CRISPRseek user's guide

Lihua Julie Zhu, Michael Brodsky

October 27, 2020

Contents

1 Introduction

CRISPR-Cas9 nucleases and their derivatives have rapidly become widely used tools for both genome modification and regulation of gene expression. They can create genetic changes with high efficiency in human stem cells, in model organisms such as mice and Drosophila and in a wide variety of other organisms. CRISPR-Cas9 nucleases create double strand DNA breaks that can facilitate a variety of genome modifications including short insertions and/or deletions (indels) or specific sequence changes introduced by homology directed repair with a DNA donor molecule. The high activity and relative ease of construction has made CRISPR-Cas9 nucleases a popular replacement for related technologies such as zinc finger nucleases and TALENs. Derivatives of CRISPR-Cas9 complexes include nickases, which only cleave one DNA strand, and gene expression regulators, which lack any DNA cleavage activity but can increase or decrease gene transcription. CRISPR-Cas9 nucleases are composed of an RNA-protein complex that can target a variable sequence (guide RNA, abbreviated as "gRNA") that is directly adjacent to a constant motif (the "PAM" sequence). In the most widely use version from the species Streptococcus pyogenes, the gRNA is composed of a variable region of 20 bases and the preferred PAM sequence is an adjacent 3 base sequence of NGG (or NAG with lower activity). One potential limitation for CRISPR-Cas9 nucleases is that they can cleave at some sequences that do not precisely match the sequence targeted by the gRNA sequence. Thus, an important consideration for the design and application of CRISPR-Cas9 nucleases is the identification of gRNA regions with low rates of off-target cleavage.

Several computational analyses can assist with the construction and application of CRISPR-Cas9 nucleases with high on-target and low off-target cleavage. First, gRNA sequences can be evaluated for possible off-target sequences in the target genome. Second, sequences flanking possible off-target sequences can be reported to assist in the experimental analysis of off-target cleavage and to determine if these sequences are within critical regions for gene func-

tion such as exons. Third, specific arrangements of target sequences can be selected; one alternate approach to lower off-target rates is to introduce pairs of CRISPR-Cas9 nickases, which will only create double strand DNA breaks at genomic regions where the two sites have the proper spacing and orientation. Finally, in some applications, it is useful to use restriction enzyme sequences that overlap CRISPR-Cas9 target sites in order to monitor cleavage events. We developed *CRISPRseek* package that identifies candidate CRISPR-Cas9 target sequences within a given input sequence using a variety of experimentally useful constraints and also reports and ranks potential off-target sequences for each recovered target sequence. CRISPRseek will automatically find potential target sequences that are/are not present as pairs that can be used as double nickases or that have/don't have overlapping restriction enzyme cut site(s). It will then search genome-wide for off-targets with a user defined maximum number of mismatches, calculate the score of each off-target based on mismatches in the off-target and a penalty weight matrix, filter off-targets with user-defined criteria, and annotate off-targets with flanking sequences, and whether located in exon or not. Several reports are generated including a summary report with gRNAs ranked by total topN off-target score, restriction enzyme cut sites and possible paired gRNAs. Detailed paired gRNAs information, restriction enzyme cut sites, and off-target sequences and scores are stored in separate files in the output directory specified by the user. In total, four tab delimited files are generated in the output directory: OfftargetAnalysis.xls (off-target details), Summary.xls (gRNA summary), REcutDetails.xls (restriction enzyme cut sites of each gRNA), and pairedgRNAs.xls (potential paired gRNAs). These reports provide a comprehensive set of information to identify, select and utilize Streptococcus pyogenes CRISPR-Cas9 nucleases and their derivatives. The package can also be readily modified to accept different gRNA and PAM sequence requirements for CRISPR-Cas9 complexes from other bacterial species that can be used to target alternative genomic sequences. The package can also be modified to incorporate improved weight matrices or scoring.method for scoring off-target sequences as new experimental and computational results become available for CRISPR-Cas9 nucleases for Streptococcus pyogenes and other species.

2 Examples of using CRISPRseek

In this guide, we will illustrate five different gRNA search scenarios with a human sequence. First load *CRISPRseek*, *BSgenome.Hsapiens.UCSC.hg19* and *TxDb.Hsapiens.UCSC.hg19.knownGene*. Then specify the sequence file path as inputFilePath, a fasta/fastq file containing a genomic sequence, restriction enzyme pattern file as REpatternFile and output directory as outputDir. Once the analysis is done, analysis results will be saved in the output directory.

