

Package ‘MIMOSA’

October 10, 2021

Type Package

Title Mixture Models for Single-Cell Assays

Version 1.30.0

Date 2014-10-01

Author Greg Finak <gfinak@fhcrc.org>

Maintainer Greg Finak <gfinak@fhcrc.org>

Description Modeling count data using Dirichlet-multinomial and beta-binomial mixtures with applications to single-cell assays.

License MIT + file LICENSE

VignetteBuilder knitr

Imports methods, Formula, data.table, pracma, MCMCpack, coda, modeest, testthat, Rcpp, scales, dplyr, tidyr, rlang

Suggests parallel, knitr

Depends R (>= 3.0.2), MASS, plyr, reshape, Biobase, ggplot2

LinkingTo Rcpp, RcppArmadillo

LazyLoad yes

LazyData yes

RoxygenNote 7.1.1

Encoding UTF-8

biocViews ImmunoOncology, FlowCytometry, CellBasedAssays

git_url <https://git.bioconductor.org/packages/MIMOSA>

git_branch RELEASE_3_13

git_last_commit 8910727

git_last_commit_date 2021-05-19

Date/Publication 2021-10-10

R topics documented:

MIMOSA-package	2
.fitMCMC	3
asinh_trans	4
BetaMixResult-class	4
boxplotMIMOSAResultList	5
combine.MIMOSA	6
ConstructMIMOSAExpressionSet	6
countsTable	8
fdr	9
getPosteriorResponseRate	10
getZ	11
ICS	12
MDMix	13
MIMOSA	13
MIMOSAExpressionSet	14
MIMOSAResult	15
MIMOSAResult-class	15
pData<-,MCMCResult,ANY-method	16
print.MIMOSAResultList	17
show	17
volcanoPlot	18
Index	19

MIMOSA-package	<i>MIMOSA: Mixture Models for Single Cell Assays</i>
----------------	--

Description

MIMOSA implements mixtures of Dirichlet-multinomial or Beta-binomial models for paired count data from single-cell assays that typically arise in immunological studies. It can be used for ICS (Intracellular Cytokine Staining) assays to detect vaccine responders, for example, or to detect changes in proportions of cells expressing a gene, such as in Fluidigm Biomark Single-cell gene expression.

References

Greg Finak, Andrew McDavid, Pratip Chattopadhyay, Maria Dominguez, Stephen C De Rosa, Mario Roederer, Raphael Gottardo Mixture Models for Single Cell Assays with Applications to Vaccine Studies Biostatistics, 2013, <http://biostatistics.oxfordjournals.org/content/early/2013/07/24/biostatistics.kxt024.abstract>

See Also

[MIMOSA](#), [ConstructMIMOSAExpressionSet](#)

.fitMCMC*Fit the MIMOSA model via MCMC*

Description

This is an internal function that fits the MIMOSA model via MCMC. It is called from MIMOSA

Usage

```
.fitMCMC(
  data,
  inits = NULL,
  iter = 250000,
  burn = 50000,
  thin = 1,
  tune = 100,
  outfile = basename(tempfile(tmpdir = ".", fileext = ".dat")),
  alternative = "greater",
  UPPER = 0.5,
  LOWER = 0.15,
  FAST = TRUE,
  EXPRATE = 1e-04,
  pXi = c(1, 1),
  seed = 10
)
```

Arguments

data	a list with elements names 'n.stim' and 'n.unstim', the stimulated and unstimulated counts. Must be at least of dimension 2.
inits	the initialization parameters for the MCMC routine. Can be initialized from MDMix with <code>initonly=TRUE</code> .
iter	the number of Mote Carlo iterations
burn	the number of burn-in iterations
thin	The thinning interval
tune	the number of iterations used for tuning the step size
outfile	the output file name
alternative	either 'greater' or 'not equal' for fitting the one-sided or two-sided MIMOSA model, respectively.
UPPER	tuning parameter for the upper bound on the acceptance ratio of each paramter
LOWER	tuning parmeter for the lower bound on the acceptance ratio of each paramter
FAST	TRUE,FALSE. Use the heuristic (FAST=TRUE) for fitting a one-sided model rather than recomputing the normalization constant via MCMC for each step. @importFrom coda mcmc

EXPRATE	the mean of the prior distribution for the model hyperparameters.
pXi	is the prior on the w, beta(1,1) by default).
seed	numeric random seed @rdname fitMCMC @name .fitMCMC @importFrom data.table fread

asinh_trans	<i>asinh_trans</i>
-------------	--------------------

Description

Arcsinh transform for ggplot2

Usage

```
asinh_trans(c)
```

Arguments

c numeric cofactor for asinh tranform. Default 1.

