Package 'surfaltr'

April 10, 2025

Type Package

Title Rapid Comparison of Surface Protein Isoform Membrane Topologies Through surfaltr

Version 1.13.0

Description Cell surface proteins form a major fraction of the druggable proteome and can be used for tissue-specific delivery of oligonucleotide/cell-based therapeutics. Alternatively spliced surface protein isoforms have been shown to differ in their subcellular localization and/or their transmembrane (TM) topology. Surface proteins are hydrophobic and remain difficult to study thereby necessitating the use of TM topology prediction methods such as TMHMM and Phobius. However, there exists a need for bioinformatic approaches to streamline batch processing of isoforms for comparing and visualizing topologies. To address this gap, we have developed an R package, surfaltr. It pairs inputted isoforms, either known alternatively spliced or novel, with their APPRIS annotated principal counterparts, predicts their TM topologies using TMHMM or Phobius, and generates a customizable graphical output. Further, surfaltr facilitates the prioritization of biologically diverse isoform pairs through the incorporation of three different ranking metrics and through protein alignment functions. Citations for programs mentioned here can be found in the vignette.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr (>= 1.0.6), biomaRt (>= 2.46.0), protr (>= 1.6-2), seqinr (>= 4.2-5), ggplot2 (>= 3.3.2), utils (>= 2.10.1), stringr (>= 1.4.0), Biostrings (>= 2.58.0),readr (>= 1.4.0), httr (>= 1.4.2), testthat(>= 3.0.0), xml2(>= 1.3.2), msa (>= 1.22.0), methods (>= 4.0.3)

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, devtools, kableExtra

VignetteBuilder knitr

Depends R (>= 4.0)

NeedsCompilation no

biocViews Software, Visualization, DataRepresentation, SplicedAlignment, Alignment, MultipleSequenceAlignment, MultipleComparison

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/surfaltr

git_branch devel

git_last_commit 98fe9a8

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2025-04-09

Author Pooja Gangras [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0638-3941>),
Aditi Merchant [aut]

Maintainer Pooja Gangras <gangras_pooja@lilly.com>

Contents

align_org_prts . . . 3
align_prts . . . 4
check_tmhmm_install . . . 5
clean_data . . . 5
Crb1 . . . 6
ensembl_db_retrieval . . . 7
format_ids . . . 7
get_aas . . . 8
get_pairs . . . 8
get_phobius . . . 9
get_prts . . . 11
get_tmhmm . . . 12
graph_from_aas . . . 13
graph_from_ids . . . 14
graph_prots . . . 15
hpa_genes . . . 16
hpa_mouse_genes . . . 16
merge_trans . . . 17
plot_isoforms . . . 17
process_tmhmm . . . 18
rank_prts . . . 19
run_phobius . . . 20
split_fasta . . . 21
test_surfaltr . . . 22
tmhmm_fix_path . . . 22

Index **24**

`align_org_prts` *Get aligned amino acid sequences for gene transcripts from multiple organisms*

Description

This function allows a user to specify genes of interest and subsequently receive a pdf of all the corresponding aligned human and mouse amino acid sequences. In order for this to work, transcripts for the same genes from both organisms need to be provided in separate files.

Usage

```
align_org_prts(gene_names, hs_data_file, mm_data_file, if_aa = FALSE,
temp = FALSE)
```

Arguments

`gene_names`	Vector containing names of genes of interest (e.g. c("Crb1", "Adgrl1"))
`hs_data_file`	Path to the input file containing the human transcripts
`mm_data_file`	Path to the input file containing the mouse transcripts
`if_aa`	Boolean value indicating if the input file contains amino acid sequence. TRUE indicates that sequences are present and FALSE indicates that only IDs are present.
`temp`	Boolean indicating if the fasta file should be saved to the working directory or no

Value

Nothing is returned.

