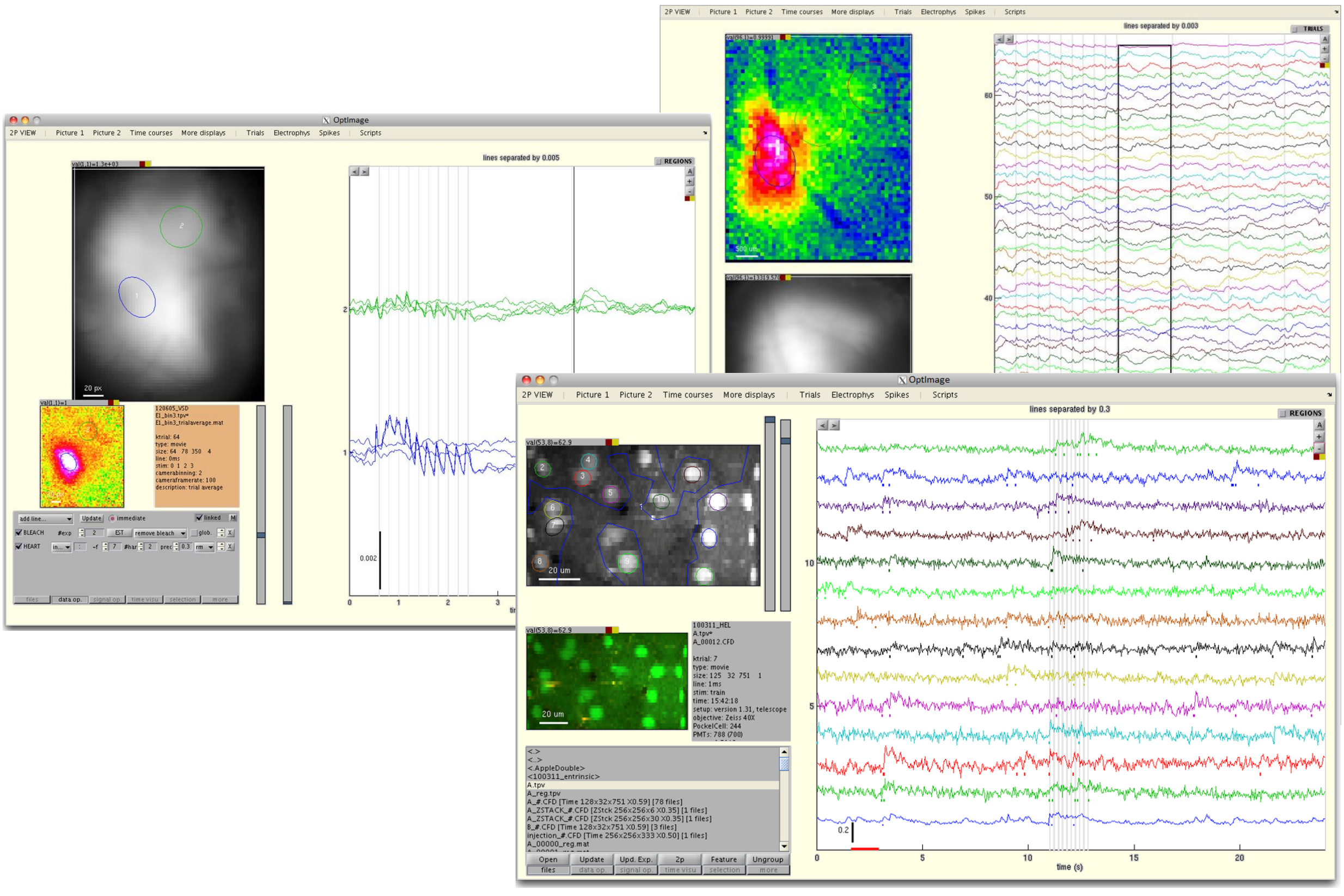
OptImage


OptImage
2P VIEW | Picture 1 | Picture 2 | Time courses | More displays | Trials | Electrophys | Spikes | Scripts
lines separated by 0.005
REGIONS
20 px
120605_VSD
E1_bin3.tpv*
E1_bin3_trialaverage.mat
ktrial: 64
type: movie
size: 64 78 350 4
line: 0ms
stim: 0 1 2 3
camerabinning: 2
cameraframerate: 100
description: trial average
add line...
Update
immediate
linked
BLEACH
HEART
remove bleach
files
data op.
signal op.
time visu
selection
more
lines separated by 0.003
TRIALS
500 um
OptImage
lines separated by 0.3
REGIONS
20 um
100311_HEL
A.tpv*
A_00012.CFD
ktrial: 7
type: movie
size: 125 32 751 1
line: 1ms
stim: train
time: 15:42:18
setup: version 1.31, telescope
objective: Zeiss 40X
PockelCell: 244
PMTs: 788 (700)
A.tpv
A_reg.tpv
A_#.CFD [Time 128x32x751 X0.59] [78 files]
A_ZSTACK_#.CFD [ZStck 256x256x6 X0.35] [1 files]
A_ZSTACK_#.CFD [ZStck 256x256x30 X0.35] [1 files]
B_#.CFD [Time 128x32x751 X0.59] [3 files]
injection_#.CFD [Time 256x256x333 X0.50] [1 files]
A_00000_reg.mat
Open
Update
Upd. Exp.
2p
Feature
Ungroup
time (s)

Requirements

- Matlab 7.14 (R2012a) or later with
  - Image Toolbox
  - Optimization toolbox (necessary only for specific functionalities such as the correction for bleaching and for heart artifact)
- 8GB RAM or more is recommended for analyzing large data

Installation

- Unzip the program to a target location
- Run the function ‘optimage.m’ in Matlab
- Press the button ‘request a license number’ to get an identifier unique to your machine and send it to your provider
- Once your provider returned a license number to you, start again the program, press button ‘activate your license’ and copy the number there

