



THE UNIVERSITY *of* EDINBURGH

Using Git and GitHub for Team Collaboration

Fei Yao
28/02/2022

<https://feiyao-edinburgh.github.io/>

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.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

FeiYao-Edinburgh

Repository name *

mesd

Great repository names are short and memorable. Need inspiration? How about [studious-octo-adventure](#)?

Description (optional)

Methane Emission Service Development

☐ Public

Anyone on the internet can see this repository. You choose who can commit.

☒ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Quick setup — if you've done this kind of thing before

or

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# mesd" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/FeiYao-Edinburgh/mesd.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/FeiYao-Edinburgh/mesd.git
git branch -M main
git push -u origin main
```

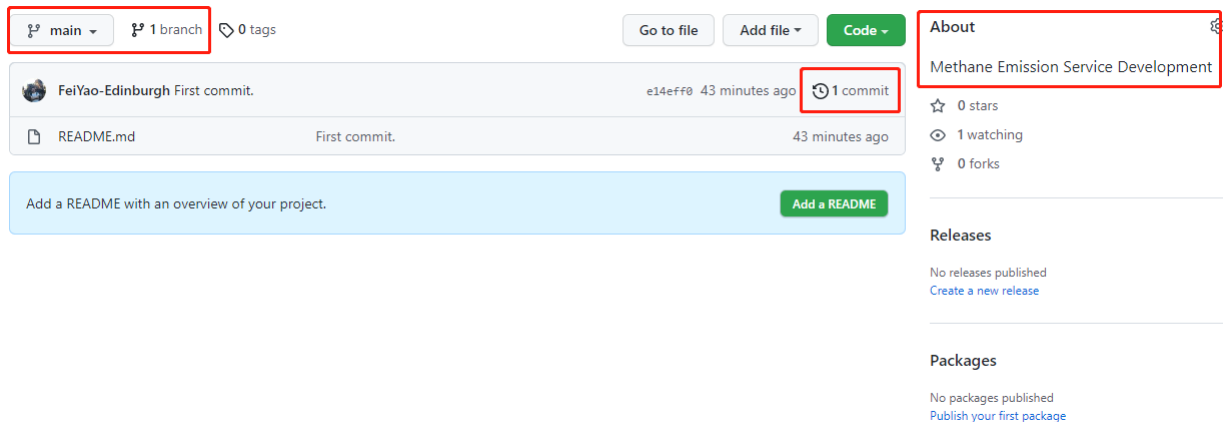
Only needed if no git checkout -b main after git init

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

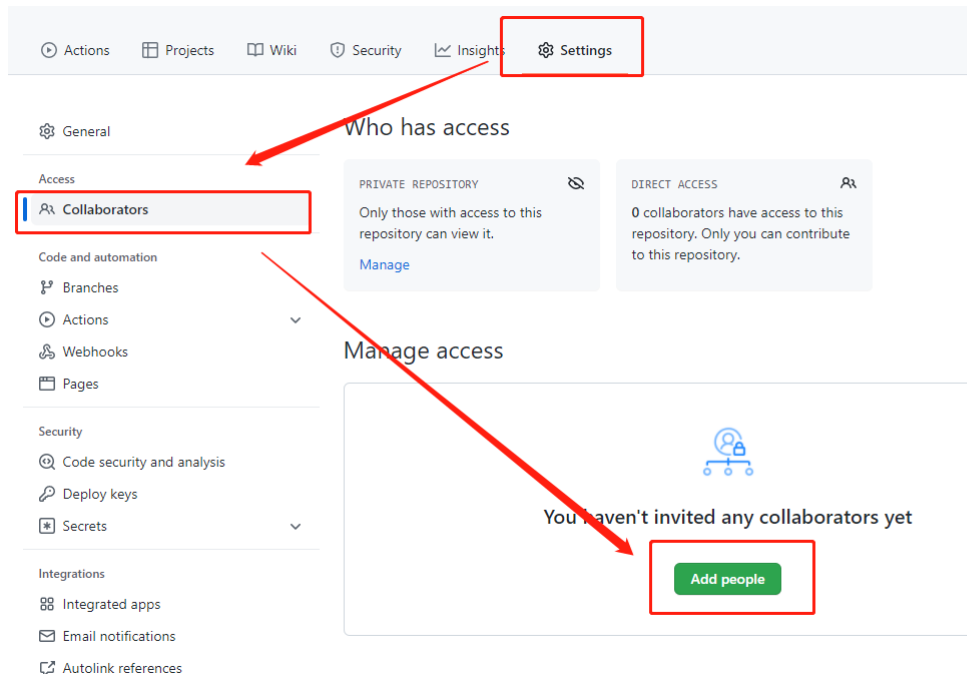
An example workflow

- 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.



The screenshot shows the GitHub repository settings page. The 'Settings' tab is selected in the top navigation bar. In the left sidebar, the 'Collaborators' tab is highlighted. The main content area shows the 'Who has access' section with two cards: 'PRIVATE REPOSITORY' and 'DIRECT ACCESS'. Below this is the 'Manage access' section, which displays a message: 'You haven't invited any collaborators yet' and a green 'Add people' button. Red arrows indicate the workflow: from the 'Settings' tab to the 'Collaborators' tab, and then to the 'Add people' button.

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS

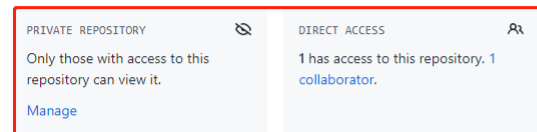
0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

[Add people](#)

Who has access



The screenshot shows the 'Who has access' section. It contains two cards: 'PRIVATE REPOSITORY' and 'DIRECT ACCESS'. The 'PRIVATE REPOSITORY' card states: 'Only those with access to this repository can view it.' and has a 'Manage' link. The 'DIRECT ACCESS' card states: '1 has access to this repository. 1 collaborator.' and has a 'Manage' link. A red box highlights the entire section.

PRIVATE REPOSITORY

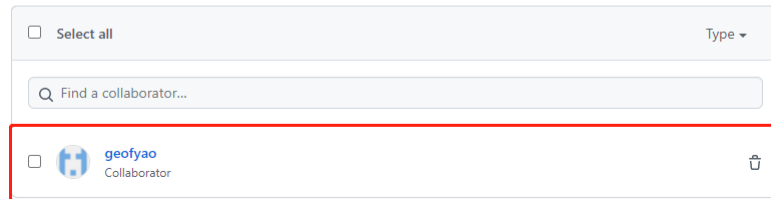
Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS

1 has access to this repository. 1 collaborator.

Manage access




The screenshot shows the 'Manage access' section. It features a search bar with the placeholder text 'Find a collaborator...'. Below the search bar, there is a list of collaborators. The first entry is 'geofyao' with the role 'Collaborator'. A red box highlights the entire section.

[Add people](#)

☐ Select all

Type ▾

Find a collaborator...

☐  geofyao
Collaborator

An example workflow

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

