

Mass Spec Academy

Table of contents

Preface	3
Funding	3
I Overview	4
What is a Mass Spectrometer?	5
Background Contents	5
1 Physics and Chemistry Background	6
1.1 Forget the Spectrometer, What is a Mass?	6
1.1.1 SI units for mass	6
1.1.2 Other units for mass	6
1.1.3 Atomic masses of your favorite isotopes	7
1.2 Energy, Electricity, and Magnetism	8
1.2.1 Energy	8
References	10
2 Ion Sources	11
II Analytical Considerations	12
3 Baselines	13
4 Peak Tails and Abundance Sensitivity	14
References	15
III Isotope Systems	16
5 U-Pb	17
IV Interpreting Data	18
6 Statistics for Mass Spectrometer Jockeys	19

References	20
Appendices	21
A Contributing to Mass Spec Academy	21
A.1 Overview	21
A.2 Software Installation	21
A.2.1 Step 1: (Mini)conda	22
A.2.2 Step 2: VS Code	23
A.2.3 Step 3: Quarto CLI	23
A.2.4 Step 4: GitHub Desktop	25
A.3 Getting started!	26
A.3.1 1. Fork the MassSpecAcademy GitHub repository	26
A.3.2 2. Create a new branch for your edits	28
A.3.3 3. Use conda to download dependencies	29
A.3.4 3. Open VS Code	31
A.3.5 4. Commit your edits on GitHub Desktop	32
A.3.6 5. Create a Pull Request	36
A.4 Get contributing!	40

Preface

The goal of this project is to create an online resource that provides comprehensive and accessible information on mass spectrometry methods used in geochronology, comprising:

- A detailed description and history of the method
- Best practices for sample preparation and analysis
- A guide to data and uncertainty analysis
- Applications, and case studies from peer-reviewed research
- Short exercises and worked solutions appropriate for graduate students.

Funding

This material is based upon work supported by the National Science Foundation under Grant Nos. EAR-2218547 and -2218544. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Thanks to the AGeS program for its support.



Part I

Overview

What is a Mass Spectrometer?

A mass spectrometer separates atoms by their atomic mass. Scientists have long known how to separate different elements based on their chemical properties. Thousands of years ago, metallic copper was first smelted from copper ore, giving rise to the [Copper Age](#). Modern geochemical labs efficiently separate even very chemically similar elements, such as the rare earth elements, via techniques like anion exchange chromatography. These techniques separate one element from another, like separating samarium (Sm) from neodymium (Nd).

Recall that the number of protons in an atom determines the number of electrons that are needed to balance their charge. The number of electrons in an atom determines its chemical behavior — whether and how it makes chemical bonds with other atoms. But the nucleus of an atom doesn't just contain protons, it also contains neutrons. Atoms with the same number of protons but different numbers of neutrons are called isotopes. Different isotopes of the same element behave in a chemically similar manner: you can make CO_2 with ^{12}C , ^{13}C , or ^{14}C .

The relative abundances of the isotopes of an element are key to a broad array of geological processes, including the radioactivity used as a clock in geochronology, geochemical processes that fractionate radioactive parent isotopes from their radiogenic daughter products, and temperature- and environment-dependent kinetic reaction rates. To separate isotopes by their atomic mass, we need a mass spectrometer.

Separating atoms by their atomic mass is usually accomplished by first ionizing the atoms, for instance by stripping an electron off to create an ion with a +1 positive charge. The ions can then be separated according to their mass-to-charge ratio, often denoted m/z , using electric and magnetic fields.

Background Contents

1 Physics and Chemistry Background

1.1 Forget the Spectrometer, What is a Mass?

1.1.1 SI units for mass

The SI unit for mass is the kilogram, but samples analyzed for mass spectrometry are usually much smaller. The table below lists some typical sample sizes for geochemistry, both as SI terms with prefixes and as fractions of a gram. As we will soon learn, the number of atoms in a gram depends on the atomic mass of the atoms. The third column gives the number of atoms of that mass. It starts with atoms of mass 12 unified mass units (i.e., ^{12}C), but you can hover your slider over the blue atomic mass and drag left or right to increase or decrease its value.

Table 1.1: SI prefixes for small things.

Mass with Prefix	Mass in grams	Atoms of ^{12}C	Atoms of ^{238}U
kilogram	10^3 grams	6×10^{26}	3×10^{25}
gram	1 gram	6×10^{23}	3×10^{22}
milligram	10^{-3} grams	6×10^{20}	3×10^{19}
microgram	10^{-6} grams	6×10^{17}	3×10^{16}
nanogram	10^{-9} grams	6×10^{14}	3×10^{13}
picogram	10^{-12} grams	6×10^{11}	3×10^{10}
femtogram	10^{-15} grams	6×10^8	3×10^7
attogram	10^{-18} grams	602214	30357

1.1.2 Other units for mass

The familiar (and perhaps unfamiliar!) SI prefixes down to the attogram still don't reach a small enough value to easily compare the masses of single atoms, like ^{238}U and ^{235}U . For that, we'll need a new unit, the unified mass unit, also known as the Dalton (symbols: u or Da). The unified atomic mass unit is not in the SI, but it's commonly used in physics and chemistry for very small masses, like the mass of a single atom or molecule. It's defined as $\frac{1}{12}$ the mass of a ^{12}C atom. That's about 1.660539×10^{-27} kilograms. The equivalent unit Dalton is more widely used in the organic chemistry community.

What about the atomic mass unit, or amu? This very similar unit was used widely in the mid-twentieth century but was defined differently by physicists and chemists. It was formally abandoned in 1961, replaced by the unified atomic mass unit and the Dalton, and assigned unique unit abbreviations. However, many scientific communities still use amu to abbreviate the unified atomic mass unit. The inorganic mass spectrometry community is among them, and this textbook will use amu below.

1.1.3 Atomic masses of your favorite isotopes

The isotope ^{12}C is the only isotope with an integer mass (it has a mass of 12 amu). Other isotopes have non-integer masses, which are determined to high precision by nuclear physicists. Masses and 1σ uncertainty in parentheses are from Wang et al. (2021):

- ^1H has a mass of 1.007825031898(14) amu
- ^{86}Sr has a mass of 85.9092607309(91) amu
- ^{144}Nd has a mass of 143.9100873(25) amu
- ^{208}Pb has a mass of 207.9766521(13) amu
- ^{238}U has a mass of 238.0507882(20) amu

Isotopic masses aren't integers for several reasons. First, neutrons and protons don't have exactly the same mass. Neutrons are slightly heavier than protons (1.0087 vs. 1.0073 amu, respectively). But an atomic mass is different from the sum of the masses of its protons, neutrons, and much lighter electrons. The difference is the binding energy of the atom and specifically the nucleus, or the energy released by the formation of the nucleus from its constituent parts. This energy of fusion, which powers the sun and stars, can be converted to mass via Einstein's famous equation $e = mc^2$. So the combined mass of 6 protons + 6 neutrons + 6 electrons is 12.0989 amu, and the difference between that mass and the 12 amu mass of a ^{12}C atom is the energy released by putting the atom together.

The chemical energy released by forming a molecule out of atoms is small relative to the nuclear forces responsible for forming atoms, so the molecular mass of a molecule is very close to the sum of the atomic masses of its atoms. Note that two molecules with the same chemical formula might have two different molecular masses. For instance, $^{12}\text{C}^{16}\text{O}_2$ will have a different molecular mass than $^{13}\text{C}^{16}\text{O}_2$ will have a different mass than $^{12}\text{C}^{18}\text{O}^{16}\text{O}$. These three molecules, all with a different molecular mass, are called isotopologues.

