Package: biwt (via r-universe)

October 25, 2024

Type Package

Title Functions to Compute the Biweight Mean Vector and Covariance and Correlation Matrices

Version 1.1.0

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

Description The base functions compute multivariate location, scale, and correlation estimates based on Tukey's biweight M-estimator. Using the base function, the computations can be applied to a large number of observations to create either a matrix of biweight distances or biweight correlations.

Depends R (>= 3.0.0)

Imports robustbase, stats, MASS

License MIT + file LICENSE

Encoding UTF-8

LazyLoad yes

URL https://github.com/hardin47/biwt, https://hardin47.github.io/biwt/

BugReports https://github.com/hardin47/biwt/issues

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggplot2, dplyr, purrr, tidyr, readr, gt, dendextend, colorspace, cluster, usedist, gplots

Config/testthat/edition 3

VignetteBuilder knitr

Repository https://hardin47.r-universe.dev

RemoteUrl https://github.com/hardin47/biwt

RemoteRef HEAD

RemoteSha f0e6c30a799bc6c7d29466108e7d53e2c856398d

Contents

`biwt.cor` *A function to compute the biweight mean vector and covariance matrix*

Description

A function to compute the biweight mean vector and covariance matrix

Usage

```
biwt.cor(
  x,
  r = 0.2,
  output = "matrix",
  median = TRUE,
  full.init = TRUE,
  absval = TRUE
)
```

Arguments

x	a g x n matrix or data frame (n is the number of measurements, g is the number of observations (genes))
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). Default is r = 0.2.
output	a character string specifying the output format. Options are "matrix" (default), "vector", or "distance". See value below.
median	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 R help file on mad.
full.init	a logical command to determine whether the initialization is done for each pair separately (FALSE) or only one time at the beginning using the entire data matrix (TRUE, default). Initializing for each pair separately is substantially slower.
absval	a logical command to determine whether the distance should be measured as 1 minus the absolute value of the correlation (TRUE, default) or as 1 minus the correlation (FALSE).

Value

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 $i < j <= g$, 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 "matrix" for the output argument returns a matrix of the biweight correlations.

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

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.

Returns a list with components:

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

dist.mat	a matrix consisting of the correlations converted to distances (either 1 - correlation or 1 - abs(correlation)).

Examples

```
# note that biwt.cor() takes data that is gxn where the
# goal is to find correlations or distances between each of the g items

samp.data <- t(MASS::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 the 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 = 0.2, med.init = robustbase::covMcd(x))
```

Arguments

x	a 2 x n matrix or data frame (n is the number of measurements)
r	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 = 0.2.
med.init	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 covMcd() from the rrcov package). Default is the minimum covariance determinant (MCD) on the data.

Value

A list with components

`biwt.mu`	the final estimate of center
`biwt.sig`	the final estimate of shape

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.

Examples

```
# note that biwt.est() takes data that is 2xn where the
# goal is to find the correlation between the 2 items

samp.data <- t(MASS::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$cov <- diag(apply(samp.data, 1, stats::mad, na.rm=TRUE))
samp.init$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
```

biwt_cor *A function to compute the biweight mean vector and covariance matrix*

Description

A function to compute the biweight mean vector and covariance matrix

Usage

```
biwt_cor(x, r, median = TRUE, full_init = TRUE)
```

Arguments

x	an n x g matrix or data frame (n is the number of measurements, g is the number of observations (genes))
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)
median	a logical command to determine whether the initialization is done using the coordinate-wise median and MAD (TRUE) or using the minimum covariance determinant (MCD) (FALSE). Using the MCD is substantially slower.
full_init	a logical command to determine whether the initialization is done for each pair separately (FALSE) or only one time at the beginning using the entire data matrix (TRUE). Initializing for each pair separately is substantially slower.

Value

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 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.

Returns a list with components:

biwt_corr	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, i.e., then for $i < j \leq g$, the biweight correlation between (rows) i and j is bwcor[$g*(i-1)-i*(i-1)/2+j-i$]. The length of the vector is $g*(g-1)/2$, i.e., of order g^2.

biwt_NAid	a vector which is indexed in the same way as biwt_corr. The entries represent whether the biweight correlation was possible to compute (will be NA if too much data is missing or if the initializations are not accurate). 0 if computed accurately, 1 if NA.

