the corresponding box (if the files can be found in the current working path) to select a single numor. All the recorded events are read and stored in internal arrays for further treatment (see appendix 1).


Event file analysis (new electronics) - Beta v. (1/12/2018) - d22

Input File: .lst

Remove: Choose ...

Time Range (s): Min 0. Max 0.

Use Trigger? (● No ○ Chopper ○ Sample)

(min delay in ms or trigger pattern, e.g. xox)

Time/cycle (ms) =

Number of slices = 1

Time/slice (ms) = (used for linear binning only)

Binning mode: Linear bin

View: Choose ...

0

Fig. 2: Interface of the event file manipulation tool.

After reading a file, the interface will be populated as shown in Fig. 3. The first 3 lines contain information about the content of the file and correspond to the number of events actually stored in the different internal arrays (appendix 1). The first number gives the total number of events registered during the acquisition and it will not change until a new file is read. Initially, the number of detector events given in the 2nd line should be the same, but this second number can change if a trigger is selected or some events removed. The 4th line gives some information about the last operation performed. After reading a file, the useful *time range* (in seconds) is determined by the times of the first and last events, t_{min} and t_{max}. The minimum and maximum values of the time range can be changed by the user, in order to use only part of the recorded data. In this case, the number of detector events having times inside the new limits is calculated and shown in the 4th line of the message window.

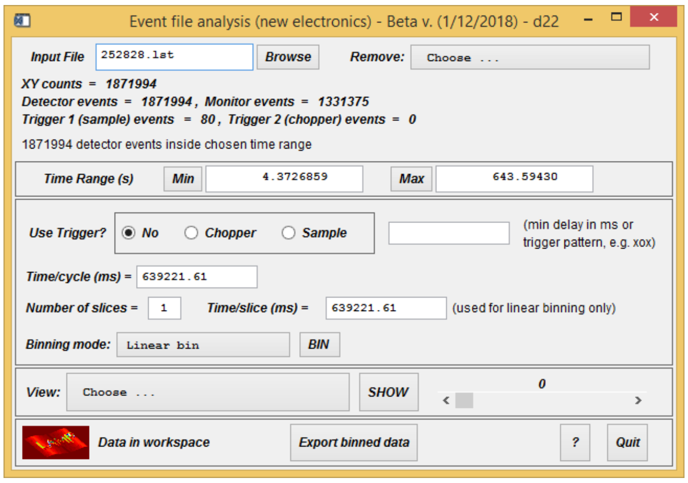
Useful: The time values of the first and last event are kept in memory, so if after selecting a time range you want to come back to the initial values you can find them by clicking in the Min or Max buttons!

The next row of the interface allows selecting a trigger option. The default option is not to use any trigger, so the binning of the data is done using the absolute times recorded in the file. In this case, there is a single cycle of duration $t_{max} – t_{min}$, and this value (in milliseconds) is shown in the *Time/cycle* box. The default *number of slices* is set to 1, and accordingly, the *time/slice* is initially equal to the cycle time.

The default *binning mode* is set to linear binning so that all bins have the same time duration. However, two alternative options are available: log bins and variable time bins. In the latter case, each bin has a different time duration, such that the number of detector events in each bin is approximately the same.

The *View* row allows selecting what information to plot, as explained in the following section. Note that the data to be plotted are sent to a Lamp workspace, so you need to check the Lamp main interface to see the plot. You can also use Superplot (Lamp: Tools → Overplotting) to look at the data.

The final row is used to provide information about the workspace used to store the last data sent to plot and allows to *export the binned data*.



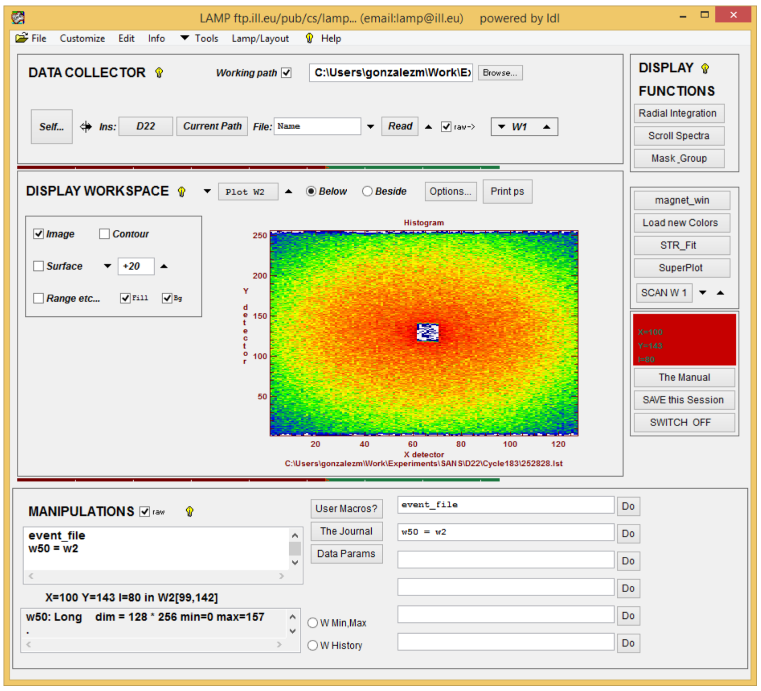
Event file analysis (new electronics) - Beta v. (1/12/2018) - d22
Browse
Remove:
SHOW
?
Quit

```
Input File  252828.lst
XY counts = 1871994
Detector events = 1871994 , Monitor events = 1331375
Trigger 1 (sample) events = 80 , Trigger 2 (chopper) events = 0
1871994 detector events inside chosen time range
Time Range (s)  Min 4.3726859   Max 643.59430
Use Trigger? (● No  ○ Chopper  ○ Sample)   (min delay in ms or trigger pattern, e.g. xox)
Time/cycle (ms) = 639221.61
Number of slices = 1   Time/slice (ms) = 639221.61   (used for linear binning only)
Binning mode: Linear bin   BIN
View: Choose ...
Data in workspace   Export binned data
```

Fig. 3: Event file interface after reading a list mode file.

