Package 'CRISPRseek'

July 25, 2025

Version 1.49.1

Title Design of guide RNAs in CRISPR genome-editing systems

Author Lihua Julie Zhu
Paul Scemama
Benjamin R. Holmes
Hervé Pagès
Kai Hu
Hui Mao
Michael Lawrence
Isana Veksler-Lublinsky
Victor Ambros
Neil Aronin
Michael Brodsky
Devin M Burris

Maintainer Lihua Julie Zhu <julie.zhu@umassmed.edu>
Kai Hu <kai.hu@umassmed.edu>

Date 2024-11-14

Description The package encompasses functions to find potential guide RNAs for the CRISPR-based genome-editing systems including the Base Editors and the Prime Editors when supplied with target sequences as input. Users have the flexibility to filter resulting guide RNAs based on parameters such as the absence of restriction enzyme cut sites or the lack of paired guide RNAs. The package also facilitates genome-wide exploration for off-targets, offering features to score and rank off-targets, retrieve flanking sequences, and indicate whether the hits are located within exon regions. All detected guide RNAs are annotated with the cumulative scores of the top5 and topN off-targets together with the detailed information such as mismatch sites and restrictuion enzyme cut sites. The package also outputs INDELs and their frequencies for Cas9 targeted sites.

Depends R (>= 3.5.0), BiocGenerics, Biostrings, GenomicFeatures

Imports parallel, data.table, seqinr, S4Vectors (>= 0.9.25), IRanges, BSgenome, hash, methods,reticulate,rhdf5,XVector, DelayedArray, Seqinfo, GenomicRanges, dplyr, keras, mltools, gtools, openxlsx, rio, rlang, stringr

Suggests RUnit, BiocStyle, BSgenome.Hsapiens.UCSC.hg19, TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db, BSgenome.Mmusculus.UCSC.mm10, TxDb.Mmusculus.UCSC.mm10.knownGene, org.Mm.eg.db, lattice, MASS, tensorflow, BSgenome.Hsapiens.UCSC.hg38, BiocFileCache, TxDb.Hsapiens.UCSC.hg38.knownGene, testthat, knitr

License file LICENSE

LazyData yes

biocViews ImmunoOncology, GeneRegulation, SequenceMatching, CRISPR

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/CRISPRseek

git_branch devel

git_last_commit 1c3b409

git_last_commit_date 2025-07-22

Repository Bioconductor 3.22

Date/Publication 2025-07-25

Contents

`CRISPRseek-package` *CRISPRseek: Design of guide RNAs in CRISPR genome-editing systems*

Description

The package encompasses functions to find potential guide RNAs for the CRISPR-based genome-editing systems including the Base Editors and the Prime Editors when supplied with target sequences as input. Users have the flexibility to filter resulting guide RNAs based on parameters such as the absence of restriction enzyme cut sites or the lack of paired guide RNAs. The package also facilitates genome-wide exploration for off-targets, offering features to score and rank off-targets, retrieve flanking sequences, and indicate whether the hits are located within exon regions. All detected guide RNAs are annotated with the cumulative scores of the top5 and topN off-targets together with the detailed information such as mismatch sites and restrictuion enzyme cut sites. The package also outputs INDELs and their frequencies for Cas9 targeted sites.

`annotateOffTargets` *annotate off targets*

Description

Annotate Off targets to indicate whether each one (respectively) is inside an exon or intron, as well as the gene ID if inside the gene.

Usage

```
annotateOffTargets(scores, txdb, orgAnn, ignore.strand = TRUE)
```

Arguments

`scores` A data frame output from getOfftargetScore or filterOfftarget. It contains

- strand - strand of the off-target ((+) for plus and (-) for minus strand)
- chrom - chromosome of the off-target
- chromStart - start position of the off-target
- chromEnd - end position of the off-target
- name - gRNA name
- gRNAPlusPAM - gRNA sequence with PAM sequence concatenated
- OffTargetSequence - the genomic sequence of the off-target
- n.mismatch - number of mismatches between the off-target and the gRNA
- forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707
- score - score of the off-target
- mismatch.distance2PAM - a comma-separated list of all mismatch distances to PAM, e.g., 14,11 means one mismatch is 14 bp away from PAM and the other mismatch is 11 bp away from PAM
- alignment - alignment between gRNA and off-target, e.g., ...G..C..... means that this off-target aligns with gRNA except that G and C are mismatches

- NGG - whether this off-target contains a canonical PAM (1 for yes, 0 for no)
- mean.neighbor.distance.mismatch - mean distance between neighboring mismatches

`txdb` TxDb object. For creating and using TxDb object, please refer to GenomicFeatures package. \ For a list of existing TxDb object, please search for annotation package starting with Txdb at http://www.bioconductor.org/packages/release/BiocViews.html#___An such as

- TxDb.Rnorvegicus.UCSC.rn5.refGene - for rat
- TxDb.Mmusculus.UCSC.mm10.knownGene - for mouse
- TxDb.Hsapiens.UCSC.hg19.knownGene - for human
- TxDb.Dmelanogaster.UCSC.dm3.ensGene - for Drosophila
- TxDb.Celegans.UCSC.ce6.ensGene - for C.elegans

`orgAnn` organism annotation mapping such as org.Hs.egSYMBOL. Which lives in the org.Hs.eg.db package for humans.

`ignore.strand` default to TRUE

Value

a Data Frame with Off Target annotation

Author(s)

Lihua Julie Zhu

References

Lihua Julie Zhu, Benjamin R. Holmes, Neil Aronin and Michael Brodsky. CRISPRseek: a Bioconductor package to identify target-specific guide RNAs for CRISPR-Cas9 genome-editing systems. Plos One Sept 23rd 2014

See Also

offTargetAnalysis

Examples

```
library(CRISPRseek)
#library("BSgenome.Hsapiens.UCSC.hg19")
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)
hitsFile <-  system.file("extdata", "hits.txt", package="CRISPRseek")
hits <- read.table(hitsFile, sep = "\t", header = TRUE,
    stringsAsFactors = FALSE)
featureVectors <- buildFeatureVectorForScoring(hits)
scores <- getOfftargetScore(featureVectors)
outputDir <- getwd()
results <- annotateOffTargets(scores,
    txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
     orgAnn = org.Hs.egSYMBOL)
results
```

`buildFeatureVectorForScoring` *Build feature vectors*

Description

Build feature vectors for calculating scores of off targets

Usage

```
buildFeatureVectorForScoring(
  hits,
  gRNA.size = 20,
  canonical.PAM = "NGG",
  subPAM.position = c(22, 23),
  PAM.size = 3,
  PAM.location = "3prime"
)
```

Arguments

`hits`	A Data frame generated from searchHits, which contains • IsMismatch.posX - Indicator variable indicating whether this position X is a mismatch or not, (1 means yes and 0 means no). X takes on values from 1 to gRNA.size, representing all positions in the guide RNA (gRNA). • strand - strand of the off-target, + for plus and - for minus strand • chrom - chromosome of the off-target • chromStart - start position of the off-target • chromEnd - end position of the off-target • name - gRNA name • gRNAPlusPAM - gRNA sequence with PAM sequence concatenated • OffTargetSequence - the genomic sequence of the off-target • n.mismatch - number of mismatches between the off-target and the gRNA • forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707 • score - Set to 100, and will be calculated in getOfftargetScore
`gRNA.size`	gRNA size. The default is 20
`canonical.PAM`	Canonical PAM. The default is NGG for spCas9, TTTN for Cpf1
`subPAM.position`	The start and end positions of the sub PAM to fetch. Default to 22 and 23 for SP with 20bp gRNA and NGG as preferred PAM
`PAM.size`	Size of PAM, default to 3 for spCas9, 4 for Cpf1
`PAM.location`	PAM location relative to gRNA. For example, default to 3prime for spCas9 PAM. Please set to 5prime for cpf1 PAM since it's PAM is located on the 5 prime end

Value

A data frame with hits plus features used for calculating scores and for generating report, including

- IsMismatch.posX - Indicator variable indicating whether this position X is a mismatch or not, (1 means yes and 0 means no, X = 1 - gRNA.size), representing all positions in the gRNA.
- strand - strand of the off-target, + for plus and - for minus strand
- chrom - chromosome of the off-target
- chromStart - start position of the off-target
- chromEnd - end position of the off-target
- name - gRNA name
- gRNAPlusPAM - gRNA sequence with PAM sequence concatenated
- OffTargetSequence - the genomic sequence of the off-target
- n.mismatch - number of mismatches between the off-target and the gRNA
- forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707
- score - score of the off-target
- mismatch.distance2PAM - a comma-separated list of all mismatches' distances to PAM, e.g., 14,11 means one mismatch is 14 bp away from PAM and the other mismatch is 11 bp away from PAM
- alignment - alignment between gRNA and off-target, e.g., ...G..C..... means that this off-target aligns with gRNA except that G and C are mismatches
- NGG - whether this off-target contains canonical PAM (1 for yes and 0 for no)
- mean.neighbor.distance.mismatch - mean distance between neighboring mismatches

Author(s)

Lihua Julie Zhu

See Also

offTargetAnalysis

Examples

```
hitsFile <-  system.file("extdata", "hits.txt", package = "CRISPRseek")
hits <- read.table(hitsFile, sep= "\t", header = TRUE,
    stringsAsFactors = FALSE)
buildFeatureVectorForScoring(hits)
```

`calculategRNAEfficiency` *Calculate gRNA Efficiency*

Description

Calculate gRNA Efficiency for a given set of sequences and feature weight matrix

Usage

```
calculategRNAEfficiency(
  extendedSequence,
  baseBeforegRNA,
  featureWeightMatrix,
  gRNA.size = 20,
  enable.multicore = FALSE,
  n.cores.max = 6
)
```

Arguments

`extendedSequence`
: Sequences containing gRNA plus PAM plus flanking sequences. Each sequence should be long enough for building features specified in the featureWeightMatrix

`baseBeforegRNA`
: Number of bases before gRNA used for calculating gRNA efficiency, default 4

`featureWeightMatrix`
: a data frame with the first column containing significant features and the second column containing the weight of corresponding features. In the following example, DoenchNBT2014 weight matrix is used. Briefly, features include

  - INTERCEPT
  - GC_LOW - penalty for low GC content in the gRNA sequence
  - GC_HIGH - penalty for high GC content in the gRNA sequence
  - G02 - means G at the second position of the extendedSequence
  - GT02 - means GT di-nucleotides starting at the 2nd position of the extendedSequence

  To understand how is the feature weight matrix is identified, or how to use alternative feature weight matrix file, please see Doench et al., 2014 for details.

`gRNA.size`
: The size of the gRNA, default 20

`enable.multicore`
: Indicate whether enable parallel processing, default FALSE. For super long sequences with lots of gRNAs, suggest set it to TRUE

`n.cores.max`
: Indicating maximum number of cores to use in multi core mode, i.e., parallel processing, default 6. Please set it to 1 to disable multicore processing for small dataset.

Value

DNAStringSet consists of potential gRNAs that can be input to filtergRNAs function directly

Author(s)

Lihua Julie Zhu

References

Doench JG, Hartenian E, Graham DB, Tothova Z, Hegde M, Smith I, Sullender M, Ebert BL, Xavier RJ, Root DE. Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotechnol. 2014 Sep 3. doi: 10.1038 nbt.3026 http://www.broadinstitute.org/rnai/public/analysis-tools/sgrna-design

See Also

offTargetAnalysis

Examples

```
 extendedSequence <- c("TGGATTGTATAATCAGCATGGATTTGGAAC",
 "TCAACGAGGATATTCTCAGGCTTCAGGTCC",
 "GTTACCTGAATTTGACCTGCTCGGAGGTAA",
 "CTTGGTGTGGCTTCCTTTAAGACATGGAGC",
 "CATACAGGCATTGAAGAAGAATTTAGGCCT",
 "AGTACTATACATTTGGCTTAGATTTGGCGG",
 "TTTTCCAGATAGCCGATCTTGGTGTGGCTT",
 "AAGAAGGGAACTATTCGCTGGTGATGGAGT"
)
featureWeightMatrixFile <- system.file("extdata", "DoenchNBT2014.csv",
 package = "CRISPRseek")
featureWeightMatrix <- read.csv(featureWeightMatrixFile, header=TRUE)
calculategRNAEfficiency(extendedSequence, baseBeforegRNA = 4,
 featureWeightMatrix, gRNA.size = 20)
```

---

chromToExclude_default

Default lengthy arguments

Description

This contains a list of long constant values used as defaults in many function.

Usage

```
chromToExclude_default
```

Format

A character string.

Examples

```
REpatternFile_default  # Display the default value for REpatternFile.

chromToExclude_default
```

compare2Sequences *Compare two input sequences/sequence sets for possible guide RNAs (gRNAs)*

Description

Generate all possible guide RNAs (gRNAs) for two input sequences, or two sets of sequences, and generate scores for potential off-targets in the other sequence.