Examples

```
# note that biwt_cor() takes data that is nxg where the
# goal is to find correlations between each of the g items

samp_data <- MASS::mvrnorm(30,mu=c(0,0,0),Sigma=matrix(c(1,.75,-.75,.75,1,-.75,-.75,-.75,1),ncol=3))
r <- 0.2 # breakdown

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

samp_bw_cor <- biwt_cor(samp_data,r)
samp_bw_cor
```

---

biwt_cor_matrix *A function to compute the biweight mean vector and covariance matrix*

Description

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

Usage

```
biwt_cor_matrix(x, r, median = TRUE, full_init = TRUE)
```

Arguments

x	an n x g matrix or data frame (n is the number of measurements, g is the number of observations (genes))
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)
median	a logical command to determine whether the initialization is done using the coordinate-wise median and MAD (TRUE) or using the minimum covariance determinant (MCD) (FALSE). Using the MCD is substantially slower.
full_init	a logical command to determine whether the initialization is done for each pair separately (FALSE) or only one time at the beginning using the entire data matrix (TRUE). Initializing for each pair separately is substantially slower.

Value

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 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.

Returns a list with components:

`biwt_corr_matrix`
: a matrix of the biweight correlations.

`biwt_NAid_matrix`
: a matrix representing whether the biweight correlation was possible to compute (will be NA if too much data is missing or if the initializations are not accurate). 0 if computed accurately, 1 if NA.

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.

Examples

```
# note that biwt_cor_matrix() takes data that is nxg where the
# goal is to find correlations between each of the g items

samp_data <- MASS::mvrnorm(30,mu=c(0,0,0),Sigma=matrix(c(1,.75,-.75,.75,1,-.75,-.75,-.75,1),ncol=3))
r <- 0.2 # breakdown

# To compute the 3 pairwise correlations in matrix form:

samp_bw_cor_mat <- biwt_cor_matrix(samp_data,r)
samp_bw_cor_mat
```

`biwt_dist_matrix` *A function to compute the biweight mean vector and covariance matrix*

Description

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

Usage

```
biwt_dist_matrix(x, r, median = TRUE, full_init = TRUE, absval = TRUE)
```

Arguments

x	an n x g matrix or data frame (n is the number of measurements, g is the number of observations (genes))
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)
median	a logical command to determine whether the initialization is done using the coordinate-wise median and MAD (TRUE) or using the minimum covariance determinant (MCD) (FALSE). Using the MCD is substantially slower.
full_init	a logical command to determine whether the initialization is done for each pair separately (FALSE) or only one time at the beginning using the entire data matrix (TRUE). Initializing for each pair separately is substantially slower.
absval	a logical command to determine whether the distance should be measured as 1 minus the absolute value of the correlation (TRUE) or simply 1 minus the correlation (FALSE)

Value

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 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.

Returns a list with components:

biwt_dist_matrix	a matrix of the biweight distances (default is 1 minus absolute value of the biweight correlation).
biwt_NAid_matrix	a matrix representing whether the biweight correlation was possible to compute (will be NA if too much data is missing or if the initializations are not accurate). 0 if computed accurately, 1 if NA.

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.

Examples

```
# note that biwt_dist_matrix() takes data that is nxg where the
# goal is to find distances between each of the g items

samp_data <- MASS::mvrnorm(30,mu=c(0,0,0),Sigma=matrix(c(1,.75,-.75,.75,1,-.75,-.75,-.75,1),ncol=3))
r <- 0.2 # breakdown

# To compute the 3 pairwise distances in matrix form:
samp_bw_dist_mat <- biwt_dist_matrix(samp_data, r)
samp_bw_dist_mat

# To convert the distances into an element of class 'dist'
as.dist(samp_bw_dist_mat$biwt_dist_mat)
```

`biwt_est` *A function to compute the 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, med.init)
```

Arguments

`x`	an `n` x 2 matrix or data frame (`n` is the number of observations)
`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)
`med.init`	a (robust) initial estimate of the center and shape of the data. form is a list with components center and cov.

Value

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.

Returns a list with components:

`biwt_mu`	the final estimate of location
`biwt_sig`	the final estimate of scatter
`biwt_NAid`	a logical of whether the given initialization was used (coded as 0) or whether a more precise initialization was used (namely, the pair by pair MCD, coded as 1).

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.

Examples

