

Collaboration with Git

Patrick McCann

Research Computing, University of St Andrews

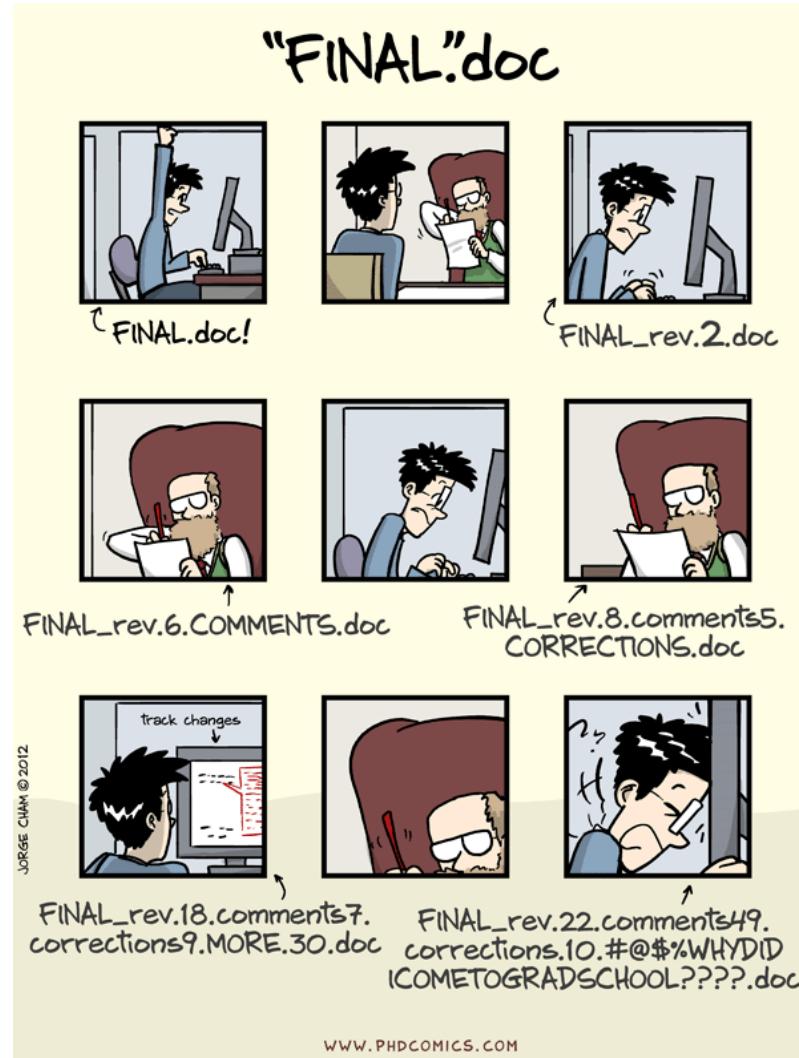
research-computing@st-andrews.ac.uk

Software Carpentry

Much of this presentation will draw on the Software Carpentry lesson [Version Control with Git](#).

That lesson goes into topics in more detail, with examples and exercises. There are regular Software Carpentry workshops at the University, open to all researchers.

Why use Version Control?



Version control systems record the details of changes made to a base version of a document or documents.

As well as allowing you to track changes over time - and to move back and forward between versions - they allow collaborators to maintain differing versions and provide mechanisms for resolving those differences.

You get to decide what changes get grouped together in a *commit*, marking a new version.

A project's commit history and metadata make up a *repository*. Repositories can be kept in sync between different computers.

Why use Git?



git

Version control systems have been around since the 1980s - you may have heard of e.g. CVS or Subversion. Modern systems like **Git** and Mercurial are *distributed*, so they don't need a central server to host repositories.

Git has become the de facto standard.

Why use GitHub?



GitHub

[GitHub.com](https://github.com) has become the most popular platform for hosting Git Repositories - especially public repositories for open-source software.

Others platforms include [BitBucket](https://bitbucket.org) and [GitLab.com](https://gitlab.com).

The University has an instance of **GitLab** (VPN) available to researchers - this is not available to users outside the University.



GitLab

GitHub also provides some additional features which are particularly useful to academic researchers.

Using Git



<https://xkcd.com/1597/>

There are a number of ways to work with Git on your computer:

- The command-line interface referenced in the XKCD cartoon
- Terminal based user interfaces like [gitui](#) and [lazygit](#)
- Graphical user interfaces like [GitHub Desktop](#) and [SourceTree](#)
- As a feature (or plugin) of development tools and editors like RStudio.

Other user interfaces can be considered as wrappers around the command-line interface. They tend to hide some of the details of how Git works.

This presentation includes screenshots from GitHub Desktop alongside the equivalent commands.

A Git Repository

-  my-project
 -  .git
 -  data
 -  src
 -  test
 -  .gitignore
 -  LICENSE.txt
 -  README.md
 -  run.sh

-  my-project
 -  .git
 -  data
 -  src
 -  test
 -  .gitignore
 -  LICENSE.txt
 -  README.md
 -  run.sh

.git is where Git stores the metadata about the project.

We (almost) never edit its contents directly.

Technically, .git is the *Git Repository*, but you will often see **my-project** described as such.

-  my-project
 -  .git
 -  data
 -  src
 -  test
 -  .gitignore
 -  LICENSE.txt
 -  README.md
 -  run.sh

Many Git-managed projects will have a **.gitignore** file, which allows us to exclude some files from version control.

Initialising

```
$ cd /Users/paddy/Documents/GitHub  
$ mkdir my-project  
$ cd my-project  
$ git init
```

Create a New Repository X

Name

Description

Local Path Choose...

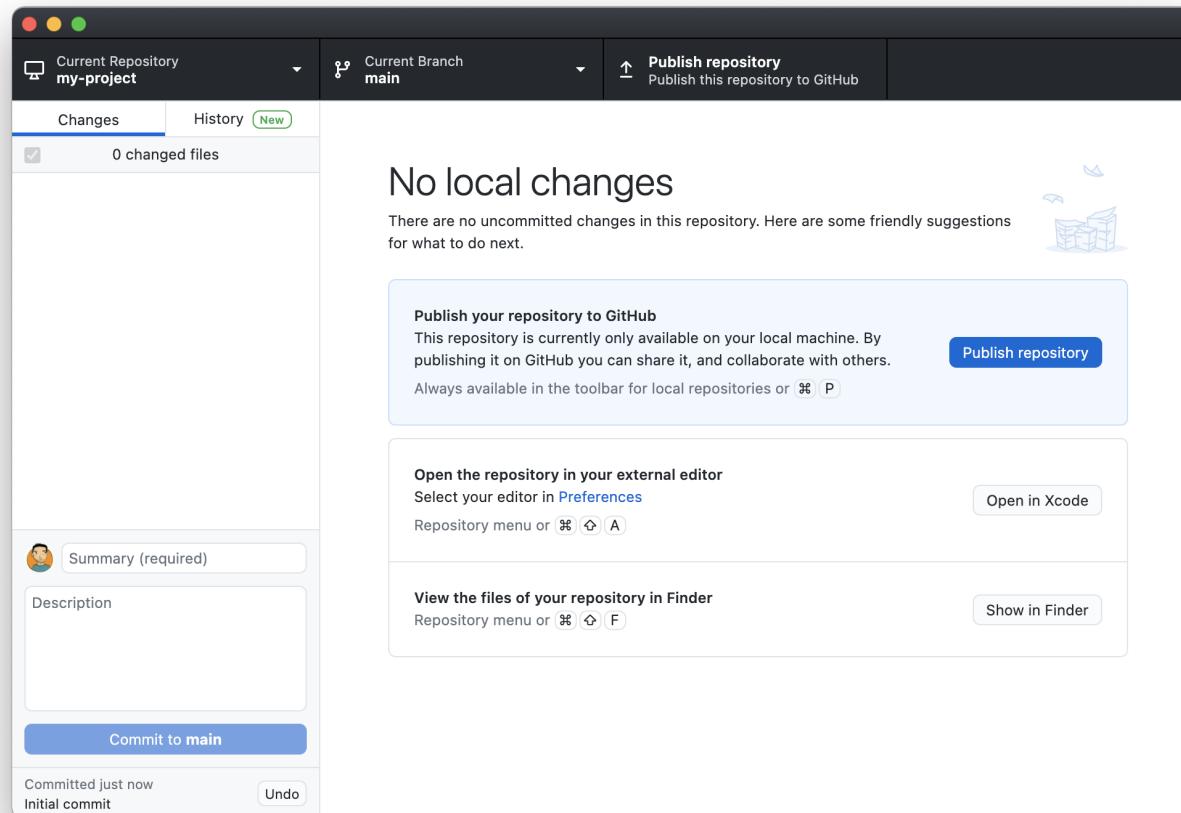
Initialize this repository with a README

Git Ignore

License

Cancel Create Repository

```
$ ls -a
.
..
.git
$ git status
On branch main
nothing to commit, working tree clean
```



Adding and Committing

We can add a file to our project folder using any text editor or, indeed, any piece of software which allows us to save files.

Here, we've used a text editor to create a README file using markdown syntax and save it as README . md.

```
$ ls -a
.          ..        .git      README.md
$ cat README.md
# My Project
```

This is an example project to illustrate the use of Git for collaboration in a research context.

Current Repository **my-project**

Current Branch **main**

Publish repository
Publish this repository to GitHub

Changes 1 History New

1 changed file

README.md

@@ -0,0 +1,4 @@

		1	+# My Project
		2	+
		3	+This is an example project to illustrate the use of Git for
		4	+collaboration in a research context.

