Package 'SigsPack'

December 25, 2024

Type Package

Title Mutational Signature Estimation for Single Samples

Version 1.21.0

Author Franziska Schumann <franziska.schumann@student.hpi.de>

Maintainer Franziska Schumann <franziska.schumann@student.hpi.de>

Description Single sample estimation of exposure to mutational signatures. Exposures to known mutational signatures are estimated for single samples, based on quadratic programming algorithms. Bootstrapping the input mutational catalogues provides estimations on the stability of these exposures. The effect of the sequence composition of mutational context can be taken into account by normalising the catalogues.

License GPL-3

Encoding UTF-8

Depends R (>= 3.6)

biocViews SomaticMutation, SNP, VariantAnnotation, BiomedicalInformatics, DNASeq

Imports quadprog (>= 1.5-5), methods, Biobase, BSgenome (>= 1.46.0), VariantAnnotation (>= 1.24.5), Biostrings, GenomeInfoDb, GenomicRanges, rtracklayer, SummarizedExperiment, graphics, stats, utils

Suggests IRanges, BSgenome.Hsapiens.UCSC.hg19, BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

BugReports https://github.com/bihealth/SigsPack/issues

URL https://github.com/bihealth/SigsPack

Collate bootstrap_mut_catalogues.R cosmicSigs.R sigProfilerExome.R sigProfiler20190522.R create_mut_catalogues.R decomposeQP.R get_context_freq.R hg19context_freq.R normalize.R plot_bootstrapped_exposure.R signature_exposure.R summarize_exposures.R vcf2mut_cat.R

RoxygenNote 6.1.1

LazyData true

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

git_branch devel

git_last_commit 7943dfb

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-12-24

Contents

`bootstrap_mut_catalogues`

Bootstraps a given mutational catalogue

Description

Bootstraps a given mutational catalogue by replicating samples from the original catalogue's distribution of mutational features. The output can be input to signature_exposure.

Usage

```
bootstrap_mut_catalogues(n, original, m = NULL)
```

Arguments

n	Amount of bootstrapped replicates that are created (they will by default have the same amount of mutations as the original catalogue)
original	Mutational catalogue (matrix) of a sample that is taken as the distribution from which the replicates are sampled
m	Amount of mutations the replicates are supposed to have (e.g. if this differs from the original or if the original is provided as probabilities instead of total counts)

Value

matrix containing the mutational catalogues of the replicates

Examples

```
data(cosmicSigs)
reps <- bootstrap_mut_catalogues(n = 150, original = create_mut_catalogues(
                                  10, 500)[["catalogues"]][,1])
```

---

cosmicSigs *COSMIC Signature Profiles*

Description

COSMIC Signature Profiles

Usage

```
cosmicSigs
```

Format

A matrix with 30 rows and 96 columns containing the signature profiles of the 30 consensus signatures as reported on COSMIC.

Signature N The profile of signature N representing the relative proportions of mutations generated by each signature based on the trinucleotide frequencies of hte reference human genome version GRCh37.#'

Source

https://cancer.sanger.ac.uk/cancergenome/assets/signatures_probabilities.txt

`create_mut_catalogues` *Creates simulated mutational catalogues*

Description

Creates mutational catalogues from chosen mutational signature profiles to gain simualted data sampled from a distirbution with with known signature contribution

Usage

```
create_mut_catalogues(n, m, P = get(utils::data("cosmicSigs", package =
  "SigsPack")), sig_set = NULL, c_exposure = NULL)
```

Arguments

`n`	Amount of mutational catalogues that will be created
`m`	Amount of mutations that each catalogues will have
`P`	Matrix (f x k) containing the signature profiles of k signatures which will be used to create the catalogues k is the amount of signature profiles that P contains f is the amount of features that the profiles are defined on (default: COSMIC signature matrix 96 x 30) (optional)
`sig_set`	Numeric vector containing the index of the columns from P (which are the signature profiles) that will be used to create the catalogues (optional). Defaults to all columns of P.
`c_exposure`	Numeric vector specifying the contribution of each signature to the distribution each sample is drawn from. Samples will have an appr. exposure of c_exposure[idx] to signature sig_set[idx] (if sum(c_exposure)==1). (optional) Default: random exposure.

