COMP 561 Final project – Fall 2025

Due date: Dec. 3rd (but extensions could be granted until Dec. 12th)

Project can be done in teams of up to 3, or individually.

What to submit: A 5-6 page report (8-10 pages if in team of 3) in the form of a research paper, that includes:

1. Introduction: problem definition and motivation
2. Methodology: description of the algorithm
3. Results: Evaluation of the performance (running time, accuracy, etc. as relevant depending on the project)
4. Discussion and future work: What worked, what didn't? How could it be improved?
5. Bibliography

Evaluation: Your project will be evaluated based only on your report

1. Originality/quality of approach 50%
2. Quality of your assessment of the performance 30%
3. Overall presentation of the report 20%

Note about item (2) above: What will determine your grade here is not the performance of the approach itself, but the quality of the work done to assess it and the interpretation of the results you are getting. If the method you developed produces poor results but was potentially interesting and you've done a good job at evaluating it and understanding why it failed, you will still get a good grade.

Project suggestion #1: Alignment of PacBio reads

Pacific BioScience is the manufacturer of a DNA sequencing machine with unique capabilities: it produces reads that are very long (5,000-20,000bp), but also that have a very high sequencing error rate. Having long reads is a great help to sequence and assemble genomes, but the high error rate is a major obstacle.

The key operation that one needs to be able to perform assembly based on the overlap-layout-consensus approach is the following:
Given: a set of reads R1…Rn (where n is of the order of 1,000,000), where each read Ri originates from a particular portion $G_i=[s_i,e_i]$ of the (unknown) genome being sequence (s_i and e_i are the starting and ending positions of the genomic region that resulted in read Ri.) .
Find: All pairs of reads (Ri, Rj) that likely originate from overlapping regions of the genome, i.e. where [si,ei] overlaps substantially [sj,ej]. However, since we don't know the sequence of the genome being sequenced, we can't know what genomic region each read originates from. The only thing we know is that if two reads originate from overlapping regions, then some suffix of Ri should have high

similarity to some prefix of Rj. If there were no sequencing errors, then the match would be perfect, but because of sequencing errors in both reads, they will not match perfectly, and may actually be quite different in their sequence.

A possible approach (which doesn't work very well) is to repeatedly use Blast, where each read is used in turn as query sequence, and where the database is the entire set of reads. Any high-scoring alignments would be likely to correspond to pairs of reads that originate from the same portion of the genome. There are two issues with this approach: (1) Because the sequencing error rate is so high, Blast may fail to identify alignments between two reads even if they originate from the same region. (2) It is quite slow (weeks of computation) for a complete data set.

The goal of the project is to design, implement, and test algorithms to improve on one or both of these aspects (sensitivity and running time). You will use PacBio sequencing data from a genome sequencing project on Leishmania, which is a parasite that is an important cause of disease in many parts of the world.

All data is available at: http://www.cs.mcgill.ca/~blanchem/561/pacbio.html

Conveniently, Leishmania is a species whose genome has already been sequenced using other approaches, and its genome is small (33Mb). You can download it from the link above.

Suggestions:
- You can use Blast to identify what genomic region each read originates from, by aligning each read (query) to the actual genome (database). This is an easier alignment task than aligning reads to reads, because the genome can be considered to be almost error-free. If two reads align to overlapping portions of the genome, they constitute a pair that your alignment method should be able to identify. This will serve as the ground truth that your read-to-read alignment will try to reach.
- To do a good job at this alignment task, you will need to take in consideration the type of sequencing errors the PacBio sequencer makes. You can get a grasp for this by looking at the read-to-genome alignments produced by Blast.

Approach: You are free to explore any approach you may feel has potential. You will not be evaluated on whether or not your approach works or not but rather on the innovation of the approach you are proposing.

Project suggestion #2: Alignment algorithms for probabilistic sequences

Context: In certain situations, we may not know exactly the genome of a species, but instead we could have some uncertainty about the identity of nucleotides at a given position. In that case, the genome of a species may be described using a probability matrix with four rows and L columns, where L is the length of the genome. For example:

		1	**2**	**3**	**4**	**5**	**6**	**7**	**…**
	A	0.01	0.5	0	1	0.2	0.1	0	
Genome =	**C**	0	0.5	0	0	0.3	0.1	0	
	G	0	0	1	0	0.4	0.8	1	
	T	0.99	0	0	0	0.1	0	0	

This probabilistic generalization of a pure sequence can be used when the sequencing data that resulted in the genome assembly is ambiguous at certain positions (e.g. at polymorphic sites). More interestingly (for me), it can be used to represent computationally inferred ancestral sequences, which is something our lab has been working on for a while. Here, we have algorithms to predict what specific ancestral DNA sequences looked like (e.g. the genome of one of the first mammals, that lived 75 Myrs ago), based on the genomes that live today. The output of this method is exactly a probabilistic genome sequence as shown above.

