BCB410H1: Applied Bioinformatics 2022

Lecture 1
15 Sept 2022

Anjali Silva, PhD

a.silva@utoronto.ca anjalisilva.github.io @Silva_Anjali

Welcome

- Instructor: Anjali Silva, PhD
  - Researcher, University of Toronto
  - Pronouns: she/her
  - Email: a.silva@utoronto.ca
    - **Use subject line "BCB410" for emails**.
    - E.g., BCB410: inquiry regarding assessment I.

Outline

- Introduction
- Syllabus
- Academic Integrity
- Bioinformatics
- R, Packages and RStudio
- Practical

BCB410H1

- BCB410H1: Applied Bioinformatics
  - Practical introduction to concepts, standards and tools for the implementation of strategies in bioinformatics and computational biology. **Student led discussions** plus a strong component of **hands-on exercises** [2022 UToronto Calendar].

Prerequisites

- Prerequisites
  - BCH311H1/ MGY311Y1; (CSC324H1/ CSC373H1/ CSC375H1) or permission of the course coordinator.
  - Will assume prior knowledge of biological systems.
  - Will assume prior knowledge of programming principles.
  - Not an introductory course to R; if you have no/limited prior knowledge, will have to pick up fast.

Welcome

- Class
  - Wednesday 10:00 am – 12 noon EST; online - synchronous.

Welcome

- Aim
  - In this year's course you will define a useful tool for the analysis of biological data, write an R package to support it, review and critique other packages, and improve and document your work.

Welcome

- Content - 5 phases:
  - Section I: Learn R and basic structure of an R package.
  - Section II: Define a tool for data analysis.
  - Section III: Develop the tool into an R package and keep track using Git.
  - Section IV: Review peer packages.
  - Section V: Improve own package based on peer review and final submission.

Course Overview

- Tentative Calendar*

  * This may be modified as needed.

  - See syllabus.

Course Overview

- Assessments

Activity	Weight
1) Participation	see syllabus
2) Journal Entries/Assignments	
3) Initial submission of R package + Presentation	
4) Review of peer R packages*	
5) Final submission of R package	
Total	100

* Number of participation panels maybe altered. But total marks remain the same.

Course Overview

- Read syllabus carefully for deadlines and late policy.
- If any questions, send me an email or see me in office hours.

Course Overview

- Lecture material will be available on Quercus.
- Assessment submissions will be done on Quercus and GitHub, unless stated otherwise.
- You are responsible for uploading files on time, in the format specified.
- If technical difficulties with Quercus, must email instructor a copy of submission before the deadline.
- Quercus history will be checked and emailed copy will be graded, accordingly.

Course Overview

- Textbooks
  - *R packages* by Hadley Wickham and Jennifer Bryan
    https://r-pkgs.org/index.html
  - Available free online.

Course Overview

- Other recommendations:
  - *R packages* covers the basic components.
  - More in-depth details at:

    Writing R Extensions: Creating R packages
    https://cran.r-project.org/doc/manuals/R-exts.html#Creating-R-packages.
  - Recommended once you master the basics.

Course Overview

- Other recommendations:
  - *Advanced R* by Hadley Wickham
    https://adv-r.hadley.nz/functions.html
  - *What They Forgot to Teach You About R* by Jennifer Bryan and Jim Hester
    https://rstats.wtf/
  - Available free online.

Course Overview

- Academic Integrity
  - You are responsible for understanding policies on academic integrity.
  - https://www.academicintegrity.utoronto.ca/
  - https://guides.library.utoronto.ca/plagiarism
  - http://steipe.biochemistry.utoronto.ca/abc/index.php/ABC-Plagiarism

Course Overview

- Full disclosure policy for this course:
  - If it's not your own, new idea, it has a source.
  - All sources must be referenced.
- For advice:
  - How not to plagiarize https://advice.writing.utoronto.ca/using-sources/

Course Expectations

- When submitting assessment files to Quercus, label using this format: LASTNAME_FirstInitial_**Assessment**.format.
  - E.g., SILVA_A_A1.PDF
- Instructions of each assessment will specify **Assessment** name.
- Must follow this format to avoid confusion with name and assessment weight.

Course Expectations

- Bioinformatics is a highly interdisciplinary field. Be open to different views.
- Will need Internet access to use or download free bioinformatics software and tools required for class work.

