CMSC828H: Computational Gene Finding and Genome Assembly
Lab Assignment #1

This lab requires you to implement simplified versions of a few of the steps of a shotgun assembly pipeline. There are 3 required parts to the assignment, plus an optional extra-credit part.

Due: see syllabus

Part 1: Overlaps

Write a program to find all 40 bp or longer exact overlaps among an input set of reads. Input will be a multi-fasta format file containing the DNA sequences of 300 reads. For simplicity, all reads are error-free and exactly 500 bp long. Each read has a unique ID, given on a header line beginning with the '>' character. The sequence for the read is on the following lines that do not begin with a '>'.

Output should be a list of overlaps, one per line, with 4 columns:

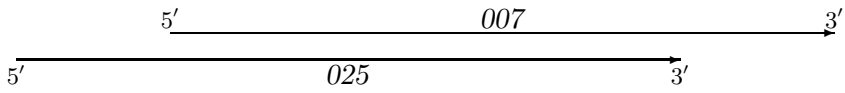
ReadID1 *ReadID2* *Orientation* *Offset*

The first two columns are the ID's of the overlapping reads, with *ReadID1* < *ReadID2*. The *Orientation* value is a single letter, either '**F**' (forward) or '**R**' (reverse), indicating the direction of *ReadID2* in the overlap. We assume that each overlap is normalized so that *ReadID1* is in the forward direction. Finally, *Offset* gives the (signed) number of bases that the left end of *ReadID2* is shifted from the left end of *ReadID1* to make the overlap alignment.

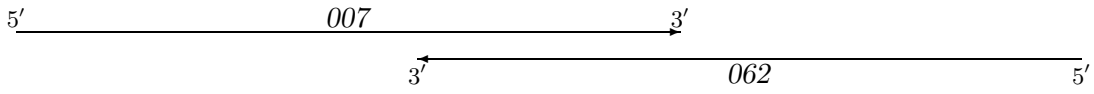
For example, if the following two lines were in the output:

```
007   025   F   -116
007   062   R    302
```

the first line would indicate an overlap of $500 - 116 = 384$ bp between reads *007* and *025* oriented like this



where each read "sticks out" 116 bp past the end of the other. The second line would indicate an overlap of $500 - 302 = 198$ bp between read *007* and the reverse complement of read *062* oriented like this



with each read sticking out by 302 bp.
What to turn in:

- A file (or files) with the source code for your program.
- A file with the output your program produces when run using file lab01.fasta (available on the course web site) as input. Name your output file lab01.olaps and have it sorted first by *ReadID1*, then by *ReadID2*. Make sure there are no extraneous lines or values in the output file. There should be one overlap on each line with exactly 4 fields—no header lines, no blank lines, no comments—just the overlaps.

Notes:

Your code will be graded only for correctness. In particular, since the reads are error free, you don't have to use the dynamic programming matrix (although you may if you'd like). Simple loops and substring comparisons will suffice, although you may want to use k-mer counting to speedup the overlapping.

- When reversing a string you must complement it too, i.e., swap A $\leftrightarrow$ T and G $\leftrightarrow$ C.
- You can assume there is at most one overlap per pair of reads. (In general this need not be true.)
- Remember: only exact overlaps of 40 or more letters should be output.

Part 2: Unitig Layouts

Write a program to read a list of overlaps and determine the set of unitigs they imply. Compute unitigs using either the "classic" graph reduction algorithm or the "best-buddy" algorithm, although you will get slightly different results depending on which algorithm you use. Input to your program is a

list of overlaps in the above-described format. Output is a list of unitigs, each one with a list of the reads it contains and the relative positions of those reads—this is called a "layout".

The format of the output should be a header line for each unitig, followed by the information for each read in the unitig. The header line should look like this:

UNI *UnitigID* *#Reads* *BpLength*

UnitigID should be a 2-digit number, starting with 01, that identifies the unitig. *#Reads* is the number of reads in the unitig (and hence the number of following read-lines for the unitig in the file). *BpLength* is the length of the unitig in DNA bases.

Each read in the unitig should have a line with the following information:

ReadID *Direction* *Offset*

Direction is either '**F**' (forward) or '**R**' (reverse) and indicates the orientation of the read in the layout. *Offset* is the number of bases from the left end of the preceding read to the left end of this read. The reads in each unitig should be listed in the order they would be encountered in a left-to-right scan of the unitig. Thus all offsets will be non-negative. The offset for the first read in the unitig should automatically be zero.

