Weekly Discussion Sections & Readings

Teaching Fellows (TA)

Name	Office	Email
Mengting Gu	Bass 437	mengting.gu (at) yale.edu
Paul Muir	Bass 437	Paul.muir (at) yale.edu

Please E-mail cbb752@gersteinlab.org so both TAs will be informed

DS Format

- Group 1: Friday 1:00 - 2:00 PM, BASS 405
- Group 2: Tuesday 6:00 - 7:00 PM, BASS 405

1. We will discuss **two** selected articles from the primary literature.
2. Write-up approx. a half page (2-3 paragraph) summaries of each paper.
3. Each student will give approx. **20 min presentations** about the paper.

Write-ups and Presentations

- For write-ups and presentation, think about the following:
  - What was missing in the field? (introduction/background)
  - What were the questions the paper aim to address? (hypothesis)
  - What they did and what was the result? (method/results)
  - Conclusion and future direction (discussion/conclusion)
- During each presentation, student audiences will ask questions and discuss the topic presented.
- Please sign up for the presentation using this spreadsheet. Also available on class website: cbb752b17.gersteinlab.org/materials

Introduction to R, Python & GitHub

Slides adapted from last year

Programming Assignments (Req'd for CBB and CS students)

- For the programming assignments, you can use either R or Python. However, if you would like to use other programming languages, please contact the TAs and request for a permission.
- There will be **THREE** homework assignments. We will try to promote the idea of reproducible research and using version control system, specifically GitHub, in facilitating the process of homework submission.

Introduction to R

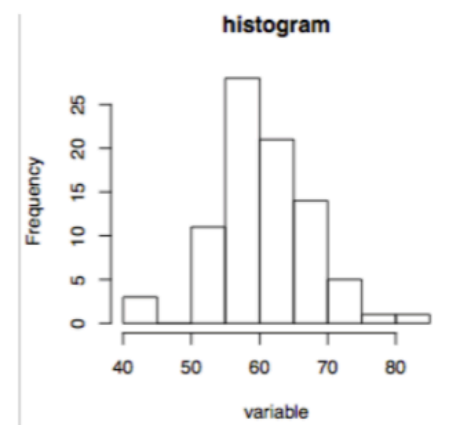
What is R

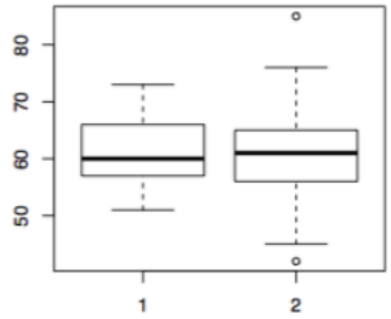
- A well-developed, simple and effective programming language;
- Statistical packages:
  - Statistical computing and graphics (linear and nonlinear modelling, classical statistical tests, time- series analysis, classification, clustering, …);
- Command-line interface;
- Matrix-based programming language;
- Free.

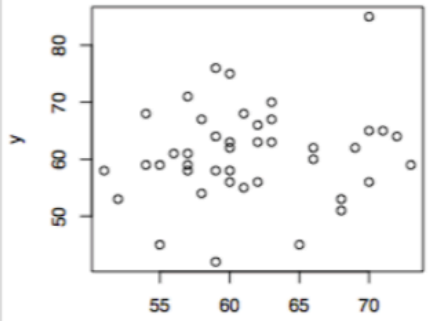
```
> 3+5
[1] 8
> 2*7/9+2/(3+10)
[1] 1.709402
> a<-matrix(1:9,3,3)
> a
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> t(a)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
> b<-c(1,2,3)
> a%*%b
     [,1]
[1,]   30
[2,]   36
[3,]   42
```

R examples

Mathematical calculations



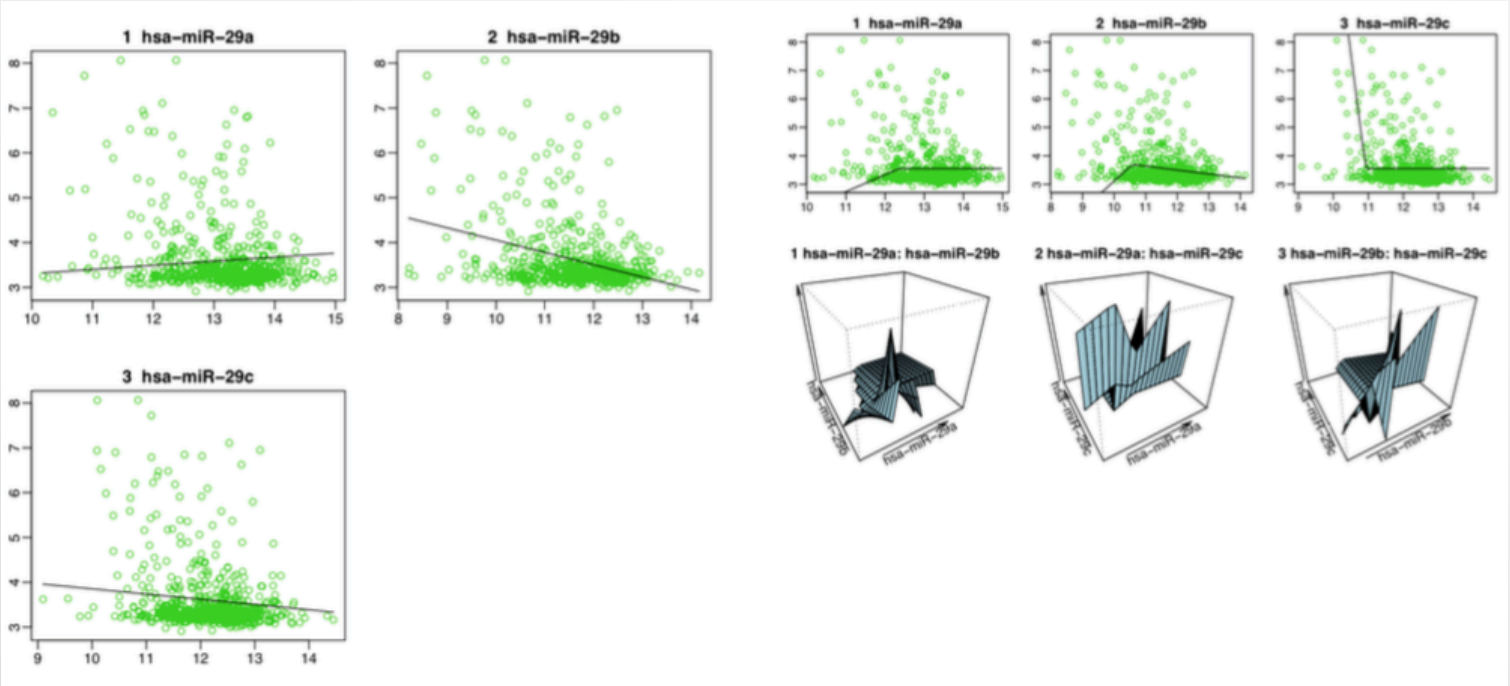





```
> summary(dat)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  42.00   57.00   60.50   61.24   65.25   85.00
```