I. Quick Tour

1) Processing the data
2) Visualizing the data
3) Other useful tools

1) Processing the data

a. Open the data

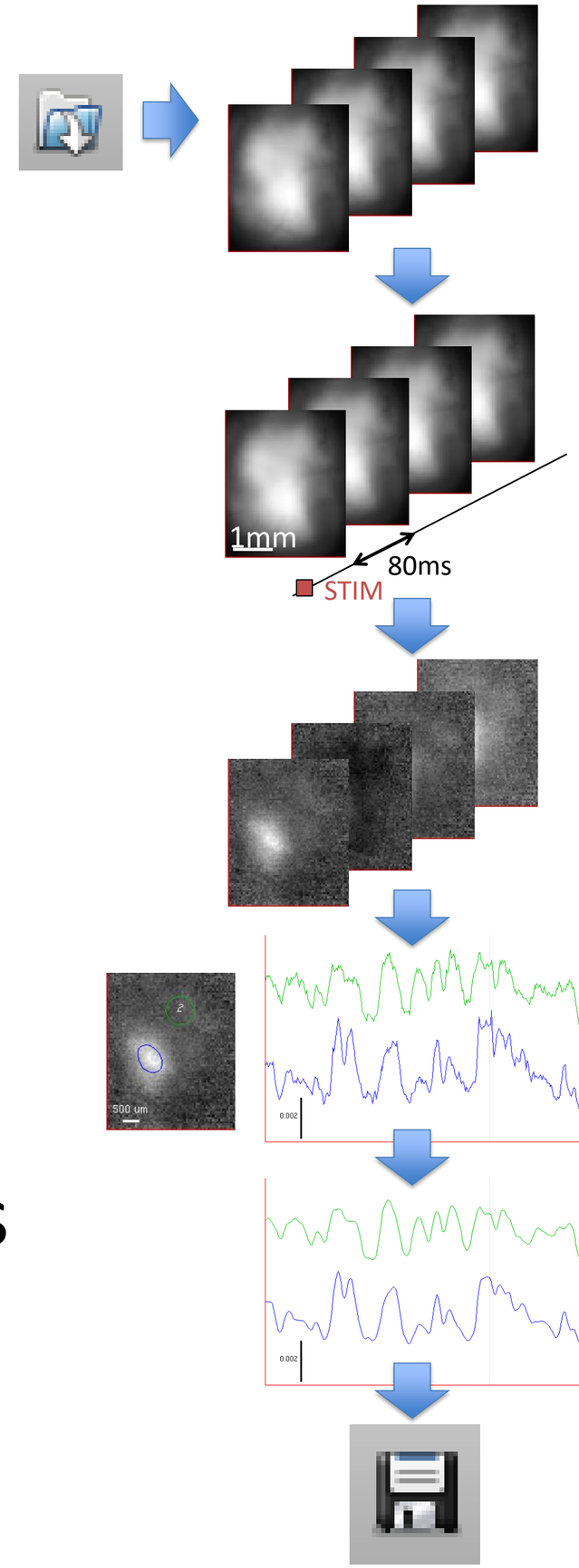
b. Header information

c. Operations on movie data

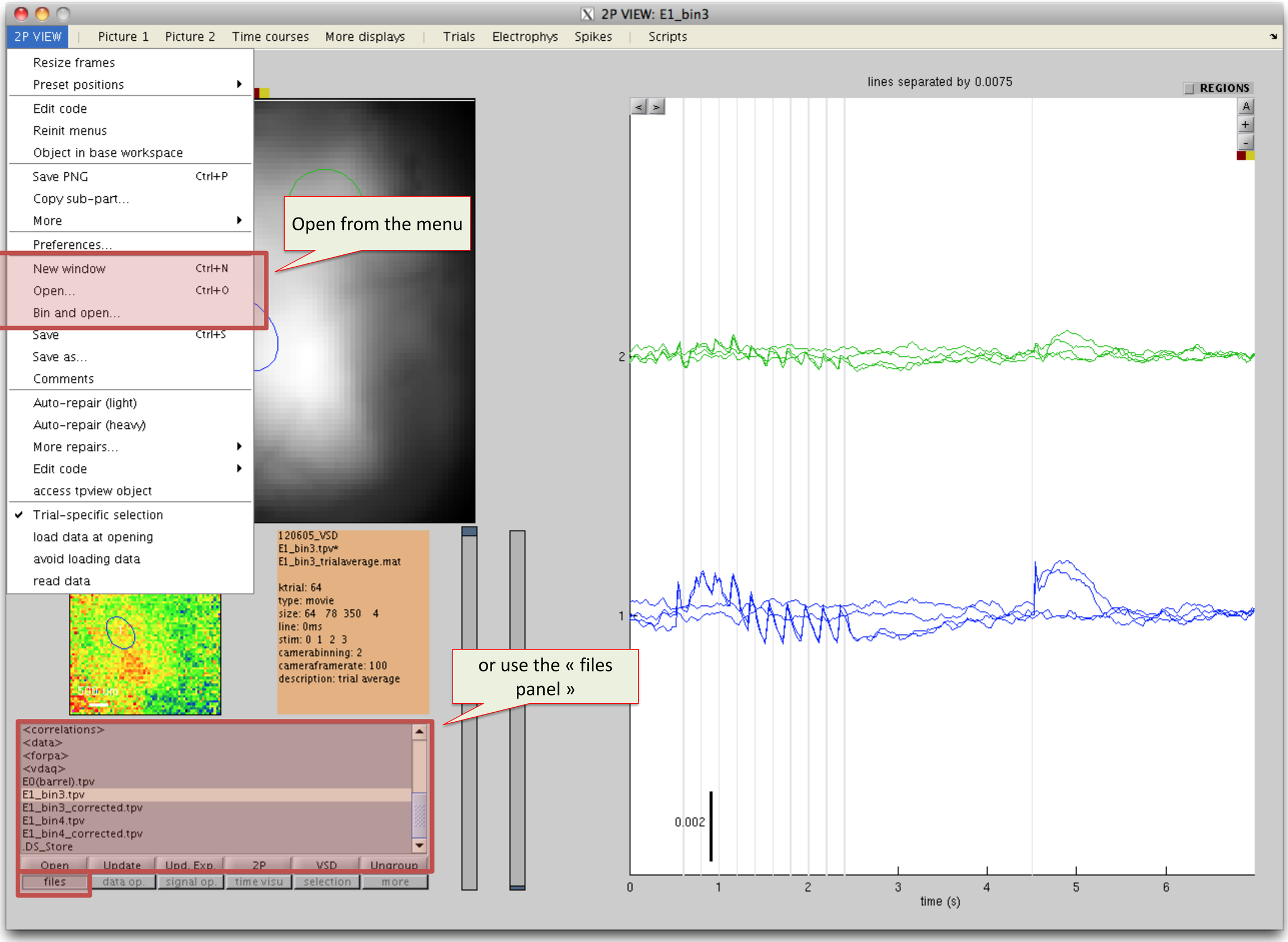
d. Extraction of time courses

e. Operations on time courses

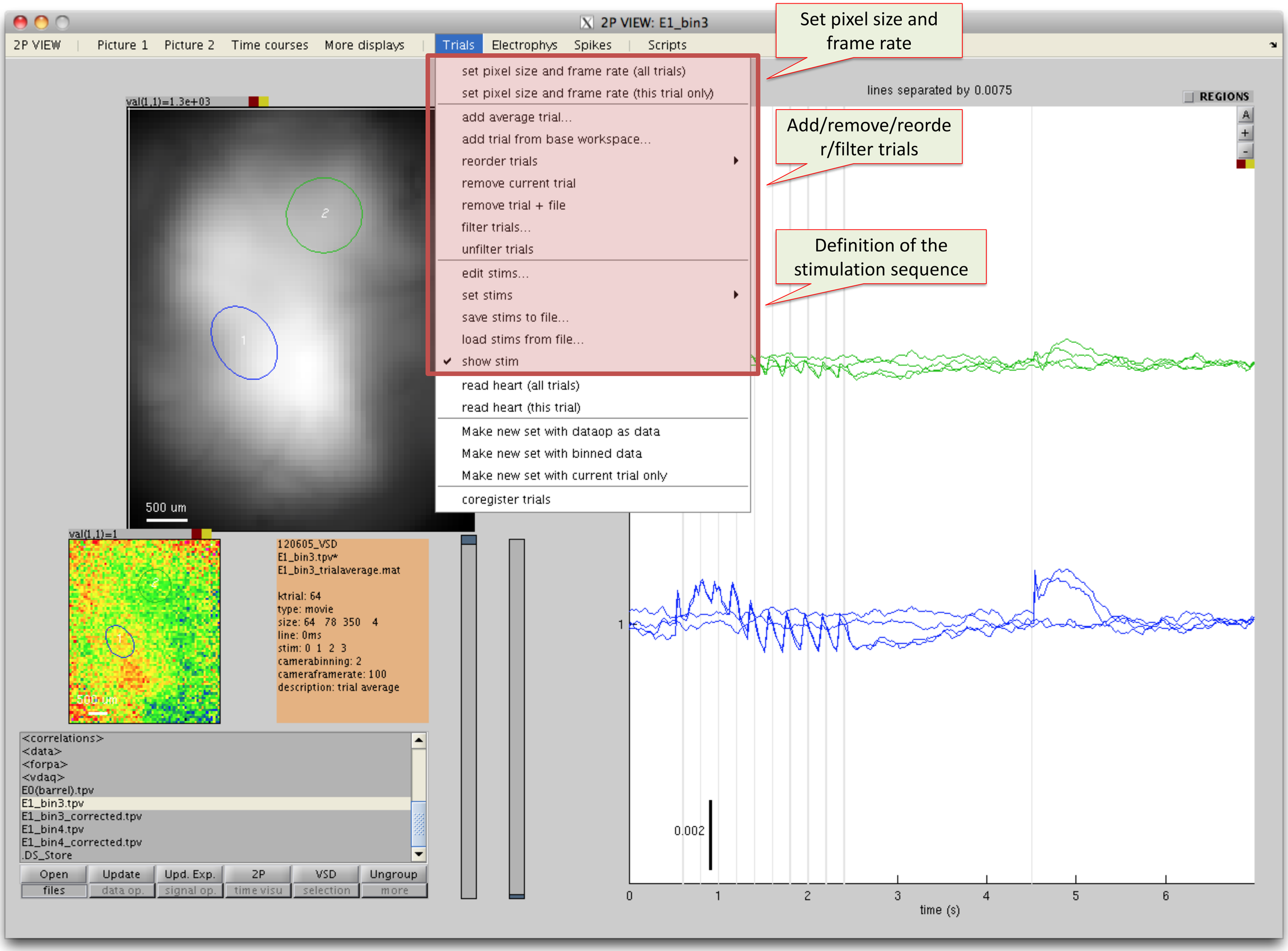
f. Save



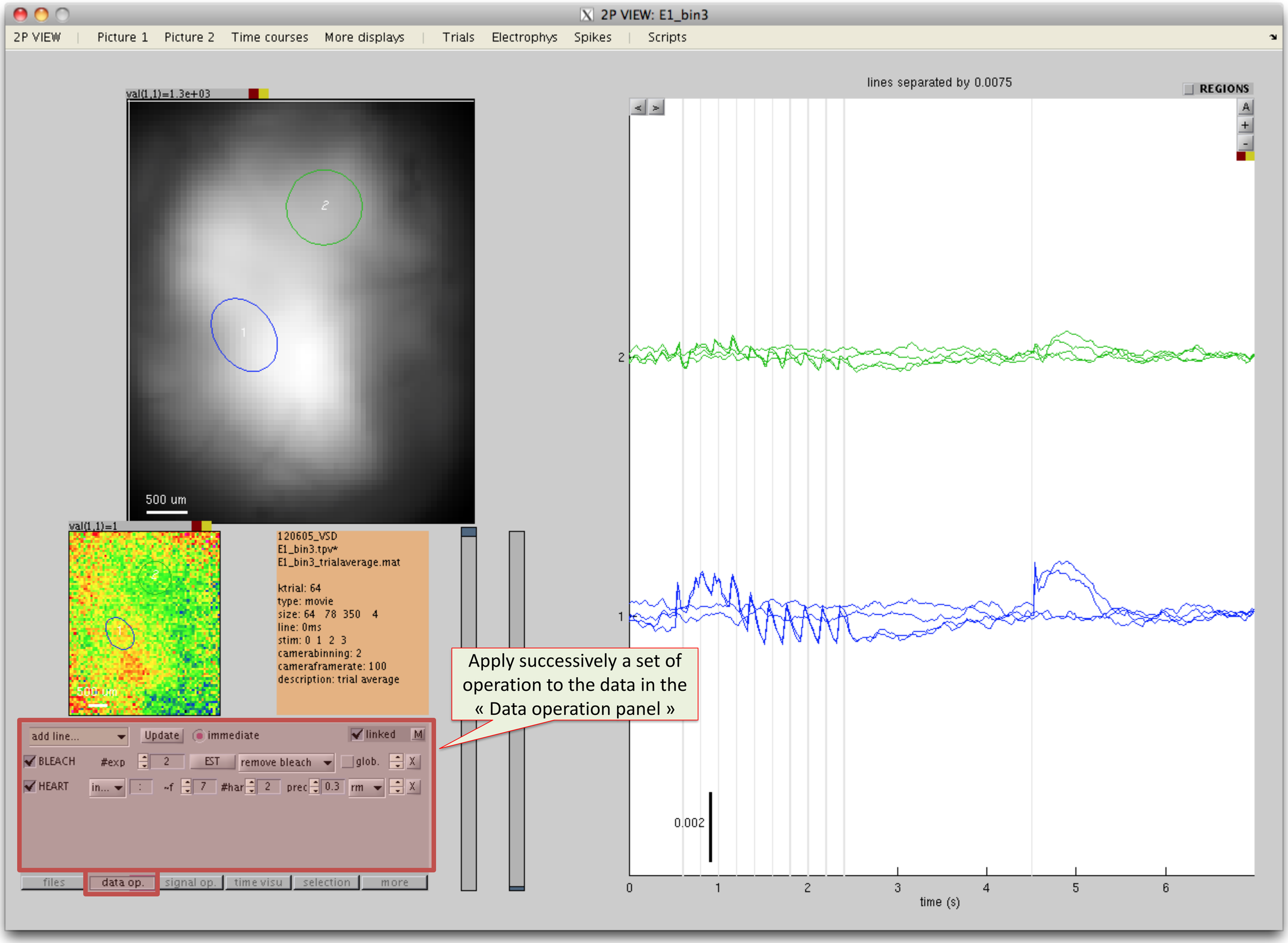
1)a. Open the data



1)b. Header information

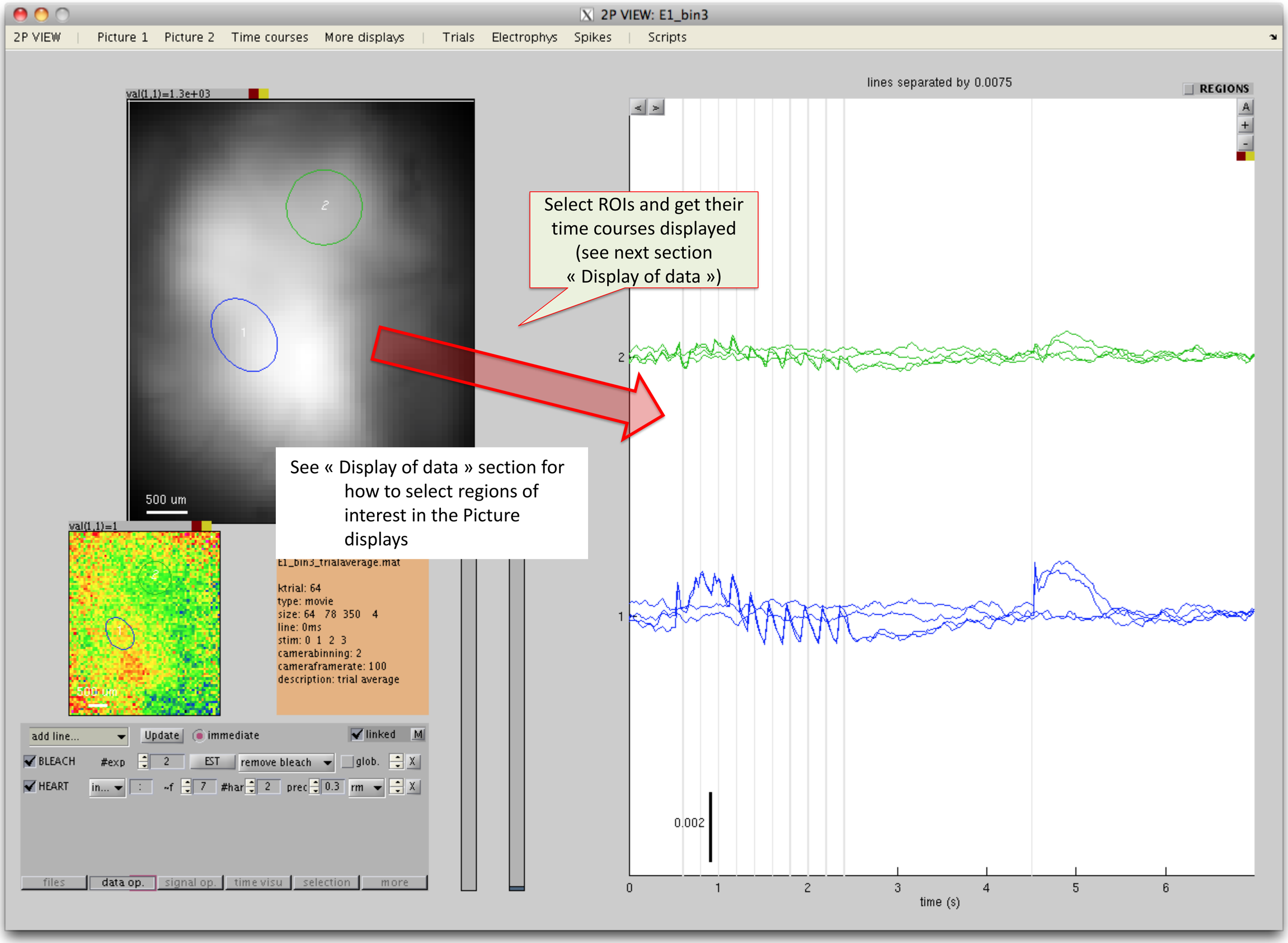


1)c. Operations on movie data

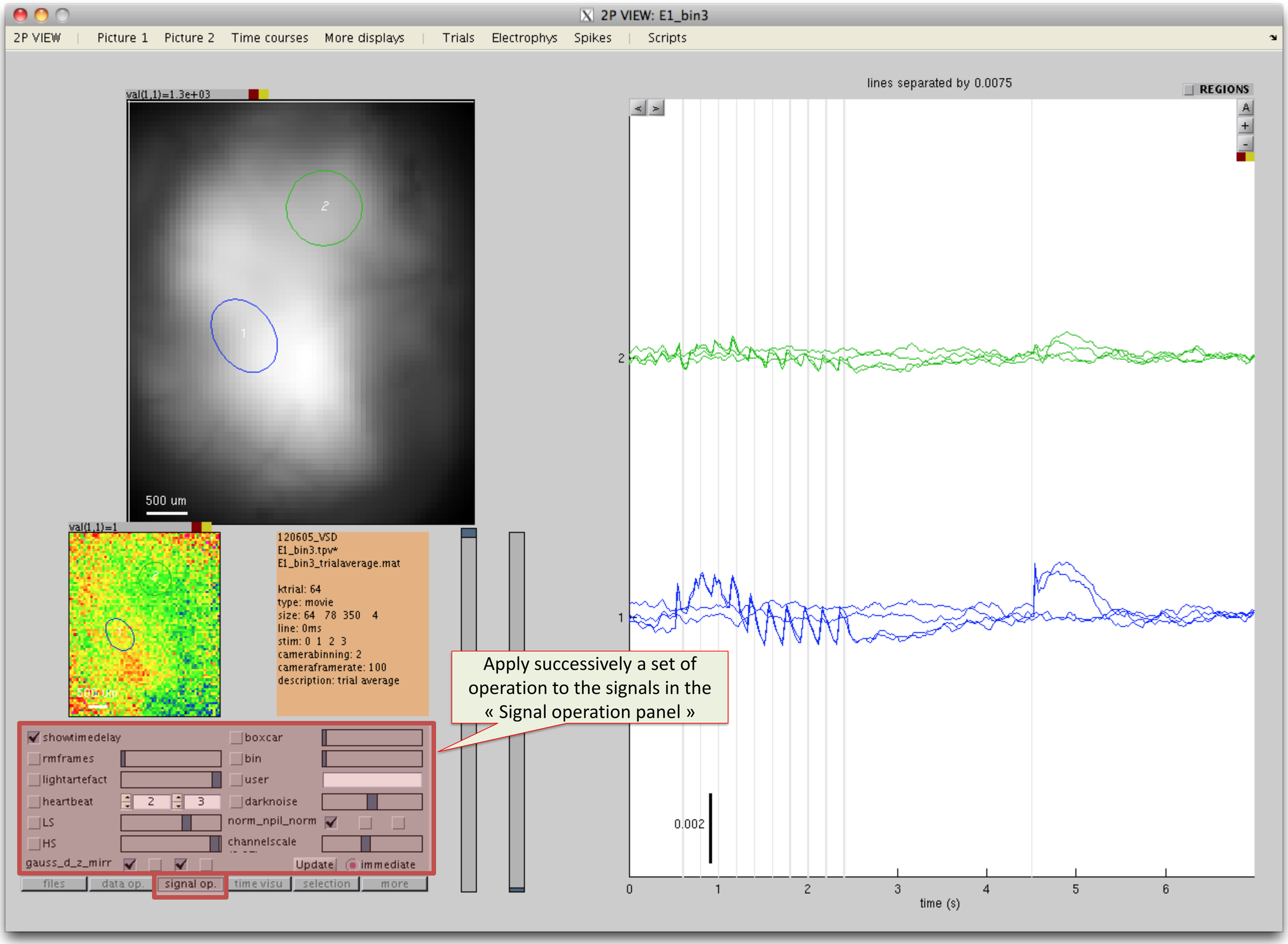


Apply successively a set of operation to the data in the « Data operation panel »

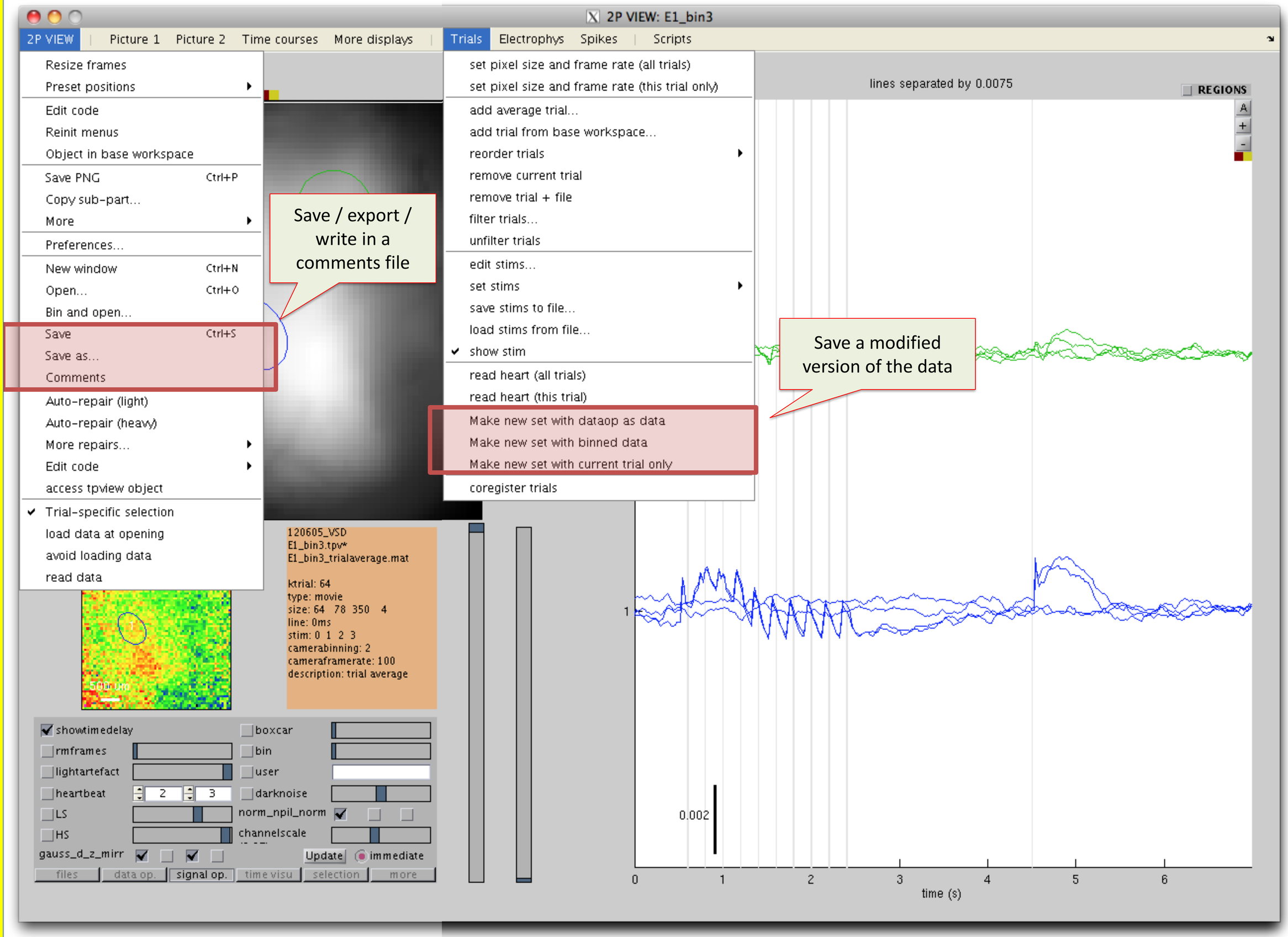
1)d. Extraction of time courses



1)e. Operations on time courses



1)f. Save


2P VIEW: E1_bin3
2P VIEW
Picture 1
Picture 2
Time courses
More displays
Trials
Electrophys
Spikes
Scripts
Resize frames
Preset positions
Edit code
Reinit menus
Object in base workspace
Save PNG Ctrl+P
Copy sub-part...
More
Preferences...
New window Ctrl+N
Open... Ctrl+O
Bin and open...
Save Ctrl+S
Save as...
Comments
Auto-repair (light)
Auto-repair (heavy)
More repairs...
Edit code
access tpview object
Trial-specific selection
load data at opening
avoid loading data
read data
Save / export / write in a comments file
set pixel size and frame rate (all trials)
set pixel size and frame rate (this trial only)
add average trial...
add trial from base workspace...
reorder trials
remove current trial
remove trial + file
filter trials...
unfilter trials
edit stims...
set stims
save stims to file...
load stims from file...
show stim
read heart (all trials)
read heart (this trial)
Make new set with dataop as data
Make new set with binned data
Make new set with current trial only
coregister trials
Save a modified version of the data
lines separated by 0.0075
REGIONS
A
+
-
120605_VSD
E1_bin3.tpv*
E1_bin3_trialaverage.mat
ktrial: 64
type: movie
size: 64 78 350 4
line: 0ms
stim: 0 1 2 3
camerabinning: 2
cameraframerate: 100
description: trial average
500 µm
showtimedelay
rmframes
lightartefact
heartbeat
LS
HS
gauss_d_z_mirr
boxcar
bin
user
darknoise
norm_npil_norm
channelscale
Update
immediate
files
data op.
signal op.
time visu
selection
more
1
0.002
0
1
2
3
4
5
6
time (s)

