Package 'rgenie'

September 30, 2020

Title Analysis of GenIE Experiments

Version 1.0.0

Description The R package for GenIE analysis is designed to analyze the next-generation sequencing output from a set of GenIE experimental replicates. GenIE (genome-editing interrogation of enhancers) is an experimental method to evaluate the effects of individual single-nucleotide polymorphisms (SNPs) on gene transcription. It is based on targeted CRISPR-Cas9 genome editing in cultured cells to produce indels at a target locus, optionally with a homology-dependent recombination (HDR) construct to create precisely defined genomic changes. Following this, both genomic DNA (gDNA) and RNA are extracted from the same pool of cells, then complementary DNA (cDNA) is generated from the RNA. Both gDNA and cDNA are amplified with locus-specific PCR primers, in multiple replicates, and libraries of the experimental replicates are prepared for next-generation sequencing. The sequencing data from each replicate for each region should be aligned to either the full reference genome or to a locus-specific (amplicon) reference genome for analysis by 'rgenie'.

License MIT + file LICENSE

URL https://github.com/jeremy37/rgenie

BugReports https://github.com/jeremy37/rgenie/issues

Encoding UTF-8

LazyData true

biocViews

Imports dplyr (>= 1.0.2), ggplot2 (>= 3.3.2), magrittr (>= 1.5), stringr (>= 1.4.0), readr (>= 1.3.1), tidyr (>= 1.1.2), tibble (>= 3.0.3), forcats (>= 0.5.0), stats, cowplot (>= 0.9.4), egg (>= 0.4.2), ggdendro (>= 0.1-20), gridExtra (>= 2.3), FNN (>= 1.1.3), variancePartition (>= 1.14.0), Rsamtools (>= 2.0.0), IRanges (>= 2.18.1)

Suggests knitr, rmarkdown, devtools

VignetteBuilder knitr

RoxygenNote 7.1.1

Depends R (>= 3.3)

NeedsCompilation no

Author Jeremy Schwartzentruber [aut, cre]

Maintainer Jeremy Schwartzentruber <jeremy37@gmail.com>

Repository CRAN

Date/Publication 2020-09-30 09:20:03 UTC

R topics documented:

`allele_effect_plot` *Plots estimated effect sizes and confidence intervals for top alleles from a deletion analysis.*

Description

Plots estimated effect sizes and confidence intervals for top alleles from a deletion analysis.

Usage

```
allele_effect_plot(del_result, viewing_window = 40, max_alleles = 40)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`viewing_window`	Window on either size of the CRISPR cut site to show in the plot.
`max_alleles`	The maximum number of alleles to show in the plot.

Value

Returns a ggplot object plotting effect sizes and confidence intervals for top alleles from a deletion analysis. Top alleles are in decreasing order of their total read count in gDNA across replicates.

See Also

`deletion_analysis`

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

allele_effect_plot(mul1_del_results[[1]])
```

`bind_results` *Given results for rgenie grep or deletion analysis of a set of regions, merges together tables across regions.*

Description

Given results for rgenie grep or deletion analysis of a set of regions, merges together tables across regions.

Usage

```
bind_results(results)
```

Arguments

results	A list of rgenie results from either grep or deletion analyses.

Value

Returns a list containing the same tables as in an individual result, but concatenated across regions.

See Also

grep_analysis

deletion_analysis

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

del_tables = bind_results(mul1_del_results)
```

---

deletion_alleles_plot *Plots unique deletion alleles and their "pileup" count profile separately for cDNA and gDNA.*

Description

Plots unique deletion alleles and their "pileup" count profile separately for cDNA and gDNA.

Usage

```
deletion_alleles_plot(del_result, viewing_window = 40, color_by = "window")
```

Arguments

del_result	Result from a call to deletion_analysis.
viewing_window	Window on either size of the CRISPR cut site to show in the plot.
color_by	A string with one of the values: "none" (default), "window", or "sharing".

Value

Returns a ggplot object.

See Also

deletion_analysis

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

deletion_alleles_plot(mul1_del_results[[1]])
```

`deletion_analysis` *Alignment-based GenIE analysis*

Description

For each replicate associated with an input region, `deletion_analysis` identifies HDR or WT sequences, as well as deletion alleles, and returns statistics that indicate for every allele whether the allele:WT ratio differs in cDNA and gDNA. Statistics are also computed for deletion alleles aggregated together.

