WTAC Next Generation Sequencing Bioinformatics Course

Read Alignment Practical Exercises

Introduction

We will first “hand-run” a Smith Waterman (local) alignment between two sequences.

We will then align whole-genome sequencing from a mouse zygote which was subject to CRISPR-induced mutagenesis. We will find the resulting engineered alleles, and track down some other alleles in this mouse. This section will be “cued”.

We will then align sequences to a yeast genome, this time with less prompting!

Required Data

For this lab, some pre-prepared datasets have been installed on the Virtual Machine (VM) for you. Double-click the “Module 3 Read Alignment” icon on your desktop to open a terminal window. This window should open in the Read Alignment module directory containing all of the materials needed to complete this practical.

In this directory, two are two exercise directories (Exercise2 and Exercise4) which contain the various data used throughout the practical. There is also a directory titled ‘ref’ which contains the reference genome (FASTA format) of *M. musculus* and *S. Cerevisiae* which will be used in Exercises 2 and 4.

Exercise 1: Do a Smith-Waterman alignment

1.1: Here are the rules and an example of aligning “CTGAG” vs “CGA”.

Scoring:
Match score: +1
Mismatch score: -1
Gap penalty: -1

Starting cells: fill in first row & first column with “0”s.

Rules for cell scoring:
For any given cell, you can enter from the left side, top or diagonally.
If you entered from top or side:
Score in cell = score in prior cell + gap penalty = score in prior cell - 1
If you entered diagonally:
Score in cell = score in prior cell + 1 if sequences match
OR
score in prior cell -1 if sequences mismatch
Choose the highest of these scores! And remember *which* entry cell generated the highest score!
If score in cell is negative, replace score with 0!

Rules for traceback:
Start with highest scoring cell, and trace back to cell of entry (ie did the high score arise when you enter your current cell diagonally or from the sides).

Writing out alignment.
If cell is entered diagonally, write out match or mismatch.
If cell is entered from sides, then write out a gap in the appropriate sequence.

Example of scoring:

		C	T	G	A	G
	0	0	0	0	0	0
C	0	1 −1 −1 1	−1 −1 0 0	−1 −1 −1 0	−1 −1 −1 0	−1 −1 −1 0
G	0	−1 0 −1 0	0 −1 −1 0	1 −1 −1 1	−1 −1 0 0	−1 −1 −1 0
A	0	−1 −1 −1 0	−1 −1 −1 0	−1 0 −1 0	2 −1 −1 2	−1 −1 1 1

Now choose the highest score in the matrix and trace-back , choosing the cell which resulted in the highest score.

		C	T	G	A	G
	0	0	0	0	0	0
C	0	1 −1 −1 1	−1 −1 0 0	−1 −1 −1 0	−1 −1 −1 0	−1 −1 −1 0
G	0	−1 0 −1 0	0 −1 −1 0	1 −1 −1 1	−1 −1 0 0	−1 −1 −1 0
A	0	−1 −1 −1 0	−1 −1 −1 0	−1 0 −1 0	2 −1 −1 2	−1 −1 1 1

Alignment: If a cell was entered diagonally, write a match/mismatch between two sequences. If it was entered horizontally / vertically, enter a gap in one sequence or the other:

```
C T G A G
C — G A –
```

1.2: Repeat with these two sequences: "CTGAG" and "TAG"

CTGAG
And
TAG

		C	T	G	A	G
	0	0	0	0	0	0
T	0					
A	0					
G	0					

Solution:

		C	T	G	A	G
	0	0	0	0	0	0
T	0	−1 −1 −1 0	1 −1 −1 1	−1 −1 0 0	−1 −1 −1 0	−1 −1 −1 0
A	0	−1 −1 −1 0	−1 0 −1 0	0 −1 −1 0	1 −1 −1 1	−1 −1 0 0
G	0	−1 −1 −1 0	−1 −1 −1 0	1 −1 −1 1	−1 0 0 0	2 −1 −1 2

```
C T G A G
- T — A G
```

Exercise 2: BWA Alignment / inspection of mouse variation

We will use the BWA aligner to align one small region of illumina sequencing data to the *Mus Musculus* genome. You will align genomic sequence (from Whole-Genome Sequencing) from a mouse embryo which has been mutagenised while the one-cell stage using CRISPR-Cas9 and a gRNA targeting an exon of the *Tyr* gene. The successful mutation of the gene will delete one or both alleles. A bi-allelic null *Tyr* mouse will be albino, but otherwise healthy.

