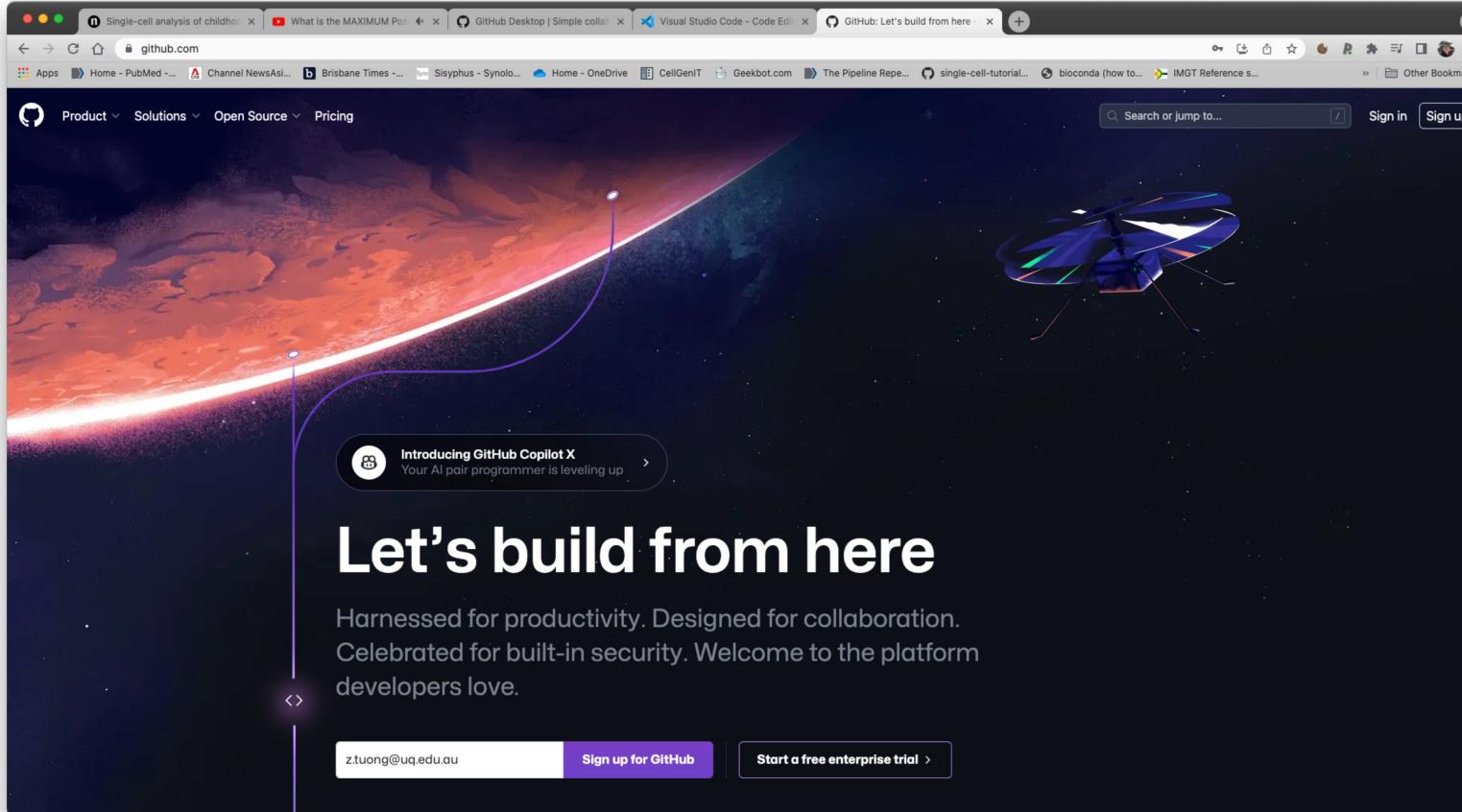


Tuong lab notebook and scripts management onboarding

14/07/2023

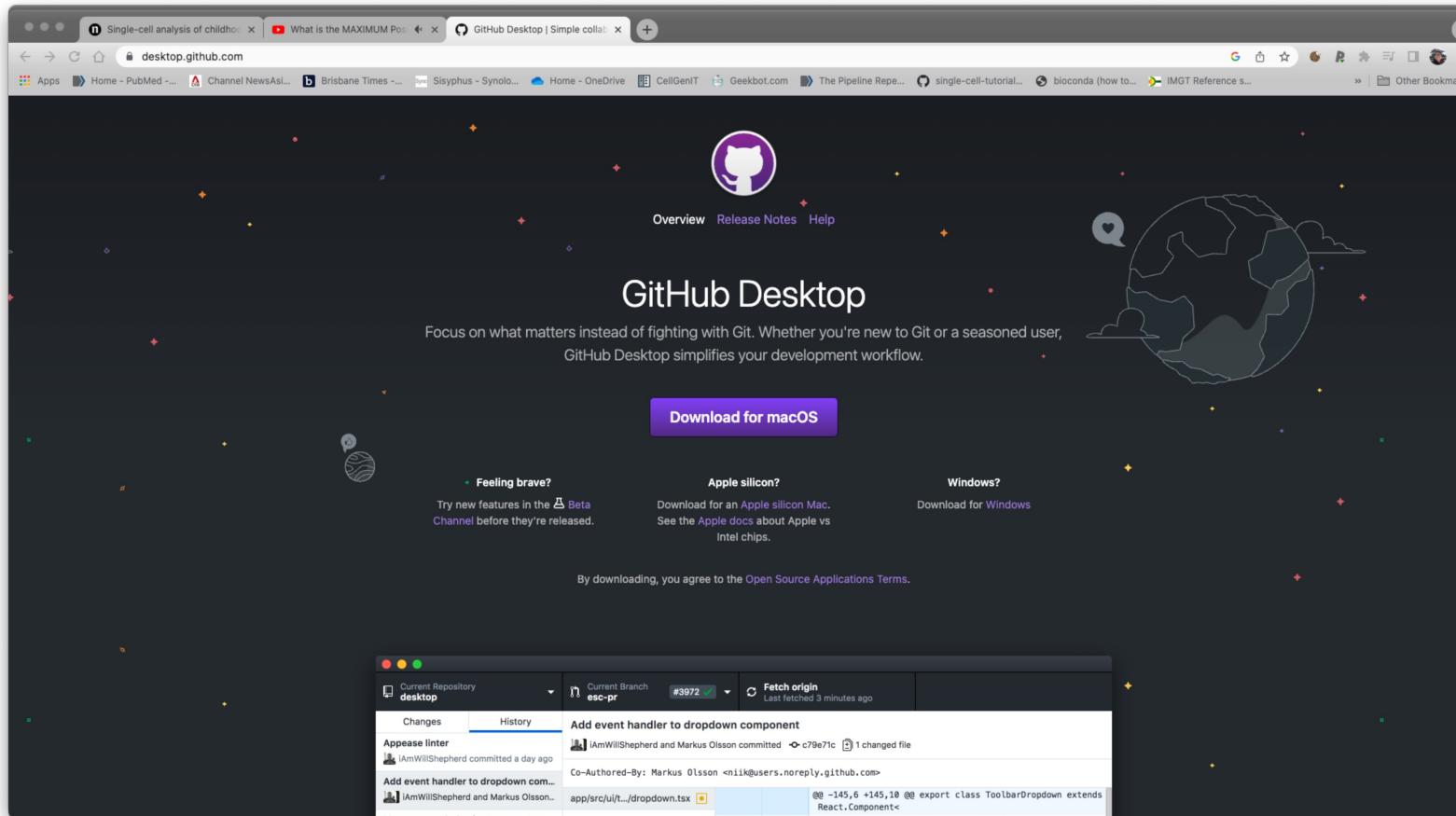
1) Pre-requisites

Create a github account and let me know your user id



<https://github.com/>

