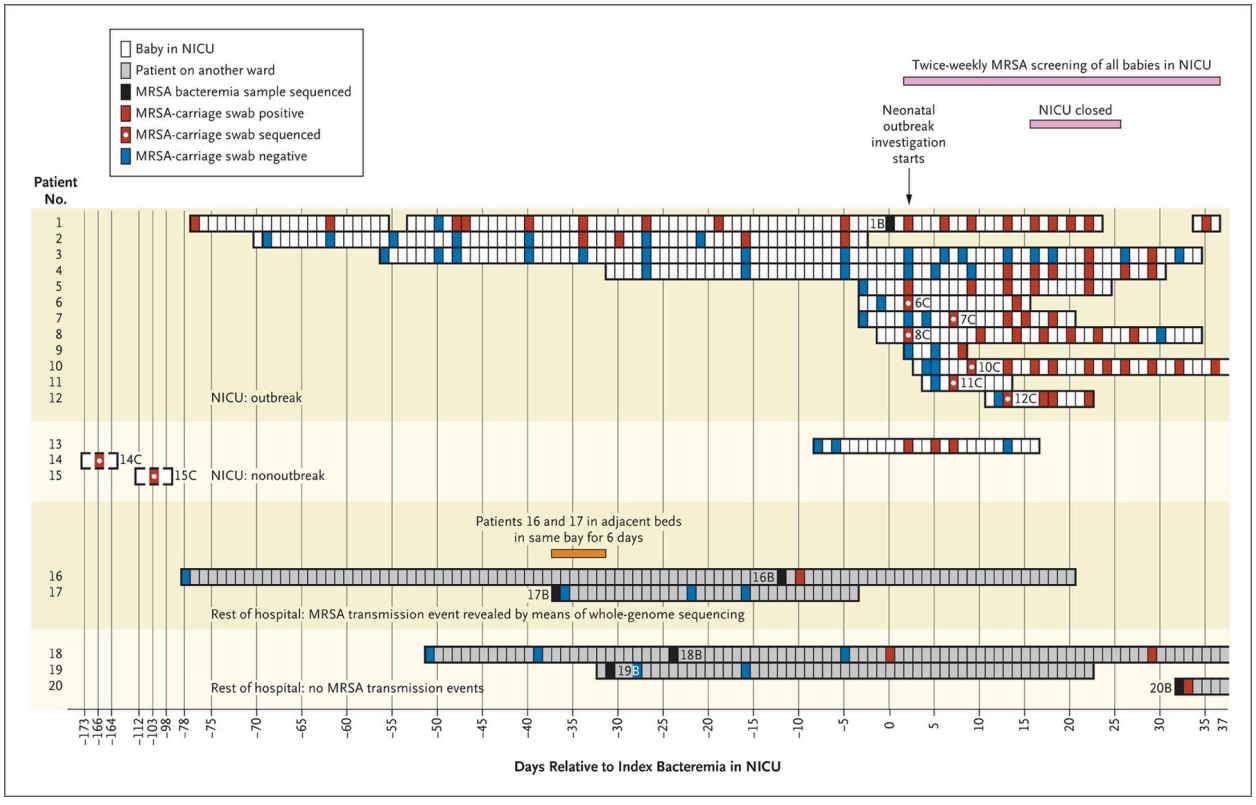
QBS146 Homework #3

Staphylococcus is a genus of Gram-positive bacteria that is commonly found on the skin and in the mucous membranes of humans and other warm-blooded animals. They are generally round in shape and form clusters that resemble a bunch of grapes when observed under a microscope. Among the numerous species within the Staphylococcus genus, Staphylococcus aureus is the most pathogenic, and it is responsible for a wide range of infections.

Methicillin-resistant Staphylococcus aureus (MRSA) is a particularly concerning strain of staph that has developed resistance to many commonly used antibiotics, making treatment more challenging. The rise of MRSA highlights the importance of proper hygiene, infection control measures in healthcare settings, and the judicious use of antibiotics to prevent the further emergence and spread of antibiotic-resistant bacteria.

