

HOW TO KEEP A DRY LAB NOTEBOOK

PhD retreat 2019

OVERVIEW

- Introduction to Git
- Examples

INTRO

Record keeping in the dry lab has many different types

- Record of daily tasks/commands

June 1st:

I downloaded 50000 CRAMs to scratch 110

June 2nd:

Ran *this tool* using *these parameters*:

- History of code development



The screenshot shows a GitHub-style diff interface for a file named 'SVtools/write_run_tidy_scripts/set_up_dir.sh'. The top bar indicates 3 changes (1 insertion, 2 deletions) across 46 lines. The main area displays the script's content with color-coded changes:

```
@@ -39,8 +39,7 @@ export fileListDIR=batch${batchNumber}_samples1.list # 1 because it
39    39
40    40      ## Make subfolders
41    41      cd ${inDIR}
42    -      mkdir WDL_scripts
43    -      mkdir write_run_tidy_scripts
42  +      mkdir -pv WDL_scripts write_run_tidy_scripts ${info_outDIR} ${vcf_PPM_outDIR}
44    43
45    44      ## Go to the local comp, copy scripts over
46    45      scp -r WDL_scripts/ farm3-login:${inDIR}
```

GIT

What is Git?

A method to store code, and track changes

You upload your code, then with subsequent uploads, you track changes

Example code:

```
# Get average depth  
samtools depth -r 1:1000-2000 sample.bam
```

Make a change in quality threshold:

```
# Get average depth  
samtools depth -r 1:1000-2000 -Q 30 sample.bam
```



GIT

What is Git?

A method to store code, and track changes

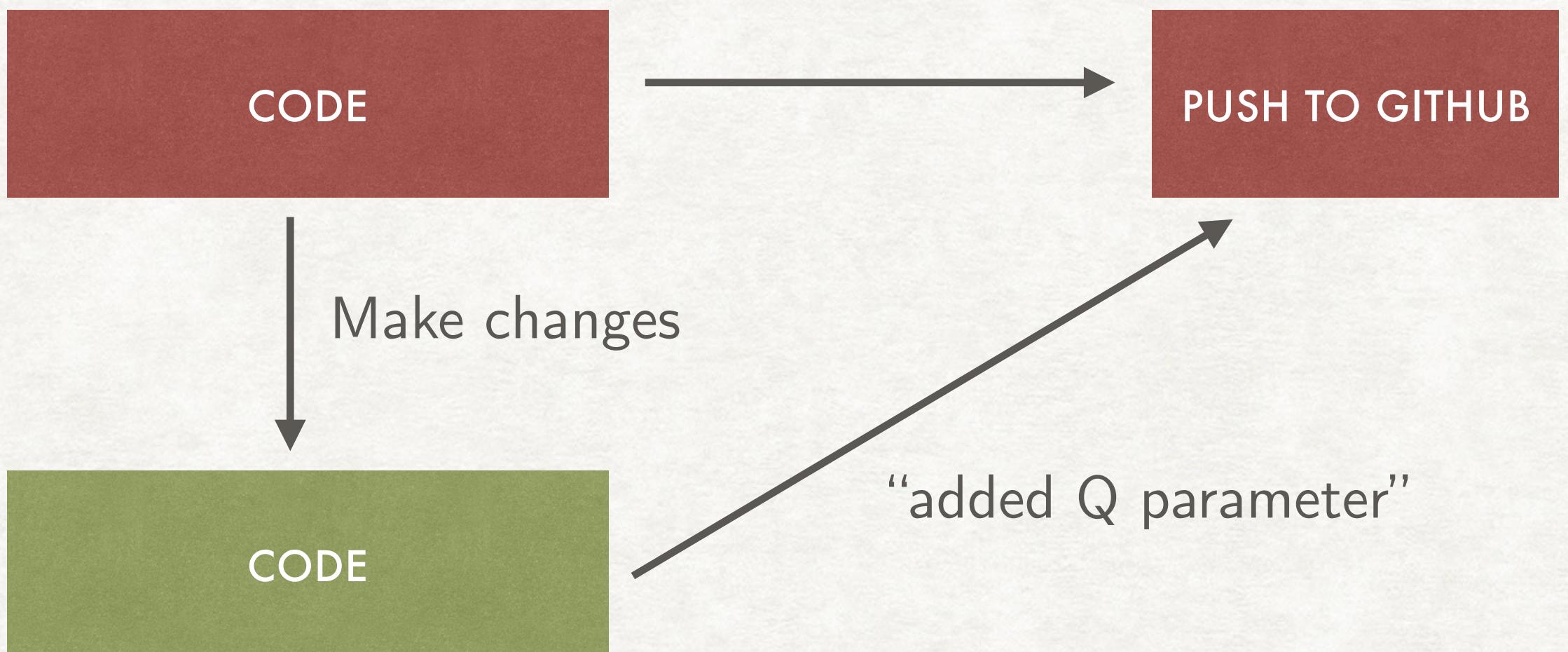
You upload your code, then with subsequent uploads, you track changes

