02-251 COVID-19 Challenge
Genome Assembly
Daniel Schaffer & Phillip Compeau

Introduction

In this collection of assignments, we will use data and software produced by others to draw conclusions about the SARS-CoV-2 virus that causes COVID-19. In this sense, we are "research parasites", a term coined by a now infamous 2016 editorial[1] in the *New England Journal of Medicine*:

> *A … concern held by some is that a new class of research person will emerge — people who had nothing to do with the design and execution of the study but use another group's data for their own ends, possibly stealing from the research productivity planned by the data gatherers, or even use the data to try to disprove what the original investigators had posited. There is concern among some front-line researchers that the system will be taken over by what some researchers have characterized as "research parasites."*

Just as we started this course with a discussion of how genomes are assembled, we start our study of SARS-CoV-2 by assembling its genome. This task was first completed on January 25, 2020 using a sample collected from the first known human patient[2].

Many patient DNA or RNA samples taken from swabs are metagenomic, meaning that they capture genetic material not only from the virus but also other viruses, bacteria, and even human cells. This means that researchers either have to apply an algorithm that is able to assemble genomes from information derived from multiple species, or they need to perform additional laboratory work to isolate the viral information. The researchers who assembled the first SARS-CoV-2 genome did the former, wrangling a 30,000 base pair genome out of a file consisting of 8 billion base pairs[3], most of which do not derive from SARS-CoV-2. (If genome assembly is like assembling a puzzle, metagenome assembly is like assembling multiple similar puzzles from a jumbled set of pieces.)

To prevent the complications of a metagenomics sample, we will work with a different viral read dataset that was sampled directly from an infected cultured cell.

Sequencing a virus is more straightforward than assembling a bacterium for a couple of reasons. First, the virus's genome is much shorter. Second, SARS-CoV-2 is an **RNA virus**,

---

[1] Longo, D. L. and Drazen, J. M. (2016). Data Sharing [Editorial]. *N Engl J Med*, *374*, 276-277.
[2] Wu., F, *et al*. (2020). A new coronavirus associated with human respiratory disease in China. *Nature, 579*, 265-269.
[3] https://www.ncbi.nlm.nih.gov/sra?LinkName=bioproject_sra_all&from_uid=603194

meaning that its genome does not contain DNA, but rather a single (linear) strand of RNA, which is single-stranded; as a result, we do not encounter the complications that we do when assembling double-stranded DNA. Researchers convert RNA to DNA by using an enzyme called **reverse transcriptase**, a molecular tool that was invented by viruses to help them replicate their genome and now has been borrowed by humans to help us sequence this genome.

In this assignment, we will assemble the SARS-CoV-2 genome, and then we will **annotate** the contigs resulting from this assembly, meaning that we will compare our genome against a database of known genomes to label putative genes. Knowing the locations of these genes will allow us to study them in future work.

Getting started with Galaxy

To assemble the genome from the reads, we will be using the SPAdes assembler[4] that is built upon a de Bruijn graph approach and is the most cited genome assembler of all time. To run SPAdes, we will use the Australian service of Galaxy, an open source project that allows us to run often testy bioinformatics software in the cloud without the hassle of dealing with local installations. Please follow these step-by-step instructions to register on Galaxy and run SPAdes:

First, create an account on Galaxy here. You don't need to use your university email address, because we are only grading the results of your analysis. Your "public name" can be whatever you like, but you should fill out all fields. After creating an account, log in to Galaxy.

Click on the plus '+' on the top right of the page and create a new "history" (i.e., project that will contain data and the results of software runs) named "SARS-CoV-2". All of the following analysis will be performed under this history.

Grabbing some sequencing data

The National Center for Biotechnology Information (NCBI) hosts an enormous amount of publicly available biological data. You may be interested in their website for SARS-CoV-2 (https://www.ncbi.nlm.nih.gov/sars-cov-2/), which contains sequencing data from hundreds of thousands of sequencing "runs". These runs are stored in an arm of NCBI called the Sequence Read Archive (SRA), where researchers can upload sequencing reads from their experiments.

