Package 'MixviR'

October 23, 2022

Title Analysis and Exploration of Mixed Microbial Genomic Samples

Version 3.5.0

Description Tool for exploring DNA and amino acid variation and inferring the presence of target lineages from microbial high-throughput genomic DNA samples that potentially contain mixtures of variants/lineages. MixviR was originally created to help analyze environmental SARS-CoV-2/Covid-19 samples from environmental sources such as wastewater or dust, but can be applied to any microbial group. Inputs include reference genome information in commonly-used file formats (fasta, bed) and one or more variant call format (VCF) files, which can be generated with programs such as Illumina's DRAGEN, the Genome Analysis Toolkit, or bcftools. See DePristo et al (2011) <doi:10.1038/ng.806> and Danecek et al (2021) <doi:10.1093/gigascience/giab008> for these tools, respectively. Available outputs include a table of mutations observed in the sample(s), estimates of proportions of target lineages in the sample(s), and an R Shiny dashboard to interactively explore the data.

License GPL-3

Encoding UTF-8

URL https://github.com/mikesovic/MixviR

BugReports https://groups.google.com/g/mixvir

RoxygenNote 7.1.1

Imports Biostrings, dplyr, DT, ggplot2 (>= 3.1.0), glue, httr, lubridate, magrittr, plotly (>= 4.9.4), readr (>= 2.0.0), shiny, stats (>= 1.4.0), stringr (>= 1.1.0), tidyr (>= 0.8.0), utils, vcfR (>= 1.11.0)

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Michael Sovic [aut, ccp, cre] (<https://orcid.org/0000-0002-8556-3704>), Francesca Savona [res], Zuzana Bohrerova [res, fnd], Seth Faith [ccp, ctb] (0000-0002-0441-9859)

Maintainer Michael Sovic <sovic.1@osu.edu>

Repository CRAN

Date/Publication 2022-10-22 22:17:49 UTC

R topics documented:

`add_depths_to_ref` *Add Read Depths To Ref*

Description

Add column of total read depths for a given sample to reference df. Run as part of id_mutations().

Usage

```
add_depths_to_ref(ref, samp.variants)
```

Arguments

`ref`	reference genome in "MixVir" format (from create_ref() function)
`samp.variants`	Data frame produced by function vcf_to_mixvir(). Contains columns "CHR", "POS", "REF", "ALT", "DP", "ALT_COUNT".

Value

Original 'ref' data frame with depth (DP) column added: cols "genomic_pos" "ref_base" "gene" "ref_codon" "ref_AA" "GENE_AA_POS" "ref_identity" "DP"

call_mutations *Identify Variants From A Potentially Mixed Sample*

Description

Identify full set of amino acid/SNP/indel changes from one or more samples (includes changes based on both SNVs and indels). This is generally the first function run in a MixviR analysis.

Usage

```
call_mutations(
  sample.dir = NULL,
  fasta.genome,
  bed,
  reference = "custom",
  min.alt.freq = 0.01,
  name.sep = NULL,
  write.all.targets = FALSE,
  lineage.muts = NULL,
  genetic.code.num = "1",
  out.cols = c("SAMP_NAME", "CHR", "POS", "GENE", "ALT_ID", "AF", "DP"),
  write.mut.table = FALSE,
  outfile.name = "sample_mutations.csv",
  indel.format = "Fwd",
  csv.infiles = FALSE
)
```

Arguments

sample.dir	**Required** Path to directory containing vcf files for each sample to be analyzed. VCF's need to contain "DP" and "AD" flags in the FORMAT field. This directory should not contain any other files.
fasta.genome	Path to fasta formatted reference genome file. **Required unless *reference* is defined**.
bed	Path to bed file defining features of interest (open reading frames to translate). Should be tab delimited and have 6 columns (no column names): chr, start, end, feature_name, score (not used), strand. **Required unless *reference* is defined**. See example at https://github.com/mikesovic/MixviR/blob/main/raw_files/sars_cov2_genes.bed.
reference	Optional character defining a pre-constructed MixviR reference (created with 'create_ref()"). "Wuhan" uses pre-generated Sars-Cov2 ref genome. Otherwise, *fasta.genome* and *bed* are required to generate MixviR formatted reference as part of the analysis.
min.alt.freq	Minimum frequency (0-1) for retaining alternate alleles. Default = 0.01. Extremely low values (i.e. zero) are not recommended here - see vignette for details.

`name.sep` Optional character in input file names that separates the unique sample identifier (characters preceeding the separator) from any additional text. Only text preceeding the first instance of the character will be retained and used as the sample name.

