Package 'pureseqtmr'

May 21, 2020

Title Predict Transmembrane Helices

Version 0.3

Description Proteins reside in either the cell plasma of in the cell membrane. A membrane protein goes through the membrane at least once. There are multiple ways to span this hydrophobic layer. One common structure is the transmembrane (alpha) helix (TMH). Given the amino acid sequence of a membrane protein, this package predicts which parts of the protein are TMHs.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

Imports ggplot2, rappdirs, stringr, tibble

Suggests testthat

URL `https://github.com/richelbilderbeek/pureseqtmr`

BugReports `https://github.com/richelbilderbeek/pureseqtmr`

NeedsCompilation no

Author Richèl J.C. Bilderbeek [aut, cre]
(<https://orcid.org/0000-0003-1107-7049>)

Maintainer Richèl J.C. Bilderbeek `<richel@richelbilderbeek.nl>`

R topics documented:

`are_tmhs` *Are the sequences TMHs?*

Description

Are the sequences TMHs?

Usage

```
are_tmhs(protein_sequences, folder_name = get_default_pureseqtm_folder())
```

Arguments

`protein_sequences`	one or more protein sequences
`folder_name`	superfolder of PureseqTM. The superfolder's name is `/home/[user_name]/.local/share` by default, as can be obtained by get_default_pureseqtm_folder

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  sequences <- c(
    "QEKNWSALLTAVVIILTIAGNILVIMAVSLEKKLQNATNYFLM",
    "VVIILTIRGNILVIMAVSLE"
  )
  expect_equal(c(TRUE, FALSE), are_tmhs(sequences))
}
```

---

check_pureseqtm_installation

Checks the installation of PureseqTM. Throws a helpful error message if incomplete, else does nothing

---

Description

Checks the installation of PureseqTM. Throws a helpful error message if incomplete, else does nothing

Usage

```
check_pureseqtm_installation(folder_name = get_default_pureseqtm_folder())
```

Arguments

folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Value

Nothing

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  expect_silent(check_pureseqtm_installation())
} else {
  expect_error(check_pureseqtm_installation())
}
```

check_topology *Check if the topology is valid. Will stop if not.*

Description

Check if the topology is valid. Will stop if not.

Usage

```
check_topology(topology)
```

Arguments

topology	the topology as a tibble as returned by predict_proteome_topology

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  topology <- predict_proteome_topology(fasta_filename)
  expect_silent(check_topology(topology))
}
```

create_pureseqtm_files *Create the five PureseqTM output files, by running PureseqTM.*

Description

Create the five PureseqTM output files, by running PureseqTM.

Usage

```
create_pureseqtm_files(
  fasta_filename,
  folder_name = get_default_pureseqtm_folder(),
  temp_folder_name = tempfile(pattern = "pureseqt_")
)
```

Arguments

fasta_filename	path to a FASTA file
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder
temp_folder_name	path of a temporary folder. The folder does not need to exist. Files that are out in this folder are not automatically deleted, which is not a problem, as the default path given by tempdir is automatically cleaned by the operating system

Value

full path to the files created

Author(s)

Richèl J.C. Bilderbeek

See Also

use run_pureseqtm to received also the parsed output

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  filenames <- create_pureseqtm_files(fasta_filename)
  expect_equal(5, length(filenames))
}
```

---

create_pureseqtm_proteome_file

Create the output file of a PureseqTM proteome run

Description

Create the output file of a PureseqTM proteome run

Usage

```
create_pureseqtm_proteome_file(
  fasta_filename,
  topology_filename = tempfile(fileext = ".top"),
  folder_name = get_default_pureseqtm_folder()
)
```

Arguments

fasta_filename	path to a FASTA file
topology_filename	name of the file to save a protein's topology to
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Value

the filename

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  topology_filename <- create_pureseqtm_proteome_file(fasta_filename)
  expect_equal(1, length(topology_filename))
  expect_true(file.exists((topology_filename)))
  expect_equal(0, length(readLines(topology_filename)) %% 3)
}
```

---

default_params_doc *This function does nothing. It is intended to inherit is parameters' documentation.*

Description

This function does nothing. It is intended to inherit is parameters' documentation.

Usage

```
default_params_doc(
  download_url,
  fasta_filename,
  fasta_file_text,
  folder_name,
  protein_sequence,
  protein_sequences,
  pureseqtm_filename,
  pureseqtm_result,
  pureseqtm_url,
```

