Package ‘farms’

November 21, 2016

Type Package
Version 1.26.0
Date 2012-04-17
Title FARMS - Factor Analysis for Robust Microarray Summarization
Author Djork-Arne Clevert <okko@clevert.de>
Maintainer Djork-Arne Clevert <okko@clevert.de>
Depends R (>= 2.8), affy (>= 1.20.0), MASS, methods
Imports affy, MASS, Biobase (>= 1.13.41), methods, graphics
Suggests affydata, Biobase, utils
biocViews GeneExpression, Microarray, Preprocessing, QualityControl
Description The package provides the summarization algorithm called Factor Analysis for Robust Microarray Summarization (FARMS) and a novel unsupervised feature selection criterion called "I/NI-calls"
License LGPL (>= 2.1)
getNI_ProbeSets-methods.R getNI_Eset-methods.R
farms.R zzz.R

URL http://www.bioinf.jku.at/software/farms/farms.html
LazyLoad yes
NeedsCompilation no

R topics documented:

  dummy .......................................................... 2
  expFarms ....................................................... 2
  generateExprVal.method.farms ................................ 3
  getI_Eset-methods ........................................... 5
  getI_ProbeSets ............................................... 6
  getNI_Eset-methods .......................................... 6
  getNI_ProbeSets ............................................. 7
  INIcalls-methods ............................................. 8
  INI_Calls-class ............................................... 9
  lFarms ......................................................... 10
**expFarms**

**Description**

This function converts an instance of `AffyBatch` into an instance of `exprSet-class` using a factor analysis model for which a Bayesian Maximum a Posteriori method optimizes the model parameters under the assumption of Gaussian measurement noise.

**Usage**

```r
expFarms(object, bgcorrect.method = "none", pmcorrect.method = "pmonly",
         normalize.method = "quantiles", weight, mu, weighted.mean, laplacian, robust, correction, ...)
```

**Arguments**

- `object` An instance of `AffyBatch`
- `weight` Hyperparameter value in the range of [0,1] which determines the influence of the prior. The default value is 0.5
- `bgcorrect.method` the name of the background adjustment method
- `pmcorrect.method` the name of the PM adjustment method
- `normalize.method` the normalization method to use
mu Hyper-parameter value which allows to quantify different aspects of potential prior knowledge. Values near zero assumes that most genes do not contain a signal, and introduces a bias for loading matrix elements near zero. Default value is 0.

weighted.mean Boolean flag, that indicates whether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to FALSE.

laplacian Boolean flag, indicates whether a Laplacian prior for the factor is employed or not. Default value is FALSE.

robust Boolean flag, that ensures non-constant results. Default value is TRUE.

correction Value that indicates whether the covariance matrix should be corrected for negative eigenvalues which might emerge from the non-negative correlation constraints or not. Default = 0 (means that no correction is done), 1 (minimal noise (0.0001) is added to the diagonal elements of the covariance matrix to force positive definiteness), 2 (Maximum Likelihood solution to compute the nearest positive definite matrix under the given non-negative correlation constraints of the covariance matrix)

centering Indicates whether the data is "median" or "mean" centered. Default value is "median".

spuriousCorrelation Numeric value in the range of [0,1] that quantifies the suppression of spurious correlation when using the Laplacian prior. Default value is 0 (no suppression). Note, that this parameter is only active when the laplacian parameter is set to TRUE.

... other arguments to be passed to `expresso`.

Details

This function is a wrapper for `expresso`.

Value

`exprSet-class`

See Also

`expresso, qFarms, lFarms`.

