Published in final edited form as:
Neural Netw. 2008 October ; 21(8): 1094–1104. doi:10.1016/j.neunet.2008.05.007.

BSMART: A MATLAB/C toolbox for analysis of multichannel neural time series

Jie Cui[a], **Lei Xu**[a], **Steven L. Bressler**[b], **Mingzhou Ding**[c], and **Hualou Liang**[a*]

a *School of Health Information Science, University of Texas Health Science Center at Houston, 7000 Fannin Street, Suite 600, Houston, TX 77030, USA.*

b *Center for Complex Systems and Brain Sciences, Florida Atlantic University, Boca Raton, FL 33431, USA.*

c *Department of Biomedical Engineering, University of Florida, Gainesville, FL 32611, USA.*

Abstract

We have developed a MATLAB/C toolbox, Brain-SMART (System for Multivariate AutoRegressive Time series, or BSMART), for spectral analysis of continuous neural time series data recorded simultaneously from multiple sensors. Available functions include time series data importing/ exporting, preprocessing (normalization and trend removal), AutoRegressive (AR) modeling (multivariate/bivariate model estimation and validation), spectral quantity estimation (auto power, coherence and Granger causality spectra), network analysis (including coherence and causality networks) and visualization (including data, power, coherence and causality views). The tools for investigating causal network structures are unique functions provided by this toolbox. All functionality has been integrated into a simple and user-friendly graphical user interface (GUI) environment designed for easy accessibility. Although we have tested the toolbox only on Windows and Linux operating systems, BSMART itself is system independent. This toolbox is freely available (http://www.sahs.uth.tmc.edu/hliang/software.htm) under the GNU public license for open source development.

Keywords

Open source toolbox; Neural time series; Multivariate signal analysis; Network analysis; Granger causality

1. Introduction

Recent advances in various technologies for multichannel (multisensor) recording now afford neuroscientists an expanded capacity for the investigation of neural interdependency during cognitive processing (Brovelli et al., 2004; Knight, 2007; Le Van Quyen, Amor, & Rudrauf, 2006; Nicolelis et al., 2003; Super & Roelfsema, 2005). Progress has been hindered, however, because the standard techniques for single channel time series analysis are often insufficient to detect multichannel interactions, and a tool set designed specifically for multichannel data analysis has been lacking.

* Corresponding author: Tel.: +713−500−3914 Fax: +713−500−3929 E-mail: Hualou.Liang@uth.tmc.edu.

In view of this, several open source software toolboxes have been published and distributed for multichannel neural time series analysis, including MEA-Tools (Egert et al., 2002), EEGLAB (Delorme & Makeig, 2004) and ERPWAVELAB (Morup, Hansen, & Arnfred, 2007). All of them include functions for data format conversion to allow users to retrieve data from different hardware systems and provide non-experienced MATLAB (The Mathworks, Inc.) users with graphical user interfaces (GUI) to facilitate access to functional routines. In these toolboxes, however, the measures for quantifying channel interactions are mainly confined to the temporal cross-correlation (Egert et al., 2002) and the coherence spectrum (Delorme & Makeig, 2004; Morup et al., 2007). Questions about causal influence based on the predictability of activity at one recording site from that at another are not addressed by most existing toolboxes. Moreover, the time series data involved in the estimation of cross-correlation or coherence are typically limited to only two channels. It is known that this approach generally may not provide an entirely correct interpretation of inter-channel interactions, as it ignores influences from other channels (Blinowska, Kus, & Kaminski, 2004; Kus, Kaminski, & Blinowska, 2004).

We also have noted that many toolboxes were originally developed to specifically process neuronal spike activity. Measures of continuous neural population activity, such as the Local Field Potential (LFP), ElectroCorticoGram (ECoG), ElectroEncephaloGram (EEG), or MagnetoEncephalogram (MEG), have often not been their primary concern. The Chronux software package (http://chronux.org) is a widely used open-source Matlab toolbox (included in Neuroscience Database Gateway, http://ndg.sfn.org) for analyzing neural data. Its target signal type is either a point process or a continuous time series, as recorded from a single channel, rather than multichannel neural signals. Moreover, the methods used for spectral estimation in this toolbox are based on a non-parametric approach (e.g. multitaper spectral estimation). Generally, non-parametric methods result in lower frequency resolution than do parametric ones, such as the approach based on AutoRegressive (AR) modeling employed in our toolbox. Finally, Chronux does not provide measures of directional information to indicate causal influences between different sources. The Granger causality measure is a unique feature of the software described in this paper. Recently, a similar method (Seth, 2005) has been proposed for characterizing the causal connectivity of a neural system from the graph-theoretic perspective, but is limited to the time domain. Our approach relies on the fundamental concept of Granger causality (Granger, 1969), but is extended to the frequency domain. As such, we anticipate that our software package will find widespread use in the field of neuroscience research and provide insights into the functional dynamics of the brain.

It is the goal of the toolbox, Brain-SMART (System for Multivariate AutoRegressive Time series analysis, or BSMART), to provide tools for analysis of multichannel neural signals recorded in cognitive experiments. The set of available core tools may be classified into three categories, i.e., tools for autoregressive model estimation, spectral quantity analysis, and *network analysis*. A unique feature of BSMART is that instead of using the nonparametric approach to analysis adopted in most previous toolboxes (Delorme & Makeig, 2004; Morup et al., 2007), it carries out analyses based on a parametric method involving the Adaptive Multivariate AutoRegressive (AMAR) modeling technique. Previous studies (Brovelli et al., 2004; Ding, Bressler, Yang, & Liang, 2000) have shown that AMAR modeling can solve two important problems involved in the analysis of neural time series: (1) spectral analysis of fully multivariate (multichannel) time series and (2) spectral analysis of time series in a short time window. The first problem arises from the challenge of understanding the interactions of different channels belonging to the same integrated system. The second comes from the need to examine neurocognitive processing in brief time intervals to characterize its rapid temporal dynamics. The toolbox includes new tools to obtain fully multivariate power and coherence spectra by taking into account the influences from all channels in the data set. It also provides the new tools to obtain the Granger causality spectrum (Granger, 1969), a measure of causal influence from one channel to another. Moreover, BSMART is equipped with tools for the analysis of coherence and Granger causality networks, which have not been available, to our knowledge, in previous toolboxes.

Next, we present the most basic concepts used in BSMART, followed by a detailed description of the implementation of the toolbox to assist the user to utilize these tools most effectively. A summary and a discussion of possible future work will be given at the end of the paper. BSMART can be freely downloaded from http://www.sahs.uth.tmc.edu/hliang/software.htm under the GNU public license[1].

2. Methods

In this section we present a brief introduction to the theoretical background of the methods used in the toolbox to acquaint a reader new to the field with relevant materials. For interested readers, we refer to a set of references (Brovelli et al., 2004; Ding et al., 2000; Maciey Kaminski & Liang, 2005; Liang, Ding, & Bressler, 2000) for more elaborate descriptions.

2.1. Estimation of AMAR model

Let $\mathbf{X}(t) = [X_1(t), X_2(t),...,X_k(t)]^T$ be a k-dimensional random process defined in a segment of windowed time series data, where T stands for matrix transposition. Assuming stationarity of the process $\mathbf{X}(t)$ in each window, one can describe $\mathbf{X}(t)$ by a pth-order autoregressive process:

$$\mathbf{E}(t) = \sum_{m=0}^{p} \mathbf{A}_m \mathbf{X}(t-m) \tag{2.1}$$

where $\mathbf{A}_m$, $m = 1, 2,..., p$ are $k \times k$ coefficient matrices and $\mathbf{E}(t)$ is a k-dimensional, zero mean, uncorrelated noise vector. BSMART employs the Levinson, Wiggins, Robinson (LWR) algorithm (Haykin, 2002) to estimate the $\mathbf{A}_m$ matrices and the covariance matrix of the noise vector ($\mathbf{V}$) from the Yule-Walker equations of the model.

To determine the model order, p, BSMART provides functions to calculate the multivariate Akaike information criterion (AIC) (Box, Jenkins, & Reinsel, 1994) as a function of p,

$$\text{AIC}(p) = -\log|\mathbf{V}| + 2pk^2/N_{total}, \tag{2.2}$$

where $|\mathbf{V}|$ denotes the determinant of the noise covariance matrix and N_{total} is the total number of data points from all the trials. The correct p is usually decided when AIC reaches its first substantial minimum.

