Granger Causal Connectivity Analysis:
A MATLAB Toolbox

Anil K. Seth

Neurodynamics and Consciousness Laboratory (NCL), and
Sackler Centre for Consciousness Science (SCCS)

School of Informatics,
University of Sussex,
Brighton, BN1 9QJ, UK

a.k.seth@sussex.ac.uk,
www.anilseth.com

v2.9, January 07 2011

University of Sussex
Sackler Centre for Consciousness Science

Contents

1. Introduction

This toolbox provides an array of MATLAB (Mathworks, Natick, MA) functions for evaluating the causal connectivity in the dynamics of a set of variables, for calculating various statistical descriptors of causal connectivity, and for displaying the results in graphical form. It is a substantially upgraded version of a previous MATLAB toolbox first released in 2005 (Seth, 2005). The primary reference for the present toolbox is

Seth, A.K. (2010). A MATLAB toolbox for Granger causal connectivity analysis. *Journal of Neuroscience Methods* 186:262-273

which should be cited whenever this toolbox is used.

The toolbox is based on 'Granger-causality' (G-causality) which implements a statistical interpretation of causality in which *A* 'Granger causes' *B* if knowing the past of *A* can help predict *B,* better than knowing the past of *B* alone (Ding, Chen, & Bressler, 2006; Granger, 1969; Seth, 2007a). The standard implementation of G-causality is via vector autoregressive (MVAR) modeling, in which a set of time series are modeled as weighted sums of past values. This is the implementation underlying the present toolbox.

The main objective of the toolbox is to provide exploratory tools for general purpose G-causality analysis which can be assimilated and deployed quickly, with minimal self-training, by any reasonably competent MATLAB programmer. There is no graphical user interface.

Comments, bug-fixes, and proposed enhancements are always welcome. There is a 'wiki' site for sharing experiences with the toolbox, which you are encouraged to read and contribute to, at http://ccatoolbox.pbwiki.com/. The toolbox is free – use it at your own risk. The toolbox is offered under the GNU general public license (version 3, see www.gnu.org).

1.1 The toolbox

The toolbox consists of seven types of utilities:

- Data validation and preprocessing
- Multivariate autoregression (MVAR) modelling and G-causality analysis
- Statistical tools for assessing significance and model validity etc.
- Causal connectivity statistics, operating on significant interactions
- Graphical display
- Utility tools for e.g., removal of specific frequency components
- Demonstration functions

To cut to the chase, before doing anything else, run `ccaStartup` which adds various MATLAB paths. This function will throw an error if (i) certain files are not found (in which case instructions are given to find them), or (ii) a support ‘mex’ file is not compiled properly (in which case instructions will be given on how to compile the file).

Assuming all is OK, given a matrix `M` of the form variables (rows) by observations (cols), the corresponding G-causalities can be computed by:

```
[ret] = cca_granger_regress(M,nlags);
```

where `nlags` is the ‘model order’ (the number of lagged observations to include in the multivariate regression model). Typically `2<=nlags<=10`, though this depends on the data. This will return a structure `ret` which has several fields. These fields include `ret.prb` which gives the *p*-value for the variable `j` (the column variable) ‘G-causing’ the variable `i` (the row variable). **A low *p*-value (~0) indicates a significant causal interaction.** The field `ret.gc` gives the magnitude of the corresponding interactions, for both significant and non-significant interactions.

To display the results in graphical form, do the following:

```
[PR] = cca_findsignificance(ret,0.01,1);
GC = GC.*PR;
cca_plotcausality(GC);
```

To get a better idea of what’s going on, let’s walk through the function `cca_demo()`.

2. Demonstration

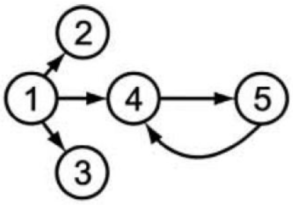
A simple demonstration of the toolbox is provided by

```
cca_demo(N);
```

where `N` determines the data set used. There are two options:

- `N=1`; data from (Baccala & Sameshima, 2001).
- `N=2`; data from (Schelter et al., 2006).

Let's explore the data set from (Baccala & Sameshima, 2001). In this data set, the actual (structural) connectivities are illustrated below:



2.1 Time-domain G-causality

Now let's see what happens, step by step, as the demo runs:

```
>> cca_demo(1)
```

```
detrending and demeaning data
```

- this step subtracts the best fitting line and removes the mean from each time series.

```
checking for covariance stationarity ...
WARNING, data is NOT covariance stationary by ADF
unit roots found in variables: 1
```

- this step checks that all variables are covariance stationary using the ADF test. In this case there is a warning since variable 1 is highly auto-correlated. Note that this is not always a problem (see Section 3).

```
cca_kpss: using default covariance lags = 44
OK, data is covariance stationary by KPSS
```

- this step checks that all variables are covariance stationary using the KPSS test. This test is complementary to the ADF test and can go in a different direction.

```
finding best model order ...
VAR order 2, BIC = 0.31881, AIC = 0.17879
VAR order 3, BIC = 0.3268, AIC = 0.11677
VAR order 4, BIC = 0.40786, AIC = 0.12782
VAR order 5, BIC = 0.49422, AIC = 0.14416
VAR order 6, BIC = 0.5723, AIC = 0.15223
VAR order 7, BIC = 0.65611, AIC = 0.16603
VAR order 8, BIC = 0.73797, AIC = 0.17788
best model order by Bayesian Information Criterion = 2
best model order by Akaike Information Criterion = 3
```

- this step identifies the best ‘model order’ to use, i.e. the number of past observations to incorporate into the regression model. Either the Akaike criterion (AIC) or the Bayesian/Schwartz criterion (BIC) can be used. Alternatively the model order can be set on the basis of *a priori* knowledge of the system. In this case we will use the AIC which specifies a model order of 3. (Note: for bandpass filtered data large model orders may be best, see Section 6).

```
finding conditional Granger causalities ...
```

- this key step runs the time-domain G-causality analysis code.

```
All residuals are white by corrected Durbin-Watson test
Model consistency is OK, value=89.9704
Adjusted r-square is OK: >0.3 of variance is accounted for
by model
```

- here we check that the regression model accounts for a sufficient amount of the variance in the data (i.e., to ensure that the model can be trusted). The Durbin-Watson test assesses whether the residuals are uncorrelated (which they should be). The model consistency assesses the proportion of the correlation structure that is shared by the real data and ‘simulated’ data generated from the MVAR model. The final check shows the adjusted sum-square-error. The latter two conditions have no significant/critical values, but low/high values can be taken as a rule of thumb to be 80 (consistency) and (0.3) adjusted r-square respectively.

```
testing significance at P < 0.01, corrected P-val = 0.0005
```

- here we check which interactions are statistically significant. Various corrections can be selected; here we use the Bonferroni.

```
calculating causal connectivity statistics
time-domain causal density = 0.25
time-domain causal density (weighted) = 0.073279
```

- this step computes the causal density and causal flow values for the causal connectivity network.

The above steps complete the time-domain element of the demonstration. G-causality in the time-domain has the advantage that its statistical properties are known, permitting straightforward significance tests. The frequency-dependent version (below) provides more information but at the expense of requiring surrogate data methods to assess significance (see Section 7).

The graphical output produced by the time-domain analysis looks like this:

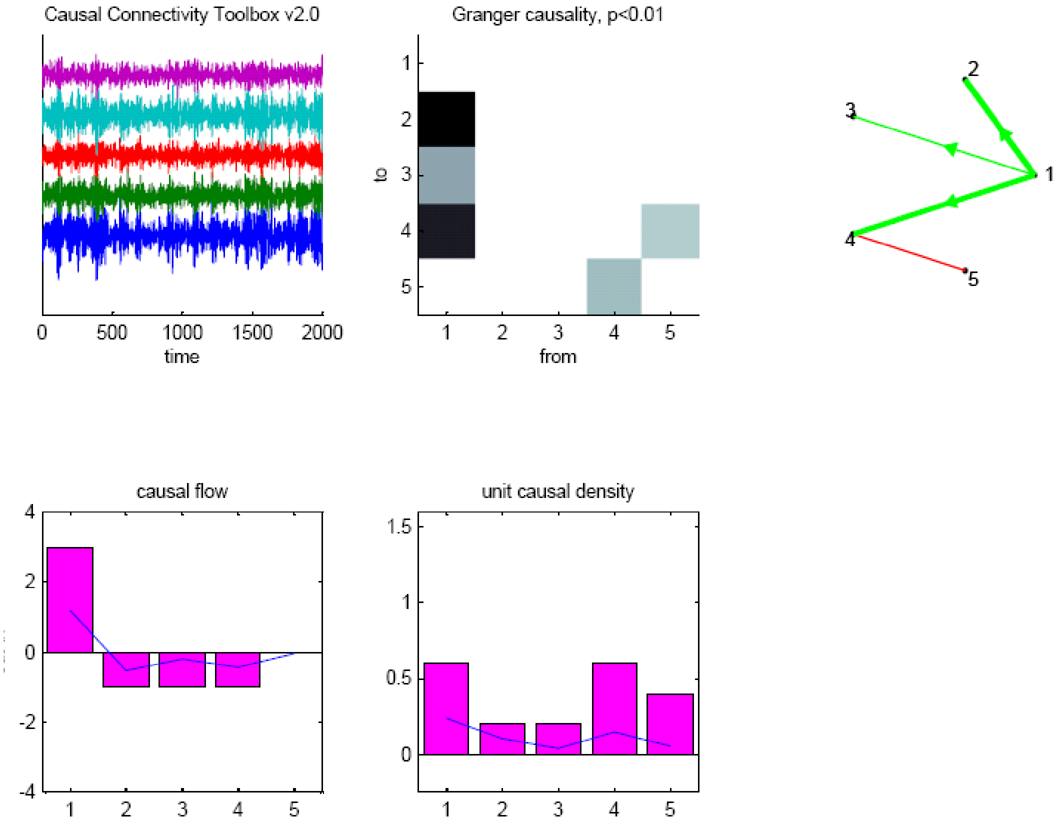


Figure 1. Output of time-domain G-causality analysis for the data set from (Baccala & Sameshima, 2001). This figure shows the raw data (top left), the G-causality networks in matrix form (middle, column causes row) and network form (right), and, along the bottom, causal flow and causal density by variable (node). The bar charts show unweighted causal density/flow, and the lines show density/flow weighted by magnitude of causal interaction. The G-causality network accurately recovers the underlying connectivity. Node 1 is correctly identified as a *causal source* and nodes 2-4 as *causal sinks*. Nodes 1 and 4 are *causal hubs* because of their high degree of causal interactivity with the rest of the network.

