GUIDEseq user's guide

Lihua Julie Zhu, Michael Lawrence, Ankit Gupta,
Alper Kucukural, Manuel Garber, Scot Wolfe

October 28, 2015

Contents

1 Introduction

The most recently developed genome editing system, CRISPR-Cas9 has greater inherent flexibility than prior programmable nuclease platforms because sequence-specific recognition resides primarily within the associated sgRNA, which permits a simple alteration of its recognition sequence. The short Protospacer Adjacent Motif (PAM), which is recognized by Cas9, is the chief constraint on the target site design density. Because of its simplicity and efficacy, this technology is revolutionizing biological studies and holds tremendous promise for therapeutic applications(Ledford, 2015; Cox et al., 2015).

However, imperfect cleavage specificity of CRISPR/Cas9 nuclease within the genome is a cause for concern for its therapeutic application. S. pyogenes Cas9 (SpyCas9)-based nucleases can cleave an imperfect heteroduplex formed between the guide sequence and a DNA sequence containing a functional PAM where the number, position and type of base mismatches can impact the level of activity (Hsu et al., 2013; Mali et al., 2013; Fu et al., 2013). This degree of promiscuity is problematic for therapeutic applications, since the generation of DNA breaks at unintended (off-target) sites has the potential to

alter gene expression and function through direct mutagenesis or the generation of genomic rearrangements. Experimentally defining the number and activity of off-target sites for a given Cas9/sgRNA complex genome-wide is critical to assess and improve nuclease precision.

A new suite of genome-wide off-target detection methods have recently been described that can identify sites with low cleavage activity within a population of nuclease-treated cells. One of the most sensitive and straightforward methods to employ is GUIDE-seq (Tsai et al., 2015). This method relies on erroneous NHEJ-mediated DNA repair to capture co-introduced blunt-ended double stranded oligonucleotides (dsODNs) at Cas9-induced breakpoints within the genome. The GUIDE-seq dsODNs display high insertion frequency (up to 50% of the measured indel rate (Tsai et al., 2015)) at Cas9-induced DSBs, thereby tagging these loci for selective amplification and subsequent deep sequencing. The method is quite sensitive as off-target sites with $>$0.1% indel frequency can be detected, and the frequency of dsODN insertion appears to be correlated with the frequency of Cas9-induced lesions at each site (Tsai et al., 2015). This method has been used successfully to evaluate the precision of Cas9 and its variants (tru-sgRNAs (Tsai et al., 2015) or PAM variants (Kleinstiver et al., 2015)). Given its favorable properties, GUIDE-seq could become a standard in the nuclease field for off-target analysis.

While the GUIDE-seq method is straightforward to employ, to date no bioinformatic tools have been released to the community to support the analysis of this data. We developed *GUIDEseq* package to faciliate the analysis of GUIDE-seq dataset, including retaining one read per unique molecular identifier (UMI), filtering reads lacking integration oligo sequence (dsODNs), identifying peak locations (cleavage sites) and heights, merging cleavage sites from plus strand and those from minus strand, and performing target and off target search of the input gRNA. This analysis leverages our *ChIPpeakAnno* package (Zhu et al., 2010) for merging cleavage sites from plus strand and minus strand, and *CRISPRseek* package (Zhu et al., 2014) for defining the homology of any identified off-target site to the guide sequence and Cas9 PAM specificity.

2 Workflow of GUIDE-seq data analysis

Here is the workflow of GUIDE-seq data analysis with human sequence. First load *GUIDEseq* and *BSgenome.Hsapiens.UCSC.hg19*.

