Comparative Bacterial Genomics
Genome Databases and File Formats

Teacher: Prof. David W. Ussery
Assistant teacher: Asli Ismihan Ozen
Adapted from exercises of Nepal workshop by Tammi Vesth

June 10, 2014

Contents

Chapter 1

Genome Databases and File Formats

Biological data has been growing with a tremendous rate especially after the advances in the molecular biology techniques in 1940's. The studies for this vast information to be accessed through libraries called biological databases which hold records for the experimental data, classification schemes, literature and some may provide computational analysis tools. There are various databases holding various types of biological information public some of which will be introduced in this section (`http://en.wikipedia.org/wiki/List_of_biological_databases`). As a part of this course, you will be dealing with bacterial genomes, which are a type of biological data.

1.1 The National Center for Biotechnology Information (NCBI)

NCBI [1] is one of the institutions that maintains one of the largest databases holding records of molecular biological data, computational analysis tools and bibliographic information. As a part of NCBI, **GenBank** (`http://www.ncbi.nlm.nih.gov/genbank/` holds the nucleotide sequence data from expression sequence tag (EST), genome survey sequences, other high throughput sequences such as whole genome sequences and genome annotations of thousands of organisms, both prokaryotic and eukaryotic are also available [1].
Although there are several institutions that publish genome sequences, NCBI remains a ma-

[1] `http://www.ncbi.nlm.nih.gov/`

jor source of genomic sequence data. The website `http://www.ncbi.nlm.nih.gov/genome/browse/` lists thousands of Prokaryotic genome sequence projects. Each genome replicon is represented by a unique ID accession number, which is the same across major databases in the US (GenBank) Europe (EMBL) and the DNA Database of Japan (DDBJ). These three databases share the same accession numbers, and is called INSDC, International Nucleotide Sequence Database Collaboration.These numbers can be used to download individual genome sequences for a given organism. The NCBI page has information about on-going projects (unfinished genomes) and finished projects (complete genomes).

Another database within NCBI that holds genome sequences is called **RefSeq**, which is "a comprehensive, integrated, non-redundant, well-annotated set of reference sequences including genomic, transcript, and protein." The Accession numbers for Refseq are slightly different than Genbank Accession numbers. For more information, go to `http://www.ncbi.nlm.nih.gov/Sequin/acc.html`.

NOTE:

The sequences handed out during this course were downloaded from this website using Refseq IDs. There are also Genbank IDs for these files. During the exercises, we will be using Genbank IDs to download genomes.

1.2 File-formats

There are certain type of files that you will be introduced and using during this course. Although these are mostly plain text files, they have certain formats and they are always expected to be found that way. Among these are the GenBank and FASTA format (examples shown below).

1.2.1 Genbank format

The GenBank sequence format is a rich format for storing sequences and associated annotations. It is called a Genbank file because this is the format that the NCBI database requires researchers to upload their data to their databases.

Listing 1.1: GenBank file format example

```
LOCUS       CAA89576                109 aa            linear   PLN 11-AUG-1997
DEFINITION  CYC1 [Saccharomyces cerevisiae].
ACCESSION   CAA89576
VERSION     CAA89576.1  GI:1015707
DBSOURCE    embl locus SCYJR048W, accession Z49548.1
KEYWORDS    .
SOURCE      Saccharomyces cerevisiae (baker's yeast)
  ORGANISM  Saccharomyces cerevisiae
            Eukaryota; Fungi; Ascomycota; Saccharomycotina; Saccharomycetes;
            Saccharomycetales; Saccharomycetaceae; Saccharomyces.
REFERENCE   1  (residues 1 to 109)
  AUTHORS   Huang,M.E., Chuat,J.C. and Galibert,F.
  JOURNAL   Unpublished
REFERENCE   2  (residues 1 to 109)
  AUTHORS   MIPS.
  TITLE     Direct Submission
  JOURNAL   Submitted (25-SEP-1995) Data collected by MIPS on behalf of the
            European yeast chromosome X sequencing project. MIPS at the
            Max-Planck-Institut fuer Biochemie, Am Klopferspitz 18a D-82152
            Martinsried, FRG; E-mail: Mewes@mips.embnet.org
FEATURES             Location/Qualifiers
     source          1..109
                     /organism="Saccharomyces cerevisiae"
                     /db_xref="taxon:4932"
                     /chromosome="X"
     Protein         1..109
                     /name="CYC1"
     CDS             1..109
                     /gene="CYC1"
                     /coded_by="Z49548.1:954..1283"
                     /note="ORF YJR048w"
                     /db_xref="GOA:P00044"
                     /db_xref="SGD:S0003809"
                     /db_xref="UniProtKB/Swiss-Prot:P00044"
ORIGIN
        1 mtefkagsak kgatlfktrc lqchtvekgg phkvgpnlhg ifgrhsgqae gysytdanik
       61 knvlwdennm seyltnpkky ipgtkmafgg lkkekdrndl itylkkace
//
```

