



**Barry Grant**

[bjgrant@umich.edu](mailto:bjgrant@umich.edu)

<http://thegrantlab.org>

# Ireland



B.Sc. 1 Biochemistry

M.Res. 2 Bioinformatics

Ph.D. 3 Chemistry

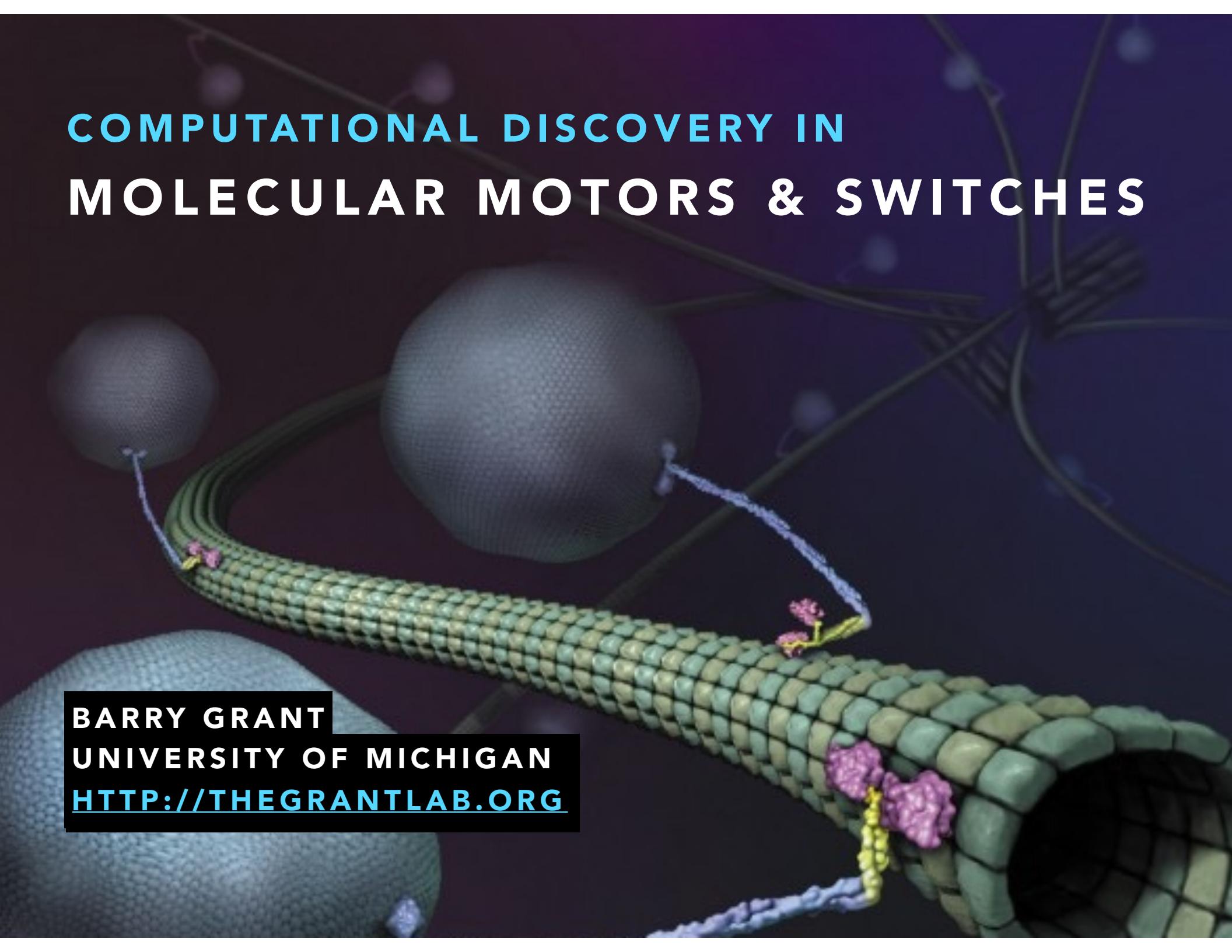
Posdoc 4 Biophysics

Scientist 5 Bioinformatics

Faculty 6 Bioinformatics

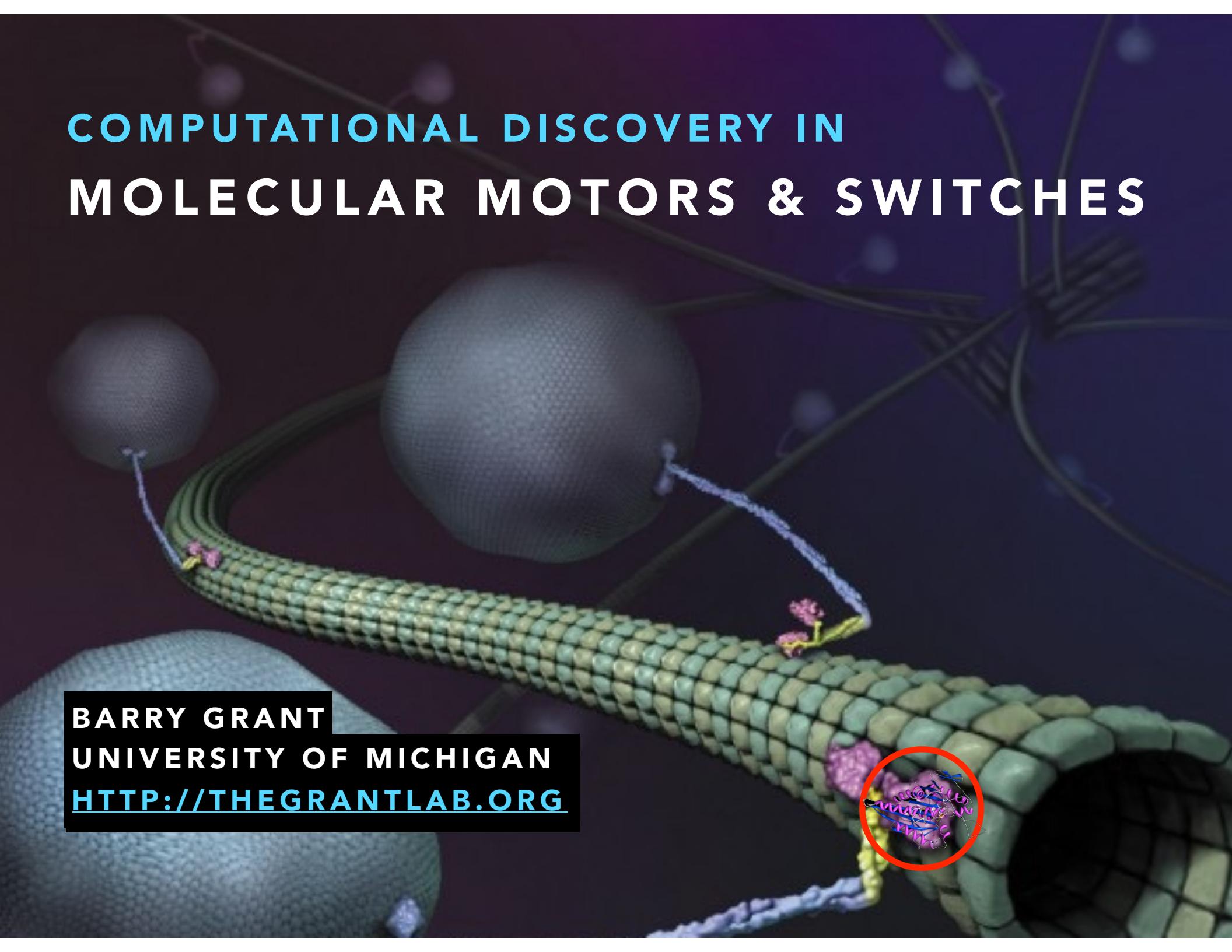


# COMPUTATIONAL DISCOVERY IN MOLECULAR MOTORS & SWITCHES



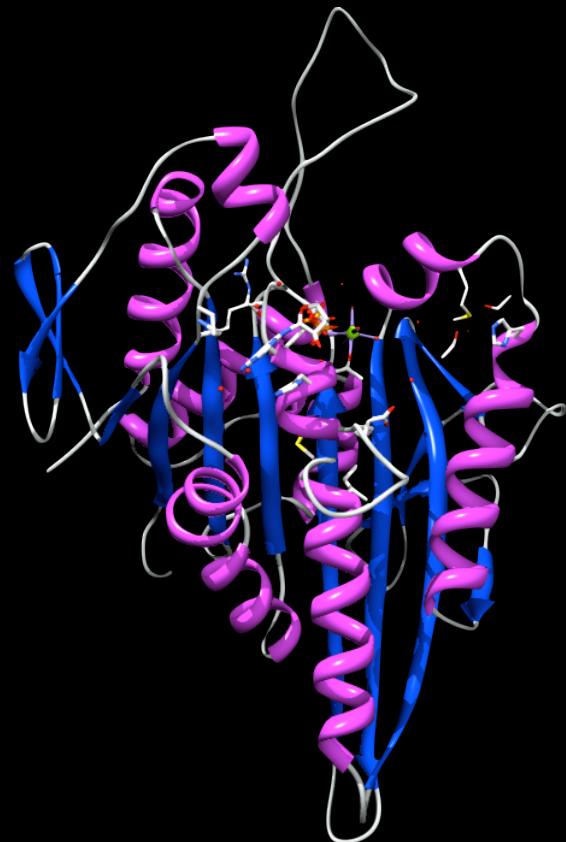
BARRY GRANT  
UNIVERSITY OF MICHIGAN  
[HTTP://THEGRANTLAB.ORG](http://THEGRANTLAB.ORG)

