**Chapter 9**

**Keeping the Model Up-To Date Through the Github Shared Platform**

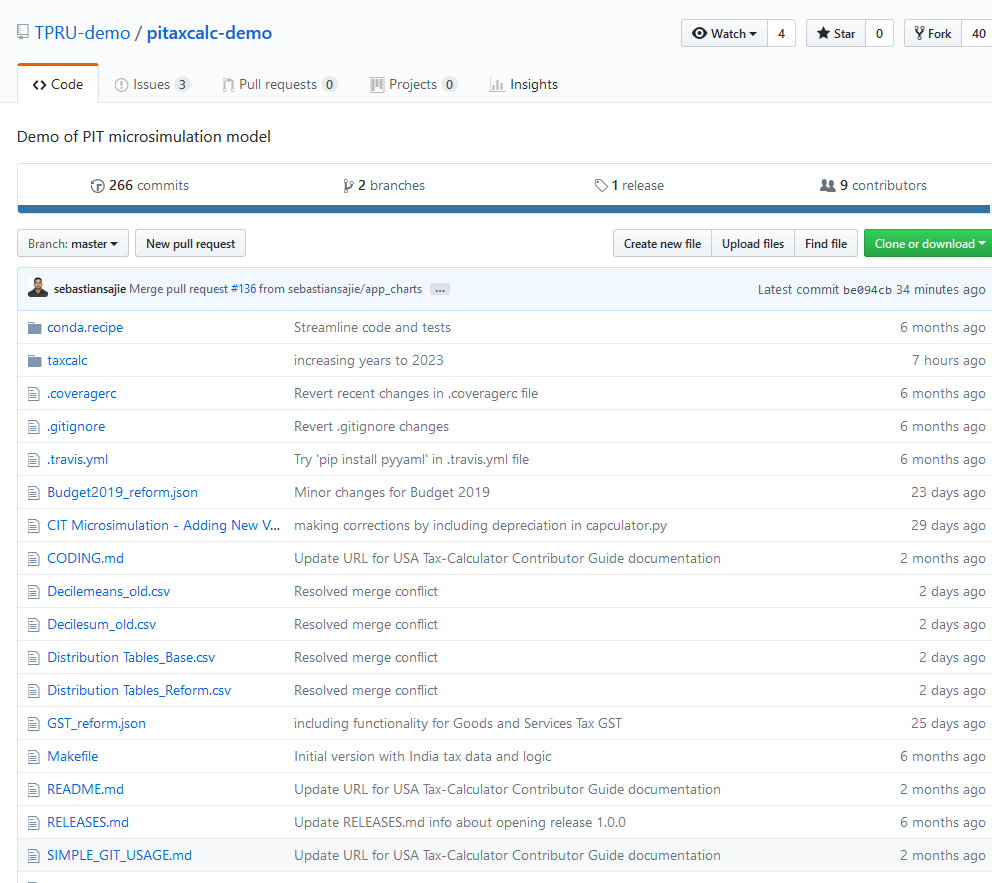
Github is a software platform that allows users to work on a project in a collaborative manner. The Git software is designed specifically for workflows where several users work on the same set of files (usually coding projects), contribute to changes and develop the project in a structured and ordered fashion. The Github platform is particularly well suited for the microsimulation model since several users are simultaneously contributing Python code and adding new functionalities to the model from different locations.

Github.com is an online host of open source repositories[[1]](#footnote-1) which is free to join and use. Each user can make an account on **www.github.com** and obtain access to all the repositories hosted there. The microsimulation model is currently hosted on [www.github.com](http://www.github.com) under two open-access repositories-