Usage

```
deletion_analysis(
  regions,
  replicates,
  required_match_left = 10,
  required_match_right = 10,
  crispr_del_window = 100,
  min_mapq = 0,
  max_mismatch_frac = 0.05,
  min_aligned_bases = 50,
  exclude_multiple_deletions = FALSE,
  exclude_nonspanning_reads = TRUE,
  allele_profile = FALSE,
  del_span_start = -20,
  del_span_end = 20,
  quiet = FALSE
)
```

Arguments

`regions`	A data.frame defining GenIE regions.
`replicates`	A data.frame defining GenIE replicates.
`required_match_left`	The length of sequence to the left of the HDR site that must exactly match to identify HDR or WT reads.
`required_match_right`	The length of sequence to the right of the HDR site that must exactly match to identify HDR or WT reads.

`crispr_del_window`
: The window around the cut site within which any deletion is considered a CRISPR deletion. deletions that do not span the region [cut_site - crispr_del_window, cut_site + crispr_del_window] will be ignored; i.e. such reads can be considered HDR or WT.

`min_mapq`
: The minimum mapping quality for reads to be included in the analysis.

`max_mismatch_frac`
: The maximum fraction of mismatches a read can have and be included in the analysis.

`min_aligned_bases`
: The minimum number of aligned bases within the region of interest for a read to be included in the analysis.

`exclude_multiple_deletions`
: If TRUE, then reads with multiple deletions will be excluded from the analysis.

`exclude_nonspanning_reads`
: If TRUE, then reads are excluded if their alignment does not overlap the region's highlight_site (or cut site if no highlight_site is specified)

`allele_profile`
: If TRUE, then the result object will contain data.frames named site_profiles and mismatch_profiles, as detailed in the description below.

`del_span_start`
: An integer that specifies the start of a window, relative to the region's highlight site, within which deletions are counted.

`del_span_end`
: An integer that specifies the end of a window, relative to the region's highlight site, within which deletions are counted.

`quiet`
: If TRUE, then no messages are printing during the analysis.

Details

For a deletion analysis, the regions parameter is a data.frame with a format as follows. All of the column names below must be specified.

name	sequence_name	start	end	highlight_site	cut_site	hdr_allele_profile	wt_allele_profile	re
MUL1_rs6700034	MUL1	1	21	11	9	————-A————-	————-C————-	A

name
: A unique identifier for the region.

sequence_name
: The chromosome or amplicon sequence name.

start
: The start coordinate of the amplicon relative to the chromosome or amplicon reference.

end
: The end coordinate of the amplicon relative to the chromosome or amplicon reference (the end coordinat

highlight_site
: The relative position of the site of interest, usually the HDR SNP site.

cut_site
: The relative position of the cut site.

ref_sequence
: The reference sequence for the amplicon region, which must have length (end - start + 1).

hdr_allele_profile
: An allele profile describing the HDR allele. See details below.

wt_allele_profile
: An allele profile describing the WT allele. See details below.

If multiple rows are defined for ‘regions‘, then a separate analysis is run for each region, using matched replicates from ‘replicates‘.

The allele_profile columns indicate the positions in the amplicon sequence that must match a given nucleotide for a read to be considered either HDR or WT. This sequence must be the same length as the reference sequence, and all other positions should be "-". The total sequence region that must match is determined by both the positions of specified nucleotides and by the **required_match_left** and **required_match_right** parameters. These parameters give the length of sequence which must match the provided reference sequence to the left of the leftmost specified nucleotide, or to the right of the rightmost specified nucleotide.