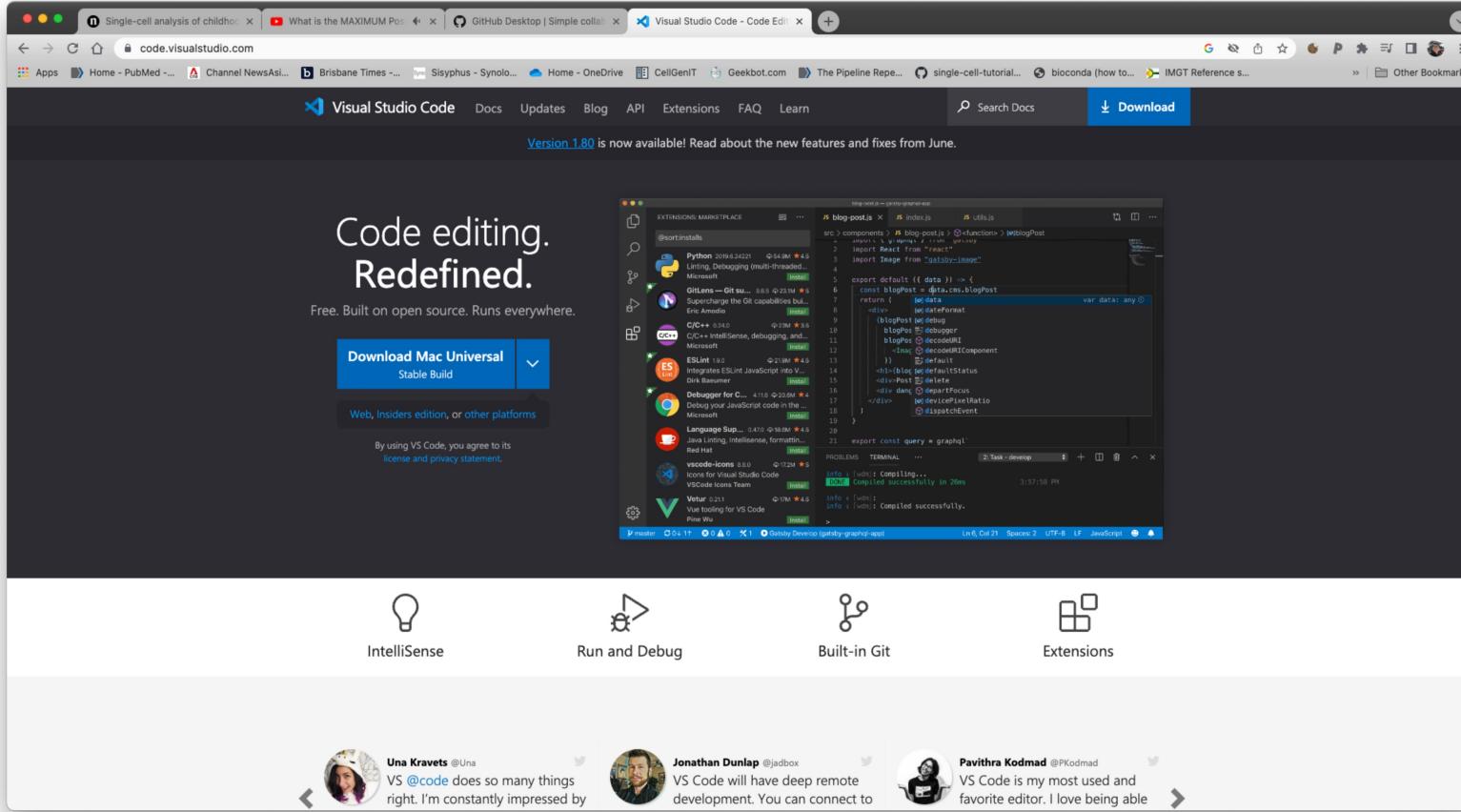
Install GitHub Desktop



<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

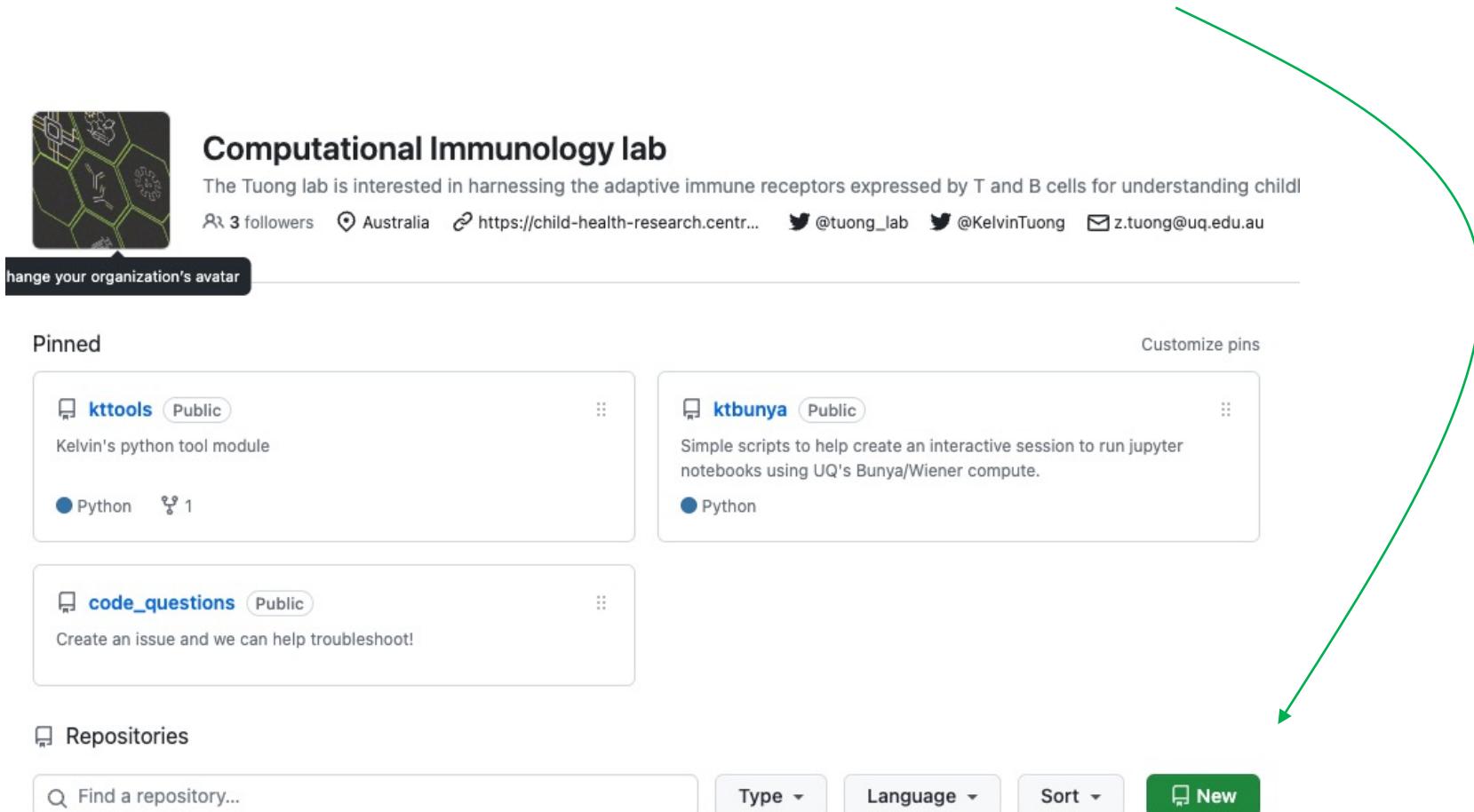
Jonathan Dunlap @jadbox
VS Code will have deep remote development. You can connect to