Because each isotope has a slightly different mass, different atoms and/or molecules may have very similar masses. For instance, the mass of ^{40}Ar is 39.96238 amu, the mass of ^{40}Ca is 39.96259 amu, and the mass of ^{40}K is 39.96400 amu. Their proximity in mass makes these isotopes difficult (but not impossible) to separate with mass spectrometers. The more atoms a molecule has, the more opportunities isotopic substitution has to create near-overlaps. For instance, natural U is often measured by TIMS as UO_2^+ after adding a tracer containing synthetic U isotopes.

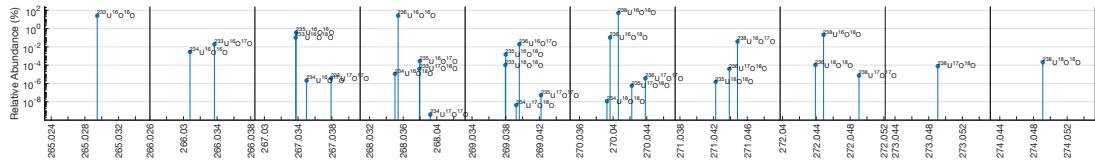


Figure 1.1: Uranium oxide (UO_2) isotopologues for a natural U sample with a ^{233}U - ^{236}U tracer added. The $^{238}\text{U}/^{236}\text{U}$ ratio is 0.5 for this sample, and the tracer $^{233}\text{U}/^{236}\text{U}$ is 1. Click to enlarge the figure.

1.2 Energy, Electricity, and Magnetism

To separate dissimilar objects like minerals or legos, one good strategy is to place them all together and then sort through and choose different elements from the pile. A chemical reaction might dissolve or precipitate one element and leave another behind. However, isotopes of the same element have nearly identical chemical behavior. Mass spectrometers don't inspect and sort a stack of static individual atoms like sorting legos, and they can't rely on chemical reactions to sort isotopes. Instead, mass spectrometers move the atoms by first ionizing them and then manipulating the ions with electrical and magnetic forces. The resulting kinetic changes in the isotopes' motion depend on their atomic or molecular mass, which can be exploited to separate different isotopes. Once separated, the streams of ions in motion must be measured by sensitive electronic instruments. Here again, the ions' electrical properties are important.

1.2.1 Energy

$$KE = qV = \frac{1}{2}mv^2 \quad (1.1)$$

```
import numpy as np
import matplotlib.pyplot as plt

kg_per_amu = 1.66e-27
mass_amu = 238
mass_kg = mass_amu * kg_per_amu
velocity_meters_per_second = np.linspace(0, 2, 100)
kinetic_energy_joule = mass_kg * velocity_meters_per_second ** 2

fig, ax = plt.subplots()

ax.plot(velocity_meters_per_second, kinetic_energy_joule)
```

```
ax.set_xlabel('Veclocity (m/s)')
ax.set_ylabel('Kinetic Energy (J)')
ax.set_title('Quadratic Function')
fig.tight_layout

plt.show()
```

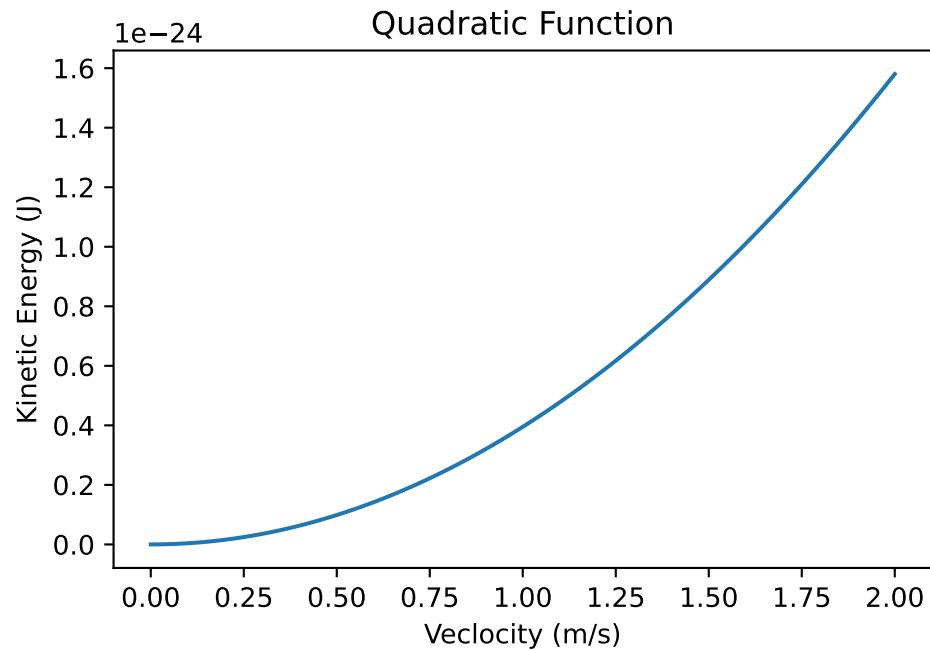
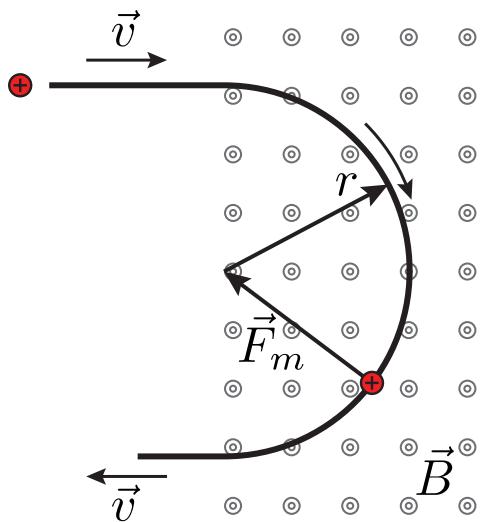


Figure 1.2: Kinetic energy as a function of velocity for ^{238}U



Magnetic force acts on a positively charged ion. Figure after Bill O.

References

- Wang, Meng, W. J. Huang, F. G. Kondev, G. Audi, and S. Naimi. 2021. “The AME 2020 Atomic Mass Evaluation (II). Tables, Graphs and References*.” *Chinese Physics C* 45 (3): 030003. <https://doi.org/10.1088/1674-1137/abddaf>.
- Yergey, Alfred L., and A. Karl Yergey. 1997. “Preparative Scale Mass Spectrometry: A Brief History of the Calutron.” *Journal of the American Society for Mass Spectrometry* 8 (9): 943–53. [https://doi.org/https://doi.org/10.1016/S1044-0305\(97\)00123-2](https://doi.org/10.1016/S1044-0305(97)00123-2).

2 Ion Sources



Figure 2.1: One of the beta unit calutron sources. ORNL Photo 24574, from Yergey and Yergey (1997).

