FieldTrip Cheatsheet v2024.01

Description of the data input processes

Brought to you by “The FUNNI Pages”,
The FUNctional NeuroImaging newsletter
www.thefunnipages.com

ft_definetrial

Use as: [cfg] = ft_definetrial(cfg) where the cfg structure should contain

cfg.trialdef = structure with the details of trial definition, see below

cfg.trialfun = string with the function name, see below (default = 'ft_trialfun_general')

cfg.representation = 'numeric' or 'table', determines how the trial definition is returned (default is 'automatic')

cfg.dataset = string with the dataset file name

or

cfg.headerfile = string with the header file name

cfg.datafile = string with the data file name

Optional:

cfg.headerformat = string, (default is 'automatic')

cfg.dataformat = string, (default is 'automatic')

cfg.eventformat = string, (default is 'automatic')

see FT_FILETYPE for format def'n

ft_preprocessing

Use as: [data] = ft_preprocessing(cfg) for initial input and prep-processing of data

Required fields:

cfg.dataset = string with the dataset file name

OR

cfg.datafile = string with the data file name

cfg.headerfile = string with the header file name

cfg.trl = Nx3 matrix with the trial definition

cfg.padding = length (in seconds) to which the trials are padded for filtering (default = 0)

cfg.padtype = string, type of padding (default: 'data' padding or 'mirror', depending on feasibility)

cfg.continuous = 'yes' or 'no'

ft_preprocessing

Use as: [data] = ft_preprocessing(cfg,data) for secondary pre-processing of data already input

Common fields:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),

cfg.lpfilter = 'no' or 'yes' lowpass filter (default = 'no')

cfg.hpfilter, cfg.bpfilter, cfg.bsfilter, cfg.dftfilter, cfg.medianfilter = 'no' or 'yes' (default = 'no')

cfg.lpfreq, cfg.hpfreq = cutoff freqs in Hz

cfg.bpfreq, cfg.bsfreq = filter band frequency range, specified as [low high] in Hz

cfg.dftfreq = line noise frequencies in Hz for DFT filter (default = [50 100 150])

cfg.lpfiltord, cfg.hpfiltord, cfg.bpfiltord, cfg.bsfiltord = filter order

cfg.lpfilttype, cfg.hpfilttype, cfg.bpfilttype, cfg.bsfilttype = digital filter type, (default = 'but')

cfg.lpfiltdir, cfg.hpfiltdir, cfg.bpfiltdir, cfg.bsfiltdir = filter direction, 'twopass' (default), 'onepass' or 'onepass-reverse' or 'onepass-zerophase' (default for firws) or 'onepass-minphase' (firws, non-linear!)

cfg.lpinstabilityfix, cfg.hpinstabilityfix, cfg.bpinstabilityfix, cfg.bsinstabilityfix = deal with filter instability, 'no', 'reduce', 'split' (default = 'no')

cfg.lpfiltdf, cfg.hpfiltdf, cfg.bpfiltdf, cfg.bsfiltdf = bandstop transition width (firws, overrides order)

cfg.lpfiltwintype, cfg.hpfiltwintype, cfg.bpfiltwintype, cfg.bsfiltwintype = bandstop window type, 'hann' or 'hamming' (default) or 'blackman' or 'kaiser' (firws)

cfg.lpfiltdev, cfg.hpfiltdev, cfg.bpfiltdev, cfg.bsfiltdev = bandstop max passband deviation (firws with 'kaiser' window, default 0.001 set in low-level function)

cfg.dftreplace = 'zero' or 'neighbour', method used to reduce line noise, 'zero' implies DFT filter, 'neighbour' implies spectrum interpolation (default = 'zero')

cfg.dftbandwidth = bandwidth of line noise frequencies, applies to spectrum interpolation, in Hz (default = [1 2 3])

cfg.dftneighbourwidth = bandwidth of frequencies neighbouring line noise frequencies, applies to spectrum interpolation, in Hz (default = [2 2 2])

cfg.plotfiltresp = 'no' or 'yes', plot filter responses (firws, default = 'no')

cfg.usefftfilt = 'no' or 'yes', use fftfilt instead of filter (firws, default = 'no')

cfg.medianfiltord = length of median filter (default = 9)

cfg.demean = 'no' or 'yes', whether to apply baseline correction (default = 'no')

cfg.baselinewindow = [begin end] in seconds, the default is the complete trial (default = 'all')

cfg.detrend = 'no' or 'yes', remove linear trend from the data (done per trial) (default = 'no')

cfg.polyremoval = 'no' or 'yes', remove higher order trend from the data (done per trial) (default = 'no')

cfg.polyorder = polynome order for poly trend removal (default = 2; note that all lower-order trends will also be removed when using cfg.polyremoval)

cfg.derivative = 'no' or 'yes', computes the first order derivative of the data (default = 'no')

cfg.hilbert = 'no', 'abs', 'complex', 'real', 'imag', 'absreal', 'absimag' or 'angle' (default = 'no')

cfg.rectify = 'no' or 'yes' (default = 'no')

cfg.precision = 'single' or 'double' (default = 'double')

cfg.absdiff = 'no' or 'yes', computes absolute derivative (i.e. first derivative then rectify)

ft_redefinetrial

Use as: [data] = ft_redefinetrial(cfg, data)

Common fields:

cfg.trials = 'all' or a selection given as a 1xN vector (default = 'all')

cfg.minlength = length in seconds, can be 'maxperlen' (default = [])

cfg.offset = single number or Nx1 vector, by how many samples should the time axes be shifted.

cfg.toilim = [tmin tmax] to specify a latency window in seconds, can be Nx2 vector

cfg.begsample = single number or Nx1 vector, expressed in samples relative to the start of the input trial

cfg.endsample = single number or Nx1 vector, expressed in samples relative to the start of the input trial

cfg.trl = Nx3 matrix with the trial definition, see FT_DEFINETRIAL

cfg.length = number (in seconds) that specifies the length of the required snippets

cfg.overlap = number between 0 and 1 (exclusive) specifying the fraction of overlap between snippets

ft_appenddata

Use as: [data] = ft_appenddata(cfg, data1, data2, data3, ...)