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



The screenshot shows the GitHub profile page for 'Computational Immunology lab'. The page includes the organization's avatar (a stylized cell diagram), name, description, follower count (3), location (Australia), website, and contact information. A 'Change your organization's avatar' button is visible. The 'Pinned' section displays three repositories: 'kttools' (Python tool module), 'ktbunya' (scripts for running Jupyter notebooks on Bunya/Wiener compute), and 'code_questions' (a troubleshooting resource). A 'Customize pins' link is located above the pinned repositories. At the bottom, there's a 'Repositories' section with a search bar and filters for Type, Language, Sort, and a prominent green '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

Change your organization's avatar

Pinned

Customize pins

kttools Public Kelvin's python tool module

ktbunya Public Simple scripts to help create an interactive session to run jupyter notebooks using UQ's Bunya/Wiener compute.

code_questions Public Create an issue and we can help troubleshoot!

Repositories

Find a repository... Type Language Sort New

Create repository with template

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 *

 /

⚠ 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)

 **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.

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

Create repository

Make sure to select the tuong-group-onboarding template

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 *

 /

⚠ 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)

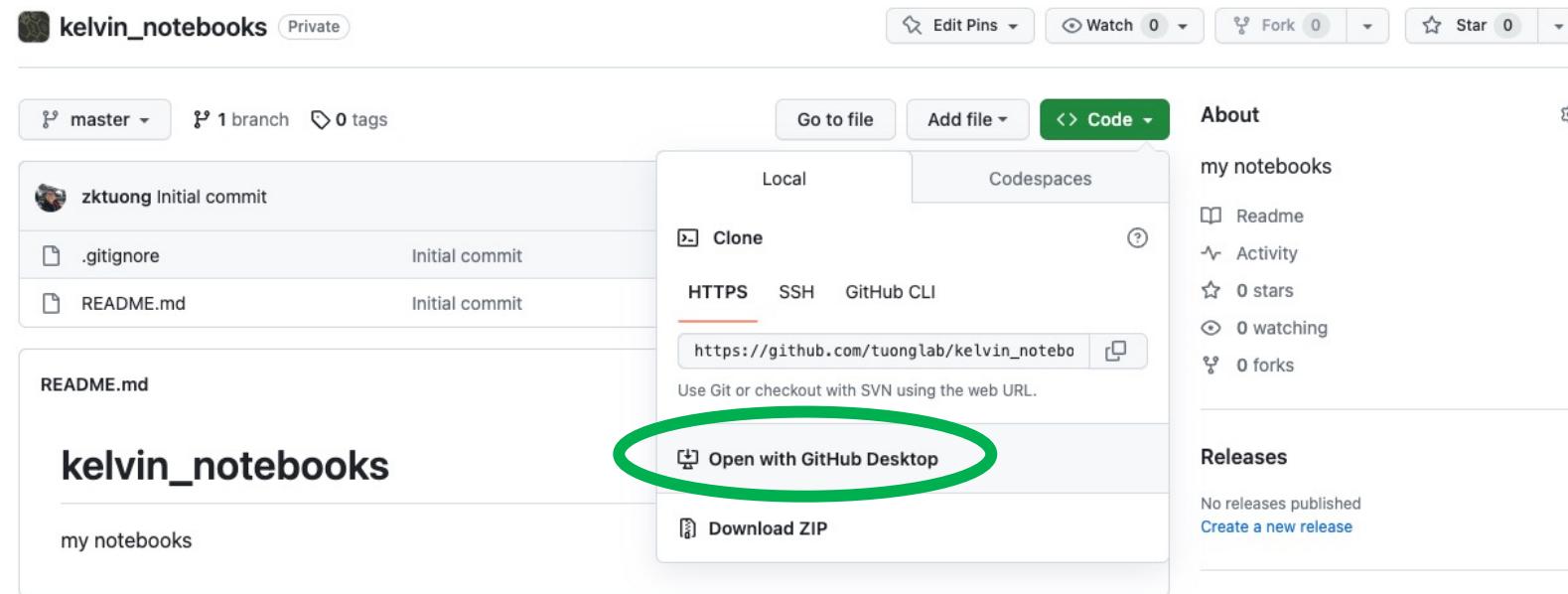
Call your repository whatever you want
click **Create Repository** when done

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

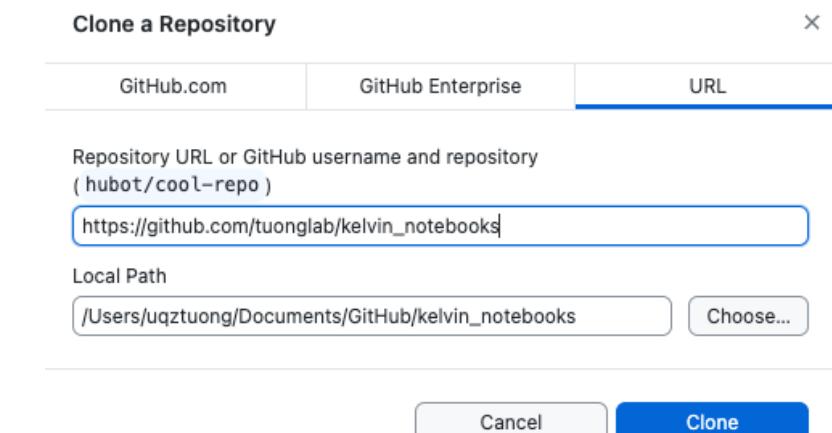
The template will
ensure you have
some basic folders

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 is the option 'Open with GitHub Desktop', which is circled in green.

