Dun dun *dunnnnnnnn.*
Alright, here are the rules:

- You may collaborate in groups of at most 3.
- If you choose to collaborate, you *must* do your own write-up.
- If you choose to collaborate, you *must* do write your collaborator's names.
- You *cannot* cross talk amongst groups. Just. Don't. Do. It.
- You *cannot* go swimming right after you eat. Sorry. I don't make the rules.
- If you would like to get set up into a group OR if you are willing to add an extra person to your group (help me out here), please fill out this form: `https://forms.gle/JbsivtSDyFEFz2ZH7` Note: if you are adding people to your group, only one person should fill out the form.
- I will assign groups that submit to the form at Monday 10:30AM so please fill out the form before then.

If you find any errors, please email me. I'll have several office hours for clarifying questions in the coming week.

Problem 1

Non-linear dimension reduction

[30 points]

There is some data on BruinLearn in `final/problem1.tsv` you need to load for this problem.

You don't need to know the generating process for the data, but in case it is helpful, here it is. Rows 1 to 100 correspond to the first 'cluster' and rows 101 to 200 correspond the second 'cluster'. The data is generated by:

$$y_i \sim \text{Normal}_2(\mu_{k(i)}, I_2),$$

where $k(i) = 1$ for $i = \{1, 2, \ldots, 100\}$ and $k(i) = 2$ for $i = \{101, 101, \ldots, 200\}$.$\mu_1 = (0, 0)$ and $\mu_2 = (10, 10)$.

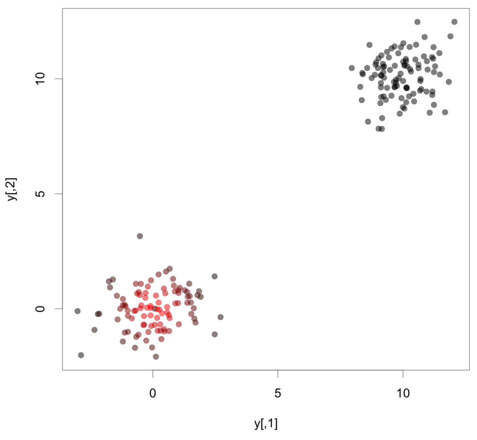
You are going to implement some components of t-SNE to get some intuition about how different components of the algorithm work.

Finally, everything you need to know is in the non-linear dimension reduction slides.

(a) (Slide 11) Implement the $p_{j|i}$ matrix. Do yourself a favor and make it a function because you're going to use it quite a bit. You can make your function take a single shared $\sigma_i^2 = \sigma^2$. Sanity check: each row should sum to 1. Side note: if you decide to do the extra credit (see (k)), you should allow your algorithm to utilize different σ_i^2 otherwise you're gonna have a bad time.

(b) (Slide 11) Implement the p_{ij} matrix. Sanity check: the entire matrix should sum to 1.

(c) Using $\sigma^2 = 1$, plot the entire dataset and color the points based on their probability reference to the first data point (p_{1j}). A reasonable color scale might be: $p_{1j}/\max_j(p_1 j)$. Your plot should show a change of color away from the first data point. Do the same for $\sigma^2 = \{0.1, 10, 100\}$. Each plot might look something like this:



(d) (Slide 13) Implement the q_{ij} matrix. Sanity check: the entire matrix should sum to 1.

(e) Plot the entire dataset and color the points based on their q_{1j} probability relative to the first data point. How is it different than p_{1j}?

(f) (Slide 13) Implement the KL-divergence. Note, the contribution of $\{ij\}$ is zero when $p_{ij} = 0$.

(g) Using the real data as the low-dimensional projection, compute the KL-divergence when:

   i. $\sigma^2 = 0.1$.
   ii. $\sigma^2 = 1$.
   iii. $\sigma^2 = 100$.

   Any thoughts on what might be happening?

(h) Using $\sigma^2 = 1$, can you find a projection that reduces the KL-divergence? Note, there are plenty linear or non-linear ones. The easiest might be to do might be 'move' one cluster. Show the projection and the KL-divergence.

(i) Take the first two principal components and use them as your projection. How does the KL-divergence change from when you used the real data and why? No, you don't need to implement PCA.