`write.all.targets` Logical that, if TRUE, reports sequencing depths for genomic positions associated with mutations of interest that are not observed in the sample, in addition to all mutations observed in the sample. If TRUE, requires columns "Chr" and "Pos" to be included in the *lineage.muts* file. Default FALSE.

`lineage.muts` Path to optional csv file defining target mutations and their underlying genomic positions. Requires cols "Gene", "Mutation", "Lineage", "Chr" and "Pos". This is used to report the sequencing depths for relevant positions when the mutation of interest is not observed in the sample. See *write.all.targets*. This file is also used in `explore_mutations()`, where the "Chr" and "Pos" columns are optional. Only necessary here in conjunction with *write.all.targets*.

`genetic.code.num` Number (character) associated with the genetic code to be used for translation. Details can be found at https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi.

`out.cols` Character vector with names of columns to be returned. Choose from: CHR, POS, REF_BASE, GENE, STRAND, REF_CODON, REF_AA, GENE_AA_POS, REF_IDENT, REF, ALT, AF, ALT_COUNT, SAMP_CODON, SAMP_AA, ALT_ID, DP, SAMP_NAME, TYPE. The default columns SAMP_NAME, CHR, POS, ALT_ID, AF, DP must be included to run explore_mutations().

`write.mut.table` Logical indicating whether to write the 'samp_mutations' data frame (see "Value" below) to a text file (csv). Default = FALSE. See *outfile.name*.

`outfile.name` Path to file where (csv) output will be written if *write.mut.table* is TRUE. Default: "sample_mutations.csv" (written to working directory)

`indel.format` Defines the naming convention for indels. Default is "Fwd", meaning the name would look like S_del144/144. "Rev" switches this to S_144/144del.

`csv.infiles` Logical to indicate whether files in *sample.dir* directory are in vcf or csv format. All files must be of the same format. If csv, they must contain columns named: "CHR" "POS" "REF" "ALT" "DP" "ALT_COUNT". See the 'batch_vcf_to_mixvir()" function to convert vcfs to csv format). Default is FALSE (input is in vcf format). This exists primarily for legacy reasons.

Value

A data frame containing variants observed for each sample, positions of the underlying mutations, and other (customizable) information. This data frame can be saved as an object in the global environment and/or written to a file (see *write.mut.table*), and in either case serves as input for the *MixviR* functions `explore_mutations()` and `estimate_lineages()`.

Examples

```
##For SARS-CoV-2
```

```
#call_mutations(sample.dir = system.file("extdata", "vcf", package = "MixviR"),
#                name.sep = "_", reference = "Wuhan")

##OR if defining a custom reference, follow this pattern...
#genome<-"https://raw.githubusercontent.com/mikesovic/MixviR/main/raw_files/GCF_ASM985889v3.fa"
#features<-"https://raw.githubusercontent.com/mikesovic/MixviR/main/raw_files/sars_cov2_genes.bed"

#call_mutations(sample.dir = system.file("extdata", "vcf", package = "MixviR"),
#                name.sep = "_",
#                fasta.genome = genome,
#                bed = features)
```

`create_ref` *Create MixVir-formatted reference genome object*

Description

Uses a fasta genome and bed file defining features of interest (genes/ORFs) to create a data frame that's used as a reference to translate nucleotide data to amino acids and subsequently call variants/mutations from a sample.

Usage

```
create_ref(genome, feature.bed, code.num = "1", removed.genes = NULL)
```

Arguments

`genome`	*(Required)* Path to fasta formatted genome file
`feature.bed`	*(Required)* Path to bed file defining features of interest (open reading frames to translate). Tab delimited with 6 columns (without column names):"chr", "start", "end", "feature_name", "score" (not used), and "strand".
`code.num`	Number (character) associated with the genetic code to be used for translation. Details can be found at https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi.
`removed.genes`	Character providing path/name of tab-separated file that will be written that stores names of genes (if any) in the feature.bed file that were removed because they didn't have an allowed size (not even multiples of 3). If NULL (default), file is not written.

Value

A data frame with columns CHR,POS,REF_BASE,GENE,STRAND,REF_CODON,REF_AA,GENE_AA_POS,REF_IDEN

Examples

```
site1 <- "https://raw.githubusercontent.com/mikesovic/MixviR/main/raw_files/GCF_ASM985889v3.fa"
site2 <- "https://raw.githubusercontent.com/mikesovic/MixviR/main/raw_files/sars_cov2_genes.bed"

if (httr::http_error(site1) | httr::http_error(site2)) {
     message("No internet connection or data source broken.")
     return(NULL)
} else {
create_ref(
 genome = site1,
 feature.bed = site2,
 code.num = "1")
}
```

`estimate_lineages` *Estimate Lineage Proportions In Samples*

Description

Create summary tables containing data on lineages identified in samples, including estimates of relative proportions of lineages and identities of associated characteristic mutations.

