Published in final edited form as:
Neuroimage. 2014 April 15; 90: 449–468. doi:10.1016/j.neuroimage.2013.11.046.

Automatic Denoising of Functional MRI Data: Combining Independent Component Analysis and Hierarchical Fusion of Classifiers

Gholamreza Salimi-Khorshidi[1,*], **Gwenaëlle Douaud**[1], **Christian F Beckmann**[2,3], **Matthew F Glasser**[4], **Ludovica Griffanti**[1,5,6], and **Stephen M Smith**[1]

[1]Oxford University Centre for Functional MRI of the Brain (FMRIB), Oxford, UK [2]Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands [3]MIRA Institute for Biomedical Technology and Technical Medicine, University of Twente, Enschede, The Netherlands [4]Washington University School of Medicine, Washington University, St. Louis, Missouri, USA [5]Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy [6]MR Laboratory, Fondazione Don Carlo Gnocchi ONLUS, Milan, Italy

Abstract

Many sources of fluctuation contribute to the fMRI signal, and this makes identifying the effects that are truly related to the underlying neuronal activity difficult. Independent component analysis (ICA) - one of the most widely used techniques for the exploratory analysis of fMRI data - has shown to be a powerful technique in identifying various sources of neuronally-related and artefactual fluctuation in fMRI data (both with the application of external stimuli and with the subject “at rest”). ICA decomposes fMRI data into patterns of activity (a set of spatial maps and their corresponding time series) that are statistically independent and add linearly to explain voxel-wise time series. Given the set of ICA components, if the components representing “signal” (brain activity) can be distinguished form the “noise” components (effects of motion, non-neuronal physiology, scanner artefacts and other nuisance sources), the latter can then be removed from the data, providing an effective cleanup of structured noise. Manual classification of components is labour intensive and requires expertise; hence, a fully automatic noise detection algorithm that can reliably detect various types of noise sources (in both task and resting fMRI) is desirable. In this paper, we introduce FIX (“FMRIB’s ICA-based X-noiseifier”), which provides an automatic solution for denoising fMRI data via accurate classification of ICA components. For each ICA component FIX generates a large number of distinct spatial and temporal features, each describing a different aspect of the data (e.g., what proportion of temporal fluctuations are at high frequencies). The set of features is then fed into a multi-level classifier (built around several different Classifiers). Once trained through the hand-classification of a sufficient number of training datasets, the classifier can then automatically classify new datasets. The noise components can then be subtracted from (or regressed out of) the original data, to provide automated cleanup. On conventional resting-state fMRI (rfMRI) single-run datasets, FIX achieved about 95% overall

[*]Corresponding Author Details: Oxford University Centre for Functional MRI of the Brain (FMRIB), Oxford, UK. reza@fmrib.ox.ac.uk, phone: +44 (0) 1865 222726, fax: +44 (0) 1865 222717.

accuracy. On high-quality rfMRI data from the Human Connectome Project, FIX achieves over 99% classification accuracy, and as a result is being used in the default rfMRI processing pipeline for generating HCP connectomes. FIX is publicly available as a plugin for FSL.

1 Introduction

Functional magnetic resonance imaging (fMRI) has become a widely-used approach for mapping brain function. In most fMRI experiments, however, many sources of temporal fluctuation (e.g., head movement, respiratory motion, scanner artifacts, etc.) contribute to the recorded voxel-wise time series. Such artifacts reduce the signal-to-noise ratio, complicate the interpretation of the data, and can mislead statistical analyses (in both subject- and group-level inference) that attempt to investigate neuronally-related brain activation. Thus, separating "signal" from "noise"[1] is a very important challenge in fMRI neuroscience. This is particularly important for resting-state fMRI, because functional networks are identified on the basis of spontaneous correlations between distinct regions, where spatially-extended artefacts can easily contribute problematically to estimated correlations.

There are two major types of noise removal techniques for fMRI datasets - approaches that employ additional physiological recordings (or, "model-based approaches") and those that are data driven (for a detailed review, see Murphy et al., NeuroImage Special Issue on Mapping the Connectome, in press). One of the most well-known techniques of the former type, RETROspective Image CORrection (RETROICOR [Golver et al., 2000]), measures the phases of the cardiac and respiratory cycles, and attempts to remove low-order Fourier terms that are synchronised with these exogenous measurements. Similar approaches are taken in Shmueli et al. [2007] and Birn et al. [2006]: these filter the aspects of the imaging data that demonstrate strong correspondence with the measurements (e.g., in terms of phase or correlation). While these approaches can perform quite well in cleaning respiratory and cardiac noises, their success depends heavily on the availability and quality of the physiological measurements. Moreover, physiological monitoring data, if available/ collected, are not expected to relate to all common forms of artefact (e.g., scanner artefacts and head movements). This is the fundamental reason behind development and adoption of "data-driven" approaches.

Many data-driven approaches employ independent component analysis (ICA), which has been shown to be a powerful tool for separating various sources of fluctuations found in fMRI data. ICA was first used for fMRI by McKeown et al. [1998] for decomposing the data into distinct components (each consisting of a map and its representative time course) that are maximally spatially independent. Some components were considered artefactual, while others reflected the brain's activation in response to the task imposed on the subject. Later, (e.g., Kiviniemi et al. [2003]) it was shown that amongst the structured processes

[1]Throughout this paper we use the terms "artefact" and "noise" interchangeably, in both cases referring to *structured* noise in the data, and not *unstructured* noise (e.g., MRI thermal noise, which in practice in fMRI data is close to being Gaussian and uncorrelated in space and time).

identifiable through ICA, resting-state networks could be found as components distinct from each other and from artefactual effects in the data.

Since ICA requires a large number of samples to function well, its application to fMRI (where there are normally orders of magnitude more voxels than time points) is believed to be more robust in the spatial than the temporal domain. Also, the underlying neural processes in the data may well be more non-Gaussian in space than in time (particularly for resting-state data), adding to the greater robustness of spatial ICA [Smith et al., 2012]. With respect to the separation of activation from artifacts, and of spatially distinct activations from each other, spatial independence has been a successful and enduring model, and nearly all applications of ICA (to both task and resting fMRI) to date have used spatial ICA.

The success of ICA in separating BOLD signal from noise makes it an attractive preprocessing tool for denoising both task and resting fMRI. If ICA can decompose the data into a set of noisy components (i.e., artifactual fluctuations) and non-artefactual components (i.e., fluctuations of interest), one can “clean” the data by subtracting the artefactual components from the data (or regressing them out of the data). However, identifying the artefact components manually can be very labour-intensive, and requires in-depth knowledge of (ideally all possible) signal and noise fluctuations’ spatiotemporal characteristics. Therefore, several previous approaches have attempted to offer fully-automatic solutions to ICA classification. As one of the first attempts, Kochiyama et al. [2005] proposed an automatic solution for removing the effects of task-related motion, which characterises the ICs by their task-related changes in signal intensity and variance; therefore this may be effective for task fMRI, but does not naturally extend to resting-state. Perlbarg et al. [2007] proposed an approach that characterises the activity of the voxels in certain regions of interest (ROIs) that are known *a priori* to correspond to noisy behaviour. Given the wide range of artifacts that can be present in fMRI data, Tohka et al. [2008] proposed a set of 6 spatial and temporal features that capture a wider range of ICs’ characteristics, while [De Martino et al., 2007] defined 11 features. Such features might include the fraction of spatial map supra-threshold voxels lying on the brain edge, or the fraction of temporal spectral power lying above some frequency threshold. In both cases the features were then fed into a trained multivariate classifier, which attempted to automatically classify newly-seen components into signal vs. noise. Our approach is roughly similar, but we defined more than 180 features (including features similar to those defined in the previous papers), and utilise multiple different classifier approaches, combined via classifier stacking.

In this paper, we introduce FIX (FMRIB’s ICA-based X-noiseifier), which is a fully automatic (once hand-trained) solution for cleaning (both task and resting) fMRI data of various types of structured noise. Using FIX consists of five steps: spatial ICA, estimation of a large number of spatial/temporal features for each component of each dataset, classifier training (using hand labelling of components), application of the classifier to new datasets, and denoising (removal of artefact components from the data). In the ICA step, we employ MELODIC (Multivariate Exploratory Linear Optimised Decomposition into Independent Components) [Beckmann and Smith, 2004] from the FMRIB Software Library (FSL[2]). We assessed the performance of FIX against manual component classifications across various

fMRI datasets and found good to excellent performance across a wide range of resting fMRI datasets.

In an associated paper [Griffanti et al., in submission], we have evaluated in detail the effect of ICA+FIX fMRI cleanup on both standard fMRI datasets and accelerated [Feinberg et al., 2010, Moeller et al., 2010] datasets. We also compared the various approaches that one might take to remove the artefactual components from the data once they have been classified as artefact by FIX. These investigations include evaluation (of the effect of the various cleanup options) on both the spatial and temporal (and hence network) characteristics of resting-state networks.

2 Methods

The general approach for applying FIX is:

1. Apply standard preprocessing steps, typically: rigid-body head motion correction, optional spatial smoothing, and high-pass temporal filtering to remove slow drifts.
2. Apply ICA to decompose the preprocessed data into a set of independent components.
3. Use FIX to identify which of the ICA components correspond to artefactual processes in the data.
4. Remove those components from the preprocessed fMRI data.

The spatial smoothing step in the pre-processing might reduce the sensitivity of ICA (and hence FIX) to certain kinds of artefacts and signal. However, in some datasets, the signal-to-noise ratio and amount of data (in particular, number of timepoints) might be sufficiently poor that application of smoothing before running ICA may be helpful overall.

We now include a brief introduction to ICA (the first step of FIX's approach for denoising fMRI data) and then describe FIX's overall architecture, statistical-learning model, set of input features, and hierarchical classifier.

2.1 Independent Component Analysis

We decompose a single run of fMRI space-time data into multiple components using MELODIC [Beckmann and Smith, 2004], built around FastICA [Hyvärinen and Oja, 1997]. This models the data as a linear mixture of different processes, the spatial distributions of which are time-invariant (apart from overall amplitude modulation by the associated timecourse) and statistically independent. ICA assumes the following linear model

$$\mathbf{Y}=\mathbf{AM}+\mathbf{E}, \quad (1)$$

where **Y** is the $T \times V$ matrix of fMRI time series with T time samples and V voxels; **M** is a $K \times V$ matrix of $K \ll T$ spatial components of the independent sources (comprising V voxels

[2] http://www.fmrib.ox.ac.uk/fsl

each) and **A** is the $T \times K$ matrix of the K corresponding time courses (comprising T samples each). **E** is the residuals in the probabilistic ICA model [Beckmann and Smith, 2004], and is assumed to comprise the unstructured noise that dominates the weakest eigenvectors of an initial principal components analysis decomposition applied before the main ICA algorithm.

To reduce the structured noise using ICA, it is necessary to identify the subset of **A** and **M** that demonstrate artefactual behaviour temporally and/or spatially. Having found such a subset, one can clean the data by (for example) regressing the set of artefactual time courses $\mathbf{A_b}$ out of the original data, or by taking the product of artefact time courses and spatial maps $\mathbf{A_b M_b}$ and subtracting that from the data. For detailed investigations of different methods for regressing the artefactual components out of the data, see [Griffanti et al., in submission].

2.1.1 Example Good and Bad ICA Components—We now show several example "good" and "bad" ICA components from typical fMRI datasets, primarily in order to help clarify the following descriptions of FIX's spatial and temporal features.

Independent components were manually labelled into different classes - primarily "good" (for signal) and "bad" (for noise). Components which could not be unambiguously identified as good or bad were labelled as "unknown"; in such cases, FIX treats these components as "good", as the desired final behaviour is generally to be conservative with respect to minimising the chance of incorrectly removing valid neuronal signal. When possible, the "noise" components were further sub-classified as: movement-related, white matter "signal", interaction between susceptibility artefacts and head motion, cardiac pulsation/arterial contribution, large veins, or MRI acquisition-related issues (although to date FIX does not make use of these sub-categorisations). The manual identification of each component was carried out by first looking at the thresholded spatial map (typically abs(Z)>2.3 [3]), then at the temporal power spectrum, and finally at the time series. When necessary, the spatial map of the component was viewed unthresholded.

The examples are shown as viewed by the "Melview" program written specifically to display and hand-classify ICA components for FIX training. The list of components (and their assigned classifications) appears on the right, and, for the currently selected component, the spatial map, temporal power spectrum and time course are displayed.

Figure 1 illustrates an example components identified as good for 3 different types of acquisition: (A) 3×3×3.5 mm resolution, TR = 2s, smoothed with a 5mm full-width-half-maximum (FWHM) Gaussian kernel; (B) 3×3×3 mm resolution, TR = 3s, smoothed with a 5mm (FWHM) Gaussian kernel, automatic estimation of the number of ICs; C: 1.6×1.6×1.6 mm resolution, TR = 1.11s, multi-band acceleration factor = 6, unsmoothed spatially.

Figure 2 shows example movement-related bad components. Figure 3 demonstrates how two more noise components (respectively, white matter fluctuations and susceptibility-related artefact) are clearly artefact as judged spatially, though the spectrum of the second example

[3]Voxel-wise Z-statistics are derived from standardising the spatial maps' initial voxel-wise statistics by their corresponding residual's standard deviation (more details in Section 2.1 and Beckmann and Smith [2004])

does not look very strongly artefactual. Figure 4 shows example cardiac pulsation (artery) bad components, identified in the CSF in the ventricles in one case, and anatomically following arteries (most commonly around the posterior cerebral artery and middle cerebral branches) in the other. Figure 5 shows components relating to major veins - in these cases, the sagittal sinus vein. Vein components tend to have similar temporal characteristics (including power spectra) to those of good components. Figure 6 shows two examples of MRI acquisition/reconstruction related artefacts - they do not look like artefacts arising directly from any aspect of physiology. Figure 7 shows two example “unknown” components, which do not look like clean neuronally-related signal, but may contain some aspects of it.

