Package 'fishHook'

June 12, 2018

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.

biocViews

Depends R (>= 3.1.0),
GenomicRanges (>= 1.18),
gUtils,
ffTrack,
data.table (>= 1.9),

Imports MASS,
Matrix,
rtracklayer (>= 1.26),
zoo,
GenomeInfoDb,
S4Vectors,
BiocGenerics,
R6,
plotly

Suggests parallel,
testthat,
BSgenome.Hsapiens.UCSC.hg19

License GPL-2

LazyData true

RoxygenNote 6.0.1.9000

R topics documented:

`aggregate.hypotheses` *title*

Description

Gathers annotated hypotheses 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)

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 hypotheses 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.

Usage

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

Arguments

`hypotheses`	annotated GRanges of hypotheses with fields $coverage, optional field, $count and additional numeric covariates, or path to .rds file of the same; path to bed or rds containing genomic target regions with optional target name
`by`	character vector with which to split into meta-territories (default = NULL)
`fields`	by default all meta data fields of hypotheses EXCEPT reserved field names $coverage, $counts, $query.id (default = NULL)
`rolling`	if specified, positive integer specifying how many (genome coordinate) adjacent to aggregate in a rolling fashion; positive integer with which to performa rolling sum / weighted average WITHIN chromosomes of "rolling" ranges" –> return a granges (default = NULL)
`disjoint`	boolean only take disjoint bins of input (default = TRUE)

na.rm	boolean only applicable for sample wise aggregation (i.e. if by = NULL) (default = FALSE)
FUN	list only applies (for now) if by = NULL, this is a named list of functions, where each item named "nm" corresponds to an optional function of how to alternatively aggregate field "nm" per samples, for alternative aggregation of coverage and count. This function is applied at every iteration of loading a new sample and adding to the existing set. It is normally sum [for coverage and count] and coverage weighted mean [for all other covariates]. Alternative coverage / count aggregation functions should have two arguments (val1, val2) and all other alt covariate aggregation functions should have four arguments (val1, cov1, val2, cov2) where val1 is the accumulating vector and val2 is the new vector of values.
verbose	boolean verbose flag (default = TRUE)

Value

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

Author(s)

Marcin Imielinski

`annotate.hypotheses`

title

Description

Takes input of GRanges hypotheses, 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.

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

Usage

```
annotate.hypotheses(hypotheses, covered = NULL, events = NULL,
  mc.cores = 1, na.rm = TRUE, pad = 0, verbose = TRUE,
  max.slice = 10000, ff.chunk = 1e+06, max.chunk = 1e+11,
  out.path = NULL, covariates = list(), idcap = Inf, idcol = NULL,
  weightEvents = FALSE, ...)
```

Arguments

hypotheses	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 (default = NULL)
events	optional path to bed or rds containing ranges corresponding to events (ie mutations etc) (default = NULL)
mc.cores	integer info (default = 1)
na.rm	info (default = TRUE)
pad	info (default = 0)
verbose	boolean verbose flag (default = FALSE)
max.slice	integer Max slice of intervals to evaluate with gr.val (default = 1e3)
ff.chunk	integer Max chunk to evaluate with fftab (default = 1e6)
max.chunk	integer gr.findoverlaps parameter (default = 1e11)
out.path	out.path to save variable to (default = NULL)
covariates	list of lists where each internal list represents a covariate, the internal list can have elements: track, type,signature,name,pad,na.rm = na.rm,field,grep. See Covariate class for descriptions of what each of these elements do. Note that track is equivalent to the 'Covariate' parameter in Covariate
idcap	Sets the maximum number of events a patient can contribute per target (default = Inf)
idcol	string Column where patient ID is stored
...	paths to sequence covariates whose output names will be their argument names, and each consists of a list with (default = FALSE) $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) fftab signature: signatures is a named list that specify what is to be tallied. Each signature (ie list element) consist of an arbitrary length character vector specifying strings to or length 1 character vector to grepl (if grep = TRUE) or a length 1 or 2 numeric vector specifying exact value or interval to match (for numeric data) Every list element of signature will become a metadata column in the output GRanges specifying how many positions in the given interval match the given query Interval covariates must be Granges (or paths to GRanges rds) or paths to bed files
weightEvetns	boolean If TRUE, will weight events by their overlap with hypotheses. 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.