Options:

cfg.keepsampleinfo = 'yes', 'no', 'ifmakessense' (default = 'ifmakessense')

OTHER INPUT PROCESSING:

ft_artifact_eog
ft_artifact_jump
ft_artifact_muscle
ft_rejectartifact
ft_rejectvisual
ft_resampledata
ft_channelrepair
ft_recodeevent

Description of the event-related processes

ft_timelockanalysis

Use as [timelock] = ft_timelockanalysis(cfg, data)

Common fields:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'), see FT_CHANNELSELECTION for details
cfg.latency = [begin end] in seconds, or 'all', 'minperiod', 'maxperiod', 'prestim', 'poststim' (default = 'all')
cfg.trials = 'all' or a selection given as a 1xN vector (default = 'all')
cfg.keeptrials = 'yes' or 'no', return individual trials or average (default = 'no')
cfg.nanmean = string, can be 'yes' or 'no' (default = 'yes')
cfg.normalizevar = 'N' or 'N-1' (default = 'N-1')
cfg.covariance = 'no' or 'yes' (default = 'no')
cfg.covariancewindow = [begin end] in seconds, or 'all', 'minperiod', 'maxperiod', 'prestim', 'poststim' (default = 'all')
cfg.removemean = 'yes' or 'no', for the covariance computation (default = 'yes')

Optional:

cfg.headerformat = string, (default is 'automatic')
cfg.dataformat = string, (default is 'automatic')
cfg.eventformat = string, (default is 'automatic')
see FT_FILETYPE for format def'n

ER plotting routines:

ft_singleplotER
ft_topoplotER
ft_multiplotER

ft_timelockgrandaverage

Use as [grandavg] = ft_timelockgrandaverage(cfg, avg1, avg2, avg3, ...)

Common fields:

cfg.method = string, 'across' or 'within' (default = 'across'), see below for details
cfg.parameter = string, which parameter to average (default = 'avg')
cfg.channel = Nx1 cell-array with selection of channels (default = 'all'), see FT_CHANNELSELECTION for details
cfg.latency = [begin end] in seconds or 'all' (default = 'all')
cfg.keepindividual = string, 'yes' or 'no' (default = 'no')
cfg.nanmean = string, can be 'yes' or 'no' (default = 'yes')
cfg.normalizevar = string, 'N' or 'N-1' (default = 'N-1')

ft_timelockstatistics

Use as [stat] = ft_timelockstatistics(cfg, timelock1, timelock2, ...)

Common fields:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),
cfg.latency = [begin end] in seconds or 'all' (default = 'all')
cfg.avgoverchan = 'yes' or 'no' (default = 'no')
cfg.avgovertime = 'yes' or 'no' (default = 'no')
cfg.parameter = string (default = 'trial' or 'avg')
cfg.method = 'montecarlo', 'analytic', 'stats', or 'mvpa'

Description of the frequency and time-frequency processes

ft_freqanalysis

Use as [freq] = ft_freqanalysis(cfg, data)

Common options:

cfg.method = different methods of calculating the spectra, such as:'mtmfft', 'mtmconvol', 'wavelet', 'tfr', 'mvar', 'superlet', 'irasa', 'hilbert'

Cfg.output = 'pow', 'powandcsd', 'fourier', 'fractal','original','fooof','fooof_aperiodic','fooof_peaks'

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),

cfg.channelcmb = Mx2 cell-array with selection of channel pairs (default = {'all' 'all'}), see FT_CHANNELCOMBINATION for details

cfg.trials = 'all' or a selection given as a 1xN vector (default = 'all')

cfg.keeptrials = 'yes' or 'no', return individual trials or average (default = 'no')

cfg.keeptapers = 'yes' or 'no', return individual tapers or average (default = 'no')

cfg.pad = number, 'nextpow2', or 'maxperlen' (default), length in seconds

cfg.padtype = string, type of padding (default 'zero', see ft_preproc_padding)

cfg.polyremoval = number (default = 0)

ft_freqbaseline

Use as [freq] = ft_freqbaseline(cfg, freq)

Config options:

cfg.baseline = [begin end] (default = 'no'), alternatively an Nfreq x 2 matrix can be specified, that provides frequency specific baseline windows.

cfg.baselinetype = 'absolute', 'relative', 'relchange', 'normchange', 'db', 'vssum' or 'zscore' (default = 'absolute')

cfg.parameter = field for which to apply baseline normalization, or cell-array of strings to specify multiple fields to normalize (default = 'powspctrm')

ft_freqgrandaverage

Use as [grandavg] = ft_freqgrandaverage(cfg, freq1, freq2, freq3...)

Config options:

cfg.keepindividual = 'yes' or 'no' (default = 'no')

cfg.foilim = [fmin fmax] or 'all', to specify a subset of frequencies (default = 'all')

cfg.toilim = [tmin tmax] or 'all', to specify a subset of latencies (default = 'all')

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'), see FT_CHANNELSELECTION for details

cfg.parameter = string or cell-array of strings indicating which parameter(s) to average. default is set to 'powspctrm', if it is present in the data.

ft_freqdescriptives

Use as: [freq] = ft_freqdescriptives(cfg, freq)
[freq] = ft_freqdescriptives(cfg, freqmvar)

Config options:

cfg.variance = 'yes' or 'no', estimate standard error in the standard way (default = 'no')

cfg.jackknife = 'yes' or 'no', estimate standard error by means of the jack-knife (default = 'no')

cfg.keeptrials = 'yes' or 'no', estimate single trial power (useful for fourier data) (default = 'no')

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),

cfg.trials = 'all' or a selection given as a 1xN vector (default = 'all')

cfg.frequency = [fmin fmax] or 'all', to specify a subset of frequencies (default = 'all')

cfg.latency = [tmin tmax] or 'all', to specify a subset of latencies (default = 'all')

ft_freqstatistics

Use as: [stat] = ft_freqstatistics(cfg, freq1, freq2, ...)

Config options:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),

cfg.latency = [begin end] in seconds or 'all' (default = 'all')

cfg.frequency = [begin end], can be 'all' (default = 'all')

cfg.avgoverchan = 'yes' or 'no' (default = 'no')

cfg.avgovertime = 'yes' or 'no' (default = 'no')

cfg.avgoverfreq = 'yes' or 'no' (default = 'no')

cfg.parameter = string (default = 'powspctrm')

cfg.correctm='cluster', if so then the following is required

cfg.neighbours = neighbourhood structure

cfg.method = 'montecarlo','analytic', 'stats', 'crossvalidate'

cfg.design = Nxnum observations: design matrix

Description of the source analysis processes

ft_dipolefitting

Use as [source] = ft_dipolefitting(cfg, data)

Common options:

cfg.numdipoles = number, default is 1
cfg.symmetry = 'x', 'y' or 'z' symmetry for two dipoles, can be empty (default = [])
cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),
cfg.gridsearch = 'yes' or 'no', (default = 'yes')
cfg.nonlinear = 'yes' or 'no', (default = 'yes')
cfg.dip.pos = initial dipole position, matrix Ndipoles x 3
cfg.latency = [begin end] in seconds or 'all' (default = 'all')
cfg.model = 'moving' or 'regional'
cfg.headmodel = structure with volume conduction model
cfg.elec = structure with electrode positions or filename
cfg.grad = structure with gradiometer def. or filename

