# Package ‘FSTruct’

December 9, 2021

**Type** Package

**Title** Measure variability in population structure estimates

**Version** 0.0.0.9000

**Description** An Fst-based tool to quantify and compare the variability in Q matrices that contain rows of individual membership coefficient vectors (the default output of population structure inference programs such as STRUCTURE and ADMIXTURE). Included functions simulate random Q matrices,
plot Q matrices using ggplot2, calculate Fst/FstMax (a
normalized measure of variability) for a Q matrix, and generate
bootstrap replicates of one or more Q matrices along with associated statistics.
Accompanies (insert paper citation here).

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr,
ggplot2,
tidyr,
purrr,
stats,
gtools,
latex2exp,
rlang

**RoxygenNote** 7.1.1

**Suggests** rmarkdown,
knitr,
cowplot

**VignetteBuilder** knitr

## R topics documented:

Q_bootstrap . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
Q_plot . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
Q_simulate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
Q_stat . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

**Index** **8**

`Q_bootstrap` *Generate and analyze bootstrap replicates of one or more Q matrices*

## Description

Generates bootstrap replicate Q matrices, computes Fst/FstMax for each bootstrap replicate, produces several plots of the bootstrap distributions of Fst/FstMax for each provided Q matrix, and runs two statistical tests comparing these bootstrap distributions. The tests comparing bootstrap distributions of Fst/FstMax facilitate statistical comparison of the variability in each of multiple Q matrices.

## Usage

```
Q_bootstrap(matrices, n_replicates, K, seed)
```

## Arguments

| | |
|---|---|
| `matrices` | A dataframe, matrix, or array representing a Q matrix, or a (possibly named) list of arbitrarily many such objects. For each Q matrix, matrix rows represent an individual and the last `K` columns contain individual membership coefficients (when restricted to the last `K` columns, the rows must sum to approximately 1). If the matrices are not named (e.g., `matrices = list(matrix1,matrix2)` instead of `matrices = list(A = matrix1,B = matrix2)`), the matrices will be numbered in the order they are provided in the list. |
| `n_replicates` | The number of bootstrap replicate matrices to generate for each provided Q matrix. |
| `K` | The number of ancestral clusters in each provided Q matrix, or a vector of such K values if the value of Q differs between matrices. If a single K is provided, each individual in every matrix must have `K` membership coefficients. If a vector of multiple K values is provided, each must correspond to a Q matrix in `matrices` and be provided in the same order as the matrices. |
| `seed` | Optional; sets the random seed. Use if reproducibility of random results is desired. |

## Value

A named list containing the following entries:

- `bootstrap_replicates`: A named list of lists. Each element is named for a Q matrix provided in `matrices` and contains a list of `n_replicates` bootstrap replicates of the provided matrix. E.g., if `n_replicates = 100` and the first Q matrix in `matrices` is named `A`, then the first element of `bootstrap_replicates`, `bootstrap_replicates$bootstrap_matrices_A`, is itself a list of 100 matrices, each representing a bootstrap replicate of matrix A.
- `statistics`: A dataframe containing the output of `Q_stat`: `Fst`, `FstMax`, and `ratio` (Fst/FstMax), computed for each bootstrap replicate matrix in `bootstrap_replicates`. The ratio Fst/FstMax quantifies the variability of each Q matrix. The first column, titled `Matrix`, is a factor indicating which provided Q matrix the row corresponds to (the matrix name if `matrices` is a named list, or a number otherwise). The row names are of the form `stats_matrix.replicate` where `matrix` is the name of one of the provided Q matrices (or the entry number if the list elements were not named) and replicate is the number of bootstrap replicate (rep takes values from 1 to `n_replicates`).