The replicates parameter is a data.frame with a format as below.

name	replicate	type	bam
MUL1_rs6700034	c1.2	cDNA	bam_amplicon/MUL1_rs6700034_cDNA_rep1_pcr2.sortedByCoord.bam
MUL1_rs6700034	c1.3	cDNA	bam_amplicon/MUL1_rs6700034_cDNA_rep1_pcr3.sortedByCoord.bam

name Indicates the region that a given replicate corresponds with. All replicates matching the name in the regions table
replicate an ID for the replicate, which must be unique among replicates for the region.
type Must have the value "cDNA" or "gDNA", indicating whether a given replicate contains data for cDNA or gDNA.
bam the path (relative to the working directory) to a BAM file with sequencing reads for the replicate.

Statistics can only be computed if there are at least 2 replicates of each type (cDNA and gDNA). Replicates are matched to the region based on the **name** column.

Results

The returned object is a list, where each item is the result for one region. The result for a region (e.g. results[[1]]) is itself a list, with the following items:

region_stats Main analysis output, with statistics indicating whether the HDR/WT levels differ in cDNA relative to gDN
replicate_stats A data.frame with a row for each replicate, which has counts of reads in different categories and some summ
region Details of the input region the result corresponds to.
replicates Details of the input replicates the result corresponds to.
opts A list containing the options that were given for the analysis.
type Has the value “deletion_analysis”, and is used by plotting functions that take a full del_result list as input.

The main output of interest is the ‘region_stats‘ field, which is a one-row data.frame with the following values:

name Name of the region.
hdr_rate_gDNA Fraction of reads in gDNA identified as HDR.
hdr_rate_cDNA Fraction of reads in cDNA identified as HDR.
wt_rate_gDNA Fraction of reads in gDNA identified as WT
wt_rate_cDNA Fraction of reads in cDNA identified as WT.
del_rate_gDNA Fraction of reads in gDNA identified as having a CRISPR deletion.

del_rate_cDNA	Fraction of reads in cDNA identified as having a CRISPR deletion.
hdr_effect	HDR allele: Estimated effect size - the amount by which the HDR:WT ratio differs in cDNA relative
hdr_effect_sd	HDR allele: Standard deviation of the estimated effect size.
hdr_effect_confint_lo	HDR allele: Lower bound of the 95% confidence interval for the effect size.
hdr_effect_confint_hi	HDR allele: Upper bound of the 95% confidence interval for the effect size.
hdr_df_estimated	THDR allele: he degrees of freedom used in the unequal variances t-test, which is estimated from th
hdr_pval	HDR allele: A p value from the unequal variants t-test.

The fields above beginning with 'hdr_' give statistics relating to the HDR allele. There are 6 equivalent fields that begin with 'del_' which relate to all deletions. There are also 6 equivalent fields that begin with 'del_window_', which relate to deletions that are contained within a window around the 'highlight_site' (defined in the region input), the extent of which is determined by 'crispr_del_window' parameter.

Additional fields

The deletion_analysis result object additionally has the following fields:

replicate_qc	A data.frame of summary information for each replicate, with counts of reads in different catego
replicate_alleles	A data.frame of summary information for each unique allele in each replicate.
region_alleles	A data.frame of summary information for each unique allele averaged across all replicates.
replicate_allele_fractions	A data.frame of summary information for the top 20 alleles across all replicates, which is used fo
allele_effect	A data.frame of GenIE effect size estimates (difference in allele:WT ratio in cDNA vs. gDNA) fo
site_profiles	A data.frame which indicates read counts for each combination of observed nucleotides at each s
mismatch_profiles	A data.frame which contains every unique read profile, including mismatches, for reads consider
replicate_list	Internal data for each replicate. Not likely to be of interest to the user.
type	

See Also

grep_analysis

deletion_plots

deletion_summary_plot

experiment_summary_plot

deletion_alleles_plot

deletion_profile_plot

replicate_summary_plot

replicate_qc_plot

allele_effect_plot

get_variance_components

variance_components_plot

power_plots

bind_results

Examples

```
# Note: to run an analysis you need BAM files from a GenIE experiment.
# An example is available for download using download_example().


download_example(dir = "~/genie_example", name = "MUL1")
# Data are downloaded and we can run an rgenie analysis
setwd("~/genie_example/MUL1/")
regions = readr::read_tsv("mul1.genie_regions.tsv")
replicates = readr::read_tsv("mul1.genie_replicates.tsv")
delresults = deletion_analysis(regions, replicates)
deletion_plots(delresults[[1]])
```

`deletion_plots` *Returns all main plots for a single deletion analysis result.*

Description

Returns all main plots for a single deletion analysis result.

