COMP 204 - Assignment 1

Yue Li

Important instructions:

- Release date: January 17, 2019 at 12:00 AM
- Due date: **February 1, 2019 at 23:59**
- Submit each of your Python programs on MyCourses. Do not zip multiple files together. Submit each file separately
- Write your name and student ID at the top of each program
- For each question, start off from the Python file given to you. *Do not change its name.*
- Do not use any modules or any pieces of code you did not write yourself
- For each question, your program will be *automatically* tested on a variety of test cases, which will account for 75% of the mark. To be considered correct, your program should print out *exactly and only* what the question specifies. Do not add extra text, extra spaces or extra punctuation. Do not ask the user to enter any other information than what is needed.
- For each question, 25% of the mark will be assigned by the TA based on (i) your appropriate naming of variables; (ii) commenting of your program; (iii) simplicity of your program (simpler = better).

1 Decision tree risk assessment for Deficiency of Adenosine Deaminase 2 (DADA2) (25 points)

Download the `dada2DecisionTree.py` Python program on MyCourses.

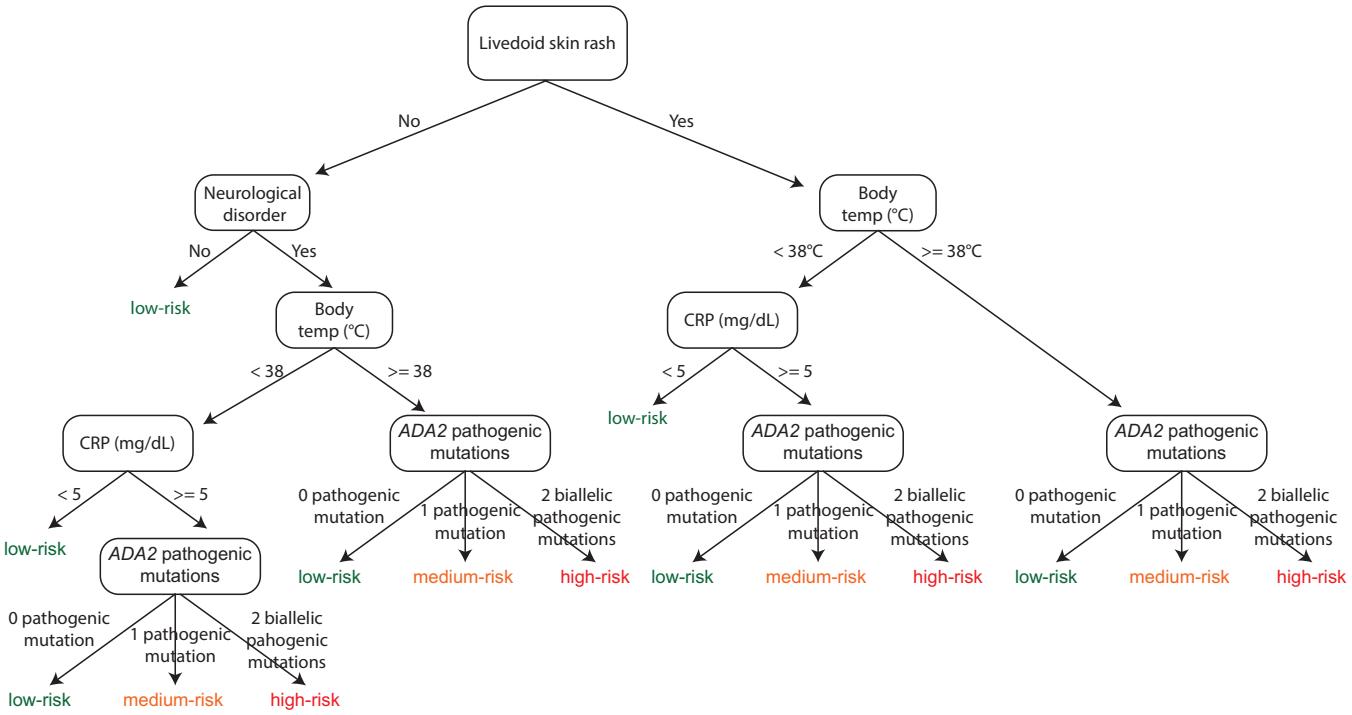

Figure 1: Decision tree for genetic diagnosis of DADA2. The schematics is adapted from Figure 3 presented by Rama et al., (2018) [2].