Example code:

```
# Get average depth  
samtools depth -r 1:1000-2000 sample.bam
```

Make a change in quality threshold:

```
# Get average depth  
samtools depth -r 1:1000-2000 -Q sample.bam
```



GIT

What is Git?

A method to store code, and track changes

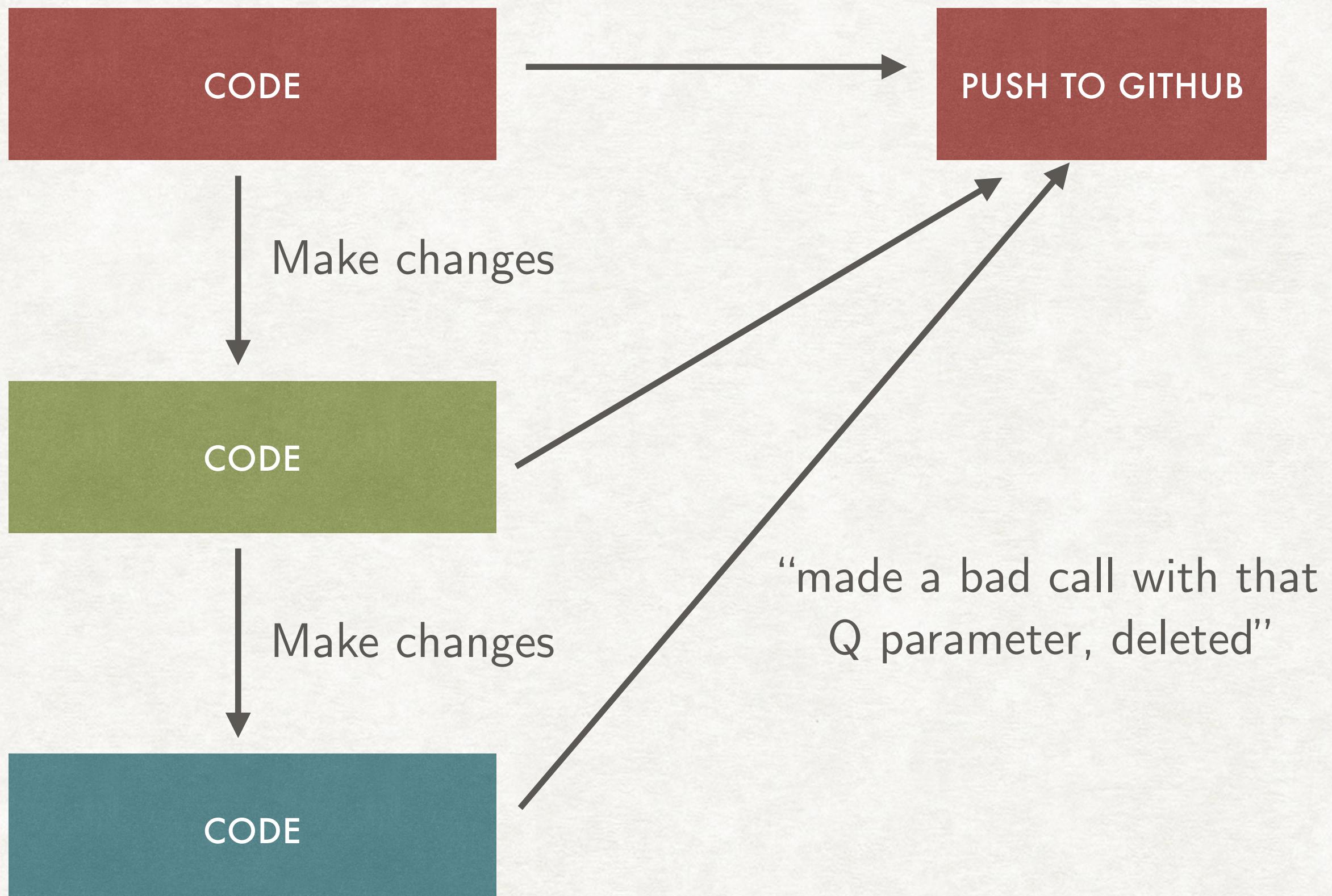
You upload your code, then with subsequent uploads, you track changes

