Package: PRC (via r-universe)

October 31, 2024

Type Package

Title Principal Response Curves (PRC) For Displaying Multivariate Main Effects And Interactions

Version 0.2.2

Maintainer Cajo J.F. ter Braak <cajo.terbraak@wur.nl>

Description PRC is a small \code{vegan}-based package to create and display Principal Response Curves (PRC), which are a special case of Redundancy Analysis (rda) for multivariate responses in designed experiments and repeated measurement designs. PRC expresses ordination scores as deviation from a reference, usually to visualize multivariate A-dependent effects (e.g time-dependent effects) of a treatment B in (un)balanced factorial experiments. The main user functions are \code{\link{doPRC}}, \code{\link{plotPRC}} and \code{\link{plotPRC2d}}. The easiest example is the code demo \code{PRC_pyrifos}.

URL https://doi.org/10.1002/etc.5620180207,
https://doi.org/10.1016/j.chemolab.2023.104898

BugReports https://github.com/CajoterBraak/PRC/issues

Depends R (>= 3.5.0)

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Imports vegan, dplyr, ggplot2, gridExtra, latex2exp, ggrepel

RoxygenNote 7.3.0

NeedsCompilation no

Author Cajo J.F ter Braak [aut, cre]
(<https://orcid.org/0000-0002-0414-8745>)

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

RemoteUrl https://github.com/CajoterBraak/PRC

RemoteRef HEAD

RemoteSha 4461bebecd36b7dbbd164a929f18b2764916e22c

Contents

biomonitoring 2
CPTNUT_phytoplankton 3
doPRC 4
fvalues4levels 6
get_focal_and_conditioning_factors 7
Ossenkampen 8
plotPRC 9
plotPRC2d 12
plot_sample_scores_cdt 15
plot_species_scores_2d 17
plot_species_scores_bk 19
PRC 20
PRC_scores 21
rdacca_taxon_stats 23

Index **25**

`biomonitoring` *Biomonitoring example from van den Brink et al. (2009)*

Description

The data frame `biomonitoring` contains the design and species data from van den Brink et al.(2009). The data is on biomonitoring of macroinvertebrate species in the Rhine and Meuse rivers, near the towns of Kampen and Grave, respectively, from 1992 - 2000, sampled six times per year, except in 1998 (4x), 1999 and 2000 (7x). The first example uses reference coding of an reduncancy analysis without covariates so as to emphasize the compositional difference from a reference. The second example is a more regular PRC, contrasting the compositional changes at one site without those in another.

- `sample` character identifier for the sample.
- `year` numeric year of sampling
- `location` location of sampling: factor with levels c("Grave","Kampen")
- counts of 139 macroinvertebrate species

References

van den Brink, PJ, den Besten, PJ, bij de Vaate, A and ter Braak, CJF, 2009. Principal response curves technique for the analysis of multivariate biomonitoring time series. Environmental Monitoring and Assessment. 152, 271-281. http://dx.doi.org/10.1007/s10661-008-0314-6

`CPTNUT_phytoplankton` *Phytoplankon data from the chlorpyrifos-nutrient (CPF-NUT) experiment from van den Brink & ter Braak (1998)*

Description

The dataframe `CPTNUT_phytoplankton` contains the design and phytoplankton data from the microcosm example of PRC from van den Brink & ter Braak (1998). In this experiment from van Donk et al. (1995), there are three treatments: control (`con`), nutrients (`nut`) and nutrients with chlorpyrifos application (`cpf-nut`), each with four cosms (replications) sampled 13 times (twice before nutrients where added and six times before the additional application of chorpyrifos). The phytoplankton data are counts and there are 26 species in total.

- `samples` character identifier for the sample.
- `Time` factor coding for the week of sampling with respect to first application of nutrients.
- `Treatment` treatment factor as in van den Brink & ter Braak (1998). The levels are `con`, `nut` and `cpf-nut` for the four control cosms, the four cosms that receive nutrients after the first two sampling points, and the four cosms to which both nutrients and chorpyrifos are applied, respectively. Note that the cosms of the nutrient-chlorpyrifos treatment are coded herein as `nut` in the weeks before the first chlorpyrifos application, as they are treated experiment-wise similarly in these weeks. But, note that the samples from `con` and `nut` are coded differently on pre-treatment times (although they are similar experiment-wise), so as to see the pre-treatment variability. See also the factor `con_nut_cpf`.
- `Plot` numeric identifier for the microcosm.
- `con_nut_cpf` treatment factor. Before application of nutrients or chlorpyrifos, the samples are all treated similarly, but are coded differently (as often in PRC).
- cell counts of 25 different phytoplankton taxa in 1L and density for the 26th species, Volvox. For details, see van den Donk et al. (1995).

References

van den Brink, PJ & ter Braak, CJF (1998). Multivariate analysis of stress on experimental ecosystems by principal response curves and similarity analysis. Aquatic ecology, 32, 163-178. https://doi.org/10.1023/A:10099440

Van Donk E, Prins H, Voogd HM, Crum SJH and Brock TCM (1995) Effects of nutrient loading and insecticide application on the ecology of Elodea – dominated freshwater microcosms. I. Responses of plankton and zooplanktivorous insects. Arch Hydrobiol 133: 417–439. https://doi.org/10.1127/archiv-hydrobiol/133/1995/417

doPRC — *Perform a Principal Response Curve analysis (PRC)*

Description

`doPRC` perform a Principal Response Curve analysis (PRC) using the `S3` `formula` interface of `rda` or `cca`

Usage

```
doPRC(
  formula,
  scale = FALSE,
  referencelevel = NULL,
  rank = 2,
  flip = rep(FALSE, rank),
  scaling = "ms",
  method = "rda",
  data,
  ...
)
```

Arguments

`formula` formula specifying how the response (species data) must modeled by predictors and covariates. A typical `formula` is `Y~treatment*week+Condition(week)`, where `Y` is the global response data matrix or data frame. Typically these are community (species) data. If count-like, they probably should be log transformed prior to the analysis, because a multiplicative model has more realism for non-negative data than an additive model.