3. Checking the data

After reading a list mode file (Fig. 3), you can immediately select the option *View: Total 2D (x,y)*. This will show in Lamp's workspace 2 the 2D image corresponding to the total number of counts detected during the selected time range (see Fig. 4).



```
Working path ✓  C:\Users\gonzalezm\Work\E
Ins: D22   Current Path   File: Name   ✓ raw->   W1
Plot W2   ● Below  ○ Beside
✓ Image  ☐ Contour  ☐ Surface  +20  ☐ Range etc...  ✓ Fill  ✓ Bg
C:\Users\gonzalezm\Work\Experiments\SANS\D22\Cycle183\252828.lst
X=100 Y=143 I=80
MANIPULATIONS ✓ raw
event_file
w50 = w2
X=100 Y=143 I=80 in W2[99,142]
w50: Long   dim = 128 * 256 min=0 max=157
○ W Min,Max
○ W History
```

Fig. 4: Main Lamp interface showing the 2D image corresponding to the total number of counts in the detector (stored in workspace W2).

Care: If the show option is called before any binning has been done, the program will perform the binning automatically using the current settings and then will send the output data to Lamp. But if a binning has already been done, the last set of data in memory is plotted, without doing a new rebinning. So if you change some settings, perform the binning explicitly with the *BIN* button before using *SHOW*.

The option *View: Time vs event number* displays the time in seconds of each event (Fig. 5). This can be useful to check if the counting rate is constant along the whole run or if there are variations, as well as to observe if there are any "bad" events having an unphysical time (either too large or too small). Specific details can be displayed either by using the mouse to enlarge the image shown in the main Lamp window or by plotting the workspace using Superplot.

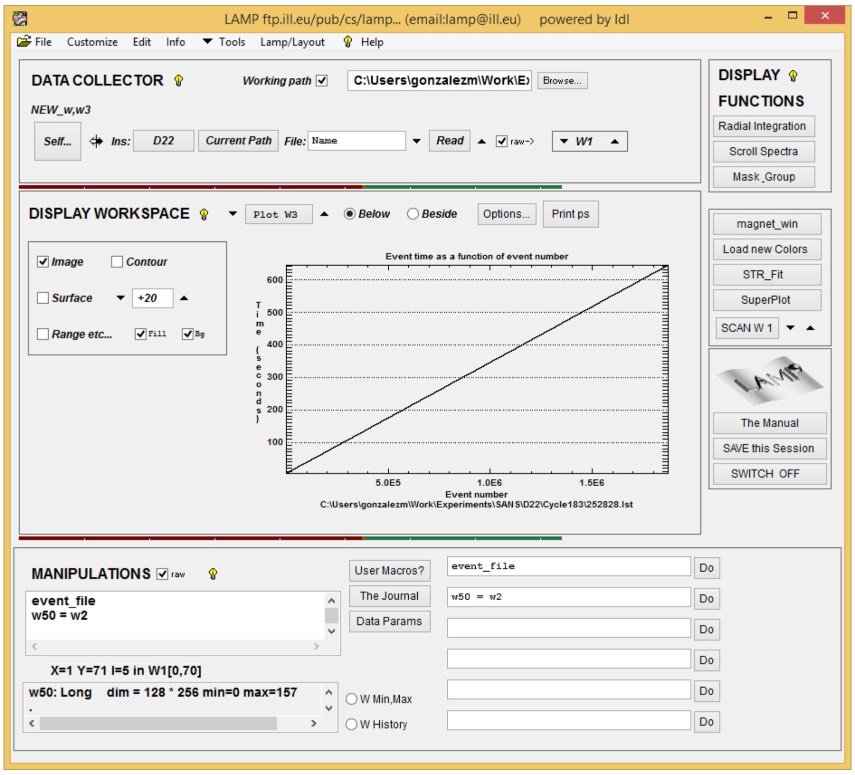
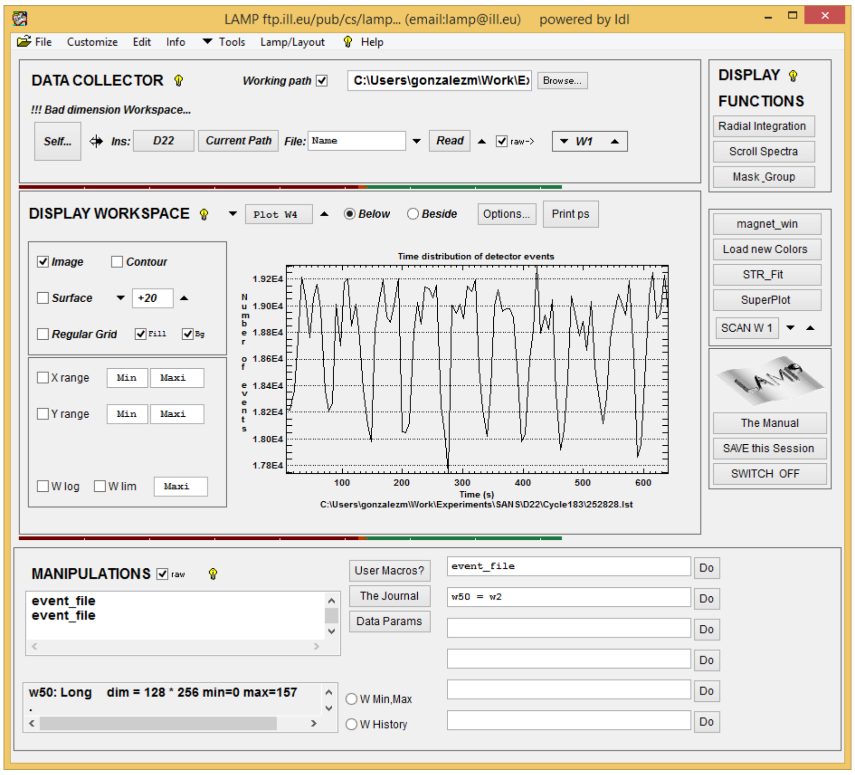


Fig. 5: Main Lamp interface showing the time corresponding to each detector event (workspace W3).

