GitHub Manual for EGAP NRG Metaketa III

Eugenia Nazrullaeva, `enazrullaeva@ucla.edu`
Miriam Golden, `golden@ucla.edu`

March 19, 2018

Contents

1 Introduction (adapted from GitHub Guides)

"**GitHub** is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere."[1] The GitHub workflow includes working with files in repositories stored on GitHub, committing changes to them, and pushing (or syncing) your changes back to GitHub.

A **repository** (repo, for short) organizes a single project. "Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs."[2] Our repository is called "Metaketa-III". It is private, meaning only collaborators invited by Miriam can view and change files.

"On GitHub, saved changes [to any file] are called **commits**. Each commit has an associated commit message, which is a description... [of each change]. Commit messages capture the history of your changes, so other contributors can understand what you have done and why."[3] Good work practices include making modular commits — one commit for each set of changes — and writing clear, concise commit messages.

The workflow at the heart of collaboration on GitHub includes: (1) establishing (**cloning**) an initial copy of a remote repo on your computer, (2) **pulling** (**syncing**) the updated version of the repo from GitHub, which reflects changes made by your collaborators since you last pulled, and (3) **committing and pushing** (**committing and syncing**) the changes you make on your local machine to any document back to the remote repo on GitHub for your collaborators to see.

With GitHub, multiple users can work on the same document simultaneously. GitHub will automatically integrate changes made by multiple users. If two users simultaneously edit the same line in a file, GitHub will inform you that there is a conflict and ask you to resolve it manually. If this happens, please feel free to ask Miriam for real-time help if you are not sure what to do!

GitHub offers the best version control system currently available. This not only allows you to see and revert to different versions of the file, knowing who changed it and how, but **traces changes within a file made by multiple collaborators** using *git blame* (more on this here). Also, every commit saves the entire repository, so it is easy to go back in time to earlier versions of your project if you want.

Once you have cloned (i.e. copied) the repo (i.e. all the project files) to your local computer(-s), your **workflow** will always be: (1) **pull** (**sync**), so you have the most recent version of all files; (2) edit any file(-s) you wish, using your normal editor; (3) **commit** your changes and record what you did with a commit message; (4) **push** (**sync**) your changes back to GitHub so everyone else has them. A good practice is to commit and push (sync) whenever you walk away from your computer, in case you get distracted — you don't want your work to get lost! And don't forget to pull when you sit down so you

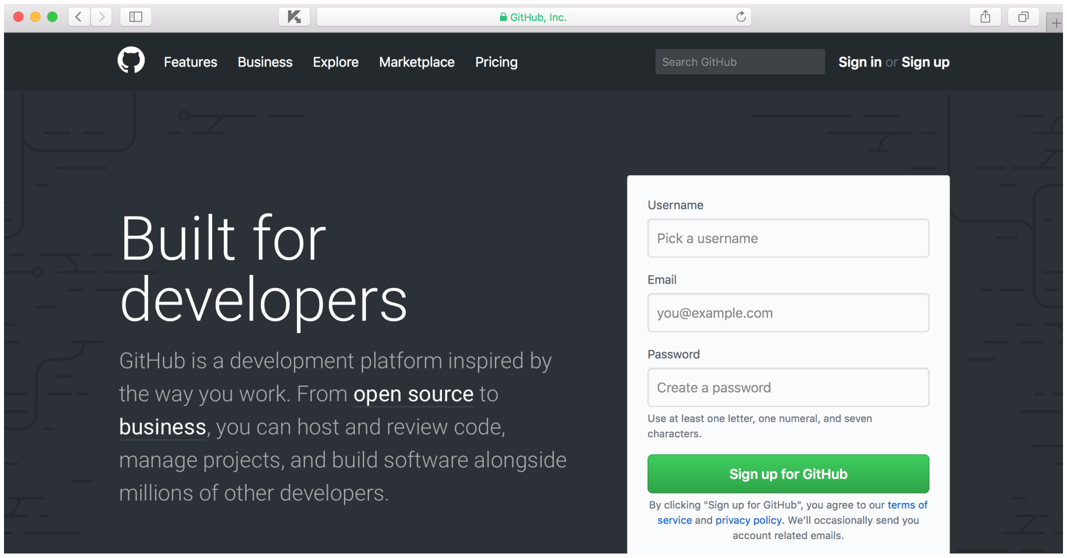
[1] GitHub Guides.
[2] Ibid.
[3] Ibid.

have everyone else's updates.

Finally, at the top of the Metaketa repo on GitHub, you will see the **Issues** tab. Click on it to see any tasks assigned to you, and use it to discuss progress with or ask questions of others assigned to the same task. Use of the Issues tab will automatically generate email records of the discussion.

2 Sign up for a GitHub account (and upgrade to a free educational account)

Let's create a GitHub Account first if you do not have one. Sign up for GitHub here: https://github.com. Make sure to remember your username, because you are going to need it later in order to be added to the Metaketa III repository on GitHub.



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.

Username: Pick a username

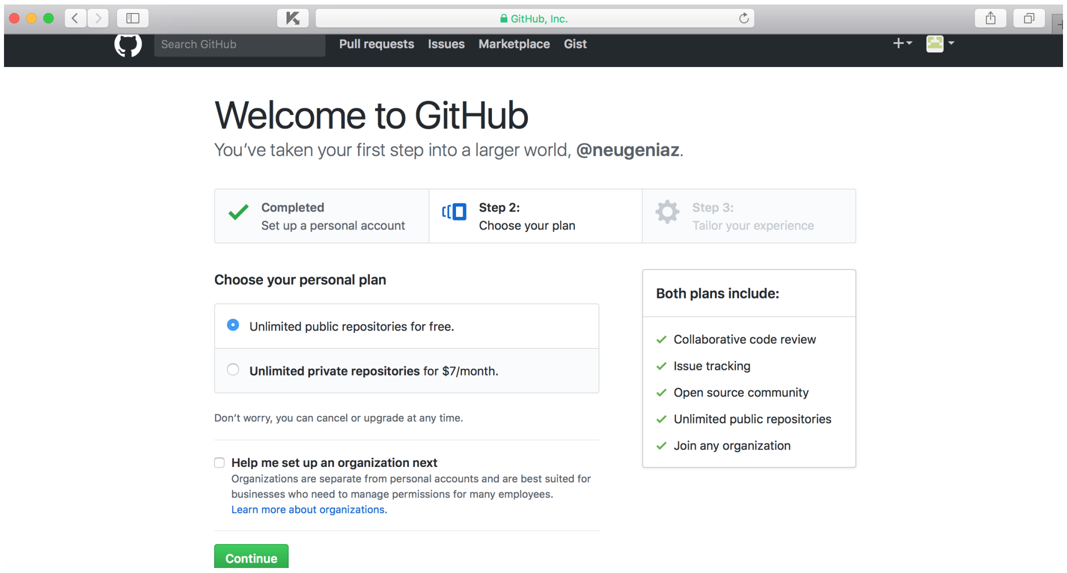
Email: you@example.com

Password: Create a password

Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

At the next step, you see options for public/private repositories. Choose the first option, "Unlimited public repositories for free," because we are going to upgrade it later to "Unlimited private repositories" for free by setting up a free educational account with private repositories.



Welcome to GitHub

You've taken your first step into a larger world, **@neugeniaz**.

✓ Completed — Set up a personal account | Step 2: Choose your plan | Step 3: Tailor your experience

Choose your personal plan

● Unlimited public repositories for free.

○ **Unlimited private repositories** for $7/month.

Don't worry, you can cancel or upgrade at any time.

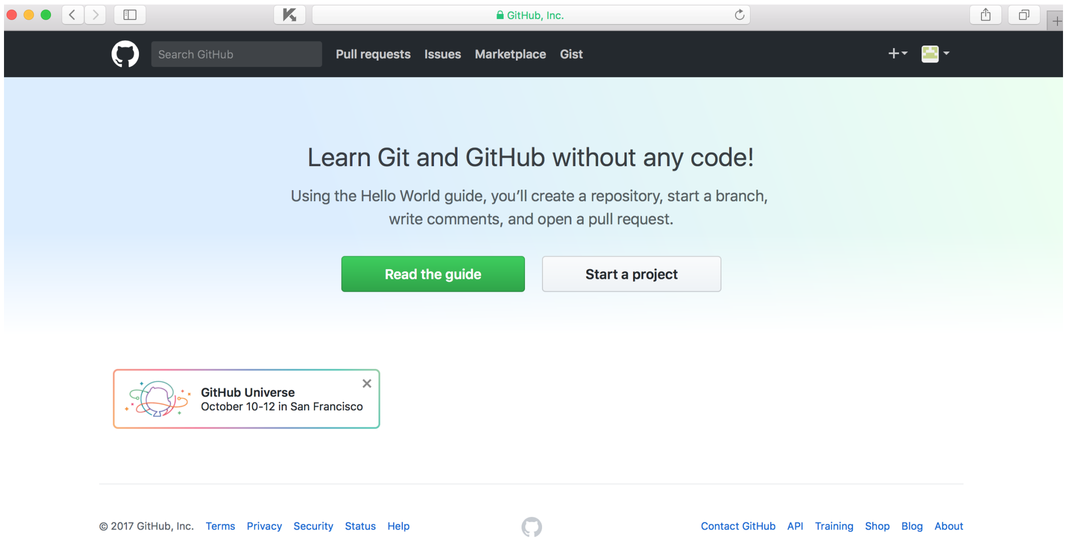
☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations.

