



Quantitative Insights Into Microbial Ecology

Mahidol University QIIME Workshop

December 14th-15th, 2015

William (Tony) Walters
Jai Ram Rideout

Agenda

<http://bit.ly/1QfnzBb>

Getting set up for QIIME analyses

Working with AWS

- We'll learn how to:
 - Log into an AWS EC2 instance (ssh)
 - Jupyter Notebook terminal
 - Transfer files to/from EC2 (scp/sftp)
 - Cyberduck
 - Download files within EC2 (wget)
 - Basic Linux command-line crash course

More details: http://qiime.org/tutorials/working_with_aws.html

Managing EC2 Instances

- Manage EC2 instances via:
 - Amazon's web interface
 - Good for starting up single instances
 - <http://aws.amazon.com/>
 - StarCluster
 - Good for creating EC2 clusters
 - <http://star.mit.edu/cluster/>

Managing EC2 Instances

- EC2 charges hourly rate for **running** instances
- **Stop** an instance when you're not using it
 - Small storage fee for stopped instances
- **Terminate** an instance when you're permanently done with it
 - **Warning:** ALL data and configuration will be permanently lost unless you saved it on an EBS volume!
 - Make sure to copy your files somewhere before terminating!

ssh

- Use ssh to log into a remote computer
 - Available on Mac and Linux
 - Windows: use MobaXterm
- For help with ssh, run `man ssh`
- We'll use the Jupyter Notebook terminal in this workshop
 - Consistent interface across operating systems
 - Easy to access

Log into EC2

- Web browser: <ip-address>:8888
- Open a new terminal: New -> Terminal
- Move into cli directory: cd cli
- Create personal directory:
 - mkdir <first-name>-<last-name>
 - Example: mkdir jai-rideout
 - **This is your personal directory;** we'll use it in the following exercises
- Move into the new directory (cd)
 - Example: cd jai-rideout

Download files using wget

- wget: download files from the command-line
- Convenient for publicly-hosted files
- Download the following file:

```
wget http://bit.ly/1Onj3u0 -o example.txt
```

- View file contents: cat example.txt

scp/sftp

- Use scp or sftp to copy files/directories to/
from a remote computer
 - Available on Mac and Linux
- Cyberduck: a graphical sftp program
 - Available on Mac and Windows
- WinSCP: another graphical program for
Windows

Cyberduck

- Download and install Cyberduck
 - <https://cyberduck.io/>
- I'll show you how to connect to EC2 and navigate to your personal directory

More details:

[http://qiime.org/tutorials/working with aws.html](http://qiime.org/tutorials/working_with_aws.html)

Copy files from EC2

- Copy the example file from EC2 with Cyberduck
 - Double-click example.txt
 - Will be put in your Downloads folder
 - OR: Click and drag the file where you want it saved
- Now open the file, make some changes, and save it

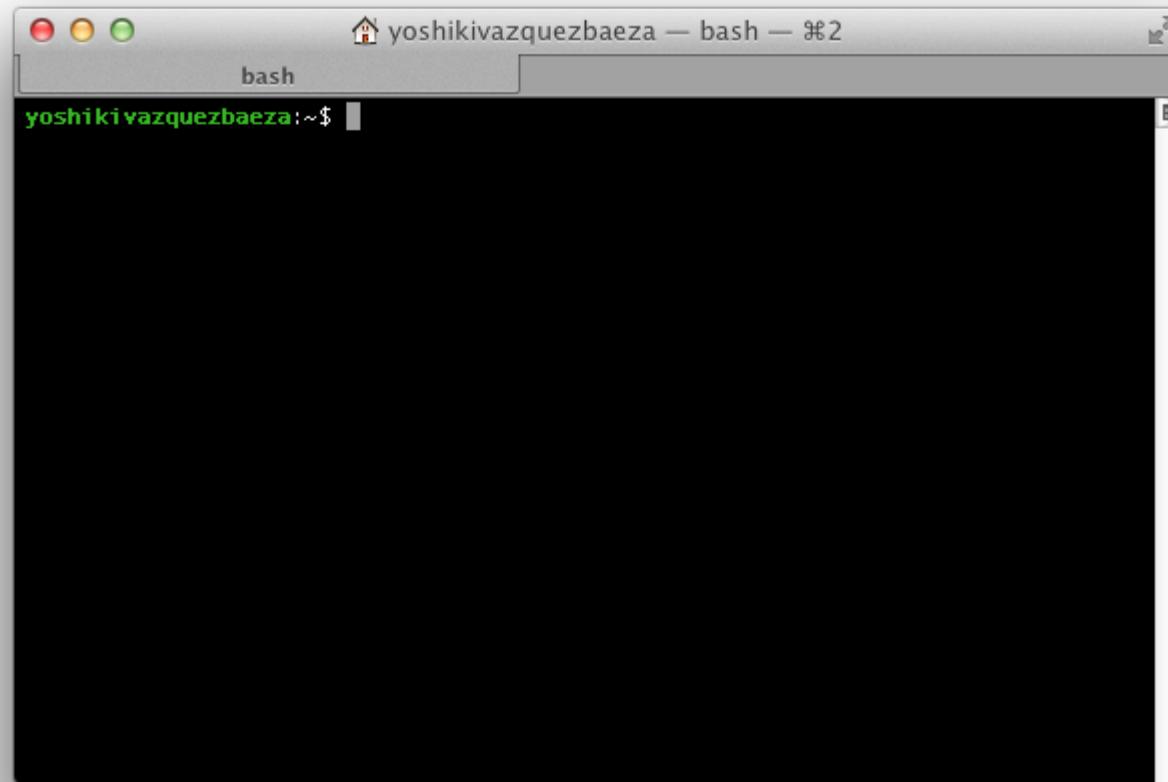
Copy files to EC2

- Copy the example file to EC2 with Cyberduck
 - Navigate to your personal directory in Cyberduck
 - Drag and drop the file to upload it
- In your EC2 terminal, view the new contents of the file: `cat example.txt`

Command line crash course

Talking to UNIX

- Through a terminal emulator



more specifically using a shell ...

