Draft Genomes Information & Protocol

Pre-Processing with Provided Script

Due to the variable nature of draft genome annotations, these files must be pre-processed before input into the guide finder program. The annotations can be pre-processed with custom scripts, or with the pre-processing script provided.

1. The supplied script requires the following inputs:

A. **Fasta file.** You may input a single-sequence fasta file directly into the script or you may input a multi-sequence fasta file. If inputting the latter, the program will concatenate all of the sequences in the file (with a string of N’s in between) to create a single sequence fasta file. A single-sequence fasta file is required for the rest of the pre-processing script and for the main guide finder script. May be a .fasta. .fna or .txt file.

B. **Coding Sequences file.** This file should contain the coding sequences in the 5’ 🡪 3’ direction for each of the genes you want to find guides for. For example, if you intend to run the guide finder script genome-wide, input a file containing all of the coding sequences for each gene. However, if you intend to only identify guides for a subset of genes, only include the coding sequences of these genes in this file. May be a .fasta, .fna or .txt file.

2. The pre-processing script performs the following functions:

A. **Concatenates a multi-sequence fasta file into a single sequence**: if a multi-sequence fasta file is input into the program, the script will concatenate all sequences and add a series of N’s between contigs. The program will output the file to the working directory and then the file will be read back into the program for downstream use. If a single-sequence fasta file is already available, the first concatenation step is not necessary and the user can directly read in the file.

B. **Parses gene identification information from coding sequence file:** the coding sequence file is parsed to identify the gene names/unique identifiers. Due to the variable nature of genome annotations and the functions used to parse the gene names/IDs, this section may not be stable. The functions parsing the gene names/IDs from the headers of the coding sequence file may need adjustment. Testing has been completed on several files downloaded from PATRIC and was run successfully for each of these genomes, but please be aware that this issue exists and files downloaded from this and other databases may behave differently.

B. **Identifies gene coordinates:** aligning the coding sequence of each gene to the single-sequence fasta file using BLAST, the script identifies the coordinates of each gene.

C. **Modifies gene coordinates**: coordinates identified from the BLAST output are adjusted to the format required for the guide finder program (e.g. the smaller coordinate is listed as the “start” coordinate, regardless of whether the gene is coded on the plus or minus strand).

D. **Outputs gene coordinates:** a file containing the gene coordinates and strand type is output into the working directory. This file can then be input into the main guide finder script.

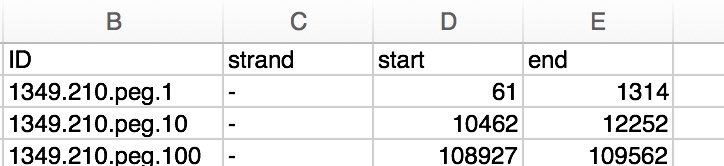
Custom Pre-processing

If you have custom scripts for pre-processing or do not wish to use the provided pre-processing script, you can still use the guide finder program. Since it is crucial that the information input into the guide finder program is in the proper format required by the program, please use the following as a guideline for formatting data prior to use.

**1. FASTA File: your fasta file—containing the nucleotide sequence of the entire genome—should be a single sequence.**

If you only have a multi-sequence fasta file available (e.g. multiple contigs), these sequences must be concatenated with a series of 100 or more N’s between each sequence prior to input into the guide finder program. The program requires this format for draft genomes because it is set to search for PAM sites in a user-defined (0-100 bp) upstream of the start coordinate for each gene to identify guides in this putative promoter region. Adding a series of 100 or more N’s between each contig prevents the program from mistakenly identifying guides. For example, consider the situation where a gene annotation does not have 100 bp of upstream sequence available (e.g. the first gene annotated on a contig). In this case, if the multi-sequence file were simply concatenated without Ns between, the program would use the final few bp of the sequence on the prior contig as the putative promoter region for this gene, despite the fact that this does not represent the true upstream region of the gene. By adding a series of N’s between the contigs, the program avoids this issue (guides will be made from this region of Ns but then subsequently filtered out by the program) and will effectively only produce guides targeting the gene body for this gene. If your fasta file is composed of just a few sequences, it may be easiest to complete the concatenation and addition of N’s manually (simply delete headers—leaving the first header—and add a sequence of 100 or more Ns between the sequences). For multi-sequence fasta files with a large number of sequences, concatenation and addition of Ns may be completed by the provided pre-processing script or through a custom script.

**2. Coordinates File: in addition to the fasta file, you must also provide a file that contains for each gene: the unique gene identifier, start and end coordinates, and whether the gene is coded on the plus or minus strand.**

An example of an appropriately formatted draft genome input file is included. The file should be saved as .csv (can be created in Excel as saved as .csv) where each row corresponds to a unique coding sequence and each column describes elements of that sequences. There should be a column for the gene name/ID, a column for the start coordinate, a column for the end coordinate, and a column describing whether the gene is coded on the plus or minus strand (+/-)

2.1. Gene Identifier: input file must include a unique identifier for each coding sequence. This will be present in the gene annotation as some sort of alphanumeric identifier and can often be parsed from the coding sequences file (as the supplied pre-processing script does). Avoid using the gene product (e.g. translation initiation factor, hypothetical protein etc) as unique identifiers as these are not unique to each gene.

