Package ‘Peptides’

October 12, 2022

Version 2.4.4
Title Calculate Indices and Theoretical Physicochemical Properties of Protein Sequences
URL https://github.com/dosorio/Peptides/
Suggests testthat (>= 2.1.0)
Description Includes functions to calculate several physicochemical properties and indices for amino-acid sequences as well as to read and plot 'XVG' output files from the 'GROMACS' molecular dynamics package.
License GPL-2
LinkingTo Rcpp
Imports Rcpp
RoxygenNote 7.1.1
Encoding UTF-8
NeedsCompilation yes
Author Daniel Osorio [aut, cre] (<https://orcid.org/0000-0003-4424-8422>), Paola Rondon-Villarreal [aut, ths] (<https://orcid.org/0000-0001-8209-3885>), Rodrigo Torres [aut, ths] (<https://orcid.org/0000-0003-1134-3020>), J. Sebastian Paez [ctb] (<https://orcid.org/0000-0002-0065-1474>), Luis Pedro Coelho [ctb] (<https://orcid.org/0000-0002-9280-7885>), Richel J.C. Bilderbeek [ctb] (<https://orcid.org/0000-0003-1107-7049>), Florian C. Sigloch [ctb] (<https://orcid.org/0000-0001-8130-6885>)
Maintainer Daniel Osorio <daniecos@uib.no>
Repository CRAN
Date/Publication 2021-05-14 15:42:11 UTC

R topics documented:

aaComp ................................................................. 2
AAdata ................................................................. 4
aaDescriptors ......................................................... 9
Compute the amino acid composition of a protein sequence

This function calculates the amount of amino acids of a particular class and classified as: Tiny, Small, Aliphatic, Aromatic, Non-polar, Polar, Charged, Basic and Acidic based on their size and R-groups using same function implemented in EMBOSS 'pepstat'. The output is a matrix with the number and percentage of amino acids of a particular class

Usage

aaComp(seq)
aaComp

Arguments

seq  An amino-acid sequence

Details

Amino acids are zwitterionic molecules with an amine and a carboxyl group present in their structure. Some amino acids possess side chains with specific properties that allow grouping them in different ways. The aaComp function classifies amino acids based on their size, side chains, hydrophobicity, charge and their response to pH 7.

Value

The output is a matrix with the number and percentage of amino acids of a particular class

- Tiny (A + C + G + S + T)
- Small (A + B + C + D + G + N + P + S + T + V)
- Aliphatic (A + I + L + V)
- Aromatic (F + H + W + Y)
- Non-polar (A + C + F + G + I + L + M + P + V + W + Y)
- Polar (D + E + H + K + N + Q + R + S + T + Z)
- Charged (B + D + E + H + K + R + Z)
- Basic (H + K + R)
- Acidic (B + D + E + Z)

Note

This function was originally written by Alan Bleasby (ajb@ebi.ac.uk) for the EMBOSS package. Further information: http://emboss.sourceforge.net/apps/cvs/emboss/apps/pepstats.html

References


Examples

# COMPARED TO PEPSSTATS
# http://emboss.bioinformatics.nl/cgi-bin/emboss/pepstats
# Property   Residues   Number   Mole%
# Tiny       (A+C+G+S+T) 4 19.048
# Small      (A+B+C+D+G+N+P+S+T+V) 4 19.048
# Aliphatic   (A+I+L+V) 5 23.810
# Aromatic    (F+H+W+Y) 5 23.810
# Non-polar   (A+C+F+G+I+L+M+P+V+W+Y) 11 52.381
# Polar       (D+E+H+K+N+Q+R+S+T+Z) 9 42.857
# Charged     (B+D+E+H+K+R+Z) 8 38.095
# Basic       (H+K+R) 8 38.095
# Acidic      (B+D+E+Z) 0 00.000
## AA composition of PDB: 1D9J Cecropin Peptide

```
aaComp(seq= "KWKLFKIGIGKFLHSAKKFX")
```

## Output

<table>
<thead>
<tr>
<th></th>
<th>Number</th>
<th>Mole %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tiny</td>
<td>4</td>
<td>19.048</td>
</tr>
<tr>
<td>Small</td>
<td>4</td>
<td>19.048</td>
</tr>
<tr>
<td>Aliphatic</td>
<td>5</td>
<td>23.810</td>
</tr>
<tr>
<td>Aromatic</td>
<td>5</td>
<td>23.810</td>
</tr>
<tr>
<td>NonPolar</td>
<td>11</td>
<td>52.381</td>
</tr>
<tr>
<td>Polar</td>
<td>9</td>
<td>42.857</td>
</tr>
<tr>
<td>Charged</td>
<td>8</td>
<td>38.095</td>
</tr>
<tr>
<td>Basic</td>
<td>8</td>
<td>38.095</td>
</tr>
<tr>
<td>Acidic</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### Description

A list with a collection of properties, scales and indices for the 20 naturally occurring amino acids from various sources.

### Usage

```
data(AAdata)
```

### Format

A list as follows:

- **Hydrophobicity** The hydrophobicity is an important stabilization force in protein folding; this force changes depending on the solvent in which the protein is found.


- crucianiProperties: The three Cruciani et. al (2004) properties, are the scaled principal component scores that summarize a broad set of descriptors calculated based on the interaction of each amino acid residue with several chemical groups (or "probes"), such as charged ions, methyl, hydroxyl groups, and so forth.
  - PP1: Polarity,
  - PP2: Hydrophobicity,
  - PP3: H-bonding

- kideraFactors: The Kidera Factors were originally derived by applying multivariate analysis to 188 physical properties of the 20 amino acids and using dimension reduction techniques. A 10-dimensional vector of orthogonal factors was then obtained for each amino acid. The first
four factors are essentially pure physical properties; the remaining six factors are superpositions of several physical properties, and are labelled for convenience by the name of the most heavily weighted component

- helix.bend.pref: Helix/bend preference
- side.chain.size: Side-chain size
- extended.str.pref: Extended structure preference
- hydrophobicity: Hydrophobicity
- double.bend.pref: Double-bend preference
- partial.spec.vol: Partial specific volume
- flat.ext.pref: Flat extended preference
- occurrence.alpha.reg: Occurrence in alpha region
- pK.C: pK-C
- surrounding.hydrop: Surrounding hydrophobicity

• pK

- EMBOSS: EMBOSS data are from http://emboss.sourceforge.net/apps/release/5.0/emboss/apps/iep.html.

