

# Package ‘ActiveDriverWGS’

October 12, 2022

**Title** A Driver Discovery Tool for Cancer Whole Genomes

**Version** 1.2.0

**Description** A method for finding an enrichment of cancer simple somatic mutations (SNVs and Indels) in functional elements across the human genome. 'ActiveDriverWGS' detects coding and noncoding driver elements using whole genome sequencing data. The method is part of the following publication: Candidate Cancer Driver Mutations in Distal Regulatory Elements and Long-Range Chromatin Interaction Networks. Molecular Cell (2020) <doi:10.1016/j.molcel.2019.12.027>.

**Depends** R (>= 3.5)

**Imports** BSgenome, BSgenome.Hsapiens.UCSC.hg19,  
BSgenome.Hsapiens.UCSC.hg38, BSgenome.Mmusculus.UCSC.mm9,  
BSgenome.Mmusculus.UCSC.mm10, Biostrings, GenomeInfoDb,  
GenomicRanges, IRanges, S4Vectors

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, testthat, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**NeedsCompilation** no

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

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

**Repository** CRAN

**Date/Publication** 2022-09-03 17:00:08 UTC

## R topics documented:

.fix_all_results . . . . .	2
.get_3n_context_of_mutations . . . . .	3

<code>.get_obs_exp</code>	3
<code>.get_signf_results</code>	4
<code>.make_mut_signatures</code>	5
<code>.split_coord_fragments_in_BED</code>	5
<code>ActiveDriverWGS</code>	6
<code>ADWGS_test</code>	8
<code>cancer_genes</code>	10
<code>cancer_gene_sites</code>	11
<code>cll_mutations</code>	12
<code>format_muts</code>	13
<code>prepare_elements_from_BED12</code>	14
<code>prepare_elements_from_BED4</code>	14

## Index 16

---

<code>.fix_all_results</code>	<i>fix_all_results verifies that the results table has the correct format and p-values</i>
-------------------------------	--

---

### 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 <b>chr</b> autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY <b>pos1</b> the start position of the mutation in base 1 coordinates <b>pos2</b> the end position of the mutation in base 1 coordinates <b>ref</b> the reference allele as a string containing the bases A, T, C or G <b>alt</b> the alternate allele as a string containing the bases A, T, C or G <b>patient</b> 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**

<code>hyp</code>	hypothesis to be tested
<code>select_positions</code>	boolean column which indicates which positions are in the element of interest
<code>dfr</code>	a dataframe containing the data to be tested
<code>colname</code>	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**

<code>all_res</code>	a data frame containing the following columns
<b>id</b>	A string identifying the element of interest
<b>pp_element</b>	The p-value of the element
<b>element_muts_obs</b>	The number of patients with a mutation in the element
<b>element_muts_exp</b>	The expected number of patients with a mutation in the element with respect to background
<b>element_enriched</b>	A boolean indicating whether the element is enriched in mutations
<b>pp_site</b>	The p-value of the element
<b>site_muts_obs</b>	The number of patients with a mutation in the site
<b>site_muts_exp</b>	The expected number of patients with a mutation in the site with respect to element
<b>site_enriched</b>	A boolean indicating whether the site is enriched in mutations
<b>result_number</b>	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	<i>ActiveDriverWGS is a driver discovery tool for simple somatic mutations in cancer whole genomes</i>
-----------------	--

---

**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. <b>chr</b> autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY <b>pos1</b> the start position of the mutation in base 1 coordinates <b>pos2</b> the end position of the mutation in base 1 coordinates <b>ref</b> the reference allele as a string containing the bases A, T, C, G or - <b>alt</b> the alternate allele as a string containing the bases A, T, C, G or - <b>patient</b> the patient identifier as a string
elements	A data frame containing the following columns: chr, start, end, id <b>chr</b> autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY

	<b>start</b> the start position of the element in base 0 coordinates (BED format)
	<b>end</b> the end position of the element in base 0 coordinates (BED format)
	<b>id</b> 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
	<b>chr</b> autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY
	<b>start</b> the start position of the site in base 0 coordinates (BED format)
	<b>end</b> the end position of the site in base 0 coordinates (BED format)
	<b>id</b> 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(c11_mutations)

some_genes = c("ATM", "MYD88", "NOTCH1", "SF3B1", "XP01",
"SOCS1", "CNOT3", "DDX3X", "KMT2A", "HIF1A", "APC")

result = ActiveDriverWGS(mutations = c11_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