Create README.md

Description

Commit to main

Committed 3 days ago

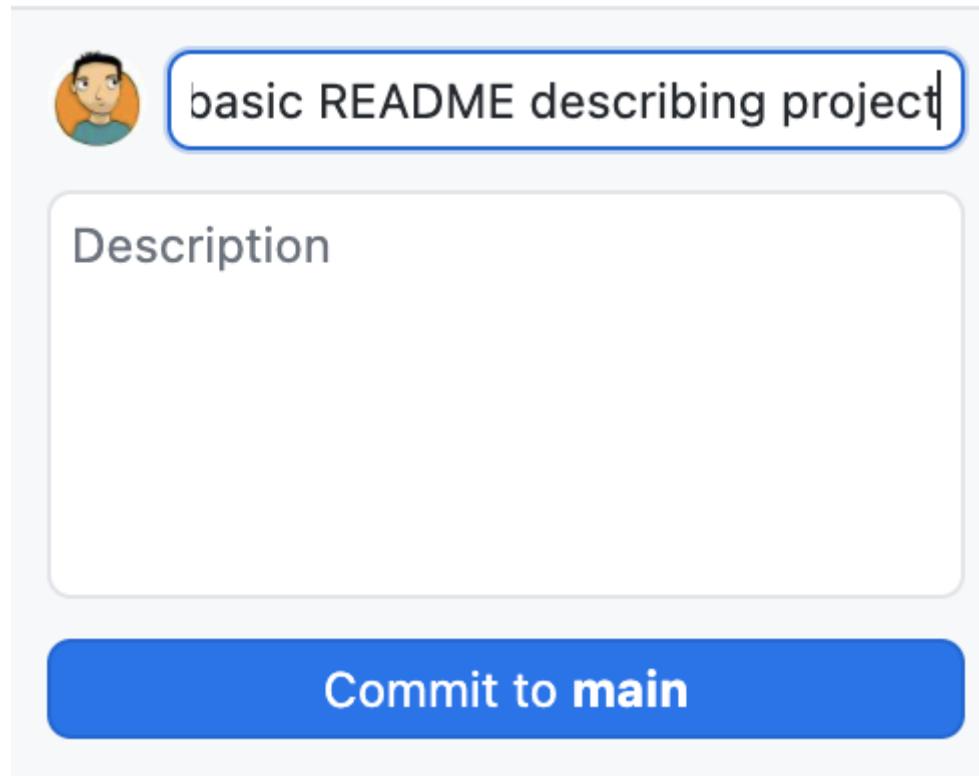
Initial commit

Undo

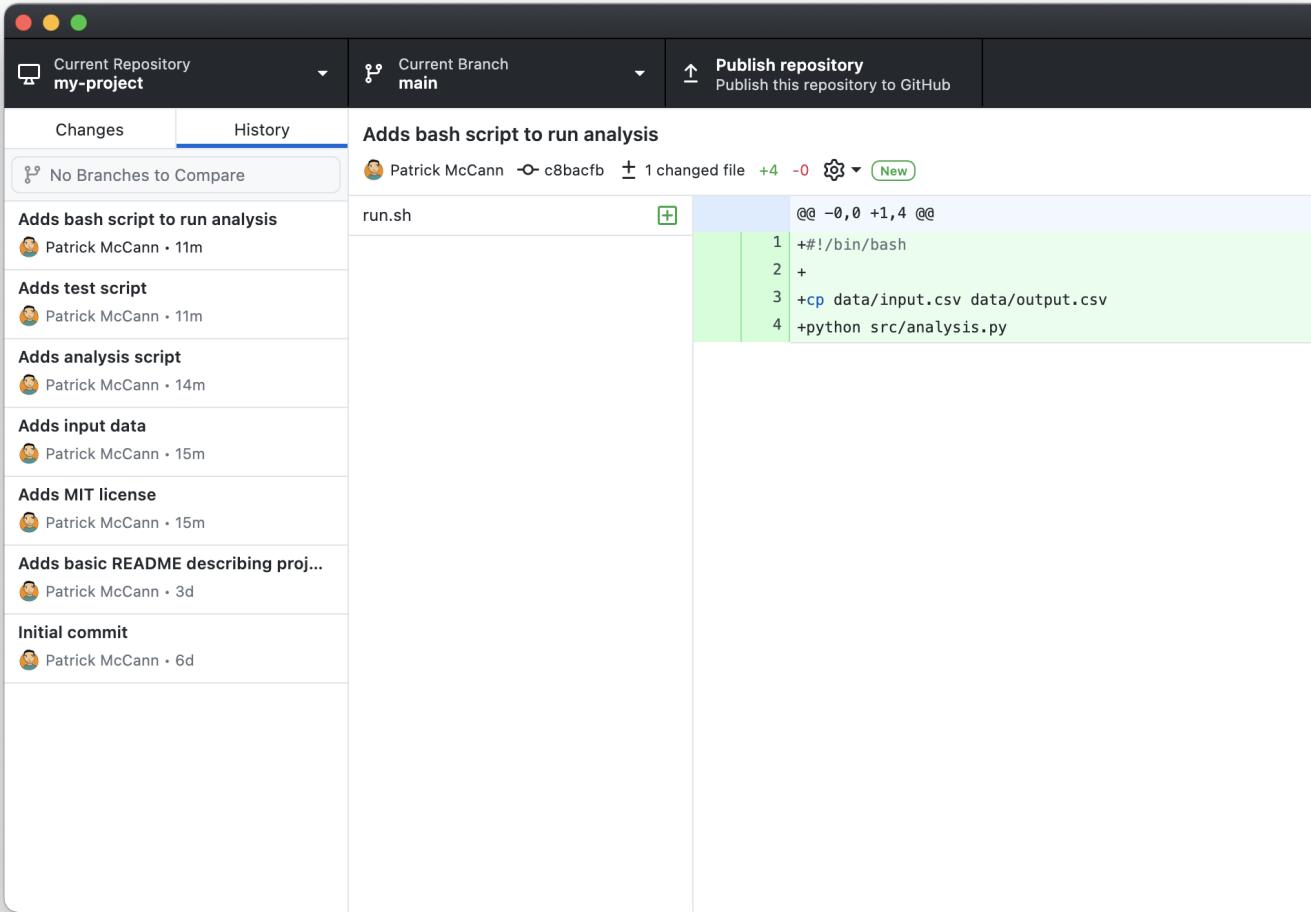
```
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add
$ git add README.md
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   README.md
```

```
$ git commit -m 'Adds basic README describing project'
```



Viewing the Log



The screenshot shows a GitHub desktop application window. At the top, there are three dropdown menus: 'Current Repository' set to 'my-project', 'Current Branch' set to 'main', and a publish button. Below the header, there are two tabs: 'Changes' and 'History', with 'History' being the active tab. The main area displays a list of commits:

- Adds bash script to run analysis (Patrick McCann, 11m ago)
- Adds test script (Patrick McCann, 11m ago)
- Adds analysis script (Patrick McCann, 14m ago)
- Adds input data (Patrick McCann, 15m ago)
- Adds MIT license (Patrick McCann, 15m ago)
- Adds basic README describing proj... (Patrick McCann, 3d ago)
- Initial commit (Patrick McCann, 6d ago)

On the right side, the details for the most recent commit ('Adds bash script to run analysis') are expanded. It shows the commit message, author, date, and a diff viewer. The diff viewer shows the following changes:

```
@@ -0,0 +1,4 @@
1 +#!/bin/bash
2 +
3 +cp data/input.csv data/output.csv
4 +python src/analysis.py
```

```
$ git log
```

```
commit c8bacfbfd15e5f0924dde8461662ba8fdb676da
Author: Patrick McCann <pgm5@st-andrews.ac.uk>
Date:   Mon Nov 1 15:36:13 2021 +0000
```

```
    Adds bash script to run analysis
```

```
commit 3fa305a9d4e76e9a4bb58f724a15c2f5ae032edc
Author: Patrick McCann <pgm5@st-andrews.ac.uk>
Date:   Mon Nov 1 15:35:39 2021 +0000
```

```
    Adds test script
```

```
commit 382abfc258f398954ad897e52ff224a75188a7ca
Author: Patrick McCann <pgm5@st-andrews.ac.uk>
Date:   Mon Nov 1 15:32:32 2021 +0000
```

```
$ git diff HEAD~1 HEAD
diff --git a/run.sh b/run.sh
new file mode 100755
index 000000..a84485c
--- /dev/null
+++ b/run.sh
@@ -0,0 +1,4 @@
+#!/bin/bash
+
+cp data/input.csv data/output.csv
+python src/analysis.py
```

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL.	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

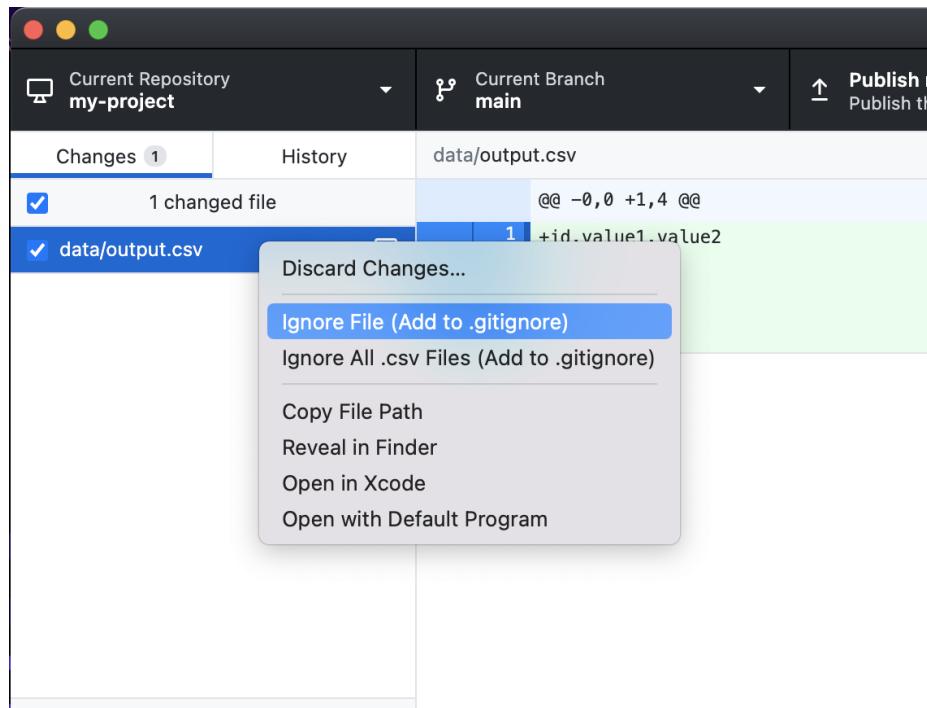
AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

<https://xkcd.com/1296/>

Ignoring Things

There are situations where we have a file or files in our repository which we don't want to place under version control e.g.

