Module 02: NGS and Genome Assembly

Sequencing and assembling a genome was once the domain of large, well-funded organizations. Early draft genomes were sequenced and assembled with budgets in the tens of millions of dollars, and draft genomes published prior to 2009 were primarily sequenced using Sanger sequencing. While Sanger sequencing produces quality data, the cost puts it out of reach for most small or medium-sized labs. In 2009 several labs began publishing draft genomes based on NGS sequencing technology. With the cost of sequencing a genome now approaching $10,000, it's not unusual for a bioinformatician to take on the task of assembling a draft genome.

Assembling a draft genome requires communication and coordination between the biologist extracting and preparing the DNA and the bioinformatician assembling the genome. There are many decisions that need to be made regarding DNA extraction, fragmentation, library preparation, and sequencing that affect the assembly process. A genome project that lacks communication between biologists and bioinformaticians, with sequence data being obtained without regard to the assembly method, is unlikely to produce good results. In this module you'll learn the high-level steps from DNA extraction to genome assembly so you can participate in and contribute to the planning of a genome project.

Learning Objectives

After completing this module you'll be able to:

- Choose a sequencing strategy and assembly program
- Calculate the number of lanes required for a sequencing experiment given a target coverage level and estimated genome size
- Estimate the cost of a sequencing experiment given the approximate genome size and target coverage level
- Assemble a bacterial genome

Required Reading

The text of this module gives an overview of the material in the papers, so read the module text first, then read the papers. The quiz for this module includes questions from these papers. The papers have been loaded into the Blackboard learning module, so as you page through the learning module the papers will be displayed. From Blackboard you can right-click to print or download the PDFs of these papers.

- Sequencing technologies — the next generation
- Sequence assembly demystified
- GAGE: a critical evaluation of genome assemblies and assembly algorithms

- High-quality draft assemblies of mammalian genomes from massively parallel sequence data.
- SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler

DNA Prep, Library Prep, Sequencing, and Assembly

At a high level, the steps in a sequencing and assembly project are:

- DNA Extraction
- Fragmentation
- Library Prep
- Sequencing
- Error correction
- Assembly

DNA extraction requires lysis of cell membranes, removal of membrane lipids, removal of proteins and RNA, and precipitation of DNA. Fragmentation breaks DNA molecules into fragments within the size range required by the sequencer. Library prep attaches adapters to DNA fragments so they will attach to the surface of the sequencer flow cell. If multiple samples are to be sequenced together, during library prep short index sequences, usually 6 bases, are attached so the individual samples can be identified (demultiplexed) from the sequence reads. Libraries are loaded into flow cells and sequenced, producing short reads of about 150 bases. Error correction programs compare reads using k-mer frequencies and other attributes. Reads with bases that are inconsistent with expected k-mer frequencies are then corrected. Some assemblers, like ALLPATHS-LG, have built-in error correction. After error correction the short reads are then assembled into contigs, contigs are assembled into scaffolds, and scaffolds are assembled into chromosomes.

Read Length

Read length is one of the main differences between the output of Sanger sequencing and NGS. Sanger sequencing produces reads of about 700 bases. The shorter reads of NGS add complexity to the assembly process, and assembly is further complicated by repeat regions in genomes. A repeat region that’s longer than the read length can’t be resolved by short reads without some method to determine the number of occurrences of that repeat.

Library prep and assembly strategies have been devised to overcome the limitations of short reads. Paired-end libraries are read up to the sequencer length limit from both ends of the DNA fragment. The advantage of this approach is that the two ends can either overlap and be combined into a “super-read”, or be separated by an insert that falls in a narrow range of lengths. The assembler can use this information to help determine the location of the two ends in a contig. Since paired-end reads are sequenced from the ends toward the middle of the fragment, these may also be referred to as inward-pairs.

Mate-pair libraries are similar to paired-end libraries, but the insert size is much longer - usually 3-10 kb . Fosmid libraries are similar to mate-pair libraries, but the insert size can be up to 40 kb. Library prep for mate-pair and fosmid libraries results in these fragments being read from the middle out, so they may also be referred to as outward-pairs.

Knowing that two ends of a long DNA sequence should align within a specific range of distances from each other helps assemble contigs into scaffolds.

Choosing an Assembler

Sequencing technology is changing rapidly, and assemblers are frequently updated to take advantage of new sequencer features. This module provides a snapshot of sequencer and assembler technology, but before starting a genome project you should read the latest papers regarding both sequencers and assemblers. Assembly competitions like Genome Assembly Gold Standard Evaluation (GAGE) and the Assemblathon provide objective comparisons to help choose an assembler. Assemblers vary in terms of computational and library requirements so become familiar with the computational resources available and take that into account when choosing an assembler. Library requirements affect the library prep and sequencing costs, so take library requirements into account as well.

Assembler Metrics

There are a number of metrics by which assemblies can be evaluated. One of the most common is the N50. N50 can be applied to both contigs and scaffolds, and it’s the length at which 50% of the entire assembly is contained within contigs of that length or longer. When a reference genome is available to check the assembly, error rates and an adjusted N50 that takes errors into account can be calculated. Since N50 can be increased at the expense of assembly accuracy, it shouldn't be the sole measure of assembly quality. *Adjusted* N50 based on comparison to a reference genome is much more useful because misassemblies are accounted for, but adjusted N50 isn't available for the first assembly of an organism's genome.

Computational Requirements

Competitions like GAGE and Assemblathon usually provide information on assembler memory and CPU use. Memory and CPU use vary widely depending on the size and number of repeats in a genome, but the reference genome of the closest phylogenetic relative to the organism you’re sequencing can at least give a rough measure. When new genomes are published, the authors often include computational requirements of the assembly. Vertebrate genomes can take 3 weeks or more to assemble even on a high-memory computing cluster, so estimating the run-time is important. You want to know what to expect so you can coordinate computing cluster use based on the expected run-time. If an assembly has been running far longer than the estimated run-time, at some point you have to conclude something is wrong with the assembly process, kill the process, and diagnose the problem.

