

# BIOL365 Prac 1 — Getting ready!

Dr James B Dorey and Dr Damien Esquerré, University of Wollongong

Version 2024-07-24; practical 2024-07-26

## Abstract

This practical class should not be particularly onerous. Today is mostly focussed on giving you a very brief overview of what is to come in this practical series and to get you thinking about what you might like to do for your Grant Proposal assignment.

R is a very useful programming language that's open access and widely used in science. We don't expect you to become a master by the end of this topic or to understand everything that you're doing right now. But, by helping you get started and giving you robust example code we hope to enable you to succeed using R in the future, should you chose to pick it up more consistently.

## Contents

<b>1</b>	<b>Introduction to the practical schedule</b>	<b>1</b>
<b>2</b>	<b>Working together</b>	<b>2</b>
<b>3</b>	<b>Start thinking about a macroevolutionary question</b>	<b>2</b>
<b>4</b>	<b>Getting ready with <i>R</i> and <i>RStudio</i></b>	<b>4</b>
4.1	Install <i>R</i> . . . . .	4
4.2	Install <i>RStudio</i> . . . . .	4
4.3	Running <i>R</i> in <i>RStudio</i> . . . . .	4
4.4	Script preparation . . . . .	7
4.4.1	Working directory . . . . .	7
4.4.2	Install packages . . . . .	7
4.4.3	Load packages . . . . .	7
<b>5</b>	<b>What's coming next week?</b>	<b>8</b>
<b>6</b>	<b>Packages used today</b>	<b>8</b>

## 1 Introduction to the practical schedule

Practicals will be run as the below practical schedule from the course handbook. Following this practical, we will have:

- Four weeks of the *Phylogenetics Module*. The first three weeks will be focused on teaching you phylogenetics in the *R* programming language and the fourth will be aimed at providing support for your **Grant Proposal** assignment.
- We will have four weeks of the *Conservation Behaviour Module*. This module will focus on teaching you about human-wildlife conflicts and supporting you in your **Elevator Pitch** assignment.
- In weeks **7** and **13**, instead of a prac, we will have in-class quizzes, each worth 20% of your final grade and base on the lectures delivered previously.
- In weeks **11** and **12**, you will deliver your seminars to the class, instead of undertaking a standard practical.

<b>Week</b>	<b>Week Commencing</b>	<b>Topics Covered</b>	<b>Readings and Activities</b>
1	22 Jul 2024	Introduction to the practical schedule of learning	
2	29 Jul 2024	Phylogenetic module	
3	05 Aug 2024	Phylogenetic module	
4	12 Aug 2024	Phylogenetic module	
5	19 Aug 2024	Phylogenetic module	
6	26 Aug 2024	Conservation Behaviour module	
7	02 Sep 2024	Quiz 1 (covering lectures 1–13). Conservation Behaviour module	
8	09 Sep 2024	Conservation Behaviour module	
9	16 Sep 2024	Conservation Behaviour module	
10	23 Sep 2024	Conservation Behaviour module — Elevator pitch delivered to the class	
	30 Sep 2024	<b>Mid-Session Recess</b>	
11	07 Oct 2024	Seminar delivery (group 1)	
12	14 Oct 2024	Seminar delivery (group 2)	
13	21 Oct 2024	Quiz 2 (covering lectures 14–26)	
	28 Oct 2024	<b>Study Recess</b>	
	02 Nov 2024	<b>Examinations</b>	
	09 Nov 2024	<b>Examinations</b>	

## 2 Working together

You won't be forming groups *per se*, but it is possible to work together on benches to provide your neighbours help with code and the like. I do encourage this as your neighbour may have already overcome errors that you have come across and both the student helping and the one receiving help should benefit! Of course, we will be here to help you as well; your demonstrators should have the answers and, if not, your lecturers will.

