

Package ‘biwt’

May 7, 2026

Type Package

Title Compute the Biweight Mean Vector and Covariance & Correlation Matrice

Version 1.0.1

Date 2022-06-13

Author Jo Hardin <jo.hardin@pomona.edu>

Maintainer Jo Hardin <jo.hardin@pomona.edu>

Depends R (>= 2.1.0), robustbase, MASS

Description Compute multivariate location, scale, and correlation estimates based on Tukey's biweight M-estimator.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2022-06-13 15:50:02 UTC

NeedsCompilation no

Contents

biwt-package	2
biwt.est	3
biwtCorrelation	5
biwtInternalFunctions	7

Index	10
--------------	-----------

biwt-package *A package to compute the biweight mean vector and covariance & correlation matrices*

Description

Compute multivariate location, scale, and correlation estimates based on Tukey's biweight weight function.

Details

Package: biwt
Type: Package
Version: 1.0
Date: 2009-07-20
License: GPL-2
LazyLoad: yes

The two basic functions (1) calculate multivariate estimates of location and shape based on Tukey's biweight, and (2) compute correlations based on the biweight. The correlation functions also have options to output the data as a correlation matrix or a distance matrix (typically one minus the correlation or one minus the absolute correlation). Once the output is in a distance matrix, it can easily be converted (`as.dist()`) to an object of the class "dist" which stores the lower triangle of the correlation matrix in a vector. Many clustering algorithms take as input objects of the class "dist".

Author(s)

Jo Hardin <jo.hardin@pomona.edu>
Maintainer: Jo Hardin <jo.hardin@pomona.edu>

References

Hardin, J., Mitani, A., Hicks, L., VanKoten, B.; **A Robust Measure of Correlation Between Two Genes on a Microarray**, *BMC Bioinformatics*, **8**:220; 2007.

See Also

[biwt.est](#), [biwt.cor](#)

Examples

```
### To calculate the multivariate location vector and scale matrix:  
  
samp.data <- t(mvrnorm(30,mu=c(0,0),Sigma=matrix(c(1,.75,.75,1),ncol=2)))  
  
samp.bw <- biwt.est(samp.data)
```

```

samp.bw

samp.bw.var1 <- samp.bw$biwt.sig[1,1]
samp.bw.var2 <- samp.bw$biwt.sig[2,2]
samp.bw.cov <- samp.bw$biwt.sig[1,2]

samp.bw.cor <- samp.bw$biwt.sig[1,2] /
sqrt(samp.bw$biwt.sig[1,1]*samp.bw$biwt.sig[2,2])
samp.bw.cor

### To calculate the correlation(s):

samp.data <- t(mvrnorm(30,mu=c(0,0,0),
Sigma=matrix(c(1,.75,-.75,.75,1,-.75,-.75,-.75,1),ncol=3)))

# To compute the 3 pairwise correlations from the sample data:

samp.bw.cor <- biwt.cor(samp.data, output="vector")
samp.bw.cor

# To compute the 3 pairwise correlations in matrix form:

samp.bw.cor.mat <- biwt.cor(samp.data)
samp.bw.cor.mat

# To compute the 3 pairwise distances in matrix form:

samp.bw.dist.mat <- biwt.cor(samp.data, output="distance")
samp.bw.dist.mat

# To convert the distances into an object of class `dist'

as.dist(samp.bw.dist.mat)

```

biwt.est

A function to compute Tukey's biweight mean vector and covariance matrix

Description

Compute a multivariate location and scale estimate based on Tukey's biweight weight function.

Usage

```
biwt.est(x, r=.2, med.init=covMcd(x))
```

Arguments

<code>x</code>	a $2 \times n$ matrix or data frame (n is the number of measurements)
<code>r</code>	breakdown (k/n where k is the largest number of measurements that can be replaced with arbitrarily large values while keeping the estimates bounded). Default is $r=.2$.
<code>med.init</code>	a (robust) initial estimate of the center and shape of the data. The format is a list with components center and cov (as in the output of <code>covMcd</code> from the <code>rrocv</code> library). Default is the minimum covariance determinant (MCD) on the data.

Details

A robust measure of center and shape is computed using Tukey's biweight M-estimator. The biweight estimates are essentially weighted means and covariances where the weights are calculated based on the distance of each measurement to the data center with respect to the shape of the data. The estimates should be computed pair-by-pair because the weights should depend only on the pairwise relationship at hand and not the relationship between all the observations globally.

Value

A list with components:

<code>biwt.mu</code>	the final estimate of center
<code>biwt.sig</code>	the final estimate of shape

Note

If there is too much missing data or if the initialization is not accurate, the function will compute the MCD for a given pair of observations before computing the biweight correlation (regardless of the initial settings given in the call to the function).

