Fundamentals of computational biology

Camilo García-Botero

2022-04-11

Table of contents

Preface

Here we present a course centered book of the Fundamentals of Computational Biology. We will cover several topics, from using the unix tools, the importance of package manager systems (such as homebrew and conda), sequencing technologies, sequence alignments, molecular phylogenetics, genome assembly and annotation, and variant calling analysis.

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

2 Welcome to the command line

In this chapter we will explore the fundamentals of the command line. That is the concepts of Unix based systems the command line (CLI) and how we can use it to access information programmatically.

3 sequence-analysis

In this chapter we will use several tools to download a genome from the command line. We will identify some features

3.1 Downloading a genome

```
ncbi-genome-download
```

3.2 Downloading from NCBI

The first step in this journey is to download a bunch of sequences programatically. To do so, we will use the program ncbi-genome-download.

You could inspect all the options it provides, now we will set our command as the following:

```
ngd --genera "Bacillus subtilis"\
    -s refseq\
    -l complete\
    -o Data\
    --flat-output\
    --format features\
    -n bacteria\
    | head -n 1
```

```
Considering the following 193 assemblies for download:
```

3.3 Listing files

```
ls Data | head -n 10
```

3.4 Decompressing using `gzip`

```
gzip -d *
```

. . .

3.4.1 Some files in our data dir

```
ls Data | head
```

3.5 Importing the files into R

```
library(tidyverse)
library(fs)

all_features <- dir_ls("Data/") %>%
  map_df(read_tsv)

all_features %>%
  head()
```

. . .

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.6     v dplyr   1.0.8
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(fs)

all_features <- dir_ls("Data/") %>%
  map_df(read_tsv)

all_features %>%
  head()
```

```
# A tibble: 0 x 0
```

3.6 Data processing

```
all_features_grouped <- all_features %>% #create a new dataset that will group by features
  rename(feature = `# feature`) %>% # get read of the weird name of the column
  select(assembly, feature) %>% # Select these two columns
  group_by(assembly, feature) %>% # Group by these two columns to perform operations
  count() %>% # count the numbers of rows based on the applied group
  pivot_wider(names_from = feature, values_from = n) %>% # generate a wide dataset sending
  arrange(desc(CDS)) # Arrange descending by the number of CDSs

all_features_grouped %>%
  head()
```

4 Summary

5 Challenges demos

5.1 Genome searching (Ch. 01)

In this chapter we will use several tools to download a genome from the command line. We will identify some features

5.1.1 Downloading a genome

```
ncbi-genome-download
```

5.1.2 Downloading from NCBI

The first step in this journey is to download a bunch of sequences programatically. To do so, we will use the program ncbi-genome-download.

You could inspect all the options it provides, now we will set our command as the following:

```
ngd --genera "Bacillus subtilis"\
    -s refseq\
    -l complete\
    -o Data\
    --flat-output\
    --format features\
    -n bacteria\
    | head -n 10
```

```
Considering the following 193 assemblies for download:
GCF_000772125.1 Bacillus subtilis   ATCC 13952
GCF_000772165.1 Bacillus subtilis   ATCC 19217
GCF_000772205.1 Bacillus subtilis   Bs-916
GCF_000782835.1 Bacillus subtilis   SG6
GCF_000789295.1 Bacillus subtilis   PS832
```

```
GCF_000952895.1 Bacillus subtilis   BS34A
GCF_000953615.1 Bacillus subtilis   BS49
GCF_001015095.1 Bacillus subtilis   UD1022
GCF_001037985.1 Bacillus subtilis   TO-A JPC
```

5.1.3 Listing files

```
ls Data | head -n 10
```

5.1.4 Decompressing using gzip

```
gzip -d *
```

. . .

5.1.4.1 Some files in our data dir

```
ls Data | head
```

5.1.5 Importing the files into R

```
library(tidyverse)
library(fs)

all_features <- dir_ls("Data/") %>%
  map_df(read_tsv)

all_features %>%
  head()
```

. . .

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.6     v dplyr   1.0.8
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
library(fs)

all_features <- dir_ls("Data/") %>%
  map_df(read_tsv)

all_features %>%
  head()
```

```
# A tibble: 0 x 0
```

5.1.6 Data processing

```r
all_features_grouped <- all_features %>% #create a new dataset that will group by features
  rename(feature = `# feature`) %>% # get read of the weird name of the column
  select(assembly, feature) %>% # Select these two columns
  group_by(assembly, feature) %>% # Group by these two columns to perform operations
  count() %>% # count the numbers of rows based on the applied group
  pivot_wider(names_from = feature, values_from = n) %>% # generate a wide dataset sending
  arrange(desc(CDS)) # Arrange descending by the number of CDSs

all_features_grouped %>%
  head()
```

References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.