The screenshot shows the GitHub interface for the repository 'FeiYao-Edinburgh / mesd'. A red box highlights the repository name in the top navigation bar. A red arrow points from the user profile icon in the top right corner to this repository name. Another red box highlights the 'Clone' button in the 'Code' dropdown menu, which is open. The 'Clone' dropdown shows the HTTPS URL 'https://github.com/FeiYao-Edinburgh/mesd.' with a copy icon next to it. The repository page also shows the 'main' branch, 1 branch, 0 tags, and a commit by 'FeiYao-Edinburgh'.

Search or jump to...

Pull requests Issues Marketplace Explore

FeiYao-Edinburgh / mesd Private

Watch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

FeiYao-Edinburgh First commit.

README.md First commit.

Help people interested in this repository understand your project by adding a README file.

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI New

https://github.com/FeiYao-Edinburgh/mesd. Copy

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Methane Emission Service Development

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

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

Settings / Developer settings

GitHub repo

Oauth Apps

Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth2 tokens, or can be used to [authenticate to the API](#) on

Note

mesd

What's this token for?

Expiration *

No expiration

The token will never expire!

GitHub strongly recommends that you set an expiration date for your token. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read](#)

<input checked="" type="checkbox"/>	repo	Full control of repository
<input type="checkbox"/>	repo:status	Access commit status
<input type="checkbox"/>	repo:deployment	Access deployment
<input type="checkbox"/>	public_repo	Access public repositories
<input type="checkbox"/>	repo:invite	Access repository invitations
<input type="checkbox"/>	security_events	Read and write security events

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.

The screenshot displays the GitHub interface for creating a pull request. The repository is **FeiYao-Edinburgh / mesd**. The main section is titled **Open a pull request** and includes a comparison between **base: main** and **compare: feature/radiative**, with a status of **✓ Able to merge**. The **Second commit.** input field is highlighted. The **Create pull request** button is at the bottom right. The right sidebar shows the **Compare & pull request** button, the repository name, and the **Releases** section.

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.

/ mesd Private

Pull requests 1 Actions Projects Wiki Security Insights Settings

Second commit. #1

Open geofyao wants to merge 1 commit into `main` from `feature/radiative`

Conversation 0 Commits 1 Checks 0 Files changed 1

geofyao commented 5 minutes ago

No description provided.

Second commit.

Add more commits by pushing to the `feature/radiative` branch on FeiYao-Edinburgh/m

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and e



✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command l

An example workflow

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

Second commit. #1

 **Open** geofyao wants to merge 1 commit into `main` from `feature/radiative` 


Edit

<> Code ▾


 Conversation 0

 Commits 1

 Checks 0

