Package 'GenomicSig'

January 20, 2025

Type Package

Title Computation of Genomic Signatures

Version 0.1.0

Maintainer Anu Sharma <anu.sharma@icar.gov.in>

Description

Genomic signatures represent unique features within a species' DNA, enabling the differentiation of species and offering broad applications across various fields. This package provides essential tools for calculating these specific signatures, streamlining the process for researchers and offering a comprehensive and time-saving solution for genomic analysis.The amino acid contents are identified based on the work published by Sandberg et al. (2003) <doi:10.1016/s0378-1119(03)00581-x> and Xiao et al. (2015) <doi:10.1093/bioinformatics/btv042>. The Average Mutual Information Profiles (AMIP) values are calculated based on the work of Bauer et al. (2008) <doi:10.1186/1471-2105-9-48>. The Chaos Game Representation (CGR) plot visualization was done based on the work of Deschavanne et al. (1999) <doi:10.1093/oxfordjournals.molbev.a026048> and Jeffrey et al. (1990) <doi:10.1093/nar/18.8.2163>. The GC content is calculated based on the work published by Nakabachi et al. (2006) <doi:10.1126/science.1134196> and Barbu et al. (1956) <https://pubmed.ncbi.nlm.nih.gov/13363015>. The Oligonucleotide Frequency Derived Error Gradient (OFDEG) values are computed based on the work published by Saeed et al. (2009) <doi:10.1186/1471-2164-10-S3-S10>. The Relative Synonymous Codon Usage (RSCU) values are calculated based on the work published by Elek (2018) <https://urn.nsk.hr/urn:nbn:hr:217:686131>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

RoxygenNote 7.3.2

Imports kaos, Biostrings, stats, entropy, seqinr

NeedsCompilation no

Author Mailarlinga [aut],
Shashi Bhushan Lal [aut],
Anu Sharma [aut, cre],
Dwijesh Chandra Mishra [aut],
Sudhir Srivastava [aut],
Sanjeev Kumar [aut],
Girish Kumar Jha [aut],
Sayanti Guha Majumdar [aut],
Megha Garg [aut],
Sharanbasappa [ctb],
Kabilan S [ctb]

Repository CRAN

Date/Publication 2024-09-11 16:30:02 UTC

Contents

`AminoAcidContent` *Amino Acid Content*

Description

Amino acid content refers to the relative frequencies of amino acids used in a protein or a proteome with 20 different amino acids as 20 dimensional vectors.

Usage

```
AminoAcidContent(fasta_file, type = c("DNA", "protein"))
```

Arguments

`fasta_file`	Path of a fasta file containing nucleotide or protein sequence.
`type`	Type of the sequence, can be either "DNA" or "protein".

Details

Amino acid content refers to the relative frequencies of amino acids in the protein.If DNA sequence is given as input, it will be translated to a protein sequence. Then amino acid content will be calculated.

Value

This function returns a data frame containing the sequence identifier fetched from the input fasta file and amino acid content of that sequence.

Author(s)

Dr. Anu Sharma, Megha Garg

References

Sandberg, R., Bränden, C. I., Ernberg, I., & Cöster, J. (2003). Quantifying the species-specificity in genomic signatures, synonymous codon choice, amino acid usage and G+ C content. Gene, 311, 35-42.

Xiao, N., Cao, D. S., Zhu, M. F., & Xu, Q. S. (2015). protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. Bioinformatics, 31(11), 1857-1859.

Examples

```
library(GenomicSig)
AminoAcidContent(fasta_file= system.file("extdata/Nuc_sequence.fasta", package = "GenomicSig"),
type = "DNA")
AminoAcidContent(fasta_file= system.file("extdata/prot_sequence.fasta", package = "GenomicSig"),
type = "protein")
```

AMIP *Average Mutual Information Profile (AMIP)*

Description

The Average Mutual Information Profile (AMIP) detects long-range correlations in a given DNA sequence by estimating the shared information between nucleotides situated k bases apart.

Usage

```
AMIP(fasta_file, n1 = 1, n2 = 4)
```

Arguments

`fasta_file`	Path to the input FASTA file containing the DNA sequence.
`n1`	The starting position (in bases) for Mutual Information calculation.
`n2`	The end position (in bases) for Mutual Information calculation.

Details

The Average Mutual Information (AMI) provides a statistical estimate of the shared information between nucleotides situated k bases apart in the DNA sequence, where k ranges from `n1` to `n2`. This method helps identify potential patterns or correlations in the nucleotide arrangement.

Value

This function returns a data frame containing the mutual information values for the specified nucleotide positions.

Author(s)

Dr. Anu Sharma, Dr. Shashi Bhushan Lal

References

Bauer, M., Schuster, S. M., & Sayood, K. (2008). The average mutual information profile as a genomic signature. BMC Bioinformatics, 9(1), 48.

Examples

```
library(GenomicSig)
AMIP(fasta_file = system.file("extdata/Nuc_sequence.fasta", package = "GenomicSig"), n1 = 1, n2 = 4)
```

CGR *Chaos Game Representation*

Description

Chaos game representation is the depiction of sequence in graphical form. It converts long single dimensional sequence (in this case genetic sequence) into graphical form.

Usage

```
CGR(data)
```

Arguments

data	Input as a nucleic acid sequence of characters from fasta file.