Example code:

```
# Get average depth  
samtools depth -r 1:1000-2000 sample.bam
```

Make a change in quality threshold:

```
# Get average depth  
samtools depth -r 1:1000-2000 -Q sample.bam
```

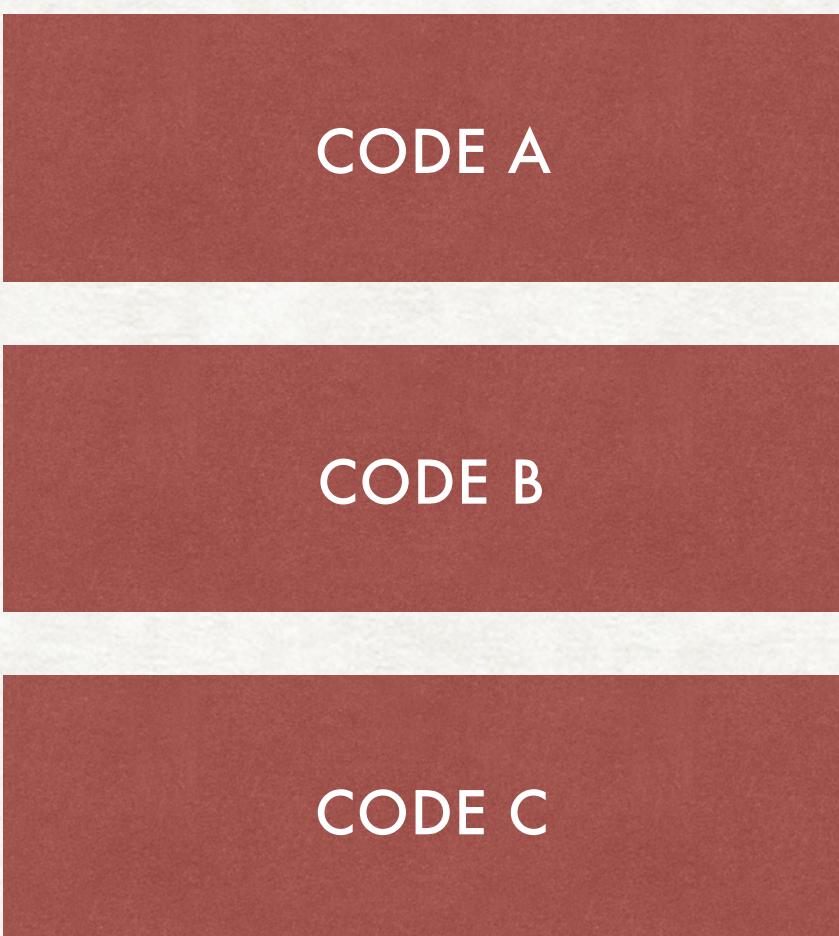


GIT TERMINOLOGY

Git add: Add files to the “staging area”

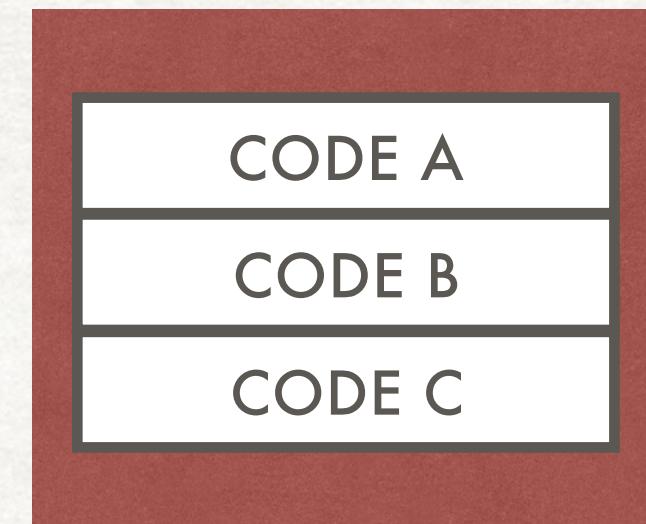
Git commit: Write a description

Git push: Send to the server



Git add
<fileA, fileB, fileC>

The “staging area”



