Guide Finder Complete Genome Example Exercise:

Prior to beginning the example exercise, read the instructions listed in the guide finder protocol for complete genomes. Download the R and RStudio.

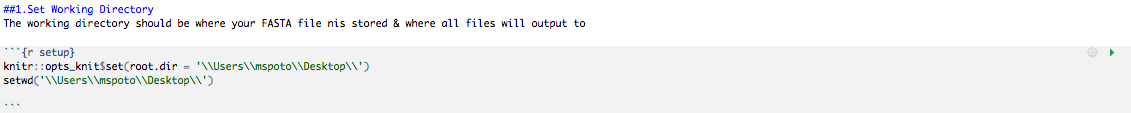
1. Identify required files and information:

* Fasta file (single-sequence)
* GenBank Accession #

In the “Complete Genome Example” folder, you will find a fasta file containing the nucleotide sequence of the complete genome for a strain of *Streptococcus mutans*. This file was downloaded from GenBank. The accession number for the genome was also located on the NCBI website, as shown below (red arrow).



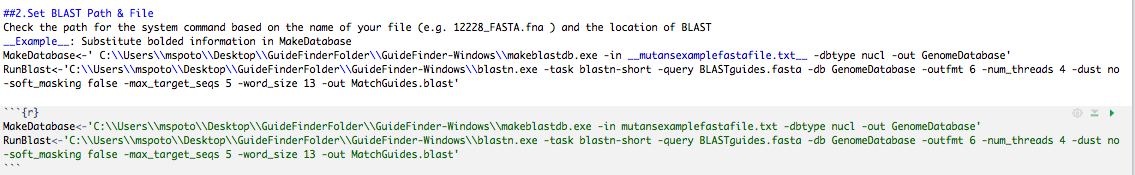
2. Open the guide finder script with the RStudio program and set the working directory.   
The working directory is where all of the files you input into this program should be kept and where all files output by this program will save to. For ease, you can set your working directory to your Desktop, similar to the example shown below, if desired. Set the working directory by identifying the file path to this location. Press the green arrow to run this chunk of code



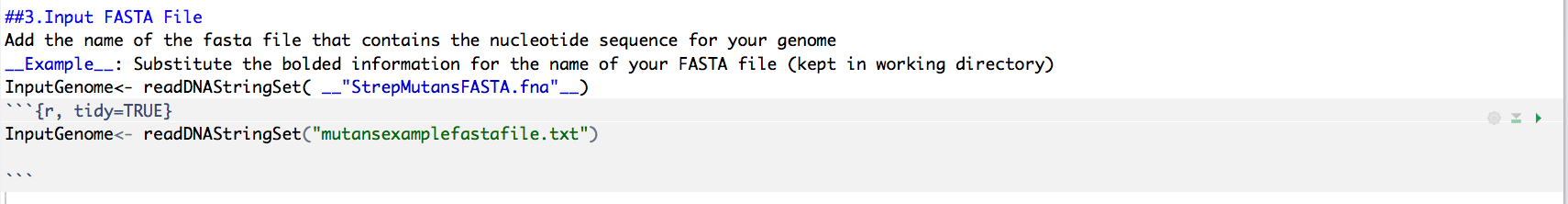
3. Run the next three chunks of code (“Install Packages”, “Loading Packages”, and “Lists Functions”). You can do this by pressing the green arrow at the top of each chunk. You do not need to make any edits to any of these sections of code. Note that you only need to install the packages once and you only need to load the packages each time you run the guide finder script.

4. Set the BLAST file path and the name of the fasta file in the “Set BLAST Path & File” chunk   
**A**. As shown in the red boxes, set the file path for the location of the makeblastdb file and the blastn file.

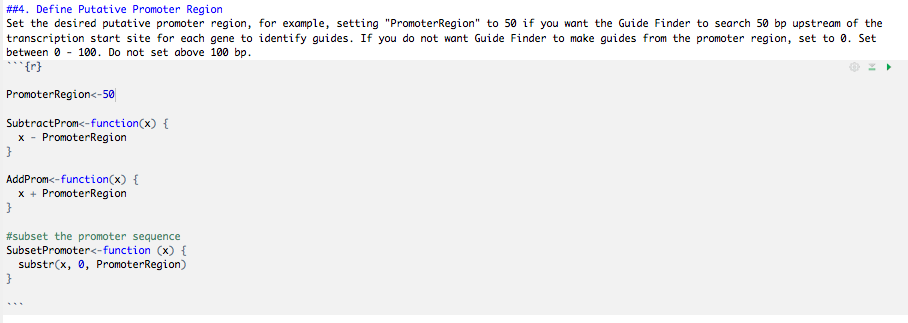
**B.** Set the name of the fasta file here, as shown in the blue box. If not already, change to mutansexamplefastafile.txt for this example. Make sure this file is held in the working directory.



