Package 'massiR'

April 16, 2019

Type Package

Title massiR: MicroArray Sample Sex Identifier

Version 1.18.0

Date 2014-09-15

Author Sam Buckberry

Maintainer Sam Buckberry <sam.buckberry@adelaide.edu.au>

Description Predicts the sex of samples in gene expression microarray datasets

License GPL-3

Collate 'massi_y.R' 'massi_y_plot.R' 'massi_select.R' 'massi_cluster.R' 'massi_cluster_plot.R' 'massi_dip.R'

LazyData true

Depends cluster, gplots, diptest, Biobase, R (>= 3.0.2)

Suggests biomaRt, RUnit, BiocGenerics

biocViews Software, Microarray, GeneExpression, Clustering, Classification, QualityControl

git_url https://git.bioconductor.org/packages/massiR

git_branch RELEASE_3_8

git_last_commit 643b551

git_last_commit_date 2018-10-30

Date/Publication 2019-04-15

R topics documented:

`massi-package` *massiR: a microarry Gene Expression Sample Sex Identifier*

Description

massi uses y chromosome probe information to cluster samples and predict the sex of each sample in gene expression microarray datasets.

Details

Package:	massi
Type:	Package
Version:	0.99.0
Date:	2014-01-27
License:	GPL-3

The massi analysis requires a typical normalized sample/probe values produced by a microarray experiment. The `massi_y` function will extract the y chromosome probe information and cacu-late y chromosome probe variance to allow the used to select the most informative probes. Using the `massi_select` function the used can select a probe variation threshold to reduce the number of probes used in the massi.cluster step. The `massi_cluster` function clusters samples into two clusters using the y chromosome probe values. Clustering is performed using the K-medoids method as implimented in the "fpc" package. There are two plotting fucntions, `massi_y_plot` and `massi_cluster_plot`, that allow the user to explore the data at various stages of the analysis. There is also a function, `massi_dip`, that can be used to test if there may be a sample sex-bias in the dataset.

Author(s)

Sam Buckberry

Maintainer: Sam Buckberry <sam.buckberry@adelaide.edu.au>

References

Christian Hennig (2013). fpc: Flexible procedures for clustering. R package version 2.1-6. http://CRAN.R-project.org/package=fpc

Martin Maechler (2013). diptest: Hartigan's dip test statistic for unimodality - corrected code. R package version 0.75-5. http://CRAN.R-project.org/package=diptest

Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz and Bill Venables (2013). gplots: Various R programming tools for plotting data. R package version 2.12.1. http://CRAN.R-project.org/package=gplots

See Also

`massi_y`, `massi_select`, `massi_cluster`, `massi_y_plot`, `massi_dip`, `massi_cluster_plot`

Examples

```
# load the test datasets
data(massi.test.dataset, massi.test.probes)

# use the massi.y function to calculate probe variation
massi_y_out <- massi_y(expression_data=massi.test.dataset, y_probes=massi.test.probes)

# plot probe variation to aid in deciding on the most informative subset of y chromosome probes
massi_y_plot(massi_y_out)

# Extract the informative probes for clustering
massi_select_out <- massi_select(massi.test.dataset, massi.test.probes, threshold=4)

# cluster samples to predict the sex for each sample
massi_cluster_out <- massi_cluster(massi_select_out)

# get the predicted sex for each sample
data.frame(massi_cluster_out[[2]])
```

`massi.eset` *massi.eset*

Description

Object of ExpressionSet class containing the massiR test expression data.

Usage

```
data(massi.eset)
```

Format

The format is: Formal class 'ExpressionSet' [package "Biobase"] with 7 slots ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots .. .. ..@ name : chr "" .. .. ..@ lab : chr "" .. .. ..@ contact : chr "" .. .. ..@ title : chr "" .. .. ..@ abstract : chr "" .. .. ..@ url : chr "" .. .. ..@ pubMedIds : chr "" .. .. ..@ samples : list() .. .. ..@ hybridizations : list() .. .. ..@ normControls : list() .. .. ..@ preprocessing : list() .. .. ..@ other : list() .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 2 .. ..$: int [1:3] 1 0 0 .. ..$: int [1:3] 1 1 0 ..@ assayData :<environment: 0x7fd91fda9d50> ..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 60 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "rowNames" "columnNames" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. .. ..@ data :'data.frame': 1026 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ annotation : chr(0) ..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$

labelDescription: chr(0) .. .. ..@ data :'data.frame': 60 obs. of 0 variables .. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. .. ..@ .Data:List of 4 .. ..$: int [1:3] 3 0 2 .. ..$: int [1:3] 2 22 0 .. ..$: int [1:3] 1 3 0 .. ..$: int [1:3] 1 0 0

`massi.test.dataset` *The massi test dataset*

Description

This data.frame object contains expression data for 60 samples and 1026 probes

Usage

```
data(massi.test.dataset)
```

Format

A data frame with 1026 observations on the following 60 variables.

