

# Package ‘RRPP’

May 19, 2019

**Date** 2019-05-19

**Type** Package

**Title** Linear Model Evaluation with Randomized Residuals in a  
Permutation Procedure

**Version** 0.4.2

**Author** Michael Collyer, Dean Adams

**Maintainer** Michael Collyer <mlcollyer@gmail.com>

**Description** Linear model calculations are made for many random versions of data. Using residual randomization in a permutation procedure, sums of squares are calculated over many permutations to generate empirical probability distributions for evaluating model effects. This package is described by Collyer & Adams (2018) <doi:10.1111/2041-210X.13029>. Additionally, coefficients, statistics, fitted values, and residuals generated over many permutations can be used for various procedures including pairwise tests, prediction, classification, and model comparison. This package should provide most tools one could need for the analysis of high-dimensional data, especially in ecology and evolutionary biology, but certainly other fields, as well.

**License** GPL (>= 2)

**URL** <https://github.com/mlcollyer/RRPP>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-19 21:40:03 UTC

**R topics documented:**

RRPP-package	3
add.trajectories	3
anova.lm.rrpp	5
classify	6
coef.lm.rrpp	9
fitted.lm.rrpp	10
lm.rrpp	11
manova.update	17
model.comparison	20
motionpaths	22
pairwise	22
PlethMorph	25
plot.lm.rrpp	26
plot.model.comparison	27
plot.predict.lm.rrpp	27
plot.trajectory.analysis	28
predict.lm.rrpp	29
print.anova.lm.rrpp	31
print.classify	31
print.coef.lm.rrpp	32
print.lm.rrpp	32
print.model.comparison	33
print.pairwise	33
print.predict.lm.rrpp	34
print.summary.lm.rrpp	34
print.summary.manova.lm.rrpp	35
print.summary.pairwise	35
print.summary.trajectory.analysis	36
print.trajectory.analysis	36
Pupfish	37
PupfishHeads	37
residuals.lm.rrpp	38
reveal.model.designs	38
rrpp.data.frame	39
summary.anova.lm.rrpp	40
summary.classify	41
summary.coef.lm.rrpp	41
summary.lm.rrpp	42
summary.manova.lm.rrpp	42
summary.model.comparison	43
summary.pairwise	43
summary.predict.lm.rrpp	44
summary.trajectory.analysis	44
trajectory.analysis	45
vec.cor.matrix	48

---

RRPP-package	<i>Linear Model Evaluation with Randomized Residual Permutation Procedures</i>
--------------	--

---

### Description

Functions in this package allow one to evaluate linear models with residual randomization. The name, "RRPP", is an acronym for, "Randomization of Residuals in a Permutation Procedure." Through the various functions in this package, one can use randomization of residuals to generate empirical probability distributions for linear model effects, for high-dimensional data or distance matrices.

An especially useful option of this package is to fit models with either ordinary or generalized least squares estimation (OLS or GLS, respectively), using theoretic covariance matrices. Mixed linear effects can also be evaluated.

### Value

Key functions for this package:

<code>lm.rppp</code>	Fits linear models, using RRPP.
<code>anova.lm.rppp</code>	ANOVA on linear models, using RRPP, plus model comparisons.
<code>coef.lm.rppp</code>	Extract coefficients or perform test on coefficients, using RRPP.
<code>predict.lm.rppp</code>	Predict values from <code>lm.rppp</code> fits and generate bootstrapped confidence intervals.
<code>pairwise</code>	Perform pairwise tests, based on <code>lm.rppp</code> model fits.

### RRPP TOC

RRPP-package

### Author(s)

Michael Collyer and Dean Adams

---

<code>add.trajectories</code>	<i>Plot Function for RRPP</i>
-------------------------------	-------------------------------

---

### Description

Function adds trajectories to a principal component plot

### Usage

```
add.trajectories(TP, traj.pch = 21, traj.col = 1, traj.lty = 1,
  traj.lwd = 1, traj.cex = 1.5, traj.bg = 1, start.bg = 3,
  end.bg = 2)
```

**Arguments**

TP	plot object (from <a href="#">plot.trajectory.analysis</a> )
traj.pch	Plotting "character" for trajectory points. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for pch.
traj.col	The color of trajectory lines. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for col.
traj.lty	Trajectory line type. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for lty.
traj.lwd	Trajectory line width. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for lwd.
traj.cex	Trajectory point character expansion. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for cex.
traj.bg	Trajectory point background. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for bg.
start.bg	Trajectory point background, just the start points. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for bg. Green start points are the default.
end.bg	Trajectory point background, just the end points. Can be a single value or vector of length equal to the number of trajectories. See <a href="#">par</a> and its description for bg. Red end points are the default.

**Details**

The function adds trajectories to a plot made by [plot.trajectory.analysis](#). This function has a restricted set of plot parameters based on the number of trajectories to be added to the plot.

**Author(s)**

Michael Collyer

**References**

- Adams, D. C., and M. M. Cerney. 2007. Quantifying biomechanical motion using Procrustes motion analysis. *J. Biomech.* 40:437-444.
- Adams, D. C., and M. L. Collyer. 2007. The analysis of character divergence along environmental gradients and other covariates. *Evolution* 61:510-515.
- Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. *Evolution* 63:1143-1154.
- Collyer, M. L., and D. C. Adams. 2007. Analysis of two-state multivariate phenotypic change in ecological studies. *Ecology* 88:683-692.
- Collyer, M. L., and D. C. Adams. 2013. Phenotypic trajectory analysis: comparison of shape change patterns in evolution and ecology. *Hystrix* 24: 75-83.
- Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 115:357-365.

**See Also**

[plot.default](#) and [par](#)

---

 anova.lm.rppp

*ANOVA for lm.rppp model fits*


---

**Description**

Computes an analysis of variance (ANOVA) table using distributions of random statistics from [lm.rppp](#). ANOVA can be performed on one model or multiple models. If the latter, the first model is considered a null model for comparison to other models. The ANOVA is functionally similar to a non-parametric likelihood ratio test for all null-full model comparisons. Residuals from the null model will be used to generate random pseudovalues via RRPP for evaluation of subsequent models. The permutation schedule from the null model will be used for random permutations. This function does not correct for improper null models. One must assure that the null model is nested within the other models. Illogical results can be generated if this is not the case.

**Usage**

```
## S3 method for class 'lm.rppp'
anova(object, ..., effect.type = c("F", "cohenf", "SS",
  "MS", "Rsqr"), error = NULL, print.progress = TRUE)
```

**Arguments**

object	Object from <a href="#">lm.rppp</a>
...	Additional <a href="#">lm.rppp</a> model fits or other arguments passed to <code>anova</code> .
effect.type	One of "F", "cohenf", "SS", "MS", "Rsqr" to choose from which distribution of statistics to calculate effect sizes (Z). See <a href="#">lm.rppp</a> .
error	An optional character string to define MS error term for calculation of F values. See <a href="#">lm.rppp</a> for examples.
print.progress	A logical argument if multiple models are used and one wishes to view progress for sums of squares (SS) calculations.

**Author(s)**

Michael Collyer

**Examples**

```
# See examples for lm.rppp to see how anova.lm.rppp works in conjunction
# with other functions

data(Pupfish)
names(Pupfish)
Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas
```

```

# Single-Model ANOVA

# Note: one should increase RRPP iterations but a smaller number is used here for demonstration
# efficiency. Generally, iter = 999 will take less
# than 1s for this example with a modern computer.

fit <- lm.rrpp(coords ~ logSize + Sex*Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 499)
anova(fit)
anova(fit, effect.type = "MS")
anova(fit, effect.type = "Rsq")
anova(fit, effect.type = "cohenf")

# Multi-Model ANOVA (like a Likelihood Ratio Test)
fit.size <- lm.rrpp(coords ~ logSize, SS.type = "I", data = Pupfish,
print.progress = FALSE, iter = 499)
fit.sex <- lm.rrpp(coords ~ logSize + Sex, SS.type = "I", data = Pupfish,
print.progress = FALSE, iter = 499)
fit.pop <- lm.rrpp(coords ~ logSize + Pop, SS.type = "I", data = Pupfish,
print.progress = FALSE, iter = 499)
anova(fit.size, fit.sex, fit.pop) # compares two models to the first

# see lm.rrpp examples for mixed model ANOVA example and how to vary SS type

```

---

classify

*Classification probabilities for lm.rrpp model fits*

---

## Description

Function calculates posterior probabilities for test data, based on factors in a `lm.rrpp` fit and values for continuous covariates.

## Usage

```

classify(fit, test.data = NULL, covariate.values = NULL,
prior = NULL, CV = TRUE, pc.space = 1, inherent.groups = FALSE)

```

## Arguments

<code>fit</code>	A linear model fit using <code>lm.rrpp</code> .
<code>test.data</code>	An optional matrix (or object coercible to a matrix) for classification. If <code>NULL</code> , all observed data are used.
<code>covariate.values</code>	An optional data frame (with one value per covariate) to control covariate values. For example, <code>covariate.values = data.frame(x1 = 2, x2 = 4)</code> . This option is used if estimation at a point on a covariate axis other than the mean value is desired.

<code>prior</code>	An option vector with values between 0 and 1 in the order of groups. If the groups extracted from the <code>lm.rpp</code> fit or their order are not clear, it is wise to use the <code>inherent.groups = TRUE</code> option first. This will display the group information. If NULL, prior probabilities will be based on group size. A choice of "equal" will force equal prior probabilities.
<code>CV</code>	A logical argument to indicate whether a leave-one-out cross-validation (Jack-knife) procedure should be performed. This is only a logical choice if test data are the same data used for the <code>lm.rpp</code> fit. In other cases, the argument will be ignored. The default is <code>CV = TRUE</code> .
<code>pc.space</code>	A value between 0 and 1 to indicate the portion of PC space to use for classification. The default is 1.0 (the entire PC space). Note that this is an upper limit, and the PCs included might (and probably do) comprise less space.
<code>inherent.groups</code>	A logical argument in case one wishes to have the inherent groups in the model fit revealed. If TRUE, no other analysis will be done than to reveal the groups. This argument should always be FALSE to perform a classification analysis.

