Package ‘PTMods’

March 6, 2026

Type Package

Version 0.99.2

Title Managing Post-Translational Modifications in R

Description An interface to the community supported database for
amino acid/protein modifications using mass
spectrometry.

Depends R (>= 4.5.0), methods

Suggests xml2, testthat, knitr, BiocStyle, Biostrings

License GPL-3

VignetteBuilder knitr

biocViews Proteomics, MassSpectrometry

BugReports `https://github.com/RforMassSpectrometry/PTMods/issues`

URL `https://github.com/RforMassSpectrometry/PTMods`

RoxygenNote 7.3.3

Encoding UTF-8

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

git_branch devel

git_last_commit 294fce2

git_last_commit_date 2026-03-05

Repository Bioconductor 3.23

Date/Publication 2026-03-05

Author Laurent Gatto [aut] (ORCID: <`https://orcid.org/0000-0002-1520-2268`>),
Sebastian Gibb [aut] (ORCID: <`https://orcid.org/0000-0001-7406-4443`>),
Guillaume Deflandre [cre] (ORCID:
<`https://orcid.org/0009-0008-1257-2416`>)

Maintainer Guillaume Deflandre <`guillaume.deflandre@uclouvain.be`>

Contents

`PTMods-package` *Managing amino acid modifications for mass spectrometry in R.*

Description

The ‘PTMods‘ package focuses on handling post-translational modifications (PTMs). It distributes PTMs from the Unimod database and allows to convert PTM annotations between multiple formats.

Author(s)

Maintainer: Guillaume Deflandre <`guillaume.deflandre@uclouvain.be`> (ORCID)

Authors:

- Laurent Gatto <`laurent.gatto@uclouvain.be`> (ORCID)
- Sebastian Gibb <`mail@sebastiangibb.de`> (ORCID)

References

`https://github.com/RforMassSpectrometry/PTMods/`

See Also

Useful links:

- `https://github.com/RforMassSpectrometry/PTMods`
- Report bugs at `https://github.com/RforMassSpectrometry/PTMods/issues`

`aminoacids` *Amminoacids data set.*

Description

‘data.frame‘ of aminoacids from the unimod database.

Usage

```
data("aminoacids")
```

Format

A ‘data.frame‘ with 11 columns (OneLetter, ThreeLetter, FullName, AvgMass, MonoMass, H, C, N, O S, Se) for the aminoacids. The H/C/N/O/S/Se columns contain the number of elements that build the aminoacid.

Details

It was created as follows:

“‘ PTMods:::.createDataSets() “‘

Source

Taken from the unimod database: http://www.unimod.org/xml/unimod.xml.

Examples

```
data(aminoacids)
head(aminoacids)
```

`convertAnnotation` *Convert sequences from one annotation style to another*

Description

Converts modifications between different annotation formats for multiple sequences at once. See the details and examples sections for more information. The annotation styles are inferred from the ‘modifications‘ dataframe (see ‘?modifications‘).

Usage

```
convertAnnotation(
  x,
  convertToStyle = c("deltaMass", "unimodId", "name"),
  massTolerance = 0.01
)
```

Arguments

`x`	Character vector with peptide sequences in ProForma format
`convertToStyle`	Character string specifying target format. Options: "deltaMass", "unimodId", "name"
`massTolerance`	Numeric mass tolerance in Daltons for matching modifications (default: 0.01). Used when converting from deltaMass.

Details

The function handles three main conversion scenarios:

- Name to deltaMass: "M[Oxidation]PEPTIDE" -> "M[+15.994915]PEPTIDE" - Name to unimodId: "M[Oxidation]PEPTIDE" -> "M[UNIMOD:35]PEPTIDE" - deltaMass to name: "M[+15.995]PEPTIDE" -> "M[Oxidation]PEPTIDE" - deltaMass to unimodId: "M[+15.995]PEPTIDE" -> "M[UNIMOD:35]PEPTIDE" - unimodId to name: "M[UNIMOD:35]PEPTIDE" -> "M[Oxidation]PEPTIDE" - unimodId to deltaMass: "M[UNIMOD:35]PEPTIDE" -> "M[+15.994915]PEPTIDE"

Value

Character vector with the sequences in the target annotation format

Author(s)

Guillaume Deflandre <guillaume.deflandre@uclouvain.be>

Examples

```
# Convert sequence from name to delta mass
convertAnnotation("M[Oxidation]PEPTIDE", convertToStyle = "deltaMass")
# Result: "M[+15.994915]PEPTIDE"

# Name to Unimod ID
convertAnnotation("M[Oxidation]PEPTIDE", convertToStyle = "unimodId")
# Result: "M[UNIMOD:35]PEPTIDE"

# Delta mass to name
convertAnnotation("M[+15.995]PEPTIDE", convertToStyle = "name")
# Result: "M[Oxidation]PEPTIDE"

# Multiple modifications
convertAnnotation("M[Oxidation]EVNES[Phospho]PEK", convertToStyle = "deltaMass")
# Result: "M[+15.994915]EVNES[+79.966331]PEK"

# Convert multiple sequences from name to delta mass
sequences <- c("M[Oxidation]PEPTIDE", "EVNES[Phospho]PEK", "PEPTIDE")
convertAnnotation(sequences, convertToStyle = "deltaMass")
# Result: c("M[+15.994915]PEPTIDE", "EVNES[+79.966331]PEK", "PEPTIDE")

# Convert from delta mass to name
sequences <- c("M[+15.995]PEPTIDE", "S[+79.966]EQUENCE")
convertAnnotation(sequences, convertToStyle = "name")
```

```
# Result: c("M[Oxidation]PEPTIDE", "S[Phospho]EQUENCE")

# Convert to Unimod IDs
sequences <- c("M[Oxidation]PEPTIDE", "C[Carbamidomethyl]PEPTIDE")
convertAnnotation(sequences, convertToStyle = "unimodId")
# Result: c("M[UNIMOD:35]PEPTIDE", "C[UNIMOD:4]PEPTIDE")
```

`elements` *Elements data set.*

Description

‘data.frame‘ of chemical elements from the unimod database.

Usage

```
data("elements")
```

Format

A ‘data.frame‘ with 4 columns (Name, FullName, AvgMass, MonoMass) for the chemical elements.

Details

It was created as follows:

“‘ PTMods:::.createDataSets() “‘

Source

Taken from the unimod database: http://www.unimod.org/xml/unimod.xml.

Examples

```
data(elements)
head(elements)
```

`modifications` *Modifications data set.*

Description

‘data.frame‘ of modifications from the unimod database.

Usage

```
data("modifications")
```

Format

A ‘data.frame‘ with 15 columns (Id (created by Name:(Position-)Site(:NeutralLoss) because unimod id is not unique), UnimodId, Name, Description, AvgMass, MonoMass, Site, Position, Classification, SpecGroup, NeutralLoss, LastModification, Approved, Hidden) for the modifications.

Details

It was created as follows:

“‘ PTMods:::.createDataSets() “‘

Source

Taken from the unimod database: http://www.unimod.org/xml/unimod.xml.

Examples

```
data(modifications)
head(modifications)
```

Index