Package: GeneGA (via r-universe)

November 25, 2026

Type Package

Title Design gene based on both mRNA secondary structure and codon usage bias using Genetic algorithm

Version 1.64.0

Date 2013-3-6

Author Zhenpeng Li and Haixiu Huang

Maintainer Zhenpeng Li <zpli21@gmail.com>

Depends seqinr, hash, methods

biocViews GeneExpression

Description R based Genetic algorithm for gene expression optimization by considering both mRNA secondary structure and codon usage bias, GeneGA includes the information of highly expressed genes of almost 200 genomes. Meanwhile, Vienna RNA Package is needed to ensure GeneGA to function properly.

License GPL version 2

URL http://www.tbi.univie.ac.at/~ivo/RNA/

Config/Bioconductor/UnsupportedPlatforms windows

Config/pak/sysreqs zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:30:31 UTC

RemoteUrl https://github.com/bioc/GeneGA

RemoteRef RELEASE_3_24

RemoteSha 618dbb62042cb0d174034301fab21e6a62971153

Contents

GeneGA-package 2
GeneCodon 3
GeneFoldGA 4
GeneFoldGA-class 5

GeneGA 6
GeneGA-class 8
plotGeneGA-methods 9
wSet 9

Index **11**

GeneGA-package *Designing gene based on mRNA secondary structure and codon usage bias using Genetic algorithm, GeneGA includes the information of highly expressed genes of almost 200 genomes.*

Description

R based Genetic algorithm for gene expression optimization considering mRNA secondary structure and codon usage bias

Details

Package:	GeneGA
Type:	Package
Version:	1.1.2
Date:	2010-11-19
License:	GPL version 2
LazyLoad:	yes

Author(s)

Zhenpeng Li, Fei Li, Xiaochen Bo and Shengqi Wang

Maintainer: Zhenpeng Li <zpli21@gmail.com>

References

Liu L.,Kang L.S.,Chen Y.P.. (1993). Non-numerical parallel algorithms(The second volume)–genetic algorithms,Science Press(In Chinese)

Tuller T, Carmi A, Vestsigian K, et al. An Evolutionarily Conserved Mechanism for Controlling the Efficiency of Protein Translation. Cell 2010, 141:344-354

See Also

GeneGA, GeneFoldGA, GeneCodon, GeneGA-class, GeneFoldGA-class, show-methods, plotGeneGA-methods, wSet

Examples

```
seqfile=system.file("sequence","EGFP.fasta",package="GeneGA")
seq=unlist(getSequence(read.fasta(seqfile),as.string=TRUE))
GeneGA.result=GeneFoldGA(sequence=seq,popSize=40,iters=100,crossoverRate=0.3,
        mutationChance=0.05,region=c(1,42))
plotGeneGA(GeneGA.result)
show(GeneGA.result)
```

GeneCodon *Optimizing the codon usage of gene by replacing less usage codons with the most preference used codons or replacing the most preference used codons with less usage codons.*

Description

The function adjusts the codon usage of gene by replacing less usage codons(the most preference codons) with the most preference codons(less usage codons).

Usage

```
GeneCodon(seq, organism = "ec", max=TRUE, scale=0.5, numcode= 1)
```

Arguments

seq	the sequence to optimize
organism	the organism that gene due to express, GeneGA contains codon usage bias information of almost 200 genomes
max	if max is TRUE, less usage codons will be replaced with the most preference codons, and vice versa.
scale	When max is FALSE, scale is used to assign the range of less used synonymous codons of each amino acid to sample. The default value is 0.5, it means each codon will be sampled from its 50% less used synonymous codons.
numcode	The ncbi genetic code number for translation. By default the standard genetic code is used. Referring to the help page of "translate" function in seqinr package for details

Value

This function returns the optimized sequence as string. If max is True, sequence consisting of most preference used codons will be returned,

Author(s)

Zhenpeng Li, Fei Li, Xiaochen Bo and Shengqi Wang

See Also

wSet, GeneGA

Examples

```
seqfile=system.file("sequence","EGFP.fasta",package="GeneGA")
seq=unlist(getSequence(read.fasta(seqfile),as.string=TRUE))
GeneCodon(seq)
```

GeneFoldGA *the function optimizes gene merely considering mRNA secondary structure.*

Description

Optimizing gene by only considering mRNA secondary structure using genetic algorithm. The function has the default evaluation function with minimum free energy as variable. The optimum is the gene for which the minimum free energy is maximum. Results can be visualized with plotGeneGA and displayed with show.

