Shared and Specific Independent Components Analysis for Between-Group Comparison

Shahabeddin Vahdat
shahabeddin.vahdat@mail.mcgill.ca
Department of Kinesiology and PE, McGill University,
Montreal, QC H2W 1S4, Canada

Mona Maneshi
mona.maneshi@mail.mcgill.ca
Christophe Grova
christophe.grova@mcgill.ca
Biomedical Engineering Department, McGill University,
Montreal, QC H2W 1S4, Canada, and Montreal Neurological
Institute and Hospital, Montreal, QC H3A 2B4, Canada

Jean Gotman
jean.gotman@mcgill.ca
Montreal Neurological Institute and Hospital, Montreal,
QC H3A 2B4, Canada

Theodore E. Milner
theodore.milner@mcgill.ca
Department of Kinesiology and PE, McGill University,
Montreal, QC H2W 1S4, Canada

Independent component analysis (ICA) has been extensively used in individual and within-group data sets in real-world applications, but how can it be employed in a between-groups or conditions design? Here, we propose a new method to embed group membership information into the FastICA algorithm so as to extract components that are either shared between groups or specific to one or a subset of groups. The proposed algorithm is designed to automatically extract the pattern of differences between different experimental groups or conditions. A new constraint is added to the FastICA algorithm to simultaneously deal with the data of multiple groups in a single ICA run. This cost function restricts the specific components of one group to be orthogonal to the subspace spanned by the data of the other groups. As a result of performing a single ICA on the aggregate data of several experimental groups, the entire variability of data sets is used to extract the shared components. The results of simulations show that the proposed algorithm performs

better than the regular method in both the reconstruction of the source signals and classification of shared and specific components. Also, the sensitivity to detect variations in the amplitude of shared components across groups is enhanced. A rigorous proof of convergence is provided for the proposed iterative algorithm. Thus, this algorithm is guaranteed to extract and classify shared and specific independent components across different experimental groups and conditions in a systematic way.

1 Introduction

In many biological data analysis applications, the goal is to find the similarities and differences between various experimental groups (or conditions) based on some observed high-dimensional data sets. If the information regarding the generative model of the data is inaccurate or lacking, data-driven approaches are preferred to explore the main features of the data based on some basic assumptions about the data set (Bartlett, 2001). As one of these data-driven approaches, independent component analysis (ICA) uses the often biologically fitting assumption that the underlying sources (networks, components, or synergies) that comprise the data set of each group are independent (Stone, 2002). ICA has been extensively used in analysis of functional magnetic resonance imaging (fMRI) (McKeown, Makeig, et al., 1998), electroencephalographic (EEG) (Makeig, Jung, Bell, Ghahremani, & Sejnowski, 1997), and magnetoencephalographic (MEG) data (Vigario, Sarela, Jousmaki, Hamalainen, & Oja, 2000), as well as electromyographic data (EMG) (Tresch, Cheung, & d'Avella, 2006).

In the context of fMRI, the data are blood oxygenation level–dependent (BOLD) signals recorded at different times and in different voxels. Usually ICA is used to decompose multivariate fMRI data into a sum of spatially independent and temporally fixed components (spatial-ICA) (Calhoun, Liu, & Adali, 2009; McKeown, Makeig, et al., 1998). Using multichannel EMG signals, ICA is used to extract different patterns of muscle synergies (sets of muscles that work in synchrony) in a variety of motor tasks (Kargo & Nitz, 2003; Tresch et al., 2006). In these cases, ICA is primarily used on either individual data or in single-group-based analyses (Calhoun, Adali, Pearlson, & Pekar, 2001; Calhoun, Liu, & Adali, 2009; McKeown, Jung, et al., 1998). There is little in the literature on the application of ICA to identify between-group differences (Albert, Robertson, & Miall, 2009; Calhoun et al., 2009; Li, Guo, Nie, Li, & Liu, 2009). This is in part because of the difficulty in interpreting the results of ICA when used in a multigroup or factorial design. In multigroup designs, usually a separate ICA run is performed on the concatenated data of each group (Calhoun et al., 2001). The problem in treating the individual experimental group (or session) separately is that a component may be present in one group and absent in the other for a variety of reasons. This may be caused by an insufficient number of extracted components or the missing component being split into subcomponents, or it may represent a real difference between the two groups (we refer to this as a specific component). This ambiguity in the classification and detection of components attributed to one or more groups represents a computational challenge.

Another drawback of performing a distinct ICA on each group's data set is that the variability of the whole data set is not used to extract the components that are common between groups (we label these components as shared components). This reduces the reconstruction performance in extracting shared components (reconstruction error is defined as the root mean square error, (RMSE, between the actual and extracted independent components). A further important consideration is the need for sufficient sensitivity to detect changes in the amplitude of shared components between groups. A component's amplitude can be defined as the percentage of variability that that component can explain in each group. If there are even slight variations between different realizations of a shared component extracted from separate within-group ICA runs, it may cause a significant false difference in that component's amplitude across groups.

In the context of fMRI analysis, several methods have been proposed to increase the sensitivity of detecting group differences using ICA (Sui, Adali, Pearlson, & Calhoun, 2009; Sui, Adali, Pearlson, Clark, & Calhoun, 2009) or local linear discriminant analysis (McKeown et al., 2007; Palmer, Li, Wang, & McKeown, 2010). These methods in general require extracting a set of features from data sets and then maximizing the separability of the extracted features between groups. It limits their application because they cannot be applied directly to raw time series. This limitation is highlighted in cases such as fMRI resting-state analysis where a fixed model of task timing across subjects is not available and feature extraction in time domain is challenging.

Here, we propose a new algorithm to address the limitations of the regular ICA approach in the problem of multigroup and factorial design. We based our method on the FastICA algorithm proposed by Hyvarinen and Oja (1997). FastICA is an efficient and fast-converging ICA algorithm that has been extensively used in different fields of biomedical signal analysis (Hyvarinen & Oja, 2000). The simple iterative formulation of the FastICA algorithm allowed us to incorporate an additional constraint into its cost function to integrate the group membership information. The proposed algorithm aims at adaptively extracting the shared and specific components at each iteration while performing a single ICA on the aggregate data of all groups. The adaptive extraction of the shared and specific components in the proposed algorithm allows the investigation of the intergroup differences in a systematic way.

In section 2.2, after briefly describing the FastICA algorithm, we define the concepts of shared and specific components in the case of an ICA linear generative model. The shared and specific component decomposition has been applied to the nonnegative matrix factorization algorithm, a principal components analysis (PCA)–based approach to decompose data into two matrices with nonnegativity constraints (Cheung, d′Avella, Tresch, & Bizzi, 2005). Here, the definition of shared and specific components has been modified based on the independency constraints. Accordingly, we propose a converging algorithm based on the FastICA method to incorporate the group membership information. We focus on the special case of having only two groups, but the extension to more than two groups or factorial design is straightforward, as outlined in section 2.7. The performance of the proposed method has been validated using different simulations. The results of simulations on a one-dimensional and an fMRI-like two-dimensional data set are reported in section 3. Also, the results of applying the proposed algorithm to a hybrid resting-state fMRI data set are compared with one of the published methods for conducting between-groups comparison using ICA. In section 4, we present our conclusions and discuss how the proposed algorithm can be integrated with some state-of-the-art methods that can be applied to the between-group ICA problem in the context of fMRI data analysis.

Generally, lowercase letters in bold italics indicate vectors and capital letters in italics denote matrices. All vectors are understood as column vectors; thus, $\boldsymbol{x}^T$, or the transpose of $\boldsymbol{x}$, is a row vector. E{.} indicates the expectation operator.

2 Methods

In linear ICA, it is assumed that an observed T-dimensional zero-mean vector of random variables y is generated according to

$$\boldsymbol{y} = A\boldsymbol{s}, \tag{2.1}$$

where $\boldsymbol{s}$ is an N-dimensional random vector (N < T), whose components are assumed mutually independent (the sources), and A is a nonsingular T × N mixing matrix to be estimated. If we let S be a matrix where each column of S is one of M samples from the random vector $\boldsymbol{s}$, then we can write equation 2.1 in matrix format: $Y= AS$, where Y is the matrix of observed data whose columns are the corresponding M samples from $\boldsymbol{y}$. A common and useful preprocessing step in ICA is to whiten the observed data using a T × T whitening matrix V (Hyvarinen & Oja, 2000). By definition, a vector is whitened (or sphered) when its components are uncorrelated and their variances equal unity. For a zero-mean vector of random variables $\boldsymbol{x}$, it implies that the covariance matrix of $\boldsymbol{x}$ equals the identity matrix: $C_x = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\} = I$. If E is the orthogonal matrix of eigenvectors and D is the diagonal matrix of eigenvalues of $C_y = \mathrm{E}\{\boldsymbol{y}\boldsymbol{y}^T\} = EDE^T$, then a suitable whitening matrix is $V = C_y^{-\frac{1}{2}} = ED^{-\frac{1}{2}}E^T$. At the same time, we can reduce dimensions of data to match the number of extracted independent components using the same technique as in principal component analysis (PCA). This can be done by discarding the (T − N) smallest eigenvalues of C_y during calculation of the whitening matrix to get an N × T transformation matrix Z: $\boldsymbol{x} = Z\boldsymbol{y}$, where $C_x = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\} = I_N$ (Hyvarinen & Oja, 2000). Multiplying both sides of equation 2.1 by Z yields $\boldsymbol{x} = ZA\boldsymbol{s} = \tilde{A}\boldsymbol{s}$, where square matrix $\tilde{A}$ can be considered the new mixing matrix of reduced dimension. So the task in ICA would be to find an unmixing matrix $W = \tilde{A}^{-1}$, given the whitened data. Then the sources can be estimated as follows:

$$\hat{\boldsymbol{s}} = W\boldsymbol{x}. \tag{2.2}$$

Typically we assume that the sources have unit variance, with any scaling factor being absorbed into the mixing matrix. So it easy to show that matrix $\tilde{A}$ is an orthonormal matrix. This implies that matrix W is also an orthonormal matrix and $W^T = \tilde{A}$.

2.1 FastICA Algorithm. FastICA is based on the maximization of negentropy, which can also be reformulated as minimization of the mutual information of the extracted sources $\hat{\boldsymbol{s}}_i$, constraining them to be uncorrelated, from an information-theoretic view of independence (Hyvarinen, 1999). The classic FastICA algorithm uses an approximation of negentropy as an objective function in order to find maximally nongaussian projections $\boldsymbol{w}^T\boldsymbol{x}$, where $\boldsymbol{w}^T$ is one of the rows of the unmixing matrix, $W^T[\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N]$. This minimization is simplified to find the extrema of a generic cost function $\mathrm{E}\{\mathrm{G}(\boldsymbol{w}^T\boldsymbol{x})\}$ under the constraint $\mathrm{E}\{(\boldsymbol{w}^T\boldsymbol{x})^2\} = \|\boldsymbol{w}\|^2 = 1$ (Hyvarinen, 1999). The constraint results from the fact that the sources have unit variance and the data are whitened. FastICA updates $\boldsymbol{w}$ according to the following approximate Newton iteration (also called a fixed-point algorithm):

$$\begin{aligned} \boldsymbol{w}^+ &= \mathrm{E}\{\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x})\} - \mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\}\boldsymbol{w}, \\ \boldsymbol{w}^* &= \boldsymbol{w}^+/\|\boldsymbol{w}^+\|, \end{aligned} \tag{2.3}$$

where g is the derivative of G (any nonquadratic function) and g' is the derivative of g. The normalization has been added to improve convergence stability (Hyvarinen, 1999). In order to avoid the convergence of different independent components to the same solution, after each iteration of the algorithm, the vectors $\boldsymbol{w}$ are orthogonalized in either a deflation mode or symmetrically (Hyvarinen, 1999). A simpler alternative to the symmetric decorrelation approach is the following iterative algorithm (Hyvarinen, 1999):

$$\begin{cases} 1.\ \text{Let}\, W = W/\sqrt{\|WW^T\|}. \\ \text{Repeat}\, 2.\ \text{until convergence:} \\ 2.\ \text{Let}\, W = \dfrac{3}{2}W - \dfrac{1}{2}WW^T W. \end{cases} \tag{2.4}$$

The fixed-point algorithm typically converges very fast (convergence is proven to be quadratic) to an unmixing matrix W, and the order and sign of components in the estimated source vector $\hat{s}$ may be different from those in the source vector $\boldsymbol{s}$.

