Barry Grant
bjgrant@umich.edu
http://thegrantlab.org

Ireland


Finland
Norway
Denmark
United Kingdom
Ireland
Poland
Belarus
Germany
France
Austria
Romania
Italy
Spain
Portugal
Greece
North


ŠKODA

B.Sc. 1 Biochemistry

M.Res. 2 Bioinformatics

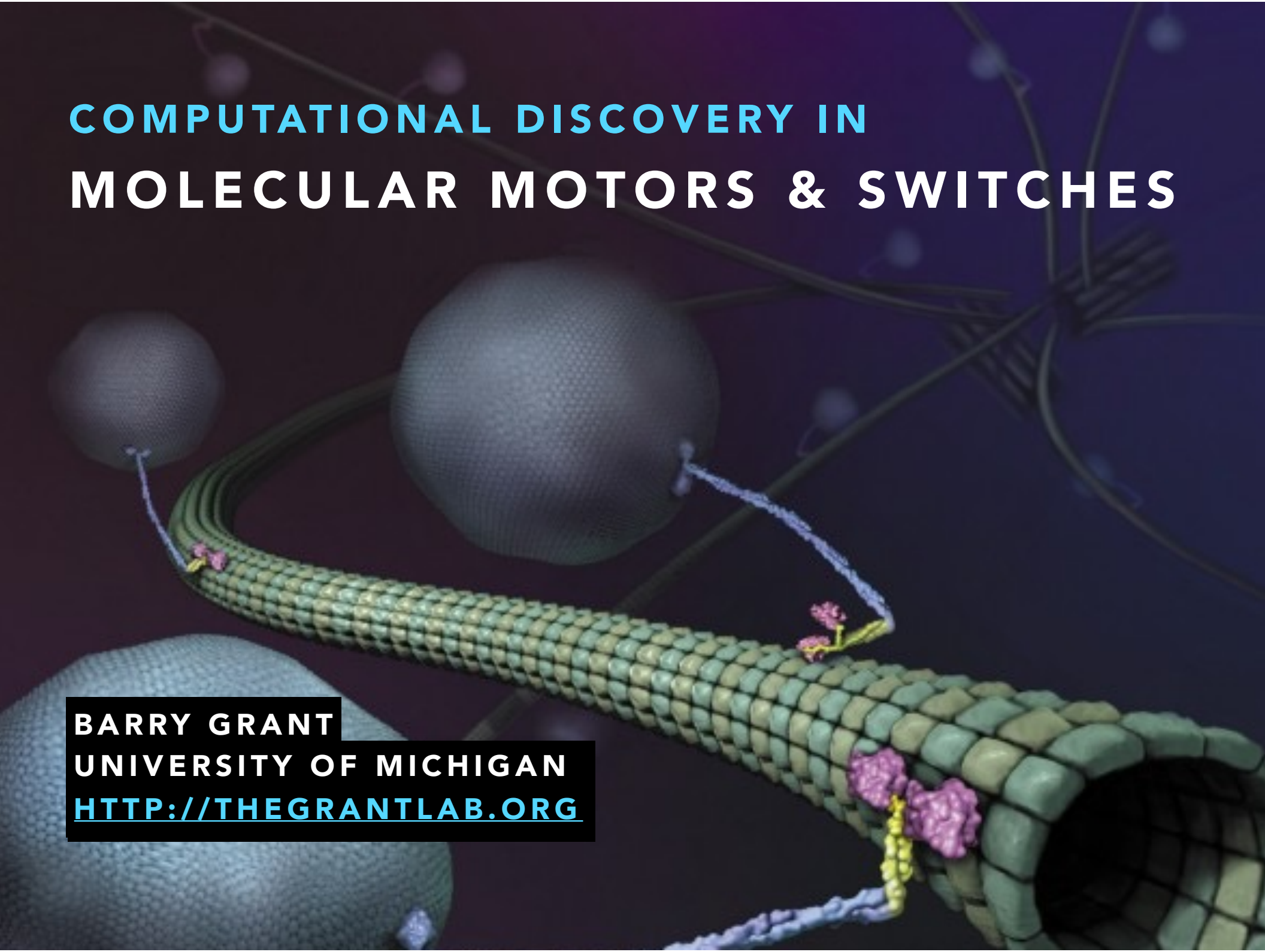
Ph.D. 3 Chemistry

Posdoc 4 Biophysics

Scientist 5 Bioinformatics

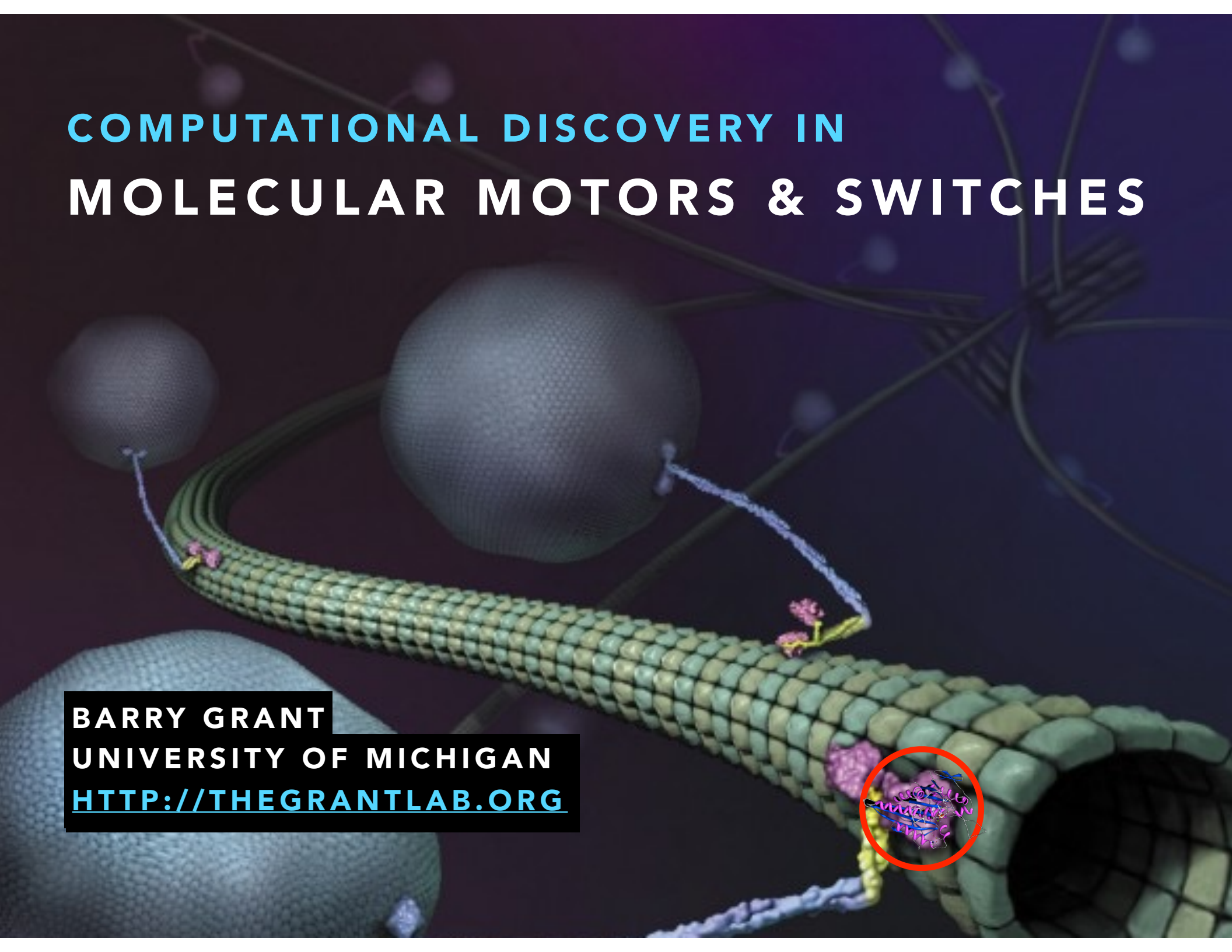
Faculty 6 Bioinformatics

COMPUTATIONAL DISCOVERY IN MOLECULAR MOTORS & SWITCHES

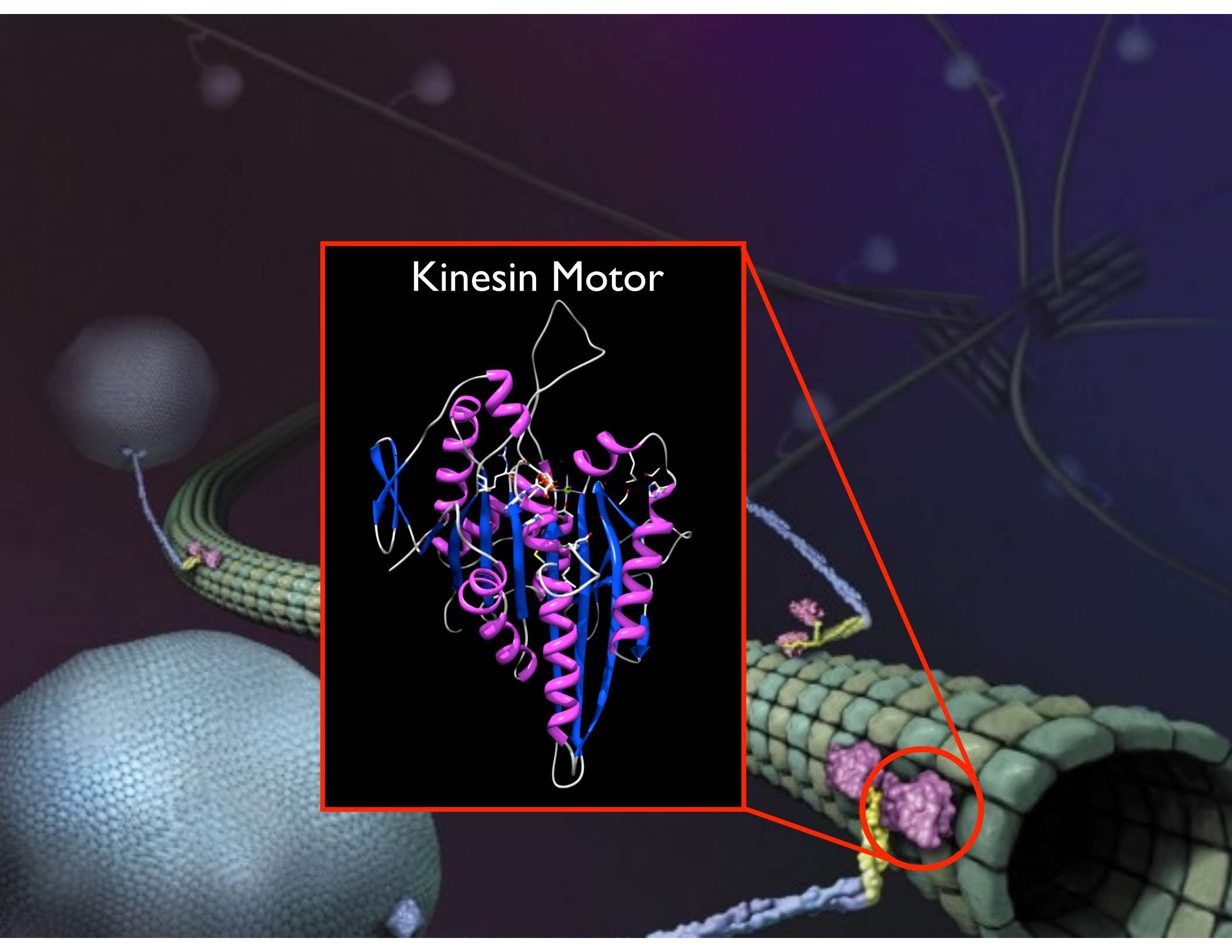


BARRY GRANT
UNIVERSITY OF MICHIGAN
HTTP://THEGRANTLAB.ORG

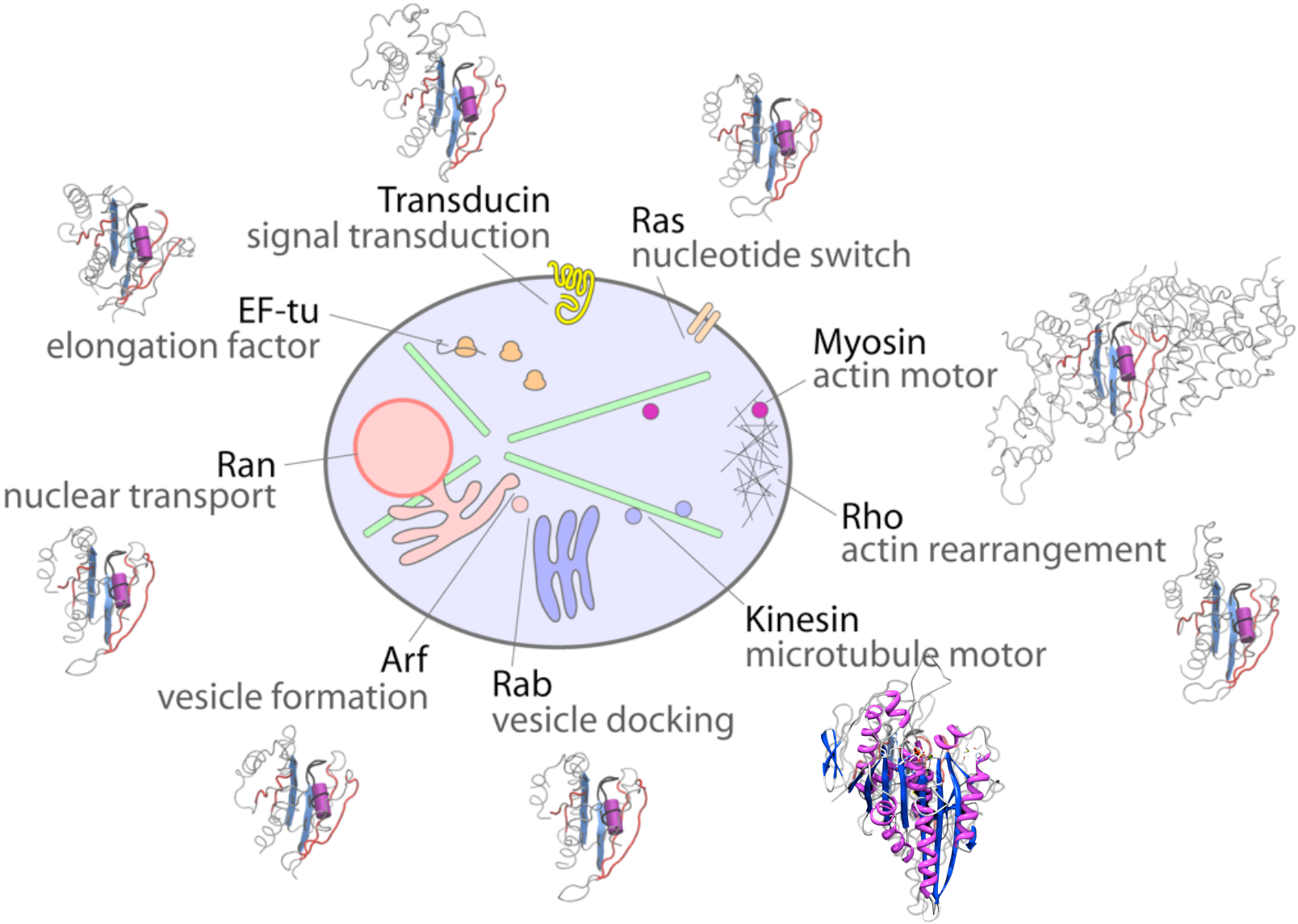
COMPUTATIONAL DISCOVERY IN
MOLECULAR MOTORS & SWITCHES