# COMPUTATIONAL DISCOVERY IN MOLECULAR MOTORS & SWITCHES

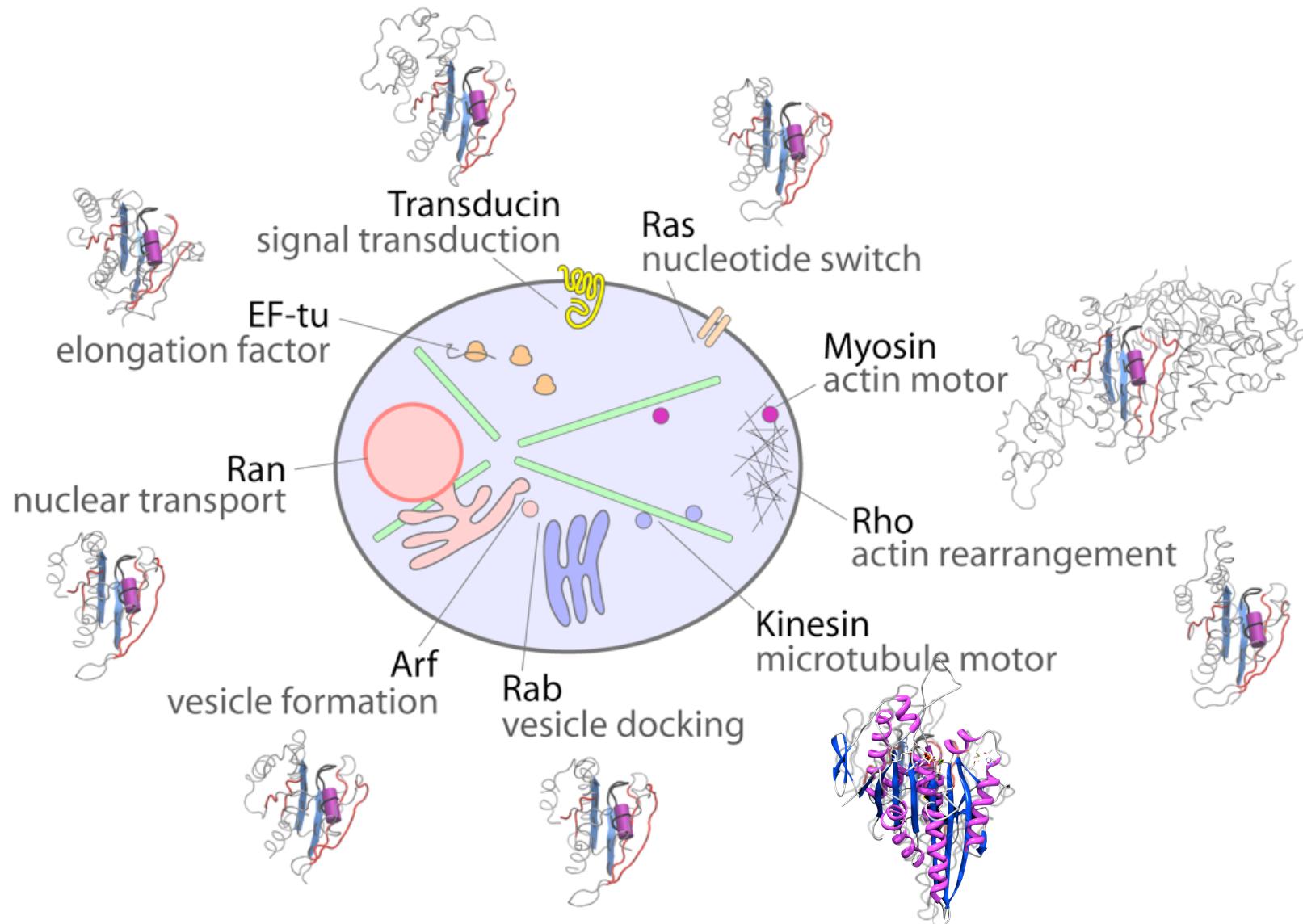


BARRY GRANT  
UNIVERSITY OF MICHIGAN  
[HTTP://THEGRANTLAB.ORG](http://THEGRANTLAB.ORG)

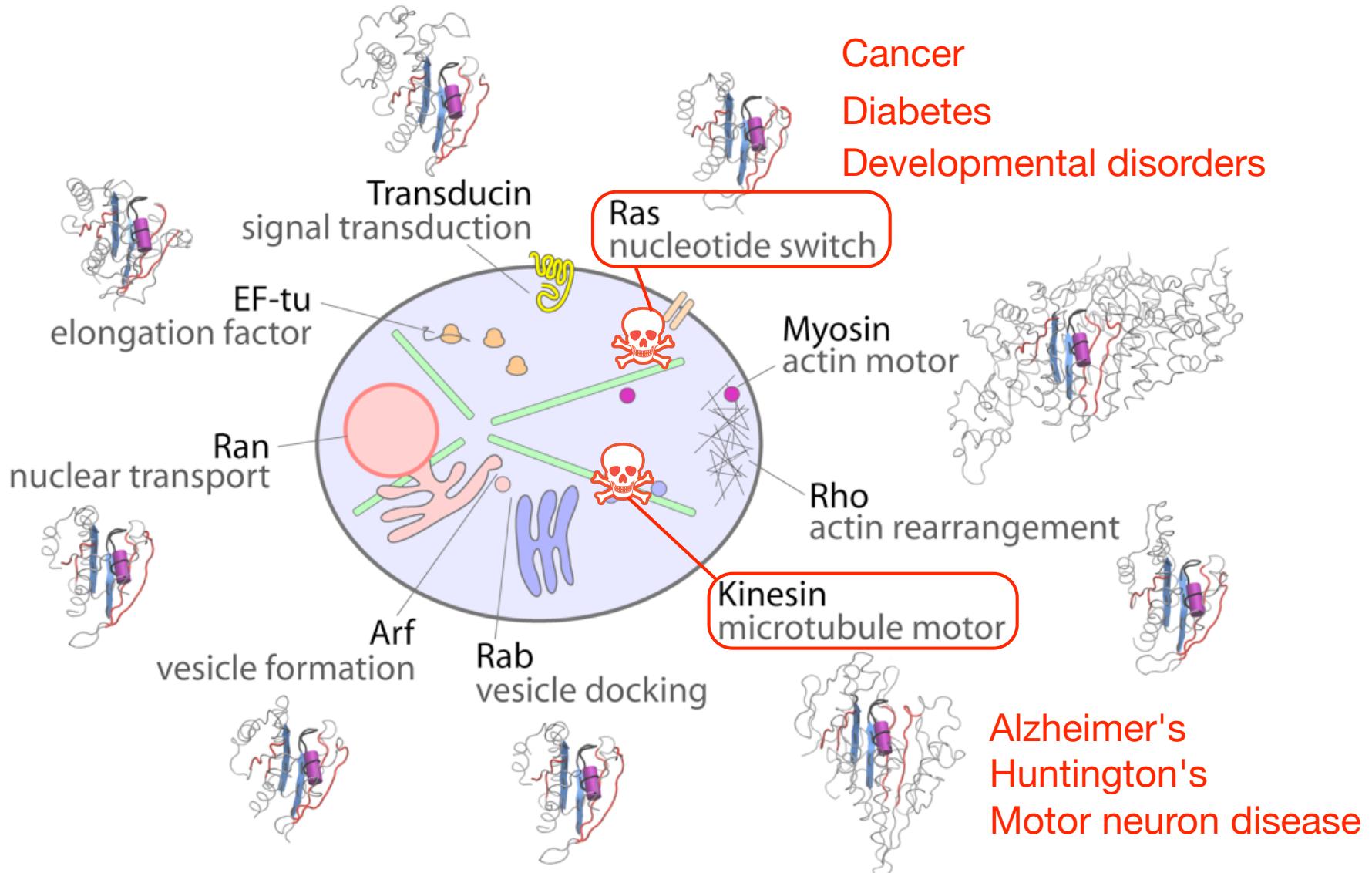
## Kinesin Motor



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

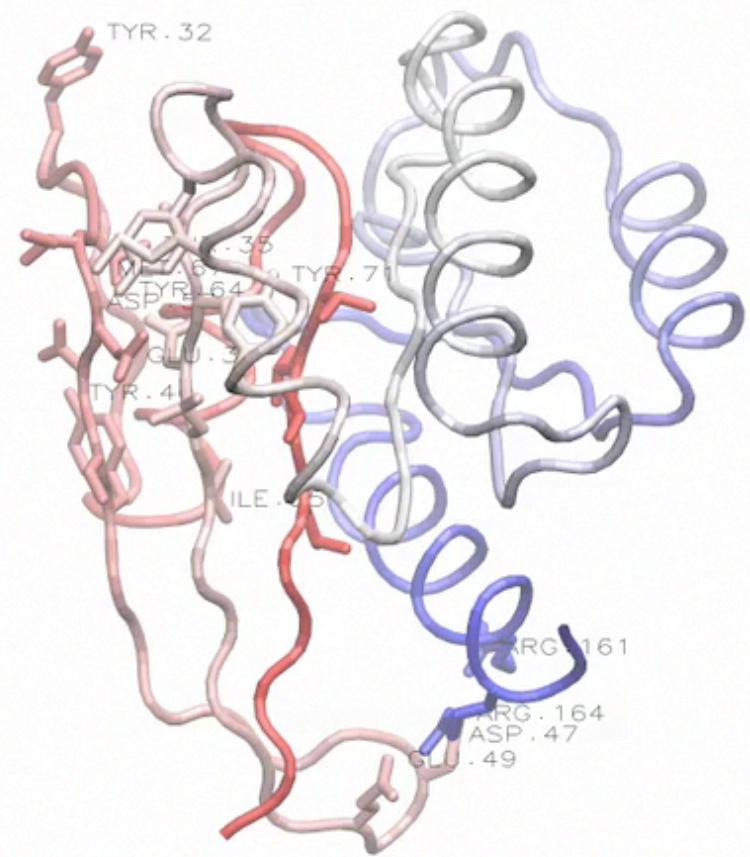
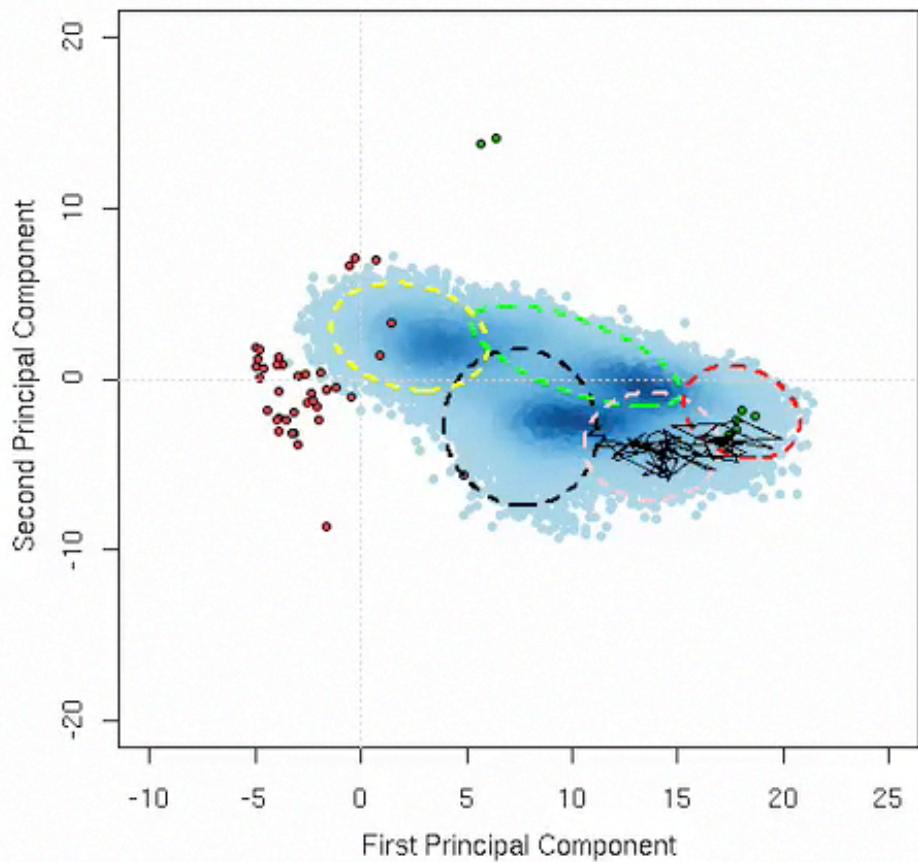


