CRISPRseek user's guide

Lihua Julie Zhu, Michael Brodsky

July 22, 2014

Contents

1 Introduction **1**

2 Examples of using CRISPRseek **2**
2.1 Scenario 1: Finding paired gRNAs with restriction enzyme cut site(s) 3
2.2 Scenario 2: Finding paired gRNAs with/without restriction enzyme cut site(s) 4
2.3 Scenario 3: Finding all gRNAs with restriction enzyme cut site(s) 4
2.4 Scenario 4: Finding all gRNAs 4
2.5 Scenario 5: Target and off-target analysis for user specified gRNAs 5
2.6 Scenario 6. Quick gRNA finding without target or off-target analysis 5
2.7 Scenario 7. Find potential gRNAs preferentially targeting one of two alleles without running time-consuming off-target analysis on all possible gRNAs. 6

3 References **7**

4 Session Info **7**

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 function 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 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, re-

striction 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 TranscriptDb objects, please refer to the *GenomicFeatures* package. For a list of existing TranscriptDb objects, please search for annotation package starting with Txdb at http://www.bioconductor.org/packages/release/BiocViews.html, 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 and *TxDb.Celegans.UCSC.ce6.ensGene* for C.elegans

```
> library(CRISPRseek)
> library(BSgenome.Hsapiens.UCSC.hg19)
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> 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. For detailed parameter settings using function `offTargetAnalysis`, please type help(offTargetAnalysis)

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

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequences chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/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.

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

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequences chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/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.

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

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequences chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/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.

```
> offTargetAnalysis(inputFilePath, findgRNAsWithREcutOnly = FALSE,
+ REpatternFile = REpatternFile,findPairedgRNAOnly = FALSE,
+ BSgenomeName = Hsapiens, chromToSearch = "chrX",
```

```
+ txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
+ max.mismatch = 1, outputDir = outputDir, overwrite = TRUE)

Validating input ...
Searching for gRNAs ...
>>> Finding all hits in sequences chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/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')
> offTargetAnalysis(inputFilePath = gRNAFilePath,
+ findgRNAsWithREcutOnly = TRUE, REpatternFile = REpatternFile,
+ findPairedgRNAOnly = FALSE, findgRNAs = FALSE,
+ BSgenomeName = Hsapiens, chromToSearch = 'chrX',
+ txdb = TxDb.Hsapiens.UCSC.hg19.knownGene,
+ max.mismatch = 1, outputDir = outputDir, overwrite = TRUE)

Validating input ...
>>> Finding all hits in sequences chrX ...
>>> DONE searching
Building feature vectors for scoring ...
Calculating scores ...
Annotating, filtering and generating reports ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/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.

```
> 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  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/CRISPRseek/vignettes/
  A DNAStringSet instance of length 2
    width seq                                                 names
[1]    23 TGTCCTCCACACCAGAATCAGGG                             gRNAf1_Hsap_GATA1...
[2]    23 CCAGAGCAGGATCCACAAACTGG                             gRNAr1_Hsap_GATA1...
```

2.7 Scenario 7. 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)

Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/CRISPRseek/vignettes/rs362331C.fa/
Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /private/tmp/RtmpA6vTZN/Rbuild80b38889c04/CRISPRseek/vignettes/rs362331T.fa/
[1] "Scoring ..."
[1] "Done!"