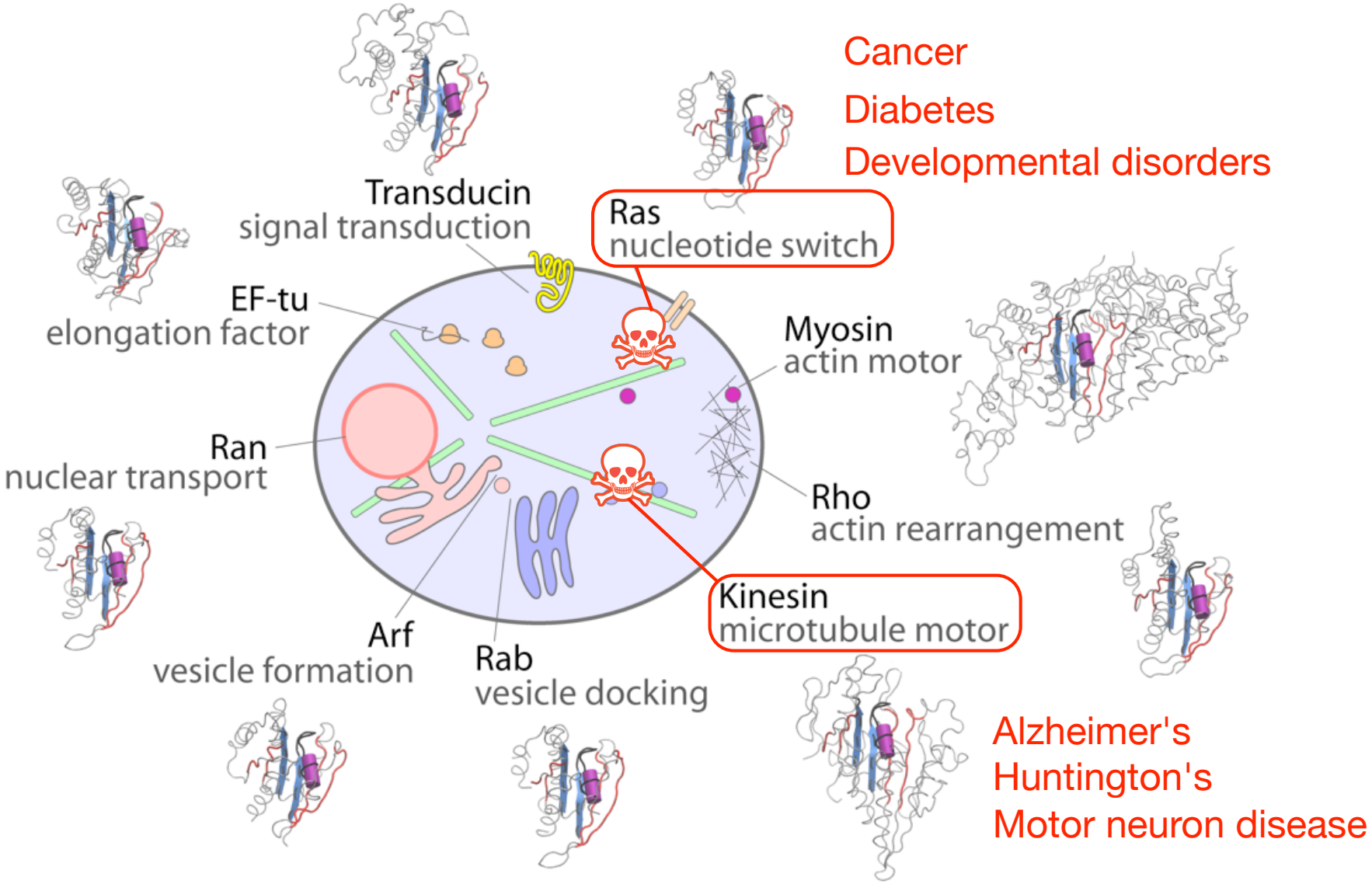
BARRY GRANT
UNIVERSITY OF MICHIGAN
HTTP://THEGRANTLAB.ORG


Kinesin Motor

Molecular **motors** and **switches** lie at the heart of key biological processes

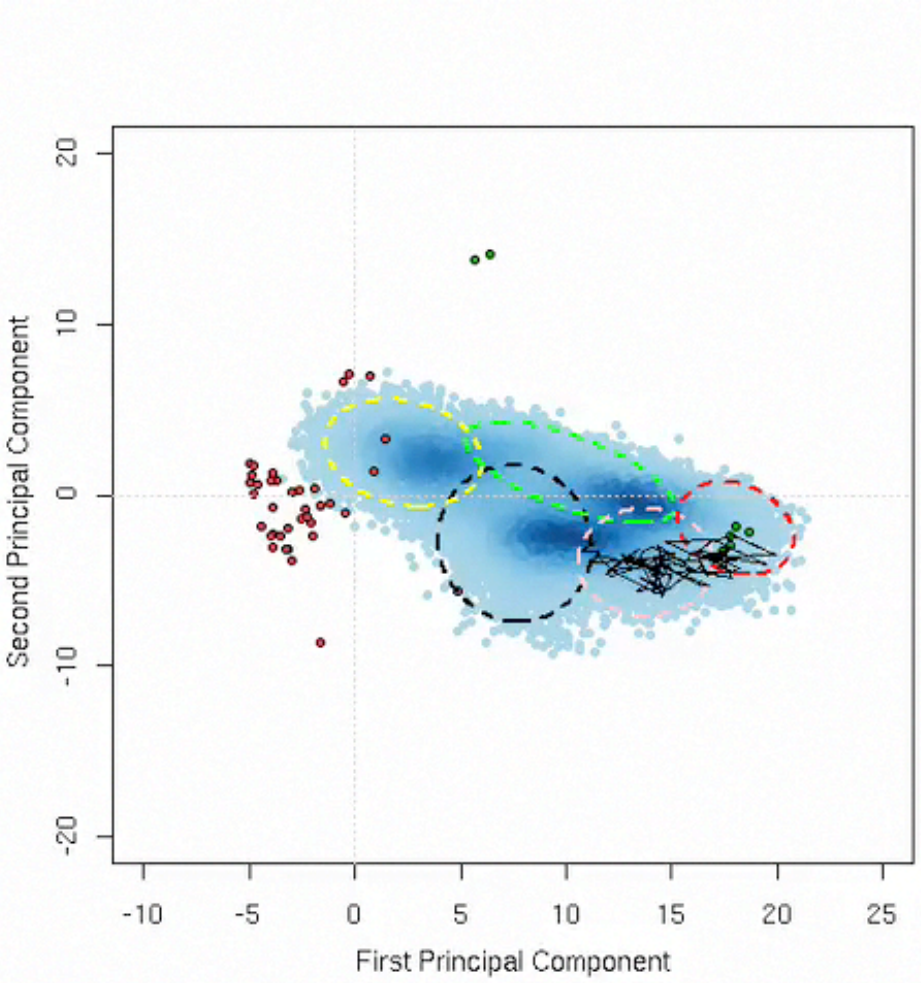


Aberrant function of motors and switches is associated with many diseases



Simulations of conformational activation

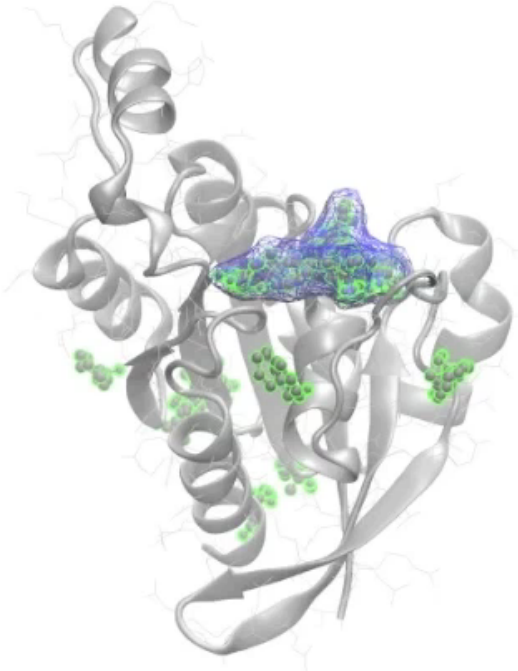
Accelerated molecular dynamics simulations reveal the mechanism of Ras molecular switch activation and the perturbing effect of mutations



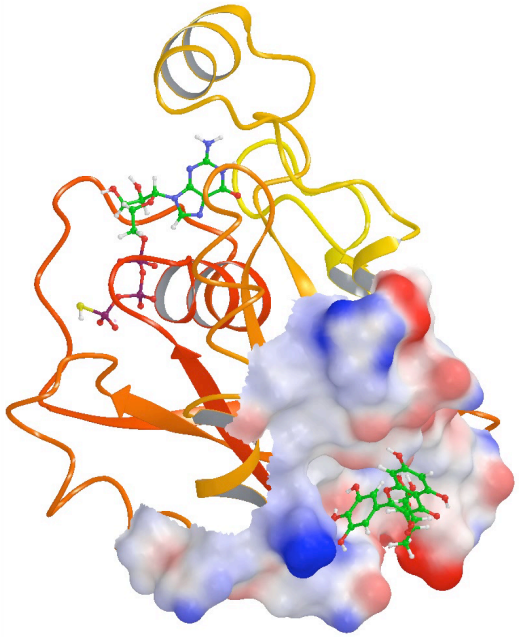
Grant *et al.* PLoS Comp Biol (2009, 2010, 2011, 2013, 2014); Biophys J. (2010, 2013, 2014)

