Module 1.2: Learn - Coding

1.2 Coding

In this module, we will help you to get started with programming in R to analyze biological data. We will be using a lesson from the Software Carpentries program, which is specifically designed for novice programmers by experts in the field.

We have chosen to program in R programming language in this course for the following reasons:

1. A lot of very useful programs and software packages for biological and genetic analysis and data visualization have been developed in R and featured in publications.
2. R and environments that help you to write R code are freely available with versions that work on common operating systems including Linux, Windows, and Mac.
3. R is one of the languages we use commonly in the Sex Chromosomes Lab so there is a lot of in-house expertise to help troubleshoot.
    - It is also very commonly used in modern science so searching the Internet for coding solutions often yields helpful results.

Setting up to use R

In the tutorial lesson we have chosen, setup instructions say that you need to install R and RStudio. There are a few options for this which you can think about when considering your computing setup.

First, R, RStudio, what is all this stuff? R is a programming language that is released as part of a programming environment. This means that when you install R, you will be able to write code using the R programming language that will allow you to read, manipulate, and visualize data and then use the R engine to actually perform the tasks in the code you have written. R comes with a graphical user interface called R GUI. This interface is helpful in basic ways such as being able to display your code, enter commands, and seeing the plots generated all in a single window, but RStudio is a more sophisticated coding environment with many features that will help you to develop code more efficiently.

The tutorial will give you links to download R and RStudio Desktop to your own computer, but you have other options for this. As a perk of being students in this class, you have been given an account on the ASU supercomputer Sol. ASU hosts an RStudio server which allows you to use R and RStudio as a job on the supercomputer over the Internet.

Here are the pros/cons and recommendations to do this:

1. Using R and RStudio on ASU Sol biocomputing cluster
    - Recommended if you have never programmed in R
    - Pros:
        - Don’t need hard drive space or processing power on your own computer because you would be using space and processors on the supercomputer
        - Help available from ASU Research Computing if you need it
        - Will teach you the basics of how to use the Sol supercomputer
    - Cons:
        - Requires a reliable high-speed internet connection
    - Notes:
        - This option might take an extra few minutes to learn but knowing how to use the supercomputer is a great professional skill in genomics
        - You are currently permitted to use the cluster as part of this course and that access can be extended if you choose to continue this research with Dr.Wilson as your mentor.
2. Downloading R and RStudio your own computer
    - If you have already worked with R and RStudio in the past and have it on your computer already, feel free to continue using that
        - You should be using version 4 of R at minimum, so please check the version and update if you are using R version 3 or less
    - Pros:
        - All the libraries you install will be ready for use after this course with no further need for renewing your access
    - Cons:
        - You will need space and processing power on your computer to download the software and some larger data files we will be working with
        - The R packages you install for use in your code can be quite computationally demanding so how fast they run will be dependent on how fast your computer is

- If your computer crashes, you will lose any work that is not backed up (more options for recovery on ASU supercomputer data storage)
- If you need help, you will need to do a better job describing your specific setup

We encourage you to try the RStudio Server on ASU Sol supercomputer first, and download to your computer if you have problems that are hard to solve even with the instructors’ help.

RStudio Server on Sol supercomputer

Sol is a High-Performance Computing (HPC) cluster, FREE for faculty, staff, and students. This cluster architecture uses hundreds of computer servers, also called nodes, and their collective cores to help users optimize their research. This gives researchers access to high memory computation and storage while freeing the researcher’s local machine. All students must be supported by ASU faculty in order to use the research computing cluster.

Step-by-step guide on accessing RStudio Server on the Sol Cluster

Step 1. Downloading a Virtual Private Network (VPN) in order to remotely access the ASU network.

> Using a VPN to remotely access the ASU network is required for using the Sol computing cluster. Cisco SSL VPN is a security tool used by all ASU faculty, staff, and students to connect to campus resources securely from off-campus and in addition, providing a secure, encrypted method of data transfer. You will have had some practice using the VPN in the prerequisite course for this CURE, but here are instructions again if you need them.
>
> The CISCO ConnectAnywhere app is free to download at myasu.edu:
>
> MyASU → My Apps → Keyword "VPN" → Download the Cisco ConnectAnywhere for your operating system (PC or Mac)

Figure. How to Download the Cisco SSL VPN through myasu.edu. [1.2_A_CiscoVPN.png (https://drive.google.com/open?id=1qdJW8bU4f1dlN2nkydBe52jfuDRpzfW_)].

Now we’re ready to connect. This video will walk you through connecting to the ASU cluster via the VPN.

Logging into the VPN (https://asuonline.wistia.com/medias/3yhy9k7z6k) | **transcript** (https://drive.google.com/file/d/1klqaZ92YKuzoMi4cAqJU5hlltlYlS65E/view?usp=sharing)

Video. Logging into the Cisco SSL VPN

Step 2. Accessing RStudio Server via the ASU Sol computing cluster.