2.2 Frequency-domain G-causality

The demonstration function continues as follow (text and figure outputs):

```
finding pairwise frequency-domain Granger causalities ...
calculating causal connectivity statistics
freq-domain causal density (weighted) = 44.4897 thresh=0.2
```

- this step computes the G-causalities in the frequency domain. Here, we have a separate causal network for each frequency (1-100 Hz). Causal densities and causal flows are also calculated by frequency.

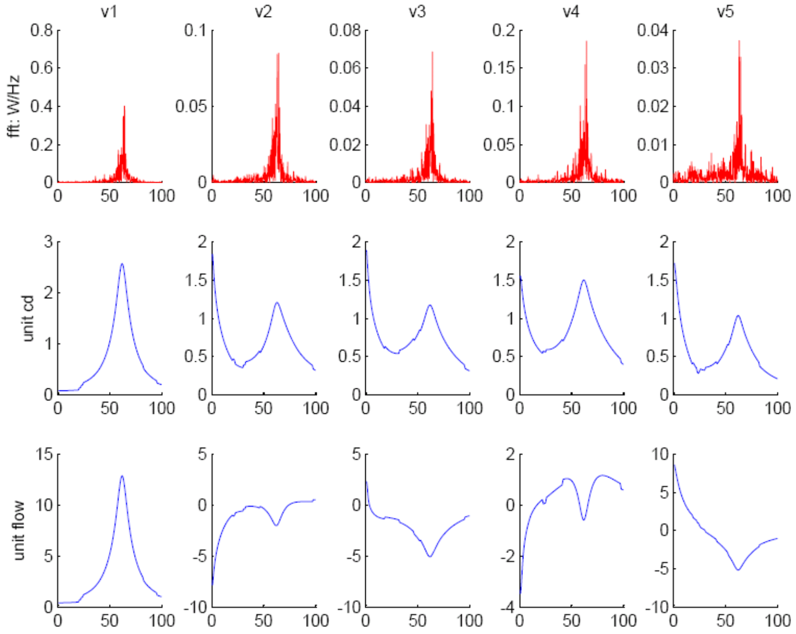


Figure 2. The top row shows the power spectrum for each variable (node), assuming a sampling rate of 500 Hz. The middle and bottom rows show the by-variable causal density and causal flow, following a spectral G-causality analysis. Note that peaks in the causal density spectra coincide with peaks in the power spectra. Note also that y-axes have different scales.

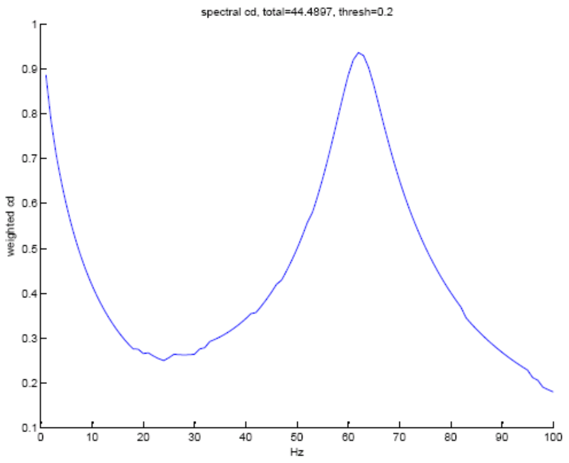


Figure 3. This is the average spectral causal density taken across all variables. Note again the peak at ~60 Hz.

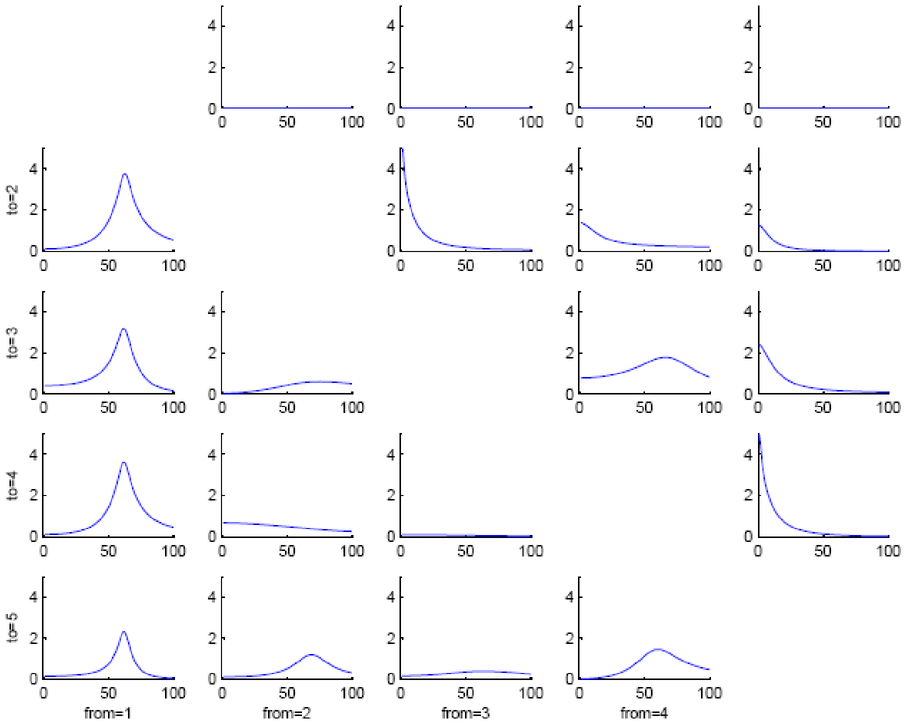


Figure 4. Here we have the pairwise G-causalities broken-down by frequency (column causes row). Again, it is clear that variable 1 'causes' the other variables at ~ 60 Hz.

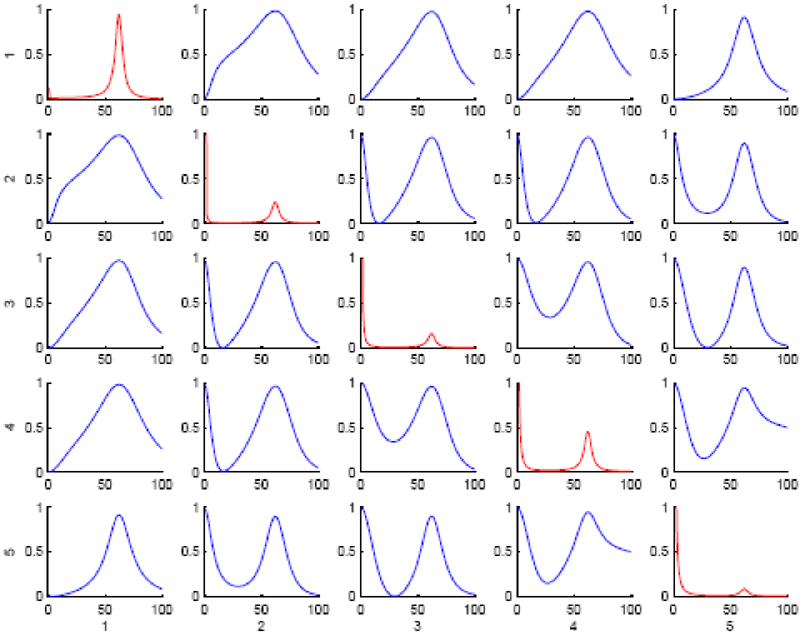


Figure 5. Finally, for comparison, here are the coherence spectra, which are returned along with the frequency-dependent G-causalities. The matrix is symmetric along the diagonal, and the diagonal spectra (in red) give the power spectra again.

3. Data suitability, validation, and preprocessing

Appropriate data consists of sets of time series, each of which represents many observations of a particular variable over time. The variables must be recorded in parallel, so that $t = t_n$ in time series one corresponds to $t = t_n$ in all other series. It is helpful if the number of observations significantly exceeds the number of variables. As the number of variables increases, the number of parameters that need to be estimated per variable increases as *np* (for *n* variables and a model order *p*).

In this section, we focus on establishing covariance stationarity, a necessary requirement for G-causality. The more complicated issues surrounding filtering are left for Section 6. In what follows, assume we have a data matrix `X` of the form variables/nodes (`nvar` rows) by observations (`T` cols).

3.1 Covariance stationarity

Application of G-causality requires that each time series is 'covariance stationary' (CS), i.e., that its mean and variance do not change over time. CS can be assessed in a rule-of-thumb way by examining the auto-correlation function. A non-CS time series will have an autocorrelation function that falls off slowly (Figure 6, left); a CS time series will have a sharply declining autocorrelation function (Figure 6, right). The autocorrelation function of a time series *x* can be plotted using:

- `cca_sacf(x,nlags,1);`

where `nlags` should be quite large (>10).