```
# note that biwt_est() takes data that is nx2 where the
# goal is to find the correlation between the 2 items

samp_data <- MASS::mvrnorm(30,mu=c(0,0),Sigma=matrix(c(1,.75,.75,1),ncol=2))
r <- 0.2 # breakdown

samp_mcd <- robustbase::covMcd(samp_data)
samp_bw <- biwt_est(samp_data, r, samp_mcd)

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_corr <- samp_bw_cov / sqrt(samp_bw_var1 * samp_bw_var2)

# or:

samp_bw_corr <- samp_bw$biwt_sig[1,2] / sqrt(samp_bw$biwt_sig[1,1]*samp_bw$biwt_sig[2,2])
samp_bw_corr

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

samp_init <- list()
samp_init$cov <- diag(apply(samp_data, 2, stats::mad, na.rm=TRUE))
samp_init$center <- apply(samp_data, 2, median, na.rm=TRUE)

samp_bw <- biwt_est(samp_data, r, samp_init)
samp_bw_corr <- samp_bw$biwt.sig[1,2] / sqrt(samp_bw$biwt.sig[1,1]*samp_bw$biwt.sig[2,2])
```

---

chi.int *Internal function*

Description

Internal function

Usage

```
chi.int(p, a, c1)
```

Arguments

p	a number
a	a number
c1	a number

Value

a number

Examples

```
chi.int(2,3,4)
```

`chi.int.p` *Title*

Description

Title

Usage

```
chi.int.p(p, a, c1)
```

Arguments

p	a number
a	a number
c1	a number

Value

a number

Examples

```
chi.int.p(1,2,3)
```

chi.int2 *Internal function*

Description

Internal function

Usage

```
chi.int2(p, a, c1)
```

Arguments

p	a number
a	a number
c1	a number

Value

a number

Examples

```
chi.int(2,3,4)
```

chi.int2.p *Internal function*

Description

Internal function

Usage

```
chi.int2.p(p, a, c1)
```

Arguments

p	a number
a	a number
c1	a number

Value

a number

Examples

```
chi.int2.p(2,3,4)
```

`erho.bw` *Internal function*

Description

Internal function

Usage

```
erho.bw(p, c1)
```

Arguments

p	a number
c1	a cutoff

Value

a number, The expected value of rho

Examples

```
erho.bw(2,3)
```

`erho.bw.p` *Internal function*

Description

Internal function

Usage

```
erho.bw.p(p, c1)
```

Arguments

p	a number
c1	a cutoff

Value

a number, The derivative of the expected value of rho

Examples

```
erho.bw.p(2,3)
```

`ksolve` *Internal function*

Description

Internal function

Usage

```
ksolve(d, p, c1, b0)
```

Arguments

`d`	a vector
`p`	a number
`c1`	a number
`b0`	a number

Value

a number

Examples

```
ksolve(rnorm(20,.1,2),1,3,1)
```

`psibw` *Internal function*

Description

Internal function

Usage

```
psibw(x, c1)
```

Arguments

`x`	a vector
`c1`	a cutoff

Value

a vector

Examples

```
psibw(rnorm(10),3)
```

`rejpt.bw` *Internal function*

Description

Internal function

Usage

```
rejpt.bw(p, r)
```

Arguments

p	a number
r	the breakdown

Value

the asymptotic rejection point

Examples

```
rejpt.bw(2,3)
```

`rhobw` *Internal function*

Description

Internal function

Usage

```
rhobw(x, c1)
```

Arguments

x	a number
c1	a cutoff

Value

a number

Examples

```
rhobw(2,3)
```

---

vbw *Internal function*

Description

Internal function

Usage

```
vbw(x, c1)
```

Arguments

x	a vector
c1	a cutoff

Value

a vector

Examples

```
vbw(rnorm(10),3)
```

---

vect2diss *Internal function*

Description

Internal function

Usage

```
vect2diss(v)
```

Arguments

v	a vector

Value

a vector

Examples

```
vect2diss(rnorm(10))
```

---

vect2diss_hop *Internal function*

Description

Internal function

Usage

```
vect2diss_hop(v)
```

Arguments

v	a vector

Value

a vector

Examples

```
vect2diss_hop(rnorm(10))
```

---

wtbw *Internal function*

Description

Internal function

Usage

```
wtbw(x, c1)
```

Arguments

x	a vector
c1	a number

Value

a vector

Examples

```
wtbw(rnorm(10),3)
```

Index