Using crisprDesign to design gRNAs with minor and major alleles

Jean-Philippe Fortin, Luke Hoberecht

Introduction

Genetic variants such as single nucleotide polymorphisms (SNPs) can be problematic in guide RNA (gRNA) design, as different alleles can result in unintended gRNA:DNA mismatches for on-targets that reduce gRNA efficacy. To circumvent this, it is advisable to generally avoid targeting sequences that contain variants. However, this may not always be possible, due to a small target window and/or few target options, or desirable, if, for example, a CRISPR application intends to target a pathogenic variant.

Functions in `crisprDesign` are well equipped to handle these cases. gRNAs overlapping SNPs can be identified with the `addSNPAnnotation` function, as documented in the CRISPRko design with Cas9 tutorial. Should the user wish to target region despite (or because of) the presence of variants, the user only needs to take care in the choice of `BSgenome` when constructing the `GuideSet` (an alternative option is to supply a custom target sequence; see the Working with a custom sequence tutorial for more information).

This tutorial covers use cases for `BSgenome` objects that store variants of the reference human genome (hg38) injected with major and minor alleles. It assumes the reader is familiar with constructing and using gene annotation objects (see the Building a gene annotation object tutorial) and `GuideSet` objects (see the CRISPRko design with Cas9 tutorial) so that the content may focus on the utility of the `BSgenome` variants discussed herein. Please consult the applicable tutorials if necessary.

Finally, it goes without saying that the user should be knowledgeable of the sequence(s), including possible variations in such, he or she is designing gRNAs for.

Installation

See the Installation tutorial to learn how to install the crisprVerse packages necessary for this tutorial.

Terminology

See the CRISPRko design tutorial to get familiar with the terminology used throughout this tutorial.

gRNA design

Loading packages

We first load the necessary packages for this tutorial:

```
library(crisprBase)
library(crisprDesign)
library(crisprDesignData)
library(BSgenome.Hsapiens.UCSC.hg38)
library(BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major)
library(BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor)
```

We will also load the gene annotation model `txdb_human` from the `crisprDesignData` package:

```
data(txdb_human, package="crisprDesignData")
```

The `BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major` and `BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor` packages are `BSgenome` packages that contain the major and minor alleles of the human reference genome hg38 based dbSNP151.For more information, type the following:

```
help(BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major)
help(BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor)
```

Designing gRNAs for human (hg38) with injected major alleles

It is worth noting that the human reference genome sequence (GRCh38.p12), does not give the major allele (i.e. most common allele in a population) at all nucleotide locations. Indeed, given that it was historically constructed from a small set of human genomes, it contains minor alleles that were common across this set of human genomes.

For example, in the coding region (CDS) of the human gene SMC3, the reference genome contains the minor allele of the SNP rs2419565; the reference allele (A) frequency is 0.00577, and the alternative allele (G) frequency is 0.99423 as indicated in the table here here).

Designing gRNAs targeting the CDS of SMC3 with the SpCas9 nuclease returns one gRNA that overlaps this SNP. Below, we first construct a `GuideSet` object using the reference `BSgenome` object:

```
smc3 <- queryTxObject(txdb_human,
                      featureType="cds",
                      queryColumn="gene_symbol",
                      queryValue="SMC3")
gs_reference <- findSpacers(smc3,
                            crisprNuclease=SpCas9,
                            bsgenome=BSgenome.Hsapiens.UCSC.hg38)
```

and a `GuideSet` object with the `BSgenome` object that contains the major alleles:

```
gs_major <- findSpacers(smc3,
                        crisprNuclease=SpCas9,
                        bsgenome=BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major)
```

Let's compare the protospacer sequences from both objects:

```
protospacers(gs_reference["spacer_199"])
## DNAStringSet object of length 1:
##     width seq                                                names
## [1]    20 TAGGGGTTACAAATCAATCA                               spacer_199
protospacers(gs_major["spacer_199"])
## DNAStringSet object of length 1:
##     width seq                                                names
## [1]    20 TAGGGGTTACAAATCGATCA                               spacer_199
```

The variant occurs in the seed sequence of this gRNA, 5 bases upstream of the `pam_site`, so a gRNA:DNA mismatch at this location is likely detrimental to its efficacy. Also, as this major allele occurs at >99% frequency, it may be more beneficial to design gRNAs in this example using the major allele genome contained in `BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major`.

