

# Welcome to Quantitative Biology

---



QB Bootcamp, Day 1  
Wednesday, 28 August 2019  
10:00am - 10:30am

## QB Bootcamp team

---



**Justin B. Kinney**  
(Assistant Professor)



**Wei-Chia Chen**  
Postdoc, Kinney Lab



**Ben Harris**  
WSBS, Gillis Lab



**Hussein Hijazi**  
Postdoc, Siepel Lab



**Shaina Lu**  
WSBS, Gillis /  
Zador Labs

# 2019 QB Bootcamp Schedule

---

- **Wednesday, August 28, 10a-5p**
  - 10:00a - 10:45a: Welcome to quantitative biology (**Justin**)
  - 10:45a - 12:00p: The UNIX command line (**Justin**)
  - *12:00p - 1:00p: Lunch*
  - 1:00p - 1:30p: Introduction to Python and Jupyter Notebooks (**Justin**)
  - 1:30p - 3:00p: Python: data types (**Ben**)
  - *3:00p - 3:30p: Break*
  - 3:30p - 5:00p: Python: flow control (**Ben**)
- **Thursday, August 29, 10a-5p**
  - 10:00a - 10:30a: Overview of high-performance computing (**Justin**)
  - 10:30a - 12:00p: BlackNBlue, read mapping (**Justin**)
  - *12:00p - 1:00p: Lunch*
  - 1:00p - 1:30p: Introduction to dataframes (**Justin**)
  - 1:30p - 3:00p: Pandas I, binding site analysis (**Hussein**)
  - *3:00p - 3:30p: Break*
  - 3:30p - 5:00p: Pandas II, replication origin analysis (**Hussein**)
- **Friday, August 30, 2p-6p**
  - 2:00p - 2:30p: Overview of data visualization (**Justin**)
  - 2:30p - 4:00p: Matplotlib (**Shaina**)
  - *4:00p - 4:30p: Break*
  - 4:30p - 6:00p: Seaborn (**Shaina**)

[https://github.com/jbkinney/19\\_qbbootcamp](https://github.com/jbkinney/19_qbbootcamp)

Download this repository

The screenshot shows a GitHub repository page for 'jbkinney / 19\_qbbootcamp'. The page includes a header with navigation links like Pull requests, Issues, Marketplace, and Explore. Below the header, there's a summary bar with metrics: 12 commits, 1 branch, 0 releases, and 2 contributors. A green oval highlights the 'Clone or download' dropdown menu, which contains options for 'Clone with HTTPS' and 'Use SSH', along with a copy icon and two buttons: 'Open in Desktop' and 'Download ZIP'. The main content area displays a list of files and their commit history, including 'bash', 'cheatsheets', 'python', 'README.md', 'bnb\_exercise.tar.gz', 'mac\_install.sh', and 'qb\_syllabus.pdf'. At the bottom, there's a section titled '2019 Quantitative Biology Bootcamp' with a welcome message and a 'Summary' link.

jbkinney/19\_qbbootcamp: GitHub repository for the 2019 QB Bootcamp

Manage topics

Branch: master ▾ New pull request

12 commits 1 branch 0 releases 2 contributors

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

[https://github.com/jbkinney/19\\_qbbootcamp](https://github.com/jbkinney/19_qbbootcamp)

