### Description

`gettRNABasePairing` converts the dot bracket annotation into a `DotBracketDataFrame`. Base pairing is indicated by corresponding numbers in the forward and reverse columns. For more detail have a look at `getBasePairing`.

`gettRNALoopIDs` converts the dot bracket annotation into a `LoopIDList`. For more details have a look at `getLoopIDList`.

### Usage

```r
gettRNABasePairing(x, with.nucleotides = FALSE)
gettRNALoopIDs(x)
```

```r
## S4 method for signature 'GRanges'
gettRNABasePairing(x, with.nucleotides = FALSE)
```

```r
## S4 method for signature 'GRanges'
gettRNALoopIDs(x)
```

### Arguments

- `x` a GRanges object created by `import.tRNAscanAsGRanges` or GRanges with equivalent information. The `tRNA_str` and `tRNA_seq` columns will be used to construct a `StructuredXStringSet` and used for input into `getBasePairing`.

- `with.nucleotides` a single logical value: should the nucleotides be saved alongside the base pairing information in the 'base' column?
Value

gettRNABasePairing: The result is a DotBracketDataFrame with following columns: pos, forward, reverse, character and base. If a position is unpaired, forward and reverse will be 0, otherwise it will match the base paired positions.

gettRNALoopIDs: return a list of list of loop ids.

Examples

data("gr", package = "tRNA")
gettRNABasePairing(gr[1])
gettRNALoopIDs(gr[1])

Description

gettRNAFeaturePlots generates a plot for every feature found with gettRNASummary. Based on the datatype, it will generate suitable point or bar plots. Names of the GRangesList will be used as sample identifiers and used for colouring.

The options tRNA_colour_palette, tRNA_colour_yes and tRNA_colour_no will be used for colours.

Usage

gettRNAFeaturePlots(x, plotScores = FALSE, scores = NA, scoreLabel = "Score")

## S4 method for signature 'GRangesList'
gettRNAFeaturePlots(x, plotScores = FALSE, scores = NA, scoreLabel = "Score")

Arguments

x a named GRangesList object.
plotScores logical value, whether to plot scores. If scores are not provided with an additional argument, it will try to use the column "score" of the GRanges objects.
scores a list of scores, which have to have the same dimensions as the GRangesList or GRanges object.
scoreLabel a string to use as a label for the x axis.

Value

a list of ggplot2 plots. These can be customized further.
Examples

```r
data("gr", package = "tRNA")
data("gr_eco", package = "tRNA")
grl <- GRangesList(Sce = gr,
                   Eco = gr_eco)
plots <- gettRNAFeaturePlots(grl)
# customized plots
plots$length$layers <- plots$length$layers[c(-1,-2)]
plots$length + ggplot2::geom_boxplot()
```

Description

gettRNAstructureGRanges returns a list of GRanges describing the boundaries of tRNA structures as extracted from the dot bracket annotation. The dot bracket annotation is parsed using gettRNABasePairing, which internally uses getBasePairing.

gettRNAstructureSeq returns split or partial tRNA sequences based on the structure information. Variations in the length of structure features can be padded to retrieve sequences of equal length. If sequences are joined by setting joinCompletely = FALSE, the boundaries of the tRNA structure are stored in the result as metadata. They can be accessed as an IRanges object by using metadata(seq)[["tRNA_structures"]].

Usage

```r
gettRNAstructureGRanges(x, structure = "")
gettRNAstructureSeqs(
  x,
  structure = "",
  joinCompletely = FALSE,
  joinFeatures = FALSE,
  padSequences = TRUE
)
```

```r
## S4 method for signature 'GRanges'
gettRNAstructureSeqs(
  x,
  structure = "",
  joinCompletely = FALSE,
  joinFeatures = FALSE,
  padSequences = TRUE
)
```
## S4 method for signature 'GRanges'
gettRNAstructureGRanges(x, structure = "")

