Package 'BioGA'

November 11, 2026

Type Package

Title Bioinformatics Genetic Algorithm (BioGA)

Version 1.6.0

Description Genetic algorithm are a class of optimization algorithms inspired by the process of natural selection and genetics. This package allows users to analyze and optimize high throughput genomic data using genetic algorithms. The functions provided are implemented in C++ for improved speed and efficiency, with an easy-to-use interface for use within R.

License MIT + file LICENSE

URL https://danymukesha.github.io/BioGA/

BugReports https://github.com/danymukesha/BioGA/issues

Imports ggplot2, graphics, Rcpp, SummarizedExperiment, animation, rlang, biocViews, sessioninfo, BiocStyle

Depends R (>= 4.4)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

LinkingTo Rcpp

VignetteBuilder knitr

biocViews ExperimentalDesign, Technology

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/BioGA

git_branch RELEASE_3_23

git_last_commit ad6ac05

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Dany Mukesha [aut, cre] (ORCID:
<https://orcid.org/0009-0001-9514-751X>)

Maintainer Dany Mukesha <danymukesha@gmail.com>

Contents

BioGA-package 2
crossover_cpp 3
evaluate_fitness_cpp 3
initialize_population_cpp 4
mutation_cpp 5
plot_fitness 5
plot_fitness_history 6
plot_population 7
replacement_cpp 7
selection_cpp 8

Index **9**

`BioGA-package` *BioGA: Bioinformatics Genetic Algorithm (BioGA)*

Description

Genetic algorithm are a class of optimization algorithms inspired by the process of natural selection and genetics. This package allows users to analyze and optimize high throughput genomic data using genetic algorithms. The functions provided are implemented in C++ for improved speed and efficiency, with an easy-to-use interface for use within R.

Author(s)

Maintainer: Dany Mukesha <danymukesha@gmail.com> (ORCID)

See Also

Useful links:

- https://danymukesha.github.io/BioGA/
- Report bugs at https://github.com/danymukesha/BioGA/issues

crossover_cpp *Function to perform crossover between selected individuals*

Description

Function to perform crossover between selected individuals

Usage

```
crossover_cpp(selected_parents, offspring_size)
```

Arguments

selected_parents	Numeric matrix representing the selected individuals.
offspring_size	Number of offspring to generate.

Value

Numeric matrix representing the offspring.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
population <- BioGA::initialize_population_cpp(genomic_data,
               population_size = 5)
fitness <- BioGA::evaluate_fitness_cpp(genomic_data, population)
selected_parents <- BioGA::selection_cpp(population, fitness,
               num_parents = 2)
BioGA::crossover_cpp(selected_parents, offspring_size = 2)
```

evaluate_fitness_cpp *Function to evaluate fitness using genomic data*

Description

Function to evaluate fitness using genomic data

Usage

```
evaluate_fitness_cpp(genomic_data, population)
```

Arguments

genomic_data	Numeric matrix of genomic data where rows represent genes/features and columns represent samples.
population	Numeric matrix representing the population of individuals.

Value

Numeric vector of fitness scores for each individual.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
population <- BioGA::initialize_population_cpp(genomic_data,
                population_size = 5)
BioGA::evaluate_fitness_cpp(genomic_data, population)
```

---

initialize_population_cpp

Function to initialize the population from genomic data

Description

Function to initialize the population from genomic data

Usage

```
initialize_population_cpp(genomic_data, population_size)
```

Arguments

genomic_data	Numeric matrix of genomic data where rows represent genes/features and columns represent samples.
population_size	Number of individuals in the population.

Value

Numeric matrix representing the initialized population.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
BioGA::initialize_population_cpp(genomic_data, population_size = 5)
```

mutation_cpp *Function to mutate the offspring*

Description

Function to mutate the offspring

Usage

```
mutation_cpp(offspring, mutation_rate)
```

Arguments

offspring	Numeric matrix representing the offspring.
mutation_rate	Probability of mutation for each individual.

Value

Numeric matrix representing the mutated offspring.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
population <- BioGA::initialize_population_cpp(genomic_data,
                population_size = 5)
fitness <- BioGA::evaluate_fitness_cpp(genomic_data, population)
selected_parents <- BioGA::selection_cpp(population,
                fitness, num_parents = 2)
offspring <- BioGA::crossover_cpp(selected_parents, offspring_size = 2)
BioGA::mutation_cpp(offspring, mutation_rate = 0)
```

plot_fitness *Plot Fitness Values*

Description

Plot the fitness values of the population over generations.

Usage

```
plot_fitness(fitness_values)
```

Arguments

fitness_values	A numeric vector containing fitness values.

Value

Plot of fitness

Examples

```
# example of usage
fitness_values <- c(10, 8, 6, 4, 2)
plot_fitness(fitness_values)
```

`plot_fitness_history` *Plot Fitness Change Over Generations*

Description

Plot the change in fitness values over generations.

Usage

```
plot_fitness_history(fitness_history)
```

Arguments

`fitness_history`
: A list containing fitness values for each generation.

Value

Plot of fitness history

Examples

```
# example of usage
fitness_history <- list(c(10, 8, 6, 4, 2), c(9, 7, 5, 3, 1))
plot_fitness_history(fitness_history)
```

plot_population *Plot Population Distribution*

Description

Plot the distribution of individuals in the population.

Usage

```
plot_population(population)
```

Arguments

population	A numeric matrix containing the population data.

Value

Plot of population

Examples

```
# example of usage
population <- matrix(runif(100), nrow = 10, ncol = 10)
plot_population(population)
```

replacement_cpp *Function to replace non-selected individuals in the population*

Description

Replace non-selected individuals in the population

Usage

```
replacement_cpp(population, offspring, num_to_replace)
```

Arguments

population	Numeric matrix representing the population of individuals.
offspring	Numeric matrix representing the offspring.
num_to_replace	Number of individuals to replace.

Value

Numeric matrix representing the updated population.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
population <- BioGA::initialize_population_cpp(genomic_data,
                population_size = 5)
fitness <- BioGA::evaluate_fitness_cpp(genomic_data, population)
selected_parents <- BioGA::selection_cpp(population, fitness,
                      num_parents = 2)
offspring <- BioGA::crossover_cpp(selected_parents, offspring_size = 2)
mutated_offspring <- BioGA::mutation_cpp(offspring, mutation_rate = 0)
BioGA::replacement_cpp(population, mutated_offspring, num_to_replace = 1)
```

---

selection_cpp *Function to select individuals based on fitness scores*

Description

Function to select individuals based on fitness scores

Usage

```
selection_cpp(population, fitness, num_parents)
```

Arguments

population	Numeric matrix representing the population of individuals.
fitness	Numeric vector of fitness scores for each individual.
num_parents	Number of individuals to select.

Value

Numeric matrix representing the selected individuals.

Examples

```
# example of usage
genomic_data <- matrix(rnorm(100), nrow = 10, ncol = 10)
population <- BioGA::initialize_population_cpp(genomic_data,
                population_size = 5)
fitness <- BioGA::evaluate_fitness_cpp(genomic_data, population)
BioGA::selection_cpp(population, fitness, num_parents = 2)
```

Index

∗ **internal**
BioGA-package, 2

BioGA *(*BioGA-package*)*, 2
BioGA-package, 2

crossover_cpp, 3

evaluate_fitness_cpp, 3

initialize_population_cpp, 4

mutation_cpp, 5

plot_fitness, 5
plot_fitness_history, 6
plot_population, 7

replacement_cpp, 7

selection_cpp, 8