

Package ‘GenePack’

January 10, 2025

Title Package for Gene Objects which Represent Different Gene Types

Version 1.0.0

Description Provides a comprehensive set of S4 classes to represent various human gene types, including both protein-coding and non-coding genes. The package includes constructors, validators, and accessor functions to manage and modify the gene objects created. Each gene object can store its essential information, particularly the Ensembl transcript ID, HUGO symbol, name, description, gene structure, and gene product (the product ID and sequence). In this implementation, each transcript isoform is treated as a separate gene, using the Ensembl transcript ID to simplify the representation and management of gene products, and provide more flexibility.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.4)

Imports methods, GenomicRanges, IRanges

biocViews Software, GeneExpression, Transcriptomics, RNASeq

VignetteBuilder knitr

Suggests testthat (>= 3.2.2), knitr, rmarkdown, devtools

Config/testthat/edition 3

URL <https://github.com/lulepo99/GenePack>

BugReports <https://github.com/lulepo99/GenePack/issues>

NeedsCompilation no

Author Luca Lepore [aut, cre]

Maintainer Luca Lepore <luca1.lepore@mail.polimi.it>

Contents

createGene	2
createLncRNAGene	4
createMicroRNAGene	6

createPiRNAGene	7
createProteinCodingGene	9
createRRNAGene	11
createSiRNAGene	13
createSnoRNAGene	15
createSnRNAGene	16
createTRNAGene	18
Gene-class	20
getClinicalSignificance	21
getDescription	22
getGeneStructure	23
getHugoSymbol	24
getID	25
getName	26
getProductID	27
getProductSequence	28
getTissues	29
HousekeepingRNAGene-class	30
lengthProduct	30
LncRNAGene-class	31
MicroRNAGene-class	32
PiRNAGene-class	33
ProteinCodingGene-class	33
RRNAGene-class	34
setClinicalSignificance<-	34
setDescription<-	35
setGeneStructure<-	36
setHugoSymbol<-	37
setID<-	39
setName<-	40
setProductID<-	41
setProductSequence<-	42
setTissues<-	43
showGeneObject	44
SiRNAGene-class	45
SncRNAGene-class	46
SnoRNAGene-class	46
SnRNAGene-class	47
TRNAGene-class	47
validateGeneProduct	48
validateID	48
validateSequence	49

createGene	<i>Create a Gene object</i>
------------	-----------------------------

Description

This function creates a Gene object, serving as a virtual base class for the creation of more specific gene types. The Gene class is not designed to be instantiated directly by users. It forms the basic structure from which all the specialized gene classes inherit.

Usage

```
createGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  clinical_significance = NA_character_
)
```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Optional parameter. Name of the gene.
description	character. Optional parameter. Description of the gene.
tissue_specificity	list. Optional parameter. List of tissues (character) where the gene is specifically expressed.
gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
clinical_significance	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets).

Details

The validity function within the class definition ensures that

- The id follows the Ensembl transcript format `^ENST[0-9]+$`.
- The hugo_symbol slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The name, and description are non-empty strings, if specified.
- The tissue_specificity slot is a list of tissues, if specified.

- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that 'start' is less than or equal to 'end' when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `clinical_significance` is a non-empty string, if specified.

Value

A `Gene` object.

Note

The `gene_structure` argument is complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` object in other contexts.

createLncRNAGene

Create a Long Non-Coding RNA Gene object

Description

This function creates a `LncRNAGene` object, which represents a gene that encodes for long non-coding RNAs. The `LncRNAGene` object includes base information inherited from the `Gene` class with the specific information regarding its lncRNA product.

Usage

```
createLncRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

<code>id</code>	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of <code>Gene</code> objects, avoiding the complexity of handling multiple gene products of different types within a single <code>Gene</code> object.
<code>hugo_symbol</code>	character. Hugo symbol of the gene.
<code>name</code>	character. Name of the gene.
<code>description</code>	character. Description of the gene.
<code>tissue_specificity</code>	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".

gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
gene_product	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
clinical_significance	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The id follows the Ensembl transcript format `^ENST[0-9]+$`.
- The hugo_symbol slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The name, and description are non-empty strings, if specified.
- The tissue_specificity slot is a list of tissues, if specified.
- The gene_structure slot is a GRanges object with valid chromosome, and strand. The IRanges object in GRanges directly verify that 'start' is less than or equal to 'end' when the user creates the GRanges object, so this check is not implemented in the package.
- The gene_product slot is a list that represents a gene product and contains two elements: the lncrna_id and the lncrna_sequence.
- The clinical_significance is a non-empty string, if specified.

Value

A LncRNAGene object.

Note

The gene_structure and gene_product arguments are complex objects and should be created separately by the user before constructing the Gene object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the GenomicRanges package and to easily reuse the gene_structure object in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGC UUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUU AACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUU AACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUA ACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUU AACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
```

```

name = "lncRNA gene name",
description = "gene description",
tissue_specificity = list("liver",
                          "small bowel"),
gene_structure = gene_structure,
gene_product = gene_product,
clinical_significance = "association with disease")

```

createMicroRNAGene	<i>Create a Micro RNA Gene object</i>
--------------------	---------------------------------------

Description

This function creates a MicroRNAGene object, which represents a gene that encodes for microRNAs. The MicroRNAGene object includes base information inherited from the Gene class with the specific information regarding its miRNA product.

