RPA

October 5, 2010

`RPA-package` *RPA: probe reliability and differential expression analysis*

Description

RPA estimates probe-specific variances and differential gene expression using probe-level observations of differential gene expression.

Details

Package:	RPA
Type:	Package
Version:	1.3.7
Date:	2010-04-02
License:	GPL >=2
LazyLoad:	yes

RPA.pointestimate is used to calculate probe reliability and differential expression estimates: 'rpa.results <- RPA.pointestimate(affybatch)'. The other functions are provided for users who wish to investigate the details of the algorithm more closely.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
## Not run:

## Load example data set (Dilution affybatch).
## This is a toy example with a small example dataset
## for probe reliability analysis (4 arrays).
## For practical applications, a larger sample size is
## recommended.

#require(affy)
#require(affydata)
#data(Dilution)

## Compute RPA for whole data set
## ... slow, not executed here
#rpa.results <- RPA.pointestimate(Dilution)
```

`RPA.iteration` *Estimating model parameters d and sigma2.*

Description

Finds point estimates of the model parameters d (estimated true signal underlying probe-level observations), and sigma2 (probe-specific variances).

Usage

```
RPA.iteration(S, epsilon = 1e-3,
                          alpha = NULL, beta = NULL,
                          sigma2.method = "robust", d.method = "fast",
                          maxloop = 1e6)
```

Arguments

`S`	Matrix of probe-level observations for a single probeset: samples x probes.
`epsilon`	Convergence tolerance. The iteration is deemed converged when the change in all parameters is < epsilon.
`alpha, beta`	Priors for inverse Gamma distribution of probe-specific variances. Noninformative prior is obtained with alpha, beta -> 0. Not used with sigma2.method 'var'. Scalar alpha and beta are specify equal inverse Gamma prior for all probes to regularize the solution. The defaults depend on the method.
`sigma2.method`	Optimization method for sigma2 (probe-specific variances). "robust": (default) update sigma2 by posterior mean, regularized by informative priors that are identical for all probes (user-specified by setting scalar values for alpha, beta). This regularizes the solution, and avoids overfitting where a single probe obtains infinite reliability. This is a potential problem in the other sigma2 update methods with non-informative variance priors. The default values alpha = 2; beta = 1 are used if alpha and beta are not specified. "mode": update sigma2 with posterior mean "mean": update sigma2 with posterior mean "var": update sigma2 with variance around d. Applies the fact that sigma2 cost function converges to variance with large sample sizes.

`d.method`	Method to optimize d. "fast": (default) weighted mean over the probes, weighted by probe variances The solution converges to this with large sample size. "basic": optimization scheme to find a mode used in Lahti et al. TCBB/IEEE; relatively slow; this is the preferred method with small sample sizes.
`maxloop`	Maximum number of iterations in the estimation process.

Details

Assuming data set S with P observations of signal d with Gaussian noise that is specific for each observation (specified by a vector sigma2 of length P), this method gives a point estimate of d and sigma2. Probe-level variance priors alpha, beta can be used with sigma2.methods 'robust', 'mode', and 'mean'. The d.method = "fast" is the recommended method for point computing point estimates with large sample size.

Value

A list with the following elements:

`d`	A vector. Estimated 'true' signal underlying the noisy probe-level observations.
`sigma2`	A vector. Estimated variances for each measurement (or probe).

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
## Not run:

## Preprocess probe-level data
## cind determines the 'reference' array
#Smat <- RPA.preprocess(Dilution, cind = 1)

## Pick probe-level data for one probe set
#pmindices <- pmindex(Dilution, "1000_at")[[1]]
#S <- t(Smat$fcmat[pmindices, ])

## RPA with default parameters:
#res <- RPA.iteration(S)
```

RPA.pointestimate *Computing point estimate for the model parameters for all probe sets.*

Description

Computes point estimate

Usage

```
RPA.pointestimate(abatch, sets = NULL, myseed = 101, priors =
NULL, epsilon = 1e-2, cind = 1, sigma2.method = "robust", d.method =
"fast", verbose = TRUE, bg.method = "rma", normalization.method =
"quantiles.robust", cdf = NULL, alpha = NULL, beta = NULL)
```

