NeuroSpec 2.1 User Guide

David M. Halliday

Contents

1 Introduction

This guide is for the NeuroSpec statistical signal processing archive version 2.1. NeuroSpec 2.1 is an extension to NeuroSpec 2.0 that includes routines for estimation of non parametric directionality analysis for bivariate time series and/or point process data, and plotting of the results. It includes functions to calculate a set of scalar metrics which decompose the total product moment correlation coefficient summatively into three terms by direction and a set of functions which decompose the coherence summatively at each frequency into three terms by direction: forward direction, reverse direction and instantaneous (zero lag) interaction. Four categories of routine are included in the software:

1. Analysis routines for directionality analysis of two channels of data.
2. Routine for calculating R^2 metrics over reduced frequency range.
3. Plotting routines.
4. Demonstration scripts.

2 Software requirements

These routines require the MATLAB system to be installed. Version 7.x or higher is required to run NeuroSpec 2.1. They have been tested using PC versions 7.1 and 8.1. No additional toolboxes are required, only the base system is needed. If you have an older version of MATLAB please read the FAQ page on the NeuroSpec web site.

3 Installation, quick start guide & getting help

To install the archive:

1. Download and Unzip the file `neurospec21.zip`.
2. Install the directory `neurospec21` in your MATLAB path. See the MATLAB Help system for details of how to do this (In MATLAB 8.x the Set Path option is accessed from the HOME tab).
3. Change the MATLAB working directory to folder `neurospec21_demos` and run the demonstration scripts, starting with `R2_cn_demo1`.

Once installed, additional help is available via the MATLAB help system. The command `helpwin neurospec21` should display a hyperlinked list of files, similar to that in Appendix A. Help for individual commands is available using the `help` or `helpwin` commands followed by the function name. The MATLAB source for the demonstration scripts in directory `neurospec21_demos` should provide additional guidance in the use of the software.

4 Format of raw data in MATLAB

This section describes the format that raw data should be input to the NeuroSpec routines. NeuroSpec can process either time series or point process (spike train) data. The formats specified below should readily accommodate most types of experimental and/or simulated data. We adopt the MATLAB convention of specifying multi-channel data in column format. *Note*: Failure to specify input data in this format may give incorrect results.

4.1 Time series data

Times series data in NeuroSpec 2.1 should be in the same format as in NeuroSpec 2.0. Readers familiar with NeuroSpec 2.0 can skip to point-process data in section 4.2, which is handled differently in NeuroSpec 2.1. Time series data is assumed to represent regularly sampled data with a known sampling rate. Integer or real values can be processed. Time series data should be in column format. Each column should represent one signal, with one sample per row. The sampling rate is input to the analysis - this, in conjunction with the FFT segment length, defines the spacing of the Fourier frequencies returned after analysis. Both single column and 2D matrices can be processed. Each column in a 2D matrix should contain data from one signal. Each row in a 2D matrix should contain data samples at the same time instant from several channels. See the demonstration script `R2_ts_demo1` for an example of processing matrix data. Two single columns of data passed to a NeuroSpec core routine should have the same number of data points in each column and be from simultaneously sampled signals.

4.2 Point process (spike train) data

The handling of Point Process (Spike Train) data in NeuroSpec 2.1 differs from that in NeuroSpec 2.0. Spike train data should be in column format where spike timings are mapped into a 0/1 time series. This format represents spike trains as a regularly sampled sequence where the occurrence of a spike in a particular time bin is represented by the value 1, all other time bins have the value 0.

4.2.1 Conversion of point process data from NeuroSpec 2.0 format

The conversion of spike train data from that used for the routines in NeuroSpec 2.0 is straightforward, and can be illustrated by an example. If a vector of integer spike times is contained in the MATLAB variable: `sp1`, and assuming the spike train has duration: `sec_tot` (seconds), with spike times specified as an integer multiple of a sampling rate: `rate` (samples/sec), then the following MATLAB commands will generate a vector `dat_sp1` which can be used in the NeuroSpec 2.1 routines.

```
dat_sp1=zeros(sec_tot*rate,1);

dat_sp1(sp1)=1;
```

For an example of working with spike train data see the demonstration script `R2_cn_demo1.m`.

5 Using the NeuroSpec 2.1 analysis routines

Two functions are provided to undertake the bivariate directionality analysis.

1. `sp2a2_R2` Self contained function.
2. `sp2a_R2_mt` Function with additional options, requires functions `sp2_fn2_R2a` and `sp2_fnR2b`

The analysis implemented in these two functions is equivalent to what is referred to as Type 0 analysis in NeuroSpec 2.0. The bivariate data is split into L non overlapping segments, each containing T data points, analysed with the DFT segment length, T. The total number of samples analysed is $R = LT$. Only complete segments are analysed, data points at the end of the record that do not make a complete segment are *not* included in the analysis. The segment length T is specified as an input argument to the above functions routines, as a power of 2. The routines assume that the first channel of data is the reference (input) signal, and the second channel of data is the response (output) signal. This is important for directionality estimates. The additional data segmentation strategies in NeuroSpec 2.0 (referred to as Type 1 and Type 2 analysis) are not supported in NeuroSpec 2.1.

The use of the two routines can be illustrated by an example. Suppose you have in MATLAB the following variables:

`x, y` Two time series (or 0/1 spike train signals) of equal length.

Suppose the sampling rate for all signals is 1000/sec, in MATLAB: `rate=1000`. Define the segment length for the FFT as 2^{10} (1024 points), in MATLAB: `seg_pwr=10`. The segment length in NeuroSpec is specified as a power of 2.

5.1 Analysis with sp2a2_R2

For directionality analysis between the signals x and y use the command:

```
[f,t,cl]=sp2a2_R2(x,y,rate,seg_pwr);
```

This assumes that the signal in x is the input (or reference) signal and the signal in y is the output signal. Thus the forward directionality measures refer to the interaction $x \to y$ and the reverse directionality measures refer to the interaction $x \leftarrow y$, following the convention in [1].

5.2 Analysis with sp2a2_R2_mt

For directionality analysis between the signals x and y use the command:

```
[f,t,cl]=sp2a2_R2_mt(x,y,rate,seg_pwr);
```

The convention regarding input and output signals is the same in both functions, thus x is the input (or reference) signal and y is the output signal for directionality analysis. The function sp2a2_R2_mt supports additional options, which are included by specifying an option list in a single text string, matching the approach in NeuroSpec 2.0. For example, to undertake the same analysis using a linear de-trend on both signals and using a multi-taper analysis with $NW = 2$ use the command

```
[f,t,cl]=sp2a2_R2_mt(x,y,rate,seg_pwr,'t2 M2');
```

If you wish to rectify both channels and then perform a linear de-trend prior to analysis without multi-taper windows use:

```
[f,t,cl]=sp2a2_R2_mt(x,y,rate,seg_pwr,'t2 r2');
```

For further details on the use of options, see the source file, or help entry for sp2a2_R2_mt.The order in which the options are specified is not important, however, only a single option string should be used, multiple options should be separated by spaces. Note that the 'w' and 'M' options are mutually exclusive, the first applies split cosine data windows, the second applies multi-taper (slepian) data windows. The optional fourth output parameter, sc, is included for compatibility with the NeuroSpec 2.0 comparison of spectra and spectra and pooled spectra analysis routines. See the NeuroSpec 2.0 user guide for details of these.