### Arguments

- **x**: a GRanges object with tRNA information. It has to pass the istRNAGRanges function.
- **structure**: optional parameter for returning just partial sequences. The following values are accepted: anticodonStem, Dprime5, DStem, Dloop, Dprime3, acceptorStem, anticodonloop, variableLoop, TStem, Tloop, discriminator. (default: structure = "")
- **joinCompletely**: Should the sequence parts, which are to be returned, be joined into one sequence? (default: joinCompletely = FALSE) Setting this to TRUE excludes joinFeatures be set to TRUE as well. In addition, joinCompletely = TRUE uses automatically all sequence structures.
- **joinFeatures**: Should the sequence parts, which are to be returned and are from the same structure type, be joined into one sequence? (default: joinCompletely = FALSE) Setting this to TRUE excludes joinCompletely be set to TRUE as well. joinCompletely takes precedence.
- **padSequences**: parameter whether sequences of the same type should be returned with the same length. For stems missing positions will be filled up in the middle, for loops at the ends. (default: padSequences = TRUE). If joinCompletely == TRUE this is set to TRUE automatically.

### Value

A list of GRanges or DNAStringSet objects. In case joinCompletely is set to TRUE a single DNAStringSet is returned.

### Examples

```r
data("gr", package = "tRNA")
gettRNAstructureGRanges(gr, structure = "anticodonLoop")
gettRNAstructureSeqs(gr, structure = "anticodonLoop")
gettRNABasePairing(gr[1:10])
```

---

### gettRNASummary

**Summary of tRNA features**

#### Description

gettRNASummary prepares a DataFrame with the aggregated features of tRNAs from a GRanges object. Logical values are converted to numeric values.
Usage

```r
gettRNASummary(x)
## S4 method for signature 'GRangesList'
gettRNASummary(x)
## S4 method for signature 'GRanges'
gettRNASummary(x)
```

Arguments

- `x`: a GRanges or a GRangesList object. All elements have to pass the `istRNAGRanges` test.

Value

- a DataFrame object

Examples

```r
data("gr", package = "tRNA")
gettRNASummary(gr)
```

Description

The functions `has*` can be used to subset the GRanges object containing information about tRNAs. Please not that the settings `mismatches` and `bulged` take precedence before `unpaired` or `paired`. This means that by setting either mismatches or bulged to either `TRUE` or `FALSE`, `unpaired = TRUE` or `paired = TRUE` are automatically set to allow specific subsetting. If this removes elements from the results, please consider constructing a logical vectors with two calls as suggested in the examples.

Usage

```r
hasTStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)
hasDStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)
hasAcceptorStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)
hasAnticodonStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)
hasTloop(x, length = NA)
```
hasTStem

hasDloop(x, length = NA)

hasAnticodonLoop(x, length = NA)

hasVariableLoop(x, length = NA, paired = NA, mismatches = NA, bulged = NA)

## S4 method for signature 'GRanges'
hasTStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)

## S4 method for signature 'GRanges'
hasDStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)

## S4 method for signature 'GRanges'
hasAcceptorStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)

## S4 method for signature 'GRanges'
hasAnticodonStem(x, length = NA, unpaired = NA, mismatches = NA, bulged = NA)

## S4 method for signature 'GRanges'
hasTloop(x, length = NA)

## S4 method for signature 'GRanges'
hasDloop(x, length = NA)

## S4 method for signature 'GRanges'
hasAnticodonLoop(x, length = NA)

## S4 method for signature 'GRanges'
hasVariableLoop(x, length = NA, paired = NA, mismatches = NA, bulged = NA)

Arguments

x
a GRanges object from a tRNAscan import or with equivalent information

length
the length as integer

unpaired
logical: has unpaired nucleotides

mismatches
logical: has mismatched nucleotides

bulged
logical: has mismatched nucleotides of different length creating a bulge

paired
logical: has paired nucleotides (only used for loops)

Value

a logical vector of the length or input GRanges object

Examples

data("gr", package = "tRNA")
hasTStem(gr, length = 5, mismatches = TRUE)
gr[hasTStem(gr, length = 5, mismatches = TRUE)]
gr[hasDStem(gr, unpaired = FALSE) & hasDStem(gr, mismatches = FALSE)]
istRNAGRanges

**Description**

istRNAGRanges checks whether a GRanges object contains the information expected for a tRNA result. This is used internally to ensure the required data is present in the input.