The option *View: Time distribution of events* shows the complementary view, displaying the number of events registered as a function of time (Fig. 6). The selected time range is divided into 100 bins of equal duration and the number of counts registered in each of the periods is calculated and shown.

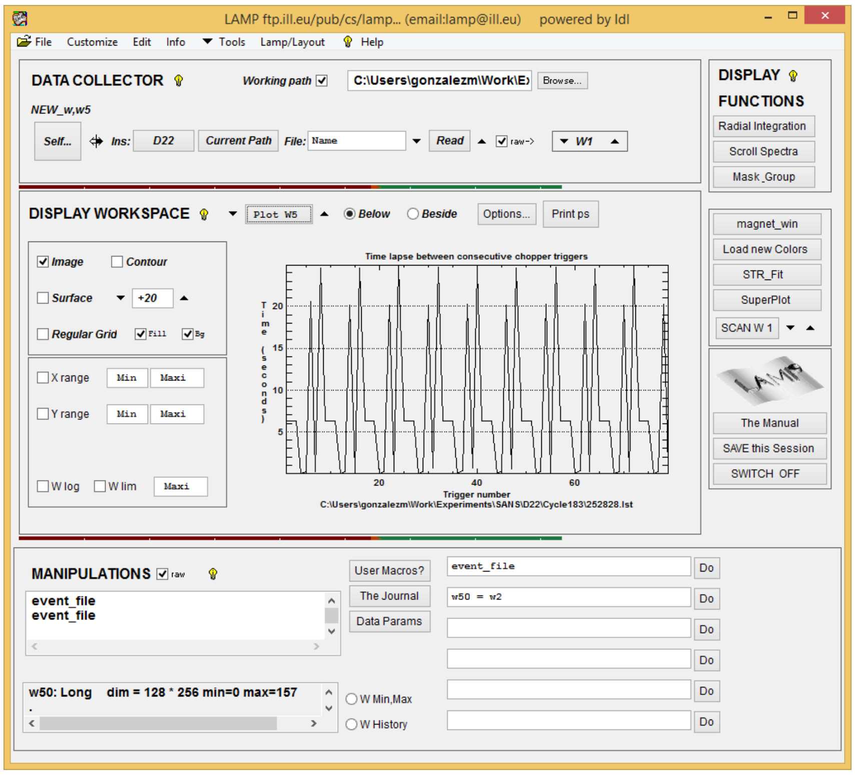
The next three options will only work if one of the trigger options (chopper or sample) has been selected. The first one, *View: Trigger period*, displays the lapse of time between consecutive triggers (Fig. 7), while *View: Time distribution of trigger periods* shows the distribution of lapses between triggers. In this case, the maximum lapse is divided into 100 bins of equal duration and a histogram with the number of trigger periods falling into each bin calculated (Fig. 8). Finally *View: Trigger history* will create a workspace where the x-axis contains the absolute times of each trigger. Then, Superplot can be used to plot the trigger history, as shown in Fig. 9.

If needed, some triggers can be removed in order to keep only those which are relevant for the desired binning (see section on removing bad events and triggers).



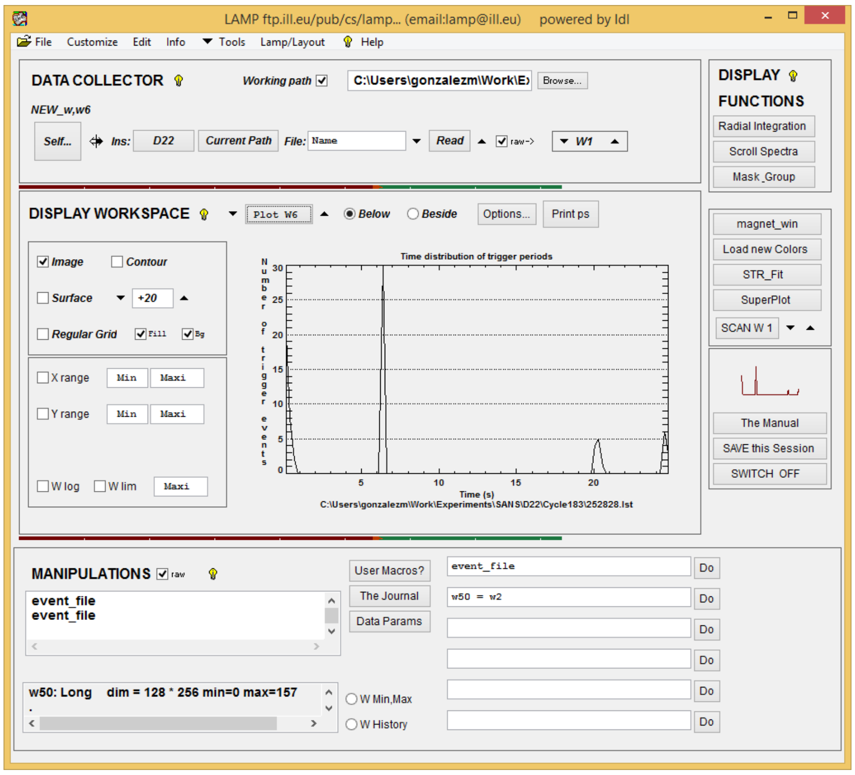
```
C:\Users\gonzalezm\Work\E
!!! Bad dimension Workspace...
C:\Users\gonzalezm\Work\Experiments\SANS\D22\Cycle183\252828.lst
event_file
event_file
event_file
w50 = w2
w50: Long   dim = 128 * 256 min=0 max=157
```

Fig. 6: Main Lamp interface showing the number of events registered as a function of time (workspace W4).



```
C:\Users\gonzalezm\Work\E
NEW_w,w5
C:\Users\gonzalezm\Work\Experiments\SANS\D22\Cycle183\252828.lst
event_file
event_file
event_file
w50 = w2
w50: Long   dim = 128 * 256 min=0 max=157
```

Fig. 7: Main Lamp interface showing the lapse of time between consecutive triggers (workspace W5).



```
event_file
event_file
```

```
w50: Long   dim = 128 * 256 min=0 max=157
```

```
event_file
w50 = w2
```

Fig. 8: Main Lamp interface showing the distribution of trigger times (workspace W6).