5.3 Using point process (spike train) data

Spike train data can be analysed using the above two routines. The format of the data is explained in section 4.2.1. Using this approach also supports analysis of hybrid (mixed time-series/point-process) data.

6 Outputs from the NeuroSpec 2.1 core routines

The outputs from the two functions have the same format, both return three variables, with an optional fourth output argument for function sp2a2_R2_mt. These outputs adopt the same format and are compatible with the outputs from the NeuroSpec 2.0 routines, although the NeuroSpec 2.1 outputs contain more columns in the matrices and more variables in the structure, which contain the results of the directionality analysis. The outputs are:

1. f A matrix of frequency domain results.
2. t A matrix of time domain results.
3. cl A structure containing confidence limits, directionality metrics and other analysis parameters.
4. sc An optional matrix of spectral coefficients (sp2a2_R2_mt only).

For further details and the format of these variables see the sp2a2*.m source files. The fourth output parameter in sp2a2_R2_mt is optional. It can be used for the comparison of spectra, comparison of coherence or pooled spectra extensions available in NeuroSpec 2.0 (see the NeuroSpec 2.0 user guide for details).

7 Directionality metrics

The directionality metrics are contained in the output structure `cl`. The specific values are contained in the four variables:

`cl.R2`	Overall R^2 value: R^2_{yx}
`cl.R2_0`	Component of R^2 at zero lag:$R^2_{yx;0}$
`cl.R2_p`	Component of R^2 in forward direction, positive lag component: $R^2_{yx;+}$
`cl.R2_n`	Component of R^2 in reverse direction, negative lag component, $R^2_{yx;-}$

7.1 Directionality metrics over a reduced frequency range - function `R2_f1`

Directionality metrics can be calculated over a reduced frequency range using the function `R2_f1`. The syntax for this function is

```
[cl_out,R2_out]=R2_f1(f,cl_in,max_freq);
```

The input arguments `f` and `cl_in` are the parameters `f` and `cl`, respectively, which are output from the analysis function `sp2a2_R2` or `sp2a2_R2_mt`. The third parameter, `max_freq`, is the upper frequency limit (in Hz) for calculation of the directionality metrics, $f\alpha$. As an illustration following the example in section 5.1 the following lines will generate R^2 metrics at frequencies up to 250 Hz.

```
[f,t,cl]=sp2a2_R2(x,y,rate,seg_pwr);
```

```
[cl_out,R2_out]=R2_f1(f,cl,250);
```

As well as the on screen message the R^2 metrics are contained in the structure output from the function `R2_f1`:

`cl_out.R2_ch_f`	R^2 at frequencies up to $f\alpha$: $R^2_{yx,f\alpha}$
`cl_out.R2_0_f`	Component of R^2 at zero lag at frequencies up to $f\alpha$:$R^2_{yx;0,f\alpha}$
`cl_out.R2_p_f`	Component of R^2 in forward direction at frequencies up to $f\alpha$: $R^2_{yx;+,f\alpha}$
`cl_out.R2_n_f`	Component of R^2 in reverse direction at frequencies up to $f\alpha$: $R^2_{yx;-,f\alpha}$

The output `R2_out` is a column vector of the above four numeric values. Additional variables containing the percentage of R^2 in each direction are also contained in the `cl_out` structure. See the source file `R2_f1.m` for further details.

8 Plotting

A single plotting function is included to plot the results from a single analysis with the two `sp2a2*` core routines. The plotting function is `psp2_R2`, which takes the parameters `f`, `t` and `cl` as input along with scalar numeric values to specify the frequency range (Hz), lag range (ms, 2 values) and coherence limit (range $0-1$). A final optional parameter is a text string used as a title for individual subplot graphs, otherwise the text string in the structure variable `cl.what` is used as a plot label. For example, assuming that `[f,t,cl]` are returned from a directionality analysis, an example usage of this function would be:

```
psp2_R2(f,t,cl,100,150,75,0.5)
```

The scalar parameters here (in order) indicate plotting at frequencies up to 100 Hz, plotting over a total of 150 ms lags, 75 ms of which are negative, and using a maximum of 0.5 on the coherence plot. For more details see the file `psp2_R2.m`. The frequency range should not exceed the nyquist frequency. The number of time domain lags should not exceed the DFT segment length. Passing values to `psp2_R2` which exceed these limits will generate an error message. The function `psp2_R2` generates 7 plots: two autospectra, two coherence, phase, cumulant density and the time domain directionality function $\rho_{yx}(\tau)$.

The two coherence plots are 1) The original coherence estimate (in black), and 2) The original coherence estimate (black) with the forward (red), reverse (blue) and zero-lag (grey) components overlaid. The two time domain plots use the same lag range for plotting.

The outputs from the NeuroSpec 2.1 `sp2a2*` routines are also compatible with the plotting function `psp2` provided in NeuroSpec 2.0, see the NeuroSpec 2.0 user guide for details. The specification of the frequency range, lag range, maximum coherence and optional text string for labelling in function `psp2_R2` is the same as used in the function `psp2` included in NeuroSpec 2.0

9 Demonstration scripts

Three demonstration scripts are included as part of NeuroSpec 2.1. These are `R2_cn_demo1`, `R2_ts_demo1` and `R2_AR4_demo1`, they are located in the directory `neurospec21_demos`.

9.1 Script `R2_cn_demo1`

This demonstrates non-parametric directionality using the simulated cortical neuron data from [1]. The spike data and associated parameters are contained in the file `JIN_dat1.mat`. The demonstration script analyses this data and generates the R^2 metrics presented in Table 1 in [1]. It further generates a set of 17 figures the first three corresponding to Figures 2, 3 and 4 in [1]. The remaining figures demonstrate the use of `psp_R2` plotting on individual runs for specific configurations. Two examples are illustrated in figures 1 and 2 (see pages 7, 8).

The example in Figure 1 uses simulated data with directionality in the forward direction from neurons $1 \rightarrow 2$. The two directionality estimates highlight this clearly, the forward coherence component (red line) dominates, and is either just below or on top of the original coherence estimate (black line). In contrast the reverse component (blue line) and zero-lag component (grey line) are close to zero. The time domain plot (lower right) has a distinct peak only at positive lags indicating an interaction in the forward direction only.