2.2 Proposed Decomposition into Shared and Specific Components. Let X_1 denote the whitened concatenated observed data matrix of group 1, and X_2 denote the whitened concatenated data from group 2. The concatenation is done by stacking the two-dimensional data matrices of every individual in each group (see section 2.3). Here, group 1 and group 2 may represent data from two groups of individuals or the same individuals under two experimental conditions. Assume that X_1 can be reconstructed by N_{g1} and X_2 by N_{g2} independent components (ICs) using the generative model in equation 2.1: $X_j = A_j.S_j,\ j = 1, 2$ Assuming K shared ICs between groups ($\boldsymbol{s}_i^{sh},\ i = 1, .., \text{K}$), we can decompose each generative model into two parts: ICs shared between the two groups ($\boldsymbol{s}_i^{sh}$) and ICs specific to each group ($\boldsymbol{s}_i^{sp,1}$ or $\boldsymbol{s}_i^{sp,2}$):

$$X_1 = \sum_{i=1}^{K} \boldsymbol{a}_i^{sh,1}\left(\boldsymbol{s}_i^{sh}\right)^T + \sum_{i=1}^{K_1} \boldsymbol{a}_i^{sp,1}\left(\boldsymbol{s}_i^{sp,1}\right)^T,$$

$$X_2 = \sum_{i=1}^{K} \boldsymbol{a}_i^{sh,2}\left(\boldsymbol{s}_i^{sh}\right)^T + \sum_{i=1}^{K_2} \boldsymbol{a}_i^{sp,2}\left(\boldsymbol{s}_i^{sp,2}\right)^T, \tag{2.5}$$

where $\boldsymbol{s}^{sh}$ and $\boldsymbol{s}^{sp,j}$ are columns of S_j^T, $j = 1, 2$ arranged according to the shared or specific label, and $\boldsymbol{a}^{sh,j}$, $\boldsymbol{a}^{sp,j}$ are the corresponding columns of A_j. Also, $K_1 = N_{g1} - K$ is the number of specific components of group 1, while $K_2 = N_{g2} - K$ is the number of specific components for group 2. Since, by definition, all the dependencies across groups are factored into the shared components, the specific ICs of two groups are assumed to be mutually independent. Also note that the mixing matrix signals related to group 1 and group 2 corresponding to the same shared component can be different. The above equations can be summarized in the following matrix format:

$$X_j = A_j^{sh} S^{sh} + A_j^{sp} A_j^{sp}, \quad j = 1, 2. \tag{2.6}$$

Here $\boldsymbol{a}^{sh,j}$ refers to the columns of A_j^{sh}, $\boldsymbol{a}^{sp,j}$ to the columns A_j^{sp}, $\boldsymbol{s}^{sh}$ to the columns of $(S^{sh})^T$, and $\boldsymbol{s}^{sp,j}$ to the columns of $(S_j^{sp})^T$. If we combine the data from two groups in a single concatenated matrix, using equation 2.6, we can write:

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} A_1^{sh} & A_1^{sp} & 0 \\ A_2^{sh} & 0 & A_2^{sp} \end{bmatrix} \begin{bmatrix} S^{sh} \\ S_1^{sp} \\ S_2^{sp} \end{bmatrix}. \tag{2.7}$$

Since all the rows in S^{sh}, S_1^{sp}, and S_2^{sp} are mutually independent, this equation gives the independent component factorization for the concatenated matrix, with the additional constraint that two blocks of the new mixing matrix should be zero. In sections 2.4 and 2.5, we will show mathematically how this new constraint can be integrated with the FastICA method. We start with some necessary preprocessing steps for applying the group-level ICA to real data.

2.3 Three-Level Dimension Reduction. Let us say we have n_1 subjects in group 1 and n_2 subjects in group 2. The first step is to subtract the sample mean of the observed variables. Let Y_i^j denote the $\mathrm{T}_j^i \times \mathrm{M}$ zero-mean data of subject $i$of group j. To reduce computational burden as well as noise, it is common to reduce the dimension of the data matrix using PCA, at both subject- and group-level analysis, specifically for huge data sets like fMRI signal time series. The first-level PCA reduces the dimensions of the data matrix Y_i^j of each subject in group j from T_j^i to T_j, using the appropriate transformation matrix Z_i^j as explained above: $X_i^j = Z_i^j Y_i^j$. To perform the group-level ICA, a common approach is to concatenate the data of different subjects within the same group to get a single data matrix corresponding to each group (Calhoun et al., 2001):

$$\tilde{X}_1 = \begin{bmatrix} X_1^1 \\ \cdot \\ \cdot \\ \cdot \\ X_{n_1}^1 \end{bmatrix}; \quad \tilde{X}_2 = \begin{bmatrix} X_1^2 \\ \cdot \\ \cdot \\ \cdot \\ X_{n_2}^2 \end{bmatrix}.$$

We can then repeat the process by selecting an appropriate transformation matrix to whiten the data of each group and reduce its dimension to the desired number of components—N_{g1} for group 1 and N_{g2} for group 2. This constitutes the middle-level PCA model: $X_j = Z_j \tilde{X}_j$. Here, Z_j is an $N_{g1} \times n_1 T_1$ whitening matrix for group 1 and an $N_{g2} \times n_2 T_2$ whitening matrix for group 2. As discussed earlier, we propose to concatenate the data of two groups into a single aggregate matrix. Then we can perform a third-level

PCA-like data reduction to reduce the dimension of the aggregate data further to N and at the same time whiten the data:

$$X = Z \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \to \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = pinv(Z)X \triangleq HX, \tag{2.8}$$

where Z is an $N \times (N_{g1} + N_{g2})$ whitening matrix and H is the pseudo-inverse of matrix Z, or dewhitening matrix (note that $N < (N_{g1} + N_{g2})$).

2.4 New Constraints for Between-Group ICA Based on Shared-Specific Decomposition. One can construct an ICA model for the whitened aggregate matrix X and use equation 2.8 to get

$$X = AS \to \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = HAS. \tag{2.9}$$

Here HA can be viewed as the new mixing matrix for the aggregate data of two groups. Comparing equations 2.7 and 2.9 results in

$$(HA)S = \begin{bmatrix} A_1^{sh} & A_1^{sp} & 0 \\ A_2^{sh} & 0 & A_2^{sp} \end{bmatrix} \begin{bmatrix} S^{sh} \\ S_1^{sp} \\ S_2^{sp} \end{bmatrix}. \tag{2.10}$$

So to achieve the same separation of shared and specific components for matrix S as in the source matrix of the right-hand side of the equation, the two mixing matrices should be identical. This forces two blocks of matrix HA to be zero. Since there is a correspondence between the columns of the mixing matrix and the rows of the source matrix, equation 2.10 constrains the last N_{g2} elements of H multiplied by the columns corresponding to specific components of group 1 and the first N_{g1} elements of H multiplied by the columns corresponding to specific components of group 2 to be zero. If we define $P_1 = [\, I_{N_{g1}} \;\; 0_{N_{g1} \times N_{g2}} \,]$ and $P_2 = [\, 0_{N_{g2} \times N_{g1}} \;\; I_{N_{g2}} \,]$, one can formulate the constraints as follows:

$$\begin{cases} P_1 \, H \, \boldsymbol{a}_i = [0]_{N_{g1} \times 1}, & for\ \boldsymbol{s}_i \in SP2 \\ P_2 \, H \, \boldsymbol{a}_i = [0]_{N_{g2} \times 1}, & for\ \boldsymbol{s}_i \in SP1 \end{cases}, \tag{2.11}$$

where $\boldsymbol{a}_i$ represents a column of matrix A and $\boldsymbol{s}_i^T$ the corresponding row in matrix S. Also $SP1$ specifies the subset of specific components of group 1, and $SP2$ specifies the specific components of group 2. If we split matrix H into an upper part H_1 with N_{g1} rows and a lower part H_2 with N_{g2} rows, we can write

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \rightarrow H_1 = P_1 H, \quad H_2 = P_2 H. \tag{2.12}$$

As we mentioned earlier, if the data are sphered, then $W^T = A$. So equation 2.11 is simplified as

$$\begin{cases} H_1 \boldsymbol{w}_i = [0]_{N_{g1} \times 1}, & \mathit{for}\ \boldsymbol{w}_i^T \boldsymbol{x} \in SP2 \\ H_2 \boldsymbol{w}_i = [0]_{N_{g2} \times 1}, & \mathit{for}\ \boldsymbol{w}_i^T \boldsymbol{x} \in SP1 \end{cases}, \tag{2.13}$$

where $\boldsymbol{w}_i$ are columns of matrix W^T.

2.5 A Converging Algorithm Based on FastICA Solution. To extract the specific components of group 1, we use the Lagrange multipliers method (Lang, 1987) to define a new cost function C_{new} (the so-called Lagrangian) based on the FastICA algorithm (Hyvarinen, 1999). The goal is to minimize $\mathrm{E}\{G(\boldsymbol{w}^T \boldsymbol{x})\}$, under the constraints $\|\boldsymbol{w}\|^2 = 1$, and $H_2 \boldsymbol{w} = \mathbf{0}$:

$$C_{new} = C_{FastICA} + \boldsymbol{\lambda}^T H_2 \boldsymbol{w} = \mathrm{E}\{G(\boldsymbol{w}^T \boldsymbol{x})\} + \lambda_1 (\|\boldsymbol{w}\|^2 - 1) + \boldsymbol{\lambda}^T H_2 \boldsymbol{w},$$

where the scalar λ_1 and vector $\boldsymbol{\lambda}$ are Lagrange multipliers, the latter handling N_{g2} constraints. Based on Lagrange multipliers method, the partial derivative of the Lagrangian with respect to $\boldsymbol{w}$, denoted by $F(\boldsymbol{w})$, is to be zero (let $\beta \triangleq -2\lambda_1$):

$$F(\boldsymbol{w}) = \frac{\partial C_{new}}{\partial \boldsymbol{w}} = \mathrm{E}\{\boldsymbol{x} g(\boldsymbol{w}^T \boldsymbol{x})\} - \beta \boldsymbol{w} + H_2^T \boldsymbol{\lambda} = 0. \tag{2.14}$$

To find the vector $\boldsymbol{\lambda}$, we multiply both sides of equation 2.14 by H_2:

$$H_2 \mathrm{E}\{\boldsymbol{x} g(\boldsymbol{w}^T \boldsymbol{x})\} - \beta H_2 \boldsymbol{w} + H_2 H_2^T \boldsymbol{\lambda} = 0.$$

By substituting the above equation at $\boldsymbol{w}_0$, the value of $\boldsymbol{w}$ at optimum, $\boldsymbol{\lambda}$ is determined:

$$\boldsymbol{\lambda} = -\left(H_2 H_2^T\right)^{-1} H_2 \mathrm{E}\{\boldsymbol{x} g(\boldsymbol{w}_0^T \boldsymbol{x})\}. \tag{2.15}$$

Note that $H_2 \boldsymbol{w}_0 = 0$, and $H_2 H_2^T$ is a full rank $N_{g2} \times N_{g2}$ matrix (see appendix A for the proof). Also after multiplying both sides of equation 2.14 by $\boldsymbol{w}_0^T$ and performing simple algebraic operations, one can find:

$$\beta = \mathrm{E}\left\{\boldsymbol{w}_0^T \boldsymbol{x} g(\boldsymbol{w}_0^T \boldsymbol{x})\right\}. \tag{2.16}$$

Note that the formula to find β is the same as the one found in the original FastICA algorithm (Hyvarinen, 1999). As in the FastICA approach, let us try to solve equation 2.14 by Newton's method. We obtain the Jacobian of the function $F(\boldsymbol{w})$, denoted by $JF(\boldsymbol{w})$, as follows:

$$JF(\boldsymbol{w}) = \mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T g'(\boldsymbol{w}^T\boldsymbol{x})\} - \beta I. \tag{2.17}$$

Since in equation 2.14, $H_2^T\boldsymbol{\lambda}$ is constant with regard to $\boldsymbol{w}$, the Jacobian function is exactly the same as the one in the original FastICA algorithm. So we use the same estimate as proposed by Hyvarinen (1999) to approximate $\mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T g'(\boldsymbol{w}^T\boldsymbol{x})\}$ by $\mathrm{E}\{\boldsymbol{x}\boldsymbol{x}^T\}\mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\} = \mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\}I$ (note that the data are sphered). This makes the Jacobian matrix diagonal, which can be easily inverted. Again using trick similar to that proposed by Hyvarinen (1999), we approximate β and $\boldsymbol{\lambda}$ using the current value of $\boldsymbol{w}$ instead of $\boldsymbol{w}_0$. This gives the following approximate Newton iteration:

$$\begin{aligned}\boldsymbol{w}^+ &= \boldsymbol{w} - \frac{(I - H_2^T(H_2H_2^T)^{-1}H_2)\mathrm{E}\{\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x})\} - \beta\boldsymbol{w}}{\mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\} - \beta},\\ \boldsymbol{w}^* &= \boldsymbol{w}^+/\|\boldsymbol{w}^+\|.\end{aligned} \tag{2.18}$$

As in equation 2.3, the normalization is added to improve the stability. Inspired by the original FastICA approach, to further simplify the above iterative algorithm, we multiply both sides of the first equation in 2.18 by $\beta - \mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\}$ to obtain the following iterative algorithm for finding the specific components of group 1:

$$\begin{aligned}\boldsymbol{w}^+ &= (I - H_2^T(H_2H_2^T)^{-1}H_2)\mathrm{E}\{\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x})\} - \mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\}\boldsymbol{w},\\ \boldsymbol{w}^* &= \boldsymbol{w}^+/\|\boldsymbol{w}^+\|.\end{aligned} \tag{2.19}$$