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

Then specify the alignment file path as alignment.inputfile, and a umi file path as umi.inputfile containing unique molecularidentifier for each sequence.

```
> library(GUIDEseq)
> library(BSgenome.Hsapiens.UCSC.hg19)
> umi.inputfile <- system.file("extdata", "UMI-HEK293_site4_R1.txt",
+     package = "GUIDEseq")
> alignment.inputfile <- system.file("extdata","bowtie2.HEK293_site4.sort.bed",
+     package = "GUIDEseq")
```

The alignment.inputfile is an alignment file in bed format containing CIGAR information. The align-

ment.inputfile contains chromosome, start, end, readID, mapping quality, strand and CIGAR information as a tab delimited file. Here is an example line.
chr13 27629253 27629403 HWI-M01326:156:1:113:4572:6938/1 44 + 150M
Scripts for bin reads, remove adaptor, mapping to genome are available at http://mccb.umassmed.edu/GUIDE-seq/.

The umi.inputfile is a tab delimited file containing at least two columns, read IDs and corresponding unique molecular identifiers (UMI). Script for creating umi.inputfile is available at http://mccb.umassmed.edu/GUIDE-seq/getUmi.pl.
An example input file is at http://mccb.umassmed.edu/GUIDE-seq/testGetUmi/.
Please make sure to use R1 reads as input to getUmi.pl.

2.1 Step 1: Remove PCR bias and obtain unique cleavage events

PCR amplification often leads to biased representation of the starting sequence population. To track the sequence tags present in the initial sequence library, unique molecular identifiers (UMI) are added to the 5 prime of each sequence in the staring library. The function `getUniqueCleavageEvents` uses the UMI sequence in the umi.inputfile (optionally contains umi plus the first few sequence from R1 reads) to obtain the starting sequence library. It also filters out reads that does not contain the integration oligo sequence, too short or not in the right paired configuration.

For detailed parameter settings for function `getUniqueCleavageEvents`, please type help(getUniqueCleavageEvents).

```
> system.time(uniqueCleavageEvents <- getUniqueCleavageEvents(
+     alignment.inputfile = alignment.inputfile , umi.inputfile = umi.inputfile))

   user  system elapsed
  0.393   0.153 129.517

> uniqueCleavageEvents$cleavage.gr

GRanges object with 1607 ranges and 1 metadata column:
         seqnames              ranges strand   |     total
            <Rle>           <IRanges>  <Rle>   | <numeric>
     [1]    chr13 [27629409, 27629409]      +   |         1
     [2]    chr20 [31349772, 31349772]      +   |         1
     [3]    chr13 [27629409, 27629409]      +   |         1
     [4]    chr13 [27629409, 27629409]      +   |         1
     [5]    chr20 [31349772, 31349772]      +   |         1
     ...      ...                 ...    ... ...       ...
  [1603]    chr20 [31349726, 31349726]      -   |         1
  [1604]    chr20 [31349772, 31349772]      -   |         1
  [1605]    chr13 [27629409, 27629409]      -   |         1
  [1606]    chr20 [31349772, 31349772]      -   |         1
  [1607]    chr20 [31349761, 31349761]      -   |         1
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

2.2 Step 2: Summarize cleavage events

Calling the function getPeaks with the results from getUniqueCleavageEvents outputs summarized cleavage events for each moving window with at least min.reads of cleavage events.

By default, window size is set to 20, step is set to 20, and min.reads is set to 2. For detailed parameter settings using function getPeaks, please type help(getPeaks).

```
> peaks <- getPeaks(uniqueCleavageEvents$cleavage.gr, min.reads = 80)
> peaks.gr <- peaks$peaks
> peaks.gr

GRanges object with 4 ranges and 4 metadata columns:
      seqnames              ranges strand |     count        bg   p.value
         <Rle>           <IRanges>  <Rle> | <numeric> <numeric> <numeric>
  [1]    chr13 [27629412, 27629422]      + |       373     1.492         0
  [2]    chr13 [27629400, 27629410]      - |       264     1.056         0
  [3]    chr20 [31349773, 31349783]      + |       437     1.748         0
  [4]    chr20 [31349762, 31349772]      - |       509     2.132         0
            SNratio
           <numeric>
  [1]             250
  [2]             250
  [3]             250
  [4] 238.74296435272
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths
```

2.3 Step 3: Merge peaks from plus and minus strand

Calling the function mergePlusMinusPeaks with the output from getPeaks to merge peaks from plus strand and minus strand with specific orientation and within certain distance apart.

By default, plus.strand.start.gt.minus.strand.end is set to TRUE and distance.threshold is set to 40, i.e., twice of the window size. For detailed parameter settings using function mergePlusMinusPeaks, please type help(mergePlusMinusPeaks).