Usage

```
deletion_plots(
  del_result,
  opts = genie_plot_options(),
  variance_components_plot = FALSE,
  power_plots = FALSE
)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`opts`	A list with all options needed for rgenie plotting functions.
`variance_components_plot`	If TRUE, then variance_components_plot() is called.
`power_plots`	If TRUE, then power_plots() is called.

Value

Returns a list of ggplot objects.

See Also

genie_plot_options

deletion_analysis

deletion_summary_plot

deletion_alleles_plot

deletion_profile_plot

replicate_summary_plot

replicate_qc_plot

allele_effect_plot

variance_components_plot

power_plots

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

deletion_plots(mul1_del_results[[1]], genie_plot_options())
```

`deletion_profile_plot` *Plots the deletion "pileup" profile separately for each replicate.*

Description

Plots the deletion "pileup" profile separately for each replicate.

Usage

```
deletion_profile_plot(del_result, viewing_window = 40)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`viewing_window`	Window on either size of the CRISPR cut site to show in the plot.

Value

Returns a ggplot object.

See Also

deletion_analysis

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

deletion_profile_plot(mul1_del_results[[1]])
```

`deletion_summary_plot` *Plots a summary of deletion analysis results for a single region.*

Description

Plots a summary of deletion analysis results for a single region.

Usage

```
deletion_summary_plot(del_result)
```

Arguments

`del_result`	Result from a call to deletion_analysis.

Value

Returns a ggplot object with a summary of deletion analysis results for a single region.

See Also

`deletion_analysis`

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

deletion_summary_plot(mul1_del_results[[1]])
```

download_example *Downloads example data for rgenie.*

Description

The example data is a set of BAM files for GenIE replicates.

Usage

```
download_example(dir, name = "MUL1", overwrite = FALSE, quiet = FALSE)
```

Arguments

dir	Directory where example data should be put.
name	The name of the example to download.
overwrite	If FALSE, then data are not downloaded if directory 'dir/name' already exists. Otherwise, all data are downloaded (possibly overwriting files).
quiet	If TRUE, then no messages are printing during the analysis.

Value

Returns a list containing the same tables as in an individual result, but concatenated across regions.

See Also

grep_analysis

deletion_analysis

Examples

```
download_example(dir = "~/genie_example", name = "MUL1")
# Data are downloaded and we can run an rgenie analysis
setwd("~/genie_example/MUL1/")
regions = readr::read_tsv("mul1.genie_regions.tsv")
replicates = readr::read_tsv("mul1.genie_replicates.tsv")
grep_results = grep_analysis(regions, replicates)
```

`experiment_summary_plot` *Plots a summary of genie results across multiple regions.*

Description

Plots a summary of genie results across multiple regions.

Usage

```
experiment_summary_plot(grep_results, del_results)
```

Arguments

`grep_results`	Result from a call to grep_analysis (or NULL).
`del_results`	Result from a call to deletion_analysis (or NULL).

Value

Returns a ggplot object.

See Also

`grep_analysis`

`deletion_analysis`

`bind_results`

Examples

```
# Note: First run deletion_analysis() and/or grep_analysis()
# mul1_del_results and mul1_grep_results are pre-loaded

experiment_summary_plot(mul1_grep_results, mul1_del_results)
```

`genie_plot_options` *Returns a list with default options for all rgenie plots, useful in calls to deletion_plots().*

Description

Returns a list with default options for all rgenie plots, useful in calls to deletion_plots().

Usage

```
genie_plot_options()
```

See Also

deletion_analysis

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

opts = genie_plot_options()
plot_list = deletion_plots(mul1_del_results[[1]], opts)
```

`get_variance_components` *Performs a variance components estimate for each deletion allele based on the replicate metadata provided.*

Description

Performs a variance components estimate for each deletion allele based on the replicate metadata provided.