Once a model is estimated, an important step is to examine its suitability for the given data set. BSMART provides a set of three tools to test the fitness of the model (Ding et al., 2000). (1) The *whiteness test* can be used to validate the white noise assumption for the model residuals. The null hypothesis is that the residual noise is white and has no temporal correlation. This is tested according to the auto- and cross-correlation coefficients of the residuals. For the null hypothesis to be true, fewer than 5% of the coefficients will fall outside the interval $\left[-2/\sqrt{5}, 2/\sqrt{5}\right]$ by pure chance. If the percentage is larger than 5%, one will reject the null hypothesis. (2) The *stability test* is useful for checking whether a fitted model is stable. BSMART calculates the stability index (SI) as:

$$\text{SI} = \log|\lambda_1| \tag{2.3}$$

[1]Version 2 of June 1991. For further details on GNU public license please refer to http://www.gun.org.

where λ_1 is the largest root of the eigen equation of the model (2.1). Negative SI indicates that the model is stable. (3) The *consistency test* can be performed to examine which portion of the correlation structure in the data is captured by the model. Toward this purpose, a set of simulated data is generated by iterating the equations of the fitted model. Then, auto-correlations and pairwise cross-correlations of both the real and simulated data are calculated. The statistical consistency between the two data sets is measured by the percent consistency (PC):

$$\mathrm{PC}=\left(1-\frac{|\mathbf{R}_s-\mathbf{R}_r|}{|\mathbf{R}_r|}\right)\times 100 \tag{2.4}$$

where $\mathbf{R}_r$ and $\mathbf{R}_s$ are the correlation vectors of the real and simulated data, respectively. Note that we can obtain a bivariate model for two time series $X_i(t)$ and $X_j(t)$, and test its validity according to the procedure above. The bivariate approach will produce $\frac{1}{2}k\,(k-1)$ models, where k is the number of channels.

2.2. Measurement of spectral quantities

Auto Power—Once the model coefficient matrices ($\mathbf{A}_m$) and the covariance matrix of the noise ($\mathbf{V}$) are estimated, all the spectral quantities can be readily derived in the frequency domain. From the transfer function,

$$\mathbf{H}\,(f)=\left(\sum_{m=0}^{p}\mathbf{A}_m e^{-im2\pi f}\right)^{-1}, \tag{2.5}$$

the spectral matrix of the time series data is given by

$$\mathbf{S}\,(f)=\mathbf{X}\,(f)\,\mathbf{X}^*\,(f)=\mathbf{H}\,(f)\,\mathbf{V}\mathbf{H}^*\,(f)\,, \tag{2.6}$$

where ‘*’ denotes matrix transposition and complex conjugation. Subsequently, the auto power of channel i is given by $S_{i,i}(f)$, which is the ith diagonal element of the spectral matrix.

Ordinary Coherence—The ordinary coherence, defined as:

$$C_{i,j}\,(f)=\frac{S_{i,j}\,(f)}{\sqrt{S_{i,i}\,(f)\,S_{j,j}\,(f)}} \tag{2.7}$$

where $S_{i,j}(f)$ is the (i,j)th element of $\mathbf{S}(f)$, can be used to measure the amount of interdependency between two channels i and j. The values of coherence range from zero to one, where zero indicates no interdependence between two channels, while one indicates maximum interdependence.

Note that both power and coherence spectra can be estimated from either a multivariate model or a bivariate model. However, the Granger causality spectrum is found through bivariate autoregressive models only in the current version of BSMART.

Granger Causality—The power and coherence measures do not tell us the direction of influence among different channels. BSMART offers a method to measure the direction of causal influence (M. J. Kaminski & Blinowska, 1991; Liang, Ding, & Bressler, 2000) based on the concept of Granger causality (Granger, 1969). According to Geweke's formulation (Brovelli et al., 2004; Geweke, 1982), the Granger casual influence from channel j to channel i is given by:

$$I_{j\to i}(f) = -\log\left(1 - \frac{\left(V_{j,j} - \frac{V_{i,j}^2}{V_{j,j}}\right)|H_{i,j}(f)|^2}{S_{i,i}(f)}\right), \tag{2.8}$$

where **V**(*f*) , **H**(*f*) and **S**(*f*) are the noise covariance, the transfer function and the spectral matrix of the *bivariate* model of $X_i(t)$ and $X_j(t)$, respectively.

2.3. Network analysis

BSMART also provides a useful tool for analyzing network functional connectivity between channels. After analysis of coherence or Granger causality, a user can manually set a threshold to select those channel pairs with above-threshold measures. These channel pairs are then represented in a network graph, from which one can obtain an insight into the dynamic patterns of information flow and network reorganization during a cognitive process (Brovelli et al., 2004; Ding et al., 2000). The specific steps to find coherence and Granger causality networks will be detailed in the next section.

3. Results

3.1. Overview

The toolbox was written for MATLAB version 6.5 and tested up to version 7.4 (R2007a). Although we have tested it only on Microsoft Windows (XP, Vista) and Linux (Fedora), the toolbox is expected to be compatible with other operating systems within the framework of MATLAB.

3.1.1. Software Structure—Figure 1 schematically shows the organization of the software package including a collection of functions for data format conversion, preprocessing, AMAR data modeling, spectral and network analysis, and *visualization*, as well as functions for displaying *help/demonstration documentation*. A new task of data analysis usually begins with data format conversion. BSMART can read binary data files into MATLAB space as MAT data structures (§3.2.1) accessible by other MATLAB routines. Alternatively, the toolbox can write the MAT data into binary files for usage on other system platforms. The preprocessing procedures include data normalization and trend removal (§3.2.2). It has been shown (Ding et al., 2000) that the preprocessing steps are essential for accurate estimation of spectral quantities and the model stability. The tools for AMAR data modeling (§3.2.3) mainly serve two objectives: one is model estimation, involving tools for determination of the model order using the AIC measure and model estimation using the LWR algorithm; the other is model validation, supported by tools for the whiteness test, stability test and consistency test. After identifying the AMAR model, one can investigate the spectral properties of the time series (§3.2.4) data with functions for power, coherence and Granger causality spectra. The pattern of interactions between channels can also be investigated with the tools of coherence network and Granger causality network structures (§3.2.5). Signal waveforms and the results of spectral analysis can be displayed with visualization tools: data, view, coherence and Granger causality views. The main help documents include a "Users Guide" that explains in detail how to import and process data in BSMART, and a "Function Reference" that illustrates the syntax and usage of each MATLAB function.

3.1.2. GUI Menu Structure—More experienced MATLAB users can automate data analysis by producing batch cripts of function routines in a MATLAB command window (e.g. Table I). For non-experienced users, BSMART also provides a GUI environment, in which a user is able to complete data examination without knowing the details of MATLAB syntax.

The GUI menu structure is split into sections oriented by function type (Figure 2). The first level consists of six menus, namely, “File”, “Tools”, “Model”, “Analysis”, “Plots” and “Help”. The “File” menu is mainly responsible for data format conversion. The “Tools” menu includes shortcuts to call preprocessing routines. An option for calculating the power spectrum by the conventional fast Fourier transform (FFT) method is available, so that its relative merits can be explored in a specific data context. Shortcuts to model estimation and validation procedures have been grouped under the “Model” menu. The “Analysis” menu consists of shortcuts to spectral quantity estimation and network analysis methods. In the “Plots” menu are options for displaying signal waveforms, power spectra and coherence spectra. The final menu section gives access to help documents as well as credits contributions. Figure 3 shows a screen capture of a BSMART user session with GUI interface running under Windows.

Next, we guide the reader through the main functional blocks by describing the steps of processing a sample data set. In addition, usage of the visualization tools will be described within the context of data analysis.

3.2. Functional blocks

To illustrate the utility of BSMART, we employ a small sample data set of event-related LFP time series, sampled at 200 Hz, from 15 bipolar electrodes distributed in the right cerebral cortex of a macaque monkey (details in (Bressler, Coppola, & Nakamura, 1993). Each of the 15 electrode recording sites is considered to be a separate data channel. The monkey was trained to perform a visuomotor pattern discrimination task with GO/NO-GO (motor response/ withhold response) behavior. The data set consists of 137 trials, each lasting 90 ms or 18 time points, from all 15 channels (15×137 = 2055 time series in total). This sample set is also included in the software package and available for downloading.

3.2.1. Data Format Conversion—BSMART allows reading of data in binary format. The function *readdata()* is responsible for such a conversion. The resulting MATLAB data is a three dimensional array in a format of “time points×channels×trials”. The converted sample data set, for example, is an array of 18×15×137. The inverse operation to export MATLAB data into binary data can be done by the function *writedata()*.

3.2.2. Data Preprocessing—Four types of preprocessing are available in BSMART. The function *pre_sube()* subtracts the trial-ensemble mean of each channel from the single-trial time series of that channel. The function *pre_sube_divs()* not only subtracts the trial-ensemble mean, but also normalizes variance by dividing each single-trial time series of each channel by the trial-ensemble standard deviation of that channel. For each channel, the function *pre_subt()* subtracts the temporal mean and the function *pre_subt_divs()* subtracts the temporal mean and also divides each trail by the temporal standard deviation. Note that the specific requirements for data preprocessing are task dependent. For example, the step of variance normalization (in *pre_sube_divs()* and *pre_subt_divs()*) should be avoided in studies where the comparison of signal powers in different conditions is of primary interest. The four available functions may be repeated several times, or in different combinations, until the appropriate preprocessing effects are obtained.