Usage

```

createMicroRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)

```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Name of the gene.
description	character. Description of the gene.
tissue_specificity	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
gene_product	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).

clinical_significance

character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The id follows the Ensembl transcript format `^ENST[0-9]+$`.
- The hugo_symbol slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The name, and description are non-empty strings, if specified.
- The tissue_specificity slot is a list of tissues, if specified.
- The gene_structure slot is a GRanges object with valid chromosome, and strand. The IRanges object in GRanges directly verify that 'start' is less than or equal to 'end' when the user creates the GRanges object, so this check is not implemented in the package.
- The gene_product slot is a list that represents a gene product and contains two elements: the microrna_id and the microrna_sequence.
- The clinical_significance is a non-empty string, if specified.

Value

A MicroRNAGene object.

Note

The gene_structure and gene_product arguments are complex objects and should be created separately by the user before constructing the Gene object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the GenomicRanges package and to easily reuse the gene_structure in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(microrna_id = "miRNAID",
  microrna_sequence = "UAUGCACUGAUCCUAGCAUUG")

microrna_gene <- createMicroRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "miRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver",
    "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")
```

createPiRNAGene

Create a Piwi-interacting RNA Gene object

Description

This function creates a PiRNAGene object, which represents a gene that encodes for Piwi interacting RNAs. The PiRNAGene object includes base information inherited from the Gene class with the specific information regarding its piRNA product.

Usage

```
createPiRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Name of the gene.
description	character. Description of the gene.
tissue_specificity	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
gene_product	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
clinical_significance	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The id follows the Ensembl transcript format `^ENST[0-9]+$`.

- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters '-' or '_'.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that 'start' is less than or equal to 'end' when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `pirna_id` and the `pirna_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A `PiRNAGene` object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(pirna_id = "piRNAID",
                    pirna_sequence = "CUAUCGGCUUACGAUACCUUGGAUACGGA")

pirna_gene <- createPiRNAGene(id = "ENST000001",
                             hugo_symbol = "SYMBOL1",
                             name = "piRNA gene name",
                             description = "gene description",
                             tissue_specificity = list("liver",
                                                       "small bowel"),
                             gene_structure = gene_structure,
                             gene_product = gene_product,
                             clinical_significance = "association with disease")
```

createProteinCodingGene

Create a Protein Coding Gene object

Description

This function creates a `ProteinCodingGene` object, which represents a gene that encodes for proteins. The `ProteinCodingGene` object includes base information inherited from the `Gene` class with the specific information regarding its protein product.

Usage

```
createProteinCodingGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

<code>id</code>	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
<code>hugo_symbol</code>	character. Hugo symbol of the gene.
<code>name</code>	character. Name of the gene.
<code>description</code>	character. Description of the gene.
<code>tissue_specificity</code>	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
<code>gene_structure</code>	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
<code>gene_product</code>	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
<code>clinical_significance</code>	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The `id` follows the Ensembl transcript format `^ENST[0-9]+$`.
- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that `'start'` is less than or equal to `'end'` when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `protein_id` and the `protein_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A ProteinCodingGene object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the Gene object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(protein_id = "proteinID",
                    protein_sequence = "MASTATGVFRRAGY")

proteincoding_gene <- createProteinCodingGene(id = "ENST000001",
                                              hugo_symbol = "SYMBOL1",
                                              name = "protein-coding gene name",
                                              description = "gene description",
                                              tissue_specificity = list("liver",
                                                                      "small bowel"),
                                              gene_structure = gene_structure,
                                              gene_product = gene_product,
                                              clinical_significance = "association with disease")
```

createRRNAGene

Create a Ribosomal RNA Gene object

Description

This function creates a `RRNAGene` object, which represents a gene that encodes for ribosomal RNAs. The `RRNAGene` object includes base information inherited from the `Gene` class with the specific information regarding its rRNA product.

Usage

```
createRRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

<code>id</code>	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
<code>hugo_symbol</code>	character. Hugo symbol of the gene.
<code>name</code>	character. Name of the gene.
<code>description</code>	character. Description of the gene.
<code>tissue_specificity</code>	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
<code>gene_structure</code>	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
<code>gene_product</code>	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
<code>clinical_significance</code>	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The `id` follows the Ensembl transcript format `^ENST[0-9]+$`.
- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that `'start'` is less than or equal to `'end'` when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `rrna_id` and the `rrna_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A `RRNAGene` object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(rrna_id = "rRNAID",
                    rrna_sequence = paste0(
                        "GGUCUAUGCCUUCUAAGGGAUUGGCU",
                        "ACCUAAGGUACUCGGGAUCUUGAU"))

rrna_gene <- createRRNAGene(id = "ENST000001",
                           hugo_symbol = "SYMBOL1",
                           name = "rRNA gene name",
                           description = "gene description",
                           tissue_specificity = list("liver",
                                                    "small bowel"),
                           gene_structure = gene_structure,
                           gene_product = gene_product,
                           clinical_significance = "association with disease")
```

createSiRNAGene

Create a Short-interfering RNA Gene object

Description

This function creates a SiRNAGene object, which represents a gene that encodes for short interfering RNAs. The SiRNAGene object includes base information inherited from the Gene class with the specific information regarding its siRNA product.

Usage

```
createSiRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Name of the gene.


```

description = "gene description",
tissue_specificity = list("liver",
                        "small bowel"),
gene_structure = gene_structure,
gene_product = gene_product,
clinical_significance = "association with disease")

```

createSnoRNAGene

Create a Small Nucleolar RNA Gene object

Description

This function creates a SnoRNAGene object, which represents a gene that encodes for small nucleolar RNAs. The SnoRNAGene object includes base information inherited from the Gene class with the specific information regarding its snoRNA product.