Arguments

`abatch`	An AffyBatch object.
`sets`	Specifies the probesets for which RPA estimates will be computed. Default: all probe sets.
`myseed`	Specifies the random seed.
`priors`	An 'rpa.priors' object. Can be used to set user-specified priors for the model parameters. Not applicable for sigma2.method = "var".
`epsilon`	Convergence tolerance. The iteration is deemed converged when the change in the d parameter is < epsilon.
`cind`	Specifies which array in abatch is used as a reference in computing probe-level differential expression.
`sigma2.method`	Optimization method for sigma2 (probe-specific variances). "robust": (default) update sigma2 by posterior mean, regularized by informative priors that are identical for all probes (user-specified by setting scalar values for alpha, beta). This regularizes the solution, and avoids overfitting where a single probe obtains infinite reliability. This is a potential problem in the other sigma2 update methods with non-informative variance priors. The default values alpha = 2; beta = 1 are used if alpha and beta are not specified. "mode": update sigma2 with posterior mean "mean": update sigma2 with posterior mean "var": update sigma2 with variance around d. Applies the fact that sigma2 cost function converges to variance with large sample sizes.
`d.method`	Method to optimize d. "fast": (default) weighted mean over the probes, weighted by probe variances The solution converges to this with large sample size. "basic": optimization scheme to find a mode used in Lahti et al. TCBB/IEEE; relatively slow; this is the preferred method with small sample sizes.
`verbose`	Print progress information during computation. Default: TRUE.
`bg.method`	Specify background correction method. Default: "rma". See bgcorrect.methods() for other options.

`normalization.method`	Specify quantile normalization method. Default: "pmonly". See normalize.methods(Dilution) for other options.
`cdf`	Specify an alternative CDF environment. Default: none.
`alpha, beta`	Prior parameters for inverse Gamma distribution of probe-specific variances. Noninformative prior is obtained with alpha, beta -> 0. Not used with sigma2.method 'var'. Scalar alpha and beta specify an identical inverse Gamma prior for all probes, which regularizes the solution. Probe-specific priors can be set with the 'priors' parameter.

Details

Calculates RPA estimates of probe reliability and differential expression between the user-specified reference array (cind) and the other arrays in the data set. The model assumes P observations for each transcript target (i.e. a probeset) with Gaussian noise which is specific for each probe (variance is specified by sigma2). The mean (affinity) parameters of the Gaussian noise model cancel out in calculating probe-level differential expression. RPA.pointestimate gives a point estimate for d and sigma2. The 'prior' parameter is not applicable with sigma2.method = "var". The d.method = "fast" is recommended with large sample size.

Value

An instance of class 'rpa'. This is an extended list containing the following elements:

`d`	A matrix of probesets x arrays. Specifies the estimated 'true' underlying differential gene expression signal over the arrays (vs. the reference array 'cind') for each investigated probeset. Note that the reference array is not included.
`sigma2`	A list. Each element corresponds to a probeset, and contains a vector that gives the estimated variance for each probe in that probeset.
`cind`	Specifies which of the arrays in the abatch (the affybatch object to be analyzed) has been used as the reference for computing probe-level differential expression.
`sets`	A character vector listing the investigated probesets.

Note

sigma2.method = "robust" and d.method = "fast" are recommended. With small sample size and informative priors, d.method = "basic" may be preferable.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE, to appear. See http://www.cis.hut.fi/projects/mi/software/RPA/

See Also

rpa.plot, rpa, set.priors, RPA.preprocess, AffyBatch

Examples

```
## Load example data set
#require(affydata)
#data(Dilution)

## Compute RPA for whole data set
## ... slow, not executed here
# rpa.results <- RPA.pointestimate(Dilution)

## Visualize the results for one of the probe sets
#rpa.plot(set, rpa.results)
```

`RPA.preprocess` *Preprocess AffyBatch object for RPA.*

Description

Background correction, quantile normalization and log2-transformation for probe-level raw data in abatch. Then probe-level differential expression is computed between the specified 'reference' array (cind) and the other arrays.