Value

List containing a matrix (f x n) with the simulated catalogues and a matrix (k x n) detailing the signature exposure of each catalogue

Examples

```
data(cosmicSigs)
sim_data <- create_mut_catalogues(10, 300, cosmicSigs, c(2,6,15,27))
sim_data <- create_mut_catalogues(1000, 500)
sim_data <- create_mut_catalogues(1, 500, sig_set = c(1,4,29), c_exposure = c(0.25, 0.65, 0.1))
```

decomposeQP *decomposeQP function*

Description

This function is taken from the package 'SignatureEstimation' by Xiaoqing Huang and Damian Wojtowicz (source: https://www.ncbi.nlm.nih.gov/CBBresearch/Przytycka/index.cgi#signatureestimation) The function allows to get the optimal solution by using dual method to solve the quadratic programming problem.

Usage

```
decomposeQP(m, P, ...)
```

Arguments

m	observed tumor profile vector for a single patient/sample, 96 by 1. m is normalized.
P	signature profile matrix, 96 by N (N = # signatures, COSMIC: N=30)
...	control parameter that can be passed into the solve.QP function

Value

matrix containing estimated signature exposures

Examples

```
data(cosmicSigs)
mut_cat<- (create_mut_catalogues(1,500)[['catalogues']])/500
decomposeQP(mut_cat, cosmicSigs)
```

get_context_freq *Extract occurence of tri-nucleotide contexts*

Description

Extracts the frequencies of the tri-nucleotide contexts in a given region of the genome. These frequencies are needed to normalize a mutational catalogue. The output can be input to normalize().

Usage

```
get_context_freq(genome, region = NULL)
```

Arguments

genome	a BSgenome object
region	a GRanges object, path, URL, connection or BEDFile object.

Value

matrix containing the frequencies of the trinucleotide contexts

Examples

```
gr<-GenomicRanges::GRanges(seqnames=c("chr1"),
          ranges=IRanges::IRanges(start=c(100000),end=c(1000000)),
          strand=c("+"))
get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19, gr)
get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19)

## Not run:
get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19, 'example.bed')

## End(Not run)
```

---

hg19context_freq *Trinuecleotide frequencies of the human reference genome hg19*

Description

Trinuecleotide frequencies of the human reference genome hg19

Usage

```
hg19context_freq
```

Format

A numeric vector with 32 elements

Trinucleotide The frequency of a certain trinucleotide context in the genome.

Source

BSgenome.Hsapiens.UCSC.hg19

normalize *Normalize mutational catalogues*

Description

Normalizes the catalogues to a target distribution (e.g. to match the distribution of the reference signatures).

Usage

```
normalize(mut_cat, source_context,
  target_context = get(utils::data("hg19context_freq", package =
  "SigsPack")))
```

Arguments

mut_cat	mutational catalogues (96 by n, n being the amounts of catalogues) that will be normalized. The tri-nucleotide contexts are expected to be in the default lexicographical order (see simulated data or cosmicSigs)
source_context	Distribution of tri-nucleotides in the source region.
target_context	Distribution of tri-nucleotides in the target region. Defaults to the context frequencies of BSgenome.Hsapiens.UCSC.hg19 since that corresponds to the COSMIC signatures

Value

mutational catalogues (96 by n, n being the amounts of catalogues) normalized to match the target distribution (context)