For grid search:

cfg.xgrid = vector (e.g. -20:1:20) or 'auto' (default = 'auto')
cfg.ygrid = vector (e.g. -20:1:20) or 'auto' (default = 'auto')
cfg.zgrid = vector (e.g. 0:1:20) or 'auto' (default = 'auto')
cfg.resolution = number (e.g. 1 cm) for auto grid generation

ft_sourcedescriptives

Use as: [source] = ft_sourcedescriptives(cfg, source)

Config options:

cfg.cohmethod = 'regular', 'lambda1', 'canonical'
cfg.powmethod = 'regular', 'lambda1', 'trace', 'none'
cfg.supmethod = 'chan_dip', 'chan', 'dip', 'none' (default)
cfg.projectmom = 'yes' or 'no' (default = 'no')
cfg.eta = 'yes' or 'no' (default = 'no')
cfg.kurtosis = 'yes' or 'no' (default = 'no')
cfg.keeptrials = 'yes' or 'no' (default = 'no')
cfg.keepcsd = 'yes' or 'no' (default = 'no')
cfg.keepnoisecsd = 'yes' or 'no' (default = 'no')
cfg.keepmom = 'yes' or 'no' (default = 'yes')
cfg.keepnoisemom = 'yes' or 'no' (default = 'yes')
cfg.resolutionmatrix = 'yes' or 'no' (default = 'no')
cfg.feedback = 'no', 'text' (default), 'textbar', 'gui'

The following option only applies to timecourses.
cfg.flipori = 'yes' or 'no' (default = 'no')
The following option only applies to single-trial timecourses.
cfg.fixedori = 'within_trials' or 'over_trials' (default = 'over_trials')

ft_sourceanalysis

Use as [source] = ft_sourceanalysis(cfg, freq), or
[source] = ft_sourceanalysis(cfg, timelock)

Config options:

cfg.method = 'lcmv,'sam','dics','pcc','mne','rv','music','sloreta','eloreta'
Cfg.xgrid, cfg.ygrid,cfg.zgrid, cfg.resolution (as for ft_dipolefitting)
cfg.sourcemodel.pos, cfg.sourcemodel.tri,
cfg.sourcemodel.inside, cfg.sourcemodel.dim
cfg.sourcemodel.filter, cfg.sourcemodel.leadfield

Many other options are available, click the link to see them all

ft_sourcestatistics

Use as: [stat] = ft_sourcestatistics(cfg, source1, source2, ...)

Config options:

cfg.parameter = string, describing the functional data to be processed, e.g. 'pow', 'nai' or 'coh'
cfg.method = 'montecarlo', 'analytic', 'stats'

ft_sourceinterpolate

Use as:

[interp] = ft_sourceinterpolate(cfg, source, anatomy) OR
[interp] = ft_sourceinterpolate(cfg, stat, anatomy)

Config options:

cfg.parameter = string or cell-array with the functional parameter(s) to be interpolated
cfg.downsample = integer (default = 1, i.e. no DS)
cfg.interpmethod = 'nearest', 'linear', 'cubic', 'spline', 'sphere_avg', 'sphere_weighteddistance', or 'smudge' (default = 'linear for interpolating two 3D volumes, 'nearest' for all other cases)

ft_sourcegrandaverage

Use as [grandavg] = ft_sourcegrandaverage(cfg, source1, source2, ...)

Config options:

cfg.parameter = string, describing the functional data to be processed, e.g. 'pow', 'nai' or 'coh'
cfg.keepindividual = 'no' or 'yes'

ft_sourceparcellate

Use as: output = ft_sourceparcellate(cfg, source, parcellation)

Config options:

cfg.method = 'mean','median, 'eig', 'min', 'max', 'maxabs', 'std' (default = 'mean')
cfg.parcellation = string, fieldname that contains the desired parcellation
cfg.parameter = cell-array with strings, fields that should be parcellated (default = 'all')

ft_prepare_leadfield

Use as: [sourcemodel] = ft_prepare_leadfield(cfg, data)

Config options:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
Cfg.xgrid, cfg.ygrid,cfg.zgrid, cfg.resolution (as for ft_dipolefitting)
cfg.sourcemodel.pos, cfg.sourcemodel.inside,
cfg.sourcemodel.dim
cfg.headmodel = structure with volume conduction model
cfg.elec = structure with electrode positions or filename,
cfg.grad = structure with gradiometer definition or filename

Description of the statistical analysis processes

ft_timelockstatistics

Use as [stat] = ft_timelockstatistics(cfg, timelock1, timelock2, ...)

Common options:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),
cfg.latency = [begin end] in seconds or 'all' (default = 'all')
cfg.avgoverchan = 'yes' or 'no' (default = 'no')
cfg.avgovertime = 'yes' or 'no' (default = 'no')
cfg.parameter = string (default = 'trial' or 'avg')
cfg.method = 'montecarlo,'analytic, 'stats, 'mvpa'

ft_sourcestatistics

Use as: [stat] = ft_sourcestatistics(cfg, source1, source2, ...)

Config options:

cfg.parameter = string, describing the functional data to be processed, e.g. 'pow', 'nai' or 'coh'
cfg.method = 'montecarlo', 'analytic', 'stats'

ft_freqstatistics

Use as: [stat] = ft_freqstatistics(cfg, freq1, freq2, ...)

Config options:

cfg.channel = Nx1 cell-array with selection of channels (default = 'all'),
cfg.latency = [begin end] in seconds or 'all' (default = 'all')
cfg.frequency = [begin end], or 'all' (default = 'all')
cfg.avgoverchan = 'yes' or 'no' (default = 'no')
cfg.avgovertime = 'yes' or 'no' (default = 'no')
cfg.avgoverfreq = 'yes' or 'no' (default = 'no')
cfg.parameter = string (default = 'powspctrm')
cfg.correctm ='cluster', then the following is required
    cfg.neighbours = neighbourhood structure
cfg.method = 'montecarlo', 'analytic, 'stats', 'crossvalidate'
cfg.design = Nxnum observations: design matrix

Lower level statistics functions Group 1

Use these functions by calling one of the high-level statistics functions as
  [stat] = ft_timelockstatistics(cfg, timelock1, timelock2, ...)
  [stat] = ft_freqstatistics(cfg, freq1, freq2, ...)
  [stat] = ft_sourcestatistics(cfg, source1, source2, ...)

with one of the following configuration options:
cfg.statistic = 'ft_statfun_actvsblT'
cfg.statistic = 'ft_statfun_depsamplesFunivariate'
cfg.statistic = 'ft_statfun_depsamplesFmultivariate'
cfg.statistic = 'ft_statfun_depsamplesT'
cfg.statistic = 'ft_statfun_depsamplesregrT'
cfg.statistic = 'ft_statfun_indepsamplesF'
cfg.statistic = 'ft_statfun_indepsamplesT'
cfg.statistic = 'ft_statfun_indepsamplesZcoh'
cfg.statistic = 'ft_statfun_indepsamplesregrT'