Usage

```
RPA.preprocess(abatch, cind = 1, bg.method = "rma",
                          normalization.method = "quantiles.robust", cdf = NULL)
```

Arguments

`abatch`	An AffyBatch object.
`cind`	Specify which of the arrays in abatch is used as a reference for computing probe-level differential expression.
`bg.method`	Specify background correction method. See bgcorrect.methods(abatch) for options.
`normalization.method`	Specify normalization method. See normalize.methods(abatch) for options.
`cdf`	The CDF environment used in the analysis.

Details

Probe-specific variance estimates are robust against the choice of reference array.

Value

`fcmat`	Probes x arrays preprocessed differential expression matrix.
`cind`	Specifies which array in abatch was selected as a reference in calculating probe-level differential expression.
`cdf`	The CDF environment used in the analysis.
`set.inds`	Indices for probes in each probeset, corresponding to the rows of fcmat.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

See Also

AffyBatch

`get.probe.noise.estimates`
Fetch probe-level noise estimates from an rpa object

Description

Provides probe-level estimates of noise, as given by the RPA algorithm.

Usage

```
get.probe.noise.estimates(rpa.res, sets = NULL, normalization = NULL, verbose =
```

Arguments

`rpa.res`	An rpa object.
`sets`	Probesets to check.
`normalization`	Normalization method for probe noise estimates. The higher the value, the higher the probe-level noise. By default, the estimated probe-level variances (sigma2) of the RPA model are returned. Other options include: "withinset.weights": inverse of probe-wise weights used to summarize the probe-level observations into probeset-level signal "withinset.relative": probe-wise standard deviations versus standard deviation of the probeset-level signal d. This compensates for the effect that the detected probe-level noise is typically coupled with overall signal levels. "withinset.categorical": indexes the probes according to their reliability within each probeset. Probes with higher indices are more noisy.
`verbose`	Print progress information during computation.

Details

Provides probe-specific noise estimates. The normalization options are included to improve comparability across probesets.

Value

A list. Each element corresponds to one probeset (of the input object). The element lists noise estimates for each probe within the probeset.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

See Also

RPA.pointestimate

Examples

```
## Load example data set
require(affydata)
data(Dilution)
## Compute RPA
rpa.results <- RPA.pointestimate(Dilution, set = "1000_at")
noise <- get.probe.noise.estimates(rpa.results)
```

`rpa-class` *Class "rpa"*

Description

Class for the RPA package.

Objects from the Class

Returned by `RPA.pointestimate` function. Objects can be created by calls of the form `new("rpa", ...)`.

Slots

Contains the following information for analyzed probesets and data:

Object of class `"list"`

`.Data`**d** A matrix of probesets x arrays. Specifies the estimated 'true' underlying differential gene expression signal over the arrays (vs. the reference array 'cind') for each investigated probeset. Note that the reference array is not included.

sigma2 A list. Each element corresponds to a probeset, and contains a vector that gives the estimated variance for each probe in that probeset.

cind Specifies which of the arrays in abatch was used as a reference for computing probe-level differential expression.

sets A character vector listing the investigated probesets.

cdf Alternative CDF that was used in the analysis.

data Preprocessed probe-level data on which the model was fitted.

abatch The associated affybatch object.

Extends

Class "`list`", from data part. Class "`vector`", by class "list", distance 2.

Methods

[`signature(x = "rpa")`: ...

[[`signature(x = "rpa")`: ...

show `signature(x = "rpa")`: ...

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
showClass("rpa")
```

`rpa` *RPA for preprocessing.*

Description

Returns an expressionSet object preprocessed with RPA. If 'cind' is not specified, uses the first array of affybatch as the reference.

Usage