2.2 Features

Probably the most important element of multivariate classification is the extraction of an effective set of features to feed into the core classifier. A set of fairly independent features, each often correlating well with the target variable (or class) will make the learning/ classification task easy; on the other hand, if the class is a very complex function of the features, learning may become difficult. FIX uses over 180 features, capturing components’ spatial and temporal characteristics.

Temporal features capture the dynamics of an IC time series (denoted by **a**), and spatial features model various characteristics of an IC’s spatial map (denoted by **m**). The first feature, however, is the number of ICs as determined by MELODIC and is therefore considered “spatio-temporal”; the presence and extent of various noise types in the data is expected to affect the number of independent components. Thus, f_1 may be a contributing factor to or a predictor of the likelihood of an IC being signal or noise. The rest of the features are classed as temporal or spatial, and are described below.

2.2.1 Temporal Features

Autoregressive (AR) Properties: Temporal smoothness, which can be estimated by fitting AR(n) models to a component’s time series, is expected to help differentiate signal from certain artefacts. Let $c_{1,1}$ denote the parameter of an AR(1) model, $c_{1,2}$ and $c_{2,2}$ denote the parameters of an AR(2) model, v_p denotes the variance of the residual of AR(p) models up to order p=6. The first AR-based features are the slope and intercept of the straight line [4] that explains v as a function of p (increasing AR model order will result in a better fit and hence a smaller residual variance). The extent of such improvement in goodness of fit decreases as the extent of noise in the time series increases (e.g., in case of white Gaussian noise, no meaningful improvement is expected). Thus, these features are expected to be help separate signal and noise components.

This is a valid point, in that . However, what this feature measures is the extent of deviation of the trend (as we increase the order of the AR model) from a straight line; more deviation meaning more signal. In case of linear relationship, this feature will perfectly capture the

[4]In case of a nonlinear (e.g., logarithmic) relationship between the order of the AR model and the goodness of fit, this feature is still valid for capturing the direction of this relationship.

extent of signal; otherwise (e.g., for a logarithmic relationship) this feature will still capture the part of the story. Moreover, the data supports the validity of this statement as it is a fairly discriminant feature.

The next AR-based features are simply $c_{1,1}$, $c_{1,2}$, $c_{2,2}$, v_1 and v_2. These features capture the extent of autocorrelation and the power of uncorrelated noise as estimated from the lowest order AR models. In general, signal components are expected to have higher temporal autocorrelation and smaller residual variance, compared to *unstructured* noise components.

An Ornstein-Uhlenbeck process can be considered as the continuous-time analogue of the discrete-time AR(1) process and hence has similar properties. An Ornstein-Uhlenbeck process $\mathbf{a}_t$ satisfies $d\mathbf{a}_t = \theta(\mu - \mathbf{a}_t)dt + \sigma dW_t$, where $\theta > 0$ and $\sigma > 0$ denote the speed of mean reversion and volatility, respectively, and W_t denotes a Wiener process. Overall, signal components are expected to have smoother (i.e., slower mean reversion) and less volatile dynamics than noise components. Thus, including θ and σ in FIX's feature base is expected to boost its discriminant power.

Distributional Properties: While signal components' time series are expected to have *fairly* normal distributions, noise components can be, for instance, bimodal (e.g., due to scanner artefacts) or have long-tailed distributions (e.g., sharp peaks in the time series that are due to rapid head movements). Distributional features can summarise the shape of a time series' distribution (e.g., as measured via a histogram), in terms of its mean, median, tail, etc., and aid FIX in detecting signal vs. noise. Thus, the next features correspond to the time series' kurtosis (considering the width of peak and tail weight for the distribution), skewness (measuring the asymmetry of the distribution), mean-median difference (another indicator of the asymmetry), entropy ($-\Sigma_i p_i \log p_i$, another measure of "information content") and negentropy ($\text{mean}(\mathbf{a}^3)^2/12+\text{kurtosis}(\mathbf{a})^2$, which quantifies the extent of normality).

Jump Amplitudes: The extent of jumps (or, sudden changes) in time series' amplitude are important characteristics of components; while signal time series are expected to be fairly smooth, large jumps can be present in noise components' time series (e.g., due to motion, or scanner artefact). Thus, FIX's next features reffect the extent of such properties:

$$\begin{gathered}\max(|\tfrac{d\mathbf{a}}{dt}|)/\text{std}(\mathbf{a}) \\ \max(|\tfrac{d\mathbf{a}}{dt}|)/\text{std}(\tfrac{d\mathbf{a}}{dt}) \\ \text{mean}(|\tfrac{d\mathbf{a}}{dt}|)/\text{std}(\mathbf{a}) \\ \max(|\tfrac{d\mathbf{a}}{dt}|)/\text{mean}(\mathbf{a}_{sub}) \\ \max(|\tfrac{d\mathbf{a}}{dt}|)/\text{sum}(\mathbf{a}_{sub}),\end{gathered} \quad (2)$$

where std(.) denotes standard deviation and $\mathbf{a}_{sub}$ is $\mathbf{a}$ after excluding the largest jump's vicinity (a window of 5 time points). These features are different from each other by virtue of different measures of normalisation. The last of these features has already been found useful for detecting noise ICs and recommended by Tohka et al. [2008].

Fourier transform: The Fourier transform can attempt to distinguish signal components from noise components in terms of the distribution of their power in the frequency domain. Typically, signal time series are expected to have higher content/power in low frequencies

and almost no content in high frequencies (because of the smoothing effect of the haemodynamics on the fMRI signal), whereas noise time series may have content anywhere (or even "everywhere") in the frequency range. Therefore, FIX's next set of features is derived from the fast Fourier transform (FFT) applied to the time series. The first group of FFT-based features are quite coarse, in that they are the ratio of total power above a given frequency to the power below that frequency, with several different frequency thresholds (one per new feature): 0.1, 0.15, 0.2 and 0.25 Hz. The second group of FFT-based features are finer evaluations of the power spectrum; they measure the percent of total power that falls in each frequency bin, for the binned frequency ranges: 0:0.01, 0.01:0.025, 0.025:0.05, 0.05:0.1, 0.1:0.15, 0.15:0.2 and 0.2:0.25 Hz. Clearly for datasets with longer TR, some of these frequency bins will not be relevant.

A further frequency-based set of features is derived by assuming that in signal components, the neural signals take the form of a flat power spectrum [Niazy et al., 2011]. Therefore convolving a canonical haemodynamic response with a white-noise neural signal results in a sample from such a model. Similarity of a given components spectrum to such simulated spectra will decrease its likelihood of being noise. We therefore compare the actual power spectrum with the mean spectrum generated under the assumption of pure neural signal, and generate a new set of features, where each feature quantitates how different these two spectra are, for one of 7 frequency bins in the spectra. Assume that $\mathbf{p}$ is the vector of aforementioned FFT-based "fine" features for a given component. One can derive $\mathbf{p}$'s equivalent under the "neural noise" hypothesis by simulating [5] 100 time series, and averaging and binning the resulting spectra. That is, given the simulated time series (i.e., $\mathbf{a}^{(i)}$, where i =1, 2, …, 100), we extract their corresponding $\mathbf{p}$ vectors (i.e., $\mathbf{p}^{(i)}$, where i =1, 2, …, 100) and average them (denoted by $\mathbf{p_0} = \Sigma_i \mathbf{p}^{(i)}/100$). The new FFT-based features are derived by comparing $\mathbf{p}$ and $\mathbf{p_0}$ vectors and calculating the sum of standardised errors (i.e., $\Sigma_{bins}((\mathbf{p} - \mathbf{p_0})^2/\mathbf{p}^2)$) as well as the vector of squared standardised errors (i.e., $(\mathbf{p} - \mathbf{p_0})^2/\mathbf{p}^2$).

Correlation: Correlation of a time series with other reference time series (e.g., head motion) is the basis of FIX's next set of temporal features.

Functional time series (i.e., signal fluctuations) are strongly associated with the brain's grey matter (GM), while fluctuations in white matter (WM) and cerebrospinal fluid (CSF) are mostly associated with artefacts. In order to quantify a time series' association with each of these tissue types, FIX's next of features is derived from the time series' correlation with GM-, WM- and CSF-derived time series. In order to extract these reference time series, WM, GM and CSF masks are extracted using FSL's tissue-type segmentation tool (FAST) [Zhang et al., 2001]. Each tissue type's reference time series is simply the average of all time series that correspond to voxels that belong to that tissue type. If we have available a high-quality structural image (such as with HCP data), which has already been pre-processed by FreeSurfer cortical/subcortical modelling, we utilise tissue-type segmentations from that instead of using FAST.

[5] In order to simulate such time series, we assume a Gamma($(\delta/\sigma)^2$, δ/σ^2) HRF, where δ =6/TR and $\sigma = \delta/2$. Next, we convolve the HRF with a white noise "neural" signal, which is a vector of white Gaussian noise with a length equal to the real data time series.

The next set of correlation-based features employ head motion time series. We take the 6 rigid-body head motion parameters as estimated by the head motion correction applied in the data pre-processing (3 translations and 3 rotations), resulting in 6 timeseries (i.e., 6 parameters per time point). We also take the backwards-looking temporal derivatives (resulting in 6 further timeseries), and then the squares of all 12 timeseries (resulting in a further 12). We derive several new features by correlating the IC time series with each of these 24 motion parameter time series. From correlating the IC time series with the 24 motion time series, we derive 24 new features, and also add further features that summarise the maximum of the first 6, the maximum of the remaining 18, and the maximum of all 24. We also *regress* the IC time series against the 24 motion time series, take the magnitude of the 24 regression parameters, and add new features corresponding to the largest two of these, and also the average of all 24.

2.2.2 Spatial Features—In order to extract some of the spatial features, spatial maps are required to be processed/transformed first. Table 1 shows these transformations and their corresponding abbreviations that the rest of this section will refer to. To date, the threshold τ for components' maps has been fixed at 2.5 (note that MELODIC-generated ICA spatial maps are in "units" of Z-statistics).

Clusters' Sizes and Spatial Distribution: The distribution of the activation and deactivation cluster-sizes are useful indicators of the extent of noise in components. For instance, a signal component might be expected to have a relatively small number of *fairly* large clusters, whereas some types of artefact component are expected to have a large number of smaller clusters. We form **c**, to be a list of a spatial map's cluster (i.e., connected components that survive cluster-forming threshold τ) sizes (in mm^3), sorted in descending order, excluding clusters smaller than 5 voxels. Features that summarise **c** are length(**c**), mean(**c**)-median(**c**), max(**c**), var(**c**), skewness(**c**), kurtosis(**c**), **c**[1], **c**[2], and **c**[3] (i.e., the first third elements of **c**).

An alternative way of looking at the spatial distribution of clusters can help detect the presence of scanner noise (e.g., rapid movements when the acquisition is interleaved). Assume that $\mathbf{v} = [v_1, v_2, \ldots, v_n]$ and $\mathbf{u} = [u_1, u_2, \ldots, u_n]$ contain n slice-specific measures derived from $\mathbf{m}$ and $\mathbf{m}_p^{\tau}$, respectively, where v_i and u_i denote the percent of total variance that falls in the ith slice in $\mathbf{m}$ and $\mathbf{m}_p^{\tau}$, respectively. FIX's next features are max(**v**) and max(**u**), as well as the number of slices that have more than 15% of total variance of $\mathbf{m}$ and $\mathbf{m}_p^{\tau}$. These features will detect the presence of slices that contain a high percentage of component maps' total variance (see Figure 6). It is also likely for neighbouring slices to contain a high percentage of maps' total variance in signal components, whereas some noise components may have most of the signal in just the odd or just the even slices. Thus, next features attempt to distinguish such cases by calculating the difference between the percentage of variance that is explained by even and odd slices (for both $\mathbf{m}$ and $\mathbf{m}_p^{\tau}$) and also the difference between the percentage of variance that is explained by slices [1,2,5,6,9,10, …] and [3,4,7,8,11,12, …] (for both $\mathbf{m}$ and $\mathbf{m}_p^{\tau}$).

Another useful property of spatial maps/distributions is that it is quite unlikely for signal components to have strong presence of both activation and deactivation in their spatial maps and hence the presence of such patterns can be an indicator of noise. In order to assess the strength of this property in components, FIX devises an alternative approach in looking at the spatial distribution of statistics that measures and compares summary statistics such as mean and SD of intensity for **m** and $\mathbf{m}_a$. If m and s denote the mean and SD of nonzero voxels in **m**, m_a and s_a denote the mean and SD of nonzero voxels in $\mathbf{m}_a$, $z = m/s$, $z_a = m_a/s_a$, and e and e_a denote the entropies of nonzero voxels of **m** and $\mathbf{m}_a$. The next set of features are e and e_a (measuring the randomness in voxel-wise distributions), z (measuring how the overall distribution of voxel-wise statistics differs from zero, i.e., whether positive and negative voxels are equally present), z/z_a (for a negative or positive image, this number is expected to be very close to −1 or 1, whereas for an image that has both positive and negative present, this will be close to 0), $1-\mathrm{sum}(\mathbf{m}_n^b)/\mathrm{sum}(\mathbf{m}_p^b)$ and $1-\mathrm{sum}(\mathbf{m}_n^{\tau,b})/\mathrm{sum}(\mathbf{m}_p^{\tau,b})$.

