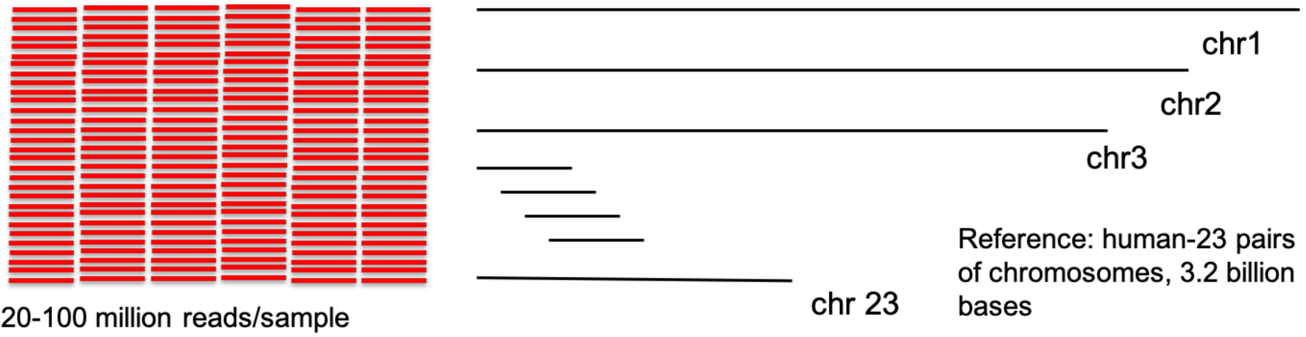
DIFFERENTIAL GENE EXPRESSION ANALYSIS MODULE

June 11, 2020

ALIGN READS TO A GENOME

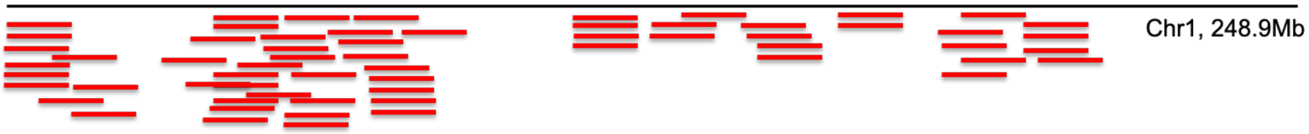


~20-100 million reads/sample

Reference: human-23 pairs of chromosomes, 3.2 billion bases

Where do I belong on the genome?

STEP1: Alignments – Find a location for the reads on the genome



I found a spot. What is the biological context?

GENERATE ALIGNMENTS USING HISAT2 (You will need raw reads (fastq) as input and your genome fasta file as input

```
# Generating Alignments

# Create/Build a genome index/database.

# Change directory to where the genome fasta file exists

cd /home/usrname/DGE_Virtual_June2020/human_reference/

mkdir hisat2_index/

cd hisat2_index/
```

```
source activate bio

# HISAT2 Aligner

# To index a reference genome

# PLEASE DO NOT RUN THE FOLLOWING COMMAND

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 and compute limiting step (for human genome reference it
# took approximatley 72 minutes) and hence you will use the genome index files that have
# been created already. Remember if you have your own dataset with another version of human
# genome or any other organism, you must run this step before kick starting the alignments in the next command.

# In the following steps we will use a small subset of input reads to
# demonstrate the process of generating alignments.
# We have created a folder that has only first 100,000 sequence reads for 12 samples

cd /home/usrname/DGE_Virtual_June2020/

mkdir subset_input_reads/

cd subset_input_reads/

cp /home/asundara/DGE_Virtual_May2020/subset_input_reads/* ./

cd ../

mkdir hisat2_alignments_subset/

cd /home/usrname/DGE_Virtual_May2020/subset_input_reads/

#Start a screen

screen -S <screen_Name>

source activate bio

#For single-end reads

#To run one sample at a time

hisat2 --help

# The following command takes approximately 6 seconds to run to completion (this may vary)

hisat2 -x /home/asundara/DGE_Virtual_May2020/human_reference/hisat2_index/GRCh38.p12.genome \
-U 2S1Flag-p5-2_lib_idx5_ACAGTG_0MM.subset.fq.gz \
--threads 4 \
-S /home/usrname/DGE_Virtual_May2020/hisat2_alignments_subset/2S1Flag-p5-2_lib_idx5_ACAGTG_0MM.sam

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

#The backslashes are just to escape the enter 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/asundara/DGE_Virtual_May2020/human_reference/hisat2_index/GRCh38.p12.genome \
-U ${file} \
--threads 4 \
-S /home/usrname/DGE_Virtual_May2020/hisat2_alignments_subset/${file}.sam; done

#Detach from screen

Cntrl ^a^d
```