 Files changed 1



+0 -0 

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ 

0 / 1 files viewed



Review changes ▾

▼ 0  radiative/README.md 

Empty file.

Finish your review



Write

Preview

H

B

I

≡

<>









@



← ▾

Add some words to your README.md.

Attach files by dragging & dropping, selecting or pasting them. 

☐ **Comment**

Submit general feedback without explicit approval.

☐ **Approve**

Submit feedback and approve merging these changes.

☒ **Request changes**

Submit feedback that must be addressed before merging.

Submit review

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.

Second commit. #1

geofyao wants to merge 2 commits into `main` from `feature/radiative`

Conversation 1 **Commits 2** Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to ⚙

1 radiative/README.md

... @@ -0,0 +1 @@

1 + This is the radiative module of the project.

Finish your review

Write Preview H B I

README.md is much better now.

Attach files by dragging & dropping, selecting or pasting them.

☐ Comment
Submit general feedback without explicit approval.

☒ **Approve**
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

FeiYao-Edinburgh approved these changes 1 minute ago

FeiYao-Edinburgh left a comment

README.md is much better now.

Add more commits by pushing to the `feature/radiative` branch on F

Changes approved
1 approving review [Learn more](#).

1 approval


Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatic

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also [open this in GitHub Desk](#)

An example workflow


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



Pull request successfully merged and closed

You're all set—the `feature/radiative` branch can be safely deleted.

Delete branch

 main ▾

Commits on Feb 22, 2022

Merge pull request #1 from FeiYao-Edinburgh/feature/radiative ...

Verified



859fd3e



 FeiYao-Edinburgh committed 2 minutes ago



94a2c3a



Third commit.


 FeiYao-Edinburgh committed 16 minutes ago



f602758



Second commit.

 FeiYao-Edinburgh committed 1 hour ago



7243d5d



First commit.

 FeiYao-Edinburgh committed 1 hour ago

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

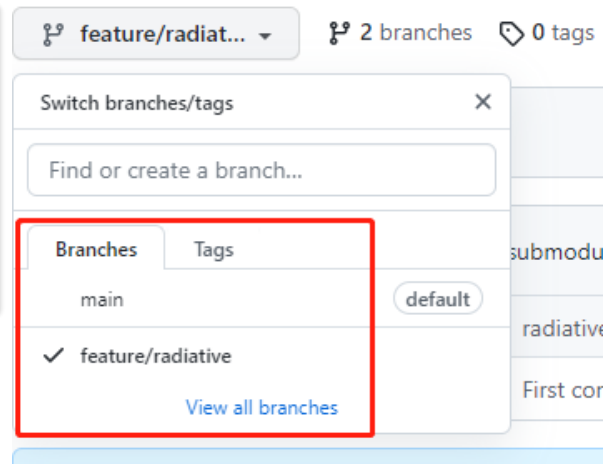
Recap

- 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

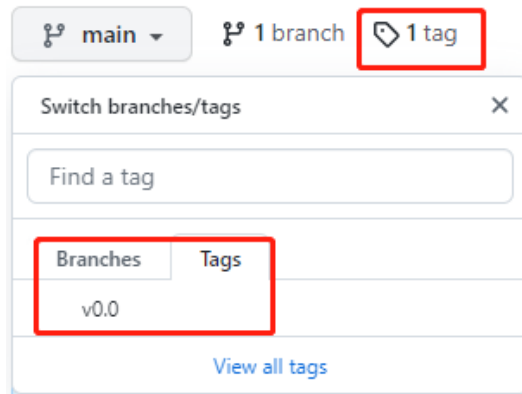
- A safeguard against merge conflicts:

- 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.

Releases

Tags

v0.0

Existing tag

v0.0

Write

Preview

H

B

I

≡

<>

🔗

≡

≡

📄

@

📄

↶

✓ Auto-generate release notes

What's Changed

* Second commit. by @geofyao in <https://github.com/FeiYao-Edinburgh/mesd/pull/1>

* radiative.sh to call all submodules of radiative transfer models. by @geofyao in <https://github.com/FeiYao-Edinburgh/mesd/pull/2>

New Contributors

* @geofyao made their first contribution in <https://github.com/FeiYao-Edinburgh/mesd/pull/1>

Full Changelog: <https://github.com/FeiYao-Edinburgh/mesd/commits/v0.0>

Attach files by dragging & dropping, selecting or pasting them.

↓ Attach binaries by dropping them here or selecting them.

☐ This is a pre-release

