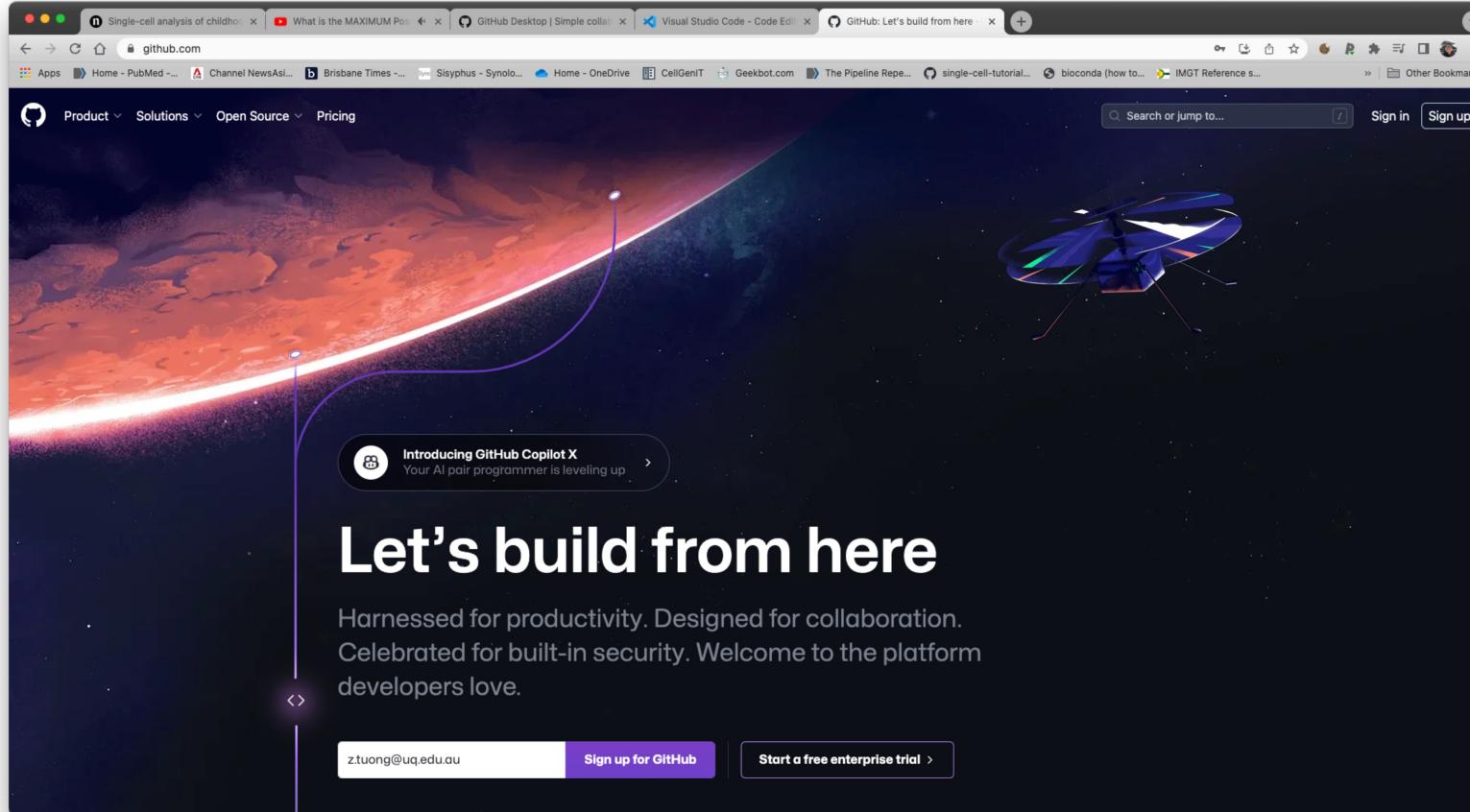


Tuong lab notebook and scripts management onboarding

14/07/2023

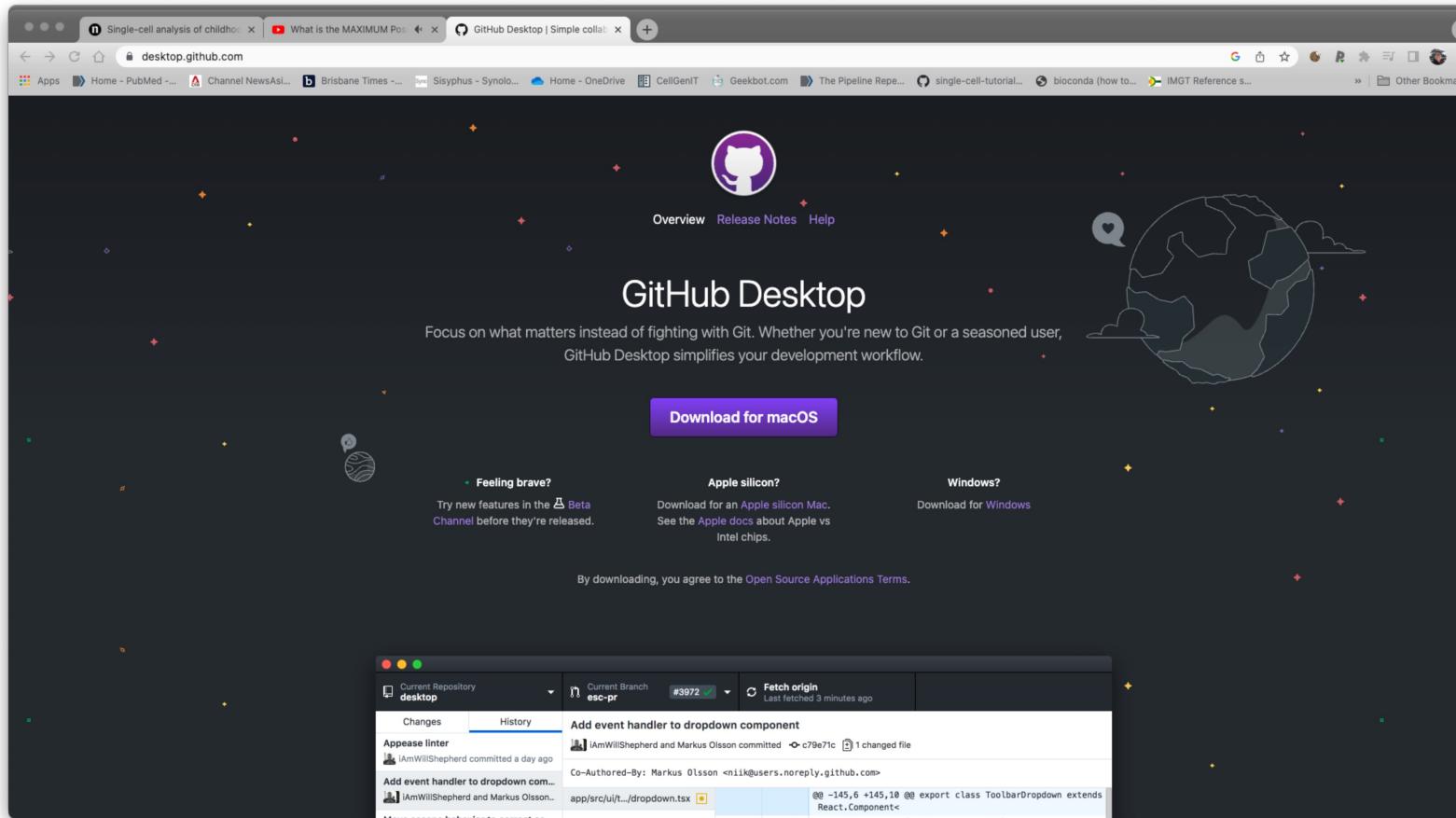
Pre-requisites

Create a github account and let me know your user id



<https://github.com/>

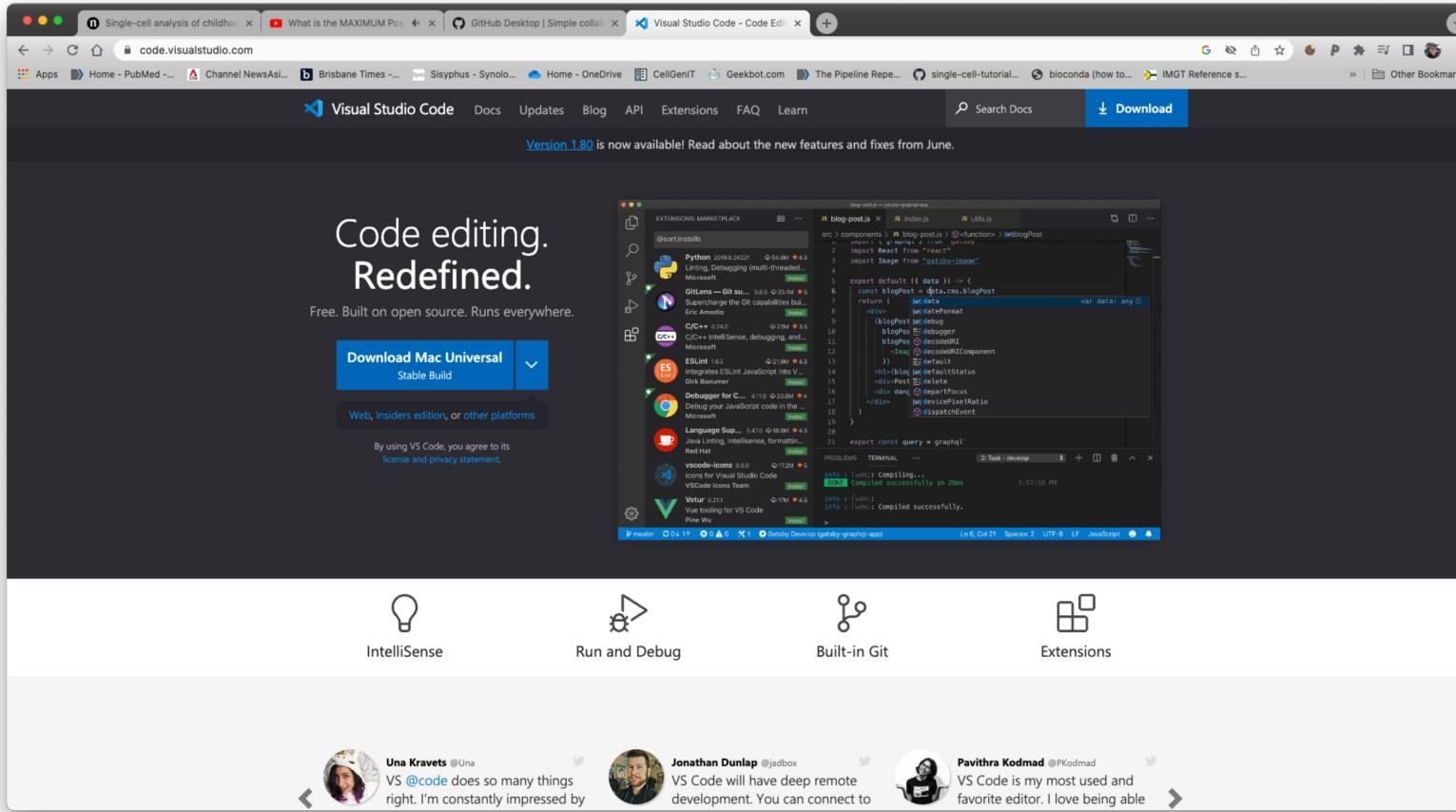
Install GitHub Desktop



<https://desktop.github.com/>

Open it up and sign in with your GitHub credentials when the download is complete

Install VS Code



The screenshot shows the official Visual Studio Code website. At the top, there's a navigation bar with links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A prominent blue button labeled "Download" is visible. Below the navigation, a message states "Version 1.80 is now available! Read about the new features and fixes from June." The main content area features a large heading "Code editing. Redefined." with the subtitle "Free. Built on open source. Runs everywhere." Below this, there are download links for "Download Mac Universal" (Stable Build) and "Web, Insiders edition, or other platforms". A note below the download links says "By using VS Code, you agree to its license and privacy statement." To the right, a large screenshot of the VS Code interface is displayed, showing a code editor with several files open, a sidebar with extensions like Python, ESLint, and GitLens, and a terminal window at the bottom.

Code editing.
Redefined.

Free. Built on open source. Runs everywhere.

[Download Mac Universal](#)
Stable Build

[Web, Insiders edition, or other platforms](#)

By using VS Code, you agree to its license and privacy statement.