To find the specific components of group 2, H_2 should be replaced with H_1 in the above equation. Also the shared components would be extracted using the original FastICA algorithm, equation 2.3, since there is no constraint on the shared components in equation 2.10. Because of the approximations used in the derivation of equation 2.19, a proof of convergence may seem necessary. In appendix B, we show that due to the similarity of equation 2.19 with the original FastICA algorithm in equation 2.3, their convergence shares the same properties. Also Newton's method is notorious for not working well if the starting point is far from the solution. So we derive a stabilized version of our algorithm as follows:

$$\boldsymbol{w}^+ = \boldsymbol{w} - \mu\frac{(I - H_2^T(H_2H_2^T)^{-1}H_2)\mathrm{E}\{\boldsymbol{x}g(\boldsymbol{w}^T\boldsymbol{x})\} - \beta\boldsymbol{w}}{\mathrm{E}\{g'(\boldsymbol{w}^T\boldsymbol{x})\} - \beta}. \tag{2.20}$$

By taking small values of μ, which corresponds to the gradient descent method, equation 2.20 converges with more reliability than equation 2.19 does. Also as a general rule, we can always start with $\mu = 1$ and then gradually decrease the value of μ, if the convergence is not satisfactory. To perform decorrelation following each iteration for extraction of several ICs (Hyvarinen & Oja, 2000), a symmetric approach is preferable since the exact number of specific and shared components is unknown to begin with. Recall that the goal of the additional constraint is to make two blocks of matrix $\tilde{A}$ in equation 2.10 as small as possible. To get a faster and more reliable convergence, we can also incorporate the constraint into the decorrelation step using the following iterative algorithm. This algorithm is an extension of equation 2.4 to perform symmetric decorrelation and at the same time to keep the target blocks of matrix $\tilde{A}$ as small as possible, using an adjustment function Φ:

$$\begin{cases} \text{1. Let}\, W = W/\sqrt{WW^T}. \\ \text{Repeat 2 and 3 until convergence :} \\ \text{2. Let}\, W = \dfrac{3}{2}W - \dfrac{1}{2}WW^TW. \\ \text{3. } \tilde{A} = HW^T \rightarrow \tilde{A}_c = \Phi(\tilde{A}) \rightarrow W = (Z\tilde{A}_c)^T. \end{cases} \tag{2.21}$$

Here Z and H are whitening and dewhitening matrices, respectively. Note that in FastICA, $W = A^T = (Z\tilde{A})^T$. Also the adjustment function Φ, which independently operates on each column of matrix $\tilde{A}$, $\tilde{\boldsymbol{a}}_i$, is defined in equation 2.22. If we split vector $\tilde{\boldsymbol{a}}_i$ into two subvectors $\tilde{\boldsymbol{a}}_i^{up}$, and $\tilde{\boldsymbol{a}}_i^{down}$, with N_{g1} and N_{g2} elements, respectively, $\tilde{\boldsymbol{a}}_i = [\begin{matrix}\tilde{\boldsymbol{a}}_i^{up} \\ \tilde{\boldsymbol{a}}_i^{down}\end{matrix}]$, we define Φ as

$$\tilde{\boldsymbol{a}}_i^c = \begin{cases} \tilde{\boldsymbol{a}}_i, & if\ s_i \in shared \\ \begin{bmatrix} \tilde{\boldsymbol{a}}_i^{up} \\ \varphi\tilde{\boldsymbol{a}}_i^{down} \end{bmatrix}, & if\ s_i \in SP1 \\ \begin{bmatrix} \varphi\tilde{\boldsymbol{a}}_i^{up} \\ \tilde{\boldsymbol{a}}_i^{down} \end{bmatrix}, & if\ s_i \in SP2 \end{cases},$$

$$\Phi(\tilde{\boldsymbol{a}}_i) = \frac{\tilde{\boldsymbol{a}}_i}{\tilde{\boldsymbol{a}}_i^c}\tilde{\boldsymbol{a}}_i^c. \tag{2.22}$$

So if $\tilde{\boldsymbol{a}}_i$ corresponds to a shared component, the adjustment function does not change it. If it belongs to *SP1*, the adjustment function multiplies the last N_{g2} elements of $\tilde{\boldsymbol{a}}_i$ by an adjustment factor φ ($\varphi < 1$). If it belongs to *SP2*, the adjustment function multiplies the first N_{g1} elements of $\tilde{\boldsymbol{a}}_i$ by φ. Also $\tilde{\boldsymbol{a}}_i^c$ is scaled to have the same norm as $\tilde{\boldsymbol{a}}_i$. The convergence proof of the original algorithm (equation 2.4) is reported in Hyvarinen (1999). Taking a φ that is of the order of unity (say, between 0.7 and 0.9), ensures that the algorithm of equation 2.21 converges to a symmetric solution. Note that by taking φ equal to unity, equation 2.21 will be simplified to equation 2.4.

As a requisite of our algorithm, we need to estimate the membership index of each component (shared, *SP1*, or *SP2*) at every iteration of equation 2.19. One simple way to check this is to compare the root mean square of $\tilde{\boldsymbol{a}}_i^{up}$ (or $\tilde{\boldsymbol{a}}_i^{down}$) with that of $\tilde{\boldsymbol{a}}_i$, as follows:

$$\begin{cases} \dfrac{RMS(\tilde{\boldsymbol{a}}_i^{down})}{RMS(\tilde{\boldsymbol{a}}_i)} < threshold \rightarrow s_i \in SP1 \\ \dfrac{RMS(\tilde{\boldsymbol{a}}_i^{up})}{RMS(\tilde{\boldsymbol{a}}_i)} < threshold \rightarrow s_i \in SP2 \\ else \qquad\qquad\qquad \rightarrow s_i \in shared \end{cases} . \tag{2.23}$$

Here *threshold* is in the range of (0 to 1). We took *threshold* = 0.5 in our simulations unless otherwise mentioned. Increasing this parameter toward 1 will increase the sensitivity of analysis to extract the specific components, but it will increase the false-positive rate as well. Decreasing the *threshold*value toward zero will increase the specificity of analysis in extracting the specific components, but it will increase the false-negative rate at the same time. We suggest a technique to select *threshold*value in real-world applications in section 4.

In summary, at each iteration of our algorithm, first the components are evaluated to estimate their membership indices using equation 2.23. Then, based on the membership index, matrix W will be updated using equation 2.19. Note that equation 2.19 is derived for specific components of group 1. To obtain the specific components of group 2 or the shared components, H_2 should be replaced with H_1 or the zero matrix, respectively, in equation 2.19. Finally, at the end of each iteration, the modified symmetric decorrelation algorithm in equation 2.21 will be used to avoid convergence of different ICs to the same solution.

2.6 Null Space Interpretation of the New Algorithm. Based on equation 2.13, the specific components of group 1 are in the null space (or the kernel) of matrix H_2, and the specific components of group 2 are in the null space of matrix H_1. Since the rank (the dimension of the row space) of matrix H_2 is N_{g2} (see appendix A for the proof), the dimension of the null space of H_2, nullity(H_2), is $N - N_{g2}$. So the maximum number of specific components of group 1 which can be obtained from the dimension-reduced data, is $N - N_{g2}$. Similarly, the number of specific components of group 2 is limited to $N - N_{g1}$. Recall that the dimension of the concatenated subject-level data is reduced via the middle-level PCA models from n_2T_2 to N_{g2} for group 2 and from n_1T_1 to N_{g1} for group 1. In fact, this step is necessary to provide some free dimensions for the specific components. This means that the specific components of one group are orthogonal to the subspace spanned by the major principal components extracted from the data of the other group.

Also note that the $(I - H_2^T(H_2H_2^T)^{-1}H_2)$ term has appeared as an additional multiplier in equation 2.19 compared to the original algorithm in equation 2.3. The columns of this matrix are in the null space of H_2: $(H_2.(I - H_2^T(H_2H_2^T)^{-1}H_2) = \emptyset)$. Accordingly, this symmetric idempotent matrix integrates the restriction of the specific components into the update algorithm of the original FastICA.

2.7 Extension to Multigroup and Factorial Design. So far we have seen the mathematical formulation to handle the between-group analyses for a two-group (or a two-condition) design. We can extend our algorithm to compare more than two groups or to the factorial design. Here we report the formulation for a three-group design and then for a two-factor design. The generalization to more complex conditions is straightforward.

2.7.1 Extension to Three-Group Design. Let X_1, X_2, and X_3 denote the whitened concatenated observed data matrix for group 1, group 2, and group 3, respectively. By concatenating the data from all groups, and performing the third-level PCA, we obtain

$$X = Z\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \to \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = pinv(Z)X \triangleq HX, \tag{2.24}$$

where Z and H are whitening and dewhitening matrices, respectively. Each independent component of group 1 may belong to one of these four categories: ICs shared across groups (S^{sh}), ICs shared between group 1 and group 2 ($S^{sh(1,2)}$), ICs shared between group 1 and group 3 ($S^{sh(1,3)}$), and ICs specific to group 1 (S^{sp1}). Similar categorization of generative models can be done for groups 2 and 3:

$$\begin{aligned} X_1 &= A_1^{sh}.S^{sh} + A_1^{sh(1,2)}.S^{sh(1,2)} + A_1^{sh(1,3)}.S^{sh(1,3)} + A_1^{sp1}.S^{sp1}, \\ X_2 &= A_2^{sh}.S^{sh} + A_2^{sh(1,2)}.S^{sh(1,2)} + A_2^{sh(2,3)}.S^{sh(2,3)} + A_2^{sp2}.S^{sp2}, \\ X_3 &= A_3^{sh}.S^{sh} + A_3^{sh(1,3)}.S^{sh(1,3)} + A_3^{sh(2,3)}.S^{sh(2,3)} + A_3^{sp3}.S^{sp3}. \end{aligned} \tag{2.25}$$

Equations 2.24 and 2.25 give the following formula for the aggregated data of three groups:

$$HX = HAS = \begin{bmatrix} A_1^{sh} & A_1^{sh(1,2)} & A_1^{sh(1,3)} & 0 & A_1^{sp1} & 0 & 0 \\ A_2^{sh} & A_2^{sh(1,2)} & 0 & A_2^{sh(2,3)} & 0 & A_2^{sp2} & 0 \\ A_3^{sh} & 0 & A_3^{sh(1,3)} & A_3^{sh(2,3)} & 0 & 0 & A_3^{sp3} \end{bmatrix} \times \begin{bmatrix} S^{sh} \\ S^{sh(1,2)} \\ S^{sh(1,3)} \\ S^{sh(2,3)} \\ S^{sp1} \\ S^{sp2} \\ S^{sp3} \end{bmatrix}. \tag{2.26}$$

This equation implies that nine blocks of matrix HA are to be zero. Depending on the type of extracted component, zero (where $\boldsymbol{s}_i \in S^{sh}$), one ($\boldsymbol{s}_i \in S^{sh(1,2)}, S^{sh(1,3)},$ *or* $S^{sh(2,3)}$), or two ($\boldsymbol{s}_i \in S^{sp1}, S^{sp2},$ *or* S^{sp3}) constraints are to be considered. Lagrange multipliers and then Newton's method can be employed in a similar manner, as explained in the case of two groups to iteratively extract the converging solution of the three-group design. Prior knowledge of the problem may allow some source types to be eliminated from the general generative model reported in equation 2.25 to simplify the equation.

2.7.2 Extension to Factorial Design. Let X_{C1}, X_{C2}, X_{P1}, and X_{P2} denote the whitened concatenated data matrix for a control group tested under condition 1, control group in condition 2, patient group in condition 1, and patient group in condition 2, respectively. Again, we may classify ICs based on their types:

$$\begin{aligned} X_{C1} &= A_1^{sh}.S^{sh} + A_1^{C}.S^{C} + A_1^{1}.S^{1} + A_1^{C1}.S^{C1}, \\ X_{C2} &= A_2^{sh}.S^{sh} + A_2^{C}.S^{C} + A_2^{2}.S^{2} + A_2^{C2}.S^{C2}, \\ X_{P1} &= A_3^{sh}.S^{sh} + A_3^{P}.S^{P} + A_3^{1}.S^{1} + A_3^{P1}.S^{P1}, \\ X_{P2} &= A_4^{sh}.S^{sh} + A_4^{P}.S^{P} + A_4^{2}.S^{2} + A_4^{P2}.S^{P2}, \end{aligned} \tag{2.27}$$

where S^{sh} denotes shared components across all groups and conditions, S^C is the shared components between two conditions of the control group, S^1 is the shared components between the control and patient groups in condition 1, and S^{P2} is the specific components of patient group in condition

2 (ditto for the other terms). Obtaining the matrix form of equation 2.27 is straightforward. It results in nine different source types and 20 blocks of zero. Depending on the type of extracted component, zero (where $\boldsymbol{s_i} \in S^{sh}$), two ($\boldsymbol{s_i} \in S^C, S^P, S^1, \mathit{or}\ S^2$), or three ($\boldsymbol{s_i} \in S^{C1}, S^{C2}, S^{P1}, \mathit{or}\ S^{P2}$) constraints should be included in the cost function. Prior knowledge of the problem may allow some of the nine different source types to be eliminated from the general generative model reported in equation 2.27.

