

Package: PhenotypeSpace (via r-universe)

October 8, 2026

Type Package

Title Quantifying phenotypic spaces

Version 0.1.1

Author Marcelo Araya-Salas, Karan Odom & Alejandro Rico-Guevara

Maintainer Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

Description Facilitates quantifying phenotypic trait spaces and comparing spaces across groups.

License GPL (>= 2)

Encoding UTF-8

URL <https://marce10.github.io/PhenotypeSpace/>

BugReports <https://github.com/maRce10/PhenotypeSpace/issues/>

RoxygenNote 7.3.2

Language en-US

Imports pbapply, viridis, adehabitatHR, sp, stats, raster, MASS, vegan, spatstat.explore, spatstat.geom, proxy, grDevices, methods, rlang, sf, nicheROVER

Suggests caret, randomForest, C50, plotly

Depends R (>= 4.0.0)

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://marce10.r-universe.dev>

Date/Publication 2026-04-11 00:02:06 UTC

RemoteUrl <https://github.com/maRce10/PhenotypeSpace>

RemoteRef HEAD

RemoteSha 6e3b9a80b3003494449730fc400ba0e86e554300

Contents

binary_triangular_matrix	2
distance_to_rectangular	3
example_space	5
plot_space	6
rarefact_space_similarity	9
rarefact_space_size	11
rarefact_space_size_difference	13
rectangular_to_triangular	15
space_similarity	16
space_size	21
space_size_difference	23
Index	26

binary_triangular_matrix
<i>Get binary triangular matrices</i>

Description

binary_triangular_matrix creates binary triangular matrices representing categorical data in a distance matrix form

Usage

```
binary_triangular_matrix(group, labels = NULL)
```

Arguments

group	Character vector or factor containing categories to be represented as a pairwise binary matrix. Several observations per categories (at least some categories) are required.
labels	Character vector or factor containing labels to be used for rows/columns in the output matrix. Optional. Default is NULL.

Details

The function creates binary triangular matrices representing categorical data in a pairwise distance matrix form. Such matrices represent group membership by assigning 0 to pairs of observations that belong to the same category (individual, group, population) and 1 to those belonging to different categories. Binary pairwise matrices can be useful to evaluate association between a categorical and continuous variable (represented as pairwise distances) using Mantel test (as in Araya-Salas et al. 2019).

Value

A pairwise distance matrix that represents group membership. See details.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

Araya-Salas M, G Smith-vidaurre, D Mennill, P González-Gómez, J Cahill & T Wright. 2019. Social group signatures in hummingbird displays provide evidence of co-occurrence of vocal and visual learning. Proceedings of the Royal Society B. 286: 20190666.

Smouse PE, Long JC, Sokal RR. 1986 Multiple regression and correlation extensions of the Mantel test of matrix correspondence. Syst. Zool. 35, 627–632.

See Also

[distance_to_rectangular](#), [rectangular_to_triangular](#)

Examples

```
{
# create 3 groups each one with 2 observations
groups <- paste0("G", rep(1:3, each = 2))
# create binary matrix
binary_triangular_matrix(group = groups)

# create binary matrix using labels
binary_triangular_matrix(group = groups, labels = paste(groups, 1:6, sep = "-"))
}
```

distance_to_rectangular

Convert pairwise distance matrices to rectangular matrices

Description

distance_to_rectangular converts binary triangular matrices to rectangular matrices using Multidimensional Scaling.

Usage

```
distance_to_rectangular(
  distance.matrix,
  labels = names(distance.matrix),
  n.dimensions = 2,
  metric = TRUE,
  ...
)
```

Arguments

<code>distance.matrix</code>	Distance matrix (i.e. object of class 'dist'). Can be created using the function dist or converted to using as.dist .
<code>labels</code>	Character vector or factor containing labels to be used for rows/columns in the output data frame. Default is <code>names(distance.matrix)</code> . Must be the same length as the number of observations in 'distance.matrix'.
<code>n.dimensions</code>	Integer vector of length 1 indicating the number of dimensions to represent distances in a new space. Default is 2.
<code>metric</code>	Logical argument to control if Metric (a.k.a. Classical, TRUE, default) or Non-Metric MULTidimensional Scaling (FALSE) is used to project in a new n-dimension space. Non-Metric MDS is conducted using the function isoMDS while Classical MDS uses the function cmdscale . So yes, it is a silly wrapper over those 2 functions.
<code>...</code>	Additional arguments to be passed to train (only used for machine learning models).

