Table Of Contents

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

Time-Frequency Processing Tools

tfavg

- ***Description***

  Computes from an EEG file the mean of the time-frequency powers obtained from each single-trial (mean squared modulus of the wavelet transform). The output is stored in TF file format (one **.avg.tf** file by event code). The stimulus phase-locking factor is also computed for each event code and stored in TF file format (one **.pl.tf** file by event code). An averaged EP file (**.p**) can be substracted to each single trial prior to time-frequency transform.
  These computations requires a parameter file **.par** and an event position file **.pos**. Usually, the **.pos** file is the output of the averaging/rejection program eegavg [1].

- ***Usage***

  tfavg myeegfile.eeg myposfile.pos myparfile.par subsampling [+v] [+ri] [+s] [+z] [+allchannels] [+noreref]

  with :

  - *myeegfile**.eeg***: input ***.eeg*** file to process (with extension).
  - *myposfile**.pos***: input event file (with extension).
  - *myparfile**.par***: text file containing computation parameters (with extension).
  - subsampling: time-subsampling factor applied to the tf file created. In all cases, the time sample corresponding to the event onset (0 msec) is kept.

    1 : all time samples are saved,
    2 : one over 2 samples are saved,
    3 : one over 3 samples are saved,
    ….

  - options:
    +v : verbose mode on: all events are displayed during the processing progress. If omitted, verbose mode is off.
    +ri : creation of files **.avgr.tf** and **.avgi.tf** in TF format, containing the normalized mean real and imaginery parts of the time-frequency transforms; these values are useful for computing the mean of the stimulus-phase-locking factor across several files. If omitted, no file created.
    +s : creation of files **.avg2.tf** in TF format, containing the mean of the squared values of the time-frequency power (mean of the modulus4 of the wavelet transform). These values are useful for computing the standard deviation of the time-frequency power estimate. If omitted, no file created.
    +z : computes and creates a **.Z.avg.tf** file with the mean value across trials of the Z-transform of each trial, computed with respect to the baseline defined in the parameter file *myparfile.par*: (data – mean baseline)/standard deviation of baseline. If omitted, the file is not computed.
    +allchannels : compute on all analog channels of *myeegfile**.eeg***. The "tf_channel_flag" field of parameter file isn't read.
    +noreref : don't change reference of *myeegfile**.eeg*** channels. The "tf_channel_ref" field of parameter file isn't read.

- ***Fields of parameter file and examples***

fileprefix myfilename	Prefix of the output TF files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1,

	the extra sample corresponds to the event itself.
baseline_msec_start -200 -200	Required only for computing the Z-transform in **.Z.avg.tf** file (option **+z** in the command line); list of the baseline start latencies (in ms); one value for each event code.
baseline_msec_stop -50 -50	Required only for computing the Z-transform in **.Z.avg.tf** file (option **+z** in the command line); list of the baseline stop latencies (in ms); one value for each event code.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed and stored in the output **.avg.tf** files.
tf_channel_ref 0 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is unchanged, channel 2 is referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 18 18	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
tf_substract_epfile ep.2.p ep.5.p	List of the EP file names with extension (**.p**) used for substracting an averaged response to each single trial prior to time-frequency computation (one string for each event code). These **.p** files should be compatible to the time-frequency analysis parameters (number of channels, number of samples pre and post-stimulus). If omitted, no **.p** file is substracted to the single trials.

- ***Examples***

In this example, all trials with codes 2 or 5 will be processed with a time window ranging from from 400 samples prior to 1000 samples after event code. The time-frequency analysis will be performed on channels 2 and 5, from 18 to 80 Hz by steps of 2 Hz with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7.Wavelet transform is performed on channels 1, 2 and 4; channel 1 is unchanged, channel 2 is re-referenced to channel 5, and channel 4 to channel 6.

Output files:

Name	Comments#
myfilename.2.avg.tf myfilename.5.avg.tf	Mean of the time-frequency power across single trials (event codes 2 and 5 in this example).
myfilename.2.pl.tf myfilename.5.pl.tf	Stimulus-phase-locking factor in the time-frequency domain computed across single trials (event codes 2 and 5 in this example).

myfilename.2.avgr.tf myfilename.2.avgi.tf myfilename.5.avgr.tf myfilename.5.avgi.tf	If option +ri: Normalized mean real and imaginery parts of the time-frequency transforms obtained over single trials (event codes 2 and 5 in this example).
myfilename.2.avg2.tf myfilename.5.avg2.tf	If option +s: Mean of the squared values of the time-frequency power across single trials (event codes 2 and 5 in this example).
myfilename.2.Z.avg.tf myfilename.5.Z.avg.tf	If option +z: Z-transform of the data with respect to the baseline computed on each single trial and averaged across trials (event codes 2 and 5 in this example).

- ***Comments***
  1. Note to CTF 275 MEG users : an example of parameter file with 275 channels is available to download (ctf275_meg.par [2])
  2. See eegchref [3] to create a re-referenced .eeg data file (several referencing options available).

- ***Current version***

  1.38 23-07-2014

- ***History***
  - 1.00 01-12-2001 (OB/CTB/PEA) : 1st documented version.
  - 1.10 09-12-2001 (PEA) : changes labels of parameter file.
  - 1.11 31-01-2002 (PEA) : checks for subsampling step validity.
  - 1.12 04-02-2002 (PEA) : minor modification.
  - 1.13 23-04-2002 (PEA) : changes file extensions. Adds +z option.
  - 1.14 29-05-2002 (PEA) : changes **.z.tf** to **.Z.avg.tf**.
  - 1.15 17-06-2002 (PEA) : supports readpos function (from libpos).
  - 1.16 26-06-2002 (PEA) : changes integer types (long, short to int).
  - 1.17 25-02-2003 (PEA) : minor modification.
  - 1.18 18-09-2003 (PEA) : changes EEG values process. Removes unused variables.
  - 1.19 03-11-2003 (PEA) : adds check for dynamic memory allocation.
  - 1.20 13-11-2003 (PEA) : adds 32 bits EEG support.
  - 1.21 12-10-2004 (PEA) : adds EP file substraction.
  - 1.22 03-11-2004 (PEA) : uses rejection flag of **.pos** files (libpos).
  - 1.24 18-01-2006 (PEA) : ???.
  - 1.25 02-03-2006 (PEA) : improves memory allocations.
  - 1.26 10-03-2006 (PEA) : improves memory allocations.
  - 1.27 09-01-2007 (PEA) : fixes channel names when no number (elec.dat) is given (adds -1).
  - 1.28 13-08-2007 (PEA) : minor modification.
  - 1.29 25-11-2008 (PEA) : adds MPI usage with OpenMPI. Removes static allocations. Changes array indexation for more readability. Fixes signtest function (libstat) for 64 bits. Adds write_tf_frequency function to libtfmoy.
  - 1.30 19-12-2008 (PEA) : improves memory usage for MPI version (allocation of used frequencies only).
  - 1.31 13-03-2009 (PEA) : improves frequency distribution in MPI version.
  - 1.32 22-02-2010 (PEA) : fixes reading EP file to substract in libpem.
  - 1.33 17-06-2011 (PEA) : uses threads to compute TF transform (TF transform function of libelansignal).
  - 1.34 20-06-2011 (PEA) : minor modification : optimizes memory usage for signal.
  - 1.35 12-06-2012 (PEA) : minor modification : output file name creation.
  - 1.36 22-10-2012 (PEA) : add "+allchannels" and "+noreref" options.
  - 1.37 21-07-2014 (PEA) : fixes an error with +z option introduced with thread usage. Removes baseline_eventcode field (not used).
  - 1.38 23-07-2014 (PEA) : fixes some memory leaks (no deallocation).

- ***Files***

  $ELANPATH/bin/tfavg

- ***See also***

  tfep [4], tfsync [5]

tfavgavg

- ***Description***

  Computes the grand-average of tf files. Output file in tf format (*.avg.tf). The averaged files must have compatible headers (same channels, same frequency list, same time-window, etc…).

- ***Usage***

  tfavgavg [+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):
  Weigthing with number of averaged trials (y/n) ?
  y
  y: the number of trials averaged in each tf file is used to weight the grand-average
  n: each tf file has the same weight in the grand-average.
  Number of TF files to average:
  3
  Baseline correction: none (0), mean (1) or median (2) in a time-window (0/1/2) :
  1
  Start latency for baseline correction (in ms) :
  -400
  Stop latency of baseline correction (in ms) :
  -100
  Name of TF file # 1 (with extension) :
  myfile1.avg.tf
  Name of TF file # 2 (with extension) :
  myfile2.avg.tf
  Name of TF file # 3 (with extension) :
  myfile3.avg.tf
  Name of the averaged TF file (with extension) :
  myfile.grand.avg.tf

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

- ***Examples***

  In this example, myfile.grand.avg.tf contains the baseline-corrected weighted average.

- ***Comments***

- ***Current version***

  1.04 02-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 17-12-2003 (PEA) : adds +force option.
  - 1.03 13-08-2007 (PEA) : minor modification.
  - 1.04 02-02-2011 (PEA) : updates to use cmake and free release of Elan. Removes static allocations.

- ***Files***

  $ELANPATH/bin/tfavgavg

- ***See also***

  tfavgdiff [6]

tfavgbline

- ***Description***

  Creates a new TF file with baseline correction. The baseline value is computed for each frequency, between two latencies (usually in the prestimulus period). The baseline value can be computed on a given TF file and substracted from either the same or another TF file.

- ***Usage***

  tfavgbline

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  BaseLine computed on the same TF file (Y/N)? :
  n
  y: the baseline value is computed on the file to correct
  n: the baseline value is computed on one file, and the correction using these baseline values is applied to another file.
  Type of baseline correction (1=mean, 2= median)
  1
  1: the baseline is defined by the mean value between start and stop latencies
  2: the baseline is defined by the median value between start and stop latencies
  Start latency of baseline (in ms) :
  -400
  Stop latency of baseline (in ms) :
  -100
  Substract (1) or Divide (2) by the baseline ? (1/2):
  2
  1: the baseline value is substracted from the original data (in each frequency band)
  2: the original data are divided by the baseline value (in each frequency band)
  TF file name to correct (with extension) :
  myfile1.avg.tf
  TF file name on which baseline is computed (with extension) :
  myfile2.avg.tf
  Name of the baseline corrected file (with extension) :
  myfile1.bl2.avg.tf

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

- ***Examples***

  In this example, the output file myfile1.bl2.avg.tf contains the data from myfile1.avg.tf divided by the mean value between –400 and –100 ms of myfile2.avg.tf, for each frequency band.

- ***Comments***

- ***Current version***

  1.04 03-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 20-05-2005 (PEA) : minor modification.
  - 1.03 13-08-2007 (PEA) : adds RMS and maximum options.
  - 1.04 03-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgbline

- ***See also***

tfavgchannel

- ***Description***

  Computes the grand-average across-channels of a TF file (output file in TF format).

