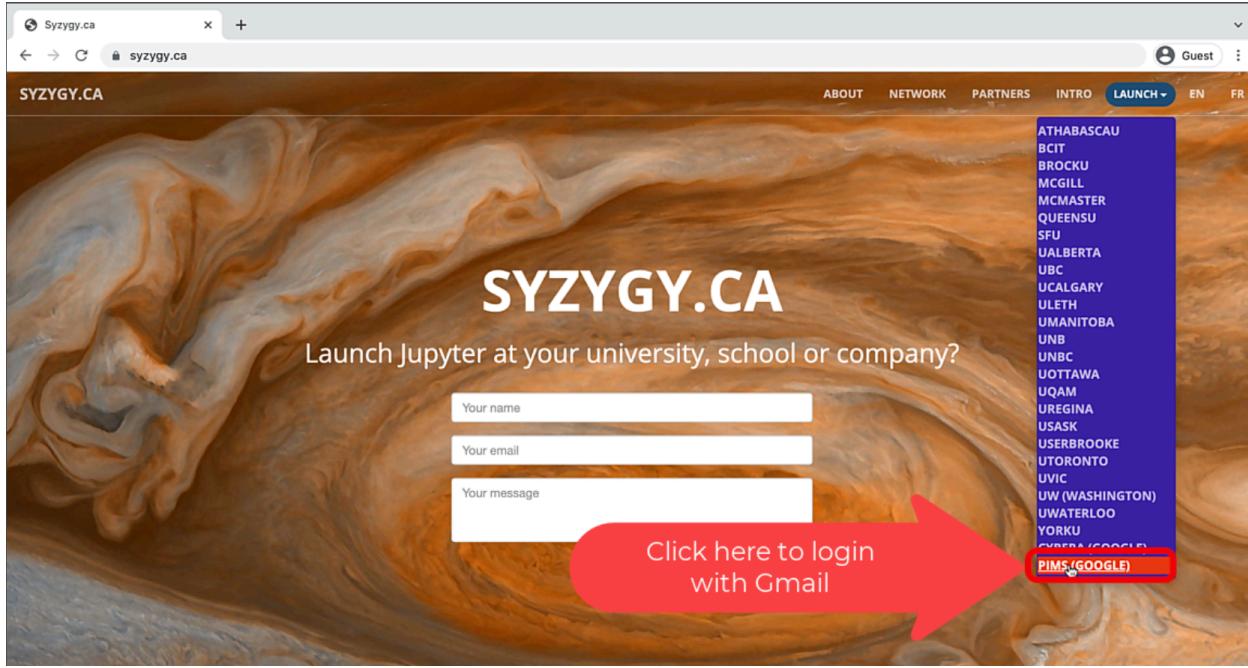


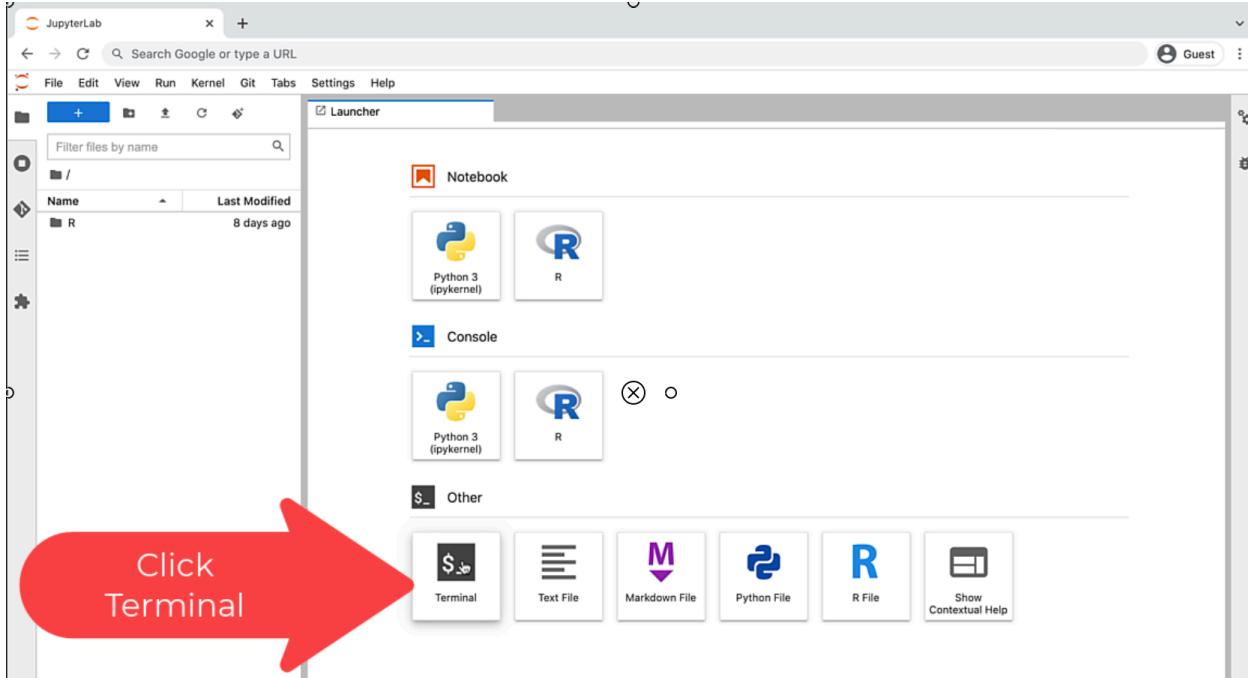
Scholar Metrics Scraper Instructions

Step 0: Setup

Go to syzygy.ca, click Launch, and either sign in with your Gmail or institutional account.

