Source code documentation for:

"The part and the whole: how single nodes contribute to large-scale phase locking in functional EEG networks"

Anaïs Espinoso[1], Marc G. Leguia[1], Christian Rummel[2], Kaspar Schindler[3] and Ralph G. Andrzejak[1]

[1]**Department of Information and Communications Technologies, Universitat Pompeu Fabra, Barcelona, Spain**
[2]**Support Center for Advanced Neuroimaging, University Institute for Diagnostic and Interventional Neuroradiology, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland**
[3]**Sleep-Wake-Epilepsy-Center, Department of Neurology, Inselspital, Bern University Hospital, University of Bern, Bern, Switzerland**

In this document, you will find how to use the source code for the paper Espinoso_2024. Follow the next steps for applying the methodology presented in the paper. The code is done using MATLAB and it uses the same notation for mathematical symbols as in the paper. Additional comments can be found throughout the source code. More details can be found below.
In case you have any questions, please contact the corresponding author anais.espinoso@gmail.com.

Throughout the document underlined text means name of MATLAB file and **bold text** means name of a variable.

1. Main script

The main script AE_Main.m shows how to call the 3 functions explained below to follow the methodological steps presented in the paper. The functions are:

- AE_Delta_test.m: computes the phase-locking contribution measure δ_m and phase-locking contribution test in the 2 frequency ranges: low frequencies $\mathcal{D}_L$ and high frequencies $\mathcal{D}_H$.
- AE_joint_test.m: computes the joint test $\mathcal{D}_J$ using the tests $\mathcal{D}_L$ and $\mathcal{D}_H$ from the previous function.
- AE_lambda.m: computes the relative difference λ between the fraction of rejections from tests $\mathcal{D}_L$, $\mathcal{D}_H$, and $\mathcal{D}_J$ between SOZ and nonSOZ channels.

The script needs **signal** that is a set of L EEG channels with N samples. This variable must enter the first function as $L \times N$. Then, it should be indicated the sampling frequency **Fs**, which in the paper is 512 Hz.

2. Function AE_Delta_test.m

This function computes the phase-locking contribution measure δ_m and test $\mathcal{D}$ for a given frequency band for multichannel EEG recordings. How to call it:

[d,Test_D_A,Test_D_B] = AE_Delta_test(signal,Fs,filt_idx)

where **signal** is a matrix of signals with dimension $L \times N$, where L is the number of channels and N is the number of samples. The variable **Fs** indicates the sampling frequency of **signal**. The input **filt_idx** indicates the frequency range use to band-pass filter **signal**:

i) **filt_idx** = 1: low-frequency range [4, 30] Hz.

ii) **filt_idx** = 2: high-frequency range [80, 150] Hz.

The frequency selection in low and high frequencies was extracted from Ref. [1].

The first output **d** is the phase-locking contribution measure δ_m for a selected **filt_idx** frequency range. The dimension is $L \times W \times (S+1)$, where L is the number of channels, W is the number of windows, and S is the number of surrogates. In the code, the length of the windows is set to 20 s with 75% overlap between subsequent windows. The number of surrogates S is set to 19, thus, the measure is applied to S surrogates and to the original signal, which is why the third dimension is $S + 1$.

To compute the phase-locking contribution **d**, as explained in Section II.B and II.C in the paper, the re-normalized mean resultant length ζ [2] is obtained as a preceding step. To do so, inside AE_Delta_test.m the function mean_resultant_length.m is called. How to call it:

$$[\text{MRL}] = \text{mean\_resultant\_length}(\text{data})$$

where **data** is a 2D matrix of $L \times N$. The output **MRL** is the value of the renormalized mean resultant length.

The phase-locking contribution measure is applied to a set of surrogates generated from the original signal. As explained in Section II.D of the paper, we use iterative amplitude adjusted Fourier transform (IAAFT) surrogates. We here compute multivariate surrogates using the function ASR_SurrogateMulti.m from the source code available in Ref. [3, 4].

The other outputs of AE_Delta_test.m, **Test_D_A** and **Test_D_B**, correspond to the phase-locking contribution test $\mathcal{D}$ ($\mathcal{D}_L$ or $\mathcal{D}_H$ depending **filt_idx**), determined by the direction of the test rejection. The dimension of the variables is $L \times W$. As explained in the paper:

- **Test_D_A**: test $\mathcal{D}$ rejected in the direction of outcome A, i.e., **d**(i,w,1) > max(**d**(i,w,2:(S+1))).

- **Test_D_B**: test $\mathcal{D}$ rejected in the direction of outcome A, i.e., $\mathbf{d}$(i,w,1) < min($\mathbf{d}$(i,w,2:(S+1))).

where $\mathbf{d}$(i,w,1) is the result of the phase-locking contribution measure δ_m for a specific channel i and time windows w for the original signal. Then, $\mathbf{d}$(i,w,2:(S+1)) is the result of the measure δ_m for a channel i and windows w for the set of S surrogates. The variables **Test_D_A** and **Test_D_B** give a binary result:

- 1 is obtained if there is a rejection of the test.
- 0 is obtained if there is a non-rejection of the test.