For the sample data set, the preprocessing steps followed the methods proposed in (Ding et al., 2000), which is shown in the sample scripts of Table I. We will assume that the analysis in the following description is based on this preprocessed data set.

3.2.3. AMAR Data modeling—To obtain an appropriate order of the model, BSMART provides the *aci_test()*, which computes the AIC measure function for model order estimation. Given a fixed time window length, this function returns the AIC measure (equation (2.2)) as a function of model order. The maximum order to be tested is provided by the user as an input parameter. The actual calculation of the AIC measure is performed by the function *opssaic()*, which is implemented in C-code. For the analysis of the sample data, we fixed the sliding time window length at 60 ms (12 points) and chose order eight as the maximum order. Figure 4 shows the AIC measure at each model order when the time window was centered at 45 ms (middle of the time series). The curve is almost flat beyond order four. In fact, the shape of the curve is extremely similar for all time windows in the trial. We then chose a value of four as the model order since the AIC measure reached the minimum value of −3.20 at this order.

In the next step the model coefficients ($\mathbf{A}_m$) and the covariance matrix of noise (**V**) can be estimated by one of the four functions in BSMART. The function *mov_mul_model()* estimates the AMAR model. The model order, time window length, starting and ending points of the moving window are provided to the function as input parameters. Actually, the function *mov_mul_model()* accomplishes model estimation by calling the C-coded function *opssmov ()*, which outputs $\mathbf{A}_m$ and **V** as two MATLAB variables ‘A’ and ‘Ve’ at each time window position. The function *mov_bi_model()* is similar to *mov_mul _model()* except that it estimates a bivariate AR model (cf. §2.1). As a special case of the above two functions, the functions *one_mul_model()* and *one_bi_model()* can be used to estimate the multivariate AR model or bivariate AR model, respectively, in a specified window. As an example of applying the function *mov_mul_model()*, we chose a window length of 12 points (sampling at 200 Hz) and the values of 1 and 18 as the beginning and ending points, respectively (to cover the entire trial). The estimated coefficients and noise variances of the AR model of order five (based on the above AIC measure) corresponding to each time window were saved in two files *A_mov.mat* and *Ve_mov.mat*.

For the whiteness test, given the data set, the window length and model order, the function *whiteness_test()* returns the percent of correlation coefficients in relation to the residual noise, after invoking the C-code core function *opsswhite()*. The stability test can be implemented by calling function *lyap_batch()*. This function requires the model coefficients and noise covariance matrix as the inputs and returns the SI values for each time window. The function *consistencytest()* is designed for the consistency test. It returns the measure of PC defined in equation (2.4). As an example, Figure 5 shows the results of the model validation regarding the model specified above. It can be seen that in the whiteness test the percent of correlation coefficients was well below 5%, indicating the whiteness of the residuals. The stability of the model was verified by the negative SIs. High consistency of the model was observed as the PCs were near 80%. Therefore, according to the test results above, we concluded that the model was suited for the sample data set.

It should be noted that, in practice, the procedures of order estimation, time window length selection, model estimation and validation may be repeated several times in order to tune the model for the best fit to the data.

3.2.4. Estimation of Spectral Quantities—The auto power spectrum of each channel can be estimated either from a multivariate AR model by calling the function *autopower()* or from a bivariate model by calling the function *bi_power()*. In both cases, the model coefficients and noise covariance matrices, the number of frequency bins and the sampling frequency are required as the input parameters. However, *bi_power()* identifies the coefficients and matrices by using the directory pathway, as there are many models produced by the bivariate model approach. Figure 6 demonstrates the usage of the “Power View” of the power spectra of channels 9, 10 and 11 in the sample data set by calling the function *po_view()*. The results can be shown in the time-frequency plane (the left column) to expose the time-varying characteristic of the power spectrum. The user can also focus on the spectrum obtained in the time window centered at a time of interest (the right column). We note that all channels exhibit peaks near 20 Hz.

Similar to the methods for finding the power spectrum, the coherence spectrum can be obtained based on either the multivariate model by calling the function *paircoherence()* or the bivariate model by calling the function *bi_coherence()*. Figure 7 shows the “Coherence View” of the results via the function *co_view()*.

The function *mov_bi_ga()* calculates the Granger causalities for a series of time windows, while the function *one_bi_ga()* finds the causal measure at a fixed time window. The user can use BSMART to specify two channels of interest and obtain Granger causality between them. The specification of a channel pair is done by forming a MATLAB array only including the data from these two channels. This procedure may be repeated until the causal influence has been obtained in both directions for all channel pairs. Figure 8 demonstrates the results of the Granger causality spectra by calling *mov_bi_ga()*. The results are shown in “Granger Causality View” via the function *ga_view()*. The arrangement of the visualization is similar to that shown in Figure 6 and Figure 7, except that the causal influence is bidirectional and two sets of plots are required for each channel pair. However, the causality measures from Channel 10 to Channel 9 and from Channel 10 to Channel 11 are not shown, because all values were smaller than an arbitrarily given threshold (= 0.03). In future version of BSMART, an automatic algorithm for identifying significant peaks will be included (M. Kaminski, Ding, Truccolo, & Bressler, 2001).

3.2.5. Coherence and Granger Causality Networks—

Two network analysis tools are available in BSMART, “Coherence Network” and “Granger Causality Network”, to facilitate identification of network patterns. The corresponding BSMART functions are *conetwork()* and *ganetwork()*. Since the usage of these two functions is quite similar, we next describe the tools in a parallel way.

Information about the channels of interest is provided to the functions as input parameters. These channels are usually identified as those involved in coordinated interactions. The task of channel identification can often be done with the above tools for estimating spectral quantities. For example, the power spectra (Figure 6) of channels 9, 10 and 11 showed peaks near 20 Hz within the beta frequency (14–30 Hz) band. Their coherence spectra (Figure 7) also displayed peaks within the beta frequency range. In addition, the Granger causality measures shown in Figure 8 indicated strong causal influence among these three channels within the same frequency band. Thus, a user may choose these three channels as the channels of interest. Both functions of network analysis acquire this information through an input parameter ‘chan’ and two frequency parameters, ‘fre1’ and ‘fre2’, relating to the lower and uper limits of the frequency range. For analysis of the sample data, these parameters were set as *chan* = [9,10,11], *fre1* = 14 and *fre2* = 30.

The functions also require thresholds to identify significant peaks in the coherence and causality spectra. In practice, these thresholds can be determined by a variety of ways, such as baseline statistics (Ding et al., 2000) or permutation testing (Brovelli et al., 2004). In the latter, for instance, a new data set can be constructed by randomly and independently shuffling the trial order of the real time series data from each channel, and can then be supplied to BSMART. Then spectral measures are derived from models fitted to the surrogate data. Carrying out this procedure many times can yield an empirical distribution for the causal measures, and a significant peak may thus be identified according to this distribution. For the sample data, we simply set the thresholds as a fraction of the maximum coherence or causality peak, as described below. Given a frequency range and the time window location, the significant peaks are subsequently defined as those measures that are above threshold at a particular time. For example, we chose a threshold of *thre* = 0.160 for identifying significant coherence peaks. The threshold is close to 1/3 of the maximum coherence found in channels 9, 10 and 11 (Figure 7. C2). Similarly, we chose a threshold of *thre* = 0.075 for significant Granger causality peaks, which is about 1/3 of the maximum Granger causality (Figure 8. D2). Other parameters, such as the coherence, causality measures and time window positions, are provided as input arguments to the functions *conetwork()* and *ganetwork()* as well.

The output of the function is a plot of a "network" graph, consisting of channel symbols and lines/arrows connecting them. In a coherence network, a line connecting two channels means that significant peaks have been identified in a certain frequency range and at a specified time. In a Granger causality network, an arrow indicates significant causal influence from one channel to the other. To display graphs, the positions of the channel symbols should be identified in an input parameter 'location'. The parameter 'location' is a two-dimensional array defining the x- and y-coordinates of each channel symbol on the graph. A detailed description of the construction of 'location' parameters is provided in the help documentation.

We show the coherence and Granger causality networks among Channels 9, 10 and 11 in Figure 9 at two time instants (30 ms and 55 ms). Note the different patterns of interaction at different times, and that the causality networks display directional influence information that is not revealed in the coherence network.

