farms

October 25, 2011

INI_Calls-class *Class INI_Calls*

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

```
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "none", pmcorrect.method = "pmonly", n
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 n
```

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

Examples

```
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "rma", pmcorrect.method = "pmonly", no
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 n
```

dummy *Example cdfenv*

Description

Example cdfenv (environment containing the probe locations).

Usage

```
data(testAffyBatch)
```

Format

Containing an `environment` `dummy` containing the probe locations

expFarms *Factor Analysis for Robust Microarray Summarization*

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

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

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 wether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to TRUE .

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 = O (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".
...	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", n
```

---

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.m
```

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 number of arrays/2
tol	Value which determinates the termination tolerance. Convergence threshold is set to 1E-05.
weighted.mean	Boolean flag, that indicates wether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to TRUE .
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 = O (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".
...	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

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 = "rma", pmcorrect.method = "pmonly", no
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 n
```

getI_ProbeSets *Method to generate a vector of informative probe set names*

Description

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

Usage

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

Arguments

object	An instance of INI_Calls-class.

Value

vector

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 = "rma", pmcorrect.method = "pmonly", nc
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 n
```

---

getNI_Eset-methods *Method to generate an ExpressionSet of non-informative genes*

Description

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

Usage

```
## 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

```
data(testAffyBatch)
eset <- expFarms(testAffyBatch, bgcorrect.method = "rma", pmcorrect.method = "pmonly", no
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 n
```

---

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

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

Arguments

object	An instance of INI_Calls-class.

Value

vector

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 = "rma", pmcorrect.method = "pmonly", nc
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 n
```

`lFarms` *lFarms 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.loess` for array normalization.

Usage

```
lFarms(object, weight, mu, weighted.mean, laplacian, robust, correctio
```

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 wether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to TRUE .
`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 = O (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".
`...`	other arguments to be passed to `expresso`.

Details

This function is a wrapper for expresso.

Value

exprSet-class

See Also

expresso, expFarms, qFarms, normalize.loess

Examples

```
data(testAffyBatch)
eset <- qFarms(testAffyBatch,  weight=0.5, weighted.mean=TRUE)
```

---

plot-methods *Visualizes the distribution of informative and non-informatives 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 = "rma", pmcorrect.method = "pmonly", nc
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 n
```

`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, correcti
```

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 wether a weighted mean or a least square fit is used to summarize the loading matrix. The default value is set to TRUE.
`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 = O (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".
`...`	other arguments to be passed to `expresso`.

Details

This function is a wrapper for expresso.

Value

exprSet-class

See Also

expresso, expFarms, lFarms, normalize.quantiles

Examples

```
data(testAffyBatch)
eset <- qFarms(testAffyBatch,  weight=0.5, weighted.mean=TRUE)
```

---

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 = "rma", pmcorrect.method = "pmonly", nc
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 n
```

---

testAffyBatch *AffyBatch instance testAffyBatch*

Description

This is an artifical 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