Journal of Statistical Software

October 2011, Volume 44, Issue 13. *http://www.jstatsoft.org/*

FIAR: An R Package for Analyzing Functional Integration in the Brain

Bjorn Roelstraete
Ghent University

Yves Rosseel
Ghent University

Abstract

Functional integration in the brain refers to distributed interactions among functionally segregated regions. Investigation of effective connectivity in brain networks, i.e, the directed causal influence that one brain region exerts over another region, is being increasingly recognized as an important tool for understanding brain function in neuroimaging studies. Methods for identifying intrinsic relationships among elements in a network are increasingly in demand.

Over the last few decades several techniques such as Bayesian networks, Granger causality, and dynamic causal models have been developed to identify causal relations in dynamic systems. At the same time, established techniques such as structural equation modeling (SEM) are being modified and extended in order to reveal underlying interactions in imaging data. In the R package **FIAR**, which stands for Functional Integration Analysis in R, we have implemented many of the latest techniques for analyzing brain networks based on functional magnetic resonance imaging (fMRI) data. The package can be used to analyze experimental data, but also to simulate data under certain models.

Keywords: functional integration, functional magnetic resonance imaging, dynamic causal modeling, structural equation modeling, Granger causality.

1. Introduction

One of the most intruiging problems in simultaneous recordings of two or more signals from the nervous system is the detection of information flow such as causal relations and the timing between them. This principle of brain organization is known as functional integration (Friston *et al.* 2005). Functional integration refers to distributed interactions among functionally segregated regions. Studies of functional integration try to understand how regional responses are mediated by connections between brain areas and how these connections change with experimental manipulations or disease.

In the case of functional magnetic resonance imaging (fMRI) data this is especially challenging, since the neural data are blurred with the hemodynamic signal. The result is that the signal of interest is only measured in an indirect way. Furthermore, the relation between the neural signal and the measured hemodynamic signal differs from region to region and even from subject to subject (Aguirre *et al.* 1998; Handwerker *et al.* 2004), so the hemodynamic response function can only be estimated. Nevertheless, distinguishing between efferent and afferent connections in brain networks is crucial to construct formal theories of brain function. As a consequence, many statistical methods to study brain connectivity based on hemodynamic measurements have been developed.

The three most widely used methods to study functional integration are (1) dynamic causal modeling (DCM), (2) structural equation modeling (SEM), and (3) Granger causality (GC). In the past years, DCM (which is part of the Statistical Parametric Mapping software, **SPM**, Ashburner *et al.* 2008) has become a "gold standard" for studying effective connectivity between brain regions, i.e., the direct influence one brain region has over another (Friston 2009). DCM employs an explicit forward model for explaining which (neural) states caused the (hemodynamic) data. DCM assumes that hemodynamic signals are caused by changes in local neural activity, mediated by experimental inputs (e.g., the presentation of a visual stimulus or the instruction to attend to motion) and the distributed neural interactions among brain regions. DCM is based on a model of this distributed processing and is parameterized by the strength of coupling among the neural regions. This neural model is then supplemented with a hemodynamic model that converts the neural activity into predicted hemodynamic signals. The convolution or impulse response function, mapping from underlying neural activity to observed fMRI responses, is called a hemodynamic response function (HRF, Buxton *et al.* 1998). DCM runs in the MATLAB (The MathWorks, Inc. 2010) environment and to our knowledge, there is no package for R (R Development Core Team 2011) that can perform DCM.

Perhaps the most widely used method to study effective connectivity in the brain is SEM. Although developed in the fields of econometry and the social sciences, it has been successfully introduced into the neurosciences to study causal pathways in the brain (McIntosh and Gonzales-Lima 1994). Since its introduction, extensions of the classical SEM framework have been developed to specifically analyze time series data. For example, Kim and colleagues proposed a unified structural equation model (USEM) approach for modeling brain connectivity (Kim *et al.* 2007). USEM unifies a vector autoregressive (VAR) model (Harrison *et al.* 2003), represented by longitudinal pathways, and a conventional SEM, represented by contemporaneous pathways. Consequently, USEM is able to model the autoregressive nature within each time series and the correlations between the d-dimensional time series simultaneously.

USEM may be performed with standard SEM software such as **Mplus** (Muthèn and Muthèn 2004), **LISREL** (Jöreskog and Sörbom 2005), **EQS** (Bentler 1995), or some packages in R like **sem** (Fox 2010) or **lavaan** (Rosseel 2011). However, the autoregressive connectivity model and data matrix need to be specified manually. For networks with a large number of brain regions and a high autoregressive order, manually specifying the model is difficult. An R function extending the model and data matrix automatically to the desired autoregressive order and consecutively computing the model fit is not available.

A third very popular tool to study brain connectivity is Granger causality (Granger 1969). The idea of GC is that the causal influence of one time series on another can be conceived by the notion that the prediction of one time series is improved by incorporating knowledge about the other. Only very recently the standard framework for GC has been extended to the multivariate case, where predictor and dependent variables are no longer constrained to be univariate (Barrett *et al.* 2010).

There are several R packages like **lmtest** (Zeileis and Hothorn 2002) and **vars** (Pfaff 2008) that provide a GC test. Unfortunately, only bivariate relations may be examined where one time series is caused by one single other time series. However, when studying brain networks usually more than two regions are considered at the same time and repeated bivariate analysis may be misleading. For example, one time series may falsely appear to cause another if they are both influenced by a third time series but with a different delay. Therefore, it would be useful to be able to perform multivariate GC on fMRI time series. These multivariate GC techniques have not been implemented into an R package yet.

These facts led us to believe that there is a need for an assembled package to perform some of the most popular and recent techniques for studying functional integration in brain networks. We call this package **FIAR**, which stands for functional integration analysis in R. The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=FIAR` and currently covers DCM, (unified) SEM, (multivariate) GC. We plan to keep it up to date and even extend it with other techniques in the future. In the next section the implemented techniques are briefly discussed and the corresponding R functions demonstrated. In the final section, all techniques are applied to the attention to visual motion dataset that may be downloaded from the **SPM** website (`http://www.fil.ion.ucl.ac.uk/spm/data/`).

2. Dynamic causal modeling

2.1. Theoretical background

Whereas SEM and GC were developed in other areas of science, DCM has been specifically designed for the analysis of functional imaging data. In fMRI, the blood oxygen level-dependent (BOLD) signal is observed, while the signal of interest is the (hidden) neural activity in each region. DCM therefore estimates two models simultaneously: a causal model, indicating which neural activity in one region causes changes in activity in other regions and a hemodynamic model of how the fMRI signals were produced by complicated physiological events, initiated by changes in neural activity.

Overall, DCM models the temporal change in the neural activity $\dot{z}$ as a bilinear function of the current state z, the inputs u (usually the experimental design) and some neural coupling parameters A, B, and C:

$$\dot{z} = Az + \sum_{j=1}^{m} u_j B_j z + Cu \qquad (1)$$

where A represents the anatomical connections between brain regions (Friston *et al.* 2003). These connections can be seen as average connections between regions, irrespective of the inputs. The B connections are modulated by the inputs j (modulatory or functional connections) and can be added to the A connections in order to obtain the total strength of a connection under input j. Finally, C connections are the direct inputs to the nodes of the system. These parameters can be expressed as partial derivatives:

$$A = \frac{\partial \dot{z}}{\partial z}, \quad B_j = \frac{\partial^2 \dot{z}}{\partial u_j \partial z}, \quad C = \frac{\partial \dot{z}}{\partial u}. \tag{2}$$

DCM combines this neural model with a plausible and experimentally validated hemodynamic model (the so called Balloon model, Buxton *et al.* 1998; Friston *et al.* 2000) with six more parameters that describes the transformation from neural activity to BOLD activity (Stephan *et al.* 2007). The hemodynamic model is described in full detail in Friston *et al.* (2000). Combining the neural and hemodynamic states in a joint state vector x, and their respective parameters into a joint parameter vector θ, results in the state equation

$$\dot{x} = F(x, u, \theta), \tag{3}$$

which can be integrated and passed through the output nonlinearity λ to predict the BOLD signal y

$$y = \lambda(x). \tag{4}$$

This so called forward model is the basis for estimating the neural and hemodynamic parameters from the measured data. The term "causal" in DCM also refers to this forward model, because it models how hidden neural changes are causing the observed data. A fully Bayesian approach is used to estimate the neurodynamic and hemodynamic parameters (Friston *et al.* 2003). Experimentally validated priors are used for the hemodynamic parameters and conservative shrinking parameters for the neural coupling parameters, embedded in an expectation maximization (EM) algorithm.