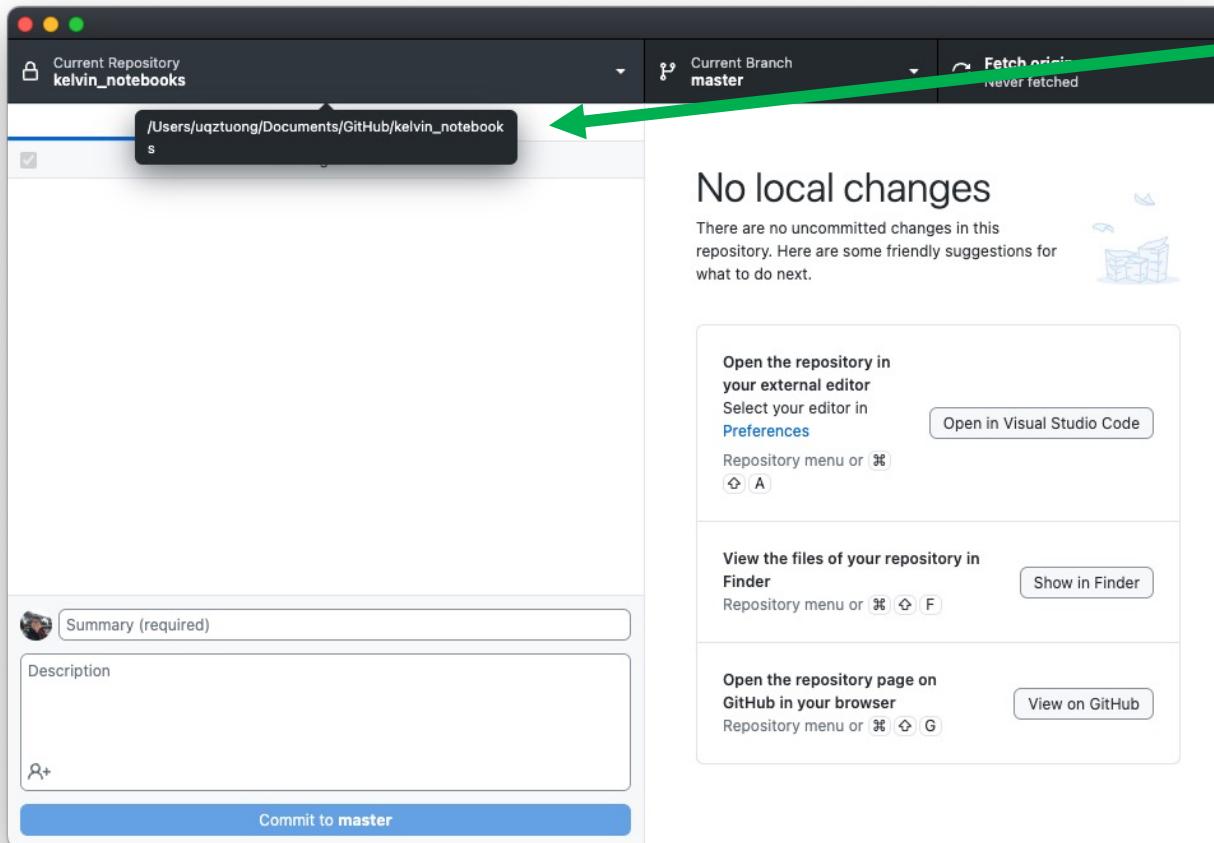


The screenshot shows the 'Clone a Repository' dialog box. It has tabs for 'GitHub.com' and 'GitHub Enterprise', with 'URL' selected. The 'Repository URL or GitHub username and repository' field contains 'hubot/cool-repo'. The 'Local Path' field contains '/Users/uqztuong/Documents/GitHub/kelvin_notebooks'. At the bottom are 'Cancel' and 'Clone' buttons.

Click **Clone**

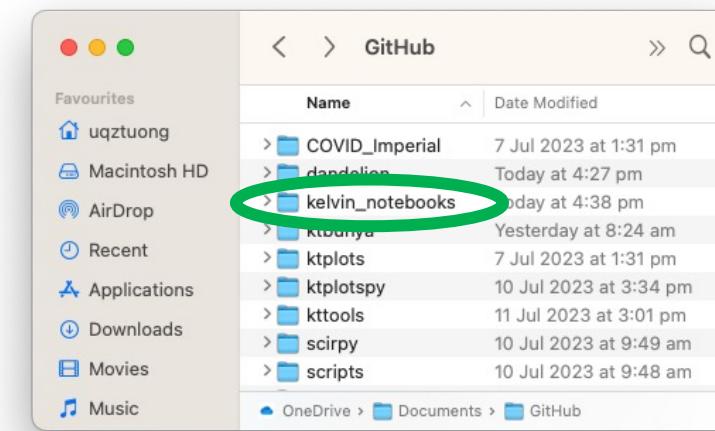
Create a repository - continue

This is what it will look like.