Details

It is a silly wrapper over 2 multidimensional scaling functions ([isoMDS](#) and [cmdscale](#)) that simplifies the calculation of Multidimensional Scaling and formatting of its output to be used with other functions in the package.

Value

A data frame with the new dimensions representing the position of observations in a new n-dimension space. If `metric = FALSE` the output data frame is embedded in a list that also includes the stress value.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[distance_to_rectangular](#), [rectangular_to_triangular](#)

Examples

```
{
data("example_space")

dist_example <- dist(example_space[example_space$group %in% c("G1", "G2"),
```

```

c("dimension_1", "dimension_2"]])

# convert into a 2-dimension space
rect_example <- distance_to_rectangular(distance.matrix = dist_example,
metric = TRUE)

head(rect_example)

# convert into a 2-dimension space with non-metric MDS
rect_example <- distance_to_rectangular(distance.matrix = dist_example,
metric = FALSE, maxit = 3)

}

```

example_space

Example bi-dimensional space data

Description

example_space is a data frame with 1550 observations of a simulated phenotypic tri-dimensional space including 5 groups.

Usage

```
data(example_space)
```

Format

An object of class `data.frame` with 1550 rows and 5 columns.

Source

Marcelo Araya Salas, PhenotypeSpace

Examples

```

{
# load data
data("example_space")

#plot space in 2 dimensions
xs <- tapply(example_space$dimension_1, example_space$group, mean)
ys <- tapply(example_space$dimension_2, example_space$group, mean)
plot(example_space[, c("dimension_1", "dimension_2")],
      col = example_space$color, pch = 20, cex = 1.8)
text(xs, ys, labels = names(xs), cex = 2.5)

# Install and load necessary libraries

```

```

rlang::check_installed("plotly")

library(plotly)

# plot space in 4 dimensions
plot_ly(
  data = example_space,
  x = ~dimension_1,
  y = ~dimension_2,
  z = ~dimension_3,
  type = "scatter3d",
  mode = "markers",
  alpha = 0.8,
  marker = list(size = 4),
  color = ~ group,
  colors = unique(example_space$color)
)
}

```

plot_space

Plot bidimensional trait spaces

Description

plot_space plots bidimensional trait spaces

Usage

```

plot_space(
  X,
  dimensions,
  indices,
  basecex = 1,
  title = NULL,
  colors = c("#3E4A89FF", "#35B779FF"),
  point.colors = colors,
  point.alpha = 0.7,
  point.cex = 1,
  background.indices = NULL,
  pch = 1,
  labels = c("sub-space", "total space"),
  legend.pos = "topright",
  density.alpha = 0.6,
  ...
)

```

Arguments

<code>X</code>	Data frame containing columns for the dimensions of the phenotypic space (numeric) of the total trait space. This is required so the extent of the plotting area represents the overall trait space in which the sub-space is found.
<code>dimensions</code>	Character vector of length 2 with the names of the columns containing the dimensions of the phenotypic space.
<code>indices</code>	Numeric vector with the indices of the rows in 'X' to be used as sub-space for plotting.
<code>basecex</code>	Numeric vector of length 1 controlling the relative size of the axis labels and tick labels, legend and title. Legend and title are multiply by 1.5 (<code>basecex * 1.5</code>) to increase size compare to axis text.
<code>title</code>	Character vector of length 1 to be used as the plot title. Default is NULL.
<code>colors</code>	Character vector with the colors to use for density plotting. 2 values must be supplied if 'background.indices' is supplied. Default is <code>c("#3E4A89FF", "#35B779FF")</code> .
<code>point.colors</code>	Character vector with the colors to use for point plotting. 2 values must be supplied if 'background.indices' is supplied. Default is the same as "colors".
<code>point.alpha</code>	Numeric vector of length 1 ≥ 0 and ≤ 1 with the alpha value for color transparency. Default is 0.7. If 0 points are not plotted.
<code>point.cex</code>	Numeric vector of length 1 controlling the relative size of the points. Default is 1. If 0 points are not plotted.
<code>background.indices</code>	Numeric vector with the indices of the rows in 'X' to be used as background traits space for plotting. Points from 'indices' will be plotted on top of these points.
<code>pch</code>	Either an integer specifying a symbol or a single character to be used as the default in plotting points. See points for possible values and their interpretation.
<code>labels</code>	Character vector with the labels to be used in the legend. Not used if <code>legend.pos = NULL</code> or if 'background.indices' is not supplied. Default is <code>c("sub-space", "total space")</code> .
<code>legend.pos</code>	Controls the position of the legend. Can take the following values: "bottom-right", "bottom", "bottomleft", "left", "topleft", "top", "topright" (default), "right" and "center". If NULL the legend is not plotted.
<code>density.alpha</code>	Numeric vector of length 1 ≥ 0 and ≤ 1 with the alpha value for color transparency to be used in the highest density regions. Lower density regions will gradually increase in transparency starting from the supplied value. Default is 0.6. If 0 densities are not plotted.
<code>...</code>	Additional arguments to be passed to plot for plot customization.

