Package 'fishhook'

April 18, 2017

Title R Package for performing Gamma-Poisson regression on somatic mutation count data

Version 0.1

Description Package for performing Gamma-Poisson regression on somatic mutation count data with covariates to identify mutational enrichment or depletion in a statistically calibrated fashion.

Depends R (>= 3.1.0),
GenomicRanges (>= 1.18),
gUtils

Imports MASS,
rtracklayer (>= 1.26),
zoo,
ffTrack,
data.table (>= 1.9),
gUtils,
GenomeInfoDb,
S4Vectors,
BiocGenerics,
R6

Suggests parallel

License GPL-2

LazyData true

RoxygenNote 6.0.1

R topics documented:

`aggregate.targets` *aggregate.targets*

Description

Gathers annotated targets across a vector "by" into meta-intervals returned as a GRangesList, and returns the aggregated statistics for these meta intervals by summing coverage and counts, and performing a weighted average of all other meta data fields (except query.id)

Usage

```
aggregate.targets(targets, by = NULL, fields = NULL, rolling = NULL,
  disjoint = TRUE, na.rm = FALSE, FUN = list(), verbose = TRUE)
```

Arguments

`targets`	annotated GRanges of targets with fields $coverage, optional field, $count and additional numeric covariates, or path to .rds file of the same
`by`	character vector with which to split into meta-territories
`fields`	by default all meta data fields of targets EXCEPT reserved field names $coverage, $counts, $query.id
`rolling`	if specified, positive integer specifying how many (genome coordinate) adjacent to aggregate in a rolling fashion

Details

If rolling = TRUE, will return a rolling collapse of the sorted input where "rolling" specifies the number of adjacent intervals that are aggregated in a rolling manner. (only makes sense for tiled target sets)

If by = NULL and targets is a vector of path names, then aggregation will be done "sample wise" on the files, ie each .rds input will be assumed to comprise the same intervals in teh same order and aggregation will be computed coverage-weighted mean of covariates, a sum of coverage and counts, and (if present) a Fisher combined of $p values. Covariates are inferred from the first file in the list.

Value

GRangesList of input targets annotated with new aggregate covariate statistics OR GRanges if rolling is specified

Author(s)

Marcin Imielinski

`annotate.targets` *annotate.targets*

Description

Takes input of GRanges targets, an optional set of "covered" intervals, and an indefinite list of covariates which can be R objects (GRanges, ffTrack, Rle) or file paths to .rds, .bw, .bed files, and an annotated target intervals GRanges with covariates computed for each interval. These target intervals can be further annotated with mutation counts and plugged into a generalized linear regression (or other) model downstream.

Usage

```
annotate.targets(targets, covered = NULL, events = NULL, ...,
  mc.cores = 1, na.rm = TRUE, pad = 0, verbose = TRUE,
  max.slice = 1000, ff.chunk = 1e+06, max.chunk = 1e+11,
  out.path = NULL, covariates = list(), maxPtGene = Inf,
  weightEvents = FALSE)
```

Arguments

`targets`	path to bed or rds containing genomic target regions with optional target name
`covered`	optional path to bed or rds containing granges object containing "covered" genomic regions
`events`	optional path to bed or rds containing ranges corresponding to events (ie mutations etc)
`...`	paths to sequence covariates whose output names will be their argument names, and each consists of a list with $track field corresponding to a GRanges, RleList, ffTrack object (or path to rds containing that object), $type which can have one of three values "numeric", "sequence", "interval". Numeric tracks must have $score field if they are GRanges), and can have a $na.rm logical field describing how to treat NA values (set to na.rm argument by default) Sequence covariates must be ffTrack objects (or paths to ffTrack rds) and require an additional variables $signatures, which will be used as input to fftab, and can have optional logical argument $grep to specify inexact matches (see fftab) Interval covariates must be Granges (or paths to GRanges rds) or paths to bed files
`out.path`	out.path to save variable to
`maxPtGene`	Sets the maximum number of events a patient can contribute per target
`out.path`	out.path to save variable to
`weightEvetns`	If true, will weight events by thier overlap with targets. e.g. if 10 region, that target region will get assigned a score of 0.1 for that event. If false, any overlap will be given a weight of 1.

