Package 'pgxRpi'

October 30, 2025

Title R wrapper for Progenetix

Version 1.7.0

Description The package is an R wrapper for Progenetix REST API built upon the Beacon v2 protocol. Its purpose is to provide a seamless way for retrieving genomic data from Progenetix database—an open resource dedicated to curated oncogenomic profiles. Empowered by this package, users can effortlessly access and visualize data from Progenetix.

biocViews CopyNumberVariation, GenomicVariation, DataImport, Software

License Artistic-2.0

Encoding UTF-8

LazyData FALSE

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports utils, methods, grDevices, graphics, circlize, httr, dplyr, attempt, lubridate, survival, survminer, ggplot2, GenomicRanges, SummarizedExperiment, S4Vectors, yaml, parallel, future, future.apply

Depends R (>= 4.2)

Suggests BiocStyle, rmarkdown, knitr, testthat

BugReports https://github.com/progenetix/pgxRpi/issues

URL https://github.com/progenetix/pgxRpi

VignetteBuilder knitr

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

git_branch devel

git_last_commit 8de3d7a

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-10-30

Author Hangjia Zhao [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8376-5751>),
Michael Baudis [aut] (ORCID: <https://orcid.org/0000-0002-9903-4248>)

Maintainer Hangjia Zhao <hangjia.zhao@uzh.ch>

Contents

`hg19_cytoband` *A dataframe containing cytoband annotation details extracted from the hg19 gennome. It is used for CNV frequency visualization.*

Description

A dataframe containing cytoband annotation details extracted from the hg19 gennome. It is used for CNV frequency visualization.

Usage

```
hg19_cytoband
```

Format

An object of class `data.frame` with 862 rows and 5 columns.

Value

cytoband of hg19 genome

Source

http://hgdownload.cse.ucsc.edu/goldenpath/hg19/database/cytoBand.txt.gz

hg38_cytoband *A dataframe containing cytoband annotation details extracted from the hg38 gennome. It is used for CNV frequency visualization.*

Description

A dataframe containing cytoband annotation details extracted from the hg38 gennome. It is used for CNV frequency visualization.

Usage

```
hg38_cytoband
```

Format

An object of class `data.frame` with 862 rows and 5 columns.

Value

cytoband of hg38 genome

Source

http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz

pgxFreqplot *Plot CNV frequency data*

Description

Thie function plots the frequency of deletions and duplications

Usage

```
pgxFreqplot(
  data,
  chrom = NULL,
  layout = c(1, 1),
  filters = NULL,
  circos = FALSE,
  assembly = "hg38"
)
```

Arguments

data	CNV frequency object returned by the pgxLoader or segtoFreq functions.
chrom	A vector specifying which chromosomes to plot. If NULL, the plot will cover the entire genome. If specified, the frequencies are plotted with one panel for each chromosome. Default is NULL.
layout	Number of rows and columns in plot. Only used in plot by chromosome. Default is c(1,1).
filters	Index or string value indicating which filter to plot. The length of filters is limited to one if the parameter circos is FALSE. Default is the first filter.
circos	A logical value indicating whether to return a circos plot. If TRUE, it returns a circos plot that can display and compare multiple filters. Default is FALSE.
assembly	A string specifying the genome assembly version to apply to CNV frequency plotting. Allowed options are "hg19" and "hg38". Default is "hg38".

Value

The binned CNV frequency plot

Examples

```
## load necessary data (this step can be skipped in real implementation)
data("hg38_cytoband")
## get frequency data
freq <- pgxLoader(type="cnv_frequency", output ='pgxfreq', filters="NCIT:C3512")
## visualize
pgxFreqplot(freq)
```

---

pgxLoader *Load data from Progenetix database via the Beacon v2 API with some extensions*

Description

This function loads various data from Progenetix database via the Beacon v2 API with some extensions (BeaconPlus). It is also compatible with other Beacon v2-compliant resources.

