Introduction to Prokaryotic gene prediction (CDS and rRNA)

Version 2

Frank Aylward

Abstract

This is a short tutorial on how to get started with gene prediciton in Prokaryotic genomes via the command line.

Code is intended for use on an Ubuntu 16.04 LTS OS.

The tools we will use here are:

Prodigal. https://github.com/hyattpd/Prodigal

Barrnap. https://github.com/tseemann/barrnap

This tutorial is intended for teaching purposes, but if you use any of the tools in a scientific paper do not forget to cite the appropriate publications!

Prodigal:

Prodigal: prokaryotic gene recognition and translation initiation site identification.

BMC Bioinformatics, 2010, Volume 11, Number 1, Page 1

Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, Loren J Hauser

Barrnap:

Seemann T (2013), barrnap 0.8 : rap. id ribosomal RNA predictionhttps://github.com/tseemann/barrnap

Bedtools

BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, Volume 26, Issue 6, 15 March 2010, Pages 841–842, https://doi.org/10.1093/bioinformatics/btq033

Citation: Frank Aylward Introduction to Prokaryotic gene prediction (CDS and rRNA). **protocols.io** dx.doi.org/10.17504/protocols.io.pjrdkm6
Published: 25 May 2018

Protocol

Download a Prokaryotic genome to start analyzing

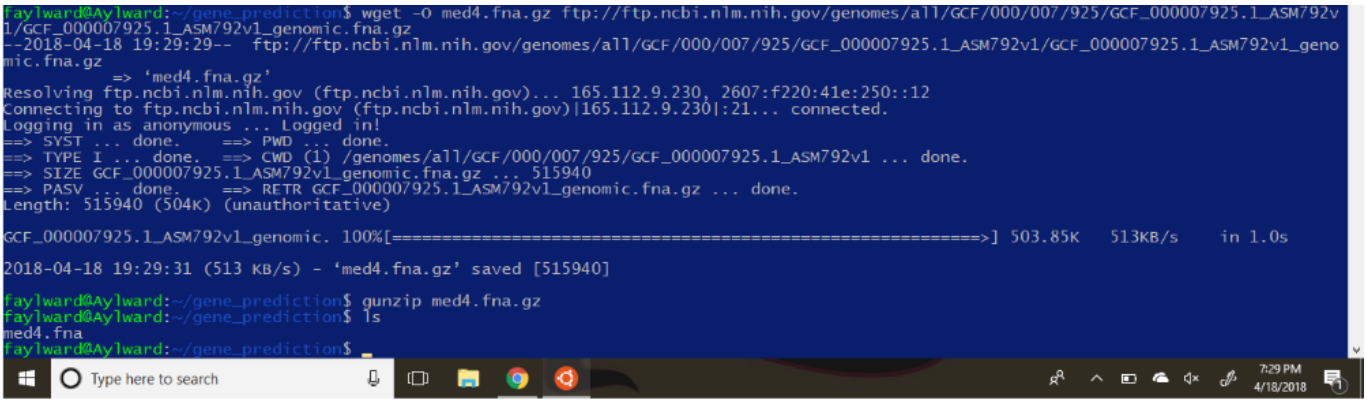
Step 1.

Here we'll be working with *Prochlorococcus marinus* MED4.

wget -O med4_genome.fna.gz ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/007/925/GCF_000007925.1_ASM792v1/GCF_000007925.1_ASM792v1_genomic.fna.gz

gunzip med4_genome.fna.gz

You should see something like this:



Predict protein-coding genes using Prodigal

Step 2.