The example in Figure 2 uses simulated data with recurrent excitatory connections between neurones 1 and 3, with an additional delay of 50 ms in the connection $1 \leftarrow 3$ in addition to the delay from synaptic activation. The analysis considers the spike train of neuron 1 as the reference, the interaction will have components in the forward, $1 \rightarrow 3$, and reverse, $1 \leftarrow 3$, directions. The reciprocal connections with additional delay result in oscillatory components in the spike train firing and the correlation structure, as evidenced by the peaks at 15-20 Hz and multiples of these in the log spectra and coherence estimates. The directional coherence has significant components in the forward (red line) and reverse (blue line) as expected. The time domain estimate (lower right) clearly identifies the latencies involved in the forward (peak +2 ms) and reverse (peak -52 ms) directions. Some of the other features outside the confidence limits (e.g. +58 ms, -5 ms) in the cumulant density (covariance) estimate are not present in the correlation function which only reflects the interaction between the neurons, not the periodic aspects of the spike trains. The script can be edited by altering the list of configurations (line 226) and list of runs (line 229) to plot with `psp2_R2`. Further discussion of this simulated data is in [1].

9.2 Script `R2_ts_demo1`

This script demonstrate application of the directionality analysis to time series data with three types of interaction:

1. Instantaneous mixing (zero-lag) with additional uncorrelated noise.
2. Pure delay with additional uncorrelated noise.
3. Low pass filtered copy with additional uncorrelated noise.

The noise signals are either white noise, $\mathcal{N}(0, 1)$, or coloured noise, generated as AR(1) processes using the external function `AR1_gen`. The script generates three figures, illustrative examples for AR(1) noise are shown in Figures 3, 4, 5. Since the MATLAB source data is generated randomly each time the script is run the precise form of figures will vary, however the general form and key features of the correlation

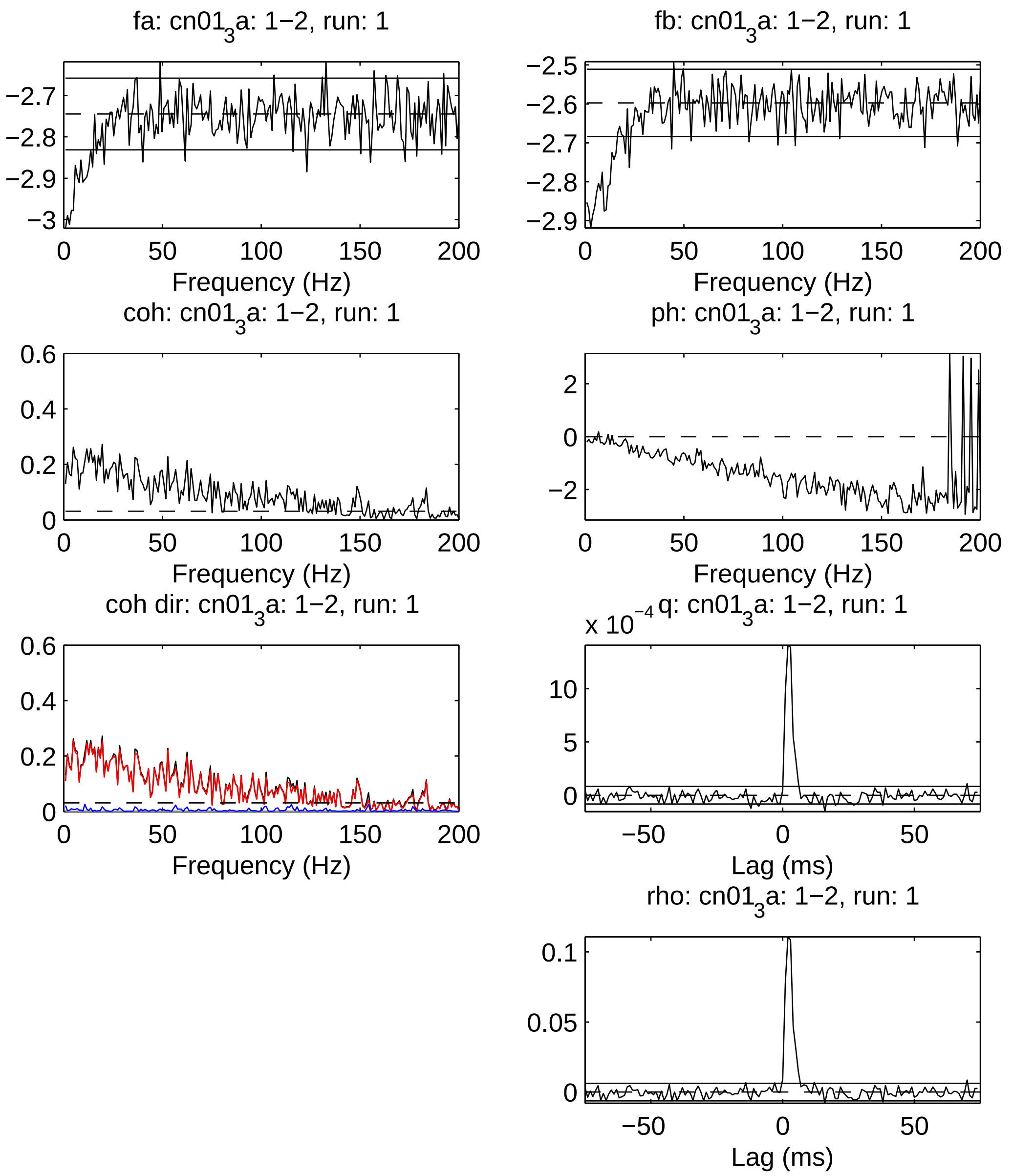

Figure 1: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_cn_demo1` for simulated cortical neuron data, corresponding to configuration a) neurons 1 and 2, in Figure 1 in [1]. Top two graphs are $\log_{10}$ plots of the estimated power spectrum of each spike train. The horizontal lines are the upper and lower 95% confidence limits. In row 2 are the coherence (left) and phase (right) estimates. The horizontal dashed line in the coherence estimate is the upper 95% confidence limit based on the assumption of uncorrelated spike trains. In row 3 are the directional coherence estimates (left; original: black, forward: red, reverse: blue, zero-lag: grey) and cumulant density (right). Below the cumulant density is the estimate of the time domain directionality function, $\rho_{yx}(\tau)$. For the two time domain plots the horizontal dashed line at zero indicates the null value (zero) and the solid horizontal lines indicate the upper and lower 95% confidence limits based on the assumption of uncorrelated spike trains.

