Department of Computing and Information Systems
COMP90016
Workshop 4

You worked on the task of implementing an aligner for short reads and a reference in the group assignment. The assignment only required you to find perfect matches of sequences between the reads and the reference. While this is a favorable approach for computational complexity, it does not utilize the data well, as it misses potential alignments.
In this workshop we are attempting to get closer to a useful alignment program by allowing differences between reads and reference.
The reference and reads data is the same from the group assignment, which can be accessed from /home/subjects/comp90016/assignments/group_assignment/ or LMS.

1. Write a program that takes a reference *reference.fa*, a set of reads *reads.fa*, and an integer value [Z] as inputs.
   The program should identify the position in the reference with the **lowest Hamming distance** (up to [Z]). Both strands should be explored for this purpose.
   The program should report one line per read as output as specified in the group assignment: *READ_NAME, REF_NAME, POS, STRAND, NUMBER_OF_ALIGNMENTS.*
   An additional Field *HAMMING_DISTANCE* at the end of each alignment should report the Hamming distance of the alignment, if smaller of equal to [Z], or "*" otherwise.
   As before, report the lowest coordinate with respect to the reference's 5' end if multiple lowest distance alignments are found.

2. Run you program on the data from the group assignment and compare the results.
   Are all of the reads now matched to a position in the reference? What is the minimum value [Z] that allows all reads to be aligned to at least one position in the reference? Should different hamming distance threshold have an impact on the number of alignments? Discuss in the tutorial, and write a program to show the distribution.