Usage

```
compare2Sequences(
  inputFile1Path = NULL,
  inputFile2Path = NULL,
  inputNames = c("Seq1", "Seq2"),
  format = c("fasta", "fasta"),
  header = FALSE,
  findgRNAsWithREcutOnly = FALSE,
  searchDirection = c("both", "1to2", "2to1"),
  BSgenomeName = NULL,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8,
  editingWindow.offtargets = 4:8,
  REpatternFile = REpatternFile_default(),
  minREpatternSize = 6,
  findgRNAs = c(TRUE, TRUE),
  removegRNADetails = c(FALSE, FALSE),
  exportAllgRNAs = c("no", "all", "fasta", "genbank"),
  annotatePaired = FALSE,
  overlap.gRNA.positions = c(17, 18),
  findPairedgRNAOnly = FALSE,
  min.gap = 0,
  max.gap = 20,
  gRNA.name.prefix = "_gR",
  PAM.size = 3,
  gRNA.size = 20,
  PAM = "NGG",
  PAM.pattern = "NNG$|NGN$",
  allowed.mismatch.PAM = 1,
  max.mismatch = 3,
  outputDir = NULL,
  upstream = 0,
  downstream = 0,
  weights = weights_default,
  overwrite = FALSE,
  baseBeforegRNA = 4,
  baseAfterPAM = 3,
  featureWeightMatrixFile = featureWeightMatrixFile_default(),
  foldgRNAs = FALSE,
  gRNA.backbone = gRNA.backbone_default,
```

```
  temperature = 37,
  scoring.method = c("Hsu-Zhang", "CFDscore"),
  subPAM.activity = subPAM.activity_default,
  subPAM.position = c(22, 23),
  PAM.location = "3prime",
 rule.set = c("Root_RuleSet1_2014", "Root_RuleSet2_2016", "CRISPRscan", "DeepCpf1"),
  mismatch.activity.file = mismatch.activity.file_default()
)
```

Arguments

inputFile1Path	Sequence input file 1 path that contains one of the two sequences to be searched for potential gRNAs. It can also be a DNAStringSet object with names field set. Please see examples below.
inputFile2Path	Sequence input file 2 path that contains one of the two sequences to be searched for potential gRNAs. It can also be a DNAStringSet object with names field set. Please see examples below.
inputNames	Name of the input sequences when inputFile1Path and inputFile2Path are DNAStringSet instead of file path
format	Format of the input files, fasta, fastq and bed format are supported, default fasta
header	Indicate whether the input file contains header, default FALSE, only applies to bed format
findgRNAsWithREcutOnly	Indicate whether to find gRNAs overlap with restriction enzyme recognition pattern
searchDirection	Indicate whether perfrom gRNA in both sequences and off-target search against each other (both) or search gRNA in input1 and off-target analysis in input2 (1to2), or vice versa (2to1)
BSgenomeName	BSgenome object. Please refer to available.genomes in BSgenome package. For example, BSgenome.Hsapiens.UCSC.hg19 for hg19, BSgenome.Mmusculus.UCSC.mm10 for mm10, BSgenome.Celegans.UCSC.ce6 for ce6, BSgenome.Rnorvegicus.UCSC.rn5 for rn5, BSgenome.Drerio.UCSC.danRer7 for Zv9, and BSgenome.Dmelanogaster.UCSC.dm3 for dm3
baseEditing	Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.
targetBase	Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.
editingWindow	Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window.
editingWindow.offtargets	Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window to consider for the offtargets search only, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window, or you would like to include offtargets with the target base in a larger editing window.
REpatternFile	File path containing restriction enzyme cut patters

`minREpatternSize`
: Minimum restriction enzyme recognition pattern length required for the enzyme pattern to be searched for, default 6

`findgRNAs`
: Indicate whether to find gRNAs from the sequences in the input file or skip the step of finding gRNAs, default TRUE for both input sequences. Set it to FALSE if the input file contains user selected gRNAs plus PAM already.

`removegRNADetails`
: Indicate whether to remove the detailed gRNA information such as efficacy file and restriction enzyme cut sites, default false for both input sequences. Set it to TRUE if the input file contains the user selected gRNAs plus PAM already.

`exportAllgRNAs`
: Indicate whether to output all potential gRNAs to a file in fasta format, genbank format or both. Default to no.

`annotatePaired`
: Indicate whether to output paired information, default to FALSE

`overlap.gRNA.positions`
: The required overlap positions of gRNA and restriction enzyme cut site, default 17 and 18

`findPairedgRNAOnly`
: Choose whether to only search for paired gRNAs in such an orientation that the first one is on minus strand called reverse gRNA and the second one is on plus strand called forward gRNA. TRUE or FALSE, default FALSE

`min.gap`
: Minimum distance between two oppositely oriented gRNAs to be valid paired gRNAs. Default 0

`max.gap`
: Maximum distance between two oppositely oriented gRNAs to be valid paired gRNAs. Default 20

`gRNA.name.prefix`
: The prefix used when assign name to found gRNAs, default _gR, short for guided RNA.

`PAM.size`
: PAM length, default 3

`gRNA.size`
: The size of the gRNA, default 20

`PAM`
: PAM sequence after the gRNA, default NGG

`PAM.pattern`
: Regular expression of PAM, default NNG or NGN for spCas9. For cpf1, ^TTTN since it is a 5 prime PAM sequence

`allowed.mismatch.PAM`
: Maximum number of mismatches allowed to the PAM sequence, default to 1 for PAM.pattern NNG or NGN PAM

`max.mismatch`
: Maximum mismatch allowed to search the off targets in the other sequence, default 3

`outputDir`
: the directory where the sequence comparison results will be written to

`upstream`
: upstream offset from the bed input starts to search for gRNA and/or offtargets, default 0

`downstream`
: downstream offset from the bed input ends to search for gRNA and/or offtargets, default 0

`weights`
: numeric vector size of gRNA length, default c(0, 0, 0.014, 0, 0, 0.395, 0.317, 0, 0.389, 0.079, 0.445, 0.508, 0.613, 0.851, 0.732, 0.828, 0.615, 0.804, 0.685, 0.583) which is used in Hsu et al., 2013 cited in the reference section

`overwrite`
: overwrite the existing files in the output directory or not, default TRUE

`baseBeforegRNA` Number of bases before gRNA used for calculating gRNA efficiency, default 4 Please note, for PAM located on the 5 prime, need to specify the number of bases before the PAM sequence plus PAM size.

`baseAfterPAM` Number of bases after PAM used for calculating gRNA efficiency, default 3 for spCas9 Please note, for PAM located on the 5 prime, need to include the length of the gRNA plus the extended sequence on the 3 prime

`featureWeightMatrixFile`
Feature weight matrix file used for calculating gRNA efficiency. By default DoenchNBT2014 weight matrix is used. To use alternative weight matrix file, please input a csv file with first column containing significant features and the second column containing the corresponding weights for the features. Please see Doench et al., 2014 for details.

`foldgRNAs` Default FALSE. If set to TRUE, summary file will contain minimum free energy of the secondary structure of gRNA with gRNA backbone from GeneRfold package provided that GeneRfold package has been installed.

`gRNA.backbone` gRNA backbone constant region sequence. Default to the sequence in Sp gRNA backbone.

`temperature` temperature in celsius. Default to 37 celsius.

`scoring.method` Indicates which method to use for offtarget cleavage rate estimation, currently two methods are supported, Hsu-Zhang and CFDscore

`subPAM.activity`
Applicable only when scoring.method is set to CFDscore A hash to represent the cleavage rate for each alternative sub PAM sequence relative to preferred PAM sequence

`subPAM.position`
Applicable only when scoring.method is set to CFDscore The start and end positions of the sub PAM. Default to 22 and 23 for SP with 20bp gRNA and NGG as preferred PAM

`PAM.location` PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime (3prime) while cpf1 PAM is located on the 5 prime (5prime)

`rule.set` Specify a rule set scoring system for calculating gRNA efficacy. Please note that Root_RuleSet2_2016 requires the following python packages with specified verion and python 2.7. 1. scikit-learn 0.16.1 2. pickle 3. pandas 4. numpy 5. scipy

`mismatch.activity.file`
Applicable only when scoring.method is set to CFDscore A comma separated (csv) file containing the cleavage rates for all possible types of single nucleotide mismatche at each position of the gRNA. By default, using the supplemental Table 19 from Doench et al., Nature Biotechnology 2016

Value

Return a data frame with all potential gRNAs from both sequences. In addition, a tab-delimited file 'scoresFor2InputSequences.xlsx' is also saved in the 'outputDir', sorted by 'scoreDiff' descending.

- name - name of the gRNA
- gRNAPlusPAM - gRNA plus PAM sequence
- targetInSeq1 - target/off-target sequence including PAM in the 1st input sequence file
- targetInSeq2 - target/off-target sequence including PAM in the 2nd input sequence file

- guideAlignment2Offtarget - alignment of gRNA to the other input sequence (off-target sequence)
- offTargetStrand - strand of the other sequence (off-target sequence) the gRNA aligns to
- scoreForSeq1 - score for the target sequence in the 1st input sequence file
- scoreForSeq2 - score for the target sequence in the 2nd input sequence file
- mismatch.distance2PAM - distances of mismatch to PAM, e.g., 14 means the mismatch is 14 bp away from PAM
- n.mismatch - number of mismatches between the off-target and the gRNA
- targetSeqName - the name of the input sequence where the target sequence is located
- scoreDiff - scoreForSeq1 - scoreForSeq2
- bracket.notation - folded gRNA in bracket notation
- mfe.sgRNA - minimum free energy of sgRNA
- mfe.diff - mfe.sgRNA - mfe.backbone
- mfe.backbone - minimum free energy of the gRNA backbone by itself

Author(s)

Lihua Julie Zhu

References

Patrick D Hsu, David A Scott, Joshua A Weinstein, F Ann Ran, Silvana Konermann, Vineeta Agarwala, Yinqing Li, Eli J Fine, Xuebing Wu, Ophir Shalem, Thomas J Cradick, Luciano A Marraffini, Gang Bao & Feng Zhang (2013) DNA targeting specificity of rNA-guided Cas9 nucleases. Nature Biotechnology 31:827-834

See Also

CRISPRseek

Examples

```
    library(CRISPRseek)
    inputFile1Path <- system.file("extdata", "rs362331T.fa",
            package = "CRISPRseek")
    inputFile2Path <- system.file("extdata", "rs362331C.fa",
            package = "CRISPRseek")
    REpatternFile <- system.file("extdata", "NEBenzymes.fa",
            package = "CRISPRseek")
    outputDir <- tempdir()
    seqs <- compare2Sequences(inputFile1Path, inputFile2Path,
        outputDir = outputDir,
        REpatternFile = REpatternFile, overwrite = TRUE)

    seqs2 <- compare2Sequences(inputFile1Path, inputFile2Path,
               inputNames=c("Seq1", "Seq2"),
               scoring.method = "CFDscore",
               outputDir = outputDir,
               overwrite = TRUE, baseEditing = TRUE)

    inputFile1Path <-
DNAStringSet(
```

```
"TAATATTTTAAAATCGGTGACGTGGGCCCAAAACGAGTGCAGTTCCAAAGGCACCCACCTGTGGCAG"
)
    ## when set inputFile1Path to a DNAStringSet object, it is important
    ## to call names
    names(inputFile1Path) <- "seq1"

    inputFile2Path <-
DNAStringSet(
"TAATATTTTAAAATCGGTGACGTGGGCCCAAAACGAGTGCAGTTCCAAAGGCACCCACCTGTGGCAG"
)
     ## when set inputFile2Path to a DNAStringSet object, it is important
    ## to call names

    names(inputFile2Path) <- "seq2"

    seqs <- compare2Sequences(inputFile1Path, inputFile2Path,
          inputNames=c("Seq1", "Seq2"),
          scoring.method = "CFDscore",
          outputDir = outputDir,
          overwrite = TRUE)

    seqs2 <- compare2Sequences(inputFile1Path, inputFile2Path,
               inputNames=c("Seq1", "Seq2"),
               scoring.method = "CFDscore",
               outputDir = outputDir,
               overwrite = TRUE, baseEditing = TRUE)
```

`deepCpf1` *DeepCpf1 Algorithm for predicting CRISPR-Cpf1 gRNA Efficacy*

Description

DeepCpf1 algorithm from https://doi.org/10.1038/nbt.4061, which takes in 34 bp target sequences with/without chromatin accessibility information and returns predicted CRISPR-Cpf1 gRNA efficacy for each input sequence.

Usage

```
deepCpf1(extendedSequence = NULL, chrom_acc = NULL)
```

Arguments

`extendedSequence`	Sequences containing gRNA plus PAM plus flanking sequences. Each sequence should be 34 bp long as specified by http://deepcrispr.info/, i.e., 4bp before the 5' PAM, 4bp PAM, 20bp gRNA, and 6bp after 3' of gRNA.
`chrom_acc`	Optional binary variable indicating chromatin accessibility information with 1 indicating accessible and 0 not accessible.

Details

Having chromatin accessibility information will aid in the accuracy of the scores, but one can still get accurate scoring with only the 34 bp target sequences.

Value

a numeric vector with prediced CRISPR-Cpf1 gRNA efficacy taking into account chromatin accessibility information if accessibility information is provided

Author(s)

Paul Scemama and Lihua Julie Zhu

References

Kim et al., Deep learning improves prediction of CRISPR–Cpf1 guide RNA activityNat Biotechnol 36, 239–241 (2018). https://doi.org/10.1038/nbt.4061

Examples

```
library(keras)
library(mltools)
library(dplyr)
library(data.table)

use_implementation("tensorflow")

extendedSequence <- c('GTTATTTGAGCAATGCCACTTAATAAACATGTAA',
 'TGACTTTGAATGGAGTCGTGAGCGCAAGAACGCT',
 'GTTATTTGAGCAATGCCACTTAATAAACATGTAA',
 'TGACTTTGAATGGAGTCGTGAGCGCAAGAACGCT')
chrom_acc <- c(0, 1, 0, 1)

if (interactive()) {
 deepCpf1(extendedSequence = extendedSequence, chrom_acc = chrom_acc)
}
```

---

featureWeightMatrixFile_default

featureWeightMatrixFile_default

Description

Default value for featureWeightMatrixFile, use featureWeightMatrixFile() to show its value.

