Package 'crispRdesignR'

May 8, 2026

Type Package

Title Guide Sequence Design for CRISPR/Cas9

Version 1.1.7

Encoding UTF-8

Description Designs guide sequences for CRISPR/Cas9 genome editing and provides information on sequence features pertinent to guide efficiency. Sequence features include annotated off-target predictions in a user-selected genome and a predicted efficiency score based on the model described in Doench et al. (2016) <doi:10.1038/nbt.3437>. Users are able to import additional genomes and genome annotation files to use when searching and annotating off-target hits. All guide sequences and off-target data can be generated through the 'R' console with sgRNA_Design() or through 'crispRdesignR's' user interface with crispRdesignRUI(). CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) and the associated protein Cas9 refer to a technique used in genome editing.

URL <https://github.com/dylanbeeber/crispRdesignR>

License GPL-3

Depends R (>= 2.10)

Imports Biostrings, gbm, GenomicRanges, BiocGenerics, IRanges, GenomeInfoDb, S4Vectors, rtracklayer, stringr, vtreat, shiny, DT

Suggests BSgenome.Scerevisiae.UCSC.sacCer3

RoxygenNote 7.1.0

NeedsCompilation no

Author Dylan Beeber [aut, cre], Frederic Chain [aut]

Maintainer Dylan Beeber <dylan.beeber@gmail.com>

Repository CRAN

Date/Publication 2023-04-12 09:50:02 UTC

Contents

`crispRdesignRUI` *UI caller for crispRdesignR*

Description

Activates the shiny UI for the crispRdesignR package

Usage

```
crispRdesignRUI(max_gtf_size = 150)
```

Arguments

`max_gtf_size`	The maximum size (in MB) of the geneome annotation file (.gtf) that can be used with the shiny App. By default this is set to 150.

Value

No return value, called to initiate user interface.

Author(s)

Dylan Beeber

Examples

```
requireNamespace("gbm", quietly = TRUE)
requireNamespace("Biostrings", quietly = TRUE)
if (interactive()) {
  crispRdesignRUI()
  }
```

`Doench_2016_processing` *Donech 2016 Processsing*

Description

Warning: This function is not designed to be directly called by the user. This function is used internally in sgRNA_design() and sgRNA_design_function().

Internal function that encodes all sgRNA sequence information into a data frame. This data frame is then used in conjunction with the Rule_Set_2_Model to predict effciency scores for the generated sgRNA.

Usage

```
Doench_2016_processing(seqlist)
```

Arguments

`seqlist`	A list of 30-mer sgRNA (as a character string) with the sgRNA sequence spanning from positions 5 to 24.

Value

A data frame containing processed data on the presence of relevant sequence features to the Rule_Set_2_Model for effciency scoring. Includes information on single nucleotide positions, dinucleotide positions, single nucleotide count, dinucleotide count, GC count, PAM neighboring nucleotides, and melting temperatures. Single nulceotide positions, dinucleotide positions, and PAM neighboring nucleotides are all one-hot encoded.

Author(s)

Dylan Beeber

`FindsgRNAfunction` *sgRNA target design for Shiny App*

Description

Warning: This function should not be directly called by the user - it must be called though RunShiny.R

Designs sgRNA based on inputs provided in the Shiny App.

Usage

```
sgRNA_design_function(userseq, genomename, gtf,
designprogress, userPAM, calloffs, annotateoffs)
```

Arguments

userseq	The target sequence to generate sgRNA guides for. Can either be a character sequence containing DNA bases or the name of a fasta file in the working directory.
genomename	The name of a geneome (from the BSgenome package) to check off-targets for.
gtf	The name of a genome annotation file (.gtf) in the working directory to check off-target sequences against.
designprogress	Assists in communicating the progress of the sgRNA design to the Shiny App.
userPAM	An optional argument used to set a custom PAM for the sgRNA. If not set, the function will default to the "NGG" PAM. Warning: Doench efficieny scores are only accurate for the "NGG" PAM.
calloffs	If TRUE, the function will search for off-targets in the genome chosen specified by the genomename argument. If FALSE, off-target calling will be skipped.
annotateoffs	If TRUE, the function will provide annotations for the off-targets called using the genome annotation file specified by the gtfname argument. If FALSE, off-target annotation will be skipped.

Value

A list containing all data on the generated sgRNA and all off-target information. List items 1 through 15 include information on each individual sgRNA, including the sgRNA sequence itself, PAM, location, direction relative to the target sequence, GC content, homopolymer presence, presence of self-complementarity, off-target matches, predicted efficiency score, and a notes column that summarizes unfavorable sequence features. List items 16 through 27 include all information on off-target matches, including the original sgRNA sequence, off-target sequence, chromosome, location, direction relative to the target sequence, number of mismatches, gene ID, gene name, type of DNA, and exon number.

Author(s)

Dylan Beeber

---

getofftargetdata *Off Target Data Frame Creation*

Description

Will provide a data frame with all information about the generated sgRNA returned by the sgRNA_design function.

Usage

```
getofftargetdata(x)
```

Arguments

x	the data list generated by the sgRNA_design function

Value

A data frame containing all information on potential off-target sequences generated by the sgRNA_design function. Information includes the original sgRNA sequence, off-target sequence, chromosome, location, direction relative to the target sequence, number of mismatches, gene ID, gene name, type of DNA, and exon number.

