



Johns Hopkins University, Dept. of Biostatistics Working Papers

7-14-2012

ANALYTIC PROGRAMMING WITH fMRI DATA: A QUICK-START GUIDE FOR STATISTICIANS USING R

Ani Eloyan

Johns Hopkins Bloomberg School of Public Health, aeloyan@jhsph.edu

Shanshan Li

Johns Hopkins Bloomberg School of Public Health

John Muschelli

Johns Hopkins Bloomberg School of Public Health

Jim Pekar

Kennedy Krieger Institute, Johns Hopkins University

Stewart Mostofsky

Kennedy Krieger Institute, Johns Hopkins University

See next page for additional authors

Suggested Citation

Eloyan, Ani; Li, Shanshan; Muschelli, John; Pekar, Jim; Mostofsky, Stewart; and Caffo, Brian S., "ANALYTIC PROGRAMMING WITH fMRI DATA: A QUICK-START GUIDE FOR STATISTICIANS USING R" (July 2012). *Johns Hopkins University, Dept. of Biostatistics Working Papers*. Working Paper 245.
<http://biostats.bepress.com/jhubiostat/paper245>

This working paper is hosted by The Berkeley Electronic Press (bepress) and may not be commercially reproduced without the permission of the copyright holder.

Copyright © 2011 by the authors

Authors

Ani Eloyan, Shanshan Li, John Muschelli, Jim Pekar, Stewart Mostofsky, and Brian S. Caffo

Analytic programming with fMRI data: a quick-start guide for statisticians using R

Ani Eloyan, Shanshan Li, John Muschelli,
Jim Pekar, Stewart Mostofsky, Brian S. Caffo

July 11, 2012

Abstract

Functional magnetic resonance imaging (fMRI) is a thriving field that plays an important role in medical imaging analysis, biological and neuroscience research and practice. This manuscript gives a didactic introduction to the statistical analysis of fMRI data using the R project along with the relevant R code. The goal is to give statisticians who would like to pursue research in this area a quick start for programming with fMRI data along with the available data visualization tools.

1 Introduction

This primer was created to get statisticians who are new to the field of imaging data analysis a quick overview and tools for programming their own analyses of fMRI data; targeted at those who would like to pursue the programming using the R project (R Development

Core Team, 2011). The key benefits of programming on one's own is the ability to extend analyses and build and create new tools. Of course, the key benefits of using the R project are that it's open source, cross-platform, free (as in cost), has excellent graphical capabilities and is an award-winning lingua franca of statisticians. Matlab (www.mathworks.com) is the standard scripting language in neuroimaging and signal processing, has a widely used integrated development environment and GUI creation software. Both R and Matlab are cross-platform and have excellent graphics capabilities. Both are high level scripting languages that are easy to learn and have a large collection of add-ons and subroutines.

For those wanting more automated methods for the analysis of fMRI data, several programs exist to perform the most standard analyses. In addition, power users should learn these programs for comparison purposes to avoid reinventing existing and well established tools. We give a non-exhaustive list of some of our favorite freely available software below.

SPM SPM (<http://www.fil.ion.ucl.ac.uk/spm/>) is a collection of open source Matlab scripts, often calling compiled code. SPM is arguably the most popular software for the analysis of fMRI data, in no small part due to a well developed GUI. It has methods for single-subject and multi-subject analyses and tools for displaying the results and data visualization. It can perform all of the basic preprocessing procedures of the imaging data. Moreover, it has an active user community with several user contributed add-ons. The SPM scripts are well documented, and are easily understood.

FSL FSL (<http://www.fmrib.ox.ac.uk/fsl/>) is UNIX-based software that performs sin-

gle and multi subject analyses, preprocessing and visualization of results. It has a graphical user interface, though is most effectively used at the command line, typing routines together using shell scripts. FSL is open source and can be compiled from scratch. In addition, pre-compiled binaries are available, notably for OSX. For Windows, FSL can be run in a virtual machine.

MIPAV MIPAV (<http://mipav.cit.nih.gov/>) is JAVA software created by the US National Institutes of Health. It has very broad functionality for preprocessing, analysis and visualization of imaging data. However, it is most useful through an active collection of modules.

AFNI AFNI (<http://afni.nimh.nih.gov/afni/>) is UNIX-based software. It offers a full analysis and processing suite and we often use it in conjunction with FSL and BASH scripting.

FreeSurfer FreeSurfer (<http://surfer.nmr.mgh.harvard.edu/>) is a UNIX-based software for registration (often atlas based) and analysis of medical imaging data. It is often discussed as having excellent registration and visualization tools.

In addition to the above packages for analysis and processing of fMRI data, there are R packages related to fMRI, primarily focused on analysis post-registration.

AnalyzeFMRI Analysis package for fMRI including a graphical user interface.

oro.nifti Package for reading in nifti files, a common fMRI imaging format.

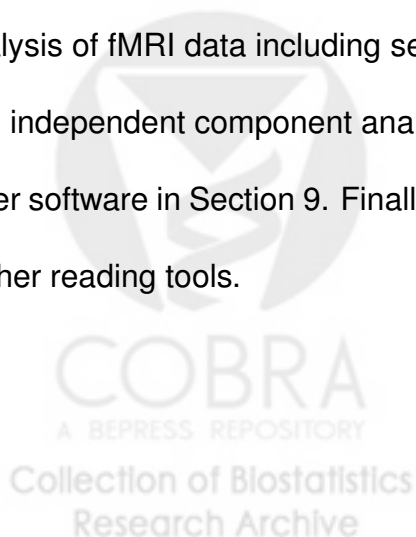
fmri Package for reading in and analyzing fMRI data. Includes data for HRF convolution models and ICA.

neuroSim Package for simulating fMRI data. As a consequence, has tools for creating and investigating designs with HRF convolution models.

Rniftilib Package for loading and writing the 3D or 4D images.

There is a frequently updated list on CRAN of packages for medical image analysis at <http://cran.r-project.org/web/views/MedicalImaging.html>.

The remainder of the manuscript is organized as follows. Sections 2, 3 and 4 describe the structure of the functional magnetic resonance imaging (fMRI) data, discuss ways of obtaining the data and give a brief overview of the steps of preprocessing of the fMRI data. Section 5 describes the different formats that can be used for viewing the fMRI data in R along with examples of a Gaussian smoother of the data matrix and obtaining the mask of the three dimensional image. A discussion of the HR function and the role of the design matrix in task based fMRI studies is discussed in Section 6, an example of voxel-wise regression is shown in Section 6.2.2. The between subjects random effect models are discussed in Section 7. Section 8 gives an overview of the connectivity based analysis of fMRI data including seed voxel based methods, singular value decomposition and independent component analysis. We discuss some of the visualization tools in R or other software in Section 9. Finally, Section 10 completes the paper by pointing out useful further reading tools.



2 fMRI

Functional magnetic resonance imaging (fMRI) is a modern brain imaging measurement technology. As its name suggests, the technology images the brain over time as it functions using a magnetic resonance imaging (MRI) machine. Standard fMRI images the so-called BOLD (blood oxygen level dependent) signal, described further below. BOLD fMRI is not the only functional brain imaging technology available using an MRI scanner. For example, arterial spin labeling (Zhou et al., 2001) is a closely related functional MRI technology, while dynamic contrast enhanced imaging (see Roberts et al., 2006; Shinohara et al., 2011) is another, very different, functional brain imaging use of an MRI scanner. However, the term fMRI, when used without qualifying statements, refers to BOLD fMRI. Briefly opening the discussion to non-magnetic resonance (MR) based technologies we note that positron emissions tomography (PET), single photon emissions computed tomography (SPECT), electroencephalograms (EEG), magnetoencephalograms (MEG) and other measurement devices can be used for functional brain imaging, each with different goals, limitations, strengths and scientific interpretations.