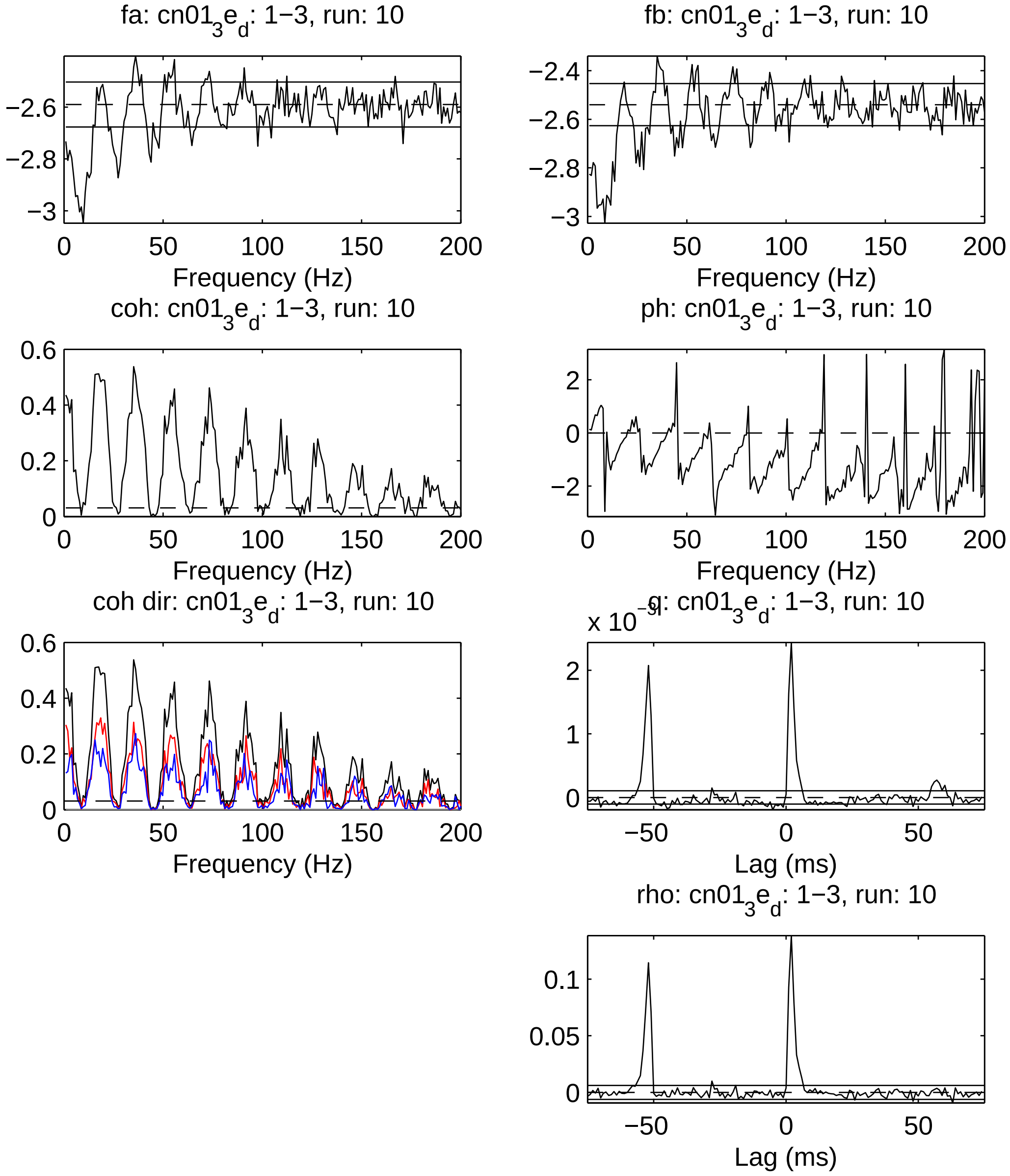

Figure 2: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_cn_demo1` for simulated cortical neuron data, corresponding to configuration h) neurons 1 and 3, in Figure 1 in [1]. Layout as in Figure 1.

