Comp 598: Assignment 1

RNA bioinformatics

Collaborators: Faizy Ahsan

# Exercise 1

*Stochastic context free grammar*

G = {Σ, VN, P, S}

Σ = {(, ), •}

VN = {S, T}

P = S → ( T ) S | • S | ε

P = T → ( T ) S | • T | •

|  |  |
| --- | --- |
| production rule | rule name |
| S → ( T ) S | *pairsplit\_S* |
| | • S | *dot\_S* |
| | ε | *end* |
| S → ( T ) S | *pairsplit\_T* |
| | • T | *dot\_T* |
| | • | *dot* |

|  |  |  |
| --- | --- | --- |
|  | S | *axiom* |
|  | ( T ) S | *pairsplit\_S* |
|  | ( T ) ε | *end* |
|  | ( ( T ) S ) | *pairsplit\_T* |
|  | ( ( T ) • S ) | *dot\_S* |
|  | ( ( T ) • ( T ) S ) | *pairsplit\_S* |
|  | ( ( T ) • ( T ) ε ) | *end* |
|  | ( ( ( T ) S ) • ( T ) ) | *pairsplit\_T* |
|  | ( ( ( T ) ε ) • ( T ) ) | *end* |
|  | ( ( ( • ) ) • ( T ) ) | *dot* |
|  | ( ( ( • ) ) • ( • T ) ) | *dot\_T* |
|  | ( ( ( • ) ) • ( • ( T ) S ) ) | *pairsplit\_T* |
|  | ( ( ( • ) ) • ( • ( T ) • S ) ) | *dot\_S* |
|  | ( ( ( • ) ) • ( • ( T ) • ε ) ) | *end* |
|  | ( ( ( • ) ) • ( • ( • ) • ) ) | *dot* |



Reference:

Giegerich, Robert. "Introduction to stochastic context free grammars." RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods (2014): 85-106.

www.techfak.uni-bielefeld.de/ags/pi/lehre/RNA\_StrukturSS11/IntroStochGram.pdf

# Exercise 2

sample 1: ( ( . ( ( . . . . . ) ) ) )

sample 2: ( ( . . ( ( . . . ) ) . ) )

sample 3: ( ( ( . ( . . . . . ) ) ) )

|  |  |  |  |
| --- | --- | --- | --- |
| sample 1 | sample 2 | sample 3 | Consensus |
| 1,14 | 1,14 | 1,14 | ✔ |
| 2,13 | 2,13 | 2,13 | ✔ |
|  |  | 3,12 |  |
| 4,12 |  |  |  |
| 5,11 | 5,11 | 5,11 | ✔ |
|  | 6,10 |  |  |

Consensus: ( ( . . ( . . . . . ) . ) )

The secondary structure S of an RNA strand of length L is defined as the set of paired base positions in such that each base is paired with at most one other base, i.e., for all (i,j) in S and (i',j') in S: i=i' iff j=j'.

Pseudo-knots occur when the bonds between the base-pairs overlap. Formally, an RNA structure is pseudo-knotted iff exist (i,j) in S, (i',j') in S: i<i'<j<j'.

Now let us assume our RNA consensus structure has two base pairs. From our definition of the consensus structure, both these base pairs must occur in a majority of the samples. For our consensus structure to be pseudo-knotted, the two base-pairs must overlap. Since both the base pairs are present in majority of the samples, there have to be some samples, which contain both the base pairs, resulting in pseudo-knotting of those samples. Since we know that none of the samples have pseudo-knots, such samples are not possible and therefore, it is not possible for the consensus structure to have pseudo-knots.

Reference:

http://www.cs.ubc.ca/labs/beta/Courses/CPSC545-03/Class%20notes/class14-detailed.txt

# Exercise 3

Part 1 and 2 have been provided in HW1Q3.py. The file takes as input HW1Q3.fasta.

Part3

|  |  |  |
| --- | --- | --- |
| K | mean error | standard deviation |
| 100 | 0.47 | 0.104 |
| 1000 | 0.15 | 0.029 |
| 10000 | 0.04 | 0.010 |

We notice that as K increase, mean error decreases. More specifically, we notice that for every order increase in K, mean decreases by a factor of ~3 to 4. Therefore, as a rough estimate, we can say that for K = 100,000, mean error should be close to 0.01.