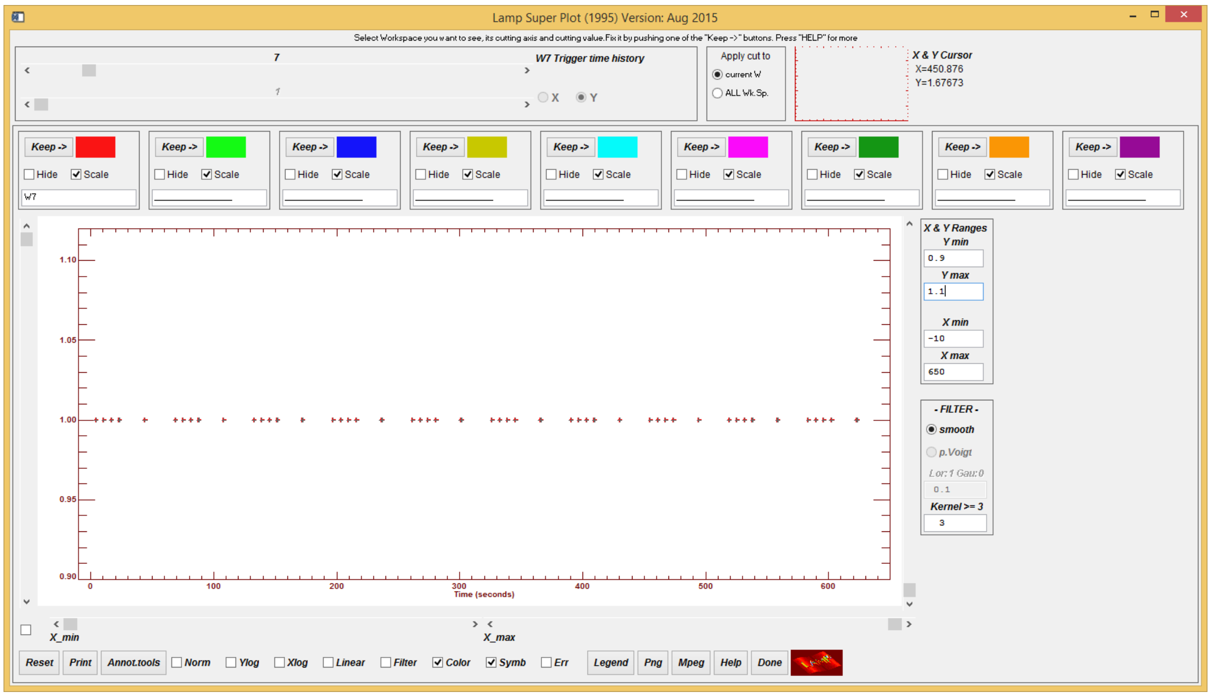
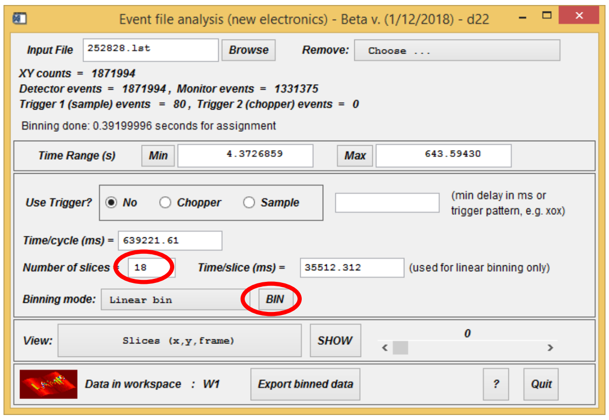


Fig. 9: Superplot window showing the absolute times of the recorded triggers (workspace W7).

4. Doing a simple binning

After reading the data, select the desired number of slices and the binning mode, and click on the BIN button (Fig. 10).



Event file analysis (new electronics) - Beta v. (1/12/2018) - d22

Input File: 252828.lst

Remove: Choose ...

XY counts = 1871994
Detector events = 1871994 , Monitor events = 1331375
Trigger 1 (sample) events = 80 , Trigger 2 (chopper) events = 0

Binning done: 0.39199996 seconds for assignment

Time Range (s) Min 4.3726859 Max 643.59430

Use Trigger? [✓ No] Chopper Sample (min delay in ms or trigger pattern, e.g. xox)

Time/cycle (ms) = 639221.61

Number of slices = 18 Time/slice (ms) = 35512.312 (used for linear binning only)

Binning mode: Linear bin

View: Slices (x,y,frame)

Data in workspace : W1

Fig. 10: Binning a list mode file.

The binning procedure produces a 3D array of dimensions [NX_{pixels}, NY_{pixels}, N_{slices}], which can be plotted using the option *View: Slices (x, y, frame)* (Fig. 11).