Value

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

Author(s)

Marcin Imielinski

`c.Covariate` *title*

Description

Override the c operator for covariates so that you can merge them like a vector

Usage

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

Arguments

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

Value

Covariate object that can be passed directly into the FishHook object constructor that contains all of the Covariate covariates Passed in the ... param

Author(s)

Zoran Z. Gajic

`Cov` *Cov*

Description

function to initialize Covariates for passing to FishHook object constructor.

Can also be initiated by passing a vector of multiple vectors of equal length, each representing one of the internal variable names You must also include a list containg all of the covariates (Granges, chracters, RLELists, ffTracks)

Covariate 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(name = as.character(NA), data = NULL, pad = 0,
  type = as.character(NA), signature = as.character(NA),
  field = as.character(NA), na.rm = NA, grep = NA)
```

Arguments

name	character vector Contains names of the covariates to be created, this should not include the names of any Cov objects passed
data,	a list of covariate data that can include any of the covariate classes (GRanges, ffTrack, RleList, character)
pad	numeric vector Indicates the width to extend each item in the covarite. e.g. if you have a GRanges covariate with two ranges (5:10) and (20:30) with a pad of 5, These ranges wil become (0:15) and (15:35)
type	character vector Contains the types of each covariate (numeric, interval, sequencing)
signature,	see ffTrack, a vector of signatures for use with ffTrack sequence covariates fftab signature: signatures is a named list that specify what is to be tallied. Each signature (ie list element) consist of an arbitrary length character vector specifying strings to or length 1 character vector to grepl (if grep = TRUE) or a length 1 or 2 numeric vector specifying exact value or interval to match (for numeric data) Every list element of signature will become a metadata column in the output GRanges specifying how many positions in the given interval match the given query
field,	a chracter vector for use with numeric covariates (NA otherwise) the indicates the column containing the values of that covarites. For example, if you have a covariate for replication timing and the timings are in the column 'value', the parameter field should be set to the character 'Value'
na.rm,	logical vector that indicates whether or not to remove nas in the covariates
grep,	a chracter vector of grep for use with sequence covariates of class ffTrack The function fftab is called during the processing of ffTrack sequence covariates grep is used to specify inexact matches (see fftab)

Value

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

Author(s)

Zoran Z. Gajic

Covariate *title*

Description

Stores Covariates for passing to FishHook object constructor.

Can also be initiated by passing a vector of multiple vectors of equal length, each representing one of the internal variable names You must also include a list containg all of the covariates (Granges, chracters, RLELists, ffTracks)

Covariate 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

```
Covariate
```

Arguments

name	character vector Contains names of the covariates to be created, this should not include the names of any Cov objects passed
pad	numeric vector Indicates the width to extend each item in the covarite. e.g. if you have a GRanges covariate with two ranges (5:10) and (20:30) with a pad of 5, These ranges wil become (0:15) and (15:35)
type	character vector Contains the types of each covariate (numeric, interval, sequencing)
signature,	see ffTrack, a vector of signatures for use with ffTrack sequence covariates fftab signature: signatures is a named list that specify what is to be tallied. Each signature (ie list element) consist of an arbitrary length character vector specifying strings to or length 1 character vector to grepl (if grep = TRUE) or a length 1 or 2 numeric vector specifying exact value or interval to match (for numeric data) Every list element of signature will become a metadata column in the output GRanges specifying how many positions in the given interval match the given query
field,	a chracter vector for use with numeric covariates (NA otherwise) the indicates the column containing the values of that covarites. For example, if you have a covariate for replication timing and the timings are in the column 'value', the parameter field should be set to the character 'Value'
na.rm,	logical vector that indicates whether or not to remove nas in the covariates
grep,	a chracter vector of grep for use with sequence covariates of class ffTrack The function fftab is called during the processing of ffTrack sequence covariates grep is used to specify inexact matches (see fftab)
data,	a list of covariate data that can include any of the covariate classes (GRanges, ffTrack, RleList, character)

Format

An object of class `R6ClassGenerator` of length 24.

Value

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

Author(s)

Zoran Z. Gajic

`dflm` *dflm*

Description

Formats lm, glm, or fisher.test outputs into readable data.table

Usage

