Table Of Contents

Published on *elan* (http://elan.lyon.inserm.fr)

Evoked Potential Processing Tools

epavg

- ***Description***

  Computes a grand average from several evoked potential files (**.p**) generally produced by eegavg . The output is a new evoked potential file (**.p**).

- ***Usage***

  epavg [+force]

  with :

  - option :

    +force : no test of compatibility on the electrode numbers.

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  grand average filename (no .p extension) or Return to quit :
  myoutputfile
  standard average (return), normalised (n), weighted (w)?

  Name of the .p file to average (no .p extension) or return to stop the list:
  myfile1
  Name of the .p file to average (no .p extension) or return to stop the list:
  myfile2
  Name of the .p file to average (no .p extension) or return to stop the list:
  myfile3
  Name of the .p file to average (no .p extension) or return to stop the list:

  grand average filename (no .p extension) :

- ***Fields of parameter file and example***

- ***Examples***

  This program will compute the average of myfile1, myfile2 and myfile3 and will store the results in myoutputfile. The number of channel, the number of samples, the analysis duration, the number of pre-stimulus samples and the sampling period must be the same in all input files.

- ***Comments***
  1. It should be noted that the ep file name are entered without the .p extension.
  2. The averaging options can be :

     Standard: ouput values are simple average values.
     Normalized: the values of each input files is divided by the maximum absolute value of this file, and normalized values are then averaged. The mean of the maximum values of all files is displayed at the end of the averaging: "..... mean of normalisation values = xx,xx micro-volts"

weighted: the values of each input file is multiplied by the actual number of trials averaged for this file. Then these new values are summed, and divided by the sum of trials across files.

3. Possible error messages:

   Incompatible epfile : p_xe[n]
   the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
   - 1 the number of channel,
   - 2 the number of samples,
   - 3 the epoch duration,
   - 4 the number of pre-stimulus samples,
   - 5 the sampling period.

- ***Current version***

  1.08 28-06-2012

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 01-02-2002 (PEA) : minor modification.
  - 1.02 21-01-2003 (PEA) : adds +force option : disables electrode compatibility test.
  - 1.03 10-11-2005 (PEA) : minor modification.
  - 1.04 13-08-2007 (PEA) : minor modification.
  - 1.05 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.06 24-01-2011 (PEA) : removes static allocation for reading EP file.
  - 1.07 05-03-2011 (PEA) : removes header check in reading EP file (libpem) due to Fortran compatibility problems.
  - 1.08 28-06-2012 (PEA) : adds epsilon when comparing headers for file compatibility.

- ***Files***

  $ELANPATH/bin/epavg

- ***See also***

  epdiff [1]

epavgchannel

- ***Description***

  Computes the average of group of channels of EP files (**.p**). The output is a new evoked potential file (**.p**).

- ***Usage***

  epavgchannel

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Number of channel groups :
  2
  Group 1 : number of channel to keep (0 for all) :
  2
  Indices of channels to keep (1st channel is #1) :
  1
  2
  Group 2 : number of channel to keep (0 for all) :
  0
  EP file name (without extension) to average channels or Return to quit :
  myfile1
  Output EP file name (without extension) :
  myfile1.avgelec
  EP file name (without extension) to average channels or Return to quit :

myfile2
Output EP file name (without extension) :
myfile2.avgelec
EP file name (without extension) to average channels or Return to quit :

- ***Fields of parameter file and example***

- ***Examples***

  For each file (*myfile1.p* and *myfile2.p*), the above example computes the average of channels 1 and 2, the average of all channels and creates a file with 2 channels (one for each average).
  It creates *myfile1.avgelec.p* (average of *myfile1.p* channels) and *myfile2.avgelec.p* (average of *myfile2.p* channels).

- ***Comments***
  1. It should be noted that the ep file name are entered without the .p extension.
  2. The output file's channel names are the first channel of each group.

- ***Current version***

  1.01 26-02-2013

- ***History***
  - 1.00 03-05-2011 (PEA) : 1st version.
  - 1.01 26-02-2013 (PEA) : fix error in output file (electrode number doesn't match 1st of group).

- ***Files***

  $ELANPATH/bin/epavgchannel

- ***See also***

epbline

- ***Description***

  Computes baseline correction of an evoked potential file from the same or a different evoked potential file. The output is a new evoked potential file (**.p**).

- ***Usage***

  epbline

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Name of the EP file to correct (no .p extension) or return to quit :
myfile1
Name of the EP file on which the baseline is computed (no .p extension) :
myfile2
Name of the corrected EP file (no .p extension) :
myoutfile
Start latency of baseline (in ms) :
-500
"Stop latency of baseline (in ms) :
-50
Name of the EP file to correct (no .p extension) or return to quit :

- ***Fields of parameter file and example***

- ***Examples***

This program will compute the baseline on myfile2, will correct myfile1 and will store the results in myoutfile. The number of channel, the number of samples, the analysis duration, the number of pre-stimulus samples and the sampling period must be the same in all input files.

- ***Comments***

1. It should be noted that the evoked potential file name are entered without the .p extension.
2. Possible error messages:

   Incompatible epfile : p_xe[n]
   the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
   - 1 the number of channel,
   - 2 the number of samples,
   - 3 the epoch duration,
   - 4 the number of pre-stimulus samples,
   - 5 the sampling period.

- ***Current version***

1.03 25-01-2011

- ***History***

  - 1.00 26-03-2002 (PEA) : 1st documented version.
  - 1.01 19-12-2006 (PEA) : minor modification.
  - 1.02 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 25-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

$ELANPATH/bin/epbline

- ***See also***

epchchannel

- ***Description***

  Modifies the channel number (following elec.dat nomenclature) of one channel in an ep file (**.p**) and creates a new **.p** file. This program only modifies the header part of the **.p** file. Data is not changed.

- ***Usage***

  epchchannel myfilein myfileout channel_rank channel_number

  with :

  - myfilein : input ep filename (potential data) (without extension).
  - myfileout : output filename (potential or SCD data) (without extension).
  - channel_rank : rank of the channel to modify.
  - channel_number : new channel number for the selected channel to modify. This value corresponds to the rank of the desired new channel name following the elec.dat nomenclature.

- ***Fields of parameter file and example***

- ***Examples***

  To modify the 3rd channel number and to replace its old channel number by 10 (electrode Cz in elec.dat having rank number 10).
  epchchannel file_in file_out 3 10

- ***Comments***

- ***Current version***

  1.02 25-01-2011

- ***History***
  - 1.00 13-05-2003 (PEA) : 1st version.
  - 1.01 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.02 25-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epchchannel

- ***See also***

epchref

- ***Description***

  Changes reference of channels of an EP file.

- ***Usage***

  epchref file_in.p file_out.p [chan1 refchan1 ... chanN refchanN] | [+avg]

  with :

  - file_in.p : input EP filename (with extension).
  - file_out.p : output EP filename (with extension).

  You can reference channel(s) to another channel(s) with:

  - chan1 : rank of 1st channel to change reference.
  - refchan1 : rank of reference channel for chan1.
  - ...
  - chanN : rank of Nth channel to change reference.
  - refchanN: rank of reference channel for chanN.

  Or, you can reference each channel to average of all channels by specifying option:

  - +avg

- ***Fields of parameter file and example***

- ***Examples***
  - Change reference of channel 2 to channel 3, channel 3 to 4 of file **myfilein.p** :

epchref myfilein.p myfilein.ref_2-3_3-4.p 2 3 3 4

- Change reference of all channels to average reference (average of all channels) of file **myfilein.p** :

  epchref myfilein.p myfilein.avg_ref.p +avg

- ***Comments***

- ***Current version***

  1.04 24-05-2012

- ***History***
  - 1.00 28-02-2001 (BY) : 1st version.
  - 1.02 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading EP file. Checks for channel rank and reference channel rank.
  - 1.04 24-05-2012 (PEA) : adds option for average reference.

- ***Files***

  $ELANPATH/bin/epchref

- ***See also***

epchwin

- ***Description***

  Modifies the total time-window of an EP file.