- The results of analysis.
- Anything containing a password or other sensitive data.
- Files created by your tools which aren't really part of the project.



```
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    data/output.csv

nothing added to commit but untracked files present (use "git add
$ git ignore data/output.csv
Adding pattern(s) to: .gitignore
... adding 'data/output.csv'
```

Current Repository **my-project**

Current Branch **main**

Publish

Changes 1 History .gitignore

1 changed file

.gitignore **+data/output.csv**

This screenshot shows a Git interface with three main sections: 'Current Repository' (my-project), 'Current Branch' (main), and 'Publish'. Below these are tabs for 'Changes' (1) and 'History'. The 'Changes' tab is active, showing a single change in '.gitignore'. A blue bar highlights the file name, and a green bar highlights the addition of 'data/output.csv'. The commit message shows a diff: '@@ -0,0 +1 @@'.

```
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add
```

Remotes

So far, everything has been on our own computer. This is useful, but Git really comes into its own when collaborating with others.

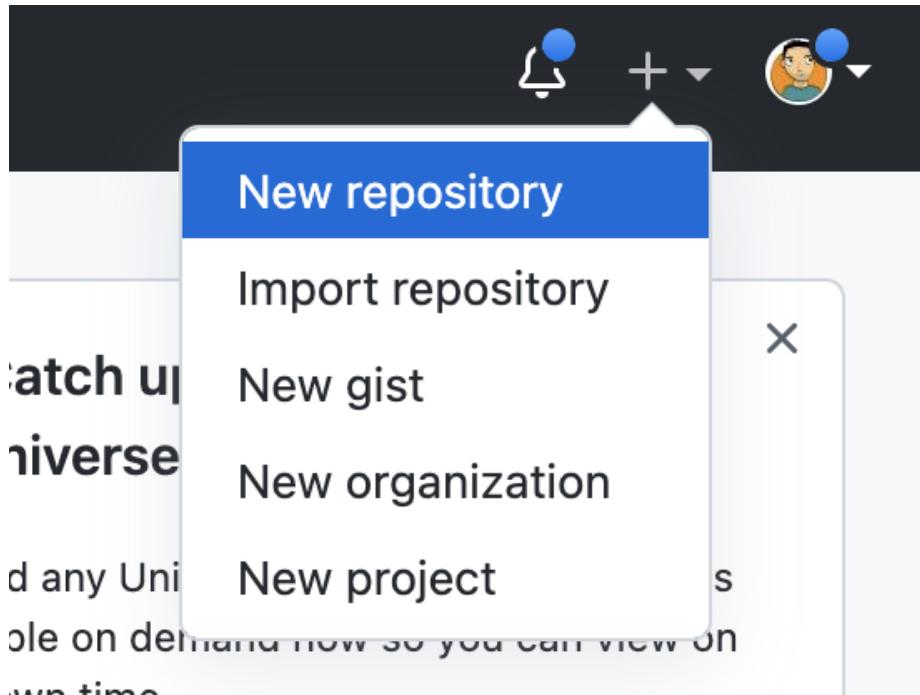
To do that, we need to keep our projects in sync between multiple computers (also useful if you work from multiple computers yourself).

Your computer is *local*, one with which you are syncing is a *remote*.

You can, in principle, have Git on your computer communicate directly with your colleagues' to keep things in sync, but it's generally easier to sync with another location to which you all have access.

That other location could be a desktop computer in your office or it could be GitHub (or GitLab) - as far as Git is concerned there isn't really a difference.

Creating a repo on GitHub.com



Create a new repository

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

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



pgmccann

/ Repository name *



my-project



Great repository names are short and memorable. Need inspiration? How about [cautious-pancake](#)?

Description (optional)

Demonstration project for talk on Collaboration using Git

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

 Set up in Desktop or [HTTPS](#) [SSH](#) git@github.com:pgmccann/my-project.git 

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 "# my-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:pgmccann/my-project.git
git push -u origin main
```



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

```
git remote add origin git@github.com:pgmccann/my-project.git
git branch -M main
git push -u origin main
```

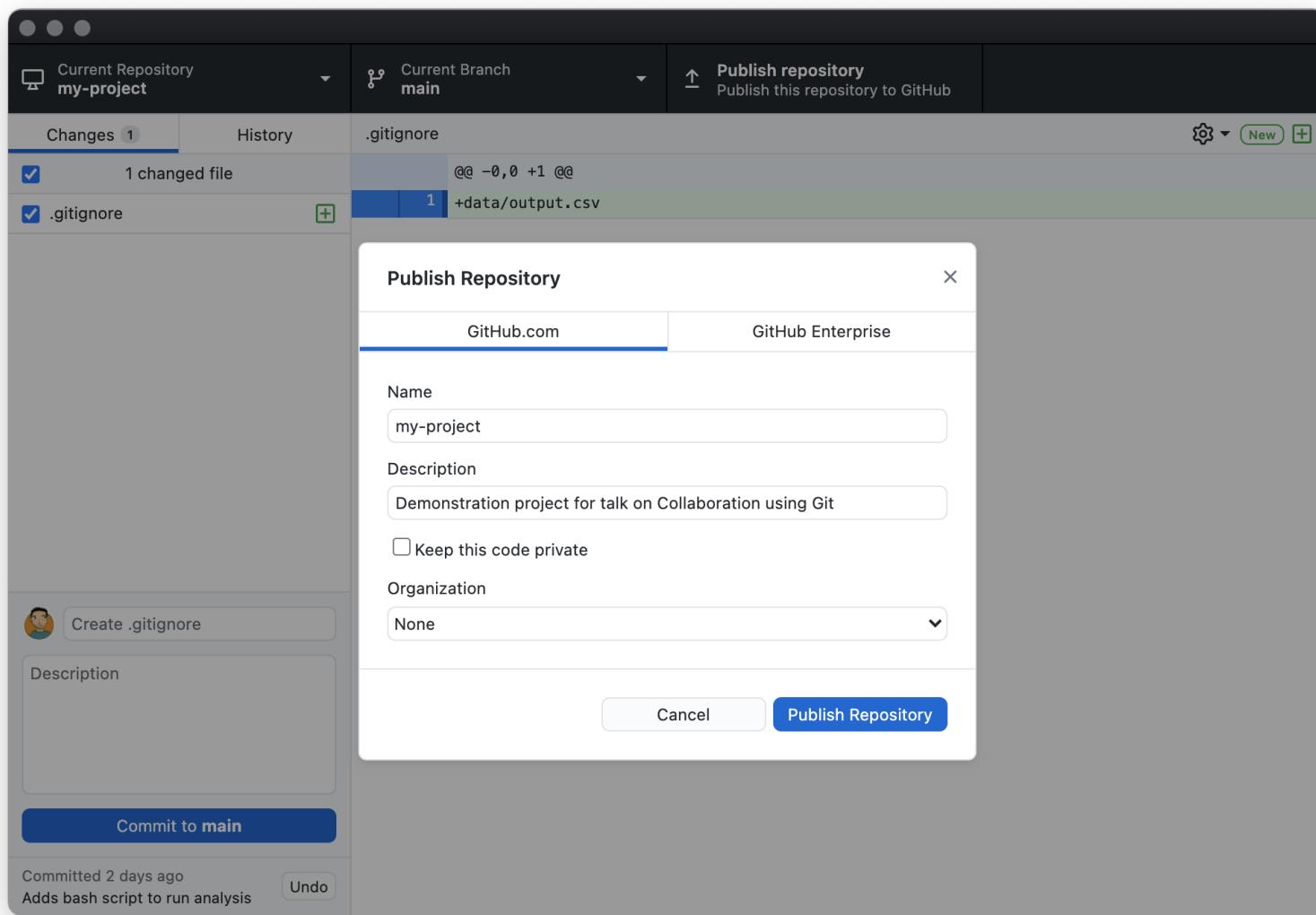


...or import code from another repository

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

[Import code](#)

Publishing a repo from GitHub Desktop



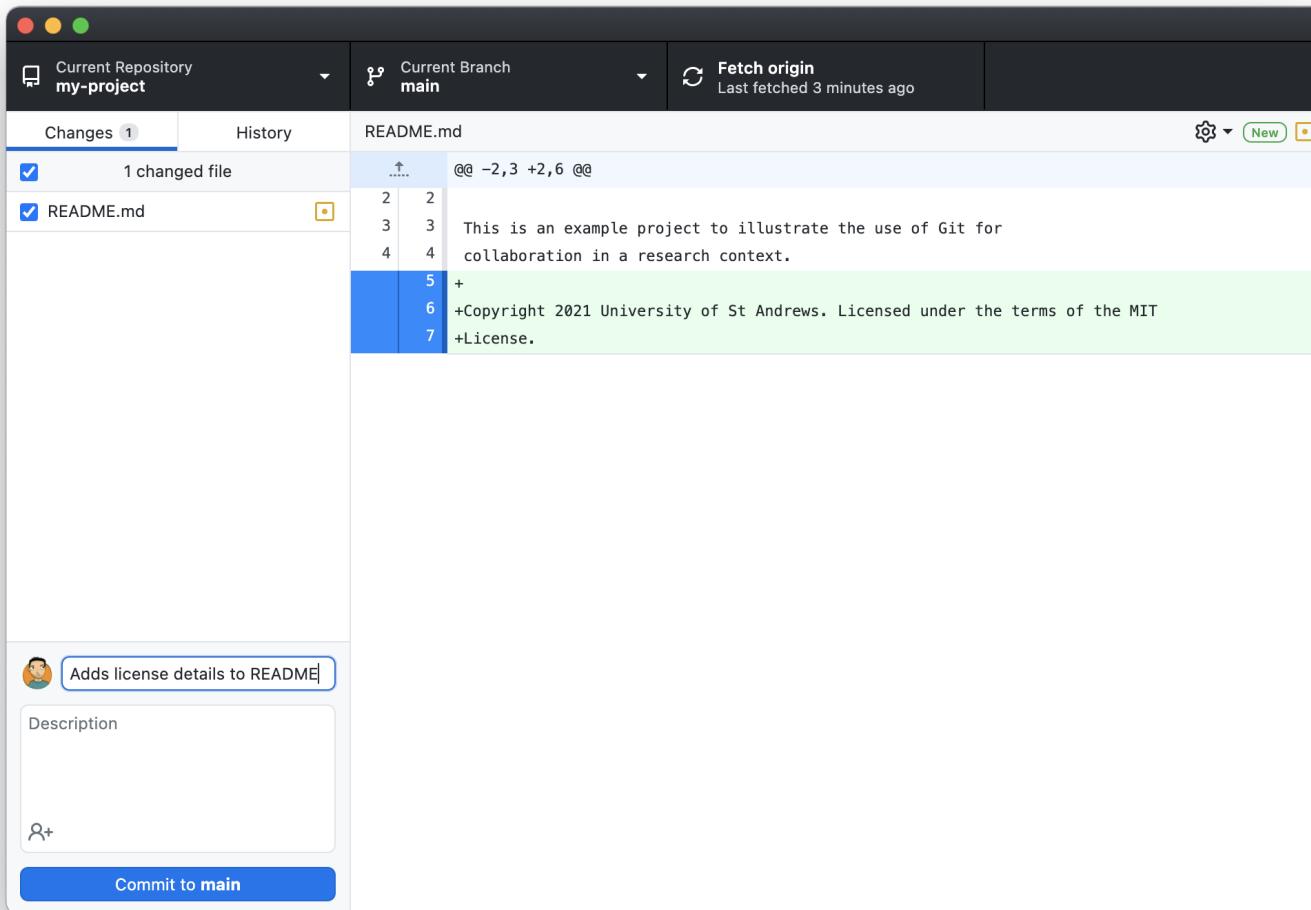
The repo on GitHub

The screenshot shows a GitHub repository page for a project named 'my-project'. The repository is public and owned by 'pgmccann'. The main navigation bar includes links for Pulls, Issues, Marketplace, Explore, and a user profile icon. Below the header, there are buttons for Unwatch (1), Star (0), and Fork (0). The repository menu includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and a three-dot menu. A dropdown menu for the 'main' branch is open. On the right side, there's an 'About' section with a description of the project as a demonstration for a talk on collaboration using Git, along with links to Readme and MIT License. There are sections for Releases (no releases published) and Packages (no packages published). The main content area displays a list of commits:

File / Commit Type	Description	Date
pgmccann	Adds bash script to run analysis	2 days ago
data	Adds input data	2 days ago
src	Adds analysis script	2 days ago
test	Adds test script	2 days ago
.gitattributes	Initial commit	8 days ago
LICENSE.txt	Adds MIT license	2 days ago
README.md	Adds basic README describing project	5 days ago
run.sh	Adds bash script to run analysis	2 days ago

Below the commits, there's a file editor for 'README.md' containing the text "My Project". A note at the bottom states: "This is an example project to illustrate the use of Git for collaboration in a research context."

Pushing



