A Quick Guide for the MixfMRI Package

Wei-Chen Chen[1] **and Ranjan Maitra**[2]

[1]pbdR Core Team
Silver Spring, MD, USA

[2]Department of Statistics
Iowa State University
Ames, IA, USA

Contents

Warning: The findings and conclusions in this article have not been formally disseminated by the U.S. Food and Drug Administration and should not be construed to represent any determination or policy of any University, Institution, Agency, Adminstration and National Laboratory.

Ranjan Maitra and this research were supported in part by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) of the National Institutes of Health (NIH) under its Award No. R21EB016212. The content of this paper however is solely the responsibility of the authors and does not represent the official views of either the NIBIB or the NIH.

This document is written to explain the main function of **MixfMRI** (Chen and Maitra 2018b), version 0.1-0. Every effort will be made to ensure future versions are consistent with these instructions, but features in later versions may not be explained in this document.

1. Introduction

The main purpose of this vignette is to demonstrate basic usage of **MixfMRI** which is prepared to implement developed methodology, simulation studies, data analyses in "Improved Activation Detection in Single-Subject fMRI Studies" (Chen and Maitra 2018a). The methodology mainly utilizes model-based clustering (unsupervised) of functional Magnetic Resonance (fMRI) data that identifies regions of brain activation associated with the performance of a task or the application of a stimulus. The implemented methods include 2D and 3D unsupervised segmentation analyses for fMRI signals. For simplification, only 2D clustering is demonstrated in this vignette. In this package, the data on fMRI signals are on the form of the p-values at each voxel of a Statistical Parametric Map (SPM).

The clustering and segmentation analyses identify activated voxels/signals (in terms of small p-values) from normal brain behaviors within a single subject but also use spatial context.

Note that the p-values may be derived from statistical models where typically a proper experiment design is required. These p-values are our data, and are used for analysis and no statements about significance level of p-values associated with activated voxels/signals are needed. Our analysis approach allows for the prespecification of *a priori* expected upper bounds for the proportion of activated voxels/signals that can guide in determining activated voxels.

For large datasets, the methods and analyses are also implemented in a distributed manner especially using SPMD programming framework. The package also includes workflows which utilize SPMD techniques. The workflows serve as examples of data analyses and large scale simulation studies. Several workflows are also built in to automatically process clusterings, hypotheses, merging clusters, and visualizations. See Section 1.5 and files in `MixfMRI/inst/workflow/` for more information.

1.1. Dependent Packages

The **MixfMRI** package depends on other R packages to be functional even though they are not always required. This is because some examples, functions and workflows of the **MixfMRI** may need utilities of those dependent packages. For instance,

Imports: MASS, **Matrix**, **RColorBrewer**, **fftw**, **MixSim**, **EMCluster**.
Enhances: pbdMPI, **AnalyzeFMRI**, **oro.nifti**.

1.2. The Main Function

The main function, `fclust()`, implements model-based clustering using the EM algorithm (McLachlan and Krishnan 1996) for fMRI signal data and provides unsupervised clustering results that identify activated regions in the brain. The `fclust()` function contains an initialization method and EM algorithms for clustering fMRI signal data which have two parts:

- `PV.gbd` for p-value of signals associated with voxels, and
- `X.gbd` for voxel information/locations in either 2D or 3D,

where `PV.gbd` is of length N (number of voxels) and `X.gbd` is of dimension $N \times 2$ or $N \times 3$ (for 2D or 3D). Each signal (per voxel) is assumed to follow a mixture distribution of `K` components with mixing proportion `ETA`. Each component has two independent coordinates (one for each part) with density functions: Beta and multivariate Normal distributions, for each part of fMRI signal data.

Beta Density:
The first component (`k = 1`) is restricted by `min.1st.prop` and $Beta(1, 1)$ (equivalently, the standard uniform) distribution. The rest `k = 2, 3, ..., K - 1` components have different $Beta(alpha, beta)$ distributions with `alpha < 1 < beta` for all `k > 1` components. This coordinate mainly represents the results of test statistics for determining activation of voxels (those that have smaller p-values). Note that the test statistics may be developed/smoothed/-computed from a time course model associated with voxel behaviors. See the main paper Chen and Maitra (2018a) for information.