More formally, we notice an exponential relationship between log K and the mean error.

|  |  |  |
| --- | --- | --- |
| K | logK | mean error |
| 100 | 2 | 0.47 |
| 1000 | 3 | 0.15 |
| 10000 | 4 | 0.04 |

We therefore plot logK against the mean error.

From the plot, we can observe that forecast line reaches mean error 0.01, close to log K = 5. Furthermore, at K = 100,000, we find the error to be 0.011. Therefore, we estimate K to be more than but close to 100,000 for error value to be less than 0.011.

# Exercise 4

t.seq < GGGCCGCCGACCCGGCCC&CCCGGCGGUCGGGCCGGG

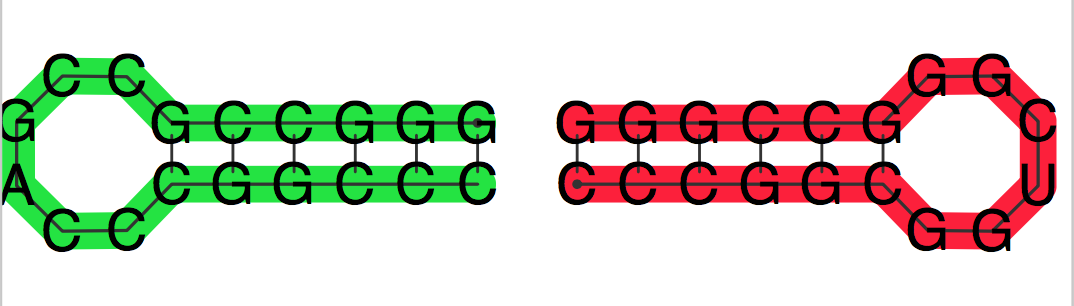
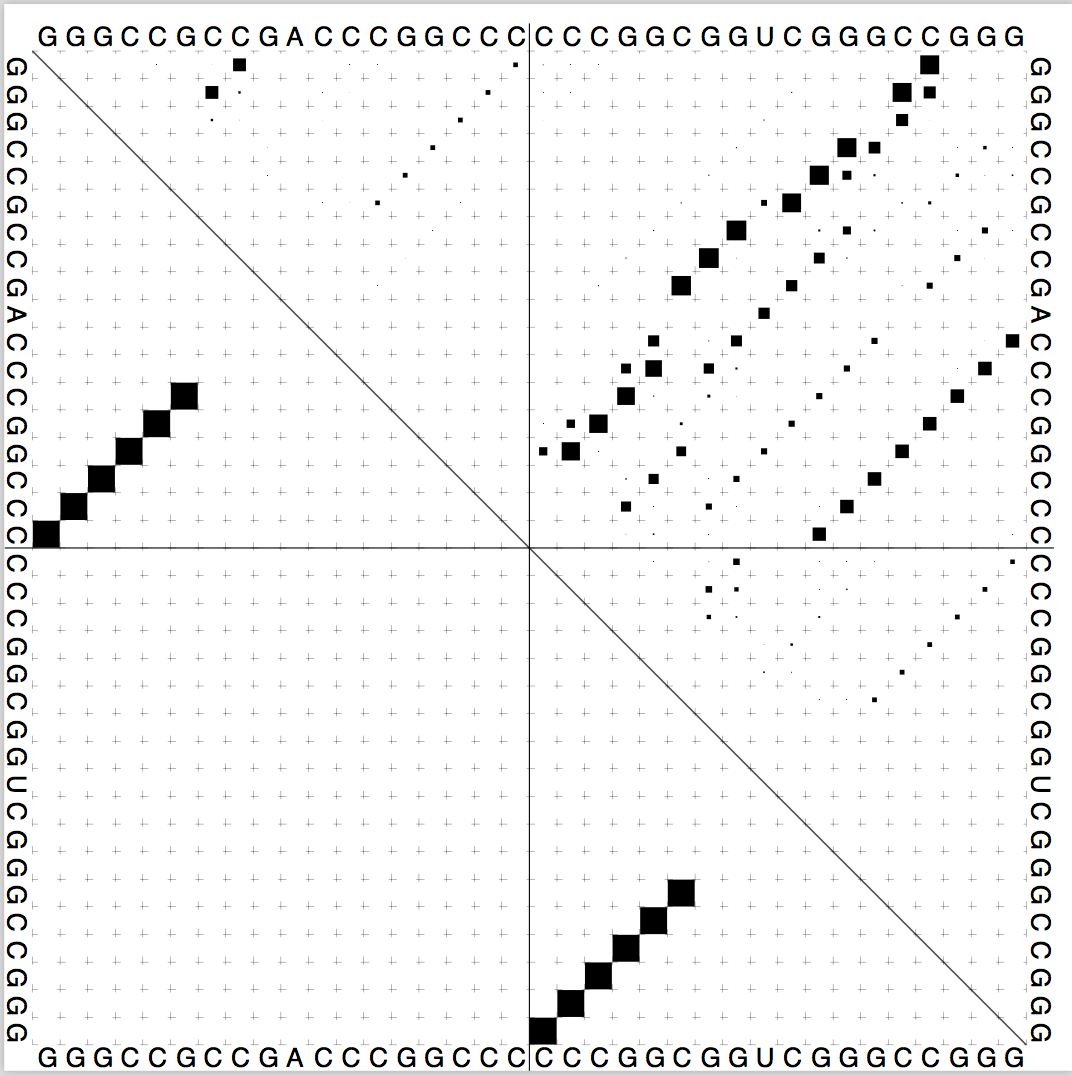
RNAcofold -p < t.seq