4. Summary and discussion

We have described BSMART, a new open source MATLAB/C toolbox for the spectral analysis of multichannel neural time series. The uniqueness of the package is the accessibility of Granger causality analysis at a sub-second time scale. The tools available in the software have been developed for data format conversion, signal preprocessing, AR data modeling, spectral quantity estimation, network analysis and visualization. All these functions, as well as the functions for calling help documents, have been integrated into a simple and user-friendly GUI environment.

A number of unique challenges exist in the analysis of multichannel neural time series. First, the multivariate nature of the data requires that individual channels be treated as parts of a whole, interacting system. Second, it is often necessary to examine neural activity on a brief time scale on the order of 40−80 ms (or possibly shorter) to capture the rapid dynamics of behavioral and cognitive function. Third, neural data sets are often event-related, and typically consist of multiple trials during which a single task is often repeated hundreds, even thousands, of times. The BSMART software package is specifically developed to meet these challenges. While the package as currently available is designed to analyze multi-trial event-related data sets, it could potentially be used for the analysis of a very long single trial data that is not event-related. In that case, the long time series could be modeled as one realization generated from an underlying stationary stochastic process, or it could be broken into shorter segments, corresponding to trials, for which stationarity would be required both within and across segments, as it is with event-related data.

The technique of AMAR modeling provided in this software is mainly derived from linear systems theory. Although it is generally accepted that nonlinear effects are abundant in the nervous system, linear effects have been found to be the most robust when dealing with, especially, large-scale neurodynamics such as interactions between discrete brain areas (Kelly, Lenz, Franaszczuk, & Truong, 1997; McIntosh et al., 1994; Sporns, Tononi, & Edelman, 2000). When properly applied, our tools can reveal characteristic information about the patterns of interactions among different brain regions, which has been demonstrated in a series of studies concerned with cognitive processes (Brovelli et al., 2004; Ding et al., 2000; Liang, Ding, & Bressler, 2000; Liang, Ding, Nakamura, & Bressler, 2000). Furthermore, from the view of software engineering, the linear approach can generally avoid substantially intensive computation and mathematical complexity incurred by nonlinear approaches, so as to enhance overall performance of the code. In addition, Granger causality can be naturally derived from the AMAR model to evaluate causal influences and directions of driving among multiple neural signals (Liang, Ding, & Bressler, 2000; Liang, Ding, Nakamura et al., 2000).

The core functionality of the software was implemented with C-code. These functions include *opssfull()* for estimating fixed-window multivariate AR models, *opssmov()* for estimating moving window multivariate AR models, *opssaic()* for estimating the AIC measure, and *opsswhite()* for testing the whiteness of the residual signal. The original purpose of adopting this strategy was to take advantage of the efficiency of the C language to increase computational speed. Moreover, since large data sets are managed by C-code, the problem of insufficient memory, a problem that occurs often when processing large data structures in MATLAB, should not be a serious concern in BSMART. Upon the completion of model estimation by C, only a small set of model coefficients and noise covariance matrix are loaded into MATLAB space. For k channels and a 100-bin frequency spectrum (1 Hz resolution for 200 Hz sampling rate), for instance, the size of the spectral matrix for a single time window will be $k \times k \times 100$, which amounts to $k^2 \times 100 \times 8 \times 2$ bytes (the 8 due to 64-bit double-precision for a number in Matlab, and the 2 due to the complex numbers of the spectral matrix). Therefore, analyzing a 128-channel data set requires about only 26 MB, which can usually be well managed on a PC equipped with more than 512 MB RAM. We are aware, however, that with the rapid advancement of computer technology and the release of new versions of MATLAB, the problems of processing speed and memory limitation will not be a major concern in the future. We are intending to rewrite the core functionality with MATLAB language in future release of the software, so as to provide the user with a coherent language environment fully based on the MATLAB platform.

Providing an open-source framework for this software package, we hope that various functions developed in other labs working with multichannel neural signals will be contributed and integrated into this package. Toward this goal, we will build plug-in facilities to allow well-tested functions or user-customized gadgets appear automatically in BSMART menu. We intend for these efforts to help facilitate the rapid distribution of advanced tools of time series analysis in the neuroscience research community.

Acknowledgment

We thank Dr. Yonghong Chen for help with the implementation of the Granger causality function, and Dr. Richard Nakamura of the National Institute of Mental Health for providing the sample data set of event-related LFP time series used in BSMART. This work is partially supported by a grant (NS054314) from the National Institute of Neurological Disorders and Stroke (NINDS) to H.L. and a NSERC postdoctoral fellowship to J.C.

References

Blinowska KJ, Kus R, Kaminski M. Granger causality and information flow in multivariate processes. Physical Review E 2004;70(5):1–4.

Box, GEP.; Jenkins, GM.; Reinsel, GC. Time series analysis: forecasting and control. 3rd ed.. Prentice Hall; Englewood Cliffs, N.J.: 1994.

Bressler SL, Coppola R, Nakamura R. Episodic multiregional cortical coherence at multiple frequencies during visual task-performance. Nature 1993;366(6451):153–156. [PubMed: 8232553]

Brovelli A, Ding MZ, Ledberg A, Chen YH, Nakamura R, Bressler SL. Beta oscillations in a large-scale sensorimotor cortical network: Directional influences revealed by Granger causality. Proceedings of the National Academy of Sciences of the United States of America 2004;101(26):9849–9854. [PubMed: 15210971]

Delorme A, Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods 2004;134(1):9–21. [PubMed: 15102499]

Ding MZ, Bressler SL, Yang WM, Liang HL. Short-window spectral analysis of cortical event-related potentials by adaptive multivariate autoregressive modeling: data preprocessing, model validation, and variability assessment. Biological Cybernetics 2000;83(1):35–45. [PubMed: 10933236]

Egert U, Knott T, Schwarz C, Nawrot M, Brandt A, Rotter S, et al. MEA-Tools: an open source toolbox for the analysis of multi-electrode data with MATLAB. Journal of Neuroscience Methods 2002;117 (1):33–42. [PubMed: 12084562]

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

Granger CWJ. Investigating causual relations by econometric models and cross-spectral methods. Econometrica 1969;37(3):424–438.

Haykin, SS. Adaptive filter theory. 4th ed.. Prentice Hall; Upper Saddle River, N.J.: 2002.

Kaminski M, Ding MZ, Truccolo WA, Bressler SL. Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. Biological Cybernetics 2001;85(2):145–157. [PubMed: 11508777]

Kaminski M, Liang HL. Causal influence: Advances in neurosignal analysis. Critical Reviews in Biomedical Engineering 2005;33(4):347–430. [PubMed: 15982186]

Kaminski MJ, Blinowska KJ. A new method of the description of the information-flow in the brain structures. Biological Cybernetics 1991;65(3):203–210. [PubMed: 1912013]

Kelly EF, Lenz JE, Franaszczuk PJ, Truong YK. A general statistical framework for frequency-domain analysis of EEG topographic structure. Computers and Biomedical Research 1997;30(2):129–164. [PubMed: 9167085]

Knight RT. Neuroscience - Neural networks debunk phrenology. Science 2007;316(5831):1578–1579. [PubMed: 17569852]

Kus R, Kaminski M, Blinowska KJ. Determination of EEG activity propagation: Pair-wise versus multichannel estimate. IEEE Transactions on Biomedical Engineering 2004;51(9):1501–1510. [PubMed: 15376498]

Le Van Quyen M, Amor F, Rudrauf D. Exploring the dynamics of collective synchronizations in large ensembles of brain signals. J Physiol Paris 2006;100(4):194–200. [PubMed: 17317119]

Liang HL, Ding MZ, Bressler SL. On the tracking of dynamic functional relations in monkey cerebral cortex. Neurocomputing 2000;32:891–896.

Liang HL, Ding MZ, Nakamura R, Bressler SL. Causal influences in primate cerebral cortex during visual pattern discrimination. Neuroreport 2000;11(13):2875–2880. [PubMed: 11006957]

McIntosh AR, Grady CL, Ungerleider LG, Haxby JV, Rapoport SI, Horwitz B. Network analysis of cortical visual pathways mapped with PET. The Journal of Neuroscience 1994;14(2):655–666. [PubMed: 8301356]

Morup M, Hansen LK, Arnfred SM. ERPWAVELAB - A toolbox for multi-channel analysis of time-frequency transformed event related potentials. Journal of Neuroscience Methods 2007;161(2):361–368. [PubMed: 17204335]

Nicolelis MAL, Dimitrov D, Carmena JM, Crist R, Lehew G, Kralik JD, et al. Chronic, multisite, multielectrode recordings in macaque monkeys. Proceedings of the National Academy of Sciences of the United States of America 2003;100(19):11041–11046. [PubMed: 12960378]

Seth AK. Causal connectivity of evolved neural networks during behavior. Network: Computation in Neural Systems 2005;16(1):35–54.

