Package 'hrdtools'

September 27, 2016

Title Homologous Recombination Deficiency Tests with Whole Genome Sequencing

Version 0.0.0.9000

Description Implementation of various tests of homologous recombination deficiency in a single convenient package.

Depends R (>= 3.3.0)

License

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

R topics documented:

count_lst . . . 2
crosses_centromere . . . 2
filter_short_contigs . . . 2
filter_worker . . . 3
get_centromere_regions . . . 3
get_contigs . . . 3
get_reference_signatures . . . 4
get_subtelomere_coordinate . . . 4
get_subtelomere_regions . . . 4
import_ranges . . . 5
import_snvs . . . 5
is_contiguous . . . 5
loh_test . . . 6
lst_test . . . 6
mutation_catalog . . . 6
nnls_exposures . . . 7
nnls_exposures_worker . . . 7
nnls_montecarlo . . . 8
run_snv . . . 8
run_test . . . 9
stitch . . . 9
tai_test . . . 9
test_multiple . . . 10

Index **11**

count_lst *LST Counter*

Description

This function counts the number of LST sites

Usage

```
count_lst(contigs, contigSizeThreshold)
```

Arguments

contigs	From get_contigs() function

crosses_centromere *Does it Cross the Centromere?*

Description

This function checks whether a region crosses the centromere

Usage

```
crosses_centromere(chr, start, end, centromeres)
```

Arguments

chr	The chromosome that the region belongs to
start	The start position of the region
end	The end position of the region
centromeres	Centromere positions from get_centromere_regions()

filter_short_contigs *Filter Short Contigs*

Description

Removes short contigs from the provided contigs using specified filters

Usage

```
filter_short_contigs(contigs, filterSizeThreshold)
```

Arguments

contigs	From get_contigs() function
filterSizeThreshold	Threshold value, below which segments are filtered out

filter_worker *Filter Worker*

Description

Does the heavy lifting of filtering

Usage

```
filter_worker(contigs, filterSizeThreshold)
```

Arguments

contigs	From get_contigs() function
filterSizeThreshold	Threshold value, below which segments are filtered out

get_centromere_regions *Get Centromere regions*

Description

This function returns centromere regions as a table

Usage

```
get_centromere_regions()
```

get_contigs *Get Contigs from Ranges*

Description

This function processes a GRanges object and returns a data frame of regions, all of which are contiguous. It does so by filling in all the regions not included in the ranges, by assuming these unreported regions are all heterozygous.

Usage

```
get_contigs(gr)
```

Arguments

gr	GRanges object, obtained from import_ranges().

get_reference_signatures

Retrieving Reference Mutation Signatures

Description

This function imports reference mutation signatures to serve as a comparison point when performing non-negative least squares decomposition.

Usage

```
get_reference_signatures(path = NULL)
```

Arguments

path	String indicating path to reference signatures table file. Defaults to bundled 30-signatures.

get_subtelomere_coordinate

Get Subtelomere Coordinate

Description

Convenience function that retrieves the start or end coordinate of a specific subtelomere

Usage

```
get_subtelomere_coordinate(chr, subtelomere, startOrEnd)
```

Arguments

chr	The chromosome of interest
subtelomere	A subtelomeres object from get_subtelomere_regions()
startOrEnd	Takes on value either "start" or "end" depending on which position to return

get_subtelomere_regions

Get Subtelomere data

Description

This function returns subtelomere regions as a table.

Usage

```
get_subtelomere_regions()
```

`import_ranges` *Imports ranges output from APOLLOH*

Description

This function returns a GRanges object based on the data contained in a TSV. This TSV can be output from APOLLOH or similar program.

Usage

```
import_ranges(path, genomeVersion = "hg19")
```

`import_snvs` *SNV Importer This function imports SNVs from a tab-delimited file. The file must be formatted in at least 4 tab-separated columns. The first four columns of the file must be chromosome number (without 'chr'), position, reference base, mutant base.*

Description

SNV Importer This function imports SNVs from a tab-delimited file. The file must be formatted in at least 4 tab-separated columns. The first four columns of the file must be chromosome number (without 'chr'), position, reference base, mutant base.

Usage

```
import_snvs(path, genomeVersion = "hg19")
```

Arguments

`path`	Path to the tab-delimited SNV data file

`is_contiguous` *Contiguity test*

Description

Returns TRUE if set of ranges provided is contiguous (has no gaps)

Usage

```
is_contiguous(contigs)
```

Arguments

`contigs`	From get_contigs() function

loh_test *The HRD-LOH Test*

Description

This function runs the Telomeric Allelic Imbalance test (HRD-LOH).