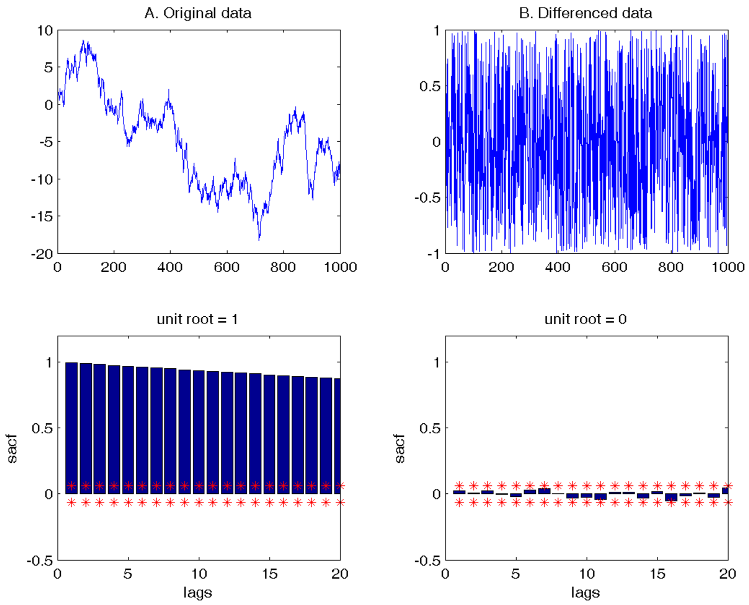


Figure 6. Demonstration of autocorrelation. **A.** A time series with high autocorrelation (a random walk). The series has a unit root. **B.** The same series following 1st order differencing. The autocorrelation is much reduced at all lags, and there is no unit root.

A more formal test for CS is the ADF (Augmented Dickey Fuller) test. The ADF tests whether 'unit roots' are present in the data (Hamilton, 1994). If they are, the time series is not CS. The ADF test can be applied using:

```
-  uroot = cca_check_cov_stat(X,nlags);
```

This function will return a vector (1 *x* `nvar`) in which 0 means 'no unit root' and 1 mean 'unit root present'. `nlags` determines how many previous observations to take into account – a good rule of thumb is to try a few different values. In the example shown in Figure 6, the random walk time series has a unit root, but the differenced time series does not.

A second formal test for stationarity is the KPSS test (Kwiatkowski, Phillips, Schmidt, & Shin, 1992) which is complementary to the ADF test. Whereas the ADF test uses the null hypothesis that there is a unit root, the KPSS test uses the null hypothesis of no unit root. It is possible for the tests to diverge, meaning that there is no clear evidence that either way.

```
-  [H,ks] = cca_kpss(X,nlags,pval);
```

Each output is a vector (1 *x* `nvar`). For `H`, 0 means 'reject null hypothesis' and 1 mean 'do not reject null hypothesis; `ks` give the corresponding KPSS test statistic values. Thus, in contrast to the ADF test, outputs of 0 imply unit root and 1 imply no unit root. `nlags` represents the number of autocovariance lags used by the Newey-West estimator in the KPSS test and, as a rule of thumb (and default) is of the order `sqrt(T)` where `T` is the number of observations in the data. `pval` cannot be arbitrary – choose from 0.01 (default), 0.05, and 0.10.

Both the ADF test and the KPSS test can be applied to multi-trial data (see Section 4.2) using the functions `cca_check_cov_stat_mtrial()` and `cca_kpss_mtrial()` respectively.

3.2 Detrending and demeaning

Two preprocessing steps should be applied as a matter of course. The first is to subtract the best-fitting line from each time series. This can be done using:

```
-  x = cca_detrend(x);
```

The second is to remove the temporal mean from each observation of the time series, to provide a 'zero-mean' situation (Ding, Bressler, Yang, & Liang, 2000). Temporal means can be removed using the function:

```
-  [X,m,e] = cca_rm_temporalmean(X,FLAG);
```

where FLAG determines whether to divide as well by the temporal standard deviation, i.e., to standardize the series (1 = yes, 0 (default) = no); m returns a vector of temporal means, and e (optional) returns a vector of temporal standard deviations.

For multi-realization data (see Section 4.2) one should subtract the *ensemble mean* (Ding, et al., 2000). The ensemble mean is determined by averaging the values for each variable at each time point across trials. Ensemble means can be removed using:

- `[X,M,E] = cca_rm_ensemblemean(X,Nr,Nl,FLAG);`

where Nr is the number of realizations (trials) and Nl is the length of each realization (see Section 4.2), FLAG is as before. Regarding outputs, M is a matrix of ensemble means (nvar by Nl), and E is an (optional) matrix of ensemble standard deviations (nvar by Nl).

Detrending and ensemble demeaning are useful for removing nonstationarity in the mean, i.e. nonstationarity that is reflected in variance, over time, of the time series mean. The standardization step can be used to remove nonstationarity in the standard deviation. It should also be noted that zero-mean time series are required for MVAR model fitting, independently of their stationarity.

3.3 Differencing

If the data is non-CS, a widely used strategy is to 'difference' each time-series (see Figure 6B). While this increases the chances that the time series will be covariance stationary, it may complicate the interpretation, since what is being assessed is now causal connectivity among changes in each time-series. First-order differencing (for a whole matrix, or a single time series) can be carried out using:

- `X = cca_diff(X);`

This procedure can be iterated as required, though the interpretation becomes trickier each time. Multi-trial data can also be differenced using cca_diff_mtrial().

Note that differencing may be problematic for analysis of frequency-dependent G-causality, because it can change the spectral properties of the time series (acting as a kind of high-pass filter). If differencing is needed, be sure to compare the frequency spectra of the original and non-differenced time series. For example, Figure 7 shows Fourier spectra for non-differenced (top) and differenced (bottom) time series for variable 2 of (Baccala & Sameshima, 2001).

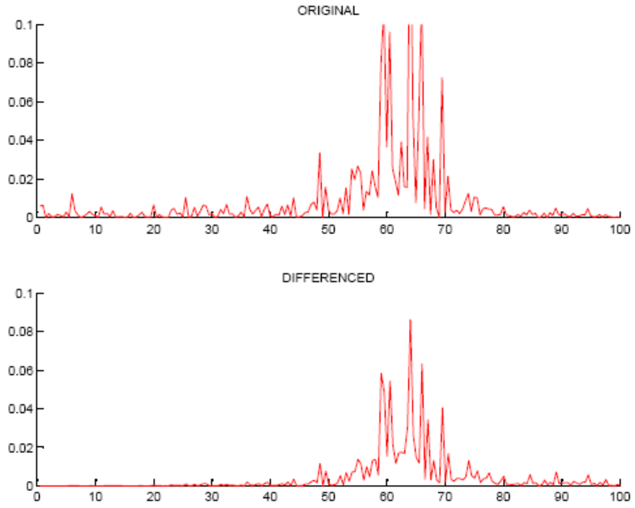


Figure 7. Fourier spectra for original (top) and differenced (bottom) time-series. In this case, the power spectrum is mostly preserved, but it is not identical. In other examples, the post-differencing power spectrum can appear grossly distorted, especially if prior to differencing the bulk of power was in the lower frequency range.

Even if the Fourier spectra are mainly preserved, there can still be subtle differences in the frequency profiles of the causal interactions. For example, compare Figure 8 below with Figure 4. There are subtle differences in the spectra shape around 50 Hz, which are brought about by the interaction between the differencing process and the sampling frequency. Nonetheless, the results are still mostly similar for both differenced and non-differenced data.

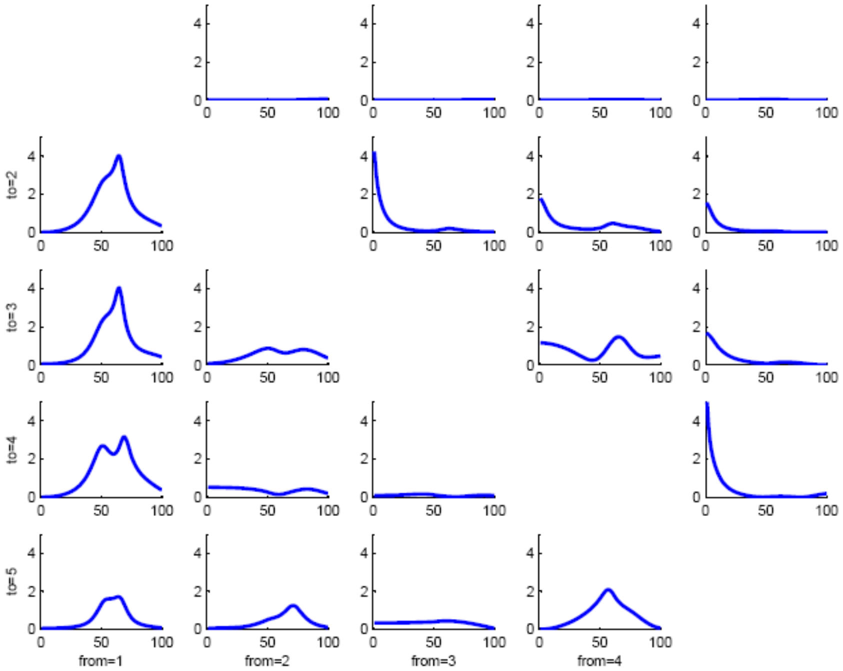


Figure 8. Pairwise G-causalities broken-down by frequency, after first-order differencing. The results are broadly comparable to the non-differenced data (Figure 4) though with some subtle differences in spectral shape.

3.4 Windowing

If a time series is non-CS, a useful approach is to take successive windows (usually overlapping), each of which by itself might be CS (the shorter the series, the more likely it is to be CS). Of course, the ability to do this depends on having sufficient data. On the other hand, an advantage is that the windowing method naturally exposes time-varying causal interactions. For this reason, windowing can be a valuable method even for series that are already CS. Note that there are no dedicated windowing functions in the toolbox.