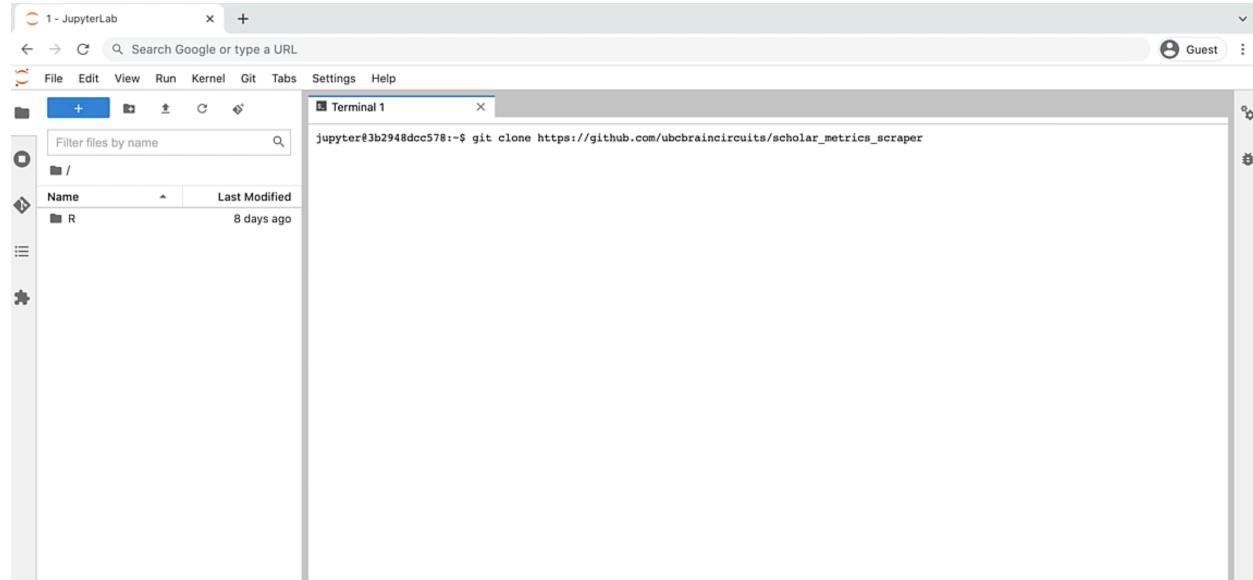


Open Terminal



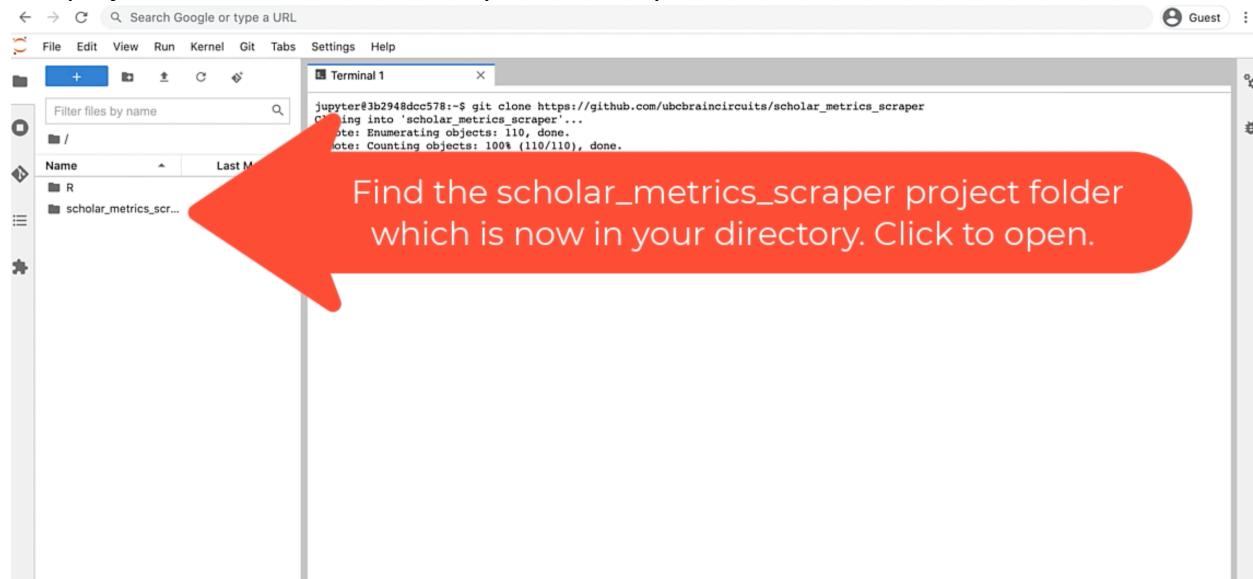
Step 1: Clone project from GitHub

Clone the project from Git. Type "git clone https://github.com/ubcbraincircuits/scholar_metrics_scraper" and press enter.



```
jupyter@3b2948dcc578:~$ git clone https://github.com/ubcbraincircuits/scholar_metrics_scraper
```

The project folder should now be in your directory

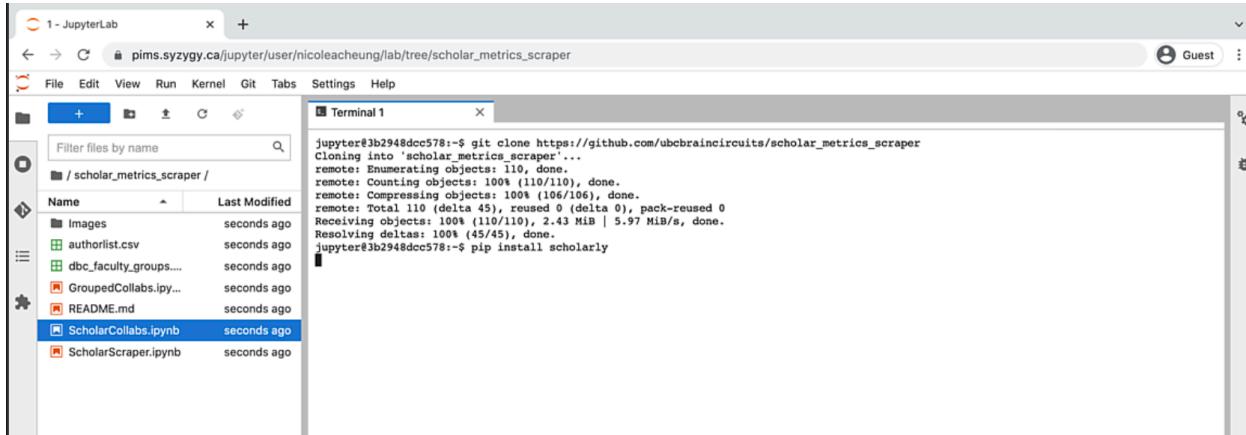


```
jupyter@3b2948dcc578:~$ git clone https://github.com/ubcbraincircuits/scholar_metrics_scraper
Cloning into 'scholar_metrics_scraper'...
Note: Enumerating objects: 110, done.
Note: Counting objects: 100% (110/110), done.
```

Find the scholar_metrics_scraper project folder which is now in your directory. Click to open.

Step 2: Install scholarly

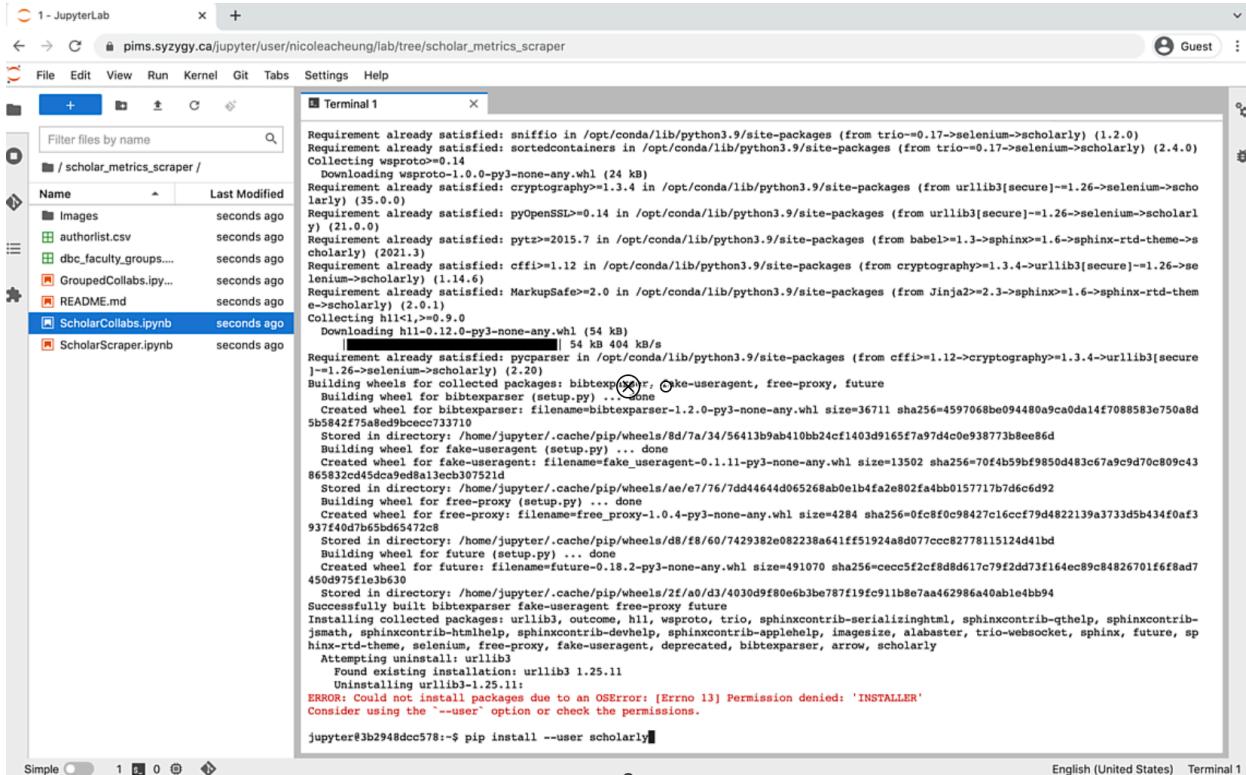
Install scholarly. Type “pip install scholarly” and press enter.