IntelliSense Run and Debug Built-in Git Extensions

Una Kravets @Una
VS @code does so many things right. I'm constantly impressed by

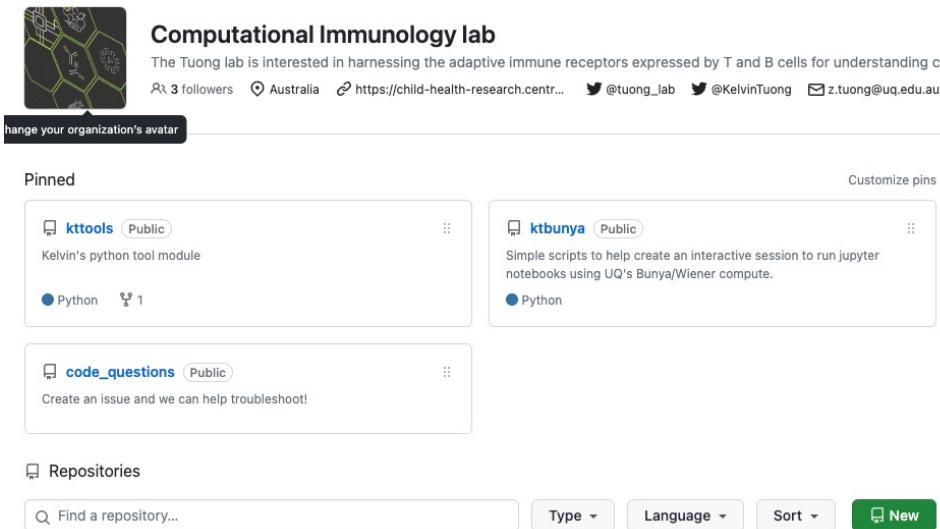
Jonathan Dunlap @jadbox
VS Code will have deep remote development. You can connect to

Pavithra Kodmad @PKodmad
VS Code is my most used and favorite editor. I love being able

<https://code.visualstudio.com/>

Create a new repository

Go to <https://github.com/tuonglab> and click the **New** button



Computational Immunology lab

The Tuong lab is interested in harnessing the adaptive immune receptors expressed by T and B cells for understanding child

3 followers Australia https://child-health-research.centr... @tuong_lab @KelvinTuong z.tuong@uq.edu.au

Pinned

- kttools Public Kelvin's python tool module Python 1
- ktbunya Public Simple scripts to help create an interactive session to run jupyter notebooks using UQ's Bunya/Wiener compute. Python
- code_questions Public Create an issue and we can help troubleshoot!

Repositories

Find a repository... Type Language Sort New



Call your repository whatever you want
click **Create Repository** when done

Create a new repository

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

Required fields are marked with an asterisk (*).

Repository template



Start your repository with a template repository's contents.

Include all branches

Copy all branches from tuonglab/tuong-group-onboarding and not just the default branch.

Owner *



Repository name *

Kelvin project 1

⚠ Your new repository will be created as Kelvin-project-1.

Great repository names are short and memorable. Need inspiration? How about **turbo-guacamole** ?

Description (optional)



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



You choose who can see and commit to this repository.

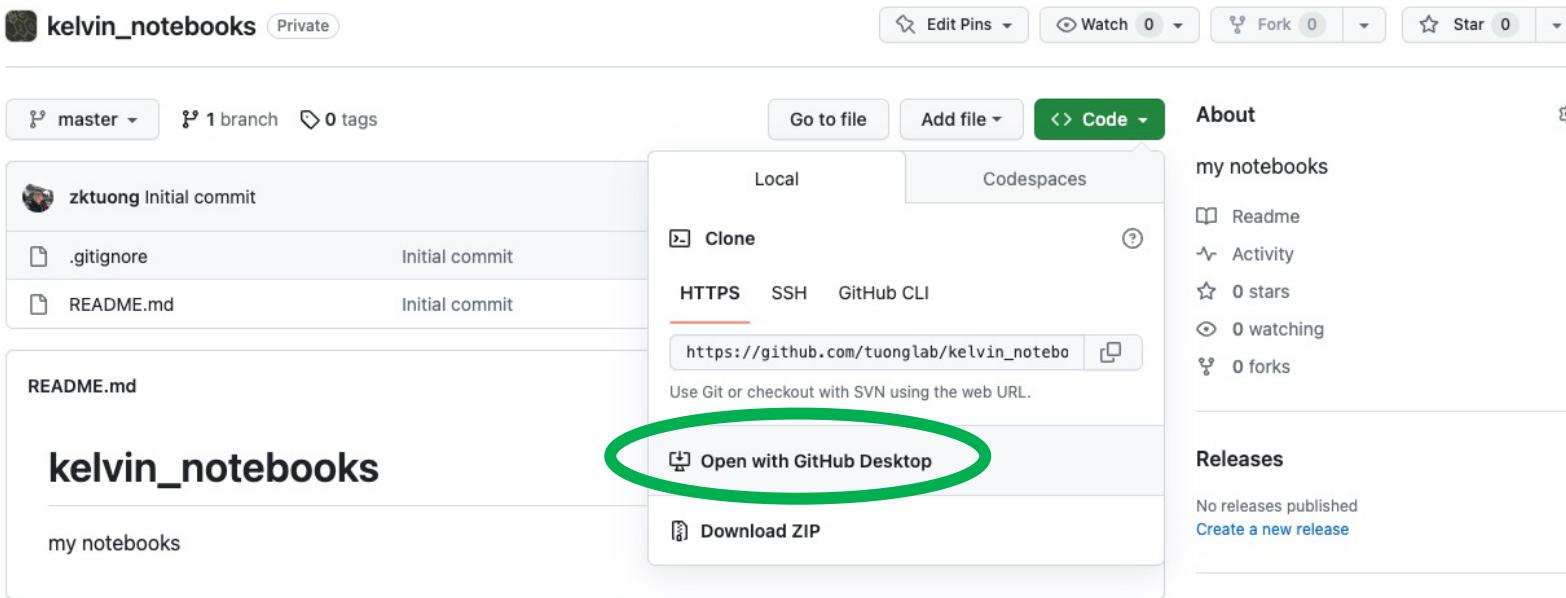
ⓘ You are creating a public repository in the tuonglab organization.

Create repository

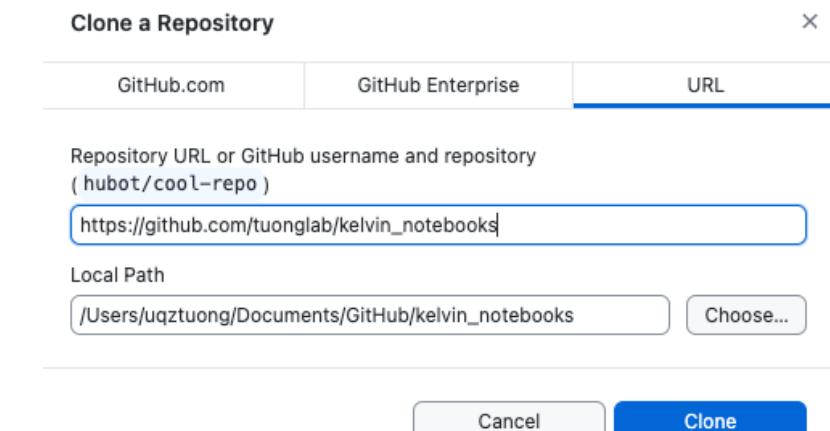
I would suggest to make it public if it's not sensitive data/scripts

Create a repository - continue

Click **Code** and select open with GitHub Desktop



The screenshot shows a GitHub repository page for 'kelvin_notebooks'. The 'Code' button in the top right corner is highlighted. A dropdown menu appears with options: 'Local' (selected), 'Codespaces', 'Clone', 'HTTPS', 'SSH', and 'GitHub CLI'. Below these options is a URL: https://github.com/tuonglab/kelvin_notebooks. At the bottom of the dropdown, there are two buttons: 'Open with GitHub Desktop' (circled in green) and 'Download ZIP'.

