THE UNIVERSITY *of* EDINBURGH

Using Git and GitHub for Team Collaboration

Fei Yao
28/02/2022

Git and GitHub

- Git and GitHub are not the same thing.
  - Git is an open-source, version control tool created in 2005 by developers working on the Linux operating system;
  - GitHub is a company founded in 2008 that makes tools which integrated git.
- You do not GitHub to use git, but you cannot use GitHub without using git.
- Alternatives to GitHub include GitLab, BitBucket, and etc. All of these are referred, in git-speak, as “remotes” that will make sharing code with others easier.

An example workflow

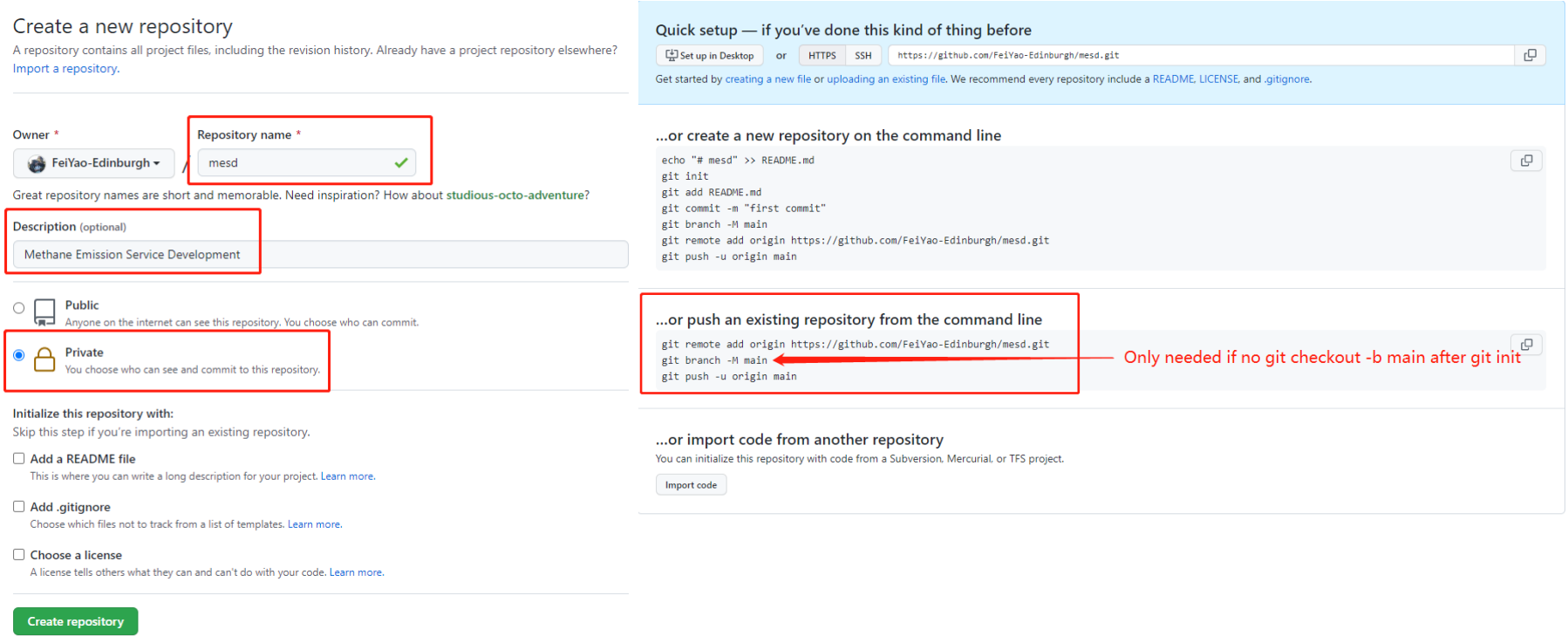
- Create a local git repository.
  - cd
  - mkdir mesd # methane emission service development
  - cd mesd
  - git init
  - git checkout -b main # primary branch name

An example workflow

- Add a new file (README.md) to the repo.
  - touch README.md
  - git status # git has noted the file but won't track it unless we explicitly add it in a commit.
- Add the file to the staging environment.
  - git add README.md
  - git status # git has added the file to a staging environment but not a commit although it's about to.
- Package all files added to the staging environment to a commit.
  - git commit -m 'First commit.' # leaving a clear explanation of your changes will be extremely helpful for future programmers (perhaps future you!)
  - git log # see that the commit has been added to the log

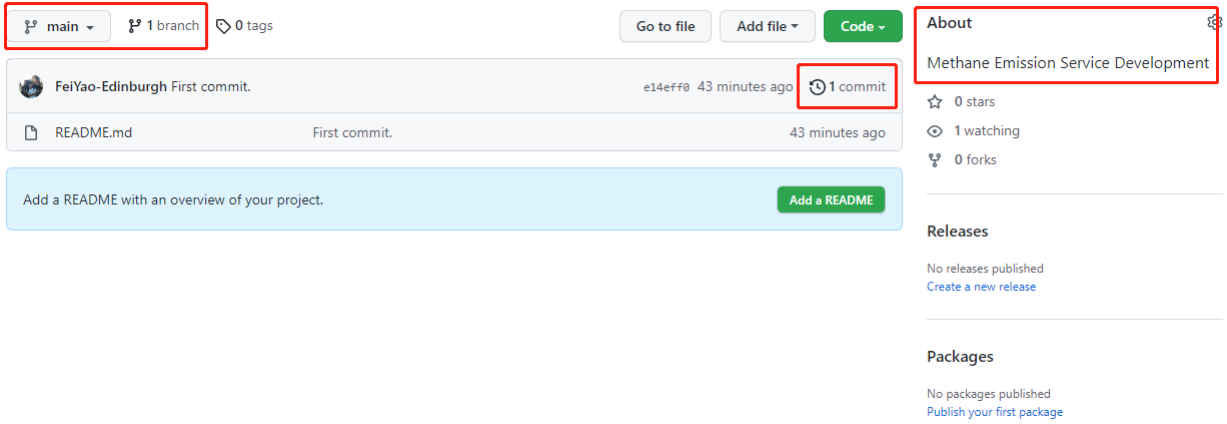
An example workflow

- Create a remote repo.