Continue

Both plans include:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

At the third step, you are asked to provide some details about your programming background and occupation, etc. Finally, GitHub is going to ask you to verify your email address. That should be it. This is what you should see at the final stage:



Learn Git and GitHub without any code!

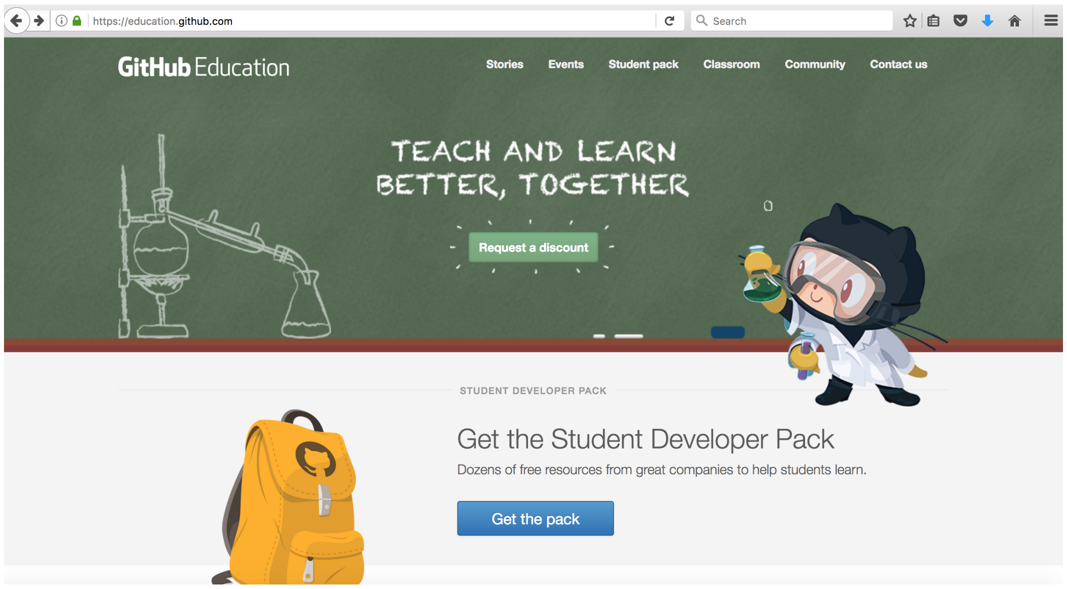
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide | Start a project

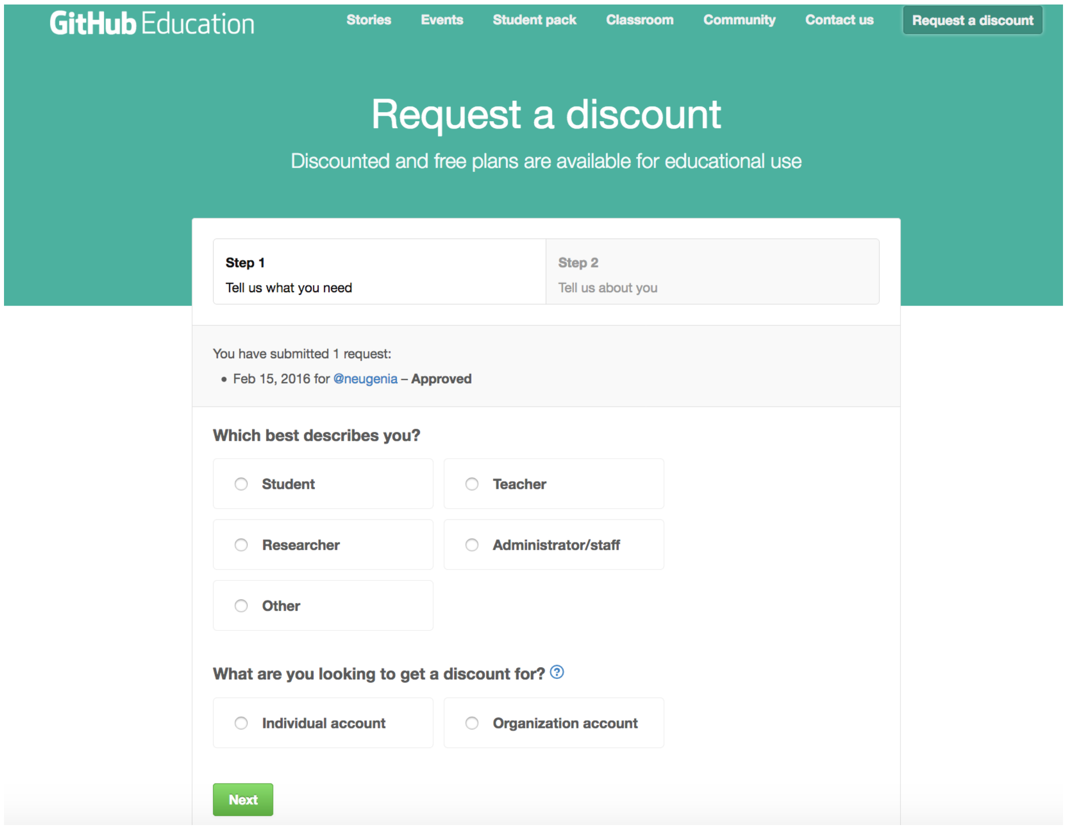
GitHub Universe
October 10-12 in San Francisco

© 2017 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

Now let's upgrade your GitHub account to make it a free educational account with private repositories. Go to https://education.github.com/ and click "Request a discount":


https://education.github.com
Search
GitHub Education
Stories
Events
Student pack
Classroom
Community
Contact us
TEACH AND LEARN BETTER, TOGETHER
Request a discount
STUDENT DEVELOPER PACK
Get the Student Developer Pack
Dozens of free resources from great companies to help students learn.
Get the pack

Fill in the necessary details (Researcher/Individual account).[4]



You have submitted 1 request:

- Feb 15, 2016 for @neugenia – **Approved**

Which best describes you?

○ Student ○ Teacher ○ Researcher ○ Administrator/staff ○ Other

What are you looking to get a discount for?

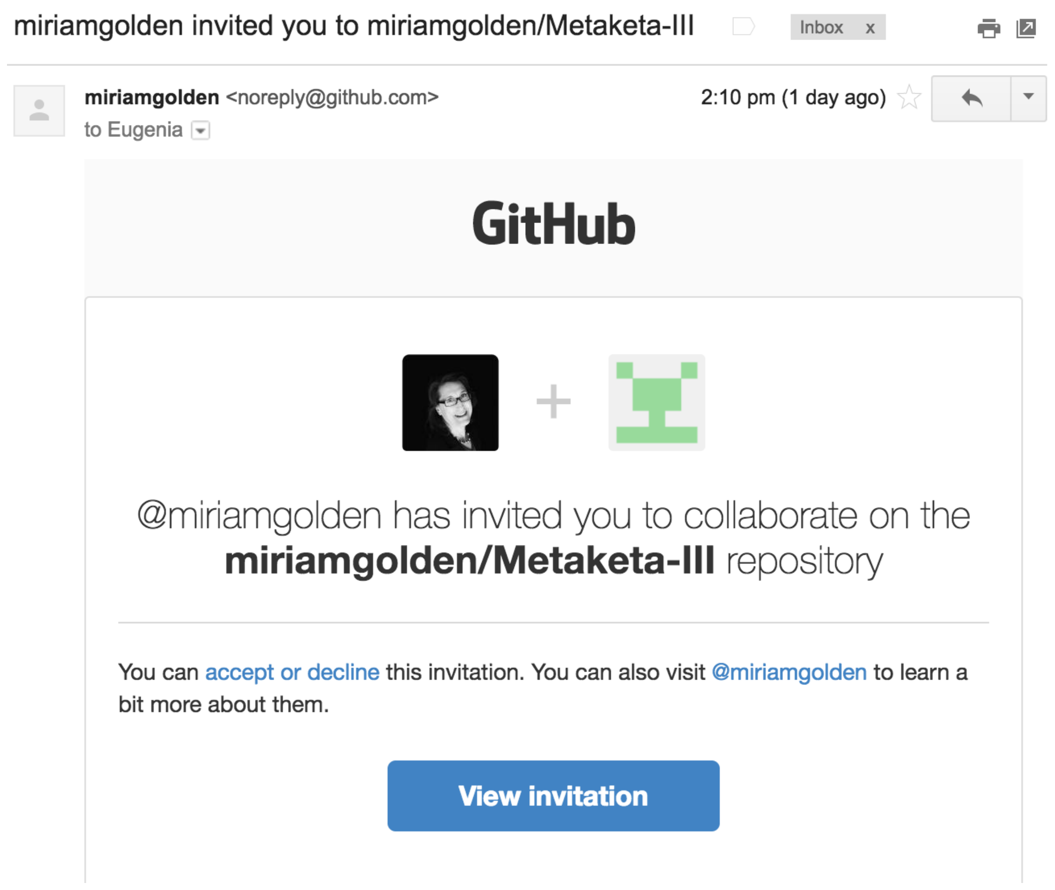
○ Individual account ○ Organization account

At the next step, you are going to be asked to fill in more details about yourself and how you are going to use GitHub, as well as provide your University email address. That should be it!

