

Easing script writing on Google Colab with structural biology snippets

Blaine Mooers, PhD

blaine-mooers@ouhsc.edu

405-271-8300

<http://github.com/MooersLab/ACA2022>

Department of Biochemistry & Molecular Biology
University of Oklahoma Health Sciences Center, Oklahoma City

American Crystallographic Association

72nd Annual Meeting

Potland, Oregon

August 1, 2022

First code cell in ColabFold Notebook

CO AlphaFold2.ipynb

File Edit View Insert Runtime Tools Help

Share   

+ Code + Text  Connect Editing

Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: Making protein folding accessible to all. *Nature Methods*, 2022

{x}  Input protein sequence(s), then hit Runtime -> Run all

query_sequence: "PIAQIHIILEGRSDEQKETLIREVSEAIRSLDAPLTSVRVIITEMAKGHFGIGGELASK"

• Use : to specify inter-protein chainbreaks for **modeling complexes** (supports homo- and hetero-oligomers). For example PI...SK:PI...SK for a homodimer

jobname: "test"

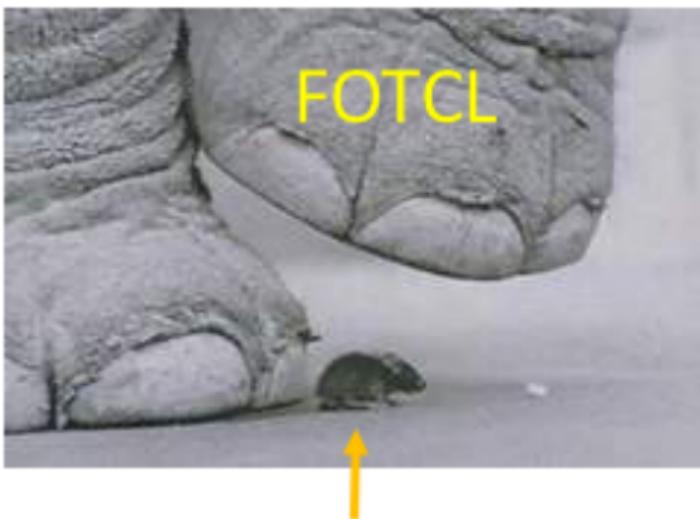
use_amber:

template_mode: none

Mirdita,M., Schütze,K., Moriwaki,Y., Heo,L., Ovchinnikov,S. and Steinegger,M. (2022) ColabFold: making protein folding accessible to all. Nature Methods.

Biologists with FOTCL do not fear ColabFold

FOTCL: Fear Of The Command Line



Command Line

Code Cell in Jupyter Notebook

The screenshot shows a Jupyter Notebook interface. At the top, there's a header with the logo, the notebook name "Untitled3", a Python icon, and a "Logout" button. Below the header is a toolbar with various icons: file operations (Save, New, Cut, Copy, Paste), cell navigation (Up, Down), a "Run" button, and other kernel-related buttons. A "Trusted" badge is visible next to the Python icon. To the right of the toolbar, the kernel name "amber.py37" is shown with a dropdown arrow. The main menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". The "Cell" menu is currently open, displaying options: "Code" (which is checked), "Markdown", "Raw NBConvert", and "Heading". At the bottom, an input cell is visible with the text "In []:".

Markdown and code cells in a Colab Notebook

The screenshot shows a Google Colab notebook interface. At the top, there's a toolbar with a 'CO' logo, the file name 'Untitled2.ipynb', and various menu options: File, Edit, View, Insert, Runtime, Tools, Help, and a dropdown set to 'All ch'. To the right of the menu are 'Comment', 'Share', 'Settings', and a profile icon. Below the toolbar, there's a navigation bar with '+ Code', '+ Text', 'Connect', 'Editing', and other icons. A search bar is also present. The main workspace contains the following content:

- A code cell with the text: `## Below is a code cell.`
- A text cell with the text: **Below is a code cell.**
- A code cell with the text: `[] from pymol import cmd`
- A text cell with the text: **Above is a code cell.**

A large pink arrow points downwards from the bottom of the second code cell towards the third text cell.