Our dataset of interest has the SRA identifier SRR11528307, and its homepage can be found here: https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR11528307. We can see at this page

[4] Bankevich, A, *et al*., (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell processing. *J Comput Biol*, *19*, 455-477.

that the dataset contains approximately 719,800 reads ("spots"), that the technology is "Illumina", and that the layout is "paired". These are short paired-end reads produced by Illumina's popular sequencing by synthesis approach that we learned about.

Exercise 1: How many nucleotide bases are found in this file? What is the GC-content of the reads? (That is, the percentage of reads that are either G or C.)

Answer (1 point each): 186.5 million; 38.2%.

Exercise 2: We know that the original SARS-CoV genome from the 2003 outbreak has approximately 30,000 nucleotides. If the SARS-CoV-2 genome has length 30,000 nucleotides, then what is the coverage of this read dataset?

Answer (2 points): 186.5 million/30,000 ~ 6217X

Clicking on the "Reads" tab will allow us to see ten read-pairs at a time. Each read is in **FASTA format**, meaning that the read has a header beginning with the ">" symbol identifying the read, followed by the read itself.

Exercise 3: What is the read-pair at spot #15213? Why do you think that one read is shorter than the other?

Answer (3 points; 1 point for first part, 2 points for second part): Reads are shown below. There are lots of reasons why the read could be shorter. The most likely correct one is that the machine at that position started losing signal, but students may have some creative ideas too.

>gnl|SRA|SRR11528307.15213.1 MN01288:4:000H32WJK:1:11101:22090:17095 *(Biological)*
CAACAAGGCCAAACTGTCACTAAGAAATCTGCTGCTGAGGCTTCTAAGAAGCCTCGGCAA
AAACGTACTGCCACTAAAGCATACAATGTAACACAAGCTTTCGGCAGACGTGGTCCAGAA
CAAACCCAAGGAAATTTTGGGGACCAGGAA

>gnl|SRA|SRR11528307.15213.2 MN01288:4:000H32WJK:1:11101:22090:17095 *(Biological)*
TCAATATGCTTATTCAGCAAAATGACTTGAT

Now that we know where our data is contained, let's import it into Galaxy. To do so, go back to Galaxy and on the left side of the page, under "Get Data", click "Download and Extract Reads in FASTA/Q Format from NCBI SRA". Make sure that "select input type" shows "SRR accession", and type our accession ID (SRR11528307) into the field below. You may want to click "Yes" for "Email notification", although the job should not take more than five minutes to run. Then click "Execute". You should see the job added to your history on the right; it will turn green when it is finished.

FASTQ format and Phred quality scores

When the job turns green, click on the "eye" symbol next to SRR11528307 under your history on the right side of the page. The page may take a few seconds to load, and Galaxy will show you the first megabyte of your read file.

This file is in **FASTQ format**, an extension of FASTA format in which each read is represented over four lines described as follows.

1. A header beginning with the "@" symbol and labeling the read. Notice that because these are read pairs, the first two reads' headers are the same and end with "/1" and "/2".
2. The read as a sequence of nucleotides.
3. A line containing a "+" symbol to indicate the end of nucleotides. This line may have more header information, but doesn't for this dataset.
4. A collection of ASCII symbols representing **Phred quality scores**, described below. The *i*-th quality score corresponds to the *i*-th nucleotide in the read.

A given base returned as the result of a sequencing run is assigned a quality score Q based on an estimate of how likely this base is correct. Because many bases are very reliable, this computation is done on a logarithmic scale. In particular, once we compute the probability p that a base is incorrect, we set $Q = -10 \log_{10}(p)$ and assign the base the ASCII symbol corresponding to $Q + 33$. For example, if p is 0.001, then $\log_{10}(p)$ is -3, and so Q is equal to 30, and we assign this base the ASCII symbol corresponding to 30 + 33, which is "?". As a result, the lower the value of p, the higher the quality score Q. The Phred quality score table is shown in the figure below.

