

# Package ‘TurboNorm’

October 7, 2025

**Type** Package

**Title** A fast scatterplot smoother suitable for microarray normalization

**Version** 1.57.0

**Date** 2023-15-08

**Author** Maarten van Iterson and Chantal van Leeuwen

**Maintainer** Maarten van Iterson <mviterson@gmail.com>

**Description** A fast scatterplot smoother based on B-splines with second-order difference penalty. Functions for microarray normalization of single-colour data i.e. Affymetrix/Illumina and two-colour data supplied as marray MarrayRaw-objects or limma RGList-objects are available.

**License** LGPL

**LazyLoad** yes

**Depends** R (>= 2.12.0), convert, limma (>= 1.7.0), marray

**Imports** stats, grDevices, affy, lattice

**Suggests** BiocStyle, affydata, hgu95av2cdf

**biocViews** Microarray, OneChannel, TwoChannel, Preprocessing, DNAMethylation, CpGIsland, MethylationArray, Normalization

**URL** <http://www.humgen.nl/MicroarrayAnalysisGroup.html>

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

**git\_branch** devel

**git\_last\_commit** 70039da

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.22

**Date/Publication** 2025-10-06

## Contents

|                             |   |
|-----------------------------|---|
| TurboNorm-package . . . . . | 2 |
| methylation . . . . .       | 3 |
| normalize.pspline . . . . . | 3 |
| panel.pspline . . . . .     | 5 |
| pspline . . . . .           | 6 |
| turbotrend . . . . .        | 8 |

**Index****10**


---

|                   |                                                                                   |
|-------------------|-----------------------------------------------------------------------------------|
| TurboNorm-package | <i>A fast scatterplot smoother with applications for microarray normalization</i> |
|-------------------|-----------------------------------------------------------------------------------|

---

**Description**

A fast scatterplot smoother based on B-splines with second order difference penalty. Functions for microarray normalization of single-colour data i.e. Affymetrix/Illumina and two-colour data supplied as marray MarrayRaw-objects or limma RGList-objects are available.

**Details**

|           |            |
|-----------|------------|
| Package:  | TurboNorm  |
| Type:     | Package    |
| Version:  | 1.7.2      |
| Date:     | 2013-29-01 |
| License:  | LGPL       |
| LazyLoad: | yes        |

This package contains an implementation of piecewise constant P-splines of Eilers and Marx (1996) that can be used for normalization of either single- or two-colour data. For two-colour data objects of type RGList from the limma package and MarrayRaw from the package marray can be normalized using the function `pspline()`. For single colour microarray data wrapper functions are written based on the affy package functions `normalize.loess()` and `normalize.AffyBatch.loess()` namely `normalize.pspline()` and `normalize.AffyBatch.pspline()`. Also a `panel.pspline()` is available for adding the smoothed curve to lattice graphics panels.

**Note**

The package `pspline` (S original by Jim Ramsey, R port by Brian Ripley) implements the B-spline/Natural Cubic Spline smoother

**Author(s)**

Chantal van Leeuwen and Maarten van Iterson Maintainer: Maarten van Iterson<mviterson@gmail.com>

**References**

van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. SAGMB, 11(4).

Paul .H.C. Eilers and Brain D. Marx (1996). Flexible smoothing with B-splines and Penalties. Statistical Science, Vol 11, No. 2, 89-121.

**See Also**

[turbotrend](#), [pspline](#), [normalize.pspline](#), [normalize.AffyBatch.pspline](#), [panel.pspline](#)

---

methylation

*CpG island DNA methylation array data*


---

**Description**

CpG island DNA methylation array data of a neuro-ectodermal cell line that was treated with a demethylating agent

**Usage**

```
data(methylation)
```

**Format**

"RGList" as defined in the package limma containing data from CpG island DNA methylation array data of a neuro-ectodermal cell line that was treated with a demethylating agent. The element "weights" of the "RGList" contains the subset of invariant fragments, those without methylation-sensitive restriction sites, as a logical vector.