Voxels overlaying bright/dark raw data voxels: For the next set of features we multiply point-wise the ICA spatial maps (Z-statistics) by the mean (across time) pre-processed fMRI time series data, because there can be intensity information in the mean data that indicates whether voxels are grey matter vs. (e.g.) large blood vessels. We also *divide* point-wise, generating a second statistical image. The 4 new features generated are the 95th and 99th percentiles of these two new images.

Percent on Brain Boundary: High overlap between the brain's boundary and a component's spatial map indicates that a component is probably motion-related [Tohka et al., 2008]. In order for FIX's features to capture such instances, first, the brain mask is extracted using FSL's brain extraction tool (BET)[Smith, 2002]. Subtracting this mask from an eroded version of itself results in an "edge mask". Given the variation in the extent of head-motion noise, FIX employs 5 edge masks, ranging from very thin/conservative (the mask minus its once-eroded version) to very thick/liberal (the mask minus its five-times-eroded version). Extracting the following features for each of the 5 masks results in the next set of 3×5=15 features:

$$\begin{array}{c}\mathrm{sum}(\mathbf{m}.*S)/\mathrm{sum}(\mathbf{m})\\ \mathrm{sum}(\mathbf{m}.*S)/\mathrm{sum}(S)\\ \mathrm{sum}(\mathbf{m}_p^{\tau,b}.*S)/\mathrm{sum}(\mathbf{m}_p^{\tau,b}),\end{array} \quad (3)$$

where sum function adds the values of all the voxels, S denotes the edge mask and .* denotes element-wise multiplication. These features measure what percent of an IC spatial map's mass and size fall on the brain intensity edges and what percent of edges is covered by the IC; higher values in these features corresponds to higher likelihood of the IC being noise (see the example at the bottom of Figure 2).

Mask-based Features: Using spatially-specific masks may the only solution for detecting some noise components that have signal-like spatio-temporal characteristics (as defined by other features) and are located in brain regions such as Sagittal Sinus, CSF, and WM. For

example, signal in major veins may look temporally like valid signal components, and may have similar cluster-like spatial characteristics (see Figure 5). FIX employs 3 hand-created standard-space masks, each comprising a distinct set of major vein voxels. They are transformed from standard-space into the subjects' native space, before being used by FIX feature extraction. Given the subject-to-subject anatomical variability, three masks are derived from each of these three masks: The first mask is the most conservative (i.e., the smallest/thinnest) and the last mask is the most liberal (i.e., the largest/thickest) one. For each of the 9 masks, features are extracted based on Equation 3, except that here, S denotes the different "vein" masks.

In the case of datasets where we have suitable structural images to derive subject-specific major vein masks, we utilise these instead of the standard-space masks. For example, from HCP data, we divide the T1-weighted structural image by the co-registered T2-weighted image; this enhances major veins strongly [Glasser and Van Essen, 2011]. The resulting image is thresholded, masked by a dilated standard-space vein mask (to add robustness to the whole process) and finally transformed into the space of the native fMRI data, before being used as vein masks as described above.

BOLD signal is expected to be found within the GM. The second group of mask-based features therefore employ tissue-type masks, since a component's overlap with these tissue types is a strong predictor of it being signal or noise. Having extracted these masks (as described in Section 2.2.1), features are extracted based on Equation 3, where S denotes the (WM, GM and CSF) masks.

Other Spatial Features: FIX uses other spatial features that are measures of an IC's map's smoothness and its TFCE (threshold-free cluster enhancement) [Smith and Nichols, 2009] statistics. It is expected that spatial maps of signal components are "smooth", i.e., a fairly small number of connected components (clusters). Some noise components, on the other hand, are expected to have a "rough" spatial map, i.e., a fairly large number of small clusters, or a patchy spatial map. In this study, smoothness of a spatial map is calculated using random field theory (as described by Salimi-Khorshidi et al. [2010]). As a result, 2 new features are the spatial smoothness in *mm* and voxels counts.

Despite the importance of cluster-size statistics in separating signal and noise components, signal-related clusters in an image are not solely defined by their extent; such clusters can also be associated with fairly high peaks. Low sensitivity of traditional cluster-based methods to the latter type of signals justifies the use of TFCE statistics, which has shown better sensitivity in detecting signals of various characteristics [Smith and Nichols, 2009]. In order improve FIX's ability in separating signal and noise, its next features are maximum TFCE statistics for $\mathbf{m}$, $\mathbf{m}_a$ and standardised $\mathbf{m}$ (i.e., $\mathbf{m}$ image divided by its SD).

The last spatial feature detects (high-spatial-frequency) "stripy" patterns of alternating positive and negative values in the spatial maps. In order to detect the presence of such a pattern, first, $\mathbf{m}$ and $\mathbf{m}_a$ are both smoothed (σ=2mm). In the presence of such a pattern these two images are expected to be very different from each other after smoothing, and hence are smoothed and then compared to define a further feature for this type of noise.

Finally, additional features added are image-acquisition parameters, i.e., spatial and temporal resolutions, and the size of the image data in the x, y, z and t dimensions. Clearly these features do not discriminate between different components within a given dataset (as they are the same for all components), but may help normalise other features when combined with those inside Classifiers, if datasets with different acquisition parameters are combined for training/classification by FIX.

2.3 Feature Selection

Feature selection attempts to automatically choose a subset of relevant features for building robust learning models. It is of particular importance where there are low number of data samples, each summarised with a large number of features. By removing most irrelevant and redundant features, feature selection helps improve the performance of many learning models by alleviating the effect of the curse of dimensionality, enhancing generalisation capability, speeding up learning process and in some cases improving model interpretability. FIX employs a combination of F-score, logistic regression and a linear support vector machine (SVM) for feature selection.

For a given feature *j*, the F-score is calculated as

$$F_j = \frac{\left(\overline{\mathbf{x}}_j^S - \overline{\mathbf{x}}_j\right)^2 + \left(\overline{\mathbf{x}}_j^N - \overline{\mathbf{x}}_j\right)^2}{\frac{1}{n_S - 1}\sum_{i=1}^{n_S}\left(\mathbf{x}_{i,j}^S - \overline{\mathbf{x}}_j^S\right)^2 + \frac{1}{n_N - 1}\sum_{i=1}^{n_N}\left(\mathbf{x}_{i,j}^N - \overline{\mathbf{x}}_j^N\right)^2}, \quad (4)$$

where n_S and n_N are the number of signal and noise data samples (from the total number of ICA components across all runs in the training data); $\overline{\mathbf{x}}_j$, $\overline{\mathbf{x}}_j^S$ and $\overline{\mathbf{x}}_j^N$ denote the average of the *j*th feature across the whole training dataset, and across the signal-only and noise-only components; and $\mathbf{x}_{i,j}^S / \mathbf{x}_{i,j}^N$ is the *j*th feature of the *i*th signal/noise component. The numerator denotes the inter-class variance, while the denominator is the sum of the variance within each class; a feature with a relatively large F-score is expected to have a relatively high signal vs. noise discriminant power. A criticism of the use of the F-score (despite its simplicity and effectiveness) in this context is that it considers each feature separately and therefore cannot reveal information shared across features. Thus, we also considered the feature ranking that is provided by logistic regression and linear SVM.

Logistic regression is widely used as a classification technique, modelling the outcome/classification as a linear combination of features. As described in detail in Appendix A.2, a given feature's coefficient in the linear model has its corresponding significance score, or P-value, which denotes its importance in prediction. FIX fits univariate logistic regression (i.e., one feature at a time) and uses the resulting $-\log_{10}$ of each feature's P-value as a score for feature ranking. The multivariate feature ranking technique in FIX is based on a linear SVM model (see Appendix A.3.3 for details). Similar to the previous two, this approach results in a vector of scores (one per feature) that can be the basis of importance-based ranking of features.

Assume that $\mathcal{L}_{\mathrm{F}}$, $\mathcal{L}_{\mathrm{LR}}$, and $\mathcal{L}_{\mathrm{SVM}}$, denote the rankings resulting from F-score, logistic regression and linear SVM, respectively. FIX aggregates the top-ranking features from these three rankings and decides on the final subset of features; If a feature is among the top 50% of at least one of the three rankings (i.e., $\mathcal{L}_{\mathrm{F}}$, $\mathcal{L}_{\mathrm{LR}}$, and $\mathcal{L}_{\mathrm{SVM}}$), then it will pass FIX's feature selection filter (see Section 2.4 for the place of feature selection in FIX's hierarchy).

2.4 Hierarchical Classifier

Assuming that MELODIC's output consists of components that are either purely signal or purely noise, FIX aims to detect the noise components and clean the fMRI data accordingly. In reality, however, it is quite likely that such components are not pure and instead consist of a mixture of signal and noise. On the other hand, an important criterion for FIX's success is to clean the fMRI data from noise, while preventing, or more realistically, minimising the loss of signal.

"Impurity" of components and the fundamental differences across the various types of artefact cause heterogeneity across different noise components' characteristics. Therefore, in a classification context, the decision boundary that separates noise from signal is expended to be a complex one. In other words, it is quite likely that in the N-dimensional feature space, signal and noise components are not two simple clusters and hence not trivially separable. Additionally, when manually classifying the components, experts tend to consider the components' spatial maps and time series separately, and then implicitly follow multiple *if-then* rules that determine the final label. This shows the complexity of the decision boundary, which consists of collecting and combining evidence in spatial and temporal domains and feeding them through a complex decision-making process. In order to learn such a multi-criteria decision process, FIX employs an ensemble learning (or classifier fusion) approach.

Assume that $\mathcal{D} = \mathcal{D}_a \cup \mathcal{D}_m$ is the dataset containing the full set of features, where $\mathcal{D}_a$ and $\mathcal{D}_m$ denote the temporal and spatial subsets of features. Applying feature selection (see Section 2.3) to $\mathcal{D}$ results in $\mathcal{D}_{sel} \subset \mathcal{D}$, which can consist of both temporal and spatial features (denoted by $\mathcal{D}_{a-sel}$ and $\mathcal{D}_{m-sel}$ respectively). This process results in 6 different datasets ($\mathcal{D}$, $\mathcal{D}_m$, $\mathcal{D}_a$, $\mathcal{D}_{sel}$, $\mathcal{D}_{m-sel}$, and $\mathcal{D}_{a-sel}$) that one can train a given classifier on (note that, all these datasets have a column that contains the components' labels, i.e., signal or noise). Using subsets of features can make the detection of signal/noise easier, as there are components that show their signal fluctuations only in spatial, temporal or other subset of features. In order to achieve this detection in a classification setting, however, there is no absolute best classifier; *k*-NN (described in Appendix A.1) is a reliable local classifier, but cannot capture the patterns that exists in the full dataset; decision trees (described in Appendix A.4) are very good at learning complex decision boundaries that can be represented as a series of *if-then* rules; support vector machines (described in Appendix A.3) are very well capable of (applying the kernel trick and) finding decision boundaries with maximum margin/ generalisation. Plus, the empirical comparisons of Classifiers showed that "the best learner" varies from application to application.

In order to compensate for one classifier's weakness through other Classifiers' strengths, ensemble learning (also known as classifier fusion) has been proposed - learners that

combine multiple Classifiers [Galar et al., 2012, Wolpert, 1992]. Here, instead of fine-tuning and choosing a single best classifier, one combines variations of multiple Classifiers in order to improve the final results. FIX utilises an ensemble technique known as *stacking* [Wolpert, 1992], where outputs of individual Classifiers become the inputs of a "higher-level" learner (in FIX's case, we tested decision tree, random forest, linear SVM and SVM with RBF kernel) that works out the best way of combining them (see Figure 8). The mathematical details of these Classifiers can be found in Appendix A. Training the ensemble consists of extracting $\mathcal{D}$, $\mathcal{D}_{set}$, $\mathcal{D}_a$, $\mathcal{D}_m$, $\mathcal{D}_{a-set}$ and $\mathcal{D}_{m-set}$, training the k-NN, decision tree and SVMs, on each of the datasets, and training the fusion-layer classifier using the lower-level Classifiers' probabilistic outputs [Dzeroski and Zenko, 2004].

2.4.1 Learning and Generalisation—The fundamental goal of machine learning is to generalise beyond the examples in the training set. This is because, no matter how much data we have, it is very unlikely that we will see those exact examples again in future datasets. Performing well on the training set can be easy (the classifier can simply memorise the examples), and can create the illusion of success. Hence, when training and/or evaluating a learner algorithm, one must devise a strategy to minimise the risk of over-fitting (i.e., effectively memorising the examples).

In this study, FIX is tested using a leave-one-out (LOO) approach across sets of ICA output components. If the training data consists of n MELODIC outputs (e.g., one per imaged subject), each fold of the cross-validation uses $n - 1$ datasets for training, and tests the learned decision boundary on the left-out dataset. In the case of having multiple runs of data from each subject (such as with the rfMRI data from the Human Connectome Project, with 4 15-minute runs per subject), it is safest to leave out all runs for a given subject and train on all datasets of all other subjects, in order to be able to generalise LOO accuracy results that will validly describe the expected classifier performance when applied to future subjects.

