

# Multiscale Oscillatory Dynamics Analysis

v1.01

## User Manual

Julian Newman, Gemma Lancaster, Aneta Stefanovska  
Department of Physics, Lancaster University

September 2018

# Contents

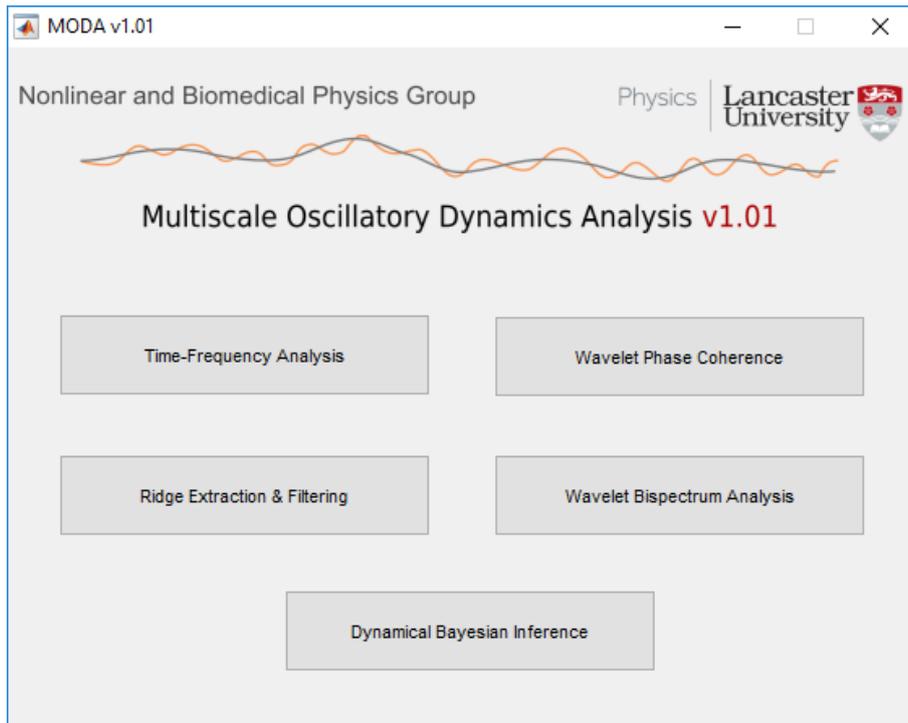
<b>Acknowledgements</b>	<b>i</b>
<b>Introduction to MODA</b>	<b>1</b>
<b>Downloading and running MODA</b>	<b>3</b>
<b>Basic format of MODA</b>	<b>4</b>
Loading time-series into MODA . . . . .	4
Analysis results . . . . .	8
Truncating the signals . . . . .	11
Plotting and saving . . . . .	12
Large arrays and downsampling . . . . .	14
<b>Overview of analysis methods</b>	<b>15</b>
Time-Frequency Analysis . . . . .	15
Why not just use the Fourier transform? . . . . .	21
Wavelet Phase Coherence . . . . .	24
Ridge Extraction & Filtering . . . . .	26
Wavelet Bispectrum Analysis . . . . .	28
Dynamical Bayesian Inference . . . . .	32
Surrogates . . . . .	34
<b>Time-Frequency Analysis</b>	<b>40</b>
<b>Wavelet Phase Coherence</b>	<b>57</b>
<b>Ridge Extraction &amp; Filtering</b>	<b>63</b>
<b>Wavelet Bispectrum Analysis</b>	<b>73</b>
<b>Dynamical Bayesian Inference</b>	<b>81</b>

*The vast majority of examples shown in this manual work with either the file 6signals\_10Hz.mat, 2signals\_10Hz.mat or 1signal\_10Hz.mat, all of which have been made available to the user.*

*These are artificially generated signals intended to help illustrate the analysis methods in MODA. The file 1signal\_10Hz.mat contains the first of the six signals stored in 6signals\_10Hz.mat, and the file 2signals\_10Hz.mat contains the first and fourth of the six signals stored in 6signals\_10Hz.mat. For all three files, the signals are stored row-wise and the results illustrated in this manual work are for a sampling frequency of 10 Hz.*

## Acknowledgements

# Introduction to MODA



Virtually all oscillatory processes are influenced by external factors which vary over time, and thus do not proceed in a strict clock-like manner. Indeed, such time-variability is a necessary component of the stable functioning of many oscillatory processes, especially biological oscillators.

**MODA** (Multiscale Oscillatory Dynamics Analysis) is a numerical toolbox developed by the Nonlinear & Biomedical Physics group at Lancaster University for analysing oscillatory behaviour in real-life time-series that are assumed to be the output of some *a priori* unknown nonautonomous dynamical system, and deriving important properties about this dynamical system from the time-series. It includes methods both for analysing the recordings of a single signal over time, and for analysing a set of recordings of multiple different signals over time. In particular, it has tools for analysing *bivariate* time-series consisting of the simultaneous recordings of two different signals over time, with a view to examining possible connections between the two signals.

The foundation of most of the methods in MODA is *time-frequency analysis*, which describes the time-evolving frequency content of a signal. (The importance of *time-frequency analysis*—as opposed to time-independent frequency analysis—for real-world oscillatory processes is discussed

in this manual, in the section “Why not just use the Fourier transform?”.) On the basis of this, one can examine whether there is either mutual interaction between or common influence on a pair of signals, by way of a “common” oscillatory component with possibly time-varying frequency, where “commonness” is measured by coherence of phases. Similarly, one can examine whether there are interactions between oscillatory components either within one signal or between two different signals, using bispectral analysis; such analysis can detect both nonlinear interactions between oscillatory components and also linear interactions between nonlinear oscillatory components. Moreover, one can extract from a signal oscillatory components of time-variable frequency, together with the time-evolution of the phase of such a component. From here, one can use dynamical Bayesian inference to reconstruct approximations of a nonautonomous stochastic differential equation describing the joint evolution of a pair of oscillatory components (of either the same signal or two different signals): under the approximation that phases of interacting oscillators evolve according to the dynamics of coupled phase oscillators, one obtains over a series of time-windows the “most likely” coupling function from within the span of a given number of orthonormal basis functions on the 2-torus.

### Remark on continuous vs. discrete signals

Generally speaking, a “signal” can come under one of two categories:

- (i) the output of a **continuous-time** process – e.g. as an output of the cardiorespiratory system, one can consider the simultaneous blood velocity through a certain blood vessel and chest circumference around the lungs;
- (ii) the output of a process that **proceeds in discrete steps** – e.g. the sequence of partial sums calculated during the process of adding up the total of a list of numbers.

Processes of type (ii) are called *discrete-time processes*, because the enumeration of the steps is conceptually analogous to the tracking of time for a continuous-time process.

Now MODA is designed for those who wish to understand the *dynamical behaviour of a continuous-time process*. Accordingly, MODA has algorithms intended to calculate various quantitative features of the output  $x(t)$  of a continuous-time process, such as the wavelet transform of the function  $x(\cdot)$  (or the individual coordinate functions of  $x(\cdot)$  if  $x(\cdot)$  is bivariate).

However, when seeking to analyse  $x(t)$ , MODA only has limited information available: namely, the time-series that MODA is given is simply a sequence of recorded values  $x(\frac{n}{f_s})$ ,  $n = 0, \dots, k-1$ , where  $f_s$  is the so-called “sampling frequency”. This is, of course, a *discrete-time signal*. In order to estimate various quantities that are defined for continuous-time signals, MODA applies algorithms whose input is a discrete-time signal. Of course, the higher the rate  $f_s$  at which recordings are made, the more fully and accurately MODA can represent the true behaviour of the continuous-time output  $x(t)$ . In particular, the presence of high-frequency oscillatory components within  $x(t)$  can only be detected if  $f_s$  is sufficiently high. (Indeed, the *Fourier transform* of the discrete-time signal is only defined for frequencies up to the *Nyquist frequency*, namely  $\frac{1}{2}f_s$ .)

# Downloading and running MODA

# Basic format of MODA

Throughout this manual, we will essentially use the terms **time-series** and **signal** interchangeably, except that we will sometimes refer to a pair of simultaneous signals as a **bivariate time-series**.

## Loading time-series into MODA

In MODA, a **time-series is defined by a series of recorded values, together with the frequency at which the recordings are made (“sampling frequency”)**; it is assumed here that the time-gap between consecutive recordings stays fixed. MODA treats all the recorded values in a signal as dimensionless numbers (i.e. numbers without a unit).

**MODA is able to analyse multiple signals at once**, on the condition that the signals all have the **same duration and sampling frequency**.

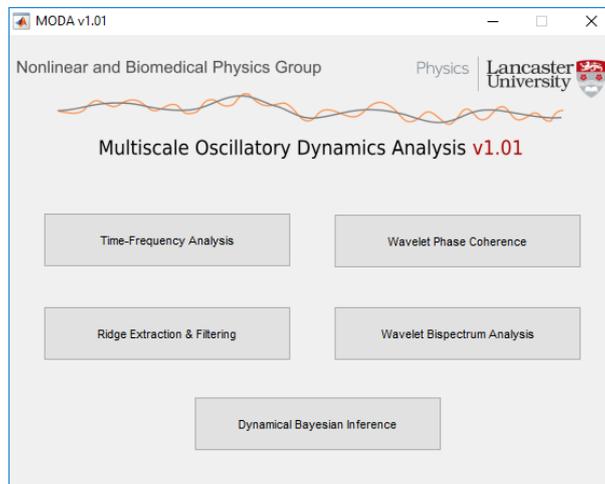
Before you can use MODA to analyse your time-series, the time-series must first be stored in a suitable file type and in the right format. This file only contains the recorded data and not the frequency at which the recordings are made. (The frequency is specified by the user when loading the file.)

- The file type must either be a .mat file, or some other file type supported by MATLAB (such as a CSV file that can be created in Microsoft Excel).
- The file must be **a single array, each of whose entries is a single real number**. If you wish to analyse a single (univariate) time-series consisting of  $k$  recordings, then this array may be either a  $1 \times k$  array (a row) or a  $k \times 1$  array (a column), and it simply consists of the  $k$  recorded values listed in order. If you wish to analyse  $N$  time-series, each consisting of  $k$  recordings, then this array may be either an  $N \times k$  array (where each row contains the recordings of one time-series), or it may be a  $k \times N$  array (where each column contains the recordings of one time-series).
- To be able to apply bivariate analysis methods, the number of time-series  $N$  stored in the file must be an even number  $N = 2\tilde{N}$ . For each  $i = 1, \dots, \tilde{N}$ , **the time-series in the  $i$ -th row/column and the time-series in the  $(\tilde{N} + i)$ -th row/column are taken as a pair** on which the bivariate analysis methods may be applied.
- For wavelet bispectral analysis, the file must contain at most two time-series. This is because the bispectrum computations for one pair of signals already take a remarkably long time.

**If the array loaded into MODA is extremely large, then MODA may run very slowly or even crash.** A potential way to overcome this is by first downsampling the signals,

as discussed further below.

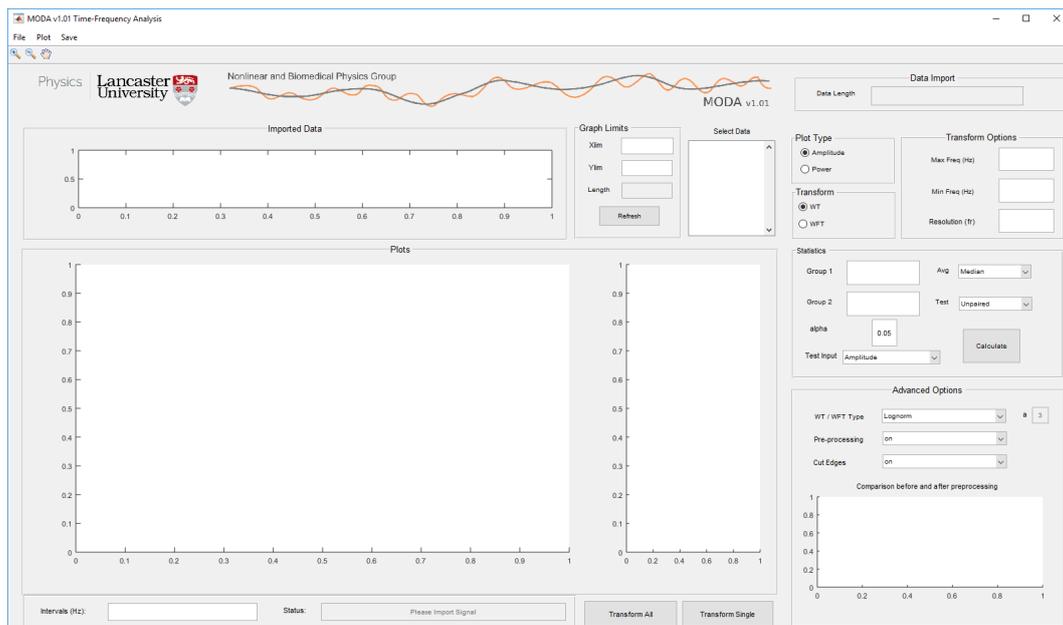
Now when you run MODA, the opening window looks as follows:



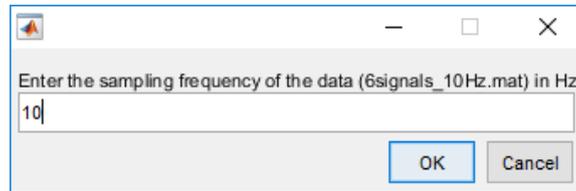
This window contains five options, each opening an application in which it is possible to carry out the labelled category of time-series analysis:

- Time-Frequency Analysis
- Wavelet Phase Coherence
- Ridge Extraction & Filtering
- Wavelet Bispectrum Analysis
- Dynamical Bayesian Inference

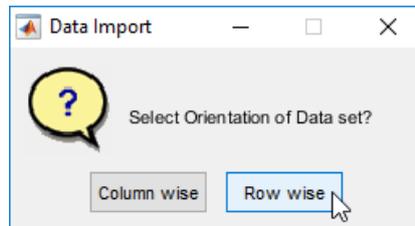
From these five options, click on the type of analysis that you want to carry out. When you do this, a new window opens, whose window title is MODA v1.01 followed by the name of the option that you clicked on.



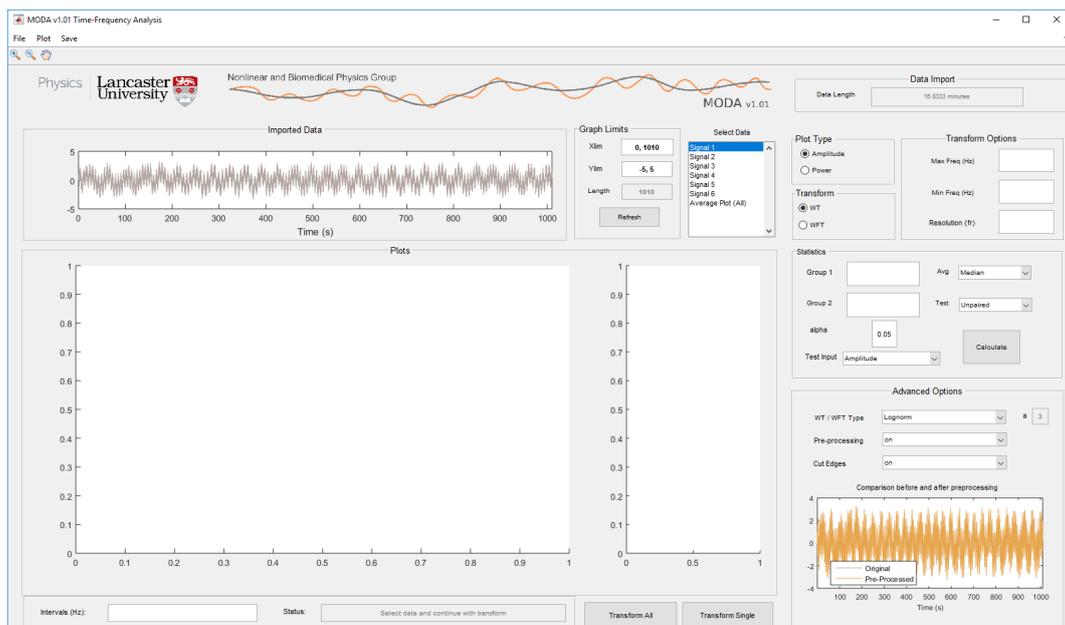
At the top left corner are the menu options “File”, “Plot”, “Save”; to load your time-series, click **File**, and then **Load time series**. This will open a file browser where you can find and select the file in which the time-series is/are stored. Once you have selected the file, you are first asked what the sampling frequency (in Hz) of your time-series is. The **sampling frequency** is defined as the reciprocal of the time-gap between consecutive recordings in the time-series. So for example, if your time-series involved one new value being recorded every 5 milliseconds, then this corresponds to 200 recordings per second, and so you would enter 200.



Once you have entered the sampling frequency, you are then asked for the **orientation** of your data set, referring to whether the time-series in your file are stored in rows or columns: “column-wise” means that each column contains one time-series, and “row-wise” means that each row contains one time-series.



Once you have clicked on the orientation of the file, the signals in the file are loaded into the already open window “MODA v1.0 . . . . .”. From here, you can perform all the analysis methods available within the category that you originally selected, and all results will also be shown in the same window.



In this window, when the file is loaded, there will be a small box listing the signals or signal pairs contained in the loaded file (numbered “Signal (Pair) 1”, “Signal (Pair) 2”, etc.). A graph of whichever signal or signal pair is currently highlighted in this box can be seen in the top-left box of the window. (In the Wavelet Bispectrum application, the box listing the signals does not exist, as one can only upload one signal or signal pair; nonetheless, the graph is still shown in the top-left box.) One can save this graph of the signal or signal pair by clicking “Plot” in the top-left corner of the window, and then clicking “Time series”—this will plot the graph in a new window as a savable figure.

Dynamical Bayesian Inference also has the (recommended) option of loading time-series of phases generated from recorded data by the Ridge Extraction & Filtering application. This will be described in the section on Dynamical Bayesian Inference.

**Example:** MODA is designed for real-world signals, but it is also often useful and important to see how artificially generated signals are treated by time-series analysis methods that one wishes to apply to real signals. So suppose we wish to generate and analyse a 100-second bivariate time-series

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} \sin(\omega_1 t) \\ \cos(\omega_2 t) \end{pmatrix}$$

sampled at rate 20 Hz (so that there are 2000 samples altogether) starting from time  $t = 0$  s, where  $\omega_1 = 6.3$  Hz and  $\omega_2 = 3$  Hz. To do this:

(a) In MATLAB, write and run the code:

```
t = linspace(0,99.95,2000);
    % generates a row vector t = (0 0.05 0.1 ... 99.9 99.95)
x = sin(6.3.*t);
    % generates the series of x-values as a row vector
y = cos(3.*t);
    % generates the series of y-values as a row vector
z = [x;y];
    % vertically concatenates x and y to give a 2 x 2000 array
save example.mat z
    % saves the array z as a MATLAB file “example.mat” in the Current Folder
```

(b) Run MODA, and select any of the five option.

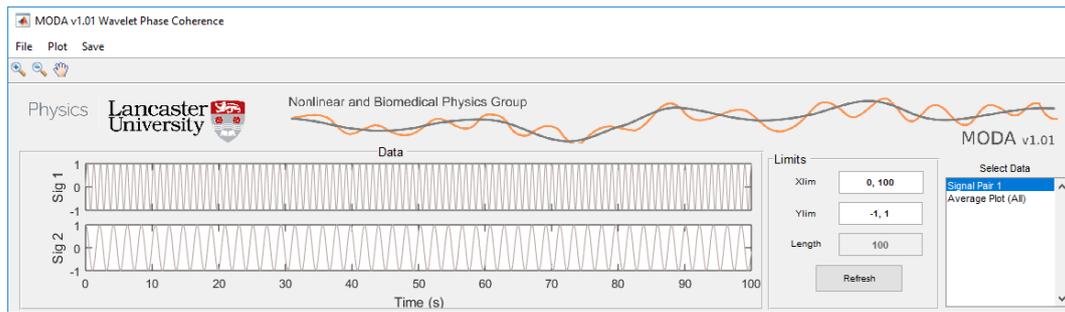
(c) Click on “File” and then “Load time series”. Browse and select the file “example.mat” created by the above MATLAB code.

(d) Enter 20 for the sampling frequency in Hz.

(e) Since the two signals are stored as rows, click on “Row wise”.

In Time-Frequency Analysis and Ridge Extraction & Filtering, the two signals  $x(t)$  and  $y(t)$  are treated as two distinct signals, with Signal 1 being  $x(t)$  as that was the signal stored in the first row, and Signal 2 being  $y(t)$  as that was the signal stored in the second row.

In the other three applications,  $x(t)$  and  $y(t)$  are treated as a single Signal Pair, with  $x(t)$  displayed in the “Sig 1” plot, and  $y(t)$  displayed in the “Sig 2” plot:



## Analysis results

To analyse a loaded signal, the user fills in all parameters as required (*including* parameters for surrogate testing where applicable), and then clicks on the button to perform the calculations; this button is clearly labelled as appropriate (e.g. “Calculate”, “Transform”, “Extract ridge(s)”, etc.). The user can subsequently choose to re-do computations under a different set of parameters, by filling in the new parameters and then clicking the suitable button again, **but re-doing computations in this way will remove all the results of previous computations performed within the session.** To keep the results from a set of parameters, one can click “File” and then “**New workspace**” to open a new window for the same application, in which one can load the same file and then perform computations with a new set of parameters. Alternatively, one can plot results in a separate window (using the “Plot” facility described below under “Plotting and saving”) before redoing computations under a new set of parameters.

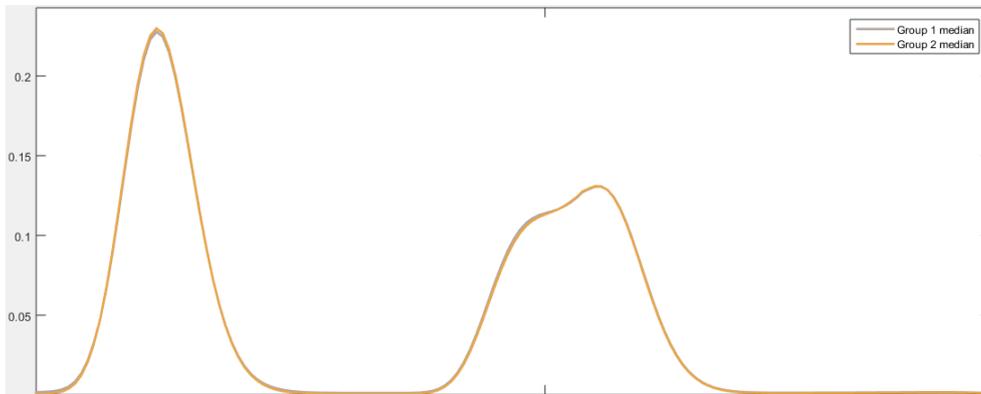
All results are presented in graphical form. For resulting showing a nonnegative-valued function of two variables, colour-coding is used. The colour-coding scale is as follows (with grey representing the smallest values and purple the largest values):



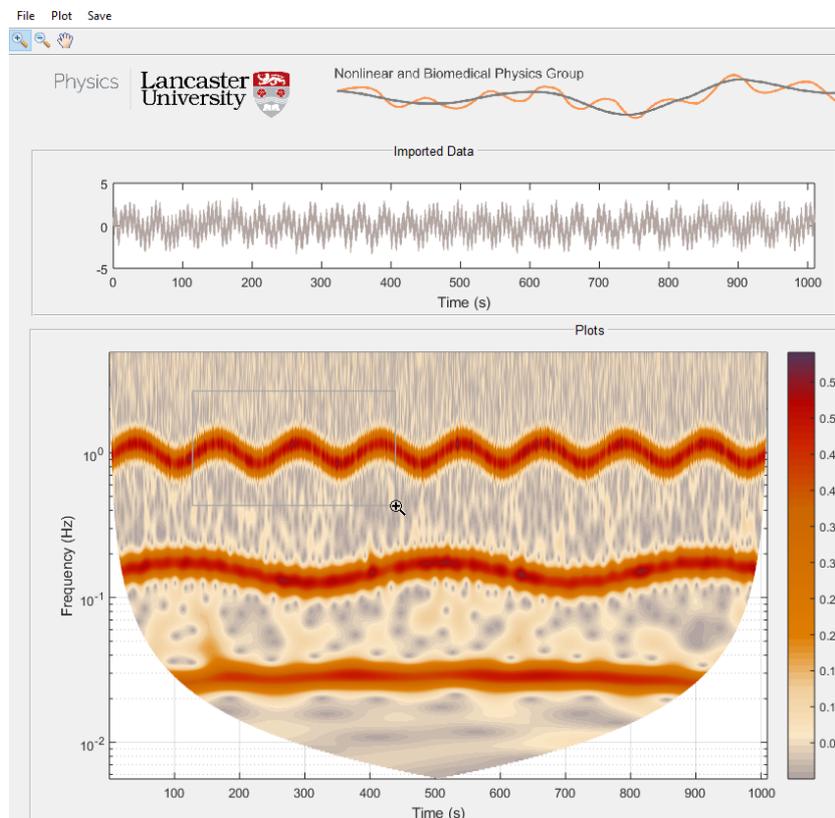
There are some results for which the user has an option as to exactly what aspects of the results are displayed: in particular, time-frequency representations can either display Amplitude or Power (with “power” simply being the square of amplitude), as selected inside a box titled “Plot” or “Plot Type”.

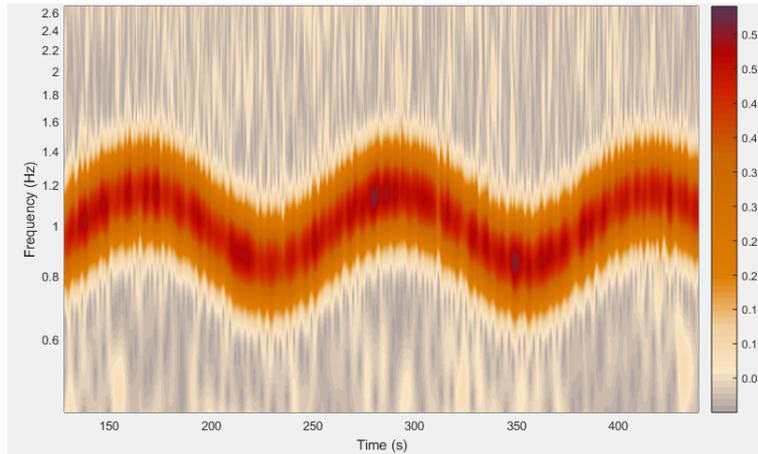
As is generally the case in MATLAB figures, when multiple curves are plotted within the same set of axis, where two or more of them overlap only one of them will be visible. Generally,

in MODA, multiple curves in the same plot will be defined over exactly the same axis for the independent variable (generally time or frequency), and so where only one curve is seen, all the curves will be overlapping.

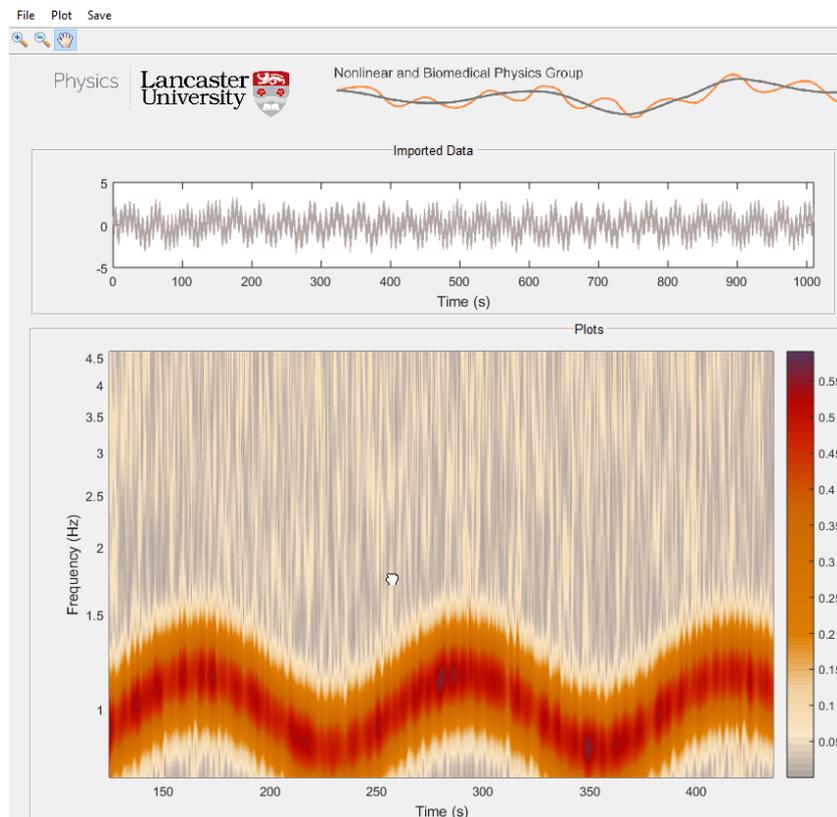


It is possible to **zoom in** on plots. To do this, click the zoom-in magnifying glass at the top-left of the window (underneath “File”), so that the cursor now appears as a magnifying glass, and then clicking on the part of the graph where you wish to zoom in. Alternatively, while the cursor appears as a magnifying glass, you can control the precise rectangle on which you wish to zoom in: press the mouse button while the cursor is at one corner of the rectangle where you wish to zoom in, and while holding down the mouse button, drag the cursor to the opposite corner of where you wish to zoom in. When you release the mouse button, the selected rectangle will fill the plot (with the horizontal and vertical axes separately rescaled to make this work):





When you have zoomed in, you can still zoom in further. (But do not do consecutive zooms by clicking in quick succession, as double-clicking returns to the original.) **When zoomed in, to see other parts of the graph at the same zoomed-in scale, click on the hand that is to the right of the two magnifying glasses, and then click on the graph and, while holding down the mouse button, drag the mouse to move the part of the graph that is visible.**



**To zoom back out to the original graph, just double-click on the graph** (while the cursor is displayed as either a magnifying glass or a hand). Alternatively, to undo one zoom from a succession of inward zooms, click the zoom-out magnifying glass, and then click on the graph. It is not possible to zoom out beyond the original full graph.

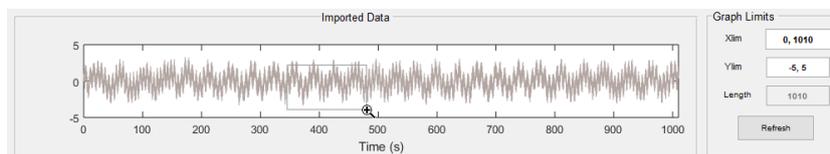
After you have clicked on either one of the magnifying glasses or the hand, **to return the**

cursor to its normal function just click again on the same icon.

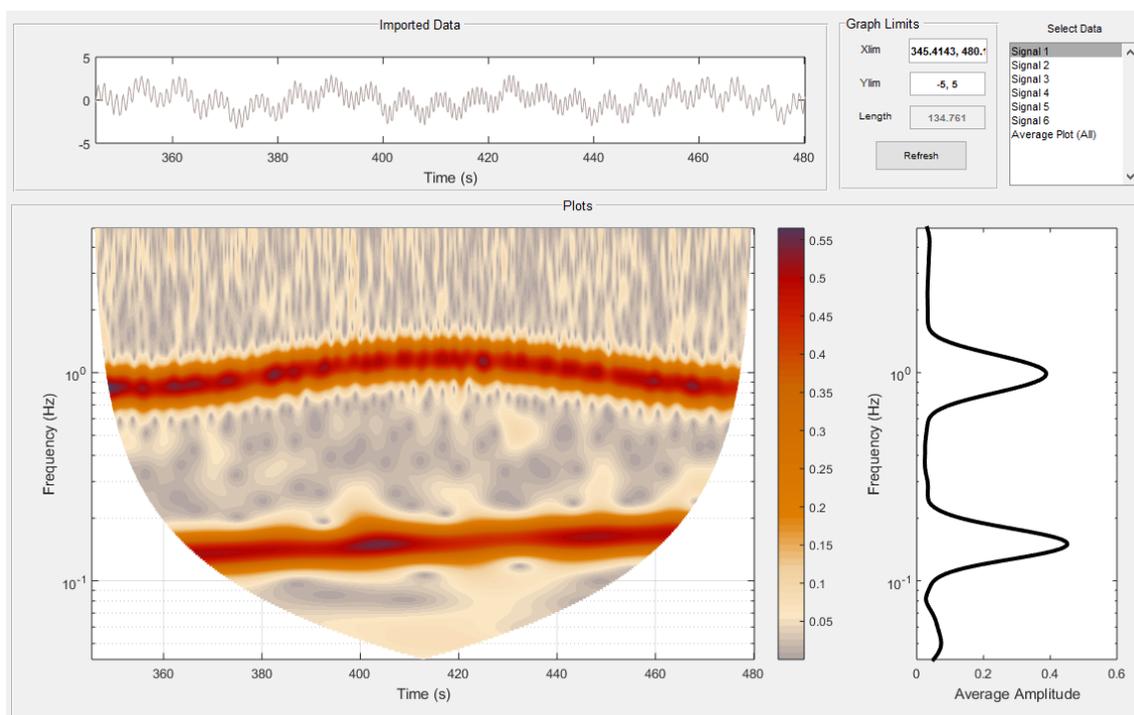
In Ridge Extraction & Filtering, Wavelet Bispectrum Analysis and Dynamical Bayesian Inference, not all of the calculated results can be displayed in the MODA window at the same time. Instead, there is a small pull-down list, in the bottom left corner of the part of the window where the results are shown, where one can choose which results to display.

## Truncating the signals

One may wish to **analyse only a portion of the duration of the recorded signal(s)**. To do this, once you have loaded the signal(s), click on the zoom-in magnifying glass, and in the plot where the selected signal is shown (or the plot of either signal from the selected signal pair), **zoom in on a rectangle whose horizontal range is the portion of the signal that you wish to restrict to** (by dragging the cursor within the plot, as described above); then **click on the “Refresh” button** in the box titled “Limits” or “Graph Limits”. The vertical range of the selected rectangle makes no difference to the results, and will automatically zoom out to the original vertical range when any of the signals in the signal list is clicked on.



Alternatively, one can manually specify the desired time-range in the “Xlim” field in the “Graph Limits” box, by entering the number of seconds into the signal where the desired time-range starts, followed by the number of seconds into the signal where the desired time-range ends, separated by a comma; and then, once again, click on the “Refresh” button. One can then analyse the truncated signal as normal.

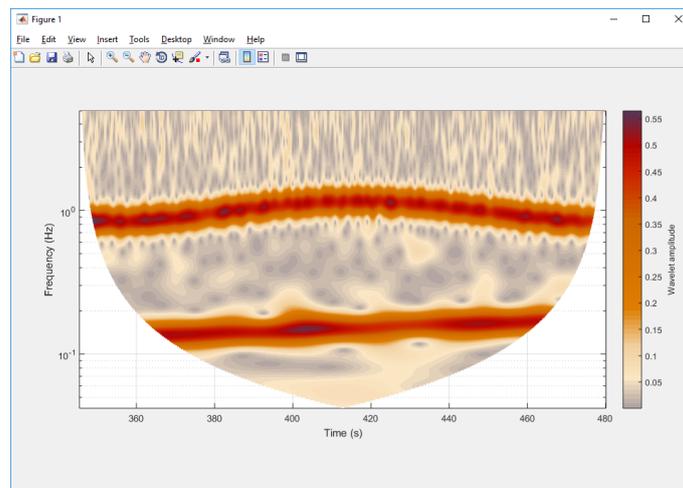


The time-range shown on both the graph of the selected signal and all results involving time will not start at 0 s but at whatever the start time of the chosen restricted time-range was. Note that performing new computations for the truncated signals will remove any previously computed results within the session (just like any other new computations will). Therefore, if the user has performed computations for the whole signal and now wishes to analyse only part of the signal, the user is advised to open a new window for the same application by clicking “File” and then “New workspace”, and in new this window to load the same file and truncate the signals as described.

**NOTE:** For results involving frequency or time-frequency analysis, **the minimum frequency for which results can be displayed increases** as the signal is truncated. For example, the above time-frequency representation for the truncated signal only shows two bands of high amplitude, whereas the time-frequency representation of the original signal had three; this is because the lowest-frequency band occurs below the minimum frequency visible for the truncated signal. In general, one cannot compute results for frequencies below the reciprocal of the duration of a signal being analysed.

## Plotting and saving

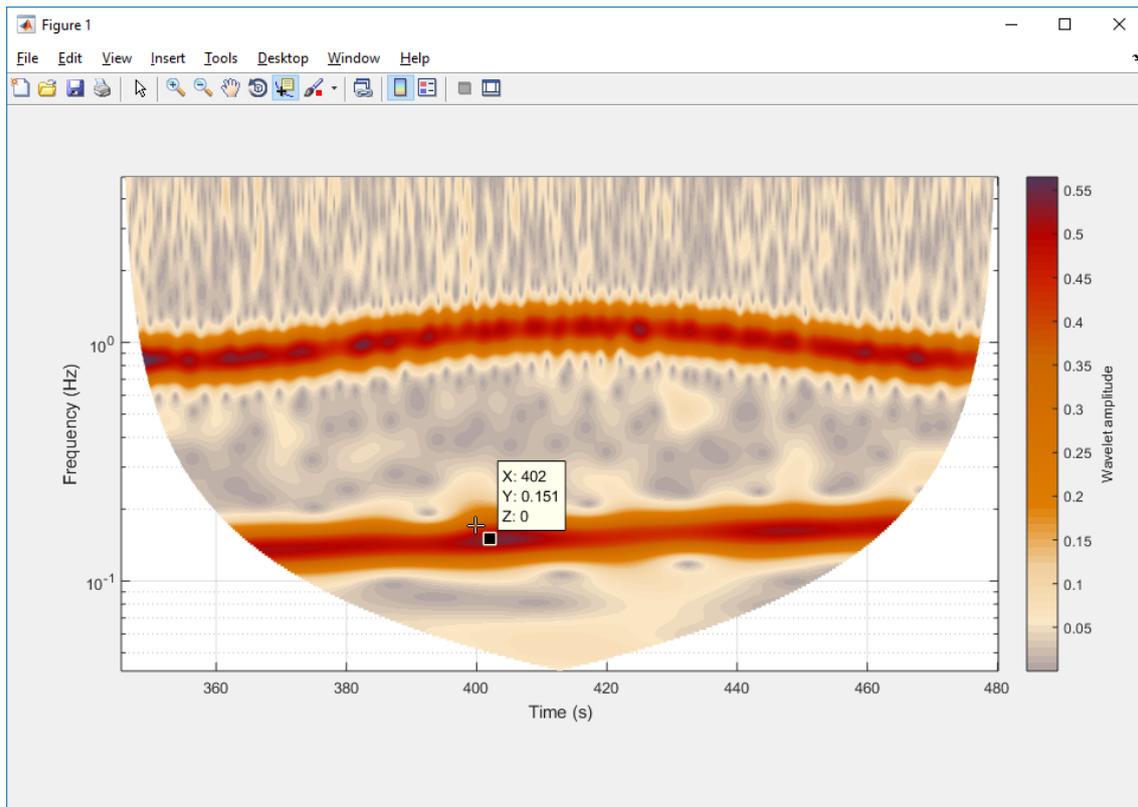
To **save one of the graphs currently displayed** in the window, click on “Plot” to the right of “File” and then select the appropriate option; this then opens the relevant graph in a new window as a MATLAB figure, which can then be saved.



If you have MATLAB, then saving the file as a .fig file means that you can edit the figure in MATLAB afterwards. Otherwise: If the figure consists of a “one-dimensional” plot, i.e. a plot simply showing graphs of one variable as a function of another variable, then we recommend saving the figure as a vector image such as .eps, .svg or .pdf, which can subsequently be edited in programs like Inkscape or Adobe Illustrator. However, if the figure includes a plot displaying some function of two variables, then we recommend saving the figure as a .png file, which is scalar.

The “Plot” facility for opening a graph in a new window has other benefits, apart from the ability to save:

- All of MATLAB’s features for analysing a figure, such as the “Data Cursor” facility for **identifying coordinates of any individual point** on a graph, become available.



- It is a convenient way of being able to keep the results of one computation before carrying out another computation, especially if you then wish to compare the results of the two computations.