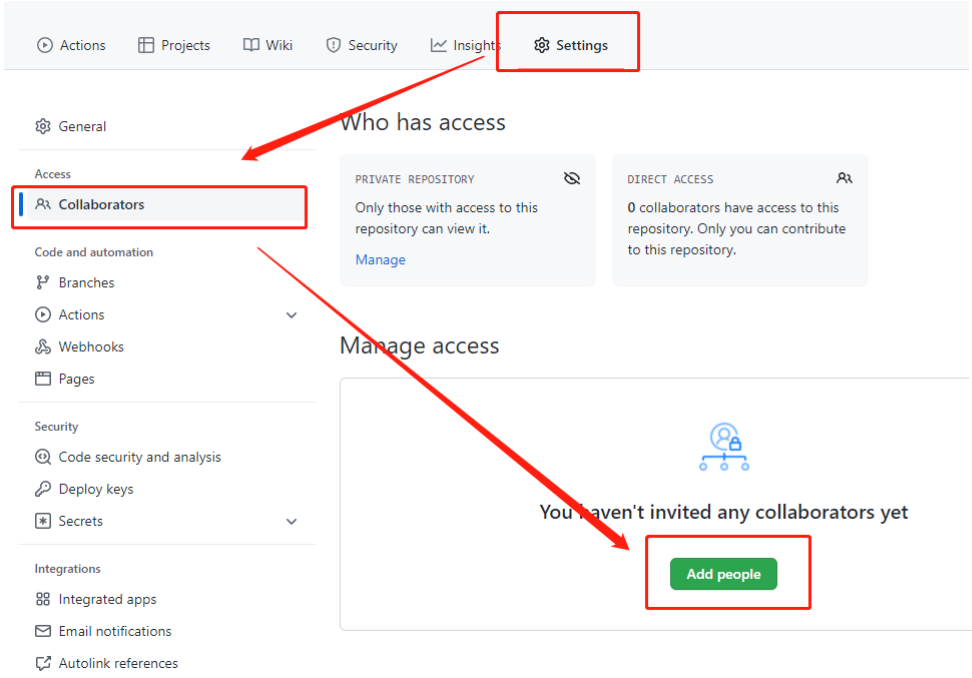
An example workflow

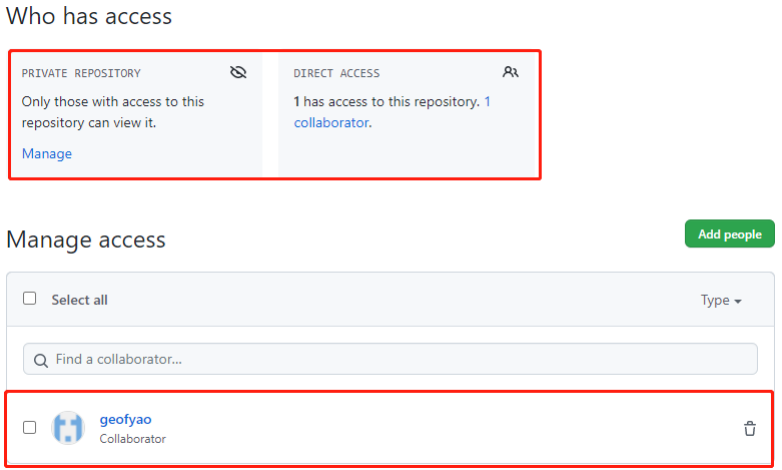
- Link local and remote repo.
  - git remote add origin https://github.com/FeiYao-Edinburgh/mesd.git # Replace where applicable.
  - git push origin main # origin is an alias of remote repo!
  - git remote -v



An example workflow

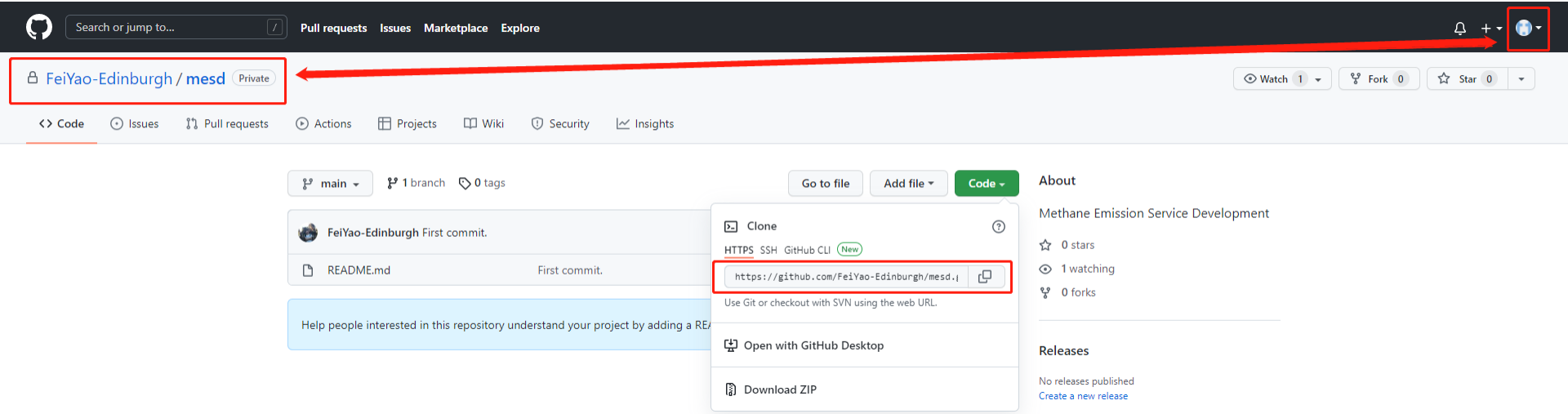
- Add collaborators.