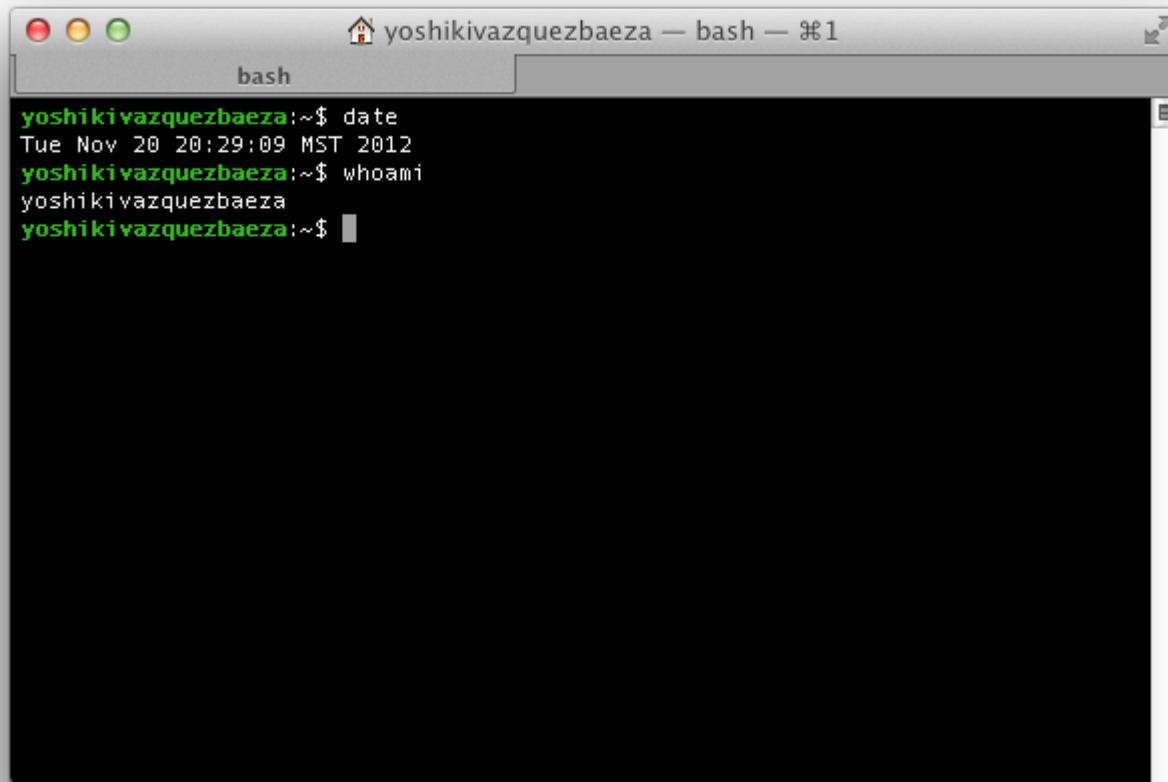
Launching a terminal window

- In Mac OS X in a Finder window or the Desktop:
 - command + shift + u
 - Search for the “Terminal”
- Ubuntu
 - In the sidebar search for the terminal icon



How do you talk to UNIX?

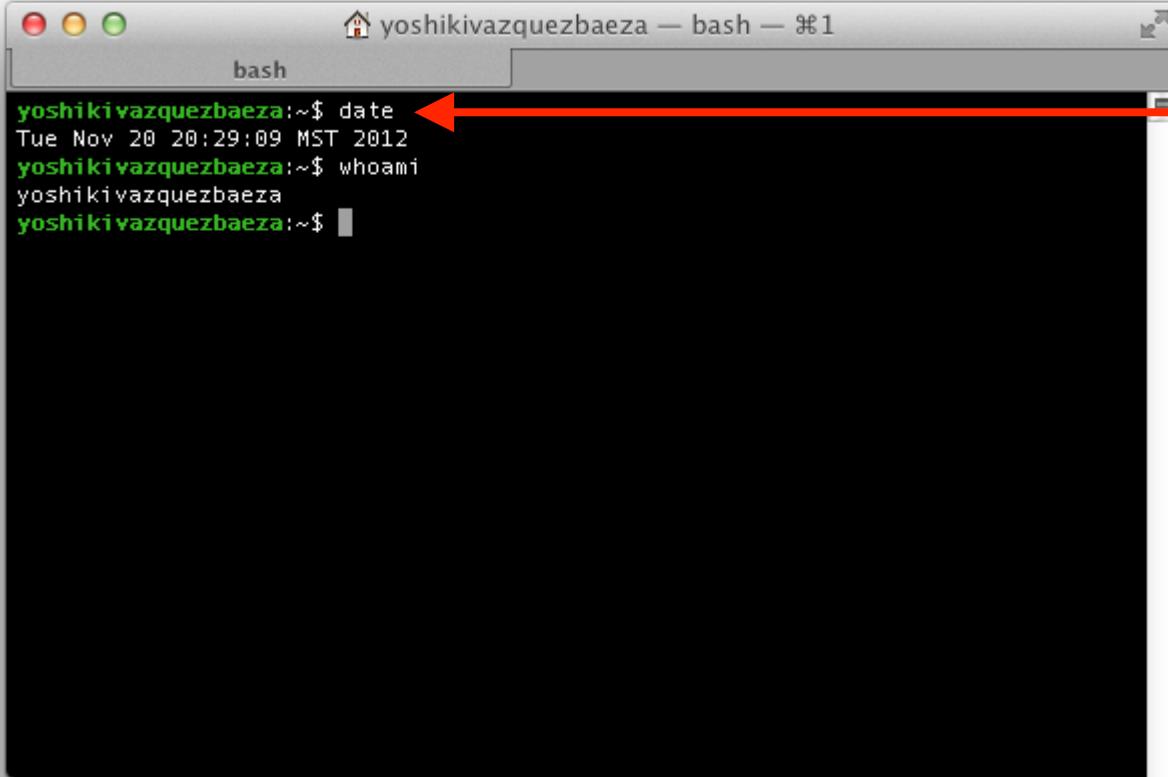
- Using some commands



```
yoshikivazquezbaeza:~$ date
Tue Nov 20 20:29:09 MST 2012
yoshikivazquezbaeza:~$ whoami
yoshikivazquezbaeza
yoshikivazquezbaeza:~$
```

How do you talk to UNIX?