In any one of the five applications, to save the current session such that you can return to it later, click “Save” in the top left corner of the window, and then “Save session”. This will enable you to save the current session as a .mat file. When you wish to return to this session, run MODA and choose the application in which the session was run, and then click on “File” and “Load previous session”; this will then enable you to browse and find the saved session.

Many of the actual numerical values displayed in the graphs can also be saved. Click “Save” and then either “Save as .mat” or “Save as .csv”. This will save numerical results for all the signals. In the Time-Frequency Analysis application, there is an option to “Save current as .mat”: this will save all the values in the time-frequency plot for the current signal; these values will not be saved under the “Save all as .mat/.csv” option which saves results for all signals, because the set of values in the time-frequency plot is very large. Saving as .mat saves the results in a MATLAB structure that can be opened by MATLAB; saving as .csv saves results in either one or more CSV files which can be read by Microsoft Excel (as well as by MATLAB). If you choose to save as .csv, *do not try to include or change the file extension in the File name, or change the file type in the “Save as type” pulldown list.*

Some numerical values saved will be “NaN” (“not a number”); this is not an error, but occurs where results are not computed. For example, in time-frequency analysis, one cannot compute meaningful results very close to the start or the end of the signal, and so (if the “Cut Edges” option is on) NaN will be saved for the values close to the start and the end. (Likewise, also, the plot itself will not show results there.)

When saving results from Ridge Extraction & Filtering for the purpose of loading into Dynamical Bayesian Inference, it is important that the results are saved as .mat, not as .csv.

**NOTE:** Files created and saved by MODA tend to be very large; so the user should *wait a while after saving* before attempting to open or run what was saved, otherwise the file which is actually still in the process of saving may become permanently corrupted.

More precise details of saving will be given in the sections of this manual devoted to each of the five applications.

## Large arrays and downsampling

If an array loaded into MODA is extremely large, this could cause MODA to run very slowly or even crash. The following factors could contribute to an array being very large:

- (i) The number of signals stored in the array is large.
- (ii) The signals were measured over a very long time.
- (iii) The signals were measured with a very high sampling frequency.

The first of these can be solved by splitting up the array into different files each containing fewer signals / signal pairs, and analysing these smaller blocks of signals separately in MODA. The downside of this is that one cannot carry out statistics (e.g. average wavelet power spectrum) over the full range of signals in the original array.

The second of the above issues can be solved by splitting up the time-interval over which the signals were measured into smaller time-intervals (preferably with slight overlap since time-frequency analysis cannot be carried out too close to the start or end of a signal); one can then create smaller arrays, each consisting of the set of signals restricted to one of the smaller time-intervals.

If the sampling frequency is many times higher (e.g. more than about 10 times higher) than the largest potential frequency of interest within the signals, then the third of the above issues can be solved by **downsampling**: Recall that in MODA, a *signal* is defined by a sequence of values  $(x_1, x_2, x_3, \dots, x_{k-1}, x_k)$  together with the sampling frequency  $f_s$  at which the recordings were made. The *downsampled version of this signal by factor  $n$*  is defined as the sequence of values  $(x_1, x_{n+1}, x_{2n+1}, \text{etc.})$  together with the new sampling frequency  $f_s^{new} = \frac{f_s}{n}$ . The number of values in this downsampled signal is equal to  $\frac{k}{n}$  (rounded up to the nearest integer), where  $k$  was the number of values in the original signal. Provided the new sampling frequency  $\frac{f_s}{n}$  is still several times higher than the largest potential frequency of interest within the signal, *all analysis of this downsampled version of the signal should give results that are essentially identical to the results of analysing the original signal, but the computation of the results will take much less time*. Note that if you are starting with an array of several signals, and you wish to downsample before analysing in MODA this set of signals, then the same downsampling must be applied to all the signals.

# Overview of analysis methods

In this section, we explain the analysis methods that the user can carry out:

- Time-Frequency Analysis
- Wavelet Phase Coherence
- Ridge Extraction & Filtering
- Wavelet Bispectrum Analysis
- Dynamical Bayesian Inference

We then explain the Method of Surrogates for testing significance of results quantifying possible connections between two oscillations (as features in Wavelet Phase Coherence, Wavelet Bispectrum Analysis and Dynamical Bayesian Inference).

## Time-Frequency Analysis

Given a signal  $x(t)$  that is periodic or quasiperiodic, it is well-known that one can express  $x(t)$  uniquely as a superposition of (typically infinitely many) sinusoidal signals  $A \cos(2\pi ft + \phi)$  of different frequencies  $f \geq 0$ Hz; this is known as **Fourier series decomposition**. For each  $f > 0$ Hz, if there exists a contribution  $A \cos(2\pi ft + \phi)$  with  $A \neq 0$ , then we may take  $A$  to be positive without loss of generality, in which case:

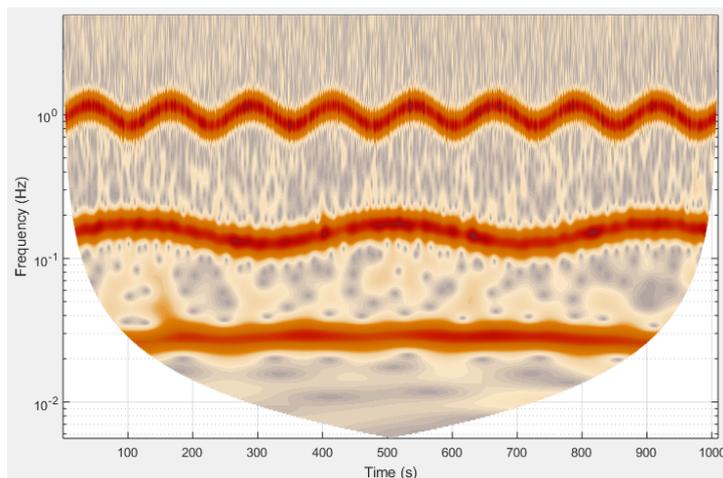
- we refer to  $A$  as the **amplitude** associated to  $f$ ;
- we refer to  $A^2$  as the **power** associated to  $f$ ;
- we define the **phase** associated to  $f$  at a time  $t$  to be the angle  $2\pi ft + \phi$  modulo  $2\pi$ ;

but if there is no contribution of frequency  $f$ , then the amplitude/power associated to  $f$  is just considered to be 0.

Of course, many signals in practice are only defined over a finite time-interval and are *not* well described as purely a quasiperiodic signal. Given a signal  $x(t)$ , a *time-frequency analysis* or *time-frequency representation* means a way to associate, *at each instant in time  $t$* , an amplitude/power and phase to each frequency-value  $f > 0$ Hz. (Naturally, if the amplitude assigned to a frequency-value  $f$  is *exactly* zero, then a phase need not be assigned to  $f$ .) The most obvious example is in the **ear**: the input is one continuous time-series—namely, air pressure against time—and the output is a time-varying spectrum of frequency values (perceived aurally as pitches) with associated amplitudes (perceived aurally as the associated loudnesses).

Now just as in the case of sound, so also many other signals that one might wish to measure (such as cardiac rhythms) will feature some finite number of particularly prominent oscillatory components whose frequency is not completely fixed but varies over time. Just as with the ear, the primary purpose of time-frequency analyses is to be able to isolate and study these oscillatory components.

To illustrate, here is an image of time-frequency analysis as computed in MODA, showing amplitude over time-frequency space; the amplitude  $A_{t,f}$  assigned to each point  $(t, f)$  in time-frequency space is denoted by colour coding, with the darker and more red colours signifying higher amplitude.



We see that there are three prominent oscillatory components in the signal, and that at least the highest two of these three components have frequencies that slowly modulate over time.

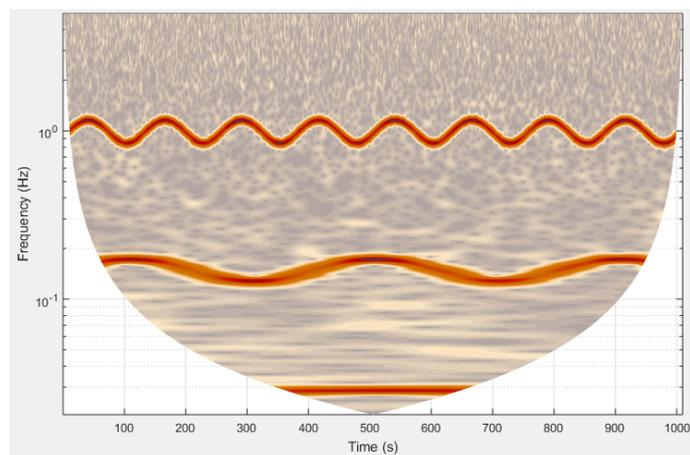
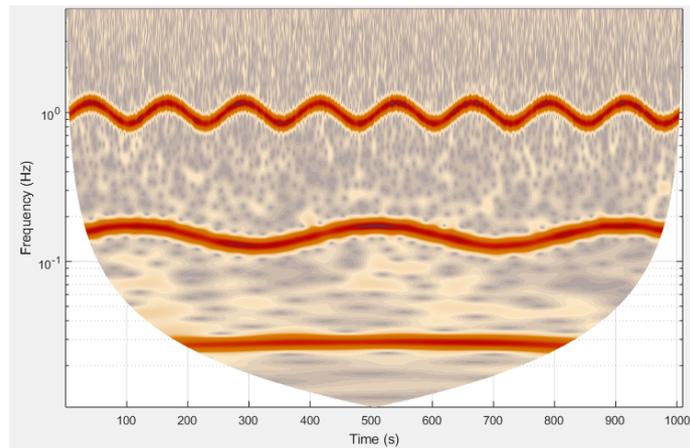
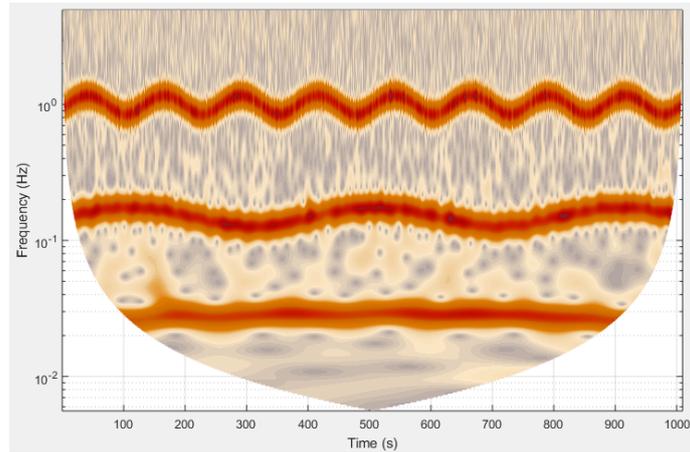
Not surprisingly, given only a small segment of a signal that does not contain several complete time-periods of the constituent oscillations, it is difficult to say precisely what the constituent frequencies are during this segment. Consequently, when seeking to give an instantaneous frequency analysis for a time-varying signal, there is always a **trade-off between time localisation and frequency resolution**: *To have a sharper and less blurry description of the constituent frequencies at any given moment, one needs to take account of how the signal looks in a wider time-interval around that moment; but doing so means that the sharpness of temporal changes in constituent frequencies is more blurred in time.* The impossibility of arbitrarily high frequency resolution at the same time as arbitrarily high time localisation is known as the **uncertainty principle**. This principle admits several possible mathematical formulations, depending on which measures of time localisation and frequency resolution are used.<sup>1</sup> For each time-frequency analysis method in MODA, there is a parameter denoted  $f_r$ , whose value the user can choose to specify; the higher this value is, the higher the frequency resolution and the lower the time resolution is. If the user leaves the  $f_r$  field blank, a default value of 1 is used. (The user is advised not to choose  $f_r$  smaller than about 0.5.)

Due to the above limitation, time-frequency analysis methods are most effective for analysing signals where the rates at which the frequencies of the main oscillatory components change in

<sup>1</sup>The most well-known formulation is the *Heisenberg-Gabor uncertainty principle*, which considers the “standard deviations” of the time-localisation window and the corresponding blurring in frequency space, and places a lower bound of  $\frac{1}{4\pi}$  on the product of these standard deviations.

time are not too fast in comparison with the instantaneous frequencies themselves. In other words, at any given time  $t$ , for each prominent oscillatory component  $\psi$  there ought to be a time-interval  $[t - \delta, t + \delta]$  about  $t$  containing several complete oscillations of  $\psi$ , such that the frequency of  $\psi$  is approximately constant over this time-interval.

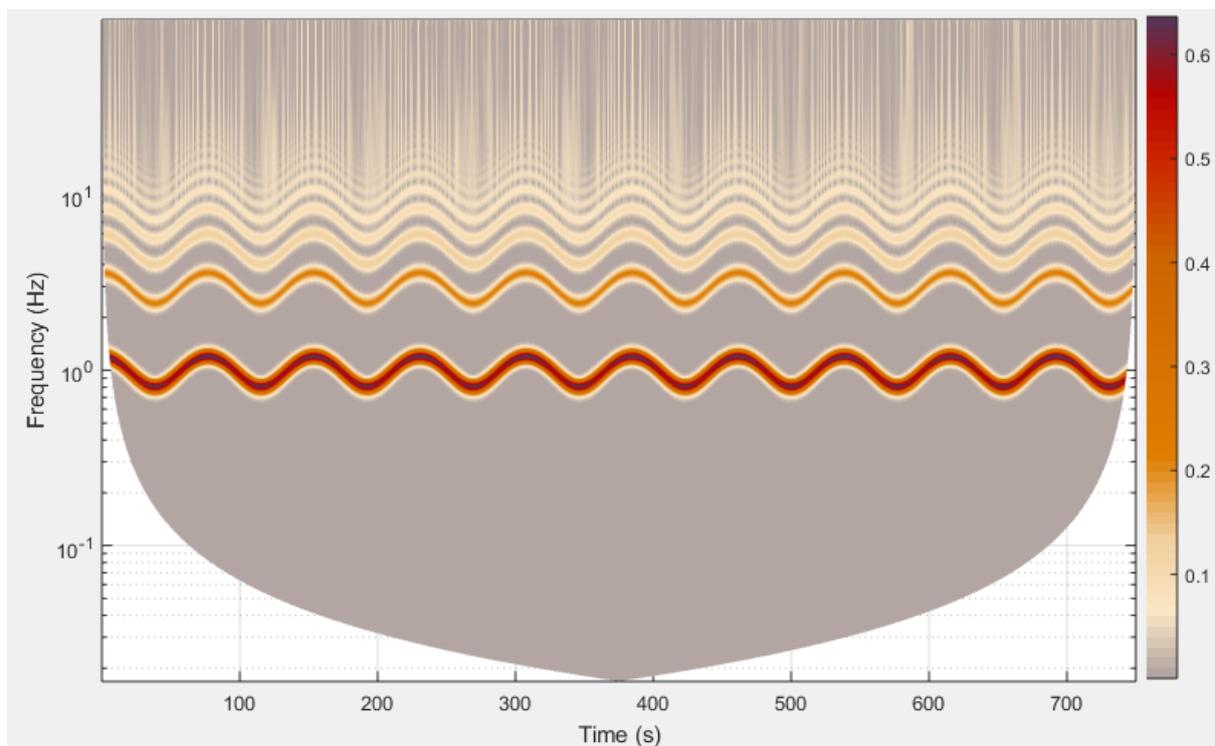
Here are time-frequency representations (with the lognormal wavelet) of the signal `1signal_10Hz.mat`, showing amplitude over time-frequency space, with  $f_r = 1$ ,  $f_r = 2$  and  $f_r = 4$  respectively:



We see that as  $f_r$  is increased, the frequency blur around the three prominent oscillatory components decreases in width. But on the other hand, for  $f_r = 4$ , the temporal variations in frequency of the middle component (approximately 0.2 Hz) are slightly distorted due to the relatively low time localisation.

## Nonlinear oscillations

Now just as periodic oscillations need not necessarily be perfect sinusoids, so likewise oscillations with time-varying frequency do not necessarily correspond to frequency-modulated sinusoids. For example, when we hear someone speaking, although we hear one main frequency in the voice at any instant in time, the shape of the oscillations at any given moment in time will not be sinusoids; and people's different-sounding voices (even when at the same pitch) correspond to different shapes of oscillation. Oscillations whose basic shape of oscillation is not sinusoidal are called *nonlinear oscillations*; this is because any periodic solution of a linear autonomous differential equation must be elliptical in shape, and so its 1-dimensional projections are sinusoidal in shape. For a periodic nonlinear oscillation of frequency  $f$ , the Fourier series decomposition involves a sinusoidal component at frequency  $f$  together with sinusoidal components at higher frequencies  $nf$  for  $n \geq 2$ ; these higher-frequency components are called *harmonics*. Likewise, a time-frequency analysis of a nonlinear oscillation with either fixed or time-varying frequency will reveal harmonics (whose frequency accordingly can vary in time as a multiple of the basic frequency). This is true also of the time-frequency-analyser in the ear; indeed, sometimes when one presses one of the lower keys on a piano, while hearing the main frequency one can also faintly hear a higher frequency, corresponding to one of the harmonics. Here is an image of a time-frequency analysis, showing amplitude over time-frequency space, of a frequency-modulated square wave:<sup>2</sup>



<sup>2</sup>The signal is  $x(t) = \text{sgn}(\sin(2\pi ft + \frac{a_m}{f_m} \sin(2\pi f_m t)))$ , with  $f = 1$  Hz,  $a_m = 0.2$  Hz and  $f_m = 0.013$  Hz, for a duration of 750 s starting at time  $t = 0$  s, sampled at 200 Hz.

The “fundamental frequency” is represented by the lowest (red) wavy band, or more precisely by the ridge of highest amplitude within this band (purple/dark red), and it oscillates between 0.8 Hz and 1.2 Hz; however, numerous higher-frequency harmonics can also be seen. Nonlinear oscillations occurring in practice are typically much less severely nonlinear (i.e. less distorted away from being sinusoidal) than a square wave, and so not as many harmonics would typically be clearly visible, and the visible harmonics would typically have smaller amplitude.

The signal represented in the above diagram is stored in the file `fm_square_200Hz.mat`, which is made available to the user. The signal is stored row-wise, i.e. as a single row vector. The above diagram was generated in MODA: the sampling frequency is 200 Hz; the time-frequency analysis was performed using a wavelet transform with the lognormal wavelet, with frequency resolution  $f_r = 2.4$ ; pre-processing was off.

### Rank sum and signed rank tests for power spectra

Time-frequency analysis enables one to calculate time-averaged amplitude-frequency or power-frequency spectra that are more physically meaningful than the traditional Fourier-based spectra for signals with time-varying-frequency components. This enables more meaningful results for the following statistical hypothesis tests, which can be performed in MODA.

**Rank sum test:** Suppose we have a population  $P$  of processes from any of which some signal  $x$  can be measured over time, and a population  $Q$  of processes from any of which some signal  $y$  can be measured over time. (The populations  $P$  and  $Q$  are allowed to overlap or even be the same.) We assume that  $x$ -values and  $y$ -values have the same units, so that spectra derived from them can be compared. Suppose we wish to know, for each frequency-value  $f$ , whether typically on average,  $x$ -signals measurable from  $P$  have the same power around frequency  $f$  as  $y$ -signals measurable from  $Q$ . One way to test this is using a *Wilcoxon rank sum test*: assuming some fixed duration  $T$  of both  $x$ -signals measured from  $P$  and  $y$ -signals measured from  $Q$ , we test at each  $f$ -value the null hypothesis that the median time-averaged power at frequency  $f$  of  $x$ -signals across the population  $P$  is the same as the median time-averaged power at frequency  $f$  of  $y$ -signals across the population  $Q$ . To carry out this test, we take some number  $m$  of independent random samples of  $P$  and measure signals  $x_1(t), \dots, x_m(t)$ , and we take some number  $n$  of independent random samples of  $Q$  and measure signals  $y_1(t), \dots, y_n(t)$ . *One should not choose to take any of the signals  $x_1, \dots, x_m$  and  $y_1, \dots, y_n$  to be measured from the same process, if there is any direct or indirect connection between  $x$ -signals and  $y$ -signals measured from the same process. The sample signals  $x_1, \dots, x_m, y_1, \dots, y_n$  are meant to be statistically independent of each other.* To implement the test in MODA, we need that all these signals are recorded at the same sampling frequency; the entire collection  $x_1, \dots, x_m, y_1, \dots, y_n$  can then be stored within a single array in a file readable by MATLAB. This file can then be loaded in the Time-Frequency Analysis application, and either the WFT or the WT computed for all the signals. MODA has the facility then to carry out the two-sided rank sum test on the pair of data sets  $\{\text{Power}_f(x_1), \dots, \text{Power}_f(x_m)\}$  and  $\{\text{Power}_f(y_1), \dots, \text{Power}_f(y_n)\}$  using MATLAB’s `ranksum` function, with a significance level  $\alpha$  that the user can specify. (Here,  $\text{Power}_f(\cdot)$  denotes time-averaged power at frequency  $f$ , under the WFT/WT with parameters as chosen by the user.)

**Signed rank test:** Suppose we have a population  $P$  of processes from any of which some pair of signals  $x$  and  $y$  can simultaneously be measured over time. Again, we assume that  $x$ -values and  $y$ -values have the same units, so that spectra derived from them can be compared. Assuming again some fixed duration  $T$  of  $(x, y)$ -signal pairs measured from  $P$ , suppose we wish to test for

each frequency-value  $f$  the null hypothesis that the mean/median over the population  $P$  of the signed difference between the time-averaged power of  $x$  at frequency  $f$  and the time-averaged power of  $y$  at frequency  $f$  is equal to 0. If we assume in addition that the distribution of this signed difference over the population  $P$  is symmetric about some (unknown) central value  $c_f$ —so that we simply wish to test whether  $c_f = 0$ —then we can apply the *Wilcoxon signed rank test*. We take some number  $n$  of independent random samples of  $P$  and measure the corresponding signal pairs  $(x_1(t), y_1(t)), \dots, (x_n(t), y_n(t))$ . Again, these must all be recorded at the same sampling frequency, so that they can then be stored as a single array  $[x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$ . Again, the file where this array is stored can be loaded in the Time-Frequency Analysis application, and either the WFT or the WT computed for all the signals. MODA has the facility then to carry out the two-sided signed rank test on the pair of ordered data sets  $(\text{Power}_f(x_1), \dots, \text{Power}_f(x_n))$  and  $(\text{Power}_f(y_1), \dots, \text{Power}_f(y_n))$  using MATLAB’s `signrank` function, with a significance level  $\alpha$  that the user can specify.

Both of the above tests can also be performed in MODA using amplitude rather than power.

### TFAs implemented in MODA

Two typical approaches to time-frequency analysis are **windowed Fourier transforms** (WFT) and **(continuous) wavelet transforms** (WT), both of which can be performed using MODA. These are defined mathematically for continuous-time signals; but as we have said, one does not input into MODA a continuous-time signal, but a discretely sampled signal. Due to the Nyquist-Shannon sampling theorem mentioned in the Introduction, MODA will only compute instantaneous amplitudes and phases for **frequencies up to half the sampling frequency**. Let us briefly describe the WFT and WT:

- **Windowed Fourier transform:** A signal  $x(t)$  can be represented by its *Fourier transform*. This is a representation that assigns an “amplitude” value and an “initial phase” value to a range of frequency values  $f$ , in such a manner as to cause approximately periodic components of the signal to stand out. Now the Fourier transform does not give any extra bias or weight to one part of the duration of the signal over any other part. However, one could carry out a “weighted” Fourier transform where the weight of priority is mostly concentrated around one moment in time  $t$ ; this is essentially what the windowed Fourier transform does, for each time  $t$  in the duration of the signal.<sup>3</sup> One can choose between different types of “window”—that is, the shape of the distribution of weight about each moment  $t$ . For any given shape of window, the frequency resolution is determined by how widely the window spreads about each moment of time. The disadvantage of windowed Fourier transforms is that the spread of the window about each moment in time needs to be sufficiently broad to capture several oscillations of the *lowest* prominent frequencies present, giving lower-than-necessary time resolution for the higher frequencies. Accordingly, the user is required to enter a minimum frequency of interest  $f_{\min}$ ; for a given shape of window, the width of the window is linearly proportional to the ratio  $f_r/f_{\min}$ .
- **Wavelet transform:** Here, the window taken about any given moment in time  $t$  is chosen afresh for each frequency-value under investigation, such that the spread of the window is inversely proportional to the frequency under investigation, maintaining constant “effective”

---

<sup>3</sup>Actually, if (as is most conventional) one simply takes this weighted Fourier transform, then the phase assigned to a time-frequency pair  $(t, f)$  will not represent the phase of a frequency  $\approx f$  component at the time  $t$  itself, but rather will represent the “initial” phase that this component would be at at time 0 if extended periodically with fixed frequency  $f$ . However, following [2], the formula used in MODA adds  $2\pi ft$  to the conventional formula, so that, just as with the WT, the phase assigned to  $(t, f)$  is genuinely the phase at time  $t$ .

frequency resolution across the different frequencies. Thus the above-mentioned problem with the windowed Fourier transform is eliminated. There are different types of “wavelet” that can be chosen in the formula for the wavelet transform. The formula for each type of wavelet makes reference to the frequency resolution parameter  $f_r$ ; the dependence on  $f_r$  is sometimes more complicated than in the case of the windowed Fourier transform, but the basic principle still holds: a higher value of  $f_r$  corresponds to a higher frequency resolution and lower time resolution.

One final remark: When performing a time-frequency analysis of a signal  $x(t)$ , one is typically interested in how the amplitude  $A_{t,f}$  associated to frequency  $f$  at time  $t$  varies over the  $(t, f)$ -space, or at least how it compares with the amplitudes of other signals at the same point in  $(t, f)$ -space. The  $A_{t,f}$ -value assigned to just a single isolated point  $(t, f)$  for just one individual signal has no useful meaning in and of itself. In other words, in general the amplitude spectrum is only physically meaningful up to an arbitrary constant of proportionality. Therefore, in the formulae for the WFT and the WT, MODA mostly uses whatever constant of proportionality keeps the formulae or the computations simple. All formulae are given at the end of the section on Time-Frequency Analysis.

Implementation of time-frequency analysis in MODA is exactly as described in detail in [2], except that negative-frequency-filtering is only performed if the “pre-processing” option is on.

### Why not just use the Fourier transform?

When investigating the presence of physically significant oscillatory components in a signal, a common approach is simply to take the Fourier transform of the whole signal and look for peaks. However, problems and limitations of this approach include the following:

- One cannot track frequency fluctuations or other changes in behaviour over time, since *only* an “overall” spectrum is shown. Real-world oscillatory processes are perpetually subject to temporal variations, and the ability to draw accurate and reliable conclusions from analysis of such processes is contingent upon temporal variations being tracked.
- Connected with the previous point: the distribution of Fourier power could depend significantly on exactly when the recording of the signal began and ended.
- If there are a few peaks that stand out against an overall “random-looking” graph for the Fourier transform, one could easily come to the conclusion that the only physically interesting contributions to the signals are those represented by these peaks, and everything else is essentially just “noise”. However, a time-frequency analysis could yield other physically important oscillatory components that only stand out clearly during parts of the duration of the signal.
- Variable frequency of an oscillatory component over a long time could yield peaks in the Fourier transform that look misleadingly like distinct and separate oscillatory components.
- As a consequence of the previous point, even if one only wishes to see an “overall” spectrum without concern for tracking variations in time, the Fourier power spectrum may be a much less physically meaningful representation than, say, a time-averaged wavelet power spectrum.

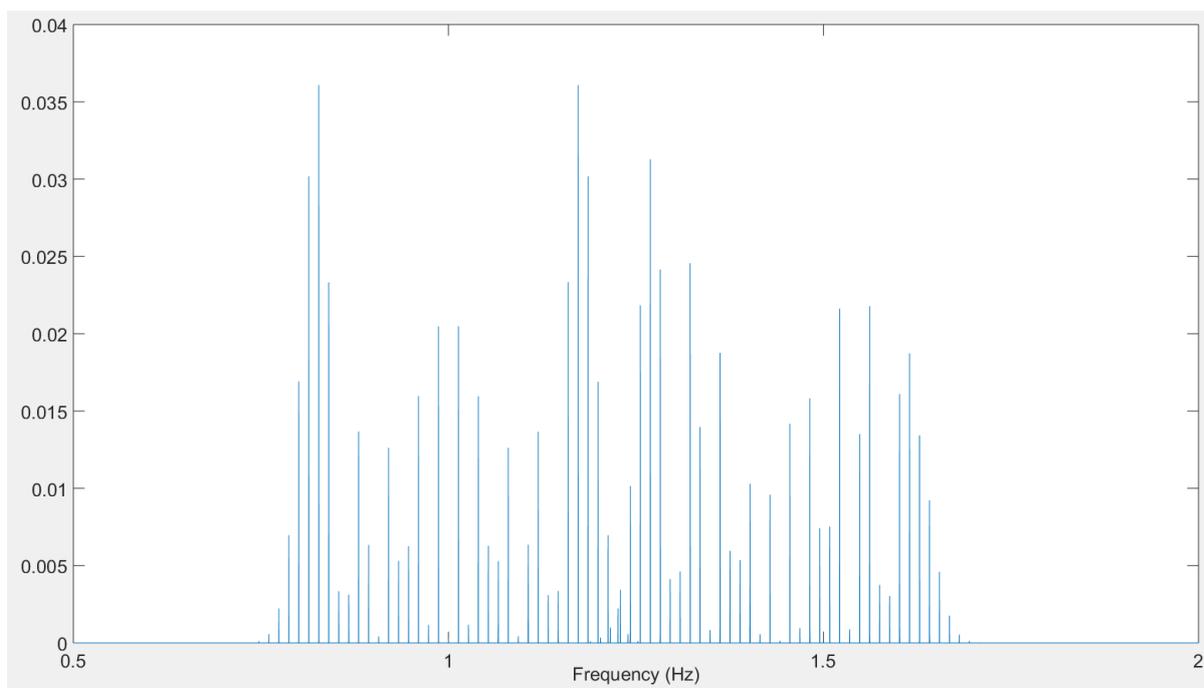
Let us expound and illustrate these last two points. Consider a slowly frequency-modulated oscillation. Physically, the oscillation is approximately a periodic oscillation on shorter time-scale,

but some process is causing the frequency of the oscillation to deviate back and forth slowly. However, if we suppose for example that the frequency modulation is itself roughly periodic, then the Fourier transform will treat the oscillation *not as locally periodic with time-varying frequency*, but more like a *quasiperiodic oscillation with fixed periodicities*. Thus, in the Fourier transform, there will be numerous *harmonics connected to the frequency of modulation itself*, which will not look remotely like the typical pattern of harmonics for nonlinear oscillations: indeed, there will be no visible peak at the “fundamental frequency” of modulation or at low multiples thereof. This is likely to be misleading, but even if one is experienced enough to know not to be misled by this, there is still no straightforward way of understanding from the Fourier transform what the true nature of the oscillations actually is.

For example, let us consider a superposition of two frequency-modulated sinusoids with sinusoidal frequency modulation. We will analyse the signal

$$x(t) = \sin(2\pi f_1 t + \frac{a_m}{f_m} \sin(2\pi f_m t)) + \sin(2\pi f_2 t + \frac{a_m}{f_m} \sin(2\pi f_m t))$$

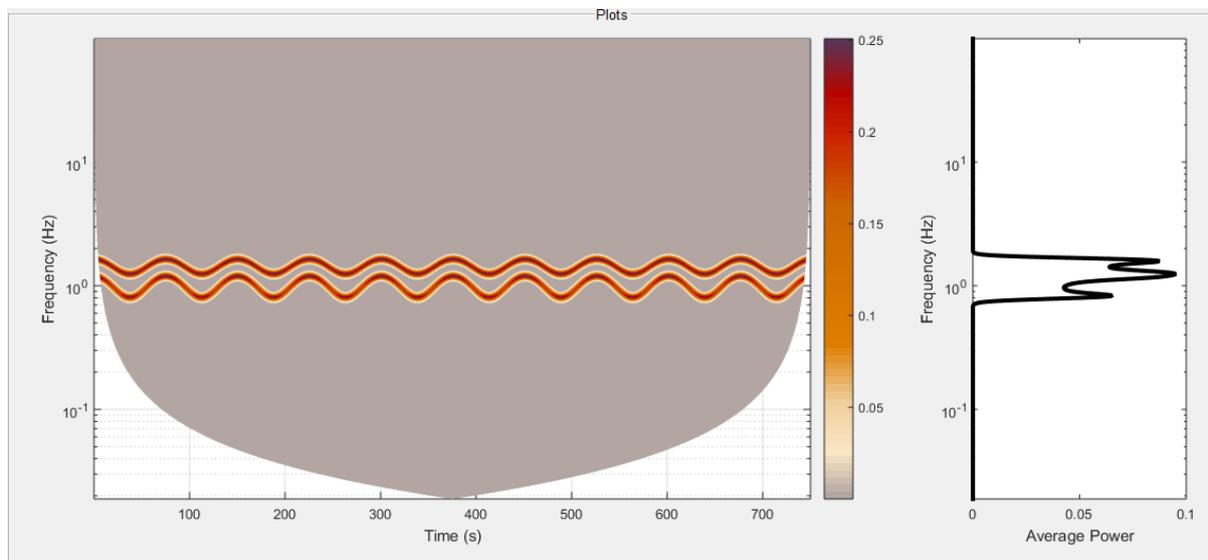
with  $f_1 = 1$  Hz,  $f_2 = 1.442$  Hz,  $a_m = 0.2$  Hz and  $f_m = 0.013297$  Hz, over a duration of 75200 s starting at time  $t = 0$  s, sampled at 200 Hz. The Fourier power spectrum looks as follows:



The apparent “vertical lines” are where the power sharply rises towards a peak and then sharply falls back down again towards 0. Since the duration of the signal is very long, the frequency resolution on the above graph is very high: the spacing between frequencies for the Fourier transform is approximately 13  $\mu$ Hz. The above graph only shows the range from 0.5 Hz to 2 Hz, but there were no other visible non-zero power values in the range from 0 Hz up to the Nyquist frequency 100 Hz.

So we see that there are numerous peaks in the Fourier transform. There is no obvious way to learn anything from this graph about the behaviour of the signal, and one could even be misled into thinking that the behaviour of the signal is highly complex, since the various peaks do not show a typical pattern of fundamental frequencies plus harmonics. The reality

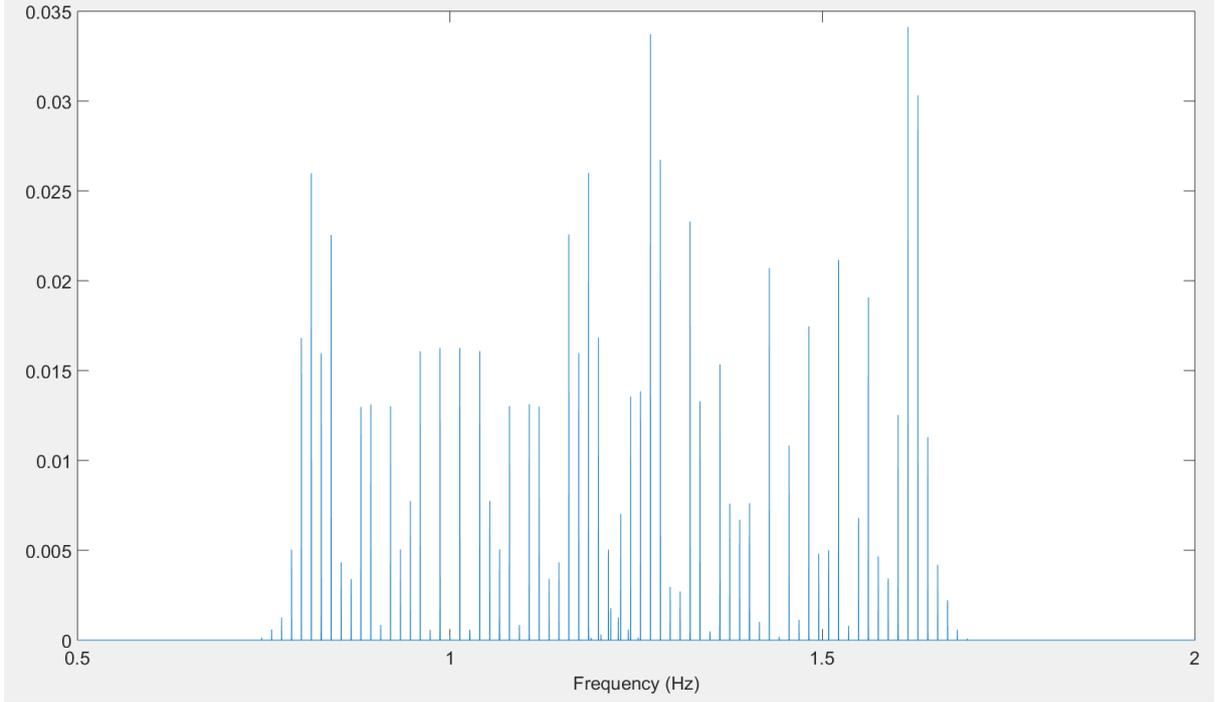
is that the signal only has two oscillatory components, both exhibiting a very straightforward behaviour, and moreover these components are variable-frequency *linear* oscillators, in that they are gradually frequency-modulated *sinusoids*.



This is seen in the above time-frequency representation of the first 750 s of the signal: two frequency-modulated oscillations are clearly seen, and the lack of harmonics suggests that these oscillations are locally sinusoidal in shape. The 750 s extract used to create the above figure is stored in the file `freq_mod_200Hz.mat` made available to the user; it is stored row-wise (i.e. as a single row vector), with a sampling frequency of 200 Hz. The above figure shows power over time-frequency space, using the wavelet transform with the lognormal wavelet with frequency resolution  $f_r = 2.7$ , with pre-processing off.

The time-averaged wavelet power, shown in the plot on the right, is taken only over the first 750 s of the original signal, but it would look essentially the same when taken over the entire 75200 s. As a time-averaged property, it cannot easily be used to deduce that there are two frequency-modulated oscillatory components in the signal. But nonetheless, it is still more physically meaningful than the Fourier power spectrum: the numerous gaps between peaks in the Fourier power spectrum arise from the fact that **each frequency-modulated oscillation is being represented as a linear superposition of fixed-frequency oscillations, even though such a representation has no physical relevance to any mechanism of frequency modulation**. By contrast, the average wavelet power shows a smoother continuum of power against frequency, corresponding to the proportion of time that the oscillations spend around each frequency value.

Finally, it is worth mentioning that there are other subtleties concerning the Fourier transform that make it less straightforward to interpret correctly. For example, one cannot necessarily tell visually, from the raw graph of the Fourier transform, how the Fourier power is actually distributed: if the signal is long enough for meaningful results, consecutive values in the frequency discretisation may be indistinguishable, making it potentially impossible to tell how much power is actually contained in an individual very narrow spike. So for example, if we ended the above 75200 s signal only 50 s earlier (such that the signal now has duration 75150 s), the actual distribution of Fourier power will remain essentially the same but the graph of Fourier power looks distinctly different:



There are methods of pre- or post-processing that could potentially help to overcome such an issue (such as adding vast amounts of zero padding), but these introduce their own complications (e.g. zero padding can actually influence the distribution of power, for example by introducing a discontinuity that shifts some of the power to high frequencies).

## Wavelet Phase Coherence

Suppose we have two simultaneous signals  $x_1(t)$  and  $x_2(t)$  defined over the same time-interval  $[0, T]$ , which (e.g. from looking at their time-frequency analyses) seem likely to share a common oscillatory influence, possibly with a time-varying frequency. One way that we can investigate this possible connection between the two signals is to look at **instantaneous phase coherence**, which in MODA is achieved using phases assigned by the wavelet transform. At each time  $t$ , let  $\phi_{t,f}^1$  be the instantaneous phase assigned to frequency  $f$  by the wavelet transform of  $x_1$ , and let  $\phi_{t,f}^2$  be the instantaneous phase assigned to frequency  $f$  by the wavelet transform of  $x_2$ . (The user can choose parameters for the wavelet transform, but the same parameters are used for both signals.) The instantaneous wavelet phase coherence at time  $t$  and frequency  $f$  is a number between 0 and 1 which measures how close to being constant the difference between  $\phi_{s,f}^1$  and  $\phi_{s,f}^2$  is over a time-interval of  $s$ -values surrounding the time  $t$ . In MODA, this time interval is taken to be 10 complete oscillations of the frequency  $f$ .