[4]You may notice that I already have an approved discount.

3 Request access to the shared private repository Metaketa III

When you have your GitHub account set up, send an email to Miriam Golden (golden@ucla.edu) and request access to the private repository Metaketa III. To be added to this repository, you are going to need your username (as you are going to see in the examples below, my username is "neugenia"). Knowing your username, MG can give you access. You are going to get an email which looks like this:



miriamgolden invited you to miriamgolden/Metaketa-III

miriamgolden <noreply@github.com>
to Eugenia

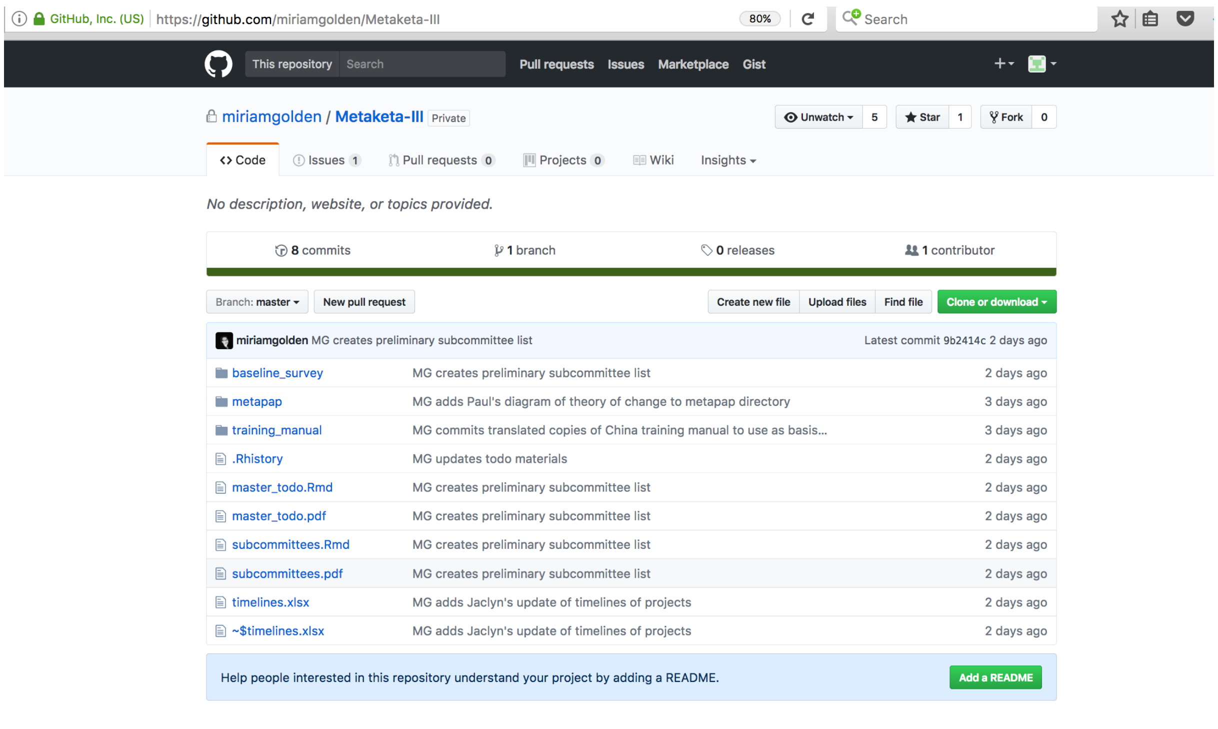
2:10 pm (1 day ago)

GitHub

@miriamgolden has invited you to collaborate on the **miriamgolden/Metaketa-III** repository

You can accept or decline this invitation. You can also visit @miriamgolden to learn a bit more about them.

Click on "View invitation" and accept it. When you accept the invitation, you should be able to see all the materials from the repository "Metaketa III" online here https://github.com/miriamgolden/Metaketa-III.


https://github.com/miriamgolden/Metaketa-III

miriamgolden / Metaketa-III Private

No description, website, or topics provided.

8 commits · 1 branch · 0 releases · 1 contributor

Branch: master

miriamgolden MG creates preliminary subcommittee list — Latest commit 9b2414c 2 days ago

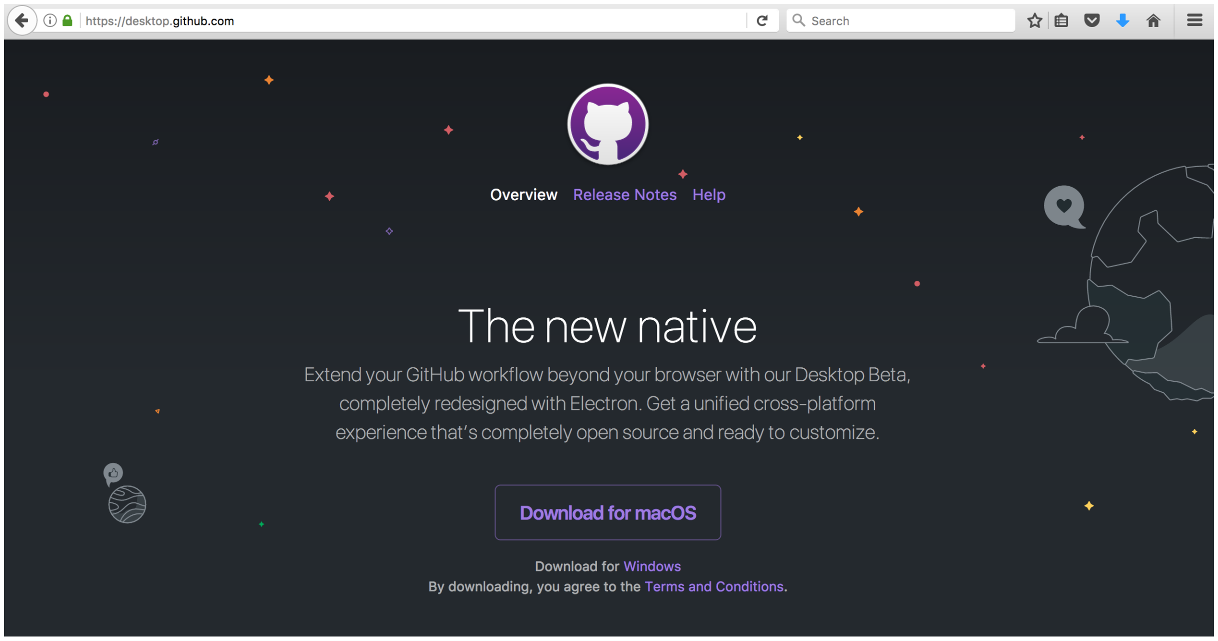
baseline_survey	MG creates preliminary subcommittee list	2 days ago
metapap	MG adds Paul's diagram of theory of change to metapap directory	3 days ago
training_manual	MG commits translated copies of China training manual to use as basis...	3 days ago
.Rhistory	MG updates todo materials	2 days ago
master_todo.Rmd	MG creates preliminary subcommittee list	2 days ago
master_todo.pdf	MG creates preliminary subcommittee list	2 days ago
subcommittees.Rmd	MG creates preliminary subcommittee list	2 days ago
subcommittees.pdf	MG creates preliminary subcommittee list	2 days ago
timelines.xlsx	MG adds Jaclyn's update of timelines of projects	2 days ago
~$timelines.xlsx	MG adds Jaclyn's update of timelines of projects	2 days ago

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

4 Collaborate on the project via the GitHub Desktop application

4.1 Install GitHub Desktop

Install the "GitHub Desktop" application/program. Go here: https://desktop.github.com/. It is available both for Mac and Windows.