- ***Usage***

  tfavgchannel

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Number of channels to average or to sum (0 for all electrodes) :
  4
  Channel rank # 1 :
  22
  (channel rank starting from #1)
  Channel rank # 2 :
  6
  Channel rank # 3 :
  11
  Channel rank # 4 :
  12
  Channel average (a) or sum (s) ?
  a
  a: average of the tf of the selected channels
  s: sum of the tf of the selected channels
  Name of TF file to process (with extension) or Enter to quit :
  myfile.avg.tf
  Name of the TF file with summed or averaged channels (with extension) :
  myfile.4ch.avg.tf
  Name of TF file to process (with extension) or Enter to quit :
  (return)
  A new **.tf** file can be processed with the same parameters, or the return key terminates the program.

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

- ***Examples***

  In this example, the output file myfile.4ch.avg.tf contains only one channel corresponding to the average of channels 22, 6, 11, 12 of myfile.avg.tf. The single resulting averaged channel of myfile.4ch.avg.tf is labelled with the name of first channel (i.e., #22) of myfile.avg.tf. The list of averaged channels is stored in myfile.4ch.avg.tf header and can be displayed using tfavgread [7].

- ***Comments***

- ***Current version***

  1.03 03-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.

- 1.01 04-02-2002 (PEA) : minor modification.
- 1.02 13-08-2007 (PEA) : minor modification.
- 1.03 03-02-2011 (PEA) : updates to use cmake and free release of Elan. Removes static allocations.

- ***Files***

  $ELANPATH/bin/tfavgchannel

- ***See also***

tfavgcutt

- ***Description***

  Truncates a TF file in time.

- ***Usage***

  tfavgcutt

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Start latency of the new file (in ms) :
  -200
  Stop latency of the new file (in ms) :
  1500
  Input file name (with extension) or Enter to quit :
  myfile1.avg.tf
  Output file name (with extension) :
  myfile1.-200_1500.tf
  Input file name (with extension) or Enter to quit :
  (return)
  a new TF file can be processed with the same parameters,
  or the return key terminates the program.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 28-02-2011

- ***History***
  - 1.00 01-08-2001 (CTB/OB) : 1st documented version.
  - 1.01 28-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgcutt

- ***See also***

tfavgdiff

- ***Description***

  Computes the difference between 2 TF files (output in TF file format). The files must have compatible headers (same channels, same frequency list, same time-window, etc…).

- ***Usage***

  tfavgdiff [+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):
  Name of first TF file (with extension) :
  myfile1.avg.tf
  Name of second TF file (with extension) :
  myfile2.avg.tf
  TF file name for difference 1-2 (with extension) :
  myfile1-2.avg.tf

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

- ***Examples***

  In this example, the output file myfile1-2.avg.tf contains the time-frequency difference (myfile1.avg.tf -myfile2.avg.tf).

- ***Comments***

- ***Current version***

  1.04 04-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 25-05-2007 (PEA) : minor modification.
  - 1.03 13-08-2007 (PEA) : minor modification.
  - 1.04 04-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgdiff

- ***See also***

  tfavgavg [8]

tfavgdiffchannel

- ***Description***

  Computes the difference between 2 channels of one TF file (output in TF file format). The output is 1st channel - 2nd channel .

- ***Usage***

  tfavgdiffchannel

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Channel 1 rank in TF file (starting from 1) :
  5
  Channel 2 rank in TF file (starting from 1) :
  4
  Input file name (with extension) or Enter to quit :
  myfile1.avg.tf
  Output file name (with extension) :
  myfile1.diff.5-4.tf
  Input file name (with extension) or Enter to quit :
  (return)
  a new TF file can be processed with the same parameters,
  or the return key terminates the program.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 01-03-2011

- ***History***
  - 1.00 01-08-2001 (CTB/OB) : 1st documented version.
  - 1.01 01-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgdiffchannel

- ***See also***

tfavglistval

- ***Description***

  Computes mean and max values of a list TF files in a time-frequency window (output in a text file).

- ***Usage***

  tfavglistval

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Mean (1) or maximum (2) value ?
  1
  1 for mean value
  2 for maximum value
  Start latency (in ms) :
  100

Stop latency (in ms) :
200
Start frequency (in Hz) :
20
Stop frequency (in Hz) :
30
Output file name (with extension) :
result.txt
name of the output text file containing the measures for all input files
Baseline correction: none (0), mean (1) or median (2) in a time-window ? (0/1/2) :
0
0 no baseline correction
1 baseline correction with mean value on time window
2 baseline correction with median value on time window
If baseline type ≠ 0 :
Start latency for baseline :
-200
Stop latency for baseline :
-50
Substract(1) or divide(2) by baseline ? (1/2)
1
1 the baseline value is substracted from the original data
2 the original data are divided by the baseline value
Input file name (with extension) or Enter to quit :
myfile1.avg.tf
Input file name (with extension) or Enter to quit :
myfile2.avg.tf
Input file name (with extension) or Enter to quit :
(return)
a new TF file can be processed with the same parameters,
or the return key terminates the program.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 02-03-2011

- ***History***
  - 1.00 01-08-2001 (CTB/OB) : 1st documented version.
  - 1.02 01-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavglistval

- ***See also***

tfavgmask

- ***Description***

  Masks a TF file with another TF file.

- ***Usage***

  tfavgmask file_mask.tf file_data.tf file_data_out.tf threshold flag new_value
  with :

  - file_mask.tf : input TF file used for masking (with extension).
  - file_data.tf : input TF file to be masked (with extension).
  - file_data_out.tf : output masked data TF file (with extension).
  - threshold : value of the threshold.
  - flag : possible values :

    0 : modification of values <= threshold.
    1 : modification of values >= threshold.
    2 : modification of values = threshold.

  - new_value : masked value written in data file.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.04 04-02-2011

- ***History***
  - 1.00 08-11-2004 (PEA) : 1st documented version.
  - 1.02 09-06-2008 (PEA) : minor modification.
  - 1.03 01-09-2009 (PEA) : minor modification.
  - 1.04 04-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgmask

- ***See also***

  tfavgsmooth [9]

tfavgmedian

- ***Description***

  Computes median values of TF files (output in TF file format).

- ***Usage***

  tfavgmedian

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Number of files to compute median :
  6
  Baseline correction: none (0), mean (1) or median (2) value on a time-window? (0/1/2) :
  1
  0: no baseline correction

1: the baseline is defined by the mean value between start and stop latencies
2: the baseline is defined by the median value between start and stop latencies
Start latency for baseline correction (in ms.) :
-400
Stop latency for baseline correction (in ms.) :
-100
Input file name 1 (with extension) :
myfile1.avg.tf
Input file name 2 (with extension) :
myfile2.avg.tf
Input file name 3 (with extension) :
myfile3.avg.tf
Input file name 4 (with extension) :
myfile4.avg.tf
Input file name 5 (with extension) :
myfile5.avg.tf
Input file name 6 (with extension) :
myfile6.avg.tf
Output file name (with extension) :
myfile.median.tf

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 03-03-2011

- ***History***
  - 1.00 01-08-2001 (CTB/OB/PEA) : 1st documented version.
  - 1.01 03-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgmedian

- ***See also***

tfavgnorm

- ***Description***

  Normalizes a TF file with maximum value. This value is the maximum of all channels, frequencies and samples.

- ***Usage***

  tfavgnorm file_data_in.tf file_data_out.tf
  with :

  - file_data_in.tf : input TF file (with extension).
  - file_data_out.tf : output normalized TF file (with extension).

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 04-03-2011

- ***History***
  - 1.00 26-07-2005 (PEA) : 1st version.
  - 1.04 03-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgnorm

- ***See also***

tfavgprofilef

- ***Description***

  Creates a frequency profile in a selected time interval of a TF file (output in EP file format **.p**). This profile (**.p**) can be displayed by erpa.

- ***Usage***

  tfavgprofilef [+sqrt] [+sem coef]

  with :

  - option :

    +sqrt : computes square root of data. If omitted, the profile of data is created.
    +sem coef : computes standard error of the mean (sem) for each point. It outputs 3 files : one file containing mean values (**.p** suffix), one with mean values + coef * SEM (**.sem1.p** suffix), and one with mean values - coef * SEM (**.sem2.p** suffix). For example :
    tfavgprofilef +sem 2
    will create files with 2*SEM values added or substracted to the mean values.

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Start latency (ms):
  200
  Stop latency (ms):
  400
  start and stop latencies of the time-window in which tf values will be averaged or summed.
  Baseline correction: none (0), mean (1) or median (2) in a time-window ? (0/1/2) :
  1
  0: no baseline correction
  1: the mean value between start and stop latencies will be used
  2: the median value between start and stop latencies will be used
  if baseline correction type ≠ 0
  Start latency of baseline (in ms.) :
  -400
  Stop latency of baseline (in ms.) :
  -100
  Substract (1) or divide (2) by baseline ? (1/2)
  1

1: the baseline value is substracted from the original data
2: the original data are divided by the baseline value
Sum (0) or average (1) :
1
0: sum over time of the tf values for each frequency band
1: average over time of the tf values for each frequency band
Input file name (with extension) or Enter to quit :
myfile1.avg.tf
Output EP file name (without extension):
myfile1.avg.lat200-400.bl
Input file name (with extension) or Enter to quit :
myfile2.avg.tf
the same processing will be repeated with new files.
Output EP file name (without extension):
myfile2.avg.lat200-400.bl
Input file name or Enter to quit :
(return)
a new TF file can be processed with the same parameters,
or the return key terminates the program.

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

- ***Examples***

  In this example, the output files myfile1.avg.lat200-400.bl.p and myfile2.avg.lat200-400.bl.p contain baseline corrected frequency profiles between 200 and 400 ms. Note that the **.p** extension is generated automatically.

- ***Comments***

- ***Current version***

  1.09 03-10-2013

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 09-12-2003 (PEA) : adds baseline correction support.
  - 1.03 04-03-2004 (PEA) : fixes EP format header size.
  - 1.04 07-10-2004 (PEA) : adds +sqrt option to compute square root of data.
  - 1.05 13-08-2007 (PEA) : minor modification.
  - 1.06 11-03-2008 (PEA) : removes static allocations.
  - 1.07 07-02-2011 (PEA) : updates to use cmake and free release of Elan.
  - 1.08 10-01-2012 (PEA) : adds the option to create output files with Standard Error of the Mean (+sem).
  - 1.09 03-10-2013 (PEA) : fixes deallocation memory after file creation.

- ***Files***

  $ELANPATH/bin/tfavgprofilef

- ***See also***

  tfavgprofilet [10]

tfavgprofilet

- ***Description***

  Creates a time profile in a selected frequency bandl of a TF file (output in EP file format **.p**). This profile (**.p**) can be displayed by erpa.