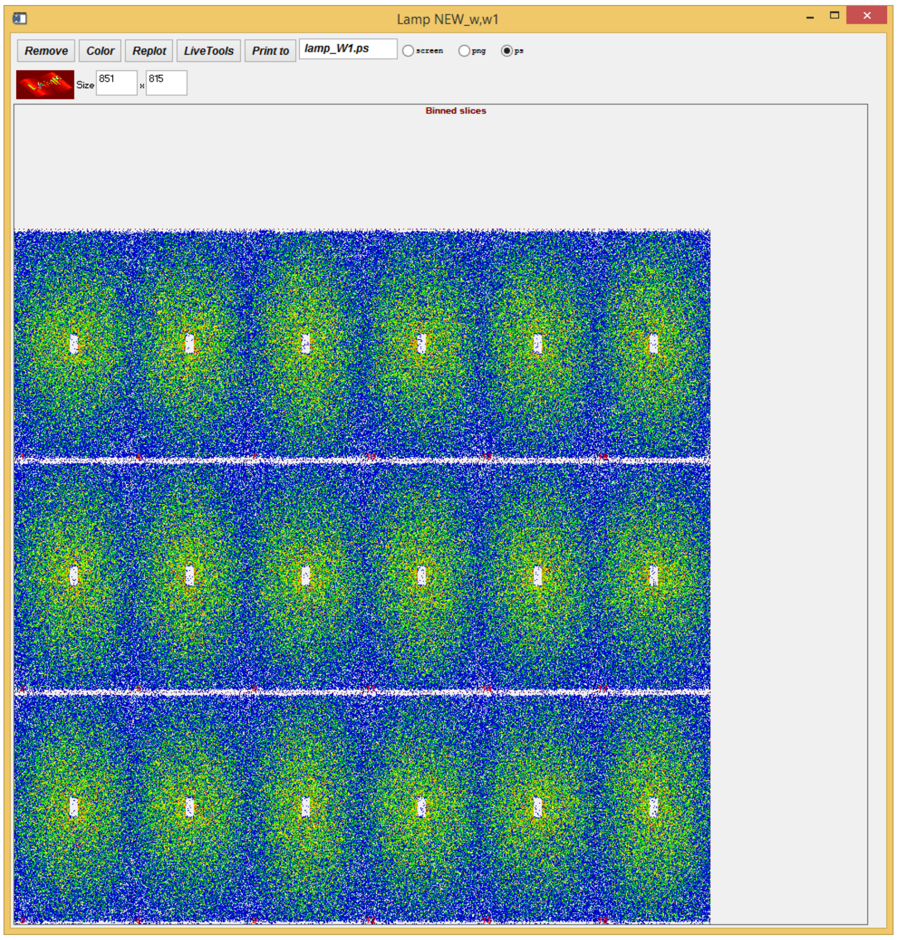


Fig. 11: Lamp graphical window showing the 2D images corresponding to each of the generated slices (workspace W1).

The initial time of each slice is calculated automatically from the duration of the full cycle and the selected number of slices. The time per slice is also calculated accordingly and shown in the *Time/slice* window, but naturally, it applies only to the linear binning case.

Useful: The slice time can be changed manually, allowing in this way to have either overlapping or non-overlapping bins. For example, for a cycle of 10 milliseconds and 10 slices, the default time per slice would be 1 ms and the binning will be done using the following bins: [0-1], [1-2], …, [9-10] ms. However setting manually the time per slice to 2 ms, the 10 slices generated will correspond to the following times: [0-2], [1-3], [2-4], …, [8-10], [9-10] ms. This can be useful in some cases to improve the statistical quality of the generated slices. Although likely less useful, it is also possible to use a smaller time per slice, in which case not all the data will be used to produce the binned slices. For example, using 0.5 ms in the previous example, the 10 bins will contain data measured at times [0-0.5], [1-1.5], [2-2.5], …, [9-9.5] ms.

The *Time/cycle* value can also be changed manually. In most cases, this is not particularly useful, but it can be a good way of forcing similar time binnings when reducing several list mode files having different durations.

On the other hand, the *Time/slice* value is only relevant when the *linear bin* option is selected. In the other three cases, the time limits of each bin are calculated automatically and they are different for different bins, so the value shown in the interface is irrelevant. The *log bin* option create time bins of different lengths using logarithmic binning, i.e. $\log t_{i+1} - \log t_i = \text{constant}$. The *inverse log bin* option calculates log bins and then reverses them, so the duration of the bins reduces with time. The difference between both types of binning can be shown schematically as follows:

Log bin: —|——|———|————|—————|——————|———————|…

Inverse log bin: …|———————|——————|—————|————|———|——|—

Finally, the last option (*Equal # events per bin*) generates also variable time bins whose duration is such that the total number of events in each slice is the same. This could be useful to generate slices having similar statistics.

5. Binning using a trigger

The procedure is analogous to the simple binning explained above, and is applied whenever a trigger (either chopper or sample) is selected in the *Use Trigger?* box. At present, the options in this box are exclusive, so only one trigger can be used at a given time.

In this case, the cycle time is not given by the run duration, but by the trigger period. At present, the code assumes that the trigger events are periodic and calculates the average period, which is then shown in the *Time/cycle* box. Evidently, this is incorrect if the time structure of the trigger is not periodic, as shown for example in Figs. 7-9. In this case, it is recommended to remove any useless trigger (see section Removing bad events or triggers).

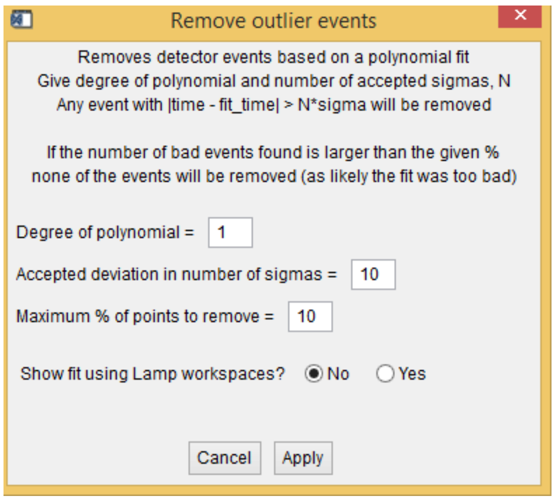
When clicking the bin option while the sample or chopper option is selected, two new internal arrays are generated: Dtimer_trigger and Mtimer_trigger (see appendix 1), containing the relative time of each event with respect to the previous trigger. The binning is done using this relative times!

