Function Reference

The main functions contain all the main functions for computing the various measures described within the paper

Table of Contents

[Filtering 2](#_Toc97796993)

[Computing Granger causality 2](#_Toc97796994)

[Computing instantaneous frequency 4](#_Toc97796995)

[Computing phase locking value (PLV) 4](#_Toc97796996)

[Computing phase transfer entropy 5](#_Toc97796997)

[Computing power correlations 6](#_Toc97796998)

# Filtering

The FiltHilb function allows the user to narrow band filter the data. This can then be used in conjunction with the Hilbert transform to obtain an estimate of frequency band specific phase and power. A demonstration of its use can be found within the tutorial section under ‘T\_PowCorr’

*Function tf\_res = FiltHilb(data, freqs, freqbloom, srate)*

call as tf\_res = FiltHilb(data, freqs, freqbloom, srate)

INPUTS:

Data should be 2d matrix of single channel EEG data e.g., in samples by trails format.

To compute this for all possible channel pairing simply loop throught channel combinations calling this function each time.

Freqs is a vector of frequencies that you want to filter the data accordidng to e.g., 2:40 Hz. The function will loop over frequencies

Freqbloom is the frequency overlap between consecutive bandwidth e.g., at 7hz with a freqbloom of 1.5 means that filter will include data from 5.5-8.5Hz. Freqbloom will determine the frequency precision of the data

Strate is the sampling frequency of the data.

OUTPUT:

Frequencies by samples by trials matrix of filtered EEG data.

# Computing Granger causality

The functions ‘GC\_td3d’ and ‘GC\_tf3d’ allow the user to compute time domain and time frequency domain granger causality. Demonstrations of how to use both functions can be found within the tutorial section.

*Function tv\_gc = GC\_td3d(data1, data2, winsize, morder, srate)*

Call as tv\_gc = GC\_td3d(data1, data2, winsize, morder, srate)

INPUTS:

Where data1 and data2 2d matrices of single channel EEG data e.g., in samples by trials format. To compute this for all possible channel pairing simply loop through channel combinations calling this function each time.

Winsize is the size of the sliding window used to compute the GC estimate - given in ms.

Morder is the model order used in the autoreggressive model fit.

Strate is the sampling frequency of the data.

OUTPUT:

Matrix of time varying GC inlfuence in format 2 by samples by where tv\_gc(1,:,:) is the influence of channel 1 to 2 and tv\_gc(2,:,:) is the influence of channel 2 to 1.

This function uses code from

[1] L. Barnett and A. K. Seth <http://www.sciencedirect.com/science/article/pii/S0165027013003701 The MVGC

Multivariate Granger Causality Toolbox: A New Approach to Granger-causal

Inference>, \_J. Neurosci. Methods\_ 223, 2014

https://users.sussex.ac.uk/~lionelb/MVGC/html/mvgchelp.html

*Function tf\_granger = GC\_tf3d(data1, data2, freqs, srate, order,twin,times2saveidx)*

Note that this function assumes that appropriate steps have been taken to ensure data is stationary and that an appropriate model order is used for the data

Call as tf\_granger = GC\_tf3d(data1, data2, freqs, srate, order,twin,times2saveidx)

INPUTS:

Data1 and data2 2d matrices of single channel EEG data e.g., in samples by trails format. To compute this for all possible channel pairing simply loop through channel combinations calling this function each time.

Order is the model order used in the autoreggressive model fit.

Strate is the sampling frequency of the data.

Twin is the size of the sliding window used to compute the GC estimate- given in ms.

OUTPUT:

Matrix of time-frequency varying GC influence in format 2 by samples by where tv\_gc(1,:,:) is the influence of channel 1 to 2 and tv\_gc(2,:,:) is the influence of channel 2 to 1

# Computing instantaneous frequency

The function ‘InstFreq’ allows the user to compute the instantaneous frequency or frequency over time of narrow band filtered EEG signals. See tutorial section for an example on implementation.

*Function inst\_freq = InstFreq(data, srate)*

call as inst\_freq = InstFreq(data, srate)

INPUT:

data is vector or a matrix (either 2d or 3d) of hilbert transformed data – but where one dimension is time.

Strate is the sampling frequency of the data.

OUTPUT:

time-varying instantaneous frequency of input

# Computing phase locking value (PLV)

The function ‘PLV’ allows the user to compute time frequency PLV. See tutorial section for guide on implementation.

*Function [plv\_time, plv\_trials] = PLV(data1, data2, freqs, srate, winsize)*

Call as [plv\_time, plv\_trials] = PLV(data1, data2, freqs, srate, winsize)

INPUTS:

Where data1 and data2 2d matrices of single channel EEG data e.g., in samples by trials format. To compute this for all possible channel pairing simply loop through channel combinations calling this function each time.

Freqs is a vector of frequencies that you want to filter the data according to e.g., 2:40 Hz.

Freqbloom is the frequency overlap between consecutive bandwidth e.g., at 7hz with a freqbloom of 1.5 means that filter will include data from 5.5-8.5Hz. Freqbloom will determine the frequency precision of the data

Winsize is the size of the sliding window used to compute the GC estimate - given in ms.

Strate is the sampling frequency of the data.

OUTPUT:

PLV\_time is matrix of frequency x time x trials of plv estimates computed within sliding window within single trials

PLV\_trials is matrix of frequency x time of plv estimates computed across single trials

# Computing phase transfer entropy

The function ‘PhaseTE\_MF’ allows the user to compute time frequency phase transfer entropy. See tutorial section for guide on implementation.

*Function [dPTE, PTE] = PhaseTE\_MF(data, delay, binsize)*

This code was adatped from freely available code original source https://figshare.com/articles/dataset/Phase\_Transfer\_Entropy/3847086

Authors: Matteo Fraschini, Arjan Hillebrand

Phase Transfer Entropy as described in:

M Lobier, F Siebenhuhner, S Palva, JM Palva (2014) Phase transfer entropy: a novel phase-based measure for directed connectivity in networks coupled by oscillatory interactions. Neuroimage 85, 853-872 with implemementation inspired by Java code by C.J. Stam (https://home.kpn.nl/stam7883/brainwave.html)

% Note that implementations differ in normalisation, as well as choices for binning and delay

Call as *[dPTE, PTE] = PhaseTE\_MF(data, delay, binsize)*

INPUTS:

data: time x channel filtered signals of two channels that you want to compute PTE between

delay: prediction delay in samples; leave empty if you want the delay to be based on the frequency content of the data

binsize: binsize for the histograms of phase occurances; provide a number, or 'scott' or 'otnes' to use the approach by Scott or by Otnes and Enochson

OUTPUT:

dPTE: channel x channel matrix of normalised PTE values

PTE: channel x channel matrix of PTE values

# Computing power correlations

The function ‘tfPow\_corr’ allows the user to compute time frequency single trial power correlations. See tutorial section for guide on implementation.

*Function corr\_ts = tfPow\_corr(data1,data2)*

% Call as corr\_ts = tfPow\_corr(a,b)

INPUTS:

data1 and data2 3d matrices of single channel EEG power in the format frequency x samples x trials

OUTPUT:

Corr\_ts is a frequency x samples x trials matrix of correlations between single trial power data of data1 and data2