Note

The output from get_context_freq() can be used as input to this function

Examples

```
# this is a toy example:
#create mutational catalogue:
sim_data <- create_mut_catalogues(1, 500)[['catalogues']]
# get trinucleotide frequencies for the genome:
genome_context <- get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19)
#get trinucleotide frequencies for a specific region:
gr<-GenomicRanges::GRanges(seqnames=c("chr1"),
          ranges=IRanges::IRanges(start=c(100000),end=c(1000000)),
          strand=c("+"))
region_context<-get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19, gr)
#normalize data:
normalized_mut_cat <- normalize(sim_data, region_context, genome_context)
```

```
## Not run:
# get the tri-nucleotide distribution of an exome region
exome_contexts <- get_context_freq(BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19,
                                   'example_exome.bed')

# normalize the mutational catalogue to match the COSMIC signatures
normalized_mut_cat <- normalize(mut_cat, exome_contexts, hg19context_freq)

## End(Not run)
```

`plot_bootstrapped_exposure` *Plot signature exposure estimation for several samples*

Description

Creates a boxplot illustrating the results of the signature estimation for several mutational catalogues (e.g. bootstrapped re-samples or a cohort). The plot shows the distribution of estimated signature exposure for all the catalogues, highlighting the one of the original mutational catalogue if one is provided.

Usage

```
plot_bootstrapped_exposure(bootstrapped_exposure,
  original_estimation = NULL, title = NULL, box_col = NULL,
  point_col = NULL, sig_names = NULL, sample_names = NULL)
```

Arguments

`bootstrapped_exposure`	matrix (n by k) containing the signature exposure of several mutational catalogues (bootstrapped re-samples) n is the amount of re-samples k is the amount of signature profiles that P contains
`original_estimation`	matrix (n by k) containing the signature exposure of one or several mutational catalogues (typically, the original sample) that will be diplayed on top of the boxplots
`title`	character, title of the plot
`box_col`	color option of the boxplot
`point_col`	color option of the points that indicate the exposure of the original profile(s). Should be a vector, if several original profiles are plotted
`sig_names`	character vector, names of the signatures to be displayed as x-axis labels
`sample_names`	character vector, names of the original profile(s)

Value

Diplays a boxplot

Note

The function can of course also be used to plot the distribution of estimated signature exposures in a cohort instead of one bootstrapped sample.

Examples

```
# prepare input
data(cosmicSigs)
mut_cat <- create_mut_catalogues(4,400)
exposures <- signature_exposure(bootstrap_mut_catalogues(
1000, mut_cat[['catalogues']][,1]))[['exposures']]
original_exposure <- signature_exposure(mut_cat[['catalogues']])[['exposures']]

plot_bootstrapped_exposure(exposures, as.matrix(original_exposure[,1]))

plot_bootstrapped_exposure(exposures, as.matrix(original_exposure),
title='Example', box_col='grey')
```

---

signature_exposure *Estimates the signature exposure of a mutational catalogue*

Description

Estimates the signature exposure of a mutational catalogue by reconstructing it from a chosen set of signatures, by default, the used method is quadratic programming

Usage

```
signature_exposure(mut_cat, P = get(utils::data("cosmicSigs", package =
  "SigsPack")), sig_set = NULL, FUN = decomposeQP, ...)
```

Arguments

mut_cat	matrix (f by n) conatining one or several mutational catalogues of samples whose signature exposures are to be estimated f is the amount of features that the profiles are defined on n is the number of catalogues
P	Matrix (f by k) containing the signature profiles of k signatures whose exposure is to be found k is the amount of signature profiles that P contains f is the amount of features that the profiles are defined on (default: COSMIC signature matrix 96 x 30)
sig_set	Numeric vector containing the index of the columns from the signature matrix (which are the signature profiles) that will be used reconstruct the mutational catalogue.