2.5 Performance Indices

FIX's performance can be summarised by its accuracy in detecting signal and noise components in comparison with labels as provided by experts. We characterise classification accuracy in terms of two measures of success: TPR ("true positive rate", meaning the percentage of true signal components correctly detected) and TNR ("true negative rate", meaning the percentage of true artefact components correctly detected). We can also average the two measures to give an overall "accuracy", although this is not on its own generally as meaningful as the two separate measures.

Given that FIX's output is a probability, a threshold is applied to determine the binary classification of any given component. Changing the threshold shifts the balance between TPR and TNR; lowering it increases the TPR and decreases the TNR. For the LOO accuracy testing, therefore, we can evaluate several thresholds in order to show how the balance between prioritising TPR vs. TNR can be varied. We are not concerned with over-fitting relating to testing several thresholds, as the TPR/TNR curves (as a function of threshold) are slowly-varying (and generally objectively shallow) monotonic functions of the threshold, and we only tested a few different values.

3 Results

Example results showing several different kinds of ICA components from a range of fMRI acquisition protocols have been presented above (Figures 1 to 7). In this section we present quantitative results relating to the accuracy of FIX in correctly classifying ICA components as signal vs. noise. As discussed above, the evaluation of optimal methods for removal of noise components (once identified by FIX), and investigation of the effects of this removal on resting-state spatial maps and network modelling, is outside the scope of this paper, and is covered in a separate followup paper [Griffanti et al., in submission].

3.1 High quality rfMRI data from the Human Connectome Project

The 3T rfMRI data being acquired in the Human Connectome Project has fairly high spatial and temporal resolution (2×2×2mm, 0.73s), utilising a multiband acceleration of ×8 [Ugurbil et al., 2013], and is acquired in relatively long runs (15 minutes) [Van Essen et al., 2013, Smith et al., 2013]. 100 runs, from 25 subjects (ages 22–35y, 17 females), were hand-labelled. We trained FIX and used leave-one-subject-out testing to evaluate classification accuracy. As described above, the final FIX classification threshold (which has arbitrary units) can be varied to change the balance between true-positive-rate (accuracy in classifying good components) and true-negative-rate (accuracy in classifying bad components).

The LOO results for a range of thresholds are shown in Table 2. From these subjects a good choice of threshold would seem to be 10, which results in a **mean TPR and TNR of 99.6 and 98.9 percent (median values across subjects of 100 and 99.2)**. We also hand-labelled ICA components from a single run from each of 20 further subjects (10 females), acquired several months later than the original subjects, also as part of of the ongoing HCP acquisition of data from over 1000 subjects. We applied FIX using the original training from the 100 runs described above. This was partly carried out to confirm that FIX was working well using the original FIX training, when applied to later acquisitions, and when the ICA components were generated by a very slightly improved version of the MELODIC ICA code (one would hope that minor changes in the data and/or the ICA program would not invalidate FIX training). **Across all components from these new 20 runs, at a FIX threshold of 5, FIX achieved TPR and TNR of 99.7 and 99.6, i.e., even better than the original LOO results.** FIX has been implemented in the HCP processing pipeline, and future rfMRI data will be made available with FIX cleanup already applied (as well as being made available without the cleanup).

Using data from 131 HCP subjects, the full set of FIX features was evaluated using principal component analysis (PCA) to see how much redundancy there is across the features. Concatenating the feature vectors from all components from all subjects' ICA decompositions resulted in 53690 feature vectors. Each feature in this concatenated feature matrix was normalised across subjects to zero mean and unit standard deviation, and the matrix was then fed into PCA. The eigenvalues showed that 36 eigenvectors are required to explain 95% of the variance in the full feature set from all subjects (with 67 eigenvectors required to explain 99% of the variance, and 99 required to explain 99.9%). From this we conclude that there is some redundancy in the full set of 185 features, but that a *much*

smaller number of features would not carry all of the information that is made available to FIX's classifier.

The aforementioned level of redundancy among features might suggest the adoption of a "global" feature selection for minimising the number of features (columns) in $\mathcal{D}$ and its subsets (e.g., $\mathcal{D}_m$ and $\mathcal{D}_{m-set}$), and hence a more optimal computation pipeline. In FIX, however, we do not advocate this approach for the following reason. Unlike noise types such as head motion, that are commonly observed in most datasets, and can sometimes have multiple components in a given dataset's ICA decomposition, there are some component types that occur much more rarely (e.g., certain types of scanner artefact, such as those with high-spatial-frequency stripy patterns of alternating positive and negative values in the spatial maps). The FIX feature set is specifically designed to detect both common and rare noise types. Employing feature selection techniques that aim to minimise the number of features while having minimal (but not zero) impact on the overall error would probably result in the exclusion/elimination of such features that are specific to low-occurrence noises. Thus, we utilise a solution that does not fully exclude any of the features from the very beginning.

3.2 Results from more standard datasets / scanners

Standard EPI acquisition, single study dataset with study-specific FIX training—We analysed rfMRI data from 45 subjects (ages 63–75y, 33% female) acquired using a single protocol on a Siemens 3T Verio using standard EPI (3×3×3mm, 3s, 10 minutes). ICA components from all subjects were hand-labelled, and used to train FIX, with accuracy evaluated using LOO testing. The mean (median) across-left-out-subject TPR & TNR results at a threshold of 10 were 96.2 & 95.1 (100 & 92.2). The average number of ICA components estimated by MELODIC was 70.7 per subject, and the average number of (hand-labelled) signal components was 8.8; hence these results mean that on average only 0.3 good components are misclassified as bad per subject (or, put another way, on average, two out of three subjects have no good-component misclassifications, with the third having a single one).

Standard EPI, different protocols and scanners combined—We combined the above dataset with 61 further subjects' datasets from a range of other 3T studies using a range of acquisition protocols. ICA components from all subjects were hand-labelled, and used to train FIX, with accuracy evaluated using LOO testing. The mean (median) TPR & TNR results at a threshold of 20 were 96.1 & 86.0 (100 & 91.5).

Multiband-accelerated EPI from a standard 3T clinical scanner, single study dataset with study-specific FIX training—We analysed rfMRI data from 25 subjects (ages 63–75y, 33% female) acquired on a standard Siemens Verio with a 32-channel head coil (2×2×2mm, 1.3s, multiband ×6, 10 minutes). ICA components from all subjects were hand-labelled, and used to train FIX, with accuracy evaluated using LOO testing. The mean (median) TPR & TNR results at a threshold of 20 were 97.9 & 96.8 (100 & 98.8).

Combined multiband EPI data from a standard 3T clinical scanner and pilot data from the HCP "Connectome Skyra"—We combined the above dataset with early HCP pilot datasets from 14 subjects (ages 18–30y, 50% female) acquired on the Siemens "Connectome Skyra" (2×2×2mm, 1.37s, multiband ×4, 23 minutes). ICA components from all subjects were hand-labelled, and used to train FIX, with accuracy evaluated using LOO testing. The mean (median) TPR & TNR results at a threshold of 10 were 98.4 & 96.1 (100 & 96.7).

3.3 Comparing FIX with standard Classifiers

In order to choose the best fusion-layer classifier, we assessed decision tree (i.e., FIX-TREE), random forest (i.e., FIX-RF), linear SVM (i.e., FIX-SVM-LIN), and SVM with RBF kernel (i.e., FIX-SVM-RBF) in the stacking layer. Moreover, in order to justify such a complex classifier architecture in FIX, we compared these four solutions with 6 widely used classification techniques, i.e., LDA (linear discriminant analysis), SVM-RBF (SVM with RBF kernel), SVM-LIN (linear SVM), TREE (decision tree), RF (random forest), and GLM (logistic regression). The comparison corresponds to 8 datasets each with different characteristics (see Sections 3.1 and 3.2). For instance, while one of the is a combination of multiband EPI data from a standard 3T clinical scanner and pilot data from the HCP "Connectome Skyra" (i.e., a fairly large dataset with a good quality), another one is a combination of two different high-TR acquisitions. According to the results in Figure 9 FIX with random forest in the stacking layer outperforms the standard Classifiers, i.e., it has the highest average "mean accuracy" across these datasets. Moreover, according to the comparisons in the early stages of FIX's development, when the quality of training data (in terms of size, mix and acquisition quality) is lower, the gap *widens* in FIX's favour.

4 Conclusions and Discussion

We have described a new tool for the automated denoising of artefacts in fMRI data, achieved by running independent component analysis, identifying which components correspond to artefactual processes in the data, and removing those from the data. Our tool, FIX, can achieve over 99% classification accuracy on the best fMRI datasets, and around 95% accuracy on more "standard" acquisitions (particularly if study-specific training is carried out). FIX therefore can be a very valuable tool for the cleanup of fMRI data.

FIX employs a large number of features in order to inform its decision making about many componentwise attributes, ranging from spatial and temporal characteristics to image-acquisition parameters. As presented in Section 3.1, features are partially correlated, which might suggest a hard feature selection prior to any classification. While hard feature pre-selection might purify the feature-base of redundancies, it introduces the risk of loosing some useful/discriminant information. Most feature-selection techniques are sensitive to the inter-class discriminant power of features, which makes them ideal for cases where there is minimal within-class heterogeneity. However, comparing various fMRI artefact components (e.g., Figures 2–6) shows that there is a huge heterogeneity across various different kinds of artefacts, in terms of their causes and their spatial and temporal characteristics. Consider a feature that is defined to be particularly helpful in identifying a *rare* artefact type. Automated feature selection might well reject this feature, as it does not provide good

discrimination between noise (as a whole, averaged over all artifact types) and signal. Therefore, FIX does not employ a global feature selection, which drops the features from the whole process; it rather advocates a stacking architecture, where a high-level learner decides (in a data-driven way) how to combine feature-selected classification's results with classification results from temporal, spatial and full-data classifications (see Figure 8). Under this hierarchy, for a particular training dataset, if feature selection did not loose any discriminant power and could outperform every other scenario (as shown in Figure 8) across all components, the high-level learner will only consider the result from the $\mathcal{D}_{sel}$ path in Figure 8.

Similar to other classification techniques, when training FIX, we ideally need "expert" labelling to be provided with the training data. In the case where there is a bias (e.g., always calling a particular type of artefact signal) in the experts labelling, the classifier will be biased as a result. In the case where there are inconsistencies in the labelling (i.e., similar components being labelled as both signal and noise), then the classifier will not be able to learn that concept (due to conflicting evidence). This is unavoidable for such approaches, but in such cases one might at least hope to ameliorate lack of expertise by utilising several "experts" to cross-check each others labelling results. Where it is necessary to apply auto-classification without the confidence of expert training (or when the training data does not well match the data to be classified) it would probably be advisable to choose a conservative classification threshold to reduce the risk of removing signal components.

In addition to the quality of expert's labelling, the quality of the data itself, in terms of acquisition quality and the heterogeneity in the training dataset (which can increase when combining different datasets), is an important factor that can influence FIX's performance. As the results in Section 3 show, FIX's performance varies from 95% on conventional datasets to 99% and more on high-quality HCP data. When combining multiple (relatively different) datasets, FIX's performance can even drop to lower than 95%. Thus, we recommend training FIX on homogeneous datasets in order to improve its accuracy. In case of training FIX on a pool of multiple datasets, the recommended approach is to first test its performance on held-out datasets, and if (slightly) less accurate than desired, utilise a conservative threshold.

FIX is publicly available; the current version (v1.05) is available as a "plugin" for FSL (the FMRIB Software Library) from www.fmrib.ox.ac.uk/fslwiki/fsl/FIX - it is not yet bundled as part of FSL, as it currently relies also on other software, in particular on Matlab (or Octave) and R. We plan to recode a future version of FIX to remove these dependencies and release it as part of FSL. The FIX download includes training-weights files for "standard" fMRI acquisitions and for Human Connectome Project rfMRI data; in our experience, new acquisition protocols do benefit from customised re-training of FIX. Hand training of FIX on new datasets ideally needs at least 10 subjects' ICA outputs to be hand labelled, and quite possibly more than that; the scripts supplied with fix make LOO evaluation very straightforward, and the value of adding further hand labelling can be established by noting whether the LOO result (as a function of number of datasets manually labelled) is asymptoting.

Acknowledgments

We are very grateful to Erin Reid and Donna Dierker (WashU), for helping with the FIX training (hand-labelling of ICA components) from HCP data, to Eugene Duff and other members of the FMRIB Analysis Group for input on the FIX feature set and scripting, and to David Flitney (Oxford), for creating the Melview ICA component viewing and labelling tool. We are grateful for partial funding via the following NIH grants: 1U54MH091657-01, P30-NS057091, P41-RR08079/EB015894, F30-MH097312. Gwenäelle Douaud is funded by the UK Medical Research Council (MR/K006673/1).

References

Amit Y, Geman D. Shape quantization and recognition with randomized trees. Neural Computation. 1997; 9(7):1545–1588.

Beckmann CF, Smith SM. Probabilistic independent component analysis for functional magnetic resonance imaging. IEEE Trans Med Imaging. 2004; 23(2):137–52. [PubMed: 14964560]

Birn R, Diamond J, Smith M, Bandettini P. Separating respiratory-variation-related fluctuations from neuronal-activity-related fluctuations in fMRI. Neuroimage. 2006; 31(4):1536–1548. [PubMed: 16632379]

Breiman L. Random forests. Machine Learning. 2001; 45(1):5–32.

Caputo, B.; Sim, K.; Furesjo, F.; Smola, A. Appearance-based Object Recognition using SVMs: Which Kernel Should I Use. Proc of NIPS workshop on Statitsical methods for computational experiments in visual processing and computer vision; 2002.