3 Results

We illustrate the operation of the proposed algorithm using two simulated data sets: a simple one-dimensional and an fMRI-like two-dimensional data set. The one-dimensional data were generated using four artificial source signals (two subgaussian, and two supergaussian signals) as in Hyvarinen (1999), plus an additional gaussian source signal, due to the importance of gaussian distributions in biological processes (McKeown, Varadarajan, Huettel, & McCarthy, 2002). Note that theoretically, ICA can extract one gaussian source signal. In order to generate two groups of data sets, three of the sources were randomly selected as shared components between groups and one as a specific component for each group. For each group, 10 individual data sets were generated. For each individual, the selected sources (three shared and one specific depending on the individual's group) were mixed using different matrices, whose elements were drawn from a normal gaussian distribution. To account for interindividual variability within each group, we added two additional types of noise to each individual's data. First, we randomly shifted each source vector for each individual in a circular fashion. This manipulation is analogous to the observed spatial variability in the anatomical maps of different individuals in the context of fMRI functional connectivity analysis, so we refer to it as spatial noise. Second, gaussian noise was added to the data matrix of each individual. As performance measures, we calculated the root mean squared error between the extracted and the original source signals (RMSE) and the squared correlation between them (r^2) averaged over different sources, defined as follows:

$$\mathrm{RMSE} = \frac{1}{N_g} \sum_{i=1}^{N_g} \left[\frac{1}{M} \sum_{j=1}^{M} (s_{i,j} - \tilde{s}_{i,j})^2 \right]^{1/2}$$

$$r^2 = \frac{1}{N_g} \sum_{i=1}^{N_g} \left[\mathit{corr}\left(\boldsymbol{s}_i, \tilde{\boldsymbol{s}}_i\right) \right]^2, \tag{3.1}$$

where $s_{i,j}$ denotes the jth element of the ith original source vector $\boldsymbol{s}_i$, and $\tilde{\boldsymbol{s}}_i$ is the ith extracted source vector. N_g is the number of extracted sources, and *corr*(.) represents Pearson correlation. RMSE and r^2 measure error and performance of source signal reconstruction, respectively. To quantify the classification error related to correct classification of shared and specific sources for each group, we used receiver operating characteristics (ROC) analysis. The area under ROC curve is calculated to measure the classifier performance.

3.1 Simulation Results on One-Dimensional Data Set. We compare the results of the proposed algorithm with two approaches that are used in between-group ICA (Albert et al., 2009; Assaf et al., 2010; Calhoun et al., 2009). One approach is based on applying ICA on the aggregate data of two groups (Assaf et al., 2010). This approach may decrease the reconstruction error for the shared components, but it is insensitive to detection of specific components. The other approach is to apply two separate runs of ICA on the concatenated data of each group and then compare the similarity of the extracted ICs between groups to investigate the shared and specific components (Albert et al., 2009). Then as a measure of similarity, the correlation between ICs of one group and every IC of the other group is calculated and ICs that pass a predefined threshold (*TR*; usually in the range of [0.5–0.7]) were identified as shared components. However, if the correlation between one component of group 1 and every component of group 2 is insignificant (e.g., less than 0.5), that component will be identified as a specific component of group 1. In all simulations, the FastICA algorithm was applied on the sphered data using the hyperbolic tangent as the derivative of contrast function $g(u) = \tanh(u)$ and the symmetric decorrelation approach. The value of φ defined in equation 2.22 is set at 0.7 for all simulations using our algorithm (we manually checked that changing φ in the range of [0.5–0.9] had a minimal effect on the performance results). Figure 1 illustrates an example of the extracted components using different approaches (FastICA on the concatenated data of two groups, FastICA on the concatenated data of each group, and the proposed algorithm) overlaid on the correct solutions in gray. As indicated in Figure 1C, the proposed algorithm correctly classified three components as shared, one as *SP1*, and one as *SP2*.

To investigate the performance of source signal reconstruction, RMSE and r^2 measures were estimated and averaged over 600 runs for each combination of shared and specific components for the five sources depicted in Figure 1. We calculated the reconstruction performance at various spatial noise levels ($n = 0, 1, 2, 3, 4$, or 5), where for each individual, the source signals are shifted by a value randomly selected from the integers in the range of $[-n, n]$. The correlation between some ground truth source signals shifted by $+3$ steps and the same signal shifted by -3 steps was as low as 0.3. So for the shift levels greater than $n = 3$, when the correlations dropped to a value around 0.1, it was not possible to extract some source signals. As shown in Figure 2, the reconstruction performance is significantly improved using our algorithm compared to the other approaches in which the

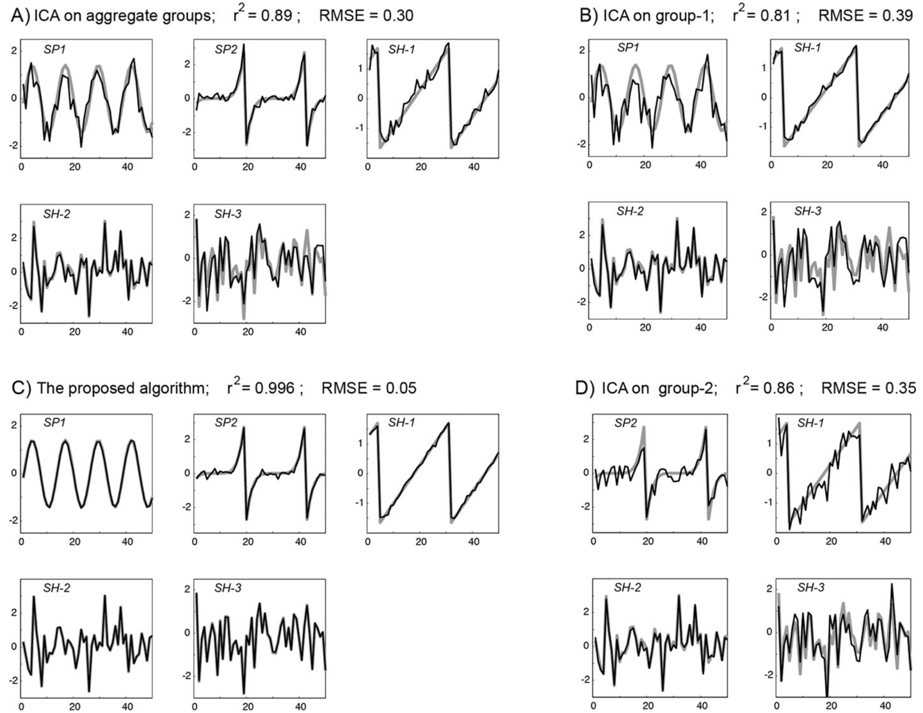

Figure 1: Samples of extracted components using (A) FastICA on the aggregate data of two groups, (B) FastICA on the concatenated data of group 1, (D) FastICA on the concatenated data of group 2, and (C) the proposed algorithm. The ground-truth signals are shown in gray. In this example, the data of two groups were generated using three shared components (*SH1-3*), one specific component for group 1 (*SP1*), and one specific component for group 2 (*SP2*) with spatial noise = 0.

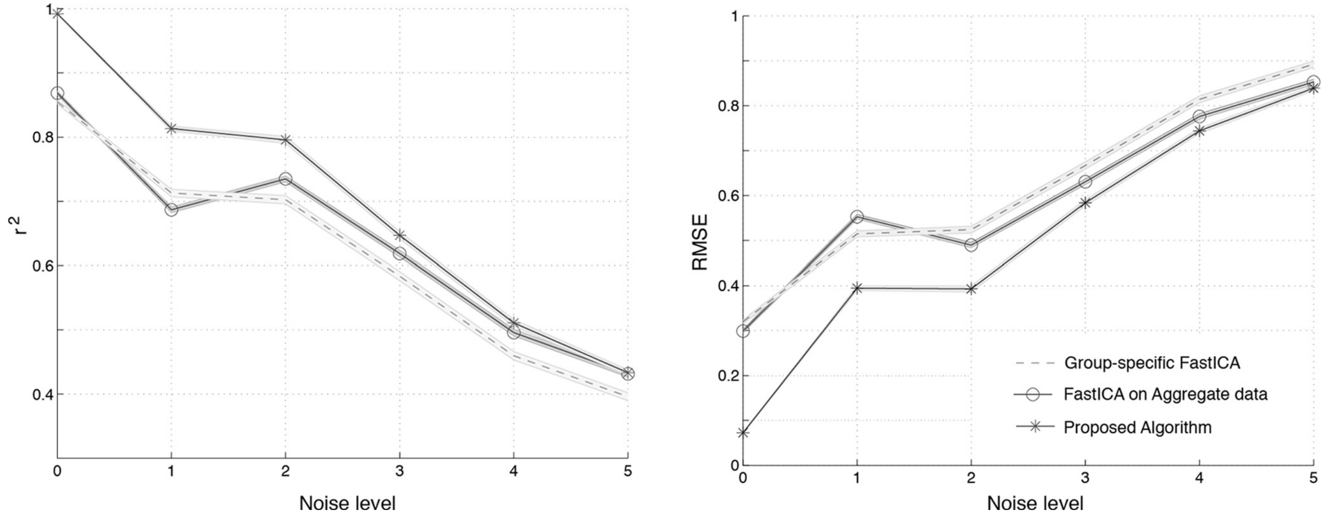

Figure 2: RMSE and r^2 averaged over 600 runs of ICA at different noise levels. Asterisks, circles, and dashed line, respectively, represent the performance of the proposed algorithm (with $N = 5$, $N_{g1} = 4$, $N_{g2} = 4$), FastICA on the aggregate data ($N = 5$), and separate FastICA runs on either group ($N = 4$ in both cases). The shaded area specifies the standard error. "Noise level" refers to the maximum magnitude of the spatial noise.

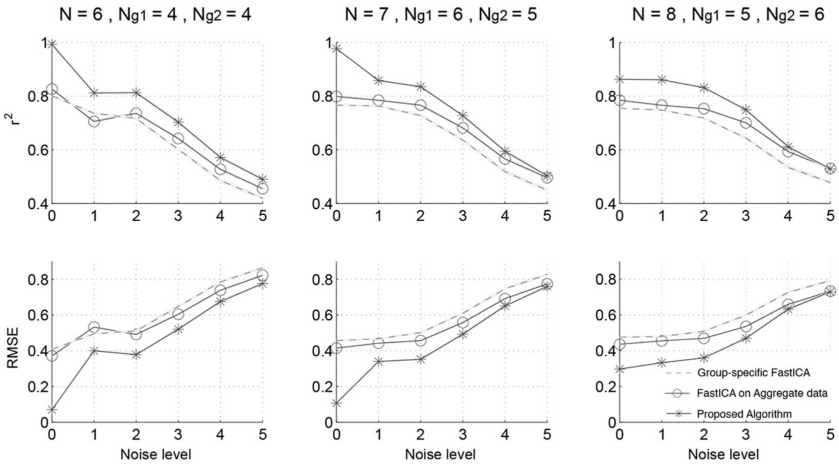

Figure 3: RMSE and r^2 averaged over 600 runs of ICA at different noise levels and with three different numbers of extracted components. Each column shows the results for different numbers of extracted components. Asterisks, circles, and the dashed line, respectively, represent the performance of the proposed algorithm (with parameters specified at the top of each column), FastICA on the aggregate data (with N extracted components as specified at the top of each column), and separate FastICA runs on either group (with $N-1$ extracted components in either case).

group membership information is neglected or half of the data set is not used in different runs of ICA on each group data set. As shown for noise levels equal to 4 or 5, when some of the source signals are almost uncorrelated between individuals, the performance of the proposed algorithm converges to that of the method on the concatenated data of two groups.

It is also possible that since we input the exact number of shared and specific components in our algorithm, the results are superior. To investigate the robustness of our algorithm when the number of shared and specific components is not correct, we ran the analysis on the same data set but with erroneous numbers of shared and specific components to be extracted. As explained in section 2.6, our algorithm extracts a total number of N ICs, with a maximum of $N - N_{g2}$ specific ICs for group 1 and up to $N - N_{g1}$ specific ICs for group 2. So the number of shared components may vary from $N - (N - N_{g1}) - (N - N_{g2}) = N_{g1} + N_{g2} - N$, where the algorithm does not find any specific components. As shown in Figure 3, the reconstruction performance is robust with respect to the incorrect number of shared and specific components. Note that to calculate the performance in Figure 3, the first five components that showed the highest correlation with the ground-truth signals are selected for all three methods.

