Disclaimer: This pipeline will **not work on Windows devices** due to some of the external software utilized by CHOPCHOP **only being compatible with MacOS and Linux** operating systems.

SETUP

Step 1: **Download Python 2.7**

- Navigate to this website: https://www.python.org/downloads/release/python-2716/
- Download the version appropriate for your system.

Step 2: **Download Conda**

- Navigate to this website: https://docs.anaconda.com/miniconda/miniconda-other-installer-links/
- Download the version appropriate for your system.
    - (you'll notice the system for the conda downloads is not 2.7, but this will not affect the functionality of the pipeline)

Step 3: **Create your virtual environment and install dependencies**

- Navigate to Terminal.
- If on MacOS paste in this command:

```
CONDA_SUBDIR=osx-64 conda create -n chopchop python=2.7
conda activate chopchop
conda config --env --set subdir osx-64
```

- If on Linux, paste in this command:

```
conda create -n chopchop python=2.7
conda activate chopchop
```

- After your virtual environment has been created and activated, paste in this command to install your needed dependencies:

```
conda install -c conda-forge biopython pandas numpy scipy mysql-python
scikit-learn=0.18.1 tqdm=4.49.0 requests
```

Step 4: **Download TwoBitToFa**

- Navigate to this website: http://hgdownload.soe.ucsc.edu/admin/exe/
    - Click on the correct operating system for your device. (Check your chip architecture, for example: Intel chip → x86_64 and M1 → arm64 for Macs.

    - You will be redirected to a new page, scroll down, locate TwoBitToFa, and download it.
    - Copy the pathname of the file downloaded, and run this command in terminal:

      chmod +x [TwoBitToFa pathname]

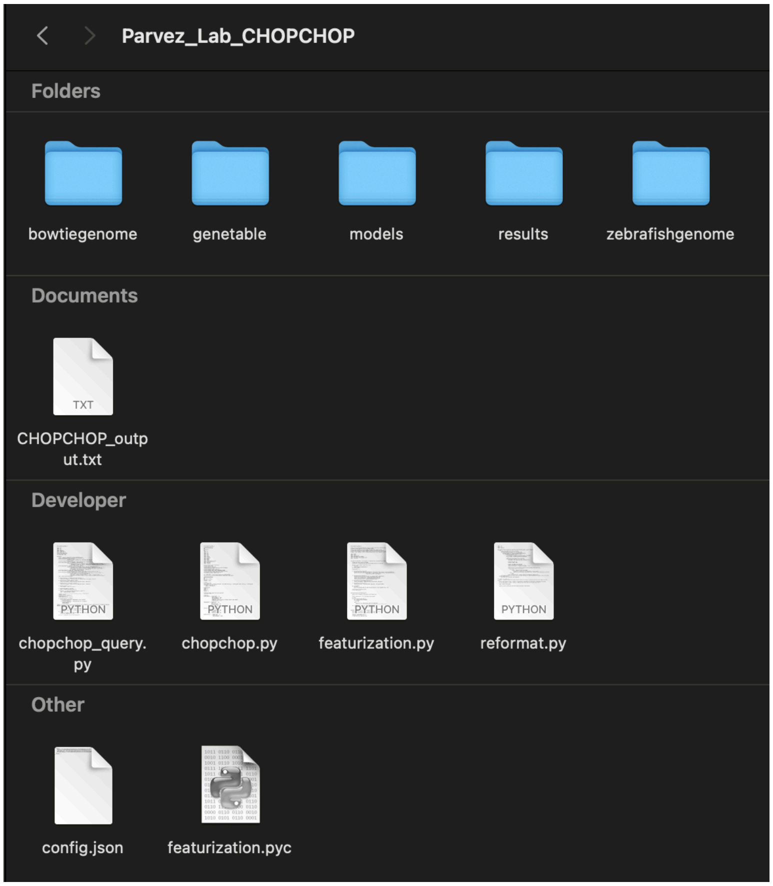
    - After this, run the pathname by itself in terminal once more and see if you encounter a security error. If so, you will have to go into file manager and manually open it so that your computer will trust the software.

Step 5: **Download Bowtie**

- Navigate to this website: https://sourceforge.net/projects/bowtie-bio/files/bowtie/1.3.1/
    - Click on the correct operating system for your device.
    - The software will download as zip file, which you may have to unzip within your file manager.
    - Navigate your way into the bowtie folder, and double click on the file named bowtie.
        - You may encounter a security error. If so, you will have to go into file manager and manually open it so that your computer will trust the software.
            - It is likely that you will need to do this for bowtie-align as well.