Usage

```
featureWeightMatrixFile_default()
```

`filtergRNAs` *Filter gRNAs*

Description

Filter gRNAs containing restriction enzyme cut site

Usage

```
filtergRNAs(
  all.gRNAs = NULL,
  pairOutputFile = NULL,
  findgRNAsWithREcutOnly = FALSE,
  REpatternFile = REpatternFile_default(),
  format = "fasta",
  minREpatternSize = 4,
  overlap.gRNA.positions = c(17, 18),
  overlap.allpos = TRUE
)
```

Arguments

`all.gRNAs`	gRNAs as DNAStringSet, such as the output from findgRNAs
`pairOutputFile`	File path with paired gRNAs
`findgRNAsWithREcutOnly`	Indicate whether to find gRNAs overlap with restriction enzyme recognition pattern
`REpatternFile`	File path containing restriction enzyme cut patters
`format`	Format of the REpatternFile, default as fasta
`minREpatternSize`	Minimum restriction enzyme recognition pattern length required for the enzyme pattern to be searched for, default 4
`overlap.gRNA.positions`	The required overlap positions of gRNA and restriction enzyme cut site, default 17 and 18
`overlap.allpos`	Default TRUE, meaning that only gRNAs overlap with all the positions are retained FALSE, meaning that gRNAs overlap with one or both of the positions are retained

Value

`gRNAs.withRE`	gRNAs as DNAStringSet that passed the filter criteria
`gRNAREcutDetails`	a data frame that contains a set of gRNAs annotated with restriction enzyme cut details

Author(s)

Lihua Julie Zhu

See Also

offTargetAnalysis

Examples

```
all.gRNAs <- findgRNAs(
  inputFilePath = system.file("extdata", "inputseq.fa",
  package = "CRISPRseek"),
  pairOutputFile = "testpairedgRNAs.xlsx",
  findPairedgRNAOnly = TRUE)

gRNAs.RE <- filtergRNAs(
  all.gRNAs = all.gRNAs,
  pairOutputFile = "testpairedgRNAs.xlsx",
  minREpatternSize = 6,
  REpatternFile = system.file("extdata", "NEBenzymes.fa",
                              package = "CRISPRseek"),
  overlap.allpos = TRUE)

gRNAs  <- gRNAs.RE$gRNAs.withRE
restriction.enzyme.cut.sites <- gRNAs.RE$gRNAREcutDetails
```

---

filterOffTarget *filter off-targets and generate reports.*

Description

filter off-targets that meet the criteria set by users such as minimum score, topN. In addition, off target was annotated with flank sequence, gRNA cleavage efficiency and whether it is inside an exon or not if fetchSequence is set to TRUE and annotateExon is set to TRUE

Usage

```
filterOffTarget(
  scores = NULL,
  min.score = 0.01,
  topN = 200,
  topN.OfftargetTotalScore = 10,
  annotateExon = TRUE,
  txdb = NULL,
  orgAnn = NULL,
  ignore.strand = TRUE,
  outputDir = NULL,
  oneFilePergRNA = FALSE,
  fetchSequence = TRUE,
  upstream = 200,
  downstream = 200,
  BSgenomeName = NULL,
  genomeSeqFile = NULL,
  baseBeforegRNA = 4,
  baseAfterPAM = 3,
```

```
  gRNA.size = 20,
  PAM.location = "3prime",
  PAM.size = 3,
  featureWeightMatrixFile = featureWeightMatrixFile_default(),
 rule.set = c("Root_RuleSet1_2014", "Root_RuleSet2_2016", "CRISPRscan", "DeepCpf1"),
  chrom_acc = NULL,
  calculategRNAefficacyForOfftargets = TRUE
)
```

Arguments

scores	a data frame output from getOfftargetScore. It contains
min.score	minimum score of an off target to included in the final output, default 0.5
topN	top N off targets to be included in the final output, default 100
topN.OfftargetTotalScore	top N off target used to calculate the total off target score, default 10
annotateExon	Choose whether or not to indicate whether the off target is inside an exon or not, default TRUE
txdb	TxDb object, for creating and using TxDb object, please refer to GenomicFeatures package. For a list of existing TxDb object, please search for annotation package starting with Txdb at http://www.bioconductor.org/packages/release/BiocViews.html#___An such as TxDb.Rnorvegicus.UCSC.rn5.refGene for rat, TxDb.Mmusculus.UCSC.mm10.knownGene for mouse, TxDb.Hsapiens.UCSC.hg19.knownGene for human, TxDb.Dmelanogaster.UCSC.dm3.en for Drosophila and TxDb.Celegans.UCSC.ce6.ensGene for C.elegans
orgAnn	organism annotation mapping such as org.Hs.egSYMBOL in org.Hs.eg.db package for human
ignore.strand	default to TRUE
outputDir	the directory where the off target analysis and reports will be written to
oneFilePergRNA	write to one file for each gRNA or not, default to FALSE
fetchSequence	Fetch flank sequence of off target or not, default TRUE
upstream	upstream offset from the off target start, default 200
downstream	downstream offset from the off target end, default 200
BSgenomeName	BSgenome object. Please refer to available.genomes in BSgenome package. For example, • BSgenome.Hsapiens.UCSC.hg19 - for hg19 • BSgenome.Mmusculus.UCSC.mm10 - for mm10 • BSgenome.Celegans.UCSC.ce6 - for ce6 • BSgenome.Rnorvegicus.UCSC.rn5 - for rn5 • BSgenome.Dmelanogaster.UCSC.dm3 - for dm3
genomeSeqFile	Other than BSgenomeName, a custome FASTA file can be supplied, if set, overwrites BSgenomeName.
baseBeforegRNA	Number of bases before gRNA used for calculating gRNA efficiency, default 4
baseAfterPAM	Number of bases after PAM used for calculating gRNA efficiency, default 3
gRNA.size	The size of the gRNA, default 20 for spCas9
PAM.location	PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime

`PAM.size` PAM length, default 3 for spCas9

`featureWeightMatrixFile`
Feature weight matrix file used for calculating gRNA efficiency. By default DoenchNBT2014 weight matrix is used. To use alternative weight matrix file, please input a csv file with first column containing significant features and the second column containing the corresponding weights for the features. Please see Doench et al., 2014 for details.

`rule.set` Specify a rule set scoring system for calculating gRNA efficacy.

`chrom_acc` Optional binary variable indicating chromatin accessibility information with 1 indicating accessible and 0 not accessible.

`calculategRNAefficacyForOfftargets`
Default to TRUE to output gRNA efficacy for offtargets as well as ontargets. Set it to FALSE if only need gRNA efficacy calculated for ontargets only to speed up the analysis. Please refer to https://support.bioconductor.org/p/133538/#133661 for potential use cases of offtarget efficacies.

Value

A data frame with details of off-targets for the given gRNA.

- strand - strand of the off-target, + for plus and - for minus strand
- chrom - chromosome of the off-target
- chromStart - start position of the off-target
- chromEnd - end position of the off-target
- name - gRNA name
- gRNAPlusPAM - gRNA sequence with PAM sequence concatenated
- OffTargetSequence - the genomic sequence of the off-target
- n.mismatch - number of mismatches between the off-target and the gRNA
- forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707
- score - score of the off-target
- mismatch.distance2PAM - comma-separated distances of all mismatches to PAM, e.g., 14,11 means one mismatch is 14 bp away from PAM and the other mismatch is 11 bp away from PAM
- alignment - alignment between gRNA and off-target, e.g., ...G..C..... means that this off-target aligns with gRNA except that G and C are mismatches
- NGG - whether this off-target contains canonical PAM (1 for yes, 0 for no)
- mean.neighbor.distance.mismatch - mean distance between neighboring mismatches

- offtargets - a data frame with off-target analysis results
- summary - a data frame with summary of the off-target analysis results

Author(s)

Lihua Julie Zhu

References

Doench JG, Hartenian E, Graham DB, Tothova Z, Hegde M, Smith I, Sullender M, Ebert BL, Xavier RJ, Root DE. Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotechnol. 2014 Sep 3. doi: 10.1038 nbt.3026 Lihua Julie Zhu, Benjamin R. Holmes, Neil Aronin and Michael Brodsky. CRISPRseek: a Bioconductor package to identify target-specific guide RNAs for CRISPR-Cas9 genome-editing systems. Plos One Sept 23rd 2014

See Also

offTargetAnalysis

Examples

```
library(CRISPRseek)
library(BSgenome.Hsapiens.UCSC.hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)

hitsFile <-  system.file("extdata", "hits.txt", package = "CRISPRseek")
hits <- read.table(hitsFile, sep = "\t",
                   header = TRUE,
                   stringsAsFactors = FALSE)
featureVectors <- buildFeatureVectorForScoring(hits)
scores <- getOfftargetScore(featureVectors)

outputDir <- tempdir()
results <- filterOffTarget(scores,
                           BSgenomeName = Hsapiens,
                           txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                           orgAnn = org.Hs.egSYMBOL,
                           outputDir = outputDir,
                           min.score = 0.1,
                           topN = 10,
                           topN.OfftargetTotalScore = 5)
results$offtargets
results$summary
```

`findgRNAs` *Find potential gRNAs*

Description

Find potential gRNAs for an input file containing sequences in fasta format

Usage

```
findgRNAs(
  inputFilePath = NULL,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8,
  format = "fasta",
```

```
  PAM = "NGG",
  PAM.size = 3,
  findPairedgRNAOnly = FALSE,
  annotatePaired = TRUE,
  paired.orientation = c("PAMout", "PAMin"),
  enable.multicore = FALSE,
  n.cores.max = 6,
  gRNA.pattern = NULL,
  gRNA.size = 20,
  overlap.gRNA.positions = c(17, 18),
  primeEditing = FALSE,
  PBS.length = 13L,
  RT.template.length = 8:28,
  RT.template.pattern = "D$",
  corrected.seq = NULL,
  targeted.seq.length.change = NULL,
  bp.after.target.end = 15L,
  target.start = NULL,
  target.end = NULL,
  primeEditingPaired.output = "pairedgRNAsForPE.xls",
  min.gap = 0,
  max.gap = 20,
  pairOutputFile = NULL,
  name.prefix = NULL,
  featureWeightMatrixFile = featureWeightMatrixFile_default(),
  baseBeforegRNA = 4,
  baseAfterPAM = 3,
  calculategRNAEfficacy = FALSE,
  efficacyFile = NULL,
  PAM.location = "3prime",
 rule.set = c("Root_RuleSet1_2014", "Root_RuleSet2_2016", "CRISPRscan", "DeepCpf1"),
  chrom_acc = NULL
)
```

Arguments

inputFilePath	Sequence input file path or a DNAStringSet object that contains sequences to be searched for potential gRNAs
baseEditing	Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.
targetBase	Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.
editingWindow	Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window.
format	Format of the input file, fasta and fastq are supported, default fasta
PAM	protospacer-adjacent motif (PAM) sequence near the gRNA, default NGG
PAM.size	PAM length, default 3

`findPairedgRNAOnly`
: Choose whether to only search for paired gRNAs in such an orientation that the first one is on minus strand called reverse gRNA and the second one is on plus strand called forward gRNA. TRUE or FALSE, default FALSE

`annotatePaired`
: Indicate whether to output paired information, default TRUE

`paired.orientation`
: PAMin orientation means the two adjacent PAMs on the sense and antisense strands face inwards towards each other like N21GG and CCN21 whereas PAMout orientation means they face away from each other like CCN21 and N21GG

`enable.multicore`
: Indicate whether enable parallel processing, default FALSE. For super long sequences with lots of gRNAs, suggest set it to TRUE

`n.cores.max`
: Indicating maximum number of cores to use in multi core mode, i.e., parallel processing, default 6. Please set it to 1 to disable multicore processing for small dataset.

`gRNA.pattern`
: Regular expression or IUPAC Extended Genetic Alphabet to represent gRNA pattern, default is no restriction. To specify that the gRNA must start with GG for example, then set it to ^GG. Please see help(translatePattern) for a list of IUPAC Extended Genetic Alphabet.

`gRNA.size`
: The size of the gRNA, default 20

`overlap.gRNA.positions`
: The required overlap positions of gRNA and restriction enzyme cut site, default 17 and 18. For Cpf1,you may set it to 19 and 23.

`primeEditing`
: Indicate whether to design gRNAs for prime editing. Default to FALSE. If true, please set PBS.length, RT.template.length, RT.template.pattern, targeted.seq.length.change, bp.after.target.end, target.start, and target.end accordingly

`PBS.length`
: Applicable only when primeEditing is set to TRUE. It is used to specify the number of bases to ouput for primer binding site.

`RT.template.length`
: Applicable only when primeEditing is set to TRUE. It is used to specify the number of bases required for RT template, default to 8 to 18. Please increase the length if the edit is large insertion. Only gRNAs with calculated RT.template.length falling into the specified range will be in the output. It is calculated as the following. RT.template.length = target.start – cut.start + (target.end - target.start) + targeted.seq.length.change + bp.after.target.end

`RT.template.pattern`
: Applicable only when primeEditing is set to TRUE. It is used to specify the RT template sequence pattern, default to not ending with C according to https://doi.org/10.1038/s41586-019-1711-4

`corrected.seq`
: Applicable only when primeEditing is set to TRUE. It is used to specify the mutated or inserted sequences after successful editing.

`targeted.seq.length.change`
: Applicable only when primeEditing is set to TRUE. It is used to specify the number of targeted sequence length change. Please set it to 0 for base changes, positive numbers for insersion, and negative number for deletion. For example, 10 means that the corrected sequence will have 10bp insertion, -10 means that the corrected sequence will have 10bp deletion, and 0 means only bases have been changed and the sequence length remains the same

`bp.after.target.end`
: Applicable only when primeEditing is set to TRUE. It is used to specify the number of bases to add after the target change end site as part of RT template. Please refer to RT.template.length for how this parameter influences the RT.template.length calculation which is used as a filtering criteria in pregRNA selection.

`target.start`
: Applicable only when primeEditing is set to TRUE. It is used to specify the start location in the input sequence to make changes, which will be used to obtain the RT template sequence. Please also refer to RT.template.length for how this parameter influences the RT.template.length calculation which is used as a filtering criteria in pregRNA selection.

`target.end`
: Applicable only when primeEditing is set to TRUE. It is used to specify the end location in the input sequnence to make changes, which will be used to obtain the RT template sequence. Please also refer to RT.template.length for how this parameter influences the RT.template.length calculation which is used as a filtering criteria in pregRNA selection.

