Estimating time-varying brain connectivity networks from functional MRI time series

CrossMark

Ricardo Pio Monti [a], Peter Hellyer [b], David Sharp [b], Robert Leech [b],
Christoforos Anagnostopoulos [a], Giovanni Montana [a,c,]*

[a] *Department of Mathematics, Imperial College London, London SW7 2AZ, UK*
[b] *Computational, Cognitive and Clinical Neuroimaging Laboratory, Imperial College London, The Hammersmith Hospital, London W12 0NN, UK*
[c] *Department of Biomedical Engineering, King's College London, St Thomas' Hospital, London SE1 7EH, UK*

A B S T R A C T

At the forefront of neuroimaging is the understanding of the functional architecture of the human brain. In most applications functional networks are assumed to be stationary, resulting in a single network estimated for the entire time course. However recent results suggest that the connectivity between brain regions is highly non-stationary even at rest. As a result, there is a need for new brain imaging methodologies that comprehensively account for the dynamic nature of functional networks. In this work we propose the Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm which estimates dynamic brain networks from fMRI data. We apply the proposed algorithm to functional MRI data from 24 healthy patients performing a Choice Reaction Task to demonstrate the dynamic changes in network structure that accompany a simple but attentionally demanding cognitive task. Using graph theoretic measures we show that the properties of the Right Inferior Frontal Gyrus and the Right Inferior Parietal lobe dynamically change with the task. These regions are frequently reported as playing an important role in cognitive control. Our results suggest that both these regions play a key role in the attention and executive function during cognitively demanding tasks and may be fundamental in regulating the balance between other brain regions.

© 2014 Published by Elsevier Inc.

Introduction

The discovery of non-invasive brain imaging techniques has greatly boosted interest in cognitive neuroscience. Specifically, the discovery of functional Magnetic Resonance Imaging (fMRI) has instigated a revolution by providing a non-invasive and readily accessible method by which to obtain high quality images of the human brain. While traditional fMRI studies focused exclusively on reporting the behaviour of individual brain regions independently, there has been a recent shift towards understanding the relationships between distinct brain regions, referred to as brain connectivity (Lindquist, 2008). The study of brain connectivity has resulted in fundamental insights such as small-world architecture (Sporns et al., 2004; Bassett and Bullmore, 2006) and the presence of hubs (Eguiluz et al., 2005).

A cornerstone in the understanding brain connectivity is the notion that connectivity can be represented as a graph or network composed of a set of nodes interconnected by a set of edges. This allows for connectivity to be studied using a rich set of graph theoretic tools (Newman, 2003; Fornito et al., 2013) and has resulted in widespread use of

graph theoretic techniques in neuroscience (Fair et al., 2009; Achard et al., 2006). The first step when looking to study brain connectivity is to define a set of nodes. This can be achieved in many ways; in the case of fMRI nodes are often defined as spatial regions of interest (ROIs). Alternatively, Independent Component Analysis (ICA) can be employed to determine independent components which are subsequently used as nodes (Calhoun et al., 2009). It follows that each node is associated with its own time course of imaging data. This is subsequently used to estimate the connections between nodes, defined as the edge structure of the network. In particular, functional connectivity estimates of the edge structure can be obtained by studying the statistical dependencies between each of the nodes (Strother et al., 1995; Lowe et al., 1998; van der Heuvel and Hulshoff Pol, 2010; Friston, 2011). The resulting networks, referred to as functional connectivity networks, are the primary focus of this work.

Traditionally functional connectivity networks have been estimated by measuring pair-wise linear dependencies between nodes, quantified by Pearson's correlation coefficient (Hutchinson et al., 2013; Fornito et al., 2013). This corresponds to estimating the covariance matrix where each entry corresponds to the correlation between a distinct pair of nodes. Partial correlations, summarised in the precision or inverse covariance matrix (Whittaker, 1990), have also been employed extensively (Huang et al., 2010; Liu et al., 2009; Marrelec et al., 2006;

* Corresponding author at: Department of Mathematics, Imperial College London, London SW7 2AZ, UK.

E-mail address: giovanni.montana@kcl.ac.uk (G. Montana).

Sun et al., 2004; Pandit et al., 2013; Hinne et al., 2013). In this case, the correlations between nodes are inferred once the effects of all other units have been removed. Partial correlations are typically preferred to Pearson's correlation coefficient as they have been shown to be better suited to detecting changes in connectivity structure (Smith et al., 2011; Marrelec et al., 2009).

Intrinsically linked to the problem of estimating the functional connectivity structure is the issue of estimating the true sparsity of the networks in question. There are numerous studies reporting brain networks to be of varying levels of sparsity. For example, Bullmore and Sporns (2009) suggest that connectivity networks have evolved to achieve high efficiency of information transfer at a low connection cost, resulting in sparse networks. On the other hand, Markov et al. (2013) propose a high-density model where efficiency is achieved via the presence of highly heterogeneous edge strengths between nodes. Here we pose the level of sparsity as a statistical question to be answered by the data. Due to the presence of noise, it follows that every entry in the estimated precision or covariance matrices will be non-zero. This results in dense, unparsimonious networks which are potentially dominated by noise. The two most common approaches to addressing this problem involve the use of multiple hypothesis tests or regularisation. The former involves testing each edge for statistical significance (Nichols and Hayasaka, 2003) while the latter involves the formulation of an objective function which contains an additional regularisation penalty to encourage sparsity. A popular example of such a penalty is the Graphical Lasso penalty (Friedman et al., 2008). This penalises the sum of the off-diagonal elements in the precision matrix thus balancing a trade-off between sparsity and goodness-of-fit. Furthermore, in many neuroimaging studies it is often the case that the number of parameters to estimate exceeds the number of observations. In such scenarios the use of regularisation is required for the formulation of a well-posed problem. Moreover, regularisation in the form of the Graphical Lasso penalty encourages only the presence of edges which are best supported by the data.

The aforementioned methods are based on the underlying assumption that functional connectivity networks are not changing over time. However, there is growing evidence that fMRI data is non-stationary (Hutchinson et al., 2012; Hellyer et al., 2014); this is especially true in task-based fMRI studies (Chang and Glover, 2010). As a result there is a clear need to quantify dynamic changes in network structure over time. Specifically, there is a need to estimate a network at each observation in order to accurately quantify temporal diversity. To date the most commonly used approach to achieve this goal involves the use of sliding windows (Hutchinson et al., 2013). Here observations lying within a time window of fixed length are used to calculate the functional connectivity. This window is then shifted, allowing for the estimation of dynamic functional networks. Examples include Handwerker et al. (2012) who use sliding windows to quantify dynamic trends in correlation as well as Allen et al. (2012) who propose a temporal clustering of covariance matrices and study the Fourier representation of edges. It follows that sliding windows can also be combined with approaches such as ICA (Esposito et al., 2003) and principal component analysis (Leonardi et al., 2013).

While sliding windows are a valuable tool for investigating high-level dynamics of functional connectivity networks there are two main issues associated with its use. First, the choice of window length can be a difficult parameter to tune. It is advised to set the window length to be large enough to allow for robust estimation of network statistics without making it too large, which would result in overlooking interesting short-term fluctuations (Sakoglu et al., 2010). Second, the use of sliding windows needs to be accompanied by an additional mechanism to determine if variations in edge structure are significant. This would result in estimated networks where the edge structure is only reported to change when substantiated by evidence in the data. We refer to this quality as temporal homogeneity. One way this can be achieved is via the use of hypothesis tests, as in the recently proposed Dynamic Connectivity Regression (DCR) algorithm (Cribben et al., 2012).

In this work we are concerned with multivariate statistical methods for inferring dynamic functional connectivity networks from fMRI data. We are particularly interested in two aspects. First, we wish to obtain individual estimates of brain connectivity at each time point as opposed to a network for the entire time series. This will allow us to fully characterise the dynamic evolution of networks over time and provide valuable insight into brain organisation and cognition. Second, we wish to encourage our estimation procedure to produce estimates with the two properties discussed previously; sparsity and temporal homogeneity

In order to achieve these goals we propose a new methodology, the Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm. SINGLE can be seen as an extension of sliding windows where the two issues mentioned previously — sparsity and temporal homogeneity — are addressed. First, we propose an objective method for estimating the window length based on cross-validation. We then introduce the SINGLE algorithm which is capable of accurately estimating dynamic networks. The proposed algorithm is able to estimate dynamic networks by minimising a penalised loss function. This function contains a likelihood term for each observation together with two penalty terms. Sparsity is achieved via the introduction of a Graphical Lasso penalty while temporal homogeneity is achieved by introducing a penalty inspired by the Fused Lasso (Tibshirani et al., 2005) which effectively penalises the difference between consecutive networks.

Throughout a series of simulations, we study the ability of the SINGLE algorithm to accurately estimate dynamic random networks resembling fMRI data and benchmark its performance against sliding window based algorithms and the DCR algorithm. We then apply the SINGLE algorithm to data collected from 24 healthy subjects while performing a Choice Reaction Time (CRT) task. During the CRT task, subjects were required to make a rapid visually-cued motor decision. Stimulus presentation was blocked into five on-task periods, each preceding a period where subjects were at rest. As a result, we expect there to be an alternating network structure depending on the task. This makes the data set particularly suitable for demonstrating the limitations involved with the assumption of stationarity as well as the capabilities of the SINGLE algorithm.

The remainder of this paper is structured as follows: in Methods section we introduce and describe the SINGLE framework and optimisation algorithm in detail. In Experimental results section we present the results of our simulation study and in Application to a choice reaction time (CRT) task fMRI dataset section we then apply the proposed algorithm to fMRI data collected for 24 healthy subjects while performing a Choice Reaction Time (CRT) task.

Methods

We assume we have observed fMRI time series data denoted by $X_1, ..., X_T$, where each vector $X_i \in \mathbb{R}^{1 \times p}$ contains the BOLD measurements of p nodes at the ith time point. Throughout the remainder we assume that each X_i follows a multivariate Gaussian distribution, $X_i \sim \mathcal{N}(\mu_i, \Sigma_i)$. Here the mean and covariance are dependent on time index in order to accommodate the non-stationary nature of fMRI data.

We aim to infer functional connectivity networks over time by estimating the corresponding precision (inverse covariance) matrices $\{\Theta_i\} = \{\Theta_1, ..., \Theta_T\}$. Here, Θ_i encodes the partial correlation structure at the ith observation (Whittaker, 1990). It follows that we can encode Θ_i as a graph or network G_i where the presence of an edge implies a non-zero entry in the corresponding precision matrix and can be interpreted as a functional relationship between the two nodes in question. Thus our objective is equivalent to estimating a sequence of time indexed graphs $\{G_i\} = \{G_1, ..., G_T\}$ where each G_i summarises the functional connectivity structure at the ith observation.

We wish for our estimated graphs $\{G_i\}$ to display the following two properties:

(a) **Sparsity**: The introduction of sparsity is motivation by two reasons; first, the number of parameters to estimate often exceeds the number of observations. In this case the introduction of regularisation is required in order to formulate a well-posed problem. Second, due to the presence of noise, all entries in the estimated precision matrices will be non-zero. This results in dense, unparsimonious networks that are potentially dominated by noise.

(b) **Temporal homogeneity**: From a biological perspective, it has been reported that functional connectivity networks exhibit changes due to task based demands (Esposito et al., 2006; Fornito et al., 2012; Fransson, 2006; Sun et al., 2007). As a result, we expect the network structure to remain constant within a neighbourhood of any observation but to vary over a larger time horizon. This is particularly true for task-based fMRI studies where stimulus presentation is blocked. In light of this, we wish to encourage estimated graphs with sparse innovations over time. This ensures that a change in the connectivity between two nodes is only reported when is it substantiated by evidence in the data.

We split the problem of estimating $\{\Theta_i\}$ into two independent tasks. First we look to obtain local estimates of sample covariance matrices $S_1, ..., S_T$. This is achieved via the use of kernel functions and discussed in detail in the Estimation of time-varying covariance matrices section below. Assuming such a sequence exists we wish to estimate the corresponding precision matrices $\{\Theta_i\}$ with the aforementioned properties while ensuring that each Θ_i adequately describes the corresponding S_i. The latter is quantified by considering the goodness-of-fit:

$$f(\{\Theta_i\}) = \sum_{i=1}^{T} - \log \ det \ \Theta_i + \text{trace} \ (S_i\Theta_i), \tag{1}$$

which is proportional to the negative log-likelihood. While it would be possible to estimate $\{\Theta_i\}$ by directly minimising f, this would not guarantee either of the properties discussed previously. In order to enforce sparsity and temporal homogeneity we introduce the following regularisation penalty:

$$g_{\lambda_1,\lambda_2}(\{\Theta_i\}) = \lambda_1 \sum_{i=1}^{T} \|\Theta_i\|_1 + \lambda_2 \sum_{i=2}^{T} \|\Theta_i - \Theta_{i-1}\|_1. \tag{2}$$

Sparsity is enforced by the first penalty term which assigns a large cost to matrices with large absolute values, thus effectively shrinking elements towards zero. This can be seen as a convex approximation to the combinatorial problem of selecting the number of edges. The second penalty term, regularised by λ_2, encourages temporal homogeneity by penalising the difference between consecutive networks. This can be seen as an extension of the Fused Lasso penalty (Tibshirani et al., 2005) from the context of linear regression (i.e., enforcing similarities across regression coefficients) to penalising the changes in network structure over time.

The proposed method minimises the following loss function:

$$l(\{\Theta_i\}) = f(\{\Theta_i\}) + g_{\lambda_1,\lambda_2}(\{\Theta_i\}). \tag{3}$$

This allows for the estimation of time-index precision matrices which display the properties of sparsity and temporal homogeneity while providing an accurate representation of the data — a schematic representation of the proposed algorithm is provided in Fig. 1. The choice of regularisation parameters λ_1 and λ_2 allow us to balance this trade-off and these are learned from the data as described in Parameter tuning section.

The remainder of this section is organised as follows: in Estimation of time-varying covariance matrices section we describe the estimation of time-varying sample covariance matrices $S_1, ..., S_T$ using kernel functions. In Optimisation algorithm section we outline the optimisation algorithm used to minimise Eq. (3) as well as discuss the computational complexity of the resulting algorithm. Finally in Parameter tuning section we describe how the related parameters can be learnt from the data and in Experimental data section we describe the experimental data used in our application.

Estimation of time-varying covariance matrices

The loss function (3) requires the input of estimated sample covariance matrices $S_1, ..., S_T$. This is itself a non-trivial and widely studied problem. Under the assumption of stationarity, the covariance matrix can be directly calculated as $S = \frac{1}{T-1}\sum_{i=1}^{T}(X_i-\overline{x})'(X_i-\overline{x})$ where $\overline{x}$ is the sample mean and we write A' to denote the transpose of a matrix A. However, when faced with non-stationary data such an approach is unsatisfactory and there is a need to obtain local estimates of the covariance matrix.

A potential approach involves the use of change-point detection to segment the data into piece-wise stationary segments, as is the case in the DCR algorithm (Cribben et al., 2012). Alternatively, a sliding window may be used to obtain a locally stationary estimate of the sample covariance at each observation. Due to the sequential nature of the observations, sliding windows allow us to obtain adaptive estimates by considering only temporally adjacent observations.