To find BSgenome of other species, 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, and *BSgenome.Dmelanogaster.UCSC.dm3* for dm3

To create and use TxDb objects, please refer to the *GenomicFeatures* package. For a list of existing TxDb objects, please search for annotation package starting with Txdb at http://www.bioconductor.org/packages/release/BiocViews.html, such as *TxDb.Rnorvegicus.UCSC.rn5.refG* for rat, *TxDb.Mmusculus.UCSC.mm10.knownGene* for mouse, *TxDb.Hsapiens.UCSC.hg19.knownGene* for human, *TxDb.Dmelanogaster.UCSC.dm3.ensGene* for Drosophila and *TxDb.Celegans.UCSC.ce6.ensGene* for C.elegans

org.Hs.eg.db is the gene ID mapping package for human. For a list of existing OrgDb packages, please search for OrgDb at http://www.bioconductor.org/packages/release/BiocViews.html, such as *org.Rn.eg.db* for rat *org.Mm.eg.db* for mouse *org.Dm.eg.db* for Drosophila *org.Ce.eg.db* for C.elegans

```
> library(CRISPRseek)
> library(BSgenome.Hsapiens.UCSC.hg19)
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> library(org.Hs.eg.db)
> outputDir <- getwd()
> inputFilePath <- system.file('extdata', 'inputseq.fa', package = 'CRISPRseek')
> REpatternFile <- system.file('extdata', 'NEBenzymes.fa', package = 'CRISPRseek')
```

2.1 Scenario 1: Finding paired gRNAs with restriction enzyme cut site(s)

Paired gRNAs in proper spacing and orientation give more specificity and gRNAs overlap with restriction enzyme cut sites facilitates cleavage monitoring. Calling the function `offTargetAnalysis` with findPairedgRNAOnly=TRUE and findgRNAsWithREcutOnly=TRUE results in searching, scoring and annotating gRNAs that are in paired configuration and at least one of the pairs overlap a restriction enzyme cut site. To be considered as a pair, gap between forward gRNA and the corresponding reverse gRNA needs to be (min.gap, max.gap) inclusive and the reverse gRNA must sit before the forward gRNA. The default (min.gap, max.gap) is (0,20). Please note that chromToSearch is set to chrX here for speed purpose, usually you would set it to all, which is the default. In order for a gRNA to be considered overlap with restriction enzyme cut site, the enzyme cut pattern must overlap with one of the gRNA positions specified in overlap.gRNA.positions, default position 17 and 18. Please note that max.mismatch

allowed for off-target finding is set to 4 by default, set it to a larger number will significantly slow down the search. org.Hs.egSYMBOL is entrezID to gene symbol mapping in org.Hs.eg.db package for human. For detailed parameter settings using function `offTargetAnalysis`, please type help(offTargetAnalysis)

```
> results <- offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly = TRUE,
+ REpatternFile = REpatternFile, findPairedgRNAOnly = TRUE,
+ BSgenomeName = Hsapiens, chromToSearch ="chrX", min.gap = 0, max.gap = 20,
+ txdb = TxDb.Hsapiens.UCSC.hg19.knownGene, orgAnn = org.Hs.egSYMBOL,
+ max.mismatch = 0,overlap.gRNA.positions = c(17, 18),
+ outputDir = outputDir,overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.2 Scenario 2: Finding paired gRNAs with/without restriction enzyme cut site(s)

Calling the function `offTargetAnalysis` with findPairedgRNAOnly = TRUE and findgRNAsWithREcutOnly = FALSE results in searching, scoring and annotating gRNAs that are in paired configuration without requiring overlap any restriction enzyme cut site. The gRNAs will be annotated with restriction enzyme cut sites for users to review later.

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

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.3 Scenario 3: Finding all gRNAs with restriction enzyme cut site(s)

Calling the function `offTargetAnalysis` with findPairedgRNAOnly=FALSE and findgRNAsWithREcutOnly = TRUE results in searching, scoring and annotating all gRNAs (paired and not paired) overlap restriction enzyme cut site(s) and off-targets. The gRNAs will be annotated with paired information for users to review later.

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

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.4 Scenario 4: Finding all gRNAs

Calling the function `offTargetAnalysis` with findPairedgRNAOnly = FALSE and findgRNAsWithREcutOnly = FALSE results in searching, scoring and annotating all gRNAs and off-targets. The gRNAs will be annotated with paired information and restriction enzyme cut sites for users to review later. Please note that this search will be the slowest among all type of searches aforementioned.

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

Validating input ...
Searching for gRNAs ...
```

```
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.5 Scenario 5: Target and off-target analysis for user specified gRNAs

Calling the function `offTargetAnalysis` with findgRNAs = FALSE results in target and off-target searching, scoring and annotating for the input gRNAs. The gRNAs will be annotated with restriction enzyme cut sites for users to review later. However, paired information will not be available.