> seqs

      name                           gRNAPlusPAM               targetInSeq1
 [1,] "gRNAr6_rs362331CStart25End3"  "GTGGATGAGGGAGCAGGCGTGGG" "GTGGATGAGGGAGCAGGCGTGGG"
 [2,] "gRNAr5_rs362331TStart25End3"  "GTAGATGAGGGAGCAGGCGTGGG" "GTGGATGAGGGAGCAGGCGTGGG"
 [3,] "gRNAr5_rs362331CStart26End4"  "AGTGGATGAGGGAGCAGGCGTGG" "AGTGGATGAGGGAGCAGGCGTGG"
 [4,] "gRNAr4_rs362331TStart26End4"  "AGTAGATGAGGGAGCAGGCGTGG" "AGTGGATGAGGGAGCAGGCGTGG"
 [5,] "gRNAr4_rs362331CStart31End9"  "CACACAGTGGATGAGGGAGCAGG" "CACACAGTGGATGAGGGAGCAGG"
 [6,] "gRNAr3_rs362331TStart31End9"  "CACACAGTAGATGAGGGAGCAGG" "CACACAGTGGATGAGGGAGCAGG"
 [7,] "gRNAr3_rs362331CStart37End15" "GAAGTGCACACAGTGGATGAGGG" "GAAGTGCACACAGTGGATGAGGG"
 [8,] "gRNAr2_rs362331TStart37End15" "GAAGTGCACACAGTAGATGAGGG" "GAAGTGCACACAGTGGATGAGGG"
 [9,] "gRNAr2_rs362331CStart38End16" "TGAAGTGCACACAGTGGATGAGG" "TGAAGTGCACACAGTGGATGAGG"
[10,] "gRNAr1_rs362331TStart38End16" "TGAAGTGCACACAGTAGATGAGG" "TGAAGTGCACACAGTGGATGAGG"
[11,] "gRNAr1_rs362331CStart44End22" "CCAGGATGAAGTGCACACAGTGG" "CCAGGATGAAGTGCACACAGTGG"
[12,] "gRNAf1_rs362331TStart22End44" "CTACTGTGTGCACTTCATCCTGG" "CCACTGTGTGCACTTCATCCTGG"
[13,] "gRNAf1_rs362331CStart22End44" "CCACTGTGTGCACTTCATCCTGG" "CCACTGTGTGCACTTCATCCTGG"
      targetInSeq2              guideAlignment2OffTarget offTargetStrand scoreForSeq1
 [1,] "GTAGATGAGGGAGCAGGCGTGGG" "..A................."   "-"             "100"
 [2,] "GTAGATGAGGGAGCAGGCGTGGG" "..G................."   "-"             "98.6"
 [3,] "AGTAGATGAGGGAGCAGGCGTGG" "...A................"   "-"             "100"
 [4,] "AGTAGATGAGGGAGCAGGCGTGG" "...G................"   "-"             "100"
 [5,] "CACACAGTAGATGAGGGAGCAGG" "........A..........."   "-"             "100"
 [6,] "CACACAGTAGATGAGGGAGCAGG" "........G..........."   "-"             "61.1"
 [7,] "GAAGTGCACACAGTAGATGAGGG" "..............A....."   "-"             "100"
 [8,] "GAAGTGCACACAGTAGATGAGGG" "..............G....."   "-"             "26.8"
 [9,] "TGAAGTGCACACAGTAGATGAGG" "..............A...."   "-"             "100"
[10,] "TGAAGTGCACACAGTAGATGAGG" "..............G...."   "-"             "17.2"
[11,] "CCAGGATGAAGTGCACACAGTAG" "..................."   "-"             "100"
[12,] "CTACTGTGTGCACTTCATCCTGG" ".C................."   "+"             "100"
[13,] "CTACTGTGTGCACTTCATCCTGG" ".T................."   "+"             "100"
      scoreForSeq2 mismatch.distance2PAM n.mismatch targetSeqName scoreDiff
 [1,] "98.6"       "18"                  "1"        "rs362331C"    "1.4"
 [2,] "100"        "18"                  "1"        "rs362331T"    "-1.4"
 [3,] "100"        "17"                  "1"        "rs362331C"    "0"
 [4,] "100"        "17"                  "1"        "rs362331T"    "0"
 [5,] "61.1"       "12"                  "1"        "rs362331C"    "38.9"
 [6,] "100"        "12"                  "1"        "rs362331T"    "-38.9"
 [7,] "26.8"       "6"                   "1"        "rs362331C"    "73.2"
 [8,] "100"        "6"                   "1"        "rs362331T"    "-73.2"
 [9,] "17.2"       "5"                   "1"        "rs362331C"    "82.8"
[10,] "100"        "5"                   "1"        "rs362331T"    "-82.8"
[11,] "100"        ""                    "0"        "rs362331C"    "0"
```

```
[12,] "100"       "19"                "1"       "rs362331T"   "0"
[13,] "100"       "19"                "1"       "rs362331C"   "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.

3 References

References

[1] 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

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

[3] Lihua Julie Zhu, Benjamin R. Holmes, Neil Aronin and Michael Brodsky. CRISPRseek: A Bioconductor package to help with the design of guide RNAs in CRISPR-Cas9 systems for targeted genome editing. (In preparation)

4 Session Info

```
> sessionInfo()

R version 3.1.1 (2014-07-10)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
 [1] TxDb.Hsapiens.UCSC.hg19.knownGene_2.14.0 GenomicFeatures_1.16.2
 [3] AnnotationDbi_1.26.0                     Biobase_2.24.0
 [5] BSgenome.Hsapiens.UCSC.hg19_1.3.1000     CRISPRseek_1.0.3
 [7] BSgenome_1.32.0                          GenomicRanges_1.16.3
 [9] GenomeInfoDb_1.0.2                       Biostrings_2.32.1
[11] XVector_0.4.0                            IRanges_1.22.9
```

```
[13] BiocGenerics_0.10.0

loaded via a namespace (and not attached):
 [1] BatchJobs_1.3           BBmisc_1.7              BiocParallel_0.6.1
 [4] BiocStyle_1.2.0         biomaRt_2.20.0          bitops_1.0-6
 [7] brew_1.0-6              checkmate_1.2           codetools_0.2-8
[10] DBI_0.2-7               digest_0.6.4            fail_1.2
[13] foreach_1.4.2           GenomicAlignments_1.0.3 iterators_1.0.7
[16] RCurl_1.95-4.1          Rsamtools_1.16.1        RSQLite_0.11.4
[19] rtracklayer_1.24.2      sendmailR_1.1-2         stats4_3.1.1
[22] stringr_0.6.2           tools_3.1.1             XML_3.98-1.1
[25] zlibbioc_1.10.0
```