Allosteric inhibitor design & testing

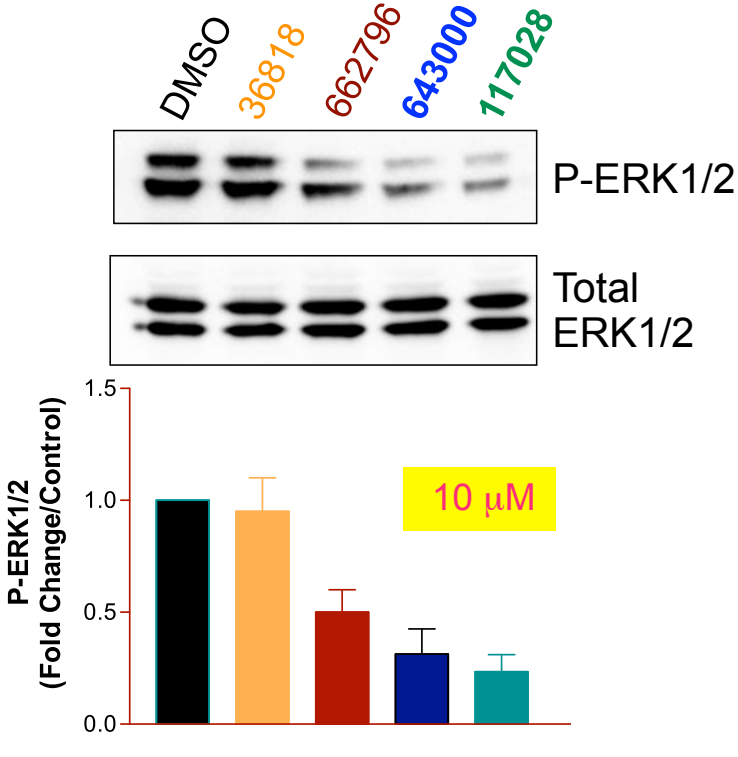
Computational methods can identify novel transient binding pockets & cognate inhibitors that may offer new avenues for future therapeutics



Ensemble binding site mapping



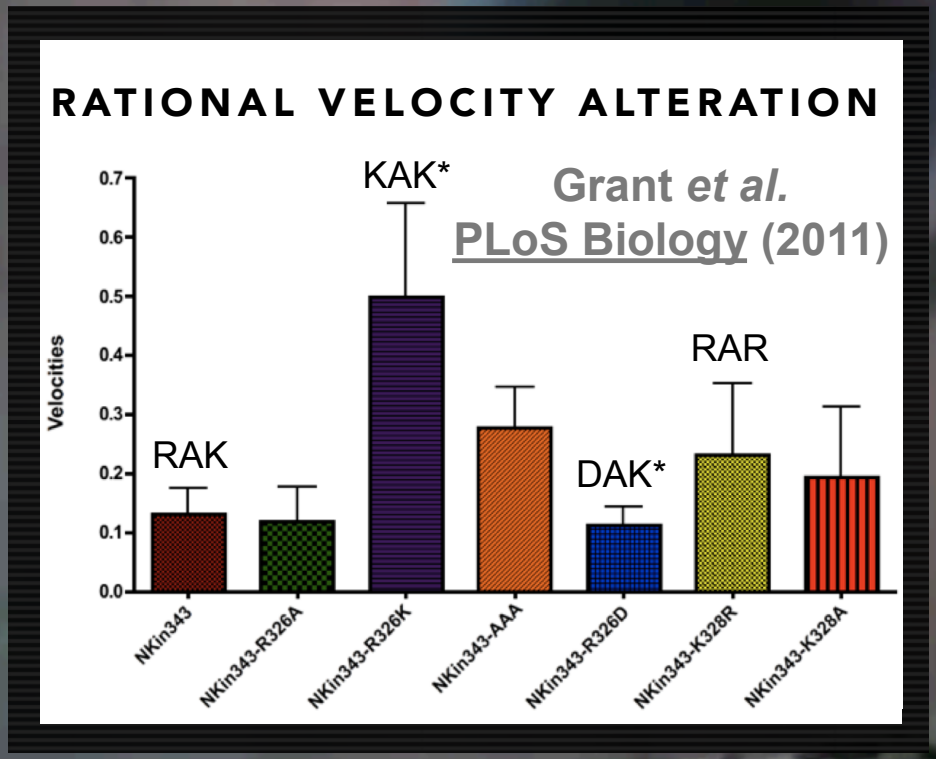
Ensemble small molecule docking



Compound testing in cancer cell lines

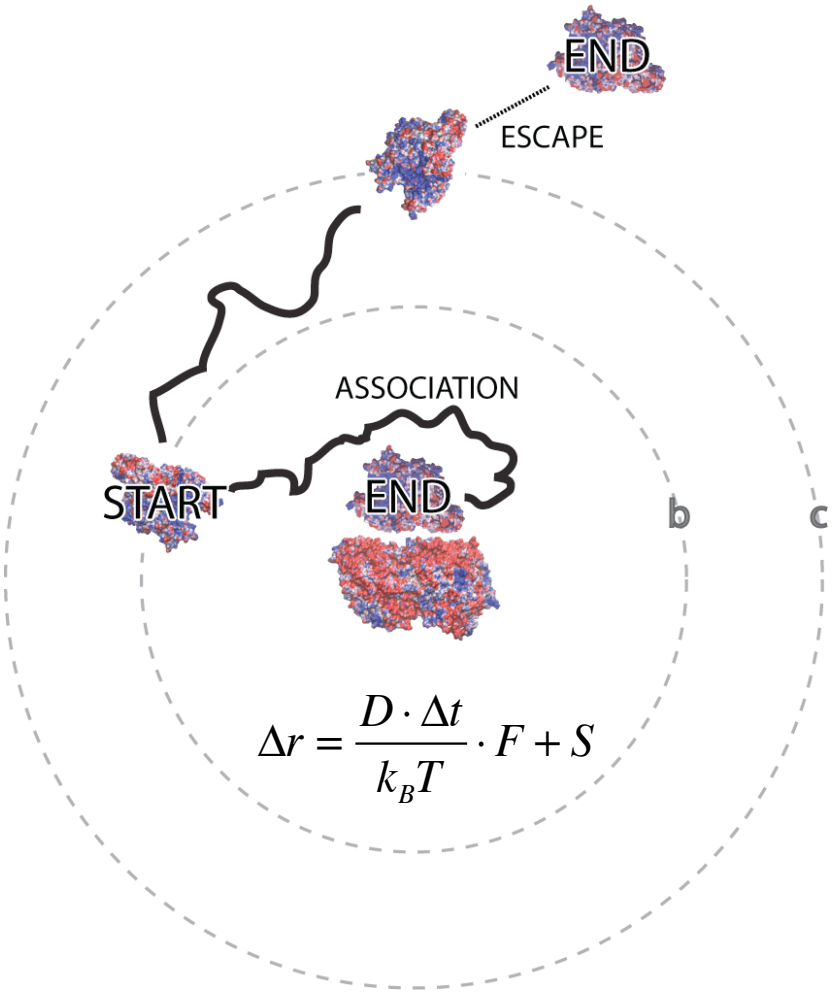
Curr Opin Struct Biol (2010); PLoS One (2011, 2012); U.S. Patent No: 61/544,237

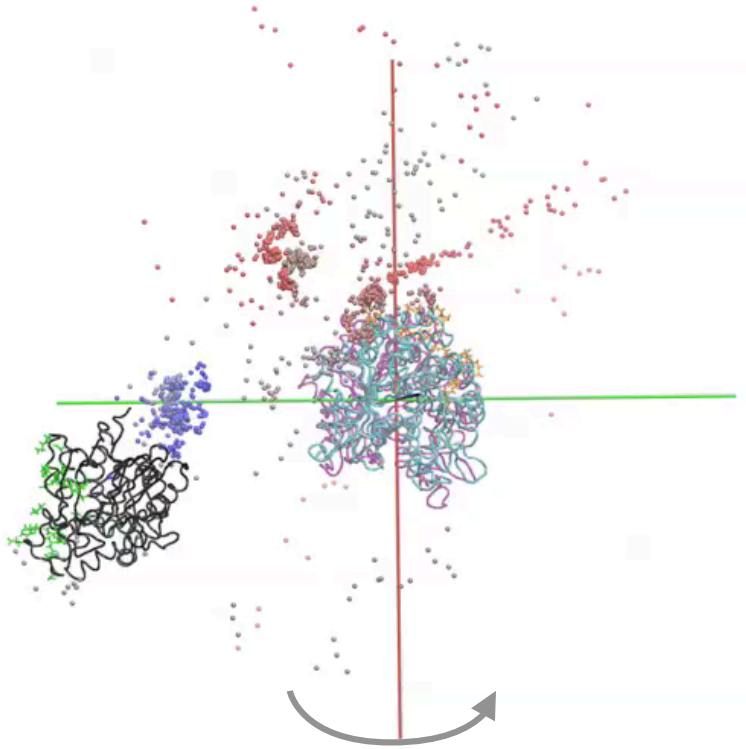
MAPPING MOTOR PROCESSIVITY DETERMINANTS



Animation available from: http://multimedia.mcb.harvard.edu/

Brownian dynamics model of kinesin-tubulin association

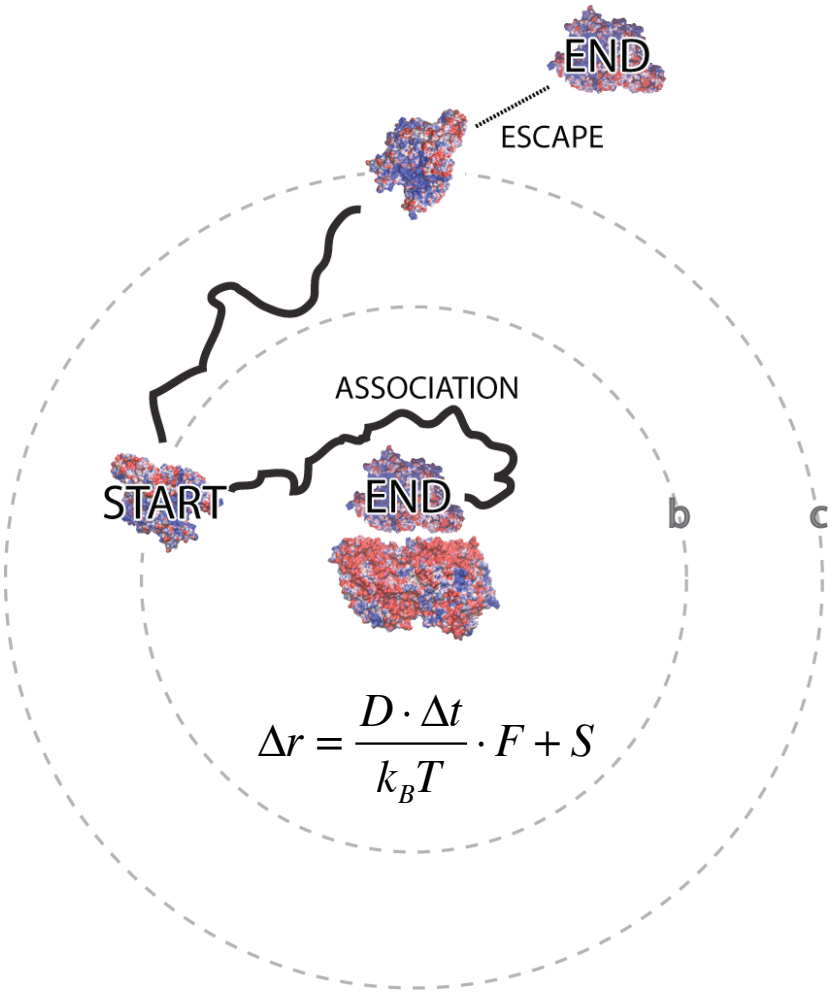




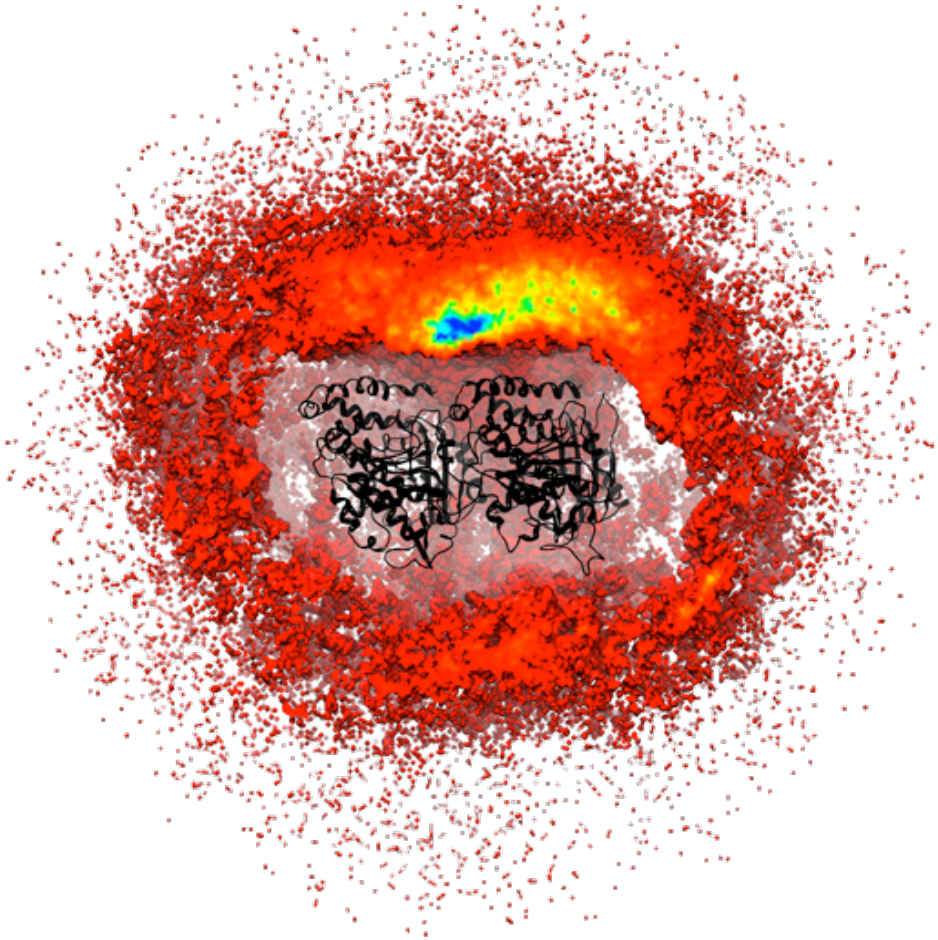
Single Trajectory

PLoS Biology (2011)