The BOLD signal (Ogawa et al. (1990), Turner et al. (1991), Ogawa et al. (1992), Turner et al. (1993), Aguirre et al. (1998), Logothetis et al. (2001), Logothetis and Wandell (2004)) measures the cerebral hemodynamic changes concomitant to neuronal activation of the brain. A localized increase in neural activity results in increase in blood flow (“reactive hyperemia”) to the activated region when excess of oxygenated hemoglobin is delivered to the region. Consequently, a reduction in deoxyhemoglobin concentration follows, resulting in an increase in magnetic field homogeneity. The gradient echo EPI

sequence allows to study the activation of specific regions of the brain during a task. Deoxyhemoglobin serves as an endogenous susceptibility contrast agent allowing MRI to report on local hemodynamic changes. Hence, the BOLD signal in response to a stimulus is given a name - the hemodynamic response (see Figure 1). This figure shows an initial delay of the response since it takes time for the vasculature of the brain to respond to the need for oxygen after the stimulus and then a subsequent brief undershoot is observed. The origin of the undershoot is still controversial. One explanation of the undershoot is that there is some ongoing oxygen consumption after reaching the point of origin. The other is that the excess volume of deoxygenated blood results in delayed vascular compliance.

A slight modulation of the MRI signal is called BOLD fMRI signal. Here the scanner records images every second or so (the so-called TR or time of repetition). At each time point a 3D image of the brain is obtained. In BOLD fMRI the image is collected one slice at a time for each time point instead of scanning the 3D image immediately. Hence, a slice time correction (e.g. by Fourier transformation (Aguirre et al., 1998)) is applied to the resulting images as part of the preprocessing. Empirical face validity of the methodology can be demonstrated by simple motor or visual tasks. For example, consider a so-called “finger tapping” task where a subject is asked to tap their finger while in the scanner and then rest, repeating this sequence for a certain time interval. When the images are averaged over the times at rest and compared to the times when tapping the finger, and appropriate statistical tests are performed, motor areas of the brain are clearly significant (see Witt et al. (2008) for overview of finger tapping task related analysis in the literature). By virtue of the well defined motor area of activation and ease of performing the task, finger tapping tasks are commonly used for scanner and methodological validation, as

Anatomy of the Hemodynamic Response Function

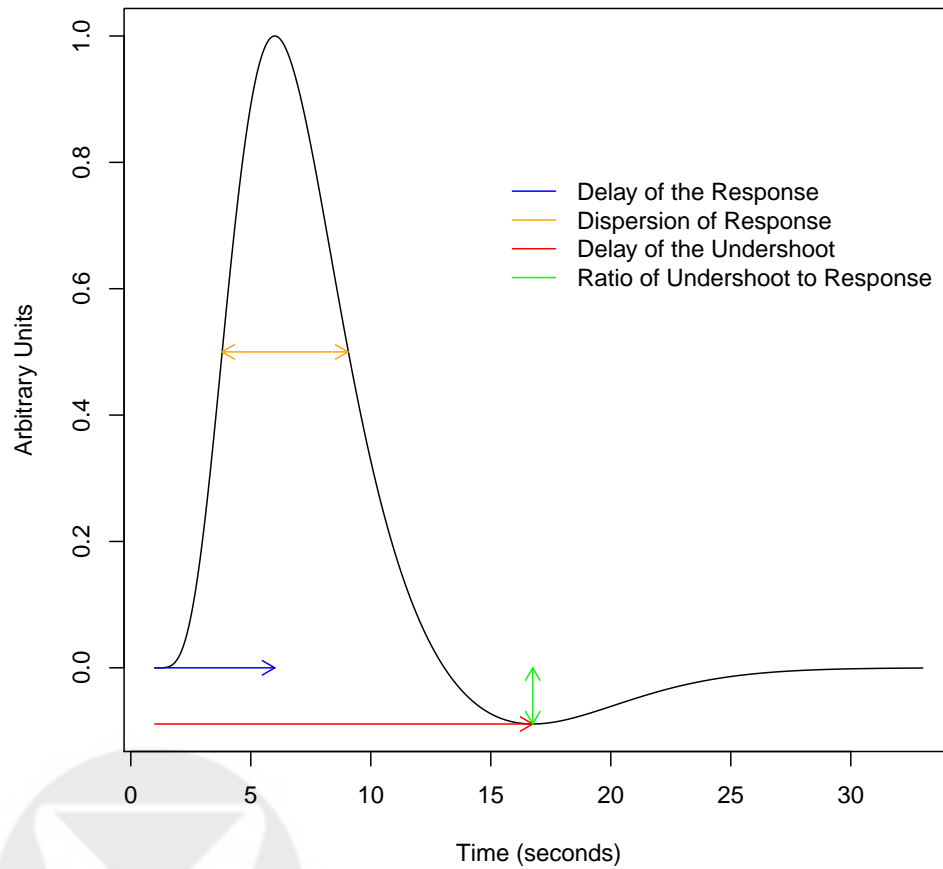


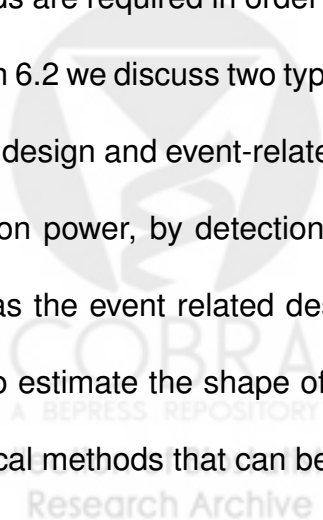
Figure 1: Anatomy of the HRF.



well as often being of intrinsic interest in the study of motor areas.

We will not thoroughly discuss the details of the methods by which the MR scanners use the principles of magnetic resonance to achieve different images. However, newcomers to the area might be surprised to find the amazing variety of biological signals that one can visualize using MR. These include separate modalities to study white matter, gray matter, overall structure, white matter tracts, tracers injected into the body and lesions in the brain. There are multiple structural and functional imaging types, all acquired using the principles of MR. See Prince and Links (2006) for a general introduction to MR imaging.

The development of the BOLD fMRI technology facilitated research in the analysis of cognitive function of the brain. For instance, one may be interested in identifying the regions in the brain that are involved in performing a mental operation or a cognitive task, how active is a brain region during a task, what is the shape of the time course of activation and quantifying the extent of connectivity between brain regions during tasks. Since there are several sources of variability introduced during data collection, statistical methods are required in order to accurately obtain the regions of interest. For instance, in Section 6.2 we discuss two types of experimental designs used frequently in fMRI analysis - block design and event-related design. It can be shown that the block designs have high detection power, by detection we refer to the ability of the method to detect activation, whereas the event related designs have better estimation efficiency in that they can be used to estimate the shape of the HR (Liu et al., 2001). In Section 6.2 we show several statistical methods that can be used to answer the questions posed above.



3 Obtaining data

As many neuroscience researchers are joining open science initiatives promoting public access for data, numerous websites offer data sets that can be downloaded freely. However, we have a few favorite canonical data sets that are good starting points for analysis. The first is SPM's example datasets that include single subject and inter-subject analyses. The second data set of interest is the NITRC 1000 Connectome project's resting state data. Most of the SPM datasets are preprocessed and ready for analysis, however, the NITRC data needs extensive preprocessing. Unix shell scripts using FSL and AFNI can be found on their website for those who are interested in preprocessing.