Usage

```

createSnoRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)

```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Name of the gene.
description	character. Description of the gene.
tissue_specificity	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
gene_product	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
clinical_significance	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The `id` follows the Ensembl transcript format `^ENST[0-9]+$`.
- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that `'start'` is less than or equal to `'end'` when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `snorna_id` and the `snorna_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A `SnoRNA Gene` object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(snorna_id = "snoRNAID",
  snorna_sequence = paste0(
    "GUAUCGAUCUAUGCAUCCUACGUACGGAU",
    "GCUAACGUUCCGGAUUGCUAAUGGCUAAC"))

snorna_gene <- createSnoRNA Gene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "snoRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver",
    "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")
```

createSnRNAGene	<i>Create a Small Nuclear RNA Gene object</i>
-----------------	---

Description

This function creates a SnRNAGene object, which represents a gene that encodes for small nuclear RNAs. The SnRNAGene object includes base information inherited from the Gene class with the specific information regarding its snRNA product.

Usage

```
createSnRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

id	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
hugo_symbol	character. Hugo symbol of the gene.
name	character. Name of the gene.
description	character. Description of the gene.
tissue_specificity	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
gene_structure	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
gene_product	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
clinical_significance	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The id follows the Ensembl transcript format `^ENST[0-9]+$`.

- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters '-' or '_'.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that 'start' is less than or equal to 'end' when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `snrna_id` and the `snrna_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A `SnRNAGene` object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(snrna_id = "snRNAID",
  snrna_sequence = paste0(
    "AUGCGGAUUUACGCCUAACGUUCCG",
    "GAUUACCUCGGAUCCGA"))

snrna_gene <- createSnRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "snRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver",
    "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")
```

createTRNAGene

Create a Transfer RNA Gene object

Description

This function creates a `TRNAGene` object, which represents a gene that encodes for transfer RNAs. The `TRNAGene` object includes base information inherited from the `Gene` class with the specific information regarding its tRNA product.

Usage

```
createTRNAGene(
  id,
  hugo_symbol,
  name = NA_character_,
  description = NA_character_,
  tissue_specificity = list(),
  gene_structure,
  gene_product,
  clinical_significance = NA_character_
)
```

Arguments

<code>id</code>	character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.
<code>hugo_symbol</code>	character. Hugo symbol of the gene.
<code>name</code>	character. Name of the gene.
<code>description</code>	character. Description of the gene.
<code>tissue_specificity</code>	list. Optional parameter. List of tissues (character) where the gene is specifically expressed. If not specified, its default will be "-".
<code>gene_structure</code>	GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).
<code>gene_product</code>	list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).
<code>clinical_significance</code>	character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets). If not specified, its default will be "-".

Details

The validity function within the class definition ensures that

- The `id` follows the Ensembl transcript format `^ENST[0-9]+$`.
- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that `'start'` is less than or equal to `'end'` when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: the `trna_id` and the `trna_sequence`.
- The `clinical_significance` is a non-empty string, if specified.

Value

A TRNAGene object.

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the `Gene` object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the `GenomicRanges` package and to easily reuse the `gene_structure` in other contexts.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(trna_id = "tRNAID",
                    trna_sequence = paste0(
                      "GCCUAUACGUUACGAUCUCGAAUAGCCUUACGUUAACCG",
                      "UAUGCUAUGGUGACCGUAUGCUAUGGUACGUACGAA"))

trna_gene <- createTRNAGene(id = "ENST000001",
                           hugo_symbol = "SYMBOL1",
                           name = "tRNA gene name",
                           description = "gene description",
                           tissue_specificity = list("liver", "small bowel"),
                           gene_structure = gene_structure,
                           gene_product = gene_product,
                           clinical_significance = "association with disease")
```

Gene-class	<i>Gene class</i>
------------	-------------------

Description

A virtual class to represent genes

Details

The validity function ensures that

- The `id` follows the Ensembl transcript format `^ENST[0-9]+$`.
- The `hugo_symbol` slot contains only uppercase letters, digits, and special characters `'-'` or `'_'`.
- The `name`, and `description` are non-empty strings, if specified.
- The `tissue_specificity` slot is a list of tissues, if specified.
- The `gene_structure` slot is a `GRanges` object with valid chromosome, and strand. The `IRanges` object in `GRanges` directly verify that `'start'` is less than or equal to `'end'` when the user creates the `GRanges` object, so this check is not implemented in the package.
- The `gene_product` slot is a list that represents a gene product and contains two elements: an ID and a sequence.
- The `clinical_significance` is a non-empty string, if specified.

Value

This documentation describes the structure of the virtual Gene class.

Slots

`id` character. Ensembl transcript ID of the gene. In this implementation, each transcript isoform is considered as an individual gene. Hence, the Ensembl transcript ID ("ENST") is used instead of the Ensembl gene ID ("ENSG"). This approach simplifies the management of Gene objects, avoiding the complexity of handling multiple gene products of different types within a single Gene object.

`hugo_symbol` character. Hugo symbol of the gene.

`name` character. Optional parameter. Name of the gene.

`description` character. Optional parameter. Description of the gene.

`tissue_specificity` list. Optional parameter. List of tissues (character) where the gene is specifically expressed.

