Package ‘sscu’

October 12, 2016

Type Package

Title Strength of Selected Codon Usage

Version 1.0.2

Date 2016-3-18

Author Yu Sun

Maintainer Yu Sun <sunyu1357@gmail.com>

Description The package can calculate the selection in codon usage in bacteria species. First and most important, the package can calculate the strength of selected codon usage bias (sscu) based on Paul Sharp's method. The method take into account of background mutation rate, and focus only on codons with universal translational advantages in all bacterial species. Thus the sscu index is comparable among different species. In addition, detainled optimal codons (selected codons) information can be calculated by optimal_codons function, so the users will have a more accurate selective scheme for each codons. Furthermore, we added one more function optimal_index in the package. The function has similar mathematical formula as s index, but focus on the estimates the amount of GC-ending optimal codon for the highly expressed genes in the four and six codon boxes. The function takes into account of background mutation rate, and it is comparable with the s index. However, since the set of GC-ending optimal codons are likely to be different among different species, the index can not be compared among different species.

Depends R (>= 3.3)

Imports Biostrings (>= 2.36.4), seqinr (>= 3.1-3), BiocGenerics (>= 0.16.1)

Suggests knitr, rmarkdown

VignetteBuilder knitr

LazyLoad yes

License GPL (>= 2)

biocViews Genetics, GeneExpression, WholeGenome

NeedsCompilation no

R topics documented:

sscu-package *Strength of Selected Codon Usage*

Description

The package sscu (Strength of Selected Codon Usage) calculates the selective profile in codon usage in bacteria species. First of all, the package calculate the s index based on Paul Sharp's method . The method take into account of background mutation rate, and focus only on codons with universal translational advantages in all bacterial species. Thus the s index is comparable among different species. In addition, detailed optimal codons (selected codons) information can be calculated by optimal_codons function, so the users will have a more accurate selective scheme for each codons. Furthermore, we added one more function optimal_index in the package. The function has similar mathematical formula as s index, but focus on the estimates the amount of GC-ending optimal codon for the highly expressed genes in the four and six codon boxes. The function takes into account of background mutation rate, and it is comparable with the s index. However, since the set of GC-ending optimal codons are likely to be different among different species, the index can not be compared among different species.

Details

The DESCRIPTION file:

Package:	sscu
Type:	Package
Title:	Strength of Selected Codon Usage
Version:	1.0.2
Date:	2016-3-18
Author:	Yu Sun
Maintainer:	Yu Sun <sunyu1357@gmail.com>
Description:	The package can calculate the selection in codon usage in bacteria species. First and most important, the
Depends:	R (>= 3.3)
Imports:	Biostrings (>= 2.36.4), seqinr (>= 3.1-3), BiocGenerics (>= 0.16.1)
Suggests:	knitr, rmarkdown
VignetteBuilder:	knitr

LazyLoad:	yes
License:	GPL (>= 2)
biocViews:	Genetics, GeneExpression, WholeGenome
NeedsCompilation:	no

Index of help topics:

```
genomic_gc3           genomic gc3 for an multifasta genomic file
optimal_codons        statistical table for the optimal codons
optimal_index         optimal codons index for the four and six codon
                      boxes
s_index               S index (Strength of Selected Codon Usage)
sscu-package          Strength of Selected Codon Usage
```

Author(s)

Yu Sun

Maintainer: Yu Sun <sunyu1357@gmail.com>

References

Sharp PM, Bailes E, Grocock RJ, Peden JF, Sockett RE (2005). Variation in the strength of selected codon usage bias among bacteria. Nucleic Acids Research.

`genomic_gc3` *genomic gc3 for an multifasta genomic file*

Description

The function calculates the genomic gc3 for an multifasta genomic CDS file. The function first concatenated all the CDS sequences in the file into one long CDS string, than calculated the gc3 from the GC3 function in seqinr package. You can also use the function to calculate the gc3 for a single gene, or a set of genes, depends what content you put in the input file.

Usage

```
genomic_gc3(inputfile)
```

Arguments

`inputfile`	a character vector for the filepath of the whole genome cds file

Details

The function calculates the genomic gc3 for an multifasta genomic CDS file. The function first concatenated all the CDS sequences in the file into one long CDS string, than calculated the gc3 from the GC3 function in seqinr package. You can also use the function to calculate the gc3 for a single gene, or a set of genes, depends what content you put in the input file. The result can be used as input for the s_index calculation.

Value

a numeric vector genomic_gc3 is returned

Author(s)

Yu Sun

See Also

`GC3` in seqinr library

Examples

```
# ----------------------------------------------- #
#     Lactobacillus kunkeei example               #
# ----------------------------------------------- #

  # Here is an example to calculate the genomic gc3
  # input the one multifasta files to calculate genomic gc3
  genomic_gc3(system.file("sequences/L_kunkeei_genome_cds.ffn",package="sscu"))
```

`optimal_codons` *statistical table for the optimal codons*

Description

Optimal codons are defined as codons significantly enriched in the highly expressed genes compared to the lowly expressed genes, or other set of appropriate reference genes. In another word, these codons were favored by translational selection. This function calculate the optimal codon list with p-values, thus user could have a general idea of which codons were preferred by selection in the genome.

