Package ‘mapmixture’

July 22, 2025

Title Spatial Visualisation of Admixture on a Projected Map

Version 1.2.0

Description Visualise admixture as pie charts on a projected map, admixture as traditional structure barplots or facet barplots, and scatter plots from genotype principal components analysis. A 'shiny' app allows users to create admixture maps interactively. Jenkins TL (2024) <doi:10.1111/1755-0998.13943>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

URL `https://github.com/Tom-Jenkins/mapmixture`

BugReports `https://github.com/Tom-Jenkins/mapmixture/issues`

Imports dplyr, ggplot2, ggspatial, grid, purrr, rlang, rnaturalearthdata, sf, stringr, tidyr

Suggests bslib, colourpicker, htmltools, shiny, shinyFeedback, shinyjs, shinytest2, shinyWidgets, terra, testthat (>= 3.2.0), waiter

Config/testthat/edition 3

Depends R (>= 4.2)

NeedsCompilation no

Author Tom Jenkins [aut, cre, cph]

Maintainer Tom Jenkins <`tom.l.jenkins@outlook.com`>

Repository CRAN

Date/Publication 2025-01-23 16:00:02 UTC

Contents

`launch_mapmixture` *Launch Shiny App*

Description

Wrapper function used to start `mapmixture` interactive app.

App tested with the following package versions:

shiny v1.8.0, shinyFeedback v0.4.0, shinyjs v2.1.0, shinyWidgets 0.8.4, bslib 0.7.0, colourpicker 1.3.0, htmltools v0.5.8.1, waiter 0.2.5.

Usage

```
launch_mapmixture(...)
```

Arguments

`...`	additional arguments passed to `shiny::runApp()`.

Value

No return value.

Examples

```
if (interactive()){
  launch_mapmixture(launch.browser = TRUE)
}
```

`mapmixture` *Plot Pie Charts on Map*

Description

Plot admixture proportions as pie charts on a projected map. In data sets where there are multiple individuals per site, the function will calculate the mean average admixture proportion for each site.

Usage

```
mapmixture(
  admixture_df,
  coords_df,
  cluster_cols = NULL,
  cluster_names = NULL,
  boundary = NULL,
  crs = 4326,
```

```
  basemap = NULL,
  pie_size = 1,
  pie_border = 0.2,
  pie_border_col = "black",
  pie_opacity = 1,
  land_colour = "#d9d9d9",
  sea_colour = "#deebf7",
  expand = FALSE,
  arrow = TRUE,
  arrow_size = 1,
  arrow_position = "tl",
  scalebar = TRUE,
  scalebar_size = 1,
  scalebar_position = "tl",
  plot_title = "",
  plot_title_size = 12,
  axis_title_size = 10,
  axis_text_size = 8,
  basemap_border = TRUE,
  basemap_border_col = "black",
  basemap_border_lwd = 0.1
)
```

Arguments

`admixture_df`	data.frame or tibble containing admixture data (see examples).
`coords_df`	data.frame or tibble containing coordinates data (see examples).
`cluster_cols`	character vector of colours the same length as the number of clusters. If `NULL`, a blue-green palette is used.
`cluster_names`	character vector of names the same length as the number of clusters. If `NULL`, the cluster column names are used.
`boundary`	named numeric vector defining the map bounding. e.g. `c(xmin=-15, xmax=15, ymin=30, ymax=50)`. If `NULL`, a default bounding box is calculated.
`crs`	coordinate reference system. Default is the WGS 84 - World Geodetic System 1984 (EPSG:`4326`). See `?sf::st_crs` for details.
`basemap`	SpatRaster or sf object to use as the basemap. A SpatRaster object can be created from a file using the `terra::rast()` function. A sf object can be created from a file using the `sf::st_read()` function. If `NULL`, world country boundaries are used.
`pie_size`	vector of numeric values of zero or greater. Can be a single value or a vector the same length as the number of sites.
`pie_border`	numeric value of zero or greater.
`pie_border_col`	string denoting colour of pie border.
`pie_opacity`	numeric value of zero to one.
`land_colour`	string defining the colour of land.

