Package 'PepPrep'

February 19, 2015

Type Package

Classification/ACM-2012 Computational genomics, Computational transcriptomics

Title Insilico peptide mutation, digestion and homologous comparison.

Version 1.1.0

Date 2014-09-10

Author Rafael Dellen

Maintainer Rafael Dellen <Rafael.Dellen@uni-duesseldorf.de>

Description Amino acid exchange based on single nucleotide variant (SNV) information and tryptic digestion on peptide sequence. Searching for homologous by comparison of tryptic digested peptide sequences.

License GPL-3

Depends biomaRt, stringr

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-12 10:12:09

R topics documented:

`compHomToPepFasta` *Comparison of proteins and creating homologous peptides workflow*

Description

This is a wrapper for searching pairs of protein sequences by UniProt EntryName, digesting both sequences with trypsin, find homologous parts, remove duplicates, build a new sequence out of them and write the result into a FASTA file, that can be used for further analysis (e.g. compare to mass spectrometry results).

Usage

```
compHomToPepFasta(path_o1, path_o2, path, width = 60,
intermediate = FALSE, target = "K|R", exception = "P")
```

Arguments

`path_o1`	Character string indicating the path to a uniprot proteom FASTA database, for the first organism.
`path_o2`	Character string indicating the path to a uniprot proteom FASTA database, for the second organism.
`path`	Character string indicating the path where to write the resulting FASTA file.
`width`	Width of the sequence in the result (default 60).
`intermediate`	Logical, TRUE if you would like to have intermediate output, FALSE if not (default).
`target`	Character string, pattern to be matched before the cleavage site (default "K|R").
`exception`	Character string, pattern that avoids a cleavage when it can be found behind it. (default"P").

Details

Searching pairs of protein sequences by UniProt EntryName in both organisms:
Org1: Human
>sp|P31946|**1433B_HUMAN** 14-3-3 protein beta/alpha OS=Homo sapiens GN=YWHAB PE=1 SV=3
Org2: Mouse
>sp|Q9CQV8-2|**1433B_MOUSE** Isoform Short of 14-3-3 protein beta/alpha OS=Mus musculus GN=Ywhab
>sp|Q9CQV8|**1433B_MOUSE** 14-3-3 protein beta/alpha OS=Mus musculus GN=Ywhab PE=1 SV=3
Pairs:
P31946|1433B_HUMAN Q9CQV8-2|1433B_MOUSE
P31946|1433B_HUMAN Q9CQV8|1433B_MOUSE

Digesting both sequences with trypsin:
Org1: >sp|P31946|1433B_HUMAN ...
MTMD**K**SELVQ**K**A**K**LAEQAE**R**YDDMAAAM**K**...
Org2: >sp|Q9CQV8-2|1433B_MOUSE ...
MD**K**SELVQ**K**A**K**LAEQAE**R**YDDMAAAM**K**...

Find homologous parts, remove duplicates, build a new sequence out of them:
Homolog Org1Org2: >sp|P31946|1433B_HUMAN ... org2:sp|Q9CQV8-2|1433B_MOUSE ...
SELVQ**K**A**K**LAEQAE**R**YDDMAAAM**K**...

Write the result into a FASTA file, that can be used for further analysis (e.g. compare to mass spectrometry results).

You can use target and exception to set other rules for digestion.
The patterns for target and exception are restricted to one aminoacid.
Aminoacids: ARNDCQEGHILKMFPSTWYV
valid patterns: A|R|W|H, P|S
invalid patterns: Z|F|A|D, AR|NDC|STW

UniProt, the source of the proteoms:
http://www.uniprot.org/

Value

If you set intermediate to TRUE you will get the following output:

`tbl`	A data.frame that contains the proteinpairs, the header and the homologous sequence.
`fasta`	Character vector of the resulting FASTA file.

Otherwise just a character vector where to find the FASTA file or an error message.

Note

The intermediate output will be big (in most cases), use a variable to save the result.

Author(s)

Rafael Dellen
<Rafael.Dellen@uni-duesseldorf.de>

Examples

```
#load data and set arguments

#Uniprot proteom FASTA databases
#(just a small example with two proteins each)
path_o1 <- system.file("extdata", "ExampleHumanProt.fasta", package="PepPrep")
```

```
path_o2 <- system.file("extdata", "ExampleMouseProt.fasta", package="PepPrep")

#where to write the result and how to formate
path <- paste0(getwd(), "/myTest_compHomToPep.fasta")
width <- 60

#call workflow
test <- compHomToPepFasta(path_o1, path_o2, path, width)
test <- compHomToPepFasta(path_o1, path_o2, path, width, intermediate=TRUE)
```

ExampleDataSets *Summary of example data sets.*

Description

Explaining example data of PepPrep package.

