**Assembling the *Staphylococcus aureus* Genome**

Every year in the United States, half a million patients contract a ***Staphylococcus* (Staph)** infection after surgery. Many of these patients are infected with drug-resistant strains such as **methicillin-resistant *Staphylococcus aureus* (MRSA)**, which can resist even last-resort antibiotics like Vancomycin and Daptomycin. As a result, MRSA causes over 20,000 deaths a year in the U.S. alone. Since there are over 40 different types of Staph bacteria that could be causing these infections, you want to determine which species is causing a Staph infection in a given patient by isolating this species in the patient and sequencing its genome. After you have sequenced its genome, scientists can start analyzing mutations that have led to antibiotics resistance.

**ASSEMBLY**

Let’s assume that we have isolated bacteria in the patient and generated reads for these bacteria. To assemble the genome from the reads, you will be using the **SPAdes** assembler (Bankevich *et al*, 2012) through the Galaxy service. Please follow these step-by-step instructions to register on Galaxy and run SPAdes:

**Register:**Create an account on Galaxy [here](https://usegalaxy.org/login). You will need to fill out all fields. After logging into Galaxy, you will see the following dashboard on the right side of the page. Click on the plus ‘+’ and then create a new history. Name it whatever you like. All of the following analysis will be performed under this history.

A screenshot of a phone

AI-generated content may be incorrect.

Next, we need to import our data. For this assignment, we will import raw data directly from the SRA database. Click on “Get Data” in the left side menu and search for the “**Download and Extract Reads in FASTA/Q** format from NCBI SRA”.

A screenshot of a computer

AI-generated content may be incorrect.

Launch the tool and use accession number **SRR643156**to import the data. Click "Execute" to import the data to your current history. It may take about 30 minutes for the app to begin execution and to load the data. When the tool is finished, you will be able to see the imported files and related information in your history on the right hand side of the page.

A screenshot of a computer program

AI-generated content may be incorrect.

Next, we will use SPAdes to assemble the genome. Go to "tools" on the left side of the page and search for SPAdes , clicking on it to launch it. Use default settings except for the value of kmer size. For this homework assignment, you will need to run the program three separate times (for *k* = 25, *k* = 55, and *k* = 85) to investigate how the choice of parameter *k* affects the resulting assembly.

For each value of k, enter the value of k and select ‘Interleaved files’ as the file format and select the file you imported in the previous step in the reads section. Hit execute to run SPAdes on the file.

A few important notes. First, you do not need to wait for the results to finish can queue all three runs at once – you will need to repeat all of the steps in this paragraph for each run. Second, Galaxy is an excellent public resource, but sometimes jobs may be slow to run or fail. If you have a failed run, or you would rather complete the analysis in this challenge without running SPAdes yourself, you can find files with the results of running SPAdes here:

<https://drive.google.com/drive/folders/13ZncRaPzPzQp5gPOVW3jcTIfxODvF2C2>

The contents of the above folder are as follows:

* SRR643156\_(fastq-dump).fastqsanger.gz  : Compressed interleaved fastqsanger file containing the reads
* Staph\_genome.fasta : Reference staph genome
* k\_25,55,85 : Folders containing the contigs.fasta and contigs.tabular files generated at the end of a spades run. The fasta file contains the contigs generated in the assembly process while the tabular file summarizes them. You can also find a log file for the Spades run within this folder.

If you are running SPAdes, then you will see the results appear in your history when the app has finished running. While the app runs, please continue reading.

**DEFINITIONS**

There are many assembly tools, but none of them is perfect. Biologists therefore need to evaluate the quality of various assemblers by comparing their results. In our case, once we have run the SPAdes assembler on a set of reads, we need to test the quality of the resulting assembly.

**Contig:** A *contiguou*s segment of the genome that has been reconstructed by an assembly algorithm

**Scaffold:** An ordered sequence of contigs (possibly separated by gaps between them) that are reconstructed by an assembly algorithm. The order of contigs in a correctly assembled scaffold corresponds to their order in the genome. Existing assemblers specify the approximate lengths of gaps between contigs in a scaffold.

A green rectangular object with text

AI-generated content may be incorrect.