2.1: View the the reference genome

Goto the 'ref' directory that contains the fasta files of the reference genomes:
~/course_data/Module3_ReadAlignment/ref

Fasta files (.fa) are used to store raw sequencing information before aligning data. The mouse genome file is here: GRCm38.68.dna.toplevel.fa

View the file with less:
$ less GRCm38.68.dna.toplevel.fa

Question: What is the length of chromosome 1 of the mouse genome? *(Look at the fasta header for chromosome 1)*

Question: Can you *quickly* check if there other sequences in the assembly other than the 'standard' chromosomes? *(Try grep '>' GRCm38.68.dna.toplevel.fa)*

Similar to a BAM file, to allow fast retrieval of data, and index file is often required. In this case we have already created both the fasta index for the genome:

GRCm38.68.dna.toplevel.fa.fai – allows rapid sequence retrieval with samtools
GRCm38.68.dna.toplevel.fa.amb ... GRCm38.68.dna.toplevel.fa.sa – created by bwa: suffix trees, bwt transform etc etc.

2.2: Align the paired fastq files with bwa

Goto the '~/course_data/Module3_ReadAlignment/Exercise2/fastq/' directory. We will align the fastq files to the mouse reference genome using bwa.

Use the 'bwa mem ' command to align the fastq files. Bwa outputs sam files by default, so you will have to pipe the result into a sam file.

```
bwa mem ~/course_data/Module3_ReadAlignment/ref/GRCm38.68.dna.toplevel.fa
md5638a_7_87000000_R1.fastq md5638a_7_87000000_R2.fastq > md5638.sam
```

2.3: Convert a SAM file to a BAM file

Now we need to convert the SAM file ('md5638.sam') from the previous step into a BAM file.

Convert the SAM file into a BAM file called 'md5638.bam' using samtools.

Hint: to do this conversion use 'samtools view'. What options are required to input a SAM file and output a BAM file?

```
samtools view -O BAM -o md5638.bam md5638.sam
```

How much space is saved by using a bam file instead of sam?

2.4: Sort and index the BAM file

The BAM files produced by BWA are sorted by read name (same order as the original fastq files). However, most viewing and variant calling software required the BAM files to be sorted by reference co-ordinate position and indexed for rapid retrieval:

Therefore, use 'samtools sort' to produce a new BAM file ('md5638.sorted.bam') that is sorted by position.

Finally, can you index the sorted BAM file using 'samtools-1.5 index' command?

Note: indexing a BAM file is also a good way to check that the BAM file has not been truncated (e.g. your disk becomes full when writing the BAM file). At the end of every BAM file, a special end of file (EOF) marker is written. The Samtools index command will first check for this and produce an error message if it is not found.

```
samtools sort -T temp -O bam -o md5638.sorted.bam md5638.bam

samtools index md5638.sorted.bam
```

2.5: Unix pipes to combine the commands together

To produce the sorted BAM file in 2.1-2.3 we had to carry out several separate commands and produce intermediate files. The Unix pipe command allows you to feed the output of one command into the next command.

Combine all of these commands together using unix pipes, and do all of this data processing together and avoid writing intermediate files.

```
bwa mem ~/course_data/Module3_ReadAlignment/ref/GRCm38.68.dna.toplevel.fa
md5638a_7_87000000_R1.fastq md5638a_7_87000000_R2.fastq | samtools view -O BAM
- | samtools sort -T temp -O bam -o md5638.sorted.bam -
```

2.5: Mark PCR Duplicates

We will use a program called 'MarkDuplicates' that is part of Picard tools (http://picard.sourceforge.net) to remove PCR duplicates that may have been introduced during the library construction stage. To find the options for 'MarkDuplicates' – type:

```
picard-tools MarkDuplicates
```

Now run MarkDuplicates using the 'I=' option to specify the input BAM file and the 'O=' option to specify the output file (e.g. md5638.markdup.bam). You will also need to specify the duplication metrics output file using 'M=' (e.g. md5638.markdup.metrics).
Don't forget to index your final bam file using 'samtools index'.