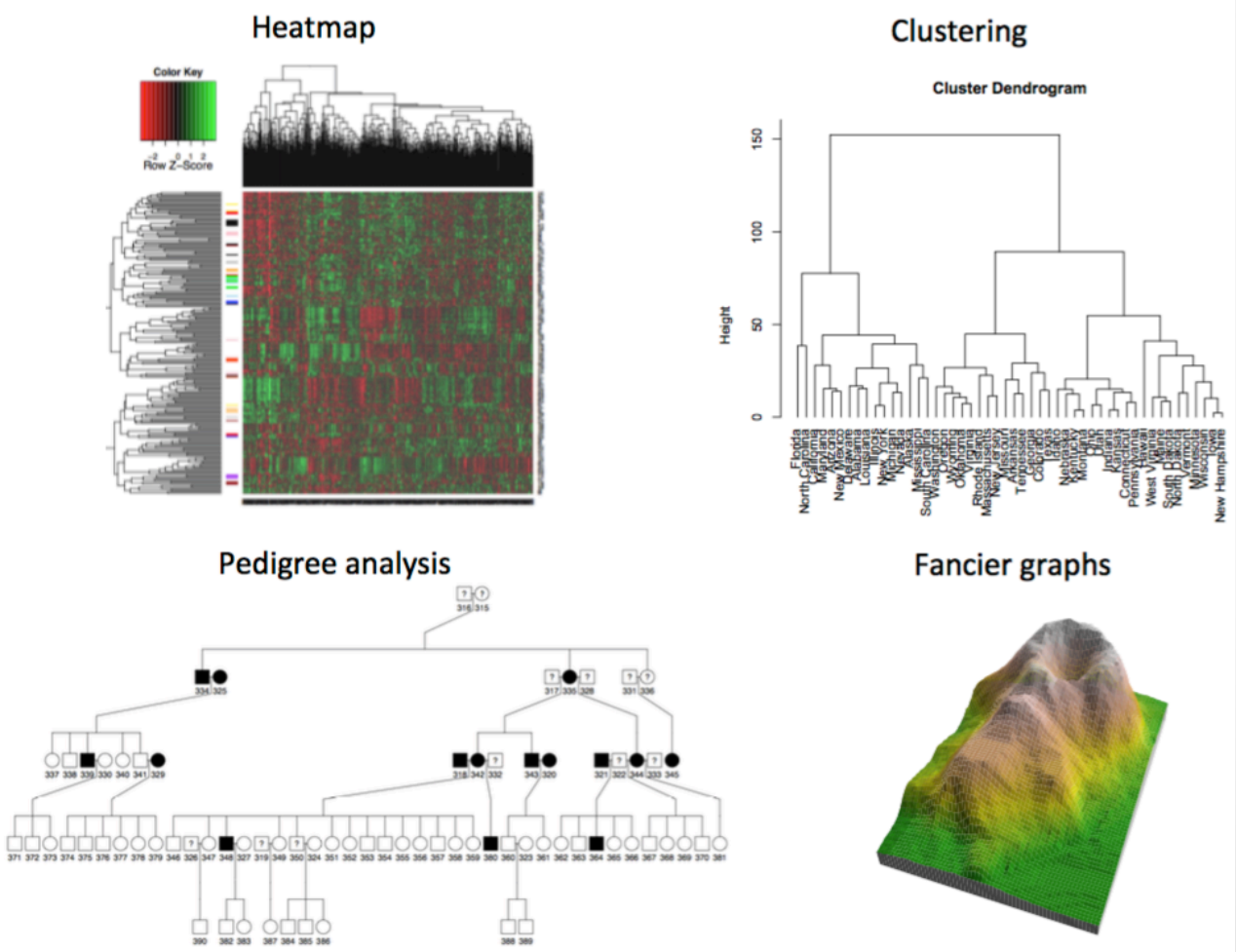
R examples

Data Summary



R examples

Data modeling

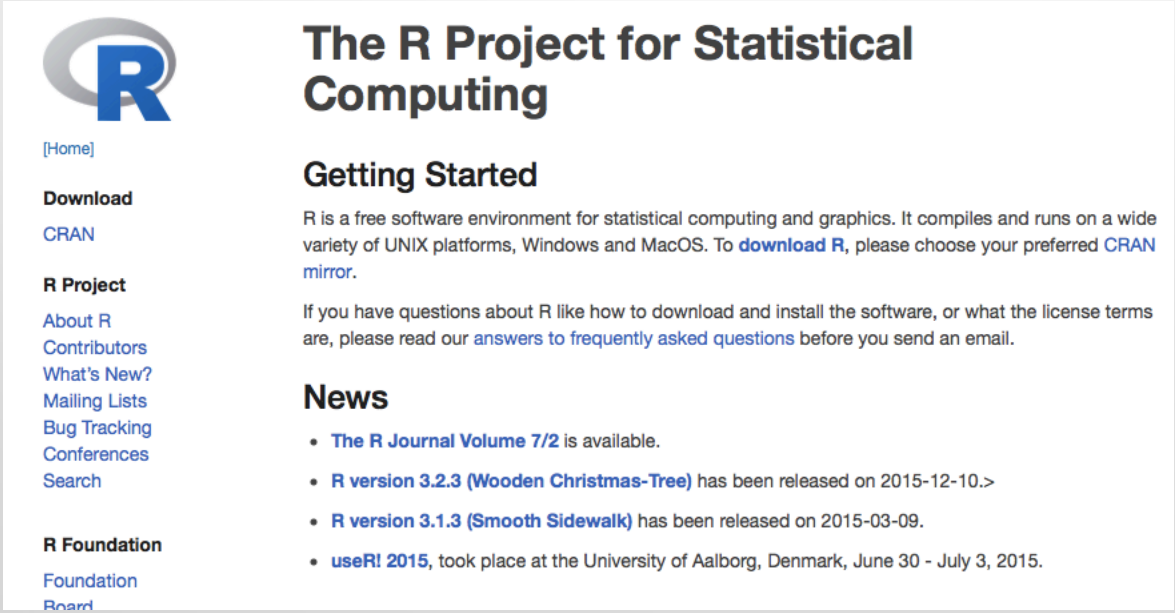


R examples

Data Visualization

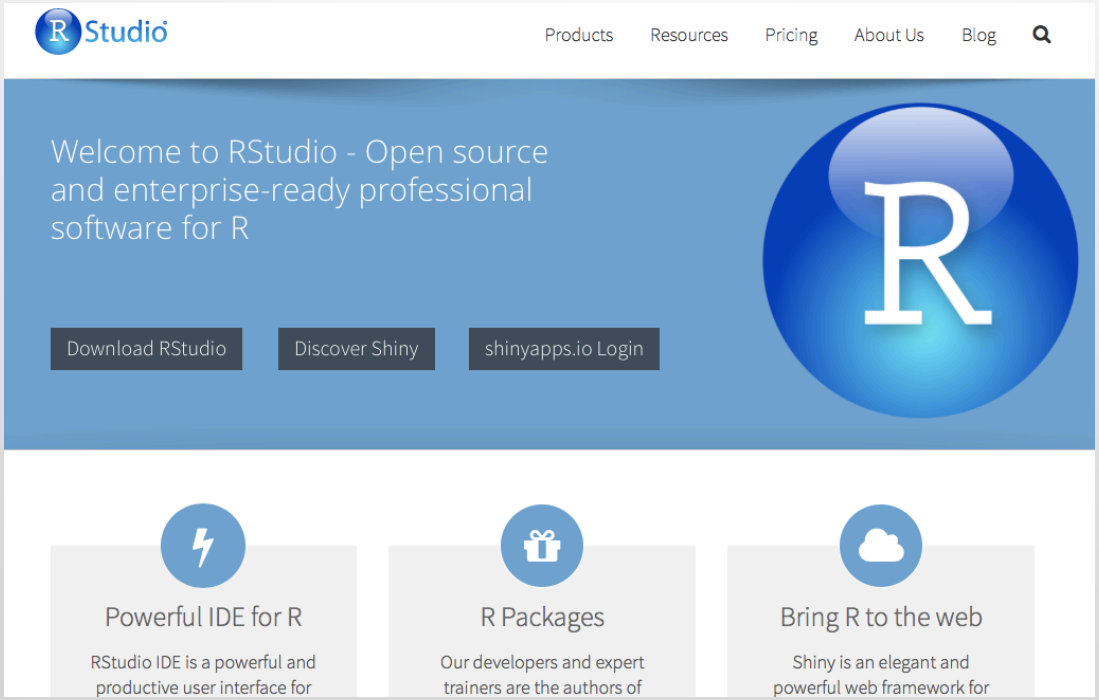
Installing R

- Website: https://www.r-project.org

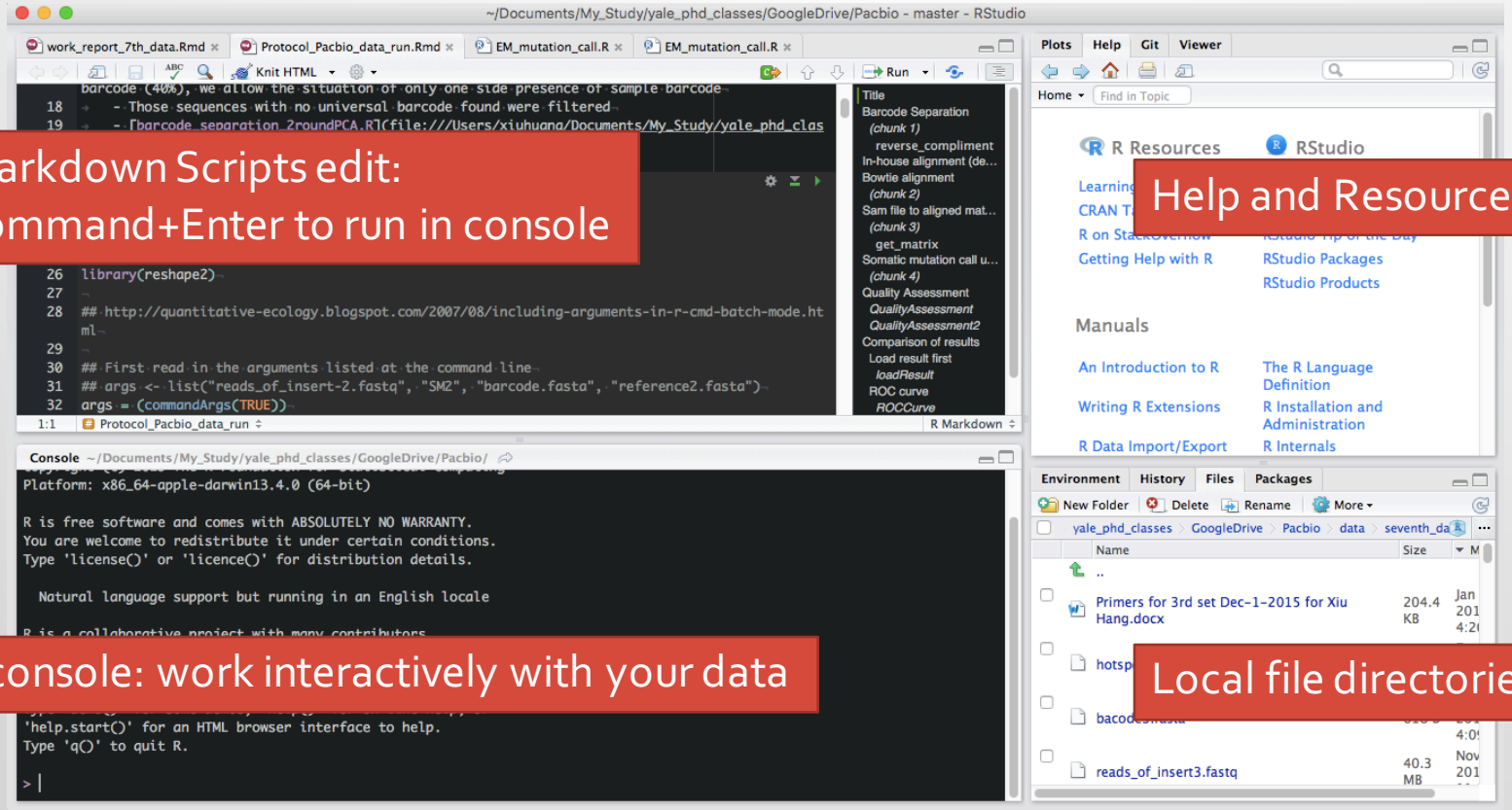


Installing RStudio

- Website: https://www.rstudio.com



(my) RStudio Interface



Basic Manipulation

- Create objects with <- or =;
- Remove objects by rm();
- Show current objects by ls();
- Names of objects and functions are case sensitive;
- Indices begin with 1, not 0;
- Comments: #;

- Math operators, vector, list, matrix, data frame, logical. (See code.)

