Package 'ActiveDriverWGS'

May 6, 2026

Title A Driver Discovery Tool for Cancer Whole Genomes

Version 1.2.1

Description A method for finding enrichments of somatic single nucleotide variants (SNVs) and small insertions-deletions (Indels) in functional elements in the human genome. 'ActiveDriverWGS' detects coding and noncoding cancer driver elements using whole genome sequencing data. The method is part of the publication H. Zhu et al. (2020) <doi:10.1016/j.molcel.2019.12.027> ``Candidate Cancer Driver Mutations in Distal Regulatory Elements and Long-Range Chromatin Interaction Networks" in Molecular Cell.

Depends R (>= 3.5)

Imports BSgenome, Biostrings, GenomeInfoDb, GenomicRanges, IRanges, S4Vectors

License GPL-3

Encoding UTF-8

LazyData true

Suggests BSgenome.Hsapiens.UCSC.hg19, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Mmusculus.UCSC.mm9, BSgenome.Mmusculus.UCSC.mm10, knitr, testthat, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Juri Reimand [aut, cre], Helen Zhu [ctb], Kevin Cheng [ctb]

Maintainer Juri Reimand <juri.reimand@utoronto.ca>

Repository CRAN

Date/Publication 2026-03-13 13:00:08 UTC

Contents

`.fix_all_results` *fix_all_results verifies that the results table has the correct format and p-values*

Description

fix_all_results verifies that the results table has the correct format and p-values

Usage

```
.fix_all_results(all_results)
```

Arguments

`all_results` a data frame containing the following columns

id A string identifying the element of interest

pp_element The p-value of the element

element_muts_obs The number of patients with a mutation in the element

element_muts_exp The expected number of patients with a mutation in the element with respect to background

element_enriched A boolean indicating whether the element is enriched in mutations

pp_site The p-value of the element

site_muts_obs The number of patients with a mutation in the site

site_muts_exp The expected number of patients with a mutation in the site with respect to element

site_enriched A boolean indicating whether the site is enriched in mutations

result_number A numeric indicator denoting the order in which the results were calculated

Value

the same data frame

`.get_3n_context_of_mutations` *This function finds the tri-nucleotide context of mutations*

Description

This function finds the tri-nucleotide context of mutations

Usage

```
.get_3n_context_of_mutations(mutations, this_genome)
```

Arguments

`mutations`	A data frame with the following columns: chr, pos1, pos2, ref, alt, patient **chr** autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY **pos1** the start position of the mutation in base 1 coordinates **pos2** the end position of the mutation in base 1 coordinates **ref** the reference allele as a string containing the bases A, T, C or G **alt** the alternate allele as a string containing the bases A, T, C or G **patient** the patient identifier as a string
`this_genome`	The reference genome object of BSgenome, for example BSgenome.Hsapiens.UCSC.hg19::Hsapiens

Value

A data frame consisting of the same columns as the original mutations data frame and sorted by SNVs and Indels with an additional column `tag` which indicates the trinucleotide context of the mutation

---

`.get_obs_exp` *Calculates the number of expected mutations based*

Description

Calculates the number of expected mutations based

Usage

```
.get_obs_exp(hyp, select_positions, dfr, colname)
```

Arguments

hyp	hypothesis to be tested
select_positions	boolean column which indicates which positions are in the element of interest
dfr	a dataframe containing the data to be tested
colname	name of the column which indicates the count of mutations in the positions of interest

Value

a list of observed mutations and expected mutations

.get_signf_results *Returns significant results*

Description

Returns significant results

Usage

```
.get_signf_results(all_res)
```

Arguments

all_res	a data frame containing the following columns **id** A string identifying the element of interest **pp_element** The p-value of the element **element_muts_obs** The number of patients with a mutation in the element **element_muts_exp** The expected number of patients with a mutation in the element with respect to background **element_enriched** A boolean indicating whether the element is enriched in mutations **pp_site** The p-value of the element **site_muts_obs** The number of patients with a mutation in the site **site_muts_exp** The expected number of patients with a mutation in the site with respect to element **site_enriched** A boolean indicating whether the site is enriched in mutations **result_number** A numeric indicator denoting the order in which the results were calculated