Author(s)

Jo Hardin <jo.hardin@pomona.edu>

References

Hardin, J., Mitani, A., Hicks, L., VanKoten, B.; **A Robust Measure of Correlation Between Two Genes on a Microarray**, *BMC Bioinformatics*, **8**:220; 2007.

See Also

biwt.cor

Examples

```
samp.data <- t(mvrnorm(30,mu=c(0,0),Sigma=matrix(c(1,.75,.75,1),ncol=2)))
samp.bw <- biwt.est(samp.data)
```

```

samp.bw

samp.bw.var1 <- samp.bw$biwt.sig[1,1]
samp.bw.var2 <- samp.bw$biwt.sig[2,2]
samp.bw.cov <- samp.bw$biwt.sig[1,2]

samp.bw.cor <- samp.bw.cov / sqrt(samp.bw.var1 * samp.bw.var2)
samp.bw.cor

# or:

samp.bw.cor <- samp.bw$biwt.sig[1,2] /
sqrt(samp.bw$biwt.sig[1,1]*samp.bw$biwt.sig[2,2])
samp.bw.cor

#####
# to speed up the calculations, use the median/mad for the initialization:
#####

samp.init <- list()
samp.init$cov <- diag(apply(samp.data,1,mad,na.rm=TRUE))
samp.init$center <- apply(samp.data,1,median,na.rm=TRUE)

samp.init

samp.bw <- biwt.est(samp.data,med.init = samp.init)
samp.bw.cor <- samp.bw$biwt.sig[1,2] /
sqrt(samp.bw$biwt.sig[1,1]*samp.bw$biwt.sig[2,2])
samp.bw.cor

```

biwtCorrelation	<i>A function to compute a weighted correlation based on Tukey's bi-weight</i>
-----------------	--

Description

The following function compute a multivariate location and scale estimate based on Tukey's bi-weight weight function.

Usage

```
biwt.cor(x, r=.2, output="matrix", median=TRUE, full.init=TRUE, absval=TRUE)
```

Arguments

x a $g \times n$ matrix or data frame (g is the number of observations (genes), n is the number of measurements)

<code>r</code>	breakdown (k/n where k is the largest number of measurements that can be replaced with arbitrarily large values while keeping the estimates bounded). Default is <code>r=.2</code> .
<code>output</code>	a character string specifying the output format. Options are "matrix" (default), "vector", or "distance". See value below
<code>median</code>	a logical command to determine whether the initialization is done using the coordinate-wise median and MAD^2 (TRUE, default) or using the minimum covariance determinant (MCD) (FALSE). Using the MCD is substantially slower. The MAD is the median of the absolute deviations from the median. See the R help file on <code>mad</code> .
<code>full.init</code>	a logical command to determine whether the initialization is done for each pair separately (FALSE) or only one time at the beginning using a random sample from the data matrix (TRUE, default). Initializing for each pair separately is substantially slower.
<code>absval</code>	a logical command to determine whether the distance should be measured as 1 minus the absolute value of the correlation (TRUE, default) or simply 1 minus the correlation (FALSE)

Details

Using `biwt.est` to estimate the robust covariance matrix, a robust measure of correlation is computed using Tukey's biweight M-estimator. The biweight correlation is essentially a weighted correlation where the weights are calculated based on the distance of each measurement to the data center with respect to the shape of the data. The correlations are computed pair-by-pair because the weights should depend only on the pairwise relationship at hand and not the relationship between all the observations globally. The `biwt` functions simply compute many pairwise correlations and create distance matrices for use in other algorithms (e.g., clustering).

In order for the biweight estimates to converge, a reasonable initialization must be given. Typically, using TRUE for the `median` and `full.init` arguments will provide acceptable initializations. With particularly irregular data, the MCD should be used to give the initial estimate of center and shape. With data sets in which the observations are orders of magnitudes different, `full.init=FALSE` should be specified.

Value

Specifying "matrix" for the `output` argument returns a matrix of the biweight correlations.

Specifying "vector" for the `output` argument returns a vector consisting of the lower triangle of the correlation matrix stored by columns in a vector, say `bwcor`. If g is the number of observations and `bwcor` is the correlation vector, then for $i < j \leq g$, the biweight correlation between (rows) i and j is `bwcor[(j - 1) * (j - 2) / 2 + i]`. The length of the vector is $g * (g - 1) / 2$, i.e., of order g^2 .

Specifying "distance" for the `output` argument returns a matrix of the biweight distances (default is 1 minus absolute value of the biweight correlation).

Note

If there is too much missing data or if the initialization is not accurate, the function will compute the MCD for a given pair of observations before computing the biweight correlation (regardless of the initial settings given in the call to the function).

The "vector" output option is given so that correlations can be stored as vectors which are less computationally intensive than matrices.