`primeEditingPaired.output`
: Applicable only when primeEditing is set to TRUE. It is used to specify the file path to save pegRNA and the second gRNA with PBS, RT.template, gRNA sequences, default pairedgRNAsForPE.xls

`min.gap`
: Minimum distance between two oppositely oriented gRNAs to be valid paired gRNAs. Default 0

`max.gap`
: Maximum distance between two oppositely oriented gRNAs to be valid paired gRNAs. Default 20

`pairOutputFile`
: The output file for writing paired gRNA information to

`name.prefix`
: The prefix used when assign name to found gRNAs, default gRNA, short for guided RNA.

`featureWeightMatrixFile`
: Feature weight matrix file used for calculating gRNA efficiency. By default DoenchNBT2014 weight matrix is used. To use alternative weight matrix file, please input a csv file with first column containing significant features and the second column containing the corresponding weights for the features. Please see Doench et al., 2014 for details.

`baseBeforegRNA`
: Number of bases before gRNA used for calculating gRNA efficiency, default 4 for spCas9 Please note, for PAM located on the 5 prime, need to specify the number of bases before the PAM sequence plus PAM size.

`baseAfterPAM`
: Number of bases after PAM used for calculating gRNA efficiency, default 3 for spCas9 Please note, for PAM located on the 5 prime, need to include the length of the gRNA plus the extended sequence on the 3 prime

`calculategRNAEfficacy`
: Default to FALSE, not to calculate gRNA efficacy

`efficacyFile`
: File path to write gRNA efficacies

`PAM.location`
: PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime

`rule.set`
: Specify a rule set scoring system for calculating gRNA efficacy. Please note that if specifying DeepCpf1, please specify other parameters accordingly for CRISPR-Cpf1 gRNAs.

`chrom_acc`
: Optional binary variable indicating chromatin accessibility information with 1 indicating accessible and 0 not accessible.

Details

If users already has a fasta file that contains a set of potential gRNAs, then users can call filergRNAs directly although the easiest way is to call the one-stop-shopping function OffTargetAnalysis with findgRNAs set to FALSE.

Value

DNAStringSet consists of potential gRNAs that can be input to filtergRNAs function directly

Note

If the input sequence file contains multiple >300 bp sequences, suggest create one input file for each sequence and run the OffTargetAnalysis separately.

Author(s)

Lihua Julie Zhu

See Also

offTargetAnalysis

Examples

```
# Example1: DNAStringSet as input, only output paired gRNAs
inputSeq <-  DNAStringSet(paste0("CCAGTTTGTGGATCCTGCTCTGGTGTC",
                                 "CTCCACACCAGAATCAGGGATCGAAAA",
                                 "CTCATCAGTCGATGCGAGTCATCTAAA",
                                 "TTCCGATCAATTTCACACTTTAAACG"))
findgRNAs(inputFilePath =  inputSeq,
          findPairedgRNAOnly = TRUE,
          pairOutputFile = "test_findgRNAs1.xlsx",
          PAM.size = 3L,
          gRNA.size = 20L,
          overlap.gRNA.positions = c(17L,18L),
          PBS.length = 15,
          corrected.seq = "T",
          RT.template.pattern = "D$",
          RT.template.length = 8:30,
          targeted.seq.length.change = 0,
          bp.after.target.end = 15,
          target.start = 46,
          target.end = 46,
          paired.orientation = "PAMin",
          min.gap = 20,
          max.gap = 90,
          primeEditing = TRUE)

# Example2: FASTA as input, only output paired gRNAs
findgRNAs(inputFilePath = system.file("extdata",
                                      "inputseq.fa",
                                      package = "CRISPRseek"),
          findPairedgRNAOnly = TRUE,
          pairOutputFile = "test_findgRNAs2.xlsx")

# Example3: predict gRNA efficacy using CRISPRscan
```

```
featureWeightMatrixFile <- system.file("extdata",
                                       "Morenos-Mateo.csv",
                                       package = "CRISPRseek")
findgRNAs(inputFilePath = system.file("extdata",
                                      "testCRISPRscan.fa",
                                      package = "CRISPRseek"),
          pairOutputFile = "test_findgRNAs3.xlsx",
          findPairedgRNAOnly = FALSE,
          calculategRNAEfficacy= TRUE,
          rule.set = "CRISPRscan",
          baseBeforegRNA = 6,
          baseAfterPAM = 6,
          featureWeightMatrixFile = featureWeightMatrixFile,
          efficacyFile = "testCRISPRscanEfficacy.xlsx")

# Example 4: predict gRNA efficacy using DeepCpf1
# Note: that these examples may fail during build/check on Bioconductor when
# running on MacOS Monterey due to compatibility issues with keras. To avoid
# errors, wrap the code in `if (interactive)`.
if (interactive()) {
  findgRNAs(inputFilePath = system.file("extdata",
                                        "cpf1.fa",
                                        package = "CRISPRseek"),
            findPairedgRNAOnly = FALSE,
            pairOutputFile = "test_findgRNAs_cpf1.xlsx",
            PAM = "TTTN",
            PAM.location = "5prime",
            PAM.size = 4,
            overlap.gRNA.positions = c(19, 23),
            baseBeforegRNA = 8,
            baseAfterPAM = 26,
            calculategRNAEfficacy = TRUE,
            rule.set = "DeepCpf1",
            efficacyFile = "testcpf1Efficacy.xlsx")

  findgRNAs(inputFilePath = system.file("extdata",
                                        "cpf1.fa",
                                        package = "CRISPRseek"),
            findPairedgRNAOnly = FALSE,
            pairOutputFile = "test_findgRNAs_cpf1.xlsx",
            PAM = "TTTN",
            PAM.location = "5prime",
            PAM.size = 4,
            overlap.gRNA.positions = c(19, 23),
            baseBeforegRNA = 8,
            baseAfterPAM = 26,
            calculategRNAEfficacy= TRUE,
            rule.set = "DeepCpf1",
            efficacyFile = "testcpf1Efficacy.xlsx",
            baseEditing = TRUE,
            editingWindow = 20,
            targetBase = "X")

  findgRNAs(inputFilePath = system.file("extdata",
                                        "cpf1.fa",
                                        package = "CRISPRseek"),
            findPairedgRNAOnly = FALSE,
```

```
            pairOutputFile = "test_findgRNAs_cpf1.xlsx",
            PAM = "TTTN",
            PAM.location = "5prime",
            PAM.size = 4,
            overlap.gRNA.positions = c(19, 23),
            baseBeforegRNA = 8,
            baseAfterPAM = 26,
            calculategRNAEfficacy = TRUE,
            rule.set = "DeepCpf1",
            efficacyFile = "testcpf1Efficacy.xlsx",
            baseEditing = TRUE,
            editingWindow = 20,
            targetBase = "C")
}
```

`getOfftargetScore` *Calculate score for each off target*

Description

Calculate score for each off target with given feature vectors and weights vector

Usage

```
getOfftargetScore(
  featureVectors,
 weights = c(0, 0, 0.014, 0, 0, 0.395, 0.317, 0, 0.389, 0.079, 0.445, 0.508, 0.613,
    0.851, 0.732, 0.828, 0.615, 0.804, 0.685, 0.583)
)
```

Arguments

`featureVectors`	a data frame generated from buildFeatureVectorForScoring. It contains
`weights`	a numeric vector size of gRNA length, default c(0, 0, 0.014, 0, 0, 0.395, 0.317, 0, 0.389, 0.079, 0.445, 0.508, 0.613, 0.851, 0.732, 0.828, 0.615, 0.804, 0.685, 0.583) which is used in Hsu et al., 2013 cited in the reference section

Details

score is calculated using the weights and algorithm by Hsu et al., 2013 cited in the reference section

Value

A data frame containing details of off-targets for the given gRNA.

- IsMismatch.posX - Indicator variable indicating whether this position X is a mismatch or not, (1 means yes and 0 means no). X takes on values from 1 to gRNA.size, representing all positions in the guide RNA (gRNA).
- strand - strand of the off-target, + for plus and - for minus strand
- chrom - chromosome of the off-target
- chromStart - start position of the off-target

- chromEnd - end position of the off-target
- name - gRNA name
- gRNAPlusPAM - gRNA sequence with PAM sequence concatenated
- OffTargetSequence - the genomic sequence of the off-target
- n.mismatch - number of mismatches between the off-target and the gRNA
- forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707
- score - score of the off-target
- mismatch.distance2PAM - comma-separated distances of all mismatches to PAM, e.g., 14,11 means one mismatch is 14 bp away from PAM and the other mismatch is 11 bp away from PAM
- alignment - alignment between gRNA and off-target, e.g., ...G..C..... means that this off-target aligns with gRNA except that G and C are mismatches
- NGG - whether this off-target contains canonical PAM (1 for yes, 0 for no)
- mean.neighbor.distance.mismatch - mean distance between neighboring mismatches

A data frame containing details of off-targets for the given gRNA.

- strand - strand of the match, + for plus and - for minus strand
- chrom - chromosome of the off-target
- chromStart - start position of the off-target
- chromEnd - end position of the off-target
- name - gRNA name
- gRNAPlusPAM - gRNA sequence with PAM sequence concatenated
- OffTargetSequence - the genomic sequence of the off-target
- n.mismatch - number of mismatches between the off-target and the gRNA
- forViewInUCSC - string for viewing in UCSC genome browser, e.g., chr14:31665685-31665707
- score - score of the off-target
- mismatch.distance2PAM - comma-separated distances of all mismatches to PAM, e.g., 14,11 means one mismatch is 14 bp away from PAM and the other mismatch is 11 bp away from PAM
- alignment - alignment between gRNA and off-target, e.g., ...G..C..... means that this off-target aligns with gRNA except that G and C are mismatches
- NGG - whether this off-target contains canonical PAM (1 for yes, 0 for no)
- mean.neighbor.distance.mismatch - mean distance between neighboring mismatches

Author(s)

Lihua Julie Zhu

References

Patrick D Hsu, David A Scott, Joshua A Weinstein, F Ann Ran, Silvana Konermann, Vineeta Agarwala, Yinqing Li, Eli J Fine, Xuebing Wu, Ophir Shalem, Thomas J Cradick, Luciano A Marraffini, Gang Bao & Feng Zhang (2013) DNA targeting specificity of rNA-guided Cas9 nucleases. Nature Biotechnology 31:827-834

See Also

offTargetAnalysis

Examples

```
hitsFile <-  system.file("extdata", "hits.txt",
    package = "CRISPRseek")
hits <- read.table(hitsFile, sep = "\t", header = TRUE,
    stringsAsFactors = FALSE)
featureVectors <- buildFeatureVectorForScoring(hits)
getOfftargetScore(featureVectors)
```

---

getOfftargetWithBulge *Identify off-targets with bulges for target-specific gRNAs designed for CRISPR-Cas9 systems.*

Description

This function extends the off-targets identified by offTargetAnalysis() by detecting off-targets that contain bulges. In gRNA design, "bulges" refer to insertions ("RNA bulges") or deletions ("DNA bulges") in the gRNA sequence relative to the target DNA sequence. Bulges can affect the binding affinity and specificity of the gRNA to its target. The function wraps around [‘Cas-OFFinder‘](http://www.rgenome.net/ca
offinder/) internally.

Usage

```
getOfftargetWithBulge(
  gRNA_PAM = NULL,
  output_csv_name = NULL,
  PAM.size = 3,
  PAM.pattern = "NNG$|NGN$",
  PAM.location = c("3prime", "5prime"),
  max.mismatch = 3,
  DNA_bulge = 2,
  RNA_bulge = 2,
  BSgenomeName = NULL,
  genomeSeqFile = NULL,
  chromToSearch = "all",
  chromToExclude = NULL,
  cas_offinder_version = c("2.4.1", "3.0.0b3")
)
```

Arguments

gRNA_PAM	A ‘DNAStringSet‘ object returned by ‘findgRNA()‘ that contains gRNA plus PAM sequences. Alternatively, you can supply the ‘list‘ object returned by the ‘offTargetAnalysis()‘ function.
output_csv_name	A string specifying the output CSV file name. Defaults to ‘NULL‘, meaning that the output will be printed to the console.
PAM.size	See ‘offTargetAnalysis()‘.

PAM.pattern	See ‘offTargetAnalysis()‘.
PAM.location	See ‘offTargetAnalysis()‘.
max.mismatch	See ‘offTargetAnalysis()‘.
DNA_bulge	The maximum size of DNA bulges, specified in nucleotides. Defaults to 2.
RNA_bulge	The maximum size of RNA bulges, specified in nucleotides. Defaults to 2.
BSgenomeName	See ‘offTargetAnalysis()‘. Alternatively, use ‘genomeSeqFile‘ to specify the file path to custom genome fasta file. Note, ‘genomeSeqFile‘ overwrites ‘BSgenomeName‘ if both set.
genomeSeqFile	If you are using a custom genome, specify the file path to the FASTA file using ‘genomeSeqFile‘.
chromToSearch	See ‘offTargetAnalysis()‘.
chromToExclude	See ‘offTargetAnalysis()‘.
cas_offinder_version	The version of "Cas-OFFinder" to use. Currently supported versions are "2.4.1" and "3.0.0b3". Defaults to "2.4.1".

Value

If ‘output_csv_name‘ is not set, the function returns a data frame containing the output generated by ‘Cas-OFFinder‘. Otherwise, it saves the data frame to the CSV file specified by ‘output_csv_name‘. When ‘cas_offinder_version == "2.4.1"‘, the following columns will be included: "bulge_type", "gRNA", "DNA", "chr", "start_0_based", "strand", "mismatches", "bulge_size". For ‘cas_offinder_version == "3.0.0b3"‘, the included columns will be: "gRNA_id", "bulge_type", "gRNA", "DNA", "chr", "start_0_based", "strand", "mismatches", "bulge_size".

