# Package ‘rosenblatt1’

August 20, 2012

**Type** Package

**Title** Mixture random effects in group fMRI studies

**Version** 0.5

**Date** 2012-08-19

**Author** Jonathan Rosenblatt <john.ros@gmail.com>

**Maintainer** Jonathan Rosenblatt <john.ros@gmail.com>

**Description**
Takes coeffient maps of fMRI group studies and returns maps of the estimates of the random-effect mixture distribution.

**License** GPL-2

**Collate** 'functions.R'

**Depends** mixtools, tractor.base

## R topics documented:

---

`rosenblatt1-package` *Fit mixture models to fMRI group studies*

---

### Description

Fit mixture models to fMRI group studies.

### Details

| | |
|---|---|
| Package: | rosenblatt1 |
| Type: | Package |
| Version: | 0.5 |
| Date: | 2012-08-19 |
| License: | GPL (>= 2) |

Mixture model fitting for random effect in fMRI group studies.

### Author(s)

Jonathan Rosenblatt

---

`brainMixtureFit` *Fits a mixture models to a group SPMbeta*

---

### Description

Fits a mixture models to a group SPMbeta

### Usage

```
brainMixtureFit(MRImage.list,
  fit.control = generateMixtureControl(), ...)
```

### Arguments

`MRImage.list` List of Mriimage objects. All assumed to have same resolution and representing different subjects.

`fit.control` List of control parameters generated by `generateMixtureControl`

`...` Deprecated

## Value

List of MriImage objects. Each holding the SPM of a different parameters:

| | |
|---|---|
| `initial.p1` | Deprecated |
| `initial.p2` | Deprecated |
| `initial.p3` | Deprecated |
| `initial.mu` | Deprecated |
| `initial.A` | Deprecated |
| `initial.B` | Deprecated |
| `initial.C` | Deprecated |
| `p1.1` | Estimate of *full* mixture parameters. |
| `p2.1` | Estimate of *full* mixture parameters. |
| `p3.1` | Estimate of *full* mixture parameters. |
| `mu.1` | Estimate of *full* mixture parameters. |
| `A.1` | Estimate of *full* mixture parameters. |
| `B.1` | Estimate of *full* mixture parameters. |
| `C.1` | Estimate of *full* mixture parameters. |
| `likelihood.1` | *Full* maximal likelihood. |
| `p3.bound` | Maximal allowed value for p.3 given other parameter estimates. Serves to check if parameter space constraints are violated. |
| `p1.0` | Estimate of *null* mixture parameters. |
| `p2.0` | Estimate of *null* mixture parameters. |
| `A.0` | Estimate of *null* mixture parameters. |
| `B.0` | Estimate of *null* mixture parameters. |
| `likelihood.0` | *Null* maximal likelihood |
| `n` | Sample size. |

Given a list of MriImage objects– all registered to a common template– the function will fit a mixture model at each voxel and return an array of the voxel-wise parameter estimates. The fitting is done using the normalmixEM package. The mixture being fitted is parametrized as:

$$p_1\phi_{0,\sigma_1} + p_2\phi_{0,\sigma_2} + p_3\phi_{\mu,\sigma_3}$$

where $\phi_{\mu,\sigma}$ is the Gaussian PDF.

## Author(s)

Jonathan Rosenblatt

## References

Revisiting Multi-Subject Random Effects in fMRI Rosenblatt J.D., Vink M., Benjamini Y. (2012)

### Examples

```
data(VinkData)
## Not run:
### No need to run. Output already in VinkDataFit in order to save time.
test.brain.fit<- brainMixtureFit(scans, fit.control = generateMixtureControl())

## End(Not run)
data(VinkDataFit)
createSliceGraphic(test.brain.fit[[”p3.1”]], z=26)
image(test.brain.fit[[”p3.1”]]$getData()[,,26])
lapply(test.brain.fit, function(x) x$getData()[20,20,20])
```

## computeMask *Computes a rejection group mask given beta maps.*

### Description

Computes a rejection group mask given beta maps.

### Usage

```
computeMask(MRImage.list, test.statistic = ”Wilcoxon”, FDR.level = 0.1, fit.control=generateMixtureC
```

### Arguments

`MRImage.list` List of MriImage objects. All assumed to have same resolution and representing different subjects.