Usage

```
estimate_lineages(
  muts.df,
  min.alt.freq = 0.01,
  dates = NULL,
  lineage.muts = NULL,
  read.muts.from = NULL,
  scale = TRUE,
  use.median = FALSE,
  outfile.name = NULL,
  presence.thresh = 0.5,
  samps.to.inc = NULL,
  locs.to.inc = NULL,
  lineages.to.inc = NULL,
  report.all = FALSE,
  depths.from = "all"
)
```

Arguments

`muts.df`	A data frame (produced by `call_mutations()`) storing mutation information for samples to analyze. Must contain columns SAMP_NAME, CHR, POS, GENE, ALT_ID, AF, & DP. Alternatively, the mutation data can be read in from a (comma-separated) file with the `read.muts.from()` argument. See the *write.mut.table* argument in `call_mutations()`.

`min.alt.freq` Minimum frequency (0-1) for mutation to be counted. Default = 0.01.

`dates` Path to optional csv file with cols "SAMP_NAME", "LOCATION", and "DATE". Sample names need to match those in *samp_mutations* data frame created by `call_mutations()`. Dates should be provided in the format *mmddyyyy*.

`lineage.muts` *(Required)* Path to csv file with cols "Gene", "Mutation", and "Lineage" defining mutations associated with lineages of interest. See example file at "https://github.com/mikesovic/MixviR/l Additional columns will be ignored.

`read.muts.from` An alternative to *muts.df* for providing input. If a data frame generated by `call_mutations()` was previously written to a (comma-separated) file (see *write.mut.table* in `call_mutations()`), the mutation data can be read in from that file by providing its path.

`scale` Logical to indicate whether estimated proportions of lineages within a sample should be scaled down to sum to 1 if the sum of the initial estimates is > 1. Default = TRUE.

`use.median` Logical to define the metric used to estimate frequencies of lineages in samples. Default = FALSE (mean is used).

`outfile.name` If writing output to file, a character string giving the name/path of the file (csv) to be written.

`presence.thresh` Numeric (0-1) defining a proportion of characteristic mutations that must be present in the sample for a lineage to be considered present. This threshold is applied if *report.all* = FALSE (the default).

`samps.to.inc` Character vector of one or more sample names to include. If NULL (default), all samples are included.

`locs.to.inc` Character vector of one or more locations to include. If NULL (default), all locations are included. Applies only if a dates file is provided, and these locations must match those in the 'LOCATION' column of that file.

`lineages.to.inc` Character vector of one or more lineages to test for and report in results. If NULL (default), all lineages listed in the lineage.muts file are evaluated and reported.

`report.all` Logical indicating whether to report results for all lineages (TRUE), or just those with a proportion of mutations present that exceeds *presence.thresh*. Default FALSE.

`depths.from` Character, one of "all" (default) or "characteristic". If "all", average sequencing depths are calculated based on all mutations in a sample. If "characteristic", mean depths are calculated from the set of mutations that occur in only one analyzed lineage (mutations shared by two or more lineages are filtered out prior to calculating depths).

Value

Data frame containing estimates of proportions of each lineage in the sample.

Examples

```
estimate_lineages(lineage.muts = system.file("extdata",
                                             "example_lineage_muts.csv",
                                             package = "MixviR"),
                  read.muts.from = system.file("extdata",
                                               "sample_mutations.csv",
                                               package = "MixviR"))
```

`explore_mutations` *MixviR Shiny Dashboard*

Description

Open dashboard to explore mutation data generated with `call_mutations()`.

Usage

```
explore_mutations(
  muts.df,
  dates = NULL,
  lineage.muts = NULL,
  read.muts.from = NULL,
  all.target.muts = FALSE
)
```

Arguments

`muts.df`	A data frame (produced by `call_mutations()`) storing mutation information for samples to analyze. Must contain columns SAMP_NAME, CHR, POS, GENE, ALT_ID, AF, & DP. Alternatively, the mutation data can be read in from a (tab-separated) file with the `read.muts.from()` argument. See also the *write.mut.table* argument in `call_mutations()`.
`dates`	Path to optional csv file with cols "SAMP_NAME", "LOCATION", and "DATE". Sample names need to match those in the *muts.df* data frame created by `call_mutations()`. Dates should be provided in the format *mmddyyyy*.
`lineage.muts`	Path to optional csv file with required cols "Gene", "Mutation", and "Lineage" defining mutations associated with lineages of interest. See example file at "https://github.com/mikesovic/MixviR/blob/main/mutation_files/outbreak_20211202.csv". Additional columns will be ignored.
`read.muts.from`	An alternative to *muts.df* for providing mutation input. If a data frame generated by `call_mutations()` was previously written to a (comma separated) file (see *write.mut.table* in `call_mutations()`), the mutation data can be read in from that file by providing its path. The fields "SAMP_NAME, CHR, POS, GENE, ALT_ID, AF, DP" must be present (additional fields will be ignored).