Author(s)

Kai Hu

References

1. Sangsu Bae, Jeongbin Park, Jin-Soo Kim, Cas-OFFinder: a fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases, Bioinformatics, Volume 30, Issue 10, May 2014, Pages 1473–1475, https://doi.org/10.1093/bioinformatics/btu048

See Also

‘offTargetAnalysis()‘ for off-targets analysis, ‘Cas-OFFinder‘ (https://github.com/snugel/cas-offinder) for more on output format.

Examples

```
# Example with `DNAStringSet` as input
if (interactive()) {
  library(CRISPRseek)
  library(BSgenome.Hsapiens.UCSC.hg19)

  gRNA_PAM <- findgRNAs(inputFilePath = system.file("extdata",
                                                    "inputseq.fa",
                                                     package = "CRISPRseek"),
                        pairOutputFile = "testpairedgRNAs.xls",
                        findPairedgRNAOnly = TRUE)
```

```
 df <- getOfftargetWithBulge(gRNA_PAM, PAM.pattern = "NNG$|NGN$",
                             DNA_bulge = 2, RNA_bulge = 2,
                             BSgenomeName = Hsapiens, chromToSearch = "chrX")

# Example with `list` output from `offTargetAnalysis` as input
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)

inputFilePath <- system.file("extdata", "inputseq.fa", package = "CRISPRseek")
REpatternFile <- system.file("extdata", "NEBenzymes.fa", package = "CRISPRseek")
res <- offTargetAnalysis(inputFilePath,
                         findgRNAsWithREcutOnly = TRUE,
                         REpatternFile = REpatternFile,
                         findPairedgRNAOnly = FALSE,
                         annotatePaired = FALSE,
                         BSgenomeName = Hsapiens,
                         chromToSearch = "chrX",
                         txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                         orgAnn = org.Hs.egSYMBOL, max.mismatch = 1,
                         outputDir = tempdir(),
                         overwrite = TRUE)
df <- getOfftargetWithBulge(res, PAM.pattern = "NNG$|NGN$",
                            DNA_bulge = 2,
                            RNA_bulge = 2,
                            BSgenomeName = Hsapiens,
                            chromToSearch = "chrX")
}
```

---

gRNA.backbone_default *gRNA.backbone_default*

Description

gRNA.backbone_default

Usage

```
gRNA.backbone_default
```

Format

An object of class `character` of length 1.

---

isPatternUnique *Output whether the input patterns occurs only once in the sequence*

Description

Input a sequence and a list of patterns and determine if the patterns occurs only once in the sequence. Used for determining whether an RE site in gRNA also occurs in the flanking region.

Usage

```
isPatternUnique(seq, patterns)
```

Arguments

seq	flanking sequence of a gRNA
patterns	patterns as DNAStringSet, such as a list of RE sites

Value

returns a character vectors containing the uniqueness of each pattern/RE site

Author(s)

Lihua Julie Zhu

Examples

```
seq <- "TGGATTGTATAATCAGCATGGATTTGGAAC"
patterns <- DNAStringSet(c("TGG", "TGGA", "TGGATA", "TTGGAAC", ""))
isPatternUnique(seq, patterns)
```

---

mismatch.activity.file_default

mismatch.activity.file_default

Description

Default value for mismatch.activity.file (csv), use mismatch.activity.file_default() to show its value

Usage

```
mismatch.activity.file_default()
```

---

mismatch.activity.file_default_xlsx

mismatch.activity.file_default_xlsx

Description

Default value for mismatch.activity.file (xlsx), use mismatch.activity.file_default_xlsx() to show its value.

Usage

```
mismatch.activity.file_default_xlsx()
```

offTargetAnalysis *Design target-specific guide RNAs for CRISPR-Cas9 system in one function*

Description

Design target-specific guide RNAs (gRNAs) and predict relative indel fequencies for CRISPR-Cas9 system by automatically calling findgRNAs, filtergRNAs, searchHits, buildFeatureVectorForScoring, getOfftargetScore, filterOfftarget, calculating gRNA cleavage efficiency, and predict gRNA efficacy, indels and their frequencies.

Usage

```
offTargetAnalysis(
  inputFilePath = NULL,
  format = c("fasta", "fastq", "bed"),
  header = FALSE,
  gRNAoutputName = "test",
  findgRNAs = TRUE,
  exportAllgRNAs = c("all", "fasta", "genbank", "no"),
  findgRNAsWithREcutOnly = FALSE,
  REpatternFile = REpatternFile_default(),
  minREpatternSize = 4,
  overlap.gRNA.positions = c(17, 18),
  findPairedgRNAOnly = FALSE,
  annotatePaired = TRUE,
  paired.orientation = c("PAMout", "PAMin"),
  enable.multicore = FALSE,
  n.cores.max = 6,
  min.gap = 0,
  max.gap = 20,
  gRNA.name.prefix = NULL,
  gRNA.size = 20,
  PAM = "NGG",
  PAM.size = width(PAM),
  PAM.pattern = "NNG$|NGN$",
  BSgenomeName = NULL,
  genomeSeqFile = NULL,
  chromToSearch = "all",
  chromToExclude = chromToExclude_default,
  max.mismatch = 3,
  allowed.mismatch.PAM = 1,
  gRNA.pattern = NULL,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8,
  editingWindow.offtargets = 4:8,
  primeEditing = FALSE,
  PBS.length = 13L,
  RT.template.length = 8:28,
  RT.template.pattern = "D$",
```

```
  corrected.seq = NULL,
  targeted.seq.length.change = NULL,
  bp.after.target.end = 15L,
  target.start = NULL,
  target.end = NULL,
  primeEditingPaired.output = "pairedgRNAsForPE.xls",
  min.score = 0,
  topN = 1000,
  topN.OfftargetTotalScore = 10,
  annotateExon = TRUE,
  txdb = NULL,
  orgAnn = NULL,
  ignore.strand = TRUE,
  outputDir = getwd(),
  fetchSequence = TRUE,
  upstream = 200,
  downstream = 200,
  weights = weights_default,
  baseBeforegRNA = 4,
  baseAfterPAM = 3,
  featureWeightMatrixFile = featureWeightMatrixFile_default(),
  useScore = TRUE,
  useEfficacyFromInputSeq = FALSE,
  outputUniqueREs = TRUE,
  foldgRNAs = FALSE,
  gRNA.backbone = gRNA.backbone_default,
  temperature = 37,
  overwrite = FALSE,
  scoring.method = c("Hsu-Zhang", "CFDscore"),
  subPAM.activity = subPAM.activity_default,
  subPAM.position = c(22, 23),
  PAM.location = "3prime",
 rule.set = c("Root_RuleSet1_2014", "Root_RuleSet2_2016", "CRISPRscan", "DeepCpf1"),
  chrom_acc = NULL,
  calculategRNAefficacyForOfftargets = TRUE,
  mismatch.activity.file = mismatch.activity.file_default(),
  predIndelFreq = FALSE,
  predictIndelFreq.onTargetOnly = TRUE,
  method.indelFreq = "Lindel",
  baseBeforegRNA.indelFreq = 13,
  baseAfterPAM.indelFreq = 24,
  findOffTargetsWithBulge = FALSE,
  method.findOffTargetsWithBulge = c("CasOFFinder_v3.0.0b3"),
  DNA_bulge = 2,
  RNA_bulge = 2
)
```

Arguments

inputFilePath	Path to an input sequence file or a 'DNAStringSet' object containing sequences to be searched for potential gRNAs.
format	Defaults to "fasta". Format of the input file, "fasta", "fastq", and "bed" are sup-

ported.

header Defaults to FALSE. Indicates whether the input file contains header. Only relevant when ‘format‘ is set to "bed".

gRNAoutputName Defaults to "test". Specifies the name of the gRNA output file when ‘inputFilePath‘ is a ‘DNAStringSet‘ object instead of a file path.

findgRNAs Defaults to TRUE. Specifies whether to find gRNAs from the sequences in ‘inputFilePath‘. Set to FALSE if the input file already contains user-selected gRNAs plus PAM.

exportAllgRNAs Defaults to "both". Indicates whether to output all potential gRNAs to a file in fasta format, genbank format, or both.

findgRNAsWithREcutOnly
Defaults to TRUE. Specifies whether to search for gRNAs that overlap with restriction enzyme recognition sites only.

REpatternFile Path to a file containing restriction enzyme cut patterns.

minREpatternSize
Defaults to 4. Minimum restriction enzyme recognition pattern length required for the enzyme pattern to be searched for.

overlap.gRNA.positions
Defaults to ‘c(17, 18)‘. Specifies the required overlapping positions of the gRNA and restriction enzyme cut site. For Cpf1, you can set it to ‘c(19, 23)‘.

findPairedgRNAOnly
Defaults to FALSE. Specifies whether to search only for paired gRNAs in such an orientation that the first one is on the minus strand (reverse gRNA) and the second one is on plus strand (forward gRNA).

annotatePaired Defaults to TRUE. Specifies whether to output paired gRNA information.

paired.orientation
The "PAMin" orientation refers to the scenario where the two adjacent PAMs on the sense and antisense strands face inward toward each other, such as in "N21GG" and "CCN21". In contrast, the "PAMout" orientation occurs when the PAMs face away from each other, as seen in "CCN21" and "N21GG".

enable.multicore
Defaults to FALSE. Indicates whether to enable parallel. For super long sequences with lots of gRNAs, set it to TRUE.

n.cores.max Defaults to 6. Specifies the maximum number of cores to use in multicore mode. Set it to 1 to disable multicore processing for small dataset.

min.gap Defaults to 0. Minimum distance between two oppositely oriented gRNAs to be considered as valid paired gRNAs.

max.gap Defaults to 20. Specifies the maximum distance between two oppositely oriented gRNAs to be considered as valid paired gRNAs.

gRNA.name.prefix
Defaults to "gRNA". Specifies the prefix used when assigning names to detected gRNAs.

gRNA.size Defaults to 20. The size of the gRNA.

PAM Defaults to "NGG". Defines the protospacer adjacent motif sequence.

PAM.size Defaults to ‘width(PAM)‘. Specifies the PAM length.

PAM.pattern Defaults to "NNG$|NGN$" (for spCas9). Specifies the regular expression of PAM. For cpf1, set to "^TTTN" since its PAM is at the 5 prime end.

`BSgenomeName` A 'BSgenome' object containing the target genome sequence, used for off-target search. Please refer to available genomes in the "BSgenome" package. For example,

- BSgenome.Hsapiens.UCSC.hg19 - for hg19,
- BSgenome.Mmusculus.UCSC.mm10 - for mm10
- BSgenome.Celegans.UCSC.ce6 - for ce6
- BSgenome.Rnorvegicus.UCSC.rn5 - for rn5
- BSgenome.Drerio.UCSC.danRer7 - for Zv9
- BSgenome.Dmelanogaster.UCSC.dm3 - for dm3

`genomeSeqFile` Alternative to 'BSgenomeName'. Specifies the path to a custom target genome file in FASTA format, used for off-target search. It is applicable when 'BSgenomeName' is NOT set. When 'genomeSeqFile' is set, the 'annotateExon', 'txdb', and 'orgAnn' parameters will be ignored.

`chromToSearch` Defaults to "all", meaning all chromosomes in the target genome are searched for off-targets. Set to a specific chromosome (e.g., "chrX") to restrict the search to that chromsome only.

`chromToExclude` If set to "", means to search off-targets in chromosomes specified in 'chromToSearch'. By default, to exclude haplotype blocks from off-target search assuming using 'hg19' genome, i.e., 'chromToExclude = c("chr17_ctg5_hap1", "chr4_ctg9_hap1", "chr6_apd_hap1", "chr6_cox_hap2", "chr6_dbb_hap3", "chr6_mann_hap4", "chr6_mcf_hap5", "chr6_qbl_hap6", "chr6_ssto_hap7")'.

`max.mismatch` Defaults to 3. Maximum number of mismatches allowed in off-target search. Warning: search will be considerably slower if set to a value greater than 3.

`allowed.mismatch.PAM` Defaults to 1. Maximum number of mismatches allowed in the PAM sequence for off-target search. The default value 1 allows "NGN" and "NNG" PAM patterns for off-target identification.

`gRNA.pattern` Defaults to NULL (meaning no restriction). Specifies regular expression or IUPAC Extended Genetic Alphabet to represent gRNA pattern. E.g. to specify that the gRNA must start with "GG", set it to "^GG". Type '?translatePattern' for a list of IUPAC Extended Genetic Alphabet.

`baseEditing` Defaults to FALSE. Specifies whether to design gRNAs for base editing. If set to TRUE, please set 'targetBase' and 'editingWidow'.

`targetBase` Defaults to "C" (for converting C to T in the CBE system). Applicable only when 'baseEditing = TRUE'. Specifies the target base for base editing systems. Please change it to "A" if you intend to use the ABE system.

`editingWindow` Defaults to '4:8' (for the CBE system). Applicable only when 'baseEditing = TRUE', and specifies the effective editing window. Please change it accordingly if the system you use have a different editing window.

`editingWindow.offtargets` Defaults to '4:8' (for the original CBE system, 1 means the most distal site from the 3' PAM, the most proximal site from the 5' PAM). Applicable only when 'baseEditing = TRUE'. Indicates the effective editing window to consider for the off-targets search only. Please change it accordingly if the system you use have a different editing window, or if you would like to include off-targets with the target base in a larger editing window.

`primeEditing` Defaults to FALSE. Specifies whether to design gRNAs for prime editing. If set to TRUE, please set 'PBS.length', 'RT.template.length', 'RT.template.pattern', 'targeted.seq.length.change', 'bp.after.target.end', 'target.start', 'target.end', and 'corrected.seq' accordingly.

`PBS.length` Applicable only when 'primeEditing = TRUE'. Specifies the number of bases to output for primer binding site.