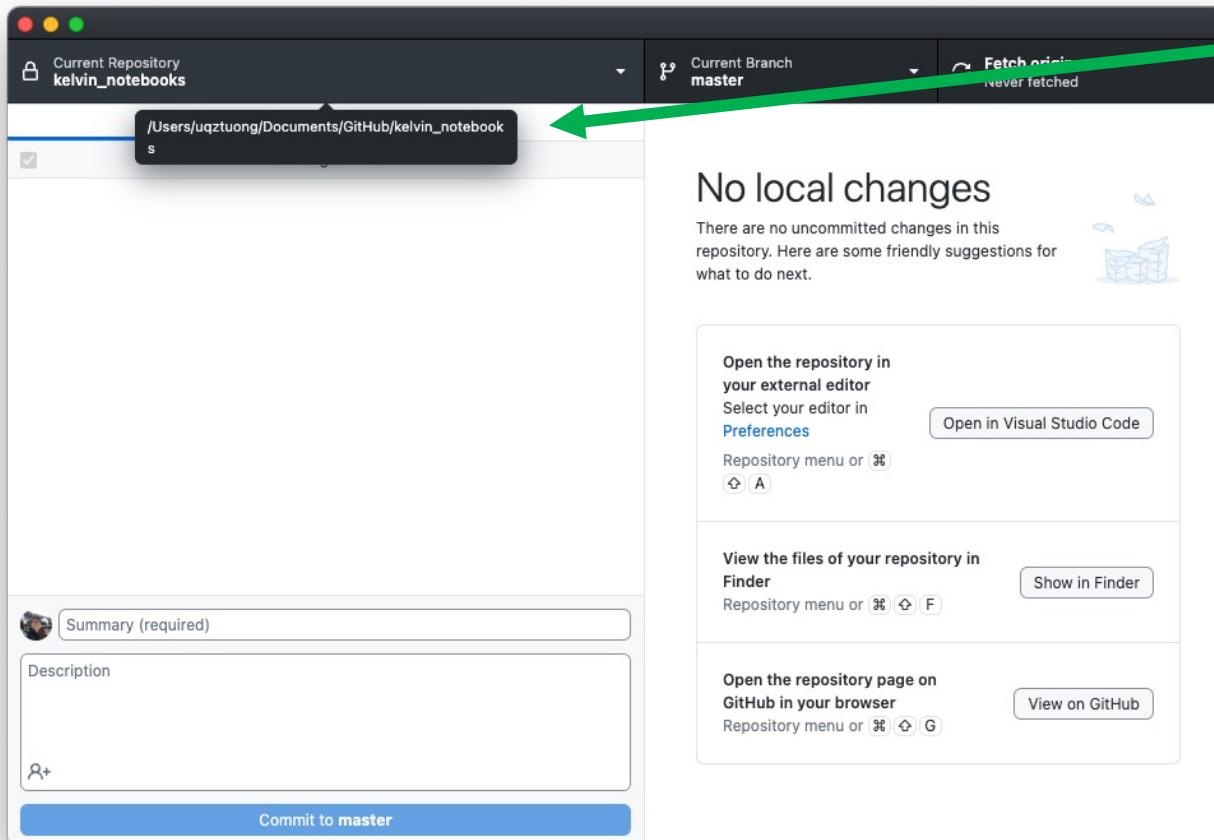


The screenshot shows the 'Clone a Repository' dialog box. It has tabs for 'GitHub.com' (selected), 'GitHub Enterprise', and 'URL'. The 'URL' tab is active, showing the repository URL: https://github.com/tuonglab/kelvin_notebooks. Below the URL is a 'Local Path' input field containing the path: /Users/uqztuong/Documents/GitHub/kelvin_notebooks. At the bottom are 'Cancel' and 'Clone' buttons.

Click **Clone**

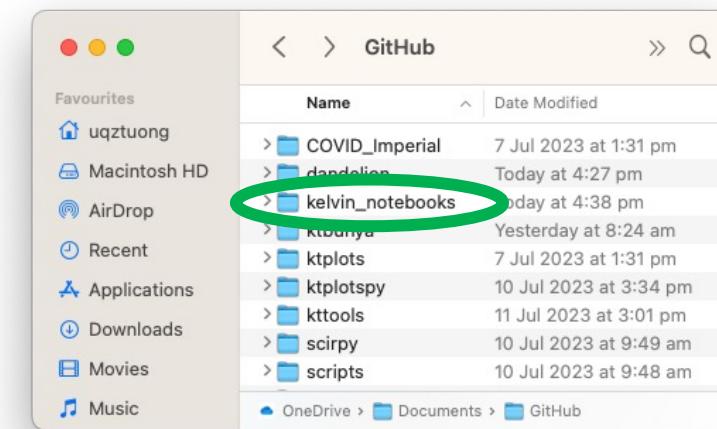
Create a repository - continue

This is what it will look like.