Multivariate Normal Density:
The logarithm of the multivariate normal density is used as a penalty to provide regularization of the estimated parameters and in the estimated activation. `model.X = "I"` is for identity covariance matrix of this multivariate Normal distribution, and `"V"` for unstructured covariance matrix. `ignore.X = TRUE` is to ignore `X.gbd` and normal density, *i.e.* there is no regularization and only the Beta density is used. Note that this coordinate (for each axis) is recommended to be normalized in the $(0, 1)$ scale which is on the same scale of Beta density. From a modeling perspective, rescaling `X.gbd` does not have an effect.

In this package, the two parts `PV.gbd` and `X.gbd` are assumed to be independent because the latter comes through the addition of a penalty term to the log likelihood of the voxel-wise data on p-values. The goal of the main function is to provide spatial clusters (in addition to the `PV.gbd`) indicating spatial correlations.

Currently, APECMa (Chen and Maitra 2011) and EM algorithms are implemented with EGM algorithm (Chen *et al.* 2013) to speed up convergence when MPI and **pbdMPI** (Chen *et al.* 2012) are available. RndEM initialization (Maitra 2009) with a specific way of choosing initial seeds is implemented for obtaining good initial values that has the potential to increase the chances of convergence.

1.3. Datasets

The package has been built with several datasets including

- three 2D phantoms, `shepp0fMRI`, `shepp1fMRI`, and `shepp2fMRI`,
- one 3D dataset, `pstats`, with p-values obtained from the SPM obtained after running the Analysis of Functional Neuroimaging (AFNI) software (Cox 1996; Cox and Hyde 1997; Cox 2012) on the imagination dataset of Tabelow and Polzehl (2011).
- two small 2D voxels datasets, `pval.2d.complex` and `pval.2d.mag`, in p-values
- two toy examples, `toy1` and `toy2`.

1.4. Examples

The scripts in `MixfMRI/demo/` have several examples that demonstrate the main function, the example datasets and other utilities in this package. For a quick start,

- the scripts `MixfMRI/demo/fclust2d.r` and `MixfMRI/demo/fclust3d.r` show the basic usage of the main function `fclust()` using the two toy datasets,
- the scripts `MixfMRI/demo/maitra_2d.r` and `MixfMRI/demo/shepp.r` show and visualize examples on how to generate simulated datasets with given overlap levels, and
- the scripts `MixfMRI/demo/alter_*.r` show alternative methods.

1.5. Workflows

The package also has several workflows established for simulation studies. The main examples are located in `MixfMRI/inst/workflow/simulation/`. See the file `create_simu.txt` that generates scripts for simulations.

The files under `MixfMRI/inst/workflow/spmd/` have the main scripts for the workflows. Note that MPI and **pbdMPI** are required for workflows because these simulations require potentially long computing times.

2. Demonstration

The examples presented below are simulated and are not necessarily meant to represent meaningful activation study on the brain. Their purpose is to demonstrate our segmentation methodology in activation detection.

2.1. 2D Phantoms

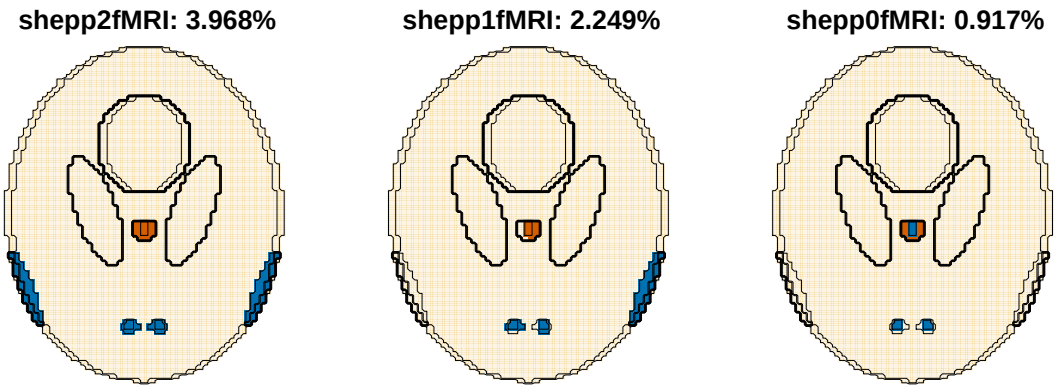
Three 2D phantoms built in the **MixfMRI** package can be displayed in R from the demo as simple as

Maitra's Phantoms

```
R> demo(maitra_phantom, package = "MixfMRI", ask = F)
```

which performs the code in `MixfMRI/demo/maitra_phantom.r`. The R command should give a plot similar to Figure 1 containing three different simulated 2D slices of a hypothesized brain. Each phantom may have different amounts of activated voxels (in terms of smaller p-values). Colors represent different activation intensities. The total proportions of truly active voxels are listed in the title of each phantom.