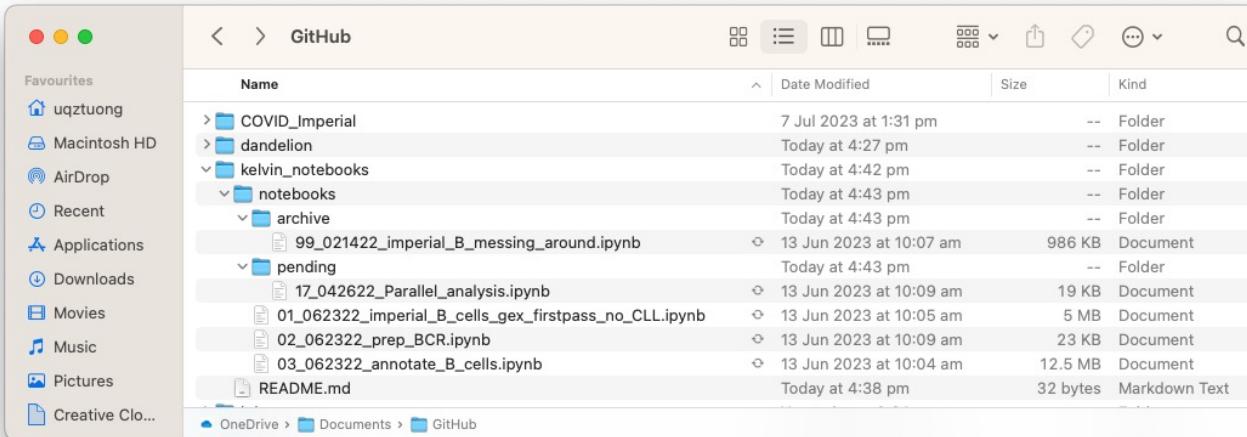
Now you can make changes to this folder

You can find the path by hovering on the left



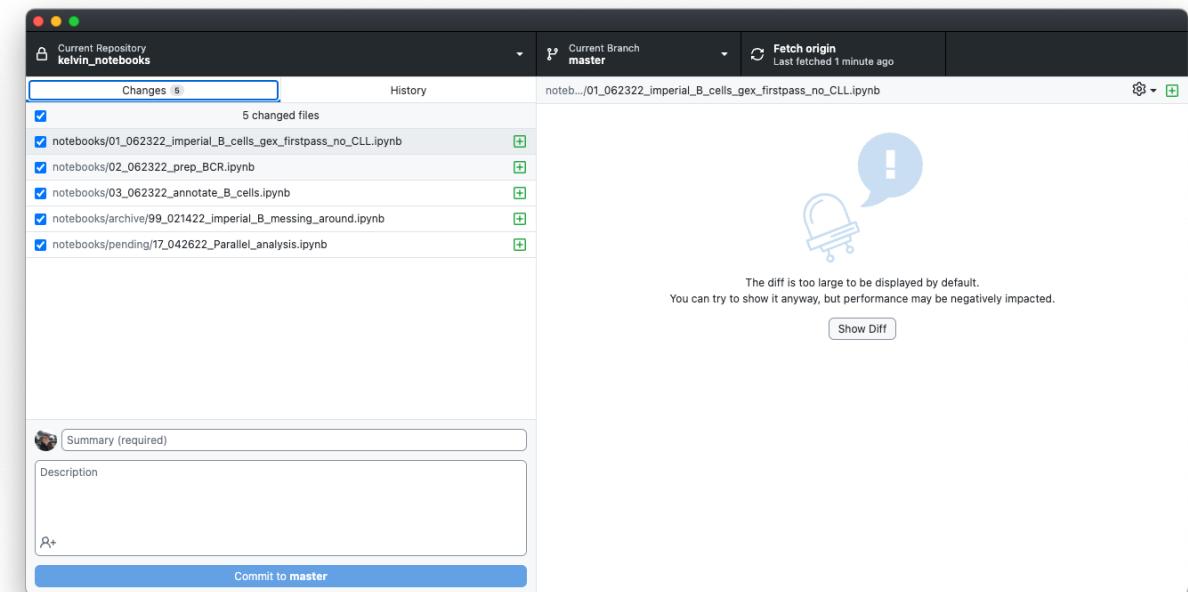
Let's do some work and put the analysis script into the folder

If you check your GitHub desktop app,

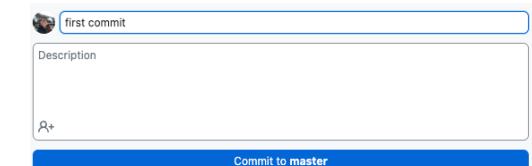


Say I did a bunch of analyses in jupyter notebooks and stored my notebooks, scripts, data and other files in the folders within the kelvin_notebooks folder

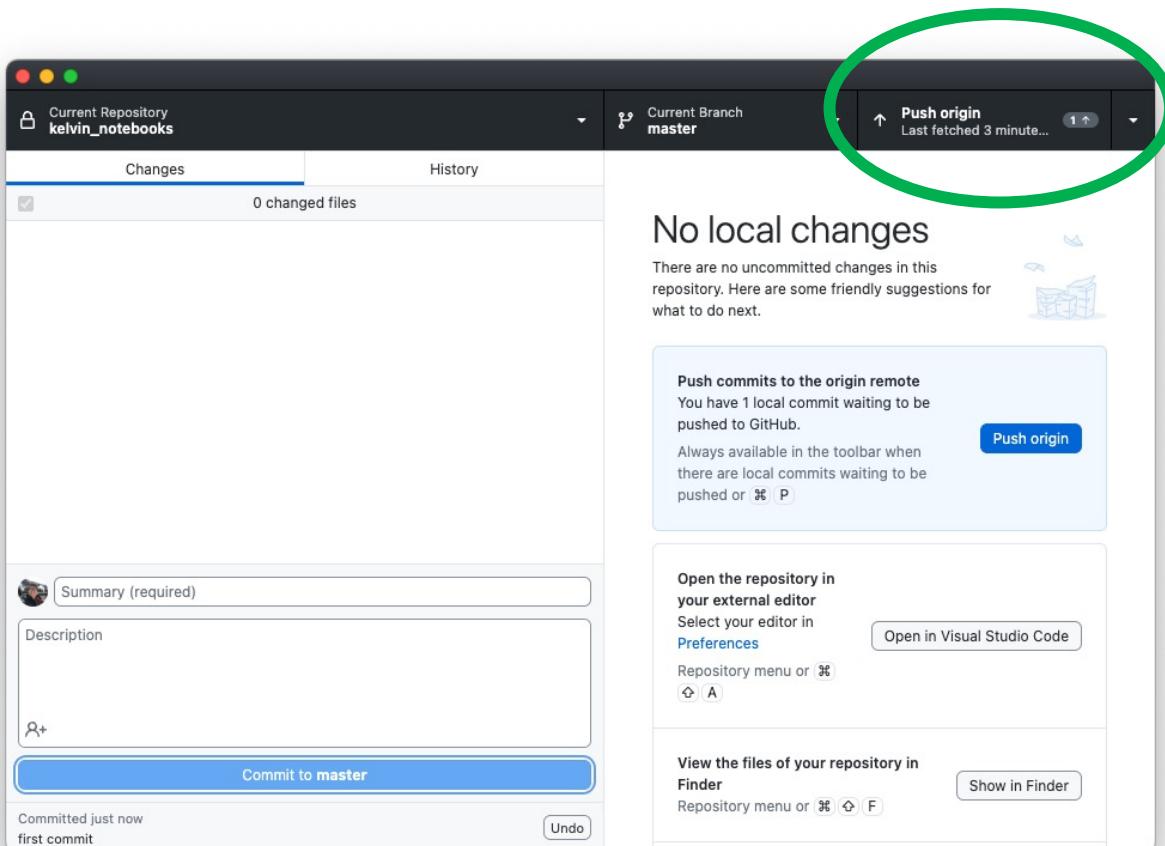
You will notice that the files in the data, write and figures folders are not tracked – this is intentional



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!