```
> gRNAFilePath <- system.file('extdata', 'testHsap_GATA1_ex2_gRNA1.fa',
+ package = 'CRISPRseek')
> results <- offTargetAnalysis(inputFilePath = gRNAFilePath,
+ findgRNAsWithREcutOnly = FALSE, REpatternFile = REpatternFile,
+ findPairedgRNAOnly = FALSE, findgRNAs = FALSE,
+ BSgenomeName = Hsapiens, chromToSearch = 'chrX',
+ txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
+ orgAnn = org.Hs.egSYMBOL,
+ max.mismatch = 1, outputDir = outputDir, overwrite = TRUE)

Validating input ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.6 Scenario 6. Quick gRNA finding without target or off-target analysis

Calling the function `offTargetAnalysis` with chromToSearch = "" results in quick gRNA search without performing on-target and off-target analysis. Parameters findgRNAsWithREcutOnly and findPairedgRNAOnly can be tuned to indicate whether searching for gRNAs overlap restriction enzyme cut sites or not, and whether searching for gRNAs in paired configuration or not.

```
> results <- offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly = TRUE,
+ REpatternFile = REpatternFile,findPairedgRNAOnly = TRUE,
+ chromToSearch = "", outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.7 Scenario 7. Quick gRNA finding with gRNA efficacy prediction

Calling the function `offTargetAnalysis` with max.mismatch = 0 results in quick gRNA search with gRNA efficacy prediction without off-target analysis. Parameter rule.set can be set to CRISPRscan published in 2015 or Root_RuleSet2_2016 to calculate efficacy using the rule set 2 published in 2016. By default, gRNA efficacy is predicted using the rule set 1 published in 2014. To use the rule set 2, first install python 2.7, then install the python packages: scikit-learn 0.16.1, pickle, pandas, numpy nd scipy. In a R session, type the following script to use python 2.7 since rule set 2 is implemented in python 2.7. Sys.setenv(PATH = paste(" /anaconda2/bin", Sys.getenv("PATH"), sep=":")) system("python –version") should output Python 2.7.15. In addition, parameters findgRNAsWithREcutOnly and findPairedgRNAOnly can be tuned to indicate whether searching for gRNAs overlap restriction enzyme cut sites or not, and whether searching for gRNAs in paired configuration or not.

```
> results <- offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly = TRUE,
+ annotateExon = FALSE,findPairedgRNAOnly = TRUE, chromToSearch = "chrX",
+ max.mismatch = 0, BSgenomeName = Hsapiens, outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