Details

This function produces visual image of DNA sequence different from the usual linear ordering of nucleotides.

Value

This function produces a chaos game representation (CGR) plot and a data frame.

Author(s)

Dr. Anu Sharma, Dr. Dwijesh Chandra Mishra

References

Deschavanne, P. J., Giron, A., Vilain, J., Fagot, G., & Fertil, B. (1999). Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. Molecular biology and evolution, 16(10), 1391-1399.

Jeffrey, H. J. (1990). Chaos game representation of gene structure. Nucleic acids research, 18(8), 2163-2170.

Examples

```
library(GenomicSig)
data(Genomicdata)
CGR(Genomicdata)
```

`GC_content` *GC content of nucleic acid sequences*

Description

This function calculates the percentage of guanine (G) or cytosine (C) nitrogenous bases in a DNA or RNA molecule. This measure indicates the proportion of G and C bases out of an implied four total bases, which including adenine (A) and thymine (T) in DNA. And adenine and uracil (U) in RNA along with G and C.

Usage

```
GC_content(sequence)
```

Arguments

`sequence`	Input as a nucleic acid sequence of characters from fasta file.

Details

G+C content is estimated with ambiguous bases taken into account.

Value

This function returns the fraction of G+C as a numeric vector of length one for all sequences.

Author(s)

Dr. Anu Sharma, Dr. Girish Kumar Jha

References

Nakabachi, A., Yamashita, A., Toh, H., Ishikawa, H., Dunbar, H. E., Moran, N. A., & Hattori, M. (2006). The 160-kilobase genome of the bacterial endosymbiont Carsonella. Science, 314(5797), 267-267.

Barbu, E., Lee, K. Y., & Wahl, R. (1956, August). Content of purine and pyrimidine base in desoxyribonucleic acid of bacteria. In Annales de l'Institut Pasteur (Vol. 91, No. 2, p. 212).

Examples

```
library(GenomicSig)
GC_content(sequence = system.file("extdata/Nuc_sequence.fasta", package = "GenomicSig"))
```

Genomicdata *Nucleotide sequence Data*

Description

A multifasta file of 34 sequences.

Usage

```
data("Genomicdata")
```

Details

Dataset contains 34 nucleic acid sequences.

Examples

```
data(Genomicdata)
```

OFDEG *Oligonucleotide Frequency Derived Error Gradient*

Description

Oligonucleotide Frequency Derived Error Gradient computes approximate convergence rate of oligonucleotide frequencies with subsequent increasing sequence length.

Usage

```
OFDEG(sequence, c, rc, d, m, t, k, norm=0)
```

Arguments

sequence	Input is a fasta file nucleic acid sequence.It accepts RData object of the fasta file
c	Minimum sequence cutoff c (which corresponds to the length of the shortest sequence in the data set). Default is 160.
rc	Cutoff of Resampling Depth (Number of subsequence of cutoff length). Default is set to 10.
d	Sampling depth (The sampling depth refers to the number of equal length subsequences randomly selected from the entire sequence). Default is set to 10. Larger sequence lengths will require greater sampling depths.
m	Word size which is initial subsequence length. Default is set to 100.
t	Step size (The step size is the change in sub-sequence length from one sampling instance to the next). Default is set to 6.
k	Size of the oligonucleotide (e.g.for tetranucleotide it is 4,for hexanucleotide it is 6). Default is set to 1.
norm	normalization of oligonucleotide frequency (OF) Profile (0 - no normalization, 1 - normalize the OF profile). Default is set to norm = 0.

Details

Oligonucleotide Frequency Derived Error Gradient (OFDEG) attempts to capture the convergence behavior by subsampling the genomic fragment and measuring the decrease in error as the length of the subsamples increases upto the fragment lenth.OFDEG, derived from the oligonucleotide frequency profile of a DNA sequence shows that it is possible to obtain a meaningful phylogenetic signal for relatively short DNA sequences.

Value

This function returns a data frame containing error gradients of each nucleotide sequence.

Author(s)

Dr. Anu Sharma, Dr. Sanjeev Kumar

References

Saeed, I., Halgamuge, S.K. The oligonucleotide frequency derived error gradient and its application to the binning of metagenome fragments. BMC Genomics 10, S10 (2009).

Examples

```
library(GenomicSig)

OFDEG(sequence= system.file("extdata/Nuc_sequence.fasta", package = "GenomicSig")[1], c=60, rc=10
, d=10, m=50, t=6, k=1, norm=0)
```

RSCU *Relative Synonymous Codon Usage*

Description

This function computes the relative frequency of each codon coding for an amino acid.

Usage

```
RSCU(sequence)
```

Arguments

sequence	Input as a nucleic acid sequence of characters from fasta file.

Details

RSCU values are the number of times a particular codon is observed, relative to the number of times that the codon would be observed for a uniform synonymous codon usage.

Value

RSCU returns the data frame with all indices.

Author(s)

Dr. Anu Sharma, Dr. Sudhir Srivastava

References

Elek, A. (2018). coRdon: an R package for codon usage analysis and prediction of gene expressivity (Master's thesis, University of Zagreb. Faculty of Science. Department of Biology).

Examples

```
library(GenomicSig)
RSCU(sequence= system.file("extdata/Nuc_sequence.fasta", package = "GenomicSig"))
```

Index