Course Expectations

- **Use subject line "BCB410" for emails**.
  - E.g. BCB410: inquiry regarding assignment I.
- Weekday: 48h and Weekends: 48h - 72h reply.
- Use practical time to ask questions.

Course Etiquette

- Respectful listening and sharing; One speaker at a time.
- Keep 'mute' status unless you need to ask a question.
- Say your name before asking/answering a question.
- Personal grade matters must be discussed privately.

Discussion Boards

- Introduce yourself in the 'Get to know each other: BCB410' Discussion Board.
- Use 'Lecture/Practical Questions' Discussion Board to ask questions.
- Will also be used to post presentation links for the entire class.
- Personal grade matters must be discussed privately.

Discussion Boards

- Respect Discussion Board rules at all times.

/Users/as/Desktop/Images/Figure8

Any questions?

Let's begin the lecture...

Central Dogma

/Users/as/Desktop/Images/Figure4.

Figure: The classic view of the central dogma of biology states that genetic information in DNA is transcribed into messenger RNA (mRNA) and each mRNA contain information to synthesize a protein. But with new discoveries, there are exceptions to this. For example, DNA that does not encode proteins may encode different types of functional RNAs. [Image by user EEPUCKETT, 2015; Transcriptomics for Conservation, wildlifesnpits.wordpress.com]

OMICS

OMICS is generating large amounts of data that require analysis.

/Users/as/Desktop/Images/Figure2

Figure: Central role of bioinformatics in the modern biological investigation based on omics sciences [Facchiano, 2016].

Bioinformatics

- Term "bioinformatics" was coined by **Paulien Hogeweg** and **Ben Hesper** in **1970s** to describe "the study of informatic processes in biotic systems" [Hogeweg, P. *PLoS Computational Biology*, 2011].
- Can broadly be defined as the science of storing, retrieving and analysing large amounts of biological information [ebi.ac.uk, 2018].
- Involves information related to biological macromolecules such as **DNA, RNA, proteins and metabolites**.

Bioinformatics

/Users/as/Desktop/Images/Figure1.

Figure: Bioinformatics is a highly interdisciplinary, fast growing field. This figure from De Maio et al., 2018 shows the number of publications in PubMed on Bioinformatics.

Tools

/Users/as/Desktop/Images/Figure10

Figure: Many bioinformatics tools and resources are available on the internet. However, biological datasets can be very large, requiring command-line tools to manipulate the data. [Pevsner, Bioinformatics And Functional Genomics, 3rd ed].

Databases...

Central Bioinformatics Resources

/Users/as/Desktop/Images/Figure11.

Figure: NCBI is one of the central bioinformatics sites. NCBI develops/ maintains databases, software for searching and analysis of data. Link: https://www.ncbi.nlm.nih.gov.

Central Bioinformatics Resources

/Users/as/Desktop/Images/Figure12

Figure: NCBI is one of the central bioinformatics sites.

Central Bioinformatics Resources

/Users/as/Desktop/Images/Figure13.

Examples of Databases

/Users/as/Desktop/Images/Figure6.p

Figure: INSDC (http://www.insdc.org/) coordinates DNA sequence data. GenBank is maintained by National Center for Biotechnology Information (NCBI), European Nucleotide Archive (ENA) is maintained by European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI). DDBJ is the DNA Data Bank of Japan.

Examples of Databases

/Users/as/Desktop/Images/Figure7.

Figure: Different databses available for RNA (https://rnacentral.org/expert-databases), including RNA alignments, RNA sequences and RNA structures.

Examples of Databases

/Users/as/Desktop/Images/Figure8.

Figure: UniProt: the **Universal Protein** provides high-quality and freely accessible resource of protein sequence and functional information. The UniProt Consortium comprises of the EBI, the SIB and the PIR. Prior to 2002, the Swiss-Prot and TrEMBL were protein databases maintained by EBI and SIB. Protein Sequence Database (PSD) was a protein database maintained by PIR. Around 2002, these three databases were merged to form the UniProt.

Any questions?

R

What is R?

- A language and environment for statistical computing and graphics.
- R was initially written by Ross Ihaka and Robert Gentleman.
- Since mid-1997, the R Core Team modify the R source.
- R runs on a wide variety of UNIX platforms, Windows and MacOS.