Question	Valid answers
`Livedoid skin rash:`	`Yes` or `No`
`Body temp:`	numeric float
`Neurological disorder:`	`Yes` or `No`
`CRP (mg/dL):`	numeric float
`ADA2 pathogenic mutations:`	`0 pathogenic mutation` or `1 pathogenic mutation` or `2 biallelic pathogenic mutations`

Table 1: Valid questions and answer from the program

Deficiency of adenosine deaminase 2 (DADA2) is a recently discovered rare autosomal recessive systemic autoinflammatory disorder [1]. It is due to mutations in the *ADA2* gene, which encodes the adenosine deaminase 2 protein. The disease onset usually occurs prior to 24 years of age. The clinical spectrum ranges from single cutaneous lesions to severe systemic inflammatory disease with cerebral complications.

To help diagnosing DADA2, we will implement a decision tree (Figure 1) adapted from [2]. Based on this decision tree, write a Python program that asks a series of questions to the user to determine their level of risk. Your program should only ask the questions that are necessary.

Specifically, use exactly the text of the questions and only the user answers listed in Table 1 are valid answers:

If the answer provided by the user is invalid (see below), the program should print *Invalid*, stop asking further questions, and print nothing else. At the end, your program should print either *low-risk*, *medium-risk*, or *high-risk*, based on the user's answers, or *Invalid*.

Below are a few examples of the way your program should behave. The text in blue are the user's answers.

```
Livedoid skin rash (Yes or No)?  No
Neurological disorder (Yes or No)?  No
low-risk

Livedoid skin rash (Yes or No)?  No
Neurological disorder (Yes or No)?  Yes
Body temp:  39
ADA2 pathogenic mutations?  2 biallelic pathogenic mutations
high-risk

Livedoid skin rash?  Yes
Body temp:  37.5
CRP (mg/dL)? 5
ADA2 pathogenic mutations:  0 pathogenic mutation
low-risk

Livedoid skin rash:  Maybe
Invalid

Livedoid skin rash:  Yes
Body temp:  37
CRP (mg/dL): high
invalid
```

2 Palindromic sequence (25 points)

Download the `palindrome.py` Python program on MyCourses.

A palindromic sequence is a nucleic acid sequence on double-stranded DNA or RNA wherein reading 5' (five-prime) to 3' (three prime) forward on one strand matches the sequence reading 5' to 3' on the complementary strand with which it forms a double helix. For example, the DNA sequence ACCTAGGT is palindromic because its nucleotide-by-nucleotide complement is TGGATCCA, and reversing the order of the nucleotides in the complement gives the original sequence. Palindromic sequences play an important role in molecular biology. Many restriction endonucleases (restriction enzymes) recognize specific palindromic sequences and cut them. Learn more from Wikipedia (`https://en.wikipedia.org/wiki/Palindromic_sequence`). Write a program that checks whether a user-input DNA sequence is a palindromic sequence.

For example, if user inputs GAATTC, then your program should first convert GAATTC to the complementary sequence CTTAAG. It will then reverse the sequence to GAATTC. Finally, it will compare GAATTC with the original sequence to see whether the two sequences are identical. It will then prints to the screen "GAATTC is a palindromic sequence". If user inputs AGCGAT, your program should output "AGCGAT is not a palindromic sequence". If user inputs a sequence containing characters other than A, C, T, and G, your program should output "Invalid sequence" and nothing else.