`gene_structure` GenomicRanges::GRanges. Structure of the gene, including chromosomes ("seqnames", character), start position ("start", numeric), end position ("end", numeric), strand ("strand", character).

`gene_product` list. It represents the product of the gene and contains its product id (character) and the corresponding sequence (character).

`clinical_significance` character. Optional parameter. Clinical relevance of the gene or its product (e.g., association with a disease or therapeutic targets).

Note

The `gene_structure` and `gene_product` arguments are complex objects and should be created separately by the user before constructing the Gene object. This approach maximizes flexibility, allowing the user to better exploit the functionalities provided by the GenomicRanges package and to easily reuse the `gene_structure` object in other contexts.

```
getClinicalSignificance
```

Get the clinical significance

Description

This function retrieves the clinical significance of the gene.

Usage

```
getClinicalSignificance(object)
```

```
## S4 method for signature 'Gene'
getClinicalSignificance(object)
```

Arguments

`object` Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the clinical significance of the gene, or "-" if it was not specified at the creation.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

getClinicalSignificance(lncrna_gene)
```

getDescription	<i>Get the gene description</i>
----------------	---------------------------------

Description

This function retrieves the description of the gene.

Usage

```
getDescription(object)

## S4 method for signature 'Gene'
getDescription(object)
```

Arguments

object	Gene object
--------	-------------

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the description of the gene.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

getDescription(lncrna_gene)
```

getGeneStructure	<i>Get the gene structure</i>
------------------	-------------------------------

Description

This function retrieves the gene_structure GRanges object.

Usage

```
getGeneStructure(object)

## S4 method for signature 'Gene'
getGeneStructure(object)
```

Arguments

object	Gene object
--------	-------------

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the gene_structure object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

getGeneStructure(lncrna_gene)
```

getHugoSymbol

Get the Hugo symbol

Description

This function retrieves the HUGO symbol associated with a Gene object.

Usage

```
getHugoSymbol(object)

## S4 method for signature 'Gene'
getHugoSymbol(object)
```

Arguments

object Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the HUGO symbol of the gene.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                        "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
                        "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
                        "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
                        "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
                        "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
                        "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNA Gene(id = "ENST000001",
                                hugo_symbol = "SYMBOL1",
                                name = "lncRNA gene name",
                                description = "gene description",
                                tissue_specificity = list("liver", "small bowel"),
                                gene_structure = gene_structure,
                                gene_product = gene_product,
                                clinical_significance = "association with disease")

getHugoSymbol(lncrna_gene)
```

getID

Get the Ensembl transcript ID

Description

This function retrieves the Ensembl transcript ID associated with a Gene object.

Usage

```
getID(object)

## S4 method for signature 'Gene'
getID(object)
```

Arguments

object Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the Ensembl transcript ID of the gene.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                        "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
                        "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
                        "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
                        "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
                        "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
                        "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
                               hugo_symbol = "SYMBOL1",
                               name = "lncRNA gene name",
                               description = "gene description",
                               tissue_specificity = list("liver", "small bowel"),
                               gene_structure = gene_structure,
                               gene_product = gene_product,
                               clinical_significance = "association with disease")

getID(lncrna_gene)
```

getName	<i>Get the gene name</i>
---------	--------------------------

Description

This function retrieves the name of the gene.

Usage

```
getName(object)

## S4 method for signature 'Gene'
getName(object)
```

Arguments

object Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the name of the gene.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                        "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
                        "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
                        "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
                        "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
                        "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
                        "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
                               hugo_symbol = "SYMBOL1",
                               name = "lncRNA gene name",
                               description = "gene description",
                               tissue_specificity = list("liver", "small bowel"),
                               gene_structure = gene_structure,
                               gene_product = gene_product,
                               clinical_significance = "association with disease")

getName(lncrna_gene)
```

getProductID

Get the gene product ID

Description

This function retrieves the gene product ID.

Usage

```
getProductID(object)

## S4 method for signature 'Gene'
getProductID(object)
```

Arguments

object Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. The gene product ID is stored as the second element in the "gene_product" list.

Value

the gene product ID.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

getProductID(lncrna_gene)
```

getProductSequence	<i>Get the gene product sequence</i>
--------------------	--------------------------------------

Description

This function retrieves the gene product sequence.

Usage

```
getProductSequence(object)

## S4 method for signature 'Gene'
getProductSequence(object)
```

Arguments

object	Gene object
--------	-------------

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. The gene product sequence is stored as the second element in the "gene_product" list. The function is implemented to provide a more readable output, displaying the sequence in blocks of 80 bases per line.

Value

the gene product sequence.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                        "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
                        "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
                        "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
                        "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
                        "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
                        "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
                               hugo_symbol = "SYMBOL1",
                               name = "lncRNA gene name",
                               description = "gene description",
                               tissue_specificity = list("liver", "small bowel"),
                               gene_structure = gene_structure,
                               gene_product = gene_product,
                               clinical_significance = "association with disease")

getProductSequence(lncrna_gene)
```

getTissues

Get the tissue specificity list

Description

This function retrieves the list of tissues where the gene is specifically expressed.

Usage

```
getTissues(object)

## S4 method for signature 'Gene'
getTissues(object)
```

Arguments

object Gene object

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class.