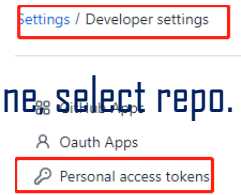
An example workflow

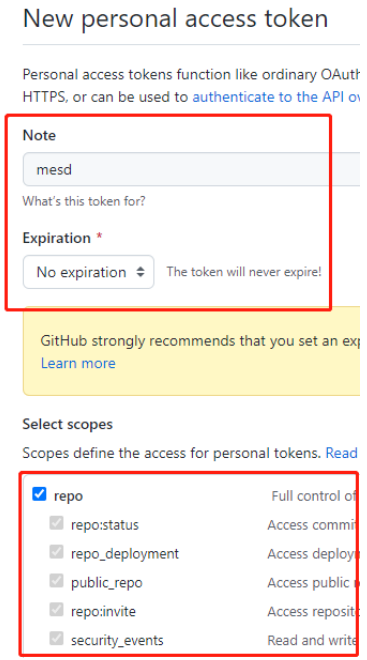
- The collaborators can access the repo directly from their GitHub page.



An example workflow

- The collaborators can clone the repo to their local machines using their username and personal access token (PAT).
  - To use your token to access repositories from the command line, select repo.
  - Ensure you copy your PAT to a safe place.
  - git clone https://github.com/FeiYao-Edinburgh/mesd.git
  - cd mesd






```
[s1855106@renjie s1855106]$ git clone https://github.com/FeiYao-Edinburgh/mesd.git
Cloning into 'mesd'...
Username for 'https://github.com': geofyao
Password for 'https://geofyao@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

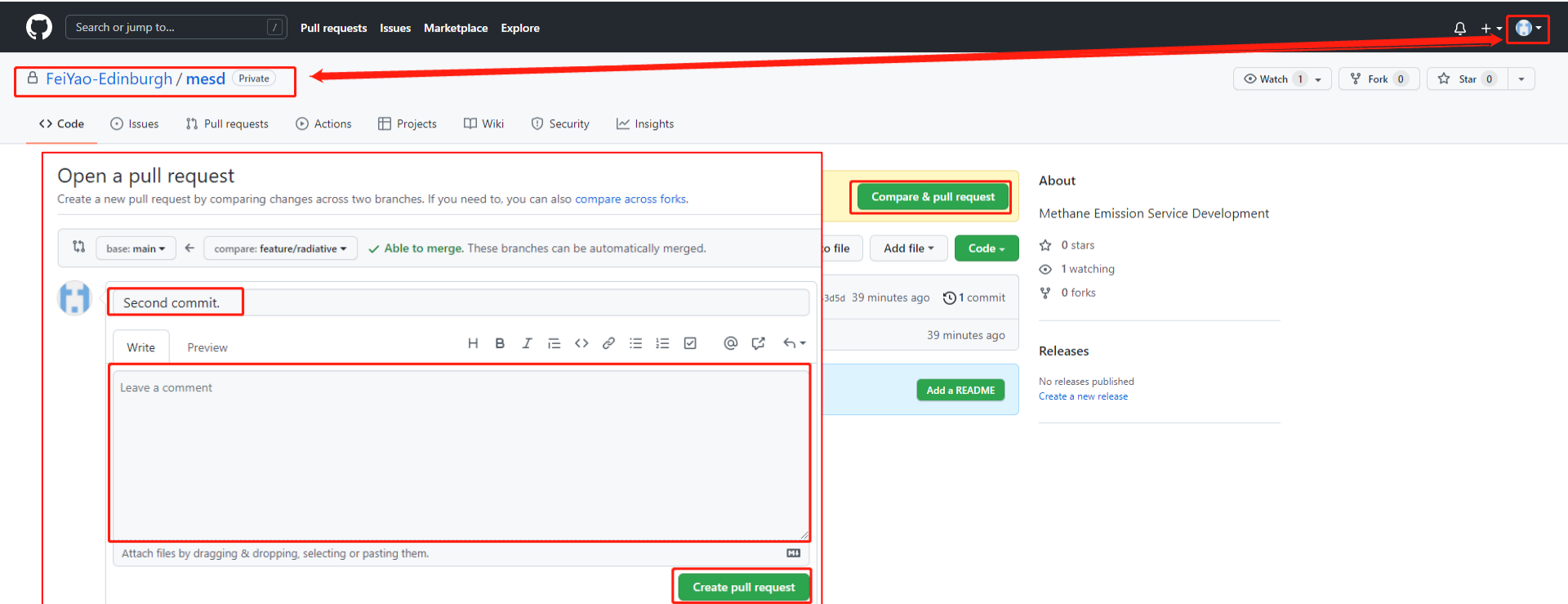
Type PAT here

An example workflow

- The collaborators create a new branch where they will develop codes **without affecting the main branch**.
  - git branch feature/radiative # Better to use a branch name indicating what the collaborators are working.
  - git checkout feature/radiative # Switch to the branch
  - # Can combine the above two lines of codes to git checkout -b feature/radiative
  - mkdir -p radiative # Now make some changes
  - cd radiative
  - touch README.md # Each subdir can have its own README.md
  - git add --all
  - git commit -m 'Second commit.' # can make as many commits as needed and then push them together!
  - git pull origin feature/radiative # In case new commits are made by others
  - git push origin feature/radiative