```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working dire
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
$ git diff
diff --git a/README.md b/README.md
index e0a3173..b9fd878 100644
--- a/README.md
+++ b/README.md
@@ -2 +2 @@
```

After committing...

```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 429 bytes | 429.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object
To https://github.com/pgmccann/my-project.git
  2bf80ab..3ec775a  main > main
```

 README.md	Adds license details to README	34 minutes ago
 run.sh	Adds bash script to run analysis	2 days ago

README.md 

My Project

This is an example project to illustrate the use of Git for collaboration in a research context.

Copyright 2021 University of St Andrews. Licensed under the terms of the MIT License.

Pulling

Let's add a License **badge** using GitHub's editing interface.

License MIT

[Edit file](#)[Preview](#)

Spaces

2

Soft wrap

```
1 # My Project
2
3 This is an example project to illustrate the use of Git for
4 collaboration in a research context.
5
6 Copyright 2021 University of St Andrews. Licensed under the terms of the MIT
7 License.
8
```

<> Edit file

Preview

Spaces

2

Soft wrap

```
1  [![License: MIT](https://img.shields.io/badge/License-MIT-yellow.svg)](https://opensource.org/licenses/MIT)
2
3 # My Project
4
5 This is an example project to illustrate the use of Git for
6 collaboration in a research context.
7
8 Copyright 2021 University of St Andrews. Licensed under the terms of the MIT
9 License.
10
```

Commit changes

Adds License badge to README

Add an optional extended description...



pgm5@st-andrews.ac.uk



Choose which email address to associate with this commit

- o- Commit directly to the `main` branch.
- ↗ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel

 README.md	Adds License badge to README	now
 run.sh	Adds bash script to run analysis	2 days ago

README.md



License MIT

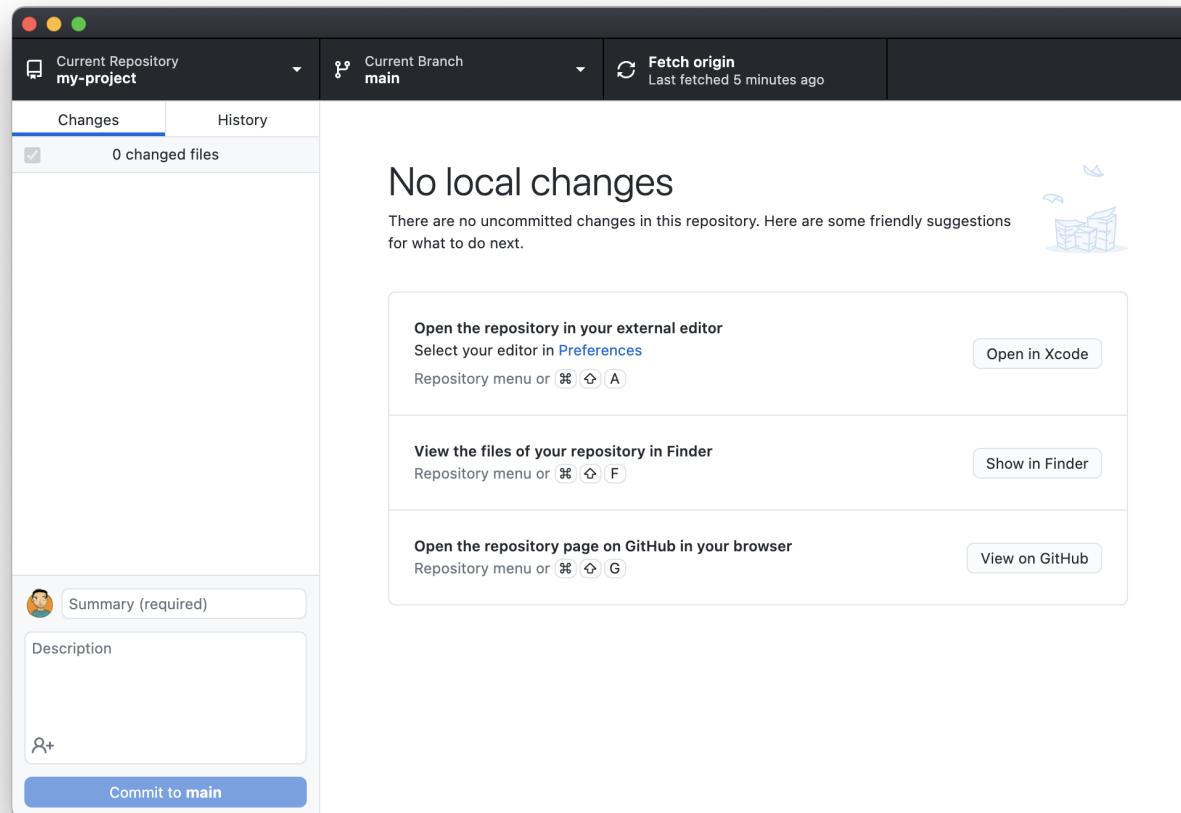
My Project

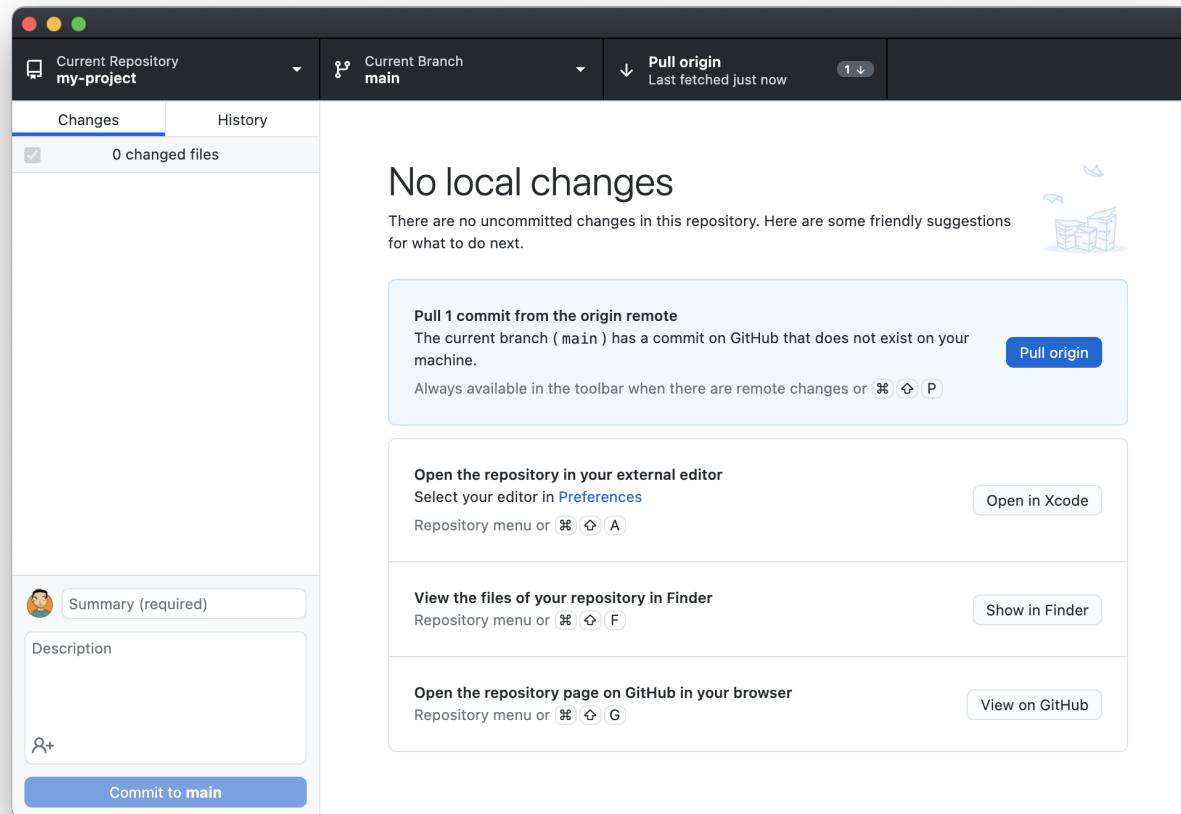
This is an example project to illustrate the use of Git for collaboration in a research context.

Copyright 2021 University of St Andrews. Licensed under the terms of the MIT License.

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```





```
$ git pull
Updating 3ec775a..658258a
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```

Current Repository my-project Current Branch main Fetch origin Last fetched 8 minutes ago

Changes History Adds License badge to README

No Branches to Compare

Adds License badge to README Paddy McCann 658258a 1 changed file +2 -0 New

README.md

		@@ -1,3 +1,5 @@
	1	+![License: MIT](https://img.shields.io/badge/License-MIT-yellow.svg)](https://opensource.org/licenses/MIT)
	2	+
	3	# My Project
	4	
	5	This is an example project to illustrate the use of Git for
	...	
		↓

Adds license details to README Patrick McCann 1h

Ignores the output data Patrick McCann 1h

Adds bash script to run analysis Patrick McCann 1d

Adds test script Patrick McCann 1d

Adds analysis script Patrick McCann 1d

Adds input data Patrick McCann 1d

Adds MIT license Patrick McCann 1d

Adds basic README describing proj... Patrick McCann 4d

Initial commit Patrick McCann Oct 26, 2021

Managing access on GitHub

My project is in a public GitHub repository, so anyone can see it - but only I can make commits, either by editing on the GitHub website, or pushing from my computer.