- Get the current date and hour



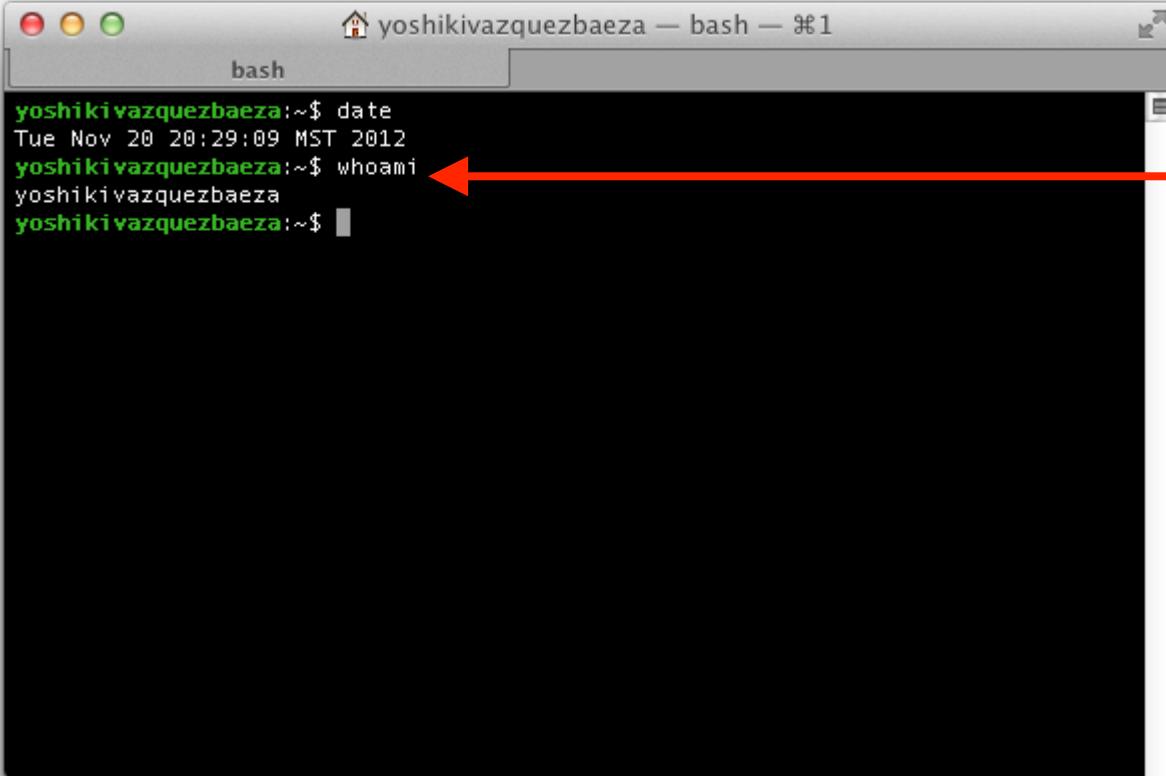
```
yoshikivazquezbaeza:~$ date
Tue Nov 20 20:29:09 MST 2012
yoshikivazquezbaeza:~$ whoami
yoshikivazquezbaeza
yoshikivazquezbaeza:~$
```

date

open a terminal window and try them ...

How do you talk to UNIX?

- See what is your user name



A screenshot of a Mac OS X terminal window titled "yoshikivazquezbaeza — bash — %1". The window contains the following text:

```
yoshikivazquezbaeza:~$ date
Tue Nov 20 20:29:09 MST 2012
yoshikivazquezbaeza:~$ whoami
yoshikivazquezbaeza
yoshikivazquezbaeza:~$
```

A red arrow points from the word "whoami" on the right side of the slide to the "whoami" command in the terminal window. The word "whoami" is also underlined in red.

Some general concepts

Home Sweet Home

Can be referred to as:

\$HOME or **\$ { HOME }**

Or also as:

~

Mac OS X:

/Users/*username*/

Linux based systems:

/home/*username*/

Try:

```
echo ${HOME}  
ls $HOME  
ls ~/
```



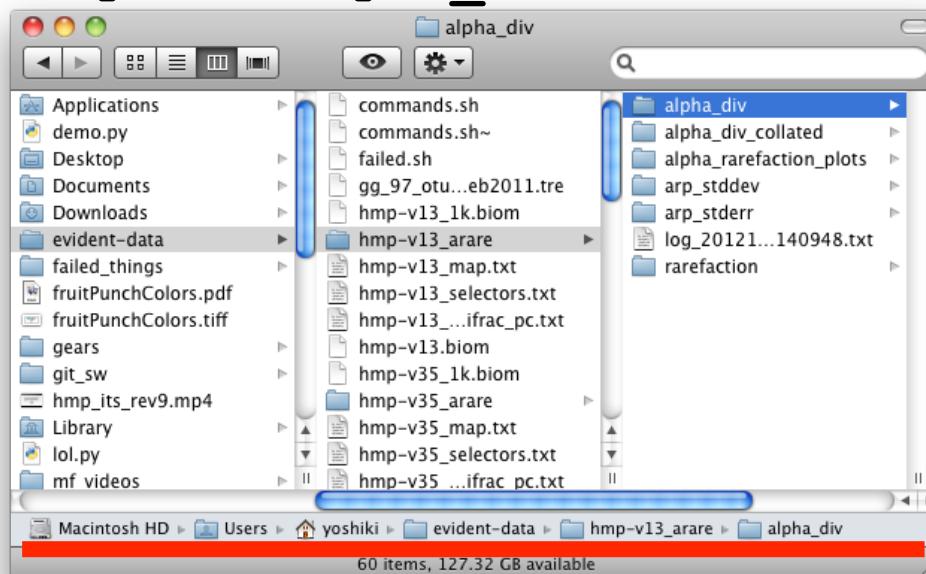
username stands for your user in your machine

Paths (absolute)