## Details

This function uses a `lm.rpp` fit, optional covariate values, optional prior probabilities, and an optional cross-validation procedure to estimate posterior probabilities. One can choose the principal component subspace in which to do the classification. For most cases, 100 percent of the PC space is preferred; however, when there is a strong signal - groups have different means and observations within groups are close to means - it might be desirable to ascertain if the signal remains strong in fewer dimensions. The `pc.space` argument can be toggled, and one might wish to run several analyses with different selections, to get a feel for how classification improves with increased data dimensionality.

For high dimensional data, training and testing data are first projected onto the principal components of the appropriate error covariance matrix from the `lm.rpp` fit. (The function will detect whether ordinary least squares or generalized least squares was used.) The model covariance is then re-estimated in the defined amount of the PC space. A ridge regularization step is also employed if it is determined that the covariance matrix is ill-conditioned. In this space, generalized (Mahalanobis) distances are calculated and used to calculate Bayesian posterior probabilities, based on a multivariate normal density function. An option for user-supplied prior probabilities is available, or a user can choose between equal or sample-size corrected prior probabilities.

This function should produce similar results as `lda` for models that can be used in that function, but it will also calculate least squares means if there are covariates in the `lm.rpp` fit. Furthermore, the user can choose specific covariate values for estimation. For example, one could make classifications at small organism size or large size (rather than mean size), if size is a covariate. This is helpful if there are factor-covariate interactions to consider, as classification could differ across the span of a covariate. The function does not find canonical (discriminant) axes, like `lda`, however these can be calculated from `lm.rpp` output. Whereas `lda` only allows a single factor model, `classify` can be used with various model types, and "discriminant" axes would not make as much sense for complex models. Finally, `lda` scales data and model design matrices to find (transformed) discriminant functions for within-group covariances that are forced to be spherical; i.e., the discriminant functions are normalized. This is not a step taken with `classify`, as model design matrices might be more complex than single-factor designs, and obtaining discriminant functions is not the goal. Nevertheless, posterior probabilities will differ slightly between the two functions as a result.

Regardless of variables input, data are projected onto PCs. The purpose of this function is to predict group association, and working in PC space facilitates this objective.

This is a new function and not all limits and scenarios have been tested before its release. Please report any issues or limitations or strange results to the maintainer. (The function will likely be updated frequently until all kinks are worked out.)

### Value

An object of class `classify` is a list containing the following

<code>means</code>	LS means (PC scores) for inherent groups.
<code>group</code>	A factor derived for groups.
<code>group.n</code>	Sample size for each group.
<code>test.data</code>	PC scores for the test data used.
<code>Mah.dist.sq</code>	Generalized Mahalanobis squared distance from test data to group means.
<code>prior</code>	Prior probabilities.
<code>posterior</code>	Posterior probabilities.
<code>class</code>	Classification based on highest posterior probability
<code>class.matrix</code>	An indicator matrix for classifications.

### Author(s)

Michael Collyer

### Examples

```
# Using the Pupfish data (see lm.rpp help for more detail)

data(Pupfish)
Pupfish$logSize <- log(Pupfish$CS)
fit <- lm.rpp(coords ~ logSize + Sex * Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 0)

classify(fit, inherent.groups = TRUE) # see groups available
class1 <- classify(fit, CV = TRUE)
summary(class1)

# Perfect classification

table(interaction(Pupfish$Sex, Pupfish$Pop), class1$class)

class2 <- classify(fit, CV = TRUE, pc.space = 0.6)
summary(class2)
table(interaction(Pupfish$Sex, Pupfish$Pop), class2$class)

# Allow for heterogenous slopes in the model
fit2 <- lm.rpp(coords ~ logSize * Sex * Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 0)
```



```

# Classification for small-sized fish (previous example for mean-sized fish)

class3 <- classify(fit2, CV = TRUE,
covariate.values = data.frame(logSize = 4))
summary(class3)

# Classification for large-sized fish

class4 <- classify(fit2, CV = TRUE,
covariate.values = data.frame(logSize = 4.5))
summary(class4)

```

---

coef.lm.rppp

*coef for lm.rppp model fits*


---

## Description

Computes ordinary or generalized least squares coefficients over the permutations of an [lm.rppp](#) model fit with predefined random permutations. For each coefficient vector, the Euclidean distance is calculated as an estimate of the amount of change in Y, the  $n \times p$  matrix of dependent variables; larger distances mean more change in location in the data space associated with a one unit change in the model design, for the parameter described. Random coefficients are based on either RRPP or FRPP, as defined by the [lm.rppp](#) model fit. If RRPP is used, all distributions of coefficient vector distances are based on appropriate null models as defined by SS type.

This function can be used to test the specific coefficients of an [lm.rppp](#) fit. The test statistics are the distances (d), which are also standardized (Z-scores). The Z-scores might be easier to compare, as the expected values for random distances can vary among coefficient vectors (Adams and Collyer 2016).

## Usage

```

## S3 method for class 'lm.rppp'
coef(object, test = FALSE, confidence = 0.95, ...)

```

## Arguments

object	Object from <a href="#">lm.rppp</a>
test	Logical argument that if TRUE, performs hypothesis tests (Null hypothesis is vector distance = 0) for the observed coefficients. If FALSE, only the observed coefficients are returned.
confidence	The desired confidence limit to print with a table of summary statistics, if test = TRUE. Because distances are directionless, confidence limits are one-tailed.
...	Other arguments (currently none)

## Author(s)

Michael Collyer

**Examples**

```
# See examples for lm.rpp to see how anova.lm.rpp works in conjunction
# with other functions

data(Pupfish)
names(Pupfish)
Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas

fit <- lm.rpp(coords ~ logSize + Sex*Pop, SS.type = "I", data = Pupfish)

coef(fit)
coef(fit, test = TRUE, confidence = 0.99)
```

---

fitted.lm.rpp	<i>Extract fitted values</i>
---------------	------------------------------

---

**Description**

Extract fitted values

**Usage**

```
## S3 method for class 'lm.rpp'
fitted(object, weighted = TRUE, ...)
```

**Arguments**

object	plot object (from <a href="#">lm.rpp</a> )
weighted	A logical argument to return weighted or unweighted residuals.
...	Arguments passed to other functions

**Author(s)**

Michael Collyer

**Examples**

```
# See examples for lm.rpp
```

---

lm.rppp	<i>Linear Model Evaluation with a Randomized Residual Permutation Procedure</i>
---------	---

---

**Description**

Function performs a linear model fit over many random permutations of data, using a randomized residual permutation procedure.

**Usage**

```
lm.rppp(f1, iter = 999, seed = NULL, int.first = FALSE,
        RRPP = TRUE, SS.type = c("I", "II", "III"), data = NULL,
        Cov = NULL, print.progress = TRUE, Parallel = FALSE, ...)
```

**Arguments**

f1	A formula for the linear model (e.g., $y \sim x_1 + x_2$ ). Can also be a linear model fit from <code>lm</code> .
iter	Number of iterations for significance testing
seed	An optional argument for setting the seed for random permutations of the resampling procedure. If left NULL (the default), the exact same P-values will be found for repeated runs of the analysis (with the same number of iterations). If seed = "random", a random seed will be used, and P-values will vary. One can also specify an integer for specific seed values, which might be of interest for advanced users.
int.first	A logical value to indicate if interactions of first main effects should precede subsequent main effects
RRPP	A logical value indicating whether residual randomization should be used for significance testing
SS.type	A choice between type I (sequential), type II (hierarchical), or type III (marginal) sums of squares and cross-products computations.
data	A data frame for the function environment, see <a href="#">rppp.data.frame</a>
Cov	An optional argument for including a covariance matrix to address the non-independence of error in the estimation of coefficients (via GLS).
print.progress	A logical value to indicate whether a progress bar should be printed to the screen. This is helpful for long-running analyses.
Parallel	A logical value to indicate whether parallel processing should be used. If TRUE, this argument invokes forking of processor cores, using the <code>parallel</code> library. This option is only available to unix systems and should only be used for rather long analyses (that would normally take over 10 seconds on a single core). Currently, parallel processing is performed on all but one core with no option to change the number of cores. Systems with Windows platforms will automatically default to a single-core application of this function.
...	Arguments typically used in <code>lm</code> , such as weights or offset, passed on to <code>rppp.fit</code> for estimation of coefficients.

## Details

The function fits a linear model using ordinary least squares (OLS) or generalized least squares (GLS) estimation of coefficients over any number of random permutations of the data. A permutation procedure that randomizes vectors of residuals is employed. This procedure can randomize two types of residuals: residuals from null models or residuals from an intercept model. The latter is the same as randomizing full values, and is referred to as a full randomization permutation procedure (FRPP); the former uses the residuals from null models, which are defined by the type of sums of squares and cross-products (SSCP) sought in an analysis of variance (ANOVA), and is referred to as a randomized residual permutation procedure (RRPP). Types I, II, and III SSCPs are supported.

