Linux
+
Genome Assembly Tutorial

Pei-Chen Peng

Step 1A: Save login credential

For viewing and manipulating the files needed for this laboratory exercise, insert your flash drive.

Denote the path to the flash drive as the following:

`[course_directory]`

We will use the files found in:

`[course_directory]/password.txt`

Enter login credentials assigned to you, and save the file. We will use this account to login for lab sessions this week.

Linux commands

Using ClustW to align two sequences

Step 1A: Accessing the IGB Biocluster

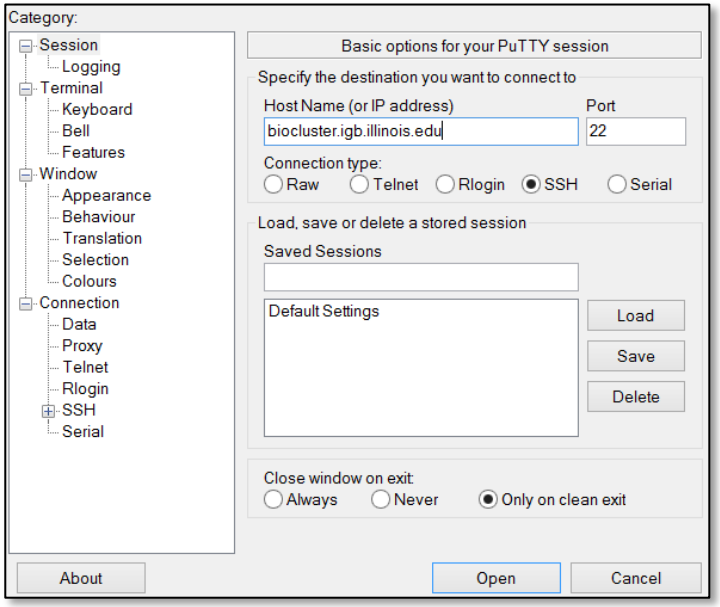
Open Putty.exe

In the **hostname** textbox type:

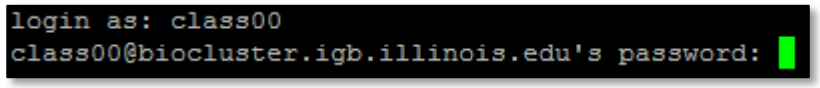
```
biologin.igb.illinois.edu
```

Click **Open**

If popup appears, Click **Yes**



Enter login credentials assigned to you; example, user **class00**. You will not see any characters on screen when typing in password. Just type it.



Now you are all set!

Step 1B: Listing files and directories (ls)

```
$ ls

# listing files in your current directory. When you first login, your
directory is your home directory.
```

Step 1C: Making Directories (mkdir)

```
$ mkdir ~/01_Linux_Genome_Assembly

# create a subdirectory in your home directory. The tilde ~ character refers
to your home directory.

$ ls

# to see the directory you just created.
```

Step 1D: Changing directory (cd)

The lab is located in the following directory:

`/home/classroom/mayo/2019/01_Linux_Genome_Assembly`

```
$ cd /home/classroom/mayo/2019/01_Linux_Genome_Assembly

# tip: use “tab” for auto-completetion for path

$ ls

# to see the contents. You should see seqs.fa
```

Step 1E: Print working directory (pwd)

```
$ pwd

# to see the full pathname. You should see
“/home/classroom/mayo/2018/01_Linux_Galaxy”
```

Step 1F: Copying files (cp)

Copy `seqs.fa` from the data directory to your working directory.

```
$ cp /home/classroom/mayo/2017/01_Linux_Galaxy/seqs.fa
~/01_Linux_Genome_Assembly/

# tip: use “tab” for autocompletetion for path

$ cd ~/01_Linux_Genome_Assembly/
```

Step 1G: Displaying the contents of a file on the screen (more)

```
$ more seqs.fa
# you should see two sequences on your screen
>seq1
GATCGAGCGATCGTGCAGC
GCAGAATGCGCGCTAG
>seq2
```