2) Visualizing the data

a. Which displays are available?

b. What to show in each display?
   - Picture1 and Picture2
   - Time Courses

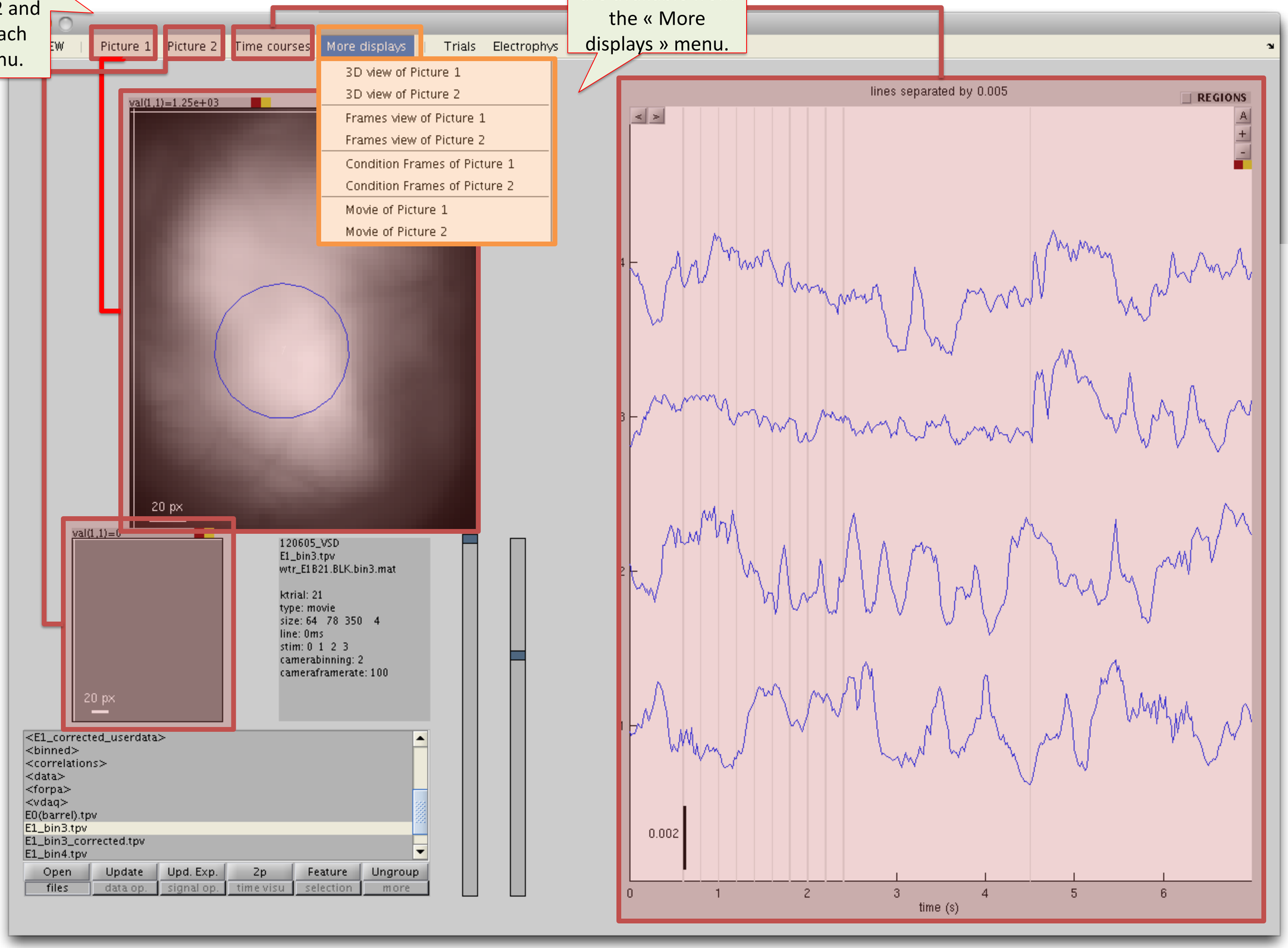
c. Display options
   - Picture1 and Picture2
   - Time Courses

d. Mouse actions
   - Principles of region selection
   - Table of mouse actions

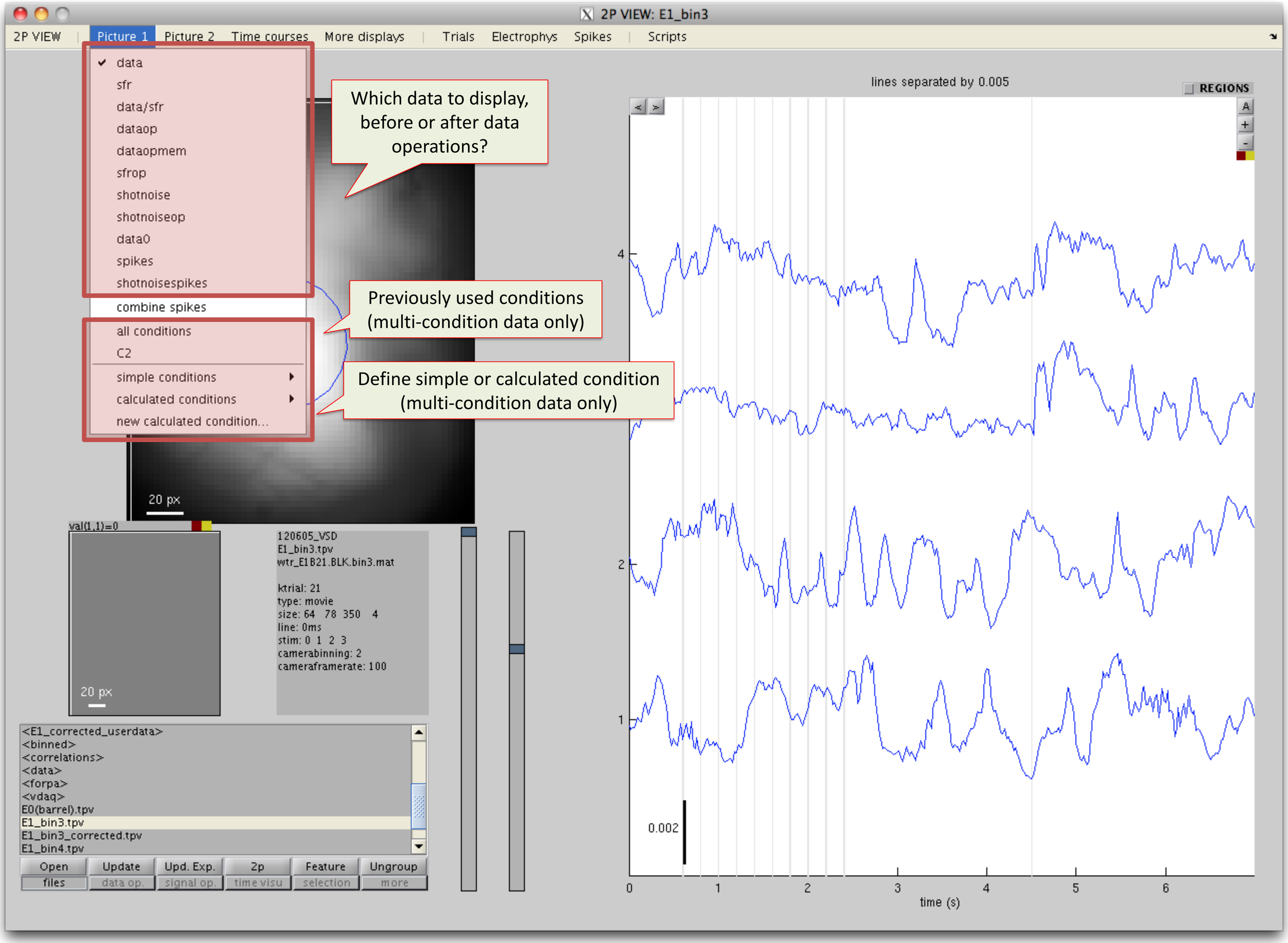
2)a. Which displays are available?

Three default displays: Picture1, Picture2 and Time courses. Each has its own menu.

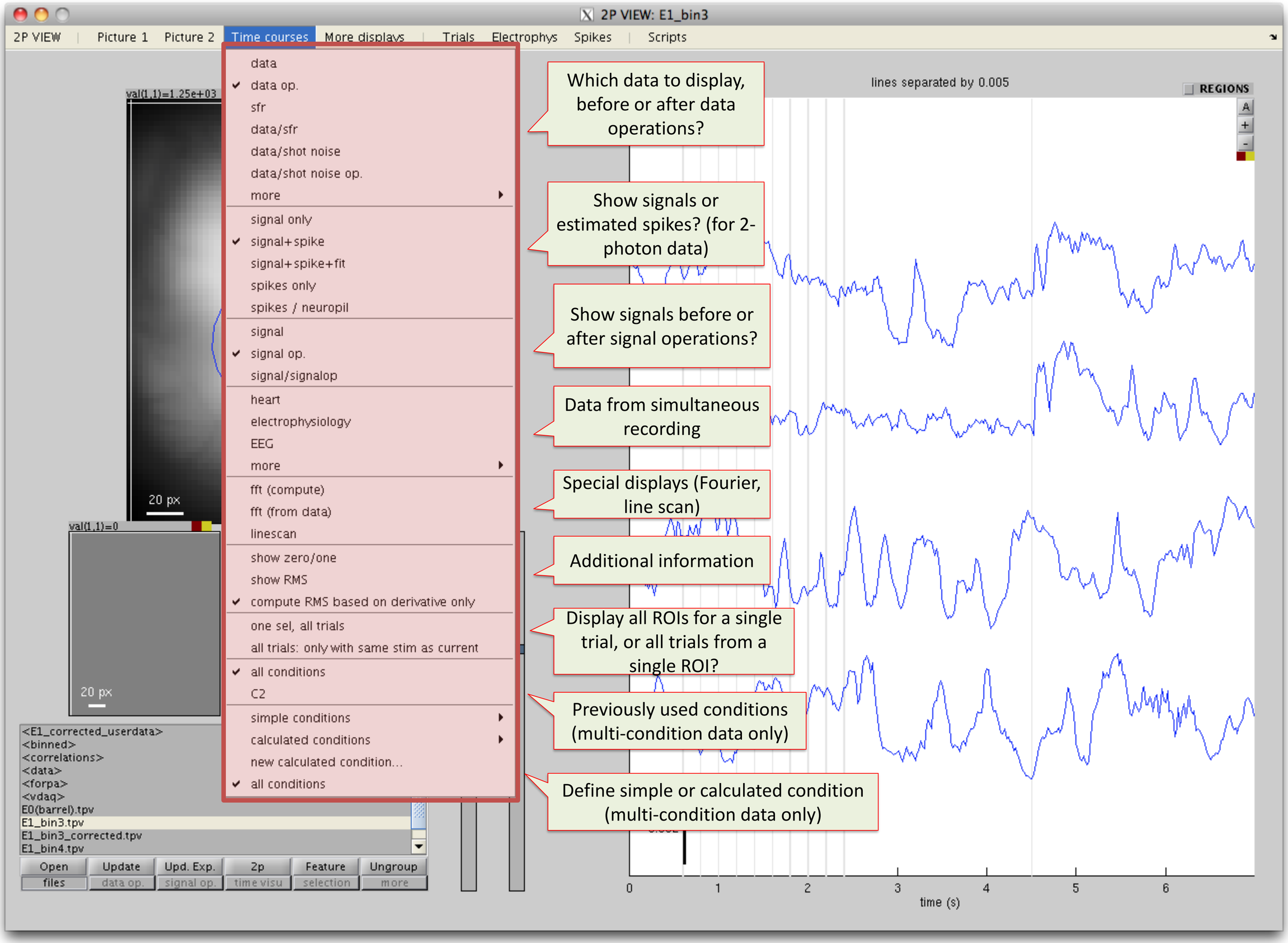
Additional displays are available from the « More displays » menu.



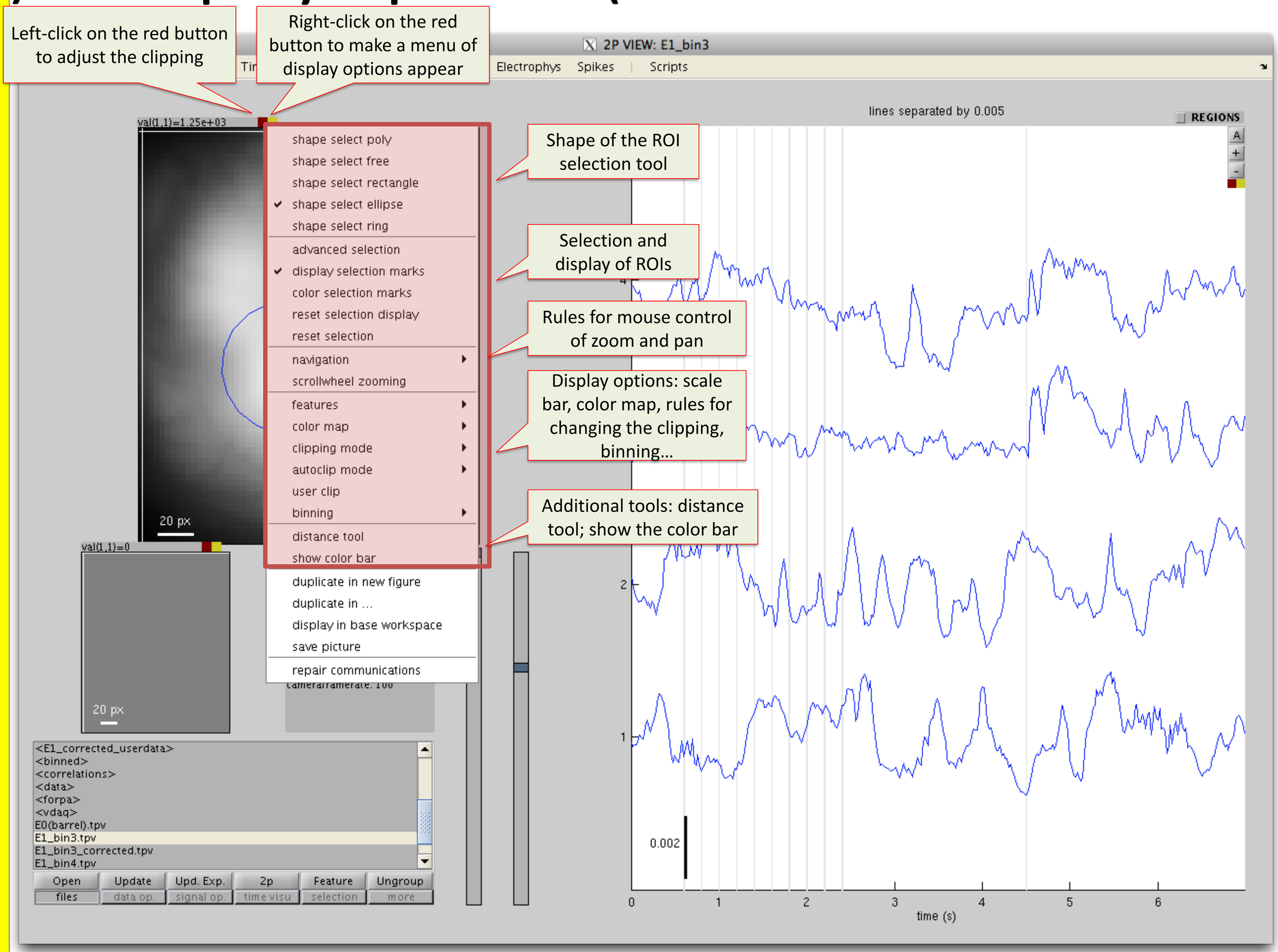
2)b. What to display in Picture1 or Picture2