Details

Arcsinh transform for use with coord_trans in ggplot2

Value

transform

Author(s)

Greg Finak

BetaMixResult-class	<i>The output of fitting Beta-Binomial EM implementation BetaMix.</i>
---------------------	---

Description

BetaMix will return an object of this class.

```
boxplotMIMOSAResultList  
  boxplotMIMOSAResultList
```

Description

Boxplots of MIMOSA

Usage

```
boxplotMIMOSAResultList(  
  data,  
  title = "A Boxplot",  
  x_axis_category = NULL,  
  cofactor = 5000,  
  line = TRUE,  
  threshold = 0.005  
)
```

Arguments

data	MIMOSAResultList
title	character Title of the plot.
x_axis_category	name the column of the phenoData frame for the x-axis of the boxplots.
cofactor	integer cofactor of the arcsinhTransform for the y axis.
line	logical whether or not to connect points from the same subject
threshold	numeric the FDR threshold (q-value) at which to classify responders as a separate category.

Details

Generate boxplots for MIMOSA positivity calls.

Value

ggplot object.

Author(s)

Greg Finak

combine.MIMOSA *Combine MIMOSAResultList objects*

Description

Combine two or more MIMOSAResultList objects

Usage

```
combine.MIMOSA(x, y, ...)
```

Arguments

x	MIMOSAResultList
y	MIMOSAResultList
...	additional MIMOSAResultList objects

Details

Combines two or more MIMOSAResultList objects. The method is light on error checking so the results should be from the same MIMOSAExpressionSet object.

Value

a MIMOSAResultList

Author(s)

Greg Finak

ConstructMIMOSAExpressionSet
A wrapper for constructing an Expression Set for MIMOSA

Description

Calls a series of other functions that will reshape and refactor the data frame into the right format for use by MIMOSA Standardized for use with internal SCHARP data sets. We provide some default arguments as examples. Currently slow, and very much prototype code.

Usage

```
ConstructMIMOSAExpressionSet(
  thisdata,
  reference = quote(STAGE %in% "CTRL" & PROTEIN %in% "Media+cells"),
  measure.columns = c("Neg", "Pos"),
  other.annotations = setdiff(colnames(thisdata), measure.columns),
  default.cast.formula = component ~ ...,
  .variables = quote(.(PTID, TESTDT, ASSAYID, PLATEID)),
  featureCols = 1,
  ref.append.replace = "_NEG"
)
```

Arguments

<code>thisdata</code>	is the input data frame
<code>reference</code>	is an expression that evaluates to a logical vector which specifies the observations in the data frame that are to be used for the negative control or reference set
<code>measure.columns</code>	is a character vector that specifies which columns hold the observed counts
<code>other.annotations</code>	is a character vector that specifies which additional columns in the data frame should be included in the returned data. By default we take everything, but you could specify only relevant phenotypic information.
<code>default.cast.formula</code>	is a formula that tells reshape how to recast the data frame so that rows correspond to different measured components and columns correspond to samples. By default <code>component~...</code> will put the components as the rows (i.e. positive and negative cell counts) and all measured phenotypic information on the columns.
<code>.variables</code>	is a dotted list that specifies the variable names (columns of the data frame) by which to group the data when organizing stimulated and unstimulated observations. i.e. <code>PTID x ANTIGEN x TCELLSUBSET x TESTDT</code> , or something else for your own data.
<code>featureCols</code>	is a numeric vector that specifies the indices of the columns to be used to name the features. If the casting formula is <code>component~...</code> then there is only one feature column (and it is the first one), so <code>featureCols = 1</code> , by default.
<code>ref.append.replace</code>	the terminating character string in the column names of the negative controls. It will be replaced with <code>_REF</code> for 'reference'