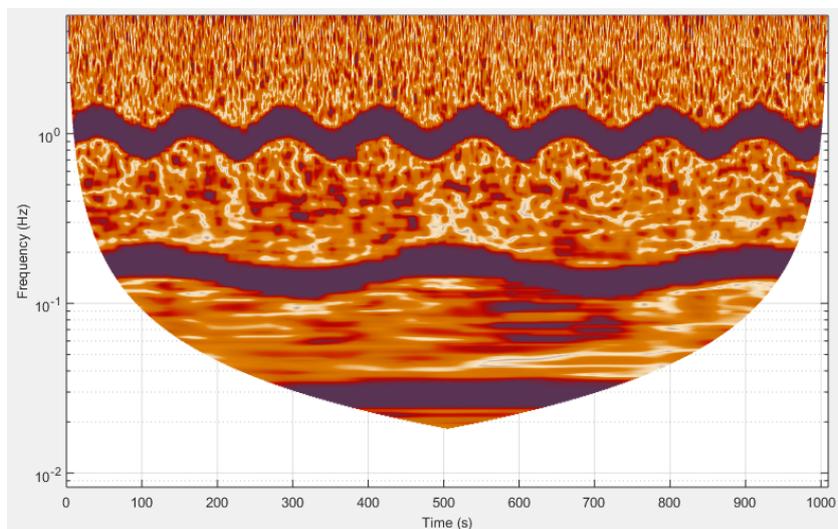
So, to be precise, the instantaneous wavelet phase coherence  $\rho_{t,f}$  (as implemented in MODA) is defined as

$$\rho_{t,f} = \frac{f}{10} \left| \int_{t-\frac{5}{f}}^{t+\frac{5}{f}} e^{i(\phi_{s,f}^1 - \phi_{s,f}^2)} ds \right|. \quad (1)$$

If the phase difference  $\phi_{s,f}^1 - \phi_{s,f}^2$  is constant over the time-interval  $[t - \frac{5}{f}, t + \frac{5}{f}]$ , then  $\rho_{t,f} = 1$ .

The presence of a shared oscillatory influence between  $x_1$  and  $x_2$  is likely to be indicated by a “ridge” in  $(t, f)$ -space, that is, a continuous path  $\{(t, f_t) : t \in [0, T]\}$  in the  $(t, f)$ -space such

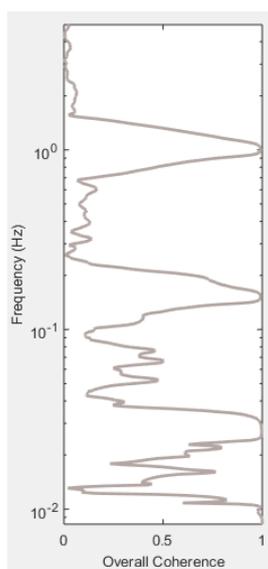
that on this curve the phase coherence values  $\rho_{t,f_t}$  are high. This suggests that some oscillatory process, with frequency approximately equal to  $f_t$  at each time  $t$ , is being picked up in both signals. In MODA, this could look as follows:



One can also plot, as a function of frequency  $f$ , the **overall phase coherence**  $\rho_f$  between the two signals at frequency  $f$ , defined as

$$\rho_f = \frac{1}{T - 2\epsilon_f} \left| \int_{\epsilon_f}^{T - \epsilon_f} e^{i(\phi_{s,f}^1 - \phi_{s,f}^2)} ds \right|. \quad (2)$$

Here, the number  $\epsilon_f$  (which is small for frequencies  $f$  that are not too low) is due to the fact that the wavelet transform is not really well-defined for times too close to the start and end of the recorded signal. This is due to the limited time localisation achievable by any time-frequency analysis. If the two signals share an oscillatory influence whose frequency fluctuates within a frequency interval  $[a, b]$ , then one expects to see high values of the overall phase coherence within this interval; for the same signals as used for the above figure, the overall phase coherence plot is as follows:



We warn that at low frequencies, the time-range  $[\epsilon_f, T - \epsilon_f]$  over which the overall phase coherence is taken is narrow, inherently leading to a bias towards high coherence values (as seen in the above figure) which generally have no real physical meaning.

From the overall phase coherence, one can test how significant the apparent presence of the shared oscillatory influence is, via the method of surrogates.

## Ridge Extraction & Filtering

MODA has facilities for locating and “extracting” individual oscillatory components of a signal from within some frequency-range of interest.

### Bandpass filtering

A signal  $x(t)$  can be represented by its *Fourier transform*. This is a representation that assigns an “amplitude” value and an “initial phase” value to a range of frequency values  $f$ , in such a manner as to cause approximately periodic components of the signal to stand out. One possible way to extract from a signal  $x(t)$  an oscillatory component whose frequency lies within a range  $[f_-, f_+]$  is to pass  $x(t)$  through a **bandpass filter with passband**  $[f_-, f_+]$ . This is an algorithm whose output  $x_{filt}(t)$  has the property that the Fourier transform assigns very low amplitudes to nonnegative<sup>4</sup> frequencies outside the range  $[f_- - \epsilon, f_+ + \epsilon]$  while roughly preserving the amplitudes and phases assigned to frequencies within the range  $[f_- + \epsilon, f_+ - \epsilon]$ , for some small  $\epsilon$ . Many bandpass filters in fact only seek to preserve well the amplitudes and not the phases, due to some other preferred property such as real-time analog implementability that is incompatible with phase-preservation. But since our purpose is to analyse and investigate signals rather than to control them, it is important that the filter preserves well both amplitudes and phases within the passband; filters with this property are called *zero-phase filters*. The bandpass filter available in the Ridge Extraction & Filtering application is the zero-phase **Butterworth** bandpass filter.

It is significant that there are small intervals  $[f_- - \epsilon, f_- + \epsilon]$  and  $[f_+ - \epsilon, f_+ + \epsilon]$  around boundary of the passband, where the filter “smoothly transitions” between the passing state and the attenuating state. The narrowness of such a transition region plays a similar role to the frequency resolution in a time-frequency analysis: Typical bandpass filters such as the Butterworth filter can be seen as a “time-localised” process, in the sense that the filtered signal  $x_{filt}$  at time  $t$  is very little affected by how the original signal  $x$  behaves at times long before or after  $t$ ; however, the sharper the transition around  $f_-$  and  $f_+$ , the more the filtered signal at time  $t$  is affected by the original signal at times further away from  $t$ . For oscillatory components with time-varying frequencies whose frequency range comes close to one of the critical values  $f_-$  or  $f_+$ , this could significantly affect the results, particularly when “instantaneous” phases are extracted from the filtered signal.

For the Butterworth filter, the sharpness of the transition is determined by the *filter order*, which is a parameter taking positive integer values (multiples of 4 for the zero-phase bandpass version). The algorithm implemented in MODA finds the highest order for which the results are safely numerically stable.

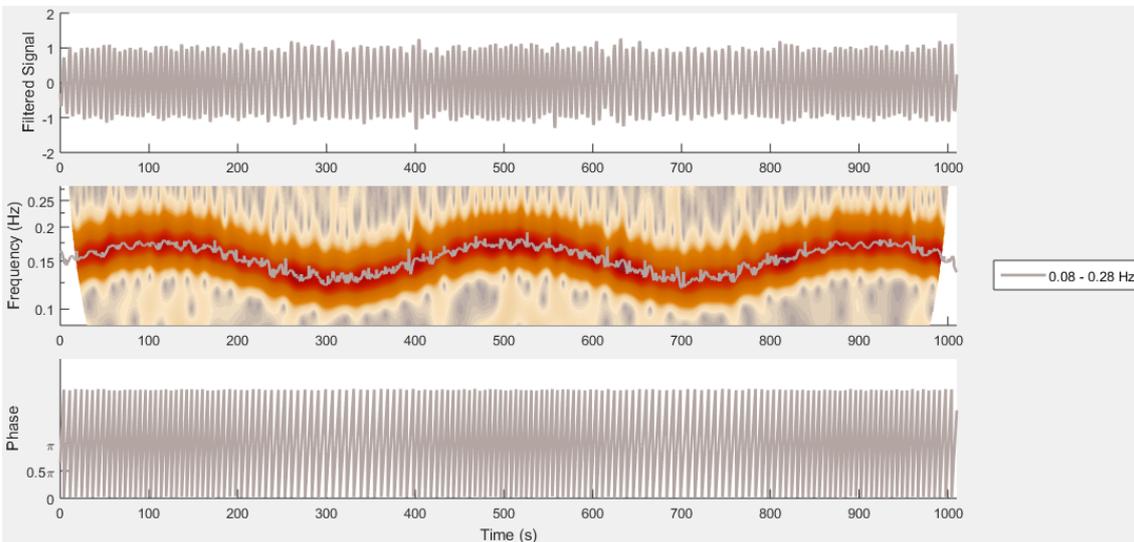
---

<sup>4</sup>For real-valued signals, amplitudes assigned to negative frequencies always match those assigned to their positive counterpart, and phases assigned to negative frequencies are always the negative of those assigned to their positive counterpart.

Once the signal has been filtered, one can associate a phase  $\phi_t$  at each time  $t$  to the filtered signal  $x_{filt}$ ; the phase  $\phi_t$  is defined as the argument of the complex number  $\mathcal{A}x_{filt}(t)$  where  $\mathcal{A}x_{filt}$  denotes the **analytic representation** of  $x_{filt}$ . Analytic representation is a linear operation with the property that a signal  $\cos(2\pi ft + \phi)$  is mapped to the complex-valued signal  $e^{2\pi ift + \phi}$ . In general, analytic representation is a type of “brick-wall filter” and so is not really a time-localised operation; however, in the context of filtering, for a typical zero-phase bandpass filter (including the Butterworth filter) the overall operation  $x \mapsto \mathcal{A}x_{filt}$  is valid as a time-localised operation (in the sense that the result at time  $t$  is affected very little by the behaviour of  $x$  at times long before and after  $t$ ). Thus, it still makes sense to construct the phase  $\phi_t$  of  $x_{filt}$  at time  $t$ . Nonetheless, the physical meaningfulness of this relies on the assumption that there is only one predominant oscillatory process influencing  $x(t)$  within the frequency band  $[f_-, f_+]$ .

## Ridge-extraction

A **higher-precision method of extracting oscillatory components** than bandpass filtering is identification and extraction of “ridges” from a time-frequency representation.



Suppose a signal  $x(t)$ , defined over a time-interval  $I$ , has a significant oscillatory component whose frequency may be time-varying but which lies within a certain frequency range  $[f_-, f_+]$ , and no other significant component in this range. Then a time-frequency analysis will show this oscillation, although with a blurry frequency at each time due to the uncertainty principle.

Nonetheless, MODA has the facility to identify a path  $\{(t, f_t) : t \in I\}$  in the time-frequency space from within a frequency strip  $[f_-, f_+]$  specified by the user, with the property that the instantaneous amplitudes at each time are generally maximised around this path. *It is important that band  $[f_-, f_+]$  specified by the user not only contains the range of frequencies within which the oscillatory component is expected to lie, but also all of the blur around these frequencies visible within the time-frequency representation.* Such a path  $(f_t)_{t \in I}$  is called a **ridge**; if there is one significant oscillatory component of  $x(t)$  within the frequency range  $[f_-, f_+]$ , then the ridge should correspond fairly accurately to the true location of this oscillatory component in time-frequency space. MODA extracts the oscillatory component  $x_{filt}(t)$  itself as

$$x_{filt}(t) = cA_{t,f_t} \cos(\phi_{t,f_t})$$

where  $A_{t,f_t}$  and  $\phi_{t,f_t}$  are the amplitude and phase assigned by the time-frequency representation to the point  $(t, f_t)$  on the ridge in time-frequency space, and  $c$  is the necessary constant (determined by the time-frequency analysis options) to recover true amplitudes, as in [2]. The procedure implemented in MODA for identification of ridges is Scheme II( $\alpha=1, \beta=1$ ) in [3]. (Issues of frequency discretisation for extracted amplitudes and phases are also dealt with as described in [2].)

## Wavelet Bispectrum Analysis

Wavelet bispectrum analysis is a method of *either* looking for coupled oscillatory components within a signal (via the **autobispectrum**) *or* looking for coupling between oscillatory components of different signals (via the **crossbispectrum**).

We will first explain the conceptual and mathematical motivation behind bispectral analysis (in the sections “Harmonics of coupled oscillators” and “Looking for coupling within a signal”). We will then state the definitions of wavelet autobispectra and wavelet crossbispectra and describe how they are used to investigate coupling.

### Harmonics of coupled oscillators

Suppose we have a system

$$\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1)$$

with an attracting  $T_1$ -periodic solution, and a system

$$\dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2)$$

with an attracting  $T_2$ -periodic solution, with  $T_2 \neq T_1$ . Suppose we measure a signal  $p_1(t)$  that is influenced by a trajectory of either or both of these systems, i.e.

$$p_1(t) \approx \pi_1(\mathbf{x}_1(t)) + \pi_2(\mathbf{x}_2(t)) + w(t) \quad (3)$$

where  $\pi_1, \pi_2$  are real-valued functions, and  $w(t)$  represents the net of any remaining influences other than  $\mathbf{x}_1$  or  $\mathbf{x}_2$ . Then we expect that a frequency analysis of  $p_1(t)$  will include peaks at frequencies  $1/T_1$  and  $1/T_2$ , as well as perhaps peaks at harmonic frequencies—that is, at multiples of  $1/T_1$  and at multiples of  $1/T_2$ —if the attracting solutions are not elliptic harmonic motions and/or if the functions  $\pi_1, \pi_2$  are nonlinear.

Now suppose that a coupling is introduced between the two systems, giving a new system

$$\begin{cases} \dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1) + \varepsilon g_1(\mathbf{x}_1, \mathbf{x}_2) \\ \dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2) + \varepsilon g_2(\mathbf{x}_1, \mathbf{x}_2). \end{cases} \quad (4)$$

(It could be that  $g_1 \equiv 0$  or that  $g_2 \equiv 0$ , in which case we refer to the coupling as *unidirectional coupling*.)

If  $\varepsilon$  were 0 then this system would have a 2-dimensional torus as an attractor, namely the product of the attracting periodic solution of  $\mathbf{x}_1$  and the attracting periodic solution of  $\mathbf{x}_2$ . So one can generally expect that for small  $\varepsilon$  the existence of a 2-dimensional attractor will persist. Moreover, solutions on this attracting manifold will approximately be quasiperiodic, with approximate periodicities  $T_1^\varepsilon$  and  $T_2^\varepsilon$  that are close to  $T_1$  and  $T_2$  respectively.

Accordingly, if the signal  $p_1(t)$  is still described by (3), then a frequency analysis of  $p_1(t)$  may well have peaks not only around  $n/T_1^\varepsilon$  and  $n/T_2^\varepsilon$  for integers  $n$ , but also around sums and differences thereof, i.e.  $\frac{m}{T_1^\varepsilon} + \frac{n}{T_2^\varepsilon}$  for integers  $m, n$  *including* negative  $m$  or  $n$ . Such extra harmonics arise from nonlinear quasiperiodic-like behaviour. The functions  $g_1, g_2$  are not required to be nonlinear for this to occur, as nonlinearity of  $f_1$  and/or  $f_2$  may already be enough.

All of the above consideration has been for autonomous dynamics. But one can also consider a nonautonomous version where, locally around each time  $t$ , the behaviour is approximately as described above; so now we have a coupled system

$$\begin{cases} \dot{\mathbf{x}}_1 &= f_1(t, \mathbf{x}_1) + \varepsilon g_1(t, \mathbf{x}_1, \mathbf{x}_2) \\ \dot{\mathbf{x}}_2 &= f_2(t, \mathbf{x}_1) + \varepsilon g_2(t, \mathbf{x}_1, \mathbf{x}_2) \end{cases} \quad (5)$$

and we still assume that the recorded signal  $p_1(t)$  is described by (3). Then we expect a *time*-frequency analysis to show, at each time  $t$ , peaks in the instantaneous amplitude around the approximate instantaneous frequencies  $\frac{1}{T_1^\varepsilon(t)}$  and  $\frac{1}{T_2^\varepsilon(t)}$ , and it may well also show peaks around some harmonic instantaneous frequencies  $\frac{m}{T_1^\varepsilon(t)} + \frac{n}{T_2^\varepsilon(t)}$ .

### Looking for coupling within a signal

In view of the above, if we wish to test for the presence of coupled oscillatory components within a signal, we may wish to carry out an investigation similar to Wavelet Phase Coherence, except now looking at the instantaneous coherence between  $m\phi_{t,f_1} + n\phi_{t,f_2}$  and  $\phi_{t,mf_1+nf_2}$  for some  $m$  and  $n$ , across the space of frequency-pairs  $(f_1, f_2)$ . One practical difficulty is that this essentially requires a four-dimensional plot, unlike Wavelet Phase Coherence which only requires a three-dimensional plot (with colour coding for one of the dimensions). Nonetheless, if we assume that the instantaneous periodicities of oscillatory components vary over time within a small interval, then a useful starting point could be to look at the *overall* coherence between  $m\phi_{t,f_1} + n\phi_{t,f_2}$  and  $\phi_{t,mf_1+nf_2}$ , defined analogously to the overall phase coherence in the section on Wavelet Phase Coherence. However, it would also be useful if our measure of overall coherence were able to take into account *amplitudes* at the different frequencies, so as not to be misled by apparent coherence with harmonic frequencies where detectable harmonics do not actually exist. Therefore, in the integral for overall coherence (analogous to (2) in the section on Wavelet Phase Coherence), one can weight the integrand by the product of the amplitudes  $A_{t,f_1}A_{t,f_2}A_{t,mf_1+nf_2}$ ; if the coherence is itself perfect, then the result is just the time-average of this product of amplitudes. In the case that  $m = n = 1$ , the amplitude-weighted integral for overall coherence is called the **wavelet bispectrum**. Given a wavelet bispectrum plot, one can then select individual frequency pairs  $(f_1, f_2)$  at which to view the time-evolution of the **instantaneous wavelet bispectrum** (i.e. the integrand in the formula for the wavelet bispectrum).

### Wavelet (auto)bispectrum

Suppose we have a signal  $p_1(t)$  defined over a time-interval  $[0, T]$ . For each time  $t$  and frequency  $f$ , write  $A_{t,f}$  and  $\phi_{t,f}$  for the amplitude and phase assigned to  $f$  at time  $t$  by the wavelet transform of  $p_1$  (under some choice of parameters, which the user can specify). We define at each time  $t$  the **instantaneous biphas** associated to a pair of frequencies  $(f_x, f_y)$  by

$$\phi_{111}(t, f_x, f_y) = \phi_{t,f_y} + \phi_{t,f_x} - \phi_{t,f_x+f_y}$$

and the **instantaneous biamplitude** associated to a pair of frequencies  $(f_x, f_y)$  by

$$A_{111}(t, f_x, f_y) = A_{t, f_y} A_{t, f_x} A_{t, f_x + f_y}.$$

Now the wavelet transform itself assigns an amplitude and phase to each time-frequency pair  $(t, f)$  in the form of a *single complex number*

$$W_1(t, f) = A_{t, f} e^{i\phi_{t, f}}.$$

With this, we define the **(auto)bispectrum**  $b_{111}$  of  $p_1$  to be the function on frequency-frequency space given by

$$b_{111}(f_x, f_y) := \frac{1}{T - 2\epsilon_{f_x, f_y}} \int_{\epsilon_{f_x, f_y}}^{T - \epsilon_{f_x, f_y}} A_{111}(t, f_x, f_y) e^{i\phi_{111}(t, f_x, f_y)} dt \quad (6)$$

$$= \frac{1}{T - 2\epsilon_{f_x, f_y}} \int_{\epsilon_{f_x, f_y}}^{T - \epsilon_{f_x, f_y}} W_1(t, f_y) W_1(t, f_x) \overline{W_1(t, f_x + f_y)} dt \quad (7)$$

where  $\overline{\phantom{x}}$  denotes the complex conjugate, and (as with Wavelet Phase Coherence)  $\epsilon_{f_x, f_y}$  is due to the impossibility of computing meaningful wavelet transform values too close to the start and end of the signal. We refer to the modulus  $|b_{111}(f_x, f_y)|$  of the bispectrum as the *(overall) biamplitude*. We will often use the terms “biamplitude” and “biphase” with no further descriptor, to refer to *either* the instantaneous version *or* (in the case of “biamplitude”) the overall version.

By plotting the biamplitude  $|b_{111}|$ , one can search for points  $(f_x, f_y)$  in frequency-frequency space around which the biamplitude is relatively high. Such a point may have relatively high biamplitude by “fluke” due to noise (actual, measurement or numerical) or other coincidental factors, so one can investigate further by looking at how the instantaneous biphase  $\phi_{111}(t, f_x, f_y)$  evolves with time: if it stays roughly flat—or at least, if there are time-intervals during which it stays roughly flat—then this is a good indicator of the likelihood of coupling. This coupling could either be between oscillatory components at approximate frequencies near  $f_x$  and  $f_y$ , or near  $f_x$  and  $f_x + f_y$ , or near  $f_y$  and  $f_x + f_y$  (or, in theory, possibly any other pair of frequencies whose integer span includes  $f_x$  and  $f_y$ ). However, if the biphase  $\phi_{111}(t, f_x, f_y)$  is either constantly drifting anticlockwise or constantly drifting clockwise, then the high biamplitude cannot be taken as evidence of coupling.

In the above description of how to look for evidence of coupling, it is not necessarily clear what should be considered as a significantly high biamplitude at any given point in frequency-frequency space. The method of surrogates can be applied to help with this question.

Note that a nonlinear oscillatory component of the signal  $p_1$  around a frequency  $f_x$  may give a high biamplitude and roughly constant instantaneous biphase at the point  $(f_x, f_x)$  on the diagonal in frequency-frequency space. More subtly, if one sees a relatively high biamplitude and roughly constant instantaneous biphase around a pair of frequencies  $(f_x, f_y)$  whose ratio is an integer or a low-denominator rational, this could potentially also be due to a single nonlinear oscillator without any coupling.

The drawback of the above approach to bispectral analysis is, as we have already hinted, that oscillators with time-varying frequencies will cause a “spread” over time in the positions of the peaks of instantaneous biamplitude, such that the overall effect on the  $|b_{111}|$  plot might be not so clear. And similarly, if the variation of frequency is not slow enough, then at any one fixed frequency pair  $(f_x, f_y)$ , there might not be clear time-intervals during which the instantaneous biphase remains roughly flat.

## Wavelet crossbispectrum

Suppose we have two simultaneous signals  $p_1(t)$  and  $p_2(t)$  defined over the time-interval  $[0, T]$ , with wavelet transforms  $W_1(t, f) = A_{1;t,f} e^{i\phi_{1;t,f}}$  and  $W_2(t, f) = A_{2;t,f} e^{i\phi_{2;t,f}}$  respectively. We can investigate the possibility of a common oscillatory influence on the two signals by doing a **crossbispectral analysis**: If there are two coupled oscillators  $\mathbf{x}_1$  and  $\mathbf{x}_2$  such that  $p_1$  is affected by  $\mathbf{x}_1$  and/or  $\mathbf{x}_2$  and also  $p_2$  is affected by  $\mathbf{x}_1$  and/or  $\mathbf{x}_2$ , or if there is a single nonlinear oscillator  $\mathbf{x}$  such that  $p_1$  and  $p_2$  are both affected by  $\mathbf{x}$ , this may be revealed by mixing together  $W_1$  and  $W_2$  in the above bispectral formulae and then carrying out the same analysis.

We define at each time  $t$  the **instantaneous biphases**  $\phi_{122}$  and  $\phi_{211}$  associated to a pair of frequencies  $(f_x, f_y)$  by

$$\begin{aligned}\phi_{122}(t, f_x, f_y) &= \phi_{1;t,f_y} + \phi_{2;t,f_x} - \phi_{2;t,f_x+f_y} \\ \phi_{211}(t, f_x, f_y) &= \phi_{2;t,f_y} + \phi_{1;t,f_x} - \phi_{1;t,f_x+f_y}\end{aligned}$$

and the **instantaneous biamplitudes**  $A_{122}$  and  $A_{211}$  associated to a pair of frequencies  $(f_x, f_y)$  by

$$\begin{aligned}A_{122}(t, f_x, f_y) &= A_{1;t,f_y} A_{2;t,f_x} A_{2;t,f_x+f_y} \\ A_{211}(t, f_x, f_y) &= A_{2;t,f_y} A_{1;t,f_x} A_{1;t,f_x+f_y}.\end{aligned}$$

We define the **crossbispectra**  $b_{122}$  and  $b_{211}$  by

$$b_{122}(f_x, f_y) := \frac{1}{T - 2\epsilon_{f_x, f_y}} \int_{\epsilon_{f_x, f_y}}^{T - \epsilon_{f_x, f_y}} A_{122}(t, f_x, f_y) e^{i\phi_{122}(t, f_x, f_y)} dt \quad (8)$$

$$= \frac{1}{T - 2\epsilon_{f_x, f_y}} \int_{\epsilon_{f_x, f_y}}^{T - \epsilon_{f_x, f_y}} W_1(t, f_y) W_2(t, f_x) \overline{W_2(t, f_x + f_y)} dt \quad (9)$$

$$b_{211}(f_x, f_y) := \frac{1}{T - 2\epsilon'_{f_x, f_y}} \int_{\epsilon'_{f_x, f_y}}^{T - \epsilon'_{f_x, f_y}} A_{211}(t, f_x, f_y) e^{i\phi_{211}(t, f_x, f_y)} dt \quad (10)$$

$$= \frac{1}{T - 2\epsilon'_{f_x, f_y}} \int_{\epsilon'_{f_x, f_y}}^{T - \epsilon'_{f_x, f_y}} W_2(t, f_y) W_1(t, f_x) \overline{W_1(t, f_x + f_y)} dt, \quad (11)$$

where, again, the numbers  $\epsilon$  and  $\epsilon'$  are due to the impossibility of computing wavelet transforms too close to the start and end of the signal. Once again, the magnitudes of these quantities are referred to as biamplitudes.

By plotting either of the two biamplitudes  $|b_{ijj}|$  ( $i \neq j$ ), one can search for points  $(f_x, f_y)$  in frequency-frequency space around which the biamplitude is relatively high, and from there one can check whether the instantaneous biphas  $\phi_{ijj}(t, f_x, f_y)$  stays roughly flat, at least during some time-intervals. For  $f_x \neq f_y$ , a positive result may be evidence of a coupled-oscillator system that is influencing both  $p_1$  and  $p_2$ ; for  $f_x = f_y$ , a positive result may be evidence of a nonlinear oscillator that is influencing both  $p_1$  and  $p_2$ . If the ratio between  $f_x$  and  $f_y$  is an integer or low-denominator rational, then a positive result could also possibly be due to a single nonlinear oscillator influencing both signals. As before, the method of surrogates can be applied to test significance of the biamplitudes.

The same potential drawbacks as for the autobispectra still hold.

## Dynamical Bayesian Inference

Suppose we have a pair of oscillators  $(\mathbf{x}^1, \mathbf{x}^2)$ , but we do not know *a priori* whether they are coupled, and if they are coupled, what precise form the coupling takes. An approximation that can sometimes be made, especially if it is known that there is at most a weak coupling, is that the joint “phase” dynamics can be decoupled (unidirectionally) from the overall dynamics, giving an equation

$$\begin{cases} \dot{\phi}^1 &= q_1(t, \phi^1, \phi^2) \\ \dot{\phi}^2 &= q_2(t, \phi^1, \phi^2) \end{cases} \quad (12)$$

where  $\phi^i(t)$  is the “phase” of  $\mathbf{x}^i$  at time  $t$ . One way that the phase  $\phi^i$  can be defined in such an approximation is as the extracted phase from a signal  $x(t)$  with an oscillatory component due to the influence of  $\mathbf{x}^i$ ; such extraction can be performed by either of the two methods described in the section on Ridge Extraction & Filtering. Of course, this requires that  $x(t)$  has no other oscillatory components occupying a frequency band that interferes with the (potentially time-evolving) frequency of  $\mathbf{x}^i$ . Since we have two oscillators  $\mathbf{x}^1$  and  $\mathbf{x}^2$ , it could either be that  $\phi^1$  and  $\phi^2$  are extracted from the same signal  $x(t)$  or that they are extracted from different signals  $x_1(t)$  and  $x_2(t)$ ; note, however, that the former is only valid if  $\mathbf{x}^1$  and  $\mathbf{x}^2$  occupy disjoint frequency bands.

Now suppose we have time-series for  $\phi^1(t)$  and  $\phi^2(t)$  extracted as above, and we wish to infer the functions  $q_1$  (called the “**coupling function from  $\phi^2$  to  $\phi^1$** ”) and  $q_2$  (called the “**coupling function from  $\phi^1$  to  $\phi^2$** ”). Of course,  $q_1$  and  $q_2$  have time  $t$  as an input; in other words, the system (12) may be nonautonomous. But if the variation in time is sufficiently slow, then it may be possible to infer a kind of “moving autonomous approximation” of the dynamical system (12) (analogous to the well-known concept of a “moving averaging” of a time-series). That is, we take a series of time-windows  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  covering the whole duration of the signal, and on each window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  a time-independent approximation  $q_i^{t_n}(\phi^1, \phi^2)$  of  $\{q_i(t, \phi^1, \phi^2) : t_n - \delta \leq t \leq t_n + \delta\}$  is inferred. The windows themselves may have overlap. In MODA, the level of overlap (if any) can be specified by the user.

In general, when seeking to infer a model from data, there is a danger of **overfitting**. As an extreme example, when trying to determine the relationship between two quantities  $x$  and  $y$ , suppose we take 100 samples  $(x_i, y_i)$  and observe that they all lie extremely close to the curve  $y = (x + \alpha)^2 + \beta$  for some  $\alpha$  and  $\beta$ ; it will still be possible to find higher-order polynomials that fit those data points better than the quadratic—indeed, there will even be an order-99 polynomial that passes exactly through all the data points—but such a model is unlikely to bear any relation to the true phenomena underlying the relationship between  $x$  and  $y$ , and is likely to give highly inaccurate predictions for future samples.

Thus, for trying to achieve a good model, one is faced with the conflicting requirements of having sufficiently many degrees of freedom on the one hand and not overfitting on the other hand. One way to help meet both requirements is to introduce some kind of noise directly into the model, where the noise is itself taken to follow some specific model with unknown parameters. One then tries to **infer probabilistically** from the data the “most likely” combination of an underlying deterministic model (within some pre-specified degrees of freedom) and a set of parameters for the noise.

This is exactly what happens in dynamical Bayesian inference. In general, the noise introduced could either be measurement noise, dynamical noise, or a combination of the two. In MODA,

only dynamical noise is treated as being present, in the form of nondegenerate additive zero-mean Gaussian uncorrelated noise. So within each time-window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$ , we seek to infer a stochastic differential equation of the form

$$\begin{cases} d\phi_t^1 &= q_1^{t_n}(\phi_t^1, \phi_t^2) dt + dW_t^1 \\ d\phi_t^2 &= q_2^{t_n}(\phi_t^1, \phi_t^2) dt + dW_t^2 \end{cases} \quad (13)$$

with  $((W_t^1, W_t^2))_{t \in [t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]}$  being a continuous stochastic process with stationary independent increments; the unknowns to be inferred are the functions  $q_i^{t_n}$  and a  $2 \times 2$  symmetric matrix  $E = (E_{ij})$  such that  $\delta t E$  is the covariance matrix of the pair of random variables  $(W_{t+\delta t}^1 - W_t^1, W_{t+\delta t}^2 - W_t^2)$  for any  $t$  and  $\delta t$ .

We refer to the set of all pairs of phases  $(\phi^1, \phi^2)$  as the **2-torus**. Any smooth function  $q$  on the 2-torus can be expressed as an infinite sum of two-dimensional Fourier components; and so the approximating coupling functions  $q_i^{t_n}$  can be chosen to lie within the span of some finite number of Fourier basis functions. To be precise: for any positive integer  $K$ , let  $V_K$  be the set of all functions  $q$  of the form

$$q(\phi^1, \phi^2) = a_0 + \left( \sum_{j=1}^K b_j \cos(j\phi^2) + c_j \sin(j\phi^2) \right) + \left( \sum_{i=1}^K \sum_{j=-K}^K d_{ij} \cos(i\phi^1 + j\phi^2) + e_{ij} \sin(i\phi^1 + j\phi^2) \right)$$

for some constants  $a_0, b_j, c_j, d_{ij}, e_{ij}$ . If we assume the model in (12), then at any time  $t$  the coupling functions  $q_1(t, \cdot)$  and  $q_2(t, \cdot)$  can be approximated arbitrarily well by functions from the set  $V_K$  if  $K$  is allowed to be large enough.

In MODA, the user specifies a  $K$ -value called the **Fourier order**; this specifies that the functions  $q_i^{t_n}(\phi^1, \phi^2)$  must come from the set  $V_K$ . Hence the Fourier order represents the level of freedom available for the inferring of these functions  $q_i^{t_n}(\phi^1, \phi^2)$ . More precisely, for a given Fourier order  $K$ , there are  $(2K + 1)^2$  degrees of freedom for each of the two functions  $q_i^{t_n}(\phi^1, \phi^2)$ .

So overall, for a given Fourier order  $K$ , for each time-window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  MODA will seek to infer statistically from the time-series  $(\phi^1(t), \phi^2(t))$ :

- values of the coefficients  $a_0^{(1)}, b_j^{(1)}, c_j^{(1)}, d_{ij}^{(1)}, e_{ij}^{(1)}$  in the Fourier expression for  $q_1^{t_n}$ ;
- values of the coefficients  $a_0^{(2)}, b_j^{(2)}, c_j^{(2)}, d_{ij}^{(2)}, e_{ij}^{(2)}$  in the Fourier expression for  $q_2^{t_n}$ ;
- the noise intensities  $E_{11}$  and  $E_{22}$ , and “co-intensity”  $E_{12}$ .

Thus, for each window, a Fourier order of  $K$  corresponds to a model with  $8K^2 + 8K + 5$  unknown parameters to be inferred. (The default Fourier order is 2, corresponding to 25 parameters for each of the two inferred coupling functions, plus the three statistical parameters for the noise, making 53 unknowns in total.)

The procedure by which MODA infers the above parameters involves calculations based on Bayes’ Theorem, hence the name *dynamical Bayesian inference*. After results are obtained for the first window  $[t_1 - \frac{1}{2}\delta, t_1 + \frac{1}{2}\delta]$  (where  $t_1$  itself is equal to  $\frac{1}{2}\delta$ ), the results obtained for each window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  are incorporated at the start of the statistical inference algorithm for the next window  $[t_{n+1} - \delta, t_{n+1} + \delta]$ . The user can specify a **propagation constant**  $p \geq 0$  determining how strongly the results of each window are incorporated into the calculation for the next window; a *larger* value of  $p$  means that the results for each window affect *less* the calculation for the

subsequent window. *On the one hand, the faster the functions  $q_1$  and  $q_2$  change with  $t$ , the larger the propagation constant needs to be for accurate results. But on the other hand, provided the functions  $q_1$  and  $q_2$  depend continuously on time  $t$ , being able to incorporate the result of the previous window improves the accuracy (especially when the level of noise in the system is high), and setting the propagation constant too large loses this opportunity for improved accuracy.* See [1, Sec. II.C] for further discussion.

We can use the method of surrogates to test whether  $\mathbf{x}^2$  really does influence the phase of  $\mathbf{x}^1$  and whether  $\mathbf{x}^1$  really does influence the phase of  $\mathbf{x}^2$ . If, during a time-window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$ , the phase of  $\mathbf{x}^i$  is not influenced by  $\mathbf{x}^j$ , and the frequency of  $\mathbf{x}^i$  remains roughly constant during this time-window, then the angular velocity of the extracted signal  $\phi^i(t)$  should be roughly constant. Therefore, we define within each window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  the test statistics

$$\epsilon_{2 \rightarrow 1}(t_n) := \left( \int_0^1 \int_0^1 [q_1^{t_n}(2\pi u, 2\pi v) - a_0^{(1)}]^2 du dv \right)^{\frac{1}{2}} \quad (14)$$

$$\epsilon_{1 \rightarrow 2}(t_n) := \left( \int_0^1 \int_0^1 [q_2^{t_n}(2\pi u, 2\pi v) - a_0^{(2)}]^2 du dv \right)^{\frac{1}{2}} \quad (15)$$

where  $a_0^{(i)}$  is the constant term in the Fourier expression for  $q_i^{t_n}$ . In other words, the test statistic  $\epsilon_{j \rightarrow i}(t_n)$  is the *standard deviation over the 2-torus of the inferred function  $q_i^{t_n}$* . It is calculated as the Euclidean norm of the  $4K(K+1)$ -dimensional vector of non-zero-order coefficients in the Fourier expression for  $q_i^{t_n}$ . We refer to this statistic  $\epsilon_{j \rightarrow i}(t_n)$  as the **coupling strength from  $j$  to  $i$**  at time  $t_n$ . We use the method of surrogates to test within each window the significance of the obtained value of  $\epsilon_{2 \rightarrow 1}(t_n)$  and of the obtained value of  $\epsilon_{1 \rightarrow 2}(t_n)$ . The surrogates are taken from the entire duration of the signals before critical values are determined within each window. Consequently, if there are some windows where the  $j$ -to- $i$  coupling strength is found to be significant and other windows where it is not found to be significant, one should not assume that within the latter windows the evolution of  $\phi^i$  is essentially independent of  $\phi^j$ .

The algorithm for dynamical Bayesian inference is described in [1].

## Surrogates

In the Wavelet Phase Coherence, Wavelet Bispectral Analysis and Dynamical Bayesian Inference applications are **methods for analysing possible connections between two signals or between two oscillatory components of one signal**. In general, such connections could signify either mutual interactions or common external influence. For the methods that analyse possible connections, the values quantifying the level of apparent connection are of no use unless one has a sense of what kinds of values to expect when there is no interaction or common influence (since these values will never be exactly 0). Not surprisingly, “what values to expect” are not at all inherent to the analysis tools themselves, but may depend drastically on quantitative and qualitative properties of the physical processes from which the signals are recorded. In view of this, one way to try and determine the significance of the results obtained by these methods is to use the **method of surrogates**. Below, we will describe the method of artificially generated surrogates (as opposed to “intersubject surrogates” where the surrogates are real signals).

### Version I: two signals, one of which is surrogated

Suppose we have a bivariate time-series  $(x_1(t), x_2(t))$  over a time-interval  $I$ . Suppose we have a quantity  $\rho(x_1, x_2)$  measuring the particular type of connection of interest—e.g.  $\rho(x_1, x_2)$  could

be a phase coherence value at some certain frequency—and suppose we wish to have an indication of how significant the value of  $\rho(x_1, x_2)$  is.

To investigate the significance, we generate from the signal  $x_2$  a probability distribution  $P$  of hypothetical signals defined over the same time-interval  $I$ ; the aim of this probability distribution is that a sample signal will share as much in common with  $x_2$  as possible (e.g. it could have the same power-frequency spectrum) within the following constraint: the output  $\tilde{x}$  of a hypothetical physical process modelled probabilistically by  $P$  cannot be considered as physically connected with  $x_1$  under the type of connection(s) indicated by the measure  $\rho$ . Numerically generated sample signals from  $P$  are called **surrogates** of  $x_2$ , and one generates surrogates in order to see how the observed value of  $\rho(x_1, x_2)$  compares with the distribution of values of  $\rho(x_1, \tilde{x})$  for such surrogates  $\tilde{x}$ .

There are many different “types of surrogate”, corresponding to different procedures for generating the probability distribution  $P$  from  $x_2$ . Often, these are mathematically derived from some qualitative approximation or assumption that one could make about the signals  $x_1$  and  $x_2$ , and surrogate results are most meaningful when such an approximation is reasonably accurate for the actual signals  $x_1$  and  $x_2$ , or at least, just the signal  $x_2$ . Of course, it is not possible for  $P$  to be such that sample signals share “everything” in common with  $x_2$  apart from the possible connection with  $x_1$ , and so there is always some **trade-off between randomisation and preservation**. Here, “randomisation” refers to destruction of the possible connection with  $x_1$ , while “preservation” refers to maintaining other features of  $x_2$  such as its power spectrum.

