Package ‘AlleleShift’

August 14, 2021

Type  Package
Title  Predict and Visualize Population-Level Changes in Allele Frequencies in Response to Climate Change
Version  1.0-2
Date  2021-8-9
Author  Roeland Kindt [cre, aut] (<https://orcid.org/0000-0002-7672-0712>)
Maintainer  Roeland Kindt <R.KINDT@CGIAR.ORG>
Description  Methods (Kindt, R. 2021. <doi:10.7717/peerj.11534>) are provided of calibrating and predicting shifts in allele frequencies through redundancy analysis ('vegan::rda()') and generalized additive models ('mgcv::gam()'). Visualization functions for predicted changes in allele frequencies include 'shift.dot.ggplot()', 'shift.pie.ggplot()', 'shift.moon.ggplot()', 'shift.waffle.ggplot()' and 'shift.surf.ggplot()' that are made input data sets that are prepared by helper functions for each visualization method. Examples in the documentation show how to prepare animated climate change graphics through a time series with the 'gganimate' package. Function 'amova.rda()' shows how Analysis of Molecular Variance can be directly conducted with the results from redundancy analysis.
License  GPL-3
Imports  vegan (>= 2.5-6), BiodiversityR (>= 2.12-3), adegenet
Depends  R (>= 3.5.0)
Suggests  poppr, mgcv, dplyr, ggplot2, GGally, ggforce, ggrepel, ggsci, gggibbous, gganimate, patchwork
NeedsCompilation  no
Repository  CRAN
Date/Publication  2021-08-14 14:20:08 UTC

R topics documented:

amova.rda ................................................................. 2
count.model ........................................................... 4
Poptri.genind ......................................................... 8
population.shift .................................................... 9
Description

The main objective of the function is to illustrate how information on 'mean squares' can be extracted from redundancy analysis results. For balanced data sets, the final 'Phi' statistics are expected to be the same as those obtained with `poppr.amova`. The function can only handle 1-level (population) and 2-level (eg, region/population or population/subpopulation) hierarchies.

Usage

```r
amova.rda(x, x.data)
```

Arguments

- `x` Result of redundancy analysis fitted via `rda`.
- `x.data` Data used to fit the redundancy analysis via `ensemble.VIF.dataframe`.

Details

A similar analysis of the relationship between AMOVA and RDA is given by Kindt (2020), also exploring relationships with the Multivariate Analysis of Variance Using Distance Matrices methods that are available via `adonis2`. For discussions on AMOVA and its relationship with the matrix of squared Euclidean distances between individuals, see Michalakis and Excoffier (1996), Peakall et al. (1995) or Meirmans and Liu (2018) (the last article also shows how AMOVA can be expanded beyond haploid and diploid organisms). These authors provide the coefficients (equations 3, 4a and 4b) that need to be used with unbalanced numbers of individuals in populations and/or groups, which could then be used to obtain the exact AMOVA statistics from the estimated Mean Squares (there is no practical point in doing this as AMOVA is available via `poppr.amova`).

I have also cross-checked the results with GenAlEx 6.5 (Peakall and Smouse 2012), using the export function of `genind2genalex` to cross-check that RDA obtains the correct Sums-of-Squares and Mean-Squares. Important to note here is that the values in GenAlEx are obtained from only the upper part of the distance matrix, hence these are exactly 50 percent of the Sums-of-Squares from RDA.

Note that for diploid organisms, the option here is not to include the hierarchical level within individuals.
The function proceeds with Analysis of Molecular Variance from data generated by redundancy analysis (RDA).

**Author(s)**

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

**References**

Kindt R. 2020. Analysis of Molecular Variance (AMOVA) with vegan and BiodiversityR, including a graphical method to identify potential migrants. [https://rpubs.com/Roeland-KINDT](https://rpubs.com/Roeland-KINDT)