# Aberrant function of motors and switches is associated with many diseases



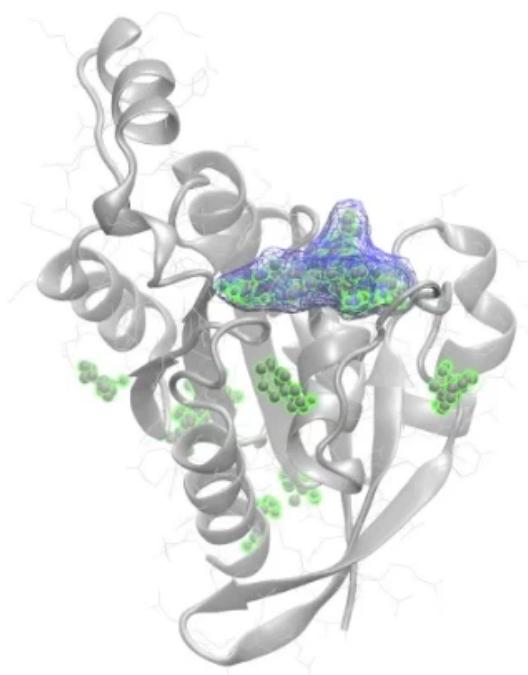
# Simulations of conformational activation

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

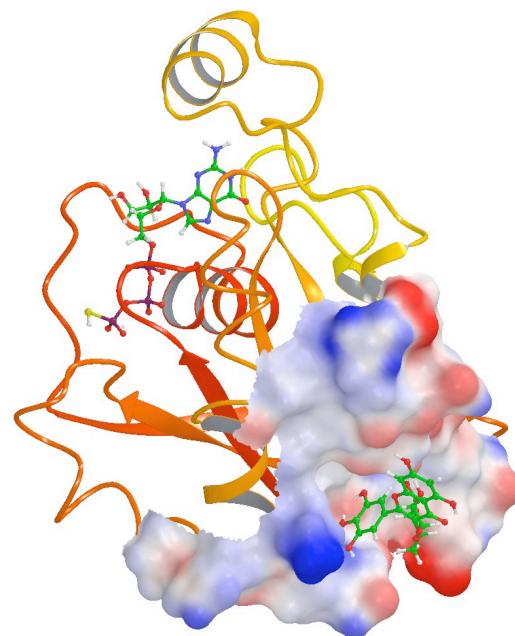


# Allosteric inhibitor design & testing

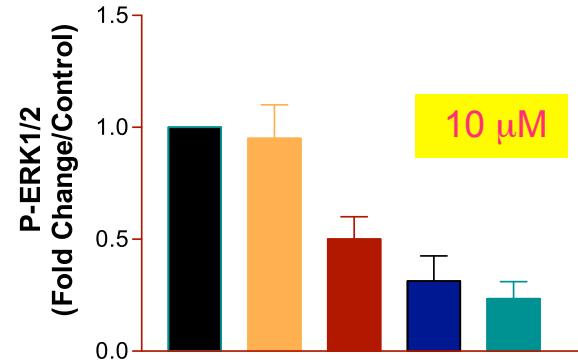
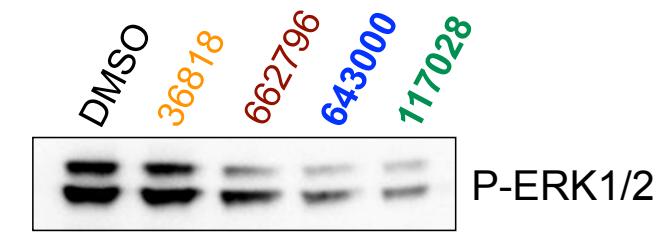
Computational methods can identify novel transient binding pockets & cognate inhibitors that may offer new avenues for future therapeutics



Ensemble binding site  
mapping

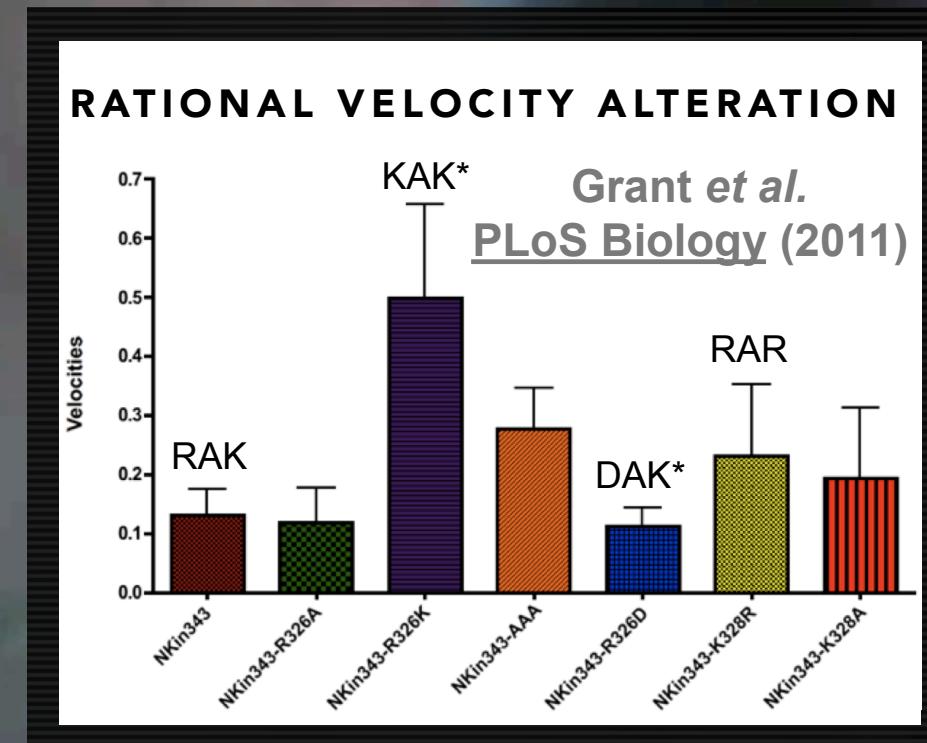


Ensemble small  
molecule docking



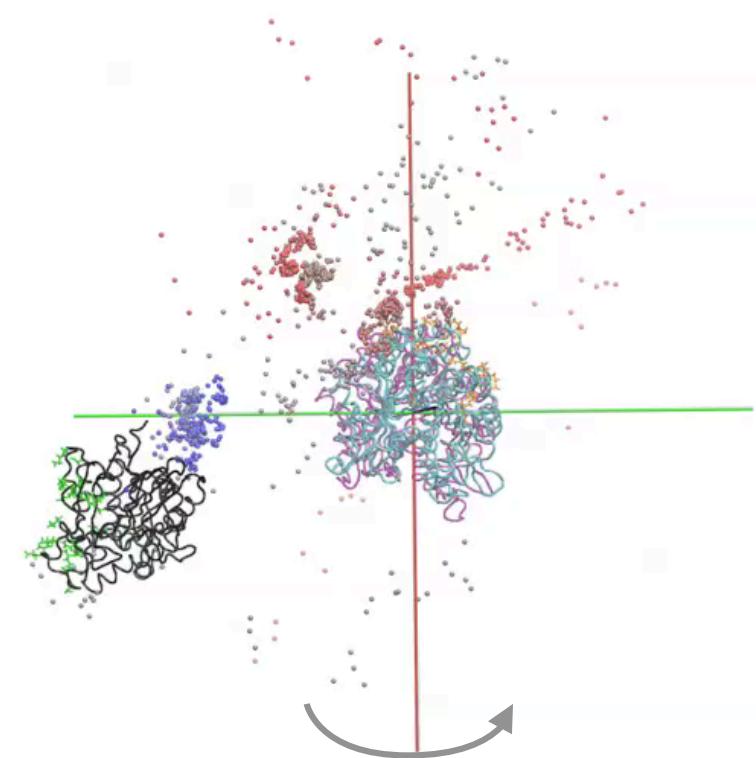
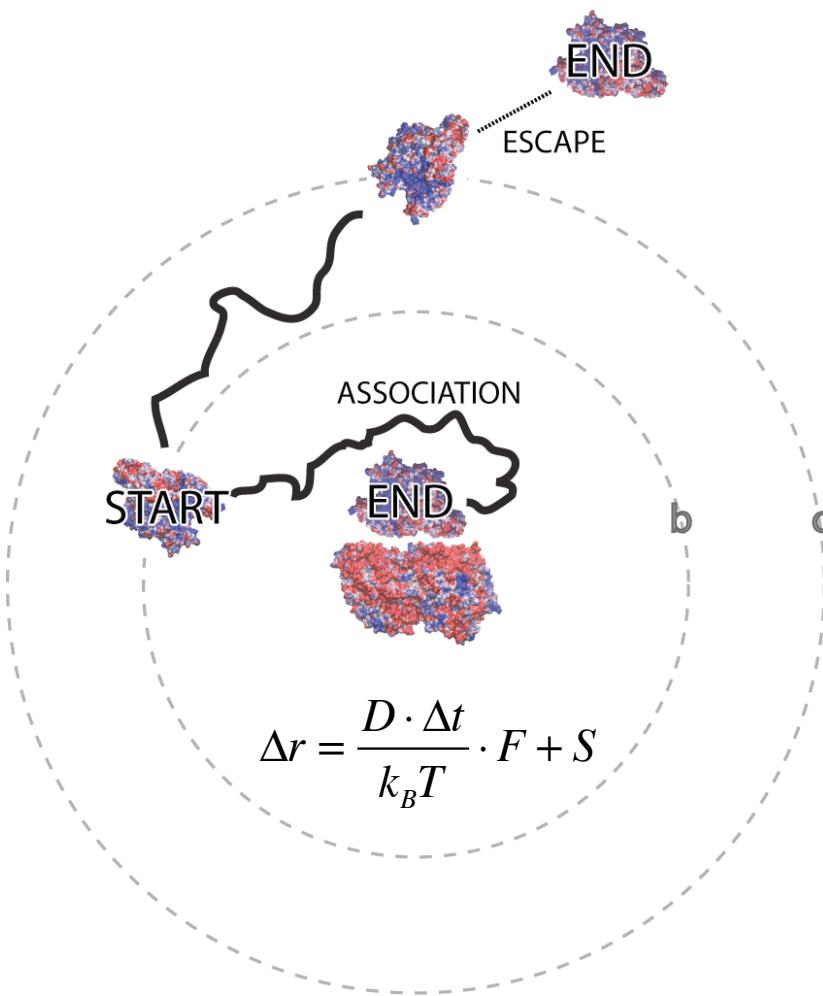
Compound testing in  
cancer cell lines