Designing gRNAs for human (hg38) with injected minor alleles

It may be desirable, in some applications, to target a genic sequence that contains a minor allele (i.e. less common allele) rather than the major or reference allele. For example, if a particular minor allele is pathogenic and the host cell has a single copy of that allele, the user may want to target that pathogenic variant and disrupt its behavior while leaving the other copy undisturbed.

As an example, using `BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor`, we can target the pathogenic minor allele (rs398122995) located in the human gene ALMS1:

```
alms1 <- queryTxObject(txdb_human, 'cds', 'gene_symbol', 'ALMS1')
gs_minor <- findSpacers(alms1,
                        crisprNuclease=SpCas9,
                        bsgenome=BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor)
gs_minor <- unique(gs_minor)
```

We also include, for comparison, the resulting `GuideSet` using the reference genome sequence:

```
gs_reference <- findSpacers(alms1,
                            crisprNuclease=SpCas9,
                            bsgenome=BSgenome.Hsapiens.UCSC.hg38)
gs_reference <- unique(gs_reference)
```

and compare the two versions of the gRNA:

```
gs_reference["spacer_615"]
```

```
## GuideSet object with 1 range and 5 metadata columns:
##              seqnames    ranges strand |          protospacer            pam
##                 <Rle> <IRanges>  <Rle> |       <DNAStringSet> <DNAStringSet>
##   spacer_615     chr2  73448425      - | TACTCTCTGGTAACTCTTGT            TGG
##              pam_site  cut_site      region
##              <numeric> <numeric> <character>
##   spacer_615  73448425  73448428    region_7
##   -------
##   seqinfo: 640 sequences (1 circular) from hg38 genome
##   crisprNuclease: SpCas9
```

```
gs_minor["spacer_612"]
```

```
## GuideSet object with 1 range and 5 metadata columns:
##              seqnames    ranges strand |          protospacer            pam
##                 <Rle> <IRanges>  <Rle> |       <DNAStringSet> <DNAStringSet>
##   spacer_612     chr2  73448425      - | TACTCTCTGGTAACTCTTGC            TGG
##              pam_site  cut_site      region
##              <numeric> <numeric> <character>
##   spacer_612  73448425  73448428    region_7
##   -------
##   seqinfo: 595 sequences (1 circular) from hg38 genome
##   crisprNuclease: SpCas9
```

The variant occurs 1 base upstream of the `pam_site`, and likely influences gRNA activity, that is, we can design a gRNA that targets the minor allele and has a much lower affinity for the reference, or major allele.

Note that while the two `GuideSet` objects differ only by their `BSgenome` object, we need to provide different indices to access protospacers at equivalent `pam_sites`. This is due to variants in one `BSgenome` (in this case theone with minor alleles) eliminating PAM sequences, that is, one of the Gs in NGG is changed to another base such that SpCas9 does not recognize it. This, where permissible, is also an effective way of ensuring gRNAs only target a specific sequence if that sequence contains the desired variant.