`RT.template.length`
Defaults to '8:18'. Applicable only when 'primeEditing = TRUE'. Specifies the number of bases required for RT template. Increase the length if the edit involves a large insertion. Only gRNAs with a calculated 'RT.template.length' within the specified range will be included in the output. It is calculated as the following: 'RT.template.length = target.start – cut.start + (target.end - target.start) + targeted.seq.length.change + bp.after.target.end'.

`RT.template.pattern`
Defaults to not end with C (per https://doi.org/10.1038/s41586-019-1711-4). Applicable only when 'primeEditing = TRUE'. Specifies the RT template sequence pattern.

`corrected.seq` Applicable only when 'primeEditing = TRUE'. Specifies the mutated or inserted sequences after successful editing.

`targeted.seq.length.change`
Applicable only when 'primeEditing = TRUE'. Specifies the change in the targeted sequence length. Set it to 0 for base changes, positive numbers for insertions, and negative number for deletions. For example, 10 indicates that the corrected sequence will have a 10-bp insertion, -10 means that the corrected sequence will have a 10-bp deletion, and 0 means that only base changes with no change in sequence length.

`bp.after.target.end`
Defaults to 15. Applicable only when 'primeEditing = TRUE'. Specifies the number of bases to add after the target change end site as part of the RT template. Refer to 'RT.template.length' for how this parameter affects the calculation of 'RT.template.length', which is used as a filtering criterion during pregRNA selection.

`target.start` Defaults to 20. Applicable only when 'primeEditing = TRUE'. Specifies the start location in the input sequence to make changes, which will be used to obtain the RT template sequence. Refer to 'RT.template.length' for how this parameter affects the 'RT.template.length' calculation, which is used as a filtering criteria in pregRNA selection.

`target.end` Defaults to 20. Applicable only when 'primeEditing = TRUE'. Specifies the end location in the input sequence to make changes, which will be used to obtain the RT template sequence. Refer to 'RT.template.length' for how this parameter affects the 'RT.template.length' calculation, which is used as a filtering criteria in pregRNA selection.

`primeEditingPaired.output`
Defaults to "pairedgRNAsForPE.xls". Applicable only when 'primeEditing = TRUE'. Specifies the file path where the pegRNA, second gRNA wit PBS, RT.template, and gRNA sequences will be saved.

`min.score` Defaults to 0. Specifies the minimum score of an off-target to be included in the final output.

`topN` Defaults to 1000. Specifies the top N off-targets to be included in the final output

`topN.OfftargetTotalScore`
Defaults to 10. Specifies the top N off-targets used to calculate the total off-target score.

`annotateExon` Defaults to TRUE. Specifies whether to indicate if the off-target is located within an exon.

`txdb` A 'TxDb' object containing organism-specific annotation data, required for 'annotateExon'. For creating and using a 'TxDb' object, refer to the 'GenomicFeatures' package. For a list of existing 'TxDb' objects, search for annotation packages starting with "Txdb" at http://www.bioconductor.org/packages/release/BiocViews.html#___Ann such as

- TxDb.Rnorvegicus.UCSC.rn5.refGene - for rat
- TxDb.Mmusculus.UCSC.mm10.knownGene - for mouse
- TxDb.Hsapiens.UCSC.hg19.knownGene - for human
- TxDb.Dmelanogaster.UCSC.dm3.ensGene - for Drosophila
- TxDb.Celegans.UCSC.ce6.ensGene - for C.elegans

`orgAnn` An 'OrgDb' object containing organism-specific annotation mapping information, required for 'annotateExon'.

`ignore.strand` Defaults to TRUE. Specifies if strandness should be ignored when annotating off-targets to genes.

`outputDir` Defaults to the current working directory. Specifies the path to the directory where the analysis results will be saved.

`fetchSequence` Defaults to TRUE. Specifies whether to fetch flanking sequences for off-targets.

`upstream` Defaults to 200. Specifies the upstream offset from the off-target start.

`downstream` Defaults to 200. Specifies the downstream offset from the off-target end.

`weights` Defaults to 'c(0, 0, 0.014, 0, 0, 0.395, 0.317, 0, 0.389, 0.079, 0.445, 0.508, 0.613, 0.851, 0.732, 0.828, 0.615, 0.804, 0.685, 0.583)' (used in Hsu et al., 2013 cited in the reference section). Applicable only when 'scoring.method = Hus-Zhang'. Specifies a numeric vector with a length equal to the size of the gRNA, containing the corresponding weight values.

`baseBeforegRNA` Defaults to 4. Specifies the number of bases preceding the gRNA. It is used to calculate gRNA efficiency. Note that for PAMs located at the 5 prime end, the number of bases should include both the bases before the PAM sequence and the PAM size.

`baseAfterPAM` Defaults to 3 (for spCas9). Specifies the number of bases after PAM. It is used to calculate gRNA efficiency. Note that for PAMs located on the 5 prime end, the number should include the length of the gRNA plus the extended sequence on the 3 prime end.

`featureWeightMatrixFile`
By default, the DoenchNBT2014 weight matrix is used. Specifies the feature weight matrix file used for calculating gRNA efficiency. To use an alternative matrix, provide a CSV where the first column contains the significant features and the second column contains the corresponding weights. For details, refer to Doench et al., 2014.

`useScore` Defaults to TRUE. Displays in grayscale, with darkness indicating gRNA efficacy. The taller bar represents the Cas9 cutting site. If set to False, efficacy will not be shown. Instead, gRNAs on the plus strand will be colored red, and gRNAs on the minus strand will be colored green.

`useEfficacyFromInputSeq`
Defaults to FALSE. If TRUE, the summary file will contain gRNA efficacy calculated from the input sequences instead of from off-target analysis. Set it to TRUE if the input sequence is from a species different from the one used for off-target analysis.

`outputUniqueREs`
: Defaults to TRUE. If set to TRUE, summary file will contain REs unique to the cleavage site within 100 or 200 bases surrounding the gRNA sequence.

`foldgRNAs`
: Defaults to FALSE. If set to TRUE, summary file will contain minimum free energy of the secondary structure of gRNA with gRNA backbone from 'GeneRfold' package given that 'GeneRfold' package has been installed.

`gRNA.backbone`
: Defaults to the sequence in Sp gRNA backbone. Applicable only when 'foldgRNAs = TRUE'. Specifies the gRNA backbone constant region sequence.

`temperature`
: Defaults to 30. Applicable only when 'foldgRNAs = TRUE'. Specifies the temperature in Celsius.

`overwrite`
: Defaults to FALSE. Specifies whether to overwrite the existing files in the output directory.

`scoring.method`
: Defaults to "Hsu-Zhang". Specifies the method to use for off-target cleavage rate estimation. Choose from "Hsu-Zhang" and "CFDscore"

`subPAM.activity`
: Defaults to "hash(AA = 0, AC = 0, AG = 0.259259259, AT = 0, CA = 0, CC = 0, CG = 0.107142857, CT = 0, GA = 0.069444444, GC = 0.022222222, GG = 1, GT = 0.016129032, TA = 0, TC = 0, TG = 0.038961039, TT = 0)". Applicable only when 'scoring.method = CFDscore'. Specifies a hash that represents the cleavage rate for each alternative sub PAM sequence relative to preferred PAM sequence.

`subPAM.position`
: Defaults to 'c(22, 23)' (For spCas9 with 20-bp gRNA and NGG as preferred PAM). Applicable only when 'scoring.method = CFDscore'. Specifies the start and end positions of the sub PAM. For Cpf1, it should be 'c(1,2)'.

`PAM.location`
: Defaults to "3prime" (for spCas9). Specifies the PAM location relative to the protospacer sequence. Set to "5prime" for cpf1 because its PAM is located at the 5 prime end of the protospacer.

`rule.set`
: Defaults to "Root_RuleSet1_2014". Specifies a rule set scoring system for calculating gRNA efficacy. Note that "Root_RuleSet2_2016" requires the following packages with specified version: python 2.7, scikit-learn 0.16.1, pickle, pandas, numpy, and scipy.

`chrom_acc`
: Specifies an optional binary variable indicating chromatin accessibility information with 1 representing accessible and 0 representing inaccessible.

`calculategRNAefficacyForOfftargets`
: Defaults to TRUE. Specifies whether to output gRNA efficacy for both off-targets and on-targets. Set to FALSE if only on-target gRNA efficacy is needed to speed up the analysis. For potential use cases of off-target efficacies, refer to https://support.bioconductor.org/p/133538/#133661.

`mismatch.activity.file`
: Defaults to use the supplementary Table 19 from Doench et al., Nature Biotechnology 2016. Applicable only when 'scoring.method = CFDscore'. Specifies a CSV file containing the cleavage rates for all possible types of single nucleotide mismatches at each position of the gRNA.

`predIndelFreq`
: Defaults to FALSE. Specifies whether to output the predicted INDELs and their frequencies.

`predictIndelFreq.onTargetOnly`
: Defaults to TRUE. Specifies whether to predict INDELs and their frequencies for on-targets only. Typically, researchers are only interested in predicting editing outcome for on-targets, as editing in off-targets is undesirable. Set to FALSE

if you want to predict INDELs and their frequencies for off-targets as well. Note that this will increase the run time.

`method.indelFreq`
: Defaults to "Lindel". Applicable only when 'predIndelFreq = TRUE'. Specifies the method to be used for predicting INDELs. Currently, only "Lindel" is supported, though additional methods can be added upon request. Type '?predictRelativeFreqIndels' to learn more.

`baseBeforegRNA.indelFreq`
: Defaults to 13. Applicable only when 'predIndelFreq = TRUE'.

`baseAfterPAM.indelFreq`
: Defaults to 24. Applicable only when 'predIndelFreq = TRUE'.

`findOffTargetsWithBulge`
: Defaults to FALSE. Specifies whether to search for off-targets with bulges.

`method.findOffTargetsWithBulge`
: Only applicable if 'findOffTargetsWithBulge = TRUE'. Choose from 'c("CasOFFinder_v3.0.0b3")'.

`DNA_bulge`
: Defaults to 2. Maximum number of DNA bulges allowed in off-target search.

`RNA_bulge`
: Defaults to 2. Maximum number of RNA bulges allowed in off-target search.

Value

Four Excel files are generated in the output directory:

`Summary.xlsx` - Summary of the gRNAs

`OfftargetAnalysis.xlsx`
: - Detailed information on off-targets

`REcutDetails.xlsx`
: - Restriction enzyme cut sites for each gRNA

`pairedgRNAs.xlsx`
: - Potential paired gRNAs

Author(s)

Lihua Julie Zhu, Kai Hu

References

Patrick D Hsu, David A Scott, Joshua A Weinstein, F Ann Ran, Silvana Konermann, Vineeta Agarwala, Yinqing Li, Eli J Fine, Xuebing Wu, Ophir Shalem, Thomas J Cradick, Luciano A Marraffini, Gang Bao & Feng Zhang (2013) DNA targeting specificity of rNA-guided Cas9 nucleases. Nature Biotechnology 31:827-834

Doench JG, Hartenian E, Graham DB, Tothova Z, Hegde M, Smith I, Sullender M, Ebert BL, Xavier RJ, Root DE. Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotechnol. 2014 Sep 3. doi: 10.1038 nbt.3026

Lihua Julie Zhu, Benjamin R. Holmes, Neil Aronin and Michael Brodsky. CRISPRseek: a Bioconductor package to identify target-specific guide RNAs for CRISPR-Cas9 genome-editing systems. Plos One Sept 23rd 2014

Moreno-Mateos, M., Vejnar, C., Beaudoin, J. et al. CRISPRscan: designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo. Nat Methods 12, 982–988 (2015) doi:10.1038/nmeth.3543

Doench JG et al., Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nature Biotechnology Jan 18th 2016

Anzalone et al., Search-and-replace genome editing without double-strand breaks or donor DNA. Nature October 2019 https://www.nature.com/articles/s41586-019-1711-4

Wei Chen, Aaron McKenna, Jacob Schreiber et al., Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair, Nucleic Acids Research, Volume 47, Issue 15, 05 September 2019, Pages 7989–8003, https://doi.org/10.1093/nar/gkz487

Kim et al., Deep learning improves prediction of CRISPR–Cpf1 guide RNA activityNat Biotechnol 36, 239–241 (2018). https://doi.org/10.1038/nbt.4061