R continue...

- R is a scripting language, thus an interpreter executes commands one line at a time.
- A Free software under the terms of the GNU General Public License.
- R home page: https://www.R-project.org/
- How can R be obtained?
  - Via CRAN, the "Comprehensive R Archive Network".
  - https://cran.r-project.org/

R continue...

- How can R be installed?
  - Unix
  - https://cran.r-project.org/doc/FAQ/R-FAQ.html#How-can-R-be-installed-_0028Unix_002dlike_0029
  - Windows
  - https://cran.r-project.org/bin/windows/base/
  - Mac
  - https://cran.r-project.org/bin/macosx/

R continue...

/Users/as/Desktop/Images/Figure15.png

/Users/as/Desktop/Images/Fi

R continue...

- R can be used interactively or non-interactively.
- Interactively, with or without an integrated development environment (IDE): RStudio.
- Non-interactively via scripts.
- R is designed with interactive data exploration in mind.
- A version of R is released each year. Current release is 4.0.2.

Documentation for R

- Online documentation for functions and variables in R exists.
- Obtained by typing *help(FunctionName)* or *?FunctionName* at the R prompt, where FunctionName is name of function.
- E.g., if 'sum' is the function then:

```
> help(sum)
> ?sum
```

R packages

- Mechanism for extending the basic functionality of R.
- It is natural to put together many functions together into a package achieving a specific goal.
  - Function for preprocessing data.
  - Function for clustering data.
  - Function for selecting best cluster.
  - Function to visualize the clustering results.
  - Put together = Package for Clustering.
- Provide a defined interface, with inputs (arguments) and outputs (return values).

R packages

- Building R packages requires tools that must be in place before process of development can start.
- Mainly R and RStudio (recommended).
- Mac OS
  - Xcode development environment
  - https://apps.apple.com/us/app/xcode/id497799835?mt=12
- Windows
  - Rtools
  - https://cran.r-project.org/bin/windows/Rtools/

R packages: Mac OS

- For more information: https://r-pkgs.org/setup.html
- Mac OS
  - Xcode development environment
  - https://apps.apple.com/us/app/xcode/id497799835?mt=12
- Then, in the shell, do:

```
xcode-select --install
```

R packages: Windows

- Windows:
  - Rtools
  - https://cran.r-project.org/bin/windows/Rtools/
- For more information: https://r-pkgs.org/setup.html
- During the Rtools installation you may see a window asking you to "Select Additional Tasks".
  - Do not select the box for "Edit the system PATH". devtools and RStudio should put Rtools on the PATH automatically when it is needed.
  - Do select the box for "Save version information to registry". It should be selected by default.

R packages: Linux

- For more information: https://r-pkgs.org/setup.html
- Install R, but also the R development tools. For example, on Ubuntu (and Debian) you need to install the r-base-dev package.

What R packages are available?

- CRAN
  - >16K packages [as of 2022]
  - https://cran.r-project.org/web/packages/
- Bioconductor
  - >1900 packages [as of 2022]
  - https://bioconductor.org/packages/release/bioc/
- GitHub
  - > 63K results [as of 2022]
  - https://github.com/search?q=r+packages&type=Repositories

RStudio

- RStudio is not required to build R packages.
- However, it contains many features that make the development process easier and faster.

RStudio

/Users/as/Desktop/Images/Figure17

Figure: Anatomy of RStudio. 1. This is the Console. 2. Environment and History. 3. Files, Plots, Packages, Help and Viewer.

RStudio

/Users/as/Desktop/Images/Figure83

Figure: Tools → Keyboard Shortcuts Help.

Any questions?

Practical

RStudio

- Let's open up RStudio.

/Users/as/Desktop/Images/Figure

- On Console, get working directory:

```
> getwd()
```

- To set to desired directory
  Session → Set Working Directory → Choose Directory...

RStudio

- Alternatively, you may use:

  ```
  > setwd("/../../..")
  ```

- To open a new script:
  File → New File → R Script

- Save this:
  File → Save → Practical_StudentName.R

- Practical_StudentName.R is called a script.

R Features

- In R, the indexing begins from 1.
- R is case sensitive ("X" is not the same as "x").
- R uses dynamic variable typing, so variables can be used over and over again.

