Package ‘cleaver’

April 25, 2017

Version 1.14.0

Date 2017-02-14

Title Cleavage of Polypeptide Sequences

Maintainer Sebastian Gibb <mail@sebastiangibb.de>

Depends R (>= 3.0.0), methods, Biostrings (>= 1.29.8)

Imports S4Vectors, IRanges

Suggests testthat (>= 0.8), knitr, BiocStyle (>= 0.0.14), BRAIN, UniProt.ws (>= 2.1.4)

Description In-silico cleavage of polypeptide sequences. The cleavage rules are taken from:

License GPL (>= 3)

URL https://github.com/sgibb/cleaver/

BugReports https://github.com/sgibb/cleaver/issues/

LazyLoad yes

VignetteBuilder knitr

biocViews Proteomics

NeedsCompilation no

Author Sebastian Gibb [aut, cre]

R topics documented:

 cleaver-package .......................................................... 2
cleave ................................................................. 2

Index 9
cleaver-package

Cleavage of polypeptide sequences

Description

This package cleaves polypeptide sequences. It provides three functions: `cleave`, `cleavageRanges` and `cleavageSites`.

Details

The cleavage rules are taken from: http://web.expasy.org/peptide_cutter/peptidcutter_enzymes.html

<table>
<thead>
<tr>
<th>Package</th>
<th>cleaver</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>GPL (&gt;= 3)</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://github.com/sgibb/cleaver/">https://github.com/sgibb/cleaver/</a></td>
</tr>
</tbody>
</table>

Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

References

https://github.com/sgibb/cleaver/


See Also

cleave, cleavageRanges and cleavageSites.

cleave

Cleavage of polypeptide sequences

Description

This functions cleave polypeptide sequences. Use `cleavageSites` to find the cleavage sites, `cleavageRanges` to find the cleavage ranges and `cleave` to get the cleavage products.

Usage

```r
## S4 method for signature 'character'
cleave(x, enzym = "trypsin", missedCleavages = 0,
custom = NULL, unique = TRUE)

## S4 method for signature 'AAString'
```
cleft(x, enzym = "trypsin", missedCleavages = 0,
        custom = NULL, unique = TRUE)

## S4 method for signature 'AAStringSet'
cleave(x, enzym = "trypsin", missedCleavages = 0,
        custom = NULL)

## S4 method for signature 'character'
cleaveRanges(x, enzym = "trypsin", missedCleavages = 0,
               custom = NULL)

## S4 method for signature 'AAString'
cleaveRanges(x, enzym = "trypsin", missedCleavages = 0,
               custom = NULL)

## S4 method for signature 'AAStringSet'
cleaveRanges(x, enzym = "trypsin", missedCleavages = 0,
               custom = NULL)

## S4 method for signature 'character'
cleaveSites(x, enzym = "trypsin", custom = NULL)

## S4 method for signature 'AAString'
cleaveSites(x, enzym = "trypsin", custom = NULL)

## S4 method for signature 'AAStringSet'
cleaveSites(x, enzym = "trypsin", custom = NULL)

Arguments

- `x` polypeptide sequences.
- `enzym` character, cleavage rule.
- `missedCleavages` numeric, number of missed cleavages.
- `custom` character, of length 1 or 2. Could be used to define own cleavage rules. The first element would be the pattern and the optional second element would be an exception (non-cleavage) pattern. Perl-like regular expressions are supported, see `gregexpr` for details. If `custom` is set the `enzym` is ignored.
- `unique` logical, if TRUE all duplicated cleavage products per peptide are removed.

Details


Cleavage rules (cleavage between P1 and P1'):