```
dflm(x, last = FALSE, nm = "")
```

`dim.FishHook` *title*

Description

Overrides the dim function 'dim(FishHook)' for use with FishHook

Usage

```
## S3 method for class 'FishHook'
dim(obj, ...)
```

Arguments

`obj`	FishHook object that is passed to the length function

Value

returns a numeric vector containing the lengths of various FishHook variables in the following order: i : number of hypotheses j : number of events k : number of covariates l : number of eligible regions

Author(s)

Zoran Z. Gajic

events *Sample events*

Description

An object of type 'GRanges' that contains a set of events dervided from the TCGA whole exome sequencing data.

Format

GRanges

Details

Metadata columns: id, inidcates to which sample (patient) the mutational event belongs to. There are a total of 8475 patients and 1985704 total events

FishHook *title*

Description

Stores Events, Hypotheses, Eligible, Covariates.

Stores Events, Hypotheses, Eligible, Covariates.

Usage

```
Fish(hypotheses = NULL, events = NULL, covariates = NULL,
  eligible = NULL, out.path = NULL, use_local_mut_density = FALSE,
  local_mut_density_bin = 1e+06,
  genome = "BSgenome.Hsapiens.UCSC.hg19::Hsapiens", mc.cores = 1,
  na.rm = TRUE, pad = 0, verbose = TRUE, max.slice = 10000,
  ff.chunk = 1e+06, max.chunk = 1e+11, idcol = NULL, idcap = Inf,
  weightEvents = FALSE, nb = TRUE)