**Examples**

```r
## Not run:
library(poppr)
library(BiodiversityR) # also loads vegan

# Example with 1 level
data(nancycats)
names(alleles(nancycats)) <- locNames(nancycats) # small bug in this data set
nancycats2 <- missingno(nancycats, type = "loci", cutoff=0)
nancy.dist <- vegdist(nancycats2@tab, method="euc")

# Same method as in GenAlEx via the squared Euclidean distance
nancy.dist.amova <- poppr.amova(nancycats2, ~Population, within=FALSE,
                               dist=nancy.dist, squared=FALSE)
nancy.dist.amova

# Via vegan::rda
library(vegan)
nancycats.rda.data <- data.frame(Population=nancycats2@pop)
nancycats.rda.result <- rda(nancycats2@tab ~ Population, data=nancycats.rda.data)
nancycats.rda.result
amova.rda(nancycats.rda.result, nancycats.rda.data)
```
# Example with 2 levels
# Same example as for poppr::poppr.amova

data(Aeut)
strata(Aeut) <- other(Aeut)$population_hierarchy[-1]
agc <- as.genclone(Aeut)

amova.result <- poppr.amova(agc, ~Pop/Subpop, within=FALSE)
amova.result

agc.rda.data <- data.frame(as.matrix(agc$other$population_hierarchy))
agc.rda.data[, 1] <- as.factor(agc.rda.data[, 1])
agc.rda.data[, 2] <- as.factor(agc.rda.data[, 2])
agc.rda.data[, 3] <- as.factor(agc.rda.data[, 3])
str(agc.rda.data)

agc.rda.result <- rda(agc@tab ~ Pop + Pop_Subpop, data=agc.rda.data)
agc.rda.result
amova.rda(agc.rda.result, agc.rda.data)

# Example with a balanced data set
# library(BiodiversityR)
data(warcom)
data(warenv)

warburgia.genind <- genind(warcom)
warburgia.genind@strata <- warenv

poppr.amova(warburgia.genind, ~ population, within=FALSE)

warburgia.rda <- rda(warcom ~ population, data=warenv)
warburgia.rda
amova.rda(warburgia.rda, warenv)

## End(Not run)

count.model  

*Calibration and Prediction Functions for Changes in Allele Frequencies*

**Description**

The functions implement a two-step calibration and prediction process for allele frequencies, whereby the second calibration step uses the predictions of the first step.

**Usage**

```r
count.model(genpop.data, env.data, permutations = 99, 
             ordistep = FALSE, cca.model = FALSE)
```
count.model

count.pred(count.model, env.data)

freq.model(count.predicted)

count.predicted

freq.predicted

freq.ggplot(freq.predicted, plot.best = TRUE, threshold = 0.50, colour.Pop = TRUE, manual.colour.values = NULL, xlim = c(0.0, 1.0), ylim = c(0.0, 1.0))

Arguments

genpop.data Data in the format of genpop.
env.data Descriptors of (bio-)climatic conditions for the populations, either for the baseline climate (to check calibrations) or future/past climates. It is expected that these are in the same sequence as the populations in the genpop.data.
permutations Number of permutations as in anova.cca.
ordistep Check the results of ordistep.
cca.model Fit a CCA model with the minor allele frequency as in Blumstein et al. 2020.
count.model Model fitted by count.model.
count.predicted Predictions made by count.pred, expected to be predicted for the baseline climate as a second step in the calibration process.
freq.model Model fitted by freq.model.
freq.predicted Allele frequencies predicted by freq.pred.
plot.best Plot the models with highest R2 (TRUE) or lowest R2 values (FALSE).
threshold Threshold used to distinguish between the best and worst model.
colour.Pop Colour populations differently (TRUE), or alleles differently (FALSE)
manual.colour.values Manual specifications for colour values
xlim limits of the x-axis
ylim limits of the y-axis

Details

These functions allow for an almost completely alternative workflow of predicting shifts in allele frequencies under climate change than the protocol developed by Blumstein et al. 2020. The methodology available here calibrates and predicts changes in allele frequencies via redundancy analysis (Blumstein et al. use canonical correspondence analysis) that were calibrated from allele counts (Blumstein et al. use allele frequencies) for all alleles (Blumstein et al. only calibrate the minor alleles) in a two-step calibration process (the second step via freq.model and freq.predict to ensure that predicted alleles are in the range of 0-1 is not included in protocol developed by Blumstein et al.). Other key differences in the methodology are that explanatory variables are expected to be bioclimatic variables (and not principal components as in Blumstein et al.) and that
the input data is expected to be in the `genpop` format. Although a method to reduce the number of explanatory variables via `ordistep` is shown, I advise against reducing the explanatory variables as this likely will reduce the explanatory power of the models, whereas explanatory power is the major objective of calibrating these functions. Motivations for the differences in methodologies of Blumstein et al. and the ones available in AlleleShift are explained by Kindt 2021.

Confidence intervals are calculated via `qt` with `p=0.95` and `df=np` (number of populations), although the GAM is fitted only once for all the alleles and all the populations (that was my choice to reduce overfitting the baseline data).