Here are the above examples shown during the runtime of the program (user input is coloured in blue):

```
Enter a DNA sequence:  GAATTC
GAATTC is a palindromic sequence

Enter a DNA sequence:  AGCGAT
AGCGAT is not a palindromic sequence

Enter a DNA sequence:  A*CGAUGXYZ
Invalid sequence
```

3 Counting CpG sites (20 points)

Download the `CpG.py` Python program on MyCourses.

The CpG sites or CG sites are regions of DNA where a cytosine nucleotide is followed by a guanine nucleotide in the linear sequence of bases along its 5' $\rightarrow$ 3' direction. Cytosines in CpG dinucleotides can be methylated to form 5-methylcytosine. Enzymes that add a methyl group are called DNA methyltransferases. In mammals, 70% to 80% of CpG cytosines are methylated. Methylating the cytosine within a gene can change its expression, a mechanism that is part of a larger field of science studying gene regulation that is called epigenetics.

Write a program that outputs the positions of all of the CpG site harboured within a given sequence. We will use zero-based system with the first nucleotide position indexes at 0. If the DNA sequence contains no CpG, your program should output "No CpG site is detected in the input sequence". If the user-input sequence contains letters other than A, C, G, and T, your program should print "Invalid sequence" and nothing else.

For example, your program should behave as follows (user inputs are coloured in blue):

```
Enter a DNA sequence:  ACGTGCGCT
CpG site is detected at position 1 of the sequence
CpG site is detected at position 5 of the sequence

Enter a DNA sequence:  AGCTGGCCT
No CpG site is detected in the input sequence.

Enter a DNA sequence:  A*CGAUGXYZ
Invalid sequence
```

4 MicroRNA target sites (30 points)

Download the `microRNA.py` Python program on MyCourses.

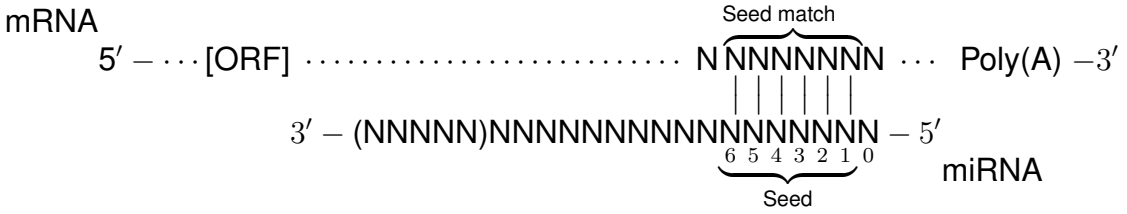

Figure 2: Canonical miRNA Watson-Crick base pairing to the 3′UTR of the mRNA target site. The most critical region is a 6-nucleotide (or 6mer) site termed as the "seed" occurs at the 1-6 position of the 5′ end of the miRNA (here once again the first nucleotide indexes at 0, the second nucleotide indexes at 1, and so forth). ORF: open reading frame

MicroRNAs (miRNA) are small 22 nucleotide endogenous RNA molecules [3, 4] (Figure 2). They regulate the gene expression by binding to the mRNA sequences. The *seed sequence* or *seed region* of miRNAs is essential for the binding to the mRNA. The seed region is an evolutionarily conserved 6-nucleotide sequence, which is mostly situated at *positions 2-7* (one-based counting is used here) from the miRNA 5'-end. Even though base pairing of miRNA and its target mRNA does not need to match perfectly, the "seed sequence" has to be perfectly complementary. Just a refresh of memory, the sequence complementarity implies Watson-Crick pairing that A pairs with U and C pairs with G.

