Gene Prediction in Prokaryotes and Eukaryotes

Julie Boisard, Courtney Stairs, Dag Ahren, Björn Canbäck, Joel Wallenius

2024-01-11

Contents

1 Gene prediction: prokaryotes

1.1 Learning objectives and reminders

Learning objectives:

- Install and run Prodigal for gene prediction of prokaryotic genomes
- Interpret genome characteristics using Prodigal
- Analyze tRNA complement of a bacterial genome

Before you begin:

- Log into your machine on campus using `ssh -X`.
- Do you remember how to copy files between your personal computer, your machine, and the biology server? Need help? Ask us or Google.

1.2 Installation of `prodigal` and `aragorn`

1.2.1 `prodigal`

Install the prokaryotic gene predictor `prodigal` from GitHub:
https://github.com/hyattpd/Prodigal

Use the "Clone or download button" and then "Download ZIP" and save the zip-file to your `bin` directory.

```
cd ~/bin
unzip Prodigal-GoogleImport.zip
cd Prodigal-GoogleImport
make
```

Move, link or copy the `prodigal` executable to the `bin` directory.

1.2.2 `aragorn`

Install the tRNA predictor `aragorn` from:
http://www.ansikte.se/ARAGORN/
Don't forget to compile the program according to the instructions on the web page. Don't forget to download the manual. Copy or link the binary and manual to your `bin` directory.

1.3 Predicting protein-coding genes

On your local machine make a new directory for the gene prediction exercises

```
1 mkdir -p ~/GenePrediction/Bacteria
```

The `-p` allows for the creation of both directories at the same time.
It will create a directory "Bacteria" in another new directory "GenePrediction" in you home (~) directory.

Copy the `spades` output file from the server to the above directory and name it `geo.fna` for convenience. The source name should be `MyFirstAssembly_LargeContigs_out_AllStrains.unpadded.fasta`

Before running the gene prediction, consider the following:

1. What translation table should be used. Link:
   http://www.ncbi.nlm.nih.gov/taxonomy and dig deeper.

`Prodigal` has a few options to investigate. The normal way of getting help for a program is to add a `-h` or `-help` argument to the program name. Here we use `prodigal -h`. Try it.

We will use the following options: `-a`, `-c`, `-d`, `-f` (gff), `-g`, `-i`, `-m`, `-o` and `-t`.

Try to run `prodigal` with these options.

Name the fasta output files as `geoGenes.fna` and `geoProteins.faa` and the gff-file (-o) as `geo.gff` and the produced training file as `geoTraining` (-t).

2. What is the purpose of the `-c` and `-m` options?

The first time `prodigal` is run, a training set is produced:

```
ls -l # Only the training file contains data
```

In the second run, the training set file is used to find other genes. Rerun `prodigal` with exactly the same arguments as before.

3. Are there any non standard nucleotides in the genes? Check the meaning of the non standard nucleotides at:
   http://en.wikipedia.org/wiki/Nucleic_acid_notation
4. Make a frequency table of start codons. What amino acids do they encode?
5. Is the number of genes similar in your gene prediction to that of the reference genome?
6. How many sequences are there in the forward strand and reverse strand?
7. Do the genes have higher or lower GC content as compared to the genome?
8. What is the gene density (how much of the genome consists of genes)? Notice that we do not include non-protein coding genes yet.

1.4 Stable RNA genes

Stable RNA genes are mainly found in two categories, tRNAs and rRNAs. In bacteria there is a third class, tmRNAs.

9. Read about tmRNAs at:
   http://en.wikipedia.org/wiki/Transfer-messenger_RNA

We will use the software `aragorn` for the detection of tRNAs. Read the manpage:

```
aragorn -h
```

For this prediction only the `-t` option should be used. You could also try the `-w` option.

9. How many tRNAs are predicted in your assembly? Look at the last lines in the output files.
10. Do you find at least one tRNA for each amino acid?
11. How many anticodons do you find? Why not 61?
12. Do you find any tmRNA (use -m)?

2 Gene prediction: eukaryotes

2.1 Learning objectives

- Run GeneMark for gene prediction
- Analyze GC content of a scaffold
- Learn how to interpret and extract information from GFF files

2.2 Check GeneMark installation

The eukaryotic gene predictor `GeneMark` is already installed on the server Read-more about the tool here: http://exon.gatech.edu/GeneMark/.
`GeneMark` has `perl` dependencies that only the system administrator can install. Thisis an example of a case when `conda` cannot be used.

2.3 Familiarize yourself with the *Paxillus involutus* genome

This exercise will be completed on the course server. Create a directory `EukaryoticGenePrediction` on the server.