Value

the tissue specificity of the gene, or "-" if it was not specified at the creation.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                        "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
                        "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
                        "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
                        "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
                        "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
                        "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
                               hugo_symbol = "SYMBOL1",
                               name = "lncRNA gene name",
                               description = "gene description",
                               tissue_specificity = list("liver", "small bowel"),
                               gene_structure = gene_structure,
                               gene_product = gene_product,
                               clinical_significance = "association with disease")
```

HousekeepingRNAGene-class

Housekeeping RNA gene class

Description

A virtual class to represent housekeeping RNA genes. This class is a general S4 class inheriting from the Gene class and serves as a base class to represent specific types of housekeeping RNA gene.

Value

This documentation describes the structure of the virtual HousekeepingRNAGene class.

lengthProduct	<i>Compute the length of a gene product for a gene object</i>
---------------	---

Description

This function computes the length of the product of a gene object. It provides an integer value that represents the number of nucleotides or amino acids in the sequence, depending on the Gene object class.

Usage

```
lengthProduct(object)

## S4 method for signature 'Gene'
lengthProduct(object)
```

Arguments

object Gene object. An object of a specific Gene class (e.g., "ProteinCodingGene", "LncRNAGene", "SiRNAGene", etc.).

Details

Based on the class of the Gene object, the function choose the appropriate method to compute the length of the gene product.

Value

an integer representing the length of the gene product.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                         ranges = IRanges::IRanges(start = 200, end = 1200),
                                         strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                    lncrna_sequence = paste0(
                      "AUGC UUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCG",
                      "GCUAAUCGGCUUAGCGUACGGUAGGCUU AACUGCGUACGAU",
                      "CGAUCGGCUAAGCGUACGGUAGGCUU AACUGCGUACGAUCG",
                      "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
                      "GGCUUAGCGUACGGUAGGCUU AACUGCGGACGAUCGAUCGG",
                      "CUUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
                               hugo_symbol = "SYMBOL1",
                               name = "lncRNA gene name",
                               description = "gene description",
                               tissue_specificity = list("liver", "small bowel"),
                               gene_structure = gene_structure,
                               gene_product = gene_product,
                               clinical_significance = "association with disease")
```

```
lengthProduct(lncrna_gene)
```

LncRNAGene-class

Long non-coding RNA Gene class

Description

A class to represent long non-coding RNA genes This class is a specialized S4 class inheriting from the Gene class and is designed to store information about long non-coding RNA genes, including their lncRNA product.

Details

The gene_product slot is expected to contain a lncRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid lncRNA sequence.

Value

This documentation describes the structure of the LncRNAGene class.

Slots

gene_product list. This slot is specific for the long non-coding RNA genes product and includes:

- lncrna_id: a string representing the ID of the lncRNA.
- lncrna_sequence: a string representing the sequence of the lncRNA.

MicroRNAGene-class

MicroRNA gene class

Description

A class to represent microRNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about microRNA genes, including their microRNA product.

Details

The gene_product slot is expected to contain a microRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid microRNA sequence.

Value

This documentation describes the structure of the MicroRNAGene class.

Slots

gene_product list. This slot is specific for the microRNA genes product and includes:

- microrna_id: a string representing the ID of the microRNA.
- microrna_sequence: a string representing the sequence of the microRNA.

PiRNAGene-class	<i>Piwi-interacting RNA gene class</i>
-----------------	--

Description

A class to represent piwi-interacting RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about piRNA genes, including their piRNA products.

Details

The gene_product slot is expected to contain a piRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid piRNA sequence.

Value

This documentation describes the structure of the PiRNAGene class.

Slots

gene_product list. This slot is specific for the piRNA genes product and includes:

- pirna_id: a string representing the ID of the piRNA.
- pirna_sequence: a string representing the sequence of the piRNA.

ProteinCodingGene-class	<i>Protein-coding RNA gene class</i>
-------------------------	--------------------------------------

Description

A class to represent protein-coding genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about protein-coding genes, including their protein product.

Details

The gene_product slot is expected to contain a protein ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid protein sequence.

Value

This documentation describes the structure of the ProteinCodingGene class.

Slots

gene_product list. This slot is specific for the protein-coding genes product and includes:

- protein_id: a string representing the ID of the protein.
- protein_sequence: a string representing the sequence of the protein.

RRNAGene-class	<i>Ribosomal RNA gene class</i>
----------------	---------------------------------

Description

A class to represent ribosomal RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about rRNA genes, including their rRNA product.

Details

The gene_product slot is expected to contain a rRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid rRNA sequence.

Value

This documentation describes the structure of the RRNAGene class.

Slots

gene_product list. This slot is specific for the rRNA genes product and includes:

- rrna_id: a string representing the ID of the rRNA.
- rrna_sequence: a string representing the sequence of the rRNA.

setClinicalSignificance<-	<i>Set the clinical significance</i>
---------------------------	--------------------------------------

Description

This function sets the clinical significance of the gene.

Usage

```
setClinicalSignificance(object) <- value

## S4 replacement method for signature 'Gene'
setClinicalSignificance(object) <- value
```

Arguments

object	Gene object.
value	the new clinical significance.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new clinical significance, the function checks that the Gene object is still valid by calling validObject.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUIAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease1")

setClinicalSignificance(lncrna_gene) <- "association with disease2"

getClinicalSignificance(lncrna_gene)
```

setDescription<-	<i>Set the gene description</i>
------------------	---------------------------------

Description

This function sets the description of the gene.

Usage

```
setDescription(object) <- value

## S4 replacement method for signature 'Gene'
setDescription(object) <- value
```

Arguments

object	Gene object.
value	the new description of the gene.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new description, the function checks that the Gene object is still valid by calling validObject.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

setDescription(lncrna_gene) <- "new gene description"

getDescription(lncrna_gene)
```

setGeneStructure<- *Set the gene structure*

Description

This function sets the gene_structure GRanges object.

