Package: rtemisbio (via r-universe)

September 16, 2024

Title rtemis Bio-informatics

Version 0.2.1

Description Bio-informatics utilities

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports rtemis, data.table

Suggests biomaRt, jsonlite, seqinr, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://egenn.r-universe.dev

RemoteUrl https://github.com/egenn/rtemisbio

RemoteRef HEAD

RemoteSha 3ce45e531c1464437975464d536580e385e19622

Contents

`rtemisbio-package` **rtemisbio**: *Bioinformatics ops*

Description

Bioinformatics utilities

Author(s)

Maintainer: E.D. Gennatas `<gennatas@gmail.com>` (ORCID)

`aa_sub` *Perform amino acid substitutions*

Description

Perform amino acid substitutions

Usage

```
aa_sub(x, substitutions, verbosity = 1)
```

Arguments

`x`	Character or charactr vector: Amino acid sequence. e.g. `"ARND"` or `c("A", "R", "N", "D")`
`substitutions`	Character vector: Substitutions to perform in the format "OriginalPositionNew", e.g. `c("C291A", "C322A")`

Value

Character vector with substitutions performed.

Author(s)

EDG

`aggregate.xt` *Aggregate method for* `xt` *object*

Description

Aggregate method for `xt` object

Usage

```
## S3 method for class 'xt'
aggregate(
  x,
  groupname,
  fn = mean,
  backend = getOption("rt.backend", "base"),
  ...
)
```

Arguments

`x`	`xt` object.
`fn`	Function: Function to apply to each group.
`backend`	Character: "base", "data.table", or "dplyr"; backend to use for aggregation.
`...`	Additional arguments passed to `fn`.
`group`	Character: Grouping variable.

Author(s)

EDG

`as.a3` *as.a3*

Description

as.a3

Usage

```
as.a3(x)
```

Arguments

`x`	Object to convert to `a3`.

Value

`a3` object.

Author(s)

EDG

`as.a3.default` *as.a3*

Description

as.a3

Usage

```
## Default S3 method:
as.a3(x)
```

Arguments

`x`	Object to convert to `a3`.

Value

`a3` object.

Author(s)

EDG

`as.a3.list` *as.a3.list method*

Description

as.a3.list method

Usage

```
## S3 method for class 'list'
as.a3(x)
```

Arguments

`x`	List: Named list with elements `Sequence`, `Annotations`, `UniprotID`. `Annotations` is a named list with possible elements `Site`, `Region`, `PTM`, `Cleavage_site`, `Variant`, `Description`, `Reference`.

Value

`a3` object.

Author(s)

EDG

`as.xt` *as.xt*

Description

as.xt

Usage

```
as.xt(x)
```

Arguments

`x`	Object to convert to `xt`.

Value

`xt` object.

Author(s)

EDG

`as.xt.default` *as.xt*

Description

as.xt

Usage

```
## Default S3 method:
as.xt(x)
```

Arguments

`x`	Object to convert to `xt`.

Value

`xt` object.

Author(s)

EDG

`as.xt.list` *as.xt.list method*

Description

as.xt.list method

Usage

```
## S3 method for class 'list'
as.xt(x)
```

Arguments

`x`	List: Named list with elements `x`, `y`, `y2`, `xunits`, `yunits`, `y2units`, `Description`, `Reference`.

Value

`xt` object.

Author(s)

EDG

`gene2sequence` *Get the sequence of a gene*

Description

Get the sequence of a gene

Usage

```
gene2sequence(
  gene,
  organism = "hsapiens",
  biomart = "ensembl",
  host = "https://www.ensembl.org",
  seq_type = "coding",
  verbosity = 1
)
```

Arguments

`gene`	Character: Gene name.
`organism`	Character: Organism name.
`biomart`	Character: Biomart name.
`host`	Character: Host address.

Value

data.frame with columns "gene", "ensembl_transcript_id" and "sequence".