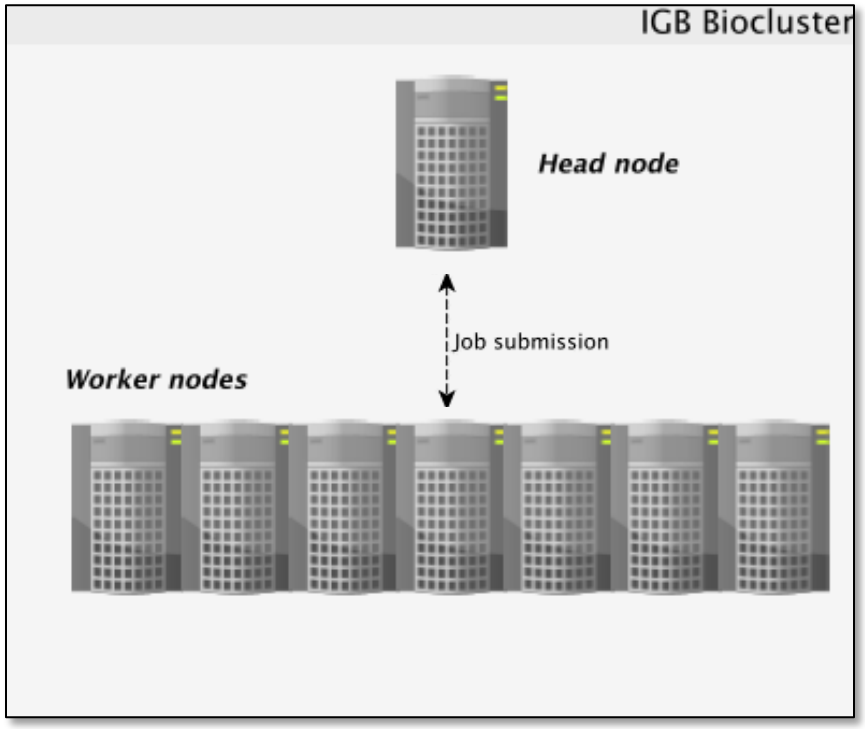
Commands Summary

Command	Meaning
ls	list files and directories
mkdir directory	make a directory
cd directory	change to named directory
cd ~	change to home directory
cd ..	change to parent directory
pwd	display the path of the current directory
cp file1 file2	cp file1 and call it file2
more file	display the contents of a file

Useful tips

Command	Meaning
tab	auto-comple path
↑	retrieve previous commands

Step 1H: Run sequence alignment program



Accessing the IGB Biocluster

Step 1H: Run sequence alignment program

```
$ srun -p classroom  -c 2 --mem 8000 --pty bash  # SKIP IF DONE

# Open interactive session on biocluster with 2 cpus and 8G memory.


$ module load ClustalW2     # Load sequence aligner into the shell environment.


$ module list #See loaded tools


$ clustalw2 -INFILE=seqs.fa  # Run the clustalW sequence aligner.
```

Step 1H: Run sequence alignment program

You will see this on your screen, when the program is done.

```
CLUSTAL 2.1 Multiple Sequence Alignments


Sequence format is Pearson
Sequence 1: seq1          35 bp
Sequence 2: seq2          32 bp
Start of Pairwise alignments
Aligning...

Sequences (1:2) Aligned. Score:  21
Guide tree file created:   [seqs.dnd]

There are 1 groups
Start of Multiple Alignment

Aligning...
Group 1:                      Delayed
Alignment Score 47

CLUSTAL-Alignment file created  [seqs.aln]
```

Step 1H: Run sequence alignment program

The alignment result is in **`seqs.aln`**. Use **more** command to see the result.

```
$ more seqs.aln

# You should see the following on your screen.

CLUSTAL 2.1 multiple sequence alignment

seq1            GATCGAGCGA-TCGTGCAGCGCAGAATGCGCGCTAG

seq2            GGTAGGGTAAATTGCCTACCGTCGATCGAGTA----

                * * * *  * * *   * **  **  * *
```

Exit putty by either closing the window or typing ‘exit’ in the command prompt.

Bacterial Genome Assembly

Chris Fields

PowerPoint by Saba Ghaffari

Introduction

Exercise

1. Perform a bacterial genome assembly using 454 data.

2. Evaluation and comparison of different datasets and parameters.

3. View the best assembly in EagleView .

.

Premise

1. We have sequenced the genomic DNA of a bacterial species that we are very interested in. Using other methods, we have determined that it's genome size is approximately 1 - 1.1 Mb

2. We chose to use Roche's 454 technology for performing this analysis because our genome of interest is relatively small and 454 gives us relatively long reads.