Step 6: **Download CHOPCHOP**

- Download the Parvez_Lab_CHOPCHOP zip file from OneDrive and unzip it on your device. This folder contains all the databases and scripts that you will use.

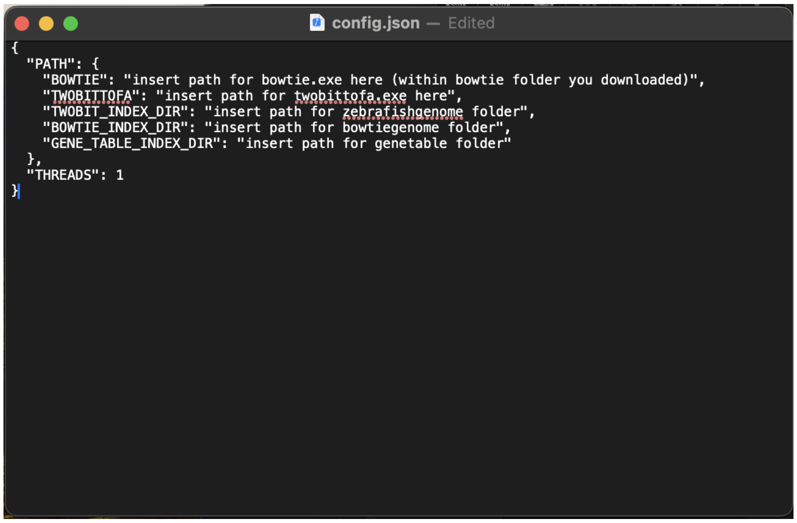


- The folder you download will look like this:
    - **bowtiegenome:** folder containing the referenced index genome for zebrafish
    - **genetable:** folder containing the table of chromosomal coordinates for every characterized gene in zebrafish.
    - **models:** folder containing the existing ML models CHOPCHOP utilizes.
    - **results:** empty folder, functions as temporary storage space for tsv output files.
    - **zebrafishgenome:** folder containing full genome of zebrafish.
    - **CHOPCHOP_output.txt:** this is where your final results will appear.
    - **chopchop_query.py:** this is the main script that will be used to determine potential gRNAs.
    - **chopchop.py:** source code for entire CHOPCHOP analysis, you won't need to interact with this.
    - **featurization.py/pyc:** source code for ML component of CHOPCHOP, you also wont need to interact with this.
    - **reformat.py:** secondary script to analyze output tsv files and produce top 5 gRNAs for each gene you are looking at.

    - **config.json:** conifig file for all your scripts and softwares, this is where you will paste all the different paths of your softwares and databases.

Step 7: **Configure File Paths**

- Navigate to config.json, and open it up as a text file. It should look like this:



```
{
  "PATH": {
    "BOWTIE": "insert path for bowtie.exe here (within bowtie folder you downloaded)",
    "TWOBITTOFA": "insert path for twobittofa.exe here",
    "TWOBIT_INDEX_DIR": "insert path for zebrafishgenome folder",
    "BOWTIE_INDEX_DIR": "insert path for bowtiegenome folder",
    "GENE_TABLE_INDEX_DIR": "insert path for genetable folder"
  },
  "THREADS": 1
}
```

- As you can see above, you should copy and paste the pathnames for all the associated folders and files into their respective classes. This config file will serve as the map for your main script to utilize when it runs an analysis.

USAGE

Step 1: **Prepare your list of genes/coordinates**

- If you are analyzing only one gene, you can skip this step.
- If you have several genes you want to analyze, you may want to format them correctly in advance:
    - The parameters of CHOPCHOP require your gene list to be in an unspaced comma-separated list.
        - Ex. rx3,tbxta,tbx16
        - Ex. chr16:35536483,chr2:29976984,chr5:67430785
    - **Tip:** If you have your genes in a line-by-line list or a column in excel, you can easily format your genes into a list by copy and pasting all of them at once into ChatGPT and asking it to return all your genes as an unspaced comma-separated list.

Step 2: **Run chopchop_query.py**

- Navigate to terminal, and paste in this command:

  conda activate chopchop

- This command will activate the virtual environment you had previously created.
- Take note of the pathname of your chopchop_query.py file and your results folder.
- Paste in this command to terminal, and run it. If the command executes successfully, a progress bar will appear.