Library Requirements

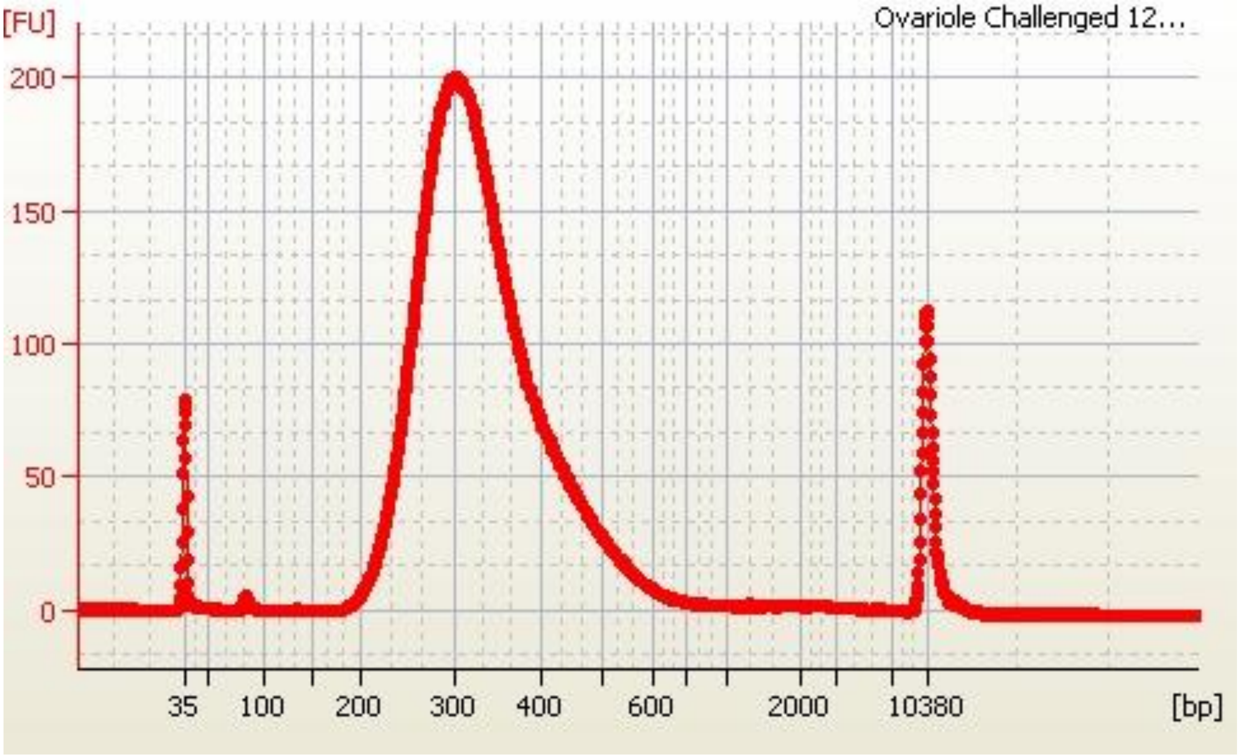
Most assemblers that are capable of assembling large genomes require at least one paired-end library and one mate-pair library. Additional libraries with a variety of insert sizes can help resolve repeats and increase the contig and scaffold lengths. Library preparation cost varies depending on the library type and insert length. Paired-end libraries are generally the easiest and least expensive to make. Mate-pair libraries up to 10,000 bases are next in terms of cost and difficulty, and fosmid libraries between 10,000 and 40,000 bases are the most difficult. Most Fosmid library preparation methods require the use of a bacterial vector.

Paired-End Library

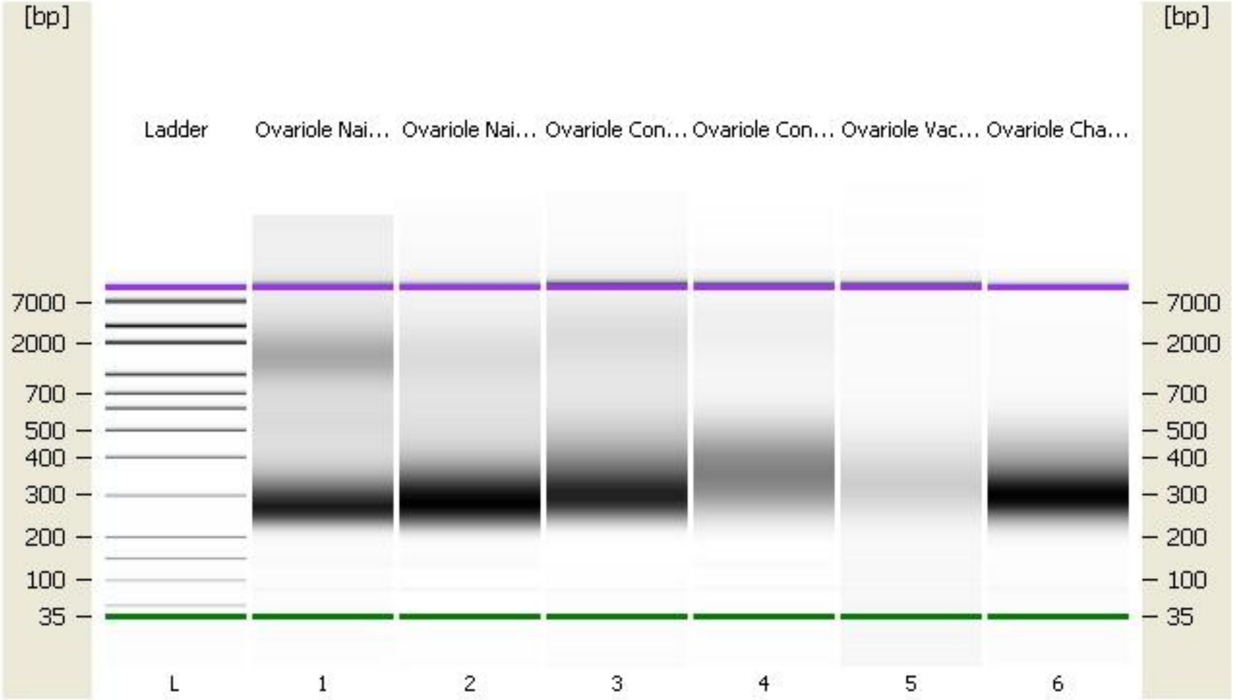
A paired-end library is produced by shearing DNA to a target length, repairing the ends, and ligating sequencer-specific adapters. The target length is usually between 1.8 times the sequencer read length and 1000 bases. Shearing methods vary in terms of fragment length variability. Ultrasonic shearers provide the narrowest range of fragment sizes. Ultrasonic shearers are very expensive, so some labs pay a fee to another institution to either perform the shearing or to allow use of their shearing equipment. DNA can be fragmented using a relatively low-cost nebulizer, but there’s usually more fragment length variation with a nebulizer. Nebulizers also require larger or more concentrated DNA samples; so scarce DNA samples are generally sheared ultrasonically.