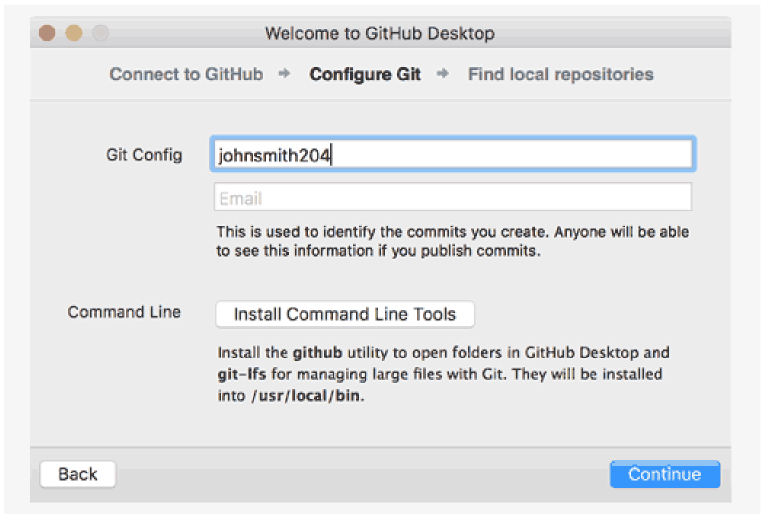

https://desktop.github.com

The new native

Extend your GitHub workflow beyond your browser with our Desktop Beta, completely redesigned with Electron. Get a unified cross-platform experience that's completely open source and ready to customize.

Download for Windows
By downloading, you agree to the Terms and Conditions.

When you open GitHub Desktop you see:



Welcome to GitHub Desktop

Connect to GitHub → **Configure Git** → Find local repositories

Git Config: johnsmith204

Email

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

Command Line: Install Command Line Tools

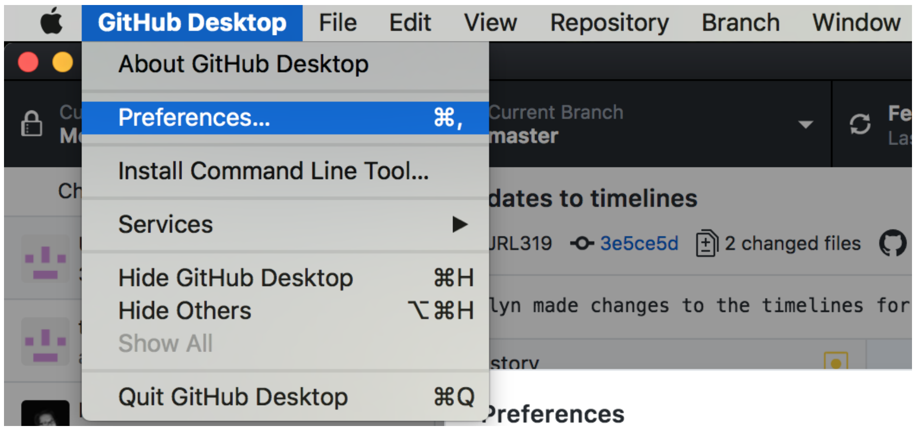
Install the **github** utility to open folders in GitHub Desktop and **git-lfs** for managing large files with Git. They will be installed into **/usr/local/bin**.

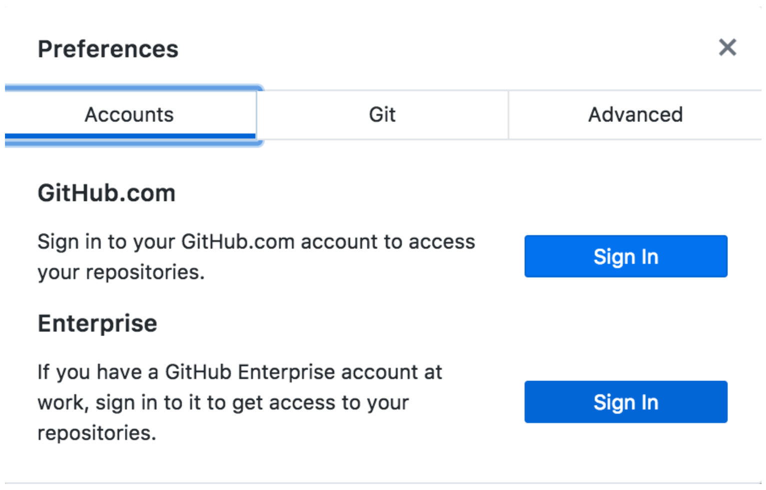
In order to work with the application you need to sign in. You need your username and password for the GitHub account.

If you skip the welcome step, you can always sign in (and sign out) using the menu bar at the top.

On Mac: Go to the upper-left corner and select: GitHub Desktop → Preferences → Accounts

On Windows: Go to the upper-right corner and click the Settings icon (log in), select: Settings → Account




Preferences

Accounts | Git | Advanced

GitHub.com

Sign in to your GitHub.com account to access your repositories.

Sign In

Enterprise

If you have a GitHub Enterprise account at work, sign in to it to get access to your repositories.

Sign In

4.2 Add the Metaketa-III repository to your computer

Now we need to add the Metaketa-III repository (repo) to your computer.

For example, I am going to copy the remote repo Metaketa III with all its files to the following directory on my computer:

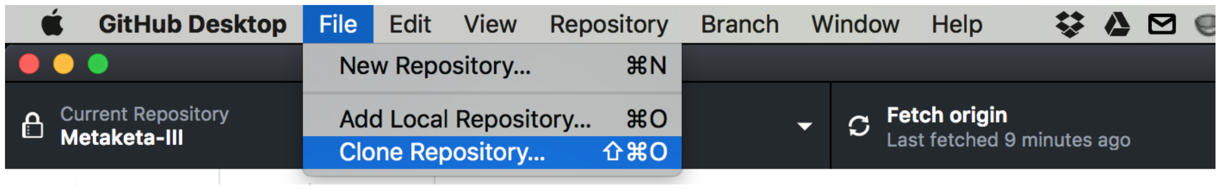
On Mac:

```
~/GitHub/metaketa-iii_egap
```

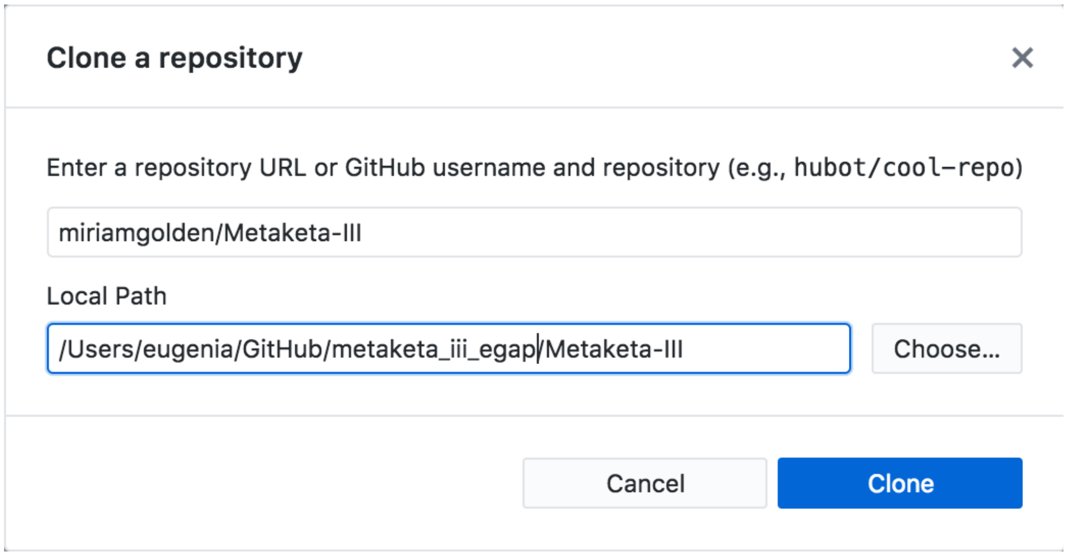
On Windows:

```
C:\Users\Eugenia\GitHub\metaketa-iii_egap
```

Let's go to the menu bar at the top-left and choose File → Clone Repository (both on Mac and Windows):



If you successfully accepted the invitation to join the private repo Metaketa III, in the menu below type in **miriamgolden/Metaketa-III**:


Clone a repository

Enter a repository URL or GitHub username and repository (e.g., hubot/cool-repo)