The SPM example data sets can be downloaded at

<http://www.fil.ion.ucl.ac.uk/spm/data/>

and (with their permission) will be used as example data in this manuscript. The NITRC example data set can be downloaded at

http://www.nitrc.org/projects/fcon_1000/

The 1,000 Connectome project data involves the most preprocessing. We therefore put a processed image in `.nii` format along with the corresponding 2D matrix in `.rda` format for download on our servers at:

www.biostat.jhsph.edu/~bcaffo/downloads/rest_res2standard.nii.gz

www.biostat.jhsph.edu/~bcaffo/downloads/imageDataMatrix.rda

Here we choose images from the Kennedy Krieger Institute (the home institution of several of the coauthors of this manuscript).

We also note that FSL has an evaluation suite, the FEEDS data set, that we have found very useful, both for learning FSL as well as debugging new methods. The FSL example data sets can be downloaded at

<http://www.fmrib.ox.ac.uk/fsl/feeds/doc/index.html>

4 Preprocessing

Preprocessing is an essential component in the analysis of fMRI data. Standard preprocessing requires, for example, skull stripping, evaluation and corrections for motion, coregistering, spatial smoothing and registration to a template. We discuss two preprocessing steps in particular, spatial smoothing and registration to a template. However, for newcomers to the area, we suggest leaving all the steps, except for possibly the spatial smoothing to standard software, or acquiring data where preprocessing has already been completed in an acceptable form. R, in particular, does not have well developed fMRI preprocessing routines in general. For those readers interested in pursuing preprocessing as a research endeavor, good theoretical foundations are given by Miller et al. (1993) and Younes (2010).

Warping the images of a subject to a template, often called spatial normalization, is the key step of (attempting) to put subject images into a common space. That is, each registered image will appear the same and hence the spatial locations of the voxels in the brain are ideally interpretable across subjects, in other words, voxel 1 for subject 1 is the same as voxel 1 for subject 2. In functional neuroimaging, this process relies on numerous assumptions. For example, it presumes anatomical similarity across subjects

at the functional resolution of interest. This assumption becomes especially problematic when studying diseased brains, where anatomy can be drastically different than healthy control template brains. Also, it presumes localization of the brain function of interest to common anatomical locales across subjects.

Setting these issues aside, normalization is accomplished by mathematically warping each subject's collection of images over time to a so-called "template". A template is typically a highly detailed structural image obtained by imaging a subject repeatedly or warping several subjects into one common image. It is important to know what template the images you are working on have been registered to. The template is especially useful for overlaying results on top of it which then helps in contextualizing the results. Because it is often a highly accurate structural image (or there is an associated image in the same space), it is useful for interpretation, as neuroscientists can visually relate findings to important structural landmarks. Moreover, some templates are in wide use and key regions have been tabulated (see Chialvo (2010) for more details).

We commonly use templates created by the Montreal Neurological Institute (MNI) (<http://imaging.mrc-cbu.cam.ac.uk/imaging/MniTalairach>). The International Consortium for Brain Mapping (ICBM) (<http://www.loni.ucla.edu/ICBM/>) releases labeled brain atlases. We should also mention the famous Talairach Atlas - a coordinate system created before MR imaging was in place. Relating results to Talairach coordinates is often done to put results in historical perspective familiar to neuroscientists. Some templates have conversion utilities to convert their coordinates to Talairach coordinates.

Finally, spatial smoothing using a filter is a step that can be easily done in R and is (or not doing so) generally a designed component of a statistician's analysis plan. Hence,

of all the preprocessing steps, this is the one we suggest omitting to be (or not to be) performed in R.

5 Loading data and structures

5.1 Array format

Two key data representations are the array format and the matrix format. Array format is the more natural format for the data. A 3D image is naturally represented as an array, say I , where $I[i, j, k]$ is the intensity of voxel i, j, k in the image. Note a 3D unit element of an image is called a **voxel** for “volumetric pixel”. If the image is collected repeatedly (like in fMRI), then 4D arrays ($I[i, j, k, t]$) are useful, though often are very large.

Medical images in neuroimaging are usually saved in one of a few available formats. However, all formats have common properties, for instance, they are binary and the data are stored as a long floating point vector. Header information includes descriptive features of the image such as the physical dimensions of a voxel and the size of each of the three dimensions, as well as other information.

DICOM DICOM is a standard image format often used as scanner output. DICOM images have the header information built into the binary file. 3D DICOM files are sometimes split into several files of 2D slices, but can be combined into a single file.

Analyze The Analyze format includes two files for each 3D image: a header file with the extension `.hdr` and an image file with the extension `.img`. The header and image files have to have the same filename and have to be saved in the same folder. The

4D images are stored as separate files, in other words the 3D image for each time point is saved as a separate file.

NIFTI Nifti files typically have the extension `.nii` and are a current standard for the analysis of fMRI data. Nifti files can have more than 3 dimensions. Often nifti files are compressed using `gzip` and so the file extension is `.nii.gz`. If your analysis program can not `gunzip` the file, simply uncompress them using any of the standard compression software tools and the result will be a standard (uncompressed) nifti file.

5.1.1 Example: reading in and spatially smoothing images using a Gaussian filter

In this example we read in a sample 3D so-called contrast image and smooth it using a 3D Gaussian filter (Mikl et al., 2008) included in the package `AnalyzeFMRI`. First we read in the data.

```
library(AnalyzeFMRI)

imageFileName <- "pathToImageFile/imageFile.img"

img <- f.read.analyze.volume(imageFileName)[,,,1]
```

Here we assume that `imageFile.img` is an Analyze image file with a header and image pair, `imageFile.hdr` and `imageFile.img`. Notice this data is 3D, yet the program reads it in as 4D, with the fourth dimension 1. Hence the array subscripting drops the dimension by 1. This data is in the format of an image and header file. It is often useful to read in the header information

```
headerFileName <- "pathToImageFile/imageFile.hdr"
```

```
hdr <- f.read.analyze.header(headerFileName)
```

We can visualize this image.

```
##visualize the image  
tempImg <- img  
tempImg[is.na(tempImg)] <- min(img, na.rm = TRUE)  
for (k in 1 : dim(img)[3])  
  image(tempImg[, ,k], axes = FALSE)
```

Note that the creation of the temporary image is only useful as the raw image stores the background as `NA`. Note also the `image` command does not scale the image relative to the actual voxel dimensions. The following smooths the image using the function `GaussSmoothArray`

```
simg <- GaussSmoothArray(tempImg,  
                           voxdim = c(3, 3, 3),  
                           ksize = 5,  
                           sigma = diag(4, 3),  
                           mask = as.integer(!is.na(img)))  
for (k in 1 : dim(img)[3])  
  image(simg[, ,k], axes = FALSE)
```

Here the mask statement contains the smoother to the non-background areas. The `voxdim` variable is the voxel dimensions, in this case three millimeters cubed. The voxel dimensions can be found using visualization programs or inspecting the `hdr` variable that

was read in. The `ksize` and `sigma` parameters are smoothing parameters. The `sigma` parameter is the standard deviation of the Gaussian kernel (specified in mm) while `ksize` is the truncation width. See the documentation for `GaussSmoothArray` for more information.

5.2 Matrix format

The array format, while useful for smoothing, is not a convenient data structure for most other analyses. Most importantly, it is not a parsimonious structure, since it includes all of the voxels outside of the brain (the background of the image). It is common to **mask** the 3D data, i.e. extract the voxels of interest, disregard the spatial structure of the voxels and put them in a long 1D vector. In fMRI, if there is one vector per time or subject, these can be stacked into a matrix for analysis. Note under this structure, any analysis must either be invariant to spatial location or the spatial location information must be kept elsewhere for back reconstruction of the image. For example, if there are 30,000 non-background voxels, one could retain the $3 \times 30,000$ matrix of location indices for those voxels.

Masks can be created in a variety of ways. They can be obtained from the template or from the data itself. The mask is an important structure to retain, as it can map a vector back to an array format for display.

In this example, we will take a collection of 3D contrast images (one per subject) and create a mask. Using the mask, we will then create the data matrix.