Details

The function plots a sub-group of data (i.e. sub-space) within the overall trait space. The total trait space can also be plotted in the background. By default both points and kernel densities are shown. Graphs are returned in the active graphic device.

Value

A single panel plot in the active graphic device.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[distance_to_rectangular](#), [rectangular_to_triangular](#)

Examples

```
{
  data("example_space")

  # no background
  plot_space(X = example_space, dimensions = c("dimension_1", "dimension_2"),
    indices = which(example_space$group == "G2"))

  # add background
  plot_space(X = example_space, dimensions = c("dimension_1", "dimension_2"),
    indices = which(example_space$group == "G2"),
    background.indices = which(example_space$group != "G2"))

  # change legend labels
  plot_space(X = example_space, dimensions = c("dimension_1", "dimension_2"),
    indices = which(example_space$group == "G2"),
    background.indices = which(example_space$group != "G2"),
    labels = c("G3", "trait space"))

  # change legend position
  plot_space(X = example_space, dimensions = c("dimension_1", "dimension_2"),
    indices = which(example_space$group == "G2"),
    background.indices = which(example_space$group != "G2"),
    labels = c("G3", "trait space"), legend.pos = "left")

  # with title
  plot_space(X = example_space, dimensions = c("dimension_1", "dimension_2"),
    indices = which(example_space$group == "G2"),
    background.indices = which(example_space$group != "G2"),
    labels = c("G3", "trait space"), legend.pos = "bottomleft", title = "G3")
}
```

rarefact_space_similarity

Calculates rarefacted space overlaps

Description

rarefact_space_similarity

Usage

```

rarefact_space_similarity(
  formula,
  data,
  n = NULL,
  replace = FALSE,
  seed = NULL,
  cores = 1,
  pb = TRUE,
  iterations = 30,
  ...
)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form group ~ dim1 + dim2 where dim1 and dim2 are the dimensions of the phenotype space and group refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
n	Integer vector of length 1 indicating the number of samples to be use for rarefaction (i.e. how many samples per group will be gather at each iteration). Default is the minimum sample size across groups.
replace	Logical argument to control if sampling is done with replacement. Default is FALSE.
seed	Integer vector of length 1 setting the seed (see set.seed). If used results should be the same on different runs, so it makes them replicable.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
pb	Logical argument to control if progress bar is shown. Default is TRUE.
iterations	Integer vector of length 1. Controls how the number of times the rarefaction routine is iterated. Default is 30.
...	Additional arguments to be passed to space_similarity for customizing similarity measurements.

Details

The function applies a rarefaction sub-sampling procedure for evaluating pairwise space similarity (internally using [space_similarity](#)). The spread and shape of a phenotypic space might change as a function of number of samples. Hence, ideally, spaces should be compared between groups of similar sample sizes. Rarefaction allows to compare groups of unbalanced sample sizes by randomly re-sampling observations using the same number samples across groups iteratively.

Value

A data frame containing the mean, minimum, maximum and standard deviation of the similarity metric across iterations for each pair of groups. If the similarity metric is not symmetric (e.g. the proportional area of A that overlaps B is not necessarily the same as the area of B that overlaps A, see [space_similarity](#)) separated columns are supplied for the two comparisons.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[space_similarity](#), [rectangular_to_triangular](#)

Examples