  [full pathname for your chopchop_query.py file]* -o *[full pathname for your results folder]* -t CODING > geneerror.txt --gene_names *[comma-separated list of genes]

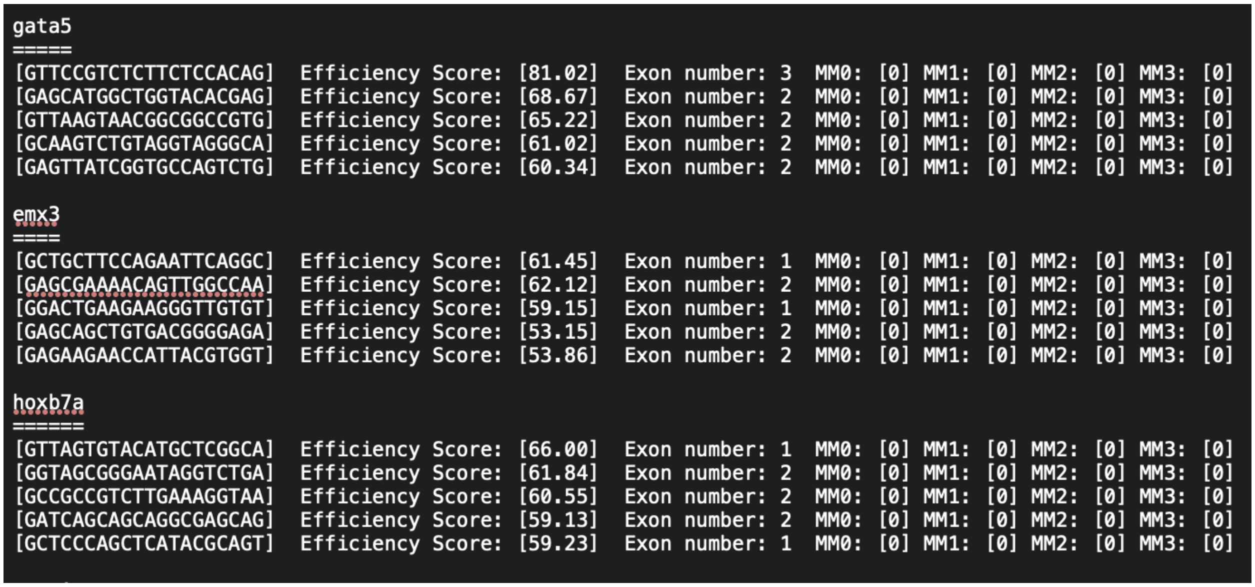
- Once the command finishes executing, you will notice that your results file will have been populated with a .tsv file for every gene you analyzed.
- **Do not close your terminal yet.**

Step 3: **Run reformat.py**

- In the same terminal, enter this command:

  python *[full pathname for your reformat.py file]* -input *[full pathname to your results folder]* -output *[full pathname to your CHOPCHOP_output.txt file*]

- After this script executes, navigate your way to your CHOPCHOP_output.txt file, and you will be met with output that looks like this.



```
gata5
=====
[GTTCCGTCTCTTCTCCACAG]  Efficiency Score: [81.02]  Exon number: 3  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GAGCATGGCTGGTACACGAG]  Efficiency Score: [68.67]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GTTAAGTAACGGCGGCCGTG]  Efficiency Score: [65.22]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GCAAGTCTGTAGGTAGGGCA]  Efficiency Score: [61.02]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GAGTTATCGGTGCCAGTCTG]  Efficiency Score: [60.34]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]

emx3
====
[GCTGCTTCCAGAATTCAGGC]  Efficiency Score: [61.45]  Exon number: 1  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GAGCGAAAACAGTTGGCCAA]  Efficiency Score: [62.12]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GGACTGAAGAAGGGTTGTGT]  Efficiency Score: [59.15]  Exon number: 1  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GAGCAGCTGTGACGGGGAGA]  Efficiency Score: [53.15]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GAGAAGAACCATTACGTGGT]  Efficiency Score: [53.86]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]

hoxb7a
======
[GTTAGTGTACATGCTCGGCA]  Efficiency Score: [66.00]  Exon number: 1  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GGTAGCGGGAATAGGTCTGA]  Efficiency Score: [61.84]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GCCGCCGTCTTGAAAGGTAA]  Efficiency Score: [60.55]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GATCAGCAGCAGGCGAGCAG]  Efficiency Score: [59.13]  Exon number: 2  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
[GCTCCCAGCTCATACGCAGT]  Efficiency Score: [59.23]  Exon number: 1  MM0: [0] MM1: [0] MM2: [0] MM3: [0]
```

- Each gene will have up to 5 optimal gRNAs listed, each with their efficiency, exon number, and mismatches listed out.
- If you analyzed only one gene, you will only receive one gRNA and its corresponding features.

If you would ever like to run a different set of gene(s), make sure to clear your results folder of all .tsv files that were generated from your previous run.