- ***Usage***

  epchwin file_in.p file_out.p nbsample_pre nbsample_post [value]

  with :

  - file_in.p : input EP filename (with extension).
  - file_out.p : output EP filename (with extension).
  - nbsample_pre : number of samples in the pre-stimulus period of the new file.
  - nbsample_post : number of samples in the post-stimulus period of the new file.
  - option :

    value : value for new samples. If omitted, the value for new samples is 0.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

  1.03 26-01-2011

- ***History***
  - 1.00 07-04-2006 (PEA) : 1st version.
  - 1.01 18-05-2006 (PEA) : minor modification.
  - 1.02 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epchwin

- ***See also***

epclean

- ***Description***

  Replace bad electrodes by spline interpolation (spline order 4, no regularization).

- ***Usage***

  epclean myepfile_in.p myepfile_out.p i1 i2 i3 ...

  with :

  - myepfile_in.p : input EP filename (with extension).
  - myepfile_out.p : output EP filename (with extension).
  - i1 : number of 1st electrode to replace by interpolation (following elec.dat).
  - i2 : number of 2nd electrode to replace by interpolation (following elec.dat).
  - i3 : number of 3rd electrode to replace by interpolation (following elec.dat).
  - ...

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

  1.06 26-06-2012

- ***History***
  - 1.00 04-04-2002 (BY) : 1st version.
  - 1.01 05-05-2005 (OB) : changes srk_size=200000 to be compatible with erpa, eegspline, epscd.
  - 1.02 28-03-2008 (PEA) : removes channels with indice -1 (don't exist in elec.dat).
  - 1.03 28-03-2008 (PEA) : uses elec_nomap.dat .
  - 1.04 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.05 26-01-2011 (PEA) : removes static allocation for reading EP file.
  - 1.06 26-06-2012 (PEA) : fixes an error when retreiving interpolated channel coordinates.

- ***Files***

  $ELANPATH/bin/epclean

- ***See also***

  epscd [2], eegspline [3]

epcumul

- ***Description***

  Cumulates an EP file (**.p**). Sample values are summed along time.

- ***Usage***

  epcumul myepfile_in myepfile_out

  with :

  - myepfile_in : input EP filename (without extension).
  - myepfile_out : output EP filename (without extension).

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

  1.03 26-01-2011

- ***History***
  - 1.00 21-10-2003 (PEA) : 1st version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epcumul

- ***See also***

epdelchan

- ***Description***

  Remove a channel list from an evoked potential file.

- ***Usage***

  epdelchan myEPfilein myEPfileout channel_number chan1 chan2 ...

  with :

  - myEPfilein : input EP filename (without extension).
  - myEPfileout : output EP filename (without extension).

- channel_number : number of channel to delete.
- chan1 : rank of the 1st channel channel to remove.
- chan2 : rank of the 2nd channel channel to remove.

- ***Fields of parameter file and example***

- ***Examples***

  To remove the 7th, 8th and 9th channel of an EP file (deletes 3 channels):
  epdelchan file_in file_out 3 7 8 9

- ***Comments***

- ***Current version***

  1.00 07-05-2014

- ***History***
  - 1.00 07-05-2014 (PEA) : 1st version.

- ***Files***

  $ELANPATH/bin/epdelchan

- ***See also***

epdetrend

- ***Description***

  Removes a best-fit line from an EP file (**;p**).

- ***Usage***

  epdetrend myepfile_in myepfile_out lat_beg lat_end

  with :

  - myepfile_in : input EP filename (without extension).
  - myepfile_out : output EP filename (without extension).
  - lat_beg : beginning latency for the best-fit line computation.
  - lat_end : ending latency for the best-fit line computation.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

1.03 26-01-2011

- ***History***
  - 1.00 29-05-2002 (PEA) : 1st version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 30-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epdetrend

- ***See also***

epdiff

- ***Description***

  Computes the difference between two evoked potential files (**.p**). The output is a new evoked potential file (**.p**).

- ***Usage***

  epdiff

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Name of the first .p file (no .p extension) or return to quit :
  myfile1
  Name of the .p file to be subtracted (no .p extension) :
  myfile2
  Name of the resulting .p file (no .p extension) :
  myoutfile
  Name of the first .p file (no .p extension) or return to quit :

- ***Fields of parameter file and example***

- ***Examples***

  This program will compute the difference between myfile1, myfile2 and will store the results in myoutputfile. The number of channel, the number of samples, the analysis duration, the number of pre-stimulus samples and the sampling period must be the same in all input files.

- ***Comments***
  1. It should be noted that the ep file name are entered without the .p extension.
  2. Possible error messages:

     Incompatible epfile : p_xe[n]
     the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
     - 1 the number of channel,
     - 2 the number of samples,
     - 3 the epoch duration,
     - 4 the number of pre-stimulus samples,
     - 5 the sampling period.

- ***Current version***

1.05 24-01-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 01-02-2002 (PEA) : minor modification.
  - 1.02 23-10-2003 (PEA) : minor modification.
  - 1.03 13-08-2007 (PEA) : minor modification.
  - 1.04 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.05 24-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epdiff

- ***See also***

  epavg [4]

epdiv

- ***Description***

  Computes the division between two evoked potential files (**.p**) file1.p/file2.p. The output is a new evoked potential file (**.p**).

- ***Usage***

  epdiv

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Name of the first .p file (no .p extension) or return to quit :
  myfile1
  Name of the second .p file (no .p extension) :
  myfile2
  Name of the resulting .p file (no .p extension) :
  myoutfile
  Name of the first .p file (no .p extension) or return to quit :

- ***Fields of parameter file and example***

- ***Examples***

  This program will compute the division between myfile1, myfile2 (myfile1.p/myfile2.p) and will store the results in myoutputfile. The number of channel, the number of samples, the analysis duration, the number of pre-stimulus samples and the sampling period must be the same in all input files.

- ***Comments***
  1. It should be noted that the ep file name are entered without the .p extension.
  2. Possible error messages:

     Incompatible epfile : p_xe[n]
     the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
     - 1 the number of channel,
     - 2 the number of samples,
     - 3 the epoch duration,
     - 4 the number of pre-stimulus samples,
     - 5 the sampling period.

- ***Current version***

  1.03 24-01-2011

- ***History***
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 24-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epdiv

- ***See also***

epfiltfilt

- ***Description***

  Computes zero-phase filtering (forward and reverse) on evoked potential channels of an EP file, and creates a new EP file. As input parameters, it can use B and A coefficients computed with Matlab (or SciPy, Octave, Scilab), or it can compute Butterworth filter coefficients from cutoff frequencies. It can compute band-pass, band-stop, low-pass or high-pass filtering.
  When computing Butterworth coefficients, a test for filter stability is processed. If the filter is unstable, no filtering is computed and a message for the new filter order to apply is displayed.

- ***Usage***

  epfiltfilt myepfilein.p myparameterfile.par myepfileout.p

  with :

  - myepfilein.p : input EP file to filter (with extension).
  - myparameterfile.par : filtering parameter file (with extension).
  - myepfileout.p : output (filtered) EP file (with extension).