The 'darkolivegreen4'-coloured reference lines shown via `freq.ggplot` correspond to a 1:1 relationship (full line) and 1:0.95 and 1:1.05 relationships (dashed lines).

**Value**

The functions enable calibration and prediction of allele frequencies.

**Author(s)**

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

**References**


**See Also**

`VIF.subset`, `population.shift`, `amova.rda`

**Examples**

```r
# A typical work flow would consist of:

# 1. Reduce the number of explanatory variables

data(Poptri.baseline.env)
data(Poptri.future.env)

VIF.select <- VIF.subset(Poptri.baseline.env, 
    keep=c("MAT", "CMI"), 
    cor.plot=TRUE)

VIF.select$VIF$vars.included

baseline.env <- Poptri.baseline.env[, VIF.select$VIF$vars.included]
save(baseline.env)
```
future.env <- Poptri.future.env[, VIF.select$VIF$vars.included]

# 2. Create the genpop object

data(Poptri.genind)
Poptri.genpop <- adegenet::genind2genpop(Poptri.genind)

# Get to know the populations and the alleles
poppr::poppr(Poptri.genind)
adegenet::makefreq(Poptri.genpop)

# 3. Calibrate the models

# Note that the ordistep procedure is not needed
# CCA model only to compare results with those by Blumstein et al. 2020
Poptri.count.model <- count.model(Poptri.genpop,
                                   env.data=baseline.env,
                                   ordistep=TRUE,
                                   cca.model=TRUE)

Poptri.pred.baseline <- count.pred(Poptri.count.model, env.data=baseline.env)
head(Poptri.pred.baseline)

Poptri.freq.model <- freq.model(Poptri.pred.baseline)
Poptri.freq.baseline <- freq.pred(Poptri.freq.model,
                                   count.predicted=Poptri.pred.baseline)

head(Poptri.freq.baseline)

# 4. Check how well the models predict baseline allele frequencies

# Populations are split in those with R2 > 0.50 and those with R2 < 0.50
# Better populations
plotA1 <- freq.ggplot(Poptri.freq.baseline,
                      plot.best=TRUE,
                      ylim=c(0.0, 0.8))
plotA1

# Populations with low R2
manual.colour.values1 <- ggsci::pal_npg()(5)
plotB1 <- freq.ggplot(Poptri.freq.baseline,
                      plot.best=FALSE,
                      manual.colour.values=manual.colour.values1,
                      xlim=c(0, 0.5),
                      ylim=c(0, 0.25))
plotB1

# Colouring by alleles
plotA2 <- freq.ggplot(Poptri.freq.baseline,
colour.Pop=FALSE, plot.best=TRUE, ylim=c(0, 0.8), manual.colour.values=manual.colour.values1)

plotA2

plotB2 <- freq.ggplot(Poptri.freq.baseline, colour.Pop=FALSE, plot.best=FALSE, manual.colour.values=manual.colour.values1, xlim=c(0, 0.5), ylim=c(0, 0.25))

plotB2

# 5. Predict future allele frequencies
Poptri.pred.future <- count.pred(Poptri.count.model, env.data=future.env)
head(Poptri.pred.future)

Poptri.freq.future <- freq.pred(Poptri.freq.model, count.predicted=Poptri.pred.future)

# The key results are variables 'Allele.freq' representing the baseline allele frequencies
# and variables 'Freq.e2', the predicted frequency for the future/past climate.
# Variable 'Freq.e1' is the predicted allele frequency in step 1
head(Poptri.freq.future)

# 5. Visualize the changes
# See functions shift.dot.ggplot, shift.pie.ggplot, shift.moon.ggplot,
# shift.waffle.ggplot and shift.surf.ggplot

---

**Poptri.genind**

*Example data sets for Populus trichocarpa.*

**Description**

These data sets include the same data that were used by Blumstein et al. 2020 to document their protocol. Genomic data has been processed further into the **genind** format. Data sets Poptri.freq.baseline, Poptri.freq.future and Poptri.1985to2085 were obtained with the calibration and prediction methods of this package. They are included to make the documentation of the various plotting methods shorter, but also show the types of results someone can obtain.

**Usage**

data(Poptri.genind)
population.shift

data(Poptri.baseline.env)
data(Poptri.future.env)
data(Poptri.loc)
data(Poptri.freq.baseline)
data(Poptri.freq.future)
data(Poptri.1985to2085)

References


Examples

data(Poptri.genind)

population.shift  

Shifts of Populations in Environmental Space as Response to Climate Change

Description

The function plots the locations of each population in baseline and future climates. Arrows indicate the shifts in positions of the populations.