`scale` Scales species to unit variance (like correlations) if `method` is `"rda"` (default: `FALSE`), else void. See `rda`.

`referencelevel` numeric or character level(s) to be used as reference level or levels of the focal treatment(s); default: 1 (first level(s)). The focal treatment is (or treatments are) given in the resulting list at ``result$focal_and_conditioning_factors$`focal factor` ``

`rank` number of axes to be computed; default 2.

`flip` logical value or vector; default: FALSE. Should the axes be flipped, i.e. reversed in orientation? `flip` can be numeric with values -1 and 1 for reverse and no reverse, respectively, i.e. multiply the scores by either -1 or 1.

`scaling` character `c("ms", "ss")`. Scales the axes of an RDA object to mean square species scores of 1 (`"ms"`) or a sum of squares of 1 (`"ss"`). The only scaling for a CCA object is `"ss"` (weighted mean square = 1, with weight = species total abundance). For a dbrda object, there are no species scores and the scaling is like "ss". The default scaling (`"ms"` for RDA and `"ss"` for CCA) has the advantage that the PRC treatment and sample scores measure the effect sizes of the ’average’ species as their (weigthed) mean square is 1 (sd = 1); it is the scaling used in Canoco5 (www.canoco5.com). Scaling "ss" is the default in `prcomp` and used in ter Braak (2023).

`data` Data frame containing the variables on the right hand side of the model formula.

Details

The function performs PRC using `rda` or `cca` with the specified `formula` and applies `PRC_scores` to the result.

Value

an object as a `cca.object` with additional entries as output of `PRC_scores`. The assigned classes are `c("PRC","rda","cca"`.

References

ter Braak (2023) Redundancy analysis includes analysis of variance-simultaneous component analysis (ASCA) and outperforms its extensions Chemometrics and Intelligent Laboratory Systems https://doi.org/10.1016/j.chemolab.2023.104898

ter Braak, C.J.F. & te Beest, D.E. 2022. Testing environmental effects on taxonomic composition with canonical correspondence analysis: alternative permutation tests are not equal. Environmental and Ecological Statistics. 29 (4), 849-868. https://doi.org/10.1007/s10651-022-00545-4

van den Brink, P.J. & ter Braak, C. (1998) Multivariate analysis of stress in experimental ecosystems by Principal Response Curves and similarity analysis. Aquatic Ecology, 32, 163-178. http://dx.doi.org/10.1023/A:100994400

van den Brink, P.J. & ter Braak, C.J.F. (1999) Principal Response Curves: Analysis of time-dependent multivariate responses of a biological community to stress. Environmental Toxicology and Chemistry, 18, 138-148. https://doi.org/10.1002/etc.5620180207

van den Brink, P.J., den Besten, P., bij de Vaate, A. & ter Braak, C.J.F. (2009) Principal response curves technique for the analysis of multivariate biomonitoring time series. Environmental Monitoring and Assessment, 152, 271-281. http://dx.doi.org/10.1007/s10661-008-0314-6

Examples

```
data(pyrifos, package = "vegan") #log-transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                   dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)),
                     ditch = gl(12, 1, length=132))
#

mod_prc <- doPRC(pyrifos ~ dose:week + Condition(week),  data = Design)
mod_prc$focal_and_conditioning_factors$`focal factor` # as expected: "dose"
print(mod_prc)

#                 Proporti Rank
# Total           1.0000
# Conditional     0.2192   10 (between-week variation)
# Constrained     0.3346   44 (between-treatment variation across weeks)
# Unconstrained   0.4462   77 (within-week variation)

# without lines, threshold is 7
plotPRC(mod_prc)
# this threshold is about P = 0.01 for PRC1 in the regression "species_k ~ week+PRC1"
#                and retains 34 species to be plotted with names
sum(mod_prc$species[,"Fratio1"] >7) # 34
# only the PRC curves:
plotPRC(mod_prc, plot = 0, symbols_on_curves = TRUE)
# with lines and a stronger threshold
plotPRC(mod_prc, plot = "ditch", threshold = 10,width = c(4,1))

# modifying the plot
gg <- plotPRC(mod_prc, plot = "ditch",width = c(4,1), verbose = FALSE)
p1 <- gg$separateplots$treatments + ggplot2::ggtitle(paste("new title:", latex2exp::TeX("$c_{dt}$")))  # PRC plot
p2 <- gg$separateplots$species   + ggplot2::ylab("new title: loadings")# loadings of species  (b_k)
# Assign these plots to symbols and use grid.arrange to produce the plot  you like, for example:
gridExtra::grid.arrange(p1+ ggplot2::ylab("")+ ggplot2::xlab("week since chorpyrifos application") + ggplot2::ggt
                        p2, ncol =2, widths = c(4,1),
                top = "new title", left ="PRC curves of treaments", right =  "PRC species loadings")

# test significance of PRC/RDA axes ----------------------------------------

## Ditches are randomized, we have a time series, and are only
## interested in the first axis
library(vegan)
ctrl <- how(plots = Plots(strata = Design$ditch,type = "free"),
            within = Within(type = "none"), nperm = 99)

anova(mod_prc, by = "axis",permutations=  ctrl, model = "reduced", cutoff = 0.10)
```

---

fvalues4levels *Change numeric-like levels of a factor to a numeric*

Description

fvalues4levels changes levels of a factor to numeric variable for plotting of the PRC diagram

Usage

```
fvalues4levels(object, fctr)
```

Arguments

object	a data frame or tibble.
fctr	name of a factor in object (character or symbol)

Value

a numeric named vector (names are levels)

Examples

```
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                   dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)))
fvalues4levels(Design, "week")

# fvalues4levels strips the starting k characters so as to find numbers,e.g.
Design$week2 <- factor(paste("Week", Design$week))
levels(Design$week2)
fvalues4levels(Design, "week2")
# they may not be in increasing order in general,
# but will be after sorting:
sort(fvalues4levels(Design, "week2"))
# the resulting numeric variable:
numvar <- fvalues4levels(Design, "week2")[Design[,"week2"]]
summary(numvar)
length(numvar)== nrow(Design)
```

`get_focal_and_conditioning_factors` *Get constraining and conditional variables from an rda or cca ordination*

Description

`get_focal_and_conditioning_factors` derives the focal constraining variables(s) and (if present) the conditioning variables(s) from a `rda` or `cca` ordination specified via a formula for use in a PRC (with `doPRC` or `PRC_scores`)

Usage

```
get_focal_and_conditioning_factors(object)
```

Arguments

`object` a result of `rda`, `cca` specified via a formula (S3 method for class 'formula'), or a result of `doPRC` or `PRC_scores`.