Depending on the tolerance of the assembler for fragment length variation, and the precision of the DNA shearing method, an additional size selection step may be needed to isolate fragments that fall within a narrow size range. Fragment sizes should be checked with an automated analyzer like the Agilent 2100 Bioanalyzer.

The Bioanalyzer produces a graph showing the distribution of fragment sizes, and a table of DNA concentrations by size. Ideally, this graph will show a single peak centered around the target fragment size. The peaks to the left and right are markers of known DNA fragment size and concentration included with every sample. The example shown below is after all library prep steps have been completed, and adapters have been attached, so the peak at around 300 bp is about 100 bp larger than the original DNA fragments.



The Bioanalyzer report also shows the equivalent as a gel image for each sample.



Mate-Pair Library

Mate-pair libraries are produced by fragmenting DNA to about 2-10 kilobases, performing size selection on the fragments, circularizing the fragments by joining the ends, fragmenting the circularized DNA to 400-600 bases, then selecting the fragments that contain the joined ends. Depending on the complexity of the genome being assembled, several mate-pair libraries with various insert sizes may be needed. As the insert size of a mate-pair library increases, size selection and circularization become more difficult, making 10kb the upper limit for insert size. As with paired-end libraries, fragment sizes should be checked with an automated analyzer like the Agilent 2100 Bioanalyzer.

Fosmid Library

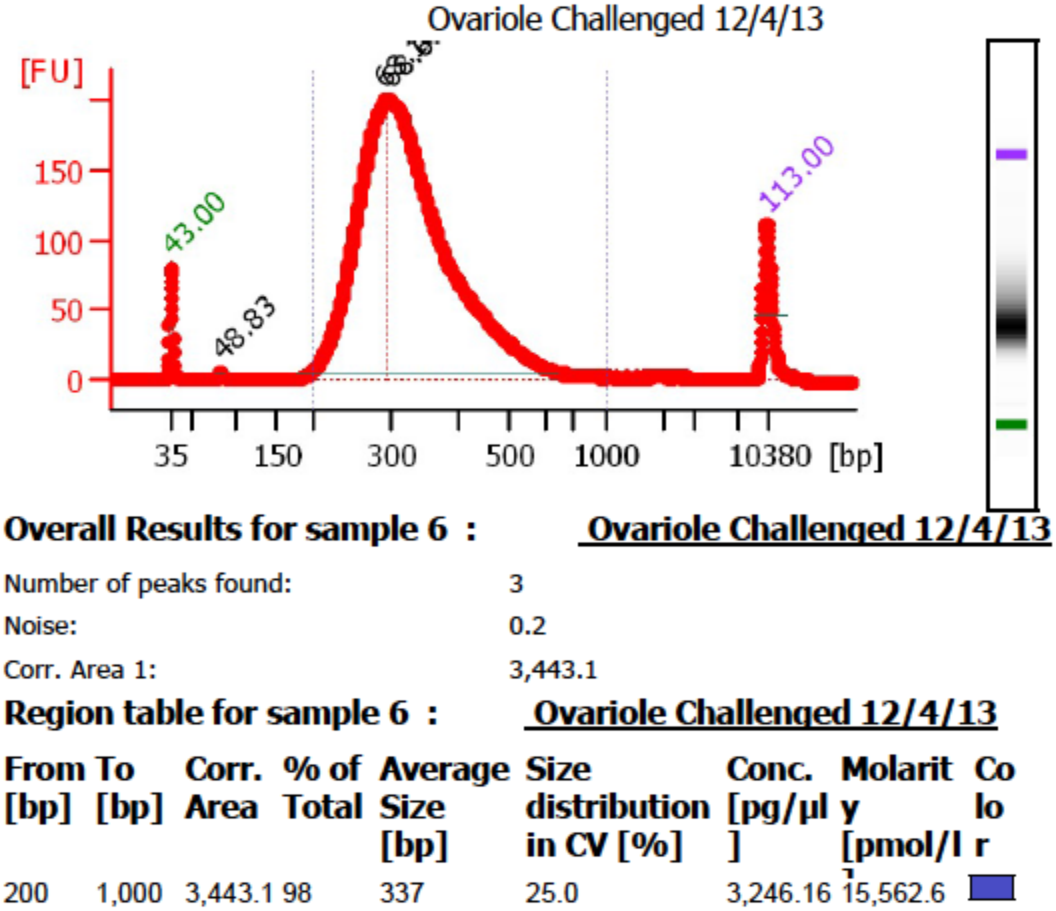
A fosmid library is a mate-pair library with an insert of up to 40 kb. Beyond about 10kb, producing mate-pair libraries by circularization becomes impractical, and transfection into a bacterial vector is required. The name fosmid comes from the bacterial F-plasmid on which the library prep method is based.

Flow Cells and Lanes

Sequencers automate the biochemical sequencing steps on flow cells. Each flow cell has multiple lanes, each of which can be used for a different library or combination of libraries. The decision regarding the number of lanes to use, and the combination of libraries to place on each lane depends on library compatibility and the desired read depth. Read depth is the number of times you want each base sequenced. Numerous overlapping fragments over a given base are used to increase confidence that the bases called and the fragments assembled are correct. Sequencing centers generally price services based on an entire flow cell, or a specific number of lanes on a flow cell.