Figure 1: Simulated 2D Phantoms.



The examples used below mimic some active regions (in 2D) depending on different types of stimuli that may trigger responses in the brain. Hypothetically, the voxels may be active by regions, but each region may not be active in the same way (or magnitude) even though they may need to collectively respond to the stimuli (for example, due to time delay, response order, or sensitivity of study design).

As an example, only 3.968% of voxels in the `shepp2fMRI` phantom are active and indicated by two different colors (blue and brown) for different activation types where p-values may be smaller than 0.05 and may follow two Beta distributions (with different configurations) for the truly active voxels and one uniform distribution (i.e. $Beta(1, 1)$) for the truly inactive voxels.

The following code provides some counts for each groups of active and inactive voxels in the `shepp2fMRI` phantom.

Summary of shepp2fMRI Phantoms

```
> table(shepp2fMRI, useNA="always")
shepp2fMRI
    0     1     2  <NA>
13408   472    82 51574
```

The summary says that this phantom has three kinds of activations with group ids: 0, 1, and 2. There are 13,408 voxels belonging to cluster 0 (inactive), followed by 472 voxels belonging to cluster 1 (active & highlighted in blue in Figure 1), and 82 voxels belonging to cluster 2 (active & high lighted in brown in Figure 1). There are 51,574 pixels (NA) of this imaging dataset which are not within the brain (contour by the black line in Figure 1). See Section 2.2 for information of generating p-values from a mixture of three Beta distributions.

2.2. 2D Simulations

The **MixfMRI** provides a function `gendataset(phantom, overlap)` to generate p-values of activations. The function needs two arguments: `phantom` and `overlap`. The `phantom` is a map containing voxel group id's where p-values will be simulated from a mixture Beta distribution with certain mixture level specified by the `overlap` argument. The example can be found in `MixfMRI/demo/maitra_2d.r` and can be done in R as simple as

Simulations of Active Voxels

```
R> demo(maitra_2d, package = "MixfMRI", ask = F)
```

Note that the overlap represents similarity of activation signals. The higher the overlap, the more difficult it becomes to distinguish between activation and inactivation and also the kinds of activation.

The command above should give a plot similar to Figure 2 containing group id's on the left and their associated p-values for stimulus responses on the right. The top row displays examples for phantom `shepp1fMRI`, and the bottom row displays examples for phantom `shepp2fMRI`.

- Inside the brain, the group id 2's are indicated by white (active is highly associated with stimula due to experiment design), 1's are indicated by light gray (slightly active), and 0's are indicated by dark gray (inactive). Note that the region with white color was the region colored blue and the light gray region corresponds to the region colored brown in Figure 1
- The simulated p-values are colored by a map using a red-orange-yellow palette from 0 to 1. Note that small p-values (redder voxels) may also occur at truly inactive voxels.

See Figure 3 for the distribution of simulated p-values for the phantom `shepp2fMRI`.

The methodology and analyses implemented in this package aim to identify those active voxels in spatial clusters. For example, regions of active voxels associated with imagining the playing of certain sports. When an experiment was conducted/designed to detect brain behaviors, the statistical model and the p-values of the treatment effect should be able to reflect the voxel activations. Typically, the statistical tests are done independently voxel-by-voxel due to complexity of computation and modeling. This package provides post hoc clustering that adds spatial contents to p-values and helps to isolate meaningful regions clouded with many small p-values. See Chen and Maitra (2018a) for information of clustering performance and comprehensive assessments for this post hoc approach.