The screenshot shows a JupyterLab interface. On the left, there's a file tree for the directory `/scholar_metrics_scraper/`. In the terminal window on the right, the command `git clone https://github.com/ubcbraincircuits/scholar_metrics_scraper` is run, followed by `pip install scholarly`.

```
jupyter@3b2948dcc578:~$ git clone https://github.com/ubcbraincircuits/scholar_metrics_scraper
Cloning into 'scholar_metrics_scraper'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (106/106), done.
remote: Writing objects: 100% (110/110), done.
Receiving objects: 100% (110/110), 2.43 MiB | 5.97 MiB/s, done.
Resolving deltas: 100% (45/45), done.
jupyter@3b2948dcc578:~$ pip install scholarly
```

- If you receive an error, type “`pip install --user scholarly`” and press enter.



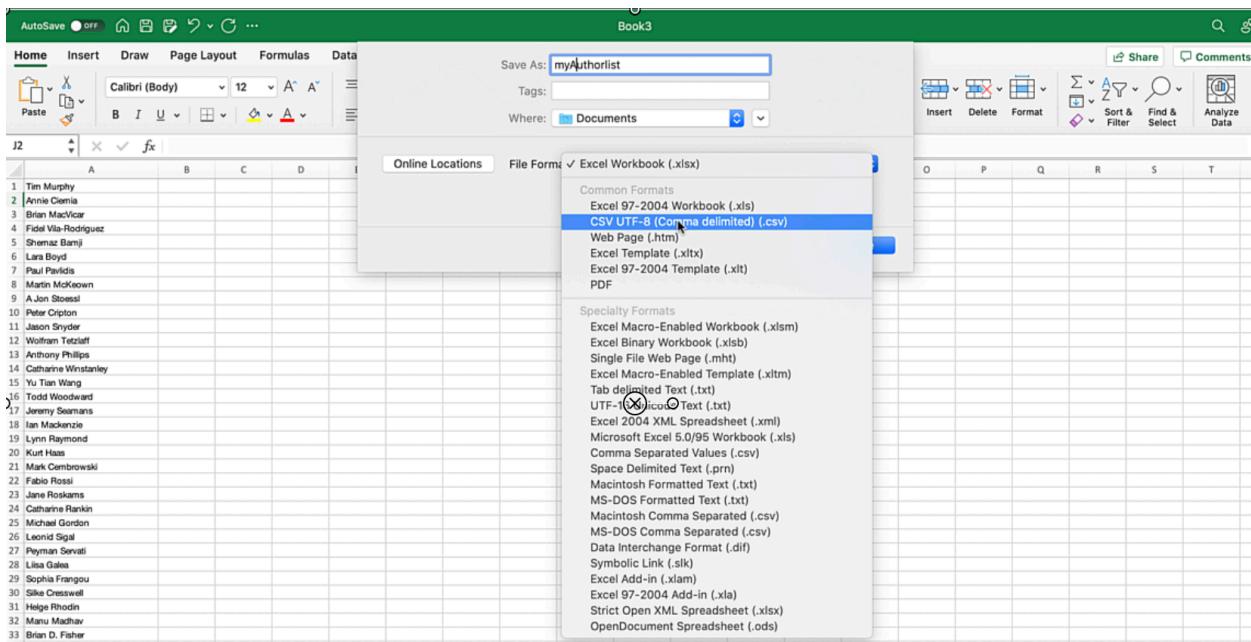
The screenshot shows a JupyterLab interface with a file tree and a terminal window. The terminal shows the command `pip install scholarly` being run, but it fails with an error message about permission denied.

```
Requirement already satisfied: sniffio in /opt/conda/lib/python3.9/site-packages (from trio==0.17->selenium->scholarly) (1.2.0)
Requirement already satisfied: sortedcontainers in /opt/conda/lib/python3.9/site-packages (from trio==0.17->selenium->scholarly) (2.4.0)
Collecting wsproto<0.14
  Downloading wsproto-1.0.0-py3-none-any.whl (24 kB)
Requirement already satisfied: cryptography<1.3.4 in /opt/conda/lib/python3.9/site-packages (from urllib3[secure]=1.26->selenium->scholarly) (35.0.0)
Requirement already satisfied: pyOpenSSL<0.14 in /opt/conda/lib/python3.9/site-packages (from urllib3[secure]=1.26->selenium->scholarly) (1.1.0)
Requirement already satisfied: pytz>=2015.7 in /opt/conda/lib/python3.9/site-packages (from babel>=1.3->sphinx>=1.6->sphinx-rtd-theme->scholarly) (2021.3)
Requirement already satisfied: cffi>=1.12 in /opt/conda/lib/python3.9/site-packages (from cryptography<1.3.4->urllib3[secure]=1.26->selenium->scholarly) (1.14.6)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.9/site-packages (from Jinja2>=2.3->sphinx>=1.6->sphinx-rtd-theme->scholarly) (2.0.1)
Collecting h1<1,>0.9.0
  Downloading h1-0.12.0-py3-none-any.whl (54 kB)
Requirement already satisfied: pyparsing in /opt/conda/lib/python3.9/site-packages (from cffi>=1.12->cryptography>=1.3.4->urllib3[secure]=1.26->selenium->scholarly) (2.2.0)
Building wheels for collected packages: bibtexparser
  Building wheel for bibtexparser: setup.py ...
    Created wheel for bibtexparser: filename=bibtexparser-1.2.0-py3-none-any.whl size=36711 sha256=4597068be094480a9ca0dal4f7088583e750a8d5b5842f75afed9bccce733710
    Stored in directory: /home/jupyter/.cache/pip/wheels/8d/7a/34/56413b5ab410bb24cf1403d9165f7a97d4c0e938773b8ee86d
  Building wheel for fake-useragent: setup.py ...
    Created wheel for fake-useragent: filename=fake_useragent-0.1.11-py3-none-any.whl size=13502 sha256=70f4b59bf9850d483c67a9c9d70c809c43865832cd45dc9ed8a13ecb307521d
    Stored in directory: /home/jupyter/.cache/pip/wheels/ae/e7/76/7d44644d065268ab0elb4fa2e802fa4bb0157717b7d6c6d92
  Building wheel for free-proxy (setup.py) ...
    Created wheel for free-proxy: filename=free_proxy-1.0.4-py3-none-any.whl size=4284 sha256=0fc8f0c98427c16ccf79d4822139a3733d5b434f0af3937240d7fd6d656
    Stored in directory: /home/jupyter/.cache/pip/wheels/d8/f8/60/7429382e082238a641ff51924a8d077ccc82778115124d41bd
  Building wheel for future (setup.py) ...
    Created wheel for future: filename=future-0.18.2-py3-none-any.whl size=491070 sha256=cecc5f2cf8dd617c79f2dd73f164ec89c84826701f6f8ad7450d975f1eb3b30
    Stored in directory: /home/jupyter/.cache/pip/wheels/2f/a0/d3/4030d9f80e6b1be787f19c911b8e7aa462986a40able4bb94
Successfully built bibtexparser fake-useragent free-proxy future
Installing collected packages: urllib3, outcome, h1, wsproto, trio, sphinxcontrib-serializinghtml, sphinxcontrib-qthelp, sphinxcontrib-jmath, sphinxcontrib-htmlhelp, sphinxcontrib-devhelp, sphinxcontrib-applehelp, imagesize, alabaster, trio-websocket, sphinx, future, sphinx-rtd-theme, selenium, free-proxy, fake-useragent, deprecated, bibtexparser, arrow, scholarly
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.25.11
    Uninstalling urllib3-1.25.11:
ERROR: Could not install packages due to an OSError: [Errno 13] Permission denied: 'INSTALLER'
Consider using the "-user" option or check the permissions.
jupyter@3b2948dcc578:~$ pip install --user scholarly
```

Step 3: Upload author list

Create a CSV file with a list of author names and/or Scholar IDs in a single column. Upload this to your project directory.