Details

'ExampleData.RData' contains a data.frame called testtbl (annotated single nucleotide variant (SNV)).
Columns in testtbl:
Chr: Chromosom number
Start: Startposition
End: Endposition
Ref: Nucleotide in referencegenome
Obs: Observed Nucleotide
AAChange: Information from ANNOVAR it should look like NM_ID:c.Ref_Pos_Obs:p.RefAminoacid_Pos_MutAminoacid
Gene: Gene name

'ExampleHomo_sapiens.GRCh37.70.pep.all.fa'
Contains the Ensembl protein ENSP00000361883 in FASTA format

'ExampleHumanProt.fasta'
Two human protein sequneces from UniProt.

'ExampleMouseProt.fasta'
Two mouse protein sequences from UniProt.

Examples

```
testtbl <- system.file("extdata", "ExampleData.RData", package="PepPrep")
load(testtbl)
```

```
testtbl
attributes(testtbl)
lapply(testtbl,class)
```

---

snvToPepFasta *Single nucleotide variant (SNV) to peptide workflow*

Description

This is a wrapper for the whole computing of SNV mutations into transcripts, digest these transcripts into small peptides and write the result into a FASTA file, that can be used for further analysis (e.g. compare to mass spectrometry results).

Usage

```
snvToPepFasta(tbl, glst, mymart, myarchive, spath, tpath, width = 60,
intermediate = FALSE, target = "K|R", exception = "P")
```

Arguments

tbl	Data.frame of ANNOVAR annotated SNVs.
glst	Data.frame of gennames, column Genes.
mymart	Mart to retrieve the ENST from via biomaRt.
myarchive	Logical that indicates if a archive mart is given, (default FALSE).
spath	Character string giving the path to HUMAN Ensemble peptide database in FASTA.
tpath	Character string giving the path where to write the mutated and digested sequences in FASTA format.
width	Width of the sequence in the result (default 60).
intermediate	Logical, TRUE if you would like to have intermediate output, FALSE if not (default).
target	Character string, pattern to be matched before the cleavage site (default "K|R").
exception	Character string, pattern that avoids a cleavage when it can be found behind it. (default"P").

Details

The Refseq mRNA ID NM_ID will be used by biomaRt to querry the Ensemble transcript ID (ENST).
http://www.ncbi.nlm.nih.gov/refseq/

The **header** of the FASTA file will look like this:
>ENST|description| originalAminoacid->mutatedAminoacid_positionAminoacid ...

If the **annotated change does not fit** to the ENST it will look like:
wrong: originalAminoacid->mutatedAminoacid_positionAminoacid

If the ENST matches **two or more NM_IDs**, there will be a counter in the header:
>ENSTxcounter|...

Trypsination rule: cut after K and R except when followed by P

You can use target and exception to set other rules for digestion.
The patterns for target and exception are restricted to one aminoacid.
Aminoacids: ARNDCQEGHILKMFPSTWYV
valid patterns: A|R|W|H, P|S
invalid patterns: Z|F|A|D, AR|NDC|STW

The analysis is based on Ensembl proteindata:
http://www.ensembl.org/index.html
The SNVs annotation has to look like ANNOVAR:
http://www.openbioinformatics.org/annovar/

Value

If you set intermediate to TRUE you will get the following output:

`aachanges`	A data.frame like tbl, with new columns that describe the aminoacid changes.
`transcripts`	Data.frame, containing: ensemble_transcript_id, nmid and pname.
`mutfasta`	Character vector that contains FASTA headers and peptide sequences.
`mutlog`	Character vector contains log entries of errors reported during mutation (mutateProtToPep()).

Otherwise just a character vector where to find the FASTA file or an error message.

Note

The intermediate output will be big (in most cases), use a variable to save the result.

Author(s)

Rafael Dellen
<Rafael.Dellen@uni-duesseldorf.de>

Examples

```
#load data and set arguments

#data.frame with SNVs
```

```
tbl <- system.file("extdata", "ExampleData.RData", package="PepPrep")
load(tbl)

glst <- data.frame(Genes="CAP1", stringsAsFactors=FALSE)

#peptide sequence
spath <- system.file("extdata", "ExampleHomo_sapiens.GRCh37.70.pep.all.fa", package="PepPrep")

#where to write the result and how to write
tpath <- paste0(getwd(), "/myTest_snvToPep.fasta")
width <- 60

#biomaRt settings
mymart <- "ensembl"
myarchive <- FALSE

#call workflow
## Not run:
test <- snvToPepFasta(testtbl, glst, mymart, myarchive, spath, tpath,width)
test2 <- snvToPepFasta(testtbl, glst, mymart, myarchive, spath, tpath, width, intermediat= TRUE)
## End(Not run)
```

Index