To investigate the classification performance, we calculated the false-positive rate and the true-positive rate in the proposed and the regular algorithms, where shared and specific components represent positive and negative class labels, respectively. To simplify the ROC analysis, we did not distinguish between specific components of either class. In our simulations, the proposed algorithm never classified a specific component of one group as the other group's specific component, and for the regular algorithm, the occurrence was very rare (3 cases in 600 runs). As explained before, the regular algorithm is based on comparing the correlation values of the extracted components of two groups with a predefined threshold (*TR*). So to construct the ROC curve in the regular algorithm, performance is calculated at different *TR* values ranging from 0 to 1. Since in the proposed algorithm the classification threshold (*threshold* in equation 2.23) is implicit in the algorithm, we ran the algorithm multiple times with different *threshold* values ranging from 0 to 1, to get the classification performance at different points of the ROC curve. Figure 4 shows the ROC curves at different spatial noise levels (0, 2, or 4) for the new and regular algorithms using the one-dimensional data set explained before. As shown in Figure 4, the classification performance is constantly superior in the proposed algorithm. Comparison of the optimum points of ROC curves shows the higher sensitivity of our algorithm in detecting shared and specific components and its greater specificity by having lower classification error.

Figure 5 shows the classification performance when the numbers of extracted components are incorrect. As can be seen in the figure, the proposed adaptive classifier is quite robust with respect to inaccuracy in numbers of extracted components. Table 1 reports false- and true-positive rates at the cut-off point for the best sensitivity and specificity in the ROC plot for the regular and proposed algorithm at different noise levels and with different numbers of extracted components. In the ROC plot, the cut-off is the point closest to (0,1). The average threshold of the ROC curve at the best cut-off point over different conditions was 0.25 for the proposed algorithm (*threshold*) and 0.35 for the regular approach (*TR*).

3.2 Sensitivity to Detect Variations in Amplitude. One of the motives behind the proposed approach is to obtain a higher sensitivity to detect those shared components whose power (variance accounted for by that component) significantly varies across groups (or conditions). We use the percentage variance accounted for (PVAF) as a measure to quantify the percentage of variability that a single IC can explain in an individual's data matrix as follows:

$$\mathrm{PVAF} = 1 - \frac{\mathrm{var}(X - \hat{X})}{\mathrm{var}(X)}, \tag{3.2}$$

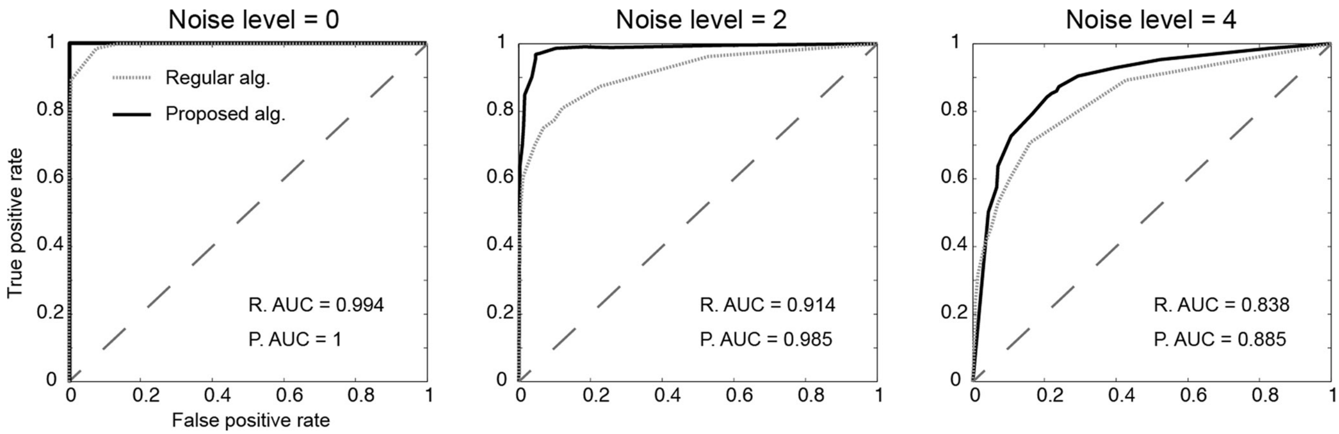

Figure 4: ROC curves representing the classification performance in specifying shared and specific components for different spatial noise levels. The solid black and dotted gray curves show the performance of the proposed algorithm (with $N = 5$, $N_{g1} = 4$, $N_{g2} = 4$) and the regular approach (with $N = 4$ for ICA runs on group 1 and group 2), respectively. Each ROC curve for the regular algorithm is generated using 600 runs of ICA (30 repetitions of 20 different selections of three shared, one *SP1*, and one *SP2*component). For our algorithm, we ran 600 runs at each of 15 different *threshold* values in the range of [0–1] to construct each ROC curve. P.AUC and R.AUC stand for the area under curve for the proposed and regular algorithms, respectively.

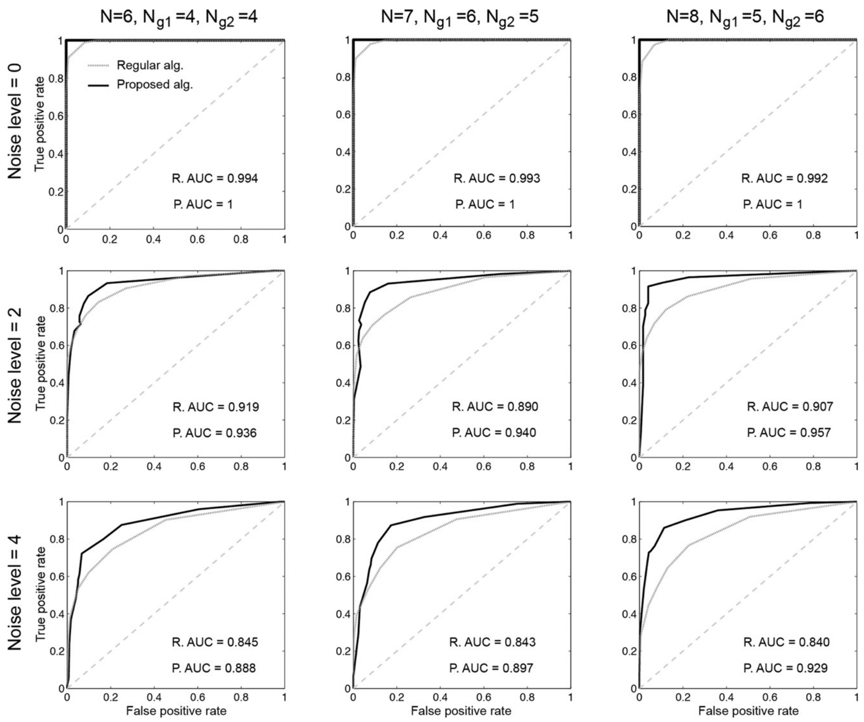

Figure 5: ROC curves representing the classification performance of shared versus specific components for different noise levels and for different numbers of extracted components. Each column shows the ROC curves for different numbers of extracted components. The solid black and dotted gray curves represent the performance of the proposed algorithm (with parameters specified on top of each column) and the regular approach (with *N*-1 extracted components in either group), respectively.

Table 1: False-Positive (FP) and True-Positive (TP) Rates at the Cut-Off Point for the Highest Sensitivity and Specificity in ROC Plots of Figures 4 (Extra Components = 0) and 5 (Extra Components = 1, 2, or 3).

		Extra Components = 0			Extra Components = 1			Extra Components = 2			Extra Components = 3		
	Noise Level	0	2	4	0	2	4	0	2	4	0	2	4
Regular	FP rate	0.077	0.122	0.21	0.083	0.143	0.208	0.078	0.145	0.202	0.068	0.122	0.227
algorithm	TP rate	0.986	0.801	0.755	0.982	0.832	0.746	0.976	0.766	0.754	0.973	0.791	0.766
Proposed	FP rate	0	0.047	0.21	0	0.097	0.167	0	0.077	0.173	0	0.040	0.113
algorithm	TP rate	1	0.969	0.842	1	0.865	0.80	1	0.885	0.873	1	0.916	0.860

Note: FP and TP rates are reported, for the regular and proposed algorithm, for different noise levels and numbers of extracted components for the one-dimensional data set.

where X is the individual's demeaned data matrix and $\hat{X}$ is the backprojection of the component of interest onto X. var(.) denotes the average of variances of different rows of a matrix. $\hat{X}$ is calculated by making all the columns of mixing matrix A zero, except the one corresponding to the component of interest, and then applying all dewhitening transformations to obtain the subject-level data matrix.

To investigate the sensitivity of our algorithm to the variations in the PVAF of the shared components across groups, we manipulated the power of a shared component in all individuals of one group by scaling the corresponding column of that component in the generative mixing matrix (see equation 2.1). At each spatial noise level ranging from 0 to 3, we performed 900 runs of the regular ICA algorithm on each group separately and 900 runs of the proposed algorithm, using all possible combinations of shared and specific components with a shared component whose power was manipulated across groups on the one-dimensional data set. In every run of either approach, we calculated the PVAF of each shared component for each individual using equation 3.2. Then in both approaches, we compared the PVAF measures of each shared component across groups using two-sample t-statistics with various critical values α. Thus, if the observed t-value for a shared component was less than a threshold α, that shared component was identified as a power-varying shared component across groups. We evaluated the performance of identification of power-varying type among all shared components using an ROC analysis where power-varying and fixed-power shared components represent positive and negative class labels, respectively. For either the proposed or the regular approach, the critical value α served as the threshold at different points of the ROC curve. The results of this analysis are illustrated in Figure 6. The sensitivity of the proposed algorithm is superior in detecting changes in the power of the shared component across groups, with a greater specificity of the classification (i.e., a lower false-positive rate).

3.3 Simulation Results on Two-Dimensional fMRI-Like Data Set. So far we have used one-dimensional data sets to investigate various features of the proposed algorithm. To investigate its performance in a more complex and realistic situation, we used fMRI Simulation Toolbox (SimTB) to generate a two-dimensional fMRI-like data set under a model of spatiotemporal separability (Allen et al., 2011; Dea, Anderson, Allen, Calhoun, & Adali, 2011; Erhardt, Allen, Wei, Eichele, & Calhoun, 2012) . In the context of fMRI functional connectivity analysis (Vahdat, Darainy, Milner, & Ostry, 2011), the simulated data can be expressed as the product of time histories (mixing matrix) and spatial maps (sources). We used the conventional parameters of SimTB to define the tissue-type weights (to model variations in the baseline intensity of different tissue types in the brain), the level of spatial noise across individuals (translation, rotation, and spread), and the

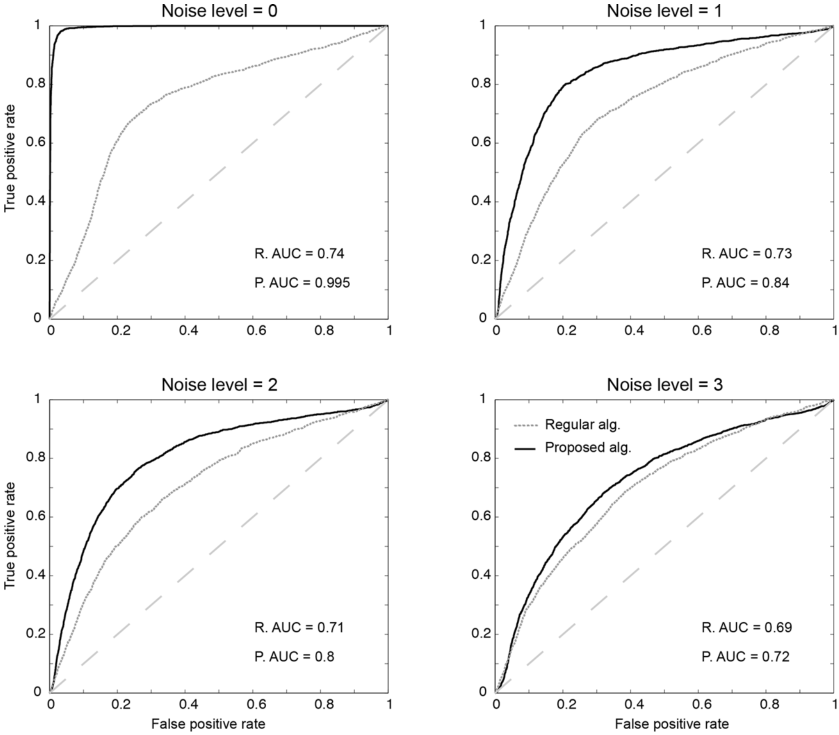

Figure 6: ROC curves representing the classification performance of power-varying versus fixed-power shared components for different noise levels. Each panel shows the results of 900 runs using all possible combinations of three shared and two specific components. Among the three shared components, two preserved the same power across groups, but the power of one shared component was manipulated between two groups (scaling the corresponding column of the mixing matrix by 0.6). $threshold = 0.7$, $N = 5$, $N_{g1} = 4$, $N_{g2} = 4$ were used in the proposed algorithm and $N= 4$ extracted components for either group in the regular approach.