IMPORTANT REMARK: depending on the size of L and W the computational time of this function can be high. Thus, parallel *parfor* computation is implemented using **numWorkers** automatically determined depending on the maximum number of workers that can support the computer where the function will be executed. To put a fixed value, set **numWorkers**.

3. Function AE_joint_test.m

This function computes the joint test $\mathcal{D}_J$ using the tests $\mathcal{D}_L$ and $\mathcal{D}_H$ obtained using AE_Delta_test.m for **filt_idx** = 1 and **filt_idx** = 2, respectively. How to call it:

[Test_joint_A,Test_joint_B,Test_low,Test_high,Test_joint] =
AE_joint_test(Test_low_A,Test_low_B,Test_high_A,Test_high_B)

where **Test_low_A** and **Test_low_B** is **Test_D_A** and **Test_D_B**, respectively, using **filt_idx** = 1 in AE_Delta_test.m. Similarly with **Test_high_A** and **Test_high_B** using **filt_idx** = 2. All inputs have dimension $L \times W$. For these variables, we have 1 if there is a rejection of the test or 0 if not.

The outputs **Test_joint_A** and **Test_joint_B** represent the joint test $\mathcal{D}_J$. Similar to the inputs, we have 1 if the test is rejected or 0 if not. The dimension of these outputs is $L \times W$. The outputs **Test_low**, **Test_high**, and **Test_joint** represent the tests $\mathcal{D}_L$, $\mathcal{D}_H$, and $\mathcal{D}_J$, respectively. Unlike the other outputs, these include the rejections in the direction of outcomes A, B, and C in the same variable. The dimension of these outputs is also $L \times W$. Thus, the variables **Test_low**, **Test_high**, and **Test_high** have the following values:

- 1 is obtained if there is a rejection of the test in the direction of outcome A.
- -1 is obtained if there is a rejection of the test in the direction of outcome B.
- 0 is obtained if there is a non-rejection of the test, represented as outcome C in the paper.

4. Function AE_lambda.m

This function computes the relative difference λ from the test rejections computed by the last 2 functions between SOZ and nonSOZ channels. How to call it:

[lambda,p_SOZ,p_nonSOZ] = AE_lambda(Test,SOZ_chan,nonSOZ_chan,varargin)

where **Test** is the $L \times W$ matrix with rejections (1) and non-rejections (0) of the selected test. This variable can be **Test_D_A** and **Test_D_B** from AE_Delta_test.m (with **filt_idx** 1 or 2), and **Test_joint_A** and **Test_joint_B** from AE_joint_test.m. The variables **SOZ_chan** and **nonSOZ_chan** are a subset of the L channels. This function computes the contrast of the rejections between these two subsets of channels. This contrast can be obtained in specific periods or all time windows W. This is determined by the input **varargin**:

- Calling the function without the input **varargin**:
  [lambda,p_SOZ,p_nonSOZ] = AE_lambda(Test,SOZ_chan,nonSOZ_chan)
  This will obtain all outputs for all time windows W.
- Calling the function as:
  [lambda,p_SOZ,p_nonSOZ] = AE_lambda(Test,SOZ_chan,nonSOZ_chan,idx_1,idx_2)
  where **idx_1** indicates the windows among all W windows to start and **idx_2** is the windows to finish the analysis. Thus, the time windows between **idx_1**

and **idx_2** represent a period. In the paper we computed 3 periods: before, during, and after seizure.

The output **lambda** is the relative difference between the **SOZ_chan** and **nonSOZ_chan** rows from **Test**. The outputs **p_SOZ** and **p_nonSOZ** are the fraction of rejections for the **SOZ_chan** and **nonSOZ_chan** rows, respectively, from **Test**.

In the script AE_Main.m we show how to call this function to obtain the different periods presented in the paper to obtain a table similar to Table II in the paper.

References

[1] M. Bandarabadi, H. Gast, C. Rummel, C. Bassetti, A. Adamantidis, K. Schindler, and F. Zubler, "Assessing epileptogenicity using phase-locked high frequency oscillations: a systematic comparison of methods," *Frontiers in neurology*, vol. 10, p. 1132, 2019.

[2] R. G. Andrzejak, A. Espinoso, E. García-Portugués, A. Pewsey, J. Epifanio, M. G. Leguia, and K. Schindler, "High expectations on phase locking: Better quantifying the concentration of circular data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 33, no. 9, 2023.

[3] R. G. Andrzejak, K. Schindler, and C. Rummel, "Bern-Barcelona database," 2012. Available: `https://www.upf.edu/web/ntsa/downloads`.

[4] R. G. Andrzejak, K. Schindler, and C. Rummel, "Bern-Barcelona database," 2012. Available: `https://repositori.upf.edu/handle/10230/42829`.