The standard procedure for **testing significance** via surrogates is as follows: First, one chooses the type of surrogate and the “significance level”  $\alpha \in (0, 1)$  that is desired, often either 0.1, 0.05 (most common), 0.02, 0.01 or 0.005; the smaller the value of  $\alpha$ , the stricter the requirement on  $\rho(x_1, x_2)$  to qualify as “significant”. One then chooses a number  $n$  of i.i.d. surrogates to generate from  $x_2$ . Ideally, the number  $n$  should be one less than an integer multiple of  $1/\alpha$ ; for example, if  $\alpha = 0.05$  then  $n$  should come from the set  $\{19, 39, 59, 79, \dots\}$ , or if  $\alpha = 0.01$  or 0.03 then  $n$  should come from the set  $\{99, 199, 299, \dots\}$ . Having generated  $n$  surrogates  $\tilde{x}_1, \dots, \tilde{x}_n$ , one computes the associated values  $\rho(x_1, \tilde{x}_i)$  for  $i = 1, \dots, n$ . Then, assuming that the measure of connection  $\rho$  is defined “positively” rather than “negatively” (i.e. a higher value is meant to correspond to *greater* evidence of connection), one lists the  $n$  values  $\rho(x_1, \tilde{x}_i)$  in descending order  $R_1, R_2, \dots, R_n$  and defines the **critical value** (or **surrogate threshold**)

$$\rho_{\text{crit}} = R_{(n+1)\alpha}. \quad (16)$$

Of course, this formula only makes sense if  $n$  is one less than an integer multiple of  $1/\alpha$ ; if the user chooses  $\alpha$  and  $n$  otherwise, then MODA will determine the critical value as follows

$$\rho_{\text{crit}} = \begin{cases} R_{\lfloor (n+1)\alpha \rfloor} & \alpha > \frac{1}{n+1} \\ R_1 & \alpha < \frac{1}{n+1} \end{cases} \quad (17)$$

where  $\lfloor \cdot \rfloor$  denotes rounding down; in this case, the actual significance level  $\alpha_{\text{true}}$  of the surrogate test is given by

$$\alpha_{\text{true}} = \begin{cases} \frac{\lfloor (n+1)\alpha \rfloor}{n+1} & \alpha > \frac{1}{n+1} \\ \frac{1}{n+1} & \alpha < \frac{1}{n+1}. \end{cases} \quad (18)$$

Having obtained the critical value  $\rho_{\text{crit}}$ , one compares this with the value of  $\rho(x_1, x_2)$ : if  $\rho(x_1, x_2) > \rho_{\text{crit}}$ , then we say that the value of  $\rho(x_1, x_2)$  is *significant* according to the chosen

significance level  $\alpha$ .

Provided  $n$  is one less than an integer multiple of  $1/\alpha$ , the meaning of this “significance” lies in the following fact: *Imagine that  $x_2$  itself came from the probability distribution  $P$ —implying in particular that there is no connection between  $x_1$  and  $x_2$  of the nature under investigation—and that we also draw  $n$  random samples from the probability distribution  $P$  (independently of each other and of  $x_2$ ). Then, prior to the observation of  $x_2$  and the drawing of the  $n$  samples, the probability that  $\rho(x_1, x_2)$  will be found to be significant relative to the  $n$  samples is equal to  $\alpha$ .*<sup>5</sup>

### **Version II: two signals, both of which are surrogated**

Once again, suppose we have a bivariate time-series  $(x_1(t), x_2(t))$  over a time-interval  $I$ . It could either be that  $x_1$  and  $x_2$  were recorded simultaneously from different (but potentially connected) processes, or that  $x_1$  and  $x_2$  are two oscillatory components extracted from the same original recorded signal. And suppose once again that we have a quantity  $\rho(x_1, x_2)$  measuring the particular type of connection of interest, and we wish to have an indication of how significant the value of  $\rho(x_1, x_2)$  is.

This time, one generates from the signal  $x_1$  a probability distribution  $P_1$  of hypothetical signals defined over the time-interval  $I$ , and one generates from the signal  $x_2$  a probability distribution  $P_2$  of hypothetical signals defined over the time-interval  $I$ . The same type of surrogate (i.e. the same procedure for generating a probability distribution from a given signal) is applied for both signals; the aim is that a sample signal from  $P_i$  will share as much in common with  $x_i$  as possible within the following constraint: the outputs  $\tilde{x}$  and  $\tilde{\tilde{x}}$  of two hypothetical physical process modelled probabilistically by  $P_1$  and  $P_2$  respectively, with the two processes being statistically independent, cannot be considered as physically connected under the type of connection(s) indicated by the measure  $\rho$ .

The procedure for testing significance is now as follows: One chooses the type of surrogate and the values of  $\alpha$  and  $n$  just as in Version I. One generates  $n$  i.i.d. surrogates  $\tilde{x}_1, \dots, \tilde{x}_n$  from  $x_1$ , and independently of this, one generates  $n$  i.i.d. surrogates  $\tilde{\tilde{x}}_1, \dots, \tilde{\tilde{x}}_n$  from  $x_2$ . One then computes the quantities  $\rho(\tilde{x}_i, \tilde{\tilde{x}}_i)$  for  $i = 1, \dots, n$ . Once again, assuming that larger  $\rho$ -values are meant to correspond to greater evidence of connection, one lists the  $n$  values  $\rho(x_1, \tilde{x}_i)$  in descending order  $R_1, R_2, \dots, R_n$  and defines the surrogate threshold  $\rho_{\text{crit}}$  just as before. Significance according to the chosen level  $\alpha$  is defined in exactly the same way as in Version I.

Provided  $n$  is one less than an integer multiple of  $1/\alpha$ , the meaning of this “significance” lies in the following fact: *Imagine that  $x_1$  itself came from the probability distribution  $P_1$ , and likewise  $x_2$  from  $P_2$ , with  $x_1$  and  $x_2$  being independent. This implies in particular that there is no connection between  $x_1$  and  $x_2$  of the nature under investigation. Imagine also that we draw  $n$  random signal pairs from the probability distribution  $P_1 \otimes P_2$  (independently of each other and of  $(x_1, x_2)$ ). Then, prior to the observation of  $x_1$  and  $x_2$  and the drawing of the  $n$  sample signal pairs, the probability that  $\rho(x_1, x_2)$  will be found to be significant relative to the  $n$  sample signal pairs is equal to  $\alpha$ .*

---

<sup>5</sup>This statement assumes that  $P$  is a continuous distribution, which is always a reasonably good approximation.

### Version III: one signal

Suppose we have a univariate time-series  $x(t)$  over a time-interval  $I$ . Suppose we have a quantity  $\rho(x)$  measuring a particular type of possible connection between oscillatory components of  $x(t)$  with frequencies close to some prespecified frequency values of interest. We generate from the signal  $x$  a probability distribution  $P$  of hypothetical signals defined over the time-interval  $I$ . The aim is that a sample signal from  $P$  will share as much in common with  $x$  as possible, within the following constraint: the output  $\tilde{x}$  of a hypothetical physical process modelled probabilistically by  $P$  cannot be considered as having oscillatory components that are physically connected under the type of connection(s) indicated by the measure  $\rho$ .

The procedure for testing significance is now as follows: One chooses the type of surrogate and the values of  $\alpha$  and  $n$  just as before. One generates  $n$  i.i.d. surrogates  $\tilde{x}_1, \dots, \tilde{x}_n$  from  $x$ , and computes the quantities  $\rho(\tilde{x}_i)$  for  $i = 1, \dots, n$ . Once again, assuming that larger  $\rho$ -values are meant to correspond to greater evidence of connection, one lists the  $n$  values  $\rho(\tilde{x}_i)$  in descending order  $R_1, R_2, \dots, R_n$  and defines the surrogate threshold  $\rho_{\text{crit}}$  just as before. Significance according to the chosen level  $\alpha$  is defined in exactly the same way as before.

Provided  $n$  is one less than an integer multiple of  $1/\alpha$ , the meaning of this “significance” lies in the following fact: *Imagine that  $x$  itself came from the probability distribution  $P$ . This implies in particular that there is no connection of the nature under investigation between oscillatory components of  $x$ . Imagine also that we draw  $n$  random samples from the probability distribution  $P$  (independently of each other and of  $x$ ). Then, prior to the observation of  $x$  and the drawing of the  $n$  samples, the probability that  $\rho(x)$  will be found to be significant relative to the  $n$  samples is equal to  $\alpha$ .*

### Pre-processing signals prior to surrogation

*This is not to be confused with MODA’s pre-processing prior to time-frequency analysis.*

A finite-time signal  $\{x(t) : 0 \leq t < T\}$  can be extended periodically by  $x_{\text{ext}}(t + nT) = x(t)$  for all  $t$  and  $n$ . Typical surrogate methods (including all the surrogates in MODA except RandPerm and CPP) work best for signals  $x(t)$  whose periodic extension is smooth: this means both that  $x(t)$  is itself smooth and also that the values of  $x(t)$  at the start and end of the signal, as well as their time-derivate, match. Of course, this will not be true of a typically signal, and so **a user wishing to apply surrogates in the Wavelet Phase Coherence or Wavelet Bispectrum Analysis applications may choose to “pre-process” the signals before loading them into MODA.** Pre-processing of a signal to be surrogated will typically consist of first detrending (e.g. subtracting the best linear fit) followed by seeing if a part of the signal at the start and a part at the end can be removed (giving rise to a slightly shorter-length signal) such that the ends better match. Note that in Version I, exactly the same truncation applied to the surrogated signal  $x_2$  must also be applied to the non-surrogated signal  $x_1$ . (For example, if 0.5 seconds at the start and 0.3 seconds at the end of signal  $x_2$  are removed to help make the ends match, then the 0.5 seconds at the start and 0.3 seconds at the end of signal  $x_1$  must also be removed from  $x_1$ .) Likewise, in Version II, the same truncation must be applied to both signals, which may mean that some compromise has to be made between how well the ends of  $x_1$  match and how well the ends of  $x_2$  match. For more detailed suggestions on how to implement pre-processing, see [4].

Nonetheless, for signals of long duration (compared to the main frequencies present in the

signal), provided there is not too strong mismatch between values at the start and end of the signal, pre-processing is likely not to make much of a difference. Of course, for signals of long duration, a trend could cause a substantial mismatch between the start and the end of the signal. In this case, the user is advised at least to remove such a trend before loading the signal into MODA.

Detrending by subtraction of a best-fit linear is straightforward to carry out in MATLAB, using the `detrend` function. For an array  $\mathbf{x}$  of signals stored column-wise (either a matrix or just a single column vector containing one signal), the command `detrend(x)` will output an array of the same dimensions, where each column has had the best-fit linear subtracted, on the assumption of equal spacing between consecutive values in a column. (Note that under this assumption, the procedure of identification and subtraction of a best-fit linear curve is not affected by the sampling frequency.) Similarly, for a single signal stored row-wise as a row vector  $\mathbf{x}$ , the command `detrend(x)` will output a row vector of the same length, where the best-fit linear curve has been subtracted, on the assumption of equal spacing between consecutive values in the row.

### Types of surrogate in MODA

Our above explanation of the surrogate method has not taken into account the discreteness of actual measured signals. Any signal  $x(t)$  in MODA is a *digital* signal, consisting of a finite series of values  $x(0), x(\frac{1}{f_s}), \dots, x(\frac{k-1}{f_s})$  together with a “sampling frequency”  $f_s$  whose reciprocal represents the spacing in time between consecutive values. The types of surrogate implemented in MODA are:

- **RandPerm** [random permutation]: randomises the order of the  $k$  values  $x(0), \dots, x(\frac{k-1}{f_s})$ .
- **FT** [Fourier transform]: randomises phases in the discrete Fourier transform (DFT) of  $(x(0), \dots, x(\frac{k-1}{f_s}))$  while preserving amplitudes.
- **AAFT** [amplitude-adjusted FT]: an adaptation of FT that rearranges the order of the  $k$  values  $x(0), \dots, x(\frac{k-1}{f_s})$  in such a manner as to roughly preserve the DFT power spectrum.
- **IAAFT1/IAAFT2** [iterative AAFT]: an iterative procedure based on the AAFT approach, whose iterates alternate between a rearrangement of  $x(0), \dots, x(\frac{k-1}{f_s})$  that roughly preserves the DFT power spectrum, and an amplitude-preserving reassignment of the phases of the DFT that roughly preserves the histogram of  $x(0), \dots, x(\frac{k-1}{f_s})$ ; **IAAFT1** terminates the procedure so that the result is a rearrangement of  $x(0), \dots, x(\frac{k-1}{f_s})$ , and **IAAFT2** terminates the procedure so that the result is a reassignment of phases in the DFT.
- **WIAAFT** [wavelet IAAFT]: an adaptation of IAAFT2 involving the discrete wavelet transform; this better takes into account possible time-variability in  $x(t)$ , which could cause the previous Fourier-based algorithms to result in an unduly low critical value, but it still preserves the overall DFT power spectrum.
- **tshift** [time-shifted]: a time-shift  $t \mapsto t + \tau$  through a random time  $\tau = \frac{\ell}{f_s}$  is applied to the periodic extension  $x_{\text{ext}}$ , generating a surrogate

$$\left(x_{\text{surr}}(0), \dots, x_{\text{surr}}\left(\frac{k-1}{f_s}\right)\right) = \left(x_{\text{ext}}\left(\frac{\ell}{f_s}\right), \dots, x_{\text{ext}}\left(\frac{k-1+\ell}{f_s}\right)\right);$$

pre-processing is particularly recommended for tshift surrogates if the signal is not very long.

- **CPP** [cyclic phase permutation]: specifically for when  $x$  is a *phase signal*, meaning that for each  $t$ ,  $x(t)$  is an angle between 0 and  $2\pi$ ; the signal is divided into “complete cycles”—where completion of a cycle occurs at a time  $\frac{i}{f_s}$  for which  $x(\frac{i}{f_s})$  is just below  $2\pi$  and  $x(\frac{i+1}{f_s})$  is just above 0—and then the complete cycles are randomly permuted (while the partial cycles at the start and the end stay fixed).

The IAAFT algorithms terminate when the power-spectrum-preserving step yields a signal whose histogram is deemed sufficiently close to that of the original signal; or if this never happens within the first 1000 iterates, then the algorithms terminate at the 1000th/1001th step as appropriate.

In Wavelet Bispectrum Analysis, WIAAFT surrogates are used, and in Dynamical Bayesian Inference, CPP surrogates are used. In Wavelet Phase Coherence, the user can choose any of the surrogates except CPP surrogates which are not applicable.

RandPerm surrogates are designed for testing the hypothesis of being a realisation of uncorrelated noise, and so it is not generally suitable to use RandPerm surrogates to conclude a significant level of connection (e.g. significant phase coherence) between two signals that are not pure noise. In particular, they are not really suitable for concluding significant phase coherence. Nonetheless, they are very quick to compute, and are good at revealing properties of the actual measure of apparent connection itself, e.g. the bias of overall phase coherence towards higher values at lower frequencies, in the absence of any actual connection. In the Wavelet Phase Coherence application, we recommend the following procedure:

- Start by performing computations using RandPerm surrogates.
- If there are no frequency intervals where the overall coherence is significant, then conclude that there is no significant coherence.
- If there are frequency intervals where the overall coherence is significant, then apply one of the other surrogate types to investigate more carefully whether there is significant coherence.

We warn that WIAAFT surrogates are very slow, and IAAFT surrogates are also rather slow.

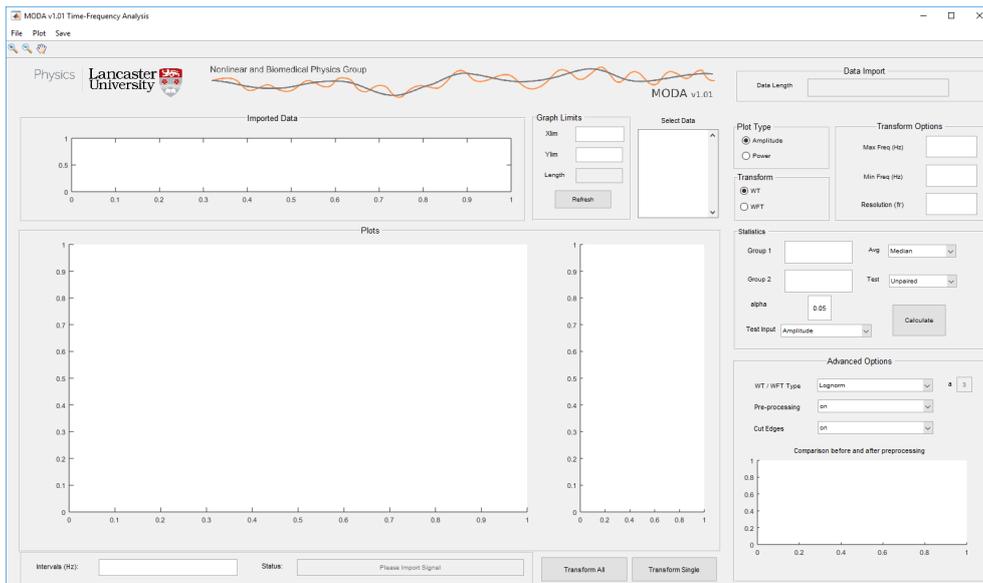
For an in-depth review of surrogate types, their history, their applications, and the precise details of the algorithms implemented to generate each type of surrogate, see [4].

### How many surrogates?

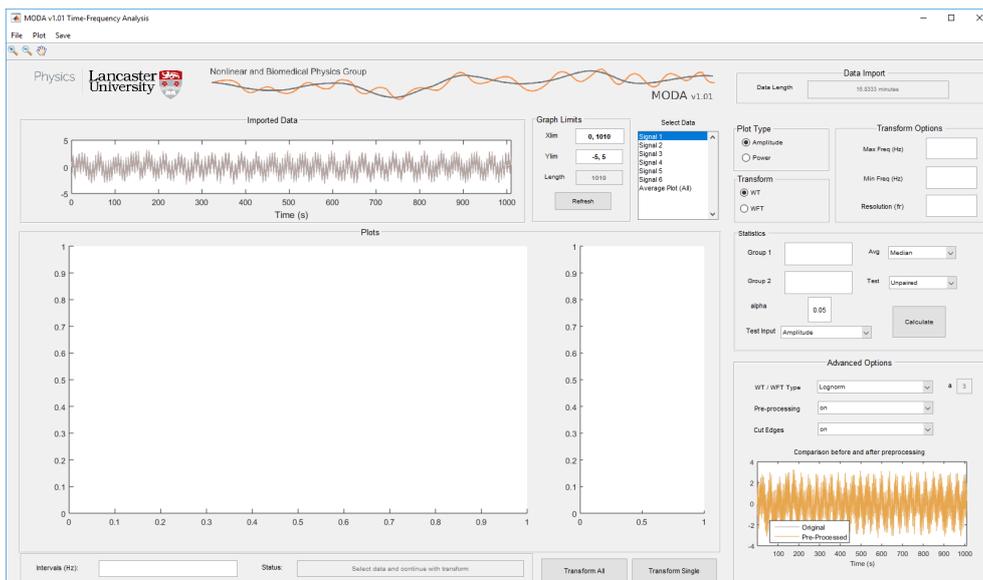
In principle, the greater the number of surrogates  $n$  (for a given significance level) the better. Nonetheless, due to the *law of large numbers*, as the number of surrogates is increased further and further, the surrogate threshold will converge to the  $(1 - \alpha)$ -quantile of the surrogate population. Generally, it is reasonable to deem a number of surrogates  $n$  as sufficient for a meaningful test if increasing the number of surrogates further appears to make little difference to the results.

# Time-Frequency Analysis

If you select “Time-Frequency Analysis”, the window that opens looks as follows:



When you then load a file in which time-series are stored, what the window shows is as follows:



In this window, there is a small box titled “Select Data”, where you should see “Signal 1” selected. You can select any of the signals listed in this box; here, “Signal 1” means the time-series stored

in the 1st column/row (depending on whether you clicked “Column Wise” or “Row Wise”) in the file that you loaded, and likewise “Signal 2” means the time-series stored in the 2nd column/row, and so on. [The option “Average Plot (All)” is not relevant until after all the signals have been analysed using the “Transform All” button.]

Whichever signal is selected, you will see a graph of that signal in the “Imported Data” box in the top-left of the window. If you wish to have the graph of one of these signals plotted in a separate window (as a savable file), select the signal that you want, and then click on “Plot” (to the right of “File”) and click on the option “Time Series”.

If you wish to carry out a time-frequency analysis of all the signals contained in the file that you loaded, you can first select various options from within some of the boxes in the right-most column of the window (*except* the “Statistics” box, which is not relevant at this point), and then click **Transform All**. Alternatively, if you wish only to carry out an analysis of one of the signals, then you can select from the “Select Data” box the signal that you wish to analyse, and then (after selecting the various options as desired) click **Transform Single**. Note that clicking “Transform Single” after results from other signals have been computed will remove all the other results (just like all new computations within a session remove all previously computed results within that same session).

The basic options for the time-frequency analysis that the user can determine are as follows:

- In the box titled “Transform”, the user can decide whether to use a **wavelet transform** abbreviated WT, or a **windowed Fourier transform** abbreviated WFT.
- The user may choose to **restrict the range of frequencies of interest** over which the analysis is carried out, in the box titled “Transform Options”. The maximum frequency (and therefore obviously also the minimum frequency) **must be at most half of the sampling frequency** that you entered when loading the file. If the user leaves the Max Freq field blank, MODA will compute the selected transform for frequencies going up to precisely half the sampling frequency of the signal. **For the WFT, the user must enter a (strictly positive) minimum frequency.**
- Again in the box titled “Transform Options”, the user may wish to specify the frequency resolution parameter  $f_r > 0$  that will be used in the time-frequency analysis, in the field “Resolution (fr)”. If user leaves this field blank, MODA uses a default value of 1 for the parameter  $f_r$ . For the bump wavelet,  $f_r$  must be at least 0.4.

Once the user has chosen the basic options (or left them all at MODA’s default settings), the user can then specify “Advanced Options”:

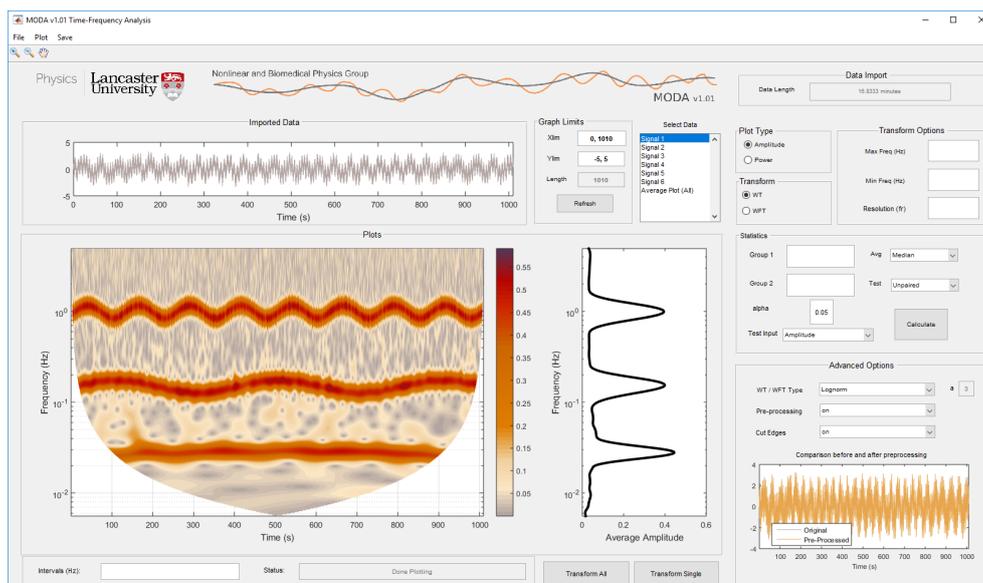
- The user can specify the type of wavelet for the WT (e.g. lognormal), or the type of window for the WFT (e.g. Gaussian). For the WFT, if the “Kaiser” window is chosen, the user can then also specify the  $a$ -parameter for the Kaiser window in the field titled “a”. (Note that for a fixed value of  $f_r$ , as the  $a$ -parameter is varied, the width of the window is accordingly rescaled in order to maintain approximately the same effective frequency resolution.)
- “**Pre-processing**” may be turned on or off: If pre-processing is on, then the selected transform is not applied directly to the original signal. First, the signal is “de-trended”, by the subtraction of a best-fit cubic approximation. This may be useful for trying to identify and analyse oscillatory components of the signal, because non-oscillatory trends in the signal will affect the time-frequency representation. Then, if the user has restricted the

frequency-interval of interest in the “Transform Options” box, MODA will filter out the frequencies outside the desired range, by nullifying the amplitudes associated to frequencies outside the desired range (and its negative) in the Fourier transform of the de-trended signal. Even if the user has not entered values into either or both of the “Min Freq” and “Max Freq” fields, the filtering will still be done using the minimum and maximum frequency for which MODA can perform computations. Either way, the result after this filtering step is the “pre-processed” signal as shown in the plot within the “Advanced Options” box. However, in addition, during the computation of the WT/WFT (as selected), “negative frequencies” are also filtered from the pre-processed signal (meaning that the pre-processed signal is replaced with half its analytic representation). By contrast to all this, if pre-processing is off, then the selected transform is simply applied to the original signal.

- Since time-frequency analysis methods rely on taking some “spread” about each moment in time (as determined by the parameter  $f_r$ ), one cannot meaningfully assign amplitudes and phases to points in time-frequency space that are very close to the start of the signal or very close to the end of the signal. When the option **Cut Edges** is on, MODA will respect this fact by simply not attempting to calculate the selected transform at points where this cannot sensibly be done; in the graphical representation of the transform that MODA produces, those regions in the time-frequency space where the transform is not computed will be left white. However, if the user decides to turn off the Cut Edges option, then MODA will show the results as defined for the “zero-padding extension” of the original signal, where the signal is simply taken to be equal to 0 before the start and after the end of the actual recorded segment.<sup>6</sup>

For whichever signal is currently selected, one can see the comparison between the original signal and the pre-processed signal (prior to negative-frequency filtering) that MODA will use if pre-processing is on; this is seen in the graph below the three options in the “Advanced Options” box.

Practical remarks regarding time-frequency analysis options are given further below. Time-frequency analysis results are displayed graphically in the box titled “Plots”.



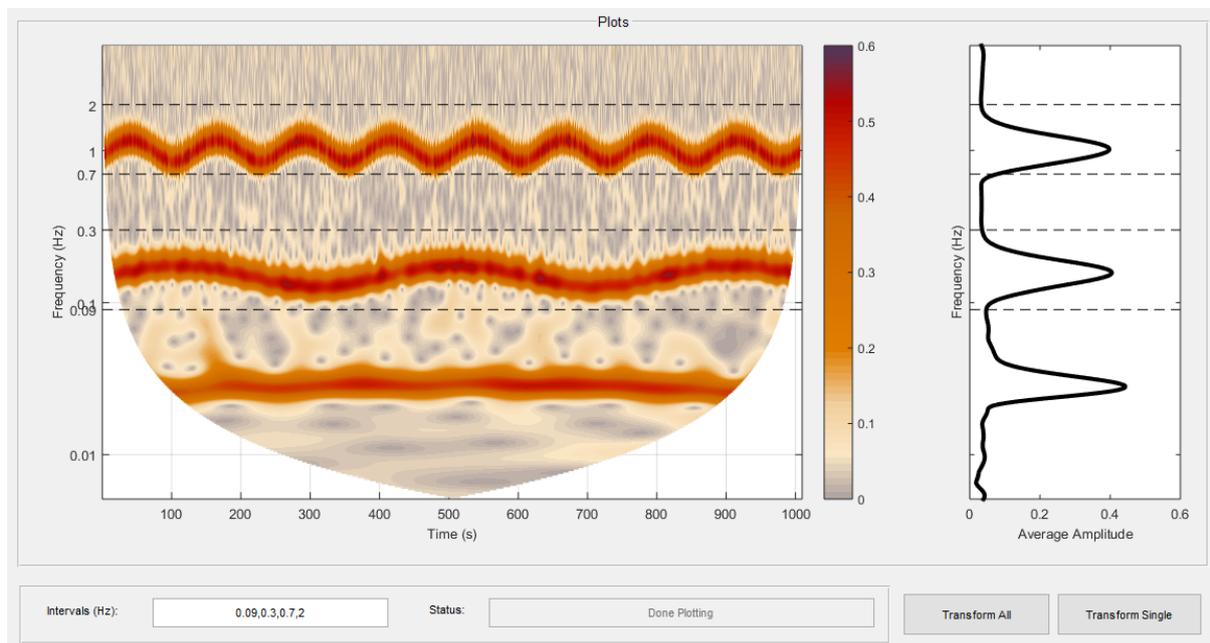
<sup>6</sup>Zero padding is used even when “Cut Edges” is on, but the edges are cut in such a manner that any other sensible form of padding would give essentially the same results.

If you selected Transform All, then you can click on any of the signals in the “Select Data” box to see the results of the analysis for that signal.

Only the results for the amplitudes (or powers) are displayed, but numerical values for the combined amplitude and phase (stored as a single complex number) can be saved for the currently displayed signal, as described in the section “Saving numerical values of results” below. The amplitudes associated to the different frequencies at the different times are represented by a colour-coding within the graph whose axes are Frequency and Time. The graph to the right of this shows the **time-averaged amplitude** associated to each frequency. (Note that in this graph, the independent variable “Frequency” is on the vertical axis and the dependent variable “Average Amplitude” is on the horizontal axis. This is important, because it means that the frequency axes on the two graphs can be visually identified with each other.)

In the box titled **Plot Type**, you can select to **display power rather than amplitude**: the instantaneous power associated to each frequency (as displayed in the left graph via colour-coding) is simply the square of the amplitude; the graph to the right then shows the **time-averaged power** for each frequency. (For each frequency, this is greater than what you would get by taking the square of the time-averaged amplitude.)

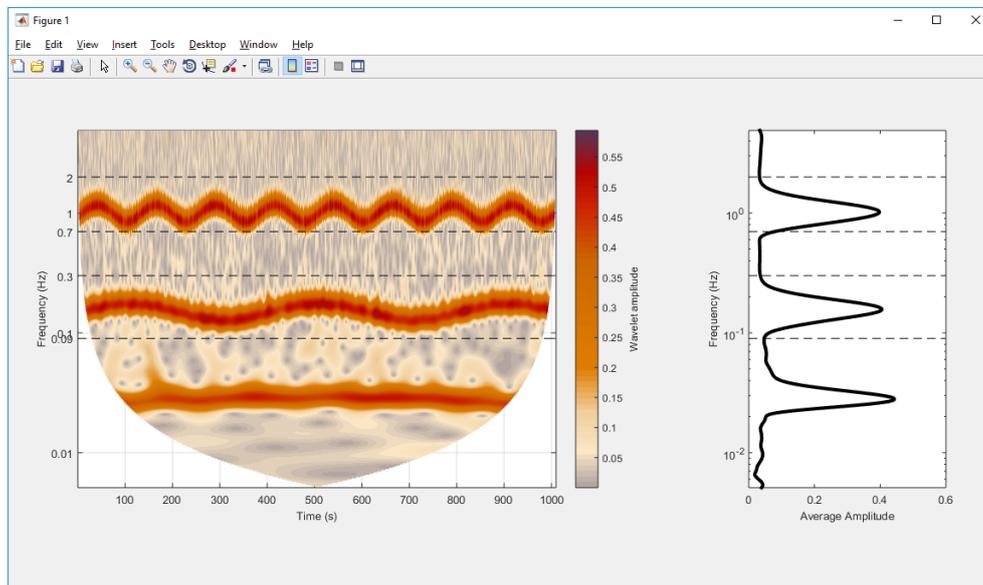
One may wish to **mark certain frequencies on the graphs** (in particular, in order to highlight an interval of frequencies by marking the lower and upper boundary). To do this, type the frequencies that you wish to mark into the “Intervals” field (below the left graph), separating the different frequencies by commas, and then either press the Enter key or click anywhere in the grey space of the window.



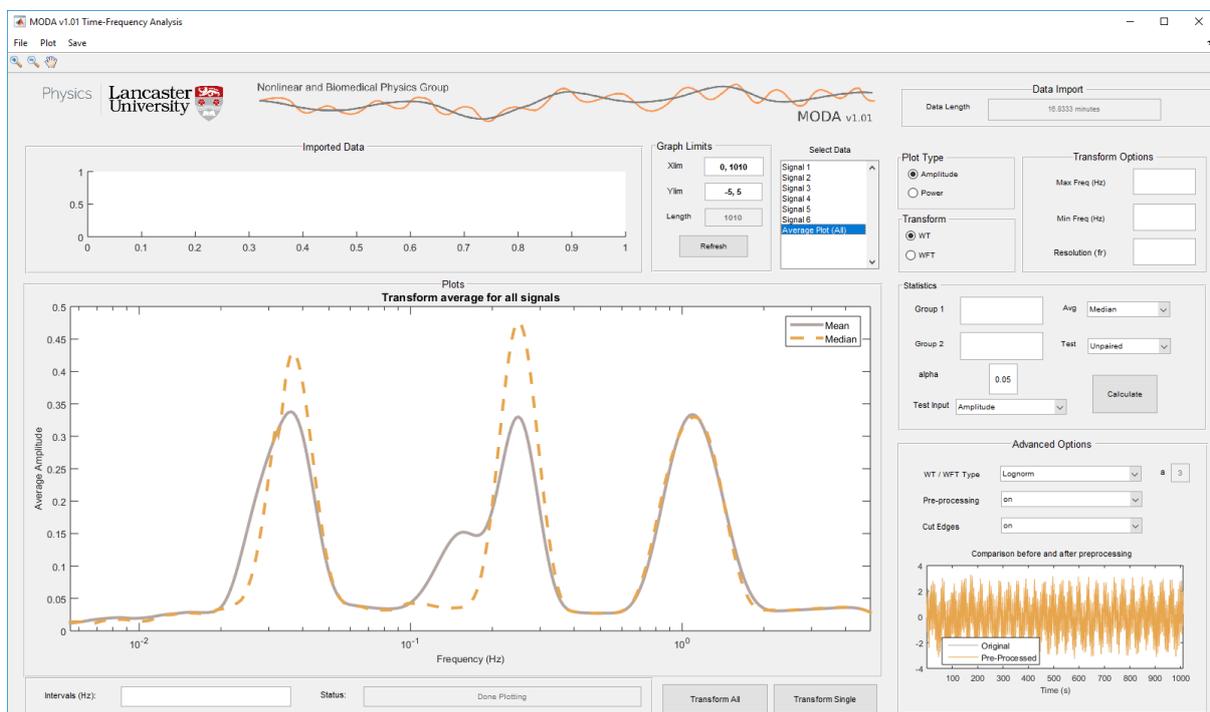
To remove the markers, simply clear the Intervals field and then press Enter or click anywhere outside this field.

Whatever is currently displayed in the left graph (with or without added frequency markers) can be plotted in a separate window as a savable file, by clicking on “Plot” and then “Time-

frequency”. Likewise, what is displayed in the right graph can be plotted as a savable file by clicking on “Plot” and then “Time-average - single”. To plot both as a single savable file, click on “Plot” and then “Time-frequency + average”.

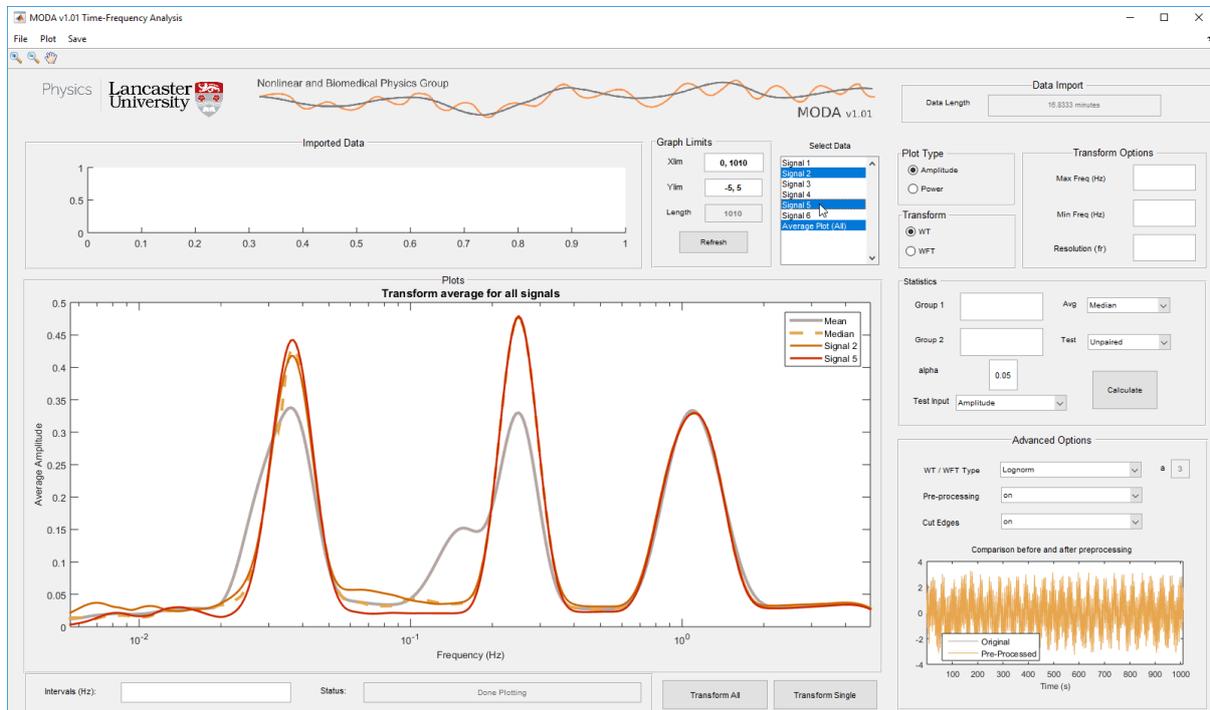


Now if you clicked on Transform All, you can see the **average amplitude-frequency or power-frequency spectrum across the different signals** by clicking on “Average Plot (All)” inside the “Select Data” box.

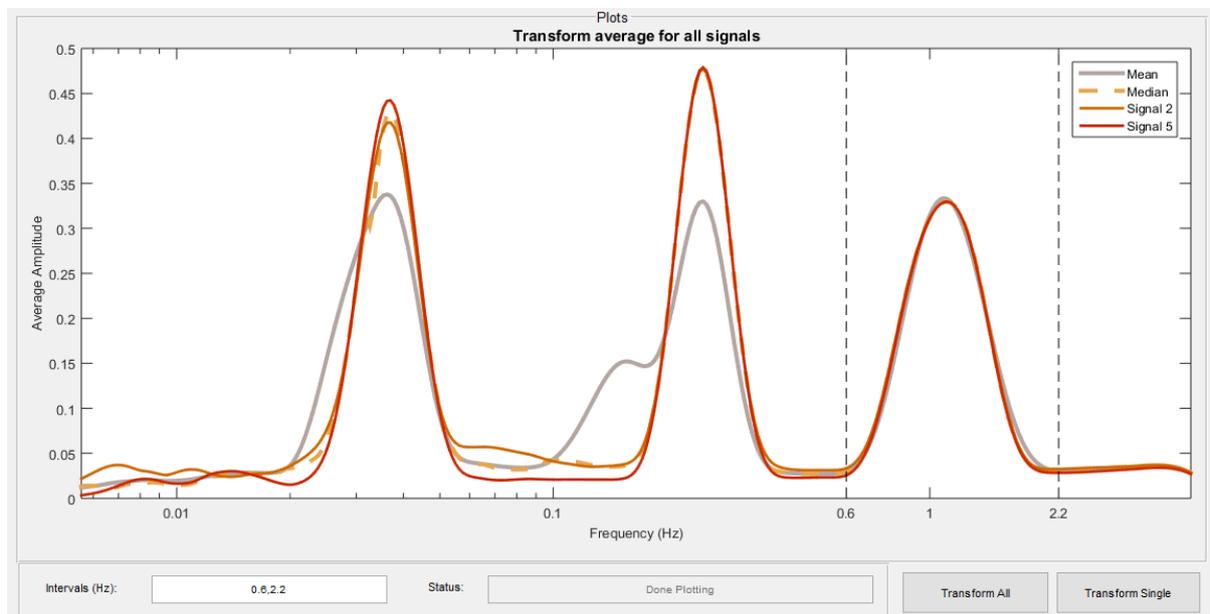


Here, the horizontal axis is Frequency, and the vertical axis is either Average Amplitude or Average Power depending on the option selected in the “Plot Type” box. For each frequency value, both the mean and the median across all the signals of the time-averaged amplitude/power is calculated, and the results are displayed in the graph. Provided there is indeed more than one

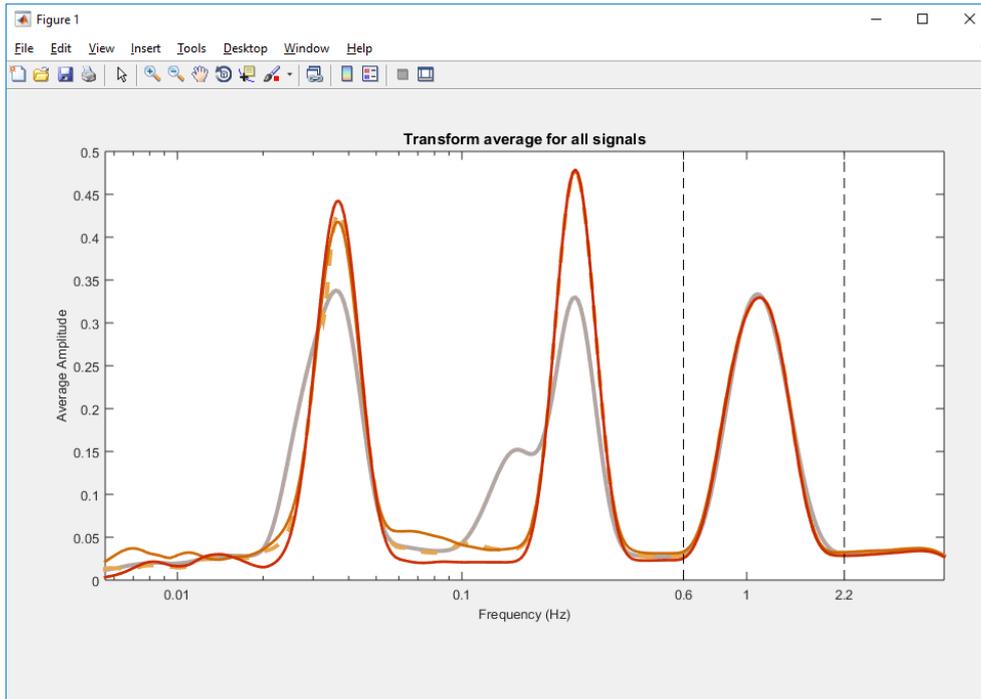
signal in the file that you loaded, it is possible to **add onto this graph the time-averaged amplitude/power of individual signals** as follows: Click “Average Plot (All)” in the “Select Data” box, then press the Ctrl key, and while holding down the Ctrl key, click on the signals in the “Select Data” box that you want to add.