Here is an example on how to use CRISPRscan to calculate efficacy. Please remember to reset baseBeforegRNA, baseAfterPAM, rule.set, and featureWeightMatrixFile.

```
> results <- offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly = TRUE,
+     annotateExon = FALSE,findPairedgRNAOnly = TRUE, chromToSearch = "chrX",
+     max.mismatch = 0, BSgenomeName = Hsapiens,
+     rule.set = "CRISPRscan",
+     baseBeforegRNA = 6,
+     baseAfterPAM = 6,
+     featureWeightMatrixFile =  system.file("extdata", "Morenos-Mateo.csv",
+             package = "CRISPRseek"),
+     outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add paired information...
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.8 Scenario 8. Find potential gRNAs preferentially targeting one of two alleles without running time-consuming off-target analysis on all possible gRNAs.

Below is an example to search for all gRNAs that target at least one of the alleles. Two files are provided containing sequences that differ by a single nucleotide polymorphism (SNP). The results are saved in file scoresFor2InputSequences.xls in outputDir directory.

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

search for gRNAs for input file1...
Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/rs362331C.fa-Oct-27-2020/
search for gRNAs for input file2...
Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/rs362331T.fa-Oct-27-2020/
[1] "Scoring ..."
>>> Finding all hits in sequence rs362331T ...
```

```
>>> DONE searching
finish off-target search in sequence 2
>>> Finding all hits in sequence rs362331C ...
>>> DONE searching
finish off-target search in sequence 1
finish feature vector building
finish score calculation
[1] "Done!"

> seqs

              name             gRNAPlusPAM            targetInSeq1
1   rs362331T_gR9r GTAGATGAGGGAGCAGGCGTNGG GTGGATGAGGGAGCAGGCGTGGG
2  rs362331T_gR38f CTACTGTGTGCACTTCATCCNGG CCACTGTGTGCACTTCATCCTGG
3  rs362331T_gR22r TGAAGTGCACACAGTAGATGNGG TGAAGTGCACACAGTGGATGAGG
4  rs362331T_gR21r GAAGTGCACACAGTAGATGANGG GAAGTGCACACAGTGGATGAGGG
5  rs362331T_gR15r CACACAGTAGATGAGGGAGCNGG CACACAGTGGATGAGGGAGCAGG
6  rs362331T_gR10r AGTAGATGAGGGAGCAGGCGNGG AGTGGATGAGGGAGCAGGCGTGG
7   rs362331C_gR9r GTGGATGAGGGAGCAGGCGTNGG GTGGATGAGGGAGCAGGCGTNGG
8  rs362331C_gR38f CCACTGTGTGCACTTCATCCNGG CCACTGTGTGCACTTCATCCNGG
9  rs362331C_gR28r CCAGGATGAAGTGCACACAGNGG CCAGGATGAAGTGCACACAGNGG
10 rs362331C_gR22r TGAAGTGCACACAGTGGATGNGG TGAAGTGCACACAGTGGATGNGG
11 rs362331C_gR21r GAAGTGCACACAGTGGATGANGG GAAGTGCACACAGTGGATGANGG
12 rs362331C_gR15r CACACAGTGGATGAGGGAGCNGG CACACAGTGGATGAGGGAGCNGG
13 rs362331C_gR10r AGTGGATGAGGGAGCAGGCGNGG AGTGGATGAGGGAGCAGGCGNGG
              targetInSeq2 guideAlignment2OffTarget offTargetStrand
1  GTAGATGAGGGAGCAGGCGTNGG     ..G.................               -
2  CTACTGTGTGCACTTCATCCNGG     .C.................               +
3  TGAAGTGCACACAGTAGATGNGG     ..............G....               -
4  GAAGTGCACACAGTAGATGANGG     ..............G.....               -
5  CACACAGTAGATGAGGGAGCNGG     ........G...........               -
6  AGTAGATGAGGGAGCAGGCGNGG     ...G...............               -
7  GTAGATGAGGGAGCAGGCGTGGG     ..A.................               -
8  CTACTGTGTGCACTTCATCCTGG     .T.................               +
9  CCAGGATGAAGTGCACACAGTAG     ...................               -
10 TGAAGTGCACACAGTAGATGAGG     ..............A....               -
11 GAAGTGCACACAGTAGATGAGGG     ..............A.....               -
12 CACACAGTAGATGAGGGAGCAGG     ........A...........               -
13 AGTAGATGAGGGAGCAGGCGTGG     ...A...............               -
   scoreForSeq1 scoreForSeq2 mismatch.distance2PAM n.mismatch       offTarget
1          98.6          100                    18          1  rs362331C:3-25
2           100          100                    19          1 rs362331C:22-44
3          17.2          100                     5          1 rs362331C:16-38
4          26.8          100                     6          1 rs362331C:15-37
5          61.1          100                    12          1  rs362331C:9-31
6           100          100                    17          1  rs362331C:4-26
7           100         98.6                    18          1  rs362331T:3-25
8           100          100                    19          1 rs362331T:22-44
9           100          100                                0 rs362331T:22-44
10          100         17.2                     5          1 rs362331T:16-38
11          100         26.8                     6          1 rs362331T:15-37
12          100         61.1                    12          1  rs362331T:9-31
13          100          100                    17          1  rs362331T:4-26
   targetSeqName                gRNAefficacy scoreDiff
1      rs362331T extended sequence too short      -1.4
2      rs362331T extended sequence too short       0.0
3      rs362331T           0.118513953473509     -82.8
4      rs362331T           0.229116108934864     -73.2
5      rs362331T          0.0546177253478725     -38.9
6      rs362331T           0.161377523263354       0.0
7      rs362331C extended sequence too short       1.4
8      rs362331C extended sequence too short       0.0
9      rs362331C extended sequence too short       0.0
10     rs362331C          0.0978516718170484      82.8
11     rs362331C           0.129816665166513      73.2
```

```
12     rs362331C          0.0169291602963948      38.9
13     rs362331C           0.140082863263818       0.0
```