Part II

Analytical Considerations

3 Baselines

4 Peak Tails and Abundance Sensitivity

References

- Wang, Meng, W. J. Huang, F. G. Kondev, G. Audi, and S. Naimi. 2021. “The AME 2020 Atomic Mass Evaluation (II). Tables, Graphs and References*.” *Chinese Physics C* 45 (3): 030003. <https://doi.org/10.1088/1674-1137/abddaf>.
- Yergey, Alfred L., and A. Karl Yergey. 1997. “Preparative Scale Mass Spectrometry: A Brief History of the Calutron.” *Journal of the American Society for Mass Spectrometry* 8 (9): 943–53. [https://doi.org/https://doi.org/10.1016/S1044-0305\(97\)00123-2](https://doi.org/10.1016/S1044-0305(97)00123-2).

Part III

Isotope Systems

5 U-Pb

Part IV

Interpreting Data

6 Statistics for Mass Spectrometer Jockeys

References

- Wang, Meng, W. J. Huang, F. G. Kondev, G. Audi, and S. Naimi. 2021. “The AME 2020 Atomic Mass Evaluation (II). Tables, Graphs and References*.” *Chinese Physics C* 45 (3): 030003. <https://doi.org/10.1088/1674-1137/abddaf>.
- Yergey, Alfred L., and A. Karl Yergey. 1997. “Preparative Scale Mass Spectrometry: A Brief History of the Calutron.” *Journal of the American Society for Mass Spectrometry* 8 (9): 943–53. [https://doi.org/https://doi.org/10.1016/S1044-0305\(97\)00123-2](https://doi.org/10.1016/S1044-0305(97)00123-2).

A Contributing to Mass Spec Academy

You can contribute to Mass Spec Academy by writing new material or editing existing content. The best way to do this and see how your new content will look is to use the Quarto publishing system and download and submit content through our GitHub repository.

A.1 Overview

We are using [Quarto](#) to render our instructional materials as user-friendly web and pdf documents. It's free and open source, and it's designed for open science projects like ours. Quarto lets you use all the components of a traditional academic manuscript or textbook, like text organized into chapters and sections, figures and tables with captions, equations with numbers, and references to cited work and to all those chapters/sections/equations/figures.

But wait, there's more: you can add movies, links to content from other chapters, code that drives computations and visualizations, callout blocks (like the topical side-boxes you see in textbooks, but with more options), and much else. Quarto renders this content both as a nice-looking webpage with lots of navigational features, and as a pdf file.

You can find nice examples of Quarto-authored books here:

- [R for Data Science \(2e\)](#)
- [Geocomputation with Python](#)
- [Causal Inference in R](#)

A.2 Software Installation

To author new or edit existing content on your computer, you'll need to install some software. This tutorial will help you get set up — please skip installs for any software you already have. You will need:

1. Conda to organize and manage our code tools
2. VSCode to write content
3. The Quarto Command Line Interface (CLI)
 - and the Quarto VS Code extension

4. GitHub Desktop as a user-friendly interface with GitHub (optional)

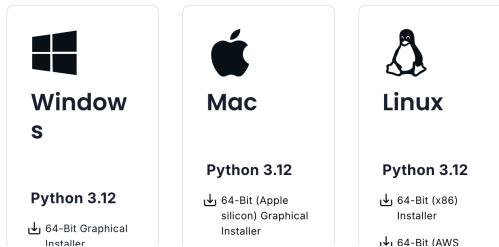
A.2.1 Step 1: (Min)conda

We'll use conda to keep track of all the smaller software packages that get used to create our source material. If you have already downloaded Anaconda for some previous data/science work, then great, you're all set! If not, you can download a smaller version without all the extra tools we won't use, called Miniconda.

This is the most involved install, so we're doing it first. The others are easy. Go to <https://www.anaconda.com/download/success>, scroll down past the Anaconda installers, and choose the Miniconda installer for your operating system. There is a Python version next to this, but don't worry about it, there's no Python involved here. If you're still reading this text, you probably want the "Graphical Installer" option. Note that *DO NOT* need to send your email address to anaconda.com to download or use this.

Screenshot of anaconda.com/download/success

Miniconda Installers



More detailed instructions for Miniconda installation are [here for each OS](#) under "Basic install instructions." In short, the instructions are to choose an install path with no spaces (the default is a good idea).

For **Windows**, you'll also want to

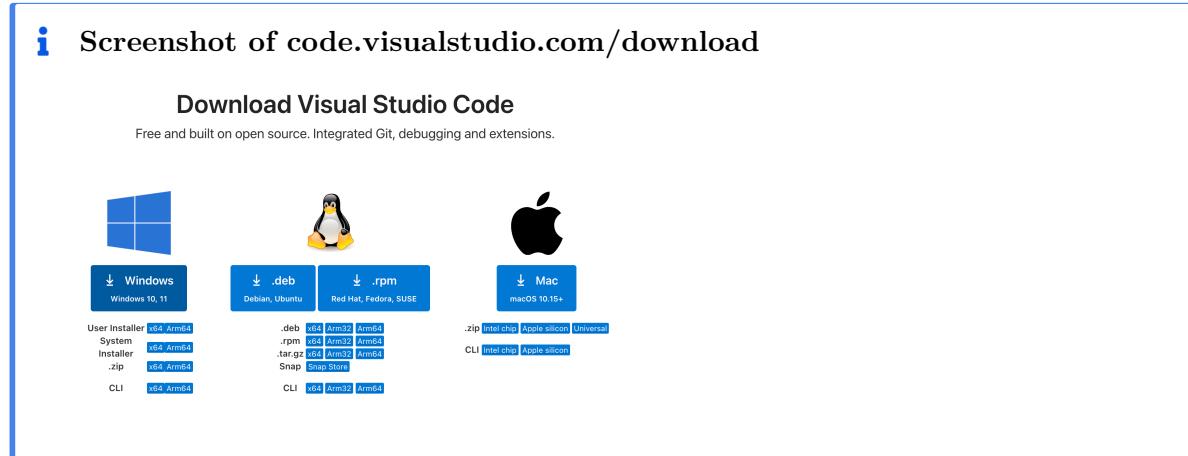
- install for "Just Me" instead of "All Users,"
- create shortcuts (selected by default) for Anaconda Prompt packages, and
- do not "Add Miniconda3 to my PATH variable"

That wasn't so bad, right?

A.2.2 Step 2: VS Code

You can use any editor you want for Quarto, but this tutorial will show you how to set up Microsoft Visual Studio (VS) Code. It's free, "built from open source," and it's widely used.

You can consider yourself a hacker once you download VS Code from <https://code.visualstudio.com/download>. If you don't know which of the buttons to choose for your operating system, then click on the big button. VS Code is smallish (< 200 MB download, < 500 MB disk space).

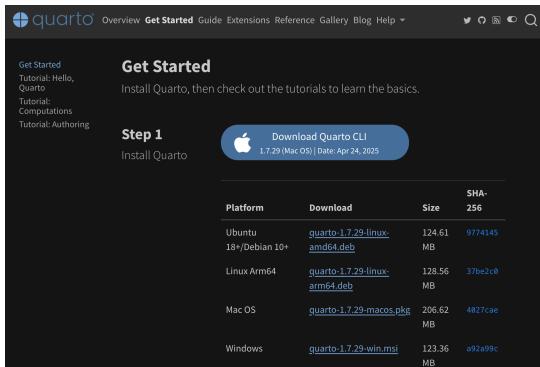


Install VSCode and follow the prompts. We'll add extensions and additional components later.