Care: All the events (either detector or monitor events) taking place before the first trigger event cannot be handled and are automatically discarded. This is done for the array Dtimer_trigger, but also for the Dtimer array (and for Mtimer_trigger and Mtimer, as well). The number of detector events removed is indicated in the information line of the message window, and now the number of detector events will be smaller than the number of XY counts. **Removed events are definitely lost**, so if at a later time you want to do a new binning without the trigger and using the full set of events, you need to read again the file.

As before, you can choose the desired number of slices and the corresponding time for a slice is adjusted automatically, but can be changed manually before pressing the button in order to produce overlapping bins. All the cycles defined by the triggers are considered equivalent, so the counts arriving during relative times corresponding to the 1st bin after each trigger are summed together into the 1st slice, counts with times corresponding to the 2nd bin contribute to the 2nd slice, etc.

6. Removing bad events or triggers

Ideally, all the detector events will be saved correctly in the list mode file. But if for any reason, there is an error in the recorded time for a given event, there are two options available to try to find these rare bad events. The first one (*Remove: Events with times < #1 or > #N*) uses the absolute times of the first and last detected events as a reference and eliminates any event that has a time outside that range. The second option to identify and remove bad events - *Remove: Outlier events (using poly fit)* – fits the time dependence of the events using a polynomial and discards any event whose time deviates too much from the fitted dependence. When this option is selected, a new window appears (Fig. 12), allowing the user to select the degree of the polynomial to use to fit the time dependence and the acceptance region. The latter is given as a multiple M of $\sigma = \sqrt{\chi^2}/N_{events}$ and any event having a time such that $|t_i - t_{fit}| > M\sigma$ will be removed. The window also provides an option to give a maximum number of events (in percentage) that could be removed. If the number of bad events found by the program is larger than this number nothing will be done, as probably the fit was not appropriate. Finally, it is possible to select the possibility of sending the data and the fit to Lamp workspaces 56-60. This is always a good idea, as it is recommended to check graphically using Superplot both the fit and the events identified as bad (see Fig. 13).


Remove outlier events

Removes detector events based on a polynomial fit
Give degree of polynomial and number of accepted sigmas, N
Any event with |time - fit_time| > N*sigma will be removed

If the number of bad events found is larger than the given %
none of the events will be removed (as likely the fit was too bad)

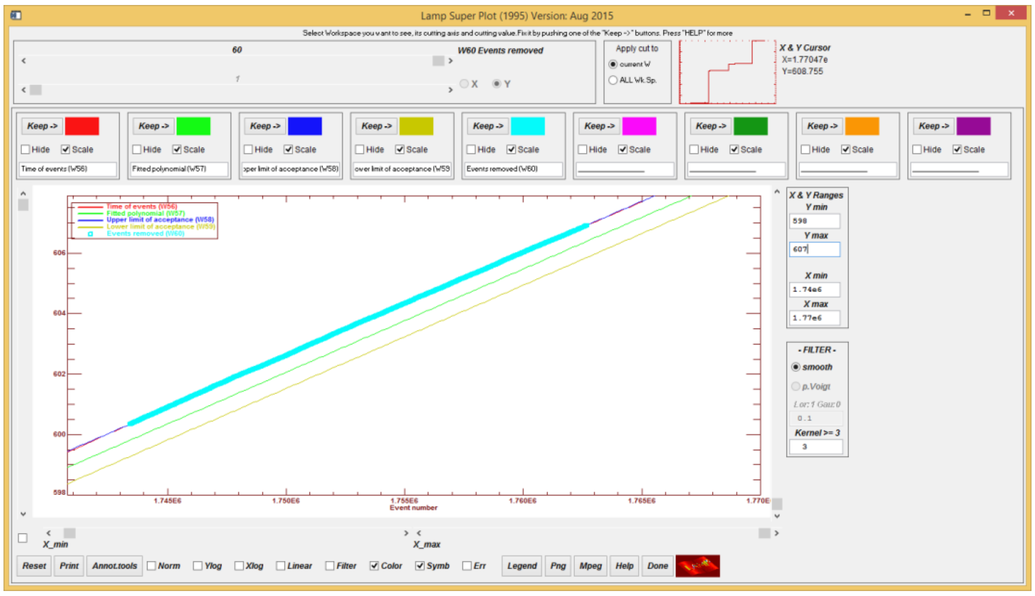
Degree of polynomial = 1

Accepted deviation in number of sigmas = 10

Maximum % of points to remove = 10

Show fit using Lamp workspaces? [✓ No] Yes

Fig. 12: Window to set options to remove events having times showing too large deviations from the expected behaviour.


Lamp Super Plot (1995) Version: Aug 2015

Select Workspace you want to see, its cutting axis and cutting value. Fix it by pushing one of the "Keep ->" buttons. Press "HELP" for more

60 W60 Events removed

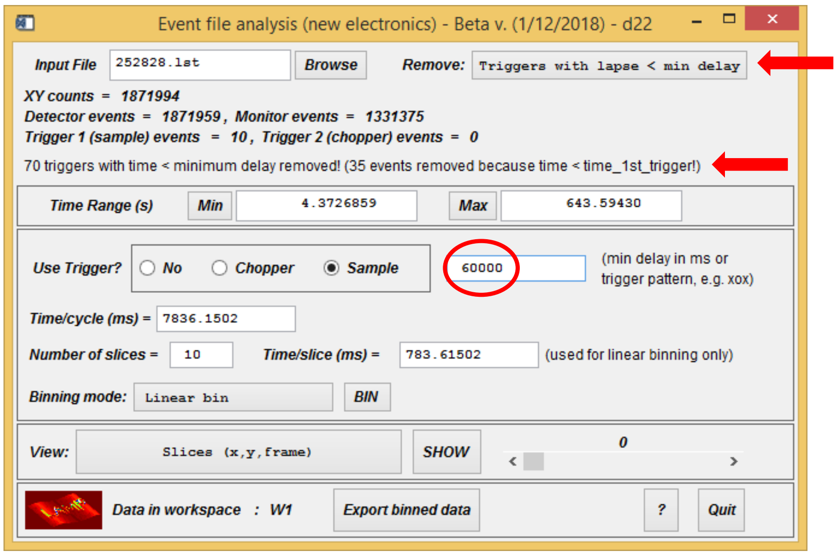
X & Y Cursor X=1.77047e Y=608.755

Time of events (W56) | Fitted polynomial (W57) | Upper limit of acceptance (W58) | Lower limit of acceptance (W59) | Events removed (W60)

X & Y Ranges: Y min 598, Y max 607, X min 1.74e6, X max 1.77e6

Fig. 13: Superplot window showing some events removed due to a too large deviation in their times with respect to the fitted polynomial.