Note

Although the function returns nothing, it saves pdfs containing the aligned sequences to the working directory under a file labeled with the gene name. It's also important to note that although the gene names will be standardized to be fully capitalized, this may not match with the case of the gene name for some organisms.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {align_org_prts( c("IGSF1"),
system.file("extdata", "hpa_example.csv", package = "surfaltr"),
system.file("extdata", "hpa_mouse_example.csv",
package = "surfaltr"),
FALSE, TRUE)}
```

`align_prts` *Get aligned amino acid sequences for gene transcripts*

Description

This function allows a user to specify genes of interest and subsequently receive a pdf of all the corresponding aligned amino acid sequences in pdf format.

Usage

```
align_prts(gene_names, data_file, if_aa = FALSE, organism = "human",
temp = FALSE)
```

Arguments

`gene_names`	Vector containing names of genes of interest (e.g. c(Crb1, Adgrl1))
`data_file`	Path to the input file
`if_aa`	Boolean value indicating if the input file contains amino acid sequence. TRUE indicates that sequences are present and FALSE indicates that only IDs are present.
`organism`	String indicating if the transcripts are from a human or a mouse
`temp`	Boolean indicating if the fasta file should be saved to the working directory or no

Value

Nothing is returned.

Note

Although the function returns nothing, it saves pdfs containing the aligned sequences to the working directory under a file labeled with the gene name.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    align_prts(c("Crb1"), system.file("extdata", "crb1_example.csv",
        package = "surfaltr"
    ), TRUE, "mouse", TRUE)
}
```

check_tmhmm_install *Check to make sure TMHMM 2.0 is installed in the file path specified*

Description

This function checks to make sure that TMHMM is installed correctly at the file path specified by the user. If TMHMM is not installed correctly, then the function will output an error message telling the user to check their installation.

Usage

```
check_tmhmm_install(tmhmm_folder_name)
```

Arguments

`tmhmm_folder_name`
: Full path to folder containing installed TMHMM 2.0 software. This value should end in TMHMM2.0c

Value

A Boolean stating if TMHMM is installed correctly, will be TRUE if TMHMM 2.0 is located at the path specified and FALSE if it is not.

Note

This function also prints a helpful method providing tips on how to fix the installation if TMHMM is not found at the folder path specified.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
install_correct <- check_tmhmm_install(tmhmm_folder_name)
```

clean_data *Retrieve, Clean, and Format Input Data*

Description

This function cleans and formats input data. The cleaning and formatting portion involves removing any non-protein coding transcripts, removing any principal transcripts, and standardizing all column names. If the sequence is provided directly, the function also extracts the APPRIS annotation and UniProt IDs of each transcript from Ensembl. Provided data can follow 2 formats — the first option only contain transcript IDs and gene names and the second option contains a unique transcript identifier, gene names, and amino acid sequences. The function will return a data frame containing the transcript IDs, gene names, and APPRIS Annotation for each inputted transcript. If the amino acid sequence is included in the input data, this will also be included in the data frame. If only gene names and transcript IDS are provided, UniProt IDs will be included in the data frame.

Usage

```
clean_data(data_file, if_aa, organism)
```

Arguments

data_file	Path to the input file
if_aa	Boolean value indicating if the input file contains amino acid sequences with TRUE indicating that sequences are present and FALSE indicating that only IDs are present
organism	String indicating if the transcripts are from a human or a mouse

Value

A data frame containing gene names, transcript IDs, and APPRIS annotations for the given data. If sequences were provided, the data frame will also contain amino acid sequences. If only IDs were provided, the data frame will also contain the UniProt Swissprot ID, UniProt Swissprot isoform ID, and UniProt TREMBL ID.

Crb1 *SurfaltR Amino Acid Test Data- Novel Mouse Retina Isoforms*

Description

For the amino acid input, we have utilized the supplementary data 1 from Ray et al 2020 (ref). This data includes novel isoforms expressed in mouse retina identified by long read sequencing and further validated by cell surface proteomics approaches. The data has been formatted to be compatible with the package.

Usage