Usage

population.shift(baseline.env.data, future.env.data, option=c("PCA", "RDA"), vector.multiply=1)

environmental.novel(baseline.env.data, future.env.data)

Arguments

baseline.env.data  
Baseline (bio-)climatic conditions for the populations.

future.env.data  
Changed (bio-)climatic conditions in future/past for the populations.

option  
Should an explanatory variable corresponding to the climate period be used by rda.

vector.multiply  
Multiplier for vector scores in the ordination diagrams.
Details

See Kindt (2020) for alternative methods of generating ordination diagrams via vegan, BiodiversityR and ggplot2.

Function `environmental.novel` identifies populations with future (or past) environmental conditions that are outside the baseline range. The function further calculates the probability of observing the future condition via `pnorm` with the mean and standard deviation from the baseline conditions. Where one or several variables are outside the baseline range, data are provided for the variable with the smallest probability.

Value

The main function generates an ordination diagram that depicts changes between baseline and future/past conditions for the populations.

Author(s)

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

References


Examples

data(Poptri.baseline.env)
data(Poptri.future.env)

environmental.novel(Poptri.baseline.env, Poptri.future.env)
# as if for past climates
environmental.novel(Poptri.future.env, Poptri.baseline.env)

VIF.select <- VIF.subset(Poptri.baseline.env, keep=c("MAT", "CMI"), cor.plot=FALSE)

VIF.select$vars.included

baseline.env <- Poptri.baseline.env[, VIF.select$vars.included]
future.env <- Poptri.future.env[, VIF.select$vars.included]

environmental.novel(baseline.env, future.env)

plotA <- population.shift(baseline.env, future.env, option="PCA")
plotA
shift.dot.ggplot

```r
plotB <- population.shift(baseline.env,
                          future.env,
                          option="RDA")
plotB
```

shift.dot.ggplot  
*Shifts of Allele Frequencies as Response to Climate Change*

**Description**

The function shows changes in allele frequencies between the baseline and future/past climate.

**Usage**

```r
shift.dot.ggplot(freq.future,
                 mean.change = FALSE, change.FUN = stats::median,
                 baseline.colour = "black",
                 future.colour = "dodgerblue3",
                 manual.colour.values=c("firebrick3", "chartreuse4"))
```

**Arguments**

- `freq.future`: Result from `AlleleShift::predFreq`
- `mean.change`: Aggregate changes among alleles.
- `change.FUN`: Function used the aggregate changes.
- `baseline.colour`, `future.colour`, `manual.colour.values`: Colours to be used in the plots.

**Value**

The function generates a ggplot that depicts changes between baseline and future/past allele frequencies of the populations.

**Author(s)**

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

**See Also**

`shift.pie.ggplot`, `shift.moon.ggplot`, `shift.waffle.ggplot`, `shift.surf.ggplot`
Examples

# The data can be obtained via the count.model and freq.model calibrations.
# These procedures are not repeated here.
data(Poptri.freq.future)

ggdot1 <- shift.dot.ggplot(Poptri.freq.future)
ggdot1

# create an animation

## Not run:
library(ggplot2)
library(gganimate)
library(gifski)

# The data is an interpolation and extrapolation between the baseline and future climate.
# For actual application, interpolate between climate data from available sources

data(Poptri.1985to2085)

ggdot.all <- ggplot(data=Poptri.1985to2085, group=Decade) +
  scale_y_continuous(limits=c(-0.1, 1.1),
  breaks=c(0.0, 0.25, 0.5, 0.75, 1.0)) +
  geom_errorbar(aes(x=Pop, ymin=LCL, ymax=UCL),
  colour="grey30", width=0.8, show.legend=FALSE) +
  geom_segment(aes(x=Pop, y=Allele.freq, xend=Pop, yend=Freq.e2, colour=increasing),
  size=1.2) +
  geom_point(aes(x=Pop, y=Allele.freq),
  colour="black", size=10, alpha=0.7) +
  geom_point(aes(x=Pop, y=Freq.e2),
  colour="dodgerblue3", size=10, alpha=0.7) +
  coord_flip() +
  xlab(element_blank()) +
  ylab("Allele frequencies") +
  theme(panel.grid.minor = element_blank()) +
  labs(colour="Future change in allele frequencies") +
  scale_colour_manual(values=c("firebrick3", "chartreuse4"),
  labels=c("decreasing", "increasing")) +
  theme(axis.text.x=element_text(angle=90, vjust=0.5, size=10)) +
  theme(legend.position="top") +
  facet_grid( ~ Allele, scales="free")