Usage

```
setGeneStructure(object) <- value

## S4 replacement method for signature 'Gene'
setGeneStructure(object) <- value
```

Arguments

object Gene object.
value the new gene_structure object.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new gene structure, the function checks that the Gene object is still valid by calling `validObject`. This ensures that all the internal attributes of the GRanges object are checked according to the validity conditions specified in the virtual Gene class.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

new_gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 205, end = 1210),
  strand = "+")

setGeneStructure(lncrna_gene) <- new_gene_structure

getGeneStructure(lncrna_gene)
```

setHugoSymbol<- *Set the Hugo symbol*

Description

This function sets the HUGO symbol for a Gene object.

Usage

```
setHugoSymbol(object) <- value

## S4 replacement method for signature 'Gene'
setHugoSymbol(object) <- value
```

Arguments

<code>object</code>	Gene object.
<code>value</code>	the new HUGO symbol.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new HUGO symbol, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

setHugoSymbol(lncrna_gene) <- "NEWSYMBOL1"

getHugoSymbol(lncrna_gene)
```

setID<-	<i>Set the Ensembl transcript ID</i>
---------	--------------------------------------

Description

This function sets the Ensembl transcript ID for a Gene object.

Usage

```
setID(object) <- value

## S4 replacement method for signature 'Gene'
setID(object) <- value
```

Arguments

object	Gene object.
value	the new Ensembl transcript ID.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new ID, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")
```

```
setID(lncrna_gene) <- "ENST000002"

getID(lncrna_gene)
```

setName<-	<i>Set the gene name</i>
-----------	--------------------------

Description

This function sets the name of the gene.

Usage

```
setName(object) <- value

## S4 replacement method for signature 'Gene'
setName(object) <- value
```

Arguments

object	Gene object.
value	the new name of the gene.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new name, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGC UUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUU AACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUU AACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUU AACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUU AACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
```



```

tissue_specificity = list("liver", "small bowel"),
gene_structure = gene_structure,
gene_product = gene_product,
clinical_significance = "association with disease")

setName(lncrna_gene) <- "new gene name"

getName(lncrna_gene)

```

setProductID<-	<i>Set the gene product ID</i>
----------------	--------------------------------

Description

This function sets the gene product ID.

Usage

```

setProductID(object) <- value

## S4 replacement method for signature 'Gene'
setProductID(object) <- value

```

Arguments

object	Gene object.
value	the new product ID of the gene product.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new product ID, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```

gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAACUGCGUACGAUCGAUCGC"))

```

```
lncrna_gene <- createLncRNAGene(id = "ENST000001",
                                hugo_symbol = "SYMBOL1",
                                name = "lncRNA gene name",
                                description = "gene description",
                                tissue_specificity = list("liver", "small bowel"),
                                gene_structure = gene_structure,
                                gene_product = gene_product,
                                clinical_significance = "association with disease")

setProductID(lncrna_gene) <- "newLncRNAID"

getProductID(lncrna_gene)
```

```
setProductSequence<- Set the gene product sequence
```

Description

This function sets the gene product sequence.

Usage

```
setProductSequence(object) <- value

## S4 replacement method for signature 'Gene'
setProductSequence(object) <- value
```

Arguments

object	Gene object.
value	the new product sequence of the gene product.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new product sequence, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
                                           ranges = IRanges(start = 200, end = 1200),
                                           strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
                     lncrna_sequence = paste0(
                       "AUGCUUAGCGUACGGUAGGCUUAACUGCGUACGAUCGAUCG",
```

```

"GCUAUUCGGCUUAGCGUACGGUAGGCUUACUGCGUACGAU",
"CGAUCGGCUAAGCGUACGGUAGGCUUACUGCGUACGAUCG",
"AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
"GGCUUAGCGUACGGUAGGCUUACUGCGGACGAUCGAUCGG",
"CUUAGCGUACGGUAGGCUUACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

setProductSequence(lncrna_gene) <- paste0(
  "AUGCUUAGCGUACGGUAGGCUUACUGCGUACGAUCGAUCG",
  "CGAUCGGCUAAGCGUACGGUAGGCUUACUGCGUACGAUCG",
  "GCUAUUCGGCUUAGCGUACGGUAGGCUUACUGCGUACGAU",
  "GGCUUAGCGUACGGUAGGCUUACUGCGGACGAUCGAUCGG",
  "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
  "CUUAGCGUACGGUAGGCUUACUGCGUACGAUCGAUCGC")

getProductSequence(lncrna_gene)

```

setTissues<-	<i>Set the tissue specificity list</i>
--------------	--

Description

This function sets the list of tissues where the gene is specifically expressed.

Usage

```

setTissues(object) <- value

## S4 replacement method for signature 'Gene'
setTissues(object) <- value

```

Arguments

object	Gene object.
value	the new list of tissues.

Details

This function is defined as a generic function to be applicable to any object that inherits from the Gene class. After setting the new tissue specificity, the function checks that the Gene object is still valid by calling `validObject`.