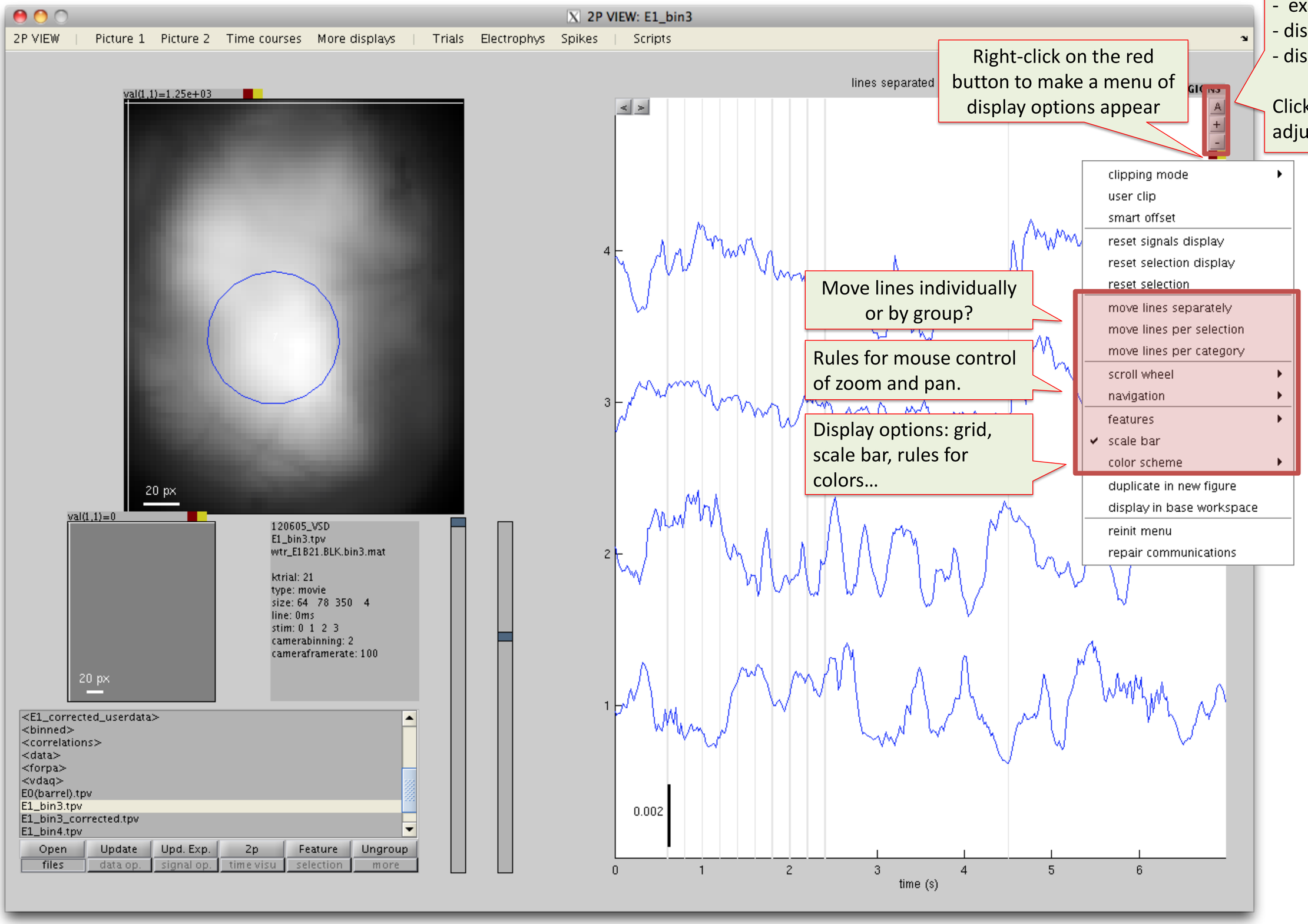
2)b. What to display in Time Courses?



2)c. Display options (Picture1 and Picture2)



2)c. Display options (Time Courses)

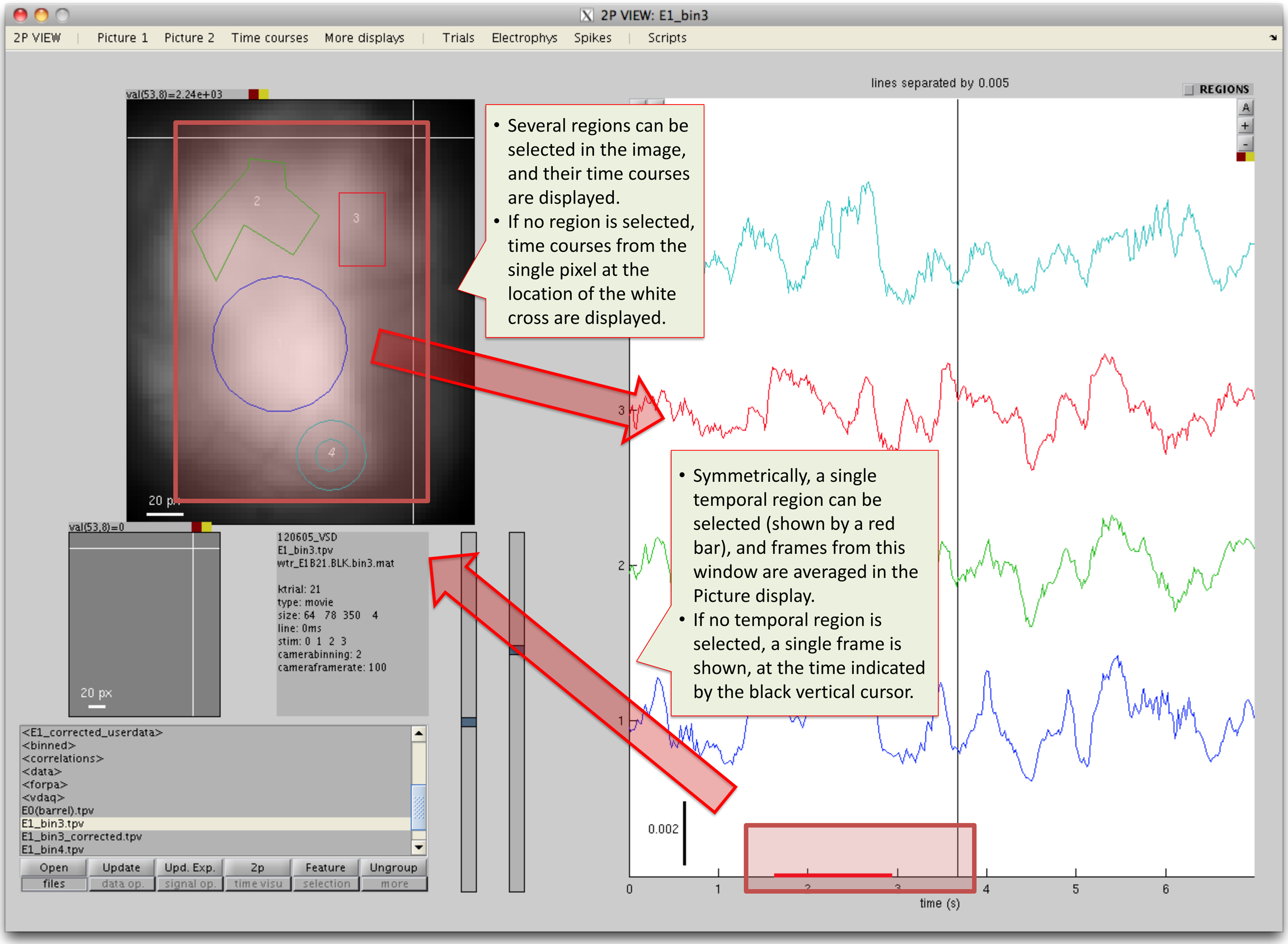


Right-click on the red button to make a menu of display options appear

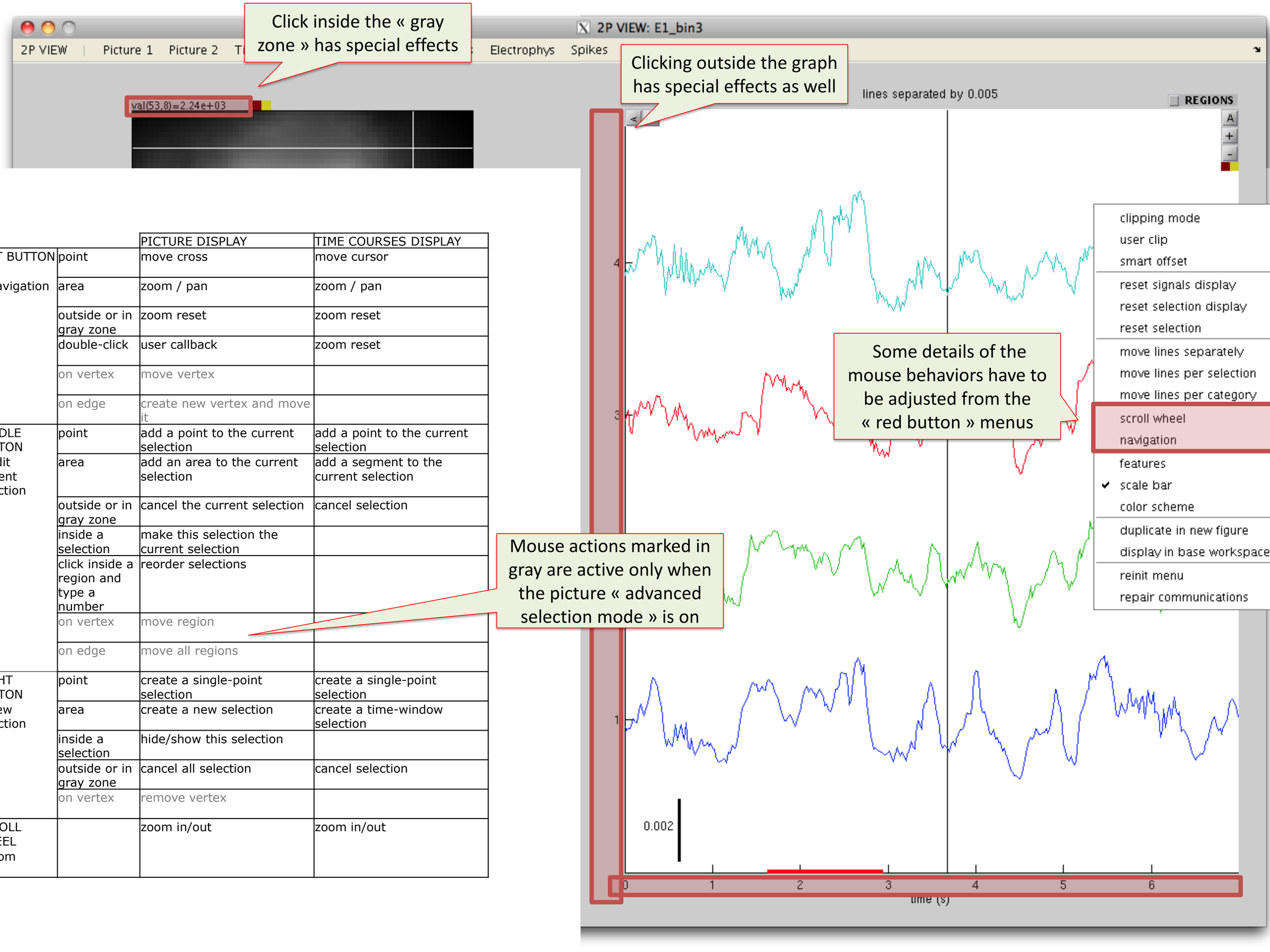
Click on ‘A’ to switch between 3 automatic display modes:
- exact display
- dispatch regions
- dispatch conditions

Click on ‘+’ or ‘-‘ to adjust the size.

2)d. Principles of region selection



2)d. Mouse actions



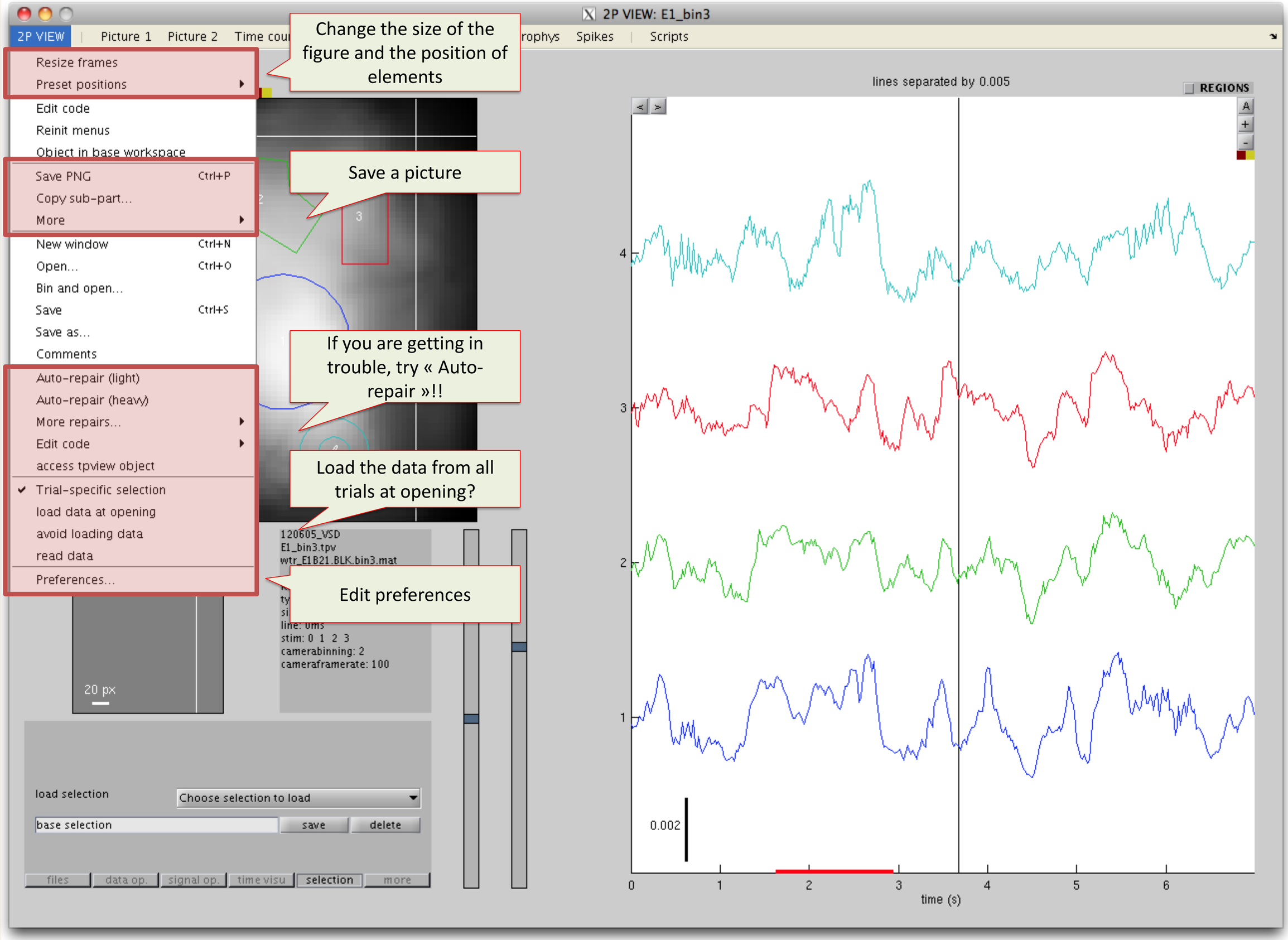
		PICTURE DISPLAY	TIME COURSES DISPLAY
LEFT BUTTON = navigation	point	move cross	move cursor
	area	zoom / pan	zoom / pan
	outside or in gray zone	zoom reset	zoom reset
	double-click	user callback	zoom reset
	on vertex	move vertex	
	on edge	create new vertex and move it	
MIDDLE BUTTON = edit current selection	point	add a point to the current selection	add a point to the current selection
	area	add an area to the current selection	add a segment to the current selection
	outside or in gray zone	cancel the current selection	cancel selection
	inside a selection	make this selection the current selection	
	click inside a region and type a number	reorder selections	
	on vertex	move region	
	on edge	move all regions	
RIGHT BUTTON = new selection	point	create a single-point selection	create a single-point selection
	area	create a new selection	create a time-window selection
	inside a selection	hide/show this selection	
	outside or in gray zone	cancel all selection	cancel selection
	on vertex	remove vertex	
SCROLL WHEEL =zoom		zoom in/out	zoom in/out