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

SAM ONLINE RESOURCES

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

```
cd /home/usrname/DGE_Virtual_May2020/hisat2_alignments_subset/

ls -ltr

# USE RENAME COMMAND TO MODIFY FILE NAMES

rename <FROM> <TO> <FILES-TO-RENAME>

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

# Total Reads from SAM file (the following command will cat the file and grab (grep) everything
# that does not (-v) begin (^) with "@". This command is combined to the next command using | where you
# use awk to just look at first column ($1) | count lines (wc -l) of unique (uniq) instances)

cat 2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.subset.sam | grep -v '^@' | awk '{print $1}' | uniq | wc -l
100000

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

for file in *.sam; do echo ${file}; cat ${file} | grep -v '^@' | awk '{print $1}'| uniq | wc -l; done

2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.subset.sam
100000
```

```
2S1Flag-p6-3_lib_idx4_TGACCA_0MM.subset.sam
100000
2S1-Flag-p7-2_lib_idx2_CGATGT_0MM.subset.sam
100000
759_7-p5-2_lib_idx6_GCCAAT_0MM.subset.sam
100000
759_7-p6-1-1_lib_idx12_CTTGTA_0MM.subset.sam
100000
759_7-p6-2-2_lib_idx7_CAGATC_0MM.subset.sam
100000
pCDNA_p6-3_lib_idx5_ACAGTG_0MM.subset.sam
100000
pCDNA_p7-2_lib_idx4_TGACCA_0MM.subset.sam
100000
pCDNA_p8-3_lib_idx2_CGATGT_0MM.subset.sam
100000
Scram_1-3_lib_idx6_GCCAAT_0MM.subset.sam
100000
Scram_1_p3-1_lib_idx12_CTTGTA_0MM.subset.sam
100000
Scram_1_p3-3_lib_idx7_CAGATC_0MM.subset.sam
100000

# Total number uniquely aligning reads (the following command will cat the file and grab (grep) everything
# that does not (-v) begin (^) with "@". This command is combined to the next command using | where you
# use awk to check if column 3 is not equal to * ($3 != "*") and print the first
# column (print $1) | count (uniq -c) the number times every ID occurs | and only do a
# word count (wc -l) of those IDs that have one in the first column ($1==1))

cat 2S1Flag-p5-2_lib_idx5_ACAGTG_0MM.subset.sam | grep -v '^@' | awk '{ if ($3 != "*") print $1}' |\
uniq -c | awk '{ if ($1 == 1) print $0}' | wc -l
77828

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

for file in *.sam; do echo ${file}; cat ${file} | grep -v '^@' | awk '{ if ($3 != "*") print $0}' \
| awk '{print $1}' | uniq -c | awk '{print $1 "\t" $2}' | awk '{ if ($1 == 1) print $0}' | wc -l; done

2S1Flag-p5-2_lib_idx5_ACAGTG_0MM.subset.sam
77828
2S1Flag-p6-3_lib_idx4_TGACCA_0MM.subset.sam
77527
2S1-Flag-p7-2_lib_idx2_CGATGT_0MM.subset.sam
77476
759_7-p5-2_lib_idx6_GCCAAT_0MM.subset.sam
80639
759_7-p6-1-1_lib_idx12_CTTGTA_0MM.subset.sam
80815
759_7-p6-2-2_lib_idx7_CAGATC_0MM.subset.sam
80841
pCDNA_p6-3_lib_idx5_ACAGTG_0MM.subset.sam
78340
pCDNA_p7-2_lib_idx4_TGACCA_0MM.subset.sam
77782
pCDNA_p8-3_lib_idx2_CGATGT_0MM.subset.sam
78305
Scram_1-3_lib_idx6_GCCAAT_0MM.subset.sam
80464
Scram_1_p3-1_lib_idx12_CTTGTA_0MM.subset.sam
```

```
80541
Scram_1_p3-3_lib_idx7_CAGATC_OMM.subset.sam
80773
```