A natural extension of sliding windows is to obtain adaptive estimates by down-weighting the contribution of past observations. This can be achieved using kernel functions.

Formally, kernel functions have the form $K_h(i, j)$ where $K_h(\cdot, \cdot)$ is a symmetric, non-negative function, h is a specified fixed width and i and j are time indices. By considering the uniform kernel, $K_h(i, j) = \mathbb{I}\{|i-j|<h\}$ [1], we can see that sliding windows are a special case of kernel functions. This allows us to contrast the behaviour of sliding windows against alternative kernels, such as the Gaussian kernel:

$$K_h(i, j) = \exp\left\{\frac{-(i-j)^2}{h}\right\}. \tag{4}$$

Fig. 2 provides clear insight into the different behaviour of each of the two kernels. While sliding windows have a sharp cutoff, the Gaussian kernel gradually reduces the importance given to observations according to their chronological proximity. In this manner, the Gaussian kernel is able to give greater importance to temporally adjacent observations. In addition to this, sliding windows inherently assume that observations arrive at equally spaced intervals while the use of more general kernel functions, such as the Gaussian kernel, naturally accommodates cases where this assumption does not hold.

Finally, given a kernel function, adaptive estimates of the ith sample mean and covariance can be directly calculated as follows:

$$\overline{x}_i = \frac{\sum_{j=1}^{T} K_h(i, j)X_j}{\sum_{j=1}^{T} K_h(i, j)}, \tag{5}$$

$$S_i = \frac{\sum_{j=1}^{T} K_h(i, j)(X_j-\overline{x}_j)'(X_j-\overline{x}_j)}{\sum_{j=1}^{T} K_h(i, j)}. \tag{6}$$

It follows that for both the Gaussian kernel as well as the sliding window the choice of h plays a fundamental role. It is typically advised to set h to be large enough to ensure robust estimation of covariance matrices without making h too large (Sakoglu et al., 2010). However, data-driven approaches are rarely proposed (Hutchinson et al., 2013).

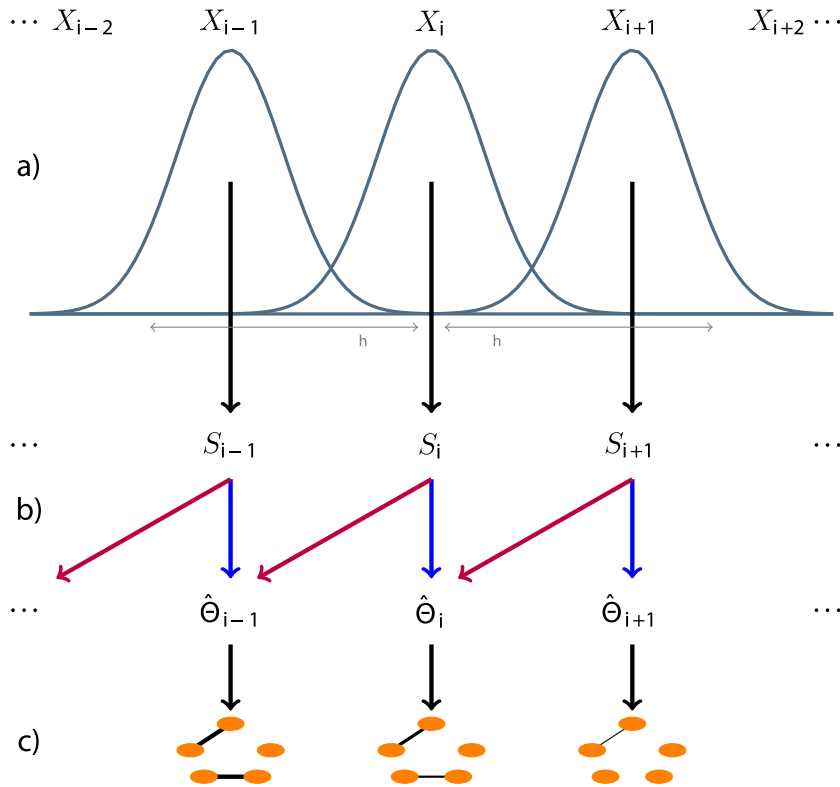[1] here $\mathbb{I}(x)$ is the indicator function.

Fig. 1. A graphical representation of the SINGLE algorithm illustrating its various components. a) Gaussian kernels are used to obtain estimate local covariance matrices at each observation. b) These are then used to obtain smooth estimates of precision matrices by combining the Graphical Lasso (blue) and Fused Lasso (purple) penalties. c) Finally the estimated precision matrices can be represented as graphs.

This is partly because the choice of h will depend on many factors, such as the rate of change of the underlying networks, which are rarely known a priori. Here we propose to estimate h using cross-validation. This is discussed in detail in the Parameter tuning section.

Optimisation algorithm

Having obtained estimated sample covariance matrices, we turn to the problem of minimising the loss function (3). While this loss is convex (see Appendix A) it is not continuously differentiable due to the presence of the penalty terms. In particular, the presence of the Fused Lasso penalty poses a real restriction. Additional difficulty is introduced by the structured nature of the problem: we require that each Θ_i be symmetric and positive definite.

The approach taken here is to exploit the separable nature of Eq. (3). As discussed previously, the loss function is composed of two components; the first of which is proportional to the sum of likelihood terms
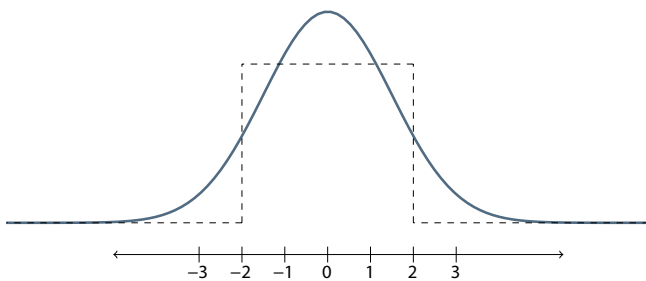

Fig. 2. Example demonstrating the difference between a Gaussian kernel and a sliding window. We note that the sliding window gives zero weighting to observations at ± 3 while this is not the case for the Gaussian kernel. We also note that the Gaussian kernel gives greater importance to chronologically adjacent observations while the sliding window gives an equal weighting to all observations within its width.

and the second containing the sum of the penalty components. This separability allows us to take advantage of the structure of each component.

There has been a rapid increase in interest in the study of such separable loss functions in the statistics, machine learning and optimisation literature. As a result, there are numerous algorithms which can be employed such as Forward-Backward Splitting (Duchi and Singer, 2009) and Regularised Dual Averaging (Xiao, 2010). Here we capitalise on the separability of our problem by implementing an Alternating Directions Method of Multipliers (ADMM) algorithm (Boyd et al., 2010). The ADMM is a form of augmented Lagrangian algorithm[2] which is particularly well suited to dealing with highly structured nature of the problem proposed here. Moreover, the use of an ADMM algorithm is able to guarantee estimated precision matrices, $\{\Theta_i\}$, that are symmetric and positive definite as we outline below.

In order to take advantage of the separability of the loss function (3) we introduce a set of auxiliary variables denoted $\{Z_i\} = \{Z_1, ..., Z_T\}$ where each $Z_i \in \mathbb{R}^{p \times p}$ corresponds to each Θ_i. This allows us to minimise the loss with respect to each set of variables, $\{\Theta_i\}$ and $\{Z_i\}$ in iterative fashion while enforcing an equality constraint on each Θ_i and Z_i respectively. Consequently, Eq. (3) can be reformulated as the following constrained minimisation problem:

$$\underset{\{\Theta_i\},\{Z_i\}}{\text{minimise}} \sum_{i=1}^{T}(-\log \det \Theta_i + \text{trace } (S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{T} \|Z_i\|_1 + \lambda_2 \sum_{i=2}^{T} \|Z_i - Z_{i-1}\|_1 \tag{7}$$

$$\text{subject to } \Theta_i = Z_i i = 1, ..., T \tag{8}$$

---

[2] See Bertsekas (1999, chap. 4) for a concise description of augmented Lagrangian methods.

where we have replaced Θ_i with Z_i in both of the penalty terms. As a result, $\{\Theta_i\}$ terms are involved only in the likelihood component of Eq. (7) while $\{Z_i\}$ terms are involved in the penalty components. This decoupling allows for the individual structure associated with the $\{\Theta_i\}$ and $\{Z_i\}$ to be leveraged.

The use of an ADMM algorithm requires the formulation of the augmented Lagrangian corresponding to Eqs. (7) and (8). This is defined as:

$$\mathcal{L}_\gamma(\{\Theta_i\}, \{Z_i\}, \{U_i\}) = -\sum_{i=1}^{T} (\log \det \Theta_i - \text{trace } (S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{T} \|Z_i\|_1$$
$$+ \lambda_2 \sum_{i=2}^{T} \|Z_i - Z_{i-1}\|_1 + \gamma 2 \sum_{i=1}^{T} \left(\|\Theta_i - Z_i + U_i\|_2^2 - \|U_i\|_2^2 \right),$$
(9)

where $\{U_i\} = \{U_1, ..., U_T\}$ are scaled Lagrange multipliers such that $U_i \in \mathbb{R}^{p \times p}$. Eq. (9) corresponds to the Lagrangian for Eqs. (7) and (8) together with an additional quadratic penalty term (see Appendix B for details). The latter is multiplied by a constant stepsize parameter γ which can typically be set to one. The introduction of this term is desirable as it often facilitates the minimisation of the Lagrangian; specifically in our case it will make our problem substantially easier as we outline below.

The proposed estimation procedure works by iteratively minimising Eq. (9) with respect to each set of variables: $\{\Theta_i\}$ and $\{Z_i\}$. This allows us to decouple the Lagrangian in such a manner that the individual structure associated with variables $\{\Theta_i\}$ and $\{Z_i\}$ can be exploited.

We write $\{\Theta_i^j\} = \{\Theta_1^j, ..., \Theta_T^j\}$ where Θ_i^j denotes the estimate of Θ_i in the jth iteration. The same notation is used for $\{Z_i\}$ and $\{U_i\}$. The algorithm is initialised with $\Theta_i^0 = I_p, Z_i^0 = U_i^0 = 0 \in \mathbb{R}^{p \times p}$ for $i = 1, ..., T$. At the jth iteration of the proposed algorithm three steps are performed as follows:

Step 1: Update $\{\Theta_i^j\}$

At the jth iteration, each Θ_i is updated independently by minimising Eq. (9). At this step we treat all $\{Z_i^j\}$, $\{U_i^j\}$ and Θ_k^j, for $k \neq i$ as constants. As a result, minimising Eq. (9) with respect to Θ_i corresponds to setting:

$$\Theta_i^j = \underset{\Theta_i}{\arg\min} \left\{ -\log \det \Theta_i + \text{trace}(S_i\Theta_i) + \gamma/2 \left\| \Theta_i - Z_i^{j-1} + U_i^{j-1} \right\|_2^2 \right\}.$$
(10)

From Eq. (10) we can further understand the process occurring at this step. If γ is set to zero only the negative log-likelihood terms will be left in Eq. (10) resulting in $\Theta_i^j = S_i^{-1}$, the maximum likelihood estimator. However, this will not enforce either sparsity or temporal homogeneity and requires the assumption that S_i is invertible. Setting γ to be a positive constant implies that Θ_i will be a compromise between minimising the negative log-likelihood and remaining in the proximity of Z_i^{j-1}. The extent to which the latter is enforced will be determined by both γ and Lagrange multiplier U_i^{j-1}. As we will see in step 2, it is the $\{Z_i\}$ terms which encode the sparsity and temporal homogeneity constraints.

Differentiating the right hand side of Eq. (10) with respect to Θ_i and setting the derivative equal to zero yields:

$$\Theta_i^{-1} - \gamma\Theta_i = S_i - \gamma \left(Z_i^{j-1} - U_i^{j-1} \right)$$
(11)

which is a matrix quadratic in Θ_i (after multiplying through by Θ_i). In order to solve this quadratic, we observe that both Θ_i and $S_i - \gamma(Z_i^{j-1} - U_i^{j-1})$ share the same eigenvectors (see Appendix C). This allows us to solve Eq. (10) using an

eigendecomposition as outlined below. Now letting θ_r and s_r denote the rth eigenvalues of Θ_i and $S_i - \gamma(Z_i^{j-1} - U_i^{j-1})$ respectively we have that:

$$\theta_r^{-1} - \gamma\theta_r = s_r.$$
(12)

Solving the quadratic in Eq. (12) yields

$$\theta_r = \frac{1}{2\gamma} \left(-s_r + \sqrt{s_r^2 + 4\gamma} \right),$$
(13)

for $r = 1, ..., p$. Due to the nature of Eq. (13) it follows that all eigenvalues, θ_i will be great than zero. Thus Step 1 involves an eigendecomposition and update

$$\Theta_i = V_i \widetilde{D}_i V_i'$$
(14)

for each $i = 1, ..., T$. Here V_i is a matrix containing the eigenvectors of $S_i - \gamma(Z_i^{j-1} - U_i^{j-1})$ and $\widetilde{D}_i$ is a diagonal matrix containing entries $\theta_1, ..., \theta_p$. As discussed, all of the entries in $\widetilde{D}_i$ will be strictly positive, ensuring that each Θ_i will be positive definite. Moreover, we also note from Eq. (14) that each Θ_i will also be symmetric.

Step 2: Update $\{Z_i^j\}$

As in step 1, all variables $\{\Theta_i^j\}$ and $\{U_i^j\}$ are treated as constants when updating $\{Z_i\}$ variables. Due to the presence of the Fused Lasso penalty in Eq. (9) we cannot update each Z_i^j separately as was the case with each Θ_i^j in step 1. Instead, at the jth iteration the $\{Z_i^j\}$ variables are updated by solving:

$$\{Z_i^j\} = \underset{\{Z_i\}}{\arg\min} \left\{ \gamma/2 \sum_{i=1}^{T} \left\| \Theta_i^j - Z_i + U_i^{j-1} \right\|_2^2 + \lambda_1 \sum_{i=1}^{T} \|Z_i\|_1 \right. \quad (15)$$
$$\left. + \lambda_2 \sum_{i=2}^{T} \|Z_i - Z_{i-1}\|_1 \right\},$$

where we note that only element-wise operations are applied. As a result it is possible to break down Eq. (15) into element-wise optimisations of the following form:

$$\underset{\{Z_i\}_{k,l}}{\arg\min} \left\{ \gamma/2 \sum_{i=1}^{T} \left\| \left(\Theta_i^j - Z_i + U_i^{j-1} \right)_{k,l} \right\|_2^2 + \lambda_1 \sum_{i=1}^{T} \left\| (Z_i)_{k,l} \right\|_1 \right. \quad (16)$$
$$\left. + \lambda_2 \sum_{i=2}^{T} \left\| (Z_i - Z_{i-1})_{k,l} \right\|_1 \right\}$$

where we write $(M)_{k,l}$ to denote the (k, l) entry for any square matrix M. This corresponds to a *Fused Lasso signal approximator* problem (Hoefling, 2010) (see Appendix D). Moreover, due to the symmetric nature of matrices $\{\Theta_i\}$, $\{Z_i\}$ and $\{U_i\}$ we require $\frac{p(p+1)}{2}$ optimisations of the form shown in Eq. (16). Thus by introducing auxiliary variables $\{Z_i\}$ and formulating the augmented Lagrangian we are able to enforce both sparsity and temporal homogeneity penalties by solving a series of one-dimensional Fused Lasso optimisations.