```
Crb1
```

Format

A data frame with 36 rows and 3 variables:

external_gene_name Gene name corresponding to amino acid sequence

transcript_id transcript ID corresponding to amino acid sequence

protein_sequence amino acid sequence of transcript ...

Source

https://www.nature.com/articles/s41467-020-17009-7#Sec45

`ensembl_db_retrieval` *Retrieve Transcript Information from Ensembl for all Primary Transcripts*

Description

This function retrieves all the primary transcripts in the given organism and their corresponding gene names, APPRIS annotations, and UniProt IDs.

Usage

```
ensembl_db_retrieval(organism)
```

Arguments

`organism`	String indicating if mouse or human transcripts should be retrieved

Value

A data frame containing the gene names, transcript IDs, APPRIS annotations, UniProt Swissprot IDs, UniProt Swissprot isoform IDs, and UniProt TREMBL IDs for all the primary transcripts in an organism.

`format_ids` *Reformat transcripts to facilitate fasta file conversion*

Description

Modify format of data to display all primary and alternative transcripts from the same gene together and remove any duplicates.

Usage

```
format_ids(final_pairs)
```

Arguments

`final_pairs`	Data frame containing original row-wise pairings of primary and alternative transcripts for inputted data without associated sequences

Value

A data frame containing the gene names, transcript IDs, APPRIS annotations, UniProt Swissprot IDs, UniProt Swissprot isoform IDs, and UniProt TREMBL IDs for all the given and associated primary transcripts in an alternating fashion

get_aas — *Create fasta file containing amino acid sequences based on user sequences*

Description

This function creates a fasta file with the transcript ID followed by the amino acid sequence for all inputted and associated primary transcripts. The file is organized so that all transcripts from a gene are next to each other. The function also returns a final table containing the gene names, transcript IDs, APPRIS annotations, and amino acid sequences for each transcript

Usage

```
get_aas(final_pairs, temp = FALSE)
```

Arguments

final_pairs	A data frame containing gene names, transcript IDs, amino acid sequences, and APPRIS annotations for all inputted data and its corresponding primary transcripts.
temp	Boolean indicating if the fasta file should be deleted after the function finishes running or not. Recommended to always be set to FALSE.

Value

A data frame containing the gene names, transcript IDs, APPRIS annotations, and protein sequences for each transcript.

Note

This function also creates a fasta file containing the transcript IDs and associated amino acid sequences in the root directory.

get_pairs — *Create csv and fasta files containing information about pairs of transcripts*

Description

This function processes the input data to retrieve information from ensembl and uniprot to generate a dataframe containing the gene names, transcript IDs, APPRIS annotations, and protein sequences for each pair of primary and alternative transcripts. Additionally, this function creates a fasta file with the transcript ID followed by the amino acid sequence for all inputted and associated primary transcripts. The file is organized so that all transcripts from a gene are next to each other. Finally, the function also produces a final table in csv form containing the gene names, transcript IDs, APPRIS annotations, and amino acid sequences for each transcript

Usage

```
get_pairs(data_file, if_aa = FALSE, organism = "human", temp = FALSE)
```

Arguments

data_file	Path to the input file
if_aa	Boolean value indicating if the input file contains amino acid sequences with TRUE indicating that sequences are present and FALSE indicating that only IDs are present
organism	String indicating if the transcripts are from a human or a mouse
temp	Boolean indicating if the fasta file should be deleted after the function finishes running or not. Recommended to always be set to FALSE.

Value

A data frame containing the gene names, transcript IDs, APPRIS annotations,and protein sequences for each pair of primary and alternative transcripts.

Note

This function also creates a fasta file containing the transcript IDs and associated amino acid sequences in the root directory. In addition to the fasta file, a csv file containing the returned dataframe is saved to the working directory.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    currwd <- getwd()
    AA_seq <- get_pairs(system.file("extdata", "crb1_example.csv",
        package = "surfaltr"
    ), TRUE, "mouse", TRUE)
    setwd(currwd)
}
```

---

get_phobius *Query Phobius web server.*

Description

Phobius web server is a combined transmembrane topology and signal peptide (N-sp) predictor. Currently only "normal prediction" of signal peptides is supported by the function.