level of Rician noise added to the data set. We defined the mixing matrix for each individual using convolution of different gaussian signals with the canonical hemodynamic response function, bandpass-filtered in the frequency range of 0.01 Hz to 0.1 Hz to model resting-state BOLD fMRI time series (Fox & Raichle, 2007). Ten compound source signals (see Figure 7A) were generated based on several one-part source signals defined in SimTB. Also to account for the intracomponent variability in the time series across distinct subparts of a compound component (e.g., because of the differences in the hemodynamic response function between different brain areas), we

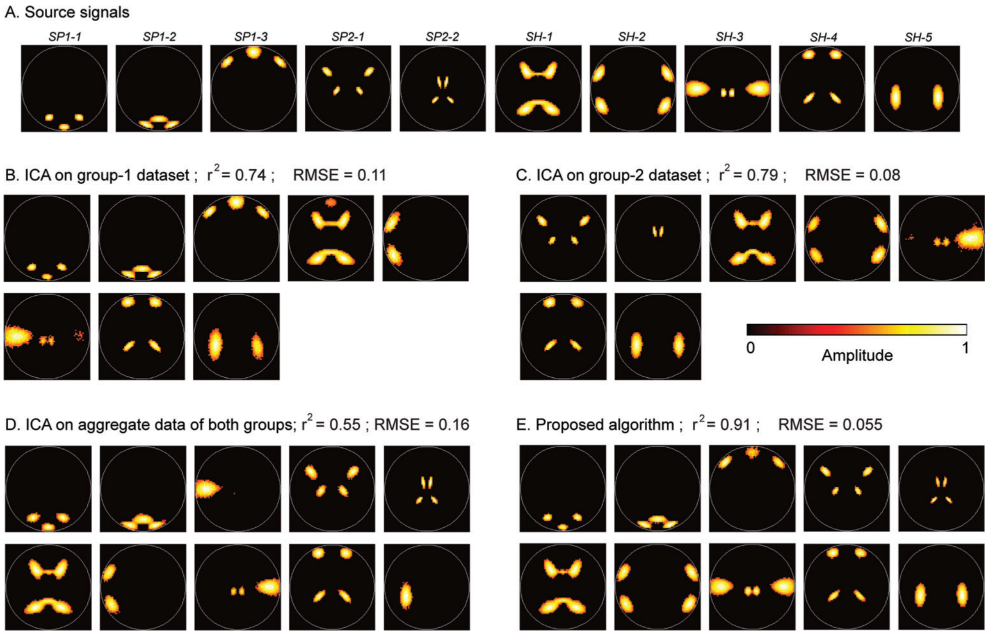

Figure 7: An example of 10 two-dimensional fMRI-like components simulated using SimTB and the extracted components using different approaches. (A) The correct solutions. Extracted components using (B) FastICA on the concatenated data of group 1 (with $N = 11$), (C) FastICA on the concatenated data of group 2 (with $N = 10$), (D) FastICA on the aggregate data of two groups (with $N = 13$), and (E) the proposed algorithm (with $N = 13$, $N_{g1} = 9$, $N_{g2} = 9$). In this example, the data of two groups were generated using three specific components for group 1 (*SP1*), two specific components for group 2 (*SP2*), and five shared components (*SH*) with intracomponent variability noise = 0.15. In all approaches, the first-level PCA reduced the data of each individual to 20 dimensions. For each approach, RMSE and r^2 measures are reported at top of each panel.

created fluctuations in the time series of distinct subparts within a component by adding different zero-mean gaussian noises with σ between 0.1 and 0.25. In order to generate two groups of data sets, K_1 and $K_2 (0 \leq K_1, K_2 \leq 3)$ sources were randomly selected as specific components for group 1 and group 2, respectively, and $K = 10 - K_1 - K_2$ sources as shared components between groups. For each group, 10 individual data sets (of 50 time points, 64 × 64 pixels image) were generated. Figure 7 illustrates an example of the extracted components based on the proposed algorithm, the regular FastICA algorithm on each group, and the aggregate data of both groups. Comparing the results of regular ICA on time-concatenated data of each group separately (see Figures 7B, and 7C) shows significant discrepancy of some shared components (e.g., *SH2* or *SH3*) between groups, which may lead to incorrectly classifying them as specific components. Also the results of the FastICA algorithm on the aggregated data of both groups (see Figure 7D) illustrate that this method might totally fail to extract some of the specific components (*SP1-3*in this example), resulting in low sensitivity in identification of the pattern of differences between groups. As illustrated in Figure 7E, in this example, the best performance is obtained using the proposed algorithm.

To further investigate the classification performance of the proposed and the regular approaches in differentiating specific from shared components, we ran both algorithms on 800 different realizations using different numbers and combinations of shared and specific components and with different levels of the intracomponent variability noise (std between [0.1–0.25]), as explained above. Figure 8 illustrates the ROC curves using different numbers of extracted components for both approaches. As shown in Figure 8, the proposed algorithm is much more robust with regard to the incorrect number of extracted components compared to the regular approach. It is worth noting that in our algorithm, the classification performance is almost perfect when allowing four extra components to be extracted in this complex data set. So in more complex regimes, the extra number of components to be extracted may help the stability of the classification of shared and specific components in the proposed algorithm.

Table 2 reports false- and true-positive rates at the point of the highest sensitivity and specificity in the ROC plot for the regular and proposed algorithm using different numbers of extracted components. In all cases, the classification performance of the proposed algorithm was superior compared to the regular approach. The average threshold of the ROC curve at the best point over different conditions was 0.6 for the proposed algorithm ($threshold$) and 0.43 for the regular approach (TR).

3.4 Simulation Results on Hybrid fMRI Data. Twelve right-handed healthy adults ages 18 to 24 were scanned at the Montreal Neurological Institute (MNI). The study was approved by the local ethics committee, and subjects participated in the research after giving written informed consent.

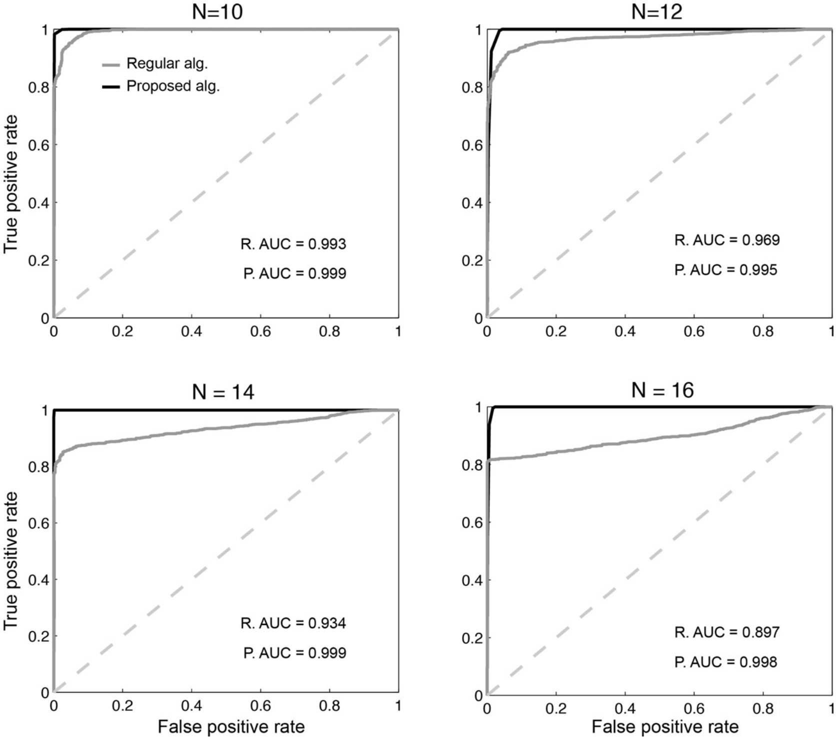

Figure 8: ROC curves representing the classification performance of shared versus specific components using different numbers of extracted components as input. Each ROC curve is derived based on the classification results using different data sets generated from randomly selected numbers of shared and specific components, with random assignment of components to categories. The numbers of *SP1*and *SP2* (K_1 and K_2) were randomly drawn from integers in the range of [0–3]. N at the top of each panel specifies the number of extracted components in the proposed algorithm. $N_{g1} = N - c_1,\ N_{g2} = N - c_2$, where $c_1 = K_2, c_2 = K_1$ (top left), $3 \leq c_1, c_2 \leq 4$ (top right), and $3 \leq c_1, c_2 \leq 6$ (bottom left and right, respectively). In the regular approach, $N - K_2$ and $N - K_1$ components were extracted for group 1 and group 2, respectively.

Functional images were continuously acquired using a 32-channel 3T MR scanner (Siemens Trio, Germany). A T1-weighted anatomical image was first acquired (1 mm slice thickness, 256 × 256 matrix; TE = 7.4 ms and TR = 23 ms; flip angle 30°) and used for superposition of the functional images and inter subject group coregistration. Two runs of 200 s each were acquired from every subject during the resting-state condition. The functional data were acquired using a T2*-weighted EPI sequence (3.5 × 3.5 × 3.5 mm

Table 2: False-Positive (FP) and True-Positive (TP) Rates at the Cut-Off Point for the Highest Sensitivity and Specificity in ROC Plots of Figure 8 for the Regular and Proposed Algorithm Using Different Numbers of Extracted Components for the Two-Dimensional Data set.

	Number of Extra Components	0	2	4	6
Regular algorithm	FP rate	0.064	0.054	0.027	0.002
	TP rate	0.967	0.909	0.849	0.810
Proposed algorithm	FP rate	0.002	0.035	.001	0.016
	TP rate	0.982	0.995	0.999	0.996

Note: Number of extra components were 0, 2, 4, and 6 corresponding to $N = 10$, $N = 12$, $N = 14$, and $N = 16$ extracted components in Figure 8, respectively.

voxels, 39 slices, 64 × 64 matrix; TE = 25 ms and TR = 2000 ms; flip angle 90°). Resting-state was defined as a state of relaxed wakefulness when subjects had their eyes open and were instructed to focus on a cross in the middle of a white screen.

Data processing was carried out using the FMRIB Software Library (FSL) (www.fmrib.ox.ac.uk, Oxford U.K.), FSL version 4.1 (Smith et al., 2004; Woolrich et al., 2009). The following preprocessing steps were applied to functional data: (1) removal of the first two volumes of each scan to allow for equilibrium magnetization, (2) slice timing correction using Fourier-space time-series phase shifting, (3) non brain tissue removal, (4) rigid-body motion correction, (5) global intensity normalization of all volumes of each run as implemented in FSL, (6) spatial smoothing using a gaussian kernel with 6 mm full width at half maximum, and (7) high-pass temporal filtering with cut-off frequency of 0.01 Hz. Conversion of the low-resolution functional data to the average standard space (MNI152) involved two transformations. The first was from the low-resolution EPI image to the T1-weighted structural image (using a 7 degree-of-freedom affine transformation), and the second was from the T1-weighted structural image to the average standard space (using a 12 degree-of-freedom linear affine transformation, voxel size = 2 × 2 × 2 mm). The transformed data in MNI space were then subsampled in 4 mm isotropic voxels. We randomly divided the subjects into two groups, resulting in 12 resting-state functional runs for each group. To create hybrid fMRI data, we added three arbitrary patches of activity: one shared between groups located in the left prefrontal region (see Figure 9A, left), one specific to group 1 located bilaterally in the cerebellum (see Figure 9A, middle), and one specific to group 2 located bilaterally in the basal ganglia (see Figure 9A, right). The maps have a relative amplitude of 1 in the central region and 0.7 in the outer region. For each subject, the time course of each component was defined separately using different realizations of standard gaussian signals bandpass-filtered in the frequency range of 0.01 Hz to 0.1 Hz to mimic the neuronal-related portion of resting-state

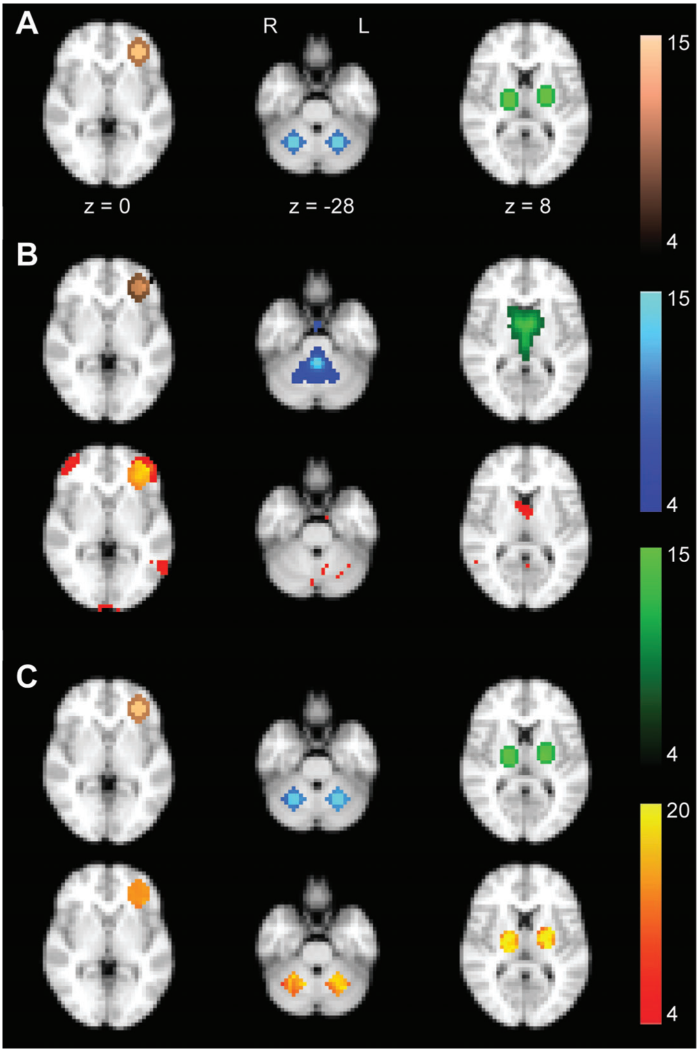

Figure 9: The manipulated ground-truth networks (A), results of the back-reconstruction method (B), and the proposed algorithm (C). The left column shows the manipulated shared network, the middle column the specific network for group 1, and the right column the specific network for group 2. In *A* the light and dark color contrast within each network encodes the relative amplitude of the inner (1) compared to the outer (0.7) part. In *B* and *C* the color-coded Z-score component maps are shown in copper (for shared), blue (specific group-1), and green (specific group 2) in the first row, and the corresponding *t*-statistics map is coded in a red to yellow scale in the second row. As shown, the sensitivity to identify specific networks is increased in the proposed method compared to the back-reconstruction method.