# MAPPING MOTOR PROCESSIVITY DETERMINANTS



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

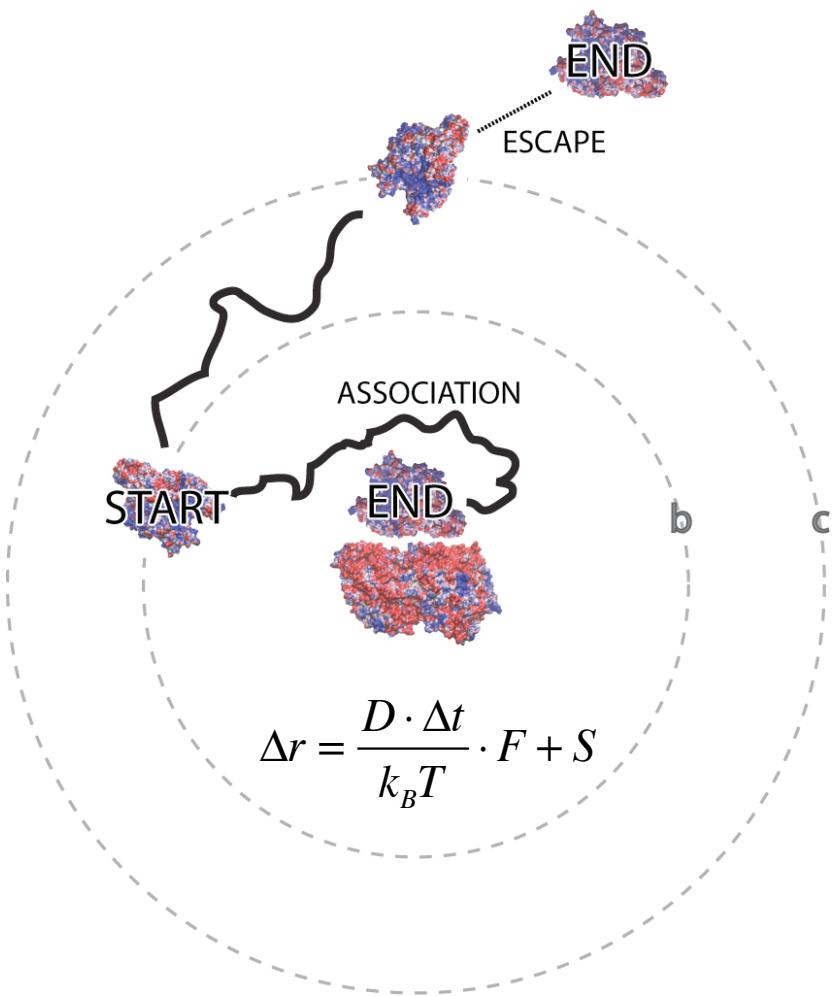
# Brownian dynamics model of kinesin-tubulin association



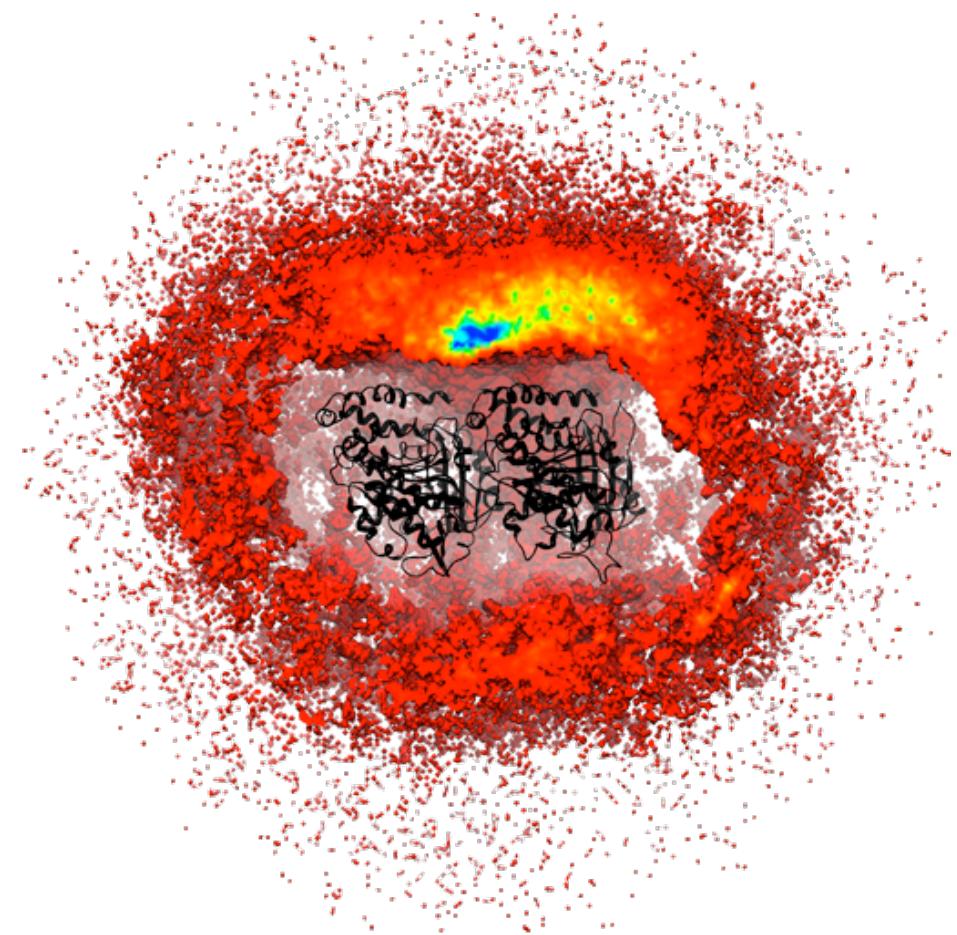
Single Trajectory

PLoS Biology (2011)

# Brownian dynamics model of kinesin-tubulin association



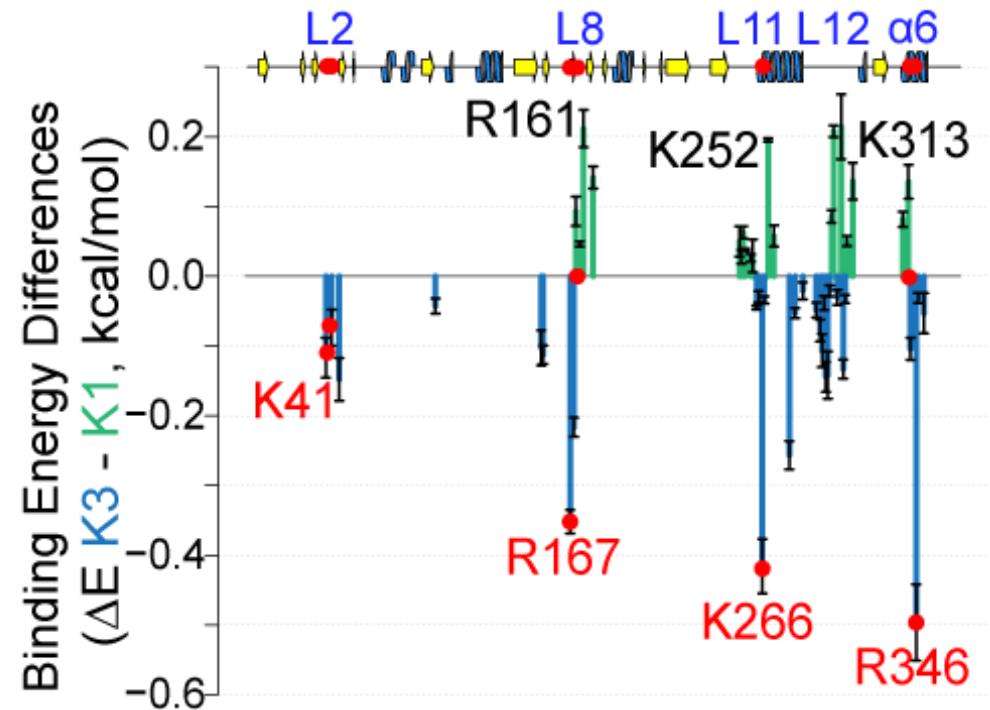
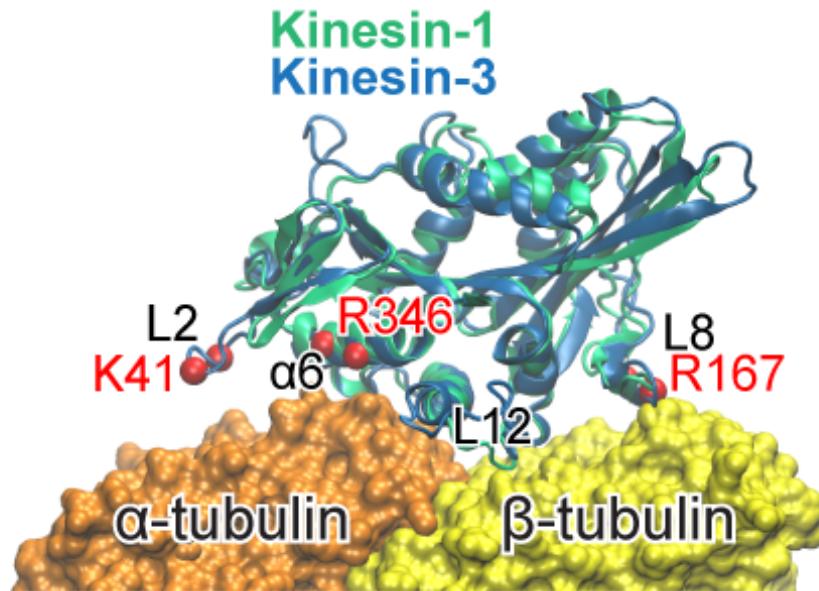
$$\Delta r = \frac{D \cdot \Delta t}{k_B T} \cdot F + S$$