## 3 Start thinking about a macroevolutionary question

“Macroevolutionary questions” may sound like an arcane concept. However, they are really just questions that we ask with the goal of uncovering broad scale (almost always across multiple species) evolutionary

patterns. Maybe that doesn't help... But, a simple question like "*I wonder how many bird species have purple feathers*" could easily be addressed as a macroevolutionary question! But, that might be too broad of a question...

Because I work on bees, I might have a question about a particular bee subgenus. Say the halictid (from the family Halictidae) bee subgenus *Lasioglossum* (*Homalictus*). I might come up with a hypothesis such as "*More bees in the subgenus Homalictus are metallic in colouration than non-metallic*". That's a simple enough question and I suspect that it's true... but these bees are found all over the place (Australia to Southern China and India)! Maybe I could narrow it down to a place where there are fewer species and lots of genetic data available. I happen to work on the Fijian *Homalictus*, so I could hypothesise that "*More FIJIAN bees in the subgenus Homalictus have metallic abdomens than non-metallic abdomens*" — a question about the prevalence of a state. The data to answer this question could be derived from [here](#). You could also hypothesise that "*Homalictus* bees are more likely to move from having metallic abdomens to non-metallic abdomens than the other way around" — a question about how frequently states evolve and their direction.



Figure 1: Photos of various Fijian *Lasioglossum* (*Homalictus*) (Hymenoptera: Halictidae) bees by James Dorey

You get the idea. Now we are talking in terms of macroevolution.

You might have a question that you want to answer, or you might have a group that you are interested in. But, you need to choose a taxon that is big enough (say  $>7$  species) and limit yourself to a subset of those species ( $<30$ ) for which both genetic data and trait data either exist or can easily be harvested. You could start by searching in **Google Scholar**, or **Scopus**, for something like "*macroevolution bears*". Then you can see what genetic data are available and what morphological data are available. While you can use those same data for your assignment, you **must ask a novel question using those data or a subset of those data**.

Talk among your table or with your desk mate and with your demonstrators. You don't need to have a

project set in stone by any means, but it's good to get started on thinking about this and what you might be able to do!

## 4 Getting ready with *R* and *RStudio*

Your computers should already have *R* and *RStudio* installed. If they do not, or you are using a personal computer, feel free to follow the steps below to prepare yourself for the week 2 practicals, where we will dive in pretty quickly.

If you have a moment of spare time you may consider skimming the below text, especially the “Script preparation” and “What’s coming next week” sections.

### 4.1 Install *R*

If *R* is NOT installed on your computer already, we will go ahead and do that now. Please visit the [CSIRO mirror for CRAN](#) and download the relevant version of *R* for your operating system. Then, follow the instructions to install it on your computer.

### 4.2 Install *RStudio*

*R* is a command-line programming language and *R*, by itself is horribly ugly and I don't like the idea of working in it directly. For this reason, most users will use *R* within a much nicer interface program, called *RStudio*. Download the free version of [RStudio](#). Now, when both programs are installed, you can simply open up *RStudio* and get coding!

We are not going to do much with *R* today, don't worry, but I'd like to get you a little acquainted and at least start installing some packages in *R* so that we are ready for the following weeks.

### 4.3 Running *R* in *RStudio*

Before we get into working in *RStudio*, let's make a folder where we will save all of our practical outputs and related files. Mine is called “*BIOL365\_pracs\_2024*”. Notice how I have used underscores instead of spaces? This is very good practice to not have ANY spaces in your folder paths where you want to do coding stuff. Most of the time it's fine... but, **pro tip**, sometimes, it will cause problems that might be hard to track down!

When you open up *RStudio* for the first time, it will look a little something like the below (Fig. 2). But, without the funky colour scheme.

In the top left-hand side drop down, there is a white page with a green plus sign on it. Click on that and you can then select to add a new “R Script” (Fig. 3). We can then go ahead and save this blank script in a folder for these pracs (Fig. 4). I have called this script “*MyFirstRScript.R*”.