Usage

```
get_phobius(data, ...)

## S3 method for class 'character'
get_phobius(data, progress = FALSE, ...)

## S3 method for class 'data.frame'
get_phobius(data, sequence, id, ...)

## S3 method for class 'list'
get_phobius(data, ...)

## Default S3 method:
get_phobius(data = NULL, sequence, id, ...)
```

Arguments

data	A data frame with protein amino acid sequences as strings in one column and corresponding id's in another. Alternatively a path to a .fasta file with protein sequences. Alternatively a list with elements of class "SeqFastaAA" resulting from `read.fasta` call. Should be left blank if vectors are provided to sequence and id arguments.
...	currently no additional arguments are accepted apart the ones documented bellow.
progress	Boolean, whether to show the progress bar, at default set to FALSE.
sequence	A vector of strings representing protein amino acid sequences, or the appropriate column name if a data.frame is supplied to data argument. If .fasta file path, or list with elements of class "SeqFastaAA" provided to data, this should be left blank.
id	A vector of strings representing protein identifiers, or the appropriate column name if a data.frame is supplied to data argument. If .fasta file path, or list with elements of class "SeqFastaAA" provided to data, this should be left blank.

Details

The topology (prediction column of the result) is given as the position of the transmembrane helices separated by 'i' if the loop is on the cytoplasmic or 'o' if it is on the non-cytoplasmic side. A signal peptide is given by the position of its h-region separated by a n and a c, and the position of the last amino acid in the signal peptide and the first of the mature protein separated by a /.

Value

A data frame with columns:

Name Character, name of the submitted sequence.

tm Integer, the number of predicted transmembrane segments.

sp Character, Y/0 indicator if a signal peptide was predicted or not.

prediction Character string, predicted topology of the protein.

cut_site Integer, first amino acid after removal of the signal peptide

is.phobius Logical, did Phobius predict the presence of a signal peptide

Note

This function creates temporary files in the working directory.

Source

https://phobius.sbc.su.se/

References

Kall O. Krogh A. Sonnhammer E. L. L. (2004) A Combined Transmembrane Topology and Signal Peptide Prediction Method. Journal of Molecular Biology 338(5): 1027-1036

---

get_prts *Create fasta file containing amino acid sequences based on IDs*

Description

This function creates a fasta file with the transcript ID followed by the amino acid sequence for all given alternative transcripts and associated primary transcripts. The file is organized so that all transcripts from a gene are next to each other. The function also returns a final table containing the gene names, transcript IDs, APPRIS annotations, and amino acid sequences for each transcript

Usage

```
get_prts(aa_trans, temp = FALSE)
```

Arguments

aa_trans	A data frame containing the gene names, transcript IDs, APPRIS annotations, UniProt Swissprot IDs, UniProt Swissprot isoform IDs, and UniProt TREMBL IDs for all transcripts.
temp	Boolean indicating if the fasta file should be deleted after the function finishes running or not. Recommended to always be set to FALSE.

Value

A data frame containing the gene names, transcript IDs, APPRIS annotations, UniProt IDs, and protein sequences for each transcript.

Note

This function also creates a fasta file containing the transcript IDs and associated amino acid sequences in the root directory.

`get_tmhmm` *Create a data frame with the membrane locations of each amino acid in a protein using TMHMM*

Description

This function creates a data frame with columns containing transcript IDs and corresponding output from TMHMM. The TMHMM output includes a location for each amino acid, with O and o representing extracellular, M representing transmembrane, and i representing intracellular.