3.5 Selecting the model order

A key parameter choice for G-causality analysis is the *model order*, i.e., the number of previous observations to take into account when estimating the autoregressive model. Different model orders can sometimes lead to different results, even for the same data. If the chosen model order is too low, the regression model is unlikely sufficiently to capture dynamic relations in the data.

The model order can be chosen based on *a priori* knowledge, for example if you know that the maximum time lag for a causal effect is *p* observations. Alternatively, one can automatically select the model order according to either the Akaike or Bayesian information criteria (Akaike, 1974; Schwartz, 1978), which, at their minimum values, represent optimal trade-offs of accuracy against number of parameters. It is good practice to ensure your results are robust to variations in the model order.

To determine the model order automatically, use:

```
   -  [bic,aic] = cca_find_model_order(X,minp,maxp);
```

where (`minp,maxp`) are the minimum and maximum model orders to consider, respectively, and (`bic,aic`) are the optimal model orders for the Bayesian and Akaike information criteria respectively. If the information criteria are monotonically decreasing (i.e., they never reach a minimum value over the range considered, indicated by `bic/aic = maxp` for many `maxp`) then it is worth eyeballing the data to ensure there are no peculiar artifacts. If there aren't, proceed by trying a few *a priori* plausible model orders.

Note: For multiple realization data (see Section 4.2), use the following function.

```
[bic,aic] = cca_find_model_order_mtrial(X,Nr,Nl,MINP,MAXP).
```

For data that is bandpass filterered, it may be advisable to use rather high model orders so that the regression does not fit the filter convolution window (see Section 6).

4. G-causality analysis

Having completed preprocessing, we are now ready to analyze the data for causal networks. This section will cover the various functions available for such an analysis, including conditional G-causality, spectral G-causality, partial G-causality, Granger autonomy, and analysis of multiple-realization data.

4.1 Conditional G-causality

The most general purpose function, given below, applies a conditional G-causality analysis to the data. The theory behind G-causality is described in (Seth, 2007a) , and will not be repeated here [see also (Granger, 1969), as well as (Hamilton, 1994; Wei, 2006) for general background on time series analysis]. To assess the G-causalities among variables, use:

```
[ret] = cca_granger_regress(X,nlags);
```

The output has several fields:

`ret.gc` - a matrix (`nvar` *x* `nvar`) of magnitudes of G-causality interactions, assessed by the log ratio of prediction errors, with the column variable causing the row variable. Higher numbers mean greater causal influence. These values are not subject to any test of statistical significance.

`ret.prb` - a matrix (`nvar` *x* `nvar`) of significance values; where each entry is the significance of the column variable G-causing the row variable.

`ret.fs` - a matrix of the F-statistics corresponding to the above significance values. Another way of estimating the magnitude of a significant G-causality interaction is by the logarithm of the corresponding F-statistic.

`ret.cov` – covariance matrix for the residuals of the unrestricted regression.

`ret.covr`–covariance matrix for the residuals of each restricted regression.

`ret.rss` – residual sum squares for each variable (see below).

`ret.rss_adj` – adjusted residual sum squares for each variable (see below).

`ret.waut` – significance values for the Durbin-Watson test for autocorrelated residuals (see Section 4.6).

`ret.cons` – model consistency, i.e., the percentage of the (linear) correlation structure of the data accounted for by the model (see Section 4.6).

`ret.rss_adj` – adjusted residual sum squares for each variable (see below).

`ret.doi` – 'difference of influence' terms, which correspond to causality in one direction (`ret.gc`) minus causality in the other. This is useful for fMRI data, or other data subject to smoothing (Roebroeck, Formisano, & Goebel, 2005). Note though that statistical significance needs to be established by surrogate data methods and **not** via filtering through `ret.prb`. (See Section 7). Also, note that this term only makes sense for *pairwise* analyses.

The simplest output of the function is `ret.gc` which gives estimates of causal influence, without worrying about statistical significance. Note that `ret.rss`, `ret.waut`, and `ret.cons` should be examined to ensure that the regression model has sufficiently described the data (see Section 4.6). If it does not, possible strategies include trying a different model order, differencing the data, including extra observations, or decreasing the number of analyzed variables.

There is a possible warning that this function may produce:

Warning: Rank deficient, rank = xxx tol = xxx.

Rank deficiency indicates that the least squares solution to the multivariate regression is not unique. This may happen if (some of) the variables are linearly dependent (i.e., one variable may be a scalar combination of others). The resulting `ret.gc`, `ret.prb`, and `ret.fs` may therefore also be non-unique. Note that rank deficiency does *not* imply that the corresponding causality values are *necessarily* non-unique: it is quite possible that the family of regression solutions all have the same G-causality structure. To avoid rank deficiency, one strategy is to add a small amount of noise to each variable.

Having applied this function to return the structure `ret`, the statistically significant set of G-causality interactions can be recovered by using:

```
[PR,q] = cca_findsignificance(ret,pval,CFLAG);
```

Here, `pval` is the desired significance threshold and `CFLAG` sets the correction required, from the choices below:

CFLAG = 0: No correction, `pval` applied to each comparison independently.
CFLAG = 1: Bonferonni correction, applied threshold is `pval/(N(N-1))` to control for multiple comparisons.
CFLAG = 2: Applied threshold is set to control the 'false discovery rate' (FDR) at `pval`, i.e. the expected proportion of incorrectly rejected null hypotheses (type I errors) (Benjamini & Hochberg, 1995). Typically, the FDR offers greater power than the Bonferroni while still controlling for multiple comparisons.
CFLAG = 3: The applied threshold is set at a rough approximation of the FDR.

The function returns PR, which is mask with 1's for significant interactions and 0's elsewhere; q is the applied threshold following the selected correction.

4.2 Multiple realization G-causality

If your data consists of multiple trials, each of a fixed length, then rather than estimating separate regression models for each trial, or concatenating the trials into a single long time series matrix, you can estimate a single model directly from the multitrial data. The logic is that each trial can be considered as a separate realization of a single underlying stochastic process (Ding, et al., 2006). Use the following function:

```
ret = cca_granger_regress_mtrial(X,Nr,Nl,nlags,STATFLAG)
```

where Nr is the number of realizations (trials) and Nl is the length of each realization. For computational reasons, the function only returns the log ratio magnitude (ret.gc) unless STATFLAG is set, in which case ret.fs, ret.prb, ret.waut and ret.cons are returned as in cca_granger_regress. Significance can be recovered as in Section 4.1. Model order can be selected using

```
[bic,aic] = cca_find_model_order_mtrial(X,Nr,Nl,MINP,MAXP).
```

Note that this approach is useful for stationary multi-realization data. If each trial has a non-stationary component (e.g., an event-related potential in M/EEG) it may be tempting to subtract the ensemble average from each trial in order to regain an ensemble of (stationary) induced responses. However, this approach may in practice not work very well due to variability in the ERP across trials (Wang, Chen, & Ding, 2008)

4.3 Spectral G-causality

The coefficients that comprise the multivariate autoregressive model can be interpreted in the frequency domain, allowing causal interactions to be analyzed by frequency (Ding, et al., 2006; Geweke, 1982). This analysis is particularly useful for neurophysiological data in which frequency bands have distinctive functional interpretations. A disadvantage of the 'spectral' G-causality is that statistical significance has to be estimated empirically, by constructing surrogate data (Efron & Tibshirani, 1994).

To perform spectral G-causality analysis, use:

```
[GW,COH,pp,waut,cons]=cca_pwcausal(X,Nr,Nl,nlags,Fs,freq,
STATFLAG)
```

X, Nr, Nl, and nlags are as above, Fs is the sampling frequency and freq is a vector containing the frequencies to be analyzed (e.g., [1:1:100]). Note that this function is naturally multitrial. For single trial, long time series data, set Nr=1 and Nl = number of observations.

GW is a 3 dimensional matrix of log-ratio G-causalities (nvar by nvar by frequency), COH is an equivalent matrix of coherence values, and pp is the power spectrum (which also appears along the diagonal of COH). As noted, there is no standard significance test for spectral G-causality; use instead bootstrapping and/or random permutation methods (see Section 7). Note that the present spectral measures compute *pairwise interactions only*. If STATFLAG is set to 1 the function will also return values for waut and cons (see Section 4.5), which are otherwise set to a default of -1.

The function cca_pwcausal() function is a wrapper function for the function pwcausal() which is part of the BSMART toolbox (Cui, Xu, Bressler, Ding, & Liang, 2008).

4.4 Partial G-causality

Most functional connectivity methods, including G-causality, are sensitive to the influence of unmeasured variables ('exogenous' or 'latent' influences). Partial G-causality is a technique which offers some robustness to this influence. The logic is that exogenous/latent influences will be reflected in correlations among the residuals, and can partly be factored out by analogy with partial coherence (Guo, Seth, Kendrick, Zhou, & Feng, 2008). As with spectral G-causality, surrogate data methods are needed in order to establish statistical significances.

To perform partial G-causality, use:

```
ret = cca_partialgc(X,nlags,STATFLAG)
```

The output has three fields: ret.gc gives the standard conditional G-causality, ret.fg gives the partial G-causality, and ret.doifg gives the partial 'difference-of-influence' terms. If STATFLAG is set to 1 the function will also return values for waut and cons (see Section 4.5), which are otherwise set to a default of -1.