Usage

```
pgxLoader(
  type = NULL,
  output = NULL,
  biosample_id = NULL,
  individual_id = NULL,
  filters = NULL,
  limit = 0,
  skip = 0,
```

```
  dataset = NULL,
  codematches = FALSE,
  filter_pattern = NULL,
  save_file = FALSE,
  filename = "variants.tsv",
  use_https = TRUE,
  domain = "progenetix.org",
  entry_point = "beacon",
  num_cores = 1
)
```

Arguments

type	A string specifying the type of output data. Available options include: • "individuals": Returns information about individuals. • "biosamples": Returns information about biosamples. • "analyses": Returns information about analyses. • "g_variants": Returns variants data. • "filtering_terms": Returns available filtering terms. • "counts": Returns the count of results based on the specified filters. • "cnv_frequency": Returns precomputed CNV frequency data from Progenetix. • "cnv_fraction": Returns CNV fraction per sample based on Progenetix data.
output	A string specifying the format of the output data. The available options depend on the value of the type parameter: • If type is "g_variants", the available options are NULL (default), "pgxseg", or "seg". • If type is "cnv_frequency", the available options are "pgxfreq" (default) or "pgxmatrix". • If type is "cnv_fraction", the available options are NULL (default) or "pgxmatrix".
biosample_id	Identifiers used in the query database for identifying biosamples.
individual_id	Identifiers used in the query database for identifying individuals.
filters	Identifiers used in public repositories, bio-ontology terms, or custom terms such as c("NCIT:C7376", "pgx:icdom-85003"). When multiple filters are used, they are combined using AND logic when the parameter type is "individuals", "biosamples", or "analyses"; OR logic when the parameter type is "counts" or "cnv_frequency".
limit	Integer to specify the number of returned profiles. Default is 0 (return all).
skip	An integer specifying the number of profiles to skip. For example, if skip = 2 and limit = 500, the first 2 * 500 = 1000 profiles are skipped, and the next 500 profiles are returned. Default is 0, meaning no profiles are skipped.
dataset	Datasets to query from the Beacon response. Default is NULL, which includes results from all datasets.

codematches	A logical value indicating whether to exclude samples from child concepts of the specified filters in the ontology tree. If TRUE, only samples that exactly match the specified filters will be included. This parameter should not be used when filters include ontology-irrelevant filters, such as pubmed or cohort identifiers. Default is FALSE. This option is applicable only when querying data resources are Progenetix or cancercelllines.org.
filter_pattern	Optional string pattern to match against the label field of available filters. Only used when the parameter type is "filtering_terms". Default is NULL, which includes all filters.
save_file	A logical value determining whether to save variant data as a local file instead of direct return. Only used when the parameter type is "g_variants". Default is FALSE.
filename	A string specifying the path and name of the file to be saved. This parameter is used only when save_file is set to TRUE. The default value is "variants.tsv", saved in the current working directory.
use_https	A logical value indicating whether to use the HTTPS protocol. If TRUE, the domain will be prefixed with "https://"; otherwise, "http://" will be used. Default is TRUE.
domain	The domain of the query data resource. Default is "progenetix.org".
entry_point	The entry point of the Beacon v2 API. Default is "beacon", resulting in the default endpoint being "https://progenetix.org/beacon".
num_cores	An integer specifying the number of cores to use for parallel processing during Beacon v2 phenotypic/meta-data queries from multiple domains or variant data queries from multiple biosamples. Default is 1.

Value

Data from Progenetix database and other Beacon v2-compatible resources

Examples

```
## query metadata
biosamples <- pgxLoader(type="biosamples", filters = "NCIT:C3512")
## query variants
seg <- pgxLoader(type="g_variants", biosample_id = "pgxbs-kftvgx4y")
## query CNV frequency
freq <- pgxLoader(type="cnv_frequency", output ='pgxfreq', filters="NCIT:C3512")
```

---

pgxMetaplot *Plot survival data of individuals*

Description

This function provides the survival plot from individual metadata.

Usage

```
pgxMetaplot(data, group_id, condition, return_data = FALSE, ...)
```

Arguments

data	The data frame returned by the pgxLoader function, containing survival data for individuals. The survival state is represented by Experimental Factor Ontology in the "followup_state_id" column, and the survival time is represented in ISO 8601 duration format in the "followup_time" column.
group_id	A string specifying which column is used for grouping in the Kaplan-Meier plot.
condition	A string for splitting individuals into younger and older groups, following the ISO 8601 duration format. Only used if group_id is "age_iso".
return_data	A logical value determining whether to return the metadata used for plotting. Default is FALSE.
...	Other parameters relevant to KM plot. These include pval, pval.coord, pval.method, conf.int, linetype, and palette (see ggsurvplot from survminer)

Value

The KM plot from input data

Examples

```
individuals <- pgxLoader(type="individuals",filters="NCIT:C3512")
pgxMetaplot(individuals, group_id="age_iso", condition="P65Y")
```

---

pgxSegprocess *Extract, analyse and visualize "pgxseg" files*

Description

This function extracts segment variants, CNV frequency, and metadata from local "pgxseg" files and supports survival data visualization.