Users define the SSCP type, the permutation procedure type, whether a covariance matrix is included (GLS estimation), and a few arguments related to computations. Analytical results comprise observed linear model results (coefficients, fitted values, residuals, etc.), random sums of squares (SS) across permutation iterations, and other parameters for performing ANOVA and other hypothesis tests, using empirically-derived probability distributions.

lm.rrpp emphasizes estimation of standard deviates of observed statistics as effect sizes from distributions of random outcomes. When performing ANOVA, using the `anova` function, the effect type (statistic choice) can be varied. See `anova.lm.rrpp` for more details. Please recognize that the type of SS must be chosen prior to running `lm.rrpp` and not when applying `anova` to the `lm.rrpp` fit, as design matrices for the linear model must be created first. Therefore, `SS.type` is an argument for `lm.rrpp` and `effect.type` is an argument for `anova.lm.rrpp`. If MANOVA statistics are preferred, eigenvalues can be added with `manova.update` and statistics summarized with `summary.manova.lm.rrpp`. See `manova.update` for examples.

The `coef.lm.rrpp` function can be used to test the specific coefficients of an `lm.rrpp` fit. The test statistics are the distances (d), which are also standardized (Z-scores). The Z-scores might be easier to compare, as the expected values for random distances can vary among coefficient vectors (Adams and Collyer 2016).

### ANOVA vs. MANOVA:

Two SSCP matrices are calculated for each linear model effect, for every random permutation: R (Residuals or Random effects) and H, the difference between SSCPs for "full" and "reduced" models. (Full models contain and reduced models lack the effect tested; SSCPs are hypothesized to be the same under a null hypothesis, if there is no effect. The difference, H, would have a trace of 0 if the null hypothesis were true.) In RRPP, ANOVA and MANOVA correspond to two different ways to calculate statistics from R and H matrices.

ANOVA statistics are those that find the trace of R and H SSCP matrices before calculating subsequent statistics, including sums of squares (SS), mean squares (MS), and F-values. These statistics can be calculated with univariate data and provide univariate-like statistics for multivariate data. These statistics are dispersion measures only (covariances among variables do not contribute) and are the same as "distance-based" stats proposed by Goodall (1991) and Anderson (2001). MANOVA stats require multivariate data and are implicitly affected by variable covariances. For MANOVA, the inverse of R times H ( $\text{invR.H}$ ) is first calculated for each effect, then eigenanalysis is performed on these matrix products. Multivariate statistics are calculated from the positive, real eigenvalues. In general, inferential conclusions will be similar with either approach, but effect sizes might differ.

ANOVA tables are generated by `anova.lm.rrpp` on `lm.rrpp` fits and MANOVA tables are generated by `summary.manova.lm.rrpp`, after running `manova.update` on `lm.rrpp` fits.

Currently, mixed model effects are only possible with \$ANOVA statistics, not \$MANOVA. More detail is found in the vignette, ANOVA versus MANOVA.

**Notes for RRPP 0.3.1 and subsequent versions:**

F-values via RRPP are calculated with residual SS (RSS) found uniquely for any model terms, as per Anderson and ter Braak (2003). This method uses the random pseudo-data generated by each term's null (reduced) model, meaning RSS can vary across terms. Previous versions used an intercept-only model for generating random pseudo-data. This generally has appropriate type I error rates but can have elevated type I error rates if the observed RSS is small relative to total SS. Allowing term by term unique RSS alleviates this concern.

**Value**

An object of class `lm.rppp` is a list containing the following

<code>call</code>	The matched call.
<code>LM</code>	Linear Model objects, including data (Y), coefficients, design matrix (X), sample size (n), number of dependent variables (p), dimension of data space (p.prime), QR decomposition of the design matrix, fitted values, residuals, weights, offset, model terms, model frame (data), random coefficients (through permutations), random vector distances for coefficients (through permutations), whether OLS or GLS was performed, and the centroid for OLS and/or GLS methods.
<code>ANOVA</code>	Analysis of variance objects, including the SS type, random SS outcomes, random MS outcomes, random R-squared outcomes, random F outcomes, random Cohen's f-squared outcomes, P-values based on random F outcomes, effect sizes for random outcomes, sample size (n), number of variables (p), and degrees of freedom for model terms (df). These objects are used to construct ANOVA tables.
<code>PermInfo</code>	Permutation procedure information, including the number of permutations (perms), The method of residual randomization (perm.method), and each permutation's sampling frame (perm.schedule), which is a list of reordered sequences of 1:n, for how residuals were randomized.

**Author(s)**

Michael Collyer

**References**

- Anderson MJ. 2001. A new method for non-parametric multivariate analysis of variance. *Austral Ecology* 26: 32-46.
- Anderson MJ. and C.J.F. ter Braak. 2003. Permutation tests for multi-factorial analysis of variance. *Journal of Statistical Computation and Simulation* 73: 85-113.
- Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 115:357-365.
- Adams, D.C. and M.L. Collyer. 2016. On the comparison of the strength of morphological integration across morphometric datasets. *Evolution*. 70:2623-2631.

Adams, D.C and M.L. Collyer. 2018. Multivariate phylogenetic anova: group-clade aggregation, biological challenges, and a refined permutation procedure. *Evolution*. 72:1204-1215.

### See Also

procD.lm and procD.pgls within geomorph; [lm](#) for more on linear model fits.

### Examples

```
# Examples use geometric morphometric data
# See the package, geomorph, for details about obtaining such data

data("PupfishHeads")
names(PupfishHeads)

# Head Size Analysis (Univariate)-----

# Note: lm.rrpp works best if one avoids functions within formulas
# Thus,

PupfishHeads$logHeadSize <- log(PupfishHeads$headSize)
names(PupfishHeads)

# Note: one should increase RRPP iterations but a smaller number is used here for demonstration
# efficiency. Generally, iter = 999 will take less
# than 1s for this example with a modern computer.

fit <- lm.rrpp(logHeadSize ~ sex + locality/year, SS.type = "I",
data = PupfishHeads, print.progress = FALSE, iter = 199)
summary(fit)
anova(fit, effect.type = "F") # Maybe not most appropriate
anova(fit, effect.type = "Rsq") # Change effect type, but still not most appropriate

# Mixed-model approach (most appropriate, as year sampled is a random effect:

anova(fit, effect.type = "F", error = c("Residuals", "locality:year", "Residuals"))

# Change to Type III SS

fit <- lm.rrpp(logHeadSize ~ sex + locality/year, SS.type = "III",
data = PupfishHeads, print.progress = FALSE, iter = 199)
summary(fit)
anova(fit, effect.type = "F", error = c("Residuals", "locality:year", "Residuals"))

# Coefficients Test

coef(fit, test = TRUE)

# Predictions (holding alternative effects constant)

sizeDF <- data.frame(sex = c("Female", "Male"))
rownames(sizeDF) <- c("Female", "Male")
```

```

sizePreds <- predict(fit, sizeDF)
summary(sizePreds)
plot(sizePreds)

# Diagnostics plots of residuals

plot(fit)

# Body Shape Analysis (Multivariate)-----

data(Pupfish)
names(Pupfish)

# Note:

dim(Pupfish$coords) # highly multivariate!

Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas
names(Pupfish)

# Note: one should increase RRPP iterations but they are not used at all
# here for a fast example. Generally, iter = 999 will take less
# than 1s for this example with a modern computer.

fit <- lm.rpp(coords ~ logSize + Sex*Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 0)
summary(fit, formula = FALSE)
anova(fit)
coef(fit, test = TRUE)

# Predictions (holding alternative effects constant)

shapeDF <- expand.grid(Sex = levels(Pupfish$Sex), Pop = levels(Pupfish$Pop))
rownames(shapeDF) <- paste(shapeDF$Sex, shapeDF$Pop, sep = ".")
shapeDF

shapePreds <- predict(fit, shapeDF)
summary(shapePreds)
summary(shapePreds, PC = TRUE)

# Plot prediction

plot(shapePreds, PC = TRUE)
plot(shapePreds, PC = TRUE, ellipse = TRUE)

# Diagnostics plots of residuals

plot(fit)

# PC-plot of fitted values

groups <- interaction(Pupfish$Sex, Pupfish$Pop)
plot(fit, type = "PC", pch = 19, col = as.numeric(groups))

```





```

anova(fitOLSm)
anova(fitGLSm)

plot(predict(fitOLSm, sizeDF), PC= TRUE) # Correlated error
plot(predict(fitGLSm, sizeDF), PC= TRUE) # Independent error

```

---

manova.update

*MANOVA update for lm.rpp model fits*


---

## Description

Function updates a `lm.rpp` fit to add `$MANOVA`, which like `$ANOVA`, provides statistics or matrices typically associated with multivariate analysis of variance (MANOVA).

MANOVA statistics or sums of squares and cross-products (SSCP) matrices are calculated over the random permutations of a `lm.rpp` fit. SSCP matrices are computed as the inverse of  $R$  time  $H$  ( $\text{invR.H}$ ), where  $R$  is a SSCP for the residuals or random effects and  $H$  is the difference between SSCP matrices of full and reduced models (see below). If the argument, `verbose`, is `TRUE`,  $\text{invR.H}$  matrices will be returned for every model effect and the entire model, for every permutation. If `verbose = FALSE`, eigen-analysis is performed on the  $\text{invR.H}$  matrices and returned, allowing MANOVA statistics to be calculated, including Roy's maximum root (eigen value), Pillai trace, Hotelling-Lawley trace, and Wilks lambda (via [summary.manova.lm.rpp](#)).