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Figure: The current Phred score table. Each row contains a quality score Q, the corresponding probability of error, and the ASCII symbol used to label a nucleotide having this quality score. Source: https://bit.ly/2NrZvS7.

Phred scores are helpful for a variety of reasons. For example, researchers will throw out reads having a significant number of bases that do not meet a certain Phred score threshold, especially if the reads have high coverage.

Exercise 4: How would Phred scores be useful when applying the de Bruijn graph approach to genome assembly?

Answer (3 points): Multiple answers exist, but the most likely one I would expect is that when removing bubbles, we could consult the base of the two conflicting paths, and then remove one if it corresponds to a bad Phred score.

Exercise 5: In class, we learned about the sequencing by synthesis approached used by Illumina for sequencing reads. What do you think that Illumina considers in order to determine the quality score of a given base?

Answer (3 points): There are compelling answers possible here, but I would imagine something that relates to the fact that a base is detected via the signal of a point on an image. Nearby clusters could cause the signal to be harder to read, and perhaps a given cluster could have a color signal that is weaker at a given position.

Exercise 6: Head back to the SRA page for our dataset. In the "Metadata" tab, there is small text that says "Quality graph". You may need to click "bigger" to see this graph, which is a histogram of Phred quality scores over the entire dataset. Interpret this chart. Are the quality scores good? What makes you think so?

Answer (3 points): It may be possible that students would say that the quality scores are bad, with a reasonable explanation. In general most of the quality scores appear to be higher than 30, which means that we are very confident in most of the base pairs in our read dataset, and we can consider these quality scores to be very good.

Exercise 7: Return to Galaxy and view your SRR11528307 dataset. What are the quality scores of read "8595/1"? Are they good? Are there any nucleotides you are worried about?

Answer (3 points): Most of the positions are "F", which is a quality score of 37 and is very good. Eight of the positions are "/", which is a quality score of 14, and we are only about 96% sure about these bases. That makes this read a bit troubling.

Assembling the SARS-CoV-2 genome

Now that we understand our data a bit better, we are ready to assemble the SARS-CoV-2 genome from our read set using SPAdes. Under `Genomics Analysis` → `Assembly` on the left side of the page, you will see over a dozen different assemblers. Click on SPAdes, which will take you to its homepage.

We will go through each of the settings offered and briefly explain them as well as what we are going to choose.

- *Single-cell?* Set to "No". This setting is used for bacterial projects when we have a bacterium that we cannot culture, and we only have DNA captured from a single cell.
- *Run only assembly*? Set to "No". We want SPAdes to try and "error correct" reads, meaning that it looks for potentially troubling bases and correcting reads. For example, if a read matches another read exactly except for a single base with a very low quality score, we may want to error correct this read before we build a de Bruijn graph.
- *Careful correction?* Set to "Yes". We will explain the BWA tool later in the course. ☺
- *Automatically choose k-mer values*? Set to "Yes". SPAdes works by building de Bruijn graphs for several values of *k*. The default values are *k* = 21, 33, and 55. We have no *a priori* information that would lead us to change SPAdes' default behavior, so we will let its "machine learning" choose the best values to use.
- *Coverage cutoff*? Set to "Off". If we had a high-coverage collection of reads of varying lengths, we might choose only the longest reads above some threshold (assuming they have sufficient quality) to save us some time in running the assembler. We are assembling a virus, so we don't need a coverage cutoff.
- *Libraries are ionTORRENT reads?* Set to "No" as these are Illumina reads. ionTORRENT reads are produced by a different company (Oxford Nanopore) and are longer and typically less accurate.
- *Library type*. Set to "Paired end/single reads". Remember that we know these are paired-end reads.
- *Orientation*. Set to "fr". Remember that the convention is to always read DNA in the 5' to 3' direction. In forward-reverse read pairs, the first read is taken from one strand, and its pair is taken from the opposing strand, as the following figure illustrates.