Brownian dynamics model of kinesin-tubulin association



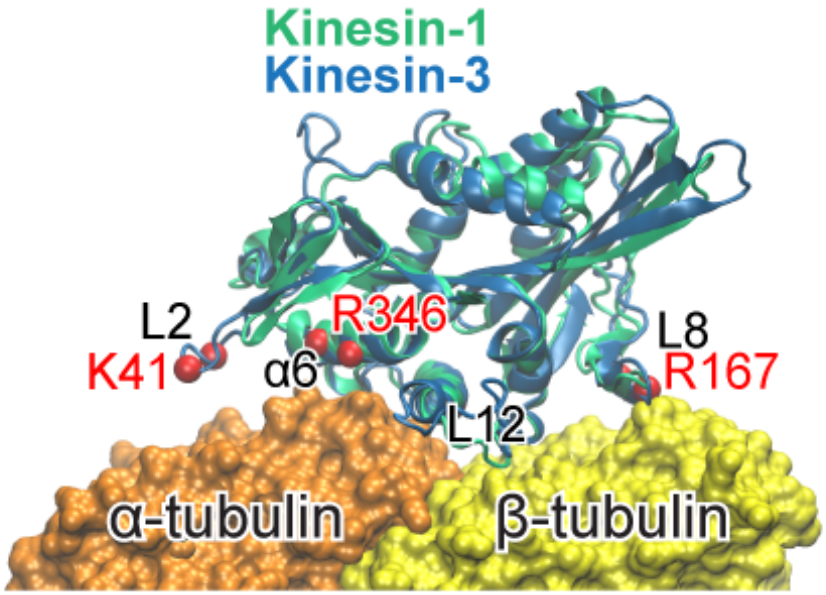
$$\Delta r = \frac{D \cdot \Delta t}{k_B T} \cdot F + S$$

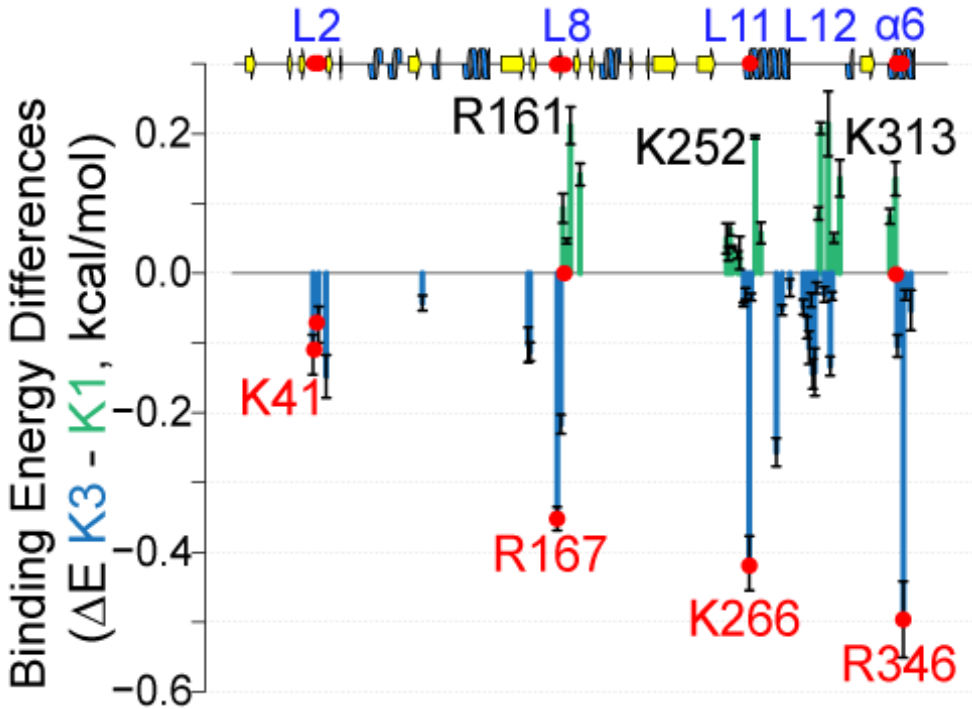


200k Trajectories

Mapping motor processivity determinants

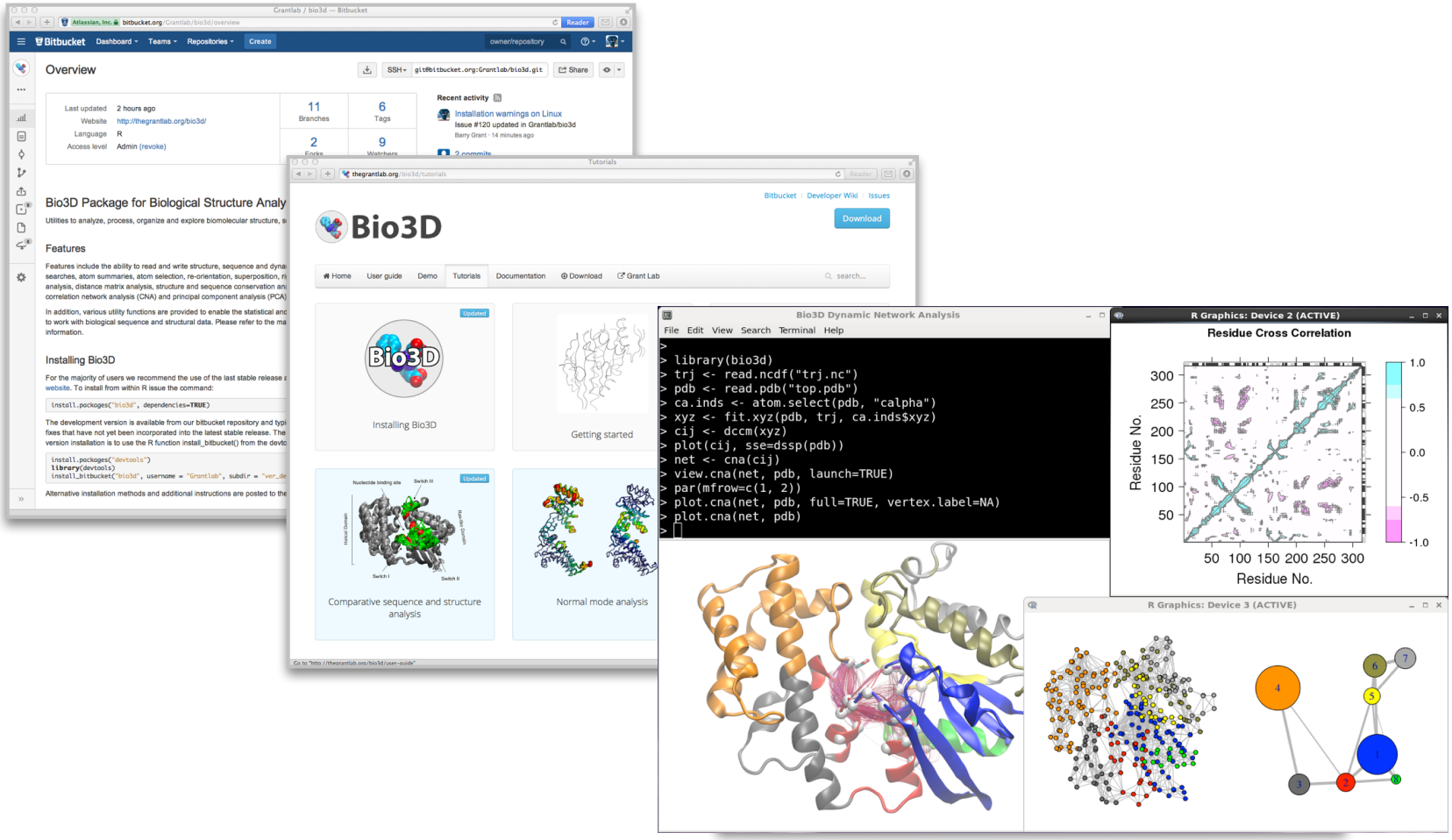
Select family-specific motor-microtubule interactions are predicted to influence complex stability and motor motility





Scarabelli … Grant Biophys J. (2015, in press)

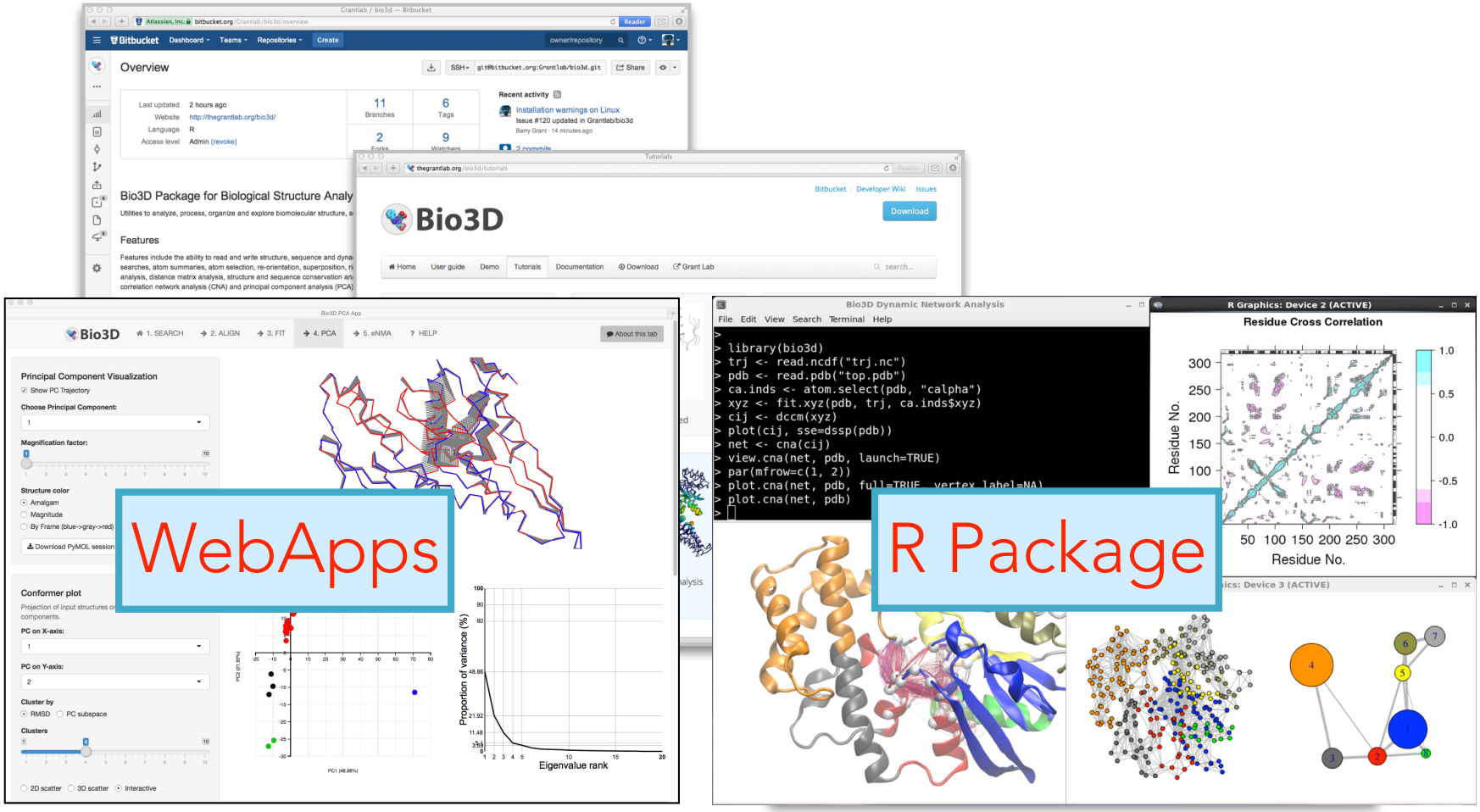
Side Note: Integrating protein structural dynamics and evolutionary analysis with **Bio3D**



http://thegrantlab.org/bio3d/

Skjærven, Yao, … Grant BMC Bioinf (2014)