```
rpa( abatch,
                    sets = NULL,
                  myseed = 101,
                  priors = NULL,
                 epsilon = 1e-2,
                    cind = 1,
            sigma2.method = "robust",
                d.method = "fast",
                 verbose = FALSE,
               bg.method = "rma",
    normalization.method = "quantiles.robust",
                     cdf = NULL,
                   alpha = NULL,
                    beta = NULL,
  exclude.reference.array = FALSE)
```

Arguments

`abatch`	An AffyBatch object.
`sets`	Probesets for which RPA will be computed. Default: all probe sets.
`myseed`	Specify random seed.
`priors`	An 'rpa.priors' object. Can be used to set user-specified priors for the model parameters. Not used sigma2.method = "var".
`epsilon`	Convergence tolerance. The iteration is deemed converged when the change in all parameters is < epsilon.
`cind`	Specify reference array for computing probe-level differential expression. Default: cind = 1. Note that if exclude.reference.array = TRUE the expression value for the reference array (cind) will be excluded in the output. Note that all values of the reference array are 0 since they indicate the differential expression of the reference array against itself.
`sigma2.method`	Optimization method for sigma2 (probe-specific variances). "robust": (default) update sigma2 by posterior mean, regularized by informative priors that are identical for all probes (user-specified by setting scalar values for alpha, beta). This regularizes the solution, and avoids overfitting where a single probe obtains infinite reliability. This is a potential problem in the other sigma2 update methods with non-informative variance priors. The default values alpha = 2; beta = 1 are used if alpha and beta are not specified. "mode": update sigma2 with posterior mean "mean": update sigma2 with posterior mean "var": update sigma2 with variance around d. Applies the fact that sigma2 cost function converges to variance with large sample sizes.
`d.method`	Method to optimize d. "fast": (default) weighted mean over the probes, weighted by probe variances The solution converges to this with large sample size. "basic": optimization scheme to find a mode used in Lahti et al. TCBB/IEEE; relatively slow; this is the preferred method with small sample sizes.
`verbose`	Print progress information during computation.
`bg.method`	Specify background correction method. Default: "rma". See bgcorrect.methods() for other options.
`normalization.method`	Specify quantile normalization method. Default: "pmonly". See normalize.methods(Dilution) for other options.
`cdf`	Specify an alternative CDF environment. Default: none.
`alpha, beta`	Prior (scalar) parameters for inverse Gamma distribution of probe-specific variances. Noninformative prior is obtained with alpha, beta -> 0. Not used with sigma2.method 'var'. Scalar alpha and beta specify an identical inverse Gamma prior for all probes, which regularizes the solution. Probe-specific priors can be set with the 'priors' parameter.
`exclude.reference.array`	Logical indicating whether the values for the reference array will be excluded in the final differential gene expression matrix. Note that all values of the reference array will be 0 since they indicate differential expression of the reference array against itself.

Details

Used for RPA preprocessing. Estimates the probeset-level mean parameter d of the RPA model. Returns an expressionSet object.

Value

An instance of the 'expressionSet' class.

Note

sigma2.method = "robust" and d.method = "fast" are recommended. With small sample size and informative prior, d.method = "basic" may be preferable.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

See Also

RPA.pointestimate, set.priors, AffyBatch, ExpressionSet

Examples

```
# Not run:

## Load example data set
#require(affydata)
#data(Dilution)

## Compute RPA for specific probesets
#sets <- geneNames(Dilution)[1:2]
#eset <- rpa(Dilution, sets)

## Compute RPA for whole data set
## ... slow, not executed here
## eset <- rpa(Dilution)
```

`rpa.list-class` *Class "rpa.list"*

Description

Class for the RPA package.

Objects from the Class

Objects can be created by calls of the form `new("rpa.list", ...)`.

Slots

An extended list. Contains the following information for one probeset.

Object of class `"list"`

`.Data` **d** A vector (probeset x arrays). Specifies the estimated 'true' underlying differential gene expression signal over the arrays (vs. the reference array 'cind') for the investigated probeset. Note that the reference array is not included.

sigma2 Contains a vector that gives the estimated variance for each probe in the investigated probeset.

cind Specifies which of the arrays in abatch was used as the reference for computing probe-level differential expression.

set Probeset name.

data Preprocessed probe-level data on which the model was fitted.

Extends

Class `"list"`, from data part. Class `"vector"`, by class "list", distance 2.

Methods

plot `signature(x = "rpa.list")`: ...

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
showClass("rpa.list")
```

`rpa.plot` *Plot RPA results and probe-level data for a specified probeset.*

Description

Plots the preprocessed probe-level observations, estimated probeset-level signal d, and probe-specific variances. It is also possible to highlight individual probes.