5.2.1 Example: creating a mask and data matrix

First we get a list of the files (with absolute paths)

```

#the directory where the images are stored

fileDir <- "PATH_TO_DIRECTORY/cons_can/"

#the collection of images

files <- dir(fileDir, pattern = "*.img", full.names= TRUE)

```

It is useful to have the image dimensions assigned to an object. We are assuming that all images are registered and have the same dimensions.

```

##get the image dimensions by reading in the first image

imageDim <- f.read.analyze.header(files[1])$dim[2 : 4]

```

Now we loop over the images and find the intersection of non-background regions across subjects. We then find the corresponding non-background indices and place them in a vector called `mask`.

```

mask3D <- array(1, imageDim)

for (file in files) {

  img <- f.read.analyze.volume(file)[,,,1]

  mask3D <- mask3D * ( !is.na(img) )

}

##get the mask as a list of indices

mask <- which( mask3D == 1 )

```

Now we loop over subjects and collect them into a matrix by grabbing the relevant indices across subjects.

```

##now cycle through the data and get the data into a matrix format

```

```

dataMatrix <-NULL

for (file in files) {

  ##load file and make 3d

  img <-f.read.analyze.volume(file)[,,1]

  dataMatrix <-rbind(dataMatrix, img[mask])

}

noSubjects <-nrow(dataMatrix)

noVoxels <-ncol(dataMatrix)

```

6 Within subject paradigm analysis

6.1 HRF modeling

In Matlab using SPM, the HRF can be explored using the function `spm_hrf`. Inspecting the function shows that the HRF is a discretized difference of two gamma functions (see Friston et al., 2007)

$$\begin{aligned}
 f(u) &= f_1(u) - f_2(u)/p_5 \\
 &= \frac{\left(\frac{T}{p_3}\right)^{p_1/p_3} u^{p_1/p_3-1} \exp(-Tu/p_3)}{\Gamma(p_1/p_3)} - \frac{\left(\frac{T}{p_4}\right)^{p_2/p_4} u^{p_2/p_4-1} \exp(-Tu/p_4)}{\Gamma(p_2/p_4)p_5},
 \end{aligned}$$

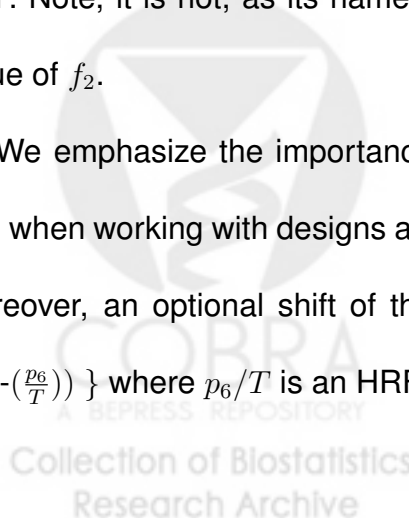
for $u > 0$. Typically, it is also normalized, either divided by its integral or maximum, to have average or peak value of 1 respectively. The logic behind this specification of the HRF is that the initial post-stimulus increase in blood oxygenation is represented by f_1 , with the

subsequent depletion represented by f_2 . Figure 1 displays an example HRF.

Notice that f_1 is a gamma density with shape p_1/p_3 and rate T/p_3 and f_2 is a gamma density with shape p_2/p_4 and rate T/p_5 . The SPM documentation defines T as the TR, p_1 as the delay of the response to the onset, p_2 as the delay of the undershoot relative to the onset, p_3 as the dispersion of the response, p_4 as the dispersion of the undershoot and p_5 as the ratio of the response to the undershoot. Conceptually, these are annotated in Figure 1.

We briefly give motivation for these parameters. Note, the mean of density f_1 is p_1/T while the mean of density f_2 is p_2/T . Hence, if p_1 and p_2 are given in seconds and T is seconds per scan, p_1/T can be thought of as the delay to the mean (not modal) response in TR units while p_2/T is the delay to the mean undershoot in TR units. Given that the gamma variance is p_1p_3/T^2 , and the mean is p_1/T , p_3/T is the factor by which the response gamma distribution mean is scaled to obtain its variance. Similarly p_4/T is the factor by which the undershoot gamma distribution mean is scaled to obtain its variance. Since the integral of f_1 and f_2 are 1, p_5 is the ratio of the two functions comprising the HRF. Note, it is not, as its name suggests, the ratio of the peak value of f_1 to the peak value of f_2 .

We emphasize the importance of keeping track of the time units (either seconds or TR) when working with designs and the HRF. The `spm_hrf` evaluates the HRF in TR units. Moreover, an optional shift of the HRF can be specified. That is, setting the HRF to $f\{u-(\frac{p_6}{T})\}$ where p_6/T is an HRF lag in TR units.



6.2 Discussion on designs

In task-based fMRI, there are two common designs: event-related and block. Event-related refers to multiple stimuli that are assumed to occur instantaneously and have some time between events. A simple task would be having the participant push a button every 20 seconds or so to look at motor cortex activation. If there is a “continuous” task being performed, such as sequential finger tapping, then the design is referred to as a block design. See Figure 6.2 to see examples of stimulus functions in each design. If the events become so frequent that they have no temporal resolution, an event-related design is essentially a block design.

Typically, the fMRI data can be modeled as a sum of response, trend and noise:

$$data = response + trend + noise.$$

The response is assumed to be a linear combination of different responses from K different stimuli. That is, the response in scan i , $i = 1, 2, \dots, N$, can be written as $x_{i1}\beta_1 + \dots + x_{iK}\beta_K$, where x_{ij} is the response from stimuli j in scan i , K is the number of stimuli and N is the number of scans. Some voxels show considerable trend over time. The trend can be linear or nonlinear hence a flexible polynomial model is often employed to allow for nonlinear effect in the trend. For example, a polynomial trend of order q for the i th scan can be written as $\sum_{l=0}^q z_l \gamma_l$, where $z_l = t^l$. Finally, a random error ϵ_i is added to the observed data, Y_i , yielding

$$Y_i = \sum_{k=1}^K x_{ik}\beta_k + \sum_{l=1}^q z_l \gamma_l + \epsilon_i.$$

Using matrix notation:

$$\mathbf{Y} = [Y_1, Y_2, \dots, Y_N]^T$$

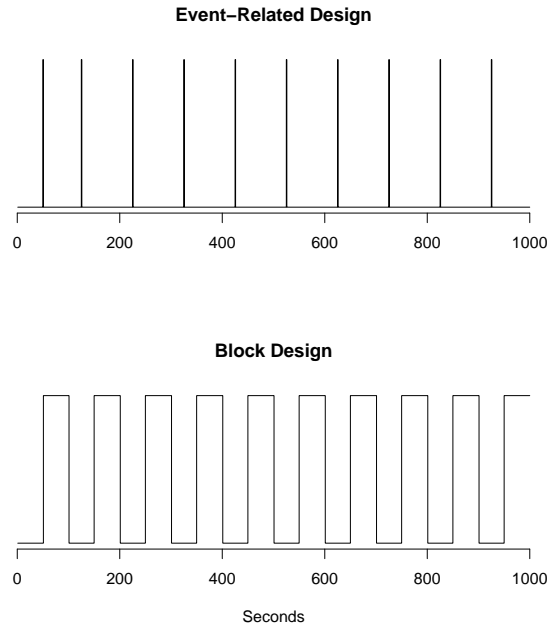
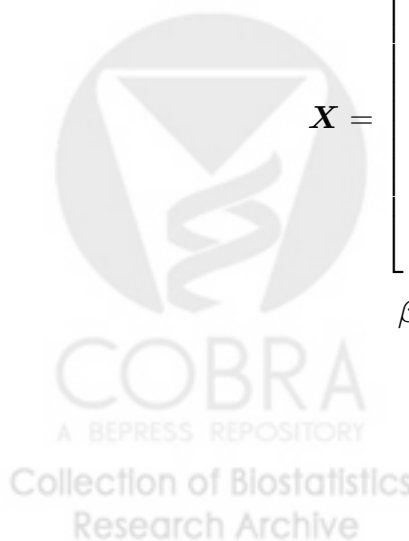


Figure 2: Stimulus vectors for Event-related and Block designs. The top figure corresponds to an event-related design, where the spikes correspond to stimuli. The spikes do not need equal spacing between events. The lower panel is a block design, where there is a sustained period of activity/task. Again, the spacing does not need to be equal in block designs, but they usually are.