Details

There are three types of covariates: numeric, sequence, interval. The covariates are computed as follows: numeric covariates: the mean value sequence covarites: fraction of bases satisfying $signature interval covariates: fraction of bases overlapping feature

Value

GRanges of input targets annotated with covariate statistics (+/- constrained to the subranges in optional argument covered)

Author(s)

Marcin Imielinski

`Annotated` *Annotated*

Description

Stores the annotated data from a FishHook object. and allows users to aggregate,manipulate and score that data. This object should be generated by calling FishHook$annotateTargets(). Note that this is where the meat of the computational burden lies. For example, in our test cases, running 8k pts worth of exome seq on 20k genes took 20seconds without covariates and 20sec + ~5min per covariate added.

Usage

```
Annotated
```

Arguments

`targets`	Examples of targets are genes, enhancers, 1kb tiles of the genome that we can then convert into a rolling window. This param must be of class "GRanges".
`events`	Events are the given mutational regions and must be of class "GRanges". Examples of events are mutational data (e.g. C->G) copy number variations and fusion events. Targets are the given regions of the genome to annotate and must be of class "GRanges".
`covered`	This is equivalent to Eligible in the FishHook class. Eligible are the regions of the genome that we feel are fit to score. For example in the case of exome sequencing where not all regions are equally represented, eligible can be a set of regions that meet an arbitrary coverage threshold. Another example of when to use eligibility is in the case of whole genomes, where your targets are 1kb tiles. Regions of the genome you would want to exclude in this case are highly repetative regions such as centromeres, telomeres, and satelite repeates. This param must be of class "GRanges".

covariates	Covariates are genomic covariates that you belive will cause your given type of event (mutations, CNVs, fusions) that are not linked to the process you are investigating (e.g. cancer biology). In the case of cancer biology we are looking for regions that are mutated as part of cancer progression, and regions that are more suceptable to random mutagenesis such as late replicating or non-expressed region (transcription coupled repair) are potential false positives. Includinig covariates for these will reduce thier prominence in the final data. This param must be of type "Cov_Arr" which can be created by wrapping Cov objects in c(). e.g. c(Cov1,Cov2,Cov3).

Format

An object of class `R6ClassGenerator` of length 24.

Value

Annotate Obeject that can be scored & manipulated and aggregated.

Author(s)

Zoran Z. Gajic

`c.Cov` *c.Cov*

Description

Override the c operator for covariates so that when you type: c(Cov1,Cov2,Cov3) it returns a Cov_Arr object that support vector like operation.

Usage

```
## S3 method for class 'Cov'
c(...)
```

Arguments

...	A series of Covariates, note all objects must be of type Cov

Value

Cov_Arr object that can be passed directly into the FishHook object constructor

Author(s)

Zoran Z. Gajic

Cov *Cov*

Description

Stores Covariate for passing to FishHook object. To be packaged in the Cov_Array Class by calling c(Cov1,Cov2,Cov3)

Usage

```
Cov
```

Arguments

`Covariate`	object of type, GRanges, ffTrack, RleList or character. Note that character objects must be paths to files containing one of the other types as a .rds file
`type`	a string indicating the type of Covariate, valid options are: numeric, sequence, interval. See Annotate Targets for more information on Covariate types
`signature`	In the case where a ffTrack object is of type sequence, a signature field is required, see fftab in ffTrack for more information.

Format

An object of class `R6ClassGenerator` of length 24.

Value

Cov object that can be passed to FishHook object constructor

Author(s)

Zoran Z. Gajic

Cov_Arr *Cov_Arr*

Description

Stores Covariates for passing to FishHook object constructor.Standard initialization involves calling c(Cov1,Cov2,Cov3). Cov_Arr serves to mask the underlieing list implemenations of Covariates in the FishHook Object. This class attempts to mimic a vector in terms of subsetting and in the future will add more vector like operations.

Usage

```
Cov_Arr
```

Arguments

...	several Cov objects for packaging.

Format

An object of class `R6ClassGenerator` of length 24.

Value

Cov_Arr object that can be passed directly to the FishHook object constructor

Author(s)

Zoran Z. Gajic

`FishHook` *FishHook*

Description

Stores Events, Targets, Eligible, Covariates.