(j) Summarize your thoughts on how these hyper-parameters matter.

(k) Extra credit (2 points): implement the Perplexity and recompute the KL-divergence from the previous projections you made using the σ_i^2 you get for Perplexity $\{5, 25, 50, 100\}$. To implement the Perplexity, find the value of σ_i^2 that approximately satisfies the equation Perplexity$(P_i) = 2^{H(P_i)}$ where P_i is the i-th row in the $p_{j|i}$ matrix and $H(P_i)$ is the Shannon entropy. Plot a histogram of your σ_i^2 values and how did your results change?

Problem 2

BWT

[15 points]

Consider the string:

$$banamabananas$$

(lol).

(a) Construct the cyclic rotations.

(b) Construct the M(*Text*) matrix and report the BWT.

(c) Align *babama* allowing one mismatch. Show the steps like in the book (Figure 9.19).

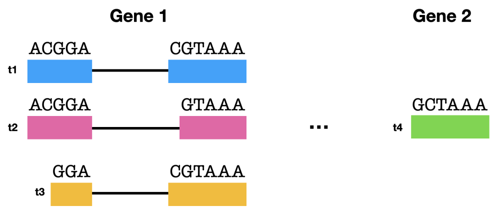
Problem 3

Pseudoalignment

[30 points]
Consider the following transcriptome:

```
>t1_g1
ACGGACGTAAA
>t2_g1
ACGGAGTAA
>t3_g1
GGACGTAAA
>t4_g2
GCTAAA
```

Here is a visual representation of the transcriptome:



(a) Draw the colored de Bruijn graph that one would use for pseudoalignment with $k = 3$. In this particular case, the nodes have 3-mers, not the edges (this is the setup we used in class). Refer to the slides on construction.

(b) Pseudoalign *GGACGT* and show the relevant steps you might take (including skips). Refer to the slides for the pseudoalignment.

(c) Pseudoalign *GGATGT* and show the relevant steps you might take (including skips). Remember, every k-mer has an equivalence class which is a set. If a k-mer is in your data but not in your transcriptome, its equivalence class is the null set.

(d) The previous read might arise if there is an error at position 4 (using 1-based indexing). Describe an algorithm to deal with the error. Describe the benefits and drawbacks of your algorithm. These properties might come in speed, loss of data, or false alignments (or other things). Note: there isn't one correct answer.

(e) The following sequence is a valid RNA-seq read *TTTACG*. Clearly it won't give you a non-empty equivalence class as-is. How did this data arise and how might you pseudoalign it? Hint: think about how RNA-seq is generated. That is, the orientation of the reads matters and can generate some annoying things we have to keep track of.

(f) Imagine you knew the haplotypes of this person and instead of an error coming from position 4, there is a single nucleotide polymorphism (SNP). Basically, one base is altered on one strand of DNA and on the other strand of DNA it is as the original reference. This means that in practice you will see reads that look like (b) and (c) in approximately equal proportion (ehh, there are caveats, but let's pretend). How might you change your colored de Bruijn graph to deal with this case?

Problem 4

More errosr, this time with de Bruijn graphs for assembly

[20 points]

(a) Consider the following sequence:

$$ATTCGGCGATTT$$

Draw the de Bruijn graph using $k = 3$.

(b) We have previously only built graphs assuming a perfect sampling of the composition. Now, build the graph from the sequence:

$$AATCGGCGATTT$$

and merge it with the graph from (a). You can build two different graphs and merge them, or build the graph using the merged composition from both genomes.

(c) Is there an Eulerian path?

(d) Is the original string a subgraph of the graph in (b)?

(e) Let's build one more annoying graph. Add another instantiation of the graph from (a) to the graph from (b). Sanity check: there should be three edges going from CG to GA. Did this help our cause of reconstructing the original sequence?

(f) What makes assembly hard if you begin considering errors in your sampling?

Problem 5

Basically free points

[2 points]
Fill out the end-of-class survey here when you are *done* with the rest of the exam:

https://forms.gle/9aEcN4WtBYmwfdrn8

Please attach a screenshot of the submission.

Problem 6

Human Genetics extra credit

[3 points]
If you did the human genetics extra credit, it will count here.

Thanks!

Hope you found the class helpful, fun, and interesting. Enjoy your break :)