**Usage**

```r
istRNAGRanges(x)
```

**Arguments**

- `x` the GRanges object to test for compatibility.

**Value**

a logical value

**Examples**

```r
data("gr", package = "tRNA")
istRNAGRanges(gr)
```

---

**tRNA**

**tRNA: analyzing tRNA sequences and structures**

**Description**

The tRNA package allows feature information of tRNAs to be accessed and list of tRNA to be subset based on these features. The main purpose is to unify overlapping functions from the tRNAscanImport and tRNAdbImport packages. The functionality is currently under development and may change. The package expects a GRanges object with certain columns as input. The following columns are a requirement: tRNA_length, tRNA_type, tRNA_anticodon, tRNA_seq, tRNA_str, tRNA_CCA.end. Outputs of tRNAscanImport and tRNAdbImport meet these requirements.

Have a look at the vignette for an overview of the functionality. Additional functions are planned to be added in the future.

**Author(s)**

Felix G M Ernst [aut]
See Also

Useful links:

- Report bugs at https://github.com/FelixErnst/tRNA/issues

---

<table>
<thead>
<tr>
<th>tRNA-data</th>
<th>tRNA example data</th>
</tr>
</thead>
</table>

Description

Example data for using the tRNA package

Usage

data(gr)
data(gr_human)
data(gr_human2)
data(gr_eco)

Format

- object of class GRanges
- An object of class GRanges of length 596.
- An object of class GRanges of length 631.
- An object of class GRanges of length 89.
Index

* datasets
  * tRNA-data, 9

* tRNA
  * tRNA-data, 9

getBasePairing, 2
getLoopIDList, 2
gettRNABasePairing, 2
gettRNABasePairing, GRanges-method
(gettRNABasePairing), 2
gettRNAFeaturePlots, 3
gettRNAFeaturePlots, GRangesList-method
(gettRNAFeaturePlots), 3
gettRNALoopIDs (gettRNABasePairing), 2
gettRNALoopIDs, GRanges-method
(gettRNABasePairing), 2
gettRNAstructureGRanges, 4
gettRNAstructureGRanges, GRanges-method
(gettRNAstructureGRanges), 4
gettRNAstructureSeqs
(gettRNAstructureGRanges), 4
gettRNAstructureSeqs, GRanges-method
(gettRNAstructureGRanges), 4
gettRNAsummary, 5
gettRNAsummary, GRanges-method
(gettRNAsummary), 5
gettRNAsummary, GRangesList-method
(gettRNAsummary), 5
gr (tRNA-data), 9
gr_eco (tRNA-data), 9
gr_human (tRNA-data), 9
gr_human2 (tRNA-data), 9
hasAcceptorStem (hasTStem), 6
hasAcceptorStem, GRanges-method
(hasTStem), 6
hasAnticodonLoop (hasTStem), 6
hasAnticodonLoop, GRanges-method
(hasTStem), 6
hasAnticodonStem (hasTStem), 6
hasDloop (hasTStem), 6
hasDloop, GRanges-method (hasTStem), 6
hasDStem (hasTStem), 6
hasDStem, GRanges-method (hasTStem), 6
hasTloop (hasTStem), 6
hasTloop, GRanges-method (hasTStem), 6
hasTStem, 6
hasTStem, GRanges-method (hasTStem), 6
hasVariableLoop (hasTStem), 6
hasVariableLoop, GRanges-method
(hasTStem), 6
istRNAGRanges, 8
istRNAGRanges, GRanges-method
(istRNAGRanges), 8
tRNA, 8
tRNA-data, 9
tRNA-package (tRNA), 8
tRNA-subset (hasTStem), 6