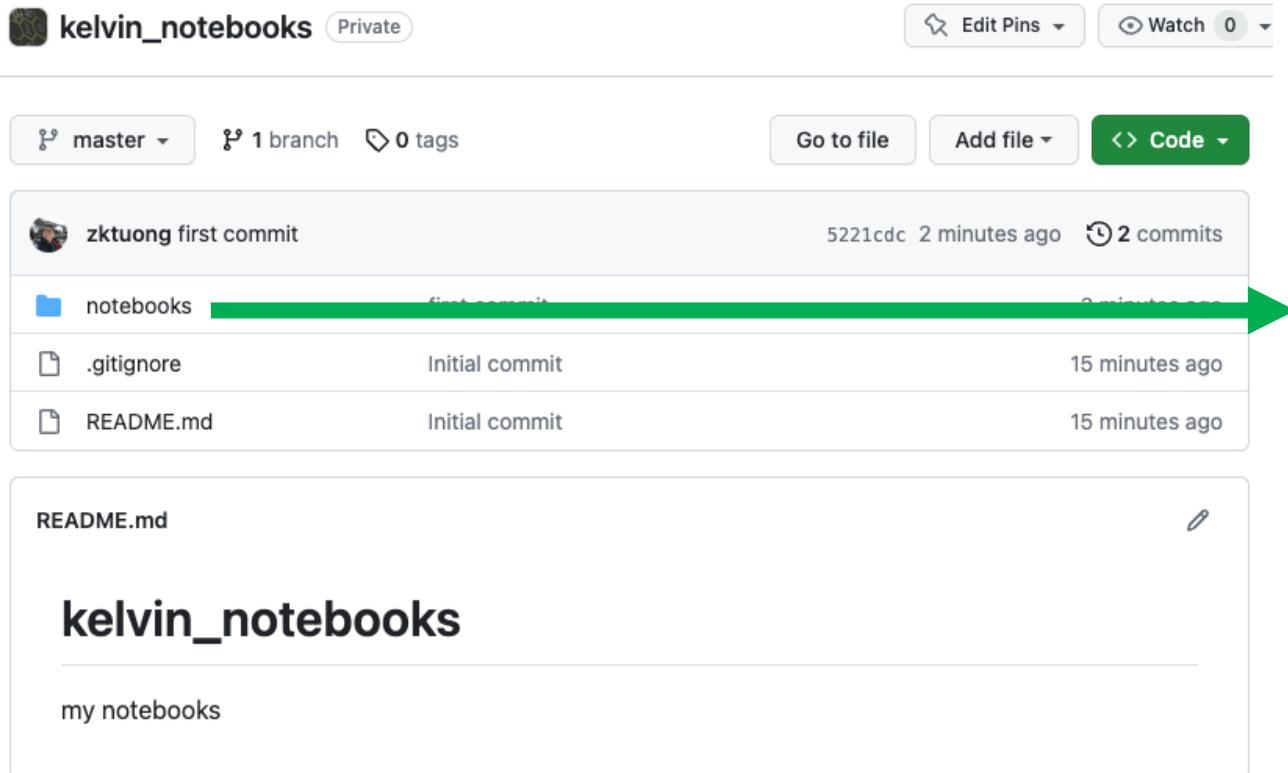
Commit -> Push -> Fetch -> Commit -> Push -> Fetch ad nauseum



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.

kelvin_notebooks Private

master 1 branch 0 tags

Go to file Add file ▾ <> Code ▾

zktuong first commit 5221cdc 2 minutes ago 2 commits

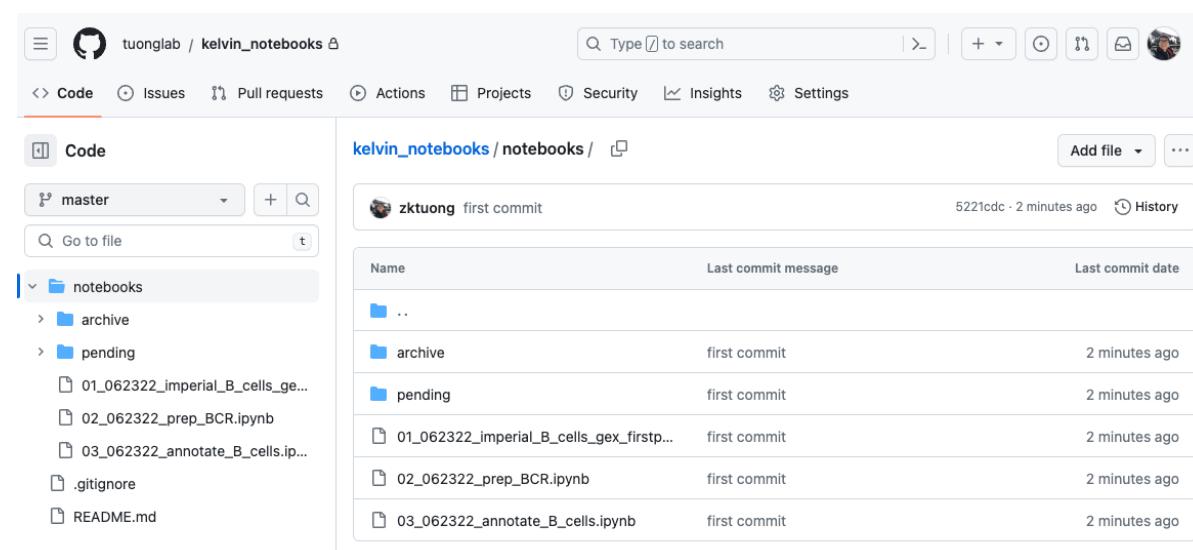
notebooks .gitignore README.md

Initial commit 15 minutes ago Initial commit 15 minutes ago

README.md

kelvin_notebooks

my notebooks

A screenshot of a GitHub repository page for "tuonglab / kelvin_notebooks". The repository has 2 commits. The first commit is from "zktuong" titled "first commit" made 2 minutes ago. The commit message is "5221cdc". The commit details show a commit history for the "notebooks" folder, listing three files: "archive", "pending", and "01_062322_imperial_B_cells_ge...". The second commit is also from "zktuong" and is identical to the first.

Code

Code

master +

Go to file

notebooks

archive pending 01_062322_imperial_B_cells_ge... 02_062322_prep_BCR.ipynb 03_062322_annotate_B_cells.ip...

