Weierstraß-Institut

für Angewandte Analysis und Stochastik

im Forschungsverbund Berlin e. V.

Preprint ISSN 0946 – 8633

Statistical parametric maps for functional MRI experiments in R: The package fmri

Karsten Tabelow[1], Jörg Polzehl[1]

submitted: November 9, 2010

[1] Weierstrass Institute
for Applied Analysis and Stochastics,
Mohrenstr. 39, 10117
Berlin, Germany
E-Mail: karsten.tabelow@wias-berlin.de
E-Mail: joerg.polzehl@wias-berlin.de

No. 1562

Berlin 2010

2010 *Mathematics Subject Classification.* 62P10, 92C55, 62G05, 62G10 .

Key words and phrases. functional magnetic resonance imaging, structural adaptive smoothing, structural adaptive segmentation, random field theory, multiscale testing.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 2044975
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract

The package **fmri** is provided for analysis of single run functional Magnetic Resonance Imaging data. It implements structural adaptive smoothing methods with signal detection for adaptive noise reduction which avoids blurring of edges of activation areas. **fmri** provides fmri analysis from time series modeling to signal detection and publication-ready images.

1 Introduction

Neuroscience is a very active field that combines challenging scientific questions with interesting methodological developments and draws expertise from as different fields as biology, medicine, physics, mathematics, statistics, and computer science. In fact, the emergence of medical imaging techniques have triggered a boost of developments for medical or scientific applications in particular for the examination of the human brain. One of them is functional Magnetic Resonance Imaging (fMRI) which due to its non-invasive character has become the most informative tool for in-vivo examination of human brain function on small spatial scales. It is nowadays utilized both in research as well as in clinical applications such as diagnosis and treatment of brain lesions.

As fMRI suffers from noise its reduction plays a very important role. Gaussian filtering is applied in standard analysis mostly to increase the signal to noise ratio (SNR) and the sensitivity of statistical tests. At the same time smoothing reduces the number of independent decisions, relaxes the severe multiple test problem and leads to a situation where critical values for signal detection can be assigned using Random Field Theory (RFT). The inherent blurring effect can be ignored as long as the precise shape and extent of the activation area is not important. However, as experiments become more and more sophisticated and explore e.g. columnar functional structures in the brain or functions near brain lesions, adaptive noise reduction methods become essential. For a comprehensive introduction into statistical issues in fMRI we refer to Lazar (2008).

In the R-package **fmri** (Tabelow and Polzehl, 2010b) two algorithms from a special class of *structural adaptive smoothing* methods are implemented together with appropriate signal detection: (a) structural adaptive smoothing with RFT for signal detection and (b) structural adaptive segmentation based on multiscale tests. The package **fmri** (version 1.0-0) has been first reported in Polzehl and Tabelow (2007). Since then, the package has evolved significantly and now includes more features, more advanced methodology, and a Graphical User Interface (GUI). Here, we will report on these updates which refer to the package version 1.4-2. For an introduction into the R environment for statistical computing (R Development Core Team, 2010) see e.g. Dalgaard (2008) or the material on `http://www.r-project.org/`. For the use of R in medical imaging problems we refer the reader to the taskview *Medical Image Analysis* on `http://cran.r-project.org/`.

2 Analyzing fMRI experiments

The package **fmri** analyzes the fMRI time series using the general linear modeling approach, see e.g. Friston *et al.* (2007). The focus of the package is on the use of structural adaptive smoothing methods for statistical parametric maps (SPM). Structural adaptive smoothing in this context allows for increasing the sensitivity of signal detection while limiting the loss in spatial resolution. For an alternative modeling of fMRI data using Independent Component Analysis (ICA) we refer to the package **AnalyzeFMRI** (Marchini and de Micheaux, 2010).

2.1 Data formats

fMRI experiments typically acquire time series of full three dimensional brain volumes when a specific stimulus or task is presented or performed, respectively. The package **fmri** provides functionality to read medical imaging data from several formats. It may occasionally be preferable to convert the data using external tools, e.g. *afni* (Cox, 1996), *MRIcro* (Rorden and Brett, 2000), to name only two, or use more specialized packages, e.g. **oro.dicom** (Whitcher, 2010) or **oro.nifti** (Whitcher *et al.*, 2010) for data input. Objects containing fMRI data in the format used within **fmri** can then be easily generated, see documentation and the description below.

The quasi-standard format for data recorded by a MR scanner is DICOM® (Digital Imaging and Communications in Medicine; `http://medical.nema.org`)[1]. The package **fmri** provides a function for reading DICOM files:

```
R> ds <- read.DICOM(<filename>)
```

where `<filename>` is the name of the file. A data description, without actually reading the data, can be obtained by setting the logical argument `includedata` to `FALSE`. The value returned by the function `read.DICOM` is a list `ds` containing the full header information as a list with the four-byte sequences (group,element) of the DICOM format as elements, the data (if read), and specific header information, e.g. voxel size, image dimension, which are of special interest.

Other formats are more common in the context of fMRI. The ANALYZE format developed at the Mayo Clinic is comprised of two files, the “img” and “hdr” files, containing the image data and information about the image acquisition, respectively. Using the **fmri** package the file `<filename>` is read by

```
R> ds <- read.ANALYZE(<filename>)
```

where `<filename>` contains the data as 4D volume. This can be achieved by using many third-party tools including *MRICro* (Rorden and Brett, 2000). Data prepared for the analysis software *SPM* (Ashburner *et al.*, 2008) generally comes as a series of 3D volume with filenames `<prefix+3digits+postfix>`. It is therefore possible to read this series using

```
R> ds <- read.ANALYZE(prefix = "", numbered = TRUE,
+                     postfix = "", picstart = 1,
+                     numbpic = 1)
```

[1] DICOM is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.

where `picstart` is the first number in the digits series and `numbpic` is the number of volumes.

The ANALYZE format has been further adapted to the NIfTI format by the DFWG (Data Format Working Group) with improvements in extending the header information and the possibility to merge both files into one (.nii). Reading these files is performed by

```
R> ds <- read.NIFTI(<filename>).
```

Note, that currently (version 1.4-2) compressed NIfTI files cannot be handled by **fmri**. Last but not least

```
R> ds <- read.AFNI(<filename>)
```

reads file pairs (HEAD/BRIK) used by *afni* (Cox, 1996).

The input functions for ANALYZE, NIfTI, and AFNI files may possess additional arguments `setmask` and `level`. These can be used to classify voxel within and outside the brain based on the mean voxel intensity. If `setmask` is `TRUE` (default) a threshold is set as the `level`-quantile of the mean voxel intensities. Voxel with mean intensity exceeding this threshold are considered to be within the brain. Further computations are restricted to such voxel.