- ***Fields of parameter file and example***

filter_channel 1 1 0 1 0	List of the channels to filter: 1/0 for selected/unselected channels; the total number of flags is N, N being the number of recorded channels in myepfilein.p file. In this example, N=5, and only channels number 1, 2, 4 will be filtered but all channels are stored in the output files.
filter_type 0	Type of Butterworth filter. Valid values are : 0 for low-pass filter 1 for high-pass filter 2 for band-pass filter 3 for band-stop filter In this example, a low-pass filter is processed.
filter_order 3	Butterworth filter order. In this example, the filter order is 3.
filter_cutoff_freq1 30	Cutoff frequency 1. In case of filter type : 0 (low-pass) filter : attenuates higher frequencies. 1 (high-pass) filter : attenuates lower frequencies. 2 (band-pass) filter : lower frequency of the range of frequencies to pass. 3 (band-stop) filter : lower frequency of the range of frequencies to attenuate.

filter_cutoff_freq2 100	Cutoff frequency 2. Needed only for band-pass and band-stop filter (filter_type 2 or 3). In case of filter type : 2 (band-pass) filter : higher frequency of the range of frequencies to pass. 3 (band-stop) filter : higher frequency of the range of frequencies to attenuate.
filter_b_coeff_nb 4	Number of B coefficients (in the Matlab way). These coefficients can be of a Butterworth filter, or any other filter.
filter_b_coeffs 2.71835675758059e-05 8.15507027274176e-05 8.15507027274176e-05 2.71835675758059e-05	List of B coefficients. **As it is text file, be careful of include a maximum of significant digits.** In this example, coefficients are corresponding to a low-pass Butterworth filter of order 3.
filter_a_coeff_nb 4	Number of A coefficients (in the Matlab way). These coefficients can be of a Butterworth filter, or any other filter.
filter_a_coeffs 1 -2.87730072411486 2.76201379931893 -0.884495606663461	List of A coefficients. **As it is text file, be careful of include a maximum of significant digits.** In this example, coefficients are corresponding to a low-pass Butterworth filter of order 3.

Use parameters when working with a Butterworth filter and setting filter order, type and cutoff frequency(ies).
Use parameters when working with a coefficients computed by another way.

- ***Examples***
  1. This parameter file (lowpass_butter_10hz_ep.par [5]) computes Butterworth coefficients from the filter definition, and filters all channels of an EP file containing 8 channels:

```
#low_pass Butterworth filter
filter_type 0

filter_order 3

filter_cutoff_freq1 10

filter_channel 1 1 1 1 1 1 1 1
```

  2. This parameter file (lowpass_butter_10hz_coefs_ep.par [6]) filters all channels of an EP file containing 8 channels, with a Butterworth filter of same characteristics designed in Matlab (with butter function):

```
filter_channel 1 1 1 1 1 1 1 1

#low_pass Butterworth filter designed with Matlab butter function
filter_b_coeff_nb 4

filter_b_coeffs
2.71835675758059e-05    8.15507027274176e-05    8.15507027274176e-05
2.71835675758059e-05

filter_a_coeff_nb 4

filter_a_coeffs
1       -2.87730072411486       2.76201379931893 -0.884495606663461
```

  3. This parameter file (bp_2-30Hz_ep.par [7]) computes a band-pass (2-30Hz) Butterworth filter, and filters all channels of an EP file containing 8 channels:

```
#band_pass
filter_type 2

filter_order 3

filter_cutoff_freq1 2
filter_cutoff_freq2 30

filter_channel 1 1 1 1 1 1 1 1
```

- ***Comments***

- ***Current version***

  1.01 05-09-2011

- ***History***
  - 1.00 17-11-2010 (PEA) : 1st version (includes test for filter stability when computing Butterworth coefficients).
  - 1.01 05-09-2011 (PEA) : changes initial conditions computation (fits to Matlab).

- ***Files***

  $ELANPATH/bin/epfiltfilt

- ***See also***

  eegfiltfilt [8], eegepochfiltfilt [9]

Attachment	Size
lowpass_butter_10hz_ep.par [10]	116 bytes
lowpass_butter_10hz_coefs_ep.par [11]	332 bytes
bp_2-30Hz_ep.par [12]	120 bytes

epfromtxt

- ***Description***

  Converts ASCII file with multiplexed data to EP file (**.p**).

- ***Usage***

  epfromtxt ascii_file.txt parameterfile.par myepfile_out.p

  with :

  - ascii_file.txt : input text file to convert (with extension).
  - parameterfile.par : conversion parameter file (with extension).
  - myepfile_out.p : output EP filename (with extension).

- ***Fields of parameter file and example***

nchan 28	Number of channels.
nsamp 1000	Total number of samples. The total number of samples of the analysis is (number of samples in the prestimulus) + (number of samples in the poststimulus) + 1, the extra sample corresponds to the event itself.
prestim_nsamp 200	Number of samples in prestimulus.
sampling_period_msec 1.0	Sampling period (in ms). In this example, the sampling period is 1 ms, which corresponds to a sampling frequency of 1000 Hz.
erpa_positivity 1	Value stored in the output .p file to set the default upward/downward positivity orientation for curve display (by erpa): -1 (positivity down) or 1 (positivity up)
electrodes 42 43 14 15 16	

48 40 41 49 32 8 9 10 11 12 33 50 13 17 51 3 4 5 6 7 1 2 39	List of channel number (indices in elec.dat).

In the example above, the electrodes list corresponds to the following labels and positions :

name	# in elec.dat	r (mm)	theta (deg)	phi (deg)
CP1	42	90.000000	31.500000	223.100006
CP2	43	90.000000	31.500000	316.899994
F3	14	90.000000	63.099998	136.899994
Fz	15	90.000000	45.000000	90.000000
F4	16	90.000000	63.099998	43.099998
FT3	48	90.000000	75.599998	159.699997
FC1	40	90.000000	31.500000	136.899994
FC2	41	90.000000	31.500000	43.099998
FT4	49	90.000000	75.599998	20.299999
M1	32	90.000000	120.000000	215.600006
T3	8	90.000000	90.000000	180.000000
C3	9	90.000000	45.000000	180.000000
Cz	10	90.000000	0.000000	0.000000
C4	11	90.000000	45.000000	0.000000
T4	12	90.000000	90.000000	0.000000
M2	33	90.000000	120.000000	324.399994
TP3	50	90.000000	75.599998	200.300003
F7	13	90.000000	90.000000	144.000000
F8	17	90.000000	90.000000	36.000000
TP4	51	90.000000	75.599998	339.700012
T5	3	90.000000	90.000000	216.000000
P3	4	90.000000	63.099998	223.100006
Pz	5	90.000000	45.000000	270.000000
P4	6	90.000000	63.099998	316.899994
T6	7	90.000000	90.000000	324.000000
O1	1	90.000000	90.000000	252.000000

O2	2	90.000000	90.000000	288.000000
Fpz	39	90.000000	90.000000	90.000000

- ***Examples***

- ***Comments***

- ***Current version***

  1.02 29-09-2010

- ***History***
  - 1.00 30-03-2000 (PEA) : 1st version.
  - 1.01 ? (PEA) : minor modification.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan. Reads electrode numbers as int instead of float.

- ***Files***

  $ELANPATH/bin/epfromtxt

- ***See also***

epkruskal

- ***Description***

  Computes the Kruskal-Wallis (non-parametric test) test between several groups (conditions) of EP files (**.p**). The output are statistical EP files. The statistical procedure is described in “Non Parametric Statistics for the Behavioural Sciences” by S. Siegel, McGraw Hill, 1956, p.184.

- ***Usage***

  epkruskal [+fdr]
  with :

  - option :
    +fdr : computes False Detection Rate (FDR) after the statistical test. It creates 2 EP files with a mask of the statistical results (H and p) with the computed FDR. This requires a probability (p) threshold value.

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
FDR threshold :
0.05
It defines the probability (p) threshold value to be used for masking the statistical results (H) by the threshold FDR statistics. Only if +fdr option is specified.
Number of conditions :
3
Number of files in condition 1 :
10
Number of files in condition 2 :
15
Number of files in condition 3 :
12
Input file name (with extension) 1 in condition 1 :
epfile1cond1.p
Input file name (with extension) 2 in condition 1 :
epfile2cond1.p
...

Input file name (with extension) 1 in condition 2 :
epfile1cond2.p
...
Input file name (with extension) 12 in condition 3 :
epfile12cond3.p
Output file name (without extension) :
out_ep.kw

- ***Fields of parameter file and example***

- ***Examples***

The example above will create the following files :
out_ep.kw.H.p : **.p** file of the H value of the Kruskal-Wallis test computed on the 3 conditions.
out_ep.kw.p.p : **.p** file of the probability value (p) of the Kruskal-Wallis test computed on the 3 conditions.

- ***Comments***

1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).