rs362331C.fa and rs362331T.fa are the names of the two input files. The output file will list all of the possible gRNA sequences for each of the two input sequences and provide a cleavage score for each of the two input sequences. To preferentially target one allele, select gRNA sequences that have the lowest score for the other allele. Selected gRNAs can then be examined for off-target sequences as described in Step 6.

2.9 Scenario 9. gRNA search and offTarget analysis of super long input sequence (longer than 200kb)

Calling the function `offTargetAnalysis` with annotatePaired = FALSE, enable.multicore = TRUE and set n.cores.max will improve the performance. We also suggest split the super long sequence into smaller chunks and perform offTarget analysis for each subsequence separately (Thank Alex Williams for sharing this use case at https://support.bioconductor.org/p/72994/). In addition, please remember to use repeat masked sequence as input.

```
> results <- offTargetAnalysis(inputFilePath, annotatePaired = FALSE,
+     chromToSearch = "chrX",
+     enable.multicore = TRUE, n.cores.max = 10, annotateExon = FALSE,
+     max.mismatch = 0, BSgenomeName = Hsapiens,
+     outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.10 Scenario 10. Output cutting frerquency determination (CFD) score for offtargets

Calling the function `offTargetAnalysis` with scoring.method set to CFDscore and PAM.pattern as NNG or NGA will output CFD score using the algorithm by Doench et al., 2016, which models the effects of both mismatch position and mimatch type on cutting frequency. By default, scoring.method is set to Hsu-Zhang, which only models the effect of mismatch position.

```
> results <- offTargetAnalysis(inputFilePath, annotatePaired = FALSE,
+     scoring.method = "CFDscore",
+     PAM.pattern  = "NNG$|NGN$",
+     chromToSearch = "chrX",
+     annotateExon = FALSE,
+     max.mismatch = 2, BSgenomeName = Hsapiens,
+     outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done annotating
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/
```

2.11 Scenario 11. Design gRNAs for base editing systems

Cytosine or adenine base editors (CBEs or ABEs) can introduce specific DNA C-to-T or A-to-G alterations. Calling the function `offTargetAnalysis` with baseEditing set to TRUE, and targetBase, editingWindow and editingWindow.offtarget set appropriately. By default, targetBase is set to C, editingWindow to 4 to 8 and editingWindow.offtargets to 4 to 8 for the CBE system developed in the Liu Laboratory.

```
> offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly =  FALSE,
+                   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, PAM.location = "5prime",
+                   PAM = "TGT", PAM.pattern = "^T[A|G]N", allowed.mismatch.PAM = 2,
+                   subPAM.position = c(1,2), baseEditing = TRUE,
+                   editingWindow = 10:20, targetBase = "A")

Validating input ...
Searching for gRNAs ...
No gRNAs found in the input sequence Hsap_GATA1_ex2no gRNAs found!
```

2.12 Scenario 12. Design gRNAs and pegRNAs for the Prime Editor

Recently, the Liu Laboratory developed the Prime Editor (PE), which is more versatile and flexible with high efficacy and without the need to make a DSB or providing donor template. It can be used to make all possible 12 single base changes, 1-44 bp insertions, or 1-80 bp deletions. This editing system can be programmed to correct about 89 percent of human pathagenic variants. To design gRNAs and pegRNAs for PE, you need to set primeEditing to TRUE, set seq.length.changed, bp.after.target.end, PBS.length (default 13), RT.template.length (default 8 to 28), min.gap (default 40), max.gap (default 90), findPairedgRNAOnly (needs to be set to TRUE), paired.orientation (needs to be set to "PAMin"), target.start, target.end, and correct.seq accordingly.

Please type help(offTargetAnalysis) for detailed description of these parameters and examples for designing PE.

2.13 Scenario 13. Predict indels and their frequencies for Cas9 genome editing systems

Calling the function `offTargetAnalysis` with predIndelFreq set to TRUE. The first command outputs the gRNA name,gRNAPlusPAM sequence, offtarget sequence, gene name, predicted offtarget score, genomic location, and predicted efficacy. The second command outputs the indel locations and frequencies for the first target site. The summary file contains Shannon entropy and the fraction of frameshift of mutational outcomes for each target. Currently, only the Lindel method by Chen et al., 2019 is available. Please type help(predictRelativeFreqIndels) for more details.