`/Users/yoshiki/evident-data/hmp-v13_arare/alpha_div`

`$HOME/evident-data/hmp-v13_arare/alpha_div`

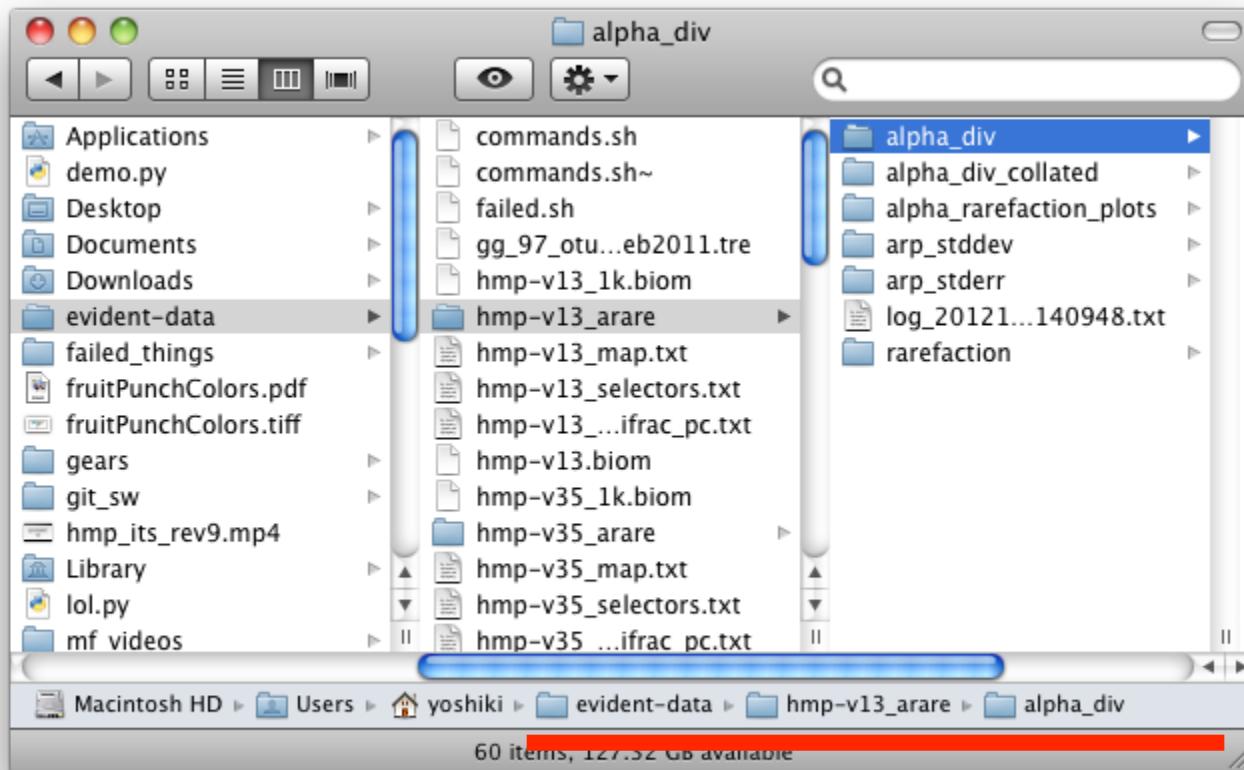
`~/evident-data/hmp-v13/alpha_div`



A **slash** at the beginning of a path denotes it as an absolute path, i. e. from the base of your hard drive.

Paths (relative)

