

Introducing Python and R: Which should you use?

Spring 2020 Lecture 0

Connor Hurd

Goals of today's lecture

- Differentiate R vs. Python for biological science applications
- Determine the best language/approach for your programming needs
(Spoiler: you should learn both languages)
- Introduce Python and R languages briefly
- Introduce useful programs for R and Python languages
- Provide links for useful resources that students should review at home
- Download R or Python (Or Both!!!)
- Lecture 30-45 minutes, assistance with downloads/questions afterwards

Python vs. R for biologists



Python

- Broader range of applications than R
- Better for learning programming fundamentals if you don't already have coding experience
- Great for Molecular Modeling/Dynamics applications

R

- Superior to Python for statistical analysis due to exclusive, specific packages
- Easier to debug than Python thanks to Rstudio
- Definitely the better choice for fast RNA-seq data and ChIP-seq data analysis.
Our upcoming CodeOn! Lectures will cover such analysis.

Python vs. R for biologists

Python

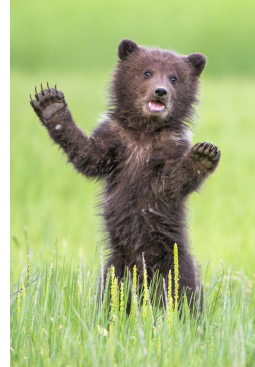
- Less suited for working with data tables than R (**pandas** package helps)
- **Matplotlib** and **seaborn** for figure making
- Has a linear learning curve; easier to become an expert in Python than in R for beginners due to the focus of readability in Python.

R

- Great for working with tables due to the data frame function
- **ggplot2** for figure making
- Easier to learn upon starting out, but has a slightly sharper learning curve and is harder to become an expert than in Python



Guiding Questions for Choosing a Language



What language does your lab use?

This possibly the most important question; it is incredibly helpful to be able to adopt and modify pre-existing scripts of the same language from your colleagues. Code is rarely written de novo.

Are your purposes for coding strictly for biological data, or do you hope to learn other programming skills as well?

If strictly biological, focus on learning R and finding the packages you need. If you hope to learn more programming skills like machine learning and web development, it is essential to be proficient in Python (in addition to knowing R for biological data packages)

Python vs. R: The Verdict

More time spent, higher reward:

Use Python to introduce yourself to programming and to get experience with CLI/bash and coding in general. This will make learning R and R's statistical packages easier, and you will have a much broader range of programming skills.

Less time demand, more specific application:

Focus on learning R only and familiarize yourself with R packages like **Sleuth** and **DESeq**



User loyalty, Python vs. R

74% OF R USERS
REMAIN LOYAL TO R



10% switch from R to Python



5% switch from Python to R

91% OF PYTHON USERS
REMAIN LOYAL TO PYTHON



Source: KDnuggets polls 2016

Why choose one? Implement both languages!

- R scripts can be run within Python using tools like **RPy2**
- Python can be run from R using **reticulate**



Very useful articles on how to use both R and Python for a single project:

<https://towardsdatascience.com/from-r-vs-python-to-r-and-python-aa25db33ce17>

<https://community.alteryx.com/t5/Data-Science-Blog/RPy2-Combining-the-Power-of-R-Python-for-Data-Science/ba-p/138432>

Before we introduce R/Python: What is Coding?

Giving human-readable commands (programming language such as R or Python) to your computer that are ultimately converted into binary (machine language) and executed.



Understanding Computers -- Why Binary?

- Nothing else would work based on how computers are designed.
- At their core, computers are billions of on and off switches.
- These switches are transistors that are on (1) or off (0).
- We can't speak binary, so we use programming languages to translate our code into binary for the computer.

`print("Hello World")`



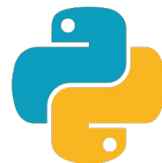
Tree of Life vs. Tree of Coding

Life

Elementary Particles - Subatomic Particles - Atoms - Molecules - Organelles
- Cells - Tissue - Organs - Person

Coding

Bit - Byte - Assembly Language - Coding Language - Program - GUI



Introducing: The Python Language

- Refers to a system of coding syntax that you can use in IDE, CLI, or simple text editor.
- Two versions, Python 2.7 and Python 3.x (currently 3.6)
- Python is an Object Oriented (OO) Language (meaning a focus on “objects and data”, not “actions and logic”)
- Python has many applications for scientific/biological data analysis/plotting, but may be inferior to R for specific biological applications.



Recommended Python IDE: Jupyter Notebooks

- Jupyter is an IDE with individual cells
- Used to be iPython notebook, evolved into Jupyter notebook.
- Runs inside a web browser
- Notebook Form (Figures, equations, text-rich components)
- Figure making programs like Illustrator that use GUIs are easier to use and more front end than creating a Jupyter notebook, but ultimately much more limited.
- Everything you make in the notebook (besides literal text strings) is with code. Through installation of Numpy, Matplotlib, and other plotting/figure-making modules, you can make figures in output with input of your code.