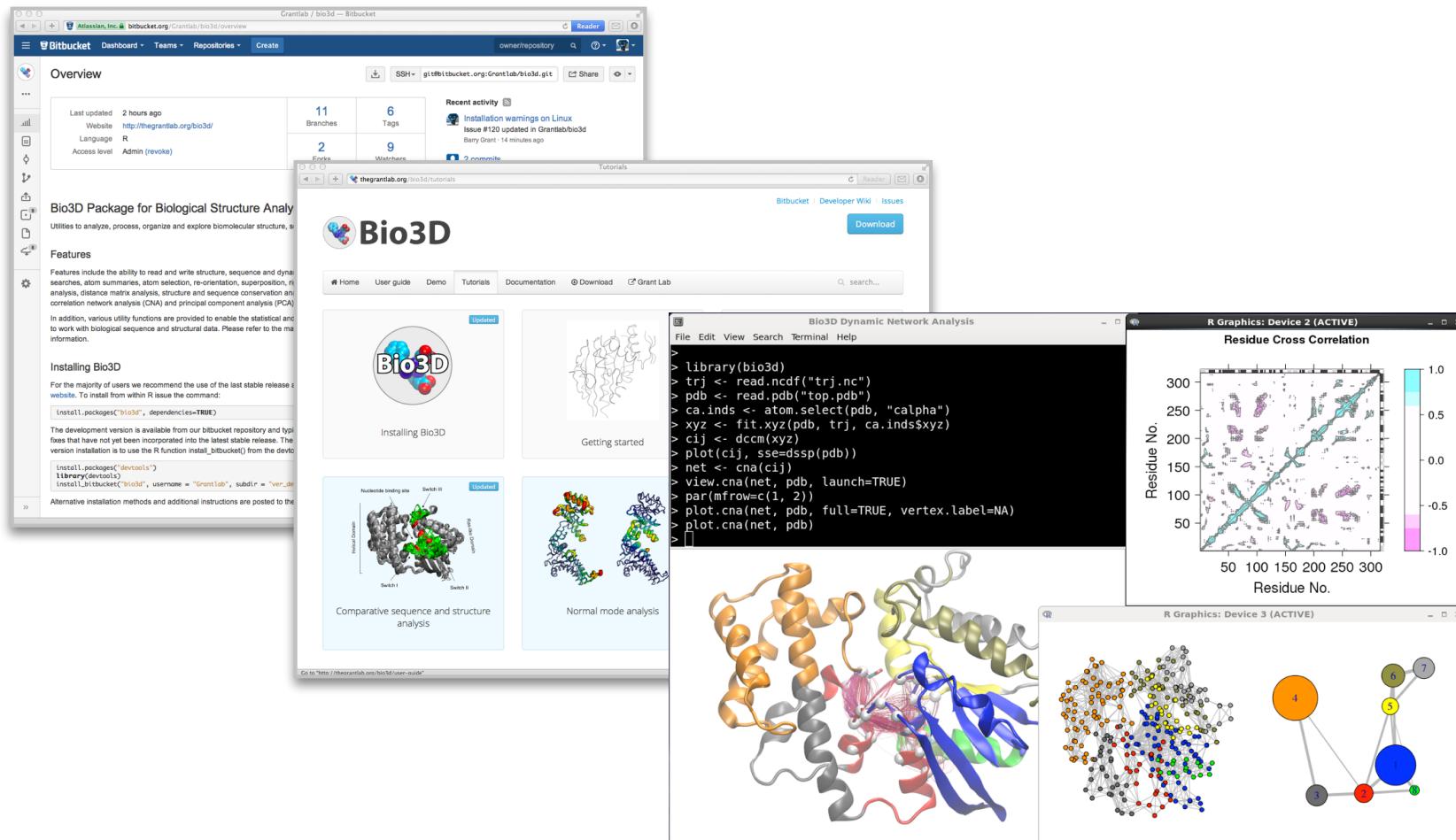
200k Trajectories

# Mapping motor processivity determinants

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

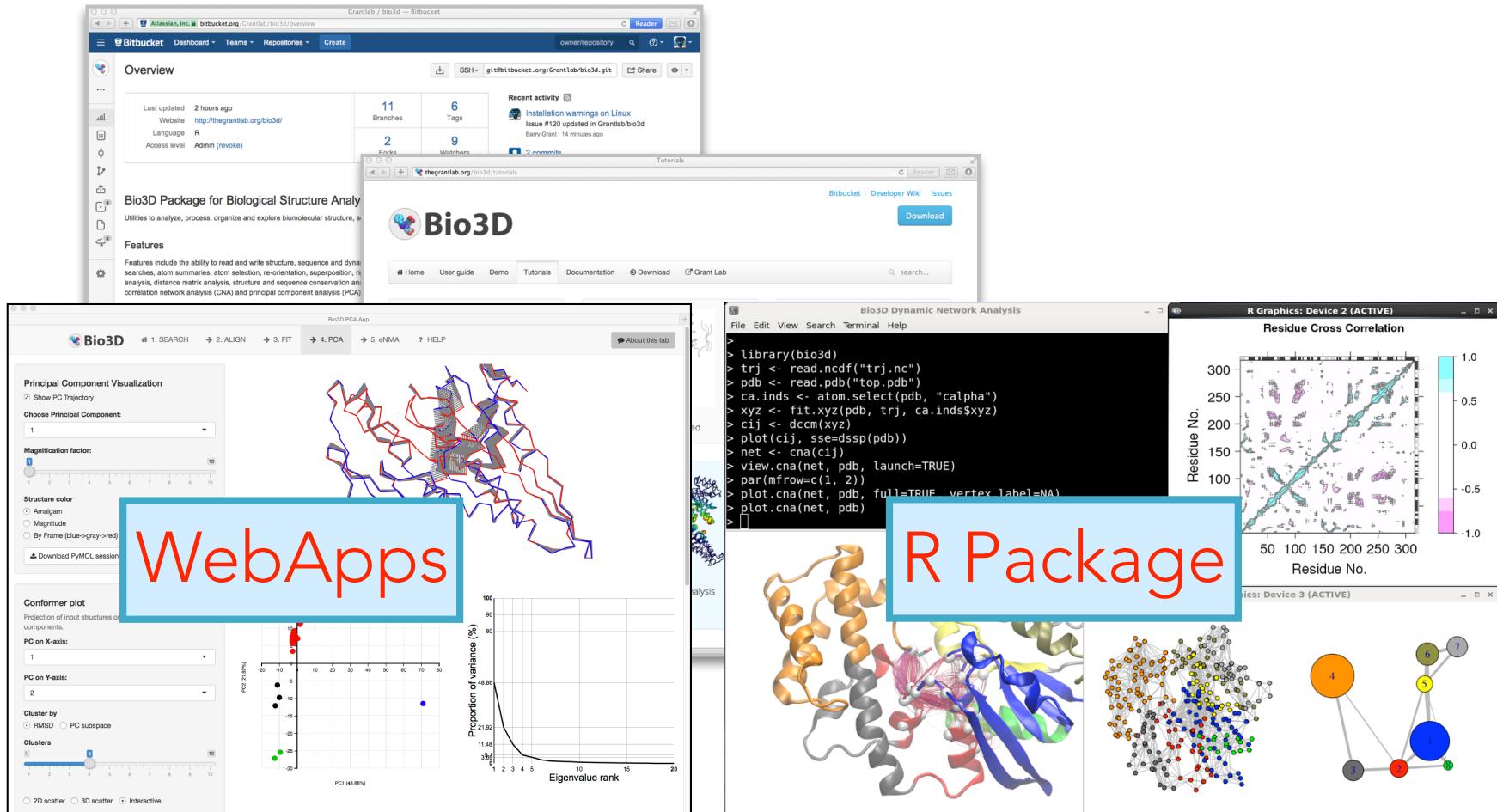


# Side Note: Integrating protein structural dynamics and evolutionary analysis with Bio3D



<http://thegrantlab.org/bio3d/>

# 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)

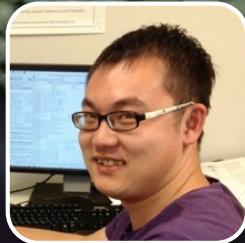
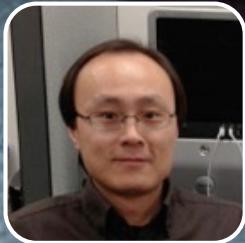
Bjoern Peters (LIAI)