Usage

```
get_variance_components(
  del_result,
  replicates,
  allele_min_reads = 100,
  allele_min_fraction = 0.001
)
```

Arguments

`del_result`	The result from deletion_analysis
`replicates`	A data.frame defining GenIE replicate metadata.
`allele_min_reads`	The minimum number of reads that a deletion allele must have across all replicates to be included.
`allele_min_fraction`	The minimum fraction of total reads that a deletion allele must have across all replications to be included.

Value

Returns a list with tables vp_cDNA and vp_gDNA, which partition variance according to the metadata columns that begin with "replicate_" in the 'replicates' parameter.

See Also

deletion_analysis

variance_components_plot

Examples

```
# Note: First run deletion_analysis()
# The below isn't run since it can take 10+ seconds to run
# mul1_del_results is a pre-loaded result

vc = get_variance_components(mul1_del_results[[1]], mul1_replicates)
variance_components_plot(vc)
```

---

grep_analysis *Grep-based GenIE analysis*

Description

For each replicate associated with an input region, grep_analysis searches for alleles matching the HDR or WT sequences, and returns statistics that indicate whether the HDR:WT ratio differs in cDNA and gDNA.

Usage

```
grep_analysis(
  regions,
  replicates,
  required_match_left = 10,
  required_match_right = 10,
  min_mapq = 0,
  quiet = FALSE
)
```

Arguments

regions	A data.frame defining GenIE regions.
replicates	A data.frame defining GenIE replicates.
required_match_left	The length of sequence to the left of the HDR site that must exactly match to identify HDR or WT reads.
required_match_right	The length of sequence to the right of the HDR site that must exactly match to identify HDR or WT reads.
min_mapq	The minimum mapping quality for reads to be included in the analysis.
quiet	If TRUE, then no messages are printing during the analysis.

Details

For a grep analysis, the regions parameter is a data.frame with a format as follows. All of the column names below must be specified.

name	sequence_name	start	end	highlight_site	cut_site	hdr_allele_profile	wt_allele_profile	re
MUL1_rs6700034	MUL1	1	21	11	9	————-A————-	————-C————-	A

name A unique identifier for the region.
sequence_name The chromosome or amplicon sequence name.
start The start coordinate of the amplicon relative to the chromosome or amplicon reference.
end The end coordinate of the amplicon relative to the chromosome or amplicon reference (the end coordinat
highlight_site The relative position of the site of interest, usually the HDR SNP site.
cut_site The relative position of the cut site.
ref_sequence The reference sequence for the amplicon region, which must have length (end - start + 1).
hdr_allele_profile An allele profile describing the HDR allele. See details below.
wt_allele_profile An allele profile describing the WT allele. See details below.

If multiple rows are defined for ‘regions‘, then a separate analysis is run for each region, using matched replicates from ‘replicates‘.

The allele_profile columns indicate the positions in the amplicon sequence that must match a given nucleotide for a read to be considered either HDR or WT. This sequence must be the same length as the reference sequence, and all other positions should be "-". The total sequence region that must match is determined by both the positions of specified nucleotides and by the **required_match_left** and **required_match_right** parameters. These parameters give the length of sequence which must match the provided reference sequence to the left of the leftmost specified nucleotide, or to the right of the rightmost specified nucleotide.

The replicates parameter is a data.frame with a format as below.

name	replicate	type	bam
MUL1_rs6700034	c1.2	cDNA	bam_amplicon/MUL1_rs6700034_cDNA_rep1_pcr2.sortedByCoord.bam
MUL1_rs6700034	c1.3	cDNA	bam_amplicon/MUL1_rs6700034_cDNA_rep1_pcr3.sortedByCoord.bam

name Indicates the region that a given replicate corresponds with. All replicates matching the name in the regions table v
replicate an ID for the replicate, which must be unique among replicates for the region.
type Must have the value "cDNA" or "gDNA", indicating whether a given replicate contains data for cDNA or gDNA.
bam the path (relative to the working directory) to a BAM file with sequencing reads for the replicate.

Statistics can only be computed if there are at least 2 replicates of each type (cDNA and gDNA). Replicates are matched to the region based on the **name** column.