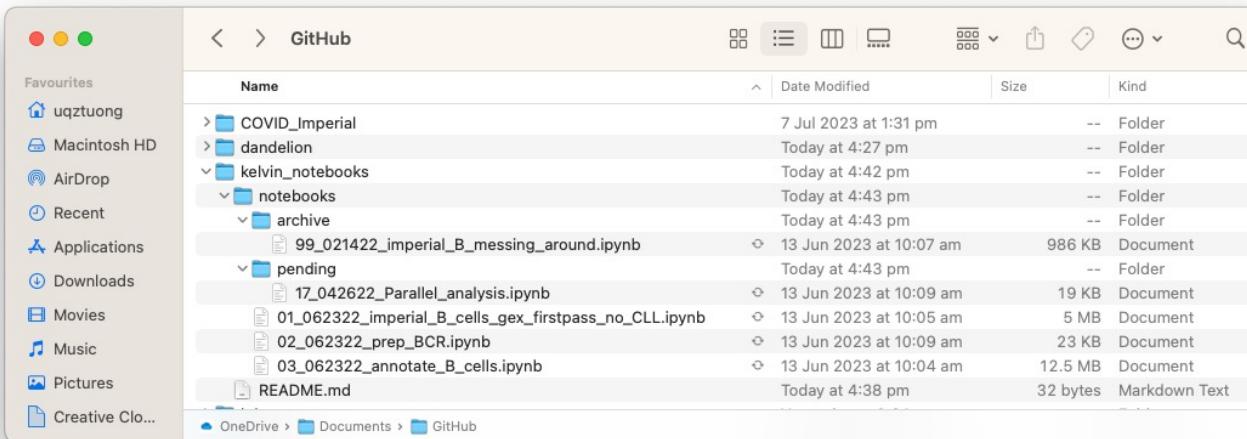
Now you can make changes to this folder

You can find the path by hovering on the left

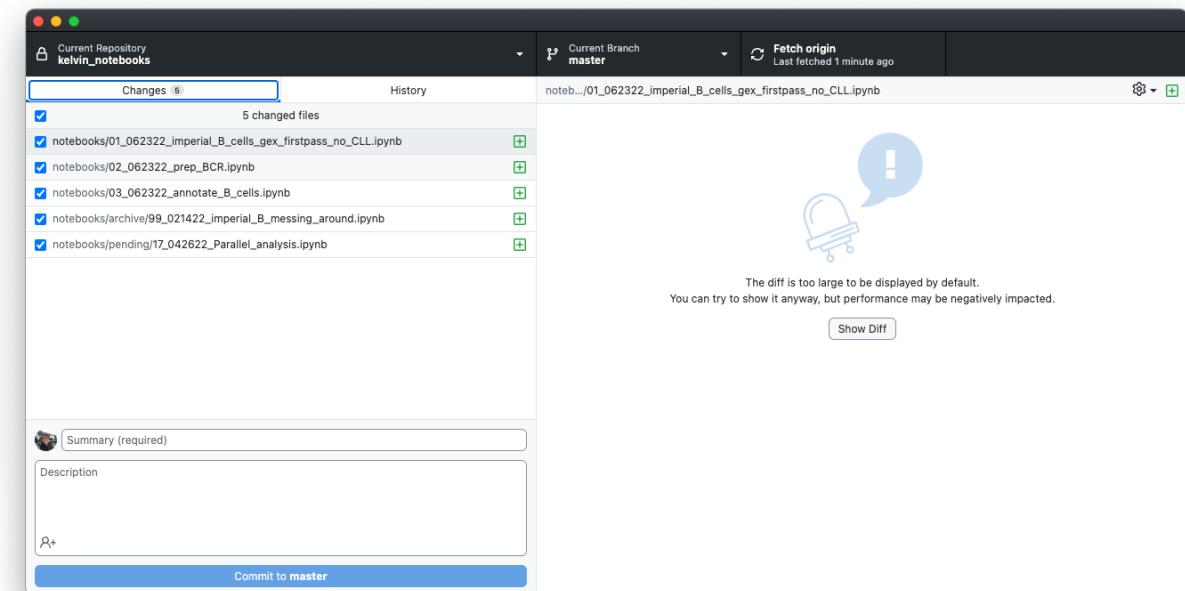


Let's do some work and put the analysis script into the folder

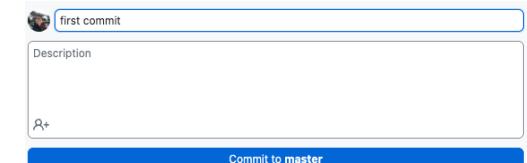
If you check your GitHub desktop app,



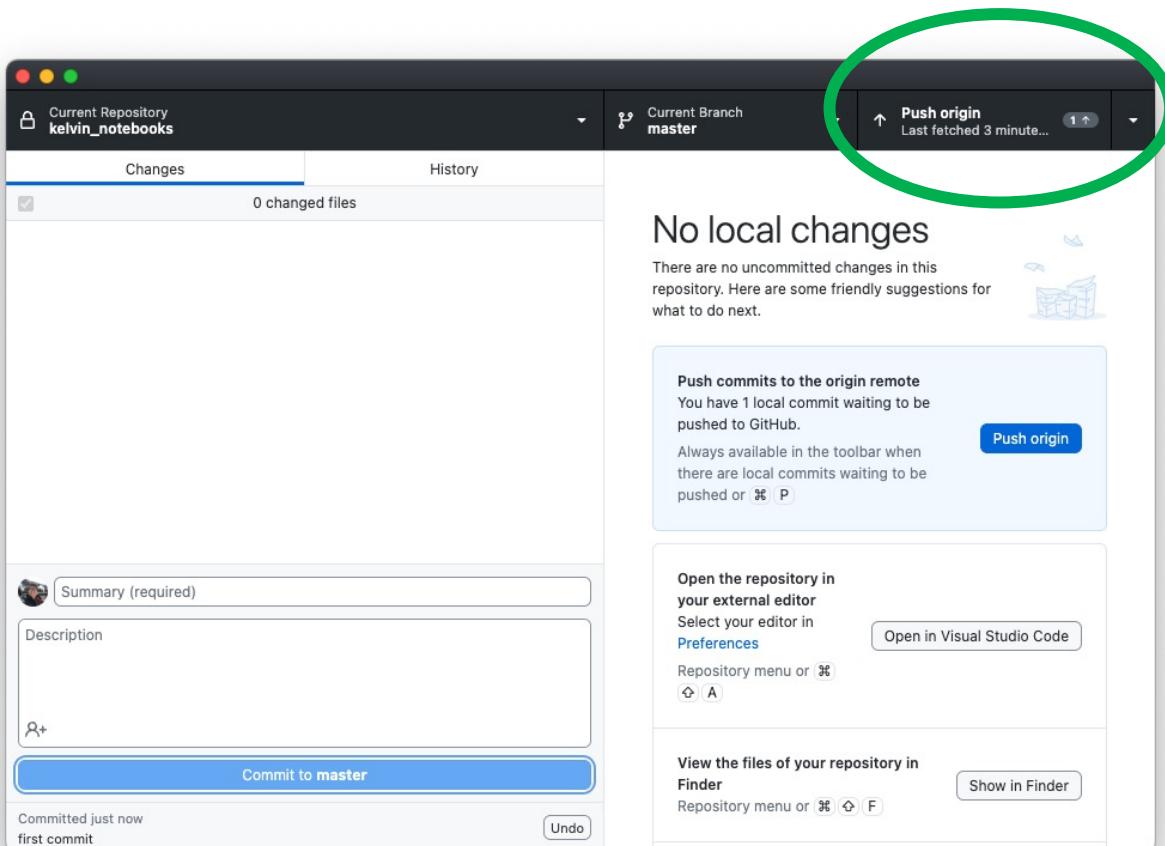
Say I did a bunch of analyses in jupyter notebooks and created folders within the `kelvin_notebooks` folder, and then these folders hold my scripts and notebooks



you will see that GitHub Desktop has tracked that you've modified/saved/created some files! Now click the **commit** button after giving a commit message!