Session Info

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.7
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## 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] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] BSgenome.Hsapiens.UCSC.hg38.dbSNP151.minor_0.0.9999
##  [2] BSgenome.Hsapiens.UCSC.hg38.dbSNP151.major_0.0.9999
##  [3] BSgenome.Mmusculus.UCSC.mm10_1.4.3
##  [4] BSgenome.Hsapiens.UCSC.hg38_1.4.4
##  [5] BSgenome_1.65.2
##  [6] rtracklayer_1.57.0
##  [7] Biostrings_2.65.2
##  [8] XVector_0.37.0
##  [9] GenomicRanges_1.49.1
## [10] GenomeInfoDb_1.33.5
## [11] IRanges_2.31.2
## [12] S4Vectors_0.35.1
## [13] crisprDesignData_0.99.17
## [14] crisprDesign_0.99.133
## [15] crisprScore_1.1.14
## [16] crisprScoreData_1.1.3
## [17] ExperimentHub_2.5.0
## [18] AnnotationHub_3.5.0
## [19] BiocFileCache_2.5.0
## [20] dbplyr_2.2.1
## [21] BiocGenerics_0.43.1
## [22] crisprBowtie_1.1.1
## [23] crisprBase_1.1.5
## [24] crisprVerse_0.99.8
## [25] rmarkdown_2.15.2
##
## loaded via a namespace (and not attached):
##  [1] rjson_0.2.21                 ellipsis_0.3.2
##  [3] Rbowtie_1.37.0               bit64_4.0.5
##  [5] lubridate_1.8.0              interactiveDisplayBase_1.35.0
##  [7] AnnotationDbi_1.59.1         fansi_1.0.3
##  [9] xml2_1.3.3                   codetools_0.2-18
## [11] cachem_1.0.6                 knitr_1.40
## [13] jsonlite_1.8.0               Rsamtools_2.13.4
```

```
## [15] png_0.1-7                     shiny_1.7.2
## [17] BiocManager_1.30.18           readr_2.1.2
## [19] compiler_4.2.1                httr_1.4.4
## [21] basilisk_1.9.2                assertthat_0.2.1
## [23] Matrix_1.4-1                  fastmap_1.1.0
## [25] cli_3.3.0                     later_1.3.0
## [27] htmltools_0.5.3               prettyunits_1.1.1
## [29] tools_4.2.1                   glue_1.6.2
## [31] GenomeInfoDbData_1.2.8        dplyr_1.0.9
## [33] rappdirs_0.3.3                tinytex_0.41
## [35] Rcpp_1.0.9                    Biobase_2.57.1
## [37] vctrs_0.4.1                   crisprBwa_1.1.3
## [39] xfun_0.32                     stringr_1.4.1
## [41] mime_0.12                     lifecycle_1.0.1
## [43] restfulr_0.0.15               XML_3.99-0.10
## [45] zlibbioc_1.43.0               basilisk.utils_1.9.1
## [47] vroom_1.5.7                   VariantAnnotation_1.43.3
## [49] hms_1.1.2                     promises_1.2.0.1
## [51] MatrixGenerics_1.9.1          parallel_4.2.1
## [53] SummarizedExperiment_1.27.1   RMariaDB_1.2.2
## [55] yaml_2.3.5                    curl_4.3.2
## [57] memoise_2.0.1                 reticulate_1.25
## [59] biomaRt_2.53.2                stringi_1.7.8
## [61] RSQLite_2.2.16                BiocVersion_3.16.0
## [63] highr_0.9                     BiocIO_1.7.1
## [65] randomForest_4.7-1.1          GenomicFeatures_1.49.6
## [67] filelock_1.0.2                BiocParallel_1.31.12
## [69] rlang_1.0.4                   pkgconfig_2.0.3
## [71] matrixStats_0.62.0            bitops_1.0-7
## [73] evaluate_0.16                 lattice_0.20-45
## [75] purrr_0.3.4                   GenomicAlignments_1.33.1
## [77] bit_4.0.4                     tidyselect_1.1.2
## [79] magrittr_2.0.3                R6_2.5.1
## [81] generics_0.1.3                DelayedArray_0.23.1
## [83] DBI_1.1.3                     pillar_1.8.1
## [85] KEGGREST_1.37.3               RCurl_1.98-1.8
## [87] tibble_3.1.8                  dir.expiry_1.5.0
## [89] crayon_1.5.1                  utf8_1.2.2
## [91] tzdb_0.3.0                    progress_1.2.2
## [93] grid_4.2.1                    blob_1.2.3
## [95] digest_0.6.29                 xtable_1.8-4
## [97] httpuv_1.6.5                  Rbwa_1.1.0
```