Snippet search box in Colab

The screenshot shows a Google Colab notebook titled "Untitled2.ipynb". The interface includes a top navigation bar with File, Edit, View, Insert, Runtime, Tools, Help, and a status bar indicating "Last saved at 12:38 PM". On the left, there are sidebar icons for code, text, and other notebook functions. The main workspace contains two code cells:

- A collapsed code cell labeled "Below is a code cell." containing the code:

```
[ ] from pymol import cmd
```
- An expanded code cell labeled "Above is a code cell." containing the code:

```
python
from pymol.cgo import *
from pymol import *
import math
class Counter:
```

A "Code snippets" sidebar is open on the right, listing various PyMOL-related functions with a "Filter ..." button and an "Insert" button. The listed functions include:

- addAxis
- addAxisipy
- Adding form fi...
- allPairs
- antialias
- ao
- AO
- aobw

Outline

- Installation of colabpymolsnips and PyMOL.
- Use of snippets in Colab.
- Elyra-snippets for JupyterLab.
- Editing Colab with GhostText.

Colab badge for colabpymolpysnips

☰ README.md



colabpymolpysnips:

Library of PyMOL Python snippets for running
Incentive PyMOL in Google Colab notebooks

This notebook requires a PyMOL license file in your Google Drive.

version 0.3

For the notebook that uses the Open Source version of PyMOL, [see](#).

Click on Colab badge to the right to open a notebook on Colab:



<https://github.com/MooersLab/colabpymolpysnips>

Save snippets notebook to Google Drive

colabIncentivePyMOLpySnip03.ipynb

File Edit View Insert Runtime Tools Help

Share

Table of contents X

+ Code + Text Copy to Drive ← 3 Connect

installPyMOL

testPyMOL

{x} ao

sas

ellipcol

sigdist

sigang

<> bs ← 1 stack

bu

doubleBond

installPyMOL

Colab PyMOL snippets library, 7 December 2021
<https://github.com/MooersLab/colabpymolpysnip>

in this notebook and are accessed via <> in the left sidebar. Log in to your Google Drive account and then open the notebook in Google Colab. Grant permission to do so. The URL of the notebook will be copied to your clipboard. Enter this address in the shaded line labeled "Custom snippet notebook URL" in the sidebar. The snippets will always be available when you log in to Google Drive. The snippet called `installPyMOL` copies the PyMOL installation files into the correct position. installs Anaconda, and then runs the `colorh1` function from the `pymolshortcuts.py` file to color protein molecules according to the Eisenberg hydrophobicity scale, scheme 1.

Code snippets × ...

color ← 2

colorh1

colorh2

obliedCartoon

colorh1 Insert

Run the `colorh1` function from the `pymolshortcuts.py` file to color protein molecules according to the Eisenberg hydrophobicity scale, scheme 1.