- ***Usage***

  tfavgprofilet [+sqrt] [+sem coef]

  with :

  - options :

    +sqrt : computes square root of data. If omitted, the profile of data is created.
    +sem coef : computes standard error of the mean (sem) for each point. It outputs 3 files : one file containing mean values (**.p** suffix), one with mean values + coef * SEM (**.sem1.p** suffix), and one with mean values - coef * SEM (**.sem2.p** suffix). For example :
    tfavgprofilet +sem 2
    will create files with 2*SEM values added or substracted to the mean values.

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Frequencies in Hz (y/n) ?
  y
  if yes:
  Start frequency :
  10
  Stop frequency :
  16
  if no:
  Number of frequencies to sum or average :
  3
  Rank of frequency 1 :
  1
  Rank of frequency 2 :
  2
  Rank of frequency 3 :
  6
  In this case the frequency bands number 1, 2, and 6 are averaged. The ranks of these frequencies are stored in the tf header.
  Number of samples to truncate :
  50
  This truncature is used to remove the first 50 and last 50 time samples from the resulting profile in the ep file, to prevent visualizing the effect of the tapering window (Blackman window).
  Baseline correction: none (0), mean (1) or median (2) on a time-window ? (0/1/2) :
  1
  0: no baseline correction
  1: the mean value between start and stop latencies will be used
  2: the median value between start and stop latencies will be used
  if baseline correction ≠ 0
  Start latency of baseline (in ms) :
  -400
  Stop latency of baseline (in ms) :
  -100
  Substract(1) or divide(2) by baseline ? (1/2)
  1
  1: the baseline value is substracted from the original data
  2: the original data are divided by the baseline value
  Sum (0) or average (1):
  1
  0: sum over frequency bands of the tf values for each time sample
  1: average over frequency bands of the tf values for each time sample
  Input file name (with extension) or Enter to quit :
  myfile1.avg.tf
  Output EP file name (without extension):
  myfile1.avg.freq10-16.bl
  Input file name (with extension) or Enter to quit :
  myfile2.avg.tf
  the same processing will be repeated with new files.
  Output EP file name (without extension):
  myfile2.avg.freq10-16.bl
  Input file name (with extension) or Enter to quit :
  (return)

a new TF file can be processed with the same parameters,
or the return key terminates the program.

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

- ***Examples***

  In this example, the output files myfile1.avg.freq10-16.bl.p and myfile2.avg.freq10-16.bl.p contain baseline corrected averaged time-profiles between 10 and 16 Hz. Note that the **.p** extension is generated automatically.

- ***Comments***

- ***Current version***

  1.08 03-10-2013

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 09-12-2003 (PEA) : adds baseline correction support.
  - 1.03 04-03-2004 (PEA) : fixes EP format header size.
  - 1.04 07-10-2004 (PEA) : adds +sqrt option to compute square root of data.
  - 1.05 13-08-2007 (PEA) : minor modification.
  - 1.06 08-02-2011 (PEA) : updates to use cmake and free release of Elan. Removes static allocations.
  - 1.07 09-01-2012 (PEA) : adds the option to create output files with Standard Error of the Mean (+sem).
  - 1.08 03-10-2013 (PEA) : fixes deallocation memory after file creation.

- ***Files***

  $ELANPATH/bin/tfavgprofilet

- ***See also***

  tfavgprofilef [11]

tfavgquade

- ***Description***

  Computes the Quade test (non-parametric test) on TF files (**.tf**). 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***

  tfavgquade

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
Baseline correction: none (0), mean (1) or median (2) value on a time-window? (0/1/2)
1
Start latency for baseline correction (in ms.) :

-500
Stop latency for baseline correction (in ms.) :
-50
Subtract (1) or divide (2) by baseline ? (1/2)
1
Input file name 1 in condition 1 (with extension) :
tffile1cond1.avg.tf
Input file name 2 in condition 1 (with extension) :
tffile2cond1.avg.tf
...
Input file name 1 in condition 2 (with extension) :
tffile1cond2.avg.tf
...
Input file name 10 in condition 3 (with extension) :
tffile10cond3.avg.tf

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

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

- ***Examples***

- ***Comments***

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

- ***Current version***

1.04 31-01-2011

- ***History***

  - 1.01 13-04-2005 (PEA) : 1st documented version.
  - 1.02 13-08-2007 (PEA) : minor modification.
  - 1.03 30-07-2008 (PEA) : minor modification.
  - 1.04 31-01-2011 (PEA) : removes static allocation. Changes Quade test input (p threshold).

- ***Files***

$ELANPATH/bin/tfavgquade

- ***See also***

tfavgwilcox [12]

tfavgread

- ***Description***

  Reads and displays header and data of a TF file (output displayed on screen).

- ***Usage***

  tfavgread myfile.tf

  with :

- myfile.tf : input TF file name (with extension) to read.

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

- ***Examples***

In the following example, the output and questions of the program are italic, and user's answers are bold:
tfavgread myfile.avg.tf
version : 1 binary TF file version1.0
header size : 856
data address : 888
data type : 1 float
free size :
file type : avg.tf
file subtype 1 : average of TF files
file subtype 2t : no stat
Signal type : potential
Event code : 12
as defined in file.par
number of channels : 1
as defined in file.par
channel number (according to elec.dat)
10
channel name (according to elec.dat)
Cz
number of time samples : 1800
as defined in file.par
number of pre-stimulus time-samples : 600
as defined in file.par
sampling frequency (Hz) : 1000
as defined in file.par
number of frequencies : 6
as defined in file.par
tf_wavelet_type : 1 Morlet's wavelet
as defined in file.par
frequency values:
20.00 Hz
24.00 Hz
28.00 Hz
32.00 Hz
36.00 Hz
40.00 Hz
tf_morlet_m: 7.00 7.00 7.00 7.00 7.00 7.00 7.00
as defined in file.par
tf_nb_sample_blackman (ms) : 100.00
as defined in file.par
sub-sampling step (samples) : 2
as defined in tfavg command-line
number of averaged trials by channel:
channel 1 213 events
channel 2 213 events
channel 3 213 events
baseline correction flag: 101 mean from a different TF file
see tfavgbline [13]
start latency of baseline (ms) : -400
stop latency of baseline (ms) : -100
flag of time-frequency smoothing : 0 no smoothing
see tfavgsmooth [9]
flag of across-channel mean : 0 no mean across channels
see tfavgchannel [14]

Data display
Number of channels to display (total number of channels=1; 0 to exit) :
1
0 to exit
Channel ranks to display (1 to 1) :

1
channel 1 :
 # name number(elec.dat) f (Hz) lat (ms) value
1 Cz 16 -600.00 187.041260
….

- ***Comments***

- ***Current version***

  1.03 09-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 13-08-2007 (PEA) : minor modification.
  - 1.03 09-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgread

- ***See also***

tfavgresample

- ***Description***

  Re-samples any TF (**.tf**) file following the time and frequency resolutions of a TF template file.
  This is required for combining smoothed or statistically windowed TF files to other high-resolution TF files (by tfavgmask for instance). The output file has exactly the same dimension as the template file. Number of channels for template an input TF files should be the same.

- ***Usage***

  tfavgresample file_template.tf file_in.tf file_out.tf [time_win time_shift]
  with :

  - file_template.tf : template TF file (with extension) : defines time and frequency sampling.
  - file_in.tf : input TF file (with extension) tor resample.
  - file_out.tf : resulting TF re-sampled file (with extension) with the same time-frequency resolution as file_template.tf .
  - options :

    time_win : time window duration (in ms) corresponding to the time step. If omitted, the time step corresponds to the sampling period in file_in.tf .
    time_shift : time shift (in ms) to start the resampling. Required if time_win is used.

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

- ***Examples***

- ***Comments***

- ***Current version***

1.03 11-03-2011

- ***History***
  - 1.01 22-04-2005 (OB) : 1st documented version (renamed from tfavgunsmooth).
  - 1.02 18-07-2005 (PEA) : minor modification.
  - 1.03 11-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgresample

- ***See also***

  epresample [15], tfstat [16], tfavgwilcox [12], tfavgsmooth [9], tfavgmask [17]

tfavgsmooth

- ***Description***

  Applies a smoothing time-frequency window on a TF file (output in TF file format). The value at the center of each time-frequency window of the output file is the mean value computed over the window.

- ***Usage***

  tfavgsmooth

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Start latency (in ms) :
  -500
  Stop latency (in ms) :
  1200
  Start frequency (in Hz) :
  8
  Stop frequency (in Hz) :
  80
  Half smoothing time-window (in ms) :
  50
  the full time window duration will be 101 ms
  Time step for time smoothing (in ms) :
  4
  Half smoothing frequency-window (in Hz) :
  2
  the full frequency window size will be 5 Hz
  Frequency step for frequency smoothing (in Hz) :
  2
  Name of the TF file to smooth (with extension) or Enter to quit :
  myfile.avg.tf
  Name of smoothed TF file :
  myfilesmoothed.avg.tf
  Name of the TF file to smooth (with extension) or Enter to quit :
  (return)
  a new TF file can be processed with the same parameters,
  or the return key terminates the program.

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

- ***Examples***

- ***Comments***
  1. The output file (**.tf**) is computed between the chosen start and stop latencies, and has a time-frequency resolution depending on the chosen time and frequency steps.

- ***Current version***

  1.04 10-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 13-08-2007 (PEA) : minor modification.
  - 1.03 08-02-2008 (PEA) : adds check for input value consistency.
  - 1.04 10-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgsmooth

- ***See also***

  tfavgmask [17]

tfavgstd

- ***Description***

  Computes the standard-deviation from the mean (**.avg.tf**) and the mean of squares (**.avg2.tf**) obtained from tfavg [18] (output in TF file format).

- ***Usage***

  tfavgstd

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Input file name (with extension) for mean (.avg.tf):
  myfile.avg.tf
  Input file name (with extension) for mean of squares (.avg2.tf):
  myfile.avg2.tf
  Confidence intervals yes(1)/no(0) : (1/0)
  0
  0 : no confidence interval generation
  1 : creates confidence interval files for lower and upper limits
  If confidence interval :
  *alpha (to compute mean +/- alpha*std) =*
  1
  Output file name (with extension) (standard deviation in a .tf file):
  myfile.sd.tf
  If confidence interval :
  Output file name (confidence interval upper limit in a .tf file):
  myfile.up.sd.tf
  If confidence interval :
  Output file name (with extension) (confidence interval lower limit in a .tf file):
  myfile.lo.sd.tf

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 14-03-2011

- ***History***
  - 1.00 01-08-2001 (OB/CTB/PEA) : 1st documented version.
  - 1.01 14-03-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfavgstd

- ***See also***

  tfavg [18]

tfavgval

- ***Description***

  Computes the mean TF value or the maximum TF value and the corresponding latency and frequency of an **.tf** file in a given time-frequency window (output in a text file or on screen). This allows to export measurements for statistical analysis from a group of subjects and for different experimental conditions in a given time-frequency window.