Sporns O, Tononi G, Edelman GM. Connectivity and complexity: the relationship between neuroanatomy and brain dynamics. Neural Networks 2000;13(8−9):909–922. [PubMed: 11156201]

Super H, Roelfsema PR. Chronic multiunit recordings in behaving animals: advantages and limitations. Progress in Brain Research 2005;147:263–282. [PubMed: 15581712]

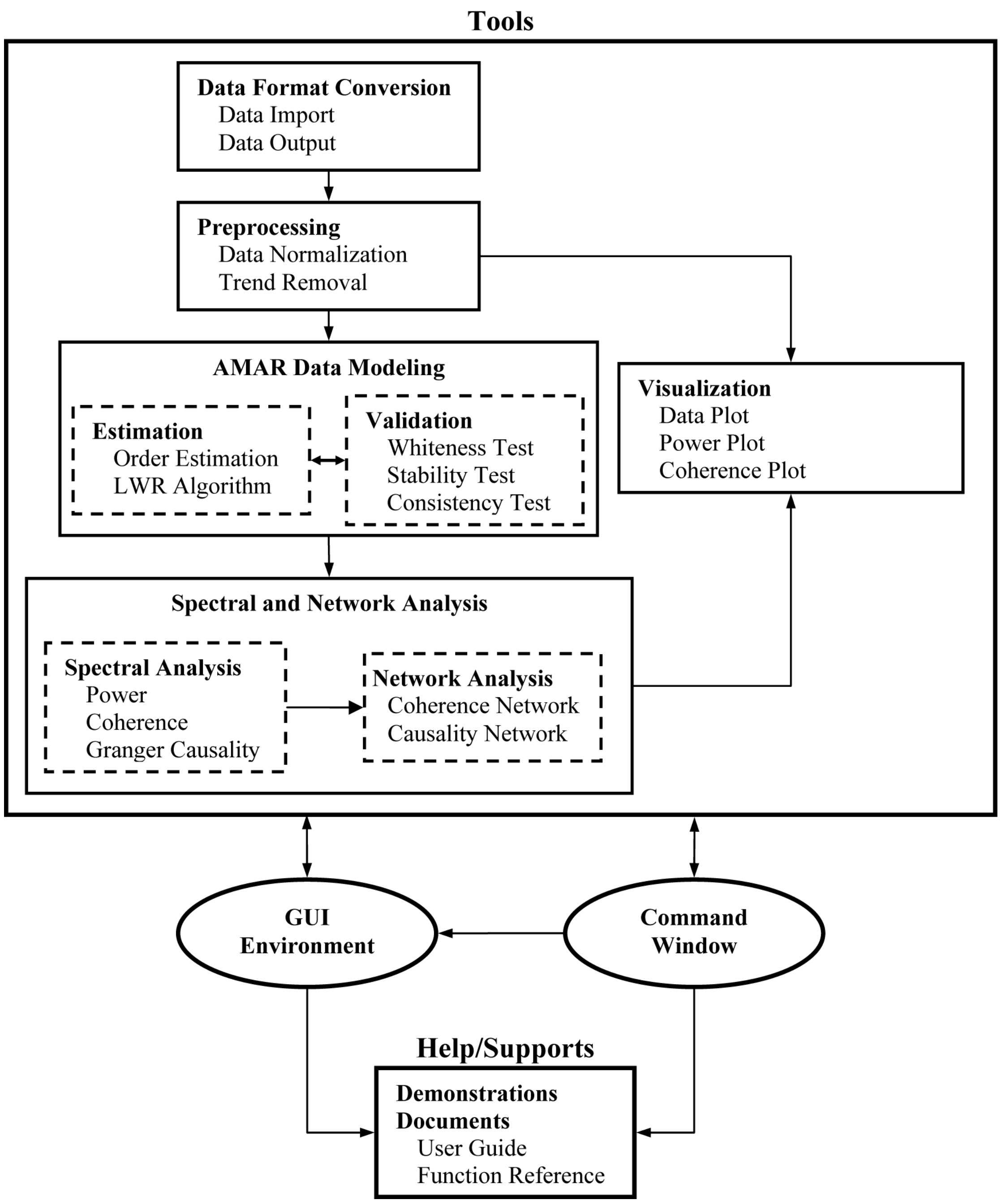

Figure 1.
Functional block diagram of BSMART showing the main computational blocks and the algorithmic options. The core tools are shown in the dash-edged boxes. Different routines of the tools and help demonstrations can be invoked either from GUI environment or MATLAB command window.

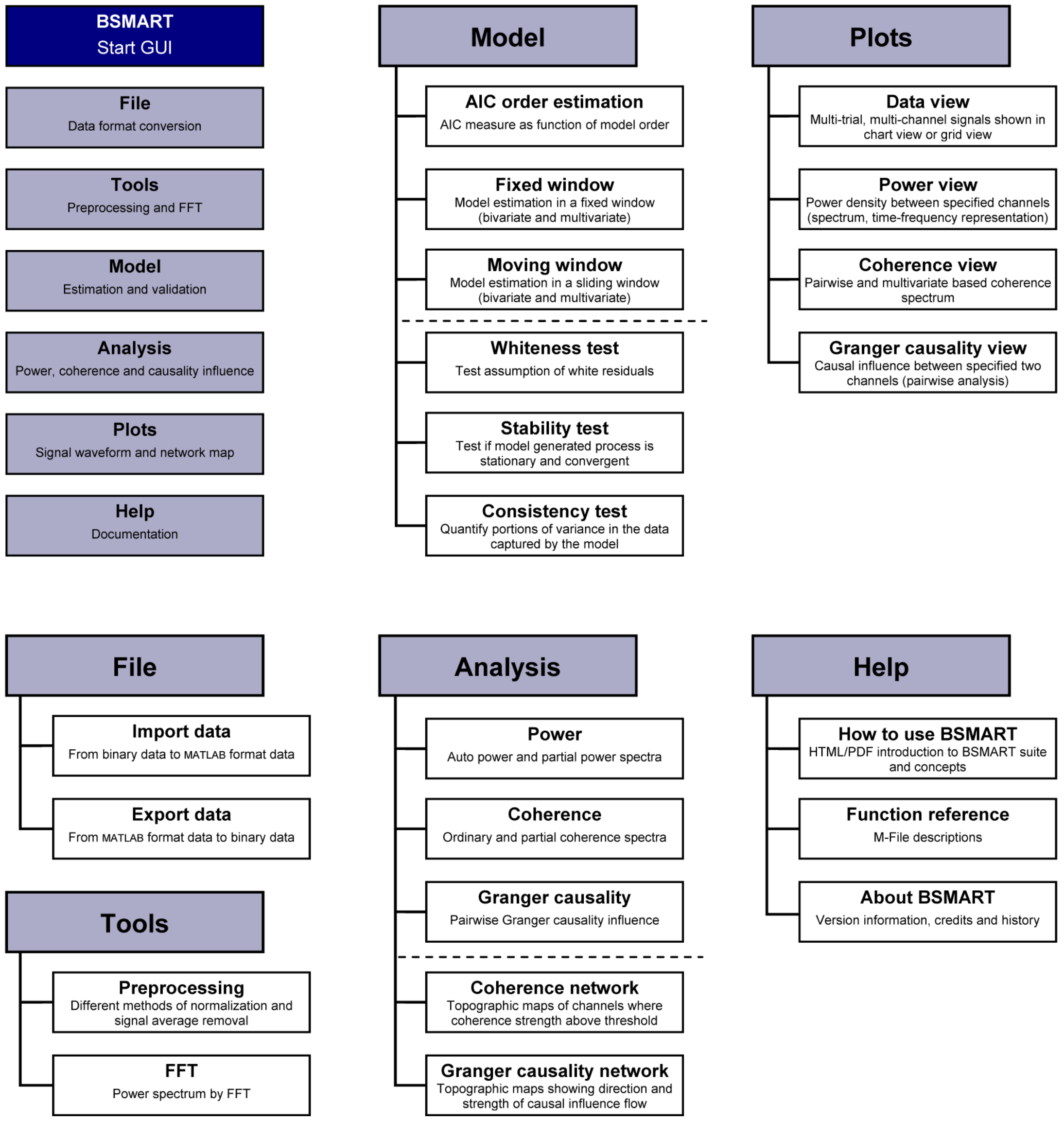
BSMART
Start GUI
File
Data format conversion
Tools
Preprocessing and FFT
Model
Estimation and validation
Analysis
Power, coherence and causality influence
Plots
Signal waveform and network map
Help
Documentation
Model
AIC order estimation
AIC measure as function of model order
Fixed window
Model estimation in a fixed window (bivariate and multivariate)
Moving window
Model estimation in a sliding window (bivariate and multivariate)
Whiteness test
Test assumption of white residuals
Stability test
Test if model generated process is stationary and convergent
Consistency test
Quantify portions of variance in the data captured by the model
Plots
Data view
Multi-trial, multi-channel signals shown in chart view or grid view
Power view
Power density between specified channels (spectrum, time-frequency representation)
Coherence view
Pairwise and multivariate based coherence spectrum
Granger causality view
Causal influence between specified two channels (pairwise analysis)
File
Import data
From binary data to MATLAB format data
Export data
From MATLAB format data to binary data
Tools
Preprocessing
Different methods of normalization and signal average removal
FFT
Power spectrum by FFT
Analysis
Power
Auto power and partial power spectra
Coherence
Ordinary and partial coherence spectra
Granger causality
Pairwise Granger causality influence
Coherence network
Topographic maps of channels where coherence strength above threshold
Granger causality network
Topographic maps showing direction and strength of causal influence flow
Help
How to use BSMART
HTML/PDF introduction to BSMART suite and concepts
Function reference
M-File descriptions
About BSMART
Version information, credits and history

Figure 2.
Top (shadowed) and second level menu structure of BSMART. The six main menus were designed to match different tasks of processing data and for common display functions. The dotted lines separate functionally different submenus.