Can interpret Python and R using kernels for each of these languages. Uses iPython kernel for Python interpretation.

Useful Python Bioinformatics Software: Biopython

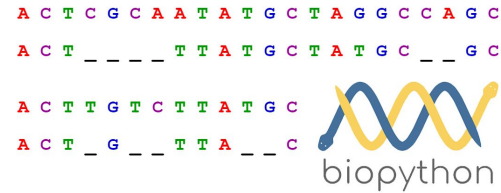
Biopython

Biopython is a library that was created with sequencing work in mind. This library contains classes that allow for easier work with RNA-Seq data, and allows for access to NCBI data within your program.

Learn more:

<http://www.bioinformatics.org/bradstuff/bp/tut/Tutorial002.html>

<https://kb.iu.edu/d/baii>



I will show this slide again at the end of the presentation!

Installing Python: Downloading Anaconda and opening Jupyter

<https://www.anaconda.com/download/>

Opening Jupyter

1. Click Anaconda icon to open GUI where you can click and open jupyter
2. Launch with *jupyter notebook* command from command line.

By navigating to certain directory in terminal and then running jupyter notebook, you will launch the jupyter Dashboard from your working directory.

Useful (and free) online Python courses

“Using Python for Research” -- a free online Harvard course with completely new material (Jan 2020)

<https://www.edx.org/course/using-python-for-research#!>

Datacamp python course -- made for complete beginners

<https://www.datacamp.com/courses/intro-to-python-for-data-science>



Introducing: The R Language

R is the programming language we use. R was created using a mixture of other programming languages in the 1990's by statisticians.

RStudio is an interface that was created to allow easier use of R.

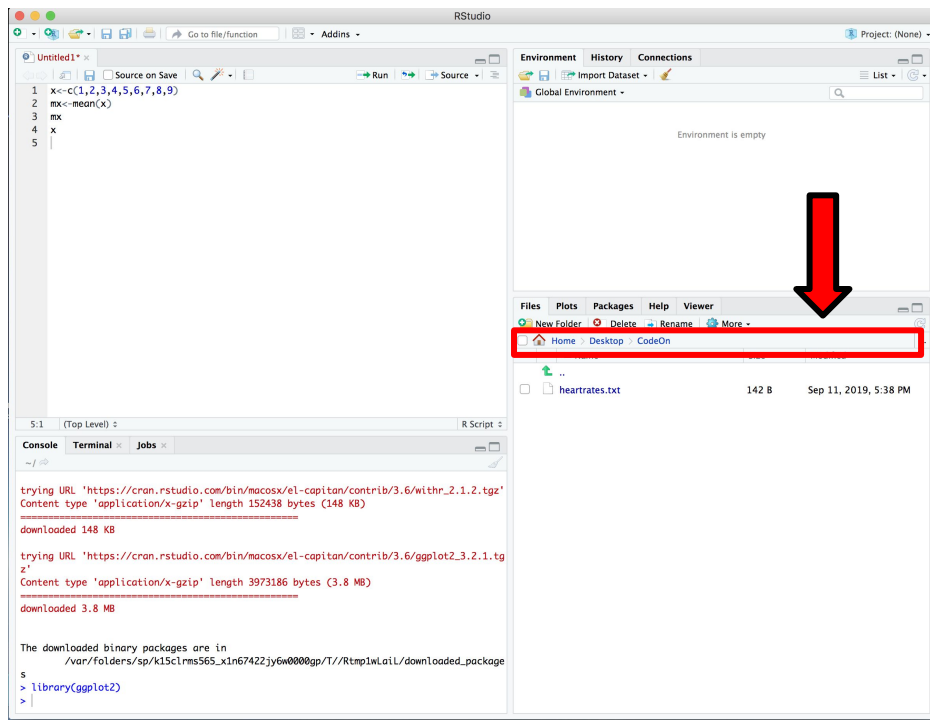
R and RStudio were created using code. This code is referred to as the **source code**. It's unlikely that you will ever need to interact with the source code for any programs you use .

The biggest advantages of R combined with Rstudio over other programming languages are:

- Better visualization of data
- R is best adapted for statistical analysis
- Easiest interface for training biological data scientists

Recommended R IDE: Rstudio

- Rstudio is a GUI/CLI hybrid. It's more common to refer to RStudio as an IDE.
- Rstudio was created from source code written in the Java and C++ programming languages.



Notice the pathway of folders and files shown in the “Files” tab. This tells us the working directory we’re operating RStudio from. This matters for loading in data from separate files.

Script file

Write code here

To run code put your cursor on the line and click the **run button**

Edit to correct errors

⇒ record of commands that worked

Save scripts with the **.R** extension

⇒ syntax will be highlighted

⇒ good practice

<- is the assignment operator

⇒ puts what is on the right in to the object on the left

⇒ Assign results if you want to use them again

Console

When you click run, code is sent to the console and executed

> is the prompt

⇒ do not type it

⇒ appears when R is ready for next command

Command output goes here by default

⇒ output is in a different colour

⇒ [1] indicates 3.4 is the first element of the output

⇒ many commands will not have output, the prompt just reappears

Script: where you write code

Console: where output goes

Environment

Name objects by assignment to use them again

All the **objects** you created in your session

Saving the environment saves all the objects, but not the code with a **.RData** extension

History

A history of every command you sent to the console, mistakes included.