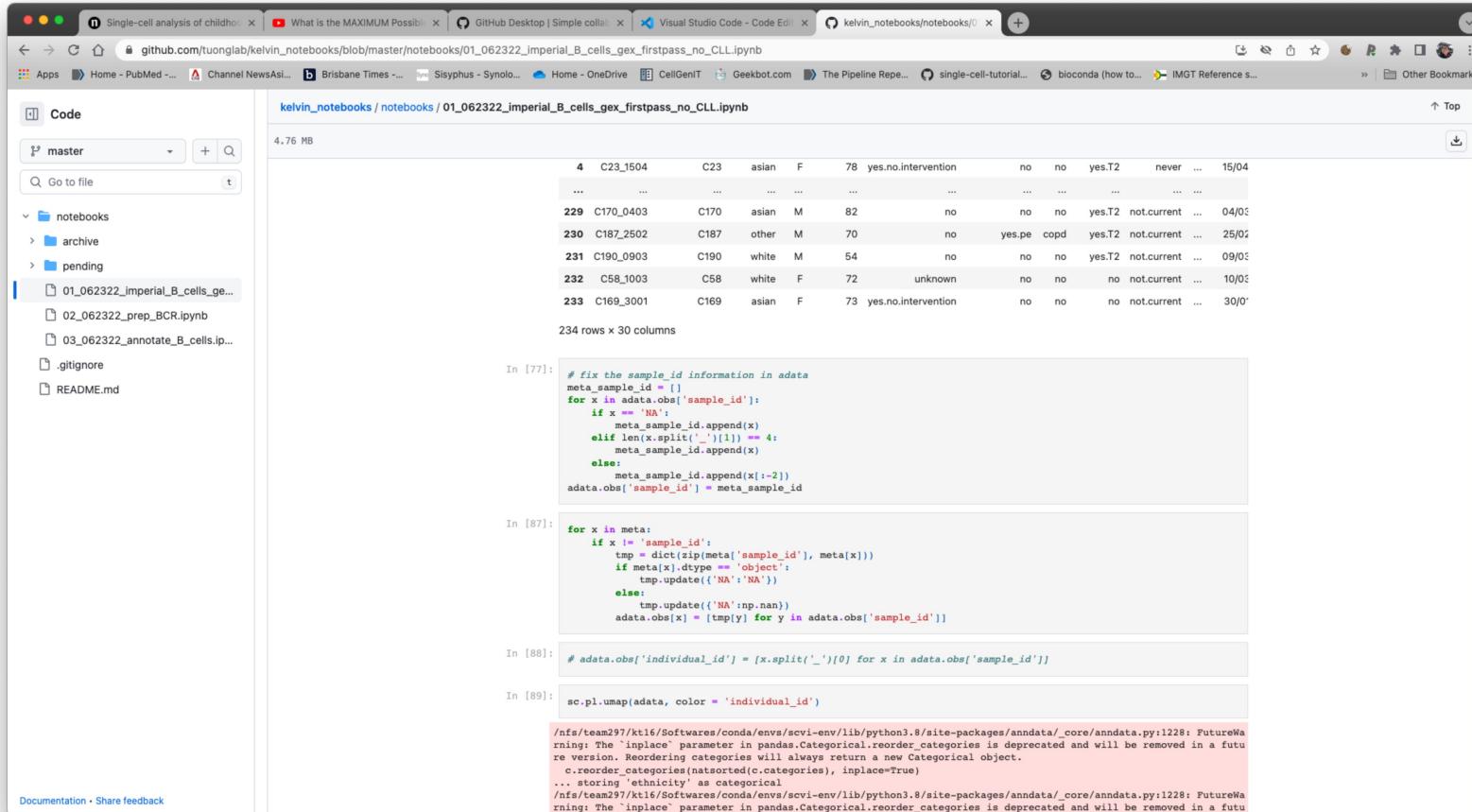
.gitignore README.md

zktuong first commit 5221cdc - 2 minutes ago History

Name	Last commit message	Last commit date
archive	first commit	2 minutes ago
pending	first commit	2 minutes ago
01_062322_imperial_B_cells_ge...	first commit	2 minutes ago
02_062322_prep_BCR.ipynb	first commit	2 minutes ago
03_062322_annotate_B_cells.ip...	first commit	2 minutes ago

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 running in a browser. The left sidebar shows a file tree with notebooks, archive, pending, and other files like .gitignore and README.md. The main area displays a code cell (In [77]) containing Python code to fix sample ID information in a dataset. Below it is another code cell (In [87]) with more data processing code. Further down are cells (In [88] and In [89]) for data cleaning and visualization using scikit-learn's umap.

```

# 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

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

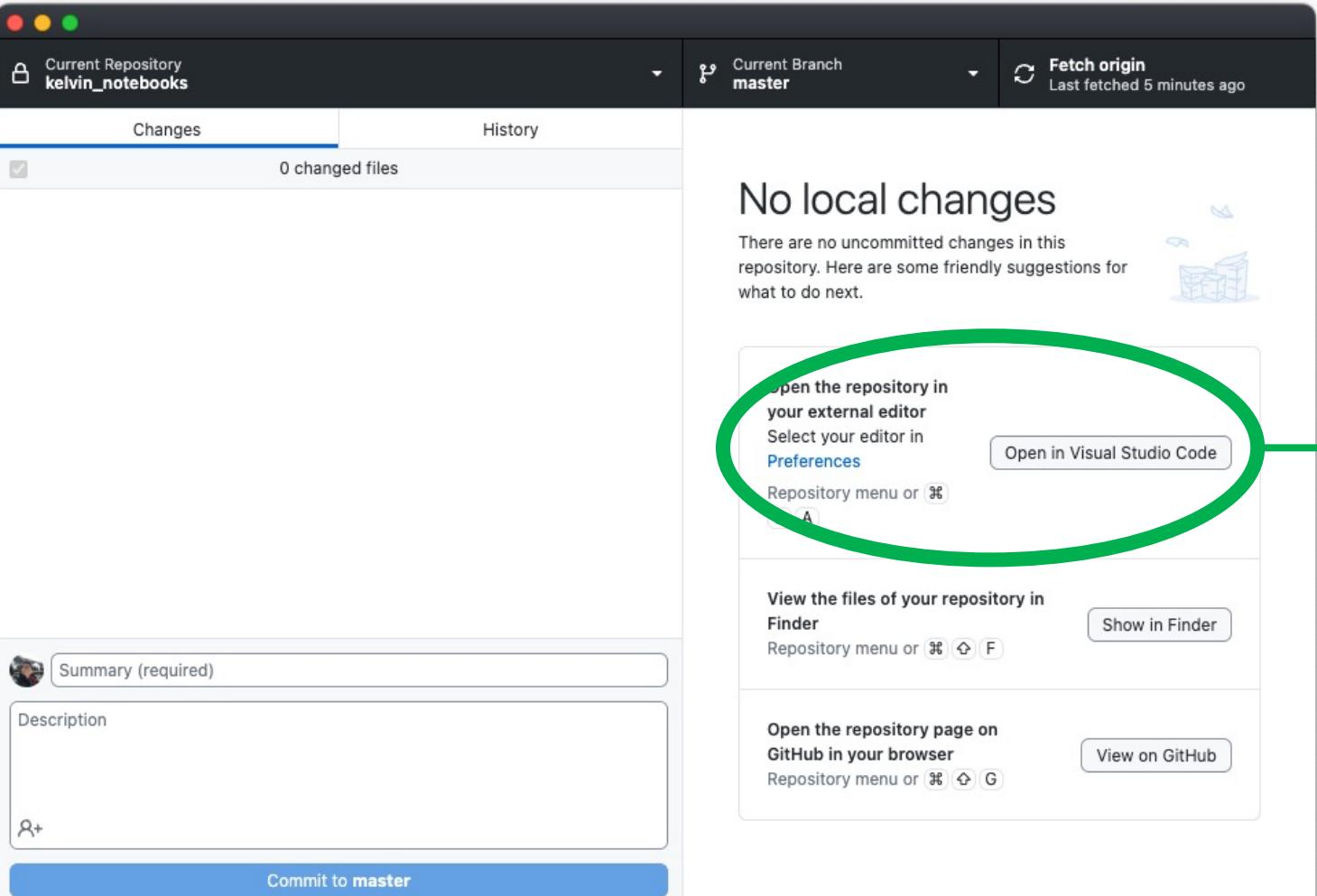
# adata.obs['individual_id'] = [x.split('_')[0] for x in adata.obs['sample_id']]

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

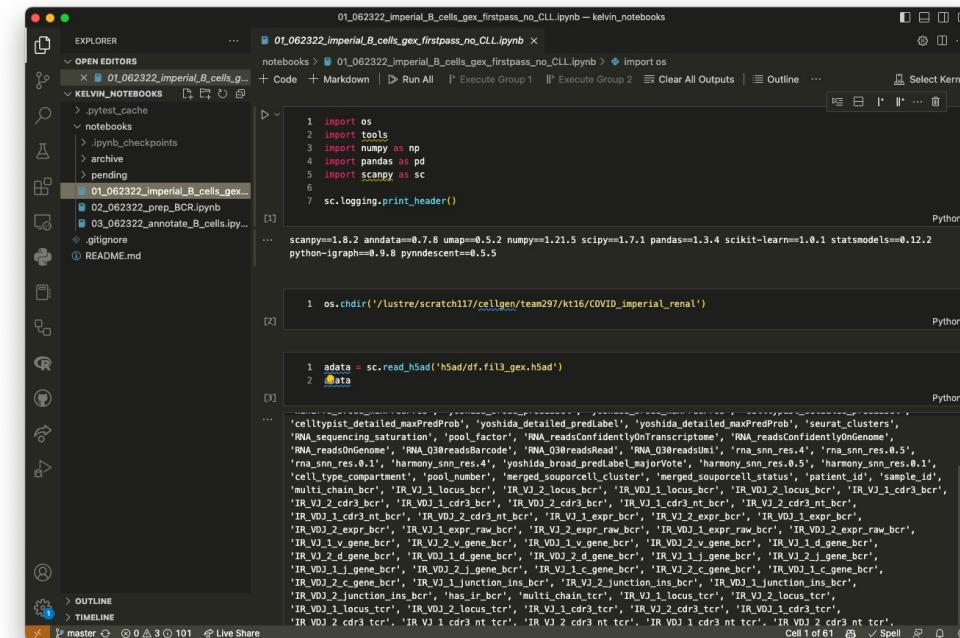
```