Data Visualization

- Histogram;
  - hist()
- Scatter plot;
  - plot()
- Pairwise scatter plot;
  - pairs()
- Boxplot;
  - boxplot()
- Some other plots;

Export Graphs

- Export to PDF
  - pdf("iris.pdf")
- Plot multiple figures in one page;
  - par(mfrow=c(2,2))
- Shut down graph exporting;
  - dev.off()

Statistics in R

- Statistical tests (t test, chi square test);
  - t.test(iris[1:50,1], iris[51:100,1])
- Linear regression;
  - lm1 <- lm(Petal.Length~Petal.Width, data=iris)

Other Functions

- For loops;
- Conditional statements;
- Your own functions;

Input and Output Data from/to Files

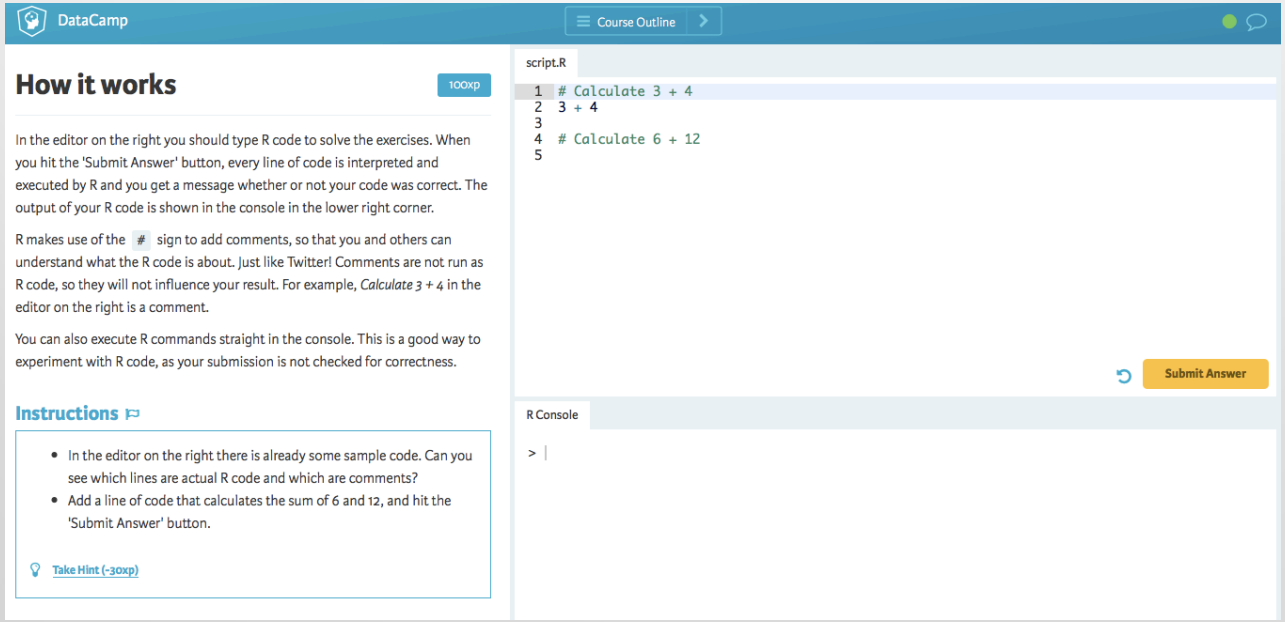
- Input data from files;
  - x <- read.table("iris.txt", header=T, as.is=T)
- Output data to files;
  - write.table(x, "iris2.txt", sep="\t", quote=F, col.names=T, row.names=F)

Getting Help

- help(function);
- ?function;
- help.search(function);
- Open source – type function name without ();

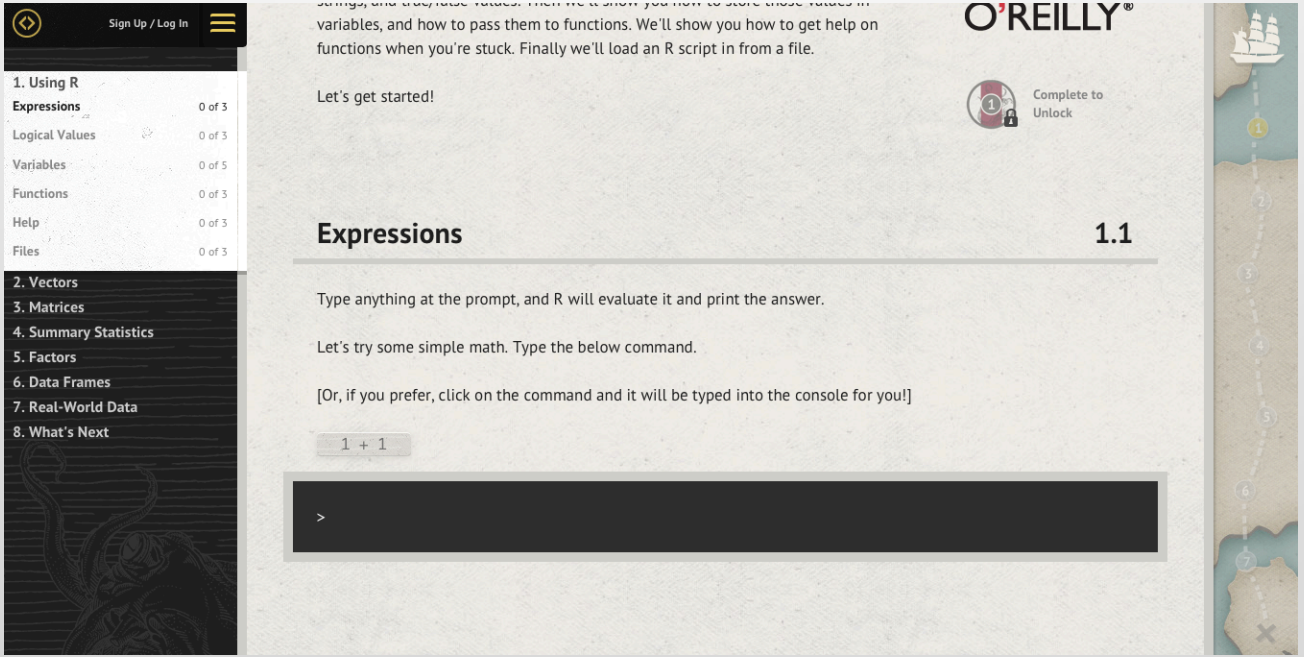
Resources to Learn More about R

- Great Resources to Learn Basics of R INTERACTIVELY:
  - DataCamp: https://www.datacamp.com/home