Config options:

cfg.computestat = 'yes' or 'no', (default='yes')
cfg.computecritval = 'yes' or 'no', (default='no')
cfg.computeprob = 'yes' or 'no', (default='no')

The following options are relevant if cfg.computecritval='yes' and/or cfg.computeprob='yes':
cfg.alpha = critical level of the stat. test (default=0.05)
cfg.tail = -1, 0, or 1, left, two-sided, or right (default=1)
  cfg.tail in combination with cfg.computecritval='yes' determines whether the critical value is computed at
  quantile cfg.alpha (with cfg.tail=-1), at
  quantiles cfg.alpha/2 and (1-cfg.alpha/2) (with cfg.tail=0), or at
  quantile (1-cfg.alpha) (with cfg.tail=1)

The experimental design is specified as:
  cfg.ivar = row number of the design that contains the independent variable, i.e. the predictor (default=1)
  cfg.uvar = unit variable, row number of design that contains the labels of the units-of-observation (default=2)

Lower level statistics functions Group 2

Use these functions by calling one of the high-level statistics functions as
  [stat] = ft_timelockstatistics(cfg, timelock1, timelock2, ...)
  [stat] = ft_freqstatistics(cfg, freq1, freq2, ...)
  [stat] = ft_sourcestatistics(cfg, source1, source2, ...)

with one of the following configuration options:
cfg.statistic = 'ft_statfun_diff_itc'
cfg.statistic = 'ft_statfun_diff_itc'

Config options:

cfg.complex = string, 'diffabs' (default) or 'absdiff' (for ft_statfun_diff_itc only)

The experimental design is specified as:
cfg.ivar = row number of the design that contains the labels of the conditions to be compared (default=1)
(The labels for the independent variable should be specified as the number 1 and 2.)

ft_mvaranalysis

Use as: [mvardata] = ft_mvaranalysis(cfg, data)

Config options:

cfg.method = 'biosig' (default) or 'bsmart'
cfg.mvarmethod = scalar (for 'biosig', default 2).
cfg.order = scalar(default=10)
cfg.channel = 'all' (default) or list of channels
cfg.channelcmb = two-column cell-array
cfg.keeptrials = 'no' (default) or 'yes'
cfg.jackknife = 'no' (default) or 'yes'
cfg.zscore = 'no' (default) or 'yes'
cfg.demean = 'yes' (default) or 'no'
cfg.ems = 'no' (default) or 'yes'

Description of the plotting and display processes

ft_clusterplot

Use as: ft_clusterplot(cfg, stat)

Common options:

cfg.alpha = number, highest cluster p-value to be plotted max 0.3 (default = 0.05)
cfg.highlightseries = 1x5 cell-array, with 'on', 'labels' or 'numbers' (default {'on', 'on', 'on', 'on', 'on'} for p < [0.01 0.05 0.1 0.2 0.3]
cfg.highlightsymbolseries = 1x5 vector, (default ['*', 'x', '+', 'o', '.'] for p < [0.01 0.05 0.1 0.2 0.3]
cfg.highlightsizeseries = 1x5 vector,(default [6 6 6 6 6] for p < [0.01 0.05 0.1 0.2 0.3])
cfg.highlightcolorpos = color of highlight marker for positive clusters (default = [0 0 0])
cfg.highlightcolorneg = color of highlight marker for negative clusters (default = [0 0 0])
cfg.subplotsize = layout of subplots ([h w], default [3 5])
cfg.saveaspng = string, filename for figures (default = 'no')
cfg.visible = string, 'on' or 'off' (default = 'on')
cfg.position = location|size of figure, specified as [left bottom width height] (default is automatic)
cfg.renderer = string: 'opengl', 'zbuffer', or 'painters' (default is automatic)
cfg.toi = vector, or 'all' (default)

ft_layoutplot

Use as: ft_layoutplot(cfg, data)

Config options:

cfg.layout = filename containg the layout
cfg.rotate = number of degrees (default = [])
cfg.projection = string: 'stereographic', 'orthographic', 'polar', 'gnomic' or 'inverse' (default = 'orthographic')
cfg.elec = structure of electrode positions or filename
cfg.grad = structure of gradiometer definition or filename
cfg.opto = structure with optode definition or filename
cfg.output = filename (default = [])
cfg.montage = 'no' or a montage structure (default = 'no')
cfg.image = filename, use image to construct a layout
cfg.box = string, 'yes' or 'no' whether box should be plotted around electrode (default = 'yes')
cfg.mask = string, 'yes' or 'no' whether the mask should be plotted (default = 'yes')
cfg.visible = string, 'on' or 'off' whether figure will be visible (default = 'on')
cfg.figure = 'yes' or 'no', whether to open a new figure. (default = 'yes')
cfg.position = location and size of the figure, specified as [left bottom width height] (default is automatic)
cfg.renderer = string: 'opengl', 'zbuffer', 'painters' (default is automatic)

ft_movieplotER

Use as: ft_movieplotER(cfg, timelock)

Config options:

cfg.parameter = string: parameter to color code (default = 'avg')
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.zlim = plotting limits for color dimension, 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.speed = number, speed for interactive (default = 1)
cfg.samperframe = number, samples per frame for non-interactive mode (default = 1)
cfg.framespersec = number, frames per second for non-interactive mode (default = 5)
cfg.framesfile = 'string' or empty, filename (default = [])
cfg.layout = specification of the layout
cfg.interpolatenan = string 'yes', 'no' interpolate over channels containing NaNs (default = 'yes')
cfg.colormap = string, or Nx3 matrix
cfg.baseline = 'yes','no' or [time1 time2] (default = 'no')
cfg.baselinetype = 'absolute' or 'relative' (default = 'absolute')
cfg.colorbar = 'yes', 'no' (default = 'no')
cfg.colorbartext = string indicating the text next to colorbar
cfg.renderer = string, 'opengl', 'zbuffer', 'painters' (default is automatic)

ft_movieplotTFR

Use as: ft_movieplotTFR(cfg, data)

Config options:

cfg.parameter = string, parameter to color code (default = 'avg')
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.ylim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.zlim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.speed = number, initial speed for interactive mode (default = 1)
cfg.samperframe = number, samples per frame for non-interactive mode (default = 1)
cfg.framespersec = number, frames per second for non-interactive mode (default = 5)
cfg.framesfile = 'string' or empty, filename (default = [])
cfg.moviefreq = number (default = [])
cfg.movietime = number (default = [])
cfg.layout = specification of the layout
cfg.interpolatenan = string 'yes', 'no' (default = 'yes')
cfg.colormap = string, or Nx3 matrix
cfg.interactive = 'no' or 'yes'
cfg.baseline = 'yes','no' or [time1 time2] (default = 'no'),
cfg.baselinetype = 'absolute', 'relative', 'relchange', 'normchange', 'db' or 'zscore' (default = 'absolute')
cfg.colorbar = 'yes', 'no' (default = 'no')

ft_multiplotER

Use as:ft_multiplotER(cfg, data, data2, ..., dataN)

