Package 'ampir'

November 14, 2019

Type Package

Title Predict Antimicrobial Peptides

Version 0.1.0

Author Legana Fingerhut

Maintainer Legana Fingerhut <legana.fingerhut@my.jcu.edu.au>

Description A toolkit to predict Antimicrobial Peptides from protein sequences.

URL https://github.com/Legana/ampir

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports Peptides, caret (>= 6.0.0), kernlab, Rcpp

RoxygenNote 6.1.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp

NeedsCompilation yes

R topics documented:

`calculate_features` *Calculate a set of numerical features from protein sequences*

Description

This function calculates set physicochemical and compositional features from protein sequences

Usage

```
calculate_features(df)
```

Arguments

`df`	A dataframe which contains protein sequence names as the first column and amino acid sequence as the second column

Value

A dataframe containing numerical values related to the protein features of each given protein

example

my_protein <- readRDS(system.file("extdata/my_protein_df.rds", package = "ampir"))

Calculate features from Hepcidin AMP from *Myotis lucifugus* (UniProt ID G1P6H5)

calculate_features(my_protein)

Output (showing the first six output columns) # seq_name Amphiphilicity Hydrophobicity pI Mw Charge .... # [1] G1P6H5_MYOLU 0.4145847 0.4373494 8.501312 9013.757 4.53015 ....

Note

This function depends on the Peptides package

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015).

calc_amphiphilicity *Calculate amphiphilicity (or hydrophobic moment)*

Description

Calculate amphiphilicity (or hydrophobic moment)

Usage

```
calc_amphiphilicity(seq)
```

Arguments

seq	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

calc_composition *Calculate the amino acid composition*

Description

Calculate the amino acid composition

Usage

```
calc_composition(seq)
```

Arguments

seq	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

calc_hydrophobicity *Calculate the hydrophobicity*

Description

Calculate the hydrophobicity

Usage

```
calc_hydrophobicity(seq)
```

Arguments

seq	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

calc_length *Calculate the length of a protein*

Description

Calculate the length of a protein

Usage

```
calc_length(seq)
```

Arguments

seq	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

`calc_mw` *Calculate the molecular weight*

Description

Calculate the molecular weight

Usage

```
calc_mw(seq)
```

Arguments

`seq`	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

`calc_net_charge` *Calculate the net charge*

Description

Calculate the net charge

Usage

```
calc_net_charge(seq)
```

Arguments

`seq`	A protein sequence

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

calc_pI *Calculate the isoelectric point (pI)*

Description

Calculate the isoelectric point (pI)

Usage

```
calc_pI(seq)
```

Arguments

seq	pI

References

Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A package for data mining of antimicrobial peptides. The R Journal. 7(1), 4–14 (2015). The imported function originates from the Peptides package (https://github.com/dosorio/Peptides/).

calc_pseudo_comp *Calculate the pseudo amino acid composition*

Description

This function is adapted from the extractPAAC function from the protr package (https://github.com/nanxstats/protr)

Usage

```
calc_pseudo_comp(seq, lambda_min = 4, lambda_max = 19)
```

Arguments

seq	A vector of protein sequences as character strings
lambda_min	Minimum allowable lambda. It is an error to provide a protein sequence shorter than lambda_min
lambda_max	For each sequence lambda will be set to one less than the sequence length or lambda_max, whichever is smaller

References

Nan Xiao, Dong-Sheng Cao, Min-Feng Zhu, and Qing-Song Xu. (2015). protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. Bioinformatics 31 (11), 1857-1859.

df_to_faa *Save a dataframe in FASTA format*

Description

This function writes a dataframe out as a FASTA format file

Usage

```
df_to_faa(df, file = "")
```

Arguments

df	a dataframe containing two columns (seq.name and seq.aa)
file	file path to save the named file to

Value

A FASTA file where protein sequences are represented in two lines: The protein name preceded by a greater than symbol, and a new second line that contains the protein sequence

Examples

```
# Use \code{read_faa} to read a FASTA file as a dataframe
my_protein <- read_faa(system.file("extdata/bat_protein.fasta", package = "ampir"))

# Use \code{df_to_faa} to write a dataframe into FASTA file format
df_to_faa(my_protein,(system.file("extdata/my_protein.fasta", package = "ampir")))


## Output written in "my_protein.fasta"
#[1] >G1P6H5_MYOLU
#[2] MALTVRIQAACLLLLLLASLTSYSLLLSQTTQLADLQTQDTAGATAGLMPGLQRRRRRDTHFPICIFCCGCCYPSKCGICCKT
```

extract_amps *Extract predicted antimicrobial peptides (AMPs)*

Description

This function extracts the protein sequences predicted to be AMPs from `predict_amps`