Although DCM is probably the most sophisticated of the three models discussed here, we would like to point out two drawbacks. First, due to the large number of free parameters DCM is computationally demanding and this restricts analysis to a small number of regions. Also, the deterministic input output nature of the model only allows to model variations explained by the inputs. This forces all dynamic information to be represented in the design matrix. In other words, it assumes that all neural dynamics can be captured without error from the chosen inputs. This assumption makes DCM estimation very dependent on the exact number and form of the exogenous inputs. Recently, stochastic extensions of DCM have been presented (Daunizeau *et al.* 2011), but these are not yet widely used and are currently not implemented in **FIAR**.

Second, the specification of the prior neural and hemodynamic model will put restrictions on the information that can be captured by them. For instance, in DCM the hemodynamic model has much more affordance for delayed coherent variations than the neurodynamic model. Therefore, the delay will be put into the hemodynamics in the fitting of the model. Not because this is actually the case in the data at hand, but because the model has assumed this to be true (Roebroeck *et al.* 2011).

2.2. Software implementation

Model specification

A DCM analysis begins with the specification of the model. In **FIAR** the model may be specified in three ways. One way is to manually create a DCM list containing all necessary

Parameter	Description
n	Number of regions in the network (integer)
names	Names of regions in the network (string)
m	Number of experimental conditions (integer)
ons$ExpcondA	Onsets (in scans) of experimental condition A (vector)
ons$ExpcondB	Onsets (in scans) of experimental condition B (vector)
dur$ExpcondA	Duration (in scans) of experimental condition A (vector)
dur$ExpcondB	Duration (in scans) of experimental condition B (vector)
a	Prior anatomical connections (vector)
b	Prior functional connections (vector)
c	Prior input connections (vector)
h	Prior hemodynamic parameters (vector)
y	Time series (v by n matrix)
TR	Repetition time in seconds (integer)
TE	Echo time in seconds (integer)
T	Number of timebins (integer)
v	Number of scans (integer)

Table 1: Overview of parameters in DCM specification lists for model parameters (top) and scanner parameters (bottom), respectively.

model and scanner parameters, Table 1 for an overview and Appendix A for an example, which can be used as a template to construct DCM objects. The second and third way is to use the function:

```
dcmParam(a, b, c, ons = list(), dur = list(), v, n, m, TR,
  h = c(0.65, 0.41, 0.98, 0.32, 0.34, 0), names = c(),
  TE = 0.04, T = 16, x = 5 * n, HPF = 0, auto = FALSE)
```

This function allows one to enter the parameters step by step in an automated fashion when `auto = TRUE`:

```
R> DCM <- dcmParam(auto = TRUE)

enter number of regions: 3
enter name of region 1 :V1
enter name of region 2 :V2
...
```

or to manually enter all parameters when `auto = FALSE` (default). For a full overview how to specify a DCM, we refer to Appendix A where an example is presented.

Data generation

FIAR allows to generate data under a specified model with a desired signal to noise ratio (SNR) and autoregressive (AR) coefficient. The function `dcmGenerate` will create the simulated time series with length v (rows) of n regions (columns). When `SNR` is set to 0, the pure

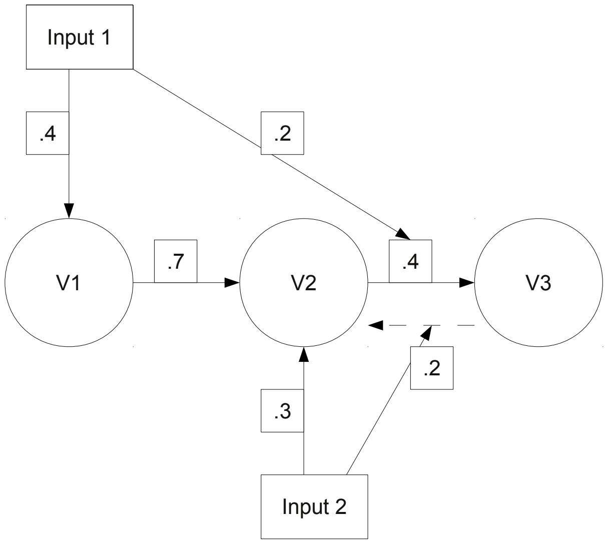

Figure 1: True connections in the example model.

signal is returned. When `SNR > 0` the signal is mixed with Gaussian white noise to achieve the specified SNR when `ar = 0`. When `ar > 0`, the noise will be autocorrelated with a coefficient `ar`. When `SNR = 0`, the argument `ar` has no function. Finally, `names` is a string vector that allows to give names to the regions. As an example we take the connectivity model in Figure 1.

The model contains three brain regions V1, V2, and V3. There is an anatomical connection between region V1 and V2 with a strength of 0.7 Hertz and an anatomical connection between region V2 and V3 of 0.4 Hertz. The first experimental input, which has an onset every 60 scans for a duration of 30 scans, directly influences region V1 with a strength of 0.4 Hertz and it also influences (modulatory influence) the connection between region V2 and V3 with a strength of 0.2 Hertz. The second experimental input, with an onset every 30 scans and a duration of 15 scans, influences region V2 with a strength of 0.3 Hertz and creates a functional pathway from region V3 to region V2 with a strength of 0.2 Hertz. The DCM object in Appendix A contains all necessary scanner and model parameters to specify the model. We can generate data from it as follows:

```
R> set.seed(11111112)
R> ts <- dcmGenerate(DCM, SNR = 1, ar = 0.2, names = c("V1", "V2", "V3"))
R> head(ts)
```

```
             V1          V2         V3
[1,] -4.4175334 -2.07116701 -2.9517392
[2,] -1.6208725 -1.57129370 -0.7291734
[3,] -2.3132625 -3.84259117 -0.8582669
[4,] -3.0271776  0.40017218  2.1827498
[5,]  0.3938042  0.06710791 -0.2695954
```

```
[6,]  4.4434802  1.92912678  0.2668325
...
```

This will produce three time series V1, V2, and V3 that are integrated as specified in `DCM$a`, `DCM$b`, and `DCM$c`, with a SNR = 1 and autocorrelation coefficient of 0.2.

Model estimation

As already mentioned, DCM is a Bayesian method, meaning that posterior model parameters are estimated based on prior information and the data. The function `dcmEstimate` takes as arguments the object `DCM`, which contains the model, and `ts`, which represents the time series. The estimation process results in posterior values of the model parameters and their probabilities. The object `ts` may be the simulated data or experimental data extracted from brain regions.

```
R> DCM <- dcmEstimate(DCM, ts = ts)

EM-step(-): 1    F: -1717.132   dF: 167.6796
EM-step(-): 2    F: -1611.116   dF: 161.8527
EM-step(-): 3    F: -1577.728   dF: 141.5641
EM-step(-): 4    F: -1488.630   dF: 19.82248
EM-step(-): 5    F: -1478.350   dF: 0.1039904
EM-step(-): 6    F: -1478.308   dF: 0.004626658
```

The posterior parameters are denoted by capital letters in analogy to the small letters of the priors. For example `DCM$A` are the posterior anatomical connections. They are represented by an $n \times n$ matrix where the colums represent the "from" region and the rows the "to" region. The unit of the connections is Hertz. For example, the following output

```
R> DCM$A

           V1         V2 V3
V1 -1.0000000  0.0000000  0
V2  0.4789123 -1.0000000  0
V3  0.0000000  0.3011896 -1
```

means that there is an inhibitory anatomical connection from region V1 to itself with a strength of 1 Hertz, an exhibitory anatomical connection from region V1 to region V2 with a strength of 0.48 Hertz, and so on. Only the off diagonal elements are important to interpret. The fields where we did not expect connections in our prior model are the fields that contain zero.

The probability that these connections differ from zero can be found in the `DCM$pA` field that is created during the estimation process.

```
R> DCM$pA
```

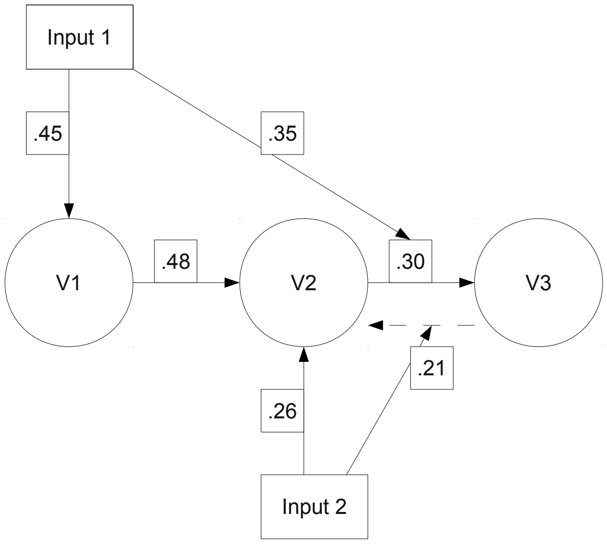

Figure 2: Posterior connections in the example model.

```
          V1        V2 V3
V1 0.000000 0.0000000  0
V2 0.999976 0.0000000  0
V3 0.000000 0.9992914  0
```

We have very strong evidence that the connections from V1 to V2 and from V2 to V3 are different from zero. The posterior connections after estimation are relatively close to the true parameters, as can be seen in Figure 2.