Usage

```
get_tmhmm(fasta_file_name, tmhmm_folder_name)
```

Arguments

`fasta_file_name`
: Name of .fasta file containing amino acid sequences

`tmhmm_folder_name`
: Full path to folder containing installed TMHMM 2.0 software. This path should end in TMHMM2.0c

Value

A data frame containing each transcript ID and the corresponding membrane location for each amino acid in its sequence formatted as a string

Note

In order for this function to work, there needs to be a .fasta file containing the amino acid sequences for each transcript called "AA.fasta" saved to a folder called output within the working directory. Additionally, the file saves a copy of the returned data frame in csv format to the output folder in the working directory.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    AA_seq <- get_pairs(system.file("extdata", "crb1_example.csv",
        package = "surfaltr"
    ), TRUE, "mouse", TRUE)
    topo <- get_tmhmm("AA.fasta", tmhmm_folder_name)
}
```

graph_from_aas *Create a plot showing membrane locations of each protein based on user provided amino acid sequences*

Description

This function creates a ggplot figure showing the differences in membrane location and length between primary and alternative transcripts from the same gene. This process is performed based on input data containing the gene names and amino acid sequences of the proteins in question. Transcripts derived from the same gene are grouped together to facilitate easy interpretation. The y axis lists the gene name and transcript ID for each transcript and the x axis lists the length in amino acids. Each fill color corresponds to a membrane location and either principal or alternative isoform.

Usage

```
graph_from_aas(data_file, organism = "human", rank = "length",
n_prts = 20, mode = "phobius", size_txt = 2, space_left = -400, temp = FALSE,
tmhmm_folder_name = NULL)
```

Arguments

data_file	Path to the input file
organism	String indicating if the transcripts are from a human or a mouse
rank	String indicating which method to use to rank proteins in graphical output. Options include "Length", "TM", and "Combo".
n_prts	Integer value indicating the number of genes that should be displayed on the graphical output. Default value is 20.
mode	String detailing whether TMHMM or Phobius should be used to predict transmembrane regions. Input values include "phobius" or "tmhmm".
size_txt	Integer value specifying the size of the row labels. Default size is 2.
space_left	Integer value specifying how far left the graph should extend.
temp	Boolean indicating if the fasta file should be deleted after the function finishes running or not. Recommended to always be set to FALSE.
tmhmm_folder_name	Full path to folder containing installed TMHMM 2.0 software. This value should end in TMHMM2.0c and needs to be provided if the mode used is TMHMM.

Value

A ggplot figure showing the protein locations for each part of the surface protein for each alternative and primary transcripts.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    graph_from_aas(
        system.file("extdata", "crb1_example.csv", package = "surfaltr"),
        "mouse", "combo", 1, "tmhmm", 4, -300, TRUE
    )
}
```

`graph_from_ids` *Create a plot showing membrane locations of each protein based on transcript IDs*

Description

This function creates a ggplot figure showing the differences in membrane location and length between primary and alternative transcripts from the same gene. This process is performed based on input data containing the gene names and transcript IDs of the proteins in question. Transcripts derived from the same gene are grouped together to facilitate easy interpretation. The y axis lists the gene name and transcript ID for each transcript and the x axis lists the length in amino acids. Each fill color corresponds to a membrane location and either principal or alternative isoform.

Usage

```
graph_from_ids(data_file, organism = "human", rank = "length",
n_prts = 20, mode = "phobius", size_txt = 2, space_left = -400, temp = FALSE,
tmhmm_folder_name = NULL)
```

Arguments

`data_file`	Path to the input file
`organism`	String indicating if the transcripts are from a human or a mouse
`rank`	String indicating which method to use to rank proteins in graphicl output. Options include "Length", "TM", and "Combo".
`n_prts`	Integer value indicating the number of genes that should be displayed on the graphical output. Default value is 20.
`mode`	String detailing whether TMHMM or Phobius should be used to predict transmembrane regions. Input values include "phobius" or "tmhmm".
`size_txt`	Integer value specifying the size of the row labels. Default size is 2.
`space_left`	Integer value specifying how far left the graph should extend.
`temp`	Boolean indicating if the fasta file should be deleted after the function finishes running or not. Recommended to always be set to FALSE.
`tmhmm_folder_name`	Full path to folder containing installed TMHMM 2.0 software. This value should end in TMHMM2.0c and needs to be provided if the mode used is TMHMM.