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1K} & 1 & \cdots & 1 \\ x_{21} & \cdots & x_{2K} & 1 & \cdots & 2^q \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N1} & \cdots & x_{NK} & 1 & \cdots & N^q \end{bmatrix}$$

$$\boldsymbol{\beta} = [\beta_1, \dots, \beta_K, \gamma_0, \dots, \gamma_q]^T$$

$$\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \dots, \epsilon_N]^T$$



The generalized linear model can be written as

$$Y = X\beta + \epsilon.$$

Note that, in the above equation, X is derived based on specific hemodynamic assumptions, for example, Gamma, Glover, and SPM canonical HRF, and β is interpreted as the size of the hemodynamic response (see Lu et al. (2006) for more details). The errors are generally not independent in time, and the correlation structure needs to be specified with caution. In the general linear model, using a wrong correlation structure could lead to bias in coefficient estimation. A simple model for the temporal correlation is the first order autoregressive model (Worsley et al., 2002), that is, $\epsilon_i = \rho\epsilon_{i-1} + \xi_i$, where $|\rho| < 1$ and ξ_i are iid.

6.2.1 Example: creating design matrix with convolution

One way of creating a design matrix is by using the function `fmri.design` in the R package `fmri`. The function `fmridesign` in the matlab package `fmristat` can also create design matrix for a linear model with given stimuli and possible polynomial trend terms.

In the following example, we read in the .nii data, extract a 4D numeric array from the data and store it as a raw vector to reduce object size. Then we create expected BOLD signals for a given task indicator function (that is, assume onset times (in scans) and duration of ON stimulus in scans are known). The design matrix is created using the function `fmri.design`, and the hemodynamic response is evaluated in the general linear model Tabelow and Polzehl (2011).

```
library(fmri)
```

```

imageFileName <- "pathToImageFile/imageFile.nii"

img <- read.NIFTI(imageFileName)

ttt <- extract.data(img)

data <- list(ttt = writeBin(as.numeric(ttt), raw(), 4),

            mask = img$mask,

            dim = img$dim0)

class(data) <- "fmridata"

hrf <- fmri.stimulus(scans = img$dim0[4],

                   onsets = onsets,

                   durations = dur)

x <- fmri.design(hrf, order = 2)

model <- fmri.lm(data, x, keep = "all")

```

If there are m experimental stimuli, there will be m expected bold responses $hrf_1, hrf_2, \dots, hrf_m$. The design matrix can be created using

```
x <- fmri.design(cbind(hrf_1, ..., hrf_m), order = 2)
```

In the following example we pass an HRF function and the stimulus parameters to get a convolved design matrix, commonly used in general linear models (GLM) in task-based fMRI.

```

by=0.01
scans <- 100
tr = 2.5
### onsets are in seconds

```



```

onsets <- c(5.25, 21.45, 100.12, 223.5)

durations <- c(5, 5, 10, 10)

ans <- design(onsets = onsets,
              durations = durations,
              tr = tr,
              scans = scans)

```

In the code above, `by` is denoting how fine of a grid one wants to compute the HRF, in seconds. Finer grids may be required due to the level of precision of the onsets of the stimuli. The function `design` takes in the number of scans, the TR, the times and durations of stimuli in seconds and returns the resulting vector of the stimulus convolved with the HRF. The result is in TR units, so if there were 100 scans, the vector would be of length 100. For event-related designs, the durations of the events would be considered instantaneous, so the duration would be 0. The example above is an example of a block design.

6.2.2 Example: voxel-wise regression modeling

Having the design matrix we may run regression models for voxel-wise analysis of the brain. Suppose Y_i is a voxel by time matrix for person i . Then we can use the following code for GLM analysis with the design matrix above.

```

Y <- matrix(rnorm(1000*100), nrow=1000, ncol=100)
lms <- apply(Y, 1, function(x) lm(x ~ ans$res))
summs <- lapply(lms, summary)

```

```

coefs <- lapply(summs, function(x) x$coefficients)

###Taking Intercept Map

int_vals <- sapply(coefs, function(x) x["(Intercept)", ])

int_est <- int_vals["Estimate",]

int_pvals <- int_vals["Pr(>|t|)",]

beta_vals <- sapply(coefs, function(x) x["ans$res", ])

beta_est <- beta_vals["Estimate",]

beta_pvals <- beta_vals["Pr(>|t|)",]

```

Overall, the code can run linear models, extract coefficients and p-values from those linear models. These values should be in the same order as the Y matrix, so the transformation from Y back into a 3-dimensional array should be the same for these β coefficients and p-values to obtain brain maps. Afterwards, one can choose their type of multiple comparisons corrections to obtain significantly activated clusters.

7 Between subjects random effect models

In the fMRI literature, “random effect analysis” refers to an inter-subject analysis of paradigm-related contrast images. Specifically, in the first stage, subject-specific regression models are fit, relating the HRF-convolved design matrix to the fMRI time series at each voxel separately. Coefficients are obtained from the fit, or estimable linear combinations of coefficients (so-called contrasts). As there is one contrast per voxel, the resulting collection of voxel-specific contrast values are referred to as contrast images, and are usually stored as a 3D image. Random effect analysis then analyzes these images across sub-

jects. That is, comparing inter-subject variation in estimated hemodynamic response to the paradigm to variation in covariates, diagnoses, treatments, and so on. The name random effect analysis is given as the process is an approximation of formal random effects.

7.1 Example: group activation

Below, we consider a simple form of random effect analysis using the SPM data. Specifically, we consider group activation. That is, we take the subject-specific contrasts and test whether their voxel specific mean is different than 0. We'll assume that the following steps have been completed: 1) a mask containing non-background voxels has been created, 2) the mask has been applied to every subject's contrast map resulting in a vectorized map, 3) the data have been concatenated into a matrix containing `noSubjects` (number of subjects) rows and `noVoxels` (number of voxels) columns. (Code for this is included in the supplementary material.) Assume that the data matrix is assigned to the variable `dataMatrix`. In our case, there are 12 subjects and roughly 50,000 non-background voxels. As a first test, consider a test of zero mean applied at each voxel.

```
##apply the function t.test to every column (voxel)

ttest.out <- apply(dataMatrix, 2,
  function(x) {
    temp <- t.test(x)
    c(temp$statistic, temp$p.value)}})

##label the rows
rownames(ttest.out) <- c("statistic", "pvalue")
```

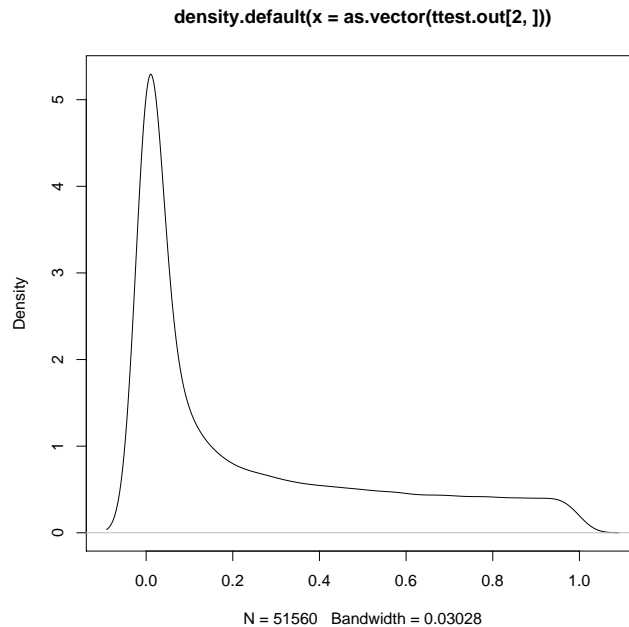


Figure 3: P-value histogram from a group-level significance test of the SPM contrast data.

