Fundamentals of computational biology

Camilo García-Botero

2022-04-11

Table of contents

Preface

We started this book with the aim of compiling the lectures of the course Fundamentals of Computational Biology offered at Universidad EAFIT for undergrad students in Biology. The course has been taught from different perspectives from its creation, yet the last iteration was divided into three modules. i) introduction to Unix (4 lectures) ii) introduction to sequence analysis and genomics (7 lectures) and iii) principles of structural biology (4 lectures).

Lectures are focused on a theoretical-practical approach were basic concepts from biology, bioinformatics and computer science and interleave with the practice to solve challenges.

1 Introduction

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.

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 Introduction to sequence analysis

4 Summary

5 Challenges demos

5.1 Genome searching

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

Downloading a genome

```
ncbi-genome-download
```

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
```

Listing files

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

Decompressing using `gzip`

```
gzip -d *
```

. . .

Some files in our data dir

```
ls Data | head
```

Importing the files into R

```
library(tidyverse)
library(fs)

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

all_features %>%
  head()
```

. . .

```
library(tidyverse)
library(fs)

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

all_features %>%
  head()
```

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

Data processing

```
all_features_grouped <- all_features %>%
  rename(feature = `# feature`) %>%
  select(assembly, feature) %>%
  group_by(assembly, feature) %>%  operations
  count() %>%
  pivot_wider(names_from = feature, values_from = n) %>%
  arrange(desc(CDS))

all_features_grouped %>%
  head()
```

create a new dataset that will group by features per accession. get read of the weird name of the column. Select these two columns. Group by these two columns to perform. count the numbers of rows based on the applied group. generate a wide dataset sending row names as columns. Arrange descending by the number of CDSs.

References

Challenges demos

Genome searching

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

Downloading a genome

```
ncbi-genome-download
```

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:

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

Listing files

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

Decompressing using `gzip`

```
gzip -d *
```

. . .

Some files in our data dir

```
ls Data | head
```

Importing the files into R

```
#| label: file-reading
#| echo: true
#| eval: false
#| message: false
#| warclass-warning: false
#| warattr-warning: false
library(tidyverse)
library(fs)

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

all_features %>%
  head()
```

. . .

```
#| label: file-reading-exe
#| ref.label: file-reading
#| eval: true
#| cache: true
#| warclass-warning: false
#| message: false
#| warattr-warning: false
#| out.width : 50%
```

Data processing

```
#| label: data-processing
#| echo: true
#| eval: false
#| message: false
all_features_grouped <- all_features %>%
  rename(feature = `# feature`) %>%
  select(assembly, feature) %>%
  group_by(assembly, feature) %>%  operations
  count() %>%
  pivot_wider(names_from = feature, values_from = n) %>%
  arrange(desc(CDS))

all_features_grouped %>%
  head()
```

create a new dataset that will group by features per accession. get read of the weird name of the column. Select these two columns. Group by these two columns to perform. count the numbers of rows based on the applied group. generate a wide dataset sending row names as columns. Arrange descending by the number of CDSs.