SYSB 3036 W03: Gene Prediction

Version 2

Frank Aylward[1]

[1]Virginia Tech

dx.doi.org/10.17504/protocols.io.v38e8rw

Frank Aylward
Virginia Tech

PROTOCOL STATUS

Working

We use this protocol in our group and it is working

1 Let's start by downloading a Staphylococcus aureus genome from NCBI:

wget
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/585/GCF_000009585.1_ASM958v1/GCF_000009585.1_ASM958v1_genomic.fna.gz

and let's make sure to unzip it so that we can access the .fna file directly (gunzip command).
Make sure to use "head" and "tail" as we did in the W1 tutorial to ensure that the file is in FASTA format.

gunzip
ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/009/585/GCF_000009585.1_ASM958v1/GCF_000009585.1_ASM958v1_genomic.fna.gz

and

ls -lh

2 Now let's get some basic stats using seqkit so we know what we're dealing with:

seqkit stats GCF_000009585.1_ASM958v1_genomic.fna

and

seqkit fx2tab -l -n -g GCF_000009585.1_ASM958v1_genomic.fna

Based on these results we can see we're working with a 2.9 Mbp genome that is split into 4 replicons. The chromosome has the majority of the sequence, with 2.87 Mbp. The chromosome has a %GC content of ~33%, slightly higher than that of the plasmids.

3 Now we can start predicting genes using Prodigal. The main page for this tool is here:
https://github.com/hyattpd/Prodigal

To start, let's take a look at the help menu:

prodigal -h

We can see that this tool is quite a bit simpler than seqkit. For example, there are no sub-commands, just the main prodigal command. The tool also only has a few options. It takes a FASTA file of DNA as the input, and it will output a file of genes (nucleic acid), proteins (amino acid), and a table of results (usually in Gene Feature Format (GFF)). The main flags we need to be concerned with are -i, -d, -a, and -o.

4 Let's run prodigal such that we get genes, proteins, and a tabular GFF file.

prodigal -i GCF_000009585.1_ASM958v1_genomic.fna -a proteins.faa -d genes.fna -o output_table.gff

And make sure to run "ls" afterwards to ensure the right files have been created.

5 OK, now we need to quality-check the files that have been created. First thing is to go through and use the "head" command to take a quick peak, and then we can use "seqkit stats" to get more details.

seqkit stats genes.fna

and

seqkit stats proteins.faa

Note that we should get the same number of genes and proteins, since each gene will have a protein translation. Also note that gene sequences will be 3 times as long as their corresponding amino acid sequences, since each protein is encoded by a 3 nt codon.

6 We can also find the longest and shortest genes, and those with the highest and lowest GC content like in W1:

seqkit fx2tab -i -n -g -l genes.fna | sort -rn -k 2,2 | head

seqkit fx2tab -i -n -g -l genes.fna | sort -rn -k 3,3 | head

For the lengths we can do the same with the proteins file, and we should find the same order but with lengths X 1/3

seqkit fx2tab -i -n -g -l proteins.faa | sort -rn -k 2,2 | head

Note that the GC content is still calculated for proteins, even though this statistic is not useful here.

7 Since this Staphylococcus aureus genome is divided up among 1 chromosome and 3 plasmids, we may wish to know how many genes were predicted on each plasmid.
We know the plasmid names from runnin seqkit fx2tab:

seqkit fx2tab -l -g -n GCF_000009585.1_ASM958v1_genomic.fna

and we can find all genes on these plasmids with commands like the following:

seqkit fx2tab -l -g -n genes.fna | grep "NC_017334.1"

If we don't want to count we can pipe the above command into a "wc" command:

seqkit fx2tab -l -g -n genes.fna | grep "NC_017334.1" | wc

And if we want to get very fancy we can use grep to identify all genes EXCEPT those on the main chromosome with the following command:

seqkit fx2tab -l -g -n genes.fna | grep -v "NC_017333.1" | wc

Note the -v flag in grep indicates that all lines that do not match to the given pattern should be returned.
With this command we can see that only 14 of 2,691 genes are present on all of the plasmids combined. So the vast majority of genes are present on the chromosome.

8 So far we have only considered protein-coding genes, but genes that encode functional RNAs are very important and should be considered as well.

To predict rRNAs and tRNAs we will use the tool barrnap

Barrnap is a relatively simple tool to use. We can see the help menu with

barrnap -h

We essentially just need to give it the input FASTA file of the genome, and specify an output file with a ">" symbol.

barrnap GCF_000009585.1_ASM958v1_genomic.fna > rRNA.gff

This gives us a GFF file with the coordinates of the predicted RNA genes, but it does not give us the sequences. To retrieve them we need to use a tool called BEDtools, which will cross reference the genome file with the GFF file and give us a FASTA file of all of the rRNA genes. The command is:

bedtools getfasta -fi GCF_000009585.1_ASM958v1_genomic.fna -bed rRNA.gff -fo rRNA.fasta

9 The rRNA.fasta file should have our rRNA genes. Let's see how many we found:

seqkit stats rRNA.fasta

seqkit fx2tab -n -l -g rRNA.fasta

and let's say we wanted to get the sequence of the first 16S rRNA gene:

seqkit fx2tab rRNA.fasta | grep "NC_017333.1:564549-566098"

Retrieving the sequence of a 16S rRNA gene can be very helpful if you are unsure of the taxonomy of the genome that you are examining. These genes can be compared to various 16S databases to help identify a microbe.