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
```

```
default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
.variables=.(TCELLSUBSET,CYTOKINE,UID),
featureCols=1,ref.append.replace='_REF')
```

countsTable

Extract the table of counts from a MIMOSA model

Description

Extract the table of counts from a MIMOSA model

Usage

```
countsTable(object, proportion = FALSE)

## S4 method for signature 'MIMOSAResult'
countsTable(object, proportion = FALSE)

## S4 method for signature 'MCMCResult'
countsTable(object, proportion = FALSE)

## S4 method for signature 'MDMixResult'
countsTable(object, proportion = FALSE)

## S3 method for class 'MIMOSAResultList'
countsTable(object, proportion = FALSE)

## S4 method for signature 'MIMOSAResultList'
countsTable(object, proportion = FALSE)
```

Arguments

object	a MIMOSAResult
proportion	logical return the counts or the proportions

Value

a data.frame of counts to which the model was fit.
a data.frame of counts for the stimulated and unstimulated samples

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
```

```
.variables=. (TCELLSUBSET,CYTOKINE,UID),
featureCols=1,ref.append.replace='_REF')

result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
  subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
  ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
head(countsTable(result))
head(countsTable(result,proportion=TRUE))
```

fdr

Compute the fdr (q-value) from posterior probabilities

Description

Given the z's from a MIMOSA model, calculates the q-values for each observation.

Usage

```
fdr(z)

## S3 method for class 'matrix'
fdr(z)

## S3 method for class 'MIMOSAResult'
fdr(z)

## S3 method for class 'MIMOSAResultList'
fdr(z)
```

Arguments

z matrix of posterior probabilities, or a MIMOSAResult, or MIMOSAResultList

Value

a vector of q-values or a list of vectors of q-values.

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=. (TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')
result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
```

```
subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
qvalues<-fdr(result)
```

```
getPosteriorResponseRate
```

Compute the posterior response rate from MCMC samples

Description

Uses the sampled response indicator when the model class is `MCMCResult` (or `MIMOSAResult` encapsulating an `MCMCResult`) to compute the posterior response rate distribution. This is summarized to its median, 2.5th and 97.5th credible interval.

Usage

```
getPosteriorResponseRate(x, ...)
```

Arguments

<code>x</code>	A <code>MIMOSAResultList</code> or <code>MIMOSAResult</code> . All models should be of type <code>MCMCResult</code> , or fitted using <code>method="mcmc"</code> in <code>MIMOSA</code> .
<code>...</code>	unquoted grouping variable name in the <code>pData()</code> table of all the models that specifies the one or more grouping variable by which to compute response rates.

Details

The posterior response rate is the correct way to compare response rates across studies and treatment groups, as it is unbiased compared to the response rate computed from hard thresholds of posterior response probabilities. The credible intervals have the correct behaviour as sample size increases.

Future versions will allow passing of quantiles for summarization, and maybe the full distribution, depending on needs.

Value

a tibble with the grouping variable, and the 2.5th, 25th, median, 75th, and 97.5th percentiles of the posterior response rate.

getZ	<i>Extract the posterior probabilities of response from a MIMOSA model</i>
------	--

Description

Extract the posterior probabilities of response from a MIMOSA model

Extract the component weights from a MIMOSA model

Usage

```
getZ(x)

## S3 method for class 'MIMOSAResultList'
getZ(x)

## S3 method for class 'MIMOSAResult'
getZ(x)

getW(x)

## S3 method for class 'MIMOSAResultList'
getW(x)

## S3 method for class 'MIMOSAResult'
getW(x)
```

Arguments

x output from a MIMOSA model

Value

a matrix of posterior probabilities

a vector of component weights

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=(TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')

result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
  subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
```

```
      ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
    getZ(result)
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=.(TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')

result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
  subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
  ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
getW(result)
```

ICS	<i>Stimulated and unstimulated T-cell counts for an ICS assay</i>
-----	---

Description

A data set containing T-cell counts for various stimulations and cytokines in an ICS assay.

Format

A data frame with 3960 rows

Details

- pos. The positive cell counts
- neg. The negative cell counts
- fname. The feature name (cytokine) measured
- parent. The parent T-cell population
- antigen. The antigen stimulation for this sample
- ID. The subject ID

MDMix

*EM fitting of the Multinomial Dirichlet MIMOSA model.***Description**

This function fits the multinomial dirichelt MIMOSA model using EM. It can also be used to initialize the model parameters for the MCMC model.

