Question 1. Coverage Analysis

Download the reads and reference genome from: <https://github.com/bxlab/qbb2020/blob/master/week1/asm.tgz?raw=true>

Note I have provided both paired-end and mate-pairs reads (see included README for details). Make sure to look at all of the reads for the coverage analysis and kmer analysis, as well as in the assembly.

* Question 1a. How long is the reference genome? [Hint: Try samtools faidx]

513 samtools faidx ref.fa

514\*

515 less -S ref.fa.fai

1a - 233806Bp

* Question 1b. How many reads are provided and how long are they? Make sure to measure each file separately [Hint: Try FastQC]

518 fastqc jump2k.1.fq

519 fastqc jump2k.2.fq

520 fastqc frag180.1.fq

521 fastqc frag180.2.fq

1b- frag180.1.fq Reads 35178 Length 100

frag180.1.fq Reads 35178 Length 100

Jump2k2.1.fq Reads 70355 Length 50

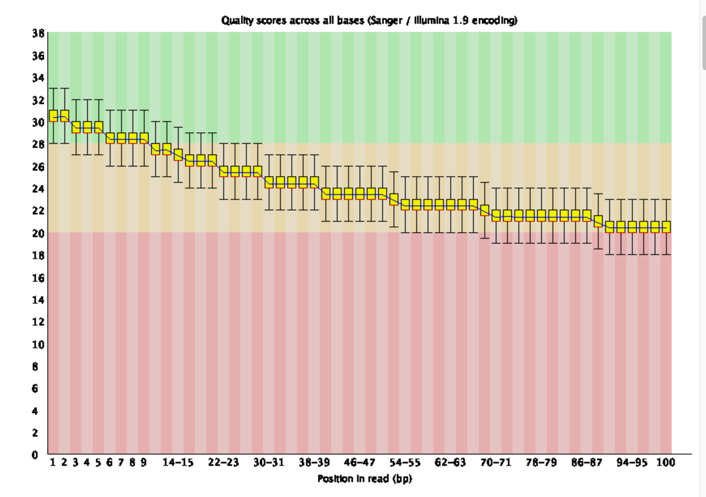
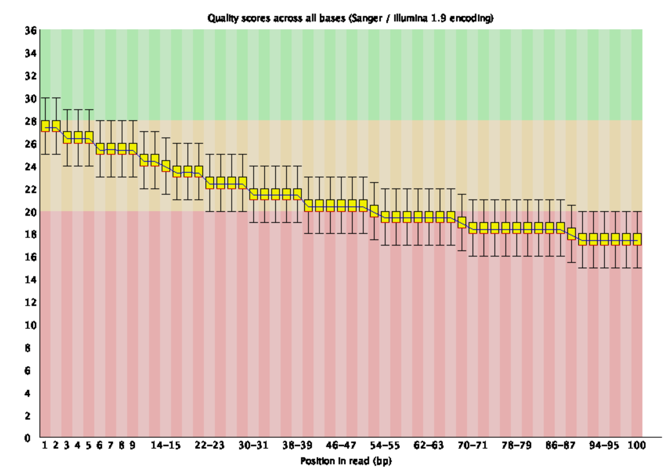
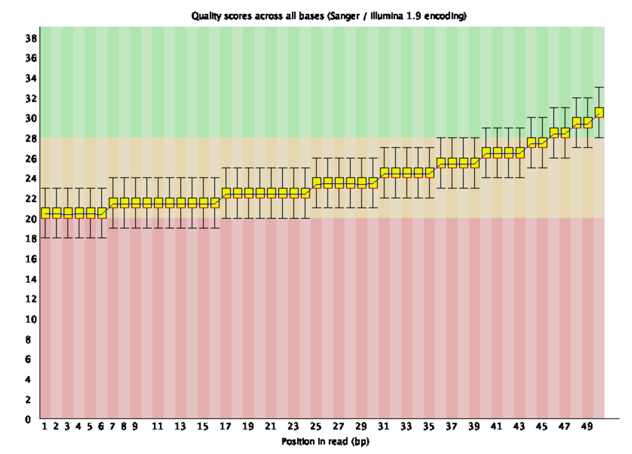
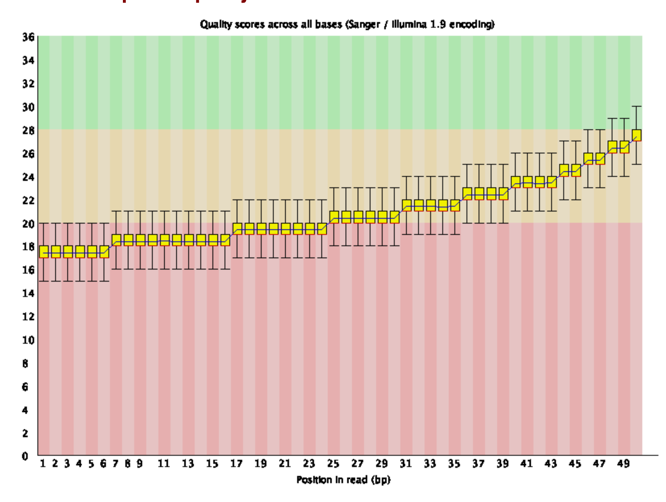
Jump2k2.2.fq Reads 70355 Length 50