Usage

```
GeneFoldGA(sequence = NULL, popSize = 50, iters = 100, crossoverRate = 0.2,
         mutationChance = 0.05, region = NULL, showGeneration = TRUE,
         frontSeq = NULL, organism="ec", ramp = FALSE,numcode=1)
```

Arguments

sequence	the mRNA sequence to optimize
popSize	the population size
iters	the number of iteration
crossoverRate	the crossover rate of each generation. By default 0.2
mutationChance	mutation chance of the gene in the unit of codons
region	the region of sequence to optimize, the other part of sequence outside the region is optimized only considering codon usage bias as well.
showGeneration	show the generation the genetic algorithm progressing, the default value is TRUE
frontSeq	frontSeq denotes the regulatory segment before the start codon. If frontSeq is specified, frontSeq will be considered when compute the minimum free energy. The default value is NULL.
organism	the organism that gene due to express, the package contains codon usage bias information of almost 200 genomes.
ramp	the organism that gene due to express, the package contains codon usage bias information of almost 200 genomes.
numcode	The ncbi genetic code number for translation. By default the standard genetic code is used. Referring to the help page of "translate" function in seqinr package for details.

Value

A GeneFoldGA instance is returned.

Author(s)

Zhenpeng Li, Fei Li, Xiaochen Bo and Shengqi Wang

References

Liu L.,Kang L.S., Chen Y.P. (1993)Non-numerical parallel algorithms(The second volume)–genetic algorithms,Science Press(In Chinese)

See Also

GeneFoldGA-class, GeneGA, show-methods, plotGeneGA-methods

Examples

```
seqfile=system.file("sequence","EGFP.fasta",package="GeneGA")
seq=unlist(getSequence(read.fasta(seqfile),as.string=TRUE))
GeneGA.result=GeneFoldGA(sequence=seq,popSize=40,iters=100,crossoverRate=0.3,
         mutationChance=0.05,region=c(1,42))
```

GeneFoldGA-class *Class "GeneFoldGA"*

Description

GeneFoldGA class for representing the GeneFoldGA results

Slots

seq: Object of class "character". The mRNA sequence to optimize.

iters: Object of class "integer". The number of iteration.

popSize: Object of class "numeric". The population size.

crossoverRate: Object of class "numeric". The crossover rate of each generation. By default 0.2.

mutationChance: Object of class "numeric". Mutation chance of the gene in the unit of codon.

region: Object of class "ANY". The region of sequence to optimize, the other part of sequence outside the region is optimized by only considering codon usage bias.

organism: Object of class "character". The organism that gene due to express,the package contains codon usage bias information of almost 200 genomes.

eval_value: Object of class "numeric". The evaluation function values of all final population.

free_en: Object of class "numeric". The minimux free energy values of all final population.

`eval_value_set`: Object of class `"numeric"`. The mean of evaluation function values of the population.

`eval_value_set02`: Object of class `"numeric"`. The maximum of evaluation function values of the population.

`population`: Object of class `"character"`. The final population undergone the genetic algorithm.

`ramp`: Object of class `"ANY"`. Ramp specifies the region with low translation efficiency. Generally,the first 90 to 150 bases are the ramp region of gene,which are deemed to be an evolutionarily conserved mechanism for controlling the efficiency of protein translation. Referring to the reference for more detailed description.

Methods

show `signature(object = "GeneFoldGA")`: Displaying the results of GeneFoldGA, the first three distinctive and optimum sequences can be returned, as well as the corresponding minimum free energys.

plotGeneGA `signature(object = "GeneFoldGA")`: Visualizing the variation of optimized and mean variable values during the progress that genetic algorithm performed. Furthermore, the plot also can be used to check the results whether well converged.