A.2.3 Step 3: Quarto CLI

Download the Quarto Command Line Interface (CLI) from <https://quarto.org/docs/get-started/> for your operating system. Install and follow the installer instructions. This is another small-ish ~200 MB installer. To update Quarto when a new version is released, you'll need to download the new installer from this website and install it again.

Screenshot of quarto.org/docs/get-started/

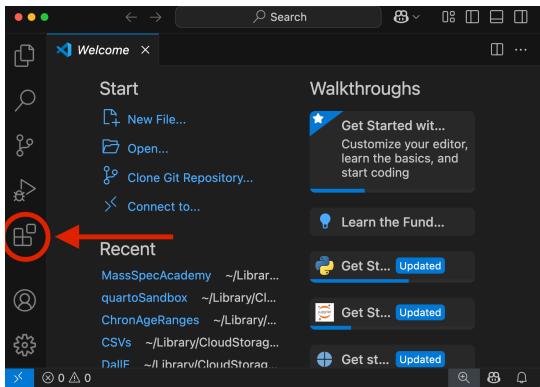


Quarto is what will turn the text we write in VS Code into content that looks pretty on the web, both on desktops and on mobile screens, and a pdf.

A.2.3.1 Step 3A: Add the Quarto VS Code Extension

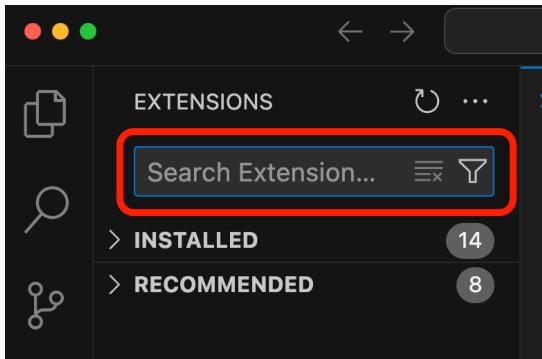
Open VS Code and then find the Extensions button on the left sidebar. It looks like some squares with the NE square levitating.

Screenshot of quarto.org/docs/get-started/



Then, click inside the Search bar and type “Quarto.”

Screenshot of quarto.org/docs/get-started/



You'll see the Quarto extension up first. Click on the extension to open an information window, then click the Install button on this window. You'll need the Quarto CLI installed in order to make the VS Code extension work.

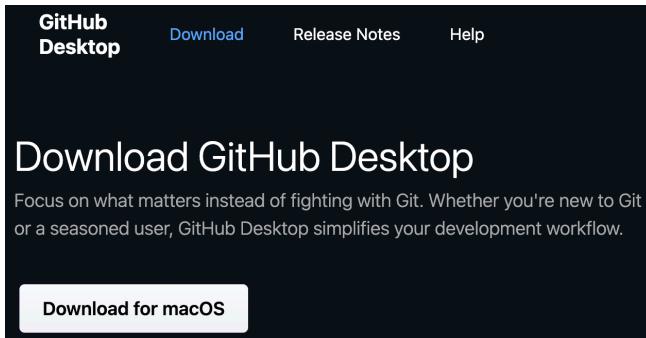
The Quarto extension lets VS Code recognize your Quarto-formatted files, then makes it easier to edit those files. It colors your text based on its purpose (headers, links, images, code), lets you drag and drop image files into documents, and provides a preview of your work with the click of a button.

A.2.4 Step 4: GitHub Desktop

We will use the version control software Git, and we will host all of our content, like writing and figures, on [GitHub.com](#). All current and previous versions of content on GitHub are publicly available through the GitHub.com website. Additionally, GitHub serves the [website version of our project](#), for free, through [GitHub Pages](#). Changes to the project that are pushed up to GitHub repository are automatically rendered to the website. If this makes no sense now, don't worry.

The easiest way to download and install Git and to start working with our GitHub repository is to download the GitHub Desktop app from <https://desktop.github.com/download/>. GitHub knows what operating system you're using, and probably what you ate for lunch last Wednesday, so just click the button to download.

i Screenshot of desktop.github.com/download



You will need to sign in with a GitHub.com username and password. Create one if you don't already — sorry, this is the only new login you'll need for this project.

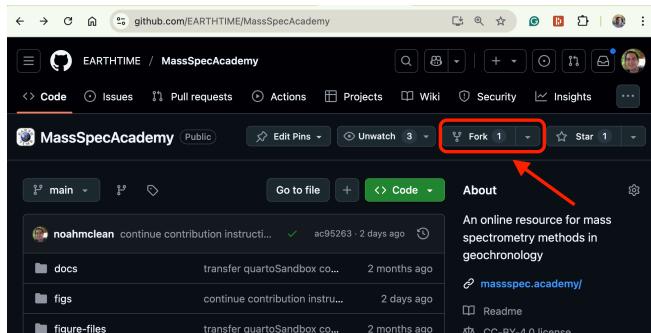
A.3 Getting started!

Now that you've installed all the software you need, we will download the current version of the project onto your computer using GitHub Desktop, install the packages it depends on using conda, and then open it for editing in VS Code.

A.3.1 1. Fork the MassSpecAcademy GitHub repository

Sign in to GitHub.com with your username and password, then navigate to <https://github.com/EARTHTIME/MassSpecAcademy>. Find and click the 'Fork' button at the top right of the screen.

i Screenshot of github.com/EARTHTIME/MassSpecAcademy

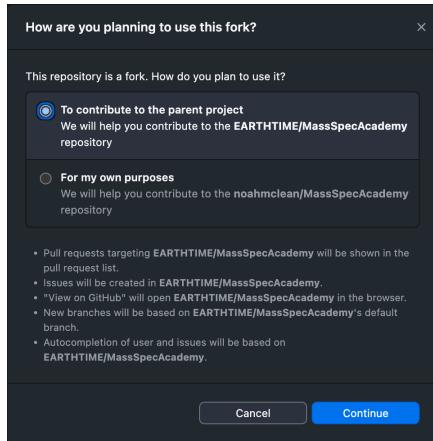


Press the Green “Create Fork” button at the bottom left of the next screen to create your own copy of the MassSpecAcademy repository on your own GitHub account. Now open the GitHub Desktop App, click the sign in with your GitHub username and password, and then from the “File” menu, choose “Clone Repository...”

In the window that pops up, find your new personal copy of the MassSpecAcademy repository in the list of repositories from GitHub.com. This repository will be a single folder with all the files and subfolders of Mass Spec Academy inside it. Choose a Local Path for this folder (the default is usually good), then click the blue Clone button.

Next, a window from GitHub Desktop will pop up and ask you how you want to use the repository. You want to contribute to the parent project, so keep the radio button in its top position and hit Continue.

i Screenshot of popup: how do you want to use this repository?



You now have a copy of Mass Spec Academy on your own computer!

A.3.2 2. Create a new branch for your edits

A.3.2.1 Quick workflow overview:

There are now two versions of the MassSpecAcademy repository on GitHub: the main project repository owned by EARTHTIME and the forked copy of the repository that you now own. The main project repository on GitHub is called “upstream” and the one attached to your GitHub account is called “origin.” We also just used GitHub Desktop to make a local copy of your fork on your computer. This local copy is a little sandbox where you can change anything you want — you could mess up your local repository and/or your forked repository on GitHub without affecting the larger project or its web presence.