`test.statistic` Which group test statistic to use. At present "T" or "Wilcoxon" (default) are supported.

`FDR.level` Level of FDR multiplicity control.

`fit.control` List of control parameters generated by `generateMixtureControl`

### Value

An MriImage class object corresponding to the binary mask of rejected locations.

### Author(s)

Jonathan Rosenblatt

### Examples

```
data(VinkData)
test.brain.mask<- computeMask(MRImage.list=scans, test.statistic = ”T”, fit.control=generateMixtureControl())
createSliceGraphic(test.brain.mask, z=30)
```

## exportFitAsMedicalImage

*Export all estimated parameters to imaging files*

### Description

Export all estimated parameters to imaging files

### Usage

```
exportFitAsMedicalImage(mixture.fit.object, format,
  file.heading)
```

### Arguments

`mixture.fit.object`
Output of `brainMixtureFit`

`format` The medical imaging format to export. See details.

`file.heading` Heading of output files.

### Value

Nothing. Called for it’s side effects.

This is a convenience wrapper around `writeMriImageToFile`.

It exports all the SPM outputed by `brainMixtureFit` and saves them in any of the formats supported by `writeMriImageToFile`.

### Author(s)

Jonathan Rosenblatt

### Examples

```
## Not run:
exportFitAsMedicalImage(mixture.fit.object = test.brain.fit, file.heading = "export", format = "test")

## End(Not run)
```

## generateMixtureControl *Generates a list of control arguments.*

### Description

Generates a list of control arguments.

### Usage

```
generateMixtureControl(numericThresh = -18,   minObservations = 15)
```

### Arguments

| | |
|---|---|
| numericThresh | Numeric value. |
| minObservations | Minimal number of observation to attempt fitting. |

### Value

List with control parameters for brainMixtureFit

### Author(s)

Jonathan Rosenblatt

## importBetaMriImages *Function imoprting a directory with scans in all formats except DICOM:*

### Description

Function imoprting a directory with scans in all formats except DICOM:

### Usage

```
importBetaMriImages(files, ...)
```

### Arguments

| | |
|---|---|
| files | A list of file names with full path. |
| ... | Arguments passed to newMriImageFromFile |

## Value

A list containin an MriImage class object for each file in the input.

The function is a convinience wrapper around newMriImageFromFile. It recieves a list of medical imaging files (in the formats supported by newMriImageFromFile) and return a list of MriImage objects– one for each file supplied.

## Author(s)

Jonathan Rosenblatt

## Examples

```
## Not run:
files<- grep("con.*img", list.files(), value=T)
files<- sub('\\.img', "", files)
scans<- importBetaMriImages(files, fileType='NIFTI')
class(scans[[1]])

## End(Not run)
```

---

# imputeArray *Utility function*

---

## Description

Impute an array with first degree neighbors. Typically called by brainMixtureFit.

## Usage

```
imputeArray(beta.array, min.neighbours = 20)
```

## Arguments

| | |
|---|---|
| beta.array | A four dimensional array of coefficients. |
| min.neighbours | Minimum number of valid neighboring observations for the imputation to occur |

## Value

An array with imputed values.

The function will look for location with more than 20 neighbours and impute the missing value with their 0.1 trimmed mean.

## Author(s)

Jonathan Rosenblatt

## `mixedtools2result` *Map mixtools output to simple vector*

### Description

Map mixtools output to simple vector

### Usage

```
mixedtools2result(mixedtools.output, result, model)
```

### Arguments

`mixedtools.output`
Output of `normalmixEM`. List of class *mixEM*.

`result` The vector to be returned with `normalmixEM` output

`model` Character vector specifying the type of model fitted with `normalmixEM`.

### Value

Vector of voxel-wise parameter estimates. Utility function for `pointWiseMixtureFit`

### Author(s)

Jonathan Rosenblatt

## `MriImage2Array` *Converts a list of MRIimage objects to a single four dimentional array.*

### Description

Converts a list of MRIimage objects to a single four dimentional array.

### Usage

```
MriImage2Array(MRImage.list)
```

### Arguments

`MRImage.list` List of Mriimage objects. All assumed to have same resolutino and representing different subjects.

### Value

A numeric four dimentional array with space as first three dimentions and subject as the fourth.