The `manova.update` to add `$MANOVA` to `lm.rpp` fits requires more computation time than the `$ANOVA` statistics that are computed automatically in `lm.rpp`. Generally, the same inferential conclusions will be found with either approach, when observations outnumber response variables. For high-dimensional data (more variables than observations) data are projected into a Euclidean space of appropriate dimensions (rank of residual covariance matrix). One can vary the tolerance for eigenvalue decay or specify the number of PCs, if a smaller set of PCs than the maximum is desired. This is advised if there is strong correlation among variables (the data space could be simplified to fewer dimensions), as spurious results are possible. When observations outnumber variables, projection of data onto PCs yields the same results as the original variables. Because distributions of MANOVA stats can be generated from the random permutations, there is no need to approximate F-values, like with parametric MANOVA. By restricting analysis to the real, positive eigen values calculated, all statistics can be calculated (but Wilks lambda, as a product but not a trace, might be less reliable as variable number approaches the number of observations).

### ANOVA vs. MANOVA:

Two SSCP matrices are calculated for each linear model effect, for every random permutation:  $R$  (Residuals or Random effects) and  $H$ , the difference between SSCPs for "full" and "reduced" models. (Full models contain and reduced models lack the effect tested; SSCPs are hypothesized to be the same under a null hypothesis, if there is no effect. The difference,  $H$ , would have a trace of 0 if the null hypothesis were true.) In RRPP, ANOVA and MANOVA correspond to two different ways to calculate statistics from  $R$  and  $H$  matrices.

ANOVA statistics are those that find the trace of  $R$  and  $H$  SSCP matrices before calculating subsequent statistics, including sums of squares (SS), mean squares (MS), and F-values. These statistics can be calculated with univariate data and provide univariate-like statistics for multivariate data. These statistics are dispersion measures only (covariances among variables do not contribute)

and are the same as "distance-based" stats proposed by Goodall (1991) and Anderson (2001). MANOVA stats require multivariate data and are implicitly affected by variable covariances. For MANOVA, the inverse of R times H (invR.H) is first calculated for each effect, then eigenanalysis is performed on these matrix products. Multivariate statistics are calculated from the positive, real eigenvalues. In general, inferential conclusions will be similar with either approach, but effect sizes might differ.

ANOVA tables are generated by `anova.lm.rpp` on `lm.rpp` fits and MANOVA tables are generated by `summary.manova.lm.rpp`, after running `manova.update` on `lm.rpp` fits.

Currently, mixed model effects are only possible with `$ANOVA` statistics, not `$MANOVA`.

More detail is found in the vignette, ANOVA versus MANOVA.

### Usage

```
manova.update(fit, error = NULL, tol = 0.01, PC.no = NULL,
  print.progress = TRUE, verbose = FALSE)
```

### Arguments

<code>fit</code>	Linear model fit from <code>lm.rpp</code>
<code>error</code>	An optional character string to define R matrices used to calculate invR.H. (Currently only Residuals can be used and this argument defaults to NULL. Future versions will update this argument.)
<code>tol</code>	A tolerance value for culling data dimensions to prevent spurious results. The distribution of eigenvalues for the data will be examined and if the decay becomes less than the tolerance, the data will be truncated to principal components ahead of this point. This will possibly prevent spurious results calculated from eigenvalues near 0. If <code>tol = 0</code> , all possible PC axes are used, which is likely not a problem if observations outnumber variables.
<code>PC.no</code>	A value that, if not NULL, overrides the tolerance argument, and forces a desired number of data PCs to use for analysis. If a value larger than the possible number of PCs is chosen, the full compliment of PCs (the full data space) will be used.
<code>print.progress</code>	A logical value to indicate whether a progress bar should be printed to the screen. This is helpful for long-running analyses.
<code>verbose</code>	A logical value to indicate whether invR.H matrices should be returned (if TRUE) or just their eigen values (if FALSE).

### Value

An object of class `lm.rpp` is updated to include class `manova.lm.rpp`, and the object, `$MANOVA`, which includes

<code>eiges</code>	The eigenvalues of invR.H, if <code>verbose = FALSE</code> .
<code>invR.H</code>	The inverse of the residuals SSCP times the H SSCP, if <code>verbose = TRUE</code>
<code>e.rank</code>	Rank of the error (residuals) covariance matrix.

### Author(s)

Michael Collyer

## References

Goodall, C.R. 1991. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B* 53:285-339.

Anderson MJ. 2001. A new method for non-parametric multivariate analysis of variance. *Austral Ecology* 26: 32-46.

## Examples

```
# Body Shape Analysis (Multivariate) -----
data(Pupfish)

# Although not recommended as a practice, this example will use only
# three principal components of body shape for demonstration. A larger
# number of random permutations should also be used.

Pupfish$shape <- prcomp(Pupfish$coords)$x[, 1:3]

Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas

fit <- lm.rpp(shape ~ logSize + Sex, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 499)
summary(fit, formula = FALSE)
anova(fit) # ANOVA table

# MANOVA

fit.m <- manova.update(fit, print.progress = FALSE, tol = 0.001)
summary(fit.m, test = "Roy")
summary(fit.m, test = "Pillai")

fit.m$MANOVA$eigs$logSize[1:3] # eigenvalues first three iterations
fit.m$MANOVA$eigs$Sex[1:3] # eigenvalues first three iterations

fit.m <- manova.update(fit, verbose = TRUE, print.progress = FALSE)
summary(fit.m, test = "Roy")
summary(fit.m, test = "Pillai")

fit.m$MANOVA$invR.H$logSize[1:3] # invR.H first three iterations
fit.m$MANOVA$invR.H$Sex[1:3] # invR.H first three iterations

# Distributions of test statistics

summ.roy <- summary(fit.m, test = "Roy")
dens <- apply(summ.roy$rand.stats, 2, density)
par(mfcol = c(1, length(dens)))
for(i in 1:length(dens)) {
  plot(dens[[i]], xlab = "Roy max root", ylab = "Density",
type = "l", main = names(dens)[[i]])
  abline(v = summ.roy$rand.stats[1, i], col = "red")
}
```

```

}
par(mfcol = c(1,1))

```

---

model.comparison	<i>Model Comparisons, in terms of the log-likelihood or covariance trace</i>
------------------	--

---

### Description

Function calculates either log-likelihoods or traces of covariance matrices for comparison with respect to parameter penalties.

### Usage

```

model.comparison(..., type = c("cov.trace", "logLik"), tol = NULL,
  pc.no = NULL)

```

### Arguments

...	Any number of <code>lm.rpp</code> class objects for model fits to be compared.
type	An argument to choose between log-likelihood or covariance trace results
tol	If type = <code>logLik</code> , tol is a tolerance value between 0 and 1, indicating the magnitude below which components should be omitted (if standard deviations of components are less than the eigenvalue of the first component times the tolerance), for calculating the log-likelihood.
pc.no	If type = <code>logLik</code> , an optional value to indicate the number of principal components (maximum rank) to use for calculating the log-likelihood.

### Details

The function calculates either log-likelihoods or traces of (residual) covariance matrices, plus parameter penalties, to assist in comparative model evaluation or selection. Because high-dimensional data often produce singular or ill-conditioned residual covariance matrices, this function does one of two things: 1) uses the trace of a covariance matrix rather than its determinant; or 2) provides a ridge-regularization (Warton, 2008) of the covariance matrix, only if it is determined that it is ill-conditioned. Regardless of implementation, covariance matrices are projected into a principal component (PC) space of appropriate dimensions.

The parameter penalty is based on that proposed by Bedrick and Tsai (1994), equal to  $2(pk + p(p + 1)/2)$ , where  $p$  is the appropriate dimension (not number of variables) of the covariance matrix. The parameter,  $k$ , is the rank of the model design matrix.

In the case that "logLik" is chosen for the argument, type, AIC scores are calculated. These scores may not perfectly match other packages or software that calculate AIC for multivariate data, if ridge regularization was used (and if other packages require  $p =$  the number of data variables). When choosing logLik as the type of comparison, it might be a good idea to adjust the tolerance or number of data principal components. The default (NULL) values will use all data dimensions to calculate log-likelihoods, which might cause problems if the number of variables exceeds the number of

observations (producing singular residual covariance matrices). However, one should not reduce data dimensions haphazardly, as this can lead to poor estimates of log-likelihood. Furthermore, using the tolerance argument could result in different numbers of principal components used for each model to calculate log-likelihoods, which might be a concern for comparing models. If both `tol` and `pc.no` arguments are used, the solution will use the fewest PCs produced by either argument. Because the trace of a covariance matrix is not sensitive to matrix singularity, no PC adjustment is used for the `cov.trace` argument.

Users can construct their own tables from the results but this function does not attempt to summarize results, as interpreting results requires some arbitrary decisions. The `anova` function explicitly tests multiple models and can be used for nested model comparisons.

Results can also be plotted using the generic `plot` function.

Caution: For models with GLS estimation, the number of parameters used to estimate the covariance matrix is not taken into consideration. A generalized information criterion is currently in development.

### Value

An object of class `model.comparison` is a data frame with either log-likelihoods or covariance traces, plus parameter penalties. AIC scores might be include, if applicable

### Author(s)

Michael Collyer

### References

Bedrick, E.J., and C.L. Tsai. 1994. Model selection for multivariate regression in small samples. *Biometrics*, 226-231.

Warton, D.I., 2008. Penalized normal likelihood and ridge regularization of correlation and covariance matrices. *Journal of the American Statistical Association*. 103: 340-349.

