Package ‘lineup2’

October 13, 2022

Version 0.6
Date 2021-06-14
Title Lining Up Two Sets of Measurements
Description Tools for detecting and correcting sample mix-ups between two sets of measurements, such as between gene expression data on two tissues. This is a revised version of the 'lineup' package, to be more general and not tied to the 'qtl' package.

Author Karl W Broman [aut, cre] (<https://orcid.org/0000-0002-4914-6671>)
Maintainer Karl W Broman <broman@wisc.edu>
Depends R (>= 3.5.0)
Imports parallel, Rcpp (>= 0.12.12)
Suggests knitr, rmarkdown, testthat, devtools, roxygen2
License GPL-3
URL https://github.com/kbroman/lineup2
BugReports https://github.com/kbroman/lineup2/issues
LinkingTo Rcpp
VignetteBuilder knitr
LazyData true
Encoding UTF-8
ByteCompile true
RoxygenNote 7.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2021-06-15 05:10:03 UTC
align_matrix_cols

Align the columns of two matrices using their column names, omitting columns that are not present in both.

Usage

align_matrix_cols(x, y)

Arguments

x       A matrix
y       Another matrix

Value

A list with the input x and y matrices, with the columns aligned using their names. Columns not in both matrices are omitted.

See Also

align_matrix_rows()
align_matrix_rows

Examples

# using the provided lineup2ex data (a list of two matrices)
# reduces to the common columns and puts the columns in the same order
# (using the column names)
aligned <- align_matrix_cols(lineup2ex$gastroc, lineup2ex$islet)

align_matrix_rows  Align the rows of two matrices

Description

Align the rows of two matrices using their row names, omitting rows that are not present in both.

Usage

align_matrix_rows(x, y)

Arguments

  x    A matrix

  y    Another matrix

Value

A list with the input x and y matrices, with the rows aligned using their names. Rows not in both matrices are omitted.

See Also

align_matrix_cols()

Examples

# using the provided lineup2ex data (a list of two matrices)
# reduces to the common rows and puts the rows in the same order
# (using the row names)
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
**Description**

For matrices x and y, calculate the correlation between columns of x and columns of y.

**Usage**

```r
corr_betw_matrices(
  x,  
  y, 
  what = c("paired", "bestright", "bestpairs", "all"), 
  corr_threshold = 0.9, 
  align_rows = TRUE, 
  cores = 1 
)
```

**Arguments**

- `x`: A numeric matrix.
- `y`: A numeric matrix with the same number of rows as `x`.
- `what`: Indicates which correlations to calculate and return. See value, below.
- `corr_threshold`: Threshold on correlations if `what"bestpairs"`.
- `align_rows`: If TRUE, align the rows in the two matrices by the row names.
- `cores`: Number of CPU cores to use, for parallel calculations. (If 0, use `parallel::detectCores()`.) Alternatively, this can be links to a set of cluster sockets, as produced by `parallel::makeCluster()`.

**Details**

Missing values (NA) are ignored, and we calculate the correlation using all complete pairs, as in `stats::cor()` with `use = "pairwise.complete.obs"`.

**Value**

If `what"paired"`, the return value is a vector of correlations, between columns of x and the corresponding column of y. x and y must have the same number of columns.

If `what="bestright"`, we return a data frame of size `ncol(x)` by 3, with the `i`th row being the maximum correlation between column `i` of x and a column of y, and then the y-column index and y-column name with that correlation. (In case of ties, we give the first one.)

If `what="bestpairs"`, we return a data frame with five columns, containing all pairs of columns (with one in x and one in y) with correlation ≥ `corr_threshold`. Each row corresponds to a column pair, and contains the correlation and then the x- and y-column indices followed by the x- and y-column names.

If `what="all"`, the output is a matrix of size `ncol(x)` by `ncol(y)`, with all correlations between columns of x and columns of y.
dist_betw_arrays

Distance between rows of two arrays

Description

Calculate the distances between the rows of two multi-dimensional arrays.