File can be saved but usually you just need the script

Environment: where saved output goes

Packages

Many functions come with R

A huge amount of extra functionality is available in packages

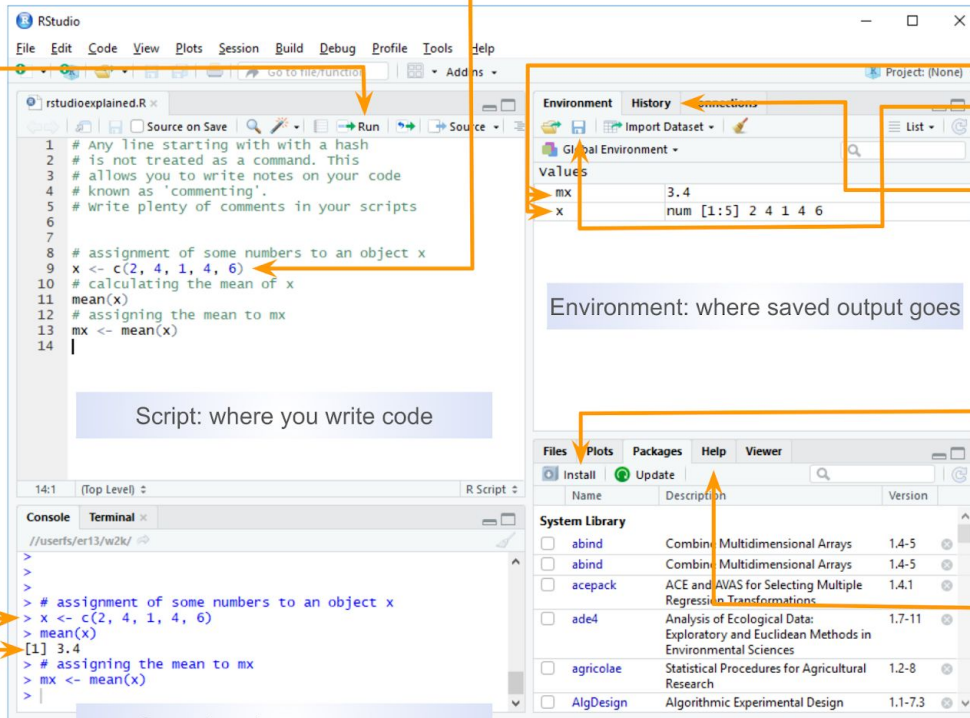
Packages can be installed by clicking the Install button

Help

Access to manual pages for all installed packages

Plots

Figure output appears here



Useful R resources

CRAN and Bioconductor

Download new packages and software for use in Rstudio. Bioconductor has specific packages for bioinformatics applications in R.

Useful blog written by an experienced computational biologist:

<https://www.badgrammargoodsyntax.com/>

- Blog by biologist who uses R for RNA seq and epigenetics applications

Useful R RNA-seq analysis software: recount


recount is a package you can download from bioconductor to analyze your RNA-seq data. **recount2** is an adaptation of this package that was used and published in a 2017 Nature Biotechnology study.

recount package download on Bioconductor:

<http://bioconductor.org/packages/release/bioc/html/recount.html>




recount RNA-seq Nature paper:

<https://www.nature.com/articles/nbt.3838?draft=collection>

nature
biotechnology

Correspondence | Published: 11 April 2017

Reproducible RNA-seq analysis using *recount2*

Leonardo Collado-Torres, Abhinav Nellore, Kai Kammers, Shannon E Ellis, Margaret A Taub, Kasper D Hansen, Andrew E Jaffe , Ben Langmead  & Jeffrey T Leek 

Nature Biotechnology **35**, 319–321(2017) | [Cite this article](#)

3234 Accesses | 65 Citations | 126 Altmetric | [Metrics](#)

Useful (and free) online R courses

Fast and easy to read crash course in R written for biologists:

<https://bioinformatics-core-shared-training.github.io/r-crash-course/>

Interactive intro course on R language designed for beginners:

<https://www.datacamp.com/courses/free-introduction-to-r>

- Not based in R studio, but has script/console that works exactly the same.

Installing R: Downloading R and Rstudio

Install R -

<https://cran.rstudio.com/>

Install RStudio -

<https://www.rstudio.com/products/rstudio/download/>

Installing Python: Downloading Anaconda and opening Jupyter

<https://www.anaconda.com/download/>

Opening Jupyter

1. Click Anaconda icon to open GUI where you can click and open jupyter
2. Launch with *jupyter notebook* command from command line.

By navigating to certain directory in terminal and then running jupyter notebook, you will launch the jupyter Dashboard from your working directory.

END