Appendix A

Installation

The following outlines system requirements and build instructions. Note: Before installing, make sure you can see hidden files and folders on your system and that you have admin privileges. You will need to access and modify your ".bashrc" file and perform other installation actions which require admin access. your system will need:

1. Nvidia graphics card (to fully utilize PyTorch)
2. Python 2.7 to run the Rosetta version
3. Python 3.8.10 should be your system default. I.e. When you type "python" in the terminal, python 3.8.10 should load up. Otherwise, you should run the code from Visual Studios Code editor with python 3.8.10 as your interpreter.
4. Packages for python3:
   (a) Numpy
   (b) Pandas
   (c) tqdm
   (d) Pytorch - no instructions for this yet
5. Cmake version 2.8.8 or greater (`https://cmake.org/download/`) and Ninja build (`https://github.com/martine/ninja.git`) to compile Rosetta.
6. Rosetta version 3.12
   (a) Go to the Rosetta Commons website and follow the instructions to obtain an academic licence (`https://www.rosettacommons.org/software`)

(b) Complete the licence process, then download Rosetta version 3.12 (the "**Rosetta 3.12 source + binaries for Linux**" option)

(c) Exact the Rosetta folder from the zip file.

(d) Rename the extracted folder to "**Rosetta**", then move it to your desired location (I chose "`/home/user1/`")

(e) In your "`/home/user1/`" directory (where "user1" is your computer user name), open your "**.bashrc**" file and enter the following:

```
export ROSETTA=/home/user1/Rosetta/
export ROSETTA_TOOLS=/home/user1/Rosetta/main/tools
export RNA_TOOLS=$ROSETTA/tools/rna_tools/
export PATH=$RNA_TOOLS/bin/:$PATH
export PYTHONPATH=$PYTHONPATH:$RNA_TOOLS/bin/
```

(f) Then save and close the ".bashrc" file.

(g) Open a terminal in "`/Rosetta/main/source`" and run the following command:

```
python2 ninja_build.py r -remake
```

(h) Once compilation finishes, close that terminal. Then open a new terminal in "`Rosetta/tools/rna_tools/`".

(i) Run this command:

```
python $RNA_TOOLS/sym_link.py
```

(j) Verify everything has been done correctly up to this point by running the following in your terminal:

```
rna_helix.py -h
```

You should see a printout of usage instructions for "`rna_helix.py`". If you do not see this, either a mistake was made in the previous steps (e.g. a spelling error), or you do not have the prerequisite items mentioned above.

(k) Now, go into your "`Rosetta/tools/rna_tools/bin/`" directory and confirm there is an "`rna_denovo`" symlink there. If it is not present, you have to create it yourself:

i. Go into the "`/Rosetta/main/source/bin/`" directory and find the "`rna_denovo`" symlink.

ii. Copy and paste the "`rna_denovo`" symlink from "`/Rosetta/main/source/bin/`" into "`Rosetta/tools/rna_tools/bin/`".

(l) To confirm the required command (`rna_denovo`) is working correctly, navigate to "`/Rosetta/demos/public/rnp_structure_prediction/`", open the "flags" file, add the following line to the bottom of the file:

```
-bps_moves false
```

Then save and close the flags file. Open a terminal in this directory and run the following command:

```
rna_denovo @flags | tee terminal_output.txt
```

You should see a lot of text output to the terminal, with a table showing scores near the end (once the command finishes). A text file named "`terminal_output`" will also be saved to this folder location which will contain the same information that was printed to the terminal. This file is created so you can obtain the scores associated with each docking attempt.

- Note: This version of Rosetta does not seem to write its scores to the "out" file when running the "`rna_denovo`" command (even though the documentation says it does). Thus, you must save the terminal output while running the "`rna_denovo`" command and then perform a "`grep`" command on the terminal output file.

(m) You can then, in the same terminal, run:

```
extract_lowscore_decoys.py 2qux_fold_and_dock.out 5
```

To get the 5 pdb files for the docked RNA-Protein complexes. Notice that the number 5 here corresponds with the number 5 in the flags file, indicating 5 structures are to be predicted.

- Note: you will have to use something like PyMol (`https://pymol.org/2/`) to view/render the pdb files.

7. Eternafold and Arnie for secondary structure prediction

(a) You can go to `https://github.com/DasLab/arnie` and follow the instructions in the "`ReadMe`" file to install Arnie and Eternafold. However, the instructions on this page assume you already well versed, and know how to do certain things, in a Linux environment. If you are new to Linux, you can follow my steps as outlined below.

(b) Install Eternafold (more accurate thermodynamic predictions) first:

i. Request a licence and Download EternaFold from `https://eternagame.org/software`

ii. Extract the zip file, rename the extracted folder to "EternaFold", then move the EternaFold folder to your desired destination. I chose "`/home/user1`" as my desired location.

- The Arnie github will tell you to simply download the package, then check your build by running a certain command (see below). This may return an error. If so, open a terminal in the "`EternaFold/src`" folder and type "make" (without quotes).

iii. Once the "make" command is finished, confirm the EternaFold build is working: within the terminal you typed "make", change directory to the EternaFold directory (type "cd .." without the quotation marks). If you did not need to run the "make" command, open a terminal in the EternaFold directory. Then type:

```
./src/contrafold predict test.seq
--params parameters/EternaFoldParams.v1
```

(the command should be all on one line with a space between ".seq" and the first hyphen). You should see this as your output:

```
~/EternaFold$ ./src/contrafold predict test.seq --params parameters/EternaFoldParams.v1
Training mode:
Use constraints: 0
Use evidence: 0
Predicting using MEA estimator.
>test.seq
CGCUGUCUGUACUUGUAUCAGUACACUGACGAGUCCCUAAAGGACGAAACAGCG
>structure
(((((((((((((.....))))))..)....((((.....))))...))))))
```

iv. Now, you can close this terminal

(c) Then install Arnie

i. First, ensure that you have a tmp folder in your "`/home/user1`" directory, where "user1" is your computer username. If you do not have this directory, go ahead and create one.

ii. Download Arnie from Github. I just downloaded the code as a zip file, unzipped it, renamed the folder to "Arnie", then moved the Arnie folder to my "`/home/user1/`" directory. I do not think the EternaFold and Arnie folders need to be in the same parent directory. I just chose to do this in an effort to reduce potential complications.

iii. Go into your Arnie folder and create a file called "arnie.txt".

iv. Open this file and enter the following:

```
eternafold: /home/user1/EternaFold/src
TMP: /home/user1/tmp
```

- This will point the Arnie program to EternaFold and the temp folder.

v. Open your ".bashrc" file and add these lines to the very bottom of the document:

```
export PYTHONPATH=$PYTHONPATH:/home/user1
export PYTHONPATH=$PYTHONPATH:/home/user1/arnie
export ARNIEFILE=/home/user1/arnie/arnie.txt
```

vi. Save your ".bashrc" document and close it.

vii. You should now be able to confirm if Arnie is set up correctly. Open up a terminal (where you open the terminal should not matter). Type the following:

```
python
>>> from arnie.mfe import mfe
>>> sequence = "CGCUGUCUGUACUUGUAUCAGUACACUGACGAGUCCCU
AAAGGACGAAACAGCG"
>>> mfe_structure = mfe(sequence,package='eternafold')
>>> print(mfe_structure)
```

(the sequence should all be on one line)

viii. Your output should look like this:

```
((((((((((((.....))))))..)....((((.....))))...))))))
```

ix. If you received an error, then you may have made a mistake in one of the previous steps or something is misspelled (the python commands, the folder names, the arnie.txt file name, the lines inside the arnie.txt file, etc.)