Value

A ggplot figure showing the protein locations for each part of the surface protein for each alternative and primary transcripts.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    graph_from_ids(
        system.file("extdata", "hpa_example.csv", package = "surfaltr"),
        "human", "length", 1, "tmhmm", 5, -300, TRUE
    )
}
```

---

graph_prots *Create a plot showing where each amino acid is located within the cell for each primary transcript compared to each alternative transcript*

Description

This function creates a ggplot figure showing the differences in membrane location and length between primary and alternative transcripts from the same gene. Transcripts derived from the same gene are grouped together to facilitate easy interpretation. The y axis lists the gene name and transcript ID for each transcript and the x axis lists the length in amino acids. Each fill color corresponds to a membrane location and either principal or alternative isoform.

Usage

```
graph_prots(counts, rank = "length", n_prts = 20, size_txt = 2,
space_left = -400)
```

Arguments

counts	A data frame containing the overall length and individual lengths of each section of the surface protein corresponding to a certain transcript.
rank	String indicating which method to use to rank proteins in graphical output. Options include "Length", "TM", and "Combo".
n_prts	Integer value indicating the number of genes that should be displayed on the graphical output. Default value is 20.
size_txt	Integer value specifying the size of the row labels. Default size is 2.
space_left	Integer value specifying how far left the graph should extend.

Value

A ggplot figure showing the protein locations for each part of the surface protein for each alternative and primary transcripts.

hpa_genes — *SurfaltR Gene Name and Transcript ID Test Data- Highly Expressed Human Alternative Transcripts*

Description

For the gene name and transcript ID input, we have included 10 unique human transcripts from 7 different genes annotated as alternative by APPRIS. These genes were derived from supplementary data 12 from Uhlén et all 2015. This data has been formatted to be compatible with the package.

Usage

```
hpa_genes
```

Format

A data frame with 10 rows and 2 variables:

gene_name Gene name corresponding to transcript ID

transcript transcript ID of gene of interest ...

Source

https://science.sciencemag.org/content/347/6220/1260419/tab-figures-data

hpa_mouse_genes — *SurfaltR Gene Name and Transcript ID Test Data- Highly Expressed Mouse Alternative Transcripts*

Description

For the gene name and transcript ID input, we have included 5 unique mouse transcripts from 5 different genes annotated as alternative by APPRIS. These genes were derived from supplementary data 12 from Uhlén et all 2015. This data has been formatted to be compatible with the package and to match the genes in the HPA human gene dataset.

Usage

```
hpa_mouse_genes
```

Format

A data frame with 5 rows and 2 variables:

gene_name Gene name corresponding to transcript ID

transcript transcript ID of gene of interest ...

Source

https://science.sciencemag.org/content/347/6220/1260419/tab-figures-data

---

`merge_trans` *Associate Inputted Transcripts with Corresponding Primary Transcripts*

Description

This function matches each inputted transcript with its corresponding primary transcripts and returns a data frame containing the gene name, transcript ID and APPRIS annotation for each.

Usage

```
merge_trans(princ, final_trans, if_aa)
```

Arguments

`princ`	Data frame containing all primary transcripts and relevant gene information for an organism
`final_trans`	Data frame containing cleaned and formatted input data
`if_aa`	Boolean value indicating if the input file contains amino acid sequences with TRUE indicating that sequences are present and FALSE indicating that only IDs are present

Value

A data frame containing gene names, transcript IDs, and APPRIS annotations for all inputted data and its corresponding primary transcripts. If sequences were provided, the data frame will also contain the amino acid sequences. If only IDs were provided, the data frame will also contain the UniProt Swissprot ID, UniProt Swissprot isoform ID, and UniProt TREMBL ID for both the inputted data and the primary transcripts.

`plot_isoforms` *Create a plot showing where each amino acid is located within the cell for each primary transcript compared to each alternative transcript*

Description

This function creates a ggplot figure showing the differences in membrane location and length between primary and alternative transcripts from the same gene. Transcripts derived from the same gene are grouped together to facilitate easy interpretation. The y axis lists the gene name and transcript ID for each transcript and the x axis lists the length in amino acids. Each fill color corresponds to a membrane location and either principal or alternative isoform.

Usage