ggdot.all

ggdot.anim <- ggdot.all +
  transition_states(as.factor(Decade), transition_length = 10, state_length = 100) +
  labs(title = "Decade: {closest_state}s")

ggdot.anim2 <- animate(ggdot.anim, fps=5, width=1280, height=720)
getwd()

amin_save(filename="Allele shift animation.gif", animation=ggdot.anim2)

## End(Not run)

---

**shift.moon.ggplot**  
Shifts of Allele Frequencies as Response to Climate Change

**Description**

The function shows changes in allele frequencies between the baseline and future/past climate.

**Usage**

```r
shift.moon.ggplot(baseline.moon, future.moon,
manual.colour.values = c("white", "grey", "firebrick3", "chartreuse4"),
manual.colour.codes = c("A baseline ", "B", "A decreasing", "A increasing"))

moon.waxer(freq.in, sort.index= "Pop.index",
mean.change = FALSE, change.FUN = stats::median,
freq.focus = "Allele.freq",
ypos = 0, right = TRUE)
```

**Arguments**

- `baseline.moon, future.moon`  
  Result from `AlleleShift::moon.waxer`.
- `manual.colour.values`  
  Colours to be used in the plot.
- `manual.colour.codes`  
  Sequence for the manual colour values.
- `freq.in`  
  Result from `AlleleShift::predFreq`.
- `sort.index`  
  Sequence of the populations in the plot.
- `mean.change`  
  Aggregate changes among alleles.
- `change.FUN`  
  Function used the aggregate changes.
- `freq.focus`  
  Allele frequency for which to calculate statistics, either 'Allele.freq' or 'Freq.e2'.
- `ypos, right`  
  Arguments used for plotting, mainly as in `geom_moon`.

**Value**

The function generates a ggplot that depicts changes between baseline and future/past allele frequencies of the populations.
shift.pie.ggplot

**Author(s)**
Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

**See Also**
shift.dot.ggplot, shift.pie.ggplot, shift.waffle.ggplot, shift.surf.ggplot

**Examples**

```r
# The data can be obtained via the count.model and freq.model calibrations. # These procedures are not repeated here.
data(Poptri.freq.baseline)
data(Poptri.freq.future)

Poptri.baseline.moon <- moon.waxer(Poptri.freq.baseline, 
sort.index="Latitude.index")

Poptri.future.moon <- moon.waxer(Poptri.freq.future, 
sort.index="Latitude.index",  
freq.focus="Freq.e2", 
ypos=1)

ggmoon1 <- shift.moon.ggplot(Poptri.baseline.moon, 
    Poptri.future.moon)

ggmoon1
```

---

**shift.pie.ggplot**  
*Shifts of Allele Frequencies as Response to Climate Change*

**Description**

The function shows changes in allele frequencies between the baseline and future/past climate.

**Usage**

```r
shift.pie.ggplot(baseline.pie, future.pie, 
manual.colour.values = c("black", "grey", "firebrick3", "chartreuse4"),
manual.colour.codes = c("A baseline ", "B", "A decreasing", "A increasing")

pie.baker(freq.in, sort.index= "Pop.index", 
mean.change = FALSE, change.FUN = stats::median, 
freq.focus = "Allele.freq", ypos = 0, 
r0 = 0.1, r = 0.5, focus = 0.2 )
```
Arguments

- baseline.pie, future.pie
  - Result from AlleleShift::pie.baker.
- manual.colour.values
  - Colours to be used in the plot.
- manual.colour.codes
  - Sequence for the manual colour values.
- freq.in
  - Result from AlleleShift::predFreq.
- sort.index
  - Sequence of the populations in the plot.
- mean.change
  - Aggregate changes among alleles.
- change.FUN
  - Function used the aggregate changes.
- freq.focus
  - Allele frequency for which to calculate statistics, either 'Allele.freq' or 'Freq.e2'.
- ypos, r0, r, focus
  - Arguments used for plotting, mainly as in `geom.arc.bar`

Value

The function generates a ggplot that depicts changes between baseline and future/past allele frequencies of the populations.