1.2.2 FASTA format

In bioinformatics, FASTA format is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. The format also allows for sequence names and comments to precede the sequences. The format originates from the FASTA software package, but has now become a standard in the field of bioinformatics. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. The word following the ">" symbol is the identifier of the sequence, and the rest of the line is the description (both are optional). There should be no space between the ">" and the first letter of the identifier. It is recommended that all lines of text be shorter than 80 characters. The sequence ends

if another line starting with a ">" appears; this indicates the start of another sequence. A simple example of one sequence in FASTA format:

Listing 1.2: FASTA file format example

```
>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGGVLALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFLPIAGX
IENY
```

1.3 Working with GenBank and FASTA files

For this course, each group has been assigned a project with a list of genomes and a focus question. Take a look at the list of genomes that your group has chosen. It is expected that each group find at least one additional genome sequences to be added to the initial genome set. Note the GenBank (INSDC) accession number and download the genome. The additional genome will be analyzed along with the remaining sequences. Copy the folder fitting the research question from the teacher (USB stick or other memory device). Use the file-manager to drop and drag the folder on to the CMG-biotools system. Use the basic UNIX commands to move around the folders and look at the content (`cd`, `ls -l`, `head`, `tail`, `cat`). Note have specific file extensions (`*.gbk`, `*.fsa` and so on) are used to indicate the content of the files. These are not strict rules, but they help you keep track of the data.

1.3.1 Quick Look at the files

- What type of files have you been supplied? Can you figure out their formats? (Use `head`, `tail` and `gedit`)

1.3.2 Download genomes from GenBank

A program has been written which accesses the NCBI webpage, downloads the individual GenBank files from the INSDC numbers. The program is called `gbk_get` and uses a GPID or an NCBI accession number as an argument. The `gbk_get` script uses the Entrez E-utils

programmatic interface made available by the NCBI to fetch sequence data. The output from the program is a GenBank file equivalent to the one that is found on the webpage. Here we will use the program option `-s` which reads the input as a INSDC number. The syntax of the program is shown below.
Note the Unix usage of the ">" sign, which is a redirection of the output into a file. If this is not included, the program will write the output, which is the GenBank file, to the screen.

Listing 1.3: GenBank - download file from NCBI

```
# Syntax:
$ gbk_get -a <INSDC number> > file.gbk
# Example:
$ gbk_get -a AE000511 > AE000511.gbk
```

1.3.3 Obtain data from GenBank files

At this point you should have a pre-downloaded folder with GenBank files along with an additional file which you downloaded as described above. You shall now investigate the GenBank file format (file extension: `*.gbk`) Open the file in a text-editor, either from the file-manager (click `Home` on the desktop) by clicking the file or in the `Terminal` application calling the program `gedit`.

Listing 1.4: Open and investigate GenBank file

```
# Syntax:
$ gedit <INSDC>.gbk
# Example:
$ gedit AE000511.gbk
```

In the beginning of the file is the metadata, which contains names, publications, habitat and similar information. The next part is the annotations, `genes` and `CDS` (`CoDing Sequences`). In this section the genes are described by their location, direction, note, and `translation`.

1.3.3.1 Exercises

1. Download one genome from GenBank.
2. What information is found in the line marked `LOCUS`?
3. How many lines are marked `LOCUS` and what does this number show (use `grep -c`)?
4. Explain the content in the re-occurring fields marked `source`, `gene` and `CDS`.

1.3.4 Extract organism name

To make it easier to recognize files they will now be re-named so they are called an organism name instead of a `INSDC` number. This procedure has already been performed on your large dataset, and you should only run this program on your newly downloaded GenBank file! From this point on, `INSDC` will be replaced with `name` and will refer to the organism name the file is given.