Results

The returned object is a list, where each item is the result for one region. The result for a region (e.g. results[[1]]) is itself a list, with the following items:

region_stats	Main analysis output, with statistics indicating whether the HDR/WT levels differ in cDNA relative to gDNA
replicate_stats	A data.frame with a row for each replicate, which has counts of reads in different categories and some summ
region	Details of the input region the result corresponds to.
replicates	Details of the input replicates the result corresponds to.
opts	A list containing the options that were given for the analysis.
type	Has the value "grep_analysis", and is used by plotting functions that take a full grep_result list as input.

The main output of interest is the 'region_stats' field, which is a one-row data.frame with the following values:

name	Name of the region.
effect	Estimated effect size - the amount by which the HDR:WT ratio differs in cDNA relative to gDNA.
effect_sd	Standard deviation of the estimated effect size.
effect_confint_lo	Lower bound of the 95% confidence interval for the effect size.
effect_confint_hi	Upper bound of the 95% confidence interval for the effect size.
df_estimated	The degrees of freedom used in the unequal variances t-test, which is estimated from the data.
pval	A p value from the unequal variants t-test.

See Also

deletion_analysis

grep_summary_plot

Examples

```
# Note: to run an analysis you need BAM files from a GenIE experiment.
# An example is available for download using download_example().


download_example(dir = "~/genie_example", name = "MUL1")
# Data are downloaded and we can run an rgenie analysis
setwd("~/genie_example/MUL1/")
regions = readr::read_tsv("mul1.genie_regions.tsv")
replicates = readr::read_tsv("mul1.genie_replicates.tsv")
grep_results = grep_analysis(regions, replicates)
grep_summary_plot(grep_results[[1]])
```

grep_summary_plot *Plots a summary of genie results from a grep analysis.*

Description

Plots a summary of genie results from a grep analysis.

Usage

```
grep_summary_plot(grep_result)
```

Arguments

grep_result	Result from a call to grep_analysis.

Value

Returns a ggplot object.

See Also

grep_analysis

Examples

```
# Note: First run grep_analysis()
# mul1_grep_results is a pre-loaded result

grep_summary_plot(mul1_grep_results[[1]])
```

mul1_del_results *GenIE replicates for the MUL1 example.*

Description

A list with a single result object from GenIE deletion analysis for the MUL1 example.

Usage

```
mul1_del_results
```

Format

A one-item list with a deletion_analysis result object.

mul1_grep_results *GenIE grep results list.*

Description

A list with a single result object from GenIE grep analysis for the MUL1 example.

Usage

```
mul1_grep_results
```

Format

A one-item list with a grep_analysis result object.

mul1_regions *GenIE regions for the MUL1 example.*

Description

A data.frame with GenIE regions details for the MUL1 example.

Usage

```
mul1_regions
```

Format

A data frame with 1 row and 9 variables:

name region name

sequence_name chromosome or amplicon name

start start coordinate of the region

end end coordinate of the region

highlight_site coordinate of the SNP site

cut_site coordinate of the cut site

hdr_allele_profile HDR allele profile

wt_allele_profile WT allele profile

ref_sequence sequence of the amplicon region

mul1_replicates *GenIE replicates for the MUL1 example.*

Description

A data.frame with GenIE replicate details for the MUL1 example.

Usage

```
mul1_replicates
```

Format

A data frame with 12 rows and 5 variables:

name region name

replicate short name of the replicate

type cDNA" or "gDNA"

vp_extraction which cDNA/gDNA extraction the replicate is from

bam path to the BAM file of data for the replicate

power_analysis *Performs estimates of power to detect effects of alleles at different read fractions, given the variance observed in the del_result replicates.*

Description

Performs estimates of power to detect effects of alleles at different read fractions, given the variance observed in the del_result replicates.

Usage

```
power_analysis(del_result, allele_min_reads = 100, WT_fraction = NA)
```

Arguments

del_result	The result from deletion_analysis
allele_min_reads	The minimum number of reads that a deletion allele must have across all replicates to be included.
WT_fraction	If specified, then the model will assume this fraction of WT reads

Value

Returns...