Assignment and Commenting

- The ← symbol is the assignment operator.
- To assign a value to a variable called 'test1'

```
test1 <- 123
test1
```

- Comment using # character

```
test1 <- 123  # This is a comment
test1 # This is called auto-printing
```

Over-writing

- From previous slide we had:

```
test1 <- 123
test1
```

- Over-write previous value of the 'test1' variable with a new value:

```
test1 <- test1 + 2
test1 # 125
```

- Over-write previous value of the 'test1' variable with a new value:

```
test1 <- 5 + 2
test1 # 7
```

Version

- To obtain session information

  `sessionInfo()`

- Version information:

  `R.Version()`

- Show objects in workspace

  `ls()`

R Built-in Functions

- There are many built-in functions. You will learn these as you go.
- The "argument" of the function is provided inside the brackets.
- The "return value" of the function is the value provided back.
- We will cover some basic functions:

```
x <- 5
x # auto-printing
print(x) # explicit printing
class(x) # "numeric"
typeof(x) # "double"
length(x) # 1
```

R Built-in Functions

- Return value from functions can be assigned to a variable or printed:

```
x <- 5
x # auto-printing

y <- x + 5
y # 10

z <- tyepof(y) # return value  assigned to variable
z # "double"
```

R Help Function

- Getting help:

```
?"<-"  # help on assignment operator
help("<-") # help on assignment operator

?typeof # help on typeof function

?class # help on class function

?print # help on print function
```

Any questions?

R Data Types

- Numeric: floating types (double precision).
- Logicals: booleans = TRUE/FALSE or T/F.
- Character strings.
- Examples:

```
xValue <- 100
xValue

yVariable <- FALSE
yVariable

zVariable <- "hello"
zVariable
```

R Class

- Numbers in R are usually treated as numeric objects (i.e. double precision real numbers).
- To explicitly assign an integer, need to specify the L suffix.

```
x <- 1L
x
class(x) # "integer"
```

R Class

- Complex class:

```
x <- c(2 + 0i, 5 + 4i)
class(x) # "complex"
```

- Inf represents infinity:

```
Inf
1 / Inf # 0
```

- NaN represents an undefined value/missing value:

```
NaN # not a number
0 / 0 # NaN
```

Concatenating

- c() function concatenating elements together:

```
x <- c(0.5, 0.6)
class(x) # "numeric"

x <- c("a", "b", "c")
class(x) # "character"

x <- c(TRUE, FALSE)
class(x) # "logical"
```

Character Strings

- Character strings are collections of characters.
- Provided as values in single or double quotes.

```
xVariable <- ‘hello’
class(xVariable) # "character"

zVariable <- "hello"
class(zVariable) # "character"
```

- “paste” converts inputs to strings, concatenate and return:

```
paste(xVariable)
```

Character Strings

- "cat" concatenates and prints the arguments to the screen:

```
cat("\n", xVariable, zVariable) # "\n" adds new line
```

- "print" prints the argument:

```
print(c(zVariable, xVariable))
```

Missing Values

- Missing values are denoted by NA (Not Available) or NaN (Not a Number).

```
x <- c(1, 3, NA, 4, 5)
class(x) # "numeric"

y <- c(1, 3, NaN, 4, 5)
class(y) # "numeric"

# is.na() is used to test objects if they are NA
# is.nan() is used to test for NaN

is.na(x) # FALSE FALSE  TRUE FALSE FALSE
is.nan(x) # FALSE FALSE FALSE FALSE FALSE
```

Question: What is the difference between NA and NaN in R?

Any questions?

- To do: Journal Entry 1 (Note, may need a distribution of Latex installed).
- Take a look at 'Initial submission + Presentation of R package'.

Practical

- Today we looked at the following topics.
  - Assignment and Commenting
  - Over-writing
  - Built-in Functions
  - Help
  - Classes
  - Concatenating
  - Character Strings
  - Missing Values

Practical - Tips for Solving Issues

- Copy and paste the entire **exact** error message into Google.
  - Someone else may have gotten this same error and has asked a question.
- Copy and paste the entire error message into Google, followed by 'r'.
- Google the name of the function with term 'tutorial r' to see tutorials.
- If struggling with code for a plot, Google 'r plot plotname', then click on Images.
- If errors with reading files, ensure path is correct. Check using getwd().