evident-data/hmp-v13_arare/alpha_div



On the other side relative paths are not preceded by a slash

One command, different forms

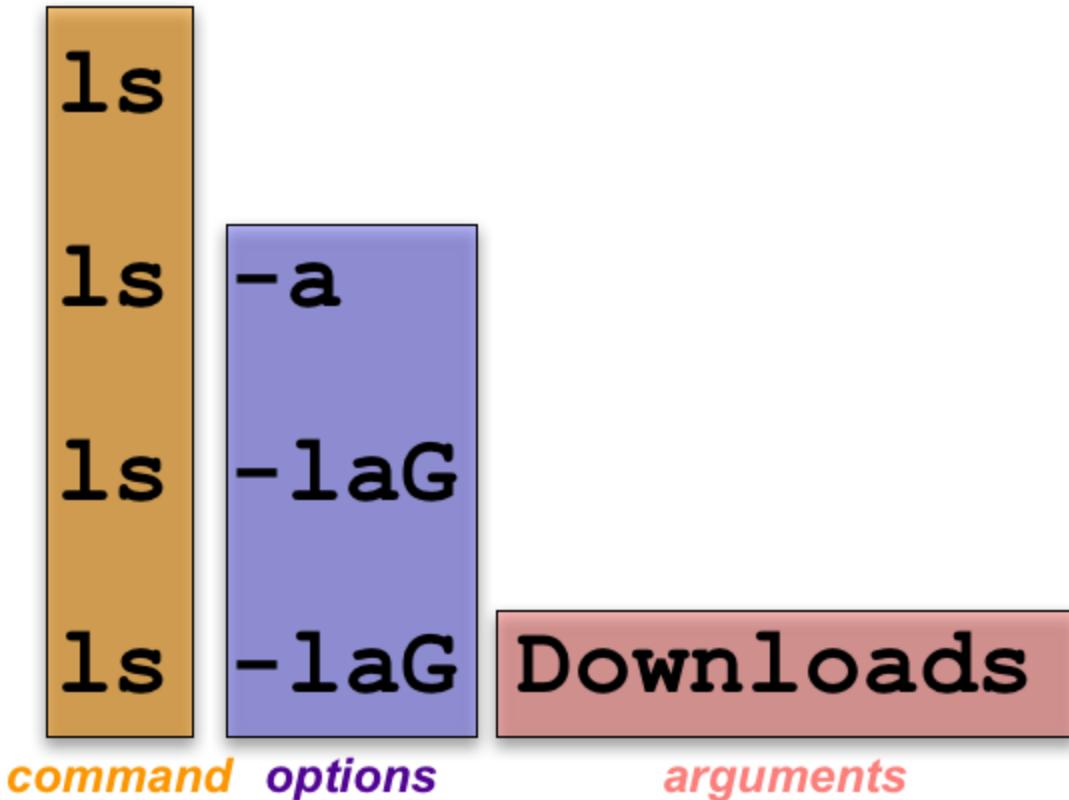
ls

ls -a

ls -laG

ls -laG Downloads

Anatomy of a command



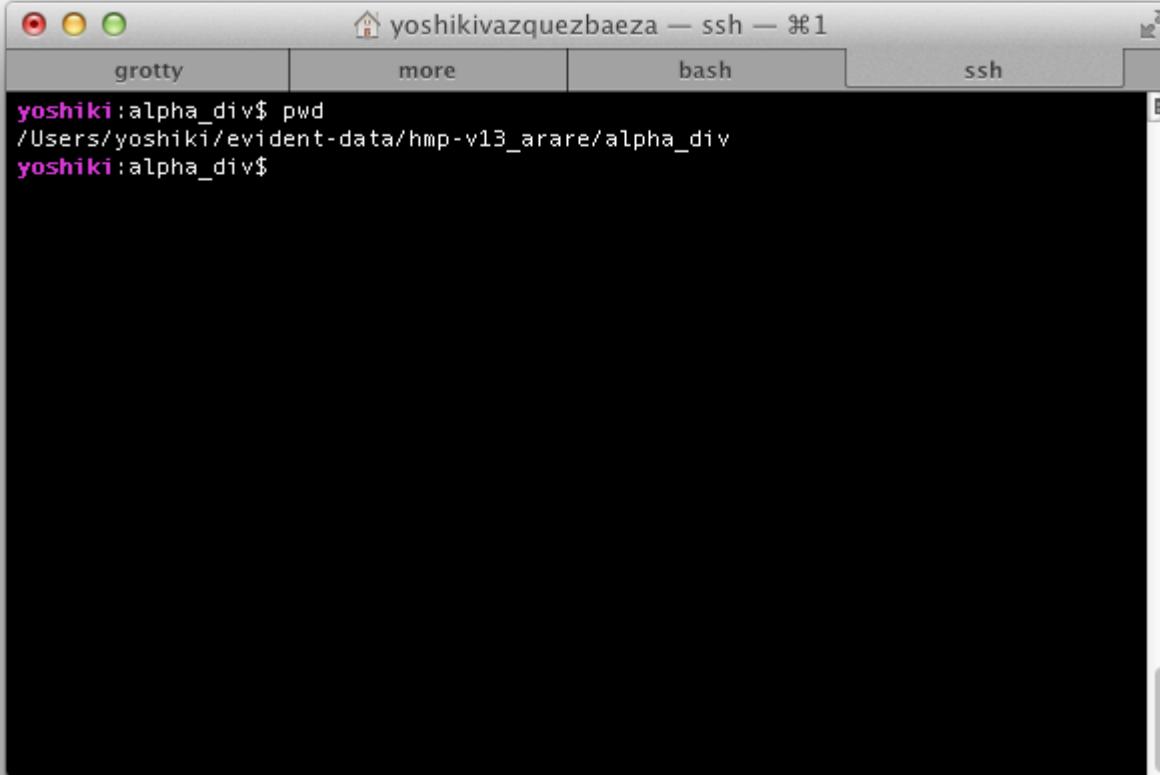
Some useful commands

... for life

You are here

Get your current working directory:

pwd



```
yoshiki:alpha_div$ pwd
/Users/yoshiki/evident-data/hmp-v13_arare/alpha_div
yoshiki:alpha_div$
```

Folders, files and its information

List files from your current working directory:

```
ls
```

List all the files in your current directory, including hidden files:

```
ls -a
```

List files in your home directory, in the long format and sort them by time:

```
ls -lt ~
```

Navigating your machine

Change from your current directory to your home

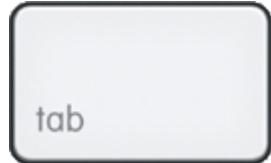
```
cd
```

Change from your current directory to a directory above it:

```
cd ..
```

Change from your current directory to your personal folder:

```
cd /mnt/workshop/cli/<your-name>
```



*to
autocomplete*



Making, copying and moving stuff

Make a new directory

```
mkdir AnExample
```

Move a file/folder or change its name

```
mv oldname.txt newname.txt
```

```
mv Files/ NewName/
```

Copy a file

```
cp homework.txt backup_homework.txt
```

Copy a directory

```
cp -r NewName Files
```

Making, copying and moving stuff

Make a new directory

mkdir AnExample

Move a file/folder or change its name

mv oldname.txt newname.txt

mv Files/ NewName/

**All of these commands have
the form:**

Copy a file **SOURCE -> DESTINATION**

cp homework.txt backup_homework.txt

Copy a directory

cp -r NewName Files

Star



It's a wildcard.

List anything that ends with a .txt

```
ls *.txt
```

List anything with the letter t

```
ls *t*
```

Copy to your desktop all text files

```
cp *.txt ~/Desktop/
```



Removing files



Remember to be careful, be very careful, there is **no undo** for this command.

Remove a file

```
rm some_file.txt
```

Remove a folder with things inside it

```
rm -r UselessFolder/
```

Force, the **removal** of a folder

```
rm -rf UselessFolder/
```

Compression and decompression

Using zip to compress

```
zip compressed.zip bigfile.txt  
zip -r compressedFolder.zip BigFolder
```

Using zip to decompress *things*

```
unzip compressed.zip
```

Using tar to compress a folder or file

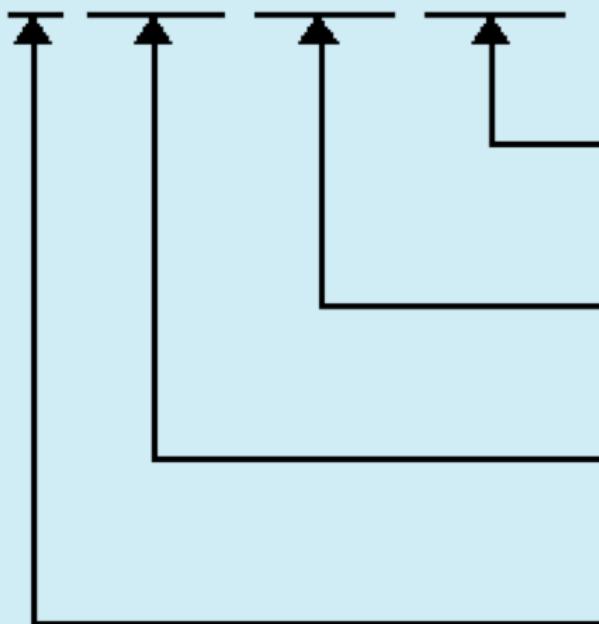
```
tar -czf output.tgz BigFolder
```

Using tar to decompress *things*

```
tar -xzf output.tgz
```

Permissions

- rwxrwx - r - -



Read, write, and execute permissions
for all other users

Read, write and execute permissions
for members of the group owning the
file.

Read, write and execute permissions
for the owner of the file.

File type. "-" indicates a regular file. A
"d" indicates a directory.

