Package 'gaiah'

April 27, 2023

Title Genetic and Isotopic Assignment Accounting for Habitat Suitability

Description Tools for using genetic markers, stable isotope data, and habitat suitability data to calculate posterior probabilities of breeding origin of migrating birds.

Version 0.0.5

Maintainer Eric C. Anderson <eric.anderson@noaa.gov>

Depends R (>= 3.5.0)

Imports dplyr, geosphere, magrittr, raster, rlang, sp

Suggests knitr, rmarkdown, ggplot2

License CC0

LazyData TRUE

RoxygenNote 7.2.1

Encoding UTF-8

NeedsCompilation no

Author Eric C. Anderson [cre, aut]

Repository CRAN

Date/Publication 2023-04-27 19:50:02 UTC

R topics documented:

breeding_wiwa_genetic_posteriors 2
breeding_wiwa_isotopes 3
comboize 3
comboize_and_fortify 4
example_isotope_posteriors 6
extract_isopredictions 6
gaiah 7
genetic_posteriors2rasters 7
genetic_regions 8
get_wrld_simpl 9

great_circle_raster 9
group_birds_by_location 10
isomap2raster 11
isomap_job54152_prediction 12
isotope_posterior_probs 12
migrant_wiwa_genetic_posteriors 14
migrant_wiwa_isotopes 15
wiwa_breed 16
wiwa_habitat_unclipped 16
wrld_simpl 17

Index **18**

`breeding_wiwa_genetic_posteriors`
Posterior probs of genetic region origin from Leave-one-out cross validation for breeding WIWAs

---

Description

A data frame of the same birds (roughly) that appear in `breeding_wiwa_isotopes`. A long format data frame with 2,358 rows and 5 columns

Usage

```
breeding_wiwa_genetic_posteriors
```

Format

A tibble with 2,358 rows and 5 variables. The variables are:

ID unique identifier for each bird

Short_Name another id for the bird

NumberOfLoci Number of loci successfully typed

region one of the genetic regions

posterior the posterior prob of originating from that region

Source

Kristen Ruegg, Eric Anderson, Thomas Smith

breeding_wiwa_isotopes

Isotope values, latitude, longitude and more data from 357 breeding Wilson's warblers

Description

A data frame containing hydrogen isotope values, lat, long, and IDs and some other columns of data for birds sampled on the breeding grounds. Notice that the latitude column is named "lat" and the longitude column is named "long". Those names are both all lowercase. That is the way we roll here. Make sure that you use "lat" and "long" instead of "Lat" and "Long".

Usage

```
breeding_wiwa_isotopes
```

Format

A tibble with 357 rows and 15 variables. The relevant variables for analyses here are:

ID unique identifier for each bird

Isotope.Value hydrogen isotope ratios measured in the bird's feather

lat latitude of the bird's breeding/sampling location

long latitude of the bird's breeding/sampling location

Source

Kristen Ruegg, Jeff Kelly, Thomas Smith

comboize

combine genetics, isotopes, and habitat raster with exponents as given

Description

This just multiplies the rasters together, each raised to the appropriate exponent, normalizes and returns the result

Usage

```
comboize(Mgen, Miso, Mhab, beta_gen, beta_iso, beta_hab)
```

Arguments

Mgen	the genetic posteriors rasterStack. Must be a rasterStack
Miso	the isotope posteriors rasterStack.
Mhab	a single layer raster with the habitat suitabiilty measure as a normalized probability surface.
beta_gen	the exponent to raise the habitat raster to
beta_iso	the exponent to raise the isotope raster to
beta_hab	the exponent to raise the genetic raster to