• zScales The five Sandberg et al. (1998) Z-scales describe each amino acid with numerical values, descriptors, which represent the physicochemical properties of the amino acids including NMR data and thin-layer chromatography (TLC) data.

- Z1: Lipophilicity
- Z2: Steric properties (Steric bulk/Polarizability)
- Z3: Electronic properties (Polarity / Charge)
- Z4: Related to electronegativity, heat of formation, electrophilicity and hardness.
- Z5: Related to electronegativity, heat of formation, electrophilicity and hardness.

• FASGAI Factor Analysis Scale of Generalized Amino Acid Information (FASGAI) proposed by Liang and Li (2007), is a set of amino acid descriptors, that reflects hydrophobicity, alpha and turn propensities, bulky properties, compositional characteristics, local flexibility, and electronic properties, was derived from multi-dimensional properties of 20 naturally occurring amino acids.
- F1: Hydrophobicity index
- F2: Alpha and turn propensities
- F3: Bulky properties
- F4: Compositional characteristic index
- F5: Local flexibility
- F6: Electronic properties

- VHSE The principal components score Vectors of Hydrophobic, Steric, and Electronic properties, is derived from principal components analysis (PCA) on independent families of 18 hydrophobic properties, 17 steric properties, and 15 electronic properties, respectively, which are included in total 50 physicochemical variables of 20 coded amino acids.
  - VHSE1 and VHSE2: Hydrophobic properties
  - VHSE3 and VHSE4: Steric properties
  - VHSE5 to VHSE8: Electronic properties

Source
- Hydrophobicity
  - ExPASy-Protscale (http://web.expasy.org/protscale/)
  - AAIndex Database (http://www.genome.jp/aaindex/)
- pK

References
- Hydrophobicity
- crucianiProperties
- kideraFactors
• **pK**:

• **zScales**

• **FASGAI**

• **VHSE**

---

**Description**

The function return 66 amino acid descriptors for the 20 natural amino acids. Available descriptors are:

- **FASGAI**: Liang, G., & Li, Z. (2007). Factor analysis scale of generalized amino acid information as the source of a new set of descriptors for elucidating the structure and activity relationships of cationic antimicrobial peptides. Molecular Informatics, 26(6), 754-763.,
• VHSE: VHSE-scales (principal components score Vectors of Hydrophobic, Steric, and Electronic properties), is derived from principal components analysis (PCA) on independent families of 18 hydrophobic properties, 17 steric properties, and 15 electronic properties, respectively, which are included in total 50 physicochemical variables of 20 coded amino acids.


Usage

\texttt{aaDescriptors(seq)}

Arguments

\texttt{seq} \hspace{0.5cm} An amino-acids sequence. If multiple sequences are given all of them must have the same length (gap symbols are allowed.)

Value

A matrix with 66 amino acid descriptors for each aminoacid in a protein sequence.

Examples

\texttt{aaDescriptors(seq = "KLKLLLLKLLK")}

\begin{tabular}{l}
\hline
\textbf{aaList} & \textit{Return a vector with the 20 standard amino acids in upper case} \\
\hline
\end{tabular}

\begin{description}
\item[Description] This function returns a vector with the 20 standard amino acids in upper case.
\item[Usage] \texttt{aaList()}
\item[Value] A character vector with the 20 standard amino acids in upper case.
\end{description}
aaSMILES

Author(s)
Richel Bilderbeek <richel@richelbilderbeek.nl>

References

Create Smiles String from aminoacid sequences

Description
This function converts peptides with aminoacid one-letter abbreviations into smiles strings to represent the structure.

Usage
aaSMILES(seq)

Arguments
seq character vector with one-letter aminoacid codes

Details
The output can be stored in a .smi file and converted using openbabel to drawings of the peptides.

Value
character vector with smiles strings

Examples
aaSMILES("AA")
aaSMILES(c("AA", "GG"))
aIndex  

Compute the aliphatic index of a protein sequence

Description

This function calculates the Ikai (1980) aliphatic index of a protein. The aindex is defined as the relative volume occupied by aliphatic side chains (Alanine, Valine, Isoleucine, and Leucine). It may be regarded as a positive factor for the increase of thermostability of globular proteins.

Usage

aIndex(seq)

Arguments

seq  
An amino-acids sequence

Details

Aliphatic amino acids (A, I, L and V) are responsible for the thermal stability of proteins. The aliphatic index was proposed by Ikai (1980) and evaluates the thermostability of proteins based on the percentage of each of the aliphatic amino acids that build up proteins.

Value

The computed aliphatic index for a given amino-acids sequence

References


Examples

# COMPARED TO ExPASy ALIPHATIC INDEX
# http://web.expasy.org/protparam/
# SEQUENCE: SDKEVDEVDAALSLEITLE
# Aliphatic index: 117.00

aIndex(seq = "SDKEVDEVDAALSLEITLE")
# [1] 117
autoCorrelation

*Compute the auto-correlation index of a protein sequence*

**Description**

This function computes the Cruciani et al (2004) auto-correlation index. The autoCorrelation index is calculated for a lag 'd' using a descriptor 'f' (centred) over a sequence of length 'L'.

**Usage**

```r
autoCorrelation(sequence, lag, property, center = TRUE)
```

**Arguments**

- `sequence`: An amino-acids sequence
- `lag`: A value for a lag, the max value is equal to the length of shortest peptide minus one.
- `property`: A property to use as value to be correlated.
- `center`: A logical value `TRUE` or `FALSE` if the property must be centered.