## Author(s)

Jonathan Rosenblatt

## Examples

```
data(VinkData)
MriImage2Array(scans)
```

---

# neighbourhoodMatrix *Utility function*

---

## Description

Generate matrix of neighboring indexes for brain image imputation.

## Usage

```
neighbourhoodMatrix(i, j, k)
```

## Arguments

| | |
|---|---|
| i | Location in brain array. |
| j | Location in brain array. |
| k | Location in brain array. |

## Value

Matrix of neighboring voxel indexes.

## Author(s)

Jonathan Rosenblatt

---

# p3Bound *Checks if constraint on p3 is met?*

---

## Description

Checks if constraint on p3 is met?

## Usage

```
p3Bound(p1, mu, A, B, n, fit.control)
```

### Arguments

`p1` Value of mixture parameter.

`mu` Value of mixture parameter.

`A` Value of mixture parameter.

`B` Value of mixture parameter.

`n` Value of mixture parameter.

`fit.control` List of control arguments generated by `generateMixtureControl`

### Value

The maximal allowed value for $p_3$.

### Author(s)

Jonathan Rosenblatt

---

## `pointWiseMixtureFit` *Pointwise fitting of group mixture*

---

### Description

`pointWiseMixtureFit` returns the point wise fitting of group mixture.
`pointWise3MixtureFit` returns a `mixEM` class object of the *three* component mixture fit.
`pointWise3MixtureFit` returns a `mixEM` class object of the *two* component mixture fit.

### Usage

```
pointWiseMixtureFit(beta.vector, fit.control, progress)

  pointWise3MixtureFit(beta.vector, fit.control)

      pointWise2MixtureFit(beta.vector, fit.control,   three.component.fit)
```

### Arguments

`beta.vector` Numeric vector of estimated contrasts. One per subject.

`fit.control` List of control parameters generated by `generateMixtureControl`

`progress` A connection object to a progress bar.

`three.component.fit`
An initial fit done by `pointWise3MixtureFit`

### Value

Numeric vector with estimates of mixture parameters.

### Author(s)

Jonathan Rosenblatt

## `scans` *Raw coefficient maps for group-level inference*

### Description

List of MriImage objects imported using `importBetaMriImages`. Each object originates from a different subject. All data has been pre-processed as described in the references.

### Usage

```
scans
```

### Format

List

### References

Zandbelt, Bram B, Mariet van Buuren, Rene S Kahn, and Matthijs Vink. "Reduced Proactive Inhibition in Schizophrenia Is Related to Corticostriatal Dysfunction and Poor Working Memory." Biological Psychiatry 70, no. 12 (December 15, 2011): 1151–1158.
van Buuren, Mariet, Matthijs Vink, Anca E. Rapcencu, and Rene S. Kahn. "Exaggerated Brain Activation During Emotion Processing in Unaffected Siblings of Patients with Schizophrenia." Biological Psychiatry 70, no. 1 (July 1, 2011): 81–87.

## `test.brain.fit` *Mixture estimate maps for group-level inference.*

### Description

List of MriImage objects created using `brainMixtureFit`. Each object contains an estimate map of one of the parameters of the group-level mixture.

### Usage

```
test.brain.fit
```

### Format

List

## References

Zandbelt, Bram B, Mariet van Buuren, Rene S Kahn, and Matthijs Vink. "Reduced Proactive Inhibition in Schizophrenia Is Related to Corticostriatal Dysfunction and Poor Working Memory." Biological Psychiatry 70, no. 12 (December 15, 2011): 1151–1158.
van Buuren, Mariet, Matthijs Vink, Anca E. Rapcencu, and Rene S. Kahn. "Exaggerated Brain Activation During Emotion Processing in Unaffected Siblings of Patients with Schizophrenia." Biological Psychiatry 70, no. 1 (July 1, 2011): 81–87.

---

# `wrapImputeArray` *Utility function*

---

## Description

Impute a selected parameters in a brain array. To be called by `brainMixtureFit`.

## Usage

```
wrapImputeArray(brain.mixture.fit.object)
```

## Arguments

`brain.mixture.fit.object`
A four dimensional array of mixture parameter estimates.

## Value

An array of parameter estimates imputed using first degree neighborhood.

## Author(s)

Jonathan Rosenblatt

# Index