Sliding window RNAfold analysis tutorial

Kim Dill-McFarland, kadm@uw.edu

version May 18, 2020

Objective

To create sliding window sequences and determine RNAfold structures using command line tools.

Tutorial

Setup

Prior to the tutorial:

1. Install RNAfold from https://www.tbi.univie.ac.at/RNA/#download. If there is a pre-compiled version for your OS, I strongly recommend using that. If not, build one from the binary following the instructions provided.
2. MAC users: Find your terminal program under `Applications/Utilities/`. Open it and type `RNAfold` [Enter]. You should see that RNAfold is installed like below.
3. Windows users: Install a Unix-style command line terminal. Any one works; recommendations can be found at https://swcarpentry.github.io/shell-novice/setup.html. Open your chosen program and type `RNAfold` [Enter]. You should see that RNAfold is installed like below.

This is the RNAfold command line program. It automatically opens in regular mode (not batch) and prompts you to enter a sequence for analysis. After checking that the program is installed, you can quit for now.

```
RNAfold

Input string (upper or lower case); @ to quit
....;....1....;....2....;....3....;....4....;....5....;....6....;....7....;....8
```

Making sliding windows

Using the `fasta_windows.sh` script, make sliding window sequences from your fasta file. The basic syntax of the script is as follows. **Be sure to change the `.sh` file name to the most recent version number.**

```
bash fasta_windows_v1.0.sh <fasta> <window size> <slide size>
```

where the input fasta file is formatted with 1 line for each name and 1 line for each sequence. The window size is the length of desired window sequences in bp, and the slide size is the length to slide each window over in bp.

Example fasta input

```
>Example sequence 1
AUAGGCGGCGCAUGAGAGAAGCCCAGACC-AAUUACCUACCCAAAA-UGGAGAAAGUUCACGUUGACAUCGAGGAAGACAGCCCAUUCCUCAGAGCUUUGCAG
>Example sequence 2
AUGGGCGGCGCAUGAGAGAAGCCCAGACC-AAUUACCUACCCAAAA-UGGAGAAAGUUCACGUUGACAUCGAGGAAGACAGCCCAUUCCUCAGAGCUUUGCAG
```

If you have multiple fastas, you can concatenate them together with the following. The `*` is a wildcard that stands for any character of any length, so the following captures all `.fasta` files in the directory specified.

```
cat folder_with_fastas/*.fasta >> all.sequences.fasta
```

Then run your sequences! Here is an example process if you place your files on your computer's desktop. Results will be saved to your desktop.

```
cd Desktop/

bash fasta_windows_v1.0.sh example.fasta 50 50
```

Or you can be in any directory on your computer and use full file paths. In this case, the results will be saved in whatever folder your terminal is currently in (check where this is with `pwd` for print working directory). Note that `\` are used to split commands across multiple lines.

```
bash /Users/kim/Desktop/fasta_windows_v1.0.sh \
     /Users/kim/Desktop/example.fasta 50 50
```

In either case, the script outputs a file named `window_example.fasta` which contains the window sequences named with `window_start_end_sequenceName`

Example result

```
>window_1_50_Example sequence 1
AUAGGCGGCGCAUGAGAGAAGCCCAGACC-AAUUACCUACCCAAAA-UGG
>window_51_100_Example sequence 1
GAAAGUUCACGUUGACAUCGAGGAAGACAGCCCAUUCCUCAGAGCUUUG
```

Run RNAfold

Next, input your window result file into RNAfold.

REQUIRED

- `--infile` Fasta resulting from sliding window script
- `--outfile=FILENAME` Name for the output file. All results will be saved in this one file

OPTIONAL

- `--noPS` When used, program does not create any postscript structures
- `--jobs=2` When used, runs the program on the number of processors provided. In general, it is best to only run software on the total number of processors on your computer minus 2

```
RNAfold --infile window_example.fasta --outfile=example.fold \
        --noPS --jobs=2
```

Example result file

```
>window_1_50_Example sequence 1
AUAGGCGGCGCAUGAGAGAAGCCCAGACC-AAUUACCUACCCAAAA-UGG
...(((..(........)..))).................(((....))) ( -5.40)
>window_51_100_Example sequence 1
AGAAAGUUCACGUUGACAUCGAGGAAGACAGCCCAUUCCUCAGAGCUUUG
..(((((((...........((((((.........))))))..))))))). (-10.30)
```