PyMOL

Prompt for Google Account

```
from IPython.utils import io
import tqdm.notebook
import os
"""
The PyMOL installation is done inside two nested context managers. This approach
was inspired by Dr. Christopher Schlick's (of the Phenix group at
Lawrence Berkeley National Laboratory) method for installing cctbx
in a Colab Notebook. He presented his work on September 1, 2021 at the IUCr
Crystallographic Computing School. I adapted Chris's approach here. It replaces my first approach
that requires seven steps. My approach was presented at the SciPy2021 conference
in July 2021 and published in the
[proceedings](http://conference.scipy.org/proceedings/scipy2021/blaine_mooers.html)
The new approach is easier for beginners to use. The old approach is easier
and could be used as a back-up approach.

I thank Professor David Oppenheimer of the University of Florida for the
"""

total = 100
with tqdm.notebook.tqdm(total=total) as pbar:
    with io.capture_output() as captured:

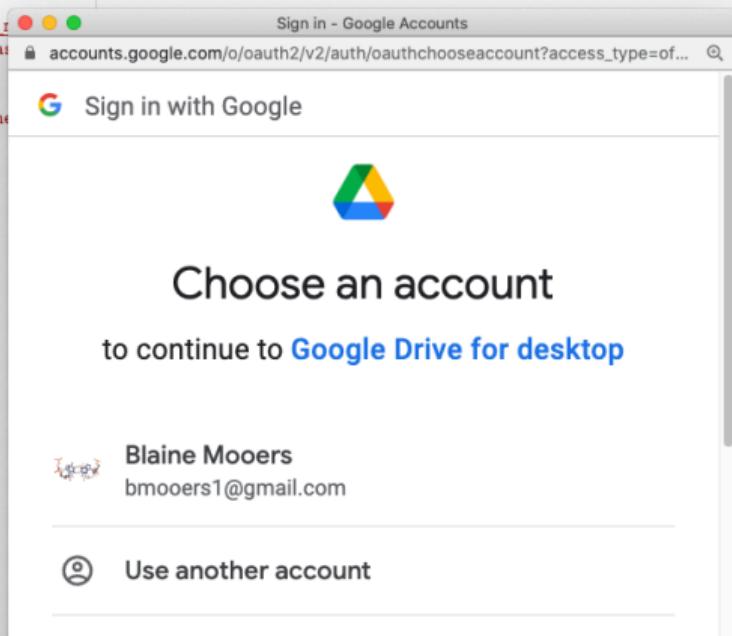
        !pip install -q condacolab
        import condacolab
        condacolab.install()
        pbar.update(10)

        import sys
        sys.path.append('/usr/local/lib/python3.7/site-packages/')
        pbar.update(20)

    # Install PyMOL
    %shell mamba install -c schrodinger pymol-bundle --yes

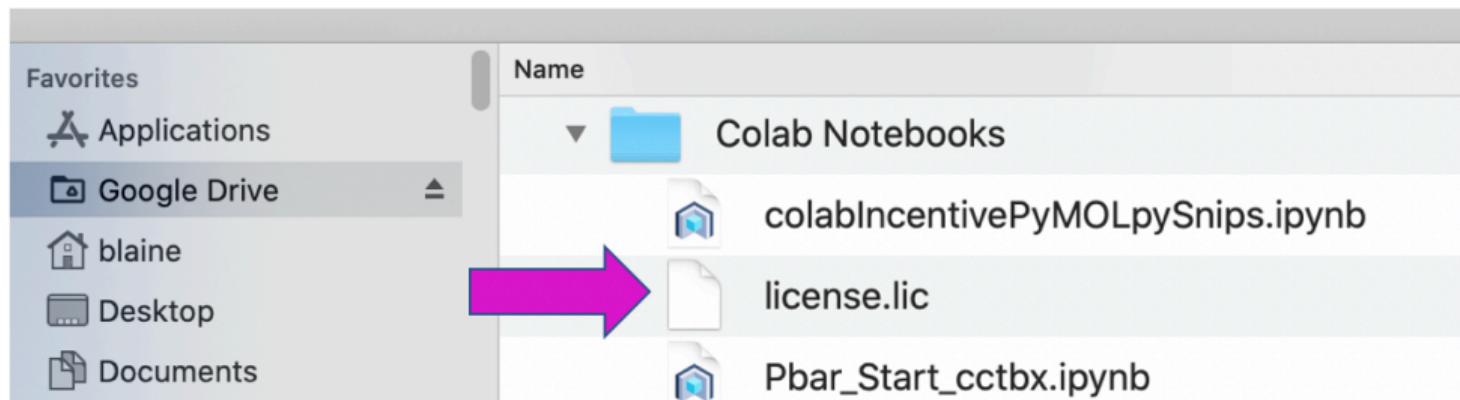
    pbar.update(90)

    # Mount Google Drive and upload your PyMOL license
    from google.colab import drive
    drive.mount("/content/drive")
    # !cp /content/drive/MyDrive/PyMOL/PyMOL_v2.2.2_Linux_x64/PyMOL /usr/local/bin/
```



'Colab Notebooks' directory on Google Drive

Use to store the snippet notebooks and PyMOL license.



Navigate to Tools->Settings->Site

The screenshot shows the Jupyter Notebook interface. On the left, there's a sidebar with icons for file operations like New, Open, Save, and Run. The main area has a code cell above and a text cell below. The code cell contains the Python command `[] from pymol import cmd`. The text cell contains the text "Below is a code cell." Below the text cell is a play button icon.