From looking at the output metrics file - how many reads were marked as duplicates? What was the percent duplication?

```
picard-tools MarkDuplicates I= md5638.sorted.bam O=md5638.markdup.bam
M=md5638.metrics.txt
```

Generate an index for the bam file using samtools.

```
samtools index md5638.markdup.bam
```

2.5: Generate QC Stats

Use samtools to collect some statistics about the alignments in the BAM file from the last step. To run the 'stats' command - type:

```
samtools stats md5638.markdup.bam > md5638.markdup.stats
```

Then look at the output and answer the following questions:

What is the total number of reads?

What proportion of the reads were mapped?

How many reads were paired correctly/properly?

How many reads mapped to a different chromosome?

What is the insert size mean and standard deviation?

Next we will create some QC plots from the output of the stats command. Make sure you have saved the output of the stats command to a file (e.g. lane1.stats.txt). We will use the 'plot-bamstats' command that is part of Samtools:

```
plot-bamstats -p md5639_plot md5638.markdup.stats
```

Now in your web browser open the file called `md5639_plot`.html to view the QC information.

How many reads have zero mapping quality?

Do *any* of the graphs look odd to you?

Which of the first fragments or second fragments are higher base quality on average? Note: Look at the first of the 'Quality per cycle' graphs.

2.6: BAM Visualisation

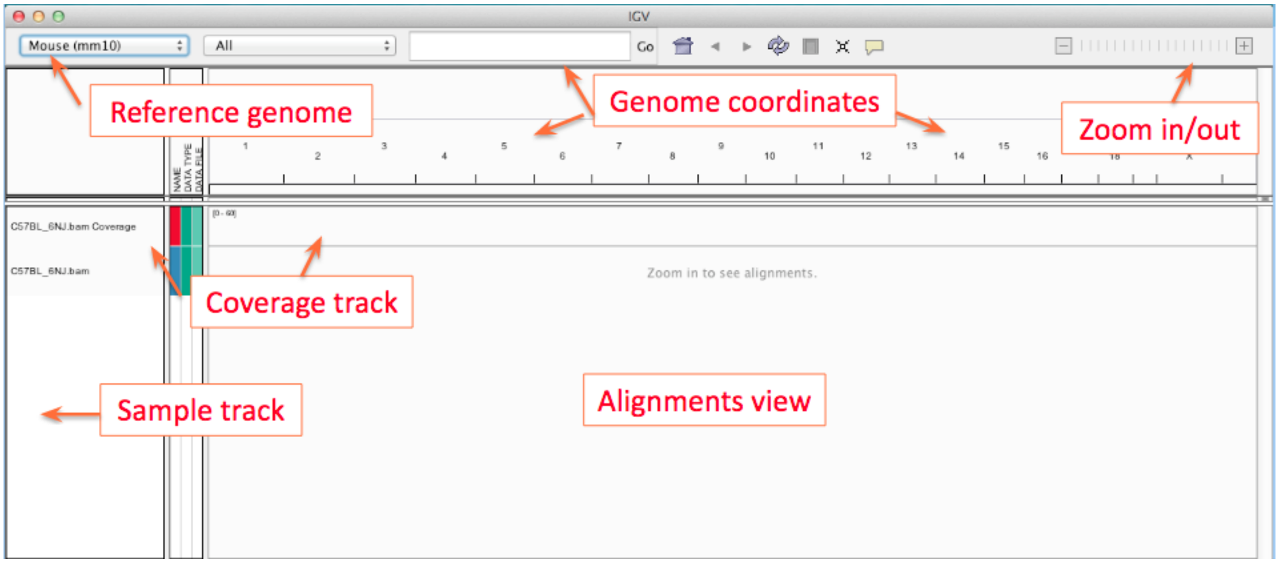
Congratulations! You made it to the Good Bit! *I find actually seeing the seeing sequence variation – natural or engineered - and pondering its relation to biological effects quite compelling. I hope you do too!*

IGV (http://www.broadinstitute.org/igv/) is a very useful visualisation tool for looking at the alignments of reads onto a reference genome from BAM files.