See Also

GeneFoldGA, GeneGA, show-methods, plotGeneGA-methods

GeneGA *the function optimizes gene expression based on both mRNA secondary structure and codon usage bias*

Description

Optimizing gene expression considering both mRNA secondary structure and codon usage bias using genetic algorithm. The function has the default evaluation function with variables CAI and minimum free energy, the sum of squares of ranks of which is used as an evaluation value to pilot the evoluation. The optimum is the gene for which the evaluation value is maximum. Results can be visualized with plotGeneGA and displayed with show.

Usage

```
GeneGA(sequence = NULL, popSize = 50, iters = 150, crossoverRate = 0.2,
         mutationChance = 0.05, region = NULL, organism = "ec",
         showGeneration = TRUE, frontSeq = NULL, ramp=FALSE, numcode=1)
```

Arguments

sequence	the mRNA sequence to optimize
popSize	the population size
iters	the number of iteration
crossoverRate	the crossover rate of each generation. By default 0.2
mutationChance	mutation chance of the gene in the unit of codon
region	the region of sequence to optimize, the other part of sequence outside the region is optimized by only considering codon usage bias.
organism	the organism that gene due to express, the package contains codon usage bias information of almost 200 genomes.
showGeneration	show the generation the genetic algorithm progressing, the default value is TRUE.
frontSeq	frontSeq denotes the regulatory segment before the start codon.If frontSeq is specified, frontSeq will be considered when compute the minimum free energy.The default value is NULL.
ramp	ramp specifies the region with low translation efficiency.Generally, the first 90 to 150 bases are the ramp region of gene, which are deemed to be an evolutionarily conserved mechanism for controlling the efficiency of protein translation.Referring the reference for more detailed description.
numcode	The ncbi genetic code number for translation. By default the standard genetic code is used. Referring to the help page of "translate" function in seqinr package for details.

Value

A GeneGA instance is returned.

Author(s)

Zhenpeng Li, Fei Li, Xiaochen Bo and Shengqi Wang

References

Liu L.,Kang L.S.,Chen Y.P.. (1993). Non-numerical parallel algorithms(The second volume)–genetic algorithms,Science Press(In Chinese).

Tuller T, Carmi A, Vestsigian K, et al. An Evolutionarily Conserved Mechanism for Controlling the Efficiency of Protein Translation. Cell 2010, 141:344-354.

See Also

GeneGA-class, GeneFoldGA, GeneCodon, show-methods, plotGeneGA-methods

Examples

```
seqfile=system.file("sequence","EGFP.fasta",package="GeneGA")
seq=unlist(getSequence(read.fasta(seqfile),as.string=TRUE))
GeneGA.result=GeneGA(sequence=seq,popSize=40,iters=100,crossoverRate=0.3,
        mutationChance=0.05,region=c(1,60))
```

GeneGA-class *Class "GeneGA"*

Description

GeneGA class for representing the GeneGA results

Slots

All the slots of the GeneFoldGA class, plus the following slots:

`CAI_value`: Object of class `"numeric"`. If ramp and specified region are not intersecting, it denotes the CAI values of all final population, otherwise, it denotes the CAI values of the intersected region of all final population.

`CAI_value_`: Object of class `"numeric"`. When ramp and specified region are intersecting, it denotes the CAI values of the specified region that not intersecting with the specified region of all final population.

`free_en_set`: Object of class `"numeric"`. The mean of minimum free energy of the population.

`CAI_value_set`: Object of class `"numeric"`. If ramp and specified region are not intersecting, it denotes the mean of CAI values of the population, otherwise, it denotes the mean CAI values of the the intersected region of the population.

`CAI_value_set_`: Object of class `"numeric"`. When ramp and specified region are intersecting, it denotes the mean CAI values of the specified region that not intersecting with the specified region of the population.

`free_en_set02`: Object of class `"numeric"`. The maximum of minimum free energy of the population.

`CAI_value_set02`: Object of class `"numeric"`. If ramp and specified region are not intersecting, it denotes the maximum of CAI values of the population, otherwise, it denotes the maximum CAI values of the the intersected region of the population.

`CAI_value_set02_`: Object of class `"numeric"`. When ramp and specified region are intersecting, it denotes the maximum CAI values of the specified region that not intersecting with the specified region of the population.