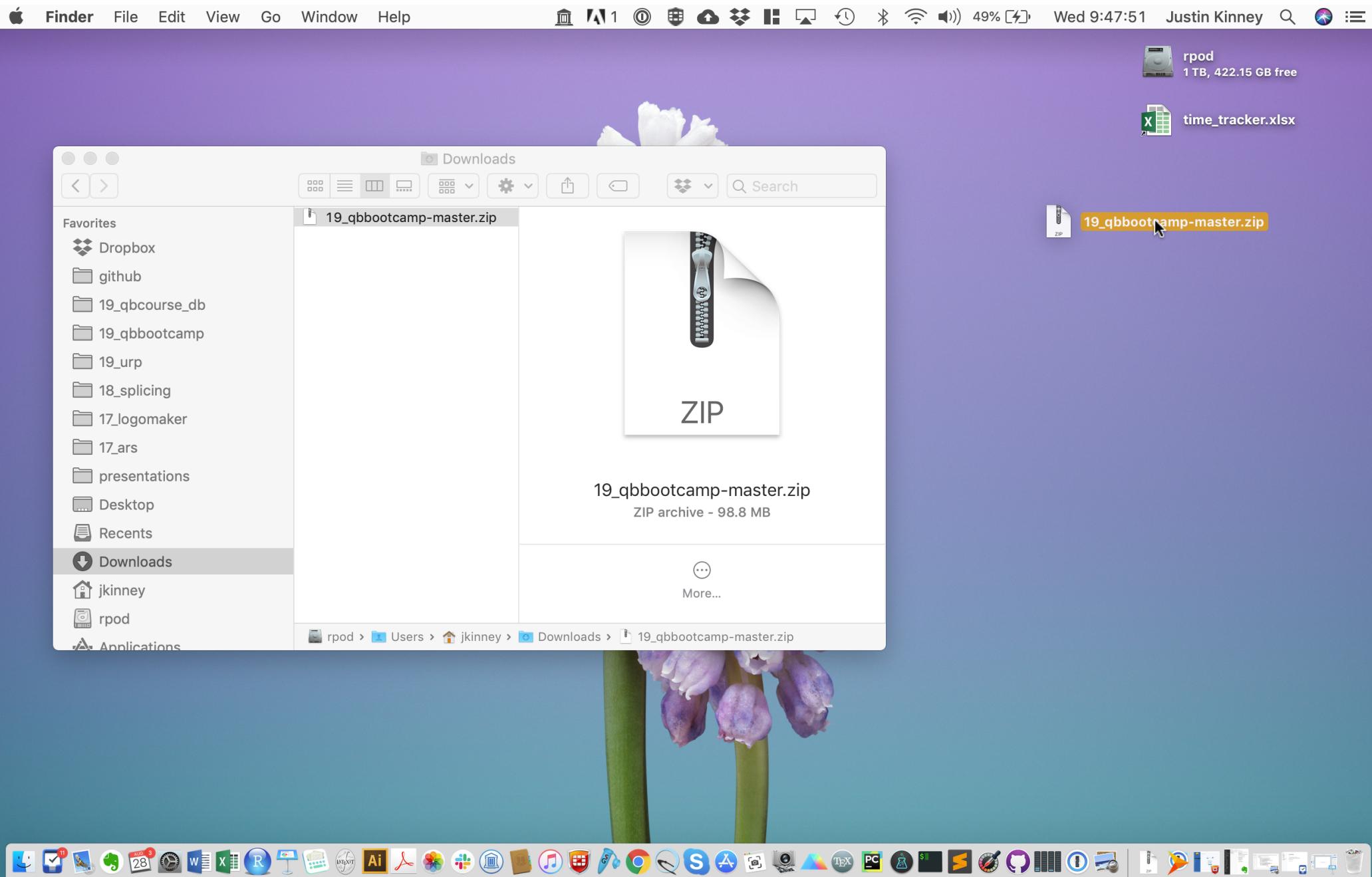
Open in Desktop Download ZIP

**2019 Quantitative Biology Bootcamp**

Welcome to the 2019 QB Bootcamp in the Watson School for Biological Sciences at Cold Spring Harbor Laboratory! This GitHub repository contains the Jupyter notebooks, shell scripts, and data sets that we will work through in this bootcamp.

Summary

# Move this repository from Downloads to your Desktop and decompress by double-clicking

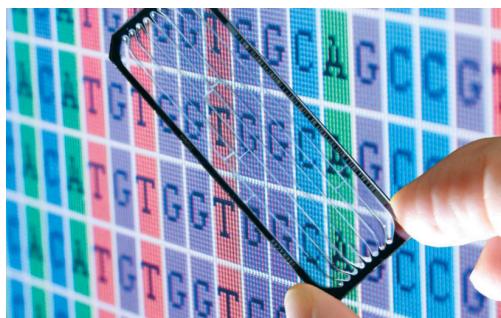


## **What is Quantitative Biology?**

# Quantitative biology is a vast field

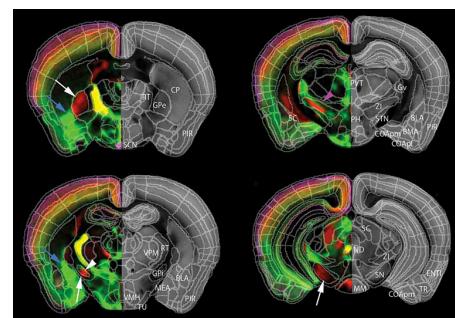
---

## Genomics



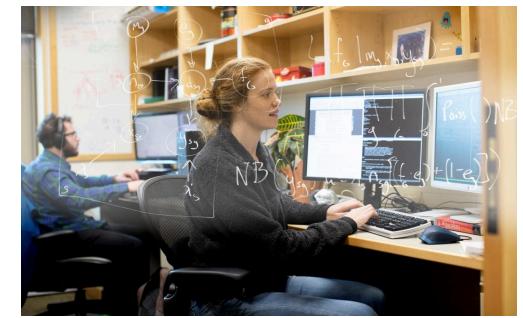
Functional genomics  
Evolutionary genomics  
Genome dynamics  
Technology development