```
{
# load data
data("example_space")

# get proportion of space that overlaps (try with more iterations on your own data)
prop_overlaps <- rarefact_space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "proportional.overlap",
  iterations = 5)

# get minimum convex polygon overlap for each group (non-symmetric)
mcp_overlaps <- rarefact_space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  iterations = 5)

# convert to non-symmetric triangular matrix
rectangular_to_triangular(mcp_overlaps, symmetric = FALSE)
}
```

rarefact_space_size	<i>Estimates rarefacted size of phenotypic spaces</i>
---------------------	---

Description

rarefact_space_size

Usage

```
rarefact_space_size(
  formula,
  data,
  n = NULL,
  replace = FALSE,
  seed = NULL,
  cores = 1,
  pb = TRUE,
  iterations = 30,
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form group ~ dim1 + dim2 where dim1 and dim2 are the dimensions of the phenotype space and group refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
n	Integer vector of length 1 indicating the number of samples to be use for rarefaction (i.e. how many samples per group will be gather at each iteration). Default is the minimum sample size across groups. Integer vector of length 1 indicating the number of samples to be use for rarefaction (i.e. how many samples per group will be gather at each iteration). Default is the minimum sample size across groups.
replace	Logical argument to control if sampling is done with replacement. Default is FALSE.
seed	Integer vector of length 1 setting the seed (see set.seed). If used results should be the same on different runs, so it makes them replicable.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
pb	Logical argument to control if progress bar is shown. Default is TRUE.
iterations	Integer vector of length 1. Controls how the number of times the rarefaction routine is iterated. Default is 30.
...	Additional arguments to be passed to space_size .

Details

The function applies a rarefaction sub-sampling procedure for evaluating pairwise space size differences (internally using [space_size](#)). The size of a phenotypic space might change as a function of number of samples. Hence, ideally, spaces should be compared between groups of similar sample sizes. Rarefaction allows to compare groups of unbalanced sample sizes by randomly re-sampling observations using the same number samples across groups iteratively.

Value

A data frame containing the mean, minimum, maximum and standard deviation of the size difference across iterations for each pair of groups.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[rarefact_space_similarity](#), [space_size_difference](#)

Examples

```
{
# load data
data("example_space")

# get rarefacted MCP space size
# (try with more iterations on your own data)
rarefact_space_size(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mcp")

# mst rarefacted
rarefact_space_size(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mst")
}
```

rarefact_space_size_difference

Calculates rarefacted space size differences

Description

rarefact_space_size_difference

Usage

```
rarefact_space_size_difference(
  formula,
  data,
  n = NULL,
  replace = FALSE,
  seed = NULL,
  cores = 1,
  pb = TRUE,
  iterations = 30,
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form <code>group ~ dim1 + dim2</code> where <code>dim1</code> and <code>dim2</code> are the dimensions of the phenotype space and <code>group</code> refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
n	Integer vector of length 1 indicating the number of samples to be use for rarefaction (i.e. how many samples per group will be gather at each iteration). Default is the minimum sample size across groups.
replace	Logical argument to control if sampling is done with replacement. Default is FALSE.
seed	Integer vector of length 1 setting the seed (see set.seed). If used results should be the same on different runs, so it makes them replicable.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
pb	Logical argument to control if progress bar is shown. Default is TRUE.
iterations	Integer vector of length 1. Controls how the number of times the rarefaction routine is iterated. Default is 30.
...	Additional arguments to be passed to space_size_difference .

Details

The function applies a rarefaction sub-sampling procedure for evaluating pairwise space size differences (internally using [space_size_difference](#)). The size of a phenotypic space might change as a function of number of samples. Hence, ideally, spaces should be compared between groups of similar sample sizes. Rarefaction allows to compare groups of unbalanced sample sizes by randomly re-sampling observations using the same number samples across groups iteratively.

Value

A data frame containing the mean, minimum, maximum and standard deviation of the space size difference across iterations for each pair of groups.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[rarefact_space_similarity](#), [space_size_difference](#)

Examples

```
{
# load data
data("example_space")

# get rarefied size difference using MCP (try with more iterations on your own data)
mcp_size_diff <- rarefact_space_size_difference(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mcp",
  iterations = 5)