**Sivaraj Sivaramakrishnan** (**UM**, CDB)

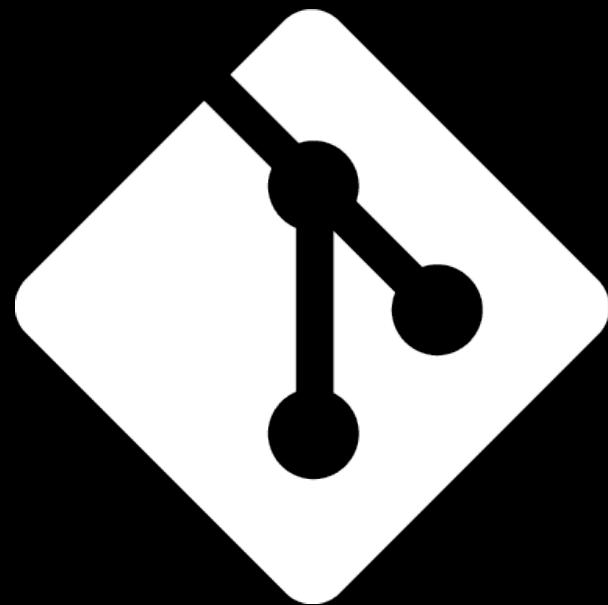
Chao-Yie Yang (**UM**, Internal Med)

**John Traynor** (**UM**, Pharmacology)

**Lars Skjærven** (U.of Bergen, Norway)



[HTTP://THEGRANTLAB.ORG](http://THEGRANTLAB.ORG)



# git

**Barry Grant**

[bjgrant@umich.edu](mailto: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

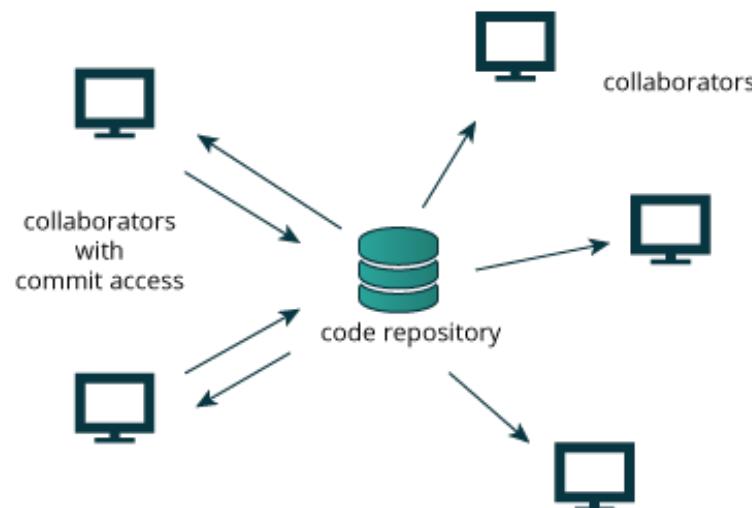
Version control systems (VCS) record changes to a file or set of files over time so that you can recall specific versions later.

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

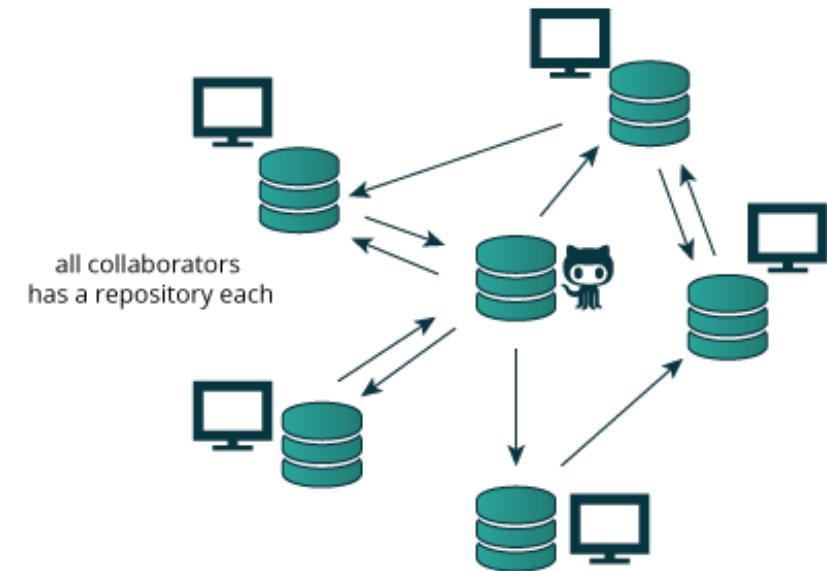
There are many VCS available, see:

[https://en.wikipedia.org/wiki/Revision\\_control](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.

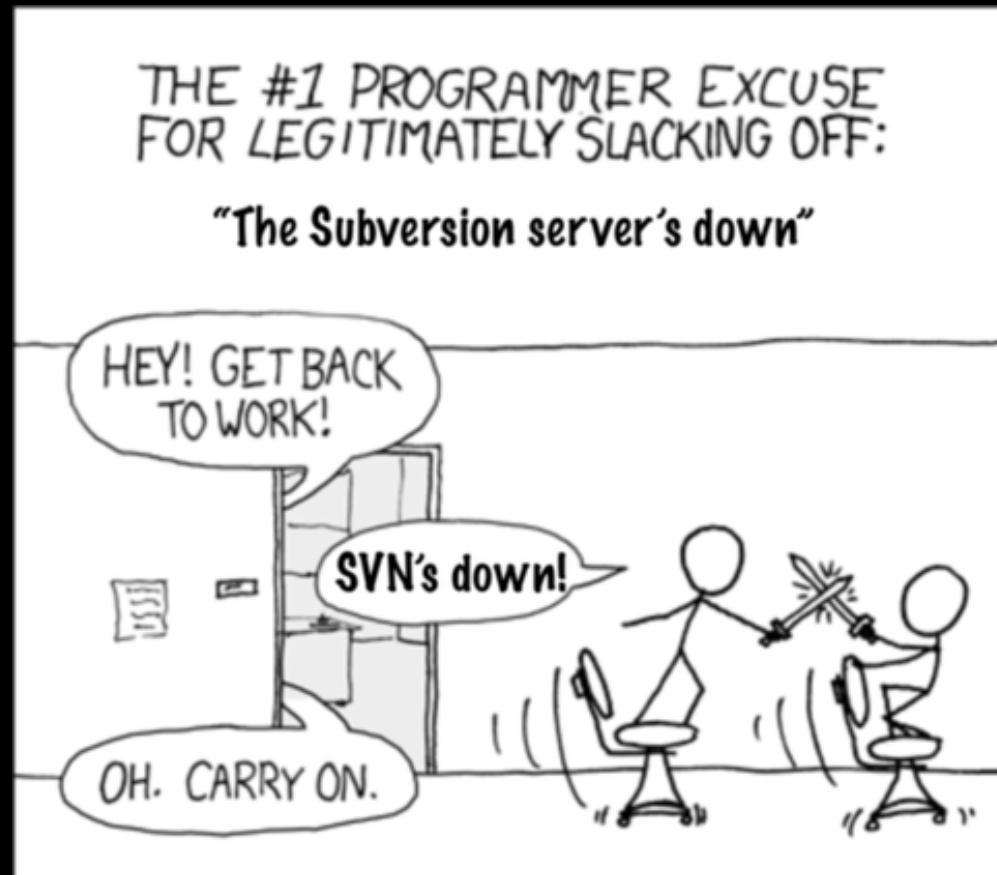
THE #1 PROGRAMMER EXCUSE  
FOR LEGITIMATELY SLACKING OFF:

"The Subversion server's down"



<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

[Create A Repo](#)[Fork A Repo](#)[Be Social](#)

Sometimes you just need a little help.



<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"

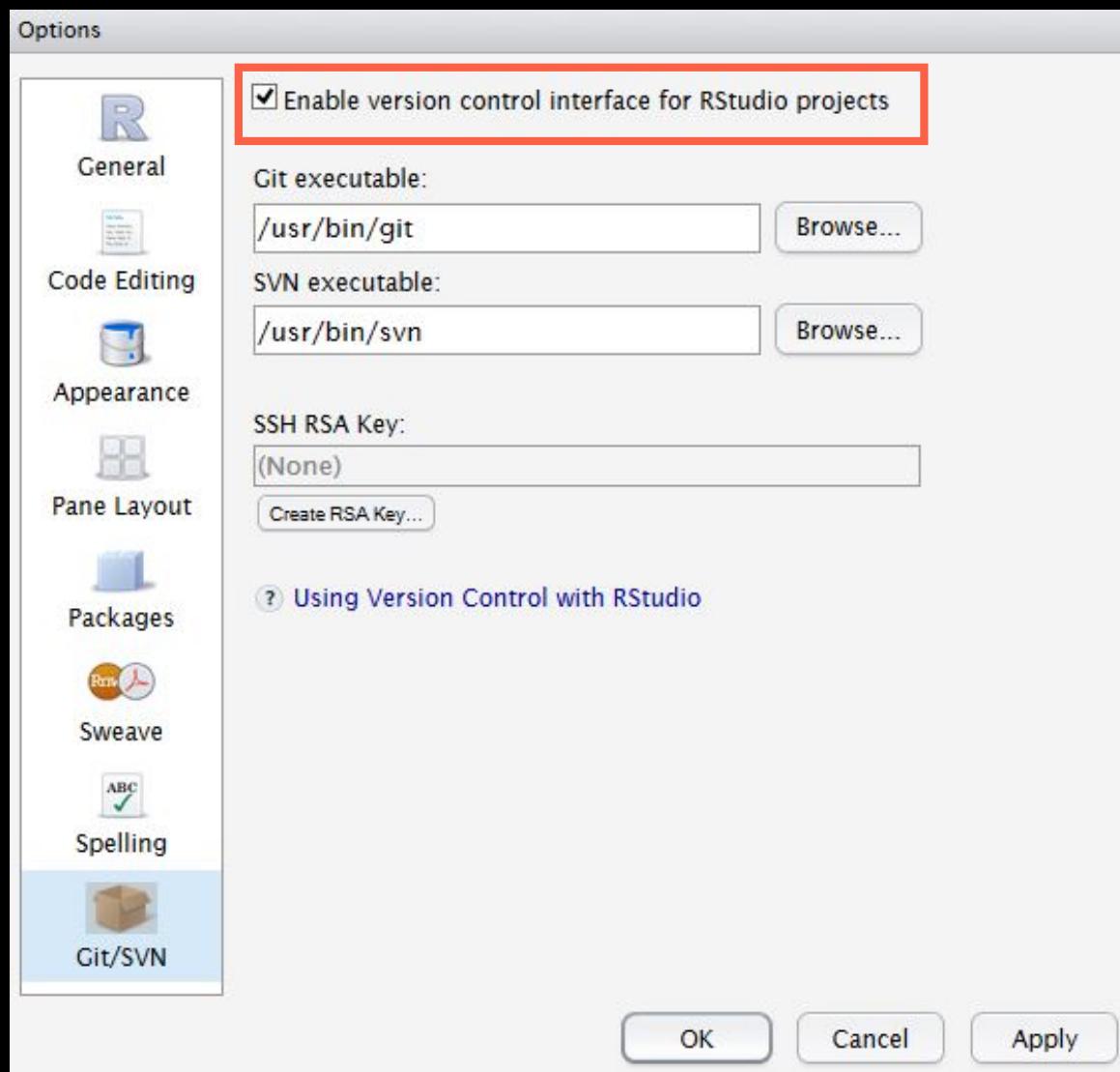
# Using Git

# 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\*.

The screenshot shows the GitHub interface. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below that is the 'GitHub Bootcamp' tutorial, which consists of four numbered cards: 1. Set up Git, 2. Create repositories, 3. Fork repositories, and 4. Work together. Each card has a small icon and a brief description. Under the tutorial, there's a list of recent activity: 'remills pushed to gh-pages at bioboot/web-2015'. At the bottom, there's a section for 'Your repositories' with a button to '+ New repository' and a link to 'web-2015'.