Side Note: Integrating protein structural dynamics and evolutionary analysis with **Bio3D**



http://thegrantlab.org/bio3d/

Skjærven, Yao, … Grant BMC Bioinf (2014)

ACKNOWLEDGEMENTS

Kristen Verhey (**UM**, CDB)

Rob Cross (U. of Warwick, UK)

Etsuko Muto (Riken Institute, Japan)

Carolyn Moores (UCL, UK)

Steven Rosenfeld (Cleveland Clinic)

Alemayehu Gorfe (UTMS-H)

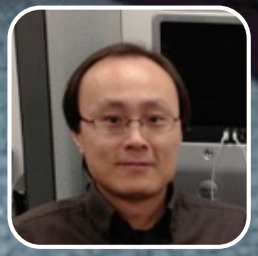
Bjoern Peters (LIAI)

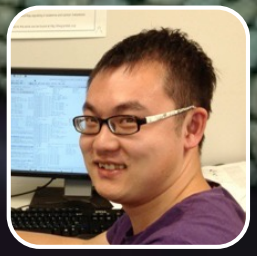
Sivaraj Sivaramakrishnan (**UM**, CDB)

Chao-Yie Yang (**UM**, Internal Med)

John Traynor (**UM**, Pharmacology)

Lars Skjærven (U.of Bergen, Norway)





HTTP://THEGRANTLAB.ORG

Barry Grant
bjgrant@umich.edu
http://thegrantlab.org

What is Git?

(1) An unpleasant or contemptible person. Often incompetent, annoying, senile, elderly or childish in character.

(2) A modern distributed version control system with an emphasis on speed and data integrity.

What is Git?

(1) An unpleasant or contemptible person. Often incompetent, annoying, senile, elderly or childish in character.

(2) A modern distributed version control system with an emphasis on speed and data integrity.

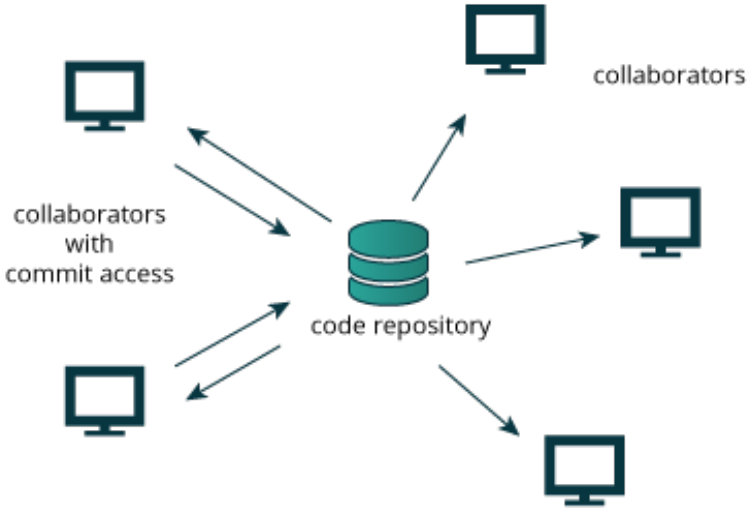
Version Control

Version control systems (VCS) record changes to a file or set of files over time so that you can recall specific versions later.

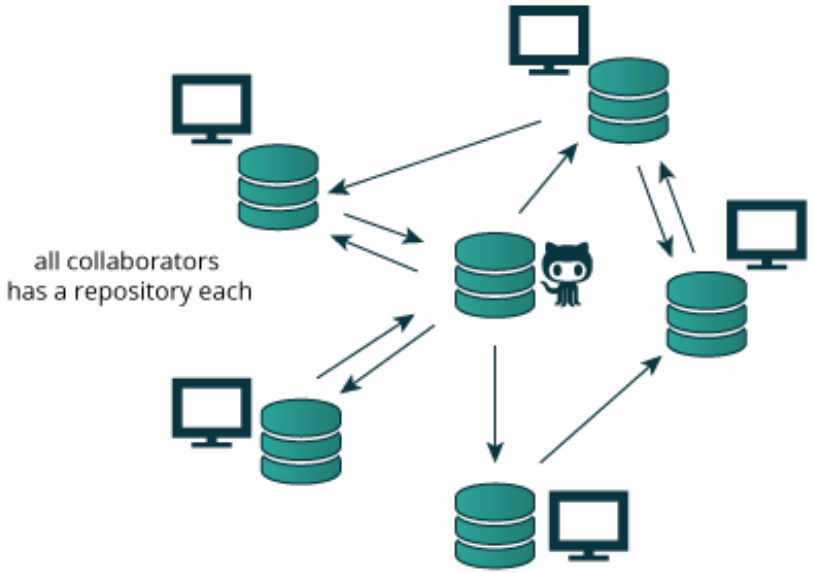
Client–server	**Free/open-source**	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)
	Proprietary	Software Change Manager (1970s) · Panvalet (1970s) · Endevor (1980s) · Dimensions CM (1980s) · *DSEE* (1984) · Synergy (1990) · ClearCase (1992) · *CMVC* (1994) · *Visual SourceSafe* (1994) · Perforce (1995) · StarTeam (1995) · Integrity (2001) · Surround SCM (2002) · AccuRev SCM (2002) · SourceAnywhere (2003) · Vault (2003) · Team Foundation Server (2005) · Team Concert (2008)
Distributed	**Free/open-source**	*GNU arch* (2001) · Darcs (2002) · *DCVS* (2002) · *ArX* (2003) · Monotone (2003) · *SVK* (2003) · *Codeville* (2005) · Bazaar (2005) · **Git** (2005) · Mercurial (2005) · Fossil (2007) · Veracity (2010)
	Proprietary	*TeamWare* (1990s?) · Code Co-op (1997) · BitKeeper (1998) · Plastic SCM (2006)

There are many VCS available, see:
https://en.wikipedia.org/wiki/Revision_control

Client-Server *vs* Distributed VCS

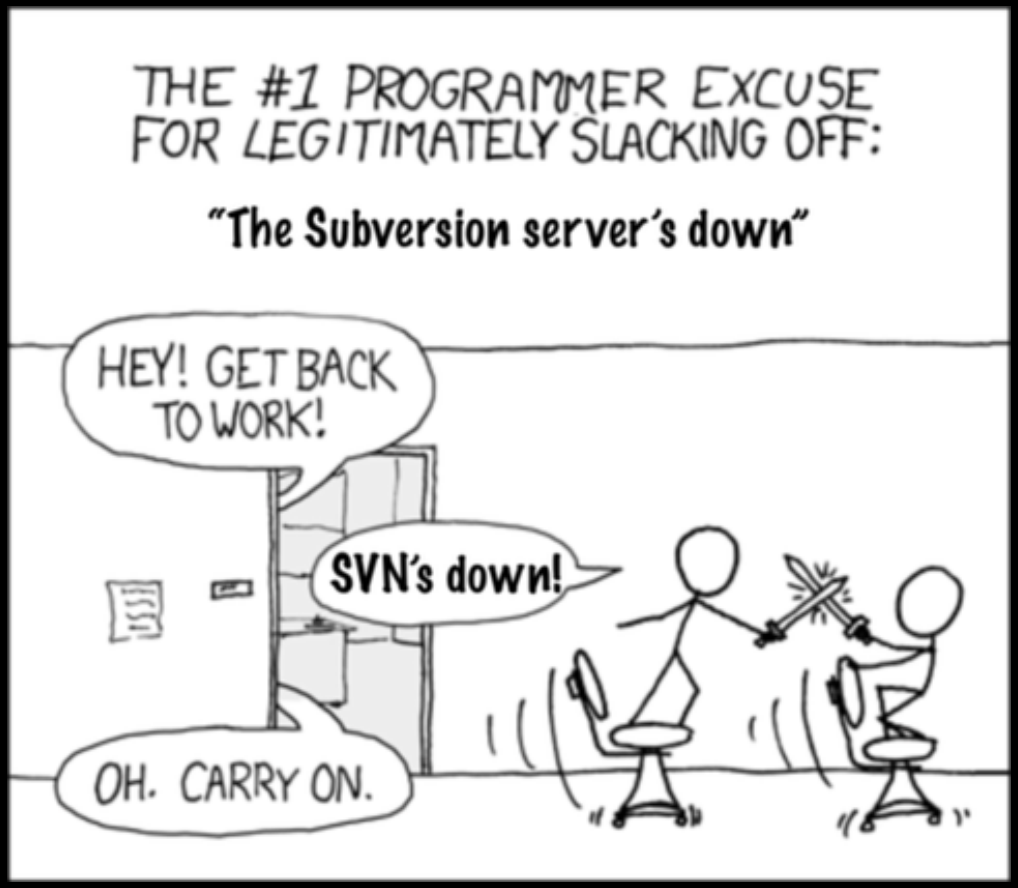


Client-server approach



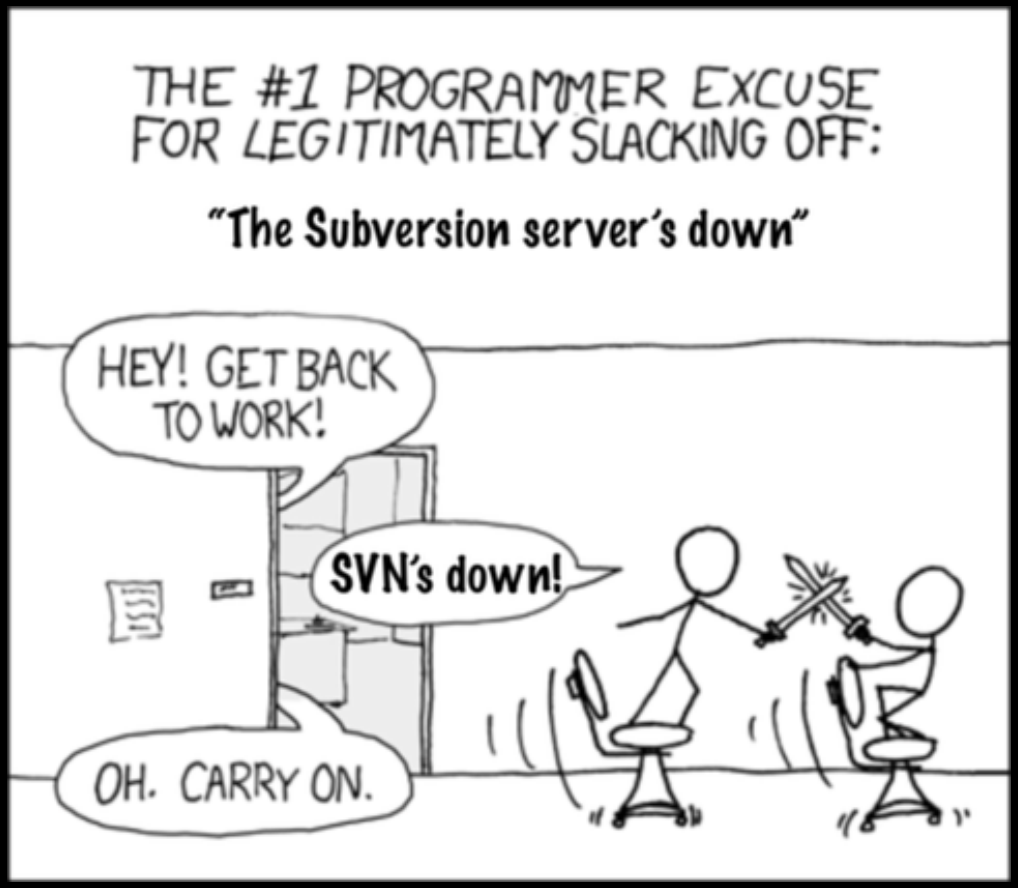
Distributed approach

Distributed version control systems (DCVS) allows multiple people to work on a given project without requiring them to share a common network.



http://tinyurl.com/distributed-advantages

Git is now the most popular free VCS!



Git offers:

- Speed
- Backups
- Off-line access
- Small footprint
- Simplicity*
- Social coding

http://tinyurl.com/distributed-advantages

Why use Git?