Figure 7.1 displays the P-value histogram. Now threshold these P-values using AnalyzeFMRI's false discovery rate threshold

```
Threshold <- Threshold.FDR(ttest.out[1,],
                             q = .05,
                             cV.type = 2,
                             type = "t",
                             df1 = noSubjects - 1)
```

Now we can create an image where we plot P-values surviving this FDR threshold overlaid on a template brain. We often do this by outputting the results then using 3rd party software for display. Note, it is, of course, possible to do this in R, say using the `image` function. Moreover, we have previously used `rgl` to perform 3D rendering. However, note the following: the template brain and the statistical image results: 1) may not

have the same array dimensions, 2) voxels may not correspond to the same real world dimensions, such as the voxels in the template being $1mm^3$ while the voxels in the statistical image being $3mm^3$, 3) the images may not have the same origin. All of this information is contained in the raw and template image's header file. The origin is the point that is used to match images. Hence, when overlaying images if one appears shifted, it is probably due to incorrectly set values of the image origins. Another problem that can occur is that the voxels are not square. For example, the Z physical dimension (in the direction from the feet to the head), is often different than that of the other two. If this is not accounted for, the images will appear strange, not unlike a geographical map with an incorrect contrast ratio. These issues can all be handled in R, but care must be taken.

```
outputImage <- array(0, imageDim)

toSave <- ttest.out[1,] >= fdrThreshold

outputImage[mask[toSave]] <- ttest.out[1, toSave]
```

7.2 Example: permutation testing

Suppose that the mask has been created and the images have been vectorized and saved as the matrix `dataMatrix`. More details on permutation testing can be found in Nichols and Holmes (2002).

```
# Number of subjects
n.s=12

n=2^n.s

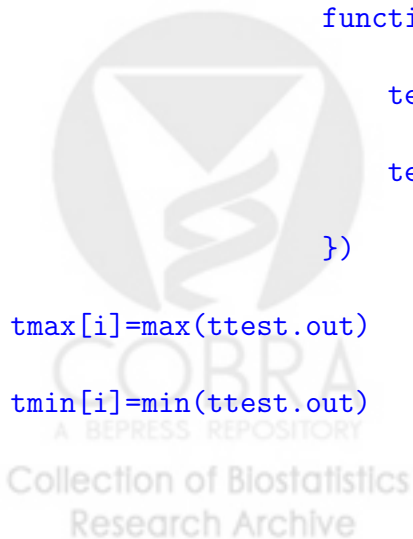
# Matrix of all possible combinations of 1 and -1 for each subject
```

```

perm.mat=matrix(1,n.s,n)
for(i in 1:n.s){
  p=n/(2^i)
  j=1
  while(j<(2^i)){
    perm.mat[i,(p*j+1):(p*(j+1))]=-1
    j=j+2
  }
}

# Get the distribution of the maximum t
# statistic for each permutation
tmax=rep(0,n)
tmin=rep(0,n)
for(i in 1:n){
  data.temp = perm.mat[,i]*dataMatrix
  ttest.out <- apply(data.temp, 2,
                    function(x) {
                      temp <- t.test(x)
                      temp$statistic
                    })
  tmax[i]=max(ttest.out)
  tmin[i]=min(ttest.out)
}

```



And then look at the histogram of the t statistic to make a decision on whether to reject the null hypothesis or not. The R function `expand.grid()` can also be used to obtain the grid for the permutation test above.

8 Connectivity analysis

Functional connectivity refers to the analysis of correlations of measured brain function between potentially remote areas of the brain (Friston et al., 2007). In fMRI, this translates to evaluating correlations, or perhaps other forms of associations between voxels or regions. We emphasize that fMRI represents only one modality to investigate functional connectivity. Others include: EEG, MEG, PET and ECoG. It is important when investigating connectivity to consider the goals of the analysis and the spatial/temporal resolution of the technology under study (Friston et al., 2007).

Such analyses have become so influential in fMRI, that they have been abbreviated as fc-fMRI (for functional connectivity fMRI). A subset of such analyses considers functional connectivity while a subject is at rest in the scanner, or so called resting state functional connectivity (rs-fc-fMRI). Much of the analysis of rs-fc-fMRI centers around the so-called “default mode network”, a hypothesized brain network that is activated in subjects at rest (Buckner et al., 2008).

8.1 Connectivity and preprocessing

The analysis of brain connectivity data requires special preprocessing that may be useful for paradigm-related studies, but is crucial for the study of connectivity. Most notably

it is addressing background nuisance signals. These include signals related to cardiac function and respiration. Because one is concerned with correlations between voxels, such background effects can create spurious relationships that are not of interest. In contrast, they are less crucial in a paradigm-related study, since the paradigm is likely not aliased with the nuisance signal, though to the contrary, they are probably not orthogonal to the paradigm. In event-related designs where events are presented randomly, the randomization mechanism helps ensure a lack of relationship with nuisance signals.

There are several methods for addressing nuisance signals in fc-fMRI. These include a connectivity analysis of the ventricles, the spaces in the brain filled with cerebrospinal fluid. There should be no measured connectivity of interest in the ventricles, hence any correlations found here and elsewhere likely represent nuisance signals. A similar approach would look at white matter, though such analyses would be highly dependent on very accurate registration.

8.2 Seed voxel approaches

Assuming that there is no nuisance signals in the brain, the correlations of the fMRI time courses can be computed for each pair of voxels in the brain. The number of these pairs in the brain is often very large and the brute force computation of the correlations may be computationally expensive. Regions of interest (ROI) are obtained (for instance, by using the anatomy of the brain or by independent component analysis as described in Section 8.3). For each ROI a seed voxel can be chosen as one of the voxels in the ROI or as the average time course of all voxels in the ROI. The correlation map of the seed voxel with

the voxels in the rest of the brain is called the connectivity map. The obtained connectivity maps may be used, for instance, to compare the connectivity of an ROI between subjects in two disease groups.

Many analyses use the Fisher R to Z transform: $\frac{1}{2} \frac{\log(1+r)}{\log(1-r)}$ where r is a correlation value, to obtain connectivity maps that should have approximately normally distributed values. Standardizing this matrix by the standard error $\frac{1}{\sqrt{T-3}}$, where T is the number of time points in the correlation, is frequently used, especially when different participants have different lengths of scans. These maps then can be analyzed using random Markov field corrections for multiple comparisons through SPM, using t-tests or the nonparametric rank-sum test, or other methods for testing differences across groups (Friston et al., 2007).

8.3 Singular Value Decomposition and Independent Component Analysis

Since the fMRI data are very large and complex, dimension reduction techniques are often used to identify important signals in the data as well as help with visualization. For example, singular value decomposition (SVD) - a well known statistical dimension reduction technique, is often used for preprocessing the data before testing or inference.

A commonly used dimension reduction method is the independent component analysis (ICA, see Hyvärinen et al. (2001) for an extensive overview). It is mainly used in fMRI to obtain functional networks in the brain. It has been implemented both for resting state and task related fMRI experiments. Here we show the use of ICA for an attention task using one of the SPM example datasets. Four different conditions were explored: fixation,

attention, no attention and stationary.

