Stat 610 Homework 6

Due Wednesday, October 15

Assignment

In this assignment, you'll practice using git.

You'll be working with code you wrote in the testing lab (https://jfukuyama.github.io/teaching/stat610/assignments/testing-lab.html) to go through a mock git workflow.

- Make a github repository (without adding a README or .gitignore), and clone a copy to your computer following the instructions on github. Once this is done, if you type `git status` in the terminal, you should get the output

  ```
  On branch main

  No commits yet

  nothing to commit (create/copy files and use "git add" to track)
  ```

- Add an initial R file called `simulate_reads.R` and a file called `test_simulations.R` that contain the initial code given in the testing lab.

  Once you've created these files, but before you've added them to the staging area or commited them, if you type `git status`, you should see something like

  ```
  On branch main

  No commits yet

  Untracked files:
    (use "git add <file>..." to include in what will be committed)

   simulate_reads.R
   test_simulations.R

  nothing added to commit but untracked files present (use "git add" to track)
  ```

  Once you're at this stage, add the two files to the staging area using `git add simulate_reads.R`, `git add test_simulations.R`.

  At this point, if you type `git status`, you should see something like

  ```
  On branch main

  No commits yet
  ```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

 new file:   simulate_reads.R
 new file:   test_simulations.R
```

Finally, commit these files by typing `git commit -m ’initial commit’`. Feel free to make a better commit message.

Once you have done that, typing `git status` should give you output

```
On branch main
nothing to commit, working tree clean
```

Question 1: What does your commit history look like now? What branches do you have, and what commits are they pointing to? Where does `HEAD` point? (You havent done very much yet; this question is mostly to give you a reference to compare against later.)

- Make a new branch called `specify-num-nonzero` by typing `git branch specify-num-nonzero`. The idea is going to be that this branch is going to contain code that is modified so that you can specify the number of sequences that have a non-zero value for each site.

  You can check that you made the branch by typing `git branch`, which should give you output

  ```
  * main
    specify-num-nonzero
  ```

  which indicates that you have two branches, one called `main` and one called `specify-num-nonzero`, and that `main` is *checked out*, i.e., `HEAD` points to main.

  You can also check this by running `git log -graph -branches`, and you may find it helpful to run this command periodically throughout the assignment.

- Check out `specify-num-nonzero` by typing `git checkout specify-num-nonzero`.

  Question 2: What changed when you checked out `specify-num-nonzero`? Where does `HEAD` point now? If you make changes and commit them, where will the `main` and `specify-num-nonzero` branches point?

- Modify the `test_simulations.R` file so that it includes the tests for the function that allows specifying the number of nonzero entries per site.

  Add this file to the staging area using `git add` and then commit it with `git commit -m ’your commit message’`.

  Question 3: What does your commit history look like now? What branches do you have, and what commits are they pointing to? Where does `HEAD` point?

- Modify the `simulate_reads.R` file so that it includes the new version of `make_true_seqs` that allows specifying the number of nonzero entries per site. Add the file to the staging area and commit the changes.

- **Question 4**: Run `git log -graph -branches` What output do you get? What does it tell you about the commit history?
- **Question 5**: If you run `git checkout main`, what happens to the contents of the files `simulate_reads.R` and `test_simulations.R`? (You might have to close and re-open the files in RStudio, because running the checkout command will change the files on the disk). How would you get back the code you wrote in the `specify-num-nonzero` branch?
- After you are done, run `git push origin main` and `git push origin specify-num-nonzero`.

Submission parameters

Submit a document with the answers to the bold-faced questions and a link to your github repository.

Your github repository should have branches `main` and `specify-num-nonzero`.