- ***Current version***

1.04 29-06-2012

- ***History***

  - 1.00 09-04-2005 (OB) : 1st documented version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading and writing EP files.
  - 1.04 29-06-2012 (PEA) : adds FDR correction option.

- ***Files***

$ELANPATH/bin/epkruskal

- ***See also***

epwilcox [13], epquade [14], epwinquade [15]

epmask

- ***Description***

  Masks evoked potential file (**.p**) with another evoked potential file. The output is a new evoked potential file (**.p**).

- ***Usage***

  epmask file_mask.p file_data.p file_data_out.p threshold flag new_value

  with :

  - file_mask.p : input EP filename (with extension) used for masking.
  - file_data.p : input EP filename (with extension) to mask.
  - file_data_out.p : output EP filename (with extension) containing masked data.
  - threshold : threshold threshold value.
  - flag : choose value to mask :

    1: masks values >= threshold.

0: masks values <= threshold.
2: masks values = threshold.

- new_value : value used for masking. It replaces the masked data.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

  1.04 15-05-2014

- ***History***
  - 1.00 29-05-2002 (PEA) : 1st documented version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 26-01-2011 (PEA) : removes static allocation for reading EP files.
  - 1.04 15-05-2014 (PEA) : add threshold type 2 (=).

- ***Files***

  $ELANPATH/bin/epmask

- ***See also***

epmeasure

- ***Description***

  Experimental feature. Description will come soon...

- ***Usage***

  epmeasure

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. It should be noted that all the EP file name are entered with the .p extension.

- ***Current version***

1.05 17-03-2016

- ***History***
  - 1.00 10-04-2013 (PEA) : 1st documented version.
  - 1.01 26-09-2013 (PEA) : ask for saving data to file text. Cosmetic modification in output.
  - 1.02 27-11-2013 (PEA) : change question for channel factor (intrasubject factor).
  - 1.03 09-10-2014 (PEA) : change question order.
  - 1.04 10-03-2016 (PEA) : fix error when only one factor (intrasubject).
  - 1.05 17-03-2016 (PEA) : fix error in latency measures.

- ***Files***

$ELANPATH/bin/epmeasure

- ***See also***

epmedian

- ***Description***

  Computes median value on EP files (**.p**) for each channel and sample.

- ***Usage***

  epmedian

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Number of EP files :
10
EP file name # 1 :
epfile1.p
EP file name # 2 :
epfile2.p
...
EP file name # 10 :
epfile10.p
Output EP file :
medianfile.p

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. It should be noted that all the EP file name are entered with the .p extension.

- ***Current version***

1.02 26-01-2011

- ***History***
  - 1.00 14-11-2001 (BY) : 1st documented version.
  - 1.01 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.02 26-01-2011 (PEA) : removes static allocation for reading and writing EP files.

- ***Files***

$ELANPATH/bin/epmedian

- ***See also***

epnorm

- ***Description***

  Normalizes individual **.p** files from several paired groups (conditions) of EP files (**.p**). The output are normalised **.norm.p** files.

- ***Usage***

  epnorm myparameterfile.par mycondfile.cond

  with :

  - myparameterfile.par : conversion parameter file (with extension).
  - mycondfile.cond : ascii file including the list of **.p** to process (see below).

- ***Fields of parameter file and example***

epnorm_channel_flag 1 1 0 0 1	List of the channels to normalize: 1/0 for selected/unselected channels.
epnorm_time_hw 50	Half-window size (in ms) of smoothing window. If the value is set to 0, no smoothing is applied, and all samples are computed.
epnorm_time_step 50	Time step (in ms) of smoothing window.
epnorm_normalization 1	Normalization type. Possible values are : 0: no normalization 1: for each subject, each sample is divided by the norm of the vector in channel □ condition space 2: for each subject and each condition, each sample is divided by the norm of the vector in channel space 3: for each condition and sample, each sample is divided by the norm of the mean vector (across subjects) in channel space 4, 5, 6: same as 1, 2, 3 except that the maximum norm in the analysis time-window is taken
epnorm_lat_beg_ms 100	Beginning latency for normalization (in ms).
epnorm_lat_end_ms 300	Ending latency for normalization (in ms).

The condition file is a text file formatted as follow:

Parameter (example)	Comment
1 2 4	Number of factors (must be 1). Number of conditions for this factor (must be 2). Number of **.p** files for each condition.
fact1 cond1 suj1 Myfile_cond1_suj1.p fact1 cond1 suj2 Myfile_cond1_suj2.p fact1 cond1 suj3 Myfile_cond1_suj3.p fact1 cond1 suj4 Myfile_cond1_suj4.p fact1 cond2 suj1 Myfile_cond2_suj1.p fact1 cond2 suj2 Myfile_cond2_suj2.p	List of **.p** files preceded by the following labels: name of the factor name of the condition name of the subject name of the **.p** file Each file in condition 1 has its paired file in condition 2. For one-sided tests, the condition with the predicted most positive mean must be placed first.

fact1 cond2 suj3 Myfile_cond2_suj3.p fact1 cond2 suj4 Myfile_cond2_suj4.p	

- ***Examples***

- ***Comments***
  1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).
  2. Note that this program may use epranddiff [16] parameters (eprand_channel_flag, eprand_time_hw, eprand_time_step, eprand_normalization, eprand_lat_beg_ms, eprand_lat_end_ms) instead of its own parameters.

- ***Current version***

  1.07 14-08-2012

- ***History***
  - 1.00 21-03-2006 (PEA) : 1st version.
  - 1.01 23-03-2006 (JB) : adds maximum norm, smoothing window.
  - 1.02 29-03-2006 (JB) : normalizes smoothed values. Adds parameter file. Reads epranddif parameters.
  - 1.03 23-03-2006 (JB) : fixes error with subsampling.
  - 1.04 22-05-2006 (JB) : fixes error with subsampling (time 0), improves test of compatibility of files.
  - 1.05 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.06 26-01-2011 (PEA) : removes static allocation for reading and writing EP files.
  - 1.07 14-08-2012 (PEA) : fixes error when reading files 2nd time (1st for compatibility check, 2nd for data). File name was freed after 1st read.

- ***Files***

  $ELANPATH/bin/epnorm

- ***See also***

  epranddiff [16]

epquade

- ***Description***

  Computes the Quade test (non-parametric test) on EP files (**.p**). It allows to compare many conditions with paired subjects.
  The Quade test is a nonparametric two-way analyses of variance. It is equivalent to an ANOVA with 1 factor, paired subjects and many conditions.

- ***Usage***

  epquade

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Number of conditions :
3
Number of files per condition :
10
Conover 2 by 2 comparisons : probability threshold (e.g. 0.01) :
0.01
Condition 1 | file 1 EP file name (without extension) :

epfile1cond1
Condition 1 | file 2 EP file name (without extension) :
epfile2cond1
...
Condition 2 | file 1 EP file name (without extension) :
epfile1cond2
...
Condition 3 | file 10 EP file name (without extension) :
epfile10cond3

Output file prefix for p, F and Conover values (without extension) :
quade_ep

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).
2. In output Conover Ep files, only significant values are stored in file (positive values). The others are replaced by 1.

- ***Current version***

1.05 14-04-2011

- ***History***

  - 1.03 28-01-2011 (PEA) : 1st documented version.
  - 1.04 12-04-2011 (PEA) : minor modification (adds "without extension" when asking for filename).
  - 1.05 14-04-2011 (PEA) : fixes an error on file names (may crash).

- ***Files***

$ELANPATH/bin/epquade

- ***See also***

epwilcox [13], epkruskal [17], epwinquade [15]

epranddiff

- ***Description***

  Performs a statistical randomization test between several paired groups (conditions) of ep files (*.p). The output is a statistical *.p file
  This program basically creates a **.rand.p.p** file representing the probability values p that the actual difference between two conditions is found by chance (the distribution of differences under the null hypothesis is estimated by randomly permuting conditions), independently computed for each channel over each moving window. They can be viewed by erpa as curve plots.

