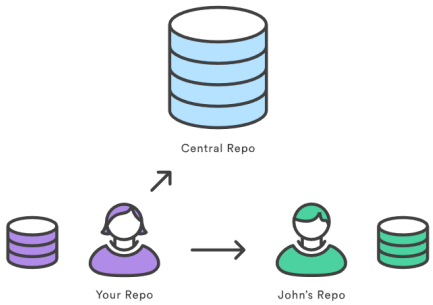
A few other general debugging questions

I've answered as many of these as I could, but it might be helpful to work through the rest if the submodule fix doesn't entirely fix things.

Ignore the arrow going from "you" to "John" in this diagram. Where is your understanding of where the "Central Repo" is (referred to as "remote" from here in)? Can we take a look at it where it is hosted?



It's on GitHub. You were able to show me the repo in screenshare. Here's where Justin's Git config thinks it is:

```
https://github.com/jhite-eco-epi/HiteLab.git
```

Is that correct?

Is everyone in the team clear on the difference between "commit" and "push"?

It sounds like you and Justin are, but the way the lab manager uses Git is different. It would be worth checking how you normally get hold of the files she uploads. You don't want to be downloading them manually if they are committed in the repo, otherwise you'll have local changes that a also in the remote and this will confuse Git, and might end up with fun messages about detached heads. Try making sure you always keep up with what's in the repo by using `git pull`, not by doing any manual downloads.

How did you and your post doc get hold of the repo in the first place?

- Cloned it from the remote / copied from flash drive or Google Drive / other approach

I'd recommend only ever getting hold of things via `git clone`. Otherwise you could inherit confusing info from your config files.

What happens when they add, commit and push?

Try doing this within GitBash (the more git-friendly command line interfact). Here's what it looks like for me in my "sandbox" repo:

```
Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.Rmd
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git add README.Rmd

Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git add README.md

Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.Rmd
        modified:   README.md


Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git commit -m "Added instructions on creating a branch"
[main c0347d1] Added instructions on creating a branch
 2 files changed, 37 insertions(+), 5 deletions(-)

Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git push
Enumerating objects: 7, done.
```

- What branch name do you see? (c.f. `(main)` in the image above)
- What happens when you run `git status`?
- What about `git status -u`? (to check for untracked files; files that Git doesn't yet know about)
- What about `git pull`?

“Have you tried turning it off and on again?”

A reference to a classic British comedy called the IT Crowd. This is their answer to most things. And it’s actually not a bad one sometimes.

In our case, it translates to “have you tried cloning the repo fresh?”. It sounds like you have, but that didn’t fix things. What happened then? Did you definitely delete (or move!) everything (including any hidden folders) before you did the clone? Try cloning to a completely different location and see what happens then.

I’ll also try cloning the repo to my laptop once you grant me access and see what happens.

The system is left hanging, unable to resolve differences

What happens when you do a commit and push?

You should get something like this:

```
Cara@DESKTOP-9NINTLP MINGW64 ~/[redacted]/sandbox (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 996 bytes | 996.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/cararthompson/sandbox.git
   1af3f19..c0347d1  main -> main
```

If that doesn't happen, check for network issues; you may need assistance with IT for this!

Moving files

It's absolutely fine to drag and drop and the tell Git what you've done. All you need to do is `add` the deleted file (which is just telling it that you've deleted it) and then `add` the new file. It will then know that the file has just moved, so when you next call `git status` it will show the file as `renamed`. You can then commit and push, and everyone should have things where they need to be.

A quicker, but more dangerous way to do this, is to add everything within a folder. To do that, navigate to the folder and then use:

```
git add -A .
```

You'll want to then run `git status` to check you're happy with everything before you commit! If you're not, you'll need to "unstage" the things you've just "staged" (`add`ed - but googling "unstage" will get you the answer you need!

Git config doesn't know where "remote" is

Let's check the remote is the same across everyone's setup.

```
git remote show origin
```

The users are getting stuck in Vim and not completing the pull/merge

To exit Vim, type:

```
:wq
```

To change to a more familiar text editor:

```
git config --global core.editor path/to/editor
```

Then check that worked using:

```
$EDITOR
```

This should open your chosen editor if the path is correctly specified.