Cortes, Vapnik V. Support-vector networks. Machine Learning. 1995; 20(3):273–297. URL http://dx.doi.org/10.1007/BF00994018. 10.1007/BF00994018

De Martino F, Gentile F, Esposito F, Balsi M, Di Salle F, Goebel R, Formisano E. Classification of fMRI independent components using IC-fingerprints and support vector machine Classifiers. Neuroimage. 2007; 34(1):177–194. [PubMed: 17070708]

Dzeroski S, Zenko B. Is combining Classifiers with stacking better than selecting the best one? Machine Learning. 2004; 54(3):255–273.

Feinberg D, Moeller S, Smith S, Auerbach E, Ramanna S, Gunther M, GMF, Miller K, Ugurbil K, Yacoub E. Multiplexed echo planar imaging for sub-second whole brain FMRI and fast diffusion imaging. PloS One. 2010; 5(12)

Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 2012; 42(4):63–484.10.1109/TSMCC.2011.2161285

Glasser MF, Van Essen DC. Mapping human cortical areas in vivo based on myelin content as revealed by T1-and T2-weighted MRI. The Journal of Neuroscience. 2011; 31(32):11597–11616. [PubMed: 21832190]

Golver GH, Li TQ, Ress D. Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. Magn Reson Med. Jul; 2000 44(1):162–7. [PubMed: 10893535]

Griffanti, L.; Salimi-Khorshidi, G.; Beckmann, CF.; Auerbach, E.; Douaud, G.; Zsoldos, E.; Ebmeier, K.; Filippini, N.; Mackay, C.; Moeller, S.; Xu, J.; Yacoub, E.; Baselli, G.; Ugurbil, K.; Miller, K.; Smith, S. Automated artefact removal and accelerated fMRI acquisition for improved resting-state network imaging. in submission

Ho T. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998; 20(8):832–844.

Hothorn T, Hornik K, van de Wiel M, Zeileis A. A lego system for conditional inference. The American Statistician. 2006a; 60(3):257–263.

Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics. 2006b; 15(3):651–674.

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

Kiviniemi V, Kantola J-H, Jauhiainen J, Hyvärinen A, Tervonen O. Independent component analysis of nondeterministic fMRI signal sources. NeuroImage. 2003; 19:253–260. [PubMed: 12814576]

Kochiyama T, Morita T, Okada T, Yonekura Y, Matsumura M, Sadato N. Removing the effects of task-related motion using independent-component analysis. Neuroimage. Apr; 2005 25(3):802–814.10.1016/j.neuroimage.2004.12.027 [PubMed: 15808981]

McKeown MJ, Makeig S, Brown GG, Jung TP, Kindermann SS, Bell AJ, Sejnowski TJ. Analysis of fMRI data by blind separation into independent spatial components. Hum Brain Mapp. 1998; 6(3): 160–188. [PubMed: 9673671]

Moeller S, Yacoub E, Olman C, Auerbach E, Strupp J, Harel N, Ugurbil K. Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. Magnetic Resonance in Medicine. 2010; 63(5):1144–1153. [PubMed: 20432285]

Niazy, R.; Xie, J.; Miller, K.; Beckmann, C.; Smith, S. Spectral characteristics of resting state networks. In: Van Someren, E., editor. Progress in Brain Research. Vol. 193. Elsevier; 2011. p. 259-276.

Perlbarg V, Bellec P, Anton J-L, Pelegrini-Issac M, Doyon J, Benali H. CORSICA: correction of structured noise in fMRI by automatic identification of ICA components. Magn Reson Imaging. Jan; 2007 25(1):35–46.10.1016/j.mri.2006.09.042 [PubMed: 17222713]

Salimi-Khorshidi G, Smith S, Nichols T. Adjusting the Effect of Nonstationarity in Cluster-based and TFCE Inference. NeuroImage. 2010

Schölkopf B, Smola A, Williamson R, Bartlett PL. New support vector algorithms. Neural Computation. 2000; 12:1207–1245. [PubMed: 10905814]

Shmueli K, van Gelderen P, de Zwart J, Horovitz S, Fukunaga M, Jansma J, Duyn J. Low-frequency fluctuations in the cardiac rate as a source of variance in the resting-state fMRI BOLD signal. Neuroimage. Aug; 2007 38(2):306–20. [PubMed: 17869543]

Smith S. Fast robust automated brain extraction. Hum Brain Mapp. 2002; 17(3):143–55. [PubMed: 12391568]

Smith S, Andersson J, Auerbach E, Beckmann C, Bijsterbosch J, Douaud G, Duff E, Feinberg D, Griffanti L, Harms M, Kelly M, Laumann T, Miller K, Moeller S, Petersen S, Power J, Salimi-Khorshidi G, Snyder A, Vu A, Woolrich M, Xu J, Yacoub E, Ugurbil K, Van Essen D, Glasser M. for the WU-Minn HCP Consortium. Resting-state fMRI in the Human Connectome Project. NeuroImage. 2013 In press.

Smith SM, Nichols TE. Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference. NeuroImage. 2009; 44(1):83–98. [PubMed: 18501637]

Smith SM, Miller KL, Moeller S, Xu J, Auerbach EJ, Woolrich MW, Beckmann CF, Jenkinson M, Andersson J, Glasser MF, Van Essen DC, Feinberg DA, Yacoub ES, Ugurbil K. Temporally-independent functional modes of spontaneous brain activity. Proc Natl Acad Sci U S A. Feb; 2012 109(8):3131–3136.10.1073/pnas.1121329109 [PubMed: 22323591]

Strasser H, Weber C. On the asymptotic theory of permutation statistics. Mathematical Methods of Statistics. 1999; 8:220–250.

Tohka J, Foerde K, Aron AR, Tom SM, Toga AW, Poldrack RA. Automatic independent component labeling for artifact removal in fMRI. Neuroimage. Feb; 2008 39(3):1227–1245.10.1016/j.neuroimage.2007.10.013 [PubMed: 18042495]

Ugurbil K, Xu J, Auerbach E, Moeller S, Vu A, Duarte-Carvajalino J, Lenglet C, Wu X, Schmit-ter S, Van de Moortele P, Strupp J, Sapiro G, De Martino F, Wang D, Harel N, Garwood M, Chen L, Feinberg D, Smith S, Miller K, Sotiropoulos S, Jbabdi S, Andersson J, Behrens T, Glasser M, Van Essen D, Yacoub E. for the WU-Minn HCP Consortium. Pushing spatial and temporal resolution for functional and diffusion MRI in the Human Connectome Project. NeuroImage. 2013 In press.

Van Essen D, Smith S, Barch D, Behrens T, Yacoub E, Ugurbil K. for the WU-Minn HCP Consortium. The WU-Minn Human Connectome Project: An overview. NeuroImage. 2013 In press.

Venables, W.; Ripley, B. Statistics and Computing. Springer; 2002. Modern Applied Statistics with S.

Wolpert D. Stacked generalization. Neural Networks. 1992; 5(2):241–259.

Zhang Y, Brady M, Smith S. Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. IEEE Trans on Medical Imaging. 2001; 20(1):45–57.

A Classifiers

This appendix provides a summary of the Classifiers used in FIX and briefly explains their mathematical models. The classification unit in FIX has two main tasks: Learning and prediction. The learning (or inference) phase consists of training a two-class classification model on dataset $\mathscr{D}=\{\mathbf{x}_i,\ y_i\}_{i=1}^{N}$, where N denotes the number of examples in the data, $\mathbf{x}_i$ denotes the features vector (of length m) corresponding to the ith example, and y_i is a scalar denoting ith example's class (e.g., 0 for noise and 1 for signal). In the prediction phase, on the other hand, the trained model scores a given feature vector $\mathbf{x}^*$ with a likelihood of it being signal or noise.

A.1 *k*-NN

The *k*-nearest neighbour algorithm (*k*-NN) is a method for classifying objects based on the closest training examples in the feature space. It is a type of instance-based learning (or lazy learning) where the decision function for each test example is only approximated locally. The *k*-NN algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its *k* nearest neighbours. Here, however, the proportion of the votes for the winning class are returned, so that *k*-NN's output becomes probabilistic (and hence more appropriate for stacking).

The *k*-NN algorithm's only parameter, *k*, is a positive integer (typically small). The best choice of *k* depends on the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques such as cross-validation. In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes. One of the main disadvantages of the *k*-NN algorithm is that its accuracy can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification [Venables and Ripley, 2002].

A.2 Logistic Regression

The fundamental model underlying ordinary linear regression posits that a continuous outcome variable is, in theory, a linear combination of a set of predictors, plus an error. In other words, for an outcome variable, y_i, and a set of m predictor variables (i.e., features), the multiple regression model is of the form

$$y_i=\beta_0+\sum_{j=1}^{m}\beta_j\mathbf{x}_i(j)+\varepsilon, \quad (5)$$

where β_j denotes the regression coefficient (i.e., the expected change in y_i per unit change in feature i assuming that all other features are held constant), and ε is the error of prediction. Given that one of the underlying assumptions of the above model is that the dependent variable, Y, is continuous, one cannot directly use this model for classification.

The generalised linear models (GLM) framework, however, provides a flexible generalisation of ordinary linear regression that allows for response variables that have other than a normal distribution. This generalisation is provided by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. Binary logistic regression is a special GLM that employs a logistic link function for modelling the probability of dichotomous outcome variables (e.g., signal/‘1’ vs. noise/‘0’).

Assume that in $\mathcal{D}$, we denote signal with 1 and noise with 0, and $p = P(Y = 1) = 1 - P(Y = 0)$. In the absence of other information, we would estimate p by the sample proportion of cases for which Y is 1. However, in the regression context, it is assumed that there is a set of predictor variables/features, $\mathbf{x}$, that are related to Y and, thus, can provide additional information for predicting Y . In binary logistic regression, this mapping from feature space, $\mathbf{x}$, to class labels is a linear model for the natural logarithm of the odds (i.e., the log-odds) in favour of $Y = 1$:

$$\log\left[\frac{P(Y=1|\mathbf{x}_i)}{1-P(Y=1|\mathbf{x})}\right]=\log\left[\frac{\pi}{1-\pi}\right]=\beta_0+\sum_{i=1}^{m}\beta_j\mathbf{x}_i(i), \quad (6)$$

or alternatively

$$P(Y=1|\mathbf{x}_i)=\frac{e^{\beta_0+\sum_{j=1}^{m}\beta_j\mathbf{x}_i(j)}}{1+e^{\beta_0+\sum_{j=1}^{m}\beta_j\mathbf{x}_i(j)}}. \quad (7)$$

Because of the nature of the model, its parameters are estimated using maximum likelihood rather than least-squares. When using logistic regression, one can decide to enter a variable into the model *if* its associated significance level is less than a given P-value (e.g., 0.05). Variable-wise P-values (that the logistic regression model provides) are useful criteria for feature selection: Only features that have significant effect on the dependent variable can be selected.

A.3 Support Vector Machines

A support vector machine (SVM) is a supervised learning approach that is used for classification and multivariate regression analyses [Cortes and Vapnik, 1995]. Suppose examples in data $\mathcal{D}$ each belong to one of two classes (i.e., with noise being −1 and signal being 1), and the goal is to decide which class a new data point (feature vector) $\mathbf{x}^*$ will be in. In the simplest case a data point is viewed as a p-dimensional vector (a list of m features), and we want to know whether we can separate such points with a (m−1)-dimensional hyperplane. This is a linear classifier. There are many hyperplanes that might classify the

data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes.

In all SVM analyses (described below), non-binary variables are scaled, i.e., data are scaled internally to zero mean and unit variance. The centre and scale values are kept and used for later predictions (i.e., for **x***).

