Hands-on exercise #1: RNA-seq alignment

The goal of these practice problems is to align RNA-seq data against the human transcriptome and compare the resulting abundance estimates. You'll:

1. Use the bowtie2 aligner on an RNA-seq sample from SARS-CoV-2 infected cells.
2. Use the kallisto aligner on the same data.
3. Compare the abundance estimates for kallisto and bowtie2 in python.

The questions you'll answer for your assignment are in these green boxes. Submit a written response plus your jupyter notebook saved as a pdf. No need to write long answers - if I ask for a number, you can just tell me the number.

We will use the following tools:

- Bowtie2: an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences, http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
- Samtools: a suite of programs for interacting with high-throughput sequencing data, http://www.htslib.org/
- kallisto: a pseudo-alignment program, https://pachterlab.github.io/kallisto/about
- python & jupyter notebook

1. Connect to the datahub and install some software.

Go to https://biology.datahub.berkeley.edu/user/[your calnet username]/lab and log in. Open a terminal (the bottom row of icons).

Let's play around with unix for a few minutes.

1. Where are you? (What directory are you in?)
2. Who are you? (What is your username?) The command is `whoami`
3. Do we have bowtie2? Type `which bowtie`
4. Do we have kallisto?
5. What is a path? Type `echo $PATH`

Now install kallisto: `conda install -c bioconda kallisto`

2. Download the data and set up your directories

Download the sequencing data:

```
% wget http://storage.ingolia-lab.org/lareaulab/BioE290/data.tar.gz
```

.gz means it is compressed with "gzip" and .tar means a whole directory structure has been saved as a single file. To extract it:

```
% tar -xzf data.tar.gz
```

Usually the first step of the analysis is to create an *index* for the transcriptome. ***This is slow so I already did it for you.*** Download and extract the bowtie 2 index that I already built:

```
% wget http://storage.ingolia-lab.org/lareaulab/BioE290/index.tar.gz
% tar -xzf index.tar.gz
```

Then download the kallisto index:

```
% cd Index
% wget https://github.com/pachterlab/kallisto-transcriptome-indices/releases/download/
ensembl-96/homo_sapiens.tar.gz
% tar -xzf homo_sapiens.tar.gz
```

Finally, make a directory called **Alignments** with subdirectories **bowtie2** and **kallisto**.

List your directory and see where everything is. Note how extracting a .tar.gz file creates a directory (because it matches the original structure that I packaged up).

The actual indexes are these:

bowtie:

```
Index/bowtie2/grch38_transcriptome
```

kallisto:

```
Index/homo_sapiens/transcriptome.idx
```

3. Aligning data to the transcriptome

The reads to align are in your Data directory.

```
Data/Series1_NHBE_SARS-CoV-2_1/*
```

We're going to use two methods to align them. I want to emphasize that the bowtie2 + samtools workflow here is not a standard approach. We're doing it because it shows a very naive way to count things. The kallisto approach is what I would recommend for real use. If you used bowtie2 in real life, you would combine it with other programs to do the analysis.

a. bowtie2 alignment:

```
bowtie2 -x [index] \
     -U [reads, comma separated if multiple files] \
     -S [output]
```

You can paste long, multi-line commands directly onto the server, because the backslashes at the end of each line indicate that the command will continue onto the next line.

There are three different arguments given to the bowtie2 aligner here:

- -x path/to/genome-index specifies the filename of the genome index

- -U path/to/reads.fastq specifies the filename of the sequencing reads to be aligned, which can be gzip-compressed. They are separated by commas, ***with no spaces***, if there's more than one.
- -S path/to/output.sam specifies the filename of the SAM-format output file. I suggest using:

  `Alignments/bowtie2/sc2_1.transcriptome.sam`

Fill in the appropriate commands to align the four fastq files from this directory:

`/mnt/lareaulab/lareau/BioE290/Data/Series1_NHBE_SARS-CoV-2_1/`

Align them all in one bowtie2 command. At the end, it will spit out some statistics about the alignments, which you should copy and paste somewhere so you can use them to answer these questions:

Assignment questions:

- What is your bowtie2 commandline?
- How many reads aligned uniquely?
- How many reads aligned at all?

b. Sorting, indexing, and counting the bowtie2 alignments

Now we're going to count up how many reads hit each transcript. Please note that the bowtie2 approach we're using is ***not*** what you would do in a real analysis – I'm setting it up this way to contrast it with the kallisto results.