There is also a multiple-realization version of partial G-causality:

```
ret = cca_partialgc_mtrial(X,Nr,Nl,nlags,STATFLAG)
```

4.5 Granger autonomy

The framework of G-causality can be used to provide a measure of the statistical 'autonomy' of a variable. The logic is that a variable is 'Granger autonomous', or 'G-autonomous', to the extent that (i) its own past helps predict its future, and (ii) these predictions are not accounted for by external variables (Seth, 2010a)

To perform a G-autonomy analysis, use:

```
[ret] = cca_autonomy_regress(X,nlags)
```

The output fields of `ret` are all `1*nvar` in dimension. `prb` gives the probability with which each variable is G-autonomous with respect to the others in the set, `gaut` gives the corresponding magnitudes of the autonomies, and `fs` gives the corresponding F-statistics. The function also returns significance values for the Durbin Watson test (`ret.waut`) and the model consistency (`ret.cons`). Statistically significant G-autonomous variables can be identified using:

```
[PR,q] = cca_findsignificance_autonomy(X,pval,CFLAG);
```

Where inputs and outputs are as in `cca_findsignificance()`. A demonstration of G-autonomy is given by the function `cca_autonomy_demo()` in the `test/` directory.

As an aside, by combining G-autonomy with G-causality it is possible to define an operational measure of 'emergence' as the extent to which a macro-variable is simultaneously (i) autonomous from its micro-level constituents, and (ii) causally dependent on them. For details see (Seth, 2010b).

4.6 Model validity

Having conducted a G-causality analysis it is important to verify that the underlying MVAR adequately captures the data. The toolbox includes three methods for achieving this.

- The Durbin-Watson test (Durbin & Watson, 1950) assesses whether the MVAR residuals (i.e., the errors) are serially uncorrelated, as they should be if the model satisfactorily represents the data. The result of this test is given as a significance value for rejecting the null hypothesis of no serial correlation. That is, values of <0.05 (or <0.01) give cause for concern; values of close to 1 are desirable. The significance value is returned as the field `waut` in various functions. These values should be corrected for multiple comparisons.

- The consistency test (Ding, et al., 2000) assesses the portion of the data captured by the MVAR model, expressed as a percentage. The percentage is calculated across all autocorrelations and cross-correlations in the real data and in a simulated data set generated by the MVAR model. A consistency value of <80% gives cause for concern. This value is returned as the field `cons` in various functions.

- Finally, one can examine the (adjusted) 'sum square error' of the regression. This is returned as the field `rss_adj` in various functions. A value of <0.3 may give cause for concern.

If the model fails any of the above validity tests, then conclusions regarding causal connectivity should be regarded cautiously. Strategies to enhance model validity include increasing the model order, incorporating additional variables into the analysis, and/or using nonlinear approaches (see Section 9). Alternatively, if there is good reason to believe that the system is influenced by exogenous/latent variables, then partial G-causality can be used.

5. Causal network statistics and visualization

This section describes functions for characterizing and visualizing causal networks. In network-theory terminology, variables correspond to network nodes, and causal interactions correspond to directed edges. The functions offered here represent only a small sample of the wide variety of graph-theoretic and network-theoretic tools that could be applied to causal networks.

5.1 Causal density

Causal density measures the total amount of causal interactivity sustained by a network. It is a useful measure of *dynamical complexity* because high causal density reflects simultaneous integration and differentiation in network dynamics (Seth, 2008; Seth, Dienes, Cleeremans, Overgaard, & Pessoa, 2008). High causal density indicates that elements within a system are both globally coordinated in their activity (in order to be useful for predicting each other's activity) and at the same time dynamically distinct (so that different elements contribute in different ways to these predictions).

The causal density of a network's dynamics can be calculated using:

- `[ret] = cca_causaldensity(GC,PR);`

The function returns several fields: `ret.cd` gives the fraction of statistically significant causal connections (bounded 0:1); `ret.cdw` gives an unbounded, weighted value for causal density. `ret.ucd` supplies a vector with causal densities per node ('unit' causal density), and `ret.ucdw` supplies a weighted version of the same. The significance matrix PR is optional; if it is omitted then `ret.cd` and `ret.ucd` will be set to -1. Nodes with high unit causal density can be thought of as 'hubs' in a causal network.

For spectral G-causality, one can calculate frequency-specific causal density:

- `[ret] = cca_causaldensity_spectral(GW,thresh);`

The function returns `ret.scdw` which is a vector of causal densities by frequency. `ret.sucdw` is a matrix with causal densities by node and by frequency. If the input `thresh` is a scalar, then these values will be computed only for causal interactions that exceed the scalar threshold. If the input thresh is a binary matrix (`nvar*nvar*nfreq`) showing significances (calculated by bootstrap or permutation resampling, see Section 7), then these thresholds will be used instead, and the function will also return additional outputs `ret.scd` and `ret.sucd`, corresponding to unweighted causal density and unit causal density in the time-domain case.

5.2 Causal flow

The causal flow of a node in a G-causality network is defined as the difference between its out-degree (number of outgoing connections) and its in-degree (number of incoming connections). Causal flow can identify nodes that have distinctive causal effects on network dynamics: A node with a highly positive flow is a causal ‘source’, a node with a highly negative flow is a causal ‘sink’.

```
- [ret] = cca_causalflow(GC,PR);
```

This function returns several fields:

`ret.indeg`:	only incoming causal influences (`1*nvar`)
`ret.outdeg`:	only outgoing causal influences
`ret.flow`:	outdeg minus indeg
`ret.windeg`:	as above but weighted
`ret.woutdeg`:	as above but weighted
`ret.wflow`:	as above but weighted

Note that the `PR` input is optional, as with `cca_causaldensity().`

There is also a spectral version of causal flow, with the above quantities defined on a per-frequency basis:

```
- [ret] = cca_causalflow_spectral(GW,thresh);
```

As with `cca_causaldensity_spectral`, the threshold can either be a scalar value or a binary three-dimensional significance matrix derived from bootstrap or permutation resampling tests. The output fields are as above but prefaced with ‘`s`’ indicating ‘spectral’.

5.3 Plotting causal networks

Causal networks generated via time-domain G-causality analyses can be graphically represented using the function

```
cca_plotcausality(M,nodenames,sfac);
```

where `M` is a matrix of (weighted or unweighted) G-causality interactions, `nodenames` is an optional vector of labels for the variables in `M`, and `sfac` is a scaling factor applied to the line widths, which can be adjusted as needed to optimize appearance. In the resulting network, unidirectional connections are shown in green with arrowheads, bidirectional connections are shown in red, and the width of each line represents causal magnitude (for bidirectional interactions the largest magnitude is used).

In the frequency domain, causal networks are plotted as a matrix of ‘sub-plots’ with each ‘sub-plot’ showing a causality-by-frequency graph:
`cca_plotcausality_spectral(M,freqs,c1,c2)`

The inputs `M` and `freqs` are essential. `M` is as before, and `freqs` is the vector of frequency values to be shown (e.g., `1:100`). The remaining optional inputs refer to confidence intervals for permutation and bootstrap resampling procedures, if these have been carried out. For permutation resampling, `c1` should be set to the output field `ret.st` from `cca_pwcausal_permute()` and no input for `c2` should be given. The significance threshold for each interaction will then be shown in red, and values exceeding this threshold will be marked by a yellow background. For bootstrap resampling, both `c1` and `c2` should be set, respectively, to the outputs `ret.ll` and `ret.ul` from `cca_pwcausal_bstrap()`, which correspond to the lower and upper confidence limits. Significant values are shown as before. Note the function will determine which if any statistical procedure has been used by the number of inputs given.

Time-domain causal networks can be transformed into a format allowing display by the Pajek network software (http://vlado.fmf.uni-lj.si/pub/networks/pajek/), which is highly recommended for carrying out advanced graphical and statistical analyses of networks:

```
cca_pajek(PR,GC,fname,nodenames);
```

The input `PR` is an `nvar*nvar` matrix of causally significant interactions, GC should contain the corresponding weights (magnitudes). The optional input `fname` specifies a filename (the saved file will be `fname.net`, the default is `demo.net`), and `nodenames` is an optional vector of labels for the nodes. Figure 9 shows a demonstration of a network plotted using Pajek.

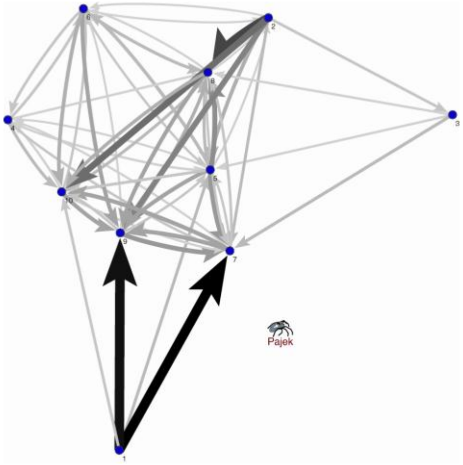


Figure 9. Causal connectivity for a demonstration, plotted using the Pajek program. Network arrangement is determined using the Kamada-Kawai energy minimization algorithm such that highly interconnected nodes tend to bunch together.

5.5 Plotting multitrial data

It is useful to be able to plot ensemble properties of multi-trial data. This facility is provided by the function

```
cca_plotevent(X,Nr,Nl);
```

which outputs separate subplots for each variable, with each subplot showing both the individual trials (thin grey lines) and the ensemble average (thick blue line).

6. Filtering

This section discusses various approaches to filtering, which has to be undertaken with care when used in combination with G-causality analysis. Filtering is a current area of interest regarding G-causality (Florin, Gross, Pfeifer, Fink, & Timmermann, 2010) (see also Barnett and Seth, in preparation).

6.1 Bandpass filtering

For many data sources, bandpass filtering is often used prior to frequency analysis, in order to remove artifacts and/or to focus on frequency bands of *a priori* interest. Although it has long been established (Geweke, 1982) that G-causality is in principle invariant to the application of invertible filters, in practice filtering can still be problematic (Barnett and Seth, in preparation). In brief, filtering (notch or bandpass) can be useful in order to render nonstationary data stationary, but should not be used in order to restrict a time-domain analysis to a frequency range of interest.