A.3.1 Linear SVM

Any hyperplane can be written as the set of points satisfying

$$\mathbf{w}.\mathbf{x}-b=0, \quad (8)$$

where . denotes the dot product and **w** the normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector. If the training data are linearly separable, we can select two hyperplanes in a way that they separate the data with a maximum margin and there are no points between them, i.e.,

$$\mathbf{w}.\mathbf{x}-b=1 \text{ and } \mathbf{w}.\mathbf{x}-b=-1. \quad (9)$$

The distance between these two hyperplanes is $\frac{2}{\|\mathbf{w}\|}$, which can be maximised by minimising ||**w**||, while preventing data points from falling into the space between the separating planes. This way, the inference problem of the SVM becomes the following optimisation problem:

$$\text{minmise}\|\mathbf{w}\|, \text{ subject to } y_i(\mathbf{w}.\mathbf{x}_i-b) \geq 1, \ 1 \leq i \leq N. \quad (10)$$

This optimisation problem is difficult to solve analytically because it depends on ||**w**||, the norm of **w**, which involves a square root. Altering the equation by substituting ||**w**|| with $\frac{1}{2}\|\mathbf{w}\|^2$, while preserving the same **w** and *b* at the minimum, results in the following quadratic programming optimisation problem

$$\text{minimise}\frac{1}{2}\|\mathbf{w}\|^2, \text{ subject to } y_i(\mathbf{w}.\mathbf{x}_i-b) \geq 1, \ 1 \leq i \leq N, \quad (11)$$

that is easier to solve. By introducing Lagrange multipliers α, the previous constrained problem can be expressed as

$$\min \max \left\{ \frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\alpha_i \left[y_i(\mathbf{w}.\mathbf{x}_i-b)-1 \right] \right\}. \quad (12)$$

Only a few α_i will be greater than zero, whose corresponding $\mathbf{x}_i$ are exactly the “support vectors” (features defining the separating hyperplane), which lie on the margin and satisfy $y_i(\mathbf{w}.\mathbf{x}_i - b) = 1$. From this, one can show that the support vectors also satisfy

$$\mathbf{w}.\mathbf{x}-b=1/y_i=y_i \iff b=\mathbf{w}.\mathbf{x}_i-y_i, \quad (13)$$

which then allows the calculation of *b* as

$$b=\frac{1}{N_{SV}}\sum_{i=1}^{N_{SV}}(\mathbf{w}\mathbf{x}_i-y_i) \quad (14)$$

This solution requires some modification if there exists no hyperplane that can perfectly split the two classes. This solution is known as the “soft margin” method [Cortes and Vapnik, 1995], which will choose a hyperplane that splits the examples as cleanly as possible, while still maximising the distance to the nearest cleanly split examples. This is achieved by introducing slack variables, ζ_i, which measure the degree of misclassification of example $\mathbf{x}_i$

$$y_i(\mathbf{w}.\mathbf{x}_i-b)\geq 1-\zeta_i, \ \ i\leq i\leq N. \quad (15)$$

The optimisation of the soft-margin objective function becomes a trade-off between a large margin and a small error penalty:

$$\min_{\mathbf{w},\zeta,b}\left\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{N}\zeta_i\right\}, \text{ subject to } y_i(\mathbf{w}.\mathbf{x}_i-b)\geq 1-\zeta_i, \ \zeta_i\geq 0. \quad (16)$$

By using Lagrange multipliers (as done above), soft-margin SVM has then to solve the following optimisation problem:

$$\min_{\mathbf{w},\zeta,b}\max_{\alpha,\beta}\left\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{N}\zeta_i-\sum_{i=1}^{N}\alpha_i[y_i(\mathbf{w}\mathbf{x}_i-b)-1+\zeta_i]-\sum_{i=1}^{N}\beta_i\zeta_i\right\}, \text{ with } \alpha_i,\beta_i\geq 0. \quad (17)$$

Having learned the SVM model, for any testing instance **x***, the decision function (predictor) is $f(\mathbf{x}^*) = \text{sgn}(\mathbf{w}^T\mathbf{x}^* + b)$. Detailed description of SVM methodology can be found in Schölkopf et al. [2000] and its references.

A.3.2 Nonlinear SVM

The nonlinear SVM generally employs the application of nonlinear kernels to the feature space, resulting in an algorithm that is formally similar to the linear case, except that every dot product (between two features) is replaced by a nonlinear kernel function. In other words, the maximum-margin hyperplane is fitted in a transformed feature space. The most common such kernels are shown in Table 3.

The ‘C’-constant of the regularisation term in the Lagrange formulation (an internal parameter of the soft-margin SVM) is optimised by cross-validation, e.g., via an internal LOO loop (run inside each fold of the outer-most LOO loop). Moreover, model selection for SVM_r is achieved empirically, where the optimal values of the width hyper-parameter are expected to lie between the 0.1 and 0.9 quantiles of the $\|\mathbf{x}-\mathbf{x}'\|^2$ statistic [Caputo et al.,

2002]. We use the median, but any value in between those two bounds has been found to produce good results.

A.3.3 Feature Selection Using SVM

After obtaining a linear SVM model, the weights **w** can be used to rank the relevance of each feature; the larger the $|\mathbf{w}(j)|$ (the weight corresponding to jth feature), the more important is the role that the jth feature plays in the decision function. Note that, only such weights in the linear SVM model have this simple interpretation, so this approach is restricted to linear SVM.

A.4 Decision Trees

A decision tree uses a tree-like graph or model of decisions and their possible consequences. A decision tree, when used as a classifier, is a predictive model which maps the features of an example ($\mathbf{x}_i$) to its class labels (y_i). Leaves represent (probabilistic) class labels (e.g., signal and noise) and branches represent conjunctions of features that lead to those class labels.

As with the other approaches described above, decision trees are first trained and then used to predict (classify). Most learning algorithms used for constructing decision trees are top-down; at each step of their operation, they work by choosing a feature, $\mathbf{x}(j)$, that is the next best feature to use in splitting the set of N items/examples in $\mathcal{D}$. Different algorithms use different specific formulae for defining “best”, but they all agree in that “best” is defined by how well a given variable splits the set into homogeneous subsets that have the same class. One of the common formulae for learning the tree is recursive partitioning in a conditional inference framework.

Conditional inference trees estimate a relationship between all $\mathbf{x}_i$ and y_i pairs in $\mathcal{D}$ by binary recursive partitioning in a conditional inference framework. In its first step, the algorithm tests the global null hypothesis of independence between any of the input variables and the response (i.e., class labels); it stops if this hypothesis cannot be rejected. Next, it selects the input variable with strongest association to the response. This association is measured by a P-value corresponding to a test for the partial null hypothesis of a single input variable and the response. In its second step, it implements a binary split in the selected input variable, and then repeats the first and second steps. The mathematical details of this approach can be found in Strasser and Weber [1999], Hothorn et al. [2006a,b].

In this study, we employed decision trees both as one of the distinct lower-level Classifiers and as higher-level (“fusion-level”) classifier that combines the lower-level classifier outputs. In both cases, we stop the tree’s growth if it results in end leaves with less than 20 examples in them, or if the discriminant test is not significant ($P > 0.05$).

A.5 Random Forests

Significant improvements in classification accuracy have resulted from growing an ensemble of Classifiers and letting them vote for the most popular class (see Section 2.4).

Random forest refers to an ensemble learning method for classification (and regression) that operates by creating multiple decision trees at training and outputting the class that is most voted for by individual trees. The method was first introduced by Breiman [Breiman, 2001], which combines his “bagging” idea with the random selection of features introduced independently by Ho [1998] and Amit and Geman [1997].

Constructing a model within the random-forests framework requires making several choices regarding the shape of the decision to use in each “node”, the type of predictor to use in each “leaf”, the splitting objective to optimise in each node, and the method for injecting randomness into the trees.

The types of decisions to make at each node vary from simple thresholding of a single dimension of the input (very common and leads to trees that partition the space into hyper-rectangular regions) to other decision shapes, such as splitting a node using linear or quadratic decisions. When in an end leaf, leaf predictors determine the prediction for a given sample/example. Choices here vary from using a histogram for categorical data, or constant predictors for real valued outputs. Note that, in theory, one could employ more complicated predictors (e.g., Support Vector Machine or any other classifier); however, in practice the simple predictors are more common (e.g., due to the lack of large number of sample in an end leaf).

One of the most important components in defining an algorithm within the random-forest framework is the splitting objective function, which refers to the process of ranking the candidate splits of a leaf as the tree grows. The most common such measures are information gain and the Gini impurity. On the last choice, in order to inject randomness into each tree, Breiman’s original algorithm (which is the technique used in this paper) proposes the following approach: Each tree is trained on a bootstrapped sample of the original data set, and each time a leaf is split, only a randomly chosen subset of the dimensions are considered for splitting. In Breiman’s model, once the dimensions are chosen the splitting objective is evaluated at every possible split point in each dimension and the best is chosen.

In this study, we employed Breiman’s random forest algorithm both as an independent classifier to compare FIX with, and as a higher-level classifier that combines the lower-level classifier outputs. In both cases, we employed a forest with 500 trees, where the splits are ranked by their Gini impurities.

A.5.1 Ensemble Learning in FIX

In the early days of machine learning, many distinct approaches were proposed, each with its own strengths and weaknesses. Hence much effort was put into comparing between approaches in order to select a single “optimal” one for a given problem. Systematic empirical comparisons showed that the best learner varies from application to application, and systems containing many different learners started to appear. Effort was then put into trying many variations of many learners, but still selecting just the best one. Following this, it was noted that, if instead of selecting the best variation found, one *combined* many variations, the results are better - often much better - and at little extra effort for the user.

Creating such model ensembles is now common. In the simplest technique, called bagging, one simply generates random variations of the training set by resampling, learns a classifier on each, and combines the results by voting. This works because it greatly reduces variance while only slightly increasing bias. In boosting, training examples have weights, and these are varied so that each new classifier focuses on the examples the previous ones tended to get wrong. In stacking, which is the ensemble technique used in FIX, the outputs of individual Classifiers become the inputs of a “higher-level” classifier that learns out how best to combine them.

FIX employs multiple Classifiers (i.e., linear SVM, SVM with RBF kernel, random forest, and conditional-inference tree) as its high-level learner. This makes FIX’s hierarchical classifier a stacking ensemble learner. The details of the inputs to this high-level learner are described in Section 2.4.

B Feature Summaries

In this appendix, we list the temporal and spatial features that are described in Section 2.2. The goal of this section is to provide a detailed figure for the number of features that FIX has in each (sub-)category of spatial and temporal features (described in Section 2.2).

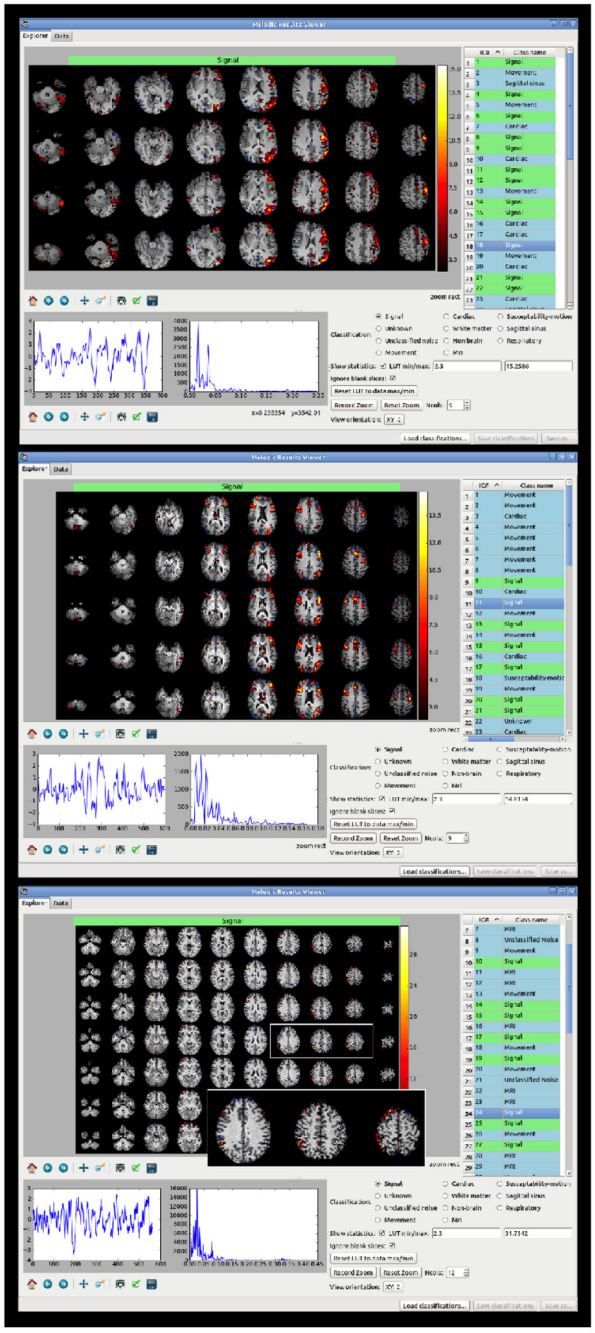

Figure 1.

Examples of “good” components from three different acquisitions. The spatial map for the high resolution, short TR acquisition (bottom; acquisition C, see text for more details) is visually strikingly different from a more standard acquisition (top and middle, acquisitions A and B, see text), with the signal above threshold following very closely the cortical gyrification. The spectral power lies primarily between 0 and 0.05 Hz for each component.

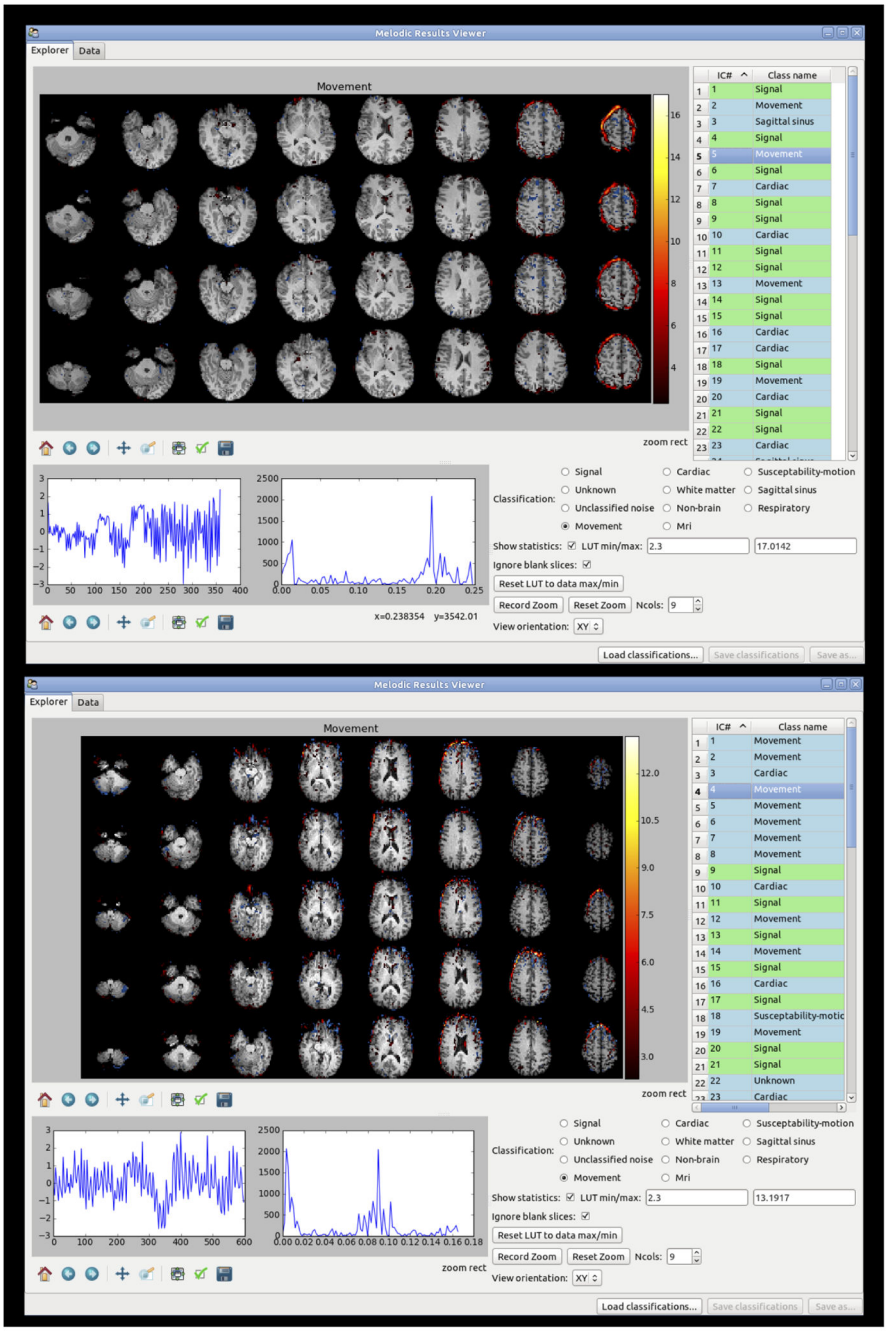

Figure 2.