Author(s)

EDG

`int2range` *Convert integer range to character with colon separator*

Description

Convert integer range to character with colon separator

Usage

```
int2range(x)
```

Arguments

`x`	Integer vector. Must be consecutive integers from lowest to highest.

Value

Character with colon separator.

Author(s)

EDG

Examples

```
## Not run:
x <- 34:42
int2range(x)
int2range(28:34)
int2range(c(3, 4, 5, 6))
# This will throw an error:
int2range(c(3, 4, 5, 6, 8))

## End(Not run)
```

`light_dark_ratio` *Calculate light/dark ratio for* `xt` *object*

Description

Calculates light/dark ratio for each `y` and `y2` timeseries in an `xt` object.

Usage

```
light_dark_ratio(
  x,
  groupname = "Lights",
  fn = mean,
  backend = getOption("rt.backend", "data.table"),
  ...
)
```

Arguments

`x`	`xt` object.
`fn`	Function: Function to apply to each group.
`backend`	Character: "base", "data.table", or "dplyr"; backend to use for aggregation.
`...`	Additional arguments passed to `fn`.

Value

data.frame with columns for group and summary statistic.

Author(s)

EDG

`plot.a3` *Plot method for `a3` object*

Description

Plot method for `a3` object

Usage

```
## S3 method for class 'a3'
plot(x, ...)
```

Arguments

`x`	`a3` object.
`...`	Additional arguments passed to rtemis::dplot3_protein.

Author(s)

EDG

`plot.xt` *Plot method for `xt` object*

Description

Plot method for `xt` object

Usage

```
## S3 method for class 'xt'
plot(x, ...)
```

Arguments

`x`	`xt` object.
`...`	Additional arguments passed to rtemis::dplot3_xt.

Author(s)

EDG

`print.a3` *Print method for `a3` object*

Description

Print method for `a3` object

Usage

```
## S3 method for class 'a3'
print(x, head.n = 10, ...)
```

Arguments

`x`	`a3` object.
`...`	Not used.

Author(s)

EDG

`print.xt` *Print method for `xt` object*

Description

Print method for `xt` object

Usage

```
## S3 method for class 'xt'
print(x, head.n = 10, ...)
```

Arguments

`x`	`xt` object.
`...`	Not used.

Author(s)

EDG

`read.a3json` *Read `a3` object from JSON file*

Description

Read `a3` object from JSON file

Usage

```
read.a3json(filepath, verbosity = 0L)
```

Arguments

`filepath`	Character: Path to JSON file.
`verbosity`	Integer: if greater than 0, print messages.

Value

`a3` object.

Author(s)

EDG

`read.xtjson` *Read `xt` object from JSON file*

Description

Read `xt` object from JSON file

Usage

```
read.xtjson(filepath, verbosity = 0L)
```

Arguments

`filepath`	Character: Path to JSON file.
`verbosity`	Integer: if greater than 0, print messages.

Details

Note that factors saved under `group` are written as character by write.xtjson and when they are read back in, they are converted back to factors using factor. This means that the levels will be set alphabetically. If needed, reorder them after reading in the JSON file using factor.

Value

`xt` object.

Author(s)

EDG

`summary.a3` *Summary method for* `a3` *object*

Description

Summary method for `a3` object

Usage

```
## S3 method for class 'a3'
summary(object, ...)
```

Arguments

`object`	`a3` object.

Author(s)

EDG

`toa3` *Create an* `a3` *object*

Description

Creates an `a3` object given amino acid sequence and annotations.

