Notes on compiling your own Bowtie2 indices

Getting reference genome assemblies and working with them on Quest.

To use bowtie2 or hisat2, a prerequisite is that you have a precompiled genome assembly for your model organism (or yourself, if you study humans and you are one). To do this, we need to download the official reference genome sequence, and we need to use a utility within bowtie2 to compile the reference genome index.

There are numerous places to get a reference genome sequence. My preferred location is UCSC, and we will use that for this demonstration.

If you know the official name of your genome assembly (hg38, mm10, dm6, ce11), you can skip to the code. If you don't, or you want to double check, please navigate to the UCSC genome sequence and downloads page at

https://hgdownload.soe.ucsc.edu/downloads.html

Here, you can browse for the appropriate file by following the links. But the important thing is to get the genome serial number (e.g., hg38) that you are looking for.

What we ultimately want to get is a filename that is .fa.gz

This is a fasta-formatted file that has been gzipped. You do not have to download it like you would a typical internet thing.

Instead, for example, for the dm6 version of the *Drosophila* genome, log in to Quest and navigate to where you want to download it (perhaps create a directory called "genomes" or something). Then enter:

```
wget --timestamping 'ftp://hgdownload.cse.ucsc.edu/goldenPath/dm6/bigZips/dm6.fa.gz'
```

After some activity, you should now have a copy of the reference genome in your directory. *Note, for a different genome, there are two places in that file path that you have to replace "dm6" with your favorite genome's short name.*

Make the index:

To make the index, you want to gather some resources and start an interactive session on a Quest node:

```
srun -A b1042 --partition=genomicsguestA -N 1 -n 24 --mem=64G --time=12:00:00 --pty bash -i
```

Now, you want to navigate to a parental directory where you want to place your indices, and make a new directory for them. I prefer to have a top directory named `Bowtie_Indices`, and then to have a separate directory for each critter inside.

From the parental directory:

```
mkdir Bowtie_Indices          ##(of course, only if this doesn't already exist)
cd Bowtie_Indices
mkdir <your reference genome codename (e.g., hg38)>
bowtie2-build --threads 16 <path to the .fa.gz file you just downloaded> <genome codename>
```

The call to `bowtie2-build` above has three inputs/options. `--threads 16` specifies 16 cores will be used. For big genomes (not worm/fly), this step will take a long time, and multithreading can reduce it by ~25-50%, but at some point going about 16 threads doesn't improve matters much. The second input is the path to the file you downloaded from UCSC. The last input is the genome codename that will be part of the **six** files that comprise the index.

When mapping, we write the path to the directory we created above plus the *common prefix* for all the six index files (e.g. `projects/sally/Bowtie_Indices/hg38/hg38`).

For the human genome, this can take 40-60 minutes to compile, so be patient. But fortunately, it only has to be done once.

hisat2

A similar procedure can be done for generating indices for the alternative mapper `hisat2`. Please see the manual for this package, but once you have a reference genome in hand, it should be as simple as replacing the hisat2 builder for the bowtie2 builder in the code above (and changing the parental directory name from Bowtie to Hisat)