Examples

```
# first, run through the example for isotope_posterior_probs() to get
# the rasters for two migrant birds. This gives us the list "birds2"
example(isotope_posterior_probs)

# extract the posterior probs rasters from  that output into a raster stack
miso <- lapply(birds2$regular, function(x) x$posterior_probs) %>%
  raster::stack()

# see the names of the birds we are dealing with:
names(miso)

# get the genetic posteriors for those two birds
mig_gen2 <- migrant_wiwa_genetic_posteriors %>%
  dplyr::filter(ID %in% c(names(miso)))

# make genetic posterior rasters for those two birds, make sure they are
# sorted in the same order as miso, and make a raster stack of it
mgen <- genetic_posteriors2rasters(G = mig_gen2, R = genetic_regions)[names(miso)] %>%
  raster::stack()

# make a normalized prior from habitat quality that is zeros everywhere
# outside of the "known" range.
tmp <- wiwa_habitat_unclipped * wiwa_breed
mhab <- tmp / raster::cellStats(tmp, sum)

# combine genetics, isotopes and habitat with exponents of 1 on each
mcombo <- comboize(mgen, miso, mhab, 1, 1, 1)
```

---

comboize_and_fortify *prepare fortified output for multipanel plot*

Description

This takes Mgen, Miso, and Mhab for a single bird and, if available, the true breeding location. Then it computes the combo-ized raster at all the requested levels of the exponents, and creates a fortified data frame of the results suitable for plotting in ggplot

Usage

```
comboize_and_fortify(
  mgen,
  miso,
  mhab,
  gen_beta_levels = 1,
  iso_beta_levels = c(1),
  hab_beta_levels = c(1)
)
```

Arguments

mgen	genetics posterior raster
miso	isotope posterior raster
mhab	habitat suitability raster
gen_beta_levels	vector of the desired values of gen_beta
iso_beta_levels	vector of the desired values of iso_beta
hab_beta_levels	vector of the desired values of hab_beta

Examples

```
# run through the example for comboize to get the variables
# mgen, miso, and mhab that we will use.
example(comboize)

# then run that on the first bird to get a data frame
# that you can use with ggplot
ff <- comboize_and_fortify(mgen[[1]], miso[[1]], mhab)

# this can be plotted with ggplot2
## Not run:
library(ggplot2)
wmap <- get_wrld_simpl()
ggplot(mapping = aes(x=long, y = lat)) +
  coord_fixed(1.3, xlim = c(-170, -50), ylim = c(33, 70)) +
  geom_polygon(data = wmap, aes(group = group), fill = NA, color = "black", size = .05) +
  geom_raster(data = ff, mapping = aes(fill = prob), interpolate = TRUE) +
  scale_fill_gradientn(colours = c("#EBEBEB", rainbow(7)), na.value = NA) +
  theme_bw() +
  facet_wrap( ~ beta_vals, ncol = 2) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

## End(Not run)
```

`example_isotope_posteriors` *Output of* `isotope_posterior_probs` *for two migrant birds.*

Description

Because it takes too long to generate this output for future examples, we just store it as a data object to use in examples. See the example in `isotope_posterior_probs` to see what this is.

Usage

```
example_isotope_posteriors
```

Format

An object of class `list` of length 2.

Source

Ruegg et al 2014

`extract_isopredictions` *Add the isomap prediction and sd in columns attached to the feather isotope data frame*

Description

Rasterizes the isomap predictions and standard deviation (using isomap2raster) and then extracts the values associated with each location from the raster and returns the full data frame with those values joined on in columns named iso_pred and iso_sd. It overwrites those columns with a warning if either of those columns already exists in the data.