```
  temp_folder_name,
  topology,
  topology_filename,
  verbose
)
```

Arguments

download_url	the URL to download PureseqTM from
fasta_filename	path to a FASTA file
fasta_file_text	text of a FASTA file
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder
protein_sequence	a protein sequence
protein_sequences	one or more protein sequences
pureseqtm_filename	filename to write the PureseqTM results to
pureseqtm_result	the result of a PureseqTM run
pureseqtm_url	URL of the PureseqTM git repository
temp_folder_name	path of a temporary folder. The folder does not need to exist. Files that are out in this folder are not automatically deleted, which is not a problem, as the default path given by tempdir is automatically cleaned by the operating system
topology	the topology as a tibble as returned by predict_proteome_topology
topology_filename	name of the file to save a protein's topology to
verbose	set to TRUE for more output

Note

This is an internal function, so it should be marked with @noRd. This is not done, as this will disallow all functions to find the documentation parameters

Author(s)

Richèl J.C. Bilderbeek

get_default_pureseqtm_folder

Get the path to the folder where this package installs PureseqTM by default

Description

Get the path to the folder where this package installs PureseqTM by default

Usage

```
get_default_pureseqtm_folder()
```

Value

the path to the folder where this package installs PureseqTM by default

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (rappdirs::app_dir()$os == "unix") {
  expect_true(
    grepl(
      "/home/[A-Za-z0-9_]*/.local/share",
      get_default_pureseqtm_folder()
    )
  )
}
```

get_example_filename

Get the full path to a PureseqTM example files

Description

Get the full path to a PureseqTM example files

Usage

```
get_example_filename(filename, folder_name = get_default_pureseqtm_folder())
```

Arguments

filename	name of the example file, without the path
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Author(s)

Richèl J.C. Bilderbeek

See Also

use get_example_filenames to get all PureseqTM example filenames

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  expect_true(file.exists(get_example_filename("1bhaA.fasta")))
}
```

---

get_example_filenames

Get the full path to all PureseqTM example files

Description

Get the full path to all PureseqTM example files

Usage

```
get_example_filenames(folder_name = get_default_pureseqtm_folder())
```

Arguments

folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Value

a character vector with all PureseqTM example files

Author(s)

Richèl J.C. Bilderbeek

See Also

use get_example_filename to get the full path to a PureseqTM example file

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  filenames <- get_example_filenames()
  expect_true(all(file.exists(filenames)))
}
```

---

get_pureseqtm_url *Get the URL of the PureseqTM source code*

Description

Get the URL of the PureseqTM source code

Usage

```
get_pureseqtm_url()
```

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

url <- get_pureseqtm_url()
expect_equal(1, length(url))
```

---

install_pureseqtm *Install PureseqTM to a local folder*

Description

Install PureseqTM to a local folder

Usage

```
install_pureseqtm(
  folder_name = get_default_pureseqtm_folder(),
  pureseqtm_url = get_pureseqtm_url()
)
```

Arguments

folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder
pureseqtm_url	URL of the PureseqTM git repository

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_on_travis() && !is_pureseqtm_installed()) {
  install_pureseqtm()
  expect_true(is_pureseqtm_installed())
}
```

---

is_on_travis *Determines if the environment is Travis CI*

Description

Determines if the environment is Travis CI

Usage

```
is_on_travis()
```

Value

TRUE if run on Travis CI, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

is_protein_name_line

Is the line of text the name of a protein, as used within a FASTA filename?

Description

Is the line of text the name of a protein, as used within a FASTA filename?

Usage

```
is_protein_name_line(line)
```

Arguments

line	line of text from a FASTA filename

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_protein_name_line(">5H2A_CRIGR"))
expect_false(is_protein_name_line("5H2A_CRIGR"))
expect_false(is_protein_name_line("000001111100000"))
expect_false(is_protein_name_line(NA))
expect_false(is_protein_name_line(NULL))
expect_false(is_protein_name_line(""))
```

is_pureseqtm_installed

Measure if PureseqTM is installed locally

Description

Measure if PureseqTM is installed locally

Usage

```
is_pureseqtm_installed(folder_name = get_default_pureseqtm_folder())
```

Arguments

folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Value

TRUE is PureseqTM is installed locally, FALSE otherwise

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

is_installed <- is_pureseqtm_installed()
expect_true(is_installed == TRUE || is_installed == FALSE)
```

---

is_tmh *Determine if the protein sequence contains at least one TMH.*

Description

Determine if the protein sequence contains at least one TMH.