Value

A list with element `` `focal factor` `` and `condition`

References

ter Braak C.J.F. & Šmilauer P. (2018): Canoco reference manual and user's guide: software for ordination, version 5.1x. Microcomputer Power, Ithaca, USA, 536 pp. (http::www.canoco5.com)

See Also

doPRC, PRC_scores

Examples

```
data(pyrifos, package = ”vegan”) #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                     dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)))

library(vegan)
mod_rda <- rda( pyrifos ~ dose:week + Condition(week),  data = Design)
species_scores= scores(mod_rda,choices =1:2, display=”sp”,scaling = 1)
library(PRC)
head(rdacca_taxon_stats(mod_rda)) # stats per species
get_focal_and_conditioning_factors(mod_rda) # factors from the model

species_scores= vegan::scores(mod_rda,choices =1:2, display=”sp”,scaling = 2)
plot_species_scores_bk(species_scores) # 1-d vertical plot of species scores (a cut of RDA1-scores at 0.5)
plot_species_scores_bk(species_scores, threshold = 14) #  a cut of RDA1-scores at 1.0 (as scoresname is unset)

plot_species_scores_2d(species_scores)
plot_species_scores_2d(species_scores, label.repel = TRUE)
plot_species_scores_2d(species_scores, max.overlaps = 30, label.repel = TRUE)
```

Ossenkampen *Data from Ossenkampen experiment*

Description

The dataframe Ossenkampen contains the data from a long-term fertilizer experiment (1958-2007) in grassland with 98 response variables (abundances of the plant species that grew in the Ossenkampen plots, measured as counts in 100 subsamples), as analysed in ter Braak (2023).

The experiment started in 1958 with 12 plots arranged in two randomized complete blocks with six treatments: four types of fertilizer (K, P, PK or NPK), liming and no fertilizer. Eight years later (in 1966) each block was extended with two plots, one with N and another with NPK fertilization. The two limed plots behaved rather differently from the rest and are omitted here. In this data, the K and P plots are combined with the no fertilizer plots so that the control treatment (Cntrl) consisted of 6 plots, with the aim to make the variability among replicates more interesting. Three plots of the control and one PK plot were not sampled in 1967 and two NPK samples were not sampled in 1984; these missing samples were not imputed in this data set.

The data analyzed in Berendse et al. (2021) also contained the limed plots and contained, for the multivariate analyses, imputations of the few missing samples after 1966.

- Sample numeric identifier for a sample
- Plot identifier for a sample (the identifier details the fertilizer treatment and Block)

- `Block` numeric identifier of the block (1 or 2)
- `A` year of sampling
- `B` fertilizer treatment, condensed to four treatments (`"Cntrl"` is the referencelevel)
- `AgrCap` count out of 100 subsamples of species AgrCap, and so on for the remaining 97 species

References

Berendse, F., Geerts, R.H.E.M., Elberse, W.T., Bezemer, T.M., Goedhart, P.W., Xue, W., Noordijk, E., ter Braak, C.J.F. & Korevaar, H. (2021) A matter of time: Recovery of plant species diversity in wild plant communities at declining nitrogen deposition. Diversity and Distributions, 27, 1180-1193. https://doi.org/10.1111/ddi.13266

Geerts, R.H.E.M., Bufe, C., Schils, P.C., Korevaar, H., Berendse, F. & Struik, P.C. (2021) The Ossekampen Grassland Experiment: Data underlying the publication: A matter of time: recovery of plant species diversity in wild plant communities at declining nitrogen deposition. https://doi.org/10.4121/13200398

ter Braak, C.J.F. (2023) Redundancy analysis includes analysis of variance-simultaneous component analysis (ASCA) and outperforms its extensions. Chemometrics and Intelligent Laboratory Systems, 104898. https://doi.org/10.1016/j.chemolab.2023.104898

`plotPRC` *1d PRC diagram*

Description

`plotPRC` creates a one-dimensional PRC diagram with vertical line plot of species loadings from a result of `doPRC` or `PRC_scores`.

Usage

```
plotPRC(
  object,
  treatment = NULL,
  condition = NULL,
  plot = 1,
  xvals = NULL,
  axis = 1,
  size = c(2, 2),
  symbols_on_curves = FALSE,
  width = c(4, 1),
  title = NULL,
  left = "Treatment and plot scores",
  right = "taxon scores",
  threshold = 7,
  y_lab_interval = 0.5,
  speciesname = NULL,
  selectname = "Fratio",
  verbose = TRUE
)
```

Arguments

`object` a result of `doPRC` or `PRC_scores` or a list with element PRCplus (a data frame with a variables to plot, see details).

`treatment` character vector of names of factors in `object$PRCplus`, the first two of which set the `color` and `linetype` of treatments. Default: derived from `object$'focal_and_conditioning_f`

`condition` character vector of names of factors in `object$PRCplus`, the first one being used as x-axis. Default: derived from `object$'focal_and_conditioning_factors'`.

`plot` 0 or 1 (no/yes individual plot points) or character name of a factor in `object$PRCplus` indicating the plots (microcosms, subjects or objects) in a longitudinal study so as to plot lines. Default: 1 yielding points instead of lines.

`xvals` numeric values for the levels of the condition factor, or name of numeric variable in `object$PRCplus` to be used for the x-axis. Default: NULL for which the values are obtained using the function `fvalues4levels` applied to `object$PRCplus[[condition[1]]]`

`axis` axis shown; default 1, showing the first PRC axis.