Step 3: Update $\{U_i^j\}$

Step 3 corresponds to an update of Lagrange multipliers $\{U_i^j\}$ as follows:

$$U_i^j = U_i^{j-1} + \Theta_i^j - Z_i^j \text{ for } i = 1, ..., T$$
(17)

Convergence criteria

The proposed algorithm is an iterative procedure consisting of Steps 1–3 described above until convergence is reached. In order to guarantee convergence we require both primal and dual feasibility: primal feasibility refers to satisfying the constraint $\Theta_i = Z_i$ while dual

feasibility refers to minimisation of the Augmented Lagrangian. That is we require both that $\nabla_\Theta \mathcal{L}(\Theta, Z^j, U^j) = 0$ and $\nabla_Z \mathcal{L}(\Theta^{j+1}, Z, U^j) = 0$. We can check for primal feasibility by considering $\|\Theta_i^j - Z_i^j\|_2^2$ at each iteration. As detailed in Appendix E, step 3 ensures that $\{Z_i\}$ are always dual feasible and it suffices to consider $\|Z^j - Z^{j-1}\|_2^2$ to verify dual feasibility in $\{\Theta_i\}$ variables. Thus the SINGLE algorithm is said to converge when $\|\Theta_i^j - Z_i^j\|_2^2 < \epsilon_1$ and $\|Z_i^j - Z_i^{j-1}\|_2^2 < \epsilon_2$ for $i = 1, ..., T$ where ϵ_1 and ϵ_2 are user specified convergence thresholds. The complete procedure is given in Algorithm 1.

Computational complexity

As discussed previously the optimisation of the SINGLE objective function involves the iteration of three steps. In step 1 we perform n eigendecompositions, each of complexity $\mathcal{O}(p^3)$ where p is the number of nodes (i.e., the dimensionality of the data). Thus step 1 has a computational complexity of $\mathcal{O}(np^3)$. We note that step 2 requires $\frac{p(p+1)}{2}$ iterations of the Fused Lasso[3] where each iteration is $\mathcal{O}(n \log(n))$ (Hoefling, 2010). Thus the computational complexity of step 2 is $\mathcal{O}(p^2 n \log(n))$. Finally step 3 only involves matrix addition implying that the final computational complexity of the SINGLE algorithm is $\mathcal{O}(p^2 n \log(n) + np^3)$. This is dominated by the number of nodes, p, not the number of observations. As a result the limiting factor is likely to be the number of nodes in a study.

Parameter tuning

The SINGLE algorithm requires the input of three parameters which can be tuned using the available data: λ_1, λ_2 and h. Each of these parameters has a direct interpretation. Parameter h is the width of the Gaussian kernel. Following from our discussions in the Introduction section, similar considerations should be made when tuning h as when tuning the width of a sliding window. Parameters λ_1 and λ_2 affect the sparsity and temporal homogeneity respectively. In particular, increasing λ_1 will result in network estimates with a higher degree of sparsity whereas increasing the value of λ_2 will encourage the fusion of temporally adjacent estimates. We discuss each of these three parameters in turn.

The choice of parameter h describes certain assumptions relating to the nature of the available data which are often not formally discussed. The use of a kernel (be it in the form of a sliding window or otherwise) also reflects an assumption of local, as opposed to global, stationarity. This assumption is that it is possible to obtain time dependent parameter estimates that accurately reflect the correlation structure within a neighbourhood of any observation but possibly not over an arbitrarily long time horizon. The choice of h can therefore be seen as an assumption relating to the extent of non-stationarity of the available data (for an attempted definition of the degree of non-stationarity see Haykin (2002, chap. 16)).

On the one hand, the choice of a large value of h is indicative of an assumption that the data is close to stationary. If this is the case, a large choice of h allows for the accurate estimation of sample covariance matrices by incorporating information across a wide range of observations. However, if this assumption is incorrect, the choice of a large h can result in overly smoothed estimates where short term variation is overlooked. On the other hand, the choice of a small h implies an assumption of a higher degree of non-stationarity. Here the choice of a small h can allow for the accurate estimation of sample covariance matrices by correctly discarding irrelevant information. However reducing the value of h will result in an increase in the variance of the estimators as it implies that a smaller sample size is used to estimate parameters. This effect is more dramatic for large values of p as a greater number of parameters must be estimated. Overall, the best performing value of h in any given setting will depend on the difficulty of the estimation task, in particular the dimensionality of p, as well as the rate of change

of the underlying networks. The latter is not known a priori in many fMRI applications.

Algorithm 1. Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm.

To avoid making specific assumptions about the nature of the tem-

Input: Multivariate fMRI time series $X_1, ..., X_T$, where each $X_i \in \mathbb{R}^{1 \times p}$, Gaussian kernel size h, penalty parameters λ_1, λ_2, convergence tolerance ϵ_1, ϵ_2, max number of iterations M;
Result: Sparse estimates of precision matrices $\Theta_1, ..., \Theta_T$.
Set $\Theta_i^0 = I_p$, $Z_i^0 = U_i^0 = \mathbf{0}$ for $i \in \{1, ..., T\}$ and $j = 1$;
for i in $\{1, ..., T\}$ **do**
$\quad \mu_i = \frac{\sum_{j=1}^T K_h(i,j) \cdot X_j}{\sum_{j=1}^T K_h(i,j)}$;
end
for i in $\{1, ..., T\}$ **do**
$\quad S_i = \frac{\sum_{j=1}^T K_h(i,j) \cdot (X_j - \mu_j)'(X_j - \mu_j)}{\sum_{j=1}^T K_h(i,j)}$;
end
while Convergence == False and $j < M$ **do**
$\quad$ ## $\{\Theta\}$ Update;
$\quad$ **for** i in $\{1, ..., T\}$ **do**
$\qquad V, D = \text{eigen}\left(S_i - \gamma\left(Z_i^{j-1} - U_i^{j-1}\right)\right)$;
$\qquad \tilde{D} = \text{diag}\left(\frac{1}{2\gamma}\left(-D + \sqrt{D^2 + 4\gamma}\right)\right)$;
$\qquad \Theta_i^j = V \tilde{D} V'$;
$\quad$ **end**
$\quad$ ## $\{Z\}$ Update;
$\quad$ **for** l in $\{1, ..., p\}$ **do**
$\qquad$ **for** k in $\{l, ..., p\}$ **do**
$\qquad\quad x = \text{concat}\left(\left(\Theta_1^j - U_1^{j-1}\right)_{k,l}, ..., \left(\Theta_T^j - U_T^{j-1}\right)_{k,l}\right)$;
$\qquad\quad \hat{Z} = \text{FLSA}(x, \lambda_1, \lambda_2)$;
$\qquad\quad \left(Z_1^j, ..., Z_T^j\right)_{k,l} = \hat{Z}$;
$\qquad$ **end**
$\quad$ **end**
$\quad$ ## $\{U\}$ Update;
$\quad$ **for** i in $\{1, ..., T\}$ **do**
$\qquad U_i^j = U_i^{j-1} + \Theta_i^j - Z_i^j$;
$\quad$ **end**
$\quad$ **if** $\|\Theta_i^j - Z_i^j\|_2^2 < \epsilon_1$ and $\|Z_i^j - Z_i^{j-1}\|_2^2 < \epsilon_2, \ \forall i$ **then**
$\qquad$ Convergence=True;
$\quad$ **else**
$\qquad$ j = j + 1;
$\quad$ **end**
end
return $\Theta_1, ..., \Theta_T$

poral variability we rely on an entirely data-driven technique when choosing h that best describes the observations. The approach taken here is to use cross-validation (Silverman, 1986). As before, goodness-of-fit is employed to quantify how well estimated sample covariance matrices describe the observed time series. We define the leave-one-out (LOO) log-likelihood for the ith observation and some fixed choice of h as follows:

$$\mathcal{L}_{-i}(h) = -\frac{1}{2} \log \det\left(S_{-i}^{(h)}\right) - \frac{1}{2}\left(X_i - \mu_{-i}^{(h)}\right)'\left(S_{-i}^{(h)}\right)^{-1}\left(X_i - \mu_{-i}^{(h)}\right), \quad (18)$$

where both $\mu_{-i}^{(h)}$ and $S_{-i}^{(h)}$ are estimated with the ith observation removed for a given h. Thus $\mathcal{L}_{-i}(h)$ allows us to estimate the goodness-of-fit at X_i for any fixed h. We subsequently choose h in order to maximise the following score function:

$$CV(h) = \sum_{i=1}^T L_{-i}(h). \quad (19)$$

Parameters λ_1 and λ_2 determine the sparsity and temporal homogeneity of the estimated networks respectively. Therefore λ_1 and λ_2 directly affect the degrees of freedom of the estimated networks. In this case we can employ a more sophisticated parameter tuning technique based on the Akaike Information Criterion (AIC). The use of AIC allows us to estimate the in-sample prediction error for each choice of parameters λ_1 and λ_2, allowing for a clear comparison across different

---

[3] $\frac{p(p-1)}{2}$ edges and p more along the diagonal.

values of each parameter (Hastie et al., 2009, chap. 7). For any pair λ_1, λ_2 we define the AIC as:

$$AIC(\lambda_1, \lambda_2) = 2\sum_{i=1}^{T} (-\log\ det\ (\Theta_i) + \text{trace}\ (S_i\Theta_i)) + 2K \qquad (20)$$

where K is the estimated degrees of freedom. For a given range of λ_1 and λ_2 values an extensive grid-search is performed with the resulting choices of λ_1 and λ_2 being the pair that minimises AIC.

Following Tibshirani et al. (2005) we define K to be the number of non-zero coefficient blocks in $\{(\Theta_i)_{r,s}\}_{i=1,\ldots,T}$ for $1 \le r \ne s \le p$. That is, we count a sequence of one or more consecutive non-zero and equal estimates of partial correlations as one degree of freedom. This can be formally written as:

$$K = \sum_{r,s} \sum_{i=2}^{T} \mathbb{1}\Big((\Theta_i)_{r,s} \ne (\Theta_{i-1})_{r,s} \cap (\Theta_i)_{r,s} \ne 0\Big). \qquad (21)$$

Comparison to related work

There are currently limited methodologies available for estimating dynamic functional connectivity networks. A novel approach has recently been proposed in the form of the DCR algorithm (Cribben et al., 2012). The DCR is able to estimate functional connectivity networks by first partitioning time series into piece-wise stationary segments. This allows the DCR to exploit the vast literature relating to stationary network estimation. Formally, the DCR algorithm detects statistically significant change points by applying a block bootstrap permutation test. The use of a block bootstrap allows the DCR algorithm to account for autocorrelation present in fMRI data.

A common approach involves the use of a sliding window (Esposito et al., 2003; Allen et al., 2012; Leonardi et al., 2013). This involves recursively estimating covariance matrices S_i by re-weighting observations according to a sliding window or kernel. Subsequently, analysis can be performed directly on S_i to infer the network structure at the ith observation. This approach is studied in detail by Zhou et al. (2010). However sliding window approaches face the potential issue of variability between temporally adjacent networks. This arises as a direct consequence of the fact that each network is estimated independently without any mechanism present to encourage temporal homogeneity. We believe this additional variability can jeopardise the accuracy of the estimation procedure and can result in networks which do not reflect the true network structure over time. The SINGLE algorithm addresses precisely this problem by introducing an additional Fused Lasso penalty. In this way, changes in the connectivity structure are only reported when strongly validated by the data. The effect of the additional Fused Lasso penalty is studied extensively in the simulation study provided in Experimental results section.

Finally, the SINGLE algorithm is formally related to the Joint Graphical Lasso (JGL) (Danaher et al., 2013). The JGL was designed with the motivation of improving network inference by leveraging information across related observations and data sets. However, while the JGL focuses on stationary network estimation the SINGLE algorithm is designed to estimate dynamic networks. This manifests itself in two main differences to the overall objective functions of each of the algorithms. Firstly, the SINGLE algorithm only employs the Fused Lasso penalty as the Group Lasso penalty proposed in Danaher et al. (2013) cannot be used in the context of temporal homogeneity. This is due to the fact that the Group Lasso penalty encourages all coefficients to either be zero or non-zero in unison and therefore ignores temporal behaviour. Secondly, while both algorithms contain a Fused Lasso penalty the nature of these penalties are vastly different. In the case of the JGL there is no natural ordering to observations and therefore *fusions* are present between all networks (i.e., the penalty is of the form $\sum_{i \ne j} \|\Theta_i - \Theta_j\|_1$). This is not the case in the SINGLE algorithm where there is a chronological ordering. This results in a penalty of the form $\sum_{i=2}^{T} \|\Theta_i - \Theta_{i-1}\|_1$.

Software

The SINGLE algorithm is freely available as an R package, and can be downloaded along with its documentation from the Comprehensive R Archive Network (CRAN) (Monti et al., 2014).

Experimental data

The data was collected from 24 healthy subjects performing a simple but attentionally demanding cognitive task. This fMRI data set is particularly challenging as the BOLD time series has a length of 126 and corresponds to 18 ROIs, implying a low ratio of observations to dimensionality. We expect there to be a change in correlation structure approximately every 15 time points. Thus the number of observations available to estimate each connectivity structure is small relative to the number of ROIs.

In the CRT task, 24 subjects were presented with an initial fixation cross for 350 ms. This was followed by a response cue in the form of an arrow in the direction of the required response and lasting 1400 ms. The inter-stimulus interval was 1750 ms. Finger-press responses were made with the index finger of each hand. Subjects were instructed to respond as quickly and as accurately as possible. To maximise design efficiency, stimulus presentation was blocked, with five repeated blocks of 14 response trials interlaced with five blocks of 14 rest trials, and four response trials at the start of the experiment. This resulted in a total of 74 response trials per subject.

Image pre-processing involved realignment of EPI images to remove the effects of motion between scans, spatial smoothing using a 6 mm full-width half-maximum Gaussian kernel, pre-whitening using FILM and temporal high-pass filtering using a cut-off frequency of 1/50 Hz to correct for baseline drifts in the signal. The cut-off frequency was determined based on the combined length of task and rest (Smith et al., 2004; Poldrack et al., 2011). FMRIB's Linear Image Registration Tool (FLIRT) (Smith et al., 2004) was used to register EPI functional data sets into standard MNI space using the participant's individual high-resolution anatomical images.

The nodes were eighteen cortical spherical regions based on Pandit et al. (2013). Briefly, these nodes were defined based on peak regions from a spatial group independent components analysis of resting state fMRI. The regions were chosen for the nodes to encompass a wide range of cortical regions including regions within two well recognised functional connectivity networks, the fronto-parietal cognitive control network (FPCN) and default mode network (DMN) regions, as well as

Table 1
Regions and MNI coordinates.