<https://github.com>

The screenshot shows the Bitbucket interface. At the top, there's a search bar and navigation links for 'Dashboard', 'Teams', 'Repositories', 'Snippets', and 'Create'. Below that is the 'Dashboard' section with tabs for 'Overview', 'Pull requests', 'Issues', and 'Snippets'. It shows a list of repositories: 'Grantlab / bio3d' (updated an hour ago), 'larsss / cheminf' (updated 2015-05-20), 'Grantlab / bio3d' (updated 2012-11-14), and 'bjgrant / test'. On the right, there's a 'New to Git?' section with a link to 'Learn more'. At the bottom, there's a list of recent activity: 'Memory problem while performing anal...' (commented on Grantlab/bio3d by Barry Grant), '3 commits' (Pushed to Grantlab/bio3d by Xin-Qiu Yao), and several merge commit messages for branches like 'master' and 'feature\_aanna'.

<https://bitbucket.org>

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%

wsj.com EXPAND

# THE WALL STREET JOURNAL.

Subscribe Now | Sign In  
**\$12 FOR 12 WEEKS**

Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate 

 Workers Get New Tools for Airing Their Grips  Cell Carriers Battle for Wi-Fi Airwaves  Snapchat Names ex-Mattel Exec Vollero Its Finance Chief   

**YOU ARE READING A PREVIEW OF A PAID ARTICLE.** [SUBSCRIBE NOW](#) **TO GET MORE GREAT CONTENT.**

**TECH**

## GitHub Raises \$250 Million at \$2 Billion Valuation

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

 3234  433     



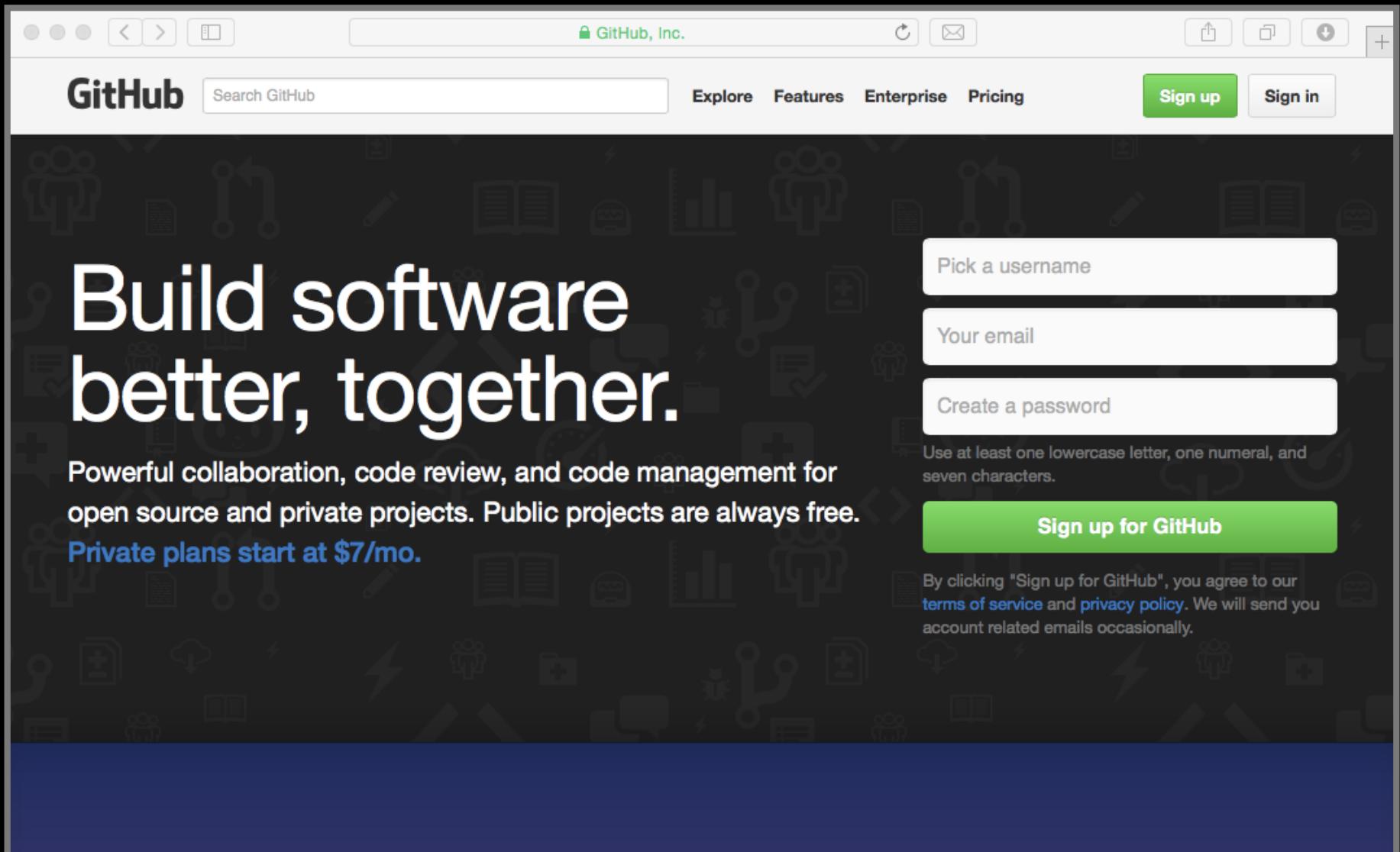
**Analytics**  
How does your organization's talent measure up to its technology?  
[Read the MIT Sloan report](#) 

# 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>**



A screenshot of the GitHub sign-up page. The page features a dark background with various white icons related to software development and collaboration. In the center, there is a large white text area containing the slogan "Build software better, together." Below this, there is descriptive text about GitHub's features and pricing. On the right side, there is a form for creating a new account, consisting of three input fields: "Pick a username", "Your email", and "Create a password". Below these fields is a note about password requirements: "Use at least one lowercase letter, one numeral, and seven characters." At the bottom of the form is a large green button labeled "Sign up for GitHub". Above the form, there is a navigation bar with links for "Explore", "Features", "Enterprise", and "Pricing", along with "Sign up" and "Sign in" buttons. The top of the page shows the GitHub logo and a search bar.

**GitHub** Search GitHub

Explore Features Enterprise Pricing

Sign up Sign in

# Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free.

Private plans start at \$7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

# Pick the FREE plan!

The screenshot shows the GitHub 'Welcome to GitHub' page. At the top, there are three steps: 'Completed' (Set up a personal account), 'Step 2: Choose your plan' (highlighted in blue), and 'Step 3: Go to your dashboard'. Below this, a section titled 'Choose your personal plan' lists five plans: Large (\$50/month, 50 repos), Medium (\$22/month, 20 repos), Small (\$12/month, 10 repos), Micro (\$7/month, 5 repos), and Free (\$0/month, 0 repos). A red circle highlights the 'Chosen' button for the Free plan. To the right, a sidebar lists 'Each plan includes:' with features like Unlimited collaborators, Unlimited public repositories, Free setup, HTTPS Protection, Email support, and Wikis, Issues, Pages, & more.

Plan	Cost	Private repositories	Action
Large	\$50/month	50	Choose
Medium	\$22/month	20	Choose
Small	\$12/month	10	Choose
Micro	\$7/month	5	Choose
Free	\$0/month	0	Chosen

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.  
Don't worry, you can cancel or upgrade at any time.

# Your GitHub homepage

Check your email for verification request

The screenshot shows a GitHub user profile for 'biobootStudent'. The profile picture is a green pixelated version of the Minecraft Creeper. The user joined on August 26, 2015, and has 0 Followers, 0 Starred, and 0 Following. A 'Pro tip' box suggests updating profile details. Below is a 'Contributions' graph for September, showing a sparse grid of squares. A summary explains the contribution graph system, mentioning pull requests, issues, and commits, and links to a 'Hello World' guide.

Pro tip: updating your profile with your name, location, and a profile picture helps other GitHub users get to know you. [Edit profile](#)

Contributions

Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug

M  
W  
F

Summary of pull requests, issues opened, and commits. Learn how we count contributions. [Less](#) [More](#)

This is your **contribution graph**. When you make a commit to a repository, you'll get a for that day. Make more contributions and you'll get a darker green square. Over time, your chart might start looking [something like this](#).