2.2 Strand Type: include – or + to indicate whether each gene is coded on the plus or minus strand.This information is important to designing guides correctly and a column containing this information must be included in the input file. MUST be in the format of + or – (i.e. not “Plus”, “Minus”)

2.2 Gene Coordinates: your gene coordinates should be identified by BLAST alignment and formatted so that the smaller coordinate is designated as the “start” coordinate.Experience indicates that the gene coordinates for draft genome annotations do not always exactly match the actual location of the coding sequence in the fasta file. Additionally, if the fasta file was originally provided as a multi-fasta file and then concatenated and a series of Ns added between sequence chunks, the coding sequence coordinates will not align with the new fasta file. For this reason, we highly recommend that BLAST is used to identify the gene coordinates (align coding sequences provided by genome annotation to the single-sequence fasta file and then subset the top hit for each query (each coding sequence)). Parse the output file to identify gene coordinates. Modify the coordinates such that the smaller coordinate is designated as the “start” coordinate (regardless of whether the gene is coded on the plus or minus strand).

Detailed Use Instructions for the Pre-Processing & Guide Finder Scripts for Draft Genomes

1. Download FASTA file and coding sequence file for your genome of interest.   
The fasta file may be in a multi-sequence format (i.e multiple contigs, separated by headers within the same file). The coding sequence file should contain the coding sequences in the 5’🡪3’ direction for each gene

2. Identify genes to target with guides

If you want to design guides to target each gene in the genome, do not edit the coding sequences file. If you wish only to design guides for a subset of genes, create a file that contains the coding sequences (+ headers) for only the genes you want to target. Use this file as your coding sequences file that will be input into the pre-processing script.

3. Download blastn and locate file path to makeblastdb and blastn

The file path is the location of each of these specific files, which should be contained within the BLAST folder. If you do not know how to identify the location of these files, use the supplied script “FindPaths-Mac”

4. Within the pre-processing script in RStudio, do the following:   
*Line-by-line instructions within the file and the Draft Genomes Example document*

A. Set working directory (where you FASTA file is saved and where files will output to)

B. Run the next three chunks of code (“Install Packages”, “Load Packages”, and “Functions Created”).

C. Input the multi-sequence fasta file and run the chunk of code to make into a single-sequence file. Input the multi-sequence fasta file by putting the name of the file in parentheses in the read.fasta function; file name should be in quotation. The code should output a file called concatenatedfastafile.txt that you should see saved in your working directory.

D. Input your concatenated sequence file (made in the previous step) and the coding sequence file and set the path to BLAST. Change to the appropriate file names and run this chunk of code.

E. Run the rest of the code chunks! Run each of the code chunks by pressing the green arrow. When the program finishes, you should see two new files in your working directory: 1) concatenatedfastafile.txt (the single-sequence fasta file made from a multi-sequence input) and 2) DraftGenome\_NewCoordinates.csv (the file with the parsed coordinate and gene ID information that should be input into the Guide Finder script)

5. Within the Guide Finder script in RStudio, do the following:

*Line-by-line instructions within the file*

A. Set working directory (where you FASTA file is saved and where files will output to) in “Set Working Directory”

B. Run the next three chunks of code (“Install Packages”, “Load Packages”, and “Functions Created”).

C. Set the path to makeblastdb for the MakeDatabase command in “Set BLAST Path & File”

D. After setting the path for makeblastdb, substitute the fasta file name for the name of your fasta file in the “Set BLAST Path & File” chunk

E. Set the path to blasn for the RunBlast command in “Set BLAST Path & File”

F. Input the fasta file in the “Input FASTA File” chunk. Input your fasta file by putting the name of your single-sequence fasta file (concatenatedfastafile.txt) in the readDNAstringSet parentheses; file name should be in quotation.

G. Define the putative promoter region by setting PromoterRegion to a number between 0-200. Set the desired putative promoter region, for example, setting PromoterRegion to 50 if you want the Guide Finder to search 50 bp upstream of the transcription start site for each gene to identify guides. If you do not want Guide Finder to make guides from the promoter region, set to 0. Do not set above 100 bp.

H. Set CompleteGenome option to FALSE in “Complete vs Draft Genomes”

I. Skip the chunk of code that says “FOR COMPLETE GENOMES ONLY:

J. Within, “FOR DRAFT GENOMES ONLY”, make sure that the file name within the fread function is the same file name as the output of the pre-processing script (i.e. DraftGenome\_NewCoordinates). If not, change the file name to match.

K. Set parameters for guide design, including GC minimum, maximum distance from 5’ end, guides per gene, and distance between guides (for paired guide design) in “Set Parameters for Guide Design”

L. Skip the chunk of code that says “Manual Guide Selection.” Manual selection for just a subset of genes should occur as part of pre-processing for draft genomes.

N. OPTIONAL: set lowered threshold parameters in “Enter Lowered Thresholds Below”. Genes that did not produce guides with primary thresholds will be re-run through the program with these lowered thresholds. Also, set path for blastn for Re\_RunBlast and Re\_RunBlast\_Lowered in “Set the Path”

O. Run the rest of the code chunks!