Number	Name	MNI coordinates (mm)		
1	Left lateral parietal (DMN)	−46	−62	24
2	Right lateral parietal (DMN)	50	−54	16
3	Posterior cingulate cortex (DMN)	−2	−46	20
4	Ventromedial PFC (DMN)	2	54	8
5	Ventromedial PFC (FPCN)	−2	54	20
6	Dorsal anterior cingulate/preSMA (FPCN)	2	26	56
7	Left inferior frontal gyrus (FPCN)	−46	22	−12
8	Right inferior fronal gyrus (FPCN)	54	22	−4
9	Left inferior parietal (FPCN)	−54	−54	20
10	Right inferior parietal (FPCN)	54	−54	16
11	Left superior temporal sulcus (FPCN)	−50	−26	−12
12	Right superior temporal sulcus (FPCN)	54	−22	−12
13	Posterior cingulate cortex (FPCN)	−2	−50	24
14	Left motor	−38	−22	52
15	Left primary auditory	−54	−18	0
16	Primary visual	2	−74	4
17	Right motor	34	−22	52
18	Right primary auditory	62	−18	8

motor, visual and auditory cortical regions (see Table 1). For each subject and node the mean time-course from within a 10 mm diameter sphere centred on each of the 18 peaks was calculated. Six motion parameters, estimated during realignment, were filtered out of each time-course, using linear regression. The resulting 18 time-courses were subsequently used.

Experimental results

Simulation settings

In this section we evaluate the performance of the SINGLE algorithm through a series of simulation studies. In each simulation we produce simulated time series data giving rise to a number of connectivity patterns which reflect those reported in real fMRI data. The objective is then to measure whether our proposed algorithm is able recover the underlying patterns. That is, we are interested primarily in the correct estimation of the presence or absence of edges.

There are two main properties of fMRI data which we wish to recreate in the simulation study. The first is the high autocorrelation which is typically present in fMRI data (Poldrack et al., 2011). The second and main property we wish to recreate is the structure of the connectivity networks themselves. It is widely reported that brain networks have a small-world topology as well as highly connected hub nodes (Bullmore and Sporns, 2009) and we therefore look to enforce these properties in our simulations.

Vector Autoregressive (VAR) processes are well suited to the task of producing autocorrelated multivariate time series as they are capable of encoding autocorrelations within components as well as cross correlations across components (Cribben et al., 2012). The focus of these simulations is to study the performance of the proposed method in the presence of non-stationarity data. As a result the simulated datasets are only locally stationary. This is achieved by concatenating multiple VAR process which are simulated independently.

Moreover, when simulating connectivity structures we study the performance of the proposed algorithm using three types of random graphs; Erdős–Rényi random graphs (Erdos and Renyi, 1959), scale-free random graphs obtained by using the preferential attachment model of Barabási and Albert (1999) and small-world random graphs obtained using the Watts and Strogatz (1998) model. Erdős–Rényi random graphs are included as they correspond to the simplest and most widely studied type of random network while the use of scale-free and small-world networks is motivated by the fact that they are each known to each resemble different aspects of fMRI networks.

When simulating Erdős–Rényi random networks we maintain the edge strength of the connectivity between nodes fixed at 0.6. However, when simulating scale-free and small-world networks we randomly sample the edge strengths uniformly from $\left[-\frac{1}{2}, -\frac{1}{4}\right] \cup \left[\frac{1}{4}, \frac{1}{2}\right]$. This additional variability in the edge strength together with the reduced expected magnitude of each edge further increases the difficulty of the estimation task.

Each of the simulations considered in this section is aimed at studying the performance of the proposed algorithm in a different scenario. We begin by considering the overall performance of the SINGLE algorithm by generating connectivity structures according to Erdős–Rényi, scale-free and small-world networks in simulations Ia, Ib and Ic respectively. In many task-based experiments it is the case that the task is repeated several times, thus we expect there to be cyclic behaviour within the true functional connectivity structure (i.e., connectivity alternates between two structures) and we study this scenario in simulations IIa, IIb and IIc. In simulation III we study the performance of the algorithm as the ratio of observations, n, to nodes, p, decreases. This simulation is critical as it is often the case that there is a low ratio of observations to nodes, especially when considering subject specific fMRI data. In simulation VI we quantify the computational cost of the SINGLE algorithm. Throughout each of these simulations we benchmark the performance of the SINGLE algorithm against both the DCR algorithm and two sliding window based algorithms. Here a sliding window is employed to obtain time-dependent estimates of the sample covariance matrices and the Graphical Lasso is subsequently used to estimate a sparse connectivity structure. In order to ensure a fair comparison, the sliding window approach is employed using both a uniform kernel as well as a Gaussian kernel. A summary of all simulations can be found in Table 2. Finally, in each simulation the parameters for the SINGLE algorithm where estimated as discussed in Parameter tuning section; thus kernel width h was estimated by maximising leave-one-out log-likelihood and λ_1 and λ_2 were estimated by minimising AIC. The same approaches were used to estimate the kernel width and sparsity parameter, λ_1, for the sliding window and Gaussian kernel algorithms respectively. In the case of the DCR algorithm the block size was set to be ξT where $\xi = 0.05$ as suggested in Cribben et al. (2012) and T is the total number of observations.

Performance measures

We are primarily interested in the estimation of the functional connectivity graphs at every time point. In our setting this corresponds to correctly identifying the non-zero entries in estimated precision matrices, $\hat{\Theta}_i$, at each $i = 1, ..., T$. An edge is assumed to be present between the jth and kth nodes if $\left(\hat{\Theta}_i\right)_{j,k} \neq 0$. At the ith observation we define the set of all reported edges as $D_i = \left\{(j, k) : \left(\hat{\Theta}_i\right)_{j,k} \neq 0\right\}$. We define the corresponding set of true edges as $T_i = \{(j, k) : (\Theta_i)_{j,k} \neq 0\}$ where we write Θ_i to denote the true precision matrix at the ith observation. Given D_i and T_i we consider a number of performance measures at each observation i.

First we measure the precision, P_i. This measures the percentage of reported edges which are actually present (i.e., true edges). Formally, the precision is given by:

$$P_i = \frac{|D_i \cap T|}{|D_i|}.$$

Table 2
A summary of simulation settings and motivation behind each simulation.

Simulation	Network	Interval length	Edge strength	Properties & motivation
Ia	Erdős–Rényi	$n = 100$	0.6	Simplest and most widely used random network
Ib	Scale-free	$n = 100$	$[-12, -14] \cup [14, 12]$	Networks with highly connected hub nodes present
Ic	Small-world	$n = 100$	$[-12, -14] \cup [14, 12]$	Networks with small-world topology and high local clustering
IIa	Erdős–Rényi	$n = 100$	0.6	Cyclic network structure which may be expected in task-based fMRI studies
IIb	Scale-free	$n = 100$	$[-12, -14] \cup [14, 12]$	
IIc	Small-world	$n = 100$	$[-12, -14] \cup [14, 12]$	
IIIa	Scale-free	$n \in \{10, ..., 100\}$	$[(-12, -14] \cup [14, 12]$	The ratio of observations, n, to the number of ROIs, p, is decreased in order to
IIIb	Small-world	$n \in \{10, ..., 100\}$	$[(-12, -14] \cup [14, 12]$	study performed in the presence of rapid changes

Second we also calculate the recall, R_i, formally defined as:

$$R_i = \frac{|D_i \cap T_i|}{|T_i|}.$$

This measures the percentage of true edges which were reported by each algorithm. Ideally we would like to have both precision and recall as close to one as possible. Finally, the F_i score, defined as

$$F_i = 2 \frac{P_i R_i}{P_i + R_i}, \tag{22}$$

summarises both the precision and recall by taking their harmonic mean.

Simulation Ia — Erdős–Rényi random networks

In order to obtain a general overview of the performance of the SINGLE algorithm we simulate data sets with the following structure: each data set consists of 3 segments each of length 100 (i.e., overall duration of 300). The correlation structure for each segment was randomly generated using an Erdős–Rényi random graph. Finally a VAR process for each corresponding correlation structure was simulated. Thus each data set consists of 2 change points at times $t = 100$ and 200 respectively resulting in a network structure that is piece-wise constant over time. For this simulation the random graphs were generated with 10 nodes and the probability of an edge between two nodes was fixed at $\theta = 0.1$ (i.e., the expected number of edges is $\theta \frac{p(p-1)}{2}$).

In the case of the SINGLE algorithm the value of h was estimated by maximising the leave-one-out log-likelihood given in Eq. (19). Values of λ_1 and λ_2 were estimated by minimising AIC. For the DCR algorithm, the block size for the block bootstrap permutation tests to be 15 and one thousand permutations where used for each permutation test. In the case of the sliding window and Gaussian kernel algorithms the kernel width was estimated using leave-one-out log-likelihood and λ_1 was estimated by minimising AIC.

In Fig. 3 shows the average F_t scores for each of the four algorithms over 500 simulations. We can see that the SINGLE algorithm performs competitively relative to the other algorithms. Specifically we note that the performance of the SINGLE algorithm mimics that of the Gaussian kernel algorithm. We also note that all four algorithms experience a dramatic drop in performance in the vicinity of change points. This effect is most pronounced for the sliding window algorithm.
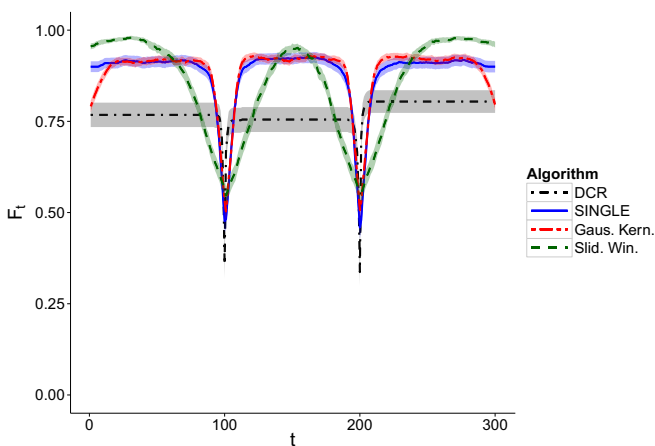
Simulation Ib — Scale-free networks

It has been reported that brain networks are scale-free, implying that their degree distribution follows a power law. From a biological perspective this implies that there are a small but finite number of hub regions which have access to most other regions (Eguiluz et al., 2005). While Erdős–Rényi random graphs offer a simple and powerful model from which to simulate random networks they fail to generate networks where the degree distribution follows a power law. In this simulation we analyse the performance of the SINGLE algorithm by simulating random networks according to the Barabási and Albert (1999) preferential attachment model. Here the power of preferential attachment was set to one. Additionally, edge strength was also simulated according to a uniform distribution on $\left[-\frac{1}{2}, -\frac{1}{4}\right] \cup \left[\frac{1}{4}, \frac{1}{2}\right]$, introducing further variability in the estimated networks.

In Fig. 4 we see the average F_t scores for each of the four algorithms over 500 simulations. We note that the performance of the SINGLE and DCR algorithms is largely unaffected by the increased complexity of the simulation. This is not true in the case of the sliding window and Gaussian kernel algorithms, both of which see their performance drop. We attribute this drop in performance to the fall in the signal-to-noise ratio and to the increased complexity of the network structure. Similar results confirming that networks with skewed degree distributions (e.g., power-law distributions) are typically harder to estimate have also been described in Peng et al. (2009). Finally,

Simulation Ic — Small-world networks

It has been widely reported that brain networks have a small-world topology (Stephan et al., 2000; Sporns et al., 2004; Bassett and Bullmore, 2006). In this simulation, multivariate time series were simulated such that the correlation structure follows a small-world graph according to the Watts–Strogatz model (Watts and Strogatz, 1998). Starting with a regular lattice, this model is parameterised by $\beta \in [0, 1]$ which quantifies the probability of randomly rewiring an edge. This results in networks where there is a tendency for nodes to form clusters, formally referred to as a high clustering coefficient. Both anatomical (Sporns et al., 2004) as well as the functional brain networks have been reported as exhibiting such a network topology (Bassett and Bullmore, 2006). Throughout this simulation we set $\beta = \frac{3}{4}$ and edge strength was simulated according to a Uniform distribution on $\left[-\frac{1}{2}, -\frac{1}{4}\right] \cup \left[\frac{1}{4}, \frac{1}{2}\right]$.
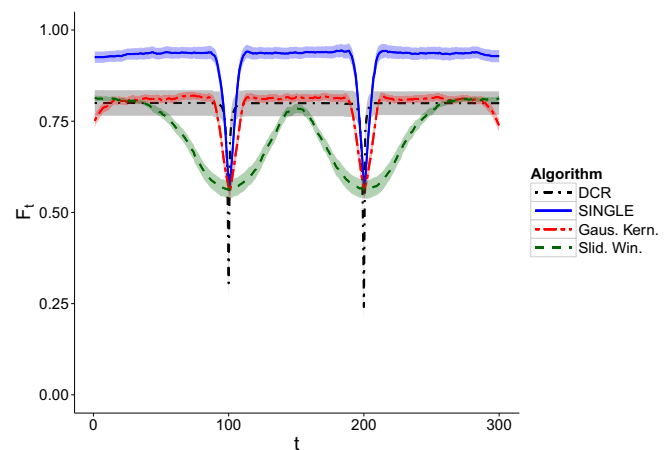

Fig. 3. Mean *F* scores for Simulation Ia (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using Erdős–Rényi random networks and a change occurred every 100 time points. We note that all four algorithms experience a drop in performance in the vicinity of these change points.

Fig. 4. Mean *F* scores for Simulation Ib (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using scale-free random networks according to the preferential attachment model of Barabási and Albert (1999). A change occurred every 100 time points. We note that all four algorithms experience a drop in performance in the vicinity of these change points. A full description of simulations is provided in Table 2.

In Fig. 5 we see the average F_t scores for each of the four algorithms over 500 simulations. We note that there is a clear drop in the performance of all the algorithms relative to their performance in simulations Ia and Ib. We believe this is due to the increased complexity of small-world networks compared to the previous networks we had considered. Formally, due to the high local clustering present in small-world networks, the path length between any two nodes is relatively short. As a result, we expect there to be a large number of correlated variables that are not directly connected. It has been reported that the Lasso (and therefore by extension the Graphical Lasso) cannot guarantee consistent variable selection in the presence of highly correlated predictors (Zou and Hastie, 2005; Zou, 2006). Since all four algorithms are related to the Graphical Lasso, we conjecture that this may be the cause of the overall drop in performance.

Simulation IIa — Cyclic Erdős–Rényi networks

In task related experiments subjects are typically asked to alternate between performing a cognitive task and resting. As a result, we expect the functional connectivity structure to alternate between two states: a task related state and the resting state. In order to recreate this scenario, network structures are simulated in a cyclic fashion such that the first and third correlation structures are identical.