Quantitation

The DNA concentration of libraries to be loaded on a flow cell must fall within a fairly narrow range to be sequenced successfully, and that concentration is usually specified in terms of molarity. The Bioanalyzer will provide the molarity for your library. For the Bioanalyzer graph shown previously, the region table and concentrations looks like this:



Overall Results for sample 6 : **Ovariole Challenged 12/4/13**

Number of peaks found: 3

Noise: 0.2

Corr. Area 1: 3,443.1

Region table for sample 6 : **Ovariole Challenged 12/4/13**

From [bp]	To [bp]	Corr. Area	% of Total	Average Size [bp]	Size distribution in CV [%]	Conc. [pg/µl]	Molarity [pmol/l]	Color
200	1,000	3,443.1	98	337	25.0	3,246.16	15,562.6	🟦

Sequencer Output

Sequencer output is usually in the FASTQ file format. FASTQ files are similar to FASTA files, but they include quality scores for each base read. These quality scores are used to exclude reads below a certain quality threshold, and some assemblers use the scores to determine the assembly order. A single read in a FASTQ file looks like this:

```
@ILLUMINA-D00365:233:H9N3UADXX:2:1101:2960:2261 1:N:0:GGCTAC
CTACGAGTTTGAGGAATTTCTTGCCGAAGTAGATCGGAAGAGCACACGTCTGAACTCCAGTCACGGCTACATCTCGT
ATGCCGTCTTCTGCTTGAAAAAAAAGAAAAATATAAACGACGTATTGCACCAATGAATCATACAATTTCGACT
+
@;@BAA0BDCBFFFE?FGHHIIIGCE@G@DD>GC>CAHGHEF;BFF>@FGIGGIADGIHHH==?/999==5(;>>=8
?A<>@:39@B@A>AC>@84>@?#####################################################
```

Assemblers, quality-trimming utilities, and correction utilities read and use the quality information associated with each read.

NGS Technology

While there are several NGS sequencers, recent high-profile successes with de novo genome assembly using short reads have been based on the Illumina platform. The reading assignments gave an overview of Illumina as well as other NGS platforms. Most of this section focuses on the Illumina platform. De novo assemblies of the panda, turkey, and Japanese eel have all been based on the Illumina platform. Mammalian genomes have been re-sequenced using Illumina technology as a proof-of-concept for assembly of complex genomes from short reads. As with assemblers, it's important to remain current by reading the latest publications of draft genomes. While Illumina may be in the lead in 2014, new leaders may emerge by the time you work on a sequencing project in a professional setting. Before starting a sequencing experiment, the first step is a search of the most recent publications to see if there are new techniques you may want to incorporate into your own experiment.

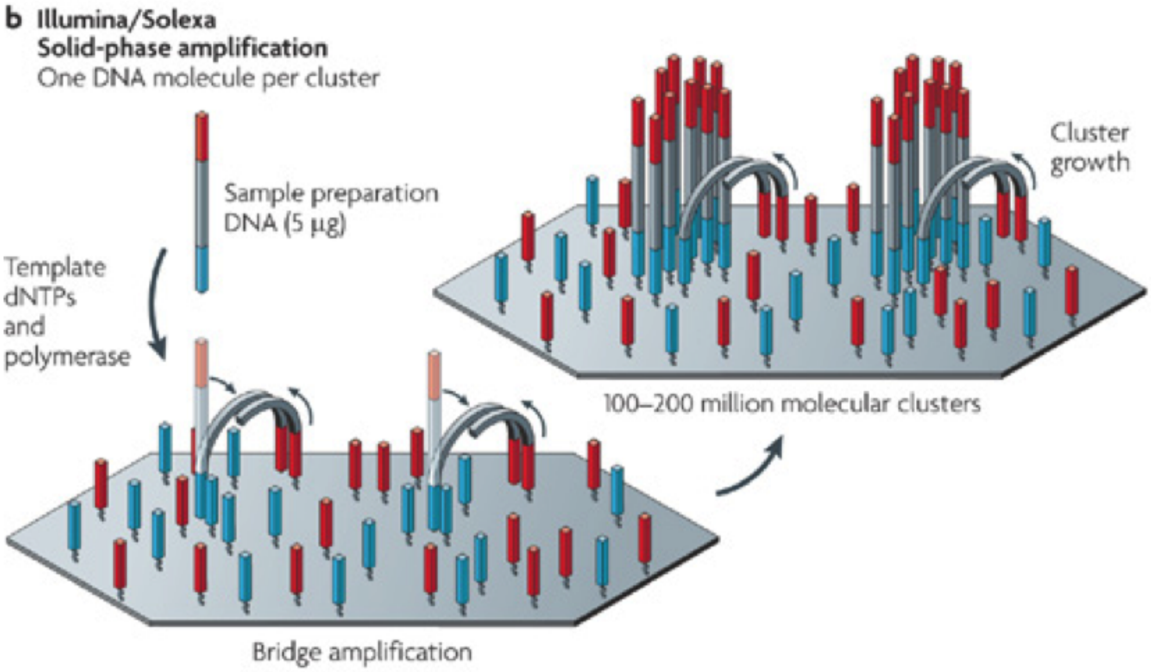
Illumina Sequencing Technology

The latest generation of Illumina sequencers is the HiSeq 2500, which supports read lengths of up to 150 nucleotides. The high-level steps in the sequencing process are in-situ cluster amplification followed by sequencing using cyclic reversible terminators. The output of the sequencing process comes in the form of FASTQ files, which contain both the sequence information and the associated quality information for the reads. The video below provides an overview of the Illumina sequencing process.

