**Junior Software Developer - QMUL31542**

Dear candidate, you must answer the following questions and return your answers in a zipped file to [mary.stapleton@qmul.ac.uk](mailto:mary.stapleton@qmul.ac.uk) by the end of Friday 14th April.

**Question 1**

Download the following files:

* <https://snp-nexus.org/static/test.vcf>
* <https://snp-nexus.org/static/gene_coordinates.tsv>

The first file contains a list of genetic variations in a VCF format. A Variant Call Format (VCF) file is a standard tab-separated text file with multiple rows as heading, widely used in bioinformatics to store genetic variants identified through DNA sequencing or genotyping. The complete format specification is found at <https://samtools.github.io/hts-specs/VCFv4.2.pdf>.

The second file is a map between a gene and its genomic coordinates. This file is also tab-separated and the columns are:

1. Gene name
2. Chromosome
3. Start coordinate
4. End coordinate

For this exercise, we want you to write a Python script that reads the genomic variants from the VCF file and map them to their corresponding gene(s) based on their chromosomal position. The output should be a new tab-separated file with the following columns:

CHROM, POS, REF, ALT, INFO, GENE

Example:

Consider a variant from a VCF file:

#CHROM      POS        ID    REF    ALT   QUAL     FILTER    INFO

chr6               410257    .       T        C       0              PASS        ADP=159;HET=1;HOM=0;NC=0;WT=0

And this gene from the gene\_coordinates file:

#Gene   CHROM   START\_POS     END\_POS

IRF4      chr6          391739              411447

As you can see, the CHROM from both files is the same, and the POS from the VCF is inside the range formed by the START\_POS and END\_POS coordinates of the gene. This means that this specific variant overlaps this gene, so we write the output with the fields required:

Output file:

#CHROM      POS        REF    ALT     INFO             GENE

chr6              410257      T        C       ADP=159;HET=1;HOM=0;NC=0;WT=0        IRF4

You must consider that a genomic variant may not overlap any gene. In that case, write a full-stop character (“.”) in the GENE column. It may also happen that a variant overlaps more than one gene. In that case, write all genes separated by a semi-colon (“;”) in the GENE column.

It’s important that your code is clear and well commented. Try to optimise your code for speed but do not use Pandas or Numpy for this exercise. Add a short text explaining how you tackle the problem and how you optimised your code for speed.

**Question 2**

For this question, assume you have two database tables with the following schemas:

* variants: id(primary key), chrom, position, ref, alt
* gene\_coordinates: gene(primary key), chrom, start\_pos, end\_pos

Write a SQL JOIN query that retrieves all fields from the variants table and adds the gene column from gene\_coordinates. For this question, assume that one variant can overlap none or only one gene. All the rows from the variants table must be present in the resulting table.

**Question 3**

Questions 1 and 2 represent two different ways to solve a similar problem. Which one do you think is faster and why? If the number of variants (in the VCF or in the variants table) was 10 million, what ideas would you have to improve the speed (could be hardware, software, configuration, etc)?