GGGCCGCCGACCCGGCCC&CCCGGCGGUCGGGCCGGG

((((((......))))))&((((((......)))))) (-23.70)

((.({{{((.{((((,,.&.}})}})).)))})),,. [-23.81]

frequency of mfe structure in ensemble 0.831356; delta G binding= 1.93

 dot.ps rna.ps

RNAcofold predicts that the two strands would have intra-structure bonds, each forming a hairpin and would not interact with each other.

RNA\_w25\_u1.out

# RNAup --include\_both

# 18

# GGGCCGCCGACCCGGCCC

# 18

# CCCGGCGGUCGGGCCGGG

# pos u4S dG

1 NA 0.000

2 NA -0.134

3 NA -0.134

4 6.167 -0.134

5 6.166 -0.134

6 8.799 -1.614

7 7.611 -11.998

8 6.355 -11.998

9 1.739 -11.998

10 0.000 -11.998

11 0.000 -11.998

12 0.001 -11.998

13 1.739 -7.290

14 6.514 -2.117

15 9.173 -2.117

16 9.143 -2.117

17 7.141 -2.117

18 7.141 0.000

# pos u4S

1 NA

2 NA

3 NA

4 8.920

5 8.931

6 10.880

7 10.045

8 7.422

9 4.792

10 0.000

11 0.000

12 0.001

13 4.799

14 8.687

15 11.014

16 9.376

17 7.176

18 7.175

RNAup -b < t.seq

((((((&)))))) 7,12 : 7,12 (-12.00 = -12.00 + 0.00 + 0.00)

CCGACC&GGUCGG

RNAup output in file: RNA\_w25\_u1.out

RNAup on the other hand provides us with the sites through which the two strands would interact with each other.

The predictions are different because the two algorithms employ two different approaches: concatenation vs accessibility based.

For this case, we would trust RNAup since it provides us with the interaction sites that are accessible for binding with the interaction partner. From RNAcofold, we notice that these sites are free and therefore, the interaction would not require the breaking of intra-molecular basepairs.

An advantage of the concatenation approach is that all techniques regularly used in RNA secondary structure prediction can be transferred to this cofolding approach. A disadvantage is that the restrictions of the approach make the prediction of important known interaction motifs like kissing hairpins, not-possible.

An advantage of accessibility based approach is that it can handle important interactions like kissing-hairpin. A disadvantage is that they assume a single interaction site, which may not contain bases participating in intramolecular pairings.

Reference:

Jan Gorodkin and Walter Russo, RNA Sequence, Structure, and Function: Computational and Bioinformatics Methods, Humana Press, 2014. Chapter 19.

# Exercise 5

The code contains the files evolutionary\_reactor.py, which contains all the functions and vienna\_wrap.py, which contains wrapper functions around RNAfold and RNAdistance.