Usage

```
pgxSegprocess(
  file,
  group_id = "group_id",
  show_KM_plot = FALSE,
  return_metadata = FALSE,
  return_seg = FALSE,
  return_frequency = FALSE,
  assembly = "hg38",
  cnv_column_idx = 6,
  bin_size = 1e+06,
```

```
  overlap = 1000,
  soft_expansion = 0.1,
  ...
)
```

Arguments

file	A string specifying the path and name of the "pgxseg" file where the data is to be read.
group_id	A string specifying which id is used for grouping in KM plot or CNV frequency calculation. Default is "group_id".
show_KM_plot	A logical value determining whether to return the Kaplan-Meier plot based on metadata. Default is FALSE.
return_metadata	A logical value determining whether to return metadata. Default is FALSE.
return_seg	A logical value determining whether to return segment data. Default is FALSE.
return_frequency	A logical value determining whether to return CNV frequency data. The frequency calculation is based on segments in segment data and specified group id in metadata. Default is FALSE.
assembly	A string specifying the genome assembly version to apply to CNV frequency calculation and plotting. Allowed options are "hg19" and "hg38". Default is "hg38".
cnv_column_idx	Index of the column specifying the CNV state used for calculating CNV frequency. The index must be at least 6, with the default set to 6. The CNV states should either contain "DUP" for duplications and "DEL" for deletions, or level-specific CNV states represented using Experimental Factor Ontology (EFO) codes.
bin_size	Size of genomic bins used in CNV frequency calculation to split the genome, in base pairs (bp). Default is 1,000,000.
overlap	Numeric value defining the amount of overlap between bins and segments considered as bin-specific CNV, in base pairs (bp). Default is 1,000.
soft_expansion	Fraction of bin_size to determine merge criteria. During the generation of genomic bins, division starts at the centromere and expands towards the telomeres on both sides. If the size of the last bin is smaller than soft_expansion * bin_size, it will be merged with the previous bin. Default is 0.1.
...	Other parameters relevant to KM plot. These include pval, pval.coord, pval.method, conf.int, linetype, and palette (see ggsurvplot from survminer)

Value

Segments data, CNV frequency object, meta data or KM plots from local "pgxseg" files

Examples

```
file_path <- system.file("extdata", "example.pgxseg",package = 'pgxRpi')
info <- pgxSegprocess(file=file_path,show_KM_plot = TRUE, return_seg = TRUE, return_metadata = TRUE)
```

segtoFreq *Calculate CNV frequency data from given segment data*

Description

Thie function calculates the frequency of deletions and duplications

Usage

```
segtoFreq(
  data,
  cnv_column_idx = 6,
  cohort_name = "unspecified cohort",
  assembly = "hg38",
  bin_size = 1e+06,
  overlap = 1000,
  soft_expansion = 0.1
)
```

Arguments

data	Segment data containing CNV states. The first four columns should represent sample ID, chromosome, start position, and end position, respectively. The fifth column can contain the number of markers or other relevant information. The column representing CNV states (with a column index of 6 or higher) should either contain "DUP" for duplications and "DEL" for deletions, or level-specific CNV states such as "EFO:0030072", "EFO:0030071", "EFO:0020073", and "EFO:0030068", which correspond to high-level duplication, low-level duplication, high-level deletion, and low-level deletion, respectively.
cnv_column_idx	Index of the column specifying the CNV state. Default is 6, based on the "pgxseg" format used in Progenetix. If the input segment data follows the general `.seg` file format, this index may need to be adjusted accordingly.
cohort_name	A string specifying the cohort name. Default is "unspecified cohort".
assembly	A string specifying the genome assembly version for CNV frequency calculation. Allowed options are "hg19" or "hg38". Default is "hg38".
bin_size	Size of genomic bins used to split the genome, in base pairs (bp). Default is 1,000,000.
overlap	Numeric value defining the amount of overlap between bins and segments considered as bin-specific CNV, in base pairs (bp). Default is 1,000.
soft_expansion	Fraction of `bin_size` to determine merge criteria. During the generation of genomic bins, division starts at the centromere and expands towards the telomeres on both sides. If the size of the last bin is smaller than `soft_expansion` * bin_size, it will be merged with the previous bin. Default is 0.1.

Value

The binned CNV frequency stored in "pgxfreq" format

Examples

```
## load necessary data (this step can be skipped in real implementation)
data("hg38_cytoband")
## get pgxseg data
seg <- read.table(system.file("extdata", "example.pgxseg",package = 'pgxRpi'),header=TRUE,sep = "\t")
## calculate frequency data
freq <- segtoFreq(seg)
## visualize
pgxFreqplot(freq)
```

Index