Package ‘cleaver’

November 21, 2016

Version 1.12.0
Date 2015-02-02
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**


<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/](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)
```

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

## S4 method for signature 'AAStringSet'
clevae(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'
 cleavageRangex(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 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>bnps-skatole-c</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>-</td>
</tr>
<tr>
<td>caspase1</td>
<td></td>
<td></td>
<td>H,A,T</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>caspase2</td>
<td></td>
<td></td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>Enzyme name</td>
<td>P4</td>
<td>P3</td>
<td>P2</td>
<td>P1</td>
<td>P1’</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>caspase3</td>
<td>D</td>
<td>M</td>
<td>Q</td>
<td>D</td>
<td>not P,E,D,Q,K,R -</td>
</tr>
<tr>
<td>caspase4</td>
<td>L</td>
<td>E</td>
<td>V</td>
<td>D</td>
<td>not P,E,D,Q,K,R -</td>
</tr>
<tr>
<td>caspase5</td>
<td>L,W</td>
<td>E</td>
<td>H</td>
<td>D</td>
<td>-</td>
</tr>
<tr>
<td>caspase6</td>
<td>V</td>
<td>E</td>
<td>H,I</td>
<td>D</td>
<td>not P,E,D,Q,K,R -</td>
</tr>
<tr>
<td>caspase7</td>
<td>D</td>
<td>E</td>
<td>V</td>
<td>D</td>
<td>not P,E,D,Q,K,R -</td>
</tr>
<tr>
<td>caspase8</td>
<td>I,L</td>
<td>E</td>
<td>T</td>
<td>D</td>
<td>not P,E,D,Q,K,R -</td>
</tr>
<tr>
<td>caspase9</td>
<td>L</td>
<td>E</td>
<td>H</td>
<td>D</td>
<td>-</td>
</tr>
<tr>
<td>caspase10</td>
<td>I</td>
<td>E</td>
<td>A</td>
<td>D</td>
<td>-</td>
</tr>
<tr>
<td>chymotrypsin-high</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>F,Y</td>
<td>not P</td>
</tr>
<tr>
<td>chymotrypsin-low</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>not M,P</td>
</tr>
<tr>
<td>chymotrypsin-low</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>F,I,Y</td>
<td>not P</td>
</tr>
<tr>
<td>chymotrypsin-low</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>not M,P</td>
</tr>
<tr>
<td>chymotrypsin-low</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>M</td>
<td>not P,Y</td>
</tr>
<tr>
<td>chymotrypsin-low</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>H</td>
<td>not D,M,P,W</td>
</tr>
<tr>
<td>clostripain</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>-</td>
</tr>
<tr>
<td>cnbr</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>M</td>
<td>-</td>
</tr>
<tr>
<td>enterokinase</td>
<td>D,E</td>
<td>D,E</td>
<td>D,E</td>
<td>K</td>
<td>-</td>
</tr>
<tr>
<td>factor xa</td>
<td>A,F,G,I,L,T,V,M</td>
<td>D,E</td>
<td>G</td>
<td>R</td>
<td>-</td>
</tr>
<tr>
<td>formic acid</td>
<td>-</td>
<td>-</td>
<td>D</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>glutamyl endopeptidase</td>
<td>-</td>
<td>-</td>
<td>D</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>granzyme-b</td>
<td>I</td>
<td>E</td>
<td>P</td>
<td>D</td>
<td>-</td>
</tr>
<tr>
<td>hydroxylamine</td>
<td>-</td>
<td>-</td>
<td>N</td>
<td>G</td>
<td>-</td>
</tr>
<tr>
<td>iodosobenzoic acid</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>lync</td>
<td>-</td>
<td>-</td>
<td>K</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>lysn</td>
<td>-</td>
<td>-</td>
<td>K</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>neutrophil elastase</td>
<td>-</td>
<td>-</td>
<td>A,V</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ntcpb</td>
<td>-</td>
<td>-</td>
<td>C</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>pepsin1.3</td>
<td>-</td>
<td>not H,K,R not P</td>
<td>not R F,I,L,W,Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pepsin</td>
<td>-</td>
<td>not H,K,R not P</td>
<td>F,I,L not P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pepsin</td>
<td>-</td>
<td>not H,K,R not P</td>
<td>F,I,L not P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>proline endopeptidase</td>
<td>-</td>
<td>not H,K,R P</td>
<td>not P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>proteinase k</td>
<td>-</td>
<td>-</td>
<td>A,E,F,I,L,T,V,W,Y</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>staphylococcal peptidase i</td>
<td>-</td>
<td>-</td>
<td>E</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>thromolysin</td>
<td>-</td>
<td>-</td>
<td>not D,E A,F,I,L,V,W,Y</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>thrombin</td>
<td>-</td>
<td>-</td>
<td>G</td>
<td>R</td>
<td>G</td>
</tr>
<tr>
<td>trypsin</td>
<td>-</td>
<td>-</td>
<td>W</td>
<td>K</td>
<td>P</td>
</tr>
<tr>
<td>trypsin</td>
<td>-</td>
<td>-</td>
<td>M</td>
<td>R</td>
<td>P</td>
</tr>
</tbody>
</table>

Exceptions:

<table>
<thead>
<tr>
<th>Rule name</th>
<th>Enzyme name</th>
<th>P4</th>
<th>P3</th>
<th>P2</th>
<th>P1</th>
<th>P1’</th>
<th>P2’</th>
</tr>
</thead>
<tbody>
<tr>
<td>trypsin</td>
<td>-</td>
<td>-</td>
<td>C,D</td>
<td>K</td>
<td>D</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>C</td>
<td>K</td>
<td>H,Y</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>C</td>
<td>R</td>
<td>K</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>R</td>
<td>H,R</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
 cleave

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 [FY], not before P)
chymotrypsin-low        Chymotrypsin-low specificity (C-term to [FYWL], not before P)
clostripain             Clostripain (Clostridiopeptidase B)
cnbr                    CNBr
enterokinase            Enteropeptidase
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                    NTNB (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:
## cleaver

<table>
<thead>
<tr>
<th>Input</th>
<th>cleave</th>
<th>cleavageRanges</th>
<th>cleavageSites</th>
</tr>
</thead>
<tbody>
<tr>
<td>character</td>
<td>list of character</td>
<td>list of matrix</td>
<td>list of integer</td>
</tr>
<tr>
<td>AAString</td>
<td>AAStringSet</td>
<td>IRanges</td>
<td>list of integer</td>
</tr>
<tr>
<td>AAStringSet</td>
<td>AAStringSetList</td>
<td>IRangesList</td>
<td>list of integer</td>
</tr>
</tbody>
</table>

### Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

### References


### See Also

AAString, AAStringSet, AAStringSetList, IRanges, IRangesList

### Examples

```r
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)
```
cleave

# $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" "SITIK" "IKAR" # [7] "AREILD" "SITIKIKAR" "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

cleave, AAString-method (cleave),
cleave, AAStringSet-method (cleave),
cleave, character-method (cleave),
cleaveer-package, 2
gregexpr, 3

IRanges, 5, 6
IRangesList, 5, 6