Since you will need the complete alignment files (sam) for the next step, please copy over the files from instrctors working directory.

```
#In the following four commands you will copy alignment output files(sam) from
instructor (asundara) working directory to a location in your workspace

cd /home/usrname/DGE_Virtual_June2020/

mkdir hisat2_alignments

cd hisat2_alignments

cp /home/asundara/DGE_Virtual_June2020/hisat2_alignments/*.sam .

#DO THE FOLLOWING AS HOMEWORK

cd /home/usrname/DGE_Virtual_June2020/

mkdir hisat2_alignments_practice

cd raw_reads/

#Start a screen

screen -S <screen_Name>

source activate bio

#For single-end reads

#To run one sample

hisat2 --help

# The following command takes approximately 6 seconds to run to completion

hisat2 -x /home/asundara/DGE_Virtual_June2020/human_reference/hisat2_index/GRCh38.p12.genome \
-U 2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.fq.gz \
--threads 4 \
-S /home/usrname/DGE_Virtual_May2020/hisat2_alignments_practice/2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.sam

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

#The backslashes are just to escape the enter 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/asundara/DGE_Virtual_June2020/human_reference/hisat2_index/GRCh38.p12.genome \
-U ${file} \
--threads 4 \
-S /home/usrname/DGE_Virtual_June2020/hisat2_alignments_practice/${file}.sam; done

#Detach from screen

Cntrl ^a^d
```

STEP 2: GENERATE ALIGNMENT METRICS

```
cd /home/usrname/DGE_Virtual_June2020/hisat2_alignments_practice/

#Start a screen

screen -S <screen_Name>

source activate bio

ls -ltr

# USE RENAME COMMAND TO MODIFY FILE NAMES

rename <FROM> <TO> <FILES-TO-RENAME>

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

# Total Reads from SAM file

cat 2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.sam | grep -v '^@' | awk '{print $1}'| uniq | wc -l

for file in *.sam; do echo ${file}; cat ${file} | grep -v '^@' | awk '{print $1}'| uniq | wc -l; done

# Total number uniquely aligning reads

cat 2S1Flag-p5-2_lib_idx5_ACAGTG_OMM.sam | grep -v '^@' | awk '{ if ($3 != "*") print $0}' \
| awk '{print $1}'| uniq -c | awk '{print $1 "\t" $2}' | awk '{ if ($1 == 1) print $0}' | wc -l


for file in *.sam; do echo ${file}; cat ${file} | grep -v '^@' | awk '{ if ($3 != "*") print $0}' \
| awk '{print $1}'| uniq -c | awk '{print $1 "\t" $2}' | awk '{ if ($1 == 1) print $0}' | wc -l; done
```

STEP 3: COMPLETE THE FOLLOWING TABLE USING RESULTS FROM STEP 2 ABOVE

File Name	Total Number of Reads	Uniquely Aligning Reads	% Uniquely Aligning Reads
2S1Flag-p5-2_lib_idx5_ACAGTG_0MM.fq.gz			
2S1Flag-p6-3_lib_idx4_TGACCA_0MM.fq.gz			
2S1-Flag-p7-2_lib_idx2_CGATGT_0MM.fq.gz			
759_7-p5-2_lib_idx6_GCCAAT_0MM.fq.gz			
759_7-p6-1-1_lib_idx12_CTTGTA_0MM.fq.gz			
759_7-p6-2-2_lib_idx7_CAGATC_0MM.fq.gz			
pCDNA_p6-3_lib_idx5_ACAGTG_0MM.fq.gz			
pCDNA_p7-2_lib_idx4_TGACCA_0MM.fq.gz			
pCDNA_p8-3_lib_idx2_CGATGT_0MM.fq.gz			
Scram_1-3_lib_idx6_GCCAAT_0MM.fq.gz			
Scram_1_p3-1_lib_idx12_CTTGTA_0MM.fq.gz			
Scram_1_p3-3_lib_idx7_CAGATC_0MM.fq.gz			