FishHook
```

Arguments

hypotheses	Examples of hypotheses are genes, enhancers, or even 1kb tiles of the genome that we can then convert into a rolling/tiled 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 SNVs (e.g. C->G) somatic copy number alterations (SCNAs), fusion events, etc.

`covariates` Covariates are genomic covariates that you belive will cause your given type of event (mutations, CNVs, fusions, case control samples) that are not linked to the process you are investigating (e.g. cancer drivers). In the case of cancer drivers, we are looking for regions that are mutated as part of cancer progression. As such, regions that are more suceptable to random mutagenesis such as late replicating or non-expressed region (transcription coupled repair) could become false positives. Including covariates for these biological processes will reduce thier visible effect in the final data. This param must be of type "Covariate".

`eligible` Eligible regions are the regions of the genome that have enough statistical power to score. For example, in the case of exome sequencing where all regions are not equally represented, eligible can be a set of regions that meet an arbitrary exome coverage threshold. Another example of when to use eligibility is in the case of whole genomes, where your hypotheses 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".

`out.path` A character that will indicate a system path in which to save the results of the analysis.

`use_local_mut_density`
A logical that when true, creates a covariate that will represent the mutational density in the genome, whose bin size will be determined by local_mut_density_bin. This covariate can be used when you have no other covariates as a way to correct for variations in mutational rates along the genome under the assumption that driving mutations will cluster in local regions as opposed to global regions. This is similar to saying, in the town of foo, there is a crime rate of X that we will assume to be the local crime rate If a region in foo have a crime rate Y such that Y »»> X, we can say that region Y has a higher crime rate than we would expect.

`local_mut_density_bin`
A numeric value that will indicate the size of the genomic bins to use if use_local_mut_density = TRUE. Note that this paramter should be a few orders of magnitude greater than the size of your targetls

e.g. if your hypotheses are 1e5 bps long, you may want a local_mut_density_bin of 1e7 or even 1e8

`genome` A character value indicating which build of the human genome to use, by default set to hg19

`mc.cores` A numeric value that indicates the amount of computing cores to use when running fishHook. This will mainly be used during the annotation step of the analysis, or during initial instantiation of the object if use_local_mut_density = T

`na.rm` A logical indicating how you handle NAs in your data, mainly used in fftab and gr.val, see these function documentations for more information

`pad` A numeric indicating how far each covariate range should be extended, see Covariate for more information, not that this will only be used if atleast on of the Covariates have pad = NA

`max.slice` integer Max slice of intervals to evaluate with gr.val (default = 1e3)

`ff.chunk` integer Max chunk to evaluate with fftab (default = 1e6)

`max.chunk` integer gr.findoverlaps parameter (default = 1e11)

`idcol` A character, that indicates the column name containing the patient ids, this is for use in conjunction with idcap. If max patientpergene is specified and and the

column referenced by idcol exists, we will limit the contributions of each patient to each target to idcap. e.g. if Patient A has 3 events in target A and Patient B has 1 event in target A, and idcap is set to 2, with thier ID column specified, target A will have a cournt of 3, 2 coming from patient A and 1 coming from patient B

`idcap` a numeric that indicates the max number of events any given patient can contribute to a given target. for use in conjction with idcol. see idcol for more info.

`weightEvents` a logical that indicates if the events should be weighted by thier overlap with the hypotheses. e.g. if we have a SCNA spanning 0:1000 and a target spanning 500:10000, the overlap of the SCNA and target is 500:1000 which is half of the original width of the SCNA event. thus if weightEvent = T, we will credit a count of 0.5 to the target for this SCNA. This deviates from the expected input for the gamma poisson as the gamma poisson measures whole event counts.

`nb` boolean negative binomial, if false then use poisson

`vebose` A logical indicating whether or not to print information to the console when running FishHook

`hypotheses` Examples of hypotheses are genes, enhancers, or even 1kb tiles of the genome that we can then convert into a rolling/tiled 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 SNVs (e.g. C->G) somatic copy number alterations (SCNAs), fusion events, etc.