`sea_colour`	string defining the colour of sea.
`expand`	expand axes (`TRUE` or `FALSE`).
`arrow`	show arrow (`TRUE` or `FALSE`). Added using the `ggspatial::annotation_north_arrow()` function.
`arrow_size`	numeric value of zero or greater.
`arrow_position`	string defining the position of the arrow (`"tl"`, `"tr"`, `"bl"`, `"br"`).
`scalebar`	show scalebar (`TRUE` or `FALSE`). Added using the `ggspatial::annotation_scale()` function.
`scalebar_size`	numeric value of zero or greater.
`scalebar_position`	string defining the position of the scalebar (`"tl"`, `"tr"`, `"bl"`, `"br"`).
`plot_title`	string defining the main title of the plot.
`plot_title_size`	numeric value of zero or greater.
`axis_title_size`	numeric value of zero or greater.
`axis_text_size`	numeric value of zero or greater.
`basemap_border`	boolean denoting whether to show basemap polygon borders.
`basemap_border_col`	string defining colour of basemap polygon borders.
`basemap_border_lwd`	numeric value defining linewidth of basemap polygon borders.

Value

A ggplot object.

Examples

```
# Admixture Format 1
file <- system.file("extdata", "admixture1.csv", package = "mapmixture")
admixture1 <- read.csv(file)

# Admixture Format 2
file <- system.file("extdata", "admixture2.csv", package = "mapmixture")
admixture2 <- read.csv(file)

# Admixture Format 3
file <- system.file("extdata", "admixture3.csv", package = "mapmixture")
admixture3 <- read.csv(file)

# Coordinates Format
file <- system.file("extdata", "coordinates.csv", package = "mapmixture")
coordinates <- read.csv(file)

# Plot using default parameters
mapmixture(admixture1, coordinates)
```

```
# Plot using the ETRS89-extended / LAEA Europe coordinate reference system
mapmixture(admixture1, coordinates, crs = 3035)

# Plot using custom parameters
mapmixture(
  admixture_df = admixture1,
  coords_df = coordinates,
  cluster_cols = c("#f1a340","#998ec3"),
  cluster_names = c("Group 1","Group 2"),
  crs = "+proj=merc +a=6378137 +b=6378137 +lat_ts=0 +lon_0=0 +x_0=0 +y_0=0 +units=m",
  boundary = c(xmin=-15, xmax=16, ymin=40, ymax=62),
  pie_size = 1.5,
  pie_border = 0.2,
  pie_opacity = 1,
  land_colour = "#d9d9d9",
  sea_colour = "#deebf7",
  expand = FALSE,
  arrow = TRUE,
  arrow_size = 1,
  arrow_position = "tl",
  scalebar = TRUE,
  scalebar_size = 1,
  scalebar_position = "tl",
  plot_title = "Mapmixture Figure",
  plot_title_size = 15,
  axis_title_size = 12,
  axis_text_size = 10
)
```

`scatter_plot` *PCA or DAPC Scatter Plot*

Description

Plot a scatter plot of PCA or DAPC results.

Usage

```
scatter_plot(
  dataframe,
  group_ids,
  other_group = NULL,
  type = "points",
  ...,
  labels = NULL,
  axes = c(1, 2),
  colours = NULL,
  centroids = TRUE,
```

```
  segments = TRUE,
  point_size = 3,
  point_type = 21,
  centroid_size = 3,
  hvline_type = "dotted",
  hvline_size = 0.5,
  hvline_colour = "black",
  xlab = "Axis",
  ylab = "Axis",
  percent = NULL,
  plot_title = ""
)
```

Arguments

`dataframe`	data.frame or tibble containing results from a PCA or DAPC (see examples).
`group_ids`	character vector of IDs representing the group each row belongs to. This is used to colour the scatter plot and (optionally) add centroids and segments. E.g. a vector of site names, a vector of biological categories such as male or female, etc.
`other_group`	secondary character vector of IDs defining how to colour the scatter plot. E.g. a vector of country names (see examples). If `NULL`, scatter plot is coloured by group_ids.
`type`	string defining whether to show points (`"points"`), labels (`"labels"`), or text (`"text"`).
`...`	additional arguments passed to `ggplot2::geom_point` when `type = "points"`, or to `ggplot2::geom_label` when `type = "labels"`, or to `ggplot2::geom_text` when `type = "text"`.
`labels`	character vector of IDs defining labels when `type = "label"` or `type = "text"`. If `NULL`, row names are used (integers from 1:nrow(dataframe)).
`axes`	integer vector of length two defining which axes to plot.
`colours`	character vector of colours the same length as the number of groups defined in group_ids or other_group.
`centroids`	add centroids to plot (`TRUE` or `FALSE`).
`segments`	add segments to plot (`TRUE` or `FALSE`).
`point_size`	numeric value for point size.
`point_type`	numeric value for point type (shape).
`centroid_size`	numeric value for centroid label size.
`hvline_type`	integer or string defining linetype (`1` or `"dotted"`). Input `0` for no horizontal and vertical lines.
`hvline_size`	integer defining linewidth.
`hvline_colour`	string defining line colour.
`xlab`	string defining x axis label.
`ylab`	string defining y axis label.