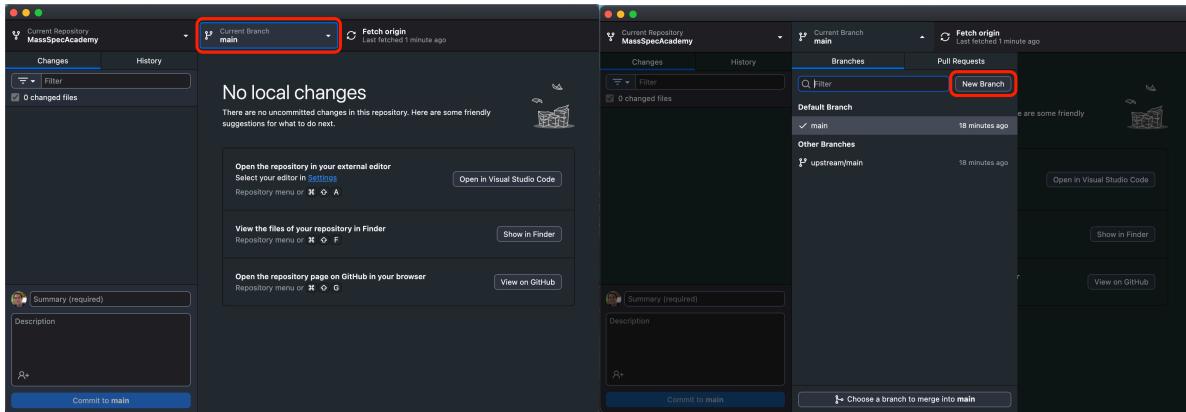
The feature branch we’re about to create in your forked GitHub repository will be a place that you create and edit your Mass Spec Academy content. You can create a new branch for each contribution you plan to make. For instance, you could make a branch for a project like “Start TIMS applications chapter” or “Edit background sections.” You’ll use GitHub desktop to package up each small batch of edits as a “commit” and to synchronize the edits on your local computer with your forked GitHub repository online.

When you’re at a place with your project where you want to merge your edits with the “upstream” main repository, you will create a Pull Request using the instructions below. This alerts the project maintainers (currently [Stephen Cox](#) and [Noah McLean](#)) that your content is ready. The maintainers can review your content, make suggestions or edits with the opportunity for a group discussion, and then merge your content into the main project. The merge will become visible to everyone else working on the project, and it will automatically update the website and pdf document. Now you can make a new branch for your next set of edits!

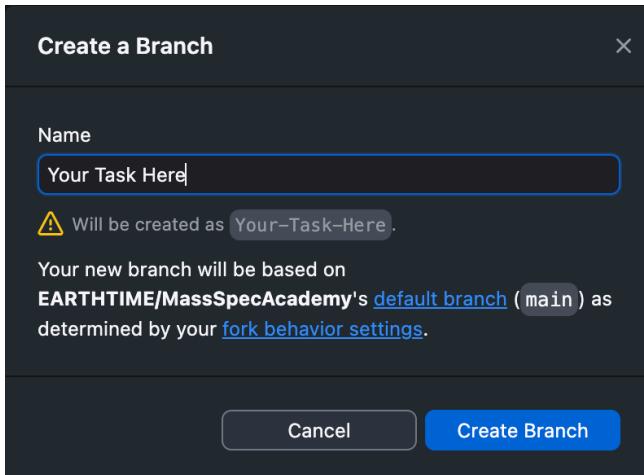
A.3.2.2 Create your new feature branch.

In GitHub Desktop, find the “Branches” dropdown menu in the middle of the top of the app window. Click anywhere in the button to make the dropdown menu appear, and click the “New Branch” button at its top right.

Type in a name for your feature branch that describes the work you’re about to do, or just “testing” if you want to try it out. Hit enter to create your new branch, and you’re off to the races.



i GitHub Desktop new branch window screenshot



As long as the name of your branch is showing under the “Current Branch” button on GitHub Desktop, your edits will belong to this branch.

A.3.3 3. Use conda to download dependencies

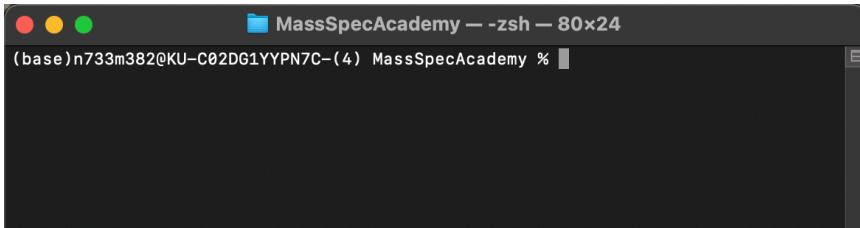
In order to reproduce the Mass Spec Academy website and run all of its code blocks, like the ones planned for its data and statistics chapters, you’ll need to install Python and some packages. This is easy with conda.

On a Mac, navigate in Finder to your new MassSpecAcademy folder that contains our local git repository. Then right-click on the MassSpecAcademy folder and, down the list of options, choose “New Terminal At Folder”.

On Windows, navigate in Windows Explorer to the MassSpecAcademy folder you created with GitHub Desktop. Copy the folder location from the address bar at the top of Windows Explorer. Now open the Anaconda Command Prompt application that was created when you installed Miniconda – it should be listed beside all your other applications, available from your Windows button. Type `cd "` (that is the letter `cd`, a space, then a double quotes), then paste in the path you copied, then type a second `"` to close the quotes. Hit the enter key.

On Windows or a Mac, you should now see a blinking cursor to the right of some text that starts with `(base)` and ends with `MassSpecAcademy`, the name of your current folder, like this:

i Terminal screenshot from the start of making a new conda environment



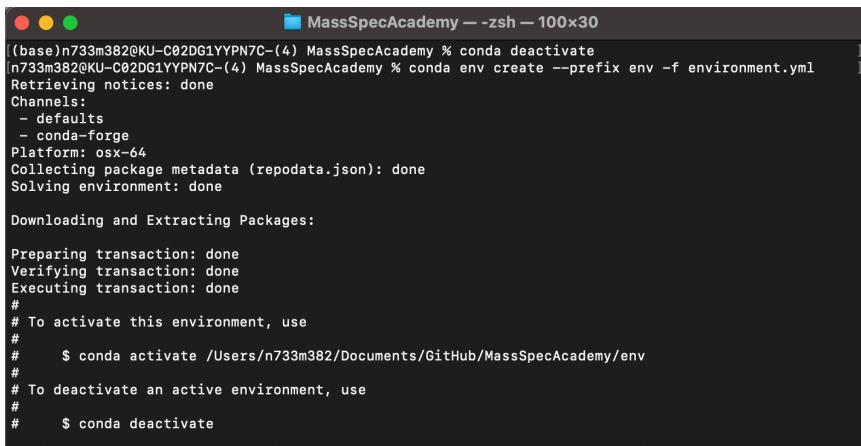
First, type `conda deactivate`, and hit enter. The `(base)` at the beginning of the line should disappear.