To install Prodigal (on Ubuntu) try:

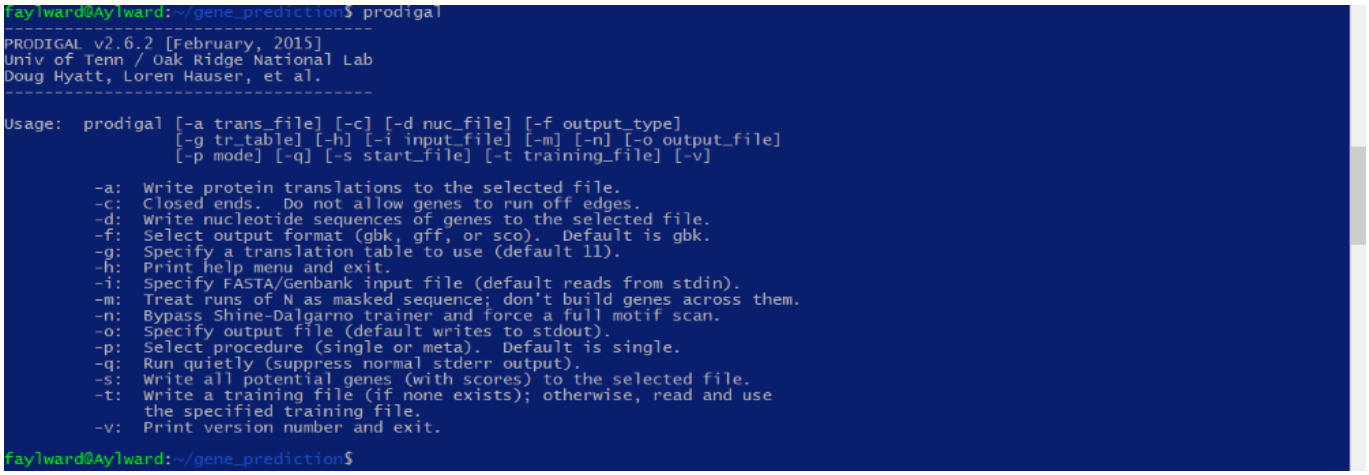
sudo apt install prodigal

You may be asked to confirm that you want to install- for this just type "Y".

For a quick description of the usage and flags you can simply type:

prodigal

And you should see these usage instructions:



Prodigal will predict genes from chromosomes (or contigs), translate those genes into amino acids, and produce annotation summary files such as gff, depending on what options you use.
prodigal -i med4_genome.fna -a med4.proteins.faa -d med4.genes.fna -f gff -o med4.prodigal.gff

Take a look at the results

Step 3.

It's always good to take a look at the output to get an idea of the format, whether it's what you expect, etc.

Let's start with a simple 'head' command.