An example workflow

- The collaborators make a pull request (PR) from their GitHub page.



An example workflow

- The repo owner will be notified that the collaborators have made a PR.

[FeiYao-Edinburgh/mesd] Second commit. (PR #1)

Some content in this message has been blocked because the sender isn't in your Safe senders list.

geofyao <notifications@github.com>
Tue 22/02/2022 18:56
To: FeiYao-Edinburgh/mesd <mesd@noreply.github.com>
Cc: Subscribed <subscribed@noreply.github.com>

This email was sent to you by someone outside the University.
You should only click on links or attachments if you are certain that the email is genuine

You can view, comment on, or merge this pull request online at:

https://github.com/FeiYao-Edinburgh/mesd/pull/1

Commit Summary

- f602758 Second commit.

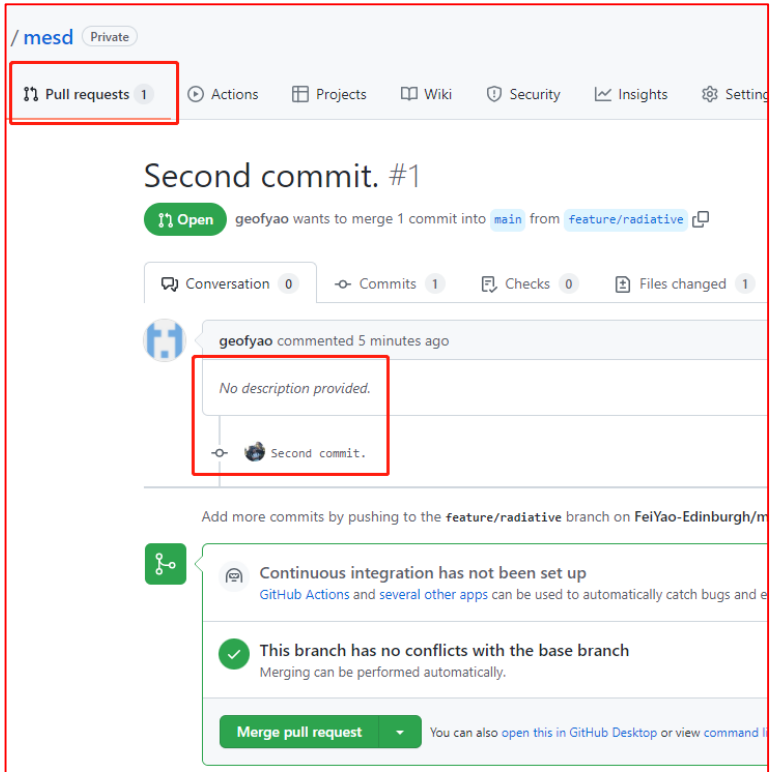
File Changes(1 file)

- **A** radiative/README.md (0)

Patch Links:

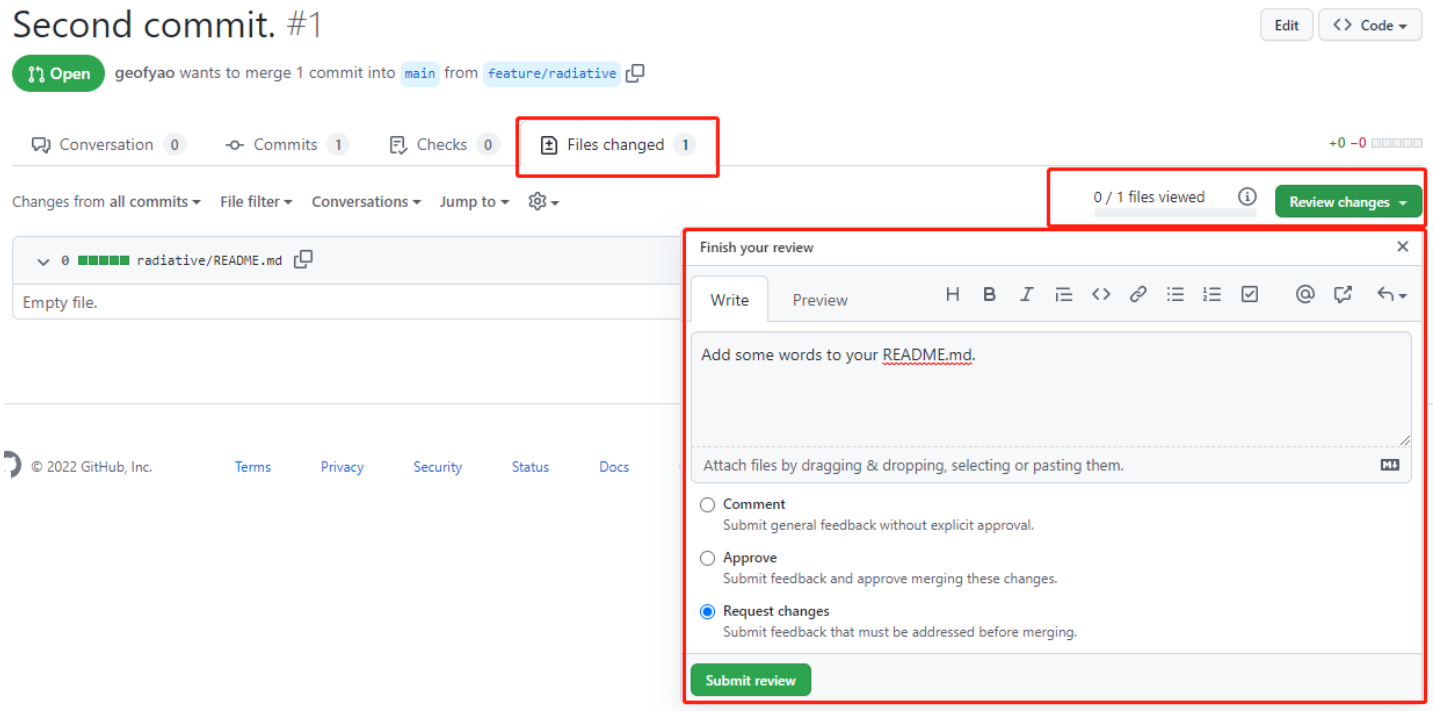
- https://github.com/FeiYao-Edinburgh/mesd/pull/1.patch
- https://github.com/FeiYao-Edinburgh/mesd/pull/1.diff

—
Reply to this email directly, view it on GitHub, or unsubscribe.
Triage notifications on the go with GitHub Mobile for iOS or Android.
You are receiving this because you are subscribed to this thread.



An example workflow

- The repo owner review the changes and may make additional requests.



An example workflow

- The collaborators make additional changes per request and re-push.
  - vi README.md
  - git add --all
  - git commit -m 'Third commit.'
  - git push origin feature/radiative