Note that both the miRNA sequence and mRNA sequence are shown from 5' to 3' direction[1]. A miRNA must pair with a mRNA in 3' to 5' direction to form stable RNA duplex (Figure 2). For example, the sequence miRNA *miR-375* is `5'-UUUGUUCGUUCGGCUCGCGUGA-3'` and part of its target mRNA sequence at the 3' untranslated region (UTR) region for the gene *Mtpn* is `5'-UAAGUUUCGUGUUGCAAGAACAAA-3'`. The pairing takes place as follows:

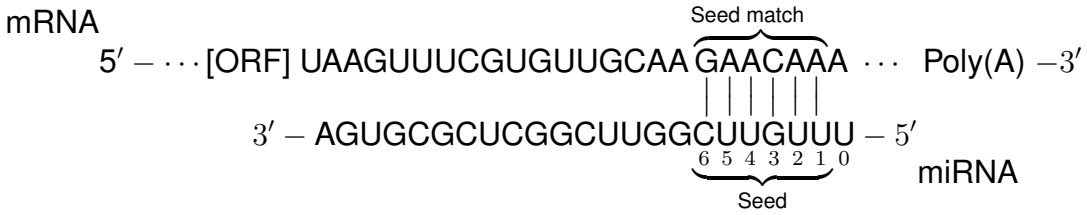

Figure 3: Example of microRNA seed region and target site.

Given a miRNA sequence and a potentially target mRNA sequence, write a program to output the number of miRNA binding sites in the mRNA sequence. For instance, in the above example, your program should produce the following (user inputs are coloured in blue):

[1]The 5' and 3' mean "five prime" and "three prime", which indicate the carbon numbers in the RNA's sugar backbone. The 5' carbon has a phosphate group attached to it and the 3' carbon a hydroxyl (-OH) group. This asymmetry gives a RNA strand a "direction". For example, ribosome reads mRNA from 5' to 3' to translate it into amino acid sequence to produce protein. Therefore, a RNA sequence is presented commonly in 5' $\rightarrow$ 3' direction, which is somewhat analogous to reading this English text from left to right.

```
Enter a mRNA sequence:  UAAGUUUCGUGUUGCAAGAACAAA
Enter a microRNA sequence:  UUUGUUCGUUCGGCUCGCGUGA
The mRNA has 1 target site(s)
```

Here is another example for longer mRNA sequence segment from gene *lin-14* in worm (*C. elegans*) with 3 target sites matching with miRNA *lin-4*. Note that although the mRNA sequence is displayed below in multiple lines, **the actual mRNA sequence entered at the user prompt must be a single-line long sequence**. To test your program, you can copy and paste the single-line RNA sequence from the text file named mRNA_lin14_3utr.txt included in this assignment.

```
Enter a mRNA sequence:  AAUGCCAAUUUUUUCGAGUCAUCCUUCGGGCAAUGUUCAUUACACUUUCUCUCU
GUUGUACUUGAGCAUGUUUCAAUUUCAAUCACAAAUGCCUUUUUUGGAGAGAAUUGAAGGCAAAACCAAACUAAAUU
GAGUUUGGUGGAAAUUGAAAUUGCAACAUGCUCAAUUCAUUUUGUUUUUUUCUUUUAACUAUAUGGAUGCCACGCUG
GAUUGACUCUUCCGUACACUCACGCUCAUUCCAAAUUAUCCCCAUCUGACCCCGGACUGUUUUAUCCAAACCUAUAC
UGAUGUAUAUUGUAUGCGGCAUGUUAUUUUUUCAUAACCACAAGCAUUUAGCGCUUUCGGCUUCAGAUUUCUAAUCG
CUGUUUCUUUACUGCUUUGCCCUUUUUUCUAACUUCUGAAUUUCGUUAUUAUGCAACAAUUCUACCUCAAUUUUUUU
GUAUCUUUCCCCAAUUCUAGUUGACAUUAUUGGUUUUGAUUGUUUUCCUAGCUCUGAAAUCCCACAACUAGGCGCCA
CUGAAUUGUAUCUUUCAUUCCCACUUUCGUUUUUCGCACAUUGUAGUAUCUCUCAAUUAGGAAACACUUGAAAACCU
CUAUUGCUUAGGAUUGAUGAGAGAUAUUGCUUUUUCCUGCACUCACUUUACCUUUGUCUCACCUCAAAAAUUGCUCU
CAGGAACAUUCAAAACUCAGGAAUUUGUCACCUUGGUCUCUCAUCAUAUAUCUUACCUCUUGUCACACCCCCCAUCC
CCAGUCUCAAGUUCAUUUCUUACUUUGUAACUCCGUUUAGUGCGCCCAAUUCUCGUCAUUUUGAUUACACUCUCUUU
UAAUCCAACUCAGGGAACCAAUUUUUUUUCUCAUUGAACUCAGGAAUUUCUUCUACCUCAGGGAACCUACCUCAUCC
ACUUUUCAGUUGUUUGGGGCCAAAUAUCUAUAUCCAAAGUAGUAGUCUACAAUUUAGUAUUUUAUUAUUACCUCCCG
CCGUUUUAGCUUUUAAUGUUAAAAUCAGGAACUUUUGAAAAUGAUCUUCACCUCAUUCAGAAGCAAAAAUCAGGCAU
UUUCCAAAGAUUUUGAAAACACAUAAACCUCCUUCCAAGUCAAAACUCACAACCAACUCAGGGACCUUUUUCUUACU
UCUGUAUCACAAAAAUGAUUAUAUUUCUGAUGAAUGUUGCUCUGUCAUAAAUCAAUUUAUUUCUUUUG

Enter a microRNA sequence:  UCCCUGAGACCUCAAGUGUGA

The mRNA has 3 target site(s)
```