Permissions

Allow all to have write permissions to a file

```
chmod a+w file.txt
```

Allow all to have write permissions to a folder
and its contents:

```
chmod -R a+w file.txt
```

Remove all the permission to write to a file

```
chmod a-w file.txt
```

Remove all the permission to write to a folder
and its contents:

```
chmod -R a-w file.txt
```

To see how permissions change, use:
`ls -l`

Inspecting a text file

Print a file in the screen

```
cat file.txt
```

Inspect the contents

```
less file.txt
```

```
more file.txt
```

to exit these commands type q

Count the words of a file

```
wc file.txt
```

Count the lines of a file

```
wc -l file.txt
```

Inspecting parts of a file

Seeing the first few lines of a file

```
head file.txt
```

Seeing the N lines of a file

```
head -n 20 file.txt
```

Seeing the last few lines of a file

```
tail file.txt
```

Seeing the N lines of a file

```
tail -n 20 file.txt
```

Searching the contents of a file

Searching for text in a file:

```
grep "yet" file.txt
```

Searching for text in a file (and show 2 lines before "-B" or 2 lines after "-A" value):

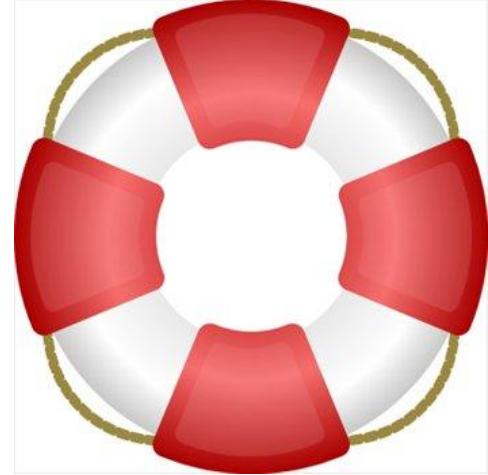
```
grep -A 2 "yet" file.txt  
grep -B 2 "yet" file.txt
```

Searching for text and highlight the matches:

```
grep --color "yet" file.txt
```

Getting help

Each command has its own way i. e.



As an argument:

`zip -h`

`tar --help`

Using the *manual* reference command:

`man ls`

`man grep`

to exit the manual just type `q`

Binaries, scripts, programs etc ...

- Try the following:

`which ls`

- PATH has a lot of information, *it's the "route"*

`echo $PATH`

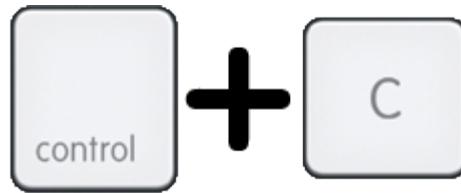
- To execute something from your current working directory

ensure it has the right permissions, then:

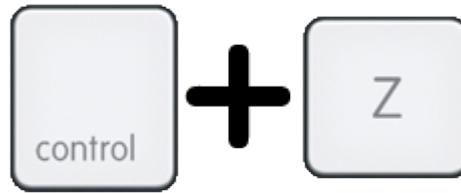
`./program_test`

if it doesn't have permissions try `chmod a+x program_test`

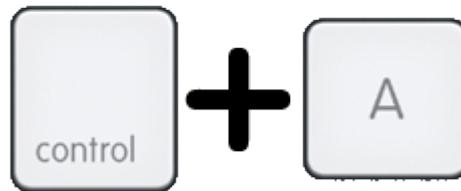
Special shortcuts



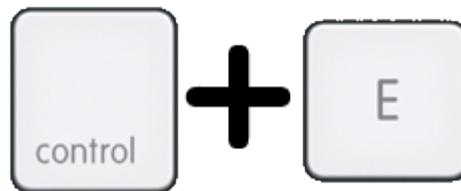
halt and kill a command



stop a command (doesn't kill it)



go to the beginning of the line in a terminal window



go to the end of the line in a terminal window

This applies for any operating system.

Intro to IPython Notebook

IPython reference

IPython acts like a hybrid python/bash environment.

The way we interact with the IPython notebook is through the ‘cells’

```
In [ ]: from random import choice
from os import chdir, mkdir, makedirs
from os.path import join
from IPython.display import FileLinks as ipFileLinks, FileLink as ipFileLink

# to support running in a multi-user environment, each user will work in
# a temporary working directory with a randomly generated name
basedir = "temp"
choices = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
choices += choices.lower()
working_dir = join(basedir, ''.join([choice(choices) for i in range(10)]))

otu_base = "/home/ubuntu/qime_software/gg_otus-12_10-release/"
reference_segs = join(otu_base, "rep_set/97_otus.fasta")
reference_tree = join(otu_base, "trees/97_otus.tree")
reference_tax = join(otu_base, "taxonomy/97_otu_taxonomy.txt")

print "Your working directory is %s" % working_dir
makedirs(working_dir)
chdir(working_dir)

!wget https://s3.amazonaws.com/s3-qime_tutorial_files/moving_pictures_tutorial-1.7.0.tgz
!tar -xzf moving_pictures_tutorial-1.7.0.tgz

# To use FileLink(s), but link to files in the user's working directory
# we wrap the call to FileLink(s) to append the working_dir to the
# url_prefix. NOTE: This is not something that you'll generally need to
# do - it's only important as we're working with multiple users in the
# IPython Notebook, which is currently only a single-user environment.
def FileLinks(path):
    return ipFileLinks(path,url_prefix='files/%s/' % working_dir)

def FileLink(path):
    return ipFileLink(path,url_prefix='files/%s/' % working_dir)
```

To view output files, you will use the commands FileLink and FileLinks. Calling FileLink('some_file.txt') produces a standard html-like link to that file which you can click on. Clicking on the link will bring up a new browser tab with the contents of 'some_file.txt' displayed. Just to practice, try executing the commands in the following cell. You should see an output of a blue html-link. Click this link.

```
In [ ]: !echo 'A test txt file.' > ./practice_filelink.txt
FileLink('practice_filelink.txt')
```

IPython reference

IPython acts like a hybrid python/bash environment.