`size` size of symbols and symbols in the legend. default: c(2, 2). If the length is not 2 than the first entry is replicated.

`symbols_on_curves`
logical, draw symbols on the PRC curves at the measurement times (default FALSE). If `TRUE`, symbols are for the treatment, if there is one, and for the second treatment if there are two.

`width` relative width of treatment PRC plot and the loading plot (see `grid.arrange`).

`title` overall title.

`left` left axis text (see `arrangeGrob`).

`right` right axis text (see `arrangeGrob`).

`threshold` species with criterion (specified by `selectname`) higher than the `threshold` are displayed. Default: 7 (which is the threshold F-ratio for testing a single regression coefficient at `p=0.01` with `60` df for the error in a multiple regression of each single species onto the condition and the ordination axis). If `selectname` is not in `species_scores`, the `threshold` is divided by `14`, so that the default is 0.5.

`y_lab_interval` interval of the y-axis ticks. A tick at no effect (0) is always included; default: 0.5.

`speciesname` name of the variable containing the species names (default `NULL` uses rownames in `object$species`)

`selectname` name of the column or variable containing the criterion for the selection of species to be displayed without `"1"` or other `axis` number. Default: `"Fratio"`;if `selectname` is not found in `object$species_scores`, set to `scoresname`. e.g. the `"RDA1"` scores.

`verbose` logical for printing the current model, `score_name`, `condition` and `treatment` (default TRUE).

Details

The function creates the treatment PRC diagram using plot_sample_scores_cdt and the vertical line plot of loadings using plot_species_scores_bk. These ggplot2-objects are combined in a single object using arrangeGrob.

Value

a list of a gridExtra object (plot using arrangeGrob) and a list of two ggplot2 objects (the treatment PRC-diagram and the vertical line plot).

See Also

doPRC, plotPRC2d, plot_sample_scores_cdt, plot_species_scores_bk and grid.arrange

Examples

```
data(pyrifos, package = "vegan") #log-transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                   dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)),
                     ditch = gl(12, 1, length=132))
#

mod_prc <- doPRC(pyrifos ~ dose:week + Condition(week),  data = Design)
mod_prc$focal_and_conditioning_factors$`focal factor` # as expected: "dose"
print(mod_prc)

#                 Proporti Rank
# Total           1.0000
# Conditional     0.2192   10 (between-week variation)
# Constrained     0.3346   44 (between-treatment variation across weeks)
# Unconstrained   0.4462   77 (within-week variation)

# without lines, threshold is 7
plotPRC(mod_prc)
# this threshold is about P = 0.01 for PRC1 in the regression "species_k ~ week+PRC1"
#                 and retains 34 species to be plotted with names
sum(mod_prc$species[,"Fratio1"] >7) # 34
# only the PRC curves:
plotPRC(mod_prc, plot = 0, symbols_on_curves = TRUE)
# with lines and a stronger threshold
plotPRC(mod_prc, plot = "ditch", threshold = 10,width = c(4,1))

# modifying the plot
gg <- plotPRC(mod_prc, plot = "ditch",width = c(4,1), verbose = FALSE)
p1 <- gg$separateplots$treatments + ggplot2::ggtitle(paste("new title:", latex2exp::TeX("$c_{dt}$")))   # PRC plot
p2 <- gg$separateplots$species   + ggplot2::ylab("new title: loadings")# loadings of species  (b_k)
# Assign these plots to symbols and use grid.arrange to produce the plot  you like, for example:
gridExtra::grid.arrange(p1+ ggplot2::ylab("")+ ggplot2::xlab("week since chorpyrifos application") + ggplot2::ggt
                        p2, ncol =2, widths = c(4,1),
                top = "new title", left ="PRC curves of treaments", right =  "PRC species loadings")
```

```
# test significance of PRC/RDA axes ---------------------------------------

## Ditches are randomized, we have a time series, and are only
## interested in the first axis
library(vegan)
ctrl <- how(plots = Plots(strata = Design$ditch,type = "free"),
            within = Within(type = "none"), nperm = 99)

anova(mod_prc, by = "axis",permutations=  ctrl, model = "reduced", cutoff = 0.10)
```

plotPRC2d *2d PRC diagram*

Description

plotPRC2d creates a two-dimensional PRC diagram with species loadings from a result of doPRC or PRC_scores

Usage

```
plotPRC2d(
  object,
  treatment = NULL,
  condition = NULL,
  plot = 1,
  xvals = NULL,
  axes = c(1, 2),
  threshold = 7,
  title = NULL,
  left = NULL,
  right = NULL,
  speciesname = NULL,
  selectname = "Fratio",
  xylab = NULL,
  label.repel = FALSE,
  withnames_only = FALSE,
  max.overlaps = 10,
  mult_expand = 0.1,
  widths = c(3, 2),
  verbose = TRUE
)
```

Arguments

object	a result of doPRC or PRC_scores or a list with element PRCplus (a data frame with a variables to plot, see details).

`treatment` character vector of names of factors in `object$PRCplus`, the first two of which set the `color` and `linetype` of treatments. Default: derived from `object$'focal_and_conditioning_f`

`condition` character vector of names of factors in `object$PRCplus`, the first one being used as x-axis. Default: derived from `object$'focal_and_conditioning_factors'`.

`plot` 0 or 1 (no/yes individual plot points) or character name of a factor in `object$PRCplus` indicating the plots (microcosms, subjects or objects) in a longitudinal study so as to plot lines. Default: 1 yielding points instead of lines.

`xvals` numeric values for the levels of the condition factor, or name of numeric variable in `object$PRCplus` to be used for the x-axis. Default: NULL for which the values are obtained using the function `fvalues4levels` applied to `object$PRCplus[[condition[1]]]`

`axes` axes shown; default `c(1,2)` showing the first two PRC axes.