Usage

```
extract_isopredictions(isoscape, birds, pred = "predkrig", sd = "stdkrig")
```

Arguments

`isoscape`	the data frame of prediction.txt from ISOMAP. The latitude column must be named "lat" and the longitude column must be named "long".
`birds`	data frame of the individual isotope values for the birds/feathers. Should be something like `breeding_wiwa_isotopes`.
`pred`	name of the column holding the prediction (like "predkrig") in the isoscape data frame
`sd`	name of the column holding the standard deviation (like "stdkrig") in the isoscape data frame

Examples

```
# Using the provided data from breeding Wilson's warblers and the provided
# predictions from isomap_job54152:
x <- extract_isopredictions(isoscape = isomap_job54152_prediction,
                      birds = breeding_wiwa_isotopes,
                      pred = "predkrig",
                      sd = "stdkrig")
```

`gaiah` *gaiah: Genetic and Isotopic Assignment Accounting for Habitat Suitability*

Description

Tools for using genetic markers, stable isotope data, and habitat suitability data to calculate posterior probabilities of breeding origin of migrating birds.

Details

There is not a tutorial within the package, currently. The best place to find an example of how to use the functions in this package is at the GitHub repository: https://github.com/eriqande/gaiah-wiwa. Go ahead and read the README there. It should provide you with everything you need to get up and running with the gaiah package.

Finally, note that the development version of gaiah is available at https://github.com/eriqande/gaiah.

`genetic_posteriors2rasters` *Convert posteriors to particular genetic reporting groups into raster*

Description

When birds have been assigned to breeding groups or "general areas" as in Ruegg et al. 2014 then the posterior probabilty with which the birds were assigned to the groups needs to be "smeared out" in a raster over the spatial extent of the groups.

Usage

```
genetic_posteriors2rasters(G, R)
```

Arguments

`G`	long format data frame like breeding_wiwa_genetic_posteriors. Has to have columns of ID, region, and posterior
`R`	a RasterStack like "genetic_regions". The sum of these should be the total known range. The names of the regions in R must be the same as the entries in the "region" column in G.

Value

This returns a list of rasters for each bird in G. The entries in the raster are the posterior probability of being from that cell. This assumes that birds are equally likely to come from any cell within the group's region. It doesn't return a rasterStack because you can't subset rasterStacks to change orders, etc., and it mangles names.

Examples

```
library(raster)  # needed to deal with "genetic_regions" variable
# get a small subset of individuals so it doesn't take too long
data(breeding_wiwa_genetic_posteriors)
data(genetic_regions)
BW <- breeding_wiwa_genetic_posteriors %>%
  dplyr::filter(Short_Name %in% c("eNBFR01", "wABCA05", "wORHA21"))

# run the function on those
GPRs <-  genetic_posteriors2rasters(BW, genetic_regions)
```

---

genetic_regions *RasterStack showing the 6 genetic regions that Wilson's warblers may be assigned to*

Description

The sum over layers gives the same as wiwa_breed

Usage

```
genetic_regions
```

Format

RasterStack with 6 layers. Each contains 1's in the genetic region and 0's elsewhere. The sum of these layers is the raster wiwa_breed.

class RasterStack

dimensions 80, 228, 18240, 6 (nrow, ncol, ncell, nlayers)

resolution 0.5, 0.5 (x, y)

extent -168.1, -54.1, 31.2, 71.2 (xmin, xmax, ymin, ymax)

coord. ref. +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0

data source in memory

names CalSierra, Basin.Rockies, Eastern, AK.EastBC.AB, Wa.To.NorCalCoast, CentCalCoast

Source

Ruegg et al 2014

get_wrld_simpl *return the wrld_simpl data set from maptools*

Description

I define this as a function so that we don't have to attach maptools, but we can just have it in the imports. Couldn't figure out how to do it otherwise.

Usage

```
get_wrld_simpl()
```

Examples

```
ws <- get_wrld_simpl()
head(ws)
## Not run: plot(ws)
```

great_circle_raster *return a raster of great circle distances (in km)*

Description

Given an input raster R, this returns a raster of the same dimension where every cell is the great circle distance between lat, and long, and the center of every cell in R.