Dataset Characteristics

Dataset #	SFF Name	FQ Name	Size	# Reads
1	dataset1.sff	dataset1.fq	9.2 Mb	16,762
2	dataset2.sff	dataset2.fq	29.2 Mb	53,207
3	dataset3.sff	dataset3.fq	29.9 Mb	55,775

The .sff file and .fq file contain the same data in each case, however the .fq file is human readable and is regular text, whereas the .sff file is a binary format used by the assembler we want to use.

.sff -> “Standard flowgram format (SFF) is a binary file format used to encode results of pyrosequencing from the 454 Life Sciences platform for high-throughput sequencing”. Excerpted from http://en.wikipedia.org/wiki/Standard_Flowgram_Format

.fq -> “FASTQ format is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. Both the sequence letter and quality score are encoded with a single ASCII character for brevity”. Excerpted from http://en.wikipedia.org/wiki/Fastq

Step 0A: Accessing the IGB Biocluster

Open Putty.exe

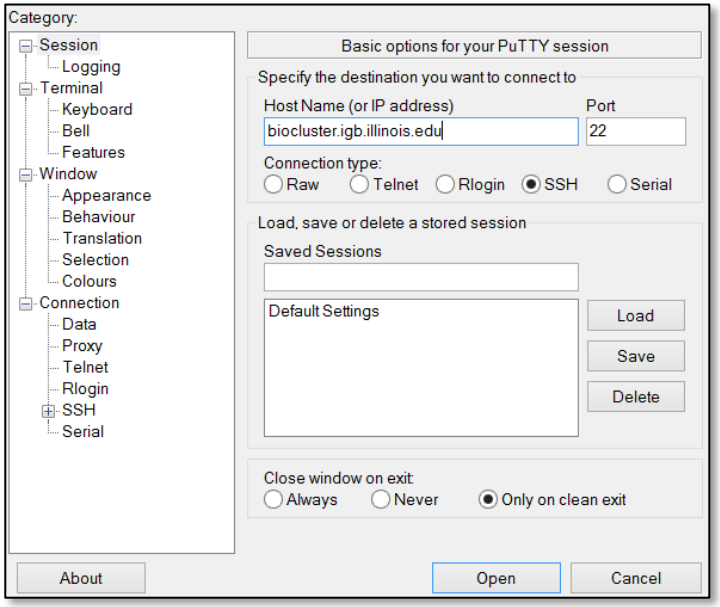
In the **hostname** textbox type:

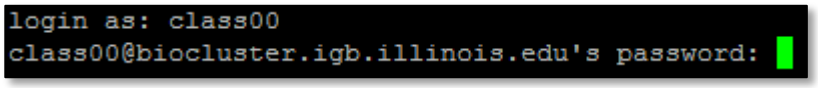
```
biologin.igb.illinois.edu
```

Click **Open**

If popup appears, Click **Yes**

Enter login credentials assigned to you; example, user **class00**.





Now you are all set!

Step 0C: Lab Setup

The lab is located in the following directory:

`/home/classroom/mayo/2019/02_Genome_Assembly`

This directory contains the initial data and the finished version of the lab (i.e. the version of the lab after the tutorial). Consult it if you unsure about your runs.

You don't have write permissions to the lab directory. Create a working directory of this lab in your home directory for your output to be stored. Note ~ is a symbol in unix paths referring to your home directory.

Make sure you login to a machine on the cluster using the **srun** command. The exact syntax for this command is given below. This particular command will login you into a reserved computer (denoted by classroom) with 2 cpus and 8000MB memory with an interactive session. **You only need to do this once**.

```
$ mkdir ~/02_Genome_Assembly

# Make working directory in your home directory

$ srun -p classroom  -c 2 --mem 8000 --pty bash   # Login to a computer on
cluster. # SKIP IF DONE
```

Step 0D: Local Files

For viewing and manipulating the files needed for this laboratory exercise, insert your flash drive.

