Assembling genomes

Max Carter-Brown
Anglia Ruskin University, Wellcome Sanger Institute
max.carter-brown@aru.ac.uk

Mark Blaxter
Wellcome Sanger Institute, University of Edinburgh

Contents

1. Introduction 1
2. CAP3 2
3. The raw data 2
   3.1. Understanding the data 3
4. Assembling your reads 3
   4.1. Open a browser window and go to the PRABI CAP3 server 3
   4.2. Submit your sequence data to the CAP3 server 3
   4.3. Select “SUBMIT” 4
5. Examine your results 4
   5.1. The “Contigs” file 4
   5.2. The “Single sequences” file 4
   5.3. The “Assembly details” file 4
6. Final questions 5
Bibliography 6

This practical is assessed

A Canvas quiz has been set up for this practical. We will undertake the practical in examination conditions, and you will **not be able to talk during the session**. In addition, you must only have the tabs open on your computer which are relevant to the practical. If there are any other tabs open, or you are seen to be using ChatGPT or any other LLM, **you will instantly fail with a score of zero marks**.

Answer all of the questions in the yellow question boxes, these are equivalent to those on Canvas.

1. Introduction

You have the full two hours to complete this assessment, there should be plenty of time. Please read all the introductory material before starting the assessment.

In this practical assessment you will be taking a set of sequence reads derived from a mitochondrial genome, and using one of the many available genome assembly programs to “assemble” these sequence reads into a complete genome. The mitochondrial genome is circular (in animals) and ~14000 base pairs in length. It encodes 2 ribosomal RNAs and 12 or 13 proteins (depending on which species is being examined). It also encodes 20 transfer RNAs (tRNAs).

We will look at *Brugia malayi*, a parasitic roundworm (nematode) which infects humans, amongst other animals. The *B. malayi* mitochondrial genome is a circular molecule, with 12 protein coding genes and two rRNAs, just under 13,600 base pairs in length. The sequencing reads were generated

by the Blaxter lab in the Institute of Evolutionary Biology, University of Edinburgh, as part of the *B. malayi* genome project.

The reads are from Sanger sequencing (or dideoxy) technology. The reads are each 300-600 bases long, and are paired-end reads from cloned inserts in plasmid vectors. This is a fairly old sequencing technology - the one that was used to assemble the original human genome - but the data is good quality. The reads have been pre-trimmed "low quality" sequence. The data is described further in a further section.

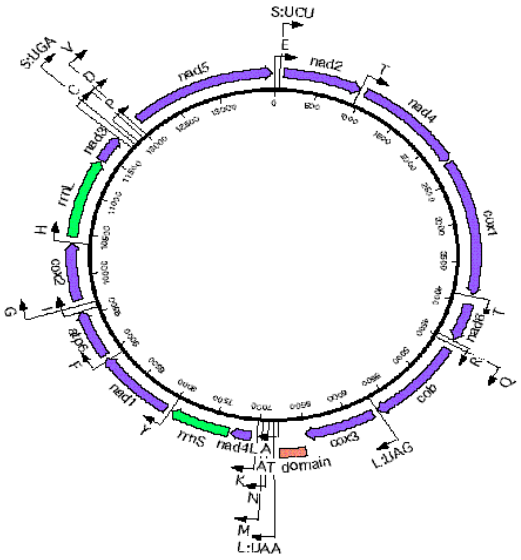

Figure 1: The *B. malayi* mitochondrial genome showing the position of the genes in the circular molecule.

2. CAP3

You will use the program "CAP3" (which stands for Contig Assembly Program). CAP was developed by Xiaoqiu Huang. It was first published in 1992 [1]. A brief description of how the program works is given below.

CAP3 is an "alignment-overlap-consensus" assembler. It first compares all the sequences to each other to identify which could possibly derive from the same portion of the genome. It then develops an overlap map of the sequences, an estimate of how the sequences could be tiled together to make an assembly. If a particular sequence could overlap two others, and these overlaps are in conflict, the program uses the quality of the alignments (length, number of identities) to choose between them. Finally, CAP3 derives a consensus sequence from the overlapped sequences, and produces a set of reports.

It is an old, and fairly naive assembler by today's comparison but it works well for the dataset that we have.

3. The raw data

You should now open your data file `BM_mt.fa`. It is a fasta file, an extremely common file format in bioinformatics which we have come across before. Open your file **in a text editor**, not Word. Word will add characters to the file which will make it unreadable by the assembler.

Questions

1. What defines the fasta format?
2. How many sequences are in the raw data file? Consider using `grep`, or another pattern matching tool.

3.1. Understanding the data

The sequences in the file are derived from performing sequencing on shotgun clones of the *B. malayi* genome. Each clone was named `Bmmt_cl_X` where `X` is a number. Each clone was sequenced twice, using primers at the start and the end of the insert (Figure 2). One sequence will represent the forward read (`_f`), and the other the reverse read (`_r`), which will be from the opposite side of the strand.

This type of sequencing is called paired-end sequencing. Forward and reverse sequences are not present for every clone, so some clones are represented by either `_f` or `_r` reads.

Thus, for example, `Bmmt_cl_123_f` and `Bmmt_cl_123_r` form a "read pair" from each end of a single clone.