Value

the same data frame with three addition columns

fdr_element The FDR corrected p-value of the element

fdr_site The FDR corrected p-value of the site

has_site_mutations A V indicates the presence of site mutations

---

.make_mut_signatures *Makes mutational signatures*

Description

Makes mutational signatures

Usage

```
.make_mut_signatures()
```

Value

a dataframe with mutational signatures

.split_coord_fragments_in_BED *Splits a BED12 file into separate regions*

Description

Splits a BED12 file into separate regions

Usage

```
.split_coord_fragments_in_BED(i, coords)
```

Arguments

i	The i-th row of the coords data frame which needs to be split into separate elements
coords	The coords data frame which is the imported BED12 file

Value

A data frame containing the following columns for a given BED12 identifier

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the element in base 0 coordinates (BED format)

end the end position of the element in base 0 coordinates (BED format)

id the element identifier - if the element contains multiple segments such as exons, each segment should be a separate row with the segment coordinates and the element identifier as id. Elements can be coding or noncoding such as exons of protein coding genes or active enhancers.

`ActiveDriverWGS` *ActiveDriverWGS is a driver discovery tool for simple somatic mutations in cancer whole genomes*

Description

ActiveDriverWGS is a driver discovery tool for simple somatic mutations in cancer whole genomes

Usage

```
ActiveDriverWGS(
  mutations,
  elements,
  sites = NULL,
  window_size = 50000,
  filter_hyper_MB = 30,
  recovery.dir = NULL,
  mc.cores = 1,
  ref_genome = "hg19",
  detect_depleted_mutations = FALSE
)
```

Arguments

`mutations` A data frame containing the following columns: chr, pos1, pos2, ref, alt, patient.

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

pos1 the start position of the mutation in base 1 coordinates

pos2 the end position of the mutation in base 1 coordinates

ref the reference allele as a string containing the bases A, T, C, G or -

alt the alternate allele as a string containing the bases A, T, C, G or -

patient the patient identifier as a string

`elements` A data frame containing the following columns: chr, start, end, id

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the element in base 0 coordinates (BED format)

end the end position of the element in base 0 coordinates (BED format)

id the element identifier - if the element contains multiple segments such as exons, each segment should be a separate row with the segment coordinates and the element identifier as id. Elements can be coding or noncoding such as exons of protein coding genes or active enhancers.

`sites` A data frame containing the following columns: chr, start, end, id

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the site in base 0 coordinates (BED format)

end the end position of the site in base 0 coordinates (BED format)

id the identifier of the element. id's need to match with those listed in the object elements.

`window_size` An integer indicating the size of the background window in base pairs that is used to establish the expected mutation rate and respective null model. The default is 50000bps

`filter_hyper_MB`

Hyper-mutated samples carry many passenger mutations and dilute the signal of true drivers. Samples with a rate greater than `filter_hyper_MB` mutations per megabase are excluded. The default is 30 mutations per megabase.

`recovery.dir` The directory for storing recovery files. If the directory does not exist, ActiveDriverWGS will create the directory. If the parameter is unspecified, recovery files will not be saved. As an ActiveDriverWGS query for large datasets may be computationally heavy, specifying a recovery directory will recover previously computed results if a query is interrupted.

`mc.cores` The number of cores which can be used if multiple cores are available. The default is 1.

`ref_genome` The reference genome used on the analysis. The default option is "hg19", other options are "hg38", "mm9" and "mm10".

`detect_depleted_mutations`

if TRUE, detect elements with significantly fewer than expected mutations. FALSE by default

Value

A data frame containing the results of driver discovery containing the following columns: id, pp_element, element_muts_obs, element_muts_exp, element_enriched, pp_site, site_muts_obs, site_muts_exp, site_enriched, fdr_element, fdr_site

id A string identifying the element of interest

pp_element The p-value of the element

element_muts_obs The number of patients with a mutation in the element

element_muts_exp The expected number of patients with a mutation in the element with respect to background

element_enriched A boolean indicating whether the element is enriched in mutations

pp_site The p-value of the site

site_muts_obs The number of patients with a mutation in the site

site_muts_exp The expected number of patients with a mutation in the site with respect to element

site_enriched A boolean indicating whether the site is enriched in mutations

fdr_element The FDR corrected p-value of the element

fdr_site The FDR corrected p-value of the site

has_site_mutations A V indicates the presence of site mutations

Examples

```
data(cancer_genes)
data(cll_mutations)

some_genes = c("ATM", "MYD88", "NOTCH1", "SF3B1", "XPO1",
"SOCS1", "CNOT3", "DDX3X", "KMT2A", "HIF1A", "APC")

result = ActiveDriverWGS(mutations = cll_mutations,
elements = cancer_genes[cancer_genes$id %in% some_genes,])
```

