

# COMP 204 - Assignment 1

Mathieu Blanchette

September 13, 2019

## Important instructions:

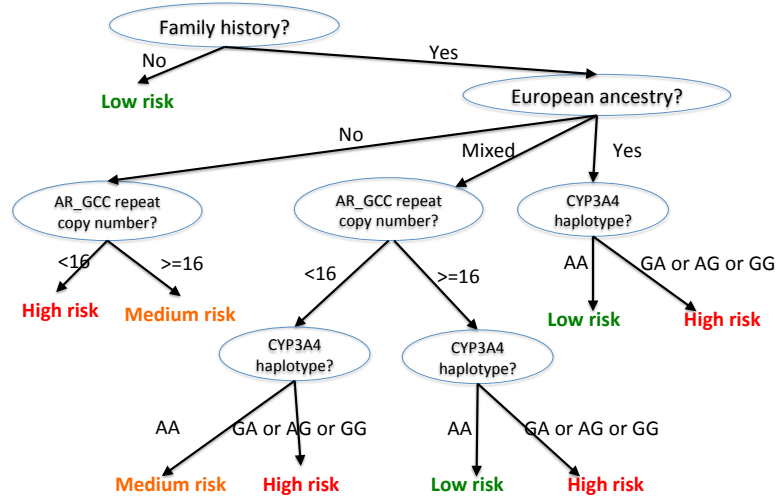
- Due date: Sept. 27, 23:59
- Submit each of your Python programs on MyCourses. Do not zip multiple files together. Submit each file separately
- Write your name and student ID at the top of each program
- For each question, start off from the Python given to you. *Do not change its name.*
- Do not use 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 Prostate Cancer Risk assessment (20 points)

Required knowledge: Conditionals and nested conditionals.

Download the prostateCancerRisk.py Python program on MyCourses.

The risk for a man to develop prostate cancer can be estimated based on their family history, their ethnicity, and two genetic factors: the number of GCC repeats in the androgen receptor (AR) gene and the genotype of the CYP3A4 gene. The decision tree below describes a strategy that could be used by a male to assess his level of risk. 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. In your program, use exactly the text used in the decision tree. 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*.



**Disclaimer:** Do not use this test to self-assess your risks of prostate cancer. Talk to a doctor instead!

Here are a couple of examples of the way your program should behave. The text in *italics* are the user's answers.

Family history? *Yes*

European ancestry? *No*

AR\_GCC repeat copy number: *19*

Medium risk

Family history? *No*

Low risk

Family history? *Yes*

European ancestry? *Mixed*

AR\_GCC repeat copy number: *12*

CYP3A4 haplotype: *GA*

High risk

Family history? *Maybe*

Invalid

Family history? *Yes*

European ancestry? *Mixed*

AR\_GCC repeat copy number: *Twelve*

Invalid

The following answers should be the only ones considered valid:

Family history? Yes or No

European ancestry? Yes or No or Mixed

AR.GCC repeat copy number: Any integer greater or equal to 0  
CYP3A4 haplotype: AA or AG or GA or GG

## 2 Blood type compatibility (20 points)

Required knowledge: Conditionals and nested conditionals.

Download the bloodTypes.py Python program on MyCourses.

Human blood types are characterized by the antigen group (either A, B, AB, or O), and by the rhesus (either positive or negative). Blood transfusions between a donor and a recipient are only possible if the donor is compatible with the recipient. The rules of antigen compatibility are described here:

<https://www.hema-quebec.qc.ca/sang/savoir-plus/groupe-sanguins.en.html>

For example, A can give to AB, but AB cannot give to B. Rhesus rules are simple: Rh- can give to both Rh- and Rh+, but Rh+ can only give to Rh+.

Complete this Python program to make it print one of three possible words, and nothing else: *Compatible* or *Incompatible* or *Invalid*. Your program should print *Compatible* if the donor is compatible with the recipient. It should print *Incompatible* if the donor is not compatible with the recipient. It should print *Invalid* if one or both of the blood types entered by the user is invalid. Valid blood types are the strings: O+, O-, A+, A-, B+, B-, AB+, AB-.

## 3 DNA sequences (20 points)

Required knowledge: Loops, conditionals, string manipulations

Download the nucleotideComposition.py Python program on MyCourses.

Write a program that counts the number of A, C, G, T in a DNA sequence. Your program should ask the user to enter a DNA sequence made of A, C, G, and T. It should then print four numbers, corresponding to the number of A's, C's, G's, and T's, separated by spaces, and nothing else. For example, given the sequence AGGCAG, your program should print 2 1 3 0. If the input sequence contains any character other than A, C, G, or T (in capital letters), your program should *not* print out any number, but instead print out the word *Invalid*.

## 4 Number of cut sites for restriction enzyme (20 points)

Required knowledge: Loops, conditionals, string manipulations, accumulator

Download the DNARestriction.py Python program on MyCourses.

A restriction enzyme is a protein that recognizes specific DNA subsequences and cuts the DNA at those positions. For example, the HindIII restriction enzyme cuts DNA whenever it encounters the 6-nucleotide pattern AAGCTT. Write a program that asks the user to enter a DNA sequence and the cut-site sequence of a restriction enzyme, and counts the number of cut sites that will result from the digestion of the given DNA sequence with that enzyme. For example:

Given sequence AAGCGATCGACAAGCTTGCAGAAAGCTTCA and cut-site AAGCTT, your program should print 2.

Given sequence AAGCGATCGACAAGCTTGCAGAAAGCTTCA and cut-site AAGC, your program should print 3.

Given sequence CAGCGAGCAGCAGAC and cut-site TTAG, your program should print 0

Given sequence ACACACACACACACA and cut-site ACAC, your program should print 6

You can assume that all sequences provided as input are valid, so you do not need to check their validity.

## 5 Translation of RNA sequence to amino acid sequence (20 points)

Required knowledge: Loops, conditionals, string manipulations, accumulator

Download the mRNATranslation.py Python program on MyCourses.

Write a program that asks the user to enter a messenger RNA sequence and that then calculates and prints the *length* of the protein sequence, measured in number of amino acids, that is encoded by the given mRNA. Refer to

Here are some examples of mRNA sequences and the length of the encoded protein:

AUGAUGAUGAUGUGAUGAUGAUGA: 5

4