In order to use RStudio from the Sol computing cluster instead of installing it to your own computer, we will be starting an interactive job on the cluster that will allow you to use **R** (https://www.r-project.org/) in **RStudio** (https://www.rstudio.com/products/rstudio/download/) . This instance of RStudio will look just like a local downloaded version of RStudio but everything needed to run it will be stored on cluster storage instead of on your own computer.

This video refers to logging on the Agave supercomputer, the predecessor to Sol supercomputer, and shows you what it looks like when you start an interactive RStudio Server connection. The details that have changed in the switch to Sol are outlined below, but this provides a nice visual explanation if you have never used RStudio before.

Logging into the RStudio Server (https://asuonline.wistia.com/medias/6sjwmegt1h) | **transcript** (https://drive.google.com/file/d/1Xp-K6YmvgjAlgi1PyVfPrBW-Xc4zHzde/view?usp=sharing)

Here are the steps for starting an instance of RStudio from the Research Computing website:

1. Login to the VPN (virtual private network) using Cisco VPN with your ASUrite credentials. You may need DuoMobile to verify your login.
2. Log into ASU network using VPN; you MUST be logged into the VPN in order to proceed.
3. Navigate to **https://sol.asu.edu** (https://sol.asu.edu) (if you are not logged into the VPN, the webpage does not load).

We will be using the ‘Interactive Apps’ tab throughout the remainder of this course, so it is recommended to bookmark or favorite this link for easy access.

4. Click the ‘Interactive Apps’ menu and select ‘RStudio Server’

Figure. ASU Research Computing site. [**interactive_sessions.png (https://drive.google.com/file/d/1nN1HWh1kwXr8R-K6jdergJkj6M9CGE75/view?usp=drive_link)**].

5. Specify your settings and launch the session

Figure. RStudio Server login screen and parameters. [**Rstudo_fit.png (https://drive.google.com/file/d/1sdfzlt-qX42nPba_nyT_9n4hcPZ3Qzd5/view?usp=sharing)**]

For this course, we will be using the ‘general’ partition, the ‘public’ QOS, and 1 core; this will give our jobs some priority on Sol. Wall time, the amount of time before you get automatically logged off, is set to be up to 3 hours (0-3) but feel free to modify this as you need. We will be using R version 4.2.2. If you have any trouble logging in, reach out to you TAs and peers in Slack channel #troubleshooting

6. Load the session and hit ‘Connect to RStudio Server’ (it may take a few minutes for the button to show up).

   You will first see something like this stating that your job is in the queue:

   Figure. Waiting for RStudio Server to start. [waiting.png (https://drive.google.com/file/d/1vCBVcggfeAc1k6m9Lyy6cOeCgGAEBrMh/view?usp=drive_link)]

   Figure. Launch RStudio Server.. [connect.png (https://drive.google.com/file/d/12nEcJ8YQRYz_oqSk0xvTJ0fdGDDWBFhl/view?usp=drive_link)]

7. You’re ready to start using RStudio Server!

Downloading RStudio to your own computer

You will see these links at the beginning of the tutorial we are using:

Download and install the latest version of R (https://www.r-project.org/)

Download and install RStudio (https://www.rstudio.com/products/rstudio/download/#download)

Asking for help and making comments using Slack

The course’s Slack channel is where you can collaborate with your peers and instructors to find solutions to your problems.

🎦 **Posting questions in Slack (https://fast.wistia.net/embed/iframe/66vf3uvrhn?seo=false)** | **transcript (https://drive.google.com/file/d/1QNWdkdMbN-lEP-qLn_YfvcWKmBLKX2Mh/view?usp=sharing)**

Video. Posting questions in Slack.

In this class we will be using Slack to communicate with each other; a channel is a shared page where multiple people can see messages and respond with reactions, text, gifs, or file sharing. The professor of the class, the TAs, and your peers will be able to view, respond to, and search the channel, so it is a great way to ask for help at any time and learn from reviewing other people’s questions.

The best way to access Slack is through the Canvas menu on the left.

If you go to the Slack channel for this class, we will see a space where you can enter text.

You can go ahead and ask any general questions you have or make announcements by typing into the text box and pressing ‘enter’ or hitting the green button with the paper airplane to the lower right-hand-corner(). To demonstrate, let’s enter a message saying, “Welcome to the Slack channel for the Genomics CURE!”

Figure. Slack message box.

A few other icons you should know are:

- The plus sign (), which allows you to attach files
- The happy face (), because emojis are a MUST

- The at sign (), to tag your TAs and peers
- Check out this link, **Tips for formatting Slack messages (https://help.zapier.com/hc/en-us/articles/8496025607181)** , for more formatting tips

If you would like help debugging code that you wrote, it is very helpful to use Slack formatting features to post code so that others can help you find your problem

How to format code in Slack

There are two options for formatting text as code in Slack:

- The code icon ( ), for a single line of code
  - ProTip: you can also wrap your text in the backtick (`)
  - This will change the font, color, and background color of the code to bring attention to this being code
- The code block ( ), for a block (or several lines) of code
  - ProTip: you can start the text with three backticks (```)

Let’s try a line of code. Let’s copy and paste this normal text into Slack:

```
Please format code like this:
```

press shift enter to make a new line without yet sending the message, press code, and enter some code

```
`print (“Hello World!”)`
```

Figure. Slack message box with one line of code using the code button. [**1.3_G_SlackTextbox_CodeFunction.png (https://drive.google.com/open?id=1_88VjjzNvT78RmGyUCppVv_oOhqNgBis)**].