  Optionally, this program can take into account the number of samples of the window of analysis and correct for multiple tests in different ways:
  - The minimum number of consecutive significant samples (at a given probability threshold proba_measure) appearing in permutations with a given probability threshold proba_compute is computed independently for each channel (i.e. the maximum number of significant samples appearing with a given probability under the null hypothesis).
  - The maximum significance probability appearing in permutations with a given probability threshold proba_compute (i.e. the minimum significance probability appearing with a given probability under the null hypothesis) is computed

across samples (i.e. independently for each channel) or accross channels and electrodes.
The output files for these options are 2 statistical **.p** files (uncorrected and corrected) and a **.txt** file including the correction value for each channel.

- ***Usage***

  epranddiff myparfile.par mycondfile.cond nb_permut
  with :

  - myparfile.par : parameter file (with extension).
  - mycondfile.cond : ascii file including the list of **.p** to process (with extension).
  - nb_permut : number of permutations that are made to estimate the distribution of differences under the null hypothesis (example: 10000 drawings enables a resolution of p=0.001).

- ***Fields of parameter file and example***

fileprefix myfilename	Prefix of the output **.p** files.
eprand_channel_flag 1 1 0 1 0	List of the channels to process: 1/0 for selected/unselected channels. In this example, N=5, and only channels number 1, 2, 4 will be processed.
eprand_time_hw 10	Time half-window (in ms), used to define the moving time-window on which statistical analysis will be performed.
eprand_time_step 5	Time step (in ms), used to define the moving step of the moving time-window on which statistical analysis will be performed.
eprand_normalization 1	Normalization type. Possible values are : 0: no normalization 1: for each subject, each sample is divided by the norm of the vector in channel □ condition space 2: for each subject and each condition, each sample is divided by the norm of the vector in channel space 3: for each condition and sample, each sample is divided by the norm of the mean vector (across subjects) in channel space 4, 5, 6: same as 1, 2, 3 except that the maximum norm in the analysis time-window is taken
eprand_measure 2	Statistic on which the permutation test will be made 0: Sum of squared Student T across electrodes 1: Sum of squared sums of values by conditions (two-sided test) 2: Sum of values in the first condition (one-sided test)
eprand_compute 1	0: no multiple test correction 1: multiple test correction on the minimum consecutive samples 2: multiple test correction on the maximum significance across samples 3: same as 2 across samples and channels
eprand_lat_beg_ms 100	Start latency of the window in which the test will be performed (in milliseconds).
eprand_lat_end_ms 300	End latency of the window in which the test will be performed (in milliseconds).
eprand_proba_measure 0.01	Statistical probability threshold for the computation of significant samples when eprand_compute is set to 1.
eprand_proba_compute 0.05	Statistical probability threshold for the computation of the minimumnumber of consecutive significant sample when eprand_compute is set to 1 or the maximum probability when eprand_compute is set to or 3.

The condition file is a text file formatted as follow:

Parameter (example)	Comment
1 2 4	Number of factors (must be 1). Number of conditions for this factor (must be 2). Number of **.p** files for each condition.
fact1 cond1 suj1 Myfile_cond1_suj1.p fact1 cond1 suj2 Myfile_cond1_suj2.p	

fact1 cond1 suj3 Myfile_cond1_suj3.p fact1 cond1 suj4 Myfile_cond1_suj4.p fact1 cond2 suj1 Myfile_cond2_suj1.p fact1 cond2 suj2 Myfile_cond2_suj2.p fact1 cond2 suj3 Myfile_cond2_suj3.p fact1 cond2 suj4 Myfile_cond2_suj4.p	List of **.p** files preceded by the following labels: name of the factor name of the condition name of the subject name of the **.p** file Each file in condition 1 has its paired file in condition 2. For one-sided tests, the condition with the predicted most positive mean must be placed first.

- ***Examples***

In the above example, two paired groups of 4 files are compared. They will be processed on a time window ranging from 100 ms post-stimulus to 300 ms post-stimulus. The analysis will be performed on the sum of values in condition one averaged over a moving window having 10 ms of half duration (total duration of 20 ms), moving by step of 5 ms, on channels 1, 2 and 4. As correction for multiple comparison is on, the minimum number of consecutive significant (with p = 0.01) samples appearing with a probability of 0.05 across permutations will be computed.
The output files are :

Name	Comments
myfilename.rand.p.p	p file of the p values of the randomization test.
myfilename.n_max.txt	For the multiple-test correction across samples. List of the maximum number of successive significant samples n_max (or the minimum probability p_min) under the null hypothesis for each channel when eprand_compute is set to 1 or 2.
myfilename.rand.mask.p.p	For the multiple-test correction. Only p values included in a succession of samples that is above the computed minimum number of significant samples (or smaller than the p_min value) are kept. The other samples are set to p=1.

- ***Comments***
  1. All ep files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first ep file (file 1 in condition 1).

- ***Current version***

2.18 15-04-2011

- ***History***
  - 1.00 24-04-2003 (PEA) : 1st version.
  - 1.01 09-11-2004 (PEA) : minor modification.
  - 1.02 25-01-2005 (PEA) : minor modification.
  - 1.04 02-05-2005 (PEA) : adds shell sort. Major modifications.
  - 1.05 02-07-2005 (PEA) : minor modification.
  - 1.06 09-08-2005 (PEA) : adds eprand_compute=2.
  - 1.07 09-08-2005 (PEA) : fixes error.
  - 1.08 06-03-2006 (JB) : adds correction for multiple test for topographical test (eprand_measure=0), and fixes error for probability if eprand_compute=1.
  - 2.01 13-03-2006 (JB) : adds eprand_measure=5 (sum of values of condition 1).
  - 2.02 15-03-2006 (JB) : adds eprand_measure=6 (sum of squared condition sums).
  - 2.03 15-03-2006 (JB) : modifies unilateral tests (eprand_measure = 1,2,5) : the user must specify the condition with the largest mean by putting it in 1st rank in the .cond conditions file.
  - 2.05 18-03-2006 (JB) : adds minimum of significativity for samples. Integrates T2 sum to general case.
  - 2.06 19-03-2006 (JB) : adds minimum of significativity for channels x samples.
  - 2.07 19-03-2006 (JB) : modifies eprand_measure field : 0=topographical test, 1=bilateral test, 2=unilateral test.
  - 2.08 21-03-2006 (JB) : includes conditions in permutations.
  - 2.09 21-03-2006 (JB) : adds normalization.
  - 2.10 24-03-2006 (JB) : adds normalization with vector of maximum of analysis window.

- 2.12 28-03-2006 (JB) : enhances correction of minimum significativity algorithm.
- 2.13 29-03-2006 (JB) : normalizes subsampled values.
- 2.14 30-03-2006 (JB) : fixes error when few number of permutations, and subsampling.
- 2.15 18-05-2006 (JB) : adds check for file compatibility.
- 2.16 13-12-2006 (JB) : fixes error (rounded value) of time step and half-window size.
- 2.17 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
- 2.18 15-04-2011 (PEA) : removes static allocation for reading and writing EP file. Changes Shell sort to Heap sort in FDR (faster algorithm).

- ***Files***

  $ELANPATH/bin/epranddiff

- ***See also***

  eegranddiff [18]

epresample

- ***Description***

  Resamples an EP file (**.p**) from a template EP file (used for sampling frequency, prestimulus and poststimulus number of samples definition). There is no temporal interpolation of data. The values are just repeated. It is used to superimposed smoothed data (created by eegstat [19], epkruskal [17] or epwilcox [13]) with actual data.