- ***Usage***

  tfavgval

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Mean (1) or maximum (2) value?
  2
  1: computes the mean value in a time-frequency window
  2: detects the maximum value, its latency and its frequency, in a time-frequency window
  Start latency (in ms) :
  200
  Stop latency (in ms) :
  400
  Start frequency (in Hz) :
  20
  Stop frequency (in Hz) :
  34
  the time-frequency measurement window is defined by these 4 parameters.
  Baseline correction: none (0), mean (1) or median (2) in a time-window ? (0/1/2) :
  1
  0: no baseline correction
  1: the mean value between start and stop latencies will be used
  2: the median value between start and stop latencies will be used
  if baseline correction ≠ 0
  Start latency of baseline (in ms) :
  -400
  Stop latency of baseline (in ms) :
  -100
  Substract(1) or divide(2) by baseline ? (1/2)
  1
  1: the baseline value is substracted from the original data
  2: the original data are divided by the baseline value
  Input file name (with extension) or Enter to quit :
  myfile1.avg.tf

Output file name (Return = display on screen only) :
myfile1.txt
Label (common to all channels in output file):
mylabel1
Input file name (with extension) or Enter to quit :
myfile2.avg.tf
the same processing will be repeated with new files.
Output file name (Return = display on screen only) :
myfile2.txt
Label (common to all channels in output file):
mylabel2
Input file name (with extension) or Enter to quit :
(return)
a new TF file can be processed with the same parameters,
or the return key terminates the program.

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

- ***Examples***

The above example will output *myfile1.txt* and *myfile2.txt*.
Content of *myfile1.txt*:
Baseline correction :
latency –400.0 to –200.0 ms
Correction type : substract the mean value of baseline
#label #chan #latency #frequency #value #boundary
mylabel1 Cz 200.00 20.00 18.5436 1
mylabel2 O1 234.00 22.00 23.5673 0
….

Boundary-flag to 1 means that the time-frequency window boundaries were reached when searching for the maximum time-frequency value at this electrode. The detected maximum is thus not reliable.

- ***Comments***

- ***Current version***

1.05 11-02-2011

- ***History***
  - 1.00 09-12-2001 (PEA) : 1st documented version.
  - 1.01 04-02-2002 (PEA) : minor modification.
  - 1.02 20-02-2002 (PEA) : fixes ROI display.
  - 1.03 17-05-2002 (PEA) : minor modification.
  - 1.04 13-08-2007 (PEA) : minor modification.
  - 1.05 11-02-2011 (PEA) : updates to use cmake and free release of Elan. Adds title to the output file before data.

- ***Files***

$ELANPATH/bin/tfavgval

- ***See also***

tfavgwilcox

- ***Description***

Computes the Wilcoxon test (non-parametric test) on TF files (**.tf**). It allows to compare 2 conditions (or the difference

between 2 conditions) with paired subjects.

- ***Usage***

  tfavgwilcox [+fdr]
  with :

  - option :
    +fdr : computes False Detection Rate (FDR) after the statistical test. It creates a TF file with a mask of the statistical results (Z) 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 for difference file) :
  2
  Number of files per condition :
  1
  Baseline correction: none (0), mean (1) or median (2) on a time-window ? (0/1/2) :
  1
  0: no baseline correction
  1: the mean value between start and stop latencies will be used
  2: the median value between start and stop latencies will be used
  if baseline correction ≠ 0
  Start latency of baseline (in ms) :
  -400
  Stop latency of baseline (in ms) :
  -100
  Substract(1) or divide(2) by baseline ? (1/2)
  1
  1: the baseline value is substracted from the original data
  2: the original data are divided by the baseline value
  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 :
  tf_file1cond1.avg.tf
  Input file name (with extension) 2 in condition 1 :
  tf_file2cond1.avg.tf
  ...
  Input file name (with extension) 1 in condition 2 :
  tf_file1cond2.avg.tf
  ...
  Input file name (with extension) 10 in condition 0 :
  tf_file10cond2.avg.tf

  Output file prefix (no extension) :
  tf_wilcox

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

- ***Examples***

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

- ***Current version***

  1.06 17-07-2012

- ***History***
  - 1.00 01-08-2001 (OB/CTB/PEA) : 1st documented version.
  - 1.01 16-03-2011 (PEA) : updates to use cmake and free release of Elan. Adds the difference between 2 conditions. Adds .p.tf creation.
  - 1.02 24-03-2011 (PEA) : fixes an error in name reading (from 1.01).
  - 1.03 06-04-2011 (PEA) : adds +fdr option to compute FDR.
  - 1.04 15-04-2011 (PEA) : changes Shell sort to Heap sort in FDR (faster algorithm).
  - 1.05 07-10-2011 (PEA) : adds test for at least 2 samples to test in Wilcoxon test.
  - 1.06 17-07-2012 (PEA) : fixes an error in wilcox function of libstat when all samples are ties.

- ***Files***

  $ELANPATH/bin/tfavgwilcox

- ***See also***

tfchchannel

- ***Description***

  Changes the elctrode name and number (in elec.dat) of a TF file and outputs in a new TF file.

- ***Usage***

  tfchchannel myinfile.tf myoutfile.tf channel_rank elec_num

  with :

  - myinfile.tf : input TF file name (with extension).
  - myoutfile.tf : output TF file name (with extension).
  - channel_rank : channel rank of electrode to change (starting from 1).
  - elec_num : new electrode number from elec.dat.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 04-04-2011

- ***History***
  - 1.00 21-07-2005 (PEA) : 1st documented version.
  - 1.01 04-04-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfchchannel

- ***See also***

tfchprestim

- ***Description***

  Substracts a duration to the prestimulus of a TF file and outputs in a new TF file.

- ***Usage***

  tfchprestim myinfile.tf myoutfile.tf duration

  with :

  - myinfile.tf : input TF file name (with extension).
  - myoutfile.tf : output TF file name (with extension).
  - duration : duration in msec to substract to prestimulus.

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

- ***Examples***

- ***Comments***

- ***Current version***

  1.01 04-04-2011

- ***History***
  - 1.00 21-07-2005 (PEA) : 1st documented version.
  - 1.01 04-04-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfchprestim

- ***See also***

tfcmppower

- ***Description***

  Performs a statistical analysis of the time-frequency representation of single-trials between 2 conditions, with randomization.
  The False Detection Rate (FDR) procedure is possible on the statistical results (see comments).

- ***Usage***

  tfcmppower myeegfile.eeg myeventfile.pos myparfile.par subsampling nb_drawings

  with :

  - myeegfile.eeg : input EEG file name (with extension).
  - myeventfile.pos : event position file (with extension) (this is usually the output pos file of eegavg, after artifact rejection).
  - myparfile.par: text file containing computation parameters (with extension).
  - subsampling: time-subsampling factor applied to the TF file created. In all cases, the time sample corresponding

to the event onset (0 msec) is kept.

1 : all time samples are saved,
2 : one over 2 samples are saved,
3 : one over 3 samples are saved,
….

- nb_drawings : number of drawings that are made to estimate the distribution of differences under the null hypothesis (example: 10000 drawings enables a resolution of p=0.0001).

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

fileprefix myfilename	Prefix of the output TF file.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed.
tf_channel_ref 0 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is unchanged, channel 2 is referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 10 10	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
tfstat_flag_fdr 0	Flag allowing to compute False Detection Rate (FDR) after the statistical test: 0: no FDR computation 1: FDR computation, and generation of a TF file with a mask of the statistical results with the computed FDR (see output files below). This requires a probability (p) threshold value (tfstat_threshold_fdr). If omitted, the default value is 0.
tfstat_threshold_fdr 0.05	To be used in case of tfstat_flag_fdr = 1. Defines the probability (p) threshold value to be used for masking the statistical results by the threshold FDR statistics.

- ***Examples***

  tfcmppower myeegfile.eeg myeventfile.pos myparfile.par 1 1000
  In this example, all trials with codes 2 and 5 be processed with a time window ranging from 400 samples prior to 1000 samples after event code. The time-frequency analysis will be performed on channels 1 (not re-referenced), 2 (re-referenced to channel 3),and 4 (re-referenced to channel 5), from 10 to 80 Hz by steps of 2 Hz with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7. All samples are saved to output file. The program performs 1000 randomizations of the 2 conditions.

  Output file :
  myfilename.2-5.rand.p.tf
  myfilename.2-5.rand.p.fdr.tf (if FDR applied)

- ***Comments***
  1. This program works with 2 conditions. The computing parameters (the frequencies, number of samples pre and post-stimulus, wavelet parameters) must be the same for the 2 conditions.
  2. **tfstat_flag_fdr** : This option allows to compute the False Detection Rate statistics after a Wilcoxon or a Kruskal-Wallis test. This is a possible solution for multiple testing problem. See: Genovese, C. R., N. A. Lazar, et al. (2002). "Thresholding of statistical maps in functional neuroimaging using the false discovery rate." Neuroimage 15(4): 870-8.

- ***Current version***

  1.06 12-06-2012

- ***History***
  - 1.00 01-08-2001 (CTB/OB) : 1st version.
  - 1.01 17-06-2002 (PEA) : changes event file reading.
  - 1.02 26-09-2003 (PEA) : changes in physical values of EEG.
  - 1.03 01-12-2003 (PEA) : adds 32 bits EEG support.
  - 1.04 04-04-2011 (PEA) : updates to use cmake and free release of Elan. Fixes memory allocations. Optimizes memory allocation and loops. Adds FDR correction.
  - 1.05 15-04-2011 (PEA) : changes Shell sort to Heap sort in FDR (faster algorithm).
  - 1.06 12-06-2012 (PEA) : minor modification : output file name creation.

- ***Files***

  $ELANPATH/bin/tfcmppower

- ***See also***

  tfavg [18], tfstat [16], tfcmpsync [19]

tfcmpsync

- ***Description***

  Performs a statistical analysis of the time-frequency synchronization between channels of single-trials between 2 conditions, with randomization.

- ***Usage***

  tfcmpsync myeegfile.eeg myeventfile.pos myparfile.par subsampling nb_drawings [+v]

  with :

  - myeegfile.eeg : input EEG file name (with extension).
  - myeventfile.pos : event position file (with extension) (this is usually the output pos file of eegavg, after artifact

rejection).
- myparfile.par: text file containing computation parameters (with extension).
- subsampling: time-subsampling factor applied to the TF file created. In all cases, the time sample corresponding to the event onset (0 msec) is kept.

  1 : all time samples are saved,
  2 : one over 2 samples are saved,
  3 : one over 3 samples are saved,
  ….