Usage

dist_betw_arrays(x, y, distance = c("rmsd", "mad", "propdiff"), cores = 1)

Arguments

x       A numeric array.
y       A second numeric array, with the same dimensions as x.
distance Indicates whether to use Euclidean distance ("rmsd" for root mean square difference), the mean absolute difference ("mad"), or the proportion of differences ("propdiff").
cores   Number of CPU cores to use, for parallel calculations. (If 0, use parallel::detectCores(). Alternatively, this can be links to a set of cluster sockets, as produced by parallel::makeCluster.)
Details

The two arrays need to have the same dimensions, except for the leading dimension (rows). They are turned into matrices by merging all but the leading dimension, and then they’re sent to `dist_betw_matrices()`.

Value

If \( x \) and \( y \) have \( m \) and \( n \) rows, respectively, the result is an \( m \) by \( n \) matrix whose \((i,j)\)th element is the distance between the \( i \)th row of \( x \) and the \( j \)th row of \( y \).

See Also

`dist_betw_matrices()`, `corr_betw_matrices()`

Examples

```r
p <- 10
k <- 6
n <- 5
m <- 3
x <- array(stats::rnorm(n*k*p), dim=c(n,k,p))
rownames(x) <- LETTERS[1:n]
y <- array(stats::rnorm(m*k*p), dim=c(m,k,p))
rownames(y) <- letters[1:m]
d <- dist_betw_arrays(x, y)
```

---

### dist_betw_matrices

**Distance between rows of two matrices**

**Description**

Calculate the distances between the rows of one matrix and the rows of a second matrix.

**Usage**

```r
dist_betw_matrices(  
x,  
y,  
distance = c("rmsd", "mad", "propdiff"),  
align_cols = TRUE,  
cores = 1  
)
```
get_2ndbest

Arguments

x       A numeric matrix.
y       A second numeric matrix, with the same number of columns as x.
distance Indicates whether to use Euclidean distance ("rmsd" for root mean square difference), the mean absolute difference ("mad"), or the proportion of differences ("propdiff").
align_cols If TRUE, align the columns in the two matrices by the column names.
cores Number of CPU cores to use, for parallel calculations. (If 0, use parallel::detectCores(). Alternatively, this can be links to a set of cluster sockets, as produced by parallel::makeCluster().

Value

If x is m by p and y is n by p, then the result is an m by n matrix whose (i,j)th element is the distance between the ith row of x and the jth row of y.

See Also

corr_betw_matrices(), dist_betw_arrays()

dist_betw_arrays()

Examples

p <- 10
n <- 5
m <- 3
x <- matrix(stats::rnorm(n*p), ncol=p)
rownames(x) <- LETTERS[1:n]
y <- matrix(stats::rnorm(m*p), ncol=p)
rownames(y) <- letters[1:m]
d <- dist_betw_matrices(x, y)

get_2ndbest

Get 2nd-smallest distance for each individual

Description

For each individual represented in a distance matrix, find the 2nd-smallest entry (with NAs for individuals present in only the rows or only the columns).

Usage

gbt_2ndbest(d, dimension = c("row", "column"), get_min = TRUE)
get_best

Arguments

d A distance matrix
dimension Whether to get the 2nd-best by row or by column
get_min If TRUE, get the 2nd-minimum; if FALSE, get the 2nd-maximum

Value

A vector with all distinct individuals, with the 2nd-smallest (or largest) value by row or column. We include all individuals so that the results are aligned with the results of get_self().

See Also

get_self(), get_best(), which_2ndbest(), get_nonself()

Examples

# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                 t(lineup2ex$islet[,selected_genes]), "all")
# second-biggest value by row
secbest_byrow <- get_2ndbest(similarity, get_min=FALSE)
# second-biggest value by column
secbest_bycol <- get_2ndbest(similarity, get_min=FALSE, dimension="column")

get_best

Get smallest distance for each individual

Description

For each individual represented in a distance matrix, find the smallest entry (with NAs for individuals present in only the rows or only the columns).

Usage

get_best(d, dimension = c("row", "column"), get_min = TRUE)

Arguments

d A distance matrix
dimension Whether to get the minimum by row or by column
get_min If TRUE, get the minimum; if FALSE, get the maximum
Value

A vector with all distinct individuals, with the minimum (or maximum) value by row or column. We include all individuals so that the results are aligned with the results of `get_self()`.