```
> mergedPeaks <- mergePlusMinusPeaks(peaks.gr = peaks.gr,
+     output.bedfile = "mergedPeaks.bed")
> mergedPeaks$mergedPeaks.gr

GRanges object with 2 ranges and 2 metadata columns:
                                                  seqnames              ranges strand
                                                     <Rle>           <IRanges>  <Rle>
  chr13+:27629412:27629422:chr13-:27629400:27629410    chr13 [27629400, 27629422]      +
  chr20+:31349773:31349783:chr20-:31349762:31349772    chr20 [31349762, 31349783]      +
                                                  |     count        bg
                                                  | <numeric> <numeric>
  chr13+:27629412:27629422:chr13-:27629400:27629410 |       637     2.548
  chr20+:31349773:31349783:chr20-:31349762:31349772 |       946      3.88
  -------
  seqinfo: 2 sequences from an unspecified genome; no seqlengths

> head(mergedPeaks$mergedPeaks.bed)

  seqnames minStart   maxEnd                                              names totalCount
1    chr13 27629400 27629422 chr13+:27629412:27629422:chr13-:27629400:27629410        637
2    chr20 31349762 31349783 chr20+:31349773:31349783:chr20-:31349762:31349772        946
  strand
1      +
2      +
```

2.4 Step 4: Off target analysis of extended regions around the identified cleavage sites

Calling the function `offTargetAnalysisOfPeakRegions` with input gRNA, peaks and genome of interest, to annotate identified cleavage sites with sequence homolgy to input gRNA. For detailed parameter settings using function `offTargetAnalysisOfPeakRegions`,
please type help(offTargetAnalysisOfPeakRegions)

```
> peaks <- system.file("extdata", "T2plus100OffTargets.bed",
+     package = "CRISPRseek")
> gRNAs <- system.file("extdata", "T2.fa",
+     package = "CRISPRseek")
> outputDir <- getwd()
> offTargets <- offTargetAnalysisOfPeakRegions(gRNA = gRNAs, peaks = peaks,
+     format=c("fasta", "bed"),
+     peaks.withHeader = TRUE, BSgenomeName = Hsapiens,
+     upstream = 50, downstream =50, PAM.size = 3, gRNA.size = 20,
+     PAM = "NGG", PAM.pattern = "(NAG|NGG|NGA)$", max.mismatch = 2,
+     outputDir = outputDir,
+     allowed.mismatch.PAM = 2, overwrite = TRUE
+    )

search for gRNAs for input file1...
Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  /private/tmp/Rtmp6kDGeR/Rbuild153df5826df59/GUIDEseq/vignettes/T2.fa-Oct-28-2015/
[1] "Scoring ..."
finish off-target search in sequence 2
finish off-target search in sequence 1
finish feature vector building
finish score calculation
[1] "Done!"
```

2.5 Run all steps in one workflow function

The function `GUIDEseqAnalysis` is a wrapper function that uses the UMI sequence or plus the first few bases of each sequence from R1 reads to estimate the starting sequence library, piles up reads with a user defined window and step size, identify the cleavage sites, merge cleavage sites from plus strand and minus strand, followed by off target analysis of extended regions around the identified cleavage sites. For detailed parameter settings using function `GUIDEseqAnalysis`,
please type help(GUIDEseqAnalysis)