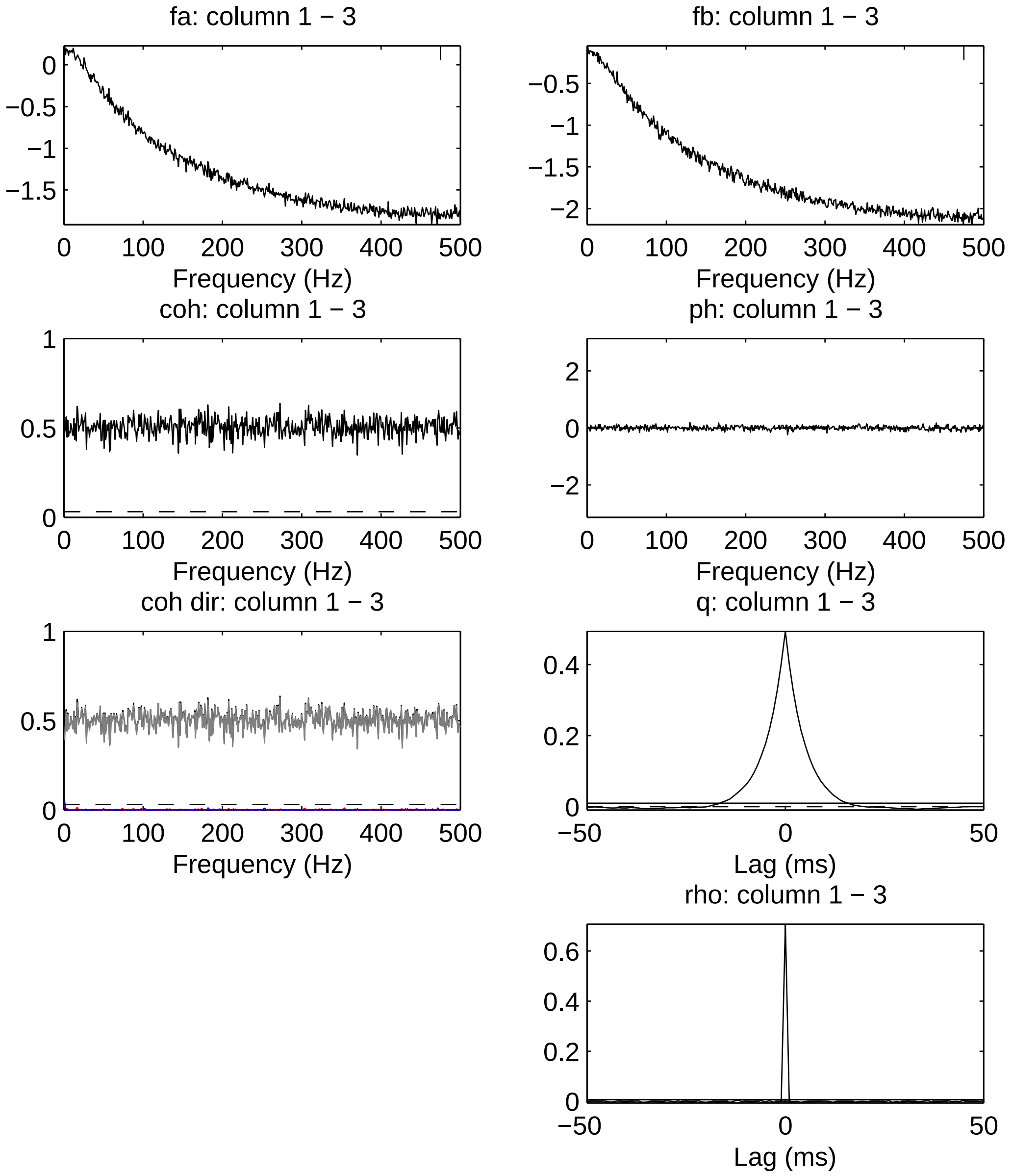

Figure 3: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_ts_demo1`. Data is time series data where correlation structure is an instantaneous mixing with additional noise. Top two graphs are $\log_{10}$ plots of the estimated power spectrum of each time series. The magnitude of the 95% confidence limit is indicated by the vertical bar in the top right of each plot. In row 2 are the coherence (left) and phase (right) estimates. The horizontal dashed line in the coherence estimate is the upper 95% confidence limit based on the assumption of uncorrelated time series. In row 3 are the directional coherence estimates (left; original: black, forward: red, reverse: blue, zero-lag: grey) and cumulant density (right). Below the cumulant density is the estimate of the time domain directionality function, $\rho_{yx}(\tau)$. For the two time domain plots the horizontal dashed line at zero indicates the null value (zero) and the solid horizontal lines indicate the upper and lower 95% confidence limits based on the assumption of uncorrelated time series.

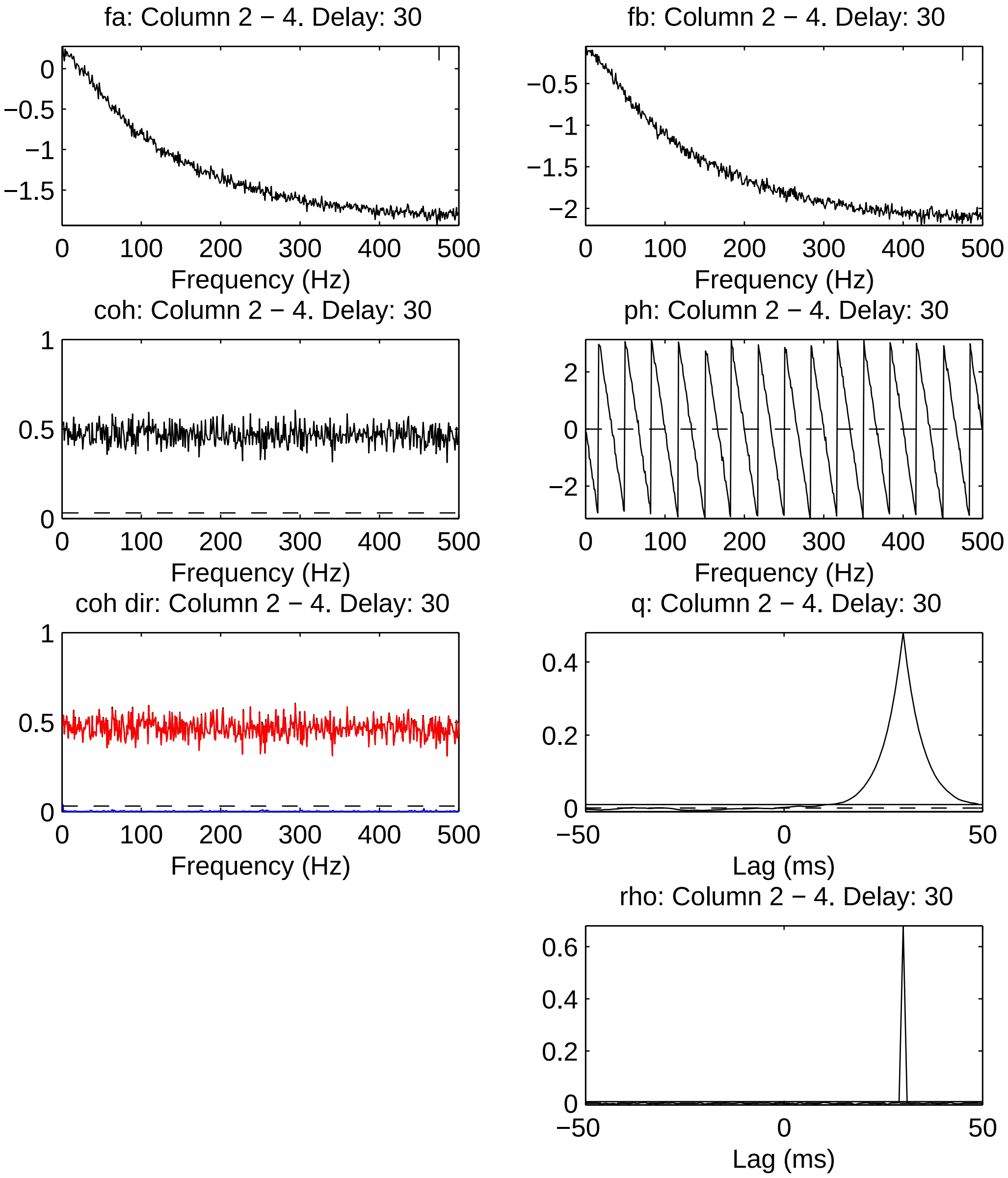

Figure 4: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_ts_demo1`. Data is time series data where correlation structure is a pure delay (30 ms) with additional noise. Layout as in Figure 3.