Based on the posteriors, we can compute the model evidence with the function `dcmEvidence`. This will add the fields `DCM$AIC` and `DCM$BIC` to the DCM object which produce the Akaike information criterion (Akaike 1973) and Bayesian information criterion (Schwartz 1978), respectively. For example:

```
R> DCM <- dcmEvidence(DCM, ts = ts)
R> DCM$AIC

[1] -784.4484

R> DCM$BIC

[1] -827.9564
```

We can do this for multiple DCMs and compare the fitvalues with the function

```
R> dcmCompare(DCM1, DCM2)
```

where DCM1 and DCM2 represent the different estimated DCMs that are being compared. The returned values of this latter function are Bayes factors (BF, Raftery 1995). This BF can be used to choose one model over another (see example below).

3. Structural equation modeling

3.1. Theoretical background

Structural equation modeling (SEM) was developed in the field of econometrics and the social sciences and first applied to neuroimaging data by McIntosh and Gonzales-Lima (1994). They comprise a set of regions with causal relations between them. Causal relations are thus not computed by the data but are assumed a priori. The strengths of the a priori connections are tuned in such a way that they minimize the discrepancy between the observed and expected instantaneous correlations between regions.

There are two major drawbacks to SEM for fMRI analysis. The first is that the observed correlations in the data are treated as if they reflect causal relations between neural activity. SEM rests upon a phenomenological model of dependencies among the (hemodynamic) data, without reference to how the data were caused at the neural level. The second drawback is that SEMs do not make use of temporal information. If the time indices of the time series were randomly permuted, this would not change the correlations between time series. In other words, SEM does not take into account the dynamics of the time series, but rather computes static correlations.

In the last years, extensions of this classical SEM framework have been developed to better fit the specific nature of fMRI data. One approach that is implemented in **FIAR** is unified SEM (Kim *et al.* 2007). USEM unifies a vector autoregressive (VAR) model (Harrison *et al.* 2003) and a conventional SEM. Consequently, USEM is able to model the autoregressive nature within each time series and the correlations between the d-dimensional time series simultaneously.

Specifically, longitudinal temporal relations are defined as relationships between brain regions involving different time points, and are represented in the form of a multivariate autoregressive model of order p, MAR(p). Conversely, contemporaneous relations reflect relationships between brain regions at the same time point, and involve conventional SEMs. Let $y_j(t)$ be the jth variable (e.g., the average BOLD intensity for the jth ROI) measured at time t, $j = 1, 2, \ldots, m$. The m-dimensional MAR(p) with an added component of contemporaneous relations can be written as:

$$y(t) = Ay(t) + \sum_{u=1}^{p} \phi(u)y(t-u) + \epsilon(t) \tag{5}$$

Here $y(t) = [y_1(t), y_2(t), \ldots, y_m(t)]^\top$ is the $(m\times 1)$ vector of observed variables measured at time t; $\epsilon(t) = [\epsilon_1(t), \epsilon_2(t), \ldots, \epsilon_m(t)]^\top$ is an $(m\times 1)$ vector of white noise with zero-mean and error covariance θ_ϵ; A is the parameter matrix of the contemporaneous relations, and $\Phi(i), i = 1, \ldots, p$ is a series of $(m\times m)$ parameter matrices representing the longitudinal relations. The diagonal elements in the $\Phi(i)$ represent the coefficients of the autoregressive process for each variable, and the off-diagonal elements represent the coefficients of the lagged relation between the variables.

If we denote ϕ as the set of free parameters of Φ, A, and θ, than these parameters are estimated so that the implied covariance matrix $\Sigma(\phi)$ is close enough to the sample covariance matrix S by minimizing a discrepancy function $F(S, \Sigma(\phi))$. The most widely used fitting function for general structural equation models is the maximum likelihood function defined by:

$$F_{ML} = \log|\Sigma(\phi)| + \mathrm{tr}[S\Sigma(\phi)^{-1}] - \log|S| - q, \tag{6}$$

with q the number of variables in the model (Kim *et al.* 2007).

3.2. Software implementation

The package **FIAR** contains the function `ARsem` which is a wrapper around the `sem` function from the R package **lavaan** (Rosseel 2011). The function takes as first argument the connectivity model of a classical SEM analysis. The model is specified as a vector that takes 1 when we assume a connection between 2 regions and 0 otherwise. The columns represent the "from" regions and the rows the "to" regions. The argument `data` should only contain the time series (rows) of the regions (colums) in the model. It is important that the time series are given a name. The third argument `order` represents the autoregressive (AR) order of the connectivity model. The function automatically transforms both model and data to the extended model of the specified AR order and the lagged dataset that it needs respectively. Notice that setting `order = 0` (default) is equivalent to performing a classical SEM analysis. Consider for example a three node network with a connection from region X to Y and from region Y to Z:

$$\begin{tikzcd} X_t \arrow[r] & Y_t \arrow[r] & Z_t \end{tikzcd} \tag{7}$$

SEM can be used to investigate causal relations between regions X, Y, and Z, but causality has a different meaning than in DCM. Causality in (7) is assumed a priori and tested as the presence of instantaneous correlations between regions. Since correlations are bidirectional, this SEM will produce the same model fit as the symmetrical model shown in (8).

$$\begin{tikzcd} Z_t \arrow[r] & Y_t \arrow[r] & X_t \end{tikzcd} \tag{8}$$

The causality tested in (7) is different than in (8), because it is assumed to be different a priori. Nevertheless, due to their symmetry the models will produce identically the same fit and can not be compared. Autoregressive SEM allows to compare symmetrical models and their different causal structures. In a first order USEM, Y at time t (Y_t) is no longer solely influenced by X at time t (X_t), but also by the previous state of X (X_{t-1}) and its own previous state (Y_{t-1}).

$$\begin{tikzcd} X_{t-1} \arrow[r] \arrow[d] \arrow[dr] & Y_{t-1} \arrow[r] \arrow[d] \arrow[dr] & Z_{t-1} \arrow[d] \\ X_t \arrow[r] & Y_t \arrow[r] & Z_t \end{tikzcd} \tag{9}$$

The arrow from X_t to Y_t is a contemporaneous arrow and reflects the covariance between the two ROIs. The contemporaneous arrows have the same meaning as in the classical approach. The extension is in the longitudinal arrows (e.g., $X_{t-1} \rightarrow X_t$), which try to model the autocorrelations within each region in a direct way. Notice that this model is no longer symmetrical to the AR model of order 1 we would obtain if we extend the model from (8). Hence, we can compare the models and test which causality structure is more likely given the data. Fitting the classical SEM from (7) in **FIAR** is performed via

```
R> model0 <- c(0, 0, 0, 1, 0, 0, 0, 1, 0)
R> fit0 <- ARsem(model0, data = semdata)
R> summary(fit0)
```

```
Lavaan (0.4-7) converged normally after 33 iterations

  Number of observations                          2000

  Estimator                                         ML
  Minimum Function Chi-square                 1340.675
  Degrees of freedom                                 1
  P-value                                        0.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                 Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  y_0 ~
    x_0             0.120    0.013    9.374    0.000
  z_0 ~
    y_0            -0.153    0.019   -8.127    0.000

Variances:
    y_0            91.814    2.903   31.623    0.000
    z_0            68.221    2.157   31.623    0.000
```

The example data `semdata` contains three variables. Based on the model, we test whether there is a connection from X to Y and from Y to Z. The function `ARsem` returns a fit object of class `lavaan` for which several methods are available, including a `summary` method. The standard output produces the model fit which is not good in this case ($\chi^2_1 = 1340, p < 0.001$) and the estimates for the connections ($X \rightarrow Y = 0.12$, $Y \rightarrow Z = -0.15$).

All new variables are of the form `originalname_order` and represent the lagged variables up to the specified order, with variables of order 0 being the original variables. To determine which AR order is suitable for the dataset at hand, we can use the function `ARorder()` by typing:

```
R> order <- ARorder(semdata)
R> order
```

```
[1] 3
```

The function `ARorder` fits all AR models from order `min` to order `max` and returns the order which produces the lowest AIC (Akaike 1973). The returned model order balances the variance accounted for, against the number of coefficients to be estimated. In our example, it appears that an AR(3) model best fits the data. We can test this USEM by typing:

```
R> fit3 <- ARsem(model0, data = semdata, order = 3)
R> summary(fit3)
```

The `summary` function produces the model fit and the estimates of all connections. In Appendix B we show the summary of the AR(3) model fit. It immediately shows that extending a simple model with only 3 nodes and 2 connections to an AR(3) model results in a lot of extra nodes (9) and connections (36). However, using the function `ARsem` there is no need to manually construct the lagged dataset nor the extended AR model.