Author(s)

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

See Also

- `shift.dot.ggplot`, `shift.moon.ggplot`, `shift.waffle.ggplot`, `shift.surf.ggplot`

Examples

```r
# The data can be obtained via the count.model and freq.model calibrations.
# These procedures are not repeated here.
data(Poptri.freq.baseline)
data(Poptri.freq.future)

Poptri.baseline.pie <- pie.baker(Poptri.freq.baseline, r0=0.1,
                                  sort.index="Latitude.index")

Poptri.future.pie <- pie.baker(Poptri.freq.future, r0=0.1,
                                freq.focus="Freq.e2",
                                sort.index="Latitude.index",
                                ypos=1)

ggpie! <- shift.pie.ggplot(Poptri.baseline.pie,
                           Poptri.future.pie)
ggpie!
```
# create an animation

## Not run:

```r
library(ggplot2)
library(ggforce)
library(gganimate)
library(gifski)
library(transformr)
```

# The data is an interpolation and extrapolation between the baseline and future climate.
# For actual application, interpolate between climate data from available sources

```r
data(Poptri.1985to2085)
decades <- sort(unique(Poptri.1985to2085$Decade))

for (d in 1:length(decades)) {
  decade.focal <- decades[d]
decade.data <- Poptri.1985to2085[Poptri.1985to2085$Decade == decade.focal, ]
decade.pie <- pie.baker(decade.data, r0=0.1,
  freq.focus="Freq.e2",
  sort.index="Latitude.index",
  ypos=1)
decade.pie <- cbind(Decade=rep(decade.focal, nrow(decade.pie)), decade.pie)
  if (d == 1) {
    future.pies <- decade.pie
  }else{
    future.pies <- rbind(future.pies, decade.pie)
  }
}

np <- length(unique(Poptri.baseline.pie$Pop))
manual.colour.values <- c("black", "grey", "firebrick3", "chartreuse4")

```r
ggpie.all <- ggplot(data=future.pies, group=Decade) +
  scale_x_continuous(limits=c(0.5, np+0.5),
    breaks=seq(from=1, to=np, by=1),
    labels=levels(Poptri.baseline.pie$Pop)) +
  geom_arc_bar(data=Poptri.baseline.pie,
    aes(x0=sort.index, y0=ypos, r0=r0, r=0.4,
      start=start, end=end, fill=colour),
    size=0.04, alpha=1, colour="snow1") +
  geom_arc_bar(data=future.pies,
    aes(x0=sort.index, y0=ypos, r0=r0, r=0.4,
      start=start, end=end, fill=change.colour),
    size=0.04, alpha=1, colour="snow1") +
  geom_point(data=subset(future.pies, increasing==TRUE),
    aes(x=sort.index, y=ypos),
    size=5, shape=21, fill=manual.colour.values[4], stroke=0.03, show.legend=FALSE) +
  geom_point(data=subset(future.pies, increasing==FALSE),
    aes(x=sort.index, y=ypos),
    size=5, shape=21, fill=manual.colour.values[4], stroke=0.03, show.legend=FALSE) +
  geom_point(data=subset(future.pies, increasing==FALSE),
    aes(x=sort.index, y=ypos),
    size=5, shape=21, fill=manual.colour.values[4], stroke=0.03, show.legend=FALSE) +
```
Description

The function shows changes in allele frequencies between the baseline and future/past climate.

Usage