Denote the path to the flash drive as the following:

```
[course_directory]
```

We will use the files found in:

```
[course_directory]/02_Genome_Assembly/results
```

Assembly

Using the GS *de novo* assembler (also known as Newbler) from 454/Roche, an assembler based on overlap identity. It is only applicable to 454 data

Step 1A: Run Assembly 1

For this 1st assembly we use **dataset2** (29 Mb)

Once you log into the biocluster with your classroom account, type the following commands.

```
$ srun -p classroom  -c 2 --mem 8000 --pty bash  # SKIP IF DONE
# Open interactive session on biocluster with 2 cpus.

$ cd /home/classroom/mayo/2019/02_Genome_Assembly/data/      # Change directory.

$ module load 454/2.8       # Load assembler into the shell environment.

$ runAssembly -force -o ~/02_Genome_Assembly/project_29Mb dataset2.sff
# Run the assembler.
```

Step 1B: Observe Assembly 1 Output

You will see this on your screen, when the assembly is running.

```
Created assembly project directory /home/a-m/mayo_instru01/02_Genome_Assembly/project_29Mb

1 read file successfully added.

    dataset2.sff

Assembly computation starting at: Wed May 30 14:48:13 2018  (v2.8 (20120726_1306))

Indexing dataset2.sff...

  -> 53207 reads, 23837200 bases.

Setting up long overlap detection...

  -> 53207 of 53207, 50525 reads to align

Building a tree for 511356 seeds...

Computing long overlap alignments...

  -> 53207 of 53207

Setting up overlap detection...

  -> 53207 of 53207, 20444 reads to align

Starting seed building...

  -> 53207 of 53207

Building a tree for 618232 seeds...
```

```
Computing alignments...

  -> 53207 of 53207

Checkpointing...

Detangling alignments...

   -> Level 4, Phase 9, Round 1...

Checkpointing...

Building contigs/scaffolds...

   -> 31 large contigs, 31 all contigs

Computing signals...

  -> 1100589 of 1100589...

Checkpointing...

Generating output...

  -> 1100589 of 1100589...

Assembly computation succeeded at: Wed May 30 14:50:42 2018
```

Step 2A: Run Assembly 2

For this 2nd assembly, we will use **dataset2** (29 Mb) again, but this time we will use a more stringent set of parameters.

The parameters we will change are **minimum overlap length** (`-ml`) and **minimum overlap identity** (`-mi`).

```
$ srun -p classroom  -c 2 --mem 8000 --pty bash
# Open interactive session on biocluster.                    SKIP IF DONE

$ module load 454/2.8  # Load assembler.                      SKIP IF DONE

$ runAssembly -force -o ~/02_Genome_Assembly/project_stringent -ml 60 -mi 96
dataset2.sff
# Run the assembler.
# Default Args: ml = 40%   and   mi = 90%
```

Step 2B: Observe Assembly 2 Output

You will see this on your screen, when the assembly is running.

```
Created assembly project directory /home/a-m/mayo_instru01/02_Genome_Assembly/project_stringent

1 read file successfully added.

    dataset2.sff

Assembly computation starting at: Wed May 30 14:56:32 2018  (v2.8 (20120726_1306))

Indexing dataset2.sff...

  -> 53207 reads, 23837200 bases.

Setting up long overlap detection...

  -> 53207 of 53207, 50525 reads to align

Building a tree for 511356 seeds...

Computing long overlap alignments...

  -> 53207 of 53207

Setting up overlap detection...

  -> 53207 of 53207, 20450 reads to align

Starting seed building...

  -> 53207 of 53207

Building a tree for 618471 seeds...
```

```
Computing alignments...

  -> 53207 of 53207

Checkpointing...

Detangling alignments...

   -> Level 4, Phase 9, Round 1...

Checkpointing...

Building contigs/scaffolds...

   -> 39 large contigs, 39 all contigs

Computing signals...

  -> 1099370 of 1099370...

Checkpointing...

Generating output...

  -> 1099370 of 1099370...

Assembly computation succeeded at: Wed May 30 14:59:01 2018
```

Step 3A: Run Assembly 3

For this 3rd assembly we use the small dataset, **dataset1** (9 Mb).