- nb_drawings : number of drawings that are made to estimate the distribution of differences under the null hypothesis (example: 10000 drawings enables a resolution of p=0.0001).
- option :
  +v : verbose mode on: all events are displayed during the processing progress. If omitted, verbose mode off.

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

fileprefix myfilename	Prefix of the output TF file.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed.
tf_channel_ref 0 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is unchanged, channel 2 is referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 10 10	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
nbsync_perchannel 2 1 0 0 0 0 0	Number of channels to compute synchronization with (for each channel). In this example, the channel 1 is tested with 2 other channels, and the channel 2 with 1 channel.
sync_list 2 4	List of channel to test synchronization for each channel defined in nbsync_perchannel

4	label. In this example, the syncronization betwwen channel 1 and 2, 1 and 4, and 2 and 4 are tested.

- ***Examples***

  tfcmpsync myeegfile.eeg myeventfile.pos myparfile.par 1 1000
  In this example, all trials with codes 2 and 5 be processed with a time window ranging from 400 samples prior to 1000 samples after event code. The time-frequency analysis will be performed on channels 1 (not re-referenced), 2 (re-referenced to channel 3),and 4 (re-referenced to channel 5), from 10 to 80 Hz by steps of 2 Hz with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7.
  The synchronizations are tested between channels : 1 and 2, 1 and 4, 2 and 4. All samples are saved to output file. The program performs 1000 randomizations of the 2 conditions.

  Output files (each file contains one channel) :
  myfilename.2.5.c1.c2-c3.rand.p.tfsync
  myfilename.2.5.c1.c3-c4.rand.p.tfsync
  myfilename.2.5.c2-c3.c3-c4.rand.p.tfsync

- ***Comments***
  1. This program works with 2 conditions. The computing parameters (the frequencies, number of samples pre and post-stimulus, wavelet parameters) must be the same for the 2 conditions.

- ***Current version***

  1.05 12-06-2012

- ***History***
  - 1.00 01-08-2001 (CTB/OB) : 1st version.
  - 1.01 17-06-2002 (PEA) : changes event file reading.
  - 1.02 26-09-2003 (PEA) : changes in physical values of EEG.
  - 1.03 05-01-2004 (PEA) : adds 32 bits EEG support.
  - 1.04 26-05-2011 (PEA) : updates to use cmake and free release of Elan. Fixes memory allocations. Optimizes memory allocation and loops.
  - 1.05 12-06-2012 (PEA) : minor modification : output file name creation.

- ***Files***

  $ELANPATH/bin/tfcmpsync

- ***See also***

  tfstatsync [20], tfcmppower [21], tfsync [5]

tfdelchan

- ***Description***

  Deletes channels from a TF file and creates a new TF file.

- ***Usage***

  tfdelchan mytf_in.tf mytf_out.tf del_chan_nb rank1 rank2 ... rankn
  with :

  - mytf_in.tf : input TF file to process (with extension).
  - mytf_out.tf : output TF file to process (with extension).
  - del_chan_nb : number of channels to remove from mytf_in.tf.
  - rank1 rank2 ... rankn : list of channel indices to delete (first is number 1).

- ***Fields of parameter file and examples***

- ***Example***
  tfdelchan mytf_in.tf mytf_out.tf 2 3 5
  mytf_out.tf contains all data from mytf_in.tf except channel 3 and 5.

- ***Comments***

- ***Current version***

  1.00 09-06-2011

- ***History***
  - 1.00 09-06-2011 (PEA) : 1st version.

- ***Files***

  $ELANPATH/bin/tfdelchan

- ***See also***

tfdetect

- ***Description***

  Detects burst on time-frequency transforms of EEG single trials. It ouputs informations in a text file and a TF (**.detect.tf** file containing the ratio of burst detection for each time and frequency.

- ***Usage***

  tfdetect myeegfile.eeg myposfile.pos myparfile.par subsampling mode coef [+v]

  with :

  - myeegfile.eeg: input EEG file to process (with extension).
  - myposfile.pos: input event file (with extension).
  - myparfile.par: text file containing computation parameters (with extension).
  - subsampling: time subsampling factor applied to the TF file created. In all cases, the time sample corresponding to the event onset (0 msec) is kept.
    1 : all time samples are saved,
    2 : one over 2 samples are saved,
    3 : one over 3 samples are saved,
    ….
  - mode: threshold type. May be one the following values :
    1: adaptative threshold (threshold = baseline average + coef * baseline standard deviation) and bursts detection is done on half-value of the local maximum. The baseline is computed for each frequency.
    2: adaptative threshold (threshold = baseline average + coef * baseline standard deviation) and burst detection is done on threshold. The baseline is computed for each frequency.
    3: same as one, but marks in TF file, the first sample of the first detected burst only.
    4: adaptative threshold (threshold = coef * maximum value of baseline) and burst detection is done on threshold. The baseline is computed for each frequency.
    5: adaptative threshold (threshold = coef * maximum value of baseline) and burst detection is done on threshold. The baseline is computed on the frequencies sum.
  - coef: multiplier of the baseline standard deviation.
  - options:
    +v : verbose mode on: all events are displayed during the processing progress. If omitted, verbose mode is off.

- ***Fields of parameter file and examples***

fileprefix myfilename	Prefix of the output TF files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed and stored in the output **.avg.tf** files.
tf_freq_start 18 18	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
bl_threshold_start -300 -300	Baseline start latency (in ms). One value for each event code. The threshold is computed from the mean and standard deviation of the baseline.
bl_threshold_stop -50 -50	Baseline stop latency (in ms). One value for each event code. The threshold is computed from the mean and standard deviation of the baseline.

- ***Examples***

tfdetect myeegfile.eeg myposfile.pos myparfile.par subsampling 1 2
In this example, all trials with codes 2 or 5 will be processed with a time window ranging from from 400 samples prior to 1000 samples after event code. The time-frequency analysis will be performed for each single trials on channels 1, 2 and 4, from 18 to 80 Hz by steps of 2 Hz with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7. The mean value and standard deviation of the baseline are computed from -300 to -50 ms.
For each trial, the threshold of each frequency is : baseline_average + 2*baseline_sd .

Output files:

Name	Comments#
myfilename.2.detect.tf myfilename.5.detect.tf	Ratio of burst detection for each time and frequency (100*number of bursts / number of events).
	Text file containing information for each burst detection. The first line is the column labels. Meaning of labels : event : event code chan : channel number

myfilename.2.detect.txt myfilename.5.detect.txt	trial : indice of trial (for event code *event*) freq : frequency (Hz) burst : indice of burst at this frequency for this trial and channel lat_start : start of burst latency (ms) lat_stop : end of burst latency (ms) dur : burst duration in (ms) Emax : burst maximum amplitude lat_max : latency of burst maximum amplitude Eavg : burst mean value Eavgbl : baseline mean value Ethreshold : threshold used freq1 : starting frequency freq2 : ending frequency peak : indice of peak (local maximum) in the burst Emax_bl : baseline maximum value fmax : frequency of the burst maximum amplitude

- ***Comments***
  1. Note to CTF 275 MEG users : an example of parameter file with 275 channels is available to download (ctf275_meg.par [2])
  2. See eegchref [3] to create a re-referenced .eeg data file (several referencing options available).
  3. See an example of use Rols, G., Tallon-Baudry, C., Girard, P., Bertrand, O. and Bullier, J. Cortical mapping of gamma oscillations in areas V1 and V4 of the macaque monkey. Vis Neurosci, 2001, 18(4): 527-40. [22]

- ***Current version***

  1.05 12-06-2012

- ***History***
  - 1.00 01-08-2001 (OB) : 1st documented version.
  - 1.01 17-06-2002 (PEA) : supports readpos function (from libpos).
  - 1.02 26-09-2003 (PEA) : changes EEG values process.
  - 1.03 01-12-2003 (PEA) : adds 32 bits EEG support.
  - 1.04 19-05-2011 (PEA) : removes static allocations. Update to use cmake and free release of Elan. Modifies baseline parameters to each event (instead one for all events).
  - 1.05 12-06-2012 (PEA) : minor modification : output file name creation.

- ***Files***

  $ELANPATH/bin/tfdetect

- ***See also***

  tfavg [18]

tfep

- ***Description***

  Computes the time-frequency power using Morlet wavelets on an EP file (**.p**), and stores it in tf file format (**.ep.tf**)

- ***Usage***

  tfep

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Start frequency (Hz) :
  18
  Stop frequency (Hz) :
  80

Frequency step (Hz) :
2
m wavelet value :
7
Number of samples of the Blackman window rise or fall time:
100
as defined in the *.par file used for tfavg.
Time sub-sampling (in samples, 1=no sub-sampling) :
2
as defined in the command line of tfavg.
Number of channels to process (0 for all channels):
3
Channel rank :
1
Channel rank :
3
Channel rank :
8
EP file name (no .p extension) or Enter to quit :
my_ep_file
this corresponds to the ep file my_ep_file.p to be processed
Name of the output TF file (with extension) or Enter to keep the same file prefix (adds .ep.tf) :
(return)
default output name if return key : creates my_ep_file.ep.tf
EP file name (no .p extension) or Enter to quit :
my_ep2_file
Name of the output TF file (with extension) or Enter to keep the same file prefix (adds .ep.tf) :
ep2.ep.tf
EP file name (no .p extension) or Enter to quit :
(return)
a new EP file can be processed with the same parameters,
or the return key terminates the program.

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

- ***Examples***

  In this example, the two ouput TF files created are *my_ep_file.ep.tf* and *ep2.ep.tf*. The two EP input files *my_ep_file.p* and *my_ep2_file.p* must have compatible headers (same number of samples, same channels). The parameters used for the time-frequency transform have the same meaning as those used in the par file for tfavg [18].

- ***Comments***
  1. The input EP files must have identical number of channels, number of samples and sampling period.
  2. The program doesn't check for identical electrode names of the input EP files.

- ***Current version***

  1.06 14-02-2011

- ***History***
  - 1.01 04-02-2002 (PEA) : 1st documented version.
  - 1.02 07-03-2002 (PEA) : adds reading of elec.dat to get electrode names.
  - 1.03 19-03-2002 (PEA) : minor modification.
  - 1.04 17-05-2002 (PEA) : minor modification.
  - 1.05 13-08-2007 (PEA) : minor modification.
  - 1.06 14-02-2011 (PEA) : updates to use cmake and free release of Elan.

- ***Files***

  $ELANPATH/bin/tfep

- ***See also***

  tfavg [18]

tfmval

- ***Description***

  Computes mean or max values on time-frequency transformed single trials of an EEG file in time-frequency windows (output in a text file).