Usage

```
MDMix(
  data = NULL,
  modelmatrix = NULL,
  alternative = "greater",
  initonly = FALSE
)
```

Arguments

data	The observed data
modelmatrix	a model matrix specifying which components should be computed
alternative	either 'greater' or 'not equal' to fit the one-sided or two-sided model.
initonly	TRUE or FALSE to return just the initialization parameters.

Value

An object of class MDMixResult

Author(s)

Greg Finak TODO filtering of pu>ps needs to be corrected here.

MIMOSA

*Fit a MIMOSA Model***Description**

This method fits a MIMOSA model to count data stored in an ExpressionSet object.

Usage

```
MIMOSA(formula, data, ...)
```

Arguments

formula	describing the features on the lhs and the phenodata on the rhs, supporting extended formula interface with conditioning.
data	an ExpressionSet object with features on rows and samples (labelled with phenoData) on columns.
...	additional arguments

Details

The ExpressionSet should be fully annotated with featureData and phenoData. For ICS data, for example, features would be positive and negative counts for different cytokine producing cell subsets (i.e. IFNg_pos, IFNg_neg) The formula lhs should contain features and the rhs should contain phenotypic variable. See the vignette for an example.

Value

an object of type MIMOSAResult

See Also

[MIMOSA-package ConstructMIMOSAExpressionSet MIMOSAResult](#)

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=(TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')

result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
  subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
  ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
```

MIMOSAExpressionSet *Construct an ExpressionSet for MIMOSA*

Description

Starting from a reshaped data frame in the correct format, construct an ExpressionSet object that can be used with MIMOSA.

Usage

```
MIMOSAExpressionSet(df, featureCols)
```

Arguments

`df` a data.frame that is in the correct form

`featureCols` the indices of the columns that identify features.

Details

The `featureCols` will be used to construct feature names, and these columns will be dropped from the `exprs` matrix. The column names are assumed to have names that contain '_' characters separating phenotypic characteristics. These would be generated automatically if the data frame was constructed with `'reshape'`. They are used to construct the `phenoData` for the expression set

Examples

```
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=(TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')
```

MIMOSAResult	<i>Stores the result of a MIMOSA fitted model</i>
--------------	---

Description

MIMOSA returns an object of `MIMOSAResult` irrespective of which method / implementation is used to fit the data.

<code>MIMOSAResult-class</code>	<i>Stores the result of a MIMOSA fitted model</i>
---------------------------------	---

Description

MIMOSA returns an object of `MIMOSAResult` irrespective of which method / implementation is used to fit the data.

```
pData<- ,MCMCResult,ANY-method
      pData
```

Description

`pData`

`pData` extract the `phenoData` table from a MIMOSA result

Usage

```
## S4 replacement method for signature 'MCMCResult,ANY'
pData(object) <- value

## S4 replacement method for signature 'MIMOSAResult,ANY'
pData(object) <- value

## S4 replacement method for signature 'BetaMixResult,data.frame'
pData(object) <- value

## S4 method for signature 'MIMOSAResult'
pData(object)

## S4 method for signature 'MDMixResult'
pData(object)

## S4 method for signature 'MCMCResult'
pData(object)

## S4 method for signature 'BetaMixResult'
pData(object)

## S3 method for class 'MIMOSAResultList'
pData(object)

## S4 method for signature 'MIMOSAResultList'
pData(object)
```

Arguments

<code>object</code>	is the <code>MIMOSAResult</code> returned from a call to <code>MIMOSA</code>
<code>value</code>	the <code>phenoData</code> table to be assigned to the object.

Details

Extracts the `phenoData` `data.frame` from a `MIMOSAResult` object

Value

an object of type `data.frame`

```
print.MIMOSAResultList
```

Print a MIMOSAResultList

Description

Print a summary of the list of results returned by a call to MIMOSA

Usage

```
## S3 method for class 'MIMOSAResultList'  
print(x, ...)
```

Arguments

<code>x</code>	a <code>MIMOSAResultList</code>
<code>...</code>	additional arguments passed down

`show`

show

Description

Show a `MIMOSAResultList`

Usage