It's also VERY good practice to leave some info about who made this script, why, and how they can get in touch with you! Go ahead and copy the below into your script and then personalise it with your details!

```
# BIOL365 at the University of Wollongong, very basic R stuff
# Getting ready to do stuff with R
# Written by FirstName LastName YEAR-MONTH-DAY University of Wollongong; your@email.here
```

Did you notice all of the hashtags? These are comments and *R* will ignore them (Fig. 5)!

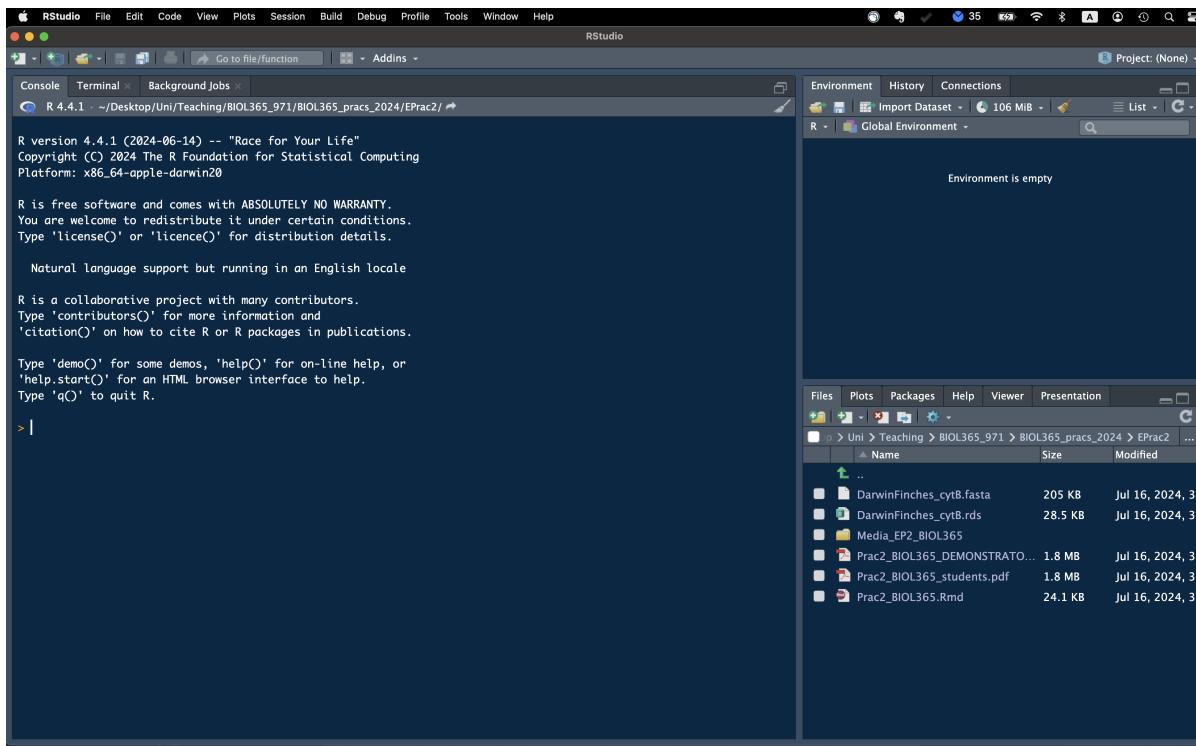


Figure 2: An empty *RStudio* page

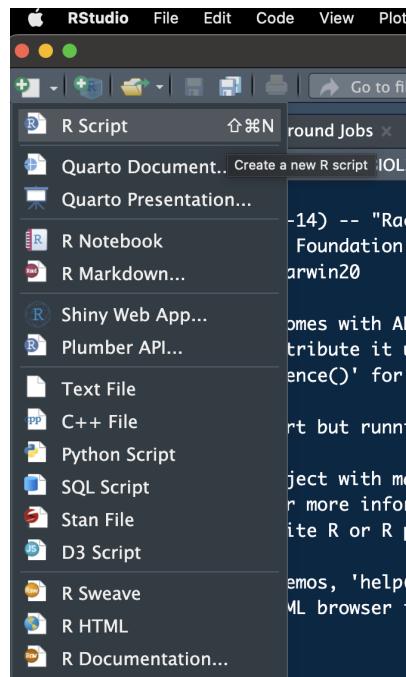


Figure 3: How to add a new *R* Script

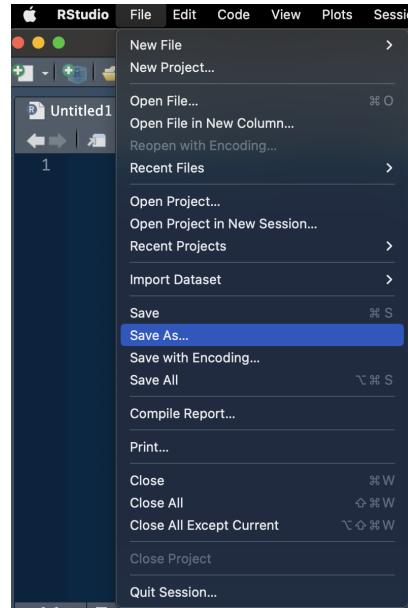


Figure 4: How to save your new *R* Script

```
# BIOL365 at the University of Wollongong, very basic R stuff
# Getting ready to do stuff with R
# Written by FirstName LastName YEAR-MONTH-DAY University of Wollongong; your@email.here
actual code is white and this wont get ignored.
```

Figure 5: My colour scheme shows hashtags in blue, and actual code in white. Very handy.

## 4.4 Script preparation

### 4.4.1 Working directory

Let us begin by telling *R* where our working directory (the folder that you made above) actually is. My “*BIOL365\_pracs\_2024*” folder is found at the end of the path:

“*/Users/jamesdorey/Desktop/Uni/Teaching/BIOL365\_971/BIOL365\_pracs\_2024/Prac1/*” (see again, no spaces!). So, I can run the below:

```
# Set the RootPath to your folder
RootPath <- "/Users/jamesdorey/Desktop/Uni/Teaching/BIOL365_971/BIOL365_pracs_2024/Prac1/BIOL365_pracs_2024"
# You can then set this as the project's working directory.
# This is where R will first look to find
# or save data as a default
setwd(RootPath)
```

Congratulations, you have set your working directory! *R* can still access files outside of this folder, but it will look there by default after you set it (this must be set each time you open *R*).