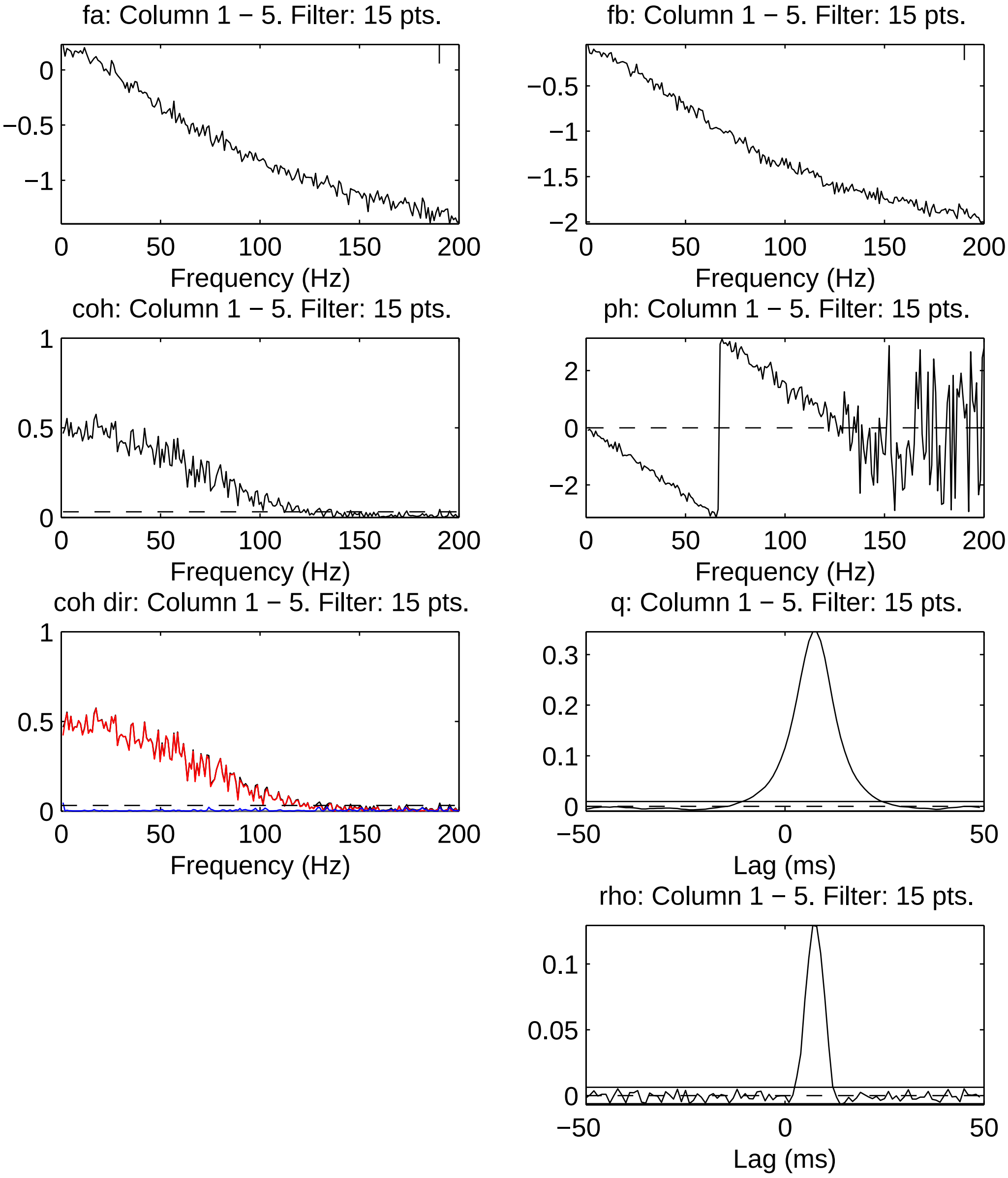

Figure 5: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_ts_demo1`. Data is time series data where correlation structure is a causal low pass filter with additional noise. The filter is implemented as a 15 pt FIR window. Layout as in Figure 3.

and directionality will be similar. The choice of input noise type is controlled by the changing the variable `noise_type` on lines 30 and 31 of the script.

The example in Figure 3 uses simulated data with zero-lag interaction. The zero-lag coherence component (grey line) is virtually identical to the ordinary coherence (black line) which it masks in the directional coherence plot. The forward (red) and reverse (blue) components are very close to zero. The time domain plot (lower right) has a distinct peak at zero (as expected) indicating an instantaneous interaction. This is in contrast to the cumulant density for the coloured signals which is strongly influenced by the autocorrelation structure in the two time series.

The example in Figure 4 is similar, except the correlation structure is a pure delay of 30 ms (against an assumed sampling interval of 1 ms). In this case the forward coherence component (red line) is virtually identical to the ordinary coherence (black line), and the reverse (blue) and zero-lag (grey) components of the coherence are very close to zero. The time domain plot (lower right) has a distinct peak at 30 ms (as expected) indicating a forward interaction. This is in contrast to the cumulant density for the coloured signals which is strongly influenced by the autocorrelation structure in the two time series, and has a broad peak covering many time bins.

The example in Figure 5 has a correlation structure where the signals are related through a causal 15 sample FIR filter. The forward coherence component (red line) is virtually identical to the ordinary coherence (black line), and the reverse (blue) and zero-lag (grey) components of the coherence are very close to zero, as expected. The time domain plot (lower left) peaks at 7-8 ms and the shape reflects that of the filter window, a 4 term Kaiser-Bessel window. Like the previous two figures for this script, the cumulant density is strongly influenced by the autocorrelation structure in the two time series.

9.3 Script `R2_AR4_demo1`

This script demonstrate application of the directionality analysis to time series data generated using a bivariate AR(4) model with coupling in only the forward direction. The data is generated using the external function `AR4_gen1`, see this function for further details of the AR model and coefficients. Autoregressive models are widely used to test parametric approaches to estimate directionality, this example explores how the non parametric approach performs on AR signals. The script generates two figures, the first using the function `sp2a2_R2` to estimate parameters using average periodograms, the second using function `sp2a2_R2_mt` to calculate multi-taper estimates, with $NW = 2$ using 3 orthogonal windows per segment. Figures 6 and 7 illustrate the results from 1 representative run of this script.

The example in Figure 6 uses function `sp2a2_R2` to analyse the bivariate AR signals using average periodograms. The directionality is in the forward direction only for this data. As expected, the forward coherence component (red line) dominates, and is either just below or on top of the original coherence estimate (black line). In contrast the reverse component (blue line) and zero-lag component (grey line) are close to zero. The time domain plots have a strong short term oscillatory structure that is typical of AR(4) processes. The estimate of the directionality function, $\rho_{yx}(\tau)$ (lower right) has all features at positive lags (as expected), apart from a transient immediately to the left of zero lag. This is in contrast to the cumulant density which exhibits significant features over the full plotted lag range (-100 to +300 time bins). The assumed sampling rate in this example is 1 sample/sec, the nyquist frequency is 0.5. The percentage of R^2_{yx} that is accounted for by the forward component, $R^2_{yx;+}$, in this analysis is 91.4% calculated over the same frequency range as plotted in Figure 6.

Figure 7 repeats the analysis using a multi-taper estimate with $NW = 2$ applying 3 orthogonal slepian tapers to each segment. This additional smoothing reduces the 95% confidence limit in the coherence (from 0.0307 in Figure 6 to 0.0103 in Figure 7). The interpretation of the estimates in Figure 7 is similar to those in Figure 6. The percentage of R^2_{yx} that is accounted for by the forward component, $R^2_{yx;+}$, in the multi-taper analysis is 96.3% calculated over the same frequency range as plotted in Figure 7.

10 Relationship to NeuroSpec 2.0

The analysis routines in NeuroSpec 2.1 are compatible with the plotting in NeuroSpec 2.0. The optional fourth output in function `sp2a2_R2_mt` the spectral coefficient matrix, `sc`, can be used in the comparison of spectra, comparison of coherence and pooled coherence routines in NeuroSpec 2.0. Any second level analysis with the multi-taper option must use the same value of NW throughout.

