2.3 Identify rRNA sequences in DNA

For identifying rRNA sequences in DNA we will use `rnammer`, a program that implements an algorithm designed to find rRNA sequences in DNA [5]. The program was made by modeling a large number of known rRNA sequences and making them into generalized patterns. These patterns are then used as search models in a new DNA sequence. If a part of the DNA matches the model, the sequence is extracted as a likely rRNA sequence. The help page for `rnammer` has the following description for the program:

> "RNAmmer predicts ribosomal RNA genes in full genome sequences by utilizing two levels of Hidden Markov Models: An initial spotter model searches both strands. The spotter model is constructed from highly conserved loci within a structural alignment of known rRNA sequences. Once the spotter model detects an approximate position of a gene, flanking regions are extracted and parsed to the full model which matches the entire gene. By enabling a two-level approach it is avoided to run a full model through an entire genome sequence allowing faster predictions.
>
> RNAmmer consists of two components: A core Perl program, 'core-rnammer', and a wrapper, 'rnammer'. The wrapper sets up the search by writing on or more temporary configuration(s). The wrapper requires the super kingdom of the input sequence (bacterial, archaeal, or eukaryotic) and the molecule type (5/8, 16/17s, and 23/28s) to search for. When the configuration files are written, they are parsed in parallel to individual instances of the core program. Each instance of the core program will in parallel search both strands, so a maximum of 3x2 hmmsearch processes will run simultaneously. The input sequences are read from sequence and must be in Pearson FASTA format. "

The program is run as follows, specifying the taxonomical kingdom (bac) and the type of molecules to search for (tsu for 5/8s rRNA, ssu for 16/18s rRNA, lsu for 23/28s rRNA). The parameter -f specifies the name of the output file.

Listing 2.5: Identify 16S rRNA sequences in genomic DNA

```
# Syntax:
$ rnammer -S bac -m ssu -f <name>.rrna <name>.gbk.dna
# Example, one line command:
$ rnammer -S bac -m ssu -f Neisseria_meningitidis_Z2491_ID_AL157959.gbk.dna.rrna
     Neisseria_meningitidis_Z2491_ID_AL157959.gbk.dna
```

Note that this takes time to go through a single genome, and if there is more than one rRNA operon (which is true for many bacteria), then multiple 16S rRNA sequenced will be found per genome. See the example below, where first the *N. meningitidis* genomes are done, and then the *N. gonorrhoeae* genomes are done.

Listing 2.6: Identify 16S rRNA sequences in genomic DNA - loop

```
for x in Neisseria*gbk.dna
> do
> echo $x
> rnammer -S bac -m ssu -f $x.rrna $x
> done
```

Have a look at the FASTA header line for the 16S rRNA sequences and look at the score and the lengths. Use `grep` to get the header lines for each file.

2.3.1 Exercises

1. Run `rnammer` on all `*.gbk.dna` files
2. The output files from RNAmmer are FASTA formatted; count the number of sequences in each file, why are there multiple entries?
3. Does the program only find 16S rRNA sequences? What type of molecules has the algorithm found?
4. Are there some files that are empty? What does it mean if the file is empty?
5. What kind of information can you get from the header lines of these FASTA files?

2.4 Multiple sequence alignment of 16S rRNA sequences

One way of comparing the 16 sRNA sequences is to do a multiple sequence alignment. Multiple alignments are often used in identifying conserved sequence regions across a group of sequences hypothesized to be evolutionarily related. If two sequences in an alignment share a common ancestor, mismatches can be interpreted as point mutations and gaps as "indels" (that is, insertion or deletion mutations) introduced in one or both lineages in the time since they diverged from one another. In this course we use `clustal` to align the sequences and find the distance/differences between them [6]. The greater the distance between two sequences the greater the difference between the organisms from which the sequences came. The `rnammer` program finds all possible rRNA sequences in a genome. Some, and indeed many, genomes have more than one copy of this operon. In order to do a 16S rRNA tree you should pick one sequence. Here we choose the one that has the highest score according to the `rnammer` models. The program `rnammer_extractseqs` takes all files in a directory named `*.rrna` and selects the best sequence within each file (each organism). For some organisms, `rnammer` might not find a sequence with a good enough score. This means that the model does not find a sequence that looks sufficiently like a rRNA sequence. These sequences, and organisms, will be excluded from the analysis.
The sequences will now be evaluated based on fixed criteria for a 16S rRNA sequence. These criteria include length and fitness score to the `rnammer` models. This extraction procedure will only work for 16s rRNA sequences that fulfill the criteria.

Listing 2.7: Identify 16S rRNA sequences in DNA - select sequences

```
# The following code works on all files in the current working directory.
# Syntax:
$ rnammer_extractseqs <allfile>.RRNA
$ rnammer_extractseqs_allLengths <allfile>.RRNA
# Example:
$ rnammer_extractseqs all.RRNA
$ rnammer_extractseqs_allLengths allLengths.RRNA
```

The output is a FASTA formatted file with rRNA genes in DNA code. Count the number of sequences in this FASTA file (using `grep`). Try using `grep` without the `-c` option and look at what the headers of this FASTA file looks like. Look at the header lines for the selected sequences. The alignment is performed using `clustalw` and the file holding one rRNA sequence from each organism.