We now also have two types of regression coefficients in the model. Next to the contemporaneous correlations (e.g., from X_0 to Y_0) as was the case for the classical SEM, the model also estimates autoregressive relations (e.g., X_1 to X_0). These estimates express the autocorrelation within one brain region up to the specified order (in this case 3).

If we want to compare the classical SEM and the AR(3) SEM in terms of model fit, we can compute both AICs:

```
R> AIC(fit0)

[1] 45774.25

R> AIC(fit3)

[1] 142437.3
```

and we see that the likelihood of the classical model (`fit0`) in this case is higher than that of the AR model (`fit3`).

4. Granger causality

4.1. Theoretical background

Wiener (1956) proposed a way to measure the causal influence of one time series on another by conceiving the notion that the prediction of one time series could be improved by incorporating knowledge about the other. Granger (1969) formalized this notion in the context of linear vector autoregression (VAR) modeling of stochastic processes (Guo *et al.* 2008). Although Granger causality (GC, sometimes called Wiener-Granger causality) was developed in the field of econometrics, it has recently received a lot of attention in the neuroscience community (Roebroeck *et al.* 2005; Guo *et al.* 2008; Deshpande *et al.* 2010; Bressler and Seth 2011).

Suppose X, Y, Z are univariate time series of length t. The joint autoregressive representation of Y and Z can be written as

$$Y_t = \sum_{i=1}^{p} a_i Y_{t-i} + \sum_{i=1}^{p} b_i Z_{t-i} + \epsilon_{1t}. \tag{10}$$

Checking whether X Granger causes Y given Z we can extend the model via

$$Y_t = \sum_{i=1}^{p} c_i Y_{t-i} + \sum_{i=1}^{p} d_i X_{t-i} + \sum_{i=1}^{p} e_i Z_{t-i} + \epsilon_{2t}, \tag{11}$$

and the Granger causality from X to Y conditioned on Z may be expressed as

$$F_2 = \ln\left(\frac{\mathsf{VAR}(\epsilon_{1t})}{\mathsf{VAR}(\epsilon_{2t})}\right). \tag{12}$$

As can be seen, the returned value is a type of F-statistic. If the information in the previous time points of X helps predicting the current time point in Y, we expect $\mathsf{VAR}(\epsilon_{2t})$ to be smaller than $\mathsf{VAR}(\epsilon_{1t})$, resulting in a F_2 value larger than zero. If X does not help predicting Y, $\mathsf{VAR}(\epsilon_{2t})$ and $\mathsf{VAR}(\epsilon_{1t})$ will be of the same magnitude, producing F_2 statistics around zero.

Only very recently, this GC measure has been defined for multivariate time series. If the dependent variable Y_t is no longer univariate, but a vector of observed responses measured at time t ($Y_t = Y_{1t}, Y_{2t}, \ldots, Y_{pt}$) then (12) is no longer useful. There have been two attempts to extend this formula to the multivariate case. The first was proposed by Ladroue *et al.* (2009) and uses the multivariate mean squared error or trace of the error covariance matrix, $\mathrm{tr}[\Sigma(\epsilon_t)]$, leading to:

$$\mathrm{F}_2 = \ln\left(\frac{\mathrm{tr}[\Sigma(\epsilon_{1t})]}{\mathrm{tr}[\Sigma(\epsilon_{2t})]}\right). \tag{13}$$

Another approach, originally proposed by Geweke (1982), is to use the generalized variance or determinant of the error covariance matrix, $|\Sigma(\epsilon_t)|$, which leads to:

$$\mathrm{F}_2 = \ln\left(\frac{|\Sigma(\epsilon_{1t})|}{|\Sigma(\epsilon_{2t})|}\right) \tag{14}$$

The advantage of (14) over (13) is that it asymptotically approximates a χ^2 distribution for large samples (Barrett *et al.* 2010). For small samples however, the distribution is unknown and resampling techniques are more appropriate to construct the null distribution.

When all variables are Gaussian, this solution is also fully equivalent to the transfer entropy $\mathcal{T}_{x\rightarrow y|z}$, an information-theoretic notion of causality (Barnett *et al.* 2009). For more properties and advantages of solution (14) over (13), we refer to Barrett *et al.* (2010). Conditional multivatiate GC (CMGC) is computed in **FIAR** based on (14) and is implemented as `condGranger`. Next to conditional MGC, **FIAR** allows to compute partial MGC (PMGC), introduced by Guo *et al.* (2008). PMGC differs from CMGC in the inclusion of the present conditioning variables Z in the joint autoregressive representation of Y and Z

$$Y_t = \sum_{i=1}^{p} a_i Y_{t-i} + \sum_{i=1}^{p} b_i Z_{t-i} + b^\top Z_t + \epsilon_{3t}. \tag{15}$$

Checking whether X partial Granger causes Y, given Z leads to

$$Y_t = \sum_{i=1}^{p} c_i Y_{t-i} + \sum_{i=1}^{p} d_i X_{t-i} + \sum_{i=1}^{p} e_i Z_{t-i} + e^\top Z_t + \epsilon_{4t}, \tag{16}$$

and the partial Granger causality from X to Y conditioned on Z may be expressed as

$$F_1 = \ln\left(\frac{|\Sigma(\epsilon_{3t})|}{|\Sigma(\epsilon_{4t})|}\right). \tag{17}$$

The F_1 statistic is again a ratio of two error variance/covariance matrices. If the logratio is significantly larger than zero, we conclude that the variables X partial Granger cause the variables Y, conditioned on the variables Z. As was the case for CMGC, the H_0 distribution of no causality is unknown. Therefore, resampling techniques must be used to construct confidence intervals around the F_1 estimator.

CMGC already controls for latent and/or exogenous influences to some extent, because the determinant of the error covariance matrix is sensitive to residual correlations. However, PMGC takes into account even more correlations, specifically to minimize the influence of sources of variation outside the model. When such influences are expected to be strong and uniform over all variables in the model, PMGC is to be preferred over CMGC (Barrett *et al.* 2010). PMGC is implemented in **FIAR** as `partGranger`. Notice that, when no conditional variables are included in the model, PMGC and CMGC will produce the same MGC measure.

Yet another type of GC test was presented in the work of Roebroeck *et al.* (2005). Geweke (1982) proposed a measure of linear dependence F_{xy} between x and y which implements GC in terms of VAR models. F_{xy} is the sum of three components

$$F_{xy} = F_{x\to y} + F_{y\to x} + F_{x.y}, \tag{18}$$

where $F_{x\to y}$ is the directed influence from x to y, $F_{y\to x}$ is the directed influence from y to x, and $F_{x.y}$ is the instantaneous influence. Roebroeck *et al.* (2005) used the difference between $F_{x\to y}$ and $F_{y\to x}$ as a GC measure for the influence region x has on region y. A drawback of this approach is that feedback connections are not modeled. On the other hand, studies have shown that, with sufficiently short repetition time (TR), this measure better controls for the loss of information that arises from the low-pass filtering introduced by the HRF than conditional GC (Roebroeck *et al.* 2005). This difference measure is implemented in **FIAR** as `diffGranger` and its partial counterpart (based on PMGC) as `pdiffGranger`.

GC is yet another type of causality than the DCM and SEM notion of causality. GC is based on temporal precedence in the sense that previous observations in one time series should help predicting the current observation in a second time series. Furthermore, this additive information should reach statistical significance before we have evidence that time series one Granger causes time series two (Bressler and Seth 2011).

The biggest difference between MGC and the other two methods is that MGC can be used in an exploratory way. There is no need to construct an a priori connectivity model. In the case there is very little known about the connectivity model under investigation, this can be seen as an advantage over DCM and SEM.

The drawbacks of this method for use with fMRI data are similar as is the case for SEM. That is, there is no (hemo)dynamic model involved, so correlations are computed between BOLD signals as if they were neural signals. Also, in their current implementation, USEM and GC are restricted to modeling linear relations, which may seem a strong assumption when modeling brain connectivity. However, nonlinear systems often have extensive linear regimes, and when dealing with large-scale interactions linear approximations are found to work extremely well (Bressler and Seth 2011).

4.2. Software implementation

The function `condGranger` computes the conditional GC of a set of predictor variables X (one or more) on a set of dependent variables Y (one or more), conditional on a third set of variables Z (zero or more).

There is a logical argument `boot` in the function controlling the bootstrap procedure. When `boot = FALSE`, the GC function merely returns the F-like statistic. When `boot = TRUE`, the F-statistic is returned together with a bootstrap approximation of F and a bootstrap bias and standard deviation (SD). This allows to make inference on the F statistic based on the bootstrap H_0 distribution. When the original F-statistic falls within ±2SD around the bootstrap mean the newly created field `fit$sig` will take the value 0, and 1 otherwise.

The bootstrap is performed by the `tsboot` function from the package **boot** (Canty and Ripley 2011). The block length of the resampling chuncks of the time series is optimized by the `b.star` function from the package **np** (Hayfield and Racine 2008). The argument `bs` is by default set to 100 and represents the number of bootstrap samples that are taken.