Author(s)

Jo Hardin <jo.hardin@pomona.edu>

References

Hardin, J., Mitani, A., Hicks, L., VanKoten, B.; **A Robust Measure of Correlation Between Two Genes on a Microarray**, *BMC Bioinformatics*, **8**:220; 2007.

See Also

[biwt.est](#)

Examples

```
samp.data <-t(mvrnorm(30,mu=c(0,0,0),
Sigma=matrix(c(1,.75,-.75,.75,1,-.75,-.75,-.75,1),ncol=3)))

# To compute the 3 pairwise correlations from the sample data:

samp.bw.cor <- biwt.cor(samp.data, output="vector")
samp.bw.cor

# To compute the 3 pairwise correlations in matrix form:

samp.bw.cor.mat <- biwt.cor(samp.data)
samp.bw.cor.mat

# To compute the 3 pairwise distances in matrix form:

samp.bw.dist.mat <- biwt.cor(samp.data, output="distance")
samp.bw.dist.mat

# To convert the distances into an object of class `dist'

as.dist(samp.bw.dist.mat)
```

biwtInternalFunctions *Functions used internally for the biwt package*

Description

Tukey's biweight gives robust estimates of a p-dimensional mean vector and covariance matrix. These functions are used internally within the biweight estimation function.

Usage

```

chi.int2.p(p, a, c1)
chi.int2(p, a, c1)
chi.int.p(p, a, c1)
chi.int(p, a, c1)
erho.bw.p(p, c1)
erho.bw(p, c1)
ksolve(d, p, c1, b0)
psibw(x, c1)
rhobw(x, c1)
vbw(x, c1)
wtbw(x, c1)
rejpt.bw(p, r)
vect2diss(v)

```

Arguments

p	the dimension of the data (should be two if computing correlations. Unlike Pearson correlation, pairwise correlations will not be the same if computed on the entire data set as compared to one pair at a time.)
a	degrees of freedom for the chi square distribution
c1	cutoff value at which the biweight function gives zero weight to any data point
d	vector of distances from each data point to mean vector
b0	expected value of the ρ function for the biweight estimator (under normality)
x	value at which the biweight (ρ, ψ, v, w) should be evaluated
r	breakdown (k/n where k is the largest number of observations that can be replaced with arbitrarily large values while keeping the estimates bounded)
v	a vector (presumably from <code>biwt.cor</code>) consisting of the lower triangle of a symmetric dissimilarity or similarity matrix

Details

These functions are used internally for the `biwt.est` and `biwt.cor` functions in the `biwt` package.

Value

The following functions evaluate partial integrals of the χ^2 distribution: `chi.int`, `chi.in2`, `chi.int.p`, `chi.int2.p`.

The following functions evaluate the biweight functions: `psibw`, `rhobw`, `wbw`, `vbw`.

The following functions calculate the expected value of the ρ function under the assumption of normally distributed data: `erho.bw`, `erho.bw.p`.

The function `ksolve` keeps the estimates from imploding by setting the mean value of ρ equal to its expected value under normality.

The function `rejpt.bw` gives the asymptotic rejection point.

The function `vect2diss` converts a vector consisting of a lower triangle of a matrix into a symmetric dissimilarity or similarity matrix. The function is similar to `dissmatrix` in the `hopach` package, except that `vect2diss` fills in the lower triangle first while `dissmatrix` fills in the upper triangle first.

Author(s)

Jo Hardin <jo.hardin@pomona.edu>

References

Hardin, J., Mitani, A., Hicks, L., VanKoten, B.; **A Robust Measure of Correlation Between Two Genes on a Microarray**, *BMC Bioinformatics*, **8**:220; 2007.

See Also

biwt.est, biwt.cor

Examples

```
## These are not user level functions
## See examples for biwt.est or biwt.cor
## ?biwt.est
## ?biwt.cor
```

Index

* cluster

- biwt-package, 2
- biwt.est, 3
- biwtCorrelation, 5

* multivariate

- biwt-package, 2
- biwt.est, 3
- biwtCorrelation, 5

* robust

- biwt-package, 2
- biwt.est, 3
- biwtCorrelation, 5

biwt, 8

biwt (biwt-package), 2

biwt-package, 2

biwt.cor, 2, 4, 8, 9

biwt.cor (biwtCorrelation), 5

biwt.est, 2, 3, 6–9

biwtCorrelation, 5

biwtInternalFunctions, 7

chi.int (biwtInternalFunctions), 7

chi.int2 (biwtInternalFunctions), 7

erho.bw (biwtInternalFunctions), 7

ksolve (biwtInternalFunctions), 7

psibw (biwtInternalFunctions), 7

rejpt.bw (biwtInternalFunctions), 7

rhobw (biwtInternalFunctions), 7

vbw (biwtInternalFunctions), 7

vect2diss (biwtInternalFunctions), 7

wtbw (biwtInternalFunctions), 7