See Also

CRISPRseek

Examples

```
# Load required libraries
library(CRISPRseek)
library(BSgenome.Hsapiens.UCSC.hg19)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(org.Hs.eg.db)

# Example 1: given FASTA input, search gRNAs and off-targets
outputDir <- tempdir()
inputFilePath <- system.file("extdata/inputseq.fa", package = "CRISPRseek")
REpatternFile <- system.file("extdata/NEBenzymes.fa", package = "CRISPRseek")

results <- offTargetAnalysis(inputFilePath,
                             findPairedgRNAOnly = FALSE,
                             findgRNAsWithREcutOnly = TRUE,
                             REpatternFile = REpatternFile,
                             annotatePaired = FALSE,
                             BSgenomeName = Hsapiens,
                             chromToSearch = "chrX",
                             txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                             orgAnn = org.Hs.egSYMBOL,
                             max.mismatch = 1,
                             outputDir = outputDir,
                             overwrite = TRUE)

# Example 2: also predict indels and frequecies at target sites
results <- offTargetAnalysis(inputFilePath,
                             predIndelFreq = TRUE,
                             predictIndelFreq.onTargetOnly = TRUE,
                             findgRNAsWithREcutOnly = TRUE,
                             findPairedgRNAOnly = FALSE,
                             annotatePaired = FALSE,
                             BSgenomeName = Hsapiens,
                             chromToSearch = "chrX",
                             txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                             orgAnn = org.Hs.egSYMBOL,
                             max.mismatch = 1,
                             outputDir = outputDir,
                             overwrite = TRUE)
names(results$indelFreq)
head(results$indelFreq[[1]])
  # Save the indel frequences to tab delimited files,
  # one file for each target or offtarget site.
```

```
mapply(openxlsx::write.xlsx, results$indelFreq,
       file = paste0(names(results$indelFreq), '.xlsx'))

# Example 3: predict gRNA efficacy using CRISPRscan
featureWeightMatrixFile <- system.file("extdata/Morenos-Mateo.csv",
                                       package = "CRISPRseek")

results <- offTargetAnalysis(inputFilePath,
                             rule.set = "CRISPRscan",
                             findgRNAsWithREcutOnly = TRUE,
                             REpatternFile = REpatternFile,
                             findPairedgRNAOnly = FALSE,
                             annotatePaired = FALSE,
                             BSgenomeName = Hsapiens,
                             chromToSearch = "chrX",
                             txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                             orgAnn = org.Hs.egSYMBOL,
                             max.mismatch = 1,
                             baseBeforegRNA = 6,
                             baseAfterPAM = 6,
                             featureWeightMatrixFile = featureWeightMatrixFile,
                             outputDir = outputDir,
                             overwrite = TRUE)

# Example 4: when PAM is on the 5 prime side, e.g., Cpf1
if (interactive()) {
  results <- offTargetAnalysis(inputFilePath =
                               system.file("extdata/cpf1-2.fa",
                                           package = "CRISPRseek"),
                               PAM.location = "5prime",
                               rule.set = "DeepCpf1",
                               PAM.size = 4,
                               PAM = "TTTN",
                               PAM.pattern = "^TNNN",
                               findgRNAsWithREcutOnly =  FALSE,
                               findPairedgRNAOnly = FALSE,
                               annotatePaired = FALSE,
                               BSgenomeName = Hsapiens,
                               chromToSearch = "chr8",
                               txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                               orgAnn = org.Hs.egSYMBOL, max.mismatch = 4,
                               baseBeforegRNA = 8, baseAfterPAM = 26,
                               overlap.gRNA.positions = c(19, 23),
                               useEfficacyFromInputSeq = FALSE,
                               outputDir = outputDir,
                               overwrite = TRUE,
                               allowed.mismatch.PAM = 2,
                               subPAM.position = c(1, 2))
}

# Example 5: when PAM is on the 5 prime side, and using Root_RuleSet1_2014
results <- offTargetAnalysis(inputFilePath,
                             PAM.location = "5prime",
                             PAM = "TGT",
                             PAM.pattern = "^T[A|G]N",
                             findgRNAsWithREcutOnly =  FALSE,
                             REpatternFile = REpatternFile,
```

```
                              findPairedgRNAOnly = FALSE,
                              annotatePaired = FALSE,
                              BSgenomeName = Hsapiens,
                              chromToSearch = "chrX",
                              txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                              orgAnn = org.Hs.egSYMBOL,
                              max.mismatch = 4,
                              outputDir = outputDir,
                              overwrite = TRUE,
                              allowed.mismatch.PAM = 2,
                              subPAM.position = c(1, 2),
                              baseEditing = TRUE,
                              editingWindow = 20,
                              targetBase = "G")

    # Example 6: base editor
    results <- offTargetAnalysis(inputFilePath,
                              baseEditing = TRUE,
                              editingWindow = 10:20,
                              targetBase = "A",
                              findgRNAsWithREcutOnly = FALSE,
                              REpatternFile = REpatternFile,
                              findPairedgRNAOnly = FALSE,
                              annotatePaired = FALSE,
                              BSgenomeName = Hsapiens,
                              chromToSearch = "chrX",
                              txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
                              orgAnn = org.Hs.egSYMBOL,
                              max.mismatch = 4,
                              PAM.location = "5prime",
                              PAM = "TGT",
                              PAM.pattern = "^T[A|G]N",
                              allowed.mismatch.PAM = 2,
                              subPAM.position = c(1, 2),
                              outputDir = outputDir,
                              overwrite = TRUE)

    # Example 7: prime editor
    inputFilePath <- DNAStringSet(paste0("CCAGTTTGTGGATCCTGCTCTGGTGTCCTCCACACC",
                                         "AGAATCAGGGATCGAAAACTCATCAGTCGATGCGAG",
                                         "TCATCTAAATTCCGATCAATTTCACACTTTAAACG"))
    results <- offTargetAnalysis(inputFilePath,
                              primeEditing = TRUE,
                              overlap.gRNA.positions = c(17, 18),
                              PBS.length = 15,
                              corrected.seq = "T",
                              RT.template.pattern = "D$",
                              RT.template.length = 8:30,
                              targeted.seq.length.change = 0,
                              bp.after.target.end = 15,
                              target.start = 20,
                              target.end = 20,
                              paired.orientation = "PAMin",
                              findPairedgRNAOnly = TRUE,
                              BSgenomeName = Hsapiens,
                              chromToSearch = "chrX",
                              txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
```

```
                                orgAnn = org.Hs.egSYMBOL,
                                max.mismatch = 1,
                                outputDir = outputDir,
                                overwrite = TRUE,
                                PAM.size = 3,
                                gRNA.size = 20,
                                min.gap = 20,
                                max.gap = 90)
```

`predictRelativeFreqIndels` *Predict insertions and deletions induced by CRISPR/Cas9 editing*

Description

Predict insertions and deletions, and associated reletive frequecies induced by CRISPR/Cas9 editing

Usage

```
predictRelativeFreqIndels(extendedSequence, method = "Lindel")
```

Arguments

`extendedSequence`	A vector of DNA sequences of length 60bp. It consists 30bp before the cut site and 30bp after the cut site.
`method`	the prediction method. default to Lindel. Currently only Lindel method are implemented.

Details

Predict relative indel frequency around target sites of CRISPR/Cas9 system. Currently only Lindel method using logistic regression is implemented in CRISPRseek.

Lindel is compatible with both Python2.7 and Python3.5 or higher.

By default, reticulate uses the version of Python found on your PATH (i.e. Sys.which("python")).

Use the function use_python in reticulate library to set the python path to a specific version. For example, use_python('/opt/anaconda3/lib/python3.7')

This function implements the Lindel method

Value

A list with the same length as the input extendedSequence.

Each list item either contains a warning message, or a predicted fraction of frameshift in the mutational outcomes plus a data frame with three columns.

The three columns are the alignment of predicted indel sequence to the original unedited sequence, predicted indel frequency, and the location of the predicted indels. The warning message for the Lindel method is as follows.

Warning: No PAM sequence is identified. Please check your sequence and try again.

A list with the same length as the input extendedSequence.

Each list item either contains a warning message, or a predicted fraction of frameshift in the mutational outcomes plus a data frame with three columns.

The three columns are the alignment of predicted indel sequence to the original unedited sequence, predicted indel frequency, and the location of the predicted indels. The warning message for the Lindel method is as follows.

Warning: No PAM sequence is identified. Please check your sequence and try again.

Author(s)

Hui Mao and Lihua Julie Zhu Predict insertions and deletions induced by CRISPR/Cas9 editing

Predict insertions and deletions, and associated reletive frequecies induced by CRISPR/Cas9 editing

Predict relative indel frequency around target sites of CRISPR/Cas9 system. Currently only Lindel method using logistic regression is implemented in CRISPRseek.

Lindel is compatible with both Python2.7 and Python3.5 or higher.

By default, reticulate uses the version of Python found on your PATH (i.e. Sys.which("python")).

Use the function use_python in reticulate library to set the python path to a specific version. For example, use_python('/opt/anaconda3/lib/python3.7')

This function implements the Lindel method

Hui Mao and Lihua Julie Zhu

References

Wei Chen, Aaron McKenna, Jacob Schreiber et al., Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair, Nucleic Acids Research, Volume 47, Issue 15, 05 September 2019, Pages 7989–8003, https://doi.org/10.1093/nar/gkz487

Wei Chen, Aaron McKenna, Jacob Schreiber et al., Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair, Nucleic Acids Research, Volume 47, Issue 15, 05 September 2019, Pages 7989–8003, https://doi.org/10.1093/nar/gkz487

Examples

```
extendedSequence <- c("AAA", "TAACGTTATCAACGCCTATATTAAAGCGACCGTCGGTTGAACTGCGTGGATCAATGCGTC")
if (interactive())
    indelFreq <- predictRelativeFreqIndels(extendedSequence, method = "Lindel")



extendedSequence <- c("AAA", "TAACGTTATCAACGCCTATATTAAAGCGACCGTCGGTTGAACTGCGTGGATCAATGCGTC")
if (interactive())
    indelFreq <- predictRelativeFreqIndels(extendedSequence, method = "Lindel")
```

predIndelFreq *Function definition place holders that are to be overwritten by reticulate This is to suppress the R CMD check NOTE about "no visible global function""*

Description

Function definition place holders that are to be overwritten by reticulate This is to suppress the R CMD check NOTE about "no visible global function""

Usage

```
predIndelFreq(thisSeq, weights)
```

Arguments

thisSeq	param1 used in Lindel
weights	param2 used in weights

REpatternFile_default *REpatternFile_default*

Description

Default value for REpatternFile, use REpatternFile_default() to show its value.

Usage

```
REpatternFile_default()
```

searchHits *Search for off targets in a sequence as DNAString*

Description

Search for off targets for given gRNAs, sequence and maximum mismatches

Usage

```
searchHits(
  gRNAs,
  seqs,
  seqname,
  max.mismatch = 3,
  PAM.size = 3,
  gRNA.size = 20,
  PAM = "NGG",
  PAM.pattern = "NNN$",
  allowed.mismatch.PAM = 2,
  PAM.location = "3prime",
  outfile,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8
)
```

Arguments

gRNAs	DNAStringSet object containing a set of gRNAs. Please note the sequences must contain PAM appended after gRNAs, e.g., ATCGAAATTCGAGCCAATCCCGG where ATCGAAATTCGAGCCAATCC is the gRNA and CGG is the PAM
seqs	DNAString object containing a DNA sequence.
seqname	Specify the name of the sequence
max.mismatch	Maximum mismatch allowed in off target search, default 3. Warning: will be considerably slower if it is set to greater than 3
PAM.size	Size of PAM, default 3
gRNA.size	Size of gRNA, default 20
PAM	PAM as regular expression for appending to the gRNA, default NGG for SpCas9, change to TTTN for cpf1.
PAM.pattern	Regular expression of PAM, default N[A|G]G$ for spCas9. For cpf1, ^TTTN since it is a 5 prime PAM sequence
allowed.mismatch.PAM	Maximum number of mismatches allowed in the offtargets comparing to the PAM sequence. Default to 2 for NGG PAM
PAM.location	PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime
outfile	File path to temporarily store the search results
baseEditing	Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.
targetBase	Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.
editingWindow	Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window to consider for the offtargets search only, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window, or you would like to include offtargets with the target base in a larger editing window.

Value

a data frame contains

- **IsMismatch.posX** - whether this position X is mismatch or not, (1 means yes and 0 means not). X takes on values from 1 to gRNA.size, representing all positions in the guide RNA (gRNA).
- **strand** - Strand of the match ('+' for plus, '-' for minus).
- **chrom** - Chromosome where the off-target is located.
- **chromStart** - Start position of the off-target site.
- **chromEnd** - End position of the off-target site.
- **name** - gRNA name.
- **gRNAPlusPAM** - gRNA sequence with PAM sequence concatenated.
- **OffTargetSequence** - Genomic sequence of the off-target.
- **n.mismatch** - Number of mismatches between the off-target and the gRNA.
- **forViewInUCSC** - String for viewing in UCSC Genome Browser (e.g., 'chr14:31665685-31665707').
- **score** - Defaulted to 100, and will be updated in 'getOfftargetScore()'.

Author(s)

Lihua Julie Zhu

See Also

offTargetAnalysis

Examples

```
all.gRNAs <- findgRNAs(inputFilePath =
      system.file("extdata", "inputseq.fa", package = "CRISPRseek"),
      pairOutputFile = "pairedgRNAs.xlsx",
      findPairedgRNAOnly = TRUE)
 hits <- searchHits(all.gRNAs[1],
     seqs = DNAString(
          "TAATATTTTAAAATCGGTGACGTGGGCCCAAAACGAGTGCAGTTCCAAAGGCACCCACCTGTGGCAG"),
     seqname = "myseq", max.mismatch = 10, outfile = "test_searchHits")
 colnames(hits)
 all.gRNAs <- findgRNAs(inputFilePath =
         DNAStringSet(
             "TAATATTTTAAAATCGGTGACGTGGGCCCAAAACGAGTGCAGTTCCAAAGGCACCCACCTGTGGCAG"),
         pairOutputFile = "pairedgRNAs.xlsx",
         findPairedgRNAOnly = FALSE,
         PAM = "TTTN", PAM.location = "5prime")
 hits <- searchHits(all.gRNAs[1], seqs = DNAString(
     "TAATATTTTAAAATCGGTGACGTGGGCCCAAAACGAGTGCAGTTCCAAAGGCACCCACCTGTGGCAG"),
     seqname = "myseq",
     max.mismatch = 0,
     outfile = "test_searchHits", PAM.location = "5prime",
     PAM.pattern = "^T[A|T]NN", allowed.mismatch.PAM = 0, PAM = "TTTN")
 colnames(hits)
```

searchHits2 *Search for off targets*

Description

Search for off targets for given gRNAs, BSgenome and maximum mismatches

Usage

```
searchHits2(
  gRNAs = NULL,
  BSgenomeName = NULL,
  chromToSearch = "all",
  chromToExclude = NULL,
  max.mismatch = 3,
  PAM.size = 3,
  gRNA.size = 20,
  PAM = "NGG",
  PAM.pattern = "N[A|G]G$",
  allowed.mismatch.PAM = 1,
  PAM.location = "3prime",
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8
)
```

Arguments

gRNAs	DNAStringSet object containing a set of gRNAs. Please note the sequences must contain PAM appended after gRNAs, e.g., ATCGAAATTCGAGCCAATCCCGG where ATCGAAATTCGAGCCAATCC is the gRNA and CGG is the PAM
BSgenomeName	BSgenome object. Please refer to available.genomes in BSgenome package. For example, • BSgenome.Hsapiens.UCSC.hg19 - for hg19, • BSgenome.Mmusculus.UCSC.mm10 - for mm10 • BSgenome.Celegans.UCSC.ce6 - for ce6 • BSgenome.Rnorvegicus.UCSC.rn5 - for rn5 • BSgenome.Drerio.UCSC.danRer7 - for Zv9 • BSgenome.Dmelanogaster.UCSC.dm3 - for dm3
chromToSearch	Specify the chromosome to search, default to all, meaning search all chromosomes. For example, chrX indicates searching for matching in chromosome X only
chromToExclude	Specify the chromosome not to search, default to none, meaning to search chromosomes specified by chromToSearch. For example, to exclude haplotype blocks from offtarget search in hg19, set chromToExclude to c(""chr17_ctg5_hap1","chr4_ctg9_hap1", "chr6_apd_hap1", "chr6_cox_hap2", "chr6_dbb_hap3", "chr6_mann_hap4", "chr6_mcf_hap5","chr6_ "chr6_ssto_hap7")

`max.mismatch` Maximum mismatch allowed in off target search, default 3. Warning: will be considerably slower if it is set to greater than 3

`PAM.size` Size of PAM, default 3

`gRNA.size` Size of gRNA, default 20

`PAM` Regular expression of protospacer-adjacent motif (PAM), default NGG for sp-Cas9. For cpf1, ^TTTN

`PAM.pattern` Regular expression of PAM, default N[A|G]G$ for spCas9. For cpf1, ^TTTN since it is a 5 prime PAM sequence

`allowed.mismatch.PAM` Number of degenerative bases in the PAM sequence, default to 1 for N[A|G]G PAM

`PAM.location` PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime

`baseEditing` Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.

