**Code associated with the paper 'The Significance of Neural Inter-Frequency Power Correlations', doi:** [**10.21203/rs.3.rs-329644/v1**](https://doi.org/10.21203/rs.3.rs-329644/v1)

**Author:** Oscar Wiljam Savolainen – NGNI Lab, Imperial College London

**Date:** 09/08/2021

This repository contains code for testing the statistical significance of correlations between the power of different frequency bands in non-stationary time series. These correlations are referred to as Inter-Frequency Power Correlations (IFPC), i.e. Inter-Scale Power Correlations (ISPC), throughout the repository. This test accounts for intra-frequency autocorrelation, inter-frequency non-dyadicity (over-sampling of frequency domain) and multiple testing under dependency. For more details on the statistical test, see the paper ('The Significance of Neural Inter-Frequency Power Correlations', doi: [10.21203/rs.3.rs-329644/v1](https://doi.org/10.21203/rs.3.rs-329644/v1)).

This repository is particularly tailored for analysing the broadband raw neural dataset from 'Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology' from Sabes lab [doi: 10.5281/zenodo.3854034], but the statistical test applies to any time series.

For the analysis of an arbitrary signal, a folder “Analyzing\_an\_arbitrary\_signal” is also provided that contains scripts that can run independently on a desktop, ideally for signals smaller than 1e6 in length.

**Respository contents:**

**A) Formatting\_the\_NWB\_data\_into\_MATLAB:** a small sub-directory for extracting the neural Sabes lab .nwb data from Sabes lab, and storing it as .mat files.

**B) Processing\_data:** a sub-directory for processing the .mat neural data and calculating their Inter-Frequency Correlation Matrices (IFCM).

**C) MC\_Sim\_for\_Producing\_Null\_Distributions:** a sub-directory for performing a Monte Carlo simulation that produces 1) IFCM from random white noise input; 2) IFCM from phase-randomised versions of the Wavelet Power Spectrum (WPS), derived from neural data. Together, these are used as the null distributions in the statistical significance test for IFCs.

**D) Testing\_the\_Significance\_and\_Visualisation:** a sub-directory for testing the statistical significance of the true inter-frequency correlations, and for presenting the results.

For analysing the Sabes lab data, these need to be run in order, although (B) and (C) can be run independently if the post-standardised neural data lengths from each session are known (see paper for details).

**Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** is a MATLAB file containing parameters and directory names used throughout this work. Relevant directory paths should be filled in.

**E) Analyzing\_an\_arbitrary\_signal:** a separate directory (not a sub-directory), for testing the statistical significance of the inter-frequency power correlations in a given arbitrary signal. The script “Script\_for\_analyzing\_arbitrary\_signal.m” can be edited and/or run to test the inter-frequency power correlations of arbitrary non-stationary signals. As it runs on desktop, it may not perform well for very long signals, e.g. above 1e6 samples, due to memory constraints.

**Recreating the results from the paper from the original Sabes lab '.nwb' files:**

This section describes how to recreate the results from the paper from the original .nwb files. However, the parts concerning **MC\_Sim\_for\_Producing\_Null\_Distributions** and **Testing\_the\_Significance\_and\_Visualisation** can be read to generalise to other datasets.

When running this code, the first step should be to add all of the code to the MATLAB path, and verify that one has Python installed (also requires module h5py). The second step should be to add the paths to the desired directories for data, results and figures to the **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** file.

Then, the **Formatting\_the\_NWB\_data\_into\_MATLAB** Python code can be used to format the original .nwb files from Sabes lab as .mat files. The .mat files should be saved to the same **raw\_neural\_data\_folder** as specified in **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m**.

Then, the **main\_recreate\_results\_from\_mat\_Sabes\_data.m** script can be run. This will process the data (perform the Spectral Interpolation and Current Source Density referencing). **However, it will throw an error (in line 98)**. This is because the null distributions need to be computed.

The null distributions (White Noise ISPCMs and ISPCMs derived from phase-randomised neural WPS (either FT or IAAFT)) need to be calculated. The code is designed for use in a High Performance Computing (HPC) cluster that operates on Linux. The master BASH scripts, located in the **MC\_Sim\_for\_Producing\_Null\_Distributions** directory, are:

* **main\_WN\_ISCM\_MC\_simulation.pbs**, which creates the white noise generated IFCM distributions . This is set to 1000 runs, for white noise processes of length 500. The length of the white noise process can be edited by changing the value of the “signal\_length” parameter (line 14). “signal\_length” is given in samples, so a white noise process of 350 s long, sampled at 24414… kHz is equal to 8544922 samples. Not all 1000 jobs will succeed, and so this script can be run multiple times, but a few hundred is probably a sufficient sample size for reliably determining the mean WN IFCM.