The list object `ds` exploits S3 classes and has class `"fmridata"`. The data cube is stored within the list element `ttt` as a raw vector generated by

```
writeBin(as.numeric(ttt), raw(), 4)
```

to reduce object size. The data can be extracted from object `ds` in form of a 4D numeric array by

```
R> ttt <- extract.data(ds).
```

The complete header information is contained in the element `header`. Other list elements contain information of special interest (voxel size, data dimension, etc.), see the according help for a complete list. If other packages like **oro.nifti** are used for reading the medical imaging data, an object `ds` as described above can be easily created and further processed by **fmri**.

fMRI datasets can be very large. Desktop computer memory may thus be a limited resource. Therefore a restriction of the analysis to a special region-of-interest (ROI) may be required and useful. **fmri** provides the function

```
R> dsselected <- cutroi(ds, xind, yind, zind, tind)
```

to create an object containing only a subcube, defined by the indices `xind`, `yind`, `zind` and `tind`, of the data.

The function `summary` provides basic information on objects, for example

```
R> ds <- read.NIFTI("Imagination.nii")
R> summary(ds)
```

generates a short characterization of the second dataset (see 4.3):

```
Data Dimension: 64 64 30 105
Data Range    : 0 ... 2934
Voxel Size    : 3.75 3.75 4
File(s) Imagination.nii
```

2.2 Data preprocessing

Data from fMRI experiments usually needs to be prepared ahead of the analysis. For example head motion during the time course of the experiment leads to mis-registration of voxel within the data cube at different time points. For group analysis the individual brain has to be mapped (normalized) onto a "standard brain" to identify corresponding brain sections.

There are currently (version 1.4-2) no tools for preprocessing steps like motion-correction, registration, normalization *within* the package **fmri**, as there exist many tools to perform these steps fast and efficiently in advance. FMRI analysis with package **fmri** is currently restricted to single subject analysis. Functions to perform group comparisons are planned for future versions.

As all imaging modalities, fMRI suffers from noise of different origin, rendering subsequent signal detection difficult. In order to increase sensitivity smoothing of fMRI data is usually part of fMRI preprocessing. It is very important to note, that within the package **fmri** smoothing should **not** be applied as preprocessing step. Instead structural adaptive smoothing, which is the main feature of the package, is applied to the statistical parametric map (SPM) derived from the linear model for the time series. If the user intends to use the package **fmri** it is advisable not to smooth the data with third-party tools. Doing so would lead to a loss of essential information and impose a spatial correlation structure, which reduces the effect of the structural adaptive smoothing used in the package and described in this paper.

2.3 Linear modeling

The observation that voxel with increased neuronal activity are characterized by a higher oxygenation level (Ogawa *et al.*, 1990, 1992) is known as BOLD (Blood Oxygenation Level Dependent) effect. In fMRI it can be used as a natural contrast and, together with fact that nuclear magnetic resonance is free of high energy radiation, forms the basis of its non-invasive character.

The expected BOLD response corresponding to the stimulus or task of the fMRI experiment can be modeled by a convolution of the task indicator function with the hemodynamic response function. This function characterizes the time delay and different form of the response in blood oxygenation. Within the package **fmri** we model the hemodynamic response function $h(t)$ as the difference of two gamma functions following the proposal in Glover (1999):

$$h(t) = \left(\frac{t}{d_1}\right)^{a_1} \exp\left(-\frac{t-d_1}{b_1}\right) - c\left(\frac{t}{d_2}\right)^{a_2} \exp\left(-\frac{t-d_2}{b_2}\right)$$

with default parameters $a_1 = 6$, $a_2 = 12$, $b_1 = 0.9$, $b_2 = 0.9$, and $d_i = a_i b_i$, $i = 1, 2$, $c = 0.35$ and t the time in seconds. Given the stimulus $s(t)$ as a task indicator function, the expected BOLD response is calculated as convolution of $s(t)$ and $h(t)$:

$$x(t) = \int_0^\infty h(u)s(t-u)du.$$

The resulting function $x(t)$ is evaluated at the T scan acquisition times. Such expected BOLD signals can be created by the function

```
R> hrf <- fmri.stimulus(scans = 1,
+                       onsets = c(1), durations = c(1),
+                       rt = 3, times= NULL, mean = TRUE,
+                       a1 = 6, a2 = 12, b1 = 0.9, b2 = 0.9,
+                       cc = 0.35).
```

`hrf` is a vector of length T. `onsets` are the scan numbers of the task or stimulus onsets, `durations` is the vector of stimulus durations in number of scans. `rt` is the time between subsequent volumes. The design can be given in seconds by providing the argument `times`, in this case `durations` should also be given in seconds. Additionally one can change the parameters of $h(t)$ to use a slightly different form for the hemodynamic response function. The vector `hrf` containes the values of the expected BOLD response at the T scan acquisition times. Note, that slice time correction and local modeling of $h(t$ are currently not available within **fmri**.

Figure 1 shows the vector `hrf` created by `fmri.stimulus()` using

```
R> hrf <- fmri.stimulus(105, c(16, 46, 76), 15, 2)
```

and used for the analysis of dataset II in Subsection 4.3.

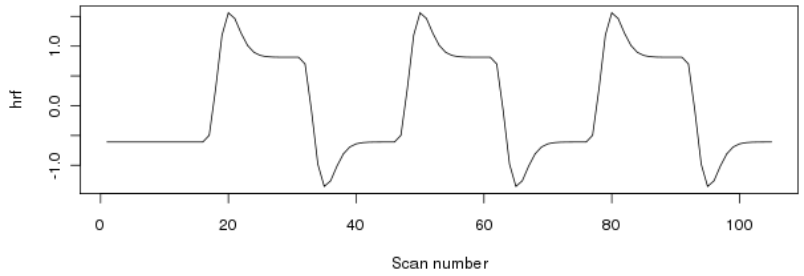

Figure 1 – *Expected BOLD response for the second dataset used in this paper*

Alternatively, a vector of length T (or number of scans) describing an expected BOLD response may be supplied and used if appropriate.

For **fmri** we adopt the common view (see e.g. Friston *et al.*, 1995; Worsley *et al.*, 2002) of a linear model for the time series $Y_i = (Y_{it})$ in each voxel i after reconstruction of the raw data and motion correction.

$$Y_i = X\beta_i + \varepsilon_i, \quad \mathsf{E}\varepsilon_i = 0_T \quad \mathsf{Cov}\varepsilon_i = \Sigma_i \tag{1}$$

where X denotes a design matrix, containing the expected bold responses `hrf1`, `hrf2`, ..., `hrfq` corresponding to q experimental stimuli as columns. Additional columns represent a mean effect and a polynomial trend of specified order. Using X we may also model $s \geq 0$ confounding effects `ce` like respiration or heart beat. The columns of X model the trend are chosen to be orthogonal to stimuli effects. The parameters $\beta_{i,j} j = 1, \ldots, q$ describe the effect of stimuli j for voxel i. The additive errors $\epsilon_i = (\epsilon_{i1}, \ldots \epsilon_{iT})$ are assumed to have zero expectation and variance depending on the underlying tissue in voxel i.

The design matrix X can be created using

```
R> x <- fmri.design(cbind(hrf1, ..., hrfq, ce1, ...), order = 2)
```

where order is the `order` of the polynomial trend. Similar to `hrf`, `x` is simply a $p \times T$ matrix ($p = q + s + order$) and could also be provided directly.