## Neuroscience



Data analysis  
Modeling neural systems  
Behavioral modeling

## Other



Biophysics  
Machine learning  
Software development

**Who does Quantitative Biology at CSHL?**

## Core QB program

---



**Molly Gale  
Hammell**



**Dan  
Levy**



**Saket  
Navlakah**



**Ivan  
Iossofov**



**David  
McCandlish**



**Justin  
Kinney**



**Hannah  
Meyer**



**Peter  
Koo**



**Alexander  
Krasnitz**



**Adam  
Siepel**

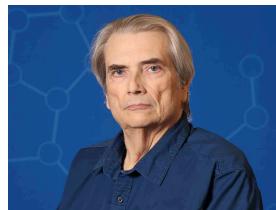
## QB Associated Faculty

---

### Genomics



Alexander  
Dobin



Richard  
McCombie

### Neuroscience



Tatiana  
Engel



Jesse  
Gillis



Doreen  
Ware



Alexei  
Koulakov



Partha  
Mitra

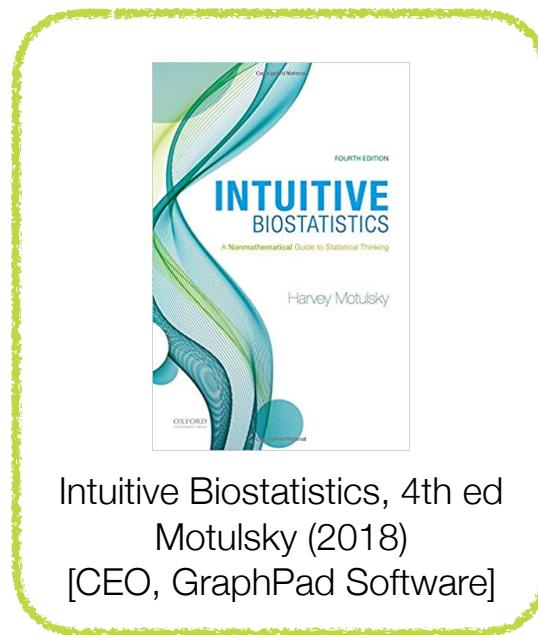
**What QB skills should all biology researchers have?**

## Learn to interpret standard statistics

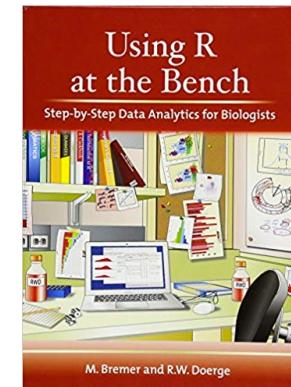
---

### Key statistical concepts:

- P-values
- Multiple hypothesis testing
- Confidence intervals
- Regression
- ANOVA
- Survival analysis