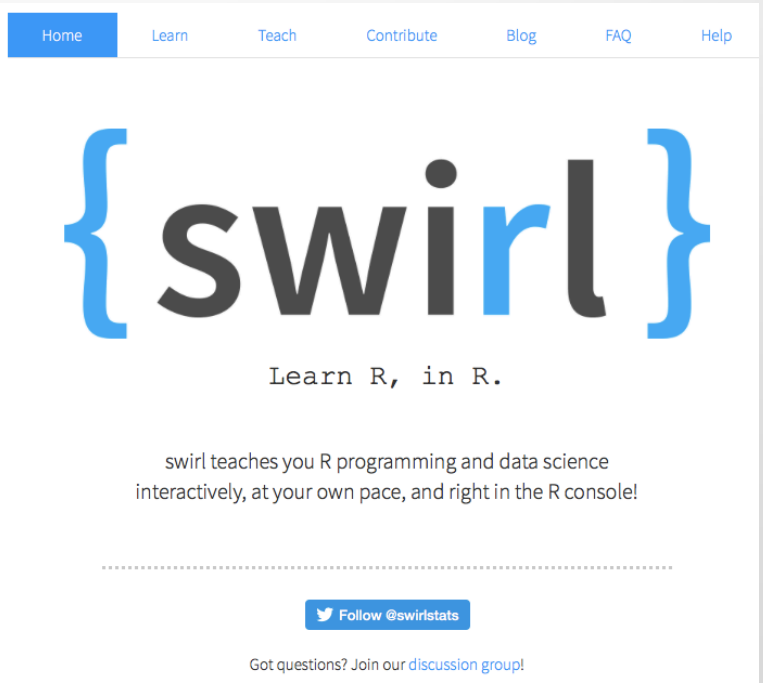
Resources to Learn More about R

- Great Resources to Learn Basics of R INTERACTIVELY :
  - TryR: http://tryr.codeschool.com



Resources to Learn More about R

- Great Resources to Learn Basics of R INTERACTIVELY :
  - Swirl: Learn R, in R!



Resources to Learn More about R

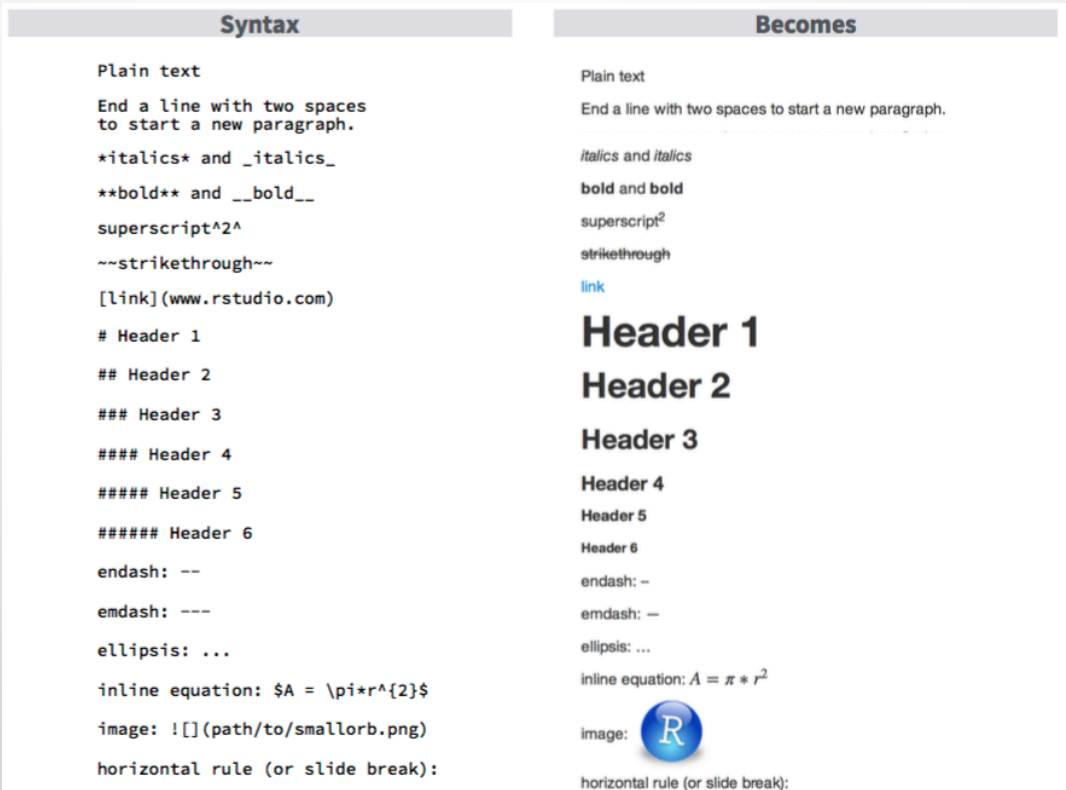
- More Resources to read about R:
  - The beginner's guide to R web posts: http://www.computerworld.com/article/2497143/business-intelligence/business-intelligence-beginner-s-guide-to-r-introduction.html

Reproducible Research in R

- The reason we choose RStudio is that it offers a convenient interface to carry out reproducible research with RMarkdown.
- Advantages of RMarkdown:
  - Easy syntax to generate reports that can easily transformed to Doc, Latex, PDF and all the other different formats;
  - Imbedded R code chunks with beautiful syntax highlight, running automatically, caching automatically, and generating results in a report kind of way. Neat!