- Save the author list as a CSV file



- Upload the author list CSV to the project directory

The screenshot shows a JupyterLab interface with a red callout pointing to the 'Upload Files' button in the top navigation bar. The sidebar displays a file tree with several files listed, including 'ScholarScrapers.ipynb'. The main content area shows the 'ScholarScrapers' notebook, which includes an introduction about automating bibliometric data retrieval from Google Scholar using the 'scholarly' Python module.

Click here to upload files

ScholarScrapers

Introduction

This script automates the process of retrieving bibliometric data from Google Scholar for a list of authors. It utilizes [scholarly](#), a Python module that allows users to retrieve bibliometrics from [Google Scholar](#).

This project currently works with scholarly 1.4.5

Installation and Setup

1. Set-up Jupyter. If your institution has access, you can use [Syzygy](#) to run in the Cloud, or install on your computer following [these instructions](#).
2. Clone the project.

Make some modifications in the ScholarScraper notebook.

- Modify author_list_csv to match the author list CSV filename which you just uploaded.

The screenshot shows a Jupyter Notebook interface with the title 'scholar_metrics - JupyterLab'. The left sidebar displays a file tree with several files, including 'myAuthorlist.csv' which is highlighted with a red box. The main area contains a code cell [12] with the following Python code:

```
#If you receive an error running this cell for the first time, try running it again.  
from scholarly import scholarly  
import csv  
import warnings  
import pandas as pd  
import numpy as np  
import matplotlib  
from matplotlib import pyplot as plt  
from matplotlib import cm as CM
```

Below this, a note says: "2. Modify the names of the input and output files. The name of the input file should match the name of the author list CSV file. If you followed the setup instructions, the CSV file should now be in the same directory as this notebook file. The output file does not have to exist yet (it will be created)."

Cell [3] contains the following code, also with 'author_list_csv' highlighted:

```
[3]: #Input file name of your author list CSV file.  
author_list_csv = 'myAuthorlist.csv'  
output_data_csv = 'ss_output_data.csv'
```

Cell [6] contains the following code:

```
[6]: author_names= []  
with open(author_list_csv, encoding ="utf-8-sig") as csv_file:  
    csv_reader = csv.reader(csv_file, delimiter=',')  
    for row in csv_reader:  
        if (len(row) == 1):
```

Cell [7] contains the following code:

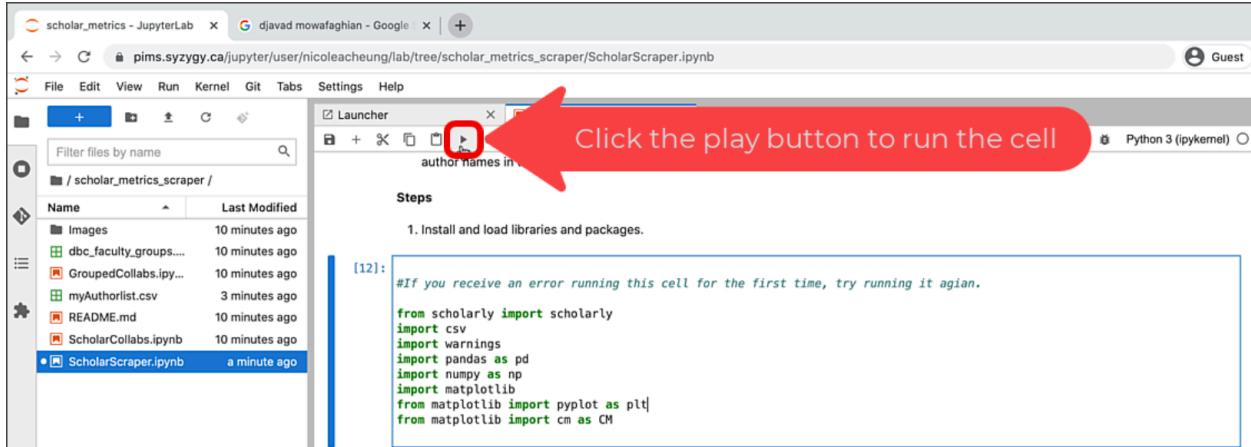
```
[7]: affiliations = ['University of British Columbia', 'UBC', 'Djavad Mowafaghian']
```

Cell [4] at the bottom is partially visible with the text: "4. Modify the affiliations list with institutions which the researchers are affiliated with."

Step 4: Run ScholarScraper (this step continues after step 5)

Run the ScholarScraper notebook code cells 1-7.

- You can run a cell by clicking anywhere inside the cell and either clicking the play button or pressing shift + enter.

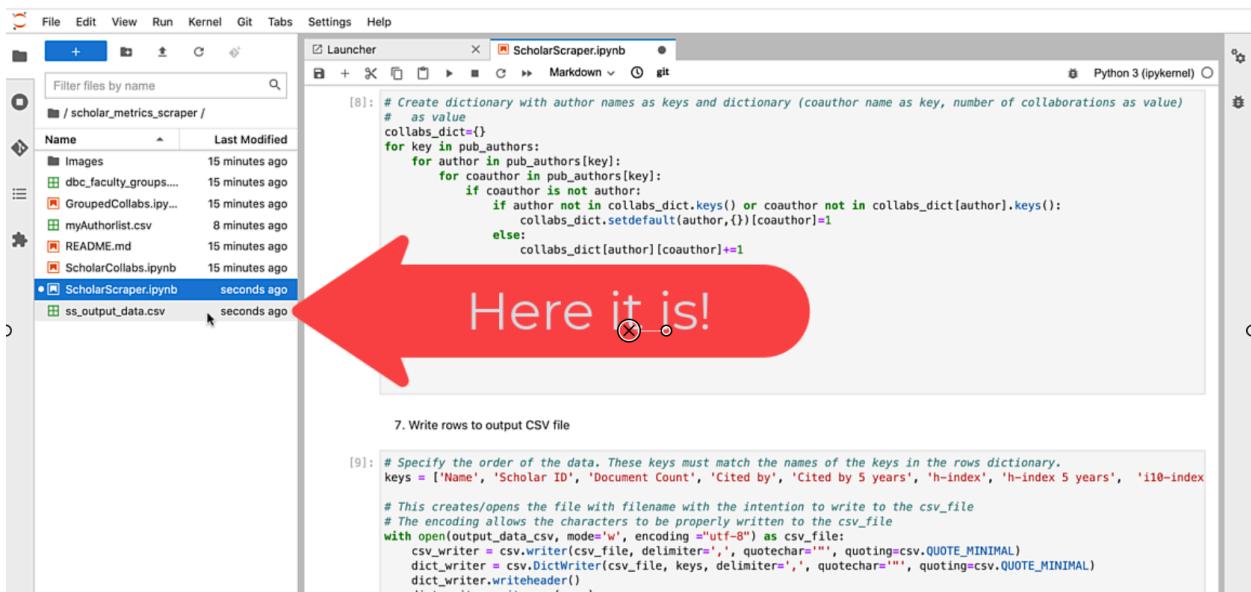


A screenshot of a JupyterLab interface. The left sidebar shows a file tree with several files: Images, dbc_faculty_groups..., GroupedCollabs.ipynb, myAuthorlist.csv, README.md, ScholarCollabs.ipynb, and ScholarScraper.ipynb. The ScholarScraper.ipynb file is selected. The main area displays a code cell numbered [12]:

```
#If you receive an error running this cell for the first time, try running it again.  
from scholarly import scholarly  
import csv  
import warnings  
import pandas as pd  
import numpy as np  
import matplotlib  
from matplotlib import pyplot as plt  
from matplotlib import cm as CM
```

A large red arrow points from the text "Click the play button to run the cell" to the play button icon in the toolbar above the code cell.