But I can add collaborators.

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Who has access

PUBLIC REPOSITORY



This repository is public and visible to anyone.

[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](#)



Add a collaborator to my-project

Search by username, full name, or email

Select a collaborator above

VISIBLE TO ANYONE.

Manage

ss to

this repository. Only you can contribute to this repository.

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Manage access



You haven't invited any collaborators yet

Add people

< > Code

Issues

Pull requests

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets



Add a collaborator to my-project



Swithun Crowe
swithun



Add swithun to this repository

Manage access



You haven't invited any collaborators yet

Add people

Insights

Settings

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Who has access

PUBLIC REPOSITORY



This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS



1 has access to this repository. [1 invitation](#).

Manage access

 [Select all](#) [Find a collaborator...](#)[Swithun Crowe](#)

Awaiting swithun's response

Pending Invite [< Previous](#) [Next >](#)

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Who has access

PUBLIC REPOSITORY



This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS



1 has access to this repository. [1 collaborator.](#)

Manage access

 [Select all](#) [Find a collaborator...](#)[Swithun Crowe](#)

swithun • Collaborator

[< Previous](#) [Next >](#)

Merging

Knowing how to push to and pull from a remote is enough to enable collaboration - later we'll come on to some techniques which can make things easier.

We can run into complications if collaborators are working on a project at the same time, requiring commits to be *merged*.

Git does as much as possible to make this painless.

It will handle merges automatically if the commits to be merged involve changes to different files, and even if the changes are in different parts of the same file.

If the changes are to the same part of a file, the merge will need to be performed manually.

Tweaks license statement in README

[Browse files](#)

Removes repetition of 'license'.

main

 pgmccann committed 14 seconds ago Verified 1 parent 658258a commit 46693932563a5786bbe322ba8a9b4d46d3ca5b20

 Showing 1 changed file with 1 addition and 1 deletion.

[Split](#)[Unified](#)

▼ ▲ 2  README.md 

  ...

... @@ -5,5 +5,5 @@

5 5 This is an example project to illustrate the use of Git for
6 6 collaboration in a research context.

7 7
8 - Copyright 2021 University of St Andrews. Licensed under the terms of the MIT
8 + Copyright 2021 University of St Andrews. Made available under the terms of the MIT
9 9 License.

Current Repository my-project Current Branch main Push origin Last fetched 17 minutes ... 1 ↑

Changes History Makes copyright statement in README more concise

No Branches to Compare

Makes copyright statement i... Patrick McCann • just now ↗

Adds License badge to README Paddy McCann • 1h

Adds license details to README Patrick McCann • 2h

Ignores the output data Patrick McCann • 3h

Adds bash script to run analysis Patrick McCann • 1d

Adds test script Patrick McCann • 1d

Adds analysis script Patrick McCann • 1d

Adds input data Patrick McCann • 1d

Adds MIT license Patrick McCann • 1d

Adds basic README describing proj... Patrick McCann • 4d

Initial commit

README.md @@ -5,5 +5,4 @@

	5	5	This is an example project to illustrate the use of Git f or
	6	6	collaboration in a research context.
	7	7	
	8		-Copyright 2021 University of St Andrews. Licensed under t he terms of the MIT
	9		-License.
		8	+© 2021 University of St Andrews. Licensed under the terms of the MIT License.

Current Repository
my-project

Current Branch
main

Push origin
Last fetched 28 minutes ... 1 ↑

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Newer Commits on Remote

 Desktop is unable to push commits to this branch because there are commits on the remote that are not present on your local branch. Fetch these new commits before pushing in order to reconcile them with your local commits.

Cancel Fetch Open in Xcode

Summary (required)

Description

+ Commit to main

Committed 11 minutes ago Undo

Makes copyright statement in RE...

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F Show in Finder

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G View on GitHub

Push origin

Current Repository
my-project

Current Branch
main

Pull origin
Last fetched just now

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 1 commit from the origin remote
The current branch (main) has a commit on GitHub that does not exist on your machine.

Pull origin

Always available in the toolbar when there are remote changes or ⌘ ⌄ P

Open the repository in your external editor
Select your editor in [Preferences](#)

Repository menu or ⌘ ⌄ A

Show in Finder

View the files of your repository in Finder
Repository menu or ⌘ ⌄ F

View on GitHub

Commit to main

Committed 12 minutes ago

Makes copyright statement in RE...

Undo

Current Repository my-project

Current Branch main

Pull origin Last fetched just now

Changes 1 History README.md

1 changed file README.md

@@ -5,4 +5,9 @@

5	5	This is an example project to illustrate the use of Git for
6	6	collaboration in a research context.
7	7	
8	8	+<<<< HEAD
8	9	© 2021 University of St Andrews. Licensed under the terms of the MIT License.

Resolve conflicts before Merge

1 conflicted file README.md 1 conflict

Open in command line, your tool of choice, or close to resolve manually.

Abort Merge Continue Merge

Update README.md

Description

Commit to main

Committed 12 minutes ago Undo Makes copyright statement in RE...

Current Repository my-project Current Branch main Pull origin Last fetched just now

⚠ Resolve conflicts to continue merge into **main**. [View conflicts](#)

Changes 1 History README.md

1 changed file README.md

5 5 @@ -5,4 +5,9 @@
6 6 This is an example project to illustrate the use of Git for
7 7 collaboration in a research context.
8 +<<<<< HEAD
8 9 © 2021 University of St Andrews. Licensed under the terms of the MIT License.
10 +=====+
11 +Copyright 2021 University of St Andrews. Made available under the terms of the MIT
12 +License.
13 +>>>>> 46693932563a5786bbe322ba8a9b4d46d3ca5b20

Update README.md

Description

+
Commit to main

Committed 12 minutes ago Undo Makes copyright statement in RE...

Current Repository my-project Current Branch main Pull origin Last fetched 3 min...

⚠ Resolve conflicts to continue merge into main. [View conflicts](#)

Changes 1	History	README.md
<input checked="" type="checkbox"/> 1 changed file	 @@ -5,4 +5,5 @@ 5 5 This is an example project to illustrate the use of Git for 6 6 collaboration in a research context. 7 7 8 -© 2021 University of St Andrews. Licensed under the terms of the MIT License. 8 +© 2021 University of St Andrews. Made available under the terms of the MIT 9 +License.
<input checked="" type="checkbox"/> README.md		

Updates license statement in RE/
Combines changes made in previous, conflicting commits.
+
Commit to main

Committed 15 minutes ago **Undo** Makes copyright statement in RE...

Current Repository **my-project**

Current Branch **main**

Push origin
Last fetched 4 minutes ago

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Push commits to the origin remote
You have 2 local commits waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or `⌘P`

Push origin

Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or `⌘↑A`

Open in Xcode

View the files of your repository in Finder
Repository menu or `⌘↑F`

Show in Finder

Open the repository page on GitHub in your browser
Repository menu or `⌘↑G`

View on GitHub

Summary (required)

Description

+ Commit to main

Committed just now
Updates license statement in RE...

Undo

Current Repository
my-project

Current Branch
main

Fetch origin
Last fetched just now

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or ⌘ ⌘ A

Show in Finder

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F

View on GitHub

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

Summary (required)

Description

+ Commit to main

 README.md	Updates license statement in README	23 seconds ago
 run.sh	Adds bash script to run analysis	2 days ago

README.md



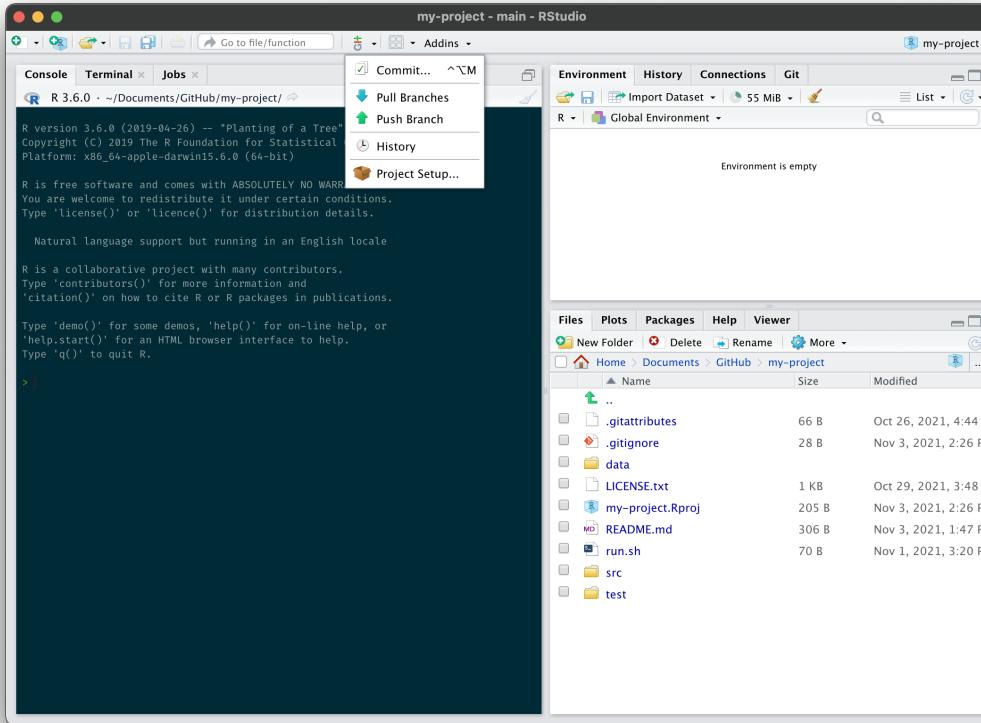
License MIT

My Project

This is an example project to illustrate the use of Git for collaboration in a research context.