FUN	Function to estimate the signature exposure. Default: Quadratic programming (P has to be of full rank to use this method)
...	control parameters that will be passed to FUN

Value

List of the estimated signature exposure, the reconstructed profile of the sample, the cosine similarity between the two and the error

Examples

```
data(cosmicSigs)
signature_exposure(create_mut_catalogues(10,500)[['catalogues']])
signature_exposure(create_mut_catalogues(10,500)[['catalogues']], sig_set = c(2,7,16,28,30))
signature_exposure(as.matrix(create_mut_catalogues(10,500)[['catalogues']][,1]))
```

---

sigProfiler20190522 *COSMIC v3 whole genome SBS Signature Profiles*

Description

COSMIC v3 whole genome SBS Signature Profiles

Usage

```
sigProfiler20190522
```

Format

A matrix with 96 rows and 67 columns containing the signature profiles of the 67 consensus signatures as reported on COSMIC version 3.

SBS<N> The profile of signature SBS<N> representing the relative proportions of mutations generated by each signature based on the trinucleotide frequencies of the reference human genome version GRCh37. The signature profiles have been normalised for mutational catalogues collected on WGS data.

Source

https://cancer.sanger.ac.uk/cosmic/signatures/SBS/

sigProfilerExome *COSMIC v3 exome SBS Signature Profiles*

Description

COSMIC v3 exome SBS Signature Profiles

Usage

```
sigProfilerExome
```

Format

A matrix with 96 rows and 67 columns containing the signature profiles of the 67 consensus signatures as reported on COSMIC version 3.

SBS<N> The profile of signature SBS<N> representing the relative proportions of mutations generated by each signature based on the trinucleotide frequencies of the reference human genome version GRCh37. The signature profiles have been normalised for mutational catalogues collected on exome datasets.

Source

https://cancer.sanger.ac.uk/cosmic/signatures/SBS/

summarize_exposures *Signature exposure analysis of a mutational catalogue*

Description

Bootstrapps a mutational catalogue and details the results about the signature estimation in a table and a boxplot

Usage

```
summarize_exposures(mut_cat, P = get(utils::data("cosmicSigs", package =
  "SigsPack")), plotting = TRUE, m = NULL)
```

Arguments

mut_cat	mutational catalogue (96 by 1)
P	Matrix (f x k) containing the signature profiles of k signatures whose exposure is to be found k is the amount of signature profiles that P contains f is the amount of features that the profiles are defined on (default: COSMIC signature matrix 96 by 30)

plotting	boolean, TRUE by default, if True, a boxplot showing the distribution of estimated signature exposure for all the re-samples, highlighting the one of the original mutational catalogue, thus providing insights on the reliability of the estimates
m	numeric, amount of mutations in the profile (needed for bootstrapping in case mut_cat doesn't contain absolute counts.)

Value

table (k by 6) containing statistics for each signature about the estimated exposure over 1000 bootstrapped resamples - original exposure estimated exposure of the original mutational profile - min minimum estimated exposure to this signature - 1. quartile - median - 3. quartile - max maximum estimated exposure to this signature

if the plotting option is chosen, a boxplot will also be displayed

Examples

```
summarize_exposures(create_mut_catalogues(10,500)[['catalogues']][,1], plotting=FALSE)
```

---

vcf2mut_cat *Derive the mutational catalogue from a vcf*

Description

Creates a matrix containing the mutational catalogue from a vcf file or object. The result can be input to the analysis functions of this package.

Usage

```
vcf2mut_cat(vcf, genome, name = NULL, seqs = NULL)
```

Arguments

vcf	*.vcf file or a vcf object containing variant calling data for one patient
genome	a BSgenome object corresponding to the genome the variants were called on
name	optional, a sample name
seqs	optional, a character vector containing the names of the sequences that are to be included in the mutational profile. If none is given everything will we included

Value

mutational catalogue (matrix) of a patient containing SNV absolute counts (in the 96 trinucleotide context) format: 1 by 96

Note

The execution can take some time, depending on the size of the vcf

Examples

```
## Not run:
vcf2mut_cat('test.vcf', BSgenome.Hsapiens.UCSC.hg19::BSgenome.Hsapiens.UCSC.hg19)

## End(Not run)
```

Index