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 suggestions with buttons: "Open the repository in your external editor", "View the files of your repository in Finder", and "Open the repository page on GitHub in your browser". A green oval highlights the "Open the repository in your external editor" button, which is circled with a green arrow pointing to the right. At the bottom, there is a "Commit to master" button.



The screenshot shows the Visual Studio Code interface 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 the following Python code:

```

1 import os
2 import tools
3 import numpy as np
4 import pandas as pd
5 import scanpy as sc
6
7 sc.logging.print_header()
8
9 ... scanpy==1.8.2 umdata==0.7.8 umap==0.5.2 numpy==1.21.5 scipy==1.7.1 pandas==1.3.4 scikit-learn==1.0.1 statsmodels==0.12.2
10 python-igraph==0.9.8 pyndescent==0.5.5

```

The second cell contains:

```

1 adata = sc.read_h5ad('h5ad/df.fil3_gex.h5ad')
2 #adata

```

The third cell contains:

```

1 ... 'celltypist_detailed_maxPredProb', 'yoshida_detailed_predLabel', 'yoshida_detailed_maxPredProb', 'seurat_clusters',
2 'RNA_sequencing_saturation', 'pool_factor', 'RNA_readsConfidentlyOnTranscript', 'RNA_readsConfidentlyOnGenome',
3 'RNA_read0nGenome', 'RNA_Q30readBarcode', 'RNA_Q30readBarcode', 'RNA_Q30readBarcode', 'RNA_Q30readBarcode', 'rna_smn_res_0.5',
4 'rna_smn_res_0.1', 'rna_smn_res_0.2', 'rna_smn_res_0.3', 'rna_smn_res_0.4', 'rna_smn_res_0.6', 'rna_smn_res_0.8',
5 'multi_chain_bcr', 'IR_V1_1_locus_bcr', 'IR_V1_2_locus_bcr', 'IR_VD1_1_locus_bcr', 'IR_VD1_2_locus_bcr',
6 'IR_V1_2_cdr3_bcr', 'IR_VD1_1_cdr3_bcr', 'IR_VD1_2_cdr3_bcr', 'IR_V1_1_cdr3_nt_bcr', 'IR_VD1_2_cdr3_nt_bcr',
7 'IR_VD1_1_cdr3_nt_bcr', 'IR_VD1_2_cdr3_nt_bcr', 'IR_V1_1_expr_bcr', 'IR_V1_2_expr_bcr', 'IR_VD1_1_expr_bcr',
8 'IR_VD1_2_expr_bcr', 'IR_V1_1_expr_raw_bcr', 'IR_V1_2_expr_raw_bcr', 'IR_VD1_1_expr_raw_bcr', 'IR_VD1_2_expr_raw_bcr',
9 'IR_V1_1_v_gene_bcr', 'IR_V1_2_v_gene_bcr', 'IR_VD1_1_v_gene_bcr', 'IR_VD1_2_v_gene_bcr', 'IR_V1_1_d_gene_bcr',
10 'IR_V1_2_d_gene_bcr', 'IR_VD1_1_d_gene_bcr', 'IR_V1_1_j_gene_bcr', 'IR_V1_2_j_gene_bcr', 'IR_VD1_1_j_gene_bcr',
11 'IR_VD1_2_j_gene_bcr', 'IR_V1_1_c_gene_bcr', 'IR_V1_2_c_gene_bcr', 'IR_VD1_1_c_gene_bcr', 'IR_VD1_2_c_gene_bcr',
12 'IR_VD1_2_c_junction_ins_bcr', 'IR_V1_2_c_junction_ins_bcr', 'IR_VD1_1_junction_ins_bcr', 'IR_VD1_2_junction_ins_bcr',
13 'IR_VD1_2_c_junction_ins_bcr', 'has_ir_bcr', 'multi_chain_tcr', 'IR_V1_1_locus_tcr', 'IR_V1_2_locus_tcr',
14 'IR_VD1_1_locus_tcr', 'IR_VD1_2_locus_tcr', 'IR_V1_1_cdr3_tcr', 'IR_V1_2_cdr3_tcr', 'IR_VD1_1_cdr3_tcr',
15 'IR_VD1_2_cdr3_tcr', 'IR_V1_1_cdr3_nt_tcr', 'IR_V1_2_cdr3_nt_tcr', 'IR_VD1_1_cdr3_nt_tcr', 'IR_VD1_2_cdr3_nt_tcr'.

```

The status bar at the bottom indicates "Cell 0 of 61" and "Spell".

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
- You wouldn't upload (push) your actual data to GitHub as there is a size limit for repo. You would upload mostly scripts, notebooks, and notes here and there.
- 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

 [tuonglab/tuong-group-onboarding](#) 

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 *

 [tuonglab](#) /

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.