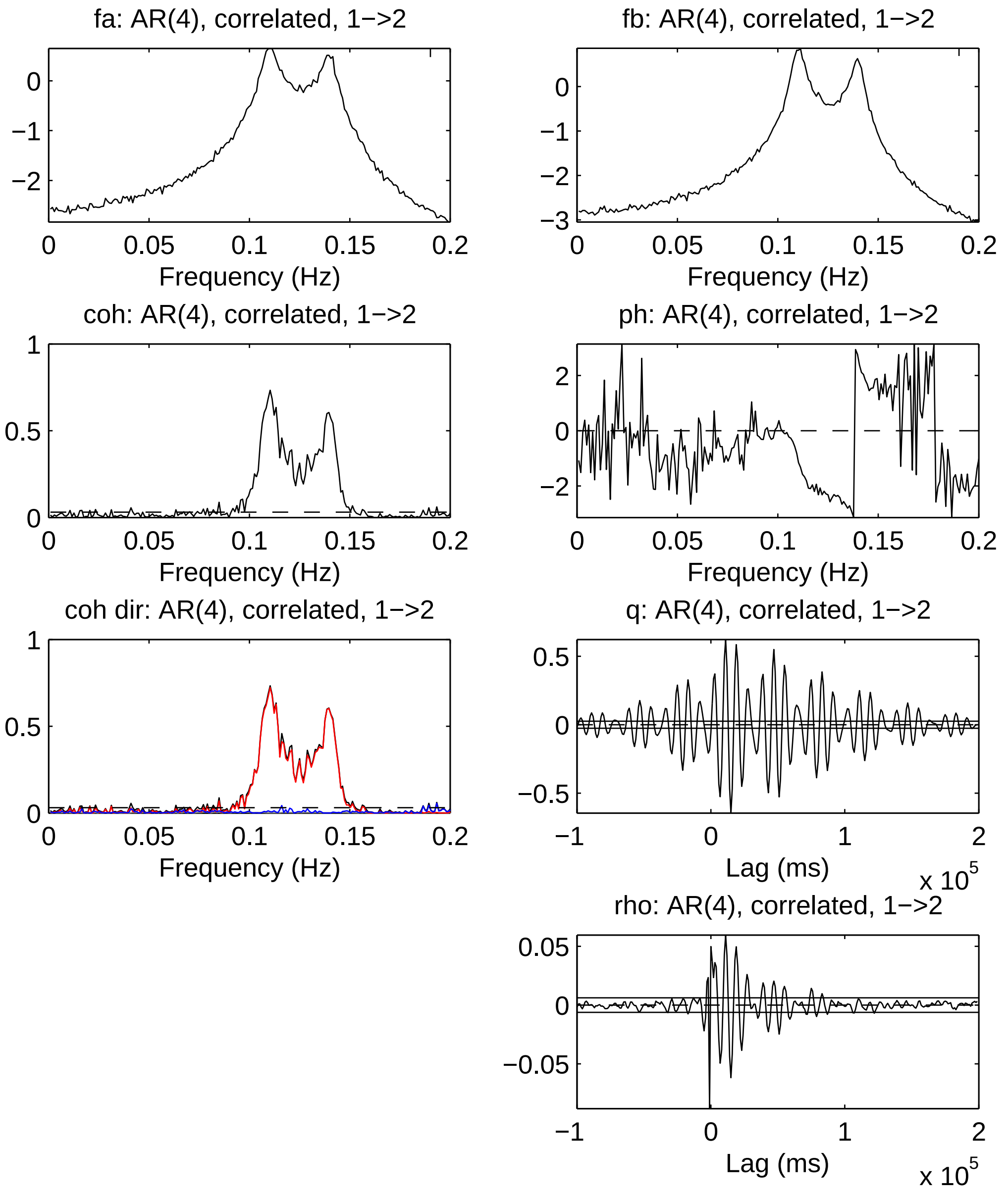

Figure 6: Plot generated by `psp2_R2` of output from `sp2a2_R2` in demonstration script `R2_AR4_demo1`. Data generated using bivariate AR(4) model with coupling in the forward direction only. Top two graphs are $\log_{10}$ plots of the estimated power spectrum of each time series. The magnitude of the 95% confidence limit is indicated by the vertical bar in the top right of each plot. In row 2 are the coherence (left) and phase (right) estimates. The horizontal dashed line in the coherence estimate is the upper 95% confidence limit based on the assumption of uncorrelated time series. In row 3 are the directional coherence estimates (left; original: black, forward: red, reverse: blue, zero-lag: grey) and cumulant density (right). Below the cumulant density is the estimate of the time domain directionality function, $\rho_{yx}(\tau)$. For the two time domain plots the horizontal dashed line at zero indicates the null value (zero) and the solid horizontal lines indicate the upper and lower 95% confidence limits based on the assumption of uncorrelated time series.