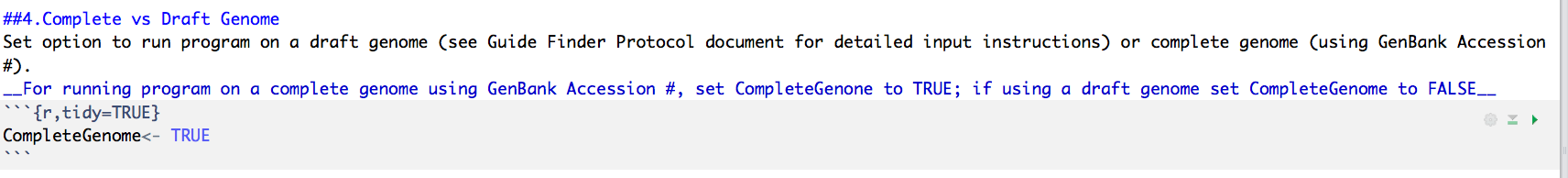
5. Input the FASTA file.   
Do this by including the name of your fasta file in the quotation marks within the readDNAStringSet function. This should be the same file as the last step. Make sure this file is held in the working directory. Run this chunk of code.



6. Define putative promoter region.   
The number you input here will determine how far upstream of a gene’s start site the Guide Finder program will look to identify guides in the promoter region. This can be set anywhere between 0-100. For this exercise, set PromoterRegion to 50 (as shown in the red box).

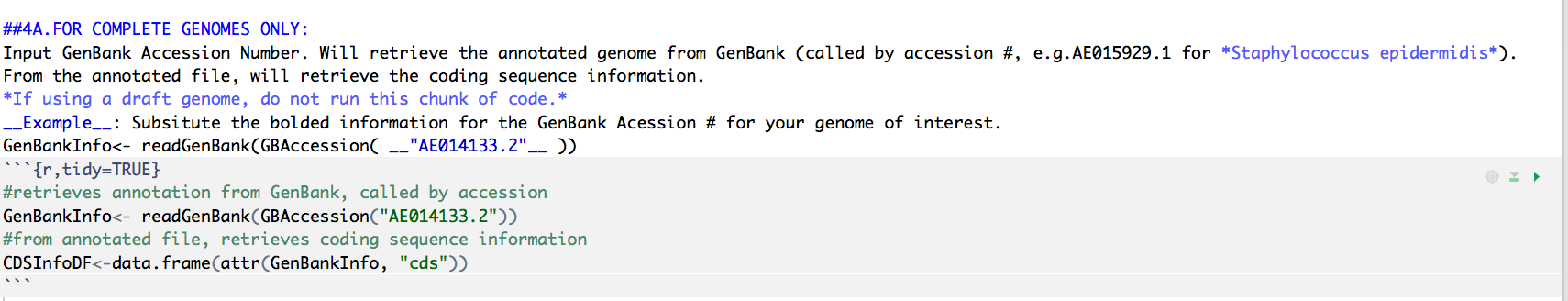


7.Set CompleteGenome to TRUE.  
In this next chunk of code, set CompleteGenome to TRUE and run this chunk.



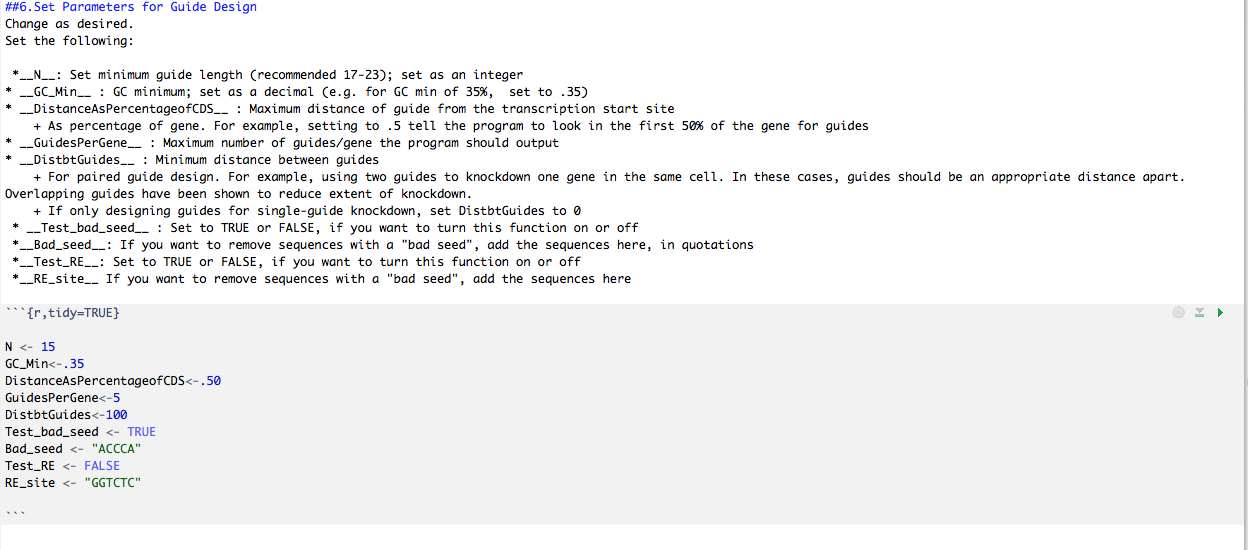
8. Set the GenBank Accession Number and run this chunk of code.