- plot_boxplot: A ggplot2 box plot depicting the bootstrap distribution of Fst/FstMax for each matrix in matrices.
- plot_violin: A ggplot2 violin plot depicting the bootstrap distribution of Fst/FstMax for each matrix in matrices.
- plot_ecdf: A ggplot2 empirical cumulative distribution function plot depicting the bootstrap distribution of Fst/FstMax for each matrix in matrices.
- test_kruskal_wallis: Results of a Kruskal-Wallis test performed on the bootstrap distributions of Fst/FstMax. This test is a non-parametric statistical test of whether all provided bootstrap distributions are identically distributed.
- test_pairwise_wilcox: Results of a Wilcoxon rank-sum test performed on the bootstrap distributions of Fst/FstMax. This test is a non-parameteric statistical test of whether *each pairwise combination* of provided bootstrap distributions is identically distributed. The result is a matrix of p-values whose entries correspond to each pair of Q matrices.

## Examples

```
# Use Q_simulate to generate 4 random Q matrices
A <- Q_simulate(
  alpha = .1,
  lambda = c(.5, .5),
  rep = 1,
  popsize = 20,
  seed = 1
)

B <- Q_simulate(
  alpha = .1,
  lambda = c(.5, .5),
  rep = 1,
  popsize = 20,
  seed = 2
)

C <- Q_simulate(
  alpha = 1,
  lambda = c(.5, .5),
  rep = 1,
  popsize = 20,
  seed = 3
)

D <- Q_simulate(
  alpha = 1,
  lambda = c(.5, .5),
  rep = 1,
  popsize = 20,
  seed = 4
)

# Draw 100 bootstrap replicates from
# each of the 4 Q matrices
bs <- Q_bootstrap(
  matrices = list(
    A = A,
    B = B,
    C = C,
    D = D
  ),
  n_replicates = 100,
  K = 2
)

# Access the elements of this list using $.
# For example:
# To look at all 400 bootstrap Q matrix
# replicates:
bs$bootstrap_replicates

# To look at Fst, FstMax, and
# the ratio (Fst/FstMax) for each replicate
bs$statistics

# To look at a plot of the distribution of
# Fst/FstMax for each Q matrix:
bs$plot_violin

# To determine if each of the 4 distibutions of
# Fst/FstMax is significantly different from
# each of the other distributions:
bs$test_pairwise_wilcox
```

## Q_plot *Plot a Q matrix using ggplot2*

### Description

This function enables graphical visualization of a Q matrix, the default output of population structure inference software programs such as STRUCTURE and ADMIXTURE. In the output plot, each vertical bar represents a single individual's ancestry; the height of each color in the bar corresponds to the individual membership coefficients given by the Q matrix. Because this function produces a ggplot object, its output can be modified using standard ggplot2 syntax. For a more comprehensive population structure visualization program, see the program *distruct*.

### Usage

```
Q_plot(Q, K, arrange)
```

### Arguments

Q	A dataframe, matrix, or array representing a Q matrix. Each row represents an individual, and the last K columns contain individual membership coefficients. The first few columns may contain information not relevant to this plot; their inclusion is optional. When restricted to the last K columns, the rows of this matrix must sum to approximately 1.

K	The number of ancestral clusters in the Q matrix. Each individual must have K membership coefficients.

arrange	Optional variable controlling horizontal ordering of individuals. If arrange = TRUE, individuals are ordered by the clusters of greatest mean membership. K values of 11 or fewer.

## Value

A ggplot object describing a bar plot of membership coefficients from the Q matrix.

## Examples

```
Q_plot(
  # Make an example matrix of membership coefficients.
  # Each row is an individual. Rows sum to 1.
  Q = matrix(c(
    .4, .2, .4,
    .5, .3, .2,
    .5, .4, .1,
    .6, .1, .3,
    .6, .3, .1
  ),
  nrow = 5,
  byrow = TRUE
  ),
 K = 3, # How many ancestry coefficients per individual?
 arrange = TRUE
) +
  # Below are example, optional modifications to the default plot
  ggplot2::ggtitle("Population A") +
  ggplot2::scale_fill_brewer("Blues") +
  ggplot2::xlab("Individuals")
```

---

# `Q_simulate` *Simulate one or more Q matrices using the Dirichlet distribution*

---

## Description

Simulates Q matrices by drawing vectors of membership coefficients from a Dirichlet distribution parameterized by two variables: $\alpha$, which controls variability, and $\lambda = (\lambda_1, \lambda_2, ...., \lambda_K)$ which controls the mean of each of the K ancestry coefficients.