In the 'fastq' directory, you can start IGV by typing:

```
igv.sh
```

Here's a great IGV tutorial and refresher:
https://github.com/sanger-pathogens/pathogen-informatics training/blob/master/Notebooks/IGV/IGV.pdf



2.6.1 Load the reference genome, on the top menu bar find the genomes dropdown (top-left) and select "mouse mm10". This is a synonym for GRCm38, which is the current mouse assembly.

If - for whatever reason, this fails - you can load the genome from a file (Genomes=>Load From File using the genome in the “ref” directory:
GRCm38.68.dna.toplevel.fa
and load gene annotations from file File=>Load from file using this file:
Mus_musculus.GRCm38.93.chr.sorted.gtf

2.6.2 Load your BAM file, on the top menu bar goto ‘File –> Load from File…’ and select the `md5638.markdup.bam` file that you created in the previous step.

2.6.3 Set up basic view preferences, using:

Popup preferences – use the little “speech bubble” button on the top icon list to set popups on *click only* ***(or you will go mad, I promise).***

Track preferences (Command-click on the track at left). Sort alignments by insert size. Colour alignments by insert size. Also, choose “View as pairs”.

View preferences (View menu item): View=>preferences=>Alignments. Show soft-clipped bases. *This colour highlighting emphasises soft-clips on the read itself.*

2.6.4 Inspect an interesting region of the mouse *Tyr* locus.

Go to chromosome 7, positions 87,483,625-87,484,330 using the navigation bar across the top. *Take in the glorious view of a genome pileup.* Stop and smell the roses! Click on stuff! Scroll around, zoom in and out a bit!

2.6.5 Questions about the pileup and visible variation:

Go back to chromosome 7:87,483,625-87,484,330. What is the (rough) coverage across this region?

There are three mutant variants (two small and one larger) in this region: Can you spot them, state what the evidence is for them, and work out their allele fraction? Can you venture a guess as to what happened here? Why are these mutations present? Why might they be subclonal?

Hints:

1. Look around 87,483,960 for an insertion. How large is it? How many reads does it occur in?
2. Look around 87,483,960 for a deletion. How large is it? How many reads does it occur in?
3. Zoom out slightly and watch the coverage track between 87,483,700 - 87,484,200. Once you’ve spotted the large change look at reference sequence the edges of the mutation to hazard a guess as to its mechanism.

What mutations can you see? *There is a 1bp insertion (at "T") with VAF ~0.3 at position 7:87,483,965. There is a 28bp deletion with VAF ~0.15 starting at 7:87,483,960. There is a 338bp deletion with a VAF of about 0.25 starting at position 7:87,483,831.* ***Why are they there?*** *The CRISPR-Cas9 has acted on the zygote at this locus to create Non-Homologous-End-Join-based damage around 87,483,960: that resulted in a subclonal 1bp insertion and a 28bp deletion. Microhomology–induced-end-joining resulted in the 338bp deletion (can you see the "TTT" motif on the 5' end of the deletion, and just inside the 3' end of the deletion? You are watching the zygote DNA-repair machinery panicking and grabbing at straws).* ***Why are these alleles subclonal?*** *Because the action of the CRISPR-Cas9 occurred both at the single-cell and the two-cell stage.*

2.6.6 Looking for natural SNV's and Indels

At each of the following genomic locations, write down the variant, its allele fraction, and whether you can find it in the Mouse Genomes Search facility:
(https://www.sanger.ac.uk/sanger/Mouse_SnpViewer/)

Location1. 7:87258490
Location2. 7:87834414
Location3. 7:87251720
Location4. 7:87303315
Location5. 7:87392116
Location6: 7:87428859