Config options:

cfg.parameter = field to be plotted on y-axis, for example 'avg', 'powspctrm' or 'cohspctrm' (default is automatic)
cfg.maskparameter = field in the first dataset to be used for masking significant data
cfg.maskstyle = style used for masking of data, 'box', 'thickness' or 'saturation' (default = 'box')
cfg.maskfacealpha = mask transparency value (0 -1)
cfg.xlim, cfg.ylim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [xmin xmax] (default = 'maxmin')
cfg.gradscale = number, scaling to apply to MEG grads
cfg.magscale = number, scaling to apply to MEG mags
cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
cfg.refchannel = name of reference channel or 'gui'
cfg.baseline = 'yes', 'no' or [time1 time2] (default = 'no'),
cfg.trials = 'all' or selection in1xN vector (default = 'all')
cfg.axes, cfg.box, cfg.showlabels, cfg.showoutline, cfg.showscale, cfg.showcomment = string, 'yes' or 'no'
cfg.comment = string of text (default = date + limits)
cfg.limittext = use text instead of cfg.comment
cfg.fontsize = size of comment and labels (default = 8)
cfg.interactive = 'yes' or 'no', (default = 'yes')
cfg.figure = 'yes' or 'no', (default = 'yes')
cfg.position = [left bottom width height]
cfg.renderer = string, 'opengl', 'zbuffer', 'painters'
cfg.colorgroups = 'sequential', 'allblack', 'labelcharN', 'chantype' or vector (default = 'condition')
cfg.linestyle = linestyle/marker type (default = '-')
cfg.linewidth = linewidth in points (default = 0.5)
cfg.linecolor = color(s) used for plotting the dataset(s).
cfg.directionality = 'inflow' or 'outflow'
cfg.layout = specify the channel layout for plotting
cfg.select = 'intersect' or 'union' (default = 'intersect')
cfg.viewmode = 'topographic' or 'butterfly', (default = 'topographic')

The following options for the scaling of the EEG, EOG, ECG, EMG, MEG and NIRS channels:

cfg.eegscale, cfg.eogscale, cfg.ecgscale, cfg.emgscale, cfg.megscale, cfg.gradscale, cfg.magscale, cfg.nirsscale = number, scaling to apply to the relevant channels
cfg.mychanscale = number, scaling to apply to the channels specified in cfg.mychan
cfg.mychan = Nx1 cell-array with selection of channels
cfg.chanscale = Nx1 vector with scaling factors, one per channel specified in cfg.channel

Description of the plotting and display processes Part 2

ft_multiplotTFR

Use as:ft_multiplotTFR(cfg, data)

Config options:

cfg.parameter = field to be plotted on y-axis, for example 'avg', 'powspctrm' or 'cohspctrm' (default is automatic)
cfg.maskparameter = field in the first dataset to be used for masking significant data
cfg.maskstyle = style used for masking of data, 'box', 'thickness' or 'saturation' (default = 'box')
cfg.maskfacealpha = mask transparency value (0 -1)
cfg.xlim, cfg.ylim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [xmin xmax] (default = 'maxmin')
cfg.gradscale = number, scaling to apply to MEG grads
cfg.magscale = number, scaling to apply to MEG mags
cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
cfg.refchannel = name of reference channel or 'gui'
cfg.baseline = 'yes', 'no' or [time1 time2] (default = 'no'),
cfg.trials = 'all' or selection in1xN vector (default = 'all')
cfg.axes, cfg.box, cfg.showlabels, cfg.showoutline, cfg.showscale, cfg.showcomment = string, 'yes' or 'no'
cfg.comment = string of text (default = date + limits)
cfg.limittext = use text instead of cfg.comment
cfg.fontsize = size of comment and labels (default = 8)
cfg.interactive = 'yes' or 'no', (default = 'yes')
cfg.figure = 'yes' or 'no', (default = 'yes')
cfg.position = [left bottom width height]
cfg.renderer = string, 'opengl', 'zbuffer', 'painters'
cfg.colorgroups = 'sequential', 'allblack', 'labelcharN', 'chantype' or vector (default = 'condition')
cfg.linestyle = linestyle/marker type (default = '-')
cfg.linewidth = linewidth in points (default = 0.5)
cfg.linecolor = color(s) used for plotting the dataset(s).
cfg.directionality = 'inflow' or 'outflow'
cfg.layout = specify the channel layout for plotting
cfg.select = 'intersect' or 'union' (default = 'intersect')
cfg.viewmode = 'topographic' or 'butterfly', (default = 'topographic')
The following options for the scaling of the EEG, EOG, ECG, EMG, MEG and NIRS channels:
cfg.eegscale, cfg.eogscale, cfg.ecgscale, cfg.emgscale, cfg.megscale, cfg.gradscale, cfg.magscale, cfg.nirsscale = number, scaling to apply to the relevant channels
cfg.mychanscale = number, scaling to apply to the channels specified in cfg.mychan
cfg.mychan = Nx1 cell-array with selection of channels
cfg.chanscale = Nx1 vector with scaling factors, one per channel specified in cfg.channel

ft_neighbourplot

Use as: ft_neighbourplot(cfg) or ft_neighbourplot(cfg, data)

Config options:

cfg.verbose = string, 'yes' or 'no'
cfg.neighbours = neighbourhood structure
cfg.enableedit = string, 'yes' or 'no', (default = 'no')
cfg.visible = string, 'on' or 'off' (default = 'on')
cfg.figure = 'yes' or 'no', (default = 'yes')
cfg.position = [left bottom width height] (default is automatic)
cfg.renderer = string, 'opengl', 'zbuffer', 'painters'

and either one of the following options
cfg.layout = filename of the layout
cfg.elec = structure with electrode positions or filename
cfg.grad = structure with grad. definition or filename
cfg.opto = structure with grad. definition or filename,

The following data fields may also be used
data.elec = structure with electrode positions
data.grad = structure with gradiometer definition
data.opto = structure with optode definition

ft_singleplotER

Use as: ft_singleplotER(cfg, data) or
ft_singleplotER(cfg, data1, data2, ..., datan)

