BM425 Workshop 1: SARS-CoV-2 Genome Analysis

Leighton Pritchard and Morgan Feeney

2024-09-27

Table of contents

Preface to 2024-25 BM425 Workshop 1

Welcome to the BM425 (Advanced Microbiology) workshop on genome analysis of SARS-CoV-2, for 2024-25.

This year is the first presentation of the workshop material in this format, and we would be very grateful to hear feedback by email or through the GitHub repository Issues page.

Overview

This workshop asks you to work through some bioinformatics exercises using the online service `Galaxy`, with data that we provide, to:

- assemble, annotate, and visualise the first SARS-CoV-2 genome isolated in Wuhan, in January 2020
- compare the genome of SARS-CoV-2 to that of an earlier coronavirus (SARS 2003)
- compare the genome of the first SARS-CoV-2 genome to that of an variant from later in the pandemic
- make a biological interpretation of your analysis, particularly of the spike (S) protein

> ⚠ Important Note
>
> There is new material in this workshop that is not covered in lectures. **This material is examinable.**
> Please take care to read the text in the expandable callout boxes, as well as that for the workshop activities, to be sure you have understood the topic and obtain full value from the exercises.

Learning Objectives

By the end of this workshop, students will be able to:

- understand and use the Galaxy scientific workflow system
- recognise and use common sequencing data formats
- use common bioinformatics tools to:

- assemble sequenced reads into a genome sequence
- annotate features on an assembled genome sequence
- map sequencing reads onto an assembled genome
- carry out comparative genome analysis
- visualise and interpret the results of genome annotation and analysis

Assessment

> **!** Important
>
> There is a formative assessment on the workshop MyPlace page that you should complete at the end of the workshop, to demonstrate you've earned your genomics wings (link below).
>
> - MyPlace formative quiz

1 Introduction

The Introduction page is intended as a short introduction to the book.

Like most Quarto books, this is a book created from markdown and executable code.

This kind of book is an example of literate programming - the intertwining of nicely-formatted text and images, and executable code. For example, the `R` code cell below executes and produces output when the book is compiled:

```
1 + 1
```

```
[1] 2
```

But the `R` code cell below does not:

```
summary(cars)
```

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

Part I

Early Section

This `.qmd` file introduces a `Part` of the Quarto book. We use the `{#sec-REFERENCE}` option to make it cross-referenceable elsewhere in the text, and the `{.unnumbered}` option to avoid giving it a section number.

2 Early Section Topic

This `.qmd` file represents some topic-related text. We use the `{#sec-REFERENCE}` option to make it cross-referenceable elsewhere in the text.

Part II

Late Section

This `.qmd` file introduces a `Part` of the Quarto book. We use the `{#sec-REFERENCE}` option to make it cross-referenceable elsewhere in the text, and the `{.unnumbered}` option to avoid giving it a section number.

3 R Playground

```
#| context: setup

# Download reporter data
download.file('https://raw.githubusercontent.com/sipbs-compbiol/BM214-Workshop-3/main/assets/

library(ggplot2)
library(palmerpenguins)
library(tidyverse)
```

3.1 Introduction

This page provides a WebR cell for you to use as a playground to experiment with some example datasets. You can use this page to explore data management and visualisation in R.

3.2 Playground

```
# Use this WebR cell to experiment with some practice biological datasets
```

3.3 Things you can do

This WebR instance has three packages installed:

- ggplot2
- GGally
- tidyverse
- palmerpenguins

Open the callout boxes below to see some examples you can try in the code cell above.

Play with data from a GitHub repository

One of our BM214 workshops involves a `WebR`-supported interactive exercise involving simulated reporter curves. We preload this data in the `setup` cell (see source code), and you can interact with it below with the code:

```
data <- read.csv("reporter_curves.csv")
glimpse(data)
```

Investigate Palmer's Penguins

The `penguins` dataset contains data about three different species of penguins. Take a look at the format of the dataset:

```
glimpse(penguins)
```

You'll see there are eight variables, including `species`, `weight`, `sex`, etc. - some of these variables are *categorical* (i.e. a category, like `species`), and others are *continuous* (i.e. numerical). You can see a visual overview of how the data is related using the `plot()` function:

```
plot(penguins)
```

We can visualise the number of penguins of each species in a bar chart:

```
fig <- ggplot(penguins, aes(species, fill=species)) +
         geom_bar()
fig
```

And break this down in a facet plot, by sex:

```
fig <- ggplot(penguins, aes(species, fill=species)) +
         geom_bar() +
         facet_wrap(~sex)
fig
```

We can make a box and whisker plot of penguin body mass by species:

```
fig <- ggplot(penguins, aes(x=species, y=body_mass_g, fill=species)) +
         geom_boxplot()
fig
```

And plot the body mass for each sex side-by-side

```
fig <- ggplot(penguins, aes(x=species, y=body_mass_g, fill=sex)) +
         geom_boxplot()
fig
```

We can investigate correlations, such as between body mass and flipper length:

```
fig <- ggplot(penguins, aes(x=body_mass_g, y=flipper_length_mm)) +
         geom_point()
fig
```

We can colour datapoints by species:

```
fig <- ggplot(penguins, aes(x=body_mass_g, y=flipper_length_mm, colour=species)) +
         geom_point()
fig
```

And fit a linear regression to each species separately:

```
fig <- ggplot(penguins, aes(x=body_mass_g, y=flipper_length_mm, colour=species)) +
         geom_point() +
         geom_smooth(method="lm")
fig
```

Note

`R` comes with a number of example datasets you can practice with, including:

- `mtcars`: fuel consumption and other statistic for 32 automobiles
- `Titanic`: information on the fate of passengers on the fatal maiden voyage of the ocean liner *Titanic*

You can see a full list by running the command

```
library(help = "datasets")
```

References

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