Usage

```
great_circle_raster(R, lat, long)
```

Arguments

R	a raster
lat	a latitude value (must be of length 1)
long	a longitude value (must be of length 1)

Examples

```
# We compute the great circle distance between the lat/long of my office in
# California, to every cell in the raster denoting the breeding habitat
# of Wilson's warbler:
gcr <- great_circle_raster(wiwa_breed, lat = 36.951564, long = -122.065116)

# plot that if you want
## Not run:
plot(gcr)
```

```
lines(get_wrld_simpl())

## End(Not run)
```

`group_birds_by_location` *Group bird isotope data by locations*

Description

This takes as input a data frame of feather isotope data that also has the isoscape predictions attached to it, just like the data frame returned by `extract_isopredictions`. The data frame must have a column that gives the general location by which you will group birds for the rescaling function. The isoscape predictions by default should be in columns named `iso_pred` for the actual prediction, and `iso_sd` for the standard deviation, as produced by `extract_isopredictions`, but those are user configurable, as well.

Usage

```
group_birds_by_location(
  D,
  feather_isotope_col,
  location_col,
  iso_pred_col = "iso_pred",
  iso_sd_col = "iso_sd"
)
```

Arguments

`D`	the data frame of feather isotope data with the isoscape predictions extracted for each location, as well, and a column giving general grouping locations for the birds.
`feather_isotope_col`	the string name of the column holding the feather isotope data.
`location_col`	the string name of the column holding the locations to be used for grouping.
`iso_pred_col`	name of the column holding the predicted values from the isoscape. Default is `iso_pred`.
`iso_sd_col`	name of the column holding the standard deviations of the predicted values from the isoscape. Default is `iso_sd_col`.

Details

This function returns a data frame with columns for the mean and SD of feather/bird values, (`meanH` and `sdH`) and the mean predicted isotope value and the mean sd of the predicted isotope values (`meaniso` and `sdiso`) for all the samples within each location. It also returns the Location column itself and a column `cnt` that gives the number of bird/tissue samples from each location.

This function throws an error if any of the locations has only 1 sample. If that is the case, you may consider merging that sample with another location (or dropping it?).

Examples

```
# first run the example for extract_isopredictions to get the variable "x"
example("extract_isopredictions")

# If this were run it gives an error because there is only 1 bird at the
# location "Charlevoix"
## Not run:
group_birds_by_location(x, feather_isotope_col = "Isotope.Value", location_col = "Location")

## End(Not run)


# remove that one bird at Charlevoix and re-run
y <- x %>%
  dplyr::filter(Location != "Charlevoix")

# then group birds by location
gbl <- group_birds_by_location(D = y,
                               feather_isotope_col = "Isotope.Value",
                               location_col = "Location")
```

---

isomap2raster *convert columns of an ISOMAP isoscape to a raster object*

Description

Just simple conversion, but nice to have this in a brief function

Usage

```
isomap2raster(isoscape, column, Proj = raster::projection(get_wrld_simpl()))
```

Arguments

isoscape	the data frame of prediction.txt from ISOMAP. The latitude column must be named "lat" and the longitude column must be named "long".
column	the name of the column to turn into a raster object. This should be a quoted string, like "predkrig".
Proj	the desired projection. By default it is raster::projection(get_wrld_simpl()), i.e. the same projection as the wrld_simpl map.

Examples

```
isorast <- isomap2raster(isomap_job54152_prediction, "predreg")
isorast
```

isomap_job54152_prediction

Predicted isotope values from ISOMAP

Description

A data frame containing predicted hydrogen isotope values, lat, long, and IDs and some other columns of data prections made by ISOMAP

Usage

```
isomap_job54152_prediction
```

Format

A tibble with 10,786 rows and 12 variables. The relevant variables for analyses here are:

lat latitude of the predicted location

long longitude of the predicted location

predreg Fill in

stdreg Fill in

predkrig Fill in

stdkrig Fill in

Source

Kristina Paxton and ISOMAP (http://isomap.rcac.purdue.edu:8080/gridsphere/gridsphere)

isotope_posterior_probs

compute posterior probabilities of origin given isotope values

Description

This function automates the whole process described in the appendix of Vander Zanden et al. (2014) for computing the posterior probability of origin of an individual (or group of individuals) given its stable isotope values, and those of a set of reference individuals, and an ISOMAP prediction of isotope values across a landscape.