Config options:

cfg.parameter = 'avg', 'powspctrm' or 'cohspctrm' (default is automatic)
cfg.maskparameter = field in the first dataset (default = [])
cfg.maskstyle ='box', 'thickness' or 'saturation' (default = 'box')
cfg.maskfacealpha = value between 0 and 1
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.ylim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [ymin ymax] (default = 'maxmin')
cfg.channel = Nx1 cell-array (default = 'all')
cfg.title = string, title of plot
cfg.showlegend = 'yes' or 'no', (default = 'no')
cfg.refchannel = name of reference channel for visualising connectivity, can be 'gui'
cfg.baseline = 'yes', 'no' or [time1 time2] (default = 'no'),
cfg.baselinetype = 'absolute', 'relative', 'relchange', 'normchange', 'db', 'vssum' or 'zscore' (default = 'absolute'), only relevant for TFR data.
cfg.trials = 'all' or a selection given as a 1xn vector (default = 'all')
cfg.fontsize = font size of title (default = 8)
cfg.hotkeys = enables hotkeys for dynamic zoom and translation (ctrl+) of the axes
cfg.interactive = interactive plot 'yes' or 'no' (default = 'yes')
cfg.figure = 'yes' or 'no', (default = 'yes')
cfg.position = [left bottom width height] (default is automatic)
cfg.renderer = string, 'opengl', 'zbuffer', 'painters'
cfg.linestyle = linestyle/marker type (default = '-')
cfg.linewidth = linewidth in points (default = 0.5)
cfg.linecolor = color(s) used for plotting the dataset(s)
cfg.directionality = '', 'inflow' or 'outflow'
cfg.select = 'intersect' or 'union' (default = 'intersect')
cfg.showlocations = 'no' (default), or 'yes
cfg.layouttopo = filename, or struct
The following options for the scaling of the EEG, EOG, ECG, EMG, MEG and NIRS channels:
cfg.eegscale, cfg.eogscale, cfg.ecgscale, cfg.emgscale, cfg.megscale, cfg.gradscale, cfg.magscale, cfg.nirsscale = number, scaling to apply to the relevant channels
cfg.mychanscale = number, scaling to apply to the channels specified in cfg.mychan
cfg.mychan = Nx1 cell-array with selection of channels
cfg.chanscale = Nx1 vector with scaling factors, one per channel specified in cfg.channel

Description of the plotting and display processes Part 3

ft_singleplotTFR

Use as: ft_singleplotTFR(cfg,data)

Config options:

cfg.parameter = field to be plotted, e.g. 'powspctrm'
cfg.maskparameter = field used for masking of data
cfg.maskstyle ='opacity' (default), 'saturation', or 'outline'
cfg.maskalpha = value between 0 and 1
cfg.masknans = 'yes' or 'no' (default = 'yes')
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.ylim = 'maxmin' or [ymin ymax] (default = 'maxmin')
cfg.zlim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.baseline = 'yes', 'no' or [time1 time2] (default = 'no'),
cfg.baselinetype = 'absolute', 'relative', 'relchange', 'normchange', 'db' or 'zscore' (default = 'absolute')
cfg.trials = 'all' (default) or a selection as a 1xN vector
cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
cfg.title = string, title of plot
cfg.refchannel = reference channel name or 'gui'
cfg.fontsize = font size of title (default = 8)
cfg.hotkeys = enables hotkeys
cfg.colormap = string, or Nx3 matrix
cfg.colorbar = 'yes', 'no' (default = 'yes')
cfg.colorbartext = string of text next to colorbar
cfg.interactive = interactive plot 'yes' (default) or 'no'
cfg.figure = 'yes' (default) or 'no
cfg.position = [left bottom width height] (default auto.)
cfg.renderer = 'opengl', 'zbuffer', 'painters', (default auto.)
cfg.directionality = 'inflow' or 'outflow'

The following options for the scaling of the EEG, EOG, ECG, EMG, MEG and NIRS channels are optional:

cfg.eegscale, cfg.eogscale, cfg.ecgscale, cfg.magscale, cfg.nirsscale, cfg.mychanscale, cfg.mychan, cfg.chanscale

ft_sourceplot

Use as: ft_sourceplot(cfg, anatomical), OR ft_sourceplot(cfg, functional) OR ft_sourceplot(cfg, functional, anatomical)

Config options:

cfg.method = 'ortho', 'slice', 'surface', 'glassbrain', 'vertex', 'cloud'
cfg.anaparameter = string, field in data with the anatomical data (default = 'anatomy' if present in data)
cfg.funparameter = string, field in data with the functional parameter of interest (default = [])
cfg.maskparameter = string, field in the data to be used for opacity masking of fun data (default = [])

The following parameters can be used in all methods:

cfg.downsample = downsampling for resolution reduction, integer value (default = 1) (orig: from surface)
cfg.atlas = string, filename of atlas to use (default = [])
cfg.visible = 'on' (default) or 'off
cfg.figure = 'yes' (default) or 'no'
cfg.position = [left bottom width height] (default auto.)
cfg.renderer = 'opengl', 'zbuffer', 'painters

The following parameters can be used for the functional data:

cfg.funcolormap = colormap for functional data
cfg.funcolorlim = color range of the functional data
cfg.colorbar = 'yes' or 'no' (default = 'yes')
cfg.colorbartext = string for text next to color bar

If your functional data has a time and/or frequency dimension, you can use

cfg.latency = scalar or string, 'all', 'prestim', 'poststim', or [beg end], specify time range in seconds
cfg.avgovertime = 'yes' or 'no' (default)
cfg.frequency = scalar or string, 'all', or [beg end],
cfg.avgoverfreq = 'yes' or 'no' (default)

The following parameters can be used for masking data:

cfg.maskstyle = 'opacity', or 'colormix'
cfg.opacitymap = opacitymap for mask data
cfg.opacitylim = range of mask values, [min max] OR 'maxabs', 'zeromax', 'minzero', 'auto'
cfg.roi = string or cell of strings

The following parameters apply when cfg.method='ortho'

cfg.location = location of cut, 'auto' (default), 'center', 'max', 'center', [x y z], coordinates
cfg.locationcoordinates = coordinate system used in cfg.location, 'head' or 'voxel' (default = 'head')
cfg.crosshair = 'yes' or 'no' (default = 'yes')
cfg.axis = 'on' or 'off' (default = 'on')
cfg.queryrange = number, in atlas voxels (default = 1)
cfg.clim = lower & upper anatomical MRI limits (default = [0 1])

ft_sourceplot continued:

The following parameters apply for cfg.method='slice'

cfg.nslices = number of slices, (default = 20)
cfg.slicerange = range of slices in data, (default = 'auto')
cfg.slicedim = dimension to slice 1,2, or3 (default)
cfg.title = string, title of the plot
cfg.figurename = string, title of the figure window

The following parameters apply to cfg.method='surface'

cfg.surffile = string, file that contains the surface (default = 'surface_white_both.mat')
cfg.surfinflated = string, file that contains the inflated surface (default = [])
cfg.surfdownsample = number (default = 1, i.e. none)
cfg.projmethod = 'nearest', 'project', 'sphere_avg', 'sphere_weighteddistance'
cfg.projvec = vector (in mm)
cfg.projcomb = 'mean', 'max'
cfg.projweight = vector (default = 1)
cfg.projthresh = scalar treshold
cfg.sphereradius = maximum distance in mm
cfg.distmat = precomputed distance matrix (default = [])
cfg.camlight = 'yes' or 'no' (default = 'yes')
cfg.facecolor = [r g b] values or string, e.g. 'brain', 'cortex', 'skin', 'black', 'red', 'r', or an Nx3 or Nx1 array where N is the number of faces
cfg.vertexcolor = [r g b] values or string, as above
cfg.edgecolor = [r g b] values or string, as above