This one clearly cannot contain the full complement of data, but we want to see what kind of an assembly we get with insufficient data.

```
$ srun -p classroom  -c 2 --mem 8000 --pty bash
# Open interactive session on biocluster.                SKIP IF DONE

$ module load 454/2.8    # Load assembler.                SKIP IF DONE

$ runAssembly -force -o ~/02_Genome_Assembly/project_9Mb dataset1.sff
# Run the assembler
```

Step 3B: Observe Assembly 3 Output

You will see this on your screen, when the assembly is running.

```
Created assembly project directory /home/a-m/mayo_instru01/02_Genome_Assembly/project_9Mb

1 read file successfully added.

    dataset1.sff

Assembly computation starting at: Wed May 30 15:02:04 2018  (v2.8 (20120726_1306))

Indexing dataset1.sff...

  -> 16762 reads, 6895867 bases.

Setting up long overlap detection...

  -> 16762 of 16762, 15108 reads to align

Building a tree for 148560 seeds...

Computing long overlap alignments...

  -> 16762 of 16762

Setting up overlap detection...

  -> 16762 of 16762, 13678 reads to align

Starting seed building...

  -> 16762 of 16762

Building a tree for 433090 seeds...
```

```
Computing alignments...

  -> 16762 of 16762

Checkpointing...

Detangling alignments...

   -> Level 4, Phase 9, Round 1...

Checkpointing...

Building contigs/scaffolds...

   -> 210 large contigs, 216 all contigs

Computing signals...

  -> 1028479 of 1028479...

Checkpointing...

Generating output...

  -> 1028479 of 1028479...

Assembly computation succeeded at: Wed May 30 15:02:55 2018
```

Step 4A: Run Assembly 4

For this fourth assembly we use both large datasets, **dataset2** and **dataset3**.

```
$ srun -p classroom  -c 2 --mem 8000 --pty bash   SKIP IF DONE

# Open interactive session on biocluster.


$ module load 454/2.8    # Load assembler.              SKIP IF DONE


$ runAssembly -force -o ~/02_Genome_Assembly/project_60Mb
dataset2.sff dataset3.sff # Assemble
```

Step 4B: Observe Assembly 4 Output

You will see this on your screen, when the assembly is running.

```
Initialized assembly project directory /home/a-m/mayo_instru01/02_Genome_Assembly/project_60Mb

2 read files successfully added.

    dataset2.sff

    dataset3.sff

Assembly computation starting at: Wed May 30 15:05:54 2018  (v2.8 (20120726_1306))

Indexing dataset3.sff...

  -> 55775 reads, 24812962 bases.

Indexing dataset2.sff...

  -> 53207 reads, 23837200 bases.

Setting up long overlap detection...

  -> 108982 of 108982, 103279 reads to align

Building a tree for 1042876 seeds...

Computing long overlap alignments...

  -> 108981 of 108981

Setting up overlap detection...

  -> 108982 of 108982, 34236 reads to align
```

```
Starting seed building...

  -> 108982 of 108982

Building a tree for 963621 seeds...

Computing alignments...

  -> 108981 of 108981

Checkpointing...

Detangling alignments...

   -> Level 4, Phase 9, Round 1...

Checkpointing...

Building contigs/scaffolds...

   -> 38 large contigs, 44 all contigs

Computing signals...

  -> 1148106 of 1148106...

Checkpointing...

Generating output...

  -> 1148106 of 1148106...

Assembly computation succeeded at: Wed May 30 15:11:25 2018
```

Results:

The following instructions guide you to the location of the results. As the needed output for the rest of the lab is provided in the flash drive you could skip this slide.

You can find the results of all previous runs in folders project_29Mb, project_60Mb, project_9Mb, and project_stringent in the following directory:

```
~/02_Genome_Assembly
```

You can go to each folder by typing the following command:

```
cd ~/02_Genome_Assembly/[Folder-Name]
```

To see the files in the above directory type "ls" command.

Make sure that you return to your previous working directory for the rest of the lab by typing

```
cd /home/classroom/mayo/2019/02_Genome_Assembly/data/
```

The description of the results is provided in the next slide.

Newbler Output: Legend

Once the **Newbler** runs are done, you will have directories for the runs, and they will contain the following information.