For example, the figure below shows a 50-80 Hz bandpass filter applied forward and then backward using the `eegfilt` routine from the EEGLAB toolbox (Delorme & Makeig, 2004), applied to the dataset from (Baccala & Sameshima, 2001). Clearly, the inferred causalities are highly disrupted despite retaining within the filter the frequencies of interest (~60 Hz).

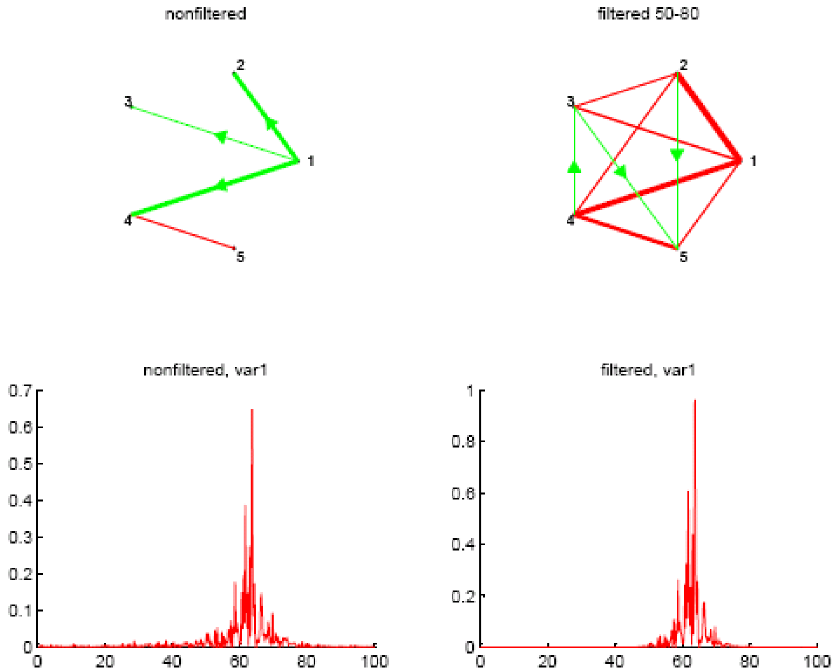


Figure 10. Artifacts induced by bandpass filtering. Left columns show unfiltered data (causal network, Fourier spectrum for variable 1). Right columns show same for filtered

data, between 50 – 80 Hz, using EEGLAB software. Similar artifacts were induced by wideband filtering (1 – 100 Hz).

Artifacts are also observable in the spectral G-causality, again within the frequency range of interest (Figure 11):

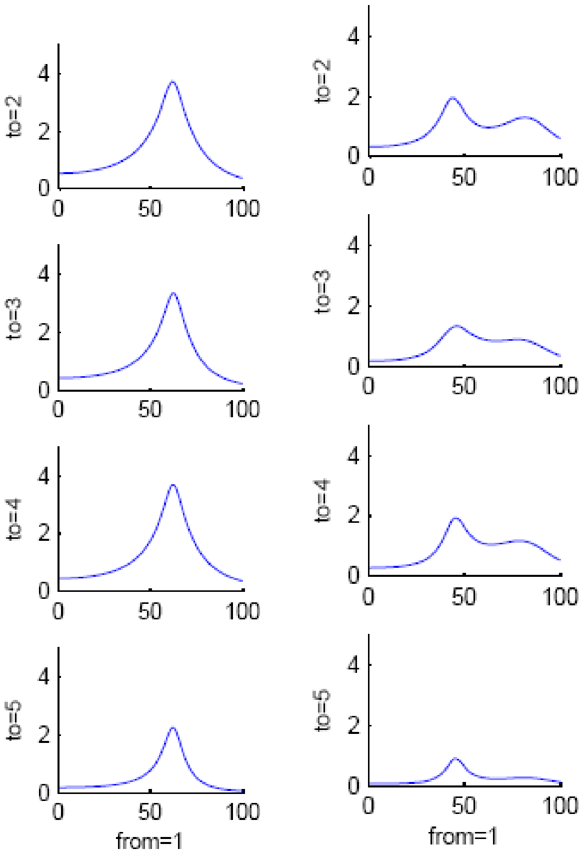


Figure 11. Artifacts induced by bandpass filtering. Left column shows spectral G-causality for unfiltered data (causal outflow from variable 1 only). Right column shows same for bandpass filtered data, between 50 – 80 Hz, using EEGLAB software.

6.2 Multi-taper and notch filtering

Some data sources, for example EEG, exhibit sharp ‘line noise’ peaks due to mains electricity, usually at 60 Hz (USA) or 50 Hz (Europe). Typically, a narrow bandstop ‘notch filter’ is used to remove this peak, but as shown above, such filters can induce artifacts. A suitable alternative is provided by ‘multitaper filtering’ (Mitra & Bokil, 2008) in which a sinusoid oscillation is fit to the noise and then subtracted. Because the amplitude of the line noise may vary over time, this procedure is best applied across windows.

Below is an example of how line-noise at 45 Hz can induce artifacts in a time-domain G-causality analysis, and how these artifacts are removed by multitaper filtering:

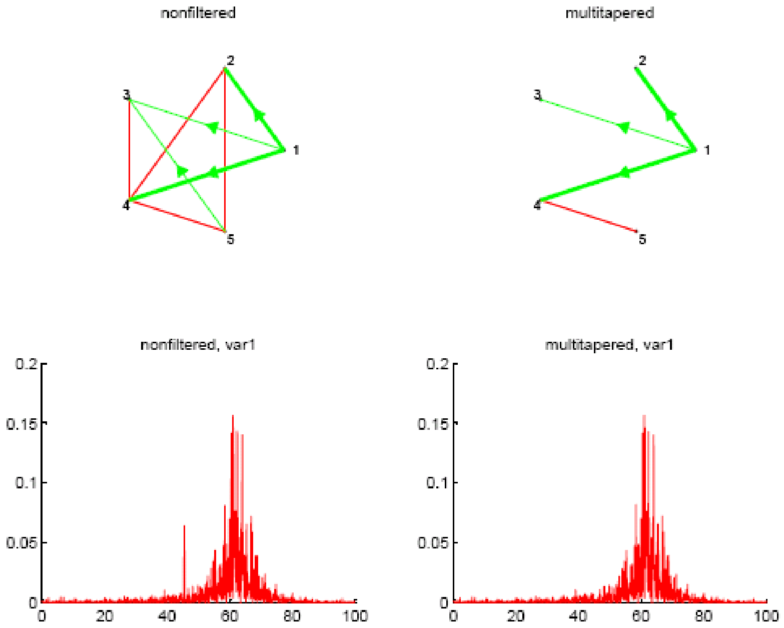


Figure 12. Multitaper filtering for line noise removal. Left column shows erroneous G-causality interactions due to an imposed line-noise source at 45 Hz (see Fourier spectrum at bottom, for variable 1). Right column shows how multitaper filtering removes the line noise and allows the true causal pattern to be recovered.

To apply multitaper filtering, use the following function:

```
- x = cca_multitaper(x,Fs,Fnoise,WSIZE);
```

where `Fs` is the sampling rate, `Fnoise` is the line noise to remove (a single frequency), and `WSIZE` is the size of the window to use. You may have to adjust this parameter in order to get good results, depending on how variable the amplitude of the line noise signal is. Within the function there are other user adjustable parameters as well, including the Fourier padding (`pad`), the taper half-bandwidth (`NW`) and the number of tapers (`K`).

- If you are using differencing, it is best to do so *after* applying the multitaper filter.

- Multitaper filtering is not guaranteed to be successful. Always check the Fourier spectra before and after application, to ensure that the line noise has been removed. Multitaper filtering works best when the sampling frequency is an integer multiple of the line noise.

Finally, note that spectral G-causality can be quite robust to noise (including line noise) that resides at frequencies beyond the frequency range of interest. The figure below shows the pairwise spectral G-causalities for both the raw data with line noise added at 30 Hz (Figure 13A) and the multi-tapered data (Figure 13B), with confidence intervals provided by bootstrap resampling (see Section 7). The results are very similar. By contrast, as shown above, the time-domain G-causality is highly disrupted when line-noise is added and multi-taper filtering is not applied.

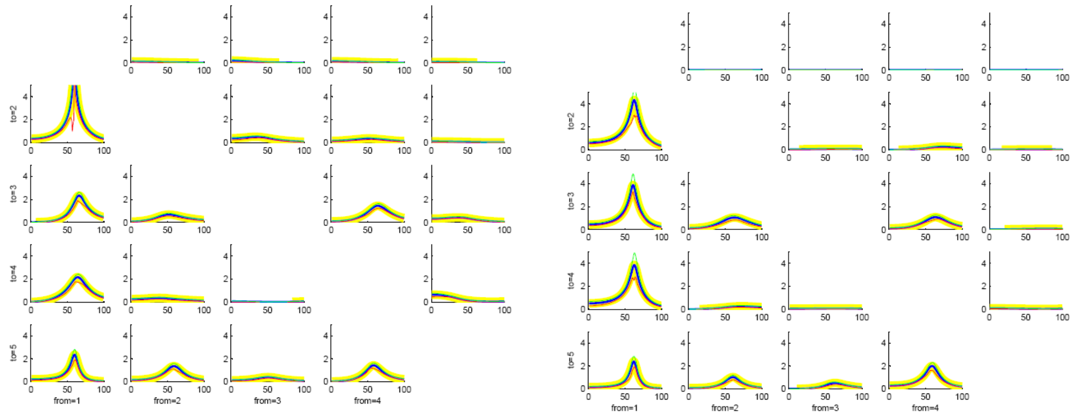


Figure 13. Left (A): Pairwise spectral G-causality with line noise at 30 Hz and no filtering. **Right (B)**. Same, after applying multitaper filtering. Bootstrap confidence intervals are shown in yellow. The two causal patterns are much the same.