ADWGS_test *ADWGS_test executes the statistical test for ActiveDriverWGS*

Description

ADWGS_test executes the statistical test for ActiveDriverWGS

Usage

```
ADWGS_test(
  id,
  gr_element_coords,
  gr_site_coords,
  gr_maf,
  win_size,
  this_genome,
  detect_depleted_mutations = FALSE
)
```

Arguments

`id`	A string used to identify the element of interest. `id` corresponds to an element in the id column of the elements file
`gr_element_coords`	A GenomicRanges object that describes the elements of interest containing the chromosome, start and end coordinates, and an mcols column corresponding to id

`gr_site_coords` A GenomicRanges object that describes the sites of interest which reside in the elements of interest containing the chromosome, start and end coordinates, and an mcols column corresponding to id. Examples of sites include transcription factor binding sites in promoter regions or phosphosites in exons of protein coding genes. An empty GenomicRanges object nullifies the requirement for sites to exist.

`gr_maf` A GenomicRanges object that describes the mutations in the dataset containing the chromosome, start and end coordinates, patient id, and trinucleotide context

`win_size` An integer indicating the size of the background window in base pairs that is used to establish the expected mutation rate and respective null model. The default is 50000bps

`this_genome` The reference genome object of BSgenome, for example BSgenome.Hsapiens.UCSC.hg19::Hsapiens

`detect_depleted_mutations`
if TRUE, detect elements with significantly fewer than expected mutations. FALSE by default

Value

A data frame containing the following columns

id A string identifying the element of interest

pp_element The p-value of the element

element_muts_obs The number of patients with a mutation in the element

element_muts_exp The expected number of patients with a mutation in the element with respect to background

element_enriched A boolean indicating whether the element is enriched in mutations

pp_site The p-value of the site

site_muts_obs The number of patients with a mutation in the site

site_muts_exp The expected number of patients with a mutation in the site with respect to element

site_enriched A boolean indicating whether the site is enriched in mutations

result_number A numeric indicator denoting the order in which the results were calculated

fdr_element The FDR corrected p-value of the element

fdr_site The FDR corrected p-value of the site

has_site_mutations A V indicates the presence of site mutations

Examples

```
library(GenomicRanges)

# Regions
data(cancer_genes)
gr_element_coords = GRanges(seqnames = cancer_genes$chr,
IRanges(start = cancer_genes$start, end = cancer_genes$end),
mcols = cancer_genes$id)
```

```
# Sites (NULL)
gr_site_coords = GRanges(c(seqnames=NULL,ranges=NULL,strand=NULL))

# Reference genome
this_genome = BSgenome.Hsapiens.UCSC.hg19::Hsapiens

# Mutations
data(cll_mutations)
cll_mutations = format_muts(cll_mutations, this_genome = this_genome)

gr_maf = GRanges(cll_mutations$chr,
IRanges(cll_mutations$pos1, cll_mutations$pos2),
mcols=cll_mutations[,c("patient", "tag")])

# ADWGS_test
id = "ATM"
result = ADWGS_test(id, gr_element_coords, gr_site_coords, gr_maf,
win_size = 50000, this_genome = this_genome)
```

`cancer_genes` *cancer_genes*

Description

protein coding genes from gencode v.19, cancer genes adapted from the Cancer Gene Census (November, 2018). Genes affected solely by amplifications, deletions and translations were removed.

Usage

```
data(cancer_genes)
```

Format

A data frame containing the following columns: chr, start, end, id

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the element in base 0 coordinates (BED format)

end the end position of the element in base 0 coordinates (BED format)

id the element identifier - if the element contains multiple segments such as exons, each segment should be a separate row with the segment coordinates and the element identifier as id. Elements can be coding or noncoding such as exons of protein coding genes or active enhancers.

Source

GENCODE

References

Harrow, Jennifer, et al. "GENCODE: the reference human genome annotation for The ENCODE Project." Genome research 22.9 (2012): 1760-1774. (PubMed)

Examples

```
data(cancer_genes)

data(cll_mutations)
ActiveDriverWGS(mutations = cll_mutations, elements = cancer_genes)
```

---

cancer_gene_sites *post-translational modification sites found in cancer genes*

Description

post-translational modification sites found in cancer genes

Usage

```
data(cancer_gene_sites)
```

Format

A data frame containing the following columns: chr, start, end, id

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the site in base 0 coordinates (BED format)

end the end position of the site in base 0 coordinates (BED format)

id the site identifier - each site should contain only 1 segment and a unique id. If ids are duplicated, each segment of the site will be treated as an individual site. Sites can be coding or noncoding such as phosphosites of protein coding genes in genomic coordinates or transcription factor binding sites of active enhancers.