### Examples

```
data(Pupfish)
Pupfish$logSize <- log(Pupfish$CS)
fit1 <- lm.rppp(coords ~ logSize, data = Pupfish, iter = 0, print.progress = FALSE)
fit2 <- lm.rppp(coords ~ Pop, data = Pupfish, iter = 0, print.progress = FALSE)
fit3 <- lm.rppp(coords ~ Sex, data = Pupfish, iter = 0, print.progress = FALSE)
fit4 <- lm.rppp(coords ~ logSize + Sex, data = Pupfish, iter = 0, print.progress = FALSE)
fit5 <- lm.rppp(coords ~ logSize + Pop, data = Pupfish, iter = 0, print.progress = FALSE)
fit6 <- lm.rppp(coords ~ logSize + Sex * Pop, data = Pupfish, iter = 0, print.progress = FALSE)

modComp1 <- model.comparison(fit1, fit2, fit3, fit4, fit5, fit6, type = "cov.trace")
modComp2 <- model.comparison(fit1, fit2, fit3, fit4, fit5, fit6, type = "logLik", tol = 0.01)

summary(modComp1)
summary(modComp2)
```

```
par(mfcol = c(1,2))
plot(modComp1)
plot(modComp2)
```

---

motionpaths	<i>Simulated motion paths</i>
-------------	-------------------------------

---

### Description

Simulated motion paths

### Author(s)

Dean Adams

### References

Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. *Evolution* 63:1143-1154.

---

pairwise	<i>Pairwise comparisons of lm.rpp fits</i>
----------	--

---

### Description

Function generates distributions of pairwise statistics for a `lm.rpp` fit and returns important statistics for hypothesis tests.

### Usage

```
pairwise(fit, fit.null = NULL, groups, covariate = NULL,
         print.progress = FALSE)
```

### Arguments

fit	A linear model fit using <code>lm.rpp</code> .
fit.null	An alternative linear model fit to use as a null model for RRPP, if the null model of the fit is not desired. Note, for FRPP this argument should remain NULL and FRPP must be established in the <code>lm.rpp</code> fit (RRPP = FALSE). If the null model is uncertain, using <code>reveal.model.designs</code> will help elucidate the inherent null model used.
groups	A factor or vector that is coercible into a factor, describing the levels of the groups for which to find LS means or slopes. Normally this factor would be part of the model fit, but it is not necessary for that to be the case in order to obtain results.

<code>covariate</code>	A numeric vector for which to calculate slopes for comparison. If NULL, LS means will be calculated instead of slopes. Normally this variable would be part of the model fit, but it is not necessary for that to be the case in order to obtain results.
<code>print.progress</code>	If a null model fit is provided, a logical value to indicate whether analytical results progress should be printed on screen. Unless large data sets are analyzed, this argument is probably not helpful.

## Details

Based on an `lm.rpp` fit, this function will find fitted values over all permutations and based on a grouping factor, calculate either least squares (LS) means or slopes, and pairwise statistics among them. Pairwise statistics have two flavors: distances and vector correlations (or angles). The distance statistics calculate either the length of vectors between LS mean vectors or the absolute difference between slope vector lengths. The vector correlations are the inner product of vectors that have been transformed to unit length. The arccosine (`acos`) of this value is the angle between vectors, which can be expressed in radians or degrees, and is used as a test statistic (with the null hypothesis that vectors are parallel; angle = 0). Over all permutations, these values can be calculated to generate random distributions using the null model. The null model is defined via `lm.rpp`, but one can also use an alternative null model as an optional argument. In this case, residual randomization in the permutation procedure (RRPP) will be performed using the alternative null model to generate fitted values. If full randomization of values (FRPP) is preferred, it must be established in the `lm.rpp` fit and an alternative model should not be chosen.

Observed statistics, effect sizes, P-values, and one-tailed confidence limits based on the confidence requested will be summarized with the `summary.pairwise` function. The `summary.pairwise` function will allow one to select between distance or vector correlation tests, whether angles are measured in radians or degrees, and the level of confidence for the test. Confidence limits are inherently one-tailed as the statistics are similar to absolute values. For example, a distance is analogous to an absolute difference. Therefore, the one-tailed confidence limits are more akin to two-tailed hypothesis tests. (A comparable example is to use the absolute value of a t-statistic, in which case the distribution has a lower bound of 0.) If rather than comparing the LS means or slopes, one wishes to compare the dispersion of residuals among groups, given the model, an option for comparing variances is also available. Variance degrees of freedom equal `n`, the group size, rather than `n-1`, as the purpose is to compare mean dispersion in the sample. (Additionally, tests with one subject in a group are possible, or at least not a hindrance to the analysis.)

If data are univariate, `test.type = 'cor'` should not be chosen because the vector correlation between univariate vectors is always 1. Rather, `cor.type = 'dist'` will return the absolute difference between slopes or between means. Please note that this function will generate results if `test.type = 'cor'` for univariate data, but the results will not make much sense.

## Value

An object of class `pairwise` is a list containing the following

<code>LS.means</code>	LS means for groups, across permutations.
<code>slopes</code>	Slopes for groups, across permutations.
<code>means.dist</code>	Pairwise distances between means, across permutations.
<code>means.vec.cor</code>	Pairwise vector correlations between means, across permutations.

slopes.lengths Slope lengths, by group, across permutations.  
slopes.dist Pairwise distances between slope lengths, across permutations.  
slopes.vec.cor Pairwise vector correlations between slope vectors, across permutations.  
n Sample size  
p Data dimensions; i.e., variable number  
PermInfo Information for random permutations, passed on from `lm.rpp` fit and possibly modified if an alternative null model was used.

**Author(s)**

Michael Collyer

**References**

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 115:357-365.  
Adams, D.C and M.L. Collyer. 2018. Multivariate phylogenetic anova: group-clade aggregation, biological challenges, and a refined permutation procedure. *Evolution*. In press.

**See Also**

`advanced.procD.lm` within `geomorph`; [lm.rpp](#) for model fits

**Examples**

```
# Examples use geometric morphometric data on pupfishes
# See the package, geomorph, for details about obtaining such data

# Body Shape Analysis (Multivariate)-----

data("Pupfish")

# Note:

dim(Pupfish$coords) # highly multivariate!

Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas

# Note: one should use all dimensions of the data but with this example, there are many
# Thus, only three principal components will be used for demonstration purposes.

Pupfish$Y <- prcomp(Pupfish$coords)$x[, 1:3]

## Pairwise comparisons of LS means

# Note: one should increase RRPP iterations but a smaller number is used here for demonstration
# efficiency. Generally, iter = 999 will take less
# than 1s for these examples with a modern computer.
```



```

fit1 <- lm.rppp(Y ~ logSize + Sex * Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 499)
summary(fit1, formula = FALSE)
anova(fit1)

pup.group <- interaction(Pupfish$Sex, Pupfish$Pop)
pup.group
PW1 <- pairwise(fit1, groups = pup.group)
PW1
summary(PW1, confidence = 0.95, test.type = "dist") # distances between means
summary(PW1, confidence = 0.95, test.type = "dist", stat.table = FALSE)
summary(PW1, confidence = 0.95, test.type = "VC",
angle.type = "deg") # correlation between mean vectors (angles in degrees)

# Can also compare the dispersion around means

summary(PW1, confidence = 0.95, test.type = "var")

## Pairwise comparisons of slopes

fit2 <- lm.rppp(Y ~ logSize * Sex * Pop, SS.type = "I",
data = Pupfish, print.progress = FALSE, iter = 199)
summary(fit2, formula = FALSE)
anova(fit1, fit2)

# Using a null fit that excludes all factor-covariate interactions, not just the last one

PW2 <- pairwise(fit2, fit.null = fit1, groups = pup.group,
covariate = Pupfish$logSize, print.progress = FALSE)
PW2
summary(PW2, confidence = 0.95, test.type = "dist") # distances between slope vector lengths
summary(PW2, confidence = 0.95, test.type = "dist", stat.table = FALSE)
summary(PW2, confidence = 0.95, test.type = "VC",
angle.type = "deg") # correlation between slope vectors (and angles)

# Can also compare the dispersion around group slopes

summary(PW2, confidence = 0.95, test.type = "var")

```

## Description

Data for 37 species of plethodontid salamanders. Variables include snout to vent length (SVL) as species size, tail length, head length, snout to eye length, body width, forelimb length, and hind limb length, all measured in mm. A grouping variable is also included for functional guild size. The data set also includes a phylogenetic covariance matrix based on a Brownian model of evolution, to assist in generalized least squares (GLS) estimation.

**Details**

The covariance matrix was estimated with the `vcv.phylo` function of the R package, `ape`, based on the tree described in Adams and Collyer (2018).

**Author(s)**

Michael Collyer and Dean Adams

**References**

Adams, D.C and Collyer, M.L. 2018. Multivariate phylogenetic anova: group-clade aggregation, biological challenges, and a refined permutation procedure. *Evolution*. In press.

---

plot.lm.rrpp

*Plot Function for RRPP*

---

**Description**

Plot Function for RRPP

**Usage**

```
## S3 method for class 'lm.rrpp'
plot(x, type = c("diagnostics", "regression", "PC"),
     predictor = NULL, reg.type = c("PredLine", "RegScore"), ...)
```

**Arguments**

<code>x</code>	plot object (from <code>lm.rrpp</code> )
<code>type</code>	Indicates which type of plot, choosing among diagnostics, regression, or principal component plots. Diagnostic plots are similar to <code>lm</code> diagnostic plots, but for multivariate data. Regression plots plot multivariate dispersion in some fashion against predictor values. PC plots project data onto the eigenvectors of the covariance matrix for fitted values.
<code>predictor</code>	An optional vector if "regression" plot type is chosen, and is a variable likely used in <code>lm.rrpp</code> . This vector is a vector of covariate values equal to the number of observations.
<code>reg.type</code>	If "regression" is chosen for plot type, this argument indicates whether prediction line (PredLine) or regression score (RegScore) plotting is performed. For explanation of prediction line, see Adams and Nistri (2010). For explanation of regression score, see Drake and Klingenberg (2008).
<code>...</code>	other arguments passed to <code>plot</code> (helpful to employ different colors or symbols for different groups). See <code>plot.default</code> and <code>par</code>

**Author(s)**

Michael Collyer

## References

Drake, A. G., and C. P. Klingenberg. 2008. The pace of morphological change: Historical transformation of skull shape in St Bernard dogs. *Proc. R. Soc. B.* 275:71-76.