```
## S4 method for signature 'MIMOSAResult'  
show(object)
```

Arguments

<code>object</code>	<code>MIMOSAResultList</code>
---------------------	-------------------------------

Details

Show a summary of a `MIMOSAResultList`.

volcanoPlot

*Volcano plot for a MIMOSA model***Description**

Plots effect size vs posterior probability of response from a MIMOSAResultList, faceting by the conditioning variables.

Usage

```
volcanoPlot(x, effect_expression = NA, facet_var = NA, threshold = 0.01)
```

Arguments

x	A MIMOSAResultList
effect_expression	an expression that defines the effect size. Usually a function of the stimulated and unstimulated proportions from countsTable(x,proportion=TRUE)
facet_var	an expression defining the faceting in ggplot parlance. i.e. ~ faceting + variables
threshold	a numeric value between [0,1] for coloring significant observations (based on the q-value)

See Also

[countsTable](#)

Examples

```
data(ICS)
E<-ConstructMIMOSAExpressionSet(ICS,
  reference=ANTIGEN%in%'negctrl',measure.columns=c('CYTNUM','NSUB'),
  other.annotations=c('CYTOKINE','TCELLSUBSET','ANTIGEN','UID'),
  default.cast.formula=component~UID+ANTIGEN+CYTOKINE+TCELLSUBSET,
  .variables=.(TCELLSUBSET,CYTOKINE,UID),
  featureCols=1,ref.append.replace='_REF')

result<-MIMOSA(NSUB+CYTNUM~UID+TCELLSUBSET+CYTOKINE|ANTIGEN,
  data=E, method='EM',
  subset=RefTreat%in%'Treatment'&ANTIGEN%in%'ENV',
  ref=ANTIGEN%in%'ENV'&RefTreat%in%'Reference')
volcanoPlot(result,CYTNUM-CYTNUM_REF)
```

Index

* datasets

ICS, [12](#)

.fitMCMC, [3](#)

asinh_trans, [4](#)

BetaMixResult-class, [4](#)

boxplotMIMOSAResultList, [5](#)

combine.MIMOSA, [6](#)

ConstructMIMOSAExpressionSet, [2](#), [6](#), [14](#)

countsTable, [8](#), [18](#)

countsTable, MCMCResult-method
(countsTable), [8](#)

countsTable, MDMixResult-method
(countsTable), [8](#)

countsTable, MIMOSAResult-method
(countsTable), [8](#)

countsTable, MIMOSAResultList-method
(countsTable), [8](#)

countsTable.MIMOSAResultList
(countsTable), [8](#)

fdr, [9](#)

getPosteriorResponseRate, [10](#)

getW (getZ), [11](#)

getZ, [11](#)

ICS, [12](#)

MDMix, [13](#)

MIMOSA, [2](#), [13](#)

MIMOSA, formula, ExpressionSet-method
(MIMOSA), [13](#)

MIMOSA-package, [2](#)

MIMOSAExpressionSet, [14](#)

MIMOSAResult, [14](#), [15](#)

MIMOSAResult-class, [15](#)

pData, BetaMixResult

(pData<-, MCMCResult, ANY-method),
[16](#)

pData, BetaMixResult-method

(pData<-, MCMCResult, ANY-method),
[16](#)

pData, MCMCResult-method

(pData<-, MCMCResult, ANY-method),
[16](#)

pData, MDMixResult-method

(pData<-, MCMCResult, ANY-method),
[16](#)

pData, MIMOSAResult-method

(pData<-, MCMCResult, ANY-method),
[16](#)

pData, MIMOSAResultList-method

(pData<-, MCMCResult, ANY-method),
[16](#)

pData.MIMOSAResultList

(pData<-, MCMCResult, ANY-method),
[16](#)

pData<-, MCMCResult, ANY-method, [16](#)

pData<-, BetaMixResult, data.frame-method
(pData<-, MCMCResult, ANY-method),
[16](#)

pData<-, MIMOSAResult, ANY-method

(pData<-, MCMCResult, ANY-method),
[16](#)

print.MIMOSAResultList, [17](#)

show, [17](#)

show, MIMOSAResult-method (show), [17](#)

volcanoPlot, [18](#)