Listing 1.5: GenBank extract - organism name

```
# Syntax:
$ gbk_ExtractName <INSDC>.gbk
# Exampel:
$ gbk_ExtractName AE000511.gbk
# The following file is produced:
Helicobacter_pylori_26695_ID_AE000511.gbk
```

Note that the files are not moved, but rather, they are copied into a new file. Delete the numbered files using the command `rm`. The new files will from here on be referred to as `<name>.gbk` in the command syntax.

1.3.4.1 Exercises

1. Extract name from one GenBank file
2. What could the `gbk_ExtractName` program be doing? How does the program create the name?
3. To look at the code, open it in the text-editor (`gedit /usr/biotools/gbk_ExtractName`).

1.3.5 Extract DNA

Further analysis of the genomes sequences requires extracting the DNA from the GenBank file. This procedure has already been performed on your large dataset, and you should only run this program on your newly downloaded GenBank file! This can be done using a program called `saco_convert` [3], which, as the name implies, converts one file format into another. Below is shown the syntax of the program (note that the length of the name makes the line wrap around, but the command is still one line):

Listing 1.6: GenBank extract - DNA

```
# Syntax:
$ saco_convert -I genbank -O fasta <name>.gbk > <name>.gbk.dna
# Exampel:
$ saco_convert -I genbank -O fasta Helicobacter_pylori_26695_ID_AE000511.gbk >
    Helicobacter_pylori_26695_ID_AE000511.gbk.dna
```

The file extension is now `*.gbk.dna`, illustrating the the file contains DNA extracted from a GenBank file. You shall now try to run this procedure on all the GenBank files in the `GBK` folder. This can be done using a so called `for-loop`, which runs a specific command a number of times in stead of one. Below is shown to versions, first a `Trial` to illustrate how the loop works and then a `Example` of how the loop looks for `saco_convert`. First try the `Trial`, type `for x in *gbk` on the command-line, this will cause a `>` sign to appear. Type the next commands, and finish each line with `Enter`. The word `done` tells the `Terminal` that the loop is now over and executes the commands typed within the loop. Read the below explanations carefully before you type and do the trials a couple of times before you go on to the `saco_convert` loop.

Listing 1.7: `Introduction - for-loop`

```
# Trial:
$ for x in *gbk
> do
> echo $x
> done
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk
Neisseria_gonorrhoeae_NCCP11945_ID_CP001050.gbk
Neisseria_gonorrhoeae_TCDC-NG08107_ID_CP002440.gbk
Neisseria_meningitidis_053442_ID_CP000381.gbk
Neisseria_meningitidis_8013_ID_FM999788.gbk
Neisseria_meningitidis_alpha14_ID_AM889136.gbk
Neisseria_meningitidis_alpha710_ID_CP001561.gbk
Neisseria_meningitidis_FAM18_ID_AM421808.gbk
......
```

The command `echo` simply writes something to the screen, try typing `echo hello world`. As is seen above, the program `for` looks at the files matching some match criteria, in this case, files with the suffix `*.gbk`. Each of these files, the name of the file, is used as a value of `x` in the loop. More elaborate patterns can also be used, like `Neisseria_gonorrhoeae*gbk` if you only want to run the loop on a subset of files. Below is shown the loop for extracting DNA from several GenBank files. Note how the extension `*.dna` is added to the value of `x`, which is the GenBank filename. The resulting filename will have the extension `*.gbk.dna`.

Listing 1.8: GenBank extract - DNA in for-loop

```
# Example:
$ for x in *gbk
> do
> echo $x
> saco_convert -I genbank -O fasta $x > $x.dna
> done

# Alternative writing
$ for x in *gbk; do saco_convert -I genbank -O fasta $x > $x.dna; done
```

None of the files generated above should be empty as this would indicate that no sequences are found in the GenBank file or that the program is not managing to find the DNA. Verify that your files are not empty using `ls -lh`. Look at the file and make sure that it contains DNA in FASTA format (use `head`, `tail` or `gedit`). Number of ">" FASTA headers should be equal to the number of replicons (chromosomes or plasmids).

1.3.5.1 Exercises

1. Extract DNA from all GenBank files
2. Count the number of `LOCUS` tags in each GenBank file (use `grep -c`)
3. Count number of FASTA headers in the `*.dna` file (use `grep -c`)

1.3.6 Extract genes and proteins + gene-finding

From the initial investigations of the GenBank files, you have probably seen that some files contain genes and proteins. These data are the result of 'gene-finding', where the DNA sequence has been analyzed and searched for possible genes. For some genes there might be some additional experimental verification, but many (most) genes are just predictions. In the following we will extract the nucleotide sequences as well as the corresponding amino acid sequences. The programs is using `BioPerl` modules[?] for the handling of GenBank formats. The code is called `gbk_ExtractGeneProt` and the output is two FASTA formatted text files, one for the genes and one for the proteins.