<code>gr_element_coords</code>	A GenomicRanges object that describes the elements of interest containing the chromosome, start and end coordinates, and an <code>mcols</code> column corresponding to <code>id</code>
<code>gr_site_coords</code>	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 <code>mcols</code> column corresponding to <code>id</code> . 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.
<code>gr_maf</code>	A GenomicRanges object that describes the mutations in the dataset containing the chromosome, start and end coordinates, patient <code>id</code> , and trinucleotide context
<code>win_size</code>	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
<code>this_genome</code>	The reference genome object of BSgenome, for example BSgenome.Hsapiens.UCSC.hg19::Hsapiens
<code>detect_depleted_mutations</code>	if TRUE, detect elements with significantly fewer than expected mutations. FALSE by default

### Value

A data frame containing the following columns

<b><code>id</code></b>	A string identifying the element of interest
<b><code>pp_element</code></b>	The p-value of the element
<b><code>element_muts_obs</code></b>	The number of patients with a mutation in the element
<b><code>element_muts_exp</code></b>	The expected number of patients with a mutation in the element with respect to background
<b><code>element_enriched</code></b>	A boolean indicating whether the element is enriched in mutations
<b><code>pp_site</code></b>	The p-value of the site
<b><code>site_muts_obs</code></b>	The number of patients with a mutation in the site
<b><code>site_muts_exp</code></b>	The expected number of patients with a mutation in the site with respect to element
<b><code>site_enriched</code></b>	A boolean indicating whether the site is enriched in mutations
<b><code>result_number</code></b>	A numeric indicator denoting the order in which the results were calculated
<b><code>fdr_element</code></b>	The FDR corrected p-value of the element
<b><code>fdr_site</code></b>	The FDR corrected p-value of the site
<b><code>has_site_mutations</code></b>	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=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(c11_mutations)
ActiveDriverWGS(mutations = c11_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(c11_mutations)
data(cancer_genes)
ActiveDriverWGS(mutations = c11_mutations, elements = cancer_genes, sites = cancer_gene_sites)
```

---

c11_mutations	<i>CLL mutations</i>
---------------	----------------------

---

## Description

CLL whole genome simple somatic mutations from Alexandrov et, 2013

## Usage

```
data(c11_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(c11_mutations)

data(cancer_genes)
ActiveDriverWGS(mutations = c11_mutations, elements = cancer_genes)
```

---

format_muts	<i>This function filters hypermutated samples and returns the formatted mutations with the appropriate trinucleotide context</i>
-------------	--

---

## 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 <b>chr</b> autosomal chromosomes as chr1 to chr22 and sex chromosomes as chrX and chrY <b>pos1</b> the start position of the mutation in base 1 coordinates <b>pos2</b> the end position of the mutation in base 1 coordinates <b>ref</b> the reference allele as a string containing the bases A, T, C or G <b>alt</b> the alternate allele as a string containing the bases A, T, C or G <b>patient</b> 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(c1l_mutations)
this_genome = BSgenome.Hsapiens.UCSC.hg19::Hsapiens
formatted_mutations = format_muts(c1l_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

## \* datasets

- cancer\_gene\_sites, [11](#)
- cancer\_genes, [10](#)
- c1l\_mutations, [12](#)
- .fix\_all\_results, [2](#)
- .get\_3n\_context\_of\_mutations, [3](#)
- .get\_obs\_exp, [3](#)
- .get\_signf\_results, [4](#)
- .make\_mut\_signatures, [5](#)
- .split\_coord\_fragments\_in\_BED, [5](#)

ActiveDriverWGS, [6](#)

ADWGS\_test, [8](#)

  

- cancer\_gene\_sites, [11](#)
- cancer\_genes, [10](#)
- c1l\_mutations, [12](#)

format\_muts, [13](#)

  

prepare\_elements\_from\_BED12, [14](#)

prepare\_elements\_from\_BED4, [14](#)