The linear model is evaluated for the fMRI data in `ds` by

```
R> spm <- fmri.lm(ds, x, contrast = c(1))
```

In order to access the variability of the estimates of β_i correctly we have to model the correlation structure in Σ_i. In the package **fmri** we assume an autoregressive model of order 1. The auto-correlation coefficients ρ_i are estimated from the residual vector $r_i = (r_{i1}, \ldots, r_{iT})$ of the fitted model (1) assuming independence, i.e. $\Sigma_i = \sigma_i^2 I_T$. We apply the bias correction given by Worsley *et al.* (2002). Temporal correlation is usually assumed to vary smoothly in space. Therefore we, by default, smooth the map of auto-correlation coefficients using a Gaussian filter in order to stabilize the estimates and to increase the number of degrees of freedom (Worsley, 2005). Using the resulting estimate $\hat{\rho}_i$ we define a transformation matrix $A_i = \hat{R}^{-1/2}(\hat{\rho}_i)$, where $\hat{R}(\rho)$ denotes the correlation matrix under the assumption of an AR(1) process with parameter ρ for voxel i.

The model (1) is then transformed into a linear model

$$\tilde{Y}_{it} = \tilde{X}_i \beta_i + \tilde{\varepsilon}_{it} \tag{2}$$

using $\tilde{Y}_i = A_i Y_i$, $\tilde{X}_i = A_i X$, and $\tilde{\varepsilon}_i = A_i \varepsilon_i$ with Cov $\varepsilon_i \approx \sigma_i^2 I_T$. Estimation of the parameters of interest $\tilde{\beta}_i$ is now straightforward by

$$\tilde{\beta}_i = (\tilde{X}_i^T \tilde{X}_i)^{-1} \tilde{X}_i^T \tilde{Y}_i.$$

The error variance σ_i^2 is estimated from the residuals $\tilde{r}_i$ of the linear model (2) by $\tilde{\sigma}_i^2 = \sum_1^T \tilde{r}_{it}^2/(T-p)$ leading to voxelwise estimated covariance matrices

$$\text{Cov } \tilde{\beta}_i = \tilde{\sigma}_i^2 (\tilde{X}_i^T \tilde{X}_i)^{-1}.$$

Usually special interest is in one of the parameters β_i or a contrast of parameters $\gamma_i = c^\top \beta_i$ represented by a vector c. Hence the result of the parameter estimation in the linear model are two three dimensional arrays $\tilde{\Gamma}$ and $\tilde{S}$ containing the estimated effects $\tilde{\gamma}_i = c^T \tilde{\beta}_i$ and their estimated standard deviations $\tilde{s}_i = \sqrt{c^T \text{Var } \tilde{\beta}_i c}$.

The function `fmri.lm()` generates a list object with class attribute `"fmrispm"` containing arrays $\tilde{\Gamma}$ and $\tilde{S}^2$ as components `beta` and `var`, respectively, and optionally residual information in `"raw"` format, similar to the imaging data. One can use the `extract.data()` function with argument `what="residuals"` to extract the residuals from the object. Note, that changing the contrast of interest c requires re-estimation of the linear model using a new value for the argument `contrast` of `fmri.lm()`.

The voxelwise quotient $\tilde{\theta}_i = \tilde{\gamma}_i / \tilde{s}_i$ of both arrays forms a statistical parametric map (SPM) $\tilde{\Theta}$ which is approximately a random t-field (see Worsley, 1994). Note that the number of degrees of freedom within the t-Field depends on the of temporal autocorrelation and the smoothing of the AR(1) coefficients, see again (Worsley, 2005). All arrays carry a correlation structure induced by the spatial correlation in the fMRI data.

$\tilde{\Theta}$ can now be used for a voxelwise signal detection classifying a voxel i as activated if the corresponding value in the $\tilde{\theta}_i$ exceeds a critical value. In this case the expected BOLD signal (or contrast) is detected to significantly deviate from zero. There are typically two intrinsic problems for the signal detection. The first is related to the large number of multiple tests in the data cube leading to a large number of false positive detections if the significance level is not adjusted for multiple testing. Possible solutions are Bonferroni corrections, assuming independence, thresholds obtained from Random Field Theory (RFT, Adler, 2000; Worsley, 1994), or the use of the concept of false discovery rate (FDR, Benjamini and Hochberg, 1995; Benjamini and Heller, 2007). The second problem is related to small values in $\tilde{\Theta}$ caused by large error variance leading to low sensitivity of the tests. For both problems smoothing of $\tilde{\Gamma}$ provides a solution, as

it reduces the number of independent tests and ideally reduces the variance of the estimated parameters or contrast. In the next section we will review two structural adaptive smoothing methods which are the core of the package **fmri**.

3 Structural adaptive data processing

3.1 Structural adaptive smoothing

The most common smoothing method for functional MRI data is the Gaussian filter: It can be easily applied using fast Fourier transform (FFT) and guarantees the assumption for RFT which require a certain smoothness in the data. Unfortunately non-adaptive smoothing leads to significant blurring and thus to less specificity in the detection of the spatial extent and form of the activation areas in fMRI. There are several applications, where the blurring renders subsequent medical decisions or neuroscientific analysis difficult if not impossible, e.g. pre-surgical planning for brain tumor resection or columnar functional structures to name only two. We therefore proposed structural adaptive smoothing methods (Tabelow *et al.*, 2006; Polzehl *et al.*, 2010) that overcome these drawbacks and allow for increased sensitivity for signal detection while not blurring the borders of the activation areas.

Smoothing is generally considered as a preprocessing step for the data like motion-correction etc. However, except for effects from prewhitening, the order in which non-adaptive spatial smoothing and evaluation of the linear model are performed is arbitrary. Moreover, if the temporal correlations are spatially homogeneous the temporal modeling and spatial smoothing can be interchanged.

Structural adaptive smoothing uses local homogeneity tests for the adaptation (Polzehl and Spokoiny, 2006). These tests are inefficient in the four dimensional data space. Structural adaptive smoothing is therefore based on the estimates for $\tilde{\Gamma}$ and $\tilde{S}$ obtained by the temporal modeling using `fmri.lm()`. The parameter estimation in the linear model serves as a variance and dimension reduction step prior to spatial smoothing and therefore allows for a much better adaptation (Tabelow *et al.*, 2006).

Structural adaptive smoothing requires an assumption on the local homogeneity structure of the true parameter γ_i. We assume a local constant function for γ_i, as non-activated areas are characterized by a parameter value zero. In areas which are activated during the scan the parameter values differ from zero and are similar, provided that the BOLD %-changes are similar. Other models are possible, but require more computational effort. $H_0 : \beta_i = 0$ serves as the null hypothesis for the statistical test of activation.