One can also mark frequencies on the graph using the Intervals field.

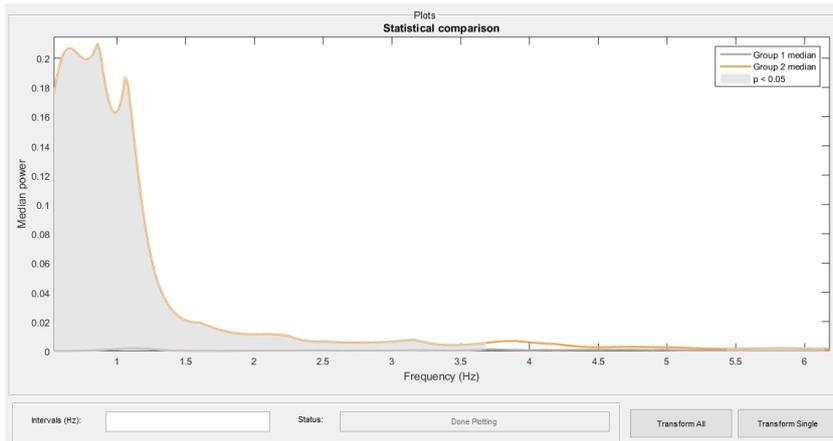


Whatever is currently displayed in the graph may be plotted in a savable file by clicking on “Plot” and then “Time-average - group”.



Finally, using the results of the transforms of all the signals, one can perform the two-sided **rank sum** (“unpaired”) or **signed rank** (“paired”) test as described in the Overview of Time-Frequency Analysis, between signals measuring some quantity  $x$  and signals measuring some quantity  $y$ : In the box titled “Statistics”, enter the signal numbers (as in “Select Data”) of the signals measuring quantity  $x$  in the field “Group 1”, and enter the signal numbers of the signals measuring quantity  $y$  in the field “Group 2”. In each field, these numbers are entered either as a list separated simply by commas, e.g. “1,3,4”, or in the case that the numbers are simply all the numbers between some number  $m$  and some number  $n$ , one can write “ $m:n$ ”. So, for example, entering “5:8” is the same as entering “5,6,7,8”. **For the signed rank test, the number of signals in Group 1 must be the same as the number of signals in Group 2.** In either case, there must be at least two signals in each group. (Hence the file loaded into the Time-Frequency Analysis must itself contain at least four signals.) Enter the significance level  $\alpha$  in the “alpha” field. In the “Test Input” field, select whether you wish the test to be performed on the **time-averaged power** or the **time-averaged amplitude**, by selecting “Power” or “Amplitude” respectively. In the “Test” field, select either “Unpaired” for the rank sum test, or “Paired” for the signed rank test. Then press “Calculate”. The results are shown in the “Plots” box, in a plot that shows either the mean or the median (as selected in the “Avg” box) amplitude/power against frequency; however, **what is chosen in the “Avg” field makes no difference whatsoever to the test carried out.** The frequency regions where the result is significant (i.e. the null hypothesis is rejected) are shown by grey shading between the graphs of the average amplitude/power of the two signals. It is also possible to save the actual  $p$ -values (as computed by MATLAB’s `ranksum/signrank` function) at all the frequencies, as described in the section “Saving numerical values of results”.

The following shows the result in MODA of a rank sum test for real signals (representing heart activity in Group 1 and respiration in Group 2) measured from five people. Of course, this is far too small a sample for a good hypothesis test, but is just included to illustrate MODA’s graphical display of the statistics results:



The frequency values for which there is grey shading are where the evidence in favour of rejecting the null hypothesis is significant (according to the chosen  $\alpha$  value).

Again, one can mark frequencies on this graph using the Intervals field. One can plot the figure as a savable file by clicking on “Plot” and then “Time-average - group”.

One can perform the test again on a different set of parameters and/or groups of signals, by filling in all the fields again, and clicking “Calculate” again. (Once again, this clears previously computed test results; but the test is rather quick to perform in any case.) To go from the results for the statistical tests back to the actual transforms, click on any of the items in the “Select Data” box.

## Saving numerical values of results

Suppose the file loaded into MODA for analysis consists of  $N$  signals.

After computing results, if you click “Save” and then “Save all as .mat”, the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled TFR\_data.

Field	Value
Preprocessed_data	6x1 cell
Analysis_type	'Wavelet'
Wavelet_type	'Morlet'
Amplitude	308x6 double
Frequency	308x1 double
Time	1x10100 double
Sampling_frequen...	10
fmax	5
fmin	0.0051
fr	1
Preprocessing	'on'
Cut_Edges	'on'
group1_index	[1,2]
group2_index	[3,4,5]
group1	2x308 double
group2	3x308 double
p_value	1x308 double
testtype	'Unpaired'
testinput	'Power'

The fields in this structure are as follows:

- “Time”: the complete discretised time axis (in seconds), as a row vector;
- “Frequency”: the complete discretised frequency axis (in Hz), as a column vector;
- [only if pre-processing was on] “Preprocessed\_data”: a single-column cell with  $N$  entries; the  $j$ -th entry is a row vector consisting of the series of values in the pre-processed version of the  $j$ -th signal; the pre-processed versions share the same time axis as the original signals;
- “Analysis\_Type”: Wavelet or Windowed Fourier, as selected;
- “Wavelet\_Type” or “Window\_Type”; for the Kaiser window, the  $a$ -parameter is also included, e.g. “kaiser-3” means the Kaiser window with  $a = 3$ ;
- “Amplitude” or “Power” (as selected under “Plot Type”): an array of  $N$  columns if “Transform All” was chosen, or just 1 column if “Transform Single” was chosen; if “Transform All” was chosen, the  $(i, j)$ -entry is the time-averaged amplitude/power at the  $i$ -th frequency value in the discretised frequency axis, for the  $j$ -th signal; similarly, if “Transform Single” was chosen, the  $i$ -th entry is the time-averaged amplitude/power at the  $i$ -th frequency value in the discretised frequency axis, for the single transformed signal;
- “Sampling\_frequency”: the number entered for the sampling frequency when the file was being loaded;
- “fmax”: what was entered into the “Max Freq” field, or the Nyquist frequency (in Hz) if this field was left blank;
- “fmin”: what was entered into the “Min Freq” field, or the smallest frequency for which computation was possible (in Hz) if this field was left blank;
- “fr”: what was entered into the “Resolution (fr)” field, or the default value 1 if this field was left blank;
- “Preprocessing”: shows whether pre-processing was chosen to be on or off;
- “Cut\_Edges”: shows whether Cut Edges was chosen to be on or off;

and if a statistical test was carried out, then included are also:

- “group1\_index”: shows the signal number of each of the signals included in group 1 (as entered in the “Group 1” field); these numbers are stored as a row vector;
- “group2\_index”: the same, for group 2;
- “test\_input”: either Amplitude or Power, as selected in the “Test Input” field;
- “group\_1”: an array whose rows are the time-averaged amplitude/power values, as selected in “Test Input”, for the signals in group 1; the  $(i, j)$ -entry is equal to the time-averaged amplitude/power at the  $j$ -th frequency value in the discretised frequency axis, for the signal whose number is the  $i$ -th entry in group\_index;<sup>7</sup>
- “group\_2”: analogously, for group 2;

---

<sup>7</sup>If “Plot Type” and “Test Input” are the same, then all the rows in group\_1 and group\_2 are already present as columns in Amplitude/Power. Nonetheless, it is very useful to have the results in each of the groups stored as a single array if one seeks to carry out further analysis (or graphical representation), especially in software such as MATLAB.

- “testtype”: either Paired (signed rank) or Unpaired (rank sum), as selected in the “Test” field;
- “p\_value”: a row vector whose  $i$ -th entry is the  $p$ -value for the hypothesis test at the  $i$ -th frequency value in the discretised frequency axis.

If you click “Save” and then “Save all as .csv”, then MODA will save either one or two .csv files, depending on whether pre-processing was on or off. If pre-processing was on, and you save with a file name, say, “Results”, then in the folder where you selected to save the results, MODA will save both a file Results\_TFdata.csv and a file Results\_ppdata.csv. The file Results\_TFdata.csv includes the following:

- the date on which the results were saved;
- the analysis type, wavelet/window type, sampling frequency, maximum and minimum frequencies, frequency resolution parameter  $f_r$ , and whether pre-processing and Cut Edges are on or off, all just as when saved as .mat;
- the start time and end time of the signal (in seconds);
- the list of frequency values (in Hz) on the discretised frequency axis, together with the time-averaged amplitude/power at each frequency for all  $N$  signals if “Transform All” was chosen, or just the one analysed signal if “Transform Single” was chosen, e.g. as follows:

17							
18							
19	Frequency	Power 1	Power 2	Power 3	Power 4	Power 5	Power 6
20	0.30573	0.00073202	0.00084206	0.00064687	0.00026872	0.00042165	0.00044627
21	0.31264	0.00082871	0.00097606	0.00075197	0.00030986	0.00045895	0.00050396
22	0.31971	0.00092051	0.0011028	0.00085659	0.00035114	0.00048945	0.00055607
23	0.32694	0.001006	0.001217	0.00095879	0.00039191	0.00051369	0.00060137
24	0.33433	0.0010844	0.001314	0.0010563	0.00043148	0.00053216	0.00063891
25	0.34189	0.0011544	0.001391	0.0011481	0.00046922	0.00054763	0.00066985
26	0.34962	0.0012155	0.0014461	0.0012317	0.0005045	0.00056089	0.00069439
27	0.35753	0.0012672	0.0014796	0.0013048	0.00053676	0.00057278	0.00071289
28	0.36561	0.0013094	0.0014929	0.0013658	0.00056559	0.00058509	0.00072718
29	0.37388	0.0013424	0.001489	0.001413	0.00059071	0.00059783	0.00073737
30	0.38233	0.0013667	0.0014717	0.0014455	0.00061194	0.00061089	0.00074338
31	0.39098	0.0013836	0.0014451	0.0014632	0.00062948	0.00062442	0.00074612
32	0.39982	0.0013945	0.0014137	0.0014672	0.00064344	0.00063765	0.0007449
33	0.40886	0.0014009	0.001381	0.0014592	0.00065431	0.00065018	0.00074013
34	0.4181	0.0014046	0.0013498	0.0014416	0.00066271	0.00066152	0.00073215
35	0.42756	0.0014065	0.0013226	0.0014173	0.00066895	0.00067141	0.00072114
36	0.43723	0.001408	0.0013001	0.001389	0.00067369	0.00067983	0.00070804
37	0.44711	0.001409	0.0012831	0.0013593	0.00067715	0.000687	0.00069374
38	0.45722	0.0014097	0.0012711	0.0013307	0.00067966	0.00069317	0.00067938
39	0.46756	0.0014099	0.0012635	0.0013052	0.00068135	0.00069855	0.00066603
40	0.47813	0.0014092	0.0012594	0.0012841	0.00068217	0.00070326	0.00065451
41	0.48894	0.0014068	0.001258	0.001269	0.00068203	0.00070693	0.00064573
42	0.5	0.0014027	0.0012581	0.0012603	0.00068077	0.00070917	0.00063998
43	0.51131	0.0013968	0.001259	0.0012579	0.0006782	0.00070946	0.00063727
44	0.52287	0.0013897	0.00126	0.0012611	0.00067421	0.00070734	0.00063733
45	0.53469	0.0013818	0.0012606	0.0012687	0.00066881	0.00070262	0.00063964
46	0.54678	0.0013739	0.0012605	0.0012787	0.00066623	0.00069554	0.00064353
47	0.55914	0.0013671	0.0012599	0.0012893	0.00065562	0.00068682	0.00064833
48	0.57179	0.0013634	0.0012591	0.0012985	0.00065127	0.00067788	0.00065343
49	0.58472	0.0013683	0.0012599	0.0013047	0.0006555	0.00067139	0.00065835
50	0.59794	0.0013948	0.0012668	0.0013067	0.00068316	0.0006728	0.00066279
51	0.61146	0.0014737	0.0012912	0.0013041	0.00076674	0.00069402	0.0006666
52	0.62528	0.0016676	0.0013589	0.0012971	0.00097095	0.00076043	0.00066979
53	0.63947	0.0029905	0.0015214	0.0012863	0.001412	0.00092242	0.00067253

and if a statistical test was carried out, then included are also:

- the signal numbers in group 1 and in group 2;
- the test input and test type, as above;
- the  $p$ -values at all the frequencies in the frequency discretisation, as an extra column to the right of the time-averaged amplitude/power values.

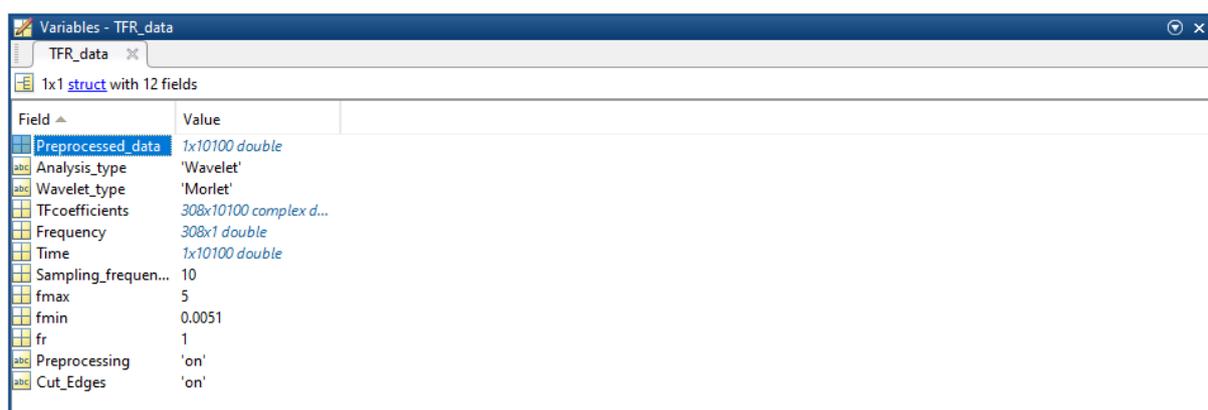
The file Results\_ppdata.csv consists of the list of times (in seconds) on the discretised time axis, together with the pre-processed versions of the  $N$  signals; this takes the same format as the amplitude-frequency or power-frequency spectrum contained in Results\_TFdata.csv.

If pre-processing was off, then clicking “Save all as .csv” will only save the \_TFdata.csv file, and not the \_ppdata.csv file.

**Note that whenever “Save all as ...” is chosen, the actual values of the WT/WFT over time-frequency space are not saved.** This is because the set of values is very large, and so saving them for all the signals could create an especially large file. To include the values of the WT/WFT in what is saved, you can only save results for one of the signals at a time: you must select the signal from the “Select Data” box, and then click “Save” and “**Save current as .mat**”.

If you select one of the signals and choose “Save current as .mat”, then the .mat file saved is just like the file saved by selected “Save all as .mat”, except that:

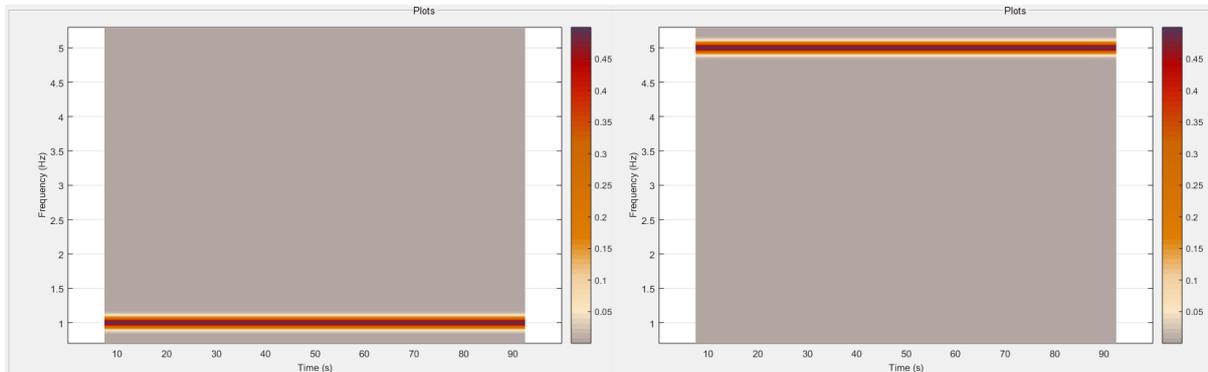
- if pre-processing is on then “Preprocessed\_Data” only has the pre-processed version of the selected signal;
- instead of the array “Amplitude” or “Power”, there is an array “**TFcoefficients**”: the  $(i, j)$ -entry is the value assigned by the WT/WFT (as selected) to the point  $(t_j, f_i)$  where  $f_i$  is the  $i$ -th value in the discretised frequency axis and  $t_j$  is the  $j$ -th value in the discretised time axis; the number assigned is a **complex number whose modulus is the amplitude and whose argument is the phase**.



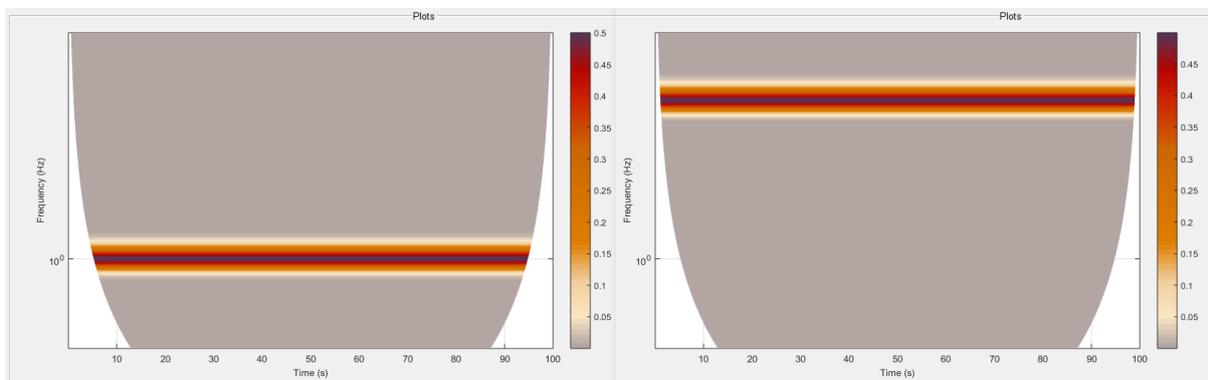
If Cut Edges is on, then in “TFcoefficients”, NaN is assigned to points  $(t, f)$  outside the region where results are shown in the time-frequency analysis plot.

## Practical remarks

The wavelet transform is advised over the windowed Fourier transform, because of how the WT adjusts the window-width to each frequency under investigation, allowing for suitably high frequency resolution at the higher frequencies. The WFT has **linear frequency resolution**, in the sense that two sine waves of equal amplitude and different frequency will be represented equally on a linear frequency axis (except obviously at different heights):



The left image is for a sinusoid of frequency 1 Hz, and the right image is for a sinusoid of frequency 5 Hz. The time-averaged WFT power over a given frequency band  $[a, b]$  can be defined as the area in the Average Power plot between the graph and the frequency axis within the frequency band  $[a, b]$ . The WT (according to how it is defined in MODA) has **logarithmic frequency resolution**, in the sense that two sine waves of equal amplitude and different frequency will be represented equally on a logarithmic frequency axis (except again at different heights):



Again, the left image is for a sinusoid of frequency 1 Hz, and the right image is for a sinusoid of frequency 5 Hz. The time-averaged WT power over a given frequency band  $[a, b]$  can be defined as the area in the Average Power plot, *with* the logarithmic scale on the frequency axis, between the graph and the frequency axis within the frequency band  $[a, b]$ .

The logarithmic axis and associated logarithmic resolution for the WT enable clear visualisation of distinct constituent components that oscillate on very different timescales; such multiscale analysis would generally be difficult with the WFT.

The optimal value of  $f_r$  depends very much on the time-series itself: On the one hand, as a general principle, faster modulations of frequency require a smaller value of  $f_r$  to be accurately represented. On the other hand, if the frequencies of two oscillatory components are quite

close to each other, then being able to resolve these two components accurately requires a sufficiently high value of  $f_r$ . The user is generally advised not to take  $f_r$  smaller than about 0.5 because (i) a window needs to be wide enough to incorporate several complete oscillations of an oscillatory process in order to give a meaningful estimate of time-localised frequency, and (ii) if the pre-processing option is on, then the slight blurring of temporal changes that is inherent to pre-processing becomes more pronounced when  $f_r$  is small. (See more below.)

The default window for the WFT is the Gaussian window, because of its optimisation of the Heisenberg measure of the trade-off between time localisation and frequency resolution. The default wavelet type for the WT is the lognormal wavelet (“Lognorm”). This has the following advantages over the more well-known Morlet wavelet:

- For a sinusoidal signal, the spread of amplitude/power around the peak is symmetric (in fact Gaussian) on a logarithmic frequency axis, respecting the logarithmic frequency resolution of the wavelet transform.
- Changing the frequency resolution parameter  $f_r$  preserves the shape of the lognormal wavelet on a logarithmic frequency axis, whereas for the Morlet wavelet, while  $f_r$  is fairly small, changing the value of  $f_r$  does not preserve the shape of the Morlet wavelet on *any* scaling of the frequency axis.
- For fairly small values of  $f_r$ , the Morlet wavelet is subject to interference from negative frequencies, which can even corrupt a pure sinusoidal signal if  $f_r$  is too small. (Simply cutting off negative frequencies, as happens when pre-processing is on, causes the Morlet wavelet to have poorer time localisation, which—ironically—gets *worse* as  $f_r$  is decreased.) By contrast, the lognormal wavelet is not subject to interference from negative frequencies.
- Investigation in [2] found the resolution in time-frequency space (with logarithmic frequency) of the lognormal wavelet to be better than for the Morlet wavelet.

Nonetheless, for  $f_r$  larger than around 1, the lognormal and Morlet wavelets give very similar results to each other. (As  $f_r \rightarrow \infty$ , the two wavelets tend towards being identical.)

The user is **advised always to have “Cut Edges” on**. This is because results too close to the start or the end of the signal have no physical meaning, being significantly affected by what is artificially added before the start and after the end of the signal in order for the computations to be possible. (In MODA, “zero padding” is used, meaning that the signals are simply assumed to be constant at 0 before and after the actual recorded segment.<sup>8</sup>)

In **pre-processing**, the purpose of detrending (by subtracting a best-fit cubic approximation) is to remove the effects of such trends on the time-frequency representation when trying to identify and study oscillatory components; the issue is that the trends themselves are represented as extremely low frequency oscillations, and due to limited frequency resolution, these may interfere with other low frequency oscillatory components of genuine interest. Likewise, the purpose of filtering within the frequency range of interest is to avoid interference from oscillations outside the range of interest that occurs due to limited frequency resolution: filtering effectively works with a “long” time-window that enables high enough frequency resolution to distinguish and separate out the different oscillations more cleanly. Thus, the subsequent more time-localised

---

<sup>8</sup>If pre-processing is on, then the detrending and filtering (*excluding* the removal of negative frequencies) are performed on the original signal, and then the zero-padding is added; then, re-filtering of the padded signal within the same frequency range of interest is performed, but now *including* the removal of negative frequencies.

analysis via the WT/WFT is able to represent time-evolving frequency with less interference from other oscillations. Similarly, in the WFT where the frequency resolution is linear, low-frequency oscillations can experience interference from their “negative-frequency” counterparts. Interference from negative frequencies can also occur in the WT with a Morlet wavelet; on the one hand, such interference will be very weak if  $f_r$  is not too small, but on the other hand, it is able to affect all frequencies equally (unlike in the WFT where it will only measurably affect lower frequencies). Therefore, pre-processing in both the WFT and WT also involves filtering of negative frequencies. However, for the lognormal and bump wavelets, interference from negative frequencies does not occur and filtering of negative frequencies makes no difference to the result.

Having said all this, there are potential dangers that come with having pre-processing on. All the above reasons for pre-processing are essentially concerned with problems about limited frequency resolution; but any measure taken to overcome problems of frequency resolution will introduce a compromise in time localisation. The value of the best-fit cubic at one moment in time will be affected by how the entire signal behaves. Slowly frequency-modulated signals will have harmonics in the overall Fourier transform that are not visible in a time-frequency representation, and cutting these off through filtering may distort the signal, even if the instantaneous frequency remains at all times within the passband. Moreover, the filters involved are “brick-wall filters”, which on the one hand provide the most definite boundary between two oscillatory components whose frequencies are in danger of interference in a time-frequency representation, but on the other hand have very poor time localisation. So when using such filters, it is important that there are no sharp transitions in the signal, and that there are no portions of the signal during which an oscillation occupies a frequency range that includes the boundaries of the passband. A sharp transition with no early warning sign will produce false evidence of an early warning sign in the time-frequency representation when filtering is used (even just filtering of negative frequencies in the WFT or Morlet WT), and paradoxically, this problem is made *worse* by increasing the time resolution of the transform (i.e. by decreasing  $f_r$ ).

For further discussion on pre-processing as implemented in MODA, together with some illustrative examples, see [2]. Alternatively, there is a simple form of pre-processing that the user may wish to carry out before loading a signal  $\{x(t) : 0 \leq t \leq T\}$  into MODA:

- Decide a minimum frequency of interest  $f_{\min}$ .
- Construct a signal  $y(t)$  on the time-interval  $[5/f_{\min}, T - 5/f_{\min}]$ , where at each time  $t$ ,  $y(t)$  is the average of the  $x$ -values over the time-interval  $[t - 5/f_{\min}, t + 5/f_{\min}]$ . (This is called a **moving average**.)
- Construct the pre-processed signal  $\tilde{x}(t)$  on the time-interval  $[0, T - 10/f_{\min}]$ , defined by  $\tilde{x}(t) = x(t + 5/f_{\min}) - y(t + 5/f_{\min})$ .

This will remove slow trends from the signal, while periodic oscillations of frequency  $f > f_{\min}$  will be very little affected. (Of course, this is all written as though  $x(t)$  were a continuous signal; some suitable discretisation for the moving average would need to be chosen.)

## Formulae

The wavelet transform and windowed Fourier transform are theoretically defined for signals  $x(t)$  that do not have a start time or an end time. When dealing with an actual finite-time signal on a time-interval  $[0, T]$ , one needs to define some “padding” for the signal, that is, a way to

extend the signal beyond its start and end times; at each frequency  $f$ , results are only taken as meaningful over a restricted time-interval  $[\epsilon_f, T - \epsilon_f]$  in which the chosen type of padding makes no significant difference to the results. For the WFT,  $\epsilon_f$  does not actually depend on  $f$ , since the window does not depend on  $f$ . For the wavelet transform,  $\epsilon_f$  is larger for lower frequencies  $f$ , due to the wider windows at lower frequencies. The region  $\{(t, f) : t \in [\epsilon_f, T - \epsilon_f]\}$  in time-frequency space where results are considered meaningful is called the *cone of influence*; the “Cut Edges” option determines whether to show only the results within this region.

In both transforms, a complex number is assigned to each point in time-frequency space, whose modulus is the instantaneous amplitude and whose argument is the instantaneous phase.

As mentioned in the Overview of Time-Frequency Analysis, an individual amplitude by itself has no meaning, but only the relative amplitudes across time-frequency space, or across different signals at the same point in time-frequency space, are meaningful. Accordingly, for the WFT, there is no point in normalising the windows (except the Gaussian where it simplifies the computation); technically, this means that for the windows other than the Gaussian, the formula for the WFT outputs values whose units are time units. Nonetheless, again since only relative amplitudes are meaningful, one may ignore units of amplitude or power; but values displayed and saved are as they would be in SI units.

## Windowed Fourier transform

In the WFT, we start with a “window function”  $g(t)$ , which is a nonnegative-valued function whose peak value occurs at  $t = 0$  s, and which has the property that  $g(t) = g(-t)$  for all  $t$ . For a signal  $X(t)$ , the windowed Fourier transform  $G_X$  is then defined by

$$G_X(t, f) = \int_{-\infty}^{\infty} X(u)g(u-t)e^{-2\pi if(u-t)} du.$$

A more commonly found definition simply has  $e^{-2\pi ifu}$  in place of  $e^{-2\pi if(u-t)}$ , but then the phase assigned to a point  $(t, f)$  would not represent the phase at time  $t$ , but rather the extrapolated phase at time 0. As in [2], MODA uses the above formula so that the phase assigned to  $(t, f)$  really is the phase at time  $t$ ; this is necessary for ridge-extraction to be possible.

There are six types of window functions in MODA, each specifying the function  $g(t)$  up to linear scaling (except that the Kaiser window also depends on the “shape” parameter  $a$ ); the scaling is then determined by the quantity  $f_0 := f_r/f_{\min}$ , where  $f_r$  is the frequency resolution parameter, and  $f_{\min}$  is the minimum frequency of interest, as specified in the “Min Freq” field. The types of window are as follows:

- Gaussian:  $g(t) = \frac{1}{\sqrt{2\pi}f_0} e^{-\frac{t^2}{2f_0^2}}$
- Hann: writing  $q := 4.4f_0$ ,  $g(t) = \mathbf{1}_{[-\frac{q}{2}, \frac{q}{2}]}(t) \cos^2(\frac{\pi t}{q})$
- Blackman: writing  $q = 5.6f_0$ ,  $g(t) = \mathbf{1}_{[-\frac{q}{2}, \frac{q}{2}]}(t) (0.42 + 0.5 \cos(\frac{2\pi t}{q}) - 0.08 \cos(\frac{4\pi t}{q}))$
- Exponential (“Exp”): writing  $q = 6.5f_0$ ,  $g(t) = e^{-|\frac{t}{q}|}$
- Rectangular (“Rect”): writing  $q = 10f_0$ ,  $g(t) = \mathbf{1}_{[-\frac{q}{2}, \frac{q}{2}]}(t)$

- Kaiser- $a$  (for  $a > 0$ ): writing  $q = 3\sqrt{1 + \left|\frac{a^2-1}{a}\right|}f_0$ ,

$$g(t) = \begin{cases} \frac{I_0(\pi a \sqrt{1 - (\frac{2t}{q})^2})}{I_0(\pi a)} & |t| \leq \frac{q}{2} \\ 0 & |t| \geq \frac{q}{2} \end{cases}$$

where

$$I_0(x) = \sum_{k=0}^{\infty} \frac{x^{2k}}{4^k (k!)^2}.$$

In the case of the Kaiser window,  $q$  is chosen according to [2] so as to maintain roughly the same effective frequency resolution across different  $a$ -values. However, for very large or very small  $a$ -values, this may not hold. MODA's default  $a$ -value is 3.

### Wavelet transform

In the WT, we start with a function  $\hat{\psi}: \mathbb{R} \rightarrow \mathbb{R}$  and a number  $\omega_\psi > 0$ , where (in MODA)  $\hat{\psi}(x)$  attains its maximum positive value uniquely at  $x = \omega_\psi$ , and is meant to fall away towards 0 fairly quickly as  $x$  moves away from  $\omega_\psi$ . (The speed at which  $\hat{\psi}(x)$  decays depends on the frequency resolution parameter  $f_r$ , with a larger value meaning more rapid decay.)

Given  $\hat{\psi}$ , we define the function  $\psi: \mathbb{R} \rightarrow \mathbb{C}$  to be the inverse Fourier transform of  $\hat{\psi}$ , according to the convention

$$\psi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\psi}(y) e^{ixy} dy.$$

The function  $\psi$  is called the *mother wavelet*.

For a signal  $X(t)$ , the wavelet transform  $W_X$  is given by

$$W_X(t, f) = \frac{2\pi f}{\omega_\psi} \int_{-\infty}^{\infty} X(u) \overline{\psi\left(\frac{2\pi f(u-t)}{\omega_\psi}\right)} du$$

where  $\overline{\phantom{x}}$  denotes the complex conjugate. One often finds the formula for the wavelet transform expressed not with reference to frequency  $f$  but with reference to the “scale”  $s = \frac{\omega_\psi}{2\pi f}$ .

It is very common for a definition of the wavelet transform to use the pre-factor of  $\left(\frac{2\pi f}{\omega_\psi}\right)^{\frac{1}{2}}$  rather than  $\frac{2\pi f}{\omega_\psi}$ ; this then has the property that the wavelet power over a frequency interval (at any time  $t$ ) is equal to the integral of the square of the amplitude of the wavelet transform over that interval. However, the convention in MODA has the following significant advantages:

- Equal-amplitude oscillatory components of a signal with different frequencies will appear the same (except concentrated around their respective frequencies) in the wavelet transform plot, assuming a logarithmic frequency scale. In particular, the “frequency blur” around the two oscillations will have the same width, and the wavelet amplitudes will be the same.
- The wavelet power over a frequency interval (at any time  $t$ ) can be obtained as the integral of the square of the wavelet amplitude over a logarithmic frequency scale.
- For the lognormal and bump wavelets (which are *analytic*, meaning that  $\hat{\psi}(x) = 0$  for all  $x < 0$ ), given a sinusoidal signal  $X(t) = \cos(2\pi\nu t + \phi)$ , at every time  $t$  the frequency  $f$  where the amplitude  $|W_X(t, f)|$  is maximised is precisely  $f = \nu$ . The same also holds for the

Morlet wavelet if pre-processing is on (since the filtering of negative frequencies effectively turns all wavelets into analytic wavelets); and even with pre-processing off, provided  $f_r$  is larger than about 0.5, the offset is immeasurable.

The three types of wavelet transform in MODA are:

- Lognormal (“Lognorm”):

$$\hat{\psi}(x) = \begin{cases} e^{-\frac{(2\pi f_r \log x)^2}{2}} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

- Morlet:  $\hat{\psi}(x) = e^{-\frac{(x-2\pi f_r)^2}{2}} (1 - e^{-2\pi f_r x})$

- Bump (requires  $f_r \geq 0.4$ ): writing  $\Delta = \frac{0.4}{f_r}$ ,

$$\hat{\psi}(x) = \begin{cases} e^{-\frac{1}{1-(\frac{x-1}{\Delta})^2}} & x \in (1-\Delta, 1+\Delta) \\ 0 & \text{otherwise} \end{cases}$$

For the lognormal and bump wavelets,  $\omega_\psi = 1$ .

For the Morlet wavelet, provided  $f_r$  is not too small (larger than about 0.5),  $\omega_\psi$  is almost exactly equal to  $2\pi f_r$  and  $\psi(x)$  can be approximated very well by

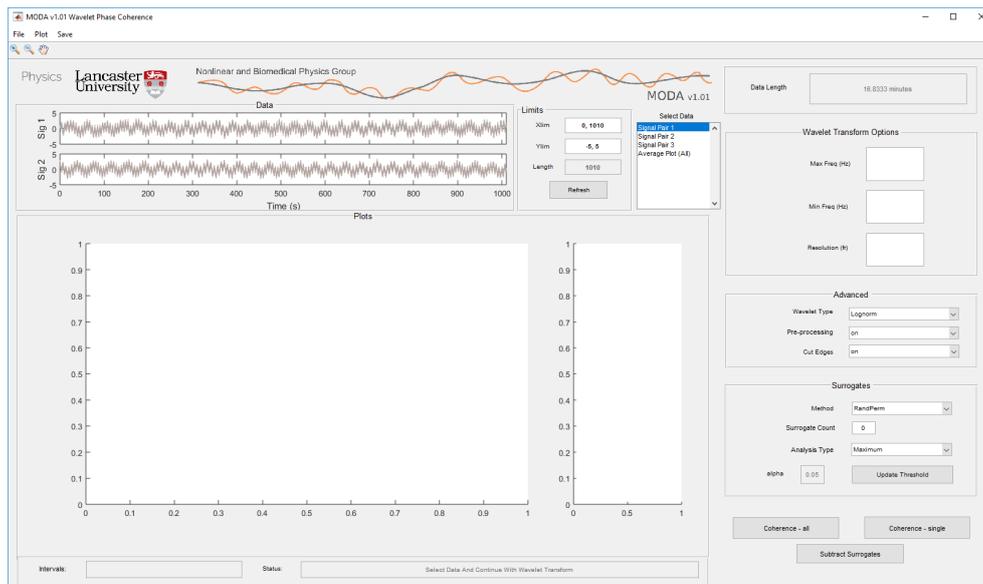
$$\psi(x) \approx \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2 + 2\pi i f_r x}.$$

Consequently: for the same frequency resolution parameter  $f_r$ , the Gaussian WFT and the Morlet WT are linked by the approximation

$$W_X(t, f) \approx G_X^{(f_{\min}=f)}(t, f).$$

# Wavelet Phase Coherence

A file loaded to Wavelet Phase Coherence must contain an even number of signals.



In the small box titled “Select Data”, you should see “Signal Pair 1” selected. You can select any of the signal pairs listed in this box. Here, if you have  $2\tilde{N}$  signals stored in the file that you loaded, then “Signal Pair 1” means the signals stored in the 1st and  $(\tilde{N} + 1)$ -th column/row (depending on whether you clicked “Column Wise” or “Row Wise”) in the file, and likewise “Signal Pair 2” means the signals stored in the 2nd and  $(\tilde{N} + 2)$ -th column/row, and so on. [The option “Average Plot (All)” is not relevant until after all the signals have been analysed using the “Coherence - all” button.] Whichever signal pair is selected, you will see graphs of the two signals in the “Data” box in the top-left of the window; for Signal Pair  $i$ , the upper signal “Sig 1” is the  $i$ -th signal in the loaded file, and the lower signal “Sig 2” is the  $(\tilde{N} + i)$ -th signal in the loaded file. If you wish to have the graph of one of these pairs of signals plotted in a separate window (as a savable file), select the signal pair that you want, and then click on “Plot” (to the right of “File”) and click on the option “Time Series”.