http://www.youtube.com/embed/womKfikWlxM

In-Situ Cluster Amplification

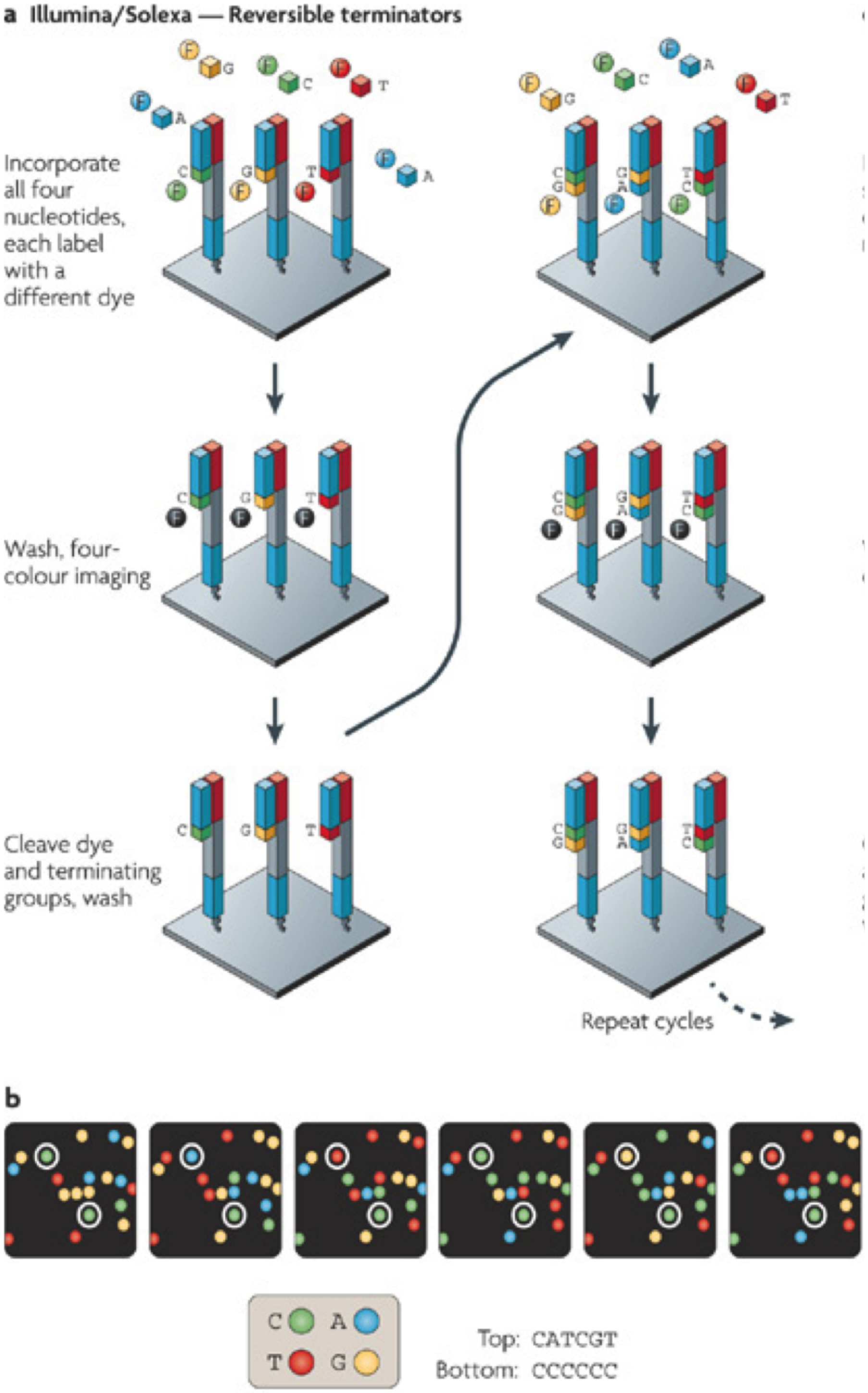
The sensors that detect the presence of fluorescently labeled nucleotides aren't sensitive enough to detect the presence of a single nucleotide, so the DNA fragments to be sequenced first have to be copied so there are many identical nucleotides being added at a time. Primer molecules are attached to a solid support within the Illumina flow cell. These primers are complementary to the adapters that were ligated to the DNA fragments during library prep. The image below illustrates this amplification step.



Sequencing technologies — the next generation, Michael L. Metzker, Nature Reviews Genetics 11, 31-46 (January 2010), doi: 10.1038/nrg2626. Figure 1b.

Cyclic Reversible Termination

After the fragments have been amplified, sequencing is performed by repeatedly adding fluorescently labeled terminating nucleotides, detecting the fluorescent signal, then removing the dye and terminator. The terminator prevents the addition of multiple nucleotides in a single cycle. For a read length of 100 nucleotides, this process would be repeated 100 times. For a read length of 150 nucleotides, it would be repeated 150 times. After each cycle, the four fluorescent colors that differentiate nucleotides are read from the flow cell. Since the clusters of DNA strands are held in a fixed position on the flow cell, each position can be tracked individually by the sequencer. The image below illustrates this process.


a Illumina/Solexa — Reversible terminators
Incorporate all four nucleotides, each label with a different dye
Wash, four-colour imaging
Cleave dye and terminating groups, wash
Repeat cycles
F
A
C
G
T
b
C
A
T
G
Top: CATCGT
Bottom: CCCCCC

Sequencing technologies — the next generation, Michael L. Metzker, Nature Reviews Genetics 11, 31-46 (January 2010), doi: 10.1038/nrg2626. Figure 2a.

When any one of the steps fails – addition of the nucleotide, detection of the fluorescent signal, or removal of the dye and terminator – that strand gets out-of-sync with other strands in the cluster. This is referred to as de-phasing, and as de-phasing increases, the signal-to-noise ratio decreases. De-phasing is one of the sources of errors, and one of the factors that limits the read length of the sequencer.

Coverage

To compensate for errors and weak signals in reads, multiple observations per base are required to reliably call bases. Multiple observations are also required to ensure sufficient overlap between reads to allow assembly into contigs. The number of observations per base is referred to as coverage or sequencing depth. Based on the assembler used and the complexity of the genome being sequenced, there's generally a coverage target. For example the mammalian genomes in the Gnerre paper had a target of 100x coverage. Illumina provides an Excel spreadsheet to allow you calculate the number of lanes you'll need to sequence a genome of a specific size to a specific read depth. You'll use this spreadsheet in the lab for this module to estimate lanes for a sequencing experiment.

Multiplexing

In addition to the adapters ligated to each DNA fragment to affix it to the flow cell, Illumina allows additional sequences, called "bar codes", to be ligated to the fragments. This allows multiple samples to be combined on a flow cell. After sequencing, the data from the samples can be differentiated based on the sample-specific bar codes. The main limitations to the number of samples that can be multiplexed on a single lane are coverage and sample compatibility. You may be able to combine more than one bacterial genome in a flow cell, while multiple flow cells may be required for a single sample for a large vertebrate genome. Assemblers generally have different coverage targets for paired-end and mate-pair libraries, so your sequencing experiment may require an entire lane for paired-end, while only requiring a fraction of a lane for mate-pairs. For an experiment like this, it would make sense to sequence the mate-pair libraries for multiple sequencing experiments in a single lane while having dedicated lanes for each experiment's paired-end libraries.