- Cell number 5 (which retrieves the data for each author from Google Scholar) may take several minutes.
- After running cell number 7, you should find a CSV saved to your directory with the data for each author. You can click it to view it or right-click to download.



A screenshot of a JupyterLab interface. The left sidebar shows a file tree with several files: Images, dbc_faculty_groups..., GroupedCollabs.ipynb, myAuthorlist.csv, README.md, ScholarCollabs.ipynb, and ScholarScraper.ipynb. A new file, ss_output_data.csv, has been added to the directory. The main area displays a code cell numbered [8]:

```
# Create dictionary with author names as keys and dictionary (coauthor name as key, number of collaborations as value)  
# as value  
collabs_dict={}  
for key in pub_authors:  
    for author in pub_authors[key]:  
        if coauthor is not author:  
            if author not in collabs_dict.keys() or coauthor not in collabs_dict[author].keys():  
                collabs_dict.setdefault(author,{})[coauthor]=1  
            else:  
                collabs_dict[author][coauthor]+=1
```

The cell below, numbered [9], contains code for writing rows to a CSV file:

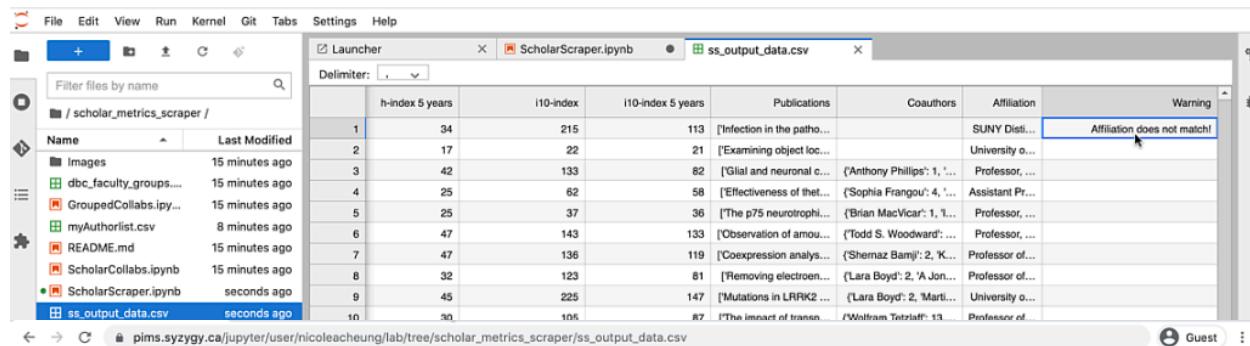
```
7. Write rows to output CSV file  
  
[9]: # Specify the order of the data. These keys must match the names of the keys in the rows dictionary.  
keys = ['Name', 'Scholar ID', 'Document Count', 'Cited by', 'Cited by 5 years', 'h-index', 'h-index 5 years', 'i10-index'  
# This creates/opens the file with filename with the intention to write to the csv_file  
# The encoding allows the characters to be properly written to the csv_file  
with open(output_data_csv, mode='w', encoding="utf-8") as csv_file:  
    csv_writer = csv.writer(csv_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)  
    dict_writer = csv.DictWriter(csv_file, keys, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)  
    dict_writer.writeheader()  
    dict_writer.writerows(rows)
```

A large red arrow points from the text "Here it is!" to the ss_output_data.csv file in the file tree.

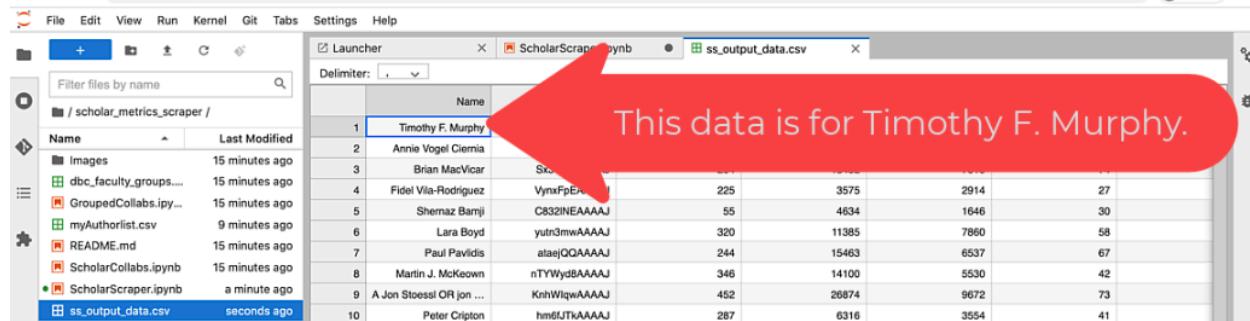
Step 5: Check for warnings

Check the last column of the table for warnings and if necessary, modify your author list CSV.

- Check the last column for warnings - the row for Dr. Tim Murphy has been flagged with a warning. The author data retrieved contains an affiliation that does not match one from the affiliations list.

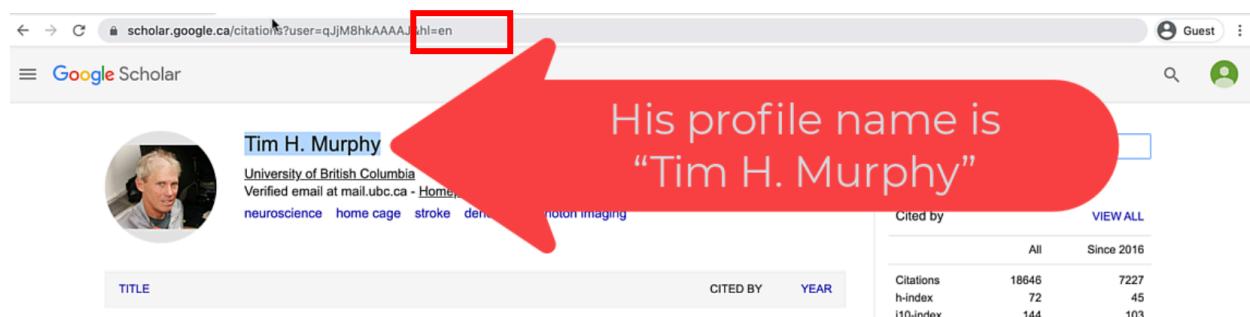


A screenshot of a Jupyter Notebook interface. The top tab bar shows 'File', 'Edit', 'View', 'Run', 'Kernel', 'Git', 'Tabs', 'Settings', and 'Help'. Below the tabs, there's a file browser sidebar with items like 'scholar_metrics_scrapers.ipynb', 'myAuthorlist.csv', and 'ss_output_data.csv'. The main area displays a table titled 'ss_output_data.csv' with columns: 'Name', 'h-index 5 years', 'i10-index', 'i10-index 5 years', 'Publications', 'Coauthors', 'Affiliation', and 'Warning'. The 'Warning' column for the first row (Timothy F. Murphy) contains the text 'Affiliation does not match!'. A red arrow points from the text 'This data is for Timothy F. Murphy.' to the 'Name' column of the table.



This screenshot shows the same Jupyter Notebook interface, but the table data has been filtered or updated. The 'Name' column now lists 'Timothy F. Murphy', 'Annie Vogel Ciernia', 'Brian MacVicar', etc. A red arrow points from the text 'This data is for Timothy F. Murphy.' to the 'Name' column of the table.