`percent`	numeric vector the same length as `ncol(dataframe)` defining the percentage of variance explained by each axis.
`plot_title`	string defining the main title of the plot.

Value

A ggplot object.

Examples

```
# Results from a Principal Components Analysis
file <- system.file("extdata", "pca_results.csv", package = "mapmixture")
pca_results <- read.csv(file)

# Define parameters
ind_names <- row.names(pca_results)
site_names <- rep(c("Pop1", "Pop2", "Pop3", "Pop4", "Pop5", "Pop6"), each = 100)
region_names <- rep(c("Region1", "Region2"), each = 300)
percent <- c(5.6, 4.5, 3.2, 2.0, 0.52)

# Scatter plot
scatter_plot(pca_results, site_names)

# Scatter plot with axes 1 and 3 and percent on axis labels
scatter_plot(pca_results, site_names, axes = c(1,3), percent = percent)

# Scatter plot with no centroids and segments
scatter_plot(pca_results, site_names, axes = c(1,2), percent = percent,
             centroids = FALSE, segments = FALSE)

# Scatter plot with custom colours and coloured by other_group
scatter_plot(pca_results, site_names, other_group = region_names,
             percent = percent, colours = c("#f1a340","#998ec3"))

# Scatter plot with individual labels
scatter_plot(pca_results, site_names, type = "labels",
             labels = rownames(pca_results))

# Scatter plot with individual text
scatter_plot(pca_results, site_names, type = "text",
             labels = rownames(pca_results))
```

`structure_plot` *STRUCTURE Barplot*

Description

Plot a traditional STRUCTURE barplot or a facet barplot from individual admixture proportions.

Usage

```
structure_plot(
  admixture_df,
  type = "structure",
  cluster_cols = NULL,
  cluster_names = NULL,
  legend = "none",
  labels = "site",
  flip_axis = FALSE,
  ylabel = "Proportion",
  site_dividers = TRUE,
  divider_width = 1,
  divider_col = "white",
  divider_type = "dashed",
  site_order = NULL,
  display_site_labels = TRUE,
  site_labels_size = 2,
  site_labels_x = 0,
  site_labels_y = -0.025,
  site_labels_angle = 0,
  site_ticks = TRUE,
  site_ticks_size = -0.01,
  facet_col = NULL,
  facet_row = NULL
)
```

Arguments

`admixture_df`	data.frame or tibble containing admixture data (see examples).
`type`	show a traditional STRUCTURE barplot ("structure") or a facet barplot ("facet").
`cluster_cols`	character vector of colours the same length as the number of clusters. If `NULL`, a blue-green palette is used.
`cluster_names`	character vector of names the same length as the number of clusters. If `NULL`, the cluster column names are used.
`legend`	add legend at position (`"none"`, `"top"`, `"right"`, `"bottom"` or `"left"`). Default is to hide legend.
`labels`	show labels at the site level or the individual level ("site" or "individual").
`flip_axis`	flip the axes so that the plot is vertical (TRUE or FALSE). Default is FALSE (horizontal barplot).
`ylabel`	string for y label.
`site_dividers`	add dotted lines that divide sites (TRUE or FALSE).
`divider_width`	width of site divider lines.
`divider_col`	colour of site divider lines.
`divider_type`	linetype of site divider line.

`site_order`	character vector of site labels used to customise the order of sites. If `NULL`, sites are ordered alphabetically.
`display_site_labels`	display site labels (TRUE or FALSE).
`site_labels_size`	numeric value for site label size.
`site_labels_x`	numeric value for site label horizontal position.
`site_labels_y`	numeric value for site label vertical position.
`site_labels_angle`	numeric value for rotating angle of site label.
`site_ticks`	show ticks when labels = "site".
`site_ticks_size`	numeric value for site tick size.
`facet_col`	number of columns to display for facet barplot.
`facet_row`	number of rows to display for facet barplot.

Value

A ggplot object.

Examples

```
# Admixture Format 1
file <- system.file("extdata", "admixture1.csv", package = "mapmixture")
admixture1 <- read.csv(file)

structure_plot(admixture1, type = "structure")
structure_plot(admixture1, type = "facet", facet_col = 5)
```

Index