See Also

`get_self()`, `get_2ndbest()`, `which_best()`, `get_nonself()`

Examples

```r
# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                 t(lineup2ex$islet[,selected_genes], "all")
# maximum value by row
best_byrow <- get_best(similarity, get_min=FALSE)
# maximum value by column
best_bycol <- get_best(similarity, get_min=FALSE, dimension="column")
```

---

**get_nonself**

Get self-nonself distances

Description

Return the distance matrix with all self-self distances replaced with NAs (and so just containing the self-self distances).

Usage

`get_nonself(d)`

Arguments

- `d` A distance matrix

Value

The input distance matrix with all self-self distances replaced with NAs.

See Also

`get_self()`, `get_best()`, `get_2ndbest()`
Examples

```r
# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                 t(lineup2ex$islet[,selected_genes]), "all")
# pull out the non-self similarities
nonself <- get_nonself(similarity)
```

get_probs

---

**get_probs**

*Summarize potential problems in a distance matrix*

**Description**

For the individuals represented in a distance matrix, collect the self-self, best, and 2nd best distances, and summarize the results in a data frame.

**Usage**

```r
get_probs(
  d,                   # A distance or similarity matrix
  dimension = c("row", "column"),
  get_min = TRUE,      # If TRUE, get the minimum (for a distance matrix); if FALSE, get the maximum (for a similarity matrix)
  subset = c("problems", "all"),
  threshold = 0        # If subset="problems", the threshold on the difference between the self and best distances.
)
```

**Arguments**

- `d`: A distance or similarity matrix.
- `dimension`: Whether to determine the best distances within rows or columns.
- `get_min`: If TRUE, get the minimum (for a distance matrix); if FALSE, get the maximum (for a similarity matrix).
- `subset`: Whether to return just the rows with potential problems, or all of the rows.
- `threshold`: If subset="problems", the threshold on the difference between the self and best distances.

**Value**

A data frame containing individual ID, distance to self, best distance and corresponding individual, 2nd best distance and the corresponding individual.
**get_self**

Get self-self distance

**Description**

For each individual represented in a distance matrix, pull the self-self entry (with NAs for individuals present in only the rows or only the columns).

**Usage**

```r
get_self(d)
```

**Arguments**

- `d` A distance matrix

**Value**

A vector with all distinct individuals, with the self-self values

**See Also**

- `get_best()`, `get_2ndbest()`, `get_nonself()`
Examples

```r
# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[, selected_genes]),
                                 t(lineup2ex$islet[, selected_genes]), "all")
# pull out the self-self similarities
self <- get_self(similarity)
```

```r
hist_self_nonself
```

Plot histograms of self-self and self-nonself distances

**Description**

Plot histograms of self-self and self-nonself distances

**Usage**

```r
hist_self_nonself(d, breaks = NULL, rug = TRUE, xlabel = "distance")
```

**Arguments**

- `d`  
  A distance matrix
- `breaks`  
  Histogram breaks (default is to use 100 intervals)
- `rug`  
  If TRUE, use `graphics::rug()` to plot tick marks at the observed values, below the histograms.
- `xlabel`  
  Label on x-axes (e.g., "similarity" vs "distance")

**Details**

We use the `mrow` arg for `graphics::par()` to make a two-panel figure.

**Value**

None.

**See Also**

- `get_self()`, `get_nonself()`
Examples

```r
# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
     t(lineup2ex$islet[,selected_genes]), "all")
# histograms of the self and non-self distances
hist_self_nonself(similarity)
```

Description

Example dataset for lineup2 package, with gene expression data for a selected set of 200 genes on two tissues on a set of about 500 mice, with 100 genes chosen to be highly correlated between the two tissues and 100 chosen at random.

Usage

```r
data(lineup2ex)
```

Format

List of two matrices, with gene expression data for gastrocnemius muscle (`gastroc`) and pancreatic islets (`islet`), at a selected set of 200 genes (100 are highly correlated between the two tissues, and 100 others chosen at random). The matrices have samples as rows and genes as columns. The row names are sample identifiers. There are 498 samples for gastroc and 499 samples for islet, with 497 samples in common.

Source

https://phenome.jax.org/projects/Attie1

References


plot_sample

Plot the distances for a given sample

Description

Plot the distances for a given sample, highlighting itself and the closest sample.