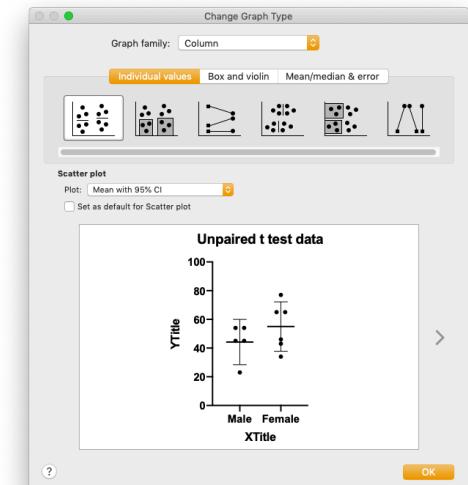
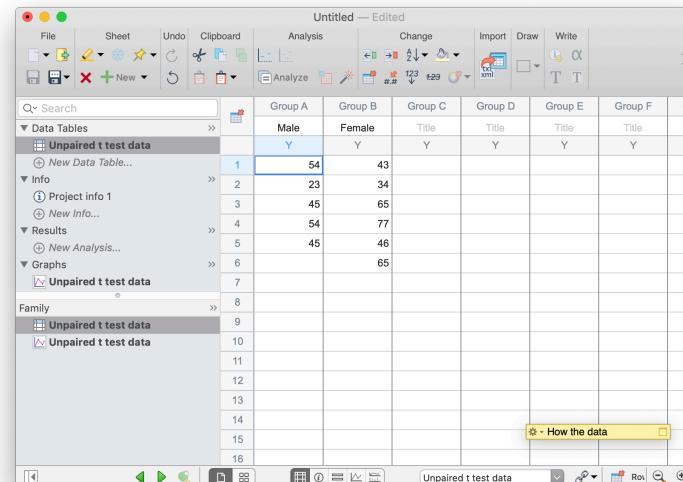
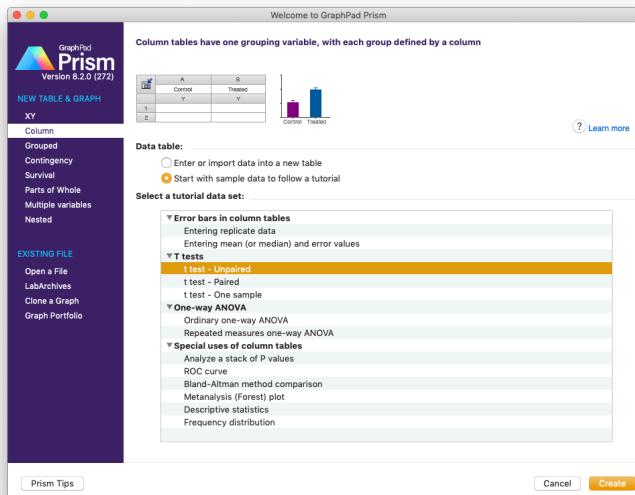


Intuitive Biostatistics, 4th ed  
Motulsky (2018)  
[CEO, GraphPad Software]



Using R at the Bench  
Bremmer & Doerge (2015)

# Learn to compute standard statistics



Alternatively:



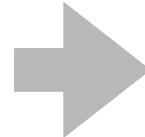
# Learn to navigate UNIX systems



Sequencer



Microscope



High Performance  
Computer Cluster

A screenshot of a UNIX command line terminal window showing a file listing. The terminal title is "jkinney — ssh bnbdev2 — 80x24". The command "ls" was run, displaying files like 15\_splicing, 18\_splicing, 19\_mrna, 19\_wpx, 18\_splicing\_local, 18\_splicing\_3ss, 19\_softy, 17\_arcs, 18\_splicing\_sim2, 19\_exercise.tor, 17\_arcs\_chip, 18\_splicing\_twistamp, big\_data, and old\_filesys.

```
jkinney@bnbdev2:~$ ls
15_splicing      18_splicing      19_mrna      bin
15_splicing_local 18_splicing_3ss  19_wpx       bnb_exercise.tor
17_arcs          18_splicing_sim2 19_softy     freezer
17_arcs_chip    18_splicing_twistamp big_data   old_filesys
jkinney@bnbdev2:~$
```

UNIX command line



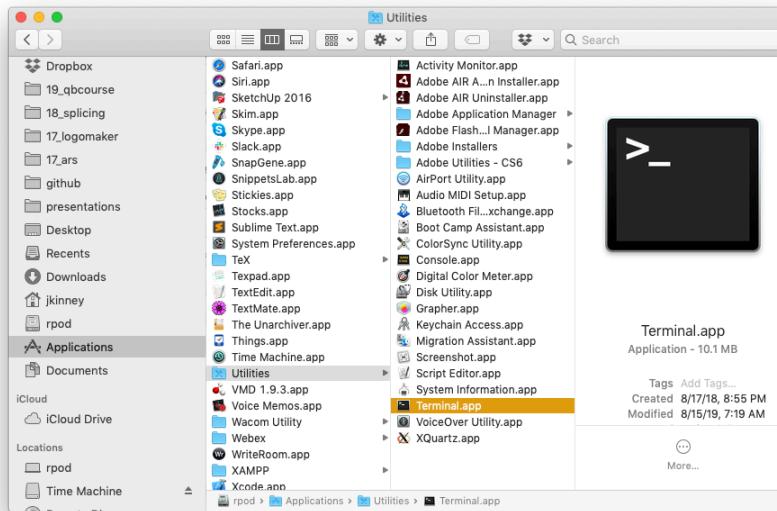
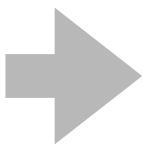
Databases