Git commit
“added PCA method”

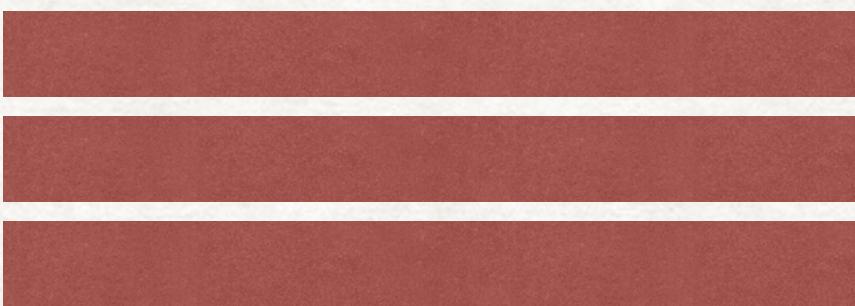


GIT TERMINOLOGY

Git add: Add files to the “staging area”

Git commit: Write a description

Git push: Send to the server



Git add

The “staging area”



Git commit
“added PCA method”

GIT TERMINOLOGY

Git add: Add files to the “staging area”

Git commit: Write a description

Git push: Send to the server



Git add

The “staging area”



Git commit

“added PCA method”



Git add



Git commit

“removed PCA”

GIT TERMINOLOGY

Git add: Add files to the “staging area”

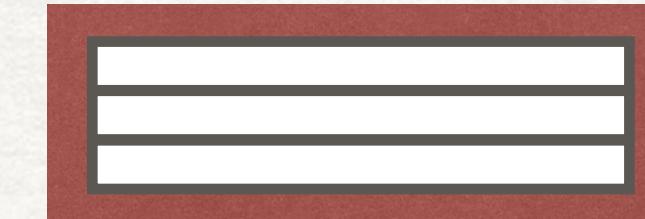
Git commit: Write a description

Git push: Send to the server



Git add

The “staging area”



Git commit

“added PCA method”



Git add

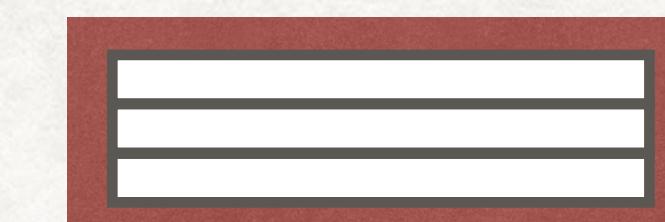


Git commit

“removed PCA”



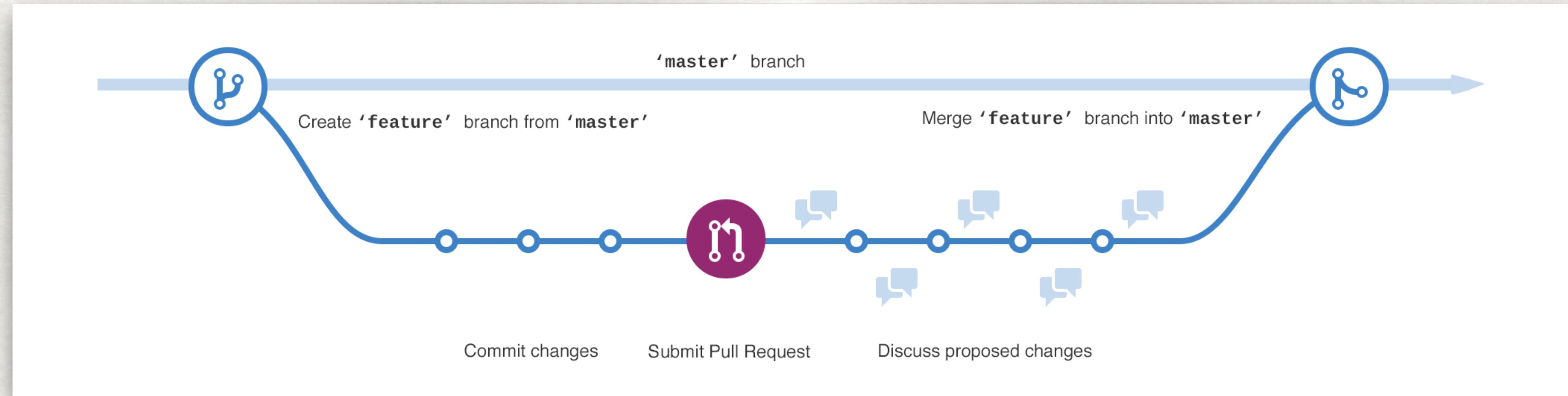
Git add



Git commit

“had another go at that
PCA method”