```
[s1855106@renjie radiative]$ git push origin feature/radiative
Username for 'https://github.com': geofyao
Password for 'https://geofyao@github.com':
Counting objects: 7, done.
Delta compression using up to 96 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 373 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/FeiYao-Edinburgh/mesd.git
   f602758..94a2c3a  feature/radiative -> feature/radiative
```

Re: [FeiYao-Edinburgh/mesd] Second commit. (PR #1)

From : "Fei Yao" <notifications@github.com> (Forward by noreply@github.com)

To : "FeiYao-Edinburgh/mesd" <mesd@noreply.github.com>

Cc : "姚飞" <yaofei@pku.edu.cn> "Author" <author@noreply.github.com>

@FeiYao-Edinburgh requested changes on this pull request.

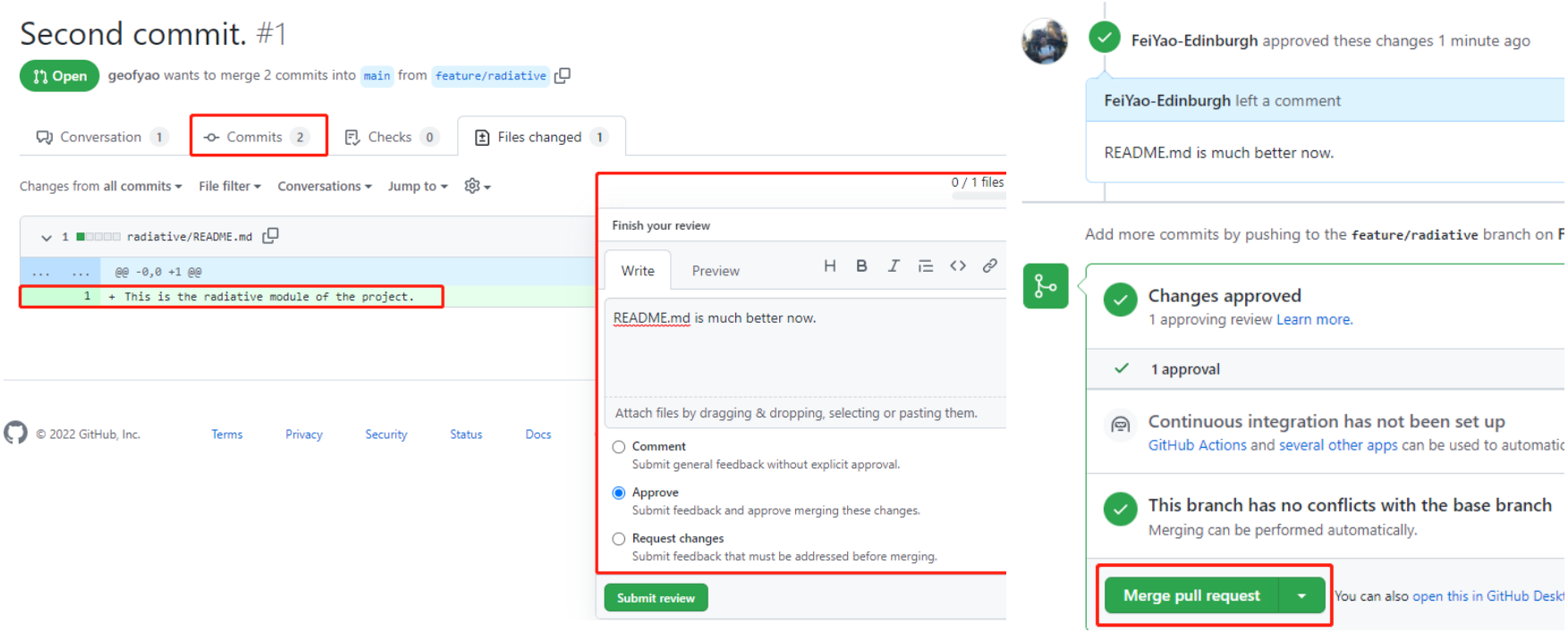
Add some words to your README.md.

—

Reply to this email directly, view it on GitHub, or unsubscribe.
Triage notifications on the go with GitHub Mobile for iOS or Android.
You are receiving this because you authored the thread.

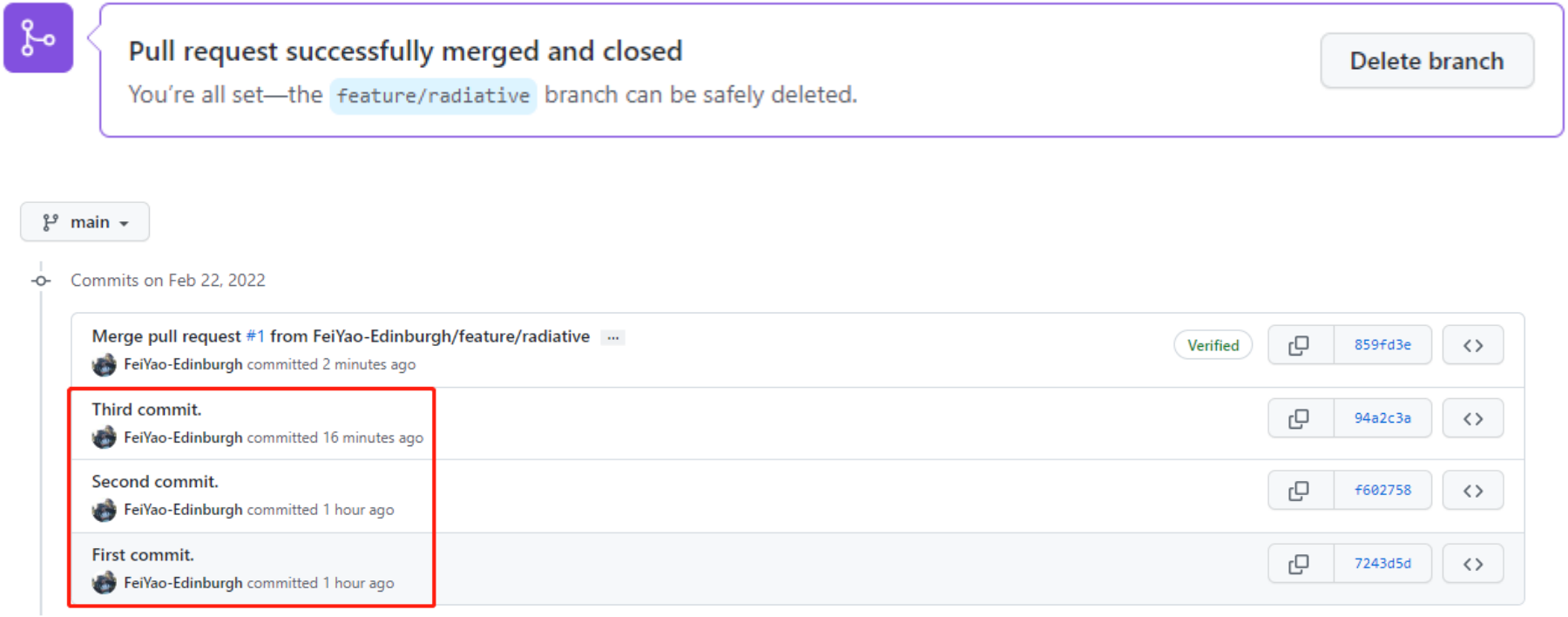
An example workflow

- The repo owner now sees and approves the changes.



An example workflow

- The repo owner always deletes branches that are already merged into the main branch.

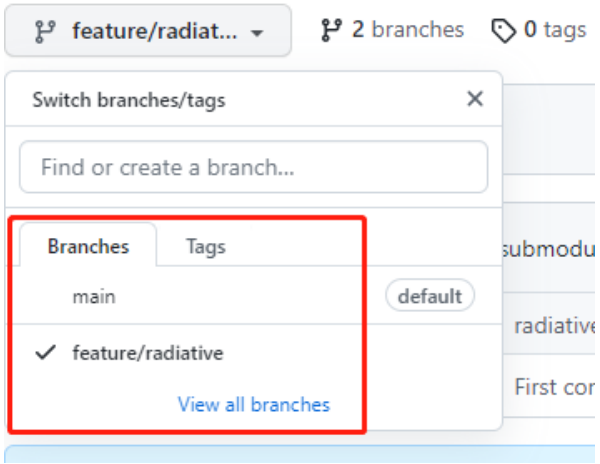


An example workflow

- The collaborators sync the remote changes back to their local machines and start next run. All collaborators behave similarly.
  - cd .. # After switch to main radiative no longer exists.
  - git checkout main # Always know where you are before running git-related codes.
  - git branch -d feature/radiative # Remove already done branches just like the remote operation.
  - git pull origin main # This is always recommended to keep update with remote!
