Name(s): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The file contigs\_to\_assemble.fasta contains several genomic DNA contigs already assembled from Illumina sequencing reads from a bacterial organism isolated in the lab. We would now like to combine the contigs. We don’t know the organism so we need a *de novo* assembler. There are many tools available for assembly, but CAP3 is a nice web-based version for relatively short contigs like those we have here. Use CAP3 to assemble the multiple contigs in to one supercontig. Then predict the ORFs in the sequence using a web-based gene finding application (see today’s slides for some options). Finally, predict the functionality of the ORFs (i.e. what proteins do they likely encode?)

1. Include the following files in your submission:

a) **(0.2pts)** Your assembled contig, named supercontig.fasta

b) **(0.2pts)** Your ORFs as amino acid sequences, named orfs.fasta

c) **(0.4pts)** Excel document (AssemblyReport.xlsx template included on Sakai) ***or*** text file (AssemblyReport.txt) listing the ORF name, best hit protein accession number, best hit E-value, and best hit description.

**Note:** You may use BLAST in your web browser like we have done in class and copy/paste the requested results into the Excel file or, **for more extra credit**, you can **use Biopython** to automate BLAST runs, see <http://biopython.org/DIST/docs/tutorial/Tutorial.html#htoc96>

d) **(1pt)** If you used Biopython to run BLAST, also upload your Python code used to retrieve the BLAST results, named blastcode.py. Your code should take your orfs.fasta file as input and write a tab-delimited text file AssemblyReport.txt as output with a header row labeled like the Excel file. Include comments in your code that explain how it works to a beginner. We will test that your Python code works for full credit.

The following questions should be answered in this document. Cite references.

2. **(0.4pts)** Briefly (2-3 sentences) describe the process of assembling the sequences (what did you do and what does the algorithm you used do?).

3. **(0.4pts)** Briefly (2-3 sentences) describe how you predicted and retrieved the ORF sequences (what did you do and what does the algorithm you used do?).

4. **(0.4pts)** Briefly (2-3 sentences) describe how you predicted the functionality of the ORFs. (If you do not explicitly list what method was used, you will not get any credit, i.e. BLAST does not suffice; what kind of BLAST and why?) **Include in your description what organism the sequences are most likely from.**