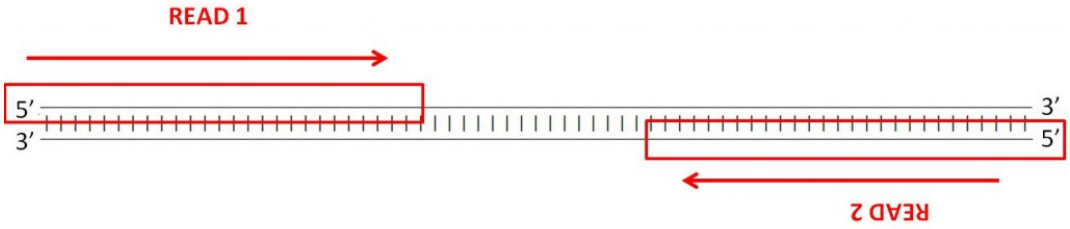


Figure: In "forward-reverse" reads, the first read is taken from one strand, and the second read comes from the opposing strand. Source: https://bit.ly/3dyv47n.

- *Select file format.* Set to "interleaved files". As you know, the read-pairs are found as consecutive reads in our fastq file. This is called "interleaved" as opposed to having the forward reads in one file and the reverse reads in another.
- *Interleaved paired reads*. You should be able to select your SRR11528307 dataset for this field.
- *PacBio reads, etc.* Don't choose anything in these sections.
- *Output final assembly graph* (*contigs*)? Set to "Yes" as we would love to know the final assembly graph produced by SPAdes.

- *Output final assembly graph* (*scaffolds*)? Set to "No". Scaffolds are formed by trying to join contigs into longer contiguous sequences. We will only view the graph of the contigs.
- *Email notification?* We suggest setting to "Yes" so that you can leave this running and get notified when it's done.

You are now ready to go! Click "Execute". SPAdes should take 15-20 minutes to finish.

Analyzing the results of our assembly

Now that SPAdes has completed, we will analyze the results. As the previous exercise indicates, our work appears to have been quite successful.

Exercise 8: Click on the eye symbol ("view data") next to the "contigs (fasta)" line on the right side. How many contigs did our assembly produce? What are their lengths? What do you think "coverage" means in this context?

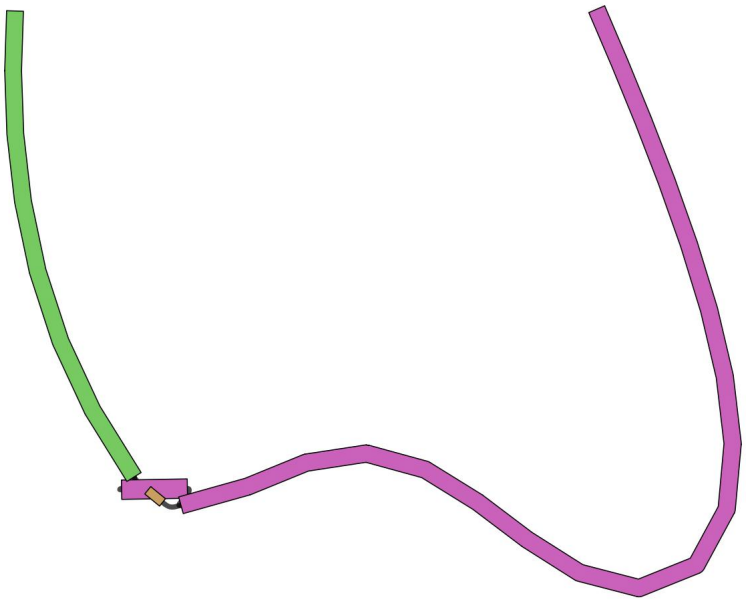
Answer (4 points; 1 point for first part, 3 for second): Two contigs, of length 29600 and 147. (It's possible that students get slightly different results here.) The coverage of a contig is the number of nucleotides in the reads that produced this contig, divided by the length of the contig.