Usage

```
loh_test(gr)
```

Arguments

gr	GRanges object obtained from import_ranges()

lst_test *The HRD-LST Test*

Description

This function carries out the large scale transition (HRD-LST) test on a GRanges object.

Usage

```
lst_test(gr)
```

Arguments

gr	GRanges object, obtained from import_ranges().

mutation_catalog *Obtain the Mutational Catalog / Spectrum*

Description

This function returns the mutational catalogue / mutational spectrum of a cancer genome.

Usage

```
mutation_catalog(vr)
```

Arguments

vr	VRanges file, obtained from import_snvs()

`nnls_exposures` *Calculate signature exposures*

Usage

```
nnls_exposures(subjectMotifs, refSignature, fractions = FALSE,
  montecarlo = FALSE, iterations = 2000)
```

Arguments

`subjectMotifs`	The mutation catalog of a subject derived from mutation_catalog()
`refSignature`	The reference signature table, which can be imported by get_reference_signatures()
`fractions`	If TRUE, will return results as a fraction of all mutations rather than number of mutations.
`montecarlo`	If TRUE, will perform Monte Carlo simulation to obtain 95
	\itemiterationsThe number of iterations of Monte Carlo to run
	This function determines the signature exposures using non-negative least squares.

`nnls_exposures_worker` *NNLS Worker Function*

Description

Performs the NNLS calculation itself.

Usage

```
nnls_exposures_worker(subjectMotifs, refSignature, fractions)
```

Arguments

`subjectMotifs`	Mutation Catalog from get_mutation_catalog()
`refSignature`	Reference signatures from get_reference_signatures()
`fractions`	If TRUE, will return exposures as a fraction of total mutation burden

nnls_montecarlo *Monte Carlo Simulation of NNLS*

Description

Performs Monte Carlo simulation to obtain NNLS exposure confidence intervals

Usage

```
nnls_montecarlo(subjectMotifs, refSignature, fractions, iterations,
  alpha = 0.05)
```

Arguments

subjectMotifs	Mutation Catalog from get_mutation_catalog()
refSignature	Reference signatures from get_reference_signatures()
fractions	If TRUE, will return exposures as a fraction of total mutation burden
iterations	Number of iterations to run
alpha	Confidence level to retrieve confidence intervals at (default 0.05)

run_snv *Deciphers SNV Signatures*

Description

This function loads an SNV data file and runs the mutation signature deciphering process on it. It returns the output of nnls_exposures.

Usage

```
run_snv(snv_file, genome = "hg19", silent = FALSE)
```

Arguments

snv_file	Path to an SNV file, suitable for import_snvs()
genome	ID of the genome being used (default: 'hg19')
silent	If TRUE, does not print exposures output (default: 'FALSE')

run_test *Runs the LOH, TAI, and LST Tests*

Description

This function loads an LOH segments file and performs the three HRD tests on it.

Usage

```
run_test(loh_file, genome = "hg19", silent = FALSE)
```

Arguments

loh_file	Path to an LOH file suitable for import_ranges()
genome	ID of the genome being used (default: 'hg19')
silent	If TRUE, does not print HRD scores, just returns them (default: 'FALSE')

stitch *Combine Contigs*

Description

This function combines neighbouring contigs based on a few set criteria.

Usage

```
stitch(contigs)
```

Arguments

contigs	Contigs output from get_contigs().

tai_test *The HRD-TAI Test*

Description

This function runs the Telomeric Allelic Imbalance test (HRD-TAI).

Usage

```
tai_test(gr)
```

Arguments

gr	GRanges object obtained from import_ranges()

`test_multiple` *Run the HRD Tests on Multiple Files*

Description

This is a convenience function to run_test() on multiple files.

Usage

```
test_multiple(loh_files, output_file, genome = "hg19", multicore = TRUE)
```

Arguments

`output_file`	Path to an output file where a table of the results will be stored.
`genome`	ID of the genome being used (default: 'hg19')
`snv_file`	Character vector of paths to SNV files
`silent`	If TRUE, does not print exposures output (default: 'FALSE')

Index

count_lst, 2
crosses_centromere, 2

filter_short_contigs, 2
filter_worker, 3

get_centromere_regions, 3
get_contigs, 3
get_reference_signatures, 4
get_subtelomere_coordinate, 4
get_subtelomere_regions, 4

import_ranges, 5
import_snvs, 5
is_contiguous, 5

loh_test, 6
lst_test, 6

mutation_catalog, 6

nnls_exposures, 7
nnls_exposures_worker, 7
nnls_montecarlo, 8

run_snv, 8
run_test, 9

stitch, 9

tai_test, 9
test_multiple, 10