  - git log # To see completely identical as remote.
  - git checkout -b new-branch # New branch to work on

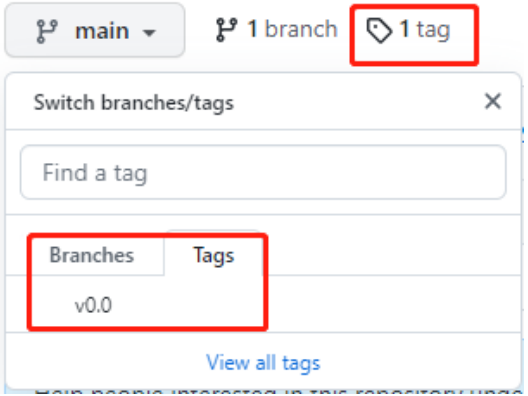
Recap

- A common git and GitHub collaboration workflow:
  - Fetch and merge changes from the remote (git pull).
  - Create a branch to work on a new project feature.
  - Develop the feature on the branch and commit the work.
  - Fetch and merge from remote again (in case new commits were made).
  - Push branch up to the remote for review
- A safeguard against merge conflicts:
  - Collaborators work on separate subdirs.



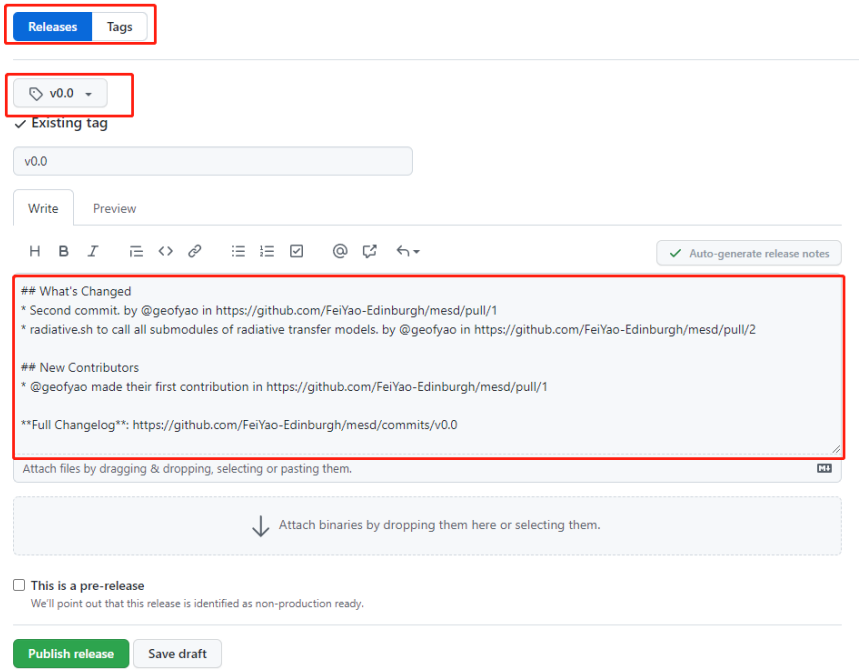
Tags: keeping track of project milestones

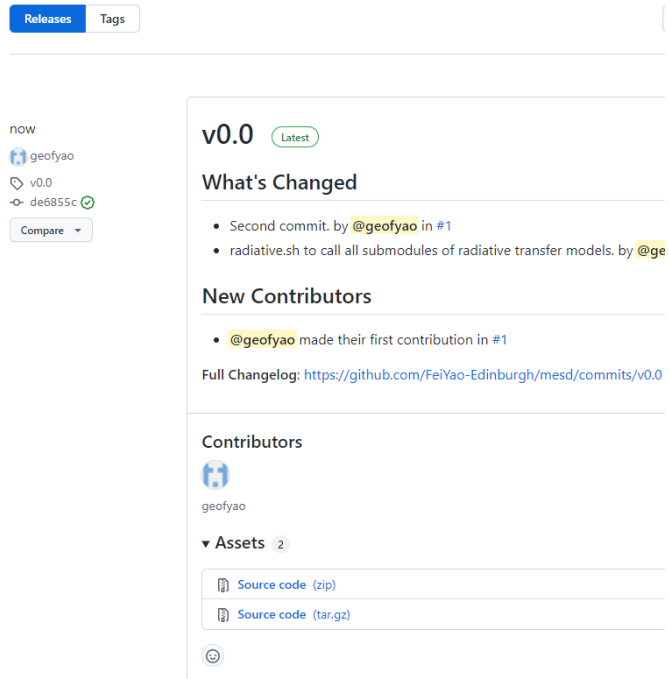
- A tag is similar to a branch; it's simply a lightweight moveable pointer to one of the commits.
  - git tag --list
  - git tag -a v0.0 –m "The 0.1 version."
  - git show v0.0
  - git push origin v0.1 # Always push code first.
  - # Create and checkout a new branch "feature/xxx" at tag "v0.0".
  - git checkout -b feature/xxx v0.0
  - git tag -d v0.0 # Can delete unwanted local tags.
  - git push origin --delete v0.0 # Can delete unwanted remote tags.



Release

- Releases, based on tags, are deployable software iterations you can package and make available for a wider audience to download and use.



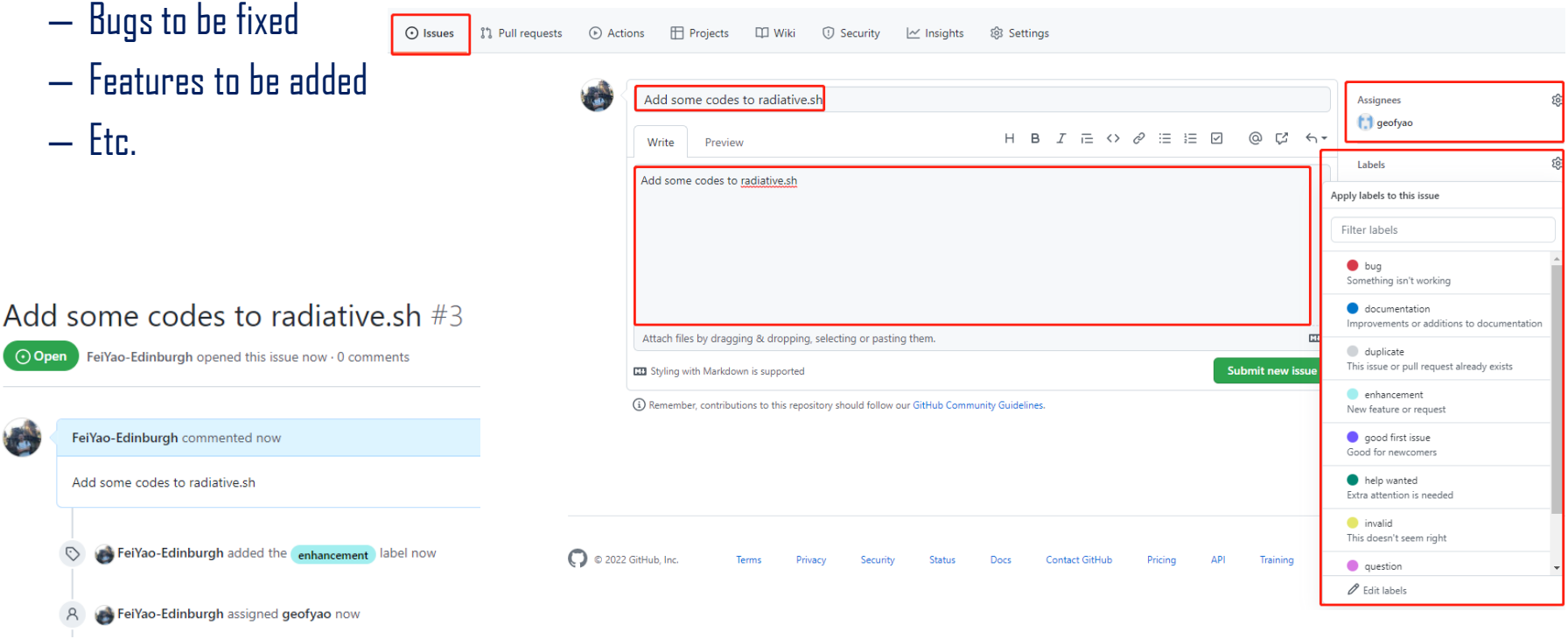


.gitignore: prevent unwanted file types

- Can reside in root and subdirs
- Prevent intermediate/backup files.
  - .ipynb_checkpoints
  - __pycache__
- Prevent large data files and so forth.
  - *.nc
  - *.nc4
  - Etc.
- Etc.

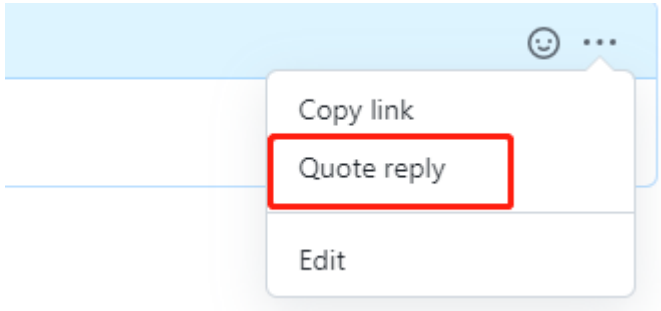
Issues

- Assign specific issues/tasks to specific collaborators:
  - Bugs to be fixed
  - Features to be added