The way we interact with the IPython notebook is through the ‘cells’

```
In [ ]: from random import choice
from os import chdir, mkdir, makedirs
from os.path import join
from IPython.display import FileLinks as ipFileLinks, FileLink as ipFileLink

# to support running in a multi-user environment, each user will work in
# a temporary working directory with a randomly generated name
basedir = "temp"
choices = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
choices += choices.lower()
working_dir = join(basedir, ''.join([choice(choices) for i in range(10)]))

otu_base = "/home/ubuntu/qime_software/gg_otus-12_10-release/"
reference_segs = join(otu_base, "rep_set/97_otus.fasta")
reference_tree = join(otu_base, "trees/97_otus.tree")
reference_tax = join(otu_base, "taxonomy/97_otu_taxonomy.txt")

print "Your working directory is %s" % working_dir
makedirs(working_dir)
chdir(working_dir)

!wget https://s3.amazonaws.com/s3-qime_tutorial_files/moving_pictures_tutorial-1.7.0.tgz
!tar -xzf moving_pictures_tutorial-1.7.0.tgz

# To use FileLink(s), but link to files in the user's working directory
# we wrap the call to FileLink(s) to append the working_dir to the
# url_prefix. NOTE: This is not something that you'll generally need to
# do - it's only important as we're working with multiple users in the
# IPython Notebook, which is currently only a single-user environment.
def FileLinks(path):
    return ipFileLinks(path,url_prefix='files/%s/' % working_dir)

def FileLink(path):
    return ipFileLink(path,url_prefix='files/%s/' % working_dir)
```

To view output files, you will use the commands FileLink and FileLinks. Calling FileLink('some_file.txt') produces a standard html-like link to that file which you can click on. Clicking on the link will bring up a new browser tab with the contents of 'some_file.txt' displayed. Just to practice, try executing the commands in the following cell. You should see an output of a blue html-link. Click this link.

```
In [ ]: !echo 'A test txt file.' > ./practice_filelink.txt
FileLink('practice_filelink.txt')
```

IPython reference

IPython acts like a hybrid python/bash environment.

The way we interact with the IPython notebook is through the ‘cells’

```
In [ ]: from random import choice
from os import chdir, mkdir, makedirs
from os.path import join
from IPython.display import FileLinks as ipFileLinks, FileLink as ipFileLink

# to support running in a multi-user environment, each user will work in
# a temporary working directory with a randomly generated name
basedir = "temp"
choices = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
choices += choices.lower()
working_dir = join(basedir, ''.join([choice(choices) for i in range(10)]))

otu_base = "/home/ubuntu/qime_software/gg_otus-12_10-release/"
reference_segs = join(otu_base, "rep_set/97_otus.fasta")
reference_tree = join(otu_base, "trees/97_otus.tree")
reference_tax = join(otu_base, "taxonomy/97_otu_taxonomy.txt")

print "Your working directory is %s" % working_dir
makedirs(working_dir)
chdir(working_dir)

!wget https://s3.amazonaws.com/s3-qime_tutorial_files/moving_pictures_tutorial-1.7.0.tgz
!tar -xzf moving_pictures_tutorial-1.7.0.tgz

# To use FileLink(s), but link to files in the user's working directory
# we wrap the call to FileLink(s) to append the working_dir to the
# url_prefix. NOTE: This is not something that you'll generally need to
# do - it's only important as we're working with multiple users in the
# IPython Notebook, which is currently only a single-user environment.
def FileLinks(path):
    return ipFileLinks(path,url_prefix='files/%s/' % working_dir)

def FileLink(path):
    return ipFileLink(path,url_prefix='files/%s/' % working_dir)
```

To view output files, you will use the commands FileLink and FileLinks. Calling FileLink('some_file.txt') produces a standard html-like link to that file which you can click on. Clicking on the link will bring up a new browser tab with the contents of 'some_file.txt' displayed. Just to practice, try

```
In [ ]: !echo 'A test txt file.' > ./practice_filelink.txt
FileLink('practice_filelink.txt')
```

IPython reference

Commands prefixed by a '!' character are issued to the shell (just like what your terminal runs).

```
!wget https://s3.amazonaws.com/s3-qime_tutorial_files/moving_pictures_tutorial-1.8.0.tgz  
!tar -xzf moving_pictures_tutorial-1.8.0.tgz  
  
In [2]: !echo 'A test txt file.' > ./practice_filelink.txt  
FileLink('practice_filelink.txt')
```

Commands not prefixed with '!' are issued to python, and behave as they normally would in python.

```
In [1]: from random import choice  
from os import chdir, mkdir, makedirs  
from os.path import join  
from IPython.display import FileLinks as ipFileLinks, FileLink as ipFileLink
```

Each 'cell' of the notebook is executable. Shift+Enter (or the play button) is the way you execute (or re-execute) the commands in a given cell. You must click in the cell to gain focus in that cell, and then type Shift+Enter or hit the play button

IP[y]: Notebook stamps_2013_wvtreuren (autosaved: Aug 02 12:49)

Logout



IPython reference

Each executable has a prefix that shows you its status (if it has been run, if it hasn't been run, or if its still running)

Hasn't been run

```
In [ ]: !biom summarize-table -i moving_pictures_tutorial-1.8.0/illumina/otus/otu_table_mc2_w_tax_no_pynast_failures.biom -o moving_pictures_tutorial-1.8.0/illumina/otus/otu_table_mc2_w_tax_no_pynast_failures.biom.stats
```

Has been run

```
In [13]: !split_libraries_fastq.py -o moving_pictures_tutorial-1.7.0/illumina/slout/ -i moving_pictures_tutorial-1.7.0/illumina/raw/subsampled.fna
```

```
In [14]: FileLinks('moving_pictures_tutorial-1.7.0/illumina/slout/')
```

```
Out[14]: moving_pictures_tutorial-1.7.0/illumina/slout/
          histograms.txt
          split_library_log.txt
          seqs.fna
```

```
In [15]: !count_seqs.py -i moving_pictures_tutorial-1.7.0/illumina/slout/seqs.fna
```

```
66189 : moving_pictures_tutorial-1.7.0/illumina/slout/seqs.fna (Sequence lengths (mean +/- std): 132.1182 +/- 9.6185)
66189 : Total
```

Still running

```
In [*]: !pack_open_reference_otus.py -o moving_pictures_tutorial-1.7.0/illumina/otus/ -i moving_pictures_tutorial-1.7.0/illumina/slout/seqs.fna
```

Illumina Overview Tutorial

[this article](#)[search](#)[submit a manuscript](#)[register](#)

Genome Biol. 2011; 12(5): R50.