Go through this 5-minute RMarkdown Guide and You are a Pro already

- https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf



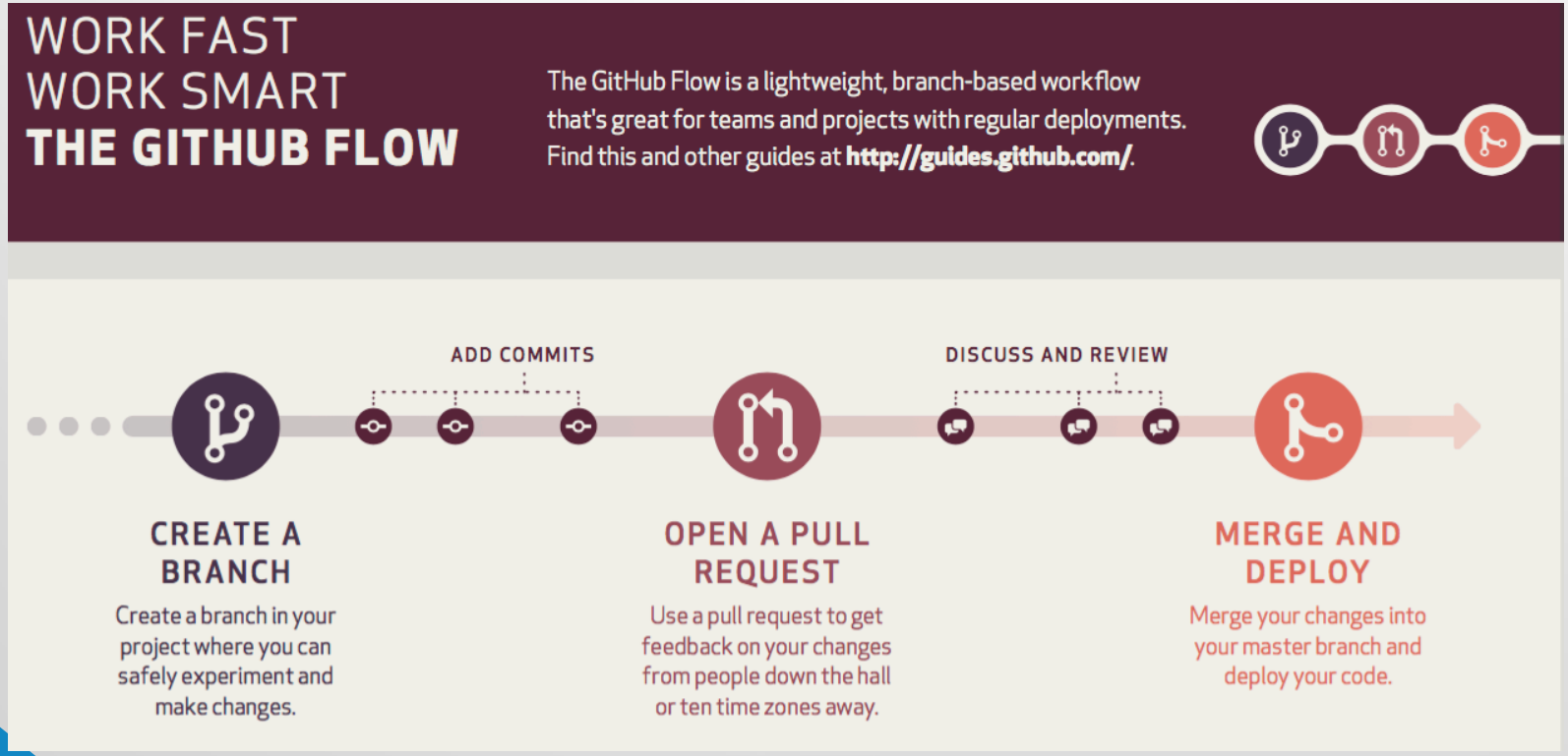
Syntax	Becomes
`Plain text`	Plain text
`End a line with two spaces` `to start a new paragraph.`	End a line with two spaces to start a new paragraph.
`*italics* and _italics_`	*italics* and *italics*
`**bold** and __bold__`	**bold** and **bold**
`superscript^2^`	superscript^2
`~~strikethrough~~`	~~strikethrough~~
`[link](www.rstudio.com)`	link
`# Header 1`	**Header 1**
`## Header 2`	**Header 2**
`### Header 3`	**Header 3**
`#### Header 4`	**Header 4**
`##### Header 5`	**Header 5**
`###### Header 6`	**Header 6**
`endash: --`	endash: –
`emdash: ---`	emdash: —
`ellipsis: ...`	ellipsis: …
`inline equation: $A = \pi*r^{2}$`	inline equation: $A = \pi * r^2$
`image: ![](path/to/smallorb.png)`	image:
`horizontal rule (or slide break):`	horizontal rule (or slide break):

Introduction to GitHub

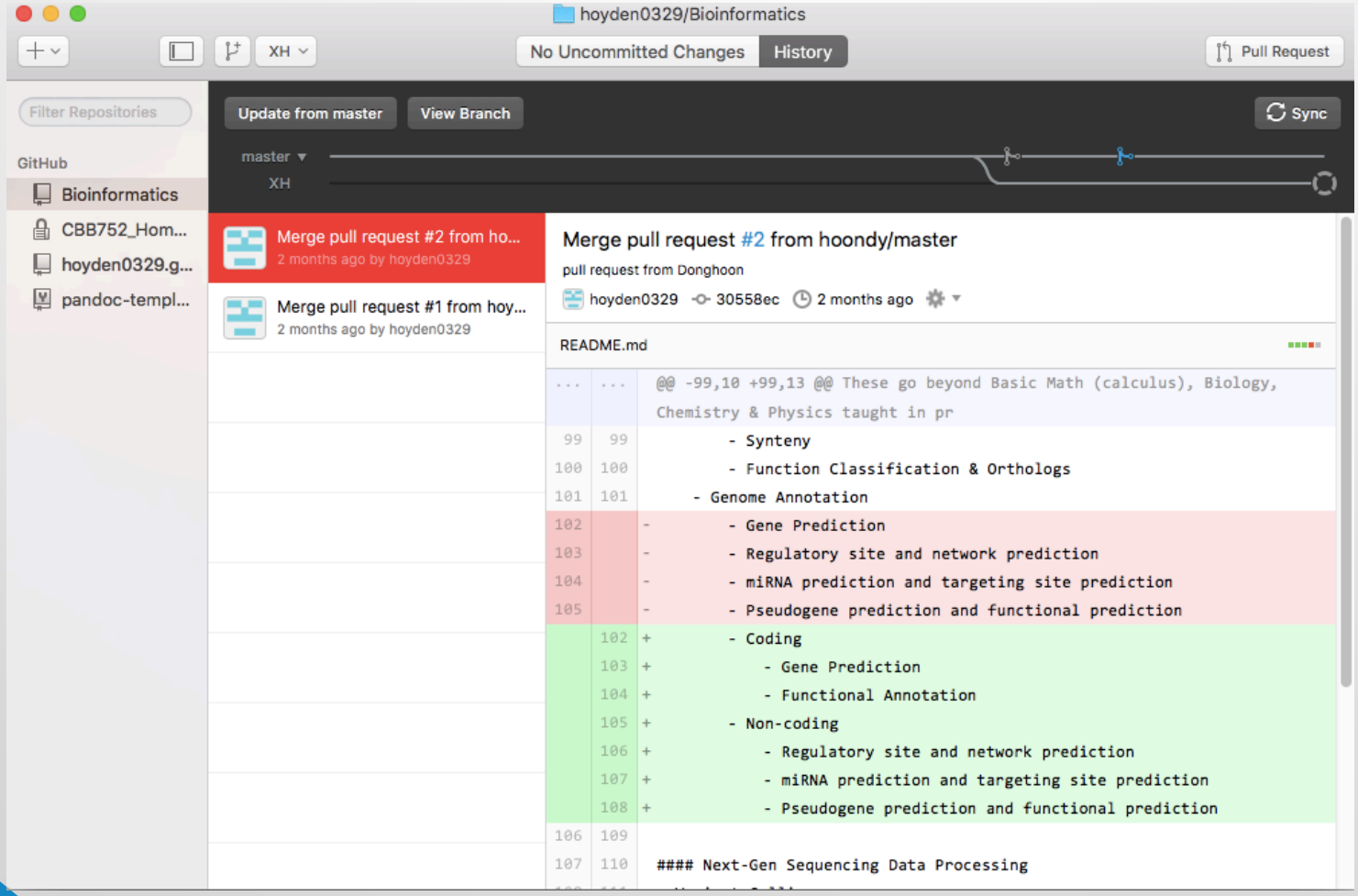
Why GitHub?