3) Other useful tools

a. The 2Pview menu

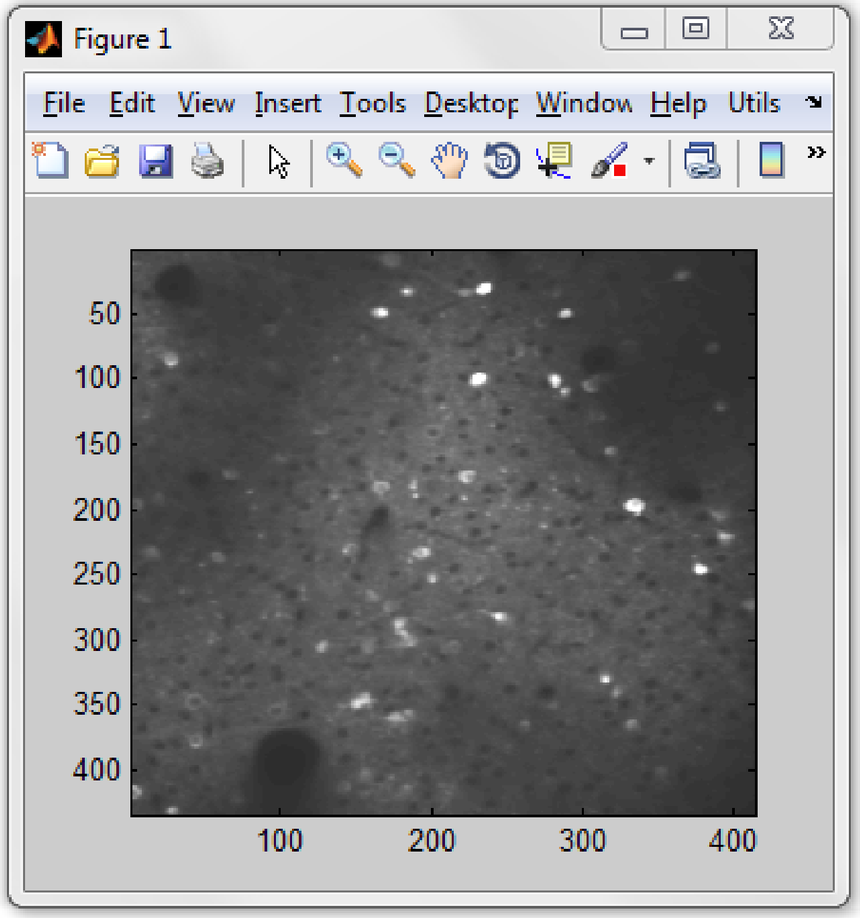
b. Access to the data from the command line

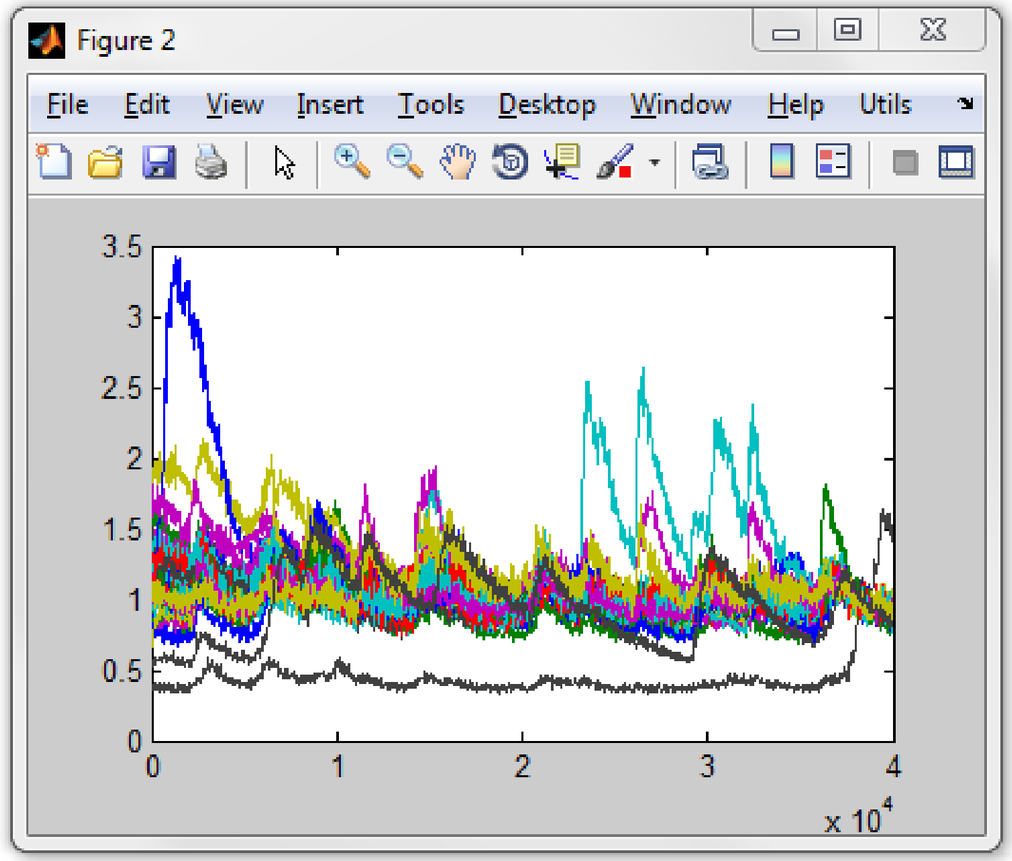
3)a. The « 2Pview » menu



3)b. Access from command line

```
>> figure(1), imagesc(V.trial.data0',[100 7000])
>> figure(2), plot(V.tidx,[V.getsignals('trial').dataop])
>>
```





Main variables of interest:

V.content	all the data	V.trial	current trial
V.content.trials	all the trials	V.data	current trial raw data
V.content.signals	all the time courses	V.dataop	current trial processed data

V. getsignals('current|sel|trial|all') get specific time courses

II. More details

1) Processing the data
2) Visualizing the data
3) Other useful tools

1) Processing the data

a. Open the data

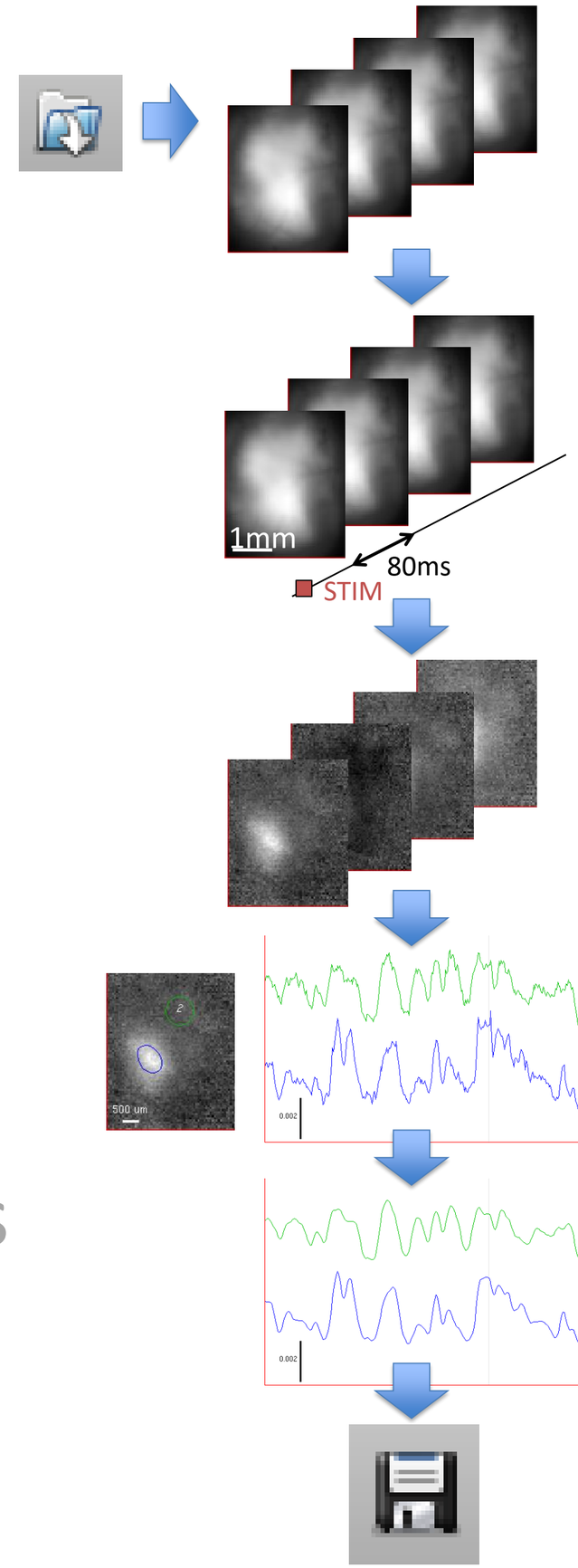
b. Header information

c. Operations on movie data

d. Extraction of time courses

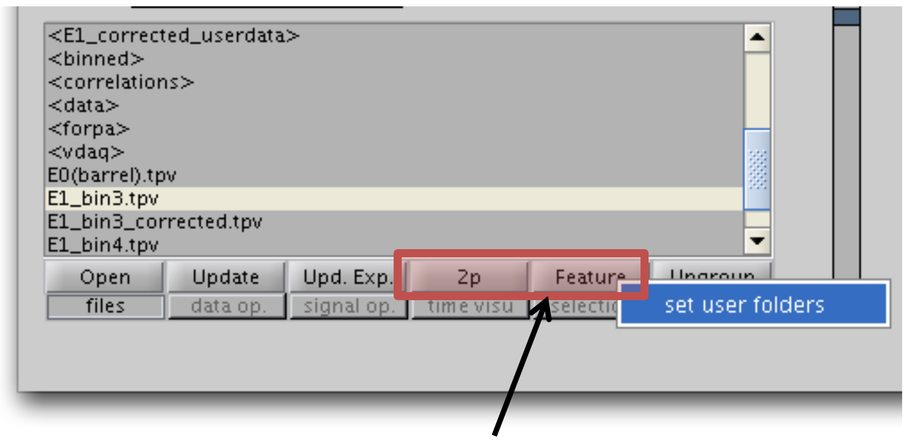
e. Operations on time courses

f. Save



1)a. Open the data

The « file panel »



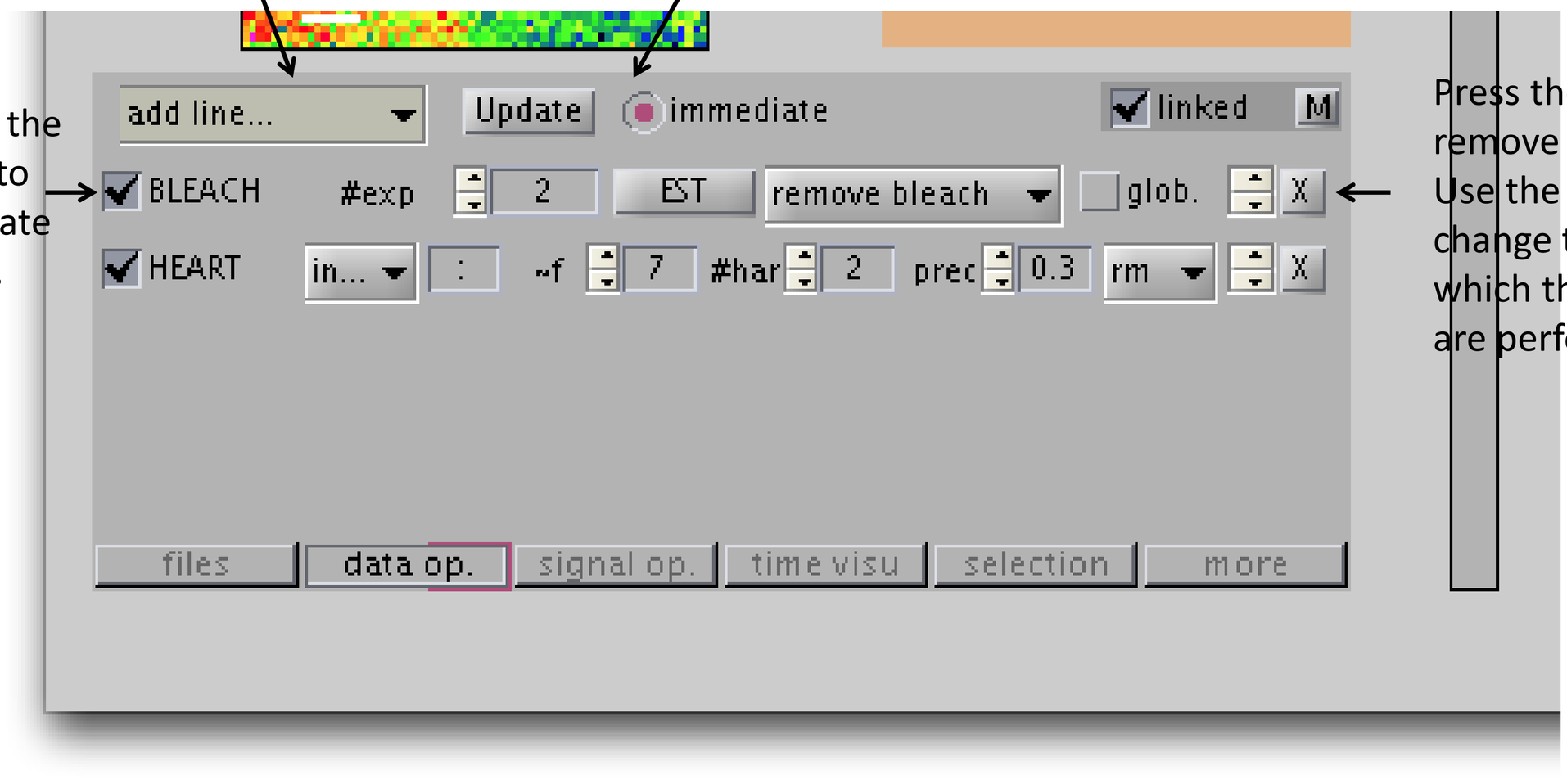
These two buttons provide the ability to create shortcuts to your two preferred data folders. Right-click on any of them to define these folders.

1)c. Operations on movie data

Apply successively a set of operations to the data in the « Data operation panel ». The processed data is called “dataop”. To see the result of the operations, the displays should show the variable “dataop” rather than “data”.