Author(s)

Dylan Beeber

Examples

```
## Quick example without off-target searching or annotation
## First generate data with the sgRNA_Design Function
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- "placeholder"
gtfname <- "placeholder"
alldata <- sgRNA_design(testseq, usergenome, gtfname, calloffs = FALSE)
## Then separate and format the off-target data with getofftargetdata()
final_data <- getofftargetdata(alldata)


## Longer example with off-target searching and annotation
## First generate data with the sgRNA_Design Function
requireNamespace("BSgenome.Scerevisiae.UCSC.sacCer3", quietly = TRUE)
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- BSgenome.Scerevisiae.UCSC.sacCer3::BSgenome.Scerevisiae.UCSC.sacCer3
gtfname <- "Saccharomyces_cerevisiae.R64-1-1.92.gtf.gz"
annotation_file <- system.file("example_data", gtfname, package = "crispRdesignR")
alldata <- sgRNA_design(testseq, usergenome, annotation_file)
## Then separate and format the sgRNA data with getofftargetdata()
final_data <- getofftargetdata(alldata)
```

---

getsgRNAdata *sgRNA Data Frame Creation*

Description

Will provide a data frame with all information about the generated sgRNA returned by the sgRNA_design function.

Usage

```
getsgRNAdata(x)
```

Arguments

x	the data list generated by the sgRNA_design function

Value

A data frame containing all information specific to sgRNA sequences generated by the sgRNA_design function. Information includes the sgRNA sequence itself, PAM, location, direction relative to the target sequence, GC content, homopolymer presence, presence of self-complementarity, off-target matches, predicted efficiency score, and a notes column that summarizes unfavorable sequence features.

Author(s)

Dylan Beeber

Examples

```
## Quick example without off-target searching or annotation
## First generate data with the sgRNA_Design Function
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- "placeholder"
gtfname <- "placeholder"
alldata <- sgRNA_design(testseq, usergenome, gtfname, calloffs = FALSE)
## Then separate and format the sgRNA data with getsgRNAdata()
final_data <- getsgRNAdata(alldata)


## Longer example with off-target searching and annotation
## First generate data with the sgRNA_Design Function
requireNamespace("BSgenome.Scerevisiae.UCSC.sacCer3", quietly = TRUE)
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- BSgenome.Scerevisiae.UCSC.sacCer3::BSgenome.Scerevisiae.UCSC.sacCer3
gtfname <- "Saccharomyces_cerevisiae.R64-1-1.92.gtf.gz"
annotation_file <- system.file("example_data", gtfname, package = "crispRdesignR")
alldata <- sgRNA_design(testseq, usergenome, annotation_file)
## Then separate and format the sgRNA data with getsgRNAdata()
final_data <- getsgRNAdata(alldata)
```

---

sgRNA_design *sgRNA Target Design*

Description

sgRNA_design returns information to design sgRNA sequences based on a given target sequence, a genome to annotate off-target information, and a genome annoation file (.gtf), to annotate the off-target findings.

Usage

```
sgRNA_design(userseq, genomename, gtfname, userPAM, calloffs = TRUE, annotateoffs = TRUE)
```

Arguments

userseq	The target sequence to generate sgRNA guides for. Can either be a character sequence containing DNA bases or the name of a fasta file in the working directory.
genomename	The name of a geneome (from the BSgenome package) to check off-targets for.
gtfname	The name of a genome annotation file (.gtf) in the working directory to check off-target sequences against.
userPAM	An optional argument used to set a custom PAM for the sgRNA. If not set, the function will default to the "NGG" PAM. Warning: Doench efficieny scores are only accurate for the "NGG" PAM.
calloffs	If TRUE, the function will search for off-targets in the genome chosen specified by the genomename argument. If FALSE, off-target calling will be skipped.
annotateoffs	If TRUE, the function will provide annotations for the off-targets called using the genome annotation file specified by the gtfname argument. If FALSE, off-target annotation will be skipped.

Details

Important Note: When designing sgRNA for large genomes (billions of base pairs), use short query DNA sequences (under 500 bp). Depending on your hardware checking for off-targets can be quite computationally intensive and may take several hours if not limited to smaller query sequences.

Value

A list containing all data on the generated sgRNA and all off-target information. List items 1 through 15 include information on each individual sgRNA, including the sgRNA sequence itself, PAM, location, direction relative to the target sequence, GC content, homopolymer presence, presence of self-complementarity, off-target matches, predicted efficiency score, and a notes column that summarizes unfavorable sequence features. List items 16 through 27 include all information on off-target matches, including the original sgRNA sequence, off-target sequence, chromosome, location, direction relative to the target sequence, number of mismatches, gene ID, gene name, type of DNA, and exon number.

Author(s)

Dylan Beeber

Examples

```
## Quick example without off-target searching or annotation
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- "placeholder"
gtfname <- "placeholder"
alldata <- sgRNA_design(testseq, usergenome, gtfname, calloffs = FALSE)
```

```
## Designing guide RNA for a target region as a test string, using
## the Saccharomyces Cerevisiae genome and genome annotation file:
requireNamespace("BSgenome.Scerevisiae.UCSC.sacCer3", quietly = TRUE)
testseq <- "GGCAGAGCTTCGTATGTCGGCGATTCATCTCAAGTAGAAGATCCTGGTGCAGTAGG"
usergenome <- BSgenome.Scerevisiae.UCSC.sacCer3::BSgenome.Scerevisiae.UCSC.sacCer3
gtfname <- "Saccharomyces_cerevisiae.R64-1-1.92.gtf.gz"
annotation_file <- system.file("example_data", gtfname, package = "crispRdesignR")
alldata <- sgRNA_design(testseq, usergenome, annotation_file)

## Designing guide RNA for a target region as a text file, using
## the Saccharomyces Cerevisiae genome and genome annotation file,
## while switching genome annotation off:
testseq <- system.file("example_data", "ExampleDAK1seq.txt", package = "crispRdesignR")
alldata2 <- sgRNA_design(testseq, usergenome, annotation_file, annotateoffs = FALSE)
```

Index