## Usage

```
Q_simulate(alpha, lambda, rep, popsize, seed)
```

## Arguments

| | |
|---|---|
| `alpha` | A number that sets the variability of the membership coefficients. The variance of coefficient k is Var[x_k] = $\lambda_k/(\alpha+1)$. Larger values of $\alpha$ lead to lower variability. |
| `lambda` | A vector that sets the mean membership of each ancestral cluster across the population. The vector must sum to 1. |
| `rep` | The number of Q matrices to generate. |
| `popsize` | The number of individuals to include in each Q matrix. |
| `seed` | Optional; sets the random seed. Use if reproducibility of random results is desired. |

### Value

A data frame containing the simulated Q matrices. Each row represents a single simulated individual. The data frame has the following columns

- rep: Which random Q matrix the row belongs to (a number between 1 and the parameter rep)
- ind: Which individual in each Q matrix the row corresponds to (a number between 1 and the parameter popsize)
- alpha: The alpha value used to simulate the Q matrix.
- Pop: alpha_rep (where rep and alpha are the first and third columns as described in this list). Serves as a unique identifier for each Q matrix (useful if running simulations with many different values of $\alpha$).
- spacer: a repeated ":" to make simulated Q matrices match output of population structure inference software.
- q1,q2,etc.: Membership coefficients (sum to 1).

### Examples

```
# Simulate 100 random Q matrices.
# In this example, each Q matrix has
# 100 individuals.
# On average these individuals have
# mean ancestry (1/2, 1/4, 1/4)
# from each of 3 ancestral clusters.
# The variance of each cluster i is
# Var[q_i] = lambda_i(1-lambda_i)/(alpha + 1)
# Here lambda_1 = 1/2,
#      lambda_2 = lambda_3 = 1/4

Q_list <- Q_simulate(
  alpha = 1,
  lambda = c(1 / 2, 1 / 4, 1 / 4),
  rep = 100,
  popsize = 50,
  seed = 1
)
```

## Q_stat *Compute Fst, FstMax, and the ratio Fst/FstMax for a Q matrix*

### Description

This function computes a statistical measure of ancestry variability, Fst/FstMax, for a Q matrix, the default output of population structure inference software programs such as STRUCTURE and ADMIXTURE. The function returns a named list containing the ratio Fst/FstMax as well as the values of Fst and FstMax.

### Usage

```
Q_stat(Q, K)
```

## Arguments

`Q` A dataframe, matrix, or array representing a Q matrix. Each row represents an individual and the last `K` columns contain individual membership coefficients. The first few columns may contain information not relevant to this plot; their inclusion is optional. When restricted to the last `K` columns, the rows of this matrix must sum to approximately 1.

`K` The number of ancestral clusters in the Q matrix. Each individual must have `K` membership coefficients.

## Details

Fst/FstMax is a statistic that takes a value of 0 when every individual in a population has identical ancestry, and a value of 1 when the ancestry is maximally variable (see *our paper* for more details). It is based on the population differentiation statistic Fst which, in its traditional application, is used to measure variability in allele frequencies

## Value

A named list of containing the following entries:

- `Fst`: Fst computed as if each individual is a population, and each ancestral cluster is an allele.
- `FstMax`: The maximum value of Fst (for fixed frequency of the most frequent allele, or, in the analogy, the membership of the most prevalent ancestral cluster).
- `ratio`: The ratio Fst/FstMax. We recommend that this statistic be used to quantify ancestry variability and to compare the variability of two or more Q matrices.

## Examples

```
Q_stat(
  # Make an example matrix of membership coefficients.
  # Each row is an individual. Rows sum to 1.
  Q = matrix(c(
    .4, .2, .4,
    .5, .3, .2,
    .5, .4, .1,
    .6, .1, .3,
    .6, .3, .1
  ),
  nrow = 5,
  byrow = TRUE
  ),
  K = 3
) # How many ancestry coefficients per individual?
```

# Index

Q_bootstrap, 2
Q_plot, 4
Q_simulate, 5
Q_stat, 6