We'll point out that this release is identified as non-production ready.

Publish release

Save draft

Releases

Tags

now

geofyao

v0.0

de6855c

Compare

v0.0

Latest

What's Changed

• Second commit. by @geofyao in #1

• radiative.sh to call all submodules of radiative transfer models. by @ge

New Contributors

• @geofyao made their first contribution in #1

Full Changelog: <https://github.com/FeiYao-Edinburgh/mesd/commits/v0.0>

Contributors

geofyao

▼ Assets 2

Source code (zip)

Source code (tar.gz)

.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.

Add some codes to radiative.sh #3

Open FeiYao-Edinburgh opened this issue now · 0 comments



FeiYao-Edinburgh commented now

Add some codes to radiative.sh



FeiYao-Edinburgh added the enhancement label now



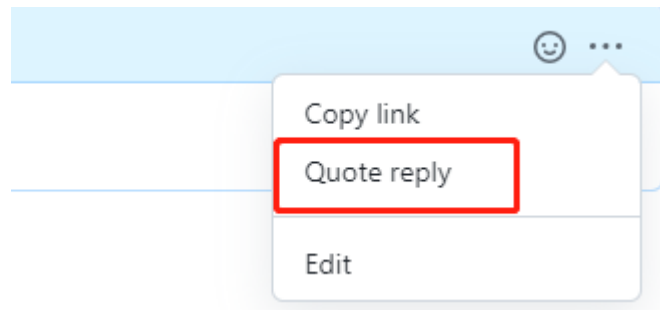
FeiYao-Edinburgh assigned geofyao now

The screenshot shows a GitHub issue page. At the top, the navigation bar includes 'Issues' (highlighted with a red box), 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The issue title is 'Add some codes to radiative.sh', which is also highlighted with a red box. Below the title is a text area for the issue description, containing the text 'Add some codes to [radiative.sh](#)', also highlighted with a red box. To the right of the text area is a sidebar with 'Assignees' (showing 'geofyao' with a red box) and 'Labels' (showing a list of labels with 'enhancement' selected, highlighted with a red box). At the bottom right of the issue description area is a green 'Submit new issue' button. The footer of the page shows the GitHub logo, copyright notice, and various links like 'Terms', 'Privacy', 'Security', etc.

Issues

- Make scientific discussions

- Simple but powerful markdown
- Quote reply



Noting down some thoughts about machine learning

- Not sure if the current model can handle this
- ML in GEE seems to require outlining the area of interest for classification
- Might need to retrain a new model
 - o This will require a new training set
 - o We would need to outline the area of interest
- This would produce a more useful map (e.g., showing the proximity to infrastructure etc)
- Allows further development of the model (e.g., showing the proximity to infrastructure etc)
- Current training data set will look different to the new 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 the areas 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

$$\Delta R_{\text{MBSP}} = \frac{c R_{12} - R_{11}}{R_{11}}, \quad (3)$$

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

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

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

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

GitHub Pages

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

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None ▾

Save

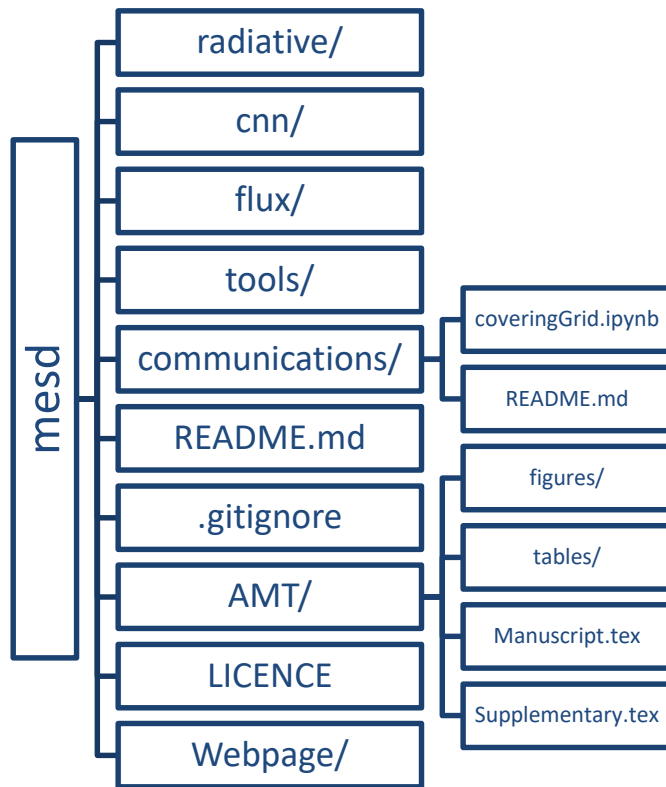


Publish privately to people with read access to this repository

Try risk-free for 30 days using a GitHub Enterprise organization, or [learn more about changing the visibility of your GitHub Pages site.](#) ✕

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>