Wavelet-based phase coherence analysis relies on instantaneous phases assigned to each frequency by a wavelet transform. If you wish to carry out a phase coherence analysis of all the pairs of signals contained in the file that you loaded, **you can first select various options for the wavelet transform and for surrogate-based significance testing**, all from within the boxes in the right-most column of the window, and then click **Coherence - all**. Alternatively, if you wish only to carry out a phase coherence analysis of just one of the pairs of signals, then you can select from the “Select Data” box the signal pair that you wish to analyse, and then

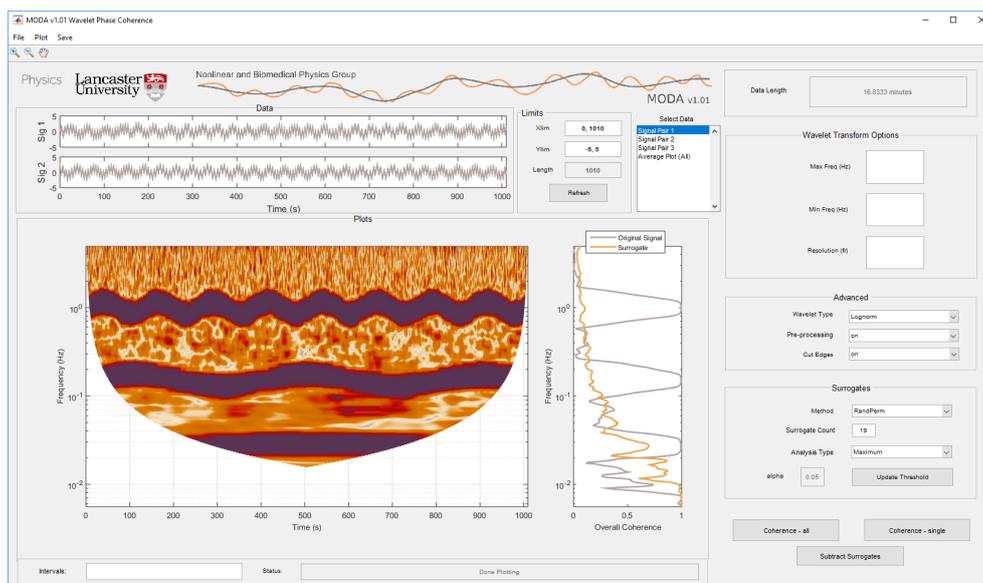
(after selecting the various WT and surrogates options as desired) click **Coherence - single**. Note that clicking “Coherence - single” after results from other signals have been computed will remove all the other results (just like all new computations within a session remove all previously computed results within that same session).

All the options for the wavelet transform are exactly as described in the section on Time-Frequency Analysis. If you do not wish to carry out a surrogate-based significance test, then leave 0 in the “Surrogate Count” field. If you do wish to carry out a surrogate-based significance test, then select the **options for the surrogates**. [“Update Threshold” is not relevant at this point.]

- In the “Method” field, select the type of surrogate; the list is as in the section on Surrogates, except that CPP surrogates are not applicable.
- In the “Analysis Type” field, there are two options. If you select “Maximum”, then the surrogate threshold at each frequency is taken to be the maximum value across the set of surrogates. This is equivalent to a test with significance  $\alpha = \frac{1}{n+1}$  where  $n$  is the chosen number of surrogates, as entered in the “Surrogate Count” field. If you select “Significance”, then the surrogate threshold is determined from the significance level  $\alpha$  entered into the “alpha” field, as in Eq. (16) or (17).
- In the “Surrogate Count” field, enter the number of surrogates.

Given a surrogate count  $n$ , MODA generates  $n$  surrogates from the second of the two signals in each Signal Pair. If pre-processing is on, then the same pre-processing is applied to the surrogates as to the original signal. At each frequency  $f$ , MODA determines the critical value of the overall coherence  $\rho_f$  from the overall coherences at frequency  $f$  between the first signal and the surrogates of the second signal, as described under Version I in the section on Surrogates. Recommendations for the use of surrogates are given in the same section.

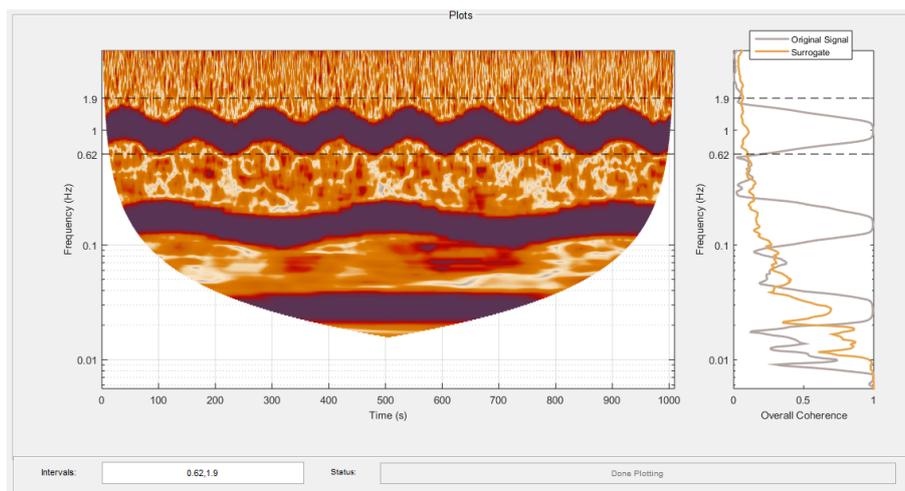
Wavelet phase coherence analysis results are displayed graphically in the box titled “Plots”.



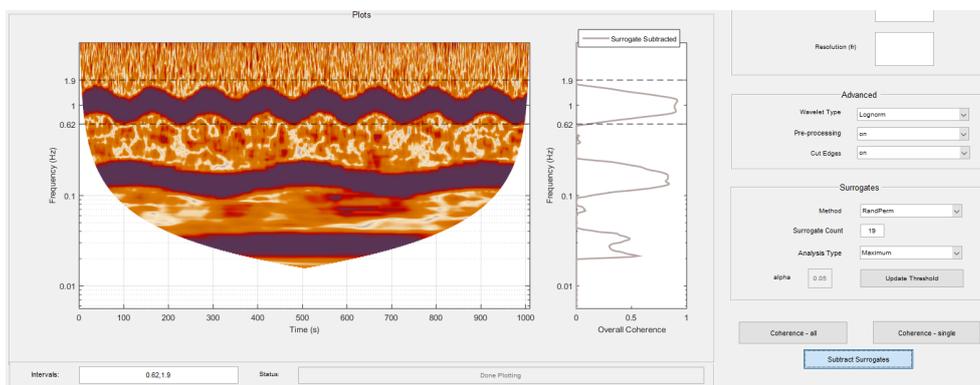
If you selected “Coherence - all”, then you can click on any of the signals in the “Select Data” box to see the results of the analysis for that signal. The phase coherence values  $\rho_{t,f}$  associated

to the different frequencies  $f$  at the different times  $t$  (as described in the Overview of Wavelet Phase Coherence) are represented by a colour-coding within the graph whose axes are Frequency and Time. The graph to the right of this shows the overall phase coherence associated to each frequency. (Note that in this graph, the independent variable “Frequency” is on the vertical axis and the dependent variable “Overall Coherence” is on the horizontal axis. This is important, because it means that the frequency axes on the two graphs can be visually identified with each other.) If a non-zero number was entered into the “Surrogate Count” field, then the surrogate-derived critical values are also shown in yellow.

One can **mark frequencies on the graphs** by entering them in the “Intervals” field, as described in the section on Time-Frequency Analysis.



If (as in the above figure) one observes one or more frequency bands within which there is a path in the time-frequency space of seemingly high coherence values, one can examine the significance of this by seeing whether, in each frequency band, the overall coherence is higher than the critical value. The easiest way to visually identify the frequency bands of “significantly high” phase coherence is by clicking **Subtract Surrogates**. This causes the Overall Coherence graph to show the difference between the actual overall coherence and the surrogate-based critical value, for those frequency-values for which the actual overall coherence is higher.



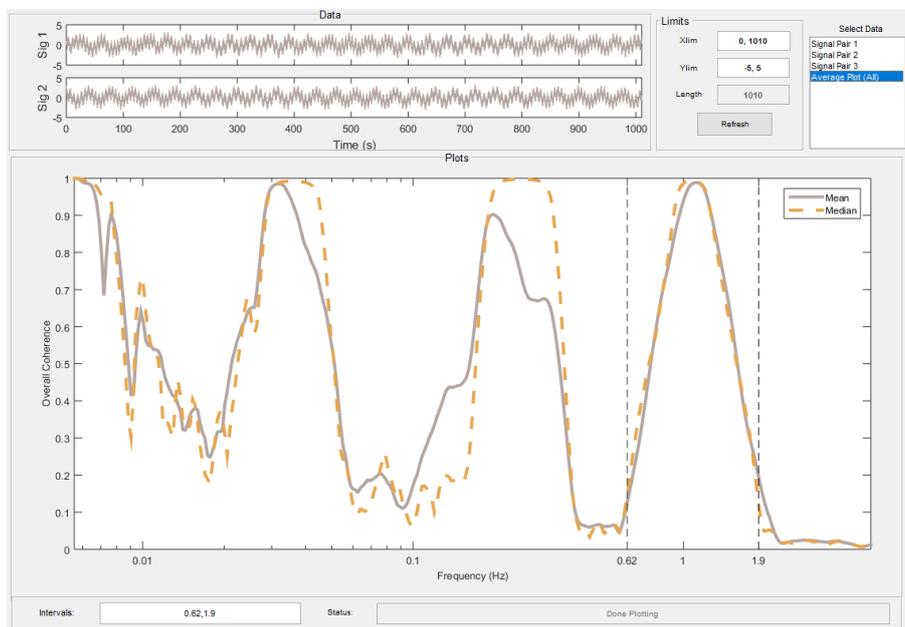
As in the Overview of Wavelet Phase Coherence, we warn again that the overall coherence at low frequencies has a bias towards being large, simply due to the admissible time-interval over which the overall coherence can be calculated being very short at low frequencies. This bias can

be seen in the surrogates. It is important not to try to make conclusions about coherence at very low frequencies.

To return back to showing both the time-averaged coherence and the critical value, just click Subtract Surrogates again.

It is also possible, having computed surrogate thresholds, to change the choice of significance level without re-computing all the results, simply by changing what is in the “Analysis Type” and/or “alpha” fields and clicking **Update Threshold**. However, if you wish to change the surrogate count and/or type of surrogate (as in the “Method” field), then you will need to re-click “Coherence all/single”.

Now if you clicked on “Coherence - all”, you can look at the overall coherence against frequency averaged across all the pairs of signals, by clicking on “Average Plot (All)” inside the “Select Data” box.



For each frequency value, both the mean and the median of the time-averaged phase coherence across all the signal pairs is calculated, and the results are displayed in the graph. Provided there is indeed more than one pair of signals in the file that you loaded, it is possible (just as with Time-Frequency Analysis) to **add onto this graph the time-averaged coherence of individual signal pairs**, as follows: Click “Average Plot (All)” in the “Select Data” box, then press the Ctrl key, and while holding down the Ctrl key, click on the signal pairs in the “Select Data” box that you want to add. One can also mark frequencies on the graph using the Intervals field.

The different graphical results for wavelet phase coherence may be plotted separately as savable files, just as described in the section on Time-Frequency Analysis: If one of the signal pairs is selected, then (under “Plot”) click “Time-frequency coherence” to plot whatever is currently in the left graph in the “Plots” box, click “Average coherence - single” to plot whatever is currently in the right graph in the “Plots” box, and click “Time-frequency + average” to plot both in a single file. If “Average Plot (All)” is selected (with or without individual signal pairs

superposed), click “Average coherence - all” to plot whatever is in the graph in the “Plots” box.

## Saving numerical values of results

Suppose the file loaded into MODA for analysis consists of  $N = 2\tilde{N}$  signals.

After computing results, if you click “Save” and then “Save as .mat”, the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled `Coherence_data`, containing the following fields:

- “Time”: the complete discretised time axis (in seconds), as a row vector;
- “Frequency”: the complete discretised frequency axis (in Hz), as a column vector;
- “Wavelet\_Type”: as selected in the “Wavelet Type” field;
- “Coherence”: an array of  $\tilde{N}$  columns if “Coherence - all” was chosen, or just 1 column if “Coherence - single” was chosen; if “Coherence - all” was chosen, the  $(i, j)$ -entry is the overall coherence at the  $i$ -th frequency value in the discretised frequency axis, for the  $j$ -th signal pair; similarly, if “Coherence - single” was chosen, the  $i$ -th entry is the overall coherence at the  $i$ -th frequency value in the discretised frequency axis, for the single signal pair;
- “Sampling\_frequency”: the number entered for the sampling frequency when the file was being loaded;
- “fmax”: what was entered into the “Max Freq” field, or the Nyquist frequency (in Hz) if this field was left blank;
- “fmin”: what was entered into the “Min Freq” field, or the smallest frequency for which computation was possible (in Hz) if this field was left blank;
- “fr”: what was entered into the “Resolution (fr)” field, or the default value 1 if this field was left blank;
- “Preprocessing”: shows whether pre-processing was chosen to be on or off;
- “Cut\_Edges”: shows whether Cut Edges was chosen to be on or off;

and if surrogates were calculated (i.e. a non-zero number was entered into the “Surrogate Count” field):

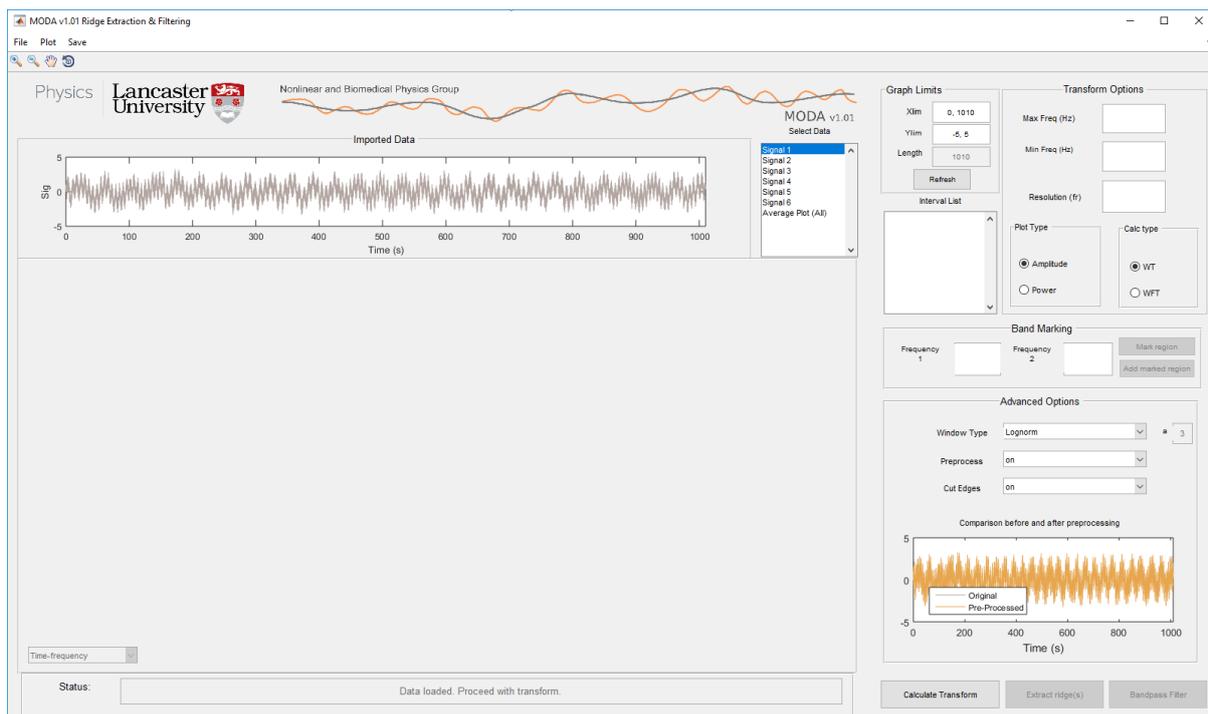
- “Surrogate\_type”: as selected in the “Method” field;
- “Surrogate\_number”: the number of surrogates for each signal pair (based on the second signal of the pair), as entered into the “Surrogate Count” field;
- “Surrogate\_threshold”: either “Maximum” or “Significance  $\alpha$ ” (as selected in the “Analysis Type” field) where  $\alpha$  is the number entered in the “alpha” field;
- “Surrogates”: an array of  $\tilde{N}$  columns if “Coherence - all” was chosen, or just 1 column if “Coherence - single” was chosen; if “Coherence - all” was chosen, the  $(i, j)$ -entry is the surrogate threshold at the  $i$ -th frequency value in the discretised frequency axis, for the  $j$ -th signal pair; similarly, if “Coherence - single” was chosen, the  $i$ -th entry is the surrogate threshold at the  $i$ -th frequency value in the discretised frequency axis, for the single signal pair.

If you click “Save” and then “Save as .csv”, then MODA will save one .csv file, which includes the following:

- the date on which the results were saved;
- the wavelet type, sampling frequency, maximum and minimum frequencies, frequency resolution parameter  $f_r$ , and whether pre-processing and Cut Edges are on or off, as well as the surrogate type, the surrogate number and the definition of the surrogate threshold if surrogates were calculated, all just as when saved as .mat;
- the start time and end time of the signal (in seconds);
- the list of frequency values (in Hz) on the discretised frequency axis, together with the overall coherence at each frequency for all  $\tilde{N}$  signal pairs if “Transform All” was chosen, or just the one analysed signal pair if “Transform Single” was chosen, as well as the corresponding surrogate threshold levels if surrogates were calculated.

# Ridge Extraction & Filtering

The userface for Ridge Extraction & Filtering (when a file is loaded) is as follows:



In the small box titled “Select Data”, you should see “Signal 1” selected. You can select any of the signals listed in this box; here, “Signal 1” means the time-series stored in the 1st column/row (depending on whether you clicked “Column Wise” or “Row Wise”) in the file that you loaded, and likewise “Signal 2” means the time-series stored in the 2nd column/row, and so on. [The option “Average Plot (All)” is not relevant until an initial time-frequency analysis has been carried out using the “Calculate Transform” button.]

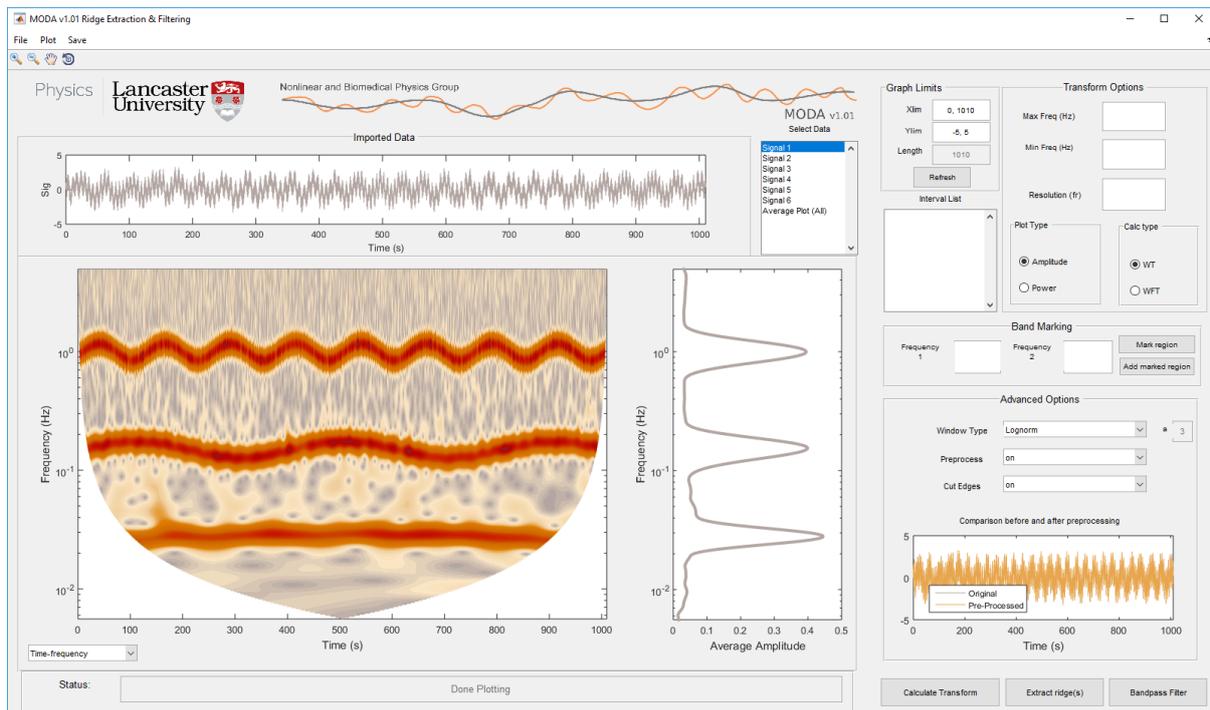
Whichever signal is selected, you will see a graph of that signal in the “Imported Data” box in the top-left of the window. If you wish to have the graph of one of these signals plotted in a separate window (as a savable file), select the signal that you want, and then click on “Plot” (to the right of “File”) and click on the option “Time Series”.

Before filtering or extracting from a signal, **MODA first requires the user to carry out a time-frequency analysis** of the signals. To do this, the user can first select options for the time-frequency analysis (or leave them at MODA’s default settings). These options are specified in some of the boxes in the right-most column of the window, namely:

- the fields “Max Freq”, “Min Freq” and “Resolution (fr)”;
- the selection of either WT or WFT;
- the three fields under “Advanced Options”—the first specifies the type of wavelet (for the WT) or window (for the WFT), and the next two specify whether to pre-process the signal and cut edges in the result.

All these options are described in the section on Time-Frequency Analysis.

Once you have selected the options for the time-frequency analysis, click “Calculate Transform”, and then MODA will apply the chosen analysis to all the signals. The results of the analysis will then be visible in the large box below the “Imported Data” box.

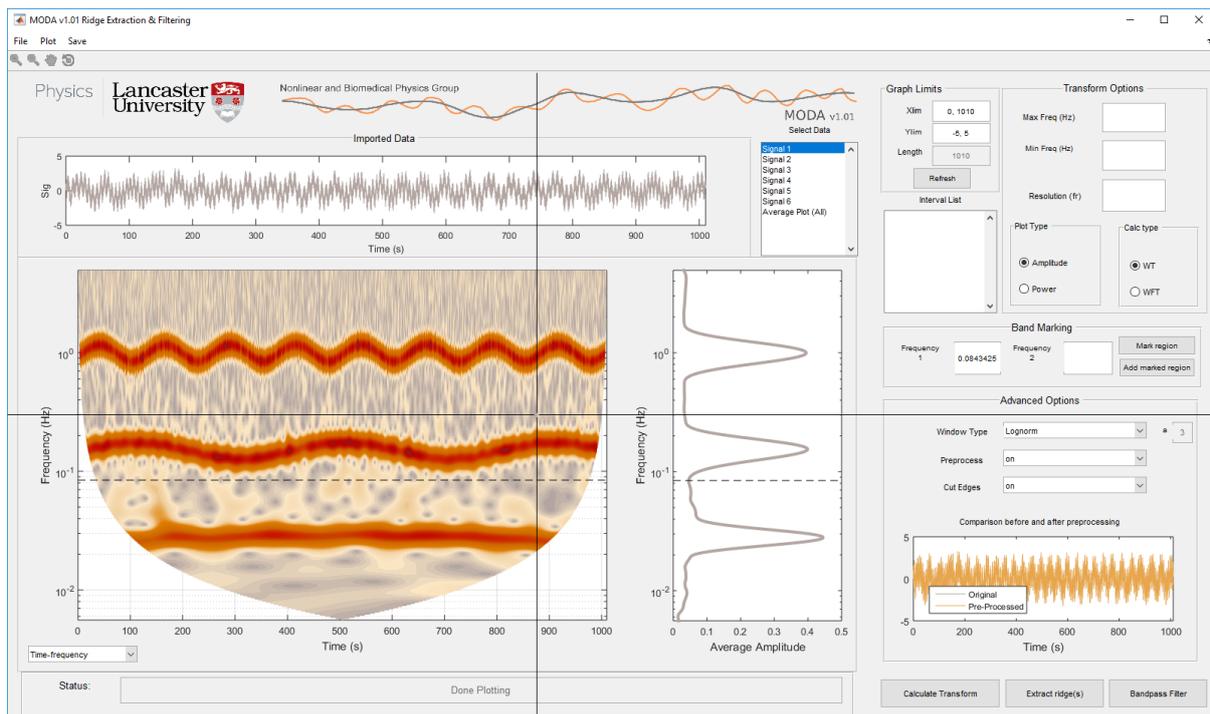


The results shown are for whichever signal is currently selected in the “Select Data” box. It is possible to select whether to display amplitude or power in the “Plot Type” box, and one can also click on “Average Plot (All)” to see the averaged results; all this is described in the section on Time-Frequency Analysis. One can also plot any of the graphs as a savable file in a separate window, as described in the section on Time-Frequency Analysis.

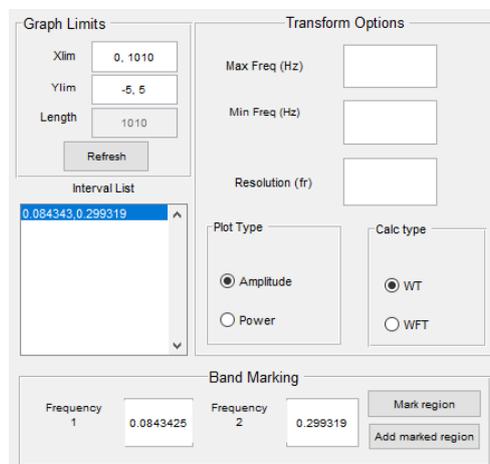
To filter or to extract a ridge from a signal, one first needs to **specify a range of frequencies**; one will then be able to either filter out of the signals all the frequencies that do not lie in this range, or to extract from the signals a ridge that lies in this range. To do this, there are two options:

- Within the box titled “Band Marking”, type the lower and upper boundaries of the desired range of frequencies, in the fields Frequency 1 and Frequency 2 respectively.
- First select any of the signals in the “Select Data” box (but not “Average Plot (All)”); you should select one in which you will most readily be able to see visually the range

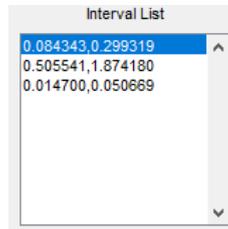
of frequencies of interest in the time-frequency analysis. Then click **Mark region**, and bring the mouse-cursor to within the graph showing the analysis results, and select the lower and upper boundaries of the desired frequency range using the mouse (by clicking when the mouse is in the correct position in the graph). For selecting the frequencies, the horizontal position of the cursor within the graph is irrelevant. When you select the boundary frequencies with the mouse in this manner, the selected frequencies will automatically be displayed in the Frequency 1 and Frequency 2 fields in the box titled “Band Marking”. It does not matter whether you select the lower boundary and then the upper boundary, or the upper boundary and then the lower boundary. If, after selecting the two frequencies, you are not happy with the frequencies that you selected, you can try again by simply clicking “Mark region” again.



Once the Frequency 1 field and the Frequency 2 field are filled with the boundary frequencies (in either order) of the frequency-interval of interest, click **Add marked region**. This will cause the frequency-interval of interest to appear in the box titled “Interval List”.



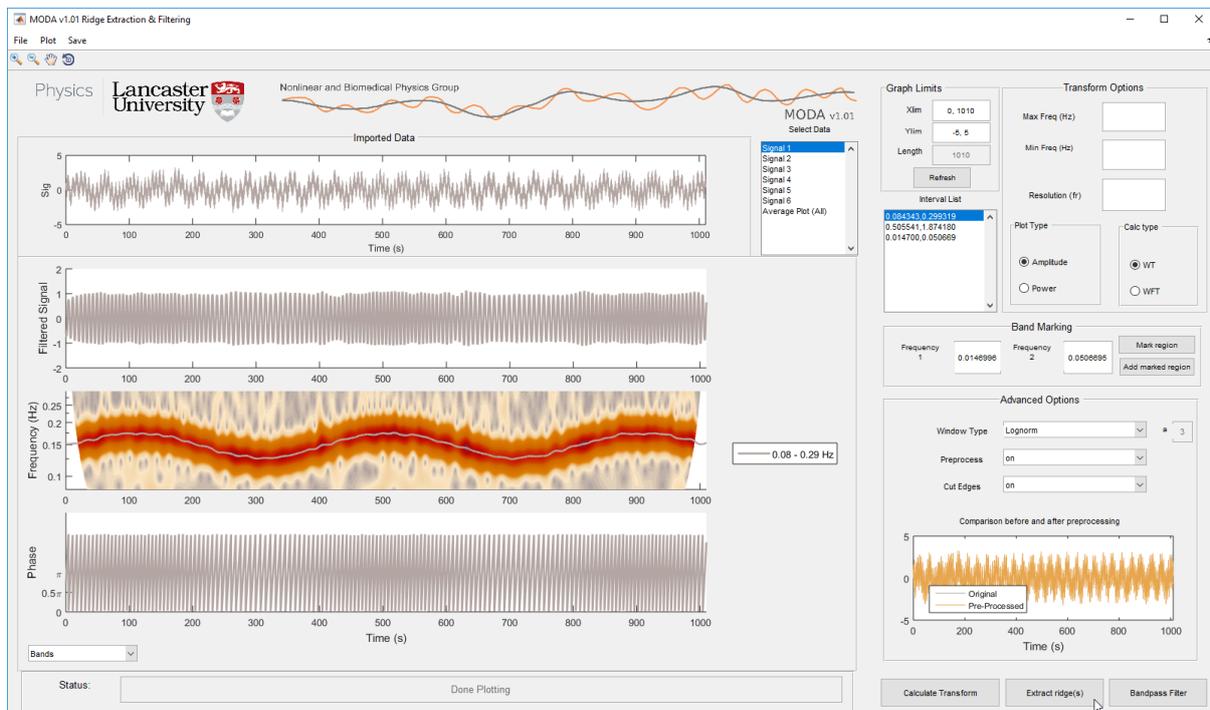
You may wish to extract ridges from, or bandpass-filter within, **several different frequency-bands of interest**. To do this, one simply repeats the above procedure for typing/marking and adding frequency intervals; each new added interval will appear at the bottom of the Interval List.

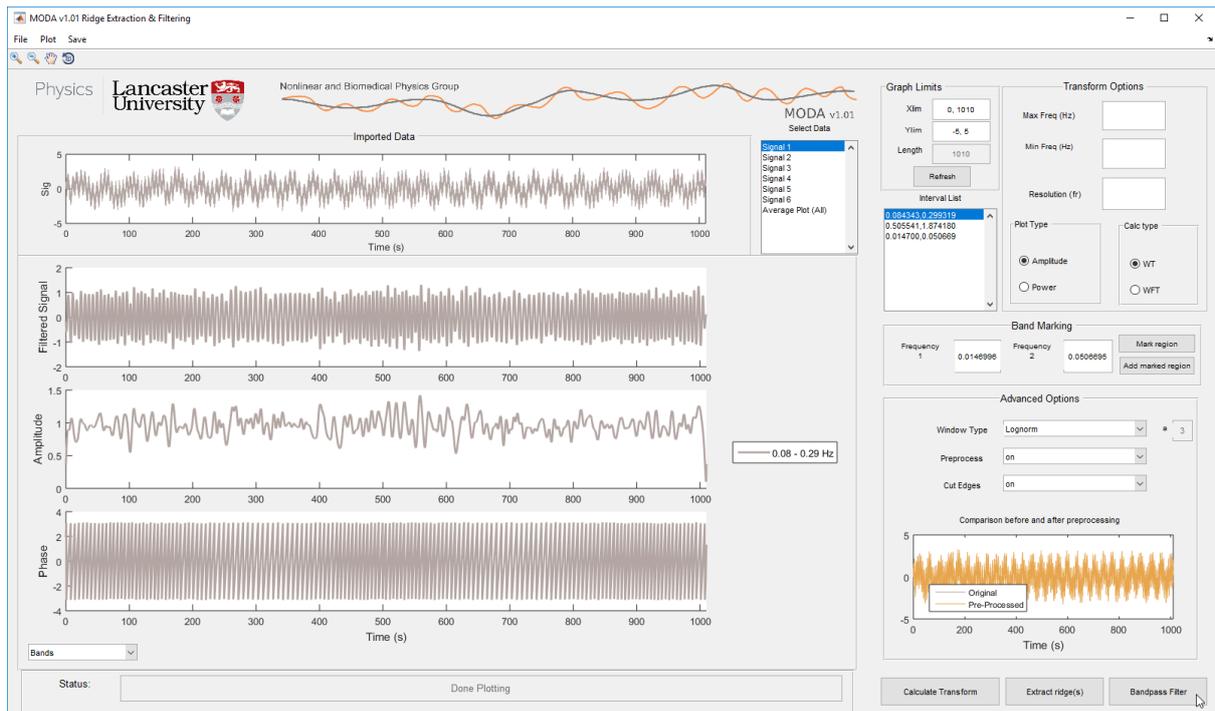


In case you are more interested certain frequency-bands for one signal and certain other frequency-bands for a different signal, you may wish to change which signal in the “Select Data” box is selected between consecutive band-markings via the “Mark region” facility.

Any interval added to the Intervals List can subsequently be removed by clicking on the interval within the Intervals List box and pressing the Delete key.

Once you have added all the frequency-bands of potential interest, you can click either **Extract ridge(s)** or **Bandpass Filter**. MODA will then carry out the ridge-extraction or filtering (as selected) for all the signals, in all the frequency-intervals present in the Interval List. If filtering was chosen, the Butterworth filter is applied to the original signals. If ridge-extraction was chosen, then *for each frequency-interval in the Intervals List*, MODA first *recomputes* time-frequency representations for the signals, and then uses these recomputed time-frequency representations to extract the ridge. These recomputations are exactly according to the options specified by the user for the original time-frequency analysis (including whether pre-processing is on or off), *except* that values for Max Freq and Min Freq are taken to be the boundary frequencies of the frequency-interval from which the ridge will be extracted.





The small pull-down list in the bottom left corner of the box where results are plotted now shows the word “Bands”, referring to the fact that the box now shows the results of the filtering/extraction. (To return to the results of the original time-frequency analysis, click on the pull-down list and select “Time-frequency”).

- **If you chose “Extract ridge(s)”**: The box which had shown the results of the time-frequency analysis will now show three graphs, sharing a common horizontal axis, namely the time-axis. For whichever signal is currently selected, for whatever interval is currently selected in the Interval List,
  - the top graph displays the resulting (reconstructed) signal from when the ridge in the time-frequency representation of the selected signal, as identified by MODA within the selected frequency range, is extracted from the selected signal;
  - the middle graph shows in grey the ridge itself (within time-frequency space), superposed over the portion of the time-frequency representation of the selected signal that lies within the selected frequency range (showing either amplitude or power as in the “Plot Type” box);
  - the bottom graph shows, for each moment in time  $t$ , the instantaneous phase that is assigned, by the time-frequency analysis of the selected signal, to the point in time-frequency space where the ridge is at time  $t$ .

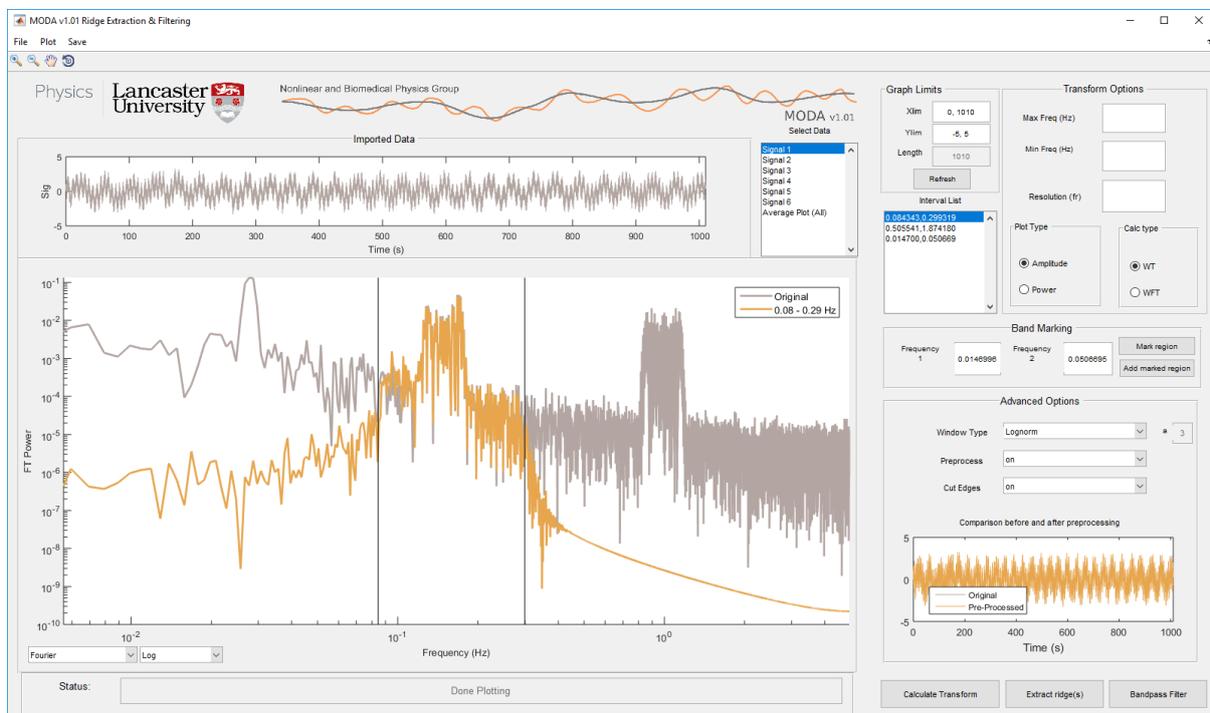
Though not shown, one can also save the instantaneous amplitudes extracted from the ridge; see the section “Saving numerical values of results” below.

- **If you chose “Bandpass Filter”**: Again, the box which had shown the results of the time-frequency analysis will now show three graphs, sharing a common horizontal axis, namely the time-axis. For whichever signal is currently selected, for whatever interval is currently selected in the Interval List,

- the top graph displays the resulting signal when all frequencies outside the selected range are filtered out of the selected signal;
- the middle and bottom graphs display, respectively, the instantaneous amplitudes and phases of the filtered signal (shown in the top graph) as defined by the analytic representation.

In either case, the top, middle, or bottom graph may be plotted as a savable file in a separate window by clicking on “Plot” and then “Filtered Signal”, “Ridge/Amplitude”, or “Phase”, respectively; and all three graphs can be plotted in one savable file by clicking on “Plot” and then “Filtered + amp/phase”.

It is also possible to **compare the Fourier power spectrum** of the original (or pre-processed if pre-processing was “on” for the time-frequency analysis) with that of the filtered/extracted signal. To do this, select “Fourier” from the pull-down list where “Time-frequency” and “Bands” are.<sup>9</sup>



The frequency band from which the resulting signal was extracted/filtered (as selected in the “Interval List” box) is also marked inside the graph. You can select either logarithmic scales for the axes, or linear scales for the axes, in the pull-down list to the right of the one where “Fourier” is selected. The graph can be plotted as a savable file by clicking on “Plot” and then “Fourier transform”.

**Practical remark.** For good results in ridge-extraction, it is important that for each frequency range specified by the user:

- there is *one* predominant oscillatory component (the component of interest) within the frequency range;

<sup>9</sup>The Fourier power at a given frequency  $f$  is defined here as twice the square of the amplitude assigned to  $f$  by the Fourier transform.

- (ii) over the entire time-range, the frequency peak represented by this component in the time-frequency analysis of the signal—as well as the entire frequency blur around this peak—is contained within the frequency range;
- (iii) over the entire time-range, the frequency peak represented by any *other* major oscillatory component—as well as the entire frequency blur around this peak—remains *outside* the frequency range;
- (iv) along the same lines as (iii), there must be no significant interference within the frequency range from “negative frequencies”.

This last point is not a problem for the wavelet transform with the lognormal or bump wavelet. For the wavelet transform with the Morlet wavelet, and for the windowed Fourier transform, the issue of negative-frequency interference is avoided by either having pre-processing on or making sure that the frequency resolution  $f_r$  is at least about 0.7.

**If you wish to use results for the Dynamical Bayesian Inference application:**

In Dynamical Bayesian Inference, you can either investigate couplings between oscillatory components of different signals, or investigate couplings between two oscillatory components within the same signal. In either case, the analysis can be performed for multiple signals or signal pairs at once, so long as the same pair of frequency bands is used for all of them.

- To investigate couplings **between** signals  $x_i(t)$  and  $y_i(t)$  ( $1 \leq i \leq N$ ): Upload the set of signals  $x_1(t), \dots, x_N(t)$  as a single file into the Ridge Extraction & Filtering application, and carry out the ridge-extraction or filtering **in only one frequency band**. Then click “Save” and then “Save as .mat”. The .mat file created by this is one of the two files that will be loaded into Dynamical Bayesian Inference. Also upload the set of signals  $y_1(t), \dots, y_N(t)$  as a single file into the Ridge Extraction & Filtering application, and carry out the ridge-extraction or filtering, again **in only one frequency band**. Then click “Save” and then “Save as .mat”. The .mat file created by this is the second of the two files that will be loaded into Dynamical Bayesian Inference. *The two original files loaded into Ridge Extraction & Filtering must have the same dimensions (i.e. same number of signals and same number of samples in each signal), and the same sampling frequency must be entered for both.*
- To investigate couplings **within** signals  $x_1(t), \dots, x_N(t)$ : Upload the set of signals  $x_1(t), \dots, x_N(t)$  as a single file into the Ridge Extraction & Filtering application, and carry out the ridge-extraction or filtering **in exactly two frequency bands**. Then click “Save” and then “Save as .mat”. The .mat file created by this will be loaded into Dynamical Bayesian Inference.