<table>
<thead>
<tr>
<th>Rule name</th>
<th>P4</th>
<th>P3</th>
<th>P2</th>
<th>P1</th>
<th>P1'</th>
</tr>
</thead>
<tbody>
<tr>
<td>arg-c proteinase</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>-</td>
</tr>
<tr>
<td>asp-n endopeptidase</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>D</td>
</tr>
<tr>
<td>bnp-skatole-c</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>-</td>
</tr>
<tr>
<td>caspase1</td>
<td>F,W,Y,L</td>
<td>-</td>
<td>H,A,T</td>
<td>D</td>
<td>not P,E,D,Q,</td>
</tr>
<tr>
<td>caspase2</td>
<td>D</td>
<td>V</td>
<td>A</td>
<td>D</td>
<td>not P,E,D,Q,</td>
</tr>
</tbody>
</table>
caspase3  D  M  Q  D  not P,E,D,Q,R
    caspase4  L  E  V  D  not P,E,D,Q,R
    caspase5  L, W  E  H  D  -
    caspase6  V  E  H, I  D  not P,E,D,Q,R
    caspase7  D  E  V  D  not P,E,D,Q,R
    caspase8  L  E  T  D  not P,E,D,Q,R
    caspase9  L  E  H  D  -
    caspase10 I  E  A  D  -
    chymotrypsin-high - - - F, Y  not P
    chymotrypsin-low - - - W  not M, P
    clostripain - - - R  -
    cnbr - - - M  -
    factor xa  A, F, G, I, L, T, V, M  D, E  K  -
    formic acid - - - D  -
    glutamyl endopeptidase - - - D  -
    granzyme-b  I  E  P  D  -
    hydroxylamine - - - N  G
    iodosobenzoic acid - - - W  -
    lysozyme - - - K  -
    lysin - - - K  -
    neutrophil elastase - - - A, V  -
    ntcb - - - C  -
    pepsin1.3 - - not H, K, R  not P  not R  F, L, W, Y
    pepsin - - not H, K, R  not P  F, L, W, Y  -
    proline endopeptidase - - not H, K, R  not P  F, L  -
    proteinase k - - - A, E, F, I, L, T, V, W, Y  -
    staphylococcal peptidase i - - - E  -
    thermolysin - - - not D, E  A, F, I, L, M, V, W  -
    trypsin - - - K, R  P  -
    - - - M  R  P

Exceptions:

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Enzyme name</th>
</tr>
</thead>
<tbody>
<tr>
<td>trypsin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P4  P3  P2  P1  P1'  P2'</td>
</tr>
<tr>
<td>trypsin</td>
<td>C, D  K  D  -</td>
</tr>
<tr>
<td></td>
<td>-  C  K  H, Y  -</td>
</tr>
<tr>
<td></td>
<td>-  C  R  K  -</td>
</tr>
<tr>
<td></td>
<td>-  R  R  H, R  -</td>
</tr>
</tbody>
</table>
arg-c proteinase  Arg-C proteinase
asp-n endopeptidase  Asp-N endopeptidase
bnps-skatole-c  BNPS-Skatole
caspase1  Caspase 1
caspase2  Caspase 2
caspase3  Caspase 3
caspase4  Caspase 4
caspase5  Caspase 5
caspase6  Caspase 6
caspase7  Caspase 7
caspase8  Caspase 8
caspase9  Caspase 9
caspase10  Caspase 10
chymotrypsin-high  Chymotrypsin-high specificity (C-term to [FYW], not before P)
chymotrypsin-low  Chymotrypsin-low specificity (C-term to [FYWML], not before P)
clostripain  Clostripain (Clostridiopeptidase B)
cnbr  CNBr
enterokinase  Enterokinase
factor xa  Factor Xa
formic acid  Formic acid
glutamyl endopeptidase  Glutamyl endopeptidase
granzyme-b  Granzyme B
hydroxylamine  Hydroxylamine
iodosobenzoic acid  Iodosobenzoic acid
lysc  LysC
lysn  LysN
neutrophil elastase  Neutrophil elastase
ntcb  NTCB (2-nitro-5-thiocyanobenzoic acid)
pepsin1.3  Pepsin (pH == 1.3)
pepsin  Pepsin (pH > 2)
proline endopeptidase  Proline-endopeptidase
proteinase k  Proteinase K
staphylococcal peptidase i  Staphylococcal Peptidase I
thermolysin  Thermolysin
thrombin  Thrombin
trypsin  Trypsin

Value

cleave  If x is a character it returns a list of the same length as x. Each element contains a character vector with the corresponding cleavage products of the polypeptides. If x is an AAString or an AAStringSet an AAStringSet or an AAStringSetList instance of the same length as x is returned. Each element contains an AAString or an AAStringSet instance with the corresponding cleavage products of the polypeptides.
cleavageRanges  If x is a character it returns a list of the same length as x. Each element contains a two-column matrix with the start and end positions of the peptides. If x is an AAString or an AAStringSet instance an IRanges or an IRangesList of the same length as x is returned.
cleavageSites  Returns a list of the same length as x. Each element contains an integer vector with the cleavage positions.