We also produced an "assembly graph" as the result of our work, which is a version of the de Bruijn graph (after some cleaning steps involving bubble removal) in which maximal nonbranching paths have been compressed to a single edge. If you "view data" for the assembly graph, you will see that this graph has four edges.

Let's visualize this assembly graph! To do so, we use a program called Bandage. Under "Assembly" on the left side of the page, click on "Bandage Image". Under "Graphical Fragment Assembly" select the assembly graph that you just produced. Feel free to keep all other parameters the same (possibly opting for an email notification), and click "Execute". The program should only take a couple of seconds to run, after which clicking "view data" on the run will show the graph.

Exercise 9: Show the image produced of the assembly graph. Where do you think the contigs of our assembly are hiding? Where do you think contig 2 exists in the genome?

Answer (4 points; 1 point for assembly graph, 3 points for remainder): The assembly graph is shown below. The long strand involving the two purple regions and the green region corresponds to contig 1, and the short brown rectangle corresponds to contig 2. We can be lenient on the last question, but contig 2 is not at the end of the genome but rather interior to the genome.



Aligning the two viruses

Because we have a single long contig, and the virus genomes are ultimately short, let's align it against the original SARS-CoV genome from the 2003 outbreak to see how the two coronavirus genomes differ.

Exercise 10: We will align the two genomes as DNA strings rather than translating them to protein strings. Why do you think this is?

Answer (3 points): The genome is formed of more than one gene, and it's not yet clear where they start. (Furthermore, not all of the genome is translated into protein.)

To align the SARS-CoV-2 genome against the original SARS-CoV genome, we will use the (affine) global alignment algorithm that we learned about in class, provided at NCBI at https://bit.ly/2NtCsWX under the accession ID NC_004718.3. To import this genome, we will use the "NCBI Accession Download" tool under "Get Data" on the left side of the Galaxy page. After clicking on this tool, follow these steps.

- Under "Select source for IDs", select "Direct Entry".
- Under ID List, type `NC_004718.3`.
- Keep the other parameters the same, add an Email notification if you like, and click "Execute". The tool should be very quick.

Now we are ready to align the two genomes. Under the "EMBOSS" section on the left side of the page, click "needle", which is short for "Needleman-Wunsch global alignment" (our algorithm from class, which was developed by Needleman and Wunsch in 1970).

For sequence 1, select the contigs that we produced as the result of SPAdes. For sequence 2, select NC_004718.3. Keep the other parameters default (you may like to play around with these parameters later to see how they affect the outcome). After opting for an Email notification, click "Execute". The algorithm should take a few minutes to run. (After all, it will take it some time to build an array for two 30,000 nucleotide sequences!)

When the algorithm has finished, view its data. This will show the alignment, in which matches are shown with "|" symbols, mismatches are shown with "." symbols, and indels do not have a symbol between the two rows.

Exercise 11: How many matched symbols does the alignment report? How many gap symbols?

Answer (2 points): 23875 matches ("identity"); 1545 "gaps".

Exercise 12: What do you notice about the ends of the alignment? What do you think happened when the genome was assembled?

Answer (3 points): This is somewhat open ended, but this contig did not capture the ends of the genome.

Exercise 13: Scroll through the interior of the alignment. Do you notice any regions that appear to be more variable than other regions?

Answer (3 points): Students' answers will vary here probably, but there is definitely a region from around 21250 to around 22200 where there is significantly more variability (we know later that this is the Spike protein region).

Annotating the SARS-CoV-2 genome

Finally, we will **annotate** the SARS-CoV-2 genome, meaning that we will identify putative genes and then compare these genes against a database of known genes to find which ones align the best.