**Value**

The computed auto-correlation index for a given amino-acids sequence

**References**


**Examples**

```r
# Loading a property to evaluate its autocorrelation
data(AAdata)

# Calculate the auto-correlation index for a lag=1
autoCorrelation(
    sequence = "SDKEVDEVDAALSDLEITLE",
    lag = 1,
    property = AAdata$Hydrophobicity$KyteDoolittle,
    center = TRUE
)
# [1] -0.3519908

# Calculate the auto-correlation index for a lag=5
autoCorrelation(
    sequence = "SDKEVDEVDAALSDLEITLE",
    lag = 5,
```
autoCovariance

Compute the auto-covariance index of a protein sequence

Description
This function computes the Cruciani et al (2004) auto-corvariance index. The autoCovariance index is calculated for a lag 'd' using a descriptor 'f' (centred) over a sequence of length 'L'.

Usage
autoCovariance(sequence, lag, property, center = TRUE)

Arguments
- sequence: An amino-acids sequence
- lag: A value for a lag, the max value is equal to the length of the shortest peptide minus one.
- property: A property to use as value to evaluate the covariance.
- center: A logical value TRUE or FALSE if the property must be centered.

Value
The computed auto-covariance index for a given amino-acids sequence

References

Examples
# Loading a property to evaluate its autocorrelation
data(AAdata)

# Calculate the auto-covariance index for a lag=1 autoCovariance(
  sequence = "SDKEVDEVDAALSDLEITLE",
  lag = 1,
  property = AAdata$Hydrophobicity$KyteDoolittle,
  center = TRUE
)
# [1] -0.4140053
blosumIndices

# Calculate the auto-covariance index for a lag=5
autoCovariance(
    sequence = "SDKEVDEVAALSDLEITLE",
    lag = 5,
    property = AData$Hydrophobicity$KyteDoolittle,
    center = TRUE
)
# [1] 0.001000336

blosumIndices

Compute the BLOSUM62 derived indices of a protein sequence

Description

BLOSUM indices were derived of physicochemical properties that have been subjected to a VARI-MAX analyses and an alignment matrix of the 20 natural AAs using the BLOSUM62 matrix.

Usage

blosumIndices(seq)

Arguments

seq An amino-acids sequence

Value

The computed average of BLOSUM indices of all the amino acids in the corresponding peptide sequence.

References


Examples

blosumIndices(seq = "KLKLLLLLKLK")
# [1]
# BLOSUM1  BLOSUM2  BLOSUM3  BLOSUM4  BLOSUM5
# -0.4827273 -0.5618182 -0.8509091 -0.4172727  0.3172727
# BLOSUM6  BLOSUM7  BLOSUM8  BLOSUM9  BLOSUM10
#  0.2527273  0.1463636  0.1427273 -0.2145455 -0.3218182
Compute the Boman (Potential Protein Interaction) index

Description
This function computes the potential protein interaction index proposed by Boman (2003) based in the amino acid sequence of a protein. The index is equal to the sum of the solubility values for all residues in a sequence, it might give an overall estimate of the potential of a peptide to bind to membranes or other proteins as receptors, to normalize it is divided by the number of residues. A protein have high binding potential if the index value is higher than 2.48.

Usage
```
boman(seq)
```

Arguments
- seq: An amino-acid sequence

Details
The potential protein interaction index was proposed by Boman (2003) as an easy way to differentiate the action mechanism of hormones (protein-protein) and antimicrobial peptides (protein-membrane) through this index. This function predicts the potential peptide interaction with another protein.

Value
- The computed potential protein-protein interaction for a given amino-acids sequence

References

Examples
```
# COMPARED TO YADAMP DATABASE
# http://yadamp.unisa.it/showItem.aspx?yadampid=845&x=0.4373912
# SEQUENCE: FLPVLAGLTPSVPKLVCLLKCC
# BOMAN INDEX -1.24

boman(seq = "FLPVLAGLTPSVPKLVCLLKCC")
# [1] -1.235833
```
Compute the theoretical net charge of a protein sequence

Description

This function computes the net charge of a protein sequence based on the Henderson-Hasselbalch equation described by Moore, D. S. (1985). The net charge can be calculated at defined pH using one of the 9 pKa scales availables: Bjellqvist, Dawson, EMBoss, Lehninger, Murray, Rodwell, Sillero, Solomon or Stryer.

Usage

```
charge(seq, pH = 7, pKscale = "Lehninger")
```

Arguments

- `seq`: An amino-acids sequence
- `pH`: A pH value
- `pKscale`: A character string specifying the pKa scale to be used; must be one of "Bjellqvist", "Dawson", "EMBOSS", "Lehninger", "Murray", "Rodwell", "Sillero", "Solomon" or "Stryer"

Author(s)

Original by Daniel Osorio <dcosorioh@tamu.edu>, C++ code optimized by Luis Pedro Coelho <luis@luispedro.org>

References


EMBOSS data are from http://emboss.sourceforge.net/apps/release/5.0/emboss/apps/iep.html.


Examples

# COMPARED TO EMBOSSEMBPEPSTATS
# http://emboss.bioinformatics.nl/cgi-bin/emboss/pepstats
# SEQUENCE: FLPVLAGTSPVPKLVCLLTKKC
# Charge = 3.0

charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Bjellqvist")
# [1] 2.737303
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "EMBOSS")
# [1] 2.914112
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Murray")
# [1] 2.907541
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Sillero")
# [1] 2.919812
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Solomon")
# [1] 2.844406
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Stryer")
# [1] 2.876504
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Lehninger")
# [1] 2.87315
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Dawson")
# [1] 2.844406
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "Rodwell")
# [1] 2.819755

# COMPARED TO YADAMP
# http://yadamp.unisa.it/showItem.aspx?yadampid=845&x=0,7055475
# SEQUENCE: FLPVLAGTSPVPKLVCLLTKKC
# pH5: Charge = 3.00
# pH7: Charge = 2.91
# pH9: Charge = 1.09

charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 5, pKscale= "EMBOSS")
# [1] 3.037398
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 7, pKscale= "EMBOSS")
# [1] 2.914112
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH= 9, pKscale= "EMBOSS")
# [1] 0.7184524

# JUST ONE COMMAND
charge(seq= "FLPVLAGTSPVPKLVCLLTKKC",pH = seq(from = 5,to = 9,by = 2), pKscale= "EMBOSS")
# [1] 3.0373984 2.9141123 0.7184524

crossCovariance

Compute the cross-covariance index of a protein sequence
**crossCovariance**

**Description**

This function computes the Cruciani et al (2004) cross-covariance index. The lagged crossCovariance index is calculated for a lag 'd' using two descriptors 'f1' and 'f2' (centred) over a sequence of length 'L'.