Usage

```
isotope_posterior_probs(
  isoscape,
  ref_birds,
  assign_birds = NULL,
  isoscape_pred_column = "predkrig",
  isoscape_sd_column = "stdkrig",
  self_assign = FALSE
)
```

Arguments

`isoscape`	the data frame read in from "prediction.txt" from ISOMAP. The latitude column must be named "lat" and the longitude column must be named "long". You have to choose which columns to use with the parameters `isoscape_pred_column` and `isoscape_sd_column`.
`ref_birds`	a data frame of reference birds. This should have (at least) columns of "ID" (for unique identifiers for each bird), "lat", "long", "Isotope.Value" and "Location". The "Location" column will be used to group samples for the Vander Zanden Rescaling.
`assign_birds`	A data frame of birds whose breeding origins are to be inferred. These must have at a minimum the column "ID" (for uniqe identifiers for the birds) and the column "Isotope.Value". This can be left NULL if there are no birds of unknown origin to assign (for example if you are performing cross-validation on the ref_birds).
`isoscape_pred_column`	the name of the column in `isoscape` to be used as the prediction (default is "predkrig").
`isoscape_sd_column`	the name of the column in `isoscape` to be used as the standard deviation (default is "stdkrig").
`self_assign`	if TRUE, then the birds in `ref_birds` will each have posterior surfaces computed for them using a leave one out procedure (i.e. each bird in turn is left out while rescaling the precip isomap to a tissue isomap). Should not be TRUE if assign_birds is non NULL.

Details

For details see:

Vander Zanden HB, Wunder MB, Hobson KA, Van Wilgenburg SL, Wassenaar LI, Welker JM, Bowen GJ (2014) Contrasting assignment of migratory organisms to geographic origins using long-term versus year-specific precipitation isotope maps. Methods in Ecology and Evolution, 5, 891-900.

And the re-explanation of that method in Ruegg et al. (2017).

Examples

```
# obtain posterior probability rasters for the first 2 birds in the migrant_wiwa_isotopes
# data set.  This takes about 15 seconds on my laptop (most of that time is preparatory---once
# that is done, each bird goes much faster). So we don't run it here.
## Not run:
birds2 <- isotope_posterior_probs(isoscape = isomap_job54152_prediction,
                        ref_birds = breeding_wiwa_isotopes,
                        assign_birds = migrant_wiwa_isotopes[1:2,]
                        )

## End(Not run)

# However, you can load the results as a saved data object to see what they look like:
birds2 <- example_isotope_posteriors


# Since the ref_birds above were separate from the migrant birds, no leave-one-out was
# performed.  Hence birds2$loo_results is NULL, and all the results are in
# birds2$regular.

# Look at the names of the resulting output for the first bird:
names(birds2$regular[[1]])

names(birds2$regular[[1]]$assignment_parameters)



# If you want to do self-assignment for a whole bunch of reference birds, it takes much longer.
# It looks like this:
## Not run:
self_ass <- isotope_posterior_probs(isoscape = isomap_job54152_prediction,
                                    ref_birds = breeding_wiwa_isotopes,
                                    self_assign = TRUE)

## End(Not run)
```

---

migrant_wiwa_genetic_posteriors

Posterior probs of genetic region of origin for 926 WIWAs sampled during migration

---

Description

A long format data frame with 5,556 rows and 6 columns

Usage

```
migrant_wiwa_genetic_posteriors
```

Format

A tibble with 5,556 rows and 6 columns. The relevant variables for analyses here are:

ID unique identifier for each bird

Short_Name same id for the bird

Collection_Date The date the bird was sampled.