```
> resultsIndelF <- offTargetAnalysis(
+                  inputFilePath, findgRNAsWithREcutOnly =  FALSE,
+                  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,
+                  scoring.method = "CFDscore",
+                  PAM.pattern  = "NNG$|NGN$",
+                  predIndelFreq = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequence chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
```

```
Done annotating
Add RE information...
write gRNAs to bed file...
Scan for REsites in flanking region...
Done. Please check output files in directory
 /tmp/RtmpPbacoX/Rbuild52804e64f199/CRISPRseek/vignettes/

> if(exists("resultsIndelF"))
+ {
+     print(head(resultsIndelF$indelFreq[[1]]))
+     mapply(write.table, resultsIndelF$indelFreq, file=paste0(names(resultsIndelF$indelFreq), '.xls'),
+        sep = "\t", row.names = FALSE)
+  }

     Indels
[1,] "GGTGTGGAGGACACCAGAGCAGGATCCACA | AACTGGGGGAGGGGCTCTGAGGTCCCCAGG"
[2,] "GGTGTGGAGGACACCAGAGCAGGATCCACA X AACTGGGGGAGGGGCTCTGAGGTCCCCAGG"
[3,] "GGTGTGGAGGACACCAGAGCAGGATCCAC- | ---TGGGGGAGGGGCTCTGAGGTCCCCAGG"
[4,] "GGTGTGGAGGACACCAGAGCAGGATCCACA | -ACTGGGGGAGGGGCTCTGAGGTCCCCAGG"
[5,] "GGTGTGGAGGACACCAGAGCAGGATCCACA AAAACTGGGGGAGGGGCTCTGAGGTCCCCAGG"
[6,] "GGTGTGGAGGACACCAGAGCAGGATCCACA | --CTGGGGGAGGGGCTCTGAGGTCCCCAGG"
     Frequency     Location
[1,] "0"           ""
[2,] "11.09818517" "I3+X"
[3,] "10.52657092" "D4  -1"
[4,] "7.30665131"  "D1  0"
[5,] "3.89461931"  "I2+AA"
[6,] "3.31476489"  "D2  0"
$`Hsap_GATA1_ex2_gR7r,CCAGAGCAGGATCCACAAACNGG,CCAGAGCAGGATCCACAAACTGG`
NULL