# convert to non-symmetric triangular matrix
rectangular_to_triangular(mcp_size_diff, symmetric = FALSE)
}
```

rectangular_to_triangular

Convert rectangular pairwise matrices to triangular matrices

Description

rectangular_to_triangular converts rectangular pairwise matrices as those output by many PhenotypeSpace functions into triangular pairwise matrices.

Usage

```
rectangular_to_triangular(X, distance = TRUE, symmetric = TRUE)
```

Arguments

X	Data frame containing three columns. The first two columns must contain group labels which will appear as rows (1 column) and column names (2 column) in the output triangular matrix. The third column (and fourth column if <code>symmetric = FALSE</code>) must have the numeric values to be included in the output triangular matrix.
distance	Logical argument to control if the input data contains pairwise distances (dissimilarities) or similarities. If <code>TRUE</code> then diagonal values are filled with 0, otherwise they are filled with 1. Note that diagonal values can be set with diag .
symmetric	Logical argument to define if values are duplicated on both off-diagonal triangles (a symmetric triangular matrix, <code>symmetric = TRUE</code> , default) or each triangle is filled with values from different columns (a non-symmetric triangular matrix, <code>symmetric = FALSE</code>). In the latter the upper triangle is filled with the first column and the lower triangle with the second column. In this case, a fourth column with numeric values should be supplied.

Details

The function converts rectangular pairwise matrices as those output by many PhenotypeSpace functions into triangular pairwise matrices. It takes a data frame in which each observation (row) contains the pairwise value and related labels of the 'groups' being compared. The first two columns must contain group labels which will appear as rows (1 column) and column names (2 column) in the output triangular matrix. The third column (and fourth column if `symmetric = FALSE`) must have the numeric values to be included in the output triangular matrix.

Value

A pairwise triangular matrix in which labels from the first group column in 'X' are shown in the columns and labels from the second group are shown in the rows. If `symmetric = FALSE` the same information is shown below and above the diagonal.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[distance_to_rectangular](#), [binary_triangular_matrix](#)

Examples

```
{
# load data
data("example_space")

# get proportion of space that overlaps
prop_overlaps <- space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "proportional.overlap")

# get symmetric triangular matrix
rectangular_to_triangular(prop_overlaps)

# get minimum convex polygon overlap for each group (non-symmetric)
mcp_overlaps <- space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mcp.overlap")

# get a non-symmetric triangular matrix
rectangular_to_triangular(mcp_overlaps, symmetric = FALSE)
}
```

space_similarity

Pairwise similarities of phenotype spaces

Description

space_similarity estimate pairwise similarities of phenotype spaces

Usage

```
space_similarity(
  formula,
  data,
  cores = 1,
  method = "mcp.overlap",
  pb = TRUE,
  outliers = 0.95,
```

```

pairwise.scale = FALSE,
distance.method = "Euclidean",
seed = NULL,
...
)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form <code>group ~ dim1 + dim2</code> where <code>dim1</code> and <code>dim2</code> are the dimensions of the phenotypic space and <code>group</code> refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
method	Character vector of length 1. Controls the method of (di)similarity metric to be compare the phenotypic sub-spaces of two groups at the time. Seven built-in metrics are available which quantify as pairwise sub-space overlap ('similarity') or pairwise distance between bi-dimensional sub-spaces ('dissimilarity'): • <code>density.overlap</code>: proportion of the phenotypic sub-spaces area that overlap, taking into account the irregular densities of the sub-spaces. Two groups that share their higher density areas will be more similar than similar sub-spaces that only share their lower density areas. Two values are supplied as the proportion of the space of A that overlaps B is not necessarily the same as the proportion of B that overlaps A. Similarity metric (higher values means more similar). The minimum sample size (per group) must be 6 observations. • <code>mean.density.overlap</code>: similar to 'density.overlap' but the two values are merged into a single pairwise mean overlap. Similarity metric (higher values means more similar). The minimum sample size (per group) must be 6 observations. • <code>mcp.overlap</code>: proportion of the phenotypic sub-spaces area that overlap, in which areas are calculated as the minimum convex polygon of all observations for each sub-space. Two values are supplied as the proportion of the space of A that overlaps B is not necessarily the same as the proportion of B that overlaps A. Similarity metric (higher values means more similar). The minimum sample size (per group) must be 5 observations. • <code>mean.mcp.overlap</code>: similar to 'mcp.overlap' but the two values are merged into a single pairwise mean overlap. Similarity metric (higher values means more similar). The minimum sample size (per group) must be 5 observations. • <code>proportional.overlap</code>: proportion of the joint area of both sub-spaces that overlaps (overlapped area / total area of both groups). Sub-space areas are calculated as the minimum convex polygon. Similarity metric (higher values means more similar). The minimum sample size (per group) must be 5 observations.