Goal: Many genome analyses require the ability to find alignments between a short query sequence and a long genome. This is what Blast is used for. But Blast doesn't work when the genome (database) sequence is probabilistic. Your goal is develop an analog of Blast that works to align a short (non-probabilistic) query sequence against a long probabilistic genome. You will first have to define what it means to optimally align a sequence against a probabilistic sequence (what are we trying to optimize?), and then you'll need to develop, implement, and test an algorithm to do so.

Data is available at:
http://www.cs.mcgill.ca/~blanchem/561/probabilisticGenome.html

Suggestion:
One potential approach to evaluate your algorithm is to use the probabilistic genome to generate a short query sequence by picking a starting position randomly in the genome, generating the query sequence by selecting nucleotides randomly according to the probabilities at each position, and possibly adding a few additional random substitutions and indels. You can then feed this query sequence to your algorithm and see if it succeeds at identifying the correct portion of the probabilistic genome.

Project suggestion #3: Structural properties of DNA

Although the DNA double helix is usually presented as a very regular structure, it is not perfectly regular. First, there is a major and a minor groove (see https://en.wikipedia.org/wiki/DNA#Grooves). Beyond this, the specific sequence of nucleotides creates subtle variation in the width of each groove and in other structural properties. These properties have a significant impact on DNA's affinity for proteins such as transcription factors. For example, a given DNA sequence may be bound by a certain transcription factor if it has the right shape (which depends on the sequence context to which it belongs), whereas it may not be bound in the wrong context. DNA physical properties can be predicted from the local DNA sequence using tools such as DNAshape:
https://academic.oup.com/nar/article/43/D1/D103/2439587

We propose two projects building on these notions:

Project 3.1: Prediction of transcription factor binding based on DNA physical properties.

Experiments such as Chip-Seq can identify a list of DNA regions bound by a given transcription factor. Combined with a computational scan for the TF's position-weight matrix, this can be used to identify sites that are occupied by the TF in the cell type and conditions where the experiments were made. We can also identify a set of DNA sequences that, based on their sequence, look like they should be bound, but that in reality are not. The goal of the project is to determine whether the sites that are bound (positive examples) can be distinguished from those that are not (negative example) based on the predicted structural properties of the sequence. The project will involve (1) Identifying a set of bound and non-bound DNA sequences for a given TF based on existing experimental data; (2) Calculating the DNA physical properties of each sequence; (3) Training a machine learning classifier to distinguish between bound and unbound sites.

Additional details about tools and data sources will be given in a separate meeting.

You will use for this project many sources of publicly available data:

1) The sequence of the human genome (assembly hg19), available at: http://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz The format is self-explanatory. Lower-case letters correspond to repetitive regions, uppercase to non-repetitive regions. For this project, you should consider both as the same.
2) A set of genomic regions that are active regulatory regions in a particular cell type (GM12878, which is a cell line derived from lymphoblasts). This is available here: http://www.cs.mcgill.ca/~blanchem/561/wgEncodeRegTfbsClusteredV3.GM12878.merged.bed.gz

3) A set of genomic coordinates of actual transcription factor binding sites for several transcription factors:
   http://www.cs.mcgill.ca/~blanchem/561/factorbookMotifPos.txt.gz
   Each line looks like this:
   585 chr1 16245 16260 CTCF 1.97 -

   Field 1: Ignore
   Field 2,3,4: Genomic coordinate
   Field 5: Name of transcription factor
   Field 6: Score of predicted binding site (could probably be ignored)
   Field 7: DNA strand (+ or -) of binding site

4) For a specific transcription factor T, positive examples of binding sites can be extracted from (3). Negative examples should be considered as any sequence that is located in a region from (2) that does not contain a binding site for T listed in (3).
5) Position weight matrices for each TF is available here:
   http://www.cs.mcgill.ca/~blanchem/561/factorbookMotifPwm.txt.gz
6) Predicted structural properties for every human genome positions are available here: http://rohsdb.cmb.usc.edu/