First, you need to sort your SAM alignments into a consistent order along the transcripts, and you'll save that as a compressed BAM file to take up less space.

```
samtools sort -o Alignments/bowtie2/sc2_1.sorted.bam \
   Alignments/bowtie2/sc2_1.transcriptome.sam
```

Then you use sam tools index to let it count them faster.

```
samtools index Alignments/bowtie2/sc2_1.sorted.bam
```

Finally, count up how many reads there are per transcript!

```
samtools idxstats Alignments/bowtie2/sc2_1.sorted.bam > \
   Alignments/bowtie2/sc2_1.idxstats
```

Assignment question:

- How many lines are in the sc2_1.idxstats file? Use the `wc -l` command.
- What does each line represent? How many genes are in the human genome, approximately? What does the difference between those two numbers tell us, and how is that relevant to the various alignment and quantification tools we discussed?

Tip for future use: BAM files are compressed, but you can view them with samtools:

```
samtools view yourfile.bam | less -S
```

This uses the view subcommand in samtools to read in BAM-format alignments (the default) and display the text, SAM-format alignments (also the default), and then "pipes" this output to less rather than printing millions of alignments to your terminal. As always, you can quit less with q.

c. kallisto alignment:

Now on to kallisto - you'll see how much faster it is! Easy to use, too!

Set up a command using the `kallisto quant` (quantify) program. This will put the alignments into the kallisto subdirectory you already created in your Alignments directory.

```
kallisto quant -i [index] -o Alignments/kallisto \
   -single -l 200 -s 20 \
   [fastq filenames, separated by spaces]
```

Include the four fastq files from the `Series1_NHBE_SARS-CoV-2_1` data directory .

*Unix tip: you can specify all the files in a directory with `*`, all the fastq files with `*.fastq`, etc. The * is a wildcard that can match any string. The command line can interpret that output as a list separated by spaces. So, what would happen if you put `/mnt/lareaulab/lareau/BioE290/Data/Series1_NHBE_SARS-CoV-2_1/*` in the kallisto command? Why didn't this work for bowtie2?*

4. How similar are bowtie2 and kallisto?

Assignment question:

- How many lines are in the kallisto output file, `abundance.tsv`?
- What do the -l 200 and -s 20 indicate? I guessed 200 as the number, but what if I was wrong and it should be more like 400? You can find a description in the kallisto manual https://pachterlab.github.io/kallisto/manual

Next, we're going to compare the count per transcript between the bowtie2 alignment and the kallisto quantification.

This portion of the assignment uses a jupyter notebook. For this first assignment, I've written out most of what to do, with blanks to fill in. This will let you get up to speed on python if you're new to it. Coding is not the main goal of this class, but it's essential for looking at your data.

This link will copy the assignment1.ipynb file to your datahub account and open it.

https://biology.datahub.berkeley.edu/hub/user-redirect/git-pull?repo=https%3A%2F%2Fgithub.com%2Flareaulab%2Fbioe_190_290_2021&urlpath=lab%2Ftree%2Fbioe_190_290_2021%2Fassignment1.ipynb&branch=main

Assignment:

- The goal is **a scatterplot of the count per transcript in the bowtie2 analysis** (third column of the idxstats output) **vs kallisto** (est_counts column), plotted on a log scale and with labeled axes. Ideally, your plot will show density, such as hexbin, so you can see how many points are plotted over each other.
- The jupyter notebook will show you how to do this if you're not familiar.

Assignment:

- Now that you've made your plot, what does this show you about the similarity of estimates?
- From what we've discussed in class, what might contribute to the differences? The difference is not due to pseudoalignment vs alignment. Rather, it has to do with what else kallisto does beyond that. What other tools that we discussed might be a good second step after bowtie alignment, to get closer to the kallisto output?