- `distance`: mean euclidean pairwise distance between all observations of the compared sub-spaces. Dissimilarity metric (higher values means less similar). The minimum sample size (per group) must be 1 observation.
- `centroid.distance`: euclidean distance between the centroid of the compared sub-spaces. Dissimilarity metric (higher values means less similar). The minimum sample size (per group) must be 1 observation.
- `probability`: Bayesian probability of observations of one group being classified as belonging to the other group. Similarity metric (higher values means less similar). The minimum sample size (per group) must be higher the number of dimensions. Probabilities are calculated using the function `overlap` from the nicheROVER package. The following values are used internally by `overlap`: `nreps` = 1000, `nprob` = 1000, `kappa` = 0, `Psi` = 0, `nu` = number of predictors + 1. Random draws are taken from the posterior distribution with Normal-Inverse-Wishart (NIW) prior using the function `niw.post`. Take a look at the nicheROVER package for further details on this method.

In addition, machine learning classification models can also be used for quantify dissimilarity as a measured of how discriminable two groups are. These models can use more than two dimensions to represent phenotypic spaces. The following classification models can be used: "AdaBag", "avNNet", "bam", "C5.0", "C5.0Cost", "C5.0Rules", "C5.0Tree", "gam", "gamLoess", "glmnet", "glmStepAIC", "kernelpls", "kknn", "lda", "lda2", "LogitBoost", "msaenet", "multinom", "nnet", "null", "ownn", "parRF", "pcaNNet", "pls", "plsRglm", "pre", "qda", "randomGLM", "rf", "rFerns", "rocc", "rotationForest", "rotationForestCp", "RRF", "RRFglobal", "sda", "simpls", "slda", "smda", "snn", "sparseLDA", "svmLinear2", "svmLinearWeights", "treebag", "widekernelpls" and "wsrf". See <https://topepo.github.io/caret/train-models-by-tag.html> for details on each of these models. Additional arguments can be passed using `...`. Note that some machine learning methods can significantly affect `com`

<code>pb</code>	Logical argument to control if progress bar is shown. Default is TRUE.
<code>outliers</code>	Numeric vector of length 1. A value between 0 and 1 controlling the proportion of outlier observations to be excluded. Outliers are determined as those farthest away from the sub-space centroid. Ignored when using machine learning methods.
<code>pairwise.scale</code>	Logical argument to control if pairwise phenotypic spaces are scaled (i.e. z-transformed) prior to similarity estimation. If so (TRUE) similarities are decoupled from the size of the global phenotypic space. Useful to compare similarities coming from different phenotypic spaces. Default is FALSE. Not available for 'density.overlap', 'mean.density.overlap' or any machine learning model.
<code>distance.method</code>	Character vector of length 1 indicating the method to be used for measuring distances (hence only applicable when distances are calculated). Available distance measures are: "Euclidean" (default), "Manhattan", "supremum", "Canberra", "Wave", "divergence", "Bray", "Soergel", "Podani", "Chord", "Geodesic" and "Whittaker". If a similarity measure is used similarities are converted to distances.

seed	Integer number containing the random number generator (RNG) state for random number generation in order to make results from the machine learning stochastic methods replicable.
...	Additional arguments to be passed to train .

Details

The function quantifies pairwise similarity between phenotypic sub-spaces. The built-in methods quantify similarity as the overlap (similarity, or machine learning based discriminability) or distance (dissimilarity) between group. Machine learning methods implemented in the caret package function [train](#) are available to assess the similarity of spaces as the proportion of observations that are incorrectly classified. In this case group overlaps are the class-wise errors (if available) while the mean overlap is calculated as 1 - model accuracy.

