Lab 2: Multiple Sequence Alignment

Due by 5pm, **Friday** February 17, 2017

This lab does not require coding. Your solution is due outside my mailbox by 5pm on the due date OR can be submitted in your git repository as `solution.pdf` *by midnight. I have provided* `solution.tex` *as a starting point, but you are not required to use it.*

The goals of this week's lab:

- Obtain experience using an popular implementation of multiple sequence alignment tools (ClustalW)
- Connect biology to algorithmic theory by interpreting algorithm parameters and results
- Obtain experience generating and evaluating data visualizations
- Analyzing research literature for methods that improve upon the base algorithms covered in class
- Practice using algorithms from class to construct multiple sequence alignments

To accomplish these goals, the lab is broken into three parts:

1. A tutorial on using **ClustalX** (a GUI version of ClustalW).
2. An analysis and short 1-page response to a research paper on the multiple sequence alignment algorithm, MUSCLE.
3. A problem set to practice and analyze algorithms we have discussed in class.

ClustalX

In order to understand how MSA algorithms are used by biologists, r ead and follow the tutorial for ClustalX. I have placed two sets of sequences you can use to test out ClustalX in your labs directory: `DnaB.txt` which is a small set of medium-length protein sequences, and `sequences12.txt` - a large set of myosin heavy chain proteins. You will run ClustalX by entering the following command on a CS machine (window forwarding is required):

```
$ /usr/swat/bin/clustalx-2.1/clustalx
```

Note: you will probably want to increase the font size after starting the program so the sequences/results are easier to read.

You can begin by using the quick start guide on your `DnaB.txt` file. Then, read the rest of the document testing out different parameters to see how they alter your results.

Once you complete the document and analyze both sequences, answer the following questions:

1. Briefly describe the trade-offs between the **Fast** and **Slow** approaches. Which option is suitable for your `DnaB.txt` example? How about the `sequences12.txt` file? Why might a biologist combine both options in their strategy; that is, attempt fast run(s) followed by a slow run?

2. ClustalW is a heuristic and thus cannot guarantee the optimal solution after being run. Read the tutorial section *Assessing the quality of alignment* for a discussion on this topic. How do the author's suggest a) assessing the quality of an alignment and b) finding the "best" alignment?

3. ClustalW color codes the results to help the user visualize the results (go to *Help* → *Colors* on the menu for an explanation). How does the color coding aid you in analysis? Specifically, think of a column which has varied content (i.e., the characters differ quite a bit) yet consistently stays the same color. How might this inform your assessment of the "conservation" of a column?

4. After running ClustalX on one of the input sets, head on over to the WebLogo page for producing representations of the entropy of a sequence. Load the alignment (an `.aln` file) from one of your examples and choose a range of approximately 25 amino acids (using the *Logo Range* option on the web form). Your range should include examples of both highly conserved and highly variable locations. Save the output file and either included it in your written solutions or add it as a file named `logo.png` in this week's lab directory. Be sure your image saved properly (some browsers seem to not work properly) by opening the image outside of a browser environment (e.g., Preview on Mac or xv on Linux).

Problem Set

1. You are given the following sequences:

   - `TGTAC`
   - `TGTTAAC`
   - `ATGTC`
   - `ATGTGGC`
   - `TGTAAC`

   Use the Star algorithm to identify the multiple sequence alignment for the given sequences. To pick the center, calculate all possible pairwise alignment scores and pick the sequence with the highest average. Use the global alignment algorithm with a linear gap penalty of -3 and substition score of 1 for matches and -1 for mismatches. To calculate pairwise global alignments, you can use the tool at http://www.bioinformatics.org/sms2/pairwise_align_dna.html. Set all three gap types to -3. Show your work if done by hand, or submit your code as `star.py`. At a minimum, you should show the average alignment scores for each sequence (for picking the center) as well as the successive mergers that lead to your final multiple sequence alignment.

2. In class, we discussed two scoring metrics: sum of pairs (SP) and minimum entropy. Consider a column m_i that contains the following characters: `A,A,A,A` and another that contains `A,A,A,D`. Using the BLOSUM62 matrix, compute the sum of pairs and minimum entropy scores for each column. Note that I have given columns and not sequences. For reference, $s(A, A) = 4$ and $s(A, D) = -2$.

3. Do the same for column `A,A,A,A,A` versus column `A,A,A,A,D`. Compare your results to the previous problem. How does this support our discussion on the relative advantages and disadvantages to each scoring metric.

4. *This is an advanced question. I want to challenge you to push beyond the class material, but I don't want you to stress for hours about one question. Think about the problem carefully, but don't worry if you can't solve it completely.*

   In class, we discussed DNA as a linear sequence of nucleotides. Not all DNA exists in this double helix form, however. The genomes of many bacteria and even some eukaryotes (e.g., mitochondrial DNA in humans) exist in circular form. That is, there is no "start" or "end" to the sequence. Devise an efficient algorithm to perform optimal global alignment on two circular DNA sequences. A $O(nm)$ algorithm exists for two circular sequences of length n and m. Even if you do not arrive at such a solution, describe your best attempt as well as your big-O run time. As a hint, your solution should build off of our existing global alignment algorithms from class.

Response Paper

As noted in class, ClustalW is one of the most widely used technique, but is not necessarily the most advanced method available. Another popular program is MUSCLE. Read the brief introduction of the paper, available here and on the course syllabus. Discuss the paper with your classmates (at least one other person) and reflect on the following aspects:

- How does the method relate to our discussion in class, in terms of the type of multiple alignment approaches?
- What problem is the paper addressing, other than the general problem of multiple sequence alignment? Is there a specific set of criticisms of available methods that authors aim to solve?
- What similarities are there between MUSCLE and other methods discussed in class?
- What new algorithmic approaches does the paper introduce that we have not seen before?
- In what ways do the authors evaluate their method? Is there a particular type of data set? What other methods do they consider? Is their assessment "fair"? What metrics do they consider?

With your homework, submit a short paper on one interesting aspect of the paper. The paper should be short - **less than 1 page single-spaced**. Rather than addressing all of the questions above, your paper should provide a **detailed analysis on a specific aspect of the paper**. This should stem from your discussion above (e.g., the use of distance functions; the relationship to phylogenetic trees; the choice of statistical methods for evaluation, etc.) or can explore some other aspect you find interesting. Your reaction should:

(a) describe the aspect of the MUSCLE paper you are discussing

(b) provide analysis of your choosing (e.g., compare and contrast to in-class methods; explain the relationship between the biology and the algorithm, etc.)

(c) pose questions of further inquiry that the paper did not cover.

The provided paper is a shortened version of the full-length paper. You should not need to go so far in detail on the methods to require reading the long version, but are free to do so to satisfy intellectually curiosity.