Below are four possible projects. Please fill out the GradeScope assignment declaring your project by the end of February 18th. The rules are: don't cheat, anything else is fine. Implement in whatever language you choose. Bonus points for some weird language (Smalltalk?).

Your code should work, and please provide everything necessary to run it.

For your report, write up what you did, any potential problems you had, and anything you found you found interesting/cool about implementation.

Tips: start early, ask questions, and talk to each other.

Pseudoalignment implementation

Given some RNA-seq data in FASTA format, find the vector of equivalence class counts. Your implementation should in the least do the following:

- Given a gene annotation, an index should be produced which can be read in by your pseudoalignment procedure.
- Your pseudoalignment procedure should take in (1) the previously mentioned index, (2) RNA-seq data and output equivalence class counts, and (3) a k-mer length.

Please provide equivalence class counts in the following format:

counts	number of items in equivalence class	isoforms in equivalence class
30	0	NA
5	1	ENST000003679
10379	2	ENST000003679,ENST000009216

You may implement either the naive version presented in lecture using the hash table or the skipping (colored de Bruijn graph) version[1]. If you are feeling extra weird, you can also implement it using a suffix array.

Finally, provide some basic statistics about the size of the equivalence classes and how much data is mapping to them. This sort of summary can be provided in a figure/plot.

RNA-seq Expectation Maximization implementation

Warning: the most difficult of all.

Given some RNA-seq alignments in BAM format and a isoform annotation table, find the relevant equivalence classes, then estimate relative abundance of the isoforms.

You can learn about the BAM format here. You can use whatever library you choose to parse the reads. In python, PySam is pretty good and in R Rsamtools is decent. If you are using C/C++ there are official libraries as well (the actual BAM format is in C). If you are a masochist, there is probably a Java implementation floating around.

You may implement the RSEM version[3] which has a likelihood in read-space or the IsoEM/kallisto[1] version which is in equivalence class space. The kallisto version is a bit more work because it will require computing equivalence classes from the reads, whereas the

RSEM version can be done directly from the reads. The kallisto version will be drastically faster than the RSEM version because of the reduced complexity of the likelihood function.

Please provide the transcript abundance table outputting the following formant:

transcript	length	effective_length	expected_counts	tpm
ENST000003679	1219	1019	517	2.93

The reads provided are single-end reads, so you will have to assume a fragment length distribution. Treat it as $N(200, 10)$ (with appropriate discretization/normalizations).

In your report you might plot some metric against the various iterations of the EM. Take your pick.

Extra credit: output the final posterior probability of each read mapping to a particular transcript.

Clustering

Implement k-means clustering. The implementation should take an arbitrary matrix with "genes" on the rows and samples on the columns.

Download the DeRisi[2] data from from the projects folder. Cluster the data using various values for k and inspect the following: (1) how different the cluster means look, (2) how the outliers look in the different clusters.

The data has missing values. Use your best judgment on how to deal with them. You can obviously use heuristics, but be clear about what you are doing.

After doing the analysis with k-means, redo the analysis with a different clustering method. No need to implement your own clustering method, you can use any of the open source packages. Pick whatever you think sounds cool.

Compare the results you get from k-means with whatever else you decided to use. Is one subjectively better than the other? Feel free to look at the paper for inspiration.

Your own project

Please propose your own project in an email with at least a paragraph long description. I reserve the right to reject your proposal or amended it, but if you submit it earlier, you can have a chance to revise it.

References

[1] Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. Near-optimal probabilistic rna-seq quantification. *Nature biotechnology 34*, 5 (2016), 525–527.

[2] DeRisi, J. L., Iyer, V. R., and Brown, P. O. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science 278*, 5338 (1997), 680–686.

[3] Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., and Dewey, C. N. Rna-seq gene expression estimation with read mapping uncertainty. *Bioinformatics 26*, 4 (2010), 493–500.