Usage

```
optimal_codons(high_cds_file = NULL,ref_cds_file = NULL,p_cutoff = 0.05)
```

Arguments

`high_cds_file` a character vector for the filepath of the highly expressed genes

`ref_cds_file` a character vector for the filepath of the reference cds file

`p_cutoff` a numeric vector to set the cutoff of p value for the chi.square test, default is set to 0.05

Details

Optimal codons are defined as codons significantly enriched in the highly expressed genes compared to the lowly expressed genes, or other set of appropriate reference genes. In another word, these codons were favored by translational selection, which was strongest among highly expressed genes. This function calculate the optimal codon list with p-values, thus user could have a general idea of which codons were preferred by selection in the genome.

The argument high_cds_file should specific the path for the highly expressed gene dataset. It is up to the users how to define which dataset of highly expressed genes. Some studies use the expression data, or Nc value to divide genes into highly/lowly sets. Other studies use a specific dataset, such as only including the very highly expressed genes (ribosomal genes).

The argument ref_cds_file should specific the path for the lowly expressed gene dataset, or any appropriate dataset. In Sharp PM paper (Forces that influence the evolution of codon bias), he used the all gene data set as neutral reference and also get a list of optimal codons.

The argument p_cutoff set the cutoff for p values in the chi.square test. Only codons are significantly enriched in the highly expressed genes are marked with + symbol in the ouotput tables. The codons are significantly lower presented in the highly expressed genes are marked with - symbol. The codons are not significantly differently presented compared to the reference dataset are marked with NA symbol.

The function also output the rscu value for the high expressed dataset and reference dataset.

Value

a dataframe is returned

`rscu_high` rscu value for the highly expressed dataset

`rscu_ref` rscu value for the reference dataset

`high_No_codon` number of codons found in the highly expressed dataset

`high_expect_No_codon` number of expected codons in the highly expressed dataset

`ref_No_codon` number of codons found in the reference dataset

`ref_expect_No_codon` number of expected codons in the reference dataset

`p_value` p value for the chi.square test

`symbol` codons are significantly enriched in the highly expressed genes are marked with +; codons are significantly lower presented in the highly expressed genes are marked with -; codons are not significantly differently presented compared to the reference dataset are marked with NA

Author(s)

Yu Sun

See Also

uco in seqinr library for rscu calculation.

Examples

```
# ---------------------------------------------- #
#     Lactobacillus kunkeei example              #
# ---------------------------------------------- #

  # Here is an example to load the data included in the sscu package
 optimal_codons(high_cds_file=system.file("sequences/L_kunkeei_highly.ffn",package="sscu"),ref_cds_file=system

  # if you want to set the p value cutoff as 0.01
 optimal_codons(high_cds_file=system.file("sequences/L_kunkeei_highly.ffn",package="sscu"),ref_cds_file=system

 # if you want to load your own data, you just specify the file path for your input as these examples
 # optimal_codons(high_cds_file = "/home/yu/Data/codon_usage/bee_endosymbionts/sharp_40_highly_dataset/Bin2.ff
```

optimal_index *optimal codons index for the four and six codon boxes*

Description

The function optimal_index can estimate the relative amount of GC-ending optimal codon for the four and six codon boxes codon in a given mutational background. The function has same mathematical formula as sscu and also take into account of background mutation rate, thus is comparable with the S index. However, since the set of GC-ending optimal codons are likely to be different among different species, the index can not be compared among different species.

Usage

```
optimal_index(high_cds_file = NULL, genomic_cds_file = NULL)
```

Arguments

high_cds_file a character vector for the filepath of the highly expressed genes

genomic_cds_file a character vector for the filepath of the whole genome cds file

Details

The function optimal_index can estimate the relative amount of GC-ending optimal codon for the four and six codon boxes codon in a given mutational background. The function has same mathematical formula as sscu and also take into account of background mutation rate, thus is comparable with the S index. However, since the set of GC-ending optimal codons are likely to be different among different species, the index can not be compared among different species.

The argument high_cds_file must be specified with the input filepath for the highly expressed genes. The file should be a multifasta file contains 40 highly, including elongation factor Tu, Ts, G, 50S ribosomal protein L1 to L6, L9 to L20, 30S ribosomal protein S2 to S20. This file can be generated by either directly extract these DNA sequence from genbank file, or parse by blast program. For the four amino acids (Phy, Tyr, Ile and Asn), the C-ending codons are always preferred than the U-ending codons. Thus, only these four codons were taken into account in the analyses.