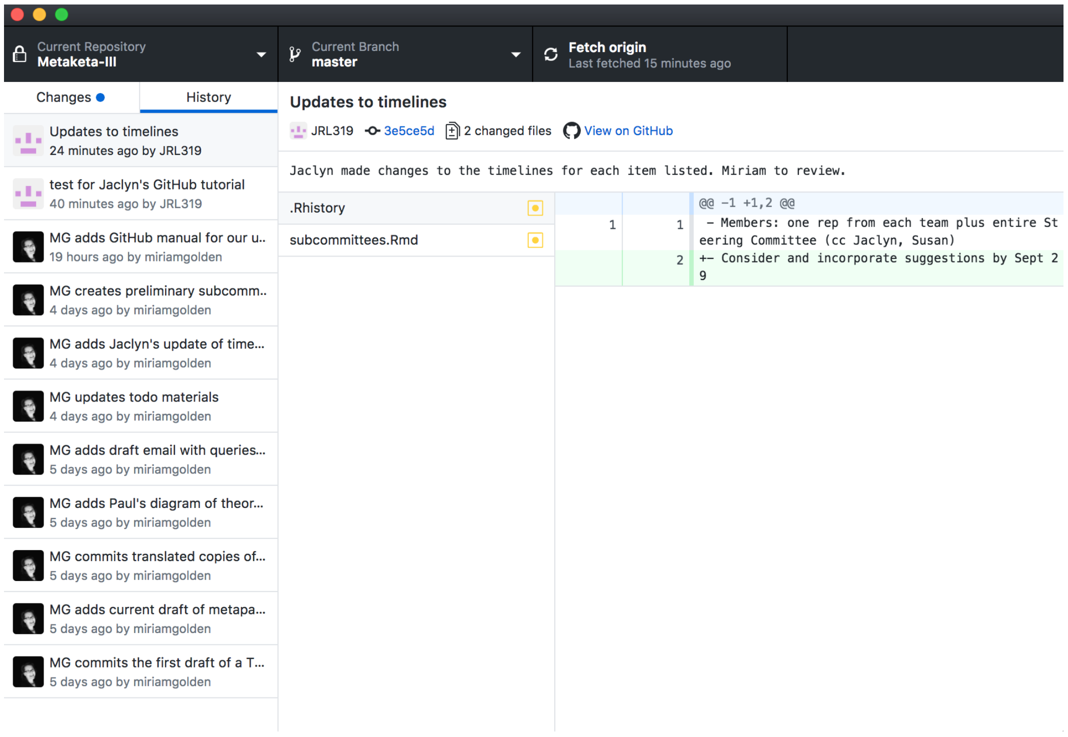
miriamgolden/Metaketa-III

Local Path

/Users/eugenia/GitHub/metaketa_iii_egap/Metaketa-III

Choose a folder ("Local Path") on your computer where you want to keep a copy of the repo (it is going to be linked to the remote repo). You could choose any folder/path. Now click "Clone".

If you successfully "cloned" the folder, you should be able to see its contents via the GitHub Desktop.


Current Repository: Metaketa-III
Current Branch: master
Fetch origin — Last fetched 15 minutes ago

- Updates to timelines — 24 minutes ago by JRL319
- test for Jaclyn's GitHub tutorial — 40 minutes ago by JRL319
- MG adds GitHub manual for our u... — 19 hours ago by miriamgolden
- MG creates preliminary subcomm... — 4 days ago by miriamgolden
- MG adds Jaclyn's update of time... — 4 days ago by miriamgolden
- MG updates todo materials — 4 days ago by miriamgolden
- MG adds draft email with queries... — 5 days ago by miriamgolden
- MG adds Paul's diagram of theor... — 5 days ago by miriamgolden
- MG commits translated copies of... — 5 days ago by miriamgolden
- MG adds current draft of metapa... — 5 days ago by miriamgolden
- MG commits the first draft of a T... — 5 days ago by miriamgolden

Updates to timelines

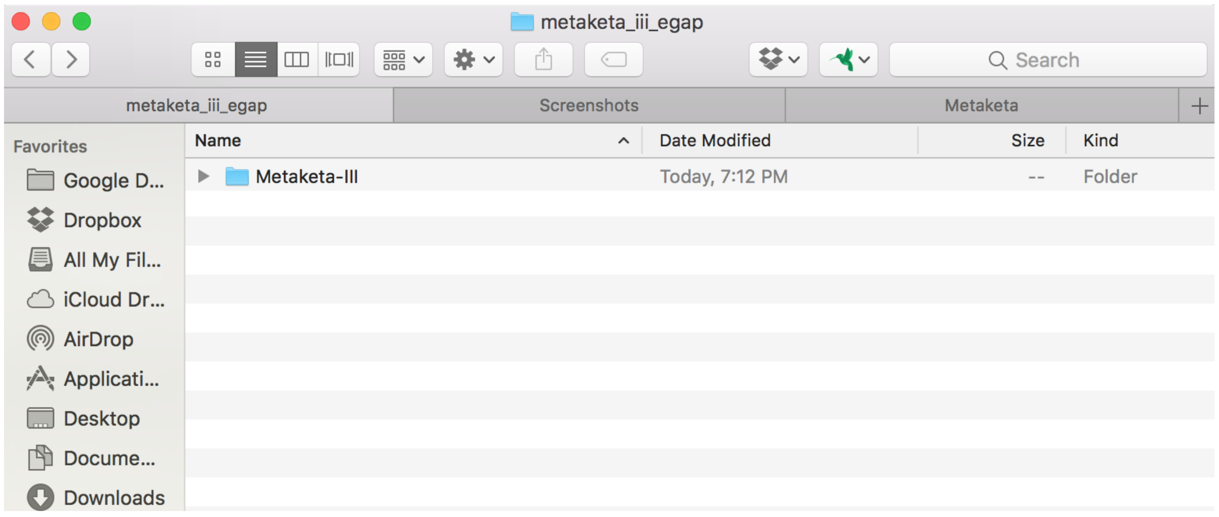
JRL319 3e5ce5d 2 changed files View on GitHub

```
Jaclyn made changes to the timelines for each item listed. Miriam to review.
```

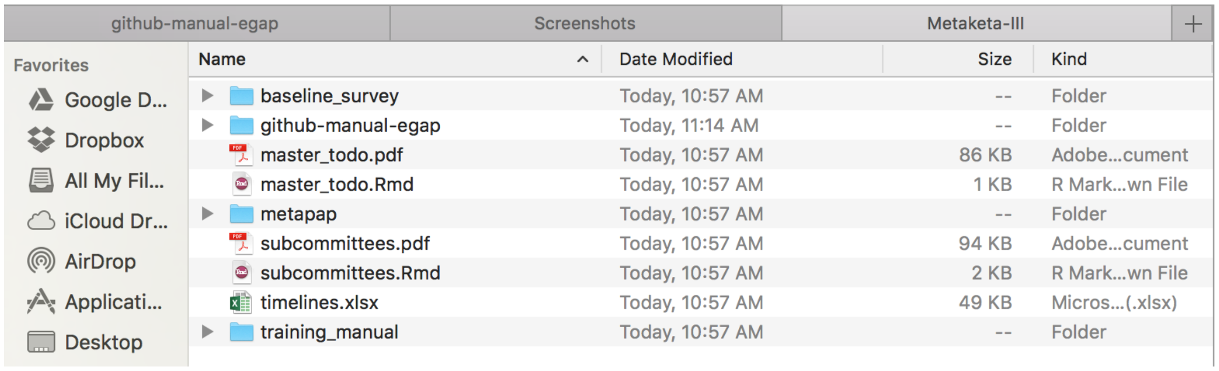
.Rhistory
subcommittees.Rmd

```
@@ -1 +1,2 @@
1  1   - Members: one rep from each team plus entire Steering Committee (cc Jaclyn, Susan)
   2  +- Consider and incorporate suggestions by Sept 29
```

as well as in the folder which you created on your computer:

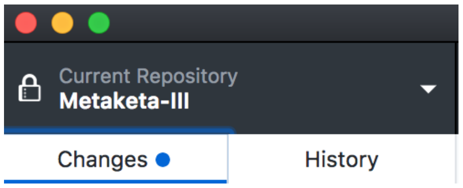


Name	Date Modified	Size	Kind
Metaketa-III	Today, 7:12 PM	--	Folder



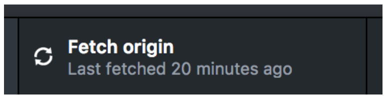
Name	Date Modified	Size	Kind
baseline_survey	Today, 10:57 AM	--	Folder
github-manual-egap	Today, 11:14 AM	--	Folder
master_todo.pdf	Today, 10:57 AM	86 KB	Adobe...cument
master_todo.Rmd	Today, 10:57 AM	1 KB	R Mark...wn File
metapap	Today, 10:57 AM	--	Folder
subcommittees.pdf	Today, 10:57 AM	94 KB	Adobe...cument
subcommittees.Rmd	Today, 10:57 AM	2 KB	R Mark...wn File
timelines.xlsx	Today, 10:57 AM	49 KB	Micros...(.xlsx)
training_manual	Today, 10:57 AM	--	Folder

To the left you can see two panels: "Changes" and "History". "Changes" reflects the changes to the repo that you made on your computer, and "History" keeps track of the changes that your collaborators made to the remote repo.