**N50 statistic:** N50 is a statistic that is used to measure the quality of an assembly. N50 is defined as the maximal contig length for which all contigs greater than or equal to that length comprise at least half of the sum of the lengths of all the contigs. For example, consider the five toy contigs with the following lengths: [10, 20, 30, 60**,** 70]. Here, the total length of contigs is 190, and contigs of length 60 and 70 account for at least 50% of the total length of contigs (60 + 70 = 130), but the contig of length 70 does not account for 50% of the total length of contigs. Thus, N50 is equal to 60.

**NG50 statistic:** The NG50 length is a modified version of N50 that is defined when the length of the genome is known (or can be estimated). It is defined as the maximal contig length for which all contigs of at least that length comprise at least half of the length of the genome. NG50 allows for meaningful comparisons between different assemblies for the same genome. For example, consider the five toy contigs we considered previously: [10, 20, 30, 60, 70]. These contigs only add to 190 nucleotides, but say that we know that the genome from which they have been generated has length 300. In this example, the contigs of length 30, 60, and 70 account for at least 50% of the genome length (30 + 60 + 70 = 160); but the contigs of length 60 and 70 no longer account for at least 50% of the genome length (60 + 70 = 130). Thus, NG50 is equal to 30.

**NGA50 statistic:** If we already know a reference genome for a species, then we can test the accuracy of a newly assembled genome against this reference. The NGA50 statistic is a modified version of NG50 accounting for assembly errors (called **misassemblies**). To compute NGA50, errors in the contigs are accounted for by comparing contigs to a reference genome. All of the misassembled contigs are broken at **misassembly breakpoints**, resulting in a larger number of contigs with the same total length. For example, if there is a missasembly breakpoint at position 10 in a contig of length 30, this contig will be broken into contigs of length 10 and 20.

NGA50 is calculated as the NG50 statistic for the set of contigs resulting after breaking at misassembly breakpoints. For example, consider our example before, for which the genome length is 300. If the largest contig in [10, 20, 30, 60, 70] is broken into two contigs of length 20 and 50 (resulting in the set of contigs [10, 20, 20, 30, 50, 60]), then. contigs of length 20, 30, 50, and 60 account for at least 50% of the genome length (20 + 30 + 50 + 60 = 160). But contigs of length 30, 50, and 60 do not account for at least 50% of the genome length (30 + 50 + 60 = 140). Thus, NGA50 is equal to 20.

**Based on the above definition of N50, define N75.**

N75 is the maximal contig length for which all contigs with greater or equal length make up at least 75% or 3/4 of the sum of the lengths across all the contigs.

**Compute N50 and N75 for the nine contigs with the following lengths:**

[20, 20, 30, 30, 60, 60, 80, 100, 200].

Total length = 600

50% of total length = 300

N50: **100** [200+100 = 300]

75% of total length = 450

N75: **60** [200+100+80+60+60+60 = 500 > 450]

**Say that we know that the genome length is 1000. What is NG50?**

[20, 20, 30, 30, 60, 60, 80, 100, 200]

For a total length of 1000, NG50 occurs when total length is at least 500.

200 + 100 + 80 + 60 + 60 = 500.

So NG50 = 60.

**If the contig in our dataset of length 100 had a misassembly breakpoint in the middle of it, what would be the value of NGA50?**

New contigs = [20, 20, 30, 30, 50, 50, 60, 60, 80, 200]

NG50 again occurs when total length is at least 500.

200+80+60+60+50+50 = 500.

So NGA50 = 50.

**Based on the definition of scaffolds, what information could we use to construct scaffolds from contigs? Justify your answer.**

Scaffolds are ordered sequences of contigs. To generate a scaffold, we need only assemble the contigs in the correct order, then we will have their order in the genome. We need information about how the contigs relate to each other position-wise and orientation-wise. This could be from read-pairs, as prepared by Galaxy in this example exercise, or use a reference genome. We could alternatively look for long regions of overlap between contigs and infer continuity.

We will now answer questions concerning the assembly of the *Staphylococcus* reads. If you are running SPAdes yourself, then continue here as soon as your assembly of the Staph reads has completed. If you are following the completed runs, you can continue here at any time.