File	Meaning	File	Meaning
454TrimStatus.txt	Tab-delimited text file providing a report of the original and revised trim points used in the assembly.	454LargeContigs.fna	FASTA file of all the “large” consensus base called contigs contained in 454AllContigs.fna (>500bp).
454AlignmentInfo.tsv	Tab-delimited file giving position-by-position consensus base and flow signal information.	454LargeContigs.qual	Corresponding Phred-equivalent quality scores for each base in 454LargeContigs.fna.
454Contigs.ace	ACE format file that can be loaded by viewer programs supporting the ACE format.	454ReadStatus.txt	Tab-delimited text file providing a per-read report of the status of each read in the assembly
454AllContigs.fna	FASTA file of all the consensus basecalled contigs longer than 100 bases.	454NewblerMetrics.txt	File providing various assembly metrics, including the number of input runs and reads, the number and size of the large consensus contigs as well as all consensus contigs.
454AllContigs.qual	Corresponding Phred-equivalent quality scores for each base in 454AllContigs.fna.	454ContigGraph.txt	A text file giving the “contig graph” that describes the branching structure between contigs.
		454NewblerProgress.txt	A text log of the messages sent to standard output during the assembly

Assembly Evaluation

What metrics do we use to evaluate the assembly?

Assembly Evaluation: Skeleton

9Mb	29Mb		60Mb
	default	stringent	
Genome Size (Mb)			
N50 (Kb)			
Number of contigs			
Longest contig (Kb)			
Shortest contig (bp)			
Mean contig size (Kb)			
GC content			

definition N50:

“Given a set of contigs, each with its own length, the *N50* length is defined as the shortest sequence length at 50% of the genome. It can be thought of as the point of half of the mass of the distribution. For example, 9 contigs with the lengths 2,3,4,5,6,7,8,9,and 10, their sum is 54, half of the sum is 27. 50% of this assembly would be 10 + 9 + 8 = 27 (half the length of the sequence). Thus the N50=8”. Excerpted from https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics#N50

Step 5A: Evaluate Assembly 1

We will evaluate the results of the 1st assembly (dataset 2) using a perl script:

assemblathon_stats.pl

```
# Use a perl script to determine the various metrics for Assembly 1

$ perl assemblathon_stats.pl
~/02_Genome_Assembly/project_29Mb/454AllContigs.fna
```

Step 5B: Output of Assembly 1 Evaluation

```
Number of scaffolds         31
Total size of scaffolds    1040658
Longest scaffold     131731
Shortest scaffold       1101
Number of scaffolds > 1K nt         31 100.0%
Number of scaffolds > 10K nt         25  80.6%
Number of scaffolds > 100K nt          2   6.5%
Number of scaffolds > 1M nt          0   0.0%
Number of scaffolds > 10M nt          0   0.0%
Mean scaffold size      33570
Median scaffold size      28079
N50 scaffold length      50527
L50 scaffold count          7
scaffold %A      29.30
scaffold %C      20.83
scaffold %G      20.43
scaffold %T      29.43
scaffold %N       0.00
scaffold %non-ACGTN       0.00
Number of scaffold non-ACGTN nt          0
Percentage of assembly in scaffolded contigs   0.0%
Percentage of assembly in unscaffolded contigs 100.0%
Average number of contigs per scaffold        1.0
```

```
Average length of break (>25 Ns) between
contigs in scaffold          0
Number of contigs         31
Number of contigs in scaffolds          0
Number of contigs not in scaffolds         31
Total size of contigs    1040658
Longest contig     131731
Shortest contig       1101
Number of contigs > 1K nt         31 100.0%
Number of contigs > 10K nt         25  80.6%
Number of contigs > 100K nt          2   6.5%
Number of contigs > 1M nt          0   0.0%
Number of contigs > 10M nt          0   0.0%
Mean contig size      33570
Median contig size      28079
N50 contig length      50527
L50 contig count          7
contig %A      29.30
contig %C      20.83
contig %G      20.43
contig %T      29.43
contig %N       0.00
contig %non-ACGTN       0.00
Number of contig non-ACGTN nt          0
```

Step 6: Evaluate Assemblies 2, 3, and 4.

We will evaluate the results of the stringent assembly using a perl script:

assemblathon_stats.pl

```
# Use a perl script to determine the various metrics for Assembly 2

perl assemblathon_stats.pl
~/02_Genome_Assembly/project_stringent/454AllContigs.fna


# Use a perl script to determine the various metrics for Assembly 3

perl assemblathon_stats.pl ~/02_Genome_Assembly/project_9Mb/454AllContigs.fna


# Use a perl script to determine the various metrics for Assembly 4

perl assemblathon_stats.pl ~/02_Genome_Assembly/project_60Mb/454AllContigs.fna
```

Step 7: Compare Assembly Statistics

	9Mb	29Mb		60Mb
		default	stringent	
Genome Size (Mb)	1.002770	1.040658	1.040516	1.049105
N50 (Kb)	7.106	50.527	39.736	77.259
Number of contigs	216	31	39	44
Longest contig (Kb)	25.092	131.731	126.716	168.246
Shortest contig (bp)	113	1101	703	270
Mean contig size (Kb)	4.642	33.570	26.680	23.843
GC content	41.31%	41.26%	41.26%	41.26%

We know that this genome size should be roughly 1 – 1.1 Mb; all of these assemblies are very close, even the 9Mb assembly with less than the ideal amount of data!