2.3. 2D Clustering

The example can be found in `MixfMRI/demo/maitra_2d_fclust.r` as simple as

Clustering of Active Voxels

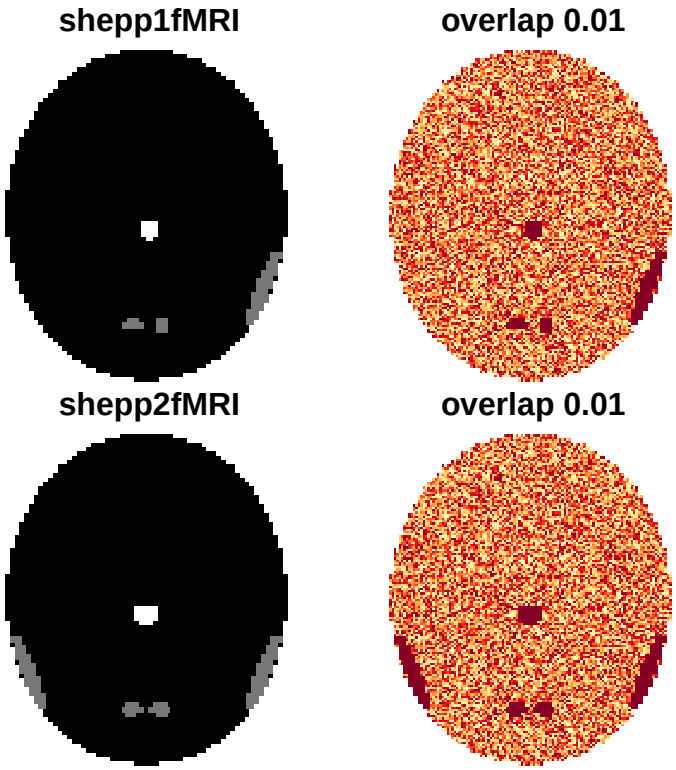
```
R> demo(maitra_2d_fclust, package = "MixfMRI", ask = F)
```

This demo (explained below) is to cluster the simulated p-values (see Section 2.2) using the developed method.

Code of `maitra_2d_fclust.r`

```
library(MixfMRI, quietly = TRUE)
```

Figure 2: Activated regions of voxels and simulated p-values.



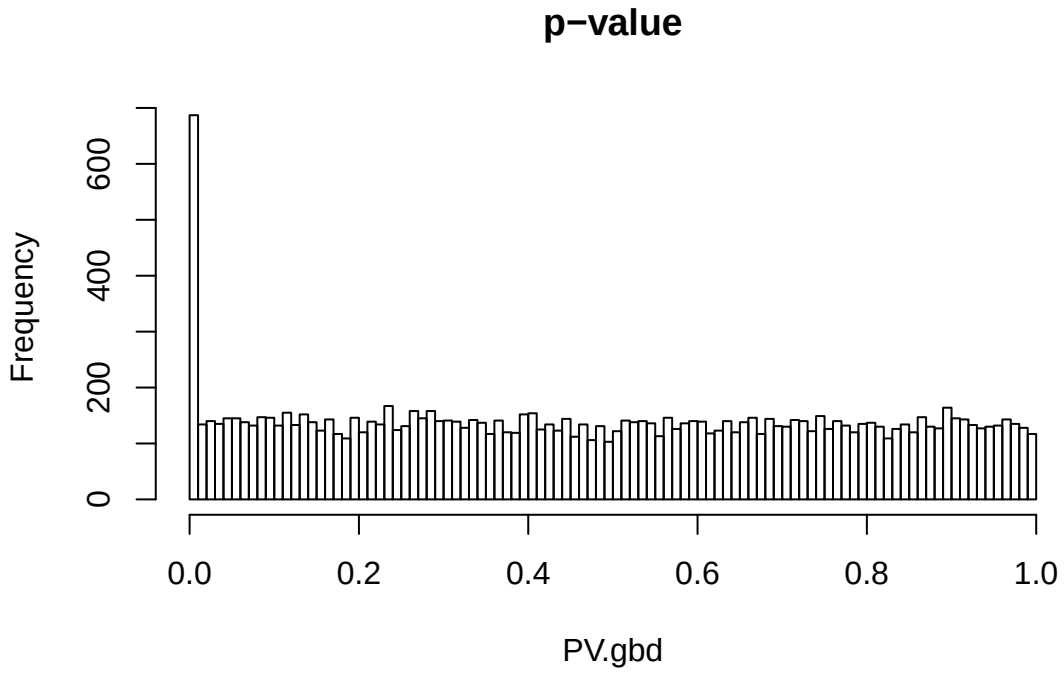

```
set.seed(1234)
da <- gendataset(phantom = shepp2fMRI, overlap = 0.01)$pval

### Check 2d data.
id <- !is.na(da)
PV.gbd <- da[id]
# pdf(file = "maitra_2d_fclust.pdf", width = 6, height = 4)
hist(PV.gbd, nclass = 100, main = "p-value")
# dev.off()

### Test 2d data.
id.loc <- which(id, arr.ind = TRUE)
X.gbd <- t(t(id.loc) / dim(da))
ret <- fclust(X.gbd, PV.gbd, K = 3)
print(ret)

### Check performance
library(EMCluster, quietly = TRUE)
RRand(ret$class, shepp2fMRI[id] + 1)
```

