

# Package ‘CASMAP’

May 5, 2020

**Type** Package

**Title** Detection of Statistically Significant Combinations of SNPs in Association Mapping

**Version** 0.6.1

**Description** A significant pattern mining-based toolbox for region-based genome-wide association studies and higher-order epistasis analyses, implementing the methods described in Llinares-López et al. (2017) <doi:10.1093/bioinformatics/btx071>.

**Depends** R (>= 3.0.2)

**Imports** methods, Rcpp

**LinkingTo** Rcpp

**Encoding** UTF-8

**LazyData** true

**License** GPL (>= 2)

**NeedsCompilation** yes

**RoxygenNote** 6.0.1

**SystemRequirements** C++11

**Suggests** testthat, knitr, rmarkdown

**Author** Felipe Llinares-López [aut, cph],  
Laetitia Papaxanthos [aut, cph],  
Damian Roqueiro [aut, cph],  
Matthew Baker [ctr],  
Mikołaj Rybiński [ctr],  
Uwe Schmitt [ctr],  
Dean Bodenham [aut, cre, cph],  
Karsten Borgwardt [aut, fnd, cph]

**Maintainer** Dean Bodenham <deanbodenhamsse@gmail.com>

**VignetteBuilder** knitr

**Repository** CRAN

**Date/Publication** 2020-05-05 18:10:02 UTC

## R topics documented:

CASMAP-class . . . . .	2
getExampleCovariatesFilename . . . . .	4
getExampleDataFilename . . . . .	5
getExampleLabelsFilename . . . . .	5
<b>Index</b>	<b>7</b>

---

CASMAP-class	<i>Constructor for CASMAP class object.</i>
--------------	---

---

### Description

Constructor for CASMAP class object.

### Details

Constructor for CASMAP class object, which needs the mode parameter to be set by the user. Please see the examples.

### Fields

mode Either 'regionGWAS' or 'higherOrderEpistasis'.

alpha A numeric value setting the Family-wise Error Rate (FWER). Must be strictly between 0 and 1. Default value is 0.05.

max\_comb\_size A numeric specifying the maximum length of combinations. For example, if set to 4, then only combinations of size between 1 and 4 (inclusive) will be considered. To consider combinations of arbitrary (maximal) length, use value 0, which is the default value.

### Base method, for both modes

readFiles Read the data, label and possibly covariates files. Parameters are genotype\_file, for the data, phenotype\_file for the labels and (optional) covariates\_file for the covariates. The option plink\_file\_root is not supported in the current version, but will be supported in future versions.

setMode Can set/change the mode, but note that any data files will need to read in again using the readFiles command.

setTargetFWER Can set/change the Family-wise Error Rate (FWER). Takes a numeric parameter alpha, strictly between 0 and 1.

execute Once the data files have been read, can execute the algorithm. Please note that, depending on the size of the data files, this could take a long time.

getSummary Returns a data frame with a summary of the results from the execution, but not any significant regions/itemsets. See getSignificantRegions, getSignificantInteractions, and getSignificantClusterRepresentatives.

writeSummary Directly write the information from getSummary to file.

**regionGWAS Methods**

- `getSignificantRegions` Returns a data frame with the the significant regions. Only valid when `mode='regionGWAS'`.
- `getSignificantClusterRepresentatives` Returns a data frame with the the representatives of the significant clusters. This will be a subset of the regions returned from `getSignificantRegions`. Only valid when `mode='regionGWAS'`.
- `writeSignificantRegions` Writes the data from `getSignificantRegions` to file, which must be specified in the parameter path. Only valid when `mode='regionGWAS'`.
- `writeSignificantClusterRepresentatives` Writes the data from `getSignificantClusterRepresentatives` to file, which must be specified in the parameter path. Only valid when `mode='regionGWAS'`.

**higherOrderEpistasis Methods**

- `getSignificantInteractions` Returns the frame from `getSignificantInteractions` to file, which must be specified in the parameter path. Only valid when `mode='higherOrderEpistasis'`.
- `writeSignificantInteractions` Writes a data frame with the significant interactions. Only valid when `mode='higherOrderEpistasis'`.