GIT HAS MUCH MORE POWER



TEXT EDITORS

Text editors are programs used to write in...

They can be command line (emacs, vim, nano)
Or not (pycharm, Rstudio, Atom, sublime text)

Ultimately, it doesn't matter, as long as you
like it.

TEXT EDITORS

Text editors are programs used to write in...

They can be command line (emacs, vim, nano)
Or not (pycharm, Rstudio, Atom, sublime text)

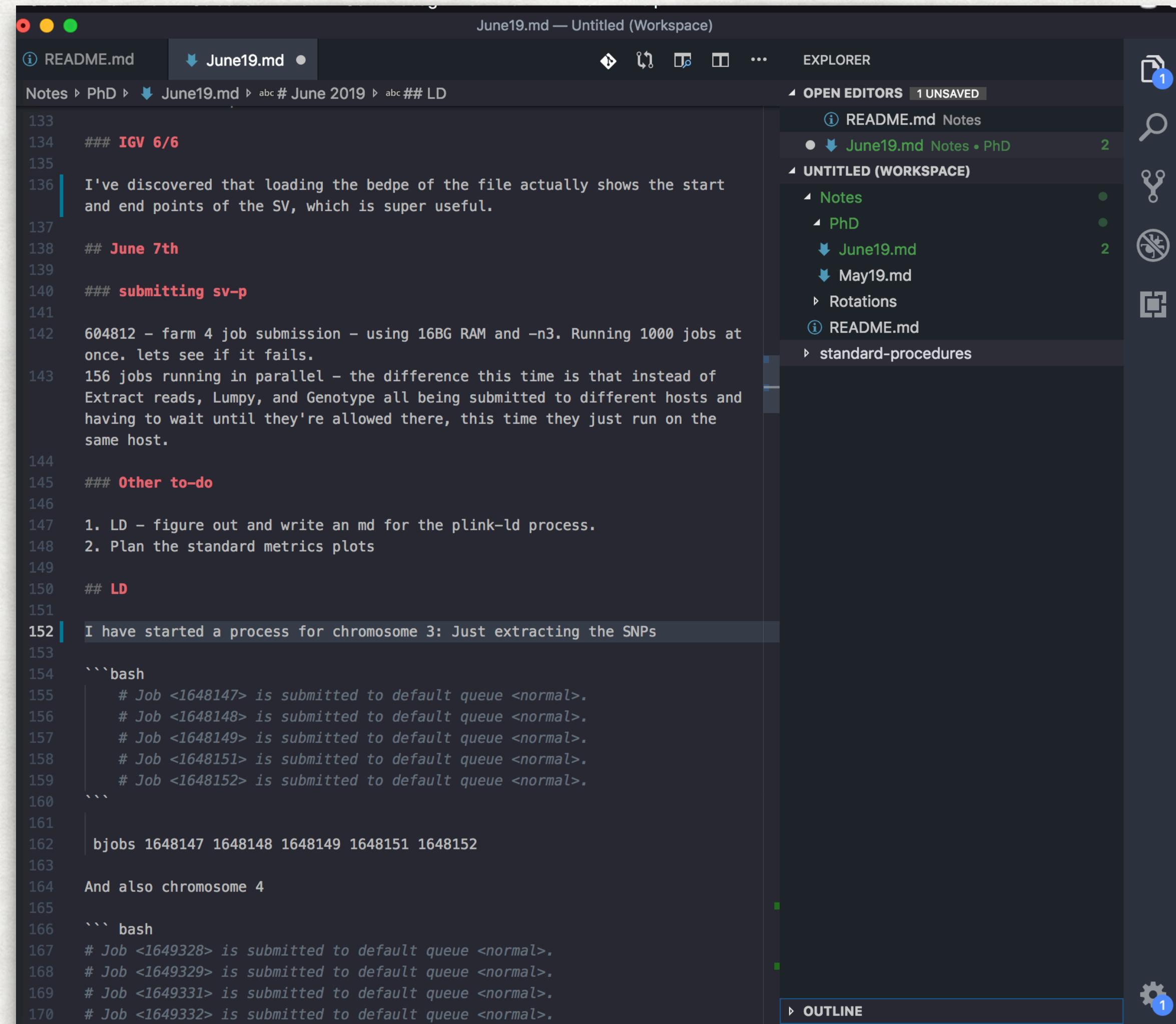
Ultimately, it doesn't matter, as long as you like it.