6.3 Filtering and model order

Work is ongoing to clarify the interaction between G-causality and filtering operations. In the meantime, numerical experiments suggest that a useful workaround solution is to use large model orders. Figure 14 shows how a model order of 75 leads to a reduction, but not elimination, of filter-induced artifacts.

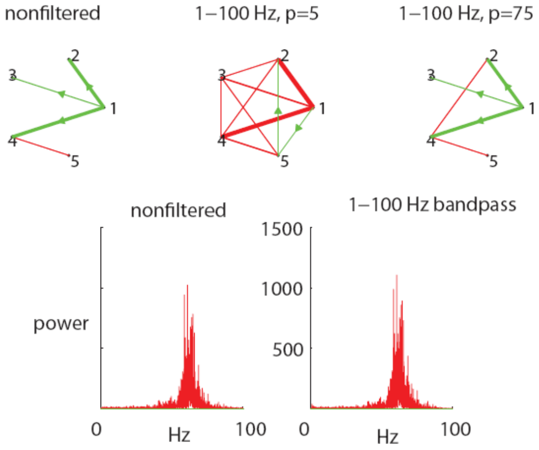


Figure 14. Artifacts induced by bandpass filtering at 1-100 Hz (middle) can be removed by substantially extending the model order (right).

7. Bootstrapping and permutation resampling

Some of the quantities calculated by the functions in this toolbox have distributions that are not analytically known. Testing for statistical significance is therefore best carried out by the computationally intensive but rigorous procedures of bootstrapping and permutation tests (Efron & Tibshirani, 1994). Functions for carrying out these tests are included for spectral G-causality, partial G-causality, and the 'difference-of-influence' terms useful for analysis of fMRI (or other temporally smoothed) data. Permutation tests are useful for assessing whether a value is different from zero. Bootstrapping tests are useful for placing confidence intervals around a value, which can either be used to distinguish that value from zero, or to compare apparently different values. Both permutation and bootstrapping can be applied to either single-trial or multi-trial data.

7.1 Bootstrap resampling

The premise of bootstrap resampling is that a single observation can stand in for a distribution if it is resampled *with replacement* (Efron & Tibshirani, 1994). In the context of vector autoregressive models, this means that a data matrix is subdivided into a number of windows which are repeatedly sampled with replacement to generate surrogate data matrices. Importantly, the causal relations within each window are *not* disturbed. Confidence intervals on the sample value are then generated by examining the empirical quantiles of the bootstrap distribution.

Bootstrap confidence intervals can be estimated for partial G-causality with the following function:

```
[ret] = cca_partialgc_doi_bstrap(X,Nr,Nl,nlags,nBoot,nBwin,
        pval,CORRTYPE,DOIFLAG)
```

This function takes many inputs. The first four are as in `cca_partialgc`. `nBoot` is the number of bootstrap resamples to carry out. The more the better; for rigorous tests, thousands of resamples are recommended. `nBwin` is the window size from which to generate bootstrap resamples; this should be set so that there are at least 10 windows in the data, but it should not be set to less than `nlags`. `pval` and `CORRTYPE` are as in `cca_granger_regress`, and `DOIFLAG` determines whether bootstrap confidence intervals are given for difference-of-influence terms (=1), or for partial difference-of-influence terms (=2) or not for at all for these terms (=0).

This function is compatible with multi-realization data. If there are many realizations, set `nBwin=Nl`. Otherwise, set `nBwin` to a smaller value which is an integer fraction of `Nl`.

The output fields are:

`ret.fg` – partial G-causality
`ret.gc` – log ratio magnitude of conditional G-causality
`ret.pr` – significance of partial G-causality interactions

`ret.ll` – lower confidence limit on partial G-causality
`ret.ul` – upper confidence limit on partial G-causality
`ret.ci` – post-correction confidence intervals (upper/lower)

`ret.doi` – difference of influence terms
`ret.waut` – significance value for Durbin Watson test for autocorrelated residuals
`ret.cons` – model consistency
`ret.lld` – lower confidence limit for doi (if DOIFLAG = 1 or 2)
`ret.uld` – upper confidence limit for doi (if DOIFLAG = 1 or 2)
`ret.prd` – significance of DOIs (if DOIFLAG = 1 or 2)

For spectral G-causality, use the function:

```
[ret] = cca_pwcausal_bstrap(X,Nr,Nl,nlags,nBoot,nBwin,
             Fs,freq,pval,CORRTYPE)
```

where the inputs are as above, apart from `Fs` and `freq` which take the sampling frequency and the vector of frequencies to analyze, respectively.

7.2 Permutation resampling

A permutation test is a test in which a reference distribution is obtained by calculating many values of the test statistic under rearrangements of the labels on the observed data points. In other words, permutation tests involve resampling consistent with the assumption that the null hypothesis is true. In the context of vector autoregressive models, this means that a data matrix is subdivided into a number of windows and then surrogate data matrices are selected by rearranging the windows for each variable separately. (This is in contrast to the bootstrap procedure.) The distribution of G-causalities over these resamples then gives the expected distributions assuming a null hypothesis of no causal connectivity. Significance thresholds for the observed values can then be derived by examining the empirical quantiles of this distribution.

Permutation significance thresholds can be estimated for partial G-causality with the following function:

```
[ret] = cca_partialgc_doi_permute(X,Nr,Nl,nlags,nBoot,
          nBwin,pval,CORRTYPE,DOIFLAG)
```

The input arguments are as for `cca_partialgc_doi_bstrap`. The output fields are:

`ret.fg` – partial G-causality
`ret.gc` – log ratio magnitude of conditional G-causality
`ret.pr` – significance of partial G-causality interactions
`ret.ll` – limit on partial G-causality of permuted series
`ret.md` – median value of permuted series

`ret.ci` - post-correction confidence intervals (upper/lower)
`ret.doi` - difference of influence terms
`ret.waut` – significance value for Durbin Watson test for autocorrelated residuals
`ret.cons` - model consistency
`ret.lld` - limit of permuted series (if DOIFLAG = 1 or 2)
`ret.mdd` – median value of permuted series (if DOIFLAG = 1 or 2)
`ret.prd` - significance of DOIs (if DOIFLAG = 1 or 2)

For spectral G-causality, use the function:

```
[ret] = cca_pwcausal_permute(X,Nr,Nl,nlags,nBoot,nBwin,
             Fs,freq,pval,CORRTYPE).
```

7.3 Demonstration

A function illustrating both permutation and bootstrap resampling is provided in the `/test/` directory. It is:

```
cca_permutebstrap_demo(GENFLAG);
```

where `GENFLAG` takes 1 for generation of new test data (from `cca_testData()`), or 0 for use of previously generated test data (from a previous run of the demo.)

8. Additional resources

As well as the primary reference for this toolbox (Seth, 2010a), causal connectivity statistics have been utilized by the author in (Krichmar, Seth, Nitz, Fleischer, & Edelman, 2005; Seth, 2005, 2007b; Seth & Edelman, 2007; Seth, Izhikevich, Reeke, & Edelman, 2006). A tutorial introduction to G-causality is provided at (Seth, 2007a). Related useful toolboxes include EEGLAB (Delorme & Makeig, 2004) and BSMART (Cui, et al., 2008).

9. Limitations

The present toolbox does not include cover several current areas of development:

- **Nonlinear G-causality**. All methods presented here rely on linear models of continuous time series. Nonlinear extensions have been developed and will be included in future revisions (Ancona, Marinazzo, & Stramaglia, 2004; Chen, Rangarajan, Feng, & Ding, 2004) see also . A simple nonlinear extension is to estimate autoregressive components to polynomials, as in a Taylor expansion. This can give useful results (Seth, 2010a) but may not be stable for large numbers of variables. Having said this, we have recently shown that, for Gaussian variables, G-causality is fully equivalent to (nonparametric) transfer entropy, so that there for such data there is nothing additional to be gained by nonlinear measures (Barnett, Barrett, & Seth, 2009).

- **Multivariate G-causality (MVGC)**. Recently, we have described properties of the MVGC measures which assesses G-causality between multivariate *sets* of variables, rather than between univariate variables. MVGC is based on comparing the determinant (the generalized variance) of the residual covariance matrices, rather than simply the residual variances as can be done in univariate situations. See (Barrett, Barnett, & Seth, 2010).

- **Multivariate spectral G-causality**. There remains debate over the best method to use to compute spectral G-causality in a fully multivariate/conditional manner. Candidate methods include partial directed coherence (Baccala & Sameshima, 2001), the directed transfer function (Kaminski, Ding, Truccolo, & Bressler, 2001), and a recent nonparametric method which remains faithful to Geweke's spectral representation of the G-causality equations (Dhamala, Rangarajan, & Ding, 2008). This method will hopefully be included in future toolbox releases.

- **Point-process G-causality**. The routines in the present toolbox are designed for continuous time series sampled at discrete intervals. To fit multivariate models to point processes (e.g., spike trains), a simple fix is to convolve the point process with a half-Gaussian (spreading into the future). Recent work by Rangarajan, Ding, and colleagues suggests, however, that it is possible to directly derive spectral G-causality from multivariate point process data (Nedungadi, Rangarajan, Jain, & Ding, 2009). This will be a useful method to incorporate into future toolbox versions.

- **fMRI and hemodynamics**. For application to fMRI, users should be aware of the potential confounds caused by variations in hemodynamic response shape and latency among brain regions, as well as by scanner downsampling. These issues are discussed in detail in (Bressler & Seth, 2010; David et al., 2008; Deshpande, Sathian, & Hu, 2010; Friston, 2009; Roebroeck, Formisano, & Goebel, 2009; Ryali, Supekar, Chen, & Menon, 2011; Smith et al., 2011). This area of research is currently very active.