`all.target.muts`
: Logical to indicate whether results for all target mutations were written to the output of the `call_mutations()` function. See *write.all.targets* option in `call_mutations()`. Default FALSE. If TRUE, more informative sequencing depth information can be provided in the dashboard.

Value

Shiny Dashboard to Explore Data

Examples

```
if (interactive()) {explore_mutations(read.muts.from = system.file("extdata",
                                                                   "sample_mutations.csv",
                                                                     package = "MixviR"),
                                      lineage.muts = system.file("extdata",
                                                                 "example_lineage_muts.csv",
                                                                     package = "MixviR"))}
if (interactive()) {explore_mutations(read.muts.from = system.file("extdata",
                                                                   "sample_mutations.csv",
                                                                     package = "MixviR"),
                                      dates = system.file("extdata",
                                                          "example_location_date.csv",
                                                          package = "MixviR"),
                                      lineage.muts = system.file("extdata",
                                                                 "example_lineage_muts.csv",
                                                                     package = "MixviR"))}
```

`get_codons` *Get Codons From Gene Sequence*

Description

Group a gene sequence into codons (triplets) that can be used for subsequent translation. If any elements of the character vector have length >1 (insertions in ALT column of VCF), they are trimmed to the first base. Used by id.snps, id.indels, and create_ref functions.

Usage

```
get_codons(gene.seq, rev = FALSE)
```

Arguments

`gene.seq`
: Character vector containing gene sequence. The length of this vector should be equal to the length of sequence.

`rev`
: Logical indicating whether the reverse complement of the gene.seq should be used.

Value

A character vector containing translated amino acids associated with each nucleotide position in the input vector.

Examples

```
get_codons(gene.seq = c("A","U","G","C","A","T","T","T","A","C","A","G","T","A","A"))
```

`id_indels` *ID Indel-based Amino Acid Changes*

Description

Identify amino acid changes associated with indel variation. Changes associated with SNVs are identified in separate function. Used by call_mutations() function.

Usage

```
id_indels(variant.calls, ref)
```

Arguments

`variant.calls`	Data frame with cols POS, REF, ALT, AF, DP. Additional columns will be ignored.
`ref`	reference genome in "MixVir" format (from create_ref() function)

Value

Data frame that includes amino acid calls based on indel variants observed in sample. Contains cols "POS", "REF_BASE", "GENE", "REF_CODON", "REF_AA", "GENE_AA_POS", "REF_IDENT", "REF", "ALT", "ALT_freq", "ALT_COUNT", "samp_codon", "samp_AA", "samp_identity", "DP"

`id_snps` *ID SNV-based Amino Acid Changes*

Description

Identify amino acid changes associated with single nucleotide variation. Changes associated with indels are identified in separate function. Used by call_mutations() function.

Usage

```
id_snps(variant.calls, ref, code.num = "1")
```

Arguments

variant.calls	Data frame with cols POS, REF, ALT, AF (alt freq), DP (total read depth).
ref	reference genome in "MixVir" format (from create_ref() function)
code.num	Number defining the genetic code to use for translation

Value

Data frame that includes amino acid calls based on SNP/SNV variants observed in sample. Contains cols "POS", "REF_BASE", "GENE", "REF_CODON", "REF_AA", "GENE_AA_POS", "REF_IDENT", "REF", "ALT", "ALT_freq", "ALT_COUNT", "samp_codon", "samp_AA", "samp_identity", "DP"

---

vcf_to_mixvir *Convert Sample VCF to MixviR Input Format*

Description

Create data frame with relevant contents of VCF

Usage

```
vcf_to_mixvir(infile, max.vcf.size = 1e+08)
```

Arguments

infile	Path to a vcf file that must contain "DP" and "AD" flags in the FORMAT field.
max.vcf.size	Max memory usage (in bytes) allowed when reading in vcf file (from vcfR).

Value

Data frame with cols "CHR" "POS" "REF" "ALT" "DP" "REF_COUNT" "ALT_COUNT"

Examples

```
vcf_to_mixvir(infile = system.file("extdata", "vcfs", "Sample1_04182021.vcf.gz", package="MixviR"))
```

Index