Source

PubMed

References

Wadi, Lina, et al. "ActiveDriverDB: human disease mutations and genome variation in post-translational modification sites of proteins." Nucleic Acids Res. (2018): Jan 4;46(D1):D901-D910. (PubMed)

Examples

```
data(cancer_gene_sites)

data(cll_mutations)
data(cancer_genes)
ActiveDriverWGS(mutations = cll_mutations, elements = cancer_genes, sites = cancer_gene_sites)
```

`cll_mutations` *CLL mutations*

Description

CLL whole genome simple somatic mutations from Alexandrov et, 2013

Usage

```
data(cll_mutations)
```

Format

A data frame containing the following columns: chr, pos1, pos2, ref, alt, patient.

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

pos1 the start position of the mutation in base 1 coordinates

pos2 the end position of the mutation in base 1 coordinates

ref the reference allele as a string containing the bases A, T, C or G

alt the alternate allele as a string containing the bases A, T, C or G

patient the patient identifier as a string

Source

Publication

References

Alexandrov, Ludmil B., et al. "Signatures of mutational processes in human cancer." Nature 500.7463 (2013): 415. (PubMed)

Examples

```
data(cll_mutations)

data(cancer_genes)
ActiveDriverWGS(mutations = cll_mutations, elements = cancer_genes)
```

format_muts *This function filters hypermutated samples and returns the formatted mutations with the appropriate trinucleotide context*

Description

This function filters hypermutated samples and returns the formatted mutations with the appropriate trinucleotide context

Usage

```
format_muts(mutations, this_genome, filter_hyper_MB = NA)
```

Arguments

mutations	A data frame with the following columns: chr, pos1, pos2, ref, alt, patient
	chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY
	pos1 the start position of the mutation in base 1 coordinates
	pos2 the end position of the mutation in base 1 coordinates
	ref the reference allele as a string containing the bases A, T, C or G
	alt the alternate allele as a string containing the bases A, T, C or G
	patient the patient identifier as a string
this_genome	The reference genome object of BSgenome
filter_hyper_MB	The number of mutations per megabase for which a sample is considered hypermutated. Hypermutated samples will be removed in further analyses.

Value

a data frame called mutations which has been formatted with an extra column for trinucleotide context

Examples

```
data(cll_mutations)
this_genome = BSgenome.Hsapiens.UCSC.hg19::Hsapiens
formatted_mutations = format_muts(cll_mutations[1:10,],
filter_hyper_MB = 30, this_genome = this_genome)
```

prepare_elements_from_BED12

Prepares element coords from a BED12 file

Description

Prepares element coords from a BED12 file

Usage

```
prepare_elements_from_BED12(fname)
```

Arguments

fname	The file name of a BED12 file containing the desired elements. For further documentation on the BED12 format, refer to the UCSC website.

Value

A data frame containing the following columns to be used as the input element coords to ActiveDriverWGS

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the element in base 0 coordinates (BED format)

end the end position of the element in base 0 coordinates (BED format)

id the element identifier - if the element contains multiple segments such as exons, each segment should be a separate row with the segment coordinates and the element identifier as id. Elements can be coding or noncoding such as exons of protein coding genes or active enhancers.

Examples

```
elements = prepare_elements_from_BED12(system.file("extdata",
"chr17.coding_regions.bed",
package = "ActiveDriverWGS",
mustWork = TRUE))
```

prepare_elements_from_BED4

Prepares element coords from a BED4 file

Description

Prepares element coords from a BED4 file

Usage

```
prepare_elements_from_BED4(fname)
```

Arguments

fname	The file name of a BED4 file containing the desired elements. For further documentation on the BED4 format, refer to the UCSC website.

Value

A data frame containing the following columns to be used as the input element coords to ActiveDriverWGS

chr autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

start the start position of the element in base 0 coordinates (BED format)

end the end position of the element in base 0 coordinates (BED format)

id the element identifier - if the element contains multiple segments such as exons, each segment should be a separate row with the segment coordinates and the element identifier as id. Elements can be coding or noncoding such as exons of protein coding genes or active enhancers.

Examples

```
elements = prepare_elements_from_BED4(system.file("extdata",
"mini.ptm.bed",
package = "ActiveDriverWGS",
mustWork = TRUE))
```

Index