The identification of putative genes can be done in a variety of ways. One way to find these regions requires the observation that regions of DNA that are translated into protein start with a **start codon** (ATG) and end with a **stop codon** (TAG, TAA, or TGA). If a genome were random, then after encountering a start codon, we would expect to see a stop codon after about 64/3 ~ 21 codons. But real genes are typically much, much longer than 21 codons. As

a result, in simple organisms like viruses and bacteria, a great way to find genes is to look for long stretches of codons connecting a start codon to a stop codon, with no intermediate stop codons.

Once we have found putative genes along the genome, we can compare them against a database of genes using an algorithm called BLAST that we will learn about soon.

These two steps are taken by our next tool, called Prokka, which is used to annotate the genomes of viruses, bacteria, and archaea. It can be found under "Annotation" on Galaxy. Under "Contigs to annotate", select our contigs from the SPAdes run. Set "Kingdom" to "Viruses" and leave all other parameters to default. Note that our shorter contig will not be included in the annotation because Prokka only annotates contigs of length at least 200. Select whether you would like an Email on completion, and click "Execute". The entire process of annotating our genome should not take more than a few seconds. Isn't bioinformatics great?

Prokka produces a collection of output files, which you might like to view. The .fna file contains the lone contig that survived. The .gff file contains regions identified by Prokka as genes. The .ffn file contains nucleotide sequences of the annotated genes, and the .faa file contains their translated amino acid sequences.

Exercise 14: First, view the .gff file containing regions identified by Prokka as putative genes. How many are there? What is the longest and shortest one?

Answer (3 points, one for each part): There are 9 putative genes. The longest has length 13335 – 118 + 1 = 13218 (give or take one); the shortest has length 27239-27054+1 = 186.

Prokka contains the annotation information in these files, but what a biologist would really like is a visualization of its genes and what they have been predicted to be. So to visualize our annotation contained in the .gff file, we will use a genome browser tool called "JBrowse" that is found in the "Graph/Display" section on the left side of the page.

When running JBrowse, take the following steps.

- *Reference genome to display*. Select "Use a genome from history" and choose the .fna file.
- Click "Insert Track Group".
- Click "Insert Annotation Track".
- Select your .gff file from the Prokka output.
- Under "Email notification", choose "Yes" if you like.
- Then click "Execute".

JBrowse should be very quick. When it is finished, click view file, which is an HTML file that we can view in the browser. (It may take a moment to load.)

It seems like nothing is there, but on the left side of the page, click on "Prokka on data XXX: gff" to show our beautiful annotation of the SARS-CoV-2 genome. Zoom out to see it in all its glory. All the arrows point in the same direction to indicate that the genes are all found on the same strand of the genome. This makes sense because SARS-CoV-2 is an RNA virus, meaning that its genome has only one strand. (It would have been a very bad sign if some genes pointed in the opposite direction.)

You can click on the genes as you like to obtain information about their length and what in the database they were found to be similar to (if anything). For example, if you click on the first protein (Replicase polyprotein 1a), you will find that it was found to be similar to the protein with Uniprot ID (https://www.uniprot.org/uniprot/P0C6U8), which is the same gene in SARS-CoV. Note that the annotation score of this protein is high because there is so much experimental evidence backing up this protein. This means that if we find a putative gene that is a hit against it, we can have a high degree of certainty that our gene serves a similar function.

Exercise 15: Why do you think that two of the genes are labeled "hypothetical proteins"?

Answer (4 points, be generous with partial credit): Because nothing with a known function aligned well against them, but they are a predicted gene because they start with a start codon and end with a stop codon.

It may amaze you that such a tiny thing can wreak so much havoc. But now that we know this annotation, we can start investigating the virus's individual genes. Which one should we focus on? How has the gene mutated as the virus spread around the world? And how do these mutations affect the function of the protein? We save these topics for the subject of future study.