- ***Usage***

  epresample file_template.p file_data_in.p file_data_out.p time_win time_shift

  with :

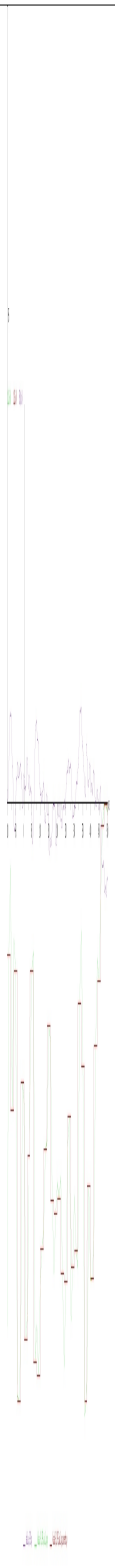
  - file_template.p : template file name (with extension). It gives sampling frequency, prestimulus and poststimulus number of samples.
  - file_data_in.p : input EP file to resample (with extension).
  - file_data_out.p : output resampled EP file (with extension).
  - time_win : time window duration in ms corresponding to the time step.
  - time_shift : time shift in ms to start the resampling.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

  1. The following figure illustrates the use of this program :



[20]

kala.f1.PS.p is the template file.

kala.f1.PS.wil.p.p is the result of epwilcox [13].

kala.f1.PS.wil.p.surech.p is resampled kala.f1.PS.wil.p.p. with values repeated along the each time window.

- ***Current version***

1.04 16-05-2014

- ***History***
  - 1.00 09-04-2006 (OB) : 1st version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 31-01-2011 (PEA) : removes static allocation for reading and writing EP files.
  - 1.04 16-05-2014 (PEA) : check indices of output files in some time window configuration (rounding latencies to samples).

- ***Files***

  $ELANPATH/bin/epresample

- ***See also***

  eegstat [19], epkruskal [17], epwilcox [13], epsmooth [21], tfavgresample [22]

Attachment	Size
epresample.jpg [20]	191.9 KB

eprms

- ***Description***

  Computes RMS (root mean square) value across channels on an EP (.p) file. The output is an EP file with only one channel containing the RMS value for each sample.

- ***Usage***

  eprms

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Number of channel to compute RMS on (0=all channels) :
8
Rank of channels to compute RMS on (starting from 1) :
1
2
3
4
5
6
7
8
Name of the file (without extension) to compute RMS on (or Return to quit) :
my_input_ep_file1
Name of the output file (without extension) :
ep_rms_8chan_file1
Name of the file (without extension) to compute RMS on (or Return to quit) :
my_input_ep_file2
Name of the output file (without extension) :
ep_rms_8chan_file2
Name of the file (without extension) to compute RMS on (or Return to quit) :

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. It should be noted that all the EP file name are entered without extension.

- ***Current version***

1.03 31-01-2011

- ***History***

  - 1.00 29-09-2006 (PEA) : 1st version.
  - 1.01 13-08-2007 (PEA) : minor modification.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 31-01-2011 (PEA) : removes static allocation for reading and writing EP files.

- ***Files***

$ELANPATH/bin/eprms

- ***See also***

epscd

- ***Description***

  Computes the interpolated values of an evoked potential file (**.p**) at the original electrodes. Scalp potential or scalp current density (SCD, e.g., surface Laplacian) can be computed. The output is a new evoked potential file (**.p**).

- ***Usage***

  epscd myfilein myfileout spline_order potscd_flag flag_regul [latbegin latend]

  with :

  - myfilein : input ep filename (potential data) (without extension).
  - myfileout : output filename (potential or SCD data) (without extension).
  - spline_order : order of the spline interpolation (usually 3 or 4).
  - potscd_flag : type of data to compute :

    0: data remains unchanged (potential values).
    1: computes SCD (Laplacian) from potential values.

  - flag_regul : type of regularization :

    0: no regularization.
    1: Tikhonov regularization.

  - options :

    latbegin latend : begin and end latencies; optional parameters for computation. If omitted, computation is performed on the whole epoch.

- ***Fields of parameter file and example***

- ***Examples***

  Command line to get the unregularized SCD for the whole epoch :
  epscd file_in file_out 4 1 0

- ***Comments***
  1. Spline order suggestion : if regularization is used for potential or SCD, take spline_order = 3, otherwise take spline_order = 4
  2. Possible error messages:

     Spline error : triangularization error (dspfa : 1) spline_calc_coef
     ERROR computing spline coefficients

     For these 2 messages, myfile_in probably contains electrodes for which SCD cannot be computed (aligned electrodes, too few electrodes, or incorrect electrode positions for example).

  3. Formulas :

     The comutation of SCD (Scalp Curent Density) is : $\Delta V^*\sigma/R2$ (en mA/m3) with
     σ = scalp conductivity = 0.45 Siemens/m = 0.45 $(Ohm.m)^{-1}$
     R = head radius = 0.09 m
     $\sigma/R2$ = 55.55 103 (with V in µV)

- ***Current version***

  1.17 24-01-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.10 11-12-2001 (PEA) : minor modification.
  - 1.11 01-02-2002 (PEA) : minor modification.
  - 1.13 10-04-2002 (PEA) : minor modification.
  - 1.14 20-05-2002 (PEA) : minor modification.
  - 1.15 13-08-2007 (PEA) : minor modification.
  - 1.16 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.17 24-01-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epscd

- ***See also***

  eegspline [3]

epsd

- ***Description***

  Computes standard deviation of EP (.p) files. The output is an EP file containing the standard deviation for each sample and each channel.

- ***Usage***

  epsd

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Number of EP files ?
3
EP filename #1 (without extension) :
my_input_ep_file1
EP filename #2 (without extension) :
my_input_ep_file2
EP filename #3 (without extension) :
my_input_ep_file3

Name of output pem file containing standard deviation values (without extension) :
ep_sd

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. It should be noted that all the EP file name are entered without extension.
2. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file.
3. Possible error messages:

   Incompatible epfile : p_xe[n]
   the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
   - 1 the number of channel,
   - 2 the number of samples,
   - 3 the epoch duration,
   - 4 the number of pre-stimulus samples,
   - 5 the sampling period.

- ***Current version***

1.02 31-01-2011

- ***History***

  - 1.00 14-11-2001 (BY) : 1st version.
  - 1.01 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.02 31-01-2011 (PEA) : removes static allocation for reading and writing EP files.

- ***Files***

$ELANPATH/bin/epsd

- ***See also***

epsmooth

- ***Description***

  Smoothes an EP (.p) files. The output is an EP file containing the smoothed value for each sample and each channel.

- ***Usage***

  epsmooth

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Window duration (msec) :
50
Smoothing beginning latency (msec) :
0
EP file name to smooth (without extension) or Return to quit :
my_input_ep_file1
Output file name (without extension) :
my_input_ep_file1.smoothed
EP file name to smooth (without extension) or Return to quit :

my_input_ep_file2
Output file name (without extension) :
my_input_ep_file2.smoothed
EP file name to smooth (without extension) or Return to quit :

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. It should be noted that all the EP file name are entered without extension.
2. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file.
3. Possible error messages:

   Incompatible epfile : p_xe[n]
   the file you just enter is not compatible with the previous ones. The 'n' indicates which field of the *.p header is wrong:
   - 1 the number of channel,
   - 2 the number of samples,
   - 3 the epoch duration,
   - 4 the number of pre-stimulus samples,
   - 5 the sampling period.

- ***Current version***

1.02 31-01-2011

- ***History***

  - 1.00 30-03-2000 (BY) : 1st version.
  - 1.01 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.02 31-01-2011 (PEA) : removes static allocation for reading and writing EP files.

- ***Files***

$ELANPATH/bin/epsmooth

- ***See also***

epspectrum

- ***Description***

  Computes the FFT of an evoked potential file (**.p**). The output is a new EP file (**.p**) containing the spectrum.

- ***Usage***

  epspectrum myfilein myfileout [-c][-begin=lat_begin_msec][-end=lat_end_msec][-blackman=nb_samples_blackman][-window=nb_samples_FFT]
  with :

  - myfilein : input EP filename (without extension).
  - myfileout : output filename (without extension).
  - options :

    -c : center signal (default is not).

-begin=lat_begin_msec : begin latency (msec) for computation (default is beginning of file).
-end=lat_end: lat_end_msec and end latency (msec) for computation (default is end of file).
-blackman=nb_samples_blackman: number of samples for Blackman window (default is 100).
-window=nb_samples_FFT: number of samples of FFT window (default is all samples, one FFT).