Usage

```
is_tmh(protein_sequence, folder_name = get_default_pureseqtm_folder())
```

Arguments

protein_sequence	a protein sequence
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  expect_true(is_tmh("QEKNWSALLTAVVIILTIAGNILVIMAVSLEKKLQNATNYFLM"))
  expect_false(is_tmh("VVIILTIRGNILVIMAVSLE"))
}
```

is_topology_line *Is the line of text the topology, as used within a FASTA filename?*

Description

Is the line of text the topology, as used within a FASTA filename?

Usage

```
is_topology_line(line)
```

Arguments

line	line of text from a FASTA filename

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

expect_true(is_topology_line("000010101011"))
expect_false(is_topology_line(">5H2A_CRIGR"))
expect_false(is_topology_line("5H2A_CRIGR"))
expect_false(is_topology_line(NA))
expect_false(is_topology_line(NULL))
expect_false(is_topology_line(""))
```

plot_topology *Plot the topology*

Description

Plot the topology

Usage

```
plot_topology(topology)
```

Arguments

topology	the topology as a tibble as returned by predict_proteome_topology

Author(s)

Richèl J.C. Bilderbeek

`predict_proteome_topology` *Predict the topology of a proteome*

Description

Predict the topology of a proteome

Usage

```
predict_proteome_topology(
  fasta_filename,
  folder_name = get_default_pureseqtm_folder(),
  topology_filename = tempfile(fileext = ".top")
)
```

Arguments

`fasta_filename`	path to a FASTA file
`folder_name`	superfolder of PureseqTM. The superfolder's name is `/home/[user_name]/.local/share` by default, as can be obtained by get_default_pureseqtm_folder
`topology_filename`	name of the file to save a protein's topology to

Value

a tibble

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  topology <- predict_proteome_topology(fasta_filename)
  expect_true("name" %in% names(topology))
  expect_true("topology" %in% names(topology))
  expect_equal(1, nrow(topology))
}
```

`predict_topology_from_sequence` *Run PureseqTM directy on a protein sequence*

Description

Run PureseqTM directy on a protein sequence

Usage

```
predict_topology_from_sequence(
  protein_sequence,
  folder_name = get_default_pureseqtm_folder()
)
```

Arguments

`protein_sequence`	a protein sequence, with the amino acids as capitals, for example `MEILCEDNTSLSSIPNSL`
`folder_name`	superfolder of PureseqTM. The superfolder's name is `/home/[user_name]/.local/share` by default, as can be obtained by get_default_pureseqtm_folder

Value

a topology as a string of zeroes and ones

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  protein_sequence <- paste0(
    "QEKNWSALLTAVVIILTIAGNILVIMAVSLEKKLQNATNYFLM",
    "SLAIADMLLGFLVMPVSMLTILYGYRWP"
  )
  topology <- predict_topology_from_sequence(protein_sequence)
  expect_true(is_topology_line(topology))
}
```

pureseqtmr *pureseqtmr: estimate the topoplogy of membrane proteins*

Description

Proteins reside in either the cell plasma of in the cell membrane. A membrane protein goes through the membrane at least once. There are multiple ways to span this hydrophobic layer. One common structure is the transmembrane (alpha) helix (TMH). Given the amino acid sequence of a membrane protein, this package predicts which parts of the protein are TMHs

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  # Obtain an example filename
  fasta_filename <- get_example_filename("1bhaA.fasta")

  # Get the topology as a tibble
  topology <- predict_proteome_topology(fasta_filename)
  expect_true("name" %in% names(topology))
  expect_true("topology" %in% names(topology))
  expect_equal(1, nrow(topology))

  # show the topology
  plot_topology(topology)
}
```

run_pureseqtm *Runs PureseqTM for one gene and returns the parsed results*

Description

Runs PureseqTM for one gene and returns the parsed results

Usage