`threshold` species with criterion (specified by `selectname`) higher than the `threshold` are displayed. Default: 5 (which is the threshold F-ratio for testing two regression coefficients (one for each axis) at `p=0.01` with `60` df for the error in a multiple regression of each single species onto the condition and the ordination axes). If `selectname` is not in the data the `threshold` is divided by ten, so that the default is 0.5.

`title` overall title.

`left` left axis text (see `arrangeGrob`).

`right` right axis text (see `arrangeGrob`).

`speciesname` name of the variable containing the species names (default `NULL` uses rownames).

`selectname` name of the column or variable containing the criterion for the selection of species to be displayed Default: `"Fratio"`; if `selectname` is not found in `object$species_scores`, set to the Euclidian norm (length) of the vectors defined by row-wise combining the values defined by `scorenames`.

`xylab` character vector of length 2 defining x- and y-axis title of the species ordination so as to overwrite the defauls axis labels

`label.repel` logical (default: `FALSE`) for using labels in white boxes with borders using `geom_label_repel` instead of using plain text labels using `geom_text_repel`.

`withnames_only` logical to plot the species that exceed the `treshold` only, i.e. to *not* plot points indicating species that do not reach the `threshold`. (Default: `FALSE`).

`max.overlaps` exclude text labels that overlap too many things. Default: 10 (see `geom_text_repel`).

`mult_expand` fraction of range expansion of the diagram (default = 0.1). See `expansion`.

`verbose` logical for printing the number of species that `geom_text_repel` or `geom_label_repel` will try to name in the plot out of the total number of species (default: `TRUE`).

Details

The function creates the treatment PRC diagrams for the `axes` using `plot_sample_scores_cdt`, with diagrams for the `x=y` and `x=-y` directions, and the 2d species loading plot using `plot_species_scores_2d`. These ggplot2-objects are combined in a single object using `arrangeGrob`.

Value

a list of a gridExtra object (plot using arrangeGrob) and a list of five ggplot2 objects, arranged in a list of two: treatments = the PRC1, PRC2-diagrams with line x=y and x= -y diagrams and and species= the 2d species ordination plot).

See Also

doPRC, plot_sample_scores_cdt, plot_species_scores_2d and grid.arrange

Examples

```
data("CPFNUT_phytoplankton")

# Read example data ----------------------------------------------------


names(CPFNUT_phytoplankton)[1:6]
Design <- CPFNUT_phytoplankton[,c("Time","Treatment","Plot","con_nut_cpf")]
summary(Design)
print(with(Design,table(Time,Treatment)))

# extract response variables
Counts <- as.matrix(CPFNUT_phytoplankton[,-(1:5)]); rownames(Counts)<-CPFNUT_phytoplankton$sample
# transform as a multiplicative model is more logical than an additive one.
# But need to add pseudocount to avoid log(0) = infinite

Y <- log(Counts+1)
# Remark: this transformation reproduces the results in van den Brink & ter Braak 1998
#         apart from scaling (but the product b_k*c_dt is identical) and
#         apart from the PRC1 score at week 4, which is erroneous in the paper.
# PRC via vegan -----------------------------------------------------------

mod_prc <- doPRC( Y~ Time * Treatment + Condition(Time),  rank = 3, data= Design, flip= c(FALSE, TRUE, FALSE))
round(mod_prc$percExp, 0)

plotPRC(mod_prc, plot = "Plot", width = c(2,1), symbols_on_curves = TRUE)

plotPRC(mod_prc, plot = "Plot", width = c(2,1), axis = 2)
# note that the default selection criterion "Fratio2" also selects species strongly reacting to axis 1

plotPRC2d(mod_prc, plot = 0, threshold = 1,title = "PRC Phytoplankton van den Brink & ter Braak (1998)")

# modifying the plot
# Assign the plots to symbols and use grid.arrange to produce the plot you like, for example:
gg <- plotPRC2d(mod_prc, plot = 0, verbose = FALSE)
pl_species <- gg$separateplots$species # species ordination
pl <- gg$separateplots$treatments
names(pl)

layout<- rbind(c(1,2), c(3,4),c(3,5))
gridExtra::grid.arrange(pl[["PRC1"]]+ggplot2::ylab("PRC1")+ggplot2::ggtitle(""),
                pl[["PRC2"]]+ggplot2::ylab("PRC2")+ggplot2::xlab("")+ggplot2::theme(legend.position="none")+gg
```

```
        pl_species,
pl[["line11"]]+ggplot2::ylab("scores x = y")+ggplot2::xlab("")+ggplot2::theme(legend.position="n
               ggplot2::ggtitle(""),
pl[["line1-1"]]+ggplot2::ylab("scores x = -y")+ggplot2::theme(legend.position="none")+
          ggplot2::ggtitle(""),
        layout_matrix = layout, widths = c(3,2),
        top = "mytitle", left ="", right =  "")
```

plot_sample_scores_cdt

PRC diagram of treatment and plot lines or points without species loading

Description

`plot_sample_scores_cdt` creates a PRC diagram of the treatments without species loadings.

Usage

```
plot_sample_scores_cdt(
  object,
  treatment = NULL,
  condition = NULL,
  plot = 1,
  xvals = NULL,
  score_name = "PRC1",
  size = c(2, 2),
  symbols_on_curves = FALSE,
  verbose = TRUE
)
```

Arguments

`object` a result of `doPRC` or `PRC_scores` or a list with element PRCplus (a data frame with a variables to plot, see details).

`treatment` character vector of names of factors in `object$PRCplus`, the first two of which set the `color` and `linetype` of treatments. Default: derived from `object$'focal_and_conditioning_f`

`condition` character vector of names of factors in `object$PRCplus`, the first one being used as x-axis. Default: derived from `object$'focal_and_conditioning_factors'`.