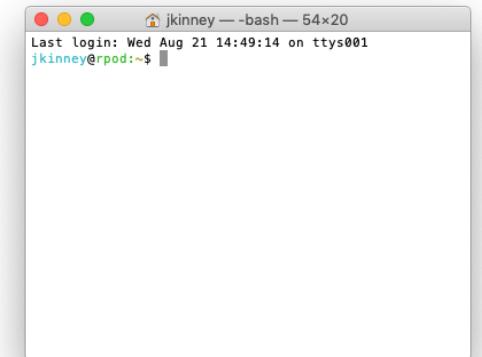
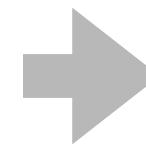
# Mac OS X is based on UNIX



Finder

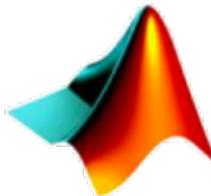


Applications/Utilities/Terminal.app

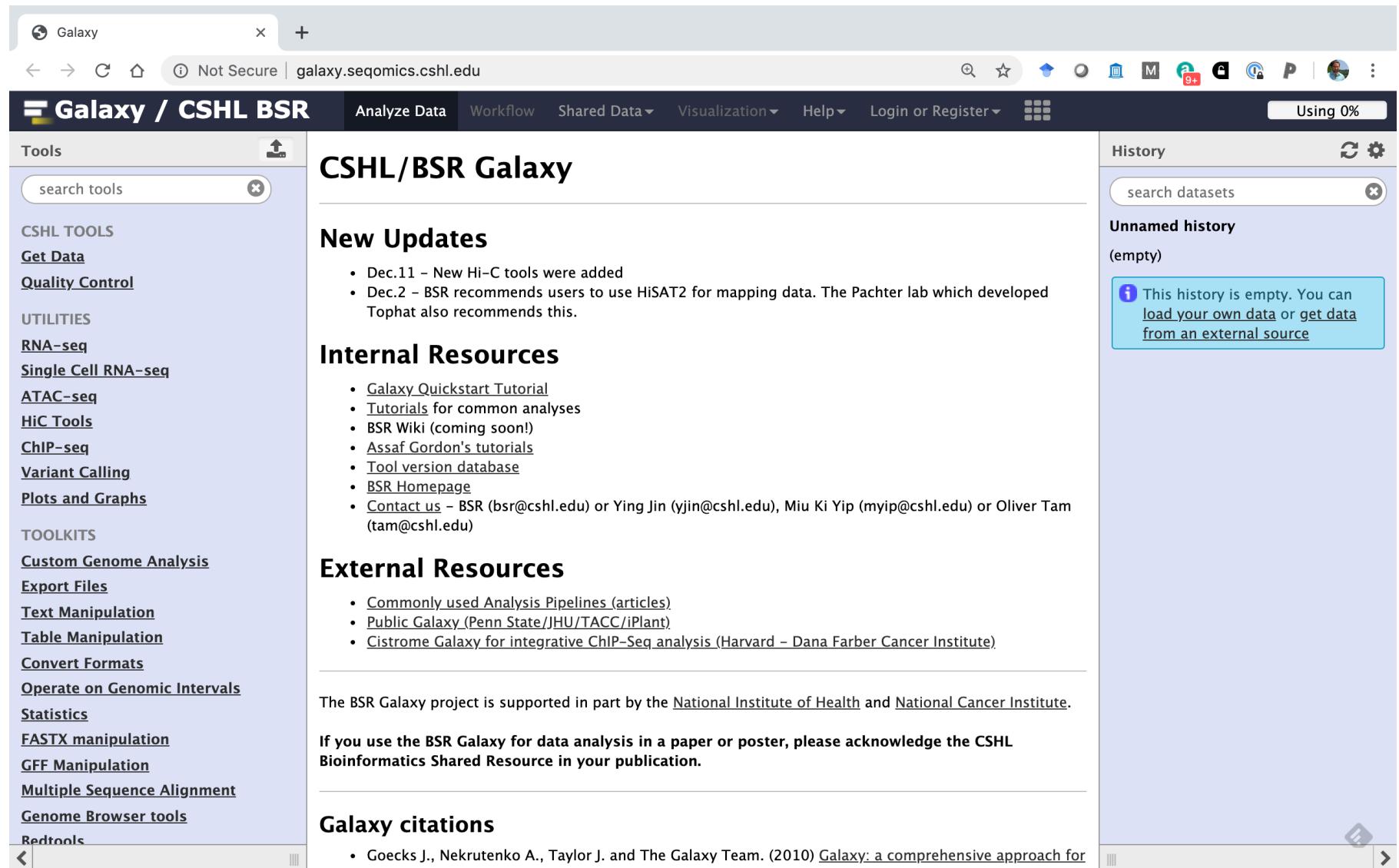


## Become familiar with at least one programming language

---

language	strengths	weaknesses
 python™	<ul style="list-style-type: none"><li>- elegant language</li><li>- easy to learn</li><li>- flexibility: use for large pipelines or local data analysis</li><li>- highly valued skill in industry</li></ul>	<ul style="list-style-type: none"><li>- clunky dataframes</li><li>- clunky statistics</li><li>- clunky graphics</li></ul>
	<ul style="list-style-type: none"><li>- streamlined for statistics</li><li>- highly developed for genomics</li><li>- great graphics</li></ul>	<ul style="list-style-type: none"><li>- strange language</li><li>- not great for building pipelines</li></ul>
 MATLAB	<ul style="list-style-type: none"><li>- used heavily in neuroscience and by old people</li></ul>	<ul style="list-style-type: none"><li>- proprietary</li><li>- poorly supported</li><li>- bad graphics</li><li>- bad for strings</li></ul>

# Learn to analyze your own sequencing data



The screenshot shows the CSHL/BSR Galaxy homepage. The top navigation bar includes links for Analyze Data, Workflow, Shared Data, Visualization, Help, Login or Register, and a search bar. The main content area features a "New Updates" section with two bullet points: "Dec.11 – New Hi-C tools were added" and "Dec.2 – BSR