head med4.proteins.faa

and

head med4.prodigal.gff

```
faylward@Aylward:~/gene_prediction$ head med4.proteins.faa
>NC_005042.1_1 # 174 # 1331 # 1 # ID=1_1;partial=00;start_type=ATG;rbs_motif=None;rbs_spacer=None;gc_cont=0.309
MKLVCSQIELNTALQLVSRAVATRPSHPVLANVLLTADAGTGKLSLTGFDLNLGIQTSLS
ASIESSGAITVPSKLFGEIISKLSSESSITLSTDDSSEQVNLKSKSGNYQVRAMSADDFP
DLPMVENGAFLKVNANSFAVSLKSTLFASSTDEAKQILTGVNLCFEGNSLKSAATDGHRL
AVLDLQNVIASETNPEINNLSEKLEVTLPSRSLRELERFLSGCKSDSEISCFYDQGQFVF
ISSGQIITTRTLDGNYPNYNQLIPDQFSNQLVLDKKYFIAALERIAVLAEQHNNVVKIST
NKELQILNISADAQDLGSGSESIPIKYDSEDIQIAFNSRYLLEGLKIIETNTILLKFNAP
TTPAIFTPNDETNFVYLVMPVQIRS*
>NC_005042.1_2 # 1333 # 2079 # 1 # ID=1_2;partial=00;start_type=TTG;rbs_motif=None;rbs_spacer=None;gc_cont=0.305
MNIPSQFLLSNLLKLNARCSNGIDHGIGTTAWMYPPVHRLLGWASRPSKLSLKRSVWRLD
faylward@Aylward:~/gene_prediction$
faylward@Aylward:~/gene_prediction$ head med4.prodigal.gff
##gff-version  3
# Sequence Data: seqnum=1;seqlen=1751080;seqhdr="NC_005042.1 Prochlorococcus marinus subsp. marinus str. CCMP1375 complete genome"
# Model Data: version=Prodigal.v2.6.2;run_type=Single;model="Ab initio";gc_cont=36.44;transl_table=11;uses_sd=0
NC_005042.1     Prodigal_v2.6.2 CDS     174     1331    173.3   +       0       ID=1_1;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.309;conf=100.00;score=173.27;cscore=170.01;sscore=3.26;rscore=0.34;uscore=-0.28;tscore=3.20;
NC_005042.1     Prodigal_v2.6.2 CDS     1333    2079    105.4   +       0       ID=1_2;partial=00;start_type=TTG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.305;conf=100.00;score=105.43;cscore=109.75;sscore=-4.32;rscore=0.34;uscore=1.14;tscore=-7.06;
NC_005042.1     Prodigal_v2.6.2 CDS     2070    4487    410.9   +       0       ID=1_3;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.336;conf=99.99;score=410.89;cscore=404.54;sscore=6.35;rscore=0.34;uscore=1.58;tscore=3.20;
NC_005042.1     Prodigal_v2.6.2 CDS     4530    5987    225.6   +       0       ID=1_4;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.335;conf=99.99;score=225.64;cscore=218.56;sscore=7.08;rscore=0.34;uscore=2.72;tscore=3.20;
NC_005042.1     Prodigal_v2.6.2 CDS     5992    8478    384.2   -       0       ID=1_5;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.316;conf=99.99;score=384.16;cscore=381.55;sscore=2.60;rscore=0.34;uscore=-0.94;tscore=3.20;
NC_005042.1     Prodigal_v2.6.2 CDS     8558    9433    104.6   -       0       ID=1_6;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.298;conf=100.00;score=104.59;cscore=95.49;sscore=9.10;rscore=0.34;uscore=4.27;tscore=3.20;
NC_005042.1     Prodigal_v2.6.2 CDS     9433    10395   131.3   -       0       ID=1_7;partial=00;start_type=ATG;rbs_motif=None;rbs_sp
acer=None;gc_cont=0.333;conf=100.00;score=131.33;cscore=128.03;sscore=3.30;rscore=0.34;uscore=0.41;tscore=3.20;
faylward@Aylward:~/gene_prediction$
```

A closer look at the results

Step 4.

If you want more stats about the output FASTA files you can also use a nifty tool called seqtk. To install on Ubuntu type:

sudo apt install seqtk

and then try:

seqtk comp med4.proteins.faa | head

```
faylward@Aylward:~/gene_prediction$
faylward@Aylward:~/gene_prediction$ seqtk comp med4.proteins.faa | head
NC_005042.1_1   386     28      4       18      25      89      46      176     0       70      19      0
NC_005042.1_2   249     11      2       10      11      68      34      113     0       47      21      0
NC_005042.1_3   806     56      14      70      34      171     109     352     6       120     51      4
NC_005042.1_4   486     30      4       37      22      133     70      190     16      85      48      9
NC_005042.1_5   829     42      5       58      34      196     85      409     2       140     56      2
NC_005042.1_6   292     29      2       10      10      77      11      153     2       63      14      1
NC_005042.1_7   321     21      9       17      8       102     29      135     4       81      21      1
NC_005042.1_8   250     11      0       15      21      53      28      122     0       36      17      0
NC_005042.1_9   212     11      2       5       10      55      27      102     0       31      24      0
NC_005042.1_10  432     39      1       26      22      101     59      184     0       83      18      0
faylward@Aylward:~/gene_prediction$
```

In the last command I piped the output to a head command so we could look at the first 10 entries. Seqtk is nice because it's pretty fast and gives you the length of the sequences (second column) and other information like the sequence composition. However, note that seqtk usually operates on nucleotide sequences, so the composition stats that are given are not valid for protein files (i.e., A in a protein sequence stands for Alanine, but it's being counted as if it's an Adenine here- the result would be some pretty funky composition info if you thought the file contained nucleotide sequences of genes!).