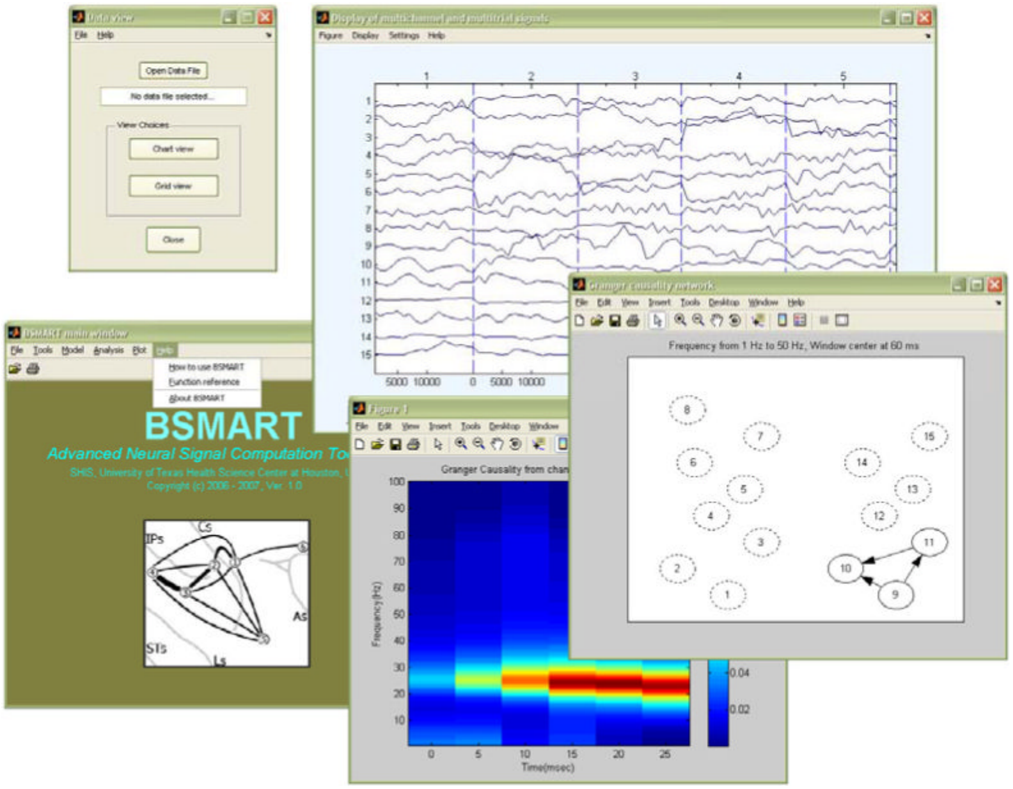
Data view
File Help
Open Data File
No data file selected...
View Choices
Chart view
Grid view
Close
Display of multichannel and multitrial signals
Figure Display Settings Help
BSMART main window
File Tools Model Analysis Plot Help
How to use BSMART
Function reference
About BSMART
BSMART
Advanced Neural Signal Computation To
SHIS, University of Texas Health Science Center at Houston
Copyright (c) 2006 - 2007, Ver. 1.0
IPs
Cs
As
STs
Ls
Figure 1
Granger Causality from chan
Frequency(Hz)
Time(msec)
Granger causality network
Frequency from 1 Hz to 50 Hz, Window center at 60 ms

Figure 3.
Screen capture of a sample BSMART session running under Windows. Users call different functional blocks from the GUI interface (lower left) and input parameters via ‘pop-up’ parameter selection window (upper left). The results can be visualized with different plots such as waveform plot, time-frequency plot and network map (center and right).

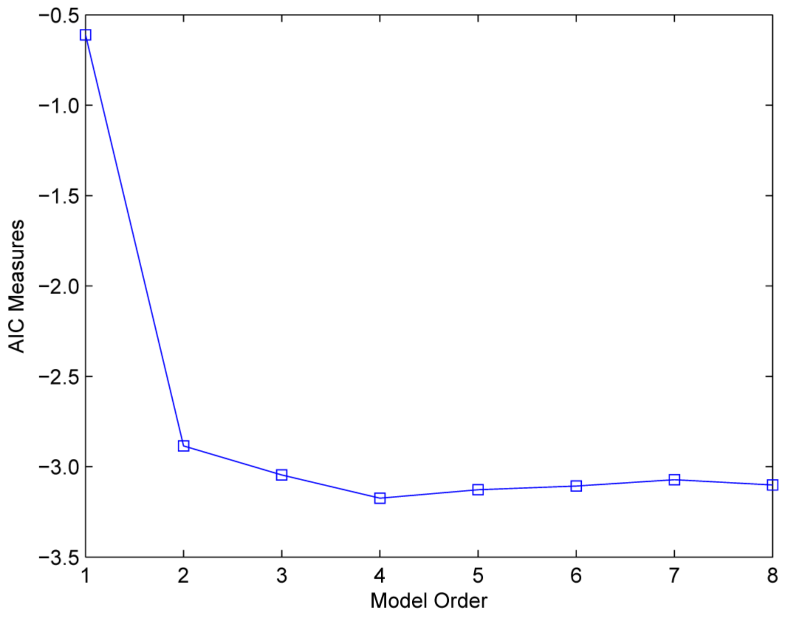

Figure 4.
The AIC as a function of model order, computed in a 60-ms time window (12 points at the sampling rate of 200 Hz) centered at 45 ms.

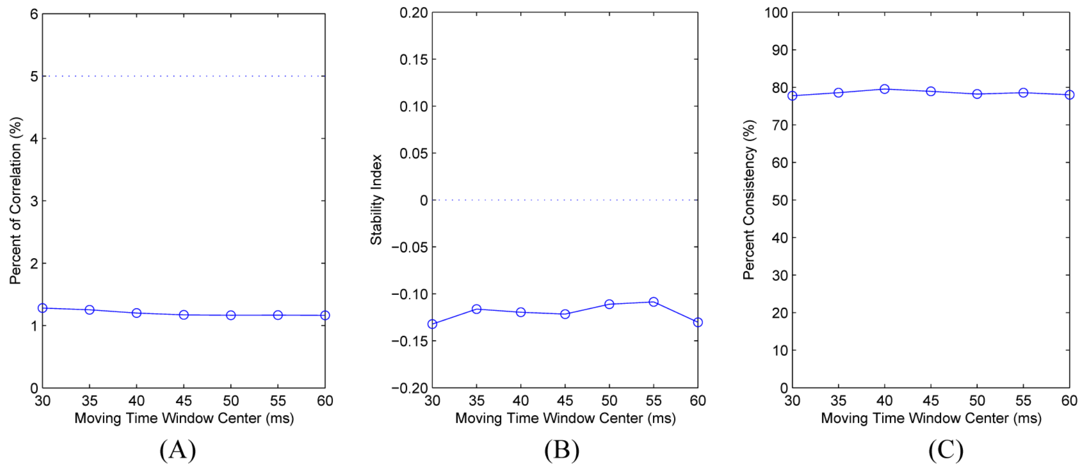

Figure 5.
Results of model validation for the sample data set: (A) whiteness test; (B) stability test; and (C) consistency test.