* Question 1c. How much coverage do you expect to have? [Hint: A little arthmetic]

Total reads \* length / genome length

Cumulative there are 60.18x coverage

Separately each file type (frag & jump sequence length) have about 30.09x coverage

* Question 1d. Plot the average quality value across the length of the reads [We want a screenshot from FastQC]
* Fag180.1.fq
* 
* Fag180.2.fq
* 
* Jump2k.1.fq
* 
* Jump2k.1.fq
* 

Question 2. Kmer Analysis

Use Jellyfish to count the 21-mers in the reads data. Make sure to use the “-C” flag to count cannonical kmers, otherwise your analysis will not correctly account for the fact that your reads come from either strand of DNA.

* Question 2a. How many kmers occur exactly 50 times? [Hint: try jellyfish histo]

Command 1: jellyfish count -m 21 -s 100M -t 10 -C jump2k.1.fq jump2k.2.fq frag180.1.fq frag180.2.fq

Command 2: jellyfish histo mer\_counts.jf

Answer: 1091 kmer occurances

* Question 2b. What are the top 10 most frequently occurring kmers [Hint: try jellyfish dump along with sort and head]

Command 1: jellyfish dump -c -t mer\_counts.jf | sort -r -g -k 2 | head -n 10 >dump1.txt

GCCCACTAATTAGTGGGCGCC 105

CGCCCACTAATTAGTGGGCGC 104

CCCACTAATTAGTGGGCGCCG 104

ACGGCGCCCACTAATTAGTGG 101

CAGGCCAGCTTATAAGCTGGC 98

AACAGGCCAGCTTATAAGCTG 98

ACAGGCCAGCTTATAAGCTGG 97

AGGCCAGCTTATAAGCTGGCC 95

AGCATCGCCCACATGTGGGCG 83

GCATCGCCCACATGTGGGCGA 82

* Question 2c. What is the estimated genome size based on the kmer frequencies? [Hint: upload the jellyfish histogram to [GenomeScope](http://genomescope.org/) and report the min “Genome Haploid Length” in the “Results” section]

Genome Haploid Length 233,468 bp

* Question 2d. How well does the GenomeScope genome size estimate compare to the reference genome? [Hint: In a sentence or two]

Minimum estimation from kmers is only a few hundred base pairs less than the actual genome length and the maximum is only one bp less than the actual value.

Question 3. De novo assembly

Assemble the reads using Spades. Spades will *not* run on Windows you must use a linux or mac environment.

* Question 3a. How many contigs were produced? [Hint: try grep -c '>' contigs.fasta]

Base command: Make an output directory

Spades frag output

* Command 1: spades.py --pe1-1 frag180.1.fq --pe1-2 frag180.2.fq --mp1-1 jump2k.1.fq --mp1-2 jump2k.2.fq -o asm -t 4 -k 31

Command 2: grep -c '>' contigs.fasta

4 contigs were found

* Question 3b. What is the total length of the contigs? [Hint: try samtools faidx, plus a short script if necessary]
* Add contigs together
* Total length = 234467
* Question 3c. What is the size of your largest contig? [Hint: check samtools faidx plus sort -n]
* Largest contig = 105831Bp
* 1st - 105831Bp
* 2nd - 47861Bp
* 3rd - 41352Bp
* 4th - 39423Bp

Samtools

* Question 3d. What is the contig N50 size? [Hint: Write a short script if necessary]
* Manually calculated by finding the contig containing the the median value bp
* N50: 47861Bp

Question 4. Whole Genome Alignment

Use MUMmer for whole genome alignment.