Usage

```
extract_amps(df_w_seq, df_w_prob, prob = 0.5)
```

Arguments

df_w_seq	a dataframe containing two columns (sequence name and sequence) (output from read_faa)
df_w_prob	a dataframe containing two columns (sequence name and AMP probability) (output from predict_amps)
prob	The greater than or equal to probability value AMP identification should be set at default is 0.50

Value

A FASTA file where protein sequences are represented in two lines: The protein name preceded by a greater than symbol, and a new second line that contains the protein sequence

Examples

```
my_protein <- readRDS(system.file("extdata/my_protein_df.rds", package = "ampir"))
my_prediction <- readRDS(system.file("extdata/my_protein_pred.rds", package = "ampir"))

extract_amps(my_protein, my_prediction, prob = 0.55)

#' ## Output
#         seq_name              seq_aa
# [1] G1P6H5_MYOLU  MALTVRIQAACLLLLLLASLTSYSL....
```

---

predict_amps *Predict the antimicrobial peptide probability of a protein*

Description

This function predicts the probability of a protein to be an antimicrobial peptide

Usage

```
predict_amps(faa_df)
```

Arguments

faa_df	A dataframe obtained from read_faa) containing two columns: the sequence name (seq_name) and amino acid sequence (seq_aa)

Value

A dataframe containing a column with the sequence name and probability of that sequence to be an antimicrobial peptide

Examples

```
my_bat_faa_df <- read_faa(system.file("extdata/bat_protein.fasta", package = "ampir"))

predict_amps(my_bat_faa_df)
#       seq_name    prob_AMP
# [1] G1P6H5_MYOLU  0.9723796
```

random_aa *Create a random amino acid sequence of given length*

Description

This function results in a single sequence of any given length. It is also a helper function for `random_aas`.

Usage

```
random_aa(x)
```

Arguments

`x`	Sequence length for random_aa

Value

A character vector of given sequence length

random_aas *Create multiple random amino acid sequences of given length*

Description

This function uses a helper function (see `random_aa`) and creates random amino acid sequences of a given number and length range

Usage

```
random_aas(n, min_length = 10, max_length = 2000)
```

Arguments

`n`	The number of sequences
`min_length, max_length`	The minimum and maximum length of the sequence (default is 10 and 2000, respectively)

Value

A character vector of given number and length range

read_faa *Read FASTA amino acid file into a dataframe*

Description

This function reads a FASTA amino acids file into a dataframe

Usage

```
read_faa(file = NULL)
```

Arguments

file	file path to the FASTA format file containing the protein sequences

Value

Dataframe containing the sequence name (seq.name) and sequence (seq.aa) columns

Note

This function was originally written by Jinlong Zhang (jinlongzhang01@gmail.com) for the phylotools package (http://github.com/helixcn/phylotools)

Examples

```
read_faa(system.file("extdata/bat_protein.fasta", package = "ampir"))

## Output
#         seq_name              seq_aa
# [1] G1P6H5_MYOLU  MALTVRIQAACLLLLLLASLTSYSL....
```

remove_nonstandard_aa *Remove non standard amino acids from protein sequences*

Description

This function removes anything that is not one of the 20 standard amino acids in protein sequences

Usage

```
remove_nonstandard_aa(df)
```

Arguments

df	A dataframe which contains protein sequence names as the first column and amino acid sequence as the second column

Value

a dataframe like the input dataframe but with removed proteins that contained non standard amino acids

Examples

```
# non_standard_df <- readRDS(system.file("extdata/non_standard_df.rds", package = "ampir"))

## Example dataframe (non_standard_df)
# non_standard_df
#       seq_name            seq_aa
# [1] G1P6H5_MYOLU    MALTVRIQAACLLLLLLASLTSYSLLLSQTTQLADLQTQ....
# [2] fake_sequence   MKVTHEUSYR$GXMBIJIDG*M80-%

# remove_nonstandard_aa(non_standard_df)

## Output
#       seq_name        seq_aa
# [1] G1P6H5_MYOLU    MALTVRIQAACLLLLLLASLTSYSLLLSQTTQLADLQTQ....
```

`rsvm_classify` *Predict the antimicrobial peptide probability of a protein*

Description

This function predicts the probability of a protein to be an antimicrobial peptide based on feature calculations (as obtained from `calculate_features`)

Usage

```
rsvm_classify(df)
```

Arguments

`df`	A dataframe containing numerical features (as calculated by `calculate_features`)

Value

A dataframe containing a single column with probability values

example

my_protein_features <- readRDS(system.file("extdata/my_protein_features.rds", package = "ampir"))

rsvm_classify(my_protein_features) # seq_name prob_AMP # [1] G1P6H5_MYOLU 0.9723796

Note

The predictive model within this function was created via the caret package (https://github.com/topepo/caret/)

Index