Hint: you can check whether your program produces the correct results by extracting the seed sequence manually from the miRNA sequence and search using a text editor. For example, the seed sequence (i.e., the nucleotide position 1-6 in zero-based system) for miRNA sequence UCCCUGAGACCUCAAGUGUGA is CCCUGA. We can convert it to reverse complement sequence as UCAGGG. We can then use a text editor to search for UCAGGG in the long mRNA sequence. Your program should produce the same number of occurrences as shown in your text editor.

In this question, you can assume that the user-input miRNA and mRNA sequence are valid (i.e., containing only A, C, U, and G).

Bonus (10 extra points)

Modify your program so that it outputs the nucleotide location(s) within the mRNA sequence, that indicates the start of the base-pairing between miRNA seed region and mRNA. Start from 0 to count the location. If there are multiple locations matching the miRNA sequence, then print each matching position one per line.

In the first above example, the program will produce (user input sequences are in blue):

```
Enter a mRNA sequence:  UAAGUUUCGUGUUGCAAGAACAAA
Enter a microRNA sequence:  UUUGUUCGUUCGGCUCGCGUGA
The mRNA has 1 target site(s)
microRNA seed match is found at mRNA sequence position 17
```

In the second example, your program should produce (user input sequences are in blue):

```
Enter a mRNA sequence:  AAUGCCAAUUUUUUCGAGUCAUCCUUCGGGCAAUGUUCAUUACACUUUCUCUCU
GUUGUACUUGAGCAUGUUUCAAUUUCAAUCACAAAUGCCUUUUUUGGAGAGAAUUGAAGGCAAAACCAAACUAAAUU
GAGUUUGGUGGAAAUUGAAAUUGCAACAUGCUCAAUUCAUUUUGUUUUUUUCUUUUAACUAUAUGGAUGCCACGCUG
GAUUGACUCUUCCGUACACUCACGCUCAUUCCAAAUUAUCCCCAUCUGACCCCGGACUGUUUUAUCCAAACCUAUAC
UGAUGUAUAUUGUAUGCGGCAUGUUAUUUUUUCAUAACCACAAGCAUUUAGCGCUUUCGGCUUCAGAUUUCUAAUCG
CUGUUUCUUUACUGCUUUGCCCUUUUUUCUAACUUCUGAAUUUCGUUAUUAUGCAACAAUUCUACCUCAAUUUUUUU
GUAUCUUUCCCCAAUUCUAGUUGACAUUAUUGGUUUUGAUUGUUUUCCUAGCUCUGAAAUCCCACAACUAGGCGCCA
CUGAAUUGUAUCUUUCAUUCCCACUUUCGUUUUUCGCACAUUGUAGUAUCUCUCAAUUAGGAAACACUUGAAAACCU
CUAUUGCUUAGGAUUGAUGAGAGAUAUUGCUUUUUCCUGCACUCACUUUACCUUUGUCUCACCUCAAAAAUUGCUCU