`plot` 0 or 1 (no/yes individual plot points) or character name of a factor in `object$PRCplus` indicating the plots (microcosms, subjects or objects) in a longitudinal study so as to plot lines. Default: 1 yielding points instead of lines.

xvals numeric values for the levels of the condition factor, or name of numeric variable in object$PRCplus to be used for the x-axis. Default: NULL for which the values are obtained using the function fvalues4levels applied to object$PRCplus[[condition[1]]]

score_name name of scores to use as y-axis; default: "PRC1".

size size of symbols and symbols in the legend. default: c(2, 2). If the length is not 2 than the first entry is replicated.

symbols_on_curves
logical, draw symbols on the PRC curves at the measurement times (default FALSE). If TRUE, symbols are for the treatment, if there is one, and for the second treatment if there are two.

verbose logical for printing the current model, score_name, condition and treatment (default TRUE).

Details

plot_sample_scores_cdt can be used to plot any ordination score or variable against the condition with treatments indicated by colors and line type by setting score_name to the name of the variable of interest and argument plot to 0. If object is not a result of doPRC or PRC_scores), set treatment and condition (or object$'focal_and_conditioning_factors') and set object to a list with element PRCplus; object$PRCplus should contain the names indicated by the arguments treatment, condition and, optionally, plot

- treatment: a factor for the treatments,
- condition: a factor or variable condition

If plot!=0, the scores (values) for individual sample points or lines must be in the variable "PRC1_E" or, in general, paste(score_name,"_E", sep = "").

Value

a ggplot object

See Also

PRC_scores, get_focal_and_conditioning_factors, plotPRC and fvalues4levels

Examples

```
data(pyrifos, package = "vegan") #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                   dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)),
                     ditch = gl(12, 1, length=132))

mod_rda <- vegan::rda( pyrifos ~ dose:week + Condition(week),  data = Design)
Design_w_PRCs <- PRC_scores(mod_rda, focal_factor_name= "dose", referencelevel = "0",
                            rank = 2, data= Design, scaling ="ms", flip = FALSE)

plot_sample_scores_cdt(Design_w_PRCs, plot = "ditch")
plot_sample_scores_cdt(Design_w_PRCs)
plot_sample_scores_cdt(Design_w_PRCs, plot = "Ditch")
```

plot_species_scores_2d

Species ordination plot

Description

plot_species_scores_2d creates an ordination diagram of species with selection criterion of which species to plot with names.

Usage

```
plot_species_scores_2d(
  species_scores,
  threshold = 7,
  speciesname = NULL,
  scorenames = c("RDA1", "RDA2"),
  selectname = "Fratio2",
  label.repel = FALSE,
  withnames_only = FALSE,
  max.overlaps = 10,
  mult_expand = 0.1,
  verbose = TRUE
)
```

Arguments

species_scores
: a species-by-scores matrix, a data frame with rownames (species names) or a tibble with variable with name speciesname containing species names and a column or variabe with names scorenames containing the scores (default: c("RDA1","RDA2") being the species score names from rda.

threshold
: species with criterion (specified by selectname) higher than the threshold are displayed. Default: 5 (which is the threshold F-ratio for testing two regression coefficients (one for each axis) at p=0.01 with 60 df for the error in a multiple regression of each single species onto the condition and the ordination axes). If selectname is not in the data the threshold is divided by ten, so that the default is 0.5.

speciesname
: name of the variable containing the species names (default NULL uses rownames).

scorenames
: names of the two columns or variables containing the species scores to be plotted (default c("RDA1","RDA2")).

`selectname` name of the column or variable containing the criterion for the selection of species to be displayed Default: `"Fratio2"`; if `selectname` is not found in `species_scores`, set to the Euclidian norm (length) of the vectors defined by row-wise combining the values defined by `scorenames`.

`label.repel` logical (default: `FALSE`) for using labels in white boxes with borders using `geom_label_repel` instead of using plain text labels using `geom_text_repel`.

`withnames_only` logical to plot the species that exceed the `treshold` only, i.e. to *not* plot points indicating species that do not reach the `threshold`. (Default: `FALSE`).

`max.overlaps` exclude text labels that overlap too many things. Default: 10 (see `geom_text_repel`).

`mult_expand` fraction of range expansion of the diagram (default = 0.1). See `expansion`.

`verbose` logical for printing the number of species that `geom_text_repel` or `geom_label_repel` will try to name in the plot out of the total number of species (default: `TRUE`).

Details

The names of species are added using `geom_text_repel` or `geom_label_repel`. These functions may not be able to plot the names of all selected species. This issue can be diminished by increasing the value of `max.overlaps`.

Value

a ggplot object

See Also

`doPRC`, `plotPRC2d`

Examples

```
data(pyrifos, package = "vegan") #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                     dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)))

library(vegan)
mod_rda <- rda( pyrifos ~ dose:week + Condition(week),  data = Design)
species_scores= scores(mod_rda,choices =1:2, display="sp",scaling = 1)
library(PRC)
head(rdacca_taxon_stats(mod_rda)) # stats per species
get_focal_and_conditioning_factors(mod_rda) # factors from the model

species_scores= vegan::scores(mod_rda,choices =1:2, display="sp",scaling = 2)
plot_species_scores_bk(species_scores) # 1-d vertical plot of species scores (a cut of RDA1-scores at 0.5)
plot_species_scores_bk(species_scores, threshold = 14) #  a cut of RDA1-scores at 1.0 (as scoresname is unset)

plot_species_scores_2d(species_scores)
plot_species_scores_2d(species_scores, label.repel = TRUE)
plot_species_scores_2d(species_scores, max.overlaps = 30, label.repel = TRUE)
```

plot_species_scores_bk   *Vertical ggplot2 line plot of species loadings in a PRC*

Description

`plot_species_scores_bk` creates a vertical line plot of species scores for a PRC diagram with selection criterion for which species to plot with names.

Usage

```
plot_species_scores_bk(
  species_scores,
  ylab = latex2exp::TeX("$b_k$"),
  threshold = 7,
  y_lab_interval = 0.5,
  speciesname = NULL,
  scoresname = "RDA1",
  selectname = "Fratio1",
  verbose = TRUE
)
```

Arguments

`species_scores` a species-by-scores matrix, a data frame with rownames (species names) or a tibble with variable with name `speciesname` containing species names and a column or variabe with name `scoresname` containing the scores (default: `"RDA1"`), e.g. species scores from library `vegan`

`ylab` y-axis label. Default: $b_k$.