`eligible` Eligible regions are the regions of the genome that have enough statistical power to score. For example, in the case of exome sequencing where all regions are not equally represented, eligible can be a set of regions that meet an arbitrary exome coverage threshold. Another example of when to use eligibility is in the case of whole genomes, where your hypotheses 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, case control samples) that are not linked to the process you are investigating (e.g. cancer drivers). In the case of cancer drivers, we are looking for regions that are mutated as part of cancer progression. As such, regions that are more suceptable to random mutagenesis such as late replicating or non-expressed region (transcription coupled repair) could become false positives. Including covariates for these biological processes will reduce thier visible effect in the final data. This param must be of type "Covariate".

`out.path` A character that will indicate a system path in which to save the results of the analysis.

`use_local_mut_density`
A logical that when true, creates a covariate that will represent the mutational density in the genome, whose bin size will be determined by local_mut_density_bin. This covariate can be used when you have no other covariates as a way to correct for variations in mutational rates along the genome under the assumption that driving mutations will cluster in local regions as opposed to global regions. This is similar to saying, in the town of foo, there is a crime rate of X that we will assume to be the local crime rate If a region in foo have a crime rate Y such that Y »»> X, we can say that region Y has a higher crime rate than we would expect.

`local_mut_density_bin`
: A numeric value that will indicate the size of the genomic bins to use if use_local_mut_density = TRUE. Note that this paramter should be a few orders of magnitude greater than the size of your targetls

  e.g. if your hypotheses are 1e5 bps long, you may want a local_mut_density_bin of 1e7 or even 1e8

`genome`
: A character value indicating which build of the human genome to use, by default set to hg19

`mc.cores`
: A numeric value that indicates the amount of computing cores to use when running fishHook. This will mainly be used during the annotation step of the analysis, or during initial instantiation of the object if use_local_mut_density = T

`na.rm`
: A logical indicating how you handle NAs in your data, mainly used in fftab and gr.val, see these function documentations for more information

`pad`
: A numeric indicating how far each covariate range should be extended, see Covariate for more information, not that this will only be used if atleast on of the Covariates have pad = NA

`vebose`
: A logical indicating whether or not to print information to the console when running FishHook

`max.slice`
: integer Max slice of intervals to evaluate with gr.val (default = 1e3)

`ff.chunk`
: integer Max chunk to evaluate with fftab (default = 1e6)

`max.chunk`
: integer gr.findoverlaps parameter (default = 1e11)

`idcol`
: A character, that indicates the column name containing the patient ids, this is for use in conjunction with idcap. If max patientpergene is specified and and the column referenced by idcol exists, we will limit the contributions of each patient to each target to idcap. e.g. if Patient A has 3 events in target A and Patient B has 1 event in target A, and idcap is set to 2, with thier ID column specified, target A will have a cournt of 3, 2 coming from patient A and 1 coming from patient B

`idcap`
: a numeric that indicates the max number of events any given patient can contribute to a given target. for use in conjction with idcol. see idcol for more info.

`weightEvents`
: a logical that indicates if the events should be weighted by thier overlap with the hypotheses. e.g. if we have a SCNA spanning 0:1000 and a target spanning 500:10000, the overlap of the SCNA and target is 500:1000 which is half of the original width of the SCNA event. thus if weightEvent = T, we will credit a count of 0.5 to the target for this SCNA. This deviates from the expected input for the gamma poisson as the gamma poisson measures whole event counts.

`nb`
: boolean negative binomial, if false then use poisson

Format

An object of class `R6ClassGenerator` of length 24.

Value

FishHook object ready for annotation/scoring.

FishHook object ready for annotation/scoring.

Author(s)

Zoran Z. Gajic

Zoran Z. Gajic

`hypotheses` *Sample hypotheses*

Description

An object of type 'GRanges' that contains 19,688 human genes

An object of type 'GRanges' that contains all of the eligible regions of whole exome sequencing. Whole exome sequencing only sequences exonic sequences and thus most of the genome should be disregarded when conducting the analysis. In addition, many exonic regions are not even captured in whole exome sequencing. We define an eligible (covered) region here as a region where 80 have mapping reads. i.e. if we sequence 10 people and only 6 (60 would consider that region uneigible.

Format

`GRanges`

Details

Metadata columns: gene_name, inidcates the name by which this gene is refered to as. e.g. TP53

Metadata columns: score, indicates the percent of samples that have reads mapping to that region.

`length.Covariate` *title*

Description

Overrides the length function 'length(Covariate)' for use with Covariate

Usage