Based on this assumption and the arrays $\tilde{\Gamma}$ and $\tilde{S}$, we use an iterative smoothing algorithm for the statistical parametric map (SPM) that is based on pairwise tests of homogeneity. We specify kernel functions $K_l(x) = (1-x^2)_+$ and $K_s(x) = \min(1, 2(1-x)_+)$. For K_l this choice is motivated by the kernels near efficiency in nonadaptive smoothing, and it's compact support and simplicity which reduces the computational effort. The second kernel should exhibit a plateu near zero, be compactly supported and monoton non-increasing on the positive axis. Let $\mathcal{H} = \{h_k\}_{k=1}^{k^*}$, with $h_0 = 1$, be a series of bandwidths generated such that $\sum_j K_l\left(\frac{d(i,j)}{h_k}\right)$, with $d(i,j)$ denoting a distance between voxel i and j, forms a geometric sequence with factor $c_h = 1.25$. At iteration step k we define for a each voxel i and all voxel j within a distance $d(i,j) < h_k$ weights

$$w_{ij}^{(k)} = K_l\left(\frac{d(i,j)}{h_k}\right) K_s\left(s_{ij}^{(k-1)}\right),$$

compute locally smoothed estimates

$$\tilde{\gamma}_i^{(k)} = \frac{1}{N_i^{(k)}} \sum_j w_{ij}^{(k)} \tilde{\gamma}_j, \quad N_i^{(k)} = \sum_j w_{ij}^{(k)}.$$

$$s_{ij}^{(k-1)} = \frac{N_i^{(k-1)}}{\lambda C(k, \rho_s) \tilde{s}_i^2} \left(\tilde{\gamma}_j^{(k-1)} - \tilde{\gamma}_i^{(k-1)} \right)^2$$

is a statistical penalty evaluating the statistical difference between the estimated parameters in voxel i and j. The parameter λ has been determined by simulation depending on the error model (but not the actual data) using a propagation condition (Polzehl and Spokoiny, 2006). This condition states that an expected loss for the adaptive estimates at each iteration should not exceed the corresponding loss of its non-adaptive counterpart, i.e. the estimate with $\lambda = \infty$, by more than a factor $1 + \alpha$. This automatically controls the bias of the estimates by their standard deviation (Polzehl and Spokoiny, 2006). λ determines the degree of adaptation for the procedure.

Structural adaptive smoothing provides an intrinsic stopping criterion, since the quality of estimates is preserved at later steps of the algorithm. The term $C(k, \rho_s)$ provides an adjustment for the effect of spatial correlation in the original data, characterized by a vector of first order correlation coefficients in coordinate directions ρ_s, at iteration k. Optionally, specifying `adaptation = "fullaws"` for the function `fmri.smooth()`, in each iteration the term $\frac{C(k,\rho_s)\tilde{s}_i^2}{N_i^{(k-1)}}$ is replaced by an estimated variance $\hat{\sigma}_i^{2;(k-1)}$ obtained from the smoothed residuals $\tilde{r}_{it}^{(k-1)} = \frac{1}{N_i^{(k-1)}} \sum_j w_{ij}^{(k-1)} \tilde{r}_{jt}$. This leads to additional computational cost but usually improves results. Finally the variances of the smoothed estimates (contrasts) are estimated from smoothed residuals using the weighting scheme $w_{ij}^{(k^*)}$ from the final iteration k^*.

The structural adaptive smoothing procedure leads to estimates $\tilde{\Gamma}^{k^*}$ and $\tilde{S}^{2(k^*)}$ and a smoothed SPM $\tilde{\Theta}^{k^*}$. In these statistics shape and borders of the activation structure are preserved. As a consequence, in contrast to non-adaptive smoothing methods, the procedure does reduce noise while preserving the resolution of the scan as required by many modern applications (see Tabelow *et al.*, 2009).

The number of iteration steps k^* determaines a maximum achievable variance reduction or equivalently a maximal achievable smoothness. It can be specified by selecting the maximum bandwidth h_{max} in the series h_k as the expected diameter of the largest area of activation. Oversmoothing structural borders is avoided in the algorithm by construction as long as differences between the parameter values of two homogeneity regions are statistically significant at any scale, i.e. for any bandwidth h_k, visited within the iterations. The largest homogeneous region is expected to be the non-activation area, where parameter values do not significantly differ from zero. Within this area, i.e. under the hypothesis of no activation, structural adaptive smoothing, due to the propagation condition, behaves like its non-adaptive counterpart with $\lambda = \infty$.

Structural adaptive smoothing within the package **fmri** is performed by

```
> spm.smooth <- fmri.smooth(spm, hmax = 4, adaptation = "aws")
```

Note, that usually only the `hmax` argument and the choice of the adaptation method (`"none"` for non-adaptive smoothing, `"aws"`, `"fullaws"` for using improved variance estimates, and `"segment"` for the structural adaptive segmentation described below) should be specified. Additional parameters of the function have only minor influence on the results and should be considered for experts usage only.

From the final estimates for $k = k^\star$ the random t-field $\tilde{\Theta}^{(k^\star)} = \tilde{\Gamma}^{(k^\star)}/(\widehat{\mathrm{Var}}\tilde{\Gamma}^{(k^\star)})^{1/2}$ can be constructed. Under the null hypotheses of no activation the propagation condition ensures a smoothness corresponding to the application of a non-adaptive filter with the bandwidth $h_{max} = h_{k^\star}$. To define thresholds for signal detection on such a smooth t-field Random Field Theory (Adler, 2000) is applied (Tabelow *et al.*, 2006) in the package **fmri**.

Signal detection is performed by function `fmri.pvalue` applied to an object of class `"fmrispm"`.

```
> pvalue <- fmri.pvalue(spm.smooth)
```

creates a list object of class `"fmripvalue"` which contains the p-values at each voxel in the list element `pvalues`.

The result can be plotted by

```
> plot(pvalue, anatomic = NULL, maxpvalue = 0.05)
```

with a possible anatomic MR image used as an underlay `anatomic` and choosing a significance level `maxpvalue` for the multiple test corrected p-values. As only NIfTI (and DICOM) format (may) provide comprehensive orientation information only for these formats a matching with T1 (T2) image is implemented. If `anatomic` is simply an array of the same size as the functional data, it will be used as underlay.

3.2 Structural adaptive segmentation

Structural adaptive smoothing and subsequent signal detection by Random Field Theory forms a sequential procedure that depends on the similar behaviour of the adaptive method and non-adaptive smoothing under the null hypothesis of no activation. It is desirable to combine adaptive smoothing and signal detection in a way that solves the multiple-comparison problem, and to directly tackle the noise and multiple test problem of fMRI analysis at once. Such integration is possible since the information used to generate weighting schemes in the structural adaptive procedure (Tabelow *et al.*, 2006) can also be used for signal detection.

The resulting method is called structural adaptive segmentation (Polzehl *et al.*, 2010). This method leads to similar signal detection results, but is conceptually more coherent and uses much less approximating assumptions concerning the signal detection. It also provides a more computationally efficient algorithm.

Let V be a region-of-interest (ROI). We construct a test for the hypothesis

$$H : \max_{i\in V} \gamma_i \leq \delta \qquad (\text{or} \quad \max_{i\in V} |\gamma_i| \leq \delta). \tag{3}$$

δ corresponds to a suitable minimal signal size for γ. The probability to reject the hypothesis in *any* voxel $i \in V$ should be less or equal a prescribed significance level α.