There is also an application for sequencing of MRSA in the clinic, where the identification of strains can help guide treatment and track hospital outbreaks; for instance see this plot blow from Koser et at. In the New England Journal of Medicine, where they tracked a neonatal outbreak using sequencing of isolates from patients and carriages:



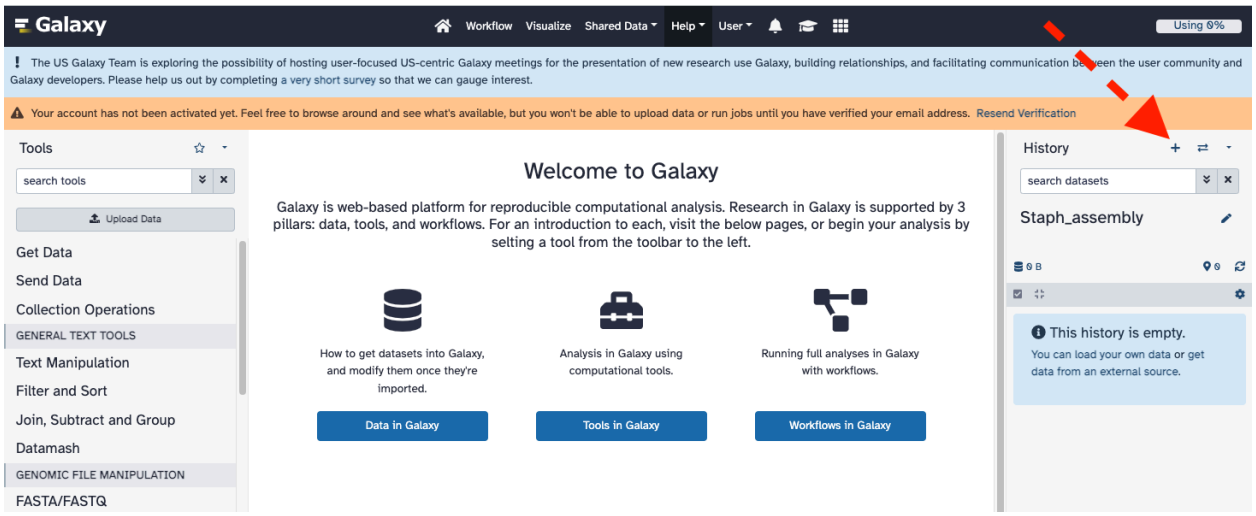
A lot can be learned from simply aligning the reads to the reference Staph genome (which we'll cover in later classes). In this case, we want to dig a little deeper: to get the most complete picture, especially in species capable of large-scale genome changes, you'll often want to

assemble each sample and compare the de novo assembled genomes to each other. So, let's try to assemble a bacterial genome using Illumina sequencing reads and see how well we do.

Galaxy assembly

We'll be using Galaxy, an online bioinformatics pipeline and engine tool, to set up and perform the assembly. This greatly simplifies our process, as we don't have to download and build the assembler, convert sequencing reads from one format to another, or deal with other minor details.

First, create a Galaxy account here. 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. In Galaxy these histories store everything you've done: downloading data, running analysis, collecting results, and saving job outputs. All of the following analyses will be performed under this history:



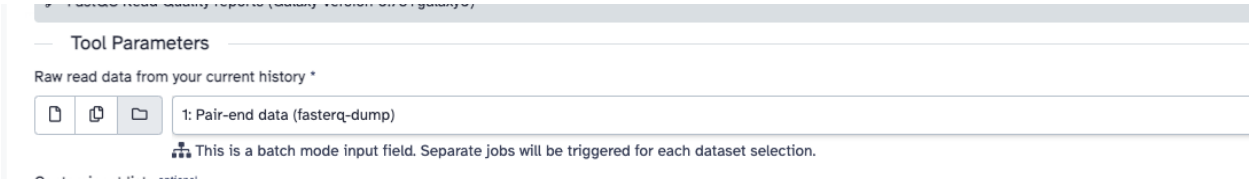
Fetching data for assembly

Next, we need to import our data. For this assignment, we will import raw data directly from the SRA database, the large NCBI storage system for 2nd generation sequencing reads. Click on "Get Data" in the left side menu and search for the "Download and Extract Reads in FASTA/Q format from NCBI SRA". Launch the tool and use accession number **SRR643156** to import the data. Click "Run Tool" (older versions have 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 side of the page (it will turn green). If it fails just try again, this step shouldn't be too much of an issue.