There are several methods one can use for applying ICA for these data. For instance, the R package `AnalyseFMRI` has a function `f.ica.fmri` that is based on the `fastICA` algorithm (Hyvärinen et al., 2001). Our example data are in Analyze format where each 3D scan is saved in a separate file. In order to use `f.ica.fmri` the individual scans per time point have to be combined into one array, since the argument has to be one 4D array instead of several 3D image files. Suppose that the 4D image where one of the dimensions is time and the other three are the three coordinates of the brain is saved as `fullimg.img`.

```
f = f.ica.fmri("fullimg.img",n.comp=3)
```

The `fastICA` package in R can be used directly to obtain ICA analysis as follows. Suppose `filesImg` is a vector of the names of image files for each time point. We create a matrix `img` that is a 2D (time by voxel) version of the fMRI data (using the `mask` that can be obtained as described in Section 5.2.1) and apply ICA directly to `img`.

```
t=length(filesImg)
img=matrix(0,t,d)
for(i in 1:t){
  f1=f.read.analyze.volume(filesImg[i]),,,1
  f=as.vector(f1[mask==1])
  img[i,]=f
}
f=fastICA(t(img),n.comp=3)
```

The variable $f\$S$ is the collection of the independent components. Next, we back-reconstruct the components as images. The networks are identified as the regions that are highly activated in the corresponding independent component. In other words, to obtain the brain networks we can threshold the values of each column in $f\$S$. We will use a simple thresholding tool here by including the values greater than a threshold θ . We also save the arrays back to image format for visualization.

```
theta=2

S=pmax(abs(x$S),theta)

S[S==theta]=0

img=array(0,c(53,63,46,m))

for(i in 1:m){

  temp2=array(0,dim(mask))

  temp2[mask==1]=S[,i]

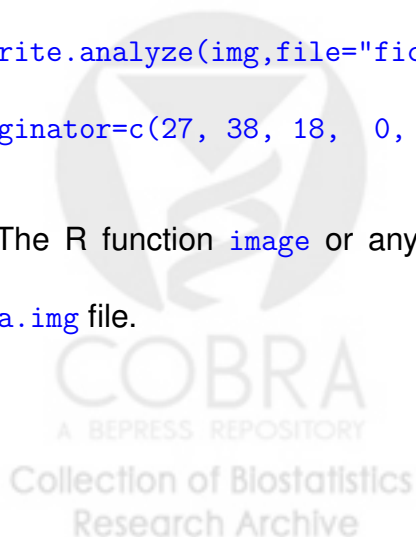
  img[,,,i]=temp2[,,]

}

f.write.analyze(img,file="fica",size="float",pixdim=c(3, 3, 3, 1, 0, 0, 0),

originator=c(27, 38, 18, 0, 0))
```

The R function `image` or any other visualization software can be used to plot the `fica.img` file.



8.4 Group ICA

Group ICA is the extension of the ICA method above to obtain brain networks from fMRI datasets from a population. The method was first introduced by Calhoun et al. (2001) and has been since widely used in the neuroimaging literature. The method is based on concatenating the two dimensional time by voxel matrices observed for each subject and applying principal component analysis to reduce the dimension to the desired number of components. Then the single subject ICA algorithm is applied to the resulting matrix to obtain the common independent components for the group.

There are two different ways of matrix concatenation: one assumes that the spatial brain networks are statistically independent (spatial group ICA), the other assumes that the time courses are statistically independent (temporal group ICA). Spatial group ICA is more frequently used, which assumes common spatial networks across subjects yet different temporal mixing matrices. By common spatial maps assumption, one can concatenate all subjects' data in the temporal domain, and apply ICA to the aggregated data matrix.

Below is an example for spatial group ICA:

```
mask = f.read.analyze.volume(fileMask)[,,1]
groupDataMatrix <- NULL
for (file in files) {
  img <- f.read.analyze.volume(file)
  imageDataMatrix <- t(apply(img, 4, function(imgi) imgi[mask]))
  rmeans <- rowMeans(imageDataMatrix)
```

```

cmeans <- colMeans(imageDataMatrix)

imageDataMatrix <- sweep(imageDataMatrix, 2, cmeans, "-")

imageDataMatrix <- sweep(imageDataMatrix, 1, rmeans, "-")

groupDataMatrix <- rbind(groupDataMatrix , imageDataMatrix)
}

m=20

f=fastICA(t(groupDataMatrix),n.comp=m)

```

9 Displaying results

Different modalities can be used for displaying the results obtained by the analysis of fMRI data described above. In most cases a group of voxels in the brain is found by the analyses and can be visualized by overlaying the voxels on a template brain. For instance, in the case of ICA the functional networks obtained can be displayed by overlaying the networks on a template brain. Some of the issues with the originator of the image and the correct voxel size used in the template brain and the image overlayed on it are discussed in Section 7.1. In this section, we will discuss some of the functions that can be used for visualization in R, Matlab or using other software.

9.1 Within R

The function `image()` in R can be used for plotting the 2D images for each value of the third dimension to display the images for each slice of the brain. The `rgl` and `misc3d`

packages have tools for creating 3D renderings and isosurfaces directly in R. We have found that images generally need to be slightly downsampled to avoid memory issues. Usually, this does not impact the appearance of the graphics dramatically.

9.2 Third party software

There are several visualization software that can be used for looking at the fMRI data. They vary from software for quick visualization to 3D renderings of the brain with complicated visualizations of slices and cross-sections. For instance 3D Slicer and Mipav can be used to obtain 3D renderings of the brain overlaid on template brains. These software can also be used for masking the brain or cropping regions of interest as well as registration. The videos at <http://www.youtube.com/watch?v=GNsWRnm7gQw> show a step by step illustration for using 3D Slicer or Mipav for visualization and analyses.

10 Further reading and discussion

Below we give a woefully incomplete list of papers in the area.

- A good starting point for fMRI is the SPM book Friston et al. (2007).
- Nicole Lazar's introduction to fMRI Lazar et al. (2001) and book Lazar (2008).
- Martin Lindquist's Statistical Science manuscript on fMRI Lindquist (2008)
- Thomas Nichol's overview paper on permutation testing in fMRI Nichols and Holmes (2002).

- Vincent Calhoun's introductions to ICA and group ICA in fMRI Calhoun et al. (2003, 2001)
- Jonathan Taylor's manuscript about the HRF Worsley and Taylor (2006)
- Highly technical papers about random field theory are Siegmund and Worsley (1995); Worsley (1995). A more accessible discussion including permutation testing can be found in Hayasaka et al. (2004) and similarly Worsley et al. (1996).
- A somewhat dated but still very relevant, overview of the analysis of fMRI data Worsley (1997).
- A more recent tutorial can be found here Smith (2004).
- An expository article on R and neuroimaging Tabelow et al. (2011)
- Our work on SVD for fMRI Caffo et al. (2010).

This manuscript gives an admittedly very brief introduction and quick start guide for fMRI. For those interested in pursuing further development in the area we recommend investigating processing pipelines as a next important step.

References

- G. K. Aguirre, E. Zarahn, and M. D'Esposito. The variability of human, BOLD hemodynamic responses. *Neuroimage*, 8(4):360–369, 1998.
- R. L. Buckner, J.R. Andrews-Hanna, and D. L. Schacter. The brain's default network: Anatomy, function, and relevance to disease. *Ann. N.Y. Acad. Sci.*, 1124:1–38, 2008.