Note that in either case, the files **must be saved as .mat**.

## Saving numerical values of results

Suppose the file loaded into MODA for analysis consists of  $N$  signals, and extraction/filtering was carried out in  $m$  frequency bands.

### Ridge-extraction

After computing ridge-extraction, if you click “Save” and then “Save as .mat”, the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled `Filtered_data`, containing the following fields:

- “Time”: the complete discretised time axis (in seconds), as a row vector;
- “Analysis\_Type”: Wavelet or Windowed Fourier, as selected;
- “Wavelet\_Type” or “Window\_Type”; for the Kaiser window, the  $a$ -parameter is also included, e.g. “kaiser-3” means the Kaiser window with  $a = 3$ ;
- “Sampling\_frequency”: the number entered for the sampling frequency when the file was being loaded;
- “fr”: what was entered into the “Resolution (fr)” field, or the default value 1 if this field was left blank;
- “Preprocessing”: shows whether pre-processing was chosen to be on or off;
- “Cut\_Edges”: shows whether Cut Edges was chosen to be on or off;
- “Filter\_type”: Ridge Extraction;
- “Freq\_bands”: a single-column cell with  $m$  entries, namely the frequency bands for which results were computed, as listed in the “Interval List” box, in the same order that they appear in the “Interval List” box; each frequency band is expressed as the lower boundary frequency and the upper boundary frequency (both in Hz) separated by a comma;
- “Ridge\_recon”: an  $N \times m$  cell, whose  $(i, j)$ -entry is a row vector consisting of the series of values in the reconstructed signal [as shown in the top of the three graphs when “Bands” is selected] from the  $i$ -th signal in the  $j$ -th frequency band; the  $l$ -th entry of this row vector is the value of the signal at the  $l$ -th time in the discretised time axis;
- “Ridge\_frequency”: the same, but for the instantaneous frequencies as shown in the middle of the three graphs when “Bands” is selected;
- “Ridge\_phase”: the same, but for the phases (between  $-\pi$  and  $\pi$ ) as shown in the bottom of the three graphs when “Bands” is selected;
- “Ridge\_amplitude”: the same, but for the amplitudes assigned along the ridge by the time-frequency representation, multiplied by the required constant (determined by TFA options) to obtain the true amplitude.

If you click “Save as .csv” and save with a file name, say, “Results”, then in the folder where you selected to save the results, MODA will save a folder with the same file name “Results”. This folder contains three .csv files, whose names are “extracted\_modes.csv”, “ridge\_amplitudes.csv” and “ridge\_frequencies.csv”. All three files contain:

- the date on which the results were saved;
- the analysis type, wavelet/window type, sampling frequency, frequency resolution parameter  $f_r$ , and whether pre-processing and Cut Edges are on or off, and the “filter type” (namely Ridge Extraction) all just as when saved as .mat;
- the start time and end time (in seconds);
- the frequency bands (Band 1, Band 2, etc.) from which the ridges were extracted, with the lower boundary frequencies in one column and the upper boundary frequencies adjacently in the next column.

Beyond this,

- “extracted\_modes.csv” contains the discretised time axis together with the reconstructed signals in all the bands;
- “ridge\_frequencies.csv” contains the discretised time axis together with the instantaneous frequencies in all the bands;
- “ridge\_amplitudes.csv” contains the discretised time axis together with the instantaneous amplitudes in all the bands.

The phases are *not* saved when saving as .csv.

### Bandpass filtering

After bandpass filtering, if you click “Save” and then “Save as .mat”, once again the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled Filtered\_data, containing the following fields:

- “Time”, “Sampling\_frequency” and “Freq\_bands” as before;
- “Filter\_type”: Butterworth Filter;
- “Filtered\_sigs”: an  $N \times m$  cell, whose  $(i, j)$ -entry is a row vector consisting of the series of values in the filtered signal [as shown in the top of the three graphs when “Bands” is selected] from the  $i$ -th signal in the  $j$ -th frequency band; the  $l$ -th entry of this row vector is the value of the signal at the  $l$ -th time in the discretised time axis;
- “Filtered\_amplitudes:” the same, but for the instantaneous amplitudes as shown in the middle of the three graphs when “Bands” is selected;
- “Filtered\_phases”: the same, but for the phases (between  $-\pi$  and  $\pi$ ) as shown in the bottom of the three graphs when “Bands” is selected.

If you click “Save as .csv” and save with a file name, say, “Results”, then in the folder where you selected to save the results, MODA will save a folder with the same file name “Results”. This folder contains three .csv files, whose names are “filtered\_signals.csv”, “filtered\_amplitudes.csv” and “filtered\_phases.csv”. These files contain:

- the date, sampling frequency, “filter type” (namely Butterworth Filter), start time and end time, and frequency bands (all just as described above for Ridge Extraction);
- the discretised time axis together with the filtered signals / instantaneous amplitudes / instantaneous phases (as according to the file name) in all the bands.

### Formula for the bandpass filter

Given a continuous-time signal  $x(t)$  with Fourier transform  $\hat{x}$ , passing this signal through a *zero-phase Butterworth bandpass filter of filter order  $4n$  with passband  $[f_{\min}, f_{\max}]$*  outputs a signal  $x_{\text{filt}}(t)$  whose Fourier transform  $\hat{x}_{\text{filt}}$  is given by

$$\hat{x}_{\text{filt}}(f) = \frac{\hat{x}(f)}{1 + \left( \frac{f}{f_{\max} - f_{\min}} + \frac{f^{-1}}{f_{\max}^{-1} - f_{\min}^{-1}} \right)^{2n}}.$$

Of course, recorded signals are not continuous but discrete. MODA uses MATLAB's `butter` and `filtfilt` functions for a discrete version of the zero-phase Butterworth filter.<sup>10</sup>

For large filter orders, the results become numerically severely unstable. MODA finds the first  $n$  for which the order- $4n$  filter “blows up”, and then applies the filter of order  $4 \max(n-2, 1)$ . “Blowing up” corresponds to when the filtered signal has values whose magnitude exceeds ten times the maximum magnitude that the filtered signal for  $n = 1$  has.

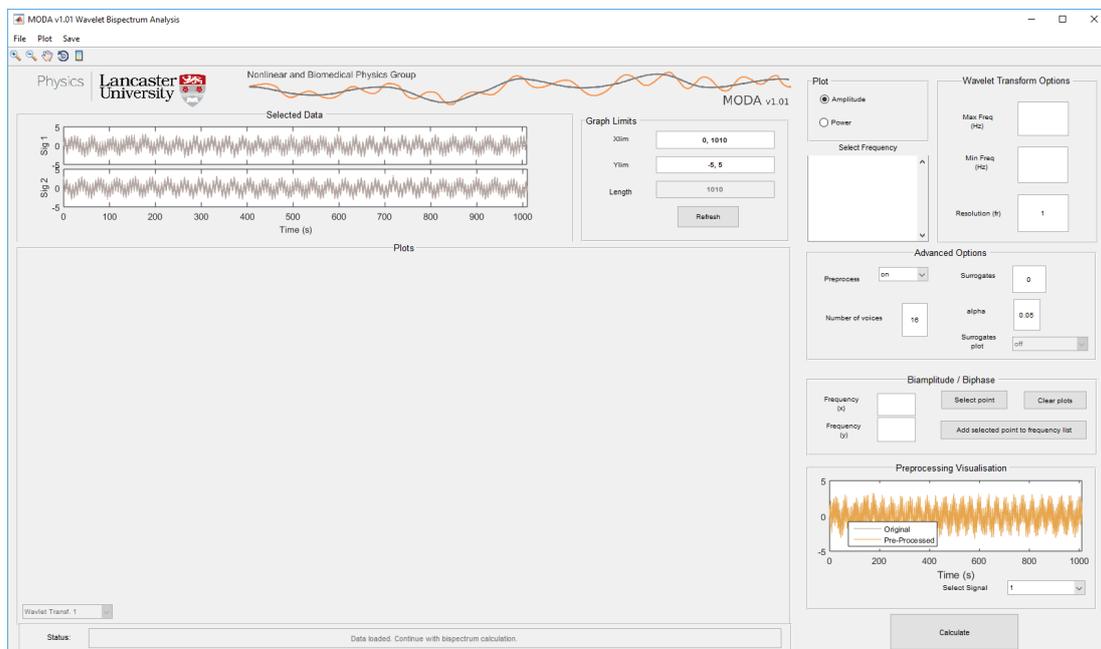
---

<sup>10</sup>`butter` generates a discrete version of the original “causal” Butterworth filter where the output at any time  $t$  only depends on the past of the input prior to time  $t$ ; however, this version of the filter has phase-distortion, making it useless for extracting phases from time-varying frequency oscillations. `filtfilt` then runs a procedure to convert a discrete version of a causal filter into a discrete version of a zero-phase-distortion filter; the main part of this procedure is “forward-backward” filtering, which cancels out the phase distortion of the original causal filter.

# Wavelet Bispectrum Analysis

A file loaded into Wavelet Bispectrum Analysis must contain either one or two signals.

- If the file contains two signals, then these are treated as a bivariate time-series: MODA will carry out both autobispectral analysis of each signal and crossbispectral analysis between the two signals.
- If the file contains one signal: MODA will carry out autobispectral analysis of this signal, and will leave blank the plots of all results pertaining to “Signal 2”.



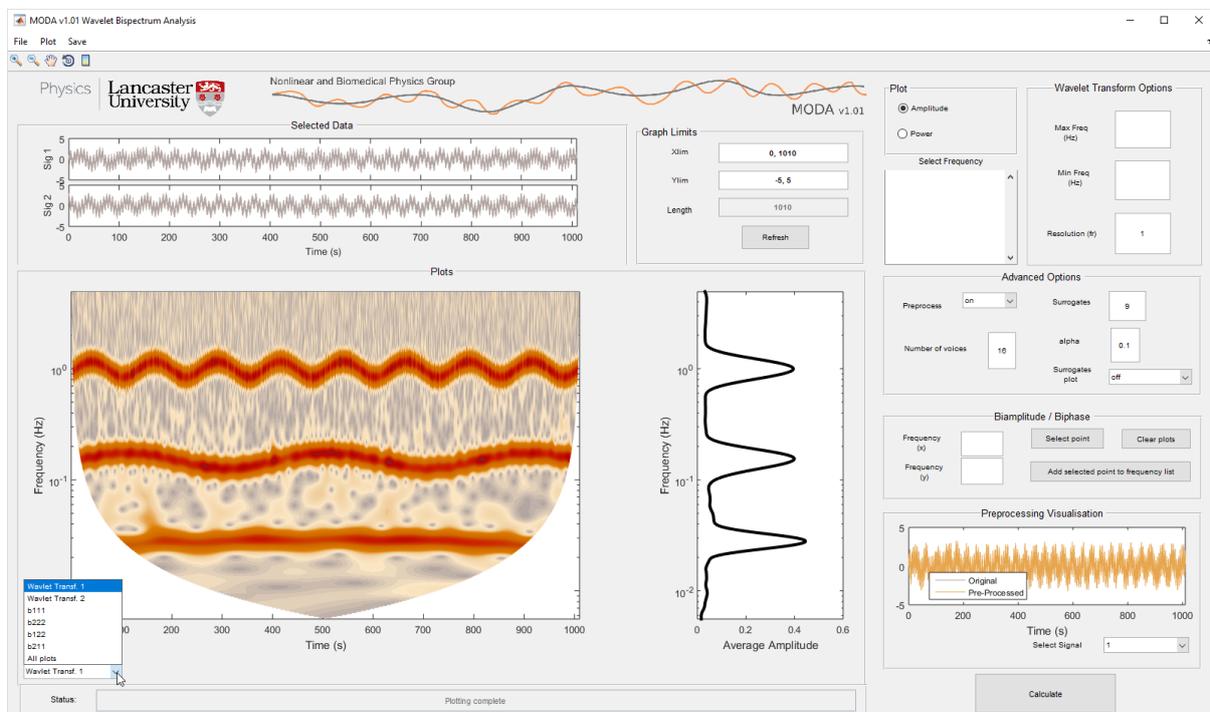
When you load a file, you will see graphs of the signal(s) in the “Selected Data” box in the top-left of the window. If you wish to have this plotted in a separate window (as a savable file), click on “Plot” (to the right of “File”) and click on the option “Time Series”.

Now one can select certain options (described in the section on Time-Frequency Analysis) for the time-frequency representation on which the bispectral analysis will be based; this will always be a **wavelet transform** based on the **lognormal wavelet**. The user specifies these options in the “Wavelet Transform Options” and “Advanced Options” boxes, which are in the right-most column of the window; apart from “Number of voices”, the options are just as for Time-Frequency Analysis. (With pre-processing on, one can view the pre-processed version of either of the two signals, as selected in the “Select Signal” pull-down list.) The user can specify the fineness of the discretisation of the frequency axes for the bispectrum. This is specified in

the “**Number of voices**” field: if a number  $m$  is entered, then the ratio between consecutive frequency values is  $2^{\frac{1}{m}}$ . The default value is 16, but due to how long it takes to compute bispectra, the user may wish to decrease this number.

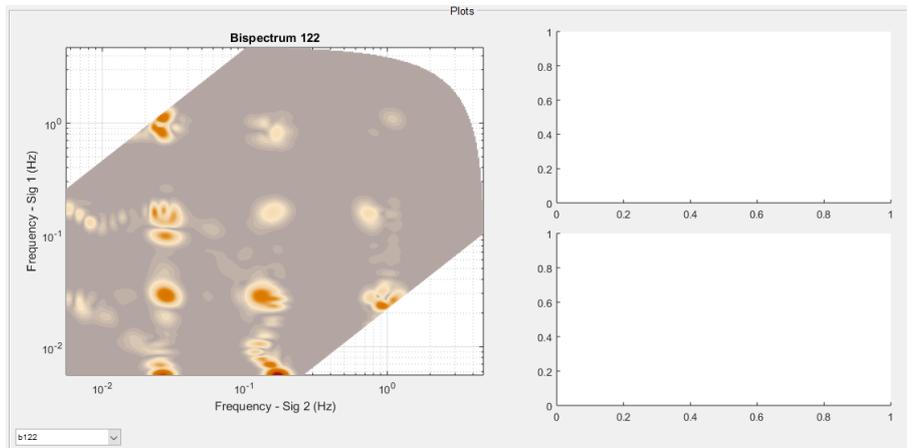
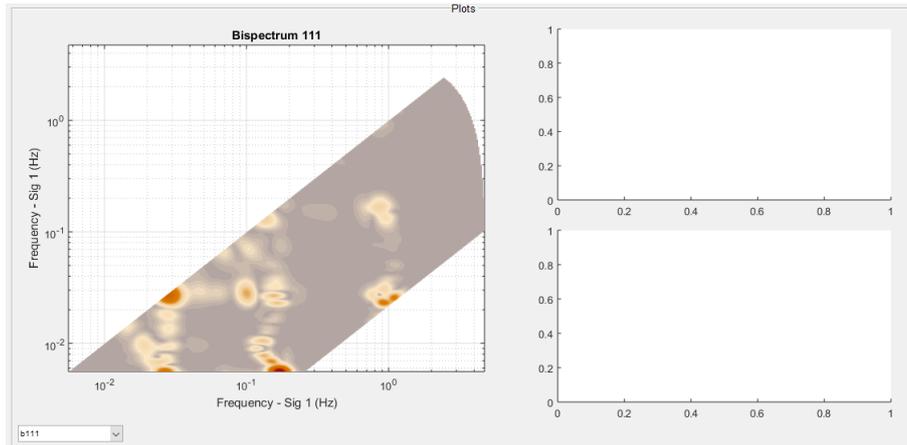
The user can also specify **options for a surrogate-based significance test**: In the “Surrogates” field, enter the number of surrogates, and in the field “alpha”, enter the desired significance level. If pre-processing is on, then the same pre-processing is applied to the surrogates as to the original signals. If you do not wish to perform a surrogate-based test, just leave 0 in the “Surrogates” field.

Having specified all the above options (or left them at MODA’s default options), click the “Calculate” button. MODA will then carry out the time-frequency analysis and subsequent bispectral analysis of the signal(s), as well as computing surrogate thresholds for the bispectra at each point  $(f_x, f_y)$  in frequency-frequency space. All results are plotted in the large box titled “Plots”. Initially, you will see the results of wavelet transform of Signal 1. (As described in the section on Time-Frequency Analysis, you can choose to see either amplitude or power, and the time-averaged versions are shown in the graph on the right.)



Using the small pull-down list in the bottom left corner of the “Plots” box, you can select to see the wavelet transform of Signal 2, if two signals were uploaded. When the transform of either signal is currently showing, you can plot the result in a separate window as a savable file, just as described in the section on Time-Frequency Analysis: Under “Plot”, click “Time-frequency” to plot the left graph in the “Plots” box, click “Time-average transform” to plot the right graph in the “Plots” box, and click “Time-frequency+ average transform” to plot both in a single file.

Using the same small pull-down list in the bottom left corner of the “Plots” box, you can select to see the results of the wavelet autobispectra  $b_{111}$  and  $b_{222}$ , and the wavelet cross-bispectra  $b_{122}$  and  $b_{211}$ . (If just one signal is uploaded, then you can select to see its autobispectrum  $b_{111}$ .)

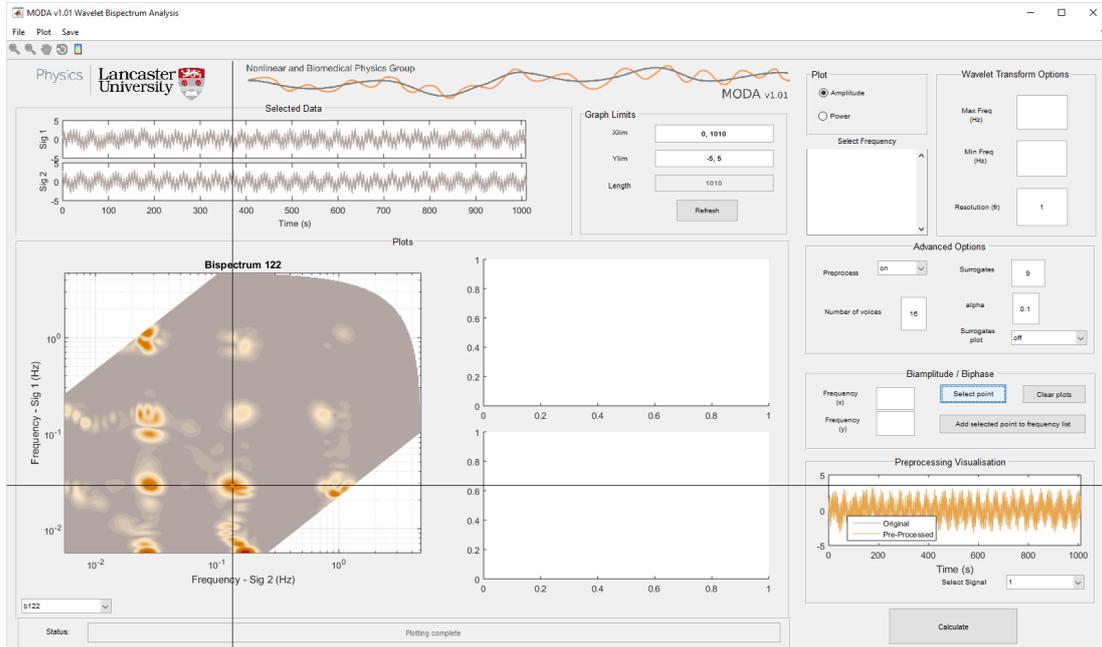


In each of the four cases, the larger graph on the left in the “Plots” box shows the biamplitude  $|b_{ijj}(f_x, f_y)|$ . For the autobispectra, **this graph is symmetric about the diagonal, and so only one half is shown**. To plot the currently shown bispectrum plot (as currently selected in the small pull-down list) as a separate savable file, click on “Plot” and then “Bispectrum”. One can also view all wavelet transforms and bispectra simultaneously, by selecting **All plots** from the small pull-down list. One can open this collection of plots as a single separate savable figure, by clicking on “Plot” and then “All Plots”.

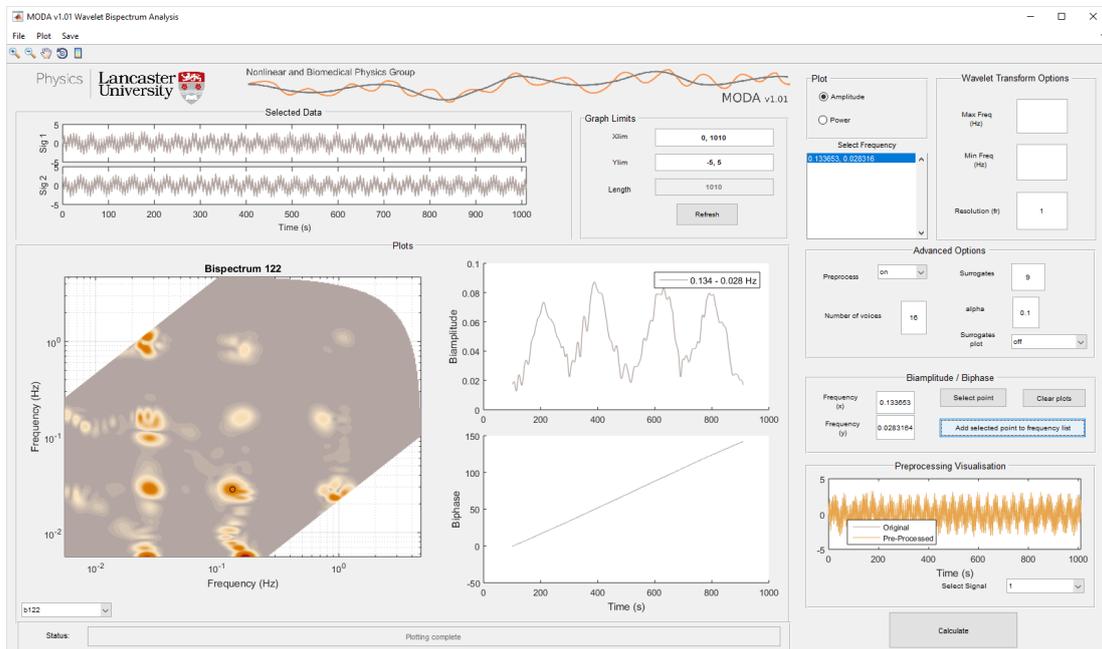
For all of the four bispectra in the pull-down list, it is also possible to **see the time-evolution of the instantaneous biamplitude and biphase** at a given pair of frequencies. To do this, first select which pair of frequencies you wish to look at; there are two ways of doing this:

- Within the box titled “Biamplitude/Biphase”, type the pair of frequencies of interest into the fields “Frequency (x)” and “Frequency (y)”, with Frequency (x) being the frequency on the horizontal axis of the graph showing the bispectrum, and Frequency (y) being the frequency on the horizontal axis of the graph showing the bispectrum.
- Provided one of the four bispectra is currently selected in the small pull-down list: Click **Select Point**, and bring the mouse-cursor to within the graph showing the bispectrum, and select the desired point in frequency-frequency space using the mouse (by clicking when the mouse is in the correct position in the graph). When you select the frequency-pair with the mouse in this manner, the selected frequency-pair will automatically be displayed in the Frequency (x) and Frequency (y) fields in the box titled “Biamplitude/Biphase”. If,

after selecting the frequencies, you are not happy with the frequencies that you selected, you can try again by simply clicking “Select Point” again.



Once the Frequency (x) field and the Frequency (y) field are filled with the frequency-pair of interest, click **Add selected point to frequency list**.

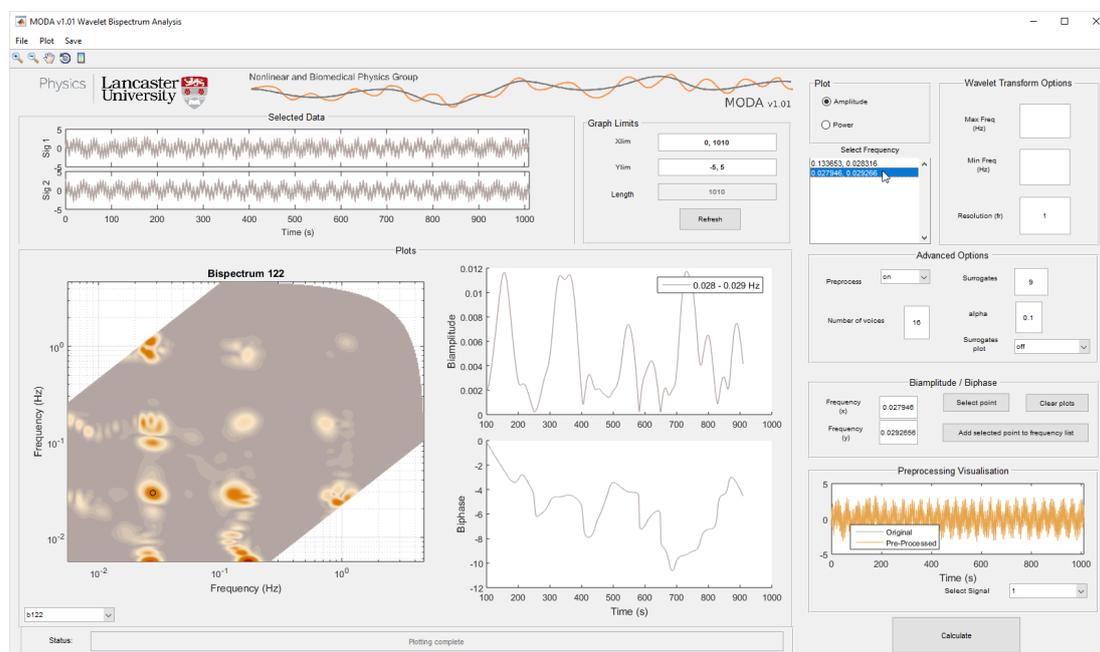


This will cause the frequency-pair of interest to appear in the box titled “Select Frequency”, and the associated instantaneous biamplitude and biphase (as a function of time) will appear in the pair of graphs on the right within the “Plots” box. These graphs share the same time-axis, with the upper graph showing instantaneous biamplitude and the lower showing instantaneous biphase. The selected frequency-pair is also marked in the left graph by a circle around the

point in frequency-frequency space.

Note that the vertical axis on the lower graph for instantaneous biphase is *not* from  $-\pi$  to  $\pi$  (or from 0 to  $2\pi$ ); this is because it shows the **“unwrapped” time-evolution of the instantaneous biphase**: The vertical axis spans a broader subset of the real line (if necessary), such that the instantaneous biphase  $\phi_{ijj}(t, f_x, f_y)$  may be seen as a continuous function of  $t$ , rather than undergoing sudden jumps at 0 or  $\pi$ . This will enable the user to detect any overall clockwise or anticlockwise trend over time.

You can add more frequency-pairs at which to look at the evolution of the instantaneous biamplitude and biphase; for simplicity of use, the “Select Frequency” list keeps no memory of which of the four bispectra was selected when each frequency-pair was added. For whichever of the four is currently selected in the small pull-down list, you can **click on any of the added frequency-pairs**, and the associated instantaneous biamplitude and biphase will be shown. **The frequency-pair for which results are currently displayed is shown by a small circle on the frequency-frequency plot.**

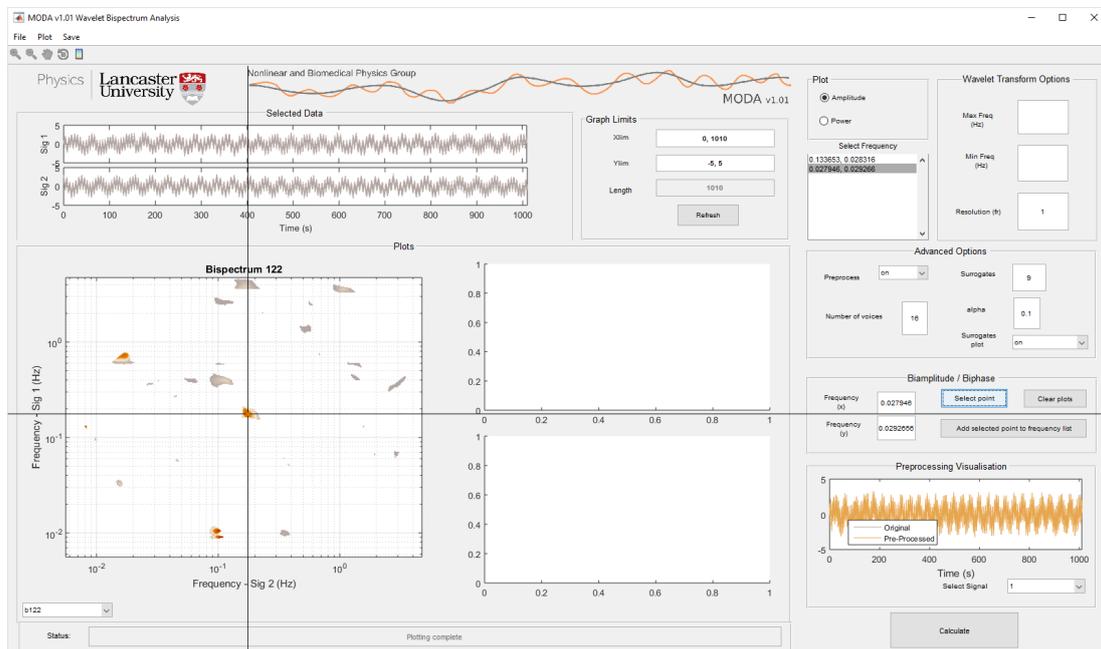


Within any one of the four bispectra, it is possible to see the evolution of the instantaneous biamplitude and biphase simultaneously for more than one of the selected frequency-pairs: select one of the desired frequency-pairs from the “Select Frequency” list, and then press the Ctrl key, and while holding down the Ctrl key, click on the other frequency-pairs of interest from the “Select Frequency” list.

Any frequency-pair added to the list can subsequently be removed by clicking on that frequency-pair within the list in the “Select Frequency” box, and then pressing the Delete key.

In any of the four bispectra, to see the results of the surrogate test (if a non-zero number was entered into the “Surrogates” field), **select “on” in the “Surrogates plot” field: all points in the plot where the actual biamplitude is less than the surrogate threshold for that point will be turned white**, so that the remaining parts of the plot are where

the biamplitude is significantly high. The colour-coding will then reflect how high above the surrogate threshold the remaining points are. (This colour-coding will be re-scaled for the distances above the surrogate threshold.) It remains possible to add frequency-pairs.



Results for the instantaneous biamplitude and biphase remain unaffected by whether “Surrogates plot” is on or off.

As described in the Overview of Wavelet Bispectrum Analysis, values exceeding the surrogate threshold can only be taken as evidence of coupling if the instantaneous biphase either is roughly flat over time, or at least is roughly flat over parts of the duration of the signal.

In the surrogate test, MODA uses **WIAAFT surrogates**. If the number of surrogates entered into the “Surrogates” field is  $n$ , then MODA generates  $n$  surrogates from Signal 1 and (independently of this)  $n$  surrogates from Signal 2. At each frequency-pair,

- the critical amplitudes of the crossbispectra are determined from the  $n$  pairs of surrogates, each consisting of the  $i$ -th surrogate of Signal 1 and the  $i$ -th surrogate of Signal 2 ( $1 \leq i \leq n$ ) as in Version II in the section on Surrogates;
- the critical amplitude of the autobispectrum  $b_{111}$  is determined from the  $n$  surrogates of Signal 1, and critical amplitude of the autobispectrum  $b_{222}$  is determined from the  $n$  surrogates of Signal 2, each as in Version III in the section on Surrogates.

Recommendations for the use of surrogates are given in the section on Surrogates.

Bispectrum results (with or without surrogates) may be plotted separately as savable files: Under “Plot”, click “Bispectrum” to plot whatever is currently in the left graph in the “Plots” box, click “Biamplitude” to plot what is in the upper-right graph in the “Plots” box, click “Biphase” to plot what is in the lower-right graph in the “Plots” box, and click “Bispectrum+biamp+biphase” to plot all three in a single file.

**Note:** The wavelet bispectrum is very slow to compute, and WIAAFT surrogates (for which the bispectrum also needs to be computed) are also slow to generate.

## Saving numerical values of results

After analysing a pair of signals, if you click “Save” and then “Save as .mat”, the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled `Bisp_data`, containing the following fields:

- “Time”: the complete discretised time axis (in seconds), as a row vector;
- “Frequency”: the complete discretised frequency axis (in Hz), as a column vector;
- “Sampling\_frequency”: the number entered for the sampling frequency when the file was being loaded;
- “fmax”: what was entered into the “Max Freq” field, or the Nyquist frequency (in Hz) if this field was left blank;
- “fmin”: what was entered into the “Min Freq” field, or the smallest frequency for which computation was possible (in Hz) if this field was left blank;
- “fr”: what was entered into the “Resolution (fr)” field, or the default value 1 if this field was left blank;
- “Preprocessing”: shows whether pre-processing was chosen to be on or off;
- “WTAmplitude”/“WTPower” (as selected in “Plot Type”): an array with two rows, where the  $(i, j)$ -entry is the time-averaged amplitude/power at the  $j$ -th frequency value in the discretised frequency axis, for the  $i$ -th signal;
- “b111”: an array whose  $(i, j)$ -entry is the biamplitude  $|b_{111}(f_i, f_j)|$ , where  $f_i$  and  $f_j$  are respectively the  $i$ -th and  $j$ -th frequency value in the discretised frequency axis, provided the bispectrum is computed at  $(f_i, f_j)$  and  $i \leq j$ ; for  $i > j$  and for each  $(i, j)$  where the bispectrum is not computed, the  $(i, j)$ -entry is just NaN; in other words, b111 shows precisely what is shown in the Bispectrum 111 plot;
- “b222”: the same for the biamplitudes  $|b_{222}|$ ;
- “b122”: the same for the biamplitudes  $|b_{122}|$ , where now (as in the Bispectrum 122 plot) values are shown for  $i > j$  as well as for  $i \leq j$ , since these are not the same;
- “b211”: likewise, for the biamplitudes  $|b_{211}|$ ;

and if surrogates were computed (with surrogate count, say,  $n$  entered in the “Surrogates” field) *and* “Surrogates plot” was *on* at the time of saving, then also:

- “surrnum”: the surrogate count  $n$ ;
- “alpha”: the significance level  $\alpha$ ;
- “b111surr”, “b222surr”, “b122surr”, “b211surr”: each of these is a 3-dimensional array such that for each  $l \in \{1, \dots, n\}$ , the 2-dimensional arrays  $b111surr(:, :, l)$ ,  $b222surr(:, :, l)$ ,  $b122surr(:, :, l)$ ,  $b211surr(:, :, l)$ , are exactly the same as b111, b222, b122, b211 except for the  $l$ -th surrogate pair rather than the original signal pair;

- “b111surr\_threshold” (resp. 222, 122, 211): a two-dimensional array whose  $(i, j)$ -entry is the surrogate threshold for the frequency pair represented by the  $(i, j)$ -entry of b111 (resp. b222, b122, b211);

and if points in frequency-frequency space were added to the list in the “Select Frequency” box *and* one of the four bispectra  $b_{uvw}$  was selected (from the small pull-down list) at the time of saving, then also:

- “selected\_points”: a single-column cell whose  $i$ -th entry is the  $i$ -th frequency pair  $(f_x^{(i)}, f_y^{(i)})$  in the list, written simply as the frequency  $f_x^{(i)}$  on the horizontal axis followed by the frequency  $f_y^{(i)}$  on the vertical axis (both in Hz) separated by a comma;
- “selected\_plot”: uvv (either 111, 222, 122 or 211), as selected at the time of saving;
- “biamp”: a single-row cell whose  $i$ -th entry is a row vector containing the evolution of the instantaneous biamplitude for the  $i$ -th frequency pair; the  $j$ -th entry of this row vector is the biamplitude  $A_{uvw}(t_j, f_x^{(i)}, f_y^{(i)})$  at the  $j$ -th time in the discretised time axis;
- “biphase”: the same for the *unwrapped* instantaneous biphase  $\phi_{uvw}$  as shown in the Biphase against Time plot.

If you click “Save as .csv” and save with a file name, say, “Results”, then in the folder where you selected to save the results, MODA will save a folder with the same file name “Results”. This folder contains the following:

- a file “params.csv” if no frequency pairs were added to the list in the “Select Frequency” box: this includes the date on which the results were saved, sampling frequency, minimum and maximum frequencies, frequency resolution parameter  $f_r$ , whether pre-processing and Cut Edges are on or off, and the start time and end time of the signal (in seconds);
- if frequency pairs were added, then a file “params\_biphase\_biamp.csv”: this contains all the above, plus
  - the selected plot uvv, as in the “selected\_plot” field in the .mat file;
  - the points added to the frequency list (“Selected point 1”, “Selected point 2”, etc.), with the frequencies on the horizontal axis in one column and the frequencies on the vertical axis adjacently in the next column;
  - the discretised time axis, together with the instantaneous biamplitudes and instantaneous biphases for the selected points.
- four files Bispectrum\_uvv.csv (u,v=1,2): shows the same as the “buvv” field in the .mat file, except also with the frequency axes written out explicitly:

	A	B	C	D	E	F
1 Freq		0.00686	0.007164	0.007481	0.007813	0.008158
2	0.00686	4.30E-06	7.64E-06	1.06E-05	1.34E-05	1.57E-05
3	0.007164	NaN	1.32E-05	1.79E-05	2.20E-05	2.49E-05
4	0.007481	NaN	NaN	2.30E-05	2.72E-05	2.96E-05
5	0.007813	NaN	NaN	NaN	2.90E-05	2.99E-05
6	0.008158	NaN	NaN	NaN	NaN	2.55E-05

- if surrogates were computed and “Surrogates plot” was on, then also four files Bispectrum\_uvv\_surr.csv (u,v=1,2): shows the same as the “buvvsurr\_threshold” field in the .mat file, except also with the frequency axes written out explicitly.

If only one signal was analysed, then all is the same except with all references to Signal 2 removed.

# Dynamical Bayesian Inference

In Dynamical Bayesian Inference, you can either investigate couplings between oscillatory components of two different signals, or investigate couplings between two oscillatory components within the same signal. In either case, the analysis can be performed for multiple signals or signal pairs at once, so long as the different pairs of oscillations are all extracted from within the same pair of frequency bands.

*As an example for the user, files load\_first.mat and load\_second.mat have been provided: load\_first.mat is the saved result of ridge-extraction for the first three signals in 6signals\_10Hz.mat within a certain frequency interval, and load\_second.mat is the saved result of ridge-extraction for the last three signals in 6signals\_10Hz.mat within another frequency interval (largely overlapping with the first). These are to be loaded consecutively as described for “case (iii)” below.*

A collection of signals to be analysed in Dynamical Bayesian Inference must be one of the following:

- (i) a set of  $N = 2\tilde{N}$  (simultaneously recorded) signals  $x_1(t), \dots, x_{\tilde{N}}(t), x_{\tilde{N}+1}(t), \dots, x_N(t)$  of equal duration and sampling frequency, where coupling will be investigated between an oscillatory component of  $x_i$  and an oscillatory component of  $x_{\tilde{N}+i}$  for each  $1 \leq i \leq \tilde{N}$ ;
- (ii) a set of  $N$  phase signals  $\phi_1^1(t), \phi_2^1(t), \dots, \phi_N^1(t)$ , extracted within some frequency band  $[a_1, b_1]$  from some signals  $x_1(t), \dots, x_N(t)$  respectively (where these signals  $x_i$  of equal duration and sampling frequency); and also another set of  $N$  phase signals  $\phi_1^2(t), \phi_2^2(t), \dots, \phi_N^2(t)$  extracted within another frequency band  $[a_2, b_2]$  from the same signals  $x_1(t), \dots, x_N(t)$  respectively; coupling is to be investigated between  $\phi_i^1$  and  $\phi_i^2$ ;
- (iii) a set of  $N$  phase signals  $\phi_1^1(t), \phi_2^1(t), \dots, \phi_N^1(t)$ , extracted within some frequency band  $[a_1, b_1]$  from some signals  $x_1(t), \dots, x_N(t)$  respectively (where these signals  $x_i$  of equal duration and sampling frequency); and also a set of  $N$  phase signals  $\phi_1^2(t), \phi_2^2(t), \dots, \phi_N^2(t)$  extracted within another frequency band  $[a_2, b_2]$  from other signals  $y_1(t), \dots, y_N(t)$  respectively (which are of the same duration and sampling frequency as the signals  $x_i(t)$ ); coupling is again to be investigated between  $\phi_i^1$  and  $\phi_i^2$ .