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

- ***Current version***

  1.03 01-02-2011

- ***History***
  - 1.00 04-03-2008 (PEA) : 1st version.
  - 1.01 14-04-2008 (PEA) : fixes computation error.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 01-02-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epspectrum

- ***See also***

epstudent

- ***Description***

  Computes the Student test on EP files (**.p**). It allows to compare 2 conditions with paired subjects. It takes the difference between the 2 conditions as input.

- ***Usage***

  epstudent

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Output EP file prefix (without extension) or Return to quit :
mystudenttestfile
Operation to apply to data : nothing (Return), normalize (n) or weight by event numbers (p or w) :

Enter **Return** to do nothing, **n** to normalize each difference file by maximum value of the file, or **w** to weight difference file by event number.
EP file name (without extension) of difference or Return to quit :
epfile1
EP file name (without extension) of difference or Return to quit :
epfile2
...
EP file name (without extension) of difference or Return to quit :

Output files :

Name	Comments#
mystudenttestfile.t.p	EP file of the t values of the Student test.

mystudenttestfile.p.p	EP file of the probability value (p) of the Student test.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).
2. The computed test is bilateral.

- ***Current version***

1.01 19-01-2012

- ***History***

  - 1.00 16-09-2011 (PEA) : 1st version.
  - 1.01 19-01-2012 (PEA) : fixes an error on test value.

- ***Files***

$ELANPATH/bin/epstudent

- ***See also***

epwilcox [13], epunpairedstudent [23]

eptopsquid

- ***Description***

  Writes the top squid number in an evoked potential file (**.p**).

- ***Usage***

  eptopsquid epfilein topsquid

  with :

  - epfilein : EP file to modify (without extension).
  - topsquid : top squid number.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. This number is : - (squid_number + 253) and stored in the 7th position of the 2nd header.

- ***Current version***

  1.03 01-02-2011

- ***History***
  - 1.01 30-03-2000 (OB/CTB) : 1st documented version.
  - 1.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 1.03 01-02-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/eptopsquid

- ***See also***

epunpairedstudent

- ***Description***

  Computes the Student test on EP files (**.p**). It allows to compare 2 conditions with unpaired subjects.

- ***Usage***

  epunpairedstudent

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Output EP file prefix (without extension) or Return to quit :
unpairedstudenttestfile
Number of files in condition 1 :
32
Number of files in condition 2 :
31
Input file name (with extension) 1 in condition 1 :
epfile1cond1.p
Input file name (with extension) 2 in condition 1 :
epfile2cond1.p
...
Input file name (with extension) 32 in condition 1 :
epfile32cond1.p
Input file name (with extension) 1 in condition 2 :
epfile1cond2.p
Input file name (with extension) 2 in condition 2 :
epfile2cond2.p
...
Input file name (with extension) 31 in condition 2 :
epfile31cond2.p

Output files :

NameComments#

unpairedstudenttestfile.t.p
EP file of the t values of the Student test.

unpairedstudenttestfile.p.p
EP file of the probability value (p) of the Student test.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).
2. The computed test is bilateral.
3. Number of subjects in each condition should be greater than 30.

- ***Current version***

1.00 12-01-2012

- ***History***

  - 1.00 12-01-2012 (PEA) : 1st version.

- ***Files***

$ELANPATH/bin/epunpairedstudent

- ***See also***

epstudent [24], epkruskal [17]

epwilcox

- ***Description***

  Computes the Wilcoxon test (non-parametric test) on EP files (**.p**). It allows to compare 2 conditions (or the difference between 2 conditions) with paired subjects.

- ***Usage***

  epwilcox [+fdr]
  with :

  - option :
    +fdr : computes False Detection Rate (FDR) after the statistical test. It creates 2 EP files with a mask of the statistical results (Z and p) with the computed FDR. This requires a probability (p) threshold value.

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Number of conditions (2 conditions or 1 = difference file) :
  2
  Number of files per condition :
  10
  FDR threshold :
  0.05
  It defines the probability (p) threshold value to be used for masking the statistical results (Z) by the threshold FDR statistics.
  Only if +fdr option is specified.

  Input file name (with extension) 1 in condition 1 :
  epfile1cond1.p
  Input file name (with extension) 2 in condition 1 :
  epfile2cond1.p
  ...
  Input file name (with extension) 1 in condition 2 :
  epfile1cond2.p
  ...
  Input file name (with extension) 10 in condition 0 :

epfile10cond2.p

Output file prefix (without extension) :
wilcox_ep

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***
  1. The programs creates 2 EP files : **.Z.p** and **.p.p** . In the above example, it will give wilcox_ep.p.p and wilcox_ep_Z.p .
  2. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).

- ***Current version***

  2.06 17-07-2012

- ***History***
  - 1.00 08-04-2003 (PEA) : 1st documented version.
  - 2.00 22-05-2003 (PEA/OB) : adds the input of the difference between 2 conditions.
  - 2.01 13-08-2007 (PEA) : minor modification.
  - 2.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 2.03 02-02-2011 (PEA) : removes static allocation for reading and writing EP files.
  - 2.04 07-10-2011 (PEA) : adds test for at least 2 samples to test in Wilcoxon test.
  - 2.05 16-05-2012 (PEA) : adds FDR correction option.
  - 2.06 17-07-2012 (PEA) : fixes an error in wilcox function of libstat when all samples are ties.

- ***Files***

  $ELANPATH/bin/epwilcox

- ***See also***

  epquade [14], epkruskal [17], epwinquade [15]

epwinquade

- ***Description***

  Computes the Quade test (non-parametric test) on a time-window of EP files (**.p**). It allows to compare many conditions with paired subjects.
  The Quade test is equivalent to an ANOVA with 1 factor, paired subjects and many conditions.

- ***Usage***

  epwinquade

This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
Number of conditions :
3
Number of files per condition :
10
Conover 2 by 2 comparisons : probability threshold (e.g. 0.01) :
0.01
Condition 1 | file 1 EP file name (with extension) :
epfile1cond1

Condition 1 | file 2 EP file name (with extension) :
epfile2cond1
...
Condition 2 | file 1 EP file name (with extension) :
epfile1cond2
...
Condition 3 | file 10 EP file name (with extension) :
epfile10cond3

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

1. All EP files should be compatible (in terms of number of channels and samples, number of pre-stimulus samples, sampling frequency) to the first EP file (file 1 in condition 1).

- ***Current version***

2.04 28-01-2011

- ***History***

  - 2.01 30-03-2000 (OB/CTB) : 1st documented version.
  - 2.02 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
  - 2.03 26-01-2011 (PEA) : removes static allocation for reading EP files. Changes Quade test input (p threshold).
  - 2.04 28-01-2011 (PEA) : moves from epquade.c to epwinquade.c.

- ***Files***

$ELANPATH/bin/epwinquade

- ***See also***

epwilcox [13], epkruskal [17], epquade [14]

epwinwrite

- ***Description***

  Writes in a text file the latency limits of the time-windows detected with a threshold on each channel of an EP file (**.p**).

- ***Usage***

  epwinwrite epfile.p threshold flag_abs flag_sign outputfile.txt [latmin latmax] [half-window]
  with :

  - epfile.p : input EP filename (with extension).
  - threshold :amplitude threshold for time-window detection (in signal units).
  - flag_abs : possible values :

    0 : window detection is based on signed signal amplitude.
    1 : window detection is based on absolute signal amplitude.

  - flag_sign : possible values :

    0 : windows detected when amplitude <= threshold.
    1 : windows detected when amplitude >= threshold.

- outputfile.txt : name of the text file where results will be written (with extension).
- options :

  latmin latmax : latency limits (in ms) between which window detection will be performed. If omitted, analysis is performed on the entire epoch.
  half-window : half-window duration for computing corrected window limits (this value refers to the half-window value used in eegstat for instance). If omitted, no corrected value is computed.

