DIFFERENTIAL GENE EXPRESSION ANALYSIS

Module 3: Alignment

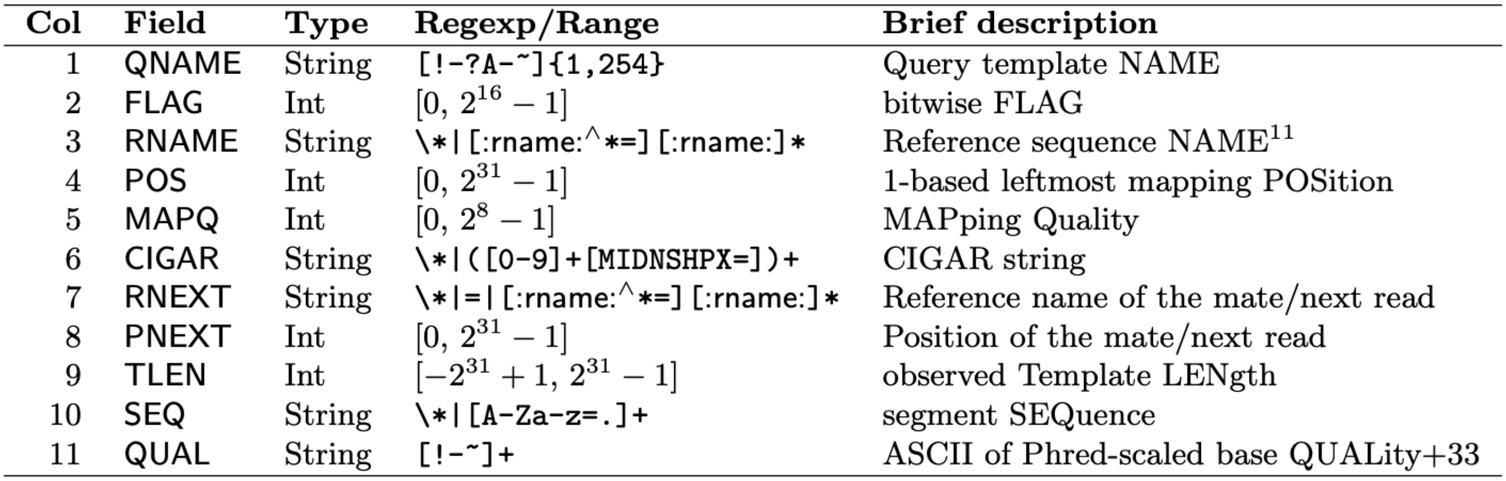
GENERATE ALIGNMENTS USING HISAT2

```
# Change directory to where the genome fasta file exists

cd /home/<username>/DGE_Virtual/human_reference/

mkdir hisat2_index/

cd hisat2_index/

################### PLEASE DO NOT RUN THE FOLLOWING COMMAND TO COMPLETION ###################

source activate HISAT

# To index a reference genome

hisat2-build --help

hisat2-build ../GRCh38.p12.genome.fa GRCh38.p12.genome

# GRCh38.p12.genome.fa -> Reference Sequence
# GRCh38.p12.genome -> Index files are created with this base name

# The above command is a time limiting step (it took approximately 72 minutes)
# We will instead use index files already created
# Remember to run this on a screen
#############################################################################################

# Since you did not run the previous step to completion, you will copy index files
# from my workspace to your current folder (hisat2_index)

cp /home/elavelle/DGE_Virtual/human_reference/hisat2_index/*ht2 ./

# Make an output directory

mkdir /home/$USER/DGE_Virtual/hisat2_alignments

#Start a screen

screen -S <screen_Name>

source activate HISAT

# Move to folder containing the read files
```

```
cd /home/$USER/DGE_Virtual/raw_reads

# To run one single-end sample

hisat2 --help

hisat2 -x /home/$USER/DGE_Virtual/human_reference/hisat2_index/GRCh38.p12.genome \
-U 2S1Flag-p5-2.fq.gz \
--threads 6 \
-S /home/$USER/DGE_Virtual/hisat2_alignments/2S1Flag-p5-2.sam

# -x: index filename prefix
# -p: threads
# -U: unpaired
# -S: SAM output

# The backslashes are just to escape the invisible newline character and continue a new line

# To run multiple samples at once using for loop on the command line:

for file in *.fq.gz; do hisat2 \
-x /home/$USER/DGE_Virtual/human_reference/hisat2_index/GRCh38.p12.genome \
-U ${file} \
--threads 4 \
-S /home/$USER/DGE_Virtual/hisat2_alignments/${file}.sam; done

#Detach from screen

Ctrl a+d (^a^d)

#Exercise: What flags will you use for paired-end reads?

hisat2 -x /path/to/GRCh38.p12.genome \
--threads 4 \
-1 /path/to/read1.fastq \
-2 /path/to/read2.fastq \
-S /path/to/outputfile.sam
```