Usage

```
rpa.plot(set, rpa.object, highlight.probes = NULL, pcol = "darkgrey", dcol = "bl
```

Arguments

`set`	Probeset to visualize.
`rpa.object`	An instance of the 'rpa' class. Provided by 'RPA.pointestimate' function.
`highlight.probes`	Optionally highlight some of the probes (with dashed line)
`pcol`	Color for probe signal visualization.
`dcol`	Color for probeset-level summary d.
`cex.lab, cex.axis`	Axis adjustment parameters.

Value

Used for its side-effects.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

See Also

RPA.pointestimate

Examples

```
# Not run:

## Load example data set
#require(affydata)
#data(Dilution)

## Compute RPA for specific probesets only
#set <- "1000_at"
#rpa.results <- RPA.pointestimate(Dilution, set)

## Visualize the results for one of the probe sets
#rpa.plot(set, rpa.results)
```

`rpa.priors-class` *Class "rpa.priors"*

Description

Class for defining prior parameters in the RPA package.

Objects from the Class

Objects can be created by calls of the form `new("rpa.priors", ...)`.

An extended list with elements alpha, beta, d.

alpha A list. Each element is a list that corresponds to one probeset and specified the shape parameters of the inverse Gamma distribution for the probe-specific variance priors.

beta A list. Each element is a list that corresponds to one probeset and specified the scale parameters of the inverse Gamma distribution for the probe-specific variance priors.

d Not implemented. Can be later used to set priors for d.

Extends

Class `"list"`, from data part. Class `"vector"`, by class "list", distance 2.

Methods

No methods defined with class "rpa.priors" in the signature.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
showClass("rpa.priors")
```

rpa2eset *Coerce 'rpa' object into an 'ExpressionSet'*

Description

An instance of 'rpa' class contains differential gene expression estimates in the variable 'd'. The function 'rpa2eset' coerces this into an ExpressionSet object to allow downstream analysis of the results using standard R/BioC tools for gene expression data.

Usage

```
rpa2eset(x)
```

Arguments

`x`	An instance of the rpa class (obtained as output from RPA.pointestimate)

Value

An 'ExpressionSet' object.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
# Not run:

#require(RPA)
#require(affy)
#require(affydata)
#data(Dilution)

## Compute RPA for specific probesets
#sets <- geneNames(Dilution)[1:2]
#rpa.results <- RPA.pointestimate(Dilution,sets)

## Coerce the rpa object into an ExpressionSet
#eset <- rpa2eset(rpa.results)
```

`set.priors` *Set prior parameter object for RPA.*

Description

Reset some of the existing priors, or create a template of priors for the whole data.

Usage

```
set.priors(abatch, set, alpha, beta, priors = NULL, alpha.template = 1e-6, beta.
```

Arguments

`abatch`	An AffyBatch object.
`set`	Probeset where priors are to be determined.
`alpha, beta`	Vectors giving the values for the probe-wise prior parameters.
`priors`	Optional. Existing prior to be modified.
`alpha.template, beta.template`	Scalars. Can be used to define the default prior values for the whole-data prior template.

Details

The method returns a prior object that specifies probe-wise priors for the whole data set. By default, it sets non-informative priors for all probes, except those specified by the parameters 'set', 'alpha' and 'beta'. If a prior object is given in the input, then only the values for the specified probeset ('set') will be modified.

Value

An instance of 'rpa.priors' class.

Author(s)

Leo Lahti <leo.lahti@iki.fi>

References

Probabilistic Analysis of Probe Reliability in Differential Gene Expression Studies with Short Oligonucleotide Arrays. Lahti et al., TCBB/IEEE, to appear. See http://www.cis.hut.fi/projects/mi/software/RPA/

Examples

```
require(affy)
require(affydata)
data(Dilution)

# Create a prior object with specific alpha, beta for one probeset
alpha <- beta <- rep(1, 16)
alpha[[5]] <- 3; beta[[5]] <- 1
priors <- set.priors(Dilution, set = "1000_at", alpha, beta)
```

```
## Run RPA using the predefined priors
# rpa.results <- RPA.pointestimate(Dilution, set, priors = priors)
```

Index