```
## S3 method for class 'Covariate'
length(obj, ...)
```

Arguments

`obj`	Covariate object that is passed to the length function

Value

number of covariates contained in the Covariate object as defined by length(Covariate$data)

Author(s)

Zoran Z. Gajic

length.FishHook *title*

Description

Overrides the length function 'length(FishHook)' for use with FishHook

Usage

```
## S3 method for class 'FishHook'
length(obj, ...)
```

Arguments

obj	FishHook object that is passed to the length function

Value

length of the hypotheses contained in the FishHook object

Author(s)

Zoran Z. Gajic

qqp *qq plot given input p values*

Description

Generates R or Shiny quantile-quantile (Q-Q) plot given (minimally) an observed vector of p values, plotted their -log1)quantiles against corresponding -log10 quantiles of the uniform distribution.

Usage

```
qqp(obs, highlight = c(), exp = NULL, lwd = 1, col = NULL,
  col.bg = "black", pch = 18, cex = 1, conf.lines = TRUE, max = NULL,
  max.x = NULL, bottomrighttext = NULL, max.y = NULL, label = NULL,
  plotly = TRUE, annotations = list(), gradient = list(),
  titleText = "", subsample = NA, key = NULL, ...)
```

Arguments

obs	vector of pvalues to plot, names of obs can be intepreted as labels, alternatively a data.frame / data.table with column $p, in which case the other (non $p) columns of obs are interpreted "annotations" in the html plot
highlight	vector optional arg specifying indices of data points to highlight (i.e. color red) (default = c())
exp	numeric vector, expected distribution. if default (NULL) will plot observed against a uniform distribution Use this if you are expecting a non-uniform distribution. Must be equal in length to obs. (default = NULL)

lwd	integer, optional, specifying thickness of line fit to data (default = 1)
col	a vector of strings (colors) equivalent in length to obs, this is the color that will be used for plotting. This is only if plotly = T (default = NULL)
col.bg	string indicating the color of the background
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
max.x	numeric, max value for the x axis
max.y	numeric, max value for the y axis
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	data.frame, data.table, or named list of vectors containing information to present as hover text (html widget), must be in same order and length 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
subsample	numeric (positive integer), number of points to use for plotting, will be taken randomly from the set of obs -> p values
key	a character that is passed to the plotly function that will link each point to a give value. For example, if key is set to gene_name The ploted points are refered to by thier gene_name. This is useful when integrating with shiny or any other tool that can integrate with plotly plots.

Author(s)

Marcin Imielinski, Eran Hodis, Zoran Z. Gajic

---

replication_timing *Sample replication_timing, GC-content score*

Description

An object of type 'GRanges' that contains information regarind how long each genomic region takes to replicate. This will be used as a covariate in the fishHook model

Format

GRanges

Details

Metadata columns: score, indicates the relative rate of replication timing in this region

`score` *score 1 or more fishHook models*

Description

Scores a set of K (>1) fishHook models defined over <identical> hypothesis sets. Each model k in K represents a background model over a (disjoin) collection of event sets.

In practice, each event set k could represent a different variant types (eg indels, SV, SNVs) that each have a separate fit (captured in model k) with respect to a set of covariates. Each event set k could also represent patient (or patient set) specific background models, e.g. colon cancer vs breast cancer, or a combination of patient set and variant type (e.g. indels in lung adenocarcinoma, SVs in breast cancer).

The goal is of score() is to combine all the input models / data and derive a hypothesis specific p value for the mutational enrichment (or depletion).

Since each input model k has already computed an expected value e_ik for each hypothesis i, we can integrate these models through a second glm which uses this e_ik as an offset, and computes a hypothesis (or hypothesis set) specific intercept. The value of this intercept and associated p value will represent the mutational enrichment (or depletion) of that hypothesis interval (or hypothesis interval set) from all the various input datasets.

Usage

```
score(..., sets = NULL, mc.cores = NULL, iter = 200, verbose = NULL,
  ignore.theta = FALSE)
```

Arguments

`...`	fishHook models with <identical> hypotheses
`sets`	named list of integers indexing the hypotheses in the input models

Value

data.table of hypotheses

Author(s)

Marcin Imielinski

`score.hypotheses` *title*

Description

Scores hypotheses based on covariates using Gamma-Poisson model with coverage as constant

Usage

```
score.hypotheses(hypotheses, covariates = names(values(hypotheses)),
  model = NULL, return.model = FALSE, nb = TRUE, verbose = TRUE,
  iter = 200, subsample = 1e+05, sets = NULL, seed = 42, mc.cores = 1,
  p.randomized = TRUE, classReturn = FALSE)
```

Arguments

hypotheses	annotated hypotheses with fields $coverage, optional field, $count and additional numeric covariates
covariates	chracter vector, indicates which columns of hypotheses contain the covariates
model	fit existing model –> covariates must be present (default = NULL)
return.model	boolean info (default = FALSE)
nb	boolean If TRUE, uses negative binomial; if FALSE then use Poisson
verbose	boolean verbose flag (default = TRUE)
iter	integer info (default = 200)
subsample	interger info (default = 1e5)
seed	integer (default = 42)
p.randomized	boolean Flag info (default = TRUE)
classReturn	boolean Flag info (default = FALSE)

Value

GRanges of scored results

Author(s)

Marcin Imielinski

[.Covariate *title*

Description

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

Usage

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

Arguments

obj	Covariate This is the Covariate to be subset
range	vector 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 Covariate object that contains only the Covs within the given range

Author(s)

Zoran Z. Gajic

`[.FishHook` *title*

Description

Overrides the subset operator x[] for use with FishHook to allow for vector like subsetting, see fishHook demo for examples

Usage

```
## S3 method for class 'FishHook'
obj[i = NULL, j = NULL]
```

Arguments

`obj`	FishHook object This is the FishHookObject to be subset
`i`	vector subset hypotheses
`j`	vector subset covariates

Value

A new FishHook object that contains only the given hypotheses and/or covariates

Author(s)

Zoran Z. Gajic

Index