© 2021 University of St Andrews. Made available under the terms of the MIT License.

Git in RStudio



<https://swcarpentry.github.io/git-novice/14-supplemental-rstudio/index.html>

Branching

Branches in Git allow different versions of a project to exist in parallel, with the repository keeping track of all of them.

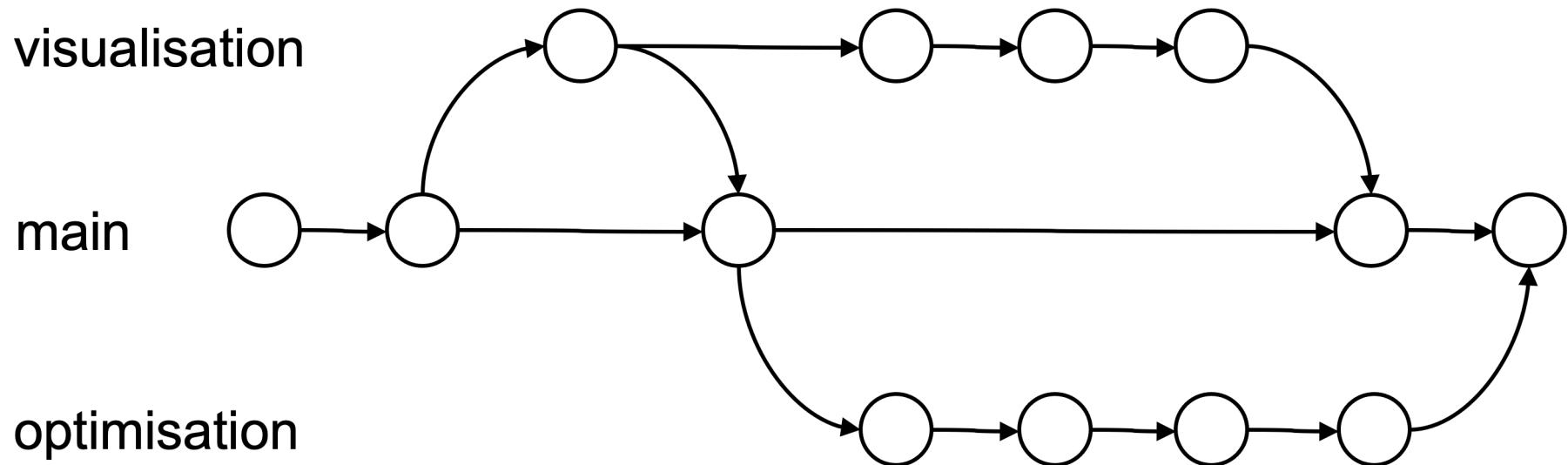
Commits can be made to a branch irrespective of what's happening on any others.

Branches can be merged, with any conflicts resolved in the same way we've already seen.

There are a number of reasons you might use branches,
including:

- To try something out without changing the content of the main branch
- To work on different things - whether it's one person switching between them or collaborators working in parallel.

Imagine two researchers are collaborating on a piece of data analysis software. One is working on new visualisations, and the other is working on optimising the analysis.



Some say you should never work directly on the main branch.

Current Repository
my-project

Current Branch
main

Fetch origin
Last fetched 7 minutes ago

Changes History Branches Pull Requests

Filter New Branch

Default Branch

✓ main 3 hours ago

Here are some friendly suggestions

Open in Xcode

Show in Finder

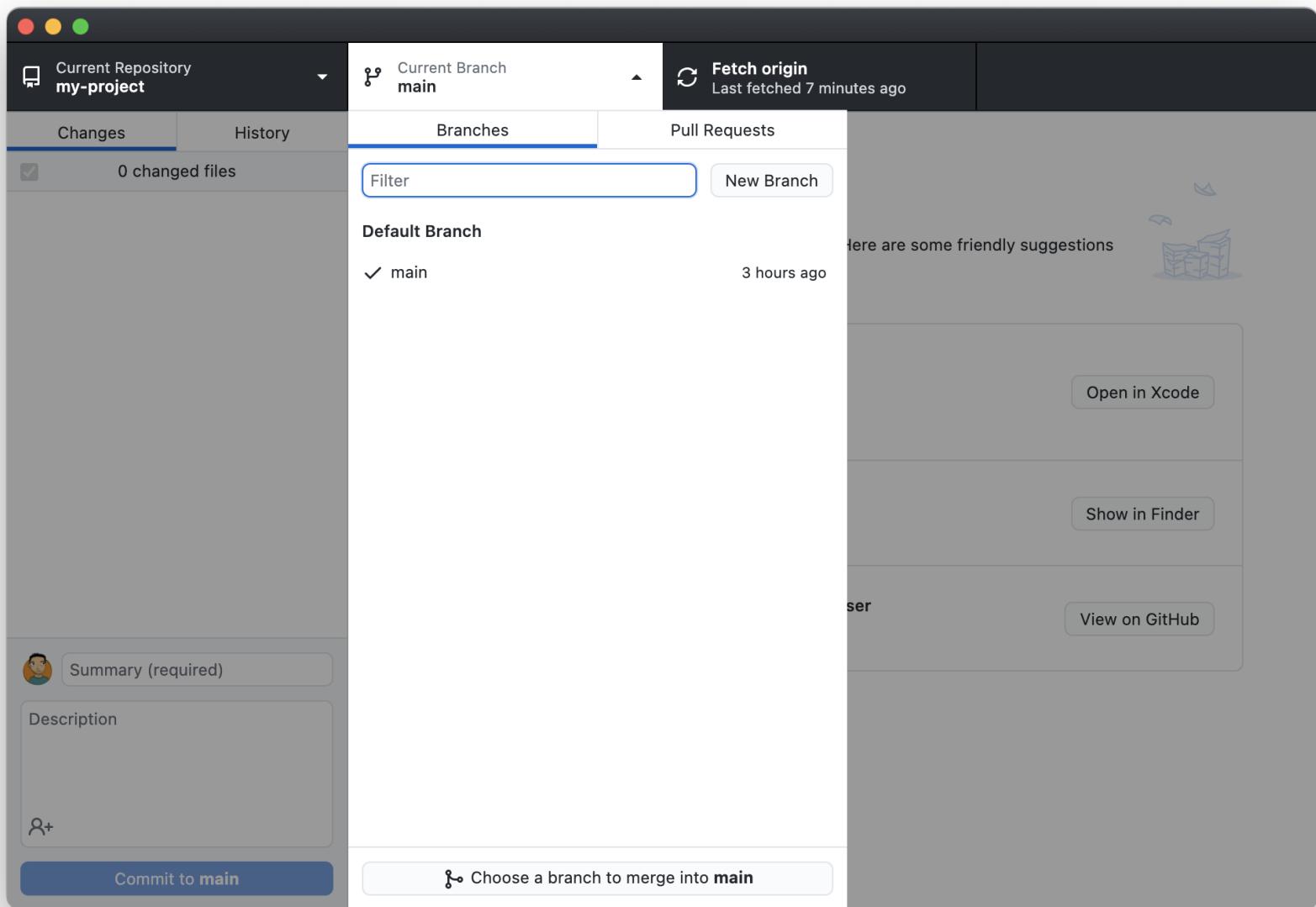
View on GitHub

Summary (required)

Description

+ Commit to main

Choose a branch to merge into **main**

A screenshot of a GitHub desktop application window. The top bar shows the current repository as "my-project". The "Branches" tab is selected. A pull request is open, targeting the "main" branch. The PR summary is required, and the description field is empty. The "Commit to main" button is visible at the bottom left. On the right side, there's a sidebar with options to "Open in Xcode", "Show in Finder", and "View on GitHub". A message suggests merging into the "main" branch. The GitHub logo is visible in the top right corner.

Current Repository
my-project

Current Branch
main

Fetch origin
Last fetched 14 minutes ago

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Create a Branch

Name

Your new branch will be based on your currently checked out branch (**main**). **main** is the [default branch](#) for your repository.