recommends users to use HiSAT2 for mapping data. The Pachter lab which developed Tophat also recommends this." Below this is an "Internal Resources" section with links to Galaxy Quickstart Tutorial, Tutorials for common analyses, BSR Wiki (coming soon!), Assaf Gordon's tutorials, Tool version database, BSR Homepage, and Contact us. The "External Resources" section lists Commonly used Analysis Pipelines (articles), Public Galaxy (Penn State/JHU/TACC/iPlant), and Cistrome Galaxy for integrative ChIP-Seq analysis (Harvard – Dana Farber Cancer Institute). A note at the bottom states that the BSR Galaxy project is supported by the National Institute of Health and National Cancer Institute, and encourages users to acknowledge the CSHL Bioinformatics Shared Resource in their publications. The left sidebar contains a "Tools" section with categories like CSHL TOOLS (Get Data, Quality Control), UTILITIES (RNA-seq, Single Cell RNA-seq, ATAC-seq, HiC Tools, ChIP-seq, Variant Calling, Plots and Graphs), and TOOLKITS (Custom Genome Analysis, Export Files, Text Manipulation, Table Manipulation, Convert Formats, Operate on Genomic Intervals, Statistics, FASTX manipulation, GFF Manipulation, Multiple Sequence Alignment, Genome Browser tools, Bedtools).

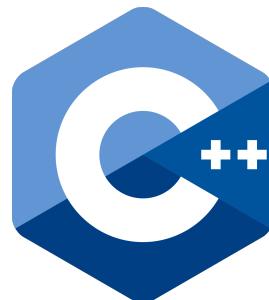
Don't be shy about asking QB labs to help you learn.

**What skills do you need to do research in Quantitative Biology?**

## Learn to program well

---

Tip: it is better to know one language well than many languages superficially.