VSCODE

A text editor which has integrated Git capabilities

Very versatile

Nice file structure



The screenshot shows the Visual Studio Code (VS Code) interface. The title bar indicates "June19.md — Untitled (Workspace)". The main editor area displays a Markdown file with code snippets and comments. The sidebar on the right shows the "OPEN EDITORS" list with "README.md Notes" and "June19.md Notes • PhD". The "UNTITLED (WORKSPACE)" section lists "Notes", "PhD", "June19.md", "May19.md", "Rotations", "README.md", and "standard-procedures". The bottom right corner shows the "OUTLINE" view.

```
133
134  ### IGV 6/6
135
136  I've discovered that loading the bedpe of the file actually shows the start and end points of the SV, which is super useful.
137
138  ## June 7th
139
140  ### submitting sv-p
141
142  604812 - farm 4 job submission - using 16BG RAM and -n3. Running 1000 jobs at once. lets see if it fails.
143  156 jobs running in parallel - the difference this time is that instead of Extract reads, Lumpy, and Genotype all being submitted to different hosts and having to wait until they're allowed there, this time they just run on the same host.
144
145  ### Other to-do
146
147  1. LD - figure out and write an md for the plink-ld process.
148  2. Plan the standard metrics plots
149
150  ## LD
151
152  I have started a process for chromosome 3: Just extracting the SNPs
153
154  ````bash
155  # Job <1648147> is submitted to default queue <normal>.
156  # Job <1648148> is submitted to default queue <normal>.
157  # Job <1648149> is submitted to default queue <normal>.
158  # Job <1648151> is submitted to default queue <normal>.
159  # Job <1648152> is submitted to default queue <normal>.
160  ...
161
162  bjobs 1648147 1648148 1648149 1648151 1648152
163
164  And also chromosome 4
165
166  ````bash
167  # Job <1649328> is submitted to default queue <normal>.
168  # Job <1649329> is submitted to default queue <normal>.
169  # Job <1649331> is submitted to default queue <normal>.
170  # Job <1649332> is submitted to default queue <normal>.
```