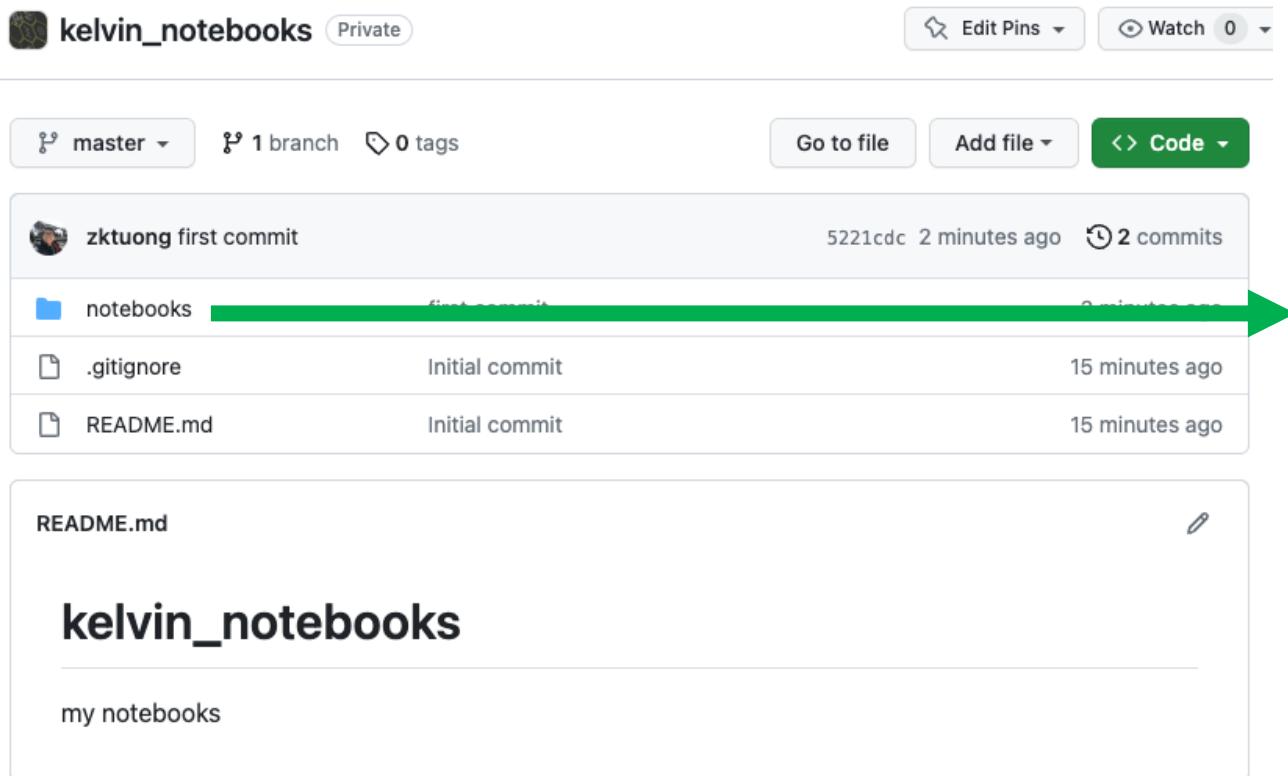
Commit -> Push -> Fetch -> Commit -> Push -> Fetch ad nauseum

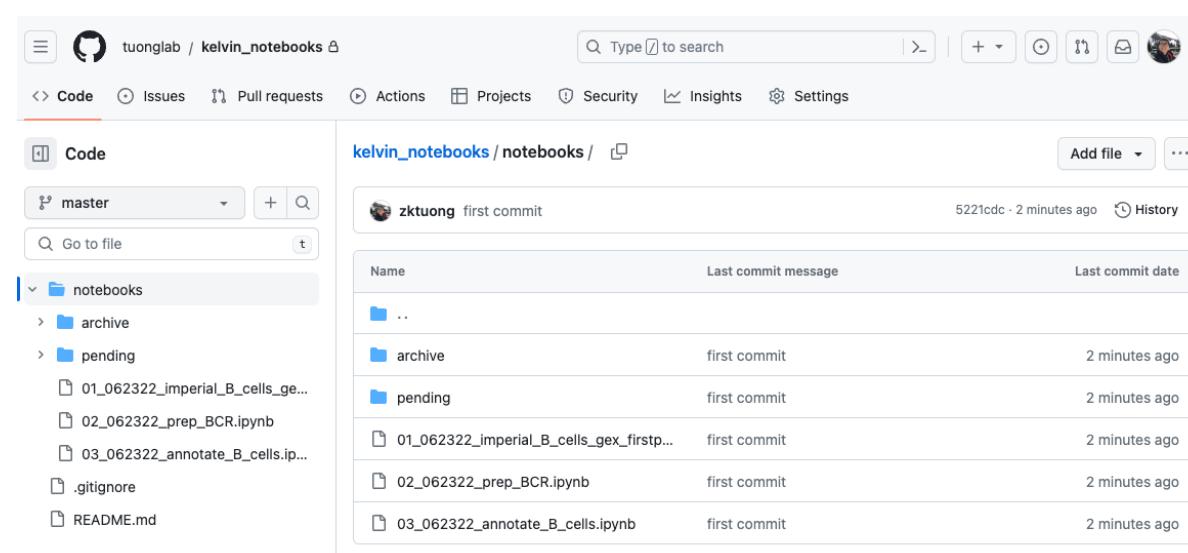


Once you are ready to push this to github, you can click on the Push button and that will send it online

Before this step, you can always continue to make changes, revert changes etc.