See Also

deletion_analysis

power_plots

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

pwr = power_analysis(mul1_del_results[[1]])
power_plots(mul1_del_results[[1]])
```

`power_plots` *Returns a set of plots summarising the expected power to detect a significant effect given various allele fraction and effect size combinations.*

Description

Returns a set of plots summarising the expected power to detect a significant effect given various allele fraction and effect size combinations.

Usage

```
power_plots(del_result, allele_min_reads = 100, WT_fraction = NA)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`allele_min_reads`	The minimum number of reads that a deletion allele must have across all replicates to be included.
`WT_fraction`	If specified, then the model will assume this fraction of WT reads

Value

Returns a list with three ggplot objects:

- cv_plot
- power_plot
- replicate_allocation_plot

See Also

deletion_analysis

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

power_plots(mul1_del_results[[1]])
```

`replicate_qc_plot` *Plots quality control metrics for deletion analysis replicates.*

Description

Plots quality control metrics for deletion analysis replicates.

Usage

```
replicate_qc_plot(del_result, outlier_threshold = NA)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`outlier_threshold`	A numeric threshold for the outlier score, above which replicates will be colored differently.

Value

Returns a ggplot object with quality control metrics for deletion analysis replicates.

See Also

`deletion_analysis`

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

replicate_qc_plot(mul1_del_results[[1]])
```

`replicate_summary_plot` *Plots a summary of deletion analysis replicates.*

Description

Plots a summary of deletion analysis replicates.

Usage

```
replicate_summary_plot(del_result, outlier_threshold = NA)
```

Arguments

`del_result`	Result from a call to deletion_analysis.
`outlier_threshold`	A numeric threshold for the outlier score, above which replicates will be colored differently.

Value

Returns a ggplot object with a summary of deletion analysis replicates.

See Also

`deletion_analysis`

Examples

```
# Note: First run deletion_analysis()
# mul1_del_results is a pre-loaded result

replicate_summary_plot(mul1_del_results[[1]])
```

`rgenie` *rgenie: Analysis of GenIE experiments*

Description

rgenie is a package to analyze the next-generation sequencing output from a set of GenIE experimental replicates.

Details

GenIE (genome-editing interrogation of enhancers) is an experimental method to evaluate the effects of individual SNPs on gene transcription. It is based on targeted CRISPR-Cas9 genome editing in cultured cells to produce indels at a target locus, optionally with a homology-dependent recombination (HDR) construct to create precisely defined genomic changes. Following this, both genomic DNA (gDNA) and RNA are extracted from the same pool of cells, and then cDNA is generated. Both gDNA and cDNA are amplified with locus-specific PCR primers, in multiple replicates, and libraries of the experimental replicates are prepared for next-generation sequencing. The sequencing data from each replicate for each region should be aligned to either the full reference genome or to a locus-specific (amplicon) reference genome for analysis by rgenie.

rgenie analyses

Two types of analyses can be run with rgenie: a grep-based analysis, which uses pattern matching to identify HDR vs. wild-type (WT) reads, and an alignment-based analysis, which uses all Cas9-generated alleles, including deletions. The analysis is run for a given **region**, which must have multiple **replicates** for both cDNA and gDNA. These analyses take as input a data.frame defining the region, and a data.frame defining replicates.

The result of an rgenie analysis is a list object with multiple tables, which depend on whether a grep or deletion analyses was done. Plotting functions take as input the result of one of these main rgenie analyses.

See Also

`grep_analysis`

`deletion_analysis`

---

variance_components_plot

Plots variance components estimates for all unique alleles.

Description

Plots variance components estimates for all unique alleles.

Usage

```
variance_components_plot(varcomp, split_by_fraction = FALSE)
```

Arguments

`varcomp`	Result from a call to get_variance_components.
`split_by_fraction`	If TRUE, then points are colored by allele fraction.

Value

Returns a ggplot object.

See Also

deletion_analysis

get_variance_components

Examples

```
# Note: First run deletion_analysis()
# The below isn't run since it can take 10+ seconds to run

# mul1_del_results is a pre-loaded result
vc = get_variance_components(mul1_del_results[[1]], mul1_replicates)
variance_components_plot(vc)
```

Index