Examples

data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant", weight=0.5)

```r
generateExprVal.method.farms

Generate an expression value from the probes informations

Description

Generate an expression from the probe```
Usage

generateExprVal.method.farms(probes, weight, mu, cyc, tol, weighted.mean, robust, minNoise, correction, laplacian, centering, spuriousCorrelation, ...)

Arguments

- **probes**: a matrix of probe intesities with rows representing probes and columns representing samples. Usually pm(probeset) where probeset is a of class `ProbeSet`
- **weight**: Hyperparameter value in the range of [0,1] which determines the influence of the prior. The default value is 0.5
- **mu**: Hyperparameter value which allows to quantify different aspects of potential prior knowledge. A value near zero assumes that most genes do not contain a signal, and introduces a bias for loading matrix elements near zero. Default value is 0
- **cyc**: Value which determinates the maximum numbers of EM-Steps. Default value is set to 30
- **tol**: Value which determinates the termination tolerance. Convergence threshold is set to 1E-05.
- **weighted.mean**: Boolean flag, that indicates whether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to FALSE.
- **robust**: Boolean flag, that ensures non-constant results. Default value is TRUE.
- **minNoise**: Value, minimal noise assumption. Default value is 0.0001.
- **correction**: Value that indicates whether the covariance matrix should be corrected for negative eigenvalues which might emerge from the non-negative correlation constraints or not. Default = 0 (means that no correction is done), 1 (minimal noise (0.0001) is added to the diagonal elements of the covariance matrix to force positive definiteness), 2 (Maximum Likelihood solution to compute the nearest positive definite matrix under the given non-negative correlation constraints of the covariance matrix)
- **laplacian**: Boolean flag, indicates whether a Laplacian prior for the factor is employed or not. Default value is FALSE.
- **centering**: Indicates whether the data is "median" or "mean" centered. Default value is "median".
- **spuriousCorrelation**: Numeric value in the range of [0,1] that quantifies the suppression of spurious correlation when using the Laplacian prior. Default value is 0 (no suppression). Note, that this parameter is only active when the laplacian parameter is set to TRUE.
- **...**: extra arguments to pass to the respective function

Value

A list containing entries:

- **exprs**: The expression values.
- **se.exprs**: Estimate of the hidden variable.

See Also

generateExprSet-methods, generateExprVal.method.playerout, li.wong, medianpolish
getI_Eset-methods

Examples

library(affy)
data(SpikeIn)  ##SpikeIn is a ProbeSets
probes <- pm(SpikeIn)
exprs.farms <- generateExprVal.method.farms(probes)

getI_Eset-methods  Method to generate an ExpressionSet of informative genes

Description

This function generates an instance of exprSet-class, that contains only informative probe sets.

Usage

## S4 method for signature 'INI_Calls'
getI_Eset(object)

Arguments

object  An instance of INI_Calls-class.

Value

exprSet-class

Methods

signature(object = "INI_Calls")  An instance of INI_Calls-class.

See Also

expFarms, qFarms, lFarms, INICalls, summary

Examples

data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INICalls(eset)  # apply I/NI calls
summary(INIs)
plot(INIs)  # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs)  # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs)  # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs)  # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs)  # vector containing only non-informative probe sets names
getI_ProbeSets

Method to generate a vector of informative probe set names

Description

This function generates an instance of \texttt{vector-class}, that return a vector of informative probe set names.

Usage

```r
## S4 method for signature 'INI_Calls'
getI_ProbeSets(object)
```

Arguments

- \texttt{object} An instance of \texttt{INI_Calls-class}.

Value

\texttt{vector}

Methods

signature(object = "INI_Calls") An instance of \texttt{INI_Calls-class}.

See Also

\texttt{expFarms, qFarms, lFarms, INIcalls, summary}

Examples

```r
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
```

getNI_Eset-methods

Method to generate an \texttt{ExpressionSet} of non-informative genes

Description

This function generates an instance of \texttt{exprSet-class}, that contains only non-informative probe sets.

```r
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
```
Usage

```r
## S4 method for signature 'INI_Calls'
getNI_Eset(object)
```

Arguments

- `object` An instance of `INI_Calls-class`.

Value

`exprSet-class`

Methods

signature(object = "INI_Calls") An instance of `INI_Calls-class`.

See Also

`expFarms`, `qFarms`, `lFarms`, `INIcalls`, `summary`

Examples

```r
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant"
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
```

---

**getNI_ProbeSets**

*Method to generate a vector of non-informative probe set names*

Description

This function generates an instance of vector, that return a vector of non-informative probe set names.

Usage

```r
## S4 method for signature 'INI_Calls'
getNI_ProbeSets(object)
```

Arguments

- `object` An instance of `INI_Calls-class`.

Value

`vector`
INICalls-methods

Methods

signature(object = "INI_Calls") An instance of INI_Calls-class.

See Also

expFarms, qFarms, lFarms, INICalls, summary

Examples

data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INICalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names

INICalls-methods Dimension reduction based on informative genes

Description

This function generates an instance of INI_Calls-class of given which has been summarized by expFarms, qFarms or lFarms before, based on the informative genes.

Usage

## S4 method for signature 'ExpressionSet'
INICalls(object)

Arguments

object An instance of exprSet-class.

Value

exprSet-class

Methods

signature(object = "ExpressionSet") An instance of exprSet-class.

See Also

expFarms, qFarms, lFarms, INICalls
**INI_Calls-class**

**Examples**

