**Mapping RNA-seq reads to reference sequences on Kepler**

(e.g. RNA-seq reads to a genome or transcriptome assembly)

**Overview**

Things to know:

1. What does an aligner/alignment program/alignment algorithm do? What are its inputs and outputs (think of what you gave to blastn and what you got back)?
2. What kind of aligner is a “mapper”?
3. What is a splice-site-aware aligner?
4. Draw a diagram of a splice-site-aware aligner aligning a mRNA sequence read that spans a splice site to a reference genome sequence.
5. The program we’ll be using for mapping reads is HISAT2. Can you figure out how to load the hisat2 module?
6. What is the file format used to store raw sequence data from modern sequencing instruments?

After a successful RNA-seq experiment, one has FASTQ files with sequence reads, which represent fragments of transcripts from the RNA sample (mRNA, rRNA, small RNA, total RNA) that was isolated and sequenced. Reads are often 100 or 150bp in length but can be shorter or longer depending on the sequencing instrument used.

The following four lines are from a FASTQ file of 150bp Apple mRNA-seq reads. These four lines constitute the repeating unit of every FASTQ file:

1. Sequence identifier/description starting with a @ symbol
2. Sequence
3. A plus symbol (+), optionally followed by the same Sequence identifier/description as (1)
4. Quality values for each nucleotide in the Sequence

@HISEQ-MFG:243:HNH2FBCXY:1:1101:2377:1999 1:N:0:TCCGGAGAGGCTCTGA

CCTGTTTGCCGCAATTGTGATTTCAACTAAAGTTGCTGGTGGAACTGCGCCTGCCCCGTTGCATGCGACCTGGCCAGAGCCACAGTCTGCAGTTTCACAAGTGAATTTTCCAGATCGGAAGAGCACACGTCTGAACTCCAGTCACTCCGG

**+**

DBDDDIIHIIIHIHHIHIHHHFHIHHEI?HHFH?GHIHIHIHCHHIHHDHIIIIHEEHIGHIIH<GDIIIIHIHIIIIIIIICHIHC@GH@FC@FH<GCFGGHIIHHHIHIIHHIIEH0EEHHHEHHHIIIIIGHIHIIIIIEEHEHIDH

Here is some more of the FASTQ file:

@HISEQ-MFG:243:HNH2FBCXY:1:1101:2990:1992 1:N:0:TCCGGAGAGGCTCTGA

CCGCTAACCATGTTGACTAATTCTCAGCAGCAGGTATCAACTCAACTCCTAATGAATCAAAGCCAAAGCCTCCCAAATTTGATATAAGGGAAGCCCCAAAGCGAAACCTTCCTTCGCCTCCAAACAACGACACCGACCGCCGACCGGAAT

+

DDDDDIIIIIIIHHHIIHIIIIIIIIIIIIFHHH<<DCGHHGHIIIIIIIIIHIIIIIIIIGIIIIIIIIHIIIIIHIIIIIHIIIIIGGEHIHIIIHHIIHIIIIIIIIIIIIIIIIIIIIIIIIIIIIHIIIIIIIIIHIIIIIIHHI

@HISEQ-MFG:243:HNH2FBCXY:1:1101:10628:1999 1:N:0:TCCGGAGAGGCTCTGA

GTACAGCAGTGGTAACCCGACGAACATTGAAAATCCCATCAAAGATGCCAGTGTTCAGGTTGACATTAAGACAGCGAGTGGAAGATTGACCCTGTATCAAACCACCCTCTGTGAAAAGCTCCAATGGGACAGATCGGAAGAGCACACGTC

+

BDDDDIIIHIIHHIIIIIIIHIIIIIIIIIIIIHIIIIIIIGIIIIIIIIIIIHIIIIIIIIIIIIIHIHIIIIIDHIHIIIIIIIIIIIIIIIIICHIIIHHHHIIIIIIHIIIIGHIIHGIIIHIIHIIIIIIHIIIIIIIIIIIIIG

@HISEQ-MFG:243:HNH2FBCXY:1:1101:13105:1960 1:N:0:TCCGGAGAGGCTCTGA

NCAAAATCACGTGTCCGTGGTTTGCATAAGGAAGGTCACAGTTCTCCAAGATCAAACCGACGAAGCAAGAAGGATTGTCAACCTTTGGAGTACAGAGATTGCGGCGGAGATTTGAAAAGCACCCATCGCATACTATTGAGATCGGAAGAG

+

#<DDDIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIHIIIIIIIIIIIIHIIIIIIIIIIIIIIIIIIIIIIHIIIIIIIIIIIIHIIIIIHHIIIIIIHIIIIIIIIHIIHHIIIIIIIIIIIIIIIIIIIIIIIIIII

@HISEQ-MFG:243:HNH2FBCXY:1:1101:13053:1990 1:N:0:TCCGGAGAGGCTCTGA

NAGACGAGTCTCATATACATCAAGAACCACAGCCAACTTGGCTTCATTTTCCTCCACAACTGCCGCGTCTGTGGTCATTTTAAACATAGGCTTTATAGCTAGCTCATAGGTCAGTTTGTGGCCGCCGGGTCGAACTTTTGGGATCGGAAG