- Version control: keep track of your changes
- Easy interface: no command line needed at all!
- Easy collaboration: work together on the same project

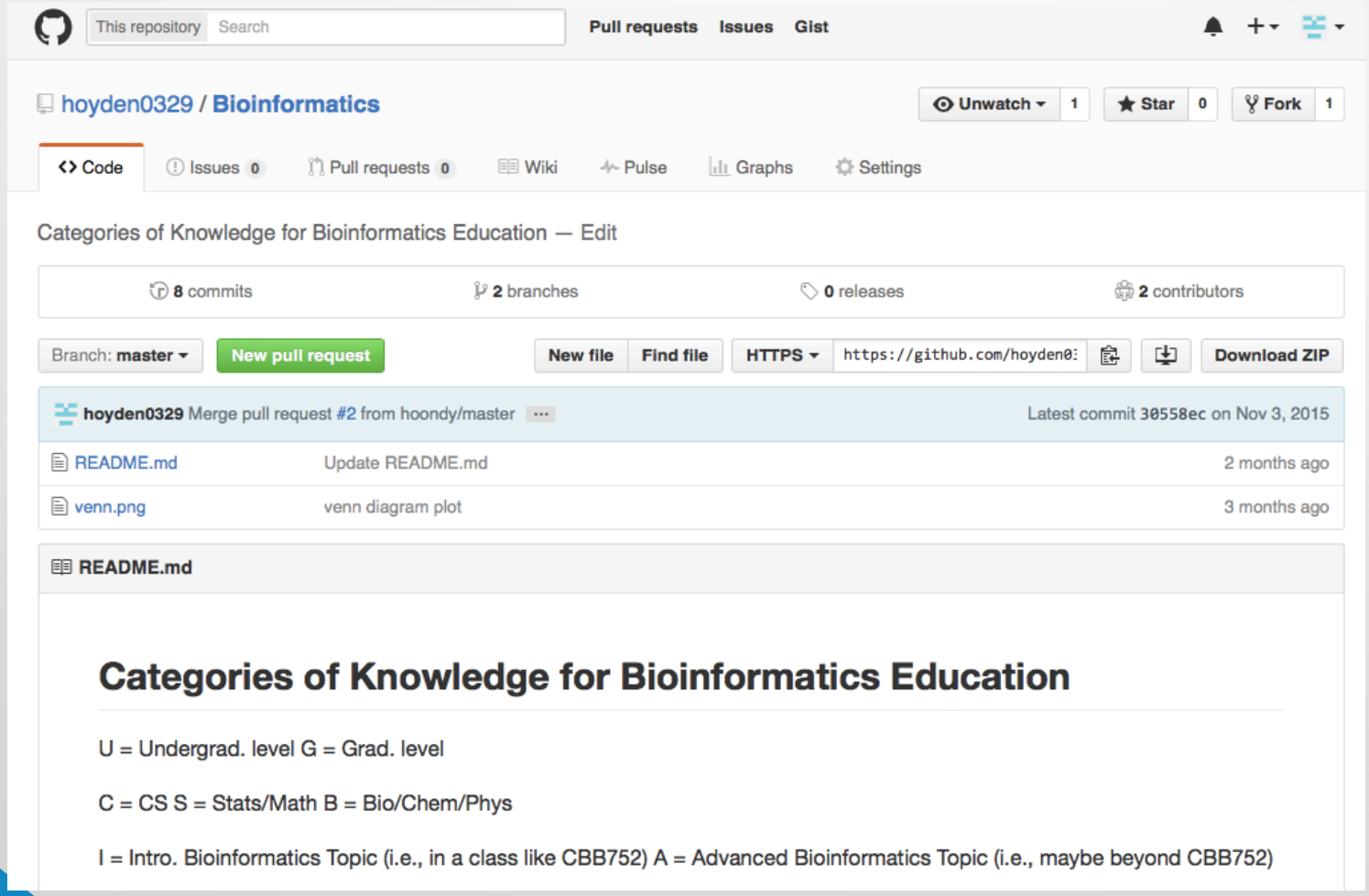
The GitHub Concept



GitHub App

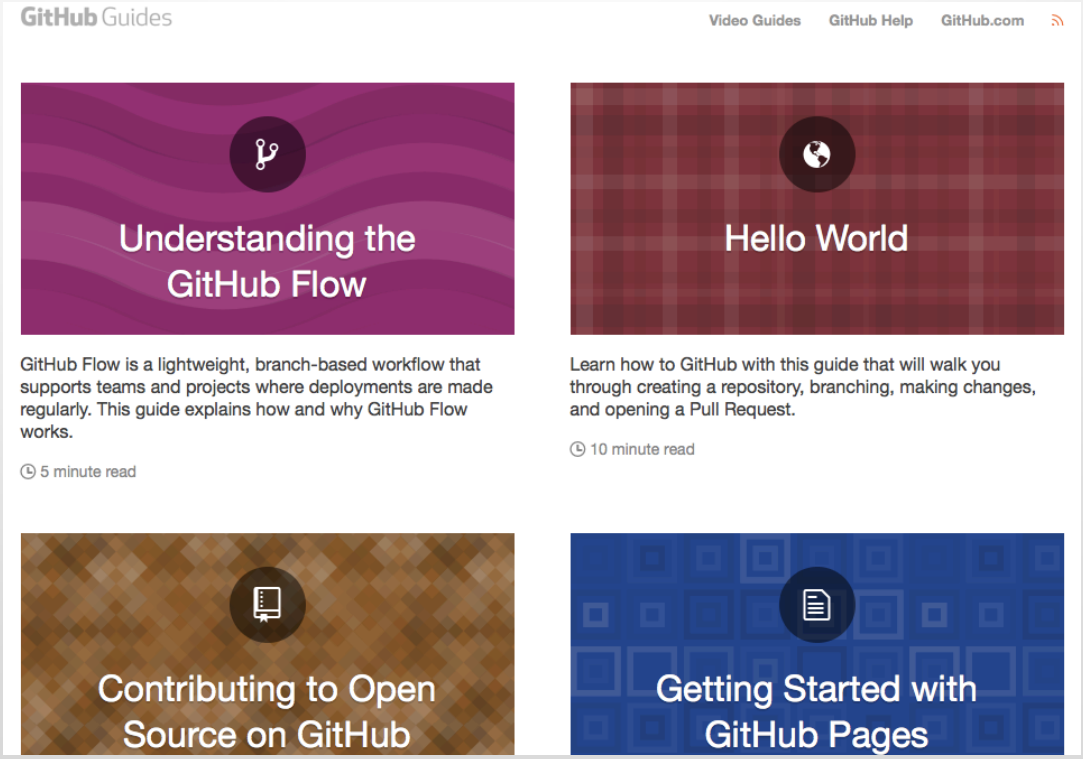

hoyden0329/Bioinformatics
XH
No Uncommitted Changes
History
Pull Request
Filter Repositories
GitHub
Bioinformatics
CBB752_Hom...
hoyden0329.g...
pandoc-templ...
Update from master
View Branch
Sync
master
XH
Merge pull request #2 from ho...
2 months ago by hoyden0329
Merge pull request #1 from hoy...
2 months ago by hoyden0329
Merge pull request #2 from hoondy/master
pull request from Donghoon
hoyden0329 30558ec 2 months ago
README.md
@@ -99,10 +99,13 @@ These go beyond Basic Math (calculus), Biology, Chemistry & Physics taught in pr
- Synteny
- Function Classification & Orthologs
- Genome Annotation
- Gene Prediction
- Regulatory site and network prediction
- miRNA prediction and targeting site prediction
- Pseudogene prediction and functional prediction
- Coding
- Gene Prediction
- Functional Annotation
- Non-coding
- Regulatory site and network prediction
- miRNA prediction and targeting site prediction
- Pseudogene prediction and functional prediction
Next-Gen Sequencing Data Processing

GitHub Web UI



More GitHub Learning Resources

- https://guides.github.com



Introduction to Python

What is Python?