Value

the modified Gene object.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUAAACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUAAACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUCAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUAAACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUAAACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

setTissues(lncrna_gene) <- list("liver", "large bowel")

getTissues(lncrna_gene)
```

showGeneObject	<i>Show Gene object information</i>
----------------	-------------------------------------

Description

This function displays the gene object information in a more suitable and detailed way with respect to the standard 'show' method implemented in R.

Usage

```
showGeneObject(object)

## S4 method for signature 'Gene'
showGeneObject(object)
```

Arguments

object	Gene object. An object of a specific Gene class (e.g., "ProteinCodingGene", "LncRNAGene", "SiRNAGene", etc.).
--------	---

Details

The gene_structure slot is an S4 object of class GRanges, which is implemented in order to give more flexibility in its creation. To ensure a more standardized and reliable output format, the R show function is used within the showGeneObject function to display it. Additionally, the product sequence is formatted into blocks, with each line containing 80 bases for better readability.

Value

This function does not return a value. It displays the gene object information to the console.

Examples

```
gene_structure <- GenomicRanges::GRanges(seqnames = "chr1",
  ranges = IRanges::IRanges(start = 200, end = 1200),
  strand = "+")

gene_product <- list(lncrna_id = "lncRNAID",
  lncrna_sequence = paste0(
    "AUGCUUAGCGUACGGUAGGCUUACUGCGUACGAUCGAUCG",
    "GCUAAUCGGCUUAGCGUACGGUAGGCUUACUGCGUACGAU",
    "CGAUCGGCUAAGCGUACGGUAGGCUUACUGCGUACGAUCG",
    "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
    "GGCUUAGCGUACGGUAGGCUUACUGCGGACGAUCGAUCGG",
    "CUUAGCGUACGGUAGGCUUACUGCGUACGAUCGAUCGC"))

lncrna_gene <- createLncRNAGene(id = "ENST000001",
  hugo_symbol = "SYMBOL1",
  name = "lncRNA gene name",
  description = "gene description",
  tissue_specificity = list("liver", "small bowel"),
  gene_structure = gene_structure,
  gene_product = gene_product,
  clinical_significance = "association with disease")

showGeneObject(lncrna_gene)
```

SiRNAGene-class

Short-interfering RNA gene class

Description

A class to represent short-interfering RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about siRNA genes, including their siRNA product.

Details

The gene_product slot is expected to contain a siRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid siRNA sequence.

Value

This documentation describes the structure of the SiRNAGene class.

Slots

gene_product list. This slot is specific for the siRNA genes product and includes:

- siRNA_id: a string representing the ID of the siRNA.
- siRNA_sequence: a string representing the sequence of the siRNA.

SncRNAGene-class	<i>Small non-coding RNA gene class</i>
------------------	--

Description

A virtual class to represent small non-coding RNA genes. This class is a general S4 class inheriting from the Gene class and serves as a base class to represent specific types of non small non-coding RNA gene.

Value

This documentation describes the structure of the virtual SncRNAGene class.

SnoRNAGene-class	<i>Small nucleolar RNA gene class</i>
------------------	---------------------------------------

Description

A class to represent small nucleolar RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about snoRNA genes, including their snoRNA products.

Details

The gene_product slot is expected to contain a snoRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid snoRNA sequence.

Value

This documentation describes the structure of the SnoRNAGene class.

Slots

gene_product list. This slot is specific for the snoRNA genes product and includes:

- snorna_id: a string representing the ID of the snoRNA.
- snorna_sequence: a string representing the sequence of the snoRNA.

SnRNAGene-class	<i>Small nuclear RNA gene class</i>
-----------------	-------------------------------------

Description

A class to represent small nuclear RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about snRNA genes, including their snRNA products.

Details

The gene_product slot is expected to contain a snRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid snRNA sequence.

Value

This documentation describes the structure of the SnRNAGene class.

Slots

gene_product list. This slot is specific for the snRNA genes product and includes:

- snrna_id: a string representing the ID of the snRNA.
- snrna_sequence: a string representing the sequence of the snRNA.

TRNAGene-class	<i>Transfer RNA gene class</i>
----------------	--------------------------------

Description

A class to represent transfer RNA genes. This class is a specialized S4 class inheriting from the Gene class and is designed to store information about tRNA genes, including their tRNA products.

Details

The gene_product slot is expected to contain a tRNA ID and the corresponding sequence. The validity function ensures that it is correctly formatted and contain a valid tRNA sequence.

Value

This documentation describes the structure of the TRNAGene class.

Slots

gene_product list. This slot is specific for the tRNA genes product and includes:

- trna_id: a string representing the ID of the tRNA.
- trna_sequence: a string representing the sequence of the tRNA.

validateGeneProduct	<i>Validate Gene Product</i>
---------------------	------------------------------

Description

This function validates both the gene product ID and sequence using internal methods. Users can use it to validate the gene products before creating the gene object. However, the function is also internal in the definition of the gene object classes to guarantee robustness.

Usage

```
validateGeneProduct(id, sequence, type)
```

Arguments

id	character. The gene product ID to validate.
sequence	character. The gene product sequence to validate.
type	character. The RNA type: "lncRNA", "sncRNA" (including microRNA, piRNA, snRNA, snoRNA, siRNA), "tRNA", "rRNA", "protein".

Value

TRUE if both the ID and sequence are valid, throws an error otherwise.

Examples

