GENOME 569

Bioinformatics workflows for high-throughput sequencing experiments

What is a bioinformatics workflow?

A bioinformatics workflow is the collection of **scripts**, **programs**, and **procedures** you use to transform your data into a paper.

Goals for the course

Learn how to write reproducible workflows to analyze your data.

Learn how to use some awesome software tools.

Become a better programmer.

A bit about me

- **High-school**: worked as student software engineer for US Army
- **College**: software engineer for stock, futures, & foreign currency broker
- **After college**: software engineer for a “Business intelligence” company
- **Grad school**: Computer science (supercomputing, then bioinformatics)
- **Postdoc**: RNA biology, stem cell biology, genomics tech dev
- **Now**: single-cell genomics tech dev, developmental bio

A bit about you?

How we’re gonna do this

I will spend about half of each class telling you about one or more new tools

You will spend the rest of the class trying it out

You will read a lot more about it after class

You will make use of it for the projects

What you will learn in this class

- UNIX scripts to automate tedious and computationally intensive jobs
- How to explore a dataset with R
- How to write an R package so others can do so on similar datasets
- How to make your data analysis and figure generation reproducible
- How to share all this code with the world

The Classes

- Before each class, I will give you a lot of stuff to read, but it will be fun
- I will present some slides on a couple of really awesome programming tools each class.
- Then you will do some in class exercises with them - these will build up a bioinformatics workflow over time.

Preliminary hassles

You will need a laptop in class.

You will need to be able to log into the GS cluster.

```
$ ssh <your-userid>@nexus.gs.washington.edu
```

Let’s get started

qlogin

Problem: launch a shell on the cluster

Solution: `qlogin -P genome569 -l mfree=2G`

Git

What is version control?

A *version control system* keeps track of changes to computer code over time

Modern VCS manage code contributions from many users, merge conflicting edits, allow changes to be compared, audited, and reverted

Almost all software companies and most open source projects use VCS.

Why do we use version control?

Share code easily with collaborators

Revert code if you make a mistake

Improves code clarity and quality

Keep a record of what you’ve done

Git

Git is a VCS (written by Linus Torvalds)

GitHub is a website that hosts Git repositories

We will use both in this class

Let’s use Git on the cluster

First, you need to generate an ssh key:

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

> Generating public/private rsa key pair.

> Enter passphrase (empty for no passphrase): [Just hit enter]
> Enter same passphrase again: [Just hit enter again]

$ cat ~/.ssh/id_rsa.pub
```

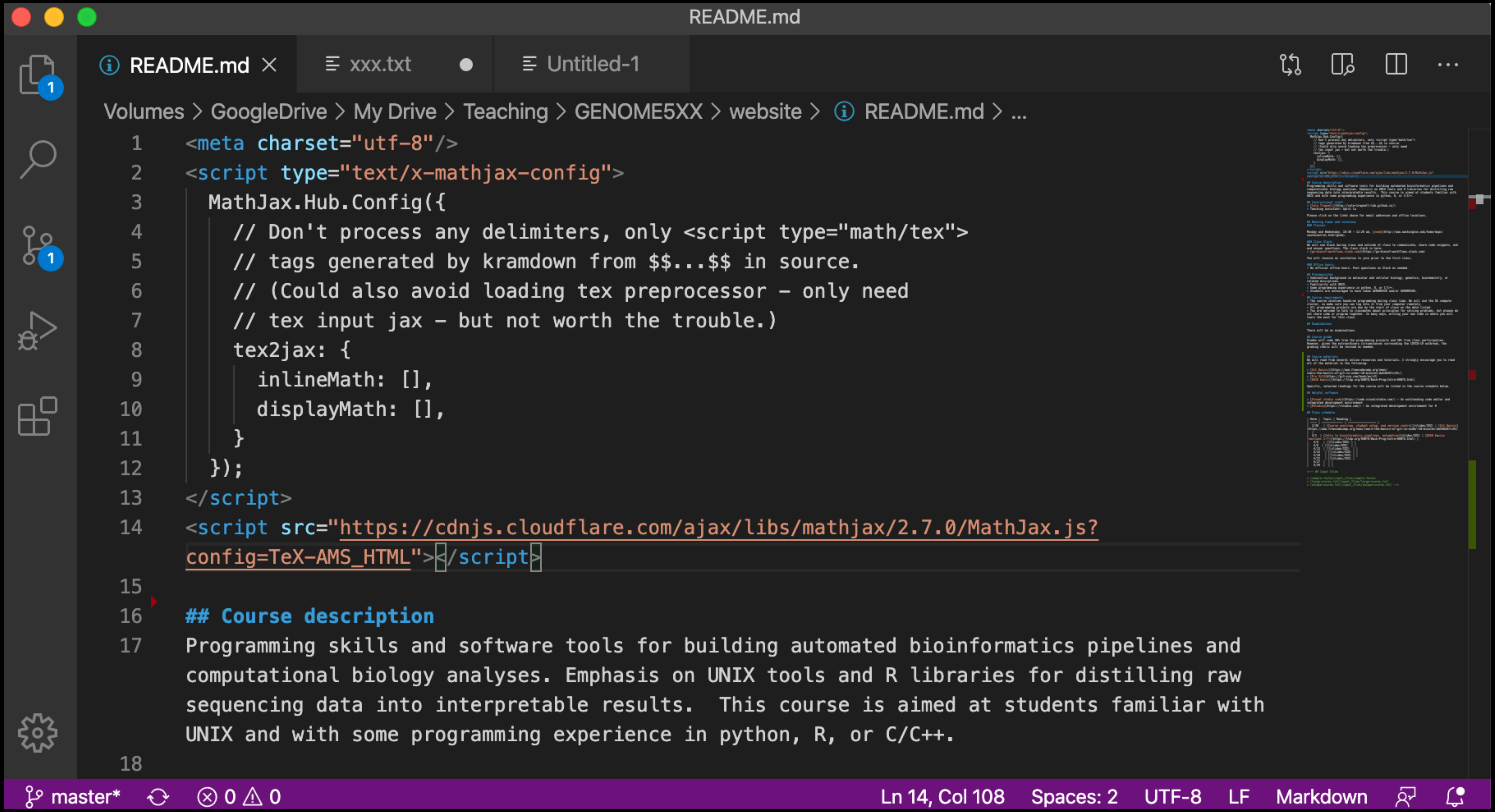
Copy that whole block of text (not the command prompt) into the clipboard

Add your ssh key to GitHub

(follow the instructions I am about to slack you)

https://help.github.com/en/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account

Visual Studio Code is a free code editor



```
<meta charset="utf-8"/>
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    // Don't process any delimiters, only <script type="math/tex">
    // tags generated by kramdown from $$...$$ in source.
    // (Could also avoid loading tex preprocessor - only need
    // tex input jax - but not worth the trouble.)
    tex2jax: {
      inlineMath: [],
      displayMath: [],
    }
  });
</script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?config=TeX-AMS_HTML"></script>

## Course description
Programming skills and software tools for building automated bioinformatics pipelines and computational biology analyses. Emphasis on UNIX tools and R libraries for distilling raw sequencing data into interpretable results.  This course is aimed at students familiar with UNIX and with some programming experience in python, R, or C/C++.

```

VS Code tunneling

Set up VS Code Tunneling

First set up and run a script on the GS cluster to enable VS code tunneling

```
$ mkdir junk
$ cd junk
$ git clone https://github.com/cole-trapnell-lab/trapnell-cluster
$ cd trapnell-cluster
$ ./install.sh
$ serve_vscode
```

<- You may need to do run this again when you log back in later

<pop over to live VS Code demo>

THE END

For now