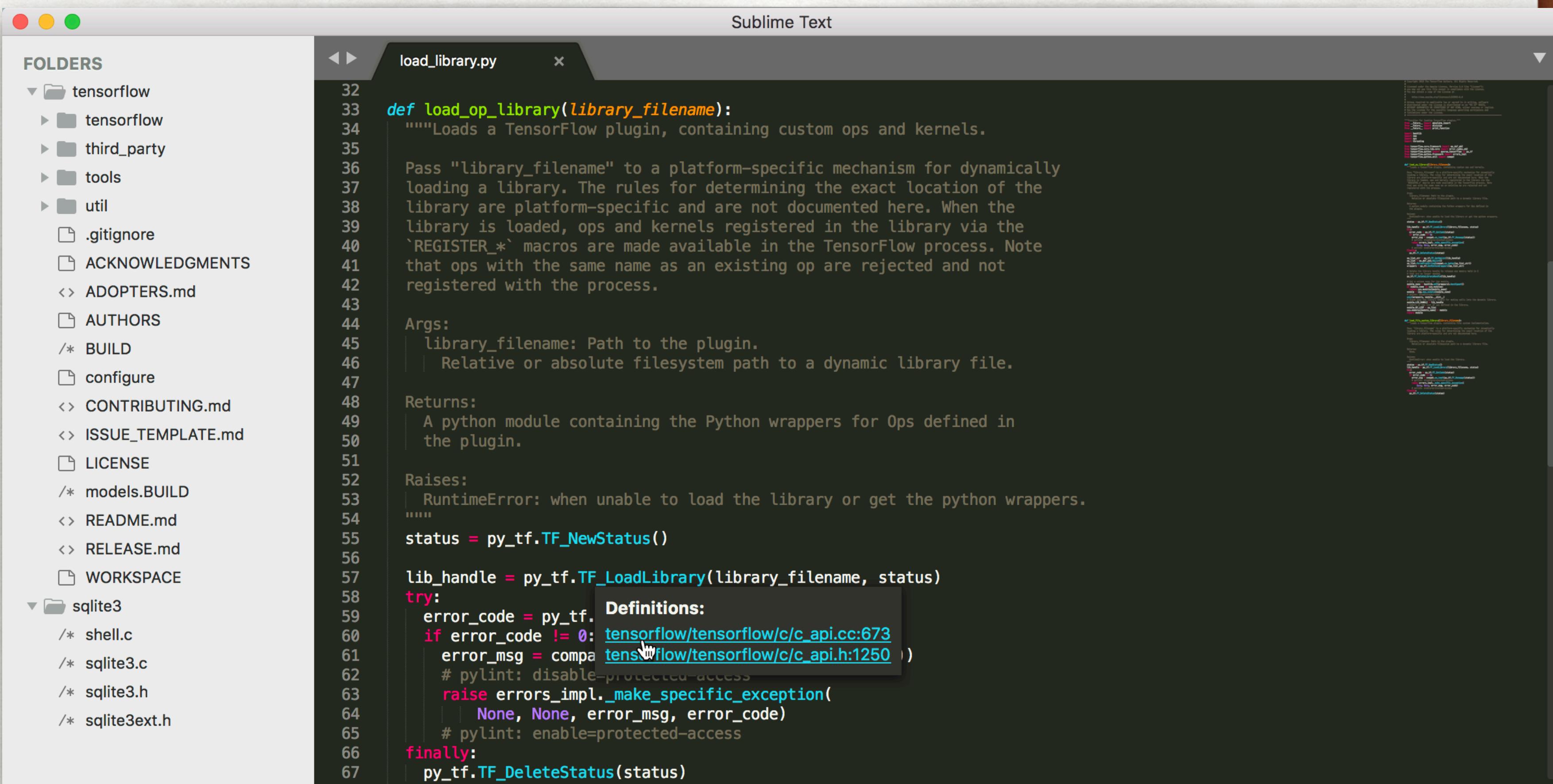
TEXT EDITORS

Sublime text

Text editor

Does have add ons, but not to the same level as vscode

Kinda just notepad with better colours...