# How to learn to program

---



## BEST ONLINE COURSES FOR PYTHON AT A GLANCE

Our picks for the best subscription / fee-based Python courses and tutorials

- 1. Ask for guidance**
- 2. Work on projects that require it**
- 3. Google your questions & read help threads**
- 4. Read package documentation**
- 5. Read select books**
- 6. Take online courses (don't worry about cost)**

- [Python For Everybody](#) [[coursera.com](https://www.coursera.com)]
- [Learning Python with PyCharm](#) [[lynda.com](https://www.lynda.com)]
- [DataCamp](#) [[datacamp.com](https://www.datacamp.com)]
- [Introduction to Python: Absolute Beginner](#) [[edx.com](https://www.edx.com)]
- [Introduction to Computer Science and Programming Using Python](#) [[edx.com](https://www.edx.com)]
- [Python and Django Full Stack Web Developer Bootcamp](#) [[udemy.com](https://www.udemy.com)]
- [AI Programming with Python](#) [[udacity.com](https://www.udacity.com)]
- [Introduction to Computing in Python](#) [[edx.com](https://www.edx.com)]
- [Python I: Essentials](#) [[quickstart.com](https://www.quickstart.com)]

# Learn to use LaTeX

The screenshot shows a LaTeX editor interface with the file `19_mclb.tex` open. The code is a LaTeX document with sections for packages, title, author information, and abstract. The right side shows the rendered PDF output.

**Code (19\_mclb.tex):**

```
22 \usepackage[utf8]{inputenc} % allow utf-8 input
23 \usepackage[T1]{fontenc} % use 8-bit T1 fonts
24 \usepackage{hyperref} % hyperlinks
25 \usepackage{url} % simple URL typesetting
26 \usepackage{booktabs} % professional-quality tables
27 \usepackage{amsfonts} % blackboard math symbols
28 \usepackage{nicefrac} % compact symbols for 1/2, etc.
29 \usepackage{microtype} % microtypography
30 \usepackage{soul} % for \ul
31 \usepackage{graphicx} % for figures
32 \usepackage{upgreek}
33
34 \title{Biophysical models of cis-regulation as\\ interpretable neural networks}
35
36
37 \author{%
38   Ammar Tareen \\
39   Simons Center for Quantitative Biology\\
40   Cold Spring Harbor Laboratory\\
41   Cold Spring Harbor, NY 11724 \\
42   \texttt{tareen@cshl.edu} \\
43   And \\
44   Justin B. Kinney \\
45   Simons Center for Quantitative Biology\\
46   Cold Spring Harbor Laboratory\\
47   Cold Spring Harbor, NY 11724 \\
48   \texttt{jkinney@cshl.edu} \\
49 }
50
51 \begin{document}
52
53 \maketitle
54
55 \begin{abstract}
56 Biophysical models that describe gene regulation, as well as other cis-regulatory processes, can be
      formulated as deep neural networks. This is true of quasi-equilibrium (a.k.a.\ thermodynamic)
      models as well as non-equilibrium (a.k.a.\ kinetic) models. This observation suggests new ways of
      using powerful deep learning frameworks for training biophysically interpretable neural networks
      using data produced by massively parallel reporter assays (MPRAs). We demonstrate this

```

**PDF Preview:**

**Biophysical models of cis-regulation as  
interpretable neural networks**

---

Ammar Tareen  
Simons Center for Quantitative Biology  
Cold Spring Harbor Laboratory  
Cold Spring Harbor, NY 11724  
[tareen@cshl.edu](mailto:tareen@cshl.edu)

Justin B. Kinney  
Simons Center for Quantitative Biology  
Cold Spring Harbor Laboratory  
Cold Spring Harbor, NY 11724  
[jkinney@cshl.edu](mailto:jkinney@cshl.edu)

**Abstract**

Biophysical models that describe gene regulation, as well as other cis-regulatory processes, can be formulated as deep neural networks. This is true of quasi-equilibrium (a.k.a. thermodynamic) models as well as non-equilibrium (a.k.a. kinetic) models. This observation suggests new ways of using powerful deep learning frameworks for training biophysically interpretable neural networks using data produced by massively parallel reporter assays (MPRAs). We demonstrate this capability using previously published MPRA data, and find that using deep learning frameworks to infer such biophysical models yields a dramatic improvement over previously reported model inference approaches.

**1 Introduction**

There are three main types of biophysical models for cis-regulation: thermodynamic, kinetic, and stochastic. Here we focus on the first two kinds of models, both of which can be inferred (at least in principle) from the type of time-averaged data that MPRA produce. Thermodynamic models are currently the standard way to biophysically model gene regulation [1–6]. These models assume that cis-regulatory complexes form as they would in thermodynamic equilibrium, and that this equilibrium is not greatly disturbed by the downstream kinetic processes that they regulate. By contrast, kinetic models assume that a cis-regulatory system is in steady state, but not necessarily thermal equilibrium. Thermodynamic models have proven remarkably successful at explaining the quantitative activity of a small number of bacterial promoters [7–9]. They have also been applied to a variety of regulatory contexts in yeast [10] and metazoans [11, 12]. Kinetic models have been applied less extensively, but there is a great deal of interest in them due to their ability to perform computations that thermodynamic models cannot [13–15]. However, confidently constructing either type of biophysical model for real biological systems remains a major challenge. A major stumbling block is the lack of available software. Although it was shown early on that biophysical models could be inferred from MPRA data [16], no general-purpose software for performing this type of MPRA data analysis has been described.