Cancel Create Branch

Open in Xcode

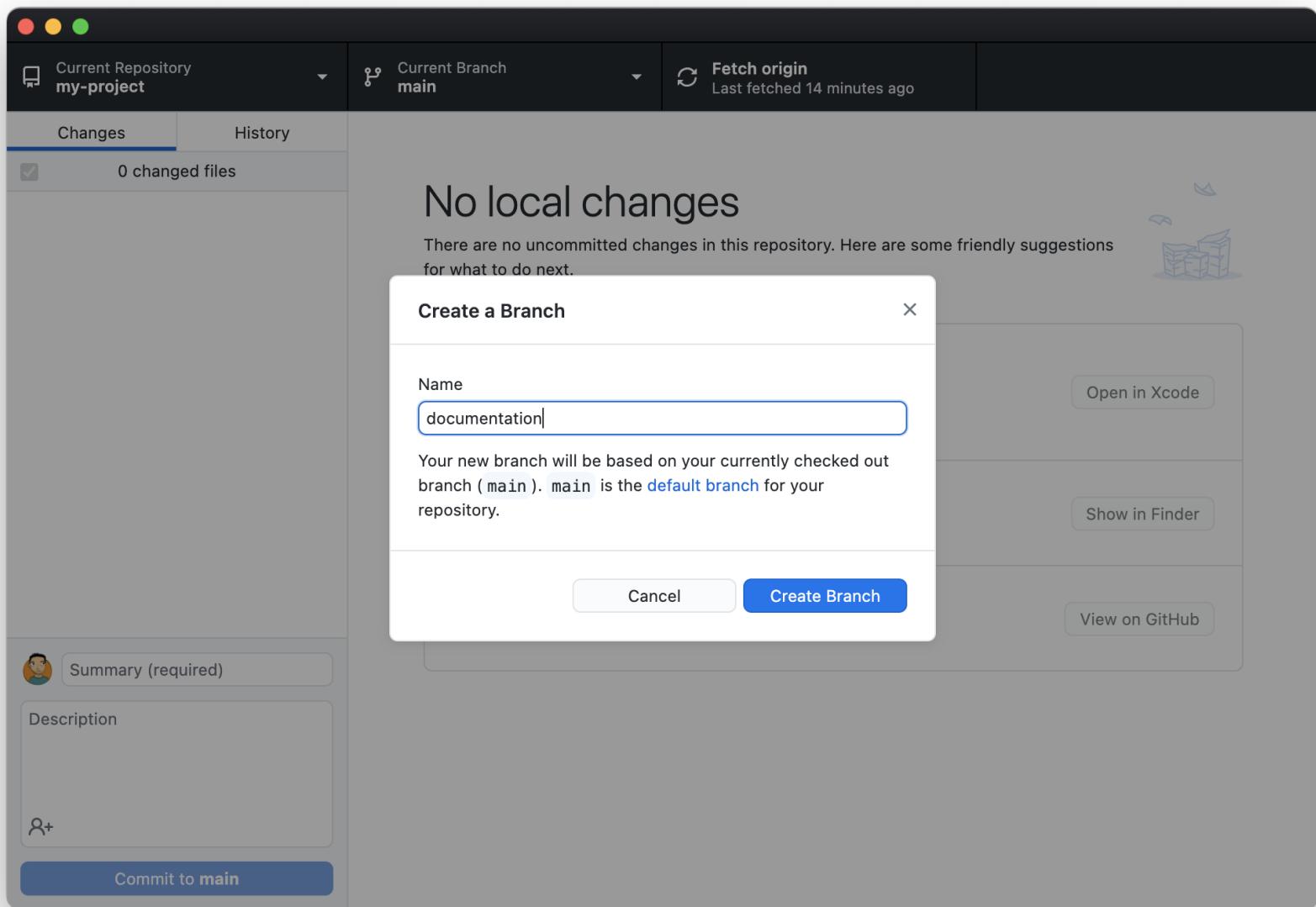
Show in Finder

View on GitHub

Summary (required)

Description

+ Commit to main

A screenshot of a GitHub desktop application window. The main interface shows a repository named 'my-project' with the current branch set to 'main'. A modal dialog box is open in the center, titled 'Create a Branch'. Inside the modal, there is a text input field labeled 'Name' containing the text 'documentation'. Below the input field, a descriptive message states: 'Your new branch will be based on your currently checked out branch (main). main is the default branch for your repository.' At the bottom of the modal, there are two buttons: 'Cancel' and 'Create Branch', with 'Create Branch' being highlighted in blue. In the background, the main application window shows a summary section with a user icon and the text 'Summary (required)', and a 'Commit to main' button at the bottom. To the right of the modal, there are three additional buttons: 'Open in Xcode', 'Show in Finder', and 'View on GitHub'. The overall interface has a clean, modern design with a light gray background and blue highlights for interactive elements.

Current Repository
my-project

Current Branch
documentation

Publish branch
Publish this branch to GitHub

Changes History

0 changed files

Branches Pull Requests

Filter New Branch

Default Branch

main 3 hours ago

Recent Branches

✓ documentation 3 hours ago

Here are some friendly suggestions

published to the remote
in a pull request, and

Publish branch

Open in Xcode

Show in Finder

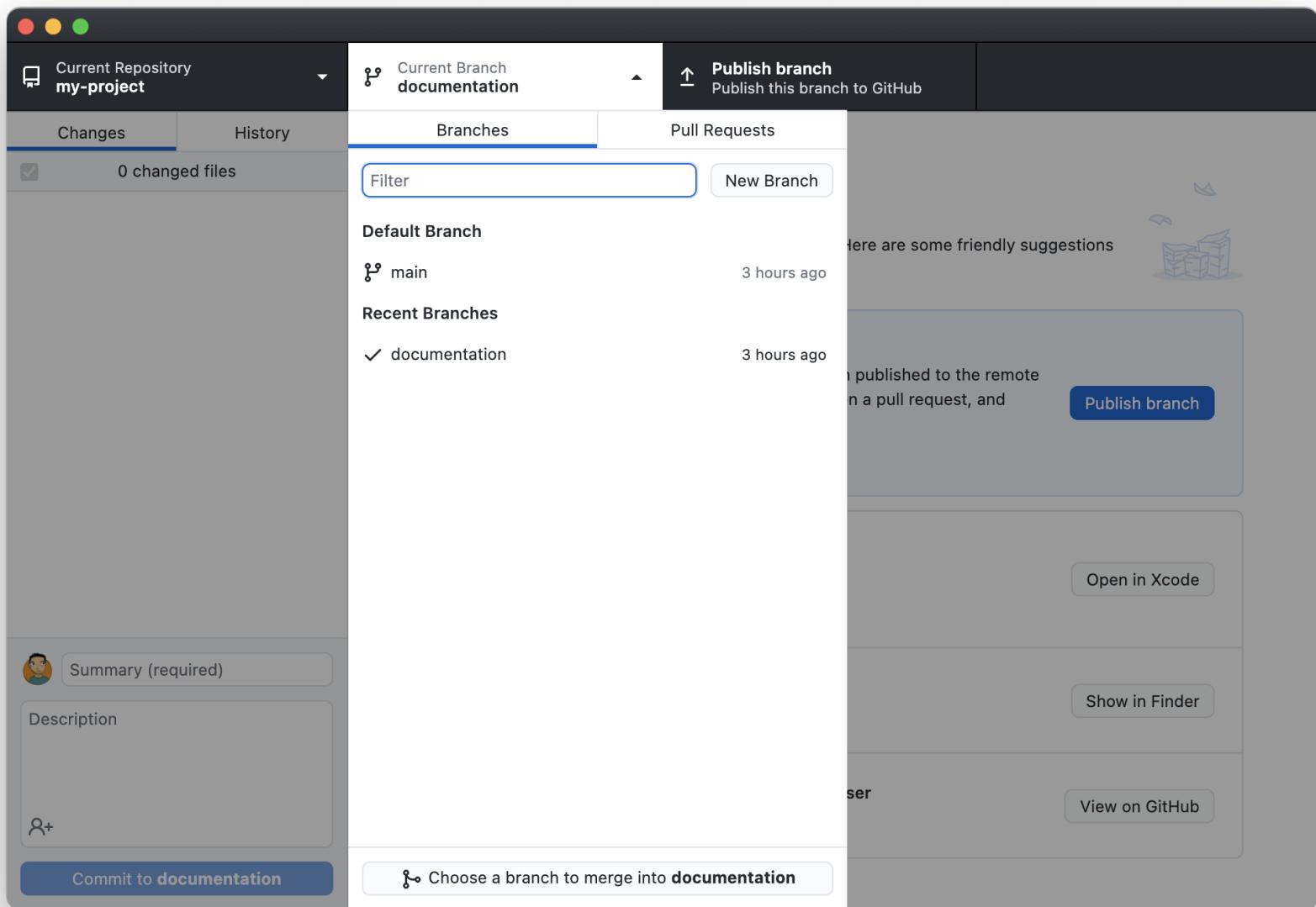
View on GitHub

Summary (required)

Description

+ Commit to documentation

Choose a branch to merge into documentation



Current Repository
my-project

Current Branch
documentation

Publish branch
Publish this branch to GitHub

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Publish your branch
The current branch (documentation) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or ⌘ P

Open in Xcode

Show in Finder

View on GitHub

Summary (required)

Description

+ Commit to documentation

Current Repository my-project

Current Branch documentation

Publish branch
Publish this branch to GitHub

Changes 2 History docs/install.txt

2 changed files

docs/install.txt

docs/running.txt

@@ -0,0 +1,12 @@

+Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula
+egest dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient
+montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque
+eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo,
+fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut,
+imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium.
+Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate
+eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac,
+enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus
+viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam
+ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui.
+Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula

Adds some dummy documentation

Description

Commit to documentation

Current Repository
my-project

Current Branch
documentation

Publish branch
Publish this branch to GitHub

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Publish your branch
The current branch (documentation) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or ⌘ P

Open in Xcode

Show in Finder

View on GitHub

Summary (required)

Description

R+

Commit to documentation

Committed just now
Adds some dummy documentation

Undo

Current Repository
my-project

Current Branch
documentation

Fetch origin
Last fetched just now

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Create a Pull Request from your current branch
The current branch (documentation) is already published to GitHub.
Create a pull request to propose and collaborate on your changes.

Branch menu or ⌘ R

Create Pull Request

Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or ⌘ ⌘ A

Open in Xcode

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F

Show in Finder

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

View on GitHub

Summary (required)

Description

+ 

Commit to **documentation**

⚠ documentation had recent pushes 1 minute ago

Compare & pull request

⚡ main ▾

Go to file

Add file ▾

Code ▾

Switch branches/tags X

Find or create a branch...

Branches

Tags

✓ main

default

documentation

[View all branches](#)

in README	...	20 hours ago	⌚ 13
data		3 days ago	
is script		3 days ago	
script		3 days ago	
it		9 days ago	

[documentation](#) ▾[Go to file](#)[Add file](#) ▾[Code](#) ▾

This branch is 1 commit ahead of main.

[Contribute](#) ▾**pgmccann** Adds some dummy documentation [...](#)

17 hours ago 14

data Adds input data 3 days ago **docs** Adds some dummy documentation 17 hours ago **src** Adds analysis script 3 days ago **test** Adds test script 3 days ago **.gitattributes** Initial commit 9 days ago

To incorporate the changes made in the new branch into the main branch, we make a *pull request*.

This point of the process can provide a good opportunity for collaborators to review changes.

It's not uncommon for projects to discourage contributors from accepting their own pull requests.

/documentation ▾

Go to file

Add file ▾

Code ▾

This branch is 1 commit ahead of main.

Contribute ▾

 pgmccann Adds some dummy docu

 data Adds input

 docs Adds some

 src Adds analysis script

3 days ago

 test Adds test script

3 days ago

 .gitattributes Initial commit

9 days ago



This branch is 1 commit ahead of main.

Open a pull request to contribute your changes upstream.

Compare

Open pull request



၆

base: main ▾

↑

compare: documentation ▾

✓ **Able to merge.** These branches can be automatically merged.