```r
shift.surf.ggplot(freq.future,
  Allele.focus=unique(freq.future$Allele)[1],
  freq.focus="Allele.freq",
  xcoord="LON", ycoord="LAT",
  mean.change = FALSE, change.FUN = stats::median,
  manual.colour.values = c("firebrick3", "chartreuse4"),
  ...)
```
Arguments

freq.future  Result from AlleleShift::predFreq.
freq.focus  Selection of the Allele.
Allele.focus  Selection of the frequency.
xcoord, ycoord  Geographical coordinates of the populations.
mean.change  Aggregate changes among alleles.
change.FUN  Function used the aggregate changes.
manual.colour.values  Colours to be used in the plot.
...  Options for ordisurf.

Details

Populations are plotted in geographical space via ordination plotting methods, which is suitable as fixed coordinate systems are recommended both in ordination diagrams and maps. See Kindt (2020) for alternative methods of generating ordination diagrams via vegan, BiodiversityR and ggplot2. Kindt (2021) shows how a STAMEN baseline map can be used to produce high resolution images via the ggmap package.

Value

The function generates a ggplot that depicts changes between baseline and future/past allele frequencies of the populations.

Author(s)

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

References


See Also

shift.dot.ggplot, shift.pie.ggplot, shift.moon.ggplot, shift.waffle.ggplot

Examples

# The data can be obtained via the count.model and freq.model calibrations.
# These procedures are not repeated here.
data(Poptri.freq.baseline)
data(Poptri.freq.future)
# Plots for the first allele
# Symbols and colours indicate future change (green, ^ = future increase)
# Symbol size reflects the frequency in the climate shown

# Baseline climate
plotA <- shift.surf.ggplot(Poptri.freq.future,
               xcoord="Long", ycoord="Lat",
               Allele.focus=unique(Poptri.freq.future$Allele)[1],
               freq.focus="Allele.freq")

plotA

# Future/past climate
plotB <- shift.surf.ggplot(Poptri.freq.future,
               xcoord="Long", ycoord="Lat",
               Allele.focus=unique(Poptri.freq.future$Allele)[1],
               freq.focus="Freq.e2")

plotB

# Plots for the fifth allele

# Baseline climate
plotC <- shift.surf.ggplot(Poptri.freq.future,
               xcoord="Long", ycoord="Lat",
               Allele.focus=unique(Poptri.freq.future$Allele)[5],
               freq.focus="Allele.freq")

plotC

# Future climate
plotD <- shift.surf.ggplot(Poptri.freq.future,
               xcoord="Long", ycoord="Lat",
               Allele.focus=unique(Poptri.freq.future$Allele)[5],
               freq.focus="Freq.e2")

plotD

# create an animation

## Not run:
library(ggplot2)
library(ggforce)
library(gganimate)
library(gifski)
library(transformr)

# The data is an interpolation and extrapolation between the baseline and future climate.
# For actual application, interpolate between climate data from available sources

data(Poptri.1985to2085)

Poptri.1985to2085$xcoord <- Poptri.1985to2085$Long
Poptri.1985to2085$ycoord <- Poptri.1985to2085$Lat

alleles <- sort(unique(as.character(Poptri.1985to2085$Allele)))
future.data <- Poptri.1985to2085[Poptri.1985to2085$Allele == alleles[1],]

decades <- sort(unique(future.data$Decade))

for (d in 1:length(decades)) {
  decade.focal <- decades[d]
  decade.data <- future.data[future.data$Decade == decade.focal,]
  plotLONLAT <- vegan::ordiplot(decade.data[, c("xcoord", "ycoord")])
  surfAllele <- BiodiversityR::ordisurfgrid.long(vegan::ordisurf(plotLONLAT, y=decade.data$Freq.e2))
  decade.surf <- cbind(Decade=rep(decade.focal, nrow(surfAllele)), surfAllele)
  if (d == 1) {
    future.surfs <- decade.surf
  } else {
    future.surfs <- rbind(future.surfs, decade.surf)
  }
}

# The function above will not be able to predict far into the future.
# The results obtained (future.surfs) can still be used for plotting.

ggsurf.all <- ggplot(data=future.surfs, group=Decade) +
  geom_contour_filled(aes(x=x, y=y, z=z),
                      breaks=seq(from=0.0, to=1.05, by=0.05)) +
  geom_point(data=subset(future.data, Decade==Decade),
             aes(x=xcoord, y=ycoord, size=Freq.e2, shape=increasing),
             colour="red", alpha=0.8, stroke=1.5, show.legend=FALSE) +
  xlab(element_blank()) +
  ylab(element_blank()) +
  labs(fill=alleles[1]) +
  scale_fill_viridis_d() +
  scale_colour_manual(values=c("firebrick3", "chartreuse4"),
                      guide=FALSE) +
  scale_size_area(max_size=6) +
  scale_shape_manual(values=c(6, 2)) +
  theme(panel.grid = element_blank()) +
  theme(axis.text= element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(legend.title = element_text(size=9)) +
  theme(legend.text = element_text(size=8)) +
  coord_fixed()

ggsurf.all

ggsurf.anim <- ggsurf.all +
  transition_states(as.factor(Decade), transition_length = 10, state_length = 100) +
  labs(title = "Decade: {closest_state}s")

ggsurf.anim2 <- animate(ggsurf.anim, fps=5, width=1280, height=720)

getwd()
anim_save(filename="Allele shift surf animation.gif", animation=ggsurf.anim2)
Description

The function shows changes in allele frequencies between the baseline and future/past climate.

Usage