The Streptococcus mutans accession number should be set as default, but if not, add the number to the function readGenBank, in parentheses as shown below. S. mutans number is: AE014133.2. Press the green arrow to run this chunk of code

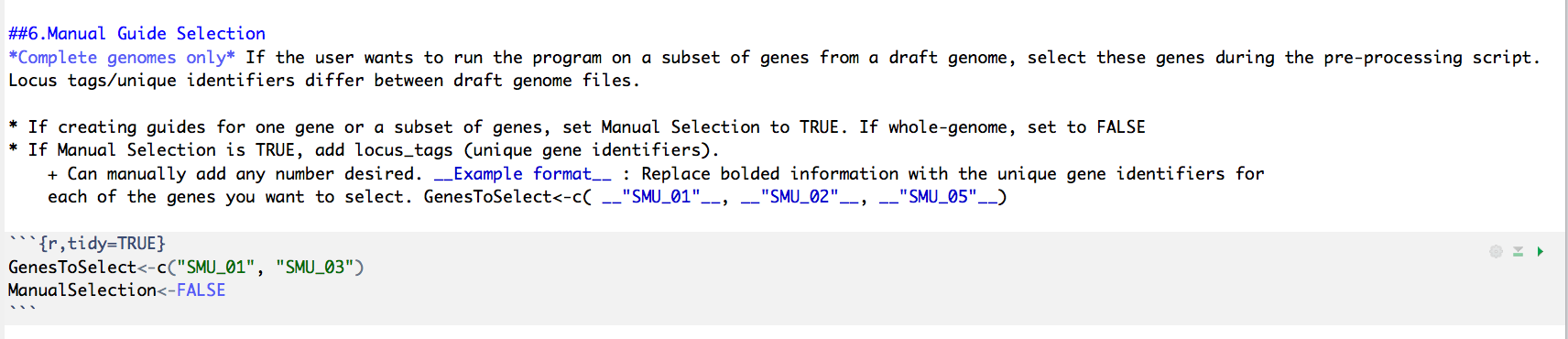


8. Skip the next chunk of code (labeled FOR DRAFT GENOMES ONLY).   
Do not run this piece.

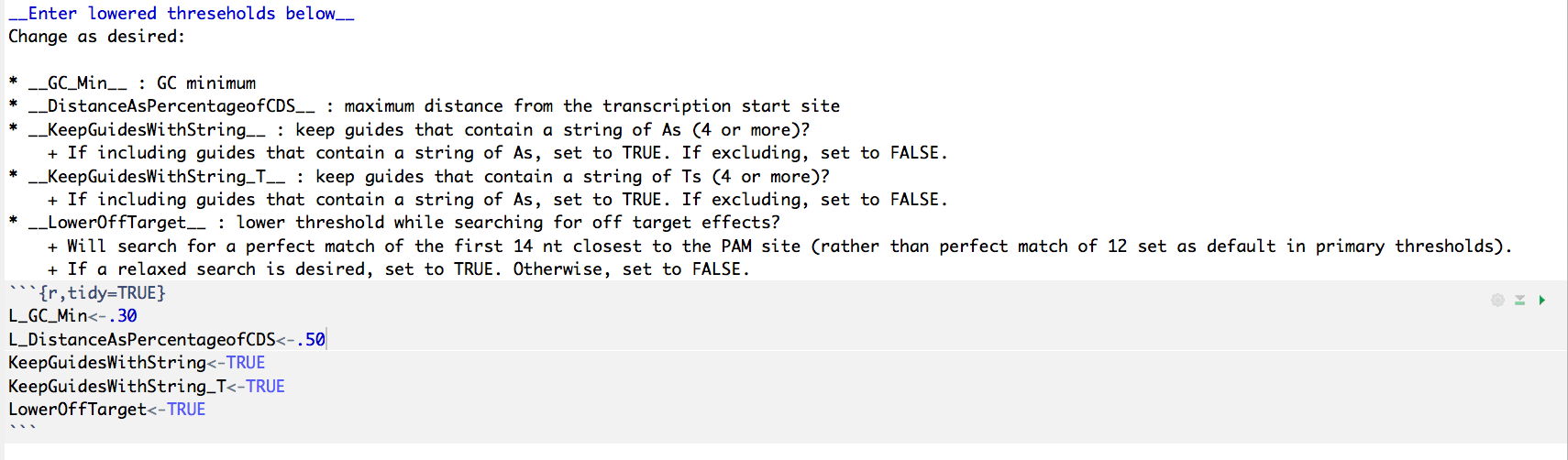
9. Set parameters for guide design   
For this *S. mutans* example, we will set the following parameters for guide design. These are based on rational design parameters that have been successfully used to design guides in our lab. The meaning of each parameter is outlined in the R script, as shown below. Press the green arrow to run this chunk of code