Start by pressing “add line…” to select an operation (all different operations are presented in the next slides).

If “immediate” is checked, the data is re-processed and re-displayed immediately after every change in the operation definitions.
To save computation time, uncheck “immediate” and press “Update” only after all operations have been properly defined.



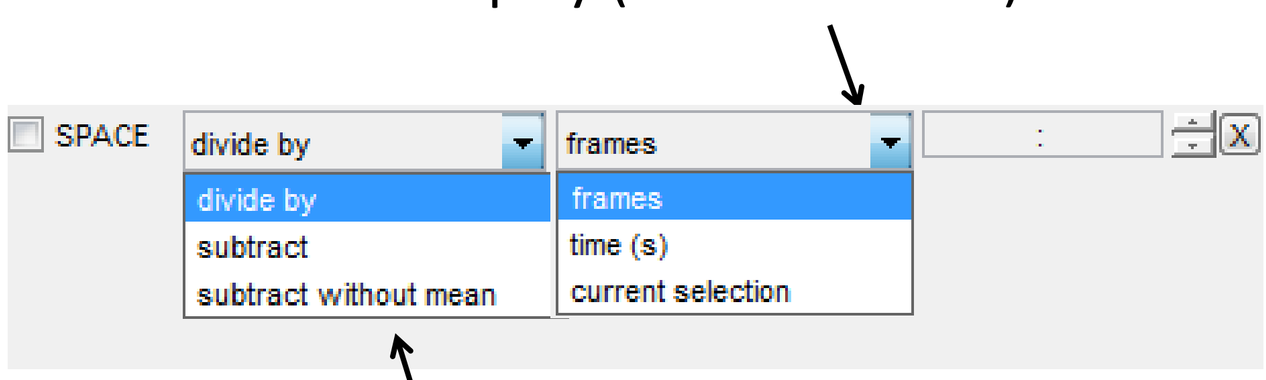
Check/uncheck the box on the left to activate/inactivate each operation.

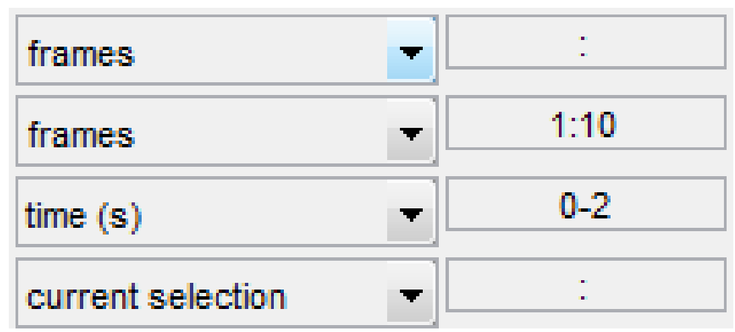
Press the cross to remove an operation. Use the arrows to change the order in which the operations are performed.

1)c “SPACE” operation : frame normalization

An average frame is calculated from the movie. There are 3 ways to specify which frames to average together:

- Define a frame range (e.g. 1:10) , or use “:” to use all frames
- Define a time range in seconds (e.g. 0-2)
- Use the current temporal selection in the Time Courses display (see section 2d).





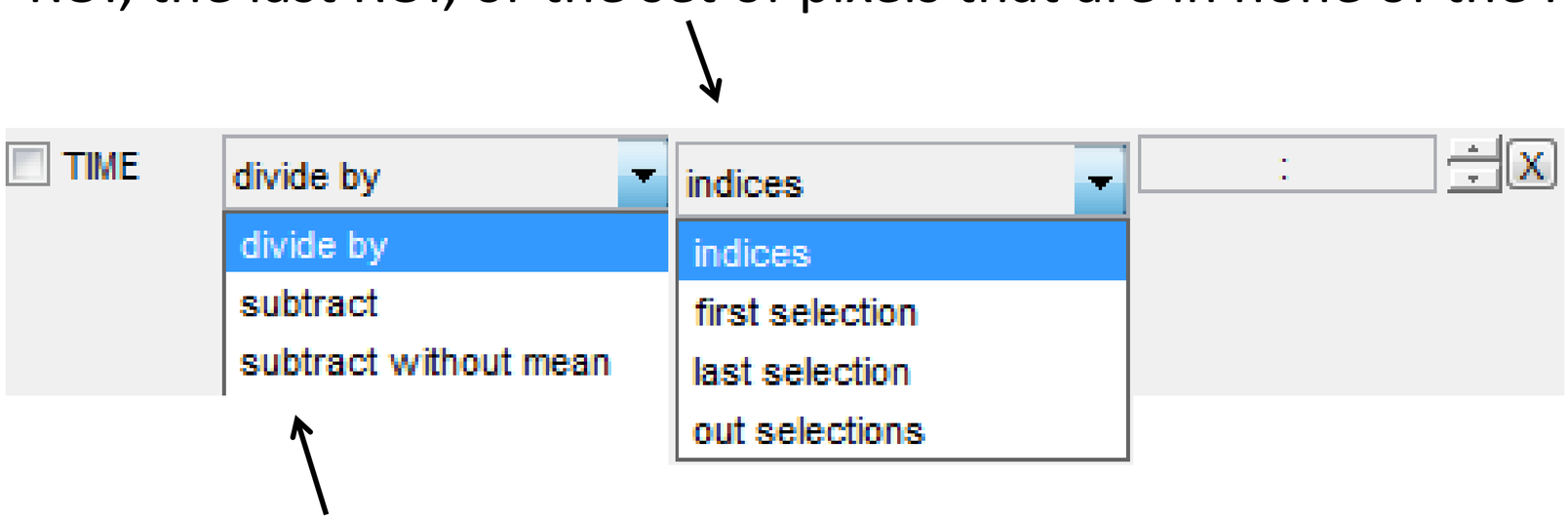
This average frame is then used to correct the movie in one of the 3 following ways:

- Each frame is divided by the average frame
- The average frame is subtracted from each frame
- The spatial pattern present in the average frame is subtracted from each frame, but not its mean value (this prevents the processed data to have all frames with their mean over all pixels equal to zero, i.e. the time courses of the data averaged over all pixels is preserved rather than being flattened to zero).

1)c “TIME” operation:
time courses normalization

Symmetrically to the “SPACE” operation, an average time courses is first calculated. There are 2 ways to define which pixels to use:

- Define a set of pixel indices (e.g. 1:10, indexed as in Matlab) , or use “:” to use all pixels
- Use the manually defined ROIs (see section 2d)): use either the first ROI, the last ROI, or the set of pixels that are in none of the ROIs.



This average time courses is used to correct the movie in one of the following ways:

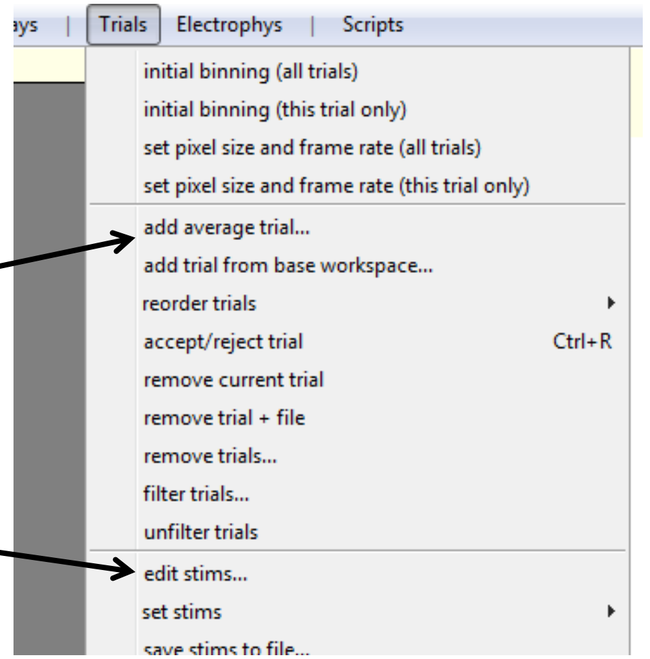
- The time courses of each pixel is divided by the average time courses
- The average time courses is subtracted to each pixel time courses
- The average time courses is subtracted but the mean pixel value over time is preserved (this results in the average movie frame to remain the same rather than being flattened to zero).

1)c Bleaching correction

Bleaching is modelled as a set of decaying exponentials. The time constants of these exponentials should be estimated once for all at the beginning (‘EST’ button).

Please follow the procedure below.

1) Time constants should be estimated on the average trial. First compute it in ‘Trials > add average trial…’

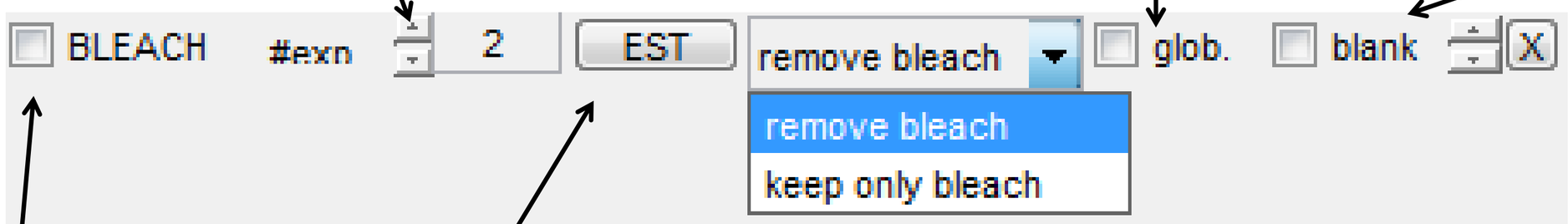


2) The algorithm needs to know which conditions are without stimulation: first do ‘Trials > edit stims…’ to set the ‘type’ of the appropriate conditions to ‘blank’.

3) Set the desired number of exponentials to use (typically 2 or 3).

4) Optional: check ‘glob’ if all trials are assumed to share exactly the same bleaching pattern.

Alternatively, check ‘blank’ to make estimations only from the blank trials to avoid the specific patterns of the responses to be fitted by the bleaching exponentials.



5) Make sure that the current trial showing is the average trial and press ‘EST’ to estimate the global parameters of the bleaching.

6) Now only activate the correction.

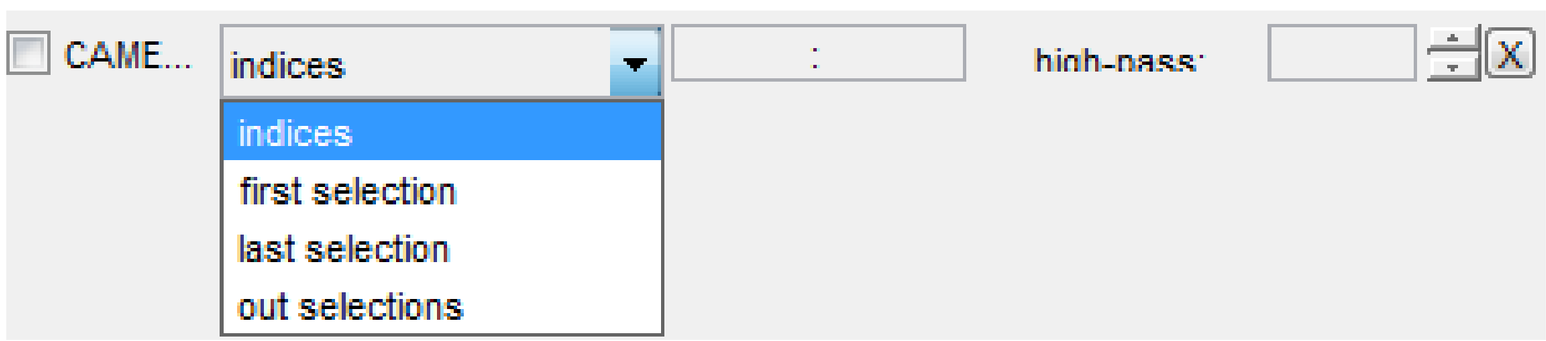
7) It can be useful to press ‘keep only bleach’ to show the estimated bleach pattern and check that it was properly estimated.

1)c Camera Noise removal

This operation can remove a specific camera noise which is common to all pixels.

Select a region which is rather dark in the image (in such a region the ratio between camera noise and physiological signals is higher) and press ‘last selection’: time courses from this region will be subtracted everywhere to correct the noise.

If the camera noise is known to be high-frequency, set a value for high-passing the time courses, and that way avoid to subtract lower-frequency physiologically meaningful signals.



1)c. Heart correction

Removal of heart beat artefact: the underlying method first estimates the phase of the heart pulsation, based on a selected sub-region that is strongly contaminated; then a general linear model estimates the specific shape of heart signals in every pixel.

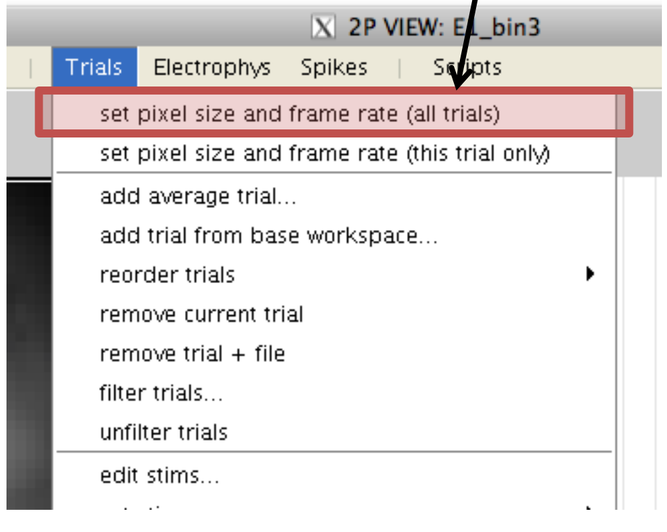
3) number of harmonics to use (1 would result in a sine wave estimation, more harmonics capture more high frequencies).

4) ‘precision’ parameter: the higher this parameter, the more fluctuations in the frequency are allowed.

5) ‘keep’ keeps only the estimated heart artefact: use this option to double-check that the estimation was accurate. Then select ‘rm’ to remove this artefact from the data.

1) Draw a region of interest on a blood vessel or any place in the image that has a lot of heart contamination. Then choose option ‘last selection’ to use this region for estimating the phase of the heart pulsation.

2) Give an approximative value of the heart frequency. Note that it is necessary that the frame duration has been defined accurately before (here).



1)c Motion correction

This coregisters and resamples the frames in a movie to correct for motion artefacts.
This computation can take time.
It is often preferable to create a whole new data set with all trials coregistered together from the ‘Trial > coregister trials’ menu rather than using this option.

The motion can be estimated from a sub-part of the image only. This can increase computation speed, or avoid misestimation due to high noises or high physiological signals in some parts of the image.

Limit to the amplitude of the motion, expressed as a fraction of the frame size

How many frames to take (counted from the first frame) to compute a reference image to which all frames will be aligned.

1)c Masking

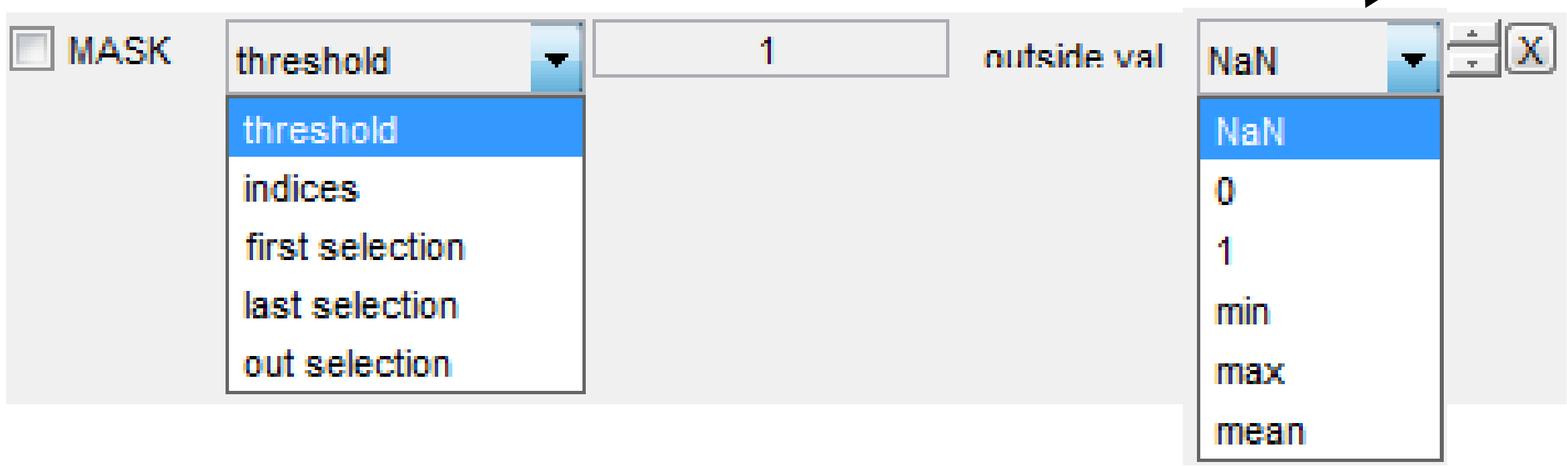
Keep only a part of the image.

