CASPER (CRISPER Associated Software for Pathway Engineering and Research) v. 1.0

User Manual

CASPER v. 1.0 is a collection of executable and python files that can be run from a text file input and a variety of text file outputs will be generated. This documentation will provide a step-by-step procedure for generating organism target files and doing on- and off- target analysis as well as multitargeting and population analysis on these files. CASPER v. 1.0 requires the user have Python Version 3.5 or later installed to run the .py files.

**CASPER\_Seq\_Finder Executable**

The CASPER\_Seq\_Finder Executable allows the user to select a FASTA file of a genome to be interrogated for potential target sequences. The input requires a set of arguments in the following order:

1. Argument 1 is always the executable file name.
2. Name of endonuclease (e.g. spCas9). This is for the name of the output file only
3. PAM sequence of the endonuclease
4. Any non-canonical PAMs. If there is no non-canonical PAM, “None” is used
5. Organism code name from the KEGG database (e.g. sce). This is for the name of the output file only. It will not make any impact if you use another name unless you use this file and the KEGG gene finding function of CASPERQuick.py
6. “TRUE” or “FALSE”. “FALSE” means the PAM is 3’, “TRUE” means the PAM is 5’ of the target sequence
7. Directory where the output file is to be placed.
8. The list of all FASTA file locations. For metagenomic studies where there is more than 1 file, the output will interpret each new file as a new genome.

*Output File*

Output files are .txt files and are named according to the input into CASPER\_Seq\_Finder executable arguments 2 and 5 (see above). For example an output file for *Saccharomyces cerevisiae* with the spCas9 endonuclease would be named: “scespCas9.txt”

The output file is composed of base-64 representations of the target sequences and their locations. It is broken down by the chromosomes on which the sequences appear. The final section of the output file contains the base-64 representation of the repeated sequences, broken down by the seed sequence, followed by the remaining “tail” sequence and the location at which they appear. This allows the user to sort repeated sequences by a common 16 PAM-proximal bases, and then further interrogate for identical matches with the tail sequences.

The output files can be quickly decompressed to sequences and locations with the SeqTranslate class in the Algorithms.py file.

**Finding Target Sequences with CASPERQuick.py**

CASPERQuick can be run by using a setup text file (“Cquicksetup.txt”). It will search for and decompress the information of a CASPER\_Seq\_Finder file. This will output a file with the sequences and their locations in the regions.

**Performing Off-target analysis**

Off-target analysis is performed on a sequence-by-sequence basis. You have the option to perform off-target analysis on all of the sequences found within the region(s) of interest with CASPERQuick, but this is not recommended as off-target sequence analysis takes a significant amount of time to perform. Off-target analysis is run by the file OffTarget.py and requires only the file listing the target sequences (CASPERofflist.txt) and the CASPER\_Seq\_Finder output file of interest. It will output into a text file with the name given in the CASPERofflist.txt file and will contain all relevant off-target sequences relating to each target sequence that was input. To perform off-target analyses on more than one organism simply add the CASPER\_Seq\_Finder output file of that organism to the list of organisms to compare against. See the example file in the GitHub repository for how to set up the CASPERofflist.txt file.

**Performing *Multitargeting* with multitargeting.py**

Multitargeting is the concept of identifying degenerate sequences across a genome or genomes. A readable output of this process into the command line is generated that can then be ported into Excel or other spreadsheet editing software. Simply change the file name inside the multitargeting.py file to your desired CASPER\_Seq\_Finder output file to perform the analysis.

**Performing population comparisons with Comparison.py**

Identifying sequences across populations requires the user to use the Comparison.py file. To perform comparisons of targets across multiple genomes, simply type the list of CASPER\_Seq\_Finder files that you want to compare into the list of files object at the top of the Comparison.py file. You can also choose the directory you would like to place the created file in by typing in the directory to the directory object. The output file will be named “compare\_” with the names of the organisms followed by the name of the endonuclease. If you select more than 5 organisms, the file name will just be given a name of a random number in place of the organisms.

The output file contains the number of unique sequences shared between pairs of organisms and the number of repeated sequences that appear between organisms. Groups of 3, 4, and up to the complete set of all organisms are also provided.