Adds some dummy documentation

Write

Preview

Leave a comment

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

Create pull request

Reviewers

No reviews

Assignees

No one—assign your

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Adds some dummy documentation #1

Open

pgmccann wants to merge 1 commit into [main](#) from [documentation](#) 

Conversation

-o Commits 1

 Checks 0

 Files changed 2



pgmccann commented now

Owner

• • •

No description provided

 Adds some dummy documentation

a98b52d

Add more commits by pushing to the **documentation** branch on [pgmccann/my-project](#).



 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 line instructions](#).



Write

Preview

[Leave a comment](#)

Adds some dummy documentation #1

 Open

pgmccann wants to merge 1 commit into `main` from `documentation` 

 Conversation 0

 -o- Commits 1

 Checks 0

 Files changed 2



pgmccann commented 10 minutes ago

Owner

 ...

No description provided.

-o- Adds some dummy documentation

a98b52d

Add more commits by pushing to the `documentation` branch on **pgmccann/my-project**.



Merge pull request #1 from **pgmccann/documentation**

Adds some dummy documentation

 pgm5@st-andrews.ac.uk 

Choose which email address to associate with this commit

Confirm merge

Cancel



Write

Preview

Adds some dummy documentation #1

↳ Merged

[pgmccann](#) merged 1 commit into [main](#) from [documentation](#)  now

Conversation

-o Commits 1

 Checks 0

 Files changed 2



pgmccann commented 11 minutes ago

Owner

• • •

No description provided



 Adds some dummy documentation

a98b52d



 pgmccann merged commit `de72e92` into `main` now

Revert



Pull request successfully merged and closed

Delete branch



Write

Preview

Leave a comment

main ▾

Go to file

Add file ▾

Code ▾



pgmccann Merge pull request #1 from pgmccann/document... ... 22 seconds ago ⏱ 15

📁 data Adds input data 3 days ago

📁 docs [Adds some dummy documentation](#) 18 hours ago

📁 src Adds analysis script Adds some dummy documentation 3 days ago

📁 test Adds test script 3 days ago

📄 .gitattributes Initial commit 9 days ago

📄 .gitignore Ignores the output data yesterday

📄 LICENSE.txt Adds MIT license 3 days ago

📄 README.md Updates license statement in README 21 hours ago

📄 run.sh Adds bash script to run analysis 3 days ago

README.md



Forking

Forking looks a bit like branching - we create a fork, can make changes there without changing things elsewhere, and we can use pull requests to merge those changes.

However, whereas branching occurs within a repository, forking takes place across repositories.

It's really a function of platforms like GitHub and GitLab rather than a part of Git itself.

Imagine coming across an open-source R package on GitHub which does exactly the analysis you need, but the you need the output in a different format.

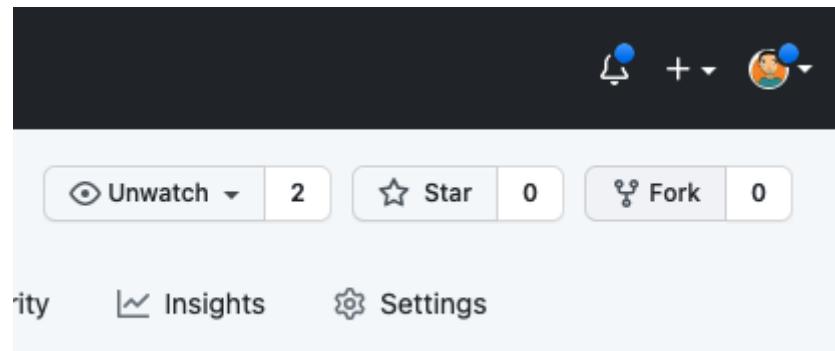
You can fork the repository, creating a clone of it in your GitHub account, including all branches and commit details.

You can then make changes in your fork of the application, so that is works as you need it to.

The new repo is linked to the old one. So, maybe you spot something in a repository that needs to be fixed - could be a bug, or just a typo in the documentation.

You can fork the repository, fix the issue, and submit a pull request to the original repository to get your changes included there.

You can generally fork any public repository.



Fork my-project X

Where should we fork my-project?



code4ref



StAResComp



KHP-Informatics

Can't find what you're looking for?

You don't have permission to fork to these organizations:



openpreserve

has MIT license

3 days ago

forked from [pgmccann/my-project](#)[Watch](#)

0

[Star](#)

0

[Fork](#)

1

[Code](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)[main](#)[2 branches](#)[0 tags](#)[Go to file](#)[Add file](#)[Code](#)

This branch is up to date with pgmccann:main.

[Contribute](#)[Fetch upstream](#)pgmccann Merge pull request [pgmccann#1](#) from pgmccann/d...

...

de72e92 3 hours ago

15 commits

[data](#)

Adds input data

3 days ago

[docs](#)

Adds some dummy documentation

21 hours ago

[src](#)

Adds analysis script

3 days ago

[test](#)

Adds test script

3 days ago

[.gitattributes](#)

Initial commit

9 days ago

[.gitignore](#)

Ignores the output data

yesterday

[LICENSE.txt](#)

Adds MIT license

3 days ago

[README.md](#)

Updates license statement in README

yesterday

[run.sh](#)

Adds bash script to run analysis

3 days ago

[README.md](#)

About



Demonstration project for talk on
Collaboration using Git

[Readme](#)[MIT License](#)

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages



Go to file Add file ▾ Code ▾

Contribute ▾ Fetch upstream ▾

This branch is not behind the upstream pgmccann:main.

No new commits to fetch. Enjoy your day!

Compare Fetch and merge

3 days ago

from pgmccann:main

data

dummy

sis script

Relo...

No re...

Crea...

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Go to file', 'Add file ▾', and 'Code ▾'. Below that is a navigation bar with 'Contribute ▾' and 'Fetch upstream ▾'. A prominent green modal box appears, stating 'This branch is not behind the upstream pgmccann:main.' followed by 'No new commits to fetch. Enjoy your day!'. Below the modal are two buttons: 'Compare' and 'Fetch and merge', with 'Fetch and merge' being green. The main repository list shows a commit from 'pgmccann:main' 3 days ago. To the right of the repository list, there are several collapsed sections labeled 'Relo...', 'No re...', and 'Crea...'. The background shows parts of other repository cards.

1 commit 0 tags

Go to file

Add file ▾

Code ▾

Compare with pgmccann:main.

Contribute ▾

Fetch upstream ▾

request

This branch is not ahead of the upstream
pgmccann:main.

No new commits yet. Enjoy your day!

Open pull request

Adds analysis script

1 hour ago 15 commits

3 days ago

21 hours ago

3 days ago

StAResComp / my-project Public

forked from pgmccann/my-project

Code

Pull requests

Actions

Projects

Wiki

Settings

my-project / README.md

in main

Edit file

Preview

```
1  [[License: MIT](https://img.shields.io/badge/License-MIT-yellow.svg)](http://
2
3 # My Project
4
5 This is an example project to illustrate the use of Git for
6 collaboration in a research context.
7
8 © 2021 University of St Andrews. Made available under the terms of the MIT
9 License.
10
```

StAResComp / my-project Public

forked from pgmccann/my-project

Code

Pull requests

Actions

Projects

Wiki

Settings

my-project / README.md

in main

Edit file

Preview

```
1  [[License: MIT](https://img.shields.io/badge/License-MIT-yellow.svg)](http://
2
3 # Our Project
4
5 This is an example project to illustrate the use of Git for
6 collaboration in a research context.
7
8 © 2021 University of St Andrews. Made available under the terms of the MIT
9 License.
10
```

[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



base repository: pgmccann/my-project

base: main



head repository: StAResComp/my-project

compare: main

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

[Create pull request](#)

-o 1 commit

1 file changed

1 contributor

-o Commits on Nov 4, 2021

Update README.md



pgmccann committed 35 seconds ago

Verified



2971047



Showing 1 changed file with 1 addition and 1 deletion.

[Split](#)[Unified](#)

v 2 README.md



... ... @@ -1,6 +1,6 @@

1 1 [! [License: MIT] (<https://img.shields.io/badge/License-MIT-yellow.svg>) (<https://opensource.org/licenses/MIT>)

2 2

3 - # My Project

[Code](#)[Issues](#)[Pull requests](#) [1](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

Update README.md #2

[Edit](#)[Code](#)[Open](#)pgmccann wants to merge 1 commit into [pgmccann:main](#) from [StAResComp:main](#)[Conversation](#) [0](#)[Commits](#) [1](#)[Checks](#) [0](#)[Files changed](#) [1](#)[+1 -1](#)

pgmccann commented now

Owner



...

No description provided.

[Update README.md](#)

Verified

2971047

Add more commits by pushing to the [main](#) branch on [StAResComp/my-project](#).**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 line instructions.[Write](#)[Preview](#)[H](#) [B](#) [I](#) [≡](#) [<>](#) [♂](#) [≡](#) [≡](#) [checkbox](#) [@](#) [🔗](#) [↶](#)

Leave a comment

Reviewers

No reviews

Still in progress? Convert to draft



Assignees

No one—assign yourself



Labels

None yet



Projects

None yet



Milestone

No milestone



Linked issues



Open Science

Open Access Papers



Open Data



Open Software

Papers are not the only way to contribute to research

Make code available online under a suitable license

Make it citable (and cite software you use)

Archive your code, get a DOI and add it to Pure

Licensing

Making something public doesn't make it open-source!

You need to attach a **suitable license**.

Citation

There are several ways to include information about how your code should be cited.

The [Citation File Format](#) describes a fairly simple text format, normally used in a CITATION.cfffile.

[CodeMeta](#) is more complex, and can capture more metadata about the project than is relevant to citation - but the website includes a codemeta.json file generator.

Zenodo

GitHub's own documentation on '[Making your code citable](#)' describes how to archive your code and get a DOI from Zenodo.

You can link your GitHub account to your Zenodo account to easily archive repositories - and every time you tag a release in GitHub, it is automatically archived.

Pure

You can also add your software to [Pure](#). The steps involved are described at [Code4REF](#).

Other GitHub features

- Issues
- Project Management
- Automation with GitHub Actions

(All of these have equivalents in GitLab).

Resources

<https://software.ac.uk/resources>

Thank-you

research-computing@st-andrews.ac.uk

<https://researchcomputing.wp.st-andrews.ac.uk/team/>

Research Software in Teams