1. TPRU-demo/pitaxdata-demo
2. TPRU-demo/pitaxcalc-demo

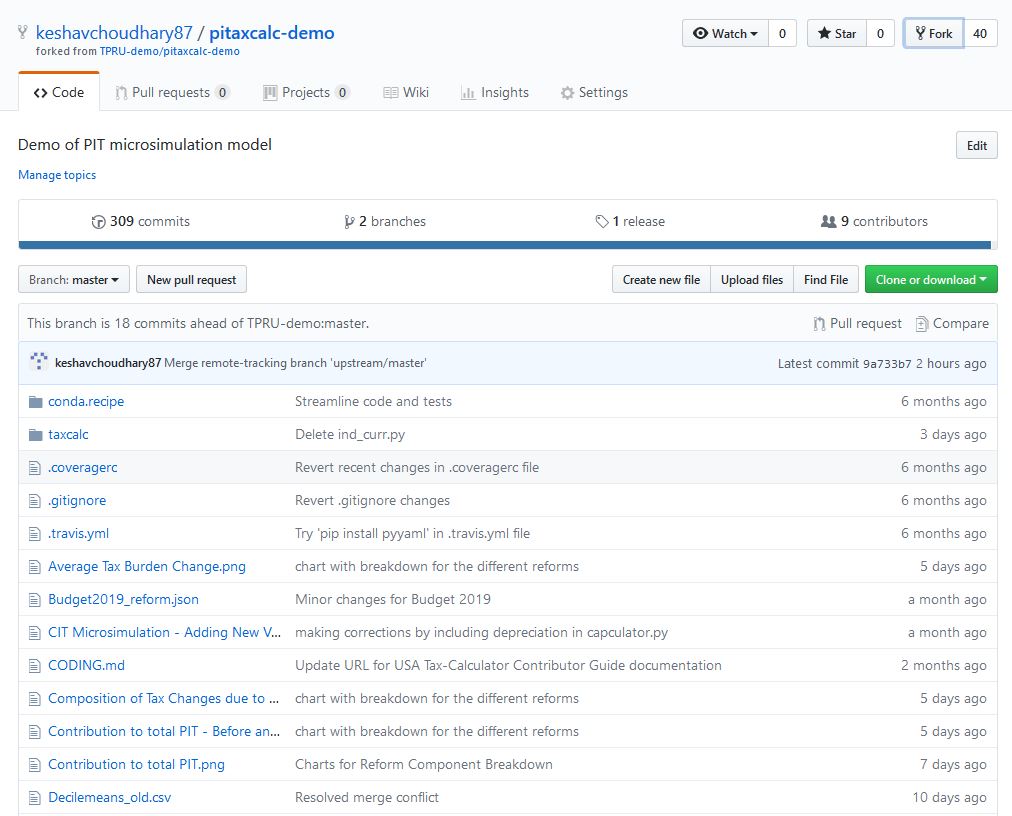
**How to get started on Github**

1. Download git software on your machine. It is freely available at <https://git-scm.com/download/win>. Note: git software comes pre-installed on Mac machines.
2. Go to [www.github.com](http://www.github.com) and create a free account.
3. In the search bar, type in the name of the repository and go to its page. To illustrate, the TPRU-demo/pitaxcalc-demo repository page should look like the following:



**Figure 1- Snapshot of the main repository: TPRU-demo/pitaxcalc-demo**

1. Click on the “Fork” button in the top right corner. This creates a copy of the main repository and links it to your Github account. Both repositories should be forked in this manner to your account. The forked copy looks nearly the same as the main repository except for the top left corner, which notes that the repository is a forked copy. The snapshot of a forked copy of TPRU-demo/pitaxcalc-demo is given below-

****

**Figure 2: Forked copy of TPRU-demo/pitaxcalc-demo**

**Using Github as a final user**

This section is relevant for those users who intend to use Github only as final consumers i.e to download (referred to as cloning), run and update the microsimulation model with the latest changes on their computers. Once cloned, the repository can be updated in future using a simple set of commands.

Cloning an Online Repository

The following steps should be followed in the sequence listed below to clone a repository. We take the example of TPRU-demo/pitaxcalc-demo to illustrate the process:

1. Create a folder in your machine where you want the repositories to reside.
2. Open Anaconda Prompt (Terminal Window for Mac users) and type in the following commands to configure git:
   1. *git config -- global user.name “****your username****”*
   2. *git config -- global user.email “****youremail@abc.com****”*
3. Next, clone the online repository into the folder you have created:
   1. In Anaconda prompt, navigate to the folder you created using the *cd* command. You can type *cd* and drag the folder into Anaconda prompt. This will automatically give the path of the folder.
   2. Type: *git clone https://github.com/****your username****/pitaxcalc-demo.git.* Alternatively, you can get the link by going to your forked copy of the repository, and clicking on the button “*Clone or Download*” which will show the link. Executing the clone command will copy all the files from your online forked repository to your machine. After the process is complete, you should see a folder named “Pitaxcalc-demo” inside the folder you created.
   3. Navigate inside “Pitaxcalc-demo” by typing cd and dragging the folder into Anaconda prompt.
   4. Now type *git remote add upstream* [*https://github.com/TPRU-demo/pitaxcalc-demo.git*](https://github.com/TPRU-demo/pitaxcalc-demo.git). Alternatively, you can get the link by going to the main repository and clicking on the button “*Clone or Download”.*

Step 3 has to be repeated for the TPRU-demo/pitaxdata-demo repository. To do so, simply repeat Step 3 by typing pitaxdata-demo in place of pitaxcalc-demo.

Once this process has been carried out, cloned copies of both repositories will be stored on your machine inside the folder you created.