- B. S. Caffo, C. M. Crainiceanu, G. Verduzco, S. Joel, S. H. Mostofsky, S. S. Bassett, and J. J. Pekar. Two-stage decompositions for the analysis of functional connectivity for fMRI with application to Alzheimer's disease risk. *NeuroImage*, 51(3):1140–1149, 2010.
- V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar. A method for making group inferences from functional MRI data using independent component analysis. *Human Brain Mapping*, 14(3):140–151, 2001.
- V. D. Calhoun, T. Adali, L. K. Hansen, J. Larsen, and J. J. Pekar. ICA of functional MRI data: an overview. In *Fourth International Symposium on Independent Component Analysis and Blind Source Separation*, pages 281–288, 2003.
- D. R. Chialvo. Emergent complex neural dynamics. *Nature Physics*, 2010.
- K. J. Friston, J. Ashburner, K. Stefan, T. E. Nichols, and W. Penny, editors. *Statistical Parametric Mapping The Analysis of Functional Brain Images*. Academic Press, 2007.
- S. Hayasaka, K. L. Phan, I. Liberzon, K. J. Worsley, and T. E. Nichols. Nonstationary cluster-size inference with random field and permutation methods. *Neuroimage*, 22(2): 676–687, 2004.
- A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Inter-Science, 2001.
- N. A. Lazar. *The statistical analysis of functional MRI data*. Springer Verlag, 2008.

- N. A. Lazar, W. F. Eddy, C. R. Genovese, and J. Welling. Statistical issues in fMRI for brain imaging. *International Statistical Review*, 69(1):105–127, 2001.
- M. A. Lindquist. The statistical analysis of fMRI data. *Statistical Science*, 23(4):439–464, 2008.
- T. Liu, L. Frank, E. Wong, and R. Buxton. Detection power, estimation efficiency, and predictability in event-related fmri. *NeuroImage*, 13(4):759–773, 2001.
- N. Logothetis, J. Pauls, M. Augath, T. Trinath, A. Oeltermann, et al. Neurophysiological investigation of the basis of the fmri signal. *Nature*, 412(6843):150–157, 2001.
- N. K. Logothetis and B. A. Wandell. Interpreting the BOLD signal. *Annual Review Physiology*, 66:735–769, 2004.
- Y. Lu, A. P. Bagshaw, C. Grova, E. Kobayashi, F. Dubeau, and J. Gotman. Using voxel-specific hemodynamic response function in eeg-fmri data analysis. *NeuroImage*, 32:238–247, 2006.
- M. Mikl, R. Mareček, P. Hlušík, and M. Pavlicová. Effects of spatial smoothing on fmri group inferences. *Magnetic Resonance Imaging*, 26:490–503, 2008.
- M. I. Miller, G. E. Christensen, Y. Amit, and U. Grenander. Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Sciences*, 90(24):11944, 1993.
- T. E. Nichols and A. P. Holmes. Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human Brain Mapping*, 15(1):1–25, 2002.

- S. Ogawa, T. Lee, A. Nayak, and P. Glynn. Oxygenation-sensitive contrast in magnetic resonance image of rodent brain at high magnetic fields. *Magnetic Resonance in Medicine*, 14(1):68–78, 1990.
- S. Ogawa, D. Tank, R. Menon, J. Ellermann, S. Kim, H. Merkle, and K. Ugurbil. Intrinsic signal changes accompanying sensory stimulation: functional brain mapping with magnetic resonance imaging. *Proceedings of the National Academy of Sciences*, 89(13):5951, 1992.
- J. L. Prince and J. M. Links. *Medical imaging signals and systems*. Pearson Prentice Hall, 2006.
- R Development Core Team. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL <http://www.R-project.org/>. ISBN 3-900051-07-0.
- C. Roberts, B. Issa, A. Stone, A. Jackson, J. C. Waterton, and G. J. M. Parker. Comparative study into the robustness of compartmental modeling and model-free analysis in DCE-MRI studies. *Journal of Magnetic Resonance Imaging*, 23(4):554–563, 2006.
- R. T. Shinohara, C. M. Crainiceanu, B. S. Caffo, M. Gaitan, and D. Reich. Quantifying cerebral hemodynamics in populations of multiple sclerosis patients. *Technical report*, 2011.
- D. O. Siegmund and K. J. Worsley. Testing for a signal with unknown location and scale in a stationary gaussian random field. *The Annals of Statistics*, 23(2):608–639, 1995.

- S. M. Smith. Overview of fMRI analysis. *British Journal of Radiology*, 77(Special Issue 2):S167, 2004.
- K. Tabelow and J. Polzehl. Statistical parametric maps for functional mri experiments in r: The package fmri. *Journal of Statistical Software*, 44(11):1–21, 2011.
- K. Tabelow, J. D. Clayden, P. Lafaye de Micheaux, J. Polzehl, V. J. Schmid, and B. Whitcher. Image analysis and statistical inference in neuroimaging with R. *Neuroimage*, 2011.
- R. Turner, D. Bihan, C. Moonen, D. Despres, and J. Frank. Echo-planar time course mri of cat brain oxygenation changes. *Magnetic Resonance in Medicine*, 22(1):159–166, 1991.
- R. Turner, P. Jezzard, H. Wen, K. Kwong, D. Le Bihan, T. Zeffiro, and R. Balaban. Functional mapping of the human visual cortex at 4 and 1.5 tesla using deoxygenation contrast epi. *Magnetic resonance in medicine*, 29(2):277–279, 1993.
- S. T. Witt, A. R. Laird, and M. E. Meyeranda. Functional neuroimaging correlates of finger-tapping task variations: An ale meta-analysis. *NeuroImage*, 42:343–356, 2008.
- K. J. Worsley. Estimating the number of peaks in a random field using the Hadwiger characteristic of excursion sets, with applications to medical images. *The Annals of Statistics*, 23(2):640–669, 1995.
- K. J. Worsley. An overview and some new developments in the statistical analysis of PET and fMRI data. *Human Brain Mapping*, 5(4):254–258, 1997.

- K. J. Worsley and J. E. Taylor. Detecting fMRI activation allowing for unknown latency of the hemodynamic response. *Neuroimage*, 29(2):649–654, 2006.
- K. J. Worsley, S. Marrett, P. Neelin, A. C. Vandal, K. J. Friston, and A. C. Evans. A unified statistical approach for determining significant signals in images of cerebral activation. *Human Brain Mapping*, 4(1):58–73, 1996.
- K. J. Worsley, C. H. Liao, J. Aston, V. Petre, G. H. Duncan, F. Morales, and A. C. Evans. A general statistical analysis for fmri data. *NeuroImage*, 15(1):1–15, 2002.
- L. Younes. *Shapes and Diffeomorphisms*. Springer Verlag, 2010.
- J. Zhou, D. A. Wilson, J. A. Ulatowski, R. J. Traystman, and P. C. M. van Zijl. Two-compartment exchange model for perfusion quantification using arterial spin tagging. *Journal of Cerebral Blood Flow & Metabolism*, 21(4):440–455, 2001.

A Appendix

A.1 Glossary of terms

Axial See “transverse”.

BOLD Blood oxygen level dependent. Refers to the target of measurement of fMRI.

Boxcar Referring to fMRI paradigms where blocks (boxcars) of stimuli are presented.

Coronal Anatomical orientation same direction as the face.

Event related design Referring to a design where stimuli are presented separately.

fc-fMRI Functional connectivity fMRI. The study of correlations in brain signals.

fMRI Functional magnetic resonance imaging. Using an MRI scanner to image brain function typically through the BOLD signal.

HRF Hemodynamic response function. The function that relates the BOLD signal to neural stimuli.

ICA Independent components analysis; a tool for

rs-fc-fMRI Resting state fc-fMRI. Investigating functional connectivity through fMRI while subjects are at rest in the scanner.

Sagittal Anatomical direction looking from the side, toward the ear.

TR Time of repetition. The time between MRI scans.

Transverse Also see “axial”, anatomical orientation representing a plane parallel to the floor for a person standing on the ground.