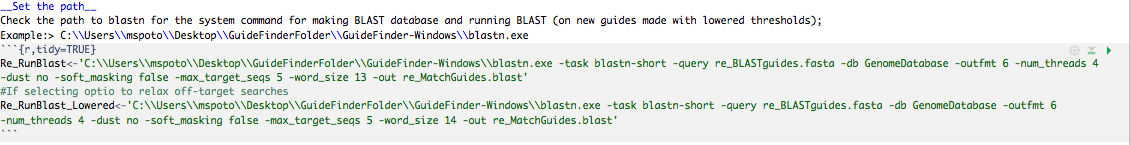
10. Set ManualGuideSelection to FALSE   
This chunk of code allows a user to only use the guide finder program to design guides for a select number of genes (if using a complete genome) The genes are subset using unique locus identification numbers. For this *S. mutans* example, we want to run the program on the entire genome so we set ManualSelection to FALSE. There is no need to change or delete the gene identifier numbers in GenesToSelect if ManualSelection is set to FALSE. Press the green arrow to run this chunk of code



11. Set parameters for relaxed thresholds   
The guide finder program has the ability to identify genes that did not produce usable guides under the primary parameters (set in step 9) and re-run these genes through the program again with relaxed thresholds in an attempt to recover guides. This allows more guides to be identified without compromising quality for the initial round of guide creation. In this *S. mutans* example, set the relaxed parameters as show below. Descriptions of relaxed parameters are described in the script, as shown below. Press the green arrow to run this chunk of code



12. Set the path for BLAST   
Set the path to BLAST again in this chunk. Like Step 11, this step is optional and only needs to be run if the user wants to find more guides for genes that did not produce any. In this *S. mutans* example, we’ll run this chunk of code because we want to eventually perform this iterative step so we need to set the parameters for relaxed guide design (above, step 11) and set parameters for relaxed off-target searching (step 12). *Set this so the same file path* ***for blastn*** *as you did in Step 4.* Do this for ReRunBlast and ReRunBlast\_Lowered. In this step, there is no need to set the path for makeblastdb anywhere. No need to make any more modifications to the code except to set the file path (seen in red, below). Press the green arrow to run this chunk of code



13. Run the rest of the chunks of code!   
Run the remaining chunks of code. There is no need to edit anywhere below the **DO NOT EDIT BELOW THIS LINE** message. Simply press the green arrow on each chunk to run it. The code is commented so that you may follow along.

14. Look at your output files.   
The program will output files (saved to your working directory):   
1) **CompleteGuidesList:** a list of all possible guides, unfiltered, for reference  
2) **CompleteFilteredGuides:** list of all possible guides, filtered  
3) **TopHits\_Guides:** a list of guides—the desired max number/gene set by the user—closest to the transcription start site  
4) **Pairwise\_Guides:** a list of all possible guide pairs for each gene, for dual gene targeting  
5) **GenesWithoutGuides:** a list of genes that did not produce guides with primary thresholds  
6) **GuidesUsingLoweredThresholds:** a list of top hits guides for genes re-run with relaxed parameters  
7) **OverallGeneInfo:** information about the number of genes that created guides (top hits versus paired guides) from primary thresholds and the number of genes that created guides with the lowered thresholds.

Make sure that these files are the same as those provided in the Complete Genome Example folder to see if the program was run correctly.