It is very important how the data are entered in the function. The first `nx` columns (default `nx = 1`) are the predictor variables, followed by the `ny` (default `ny = 1`) dependent variables, and finally the variables we want to condition on. It is important to respect this order. After the data matrix has been constructed correctly, the number of predictor (`nx`) and dependent (`ny`) variables need to be specified. For example

```
R> head(grangerdata)
```

```
              x          y         z          q           w
[1,]  3.7509252  0.2692284  1.293244  0.3135299 -0.28840307
[2,]  0.7023175 -1.8135786 -2.798717 -5.2041356 -3.85116801
[3,]  5.6832003  9.0572561  7.152931  3.8353702  9.18229397
[4,]  4.0451700 -4.0115360 -4.174242  0.4144000 -2.03339211
[5,] -1.8275448 -0.5157044 -3.132472 -6.4748229 -4.93447381
[6,] -6.8240911  1.6598595 -3.148736 -5.2256550  0.01632796
```

```
R> ARorder(grangerdata, max = 10)
```

```
[1] 3
```

```
R> condGranger(grangerdata, nx = 1, ny = 2, order = 3)
```

```
[1] 0.6457208
```

This tests whether variable x multivariate Granger causes variables y and z, conditioned on variables q and w. When `boot = TRUE`, not only the original F-value is returned, but also a bootstrap bias and standard error. This allows to construct bootstrap confidence intervals.

```
R> set.seed(22222223)
R> fit <- condGranger(grangerdata, nx = 1, ny = 2, order = 3, boot = TRUE)
R> fit
```

```
STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 70

Call:
tsboot(tseries = data, statistic = condGranger, R = bs, l = l,
    sim = "geom", nx = nx, ny = ny, order = order)


Bootstrap Statistics :
     original     bias    std. error
t1* 0.6457208 -0.4498475  0.04592638
```

This bootstrap results in an F_2 statistic that is $0.65 - 0.45 = 0.2$ with a standard deviation of 0.046. The probability that the original statistic of 0.65 comes out of this null distribution is therefore very small. Hence, we reject the null hypothesis of no effect and conclude that variable x Granger causes variables y and z, conditional on variables q and w. This is also reflected in the value of the created field `fit$sig`:

```
R> fit$sig

[1] 1
```

Calculating partial MGC is very analogous. Again, the function can be used with or without the bootstrap. For example:

```
R> partGranger(grangerdata, nx = 1, ny = 2, order = 3)

[1] 0.8596885

R> set.seed(33333334)
R> fit <- partGranger(grangerdata, nx = 1, ny = 2, order = 3, boot = TRUE)
R> fit

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 70

Call:
tsboot(tseries = data, statistic = partGranger, R = bs, l = l,
    sim = "geom", nx = nx, ny = ny, order = order)


Bootstrap Statistics :
     original     bias    std. error
t1* 0.8596885 -0.6554322  0.05589294
```

The bootstrap tells us that it is very unlikely that the F_1 statistic of 0.86 comes from the null distribution, so we conclude that variable x partial Granger causes variables y and z, conditional on variables q and w. Finally, `diffGranger` computes the difference between the MGC measures $F_{x \rightarrow y|z}$ and $F_{y \rightarrow x|z}$. Its partial counterpart is implemented as `pdiffGranger`. Both functions can be used with the bootstrap option.

```
R> set.seed(44444445)
R> fit <- diffGranger(grangerdata, nx = 1, ny = 2, order = 3, boot = TRUE)
R> fit

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 70

Call:
tsboot(tseries = data, statistic = diffGranger, R = bs, l = 1,
    sim = "geom", nx = nx, ny = ny, order = order)


Bootstrap Statistics :
     original    bias    std. error
t1* 0.6432665 -0.46078  0.05272068
```

The interpretation of this measure is somewhat different than for the other two types of GC. The measure of 0.643 is a relative measure and expresses the difference between the CMGC from variable X to variables Y and Z minus the CMGC from variables Y and Z to X. Since we know that the CMGC measure for the first causal path is 0.645, this implies that there is almost no CMGC in the other direction. Hence, X Granger causes Y and Z far more than the other way around.

5. Application to attention to visual motion data set

In this section we demonstrate the package **FIAR** on the example data set "Attention to visual motion" from the **SPM** website (http://www.fil.ion.ucl.ac.uk/spm/data/). This data was obtained by Buchel and Friston (1997). The experiment consisted of four conditions: (i) fixation (F), (ii) static (S), non-moving dots), (iii) no attention (N, moving dots but no attention required), and (iv) attention (A). The GLM analyses showed that activity in area V5 was not only enhanced by moving stimuli, but also by attention to motion. This effect in V5 was modeled using a DCM. Details about the experiment and the design parameters can be found in the **SPM**8 manual (Ashburner *et al.* 2008).

After the GLM analysis, three regions were extracted for use in a DCM analysis: region V5 ($-45/-81/-9$), region V1 ($-6/-84/-6$), and region SPC ($-18/-57/-66$). These regions were found to be highly activated during the experiment and their role in processing photic stimuli (V1), moving stimuli (V5) and attention to stimuli (SPC) was further investigated in the connectivity model depicted in Figure 3.

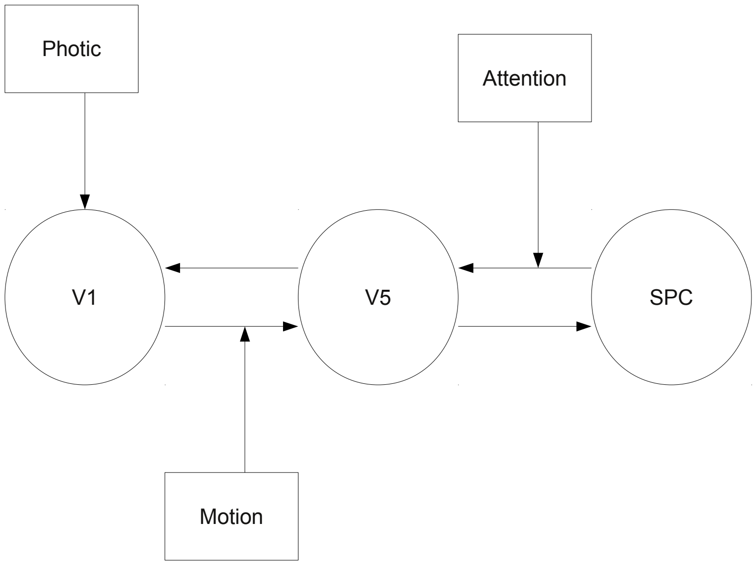

Figure 3: Model tested based on attention to visual motion data set.

5.1. DCM analysis

This is the object necessary to fit the DCM in **FIAR**:

```
R> DCM <- list()
R> DCM$a <- c(0, 1, 0, 1, 0, 1, 0, 1, 0)
R> DCM$b <-  c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
+    0, 0, 0, 1, 0, 0, 0)
R> DCM$c <- c(1, 0, 0, 0, 0, 0, 0, 0, 0)
R> DCM$h <- c(0.65, 0.41, 0.98, 0.32, 0.34, 0)
R> DCM$v <- 360
R> DCM$n <- 3
R> DCM$names <- c("V1", "V5", "SPC")
R> DCM$m <- 3
R> DCM$TE <- 0.04
R> DCM$T <- 16
R> DCM$ons <- list()
R> DCM$ons$photic <- c(10, 50, 100, 140, 210, 250, 300, 340, 30, 70, 120,
+    160, 190, 230, 280, 320, 80, 170, 260, 350)
R> DCM$ons$motion <- c(10, 50, 100, 140, 210, 250, 300, 340, 30, 70, 120,
+    160, 190, 230, 280, 320)
R> DCM$ons$attention <- c(10, 50, 100, 140, 210, 250, 300, 340)
R> DCM$dur <- list(photic = 10, motion = 10, attention = 10)
R> DCM$TR <- 3.22
R> DCM$x <- 5 * DCM$n
R> DCM$X0 <- X0
```

and the time series:

```
R> ts <- attentiondata
```

The last field (`DCM$X0`) need to be imported from the **SPM** analysis. This field represents the filtered and whitened design matrix and can be found in the field `xY.X0` from the extracted volumes of interest (VOI) in **SPM**. The field `ts` contains the data from the extracted VOIs and can be found in the fields Y from the extracted VOIs in **SPM**. Every column of `ts` represents the time series in one region of the network and the order should correspond to the order in which the model parameters `DCM$a`, `DCM$b`, and `DCM$c` are specified. Here the order is V1, V5, SPC. After specifying the model, we fit the data to the model:

```
DCM <- dcmEstimate(DCM, ts = ts)

EM-step(-): 1    F: -2567.886   dF: 188.9678
EM-step(-): 2    F: -2408.919   dF: 261.6301
EM-step(-): 3    F: -2221.106   dF: 259.8746
EM-step(-): 4    F: -2099.188   dF: 89.50041
EM-step(-): 5    F: -2051.462   dF: 8.861207
EM-step(-): 6    F: -2045.049   dF: 7.245513
EM-step(-): 7    F: -2038.092   dF: 11.93732
EM-step(-): 8    F: -2026.731   dF: 17.38173
EM-step(-): 9    F: -2011.171   dF: 17.73028
EM-step(-): 10    F: -1997.026   dF: 9.343546
EM-step(-): 11    F: -1990.633   dF: 1.961318
EM-step(-): 12    F: -1989.419   dF: 0.1684295
EM-step(-): 13    F: -1989.297   dF: 0.00902226
```