The following parameters apply to cfg.method='cloud'

cfg.cloudtype = 'point', 'cloud' (default), 'surf'
cfg.radius = scalar, (default = 4)
cfg.colorgrad = 'white' or a scalar (e.g. 1
cfg.slice = 'anatomical', '2d', '3d' (default = 'none')
cfg.ori = 'x', 'y' (default), or 'z
cfg.slicepos = 'auto' or Nx1 vector
cfg.nslices = scalar, number of slices
cfg.minspace = scalar, minimum spacing (default = 1)

Description of the plotting and display processes Part 4

ft_topoplotER

Use as: ft_topoplotER(cfg, timelock) or
ft_topoplotER(cfg, freq)

Config options:

cfg.parameter = field to be plotted, e.g. 'powspctrm'
cfg.maskparameter = field used for masking of data
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.zlim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.baseline = 'yes', 'no' or [time1 time2] (default = 'no'),
cfg.baselinetype = 'absolute', 'relative', 'relchange', 'normchange', 'db' or 'zscore' (default = 'absolute')
cfg.trials = 'all' (default) or a selection as a 1xN vector
cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
cfg.refchannel = reference channel name or 'gui'
cfg.hotkeys = enables hotkeys
cfg.colormap = string, or Nx3 matrix
cfg.colorbar = 'yes', 'no' (default = 'yes')
cfg.colorbartext = string of text next to colorbar
cfg.interactive = interactive plot 'yes' (default) or 'no'
cfg.figure = 'yes' (default) or 'no'
cfg.layout = specify the channel layout
cfg.position = [left bottom width height] (default auto.)
cfg.directionality = 'inflow' or 'outflow'
cfg.marker = 'on', 'labels', 'numbers', 'off'
cfg.markersymbol = channel marker symbol (default = 'o')
cfg.markercolor = color spec (default = [0 0 0] (black))
cfg.markersize = channel marker size (default = 2)
cfg.markerfontsize = font size for labels (default = 8)
cfg.highlight = 'off', 'on', 'labels', 'numbers'
cfg.highlightchannel = Nx1 cell-array with selection of channels, or vector containing channel indices
cfg.highlightsymbol = character (default = 'o')
cfg.highlightcolor = color spec (default = [0 0 0] (black))
cfg.highlightsize = highlight marker size (default = 6)
cfg.highlightfontsize = highlight marker size (default = 8)
cfg.interplimits = 'sensors', 'head' (default = 'head')
cfg.interpolation = 'linear', 'cubic', 'nearest', 'v4' (default)
cfg.interpolatenan = 'yes' or 'no'
cfg.style = 'straight', 'contour','both' (default),'fill', 'blank', 'straight_imsat', 'both_imsat'
cfg.gridscale = scaling grid size (default = 67)
cfg.shading = 'flat' or 'interp' (default = 'flat')
cfg.comment = 'no', 'auto' or 'xlim' (default = 'auto')
cfg.commentpos = 'lefttop', 'leftbottom' (default), 'middletop', 'middlebottom', 'righttop', 'rightbottom', 'title', 'layout', OR [x y] coordinates

ft_topoplotIC

Use as: ft_topoplotIC(cfg, comp)

Config options:

cfg.component = field that contains the independent component(s) to be plotted as color
cfg.layout = specification of the layout
cfg.colormap = string, or Nx3 matrix
cfg.zlim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.marker = 'on', 'labels', 'numbers', 'off'
cfg.markersymbol = symbol character (default = 'o')
cfg.markercolor = color spec (default = [0 0 0] (black))
cfg.markersize = channel marker size (default = 2)
cfg.markerfontsize = font size (default = 8)
cfg.highlight = 'on', 'labels', 'numbers', 'off'
cfg.highlightchannel = Nx1 cell-array
cfg.highlightsymbol = symbol character (default = 'o')
cfg.highlightcolor = color spec (default = [0 0 0] (black))
cfg.highlightsize = highlight marker size (default = 6)
cfg.highlightfontsize = font size (default = 8)
cfg.colorbar = 'yes','no' (default),'North', 'South', 'East', 'West', 'NorthOutside','SouthOutside','EastOutside', 'WestOutside'
cfg.colorbartext = string for text next to colorbar
cfg.interplimits = 'sensors', 'head' (default = 'head')
cfg.interpolation = 'linear','cubic','nearest','v4' (default)
cfg.style = 'straight', 'contour','both' (default),'fill', 'blank', 'straight_imsat', 'both_imsat'
cfg.gridscale = scaling grid size (default = 67)
cfg.shading = 'flat' 'interp' (default = 'flat')
cfg.comment ='no' 'auto' or 'xlim' (default = 'auto')
cfg.commentpos = 'lefttop', 'leftbottom' (default), 'middletop', 'middlebottom', 'righttop', 'rightbottom', 'title', 'layout', OR [x y] coordinates
cfg.title = 'auto' or 'off', specify a figure title, or use 'component N' (default) as the title
cfg.figure = 'yes' or 'no', (default = 'yes')
cfg.renderer = string, 'opengl', 'zbuffer', 'painters

ft_topoplotTFR

Use as: ft_topoplotTFR(cfg, freq)

Config options:

cfg.parameter = 'avg', 'powspctrm' or 'cohspctrm' (default is automatic)
cfg.maskparameter = field used for masking of data
cfg.xlim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.ylim = 'maxmin' or [xmin xmax] (default = 'maxmin')
cfg.zlim = 'maxmin', 'maxabs', 'zeromax', 'minzero', or [zmin zmax] (default = 'maxmin')
cfg.channel = Nx1 cell-array with selection of channels (default = 'all')
cfg.refchannel = reference name or 'gui'
cfg.baseline = 'yes','no' or [time1 time2] (default = 'no'),
cfg.baselinetype = 'absolute' (default) or 'relative'
cfg.trials = 'all' or 1xN vector (default = 'all')
cfg.colormap = string, or Nx3 matrix
cfg.marker = 'on', 'labels', 'numbers', 'off'
cfg.markersymbol = symbol character (default = 'o')
cfg.markercolor = color spec (default = [0 0 0] (black))
cfg.markersize = channel marker size (default = 2)
cfg.markerfontsize = font size labels (default = 8)
cfg.highlight = 'off', 'on', 'labels', 'numbers'
cfg.highlightchannel = Nx1 cell-array
cfg.highlightsymbol = highlight char. (default = 'o')
cfg.highlightcolor = color spec. (default = [0 0 0] (black))
cfg.highlightsize = highlight marker size (default = 6)
cfg.highlightfontsize = highlight marker size (default = 8)
cfg.hotkeys = enables hotkeys
cfg.colorbar = 'yes','no' (default),'North', 'South', 'East', 'West', 'NorthOutside','SouthOutside','EastOutside', 'WestOutside'
cfg.colorbartext = string
cfg.interplimits = 'sensors', 'head' (default = 'head')
cfg.interpolation = 'linear','cubic','nearest','v4' (default)
cfg.style = 'straight', 'contour','both' (default),'fill', 'blank', 'straight_imsat', 'both_imsat'
cfg.gridscale = scaling grid size (default = 67)
cfg.shading = 'flat' or 'interp' (default = 'flat')
cfg.comment = 'no', 'auto' or 'xlim' (default = 'auto')
cfg.commentpos = 'lefttop', 'leftbottom' (default), 'middletop', 'middlebottom', 'righttop', 'rightbottom', 'title, 'layout' or [x y] coordinates
cfg.interactive = Interactive plot 'yes' or 'no' (default = 'yes')
cfg.directionality = 'inflow' or 'outflow'
cfg.layout = specify the channel layout
cfg.interpolatenan = string 'yes', 'no' (default = 'yes')