Acknowledgements

I am grateful to Raffaella Giacomini, Eugene Izhikevich, Lionel Barnett, Adam Barrett, Joseph Becerra, Zenas Chao, Mario Forni, Murat Saglam, Stephen Crowley, Spero Nicholas, Michael Holmes, and Felipe Santos for comments, suggested corrections, and function testing. I am especially grateful to Hualou Liang and colleagues for permitting the inclusion of links to the functions `pwcausal.m` and `armorf.m` from the BSMART toolbox (Cui, et al., 2008), available at www.brain-smart.org. Some data validation routines have been adapted from James LeSage's 'econometrics toolbox', freely available at www.spatial-econometrics.com.

Appendix A. Modifications from v1.x

Substantial new features in v2 include:

- Spectral (frequency dependent) G-causality.
- Computation of 'Granger-autonomy' (Seth, 2007b).
- Estimating multivariate models from multiple realizations (multi-trial data) (Ding, et al., 2006).
- Detrending and demeaning functions for preprocessing.
- Model validity checking via Durbin-Watson test and consistency.
- Partial G-causality (Guo, et al., 2008).
- Enhanced significance testing using 'false discovery rate' (Benjamini & Hochberg, 1995) and bootstrap/permutation resampling methods.
- fMRI 'difference of influence' computation (Roebroeck, et al., 2005).
- Multitaper filtering for removal of line-noise measurement artifacts.
- General bug-fixing and improved efficiency of core algorithms.

V2.7 updates (nov 02, 2009)

- Fixed bugs in cca_adf.m and cca_findsignificance.m that caused failures for short time series. Corrected referencing in manual. Added function cca_normrnd.m to utilities.

V2.8 updates (dec 26, 2009)

- Added new functions to apply ADF stationarity test to multi-trial data (cca_check_cov_stat_mtrial.m) and to difference multi-trial data (cca_diff_mtrial.m)
- Added a new graphical function (cca_plotevent.m) for multitrial data showing the ensemble average superimposed on individual trials, lined up by the start of each trial.
- Added new functions to implement the KPSS stationarity test, both for single trial (cca_kpss.m) and multi-trial (cca_kpss_mtrial.m) data.

V2.9 updates (jan 07, 2010)

- Add precompiled mex64 file for 64-bit windows.
- Minor bug fixes throughout.
- Minor revision of manual.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Control, 19*, 716-723.

Ancona, N., Marinazzo, D., & Stramaglia, S. (2004). Radial basis function approaches to nonlinear Granger causality of time series. *Physical Review E, 70*, 056221.

Baccala, L. A., & Sameshima, K. (2001). Partial directed coherence: a new concept in neural structure determination. *Biol Cybern, 84*(6), 463-474.

Barnett, L., Barrett, A., & Seth, A. K. (2009). Granger causality and transfer entropy are equivalent for Gaussian variables. *Physical Review Letters, 103*, 238701.

Barrett, A., Barnett, L., & Seth, A. K. (2010). Multivariate Granger causality and generalized variance. *Phys Rev E Stat Nonlin Soft Matter Phys, 81*, 041907.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B (Methodological), 57*(1), 289-300.

Bressler, S. L., & Seth, A. K. (2010). Wiener-Granger Causality: A well established methodology. *Neuroimage*.

Chen, Y., Rangarajan, G., Feng, J., & Ding, M. (2004). Analyzing multiple nonlinear time series with extended Granger causality. *Physics Letters A, 324*, 26-35.

Cui, J., Xu, L., Bressler, S. L., Ding, M., & Liang, H. (2008). BSMART: a Matlab/C toolbox for analysis of multichannel neural time series. *Neural Netw, 21*(8), 1094-1104.

David, O., Guillemain, I., Saillet, S., Reyt, S., Deransart, C., Segebarth, C., et al. (2008). Identifying neural drivers with functional MRI: an electrophysiological validation. *PLoS Biol, 6*(12), 2683-2697.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J Neurosci Methods, 134*(1), 9-21.

Deshpande, G., Sathian, K., & Hu, X. (2010). Effect of hemodynamic variability on Granger causality analysis of fMRI. *Neuroimage, 52*(3), 884-896.

Dhamala, M., Rangarajan, G., & Ding, M. (2008). Analyzing information flow in brain networks with nonparametric Granger causality. *Neuroimage, 41*(2), 354-362.

Ding, M., Bressler, S. L., Yang, W., & Liang, H. (2000). Short-window spectral analysis of cortical event-related potentials by adaptive multivariate autoregressive modeling: data prepocessing, model validation, and variability assessment. *Biological Cybernetics, 83*(1), 35-45.

Ding, M., Chen, Y., & Bressler, S. L. (2006). Granger causality: Basic theory and application to neuroscience. In S. Schelter, M. Winterhalder & J. Timmer (Eds.), *Handbook of Time Series Analysis* (pp. 438-460). Wienheim: Wiley.

Durbin, J., & Watson, G. S. (1950). Testing for serial correlation in least squares regression. I. *Biometrika, 37*(3-4), 409-428.

Efron, B., & Tibshirani, R. J. (1994). *Introduction to the Bootstrap*: Chapman & Hall.

Florin, E., Gross, J., Pfeifer, J., Fink, G. R., & Timmermann, L. (2010). The effect of filtering on Granger causality based multivariate causality measures. *Neuroimage, 50*(2), 577-588.

Friston, K. (2009). Causal modelling and brain connectivity in functional magnetic resonance imaging. *PLoS Biol, 7*(2), e33.

Geweke, J. (1982). Measurement of linear dependence and feedback between multiple time series. *Journal of the American Statistical Association, 77*(378), 304-313.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica, 37*, 424-438.

Guo, S., Seth, A. K., Kendrick, K. M., Zhou, C., & Feng, J. (2008). Partial Granger causality-Eliminating exogenous inputs and latent variables. *J Neurosci Methods, 172*(1), 79-93.

Hamilton, J. D. (1994). *Time series analysis*. Princeton, NJ: Princeton University Press.

Kaminski, M., Ding, M., Truccolo, W. A., & Bressler, S. L. (2001). Evaluating causal relations in neural systems: granger causality, directed transfer function and statistical assessment of significance. *Biol Cybern, 85*(2), 145-157.

Krichmar, J. L., Seth, A. K., Nitz, D. A., Fleischer, J. G., & Edelman, G. M. (2005). Spatial navigation and causal analysis in a brain-based device having detailed cortical-hippocampal interactions. *Neuroinformatics, 3*(3), 197-222.

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics, 54*, 159-178.

Mitra, P., & Bokil, H. (2008). *Observed Brain Dynamics*. Oxford: Oxford University Press.

Nedungadi, A. G., Rangarajan, G., Jain, N., & Ding, M. (2009). Analyzing multiple spike trains with nonparametric granger causality. *J Comput Neurosci*.

Roebroeck, A., Formisano, E., & Goebel, R. (2005). Mapping directed influence over the brain using Granger causality and fMRI. *Neuroimage, 25*(1), 230-242.

Roebroeck, A., Formisano, E., & Goebel, R. (2009). The identification of interacting networks in the brain using fMRI: Model selection, causality and deconvolution. *Neuroimage*.

Ryali, S., Supekar, K., Chen, T., & Menon, V. (2011). Multivariate dynamical systems models for estimating causal interactions in fMRI. *Neuroimage, 54*(2), 807-823.

Schelter, B., Winterhalder, M., Eichler, M., Peifer, M., Hellwig, B., Guschlbauer, B., et al. (2006). Testing for directed influences among neural signals using partial directed coherence. *J Neurosci Methods, 152*(1-2), 210-219.

Schwartz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 5*(2), 461-464.

Seth, A. K. (2005). Causal connectivity analysis of evolved neural networks during behavior. *Network: Computation in Neural Systems, 16*(1), 35-55.

Seth, A. K. (2007a). Granger causality. *Scholarpedia, 2*(7), 1667.

Seth, A. K. (2007b). Measuring autonomy via multivariate autoregressive modelling. In F. Almeida e Costa (Ed.), *Proceedings of the 9th European Conference on Artificial Life* (pp. 475-484). Lisbon: Springer-Verlag.

Seth, A. K. (2008). Causal networks in simulated neural systems. *Cognitive Neurodynamics, 2*, 49-64.

Seth, A. K. (2010a). A MATLAB toolbox for Granger causal connectivity analysis. *J Neurosci Methods, 183*, 262-273.

Seth, A. K. (2010b). Mearuing autonomy and emergence via Granger causality. *Artificial Life, 16*(2), 179-196.

Seth, A. K., Dienes, Z., Cleeremans, A., Overgaard, M., & Pessoa, L. (2008). Measuring consciousness: relating behavioural and neurophysiological approaches. *Trends Cogn Sci, 12*(8), 314-321.

Seth, A. K., & Edelman, G. M. (2007). Distinguishing causal interactions in neural populations. *Neural Comput, 19*(4), 910-933.

Seth, A. K., Izhikevich, E., Reeke, G. N., & Edelman, G. M. (2006). Theories and measures of consciousness: An extended framework. *Proc Natl Acad Sci U S A, 103*(28), 10799-10804.

Smith, S. M., Miller, K. L., Salimi-Khorshidi, G., Webster, M., Beckmann, C. F., Nichols, T. E., et al. (2011). Network modelling methods for FMRI. *Neuroimage, 54*(2), 875-891.

Wang, X., Chen, Y., & Ding, M. (2008). Estimating Granger causality after stimulus onset: a cautionary note. *Neuroimage, 41*(3), 767-776.

Wei, W. W. S. (2006). *Time Series Analysis: Univariate and Multivariate Methods* (2nd ed.). Boston, MA: Pearson Addison Wesley.