Work effeciently, stay organized, collaborate, an achieve your dreams with

GitHub and version control

Alison Maxwell
PUBS 2017

Alison_seq_script_2.py3 — Not registered

Currently Open Documents

- Alison_seq_script_1.py3
- Alison_seq_script_2.py3
- Alison_seq_script_3.py3

~/Desktop/UCSF/Classes/PUBS/Sequencing/Alison_seq_script_2.py3

```
import pickle
import matplotlib.pyplot as plt
import matplotlib.pylab as lab
import operator
from collections import Counter

with open('translate.pkl', 'rb') as f:
    translate = pickle.load(f)

with open('allele_dic_with_WT.pkl', 'rb') as f:
    bar_to_codon = pickle.load(f)

with open('aminotonumber.pkl', 'rb') as f:
    amino_to_number = pickle.load(f)

bar_to_aa = {key: (value[0], translate[value[1].replace('T','U')]) for key, value in bar_to_codon.items() if value[1] != 'WT'}


with open('CTGATC.fastq', 'r') as f:
    count = 0
    all_lines = []
    for line in f:
        count += 1
        if count % 4 == 2:
            all_lines.append(line.split('\n')[0])

## add all 18-bp barcodes to a list
#barcode = []
#barcoder = {}
#def ...
#   for line in all_lines:
#        barcode.append(line[0:18])
#   #barcoder(val?
#   return


#remove all barcodes that have "N" from list; add then to a new list
#N_called = []
#for element in barcode:
#    if 'N' in element:
#        barcode.remove(element)
#        N_called.append(element)


#make a new list containing all of the reverse complements of the barcodes
#needed for comparing to bar_to_aa dictionary
```

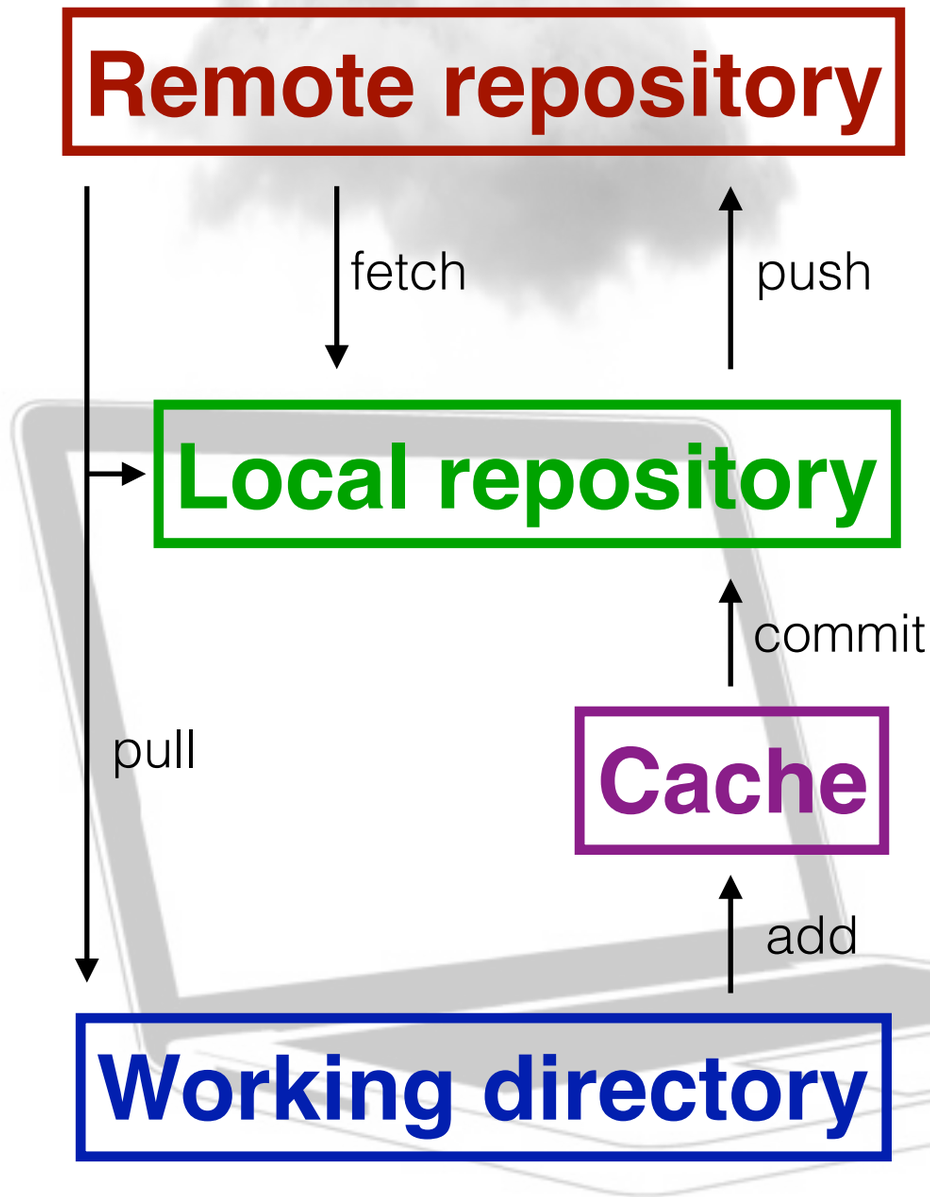
L: 27 C: 1 (none) Unicode (UTF-8) Unix (LF) Saved: 10/18/16, 3:37:01 PM 361 / 49 / 15 100%

 Version control 

- Experiment without interfering with working code
- Find when a bug arose in your code
- Maintain multiple versions of script
- Revert back to a working version of code after ruining it

GitHub: What and why

- Git: remote Version Control System (VCS)
- The Hub: a place to store your work, network, and collaborate
- Repsoitories (“repos”) for organization
- Command-line and GUI interfaces



Workflow

1. Create and share a repository
2. Create a branch—your personal version of the master project/branch
3. Make and commit changes to your branch
4. Submit and review a pull request
5. Merge the pull request with the master