The arguments, genomic_cds_file, is used to calculate the genomic mutation rate (gc3). The genomic_cds_file should be a multifasta file contains all the coding sequences in the genome, and the function use it to calculate the genomic gc3 and mutation rate.

Noted, most of the AT biased genomes do not have any GC-ending optimal codons for the four and six codon boxes, thus the function will report NA as output.

Currently, the function only calculate the usage of GC-ending optimal codon. In addition, most of the AT biased genomes do not have any GC-ending optimal codons for the four and six codon boxes, thus the function will report NA as output. The index 0 means the optimal codon usage follows the mutation pattern, whereas higher values menas more GC-ending optimal codons are used in the highly expressed genes.

Value

a numeric vector optimal_index is returned

Author(s)

Yu Sun

References

unpublished paper from Yu Sun

See Also

the s_index function in the same package

Examples

```
# ----------------------------------------------- #
#     Lactobacillus kunkeei example               #
# ----------------------------------------------- #

  # Here is an example to load the data included in the sscu package
  # input the two multifasta files to calculate sscu
 optimal_index(high_cds_file=system.file("sequences/L_kunkeei_highly.ffn",package="sscu"),genomic_cds_file=sys
# if you want to load your own data, you just specify the file path for your input as these examples
# optimal_index(high_cds_file="/home/yu/Data/codon_usage/bee_endosymbionts/sharp_40_highly_dataset/Bin2.ffn",
```

s_index *S index (Strength of Selected Codon Usage)*

Description

The function sscu calculates the S index (strength of selected codon usage bias) for bacteria species based on Paul Sharp's method. The method take into account of background mutation rate, and focus only on codons with universal translational advantages in all bacterial species. Thus the sscu index can be used to quantify the strength of translational selection and is comparable among different species.

Usage

```
s_index(high_cds_file = NULL, genomic_cds_file = NULL, gc3 = NULL)
```

Arguments

`high_cds_file`	a character vector for the filepath of the highly expressed genes
`genomic_cds_file`	a character vector for the filepath of the whole genome cds file
`gc3`	a numeric vector with gc3 value, eg, 0.5

Details

The function calculates the S index (strength of selected codon usage bias) for bacteria species based on Paul Sharp's method.The method take into account of background mutation rate (in the program, two arguments genomic_cds_file and gc3, are input to calculate mutation), and focus only on codons with universal translational advantages in all bacterial species (in the program, one argument high_cds_file, is input to calculate these codons). Thus the s index can be used to quantify the strength of translational selection and is comparable among different species.

The argument high_cds_file much be specified with the input filepath for the highly expressed genes. The file should be a multifasta file contains 40 highly, including elongation factor Tu, Ts, G, 50S ribosomal protein L1 to L6, L9 to L20, 30S ribosomal protein S2 to S20. This file can be generated by either directly extract these DNA sequence from genbank file, or parse by blast program. For the four amino acids (Phy, Tyr, Ile and Asn), the C-ending codons are always preferred than the U-ending codons. Thus, only these four codons were taken into account in the analyses.

The two arguments, genomic_cds_file or gc3, is used to calculate the genomic mutation rate, and one of them must be specified. The genomic_cds_file should be a multifasta file contains all the coding sequences in the genome, and the function use it to calculate the genomic gc3 and mutation rate. If the gc3 value for the genome is known already, you can specify it in the argument gc3. If both the genomic_cds_file and gc3 arguments are specified, the function will use the genomic_cds_file to calculate mutation rate, and neglect the gc3 argument.

Value

a numeric vector s-index is returned

Author(s)

Yu Sun

References

Sharp PM, Bailes E, Grocock RJ, Peden JF, Sockett RE (2005). Variation in the strength of selected codon usage bias among bacteria. Nucleic Acids Research.

See Also

`uco` in seqinr library for rscu calculation

Examples

```
# ----------------------------------------------- #
#     Lactobacillus kunkeei example              #
# ----------------------------------------------- #

  # Here is an example to load the data included in the sscu package
  # input the two multifasta files to calculate sscu
  s_index(high_cds_file=system.file("sequences/L_kunkeei_highly.ffn",package="sscu"),genomic_cds_file=system.fi

  # alternatively, input one multifasta file and gc3 content to calculate sscu
  s_index(high_cds_file=system.file("sequences/L_kunkeei_highly.ffn",package="sscu"),gc3=0.76)

  # if you want to load your own data, you just specify the file path for your input as these examples
  # s_index(high_cds_file="/home/yu/Data/codon_usage/bee_endosymbionts/sharp_40_highly_dataset/Bin2.ffn",genomi
  # s_index(high_cds_file="/home/yu/Data/codon_usage/bee_endosymbionts/sharp_40_highly_dataset/Bin2.ffn",gc3=0.
```

Index