  - Etc.



Issues

- Make scientific discussions
  - Simple but powerful markdown
  - Quote reply



Noting down some thoughts about mach

- Not sure if the current model can
- ML in GEE seems to require outlin
  classification
- Might need to retrain a new mod
  - This will require a new tra
  - We would need to outline
- This would produce a more usefu
- Allows further development of th
  proximity to infrastructure etc)
- Current training data set will look
  data that isn't seen in TROPOMI

*Training a new model:

- Labour intensive but possible short cuts
- If we use the old dataset of images (which I still have), upload them to GEE then outline them we don't have to search for new plumes
- We would need to manipulate the images to be uploaded (e.g., provide fake coordinates & times) and can then draw round the plumes using GEE
- Don't need 'no plume' examples as it'll just use the surrounding areas
- Possibly more errors involved because we need to decide the extent of a plume

3.2 Multi-band–single-pass (MBSP) retrieval

Our second method is a multi-band retrieval that estimates methane enhancements from differences between the band-11 and band-12 reflectances measured on a single satellite pass. For this method, the fractional change in reflectance is given by

$$\Delta R_{\mathrm{MBSP}} = \frac{cR_{12} - R_{11}}{R_{11}}, \tag{3}$$

where c is now determined by least-squares fitting of R_{12} against R_{11} across the scene. The fractional absorption model is then

$$m_{\mathrm{MBSP}}(\Delta\Omega) = \frac{T_{12}(\Omega+\Delta\Omega) - T_{12}(\Omega)}{T_{12}(\Omega)} - \frac{T_{11}(\Omega+\Delta\Omega) - T_{11}(\Omega)}{T_{11}(\Omega)}. \tag{4}$$

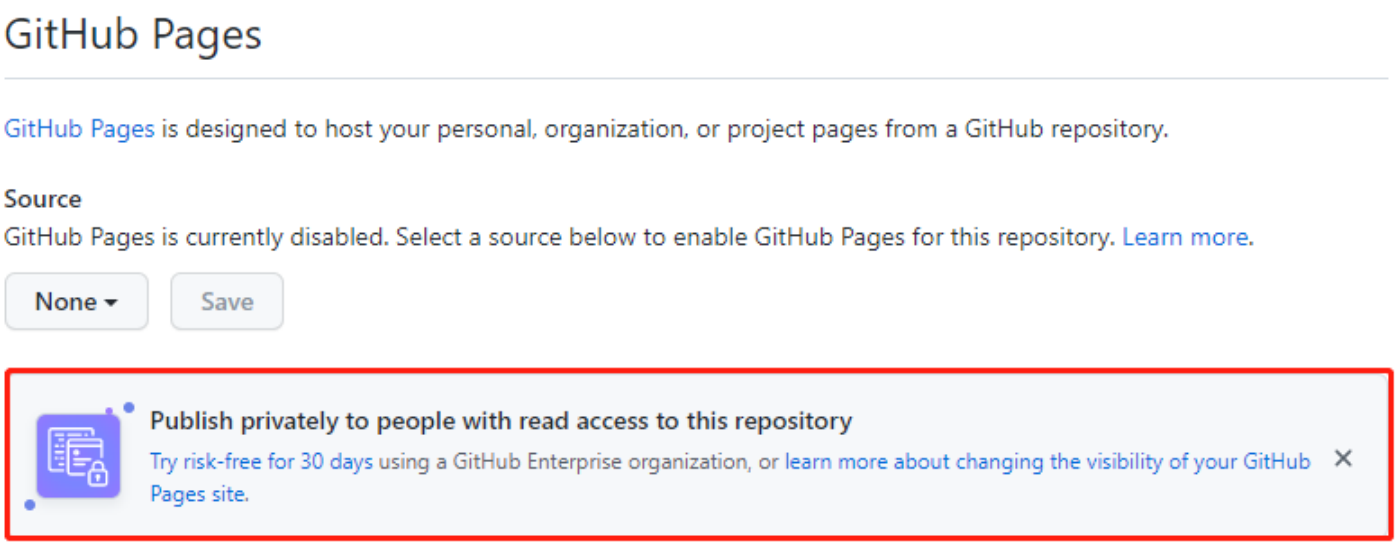