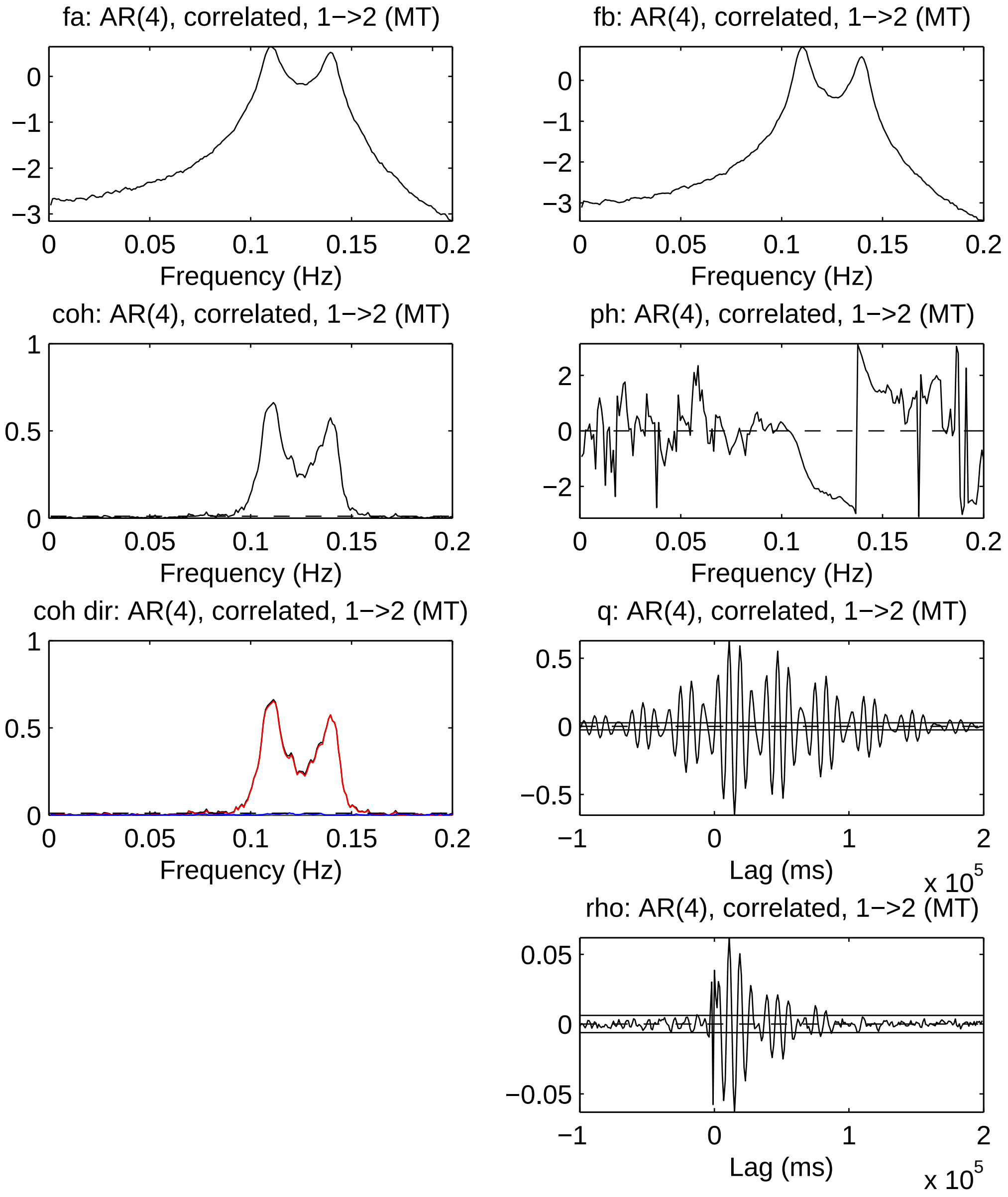

Figure 7: Plot generated by `psp2_R2` of output from `sp2a2_R2_mt` in demonstration script `R2_ts_demo1`. Data and layout are the same as in Figure 6. The analysis here uses multitaper estimates with $NW = 2$, incorporating 3 slepian windows per segment.

NeuroSpec 2.0 routines can handle a wider range of data segmentation strategies, these are not supported in NeuroSpec 2.1. The data segmentation strategy used in NeuroSpec 2.1 is equivalent to the Type 0 strategy in NeuroSpec 2.0. The record R is split into L complete segments of length T samples, where $R = LT$.

The plotting routine `psp2_R2` in NeuroSpec 2.1 uses the same plotting functions as the plotting routine `psp2` in NeuroSpec 2.0. For completeness these functions are included in the release file `neurospec21.zip`. Users who do not wish to use any of the features in NeuroSpec 2.0 do not need to download `neurospec20.zip`. Users with NeuroSpec 2.0 already installed can safely delete these duplicate files without any loss of functionality.

The handling of point-process (spike train) data differs in NeuroSpec 2.1. The difference and MATLAB conversion process is described in section 4.2.1.

11 Licensing

NeuroSpec is free software. It is covered by the GNU General Public Licence, a copy of which is included in the distribution file `neurospec21.zip`.

12 Further information

This user guide focuses on the practical aspects of using the NeuroSpec software. The references below can be consulted for further details of the analytical framework. Non parametric directionality analysis for bivariate data is described in [1]. Two tutorial reviews [2, 3] provide background for bivariate and multivariate analysis using average periodgorams. Details of how to download copies of these articles can be found on the NeuroSpec web site `www.neurospec.org`. Please refer to the articles below (as appropriate) in any scholarly publications resulting from the use of NeuroSpec software.

References

[1] Halliday, D.M. (2015). Nonparametric directionality measures for time series and point process data, *Journal of Integrative Neuroscience*, **14(2)**, 253-277. DOI: 10.1142/S0219635215300127.

[2] Rosenberg, J.R., Amjad, A.M., Breeze, P., Brillinger, D.R., & Halliday, D.M. (1989). The Fourier approach to the identification of functional coupling between neuronal spike trains. *Progress in Biophysics and molecular Biology* **53**, 1-31.

[3] Halliday, D.M., Rosenberg, J.R., Amjad, A.M., Breeze, P., Conway, B.A. & Farmer, S.F. (1995). A framework for the analysis of mixed time series/point process data - theory and application to the study of physiological tremor, single motor unit discharges and electromyograms. *Progress in Biophysics and molecular Biology* **64**, 237-278.

13 Acknowledgements

NeuroSpec version 2.1 has been written by David Halliday. Thanks are due to the many people who have contributed to the development of the NeuroSpec framework over many years: Jay Rosenberg, Bernie Conway, Abdul Majeed Amjad, Alex Rigas, David Murray-Smith, Joe Lau, Peter Breeze, Simon Farmer, Jens Nielsen, Yang Zhan, John-Stuart Brittain, Carl Stevenson, Rob Mason. The work has been supported by grants from the Wellcome Trust, BBSRC, EPSRC & MRC.

This Version 1.01, 14 Jul 2015. Changes from Version 1.0 (8 Jun 2015) to 1.01: Page numbers added to reference [1].

The URL of this document is: http://www.neurospec.org/neurospec21.pdf
The NeuroSpec home page is at: http://www.neurospec.org/
Contact: contact@neurospec.org

A Appendix A: List of files

This Appendix gives an overview of the `.m` and `.mat` files included in NeuroSpec 2.1.

Core routines	
`sp2a2_R2.m`	Non-parametric directionality analysis.
`sp2a2_R2_mt.m`	Non-parametric directionality analysis, with additional options.
`sp2_fn2_R2a.m`	Function called by `sp2a2_R2_mt`, performs spectral analysis.
`sp2_fnR2b.m`	Function called by `sp2_fn2_R2a`, performs directionality analysis.
R^2 metrics over reduced frequency range	
`R2_f1.m`	Calculation of directionality metrics over reduced frequency range.
Main Plotting Routines	
`psp2_R2.m`	Plots results from core routines, including directionality in time and frequency domains.
Other Plotting Routines	
`psp_fa1.m`	Plots input log spectral estimate (from NeuroSpec 2.0)
`psp_fb1.m`	Plots output log spectral estimate (from NeuroSpec 2.0)
`psp_ch1.m`	Plots coherence estimate (from NeuroSpec 2.0)
`psp_ph1.m`	Plots phase estimate (from NeuroSpec 2.0)
`psp_q1.m`	Plots cumulant density estimate (from NeuroSpec 2.0)
`psp_ch1_R2.m`	Plots coherence estimate with directional components
`psp_rho1.m`	Plots estimate of time domain directionality $\rho_{yx}(\tau)$
Demonstration scripts	
`R2_cn_demo1.m`	Demonstrates spike train analysis using simulated cortical neuron data from [1].
`R2_ts_demo1.m`	Demonstrates time series analysis on white and non-white data with known directionality.
`R2_AR4_demo1.m`	Demonstrates time series analysis on coupled AR(4) signals with known directionality.
MAT files	
`JIN_dat1.mat`	Simulated cortical neuron data from [1], used in `R2_cn_demo1.m`.
`NS_dpss_dat.mat`	Data windows for multi-taper analysis, used in `sp2_fn2_R2a.m`.