4.3 Sync the repo on your computer to reflect someone else's changes: "Fetch origin" / "Pull origin"

Important: Whenever you want to update your local copy of the joint/remote repo, click the button "Fetch origin" to sync changes.

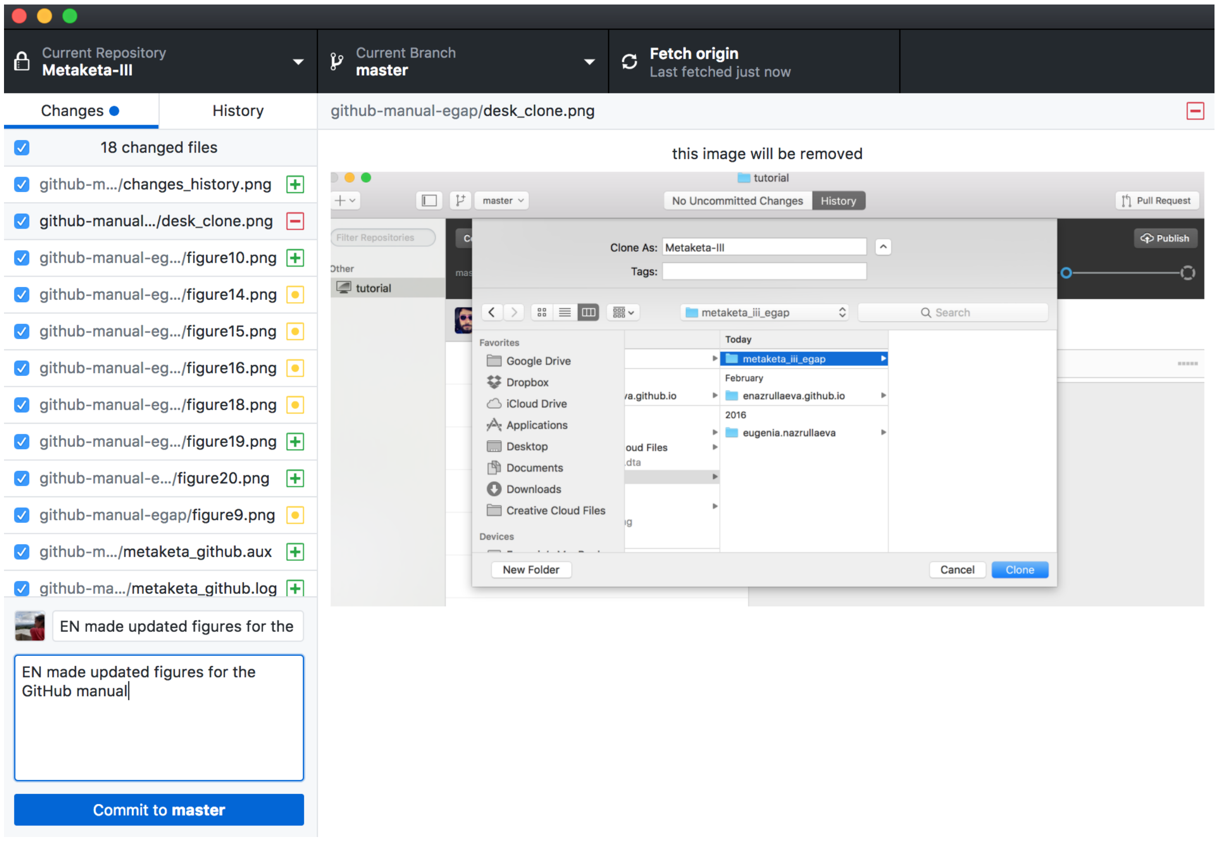


"Fetch origin" changes keeps track of the changes that you made in your local repository (on your computer) and the changes someone else made and committed to the remote repo.

If someone else made changes to the remote repo which you do not yet have on your local computer, when you click on the "Fetch origin" button it becomes the **"Pull origin"** button. To update the files on your computer click the **"Pull origin"** button. This is how you **pull** changes from the remote repo to your computer.

4.4 Sync the repo on your computer to reflect your changes: "Commit" and "Fetch origin" / "Push origin"

Suppose you have a new file to contribute, or you edited an existing file and want to commit changes. For example, I updated the figures for this manual on my local computer (in the folder which we synced with GitHub), and GitHub Desktop automatically reflects the changes under the "Changes" panel. If it does not reflect all the changes you made, click on the "Fetch origin" button.



18 changed files

- github-m.../changes_history.png
- github-manual.../desk_clone.png
- github-manual-eg.../figure10.png
- github-manual-eg.../figure14.png
- github-manual-eg.../figure15.png
- github-manual-eg.../figure16.png
- github-manual-eg.../figure18.png
- github-manual-eg.../figure19.png
- github-manual-e.../figure20.png
- github-manual-egap/figure9.png
- github-m.../metaketa_github.aux
- github-ma.../metaketa_github.log

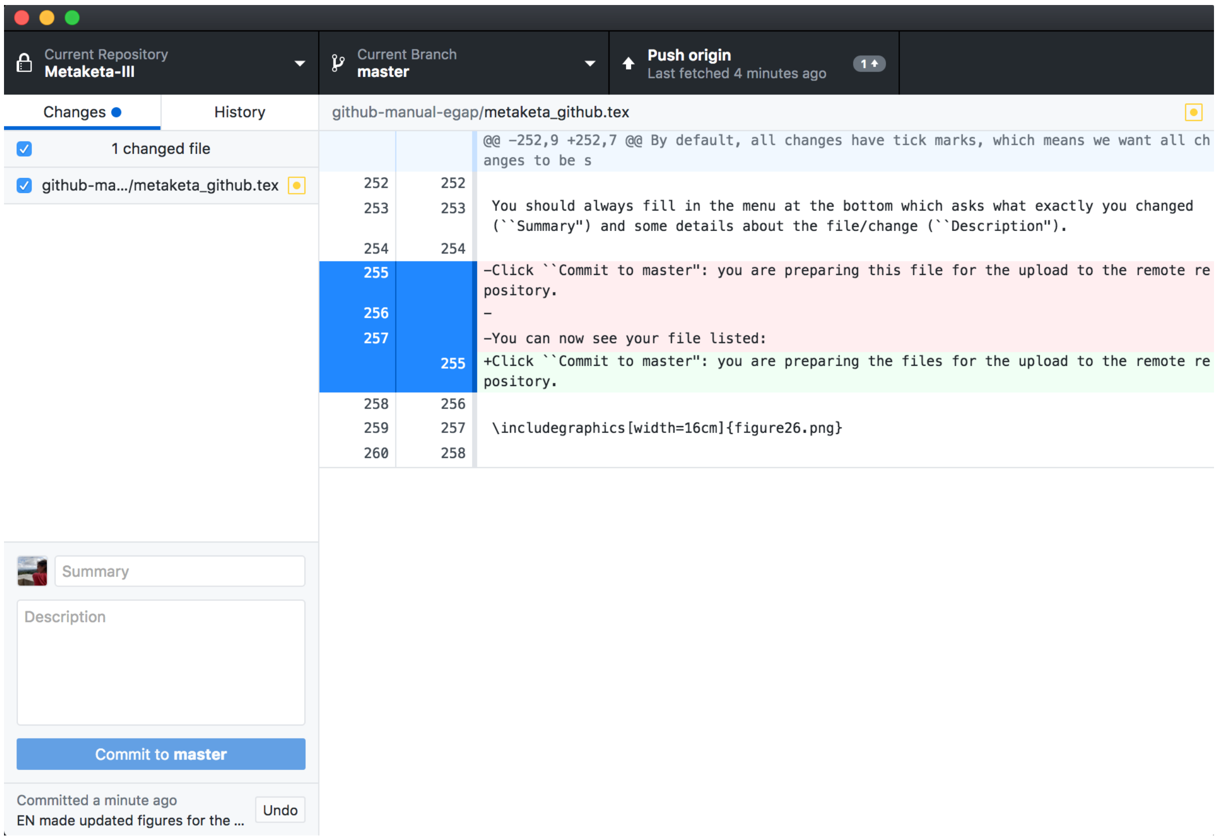
EN made updated figures for the

EN made updated figures for the GitHub manual

By default, all updates to the files are selected, which means we want all changes to be synced with the remote repo.

You should always fill in the menu at the bottom which asks what exactly you changed ("Summary") and some details about the file/change ("Description").

