Package 'multiscan'

November 12, 2026

Version 1.72.0

Date 2008-06-24

Title R package for combining multiple scans

Depends R (>= 2.3.0)

Imports Biobase, utils

biocViews Microarray, Preprocessing

Author Mizanur Khondoker <mizanur.khondoker@ed.ac.uk>, Chris Glasbey, Bruce Worton.

Maintainer Mizanur Khondoker <mizanur.khondoker@ed.ac.uk>

Description Estimates gene expressions from several laser scans of the same microarray

License GPL (>= 2)

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

git_branch RELEASE_3_23

git_last_commit c4f76b8

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

multiscan 2
murine 4
plot.multiscan 5

Index **7**

multiscan *Combining multiple laser scans of microarray data*

Description

Estimates gene expressions from multiple laser scans of microarrays using non-linear functional regression model with additive plus multiplicative errors.

Usage

```
multiscan(data, initial = NULL, na.rm = TRUE, verbose = FALSE, control = list())
```

Arguments

data	A numeric matrix or data frame containing the intensity data of a single microarray scanned at multiple (two or more) scanner settings. For dual channel arrays, multiscan should be run on each channel of data separately. The number of rows (n) is equal to the number of spots/probes on the array, and the number of columns (m) equals the number of scans. Columns will be arranged in order of scanner's sensitivity before fitting the model. Replicated probes on the array are treated as individual spots.
initial	A vector of length m+2 to be used as initial values for the scanning effects $(\beta_2, \cdots, \beta_m)$ and scale $(\sigma_1, \sigma_2, \nu)$ parameters. If it is NULL (default), the initial values are determined from the data.
na.rm	Logical. Should missing values be removed? Defaults to TRUE.
verbose	Logical. If TRUE, some intermediate results are printed as the iteration proceeds.
control	A list of control parameters. See 'Details'.

Details

The function implements the method of Khondoker *et. al.* (2006) for combining multiple laser scans of microarrays. This function is computationally slow and memory-intensive. That is due to the nested iteration loops of the numerical optimization of the likelihood function involving a large number $(n + m + 2)$ of parameters. The optimization uses an alternating algorithm with the Nelder-Mead simplex method (Nelder and Mead, 1965) in the inner loops. The function multiscan directly uses the C function nmmin, the internal code used in the general-purpose optimization tool optim, for implementing the Nelder-Mead simplex method. For large data sets with many tens of thousands of probes, it is recommended to consider first fitting the model using a random subset (e.g. 10,000 rows) of the data matrix, and then using the estimated scanning effects and scale parameters obtained as initial values for fitting the model to the full data set.

The control is a list of arguments. The users can change/supply any of the following components:

trace Indicator (0 or 1) of tracing information of Nelder-Mead algorithm. If 1, tracing information on the progress of the optimization is produced. Because Nelder-Mead may be callled thousands of times during the estimation process, setting trace = 1 will print too much information very rapidly, which may not be useful. Defaults to 0.

`gmaxit` The maximum number of global iterations. Defaults to `150`.

`maxit` The maximum number of Nelder-Mead iterations. Defaults to `5000`.

`reltol` Relative convergence tolerance of Nelder-Mead. The algorithm stops if it is unable to reduce the value by a factor of `reltol * (abs(val) + reltol)` at a step. Defaults to `1e-5`.

`globaltol` Convergence tolerance of the outer (alternating) iteration. The estimation process converges if the gain in loglikelihood from one complete cycle of the outer iteration is less than `globaltol`. Defaults to `1e-10`.

`alpha`, `beta`, `gamma` Scaling parameters for the Nelder-Mead method. `alpha` is the reflection factor (default 1.0), `beta` the contraction factor (0.5) and `gamma` the expansion factor (2.0).

Value

Returns an object of class `multiscan` with components

`call` The call of the `multiscan` function.

`beta` A vector of length `m` containing the maximum likelihood estimates of the scanning effects, the first component fixed at `1`.

`scale` A vector of length `3` containing the maximum likelihood estimates of the scale parameters $\sigma_1, \sigma_2,$ andν.

`mu` A vector of length `n` containing the estimated gene expressions.

`data` A matrix of the input data with columns rearranged in order of scanner's sensitivity.

`fitted` A matrix of the fitted model on the `data`.

`sdres` A matrix of the standardised residuals.

`outerit` Number of global iterations completed.