We note from Fig. 6 that the performance of the SINGLE sliding window and Gaussian kernel algorithms is largely unaffected. However the DCR algorithm suffers a clear drop in performance relative to simulation Ia. The drop in performance of the DCR algorithm is partly due to the presence of the recurring correlation structure. More specifically, we believe the problem to be related to the use of block bootstrapping permutation test to determine the significance of change points in the DCR. This test assumes that *local* data points are correlated but expects data points that are *far away* to be independent. Typically this assumption holds. However when there is a recurring correlation structure, points that are far away may follow the same underlying distribution. As a result the power of the permutation test is heavily reduced.

Simulation IIb — Cyclic scale-free networks

In this simulation we simulate multivariate time series where the underlying correlation structure is cyclic and follows a scale-free distribution. The results are summarised in Fig. 7. As in simulation Ib there is no noticeable difference in the performance of the SINGLE algorithm. There is however a drop in the performance of the sliding window, Gaussian kernel and DCR algorithms. This is particularly evident in the
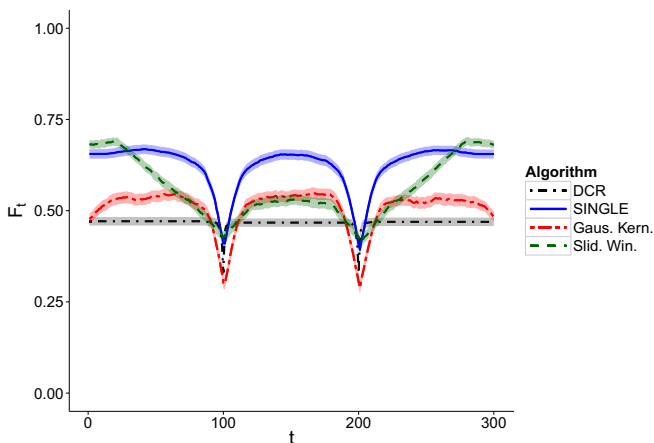

Fig. 6. Mean *F* scores for Simulation IIa (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using Erdős–Rényi random networks with the additional constraint that the first and third correlation structure be identical. This gives rise to cyclic correlation structures which may be present in task-based studies.

case of the DCR algorithm. As mentioned previously the drop in performance of the sliding window and Gaussian kernel algorithms is due to the increased complexity of the network structure as well as the fall in the signal to noise ratio. In the case of the DCR the drop in performance can be partly explained by the fact the assumptions behind the use of the block bootstrap no longer hold (see simulation IIa for a discussion) and the increased complexity of the network structure. These two factors combine to greatly affect the performance of the DCR algorithm.

Simulation IIc — Cyclic small-world networks

In this simulation we look to assess the performance of the SINGLE algorithm in a scenario that is representative of the experimental data we use in this work. As described previously, the experimental data used in this study corresponds to fMRI data from a Choice Reaction Time (CRT) task. Here subjects are required to make rapid visually-cued motor decisions. Stimulus was presented in five on-task blocks each preceding a period where subjects were at rest. As a result we expect there to be a cyclic network structure.

Fig. 5. Mean *F* scores for Simulation Ic (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using small-world random networks according to the Watts–Strogatz model. A change occurred every 100 time points. We note that all four algorithms experience a drop in performance in the vicinity of these change points. A full description of simulations is provided in Table 2.
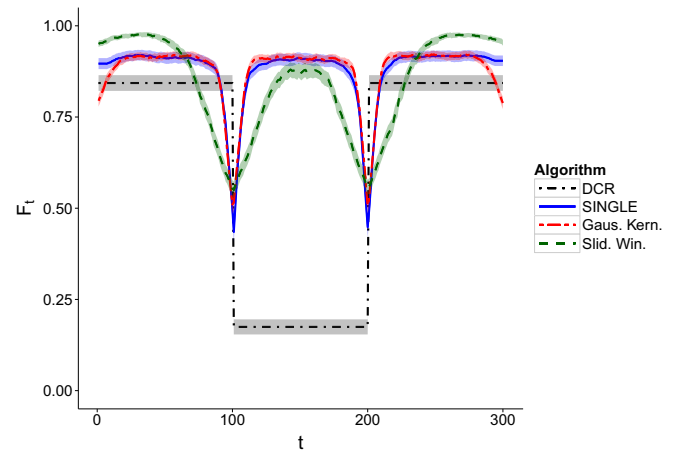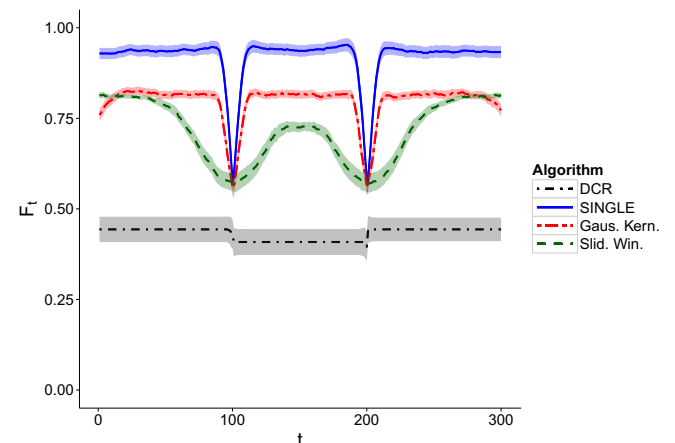

Fig. 7. Mean *F* scores for Simulation IIb (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using scale-free random networks with the additional constraint that the first and third correlation structure be identical. This gives rise to cyclic correlation structures which may be present in task-based studies.

Thus in this simulation network structures are simulated in a cyclic fashion where each network structure is simulated according to a small-world network as in Simulation Ic. This simulation gives us a clear insight into the performance of the SINGLE algorithm in a scenario that is very similar to that proposed in the experimental data.

The results are summarised in Fig. 8. We note that as in Simulation Ic there is drop in the performance of all four algorithms relative to their performance in simulations IIa and IIb. We believe this is due to the increased complexity of the underlying networks structures, specifically the high levels of clustering we experience in small-world networks which are not seen in Erdős–Rényi or scale-free random networks.

Simulation IIIa — Scale-free networks with decreasing n/p ratio

Here we study the behaviour of the proposed algorithm as the ratio of observations, n, to the number of nodes, p, decreases. This is a particularly relevant problem in the case of fMRI data as it is often the case that the number of nodes in the study (typically the number of ROIs) will be much larger than the number of observations.

In this simulation we fix $p = 10$ and allow the value of n to decrease. Here we simulate a data set with three segments each of length n where the connectivity structure within each segment is randomly simulated according to a scale-free network. Thus as the value of n decreases we are able to quantify the performance of the SINGLE algorithm in the presence of rapid changes in network structure.

In the case of the SINGLE, sliding window and Gaussian kernel algorithms all parameters are estimated as discussed previously. In the case of the DCR algorithm the value of block sizes for the block bootstrap test was also reduced accordingly.

Results for Simulation IIIa are given in Fig. 9. Error bars have been removed in the interest of clarity however detailed results are available in Table 3. We note that all four algorithms struggle when n is small relative to p. This is to be expected as the number of observations is much smaller than the number of parameters to be estimated. Fig. 9 shows that the performance of the SINGLE algorithm quickly improves as n increases at a rate which is similar to that of the sliding window and Gaussian kernel algorithms.

Simulation IIIb — Small-world networks with decreasing n/p ratio

As with Simulation IIIa, we evaluate the performance of the proposed algorithm as the ratio of observations, n, relative to the dimensionality of the data, p, decreases. However, here the underlying network structure are simulated according to small-world networks. This simulation
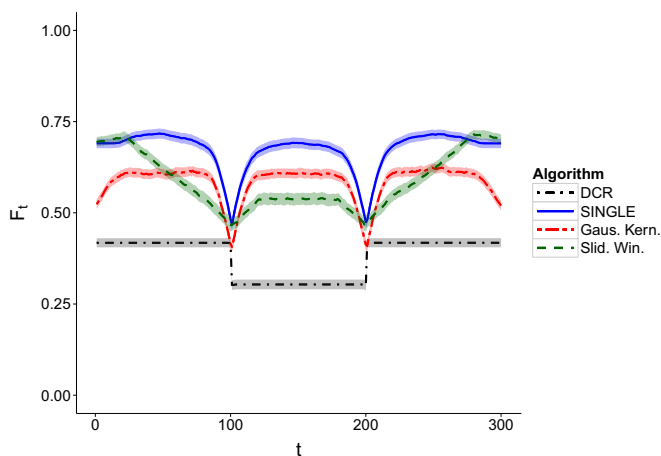

Fig. 9. Results for Simulation IIIa for the SINGLE, DCR and sliding window algorithms respectively over 500 simulations. Here networks were simulated using scale-free random networks and the performance of each algorithm was studied as the ratio of observations, n, to the number of nodes, p, decreased. Here $p = 10$ was fixed as n decreased.

therefore provides an insight into how accurately proposed algorithm is able to estimate networks in the presence of rapid changes.

Results for Simulation IIIb are given in Fig. 10 and detailed results are provided in Table 4. As with the previous simulations we note that the performance of all four algorithms is affected by the presence of small-world networks (see simulation Ic for a discussion). Furthermore, as in simulation IIIa, the performance of all four algorithms also deteriorates as the ratio n/p decreases. Moreover, as in simulation IIIa, the performance of the SINGLE algorithm improves as n/p increases.

Computational cost

From a practical perspective we are also interested in the computational cost of the SINGLE algorithm. While this has already been discussed previously we look to benchmark the computational cost of the SINGLE algorithm relative to the previously considered algorithms.

As noted in Computational complexity section, the limiting factor in the computational cost of the SINGLE algorithm is the number of nodes, p. We note that this is also the case for the sliding window, Gaussian kernel and DCR algorithms (see Appendix F). As a result we

Fig. 8. Mean F scores for Simulation IIc (shaded regions represent 95% confidence intervals). Here the underlying network structure was simulated using small-world random networks with the additional constraint that the first and third correlation structure be identical. This gives rise to cyclic correlation structures which may be present in task-based studies.
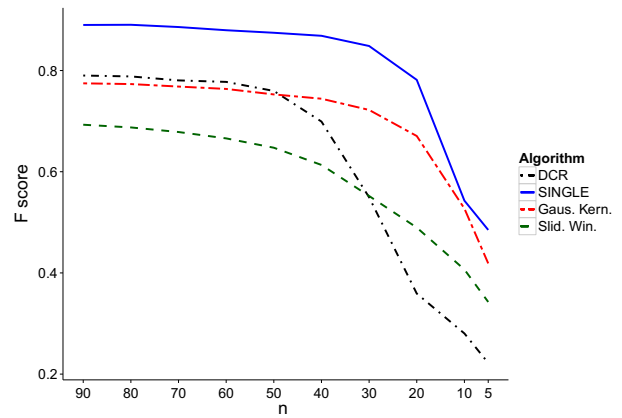
Table 3
Detailed results from Simulation IIIa. For each algorithm the mean F score, μ, is reported together with the sample standard deviation, σ.

l	μ	σ	l	μ	σ
5	0.22	0.07	5	0.49	0.14
10	0.28	0.09	10	0.54	0.13
20	0.36	0.15	20	0.78	0.08
30	0.55	0.21	30	0.85	0.06
40	0.70	0.15	40	0.87	0.05
50	0.76	0.08	50	0.87	0.05
60	0.78	0.07	60	0.88	0.05
70	0.78	0.06	70	0.89	0.04
80	0.79	0.03	80	0.89	0.04
90	0.79	0.02	90	0.89	0.04
(a) DCR			(b) SINGLE		

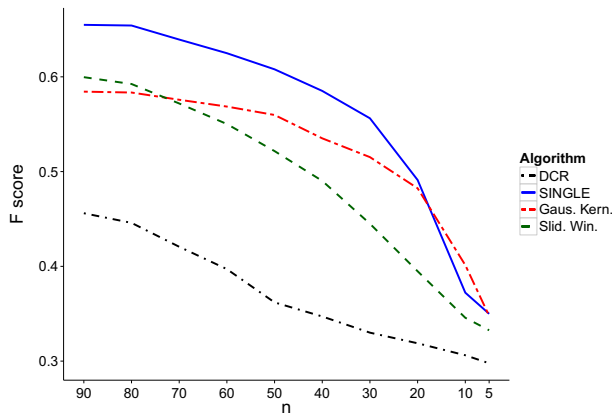| l	μ	σ	l	μ	σ
5	0.42	0.10	5	0.35	0.09
10	0.53	0.09	10	0.41	0.09
20	0.67	0.07	20	0.49	0.10
30	0.72	0.05	30	0.55	0.10
40	0.74	0.05	40	0.61	0.09
50	0.75	0.04	50	0.65	0.07
60	0.76	0.04	60	0.67	0.07
70	0.77	0.03	70	0.68	0.06
80	0.77	0.03	80	0.69	0.05
90	0.77	0.03	90	0.69	0.05
(c) Gaussian Kernel			(d) Sliding window		

Fig. 10. Results for Simulation IIIb for the SINGLE, DCR and sliding window algorithms respectively over 500 simulations. Here networks were simulated using small-world random networks and the performance of each algorithm was studied as the ratio of observations, n, to the number of nodes, p, decreased. Here $p = 10$ was fixed as n decreased.
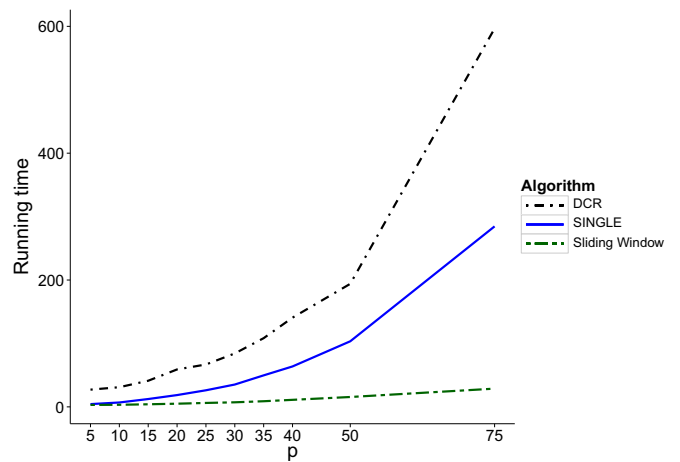

Fig. 11. Average running time (seconds) for each algorithm for increasing p. For clarity the sliding window and Gaussian kernel approaches have been plotted together as they have the same computational complexity.

compare the running times of each of the algorithms as p increases for fixed $n = 100$.

Here each data set was simulated in the same manner as in Simulation Ic. That is, each dataset consisted of 3 segments of length 100 (resulting in an overall duration of 300). The correlation structure within each segment was then randomly generated according to small-world network.

In Fig. 11 we plot the mean running time of each algorithm over 50 iterations for increasing p. It is clear that the computational cost of the SINGLE algorithm increases exponentially with p. However we note that for $p = 75$ nodes the algorithm can still be run in under 5 min, making it practically feasible. This simulation was run on a computer with an INTEL CORE I5 CPU at 2.8 GHz.