Seqtk could also be used on the predicted genes file, in which case we could trust the composition stats:

seqtk comp med4.genes.fna | head

To count the number of proteins that were predicted, we could also try:

seqtk comp med4.proteins.faa | wc -l

This would give the number of lines that the seqtk command produced, which should be equivalent to the number of sequences in the file.

We won't go over seqtk's other options here, but there are some other useful utilities in this tool. To check them out just type:

seqtk

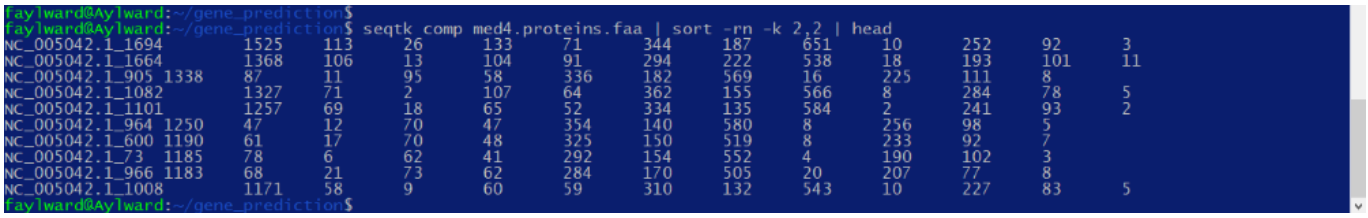
Get the longest or shortest protein

Step 5.

As a side-note, let's say you wanted to find the longest protein predicted in the genome of *Prochlorococcus*. This can be easily retrieved using the following command:

seqtk comp med4.proteins.faa | sort -rn -k 2,2 | head

You should find a protein called NC_005042.1_1694 on the top, with a length of 1525 (second column)



Now, what's happening in that last command is that the results of seqtk are being piped into the unix sort command. For sort, the flags -rn indicate that we want a reverse (-r) numeric (-n) sort. The second column of the seqtk output gives the lengths, and to specify that we want to sort by this we can give the -k 2,2 command (2,2 specifies that the sorting stars and ends on the second column, as opposed to using multiple columns to sort). Lastly, we pipe this into another head, so we should see the top 10 longest proteins now.

If you wanted the shortest proteins you could either replace 'head' with 'tail' or remove the -r flag, so the sorting is in ascending order.

The unix sort command is super handy, and for more info you can always type:

man sort

Now back to Prodigal

Step 6.

Note that in addition to the protein and gene files we also got a gff file, which is a Gene Feature Format file. This gives information on the length and location of the genes that were predicted. We won't go into much detail about this here, but GFF files are handy for downstream applications like read mapping. You can read more about this format here:
https://useast.ensembl.org/info/website/upload/gff.html

With prodigal you can also make Genbank format file instead of a GFF file, if you wish.
prodigal -i med4_genome.fna -a med4.proteins.faa -d med4.genes.fna -f gbk -o med4.prodigal.gbk

What about rRNA genes?

Step 7.

So far we have been focusing on protein-coding genes, but there are other genes we will also want to predict for a more complete genome annotation.

Barrnap is a program that is useful for predicting rRNA genes. To install:

sudo apt install barrnap

And to run:
barrnap med4_genome.fna > med4.rRNA.gff

Parsing the barrnap GFF output

Step 8.

Barrnap only provides the summary files (in this case gff). So we need to do a bit more work to get the actual sequences.

One very useful tool or parsing GFF files is called BEDtools. To install:

sudo apt install bedtools

There are many different utilities in bedtools. Here we will want to use the "getfasta" option, which will allow us to supply the fasta file of the Prochlorococcus genome and the barrnap GFF file to obtain the rRNA sequences. Note that the GFF file has the coordinates of where the rRNA genes are encoded, so between the GFF file and the .FNA file we have all the information we need.
bedtools getfasta -fi med4_genome.fna -bed med4.rRNA.gff -fo med4.rRNA.fasta

16S rRNA genes are extremely useful for classification. If you ever have a genome and you don't know what it is, a good first step is to identify any 16S ribosomal genes in the chromosome and use them for classification.