Now, copy and paste this text at that blinking cursor, then hit enter:

```
conda env create --prefix env -f environment.yml
```

When you hit enter, conda will think some and tell you what it is up to, i.e., ... “Retrieving notices,” “Solving environment,” “Collecting package metadata,” etc. After ten or twenty seconds, conda will output a few lines of instructions for activating and deactivating your new environment. You can ignore these — you’re done with conda and the command line for now. Your terminal should look something like this:

End of conda new environment from terminal



```
MassSpecAcademy -- zsh -- 100x30
(base)n733m382@KU-C02DG1YYPN7C-(4) MassSpecAcademy % conda deactivate
n733m382@KU-C02DG1YYPN7C-(4) MassSpecAcademy % conda env create --prefix env -f environment.yml
Retrieving notices: done
Channels:
- defaults
- conda-forge
Platform: osx-64
Collecting package metadata (repodata.json): done
Solving environment: done

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate /Users/n733m382/Documents/GitHub/MassSpecAcademy/env
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

A.3.4 3. Open VS Code

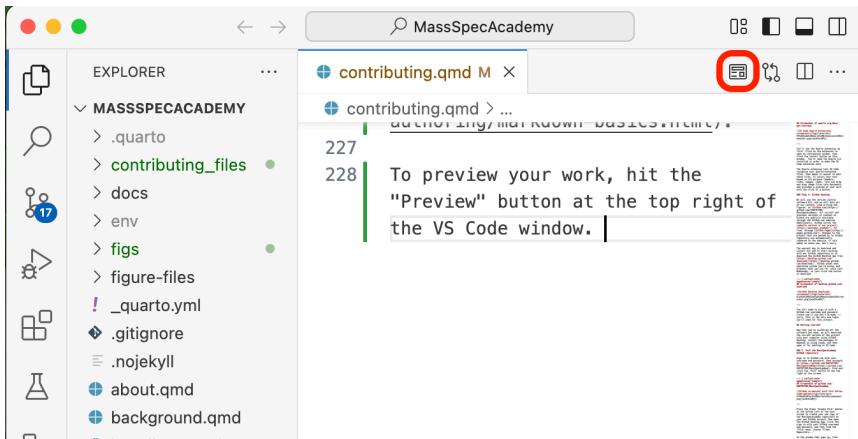
Open the VS Code application you downloaded and installed earlier. From the File menu, select “Open Folder...” and then navigate to the “MassSpecAcademy” folder that contains the forked GitHub repository you created with GitHub Desktop. It’s probably in your Documents/GitHub folder. Select the MassSpecAcademy folder and then hit the Open button to open the folder.

You’re in! You’ll see a panel to the left with a file directory for the whole repository. You can double-click a file to open it for editing, or create a new Quarto .qmd file (right-click and choose “New File”, use a button near the top of the panel, or from the “File” menu).

There’s lots more on authoring Quarto content at <https://quarto.org/docs/authoring/markdown-basics.html>.

To preview your work, hit the “Preview” button at the top right of the VS Code window:

VS Code screenshot with Preview button highlighted



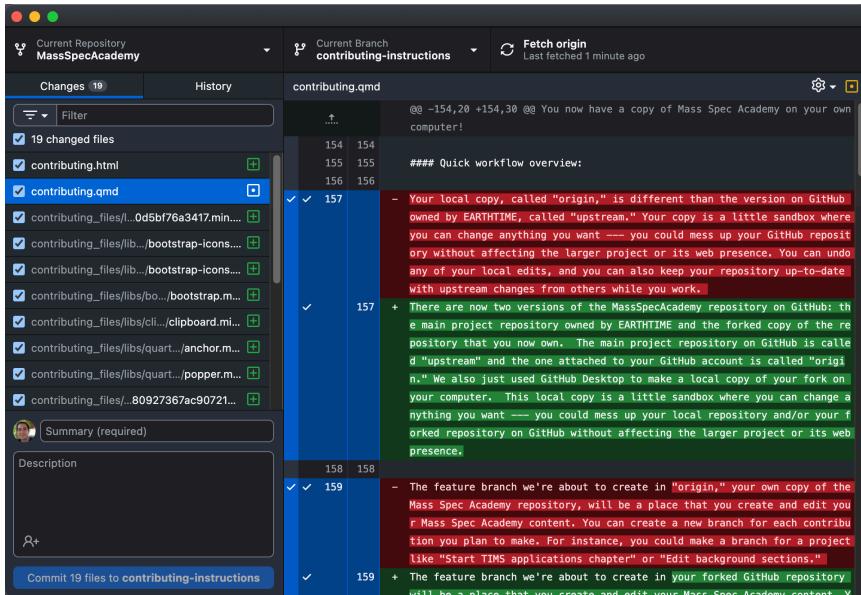
A preview panel will open on the right hand side of VS Code that you can resize to make readable. Another panel will open at the bottom that contains an integrated terminal window with output from Quarto about what it's doing.

If you've previewed a file that is already part of the website, then you will wait a minute, then see an interactive preview of the entire website. If you have previewed a new .qmd file that is not yet attached to the website navigation, you will see a preview of this page only. Command or control-click the `http://localhost:XXXX/` link that appears in the terminal window or copy the same text from the top of the preview window and paste it into a browser to preview your content in a web browser. You can hit preview as often as you like to preview your changes and make sure your content is rendering correctly.

A.3.5 4. Commit your edits on GitHub Desktop

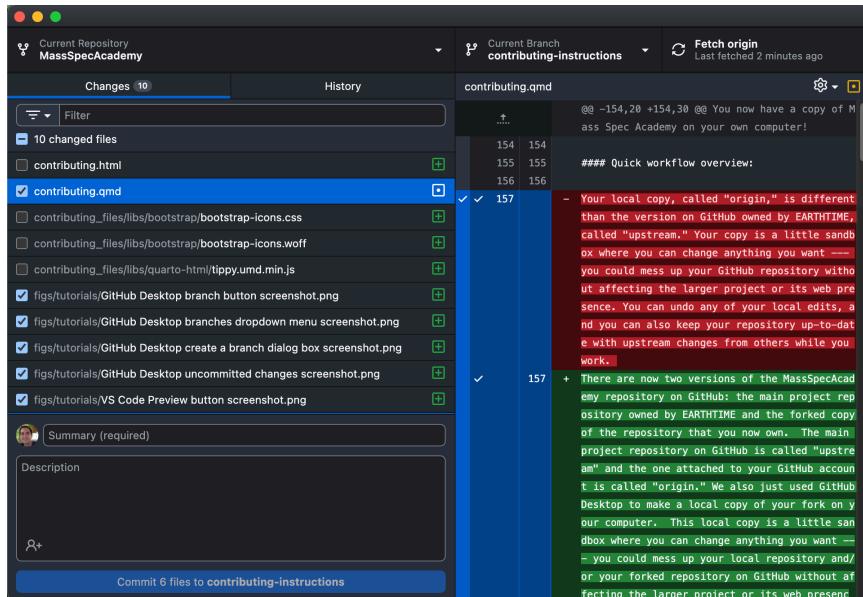