Using results from extreme value theory (Resnick, 1987; Polzehl *et al.*, 2010) and ideas from multiscale testing (Dümbgen and Spokoiny, 2001) a suitable test statistics for this hypotheses is

$$T(\tilde{\Gamma}^{\mathcal{H}}) = \max_{h\in\mathcal{H}} \max_{i\in V} \frac{\left(\tilde{\gamma}_i^{(h)} - \delta\right)}{a_{n(h)}(\nu)\hat{s}_i^{(h)}} - \frac{b_{n(h)}(\nu)}{a_{n(h)}(\nu)}$$

where the number of degrees of freedom ν of the t-statistics $\tilde{\theta}_i$ is, in the nonadaptive case, the shape parameter of the limiting Fréchet distribution. The normalizing sequences $a_n(h)(\nu)$ and $b_n(h)(\nu)$ are selected to achieve, under the hypothesis H, $\delta = 0$ and $\lambda = \infty$, a good approximation of the distribution of $T(\hat{\Gamma}^{\mathcal{H}})$ by the Fréchet distribution Φ_ν.

The adaptive segmentation algorithm employs in each iteration step a test statistic

$$T(\hat{\Gamma})^{(k)} = \max_{i \in V} \frac{\left(\hat{\gamma}_i^{(h)} - \delta\right)}{a_{n_i^{(k)}(h)}(\nu)\hat{s}_i^{(h)}} - \frac{b_{n_i^{(k)}(h)}(\nu)}{a_{n_i^{(k)}}(h)(\nu)},$$

where $n_i(h) \leq n(h)$ reflects the effect of adaptive weights, to decide whether to reject the null hypothesis at iteration k. Adaptive smoothing is performed as long as the hypothesis is not rejected. Otherwise non-adaptive smoothing is restricted to the set of detected deviances from the hypothesis in either positive or negative direction. Critical values are determined by simulation under the null hypothesis as quantile of the distribution of

$$\tilde{T}(\hat{\Gamma}^{\mathcal{H}}) = \max_k T(\hat{\Gamma})^{(k)}$$

for a wide range of n, ν and suitable values of λ. A minimum value of λ has again been selected to fulfill a propagation condition under the null hypothesis. For details see Polzehl *et al.* (2010).

Structural adaptive segmentation combines the structural adaptive smoothing algorithm described in the preceding section with a test using $T(\hat{\Gamma})^{(k)}$ at iteration k. The result are three segments, one, where the hypotheses of no activation could not be rejected, and two segments for the two-sided alternatives. Note, that no p-values are determined but the method directly uses a specified significance level α for determining the critical values for the test.

In **fmri** structural adaptive segmentation can be performed on an object `"fmrispm"` from the linear model `fmri.lm()` by choosing `adaptation = "segment"` in

```
> spm.segment <- fmri.smooth(spm, hmax = 4,
+                            adaptation = "segment",
+                            alpha = 0.05)
```

The (mutliple test corrected) significance level `alpha` may be chosen within the range 0.01 to 0.2. The result of the structural adaptive segmentation is information on three segments that can be visualized using:

```
> plot(spm.segment)
```

Within activated areas the size of the estimated signal is provided as additional color coded information.

4 Example

4.1 Methods

In our first example we use an auditory dataset recorded by Geraint Rees under the direction of Karl Friston and the FIL methods group which is available from the SPM website (Ashburner *et al.*, 2008). 96 acquisitions were made with scan to scan repeat time RT=7s, in blocks of 6, resulting in 16 blocks of 42s

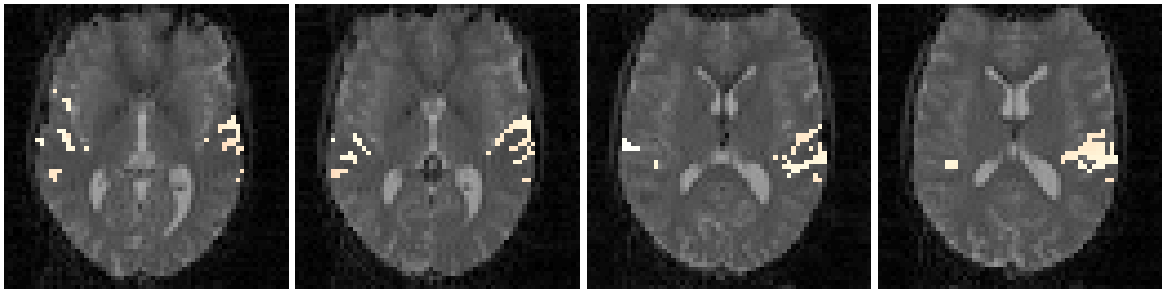

Figure 2 – *Result of signal detection using structural adaptive segmentation and the dataset described above (slices 33, 34, 36, and 37). The color scheme codes the size of the estimated signal in a voxel where a activated segment has been detected by the algorithm. The underlay is the first volume of the time series.*

duration. The condition for successive blocks alternated between rest and auditory stimulation, starting with rest. Auditory stimulation was with bi-syllabic words presented binaurally at a rate of 60 per minute. The functional data starts at acquisition 4, image `fM00223_004`. EPI images were acquired on a modified 2T Siemens MAGNETOM Vision system. Each acquisition consisted of 64 contiguous slices with matrix size 64×64 and 3mm^3 isotropic voxel.

For the second example a sports imagination task fMRI scan was performed by one healthy adult female subject within a research protocol approved by the institutional review board of Weill Cornell Medical College. For functional MRI, a GE-EPI sequence with TE/TR = 40/2000 ms was used and 30 axial slices of 4 mm thickness were acquired on a 3T GE system. A field-of-view of 24 cm with a matrix size 64×64, yielding voxel dimensions of 3.75 mm, respectively, was used. The excitation flip angle was 80 degrees. Task and rest blocks had a duration of 30 s and were played out in the following order: rest, task, rest, task, rest, task, rest, totalling 105 repetitions. Before the first block, 6 dummy scans where acquired to allow for saturation equilibrium. The task consisted of imagination of playing tennis.

For the anatomy, a sagittal 3D MP-RAGE scan was acquired (matrix 256 x 160 x 110, resampled to 256 x 256 x 110, 24 cm FOV, 1.5 mm slice thickness, TR = 8.6 ms, TE = 1.772 ms, delay time = 725 ms, flip angle = 7 degrees).

4.2 Dataset 1

The following script (spm.txt) can be used to process the first dataset. As an alternative the package provides a graphical user interface (GUI), see Section 5, to guide the user through the analysis.

```
library(fmri)

ds <- read.ANALYZE("fM00223/fM00223_", TRUE, "", 4, 96)
anatomic <- extract.data(ds)[,,,1]

hrf <- fmri.stimulus(96, (0:7)*12+7, 6, 7)
x <- fmri.design(hrf)

spm <- fmri.lm(ds, x)

spm.seg <- fmri.smooth(spm, hmax = 4, adaptation="segment")
plot(spm.seg, anatomic)
```

Images can be directly exported from the GUI invoked by `plot()`. The results for some slices can be seen in Figure 2.