ALIGNMENTS FROM HISAT2 ARE REPRESENTED IN SAM (SEQUENCE ALIGNMENT MAP) FORMAT

SAM ONLINE RESOURCES



<table>
<tr><th>Col</th><th>Field</th><th>Type</th><th>Regexp/Range</th><th>Brief description</th></tr>
<tr><td>1</td><td>QNAME</td><td>String</td><td>[!-?A-~]{1,254}</td><td>Query template NAME</td></tr>
<tr><td>2</td><td>FLAG</td><td>Int</td><td>$[0, 2^{16} - 1]$</td><td>bitwise FLAG</td></tr>
<tr><td>3</td><td>RNAME</td><td>String</td><td>*|[:rname:^*=][:rname:]*</td><td>Reference sequence NAME[11]</td></tr>
<tr><td>4</td><td>POS</td><td>Int</td><td>$[0, 2^{31} - 1]$</td><td>1-based leftmost mapping POSition</td></tr>
<tr><td>5</td><td>MAPQ</td><td>Int</td><td>$[0, 2^{8} - 1]$</td><td>MAPping Quality</td></tr>
<tr><td>6</td><td>CIGAR</td><td>String</td><td>*|([0-9]+[MIDNSHPX=])+</td><td>CIGAR string</td></tr>
<tr><td>7</td><td>RNEXT</td><td>String</td><td>*|=|[:rname:^*=][:rname:]*</td><td>Reference name of the mate/next read</td></tr>
<tr><td>8</td><td>PNEXT</td><td>Int</td><td>$[0, 2^{31} - 1]$</td><td>Position of the mate/next read</td></tr>
<tr><td>9</td><td>TLEN</td><td>Int</td><td>$[-2^{31} + 1, 2^{31} - 1]$</td><td>observed Template LENgth</td></tr>
<tr><td>10</td><td>SEQ</td><td>String</td><td>*|[A-Za-z=.]+</td><td>segment SEQuence</td></tr>
<tr><td>11</td><td>QUAL</td><td>String</td><td>[!-~]+</td><td>ASCII of Phred-scaled base QUALity+33</td></tr>
</table>

https://samtools.github.io/hts-specs/SAMv1.pdf

http://www.htslib.org/doc/sam.html

https://en.wikipedia.org/wiki/SAM_(file_format)

ALIGNMENT METRICS

Some alignment tools (HISAT2 for example) will print alignment metrics after generating alignments. However, these metrics may not be available as a result of other alignment tools. Hence, it is useful to know the following one-liners to extract information on important metrics from SAM files.

```
cd /home/$USER/DGE_Virtual/hisat2_alignments/

ls -ltr

# Use the "rename" command to edit filenames
# rename <FROM> <TO> <FILES TO RENAME>

rename .fq.gz.sam .sam *.fq.gz.sam

# Employ a function from the samtools environment to summarize statistics from a .sam file

source activate samtools

samtools flagstat 2S1Flag-p5-2.sam

14943130 + 0 in total (QC-passed reads + QC-failed reads)
3126633 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
14062988 + 0 mapped (94.11% : N/A)
0 + 0 paired in sequencing
0 + 0 read1
0 + 0 read2
0 + 0 properly paired (N/A : N/A)
0 + 0 with itself and mate mapped
0 + 0 singletons (N/A : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

# The number in the first row is the count of records in the .sam file
# Confirm this by counting the rows not including the header lines:

grep -v "^@" 2S1Flag-p5-2.sam | wc -l
14943130

# Secondary alignments counts the alignments of reads that mapped to additional locations on the genome.
# To omit these, Count the number of unique read IDs in the file:

grep -v "^@" 2S1Flag-p5-2.sam | awk '{print $1}' | uniq | wc -l
11816497

# Check this with arithmetic from the flagstat output: 14943454-3126957

# However, this count also includes reads which didn't map at all. Filter those out
# ("*" in column 3) to find the number of reads that mapped once or more.
```

```
cat 2S1Flag-p5-2.sam | grep -v '^@' | awk '{ if ($3 != "*") print $0}' \
| awk '{print $1}' | uniq | wc -l
10936355

# The mapped number from the flagstat output counts all alignments (not reads!)
# Subtract the secondary reads from this value to check our result: 14063297-3126957

# As it happens, there is another convenient samtools function to extract the desired metrics from # a .sam file:

samtools view -f 0x100 -c 2S1Flag-p5-2.sam

# Including the "-f" option in the samtools "view" command will print to stdout the records
# matching the corresponding bit flag shown in the table below.

# This command, for example, counts (due to the inclusion of the "-c" option) all secondary reads
# Notice it matches the number found with the other methods

# The "-F" option is similar to grep's "-v" option; it pulls the OPPOSITE records from what the
# bit flag describes. Moreover, these bit flags can be combined- e.g., 904 = 800 + 100 + 4
# Therefore, the number of primary alignments can also be found by:

samtools view -F 0x904 -c 2S1Flag-p5-2.sam
```

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	**SEQ** being reverse complemented
32	0x20	**SEQ** of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing filters, such as platform/vendor quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

EXERCISE: COMPLETE THE FOLLOWING TABLE

File Name	Total Number of Reads	Total Mapped Reads	Total Primary Alignments
2S1Flag-p5-2.fq.gz			
2S1Flag-p6-3.fq.gz			
2S1Flag-p7-2.fq.gz			
759_7-p5-2.fq.gz			
759_7-p6-1-1.fq.gz			
759_7-p6-2-2.fq.gz			
pCDNA_p6-3.fq.gz			
pCDNA_p7-2.fq.gz			
pCDNA_p8-3.fq.gz			
Scram_1-3.fq.gz			
Scram_1_p3-1.fq.gz			
Scram_1_p3-3.fq.gz			