Package ‘cleaver’

April 8, 2025

Version 1.45.0

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), rmarkdown, BRAIN, UniProt.ws (>= 2.36.5)

Description In-silico cleavage of polypeptide sequences. The cleavage rules are taken from: http://web.expasy.org/peptide_cutter/peptidecutter_enzymes.html

License GPL (>= 3)

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

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

LazyLoad yes

VignetteBuilder knitr

biocViews Proteomics

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

git_branch devel

git_last_commit 739ea76

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2025-04-08

Author Sebastian Gibb [aut, cre] (ORCID: <`https://orcid.org/0000-0001-7406-4443`>)

Contents

cleaver-package 2
cleave-methods 2

Index **10**

`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: `https://web.expasy.org/peptide_cutter/peptidecutter_enzymes.html`

Package:	cleaver
License:	GPL (>= 3)
URL:	https://github.com/sgibb/cleaver/

Author(s)

Sebastian Gibb `<mail@sebastiangibb.de>`

References

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

Gasteiger E., Hoogland C., Gattiker A., Duvaud S., Wilkins M.R., Appel R.D., Bairoch A.; "Protein Identification and Analysis Tools on the ExPASy Server". (In) John M. Walker (ed): The Proteomics Protocols Handbook, Humana Press (2005).
`https://web.expasy.org/peptide_cutter/peptidecutter_enzymes.html`

See Also

`cleave`, `cleavageRanges` and `cleavageSites`.

`cleave-methods` *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

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

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

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

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

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

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

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

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

## S4 method for signature 'AAStringSet'
cleavageSites(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 cleaveage 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

The cleavage rules are taken from: https://web.expasy.org/peptide_cutter/peptidecutter_enzymes.html

Cleavage rules (cleavage between P1 and P1'):

Rule name	P4	P3	P2	P1	P1'
`arg-c proteinase`	-	-	-	R	-
`asp-n endopeptidase`	-	-	-	-	D
`bnps-skatole-c`	-	-	-	W	-
`caspase1`	F,W,Y,L	-	H,A,T	D	not P,E,D,Q,K,R
`caspase2`	D	V	A	D	not P,E,D,Q,K,R
`caspase3`	D	M	Q	D	not P,E,D,Q,K,R
`caspase4`	L	E	V	D	not P,E,D,Q,K,R
`caspase5`	L,W	E	H	D	-
`caspase6`	V	E	H,I	D	not P,E,D,Q,K,R
`caspase7`	D	E	V	D	not P,E,D,Q,K,R
`caspase8`	I,L	E	T	D	not P,E,D,Q,K,R
`caspase9`	L	E	H	D	-
`caspase10`	I	E	A	D	-
`chymotrypsin-high`	-	-	-	F,Y	not P
	-	-	-	W	not M,P
`chymotrypsin-low`	-	-	-	F,L,Y	not P
	-	-	-	W	not M,P
	-	-	-	M	not P,Y
	-	-	-	H	not D,M,P,W
`clostripain`	-	-	-	R	-
`cnbr`	-	-	-	M	-
`enterokinase`	D,E	D,E	D,E	K	-
`factor xa`	A,F,G,I,L,T,V,M	D,E	G	R	-
`formic acid`	-	-	-	D	-
`glutamyl endopeptidase`	-	-	-	E	-
`granzyme-b`	I	E	P	D	-
`hydroxylamine`	-	-	-	N	G
`iodosobenzoic acid`	-	-	-	W	-
`lysc`	-	-	-	K	-
`lysn`	-	-	-	-	K
`lysarginase`	-	-	-	-	K,R
`neutrophil elastase`	-	-	-	A,V	-
`ntcb`	-	-	-	-	C
`pepsin1.3`	-	not H,K,R	not P	not R	F,L
`pepsin`	-	not H,K,R	not P	not R	F,L,W,Y
	-	not H,K,R	not P	F,L,W,Y	-
	-	not H,K,R	not P	F,L	-
`proline endopeptidase`	-	-	not H,K,R	P	not P
`proteinase k`	-	-	-	A,E,F,I,L,T,V,W,Y	-
`staphylococcal peptidase i`	-	-	not E	E	-
`thermolysin`	-	-	-	not D,E	A,F,I,L,M,V
`thrombin`	-	-	G	R	G

	A,F,G,I,L,T,V,M	A,F,G,I,L,T,V,W	P	R	not D,E
`trypsin`	-	-	-	K,R	not P
	-	-	W	K	P
	-	-	M	R	P
`trypsin-high`	-	-	-	K,R	not P
	-	-	W	K	P
	-	-	M	R	P
`trypsin-low`	-	-	-	K,R	not P
	-	-	W	K	P
	-	-	M	R	P
`trypsin-simple`	-	-	-	K,R	-

Exceptions:

Rule name	Enzyme name	P4	P3	P2	P1	P1’	P2’
trypsin	-	-	C,D	K	D	-	
		-	-	C	K	H,Y	-
		-	-	C	R	K	-
		-	-	R	R	H,R	-
trypsin-high	-	-	C,D	K	D	-	
		-	-	C	K	H,Y	-
		-	-	C	R	K	-
		-	-	R	R	H,R	-

Rule name	Enzyme name
`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
`lysarginase`	LysargiNase
`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
`trypsin-high`	Trypsin, higher specificity as defined in PeptideMass, identical to *trypsin*
`trypsin-low`	Trypsin, C-term to K/R if C-term is not P, as defined in PeptideMass
`trypsin-simple`	Trypsin, C-term to K/R, even before P, as defined in PeptideMass

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:

Input	`cleave`	`cleavageRanges`	`cleavageSites`
`character`	`list of character`	`list of matrix`	`list of integer`
`AAString`	`AAStringSet`	`IRanges`	`list of integer`
`AAStringSet`	`AAStringSetList`	`IRangesList`	`list of integer`

Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

References

Gasteiger E., Hoogland C., Gattiker A., Duvaud S., Wilkins M.R., Appel R.D., Bairoch A.; "Protein Identification and Analysis Tools on the ExPASy Server". (In) John M. Walker (ed): The Proteomics Protocols Handbook, Humana Press (2005).
`https://web.expasy.org/peptide_cutter/peptidecutter_enzymes.html`

PeptideMass `https://web.expasy.org/peptide_mass/peptide-mass-doc.html#table1`

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"
# [7] "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
# [1,]     1   4
```

```
# [2,]     5  12

cleavageSites(enob, custom=c("K", "K(?=A)"))
# $SITKIKAREILD
# [1] 4
```

Index

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

AAString, *6*, *7*
AAStringSet, *6*, *7*
AAStringSetList, *6*, *7*

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

gregexpr, *3*

IRanges, *6*, *7*
IRangesList, *6*, *7*