Mapping Reads to a Reference Genome.

Overview

Mapping refers to the process of aligning short reads to a reference sequence, whether the reference is a complete genome, transcriptome, or *de novo* assembly. There are numerous programs that have been developed to map reads to a reference sequence that vary in their algorithms and therefore speed.

The program that we will use in this class is Bowtie2 (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml). The goal of mapping is to create an alignment file, also known as a Sequence/Alignment Map (SAM) file for each of your samples. This SAM file will contain one line for each of the reads in your sample denoting the reference sequence (genes, contigs, or gene regions) to which it maps, the position in the reference sequence, and a Phred-scaled quality score of the mapping, among other details. You will use the SAM files for your samples to extract subsequent information about your sample. For example: gene expression information (the number of reads that map to each reference sequence), to identify polymorphisms (variations between people) across your data, or where a particular protein binds (chromatin immunoprecipitation).

General Steps:

1. If this is a new genome (where new means you've never worked with it before) then you must first obtain or generate an index to the genome.
2. Use bowtie2 to align reads to the indexed genome. Choose an appropriate alignment strategy for the data you have and questions you intend to ask.
3. Convert the resulting SAM file (human readable but big) into a BAM file (binary, more compacted) for downstream analysis.
4. Sort and index the BAM file for visualization.
5. Sanity check your mapping by visualizing the results in a genome viewer such as IGV.

You may find the manual for bowtie2 useful:
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

Modules necessary:

```
module load bowtie/2.2.9
module load samtools/1.3.1
```

Index a genome

Aligning reads to a genome can be viewed in the general context of approximate string matching. The goal is to find a pattern (the short-read) in a large text or corpus (the genome), allowing for mismatches and indels. Naively, you can scan the text for the pattern but this is inefficient (think of trying to find an address in a phone book where the names were all mixed up). There are techniques to pre-process (or index) the text to make queries fast and also that can even compress the size of the text.

Bowtie REQUIRES an indexed genome. On Fiji, a number of pre-existing indexes already exist for several genomes:

/scratch/Shares/public/genomes/

See the README.txt file for more information on the origin of these genome files and the directory structure therein.

Alternatively, you can download indexed genome files for your favorite organism from numerous places online including Illumina:

https://support.illumina.com/sequencing/sequencing_software/igenome.html

Alternatively you can create an index from a FASTA file. You only need to index each genome once. Subsequently you can REUSE the index over and over for multiple datasets.

For hg18 (a particular version of the human genome, as fasta) the command to index is:

```
bowtie2-build -f $GLOC/Hg18.fa $SCRATCH/Bowtie2Indexes/Hg18
```

And this took 1 hour and 40 minutes on Pando using 1 core. This job took roughly 6 Gb of memory. Note that for this notation I'm defining placeholders ($GLOC is the path to the genome fasta file location and $SCRATCH is a location on your scratch where you want to keep the index.

Map Reads

We will map the reads from each of your good quality FASTQ files. In other words, if your dataset had issues in quality control, you should address those FIRST.

Cleaning (i.e. filtering) is used to remove poor quality reads – it is not an absolutely necessary step as low quality reads are not expected to align. However, it makes things faster – significantly so when the initial data is of poor quality and/or very large.

Trimming is necessary in some cases, specifically when the sequencer reads off the end of your DNA fragment into the adapter. We won't explicitly go over either of these steps (cleaning or trimming) but you **may** have to deal with them in your term project, depending on what dataset you utilize. There were some extra slides last week showing how to use trimmomatic for trimming. Refer to those slides and the trimmomatic man pages for more details (load module trimmomatic/0.36). An alternative on Fiji is trim_galore (module load `trim_galore/0.4.3`).

Once you are comfortable with the quality of your FASTQ file, there are a number of mapping "strategies" that depend on the sequencing strategy. For example, I can sequence a 1x50 (one end of each DNA fragment, 50 bp of sequence) or 2x125 (paired end sequencing strategy, 150 bp length). Paired end mapping maintains pair information but takes more time.

For general help:
bowtie2 -h

Single end sequencing:

```
bowtie2 -q --phred33 --fast -p 16 -x $INDEX/Hg18
$SCRATCH/RNA_Eli_repA_R1.fastq
-S $SCRATCH/RNA_Eli_repA_R1.sam
```

For an RNA-seq experiment (RNA_Eli_repA_R1) with 46276981 reads, this took 46 minutes on Pando using 16 cores and required less than 4 GB memory.

Paired end sequencing:

```
bowtie2 -q --phred33 --fast -p 16 --rf --minins 200 --maxins 750
-x $INDEX/Hg18
 -1 $SCRATCH/RNA_Eli_repA_R1.fastq -2
$SCRATCH/RNA_Eli_repA_R2.fastq -S $SCRATCH
/RNA_Eli_repA.PE.sam
```

For an RNA-seq experiment with 46276981 pairs, this took 1 hour 28 minutes on Pando using 32 cores and required less than 5 Gb of memory.

Tip: Because of the long runtimes, it is advised that you testdrive your slurm scripts on a small subsampled set of the data. You can create smaller test datasets using head/tail on your primary data file and pipes.

Convert Files

The output of bowtie is a SAM file (sequence alignment file). These are human readable (but boring) and therefore BIG. To do anything with these alignments requires converting the SAM file into a more compact computer friendly format. For subsequent analysis this is the BAM (binary alignment file) and for visualization this is a sorted, indexed BAM file.

SAM to BAM:

```
samtools view -bS -o $SCRATCH/RNA_Eli_repA_R1.bam
$SCRATCH/RNA_Eli_repA_R1.sam
```

`Sort and index:`

```
samtools sort $SCRATCH/RNA_Eli_repA_R1.bam
$SCRATCH/RNA_Eli_repA_R1.sorted

samtools index $SCRATCH/RNA_Eli_repA_R1.sorted.bam
```

Typically these commands are done immediately after mapping, therefore the runtimes given above for single end and paired end bowtie2 include these steps.

Visualization

Looking at your results is a critical aspect of knowing whether your ongoing analysis is (a) worth your time, (b) doing what you think its doing, and (c) working. More importantly, LOOKING at your data is often the best way to figure out what to do next – as no computer beats the human eye at looking for patterns.

HOWEVER, this is also the hardest step in any analysis that involves a cluster computer. Visualization is typically a high memory needs, slow (requires interaction with a user) and data (I/O) intensive. The recommended method of visualization is by IGV:
http://software.broadinstitute.org/software/igv/

Next week we will attempt to visualize (using IGV) directly on the cluster using X2go. You **MUST** install X2go on your local machine before next week's class.
For information on how to install an X2go client on your machine, see:

http://bficores.colorado.edu/biofrontiers-it/cluster- computing/fiji/connecting-to-x2go-on-fiji

Note that you will be asked to log into the bficores site to access this page. You should all have accounts (same identikey and password as used on Fiji).

ALTERNATIVELY, IGV can be installed local to your laptop, ran as a Java program within your browser (if security permits). [Note that in this scenario to visualize a mapped file you must download the BAM file and it's index (.bai) file. Since these files can be big, this is not optimal or recommended.