When you've hit a natural stopping place, like completing some writing or editing a section, or you're wrapping up and taking a long break, then it's time to commit your work. Open GitHub Desktop and you'll find that the left-hand panel titled "Changes" contains a list of all the files you've added, modified, or deleted. Each file will have a check box next to it, checked by default. The larger right-hand panel will show you all the changes you've made, with new content in green and deleted content in red.

GitHub Desktop screenshot with uncommitted changes



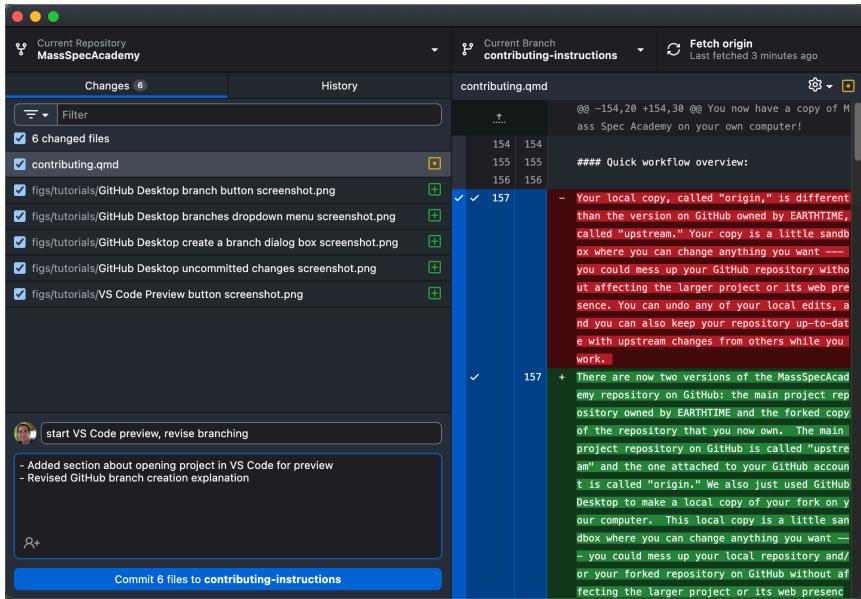
You'll recognize some of the files on the list as ones you added, edited, or deleted, and perhaps others as ones that Quarto created for your preview. You just want to commit the files that you added/edited/deleted. Un-check the preview-related files, which end with .html, .js, .css, and .woff.

- i GitHub Desktop:** only stage (leave checkbox checked) the files you added/edited/deleted



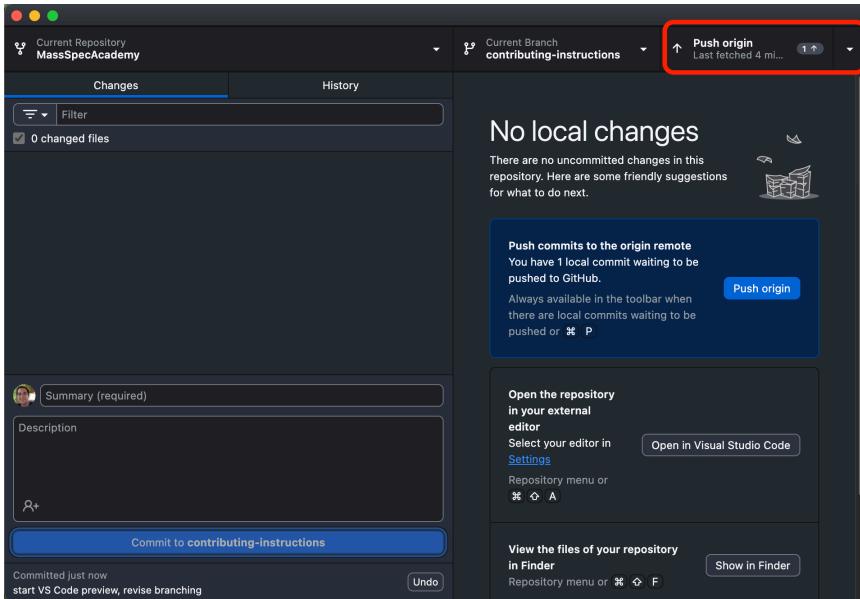
Next, find the commit message area in the bottom left of the GitHub Desktop window. In the “Summary” bar, write a short summary of your edits (ideally < 50 characters), like “Add explanation of work function for TIMS.” In the the “Description” box below, you can optionally leave a concise explanation of your edits. The commit message will be preserved alongside your edits and will help everyone understand your proposed changes and find where/when content was edited in case changes need to be reverted.

GitHub Desktop example commit message



When you're done, hit the blue “Commit” button at the bottom left of the GitHub Desktop window. Your changes have now been wrapped up as a commit in the local version of your MassSpecAcademy repository on your computer. To synch the commit with your forked GitHub repository online, called “origin”, click the “Push origin” button at the top right of the GitHub Desktop window.

GitHub Desktop screenshot of commit ready to push to origin

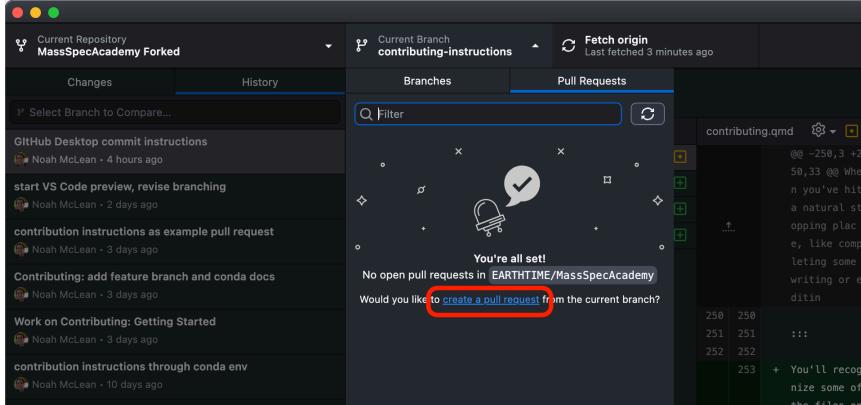


A.3.6 5. Create a Pull Request

At any time after you've made the first commit to your new branch, you can open a **draft pull request**, which alerts everyone working on the upstream EARTHTIME/MassSpecAcademy repository that you have some changes you'd like to eventually merge into the main repository. You can keep making commits to your branch, which then get added to the draft pull request. When the branch is ready, you'll indicate that your work is "Ready for review." Stephen and Noah will make sure it doesn't break anything, then merge it into the main repository, which will update the website automatically.

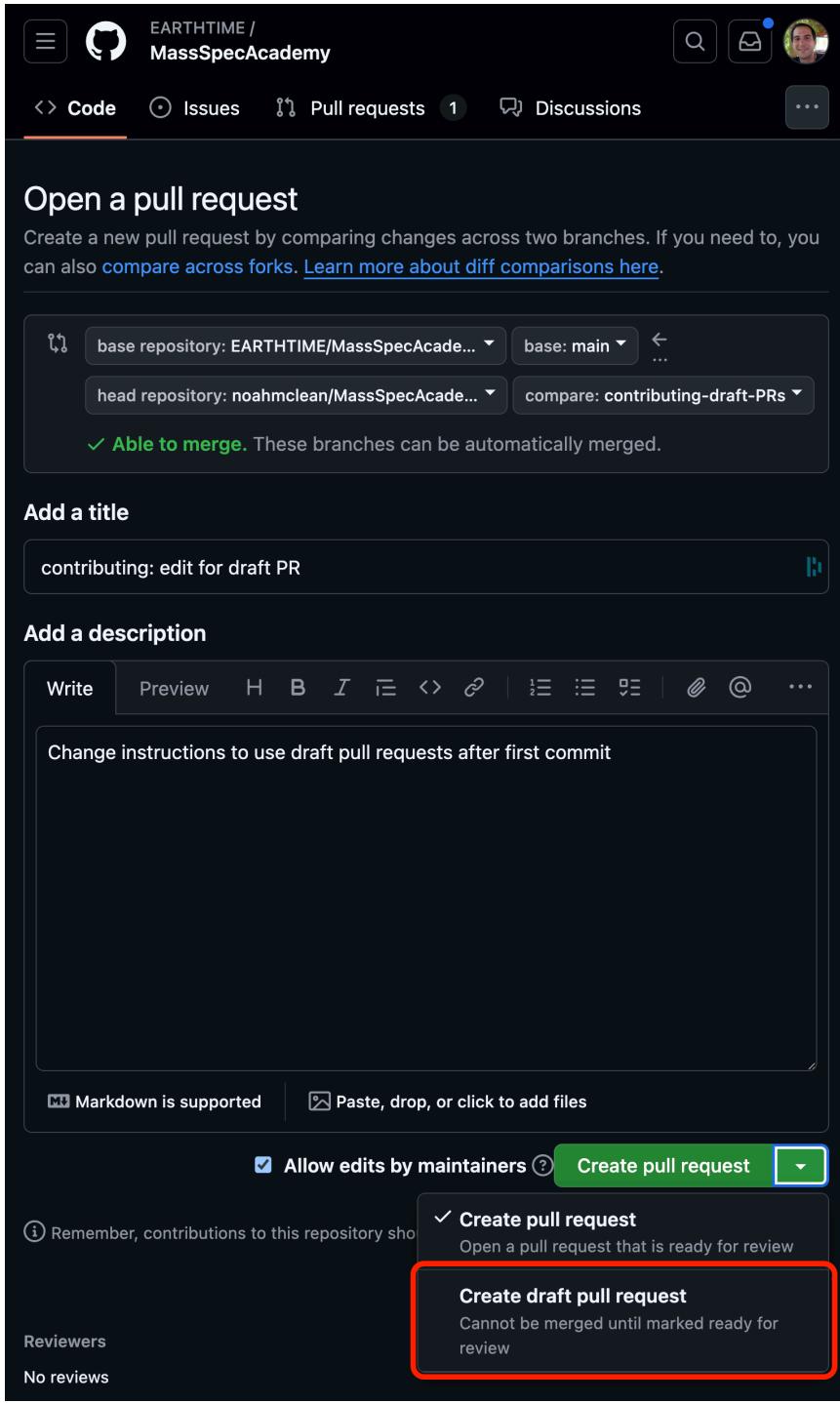
To create a draft pull request, click on the "Current Branch" dropdown button at the center top of the GitHub Desktop window. There are two tabs for the dropdown, "Branches" and "Pull Requests." Click the "Pull Requests" tab to see a blue text link that offers to open a new pull request for you.