`threshold` species with criterion (specified by `selectname`) higher than the `threshold` are displayed. Default: 7 (which is the threshold F-ratio for testing a single regression coefficient at `p=0.01` with `60` df for the error in a multiple regression of each single species onto the condition and the ordination axis). If `selectname` is not in `species_scores`, the `threshold` is divided by `14`, so that the default is 0.5.

`y_lab_interval` interval of the y-axis ticks. A tick at no effect (0) is always included; default: 0.5.

`speciesname` name of the variable containing the species names (default `NULL` uses rownames)

`scoresname` name of the column or variable containing the species scores to be plotted (default `"RDA1"`)

`selectname` name of the column or variable containing the criterion for the selection of species to be displayed Default: `"Fratio1"`; if `selectname` is not found in `species_scores`, set to `scoresname`.

`verbose` logical for printing the number of species with names out of the total number (default: `TRUE`).

Details

The absolute value of the criterion values is taken before selection, so that also the species scores of the ordination can serve as a criterion (e.g. `selectname="RDA1"`).

Value

a ggplot object

See Also

`doPRC`, `plotPRC`

Examples

```
data(pyrifos, package = "vegan") #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                     dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)))

library(vegan)
mod_rda <- rda( pyrifos ~ dose:week + Condition(week),  data = Design)
species_scores= scores(mod_rda,choices =1:2, display="sp",scaling = 1)
library(PRC)
head(rdacca_taxon_stats(mod_rda)) # stats per species
get_focal_and_conditioning_factors(mod_rda) # factors from the model

species_scores= vegan::scores(mod_rda,choices =1:2, display="sp",scaling = 2)
plot_species_scores_bk(species_scores) # 1-d vertical plot of species scores (a cut of RDA1-scores at 0.5)
plot_species_scores_bk(species_scores, threshold = 14) #  a cut of RDA1-scores at 1.0 (as scoresname is unset)

plot_species_scores_2d(species_scores)
plot_species_scores_2d(species_scores, label.repel = TRUE)
plot_species_scores_2d(species_scores, max.overlaps = 30, label.repel = TRUE)
```

PRC

PRC creates 1d and 2d PRC-diagrams using Principal Response Curves analysis (van den Brink & ter Braak 1999). The main user functions are `PRC_scores`, `plotPRC` *and* `plotPRC2d`. *The easiest example is the code demo* `PRC_pyrifos`.

Description

PRC creates 1d and 2d PRC-diagrams using Principal Response Curves analysis (van den Brink & ter Braak 1999). The main user functions are `PRC_scores`, `plotPRC` and `plotPRC2d`. The easiest example is the code demo `PRC_pyrifos`.

References

ter Braak (2023) Redundancy analysis includes analysis of variance-simultaneous component analysis (ASCA) and outperforms its extensions Chemometrics and Intelligent Laboratory Systems https://doi.org/10.1016/j.chemolab.2023.104898

van den Brink, P.J. & ter Braak, C.J.F. (1999) Principal Response Curves: Analysis of time-dependent multivariate responses of a biological community to stress. Environmental Toxicology and Chemistry, 18, 138-148. https://doi.org/10.1002/etc.5620180207

PRC_scores *Calculate principal response curve (PRC) scores from vegan::rda, cca, dbrda or capscale objects*

Description

PRC_scores calculates scores to display principal response curves with scores for individual plots from a vegan constrained ordination object (cca.object).

Usage

```
PRC_scores(
  object,
  focal_factor_name = NULL,
  referencelevel = NULL,
  rank = 2,
  flip = rep(FALSE, rank),
  scaling = "ms",
  data
)
```

Arguments

object
: a vegan cca.object, a result of rda, cca, dbrda or capscale with model specified via a formula.

focal_factor_name
: name of one or two focal factors (the treatment(s) of interest); if unspecified, derived from the formula in object via the function get_focal_and_conditioning_factors.

referencelevel
: numeric or character level(s) to be used as reference of the focal_factor_name(s); default: 1.

rank
: number of axes to be computed; default 2.

flip
: logical value or vector; default: FALSE. Should the axes be flipped, i.e. reversed in orientation? flip can be numeric with values -1 and 1 for reverse and no reverse, respectively, i.e. multiply the scores by either -1 or 1.

`scaling` character `c("ms", "ss")`. Scales the axes of an RDA object to mean square species scores of 1 (`"ms"`) or a sum of squares of 1 (`"ss"`). The only scaling for a CCA object is `"ss"` (weighted mean square = 1, with weight = species total abundance). For a dbrda object, there are no species scores and the scaling is like "ss". The default scaling (`"ms"` for RDA and `"ss"` for CCA) has the advantage that the PRC treatment and sample scores measure the effect sizes of the 'average' species as their (weigthed) mean square is 1 (sd = 1); it is the scaling used in Canoco5 (www.canoco5.com). Scaling `"ss"` is the default in `prcomp` and used in ter Braak (2023).

`data` data frame of the experimental design used in the call to `rda` or `cca`

Details

`PRC_scores` uses `scores` to calculate the loadings (species scores),if obtainable from the `object`, and the constrained and unconstrained sample scores of the units in `data`. The PRC treatment and sample scores are obtained by subtracting the reference scores from the constrained and unconstrained scores, respectively. The reference scores are calculated by applying `predict` with `type = "lc"` on `newdata`, which equals `data`, except that the level of the focal factor of each unit is changed to the reference level. By this simple method, `prc scores` can be applied with and without terms in `Condition()` in the `formula` of the call to `rda`, `cca`, `dbrda` or `capscale`. There can thus be additional (perhaps quantitative) covariates, `A:B+ Condition(A+other_covariates)` where the focal factor is B, and also with three (instead of two factors) e.g. with formula `A*B*C + Condition(A*C)` so as to display the A*C-dependent effects of focal factor B on the multivariate response in the call to `rda`, `cca`, `dbrda` or `capscale`. The element `object$'focal_and_conditioning_factors'` in value is set by applying the function `get_focal_and_conditioning_factors` to the ordination model specified in `object`.