```r
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
```

**Description**

This is a class representation for an INI_calls-class object. The INI_calls-class consists of two instances of `exprSet-class`, containing an informative `exprSet` and a non-informative `exprSet`.

**Objects from the Class**

Objects can be created using the function `INIcalls`.

**Slots**

- `I_Calls`: Object of class "vector" containing informative probe set names.
- `NI_Calls`: Object of class "vector" containing non-informative probe set names.
- `I_Exprs`: Object of class `exprSet-class` representing the informative `exprSet`.
- `NI_Exprs`: Object of class `exprSet-class` representing the non-informative `exprSet`.
- `varZX`: Object of class "vector" containing the INI-call value.

**Author(s)**

Djork Clevert

**See Also**

`expFarms`, `qFarms`, `lFarms`, `INIcalls`

**Examples**

```r
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
```
**Description**

This function converts an instance of `AffyBatch` into an instance of `exprSet-class` using a factor analysis model for which a Bayesian Maximum a Posteriori method optimizes the model parameters under the assumption of Gaussian measurement noise. This function is a wrapper for `expresso` and uses the function `normalize.loess` for array normalization.

**Usage**

```r
lFarms(object, weight, mu, weighted.mean, laplacian, robust, correction, centering, spuriousCorrelation, ...)
```

**Arguments**

- `object` An instance of `AffyBatch`.
- `weight` Hyperparameter value in the range of [0,1] which determines the influence of the prior. The default value is 0.5.
- `mu` Hyperparameter value which allows to quantify different aspects of potential prior knowledge. Values near zero assumes that most genes do not contain a signal, and introduces a bias for loading matrix elements near zero. Default value is 0.
- `weighted.mean` Boolean flag, that indicates whether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to FALSE.
- `laplacian` Boolean flag, indicates whether a Laplacian prior for the factor is employed or not. Default value is FALSE.
- `robust` Boolean flag, that ensures non-constant results. Default value is TRUE.
- `correction` Value that indicates whether the covariance matrix should be corrected for negative eigenvalues which might emerge from the non-negative correlation constraints or not. Default = 0 (means that no correction is done), 1 (minimal noise (0.0001) is added to the diagonal elements of the covariance matrix to force positive definiteness), 2 (Maximum Likelihood solution to compute the nearest positive definite matrix under the given non-negative correlation constraints of the covariance matrix).
- `centering` Indicates whether the data is "median" or "mean" centered. Default value is "median".
- `spuriousCorrelation` Numeric value in the range of [0,1] that quantifies the suppression of spurious correlation when using the Laplacian prior. Default value is 0 (no suppression). Note, that this parameter is only active when the laplacian parameter is set to TRUE.
- `...` other arguments to be passed to `expresso`.

**Details**

This function is a wrapper for `expresso`. 
Plot-methods

Value

exprSet-class

See Also

expresso, expFarms, qFarms, normalize.loess

Examples

data(testAffyBatch)
eset <- qFarms(testAffyBatch)

plot-methods

Visualizes the distribution of informative and non-informative genes

Description

This function visualizes the distribution of informative and non-informative genes of a given instance of INI_Calls-class.

Usage

## S4 method for signature 'INI_Calls,missing'
plot(x)

Arguments

x

An instance of INI_Calls-class.

Value

exprSet-class

Methods

signature(x = "INI_Calls", y = "missing") An instance of INI_Calls-class.

See Also

expFarms, qFarms, lFarms, INIcalls, summary

Examples

data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
qFarms

qFarms expression measure

Description
This function converts an instance of AffyBatch into an instance of exprSet-class using a factor analysis model for which a Bayesian Maximum a Posteriori method optimizes the model parameters under the assumption of Gaussian measurement noise. This function is a wrapper for expresso and uses the function normalize.quantiles for array normalization.

Usage
qFarms(object, weight, mu, weighted.mean, laplacian, robust, correction, centering, spuriousCorrelation, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>An instance of AffyBatch.</td>
</tr>
<tr>
<td>weight</td>
<td>Hyperparameter value in the range of [0,1] which determines the influence of the prior. The default value is 0.5</td>
</tr>
<tr>
<td>mu</td>