```
run_pureseqtm(
  fasta_filename,
  folder_name = get_default_pureseqtm_folder(),
  temp_folder_name = tempfile(pattern = "pureseqt_")
)
```

Arguments

`fasta_filename`	path to a FASTA file
`folder_name`	superfolder of PureseqTM. The superfolder's name is `/home/[user_name]/.local/share` by default, as can be obtained by get_default_pureseqtm_folder
`temp_folder_name`	path of a temporary folder. The folder does not need to exist. Files that are out in this folder are not automatically deleted, which is not a problem, as the default path given by tempdir is automatically cleaned by the operating system

Value

full path to the files created

Author(s)

Richèl J.C. Bilderbeek

See Also

- Use create_pureseqtm_files to only create the PureseqTM output files
- Use run_pureseqtm_proteome to run PureseqTM on multiple genes

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  topology_text <- run_pureseqtm_proteome(fasta_filename)
  proteome_text <- readLines(fasta_filename)
  expect_equal(3, length(topology_text))
  expect_equal(proteome_text[1], topology_text[1])
}
```

`run_pureseqtm_proteome` *Run PureseqTM on a proteome*

Description

Run PureseqTM on a proteome

Usage

```
run_pureseqtm_proteome(
  fasta_filename,
  folder_name = get_default_pureseqtm_folder(),
  topology_filename = tempfile(fileext = ".top")
)
```

Arguments

fasta_filename	path to a FASTA file
folder_name	superfolder of PureseqTM. The superfolder's name is /home/[user_name]/.local/share by default, as can be obtained by get_default_pureseqtm_folder
topology_filename	name of the file to save a protein's topology to

Value

the topology of the proteome, using the same output as PureseqTM. Use predict_proteome_topology to get the topology as a

Author(s)

Richèl J.C. Bilderbeek

See Also

- Use predict_proteome_topology to predict the topology of a proteome
- Use create_pureseqtm_files to only create the PureseqTM output files
- Use run_pureseqtm to run PureseqTM on one gene in more detail

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  topology <- run_pureseqtm_proteome(fasta_filename)

  expect_true(is_protein_name_line(topology[1]))

  # Second line is the protein's amino acid sequence
  expect_equal(
    topology[2],
    paste0(
      "QAQITGRPEWIWLALGTALMGLGTLYFLVKGMGVS",
      "DPDAKKFYAITTLVPAIAFTMYLSMLLGYGLTMVPF"
    )
  )
```

```
  expect_true(is_topology_line(topology[3]))
}
```

`run_pureseqtm_to_file` *Creates a FASTA-like file, that has the locations of the amino acids.*

Description

Creates a FASTA-like file, that has the locations of the amino acids.

Usage

```
run_pureseqtm_to_file(
  fasta_filename,
  pureseqtm_filename,
  folder_name = get_default_pureseqtm_folder()
)
```

Arguments

`fasta_filename`	path to a FASTA file
`pureseqtm_filename`	filename to write the PureseqTM results to
`folder_name`	superfolder of PureseqTM. The superfolder's name is `/home/[user_name]/.local/share` by default, as can be obtained by get_default_pureseqtm_folder

Author(s)

Richèl J.C. Bilderbeek

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  pureseqtm_filename <- tempfile()
  run_pureseqtm_to_file(
    fasta_filename = get_example_filename("1bhaA.fasta"),
    pureseqtm_filename = pureseqtm_filename
  )
  expect_true(file.exists(pureseqtm_filename))
}
```

`tally_tmhs` *Count the number of TMHs in a topology*

Description

Count the number of TMHs in a topology

Usage

```
tally_tmhs(topology)
```

Arguments

`topology`	the topology as a tibble as returned by predict_proteome_topology

Value

a tibble with the number of TMHs per protein

Examples

```
library(testthat)

if (is_pureseqtm_installed()) {
  topology <- predict_proteome_topology(
    get_example_filename("1bhaA.fasta")
  )
  tally <- tally_tmhs(topology)
  expect_true("name" %in% names(tally))
  expect_true("n_tmhs" %in% names(tally))
  expect_equal(nrow(topology), nrow(tally))
  expect_equal(1, nrow(tally))
  expect_equal(2, tally$n_tmhs[1])
}
```

`uninstall_pureseqtm` *Uninstall PureseqTM*

Description

Uninstall PureseqTM

Usage

```
uninstall_pureseqtm(folder_name = get_default_pureseqtm_folder())
```

Arguments

`folder_name`	name of the folder where the PureseqTM files are installed. The name of the PureseqTM binary file will be at `[folder_name]/PureseqTM_Package`

Author(s)

Richèl J.C. Bilderbeek