Methods

show `signature(object = "GeneGA")`: Displaying the results of GeneGA and GeneFoldGA, the first three distinctive and optimum sequences can be returned, as well as their overall evaluation values, CAI values and minimum free energys.

plotGeneGA `signature(object = "GeneGA")`: Visualizing the variation of optimized and mean overall evaluation values and variable values during the progress that genetic algorithm performed. Furthermore, the plot also can be used to check the results whether well converged.

See Also

GeneGA, GeneFoldGA, GeneCodon, show-methods, plotGeneGA-methods

plotGeneGA-methods *plotGeneGA methods of GeneGA and GeneFoldGA objects*

Description

plotGeneGA implement plotGeneGA methods for GeneGA and GeneFoldGA objects respectively.The functions visualize the variation of optimized and mean overall evaluation values and variable values during the progress that genetic algorithm performed. Furthermore, the plot also can be used to check the results whether well converged.

Methods

x = "GeneFoldGA" plotGeneGA method for GeneFoldGA

x = "GeneGA" plotGeneGA method for GeneGA, it also has a parameter "type", which can be assigned with one of the three values, the default value is "default", which show the variation of mean and optimized overall evaluation value along with the generation. while 1, 2 display the variation of mean and optimized overall CAI value and minimum free energy respectively, and 3 display the scatter plot of two variables–CAI and minimum free energy

Author(s)

Zhenpeng Li, Fei Li, Xiaochen Bo and Shengqi Wang

See Also

GeneGA-class, GeneFoldGA-class, GeneGA, GeneFoldGA, show-methods

Examples

```
seqfile=system.file("sequence", "EGFP.fasta", package="GeneGA")
seq=unlist(getSequence(read.fasta(seqfile), as.string=TRUE))
GeneGA.result=GeneFoldGA(sequence=seq, popSize=40, iters=100, crossoverRate=0.3,
         mutationChance=0.05, region=c(1,42))
plotGeneGA(GeneGA.result)
```

wSet *Codon Adaptation Index (CAI) w tables*

Description

Codon Adaptation Index (CAI) w tables of almost 200 genomes,the computation is based on the highly expressed genes of these genomes.

Usage

```
data(wSet)
```

Format

A data frame with 200 observations on the following 64 variables, which denote the 64 codes.

Details

Codons can be accessed by names(wSet) and the genome names can be accessed by row.names(wSet).

Source

The data were combined from two sources: the first part is the first three rows,which contains w tables masked from the caitab dataset of seqinr package,while the other 197 rows were computed based the predicted highly expressed genes of 197 genomes from HEG-DB database.

References

Puigbo P, Romeu A, Garcia-Vallve S. HEG-DB: a database of predicted highly expressed genes in prokaryotic complete genomes under translational selection. Nucl. Acids Res. 2008, 36:D524-527.

Charif D, Lobry J. 2007. SeqinR 1.0-2. A Contributed Package to the R Project for Statistical Computing Devoted to Biological Sequences Retrieval and Analysis. In: Structural Approaches to Sequence Evolution (Bastolla U, Porto M, Roman E, Vendruscolo M, eds.), Berlin Heidelberg: Springer, p207-232.

See Also

GeneCodon, GeneGA, GeneFoldGA

Examples

```
data(wSet)
```

Index

∗ **classes**
GeneFoldGA-class, 5
GeneGA-class, 8
∗ **datasets**
wSet, 9
∗ **methods**
plotGeneGA-methods, 9
∗ **package**
GeneGA-package, 2

GeneCodon, *2*, 3, *7, 8*, *10*
GeneFoldGA, *2*, 4, *6–10*
GeneFoldGA-class, 5
GeneGA, *2*, *4, 5*, 6, *6*, *8–10*
GeneGA-class, 8
GeneGA-package, 2

plotGeneGA *(*plotGeneGA-methods*)*, 9
plotGeneGA,GeneFoldGA-method *(*plotGeneGA-methods*)*, 9
plotGeneGA,GeneGA-method *(*plotGeneGA-methods*)*, 9
plotGeneGA-methods, 9
plotGeneGA.GeneFoldGA *(*plotGeneGA-methods*)*, 9
plotGeneGA.GeneGA *(*plotGeneGA-methods*)*, 9

show,GeneFoldGA-method *(*GeneFoldGA-class*)*, 5
show,GeneGA-method *(*GeneGA-class*)*, 8

wSet, *2*, *4*, 9