What to turn in:

1. A file (or files) with the source code for your program. Make sure to note if you implemented best buddy unitigging or classical unitigging.
2. A file with the output your program produces when run with the Part 1 overlaps as input. Name your output file lab01.unis. As before, make sure there are no extraneous lines or values in the output file.

Notes:

- Classical unitigging will give better results. Best-Buddy unitigging was created as a size and speedup heuristic, and is not necessary for a problem of this size.
- Every read should be in exactly one unitig. You may assume every read has at least one overlap. It is possible, however, for a unitig to consist of a single read, even if that read has overlaps on both ends.
- The order of unitigs in the file doesn't matter.
- Each unitig has two possible orientations because it represents a DNA sequence that has a reverse-complement. It doesn't matter which of these two orientations you produce.

Part 3: Mate-Pair Assembly

The sequences in lab01.fasta generate a single contig containing repeats. Because of these repeats, it is impossible to determine the correct assembly of all the reads using just their sequences. For this part, you will use mate-pair information to reconstruct the DNA sequence of the contig from which the reads were obtained.

All pairs came from a single 3000 bp library, meaning that the distances between the beginnings of the reads (the $5'$ ends) average 3000 bp. There is a standard deviation of 200 bp in the library size, and we allow up to 3 standard deviations in the final placement of reads. Thus any distance in the range 2400–3600 is permissible.

The mates have consecutive IDs. Thus 001 & 002 are mates, 003 & 004 are mates, etc.

IMPORTANT: Mates always "face" each other, with the $5'$ ends being farthest apart. Thus you measure the distance from $5'$ end to $5'$ end, and the orientation looks like

$$\underset{500\text{ bp}}{5'\xrightarrow{\quad A \quad}3'} \quad \cdots \text{ 1400–2600 bp separating gap } \cdots \quad \underset{500\text{ bp}}{3'\xleftarrow{\quad B \quad}5'}$$

where reads A and B are mates. In file lab01.fasta, all sequences are given in $5'$ to $3'$ orientation.

Using the mate-pair data, determine the correct sequence of the contig from which the reads were sampled.

What to turn in:

1. Just a fasta-format file containing the correct sequence. Use "`>Contig`" as the fasta-header line. Name the file lab01.contig.

Notes:

- You can determine the sequence anyway you like (except asking a classmate for his/her answer). You can write programs, draw pictures, or use a combination of the two.

  One approach would be to construct a sequence for each unitig; use your Part 1 program to find their overlaps; then use the mate pairs to determine a layout for the unitigs.

- As before, because this is a DNA sequence with a reverse complement, there are two correct answers. Either one is acceptable.

- One way to approach this is to use mate pairs to infer positions of unitigs with respect to one another. If mated reads A and B are in

unitigs 01 and 02, respectively, the positions and directions of those reads in their unitigs imply an orientation and range of separations for the unitigs. Multiple mate pairs between the same unitigs should give multiple separation ranges, and the true separation should be in the intersection of all those ranges, near the mean of the separations.

- Watch out for repeat unitigs. They represent sequence from multiple places in the genome, and hence their mate pairs will indicate that they belong in multiple places in the answer.

Part 4 (Extra Credit—Optional): Place All Reads

Use the mate pair information to determine the correct position of each read in the assembled contig sequence. This shouldn't be too hard for reads in unique unitigs, but for reads in repeat unitigs, you'll have to use the mate information to determine in which repeat copy each read belongs.

Output for this should be a list of the positions of each read, one read per line, with the following information

ReadID 5′*Position* 3′*Position*

where the *Position* values are the locations of the end bases of the reads on the assembled contig. Contig positions are numbered starting at 1.

What to turn in:
Since this is optional extra credit you needn't turn in anything. If you choose to do it, turn in the resulting position file and name it `lab01.extra`.

Sample Files

To clarify the concepts and file formats, a small sample input file and all its corresponding output files are available on the course web site. The files are named *sample.**. This sample represents ten 250 bp reads taken from a 1500 bp contig, where mate pair sizes are 950 . . . 1000 bp. The contig has a 350 bp inverted repeat on its ends.