Cost

To get an idea of the cost of a sequencing experiment, review this price list from Harvard's FAS Center for Systems Biology. Review the sequencing tab and review the library prep fees listed near the bottom of the page.

http://bauercore.fas.harvard.edu/fees

To calculate lane requirements for a sequencing project, use this online calculator:

http://support.illumina.com/downloads/sequencing_coverage_calculator.html

Lab Assignment

Note: In this and all other course assignments, you are expected to use your knowledge of Linux commands, directory structure, the assembler documentation, and common sense to figure things out when you run into problems. If you get an error, think about why you may be getting that error and figure out what needs to change. If you think there are mistakes in the procedures, figure out how to fix them then let me know what you had to change. This is an essential skill in Bioinformatics. If you are only able to type commands exactly as shown and get stuck when you hit an error, you won't get far in your Bioinformatics career.

Part 1 - Assembly using ALLPATHS-LG

In your home directory, create a BINF6309 directory. Within BINF6309, create a Module02 directory. Create this locally and rsync to the server each time you add or change files.

In this assignment you'll assemble genomes using ALLPATHS-LG. This assembler performed well in the GAGE competition, and it's been used to publish several draft genomes using NGS short reads. If your library prep and sequencing budget allows for both paired-end and mate-pair libraries, ALLPATHS-LG is probably the best choice. Genome assembly requires more memory than is available on desktop and notebook computers so this and all other assembly assignments need to be run on a server. As always, the programs you need to run on the server are already installed there, so DO NOT try to install ALLPATHS or anything else on the server.

To assemble the genome you'll need to know the all the library specifications including fragment size, fragment size standard deviation, insert size, and library type. These values are either added to configuration files or input as parameters to the assembler. In the case of ALLPATHS-LG, these values are stored in two csv files:

- in_groups.csv
- in_libs.csv

To make sure ALLPATHS-LG is in your path, run this command on the server:

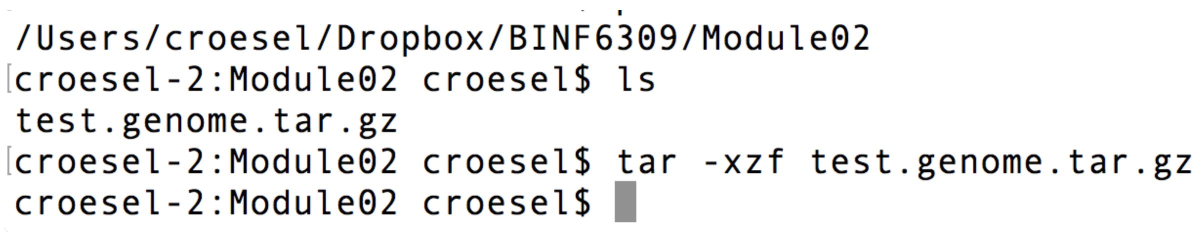
```
RunAllPathsLG –h
```

You should see help with all the parameter descriptions for the assembler.

You'll be assembling a test genome that you can download from the Broad Institute FTP site:

http://software.broadinstitute.org/allpaths-lg/blog/?p=121

Unzip this file in your BINF6309/Module02 directory and examine the directory contents.



```
/Users/croesel/Dropbox/BINF6309/Module02
[croesel-2:Module02 croesel$ ls
test.genome.tar.gz
[croesel-2:Module02 croesel$ tar -xzf test.genome.tar.gz
croesel-2:Module02 croesel$ 
```

The test reads and configuration are in

```
ALLPATHS-LG.test_genome
```

Navigate to the test genome directory and use less to view the contents of in_groups.csv, in_libs.csv, prepare.sh, and assemble.sh. Look up the parameters within those files in the ALLPATHS manual and be sure you understand them.

Run `prepare.sh` then run `assemble.sh`. You'll need to run them as shown below so they run in the background:

```
./prepare.sh &>prepare.log&
./assemble.sh &>assemble.log&
```

Until now, you've probably been running programs in the foreground, but with programs that run for more than a few minutes, you should start running them in the background. If you are running a program in the foreground, and your terminal session times out or disconnects, your program will stop running. If the program is running in the background, you can disconnect from the server, and your program will continue running. You can log back in and check progress later. The first `&` in the commands above says redirect all output. The log filenames `prepare.log` and `assemble.log` indicate where to write the output. `&` at the end says run the program in the background.

Right after you start `assemble.sh`, logout, log back in, then run the top command to see the CPU and memory use.

Run the `tail` command on prepare.log and assemble.log to see the output. To monitor output as it's written to `assemble.log`, use tail with the `-f` option:

```
tail -f assemble.log
```

The output of `assemble.sh` will include statistics from the assembly.

These stats will include:

- contig minimum size for reporting
- number of contigs
- number of contigs per Mb
- number of scaffolds

- total contig length
- total scaffold length, with gaps
- N50 contig size in kb
- N50 scaffold size in kb
- N50 scaffold size in kb, with gaps
- number of scaffolds per Mb
- median size of gaps in scaffolds
- Standard deviation of gaps in scaffolds
- % of bases in captured gaps
- % of bases in negative gaps
- % of ambiguous bases
- ambiguities per 10,000 bases

You'll find the final assembly in:

```
ALLPATHS-
LG.test_genome/test.genome/data/run/ASSEMBLIES/test/final.assembly.fasta
```

Note: The path is all on one line, but due to the length it wraps in this document.

You can check the assembly using `assemblathon_stats.pl`:

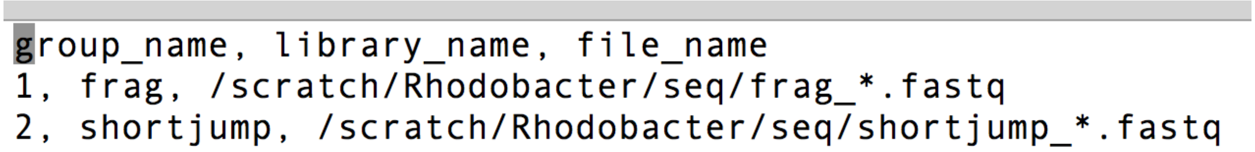
```
./assemblathon_stats.pl\
 ALLPATHS-
LG.test_genome/test.genome/data/run/ASSEMBLIES/test/final.assembly.fasta
```

Now you'll assemble Illumina reads for *Rhodobacter sphaeroides* using the configuration files of the test genome as a template.