After estimation, all connections in the model can be reviewed:

```
R> DCM$A

            V1         V5        SPC
V1  -1.0000000  0.5974770  0.0000000
V5   0.1532816 -1.0000000  0.3964244
SPC  0.0000000  0.4076706 -1.0000000
```

```
R> DCM$pA

          V1        V5       SPC
V1  0.000000 0.9999931 0.0000000
V5  0.976364 0.0000000 0.9603819
SPC 0.000000 0.9999998 0.0000000
```

For example, we find a connection from region V1 to region V5 of 0.15 Hertz and the probability that this is significantly different from zero is 0.97.

In order to test how likely this model is compared to other possible models, we can define a second model, say `DCM2`, which has no feedback connection between V5 and V1, and then compare the fit of the two models:

```
R> DCM2 <- DCM
R> DCM2$a <- c(0, 0, 0, 1, 0, 1, 0, 1, 0)
R> DCM2 <- dcmEstimate(DCM2, ts = ts)

EM-step(-): 1    F: -2567.886   dF: 188.9678
EM-step(-): 2    F: -2408.919   dF: 261.6301
EM-step(-): 3    F: -2220.86   dF: 177.5000
EM-step(-): 4    F: -2120.842   dF: 39.33094
EM-step(-): 5    F: -2096.414   dF: 12.92199
EM-step(-): 6    F: -2084.984   dF: 18.37007
EM-step(-): 7    F: -2067.615   dF: 24.78346
EM-step(-): 8    F: -2045.447   dF: 23.44739
EM-step(-): 9    F: -2026.954   dF: 10.67108
EM-step(-): 10    F: -2019.872   dF: 1.738378
EM-step(-): 11    F: -2018.860   dF: 0.1583798
EM-step(-): 12    F: -2018.776   dF: 0.01426696
EM-step(-): 13    F: -2018.771   dF: 0.002211164
```

We compute the evidence (AIC and BIC) of both models and compare them:

```
R> DCM <- dcmEvidence(DCM, ts = ts)
R> DCM2 <- dcmEvidence(DCM2, ts = ts)
R> dcmCompare(DCM, DCM2)

 AIC overall Bayes Factor

      BF:  2.009479e+12

 BIC overall Bayes Factor

      BF:  287890048360
```

The Bayes factor (BF) expresses the evidence for one model over another. In general, a BF larger than 20 is seen as strong evidence for the first model entered, while a BF lower than 0.05 is seen as strong evidence for the second model entered (Raftery 1995). In our example, we find a very large BF, so we conclude that the model with the feedback loop from V1 to V5 (DCM) is to be preferred over the other model (DCM2). This does not imply that the model with the connection from V5 to V1 is the "true" model. It merely implies that the data suggest a connection, rather than no connection, given the priors of the DCM forward model. In practice, many models are theoretically possible and could be compared to each other using the BF. One way to find further evidence for this model is to test it with a different statistical method.

5.2. USEM analysis

If we want to test this connectivity model using USEM, we have to specify it in a different way. USEM is a stochastic model, so we can not incorporate the input connections and modulatory connections directly in the model. Only the anatomical connections can be specified. In order to estimate the above mentioned model with USEM, we have to specify the connectivity matrix as follows:

```
R> Model1 <- c(0, 1, 0, 1, 0, 1, 0, 1, 0)
```

where the rows and colums correspond to region V1, V5 and SPC respectively.

The time series need to be transformed as well. When testing correlations between brain regions, only those time points corresponding to the experimental condition should be extracted (Honey *et al.* 2002). The function `SEMextract` has been provided for this purpose. The function takes as arguments `ts`, the time series, `ons`, the onsets of the experimental condition to be tested, and `dur`, the duration of the activation condition of interest. For convenience, we use the vector of onsets and duration specified earlier in the DCM object:

```
R> ts_photic <- SEMextract(DCM$attentiondata, ons = DCM$ons$photic,
+    dur = DCM$dur$photic)
```

We start with fitting the data to a classical SEM:

```
R> fit1 <- ARsem(Model1, data = ts_photic)
```

However, when we fit the model, **lavaan** produces an error. This is to be expected given the data at hand. The three time series form a 3×3 covariance matrix with six data points $(3 \cdot (3+1)/2)$. However, in the model, we need to estimate four anatomical connections and three error variances (all regions are endogenous), leading to seven free parameters for only six data points. This means this specific model is underdetermined (i.e., negative degrees of freedom) and can not be estimated with SEM.

One way to resolve this is to specify the feedback connections as one connection. Remember that SEM computes covariances between time series. If there is a feedback loop between region V1 and region V5, this should be reflected in a covariance between both regions. The same holds for the feedback loop between V5 and SPC. Therefore, we specify the following model:

```
R> Model2 <- c(0, 0, 0, 1, 0, 0, 0, 1, 0)
```

In this model, we only need to estimate two connections and two error variances (for V5 and for SPC) with six data points. Again, we fit the data to the model:

```
R> fit2 <- ARsem(Model2, data = ts_photic)
R> summary(fit2)
```

```
Lavaan (0.4-7) converged normally after 15 iterations

  Number of observations                            220

  Estimator                                          ML
  Minimum Function Chi-square                     2.329
  Degrees of freedom                                 1
  P-value                                        0.127

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  V5_0 ~
    V1_0              0.696    0.040   17.494    0.000
  SPC_0 ~
    V5_0              0.617    0.030   20.397    0.000

Variances:
    V5_0              1.380    0.132   10.488    0.000
    SPC_0             0.664    0.063   10.488    0.000
```

The model fit is satisfactory ($\chi^2_1 = 2.3$, $p = 0.13$) and we see strong correlations between region V1 and V5 (0.70) and between region V5 and SPC (0.62). The information on the direction of activation is lost (no feedback loops), but the correlations between regions under influence of visual presentation are preserved in the model. If we want to extend this classical SEM in order to estimate autoregressive connections, we can type:

```
R> fit3 <- ARsem(Model2, data = ts_photic, order = 1)
R> summary(fit3)

 Lavaan (0.4-7) converged normally after 24 iterations

  Number of observations                           219

  Estimator                                         ML
  Minimum Function Chi-square                   34.710
  Degrees of freedom                                 6
  P-value                                        0.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  V1_0 ~
    V1_1              0.506    0.062    8.183    0.000
  V5_0 ~
```

```
    V1_0              0.716    0.045   15.997    0.000
    V1_1             -0.205    0.065   -3.180    0.001
    V5_1              0.174    0.072    2.401    0.016
  SPC_0 ~
    V5_0              0.611    0.032   18.886    0.000
    V5_1             -0.047    0.053   -0.891    0.373
    SPC_1             0.082    0.079    1.043    0.297
  V5_1 ~
    V1_1              0.613    0.038   16.069    0.000
  SPC_1 ~
    V5_1              0.481    0.030   16.154    0.000

Variances:
    V1_0              2.993    0.286   10.464    0.000
    V5_0              1.313    0.125   10.464    0.000
    SPC_0             0.661    0.063   10.464    0.000
    V5_1              1.142    0.109   10.464    0.000
    SPC_1             0.484    0.046   10.464    0.000
```

This fits a SEM of order AR(1). We see that the model fit is worse than in the classical SEM ($\chi^2_6 = 34$, $p < 0.05$). This is also reflected in a much higher AIC for the AR(1) model than for the classical model:

```
R> AIC(fit2)

[1] 2158.489

R> AIC(fit3)

[1] 4099.306
```

Although this simplified model is not identical to the feedback model tested with DCM, we again find evidence for connections between V1 and V5, and V5 and SPC respectively.