4.3 Dataset 2

The following script (imagination.txt) can be used to process the second dataset:

```
library(fmri)
library(adimpro)

counter <- c(paste("0",1:9,sep=""),paste(10:99,sep=""))

ds <- read.NIFTI("Imagination.nii")

scans <- 105
onsets <- c(16, 46, 76)
duration <- 15
tr <- 2

hrf <- fmri.stimulus(scans, onsets, duration, tr)
x <- fmri.design(hrf)

spm <- fmri.lm(ds, x)

spm.segment <- fmri.smooth(spm, hmax = 4, adaptation = "segment")

ds.ana <- read.NIFTI("MPRAGEco.nii")
for (slice in 1:30) {
  img <- plot(spm.segment, ds.ana, slice = slice)
  write.image(make.image(img, gammatype="ITU"),
              file=paste("result", counter[slice], ".png", sep=""))
}
```

The result ing images of this script are shown in Figure 3 As demonstrated in the script using the `plot()` function and the package **adimpro** one can produce publication ready images in PNG, JPEG and other formats using anatomical underlay. Note, that not all medical imaging formats support orientation of the datacube in scanner space so this feature is only implemented for anatomic underlay in NIfTI format. However, many standard tools exist to provide the data in this format.

5 Organizing computational work flow (GUI)

In order to provide a user friendly environment we created a GUI (Graphical User Interface) which guides the user through the work flow described above. The GUI does not provide the full functionality of the package but gives quick results in standard analysis settings.

After invoking the package **fmri** the GUI can be started

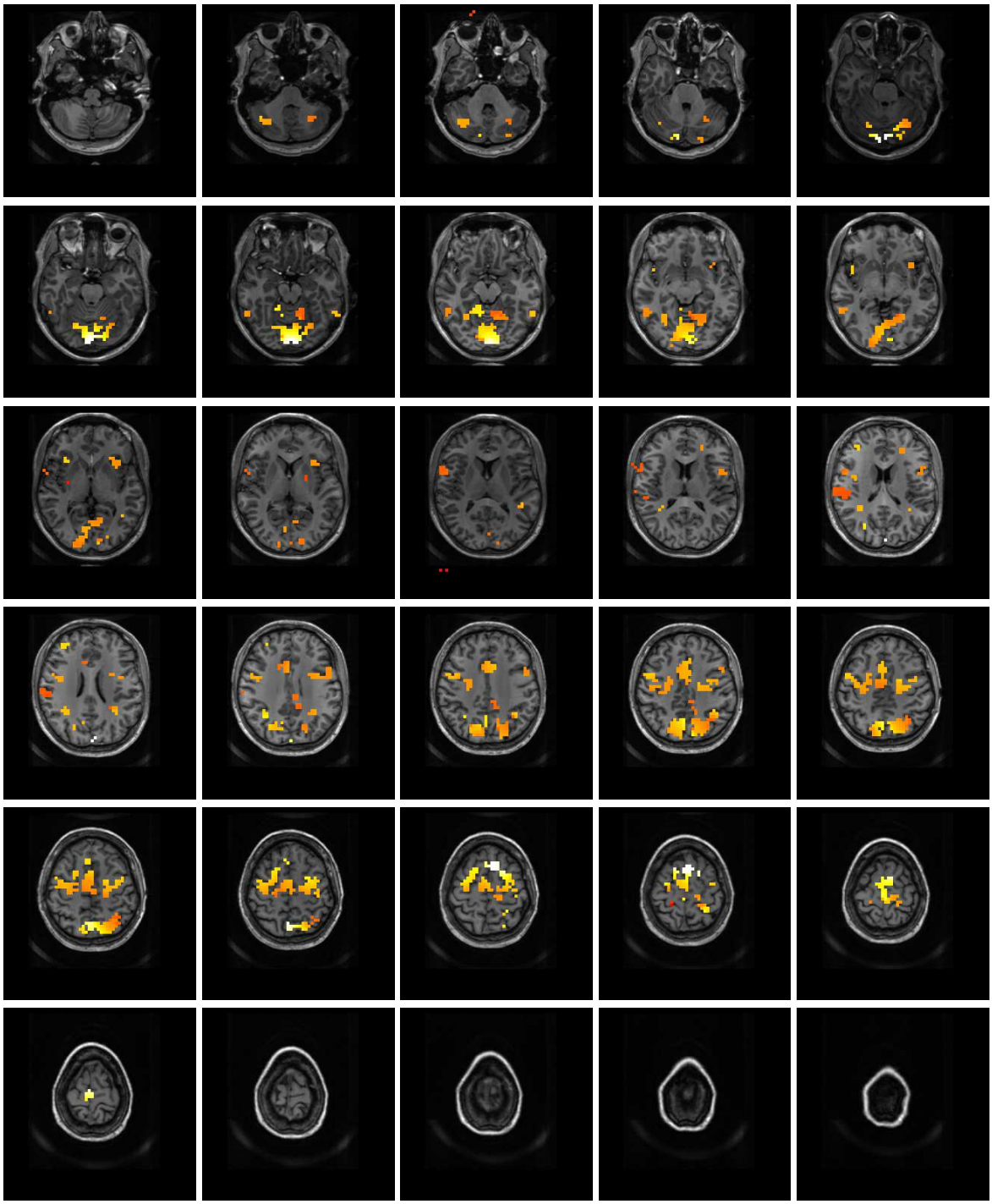

Figure 3 – *Result of signal detection using structural adaptive segmentation and the second dataset described above. The color scheme codes the size of the estimated signal in a voxel where an activated segment has been detected by the algorithm. The underlay is a sagittal 3D MP-RAGE scan.*

```
> library(fmri)
> fmrigui()
```

and guides step by step through the analysis. At each step of the analysis help information is available. The following design information needs to be provided to perform the analyisis for the first dataset described above.

```
interscan intervals:                    7
scans per session:                     96
Time unit (design) scans or seconds? scans
number of conditions:                   1
condition name:                  Auditory
onset times:        7 19 31 43 55 67 79 91
duration:                               6
```

The information can be saved in a file and re-used. Data can be accessed in form of AFNI, ANALYZE and NIfTI files. In case of our example use the file select box to navigate to the directory containing the ANALYZE files and select the first file within the directory, in our case file `fM00223_004.hdr` in folder `fM00223`. The data are read and a basic consistency check with the experimental design is performed.

In the next step a threshold for mean image intensity is proposed. This threshold is used to define a mask of voxel with mean intensity larger than the threshold. The mask should contain voxel within the brain so that computations can be restricted to voxel within this mask. The button **"`View Mask`"** provides images of the mask defined by the specified threshold together with density plots of image intensities for centered data cubes of varying size. This information is provided to assist the selection of appropriate thresholds.

Next a contrast is to be specified as a vector separated by commata or blanks. Trailing zeros may be omitted. Given this information the statistical parametric map and the corresponding variance estimates are computed following the steps described in section 2.3 using the function `fmri.lm()`.