- ***Usage***

  tfmval myeegfile.eeg myeventfile.pos myparfile.par [windows_file.tf.wnd]

  with :

  - myeegfile.eeg : input EEG file name (with extension) to read and transform.
  - myeventfile.pos : input event file (with extension) used to define single trials.
  - myparfile.par: text file containing computation parameters (with extension).
  - options:
    windows_file.tf.wnd : text file describing the windows. If omitted, windows will be created from parameter file (myparfile.par).

    .

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

fileprefix myfilename	Prefix of the output files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed and stored in the output files.
tf_channel_ref 0 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is unchanged, channel 2 is referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 18 18	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event

	code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
tf_flag_log 0	Flag to compute log10 of power. Possible values are : 1 : compute log10(power). 0 : compute power. This field is optionnal. If omitted, power is computed.
tfmval_freq_hw 6	Frequency half window (Hz).
tfmval_freq_step 12	Frequency step (Hz).
tfmval_time_hw 50	Time half window (ms).
tfmval_time_step 100	Time step (ms).

These parameters are specific to time-frequency window definition with parameter file. A window file (**.tf.wnd**) is also created. This file can be used for further computation as input window file.
In case of use of a window file (windows_file.tf.wnd), omit these fields, and create a text file as below :

wnd_nb 2	Number of windows.
wnd_list 500 1000 9 12 1500 2000 9 12	Window description list : a window is defined by 4 values (on one line) : - start latency (ms). - stop latency (ms). - start frequency (Hz). - stop frequency (Hz). In this example, the 1st window is [500; 1000] ms and [9-12] Hz, and the 2nd window is [1500; 2000] ms and [9-12] Hz.
wnd_label_list part1 10 part2 10	Label list (columns title in the results text file) : there is 2 labels for each window : - time label. - frequency label. In this example, the 1st window has "part1" for time label, and "10" for frequency label, and the 2nd window has "part2" for time label, and "10" for frequency label.

Output file :
The output file is a text file containing data extracted for each window in each signle trial.
With the above parameters, a file *myfilename.mval.tf.txt* is created.
The 1st line contains the title of each column :
file_prefix event_code trial_number channel_number f t f_label t_label mean_value min_value t_min_value f_min_value min_on_border max_value t_max_value f_max_value max_on_border

The following lines contain data corresponding to each columns.

- ***Examples***

- ***Comments***
  1. In output file, the "channel_number" columns refers to the indice of the channel in the EEG file.

- ***Current version***

  1.09 15-02-2011

- ***History***
  - 1.01 18-09-2002 (PEA) : 1st documented version.
  - 1.02 27-02-2003 (PEA) : changes output file extensions, frees memory.
  - 1.03 26-09-2003 (PEA) : changes in physical values of EEG.

- 1.04 08-01-2004 (PEA) : adds 32 bits EEG support.
- 1.05 13-08-2007 (PEA) : minor modification.
- 1.06 15-02-2010 (PEA) : adds tf_flag_log field to compute log of power.
- 1.07 25-03-2010 (PEA) : fixes log error : using log10f instead of logf.
- 1.08 26-03-2008 (PEA) : fixes problem with float mathematical functions (using log10 instead of log10f).
- 1.09 15-02-2011 (PEA) : updates to use cmake and free release of Elan. Fixes time limit problem.

- ***Files***

  $ELANPATH/bin/tfmval

- ***See also***

  tfavg [18], tfavgval [23], eegmval [24]

tfph2pl

- ***Description***

  Computes by moving circular average a phase-locking file (**.pl.tf**) from a phase file (**.ph.tf**) obtained from tfsync [5].

- ***Usage***

  tfph2pl

  This program uses an interactive input.The questions are as follows (questions (program) are italic, answers (user) are bold):
  Half smoothing time-window (in ms) :
  50
  the full time window duration will be 101 ms
  Time step for time smoothing (in ms) :
  50
  Name of the .ph.tf phase file to transform (with extension) or Return to quit :
  myfile.ph.tf
  Name of the output .pl.tf file (with extension) :
  myfile.pl.tf
  Name of the .ph.tf phase file to transform (with extension) or Return to quit :
  (return)
  a new TF file can be processed with the same parameters,
  or the return key terminates the program.

- ***Fields of parameter file and examples***

- ***Example***

- ***Comments***

- ***Current version***

  1.03 09-09-2011

- ***History***
  - 1.00 09-06-2002 (OB) : 1st version.
  - 1.01 13-08-2007 (PEA) : minor modification (library names).
  - 1.02 30-09-2010 (PEA) : minor modification.
  - 1.03 09-09-2011 (PEA) : updates to use cmake and free release of Elan. Dynamic memory allocations. Fixes an error with smoothed data (last time window larger than number of samples).

- ***Files***

  $ELANPATH/bin/tfph2pl

- ***See also***

  tfsync [5]

tfphmodulo2pi

- ***Description***

  Transform a TF phase file (.ph.tf) with values between +/-180° to a phase file with values between +/-360°.

- ***Usage***

  tfphmodulo2pi myfile_in.ph.tf myfile_out.ph.tf
  with :

  - myfile_in.ph.tf : input TF file to process (with extension).
  - myfile_out.ph.tf : output TF file to process (with extension).

- ***Fields of parameter file and examples***

- ***Example***

- ***Comments***

- ***Current version***

  1.01 12-06-2011

- ***History***
  - 1.00 14-05-2005 (OB) : 1st version.
  - 1.01 12-06-2011 (PEA) : updates to use cmake and free release of Elan. Fixes an error when writing output file (last channel only saved).

- ***Files***

  $ELANPATH/bin/tfphmodulo2pi

- ***See also***

tfstat

- ***Description***

  Performs a statistical analysis of the time-frequency representation of single-trials computed from an **.eeg** file. Two statistical tests are proposed on the TF power averaged over regions of interest defined in the time-frequency domain.
  - a Wilcoxon test (non-parametric paired comparison) for the detection of an emerging TF component with respect to a baseline period. The test is performed indepedantly for each event code and each channel.
  - a Kruskal-Wallis test (one-way analysis of variance by ranks) for the statistical comparison of the time-frequency activities across different event codes.
  The region of interest is defined by its size and a moving step (in Hz x ms) to cover the whole time-frequency domain.

This program creates **.tf** files representing the statistical values (Z for Wilcoxon, H for Kruskal-Wallis and their related probability values p) computed in each time-frequency window. They can be viewed by tfviz as time-frequency plots. The False Detection Rate (FDR) procedure is also possible on the Wilcoxon or Kruskal-Wallis statistical results (see comments).

- ***Usage***

  tfstat myeegfile.eeg myeventfile.pos myparfile.par

  with :

  - myeegfile.eeg : input EEG file name (with extension).
  - myeventfile.pos : event position file (with extension) (this is usually the output pos file of eegavg, after artifact rejection).
  - myparfile.par: text file containing computation parameters (with extension).

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

fileprefix myfilename	Prefix of the output TF files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed.
tf_channel_ref 0 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is unchanged, channel 2 is referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 10 10	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.

tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
tf_type_stat 2	Type of statistical analysis based on single-trial time-frequency regions of interest: 1: Wilcoxon test on each moving time-frequency region of interest with respect to a baseline period. 2: Kruskal-Wallis test on each moving time-frequency region of interest between the different event codes.
tfstat_baseline_flag 1 1	List of flags by event code for applying a baseline correction on single-trials prior to the Kruskal-Wallis test: 0: no baseline correction. 1: baseline correction applied. If omitted, the default value is 0.
tfstat_freq_hw 10 10	List of the frequency half-windows (in Hz) by event code, used to defined the moving time-frequency regions of interest on which statistical analysis will be performed.
tfstat_freq_step 5 5	List of the frequency steps (in Hz) by event code, used to define the moving step of the time-frequency regions of interest on which statistical analysis will be performed.
tfstat_time_hw 200 200	List of the time half-windows (in ms) by event code, used to define the moving time-frequency regions of interest on which statistical analysis will be performed. If = 0, then all time points are considered in the analysis.
tfstat_time_step 50 50	List of the time steps (in ms) by event code, used to defined the moving step of the time-frequency regions of interest on which statistical analysis will be performed.
tfstat_baseline_start -300 -300	List of the baseline start latencies (in ms) by event code. Required when tfstat_type_stat=1 (Wilcoxon test) or (tfstat_type_stat=2 and tfstat_baseline_flag=1). If omitted, the start latency is the first point of the pre-stimulus baseline defined by prestim_nbsample.
tfstat_baseline_stop -100 -100	List of the baseline end latencies (in ms) by event code. Required when tfstat_type_stat=1 (Wilcoxon test) or (tfstat_type_stat=2 and tfstat_baseline_flag=1). If omitted, the stop latency is 0 ms.
tf_flag_log 0	Statistical test on the power or the log of the power in the TF domain: 0: statistics on power 1: statistics on the log10 of the power. If omitted, the default value is 0.
tfstat_flag_fdr 0	Flag allowing to compute False Detection Rate (FDR) after the statistical test: 0: no FDR computation 1: FDR computation, and generation of a TF file with a mask of the statistical results (Z or H) with the computed FDR (see output files below). This requires a probability (p) threshold value (tfstat_threshold_fdr). If omitted, the default value is 0.
tfstat_threshold_fdr 0.05	To be used in case of tfstat_flag_fdr = 1. Defines the probability (p) threshold value to be used for masking the statistical results (Z or H) by the threshold FDR statistics.
tfstat_kruskal_modified 0	In case of Kruskal-Wallis test with 2 conditions (2 event codes), this flag allows to indicate that all TF windows related to event code #2 is compared to a baseline window related to event code #1. For the definition of the baseline, see tfstat_kruskal_modified_baseline_beg_msec and tfstat_kruskal_modified_baseline_end_msec. Possible values are: 0: standard case (no comparison of event code #2 to the baseline of even code #1) 1: comparison of event code #2 to the baseline of event code #1 If omitted, the default value is 0.

tfstat_kruskal_modified_baseline_beg_msec -300	To be used if tfstat_kruskal_modified = 1. Latency of the beginning of the baseline period (in msec) relative to event code #1.
tfstat_kruskal_modified_baseline_end_msec -50	To be used if tfstat_kruskal_modified = 1. Latency of the end of the baseline period (in msec) relative to event code #1.
tf_substract_epfile ep.2.p ep.5.p	List of the EP file names with extension (**.p**) used for substracting an averaged response to each single trial prior to time-frequency computation (one string for each event code). These **.p** files should be compatible to the time-frequency analysis parameters (number of channels, number of samples pre and post-stimulus). If omitted, no **.p** file is substracted to the single trials.
tfstat_wilcox_modified 0	In case of Wilcoxon test with 2 conditions (2 event codes), this flag allows to indicate that all TF windows related to event code #2 is compared to a baseline window related to event code #1. Events must be paired. Possible values are: 0: standard case (no comparison of event code #2 to the baseline of even code #1) 1: comparison of event code #2 to the baseline of event code #1 If omitted, the default value is 0.