Click the **"Commit to master"** button at the bottom-left corner: you are preparing the files for the upload to the remote repository.



```
@@ -252,9 +252,7 @@ By default, all changes have tick marks, which means we want all changes to be s

 You should always fill in the menu at the bottom which asks what exactly you changed (``Summary") and some details about the file/change (``Description").

-Click ``Commit to master": you are preparing this file for the upload to the remote repository.
-
-You can now see your file listed:
+Click ``Commit to master": you are preparing the files for the upload to the remote repository.

 \includegraphics[width=16cm]{figure26.png}

```

You can now see at the top-right corner that the "Fetch origin" button turned into **"Push origin"**. Click **"Push origin"**. This is how you **push** changes from your computer to the remote repo for your collaborators to see.

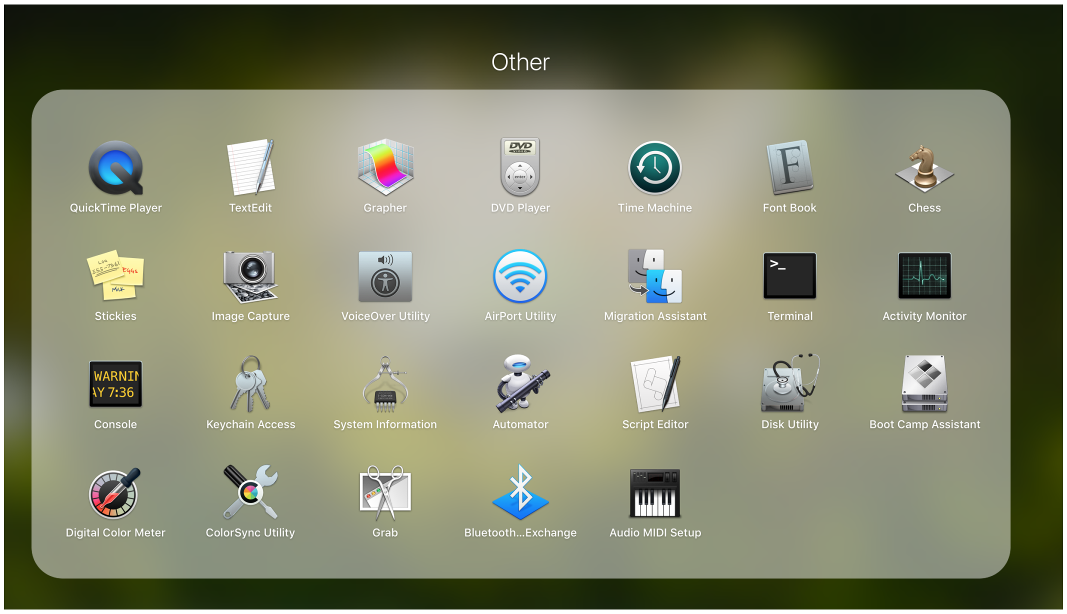
That's it! Remember to always **commit** and **push** all the changes you make.

5 Collaborate on the project via the Terminal / Command prompt

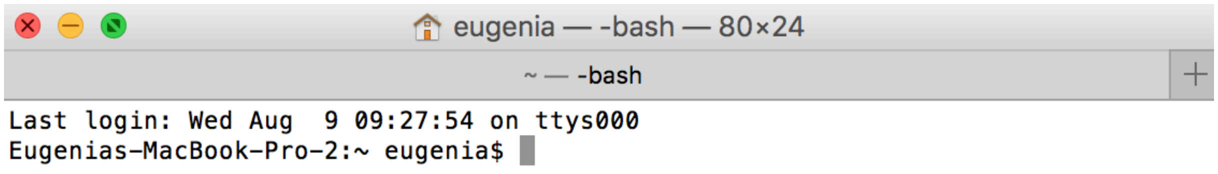
5.1 Open the Terminal on Mac

If you use the terminal, you might prefer to work in GitHub via it. Use of the terminal is more efficient than working with a GUI, and offers a clearer grasp of what you are doing. But you should go with whichever route you prefer (and you can always switch from one to the other).

Open the command shell – the terminal (on Mac you can simply search for "terminal" or find the Terminal in "Other" applications).



Here is the terminal:

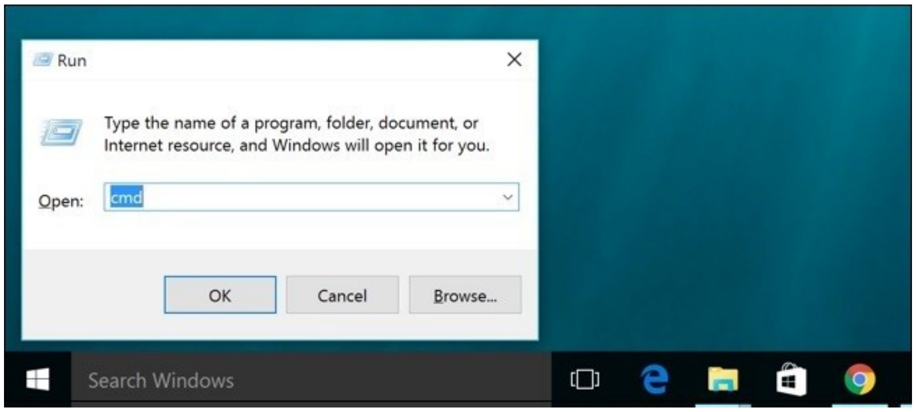


```
Last login: Wed Aug  9 09:27:54 on ttys000
Eugenias-MacBook-Pro-2:~ eugenia$
```

5.2 Open the Command Prompt on Windows

Press Windows+R to open the "Run" box.

Type "cmd" and then click "OK" to open a regular Command Prompt. Type "cmd" and then press Ctrl+Shift+Enter to open an administrator Command Prompt.


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
cmd
OK
Cancel
Browse...
Search Windows

Other ways to open the command prompt are described here.

5.3 Copy the remote repo to your computer

After you set up the GitHub account (your username and password), you need to establish your copy of the Metaketa III repository (repo) on your computer.

First, let's set the default folder where we are going to clone (copy) all files from the Metaketa-III repo. To do this, we need to create an empty folder on your computer. Any working directory is fine, except for Dropbox (GitHub can be in conflict with Dropbox).

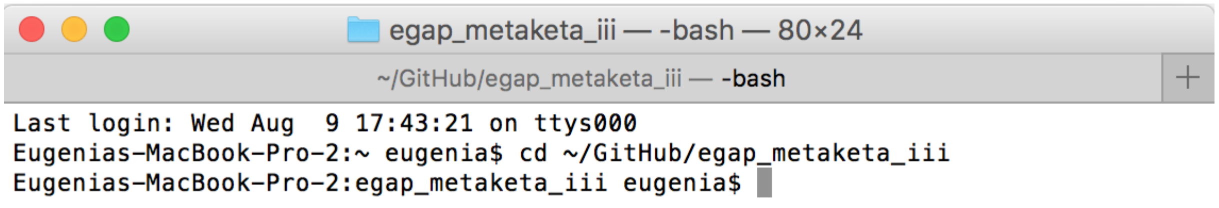
I set this folder as my working directory:

On Mac:

```
cd ~/GitHub/egap_metaketa_iii
```

On Windows:

```
cd C:\Users\Eugenia\GitHub\egap_metaketa_iii
```


egap_metaketa_iii — -bash — 80×24
~/GitHub/egap_metaketa_iii — -bash

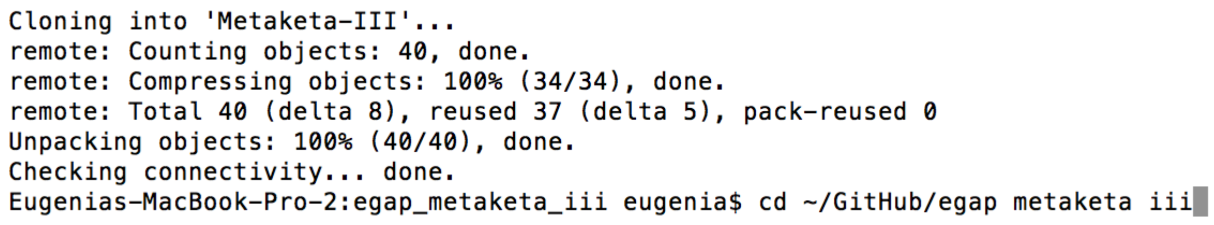
```
Last login: Wed Aug  9 17:43:21 on ttys000
Eugenias-MacBook-Pro-2:~ eugenia$ cd ~/GitHub/egap_metaketa_iii
Eugenias-MacBook-Pro-2:egap_metaketa_iii eugenia$ 
```

Now let's establish your copy of the Metaketa III private repo on your computer, that is let's "clone" this repo. In order to do this, run the following line:

```
git clone https://username:password@github.com/miriamgolden/Metaketa-III.git
```

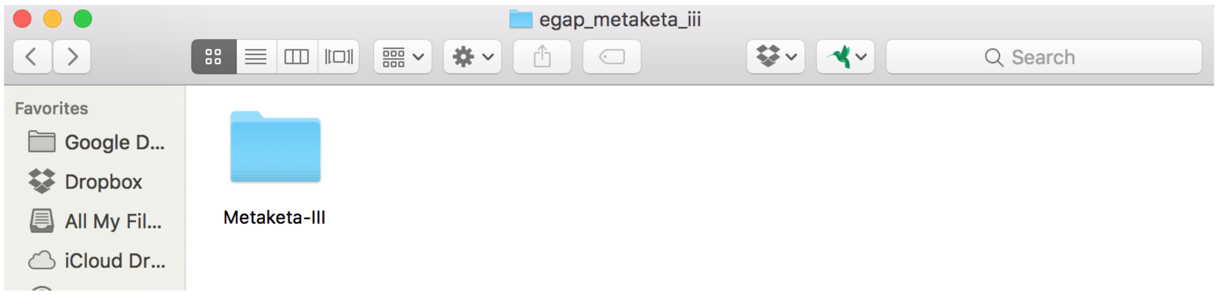
where you change "username" to your username and "password" to your password.