Finally a significance level (default: 0.05) needs to be specified. This is used within the adaptive segmentation algorithm, see 3.2, or to specify the threshold for multiple test correction using RFT after structural adaptive smoothing. Finally adaptive smoothing, see 3.1, or adaptive segmentation, see 3.2, using a specified bandwidth may be performed.

Figure 4 provides a snapshot of the GUI. Note that, if needed the GUI can be closed with or without saving the current results, i.e. copying its local environment. This enables to continue the analysis using the full functionality of the package from console.

In a last step results can be visualized. The results view in 3D shows axial, saggital and coronal slices with activations. The 2D view has more features, see Figure 5. Sliders can be used to go navigate the slices, while the view can be changed from axial to coronal, or saggital. For convenience by default only 4 slices are shown. This can be changed using the number of slices and number of slices per page fields. Use `Change View/Slices` to apply the changes.

The `Extract images` button enables to create images in JPEG or PNG format. Image processing is done using the package **adimpro** (Tabelow and Polzehl, 2010a).

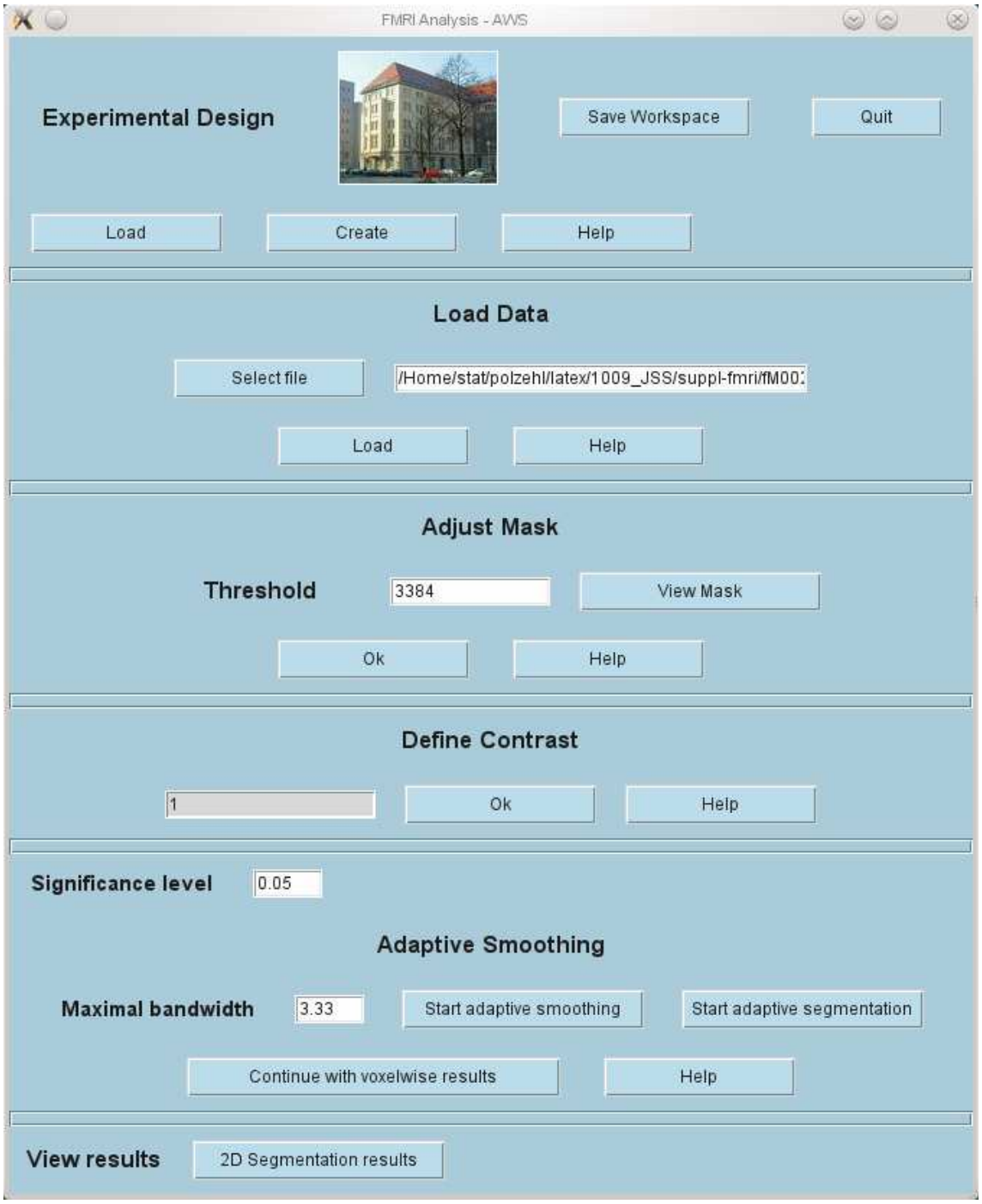

Figure 4 – *Snapshot of the* **fmri** *GUI after performing all steps.*

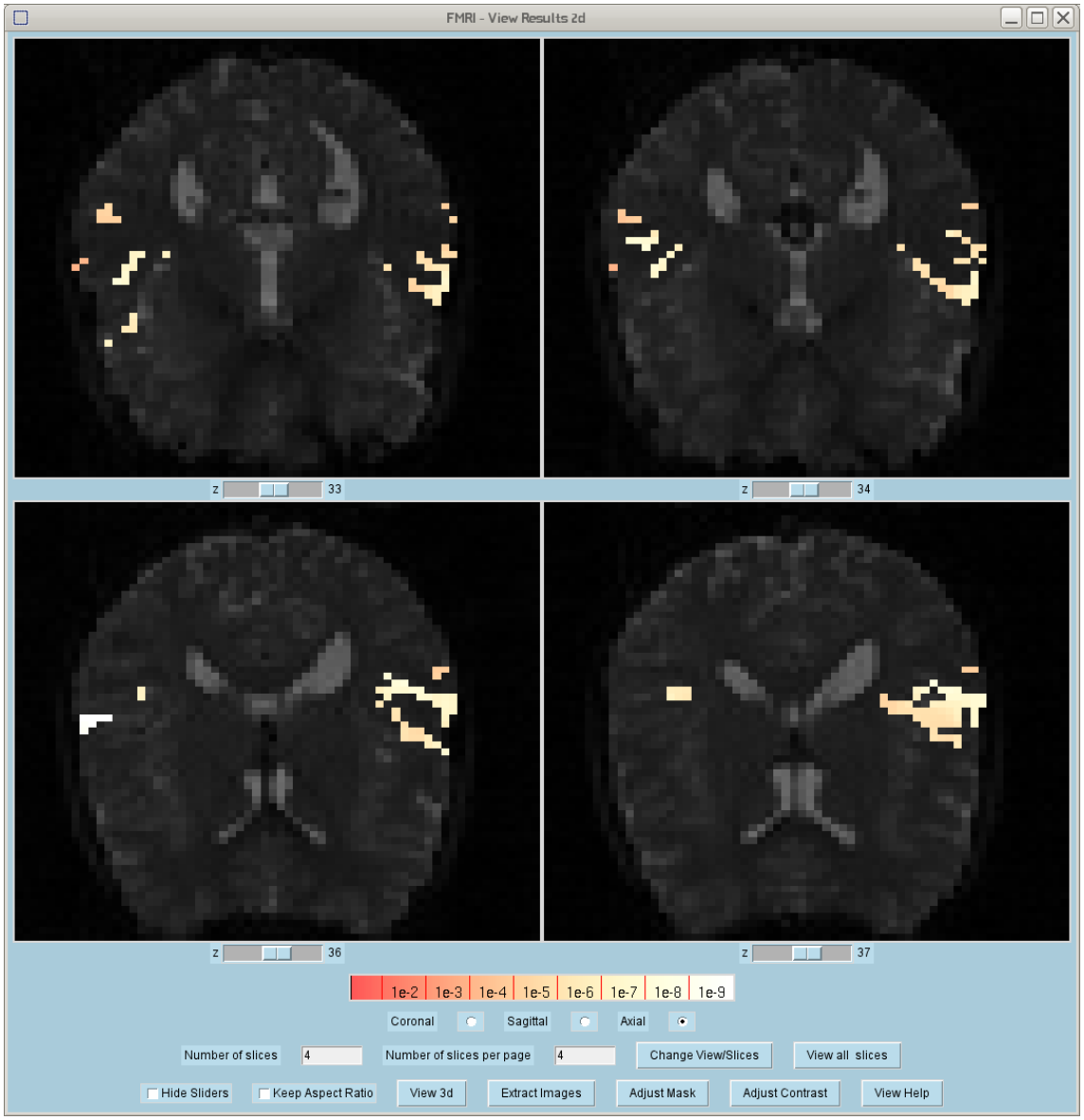

Figure 5 – *Snapshot of the results window of the GUI. This is the same as invoked by the* `plot()` *function.*

Acknowledgments

This work is supported by the DFG Research Center MATHEON. The authors would like to thank H.U. Voss at the Citigroup Biomedical Imaging Center, Weill Cornell Medical College for providing the sports imagination fMRI dataset used within this paper. The authors would also like to thank H.U. Voss for numerous intense and helpful discussions on magnetic resonance imaging and related issues.

References

Adler RJ (2000). "On excursion sets, tube formulae, and maxima of random fields." *The Annals of Applied Probability*, **10**, 1–74.

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

Benjamini Y, Heller R (2007). "False Discovery Rates for Spatial Signals." *JASA*, **102**, 1272–1281.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society, Series B*, **57**, 289–300.

Cox RW (1996). "AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages." *Computers and Biomedical Research*, **29**, 162–173.

Dalgaard P (2008). *Introductory Statistics with R*. 2nd edition. Springer.

Dümbgen L, Spokoiny V (2001). "Multiscale testing of qualitative hypotheses." *Annals of Statistics*, **29**, 124–152.

Friston KJ, Ashburner J, Kiebel S, Nichols TE, Penny WD (eds.) (2007). *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Elsevier.

Friston KJ, Holmes AP, Worsley KJ, Poline JB, Frith CD, Frackowiak RSJ (1995). "Statistical parametric maps in functional imaging: A general linear approach." *Human Brain Mapping*, **2**, 189–210.

Glover GH (1999). "Deconvolution of impulse response in event-related BOLD fMRI." *Neuroimage*, **9**, 416–429.

Lazar NA (2008). *The Statistical Analysis of Functional MRI Data*. Statistics for Biology and Health. Springer.