Adams, D. C., and A. Nistri. 2010. Ontogenetic convergence and evolution of foot morphology in European cave salamanders (Family: Plethodontidae). *BMC Evol. Biol.* 10:1-10.

---

plot.model.comparison *Plot Function for RRPP*

---

## Description

Plot Function for RRPP

## Usage

```
## S3 method for class 'model.comparison'  
plot(x, ...)
```

## Arguments

x	plot object (from <a href="#">model.comparison</a> )
...	other arguments passed to plot (helpful to employ different colors or symbols for different groups). See <a href="#">plot.default</a> and <a href="#">par</a>

## Author(s)

Michael Collyer

---

plot.predict.lm.rppp *Plot Function for RRPP*

---

## Description

Plot Function for RRPP

## Usage

```
## S3 method for class 'predict.lm.rppp'  
plot(x, PC = FALSE, ellipse = FALSE,  
      label = TRUE, ...)
```

**Arguments**

x	plot object (from <a href="#">predict.lm.rpp</a> )
PC	A logical argument for whether the data space should be rotated to its principal components
ellipse	A logical argument to change error bars to ellipses in multivariate plots. It has no function for univariate plots.
label	A logical argument for whether points should be labeled (in multivariate plots).
...	other arguments passed to plot (helpful to employ different colors or symbols for different groups). See <a href="#">plot.default</a> and <a href="#">par</a>

**Author(s)**

Michael Collyer

---

plot.trajectory.analysis

*Plot Function for RRPP*

---

**Description**

Function generates a principal component plot for trajectories

**Usage**

```
## S3 method for class 'trajectory.analysis'  
plot(x, ...)
```

**Arguments**

x	plot object (from <a href="#">trajectory.analysis</a> )
...	other arguments passed to plot (helpful to employ different colors or symbols for different groups). See <a href="#">plot.default</a> and <a href="#">par</a>

**Details**

The function calculates and plots principal components of fitted values from [lm.rpp](#) that are passed onto [trajectory.analysis](#), and projects data onto them. This function is a set.up, and [add.trajectories](#) is needed to add trajectories to the plot. By having two stages of control, the plotting functions are more flexible. This function also returns plotting information that can be valuable for making individualized plots, if [add.trajectories](#) is not preferred.

**Value**

If an object is assigned, it will return:

pca	Principal component analysis performed using <a href="#">prcomp</a> .
pc.points	Principal component scores for all data.
trajectory.analysis	Trajectory analysis passed on.
trajectories	pca Observed trajectories projected onto principal components.

**Author(s)**

Michael Collyer

**References**

- Adams, D. C., and M. M. Cerney. 2007. Quantifying biomechanical motion using Procrustes motion analysis. *J. Biomech.* 40:437-444.
- Adams, D. C., and M. L. Collyer. 2007. The analysis of character divergence along environmental gradients and other covariates. *Evolution* 61:510-515.
- Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. *Evolution* 63:1143-1154.
- Collyer, M. L., and D. C. Adams. 2007. Analysis of two-state multivariate phenotypic change in ecological studies. *Ecology* 88:683-692.
- Collyer, M. L., and D. C. Adams. 2013. Phenotypic trajectory analysis: comparison of shape change patterns in evolution and ecology. *Hystrix* 24: 75-83.
- Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 115:357-365.

**See Also**

[plot.default](#) and [par](#)

**Examples**

```
# See \link{trajectory.analysis} for examples
```

---

predict.lm.rppp	<i>predict for lm.rppp model fits</i>
-----------------	---------------------------------------

---

**Description**

Computes predicted values from an [lm.rppp](#) model fit, using bootstrapped residuals to generate confidence intervals. (Residuals are the residuals of the [lm.rppp](#) fit, not its null model. The bootstrap procedure resamples residual vectors with replacement.) The bootstrap permutations use the same number of iterations and seed as used in the [lm.rppp](#) model fit. A [predict.lm.rppp](#) object can be plotted using various options. See [plot.predict.lm.rppp](#).

**Usage**

```
## S3 method for class 'lm.rrpp'
predict(object, newdata, confidence = 0.95, ...)
```

**Arguments**

<code>object</code>	Object from <code>lm.rrpp</code> .
<code>newdata</code>	Data frame of either class <code>data.frame</code> or <code>rrpp.data.frame</code> . If null, the data frame from the <code>lm.rrpp</code> fit will be used, effectively calculating all fitted values and their confidence intervals.
<code>confidence</code>	The desired confidence interval level for prediction.
<code>...</code>	Other arguments (currently none)

**Author(s)**

Michael Collyer

**Examples**

```
# See examples for lm.rrpp to see how predict.lm.rrpp works in conjunction
# with other functions

data(Pupfish)
names(Pupfish)
Pupfish$logSize <- log(Pupfish$CS) # better to not have functions in formulas

fit <- lm.rrpp(coords ~ logSize + Sex*Pop, SS.type = "I", data = Pupfish)

# Predictions (holding alternative effects constant)

shapeDF <- expand.grid(Sex = levels(Pupfish$Sex), Pop = levels(Pupfish$Pop))
rownames(shapeDF) <- paste(shapeDF$Sex, shapeDF$Pop, sep = ".")
shapeDF

shapePreds <- predict(fit, shapeDF)
summary(shapePreds)
summary(shapePreds, PC = TRUE)

shapePreds99 <- predict(fit, shapeDF, confidence = 0.99)
summary(shapePreds99, PC = TRUE)

# Plot prediction

plot(shapePreds, PC = TRUE)
plot(shapePreds, PC = TRUE, ellipse = TRUE)
plot(shapePreds99, PC = TRUE)
plot(shapePreds99, PC = TRUE, ellipse = TRUE)
```

---

`print.anova.lm.rppp`     *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'anova.lm.rppp'  
print(x, ...)
```

**Arguments**

x                    print/summary object (from [lm.rppp](#))  
...                   other arguments passed to print/summary

**Author(s)**

Michael Collyer

---

`print.classify`             *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'classify'  
print(x, ...)
```

**Arguments**

x                    Object from [classify](#)  
...                   Other arguments passed onto classify

**Author(s)**

Michael Collyer

---

print.coef.lm.rpp      *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'coef.lm.rpp'  
print(x, ...)
```

**Arguments**

x                      Object from [coef.lm.rpp](#)  
...                     Other arguments passed onto coef.lm.rpp

**Author(s)**

Michael Collyer

---

print.lm.rpp              *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'lm.rpp'  
print(x, ...)
```

**Arguments**

x                      print/summary object (from [lm.rpp](#))  
...                     other arguments passed to print/summary

**Author(s)**

Michael Collyer



---

print.model.comparison  
*Print/Summary Function for RRPP*

---

### Description

Print/Summary Function for RRPP

### Usage

```
## S3 method for class 'model.comparison'  
print(x, ...)
```

### Arguments

x                    Object from [model.comparison](#)  
...                   Other arguments passed onto model.comparison

### Author(s)

Michael Collyer

---

print.pairwise            *Print/Summary Function for RRPP*

---

### Description

Print/Summary Function for RRPP

### Usage

```
## S3 method for class 'pairwise'  
print(x, ...)
```

### Arguments

x                    Object from [pairwise](#)  
...                   Other arguments passed onto pairwise

### Author(s)

Michael Collyer

print.predict.lm.rppp *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'predict.lm.rppp'  
print(x, PC = FALSE, ...)
```

**Arguments**

x	Object from <a href="#">predict.lm.rppp</a>
PC	Logical argument for whether to use predicted values rotated to their PCs
...	Other arguments passed onto predict.lm.rppp

**Author(s)**

Michael Collyer

---

print.summary.lm.rppp *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'summary.lm.rppp'  
print(x, ...)
```

**Arguments**

x	print/summary object (from <a href="#">summary.lm.rppp</a> )
...	other arguments passed to print/summary

**Author(s)**

Michael Collyer

---

```
print.summary.manova.lm.rppp
```

*Print/Summary Function for RRPP*

---

### **Description**

Print/Summary Function for RRPP

### **Usage**

```
## S3 method for class 'summary.manova.lm.rppp'  
print(x, ...)
```

### **Arguments**

x	Object from <a href="#">summary.manova.lm.rppp</a>
...	Other arguments passed onto <a href="#">summary.manova.lm.rppp</a>

### **Author(s)**

Michael Collyer

---

```
print.summary.pairwise
```

*Print/Summary Function for RRPP*

---

### **Description**

Print/Summary Function for RRPP

### **Usage**

```
## S3 method for class 'summary.pairwise'  
print(x, ...)
```

### **Arguments**

x	Object from <a href="#">summary.pairwise</a>
...	Other arguments passed onto <a href="#">summary.pairwise</a>

### **Author(s)**

Michael Collyer

---

```
print.summary.trajectory.analysis
    Print/Summary Function for RRPP
```

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'summary.trajectory.analysis'
print(x, ...)
```

**Arguments**

x	Object from <a href="#">summary.trajectory.analysis</a>
...	Other arguments passed onto <a href="#">summary.trajectory.analysis</a>

**Author(s)**

Michael Collyer

---

```
print.trajectory.analysis
    Print/Summary Function for RRPP
```

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'trajectory.analysis'
print(x, ...)
```

**Arguments**

x	Object from <a href="#">trajectory.analysis</a>
...	Other arguments passed onto

**Author(s)**

Michael Collyer

---

Pupfish

*Landmarks on pupfish*

---

### Description

Landmark data from *Cyprinodon pecosensis* body shapes, with indication of Sex and Population from which fish were sampled (Marsh or Sinkhole).