**Details**

The data is extracted from a larger experiment described in van Iterson et al. Because the data is from a high-dense tiling array a random subset of the data was chosen for convenience in making the vignette.

**References**

van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. SAGMB, 11(4).

**Examples**

```
data(methylation)
```

---

normalize.pspline

*Functions for single-colour microarray data normalization using the P-splines*


---

**Description**

Modified version of normalize.loess and normalize.AffyBatch.pspline from the affy package uses the P-spline smoother in stead of the loess algorithm

**Usage**

```
normalize.pspline(mat, epsilon = 10^-2, maxit = 1, log.it = TRUE,
  verbose = TRUE, weights = rep(1, nrow(mat)), ...)
normalize.AffyBatch.pspline(abatch,
  type=c("together", "pmonly", "mmonly", "separate"), ...)
```

## Arguments

|                      |                                                                                    |
|----------------------|------------------------------------------------------------------------------------|
| <code>mat</code>     | a matrix with columns containing the values of the chips to normalize.             |
| <code>abatch</code>  | an <a href="#">AffyBatch</a> object.                                               |
| <code>epsilon</code> | a tolerance value (supposed to be a small value - used as a stopping criterion).   |
| <code>maxit</code>   | maximum number of iterations.                                                      |
| <code>log.it</code>  | logical. If TRUE it takes the log2 of <code>mat</code>                             |
| <code>verbose</code> | logical. If TRUE displays current pair of chip being worked on.                    |
| <code>weights</code> | For weighted normalization. The default is NULL, so there are no weights used.     |
| <code>type</code>    | A string specifying how the normalization should be applied. See details for more. |
| <code>...</code>     | Graphical parameters can be supplied.                                              |

## Details

This function is a modified version of the function [normalize.loess](#) from the `affy` package. In stead of the loess algorithm the function uses the P-spline algorithm. The `type` argument should be one of "separate", "pmonly", "mmonly", "together" which indicates whether to normalize only one probe type(PM,MM) or both together or separately.

## Value

Normalized [AffyBatch](#)

## Author(s)

Maarten van Iterson and Chantal van Leeuwen

## References

Laurent Gautier, Leslie Cope, Benjamin M. Bolstad and Rafael A. Irizarry (2004). `affy` -analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, Vol. 20, no. 3, 307-315.

van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. *SAGMB*, 11(4).

Paul .H.C. Eilers and Brain D. Marx (1996). Flexible smoothing with B-splines and Penalties. *Statistical Science*, Vol 11, No. 2, 89-121.

## See Also

[normalize.loess](#)

## Examples

```
library(affydata)
data(Dilution)
PM <- log2(pm(Dilution[,c(1,3)]))
M <- PM[,1]-PM[,2]
A <- 0.5*(PM[,1]+PM[,2])

nPM <- log2(normalize.pspline(pm(Dilution[,c(1,3)])))
nM <- nPM[,1]-nPM[,2]
```

```

nA <- 0.5*(nPM[,1]+nPM[,2])

par(mfcol=c(2,1))
plot(M~A)
plot(nM~nA)

norm <- normalize.AffyBatch.pspline(Dilution, type="pmonly")

weights <- rep(1, nrow(exprs(Dilution)))
normw <- normalize.AffyBatch.pspline(Dilution, type="pmonly", weights=weights)

```

---

|               |                                                                                         |
|---------------|-----------------------------------------------------------------------------------------|
| panel.pspline | <i>Panel function for adding a P-spline smoothed curves to a lattice graphics panel</i> |
|---------------|-----------------------------------------------------------------------------------------|

---

## Description

The function panel.pspline is similar to panel.loess but show the P-spline smoothed curve.