NumberOfLoci Number of loci successfully typed

region one of the genetic regions

posterior the posterior prob of originating from that region

Source

Kristina Paxton, Kristen Ruegg, Eric Anderson, Thomas Smith

---

migrant_wiwa_isotopes *Isotope values and meta data for 688 migrating Wilson's Warblers*

Description

A data frame containing hydrogen isotope values, lat, long, and IDs and some other columns of data for birds sampled during migration from Arizona. 604 of the individuals in this data set also have values in migrant_wiwa_genetic_posteriors.

Usage

```
migrant_wiwa_isotopes
```

Format

A tibble with 688 rows and 14 variables. The relevant variables for analyses here are:

ID unique identifier for each bird

Isotope.Value hydrogen isotope ratios measured in the bird's feather

Source

Kristina Paxton

`wiwa_breed` *a raster of the breeding range of Wilson's warbler*

Description

a raster of the breeding range of Wilson's warbler

Usage

```
wiwa_breed
```

Format

This a rasterized version of the breeding range of Wilson's warbler It contains 1's in the breeding range and 0's elsewhere.

class RasterLayer

dimensions 80, 228, 18240 (nrow, ncol, ncell)

resolution 0.5, 0.5 (x, y)

extent -168.1, -54.1, 31.2, 71.2 (xmin, xmax, ymin, ymax)

coord. ref. +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0

data source in memory

names layer

values 0, 1 (min, max)

Source

The rasters were generated from shapefiles provided to us by BirdLife International. (BirdLife International and NatureServe (2012) Bird species distribution maps of the world. BirdLife International, Cambridge, UK and NatureServe, Arlington, USA). Persons interested in the range map should contact BirdLife International http://www.birdlife.org/ or NatureServe http://www.natureserve.org/ directly.

`wiwa_habitat_unclipped`

RasterLayer showing the MaxEnt habitat suitability model unclipped by the known breeding range

Description

class RasterLayer

dimensions 80, 228, 18240, 6 (nrow, ncol, ncell, nlayers)

resolution 0.5, 0.5 (x, y)

extent -168.1, -54.1, 31.2, 71.2 (xmin, xmax, ymin, ymax)

coord. ref. +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0

data source in memory

values 0, 0.001093349 (min, max)

Usage

```
wiwa_habitat_unclipped
```

Format

An object of class `RasterLayer` of dimension 80 x 228 x 1.

Source

Ryan Harrigan

`wrld_simpl` *Simple map of the world*

Description

This is wrld_simpl from the maptools package. It was all that I used from the maptools package which is going to be archived at the end of 2023. So, I just saved wrld_simpl as a data object in this package.

Usage

```
wrld_simpl
```

Format

a SpatialPolygonsDataFrame

Source

Got this from the old maptools package. See ?maptools::wrld_simpl

Index

∗ **datasets**
breeding_wiwa_genetic_posteriors, 2
breeding_wiwa_isotopes, 3
example_isotope_posteriors, 6
genetic_regions, 8
isomap_job54152_prediction, 12
migrant_wiwa_genetic_posteriors, 14
migrant_wiwa_isotopes, 15
wiwa_breed, 16
wiwa_habitat_unclipped, 16
wrld_simpl, 17

breeding_wiwa_genetic_posteriors, 2
breeding_wiwa_isotopes, *2*, 3, *6*

comboize, 3
comboize_and_fortify, 4

example_isotope_posteriors, 6
extract_isopredictions, 6, *10*

gaiah, 7
genetic_posteriors2rasters, 7
genetic_regions, 8
get_wrld_simpl, 9
great_circle_raster, 9
group_birds_by_location, 10

isomap2raster, 11
isomap_job54152_prediction, 12
isotope_posterior_probs, *6*, 12

migrant_wiwa_genetic_posteriors, 14, *15*
migrant_wiwa_isotopes, 15

wiwa_breed, *8*, 16
wiwa_habitat_unclipped, 16
wrld_simpl, 17