Package 'pgxRpi'

October 11, 2024

Title R wrapper for Progenetix

Version 1.1.7

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 75d23e5

git_last_commit_date 2024-09-11

Repository Bioconductor 3.20

Date/Publication 2024-10-10

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

---

pgxFilter *Query available filters*

Description

This function retrieves available filters in the Progenetix database via the Beacon v2 API.

Usage

```
pgxFilter(
  prefix = NULL,
  return_all_prefix = FALSE,
  domain = "http://progenetix.org",
  entry_point = "beacon"
)
```

Arguments

`prefix`	A string specifying the prefix of filters, such as 'NCIT'. Default is NULL, which means that all available filters will be returned. When specified, it returns all filters with the specified prefix.
`return_all_prefix`	A logical value determining whether to return all valid prefixes of filters used in database. If TRUE, the `prefix` parameter will be ignored. Default is FALSE.
`domain`	A string specifying the domain of the query data resource. Default is "http://progenetix.org".
`entry_point`	A string specifying the entry point of the Beacon v2 API. Default is "beacon", resulting in the endpoint being "http://progenetix.org/beacon".

Value

Filter terms used in the data resource that you query.

Examples

```
pgxFilter(prefix = "NCIT")
```

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,
  highlight = NULL,
  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 columns and rows 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.
highlight	Indices of genomic bins to be highlighted in red.
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).

Usage

```
pgxLoader(
  type = NULL,
  output = NULL,
  biosample_id = NULL,
  individual_id = NULL,
  filters = NULL,
  limit = 0,
  skip = NULL,
  dataset = NULL,
  codematches = FALSE,
  save_file = FALSE,
  filename = "variant",
  num_cores = 1,
  domain = "http://progenetix.org",
  entry_point = "beacon"
)
```

Arguments

`type`	A string specifying output data type. Available options are "biosamples", "individuals", "analyses", "g_variants", "cnv_frequency", "cnv_fraction", and "sample_count". The options "biosamples", "individuals", and "analyses" return corresponding information. "g_variants" returns variants data. The options "cnv_frequency", "cnv_fraction" and "sample_count" are based on data in Progenetix, returning precomputed CNV frequency, CNV fraction per sample, and the count of samples for the given filter, respectively.
`output`	A string specifying output data format. The available options depend on the `type` parameter. When `type` is "g_variants", available options are NULL (default), "pgxseg", or "seg"; When `type` is "cnv_frequency", available options are "pgxfreq" or "pgxmatrix"; when `type` is "cnv_fraction", 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", "PMID:22824167"). When multiple filters are used, they are combined using AND logic when the parameter `type` is "biosamples", "individuals", or "analyses"; OR logic when the parameter `type` is "cnv_frequency" or "sample_count".
`limit`	Integer to specify the number of returned profiles. Default is 0 (return all).

skip	Integer to specify the number of skipped profiles. E.g. if skip = 2, limit=500, the first 2*500 =1000 profiles are skipped and the next 500 profiles are returned. Default is NULL (no skip).
dataset	A string specifying the dataset to query from the Beacon response. Default is NULL, which includes results from all datasets.
codematches	A logical value determining whether to exclude samples from child concepts of specified filters in the ontology tree. If TRUE, only samples exactly matching the specified filters will be included. Do not use this parameter when filters include ontology-irrelevant filters such as PMID and cohort identifiers. Default is FALSE.
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. Only used if the parameter save_file is TRUE. Default is "variants" in current work directory.
num_cores	Integer to specify the number of cores used for the variant query. Only used when the parameter type is "g_variants". Default is 1.
domain	A string specifying the domain of the query data resource. Default is "http://progenetix.org".
entry_point	A string specifying the entry point of the Beacon v2 API. Default is "beacon", resulting in the endpoint being "http://progenetix.org/beacon".

Value

Data from Progenetix database

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 object returned by the pgxLoader function, which includes survival data about individuals.
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",
  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".
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 with CNV states. The first four columns should specify sample ID, chromosome, start position, and end position, respectively. The column representing CNV states should contain either "DUP" for duplications or "DEL" for deletions.
cnv_column_idx	Index of the column specifying CNV state. Default is 6, following the "pgxseg" format used in Progenetix. If the input segment data uses the general .seg file format, it might need to be set differently.
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)
## calculate frequency data
freq <- segtoFreq(seg)
## visualize
pgxFreqplot(freq)
```

Index