CAGGAACAUUCAAAACUCAGGAAUUUGUCACCUUGGUCUCUCAUCAUAUAUCUUACCUCUUGUCACACCCCCCAUCC
CCAGUCUCAAGUUCAUUUCUUACUUUGUAACUCCGUUUAGUGCGCCCAAUUCUCGUCAUUUUGAUUACACUCUCUUU
UAAUCCAACUCAGGGAACCAAUUUUUUUUCUCAUUGAACUCAGGAAUUUCUUCUACCUCAGGGAACCUACCUCAUCC
ACUUUUCAGUUGUUUGGGGCCAAAUAUCUAUAUCCAAAGUAGUAGUCUACAAUUUAGUAUUUUAUUAUUACCUCCCG
CCGUUUUAGCUUUUAAUGUUAAAAUCAGGAACUUUUGAAAAUGAUCUUCACCUCAUUCAGAAGCAAAAAUCAGGCAU
UUUCCAAAGAUUUUGAAAACACAUAAACCUCCUUCCAAGUCAAAACUCACAACCAACUCAGGGACCUUUUUCUUACU
UCUGUAUCACAAAAAUGAUUAUAUUUCUGAUGAAUGUUGCUCUGUCAUAAAUCAAUUUAUUUCUUUUG
Enter a microRNA sequence:  UCCCUGAGACCUCAAGUGUGA
The mRNA has 3 target site(s)
microRNA seed match is found at mRNA sequence position 832
microRNA seed match is found at mRNA sequence position 880
microRNA seed match is found at mRNA sequence position 1111
```

Hint: to save seed match location(s) in the mRNA sequence, you may use Python List structure `https://docs.python.org/3/tutorial/datastructures.html#using-lists-as-stacks`.

References

[1] Qing Zhou, Dan Yang, Amanda K Ombrello, Andrey V Zavialov, Camilo Toro, Anton V Zavialov, Deborah L Stone, Jae Jin Chae, Sergio D Rosenzweig, Kevin Bishop, et al. Early-onset stroke and vasculopathy associated with mutations in ada2. *New England Journal of Medicine*, 370(10):911–920, 2014.

[2] Mélanie Rama, Claire Duflos, Isabelle Melki, Didier Bessis, Axelle Bonhomme, Hélène Martin, Diane Doummar, Stéphanie Valence, Diana Rodriguez, Emilie Carme, David Genevieve, Ketil Heimdal, Antonella Insalaco, Nathalie Franck, Viviane Queyrel-Moranne, Nathalie Tieulie, Jonathan London, Florence Uettwiller, Sophie Georgin-Lavialle, Alexandre Belot, Isabelle Koné-Paut, Véronique Hentgen, Guilaine Boursier, Isabelle Touitou, and Guillaume

Sarrabay. A decision tree for the genetic diagnosis of deficiency of adenosine deaminase 2 (DADA2): a French reference centres experience. *European Journal of Human Genetics*, 26(7):960–971, 2018.

[3] David P Bartel. MicroRNAs: target recognition and regulatory functions. *Cell*, 136(2):215–233, January 2009.

[4] David P Bartel. Metazoan MicroRNAs. *Cell*, 173(1):20–51, March 2018.