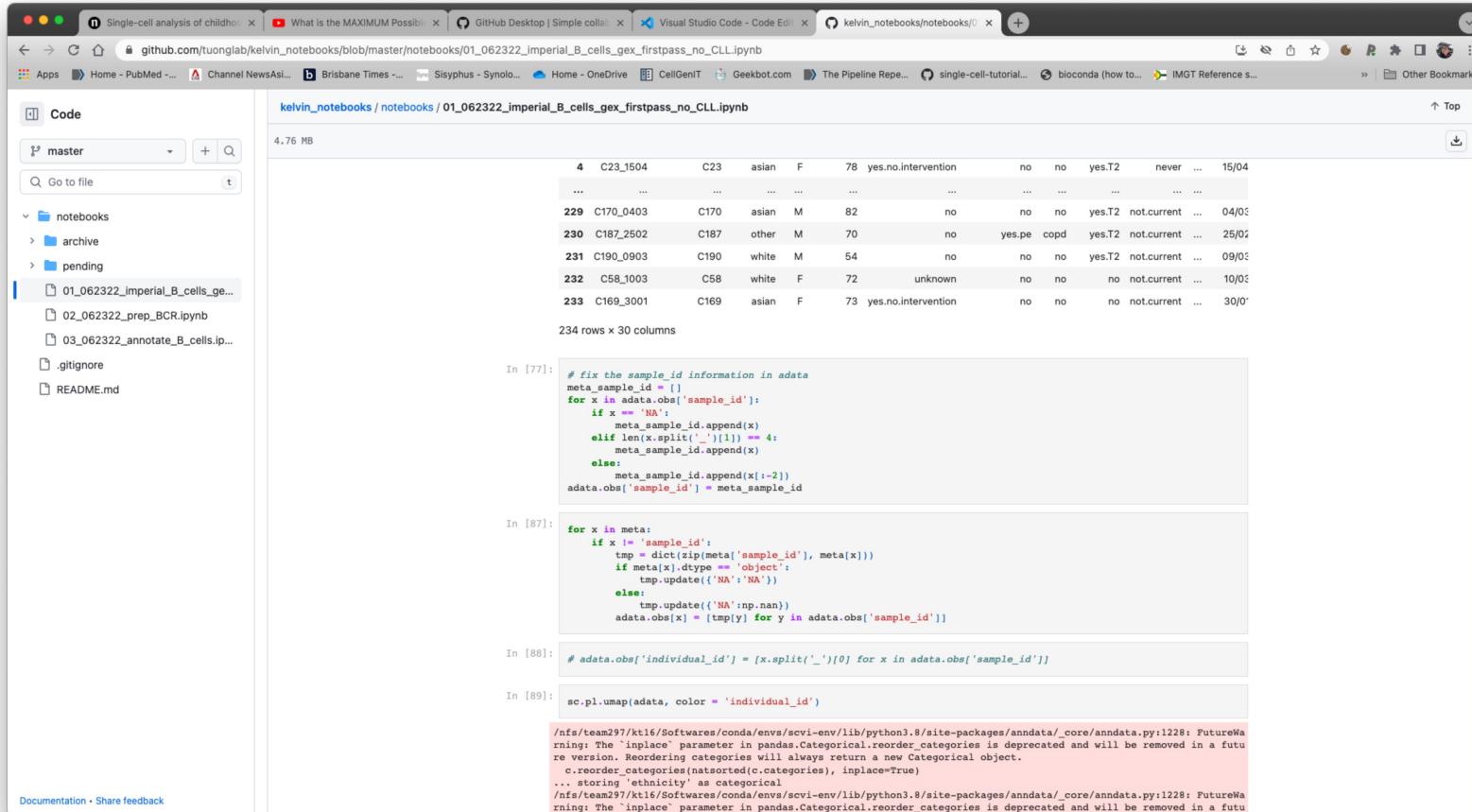
Check your github repo online

A screenshot of a GitHub repository page for 'kelvin_notebooks'. The repository is private. It shows a single commit from user 'zktuong' titled 'first commit' made 2 minutes ago. The commit message is '5221cdc'. Below the commit, there are two files: 'notebooks' and '.gitignore'. A green arrow points from the commit area to the right side of the image, indicating a comparison with the live version.

A screenshot of the same GitHub repository 'kelvin_notebooks' after changes have been pushed. The commit from 'zktuong' is now listed under the 'Code' tab, showing it has 2 commits. The commit message is 'first commit'. The commit was made 2 minutes ago. The commit history shows three files: 'archive', 'pending', and 'README.md'. The 'archive' file contains three subfiles: '01_062322_imperial_B_cells_ge...', '02_062322_prep_BCR.ipynb', and '03_062322_annotate_B_cells.ip...'. The 'pending' file contains three subfiles: '01_062322_imperial_B_cells_gex_fir...', '02_062322_prep_BCR.ipynb', and '03_062322_annotate_B_cells.ip...'. The 'README.md' file is also present.

Your changes are now live

If I were to then take a look at your repo, I can now look at it and give you my feedback live!



The screenshot shows a Jupyter Notebook interface within a web browser. The notebook is titled 'kelvin_notebooks / notebooks / 01_062322_imperial_B_cells_gex_firstpass_no_CLL.ipynb'. The code in cell [77] is:

```
# fix the sample_id information in adata
meta_sample_id = []
for x in adata.obs['sample_id']:
    if x == 'NA':
        meta_sample_id.append(x)
    elif len(x.split('_'))[1] == 4:
        meta_sample_id.append(x)
    else:
        meta_sample_id.append(x[:-2])
adata.obs['sample_id'] = meta_sample_id
```

The code in cell [87] is:

```
for x in meta:
    if x != 'sample_id':
        tmp = dict(zip(meta['sample_id'], meta[x]))
        if meta[x].dtype == 'object':
            tmp.update({'NA': 'NA'})
        else:
            tmp.update({'NA': np.nan})
        adata.obs[x] = [tmp[y] for y in adata.obs['sample_id']]
```

The code in cell [88] is:

```
# adata.obs['individual_id'] = [x.split('_')[0] for x in adata.obs['sample_id']]
```

The code in cell [89] is:

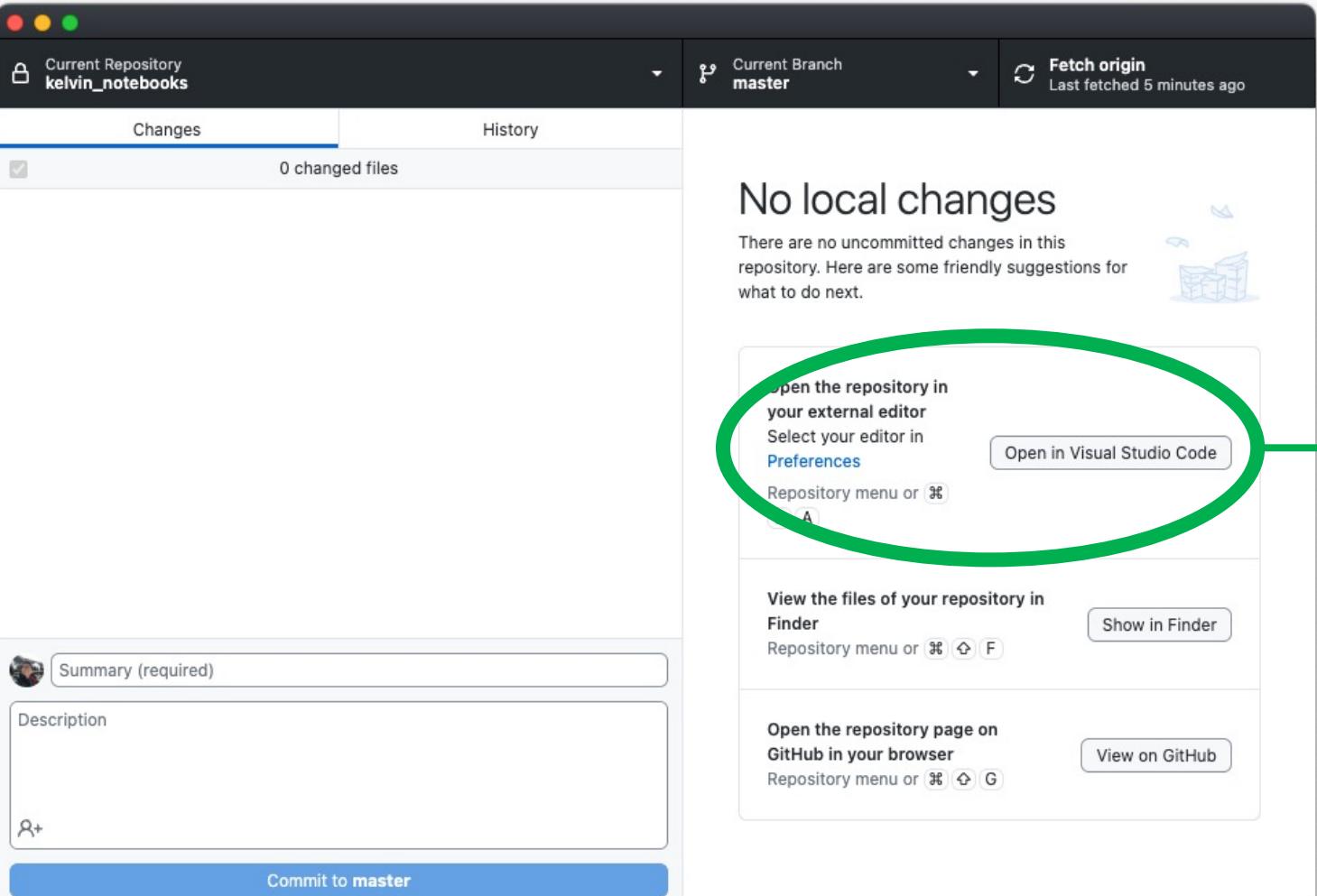
```
sc.pl.umap(adata, color = 'individual_id')
```

Output from cell [89] shows a warning message:

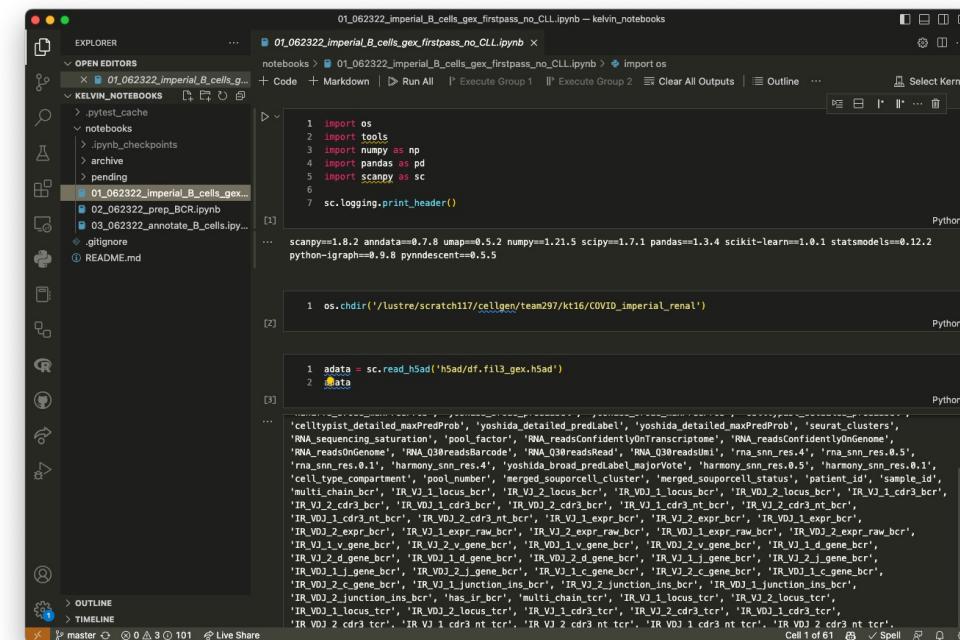
```
/nfs/team297/kt16/Softwares/conda/envs/scvi-env/lib/python3.8/site-packages/anndata/_core/anndata.py:1228: FutureWarning: The 'inplace' parameter in pandas.Categorical.reorder_categories is deprecated and will be removed in a future version. Reordering categories will always return a new Categorical object.
c.reorder_categories(natsorted(c.categories), inplace=True)
... storing 'ethnicity' as categorical
/nfs/team297/kt16/Softwares/conda/envs/scvi-env/lib/python3.8/site-packages/anndata/_core/anndata.py:1228: FutureWarning: The 'inplace' parameter in pandas.Categorical.reorder_categories is deprecated and will be removed in a future
```

I will be able to then interact with you and your code using this method, and also additional ways to help with your progress.

Before we end, why did I tell you to download VS code?



The screenshot shows the GitHub desktop application interface. At the top, it displays the current repository as "kelvin_notebooks" and the current branch as "master". Below this, there are tabs for "Changes" and "History", with "Changes" currently selected. It shows "0 changed files". The main area displays a message: "No local changes. There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next." It includes three options: "Open the repository in your external editor" (with a green oval around it), "View the files of your repository in Finder", and "Open the repository page on GitHub in your browser". At the bottom, there is a "Commit to master" button.



The screenshot shows a Visual Studio Code window with an open Python notebook titled "01_062322_imperial_B_cells_gex_firstrpass_no_CLL.ipynb" from the "kelvin_notebooks" repository. The notebook contains several code cells. The first cell contains imports like os, tools, numpy, pandas, and scipy. The second cell contains a command to change the directory to "/lustre/scratch117/cellogen/team297/kt16/COVID_imperial_renal". The third cell contains code to read an H5AD file. The interface shows the Explorer sidebar with other files in the repository, and the bottom right shows the Python kernel status.

Because you can actually do
(almost) everything just from VS code!

Final words

- Basically, the github desktop location will be where you would store your scripts and not your data
- Manage your data in other locations as GitHub has a size limit for repo, hence only scripts and no data. You can
- The good thing about this is that actually anything you do on GitHub can be linked to our Slack channel and the Trello board so it will be very easy to communicate our progress.
- When you create a repo using the '[tuong-group-onboarding](#)' as a template, then the figures and data folders are ignored – i.e. they won't get pushed to github so you can store as many as you want here.

Create a new repository

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

Required fields are marked with an asterisk (*).

Repository template



Start your repository with a template repository's contents.

Include all branches

Copy all branches from tuonglab/tuong-group-onboarding and not just the default branch.

Owner *



Repository name *

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

Description (optional)

Public

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