There are also two options available to select which trigger events to keep. The first one is *Remove: Triggers with lapse < min delay*. When selected, a forbidden time is defined after each accepted trigger, so that any trigger arriving during the forbidden period is automatically discarded. This time, in milliseconds, is defined by the user in the *(min delay in ms or trigger pattern)* box (Fig. 14).


Event file analysis (new electronics) - Beta v. (1/12/2018) - d22

Input File: 252828.lst   Remove: Triggers with lapse < min delay

XY counts = 1871994
Detector events = 1871959 , Monitor events = 1331375
Trigger 1 (sample) events = 10 , Trigger 2 (chopper) events = 0
70 triggers with time < minimum delay removed! (35 events removed because time < time_1st_trigger!)

Time Range (s): Min 4.3726859 Max 643.59430

Use Trigger? No Chopper [✓ Sample]   60000 (min delay in ms or trigger pattern, e.g. xox)

Time/cycle (ms) = 7836.1502

Number of slices = 10   Time/slice (ms) = 783.61502 (used for linear binning only)

Binning mode: Linear bin

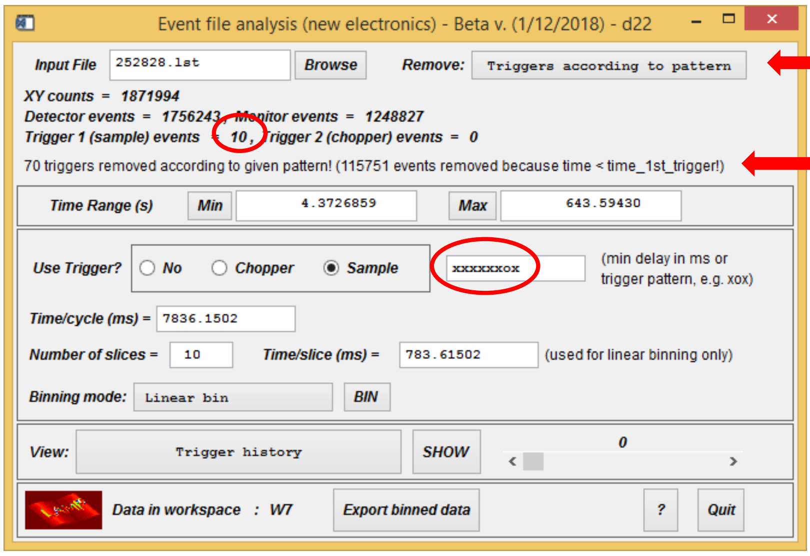
View: Slices (x,y,frame)

Data in workspace : W1

Fig. 14: Example showing how to remove any trigger happening before 1 minute has elapsed from the last accepted trigger.

A second and more flexible option consists in giving a pattern, using also the *(min delay in ms or trigger pattern)* box, to indicate which triggers are to be kept and those to discard. The pattern can be given as a sequence of letters, e.g. 'xxxo', or numbers, e.g. '1110'. The only characters recognized are 'o', 'O', and '0', which are used to identify the triggers to be kept. Thus the two sequences above will keep only the 4th, 8th, 12th, … triggers.

For example, applying the pattern 'xxxxxxox' to the complex trigger structure shown in Figs. 7-9, which repeats every 8 triggers, we will have the window shown in Fig. 15.



Event file analysis (new electronics) - Beta v. (1/12/2018) - d22

Input File: 252828.lst
Remove: Triggers according to pattern

XY counts = 1871994
Detector events = 1756243, Monitor events = 1248827
Trigger 1 (sample) events = 10, Trigger 2 (chopper) events = 0
70 triggers removed according to given pattern! (115751 events removed because time < time_1st_trigger!)

Time Range (s): Min 4.3726859 Max 643.59430

Use Trigger? No / Chopper / [✓ Sample] xxxxxxox (min delay in ms or trigger pattern, e.g. xox)

Time/cycle (ms) = 7836.1502

Number of slices = 10 Time/slice (ms) = 783.61502 (used for linear binning only)