This approach relies on surface reflectance similarities between the two adjacent bands. The empirical scaling fac-

Licence

- The LICENSE, LICENSE.md, or LICENSE.txt file is often used in a repository to indicate how the contents of the repo may be used by others.
  - The license is best chosen from the get-go, even if for a repository that is not public.
  - Choosing a license that is in common use makes life easier for contributors and users, because they are more likely to already be familiar with the license and don't have to wade through a bunch of jargon to decide if they're ok with it.
  - People who incorporate General Public License (GPL'd) software into their own software must make their software also open under the GPL license; most other open licenses do not require this.
  - People who are not lawyers should not try to write licenses from scratch.

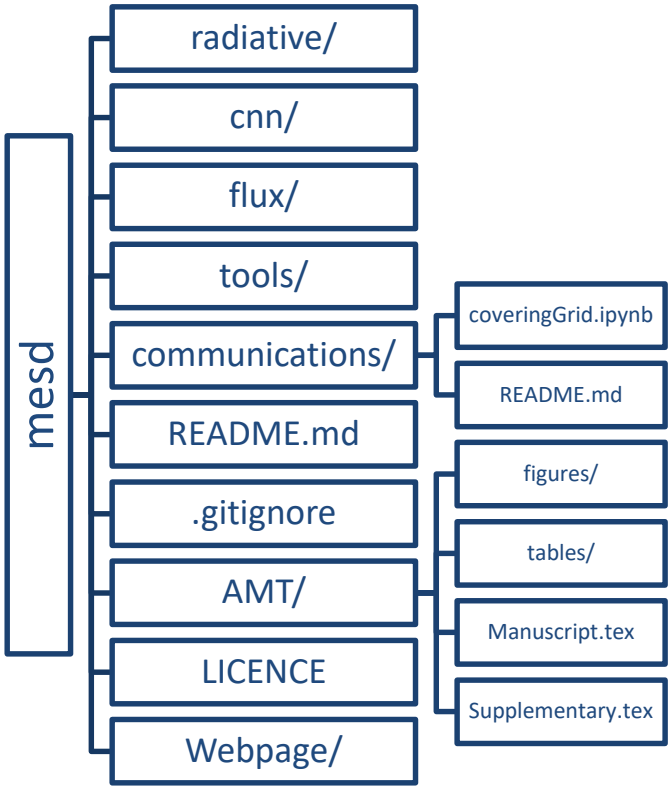
GitHub Pages

- GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.
  - Public versus private



The organization of the project

- Top-level design is pivotal
  - The entire project can be linked to an Overleaf project. This way you have all the codes including Latex scripts within one place!
  - Or we can make a subdir a repo and link it to an Overleaf project, but this will involve complicated concepts and operations.
  - methane-emission-service as repo name?



Reference

- An Intro to Git and GitHub for Beginners (Tutorial)
  - https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
- Version Control with Git
  - https://swcarpentry.github.io/git-novice/
- Using Git Tags To Version Coding Tutorials
  - https://medium.com/@emmabostian/using-git-tags-to-version-coding-tutorials-cf9ff28fad4f