In the code above, the histogram of simulated p-values is plotted in Figure 3. Then, the `fclust(X.gbd, PV.gbd, K = 3)` groups voxels in three clusters. At the end, `ret` saves the clustering results. The `print(ret)` from the above code will show the results below in detail:

- `N` is the total number of voxels to be clustered/segmented

- `K` is the total number of segments.
- `n.class` is the number of voxels in each segment
- `ETA` is the mixing proportion of each segment
- `BETA` is the set of parameters of the Beta distributions (by column)
- `MU` is the centers of the segments (the spatial location inside the brain)
- `SIGMA` is the dispersion of the segments

Figure 3: Activation (p-values) Distribution. The x-axis is for the p-values.



The numbers of voxels for each segment, in this example, are 13,394, 184, and 384 associated with new cluster ids: 0, 1, and 2, respectively. Comparing with the true classifications (see the table in Section 2.1), the adjusted Rand index gives 0.9749 indicating good agreement between the truly active and the activated (as determined by our segmentation methodology) results.

Outputs of Clustering

```
R> print(ret)
Algorithm: apecma  Model.X: I  Ignore.X: FALSE
- Convergence: 1  iter: 16  abs.err: 0.02091979  rel.err: 7.375343e-07
- N: 13962  p.X: 2  K: 3  logL: 28364.52
- AIC: -56693.04  BIC: -56557.25 ICL-BIC: -55712.18
- n.class: 13394 184 384
```

```
- init.class.method:
- ETA: (min.1st.prop: 0.8  max.PV: 0.1)
[1] 0.95266704 0.01307847 0.03425449
- BETA: (2 by K)
     [,1]          [,2]        [,3]
[1,]    1 1.127244e-01 0.04237128
[2,]    1 4.429518e+04 1.00000130
- MU: (p.X by K)
          [,1]      [,2]      [,3]
[1,] 0.5013538 0.3105460 0.5917377
[2,] 0.5076080 0.3718145 0.3749842
- SIGMA: (d.normal by K)
            [,1]          [,2]         [,3]
[1,] 0.01271198 0.0001859628 0.009462210
[2,] 0.02186662 0.0011582052 0.004284016

R> RRand(ret$class, shepp2fMRI[id] + 1)
   Rand adjRand  Eindex
 0.9964  0.9749  1.7012
```

References

Chen WC, Maitra R (2011). "Model-based clustering of regression time series data via APECM – an AECM algorithm sung to an even faster beat." *Statistical Analysis and Data Mining*, **4**, 567–578.

Chen WC, Maitra R (2018a). "Improved Activation Detection in Single-Subject fMRI Studies." *manuscript*.

Chen WC, Maitra R (2018b). "MixfMRI: fMRI Clustering Analysis." R Package, URL http://cran.r-project.org/package=MixfMRI.

Chen WC, Ostrouchov G, Pugmire D, Prabhat M, Wehner M (2013). "A Parallel EM Algorithm for Model-Based Clustering with Application to Explore Large Spatio-Temporal Data." *Technometrics*, **55**, 513–523.

Chen WC, Ostrouchov G, Schmidt D, Patel P, Yu H (2012). "pbdMPI: Programming with Big Data – Interface to MPI." R Package, URL https://cran.r-project.org/package=pbdMPI.

Cox RW (1996). "AFNI: software for analysis and visualization of functional magnetic resonance neuroimages." *Computers and biomedical research, an international journal*, **29**(3), 162–173.

Cox RW (2012). "AFNI: What a long strange trip it has been." *NeuroImage*, **62**, 743–747.

Cox RW, Hyde JS (1997). "Software tools for analysis and visualization of fMRI data." *NMR in Biomedicine*, **10**(4-5), 171–178.

Maitra R (2009). "Initializing Partition-Optimization Algorithms." *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **6**, 144–157.

McLachlan G, Krishnan T (1996). *The EM Algorithm and Extensions*. John Wiley & Sons.

Tabelow K, Polzehl J (2011). "Statistical Parametric Maps for Functional MRI Experiments in R: The Package fmri." *Journal of Statistical Software*, **44**(11), 1–21. ISSN 1548-7660. URL http://www.jstatsoft.org/v44/i11.