PMCID: PMC3271711

Published online 2011 May 30. doi: [10.1186/gb-2011-12-5-r50](https://doi.org/10.1186/gb-2011-12-5-r50)

Moving pictures of the human microbiome

J Gregory Caporaso,¹ Christian L Lauber,² Elizabeth K Costello,³ Donna Berg-Lyons,² Antonio Gonzalez,⁴ Jesse Stombaugh,¹ Dan Knights,⁴ Pawel Gajer,⁵ Jacques Ravel,⁵ Noah Fierer,^{2,6} Jeffrey I Gordon,⁷ and Rob Knight^{1,8}

Moving Pictures of the Human Microbiome

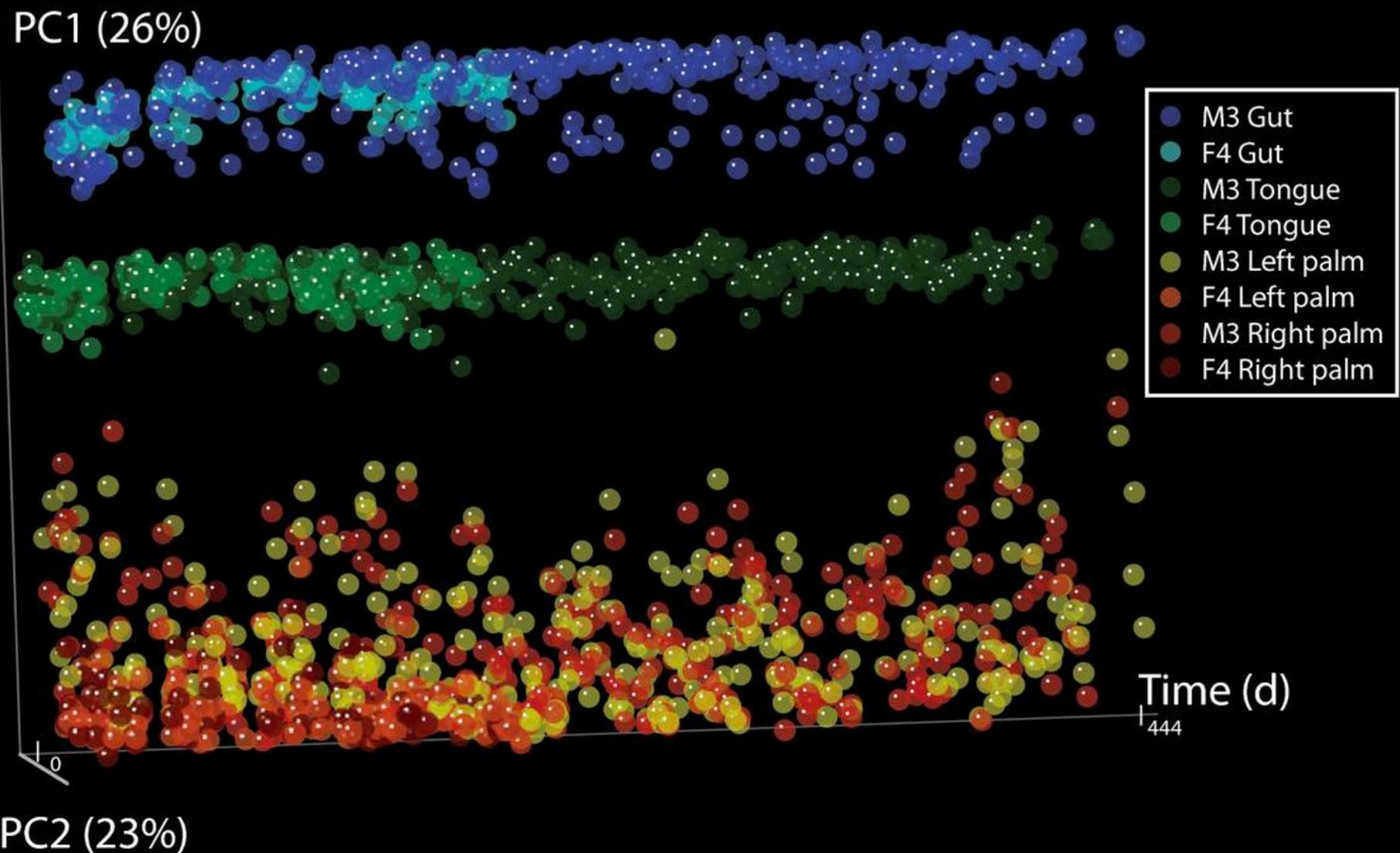
- Two subjects sampled daily, one for six months, one for 18 months
- Four body sites: tongue, palm of left hand, palm of right hand, and gut (via fecal swabs).

Moving Pictures of the Human Microbiome

- Investigate the relative temporal variability of body sites.
- Is there a temporal core microbiome?

Moving Pictures of the Human Microbiome: QIIME tutorial

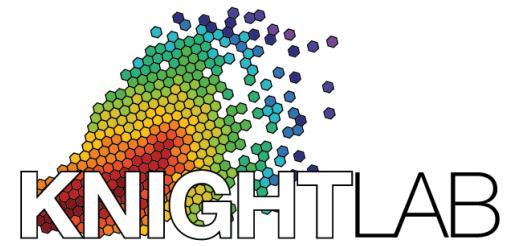
- A **small** subset of the full data set to facilitate short run time: ~0.1% of the full sequence collection.
- Sequenced across six Illumina GAIIx lanes, with a subset of the samples also sequenced with Roche 454 technology.



Acknowledgements

Evan Bolyen
Nick Bokulich
Katy Califf
Greg Caporaso
Jose Clemente
John Chase
Kevin Cohen
Antonio Gonzalez
Crystal Hepp
Rob Knight
Bruce Hungate
Larry Hunter
Paul Keim
Scott Kelley
Justin Kuczynski

Cathy Lozupone
Daniel McDonald
David Mills
Norm Pace
Fernando Perez
Jai Ram Rideout
Egbert Schwartz
Karen Schwarzberg
Jeffrey Siegel
Jesse Stombaugh
Yoshiki Vazquez
Tony Walters



ALFRED P. SLOAN
FOUNDATION



william.a.walters@gmail.com
jai.rideout@gmail.com



NORTHERN
ARIZONA
UNIVERSITY

Slides compiled by:

Greg Caporaso

John Chase

Jose Clemente

Antonio Gonzalez Peña

Rob Knight

Cathy Lozupone

Daniel McDonald

Jai Ram Rideout

Yoshiki Vázquez Baeza

Tony Walters



This work is licensed under the Creative Commons Attribution 3.0 United States License. To view a copy of this license, visit

<http://creativecommons.org/licenses/by/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Feel free to use or modify these slides, but please credit us by placing the following attribution information where you feel that it makes sense:

Slides derived from QIIME educational materials www.qiime.org.