Overview:
Author(s)
Sebastian Gibb <mail@sebastiangibb.de>

References

See Also
AAString, AAStringSet, AAStringSetList, IRanges, IRangesList

Examples
library("cleaver")

## Gastric juice peptide 1 (UniProtKB/Swiss-Prot: GAJU_HUMAN/P01358)
gaju <- "LAAGKVEDSD"
cleave(gaju, "trypsin")
# $LAAGKVEDSD
# [1] "LAAGK" "VEDSD"
cleavageRanges(gaju, "trypsin")
# $LAAGKVEDSD
# start end
# [1,] 1 5
# [2,] 6 10
cleavageSites(gaju, "trypsin")
# $LAAGKVEDSD
# [1] 5
cleave(gaju, "trypsin", missedCleavages=1)
# $LAAGKVEDSD
# [1] "LAAGKVEDSD"
cleavageRanges(gaju, "trypsin", missedCleavages=1)
# $LAAGKVEDSD
# start end
# [1,] 1 10
cleave(gaju, "trypsin", missedCleavages=0:1)
# $LAAGKVEDSD
# [1] "LAAGK" "VEDSD" "LAAGKVEDSD"
cleavageRanges(gaju, "trypsin", missedCleavages=0:1)
# $LAAGKVEDSD
# start end
# [1,]  1  5
# [2,]  6 10
# [3,]  1 10

cleave(gaju, "pepsin")
# $LAAGKVEDSD
# [1]  "LAAGKVEDSD"
# (no cleavage)

## use AAStringSet
gaju <- AAStringSet("LAAGKVEDSD")
cleave(gaju)
# AAStringSetList of length 1
# [["LAAGKVEDSD"]]
# LAAGK VEDSD

## Beta-enolase (UniProtKB/Swiss-Prot: ENOB_THUAL/P86978)
enob <- "SITKIKAREILD"
cleave(enob, "trypsin")
# $SITKIKAREILD
# [1]  "SITK"  "IK"  "AR"  "EILD"

cleave(enob, "trypsin", missedCleavages=2)
# $SITKIKAREILD
# [1]  "SITKIKAR"  "IKAREILD"

cleave(enob, "trypsin", missedCleavages=0:2)
# $SITKIKAREILD
# [1]  "SITK"  "IK"  "AR"  "EILD" "SITKIK"  "IKAR"  "AREILD"  "SITKIKAR"  "IKAREILD"

## define own cleavage rule: cleave at K
cleave(enob, custom="K")
# $SITKIKAREILD
# [1]  "SITK"  "IK"  "AREILD"

cleavageRanges(enob, custom="K")
# $SITKIKAREILD
# start end
# [1,]  1  4
# [2,]  5  6
# [3,]  7 12

## define own cleavage rule: cleave at K but not if followed by A
cleave(enob, custom=c("K", "(K?=A)"))
# $SITKIKAREILD
# [1]  "SITK"  "IKAREILD"

cleavageRanges(enob, custom=c("K", "(K?=A)"))
# $SITKIKAREILD
# start end
cleavageSites(enob, custom=c("K", "K(?=A)"))

# $SITKIKAREILD
# [1] 4
Index

*Topic **methods**
  cleave, 2
*Topic **package**
  cleaver-package, 2

AAString, 5, 6
AAStringSet, 5, 6
AAStringSetList, 5, 6

cleavageRanges, 2
cleavageRanges(cleave), 2
cleavageRanges, AAString-method (cleave), 2
cleavageRanges, AAStringSet-method (cleave), 2
cleavageRanges, character-method (cleave), 2
cleavageSites, 2
cleavageSites(cleave), 2
cleavageSites, AAString-method(cleave), 2
cleavageSites, AAStringSet-method (cleave), 2
cleavageSites, character-method (cleave), 2
cleave, 2, 2
cleave, AAString-method(cleave), 2
cleave, AAStringSet-method(cleave), 2
cleave, character-method(cleave), 2
cleave-package, 2

gregexpr, 3

IRanges, 5, 6
IRangesList, 5, 6