5.3. Granger analysis

Finally, we can investigate the functional integration between the three regions using GC. Since GC is an exploratory method, we do not need to specify an a priori connectivity model. It suffice to test the connections one at a time. For example, testing whether region V1 Granger causes region V5, conditioned on region SPC is done like this:

```
R> set.seed(5555556)
R> cgc <- condGranger(ts, boot = TRUE)
R> cgc

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 57
```

```
Call:
tsboot(tseries = data, statistic = condGranger, R = bs, l = l,
    sim = "geom", nx = nx, ny = ny, order = order)


Bootstrap Statistics :
      original      bias    std. error
t1* 0.05849308 -0.01426982  0.03079304

R> cgc$sig

 [1] 0

R> set.seed(6666667)
R> pgc <- partGranger(ts, boot = TRUE)
R> pgc

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 57

Call:
tsboot(tseries = data, statistic = partGranger, R = bs, l = l,
    sim = "geom", nx = nx, ny = ny, order = order)


Bootstrap Statistics :
      original      bias    std. error
t1* 0.03631089 0.003152783   0.0312808

R> pgc$sig

[1] 0

R> set.seed(7777778)
R> dgc <- diffGranger(ts, boot = TRUE)
R> dgc

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 57

Call:
tsboot(tseries = data, statistic = diffGranger, R = bs, l = l,
    sim = "geom", nx = nx, ny = ny, order = order)

Bootstrap Statistics :
      original      bias    std. error
t1* 0.01221986 -0.02397381  0.03105526
```

```
R> dgc$sig

 [1] 0
```

We find no evidence that region V1 Granger causes region V5. Just as was the case for USEM, computation of GC requires specification of the model order. Too low order can lead to a poor representation of the data, whereas too high order can lead to problems of model estimation. In fMRI data this order is usually very high. The exact AR order of the data may be computed using `ARorder`. Notice that in all GC analyses of the real data set we used the default model order of one. We can compute that the actual order of the data is 89. However, testing GC for model order 89 leads to a singular solution and can not be tested. In order to test if we can find a connection from region V1 to region V5 for a much higher model order, we tested a model of order 50.

```
R> set.seed(0000001)
R> cgc <- condGranger(ts, order = 50, boot = TRUE, bs = 100)
R> cgc

STATIONARY BOOTSTRAP FOR TIME SERIES

Average Block Length of 57

Call:
tsboot(tseries = data, statistic = condGranger, R = bs, l = 1,
    sim = "geom", nx = nx, ny = ny, order = order)

Bootstrap Statistics :
     original      bias    std. error
t1* 0.4585923 -0.04401649  0.08822636

R> cgc$sig

[1] 0
```

Even with model order 50, we find no evidence that region V1 Granger causes region V5.

5.4. Overview of results

DCM, SEM, and GC were applied to a real-world application. The example data set "Attention to visual motion" from the **SPM** website was analyzed with the three methods as a practical example. We focused on the feedback loop between V1 and V5.

The DCM results showed strong evidence for a model with feedback loops between V1 and V5, and V5 and SPC. Comparison between this model and a competing model with no connection between V5 and V1, confirmed the evidence for the feedback model. This does not imply that the feedback model is the "true" model. It merely implies that the data suggest such connections, given the assumptions of DCM. Other biologically plausible models could be tested and compared to each other in the same way.

The feedback model could not be tested with SEM, due to negative degrees of freedom. Simplifying the model by removing the feedback connections lead to similar conclusions as with the DCM analysis. Regions V1, V5, and SPC are exchanging information when visual stimuli are presented.

Finally, we tested the relations between the three regions with conditional, partial and difference GC. The bootstrap results of all three GC methods indicated that there is no evidence that region V1 Granger causes region V5, irrespective of the model order used. This is opposite to the DCM and SEM results.

These results show what often will happen when analyzing the same data set with different methods. In this example, a DCM user will find strong evidence for the existence of a causal path from V1 to V5 under presentation of a visual stimulus. She will even find evidence for a feedback loop going from V5 to V1 based on a model comparison. The GC user will find no evidence that V1 Granger causes V5 with a model of order 1. He might start testing more models with higher orders, but would still not find evidence that V1 adds information to the prediction of V5, given past information of V5. The exclusive SEM user would have to solve the problem of an underdetermined model, which means testing a simplified and thus different covariance structure.

That these different methods lead to different results is partly due to the different types of causality that are being measured. We believe that testing these different notions of causality and finding converging evidence between them should become standard practice when studying causal networks in the brain.

6. General conclusion

We presented the package **FIAR** which allows one to perform functional integration with many of the latest techniques such as unified SEM, multivariate GC, and DCM. All of these techniques are widely used to investigate functional brain connectivity, but none were implemented in R to date.

In the paper we discussed some similarities and differences between the methods. DCM is probably the most sophisticated of the three, but is computationally demanding. In its present implementation in **FIAR** it is a deterministic model, assuming all dynamics are captured without error by the design matrix. USEM and GC are computationally simpler and the parameters are easier to interpret. However, the models do not take into account the hemodynamics of the data, which may lead to false inference.

Another issue with USEM and GC is that a model order needs to be specified. This model order is a trade off between variance accounted for and the number of parameters to be estimated. **FIAR** allows to calculate the order that optimizes this trade off.

We implemented three types of GC in **FIAR**. GC differs from DCM and USEM in that the latter are confirmatory methods, testing an a priori causal network, while the former can be used in an exploratory way. All types of GC can be used in a univariate or multivariate way and with or without conditioning variables. As such, the univariate conditional GC with only two time series reduces to the well established bivariate GC.

All methods were applied to a real-world data set. The results illustrate that different methods for studying functional integration can lead to completely different conclusions. This is partly due to the different types of causality that are being measured. We believe that testing these different notions of causality and finding converging evidence between them should become standard practice when studying causal networks in the brain.

The brain is a very complex system that is neither linear, bilinear nor deterministic; neither bivariate, nor predictable. The abstractions and choices to be made in useful models of brain connectivity are therefore unlikely to be accommodated by one single "master" model (Roebroeck *et al.* 2011). Since "Essentially, all models are wrong, but some are useful" (Box and Draper 1987), we hope we have brought a selection of useful models to R with the package **FIAR**.

References

Aguirre G, Zarahn E, Désposito M (1998). "The Variability of Human, BOLD Hemodynamic Responses." *NeuroImage*, **8**, 360–369.

Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In BN Petrov, F Csaki (eds.), *2nd International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.

Ashburner J, Chen CC, Flandin G, Henson R, Kiebel S, Kilner J, Litvak V, Moran R, Penny W, Stephan K, Hutton C, Glauche V, Mattout J, Phillips C (2008). *The **SPM8** Manual.* Functional Imaging Laboratory, Wellcome Trust Centre for Neuroimaging, Institute of Neurology, UCL, London. URL http://www.fil.ion.ucl.ac.uk/spm/.

Barnett L, Barrett A, Seth A (2009). "Granger Causality and Transfer Entropy Are Equivalent for Gaussian Variable." *Physical Review Letters*, **103**, 238701.

Barrett A, Barnett L, Seth A (2010). "Multivariate Granger Causality and Generalized Variance." *Physical Review E*, **81**, 041907.

Bentler PM (1995). ***EQS** Program Manual.* Multivariate Software, Inc., Encino, CA.

Box GEP, Draper NR (1987). *Empirical Model-Building and Response Surfaces.* John Wiley & Sons, New York.

Bressler SL, Seth AK (2011). "Wiener-Granger Causality: A Well Established Methodology." *NeuroImage*, **58**(2), 323–329.

Buchel C, Friston KJ (1997). "Modulation of Connectivity in Visual Pathways by Attention: Cortical Interactions Evaluated with Structural Equation Modelling and fMRI." *Cerebral Cortex*, **7**, 768–778.

Buxton RB, Wong EC, Frank LR (1998). "Dynamics of Blood Flow and Oxygenation Changes During Brain Activity: The Balloon Model." *Magnetic Resonance in Medicine*, **39**, 855–864.

Canty A, Ripley BD (2011). ***boot**: Bootstrap R (S-PLUS) Functions.* R package version 1.3-2, URL http://CRAN.R-project.org/package=boot.

Daunizeau J, David O, Stephan KE (2011). "Dynamic Causal Modelling: A Critical Review of the Biophysical and Statistical Foundations." *NeuroImage*, **58**(2), 312–322.

Deshpande G, Sathian K, Hu X (2010). "Effect of Hemodynamic Variability on Granger Causality Analysis of fMRI." *NeuroImage*, **52**, 884–896.

Fox J (2010). ***sem**: Structural Equation Models.* R package version 0.9-21, URL http://CRAN.R-project.org/package=sem.

Friston K (2009). "Causal Modelling and Brain Connectivity in Functional Magnetic Resonance Imaging." *PLoS Biology*, **7**, 220–225.

Friston K, Harrison L, Penny W (2003). "Dynamic Causal Modelling." *NeuroImage*, **19**, 1273–1302.

Friston K, Michelli A, Turner R, Price C (2000). "Nonlinear Responses in fMRI: The Balloon Model, Volterra Kernels, and Other Hemodynamics." *NeuroImage*, **16**, 513–530.

Friston K, Penny W, David O (2005). "Modeling Brain Responses." *International Review of Neurobiology*, **66**, 89–124.