Updating an Online Repository

Cloning a repository is a one-time process. Thereafter, the repositories only have to be updated periodically (or synced). The process is as follows:

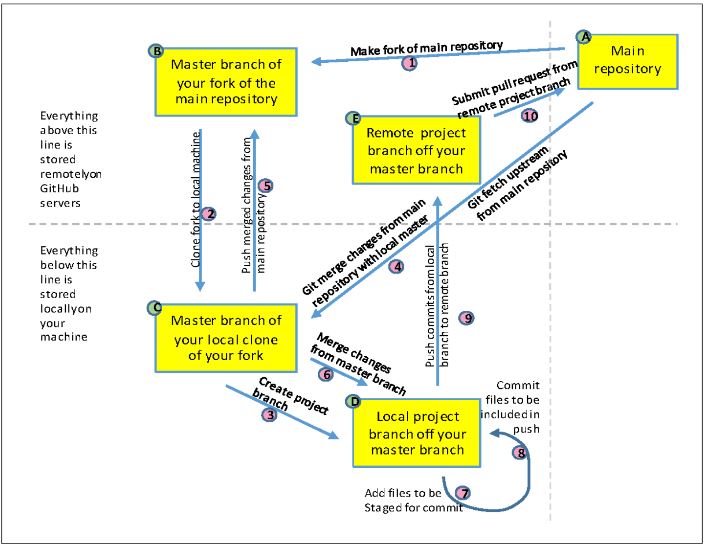
Open Anaconda prompt and navigate to the cloned copy of the repository on your machine. This can be done by typing cd and then dragging the pitaxcalc-demo or pitaxdata-demo folder into Anaconda prompt.

1. Type the following three commands-
   1. *git fetch upstream*
   2. *git merge upstream/master*
   3. *git push origin master*

This will update the repository on your machine as well as your online forked copy, in tune with the main repository.

**Using Github as a contributor/developer**

This section is relevant for users who intend to contribute to the microsimulation models by writing code in their local machines and then have them incorporated into the main repository through Github. To understand the process, it is essential to be familiar with the Github workflow. The schematic diagram is given below-

****

**Figure 2: Git workflow**

The principles governing the workflow are given below-

1. The main repository resides in the cloud. The only way to modify it is by submitting a *pull request* which is done through Github and must be approved by the administrators of the repository.
2. Users fork their copy of the main repository, which is linked to their Github account and also resides in the cloud.
3. By cloning, users create a copy of their forked repository in their local machine. This is also referred to as the ‘Master’ branch of the forked repository.
4. Whenever a user works on the model which involves modifying the code or adding new files, this should be done by creating a new branch off the master branch. Each repository can have unlimited branches.
5. If new files have been modified, added or deleted in the branch, the user needs to *add* the files before they can be saved and sent to the cloud. If the files are not added using the *add* command, they will be invisible to Git and will not be pushed to the cloud.
6. Next, the user *commits* all their changes using the *commit* command. This essentially saves all the changes made by the user.
7. The user then *pushes* the commit using the *push* command. This creates a branch of the repository in your fork in the cloud.
8. The branch then submits a pull request to the main repository.

Git commands to implement the workflow

The process of forking and cloning has already been discussed in the previous section. To create branches and modify the code, the process is as follows:

1. Open Anaconda prompt and navigate to the directory of the repository.
2. Type: *git branch.* This will list all the branches of the repository. The branch in which you are currently in will be highlighted with ‘\*’.
3. Type: *git checkout -b* ***new branch name.*** This will create a new branch of the repository and will change your directory to that branch. You can now make changes to the files, add new files or delete files etc.
4. If instead of creating a new branch, you want to navigate between branches, type: *git checkout* ***branch name.***
5. After modifying the code, changing files etc in the branch, type: *git status.* This will list if any files have been modified, added or deleted. These files will be shown in red. Additionally, files which are untracked by Git will also appear in red under the heading ‘Untracked files- ’. Typically these are data files which we want to prevent from going to the cloud and want to keep untracked throughout.
6. Next, type: *git add* ***file name*** for the modified/added files that need to be pushed to the cloud. The files which are added through this command will now start appearing in green (can be checked by typing *git status* again). The rest of the files in red (including untracked files) will remain invisible to git.
7. Next, type: *git commit -m* ***“short message within quotes”*.** This will commit or save the added files i.e the files which were appearing in green.
8. Thereafter, type: *git push origin* ***branch name*.** This will push the branch to the cloud into your forked repository.
9. To submit a pull request, go to your fork of the main repository on [www.github.com](http://www.github.com). Click on the button named “Branch” on the left hand side, and go to the branch which contains the changes which need to be submitted. Thereafter, click on the button “New Pull Request”, right next to it.
10. Once the pull request is approved by the administrators, you can type-
    1. *git fetch upstream*
    2. *git merge upstream/master*
    3. *git push origin master*

This will sync your master branch and online forked copy in tune with the main repository.

1. Finally, you can delete the branch which is no longer required by typing: *git branch -d* ***branch name.***

1. A repository is a collection of files. [↑](#footnote-ref-1)