Module 1: Align Reads to Arabidopsis Genes and Determine Read Counts per Gene

In this module we will analyze Illumina gene expression data generated from a FASTQ file (`wt_mRNA_100K_reads.fq`). This data file contains a sampling of reads from RNA sequencing of wild-type seedlings. For your reference, please access the article that describes the age of these seedlings, their genotype, how they were treated, as well as for that of the pifq mutant. This information should be recorded in the Materials and Methods section of your lab report

We will use BWA to align this data to Arabidopsis genes. BWA is a fast and efficient short read aligner geared towards quickly aligning large sets of short DNA/RNA sequences (reads) to large genomes/transcriptomes. BWA takes an index and a set of reads as input and outputs a list of alignments in a SAM file SAM stands for ‘Sequence Alignment/Map’.

We will use Samtools to access and manipulate the alignments and a Perl script to count the number of reads mapping to each gene.

Resources

You will need the following files. They are available on GitHub at https://github.com/mfcovington/BIS180L-RNAseq.

- cDNA and CDS (coding DNA sequence) FASTA reference files from TAIR (The Arabidopsis Information Resource)
    - `TAIR10_cdna_20110103_representative_gene_model_updated.fa`
    - `TAIR10_cds_20110103_representative_gene_model_updated.fa`
- Sample FASTQ file to map to the FASTA files: `wt_mRNA_100K_reads.fq`
- A Perl script to count the # of reads mapping to each gene: `bam2counts.pl`
- A markdown version of these instructions is also available in the git repository: `README.md`

Documentation for tools used for mapping reads and visualizing alignments can be found online:

- BWA: http://bio-bwa.sourceforge.net/bwa.shtml
- Samtools: http://samtools.sourceforge.net/samtools.shtml

Exercises

Mapping reads to an indexed reference

First, we will index the FASTA reference file. This is required before we can map using `bwa`. Mapping requires a few steps. The first mapping step uses `bwa aln` to create a sequence alignment index. This `.sai` file is converted into a SAM file using `bwa samse` or `bwa sampe` depending on whether your FASTQ file contains single read or paired end data. SAM files can be further converted to BAM files (described below) using `samtools`.

You'll notice that we are assigning long file names and frequently used ID values to variables. This makes our code more readable, reusable, and typo-resistant. Be careful though. Variables will not be defined in other terminal windows. To confirm that the

variable contains what you expect, you can check its contents with: `echo $VARIABLE_NAME`.

Build index of FASTA reference file before mapping:

```
FA=TAIR10_cdna_20110103_representative_gene_model_updated.fa
bwa index $FA
```

Map reads in FASTQ file to FASTA reference:

```
ID=wt_mRNA_100K_reads
bwa aln $FA $ID.fq > $ID.sai
bwa samse $FA $ID.sai $ID.fq > $ID.sam
samtools view -Sb $ID.sam > $ID.bam
```

Visualizing SAM/BAM files

SAM files are plain-text and can be visualized with commands such as `head` and `less`. BAM files contain all the information that the corresponding SAM file contains, but is significantly smaller in size. Since they are binary format, BAM files require a tool like `samtools` for visualization.

The following are equivalent (*note: `-S` turns off word wrap for `less`*):

```
less -S $ID.sam
samtools view -h $ID.bam | less -S
```

These views start by listing every sequence ID from the FASTA reference file. Samtools lets you easily ignore this header (by leaving off the `-h` flag) and get right to the actual alignments.

```
samtools view $ID.bam | less -S
```

You'll see something like this:

```
SRR477075.1 4   *   0   0   *   *   0   0   NGAAACTTCTGATCGTCATGGAAGCTACTTCACAAC    #1
SRR477075.2 16  AT1G52300.1 114 25  36M *   0   0   TTACCCCGCCGCTCGCAAGAGGACTTACAACTGG
SRR477075.3 4   *   0   0   *   *   0   0   CCTCGTTCAAGTCAAGTTGTTGGATGGCCTCCTATA    @@
SRR477075.4 16  AT2G38530.1 205 37  36M *   0   0   GACCGTCAGCAAGCTTGCCGTTGCCTTCAATCTG
...
```

There is a lot of information here. The SAM format is described in full in the Samtools documentation at http://samtools.sourceforge.net/samtools.shtml#5; however, a few key facts about the SAM format will get you started:

- Each read from the FASTQ file is represented by one line in the SAM/BAM file (The read ID is the 1st column)
- The 3rd column indicates the reference sequence that the read maps (if the read fails to map, you will see a `*`)
- The 4th column is the left-most position of the mapped read
- The 5th column is the mapping quality (MAPQ; the higher the better)

Counting number of reads mapped to each gene

In order to do differential gene expression analysis, we first need to know how many reads map to each gene.

Count the number of reads mapping to each gene:

```
perl bam2counts.pl counts.tsv $ID.bam
```

Let's see what the output looks like:

```
head counts.tsv
```

```
wt_mRNA_100K_reads
AT1G01010.1 2
AT1G01020.1 4
AT1G01030.1 0
AT1G01040.2 7
AT1G01050.1 5
AT1G01060.1 2
AT1G01070.1 0
AT1G01073.1 0
AT1G01080.2 2
```

Sorted BAM files

In order to be used to their full potential, BAM files need to be sorted and indexed. This allows the alignment information to be accessed much quicker and many downstream tools require an index (e.g., Integrated Genomics Viewer, IGV).

Sort & Index BAM file:

```
samtools sort $ID.bam $ID.sorted
samtools index $ID.sorted.bam
```

Let's see how the sorted BAM file looks (compare this to how the unsorted BAM file looked above)

```
samtools view $ID.sorted.bam | less -S
```

```
SRR477075.7939  16  AT1G50920.1 50  37  36M *   0   0   AGTTCGTTGACATCATCCTTTCACGGACTC
SRR477075.42057 16  AT1G50920.1 96  37  36M *   0   0   TGTTGTCCACAAGGGTTACAAGATTAACCG
SRR477075.51923 16  AT1G50920.1 130 37  36M *   0   0   CGTCAGTTCTCCATGAGAAAGGTTAAGTAC
SRR477075.53014 0   AT1G50920.1 742 37  36M *   0   0   CGAGCTGCTGTTTTGTTCTTTCTCGACATT
...
```

Questions

Question #1–1

What are three pros/cons of SAM vs BAM format?

Question #1–2

How does the TAIR10 CDS reference differ from the TAIR10 cDNA reference? How does the `bam2counts.pl` output for `AT1G20620.1` differ for CDS vs cDNA and why?

Question #1–3

Mapping multiple FASTQ files and subsequent read counting can be automated as in this example:

```
FA=TAIR10_cdna_20110103_representative_gene_model_updated.fa
for ID in sample1 sample2 sample3 sample4; do
    bwa index $FA
    bwa aln $FA $ID.fq > $ID.sai
    bwa samse $FA $ID.sai $ID.fq > $ID.sam
    samtools view -Sb $ID.sam > $ID.bam
done
perl bam2counts.pl counts.tsv *.bam
```

What are at least two major advantages to such automation?

Question #1–4

There are many different parameters/options available tools like `bwa` and `samtools`. How much does filtering out reads with lower mapping qualities (MAPQ) affect the # of reads mapped?

The command to remove mapped reads with MAPQ less than 20 is:

```
samtools view -Sb -q20 $ID.sam > $ID.q20.bam
```