**Usage**

crossCovariance(sequence, lag, property1, property2, center = TRUE)

**Arguments**

- **sequence**: An amino-acids sequence
- **lag**: A value for a lag, the max value is equal to the length of the shortest peptide minus one.
- **property1**: A property to use as value to evaluate the cross-covariance.
- **property2**: A property to use as value to evaluate the cross-covariance.
- **center**: A logical value TRUE or FALSE if the property must be centered.

**Value**

The computed cross-covariance index for a given amino-acids sequence

**References**


**Examples**

```r
# Loading a property to evaluate its autocorrelation
data(AAdata)

# Calculate the cross-covariance index for a lag=1
crossCovariance(
  sequence = "SDKEVDEVDAALSDLEITLE",
  lag = 1,
  property1 = AAdata$Hydrophobicity$KyteDoolittle,
  property2 = AAdata$Hydrophobicity$Eisenberg,
  center = TRUE
)
# [1] -0.3026609

# Calculate the cross-correlation index for a lag=5
crossCovariance(
  sequence = "SDKEVDEVDAALSDLEITLE",
  lag = 5,
  property1 = AAdata$Hydrophobicity$KyteDoolittle,
  property2 = AAdata$Hydrophobicity$Eisenberg,
  center = TRUE
)
# [1] -0.3026609
```
crucianiProperties

**Compute the Cruciani properties of a protein sequence**

**Description**

This function calculates the Cruciani properties of an amino-acids sequence using the scaled principal component scores that summarize a broad set of descriptors calculated based on the interaction of each amino acid residue with several chemical groups (or "probes"), such as charged ions, methyl, hydroxyl groups, and so forth.

**Usage**

```r
crucianiProperties(seq)
```

**Arguments**

- `seq`  
  An amino-acids sequence

**Value**

The computed average of Cruciani properties of all the amino acids in the corresponding peptide sequence. Each PP represent an amino-acid property as follows:

- PP1: Polarity,
- PP2: Hydrophobicity,
- PP3: H-bonding

**References**


**Examples**

```r
crucianiProperties(seq = "QWGRCCGWGPRRCVRWC")
# PP1  PP2  PP3
# -0.1130 -0.0220  0.2735
```
**fasgaiVectors**  
*Compute the FASGAI vectors of a protein sequence*

**Description**

The FASGAI vectors (Factor Analysis Scales of Generalized Amino Acid Information) is a set of amino acid descriptors, that reflects hydrophobicity, alpha and turn propensities, bulky properties, compositional characteristics, local flexibility, and electronic properties, that can be utilized to represent the sequence structural features of peptides or protein motifs.

**Usage**

`fasgaiVectors(seq)`

**Arguments**

- **seq**
  - An amino-acids sequence

**Value**

The computed average of FASGAI factors of all the amino acids in the corresponding peptide sequence. Each factor represent an amino-acid property as follows:

- F1: Hydrophobicity index,
- F2: Alpha and turn propensities,
- F3: Bulky properties,
- F4: Compositional characteristic index,
- F5: Local flexibility,
- F6: Electronic properties

**References**


**Examples**

```r
fasgaiVectors(seq = "QWGRRCCGWGPGRRYCVRWC")
#       F1   F2   F3   F4   F5   F6
# -0.13675 -0.45485 -0.11695 -0.45800 -0.38015  0.52740
```
Description

This function computes the hydrophobic moment based on Eisenberg, D., Weiss, R. M., & Terwilliger, T. C. (1984). Hydrophobic moment is a quantitative measure of the amphiphilicity perpendicular to the axis of any periodic peptide structure, such as the α-helix or β-sheet. It can be calculated for an amino acid sequence of N residues and their associated hydrophobicities Hn.

Usage

hmoment(seq, angle = 100, window = 11)

Arguments

seq       An amino-acids sequence
angle     A protein rotational angle (Suggested: α-helix = 100, β-sheet=160)
window    A sequence fraction length

Details

The hydrophobic moment was proposed by Eisenberg et al. (1982), as a quantitative measure of the amphiphilicity perpendicular to the axis of any periodic peptide structure. It is computed using the standardized Eisenberg (1984) scale, windows (fragment of sequence) of eleven amino acids (by default) and specifying the rotational angle at which it should be calculated.

Value

The computed maximal hydrophobic moment (uH) for a given amino-acids sequence

Note

This function was written by an anonymous reviewer of the RJournal

References


Examples

# COMPARED TO EMBOSS:HMOMENT
# http://emboss.bioinformatics.nl/cgi-bin/emboss/hmoment
# SEQUENCE: FLPVLAGLTPSIVPKLVCLLTKKC
# ALPHA-HELIX ANGLE=100 : 0.52
# BETA-SHEET ANGLE=160 : 0.271
# ALPHA HELIX VALUE
hmoment(seq = "FLPVLAGLTPSIVPKLVCLLTKKC", angle = 100, window = 11)
# [1] 0.5199226

# BETA SHEET VALUE
hmoment(seq = "FLPVLAGLTPSIVPKLVCLLTKKC", angle = 160, window = 11)
# [1] 0.2705906

description

This function calculates the GRAVY hydrophobicity index of an amino acids sequence using one of the 38 scales from different sources.