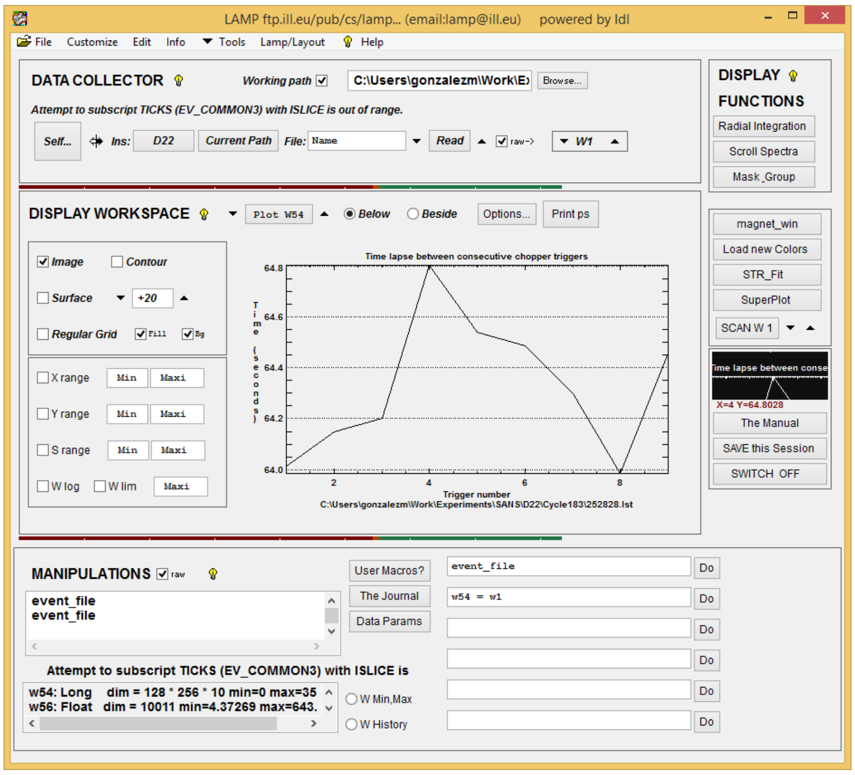
Binning mode: Linear bin

View: Trigger history 0

Data in workspace : W7

Fig. 15: Example showing how to apply a trigger pattern to select the triggers to use in the binning.

If we check now the trigger period (Fig. 16), we observe that we have 9 cycles defined by the 10 triggers kept and that they all have a similar duration.



LAMP ftp.ill.eu/pub/cs/lamp... (email:lamp@ill.eu) powered by Idl

DATA COLLECTOR
Working path ✓ C:\Users\gonzalezm\Work\E
Attempt to subscript TICKS (EV_COMMON3) with ISLICE is out of range.
Ins: D22 File: Name ✓ raw-> W1

DISPLAY WORKSPACE Plot W54 [✓ Below] Beside
✓ Image ☐ Contour
☐ Surface +20
☐ Regular Grid ✓ Fill ✓ Bg
☐ X range Min Maxi
☐ Y range Min Maxi
☐ S range Min Maxi
☐ W log ☐ W lim Maxi

MANIPULATIONS ✓ raw
event_file
event_file

Attempt to subscript TICKS (EV_COMMON3) with ISLICE is
w54: Long dim = 128 * 256 * 10 min=0 max=35
w56: Float dim = 10011 min=4.37269 max=643.

event_file Do
w54 = w1 Do

X=4 Y=64.8028

Fig. 16: Time step between selected triggers, after applying a pattern to skip 7/8 of initial triggers.

Care: Any removed event (either a detector or a trigger event) is permanently lost. The only way of recovering them (for example, to do a new binning using the full set of detector events or a different trigger as reference) is to read again the list mode file.

7. Exporting the binned data

After binning a list mode file, you can export it in order to continue the reduction and analysis of the binned data using other programs, e.g. SansSheet in Lamp or Grasp. Select *Export binned data → Current File* and the current 3D array of dimensions [NX_{pixels}, NY_{pixels}, N_{slices}] will be saved as a standard NeXus file, named **######_lst.nxs**, where ####### is the number of the list mode file. The NeXus file generated is of the kinetic type and keeps all the instrument parameters present in the original ######.nxs file, so it can be read and manipulated as a standard ILL NeXus numor.

Care: The output NeXus file is generated by modifying the corresponding NeXus file, so in order to be able to export the binned data you need to have both files (######.lst and ######.nxs) in the same directory. Additionally, the routine to modify the data of the input NeXus depends on finding the right tags, so it is very dependent on any change in the naming of the NeXus files. Contact me if you have problems exporting due to changes in the NeXus files.

If you have recorded a large number of list mode files and want to apply the same treatment to all of them, use the option *Export binned data → Apply settings to a list of files and export*. This will open a dialogue window that allows you to select as many list mode files as you want. The selected files are then read one by one, binned using the current settings, and exported.

Care: At present, when exporting automatically a list of files, no bad events or triggers are removed. Therefore if you need this, you will have to treat and export your list mode files manually one by one.

Appendix 1: Internal arrays, mapping with channels registered in list mode files and memory requirements

All the events registered in the list mode file are read in the following internal arrays:

Counts: 2D array (dimensions Xsize and Ysize, hardcoded for each instrument in function read_lst) containing the integrated detector image.

Xdet: 1D array containing the horizontal position of each event registered in the detector.

Ydet: 1D array containing the vertical position of each event registered in the detector.

Dtimer: 1D array containing the time of each event registered in the detector.

Mtimer: 1D array containing the time of each event registered in the monitor.

Ctimer: 1D array containing the time of each event registered in any equipment (but called chopper in the interface).

Stimer: 1D array containing the time of each event registered in a second equipment (but called sample in the interface).

Additionally, if the chopper or sample trigger options are selected, two new arrays are created:

Dtimer_trigger: 1D array containing the time since the last trigger of each detector event.

Mtimer_trigger: 1D array containing the time since the last trigger of each monitor event.

D22 mapping: **Note that this could change depending on how the electronic cards are connected! If the channels do not correspond to the mapping below, some type of events would either be skipped or read in the wrong array!**

Detector dimensions = 128 (horizontal) x 256 (vertical)

Cards 1-4 (type 1740 QDIV) register detector events

Card 5 (GGPickup) register monitor events and two other channels, that can be used to store e.g. a chopper pickup or a signal marking the start of a pulse applied to the sample. At present:

Channel 1 events are stored in Stimer

Channel 2 events are stored in Mtimer

Channel 4 events are stored in Ctimer

Memory requirements: All the events are stored in memory, so if the list mode file is too large, it can be impossible to read it if the computer does not have enough RAM. As an approximate reference, after reading a file occupying 505 MB on the hard disk and containing more than 33×10^6 events, the dynamic memory used by Lamp to store all the relevant information is ~760 MB. And after doing a binning using 100 slices and a trigger (so that the two additional arrays Dtimer_trigger and Mtimer_trigger are generated) it goes up to ~1020 MB. In this case, the reading time can be of the order of 10-30 seconds, and the binning takes about 5-10 seconds. As a rule of thumb, avoid generating files larger than 2 GB.