```
lncrna_id <- "lncRNAID"

lncrna_sequence <- paste0(
  "AUGCUUAGCGUACGGUAGGCUUAACUGCGUACGAUCGAUCG",
  "GCUAAUCGGCUUAGCGUACGGUAGGCUUAACUGCGUACGAU",
  "CGAUCGGCUAAGCGUACGGUAGGCUUAACUGCGUACGAUCG",
  "AUCGGCUUAGCGUACGUAGGCUAACUGCGUACGAUCGAUC",
  "GGCUUAGCGUACGGUAGGCUUAACUGCGGACGAUCGAUCGG",
  "CUUAGCGUACGGUAGGCUUAACUGCGUACGAUCGAUCGC")

type <- "lncRNA"

validateGeneProduct(lncrna_id, lncrna_sequence, type)
```

validateID	<i>Validate Gene Product ID</i>
------------	---------------------------------

Description

This function validates the gene product ID to ensure it is a non-empty string.

Usage

```
validateID(id)
```


Arguments

id character. The gene product ID to validate.

Value

TRUE if the gene product ID is valid, throws an error otherwise.

validateSequence	<i>Validate Gene Product Sequence</i>
------------------	---------------------------------------

Description

This function validates the gene product sequence, ensuring it is a string containing only valid nucleotide bases or amino acids, depending on the Gene object class. Besides, it checks if the gene product sequence meets specific length requirements based on its type.

Usage

```
validateSequence(sequence, type)
```

Arguments

sequence character. The gene product sequence to validate.
type character. The gene product type (e.g., "lncRNA", "sncRNA", "tRNA").

Details

Regarding the length requirements, the function allows some flexibility due to the variable length ranges defined for some gene types. A long non-coding RNA gene product must be at least 200 nucleotides in length. All small non coding genes are considered together and their product must be shorter than 200 nucleotides. A tRNA gene product ranges from 70 to 90 nucleotides, in general. No checks are defined for rRNA gene products because of their high variability in nucleotide sequences instead.

Value

TRUE if the gene product sequence is valid, throws an error otherwise.
TRUE if the gene product length is valid, throws an error otherwise.

Index

- * **classes**
 - Gene-class, [20](#)
- * **function**
 - validateGeneProduct, [48](#)
- * **gene,**
 - lengthProduct, [30](#)
- * **gene**
 - validateGeneProduct, [48](#)
- * **internal**
 - createGene, [2](#)
 - HousekeepingRNAGene-class, [30](#)
 - SncRNAGene-class, [46](#)
 - validateID, [48](#)
 - validateSequence, [49](#)
- * **length,**
 - lengthProduct, [30](#)
- * **product**
 - lengthProduct, [30](#)
 - validateGeneProduct, [48](#)
- * **sequence,**
 - lengthProduct, [30](#)
- * **validation**
 - validateGeneProduct, [48](#)

[createGene, 2](#)
[createLncRNAGene, 4](#)
[createMicroRNAGene, 6](#)
[createPiRNAGene, 7](#)
[createProteinCodingGene, 9](#)
[createRRNAGene, 11](#)
[createSiRNAGene, 13](#)
[createSnoRNAGene, 15](#)
[createSnRNAGene, 16](#)
[createTRNAGene, 18](#)

[Gene-class, 20](#)
[getClinicalSignificance, 21](#)
[getClinicalSignificance, Gene-method \(getClinicalSignificance\), 21](#)
[getDescription, 22](#)
[getDescription, Gene-method \(getDescription\), 22](#)
[getGeneStructure, 23](#)
[getGeneStructure, Gene-method \(getGeneStructure\), 23](#)
[getHugoSymbol, 24](#)
[getHugoSymbol, Gene-method \(getHugoSymbol\), 24](#)
[getID, 25](#)
[getID, Gene-method \(getID\), 25](#)
[getName, 26](#)
[getName, Gene-method \(getName\), 26](#)
[getProductID, 27](#)
[getProductID, Gene-method \(getProductID\), 27](#)
[getProductSequence, 28](#)
[getProductSequence, Gene-method \(getProductSequence\), 28](#)
[getTissues, 29](#)
[getTissues, Gene-method \(getTissues\), 29](#)

[HousekeepingRNAGene-class, 30](#)

[lengthProduct, 30](#)
[lengthProduct, Gene-method \(lengthProduct\), 30](#)
[LncRNAGene-class, 31](#)

[MicroRNAGene-class, 32](#)

[PiRNAGene-class, 33](#)
[ProteinCodingGene-class, 33](#)

[RRNAGene-class, 34](#)

[setClinicalSignificance<-, 34](#)
[setClinicalSignificance<-, Gene-method \(setClinicalSignificance<-\), 34](#)
[setDescription<-, 35](#)
[setDescription<-, Gene-method \(setDescription<-\), 35](#)
[setGeneStructure<-, 36](#)
[setGeneStructure<-, Gene-method \(setGeneStructure<-\), 36](#)
[setHugoSymbol<-, 37](#)
[setHugoSymbol<-, Gene-method \(setHugoSymbol<-\), 37](#)
[setID<-, 39](#)

setID<-, Gene-method (setID<-), 39
setName<-, 40
setName<-, Gene-method (setName<-), 40
setProductID<-, 41
setProductID<-, Gene-method
 (setProductID<-), 41
setProductSequence<-, 42
setProductSequence<-, Gene-method
 (setProductSequence<-), 42
setTissues<-, 43
setTissues<-, Gene-method
 (setTissues<-), 43
showGeneObject, 44
showGeneObject, Gene-method
 (showGeneObject), 44
SiRNAGene-class, 45
SncRNAGene-class, 46
SnoRNAGene-class, 46
SnRNAGene-class, 47

TRNAGene-class, 47

validateGeneProduct, 48
validateID, 48
validateSequence, 49