`S1` a numeric vector

`S2` a numeric vector

`S3` a numeric vector

`S4` a numeric vector

`S5` a numeric vector

`S6` a numeric vector

`S7` a numeric vector

`S8` a numeric vector

`S9` a numeric vector

`S10` a numeric vector

`S11` a numeric vector

`S12` a numeric vector

`S13` a numeric vector

`S14` a numeric vector

`S15` a numeric vector

`S16` a numeric vector

`S17` a numeric vector

`S18` a numeric vector

`S19` a numeric vector

`S20` a numeric vector

`S21` a numeric vector

`S22` a numeric vector

`S23` a numeric vector

`S24` a numeric vector
`S25` a numeric vector
`S26` a numeric vector
`S27` a numeric vector
`S28` a numeric vector
`S29` a numeric vector
`S30` a numeric vector
`S31` a numeric vector
`S32` a numeric vector
`S33` a numeric vector
`S34` a numeric vector
`S35` a numeric vector
`S36` a numeric vector
`S37` a numeric vector
`S38` a numeric vector
`S39` a numeric vector
`S40` a numeric vector
`S41` a numeric vector
`S42` a numeric vector
`S43` a numeric vector
`S44` a numeric vector
`S45` a numeric vector
`S46` a numeric vector
`S47` a numeric vector
`S48` a numeric vector
`S49` a numeric vector
`S50` a numeric vector
`S51` a numeric vector
`S52` a numeric vector
`S53` a numeric vector
`S54` a numeric vector
`S55` a numeric vector
`S56` a numeric vector
`S57` a numeric vector
`S58` a numeric vector
`S59` a numeric vector
`S60` a numeric vector

Details

This test dataset is in the data.frame format with sample names as column names and probe id's as row.names. This test dataset is a subset of an emperical dataset obtained from Illumina human-6 v2.0 expression beadchip ararys.

Source

Data were adapted from dataset GSE25906 <http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE25906>

Examples

```
data(massi.test.dataset)
```

`massi.test.probes` *The massi test probes*

Description

A data.frame containing 56 probes which correspond to y chromosome genes

Usage

```
data(massi.test.probes)
```

Format

data.frame

A data frame with 56 observations on the following 0 variables.

Examples

```
data(massi.test.probes)
```

`massi_cluster` *massi_cluster*

Description

The massi_cluster function predicts the sex of samples using k-medoids clustering.

Usage

```
massi_cluster(y_data)
```

Arguments

`y_data`	the y_data object is the data.frame returned from the `massi_select` function. This is a data.frame with sample names as column names and probe id's as row.names.

Details

This function clusters samples into two clusters using y chromosome probe values. K-medoids clustering is performed using the partitioning around medoids (pam) method implimented in the "fpc" package. The cluster with the highest probe values is determined to be the cluster of male samples and the cluster the lowest values as female samples.

Value

`cluster data`	Contains all of the results from the k-medoids clustering.
`massi.results`	Contains the results for each sample, including sample id, predicted sex, sample z-score and mean probe expression.

Author(s)

Sam Buckberry

References

Christian Hennig (2013). fpc: Flexible procedures for clustering. R package version 2.1-6. http://CRAN.R-project.org/package=fpc

See Also

`massi_y`, `massi_select`, `massi_y_plot`, `massi_dip`, `massi_cluster_plot`

Examples

```
# load the test dataset
data(massi.test.dataset, massi.test.probes)

# select the y chromosome probes using massi_select
massi_select_out <-
massi_select(massi.test.dataset, massi.test.probes)

# cluster samples to predict sex using massi_cluster
massi_cluster_out <-
massi_cluster(massi_select_out)

# get the results in a data.frame format
data.frame(massi_cluster_out[[2]])
```

`massi_cluster_plot` *massi_cluster_plot*

Description

This function produces three figures in a new graphics device to enable the exploration of the massi_cluster and massi_select results.