```r
shift.waffle.ggplot(future.waffle,
manual.colour.values = c("black", "grey", "firebrick3", "chartreuse4"),
manual.colour.codes = c("A baseline ", "B", "A decreasing", "A increasing"))
```

```r
waffle.baker(freq.in, sort.index = "Pop.index",
mean.change = FALSE, change.FUN = stats::median)
```

Arguments

- `future.waffle`: Result from `AlleleShift::waffle.baker`.
- `manual.colour.values`: Colours to be used in the plot.
- `manual.colour.codes`: Sequence for the manual colour values.
- `freq.in`: Result from `AlleleShift::predFreq`.
- `sort.index`: Sequence of the populations in the plot.
- `mean.change`: Aggregate changes among alleles.
- `change.FUN`: Function used the aggregate changes.

Details

Although a package `ggwaffle` exists, I opted to bake my own waffles (possibly the Belgian in me.). As a separate row is created for each square/rectangle of the waffles, the resulting data is quite large. Hence trying to animate this is probably a bad idea (unless you want to make some real waffles while your computer is busy).

Value

The function generates a `ggplot` that depicts changes between baseline and future/past allele frequencies of the populations.
Author(s)
Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

Examples

```r
# The data can be obtained via the count.model and freq.model calibrations.
# These procedures are not repeated here.
data(Poptri.freq.baseline)
data(Poptri.freq.future)

Poptri.future.waffle <- waffle.baker(Poptri.freq.future,
  sort.index="Latitude.index")

ggwaffle1 <- shift.waffle.ggplot(Poptri.future.waffle)
ggwaffle1
```

VIF.subset

Reduce the Climate Data based on Variance Inflation Factor Analysis.

Description

Through Variance Inflation Factor (VIF) analysis, a subset of variables is identified where all variables have VIF below a predefined threshold.

Usage

```r
VIF.subset(data, VIF.max, keep=NULL, silent=FALSE, cor.plot=TRUE)
```

Arguments

- `data` (Bio)-Climatic or environmental descriptors of the populations
- `VIF.max` Maximum Variance Inflation Factor as in `ensemble.VIF.dataframe` that is used internally.
- `keep` Variables to keep as in as in `ensemble.VIF.dataframe`.
- `silent` Limit the verbose output as in `ensemble.VIF.dataframe`.
- `cor.plot` Generate a correlation matrix for the final subset via `ggcorr`.

Value

The function returns information on a subset of variables where VIF is below a pre-defined threshold.
**Author(s)**

Roeland Kindt (World Agroforestry, CIFOR-ICRAF)

**Examples**

data(Poptri.baseline.env)

# error as many variables are highly correlated
# VIF.subset(Poptri.baseline.env)

VIF.subset(Poptri.baseline.env, keep=c("MAT", "CMI"), cor.plot=TRUE)
Index

* datasets
  Poptri.genind, 8
  adonis2, 2
  amova.rda, 2, 6
  anova.cca, 5
  count.model, 4
  count.pred (count.model), 4
  ensemble.VIF.dataframe, 2, 22
  environmental.novel (population.shift), 9
  freq.ggplot (count.model), 4
  freq.model (count.model), 4
  freq.pred (count.model), 4
  genind, 8
  genind2genalex, 2
  genpop, 5, 6
  geom.arc_bar, 15
  geom_moon, 13
  ggcorr, 22
  moon.waxer (shift.moon.ggplot), 13
  ordistep, 5, 6
  ordisurf, 18
  pie.baker (shift.pie.ggplot), 14
  pnorm, 10
  poppr.amova, 2
  Poptri.1985to2085 (Poptri.genind), 8
  Poptri.baseline.env (Poptri.genind), 8
  Poptri.freq.baseLINE (Poptri.genind), 8
  Poptri.freq.future (Poptri.genind), 8
  Poptri.future.env (Poptri.genind), 8
  Poptri.genind, 8
  Poptri.loc (Poptri.genind), 8
  population.shift, 6, 9
  qt, 6
  rda, 2, 9
  shift.dot.ggplot, 11, 14, 15, 18
  shift.moon.ggplot, 11, 13, 15, 18
  shift.pie.ggplot, 11, 14, 15, 18
  shift.surf.ggplot, 11, 14, 15, 17
  shift.waffle.ggplot, 11, 14, 15, 18, 21
  VIF.subset, 6, 22
  waffle.baker (shift.waffle.ggplot), 21