Homework 3

String manipulation

Write a python program to calculate some statistics from this DNA sequence. You will need to initialize a string variable in your script so your code can start something like this template code I started for you here.

1. Calculate and print the GC content of the DNA
2. Find the location of all the ATG codons in the sequence
3. Print the reverse complement of the sequence

Math calculations

Download the genome annotation for the Chr6 of Rice again (or reuse what you had from before) Rice Chr6 annotation.

1. Compute the number of gene and exon features in the file using python
2. Compute the number of bases which are in genes and in CDS features. Report the % of the chromosome which is coding (e.g. covered by CDS exons). Assume the CDS exons in the GFF files are NON-overlapping for this problem.

- If you are a more advanced programmer, try to solve this problem by also correcting for the fact that alternative splicing isoforms will produce redundant instances of a CDS. You could also manipulate the input GFF3 file if that is easier - you do not have to read in the GFF3 - you could for example convert to BED format ...

Use the following basic code template

Compute SNP frequency

Use the following files SNPs and annotation

You will likely need to re-use your solutions from Homework 2.

1. Using the data which present the SNPs and the genes, we would like to find genes which have the most number of changes. This will require counting the number of SNPs in each gene and then dividing the number of SNPs by the length of the gene in kilobases.

Generate a report that has four columns like this - you will likely need to use BEDTools to generate the information about which genes have SNPs and you will need to distill that down into the count of SNPs per gene. You will need to calculate the length of each gene feature. Finally you will need to compute the 4th column by dividing the SNP count by the gene length and adjust that to kilobases instead of bases. Here is example of expected output.

```
gene_name       length  SNP     SNPs_per_kb
OS06G0510200    1391    50      35.9453630482
OS06G0487620    553     20      36.1663652803
OS06G0120200    2568    93      36.214953271
```

There are several ways to solve this. You should try your best to do as much of it as you can in python but you can use BEDtools, grep, cut, sort, etc. In particular you will need to use some of the string functions like split to extract out some of the parts you don't want from that last column.

You should use some of the code you had from the previous problem to manage to read in the gene information and keep track of the length of the genes.

You will need one more python type which allows you to store something in a list where the lookup is a string. The basic code for this is written you just need to use it.