BI-GY 7453: Prof. Manpreet S Katari
April 02, 2023

ASSIGNMENT 4: Due Date: 16 April 2024

File Format to submit- .ipynb with output mentioned in the file (No grades will be allocated if the file format is different or it lacks output)

Next-generation sequencing technologies are making a large impact in Biology today. The ability to sequence large genomes and RNA molecules from cells in different conditions has allowed biologists to propose many different experiments. At the root of all the analysis, the sequences have to be first aligned to the genome so you know where the sequence comes from.

For this assignment, you will parse the results of an alignment (called a SAM file) and compare it to a file that contains gene annotations (GTF) . The goal is to count the number of sequences in the SAM file that match each of the genes in the GTF file. The documentation for the SAM file can be found here.(https://samtools.github.io/hts-specs/SAMv1.pdf). The name of the sam file is sample.sam. The two columns of interest for our purpose are column 3, the name of the chromosome, and column 4, the start position of the alignment. For simplicity of the homework, we are not worried about the strand and the length of the match.
An example of two lines from the File look something like this:

```
HANNIBAL_4_FC308YYAAXX:5:1:5:247	0	Chr3	14202145	60	50M	*	0	0	CGGCCGGGGGACGGACTGGGAACGGCTCTCTCGGGAGCTTTCCCCGGGCG	aaaa_]]aaaa`Z]aaaaaaaaaaaa__XXU]\[UGXKRZXPRMUOXVV[	AS:i:0	XN:i:0	XM:i:0	XO:i:0	XG:i:0	NM:i:0	MD:Z:50	YT:Z:UU	NH:i:1

HANNIBAL_4_FC308YYAAXX:5:1:5:527	0	Chr2	8023	0	46M4S	*	0	0	AATGTAGGCAAGGGAAGTCGGCAAAATGGATCCGTAACTTCGGGAACTGT	aaaaaaaaaa^X[__^X^a`_^^__`_[^_`__`X[X^`^\UGUOXU[^[	AS:i:-8	ZS:i:-8	XN:i:0	XM:i:0	XO:i:0	XG:i:0	NM:i:0	MD:Z:46	YT:Z:UU	NH:i:2
```

Now the GTF file (Arabidopsis.gtf) is an already sorted tab-separated file. It contains the chromosome in the first column and the start and end positions of an exon in the 4th and 5th columns, and the gene name in the last column. To determine the number of reads that match a gene, count the number of start positions from the SAM file that fall between the start and stop coordinates of an exon that belongs to a gene.
Here is quick look at the GTF file.

```
Chr1	TAIR10	exon	3631	3913	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	3996	4276	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	4486	4605	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	4706	5095	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	5174	5326	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	5439	5899	.	+	.	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	3760	3913	.	+	0	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	3996	4276	.	+	2	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	4486	4605	.	+	0	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	4706	5095	.	+	0	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	5174	5326	.	+	0	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	CDS	5439	5630	.	+	0	transcript_id "AT1G01010.1"; gene_id "AT1G01010"; gene_name "AT1G01010";
Chr1	TAIR10	exon	5928	6263	.	-	.	transcript_id "AT1G01020.1"; gene_id "AT1G01020"; gene_name "AT1G01020";
Chr1	TAIR10	exon	6437	7069	.	-	.	transcript_id "AT1G01020.1"; gene_id "AT1G01020"; gene_name "AT1G01020";
Chr1	TAIR10	exon	7157	7232	.	-	.	transcript_id "AT1G01020.1"; gene_id "AT1G01020"; gene_name "AT1G01020";
```

Notice that the third column defines the feature type. We are interested in the exons only, so ignore the CDS rows in the file. When you count the sequences that map to the exon, you should associate them with the gene_name provided in the last column. Many genes have multiple exons.

Suggested strategy:
- First, split all the alignments in the SAM file based on chromosomes. Create a dictionary where the key is the chromosome name and the value is a list of start coordinates. Note that some lines do not have a value for chromosomes which means that sequence did not align to the genome, so we can ignore them.
- Next go through each exon in the GTF file and count the number of reads that are between the start and end coordinates. Since we know which chromosome the gene is in you should have to go through the start positions one at a time to see how many are within the start and end coordinate.

1) **Implement without sorting the sam file**: Use the original sam file to determine the number of reads that match a gene, count the number of start positions from the SAM file that fall between the start and stop coordinates of an exon that belongs to a gene.
2) **Implement after sorting the sam file**: For each chromosome sort the alignment coordinates. You can use any of the functions discussed in class. Now again count the number of alignments that match a gene. ***Note*** Since the alignment coordinate is sorted, you don't have to check every single value in the list. Once your alignment coordinate is greater than the end coordinate of the exon, you can stop. Additionally, you can remove the alignment coordinates from the list because the genes are sorted so you will never have to look before the position you have already visited.

In both the above cases calculate the total time taken with sorting and without sorting the coordinates from the sam file and clearly explain why sorting reduces/increases the total time for execution. You can use the below-mentioned code to calculate the total time taken to execute a function.

```
# in Python
import time
start = time.process_time()
# your code here
print(time.process_time() - start)
```