# Package ‘pureseqtmr’

October 14, 2022

**Title** Predict Transmembrane Protein Topology

**Version** 1.2

**Description** Proteins reside in either the cell plasma or in the cell membrane. A membrane protein goes through the membrane at least once. Given the amino acid sequence of a membrane protein, the tool 'PureseqTM' (<https://github.com/PureseqTM/pureseqTM_package>), as described in ``Efficient And Accurate Prediction Of Transmembrane Topology From Amino acid sequence only.'', Wang, Qing, et al (2019), <doi:10.1101/627307>), can predict the topology of a membrane protein. This package allows one to use 'PureseqTM' from R.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** ggplot2, plyr, rappdirs, stringr, tibble

**Suggests** testthat, knitr, rmarkdown

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

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

**VignetteBuilder** knitr

**SystemRequirements** PureseqTM (https://github.com/PureseqTM/pureseqTM_package)

**NeedsCompilation** no

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

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

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2020-07-30 10:50:02 UTC

# R topics documented:

---

## `are_tmhs` *Are the sequences transmembrance helices?*

### Description

Are the sequences transmembrance helices?

### 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 |

### Value

a vector of booleans of the same length as the number of sequences. The ith element is TRUE if the ith protein sequence is a transmembrane helix

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
if (is_pureseqtm_installed()) {
  sequences <- c(
    "QEKNWSALLTAVVIILTIAGNILVIMAVSLEKKLQNATNYFLM",
    "VVIILTIRGNILVIMAVSLE"
  )
  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. Will stop with a helpful error message if PureseqTM is not installed.

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
if (is_pureseqtm_installed()) {
  check_pureseqtm_installation()
}
```

---

## `check_topology` *Check if the topology is valid.*

---

### Description

Check if the argument is of the same type as a predicted topology, as can be created with predict_topology. Will stop if not.

### Usage

```
check_topology(topology)
```

### Arguments

| | |
|---|---|
| `topology` | the topology as a tibble as returned by predict_topology |

### Value

Nothing. Will stop with a helpful error message if the topology is invalid.

### Author(s)

Richèl J.C. Bilderbeek

### Examples

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

### Examples

```
if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  create_pureseqtm_files(fasta_filename)
}
```

---

## `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")
  create_pureseqtm_proteome_file(fasta_filename)
}
```

---

# `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_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

```
get_default_pureseqtm_folder()
```

---

## `get_example_filename` *Get the full path to a PureseqTM example file.*

### Description

Get the full path to a PureseqTM example file. If the filename specified is not a PureseqTM example file, this function will stop

### 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 |

### Value

the full path to a PureseqTM example file

### Author(s)

Richèl J.C. Bilderbeek

### See Also

use get_example_filenames to get all PureseqTM example filenames

### Examples

```
if (is_pureseqtm_installed()) {
  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

```
if (is_pureseqtm_installed()) {
  get_example_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()
```

## Value

a URL as a character vector of one element

## Author(s)

Richèl J.C. Bilderbeek

## Examples

```
get_pureseqtm_url()
```

---

# `get_pureseqtm_version` *Get the PureseqTM version*

## Description

Get the PureseqTM version

## Usage

```
get_pureseqtm_version(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 version number as a character vector of one element, for example `v0.10`

## Author(s)

Richèl J.C. Bilderbeek

Richèl J.C. Bilderbeek

## Examples

```
if (is_pureseqtm_installed()) {
  get_pureseqtm_version()
}
```

---

## `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 |

### Value

Nothing.