Q. Would you write your lab book in pencil, then erase and overwrite it every day with new content?

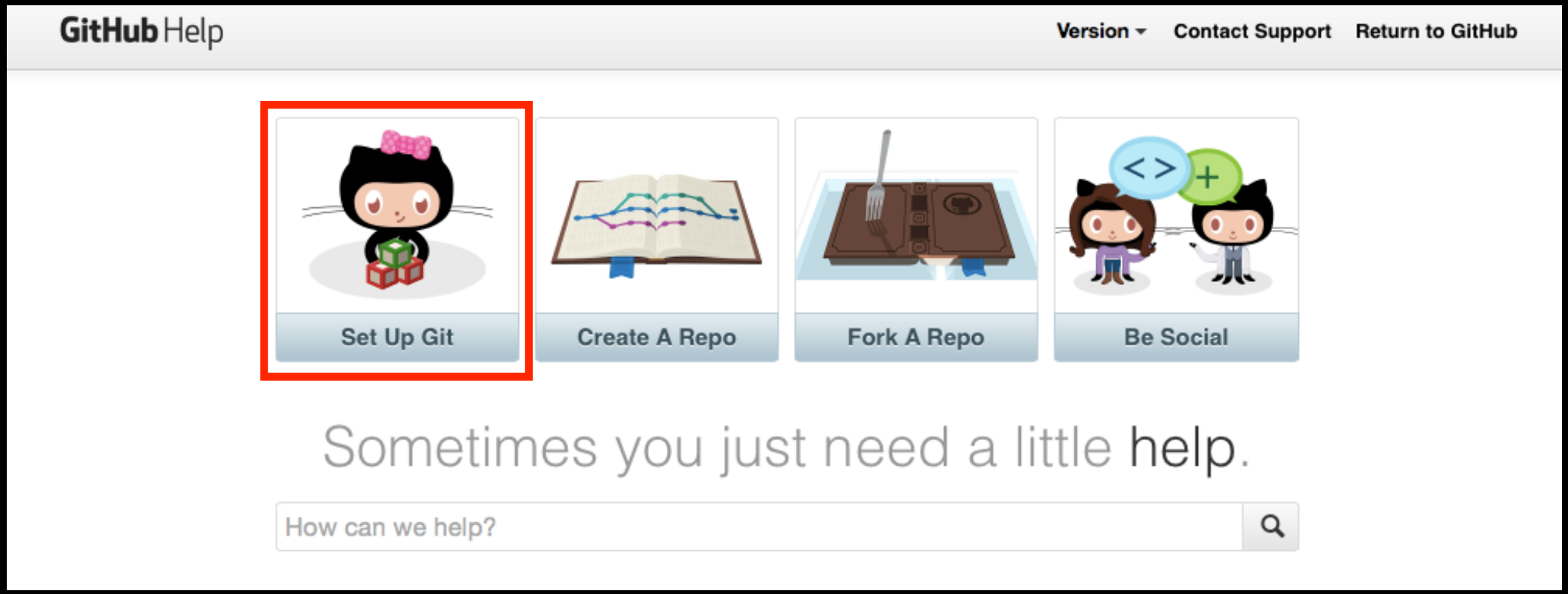
Q. Would you write your lab book in pencil, then erase and overwrite it every day with new content?

Version control is the lab notebook of the digital world: it's what professionals use to keep track of what they've done and to collaborate with others.

Why use Git?

- Provides '**snapshots**' of your project during development and provides a full record of project **history**.
- Allows you to easily **reproduce** and **rollback** to past versions of analysis and compare differences. (N.B. Helps fix software regression bugs!)
- Keeps **track of changes** to code you use from others such as fixed bugs & new features
- Provides a mechanism for sharing, updating and collaborating (like a social network)
- Helps keep your work and software organized and available

Obtaining Git



https://help.github.com

Configuring Git

Do it Yourself!

Configuring Git

(**RStudio > Tools > Shell**)

```
# First tell Git who you are
> git config --global user.name  “Barry Grant”
> git config --global user.email  “bjgrant@umich.edu”
```

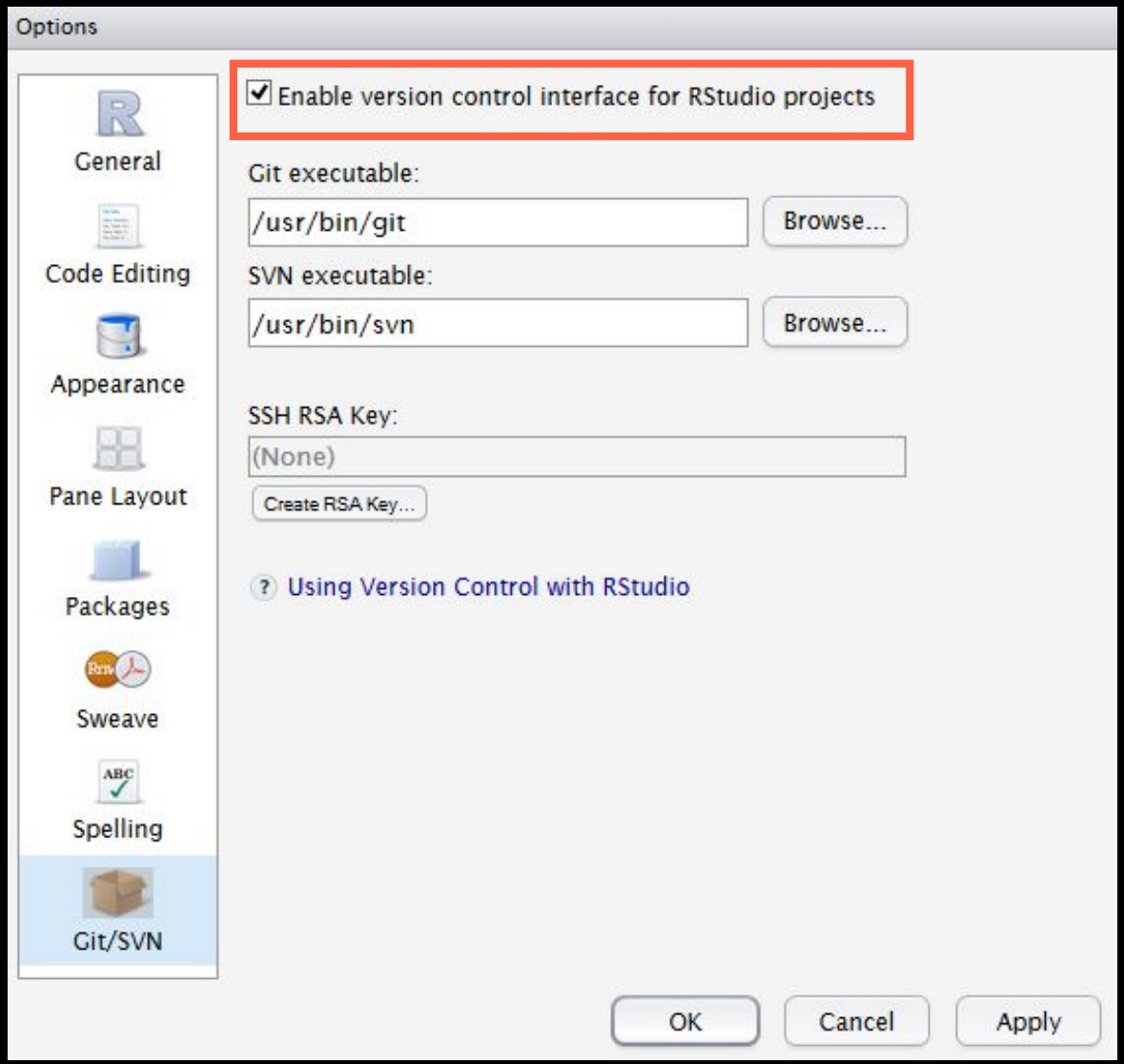
Using Git

Using Git

1. Initiate a Git repository.
2. Edit content (i.e. change some files).
3. Store a 'snapshot' of the current file state.*

Using Git with RStudio

(RStudio > Tools > Global Options > Git/SVN)



Your Turn:

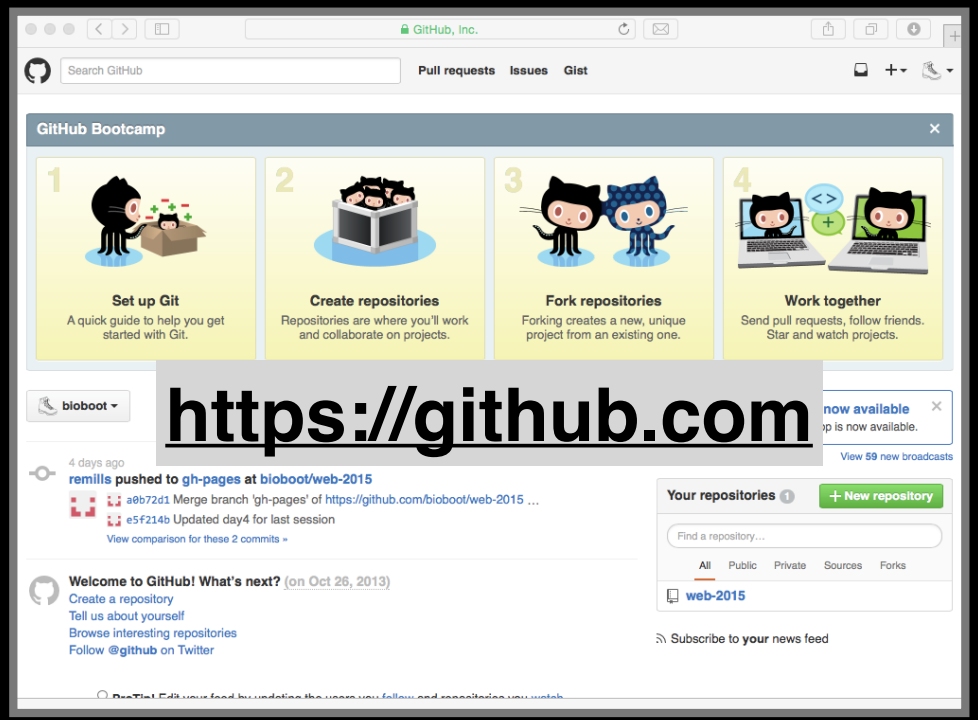
http://tinyurl.com/rclass-github

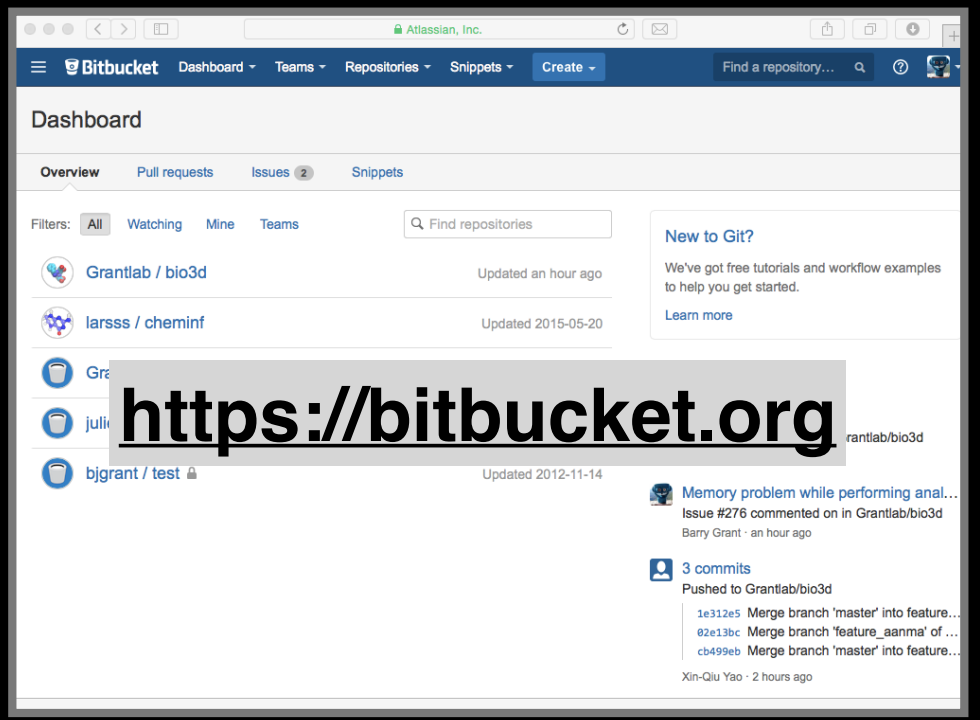
Step 3-4 only please!