For each job, it produces a string of PCG random numbers, and stores them in a .txt file. It then accesses the .txt file from MATLAB function **generate\_WN\_ISCMs.m**, which calculates the IFCM associated with the random numbers. It then stores the results.

The PCG functions, which are C functions, need to be compiled. **Makefile**, in sub-directory **PCG\_random\_number\_generation**, needs to be run in BASH to compile the functions.

* **main\_FT\_ISCM\_MC\_simulation.pbs**, which creates the FT phase-randomised distributions (set to Channels 1-96, Sessions 1-15).
* **main\_IAAFT\_ISCM\_MC\_simulation.pbs**, which creates the IAAFT phase-randomised distributions (set to Channels 1-20, Sessions 1-15).

The phase-randomisation .pbs scripts may have to be run 1-4 times to get a sufficient number of elements for the distribution, e.g. 200. If one chooses to use FT nulls, then one should set the parameter “use\_IAAFT\_or\_FT\_phase\_ran\_nulls” in **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** as ‘FT’. Else if one uses IAAFT nulls, it should be set to ‘IAAFT’. If one is only interested in FT distributions, the IAAFT distributions do not need to be computed, and vice-versa.

The nulls then need to be saved to their respective directories, as specified in **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m**.

Once this has been done, the error statement in **main\_recreate\_results\_from\_mat\_Sabes\_data.m** (line 98)can be commented out or deleted.

Then, the **main\_recreate\_results\_from\_mat\_Sabes\_data.m** script can be run without errors, which will calculate and plot the significant elements of the neural IFCMs.

**Recreating the results from the paper from the publicly available results:**

This section describes how to recreate the results from the paper using the publicly available results.

**To analyse the data with FT nulls:**

Create a folder for FT phase-randomised null zip files, e.g. 'XAX'. Place the zip files in this folder.

Enter the folder path in the 'save\_FT\_phase\_ran\_ISCMs' variable (as a string) in the **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** script.

Extract the FT phase-randomise zip files to their own respective sub-folders (in folder XAX), e.g. select "extract Channels\_1\_to\_5\_FT\_distribution.zip to '\Channels\_1\_to\_5\_FT\_distributions'"

Set the 'use\_IAAFT\_or\_FT\_phase\_ran\_nulls' (in the **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** script) variable as 'FT'.

Run the **main\_analyse\_publicly\_available\_results.m** script, which will perform the statistical testing for all neural ISPCMs.

Running script **main\_analyse\_publicly\_available\_results.m**, in particular running function 'merge\_FT\_phase\_ran\_folders', will automatically merge the separate channels files into the same folder (XAX).

The FT phase-ran files were kept in separate folders because Zenodo was having problems with the uploading of large individual zip files.

**To analyse the data with IAAFT nulls:**

Create a folder for IAAFT phase-randomised null zip file, e.g. 'XB2'.

Place the 'Channels\_1\_to\_20\_IAAFT\_distributions' zip file in this folder.

Enter the folder path in the 'save\_IAAFT\_phase\_ran\_ISCMs' variable (as a string) in the **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m** script.

Extract the IAAFT phase-randomise zip files. If extracted to its own sub-folder, adjust the path in 'save\_IAAFT\_phase\_ran\_ISCMs' so that the sub-folder with Sessions folders is the folder whose path is given in 'save\_IAAFT\_phase\_ran\_ISCMs', e.g. so that 'XB2\Session\_1' is a valid folder.

Set the 'use\_IAAFT\_or\_FT\_phase\_ran\_nulls' (in **Inter\_Scale\_Stat\_Sig\_Sabes\_Parameters.m**) variable as 'IAAFT'.

Run the **main\_analyse\_publicly\_available\_results.m** script, which will perform the statistical testing for all neural ISPCMs.

Note: if one wishes to calculate the IFCM independently for a signal, the function **calculate\_inter\_scale\_correlation\_matrix.m** (located in Processing\_data \Calc\_ISPCM\_from\_SI\_CSD) will do so.

**Analysing an arbitrary signal:**

This section describes how to determine the statistically significant elements of the inter-frequency power correlation matrix of an arbitrary signal. In the parallel directory, “Analyzing\_an\_arbitrary\_signal” (not a sub-directory), the script “Script\_for\_analyzing\_arbitrary\_signal.m” can be edited and/or run to test the inter-frequency power correlations of an arbitrary non-stationary signal, although of lengths above 1e6 samples can run slowly. For a dataset of many signals of length above 1e6 samples, it is recommended to use a High Performance Cluster to do the computation, perhaps of the multiple signals in parallel.