Usage

```r
hydrophobicity(seq, scale = "KyteDoolittle")
```

Arguments

- `seq`: An amino-acids sequence
- `scale`: A character string specifying the hydophobicity scale to be used; must be one of

  - "Aboderin",
  - "AbrahamLeo",
  - "Argos",
  - "BlackMould",
  - "BullBreese",
  - "Casari",
  - "Chothia",
  - "Cid",
  - "Cowan3.4",
  - "Cowan7.5",
  - "Eisenberg",
  - "Engelman",
  - "Fasman",
  - "Fauchere",
  - "Goldsack",
  - "Guy",
  - "HoppWoods",
  - "Janin",
  - "Jones",
  - "Juretic",
  - "Kidera",
  - "Kuhn",
  - "KyteDoolittle",
  - "Levitt",
  - "Manavalan",
  - "Miyazawa",
  - "Parker",
  - "Ponnuswamy",
  - "Prabhakaran",
  - "Rao",
  - "Rose",
  - "Roseman",
  - "Sweet",
  - "Tanford",
  - "Welling",
  - "Wilson",
  - "Wolfenden",
  - "Zimmerman",
  - "interfaceScale_pH8",
  - "interfaceScale_pH2",
  - "octanolScale_pH8",
  - "octanolScale_pH2",
  - "oiScale_pH8"
  or "oiScale_pH2".

Details

The hydrophobicity is an important stabilization force in protein folding; this force changes depending on the solvent in which the protein is found. The hydrophobicity index is calculated adding the hydrophobicity of individual amino acids and dividing this value by the length of the sequence.

Value

The computed GRAVY index for a given amino-acid sequence
References


Examples

# COMPARED TO GRAVY Grand average of hydropathicity (GRAVY) ExPASy
# http://web.expasy.org/cgi-bin/protparam/protparam
# SEQUENCE: QWGRRCCGWPGRRYCRWC
# GRAVY: -0.950

hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Aboderin")
# [1] 3.84
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "AbrahamLeo")
# [1] 0.092
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Argos")
# [1] 1.033
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "BlackMould")
# [1] 0.50125
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "BullBreese")
# [1] 0.1575
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Casari")
# [1] 0.38
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Chothia")
# [1] 0.262
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Cid")
# [1] 0.198
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Cowan3.4")
# [1] 0.0845
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Cowan7.5")
# [1] 0.0605
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Eisenberg")
# [1] -0.3265
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Engelman")
# [1] 2.31
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Fasman")
# [1] -1.2905
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Fauchere")
# [1] 0.527
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Goldsack")
# [1] 1.2245
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Guy")
# [1] 0.193
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "HoppWoods")
# [1] -0.14
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Janin")
# [1] -0.105
hydrophobicity(seq = "QWGRRCCGWPGRRYCRWC",scale = "Jones")
# instaIndex

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Scale</th>
<th>Instability Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Juretic</td>
<td>1.4675</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Kidera</td>
<td>-1.106</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Kuhn</td>
<td>-0.0405</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>KyteDoolittle</td>
<td>-0.0405</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Leivitt</td>
<td>-0.21</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Manavalan</td>
<td>13.0445</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Miyazawa</td>
<td>5.739</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Parker</td>
<td>1.095</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Ponnuswamy</td>
<td>0.851</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Prabhakaran</td>
<td>9.67</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Rao</td>
<td>0.813</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Rose</td>
<td>0.7575</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Roseman</td>
<td>-0.495</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Sweet</td>
<td>-0.1135</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Tanford</td>
<td>-0.2905</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Welling</td>
<td>-0.666</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Wilson</td>
<td>3.16</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Wolfenden</td>
<td>3.16</td>
</tr>
<tr>
<td>QWGRRCCGWGPGRRYCVWC</td>
<td>Zimmerman</td>
<td>0.943</td>
</tr>
</tbody>
</table>

**Description**

This function calculates the instability index proposed by Guruprasad (1990). This index predicts the stability of a protein based on its amino acid composition, a protein whose instability index is smaller than 40 is predicted as stable, a value above 40 predicts that the protein may be unstable.
Usage

instaIndex(seq)

Arguments

seq An amino-acids sequence

Value

The computed instability index for a given amino-acids sequence

References


Examples

# COMPARED TO ExPASy INSTAINDEX
# http://web.expasy.org/protparam/
# SEQUENCE: QWGRRCCGWPGRRYCVWC
# The instability index (II) is computed to be 83.68

instaIndex(seq = "QWGRRCCGWPGRRYCVWC")
# [1] 83.68

kideraFactors

Compute the Kidera factors of a protein sequence

Description

The Kidera Factors were originally derived by applying multivariate analysis to 188 physical properties of the 20 amino acids and using dimension reduction techniques. This function calculates the average of the ten Kidera factors for a protein sequence.

Usage

kideraFactors(seq)

Arguments

seq An amino-acids sequence
Value
A list with the average of the ten Kidera factors. The first four factors are essentially pure physical properties; the remaining six factors are superpositions of several physical properties, and are labelled for convenience by the name of the most heavily weighted component.

- **KF1**: Helix/bend preference,
- **KF2**: Side-chain size,
- **KF3**: Extended structure preference,
- **KF4**: Hydrophobicity,
- **KF5**: Double-bend preference,
- **KF6**: Partial specific volume,
- **KF7**: Flat extended preference,
- **KF8**: Occurrence in alpha region,
- **KF9**: pK-C,
- **KF10**: Surrounding hydrophobicity

References

Examples
```r
kideraFactors(seq = "KLKLLLLKLL")
# [1]
# [1,1] KF1   KF2    KF3    KF4    KF5
#        -0.78545 0.29818 -0.23636 -0.08182 0.21000
#        -1.89364 1.02909 -0.51273 0.11182 0.81000
```

lengthpep

*Compute the amino acid length of a protein sequence*

Description
This function counts the number of amino acids in a protein sequence

Usage
```r
lengthpep(seq)
```

Arguments
```r
seq  An amino-acids sequence
```
**Details**

All proteins are formed by linear chains of small residues known as amino acids attached to each other by peptide bonds. The function `lengthpep` counts the number of amino acids in a sequence and returns a vector with the count for each peptide used as argument.