Value

A data frame containing the similarity metric for each pair of groups. If the similarity metric is not symmetric (e.g. the proportional area of A that overlaps B is not necessarily the same as the area of B that overlaps A, see [space_similarity](#)) separated columns are supplied for the two comparisons.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[rarefact_space_similarity](#), [space_size_difference](#)

Examples

```
{
# load data
data("example_space")

# get proportion of space that overlaps
prop_overlaps <- space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "proportional.overlap")

#' # get symmetric triangular matrix
rectangular_to_triangular(prop_overlaps)

# get minimum convex polygon overlap for each group (non-symmetric)
mcp_overlaps <- space_similarity(
```

```

formula = group ~ dimension_1 + dimension_2,
data = example_space,
method = "mcp.overlap")

# convert to non-symmetric triangular matrix
rectangular_to_triangular(mcp_overlaps, symmetric = FALSE)

# check available distance measures
summary(proxy::pr_DB)

# get eculidean distances (default)
area_dist <- space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "distance",
  distance.method = "Euclidean")

# get Canberra distances
area_dist <- space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "distance",
  distance.method = "Canberra")

## using machine learning classification methods

# check if caret package and needed dependencies are available
rlang::check_installed("caret")
rlang::check_installed("randomForest")

# random forest 3 dimension data, using 5 repeats and repeated CV resampling
# extract data subset
sub_data <- example_space[example_space$group %in% c("G1", "G2", "G3"), ]

# set method parameters
ctrl <- caret::trainControl(method = "repeatedcv", repeats = 5)

# get similarities ("overlap")
space_similarity(
  formula = group ~ dimension_1 + dimension_2 + dimension_3,
  data = sub_data,
  method = "rf",
  trControl = ctrl,
  tuneLength = 4,
  seed = 123
)

# Single C5.0 Tree using boot resampling
ctrl <- caret::trainControl(method = "boot")

space_similarity(
  formula = group ~ dimension_1 + dimension_2,
  data = sub_data,

```

```

method = "C5.0Tree",
trControl = ctrl,
tuneLength = 3
)
}

```

space_size

Estimates the size of phenotypic spaces

Description

space_size

Usage

```

space_size(
  formula,
  data,
  cores = 1,
  method = "mcp",
  pb = TRUE,
  outliers = 0.95,
  ...
)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form <code>group ~ dim1 + dim2</code> where <code>dim1</code> and <code>dim2</code> are the dimensions of the phenotype space and <code>group</code> refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
method	Character vector of length 1. Controls the method to be used for quantifying space size. Three metrics are available: • <code>mcp</code>: minimum convex polygon area using the function mcp. The minimum sample size (per group) must be 2 observations. Only works on 2-dimensional spaces. • <code>density</code>: kernel density area using the function kernelUD. The minimum sample size (per group) must be 6 observations. Only works on 2-dimensional spaces.

- **mst**: minimum spanning tree using the function [spantree](#). The minimum sample size (per group) must be 2 observations. This method is expected to be more robust to the influence of outliers. Note that mst is not a actually measuring area but distance between observations. However, it still help to quantify the size of the sub-region in trait space. Any number of dimensions can be used with this method.
 - **ellipse**: Calculate the size of an sub-region assuming an elliptical shape. The axes of the ellipse are estimated from the covariance matrix of the data points in the sub-region. Estimated with the function [niche.size](#) from the package 'nicheROVER'. The minimum sample size (per group) must be 1 observation. Any number of dimensions can be used with this method.
- pb** Logical argument to control if progress bar is shown. Default is TRUE.
- outliers** Numeric vector of length 1. A value between 0 and 1 controlling the proportion of outlier observations to be excluded. Outliers are determined as those farthest away from the sub-space centroid.
- ...** Additional arguments to be passed to [kernelUD](#) for kernel density estimation (when method = 'density').

Details

The function quantifies the size of the phenotypic sub-spaces.

Value

A data frame containing the phenotypic space size for each group.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

[rarefact_space_size](#), [space_size_difference](#), [rarefact_space_size_difference](#)

Examples

```
{
# load data
data("example_space")

# plot data
xs <- tapply(example_space$dimension_1, example_space$group, mean)
ys <- tapply(example_space$dimension_2, example_space$group, mean)
plot(example_space[, c("dimension_1", "dimension_2")],
```