$`Hsap_GATA1_ex2_gR40f,TGTCCTCCACACCAGAATCANGG,TGTCCTCCACACCAGAATCAGGG`
NULL

$`Hsap_GATA1_ex2_gR39f,GTGTCCTCCACACCAGAATCNGG,GTGTCCTCCACACCAGAATCAGG`
NULL

$`Hsap_GATA1_ex2_gR20r,GGTGTGGAGGACACCAGAGCNGG,GGTGTGGAGGACACCAGAGCAGG`
NULL

$`Hsap_GATA1_ex2_gR17f,CCAGTTTGTGGATCCTGCTCNGG,CCAGTTTGTGGATCCTGCTCTGG`
NULL
```

3 References

References

[1] Lihua Julie Zhu. Overview of guide RNA design tools for CRISPR-Cas9 genome editing technology. Frontiers in Biology. August 2015, Volume 10, Issue 4, pp 289-296

[2] Mali P. et al., CAS9 transcriptional activators for target specificity screening and paired nickases for cooperative genome engineering. Nat Biotechnol. 2013. 31(9):833-8

[3] Hsu, P.D. et al., DNA targeting specificity of rNA-guided Cas9 nucleases. Nat Biotechnol. 2013. 31:827-834.

[4] Doench et al., 2014 Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. Nat Biotechnol. 2014 Sep 3. doi: 10.1038 nbt.3026

[5] 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

[6] Moreno-Mateos, M., Vejnar, C., Beaudoin, J. et al. CRISPRscan: designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo. Nat Methods 12, 982âĂŞ988 (2015) doi:10.1038/nmeth.3543

[7] Doench et al. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat Biotechnol. 2016 Jan 18. doi:10.1038/nbt.3437

[8] Komor, A. C., Kim, Y. B., Packer, M. S., Zuris, J. A. and Liu, D. R. Programmable editing of a target base in genomic DNA without double-stranded DNA cleavage. Nature 533, 420âĂŞ424 (2016).

[9] Gaudelli, N. M. et al. Programmable base editing of A or T to G or C in genomic DNA without DNA cleavage. Nature 551, 464âĂŞ471 (2017).

[10] 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 7989âĂŞ8003, https://doi.org/10.1093/nar/gkz487.

4 Session Info

```
> sessionInfo()

R version 4.0.3 (2020-10-10)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.5 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.12-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.12-bioc/R/lib/libRlapack.so
```

```
locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] org.Hs.eg.db_3.12.0
 [2] TxDb.Hsapiens.UCSC.hg19.knownGene_3.2.2
 [3] GenomicFeatures_1.42.0
 [4] AnnotationDbi_1.52.0
 [5] Biobase_2.50.0
 [6] BSgenome.Hsapiens.UCSC.hg19_1.4.3
 [7] BSgenome_1.58.0
 [8] rtracklayer_1.50.0
 [9] GenomicRanges_1.42.0
[10] GenomeInfoDb_1.26.0
[11] CRISPRseek_1.30.0
[12] Biostrings_2.58.0
[13] XVector_0.30.0
[14] IRanges_2.24.0
[15] S4Vectors_0.28.0
[16] BiocGenerics_0.36.0

loaded via a namespace (and not attached):
 [1] MatrixGenerics_1.2.0        httr_1.4.2
 [3] bit64_4.0.5                 jsonlite_1.7.1
 [5] assertthat_0.2.1            askpass_1.1
 [7] BiocManager_1.30.10         BiocFileCache_1.14.0
 [9] blob_1.2.1                  GenomeInfoDbData_1.2.4
[11] Rsamtools_2.6.0             yaml_2.2.1
[13] progress_1.2.2              pillar_1.4.6
[15] RSQLite_2.2.1               lattice_0.20-41
[17] glue_1.4.2                  reticulate_1.18
[19] digest_0.6.27               htmltools_0.5.0
[21] Matrix_1.2-18               XML_3.99-0.5
```

```
      [23] pkgconfig_2.0.3             biomaRt_2.46.0
      [25] zlibbioc_1.36.0             purrr_0.3.4
      [27] BiocParallel_1.24.0         tibble_3.0.4
      [29] openssl_1.4.3               generics_0.0.2
      [31] ellipsis_0.3.1              SummarizedExperiment_1.20.0
      [33] magrittr_1.5                crayon_1.3.4
      [35] memoise_1.1.0               evaluate_0.14
      [37] MASS_7.3-53                 xml2_1.3.2
      [39] tools_4.0.3                 hash_2.2.6.1
      [41] data.table_1.13.2           prettyunits_1.1.1
      [43] hms_0.5.3                   BiocStyle_2.18.0
      [45] lifecycle_0.2.0             matrixStats_0.57.0
      [47] stringr_1.4.0               Rhdf5lib_1.12.0
      [49] DelayedArray_0.16.0         ade4_1.7-15
      [51] compiler_4.0.3              rlang_0.4.8
      [53] rhdf5_2.34.0                grid_4.0.3
      [55] RCurl_1.98-1.2              rhdf5filters_1.2.0
      [57] rappdirs_0.3.1              bitops_1.0-6
      [59] rmarkdown_2.5               curl_4.3
      [61] DBI_1.1.0                   R6_2.4.1
      [63] GenomicAlignments_1.26.0    dplyr_1.0.2
      [65] knitr_1.30                  seqinr_4.2-4
      [67] bit_4.0.4                   stringi_1.5.3
      [69] Rcpp_1.0.5                  vctrs_0.3.4
      [71] tidyselect_1.1.0            dbplyr_1.4.4
      [73] xfun_0.18
```