`targetBase` Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.

`editingWindow` Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window to consider for the offtargets search only, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window, or you would like to include offtargets with the target base in a larger editing window.

Value

a data frame contains

- **IsMismatch.posX** - whether this position X is mismatch or not, (1 means yes and 0 means not). X takes on values from 1 to gRNA.size, representing all positions in the guide RNA (gRNA).
- **strand** - Strand of the match ('+' for plus, '-' for minus).
- **chrom** - Chromosome where the off-target is located.
- **chromStart** - Start position of the off-target site.
- **chromEnd** - End position of the off-target site.
- **name** - gRNA name.
- **gRNAPlusPAM** - gRNA sequence with PAM sequence concatenated.
- **OffTargetSequence** - Genomic sequence of the off-target.
- **n.mismatch** - Number of mismatches between the off-target and the gRNA.
- **forViewInUCSC** - String for viewing in UCSC Genome Browser (e.g., 'chr14:31665685-31665707').
- **score** - Defaulted to 100, and will be updated in 'getOfftargetScore()'.

Author(s)

Lihua Julie Zhu

See Also

offTargetAnalysis

Examples

```
    all.gRNAs <- findgRNAs(inputFilePath =
        system.file("extdata", "inputseq.fa", package = "CRISPRseek"),
        pairOutputFile = "pairedgRNAs.xlsx",
 findPairedgRNAOnly = TRUE)

    library("BSgenome.Hsapiens.UCSC.hg19")
    ### for speed reason, use max.mismatch = 0 for finding all targets with
    ### all variants of PAM
    hits <- searchHits2(all.gRNAs[1], BSgenomeName = Hsapiens,
        max.mismatch = 0, chromToSearch = "chrX")
    colnames(hits)

    ### test PAM located at 5 prime
    all.gRNAs <- findgRNAs(inputFilePath =
             system.file("extdata", "inputseq.fa", package = "CRISPRseek"),
             pairOutputFile = "pairedgRNAs.xlsx",
             findPairedgRNAOnly = FALSE,
             PAM = "TGT", PAM.location = "5prime")

    library("BSgenome.Hsapiens.UCSC.hg19")
         ### for speed reason, use max.mismatch = 0 for finding all targets with
         ### all variants of PAM
    hits <- searchHits2(all.gRNAs[1], BSgenomeName = Hsapiens, PAM.size = 3,
        max.mismatch = 0, chromToSearch = "chrX", PAM.location = "5prime",
        PAM = "NGG",
        PAM.pattern = "^T[A|G]N", allowed.mismatch.PAM = 2)
    colnames(hits)
```

`subPAM.activity_default`

subPAM.activity_default

Description

subPAM.activity_default

Usage

```
subPAM.activity_default
```

Format

An object of class hash of length 16.

translatePattern — *translate pattern from IUPAC Extended Genetic Alphabet to regular expression*

Description

translate pattern containing the IUPAC nucleotide ambiguity codes to regular expression. For example,Y->[C|T], R-> [A|G], S-> [G|C], W-> [A|T], K-> [T|U|G], M-> [A|C], B-> [C|G|T], D-> [A|G|T], H-> [A|C|T], V-> [A|C|G] and N-> [A|C|T|G].

Usage

```
translatePattern(pattern)
```

Arguments

pattern	a character vector with the IUPAC nucleotide ambiguity codes

Value

a character vector with the pattern represented as regular expression

Author(s)

Lihua Julie Zhu

Examples

```
pattern1 <- "AACCNWMK"
translatePattern(pattern1)
```

uniqueREs — *Output restriction enzymes that recognize only the gRNA cleavage sites*

Description

For each identified gRNA, output restriction enzymes that recognize only the gRNA cleavage sites.

Usage

```
uniqueREs(
  REcutDetails,
  summary,
  offTargets,
  scanUpstream = 100,
  scanDownstream = 100,
  BSgenomeName
)
```

Arguments

REcutDetails	REcutDetails stored in the REcutDetails.xls
summary	summary stored in the summary.xls
offTargets	offTargets stored in the offTargets.xls
scanUpstream	upstream offset from the gRNA start, default 100
scanDownstream	downstream offset from the gRNA end, default 100
BSgenomeName	BSgenome object. Please refer to available.genomes in BSgenome package. For example, • BSgenome.Hsapiens.UCSC.hg19 - for hg19 • BSgenome.Mmusculus.UCSC.mm10 - for mm10 • BSgenome.Celegans.UCSC.ce6 - for ce6 • BSgenome.Rnorvegicus.UCSC.rn5 - for rn5 • BSgenome.Drerio.UCSC.danRer7 - for Zv9 • BSgenome.Dmelanogaster.UCSC.dm3 - for dm3

Value

returns the RE sites that recognize only the gRNA cleavage sites for each gRNA.

Author(s)

Lihua Julie Zhu

Examples

```
   library("BSgenome.Hsapiens.UCSC.hg19")
   load(system.file("extdata", "ForTestinguniqueREs.RData",
           package = "CRISPRseek"))
   uniqueREs(results$REcutDetails, results$summary, results$offtarget,
scanUpstream = 50,
       scanDownstream = 50, BSgenomeName = Hsapiens)
```

---

weights_default *weights_default*

Description

weights_default

Usage

```
weights_default
```

Format

An object of class numeric of length 20.

writeHits *Write the hits of sequence search from a sequence to a file*

Description

write the hits of sequence search from a sequence instead of BSgenome to a file, internal function used by searchHits

Usage

```
writeHits(
  gRNA = NULL,
  seqname = NULL,
  matches = NULL,
  strand = NULL,
  file = NULL,
  gRNA.size = 20L,
  PAM = "NGG",
  PAM.pattern = "N[A|G]G$",
  max.mismatch = 4L,
  chrom.len = NULL,
  append = FALSE,
  PAM.location = "3prime",
  PAM.size = 3L,
  allowed.mismatch.PAM = 1L,
  seqs = NULL,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8
)
```

Arguments

gRNA	DNAString object with gRNA sequence with PAM appended immediately after,e.g., ACGTACGTACGTACTGACGTCGG with 20bp gRNA sequence plus 3bp PAM sequence CGG
seqname	sequence name as character
matches	XStringViews object storing matched chromosome locations
strand	strand of the match, + for plus strand and - for minus strand
file	file path where the hits is written to
gRNA.size	gRNA size, default 20
PAM	PAM as regular expression for appending to the gRNA, default NGG for SpCas9, change to TTTN for cpf1.
PAM.pattern	PAM as regular expression for filtering the hits, default N[A|G]G$ for spCas9. For cpf1, ^TTTN since it is a 5 prime PAM sequence.
max.mismatch	maximum mismatch allowed within the gRNA (excluding PAM portion) for filtering the hits, default 4
chrom.len	length of the matched chromosome

append	TRUE if append to existing file, false if start a new file
PAM.location	PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime
PAM.size	Size of PAM, default 3
allowed.mismatch.PAM	Maximum number of mismatches allowed in the offtargets comparing to the PAM sequence. Default to 1 for NGG PAM
seqs	DNAString object containing a DNA sequence.
baseEditing	Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.
targetBase	Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.
editingWindow	Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window to consider for the offtargets search only, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window, or you would like to include offtargets with the target base in a larger editing window.

Value

results are saved in the file specified by file

Author(s)

Lihua Julie Zhu

References

http://bioconductor.org/packages/2.8/bioc/vignettes/BSgenome/inst/doc/ GenomeSearching.pdf

See Also

offTargetAnalysis

Examples

```
if(interactive())
{
   gRNAPlusPAM <- DNAString("ACGTACGTACGTACTGACGTCGG")
   x <- DNAString("AAGCGCGATATGACGTACGTACGTACTGACGTCGG")
   chrom.len <- nchar(as.character(x))
   m <- matchPattern(gRNAPlusPAM, x)
   names(m) <- "testing"
   writeHits(gRNA = gRNAPlusPAM, seqname = "chr1",
       matches = m, strand = "+", file = "exampleWriteHits.txt",
       chrom.len = chrom.len, append = FALSE, seqs = x)
}
```

writeHits2 *Write the hits of sequence search to a file*

Description

write the hits of sequence search to a file, internal function used by searchHits

Usage

```
writeHits2(
  gRNA = NULL,
  seqname = NULL,
  matches = NULL,
  strand = NULL,
  file = tempfile(),
  gRNA.size = 20L,
  PAM = "NGG",
  PAM.pattern = "N[A|G]G$",
  max.mismatch = 4L,
  chrom.len = NULL,
  append = FALSE,
  PAM.location = "3prime",
  PAM.size = 3L,
  allowed.mismatch.PAM = 1L,
  BSgenomeName = NULL,
  baseEditing = FALSE,
  targetBase = "C",
  editingWindow = 4:8
)
```

Arguments

gRNA	DNAString object with gRNA sequence with PAM appended immediately after,e.g., ACGTACGTACGTACTGACGTCGG with 20bp gRNA sequence plus 3bp PAM sequence CGG
seqname	chromosome name as character, e.g., chr1
matches	XStringViews object storing matched chromosome locations
strand	strand of the match, + for plus strand and - for minus strand
file	file path where the hits is written to
gRNA.size	gRNA size, default 20
PAM	PAM as regular expression for filtering the hits, default NGG for spCas9. For cpf1, TTTN.
PAM.pattern	Regular expression of protospacer-adjacent motif (PAM), default N[A|G]G$ for spCas9. For cpf1, ^TTTN since it is a 5 prime PAM sequence
max.mismatch	maximum mismatch allowed within the gRNA (excluding PAM portion) for filtering the hits, default 4
chrom.len	length of the matched chromosome
append	TRUE if append to existing file, false if start a new file

`PAM.location` PAM location relative to gRNA. For example, spCas9 PAM is located on the 3 prime while cpf1 PAM is located on the 5 prime

`PAM.size` Size of PAM, default 3

`allowed.mismatch.PAM` Number of degenerative bases in the PAM sequence, default to 1 for N[AlG]G PAM

`BSgenomeName` BSgenome object. Please refer to available.genomes in BSgenome package. For example,

- BSgenome.Hsapiens.UCSC.hg19 - for hg19
- BSgenome.Mmusculus.UCSC.mm10 - for mm10
- BSgenome.Celegans.UCSC.ce6 - for ce6
- BSgenome.Rnorvegicus.UCSC.rn5 - for rn5
- BSgenome.Drerio.UCSC.danRer7 - for Zv9
- BSgenome.Dmelanogaster.UCSC.dm3 - for dm3

`baseEditing` Indicate whether to design gRNAs for base editing. Default to FALSE If TRUE, please set baseEditing = TRUE, targetBase and editingWidow accordingly.

`targetBase` Applicable only when baseEditing is set to TRUE. It is used to indicate the target base for base editing systems, default to C for converting C to T in the CBE system. Please change it to A if you intend to use the ABE system.

`editingWindow` Applicable only when baseEditing is set to TRUE. It is used to indicate the effective editing window to consider for the offtargets search only, default to 4 to 8 which is for the original CBE system. Please change it accordingly if the system you use have a different editing window, or you would like to include offtargets with the target base in a larger editing window.

Value

results are saved in the file specified by file

Author(s)

Lihua Julie Zhu

References

http://bioconductor.org/packages/2.8/bioc/vignettes/BSgenome/inst/doc/ GenomeSearching.pdf

See Also

offTargetAnalysis

Examples

```
library("BSgenome.Hsapiens.UCSC.hg19")
gRNAPlusPAM <- DNAString("ACGTACGTACGTACTGACGTCGG")
x <- DNAString("AAGCGCGATATGACGTACGTACGTACTGACGTCGG")
chrom.len <- nchar(as.character(x))
m <- matchPattern(gRNAPlusPAM, x)
names(m) <- "testing"
writeHits2(gRNA = gRNAPlusPAM, seqname = "chr1",
    PAM = "NGG", PAM.pattern = "NNN$", allowed.mismatch.PAM = 2,
    matches = m, strand = "+", file = "exampleWriteHits.txt",
    chrom.len = chrom.len, append = FALSE, BSgenomeName = Hsapiens)
```

Index