- Find the author's Google Scholar profile and make sure you have their correct information. If you don't, you will need to modify your author list. You can either correct their name or replace their name with their Google Scholar ID which can be found in their profile URL (see the red box below).
 - In our case, we attempted to retrieve data for "Tim Murphy", but the retrieval resulted in data from another author named "Timothy F. Murphy". We either need to change the name in the author list to "Tim H. Murphy" or replace it with the Scholar ID (qJjM8hkAAAAJ).



A screenshot of a Google Scholar profile page for 'Tim H. Murphy'. The URL in the address bar is 'scholar.google.ca/citations?user=qJjM8hkAAAAJ&hl=en'. A red box highlights the URL. The profile page shows a photo of a man, his name 'Tim H. Murphy', his affiliation 'University of British Columbia', and a list of publications. A red arrow points from the text 'His profile name is "Tim H. Murphy"' to the name 'Tim H. Murphy' on the profile page.

- Modify the author list. In this case, we changed “Tim Murphy” to “Tim H. Murphy”

A1 X ✓ Tim H. Murphy

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Tim H. Murphy																			
2	Annie Ciernia																			
3	Brian MacVicar																			
4	Fidel Vila-Rodríguez																			
5	Shemraz Bamji																			
6	Lara Boyd																			
7	Paul Pavlidis																			
8	Martin McKeown																			
9	J. Ivan Rosales																			

- Right click to delete the old author list

File Edit View Run Kernel Git Tabs Settings Help

Launcher ScholarScraper.ipynb ss_output_data.csv

Name	Scholar ID	Document Count	Cited by	Cited by 5 years	h-index	h-index 5
1 Timothy F. Murphy	UniVIB8AAAAJ	378	20627	5577	71	
2 Annie Vogel Ciernia	XAGIOIIAAAAAJ	37	1456	1269	17	
3 Brian MacVicar	Sx3420bAAAAAJ	204	18402	7010	74	
4 Fidel Vila-Rodríguez	VynxFpEAAAAAJ	225	3575	2914	27	
5 Shemraz Bamji	C832lNEAAAAAJ	55	4634	1646	30	
Lara Boyd	yun3mwAAAAAJ	320	11385	7860	58	
Paul Pavlidis	ataejQKAAAAAJ	244	15463	6537	67	
n J. McKeown	nTYWydAAAAAJ	346	14100	5530	42	
essl OR jon ...	KnhWIqwAAAAJ	452	26874	9672	73	
Peter Cripton	hm6fUTKAAAAAJ	287	6316	3554	41	
son S. Snyder	B6F64-4AAAAJ	34	5270	2446	18	
Wolfram Tutschaff	Hn1Ivr.IAAAAJ	21418	21418	6059	74	

- Upload the corrected author list

scholar_metrics - JupyterLab

Click here to upload files

File Edit View Run Kernel Tabs Settings Help

Launcher ScholarScraper.ipynb ss_output_data.csv

UBC THE UNIVERSITY OF BRITISH COLUMBIA Dynamic Brain Circuits in Health & Disease Research Excellence Cluster

- Run code cells 1-7 again
- Check the warnings column again to ensure the author’s data has been corrected.

File Edit View Run Kernel Git Tabs Settings Help

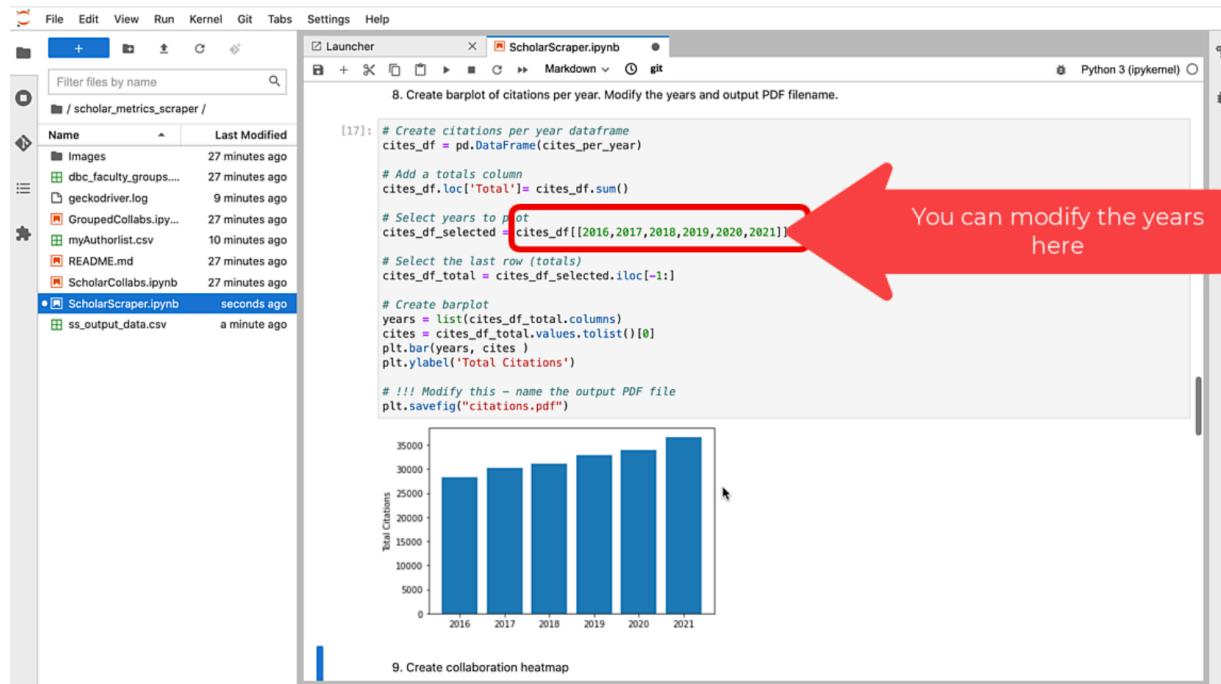
Launcher ScholarScraper.ipynb ss_output_data.csv

	h-index 5 years	i10-index	Validation	Warning
1	45	144	[...]	1
2	17	22	[...]	
3	42	133	[...]	
4	25	62	[...]	
5	25	37	[...]	
6	47	143	[...]	

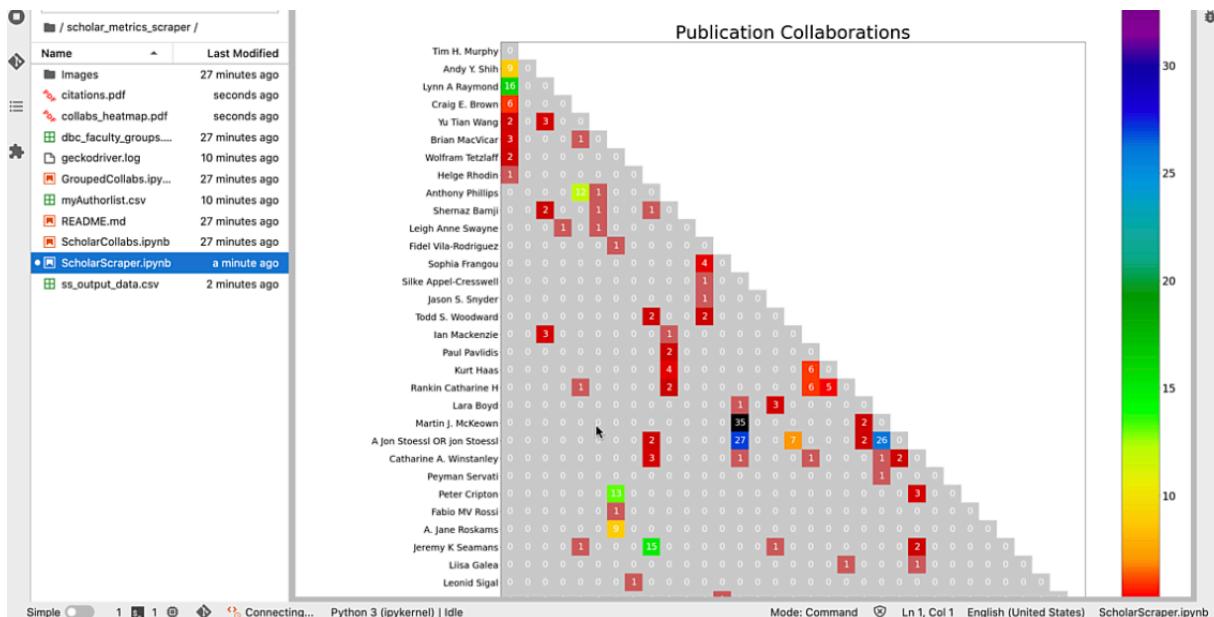
Step 4 (continued)