**Examples**

```r
data <- c("QWGRRCCGWGPGRRYCVRWC")
lengthpep(data)
# [1] 20
```

**massShift**

_Calculate the mass difference of modified peptides._

**Description**

This function calculates the mass difference of peptides introduced by chemical modifications or heavy isotope labelling.

**Usage**

```r
massShift(seq, label = "none", aaShift = NULL, monoisotopic = TRUE)
```

**Arguments**

- `seq`: An amino-acids sequence, in one letter code.
- `label`: Set a predefined heavy isotope label. Accepts "none", "silac_13c", "silac_13c15n" and "15n". Overwrites input in `aaShift`.
- `aaShift`: Define the mass difference in Dalton of given amino acids as a named vector. Use the amino acid one letter code as names and the mass shift in Dalton as values. N-terminal and C-terminal modifications can be defined by using "Nterm =" and "Cterm =", respectively.
- `monoisotopic`: A logical value ‘TRUE’ or ‘FALSE’ indicating if monoisotopic weights of amino-acids should be used.

**Source**

For the predefined heavy isotope labels, compare:

- silac_13c Unimod 188
- silac_13c15n Unimod 259 and Unimod 267
- 15n Unimod 994, Unimod 995, Unimod 996 and Unimod 897
Examples

```r
massShift("EGVNDNECEGFFSAR", label = "silac_13c")
massShift("EGVNDNECEGFFSAR", aaShift = c(K = 6.020129, R = 6.020129))
```

### membpos

Compute theoretically the class of a protein sequence

**Description**

This function calculates the theoretical class of a protein sequence based on the relationship between the hydrophobic moment and hydrophobicity scale proposed by Eisenberg (1984).

### Usage

```r
membpos(seq, angle = 100)
```

**Arguments**

- `seq`: An amino-acids sequence
- `angle`: A protein rotational angle

### Details

Eisenberg et al. (1982) found a correlation between hydrophobicity and hydrophobic moment that defines the protein section as globular, transmembrane or superficial. The function calculates the hydrophobicity (H) and hydrophobic moment (uH) based on the standardized scale of Eisenberg (1984) using windows of 11 amino acids for calculate the theoretical fragment type.

### Value

A data frame for each sequence given with the calculated class for each window of eleven amino-acids

### References


Examples

```r
membpos(seq = "ARQQNLFINFCLILIFLLLI", angle = 100)
# Pep   H   uH  MembPos
# 1 ARQQNLFINFCL 0.083 0.353  Globular
# 2 RQQNLFINFCLI 0.147 0.317  Globular
# 3 QQNLFINFCLIL 0.446 0.274  Globular
# 4 QNLFINFCLILI 0.632 0.274  Transmembrane
# 5 NLFINFCLILIF 0.802 0.253  Surface
# 6 LFINCLIFILFL 0.955 0.113  Transmembrane
# 7 FINCLIFILLLL 0.955 0.113  Transmembrane
# 8 INFCLILIFLLL 0.944 0.108  Transmembrane
# 9 NFCLILIFLLLI 0.944 0.132  Transmembrane
```

```r
membpos(seq = "ARQQNLFINFCLILIFLLLI", angle = 160)
# Pep   H   uH  MembPos
# 1 ARQQNLFINFCL 0.083 0.467  Globular
# 2 RQQNLFINFCLI 0.147 0.467  Globular
# 3 QQNLFINFCLILI 0.446 0.285  Globular
# 4 QNLFINFCLILLI 0.632 0.358  Surface
# 5 NLFINFCLILIF 0.802 0.358  Surface
# 6 LFINCLIFILFL 0.955 0.269  Surface
# 7 FINCLIFILLLL 0.955 0.269  Surface
# 8 INFCLILIFLLL 0.944 0.257  Surface
# 9 NFCLILIFLLLLI 0.944 0.229  Surface
```

---

**mswhimScores**

*Compute the MS-WHIM scores of a protein sequence*

**Description**

MS-WHIM scores were derived from 36 electrostatic potential properties derived from the three-dimensional structure of the 20 natural amino acids.

**Usage**

```r
mswhimScores(seq)
```

**Arguments**

- `seq` An amino-acids sequence

**Value**

The computed average of MS-WHIM scores of all the amino acids in the corresponding peptide sequence.
mw

References

Examples
mswhimScores(seq = "KLKLLLLLKLK")
# [[1]]
# MSWHIM1 MSWHIM2 MSWHIM3
# -0.6563636 0.4872727 0.1163636

mw

Compute the molecular weight of a protein sequence

Description
This function calculates the molecular weight of a protein sequence. It is calculated as the sum of the mass of each amino acid using the scale available on Compute pI/Mw tool. It also supports mass calculation of proteins with predefined or custom stable isotope mass labels.

Usage
mw(
  seq,
  monoisotopic = FALSE,
  avgScale = "expasy",
  label = "none",
  aaShift = NULL
)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq</td>
<td>An amino-acids sequence</td>
</tr>
<tr>
<td>monoisotopic</td>
<td>A logical 'TRUE' or 'FALSE' indicating if monoisotopic weights of amino-acids should be used</td>
</tr>
<tr>
<td>avgScale</td>
<td>Set the mass scale to use for average weight only (if 'monoisotopic == FALSE'). Accepts &quot;expasy&quot; (default) or &quot;mascot&quot;.</td>
</tr>
<tr>
<td>label</td>
<td>Set a predefined heavy isotope label. Accepts &quot;none&quot;, &quot;silac_13c&quot;, &quot;silac_13c15n&quot; and &quot;15n&quot;. Overwrites input in aaShift.</td>
</tr>
<tr>
<td>aaShift</td>
<td>Define the mass difference in Dalton of given amino acids as a named vector. Use the amino acid one letter code as names and the mass shift in Dalton as values.</td>
</tr>
</tbody>
</table>
Details

The molecular weight is the sum of the masses of each atom constituting a molecule. The molecular weight is directly related to the length of the amino acid sequence and is expressed in units called daltons (Da). In Peptides the function mw computes the molecular weight using the same formulas and weights as ExPASy's compute mw tool (Gasteiger et al., 2005). For average weight, the ExPASy tools use the following mass scale: https://web.expasy.org/findmod/findmod_masses.html#AA, while UniMod and Mascot use a slightly different one: http://www.matrixscience.com/help/aa_help.html.