Application to a choice reaction time (CRT) task fMRI dataset

In this section we assess the ability of the SINGLE algorithm when detecting changes in real fMRI data evoked using a simple cognitive

Table 4
Detailed results from Simulation IIIb. For each algorithm the mean F score, μ, is reported together with the sample standard deviation, σ.

l	μ	σ	l	μ	σ
5	0.29	0.05	5	0.35	0.08
10	0.31	0.06	10	0.37	0.08
20	0.32	0.07	20	0.49	0.07
30	0.33	0.08	30	0.56	0.07
40	0.35	0.10	40	0.59	0.07
50	0.36	0.11	50	0.61	0.07
60	0.40	0.12	60	0.62	0.07
70	0.42	0.11	70	0.64	0.07
80	0.45	0.10	80	0.65	0.06
90	0.46	0.10	90	0.65	0.06
(a) DCR			(b) SINGLE		

l	μ	σ	l	μ	σ
5	0.35	0.06	5	0.33	0.06
10	0.40	0.06	10	0.35	0.07
20	0.48	0.05	20	0.39	0.08
30	0.52	0.05	30	0.44	0.08
40	0.54	0.05	40	0.49	0.09
50	0.56	0.05	50	0.52	0.08
60	0.57	0.05	60	0.55	0.07
70	0.58	0.05	70	0.57	0.07
80	0.58	0.05	80	0.59	0.06
90	0.58	0.05	90	0.60	0.06
(c) Gaussian kernel			(d) Sliding window		

task, the Choice Reaction Time (CRT) task. The CRT is a forced choice visuo-motor decision task that reliably activates visual, motor and many cognitive control regions. The task was blocked into alternating task and rest periods. As a result we expect the task onset to evoke an abrupt change in the correlation structure that is cyclical in nature.

This is a highly challenging data set for several reasons. Firstly, it corresponds to the scenario where $n/p = 126/18$ is small. Secondly, there is a high rate of change in the correlation structure with a change in cognitive state roughly every 15 s. Finally, given the nature of the CRT task there is a recurring correlation structure with subjects alternating between two cognitive states: resting and performing the CRT task. As we have seen in the simulations the SINGLE algorithm is well equipped to handle the aforementioned challenges.

In order to study the roles of the various ROIs during the CRT task we consider the changes in betweenness centrality of each node over time. The betweenness centrality of a node is the sum of how many shortest paths between all other nodes pass through it (Pandit et al., 2013). Nodes with high betweenness centralities are considered to be of important, hub nodes in the network (Hagmann et al., 2008).

As described previously the CRT task involves subjects alternating between performing a visual stimulus task (on task) and resting state (off task). Fig. 12 shows the average estimated functional connectivity networks for a patient on and off task respectively. Here the size of each node is proportional to the sum of the betweenness centralities of the corresponding ROI and the edge thickness is proportional to the partial correlation between nodes. We notice that the on task network is appears to be more *focused* that the off task network and this can also be seen in the corresponding example video provided in the Supplementary material.

We note that there are changes in the betweenness centralities of several nodes between tasks. In order to determine the significance of any changes betweenness centrality as a result of the changing cognitive state of the subjects we study the estimated graphs for each of the 24 subjects both on and off task. Fig. 13 shows the percentage change in betweenness centrality from off task to on task for each ROI. To determine the statistical significance of reported changes a Wilcoxon rank sum test was employed. The resulting p-values where adjusted according to the Bonferroni–Holm method in order to account for multiple tests. The results indicated that at the $\alpha = 5\%$ level there was a statistically significant increase in betweenness centrality for the 8th (Right Inferior Frontal Gyrus) and 10th (Right Inferior Parietal) ROIs. This indicates that during this simple, cognitive task the Right Inferior Frontal Gyrus and the Right Inferior Parietal become more hub-like. This is particularly true in the case of the Right Inferior Frontal Gyrus where the change in betweenness centrality is particularly sizeable.
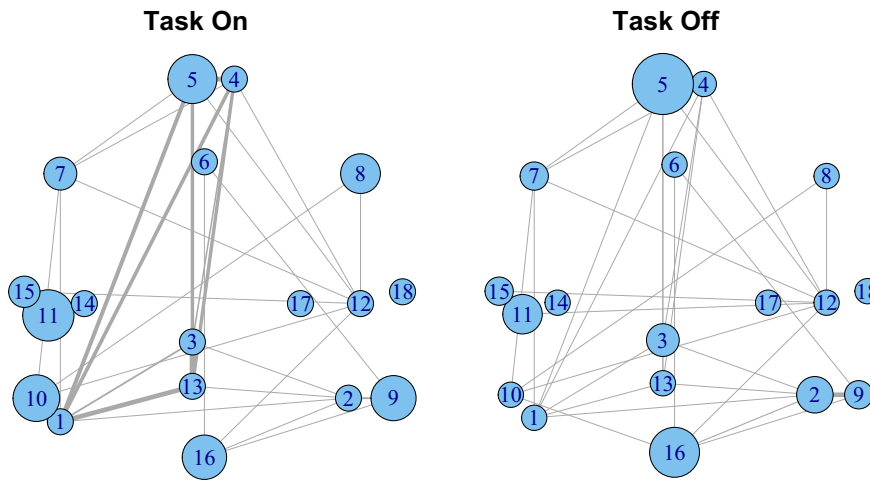
Task On **Task Off**

Fig. 12. Mean estimated graphs on and off task for a given subject. Here node size is proportional to betweenness centrality and edge width is proportional to the magnitude of their partial correlations. Each node corresponds to a ROI given in Table 1. A movie of the estimated networks showing the complete evolution is available in the Supplementary material.

These findings suggest that the Right Inferior Frontal Gyrus and Right Inferior Parietal play a key role in cognitive control and executive functions as demonstrated by their dynamically changing betweenness centrality throughout the task. This result agrees with the proposed functional roles for the Right Inferior Frontal Gyrus (and adjacent right anterior insula), which is assumed to play a fundamental role in attention and executive function during cognitively demanding tasks and may have an important role in regulating the balance between other brain regions (Aron et al., 2003; Hampshire et al., 2010; Bonnelle et al., 2012). The findings also agree with the proposed function of the Right Inferior Parietal lobe, which has been reported to play a role in high-level cognition (Mattingley et al., 1998) and sustaining attention (Corbetta and Shulman, 2002; Husain and Nachev, 2007).

One possible interpretation of the increase in betweenness centrality is that the Right Inferior Frontal Gyrus becomes more important for the flow of information around the brain during the more challenging cognitive task.

At a larger scale, these results agree with various reports of functional re-organisation driven by external stimuli. For example, function connectivity has been reported to change based on task learning (Bassett et al., 2010) and task preparation (Ekman et al., 2012).

Difference in Betweenness Centrality

Fig. 13. Estimated percentage change in betweenness centrality from off task to on task over all 24 patients. Each column corresponds to a ROI given in Table 1. The * indicates a statistically significant difference in betweenness centrality at $\alpha = 5\%$ level after correcting for multiple hypotheses.

Discussion

In this work we introduce the Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm, a new methodology for estimating sparse dynamic functional connectivity networks from non-stationary fMRI data. Our approach provides two main advantages. First, the proposed algorithm is able to accurately estimate functional connectivity networks at each observation. This allows for the quantification the dynamic behaviour of brain networks at a high temporal granularity. The second advantage lies in the SINGLE algorithm's ability to quantify network variability over time. In SINGLE, networks are estimated simultaneously in a unified framework which encourages temporal homogeneity. This results in networks with sparse innovations in edge structure over time and implies that changes in connectivity structure are only reported when substantiated by evidence in the data. Although the use of the SINGLE algorithm is particularly suitable for task related experiments, there is also a growing body of evidence to suggest that functional connectivity is highly non-stationary even in resting state, making the SINGLE algorithm well suited for these studies as well.

The SINGLE algorithm is closely related to sliding window based algorithms. We note that Zhou et al. (2010) have extensively studied the combined use of kernel methods and constrained optimisation to estimate dynamic networks and provide a theoretical guarantee that accurate estimates of time varying network structure can be obtained in such a manner under mild assumptions. The approach taken there is to estimate sample covariance matrices at each $i \in \{1, \ldots, T\}$ using kernel methods with the Graphical Lasso being used subsequently to estimate the corresponding precision matrices. However, given T time points this approach corresponds directly to T independent iterations of the Graphical Lasso. As a result, while smooth estimates of the sample covariance matrix are obtained via the use of kernels, there is no mechanism in place to enforce temporal homogeneity in the corresponding precision matrices. Consequently the estimated partial correlations may not accurately represent the functional connectivity over time. The SINGLE algorithm addresses precisely this problem by directly enforcing temporal homogeneity. This is achieved via the introduction an additional constraint inspired by the Fused Lasso. As shown in our simulation study, this additional constraint results in higher accuracy of estimated networks in a vast array of scenarios. Similarly to other proposed methods for estimating dynamic function connectivity networks the proposed method is based on empirical covariance matrices (Allen et al., 2012; Cribben et al., 2012; Leonardi et al., 2013). Here the use of the covariance is driven by the nature of the proposed method which balances goodness-of-fit with constraints on sparsity and temporal homogeneity.
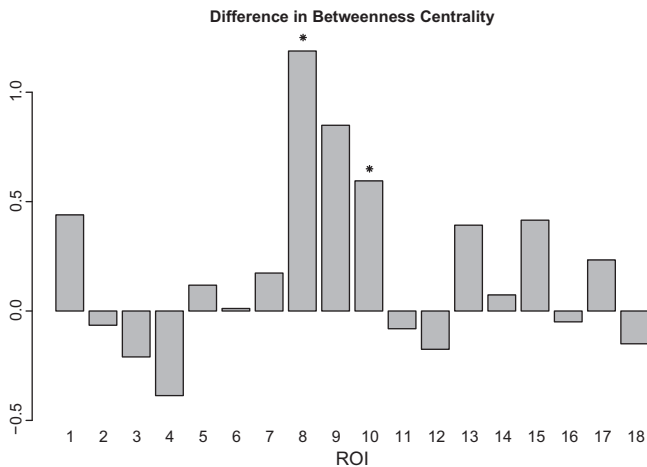
The former, given in Eq. (1), is defined on the covariance matrix. We note that the use of the covariance matrix is susceptible to being dominated by nodes of large relative variance, therefore care should be taken to ensure the time series of all nodes have comparable variance.

The SINGLE algorithm requires the input of 3 parameters, λ_1, λ_2 and h, each of which has a natural interpretation for the user. Penalty parameters λ_1 and λ_2 enforce sparsity and temporal homogeneity respectively. They can be tuned by minimising *AIC* over a given range of values. The choice of h can be interpreted as the window length and we provide an data-driven method for tuning parameter h using the leave-one-out log-likelihood. We note that the choice of h is a delicate matter as well as an active area of research in its own right. The choice of h can be seen as a trade-off between stability and temporal adaptivity. Setting h to be too large will result in network estimates that resemble the global mean and omit valuable short-term fluctuations in connectivity structure. Conversely, setting h to be too small will lead to networks that are dominated by noise and may jeopardise the reproducibility of the study (Hutchinson et al., 2013). Given this reasoning, it is often desirable to have a kernel width which is dependent on the location within the time series. This allows the kernel width to decrease in the proximity of a change-point (allowing for rapid temporal adaptivity) and increase \when data is piece-wise stationary (in order to fully exploit all relevant data). The idea of adaptive values of h has been studied in literature such as Haykin (2002) and Príncipe et al. (2011), however, we leave this for future work. A limitation of sliding windows is that the window size determines the time-scale of the analysis. As a result, it may also be interesting to study the results over a range of window sizes — allowing for the discovery of fluctuations over varying time-scales. Of course, the range of window sizes must take into consideration the aforementioned discussion and care must be taken to ensure reported fluctuations are not spurious. One potential alternative would be to consider a time–frequency analysis using a Fourier or wavelet transform. Such an approach would avoid the need for a fixed width h and provide rich description of functional relationships across multiple time scales.

Our simulation results indicate that the SINGLE algorithm can accurately estimate the true underlying functional connectivity structure when provided with non-stationary multi-variate time series data. We identify three relevant scenarios where the proposed method performs competitively. The first, demonstrated by simulation I, quantifies our claim that the SINGLE algorithm is able to accurately estimate dynamic functional connectivity networks. In task based experiments it is often the case that tasks are repetitively performed followed by a period of rest, resulting in the presence of a cyclic functional connectivity structure. This scenario is studied in simulation II which serves as an indication that the SINGLE algorithm is not adversely affected in such cases. Furthermore, we have shown that the SINGLE algorithm is relatively robust when the ratio of observations to nodes falls, meaning that the SINGLE algorithm can be applied on a subject-by-subject basis. This is a great advantage as it avoids the issue of subject-to-subject variability and allows for the estimation of functional connectivity networks for each subject. This potentially allows for estimated dynamic connectivity to be used to differentiate between subjects. Finally, the computational cost of the proposed algorithm is studied empirically in simulation VI. A summary of all the simulation results is provided in Table 5.

We do note that the performance of the SINGLE algorithm was affected by the presence of small-world network structure. We believe this may be caused by the high local clustering present in such networks. This results in the short minimum path lengths between many nodes. This would cause there be a large number of correlated nodes which are not directly connected. It has been reported that the Lasso (and by extension the Graphical Lasso) cannot guarantee consistent variable selection in the presence of highly correlated predictors (Zou and Hastie, 2005; Zou, 2006). This issue has recently been studied in the context of genetic networks by Peng et al. (2009) and in future these approaches could be adapted to address such issues.

We have presented an application showing that the SINGLE algorithm can detect cyclical changes in network structure with fMRI data acquired while subjects perform a simple cognitive task and identify the Right Inferior Frontal Gyrus as well as the Right Inferior Parietal as changing their graph theoretic properties as the functional connectivity network reorganises. We find that there is a significant increase in the betweenness centrality for both these regions. These findings suggest that the Right Inferior Frontal Gyrus together with the Right Inferior Parietal play a key role in cognitive control and the functional reorganisation of brain networks. In the case of the Right Inferior Frontal Gyrus, this result agrees with the proposed functional roles of the Right Inferior Frontal Gyrus (Aron et al., 2003; Corbetta and Shulman, 2002; Hampshire et al., 2010; Bonnelle et al., 2012). The Right Inferior Parietal lobe has also been reported to play a role in high-level cognition (Mattingley et al., 1998) and sustaining attention (Corbetta and Shulman, 2002; Husain and Nachev, 2007). One possible interpretation is that both the Right Inferior Frontal Gyrus and the Right Inferior Parietal become more important to the flow of information during the more challenging cognitive task as demonstrated by their rise in betweenness centrality.

In conclusion, the SINGLE algorithm provides an alternative and novel method for estimating the underlying network structure associated with dynamic fMRI data. It is ideally suited to analysing data where a change in the correlation structure is expected but little more is known. Going forward, the SINGLE algorithm can be applied to different types of fMRI data sets, exploring different cognitive tasks such as those with multiple task demands, exploring how networks change with more subtle differences in cognitive state (i.e., rather than just task on or off). Similarly, the approach can be used to investigate spontaneous network reorganisation in the resting state and compare this across different subject groups (e.g., comparing pathological states with healthy controls). From a methodological point of view it would be interesting to consider variations of the SINGLE objective function, particularly with respect to the Fused Lasso component of the penalty. For example, this component could be exchanged with a trend filtering penalty (Kim et al., 2009).