The screenshot shows a Sublime Text window with a dark theme. On the left is a sidebar titled 'FOLDERS' showing a project structure with folders like 'tensorflow', 'third_party', 'tools', 'util', '.gitignore', 'ACKNOWLEDGMENTS', 'ADOPTERS.md', 'AUTHORS', 'BUILD', 'configure', 'CONTRIBUTING.md', 'ISSUE_TEMPLATE.md', 'LICENSE', 'models.BUILD', 'README.md', 'RELEASE.md', and 'WORKSPACE'. The main pane contains a Python file named 'load_library.py' with the following content:

```
def load_op_library(library_filename):
    """Loads a TensorFlow plugin, containing custom ops and kernels.

    Pass "library_filename" to a platform-specific mechanism for dynamically
    loading a library. The rules for determining the exact location of the
    library are platform-specific and are not documented here. When the
    library is loaded, ops and kernels registered in the library via the
    `REGISTER_*` macros are made available in the TensorFlow process. Note
    that ops with the same name as an existing op are rejected and not
    registered with the process.

    Args:
        library_filename: Path to the plugin.
            | Relative or absolute filesystem path to a dynamic library file.

    Returns:
        A python module containing the Python wrappers for Ops defined in
        the plugin.

    Raises:
        RuntimeError: when unable to load the library or get the python wrappers.
    """
    status = py_tf.TF_NewStatus()

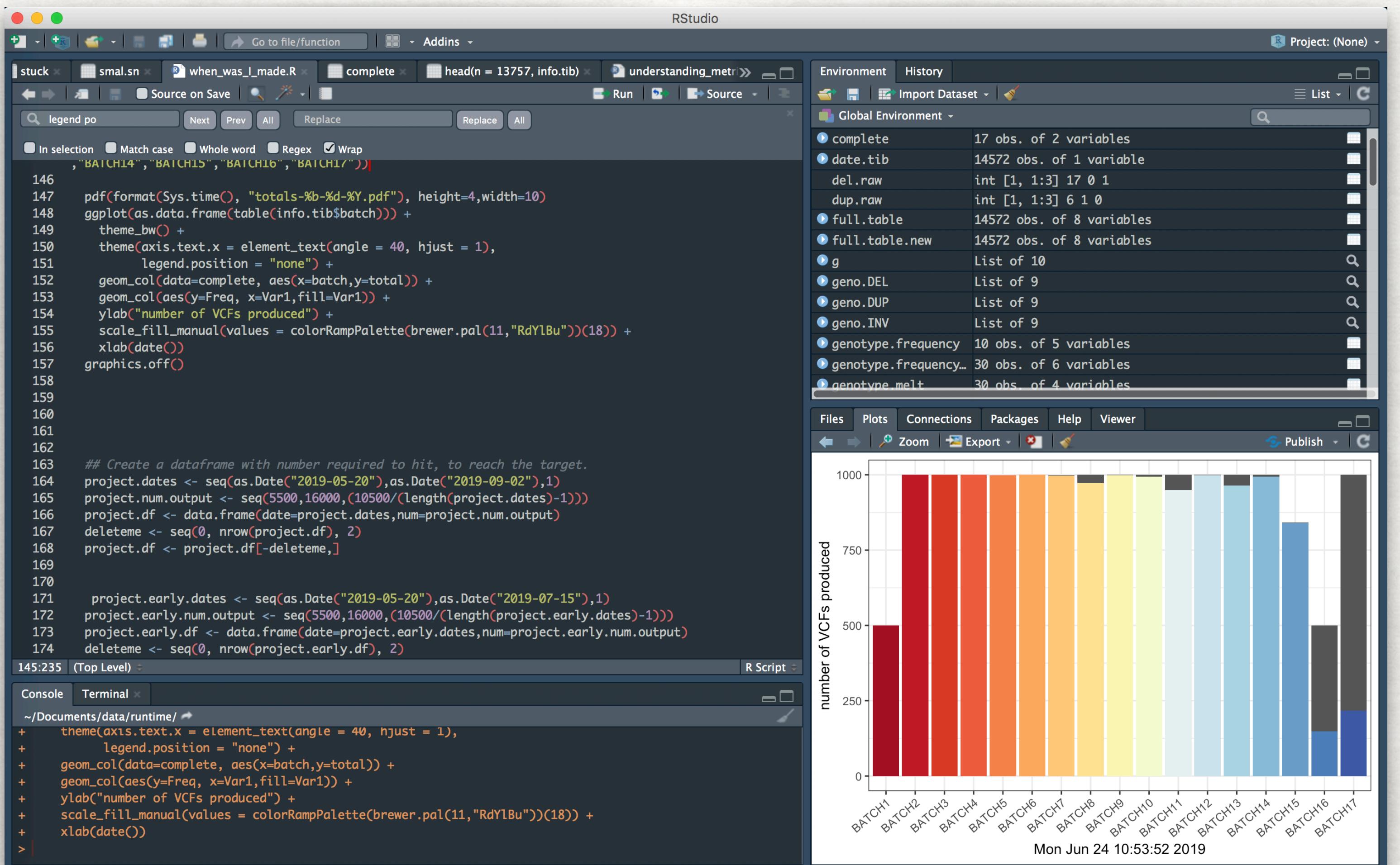
    lib_handle = py_tf.TF_LoadLibrary(library_filename, status)
    try:
        error_code = py_tf. Definitions:
            if error_code != 0: tensorflow/tensorflow/c/c_api.cc:673
            error_msg = compa tensorflow/tensorflow/c/c_api.h:1250
            # pylint: disable=protected-access
            raise errors_impl._make_specific_exception(
                None, None, error_msg, error_code)
            # pylint: enable=protected-access
    finally:
        py_tf.TF_DeleteStatus(status)
```

R STUDIO

An “IDE” (integrated development environment)

Has the console and allows plot viewing

Similar to pycharm for python



JUPYTER NOTEBOOK

Kinda an IDE, kinda a text editor??

Go to a terminal, into the directory you want.
type “jupyter notebook” to initiate the session
Start a new notebook or go to a current one



LANGUAGE: MARKDOWN

The language used to make all of my documents

<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

Very much like writing english, but with some formatting

Title

Subtitle