Source

The formula and amino acid scale are the same available on ExPASy Compute pi/mw tool: http://web.expasy.org/compute_pi/.

References


Examples

# COMPARED TO ExPASy Compute pi/Mw tool
# http://web.expasy.org/compute_pi/
# SEQUENCE: QWGRRCCGWGPGRRYCVRWC
# Theoretical pi/Mw: 9.88 / 2485.91

mw(seq = "QWGRRCCGWGPGRRYCVRWC", monoisotopic = FALSE)
# [1] 2485.911

mw(seq = "QWGRRCCGWGPGRRYCVRWC", monoisotopic = TRUE)
# [1] 2484.12

mz

Calculate the m/z for peptides.

Description

This function calculates the (monoisotopic) mass over charge ratio (m/z) for peptides, as measured in mass spectrometry.

Usage

mz(seq, charge = 2, label = "none", aaShift = NULL, cysteins = 57.021464)
Arguments

- **seq**: An amino-acids sequence, in one letter code.
- **charge**: The net charge for which the m/z should be calculated.
- **label**: Set a predefined heavy isotope label. Accepts "none", "silac_13c", "silac_13c15n" and "15n". Overwrites input in aaShift.
- **aaShift**: Define the mass difference in Dalton of given amino acids as a named vector. Use the amino acid one letter code as names and the mass shift in Dalton as values.
- **cysteins**: Define the mass shift in Dalton of blocked cysteins. Defaults to 57.021464, for cysteins blocked by iodoacetamide.

Examples

```r
mz("EGVNDNECEGFFSAR")
mz("EGVNDNECEGFFSAR", aaShift = c(K = 6.020129, R = 6.020129))
mz("EGVNDNECEGFFSAR", label = "silac_13c", cysteins = 58.005479)
```

pepdata

**Physicochemical properties and indices from 100 amino acid sequences**

Description

Physicochemical properties and indices from 100 amino acid sequences (50 antimicrobial and 50 non antimicrobial)

Usage

```r
data(pepdata)
```

Format

A data frame with 100 observations on the following 23 variables.

- **sequence**: a character vector with the sequences of 100 peptides (50 antimicrobial and 50 non-antimicrobial)
- **group**: Integer vector with the group code "0" for non antimicrobial and "1" for antimicrobial
- **length**: a numeric vector with the length of the amino acid sequence
- **mw**: a numeric vector with the molecular weight of the amino acid sequence
- **tinyAA**: A numeric vector with the fraction (as percent) of tiny amino acids that make up the sequence
- **smallAA**: A numeric vector with the fraction (as percent) of small amino acids that make up the sequence
Compute the isoelectric point (pI) of a protein sequence

**pI**

The isoelectric point (pI), is the pH at which a particular molecule or surface carries no net electrical charge.

**Usage**

```
pI(seq, pKscale = "EMBOSS")
```
Arguments

- **seq**: An amino-acids sequence
- **pKscale**: A character string specifying the pK scale to be used; must be one of "Bjellqvist", "EMBOSS", "Murray", "Sillero", "Solomon", "Stryer", "Lehninger", "Dawson" or "Rodwell"

Details

The isoelectric point (pI) is the pH at which the net charge of the protein is equal to 0. It is a variable that affects the solubility of the peptides under certain conditions of pH. When the pH of the solvent is equal to the pI of the protein, it tends to precipitate and lose its biological function.

Examples

```
# COMPARED TO ExPASy ProtParam
# http://web.expasy.org/cgi-bin/protparam/protparam
# SEQUENCE: QWGRCCGWPGRRYCVRWC
# Theoretical pI: 9.88

pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Bjellqvist")
# [1] 9.881

# COMPARED TO EMBOSS PEPSTATS
# http://emboss.bioinformatics.nl/cgi-bin/emboss/pepstats
# SEQUENCE: QWGRCCGWPGRRYCVRWC
# Isoelectric Point = 9.7158

pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "EMBOSS")
# [1] 9.716

# OTHER SCALES

pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Murray")
# [1] 9.818
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Sillero")
# [1] 9.891
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Solomon")
# [1] 9.582
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Stryer")
# [1] 9.623
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Lehninger")
# [1] 9.937
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Dawson")
# [1] 9.568
pI(seq= "QWGRCCGWPGRRYCVRWC",pKscale= "Rodwell")
# [1] 9.718
```
plotXVG

Description

Read and plot output data from a XVG format file.

Usage

plotXVG(XVGfile, ...)

Arguments

XVGfile
A .XVG output file of the GROMACS molecular dynamics package

Arguments to be passed to methods, such as graphical parameters.

Details

GROMACS (GROningen MAchine for Chemical Simulations) is a molecular dynamics package designed for simulations of proteins, lipids and nucleic acids. It is free, open source software released under the GNU General Public License. The file format used by GROMACS is XVG. This format can be displayed in graphical form through the GRACE program on UNIX/LINUX systems and the GNUPlot program on Windows. XVG files are plain text files containing tabular data separated by tabulators and two types of comments which contain data labels. Although manual editing is possible, this is not a viable option when working with multiple files of this type. For ease of reading, information management and data plotting, the functions read.xvg and plot.xvg were incorporated.

Author(s)

Latest: J. Sebastian Paez <jpaezpae@purdue.edu>
Original: Daniel Osorio <dcosorioh@unal.edu.co>

References


Examples

XVGfile <- system.file("xvg-files/epot.xvg",package="Peptides")
plotXVG(XVGfile)
**protFP**

**Compute the protFP descriptors of a protein sequence**

**Description**

The ProtFP descriptor set was constructed from a large initial selection of indices obtained from the AAindex database for all 20 naturally occurring amino acids.

**Usage**

`protFP(seq)`

**Arguments**

*seq*  
An amino-acids sequence

**Value**

The computed average of protFP descriptors of all the amino acids in the corresponding peptide sequence.