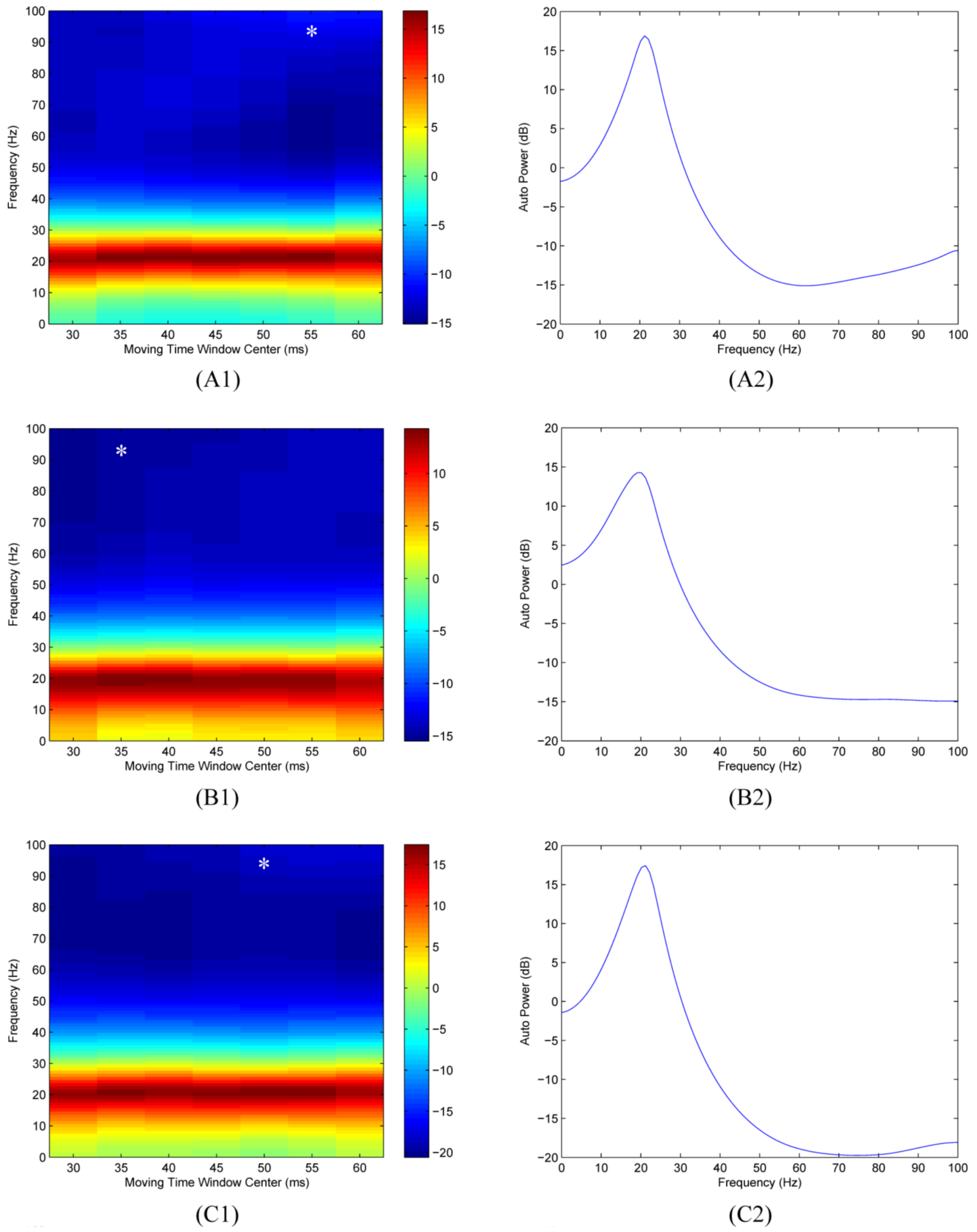

Figure 6.
Auto power of the sample data set obtained by using the AMAR model. Time-frequency plots are shown in the left column and single spectra in the right column. The times of the single spectra in the right column are indicated by asterisks in the time-frequency plots. In panel A, (A1) shows the time-frequency plot of channel 9, and (A2) shows the spectrum of the time window centered at 55 ms, where the maximum power peak is found. Panel B shows the time-frequency plot of channel 10 and the spectrum of the window centered at 35 ms. Panel C shows the time-frequency plot of channel 11 and the spectrum of the window centered at 50 ms.

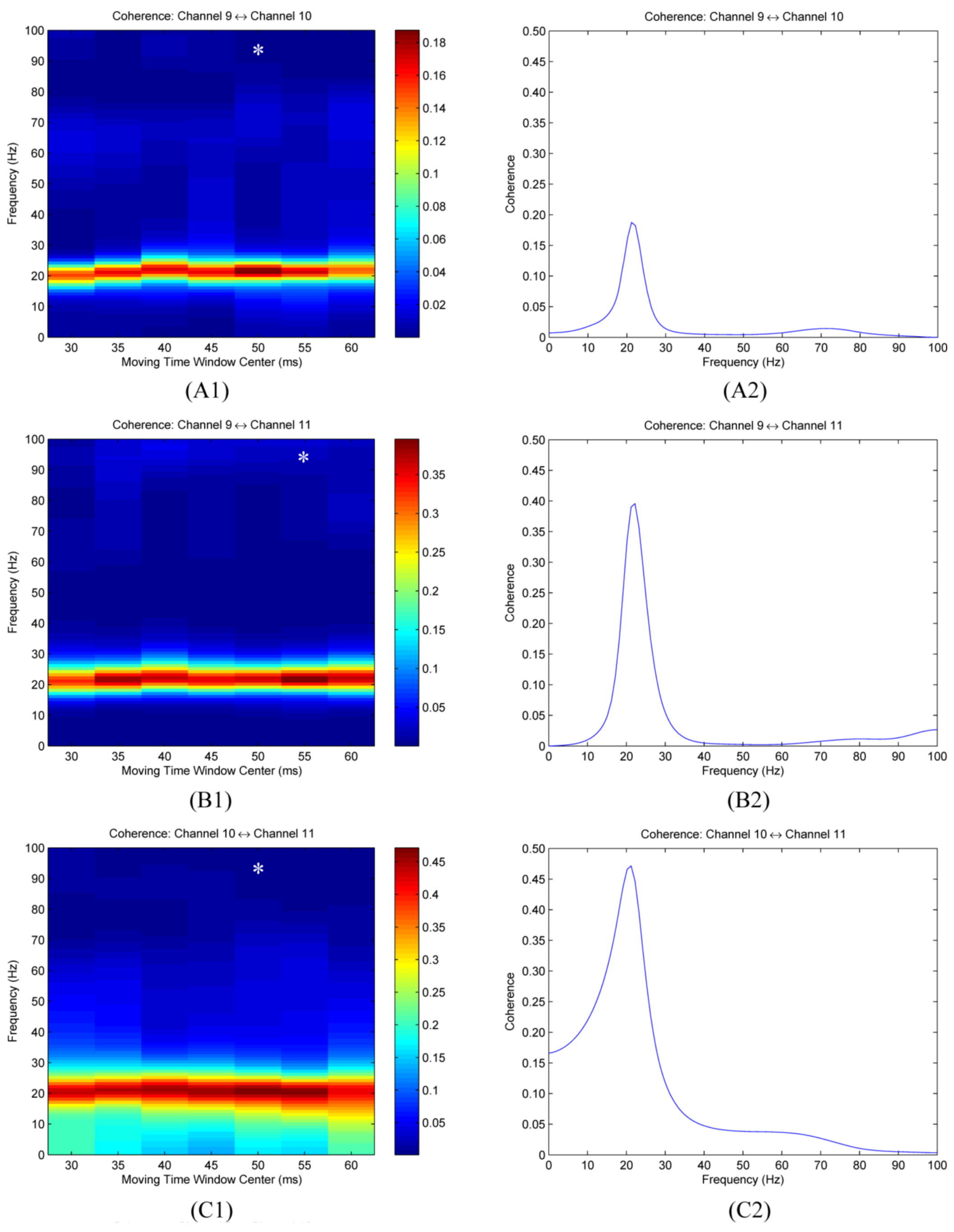

Figure 7.

Ordinary coherence spectra among channels 9, 10 and 11 of the sample data set obtained by using the AMAR model. Time-frequency plots are shown in the left column and single spectra in the right column. The times of the single spectra in the right column are indicated by asterisks in the time-frequency plots. In panel A, the time-frequency coherence plot between channels 9 and 10 is shown in (A1), and (A2) shows the specific coherence spectrum of the window centered at 50 ms, in which the maximum coherence is reached. Panel B shows the time-frequency coherence plot between channels 9 and 11 and the coherence spectrum of the window centered at 55 ms. Panel C show coherence between channels 10 and 11, and the coherence spectrum of the window centered at 50 ms.