+

#<DDDIIIIIIIIIHIIIIIIIGIIIGIIIIIIIIIIGIIIIIIIIIIIIIIIHIIIIIIIIIIIIIIHIIIIHFHGHIIIIIIIIIIIIIIIIIIIIIIIFHIIIIHHIIIIIIIIIIIHHHIIIHIIHIIIIGIIHIIGIII/DGHII

@HISEQ-MFG:243:HNH2FBCXY:1:1101:13930:1962 1:N:0:TCCGGAGAGGCTCTGA

NTTTTTCCTGCACCATAACCTTTGGTTCCAAAGTCTCCTCTCTCTCTCTCTTTCTCTCTCTTCTCTCTCTCTCTCTCCCTTTGGCAAGAAACAATCACATTTCATATCTCTGTTGGAAGTCCAGTCTGTGTCCAAACAGCATACCCACCA

+

If you want you can look at one of the FASTQ files on the cluster (you can actually use ‘less’ to view gz-compressed plain text files like this one)

~/data/apple-pear/hiseq.huck.psu.edu/170713\_7001126F\_0241\_BHNJ3LBCXY/fastq/Sample\_11/Sample\_11\_R1.fastq.gz

The sequences in the FASTQ file correspond to expressed parts of genes in the genome from which they were derived. Since we know the controlled conditions of the experiments, if we can figure out which genes they are then we can maybe learn something about which genes get expressed under the conditions present.

If you have an annotated genome assembly, that's good, because it means you have most of the complete sequence of the nuclear genome of your study organism, on which genes have been identified and labeled. That means all you have to do is find out if any/which of the RNA-seq reads have exactly or very nearly exactly the same sequence as a gene in the genome assembly, and you will have made a huge step towards understanding how many and which genes were expressed in your organism during your experiment.

Usually, not all of the RNA-seq reads will map to the reference genome. If less reads map, the mapping rate is low, and if more reads map, the mapping rate is high.

Different RNA-seq experiments will yield sets of reads whose mapping rates will probably differ from each other. This can be due to at least two different things:

1. Sequencing errors in the RNA-seq reads
2. Errors in or incompleteness of the reference sequence being used

Sequencing errors are inevitable, so sequencing instruments have been built to determine the level of confidence that the nucleotide being called by the instrument is *actually* the correct nucleotide. They output quality scores (the **blue** line in the FASTQ sample above), which can be used to decide whether or not to keep or throw out low quality sequence.

After doing a RNA-seq experiment, you should summarize the FASTQ quality information. A good way is to visualize the quality information using a program like FASTQC (which is on Kepler). FASTQC was run for the Apple RNA-seq libraries.

* Download and look at FASTQC output from Apple read dirs

Once you have use programs to look at all the RNA-seq reads and automatically remove parts of reads that are poor quality/adaptors.

In this research activity, we will

* make dir for scripts and outputs, set up with links to at least 2 sets of reads
* trim reads using trimmomatic or scythe|seqqs|sickle
* map reads to apple reference genome
* comparing mapping rates
* examine SAM files
* convert from SAM <-> BAM
* next up: running GATK and

1. Make project dir
2. Trim?
3. If there is a compute node with 10 or more processors available, **ssh** onto that node, then navigate to your project directory. If not, copy the qsub template script to your project directory and put the following commands in it. You will also need to configure it with your email, queue choice, working directory and # processors. I reuse my qsub scripts so I don't always have to put my email in. If you want to you could keep your own personalized qsub template in one of your directories for reuse.
4. Load hisat2

module load genomics/hisat2

1. Build a Burrows-Wheeler transformed (BWT) index for the reference sequence. This is a transformation of the characters in the DNA sequence that allows the computer to search very very quickly for any subsequence, like a RNA-seq read. BWT is also used in compression algorithms, like bzip2. Chapter 1. in the book Bioinformatics Algorithms Vol. 2 shows you how to implement your own BWT in any programming language, which you could use to write your own DNA mapper.

This should take about 4 min with 10 processors, and you can just let it run. If you're using a qsub script, add the following command and redirect stdout and stderr to files in your working directory ( >out 2>err )

hisat2-build -p <num\_processors> <reference\_fasta> <reference\_fasta>

1. Map the reads with HISAT2

hisat2 [options]\* -x <hisat2\_bwt\_index>

/home/derstudent/data/apple-pear/hiseq.huck.psu.edu/170713\_7001126F\_0241\_BHNJ3LBCXY/fastq

/home/derstudent/data/apple-pear/hiseq.huck.psu.edu/170720\_7001126F\_0242\_BHNNMNBCXY/fastq

/home/derstudent/data/apple-pear/hiseq.huck.psu.edu/170721\_7001126F\_0243\_AHNH2FBCXY/fastq