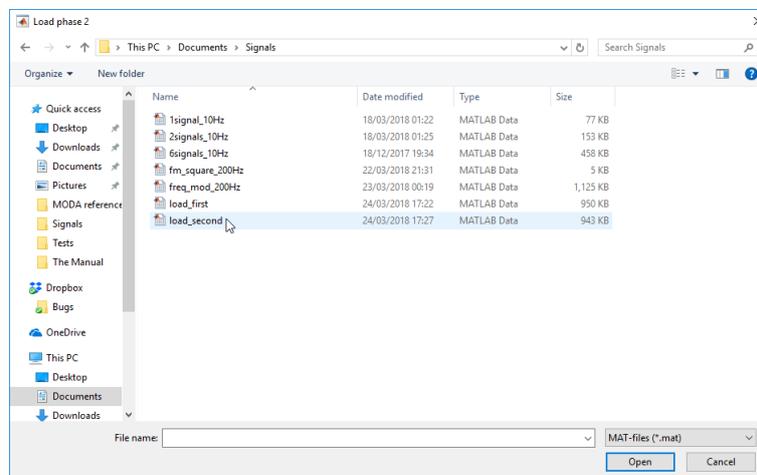
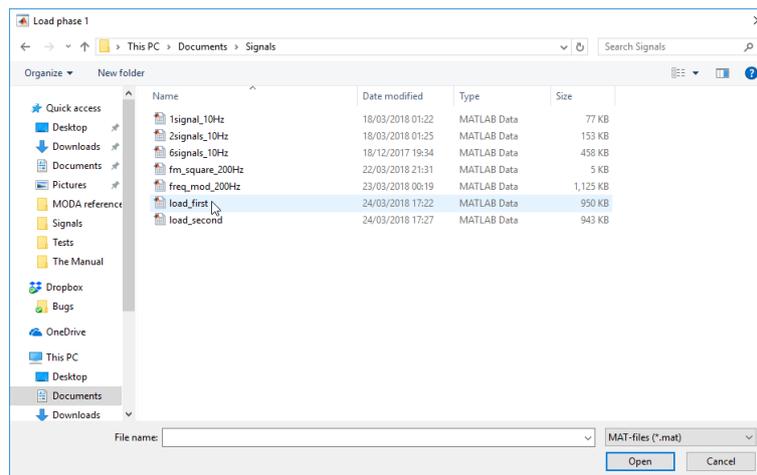
In case (i), the user will have to specify frequency ranges from within which to extract the oscillatory components of interest; MODA then applies the Butterworth filter and extracts the phases, using exactly the same algorithms as in the Ridge Extraction & Filtering application. However, there is no plot of the time-frequency representation, and the frequency ranges must be entered manually. (The maximum of the frequency ranges is required to be less than half of the sampling frequency.) Therefore, it is advised that the user follow approach (iii) instead, by using Ridge Extraction & Filtering to select frequency ranges and extract phases.

For case (i), the  $N$  signals  $x_1(t), \dots, x_N(t)$  must be stored as a single file as usual, and are

loaded by clicking on “File” and then “Load time series”. The user will have to specify the sampling frequency and orientation, as usual. As with Wavelet Phase Coherence, the  $i$ -th and  $(\tilde{N} + i)$ -th signal in the file are taken as a pair.

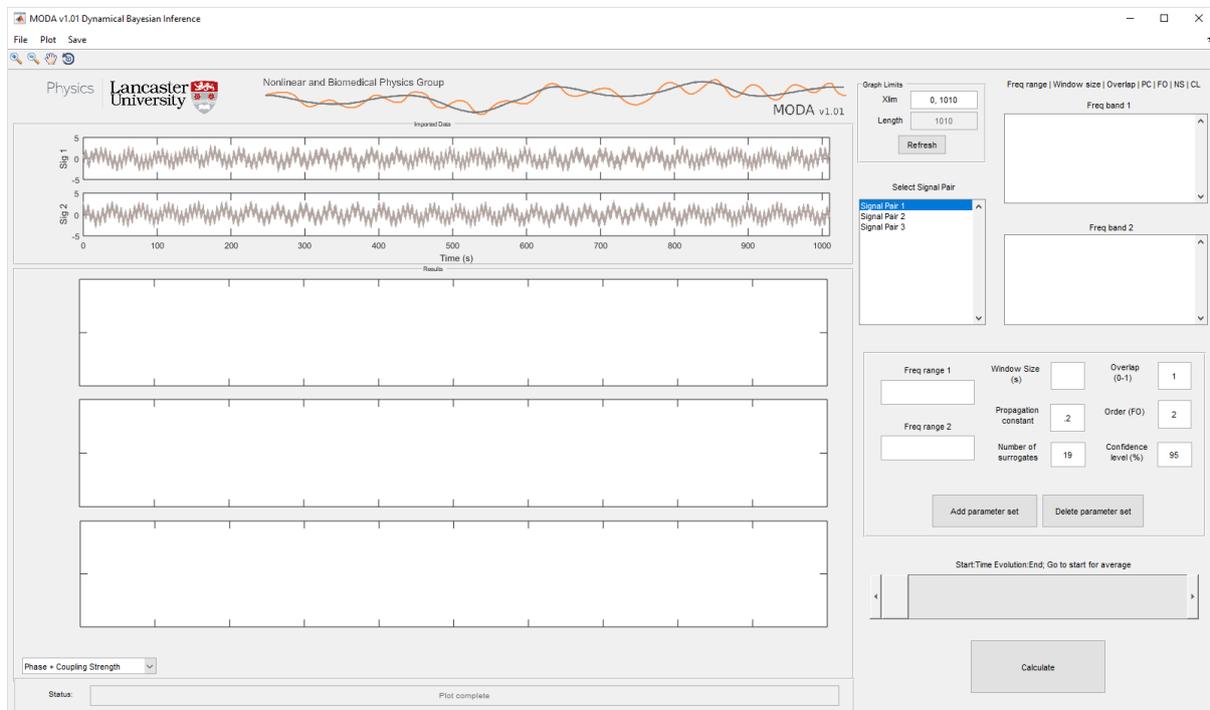
For case (ii), the pre-requisite is that the user has loaded the set of  $N$  signals  $x_1(t), \dots, x_N(t)$  into the Ridge Extraction & Frequency application, and has applied either filtering or ridge-extraction to the collection of signals in *exactly two* frequency bands, and has saved the results as a .mat file. In this case, the extracted phases are loaded by clicking on “File” and then “Load filtered phases (1 sig, 2 bands)”, and then choosing the .mat file where the results are saved.

For case (iii), the pre-requisite is that the user has (a) loaded the set of  $N$  signals  $x_1(t), \dots, x_N(t)$  into the Ridge Extraction & Frequency application, and has applied either filtering or ridge-extraction to the collection of signals in *exactly one* frequency band, and has saved the results as a .mat file, and also (b) has loaded the set of  $N$  signals  $y_1(t), \dots, y_N(t)$  into the Ridge Extraction & Filtering application, and has applied either filtering or ridge-extraction to this collection of signals *again in exactly one frequency band*, and has saved the results as another .mat file. In this case, the extracted phases are loaded by clicking on “File” and then “Load filtered phases (2 sigs, 1 bands)”; choose (in the normal way by either double-clicking or clicking and selecting the “Open” button) the .mat file containing the results for  $x_1(t), \dots, x_N(t)$ ; and then, in the next browser window that automatically opens, choose the .mat file containing the results for  $y_1(t), \dots, y_N(t)$ .



## Case (i)

When you have loaded the file, you will see the  $\tilde{N}$  signal pairs listed in the “Select Signal Pair” box, with the currently selected signal pair shown in the “Imported Data” box; for Signal Pair  $i$ , the upper signal “Sig 1” in the “Imported Data” box is the  $i$ -th signal in the loaded file, and the lower signal “Sig 2” is the  $(\tilde{N} + i)$ -th signal in the loaded file.



If you wish to have the graph of one of these pairs of signals plotted in a separate window (as a savable file), select the signal pair that you want, and then click on “Plot” (to the right of “File”) and click on the option “Time Series”.

You must then fill in the fields in the box for adding parameter sets:

Freq range 1	Window Size (s)	125	Overlap (0-1)	1
0.081,0.3	Propagation constant	.2	Order (FO)	2
Freq range 2	Number of surrogates	19	Confidence level (%)	95
0.08,0.31				
Add parameter set		Delete parameter set		

It does not matter which signal is selected when filling these in, as the computation will be performed for all the signal pairs.

- In the field “Freq range 1”, enter the frequency interval (in Hz) from which you intend to extract the oscillation  $\phi_i^1$  from the upper signal  $x_i$  in the “Imported Data” box; and in the field “Freq range 2”, enter the frequency interval (in Hz) from which you intend

to extract the oscillation  $\phi_i^2$  from the lower signal  $x_{\tilde{N}+i}$  in the “Imported Data” box. Each frequency interval is entered simply as the lower boundary frequency and the upper boundary frequency, separated by a comma; so for example, the frequency interval from 1.3 Hz to 2 Hz is entered as “1.3,2”.

- In the field “Window Size”, enter the width  $\delta$  of the time-windows  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  over which the approximate coupling function is to be computed. **It is strongly advised that the window size is at least 10 cycles of the lowest frequency across the two frequency ranges;** that is to say, if the frequency ranges are  $[a_1, b_1]$  and  $[a_2, b_2]$ , then the window size should be at least  $10/\min(a_1, a_2)$ .
- The “Overlap” field specifies what proportion of each window does *not* overlap with the previous window; so 1 represents no overlap between consecutive windows, while a number close to 0 represents high overlap. (Of course, you should not enter 0 itself.)
- The propagation constant is a nonnegative real number as described in the Overview of Dynamical Bayesian Inference. A larger value means that the computation for each new window is more independent of the results for the previous window.
- Enter into the “Order” field the desired Fourier order  $K$  as described in the Overview of Dynamical Bayesian Inference. This is a positive integer, for which a larger value implies a greater number of degrees of freedom possible for the inferred coupling functions. More precisely, given a Fourier order  $K$ , the inferred coupling functions are taken from within the span of the  $(2K + 1)^2$  most fundamental Fourier basis functions.
- The user also needs to enter a number of surrogates for surrogate testing, together with the “confidence level” defined as  $1 - \alpha$ ; so for a test with significance level  $\alpha = 0.05$ , enter 95 into the “Confidence level (%)” field.

Then click the “Add parameter set” button, and the entire set of details will appear in the boxes titled “Freq band 1” and “Freq band 2” (with all the details other than the second frequency range shown in the upper box, and all the details other than the first frequency range shown in the lower box):

The screenshot shows a software interface for configuring parameters. It includes a 'Graph Limits' section with 'Xlim' (0, 1010) and 'Length' (1010) fields, a 'Refresh' button, and a 'Select Signal Pair' dropdown menu. The main configuration area is titled 'Freq range | Window size | Overlap | PC | FO | NS | CL' and contains two frequency bands: 'Freq band 1' (0.081, 0.300 | 125.00 | 1.00 | 0.20 | 2 | 19 | 95) and 'Freq band 2' (0.080, 0.310 | 125.00 | 1.00 | 0.20 | 2 | 19 | 95). Below this is a summary table with the following values:

Freq range 1	Window Size (s)	Overlap (0-1)
0.081, 0.3	125	1
Freq range 2	Propagation constant	Order (FO)
0.08, 0.31	.2	2
	Number of surrogates	Confidence level (%)
	19	95

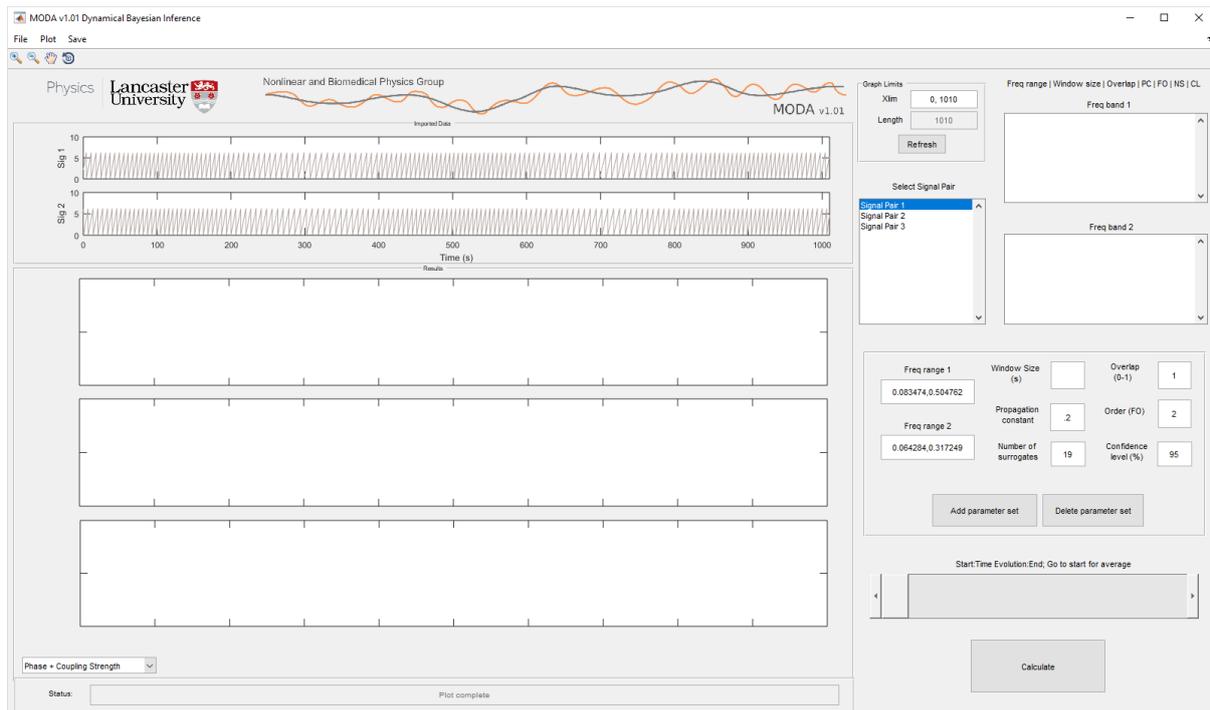
At the bottom, there are 'Add parameter set' and 'Delete parameter set' buttons.

Of course, the user may wish to investigate other pairs of frequency bands, and/or other parameters from the list. To do this, just fill in the fields again with the new set of parameters, and click “Add parameter set” again. An added parameter set can be deleted by clicking on it in either the “Freq band 1” box or in the “Freq band 2” box (so that it is then highlighted in both of these boxes), and clicking “Delete parameter set”. Once you have added all the desired parameter sets, click “Calculate” to calculate results (described below).

### Cases (ii) and (iii)

When you have loaded the file(s) [1 file in case (ii), 2 files in case (iii)], you will see the  $N$  pairs of phase signals listed in the “Select Signal Pair” box, with the currently selected signal pair shown in the “Imported Data” box:

- In case (ii), for Signal Pair  $i$ , the upper signal “Sig 1” is the phases that were extracted from signal  $i$  in the Ridge Extraction & Filtering application within the first frequency range that was shown in the “Interval List” box; and the lower signal “Sig 2” is the phases that were extracted from signal  $i$  in the Ridge Extraction & Filtering application within the second frequency range that was shown in the “Interval List” box.
- In case (iii), for Signal Pair  $i$ , the upper signal “Sig 1” is the extracted phases of signal  $i$  in the first of the two files selected for loading into the Dynamical Bayesian Inference application, and the lower signal “Sig 2” is the extracted phases of signal  $i$  in the second of the two files selected for loading into the Dynamical Bayesian Inference application.



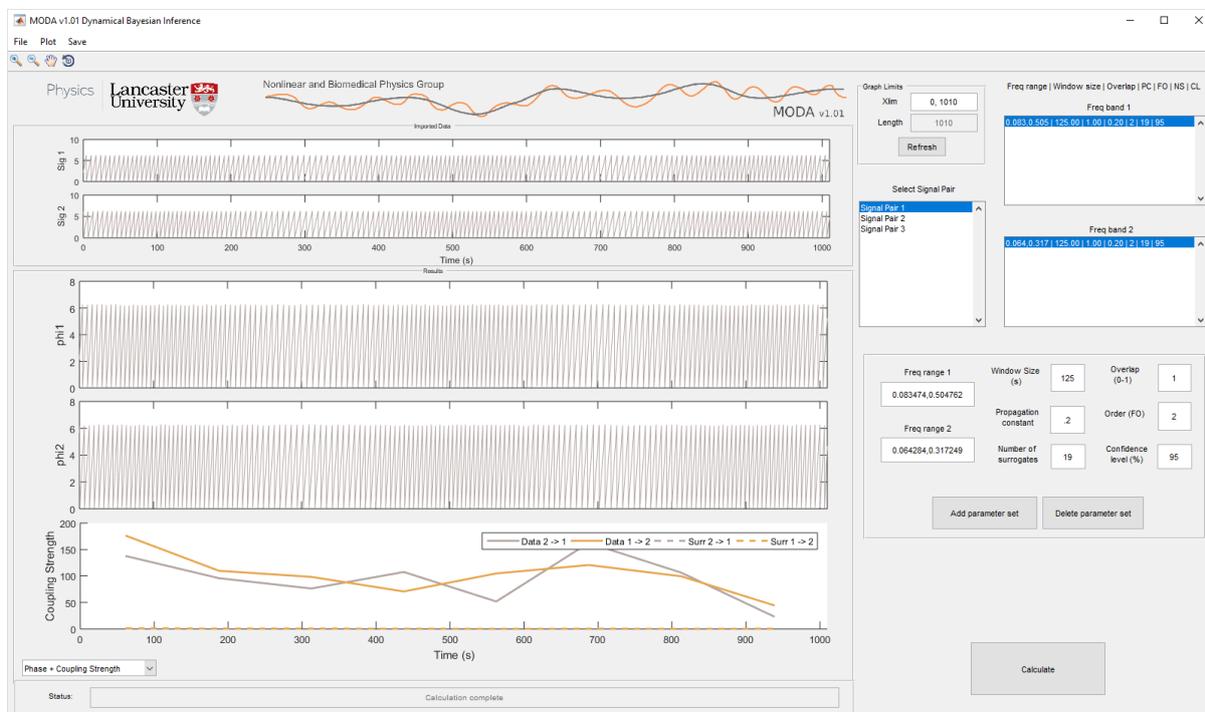
In either case, the frequency bands are automatically displayed in the “Freq range 1” and “Freq range 2” boxes. Nonetheless, the remaining fields for the parameters need to be filled in, just as in case (i). Once again, the recommended window size is at least 10 cycles of the lowest frequency across the two ranges. Multiple parameter sets can be added, but the frequency ranges must not be changed. When all desired parameter sets have been added, click “Calculate” to calculate results.

## Results

If, as will typically be the case, the window size and overlap are such that the end of the signal occurs part-way through a window, then the Bayesian inference will only be performed on all the possible complete windows.

When you click “Calculate”, in the “Results” box you will see, for the currently selected signal pair and parameter set, the following:

- the phases  $\phi^1(t)$  and  $\phi^2(t)$ ; in cases (ii) and (iii), these are exactly the same as shown in the “Imported Data” box, and in case (i) they are the extracted phases from the signals filtered in the frequency ranges as currently selected;
- the coupling strengths  $\epsilon_{2 \rightarrow 1}$  and  $\epsilon_{1 \rightarrow 2}$  as well as the surrogate thresholds for each of these; the values are shown at the midpoints  $t_n$  of the windows  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$ , and then joined up by a piecewise linear curve.



When you click on a different signal pair in the “Select Signal Pair” box, or on a different parameter set (from within either the “Freq band 1” or “Freq band 2” box), the results will change accordingly.

The surrogate type used here by MODA is **CPP surrogates**. The surrogate test in MODA involves surrogating the entire signal  $\phi^1$  and the entire signal  $\phi^2$ , and then applying the dynamical Bayesian inference algorithm with the parameters selected to the surrogate pairs (as in Version II in the section on Surrogates) in order to determine the surrogate threshold within each window. The number of surrogates for each signal is as was entered into the “Number of surrogates” field.

**Practical remark.** If there are some windows during which there is coupling from  $\phi^j$  to  $\phi^i$ , then the CPP surrogates will destroy the dependence of  $q_i$  on  $\phi^j$ , but it might still preserve

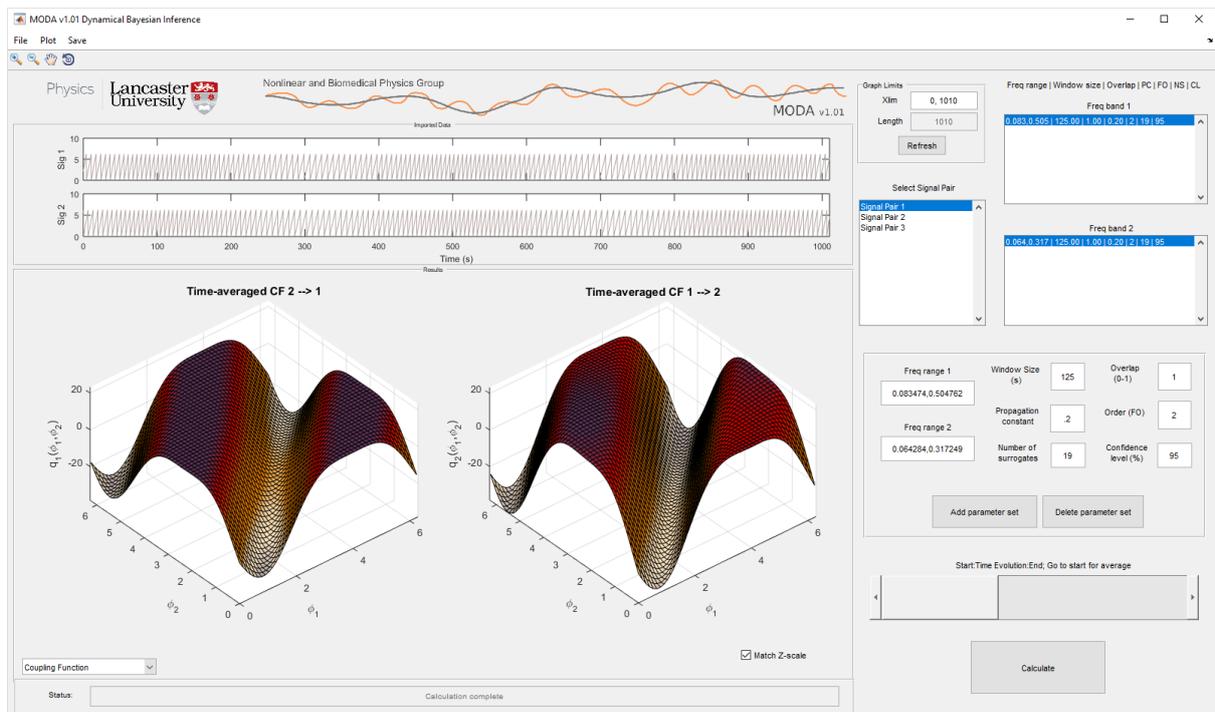
some dependence of  $q_i$  on  $\phi^i$ , and so the surrogate values of  $\epsilon_{j \rightarrow i}$  might still be quite high. Consequently (as stated in the Overview of Dynamical Bayesian Inference), one must not assume that if there are some windows in which  $\epsilon_{j \rightarrow i}$  is found to be significant and others where it is not, then the windows where it is not contain no important influence of  $\phi^j$  upon  $\phi^i$ .

One can plot as a savable figure the time-series shown in the “Imported Data” box, the phases shown in the “Results” box, the coupling strengths and surrogate thresholds shown in the “Results” box, or a combination of these, by clicking “Plot” and selecting the suitable option.

For each signal pair and parameter set, one can **view the inferred coupling functions themselves**. To do this, click on the small pull-down list in the bottom left corner of the “Results” box (which currently shows “Phase + Coupling Strength”), and select “Coupling Function”. What you will first see are the **time-averaged coupling functions**

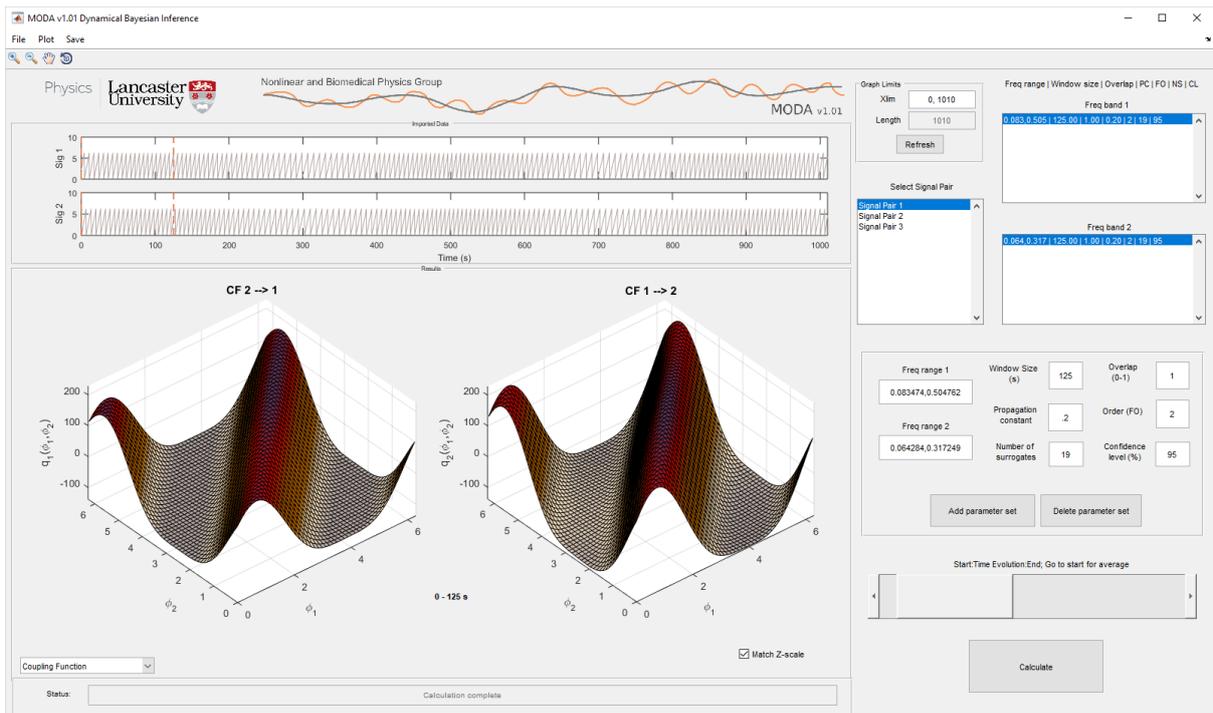
$$q_i^{\text{mean}}(\phi^1, \phi^2) = \frac{1}{M} \sum_{n=1}^M q_i^{t_n}(\phi^1, \phi^2)$$

where  $M$  is the number of complete windows (with the specified overlap) that can fit inside the duration of the signals.

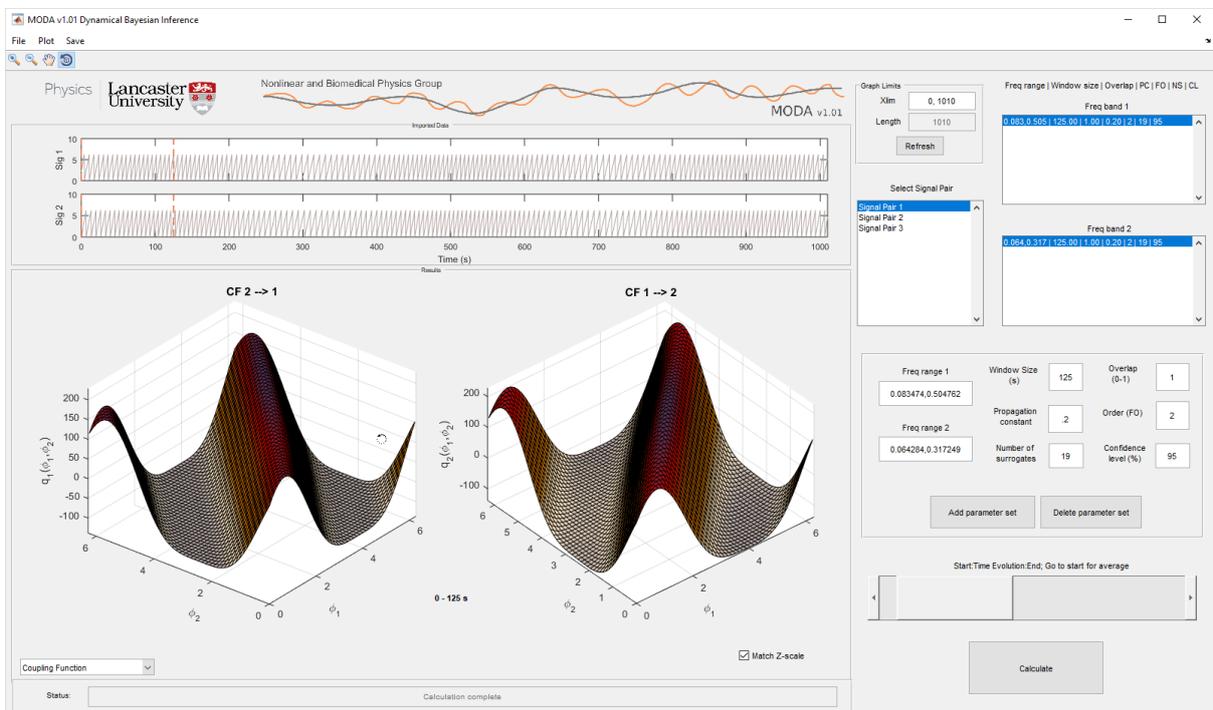


By **moving the slider** on the bottom right beneath the “Calculate” button, you can then **see the coupling functions  $q_i^{t_n}$  themselves**. To move the slider, you can either: (a) click on the slider and then press the left and right arrow keys on the keyboard—or alternatively click the left and right arrow buttons surrounding the slider bar—to see the evolution of the coupling functions; or (b) click on the slider and drag with the mouse to wherever along the signal you would like to see the coupling functions.

As you move the slider, the current time-window  $[t_n - \frac{1}{2}\delta, t_n + \frac{1}{2}\delta]$  is marked on the original time-series in the “Imported Data” box.



Since the graphs are three-dimensional, one can rotate them by clicking on the “Rotate 3D” icon in the top-left corner of the window, immediately to the right of the hand and the magnifying glasses. With this, one can click on either the left graph or the right graph, and hold the mouse button and drag to rotate the graph.



To return to the originally shown orientation, just double click on the graph. To return the cursor to its normal functioning, just click again on the “Rotate 3D” icon.

The two graphs are shown with exactly the same height on the vertical axis, so that the values of the coupling functions can be compared by eye; this is indicated by the tick in the **Match Z-scale** field in the bottom right corner of the “Results” box. However, the user can untick this box, and then the vertical scale on each plot will be adjusted to the range of values that the coupling function takes in each plot.

One can plot as a savable figure whatever is currently shown in the “Results” box by clicking “Plot” and then “Coupling Function”. One can plot the original time-series with the current time-window marked, as shown in the “Imported Data” box, by clicking “Plot” and then “Time-series”. It is also possible to **save a video of the evolving coupling function** by clicking “Save” and then “Save coupling video”. The saved video either matches the vertical scales of the two graphs or not, depending on whether the “Match Z-scale” field is ticked or not.

One can go back to seeing the phases and coupling strengths by selecting “Phase + Coupling Strength” in the small pull-down list in the bottom left corner of the “Results” box.

## Saving numerical values of results

Suppose the session involved analysing  $N$  signal pairs, with  $r$  added parameter sets.

After computing results, if you click “Save” and then “Save as .mat”, the resulting .mat file consists of a single  $1 \times 1$  MATLAB structure labelled Bayes\_data, containing the following fields:

- “time”: the complete discretised time axis (in seconds), as a row vector;
- “sampling\_freq”: the sampling frequency (in Hz) either entered as the signals were loaded into Dynamical Bayesian Inference, or entered when loading into Ridge Extraction & Filtering the signal(s) from which phases were extracted;
- “interval1”: if already extracted phase signals were loaded (“cases (ii) and (iii)”), then this shows exactly what was in the “Freq range 1” field, which is the same as what is at the start of every added parameter set as shown in the “Freq band 1” box; if the original signals were loaded directly (“case (i)”), then this is an  $r \times 2$  array where the  $j$ -th row is the lower boundary frequency and the upper boundary frequency (both in Hz) as shown in the “Freq band 1” box for the  $i$ -th added parameter set;
- “interval2”: the same, but for “Freq band 2”;
- “phase1”: an  $N \times r$  cell whose entries are row vectors; if the original signals were loaded directly (“case (i)”), then the  $(i, j)$ -entry is the phase signal (wrapped between  $-\pi$  and  $\pi$ ) obtained by filtering the first signal in  $i$ -th Signal Pair  $i$  within the  $j$ -th frequency interval; if already extracted phase signals were loaded (“cases (ii) and (iii)”), then the columns of the cell are identical, with the  $i$ -th entry of each column being first phase signal in Signal Pair  $i$ ; in either case, the  $l$ -th entry of the row vector corresponds to the value of the phase signal at the  $l$ -th time in the discretised time axis;
- “phase2”: the same, except in each case, for the second signal in the Signal Pair, rather than the first;
- “Bayes\_win”: a column vector of length  $r$ , whose  $i$ -th entry is the window size (in seconds) in the  $i$ -th parameter set;

- “overlap”: a column vector of length  $r$ , whose  $i$ -th entry is the overlap value in the  $i$ -th parameter set;
- “propagation\_const”: a column vector of length  $r$ , whose  $i$ -th entry is the propagation constant in the  $i$ -th parameter set;
- “Fourier\_base”: a column vector of length  $r$ , whose  $i$ -th entry is the Fourier order in the  $i$ -th parameter set;
- “surrnum”: a column vector of length  $r$ , whose  $i$ -th entry is the surrogate count in the  $i$ -th parameter set;
- “confidence\_level”: a column vector of length  $r$ , whose  $i$ -th entry is the confidence level in the  $i$ -th parameter set;
- “Bayestime”: an  $N \times r$  cell whose rows are identical; the  $j$ -th entry of each row is a row vector listing (in seconds) where along the time axis the centres of the windows lie, as determined by the window size and overlap in the  $j$ -th parameter set;
- “coupling\_strength\_2to1”: an  $N \times r$  cell whose  $(i, j)$ -entry is a row vector listing the coupling strengths  $\epsilon_{2 \rightarrow 1}$  over the time-windows, for the  $i$ -th Signal Pair in the  $j$ -th parameter set; the  $l$ -th entry of this row vector is the coupling strength over the window whose centre occurs at the time shown in the  $l$ -th entry of the row vector occupying the  $j$ -th entry of each row of the cell Bayestime;
- “surrogate\_coupling\_strength\_2to1”: the same, except showing the surrogate thresholds for the coupling strengths, rather than the actual coupling strengths;
- “coupling\_strength\_1to2” and “surrogate\_coupling\_strength\_1to2” are analogous, for  $\epsilon_{1 \rightarrow 2}$  rather than  $\epsilon_{2 \rightarrow 1}$ ;
- “coupling\_function\_2to1”: an  $N \times r$  cell whose  $(i, j)$ -entry is a  $49 \times 49 \times M_j$  array where  $M_j$  is the number of windows according to the  $j$ -th parameter set (i.e. the length of the row vector occupying the  $j$ -th entry of each row of the cell Bayestime); the  $(u, v, n)$ -entry of this array is the value  $q_1^{tn}(0.13(u-1), 0.13(v-1))$  for Signal Pair  $i$  under the  $j$ -th parameter set;
- “coupling\_function\_1to2”: an  $N \times r$  cell whose  $(i, j)$ -entry is a  $49 \times 49 \times M_j$  array where  $M_j$  is as above; the  $(u, v, n)$ -entry of this array is the value  $q_2^{tn}(0.13(u-1), 0.13(v-1))$  for Signal Pair  $i$  under the  $j$ -th parameter set;
- “mean\_cfunc\_2to1”: an  $N \times r$  cell whose  $(i, j)$ -entry is a  $49 \times 49$  array; the  $(u, v)$ -entry of this array is the value  $q_1^{\text{mean}}(0.13(u-1), 0.13(v-1))$  for Signal Pair  $i$  under the  $j$ -th parameter set;
- “mean\_cfunc\_1to2”: an  $N \times r$  cell whose  $(i, j)$ -entry is a  $49 \times 49$  array; the  $(u, v)$ -entry of this array is the value  $q_2^{\text{mean}}(0.13(u-1), 0.13(v-1))$  for Signal Pair  $i$  under the  $j$ -th parameter set.

If you click “Save as .csv” and save with a file name, say, “Results”, then in the folder where you selected to save the results, MODA will save a folder with the same file name “Results”. This contains  $(N + 1)r$  CSV files: “Coupling\_strength $j$ ”,  $j = 1, \dots, r$ , and “Mean.CF  $j$  - Subject  $i$ ”,  $i = 1, \dots, N$ ,  $j = 1, \dots, r$ .

The file “Coupling\_strength $j$ .csv” contains:

- the date on which the results were saved;
- the sampling frequency, the start and end time of the signals (in seconds), and the  $j$ -th added parameter set (with each of the two frequency intervals written as the lower boundary frequency and the upper boundary frequency, separated by spaces);
- the list of centre points of the time windows (as in the Bayestime field in the .mat file), together with the associated coupling strengths (under headings “Subject  $i$  - 2 to 1” or “Subject  $i$  - 1 to 2”, where Subject  $i$  refers to the  $i$ -th pair of phase signals in the “Select Signal Pair” box) and surrogate thresholds (under headings ““Surrogate  $i$  - 2 to 1” or “Surrogate  $i$  - 1 to 2””), all as computed under the  $j$ -th parameter set.

The file “Mean\_CF  $j$  - Subject  $i$ .csv” is a  $50 \times 100$  array: The first column is an empty cell followed by the discretised phase axis 0, 0.13, 0.24,  $\dots$ , 6.24; likewise, the 2nd to 50th entries of the first row are the same discretised phase axis. Accordingly, for  $2 \leq u, v \leq 50$ , the  $(u, v)$ -entry is the value  $q_1^{\text{mean}}(0.13(u-2), 0.13(v-2))$  for Signal Pair  $i$  under the  $j$ -th parameter set. The 51st column is a column of zeros. The 52nd to 100th entries of the first row are the discretised phase axis once again; accordingly, for  $2 \leq u \leq 50$  and  $52 \leq v \leq 100$ , the  $(u, v)$ -entry is the value  $q_2^{\text{mean}}(0.13(u-2), 0.13(v-52))$  for Signal Pair  $i$  under the  $j$ -th parameter set.

# Bibliography

- [1] Andrea Duggento, Tomislav Stankovski, Peter V. E. McClintock, and Aneta Stefanovska. Dynamical bayesian inference of time-evolving interactions: From a pair of coupled oscillators to networks of oscillators. *Phys. Rev. E*, 86:061126, 2012. doi:10.1103/PhysRevE.86.061126.
- [2] Dmytro Iatsenko, Peter V. E. McClintock, and Aneta Stefanovska. Linear and synchrosqueezed time-frequency representations revisited: overview, standards of use, resolution, reconstruction, concentration, and algorithms. *Digital Signal Processing*, 42:1–26, 2015. doi:10.1016/j.dsp.2015.03.004.
- [3] Dmytro Iatsenko, Peter V. E. McClintock, and Aneta Stefanovska. Extraction of instantaneous frequencies from ridges in timefrequency representations of signals. *Signal Processing*, 125:290–303, 2016. doi:10.1016/j.sigpro.2016.01.024.
- [4] Gemma Lancaster, Dmytro Iatsenko, Aleksandra Pidde, Valentina Ticcinelli, and Aneta Stefanovska. Surrogate data for hypothesis testing of physical systems. *Physics Reports*, 748:1 – 60, 2018. Surrogate data for hypothesis testing of physical systems. doi:10.1016/j.physrep.2018.06.001.