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
## Not run:
  install_pureseqtm()

## End(Not run)
```

---

## `is_on_appveyor` *Determines if the environment is AppVeyor*

---

### Description

Determines if the environment is AppVeyor

### Usage

```
is_on_appveyor()
```

## Value

TRUE if run on AppVeyor, FALSE otherwise

## Author(s)

Richèl J.C. Bilderbeek

## Examples

```
if (is_on_appveyor()) {
  print("Running on AppVeyor")
}
```

---

# `is_on_ci` *Determines if the environment is a continuous integration service*

## Description

Determines if the environment is a continuous integration service

## Usage

```
is_on_ci()
```

## Value

TRUE if run on AppVeyor or Travis CI, FALSE otherwise

## Author(s)

Richèl J.C. Bilderbeek

## Examples

```
if (is_on_ci()) {
  print("Running on a continuous integration service")
}
```

---

## `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

### Examples

```
if (is_on_travis()) {
  print("Running on Travis CI")
}
```

---

## `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 |

### Value

TRUE if the line can be the name of a protein in a FASTA file

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
is_protein_name_line(">5H2A_CRIGR")
```

---

`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

```
is_pureseqtm_installed()
```

---

## `is_tmh` *Determine if the protein sequence contains at least one transmembrane helix.*

---

### Description

Determine if the protein sequence contains at least one transmembrane helix.

### 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 |

### Value

TRUE if the protein sequence contains at least one transmembrane helix

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
if (is_pureseqtm_installed()) {
  # This sequence is a TMH
  is_tmh("QEKNWSALLTAVVIILTIAGNILVIMAVSLEKKLQNATNYFLM")

  # This sequence is not a TMH
  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? In this context, a topology is a string of zeroes and ones, in which a one denotes that that amino acid is within the membrane.

### Usage

```
is_topology_line(line)
```

### Arguments

| | |
|---|---|
| `line` | line of text from a FASTA filename |

### Value

TRUE if the line can be the text of a topology in a FASTA file.

### Author(s)

Richèl J.C. Bilderbeek

### Examples

```
# This is a valid topology
is_topology_line("000010101011")

# This is an invalid topology
is_topology_line("invalid")
```

---

## `plot_topology` *Plot the topology*

### Description

Plot the topology

### Usage

```
plot_topology(topology)
```

### Arguments

| | |
|---|---|
| `topology` | the topology as a tibble as returned by predict_topology |

### Value

a ggplot that displays the topology of one or more proteins

### Author(s)

Richèl J.C. Bilderbeek

## Examples

```
if (is_pureseqtm_installed() && is_on_ci()) {
  fasta_filename <- get_example_filename("test_proteome.fasta")
  topology <- predict_topology(fasta_filename)
  plot_topology(topology)
}
```

---

# `predict_topology` *Predict the topology of a proteome*

## Description

Predict the topology of a proteome

## Usage

```
predict_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 with the columns 'name' and 'topology', where the 'name' column hold all the proteins' names, and 'topology' contains all respective topologies.

## Author(s)

Richèl J.C. Bilderbeek

## Examples

```
if (is_pureseqtm_installed()) {
  fasta_filename <- get_example_filename("1bhaA.fasta")
  predict_topology(fasta_filename)
}
```

---

## `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, where a one denotes that the corresponding amino acid is located within the membrane.

### Author(s)

Richèl J.C. Bilderbeek

### Examples

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

## `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

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

  # Get the topology as a tibble
  topology <- predict_topology(fasta_filename)

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

## `pureseqtmr_report` *Create a pureseqtmr report, to be used when reporting bugs*

### Description

Create a pureseqtmr report, to be used when reporting bugs

### Usage

```
pureseqtmr_report(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

```
pureseqtmr_report()
```

---

`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_topology to get the topology as a tibble

### Author(s)

Richèl J.C. Bilderbeek

### See Also

- Use predict_topology to predict the topology of a proteome
- Use create_pureseqtm_files to only create the PureseqTM output files

### Examples

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

---

## `tally_tmhs` *Count the number of transmembrane helices in a topology*

---

### Description

Count the number of transmembrane helices in a topology

### Usage

```
tally_tmhs(topology)
```

### Arguments

| | |
|---|---|
| `topology` | the topology as a tibble as returned by predict_topology |

### Value

a tibble with the number of TMHs per protein

### Examples

```
if (is_pureseqtm_installed()) {
  tally_tmhs(
    predict_topology(
      get_example_filename("1bhaA.fasta")
    )
  )
}
```

---

## `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` |

### Value

Nothing.

### Author(s)

Richèl J.C. Bilderbeek

# Index