- Interpreted (no need for compile)
- Human readable clear syntax (fewer lines of code than C++ or Java)
- Libraries like **NumPy**, **SciPy**, **BioPython** allow the effective use of Python in scientific computing

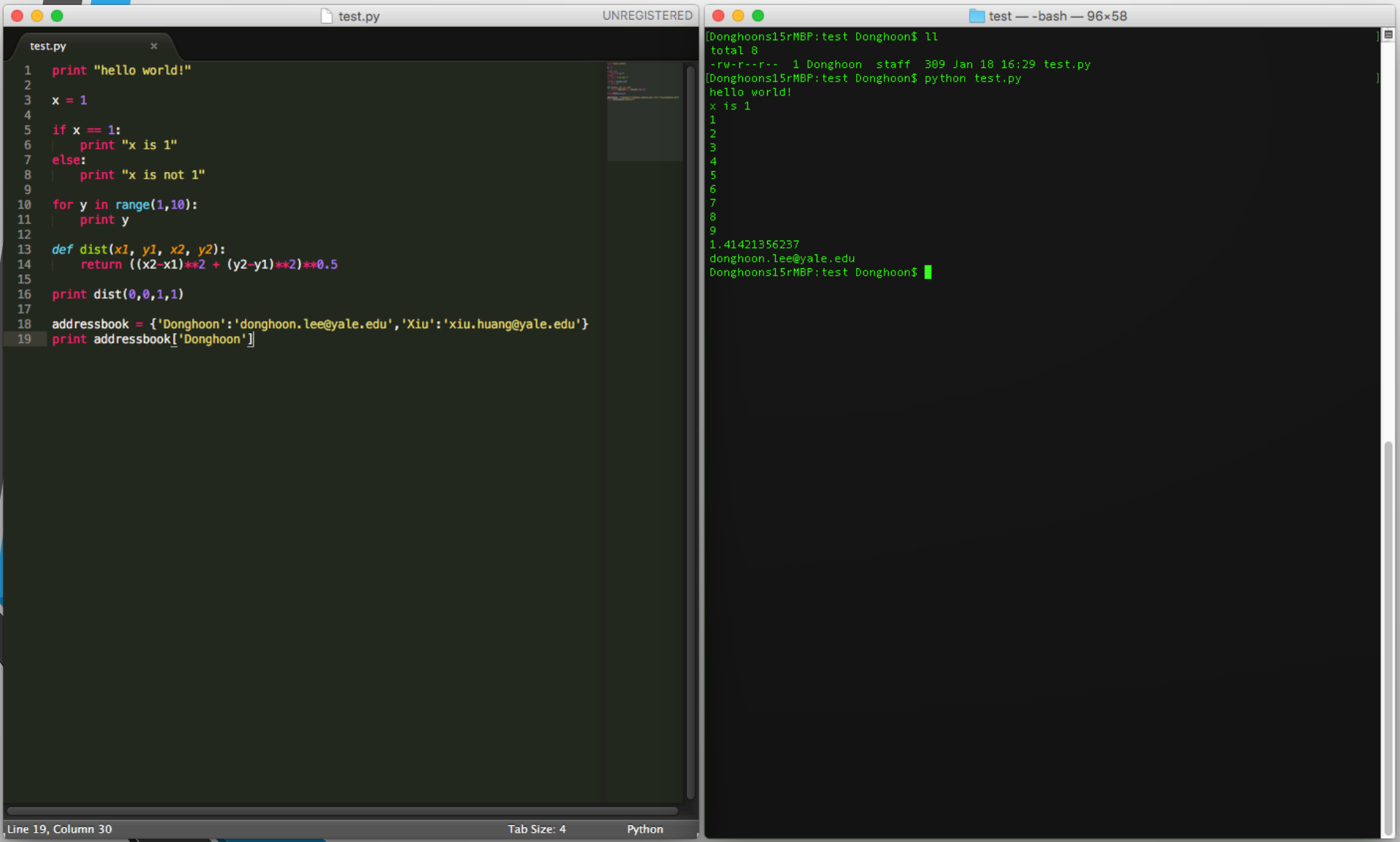
Installation

- We recommend Python 2.7 and up
- OS X (Mac), Linux and Unix: You already have one!
- Windows: https://www.python.org/downloads/windows/
- For more information: https://wiki.python.org/moin/BeginnersGuide/Download
- For writing python, you can use any text editor. I personally use Emacs.
- If you are looking for a Python IDE, there are IDLE, Eclipse (PyDev), PyCharm, etc

Python Basics

- Python uses white-space indentation rather than {}
- Use tab or spaces, but do not mix them
- No need for ; at the end of line
- In Python **==** compares by value
- Python uses ***and***, ***or***, ***not*** for its boolean operators rather than &&, ||, !
- Lists are mutable and written as [1, 2, 3]
- Tuples are immutable and written as (1, 2, 3)
- Use curly braces for dictionary (key-value pair, not ordered) e.g., {'key1': 1.0, 'key2': False}

Example


test.py
UNREGISTERED
test.py
1 print "hello world!"
2
3 x = 1
4
5 if x == 1:
6 print "x is 1"
7 else:
8 print "x is not 1"
9
10 for y in range(1,10):
11 print y
12
13 def dist(x1, y1, x2, y2):
14 return ((x2-x1)**2 + (y2-y1)**2)**0.5
15
16 print dist(0,0,1,1)
17
18 addressbook = {'Donghoon':'donghoon.lee@yale.edu','Xiu':'xiu.huang@yale.edu'}
19 print addressbook['Donghoon']
Line 19, Column 30
Tab Size: 4
Python
test — -bash — 96×58
[Donghoons15rMBP:test Donghoon$ ll
total 8
-rw-r--r-- 1 Donghoon staff 309 Jan 18 16:29 test.py
[Donghoons15rMBP:test Donghoon$ python test.py
hello world!
x is 1
1
2
3
4
5
6
7
8
9
1.41421356237
donghoon.lee@yale.edu
Donghoons15rMBP:test Donghoon$

Resources

- Python Cookbook, 3rd Edition: Recipes for Mastering Python 3 by David Beazley, Brian K. Jones (Publisher: O'Reilly Media). You can view Online copy at Yale library and this is Github source code from book.
- 13 hr self-taught crash course on python, https://www.codecademy.com/learn/python
- And of course, Google and Stack Overflow are your best programming friend.