Evaluating the input reads

It's good to know the input sequences' quality before judging the (assembly) output. A common tool for understanding the quality of sequencing reads is called FastQC. It calculates a number of metrics about the input reads, including their average quality score and over-represented

motifs. Find this tool with the search, and run it on the paired-end reads you've downloaded above:



Genome Assembly
Next, we will use SPAdes to assemble the genome. Go to "tools" on the left side of the page and search for SPAdes. There are a number of options in Galaxy, we'll use the *'SPAdes genome assembler for genomes of regular and single-cell projects'* options. You'll need to select the 'paired-end: interlaced reads' as the input dataset option and the 'FR' orientation for the reads. Also, to speed things up, check the '*Isolate: highly recommended for high-coverage isolate and multi-cell data (--isolate)*' checkbox. Leave the K-mer choice as automatic for now.

Hit *Run Tool* to run SPAdes on the file.

A few important notes. First, you do not need to wait for the results to finish, you can come back later (it takes between 5 minutes and ~ a couple of hours, depending on how busy Galaxy is). Second, Galaxy is an excellent public resource, but sometimes jobs may be slow to run or fail. If you have a failed run, you can just try again a little later (it's rare to have a job fail, but it happens).

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. There are some common terms you'll need to know in genome assembly:

Contig: A contiguous segment of the genome that has been reconstructed by an assembly algorithm. There are generally no large gaps, this is the most concrete assembly output, represented by the blue boxes above.

Scaffold: An ordered sequence of contigs (separated by variable 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. Assemblers specify the approximate lengths of gaps between contigs in a scaffold, like the Ns above, but this is a guesstimate from the input reads.

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.

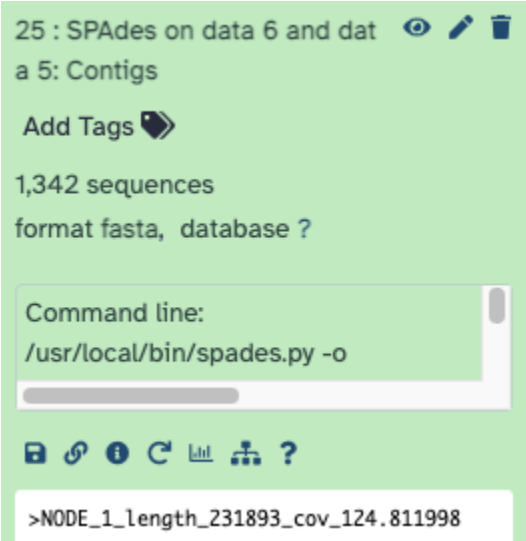
NGA50 statistic: If we already know a reference genome for a species, then we can test the accuracy of a newly assembled genome against this reference. The NGA50 statistic is a modified version of NG50 accounting for assembly errors (called misassemblies). To compute **NGA50**, errors in the contigs 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 misassembly 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 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 lengths 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.

There are a bunch of questions in the python homework to answer, head over there and try out those questions while your genome is building!

Now, get the genome output

In Galaxy, your Staphylococcus assembly should now be complete. You will see a couple of assembly outputs, the two most important outputs are the contig-level assembly and the scaffold assembly. The contig-level assembly consists of a series of contigs that the assembler has

confidently pieced together without any ambiguity. On the other hand, the scaffold assembly represents the assembler's effort to arrange contigs into larger segments, even though there might be unknown gaps between them. Here we'll just assess the contig assembly, which will look something like this on the history sidebar. Choose the disk icon to download:



You can now answer the final questions in the homework! Head on over to the GitHub Jupyter notebook and check out the questions. You can also look at the scaffold assembly later if you're curious, which should contain a set of much higher N50 DNA segments, though many internal gaps aren't exactly correct.

Footnotes:
Inspiration and text segments for this assignment came from Phillip Compeau's excellent model assignment in CMU's 02-251 class and from this online training.