BOLD fMRI time series (Fox & Raichle, 2007). For each component and each subject, the signal-to-noise ratio was set to 1.5 by scaling the time course of the added component relative to the variance of time series averaged over all gray matter voxels of that subject. In this way, we ensured that the added components are among the five strongest components.

To extract the shared and specific components across groups, all hybrid preprocessed resting-state fMRI runs were time-concatenated and fed to the proposed algorithm. We compared the results of our algorithm with the back-reconstruction method as implemented in GIFT software (Assaf et al., 2010; Calhoun et al., 2001). This method is one of the most popular methods for between-groups fMRI analysis of resting-state data using ICA (Assaf et al., 2010; Ma, Narayana, Robin, Fox, & Xiong, 2011). We used the same method as explained in Assaf et al. (2010) to obtain the group-level t-statistics maps corresponding to each component. Briefly, this method performs back-reconstruction on the results of ICA on time-concatenated preprocessed data of all groups to generate spatial maps specific to each subject. Then two-sample $t-$statistics (over a mask defined by the thresholded map of one-sample t-test, $t > 1.5$) are performed to obtain the map of differences between the two groups for every component (thresholded at $t = 4.02$, $p < 0.001$, df $= 11$). Also, one-sample t-statistics on subject-level maps of those components showing no significant difference across groups were used to generate the statistical map of the shared components. For a fair comparison with our algorithm, FastICA was selected as the ICA approach in GIFT software with parameters as described earlier, though the Infomax algorithm (Bell & Sejnowski, 1995) generated very similar results. We extracted 25 networks with both methods and let the proposed algorithm extract up to 3 specific networks for each group. To generate the t-statistics maps using our method, we also back-reconstructed components to get subject-level spatial maps. Then to extract the shared components, we performed one-sample t-statistics on the subject maps of both groups and on the maps of subjects within the same group as the corresponding specific component to extract the specific components. Note that the back-reconstruction of the specific networks on the data of the other group resulted in zero maps, since specific components were constrained to be orthogonal to the subspace spanned by the other group's data.

The results of both methods in extracting three commonly reported resting-state networks (shared among groups) are illustrated in Figure 10: visual, auditory, and default mode networks that were selected by visual inspection among all extracted components (Damoiseaux et al., 2006). As expected, both algorithms are capable of extracting the consistent shared resting-state networks, even when extraneous patches of activity were added to the hybrid fMRI data.

Figure 9 shows the results of analyses related to the manipulated networks. As demonstrated in Figure 9 (left column), both algorithms extracted the shared frontal network precisely, as projected in the Z-score

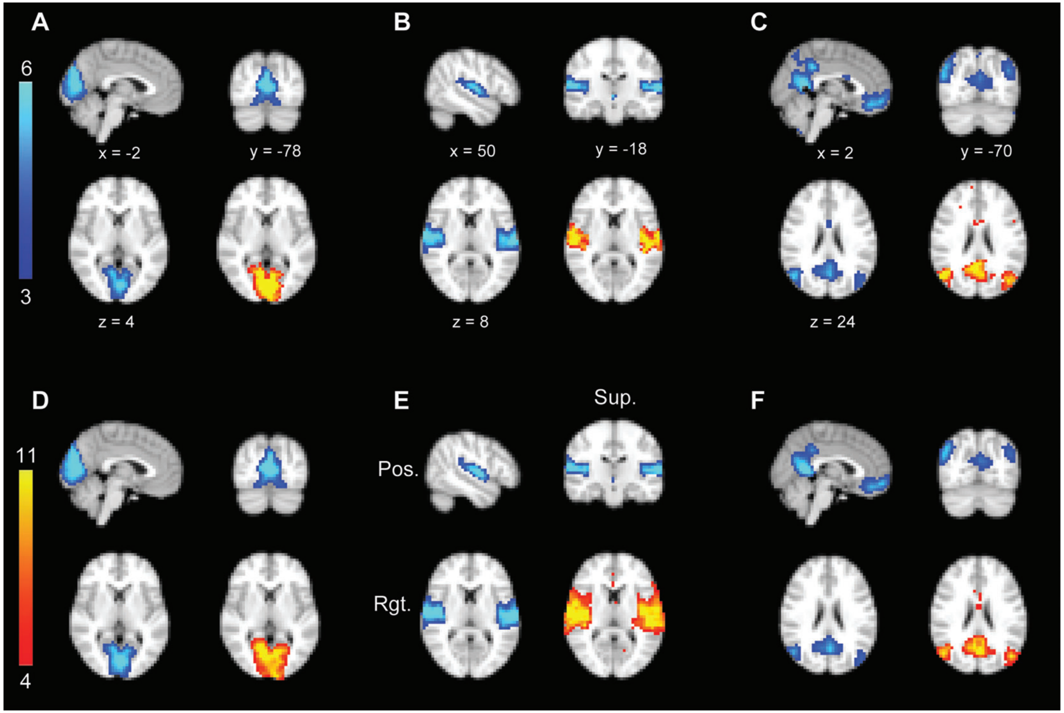

Figure 10: The results of the proposed algorithm (A–C) and the back-reconstruction method (D–F) in extracting three typical resting-state networks. *A* and *D* show visual, *B* and *E* auditory, and *C* and *F* default mode networks. Color-coded maps show the component Z-scores in blue (in sagittal, coronal, and axial views) and the corresponding *t*-statistics map in red to yellow (in sagittal view). x, y, and z reports the coordinates in MNI space. Sup., Pos., and Rgt. stand for superior, posterior, and right, respectively.

maps. However, the back-reconstruction method generated some spurious activity in the prefrontal network *t*-statistics map as well. To find the specific networks using the back-reconstruction method, first networks with the highest correlation with the gold standards (see Figure 9A) were identified. Then two-sample *t*-statistics were performed, as explained earlier. The proposed algorithm identified only one specific network per group, though we allowed extraction of up to three specific components. Figure 9B shows the results of the back-reconstruction method and Figure 9C the proposed algorithm for the specific networks. As it is shown, in contrast to the back-reconstruction method, the proposed algorithm extracted the specific networks accurately and with significant *t*-statistics maps. The regular back-reconstruction method was unable to extract the pattern of specific networks, since it seeks networks that share variance across all subjects, disregarding the group membership information.

4 Discussion

We developed a method to find shared and specific components between different experimental groups in the context of linear ICA. For preprocessing, we employed a three-step PCA to reduce the number of degrees of freedom. In real data applications, this would also be useful for denoising and eliminating redundancies. A new constraint was added to the FastICA algorithm to simultaneously deal with the data of multiple groups. The proposed method of between-group ICA has a number of desirable properties when compared with the regular approach:

- The classification of shared and specific components is adaptive and automatic. The classification improves iteratively until convergence is achieved (under the assumptions made in the proof of convergence; see appendix B).
- The source reconstruction performance is enhanced due to use of the variability of the entire data set. Also the performance in detecting and classifying specific components is improved due to the fact that the specific components are constrained to be orthogonal to the space spanned by the data set of other groups.
- Since a single ICA run is performed on the aggregate data of all groups, even if a shared component splits, the resulting subcomponents will be the same in all groups. So splitting a shared component does not result in the misclassification of some subcomponents as specific components.
- The sensitivity to detect changes in the amplitude of shared components across groups is enhanced due to the fact that the shared components are perfectly matched across groups.

One of the limitations of the proposed algorithm is in cases when the within-group variability of some independent components is so large that the true component cannot be extracted in the regular FastICA or the proposed approach. In these cases, the performance of the proposed algorithm converges to that of the regular approach (see Figures 2 and 3 at noise level = 4, 5). Another issue is the rate of convergence. Our algorithm converges linearly to the solution (see appendix B), while the regular FastICA algorithm has proven to converge quadratically. In our simulations, it took about 20 to 80 iterations for the proposed algorithm to converge at the epsilon level of 0.001 (epsilon is the stopping criterion in the FastICA algorithm).

As the number of extracted components is a free parameter in the ICA algorithm, a number of methods have been proposed to estimate its value in different applications (Li, Adali, & Calhoun, 2007). Based on separate estimations of the number of extracted components in each group, as well as in the aggregate data of all groups, one may approximate the N_{g1}, N_{g2}, and N parameters. Another approach is to start by allowing one specific component per group ($N_{g1} = N_{g2} = N - 1$) and then gradually increasing the number of specific components until adding more components results in the repetition of existing components (this was the case specifically in the two-dimensional fMRI-like data set).

Another method, which has been proposed to incorporate the group membership information into the ICA cost function, is coefficient-constrained ICA (Sui, Adali, Pearlson, & Calhoun, 2009; Sui, Adali, Pearlson, Clark, et al., 2009). It is an efficient method that increases the identification sensitivity of the components that show a significant group difference. As this method optimizes the *t*-statistics values that test the mean difference between distribution of mixing matrix coefficients among two groups, it can be applied only to a set of scalar features extracted from the data sets, and not directly to the untransformed signals (e.g., to the resting-state fMRI time series). Another method sensitive to between-group differences is local linear discriminant analysis (McKeown et al., 2007; Palmer, et al., 2010). It iteratively determines the combination of regions that maximally discriminate between groups or tasks based on sampling *t*-statistics from each region of interest to create feature vectors. Although this method can efficiently account for the spatial variations between anatomical maps of different subjects, it needs prior information regarding the selection of regions of interest, as well as extraction of some task-related features for each voxel in advance.

In the context of fMRI functional connectivity analysis, some methods for conducting between-group ICA have been proposed (Calhoun et al., 2001; Erhardt et al., 2010; Zuo et al., 2010). Two frequently used methods are back-reconstruction (Calhoun et al., 2001) and dual regression (Zuo et al., 2010). These methods in general use the results of Infomax (Bell & Sejnowski, 1995) or FastICA algorithms (applied on either the concatenated data of each group separately or the aggregate data of both) to construct Z-statistic maps for each component, based on the consistency of that component within and across the individuals of different groups (Assaf et al., 2010; Erhardt et al., 2010). This kind of analysis can be applied to the results obtained from our algorithm to generate component-based *t*-statistic maps (as shown in Figures 9 and 10). For instance, in the dual regression algorithm, the second regression model can be adjusted according to the results of our method: a contrast vector differentiating the subjects of two groups and a vector representing the average of the individuals in both groups should be used as regressors for specific components and shared components, respectively.

It is worth noting that the first level (within-subject) data-reduction step is not necessary for operating the proposed algorithm. However, in the case of huge and noisy data sets such as fMRI or EEG, it is recommended (Calhoun et al., 2009; Viviani, Gron, & Spitzer, 2005). Moreover, dimension reduction prevents overlearning, which is observed in ICA (Hyvarinen, 1999). The second (within-group) data-reduction step can be helpful in retaining the pattern of similarities shared among subjects of one group and removing the components represented in just one or a few subjects, though it may filter out some low-power shared components. The third-level (between-groups) data-reduction step is essential to limit the number of extracted components; as explained earlier, various methods have been suggested to estimate this number, for example, in real fMRI data sets (Li et al., 2007).

The values of different parameters used in our algorithm can be determined specifically based on the particular application. For example, in a data set comprising data of healthy controls and patients, the *threshold* parameter can be specified as the lowest value in the range of [0–1], which detects no specific component, when applying our algorithm to the healthy control data, randomly assigned to two subgroups. One area of concern is the quality of spatial registration of different subjects' data sets into a common space. Although simulations using SimTB software (see Figures 7 and 8) showed that the proposed algorithm is relatively robust to motion noise and variability in the location of components across subjects, consistent with a recent study of group ICA using time concatenation (Erhardt et al., 2012), future work is needed to evaluate the robustness of our algorithm to normalization errors on real data. Further analyses on real-world datasets are also needed to examine the efficiency of the proposed algorithm and the sensitivity of its parameters in different experimental designs.