Run the ScholarScraper notebook code cells 8 and 9.

- Code cell 8 creates a bar chart of total citations/year for the group. The range of years can be modified as shown below.



- Code cell 9 creates a collaboration heatmap, visualizing the number of collaborations between two coauthors

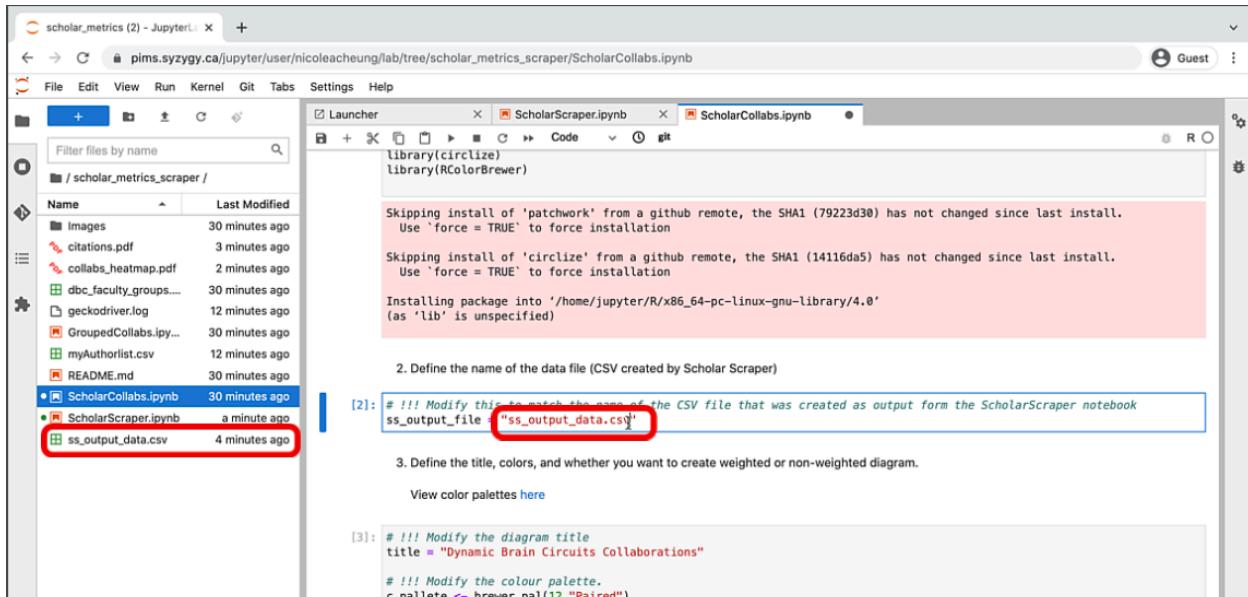


- Both diagrams are saved to the project directory as PDFs for viewing and downloading.

Step 6: Run ScholarCollabs

Open the ScholarCollabs notebook. Make some minor modifications to the notebook before running.

- Make sure the output data name matches the CSV filename.



```
# !!! Modify this to match the name of the CSV file that was created as output from the ScholarScraper notebook
ss_output_file : 'ss_output_data.csv'
```

- Modify the title, colour palette, and set weighted to TRUE if you want a weighted diagram (which weights links by the number of collaborations) or FALSE if you want an unweighted diagram.

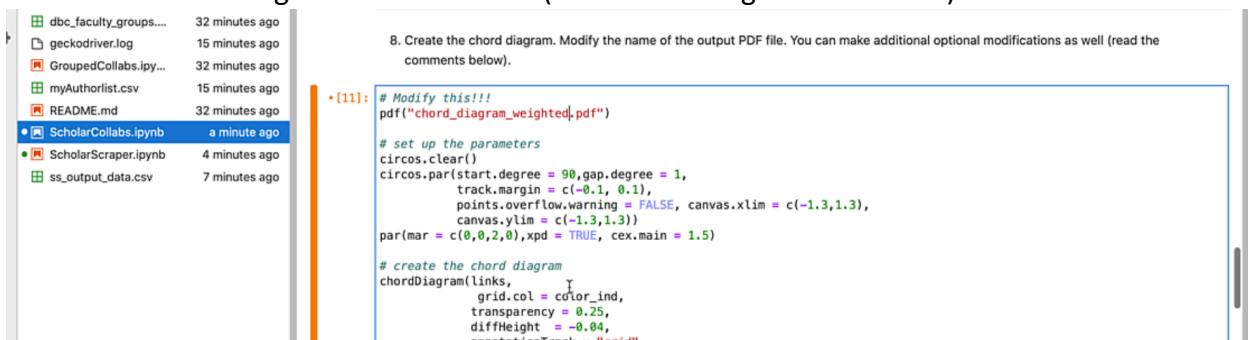


```
# !!! Modify the diagram title
title = "Publication Coauthors"

# !!! Modify the colour palette.
c_palette <- brewer.pal(12,"Paired")

# !!! Modify this - Set to TRUE if you want a weighted diagram or FALSE if you want a non-weighted diagram.
weighted = TRUE
```

- You can change the PDF filename (where the image will be saved) here.



```
# Modify this!!!
pdf("chord_diagram_weighted.pdf")

# set up the parameters
circos.clear()
circos.par(start.degree = 90,gap.degree = 1,
          track.margin = c(-0.1, 0.1),
          points.overflow.warning = FALSE, canvas.xlim = c(-1.3,1.3),
          canvas.ylim = c(-1.3,1.3))
par(mar = c(0,0,2,0),xpd = TRUE, cex.main = 1.5)

# create the chord diagram
chordDiagram(links,
             grid.col = color_idx,
             transparency = 0.25,
             diffHeight = -0.04,
             annotationTrack = "nrid".
```

Run the ScholarCollabs notebook code cells.

- After running the last code cell (8), the diagram will be saved in your project directory.

The screenshot shows a Jupyter Notebook interface. On the left is a file browser showing a directory structure with files like 'Images', 'chord_diagram_weighted.pdf', 'citations.pdf', 'collabs_heatmap.pdf', 'dbc_faculty_groups....', 'geckodriver.log', 'GroupedCollabs.ipynb', 'myAuthorlist.csv', 'README.md', 'ScholarCollabs.ipynb' (selected), 'ScholarScraper.ipynb', and 'ss_output_data.csv'. On the right, two code cells are visible. The top cell contains code for creating a chord diagram with specific axis settings. The bottom cell contains code to add a title and save the diagram as a PDF. A red arrow points from the text 'The diagram is saved as a PDF here' to the file browser area.