**2 Thermodynamic models as deep neural networks**

Thermodynamic models are specified by a set of molecular complexes, or “states”, which we index using  $s$ . Each state has both a Gibbs free energy  $\Delta G_s$  and an associated activity  $\alpha_s$ . These energies determine the probability  $P_s$  of each state occurring in thermodynamic equilibrium via the Boltzmann distribution,<sup>1</sup>

$$P_s = \frac{e^{-\Delta G_s}}{\sum_{s'} e^{-\Delta G_{s'}}}. \quad (1)$$

<sup>1</sup>To reduce notational burden, all  $\Delta G$  values are assumed to be in thermal units. At 37°C, one thermal unit is  $1 k_B T = 0.62 \text{ kcal/mol}$ , where  $k_B$  is Boltzmann’s constant and  $T$  is temperature.

## Develop core quantitative knowledge

---

### **Fundamentals**

Calculus  
Linear Algebra  
Algorithms (basic)  
Statistics (basic)

### **Intermediate material**

Bayesian inference  
Machine learning  
Sequence analysis  
Population genetics  
Theoretical neuroscience  
Algorithms (intermediate)

### **Advanced material**

Molecular biophysics  
Stochastic processes  
Dynamical systems  
Information theory  
Deep learning  
...

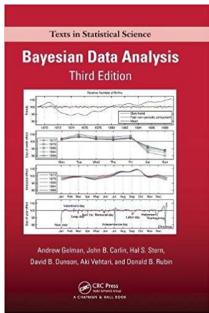
**Master all of  
these topics**

**Master at least one  
of these topics**

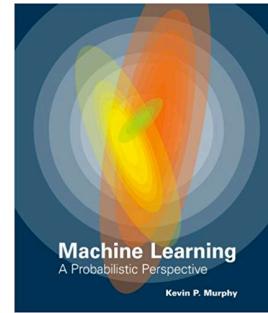
**Learn selected  
topics as needed**

## Learn to work through technical books systematically and independently

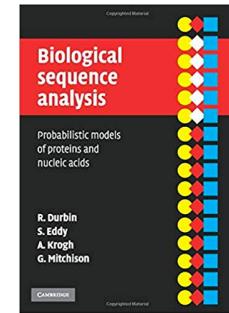
**New QB initiative:** faculty will help interested students pursue directed independent study of graduate-level material.  
Email me <[jkinney@cshl.edu](mailto:jkinney@cshl.edu)> if interested.



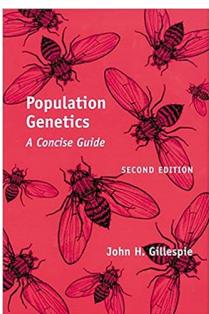
**Bayesian Data Analysis, 3rd ed**  
Gelman et al., 2013



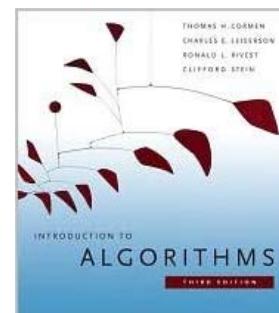
**Machine Learning:  
A Probabilistic Perspective**  
Murphy, 2012



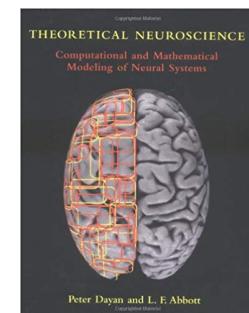
**Biological Sequence Analysis**  
Durbin et al., 1998



**Population Genetics:  
A concise guide, 2nd ed**  
Gillespie, 2004



**Introduction to Algorithms**  
Cormen et al., 2009



**Theoretical Neuroscience**  
Dayan and Abbott, 2001

## **Other tips**

---

### **Attend the weekly QB seminars**

Wednesdays at 12pm, Hawkins.

### **Attend QB Tea Time**

Wednesdays at 4pm, Samet.  
Email me to get on mailing list.