**References**

- A. Terada, M. Okada-Hatakeyama, K. Tsuda and J. Sese *Statistical significance of combinatorial regulations*, Proceedings of the National Academy of Sciences (2013) 110 (32): 12996-13001
- F. Llinares-Lopez, D. G. Grimm, D. Bodenham, U. Gieraths, M. Sugiyama, B. Rowan and K. Borgwardt, *Genome-wide detection of intervals of genetic heterogeneity associated with complex traits*, ISMB 2015, Bioinformatics (2015) 31 (12): i240-i249
- L. Papaxanthos, F. Llinares-Lopez, D. Bodenham, K. Borgwardt, *Finding significant combinations of features in the presence of categorical covariates*, Advances in Neural Information Processing Systems 29 (NIPS 2016), 2271-2279.
- F. Llinares-Lopez, L. Papaxanthos, D. Bodenham, D. Roqueiro and K. Borgwardt, *Genome-wide genetic heterogeneity discovery with categorical covariates*. Bioinformatics 2017, 33 (12): 1820-1828.

**Examples**

```
## An example using the "regionGWAS" mode
fastcmh <- CASMAP(mode="regionGWAS")      # initialise object

datafile <- getExampleDataFilename()      # file name of example data
labelsfile <- getExampleLabelsFilename()  # file name of example labels
covfile <- getExampleCovariatesFilename() # file name of example covariates

# read the data, labels and covariate files
fastcmh$readFiles(genotype_file=getExampleDataFilename(),
                  phenotype_file=getExampleLabelsFilename(),
                  covariate_file=getExampleCovariatesFilename() )

# execute the algorithm (this may take some time)
```

```
fastcmh$execute()

#get the summary results
summary_results <- fastcmh$getSummary()

#get the significant regions
sig_regions <- fastcmh$getSignificantRegions()

#get the clustered representatives for the significant regions
sig_cluster_rep <- fastcmh$getSignificantClusterRepresentatives()

## Another example of regionGWAS
fais <- CASMAP(mode="regionGWAS")      # initialise object

# read the data and labels, but no covariates
fastcmh$readFiles(genotype_file=getExampleDataFilename(),
                  phenotype_file=getExampleLabelsFilename())

## Another example, doing higher order epistasis search
facs <- CASMAP(mode="higherOrderEpistasis")      # initialise object
```

---

`getExampleCovariatesFilename`

*Get the path to the example covariates file for regionGWAS mode*

---

### **Description**

Path to `CASMAP_example_covariates_1.txt` in `inst/extdata`. The covariates categories for the data set `CASMAP_example_data_1.txt`, the path to which is given by `getExampleDataFilename`.

### **Usage**

```
getExampleCovariatesFilename()
```

### **Format**

A single column vector of 100 labels, each of which is 0 or 1 (same format as labels file).

### **Details**

Path to the file containing the labels, for reading in to CASMAP object using the `readFiles` function.

### **See Also**

`getExampleDataFilename`, `getExampleLabelsFilename`

**Examples**

```
covfile <- getExampleCovariatesFilename()
```

---

`getExampleDataFilename`

*Get the path to the example data file for regionGWAS mode*

---

**Description**

Path to `CASMAP_example_data_1.txt` in `inst/extdata`. A dataset containing binary samples for the `regionGWAS` method. There are accompanying labels and covariates dataset.

**Usage**

```
getExampleDataFilename()
```

**Format**

A matrix of 0s and 1s, with 1000 rows (features) and 100 columns (samples). In other words, each column is a sample, and each sample has 1000 binary features.

**Details**

Path to the file containing the data, for reading in to `CASMAP` object using the `readFiles` function. Note that the significant region is `[99, 102]`.

**See Also**

```
getExampleLabelsFilename, getExampleCovariatesFilename
```

**Examples**

```
datafile <- getExampleDataFilename()
```

---

`getExampleLabelsFilename`

*Get the path to the example labels file for regionGWAS mode*

---

**Description**

Path to `CASMAP_example_labels_1.txt` in `inst/extdata`. A dataset containing the binary labels for the data in the file `CASMAP_example_data_1.txt`, the path to which is given by `getExampleDataFilename`.

**Usage**

```
getExampleLabelsFilename()
```

**Format**

A single column of 100 labels, each of which is either 0 or 1.

**Details**

Path to the file containing the labels, for reading in to CASMAP object using the `readFiles` function.

**See Also**

`getExampleDataFilename`, `getExampleCovariatesFilename`

**Examples**

```
labelsfile <- getExampleLabelsFilename()
```

# Index

CASMAP (CASMAP-class), [2](#)  
CASMAP-class, [2](#)

getExampleCovariatesFilename, [4](#)  
getExampleDataFilename, [5](#)  
getExampleLabelsFilename, [5](#)