- ***Examples***

In this example, all trials with codes 2 and 5 be processed with a time window ranging from 400 samples prior to 1000 samples after event code. The evoked potential read from respectively ep.2.p and ep.5.p are substracted to single trials. The time-frequency analysis will be performed on channels 1 (not re-referenced), 2 (re-referenced to channel 3),and 4 (re-referenced to channel 5), from 10 to 80 Hz by steps of 2 Hz with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7. The statistical analysis is here a Kruskal-Wallis test between event codes 2, 5 and 8. On each trial, the time-frequency analysis is performed with a baseline correction defined between –300 and –100 ms and over a time-frequency region of interest of 20 Hz x 400 ms (2x10 Hz x 2x200 ms) moving by step of 5 Hz x 50 ms.

Output files :

Name	Comments#
myfilename.2.wil.Z.tf myfilename.5.wil.Z.tf myfilename.8.wil.Z.tf	For the Wilcoxon test: TF files of the Z value of the Wilcoxon test for each event code.
myfilename.2.wil.p.tf myfilename.5.wil.p.tf myfilename.8.wil.p.tf	For the Wilcoxon test: TF files of the probability value (p) of the Wilcoxon test for each event code.
myfilename.2.5.8.KW.H.tf	For the Kruskal-Wallis test: TF file of the H value of the Kruskal-Wallis test computed on the 3 event codes (2, 5, 8).
myfilename.2.5.8.KW.p.tf	For the Kruskal-Wallis test: TF file of the probability value (p) of the Kruskal-Wallis test computed on the 3 event codes (2, 5, 8).
myfilename.2.Z.fdr.tf	For the Wilcoxon test: TF file of the Wilcoxon Z value masked by the thresholded probability value (p) of the FDR procedure.
myfilename.2.5.8.H.fdr.tf	For the Kruskal-Wallis test: TF file of the Kruskal-Wallis H value masked by the thresholded probability value (p) of the FDR procedure.

- ***Comments***
  1. **tfstat_kruskal_modified** : This option allows, for instance, to compare each position in the TF domaine related to an active condition (related to event code #2) to a resting state defined by the mean TF computed for each frequency in a given time-window (related to event code #1).
  2. **tfstat_flag_fdr** : This option allows to compute the False Detection Rate statistics after a Wilcoxon or a Kruskal-Wallis test. This is a possible solution for multiple testing problem. See: Genovese, C. R., N. A. Lazar, et al. (2002).

"Thresholding of statistical maps in functional neuroimaging using the false discovery rate." Neuroimage 15(4): 870-8.

- ***Current version***

  1.30 22-09-2014

- ***History***
  - 1.00 18-09-2002 (PEA) : 1st documented version.
  - 1.01 22-10-2002 (PEA) : adds Kruskal-Wallis test.
  - 1.02 17-02-2003 (PEA) : changes tfmval... parameters to tfstat....
  - 1.03 26-09-2003 (PEA) : changes in physical values of EEG.
  - 1.04 03-10-2003 (PEA) : minor modification (removes debug informations).
  - 1.05 08-10-2003 (PEA) : fixes time window definition.
  - 1.06 13-10-2003 (PEA) : fixes time window definition.
  - 1.07 26-11-2003 (PEA) : supports 32 bits EEG.
  - 1.08 17-12-2003 (PEA) : fixes time window definition.
  - 1.09 08-01-2004 (PEA) : fixes time window definition.
  - 1.10 07-04-2005 (OB) : minor modification (event file reading).
  - 1.11 03-10-2005 (OB) : minor modification.
  - 1.12 30-10-2005 (OB) : limits display. Checks for compatiblity of baseline with pre and post-stimulus.
  - 1.14 18-04-2006 (OB) : fixes time window size and steps.
  - 1.15 17-10-2006 (PEA) : adds tfstat_kruskal_modified parameter.
  - 1.17 04-10-2007 (PEA) : adds ability to work on log10 of power.
  - 1.18 16-07-2008 (PEA) : adds FDR support.
  - 1.19 09-09-2008 (PEA) : minor modification (same shell_sort_float function as eegstat [25]).
  - 1.20 02-02-2009 (PEA) : changes FDR output filename for Kruskal-Wallis test (adss all event codes to the filename).
  - 1.21 02-03-2010 (PEA) : adds tf_substract_epfile (as in tfavg [18]) to substract evoked potential to all single trials before time-frequency transformation.
  - 1.22 11-05-2010 (PEA) : adds tfstat_wilcox_modified parameter to use baseline of 1st event in case of Wilcoxon test. Events must be paired.
  - 1.23 16-02-2011 (PEA) : updates to use cmake and free release of Elan. Fixes memory allocations.
  - 1.24 15-04-2011 (PEA) : changes Shell sort to Heap sort in FDR (faster algorithm).
  - 1.25 21-06-2011 (PEA) : uses threads to compute TF transform (TF transform function of libelansignal). Fixes number of frequencies problem.
  - 1.26 07-10-2011 (PEA) : adds test for at least 2 samples to test in Wilcoxon test.
  - 1.27 12-06-2012 (PEA) : minor modification for available memory for MAC OS X.
  - 1.28 17-07-2012 (PEA) : fixes an error in wilcox function of libstat when all samples are ties.
  - 1.29 25-03-2014 (PEA) : fixes an error appeared with 1.24 with 16 bits EEG data reference change. Change EEG data handling.
  - 1.30 22-09-2014 (PEA) : fixes an error baseline handling for Kruskal-Wallis test.

- ***Files***

  $ELANPATH/bin/tfstat

- ***See also***

  tfavg [18]

tfstatsync

- ***Description***

  Computes, in the time-frequency domain, a statistical test of cross-channel synchrony from an EEG file. The test is based on the randomization of shuffled channel pairs. The resulting probability values (p values) are stored in TF files **.vchan1.vchan2.p.stat.tf**. The synchrony test can be performed on the original or re-referenced channels. This allows to determine in the time-frequency domain whether the phase-locking factor represents a statistically significant synchrony.

- ***Usage***

tfstatsync myeegfile.eeg myeventfile.pos myparfile.par subsampling randomization [+v]

with :

- myeegfile.eeg : input EEG file name (with extension).
- myeventfile.pos : event position file (with extension) (this is usually the output pos file of eegavg, after artifact rejection).
- myparfile.par: text file containing computation parameters (with extension).
- subsampling : time-subsampling factor applied to the TF file created. In all cases, the time sample corresponding to the event onset (0 msec) is kept.
  1: all time-samples are saved,
  2: one over 2 samples are saved,
  3: one over 3 samples are saved,.…
- randomization : number of randomizations performed on shuffled pairs.
- option :
  +v : verbose mode on: all events are displayed during the processing progress. If omitted, verbose mode off.

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

fileprefix myfilename	Prefix of the output TF files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed.
tf_channel_ref 2 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is referenced to channel 2, channel 2 re-referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 10 10	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
nbsync_perchannel 2 1 0 0 0 0 0	Indicates, for each channel, the number of the channels for which synchrony will be computed with another channel:

	- 0: no synchrony computed from this channel, - ≠0: synchronies computed from this channel with the indicated number of channels (channel number specified in sync_list). The total number of values is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5 channels, and the synchrony will be computed with 2 channels for channel 1, and with 1 channel for channel 6.
sync_list 2 4 4	Indicates the channel ranks (starting from 1) with which synchrony will be computed, for each channel with a non-zero flag in nbsync_perchannel. According to tf_channel_ref, each channel may be re-referenced prior to computing synchrony. In this example, synchrony will be computed for the pairs channel 1-channel 2, channel 1-channel 4 and channel 2-channel 4.

- ***Examples***

  This par file should be identical to the one used to compute synchrony by tfsync. In this example, for both event codes (2 and 5), the time-frequency analysis will be performed from 18 to 80 Hz by steps of 2 H with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7. Probability values will be computed by randomization for the synchrony factors between channels: 1 vs 2, 1 vs 4, 2 vs 4. Before computing synchrony, those channels have been re-referenced, and the actual computed synchrony statistics are: (1-2) vs (2-3), (1-2) vs (4-5), and (2-3) vs (4-5).

  Output files :

Name	Comments#
myfilename.2.v1-v2.v2-v3.p.stat.tf myfilename.2.v1-v2.v4-v5.p.stat.tf myfilename.2.v2-v3.v4-v5.p.stat.tf myfilename.5.v1-v2.v2-v3.p.stat.tf myfilename.5.v1-v2.v4-v5.p.stat.tf myfilename.5.v2-v3.v4-v5.p.stat.tf	Values, in the time-frequency domain, of the probability (0 < p <1) of existence of cross-channel synchrony (phase-locking factor). One output file per event code and per channel pair.

- ***Comments***

- ***Current version***

  1.20 08-03-2017

- ***History***
  - 1.00 01-02-2001 (OB/CTB) : 1st documented version.
  - 1.10 09-12-2001 (PEA) : minor modification.
  - 1.11 04-02-2002 (PEA) : minor modification.
  - 1.12 17-06-2002 (PEA) : minor modification (event file reading).
  - 1.13 18-09-2003 (PEA) : minor modification.
  - 1.14 24-11-2003 (PEA) : supports 32 bits EEG.
  - 1.15 13-08-2007 (PEA) : minor modification (event file reading).
  - 1.16 21-05-2008 (PEA) : minor modification.
  - 1.17 17-02-2011 (PEA) : updates to use cmake and free release of Elan. Removes static memory allocations.
  - 1.18 12-06-2012 (PEA) : minor modification : output file name creation.
  - 1.19 23-09-2014 (PEA) : fix memory fault. Memory allocation change. Use thread version to compute TF.
  - 1.20 08-03-2017 (PEA) : remove events without enough samples in analysing period before computation.

- ***Files***

$ELANPATH/bin/tfstatsync

- ***See also***

  tfavg [18], tfsync [5]

tfsync

- ***Description***

  Computes, in the time-frequency domain using wavelet transform, the cross-channel synchrony factor from an eeg file. The test is based on the randomization of shuffled channel pairs. The resulting synchrony phase-locking factor is stored in tf files (**.vchan1.vchan2.pl.tf**), and the resulting cross-channel phase difference in a. tf file format (**.vchan1.vchan2.ph.tf**).

- ***Usage***

  tfsync myeegfile.eeg myeventfile.pos myparfile.par subsampling [+s] [+v]

  with :

  - myeegfile.eeg : input EEG file name (with extension).
  - myeventfile.pos : event position file (with extension) (this is usually the output pos file of eegavg, after artifact rejection).
  - myparfile.par: text file containing computation parameters (with extension).
  - subsampling : time-subsampling factor applied to the TF file created. In all cases, the time sample corresponding to the event onset (0 msec) is kept.
    1: all time-samples are saved,
    2: one over 2 samples are saved,
    3: one over 3 samples are saved,.…
  - options :
    +v : verbose mode on: all events are displayed during the processing progress. If omitted, verbose mode off.
    +s : all cross-channel synchronies or phase are stored in one single file.