Answer:
1. *Hom SNV/no,*
2. *Hom SNV/no,*
3. *Hom 2bp and Hom 4bp deletion/sort of: The MGP shows a 4p deletion only, and this is one region which may have benefited from indel-realignment! Note excess repeats in the area surrounding.*
4. *Hom single bp deletion/yes.*
5. *Hom single bp insertion/yes.*
6. *Hom (?) 2b deletion/yes.*

Exercise 4: BWA Alignment / lane merging / inspection with YEAST

We will use the BWA aligner to align 2 lanes of illumina sequencing data to the *Saccromyces cerevisiae* genome (ftp://ftp.ensembl.org/pub/current_fasta/saccharomyces_cerevisiae/dna/).

4.1: Index the reference genome with bwa

Goto the 'ref' directory that contains the fasta files of the reference genome. Fasta files (.fa) are used to store raw sequencing information before aligning data. Similar to a BAM file, to allow fast retrieval of data, and index file is often required. You can use the 'bwa index' command to create a reference genome index that bwa can use.

Do you see any new files? How many?:

Note: Generally, when a tool creates an index for a file, the index will have almost the same name as the original but include a new ending. For example, do you see a file ending in .bwt?:

4.2: Align the lane fastq files with bwa

Goto the 'Exercise4/60A_Sc_DBVPG6044/library1/lane1/' directory and we will align the fastq files using bwa.

When aligning data, we often want include additional information of relevance to a project in the header of a file. Next create a lane SAM file called 'lane1.sam' with the following SAM header (**Hint:** type 'bwa mem' and look for the "Input/output options" section of the printed help page):

```
'@RG\tID:lane1\tSM:60A_Sc_DBVPG6044'
```

Use the 'bwa mem' command to align the fastq files and use the appropriate option to include the about header information. Don't forget to use the -M option (mark shorter split hits as secondary).

Hint: This will create a SAM so needs to be output to a file called lane1.sam (> lane1.sam).

4.3: Convert a SAM file to a BAM file

Now we need to convert the SAM file ('lane1.sam') from the previous step into a BAM file. Convert the SAM file into a BAM file called 'lane1.bam' using samtools.

Hint: to do this conversion use 'samtools view'. What options are required to input a SAM file and output a BAM file?

4.4: Sort and index the BAM file

The BAM files produced by BWA are sorted by read name (same order as the original fastq files). However, most variant calling software required the BAM files to be sorted by reference co-ordinate position to allow rapid retrieval of data. Therefore, use 'samtools sort' to produce a new BAM file ('lane1.sorted.bam') that is sorted by position.

Look at the first line of the header of the BAM file, is it coordinate sorted?:

Finally, can you index the sorted BAM file using 'samtools index' command?

Note: indexing a BAM file is also a good way to check that the BAM file has not been truncated (e.g. your disk becomes full when writing the BAM file). At the end of every BAM file, a special end of file (EOF) marker is written. The Samtools index command will first check for this and produce an error message if it is not found.

4.5: Unix pipes to combine the commands together

To produce the sorted BAM file in 2.1-2.3 we had to carry out several separate commands and produce intermediate files. The Unix pipe command allows you to feed the output of one command into the next command.

So using Unix pipes, we can combine all of these commands together and do all of this data processing together and avoid writing intermediate files.

```
bwa mem -M -R '@RG\tID:lane1\tSM:60A_Sc_DBVPG6044'
../../../../ref/Saccharomyces_cerevisiae.EF4.68.dna.toplevel.fa s_7_1.fastq
s_7_2.fastq | samtools view -bS - | samtools sort -T temp -O bam -o
lane1.sorted.bam -
```

4.6: Generate QC Stats

We will use samtools to collect some statistics about the alignments in the BAM file from the last step (remember to output this to a stats file, e.g. a file ending in .stats.txt). To run the 'stats' command - type:

```
samtools stats lane1.sorted.bam
```

Then look at the output and answer the following questions (**Hint** look for rows beginning with SN):

What is the total number of reads?

What proportion of the reads were mapped?

How many reads were paired correctly/properly?

How many reads mapped to a different chromosome?

What is the insert size mean and standard deviation?