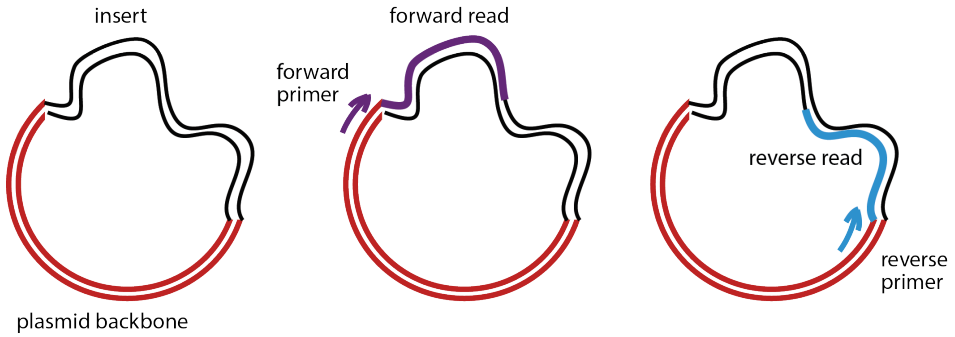

Figure 2: A read pair in a plasmid vector. The forward and reverse reads are generated from either end of the insert. The insert is between two primers which are used to amplify the DNA to be sequenced.

4. Assembling your reads

We will use an online service that offers access to the CAP3 assembly program. In the online version, a form is used to collect your data, and the webserver then formats and executes a command on the remote computer to run the program. Once it is finished, the webserver gives you access to the output files from the analysis.

4.1. Open a browser window and go to the PRABI CAP3 server

Open a link to the CAP3 server at PRABI-Doua (http://doua.prabi.fr/software/cap3) **in a new browser window**.

Further reading...

The PRABI-Doua (Pôle Rhône-Alpes de Bioinformatique Site Doua) World Wide Web server - developed at the Laboratory of Biometry and Evolutionary Biology and the Institute of Biology and Chemistry of Proteins - is dedicated to molecular biology and ecology.

4.2. Submit your sequence data to the CAP3 server

Select all of the sequence data in FASTA format from `BM_mt.fa` and paste it into the window "Enter your sequences in FASTA format".

Leave the various analysis options at their default values.

4.3. Select “SUBMIT”

Your data will be sent to the PRABI remote computing facility which will carry out the assembly. Please be patient while this happens: do not keep clicking on the page while it is loading. This may take up to 5 minutes depending on the server load.

5. Examine your results

Several text outputs are produced.

1. **Contigs**
2. **Single sequences**
3. **Assembly details**
4. **Your sequence file** (this is just a copy of the input file)

These are all hyperlinks which link to text files. Save the files to your local workspace for examination.

5.1. The “Contigs” file

This contains the assembled sequences in fasta format. A “contig” is a set of contiguous sequences.

Questions

3. What is the total number of contigs produced by the assembler?
4. What is the length of the largest contig?
5. What is the total length of the assembly?

Note that each sequence line in the fasta is 60 bases long. You may wish to put into practice your BASH/shell/terminal skills. `grep` will be useful in counting contigs. `wc` will be useful in counting bases (once the headers are excluded). Of course you can count in any way you like - such as using Word tools.

5.2. The “Single sequences” file

Singletons are sequences that did not assemble.

Questions

6. How many singletons are there?

5.3. The “Assembly details” file

This file gives an overview of how the sequences were put together from the original input file.

The read names are given, followed by a `+` or a `-`. These symbols indicate the orientation of the read in the contig. The reads are given in the order in which their first aligned base appears in the assembly contig, reading from left to right through the contig.

Below is a snippet from the output.

```
****************** Contig 2 ********************
Bmmt_cl_4_r+
Bmmt_cl_19_r-
                    Bmmt_cl_20_f+ is in Bmmt_cl_19_r-
Bmmt_cl_3_r+
```

```
Bmmt_cl_26_f+
Bmmt_cl_5_f-
Bmmt_cl_26_r-
```

This is contig 2 in my assembly, and it starts with the read `Bmmt_cl_4_r` in the forward orientation (`+`). The next read is `Bmmt_cl_19_r` in the reverse orientation (`-`). Note that `_f` and `_r` are forward and reverse with respect to the original clone insert (see Figure 2) and the `+` and `-` are how the read was oriented (or aligned to) in the contig.

The read `Bmmt_cl_20_f` is in the forward orientation and is contained within `Bmmt_cl_19_r` in the reverse orientation. The read `Bmmt_cl_3_r` is in the forward orientation, followed by the read `Bmmt_cl_26_f` is in the reverse orientation. And so on…

Don't worry if that does not make complete sense. But what you should be able to deduce here, is that if we have paired reads from the same clone (with a `_f` and an `_r`) - if the `_f` read is oriented on the `-` then the `_r` will be oriented on the `+` of the contig. And vice versa. If they appear on the same orientation in the contig, something has gone wrong!

Questions

7. Where there are pairs of reads from the same clone, do they always appear in the contig in the opposite orientation?
8. How many containments are there across all the contigs?

The "Assembly details" file also shows the alignments between the reads and the contigs. The consensus sequence is the overall agreed sequence from the assembly, and you can see where the reads align to the consensus and overlap between each other to produce the consensus.

6. Final questions

These final question draw together everything you have learned in this practical. You will need to check back on the output of the CAP3 program as you answer some of these questions.

Questions

What the program did:

9. How many sequence comparisons did the assembler carry out overall?
   - The text "Number of segment pairs" means the number of matches of any quality between sequences that were found, while "number of pairwise comparisons" indicates the number of final comparisons that were used in assembly.
10. Why do you think there is a discrepancy between the "number of segment pairs" and the "number of pairwise comparisons"?
11. Using CAP3, why would it be very difficult to assemble millions of reads in a genome?

Analysing the assembly:

12. Why did you get more than one contig in the assembly? I.e. why was it not perfectly assembled?

13. How could we get a better assembly?

Bibliography

[1] X. Huang, “A contig assembly program based on sensitive detection of fragment overlaps,” *Genomics*, vol. 14, no. 1, pp. 18–25, 1992.