**A quick pro-tip:** You can run code, once it’s entered into your script by having your mouse click on, or above, the line that you want to run and then pressing the “Run” button on the top right hand side of the script window. That’s a right pain. On mac you can simply use “**command+enter**” and on PC you can use “**control+enter**” to run your code. It’ll make you life easier and quicker.

### 4.4.2 Install packages

Let us also quickly install a few packages that we’ll need to start with next week (you mostly only need to do this once per package and per version of *R*).

```
# This package is for data management and table manipulation
install.packages("dplyr")
# This package lets us use tidy pipes; %>%
install.packages("magrittr")
```

You may have seen that more than one package was installed when you ran the above code. That’s normal, many packages depend on other packages to work.

### 4.4.3 Load packages

The last *R* thing that I will get you to do today is to load the packages into *R* (this also should be done every time you open *R*, for the relevant packages). You need to do this because you may not always want EVERY package that you have ever downloaded to be accessible from in *R*... it can cause issues. In this way, you can be more selective about which ones are active.

```
library(dplyr)
library(magrittr)
```

You can find more info about getting started with *R* and *RStudio*, along with other tutorials at [Our Coding Club](#).

## 5 What's coming next week?

Next week, we will be downloading data from GenBank using a web browser and then using *R*. Once we have those data and have played around with them a little bit, it'll be time for you to explore and maybe start thinking about the data that you want to use for your **Grant Proposal** assignment!

## 6 Packages used today