(We have already done steps 1 & 2)

GitHub & Bitbucket

GitHub and **Bitbucket** are two popular hosting services for Git repositories. These services allow you to share your projects and collaborate with others using both '**public**' and '**private**' repositories*.





wsj.com

Nikkei 17893.73 0.49% | Hang Seng 21404.96 0.72% | U.S. 10 Yr -0/32 Yield 2.074% | Crude Oil 39.17 -0.36% | Yen 119.16 0.26% | EXPAND

THE WALL STREET JOURNAL.

Workers Get New Tools for Airing Their Gripes

Cell Carriers Battle for Wi-Fi Airwaves

Snapchat Names ex-Mattel Exec Vollero Its Finance Chief

3234

433

TECH

GitHub Raises $250 Million at $2 Billion Valuation

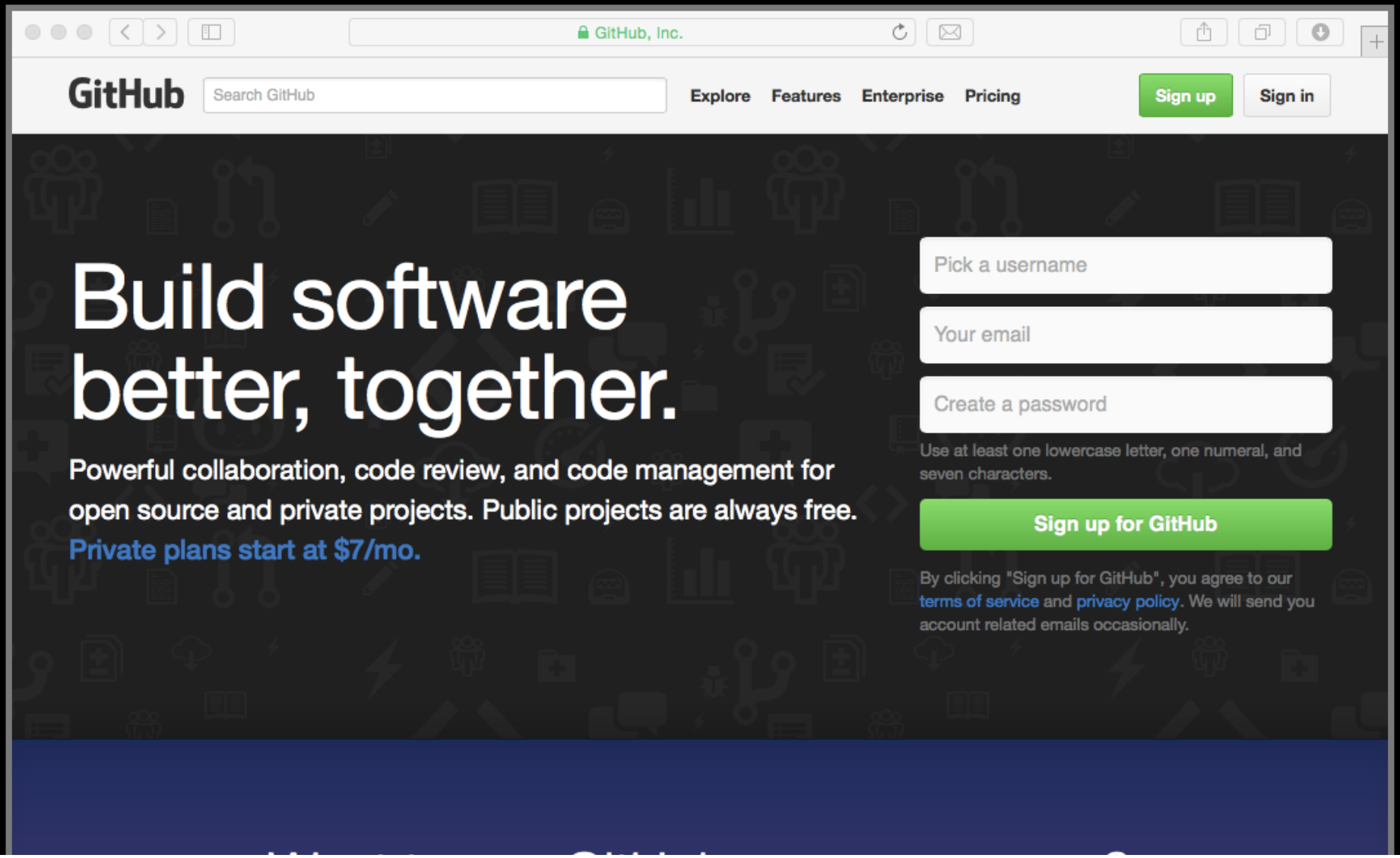
Capital raise puts company's total funding at $350 million

What is the big deal?

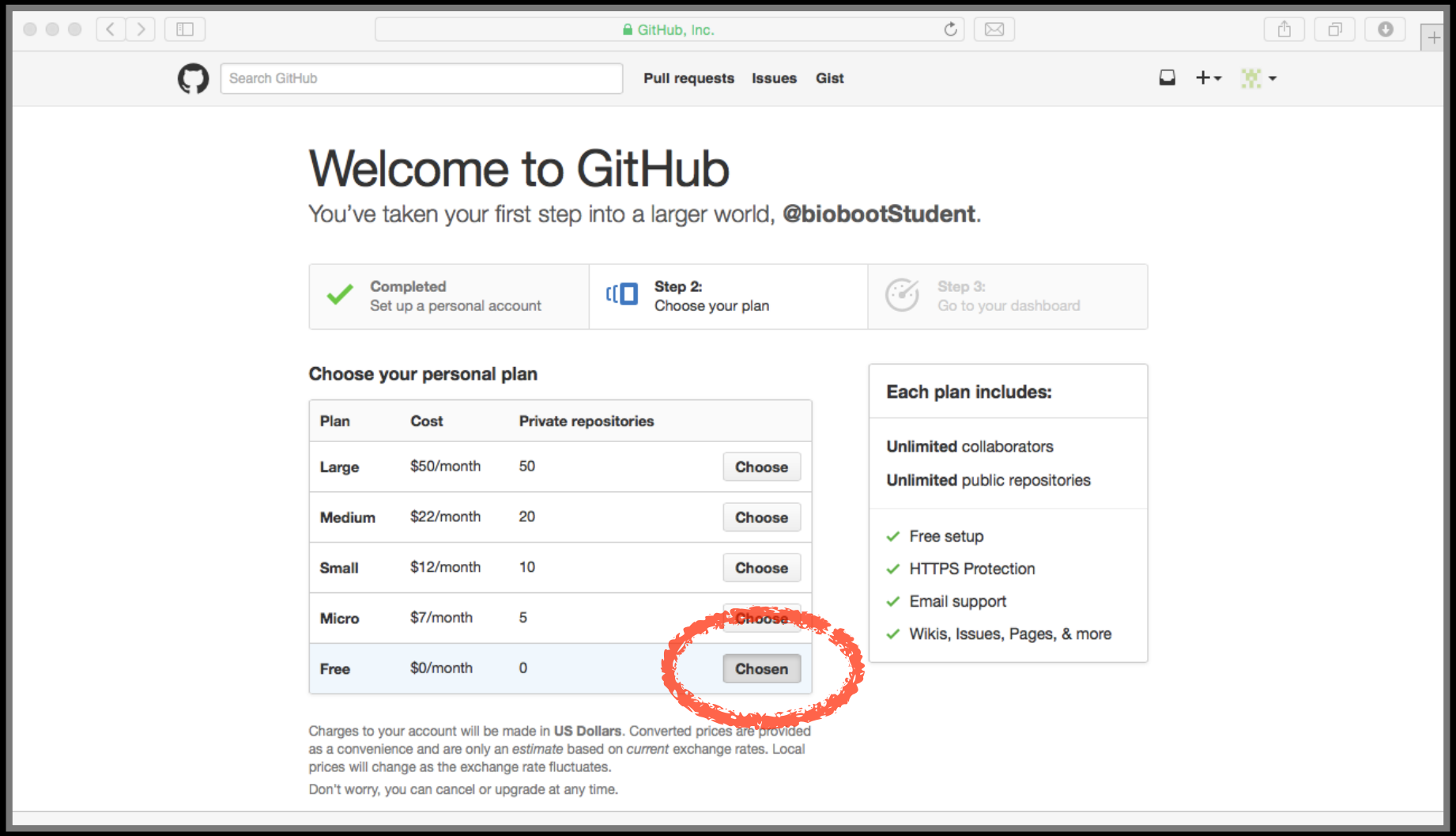
- At the simplest level GitHub and Bitbucket offer **backup** of your projects history and a centralized mechanism for **sharing** with others by putting **your Git repo online**.
    - GitHub in particular is often referred to as the “nerds FaceBook and LinkedIn combined”.
- At their core both services **offer a new paradigm for open collaborative project development**, particularly for software.
    - In essence they allow anybody to contribute to any public project and get acknowledgment.

First sign up for a GitHub account

https://github.com

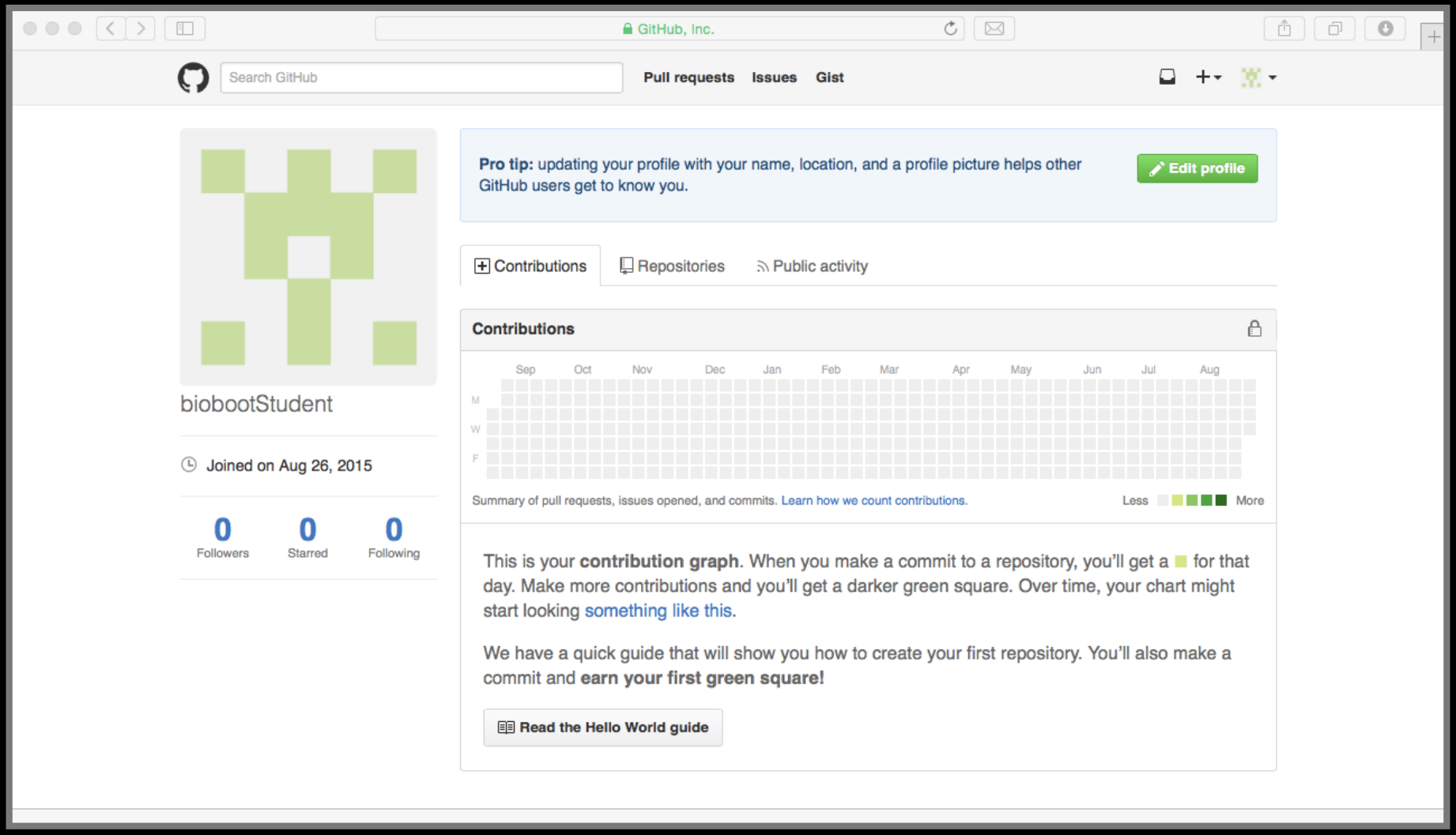


Pick the FREE plan!