<td>Hyperparameter value which allows to quantify different aspects of potential prior knowledge. Values near zero assumes that most genes do not contain a signal, and introduces a bias for loading matrix elements near zero. Default value is 0</td>
</tr>
<tr>
<td>weighted.mean</td>
<td>Boolean flag, that indicates whether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to FALSE.</td>
</tr>
<tr>
<td>laplacian</td>
<td>Boolean flag, indicates whether a Laplacian prior for the factor is employed or not. Default value is FALSE.</td>
</tr>
<tr>
<td>robust</td>
<td>Boolean flag, that ensures non-constant results. Default value is TRUE.</td>
</tr>
<tr>
<td>correction</td>
<td>Value that indicates whether the covariance matrix should be corrected for negative eigenvalues which might emerge from the non-negative correlation constraints or not. Default = 0 (means that no correction is done), 1 (minimal noise (0.0001) is added to the diagonal elements of the covariance matrix to force positive definiteness), 2 (Maximum Likelihood solution to compute the nearest positive definite matrix under the given non-negative correlation constraints of the covariance matrix)</td>
</tr>
<tr>
<td>centering</td>
<td>Indicates whether the data is &quot;median&quot; or &quot;mean&quot; centered. Default value is &quot;median&quot;.</td>
</tr>
<tr>
<td>spuriousCorrelation</td>
<td>Numeric value in the range of [0,1] that quantifies the suppression of spurious correlation when using the Laplacian prior. Default value is 0 (no suppression). Note, that this parameter is only active when the laplacian parameter is set to TRUE.</td>
</tr>
</tbody>
</table>

Details
This function is a wrapper for expresso.
Value

exprSet-class

See Also

expresso, expFarms, lFarms, normalize.quantiles

Examples

data(testAffyBatch)
eset <- qFarms(testAffyBatch)

summary-methods Summary of I/NI-calls

Description

This function determinates the percentage of informative genes of a given instance of of INI_Calls-class which has been summarized by expFarms, qFarms or lFarms before.

Usage

## S4 method for signature 'INI_Calls'
summary(object,...)

Arguments

object An instance of INI_Calls-class.
... extra arguments to pass to the respective function

Value

exprSet-class

Methods

signature(object = "INI_Calls") An instance of INI_Calls-class.

See Also

expFarms, qFarms, lFarms, plot, INIcalls

Examples

data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", normalize.method = "constant")
INIs <- INIcalls(eset) # apply I/NI calls
summary(INIs)
plot(INIs) # draws a density plot of I/NI-calls
I_data <- getI_Eset(INIs) # affybatch containing only informative probe sets
NI_data <- getNI_Eset(INIs) # affybatch containing only non-informative probe sets
I_probes <- getI_ProbeSets(INIs) # vector containing only informative probe sets names
NI_probes <- getNI_ProbeSets(INIs) # vector containing only non-informative probe sets names
Description

This is an artificial data set. It contains a 2 genes x 2 samples examples (testAffyBatch) and is suitable for testing the rd-examples in farms.

Format

An AffyBatch of 2 samples.

See Also

Dilution
Index

*Topic classes
  INI_Calls-class, 9
*Topic datasets
  dummy, 2
testAffyBatch, 14
*Topic manip
  expFarms, 2
generateExprVal.method.farms, 3
  getI_Eset-methods, 5
  getI_ProbeSets, 6
  getNI_Eset-methods, 6
  INIcalls-methods, 8
  lFarms, 10
  plot-methods, 11
  qFarms, 12
  summary-methods, 13
*Topic methods
  getI_Eset-methods, 5
  INIcalls-methods, 8
  plot-methods, 11
  summary-methods, 13
  AffyBatch, 2, 10, 12, 14
  Dilution, 14
dummy, 2
  environment, 2
  expFarms, 2, 5–9, 11, 13
  expresso, 3, 10–13
  generateExprVal.method.farms, 3
generateExprVal.method.playerout, 4
  getI_Eset (getI_Eset-methods), 5
  getI_Eset, INI_Calls-method (getI_Eset-methods), 5
  getI_Eset-methods, 5
  getI_ProbeSets, 6
  getI_ProbeSets, INI_Calls-method (getI_ProbeSets), 6
  getI_ProbeSets-methods (getI_ProbeSets), 6
  getNI_Eset (getNI_Eset-methods), 6
  getNI_Eset,INI_Calls-method (getNI_Eset-methods), 6
  getNI_Eset-methods, 6
  getNI_ProbeSets, 7
  getNI_ProbeSets,INI_Calls-method (getNI_ProbeSets), 7
  getNI_ProbeSets-methods (getNI_ProbeSets), 7
  INI_Calls (INI_Calls-class), 9
  INI_Calls-class, 9
  INIcalls, 5–9, 11, 13
  INIcalls (INIcalls-methods), 8
  INIcalls,ExpressionSet-method (INIcalls-methods), 8
  INIcalls-methods, 8
  lFarms, 3, 5–9, 10, 11, 13
  li.wong, 4
  medianpolish, 4
  normalize.loess, 10, 11
  normalize.quantiles, 12, 13
  plot, 13
  plot (plot-methods), 11
  plot,INI_Calls,missing-method (plot-methods), 11
  plot-methods, 11
  ProbeSet, 4
  qFarms, 3, 5–9, 11, 12, 13
  summary, 5–8, 11
  summary (summary-methods), 13
  summary,INI_Calls-method (summary-methods), 13
  summary-methods, 13
  testAffyBatch, 14

15