CCA-based PRC is useful in data with many zeroes and which are strictly compositional, i.e. have row sums that have no meaning for the question of interest (ter Braak and te Beest, 2022). A layman's example is that the weight of a cake is not important for its taste; taste depends on its composition only. CCA can be applied without log-transforming the data and thereby avoids the issue of replacement of zeroes (ter Braak and te Beest, 2022).

Value

A named list:

- PRCplus: a data frame that combines the design in `data` with the PRC and RDA/CCA scores with ("_E") and without error (the unconstrained [sites] and constrained [LC] scores, respectively), and, finally, the reference scores. Note that PRC1_E = RDA1_E - RDA1_Ref, etc for other axes
- species: a matrix with the loadings with respect to the PRC scores; NULL if the `object` lacks loadings.
- reference_scores: a matrix with the scores of each unit in `data` with the level of the focal factor replaced by the reference level
- focal_factor_name: name of the focal factor in the call to `PRC_scores`
- referencelevel: referencelevel in the call to `PRC_scores`
- coefficients: list of PRC-coefficients for each axis (matrix if rank is equal to 1)

- focal_and_conditioning_factors: list of focal and conditioning factors; obtained from `get_focal_and_conditioning_f` if argument `focal_factor_name` in the call is unset.
- terms: object$terms. See `cca.object`.
- terminfo: object$terminfo. See `cca.object`.
- method: object$method. See `cca.object`.
- percExp: percentage variance of each axis out of the variance that can be explained by the predictors and covariates.

See Also

`doPRC`, `plotPRC`, `plotPRC2d`

Examples

```
data(pyrifos, package = "vegan") #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                     dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)),
                       ditch = gl(12, 1, length=132))

mod_rda <- vegan::rda( pyrifos ~ dose:week + Condition(week),  data = Design)
Design_w_PRCs <- PRC_scores(mod_rda, focal_factor_name= "dose", referencelevel = "0",
                            rank = 2, data= Design, scaling ="ms", flip = FALSE)

plot_sample_scores_cdt(Design_w_PRCs, plot = "ditch")
plot_sample_scores_cdt(Design_w_PRCs)
plot_sample_scores_cdt(Design_w_PRCs, plot = "Ditch")
```

`rdacca_taxon_stats` *Taxon fit statistics of an rda or cca ordination*

Description

`rdacca_taxon_stats` calculates fit statistics for taxa in an `rda` or `cca` ordination.

Usage

```
rdacca_taxon_stats(object, rank = 1)
```

Arguments

`object`	a result of `rda` or `cca`.
`rank`	number of axes for which fits are calculated (1:rank). Default: 1.

Details

The code uses ordiYbar and scores and also works for unconstrained ordination using rdaand cca. The software Canoco 5 (ter Braak & Smilauer, 2018) contains Cfit and a rescaled version of the column Var but not the columns Fratio.k.

Value

A matrix with taxa in rows and taxon specific statistics in columns:

Var sample variance of the taxon.

FitCE R squared: fraction of the sample variance explained by the predictors and covariates.

FitE fraction of the sample variance explained by the predictors after adjustment for the covariates.

Cfit*k* fraction of the sample variance explained by the first *k* axes (the axes are constrained in a constrained ordination and unconstrained otherwise).

Fratio*k* F-ratio of the regression model with *k* axes (as predictors) and the model without axes (i.e. with covariates or the intercept only).

References

ter Braak C.J.F. & Šmilauer P. (2018): Canoco reference manual and user’s guide: software for ordination, version 5.1x. Microcomputer Power, Ithaca, USA, 536 pp. (http::www.canoco5.com)

Examples

```
data(pyrifos, package = ”vegan”) #transformed species data from package vegan
#the chlorpyrifos experiment from van den Brink & ter Braak 1999
Design <- data.frame(week=gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24)),
                     dose=factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11)))

library(vegan)
mod_rda <- rda( pyrifos ~ dose:week + Condition(week),  data = Design)
species_scores= scores(mod_rda,choices =1:2, display=”sp”,scaling = 1)
library(PRC)
head(rdacca_taxon_stats(mod_rda)) # stats per species
get_focal_and_conditioning_factors(mod_rda) # factors from the model

species_scores= vegan::scores(mod_rda,choices =1:2, display=”sp”,scaling = 2)
plot_species_scores_bk(species_scores) # 1-d vertical plot of species scores (a cut of RDA1-scores at 0.5)
plot_species_scores_bk(species_scores, threshold = 14) #  a cut of RDA1-scores at 1.0 (as scoresname is unset)

plot_species_scores_2d(species_scores)
plot_species_scores_2d(species_scores, label.repel = TRUE)
plot_species_scores_2d(species_scores, max.overlaps = 30, label.repel = TRUE)
```

Index

arrangeGrob, *10, 11*, *13, 14*

biomonitoring, 2

capscale, *21, 22*
cca, *4, 5*, *7*, *21–24*
cca.object, *5*, *21*, *23*
CPTNUT_phytoplankton, 3

dbrda, *21, 22*
doPRC, 4, *7–12*, *14–16*, *18*, *20*, *23*

expansion, *13*, *18*

fvalues4levels, 6, *10*, *13*, *16*

geom_label_repel, *13*, *18*
geom_text_repel, *13*, *18*
get_focal_and_conditioning_factors, 7, *16*, *21–23*
grid.arrange, *10, 11*, *14*

ordiYbar, *24*
Ossenkampen, 8

plot_sample_scores_cdt, *11*, *13, 14*, 15
plot_species_scores_2d, *13, 14*, 17
plot_species_scores_bk, *11*, 19
plotPRC, 9, *16*, *20*, *23*
plotPRC2d, *11*, 12, *18*, *20*, *23*
PRC, 20
PRC_scores, *5*, *7–10*, *12*, *15, 16*, *20*, 21
prcomp, *5*, *22*
predict, *22*

rda, *4, 5*, *7*, *17*, *21–24*
rdacca_taxon_stats, 23

scores, *22*, *24*