Choose how to select the mask:

- All pixels whose average value is above a fixed threshold
- Manual selection of pixel indices
- Manual selection of a ROI

What value to use outside of the mask:

- A fixed value (NaN, 0 or 1)
- The minimum, maximum or average value inside the mask.



1)c Spatial and temporal filtering

Spatial low-pass and high-pass cutoff distances expressed in pixels

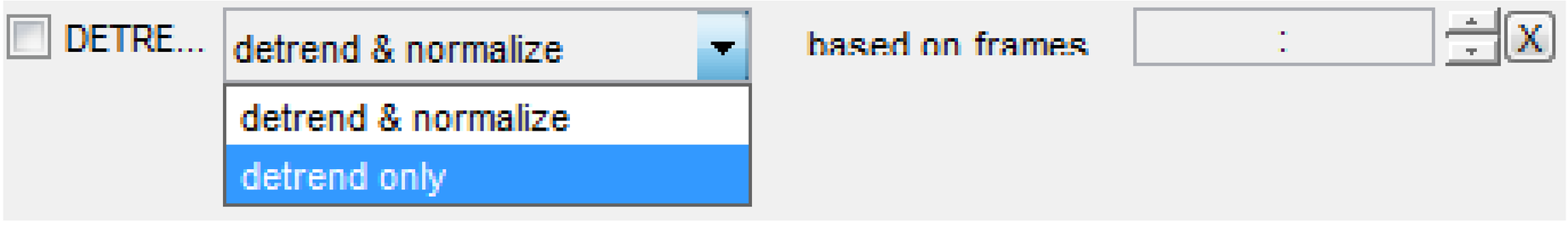
Temporal low-pass and high-pass cutoff periods expressed in seconds

Options:

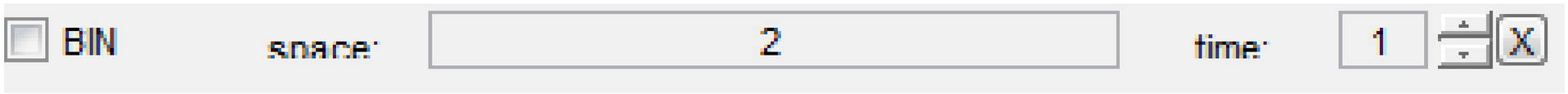
- d: Use a detrend to minimize edge effects of the temporal filter
- z: If a high-pass is applied, keep the mean rather than make the mean zero
- ph: (special) get the phase of the signal after temporal filter has been applied
- s: ‘sharp cut-off’ - the transfer function of the filter has a sharper cut-off frequency, but this can cause enhanced oscillations near this frequency

1)c Temporal Detrending

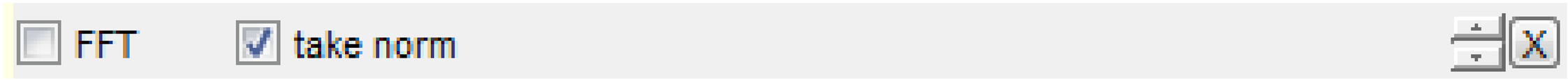
Remove a linear trend in the time courses. It is possible to specify specific frames from which this linear trend is estimated.



1)c Spatial and Temporal binning



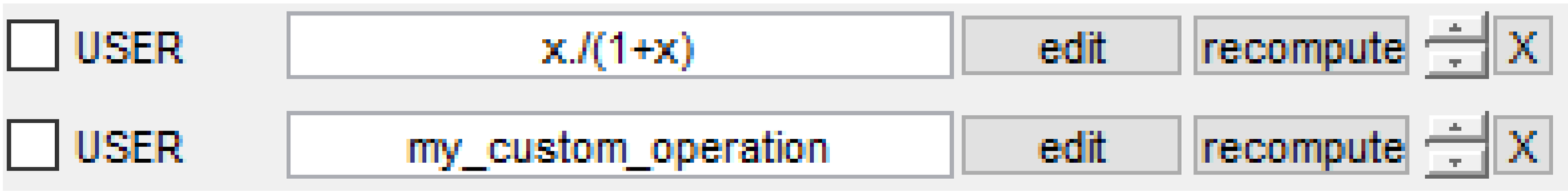
1)c Fourier Transform



Check ‘take norm’ to get the norm of the Fourier transform, otherwise the values remain complex (complex values can be visualized in the Image displays but not in the Time Courses displays)

1)c User-defined operation

Users can apply their own functions.
Define a function right inside the control using variable name “x” (1st example below), or write the name of the function to be applied (2nd example).



Press ‘edit’ to edit (or even create) the m-file.

OptImage might not well detect that the m-file content has changed. Press ‘recompute’ to force a function call.

2) Visualizing the data

a. Which displays are available?

b. What to show in each display?

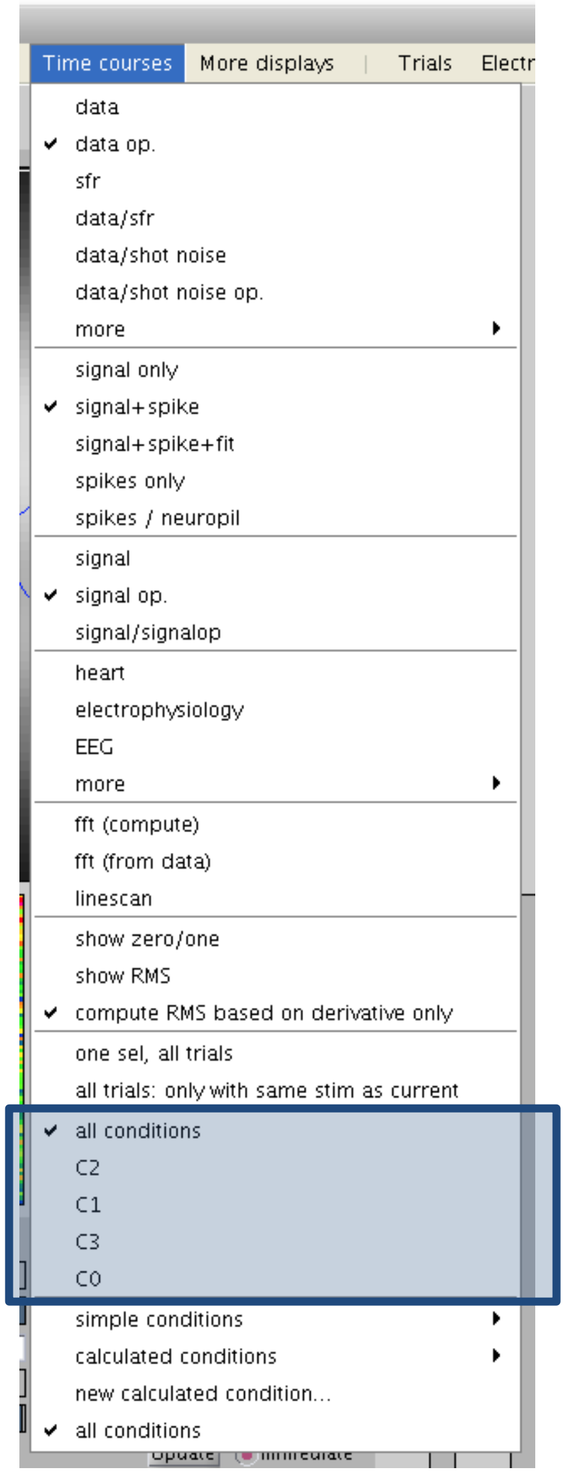
- How to select conditions

c. Display options

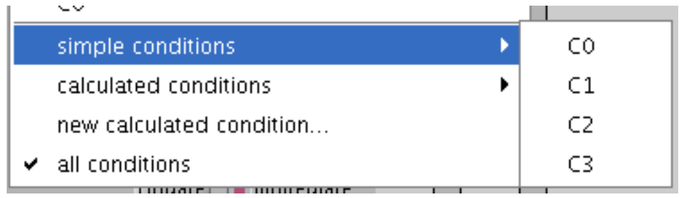
d. Mouse actions

2)b. Select condition(s)

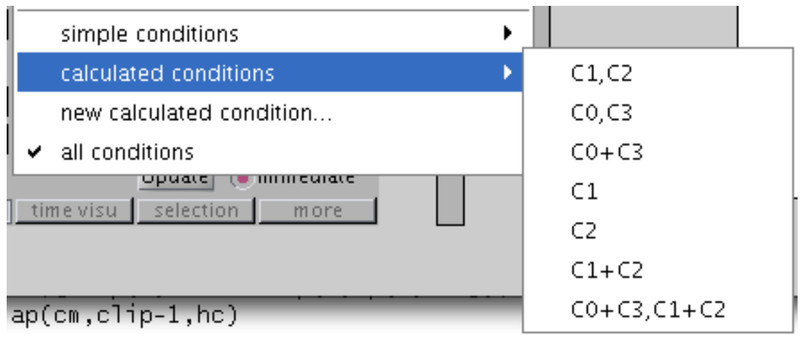
When data has mutiple conditions, it is possible to choose which to display in Picture1, Picture2 or Time Courses. Several options exist:



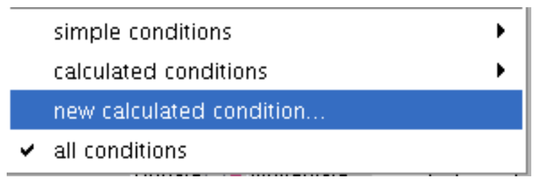
1) Choose one of the recently used conditions



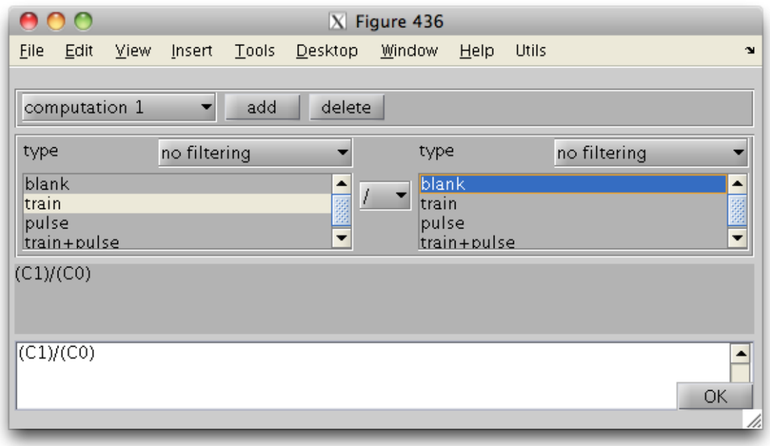
2) Choose a simple conditions

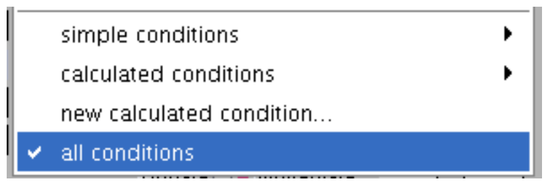


3) Choose a recent calculated condition



4) Create a new calculated condition: a new window opens that lets you define it





5) Display all simple conditions together

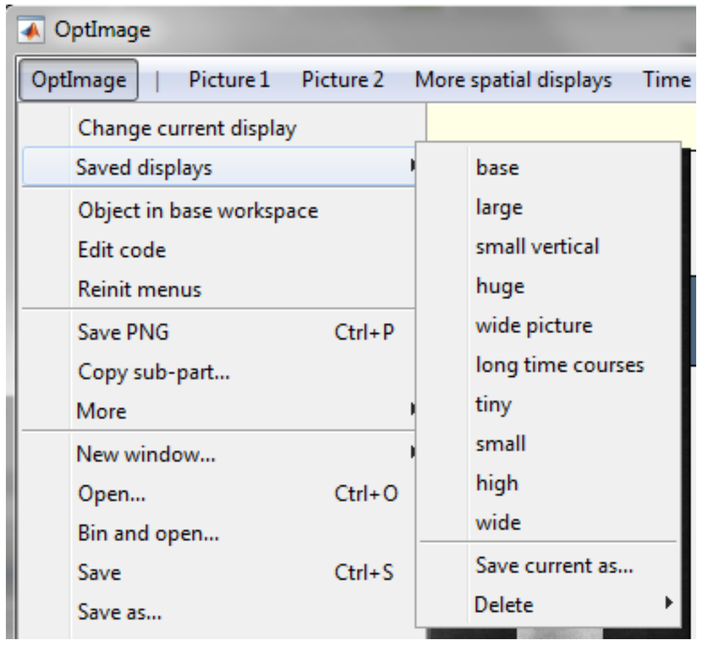
3) Other useful tools

a. The 2Pview menu

- Changing the program size and display
- Preferences

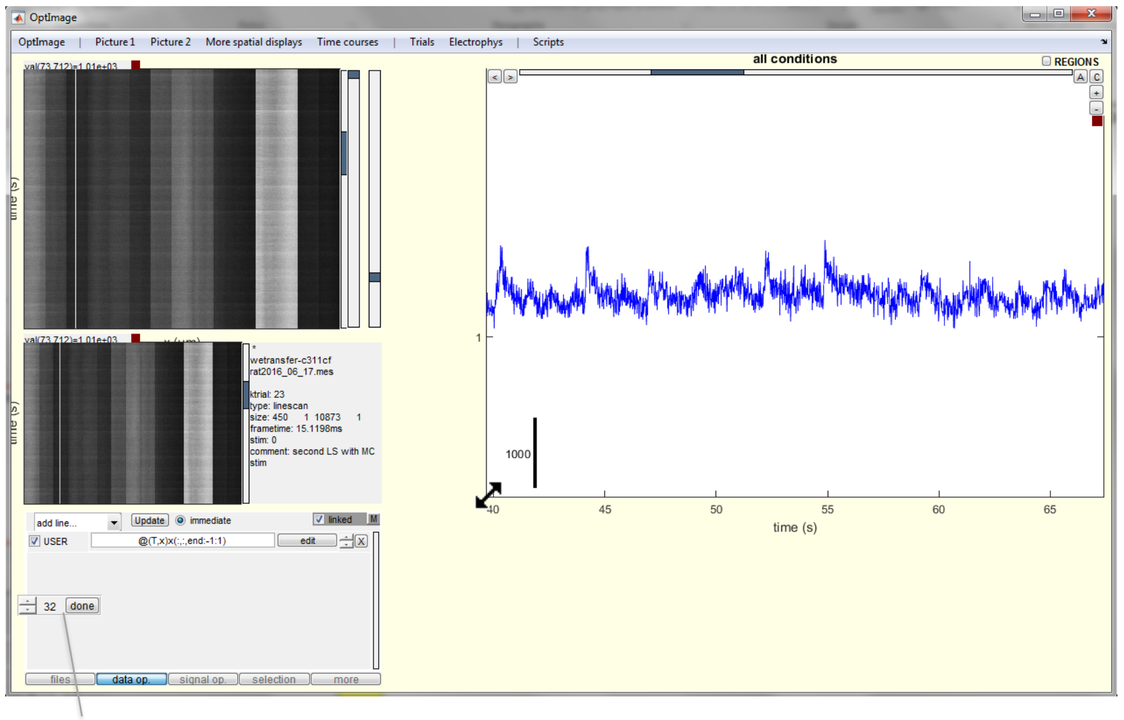
b. Access to the data from the command line

3)a. Changing the program size and display



A number of different **configurations** are pre-defined.
Select one of them to change the program size and display.