Next we will create some QC plots from the output of the stats command. Make sure you have saved the output of the stats command to a file (e.g. lane1.stats.txt). We will use the 'plot-bamstats' command that is part of Samtools:

```
plot-bamstats -p plot lane1.stats.txt
```

(Aside: If you have problems running `plot-bamstats`, you can download the results from here: **http://tinyurl.com/h37d8bx**).

Now in your web browser open the file called plots.html to view the QC information.

How many reads have zero mapping quality?

Which of the first fragments or second fragments are higher base quality on average? Note: Look at the 'Quality per cycle' graphs.

4.7: Align Lane 2

There is a second lane in the 'library1' directory called 'lane2'. We want to also align this lane also to produce a BAM file.

Goto the 'Exercise4/60A_Sc_DBVPG6044/library1/lane2' directory. Now repeat exercise 2 using the fastq files in the lane2 directory to produce a sorted BAM file. **Note:** This time when you use the 'bwa mem' command use the following header option to specify lane2 as the read group ID:

```
'@RG\tID:lane2\tSM:60A_Sc_DBVPG6044'
```

4.8: Merge the lane BAMS

Go to the '60A_Sc_DBVPG6044/library1' directory. Use 'ls' to get a listing of the files and directories contained in this directory.

You will notice that there are two directories called 'lane1' and 'lane2'. There were two sequencing lanes produced from this sequencing library. In order to mark library PCR duplicates, we need to merge the two lane BAM files together to produce a single BAM file.

We will use the picard tool called 'MergeSamFiles' (http://picard.sourceforge.net) to merge the lane BAM files. Picard-tools is a collection of commands (with their own options) for interacting with BAM files. Look at the list of commands available for Picard-tools by running 'picard-tools'. To find the options for 'MergeSamFiles' command, type:

```
picard-tools MergeSamFiles
```

Now use the 'I=' option to specify both the input BAM files and the 'O=' option to specify the output file (e.g. library1.bam). **Note:** Multiple input files can be specified using 'I='

4.6: Mark PCR Duplicates

We will use a program called 'MarkDuplicates' that is part of Picard tools (http://picard.sourceforge.net) to remove PCR duplicates that may have been introduced during the library construction stage. To find the options for 'MarkDuplicates' – type:

```
picard-tools MarkDuplicates
```

Now use the 'I=' option to specify the input BAM file and the 'O=' option to specify the output file (e.g. library1.markdup.bam). You will also need to specify the duplication metrics output file using 'M=' (e.g. library1.markdup.metrics).
Don't forget to index your final bam file using 'samtools index'.

From looking at the output metrics file - how many reads were marked as duplicates? What was the percent duplication?

4.9: BAM Visualisation

IGV (http://www.broadinstitute.org/igv/) is a very useful visualisation tool for looking at the alignments of reads onto a reference genome from BAM files.

In the 'library1' directory, you can start IGV by typing:

```
Igv.sh &
```

To load the reference genome, on the top menu bar goto 'Genomes –> Load Genome From File...' and select the reference genome in the 'ref' directory.

Next to load your BAM file, on the top menu bar goto 'File –> Load from File…' and select the library BAM file that you created in the previous step.

Now goto Chromosome IV and position 764,293 using the navigation bar across the top.

What is the reference base at this position?

Do the reads agree with the reference base?

What about the adjacent position (IV:764,292)? What is the reference base at this position? Is it supported by the reads?

Now goto Chromosome IV and position 766,588 using the navigation bar across the top.

What sort of mutation are the alignments indicating might be present?

Now goto Chromosome IV and position 770,137 using the navigation bar across the top.

What sort of mutation are the alignments indicating might be present? Is there anything in the flanking sequence of the reference genome that might make you suspicious about this mutation?

Software URLs

Name	**URL**
Burrows-Wheeler Aligner (BWA)	http://bio-bwa.sourceforge.net/ http://github.com/lh3/bwa
Samtools	http://www.htslib.org
Picard Tools	https://broadinstitute.github.io/picard/
IGV	http://software.broadinstitute.org/software/igv/