The top navigation bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a status message "All changes saved". A dropdown menu is open under the Tools tab, listing Command palette (with keyboard shortcut `⌘/Ctrl+Shift+P`), Settings (which is highlighted and grayed out), Keyboard shortcuts (with keyboard shortcut `⌘/Ctrl+M H`), and Diff notebooks.

To the right of the main workspace, there are several panels: a Comment panel with a comment icon, a Share panel with a share icon, a Pymol logo icon, a "Code snippets" panel containing "AO", another "AO" panel, an "ao" panel, and an "ACPW" panel. The "AO" panel has an "Insert" button. A descriptive text at the bottom right says: "Run the AO function from the pymolshortcuts.py file to generate the photorealistic effect."

Above is a code cell.

Below is a code cell.

```
[ ] from pymol import cmd
```

Run the AO function from the
pymolshortcuts.py file to generate the
photorealistic effect.

Paste in URL

Notebook will have URL in the colab.research.google.com domain.

Settings

The screenshot shows the Google Colab settings interface. On the left, there's a sidebar with tabs: Site (selected), Editor, Colab Pro, GitHub, and Miscellaneous. The main area has two dropdown menus: 'Theme' set to 'adaptive' and 'Default page layout' set to 'horizontal'. Below these are two checkboxes: 'Show desktop notifications for completed executions' (checked) and 'New notebooks use private outputs (omit outputs when saving)' (unchecked). At the bottom, there's a section for 'Custom snippet notebook URL' with two entries, both preceded by a red 'X': https://colab.research.google.com/drive/1HPDde7YFsjUbVr_f... and <https://colab.research.google.com/drive/1JC6bwa7rma0-07b...>.

Site

Editor

Colab Pro

GitHub

Miscellaneous

Theme
adaptive

Show desktop notifications for completed executions

New notebooks use private outputs (omit outputs when saving)

Default page layout
horizontal

Custom snippet notebook URL

×

https://colab.research.google.com/drive/1HPDde7YFsjUbVr_f...

×

https://colab.research.google.com/drive/1JC6bwa7rma0-07b...

Old way of installing PyMOL on Colab

Run seven code blocks.

The screenshot shows the Google Colab interface. On the left, there is a sidebar titled "Code snippets" with a search bar containing "pymol". Below the search bar, there are several items listed:

- Imports for pymol
- Install PyMOL in new Colab notebook
- Test pymol setup with T4L
- Install GraphViz & [PyDot](https://pypi.python.org/pypi/pydot)
- aod

Below this, there is a detailed description of the "Install PyMOL in new Colab notebook" snippet:

Whenever a notebook is shutdown, the installed software is lost. You have to re-install pymol. The pymol-bundle package is not available for Python3.6.

1. Mount your Google Drive in Colab
2. Copy pymol license from your Google Drive to the current directory.
3. Install Miniconda.
4. Update Miniconda.
5. Add site-packages to Path.
6. Install PyMOL
7. Import cmd from pymol.

At the bottom of the sidebar, there is a code block:from google.colab import drive
drive.mount("/content/drive")

```
in the main area, there are seven code blocks:
```

+ Code + Text Connect E
X
from google.colab import drive
drive.mount("/content/drive")
[] !cp ./drive/My\ Drive/Colab\ Notebooks/license.lic .
[] #!bash
MINICONDA_INSTALLER_SCRIPT=Miniconda3-4.5.4-Linux-x86_64.sh
MINICONDA_PREFIX=/usr/local
wget https://repo.conda.io/miniconda/\$MINICONDA_INSTALLER_SCRIPT
chmod +x \$MINICONDA_INSTALLER_SCRIPT
./\$MINICONDA_INSTALLER_SCRIPT -b -f -p \$MINICONDA_PREFIX
[] #!bash
conda install --channel defaults conda python=3.6 --yes
conda update --channel defaults --all --yes
[] import sys
_ = (sys.path
 .append("/usr/local/lib/python3.6/site-packages"))
[] !conda install -c schrodinger pymol --yes
[] from pymol import cmd
from IPython.display import Image

Mooers (2021) Modernizing computing by structural biologists with Jupyter and Colab. Proc. of the 20th Python in Science Conf. p. 14-22.