New configurations can be created with '*Change current display*' and then '*Save current as…*'.

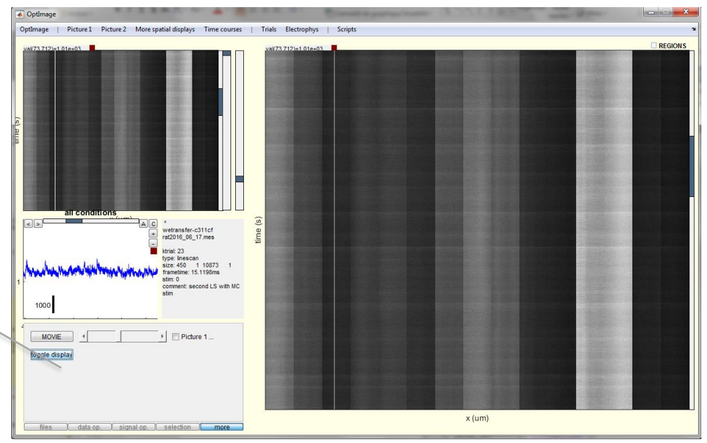


When '*Change current display*' is pressed, the different elements of the program and the figure itself become **resizeable**.

For easy alignment of the elements, position coordinates will be automatically rounded to a given number that can be adjusted (here: *32*).

- **move** elements **sides and corners** with mouse left button
- **drag** elements with mouse middle button
- **round** an element coordinates to the set number with a right-button click

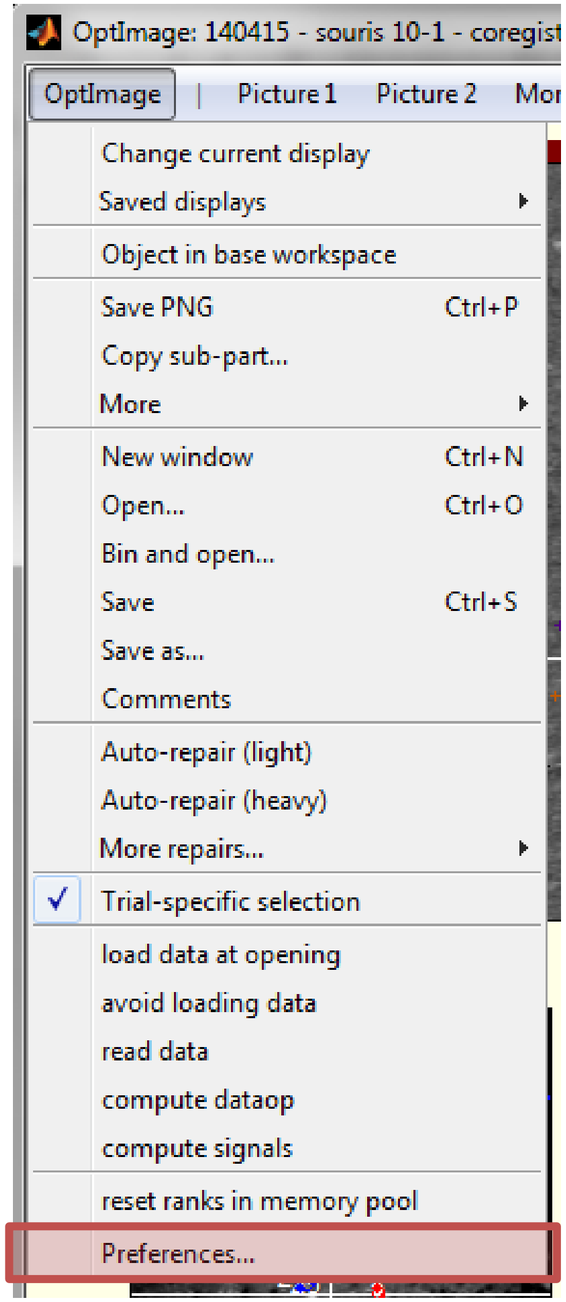
Press '*done*' once finished!

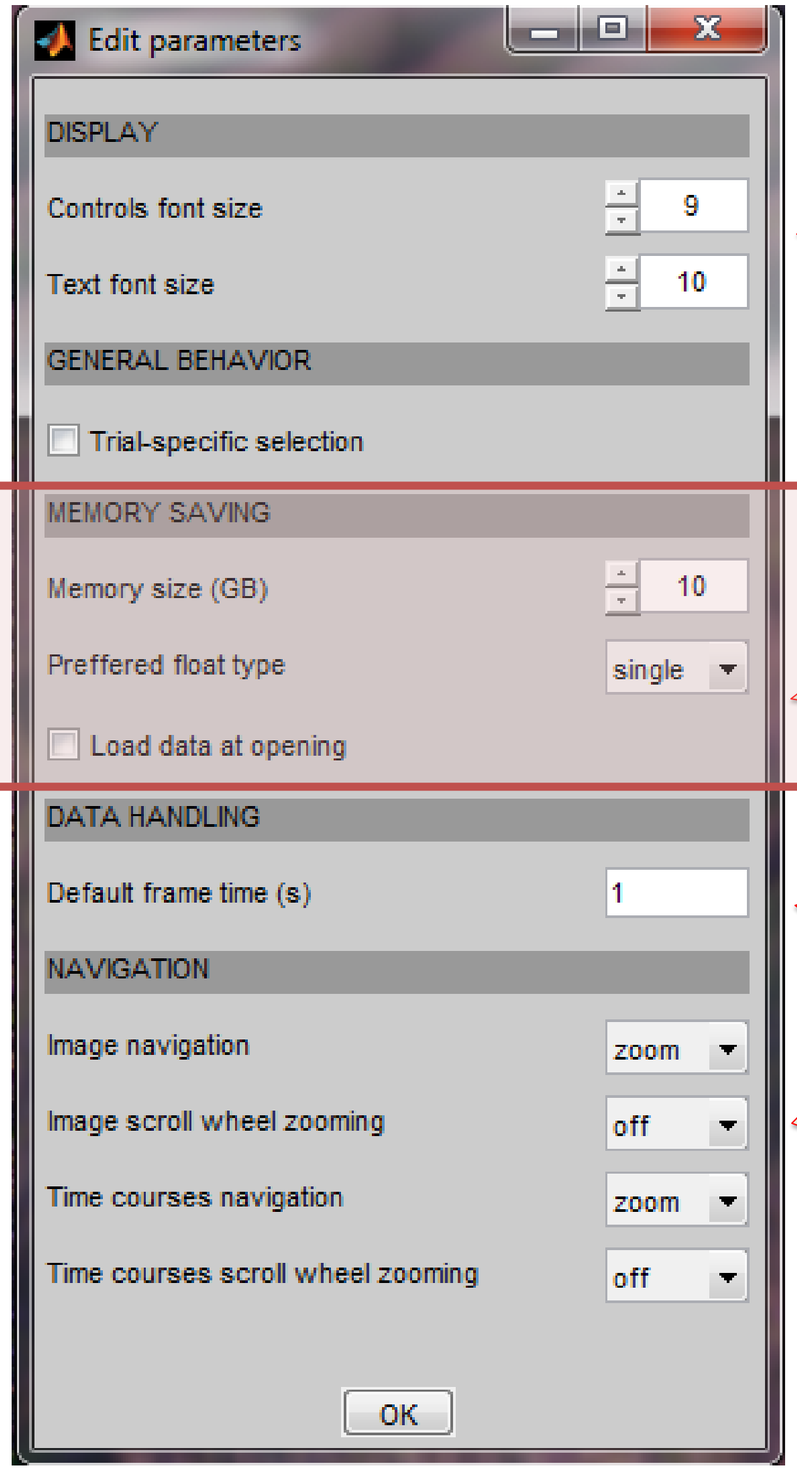


In the '*more*' panel, button '*toggle display*' allows to switch image and time courses display when one temporarily needs to see the images larger.

Finally, font sizes can be change in the '*Preferences*' (see next page).

3)a. Preferences





Change font sizes

If checked, the position of the ROIs can be different for different trials

Parameters to arbitrate between increasing speed and saving memory

If frame time is not set by user, some functions will assume this frame time

Choose between zoom and pan for image and time courses navigation. Decide whever it is possible to zoom in and out with the scroll wheel.

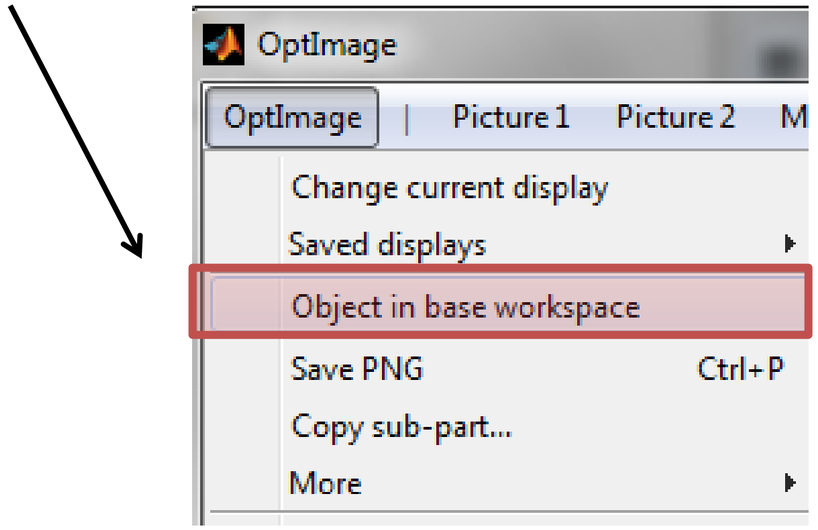
This section is very important.
Raw data and processed data are stored inside a 'pool' whose size is limited. When the pool is filled some data is forgotten and will need to be loaded/computed again when accessed again.
If Matlab runs out of memory, decrease the pool size; if data is permanently forgotten and recomputed, try increasing it.

Preferred float type should normally be set to 'double' to avoid numerical errors in computations. However in some circumstances, one can set it to 'single' to further save memory.

3)b. Access from command line

Variable ‘V’ contains all the information used by current OptImage window.

If ‘V’ has been cleared, or overwritten by another OptImage window, use this menu option to create it again.




```
>> V

V =

  tpview with properties:

            skin: 'OptImage'
          panels: [1x1 struct]
             a4d: [1x1 struct]
           timer: [1x1 timer]
         precomp: [1x1 struct]
         content: [1x1 tpv_content]
       internpar: [1x1 tpv_internpar]
         disppar: [1x1 struct]
      savingpar: [1x1 struct]
          ntrial: 64
           trial: [1x1 tps_trial]
            data: [414x434x1400 uint16]
          dataop: [414x434x1400 single]
             sfr: []
            file: [1x130 char]
        fullinfo: [1x1 struct]
           sizes: [414 434 1400 1]
              nx: 414
              ny: 434
             nfr: 1400
              nc: 1
         linedur: 0.0658
            tidx: [1x1400 double]
            nsel: 35
          signal: [1x1 tps_signal]
          ktrial: 2
         addinfo: [1x1 struct]
            type: 'movie'
        scanning: 1
              dx: 1.1921
              dy: 1.1921
              dz: 0
              dt: 28.5660
              t0: 0
              hf: 173.0052
            grob: [1x1 struct]
         options: [1x1 struct]
           menus: [1x1 struct]
      interfacepar: [1x1 struct]
```

The most important field is V.content

Number of trials
Current trial

Shortcuts to properties of the current trial

Number of ROIs
Shortcut to time courses
Index of current trial

Other fields are for internal use only

‘V.content’ contains all the data.
It is saved in a file with extension ‘tpv’. Load ‘tpv files as follows:
load(‘mydata.tpv’,’-MAT’)

```
>> V.content

ans =

  tpv_content with properties:

         version: 1.4100
          trials: [1x64 tps_trial]
         signals: [1x1 tps signal]
          ktrial: 2
      electrophys: [1x64 tps_electrophys]
            nsel: 35
              ij: [2x1 double]
     seldotrial: 1
       timeline: 0
         ntrial: 64
            nfr: 1400
             nx: 35
      datamodes: {'data'  ''}
       datacond: 'all conditions'
          trial: [1x1 tps_trial]
         signal: [1x1 tps_signal]
        channel: [1x1 tps_signalx]
      stimtable: [1x1 tps_stimtable]
        docfile: []
           user: [1x1 struct]
```

All the trials
All the time courses

User can store additional information in this structure.

‘V.content.trials(k)’ contains the information relative to trial k.

```
>> V.content.trials(1)

ans =

  tps_trial with properties:

          version: 1.7000
             file: [1x130 char]
         fileflag: ''
       analogfile: {[1x81 char]  [1]}
           origin: 'user [MESC header]'
            sizes: [414 434 1400 1]
           sizes0: [414 434 1400 1]
       sfrchannel: 0
         fullinfo: [1x1 struct]
          addinfo: [1x1 struct]
           status: 'n'
             type: 'movie'
         scanning: 1
             xbin: 1
             tbin: 1
               dx: 1.1921
               dy: 1.1921
               dz: 0
               dt: 28.5660
               t0: 0
            xunit: 'um'
            tunit: 'ms'
             user: [1x1 struct]
               nx: 414
               ny: 434
              nfr: 1400
               nc: 1
          linedur: 0.0658
             tidx: [1x1400 double]
           dt_sec: 0.0286
  usertransient: [1x1 struct]
      stimtable: [1x1 tps_stimtable]
      eventlist: [1x0 struct]
         stimid: 63873
           stim: [2x12 double]
     stimdetails: [1x1 struct]
          opdef: [1x1 tps_dataopdef]
          opmem: [1x0 tps_dataopdef]
          data0: [414x434 single]
           data: [414x434x1400 uint16]
            sfr: []
      shotnoise: [414x434x1400 single]
         dataop: [414x434x1400 single]
          sfrop: []
    shotnoiseop: [414x434x1400 single]
      dataopmem: [414x434x1400 single]
      recording: [1x1 struct]
     heartcycle: []
```

File where data is saved: note that the data is not read until it is actually used.

Trial status: ‘n’ormal, ‘r’ejected or ‘s’pecial

Main header information

Header information related to stimulation

Processing applied to the data

Average frame of the raw data

Raw data

Processed data

‘V.content.signal.x(k,i)’ contains the information relative to time courses for trial k and region of interest i.

```
>> V.content.signal

ans =

  tps_signal with properties:

          name: ''
      datamode: 'data'
      datacond: 'all conditions'
    dataopdef: [1x64 struct]
        opdef: [1x1 struct]
   seldotrial: 1
     timeline: 0
            x: [64x35 tps_signalx]
        shift: [64x2 double]
     spikepar: []
           nx: 35
          nfr: []
        nfrop: []
         nexp: 64
         data: {64x35 cell}
        data2: {64x35 cell}
       dataop: {64x35 cell}
      data2op: {64x35 cell}
          sel: [64x35 selectionND]
       spikes: {64x35 cell}
```

Processing applied to the signals

```
>> V.content.signal.x(1,1)

ans =

  tps_signalx with properties:

            tag: []
         active: 1
            sel: [1x1 selectionND]
       spikepar: []
          kcond: 1
          delay: 0.0141
     delayshown: 0.0141
           tidx: [1x1400 double]
           data: [1400x1 double]
          data2: []
         dataop: [1400x1 double]
        data2op: []
         spikes: []
       spikefit: []
     validspike: 0
        spikes2: []
      spikefit2: []
    validspike2: 0
```

Coordinates of the region of interest

Vector of time instants
Time courses before signal processing
Time courses of secondary data before signal processing
Time courses after signal processing
Time courses of secondary data after signal processing

Signals can be accessed also using the following method:
V.getsignals(ktrials,iregions), or V.getsignals(‘all’), V.getsignals(‘trial’), V.getsignals(‘sel’), V.getsignals(‘current’)