Marchini JL, de Micheaux PL (2010). *AnalyzeFMRI: Functions for analysis of fMRI datasets stored in the ANALYZE or NIFTI format*. R package version 1.1-12, URL `http://CRAN.R-project.org/package=AnalyzeFMRI`.

Ogawa S, Lee TM, Kay AR, Tank DW (1990). "Brain magentic resonance imaging with contrast dependent on blood oxygenation." *Proceedings of the National Academy of Sciences of the United States of America*, **87**, 9868–9872.

Ogawa S, Tank DW, Menon R, Ellermann JM, Kim S, Merkle H, Ugurbil K (1992). "Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging." *Proceedings of the National Academy of Sciences of the United States of America*, **89**, 5951–5955.

Polzehl J, Spokoiny V (2006). "Propagation-separation approach for local likelihood estimation." *Probability Theory and Related Fields*, **135**, 335–362.

Polzehl J, Tabelow K (2007). "fmri: A package for analyzing fmri data." *R News*, **7**, 13–17.

Polzehl J, Voss HU, Tabelow K (2010). "Structural adaptive segmentation for statistical parametric mapping." *Neuroimage*, **52**, 515–523.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org`.

Resnick SI (1987). *Extreme Values, Regular Variation, and Point Processes*. Springer-Verlag.

Rorden C, Brett M (2000). "Stereotaxic display of brain lesions." *Behavioural Neurology*, **12**, 191–200.

Tabelow K, Piëch V, Polzehl J, Voss HU (2009). "High-resolution fMRI: Overcoming the signal-to-noise problem." *Journal of Neuroscience Methods*, **178**, 357–365.

Tabelow K, Polzehl J (2010a). *adimpro: Adaptive Smoothing of Digital Images*. R package version 0.7.5, URL `http://CRAN.R-project.org/package=adimpro`.

Tabelow K, Polzehl J (2010b). *fmri: Analysis of fMRI experiments*. R package version 1.4.2, URL `http://CRAN.R-project.org/package=fmri`.

Tabelow K, Polzehl J, Uluğ AM, Dyke JP, Watts R, Heier LA, Voss HU (2008). "Accurate Localization of Brain Activity in Presurgical fMRI by Structure Adaptive Smoothing." *IEEE Transactions on Medical Imaging*, **27**, 531–537.

Tabelow K, Polzehl J, Voss HU, Spokoiny V (2006). "Analyzing fMRI experiments with structural adaptive smoothing procedures." *Neuroimage*, **33**, 55–62.

Voss HU, Tabelow K, Polzehl J, Tchernichovski O, Maul KK, Salgado-Commissariat D, Ballon D, Helekar SA (2007). "Functional MRI of the zebra finch brain during song stimulation suggests a lateralized response topography." *Proceedings of the National Academy of Sciences of the United States of America*, **104**, 10667–10672.

Whitcher B (2010). *oro.dicom: Rigorous - DICOM Input / Output*. R package version 0.2.7, URL `http://CRAN.R-project.org/package=oro.dicom`.

Whitcher B, Schmid V, Thornton A (2010). *oro.nifti: Rigorous - NIfTI Input / Output*. R package version 0.2.1, URL `http://CRAN.R-project.org/package=oro.nifti`.

Worsley KJ (1994). "Local maxima and the expected Euler characteristic of excursion sets of χ^2, F and t fields." *Advances in Applied Probability*, **26**, 13–42.

Worsley KJ (2005). "Spatial smoothing of autocorrelations to control the degrees of freedom in fMRI analysis." *Neuroimage*, **26**, 635–641.

Worsley KJ, Liao C, Aston JAD, Petre V, Duncan GH, Morales F, Evans AC (2002). "A general statistical analysis for fMRI data." *Neuroimage*, **15**, 1–15.