Appendix A

Here we formally derive some of the results discussed in the main text. Throughout this section we assume the following results relating to convex functions:

(1) A function $f : \mathbb{R}^{\mathbf{n}} \to \mathbb{R}$ is convex if and only if the function $g : \mathbb{R} \to \mathbb{R}$ where

$$g(t) = f(x + tv)$$
$$\mathbf{dom}\ g = \{t : x + tv \in \mathbf{dom}\ f\}$$

is convex in t for all $x \in \mathbf{dom}\ f$ and $v \in \mathbb{R}^{\mathbf{n}}$. Here we write $\mathbf{dom}\ f$ to denote the domain of function f.

Table 5

Comparative summary of each algorithm. A derivation of the computational cost of the DCR algorithm is provided in Appendix F where b refers to the number of bootstrap permutation tests performed at each iteration.

	SINGLE	DCR	Sliding window/ Gaussian kernel
Temporal adaptivity	✔	✔	✔
Temporal homogeneity	✔	✔	X
Cyclic correlation structure	✔	X	✔
Parameters	h, λ_1, λ_2	Δ, λ_1	h, λ_1
Computational Complexity	$\mathcal{O}(np^3 + p^2 n \log(n))$	$\mathcal{O}((n+b)p^3)$	$\mathcal{O}(np^3)$

(2) Assuming f is twice differentiable (i.e., its Hessian $\nabla f(x)$ exists for all $x \in \mathbf{dom}\ f$) then f will be convex if and only if its Hessian is positive semidefinite.
(3) The composition of convex functions is itself a convex function
(4) Any norm is convex (this follows from the definition of a norm and the triangle inequality)
(5) The sum of convex functions is convex

We note that some of these results are taken from Boyd and Vandenberghe (2004).

A The SINGLE objective function given in Eq. (3) is convex
Recall the SINGLE cost function was defined as:

$$f(\{\Theta_i\}) = \sum_{i=1}^{T} -\log\ det\ \Theta_i + \text{trace}\ (S_i\Theta_i) + \lambda_1 \sum_{i=1}^{T} \|\Theta_i\|_1 \tag{23}$$
$$+ \lambda_2 \sum_{|i-j|<k} \left\|\Theta_i - \Theta_j\right\|_1,$$

From Assumption (5) it suffices to show that each component of $f\left(\left\{\hat{\Theta}\right\}\right)$ is convex. Furthermore from Assumptions (5) and (5) it follows that $\|\Theta_i\|_1$ and $\|\Theta_i - \Theta_j\|_1$ are convex for all i and j. We note that trace $(\hat{S}_i\Theta_i) = \sum_{r=1}^{p}\sum_{q=1}^{p}(S_i)_{r,q} \cdot (\Theta_i)_{r,q}$. Therefore trace $(\hat{S}_i\Theta_i)$ is an affine function for all i as it is a linear sum. Finally we come to $-\log\ det\ \Theta_i$. It follows that showing that $-\log\ det\ \Theta_i$ is convex is equivalent to showing that $\log\ det\ \Theta_i$ is concave. In order to do so we use Assumption (5). Formally we define $f: \mathbb{R}_{++}^{p \times p} \to \mathbb{R}$ as $f(X) = \log\ det\ (X)$, where $\mathbb{R}_{++}^{p \times p}$ refers to the set of positive semi-definite p by p matrices. We also define $g(t) = \log\ det\ (X + tV)$ for all $V \in \mathbb{R}_{++}^{p \times p}$. Since X is positive semi-definite it follows that X has a square root $X^{-\frac{1}{2}}$. Thus in order to show that g is concave we can rewrite $X + tV$ as follows:

$$X + tV = X^{\frac{1}{2}}\left(I + tX^{-\frac{1}{2}}VX^{-\frac{1}{2}}\right)X^{\frac{1}{2}}$$

Thus we have that $g(t) = \log\ det\ (X + tV) = \log\ det\ (X) + \log\ det\ \left(I + tX^{-\frac{1}{2}}VX^{-\frac{1}{2}}\right)$.

Now we can take the eigendecomposition of $tX^{-\frac{1}{2}}VX^{-\frac{1}{2}} = \Omega\Lambda\Omega'$ where Ω is an orthonormal matrix of eigenvectors and Λ is a diagonal matrix where the ith entry along the diagonal is the ith eigenvalue, λ_i. Finally we note that:

$$\log\ det\ \left(I + tX^{-\frac{1}{2}}VX^{-\frac{1}{2}}\right) = \log\ det\ (\Omega(I + t\Lambda)\Omega')$$
$$= \sum_{i=1}^{p} \log\ (1 + t\lambda_i)$$

By differentiating $\log\ (1 + t\lambda)$ and using Assumption (5) we note that g is concave and thus conclude that $f(X) = \log\ det\ (X)$ is concave and that the SINGLE cost function is convex.

B. The scaled augmented Lagrangian corresponding to Eqs. (7) and (8) is given by $\mathcal{L}_\gamma(\{\Theta_i\}, \{Z_i\}, \{U_i\})$ as shown in Eq. (9).
In the case of Eqs. (7)–(8) the corresponding Lagrangian is given by:

$$\mathcal{L}(\{\Theta_i\}, \{Z_i\}, \{Y_i\}) = -\sum_{i=1}^{T}(\log\ det\ \Theta_i - \text{trace}\ (S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{T} \|Z_i\|_1$$
$$+ \lambda_2 \sum_{i=2}^{T} \|Z_i - Z_{i-1}\|_1 + \sum_{i=1}^{T} \text{vec}(Y_i)'\ \text{vec}(\Theta_i - Z_i) \tag{24}$$

where $Y_1, ..., Y_T, Y_i \in \mathbb{R}^{p \times p}$ are Lagrange multipliers or dual variables. The final term in the Lagrangian is equivalent to the sum of all elements in the matrix $Y_i \cdot (\Theta_i - Z_i)$.

The augmented Lagrangian is essentially composed of the original Lagrangian and an additional penalty term. In our case the augmented Lagrangian is given by:

$$\mathcal{L}(\{\Theta_i\}, \{Z_i\}, \{Y_i\}) = -\sum_{i=1}^{T}(\log\ det\ \Theta_i - \text{trace}\ (S_i\Theta_i)) + \lambda_1 \sum_{i=1}^{T} \|Z_i\|_1$$
$$+ \lambda \sum_{2i=2}^{T} \|Z_i - Z_{i-1}\|_1 + \sum_{i=1}^{T} \text{vec}(Y_i)'\ \text{vec}(\Theta_i - Z_i)$$
$$+ \gamma 2 \sum_{i=1}^{T} \|\Theta_i - Z_i\|_2^2 \tag{25}$$

We can simplify Eq. (25) by noting that $\text{vec}(Y_i)^T\text{vec}(\Theta_i - Z_i)$ is equivalent to the elementwise sum of entries of the matrix $Y_i \cdot (\Theta_i - Z_i)$ where $\cdot$ denotes the elementwise multiplication of matrices. Thus we can combine the linear and quadratic constraint terms as follows:

$$Y_i \cdot (\Theta_i - Z_i) + \gamma 2\|\Theta_i - Z_i\|_2^2 = \gamma 2\|\Theta_i - Z_i + (1\gamma)Y_i\|_2^2 - (12\gamma)\|Y_i\|_2^2 \tag{26}$$

$$= \gamma 2\|\Theta_i - Z_i + U_i\|_2^2 - \gamma 2\|U_i\|_2^2 \tag{27}$$

where $U_i = 1\gamma Y_i$ are the *scaled* Lagrange multipliers. This yields the scaled augmented Lagrangian given in Eq. (25).

C. If symmetric matrices $X, Y \in \mathbb{R}^{p \times p}$ satisfy $X^{-1} - \alpha X = Y$ for some constant α then it follows that X and Y have the same eigenvectors. Furthermore it is also the case that the ith eigenvectors of X and Y, denoted by λ_{X_i} and λ_{Y_i} respectively, will satisfy $\lambda_{X_i}^{-1} - \alpha\lambda_{X_i} = \lambda_{Y_i}$ for $i \in \{1, ..., p\}$
In order to prove claim 2 we begin taking the eigendecompositions of X and Y as $\Omega_X\Lambda_X\Omega_X$ and $\Omega_Y\Lambda_Y\Omega_Y$ respectively. Substituting these into $X^{-1} - \alpha X = Y$ we obtain:

$$(\Omega_X\Lambda_X\Omega_X)^{-1} - \alpha(\Omega_X\Lambda_X\Omega_X) = \Omega_Y\Lambda_Y\Omega_Y$$

Expanding the left hand side yields:

$$\Omega_X\Lambda_X^{-1}\Omega_X - \alpha(\Omega_X\Lambda_X\Omega_X) = \Omega_Y\Lambda_Y\Omega_Y$$
$$= \Omega_X\left(\Lambda_X^{-1} - \alpha\Lambda_X\right)\Omega_X$$

where we have made us of the fact that Ω_X is an orthonormal matrix. Thus it follows that $\Omega_X = \Omega_Y$ and since both Λ_X and Λ_Y are diagonal matrices we also have that $\lambda_{X_i}^{-1} - \alpha\lambda_{X_i} = \lambda_{Y_i}$ for $i \in \{1, ..., p\}$

D. Each of the $p^2 2$ optimisations of the form given in Eq. (16) can be solved by applying the Fused Lasso Signal Approximator.
The Lasso is a regularised regression method that selects a sparse subset of predictors in least squares estimation. That is, the Lasso minimises the following objective function:

$$\frac{1}{2}\left\|y - X\beta\right\|^2 + \lambda_1 \sum_{i=1}^{p} |\beta_i|$$

where $y \in \mathbb{R}^{n \times 1}$ is the response vector, $X \in \mathbb{R}^{n \times p}$ is the matrix of predictors and $\beta \in \mathbb{R}^{p \times 1}$ is a vector of coefficients. The Fused Lasso extends the Lasso under the assumption that there is a natural ordering to the coefficients β. The Fused Lasso is able to do so by adding an additional penalty to the Lasso objective function as follows:

$$\frac{1}{2}\left\|y - X\beta\right\|^2 + \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{i=2}^{n} |\beta_i - \beta_{i-1}|$$

Here only adjacent coefficients β_i and β_{i-1} are penalised but the Fused Lasso objective function can be specified to as to induce sparsity

between any subset of β. A special case of the Fused Lasso occurs when $X = I_p$. In this case the λ_2 penalty results in β being a piece-wise continuous approximation to y. We note that Eq. (16) resembles the objective function of the Fused Lasso. This can be seen by setting $y_i = (\Theta_i^j)_{kl} + (U_i^{j-1})_{kl}$ and $\beta_i = (Z_i)_{kl}$ for $i = 1, ..., T$.

E. The dual update in Step 3 guarantees dual feasibility in the $\{Z_i\}$ variables and dual feasibility in the $\{\Theta_i\}$ variables can be checked by considering $\|Z^{k+1} - Z^k\|_2^2$.

This result is taken from (Boyd et al., 2010). Consider the general (unscaled) augmented Lagrangian with arbitrary matrices A, B and c:

$$\mathcal{L}_\gamma(\Theta, Z, Y) = f(\Theta) + g(Z) + Y^T(A\Theta + BZ - c) + \gamma 2\|A\Theta + BZ - c\|_2^2$$

All solutions must satisfy the following constraints:

Primal feasibility : $A\Theta + BZ - c = 0$
Dual feasibility : $\nabla_\Theta f(\Theta) + A'Y = 0$ and $\nabla_Z g(Z) + B'Y = 0$

where dual feasibility is based on the unscaled, unaugmented Lagrangian.

The ADMM algorithm iteratively minimises Θ and Z such that at iteration $k + 1$ we obtain Z^{k+1} that minimises $\mathcal{L}_\gamma(\Theta^{k+1}, Z, Y^k)$. From this it follows that:

$$0 = \nabla_Z \mathcal{L}_\gamma(\Theta^{k+1}, Z, Y^k)$$
$$= \nabla_Z \left\{ f(\Theta^{k+1}) + g(Z^{k+1}) + Y^k(A\Theta^{k+1} + BZ^{k+1} - c) + \gamma/2 \|A\Theta^{k+1} + BZ^{k+1} - c\|_2^2 \right\}$$
$$= \nabla_Z g(Z^{k+1}) + B'Y^k + \gamma B'(A\Theta^{k+1} + BZ^{k+1} - c)$$
$$= \nabla_Z g(Z^{k+1}) + B'(Y^k + \gamma(A\Theta^{k+1} + BZ^{k+1} - c))$$

Thus it follows that by setting $Y^{k+1} = Y^k + \gamma(A\Theta^{k+1} + BZ^{k+1} - c)$ dual feasibility in the Z variable is guaranteed. Finally, after rescaling by $U = 1/\gamma Y$ and noting that in the SINGLE algorithm $A = I_n$ and $B = -I_n$ we get the update in step 3.

Now we can continue to consider criteria for confirming dual feasibility in terms of $\{\Theta\}$ variables. Since we are guaranteed dual feasibility in $\{Z\}$ variables we only need to check for dual feasibility in $\{\Theta\}$ variables. Since Θ^{k+1} minimises $\mathcal{L}_\gamma(\Theta, Z^k, Y^k)$ we have:

$$0 = \nabla_\Theta \mathcal{L}_\gamma(\Theta, Z^k, Y^k)$$
$$= \nabla_\Theta f(\Theta^{k+1}) + A'(Y^k + \gamma(A\Theta^{k+1} + BZ^k - c))$$
$$= \nabla_\Theta f(\Theta^{k+1}) + A'\left(\underbrace{Y^k + \gamma(A\Theta^{k+1} + + BZ^{k+1} - c)}_{Y^{k+1}} + \gamma(BZ^k - BZ^{k+1}) \right)$$
$$= \nabla_\Theta f(\Theta^{k+1}) + A'Y^{k+1} + \gamma A'B(Z^k - Z^{k+1})$$

Thus in order to have dual feasibility in $\{\Theta\}$ variables we require

$$\gamma A^T B(Z^{k+1} - Z^k) = \nabla_\Theta f(\Theta^{k+1}) + A'Y^{k+1}.$$

Since in our case we have that $A = I_n$ and $B = -I_n$ it follows that we can check for dual feasibility by considering $\|Z^{k+1} - Z^k\|_2^2$.

F. The computational complexity of the DCR algorithm is $\mathcal{O}((n + b)p^3)$ where b is the number of bootstrap permutations to perform
We begin by noting that the computational complexity of the Graph-ical Lasso is $\mathcal{O}(p^3)$. While it is possible to reduce the computational complexity in some special cases we do not consider this below.

Prior to outlining our proof, a brief overview of the DCR algorithm is in order. The DCR algorithm looks to estimate dynamic functional connectivity networks by segmenting data into piece-wise continuous partitions. Within each partition the network structure is assumed to be stationary, allowing for the use of a wide variety of network estimation algorithms. In the case of the DCR the Graphical Lasso is chosen. The data, $\{X_i \in \mathbb{R}^{1 \times p} : i = 1, ..., T\}$, is segmented using a greedy partitioning scheme. Here the global network is first estimated using the Graphical Lasso. The BIC of the global network is noted and subsequently used to propose change-points. The DCR algorithm then proceeds to partition the data into subsets $A_\gamma = \{X_i : i = 1, ..., \gamma\}$ and $B_\gamma = \{X_i : i = \gamma + 1, ..., T\}$ for $\gamma \in \{\Delta + 1 ..., T - \Delta\}$. Thus Δ represents the minimum number of observations between change-points.