- ***Fields of parameter file and example***

- ***Examples***

  epwinwrite file.p 10 0 1 file.txt -100 300 25

  For each channel of file.p, the windows having an amplitude value >= 10 (signal units) are detected. This detection is performed in the interval -100 to 500 ms. Latency limits (lat1 and lat2) of each window are computed and written in file.txt.
  Latency limits corrected by the half-window value (hw) are also computed (lat1-hw and lat2+hw). This refers to the windowed processing performed by other programs such as smoothing (epsmooth [21]) or statistical analysis (eegstat [19]).
  The mean value of the signal (signed or absolute values) is also computed and written for each window.

  Example of output file (file.txt):

file	chan#	chan	win#	lat1	lat2	duration	lat1_c	lat2_c	duration_c	on border	mean_amp
kw11.H.p	1	M1	1	-100.0	-70.0	30.0	-125.0	-45.0	82.0	<	13.045
kw11.H.p	1	M1	2	130.0	132.0	2.0	105.0	157.0	52.0		10.565
kw11.H.p	1	M1	3	274.0	288.0	14.0	249.0	313.0	64.0		11.955
kw11.H.p	3	F3	1	24.0	32.0	8.0	-1.0	57.0	58.0		12.926
kw11.H.p	3	F3	2	56.0	70.0	14.0	31.0	95.0	64.0		14.121
kw11.H.p	3	F3	3	264.0	270.0	6.0	239.0	295.0	56.0		11.113
kw11.H.p	3	F3	4	278.0	284.0	6.0	253.0	309.0	56.0		11.333
kw11.H.p	3	F3	5	288.0	290.0	2.0	263.0	315.0	52.0		10.565
kw11.H.p	4	T3	1	128.0	132.0	4.0	103.0	157.0	54.0		10.709
kw11.H.p	4	T3	2	256.0	300.0	38.0	231.0	319.0	88.0	>	12.366

  chan# : channel rank in the **.p** file
  chan : channel name
  win# : window number for each channel
  lat1 : latency (ms) of the beginning of the window
  lat2 : latency (ms) of the end of the window
  duration : duration (ms) of the window
  lat1_c : latency (ms) of the beginning of the window corrected by the half-window (hw) parameter (lat1_c = lat1 – hw)
  lat2_c : latency (ms) of the end of the window corrected by the half-window (hw) parameter (lat2_c = lat2 + hw)
  duration : duration (ms) of the corrected window (duration_c = duration + 2*hw)
  > or < : window limit detected at the upper or lower latency limit for the detection (lat2 or lat1, respectively).
  mean_amp : mean amplitude of the signal over the detected window (in signal units)

- ***Comments***

- ***Current version***

  1.02 02-02-2011

- ***History***

- 1.00 30-04-2005 (OB) : 1st version.
- 1.01 29-09-2010 (PEA) : updates to use cmake and free release of Elan.
- 1.02 02-02-2011 (PEA) : removes static allocation for reading EP file.

- ***Files***

  $ELANPATH/bin/epwinwrite

- ***See also***

  epsmooth [21], eegstat [19], epmask [25], epresample [26]

matrix2p

- ***Description***

  Converts a matrix XML file (.xml) to an evoked potential file format (**.p**). This is used to visualize the component matrix obtained after ICA decomposition (topography of each component, one component per latency).

- ***Usage***

  matrix2p eegfile.eeg matrixfile.xml output_epfile.p [-i]

  with :

  - eegfile.eeg : input EEG filename used to read header informations (number of channels, channel names, etc) (with extension).
  - matrixfile.xml : matrix file (XML format) (with extension).
  - output_epfile.p : output evoked potential file containing the matrix values (with extension).
  - option :

    -i : matrix inversion.

- ***Fields of parameter file and example***

- ***Examples***

- ***Comments***

  1. This programme allows to visualize as an evoked potential file (**.p**) the decomposition matrix obtained after ICA decomposition. Each latency corresponds to the topography of one component. The first component is at latency 1.
  2. Matrix XML file format and example :

     35 *Total number of channels*
     32 *Number of channels used for computation*
     32 *Number of sources or components*
     1;2;3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;28;29;30;31;32; *List of used channel indices*
     -0.108017;-0.023034;0.056247; ... *Data value : matrix nbsources (lines) x nbusedchan (columns)*

     See the following example files :
     example_matrix2p.xml [27] is an example of matrix with 35 channels of EEG, 32 used for computation, and 32 components computed.
     example_one_component.xml [28] is an example of matrix with 340 channels of MEG and EEG, 275 MEG channels used for computation and only one component computed.

- ***Current version***

  1.04 29-09-2010

- ***History***
  - 1.00 22-02-2008 (PEA) : 1st version.
  - 1.01 25-02-2008 (PEA) : adds matrix inversion option.
  - 1.02 26-02-2008 (PEA) : changes for first component at latency 1 (one sample prestim=0, latency 0 =0), "number of samples"=number of components+2.
  - 1.03 03-03-2008 (PEA) : changes for "time" scale.
  - 1.04 29-09-2010 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/matrix2p

- ***See also***

  eegcomponent [29], eegfiltica [30], eegproject [31]

Attachment	Size
example_matrix2p.xml [32]	9.83 KB
example_one_component.xml [33]	7.87 KB

Lyon Neuroscience Research Center - Brain Dynamic and Cognition team

CRNL

Source URL: http://elan.lyon.inserm.fr/?q=ref_ep_proc_tool

Links:
[1] http://elan.lyon.inserm.fr/?q=epdiff
[2] http://elan.lyon.inserm.fr/?q=epscd
[3] http://elan.lyon.inserm.fr/?q=eegspline
[4] http://elan.lyon.inserm.fr/?q=epavg
[5] http://elan.lyon.inserm.fr/?q=sites/default/files/lowpass_butter_10hz_ep.par
[6] http://elan.lyon.inserm.fr/?q=sites/default/files/lowpass_butter_10hz_coefs_ep.par
[7] http://elan.lyon.inserm.fr/?q=sites/default/files/bp_2-30Hz_ep.par
[8] http://elan.lyon.inserm.fr/?q=eegfiltfilt
[9] http://elan.lyon.inserm.fr/?q=eegepochfiltfilt
[10] http://elan.lyon.inserm.fr/sites/default/files/lowpass_butter_10hz_ep.par
[11] http://elan.lyon.inserm.fr/sites/default/files/lowpass_butter_10hz_coefs_ep.par
[12] http://elan.lyon.inserm.fr/sites/default/files/bp_2-30Hz_ep.par
[13] http://elan.lyon.inserm.fr/?q=epwilcox
[14] http://elan.lyon.inserm.fr/?q=epquade
[15] http://elan.lyon.inserm.fr/?q=epwinquade
[16] http://elan.lyon.inserm.fr/?q=epranddiff
[17] http://elan.lyon.inserm.fr/?q=epkruskal
[18] http://elan.lyon.inserm.fr/?q=eegrandadd
[19] http://elan.lyon.inserm.fr/?q=eegstat
[20] http://elan.lyon.inserm.fr/sites/default/files/epresample.jpg
[21] http://elan.lyon.inserm.fr/?q=epsmooth
[22] http://elan.lyon.inserm.fr/?q=tfavgresample
[23] http://elan.lyon.inserm.fr/?q=epunpairedstudent
[24] http://elan.lyon.inserm.fr/?q=epstudent
[25] http://elan.lyon.inserm.fr/?q=epmask
[26] http://elan.lyon.inserm.fr/?q=epresample
[27] http://elan.lyon.inserm.fr/?q=sites/default/files/example_matrix2p.xml
[28] http://elan.lyon.inserm.fr/?q=sites/default/files/example_one_component.xml
[29] http://elan.lyon.inserm.fr/?q=eegcomponent
[30] http://elan.lyon.inserm.fr/?q=eegfiltica
[31] http://elan.lyon.inserm.fr/?q=eegproject
[32] http://elan.lyon.inserm.fr/sites/default/files/example_matrix2p.xml
[33] http://elan.lyon.inserm.fr/sites/default/files/example_one_component.xml