## Usage

```
panel.pspline(x, y, weights = rep(1, length(y)), nintervals = 100, type, horizontal = FALSE, col.lin
```

## Arguments

|                    |                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------|
| x, y               | vectors giving the coordinates of the points in the scatter plot                                                            |
| weights            | vector of weights of with same length as the data for a weighted smoothing. Default all weights are 1.                      |
| nintervals         | an integer indicating the number of intervals equal to 1 + number of knots. Currently the intervals must be larger than 10. |
| type               | see <a href="#">panel.loess</a>                                                                                             |
| horizontal         | see <a href="#">panel.loess</a>                                                                                             |
| col.line, lty, lwd | line colour, type and width that will be used in the plots, defaults are col=1, lty=1 and lwd=1.                            |
| ...                | see <a href="#">panel.loess</a>                                                                                             |

## Details

?panel.loess

## Author(s)

Maarten van Iterson and Chantal van Leeuwen

## References

- Deepayan Sarkar (2009). lattice: Lattice Graphics. R package version 0.17-26. <http://CRAN.R-project.org/package=lattice>
- van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. SAGMB, 11(4).
- Paul .H.C. Eilers and Brain D. Marx (1996). Flexible smoothing with B-splines and Penalties. Statistical Science, Vol 11, No. 2, 89-121.

## See Also

[panel.loess](#)

## Examples

```
library(marray)
library(lattice)
data(swirl)
data <- data.frame(M=as.vector(maM(swirl)), A=as.vector(maA(swirl)), Sample=rep(paste("Array", 1:4), each=nr))

xyplot(M~A|Sample, data=data,
       panel = function(x, y) {
         panel.grid(h=-1, v= 2)
         panel.xyplot(x, y)
         panel.loess(x, y, span=0.25, col="black")
         panel.pspline(x, y, col="red", lwd=2)})
```

---

|         |                                                                                  |
|---------|----------------------------------------------------------------------------------|
| pspline | <i>Function for two-colour microarray data normalization using the P-splines</i> |
|---------|----------------------------------------------------------------------------------|

---

## Description

Wrapper function for two colour microarray data normalization using the P-spline smoother suitable for a RGList- or MarrayRaw-objects.

## Usage

```
pspline(object, background = c("none", "subtract"), weights = NULL, nintervals = 100, subset=NULL,
```

## Arguments

|            |                                                                                                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------|
| object     | either a RGList or an MarrayRaw-object.                                                                                         |
| background | for background subtraction use 'subtract'. Default is no background subtraction.                                                |
| weights    | vector of weights that will be used a for a weighted normalization. The default NULL assume equal weight 1 for all data points. |
| nintervals | number of bins in which the data will be divided. The default is 100 bins.                                                      |

|                              |                                                                                                            |
|------------------------------|------------------------------------------------------------------------------------------------------------|
| showArrays                   | either a integer( > 0) or a vector of integers indicating the arrays for which a MA-plot will be produced. |
| subset                       | subset of the data on which the normalization will be based. A special case of weighted normalization.     |
| verbose                      | if TRUE gives additional information on the fit                                                            |
| .                            |                                                                                                            |
| line.col, line.lty, line.lwd | line colour, type and width that will be used in the plots, defaults are col=2, lty=1 and lwd=2.           |
| ...                          | additional graphical arguments for plotting.                                                               |

### Details

if necessary?

### Value

The value that will be returned is either a MAList or MarrayNorm-object dependening on the input type.

### Author(s)

Chantal van Leeuwen and Maarten van Iterson

### References

van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. SAGMB, 11(4).

Paul .H.C. Eilers and Brain D. Marx (1996). Flexible smoothing with B-splines and Penalties. Statistical Science, Vol 11, No. 2, 89-121.