* Question 4a. What is the average identify of your assembly compared to the reference? [Hint: try dnadiff]
* [~/QUANT2020/data/lab1/asm] dnadiff ref.fa asm/contigs.fasta
* 100%! This makes sense considering our high level of coverage.
* Question 4b. What is the length of the longest alignment [Hint: try nucmer and show-coords]
* [~/QUANT2020/data/lab1/output]nucmer ../asm/ref.fa ../asm/asm/contigs.fasta
* [~/QUANT2020/data/lab1/output]show-coords out.delta
* 105831 (is the max alignment length)
* Question 4c. How many insertions and deletions are in the assembly? [Hint: try dnadiff]
* One insertion is present in the query sequence
* Five deletions are present in the query sequence

Question 5. Decoding the insertion

We need you to wget an updated script. Please wget https://raw.githubusercontent.com/bxlab/qbb2020/master/week1/ported\_decoder.py now.

* Question 5a. What is the position of the insertion in your assembly? Provide the corresponding position in the reference. [Hint: try show-coords]
* 13853
* Question 5b. How long is the novel insertion? [Hint: try show-coords]
* 712 bp
* Question 5c. What is the DNA sequence of the encoded message? [Hint: try samtools faidx to extract the insertion]

[~/QUANT2020/data/lab1/output]samtools faidx ../asm/asm/contigs.fasta NODE\_3\_length\_41352\_cov\_20.588756:13854-14565 > basepair.fasta CTAACGATTTACATCGGGAAAGCTTAATGCAATTCACGCAGATATTCAGCTTAGAAGGTACGCAGCGGTGACGGGGTGCGGTCCATAATCTATGAAGCTATGAATTCGTACCTCAAGTAATGTTTTCTTCGCTGCAGTTCAGAAGTGATAAAGGTATCCCGCTTAGCCTGGCATACTTTGTGCGTTCGTACCGCCCAGCATTAATGACTTGTGTAGGCAAGTAATGAACGACTCTTCTACGCCGCGCCTAACCTCCGCACATAATGGCAGCATGTGGTAGTTACATACGCACAGAAGTGGTTCGGTTTTAACTATAGTCAGATATGAATAAGCTGCGTGTGTCGTTGTGTCGGCGTGTCGTACTTACCTCCTGACATAGGTGAATTTCAGCCTACTGTAAGTTTGGAGTCGCGCTCTTTTCTTATTATATTCTTTGGTATGTGTGTGATGGGTTCGGGCGTGTATTGATGTCTCTAAGGCTCATGTTAGTGTTTATTTGGTCAGTTATGACGGTGTTCCTGTCGTACGTGTTGGCTTAGCGGACTTGTAGACGGGATCAAGGTTGTCTGACCCTCCGGTCGACCGTGGGTCGGCCGTCCCGGCCAGAATACAAGCCGCTTAGACTTTCGAAAGAGGGTAAGTTACTACGCGCGAACGTTA

* TACCTCGTTTCAGTATGCACTCCCTTAAGTCACTCAGAAAAGACTAAGGGGCT
* Question 5d. What is the secret message? [Hint: Run the provided script ported\_decoder.py to decode the string from 5c.]
* [~/QUANT2020/data/lab1/output]./ported\_decoder.py -d --rev\_comp --input basepair.fasta
* The decoded message : Congratulations to the 2020 CMDB @ JHU class! Keep on looking for little green aliens...

Submissions to Github

The solutions to the above questions should be submitted as a markdown or text file on Github, in your qbb2020-answers repo. Include any requested figures within the markdown file, or push them separately to Github. Make sure to clearly label each of the subproblems and give the exact commands you used for solving the question.