FieldTrip Cheatsheet v2024.01

Brought to you by “The FUNNI Pages”,
The FUNctional NeuroImaging newsletter
www.thefunnipages.com

Description of the data structures part 1

ft_datatype_raw

Required fields:

label: {NCx1 cell} the channel labels represented as a cell-array of strings

time: {1xNS cell} the time axis [1*Ntime double] per trial

trial: {1xNS cell} the numeric data as a cell array, with a matrix of [NC*NS double] per trial

Optional fields:

sampleinfo: [NSx2 double] the begin and endsample of each trial relative to the recording on disk

trialinfo: [NSx1 double] optional trigger or condition codes for each trial

hdr: [1x1 struct] the full header information of the original dataset on disk

grad: [1x1 struct] information about MEG sensor array

cfg: [1x1 struct] the configuration used by the function that generated this data structure

elec: [1x1 struct] information about EEG sensor array

opto: ?

ft_datatype_timelock

Required fields:

dimord: 'chan_time' defines how the numeric data should be interpreted

label: {151x1 cell} the channel labels (e.g. 'MRC13')

time: [1x600 double] the timepoints in seconds

Optional fields:

avg: [151x600 double] the average values of the activity for 151 channels x 600 timepoints

var: [151x600 double] the variance of the activity for 151 channels x 600 timepoints

grad: [1x1 struct] information about the MEG sensor array

elec: [1x1 struct] information about the EEG sensor array

cfg: [1x1 struct] the configuration used by the function that generated this data structure

opto: ?

plus dof, cov, trial, trialinfo, sampleinfo

ft_datatype_freq

Required fields:

dimord: 'chan_freq_time' defines how the numeric data should be interpreted

label: {306x1 cell} the channel labels, or labelcmb

freq: [1x120 double] the frequencies, expressed in Hz

cfg: [1x1 struct] the configuration used by the function that generated this data structure

Optional fields:

time: [1x60 double] the time, expressed in seconds

powspctrm: [306x120x60 double] the power spectrum

plus fourierspctrm, csdspctrm, cohspctrm, grad, elec, cumsumcnt, cumtapcnt, trialinfo

ft_datatype_comp

Required fields:

- unmixing, topo, topolabel

Optional fields:

- all fields from raw, timelock, or freq structures

ft_datatype_source

Required fields:

pos: [NPx3 double] positions at which the source activity could have been estimated

Optional fields:

inside: [NPx1 logical] boolean vector that indicates at which positions the source activity was estimated

dim: [xdim ydim zdim] if the positions can be described as a 3D regular grid, this contains the dimensionality of the 3D volume

cumtapcnt: [120x1 double] information about the number of tapers per original trial

time: [1x1 double] the latency at which the activity is estimated (in seconds)

freq: [1x1 double] the frequency at which the activity is estimated (in Hz)

pow: [6732x120 double] the estimated power at each source position

powdimord: 'pos_rpt' defines how the numeric data has to be interpreted, in this case 6732 dipole positions x 120 repetitions (i.e. trials)

cfg: [1x1 struct] the configuration used by the function that generated this data structure

- plus transform, unit, coordsys, dimord , coh, eta, mom, ori, leadfield, filter, or any other field with dimensions that are consistent with pos or dim

ft_datatype_volume

Required fields:

dim: [nx ny nz] the dimensionality of the 3D volume

transform: [4x4 double] homogenous transformation matrix, specifying the transformation from voxel coordinates to head or world coordinates

Optional fields:

anatomy: [nx x ny x nz double] the numeric data, in this case anatomical information

unit: e.g. 'mm' geometrical units of the coordinate system

coordsys: e.g. 'ctf' description of the coordinate system

- plus fid, prob, stat, grey, white, csf, or any other field with dimensions that are consistent with dim

ft_datatype_headmodel

Required fields:

type: e.g. 'singlesphere'
headmodel =
r: [1x1 double]
o: [1x3 double]

Optional fields:

unit: e.g. 'cm'

ft_datatype_sens

Included fields:

sens.label = Mx1 cell-array with channel labels

sens.chanpos = Mx3 matrix with channel positions (often the same as electrode positions)

sens.elecpos = Nx3 matrix with electrode positions

sens.tra = MxN matrix to combine coils/electrodes into channels

sens.chanori = Mx3 matrix with MEG channel orientations, used for synthetic planar gradient computation

sens.coilpos = Nx3 matrix with MEG coil positions

sens.coilori = Nx3 matrix with coil orientations

sens.tra = MxN matrix to combine coils into channels

sens.balance = structure containing info about the balancing, See FT_APPLY_MONTAGE

Optional fields:

sens.chanposold = Mx3 matrix with original channel positions (in case sens.chanpos has been updated to contain NaNs, e.g. after FT_COMPONENTANALYSIS)

sens.chanoriold = Mx3 matrix with original channel orientations

sens.labelold = Mx1 cell-array with original channel labels

Description of the data structures part 2

ft_datatype_sens

The sens structure for NIRS channels contains

sens.label = Mx1 cell-array with channel labels
sens.chanpos = Mx3 matrix with position of the channels (usually halfway the transmitter and receiver)
sens.optopos = Nx3 matrix with the position of individual optodes
sens.optotype = Nx1 cell-array with information about the type of optode (receiver or transmitter)
sens.optolabel = Nx1 cell-array with optode labels
sens.wavelength = 1xK vector of all wavelengths that were used
sens.tra = MxN matrix that specifies for each of the M channels which of the N optodes transmits at which wavelength (positive integer from 1 to K), or receives (negative ingeger from 1 to K)

Other fields:
- type, unit, fid, chantype, chanunit, coordsys

.