Package ‘MIMOSA’

November 20, 2016

Type Package
Title Mixture Models for Single-Cell Assays
Version 1.12.0
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 Artistic-2.0
VignetteBuilder knitr
Imports methods, Formula, data.table, pracma, MCMCpack, coda, modeest, testthat, Repp, scales, Kmisc
Suggests parallel, knitr
Depends R (>= 3.0.2), MASS, plyr, reshape, Biobase, ggplot2
LinkingTo Rcpp, RcppArmadillo
LazyLoad yes
LazyData yes
biocViews FlowCytometry, CellBasedAssays
NeedsCompilation yes

R topics documented:

  MIMOSA-package .......................................................... 2
  .fitMCMC ................................................................. 2
  asinh_trans ............................................................. 3
  BetaMixResult-class .................................................... 4
  boxplotMIMOSAResultList ............................................. 4
  combine.MIMOSA ....................................................... 5
  ConstructMIMOSAExpressionSet ........................................ 5
  countsTable ............................................................ 6
  fdr ................................................................. 7
  getZ ................................................................. 8
  ICS ................................................................. 9
  MDMix .............................................................. 10
MIMOSA-package

MIMOSA: Mixture Models for Single Cell Assays

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


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-\$4, 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.

- **init**
  - The initialization parameters for the MCMC routine. Can be initialized from `MDMix` with `initonly=TRUE`.

- **iter**
  - The number of Monte 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 parameter

- **LOWER**
  - Tuning parameter for the lower bound on the acceptance ratio of each parameter

- **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.

- **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

### Description

Arcsinh transform for `ggplot2`

### Usage

```r
asinh_trans(c)
```

### Arguments

- **c**
  - Numeric cofactor for `asinh` transform. Default 1.

### Details

Arcsinh transform for use with `coord_trans` in `ggplot2`
BetaMixResult-class

The output of fitting Beta-Binomial EM implementation BetaMix.

Description

BetaMix will return an object of this class.

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")
```
countsTable

Arguments

thisdata  is the input data frame

reference 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

measure.columns is a character vector that specifies which columns hold the observed counts

other.annotations 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.

default.cast.formula 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 component~... will put the components as the rows (i.e. positive and negative cell counts) and all measured phenotypic information on the columns.

:variables 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. PTID x ANTIGEN x TCELLSUBSET x TESTDT, or something else for your own data.

featureCols is a numeric vector that specifies the indices of the columns to be used to name the features. If the casting formula is component~... then there is only one feature column (and it is the first one), so featureCols = 1, by default.

default.cast.formula 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 component~... will put the components as the rows (i.e. positive and negative cell counts) and all measured phenotypic information on the columns.

:variables 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. PTID x ANTIGEN x TCELLSUBSET x TESTDT, or something else for your own data.

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

```r
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

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

getZ

Extract the posterior probabilities of response from a MIMOSA model

Description

Extract the posterior probabilities of response 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**

*Stimulated and unstimulated T-cell counts for an ICS assay*

**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

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')
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')

Stores the result of a MIMOSA fitted model

Description

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

pData extract the phenoData table from a MIMOSA result

Description
pData extract the phenoData table from a MIMOSA result

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

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

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

data.MIMOSAResultList(object)

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

Arguments
object is the MIMOSAResult returned from a call to MIMOSA

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, ...)

print.MIMOSAResultList
Print a MIMOSAResultList
volcanoPlot

Arguments

x a MIMOSAResultList
... additional arguments passed down

Description

Show a MIMOSAResultList

Usage

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

Arguments

object MIMOSAResultList

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

*Topic datasets
  ICS, 9
  fitMCMC, 2
  asinh_trans, 3
  BetaMixResult-class, 4
  boxplotMIMOSAResultList, 4
  combine.MIMOSA, 5
  ConstructMIMOSAExpressionSet, 2, 5, 11
  countsTable, 6, 14
  countsTable, MCMCResult-method
  (countsTable), 6
  countsTable, MDMixResult-method
  (countsTable), 6
  countsTable, MIMOSAResult-method
  (countsTable), 6
  countsTable, MIMOSAResultList-method
  (countsTable), 6
  countsTable, MIMOSAResultList
  (countsTable), 6
  pData, MCMCResult-method
  (pData, MIMOSAResult-method), 13
  pData, MIMOSAResultList
  (pData, MIMOSAResult-method), 13
  print.MIMOSAResultList, 13
  show, 14
  show, MIMOSAResult-method (show), 14
  volcanoPlot, 14

fdr, 7

getW (getZ), 8
getZ, 8

ICS, 9

MDMix, 10
MIMOSA, 2, 11
MIMOSA, Formula, ExpressionSet-method
  (MIMOSA), 11
MIMOSA-package, 2
MIMOSAExpressionSet, 12
MIMOSAResult, 11, 12
MIMOSAResult-class, 12

pData, MCMCResult-method
  (pData, MIMOSAResult-method), 13
pData, MDMixResult-method
  (pData, MIMOSAResult-method), 13
pData, MIMOSAResult-method, 13