```
plot_isoforms(topo, AA_seq, rank = "length", n_prts = 20,
size_txt = 2, space_left = -400)
```

Arguments

topo	Outputted data frame from the run_phobius or get_tmhmm function showing membrane locations of amino acids and transcript IDs
AA_seq	A data frame outputted by the get_pairs function containing the gene names, transcript IDs, APPRIS annotations, and protein sequences for each transcript.
rank	String indicating which method to use to rank proteins in graphicl output. Options include "length", "TM", and "combo".
n_prts	Integer value indicating the number of genes that should be displayed on the graphical output. Default value is 20.
size_txt	Integer value specifying the size of the row labels. Default size is 2.
space_left	Integer value specifying how far left the graph should extend.

Value

A ggplot figure showing the protein locations for each part of the surface protein for each alternative and primary transcripts.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    currwd <- getwd()
    AA_seq <- get_pairs(system.file("extdata", "crb1_example.csv",
        package = "surfaltr"
    ), TRUE, "mouse", TRUE)
    topo <- run_phobius(AA_seq, paste(getwd(), "/AA.fasta", sep = ""))
    plot_isoforms(topo, AA_seq, "combo", 15, 3, -400)
    setwd(currwd)
}
```

---

process_tmhmm *Create a data frame with the membrane locations of each amino acid in a sequence*

Description

This function creates a data frame with columns containing transcript IDs and corresponding output from tmhmm. The tmhmm output includes a location for each amino acid, with O and o representing extracellular, M representing transmembrane, and i representing intracellular. The data frame includes columns with the transcript ID, membrane location, gene name, starting amino acid, and ending amino acid for a certain transcript. The first row for each transcript contains the overall length of the amino acid sequence.

Usage

```
process_tmhmm(topo, AA_seq)
```

Arguments

topo	A data frame containing each transcript ID and the corresponding membrane location for each amino acid in its sequence formatted as a string.
AA_seq	A data frame containing the gene names, transcript IDs, APPRIS annotations, and protein sequences for each transcript.

Value

A data frame containing the overall length and individual lengths of each section of the surface protein corresponding to a certain transcript.

---

rank_prts *Rank the surface proteins by differences in principal and alternative isoforms*

Description

This function creates a data frame containing the primary and alternative transcripts of each gene ranked by how different the resultant surface proteins are. Transcripts can be ranked by length, number of transmembrane domains, or a combo metric that multiplied the difference in length by the number of transmembrane domains and ranks accordingly. This function can also be set to restrict the number of genes that are returned to the user to show only the most significant gene transcripts.

Usage

```
rank_prts(counts, rank, n_prts)
```

Arguments

counts	A data frame containing the overall length and individual lengths of each section of the surface protein corresponding to a certain transcript.
rank	String indicating which method to use to rank proteins in graphic output. Options include "Length", "TM", and "Combo".
n_prts	Integer value indicating the number of genes that should be displayed on the graphical output. Default value is 20.

Value

A data frame containing the overall length and individual lengths of each section of the surface protein corresponding to a certain transcript ranked by how different the primary and alternative transcripts are functionally.

run_phobius *Create a data frame with the membrane locations of each amino acid in a protein using Phobius*

Description

This function creates a data frame with columns containing transcript IDs and corresponding output from Phobius. The Phobius output includes a location for each amino acid, with O representing extracellular, M representing transmembrane, S representing signal, and i representing intracellular.

Usage

```
run_phobius(AA_seq, fasta_file_name)
```

Arguments

AA_seq	A data frame outputted by the get_pairs function containing the gene names, transcript IDs, APPRIS annotations, and protein sequences for each transcript.
fasta_file_name	Path to fasta file containing amino acid sequences

Value

A data frame containing each transcript ID and the corresponding membrane location for each amino acid in its sequence formatted as a string

Note

In order for this function to work, there needs to be a .fasta file containing the amino acid sequences for each transcript called "AA.fasta" saved to the working directory. Additionally, the file saves a copy of the returned data frame in csv format to the working directory.