Usage

plot_sample(
  d,
  sample,
  dimension = c("row", "column"),
  get_min = TRUE,
  add_labels = TRUE,
  ...
)

Arguments

d        A distance or similarity matrix
sample    Sample ID (in row or column names)
dimension Whether to look at the row or column
get_min   If TRUE, get the minimum (for a distance matrix); if FALSE, get the maximum
          (for a similarity matrix)
add_labels If TRUE, label the individual sample and the optimal sample
...       Passed to points()

Value

None.

Examples

# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                  t(lineup2ex$islet[,selected_genes], "all"))
which_2ndbest

Determine which individual has 2nd-smallest distance to each individual

Description
For each individual represented in a distance matrix, find the individual giving the 2nd-smallest entry (with NAs for individuals present in only the rows or only the columns).

Usage

```r
which_2ndbest(d, dimension = c("row", "column"), get_min = TRUE)
```

Arguments

- `d`: A distance matrix
- `dimension`: Whether to get the 2nd-best by row or by column
- `get_min`: If TRUE, get the 2nd-minimum; if FALSE, get the 2nd-maximum

Value
A vector with all distinct individuals, with the character string labels for the individuals giving the 2nd-smallest (or largest) value by row or column. We include all individuals so that the results are aligned with the results of `get_self()`.

See Also

get_2ndbest(), get_self(), get_best(), which_best()

Examples

```r
# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                  t(lineup2ex$islet[,selected_genes]), "all")
# which sample gives second-biggest value by row
secbest_byrow <- which_2ndbest(similarity, get_min=FALSE)

# which sample gives second-biggest value by column
secbest_bycol <- which_2ndbest(similarity, get_min=FALSE, dimension="column")
```
which_best

Determine which individual has smallest distance to each individual

Description

For each individual represented in a distance matrix, find the individual giving the smallest entry (with NAs for individuals present in only the rows or only the columns).

Usage

which_best(d, dimension = c("row", "column"), get_min = TRUE)

Arguments

d
A distance matrix

dimension
Whether to get the minimum by row or by column

get_min
If TRUE, get the minimum; if FALSE, get the maximum

Value

A vector with all distinct individuals, with the character string labels for the individuals giving the minimum (or maximum) value by row or column. We include all individuals so that the results are aligned with the results of get_self().

See Also

get_best(), get_self(), get_2ndbest(), which_2ndbest()

Examples

# align rows in the provided dataset, lineup2ex
aligned <- align_matrix_rows(lineup2ex$gastroc, lineup2ex$islet)
# find correlated columns
selected_genes <- (corr_betw_matrices(aligned[[1]], aligned[[2]], "paired") > 0.75)
# calculate correlation between rows
similarity <- corr_betw_matrices(t(lineup2ex$gastroc[,selected_genes]),
                                t(lineup2ex$islet[,selected_genes]), "all")
# which sample gives maximum value by row
best_byrow <- which_best(similarity, get_min=FALSE)

# which sample gives maximum value by column
best_bycol <- which_best(similarity, get_min=FALSE, dimension="column")
Index

* datasets
  lineup2ex, 13
align_matrix_cols, 2
align_matrix_cols(), 3
align_matrix_rows, 3
align_matrix_rows(), 2
corr_betw_matrices, 4
corr_betw_matrices(), 6, 7
dist_betw_arrays, 5
dist_betw_arrays(), 5, 7
dist_betw_matrices, 6
dist_betw_matrices(), 5, 6
get_2ndbest, 7
get_2ndbest(), 9, 11, 15, 16
get_best, 8
get_best(), 8, 9, 11, 15, 16
get_nonself, 9
get_nonself(), 8, 9, 11, 12
get_problems, 10
get_self, 11
get_self(), 8, 9, 11, 12, 15, 16
graphics::par(), 12
graphics::rug(), 12
hist_self_nonself, 12
lineup2ex, 13
parallel::detectCores(), 4, 5, 7
parallel::makeCluster(), 4, 5, 7
plot_sample, 14
stats::cor(), 4

which_2ndbest, 15
which_2ndbest(), 8, 16
which_best, 16
which_best(), 9, 11, 15