Appendix A: Proof of Invertibility in Equation 2.15

For matrix $H_2H_2^T$ to be invertible, we need to show that the rank of the N_{g2}-by-N matrix H_2 is N_{g2}.

Based on equations 2.8 and 2.12, we can write

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} X \rightarrow X_2 = H_2 X \rightarrow \operatorname{rank}(X_2) = \operatorname{rank}(H_2 X). \tag{A.1}$$

Since X is an N-by-M matrix with rank of M,

$$\operatorname{rank}(H_2 X) = \operatorname{rank}(H_2).$$

Also, as X_2 is the result of the middle-level PCA model of the subject-level concatenated data, the rows of X_2 are orthogonal, and $\operatorname{rank}(X_2) = N_{g2}$. So based on the above equations we have $\operatorname{rank}(H_2) = \operatorname{rank}(X_2) = N_{g2}$. Similarly, for group 1, we get $\operatorname{rank}(H_1) = \operatorname{rank}(X_1) = N_{g1}$.

Appendix B: Proof of Convergence of Algorithm 2.19

The formula to find the shared components is similar to the regular FastICA algorithm, so the convergence proof of appendix A.1 in Hyvarinen (1999) can be applied. The convergence of equation 2.19 for the case of specific components can be proven under an additional assumption that the number of specific components of group 1 is equal to or greater than $\operatorname{rank}(\operatorname{Null}(H_2))$ (the same is true for group 2). We need to make the following assumption so that the denominator in equation 2.18 becomes nonzero:

$$\mathrm{E}\{s_i g(s_i) - g'(s_i)\} \neq 0, \quad \text{for any } i. \tag{B.1}$$

Inspired by the proof of the original algorithm, if we make the change of variable $\boldsymbol{z} = \boldsymbol{A}^T \boldsymbol{w}$ in equation 2.19 and assume that $\boldsymbol{z}$ is in the neighborhood of a solution with the membership of *SP1*($z_1 \approx 1$), it is easy to show that $\|\tilde{\boldsymbol{a}}_1^{down}\| \ll \|\tilde{\boldsymbol{a}}_1\|$:

$$\tilde{\boldsymbol{a}}_1 = H\tilde{\boldsymbol{w}}_1 = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \tilde{\boldsymbol{w}}_1 = \begin{bmatrix} H_1\tilde{\boldsymbol{w}}_1 \\ H_2\tilde{\boldsymbol{w}}_1 \end{bmatrix} = \begin{bmatrix} H_1\tilde{\boldsymbol{w}}_1 \\ \varepsilon \end{bmatrix}, \tag{B.2}$$

where $\hat{\boldsymbol{w}}_1$ is in the vicinity of $\boldsymbol{w}_1$. Based on the condition in equation 2.23, $s_1 \in SP1$ and equation 2.19 should be used as the update formula. Thus, for each element of vector $\boldsymbol{z}$, we get

$$z_i^+ = \{(I - H_2^T(H_2H_2^T)^{-1}H_2)\boldsymbol{w}_i\}^T \mathrm{E}\{\boldsymbol{x}g(\boldsymbol{z}^T\boldsymbol{s})\} - \mathrm{E}\{g'(\boldsymbol{z}^T\boldsymbol{s})\}z_i. \tag{B.3}$$

Equation B.3 is derived from the fact that matrix $(I - H_2^T(H_2H_2^T)^{-1}H_2)$ is symmetric and $W = A^T$. Based on the constraint in equation 2.13, if $\boldsymbol{w}_i \in$

$SP1$, then $H_2 \boldsymbol{w_i} = \mathbf{0}$. So equation B.3 will be simplified to

$$z_i^+ = \mathrm{E}\{s_i g(\boldsymbol{z}^T \boldsymbol{s})\} - \mathrm{E}\{g'(\boldsymbol{z}^T \boldsymbol{s})\} z_i, \quad \text{for } s_i \in SP1. \tag{B.4}$$

This equation is similar to the one extracted in the original FastICA algorithm, which has been proven to quadratically converge to the correct solution (z_1 converges to 1 and z_i, $i \neq 1$, converges to zero). For the remaining independent components ($\boldsymbol{w_i} \in SP2$ or SH), because they are orthogonal to the subspace spanned by $SP1$, they become orthogonal to the null space of H_2 according to the assumption that $n(SP1) \geq \mathrm{rank}(\mathrm{Null}(H_2))$. Given that the rows of matrix $(I - H_2^T (H_2 H_2^T)^{-1} H_2)$ are in the null space of H_2, we get $(I - H_2^T (H_2 H_2^T)^{-1} H_2) \boldsymbol{w_i} = \mathbf{0}$, so equation B.3 will be simplified to

$$z_i^+ = -\mathrm{E}\{g'(\boldsymbol{z}^T \boldsymbol{s})\} z_i, \quad \text{for } s_i \in SP2, \text{ or } s_i \in SH. \tag{B.5}$$

Applying the condition that $\boldsymbol{z}^* = \boldsymbol{z}^+ / \|\boldsymbol{z}^+\|$, it is clear that z_i (for $s_i \in SP2$, or $s_i \in SH$) linearly converges to zero. Thus, algorithm of equation 2.19 converges to a solution with $z_1 = \pm 1$ and $z_i = 0$ for $i > 1$.

References

Albert, N. B., Robertson, E. M., & Miall, R. C. (2009). The resting human brain and motor learning. *Curr. Biol.*, *19*(12), 1023–1027.

Allen, E. A., Erhardt, E. B., Damaraju, E., Gruner, W., Segall, J. M., Silva, R. F., et al. (2011). A baseline for the multivariate comparison of resting-state networks. *Front. Syst. Neurosci.*, *5*, 2.

Assaf, M., Jagannathan, K., Calhoun, V. D., Miller, L., Stevens, M. C., Sahl, R., et al. (2010). Abnormal functional connectivity of default mode sub-networks in autism spectrum disorder patients. *Neuroimage*, *53*(1), 247–256.

Bartlett, M. S. (2001). *Face image analysis by unsupervised learning*. Boston: Kluwer.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, *7*(6), 1129–1159.

Calhoun, V. D., Adali, T., Pearlson, G. D., & Pekar, J. J. (2001). A method for making group inferences from functional MRI data using independent component analysis. *Hum. Brain Mapp.*, *14*(3), 140–151.

Calhoun, V. D., Liu, J., & Adali, T. (2009). A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data. *Neuroimage*, *45*(1 Suppl.), S163–172.

Cheung, V. C., d'Avella, A., Tresch, M. C., & Bizzi, E. (2005). Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *J. Neurosci.*, *25*(27), 6419–6434.

Damoiseaux, J. S., Rombouts, S. A., Barkhof, F., Scheltens, P., Stam, C. J., Smith, S. M., et al. (2006). Consistent resting-state networks across healthy subjects. *Proc. Natl. Acad. Sci. USA*, *103*(37), 13848–13853.

Dea, J. T., Anderson, M., Allen, E., Calhoun, V. D., & Adali, T. (2011). *IVA for multi-subject FMRI analysis: A comparative study using a new simulation toolbox.* Paper presented at the 2011 IEEE International Workshop on Machine Learning for Signal Processing, Beijing, China.

Erhardt, E. B., Allen, E. A., Wei, Y., Eichele, T., & Calhoun, V. D. (2012). SimTB, a simulation toolbox for fMRI data under a model of spatiotemporal separability. *Neuroimage*, *59*(4), 4160–4167.

Erhardt, E. B., Rachakonda, S., Bedrick, E. J., Allen, E. A., Adali, T., & Calhoun, V. D. (2010). Comparison of multi-subject ICA methods for analysis of fMRI data. *Hum. Brain Mapp*, *32*, 2075–2095.

Fox, M. D., & Raichle, M. E. (2007). Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nat. Rev. Neurosci.*, *8*(9), 700–711.

Hyvarinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural. Netw.*, *10*(3), 626–634.

Hyvarinen, A., & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, *9*(7), 1483–1492.

Hyvarinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Netw.*, *13*(4–5), 411–430.

Kargo, W. J., & Nitz, D. A. (2003). Early skill learning is expressed through selection and tuning of cortically represented muscle synergies. *J. Neurosci.*, *23*(35), 11255–11269.

Lang, S. (1987). *Calculus of several variables* (3rd ed.). New York: Springer-Verlag.

Li, K., Guo, L., Nie, J., Li, G., & Liu, T. (2009). Review of methods for functional brain connectivity detection using fMRI. *Comput. Med. Imaging Graph.*, *33*(2), 131–139.

Li, Y. O., Adali, T., & Calhoun, V. D. (2007). Estimating the number of independent components for functional magnetic resonance imaging data. *Hum. Brain. Mapp.*, *28*(11), 1251–1266.

Ma, L., Narayana, S., Robin, D. A., Fox, P. T., & Xiong, J. (2011). Changes occur in resting state network of motor system during 4 weeks of motor skill learning. *Neuroimage*, *58*(1), 226–233.

Makeig, S., Jung, T. P., Bell, A. J., Ghahremani, D., & Sejnowski, T. J. (1997). Blind separation of auditory event–related brain responses into independent components. *Proc. Natl. Acad. Sci. USA*, *94*(20), 10979–10984.

McKeown, M. J., Jung, T. P., Makeig, S., Brown, G., Kindermann, S. S., Lee, T. W., et al. (1998). Spatially independent activity patterns in functional MRI data during the Stroop color-naming task. *Proc. Natl. Acad. Sci. USA*, *95*(3), 803–810.

McKeown, M. J., Li, J., Huang, X., Lewis, M. M., Rhee, S., Young Truong, K. N., et al. (2007). Local linear discriminant analysis (LLDA) for group and region of interest (ROI)–based fMRI analysis. *Neuroimage*, *37*(3), 855–865.

McKeown, M. J., Makeig, S., Brown, G. G., Jung, T. P., Kindermann, S. S., Bell, A. J., et al. (1998). Analysis of fMRI data by blind separation into independent spatial components. *Hum. Brain Mapp.*, *6*(3), 160–188.

McKeown, M. J., Varadarajan, V., Huettel, S., & McCarthy, G. (2002). Deterministic and stochastic features of fMRI data: Implications for analysis of event-related experiments. *J. Neurosci. Methods*, *118*(2), 103–113.

Palmer, S. J., Li, J., Wang, Z. J., & McKeown, M. J. (2010). Joint amplitude and connectivity compensatory mechanisms in Parkinson's disease. *Neuroscience*, *166*(4), 1110–1118.

Smith, S. M., Jenkinson, M., Woolrich, M. W., Beckmann, C. F., Behrens, T. E., Johansen-Berg, H., et al. (2004). Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage*, *23* (Suppl. 1), S208–219.

Stone, J. V. (2002). Independent component analysis: An introduction. *Trends Cogn. Sci.*, *6*(2), 59–64.

Sui, J., Adali, T., Pearlson, G. D., & Calhoun, V. D. (2009). An ICA-based method for the identification of optimal FMRI features and components using combined group-discriminative techniques. *Neuroimage*, *46*(1), 73–86.

Sui, J., Adali, T., Pearlson, G. D., Clark, V. P., & Calhoun, V. D. (2009). A method for accurate group difference detection by constraining the mixing coefficients in an ICA framework. *Hum. Brain Mapp.*, *30*(9), 2953–2970.

Tresch, M. C., Cheung, V. C., & d'Avella, A. (2006). Matrix factorization algorithms for the identification of muscle synergies: Evaluation on simulated and experimental data sets. *J. Neurophysiol.*, *95*(4), 2199–2212.

Vahdat, S., Darainy, M., Milner, T. E., & Ostry, D. J. (2011). Functionally specific changes in resting-state sensorimotor networks after motor learning. *J. Neurosci.*, *31*(47), 16907–16915.

Vigario, R., Sarela, J., Jousmaki, V., Hamalainen, M., & Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans. Biomed. Eng.*, *47*(5), 589–593.

Viviani, R., Gron, G., & Spitzer, M. (2005). Functional principal component analysis of fMRI data. *Hum. Brain. Mapp.*, *24*(2), 109–129.

Woolrich, M. W., Jbabdi, S., Patenaude, B., Chappell, M., Makni, S., Behrens, T., et al. (2009). Bayesian analysis of neuroimaging data in FSL. *Neuroimage*, *45*(1 Suppl.), S173–S186.

Zuo, X. N., Kelly, C., Adelstein, J. S., Klein, D. F., Castellanos, F. X., & Milham, M. P. (2010). Reliable intrinsic connectivity networks: Test-retest evaluation using ICA and dual regression approach. *Neuroimage*, *49*(3), 2163–2177.

Received December 22, 2011; accepted May 23, 2012.