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

fileprefix myfilename	Prefix of the output TF files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed.
tf_channel_ref 2 3 0 5 0 0 0	List of the new reference for each channel before processing (bipolar montage for instance): 0: no change of the reference, ≠0: electrode number (rank) to which the current channel should be re-referenced. The total number of values is N+2, N being the number of recorded channels in myfile.eeg file; the last 2 flags should be set to 0. If omitted, the channels are not modified. In this example, N=5, and channel 1 is referenced to channel 2, channel 2 re-referenced to channel 3, and channel 4 re-referenced to channel 5.
tf_freq_start 10 10	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
tf_freq_stop 80 80	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).

tf_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).
tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
nbsync_perchannel 2 1 0 0 0 0 0	Indicates, for each channel, the number of the channels for which synchrony will be computed with another channel: - 0: no synchrony computed from this channel, - ≠0: synchronies computed from this channel with the indicated number of channels (channel number specified in sync_list). The total number of values is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5 channels, and the synchrony will be computed with 2 channels for channel 1, and with 1 channel for channel 6.
sync_list 2 4 4	Indicates the channel ranks (starting from 1) with which synchrony will be computed, for each channel with a non-zero flag in nbsync_perchannel. According to tf_channel_ref, each channel may be re-referenced prior to computing synchrony. In this example, synchrony will be computed for the pairs channel 1-channel 2, channel 1-channel 4 and channel 2-channel 4.

- ***Examples***
  1. In this example, for both event codes (2 and 5), the time-frequency analysis will be performed from 18 to 80 Hz by steps of 2 H with a Blackman window having 100 samples for the rise- and for the fall-time, and with Morlet wavelets with a m ratio set to 7. Synchrony factors will be computed between channels: 1 vs 2, 1 vs 4, 2 vs 4. Before computing synchrony, those channels have been re-referenced, and the actual computed synchrony factors are: (1-2) vs (2-3), (1-2) vs (4-5), and (2-3) vs (4-5).

     Output files :

Name	Comments#
myfilename.2.v1-v2.v2-v3.pl.tf myfilename.2.v1-v2.v4-v5.pl.tf myfilename.2.v2-v3.v4-v5.pl.tf myfilename.5.v1-v2.v2-v3.pl.tf myfilename.5.v1-v2.v4-v5.pl.tf myfilename.5.v2-v3.v4-v5.pl.tf	Values, in the time-frequency domain, of cross-channel synchronies (phase-locking factor). One output file per event code and per channel pair.
myfilename.2.v1-v2.v2-v3.ph.tf myfilename.2.v1-v2.v4-v5.ph.tf myfilename.2.v2-v3.v4-v5.ph.tf myfilename.5.v1-v2.v2-v3.ph.tf myfilename.5.v1-v2.v4-v5.ph.tf myfilename.5.v2-v3.v4-v5.ph.tf	Values, in the time-frequency domain, of cross-channel phase difference (phase difference between first and second channels, in degrees). One output file per event code and per channel pair.

2. 2) When using the +s option, the cross-channel synchrony can be computed for all channels with respect to one particular channel. See an example of the **.par** file below for MEG, with synchrony of all channels computed with respect to channel #100 (including channel #100 with respect to itself):

nbsync_perchannel
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0

sync_list
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100

Output files :

Name	Comments#
myfilename.2.sync.pl.tf	Values, in the time-frequency domain, of cross-channel synchronies (phase-locking factor). One single output file per event code for all channel pairs.
myfilename.2.sync.ph.tf	Values, in the time-frequency domain, of cross-channel phase difference (phase difference between first and second channels, in degrees). One output file per event code for all channel pairs.

- ***Comments***

- ***Current version***

  1.21 12-06-2012

- ***History***
  - 1.00 02-02-2001 (OB/CTB) : 1st documented version.
  - 1.10 09-12-2001 (PEA) : minor modification.
  - 1.11 04-02-2002 (PEA) : minor modification.
  - 1.12 17-06-2002 (PEA) : minor modification (event file reading).
  - 1.13 18-09-2003 (PEA) : minor modification.
  - 1.14 24-11-2003 (PEA) : supports 32 bits EEG.
  - 1.15 29-09-2005 (PEA) : fixes quit when not enough samples after last event.
  - 1.16 13-04-2007 (OB) : supports large files (> 2GB). Adds +s option to save all pair of channel in one file.
  - 1.17 20-04-2007 (OB) : fixes TF header for +s option.

- 1.18 13-08-2007 (PEA) : minor modification.
- 1.19 22-02-2011 (PEA) : updates to use cmake and free release of Elan. Removes static memory allocations.
- 1.20 20-06-2011 (PEA) : uses threads to compute TF transform (TF transform function of libelansignal).
- 1.21 12-06-2012 (PEA) : minor modification : output file name creation.

- ***Files***

  $ELANPATH/bin/tfsync

- ***See also***

  tfavg [18], tfstatsync [20]

tfval

- ***Description***

  Computes from an EEG file the mean of the time-frequency powers for each single-trial (mean squared modulus of the wavelet transform) in a time-frequency window. For each epoch, it computes the mean value on a baseline window and substracts this value from the mean of the measure window. The output is stored in a text file format (one **.val.txt** file by event code).
  These computations requires a parameter file **.par** and an event position file **.pos**. Usually, the **.pos** file is the output of the averaging/rejection program eegavg [1].

- ***Usage***

  tfval myeegfile.eeg myposfile.pos myparfile.par [-e]

  with :

  - *myeegfile**.eeg***: input ***.eeg*** file to process (with extension).
  - *myposfile**.pos***: input event file (with extension).
  - *myparfile**.par***: text file containing computation parameters (with extension).
  - options:
    -e : don't print column titles in output file. If omitted, the titles of each columns are printed at the beginning of the file (on the first line).

- ***Fields of parameter file and examples***

fileprefix myfilename	Prefix of the output text files.
nb_eventcode 2	Number of event codes to process.
list_eventcode 2 5	List of the event codes to process.
prestim_nbsample 400 400	List of the numbers of samples in the prestimulus period; one value for each event code.
poststim_nbsample 1000 1000	List of the numbers of samples in the poststimulus period; one value for each event code; the total number of samples of the analysis is prestim_nbsample + poststim_nbsample + 1, the extra sample corresponds to the event itself.
tf_channel_flag 1 1 0 1 0 0 0	List of the channels to process: 1/0 for selected/unselected channels; the total number of flags is N+2, N being the number of recorded channels in myeegfile.eeg file; the last 2 flags should be set to 0. In this example, N=5, and only channels number 1, 2, 4 will be processed and stored in the output **.avg.tf** files.
single_freq_min 18 18	List of the starting frequencies (in Hz) for the time-frequency analysis (one value for each event code).
	List of the ending frequencies (in Hz) for the time-frequency analysis (one value for each event code).
single_freq_step 2 2	List of the frequency steps (in Hz) for the time-frequency analysis (one value for each event code).
tf_nb_sample_blackman 100 100	List of the number of samples in the rise or fall time period of the blackman window applied on the single trials before the wavelet transform (one value for each event code).

tf_wavelet_type 1 1	List of the types of wavelet used for the time-frequency analysis (one value for each event code): 1: Morlet wavelet. 2: Gabor wavelet.
tf_morlet_m 7 7	In case of Morlet wavelet, list of the m ratio used for the time-frequency analysis (one value for each event code): m=f0/sigmaf Suggested values for m: m>5, usually m=7. This determines the number of cycles of the wavelet.
tf_gabor_sigmat 100 100	In case of Gabor wavelet, list of the half-window durations of the wavelets (in msec) irrespective of the frequency band (one value for each event code).
single_lat_min 100 100	List of the starting latencies of the measure window in ms (one value for each event code).
single_lat_max 100 100	List of the ending latencies of the measure window in ms (one value for each event code).
single_base_min -100 -100	List of the starting latencies of the baseline window in ms (one value for each event code).
single_base_max -50 -50	List of the ending latencies of the baseline window in ms (one value for each event code).

- ***Examples***

- ***Comments***

- ***Current version***

  1.06 12-06-2012

- ***History***
  - 1.00 01-08-2001 (OB/CTB/PEA) : 1st documented version.
  - 1.01 17-06-2002 (PEA) : minor modification (event file reading).
  - 1.02 26-09-2003 (PEA) : minor modification.
  - 1.03 05-01-2004 (PEA) : supports 32 bits EEG.
  - 1.04 05-02-2004 (PEA) : minor modification.
  - 1.05 13-09-2011 (PEA) : updates to use cmake and free release of Elan. Removes static allocations. Uses threads to compute TF transform (TF transform function of libelansignal).
  - 1.06 12-06-2012 (PEA) : minor modification : output file name creation.

- ***Files***

  $ELANPATH/bin/tfval

- ***See also***

  tfavg [18], tfmval [26]

Lyon Neuroscience Research Center - Brain Dynamic and Cognition team

CRNL

[2] http://elan.lyon.inserm.fr/?q=sites/default/files/ctf275_meg.par
[3] http://elan.lyon.inserm.fr/?q=eegchref
[4] http://elan.lyon.inserm.fr/?q=tfep
[5] http://elan.lyon.inserm.fr/?q=tfsync
[6] http://elan.lyon.inserm.fr/?q=tfavgdiff
[7] http://elan.lyon.inserm.fr/?q=tfavgread
[8] http://elan.lyon.inserm.fr/?q=tfavgavg
[9] http://elan.lyon.inserm.fr/?q=tfavgsmooth
[10] http://elan.lyon.inserm.fr/?q=tfavgprofilet
[11] http://elan.lyon.inserm.fr/?q=tfavgprofilef
[12] http://elan.lyon.inserm.fr/?q=tfavgwilcox
[13] http://elan.lyon.inserm.fr/?q=tfavgbline
[14] http://elan.lyon.inserm.fr/?q=tfavgchannel
[15] http://elan.lyon.inserm.fr/?q=epresample
[16] http://elan.lyon.inserm.fr/?q=tfstat
[17] http://elan.lyon.inserm.fr/?q=tfavgmask
[18] http://elan.lyon.inserm.fr/?q=tfavg
[19] http://elan.lyon.inserm.fr/?q=tfcmpsync
[20] http://elan.lyon.inserm.fr/?q=tfstatsync
[21] http://elan.lyon.inserm.fr/?q=tfcmppower
[22] http://www.ncbi.nlm.nih.gov/pubmed/11829299
[23] http://elan.lyon.inserm.fr/?q=tfavgval
[24] http://elan.lyon.inserm.fr/?q=eegmval
[25] http://elan.lyon.inserm.fr/?q=eegstat
[26] http://elan.lyon.inserm.fr/?q=tfmval