Example movement-related artefacts. The signal above threshold in the spatial maps is essentially at the edges of the brain. The frequencies of the power spectra are disparately distributed and the time courses visually dissimilar.

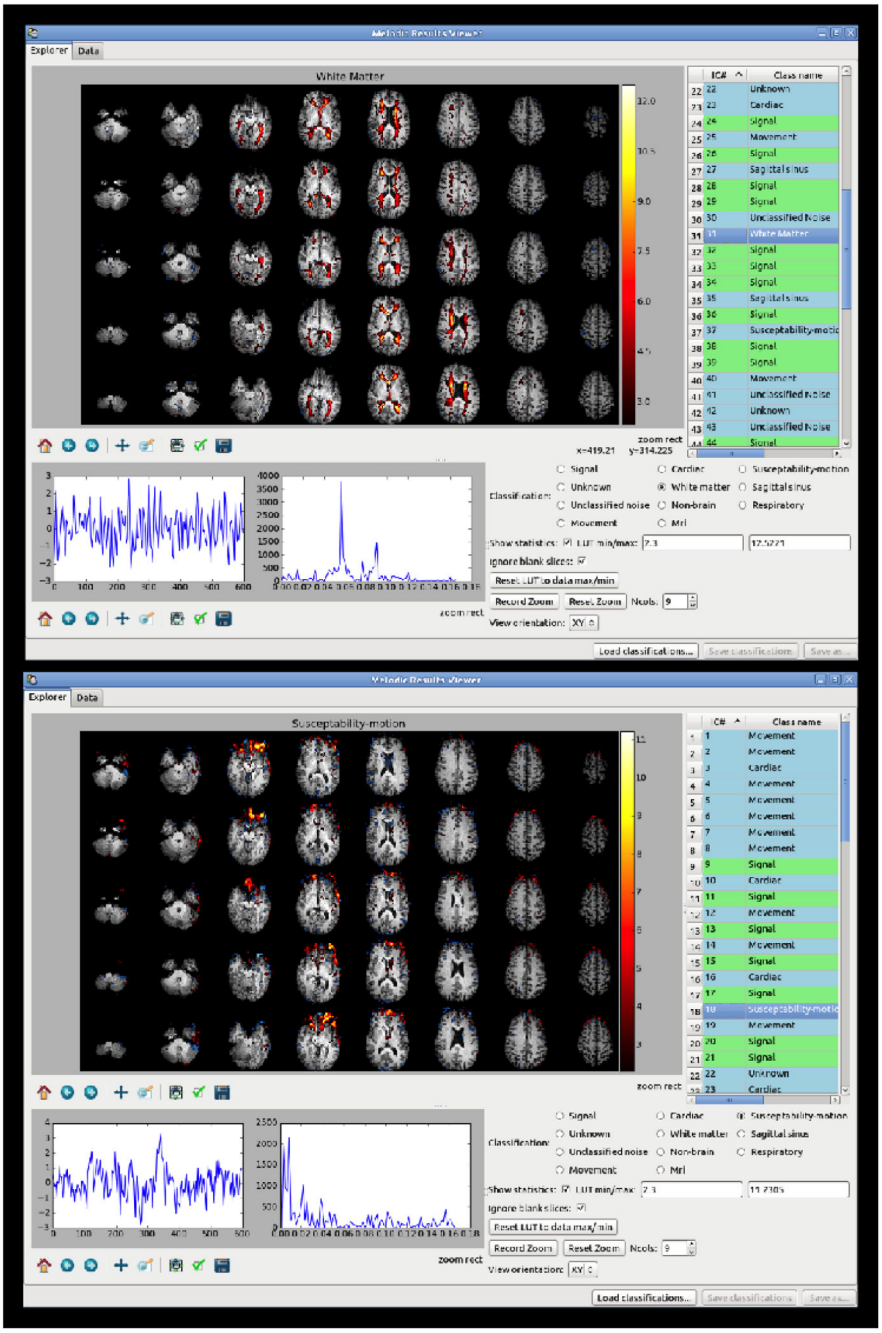

Figure 3.
Two further noise components: “white matter” and “susceptibility-motion”.

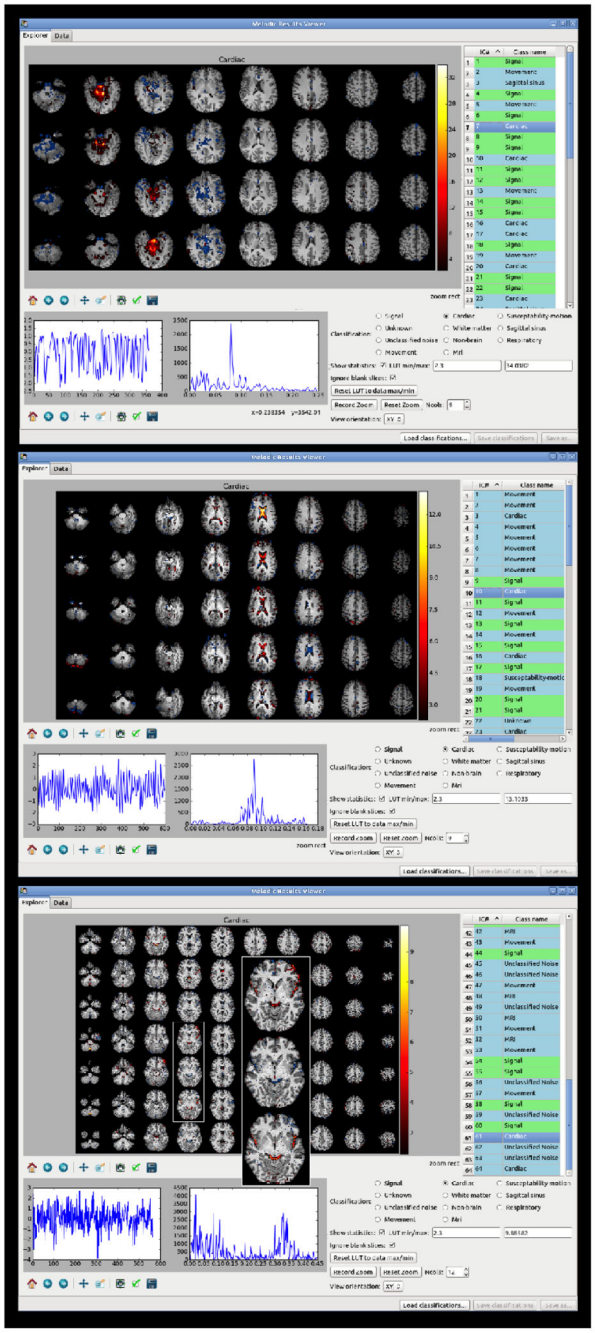

Figure 4.

Examples of cardiac-related components. This includes components due to cardiac pulsation and arterial contribution. The signal above threshold in the spatial maps is essentially located in the ventricles, or following the main arteries (posterior cerebral artery, middle cerebral branches).

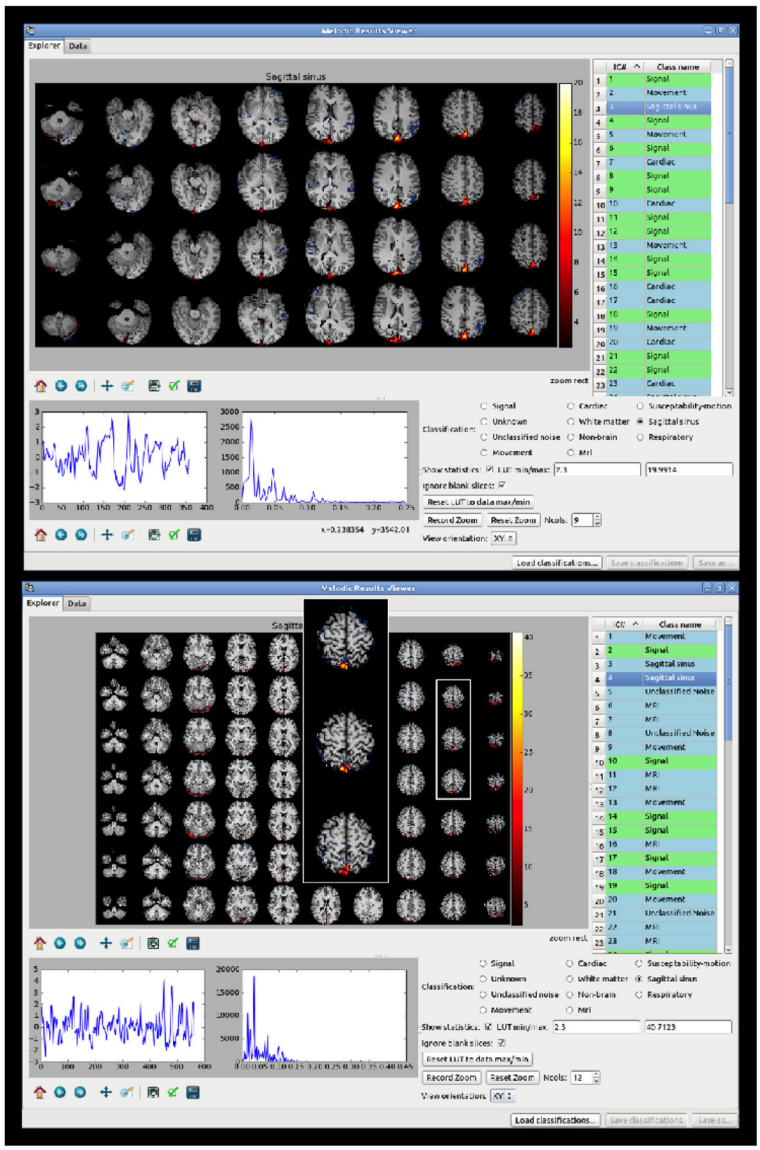

Figure 5.

Example components relating to large veins. The signal above threshold in the spatial maps is essentially following the sagittal sinus.

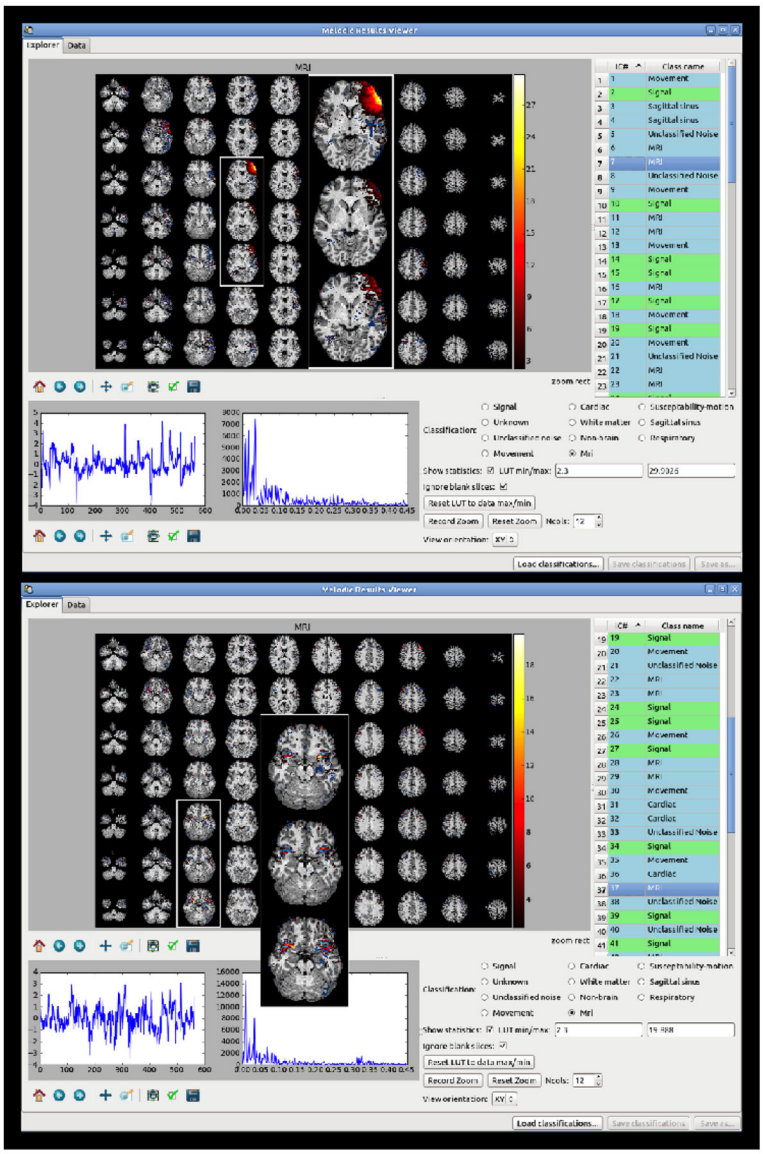

Figure 6.
Two MRI acquisition/reconstruction related artefact components.