### Details

These data were previously aligned with GPA. Centroid size (CS) is also provided. See the **geomorph** package for details.

### Author(s)

Michael Collyer

### References

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 113: doi:10.1038/hdy.2014.75.

---

PupfishHeads

*Landmarks on pupfish heads*

---

### Description

Landmark data from *Cyprinodon pecosensis* head shapes, with variables for sex, month and year sampled, locality, head size, and coordinates of landmarks for head shape, per specimen. These data are a subset of a larger data set.

### Details

The variable, "coords", are data that were previously aligned with GPA. The variable, "headSize", is the Centroid size of each vector of coordinates. See the **geomorph** package for details.

### Author(s)

Michael Collyer

### References

Gilbert, M.C. 2016. Impacts of habitat fragmentation on the cranial morphology of a threatened desert fish (*Cyprinodon pecosensis*). Masters Thesis, Western Kentucky University.

---

residuals.lm.rppp      *Extract residuals*

---

### Description

Extract residuals

### Usage

```
## S3 method for class 'lm.rppp'  
residuals(object, weighted = TRUE, ...)
```

### Arguments

object	plot object (from <a href="#">lm.rppp</a> )
weighted	A logical argument to return weighted or unweighted residuals.
...	Arguments passed to other functions

### Author(s)

Michael Collyer

### Examples

```
# See examples for lm.rppp
```

---

reveal.model.designs      *Reveal model designs used in lm.rppp fit*

---

### Description

Function returns every full and reduced model for model terms used in `lm.rppp` fits. This function is useful for revealing the null and full model that would be used in the `pairwise` function, if a specific null model is not declared as an argument (`fit.null` in the `pairwise` function). It also helps to demonstrate how sums of squares and cross-products (SSCP) are calculated in `lm.rppp` permutations (iterations), from the difference between fitted values for null and full designs.

### Usage

```
reveal.model.designs(fit)
```

### Arguments

fit	A linear model fit from <a href="#">lm.rppp</a> .
-----	---

**Author(s)**

Michael Collyer

**Examples**

```
data(Pupfish)
fit1 <- lm.rrpp(coords~ Pop*Sex, data = Pupfish,
SS.type = "I", print.progress = FALSE, iter = 0)
fit2 <- lm.rrpp(coords~ Pop*Sex, data = Pupfish,
SS.type = "II", print.progress = FALSE, iter = 0)
fit3 <- lm.rrpp(coords~ Pop*Sex, data = Pupfish,
SS.type = "III", print.progress = FALSE, iter = 0)

reveal.model.designs(fit1)
reveal.model.designs(fit2)
reveal.model.designs(fit3)
```

---

rrpp.data.frame	<i>Create a data frame for lm.rrpp analysis</i>
-----------------	---

---

**Description**

Create a data frame for `lm.rrpp` analysis, when covariance or distance matrices are used

**Usage**

```
rrpp.data.frame(...)
```

**Arguments**

... Components (objects) to combine in the data frame.

**Details**

This function is not much different than `data.frame` but is more flexible to allow distance matrices and covariance matrices to be included. Essentially, this function creates a list, much like an object of class `data.frame` is also a list. However, `rrpp.data.frame` is less concerned with coercing the list into a matrix and more concerned with matching the number of observations (`n`). It is wise to use this function with any `lm.rrpp` analysis so that `lm.rrpp` does not have to search the global environment for needed data.

It is assumed that multiple data sets for the same subjects are in the same order.

See `lm.rrpp` for examples.

**Author(s)**

Michael Collyer

## Examples

```
# Why use a rpp.data.frame?
y <- matrix(rnorm(30), 10, 3)
x <- rnorm(10)
df <- data.frame(x = x, y = y)
df
rdf <- rpp.data.frame(x = x, y = y)
rdf # looks more like a list

is.list(df)
is.list(rdf)

d <- dist(y) # distance matrix as data

# One can try this but it will result in an error
# df <- data.frame(df, d = d)
rdf <- rpp.data.frame(rdf, d = d) # works

fit <- lm.rpp(d ~ x, data = rdf)
summary(fit)
```

---

summary.anova.lm.rpp *Print/Summary Function for RRPP*

---

## Description

Print/Summary Function for RRPP

## Usage

```
## S3 method for class 'anova.lm.rpp'
summary(object, ...)
```

## Arguments

object	Object from <a href="#">predict.lm.rpp</a>
...	Other arguments passed onto <a href="#">predict.lm.rpp</a>

## Author(s)

Michael Collyer



---

summary.classify      *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'classify'  
summary(object, ...)
```

**Arguments**

object	Object from <a href="#">classify</a>
...	Other arguments passed onto classify

**Author(s)**

Michael Collyer

---

summary.coef.lm.rpp      *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'coef.lm.rpp'  
summary(object, ...)
```

**Arguments**

object	Object from <a href="#">coef.lm.rpp</a>
...	Other arguments passed onto coef.lm.rpp

**Author(s)**

Michael Collyer

---

summary.lm.rppp      *Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'lm.rppp'
summary(object, formula = TRUE, ...)
```

**Arguments**

object	print/summary object (from <a href="#">lm.rppp</a> )
formula	Logical argument for whether to include formula in summary table
...	other arguments passed to print/summary

**Author(s)**

Michael Collyer

---

summary.manova.lm.rppp  
*Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'manova.lm.rppp'
summary(object, test = c("Roy", "Pillai",
  "Hotelling-Lawley", "Wilks"), ...)
```

**Arguments**

object	Object from <a href="#">lm.rppp</a> , updated with <a href="#">manova.update</a>
test	Type of multivariate test statistic to use.
...	Other arguments passed onto manova.lm.rppp

**Author(s)**

Michael Collyer

---

summary.model.comparison

*Print/Summary Function for RRPP*


---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'model.comparison'
summary(object, ...)
```

**Arguments**

object	Object from <a href="#">model.comparison</a>
...	Other arguments passed onto model.comparison

**Author(s)**

Michael Collyer

---

summary.pairwise

*Print/Summary Function for RRPP*


---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'pairwise'
summary(object, stat.table = TRUE,
        test.type = c("dist", "VC", "var"), angle.type = c("rad", "deg"),
        confidence = 0.95, show.vectors = FALSE, ...)
```

**Arguments**

object	Object from <a href="#">pairwise</a>
stat.table	Logical argument for whether results should be returned in one table (if TRUE) or separate pairwise tables (if FALSE)
test.type	Whether distances or vector correlations between vectors or variances (dispersion of residuals) should be used in the test.
angle.type	If test.type = "VC", whether angle results are expressed in radians or degrees.
confidence	Confidence level to use for upper confidence limit; default = 0.95 (alpha = 0.05)
show.vectors	Logical value to indicate whether vectors should be printed.
...	Other arguments passed onto pairwise

**Author(s)**

Michael Collyer

---

summary.predict.lm.rppp

*Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'predict.lm.rppp'  
summary(object, ...)
```

**Arguments**

object	Object from <a href="#">predict.lm.rppp</a>
...	Other arguments passed onto predict.lm.rppp

**Author(s)**

Michael Collyer

---

summary.trajectory.analysis

*Print/Summary Function for RRPP*

---

**Description**

Print/Summary Function for RRPP

**Usage**

```
## S3 method for class 'trajectory.analysis'  
summary(object, stat.table = TRUE,  
        attribute = c("MD", "TC", "SD"), angle.type = c("rad", "deg"),  
        confidence = 0.95, show.trajectories = FALSE, ...)
```

**Arguments**

object	Object from <a href="#">trajectory.analysis</a>
stat.table	Logical argument for whether results should be returned in one table (if TRUE) or separate pairwise tables (if FALSE)
attribute	Whether magnitude differences (MD, absolute difference in trajectory path lengths), trajectory correlations (TC), or trajectory shape differences (SD) are summarized.
angle.type	If attribute = "TC", whether angle results are expressed in radians or degrees.
confidence	Confidence level to use for upper confidence limit; default = 0.95 (alpha = 0.05)
show.trajectories	Logical value to indicate whether trajectories should be printed.
...	Other arguments passed onto trajectory.analysis

**Author(s)**

Michael Collyer

---

trajectory.analysis    *Quantify and compare shape change trajectories*

---

**Description**

Function estimates attributes of multivariate trajectories

**Usage**