Usage

```
toa3(
  seq,
  site = NULL,
  region = NULL,
  ptm = NULL,
  clv = NULL,
  variant = NULL,
  uniprotid = NULL,
  description = NULL,
  reference = NULL
)
```

Arguments

`seq`	Character: Amino acid sequence.
`site`	Named list of vectors of integer indices of sites, e.g. `list("N-terminal repeat" = c(46, 47, 52), "Microtubule binding domain" = c(244, 245, 246))`
`region`	Named list of integer indices, e.g. `list("Phosphodegron" = c(46, 47, 48, 49, 50, 51), "KXGS" = c(259, 260, 261, 262))` or character vectors with index range of regions in format `start:end`, e.g. `list(Phosphodegron = c("46:51", "149:154"), KXGS = c("259:262", "290:293"))`
`ptm`	Named list of vectors with indices of post-translational modifications, e.g. `list("Phosphorylation" = c(17, 18, 29, 30), "Acetylation" = c(148, 150, 163))`
`clv`	Named list of cleavage sites, e.g. `list(CTSL = c(54, 244, 319), CTSD = c(340, 391, 426))`
`variant`	List of lists with variant information. Each list must contain a `Position` element
`uniprotid`	Character: Uniprot ID.
`description`	Character: Description of the data / experiment.
`reference`	Character: Link to reference (journal publication, preprint, etc.)

Details

We choose to keep NULL elements as empty lists in JSON, since we want users to be able to easily add annotations, whether programmaticaly, using a web app, or manually.

Value

`a3` object

Author(s)

EDG

`toxt` *Create an `xt` object*

Description

Creates an `xt` object from time series data.

Usage

```
toxt(
  x,
  y,
  x2 = NULL,
  y2 = NULL,
  zt = NULL,
```

```
  shade = NULL,
  group = NULL,
  xunits = NULL,
  yunits = NULL,
  y2units = NULL,
  description = NULL,
  reference = NULL
)
```

Arguments

`x`	Named list of datetime vectors.
`y`	Named list of numeric vectors: When plotted, these will correspond to the left y-axis.
`x2`	Named list of datetime vectors: When plotted, these will correspond to the right x-axis. If not provided, `x` will be used for both `y` and `y2`.
`y2`	Named list of numeric vectors: When plotted, these will correspond to the right y-axis.
`zt`	Numeric vector: Zeitgeber time. If provided, this will be used to label x-axis ticks. Assumes a single datetime vector in `x`. Elements in `zt` must correspond to elements in `x`.
`shade`	Binary vector: `0` indicates no shading, `1` indicates shading. If provided, this will be used to shade the plot.
`group`	Named list of factors: Grouping variable(s).
`xunits`	Character: Units for `x`.
`yunits`	Character: Units for `y`.
`y2units`	Character: Units for `y2`.
`description`	Character: Description of the data / experiment.
`reference`	Character: Link to reference (journal publication, preprint, etc.)

Value

`xt` object

Author(s)

EDG

`uniprot_get` *Get protein sequence from UniProt*

Description

Get protein sequence from UniProt

Usage

```
uniprot_get(
  accession,
  baseURL = "https://rest.uniprot.org/uniprotkb",
  verbosity = 1
)
```

Arguments

`accession`	Character: UniProt Accession number - e.g. "Q9UMX9"
`baseURL`	Character: UniProt rest API base URL. Default = "https://rest.uniprot.org/uniprotkb"
`verbose`	Logical: If TRUE, print messages to console

Value

List with two elements: Annotation & Sequence

Author(s)

E.D. Gennatas

Examples

```
## Not run:
mapt <- uniprot_get("Q9UMX9")

## End(Not run)
```

`write.a3json` *Write* `a3` *object to JSON file*

Description

Write `a3` object to JSON file

Usage

```
write.a3json(x, filepath, overwrite = FALSE)
```

Arguments

`x`	`a3` object, as created by `as.a3()`.
`filepath`	Character: Path to save JSON file.

Author(s)

EDG

`write.xtjson` *Write* `xt` *object to JSON file*

Description

Write `xt` object to JSON file

Usage

```
write.xtjson(x, filepath, overwrite = FALSE)
```

Arguments

`x`	`xt` object, as created by toxt or as.xt.
`filepath`	Character: Path to save JSON file.

Author(s)

EDG

Index