Consider the following three statistics:

* N50.
* The number of ***long* contigs**, i.e., contigs with length ≥ 1000 nucleotides. Biologists are mainly interested in long contigs and often discard short contigs, since short contigs often harbor only fragments of genes rather than complete genes.
* The total length of *long* contigs. This statistic can be combined with N50 and the number of long contigs; a good assembly is one that has relatively few long contigs, but the total length of long contigs is high, as is N50.

These three statistics can be found by analyzing the contigs.fasta file or the contigs.tabular file generated at the end of SPAdes execution. The tabular file is a summary sheet of all the contigs present in the assembly. Below is an example snapshot of what the fasta and tabular file look like, respectively, for *k* = 25.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

You will use the Quality Assessment Tool for Genome Assembly **QUAST**(Gurevich *et al*, 2013) to evaluate the quality of your assembly using the Staph reference genome as the gold standard.

* Download the contigs.fasta file as part of the SPAdes output from each value of k.
* For each of the three files, go to QUAST (<http://cab.cc.spbu.ru/quast/>) and upload your contigs.fasta file with the “Add files” button.
* Leave the “Scaffolds” and “Find genes” boxes unchecked and keep the indicator on “Prokaryotic.”
* Click on the “Another genome” link underneath “Genome.” Fill in a name and upload the [staph\_genome.fasta](http://bioinformaticsalgorithms.com/software_challenges/assembly/staph_genome.txt) file that we provided for the “Reference” file. (Note: we provide this file as a .txt, you will need to save it as .fasta). Leave the other two inputs (“Genes” and “Operons”) blank and click “Evaluate.”
* A link to the report should appear on the right side of the page in a few moments. Evaluate the report and answer the following questions.
* If you had difficulties running QUAST, please find the required reports for this part of the assignment in the Quast reports folder [here](https://drive.google.com/drive/folders/1L-cfjiyd9RJOwY1Slau-wQHGmO1g1xft?usp=sharing): <https://drive.google.com/drive/folders/1L-cfjiyd9RJOwY1Slau-wQHGmO1g1xft?usp=sharing>
* Select the report (a html file) corresponding to the k value chosen. Download it and open it using any web browser.
* All the required statistics are present on the left panel of the report. Click on ‘Extended report’ to see all statistics required for the following questions.

First, fill in the 9 missing values in the following 3 x 3 table:

(Reminder: long contigs >= 1000 bp)

k N50 #long contigs total length of long contigs

25 59,595 110 2,802,857

55 159,616 38 2,821,839

85 188,896 37 2,825,752

**Which assembly performed the best in terms of each of these statistics? Justify your answer. Why do you think that the value you chose performed the best?**

k=85 performed the best, since it has the largest N50 and smallest count of large contigs. This value likely performs the best because it contains the most continuous information. With smaller k, it is harder to accurately reconstruct overlap.

**(Multiple choice) When you increase the length of *k*-mers, the de Bruijn graph \_\_\_\_\_\_\_\_\_\_\_\_. Justify your answer.**

**A) Becomes more tangled.**

**B) Contains more nodes.**

**C) Becomes less tangled.**

**D) Remains the same.**

C) Becomes less tangled. The longer the k-mer, the less likely it is to have overlap with other k-mers. As such, there will not be as many branching paths from any given k-mer, so the graph will be less complex (although perhaps more fragmented as k increases).

**Answer the following two questions using the QUAST reports.**

**How many misassemblies were there? How significant is the effect of misassemblies on the resulting assembly?**

There were 23 misassemblies for k=25, 27 misassemblies for k=55, and 29 misassemblies for k=85. The more misassemblies present, the fewer the number of long contigs, since some of the true long contigs are instead broken down into fragments.

**What are NG50 and NGA50 for the QUAST run? How do they compare with the value of N50 that you previously calculated? Why?**

For k=25: NG50 = 77,760 and NGA50 = 35,824. Compared to N50 = 59,595, NG50 is slightly larger and NGA is slightly lower. Misassembly (or relative lack thereof) has a small effect for small k. The contigs display high coverage here.

For k=55: NG50 = 276,512 and NGA50 = 92,194. Compared to N50 = 159,616, NG50 is much larger and NGA50 is moderately smaller. Misassembly has a moderate impact for moderate k.

For k=85: NG50 = 202,267 and NGA50 = 87,161. Compared to N50 = 188,896, NG50 is slightly larger while the NGA50 is much smaller. Misassembly has a large effect for large k.