For each of these partitions a network is estimated for A_γ and B_γ and their joint BIC is noted. This step therefore involves $\mathcal{O}(n)$ iterations of the Graphical Lasso, resulting in a computation complexity of $\mathcal{O}(np^3)$.

Subsequently, the value of γ resulting in the greatest reduction in BIC relative to the global network is proposed as a change-point. In order to check the statistical significance of the proposed change-point a block bootstrap permutation permutation test is performed. This step involves a further b iterations of the Graphical Lasso where b is the number of bootstrap permutations performed. As a result this step has a computational complexity of $\mathcal{O}(bp^3)$.

This procedure is repeated until all significant change-points have been reported. We therefore conclude that the computational complexity of the DCR algorithm is $\mathcal{O}((n + b)p^3)$.

Appendix B. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.neuroimage.2014.07.033.

References

Achard, S., Salvador, R., Whitcher, B., Suckling, J., Bullmore, E.T., 2006. A resilient, low-frequency, small-world human brain function network with highly connected association of cortical hubs. J. Neurosci. 26 (1), 63–72.

Allen, E., Damaraju, E., Plis, S., Erhardt, E., Eichele, T., Calhoun, V., 2012. Tracking whole-brain connectivity dynamics in the resting state. Cereb. Cortex. http://dx.doi.org/10.1093/cercor/bhs352.

Aron, A.R., Fletcher, P.C., Bullmore, E.T., Sahakain, B.J., Robbins, T.W., 2003. Stop-signal inhibition disrupted by damage to right inferior frontal gyrus in humans. Nat. Neurosci. 6, 115–116.

Barabási, A., Albert, R., 1999. Emergence of scaling in random networks. Science 286 (5439), 509–512.

Bassett, D., Bullmore, E., 2006. Small-world brain networks. Neuroscientist 12 (6), 512–523.

Bassett, D.S., Wymbs, N.F., Porter, M.A., Mucha, P.J., Carlson, J.M., Grafton, S.T., 2010. Dynamic reconfiguration of human brain networks during learning. Proc. Natl. Acad. Sci. U. S. A. 108, 7641–7646.

Bertsekas, D.P., 1999. Nonlinear Programming. Athena Scientific.

Bonnelle, V., Ham, T.E., Leech, R., Kinnunen, K.M., Mehta, M.A., Greenwood, R.J., Sharp, D.J., 2012. Salience network integrity predicts default mode network function after traumatic brain injury. Proc. Natl. Acad. Sci. 109 (12), 4690–4695.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Bullmore, E., Sporns, O., 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. Nature 10, 186–198.

Calhoun, V.D., Liu, J., Adalı, T., 2009. A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. NeuroImage 45 (1), S163–S172.

Chang, C., Glover, G.H., 2010. Time frequency dynamics of resting-state brain connectivity measured with fMRI. NeuroImage 50 (1), 81–98.

Corbetta, M., Shulman, G.L., 2002. Control of goal-directed and stimulus-driven attention in the brain. Nat. Rev. Neurosci. 3 (3), 201–215.

Cribben, I., Haraldsdottir, R., Atlas, Y.L., Wager, T.D., Lindquist, M.A., 2012. Dynamic connectivity regression: determining state-related changes in brain connectivity. NeuroImage 61 (4), 907–920.

Danaher, P., Wang, P., Witten, D., 2013. The joint graphical lasso for inverse covariance estimation across multiple classes. J. R. Stat. Soc. Ser. B (Stat Methodol.) 76 (2), 373–397.

Duchi, J., Singer, Y., 2009. Efficient online and batch learning using forward backward splitting. J. Mach. Learn. Res. 10, 2899–2934.

Eguiluz, V.M., Chialvo, D.R., Cecchi, G.A., Baliki, M., Apkarian, A.V., 2005. Scale-free brain functional networks. Phys. Rev. Lett. 94 (1), 018102.

Ekman, M., Derrfuss, J., Tittgemeyer, M., Fiebach, C., 2012. Predicting errors from reconfiguration patterns in human brain networks. Proc. Natl. Acad. Sci. 109 (41), 16714–16719.

Erdos, P., Renyi, A., 1959. On Random Graphs. Publicationes Mathematicae Debrecen.

Esposito, F., Seifritz, E., Formisano, E., Morrone, R., Scarabino, T., Tedeschi, G., Cirillo, S., Goebel, R., Di Salle, F., 2003. Real-time independent component analysis of fMRI time-series. NeuroImage 20 (4), 2209–2224.

Esposito, F., Bertolino, A., Scarabino, T., Latorre, V., Blasi, G., Popolizio, T., Tedeschi, G., Cirillo, S., Goebel, R., Di Salle, F., 2006. Independent component model of the default-mode brain function: assessing the impact of active thinking. Brain Res. Bull. 70 (4), 263–269.

Fair, D.A., Cohen, A.L., Power, J.D., Dosenbach, N.U., Church, J.A., Meizin, F.M., Schlaggar, B.L., Petersen, S.E., 2009. Functional brain networks develop from a local to distributed organization. PLoS 5 (5).

Fornito, A., Harrison, B.J., Zalesky, A., Simons, J.S., 2012. Competitive and cooperative dynamics of large-scale brain functional networks supporting recollection. Proc. Natl. Acad. Sci. 109 (31), 12788–12793.

Fornito, A., Zalesky, A., Breakspear, M., 2013. Graph analysis of the human connectome: promise, progress, and pitfalls. NeuroImage 80, 426–444.

Fransson, P., 2006. How default is the default mode of brain function? Further evidence from intrinsic BOLD signal fluctuations. Neuropsychologia 44 (14), 2836–2845.

Friedman, T., Hastie, T., Tibshirani, R., 2008. Sparse inverse covariance estimation via the Graphical Lasso. Biostatistics 9 (3), 432–441.

Friston, K.J., 2011. Function and effective connectivity: a review. Brain Connect. 1 (1), 13–36.

Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C.J., Wedeen, V.J., Sporns, O., 2008. Mapping the structural core of human cerebral cortex. PLoS Biol. 6 (7), e159.

Hampshire, A., Chamberlain, S.R., Monti, M.M., Duncan, J., Owen, A.M., 2010. The role of the right inferior frontal gyrus: inhibition and attentional control. NeuroImage 50 (3), 1313–1319.

Handwerker, D.A., Roopchansingh, V., Gonzalez-Castillo, J., Bandettini, P.A., 2012. Periodic changes in fMRI connectivity. NeuroImage 63, 1712–1719.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. vol. 2. Springer.

Haykin, S., 2002. Adaptive Filter Theory. Prentice Hall Information and System Science Series.

Hellyer, P.J., Shanahan, M., Scott, G., Wise, R.J.S., Sharp, D.J., Leech, R., 2014. The control of global brain dynamics: opposing actions of frontoparietal control and default mode networks on attention. J. Neurosci. 34 (2), 451–461.

Hinne, M., Ambrogioni, L., Janssen, R.J., Heskes, T., van Gerven, M., 2013. Structurally-informed Bayesian functional connectivity analysis. NeuroImage 86, 294–305.

Hoefling, H., 2010. A path algorithm for the Fused Lasso Signal Approximator. J. Comput. Graph. Stat. 19 (4), 984–1006.

Huang, S., Sun, L., Ye, J., Fleisher, A., Wu, T., Chen, K., Reiman, E., 2010. Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. NeuroImage 50 (3), 935–949.

Husain, M., Nachev, P., 2007. Space and the parietal cortex. Trends Cogn. Sci. 11 (1), 30–36.

Hutchison, M.R., Womelsdoft, T., Gat, J.S., Everling, S., Menon, R.S., 2012. Resting-state networks show dynamic functional connectivity in awake humans and anesthetized macaques. Hum. Brain Mapp. 34, 2154–2177.

Hutchinson, M.R., Womelsdoft, T., Allen, P.A., Bandettini, E.A., Calhoun, V.D., Corbetta, M., Penna, S.D., Duyn, J., Glover, G., Gonzalez-Castillo, J., Handwerker, D.A., Keilholz, S., Kiviniemi, V., Leopold, D.A., de Pasquale, F., Sporns, O., Walterm, M., Chang, C., 2013. Dynamic functional connectivity: promise, issues, and interpretations. NeuroImage 80, 360–378.

Kim, S.-J., Koh, K., Boyd, S., Gorinevsky, D., 2009. l1 trend filtering. SIAM Rev. 51 (2), 339–360.

Leonardi, N., Richiardi, J., Gschwind, M., Simioni, S., Annoni, J.-M., Schluep, M., Vuilleumier, P., Van De Ville, D., 2013. Principal components of functional connectivity: a new approach to study dynamic brain connectivity during rest. NeuroImage 83, 937–950.

Lindquist, M., 2008. The statistical analysis of fMRI data. Stat. Sci. 23 (4), 439–464.

Liu, P., Zhang, Y., Yuan, K., Qin, W., Zhou, L., Liang, J., Chen, P., Dai, J., Liu, Y., Tian, J., 2009. Partial correlation investigation on the default mode network involved in acupuncture: an fMRI study. Neurosci. Lett. 462, 183–187.

Lowe, M.J., Mock, B.J., Sorenson, J.A., 1998. Functional connectivity in single and multislice echoplanar imaging using resting-state fluctuations. NeuroImage 7, 119–132.

Markov, N.T., Ercsey-Ravasz, M., Van Essen, D.C., Knoblauch, K., Toroczkai, Z., Kennedy, H., 2013. Cortical high-density counterstream architectures. Science 342 (6158).

Marrelec, G., Krainik, A., Duffau, H., Pelegrini-Issac, M., Lehericy, S., Doyon, J., Benali, H., 2006. Partial correlation for functional brain interactivity investigation in functional. NeuroImage 32, 228–237.

Marrelec, G., Kim, J., Doyon, J., Horwitz, B., 2009. Large-scale neural model validation of partial correlation analysis for effective connectivity investigation in functional MRI. Hum. Brain Mapp. 30 (3), 941–950.

Mattingley, J.B., Husain, M., Rorden, C., Kennard, C., Driver, J., 1998. Motor role of human inferior parietal lobe revealed in unilateral neglect patients. Nature 392 (6672), 179–182.

Monti, R.P., Montana, G., Anagnostopoulos, C., 2014. SINGLE: Estimate sparse dynamic graphs using the Smooth Incremental Graphical Lasso Estimation (SINGLE) algorithm. URL http://CRAN.R-project.org/package=SINGLE (R package version 1.2).

Newman, M.E.J., 2003. The structure and function of complex networks. SIAM Rev. 45 (2), 167–256.

Nichols, T., Hayasaka, S., 2003. Controlling the familywise error rate in functional neuroimaging: a comparative review. Stat. Methods Med. Res. 12 (5), 419–446.

Pandit, A.S., Expert, P., Lambiotte, R., Bonnelle, V., Leech, R., Turkheimer, F.E., Sharp, D.J., 2013. Traumatic brain injury impairs small-world topology. Neurology 80 (20), 1826–1833.

Peng, J., Wang, P., Zhou, N., Zhu, J., 2009. Partial correlation estimation by joint sparse regression models. J. Am. Stat. Assoc. 104 (486), 735–746.

Poldrack, R.A., Mumford, J.A., Nichols, T.E., 2011. Handbook of Functional MRI Data Analysis. Cambridge University Press.

Príncipe, J.C., Liu, W., Haykin, S., 2011. Kernel Adaptive Filtering: A Comprehensive Introduction. vol. 57. John Wiley & Sons.

Sakoglu, U., Pearlson, G.D., Kiehl, K.A., Wang, Y.M., Micheal, A.M., Calhoun, V.D., 2010. A method for evaluating dynamic functional network connectivity and task modulation: application to Schizophrenia. MAGMA 23, 351–366.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. vol. 26. CRC Press.

Smith, S.M., Jenkinson, M., Woolrich, M.W., Beckmann, C.F., Behrens, T.E.J., Johansen-Berg, H., Bannister, P.R., De Luca, M., Drobnjak, I., Flitney, D.E., Niazy, R.K., Saunders, J., Vickers, J., Zhang, Y., De Stefano, N., Brady, M.J., Matthews, P.M., 2004. Advances in functional and structural MR image analysis and implementation as FSL. NeuroImage 23 (Suppl. 1(0)), S208–S219.

Smith, S.M., Miller, K.L., Gholamrez, S.-K., Webster, M., Beckmann, C.F., Nichols, T.E., Ramsey, J.D., Woolrich, M.W., 2011. Network modelling methods for fMRI. NeuroImage 54 (2), 875–891.

Sporns, O., Chialvo, D.R., Kaiser, M., Hilgetag, C.C., 2004. Organization, development and function of complex brain networks. Trends Cogn. Sci. 8 (9), 418–425.

Stephan, K.E., Hilgetag, C., Burns, G., O'Neill, M.A., Young, M.P., Kotter, R., 2000. Computational analysis of functional connectivity between areas of primate cerebral cortex. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. 3550 (1393), 111–126.

Strother, S.C., Anderson, J.R., Schaper, K.A., Sidtis, J.J., Liow, J.-S., Woods, R.P., Rottenberg, D.A., 1995. Principal component analysis and the scaled subprofile model compared to intersubject averaging and statistical parametric mapping: I. "functional connectivity" of the human motor system studied with [^{15}O] water PET. J. Cereb. Blood Flow Metab. 15, 738–753.

Sun, F.T., Miller, L.M., D'Eposito, M., 2004. Measuring interregional functional connectivity using coherence and partial coherence analyses of fMRI data. NeuroImage 21, 647–658.

Sun, F.T., Miller, L.M., Rao, A.A., D'Esposito, M., 2007. Functional connectivity of cortical networks involved in bimanual motor sequence learning. Cereb. Cortex 17 (5), 1227–1234.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K., 2005. Sparsity and smoothness via the fused lasso. J. R. Stat. Soc. Ser. B 67 (1), 91–108.

van der Heuvel, M.P., Hulshoff Pol, H.E., 2010. Exploring the brain network: a review on resting-state fMRI functional connectivity. Eur. Neuropsychopharmacol. 20 (8), 519–534.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of small-world networks. Nature 393 (6684), 440–442.

Whittaker, J., 1990. Graphical Models in Applied Multivariate Statistics. John Wiley and Sons.

Xiao, L., 2010. Dual averaging methods for regularized stochastic learning and online optimization. J. Mach. Learn. Res. 11 (2543–2596), 4.

Zhou, S., Lafferty, J., Wasserman, L., 2010. Time varying undirected graphs. Mach. Learn. 80, 295–319.

Zou, H., 2006. The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. 101 (476), 1418–1429.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the Elastic Net. J. R. Stat. Soc. Ser. B (Stat Methodol.) 67 (2), 301–320.