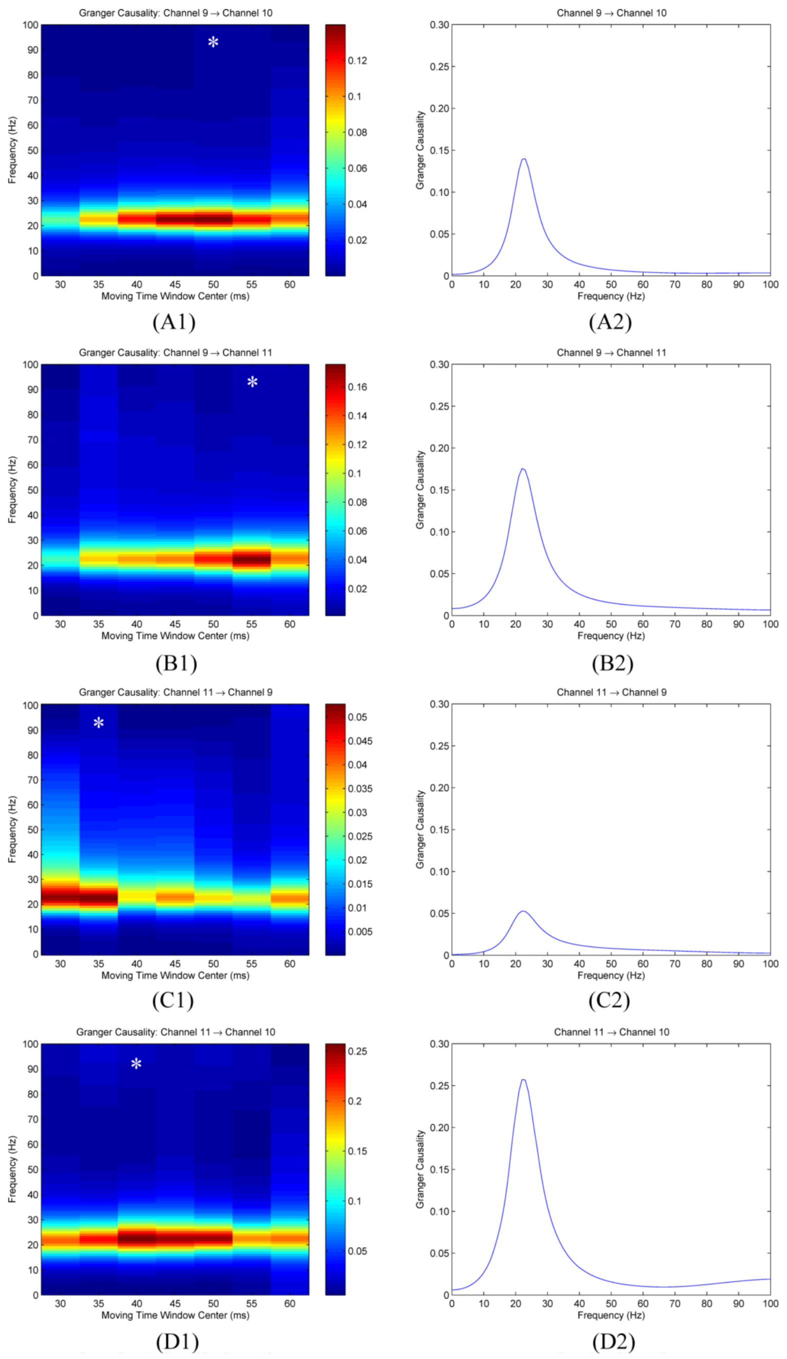

Figure 8.

Directional influence measured as Granger causality among channels 9, 10 and 11. Time-frequency plots are shown in the left column and single spectra in the right column. The times of the single spectra in the right column are indicated by asterisks in the time-frequency plots. Panel A shows the causality from Channel 9 to Channel 10, where (A1) plots the time-frequency Granger causality plot and (A2) shows the spectrum in the time window centered at 50 ms, in which the maximum Granger causality was found. The causality from channel 9 to channel 11 is shown in panel B, where (B2) is the causality spectrum at 55 ms. Panel C and panel D show the causal influence from channel 11 to channel 9 and channel 11 to channel 10, respectively. (C2) is the spectrum at 35 ms and (D2) is the spectrum at 40 ms.

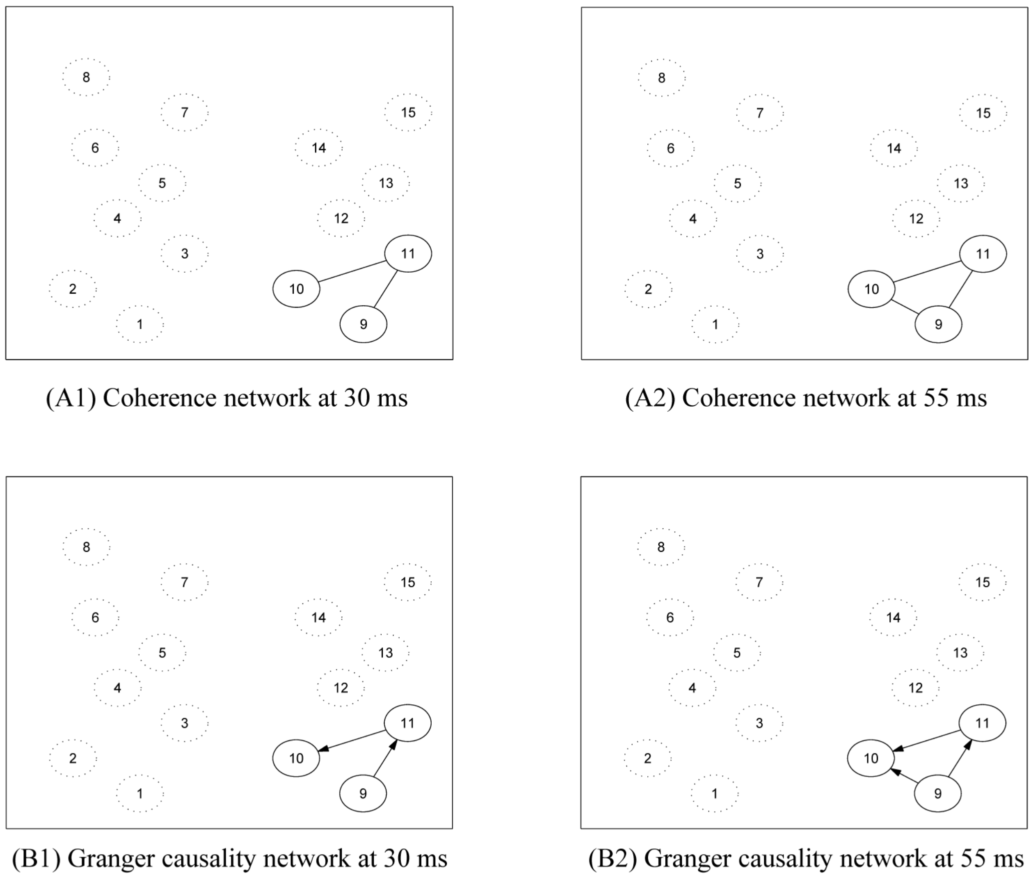
8
7
15
6
14
5
13
4
12
3
11
2
10
1
9
(A1) Coherence network at 30 ms
(A2) Coherence network at 55 ms
(B1) Granger causality network at 30 ms
(B2) Granger causality network at 55 ms

Figure 9.

Coherence and Granger causality networks are shown in panel A and panel B, respectively, where (A1) and (B1) display the networks for the time window centered at 30 ms, and (A2) and (B2) for the window centered at 55 ms. Channel symbols circled in solid lines denote the channels under examination.

Table I

Sample BSMART processing script. This script is an example of preprocessing steps. The Matlab data ‘dat’ (saved as ‘test71.mat’) contains the sample data set (described in the User's Guide and available for download). Script 1 loads the saved data set into Matlab working space. Script 2 and Script 3 complete the preprocessing steps proposed by Ding et al. (Ding et al., 2000). Specifically, function *pre_subt_divs()* in Script 2 removes the temporal mean from each LFP trial and divides it by the temporal standard deviation (Step (i) in Ding et al.). Function *pre_sube_divs()* in Script 3 performs the same action but uses ensemble mean and ensemble standard deviation (Steps (ii) and (iii)). Finally, Script 4 plots the waveforms of the preprocessed data set by calling the *sigplot()* drawing function with a parameter of sampling rate ‘fs’. Function *sigplot()* is modified from *eegplot()* in EEGLAB toolbox (Delorme & Makeig, 2004). Note the permutation operation performed before calling *sigplot()*. This is because *sigplot()* requires that the format of data be “channels×points× trials”, but ‘dat’ is in the format of “points×channels×trials”.

1.	»	load test71.mat
2.	»	dat1 = pre_subt_divs(dat); % (i) process temporal mean and STD
3.	»	dat2 = pre_sube_divs(dat1); % (ii), (iii) process ensemble mean and STD
4.	»	dat3 = permute(dat2,[2 1 3]); fs = 200; sigplot(dat3,‘srate’,fs); % chart view