**References**


**Examples**

```r
protFP(seq = "QWGRRCCGWGPGRRYCVRWC")
# [1] 
# ProtFP1 ProtFP2 ProtFP3 ProtFP4 ProtFP5 ProtFP6 ProtFP7 ProtFP8 
# 0.2065 -0.0565 1.9930 -0.2845 0.7315 0.7000 0.1715 0.1135
```

**readXVG**

*Read output data from a XVG format file.*

**Description**

XVG is the default format file of the GROMACS molecular dynamics package, contains data formatted to be imported into the Grace 2-D plotting program.

**Usage**

`readXVG(file)`
Arguments

file       A .XVG output file of the GROMACS molecular dynamics package

Details

GROMACS (GROningen MAchine for Chemical Simulations) is a molecular dynamics package designed for simulations of proteins, lipids and nucleic acids. It is free, open source software released under the GNU General Public License. The file format used by GROMACS is XVG. This format can be displayed in graphical form through the GRACE program on UNIX/LINUX systems and the GNUPlot program on Windows. XVG files are plain text files containing tabular data separated by tabulators and two types of comments which contain data labels. Although manual editing is possible, this is not a viable option when working with multiple files of this type. For ease of reading, information management and data plotting, the functions `read.xvg` and `plot.xvg` were incorporated.

Author(s)

Latest: J. Sebastian Paez <jpaezpae@purdue.edu> and hongbo-zhu-cn <@github>

Original: Daniel Osorio <dcosorioh@unal.edu.co>

References


Examples

```r
# READING FILE
XVGfile <- system.file("xvg-files/epot.xvg",package="Peptides")
readXVG(XVGfile)

    #    Time (ps) Potential
    # 1 1 6672471040
    # 2 2 6516461568
    # 3 3 6351947264
    # 4 4 6183133184
    # 5 5 6015310336
    # 6 6 5854271488
```

stScales

Compute the ST-scales of a protein sequence

Description

ST-scales were proposed by Yang et al, taking 827 properties into account which are mainly constitutional, topological, geometrical, hydrophobic, electronic, and steric properties of a total set of 167 AAs.
Usage

\texttt{stScales(seq)}

Arguments

seq \hspace{1cm} \text{An amino-acids sequence}

Value

The computed average of ST-scales of all the amino acids in the corresponding peptide sequence.

References


Examples

\begin{verbatim}
  stScales(seq = "QWGRRCGWPGRRYCVWC")
  # [1]
  # ST1 ST2 ST3 ST4 ST5 ST6 ST7 ST8
  # -0.63760 0.07965 0.05150 0.07135 -0.27905 -0.80995 0.58020 0.54400
\end{verbatim}

\textbf{tScales} \hspace{1cm} Compute the T-scales of a protein sequence

Description

T-scales are based on 67 common topological descriptors of 135 amino acids. These topological descriptors are based on the connectivity table of amino acids alone, and to not explicitly consider 3D properties of each structure.

Usage

\texttt{tScales(seq)}

Arguments

seq \hspace{1cm} \text{An amino-acids sequence}

Value

The computed average of T-scales of all the amino acids in the corresponding peptide sequence.

References

Examples

```r
tScales(seq = "QWGRRCGWRGRRYCVRWC")
# [[1]]
# T1   T2   T3   T4   T5
# -3.2700 -0.0035 -0.3855 -0.1475 0.7585
```

vhseScales

Compute the VHSE-scales of a protein sequence

Description

VHSE-scales (principal components score Vectors of Hydrophobic, Steric, and Electronic properties), is derived from principal components analysis (PCA) on independent families of 18 hydrophobic properties, 17 steric properties, and 15 electronic properties, respectively, which are included in total 50 physicochemical variables of 20 coded amino acids.

Usage

`vhseScales(seq)`

Arguments

- `seq`: An amino-acids sequence

Value

The computed average of VHSE-scales of all the amino acids in the corresponding peptide sequence. Each VSHE-scale represent an amino-acid property as follows:

- VHSE1 and VHSE2: Hydrophobic properties
- VHSE3 and VHSE4: Steric properties
- VHSE5 to VHSE8: Electronic properties

References


Examples

```r
vhseScales(seq = "QWGRRCGWRGRRYCVRWC")
# [[1]]
# VHSE1 VHSE2 VHSE3 VHSE4 VHSE5 VHSE6 VHSE7 VHSE8
# -0.1150 0.0630 -0.0055 0.7955 0.4355 0.2485 0.1740 -0.0960
```
zScales

Compute the Z-scales of a protein sequence

Description

Z-scales are based on physicochemical properties of the AAs including NMR data and thin-layer chromatography (TLC) data.

Usage

zScales(seq)

Arguments

seq An amino-acids sequence

Value

The computed average of Z-scales of all the amino acids in the corresponding peptide sequence. Each Z scale represent an amino-acid property as follows:

- Z1: Lipophilicity
- Z2: Steric properties (Steric bulk/Polarizability)
- Z3: Electronic properties (Polarity / Charge)
- Z4 and Z5: They relate electronegativity, heat of formation, electrophilicity and hardness.

References


Examples

zScales(seq = "QWGRCCGWGPYRRYCVRWC")
# [[1]]
# Z1  Z2  Z3  Z4  Z5
# 0.6200 0.0865 0.0665 0.7280 -0.8740
Index

aaComp, 2
AAdata, 4
aaDescriptors, 9
aaList, 10
aaSMILES, 11
aIndex, 12
autoCorrelation, 13
autoCovariance, 14

blosumIndices, 15
boman, 16

charge, 17
crossCovariance, 18
crucianiProperties, 20

fasgaiVectors, 21

hmoment, 22
hydropobicity, 23

instaIndex, 27
kideraFactors, 28

lengthpep, 29

massShift, 30
membpos, 31
mswhimScores, 32
mw, 33
mz, 34

pepdata, 35
pI, 36
plotXVG, 38
protFP, 39

readXVG, 39
stScales, 40
tScales, 41
vhseScales, 42
zScales, 43