However, for the 9Mb genome, N50 is very low. N50 is much better when two conditions are met: more data is used and the longest contig is provided.

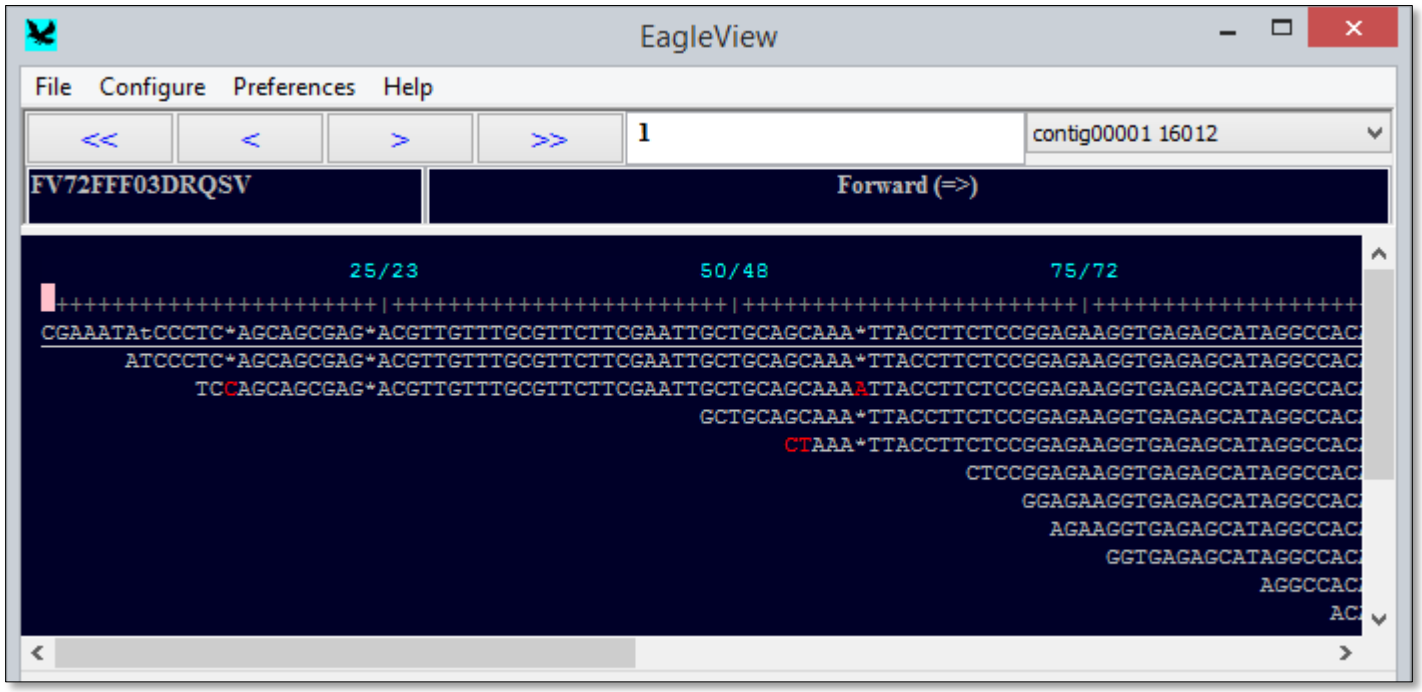
Assembly Visualization

Use EagleView to visualize the assembly.

Step 1: Assembly Visualization

Under **File,** go to **Open** and open the **project_60Mb 454Contigs.ace** file in the **results** directory:

[course_directory]/02_Genome_Assembly/results/454Contigs.ace



http://www.niehs.nih.gov/research/resources/software/biostatistics/eagleview/