Usage

```
massi_cluster_plot(massi_select_data, massi_cluster_data)
```

Arguments

`massi_select_data`
: A data.frame containing the subset of y chromosome probe values for each sample. This is returned when running the massi_select function.

`massi_cluster_data`
: This is the list returned from the massi_cluster function.

Details

The first figure is a heatmap depicting probe values for each sample. The second figure is a bar plot showing the mean probe expression and standard deviation for each sample. The bars are colored with respect to the predicted sex. The third figure is a principal component plot which represents the distances bewteen samples, with each cluster highlighted with elipses.

Value

Returns three plots in a new graphics device.

Author(s)

Sam Buckberry

References

Gregory R. Warnes, Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz and Bill Venables (2013). gplots: Various R programming tools for plotting data. R package version 2.12.1. http://CRAN.R-project.org/package=gplots

See Also

`massi_cluster`, `massi_select`

Examples

```
# load the test dataset
data(massi.test.dataset, massi.test.probes)

# select the y chromosome probes using massi_select
massi_select_out <-
massi_select(massi.test.dataset, massi.test.probes)

# cluster samples to predict sex using massi_cluster
massi_cluster_out <-
massi_cluster(massi_select_out)

# produce plots using massi_cluster_plot
massi_cluster_plot(massi_select_out, massi_cluster_out)
```

massi_dip *massi_dip*

Description

The massi_dip function applies the dip test to the subset of y chromosome probe values returned from the massi_select function. This can be used to indicate if there may be either a male or female bias in the dataset. This function returns a message indicating if the dataset may have a sex bias. The results for massi_dip are not relaible for datasets with 10 or less samples.

Usage

```
massi_dip(y_subset_values)
```

Arguments

y_subset_values	A data.frame containing the subset of y chromosome probe values for each sample, as returned from the massi_select function.

Details

This function caclulates z-scores for the y.chromosome probe values returned from the massi_select function and then checks if the average z-scores for each sample show a unimodal or multi-modal distribution by applying the dip test. If the proportion of male and female samples in the dataset is relatively balanced, the distribution of average z-scores should be bi-modal. If the distribution looks unimodal, the dataset likely contains a high proportion of one sex. By testing with empirical datasets and randomly generating data subsets with different male/female proportions, guideline values were developed to provide an indication if there is a potential sex bias in the dataset. If the dip statistic is > 0.08 then the dataset is highly likely to have a porportions of male and female samples that will allow the massi_cluster function to predict the sex of samples with a high degree of accuracy. The results of this test should only be used as a guide and the results should be interpreted in light of the massi_cluster results. For more details see the massi package vignette.

Value

This function returns a list containing

dip.statistics	The results from the dip test
sample.mean.z.score	The mean of the probe z-scores for each sample used to caclulate the dip statistics
density	Density values for the z-scores. Can be informative to plot these results

Author(s)

Sam Buckberry

References

Martin Maechler (2013). diptest: Hartigan's dip test statistic for unimodality - corrected code. R package version 0.75-5. http://CRAN.R-project.org/package=diptest

See Also

massi_y, massi_select, massi_cluster, massi_y_plot, massi_cluster_plot

Examples

```
# load the test dataset
data(massi.test.dataset, massi.test.probes)

massi_select_out <- massi_select(expression_data=massi.test.dataset, y_probes=massi.test.probes, threshold=

# Use the list returned from massi.select to calculate dip statistics and z-scores.
massi_dip_out <- massi_dip(y_subset_values=massi_select_out)

# view a density plot
plot(massi_dip_out[[3]])

# view a histogram of z-scores
hist(x=massi_dip_out[[2]])
```

---

massi_select *massi_select*

Description

This function selects the y chromosome probe data for each sample.

Usage

```
massi_select(expression_data, y_probes, threshold = 3)
```

Arguments

expression_data	The expression.data item contains normalized array expression data for all samples. This can be a data.frame with sample names as columns and probe id's as row names. This argument also allows the specification of an ExpressionSet object.
y_probes	A data.frame of probe id's in one column that match y chromosome genes for the array platform. massiR includes probes for several Illumina and Affymetrix platforms. Details on using these probes are included in the vignette and the y.probes manual.
threshold	The threshold value corresponds to probe variation quantiles. This option allows the selection of the most variable probes. Deciding on a probe threshold value should be informed by viewing the plot generated by the massi_y_plot function. Threshold must be an integer "1", "2", "3", or "4". A threshold of "1" will select all y chromosome probes matching the id's in y.probes, Thresholds of "2", "3" and "4" will select probes with a CV in the top 75%, 50% and 25% respectively. The aim here is to remove probes with little to no variance across the samples. Default = 3.