Geweke J (1982). "Measurement of Linear Dependence and Feedback between Multiple Time Series." *Journal of American Statistical Association*, **77**, 304–313.

Granger C (1969). "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods." *Econometrica*, **37**, 424–438.

Guo C, Seth A, Kendrick K, Zhou C, Feng J (2008). "Partial Granger Causality: Eliminating Exogenous Inputs and Latent Variables." *Journal of Neuroscience Methods*, **172**, 79–93.

Handwerker DA, Ollinger JM, D'Esposito M (2004). "Variation of BOLD Hemodynamic Responses across Subjects and Brain Regions and Their Effects on Statistical Analyses." *NeuroImage*, **21**, 1639–1651.

Harrison L, Penny WD, Friston K (2003). "Multivariate Autoregressive Modeling of fMRI Time Series." *NeuroImage*, **19**, 1477–1491.

Hayfield T, Racine JS (2008). "Nonparametric Econometrics: The **np** Package." *Journal of Statistical Software*, **27**(5), 1–32. URL http://www.jstatsoft.org/v27/i05/.

Honey GD, Fu CHY, Kim J, Brammer MJ (2002). "Effects of Verbal Working Memory Load on Corticocortical Connectivity Modeled by Path Analysis of Functional Magnetic Resonance Imaging Data." *NeuroImage*, **17**, 573–582.

Jöreskog KG, Sörbom D (2005). ***LISREL** 8.7 for Windows.* Lincolnwood, IL. URL http://www.ssicentral.com/lisrel/.

Kim J, Zhu W, Chang L, Bentler PM, Ernst T (2007). "Unified Structural Equation Modeling Approach for the Analysis of Multisubject, MUltivariate Functional MRI Data." *Human Brain Mapping*, **28**, 85–93.

Ladroue C, Guo S, Kendrick K Feng J (2009). "Beyond Element-Wise Interactions: Identifying Complex Interactions in Biological Processes." *PLoS ONE*, **4**. E6899.

McIntosh AR, Gonzales-Lima F (1994). "Structural Equation Modelling and Its Application to Network Analysis in Functional Brain Imaging." *Human Brain Mapping*, **2**, 2–22.

Muthèn LK, Muthèn BO (2004). ***Mplus** User's Guide.* Muthèn & Muthèn, Los Angeles, CA.

Pfaff B (2008). "VAR, SVAR and SVEC Models: Implementation within R Package **vars**." *Journal of Statistical Software*, **27**(4), 1–32. URL http://www.jstatsoft.org/v27/i04/.

Raftery AE (1995). "Bayesian Model Selection in Social Research." In PV Marsden (ed.), *Sociological Methodology*, pp. 111–196. Blackwell, Cambridge, MA.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Roebroeck A, Formisano E, Goebel R (2005). "Mapping Directed Influence over the Brain Using Granger Causality and fMRI." *NeuroImage*, **25**, 230–242.

Roebroeck A, Formisano E, Goebel R (2011). "The Identification of Interacting Networks in the Brain Using fMRI: Model Selection, Causality and Deconvolution." *NeuroImage*, **58**(2), 296–302.

Rosseel Y (2011). ***lavaan**: Latent Variable Analysis.* R package version 0.4-7, URL http://CRAN.R-project.org/package=lavaan.

Schwartz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**, 461–464.

Stephan KE, Weiskopf N, Drysdale PM, Robinson PA, Friston KJ (2007). "Comparing Hemodynamic Models with DCM." *NeuroImage*, **38**, 387–401.

The MathWorks, Inc (2010). *MATLAB – The Language of Technical Computing, Version 7.10.0.* The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Wiener N (1956). "The Theory of Prediction." In EF Beckenbach (ed.), *Modern Mathematics for Engineers*. McGraw-Hill, New York. Chapter 8.

Zeileis A, Hothorn T (2002). "Diagnostic Checking in Regression Relationships." *R News*, **2**(3), 7–10. URL http://CRAN.R-project.org/doc/Rnews/.

A. Example for DCM specification

```
DCM <- list()
DCM$n <- 3                            # number of regions
DCM$names <- c("V1", "V2", "V3")      # names of regions
DCM$m <- 2                            # number of inputs
DCM$ons <- list()                     # onsets of inputs
DCM$ons$input1 <- c(0, 60, 120, 180)
DCM$ons$input2 <- c(0, 30, 60, 90, 120, 150, 180, 210)
DCM$dur <- list()                     # duration of inputs
DCM$dur$input1 <- 30
DCM$dur$input2 <- 15
DCM$a <- c(0,   0,   0,               # anatomical connections
           0.7, 0,   0,               # from region 1 to region 2 and from
           0,   0.4, 0)               # region 2 to region 3
DCM$b <- c(0,   0,   0,               # functional connections
           0,   0,   0,               # induced by input 1 from region 2
           0,   0.2, 0,               # to region 3
           0,   0,   0,               # functional connections
           0,   0,   0.2,             # induced by input 2 from region 3
           0,   0,   0)               # to region 2
DCM$c <- c(0.4, 0,   0,               # input 1 on region 1
           0,   0.3, 0)               # input 2 on region 2
DCM$h <- c(0.65, 0.41, 0.98, 0.32, 0.34, 0) # hemodynamic parameters
DCM$x <- 5 * DCM$n                    # number of states per region
DCM$TR <- 1                           # TR
DCM$TE <- 0.04                        # TE
DCM$T <- 16                           # Timebins per TR
DCM$v <- 240                          # number of scans
```

B. Summary of AR(3) model fit

```
Lavaan (0.4-7) converged normally after 210 iterations

  Number of observations                          1997

  Estimator                                         ML
  Minimum Function Chi-square                17763.037
  Degrees of freedom                                28
  P-value                                        0.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                     Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  x_0 ~
    x_1                 1.361    0.022   60.883    0.000
    x_2                -0.920    0.032  -29.017    0.000
    x_3                 0.011    0.022    0.504    0.614
  y_0 ~
    x_0                 0.971    0.005  188.230    0.000
    x_1                -1.335    0.023  -57.074    0.000
    y_1                 0.039    0.022    1.724    0.085
    x_2                 1.417    0.030   46.617    0.000
    y_2                 0.001    0.004    0.295    0.768
    x_3                -0.047    0.031   -1.496    0.135
    y_3                 0.001    0.003    0.346    0.729
  z_0 ~
    y_0                 0.882    0.006  144.478    0.000
    y_1                -0.912    0.014  -62.949    0.000
    z_1                 0.874    0.017   52.136    0.000
    y_2                 0.350    0.020   17.549    0.000
    z_2                -0.781    0.020  -39.475    0.000
    y_3                -0.109    0.012   -9.113    0.000
    z_3                -0.392    0.016  -25.018    0.000
  x_1 ~
    x_2                 1.351    0.010  141.312    0.000
    x_3                -0.904    0.010  -94.532    0.000
  y_1 ~
    x_1                 0.971    0.005  188.276    0.000
    x_2                -1.302    0.007 -177.979    0.000
    y_2                 0.006    0.004    1.484    0.138
    x_3                 1.372    0.005  254.446    0.000
    y_3                 0.001    0.003    0.463    0.644
  z_1 ~
    y_1                 0.758    0.008   91.659    0.000
    y_2                -1.107    0.010 -113.066    0.000
    z_2                 1.028    0.013   79.505    0.000
    y_3                 0.503    0.011   44.078    0.000
    z_3                -0.838    0.009  -90.661    0.000
  x_2 ~
    x_3                 0.710    0.016   44.986    0.000
  y_2 ~
    x_2                -0.098    0.012   -7.942    0.000
    x_3                 0.388    0.012   31.100    0.000
    y_3                 0.353    0.015   23.276    0.000
  z_2 ~
    y_2                 0.187    0.014   13.465    0.000
    y_3               -0.630    0.014  -45.904    0.000
    z_3                0.334    0.014   23.613    0.000
  y_3 ~
    x_3                0.120    0.013    9.363    0.000
  z_3 ~
    y_3               -0.153    0.019   -8.123    0.000

Variances:
    x_0               25.274    0.800   31.599    0.000
    y_0                1.342    0.042   31.599    0.000
    z_0                5.110    0.162   31.599    0.000
    x_1               25.310    0.801   31.599    0.000
    y_1                1.343    0.043   31.599    0.000
    z_1                9.107    0.288   31.599    0.000
    x_2              138.624    4.387   31.599    0.000
    y_2               42.124    1.333   31.599    0.000
    z_2               27.253    0.862   31.599    0.000
    y_3               91.918    2.909   31.599    0.000
    z_3               68.297    2.161   31.599    0.000
```

Affiliation:

Bjorn Roelstraete
Department of Data Analysis
Ghent University
9000, Ghent, Belgium
E-mail: Bjorn.Roelstraete@ugent.be

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 44, Issue 13 *Submitted:* 2010-11-13
October 2011 *Accepted:* 2011-06-21