Listing 2.8: Multiple alignment of 16S rRNA sequences

```
# Syntax:
$ clustalw <allfile>.RRNA
# Example:
$ clustalw all.RRNA
# The following files are created:
all.aln
all.dnd
```

Next we create a distance tree for the alignment, using a bootstrap value of 1000. Here is a quote from Wiki about bootstrapping in statistics:

> " In statistics, bootstrapping is a computer-based method for assigning measures of accuracy to sample estimates (Efron and Tibshirani 1994). This technique allows estimation of the sample distribution of almost any statistic using only very simple methods (Varian 2005). Generally, it falls in the broader class of resampling methods. Bootstrapping is the practice of estimating properties of an estimator (such as its variance) by measuring those properties when sampling from an approximating distribution. One standard choice for an approximating distribution is the empirical distribution of the observed data. In the case where a set of observations can be assumed to be from an independent and identically distributed population, this can be implemented by constructing a number of resamples of the observed dataset (and of equal size to the observed dataset), each of which is obtained by random sampling with replacement from the original dataset. "

When using bootstrapping, different versions of the distance tree will be constructed and each time a branching point will be recorded. With a bootstrap value of 1000, the number on the tree will indicate how many, out of 1000 trees, have this branching. The closer to 1000 the more sure we are of that branch.

Listing 2.9: Multiple alignment of 16S rRNA sequences - bootstrap

```
# Syntax:
$ clustalw <allfile>.RRNA -bootstrap=1000
# Example:
$ clustalw allLengths.RRNA -bootstrap=1000
# The following files are created:
all.phb
```

The tree construction is simply a drawing program called `njplot` [9].

Listing 2.10: View 16S rRNA tree

```
$ njplot all.RRNA.phb
```

Open the bootstrap tree file `*.phb` and tick the `Display` setting by clicking `Bootstrap values`. Click `"File"` and `"Save Rooted tree"`. Output file is names `all_root.phb`. Under `"File"`, save the tree as a PDF format and open a word processor (Word, Pages) or presentation software (PowerPoint, Keynote) on your LOCAL computer. Get the PDF from the shared folder and put the picture into your presentation. Add colors to indicate taxonomic groupings or others clusters (See Figure 2.2).

2.4.1 Exercises

1. Extract the best scoring sequences within the length criteria for all `*.rrna` files and store in file `all.RRNA`
2. How many sequences are there in the `all.RRNA` file?
3. Extract the best scoring sequences for all `*.rrna` files and store in file
4. How many sequences are there in the `all.RRNA` file?
5. Run a multiple alignment on selected sequences
6. Generate a phylogenetic tree
7. Save tree as PDF and insert into presentation

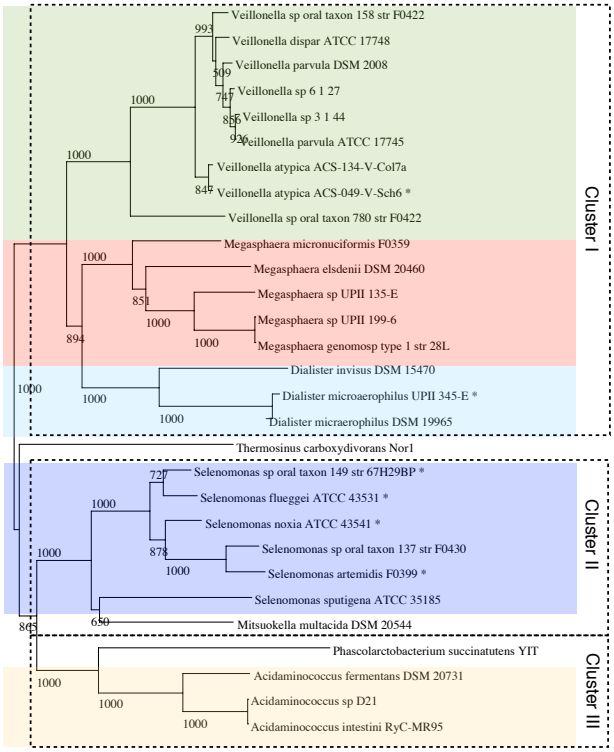

Figure 2.2: **16S rRNA tree.** Each genome sequence was searched for 16S rRNA patterns and candidate sequences were extracted. The best sequence from each genome was selected. For two genomes, no sequences were found, *Centipeda periodontii* DSM 2778, *Megamonas hypermegale* ART12 1. For 6 additional genomes, the located sequences were shorter than the default acceptable length. The short sequences sequences are marked with a "*". Length criteria was changed from minimum 1 400 to 1 100 and maximum 1 800 unchanged. The distance tree was made with 1 000 bootstraps.