

# Package ‘farms’

October 16, 2019

**Type** Package

**Version** 1.36.0

**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 “INI-calls”

**License** LGPL (>= 2.1)

**Collate** AllClasses.R AllGenerics.R plot-methods.R INIcalls-methods.R  
getNI\_ProbeSets-methods.R getNI\_Eset-methods.R  
getI\_ProbeSets-methods.R getI\_Eset-methods.R summary-methods.R  
farms.R zzz.R

**URL** <http://www.bioinf.jku.at/software/farms/farms.html>

**LazyLoad** yes

**git\_url** <https://git.bioconductor.org/packages/farms>

**git\_branch** RELEASE\_3\_9

**git\_last\_commit** 7203a56

**git\_last\_commit\_date** 2019-05-02

**Date/Publication** 2019-10-15

## R topics documented:

dummy	2
expFarms	2
generateExprVal.method.farms	4
getI_Eset-methods	5
getI_ProbeSets	6

getNI_Eset-methods . . . . .	7
getNI_ProbeSets . . . . .	7
INICalls-methods . . . . .	8
INI_Calls-class . . . . .	9
IFarms . . . . .	10
plot-methods . . . . .	11
qFarms . . . . .	12
summary-methods . . . . .	13
testAffyBatch . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

dummy	<i>Example cdfenv</i>
-------	-----------------------

---

### Description

Example cdfenv (environment containing the probe locations).

### Usage

```
data(testAffyBatch)
```

### Format

Containing an [environment](#) dummy containing the probe locations

---

expFarms	<i>Factor Analysis for Robust Microarray Summarization</i>
----------	--

---

### 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, robust, correction, ce
```

**Arguments**

object	An instance of <a href="#">AffyBatch</a> .
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 <a href="#">expresso</a> .

**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 = "co
```

---

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,

### Arguments

probes	a matrix of probe intensities with rows representing probes and columns representing samples. Usually pm(probeset) where probeset is a of class <a href="#">ProbeSet</a>
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](#)

**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 <a href="#">INI_Calls-class</a> .
--------	--

**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 = "co
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 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 = "none", pmcorrect.method = "pmonly", normalize.method = "co
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 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 = "none", pmcorrect.method = "pmonly", normalize.method = "co")  
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**

```
## 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 = "none", pmcorrect.method = "pmonly", normalize.method = "co
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](#)

**Examples**

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

---

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", normalize.method = "co
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

```

---

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, correction, centering, spurious)
```

## Arguments

object	An instance of <a href="#">AffyBatch</a> .
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](#).

**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**

```

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

```

**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 = "co
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, spurious)
```

## Arguments

object	An instance of <a href="#">AffyBatch</a> .
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](#).

### 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 = "co
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
```

---

testAffyBatch

*AffyBatch* instance *testAffyBatch*

---

**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, 4
  - getI\_Eset-methods, 5
  - getI\_ProbeSets, 6
  - getNI\_Eset-methods, 7
  - getNI\_ProbeSets, 7
  - 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, 3, 10, 12, 14
- Dilution, 14
- dummy, 2
- environment, 2
- expFarms, 2, 5–9, 11, 13, 14
- expresso, 3, 10–13
- generateExprVal.method.farms, 4
- generateExprVal.method.playerout, 5
- 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), 7
- getNI\_Eset, INI\_Calls-method (getNI\_Eset-methods), 7
- getNI\_Eset-methods, 7
- 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, 14
- INICalls (INICalls-methods), 8
- INICalls, ExpressionSet-method (INICalls-methods), 8
- INICalls-methods, 8
- lFarms, 3, 5–9, 10, 11, 13, 14
- li.wong, 5
- medianpolish, 5
- normalize.loess, 10, 11
- normalize.quantiles, 12, 13
- plot, 14
- plot (plot-methods), 11
- plot, INI\_Calls, missing-method (plot-methods), 11
- plot-methods, 11
- ProbeSet, 4
- qFarms, 3, 5–9, 11, 12, 13, 14
- summary, 5–8, 11
- summary (summary-methods), 13
- summary, INI\_Calls-method (summary-methods), 13
- summary-methods, 13
- testAffyBatch, 14