Listing 1.9: GenBank extract - genes and proteins

```
# Syntax:
$ gbk_ExtractGeneProt <name>.gbk
# Example:
$ gbk_ExtractGeneProt Helicobacter_pylori_26695_ID_AE000511.gbk
```

For the genomes/replicons with no published annotation you will run local gene-finding. Gene finding is performed using the program Prodigal[2]. The program is wrapped into a formatting program called `prodigalrunner`. The program reformats the raw output of Prodigal to FASTA formatted open reading frames, DNA and amino acids, along with a draft of a GenBank file and a raw general feature formatted file, a `*.gff` file. The Prodigal program allows for different parameter modifications, including training (`prodigalrunner -t <organism>`) of the gene finder using given data. This feature increases the computation time of the algorithm, but for less known organisms this feature might improve gene finding. It should be noted that the default behavior when encountering N's is not changed - the program treats runs of N's as masked sequence and does not build genes across them. The CMG-Biotools system also comes with the native Prodigal program, which can be used as published[2]. Identify the files that are empty and for those DNA files, run the following command:

Listing 1.10: GenBank extract - gene-finding

```
# Syntax:
$ prodigalrunner <DNA FASTA file>
# Example:
$ prodigalrunner Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gbk.dna
# The following files are produced:
Neisseria_gonorrhoeae_FA_1090_ID_AE004969.gff    #   raw prodigal output, you will not use
    this file
Neisseria_gonorrhoeae_FA_1090_ID_AE004969_prodigal.gbk  # "fake" GenBank file, you will not
    use this file
Neisseria_gonorrhoeae_FA_1090_ID_AE004969_prodigal.orf.fna  # gene file in FASTA format
Neisseria_gonorrhoeae_FA_1090_ID_AE004969_prodigal.orf.fsa # protein file   in FASTA format
# Remove un-needed files:
$ rm *gff
$ rm *prodigal.gbk
```

Move the gene FASTA (`*.fna`) and protein FASTA (`*.fsa`) to the appropriate folders. Now you can remove the GenBank gene and proteins files that were empty. Move into the folders where the gene FASTA (`*.fna`) and protein FASTA (`*.fsa`) files are stored and run the following commands. Verify the files that will be deleted by first running the `Display` command.

Listing 1.11: Remove empty files

```
# Display empty files:
$ find . -type f -empty
# Remove empty files:
```

```
$ find . -type f -empty -exec rm {} \;
```

For each of the non-empty gene and protein FASTA files, count the number of sequences and store the results in a file. These numbers describe the size of the proteome for each chromosome/plasmid and genome. The number of sequences in the protein/gene files should be the same, as all genes should be translated.

Listing 1.12: Count number of proteins/genes - loop

```
for x in *fna
> do
> echo $x
> echo $x >> proteinCounts.txt
> grep -c ">" $x >> proteinCounts.txt
> done
# If you need to run this loop again, delete the proteinCounts.txt file first
```

1.3.6.1 Exercises

1. Extract genes and proteins from all GenBank files
2. Some of the gene and protein files might be empty (use `ls -lh` to verify), can you think of a reason why?
3. Remove empty files (use `find`)
4. Make sure that all files are put in a folder corresponding to file type, for example, make a folder called `FSA` and move all files with the extension `*.fsa` to that folder.

1.4 Summary

In this chapter, you have learned how to :

- Download a genome sequence from NCBI (one genome)
- Extract the organism name from a GenBank file
- Extract DNA, genes and proteins from a GenBank file
- Organize data
- Annotate genomes, gene-finding
- Remove empty files

Bibliography

[1] Benson, D. a., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. GenBank. Nucleic acids research 39, Database issue (Jan. 2011), D32–7.

[2] Hyatt, D., Chen, G.-L., Locascio, P. F., Land, M. L., Larimer, F. W., and Hauser, L. J. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics 11 (Jan. 2010), 119.

[3] Jensen, L. J., Friis, C., and Ussery, D. W. Three views of microbial genomes. Research in microbiology 150, 9-10 (1999), 773–7.