Usage

```
FishHook
```

Arguments

`targets`	Examples of targets are genes, enhancers, 1kb tiles of the genome that we can then convert into a rolling window. This param must be of class "GRanges".
`events`	Events are the given mutational regions and must be of class "GRanges". Examples of events are mutational data (e.g. C->G) copy number variations and fusion events. Targets are the given regions of the genome to annotate and must be of class "GRanges".
`eligible`	Eligible are the regions of the genome that we feel are fit to score. For example in the case of exome sequencing where not all regions are equally represented, eligible can be a set of regions that meet an arbitrary coverage threshold. Another example of when to use eligibility is in the case of whole genomes, where your targets are 1kb tiles. Regions of the genome you would want to exclude in this case are highly repetative regions such as centromeres, telomeres, and satelite repeates. This param must be of class "GRanges".
`covariates`	Covariates are genomic covariates that you belive will cause your given type of event (mutations, CNVs, fusions) that are not linked to the process you are investigating (e.g. cancer biology). In the case of cancer biology we are looking for regions that are mutated as part of cancer progression, and regions that are more

suceptable to random mutagenesis such as late replicating or non-expressed region (transcription coupled repair) are potential false positives. Includinig covariates for these will reduce thier prominence in the final data. This param must be of type "Cov_Arr" which can be created by wrapping Cov objects in c(). e.g. c(Cov1,Cov2,Cov3).

Format

An object of class `R6ClassGenerator` of length 24.

Value

FishHook object that can be annotated.

Author(s)

Zoran Z. Gajic

`qq_pval` *qq plot given input p values*

Usage

```
qq_pval(obs, highlight = c(), exp = NULL, lwd = 1, bestfit = T,
  col = NULL, col.bg = "black", pch = 18, cex = 1, conf.lines = T,
  max = NULL, max.x = NULL, max.y = NULL, qvalues = NULL,
  label = NULL, plotly = FALSE, annotations = list(), gradient = list(),
  titleText = "", subsample = NA, ...)
```

Arguments

`obs`	vector of pvalues to plot, names of obs can be intepreted as labels
`highlight`	optional arg specifying indices of data points to highlight (ie color red)
`lwd`	integer, optional, specifying thickness of line fit to data
`pch`	integer dot type for scatter plot
`cex`	integer dot size for scatter plot
`conf.lines`	logical, optional, whether to draw 95 percent confidence interval lines around x-y line
`max`	numeric, optional, threshold to max the input p values
`label`	character vector, optional specifying which data points to label (obs vector has to be named, for this to work)
`plotly`	toggles between creating a pdf (FALSE) or an interactive html widget (TRUE)
`annotations`	named list of vectors containing information to present as hover text (html widget), must be in same order as obs input

gradient	named list that contains one vector that color codes points based on value, must bein same order as obs input
titleText	title for plotly (html) graph only
samp	integer, optional specifying how many samples to draw from input data (default NULL)

Author(s)

Marcin Imielinski, Eran Hodis, Zoran Z. Gajic

Score *Score*

Description

Stores the scored targets. Note that this constructors should be called from Annotated$scoreTargets(). Scores can also be plotted on qqplots using included functions. For other params see score.targets()

Usage

```
Score
```

Arguments

annotated	The annotated targets as an output from Annotated$scoreTargets() or the standard score.targets().

Format

An object of class `R6ClassGenerator` of length 24.

Value

Score object that can be plotted/analyzed

Author(s)

Zoran Z. Gajic

`score.targets` *score.targets*

Description

Scores targets based on covariates using gamma-poisson model with coverage as constant

Usage

```
score.targets(targets, covariates = names(values(targets)), model = NULL,
  return.model = FALSE, nb = TRUE, verbose = TRUE, iter = 200,
  subsample = 1e+05, seed = NULL, p.randomized = TRUE,
  classReturn = FALSE)
```

Arguments

`targets`	annotated targets with fields $coverage, optional field, $count and additional numeric covariates

Value

GRanges of scored results

Author(s)

Marcin Imielinski

`[.Annotate` *[.Annotate*

Description

Overrides the "[" operator for the Annotated object. This allows subsetting of the annotated data in Annotated Objects.

Usage

```
## S3 method for class 'Annotate'
obj[range]
```

Arguments

`obj`	This is the Annotated object to be subset
`range`	This is the range of targets to return, like subsetting a vector. e.g. c(1,2,3,4,5)[3:4] == c(3,4)

Value

Annotated object that can manipulated and scored, but cannot be aggregated again.

Author(s)

Zoran Z. Gajic

`[.Cov_Arr` *[.Cov_Arr*

Description

Overrides the subset operator x[] for use with Cov_Arr to allow for vector like subsetting

Usage

```
## S3 method for class 'Cov_Arr'
obj[range]
```

Arguments

`obj`	This is the Cov_Arr to be subset
`range`	This is the range of Covariates to return, like subsetting a vector. e.g. c(1,2,3,4,5)[3:4] == c(3,4)

Value

A new Cov_Arr object that contains only the Covs within the given range

Author(s)

Zoran Z. Gajic

Index