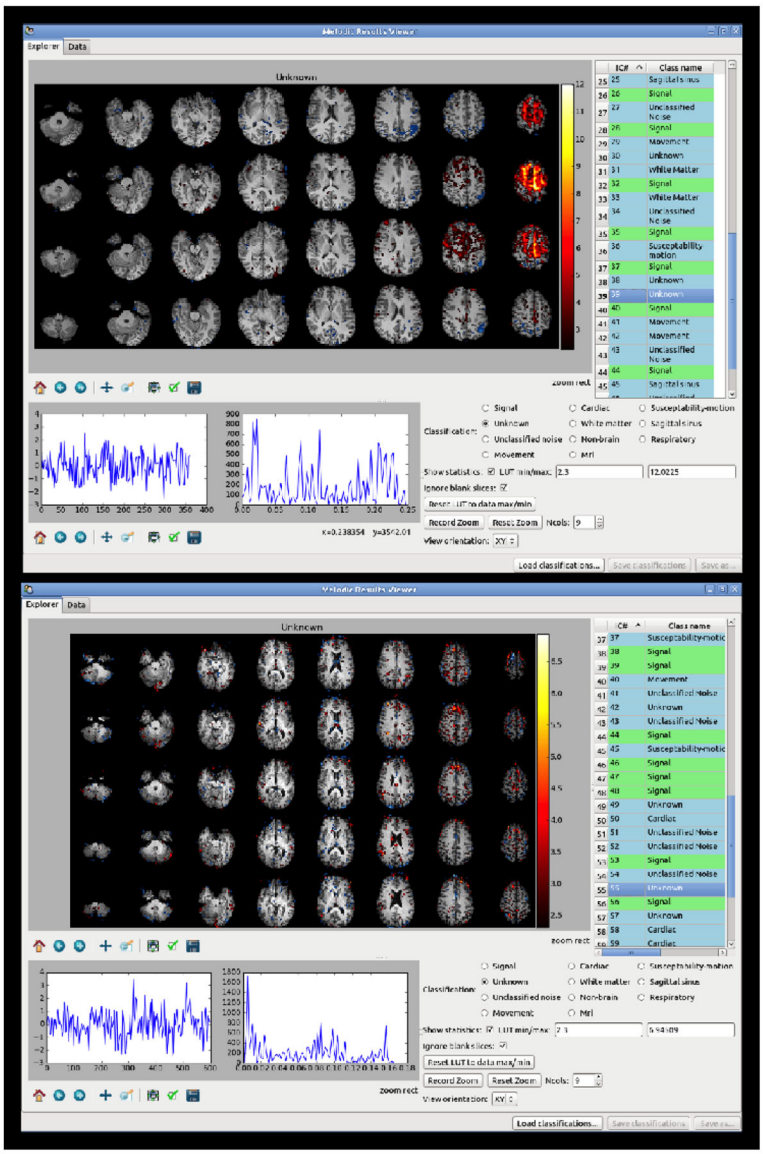

Figure 7.
Two examples of “unknown” components.

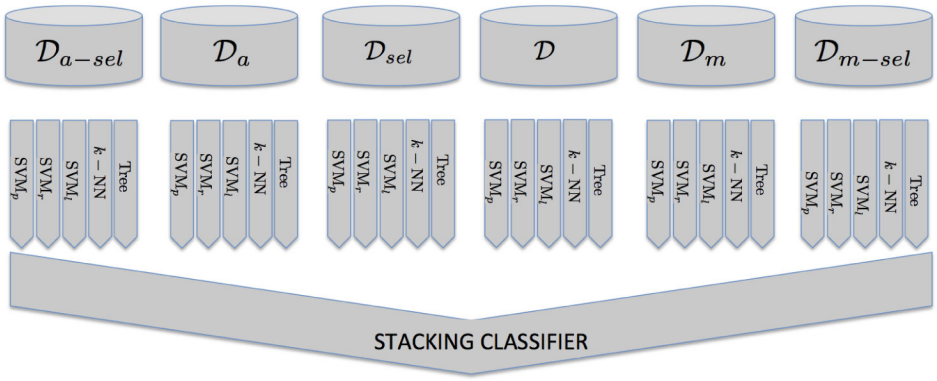

Figure 8.

FIX's hierarchical classifier. In the data layer, full, feature-selected, temporal, spatial, temporal-feature-selected and spatial-feature-selected datasets ($\mathcal{D}$, $\mathcal{D}_{sel}$, $\mathcal{D}_a$, $\mathcal{D}_m$, $\mathcal{D}_{a-sel}$ and $\mathcal{D}_{m-sel}$, respectively), are each classified by 5 Classifiers. These Classifiers consist of *k*-NN, SVM_r (SVM with RBF kernel), SVM_p (SVM with polynomial kernel), SVM_l (linear SVM) and decision tree (simply called tree here). The result is a vector of 30 (5 × 6) probabilities (0 and 1 denoting perfect noise and perfect signal, respectively), which is the input to a fusion-layer classifier, whose output is the probability of IC being signal/noise.

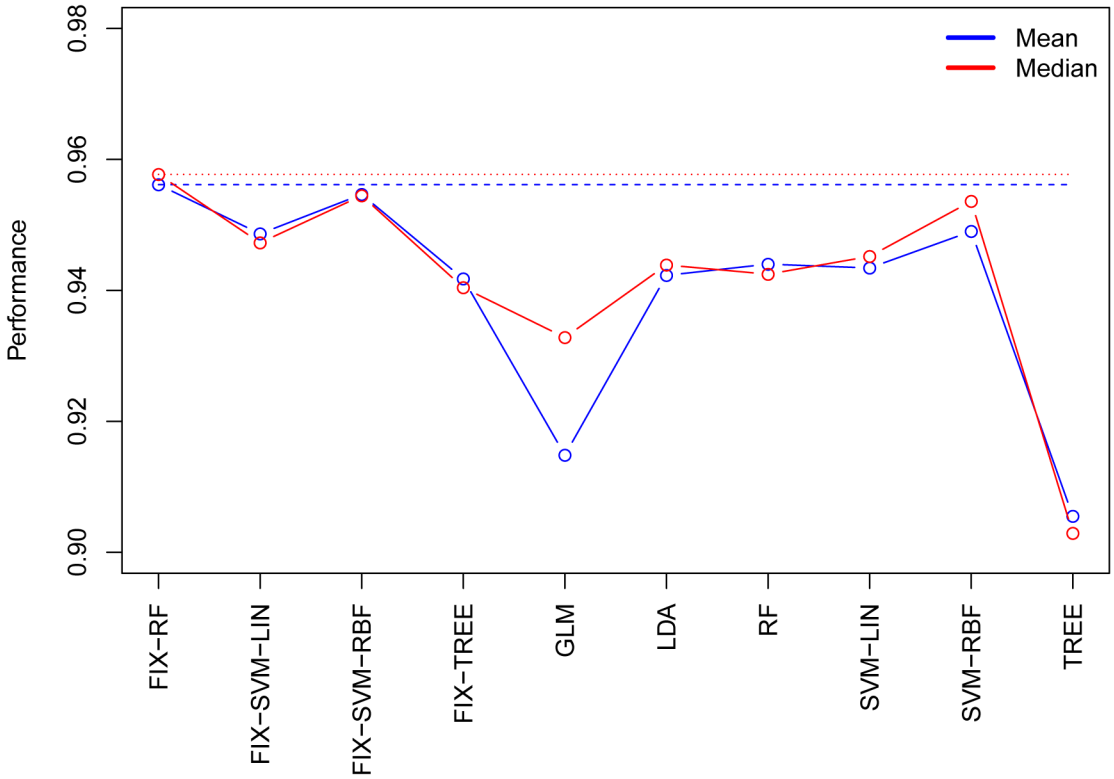

Figure 9.

FIX-RF and FIX-SVM-RBF outperform the commonly-used classifiers on a broad set of rFMRI datasets that cover a board range of data-acquisition/-quality scenarios, common in rFMRI. In the figure, classifiers are shown on the x-axis, and the y-axis shows the average accuracy across all datasets. For each dataset, accuracy is defined as the average of subject-wise (TPR+TNR)/2 (see Section 2.5), where TPR and TNR denote the true positive and true negative rates, respectively. The thick blue and red lines show the mean and median of accuracy across datasets, respectively, and dashed blue and red lines shows the best classifier's (i.e., FIX-RF) performance in terms of its mean and median, respectively. Thus, on average, FIX is expected to outperform other classifiers, and the best simple classifier next to FIX is SVM-RBF.

Table 1

The table of abbreviations for various subsets and transformations of component spatial maps. Masks have 0s and 1s for voxels' values, while non-mask options comprise a subset of original voxels, with their original Z-statistics.

Abbreviation	Description
$\mathbf{m}$	IC's spatial map
$\mathbf{m}_a$	Absolute value of $\mathbf{m}$, i.e., $\lvert\mathbf{m}\rvert$
$\mathbf{m}_p$	The positive voxels of $\mathbf{m}$
$\mathbf{m}_n$	The negative voxels of $\mathbf{m}$
$\mathbf{m}_p^b$	The mask for positive voxels of $\mathbf{m}$
$\mathbf{m}_n^b$	The mask for negative voxels of $\mathbf{m}$
$\mathbf{m}_p^\tau$	The voxels in $\mathbf{m}$ that are bigger than $\boldsymbol{\tau}$
$\mathbf{m}_n^\tau$	The voxels in $\mathbf{m}$ that are smaller than $-\boldsymbol{\tau}$
$\mathbf{m}_p^{\tau,b}$	The mask for voxels that are bigger than threshold $\boldsymbol{\tau}$
$\mathbf{m}_n^{\tau,b}$	The mask for voxels that are smaller than threshold $-\boldsymbol{\tau}$

Table 2

FIX classification accuracies from Human Connectome Project data.

FIX threshold	1	2	5	10	20	30	40	50
TPR (mean)	99.9	99.8	99.7	**99.6**	99.3	98.8	98.4	97.4
TNR (mean)	96.7	97.5	98.5	**98.9**	99.3	99.5	99.6	99.7
TPR (median)	100	100	100	**100**	100	99.0	98.8	98.0
TNR (median)	97.1	97.7	98.8	**99.2**	99.5	99.6	99.7	99.8

TPR = True Positive Rate, i.e., the percentage of true signals correctly classified. TNR = True Negative Rate, i.e., the percentage of true artefacts correctly classified. As the FIX “threshold” is lowered, TPR is maximised at the expense of high TNR; an “optimal” threshold might be considered to be 5–10.

Table 3

The most commonly used kernels for nonlinear SVM

Kernel Name	Formula
Polynomial (homogeneous)	$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j)^d$
Polynomial (inhomogeneous)	$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^d$
Gaussian radial basis function (RBF)	$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$

Table 4

FIX's Temporal Features

Index	Name & Description
1	The number of independent components, as determined by MELODIC
2–3	The relationship between the order of the AR model and its goodness of fit
4–5	The parameter and the residual of AR(1)
6–8	The parameters and the residual of AR(2)
9:10	The skewness and kurtosis of the time series
11	The difference between timeseries mean and its median
12–13	Entropy (two different calculations)
14–19	Timeseries' jump characteristics
20–23	The ratio of the sum of power above *f*Hz to the sum of power below *f*Hz, for $f = 0.1, 0.15, 0.2$ and 0.25
24–30	Percent of total power that falls in 0:0.01, 0.01:0.025, 0.025:0.05, 0.05:0.1, 0.1:0.15, 0.15:0.2 and 0.2:0.25 Hz bins
31–38	Comparing the timeseries with their null model (i.e., convolving white noise with HRF)
30–44	Timeseries' correlation with motion timeseries and their derivatives
45–46	Timeseries' mean-reversion features

Table 5

FIX's Spatial Features

Index	Name & Description
47–55	Spatial maps' supra-threshold cluster-size distribution characteristics
56–61	The balance of negative and positive voxels in spatial maps
62–65	The ratio of the Z-stat to mean functional maps
66–69	Slice-wise statistics
70–73	Slice-groups' (e.g., slices with even or odd index) statistics
74–85	Spatial maps' overlap and correlation with GM, CSF and WM masks
86–87	Smoothness estimates
88–90	TFCE features
91–105	Edge-mask features
106–177	Sagittal sinus and veins mask-based features
178	Stripiness score/feature