`gconv`, `conv`, `convmu`
Integer convergence codes.

- `gconv` Indicator of global convergence. `0` indicates successful convergence, `1` indicates premature termination.
- `conv` Convergence codes for the Nelder-Mead simplex method in the last global iteration while updating scanning effects and scale parameters. `0` for successful convergence, `1` indicates that the iteration limit `maxit` had been reached, `10` indicates degeneracy of the Nelder-Mead simplex method.
- `convmu` Convergence codes for the Nelder-Mead simplex method in the last global iteration while updating the gene expression parameters. This is an integer vector of length `n` where each component takes the value `0`, `1`, or `10` depending on whether the Nelder-Mead simplex method successfully converged, reached iteration limit `maxit` or produced degeneracy respectively while updating the corresponding gene expression parameter.

`outerit` Number of global iterations completed.

`loglf` Value of the loglikelihood function at convergence (`gconv=0`). `NA` if not converged (`gconv=1`).

References

Khondoker, M. R., Glasbey, C. A. and Worton, B. J. (2006). Statistical estimation of gene expression using multiple laser scans of microarrays. *Bioinformatics* **22**, 215–219.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal* **7** 308–313.

See Also

A web interface, created by David Nutter of Biomathematics & Statistics Scotland (BioSS), based on the original Fortran code written by Khondoker *et al.* (2006) is available at http://www.bioss.ac.uk/ktshowcase/create.cgi. Although it uses the same algorithm, results from the web interface may not be exactly identical to that of `multiscan` as it uses a different (non-free IMSL routine) implementation of Nelder-Mead simplex.

Examples

```
## load the multiscan library
library(multiscan)

## load the murine data set included in multiscan package
data(murine)
murine[1:10, ] ## see first few rows of data

## fit the model on murine data with default options
fit <- multiscan(murine)
fit

## plot the fitted model
plot(fit)

## get the estimated gene expressions
gene.exprs <- fit$mu

## see more details as iteration progresses

fit1 <- multiscan(murine, verbose = TRUE)
fit1
```

`murine` *A subset of murine macrophage gene expression data from Khondoker et al.(2006).*

Description

This data set contains a subset of 1000 rows of murine macrophage array1 data (Khondoker *et al.*, 2006). The data set has 4 columns corresponding to 4 laser scans of one channel of a microarray scanned at four different scanner settings. This data set is used in an example to illustrate the use of `multiscan` package.

Usage

```
data(murine)
```

Format

A 1000 by 4 data frame.

References

Khondoker, M. R., Glasbey, C. A. and Worton, B. J. (2006). Statistical estimation of gene expression using multiple laser scans of microarrays. *Bioinformatics* **22**, 215–219.

`plot.multiscan` *Plot fitted model and standardised residuals for a multiscan object*

Description

This function provides plots for fitted model and standardised residual for `multiscan` fitted objects.

Usage

```
## S3 method for class 'multiscan'
plot(x, residual=FALSE,...)
```

Arguments

`x`	a `multiscan` fitted object.
`residual`	Logical. Should residuals be plotted instead of the fitted model?. Defaults to `FALSE`.
`...`	Further arguments passed to the `plot` function.

Details

Fitted model on the input data, after rescaling by the corresponding scanning effects, is plotted against the estimated gene expressions. Standardised residuals for each scan of data are plotted against the rank of estimated gene expressions.

Value

Returns either one plot of the fitted model (`residual=FALSE`) on the input data or `m` plots of the residuals (`residual=TRUE`) corresponding to each scan of data.

References

Khondoker, M. R., Glasbey, C. A. and Worton, B. J. (2006). Statistical estimation of gene expression using multiple laser scans of microarrays. *Bioinformatics* **22**, 215–219.

See Also

multiscan

Examples

```
data(murine)

fit<-multiscan(murine)

## plot the fitted model

plot(fit)

## plot the residuals

op<-par(mfrow=c(2,2))
plot(fit, residual=TRUE)
par(op)
```

Index

∗ **datasets**
murine, 4
∗ **maximization**
multiscan, 2
∗ **minimization**
multiscan, 2
∗ **nonlinear**
multiscan, 2
∗ **optimize**
multiscan, 2
∗ **regression**
plot.multiscan, 5

multiscan, 2, *5, 6*
murine, 4

optim, *2*

plot.multiscan, 5