```
> gRNA.file <- system.file("extdata","gRNA.fa", package = "GUIDEseq")
> system.time(guideSeqRes <- GUIDEseqAnalysis(
+     alignment.inputfile = alignment.inputfile,
+     umi.inputfile = umi.inputfile, gRNA.file = gRNA.file,
+     BSgenomeName = Hsapiens, min.reads = 2))

search for gRNAs for input file1...
Validating input ...
Searching for gRNAs ...
Done. Please check output files in directory  gRNAHEK293_site4min2window20step20distance40/gRNA.fa-Oct-28-2015/
[1] "Scoring ..."
finish off-target search in sequence 2
finish off-target search in sequence 1
finish feature vector building
finish score calculation
[1] "Done!"
```

```
   user  system elapsed
  2.433   0.122 143.177
```

```
> names(guideSeqRes)
```

```
[1] "offTargets"      "merged.peaks"    "peaks"           "uniqueCleavages"
[5] "read.summary"
```

3 References

References

[1] Cox, D.B.T. et al. Therapeutic genome editing: prospects and challenges. Nat Med, 21, 121-131.

[2] Fu, Y. et al. High-frequency off-target mutagenesis induced by CRISPR-Cas nucleases in human cells. Nature biotechnology, 31, 822-826.

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

[4] Kleinstiver, B.P. et al. Engineered CRISPR-Cas9 nucleases with altered PAM specificities. Nature. 2015

[5] Ledford, H. CRISPR, the disruptor. Nature 2015. 522, 20-24.

[6] 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

[7] Tsai,S.Q. et al. GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. Nature biotechnology 2015. 33, 187-197.

[8] Zhu L.J. et al., 2010. ChIPpeakAnno: a Bioconductor package to annotate ChIP-seq and ChIP-chip data. BMC Bioinformatics 2010, 11:237doi:10.1186/1471-2105-11-237.

[9] 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

4 Session Info

```
> sessionInfo()
```

```
R version 3.2.2 (2015-08-14)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

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

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

other attached packages:
 [1] BSgenome.Hsapiens.UCSC.hg19_1.4.0 BSgenome_1.38.0
 [3] rtracklayer_1.30.1                Biostrings_2.38.0
 [5] XVector_0.10.0                    GenomicRanges_1.22.0
 [7] GenomeInfoDb_1.6.0                GUIDEseq_1.0.4
 [9] RSQLite_1.0.0                     DBI_0.3.1
[11] IRanges_2.4.1                     S4Vectors_0.8.0
[13] BiocGenerics_0.16.0               knitr_1.11

loaded via a namespace (and not attached):
 [1] SummarizedExperiment_1.0.0   splines_3.2.2
 [3] htmltools_0.2.6              GenomicFeatures_1.22.0
 [5] chron_2.3-47                 interactiveDisplayBase_1.8.0
 [7] RBGL_1.46.0                  survival_2.38-3
 [9] XML_3.98-1.3                 ensembldb_1.2.0
[11] BiocParallel_1.4.0           lambda.r_1.1.7
[13] matrixStats_0.15.0           stringr_1.0.0
[15] zlibbioc_1.16.0              futile.logger_1.4.1
[17] memoise_0.2.1                Biobase_2.30.0
[19] biomaRt_2.26.0               httpuv_1.3.3
[21] BiocInstaller_1.20.0         ChIPpeakAnno_3.4.1
[23] AnnotationDbi_1.32.0         Rcpp_0.12.1
[25] xtable_1.7-4                 regioneR_1.2.0
[27] limma_3.26.0                 graph_1.48.0
[29] mime_0.4                     Rsamtools_1.22.0
[31] AnnotationHub_2.2.1          BiocStyle_1.8.0
[33] digest_0.6.8                 stringi_1.0-1
[35] shiny_0.12.2                 ade4_1.7-2
[37] grid_3.2.2                   tools_3.2.2
[39] bitops_1.0-6                 magrittr_1.5
[41] RCurl_1.95-4.7               futile.options_1.0.0
[43] seqinr_3.1-3                 GO.db_3.2.2
[45] MASS_7.3-44                  data.table_1.9.6
[47] httr_1.0.0                   R6_2.1.1
[49] VennDiagram_1.6.16           GenomicAlignments_1.6.1
[51] multtest_2.26.0              CRISPRseek_1.10.0
```