Paxillus involutus is a common fungus in the woods of the northern hemisphere (see http://en.wikipedia.org/wiki/Paxillus_involutus). The genome has been sequenced by JGI (Joint Genome Institute). The genome fasta sequence (the scaffold sequences) are found in the `Data` folder on the course server with the name `Paxin1_AssemblyScaffolds_Repeatmasked.fasta.gz`.

The assembly is "masked" which means that repeats have been removed from the assembly. In this case, the masking includes ribosomal RNA genes. Download the file, copy it to the server to `EukaryoticGenePrediction` and rename it to `paxillusGenome.fna`.

15. How big is the genome and what GC content does it have? To calculate the number of g:s and c:s there is a smart combination of options to the `tr` command:

```
cat paxillusGenome.fna | grep -v \> | tr -cd cgCG | wc -c
```

16. 454 paired end sequencing has been used (in Illumina corresponding to mate pairs). What percentage of the assembled genome sequence is not known (containing Ns)?

2.4 Predicting genes from one *Paxillus involutus* scaffold using GeneMark

Gene prediction takes a long time. Therefore we will only work with the sequence for scaffold 1. We start by extracting that sequence:

```
grep -m 2 -n scaffold paxillusGenome.fna # To get line numbers
head -56224 paxillusGenome.fna > scaffold1.fna
# Alternatively with awk:
awk ' /^>/ {++i} {if (i==2) {exit}} {print} ' paxillusGenome.fna > scaffold1.fna
```

`-m 2` tells `grep` to only report the two first matching lines while `-n` makes `grep` to also output the current line number.

To run `GeneMark` (gm) a license key is needed. You get the key from the web site when you are downloading the software. For simplicity we all use the same key. The key has to be directly into your home directory. You can check to make sure you have the key:

```
ls ~/.gm_key
```

If you do not find the key file. Please copy it:

```
cp /resources/binp28/Programs/Genemark/gm_key_64 ~/.gm_key
```

Take a look at the options by simply calling 'gmes_petap.pl:

```
gmes_petap.pl
# In this program -h is not understood.
```

Start the gene prediction performed by `GeneMark`. Also check that it is running:

```
nohup gmes_petap.pl --ES --sequence scaffold1.fna &
# Then hit return and you get the prompt back
top     # check that it is running
u user  # to list only processes that the user runs
c       # display the entire command line
q       # quit top
```

Remember that `nohup` lets a program run even if internet connection is lost. The ampersand puts the process in the background so the terminal is released (you get the prompt back). The following code is optional:

```
# To get the process in foreground again
fg
# To get it back to background
Ctrl-z # Pauses execution
bg
```

It will take about 15 minutes so please take a coffee break (tea is also allowed).

2.5 Working with GFF files

`GeneMark` doesn't give a lot of information about the output files. Transcripts or genes are not found in fasta files, neither are the proteins. Look at the `gff` file produced by `GeneMark`. It is named `genemark.gtf`. This is a standard format for annotations and the `gff` file is often called *(genome) annotation file*. Take a look at the specification at:
http://www.sanger.ac.uk/resources/software/gff/spec.html at Sanger.
The web page warns you about using the format `gff2` (instead `gff3` should be used) which is the format output by `GeneMark`. Still `gff2` and the nearly identical `gtf` format are widely used. We will look more at `gff` files in the transcriptome exercises.

17. How many genes were predicted in scaffold 1?
18. How many exons are there in average in scaffold 1?
19. Optional (difficult): What is the average intron size in scaffold 1?
20. Run ARAGORN on the complete genome file and compare the number of tRNAs to that of "your" *Geobacillus*. Discuss gene dosage and expression.

If we want fasta sequence files we have to use a program that takes a genome sequence file as one argument and a `gff` file as another. It should extract the positions of the coding sequences (CDS) in the scaffolds from the `gff` file and then extract the corresponding sequences from the genome file. If there are more than one CDS (coding sequence) for a gene, the program should concatenate the CDS sequences to one sequence. Note: This software will work in this case, but if there is alternative splicing it will not work.

Copy the program `gffParse.pl` from the `/resources/binp28/Data/` folder on the server into your `bin` directory on the server. Don't forget to `chmod` it so it becomes executable. Read about the options:

```
gffParse.pl -h
#./gffParse.pl -h
#perl gffParse.pl -h
```

Then run it:

```
gffParse.pl -i scaffold1.fna -g genemark.gtf -f CDS \
-b paxillus -p -a gene_id
```

We will now use this fasta file for BLAST in the next exercises.