```

col = example_space$color, pch = 20, cex = 1.8)
text(xs, ys, labels = names(xs), cex = 2.5)

# MCP spaces
space_size(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mcp")

# MST
space_size(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mst")
}

```

space_size_difference *Estimates pairwise size differences of phenotypic spaces*

Description

space_size_difference

Usage

```

space_size_difference(
  formula,
  data,
  cores = 1,
  method = "mcp",
  pb = TRUE,
  outliers = 0.95,
  ...
)

```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class). Must follow the form <code>group ~ dim1 + dim2</code> where <code>dim1</code> and <code>dim2</code> are the dimensions of the phenotype space and <code>group</code> refers to the group labels.
data	Data frame containing columns for the dimensions of the phenotypic space (numeric) and a categorical or factor column with group labels.
cores	Numeric vector of length 1. Controls whether parallel computing is applied by specifying the number of cores to be used. Default is 1 (i.e. no parallel computing).
method	Character vector of length 1. Controls the method to be used for quantifying space size. Three metrics are available:

	• <code>mcp</code>: minimum convex polygon area using the function <code>mcp</code>. The minimum sample size (per group) must be 2 observations. Only works on 2-dimensional spaces. • <code>density</code>: kernel density area using the function <code>kernelUD</code>. The minimum sample size (per group) must be 6 observations. Only works on 2-dimensional spaces. • <code>mst</code>: minimum spanning tree using the function <code>spantree</code>. The minimum sample size (per group) must be 5 observations. This method is expected to be more robust to the influence of outliers. Any number of dimensions can be used with this method. • <code>ellipse</code>: Calculate the size of a sub-region assuming an elliptical shape. The axes of the ellipse are estimated from the covariance matrix of the data points in the sub-region. Estimated with the function <code>niche.size</code> from the package 'nicheROVER'. The minimum sample size (per group) must be 1 observation. Any number of dimensions can be used with this method.
<code>pb</code>	Logical argument to control if progress bar is shown. Default is TRUE.
<code>outliers</code>	Numeric vector of length 1. A value between 0 and 1 controlling the proportion of outlier observations to be excluded. Outliers are determined as those farthest away from the sub-space centroid.
<code>...</code>	Additional arguments to be passed to <code>space_size</code> for customizing space size calculation.

Details

The function estimates the pairwise size difference in phenotypic space as a simple subtraction between the sizes of two spaces. As such it can be seen as an additional metric of similarity complementing those found in `space_similarity`.

Value

A data frame containing the space size difference for each pair of groups.

Author(s)

Marcelo Araya-Salas <marcelo.araya@ucr.ac.cr>

References

Araya-Salas, M, K. Odom. & A. Rico-Guevara. 2022, PhenotypeSpace: an R package to quantify and compare phenotypic trait spaces R package version 0.1.0.

See Also

`space_size`, `space_similarity`, `rarefact_space_size_difference`

Examples

```
{
# load data
data("example_space")

# MCP size (try with more iterations on your own data)
mcp_size <- space_size_difference(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mcp")

# MST size
mcp_size <- space_size_difference(
  formula = group ~ dimension_1 + dimension_2,
  data = example_space,
  method = "mst")
}
```

Index

- * **datasets**
 - example_space, 5
- as.dist, 4
- binary_triangular_matrix, 2, 16
- cmdscale, 4
- diag, 15
- dist, 4
- distance_to_rectangular, 3, 3, 4, 8, 16
- example_space, 5
- isoMDS, 4
- kernelUD, 21, 22, 24
- mcp, 21, 24
- niche.size, 22, 24
- niw.post, 18
- overlap, 18
- plot, 7
- plot_space, 6
- points, 7
- rarefact_space_similarity, 9, 12, 14, 19
- rarefact_space_size, 11, 22
- rarefact_space_size_difference, 13, 22, 24
- rectangular_to_triangular, 3, 4, 8, 10, 15
- set.seed, 9, 11, 13
- space_similarity, 9, 10, 16, 19, 24
- space_size, 11, 12, 21, 24
- space_size_difference, 12–14, 19, 22, 23
- spantree, 22, 24
- train, 4, 19