Value

A data.frame containing the subset of y chromosome probe values for each sample.

Author(s)

Sam Buckberry

See Also

massi_y, massi_cluster, massi_y_plot, massi_dip, massi_cluster_plot

Examples

```
data(massi.test.dataset, massi.test.probes)

massi_select(expression_data=massi.test.dataset, y_probes=massi.test.probes)
```

---

massi_y *massi_y*

Description

The massi_y function extracts the y chromosome probe values for each sample and calculates the coefficient of variation (CV) for each probe. The returned list contains CV values (%) for each probe and quantile data. The probe variation data can be visualized using the massi_y_plot function.

Usage

```
massi_y(expression_data, y_probes)
```

Arguments

expression_data	The expression.data item contains normalized array expression data for all samples. This can be a data.frame with sample names as columns and probe id's as row names. This argument also allows the specification of an ExpressionSet object.
y_probes	A data.frame of probe id's in one column that match y chromosome genes for the array platform. massiR includes probes for several Illumina and Affymetrix platforms. Details on using these probes are included in the vignette and the y.probes manual.

Details

The expression.data must be as a data.frame with sample names as column names and probe id's as row.names. ExpressionSet objects can be input and with expression data will be exracted from the ExpressionSet and the returned list would be the same as if data as entered in data.frame format.

Value

The massi_y function returns a list containing probe id's, probe cv and quantiles.

id	Probe id's
cv	Probe cv values
quantiles	Quantiles of cv values data

Author(s)

Sam Buckberry

See Also

massi_select, massi_cluster, massi_y_plot, massi_dip, massi_cluster_plot

Examples

```
data(massi.test.dataset, massi.test.probes)
massi_y(massi.test.dataset, massi.test.probes)
```

---

massi_y_plot *massi_y_plot*

Description

The massi_y_plot function plots the data output from massi.y function.

Usage

```
massi_y_plot(massi_y_out)
```

Arguments

massi_y_out	This object is the list returned from massi_y function.

Details

This function produces a bar plot of the coefficient of variation (CV) for each probe in the dataset. This allows the user to identify the most variable probes that are likely to be the most informative in the sex prediction step. The 25%, 50% and 75% quantiles are represented as horizontal lines and represent the threshold values that can be specified for the massi_select function.

Value

This function produces a bar plot in a new graphics device.

Note

See vignette for more details.

Author(s)

Sam Buckberry

See Also

massi_y, massi_select, massi_cluster, massi_dip, massi_cluster_plot

Examples

```
data(massi.test.dataset, massi.test.probes)

massi_y_out <-
  massi_y(expression_data=massi.test.dataset, y_probes=massi.test.probes)

massi_y_plot(massi_y_out)
```

---

y.probes *Y chromosome probe list*

Description

A list containing probes corresponding to y chromosome genes for Illumina and Affymetrix platforms. Each item in the list is a data.frame of y chromosome probes that can be used in the massi analysis. The names of each item in the list correspond to the ensembl biomart attribute names.

Usage

```
data(y.probes)
```

Format

The format is: List of 6 $ illumina_humanwg_6_v1:'data.frame': 58 obs. of 0 variables $ illumina_humanwg_6_v2:'data.frame': 74 obs. of 0 variables $ illumina_humanwg_6_v1:'data.frame': 112 obs. of 0 variables $ illumina_humanht_12 :'data.frame': 112 obs. of 0 variables $ affy_hugene_1_0_st_v1:'data.fran 138 obs. of 0 variables $ affy_hg_u133_plus_2 :'data.frame': 94 obs. of 0 variables

Details

The y chromosome probes for each platform were downloaded from Ensembl biomart using the 'biomaRt' package. For more details on the methods of selecting the probes and how to obtain probes for other platform, see the vignette for the massiR package.

References

Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Steffen Durinck, Paul T. Spellman, Ewan Birney and Wolfgang Huber, Nature Protocols 4, 1184-1191 (2009).

BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. Steffen Durinck, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma and Wolfgang Huber, Bioinformatics 21, 3439-3440 (2005).

Examples

```
# load the probes list
data(y.probes)
# look at the platform names
names(y.probes)
# extract the probes using the platform name
probe.list <- y.probes[["illumina_humanwg_6_v2"]]
```

Index