### See Also

[normalizeWithinArrays](#), [maNormMain](#)

### Examples

```
library(marray)
data(swirl)

x <- pspline(swirl, showArrays=2, pch=20, col="grey")
x <- pspline(swirl, showArrays=2:4, line.col="green")
```

---

|            |                                                |
|------------|------------------------------------------------|
| turbotrend | <i>turbotrend: a fast scatterplot smoother</i> |
|------------|------------------------------------------------|

---

## Description

A fast scatterplot smoother based on B-splines with second order difference penalty

## Usage

```
turbotrend(x, y, w = rep(1, length(y)), n = 100, lambda=10^seq(-10, 10, length=1000), iter=0, method
```

## Arguments

|        |                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| x, y   | vectors giving the coordinates of the points in the scatter plot.                                                                               |
| w      | vector of weights of with same length as the data for a weighted smoothing. Default all weights are 1.                                          |
| n      | an integer indicating the number of intervals equal to 1 + number of knots. Currently the intervals must be larger than 10.                     |
| lambda | Optionally a user-defined penalty parameter can be provided, if not generalized cross-validation is used to find the optimal penalty parameter. |
| iter   | Number of robustifying iterations similar as lowess.                                                                                            |
| method | method for solving the system of linear equations either using the data in the original space or transformed to the Demmler-Reinsch basis.      |

## Details

some details about implementation

## Value

An object of type `pspline` is returned as a list with the following items:

|        |                                                                                          |
|--------|------------------------------------------------------------------------------------------|
| x      | original data vector x                                                                   |
| y      | fitted y-values with same length as vector x                                             |
| w      | vector of weights                                                                        |
| n      | number of bins                                                                           |
| ytrend | binnend fitted y-values                                                                  |
| xtrend | binned x-values                                                                          |
| lambda | if scalar penalty parameter used else if vector of two lower and upper bound of the grid |
| iter   | number of robustifying iterations                                                        |
| gcv    | generalized cross-validation                                                             |
| edf    | effective degrees of freedom (trace of the smoother matrix)                              |
| call   | function call which produced this output                                                 |

## Author(s)

Maarten van Iterson, Chantal van Leeuwen

## References

van Iterson M, Duijkers FA, Meijerink JP, Admiraal P, van Ommen GJ, Boer JM, van Noesel MM, Menezes RX (2012). A novel and fast normalization method for high-density arrays. *SAGMB*, 11(4).

Paul .H.C. Eilers and Brain D. Marx (1996). Flexible smoothing with B-splines and Penalties. *Statistical Science*, Vol 11, No. 2, 89-121.

## See Also

[loess](#), [lowess](#), [smooth](#), [smooth.spline](#) and [smooth.Pspline](#)

## Examples

```
library(marray)
data(swirl)

x <- maA(swirl)[,1]
y <- maM(swirl)[,1]
xord <- x[order(x)]
yord <- y[order(x)]

plot(xord, yord, main = "data(swirl) & smoothing splines + lowess")
lines(turbotrend(xord, yord), col = "red", lwd=2)
lines(smooth.spline(xord, yord), col = "green", lwd=2)
lines(lowess(xord, yord), col = "purple", lwd=2)
legend("topleft", c("piecewise constant P-splines", "Cubic B-splines", "lowess"), text.col=c("red", "green", "purple"))
```

# Index

- \* **datasets**
  - methylation, [3](#)
- \* **hplot**
  - panel.pspline, [5](#)
- \* **package**
  - TurboNorm-package, [2](#)
- \* **regression**
  - turbotrend, [8](#)
- \* **smooth**
  - normalize.pspline, [3](#)
  - pspline, [6](#)
  - turbotrend, [8](#)

AffyBatch, [4](#)

loess, [9](#)

lowess, [9](#)

maNormMain, [7](#)

methylation, [3](#)

normalize.AffyBatch.pspline, [2](#)

normalize.AffyBatch.pspline  
(normalize.pspline), [3](#)

normalize.loess, [4](#)

normalize.pspline, [2](#), [3](#)

normalizeWithinArrays, [7](#)

panel.loess, [5](#), [6](#)

panel.pspline, [2](#), [5](#)

pspline, [2](#), [6](#)

smooth, [9](#)

smooth.Pspline, [9](#)

smooth.spline, [9](#)

TurboNorm-package, [2](#)

turbotrend, [2](#), [8](#)