i GitHub Desktop screenshot of the pull requests tab on the “Current Branch” dropdown



Clicking the blue link to “create a pull request” will open a new browser tab at the EARTH-TIME/MassSpecAcademy repository. The tab will be pre-populated with all the right selections for creating a new pull request from the branch in your forked repository that you’ve been working on. Enter a good summary and description for what you plan to contribute in this pull request (e.g., “update SIMS figures” or “add chapter on calutron history”). Then click the dropdown arrow next to the green “Create Pull Request” button and choose “Create draft pull request”:

i GitHub website screenshot of creating a draft pull request



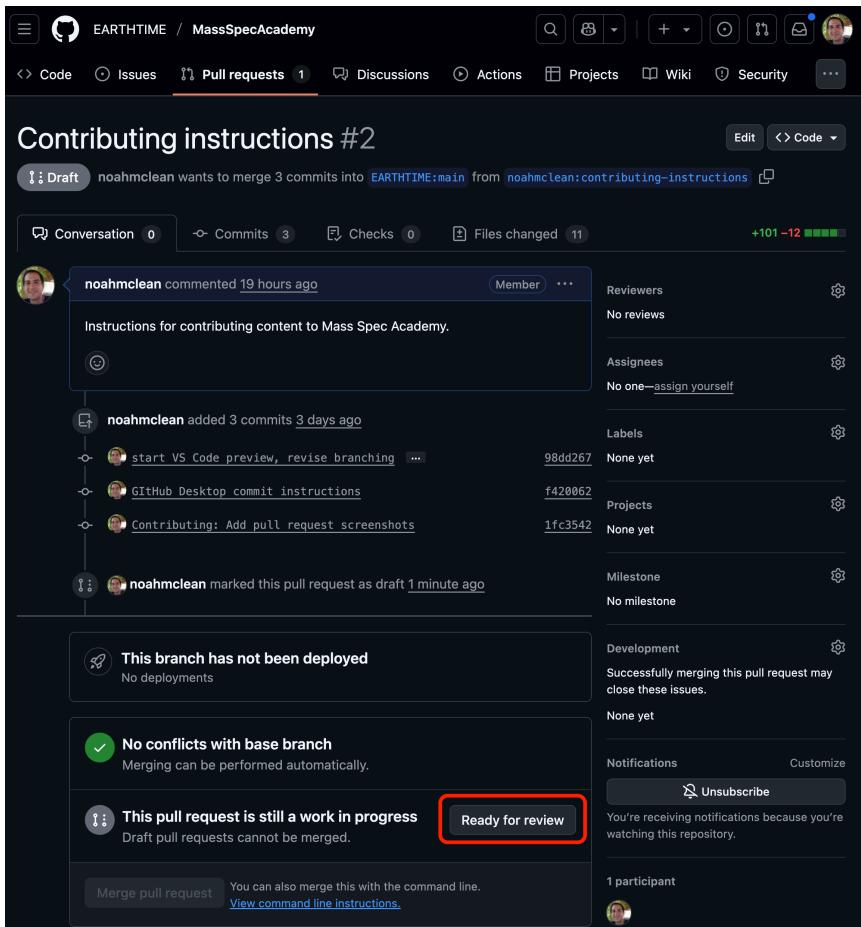
You can go to <https://github.com/EARTHTIME/MassSpecAcademy/pulls> to see all the other open draft pull requests, which represent others' in-progress contributions. Click the link for a pull request to read the commit messages and see how each project is progressing, and click a link to a commit to view its contents. This way, even though everyone is working on their own personal copy of the code on their own computer, you can get a sense of what others are doing. The pull requests page on the central repository linked above also serves as a discussion page – you can leave comments, ask questions, and make suggestions here, and the author will be notified by with an email from GitHub.

A.4 Get contributing!

Now that you have all the software installed and infrastructure in place, you should be able to add and edit content using the Quarto plugin for VS Code, previewing the results in a browser window as you write. When you've completed a task or chunk of text, you can commit those changes to your local repository on GitHub Desktop and push those changes to your online forked repository on GitHub.com.

When your content is ready to be added to the upstream EARTHTIME/MassSpecAcademy repository and the website, then change your draft pull request to a submitted pull request, which tells Stephen and Noah it's time to review and merge your branch. To do this, go to <https://github.com/EARTHTIME/MassSpecAcademy/pulls>, make sure all of your commits are present, and then click the “Ready for review” button:

i GitHub website screenshot of a draft pull request that's ready for review.



Once your branch is merged, you can check <https://massspec.academy/> to see it live – new content should appear in about a minute.

The contribution process now looks like:

1. Start another branch with a new name for your new task in GitHub Desktop,
2. create some new content in VS Code,
3. commit the content on GitHub Desktop with a descriptive commit message,
4. create a draft pull request using GitHub Desktop and GitHub.com,
5. keep writing and committing as you go, then
6. when you're done, indicate on GitHub.com your pull request is ready to review.

For Quarto syntax and formatting, like headings, figures, equations, and more, check out the [Authoring Guide](#) on the <https://quarto.org/> website.