Create a `Rhodobacter` directory within your `BINF6309/Module02` directory.

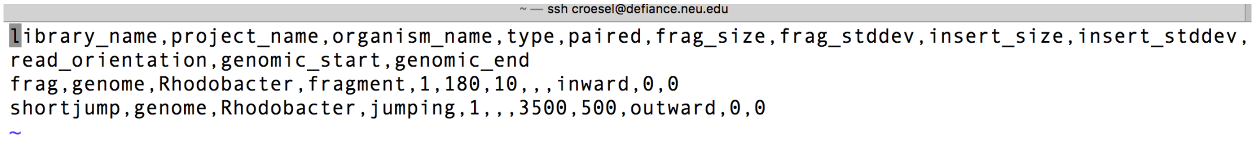
Copy the `.csv` and `.sh` files from your `test.genome` directory into your `Rhodobacter` directory.

Edit `in_groups.csv` to reflect the filenames for the *R. sphaeroides* files as shown below. Just change text, keeping all commas from the original file. This file provides ALLPATHS-LG with the naming conventions for the files containing the paired-end fragment sequences and the mate-pair shortjump sequences.



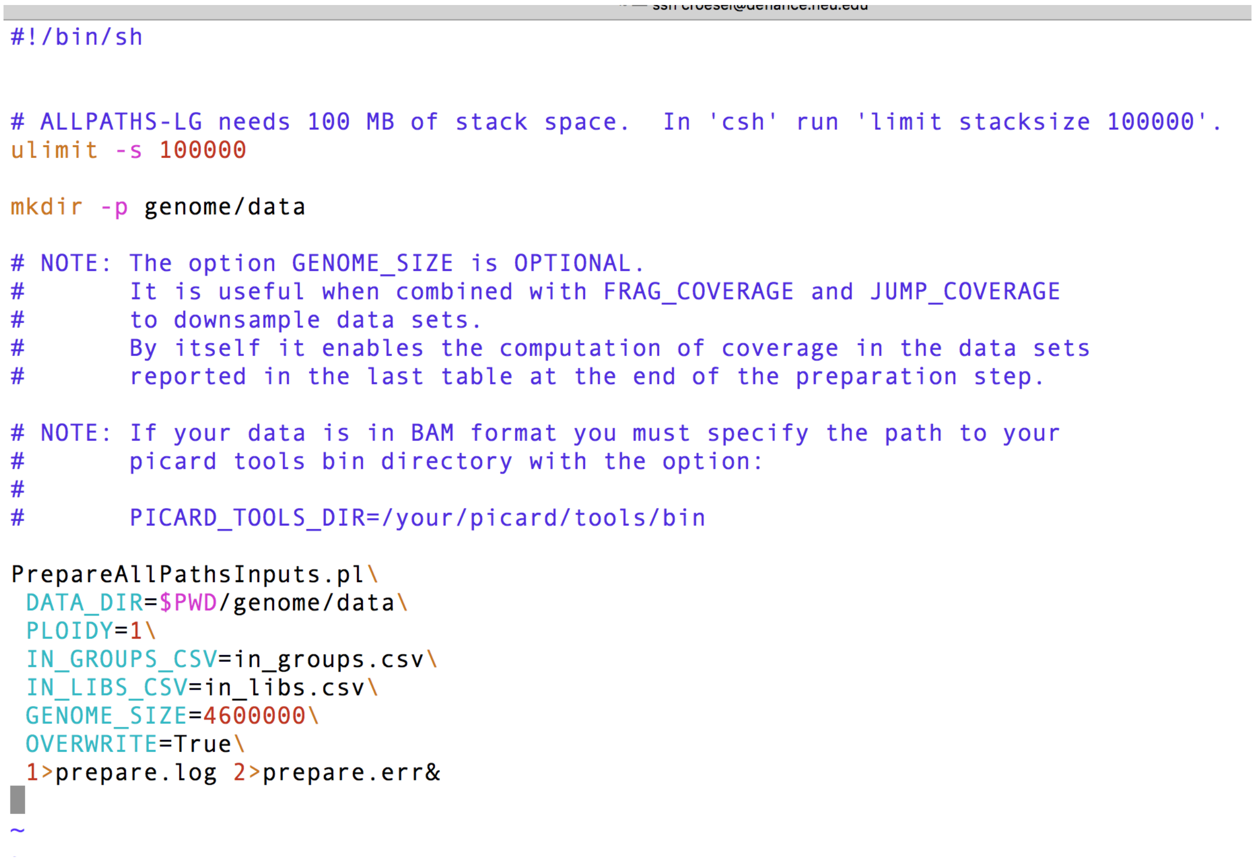
```
group_name, library_name, file_name
1, frag, /scratch/Rhodobacter/seq/frag_*.fastq
2, shortjump, /scratch/Rhodobacter/seq/shortjump_*.fastq
```

Edit `in_libs.csv` to reflect the read lengths and insert sizes for the Rhodobacter files as shown below. Just change text, keeping all commas from the original file.



```
library_name,project_name,organism_name,type,paired,frag_size,frag_stddev,insert_size,insert_stddev,
read_orientation,genomic_start,genomic_end
frag,genome,Rhodobacter,fragment,1,180,10,,,inward,0,0
shortjump,genome,Rhodobacter,jumping,1,,,3500,500,outward,0,0
~
```

Edit `prepare.sh` to reflect the genome size (4.6 mb) and output directory of the Rhodobacter genome:

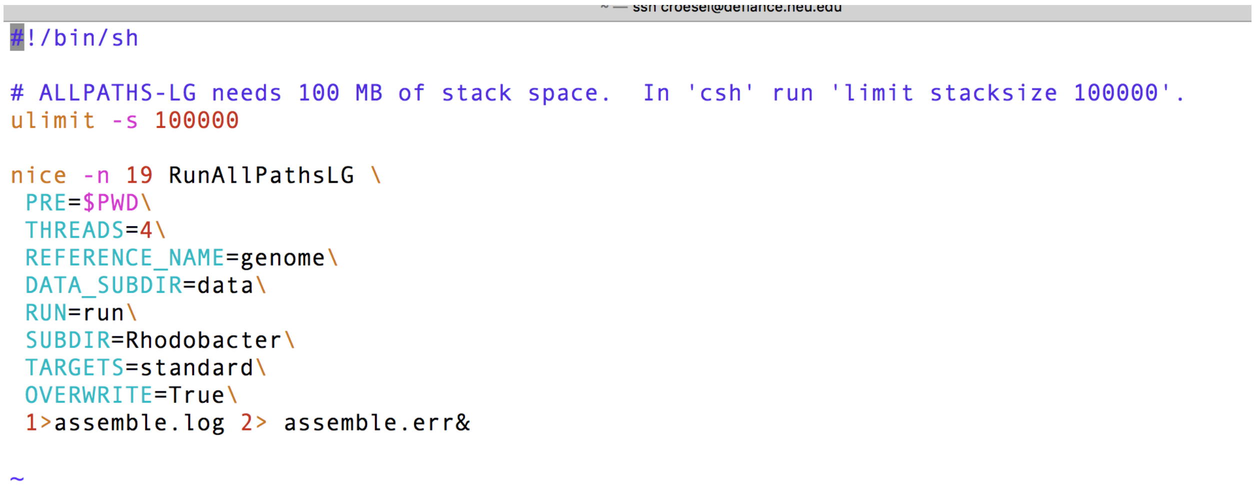


```
#!/bin/sh


# ALLPATHS-LG needs 100 MB of stack space.  In 'csh' run 'limit stacksize 100000'.
ulimit -s 100000

mkdir -p genome/data

# NOTE: The option GENOME_SIZE is OPTIONAL.
#       It is useful when combined with FRAG_COVERAGE and JUMP_COVERAGE
#       to downsample data sets.
#       By itself it enables the computation of coverage in the data sets
#       reported in the last table at the end of the preparation step.

# NOTE: If your data is in BAM format you must specify the path to your
#       picard tools bin directory with the option:
#
#       PICARD_TOOLS_DIR=/your/picard/tools/bin

PrepareAllPathsInputs.pl\
 DATA_DIR=$PWD/genome/data\
 PLOIDY=1\
 IN_GROUPS_CSV=in_groups.csv\
 IN_LIBS_CSV=in_libs.csv\
 GENOME_SIZE=4600000\
 OVERWRITE=True\
 1>prepare.log 2>prepare.err&

~
```

Run prepare.sh:

```
./prepare.sh &>prepare.log&
```

Edit assemble.sh as shown below.



```
#!/bin/sh

# ALLPATHS-LG needs 100 MB of stack space.  In 'csh' run 'limit stacksize 100000'.
ulimit -s 100000

nice -n 19 RunAllPathsLG \
 PRE=$PWD\
 THREADS=4\
 REFERENCE_NAME=genome\
 DATA_SUBDIR=data\
 RUN=run\
 SUBDIR=Rhodobacter\
 TARGETS=standard\
 OVERWRITE=True\
 1>assemble.log 2> assemble.err&

~
```

Run assemble.sh:

```
./assemble.sh &>assemble.log&
```

Check the progress periodically by running top, and by running tail on assemble.log. Run `assemblathon_stats.pl` as you did for TestGenome. Just change the path to your Rhodobacter assembly.

```
./assemblathon_stats.pl \
Rhodobacter/test.genome/data/run/ASSEMBLIES/test/final.assembly.fasta
```

You should get approximately the same results as the GAGE competition, but there may be slight differences because ALLPATHS-LG has been updated since the GAGE competition. Let me know when you've completed the assembly, and I'll check the results on the server. Since I'll be checking the results on the server you don't need to submit anything via Blackboard for this assignment.

If you don't have mate-pair libraries, you can't use ALLPATHS-LG for assembly, but you can use ALLPATHS-LG error correction and assemble the corrected reads in another assembler. The corrected reads used in the GAGE competition are on the server in /scratch/Rhodobacter/allpathsCor. As with any shared files that you need to read but not edit, you must use the reads in the shared location rather than copying them to your home directory.

Part 2 - Assembly using SOAPDenovo2

All assemblers require the same basic information:

- FASTQ file locations
- Read length
- Fragment size

- Insert size
- Fragment and insert standard deviation

Now that you've assembled with ALLPATHS-LG, read the SOAPDenovo2 instructions and assemble the ALLPATHS-LG corrected Rhodobacter reads with SOAPDenovo2.

Create soap.config following the example on the SOAPDenovo2 instructions page. Use a kmer size of 31.

Create soap.sh to run the assembler. The binaries (compiled program files) are in:

/usr/local/programs/SOAPdenovo2-r240

Choose the appropriate binary based on your kmer size.

The binaries aren't in your path, so use the absolute path to specify the binary in the shell script.

Specify 4 threads in soap.sh, and remember to put nice -n 19 at the beginning of the second line (first line is always #!/bin/sh for a shell script).

The binary path, filename, and parameters should follow nice -n 19 on line 2.

Specify RhodobacterSoap as the prefix of the output graph file name.

Run your shell script and compare the N50 from ALLPATHS-LG to the N50 from SOAPDenovo2.

SOAPDenovo2 doesn't require mate pairs, so run it a second time without the mate pairs. How does that affect the N50?

Part 3 – Plan a Sequencing Project

Using the Illumina coverage calculator and Harvard sequencing price page, write a project plan estimating the cost of sequencing a 430MB genome to a coverage depth of 100x. Indicate which assembler you plan to use and what types of libraries you will need. Assume sequencing will be done on a HiSeq 2500 at Harvard's Bauer Core. There are a number of right answers to this. The most important thing is for you to explain your choices, **use your own words**, and show your understanding of the process. As with all essay-type quiz questions and assignments in this course, any chunks of text pasted or retyped verbatim from documentation, web sources, your colleagues, etc. will result in zero for the assignment. Sources must be cited and ideas expressed **in your own words**.