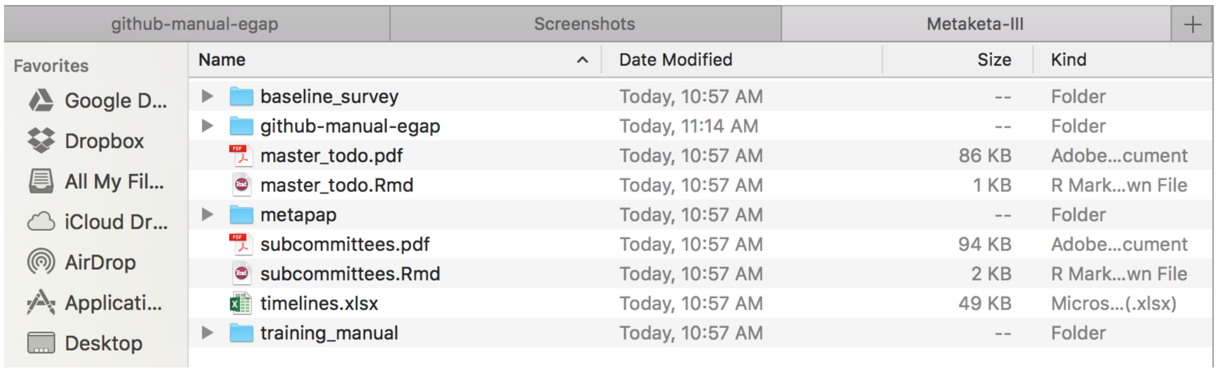
If cloning was successful you should see a report:



```
Cloning into 'Metaketa-III'...
remote: Counting objects: 40, done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 40 (delta 8), reused 37 (delta 5), pack-reused 0
Unpacking objects: 100% (40/40), done.
Checking connectivity... done.
Eugenias-MacBook-Pro-2:egap_metaketa_iii eugenia$ cd ~/GitHub/egap metaketa iii
```

Now if you check your working directory, you should be able to see the folder Metaketa-III with the files which were added from the remote repo.





Name	Date Modified	Size	Kind
baseline_survey	Today, 10:57 AM	--	Folder
github-manual-egap	Today, 11:14 AM	--	Folder
master_todo.pdf	Today, 10:57 AM	86 KB	Adobe...cument
master_todo.Rmd	Today, 10:57 AM	1 KB	R Mark...wn File
metapap	Today, 10:57 AM	--	Folder
subcommittees.pdf	Today, 10:57 AM	94 KB	Adobe...cument
subcommittees.Rmd	Today, 10:57 AM	2 KB	R Mark...wn File
timelines.xlsx	Today, 10:57 AM	49 KB	Micros...(.xlsx)
training_manual	Today, 10:57 AM	--	Folder

5.4 Sync the repo on your computer to reflect someone else's changes: "Pull"

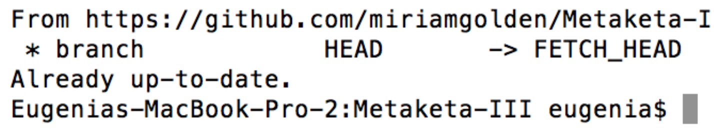
Important: Whenever you want to update your local copy of the repository to reflect changes made by others you do a "pull" from the repository as follows:

```
cd ~/GitHub/egap_metaketa_iii/Metaketa-III
git pull https://username:password@github.com/miriamgolden/Metaketa-III.git
```

where you change "username" to your username and "password" to your password.

In the cd line write down the full path to the Metaketa-III folder which is linked to the remote repo.

In my example, all the files are up to date, so you are going to see the following:

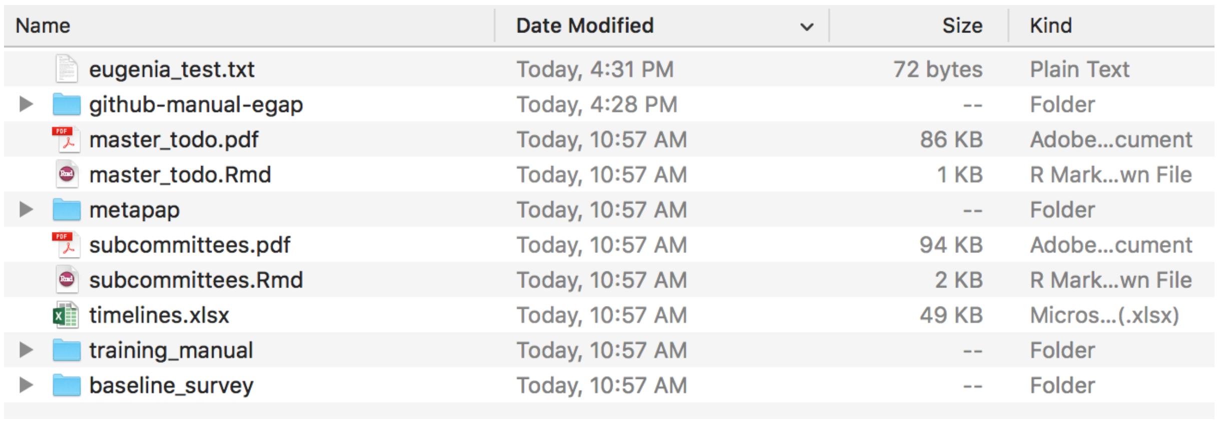


```
From https://github.com/miriamgolden/Metaketa-III
 * branch            HEAD       -> FETCH_HEAD
Already up-to-date.
Eugenias-MacBook-Pro-2:Metaketa-III eugenia$ 
```

5.5 Sync the repo on your computer to reflect your changes: "Push"

Now suppose you have a new file to contribute, or you edited an existing file. To commit changes to the remote repo for everyone to see you need to "push" your changes.

As an example, I added a new empty txt file called "eugenia_test.txt".



Name	Date Modified	Size	Kind
eugenia_test.txt	Today, 4:31 PM	72 bytes	Plain Text
▶ github-manual-egap	Today, 4:28 PM	--	Folder
master_todo.pdf	Today, 10:57 AM	86 KB	Adobe...cument
master_todo.Rmd	Today, 10:57 AM	1 KB	R Mark...wn File
▶ metapap	Today, 10:57 AM	--	Folder
subcommittees.pdf	Today, 10:57 AM	94 KB	Adobe...cument
subcommittees.Rmd	Today, 10:57 AM	2 KB	R Mark...wn File
timelines.xlsx	Today, 10:57 AM	49 KB	Micros...(.xlsx)
▶ training_manual	Today, 10:57 AM	--	Folder
▶ baseline_survey	Today, 10:57 AM	--	Folder

Let's add this text file to the joint repository.

Every time you want to update or add a file to the remote repo you should run the following sequence:

On Mac:

```
cd ~/GitHub/egap_metaketa_iii/Metaketa-III
git add eugenia_test.txt
git commit -m "Eugenia's test txt file for GitHub cheat sheet"
git push https://username:password@github.com/miriamgolden/Metaketa-III.git
```

where you change "username" to your username and "password" to your password.

On Windows:

```
cd C:\Users\Eugenia\GitHub\egap_metaketa_iii\Metaketa-III
git add eugenia_test.txt
git commit -m "Eugenia's test txt file for GitHub cheat sheet"
git push https://username:password@github.com/miriamgolden/Metaketa-III.git
```

First, you should make sure that you have the cd set up, `cd`. Then you add the file you want to sync with the remote repo and "commit" the change, `git add`.

`git commit -m` means that you want to add a message which briefly describes what the new file or the change to the existing file is about. Finally, you run `git push` to sync the remote repo with your the folder on your computer.