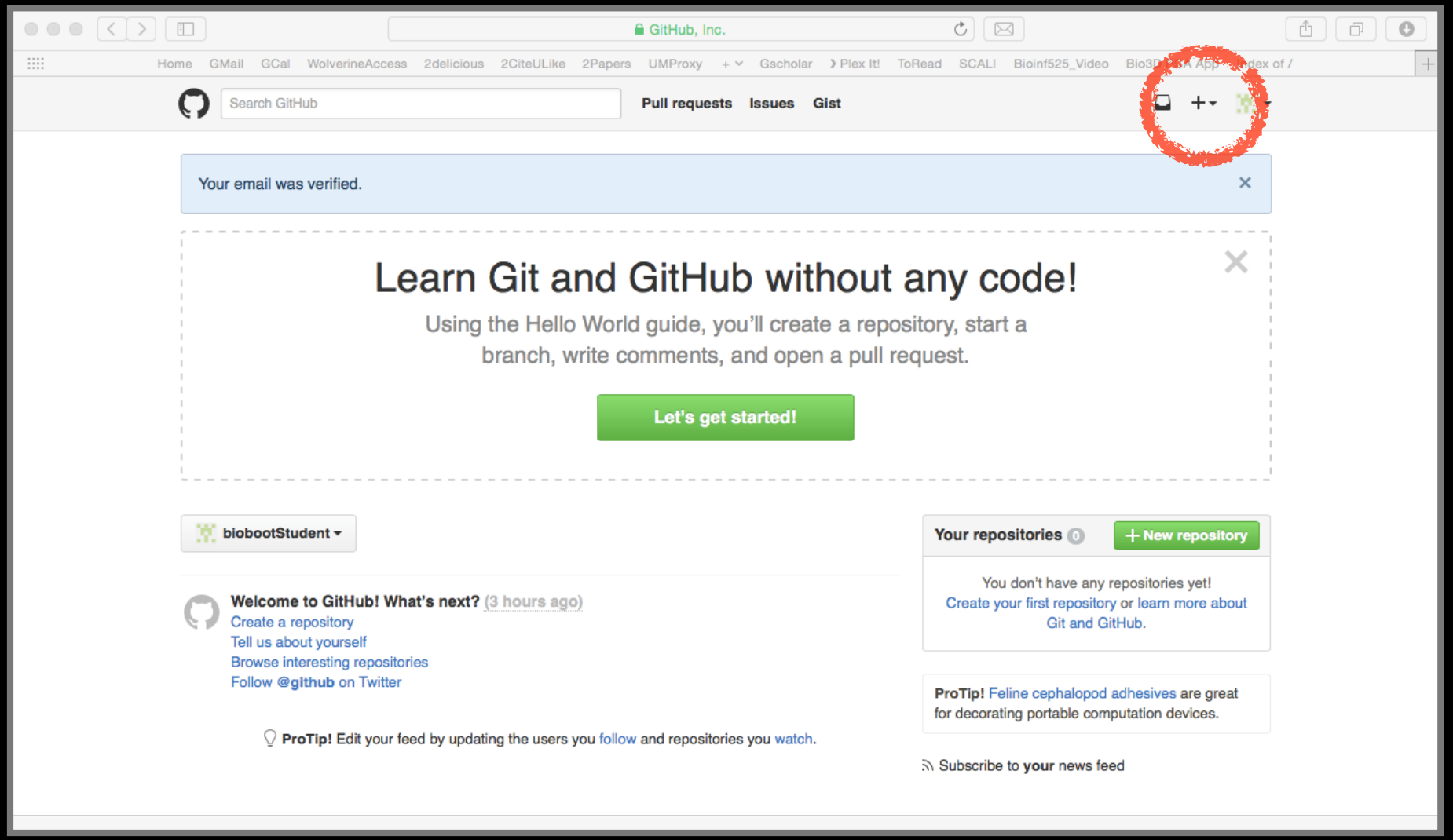
Your GitHub homepage

Check your email for verification request



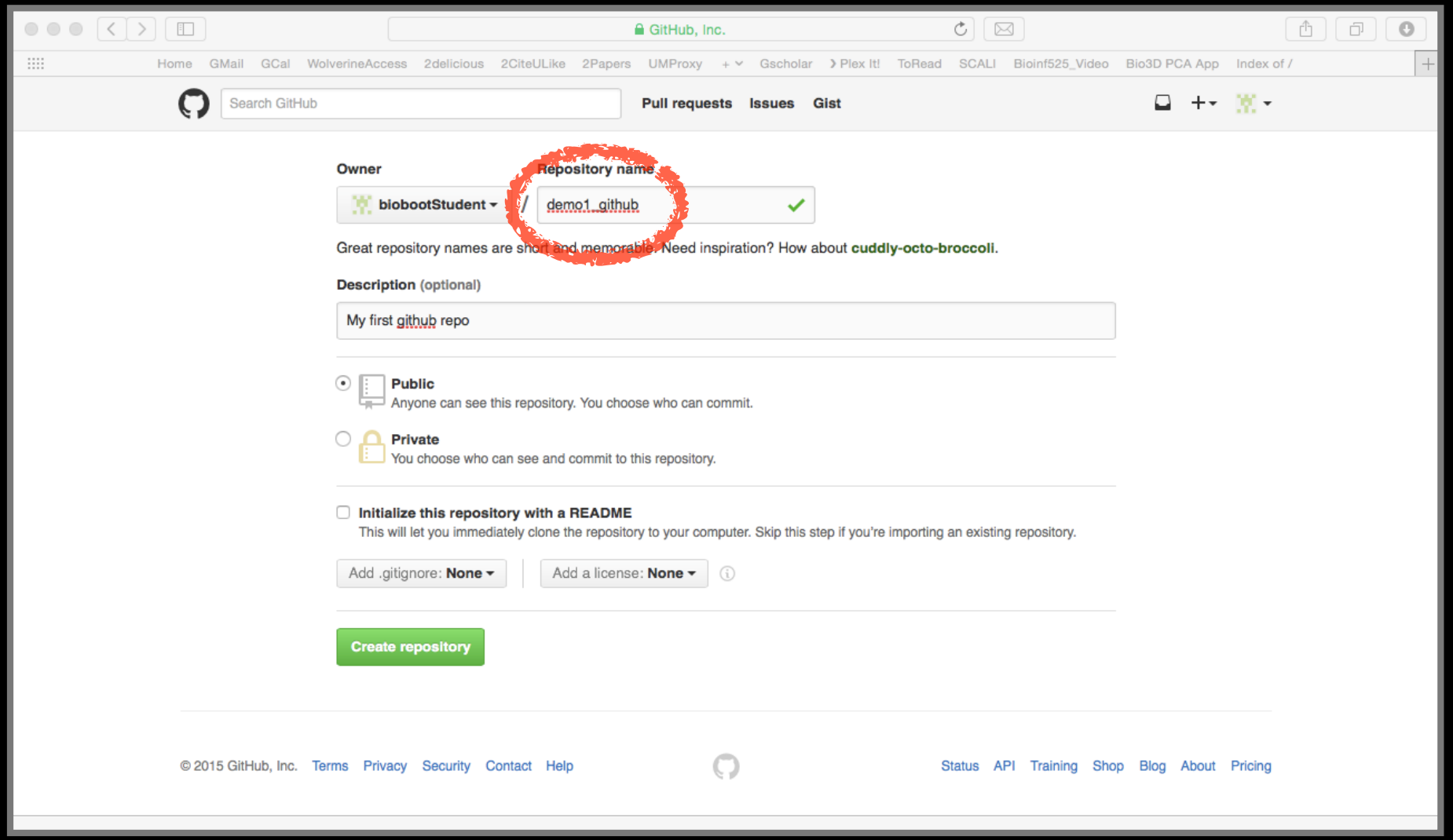
Skip the hello-world tutorial

https://guides.github.com/activities/hello-world/



Name your repo

test



Your Turn:

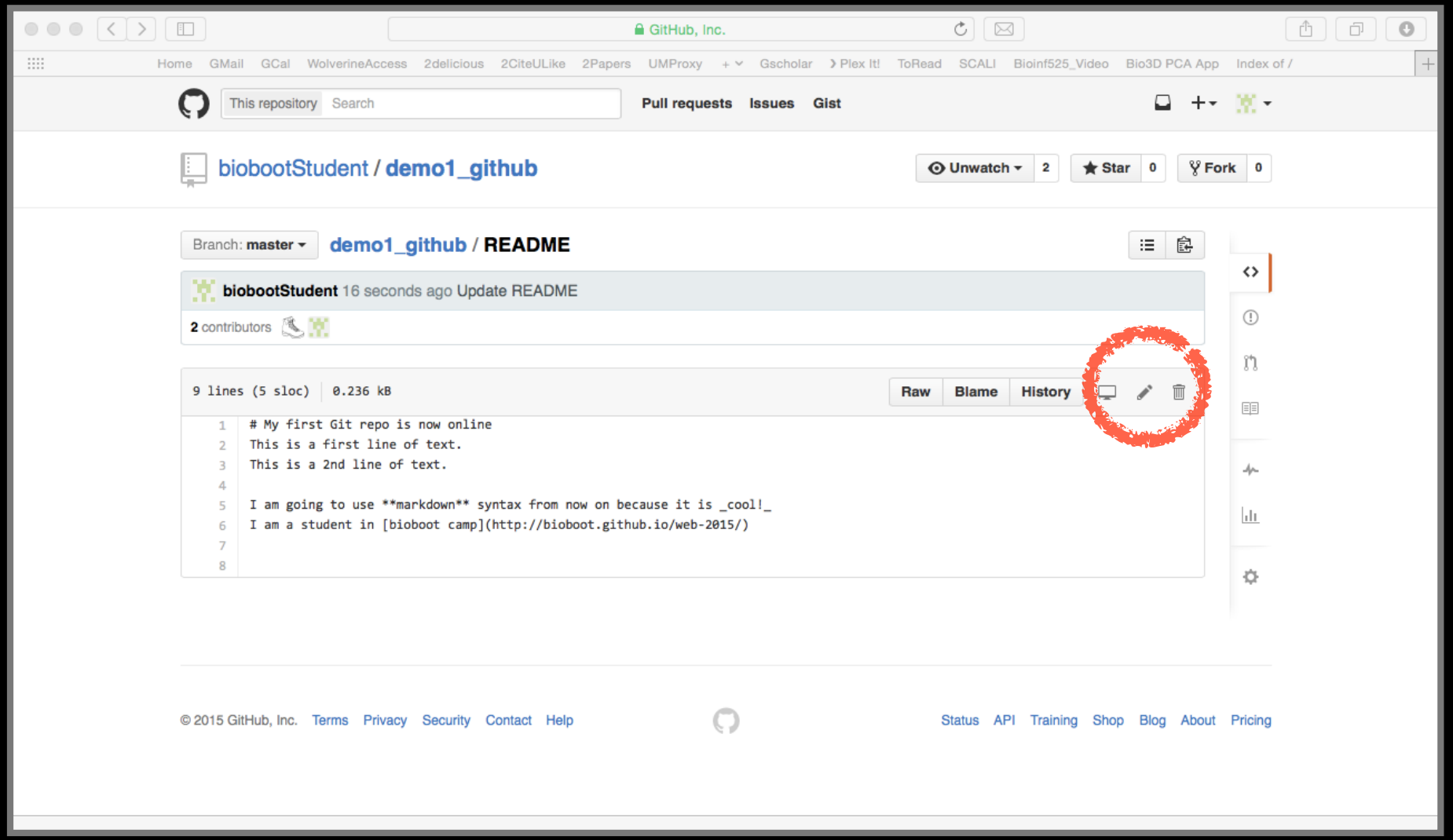
http://tinyurl.com/rclass-github

Step 6 only please!

(We have already done steps 1 to 5)

Side-note: How to edit online

Specifically lets add some Markdown content



Summary

- Git is a popular ‘distributed’ version control system that is lightweight and free
- GitHub and BitBucket are popular hosting services for git repositories that have changed the way people contribute to open source projects
- Introduced basic git and GitHub usage within RStudio and encouraged you to adopt these ‘best practices’ for your future projects.

Learning Resources

- **Set up Git**. If you will be using Git mostly or entirely via **GitHub**, look at these how-tos.
  < https://help.github.com/categories/bootcamp/ >
- **Getting Git Right**. Excellent **Bitbucket** git tutorials
  < https://www.atlassian.com/git/ >
- **Pro Git**. A complete, book-length guide and reference to Git, by Scott Chacon and Ben Straub.
  < http://git-scm.com/book/en/v2 >
- **StackOverflow**. Excellent programming and developer Q&A.
  < http://stackoverflow.com/questions/tagged/git >

Learning git can be painful!

However in practice it is not nearly as crazy-making as the alternatives:

- Documents as email attachments
- Hair-raising ZIP archives containing file salad
- Am I working with the most recent data?
- Archaelogical “digs” on old email threads and uncertainty about how/if certain changes have been made or issues solved

Finally Please remember that **GitHub** and **BitBucket** are **PUBLIC** and that you should cultivate your professional and scholarly profile with intention!