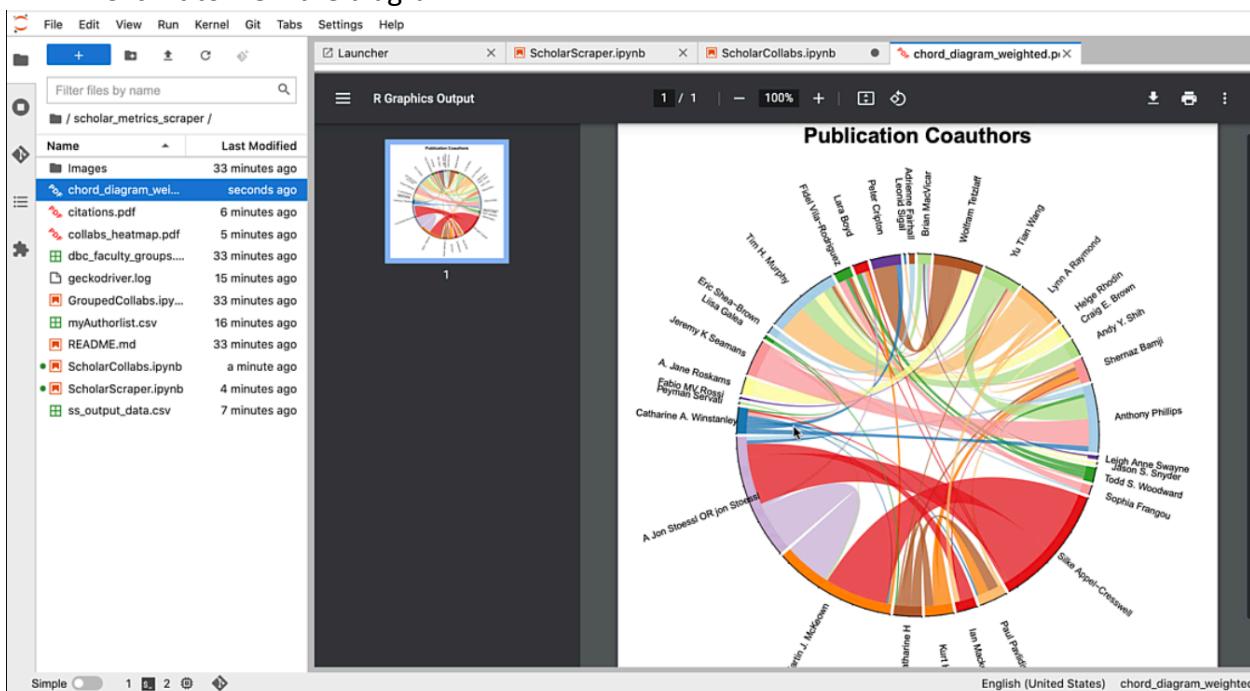
```
# Add graduation on axis
axis("top",
      h = "top",
      line = 0)

# Add the title (user can modify the title in step 3)
title(title, outer=FALSE)

dev.off()

png: 2
```

- Click it to view the diagram.



Step 7: Run GroupedCollabs

Open the GroupedCollabs notebook. You will need an additional CSV file with each author's group to run this notebook.

- Create a CSV file with author names in the first column (red box), group names in the first row (blue box), and author names in their respective rows and under their group's column (green box). Author names will need to match the names in the output data CSV file (and therefore, match their Google Scholar profile names).

	Faculty of Medicine	Faculty of Applied Science	Faculty of Science	Faculty of Arts	Not UBC
2	Tim H. Murphy		Annie Vogel Ciernia		
3	Annie Vogel Ciernia				
4	Brian MacVicar				
5	Fidel Vila-Rodriguez				
6	Shernaz Bamji				
7	Lara Boyd				
8	Paul Pavlidis				
9	Martin J. McKeown				
10	A Jon Stoessl OR jon Stoessl				
11	Peter Cripton				
12	Jason S. Snyder				
13	Vesna Sossi				
14	Wolfram Tetzlaff				
15	Anthony Phillips				
16	Catharine A. Winstanley				
17	Yu Tian Wang				
18	Todd S. Woodward				
19	Jeremy K Seamans				
20	Ian Mackenzie				
21	Lynn A Raymond				
22	Kurt Haas				
23	Mark S. Cembrowski				
24	Fabio MV Rossi				
25	A. Jane Roskams				
26	Rankin Catharine H				
27	Michael Gordon				
28	Leonid Sigal				
29	Peyman Servati				
30	Ulika Galea				
31	Sophia Frangou				
32	Silke Appel-Cresswell				
33	Helge Rhodin				

- Save as CSV and upload to the project directory.

scholar_metrics - JupyterLab

pims.syzygy.ca/jupyter/user/nicoleacheung/lab/tree/scholar_metrics_scrapers/ScholarScrapers.ipynb

File Edit View Run Kernel Take Settings Help

Upload Files

Click here to upload files

/ scholar_metrics_scrapers /

Name	Last Modified
Images	6 minutes ago
dbc_faculty.groups....	6 minutes ago
GroupedCollabs.ipynb...	6 minutes ago
README.md	6 minutes ago
ScholarCollabs.ipynb	6 minutes ago
ScholarScrapers.ipynb	6 minutes ago

THE UNIVERSITY OF BRITISH COLUMBIA
Dynamic Brain Circuits in Health & Disease
Research Excellence Cluster

Python 3 (ipykernel)

Make some modifications to the GroupedCollabs notebook.

- Modify the group_file name to match the CSV filename.

```

2. Define the name of the data file (CSV created by Scholar Scraper), investigator names CSV file, and group CSV file

• [52]: # !!! Modify this to match the name of the CSV file that was created as output from the ScholarScraper notebook
ss_output_file = "ss_output_data.csv"
    ... modify this to match the name of your groupings CSV file. See instructions above.
group_file = "myGroups.csv"

```

- Again, make sure the ss_output_data name matches the CSV file.

```

2. Define the name of the data file (CSV created by Scholar Scraper), investigator names CSV file, and group CSV file

• [52]: # !!! Modify this to match the name of the CSV file that was created as output from the ScholarScraper notebook
ss_output_file = "ss_output_data.csv"
    ... modify this to match the name of your groupings CSV file. See instructions above.
group_file = "myGroups.csv"

3. Define the title, colors, and whether you want to create weighted or non-weighted diagram.

View color options here.

```

- Modify the title, colours (one for each group), and group names. Again, set weighted to TRUE or FALSE depending on whether you want a weighted diagram.

```

• [53]: # !!! Modify the diagram title
title = "Coauthors Grouped by Faculty"

# !!! Modify the colour palette. Make sure there are the same number of colours as groups.
c_palette = c("red","orange","green","cyan","magenta")

# !!! Modify the group names. These groups will be paired with the colours in the c_palette, in the same order.
group_names = c("Medicine",
               "Applied Science",
               "Science",
               "Arts",
               "Cascadia")

# !!! Modify this - Set to TRUE if you want a weighted diagram or FALSE if you want a non-weighted diagram.
weighted = TRUE

```

- Name the PDF where the diagram will be saved to.

```

8. Create the chord diagram. Modify the name of the output PDF file. You can make additional optional modifications as well (read the comments below).

• [58]: # !!! Modify this
pdf("grouped_chord_diagram_weighted.pdf")

# set up the parameters
circos.clear()
circos.par(start.degree = 90,gap.degree = 1,
          track.margin = c(-0.1, 0.1))

```

Run the GroupedCollabs notebook code cells.

- After running the last code cell (8), the diagram will be saved to your project directory.

The diagram is saved as a PDF

```
major.tick.length = 0.1,  
labels.niceFacing = FALSE)  
}  
# Add a legend  
# You can modify the position (e.g. change "bottomright" to "topleft"), and the font (cex)  
legend("bottomright", legend=group_names,  
col=c_pallete, lty=1, cex=0.7)
```

- Click it to view the diagram.