```
trajectory.analysis(fit, fit.null = NULL, groups, traj.pts, pca = TRUE,
  print.progress = FALSE)
```

**Arguments**

fit	A linear model fit using <a href="#">lm.rppp</a> .
fit.null	An alternative linear model fit to use as a null model for RRPP, if the null model of the fit is not desired. Note, if RRPP = FALSE (FRPP rather than RRPP), then the null model has only an intercept. If the null model is uncertain, using <a href="#">reveal.model.designs</a> will help elucidate the inherent null model used.
groups	A factor or vector coercible to factor that defines trajectories.
traj.pts	Either a single value or a vector coercible to factor to define trajectory points. If only a single value, it is assumed that the data are already in the form, y1p1, y2p1, y3p1, ..., y2p2, y2p2, y3p2, ..., yjp1, yjp2, yjp3, ..., yjpk, for j variables comprising k trajectory points; i.e., traj.pts = k. If a factor, then a group * traj.pt factorial model is assumed, where traj.pts defines the levels for points within groups.

pca	A logical value to optionally project group:point means onto principal components (perform PCA on a covariance matrix of the means) This option only applies to factorial designs (traj.pts is a factor).
print.progress	A logical value to indicate whether a progress bar should be printed to the screen. This is helpful for long-running analyses.

## Details

The function quantifies multivariate trajectories from a set of observations, and assesses variation in attributes of the trajectories via RRPP. A trajectory is defined by a sequence of points in the data space. These trajectories can be quantified for various attributes (their size, orientation, and shape), and comparisons of these attribute enable the statistical comparison of shape change trajectories (Collyer and Adams 2007; Adams and Collyer 2007; Adams and Collyer 2009; Turner et al. 2010; Collyer and Adams 2013).

This function is a modified version of [pairwise](#), retaining the least squares (LS) means as trajectory points. Analysis starts with a [lm.rrpp](#) fit (but a [procD.lm](#) fit from [geomorph](#) can also be used). LS means are calculated using a grouping variable. Data can be trajectories, as a start (sensu Adams and Cerney 2007), or trajectories can be calculated from data using a factorial model (in which case trajectory points are defined by factor levels).

This function produces statistics that can be summarized with the [summary.trajectory.analysis](#) function. The summaries are consistent with those in the [summary.pairwise](#) function, pertaining to trajectory attributes including, magnitude difference (MD), the difference in path lengths of trajectories; trajectory correlations (TC), better thought of as angular differences between trajectory principal axes; and if trajectories have three or more points, shape difference (SD), the square root of summed squared point differences, after scaling, centering, and rotating trajectories. The SD is the "Procrustes" distance between trajectories (Adams and Collyer 2009), much the same way as the shape difference between anatomical landmark configurations in geometric morphometrics. If attribute = "TC" is chosen for the summary, then the angle type ("rad" or "deg", can be chosen for either radians and degrees, respectively, to return angles between principal axes.)

Plotting can be performed with [plot.trajectory.analysis](#) and [add.trajectories](#). The former plots all principal component scores for the data, and allows point-by-point control of plot parameters. The later adds trajectories points and lines, with constrained control. By saving the [plot.trajectory.analysis](#) object, plotting information can be retained and advanced plotting can be performed. See examples below.

## Value

An object of class "trajectory.analysis" returns a list of the following:

LS.means	LS.means from pairwise function.
trajectories	Trajectories from every permutation.
PD	Path distances of trajectories from every permutation.
MD	Magnitude differences between trajectories from every permutation.
TC	Trajectory correlations from every permutation.
SD	Trajectory shape differences from every permutation.

**Author(s)**

Dean Adams and Michael Collyer

**References**

Adams, D. C., and M. M. Cerney. 2007. Quantifying biomechanical motion using Procrustes motion analysis. *J. Biomech.* 40:437-444.

Adams, D. C., and M. L. Collyer. 2007. The analysis of character divergence along environmental gradients and other covariates. *Evolution* 61:510-515.

Adams, D. C., and M. L. Collyer. 2009. A general framework for the analysis of phenotypic trajectories in evolutionary studies. *Evolution* 63:1143-1154.

Collyer, M. L., and D. C. Adams. 2007. Analysis of two-state multivariate phenotypic change in ecological studies. *Ecology* 88:683-692.

Collyer, M. L., and D. C. Adams. 2013. Phenotypic trajectory analysis: comparison of shape change patterns in evolution and ecology. *Hystrix* 24: 75-83.

Collyer, M.L., D.J. Sekora, and D.C. Adams. 2015. A method for analysis of phenotypic change for phenotypes described by high-dimensional data. *Heredity*. 115:357-365.

**Examples**

```
### Analysis of sexual dimorphism vectors (factorial approach)
data(Pupfish)
fit <- lm.rpp(coords ~ Pop * Sex, data = Pupfish, iter = 199)
reveal.model.designs(fit)
TA <- trajectory.analysis(fit, groups = Pupfish$Pop,
traj.pts = Pupfish$Sex, print.progress = FALSE)
summary(TA, attribute = "MD") # Magnitude difference (absolute difference between path distances)
summary(TA, attribute = "TC", angle.type = "deg") # Correlations (angles) between trajectories
summary(TA, attribute = "SD") # No shape differences between vectors

# Retain results
TA.summary <- summary(TA, attribute = "MD")
TA.summary$summary.table

# Plot results
TP <- plot(TA, pch = as.numeric(Pupfish$Pop) + 20, bg = as.numeric(Pupfish$Sex),
cex = 0.7, col = "gray")
add.trajectories(TP, traj.pch = c(21, 22), start.bg = 1, end.bg = 2)
legend("topright", levels(Pupfish$Pop), pch = c(21, 22), pt.bg = 1)

### Analysis when data are already trajectories (motion paths)

# data are planar Cartesian coordinates (x, y) across 5 points (10 variables)
data(motionpaths)
fit <- lm.rpp(trajectories ~ groups, data = motionpaths, iter = 199)
TA <- trajectory.analysis(fit, groups = motionpaths$groups, traj.pts = 5)
summary(TA, attribute = "MD") # Magnitude difference (absolute difference between path distances)
summary(TA, attribute = "TC", angle.type = "deg") # Correlations (angles) between trajectories
summary(TA, attribute = "SD") # Shape differences between trajectories
```

```
TP <- plot(TA, pch = 21, bg = as.numeric(motionpaths$groups),  
cex = 0.7, col = "gray")  
add.trajectories(TP, traj.pch = 21, traj.bg = 1:4)
```

---

vec.cor.matrix

*Support function for RRPP*

---

### **Description**

Calculate vector correlations for a matrix (by rows). Used for pairwise comparisons.

### **Usage**

```
vec.cor.matrix(M)
```

### **Arguments**

M                    Matrix for vector correlations.

### **Author(s)**

Michael Collyer



# Index

## \*Topic **analysis**

- classify, 6
- lm.rpp, 11
- manova.update, 17
- model.comparison, 20
- pairwise, 22
- reveal.model.designs, 38
- trajectory.analysis, 45

## \*Topic **datasets**

- motionpaths, 22
- PlethMorph, 25
- Pupfish, 37

## \*Topic **utilities**

- add.trajectories, 3
- anova.lm.rpp, 5
- coef.lm.rpp, 9
- fitted.lm.rpp, 10
- plot.lm.rpp, 26
- plot.model.comparison, 27
- plot.predict.lm.rpp, 27
- plot.trajectory.analysis, 28
- predict.lm.rpp, 29
- print.anova.lm.rpp, 31
- print.classify, 31
- print.coef.lm.rpp, 32
- print.lm.rpp, 32
- print.model.comparison, 33
- print.pairwise, 33
- print.predict.lm.rpp, 34
- print.summary.lm.rpp, 34
- print.summary.manova.lm.rpp, 35
- print.summary.pairwise, 35
- print.summary.trajectory.analysis, 36
- print.trajectory.analysis, 36
- residuals.lm.rpp, 38
- rpp.data.frame, 39
- summary.anova.lm.rpp, 40
- summary.classify, 41

- summary.coef.lm.rpp, 41

- summary.lm.rpp, 42

- summary.manova.lm.rpp, 42

- summary.model.comparison, 43

- summary.pairwise, 43

- summary.predict.lm.rpp, 44

- summary.trajectory.analysis, 44

- vec.cor.matrix, 48

## \*Topic **visualization**

- add.trajectories, 3

- plot.lm.rpp, 26

- plot.model.comparison, 27

- plot.predict.lm.rpp, 27

- plot.trajectory.analysis, 28

- add.trajectories, 3, 28, 46

- anova, 12, 21

- anova.lm.rpp, 3, 5, 12, 18

- classify, 6, 31, 41

- coef.lm.rpp, 3, 9, 12, 32, 41

- data.frame, 30, 39

- fitted.lm.rpp, 10

- lda, 7

- lm, 11, 14, 26

- lm.rpp, 3, 5–7, 9, 10, 11, 17, 18, 22–24, 26, 28–32, 38, 39, 42, 45, 46

- manova.update, 12, 17, 42

- model.comparison, 20, 27, 33, 43

- motionpaths, 22

- pairwise, 3, 22, 33, 38, 43, 46

- par, 4, 5, 26–29

- PlethMorph, 25

- plot, 21

- plot.default, 5, 26–29

- plot.lm.rpp, 26

plot.model.comparison, 27  
plot.predict.lm.rrpp, 27, 29  
plot.trajectory.analysis, 4, 28, 46  
prcomp, 29  
predict.lm.rrpp, 3, 28, 29, 29, 34, 40, 44  
print.anova.lm.rrpp, 31  
print.classify, 31  
print.coef.lm.rrpp, 32  
print.lm.rrpp, 32  
print.model.comparison, 33  
print.pairwise, 33  
print.predict.lm.rrpp, 34  
print.summary.lm.rrpp, 34  
print.summary.manova.lm.rrpp, 35  
print.summary.pairwise, 35  
print.summary.trajectory.analysis, 36  
print.trajectory.analysis, 36  
Pupfish, 37  
PupfishHeads, 37  
  
residuals.lm.rrpp, 38  
reveal.model.designs, 22, 38, 45  
RRPP (RRPP-package), 3  
RRPP-package, 3  
rrpp.data.frame, 11, 30, 39  
  
summary.anova.lm.rrpp, 40  
summary.classify, 41  
summary.coef.lm.rrpp, 41  
summary.lm.rrpp, 34, 42  
summary.manova.lm.rrpp, 12, 17, 18, 35, 42  
summary.model.comparison, 43  
summary.pairwise, 23, 35, 43, 46  
summary.predict.lm.rrpp, 44  
summary.trajectory.analysis, 36, 44, 46  
  
trajectory.analysis, 28, 36, 45, 45  
  
vec.cor.matrix, 48