Note: If you are copying and pasting, you may have to delete the last backtick per line (`) and rekey it to close the line of code. Alternatively, you can type (skipping the backtick), highlight the text, and then click the code button ( ).

Now let’s try a code block. This has similar functionality as the code button, but it draws a box around the code. To demonstrate, here’s another message.

If we were to type something with multiple lines and then use the code button, it would not be as readable. Let’s see what happens when we have more than one code line:

Please format code like this:

```
`print (“Hello World!”)`
`print (“Hello starshine, the earth says hello”)`
```

Figure. Slack message box with two lines of code using the code button.

This looks like code, but it is disjointed. The more lines of code you have, the more obnoxious this looks too; we want everything to be readable and in one block. This time, let’s use code block:

~~~
Please format code like this:
```
print (“Hello World!”)
print (“Hello starshine, the earth says hello”)
~~~

Figure. Slack message box with two lines of code using the code block.

Note: If you are copying and pasting, you may have to delete the set of backticks (```) then copy and paste in the lines of code. Alternatively, you can type (skipping the backticks), highlight the text, and then click the code button ( ).

Ah, much better. The code block makes it easier to communicate your code and for others to test out your code as well. If you have any questions about using Slack, please post in the #general channel to your TAs and peers. Remember your netiquette, but most of all, that this is a judgment-free zone where finding the answers to questions together helps the entire class.

How to use RStudio

If you have never worked with RStudio before, this video gives a good overview of the most important panels and how to switch back and forth between them. **RStudio for the Total Beginner** (https://youtu.be/FIrsOBy5k58?feature=shared)

Assignment: R Programming Tutorial, part 1

In Modules 1 and 2, we are assigning parts of this tutorial to help you confidently code in R so that we can do analysis for our research project:

R for Reproducible Scientific Analysis (**https://swcarpentry.github.io/r-novice-gapminder/index.html (https://swcarpentry.github.io/r-novice-gapminder/index.html)**)

Please note that:

1. There is an option to view all episodes on one page at the bottom of the site, if that makes it easier to look back and forth
2. We will be skipping episodes 12-14 since we are not planning to use the plyr, dplyr, and tidyr packages in our analysis code
3. Do copy and paste from the tutorial into RStudio Console to prevent typos
4. You will be asked to turn in an R script that has all the commands you ran in the tutorial

Please complete the episodes up to and including episode 6 for Module 1 this week:

1. Summary and Setup
    - This episode will talk about how these concepts are broadly applicable outside of the example analysis they are showing you
    - If you are using the Sol RStudio Server, follow the instructions above, otherwise use the instructions they provide to download R and RStudio locally on your computer
    - The data set you will be using for this tutorial will be installed as a package in the ‘Introduction to R and Studio’ episode next
2. Episode 1: Introduction to R and RStudio
    - This episode will help you to identify the important panels in RStudio where you will be looking at your data, code, and results
    - It will teach you basic data manipulation you can do with R in RStudio
    - While you are in this episode, create a new R script where you can save commands you are trying in the Console so you can turn it in as an assignment
        - In RStudio, select File → New File → R Script
        - This is open a tab containing a blank R script

    - After you have gone through an episode by interacting with the Console, click the History tab in the top right to see all the commands you entered
    - Click on any command will highlight it
    - You can highlight blocks of code by pressing shift and using the arrows to go up and down
    - Once you have highlight the code you feel represents the stuff you tried during the episode, click the To Source button to copy the code into the new script
    - Save the script (in your /scratch/username directory if you are working on the server or in a directory of your choice if you are working locally)
    - Do this for every episode so you have a collection of commands you explored in this module

3. Episode 2: Project Management With RStudio
    - While we will not need to officially create a new Project in RStudio, the concepts covered in this episode about how to use data and scripts are great advice for programming
    - This episode show you how to organize data into directories and how to view and manipulate files in directories through the RStudio tabs
    - Once you create a project in RStudio, the info box called Challenge 3 will prompt you to download a data file called 'gapminder.csv' for use in this tutorial
    - You will want to save it inside the 'data/' directory that you make in this Episode
4. Episode 3: Seeking help
    - This episode shows you how to use the Help features in R to look up descriptions of functions and packages
    - It also introduces you to techniques you can use to search for coding solutions
5. Episode 4: Data Structures

- This episode covers how data is represented and stored in R

6. Episode 5: Data Frames
    - This episode specifically covers data frames, a type of multi-dimensional data variable that can contain data of different types that is used widely in R
7. Episode 6: Subsetting Data
    - This episode shows you how to access parts of multi-dimensional data

Great job on making it this far!! We will continue with episodes 7 through 11, 15, and 16 in the next module.