Examples

```
tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
    currwd <- getwd()
    AA_seq <- get_pairs(system.file("extdata", "crb1_example.csv",
        package = "surfaltr"
    ), TRUE, "mouse", TRUE)
    topo <- run_phobius(AA_seq, paste(getwd(), "/AA.fasta", sep = ""))
    setwd(currwd)
}
```

`split_fasta` *Split a fasta formatted file.*

Description

The function splits a fasta formatted file to a defined number of smaller .fasta files for further processing.

Usage

```
split_fasta(
  path_in,
  path_out,
  num_seq = 20000,
  trim = FALSE,
  trunc = NULL,
  id = FALSE
)
```

Arguments

`path_in`	A path to the .FASTA formatted file that is to be processed.
`path_out`	A path where the resulting .FASTA formatted files should be stored. The path should also contain the prefix name of the fasta files on which _n (integer from 1 to number of fasta files generated) will be appended along with the extension ".fa"
`num_seq`	Integer defining the number of sequences to be in each resulting .fasta file. Defaults to 20000.
`trim`	Logical, should the sequences be trimmed to 4000 amino acids to bypass the CBS server restrictions. Defaults to FALSE.
`trunc`	Integer, truncate the sequences to this length. First 1:trunc amino acids will be kept.
`id`	Logical, should the protein id's be returned. Defaults to FALSE.

Value

if id = FALSE, A Character vector of the paths to the resulting .FASTA formatted files.

if id = TRUE, A list with two elements:

id Character, protein identifiers.

file_list Character, paths to the resulting .FASTA formatted files.

test_surfaltr *Test the functionality of surfaltr*

Description

This function runs all of surfaltr's other functions on the CRB1 data set to ensure that the function output matches the expected output. An incorrect output or error indicates that something went wrong in installation.

Usage

```
test_surfaltr()
```

Value

Nothing is returned.

Note

If the results from the test match the expected results, a message stating that the test worked will be printed. If not, the user will be prompted to check the installation

Examples

```
 tmhmm_folder_name <- "~/TMHMM2.0c"
if (check_tmhmm_install(tmhmm_folder_name)) {
test_surfaltr()
}
```

tmhmm_fix_path *Retrieve Data from TMHMM and Fix Functionality of TMHMM R Package*

Description

This function retrieves the raw data from tmhmm containing information about the membrane location of each amino acid in a transcript. In order to set a standard path that allows tmhmm to run, the path is set to match that of the fasta file contining the amino acids.

Usage

```
tmhmm_fix_path(fasta_filename, folder_name)
```

Arguments

fasta_filename	Parameter containing input fasta file to be run on tmhmm
folder_name	Path to folder containing installed tmhmm software

Value

Raw results from tmhmm containing membrane locations for each transcript

Note

In order for this function to work, there needs to be a .fasta file containing the amino acid sequences for each transcript called "AA.fasta" saved to a folder called output within the working directory.

Index

∗ **datasets**
    Crb1, 6
    hpa_genes, 16
    hpa_mouse_genes, 16

align_org_prts, 3
align_prts, 4

check_tmhmm_install, 5
clean_data, 5
Crb1, 6

ensembl_db_retrieval, 7

format_ids, 7

get_aas, 8
get_pairs, 8
get_phobius, 9
get_prts, 11
get_tmhmm, 12
graph_from_aas, 13
graph_from_ids, 14
graph_prots, 15

hpa_genes, 16
hpa_mouse_genes, 16

merge_trans, 17

plot_isoforms, 17
process_tmhmm, 18

rank_prts, 19
read.fasta, *10*
run_phobius, 20

split_fasta, 21

test_surfaltr, 22
tmhmm_fix_path, 22