

Software Carpentry

`git`, testing, pipelines

Mark T. Holder

Department of Ecology and Evolutionary Biology, University of Kansas

Aug 23, 2013

Motivating example

human	C	G	A	C	C	A	G	G	T	A	T	G
chimp	C	G	A	C	C	A	G	G	T	A	C	G
gorilla	C	C	G	T	C	C	G	G	T	C	T	G
orang	C	C	G	C	C	T	G	G	T	C	C	G

Motivating example

human	C	G	A	C	C	A	G	G	T	A	T	G
chimp	C	G	A	C	C	A	G	G	T	A	C	G
gorilla	C	C	G	T	C	C	G	G	T	C	T	G
orang	C	C	G	C	C	T	G	G	T	C	C	G

human-with-chimp: 3

human-with-gorilla: 1

human-with-orang: 0

Motivating example

human	C	G	A	C	C	A	G	G	T	A	T	G
chimp	C	G	A	C	C	A	G	G	T	A	C	G
gorilla	C	C	G	T	C	C	G	G	T	C	T	G
orang	C	C	G	C	C	T	G	G	T	C	C	G

human-with-chimp: 3

human-with-gorilla: 1

human-with-orang: 0

“best” = largest count = 3

Motivating example

human	C	G	A	C	C	A	G	G	T	A	T	G
chimp	C	G	A	C	C	A	G	G	T	A	C	G
gorilla	C	C	G	T	C	C	G	G	T	C	T	G
orang	C	C	G	C	C	T	G	G	T	C	C	G

human-with-chimp: 3

human-with-gorilla: 1

human-with-orang: 0

“best” = largest count = 3

“support” = best - second best = 3 - 1 = 2

Motivating example

human	C	G	A	C	C	A	G	G	T	A	T	G
chimp	C	G	A	C	C	A	G	G	T	A	C	G
gorilla	C	C	G	T	C	C	G	G	T	C	T	G
orang	C	C	G	C	C	T	G	G	T	C	C	G

human-with-chimp: 3

human-with-gorilla: 1

human-with-orang: 0

“best” = largest count = 3

“support” = best - second best = 3 - 1 = 2

Motivating example

Input: A file with many character matrices, each with 4 species.

Operations:

- ➊ Calculate the support for each matrix.
- ➋ Report the smallest integer that is a larger than at least 95% of the support values that you calculated in step 1.

IPython is great, but...

Storing code in files is better for:

- Sharing code,
- Running on cluster,
- Reuse of code...

The programmer's workflow:

- ① Write code in a text editor
- ② Save the file – save to the filesystem
- ③ Execute the code from your terminal
- ④ Rewrite in text editor
- ⑤ Go back to step 2

Some useful tricks

Command-tab to switch between editor and terminal.

Arrow-up-key to recall previous commands.

Opening a GUI file browser in your terminal's working directory:

Mac: `$ open .`

GitBash on Windows: `$ explorer .`

Ubuntu: `$ nautilus .`

Version control

Except for trivial tasks, programming involves

- writing code,
- testing whether it works
- repeat

You will have many *versions* of your code.
A version control system helps you manage changes to a code-base

git – a distributed version control system

“Distributed” here means every developer has the entire version history. There is no central server.

git lets you

- return to a previous snapshot of your code,
- merge work from several developers,
- easily share your code,
- easily back up your code.

github \neq git

`https://github.com/` is a website that offers a lot of very nice services centered around code stored in `git`. But `git` is a command line tool that was developed years before github.

If you are going to use `git`, you probably want to create a github account to see what the fuss is about.

I know a lot of programmers.

I do not know *any* programmers who

- know how to use a version control system, and
- prefer to write code without using version control system

I do not know *any* programmers who:

- know how to use a *distributed* version control system, and
- prefer centralized version control to *distributed* version control

`git` is clearly one of the best distributed version control systems.

If you are going to do more than dabble with programming, you should learn `git`.

Learning `git` will hurt a bit. It is not an easy tool.

Goals for this `git` Intro

- become comfortable with the “core” `git` commands
- gain a sense of the other important features.

Core git commands

- `git init` – create an empty repository
- `git clone` – obtain an existing repository
- `git add` and `git commit` to add content to the repository
- `git checkout` – copy code out of the repository to your filesystem
- `git status` – create a report of how `git` sees the world

More `git` that I'll show you, but we won't practice (much)

- `git pull` – grab the latest changes from a repository on another computer. Get other people's changes.
- `git push` – publish your latest changes to a repository on another computer. Give other people your code.
- `git branch`
- `git merge`
- `git fetch`