We have a quick guide that will show you how to create your first repository. You'll also make a commit and [earn your first green square!](#)

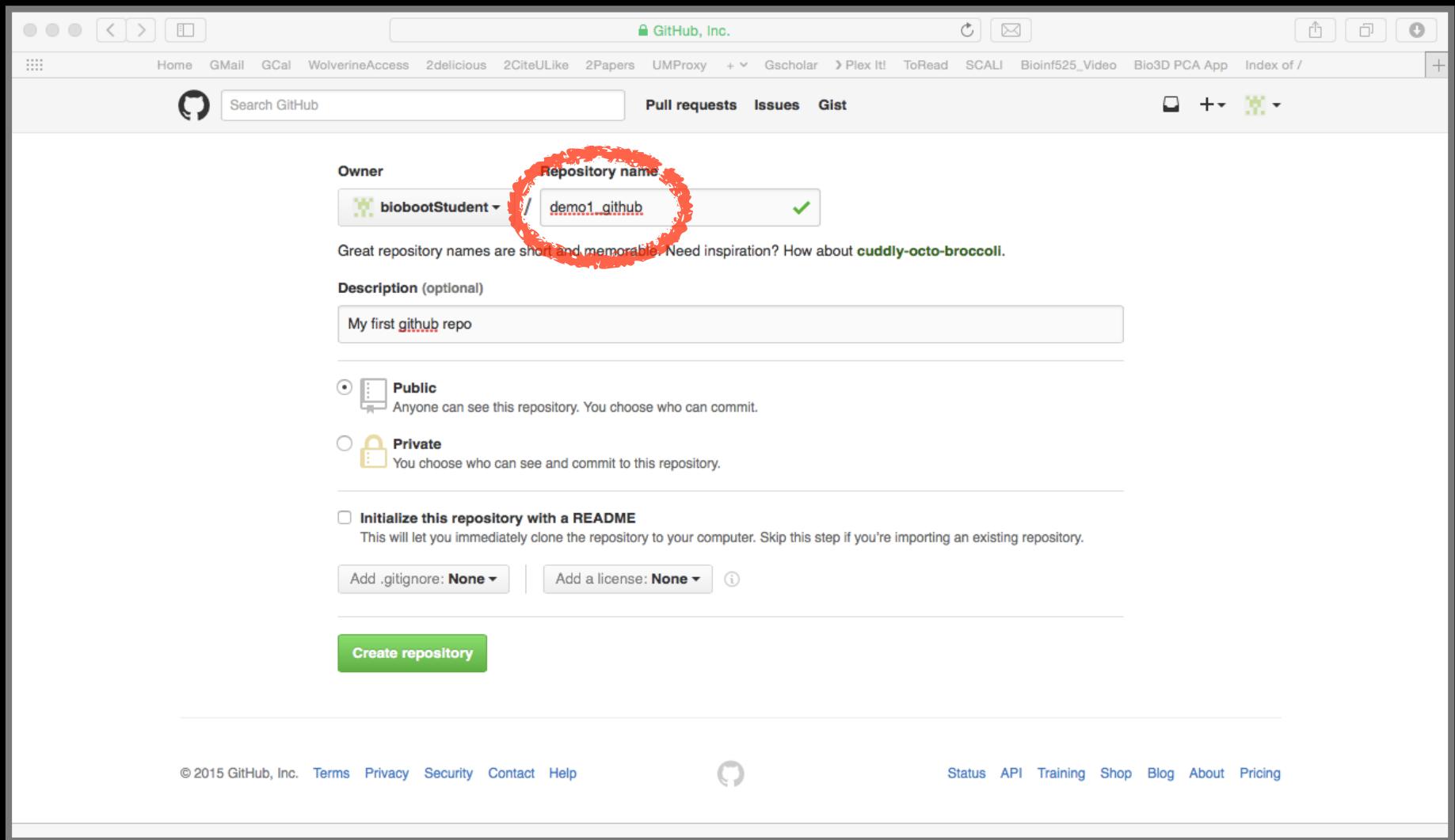
[Read the Hello World guide](#)

# Skip the hello-world tutorial

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

The screenshot shows the GitHub homepage. At the top, there's a navigation bar with links like Home, GMail, GCal, WolverineAccess, 2delicious, 2CiteULike, 2Papers, UMPProxy, +, Gscholar, Plex It!, ToRead, SCALI, Bioinf525\_Video, Bio3D, and Apps. Below the navigation is a search bar with the GitHub logo and a 'Pull requests' button. A message 'Your email was verified.' is displayed. A prominent feature is a dashed box containing the text 'Learn Git and GitHub without any code!' followed by instructions: 'Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.' A green 'Let's get started!' button is below this. On the left, a user profile for 'biobootStudent' is shown with a welcome message: 'Welcome to GitHub! What's next? (3 hours ago)'. Below it are links: 'Create a repository', 'Tell us about yourself', 'Browse interesting repositories', and 'Follow @github on Twitter'. A 'ProTip!' link is also present. On the right, there's a section for 'Your repositories' which says 'You don't have any repositories yet!' and provides links to 'Create your first repository' and 'learn more about Git and GitHub'. A 'ProTip!' box states: 'Feline cephalopod adhesives are great for decorating portable computation devices.' At the bottom, there's a link to 'Subscribe to your news feed'.

# Name your repo test



The screenshot shows the GitHub repository creation interface. At the top, there is a navigation bar with links like Home, GMail, GCal, WolverineAccess, etc. Below the navigation bar is a search bar and a header with tabs for Pull requests, Issues, and Gist.

The main form starts with the "Owner" section, which shows the user's GitHub handle "biobootStudent" followed by a dropdown menu and a repository name input field. The input field contains "demo1\_github" and has a green checkmark icon to its right. This entire input field is circled in red.

Below the owner section is a descriptive message: "Great repository names are short and memorable. Need inspiration? How about [cuddly-octo-broccoli](#)."

The next section is "Description (optional)", containing the text "My first github repo".

There are two radio button options for repository visibility: "Public" (selected) and "Private". The "Public" option is described as "Anyone can see this repository. You choose who can commit." The "Private" option is described as "You choose who can see and commit to this repository."

A checkbox labeled "Initialize this repository with a README" is present, with the note "This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository." Below this are buttons for "Add .gitignore: None" and "Add a license: None".

At the bottom of the form is a large green "Create repository" button.

At the very bottom of the page, there is a footer with links: © 2015 GitHub, Inc. Terms Privacy Security Contact Help, Status API Training Shop Blog About Pricing, and a GitHub logo.

# Your Turn:

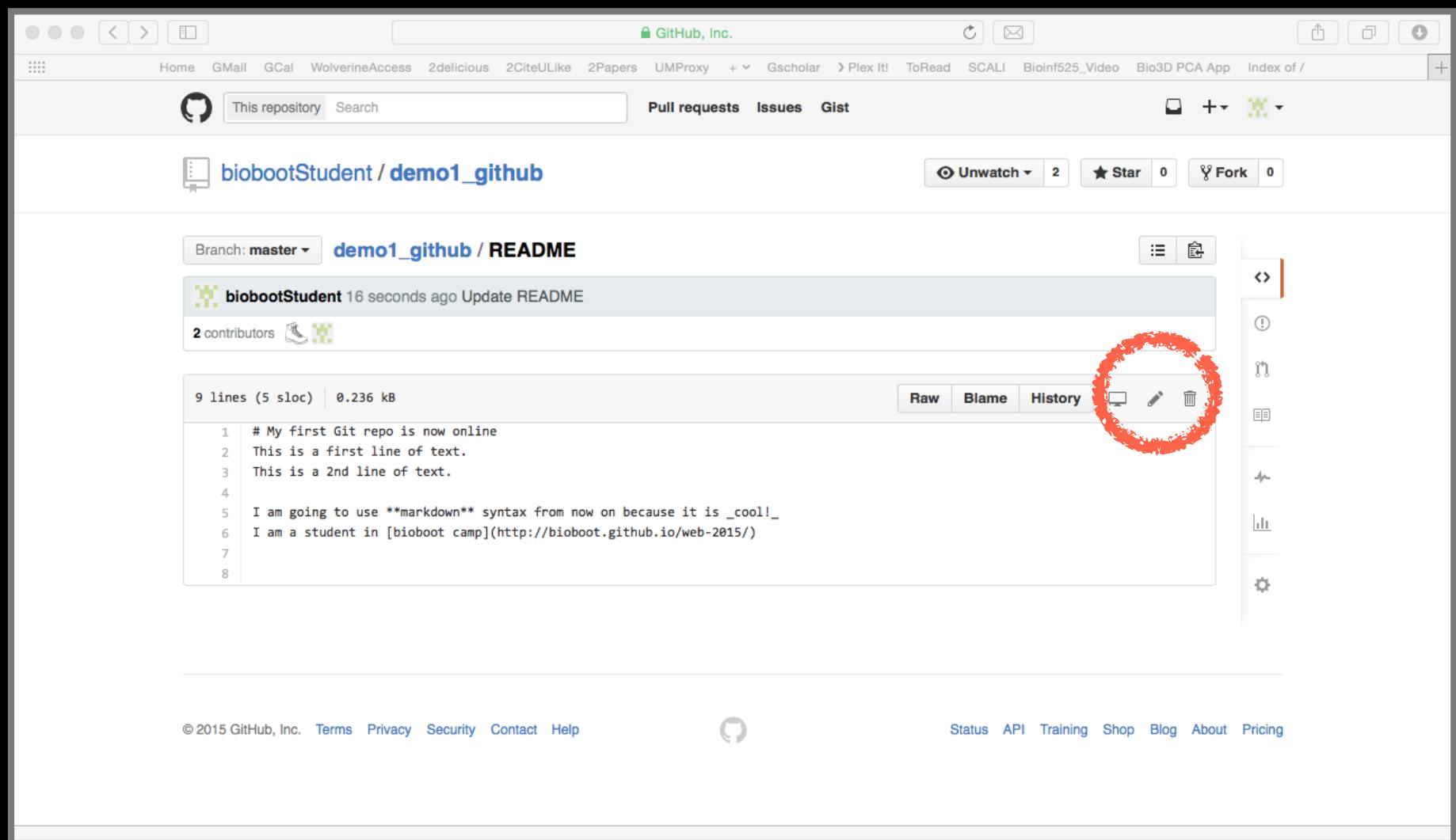
<http://tinyurl.com/rclass-github>

Step 6 only please!

(We have already done steps 1 to 5)

# Side-note: How to edit online

## Specifically lets add some Markdown content



A screenshot of a web browser displaying a GitHub repository page. The URL in the address bar is [https://github.com/biobootStudent/demo1\\_github](#). The page shows the **demo1\_github / README** file. The content of the README is:

```
1 # My first Git repo is now online
2 This is a first line of text.
3 This is a 2nd line of text.
4
5 I am going to use **markdown** syntax from now on because it is _cool_!
6 I am a student in [bioboot camp](http://bioboot.github.io/web-2015/)
```

The edit icon (a pencil icon) in the top right corner of the code preview area is circled in red.

At the bottom of the page, there is footer text: © 2015 GitHub, Inc. Terms Privacy Security Contact Help and Status API Training Shop Blog About Pricing.

# 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?
- Archaeological “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!