New way to install PyMOL with one mouse click

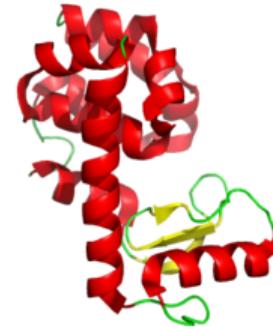
```
with tqdm.notebook.tqdm(total=total) as pbar:  
    with io.capture_output() as captured:  
  
        !pip install -q condacolab  
        import condacolab  
        condacolab.install()  
        pbar.update(10)  
  
        import sys  
        sys.path.append('/usr/local/lib/python3.7/site-packages/')  
        pbar.update(20)  
  
        # Install PyMOL  
        %shell mamba install -c schrodinger pymol-bundle --yes  
  
        pbar.update(90)  
  
        # Mount Google Drive and upload your PyMOL license  
        from google.colab import drive  
        drive.mount("/content/drive")  
        %shell cp ./drive/MyDrive/Colab\ Notebooks/license.lic .  
        pbar.update(100)
```



220/? [03:36<00:00, 1.07it/s]

Test of PyMOL installation

```
cmd.reinitialize()
cmd.fetch('3fe0', type='pdb')
cmd.orient()
cmd.turn('z', '-90')
cmd.turn('y', '5')
cmd.turn('x', '10')
cmd.hide('everything')
cmd.bg_color('white')
cmd.show('cartoon')
cmd.color('red', 'ss H')
cmd.color('yellow', 'ss S')
cmd.color('green', 'ss Lt')
cmd.set_view('(-0.18,-0.69,-0.7,0.98,-0.17,-0.09,-0.06,-0.7,0.71,0.0,0.1)
cmd.png("1lw9.png")
Image(filename = "1lw9.png", unconfined=True)
```



✓ 1s completed at 3:49 PM

<https://github.com/MooersLab/pymolshortcuts>

Mooers (2020) Shortcuts for faster image creation in PyMOL. Protein Sci.29(1):268-276.

How to add a new snippet in Colab

The screenshot shows the Google Colab interface with a sidebar on the right containing a list of code snippets. A snippet titled "Install cctbx (1/2)" is selected. The main content area displays the code for installing cctbx, followed by a section for "Install cctbx (2/2)". A snippet for "Install cctbx (2/2)" is also present in the sidebar.

+ Code + Text

Reconnect | Editing | ⌂

Code snippets × ***

install

installPyMOL +
Install [cartopy](http://scitools.org.uk/cartopy/docs/latest/) +
Install cctbx (2/2) +
Install cctbx (1/2) +
Install GraphViz & [PyDot](https://pypi.python.org/pypi/pydot) +
Install 7zip reader [libarchive](https://pypi.python.org/pypi/libar... +
Install the snippet notebook +
Importing a library that is not in Colaboratory +
rvi +

Install cctbx (1/2) Insert

This procedure is adapted from Billy Poon's example: https://github.com/phenix-project/Colabs/blob/main/Start_cctbx.ipynb

The installation of cctbx involves a two step process:

1. Install condacolab
2. Install cctbx with mamba and move certain libraries to better locations.

The module condacolab install conda and mamba. Mamba is faster. A progress bar will be updated as the snippet runs.

Completion of the first step will involve the generation of the error message:

Your session crashed for an unknown reason.

Click on the X to close the error message before running **Install cctbx (2/2)**

```
[ ] from IPython.utils import io
      import tqdm.notebook

      total = 50
      with tqdm.notebook.tqdm(total=total) as pbar:
          with io.capture_output() as captured:
              !pip install -q condacolab
              pbar.update(10)

          import condacolab
          condacolab.install()
          pbar.update(40)
```

↳ Install cctbx (2/2)

Use mamba to install cctbx-base.

[] ↳ 1 cell hidden

```
[ ] from IPython.utils import io
```

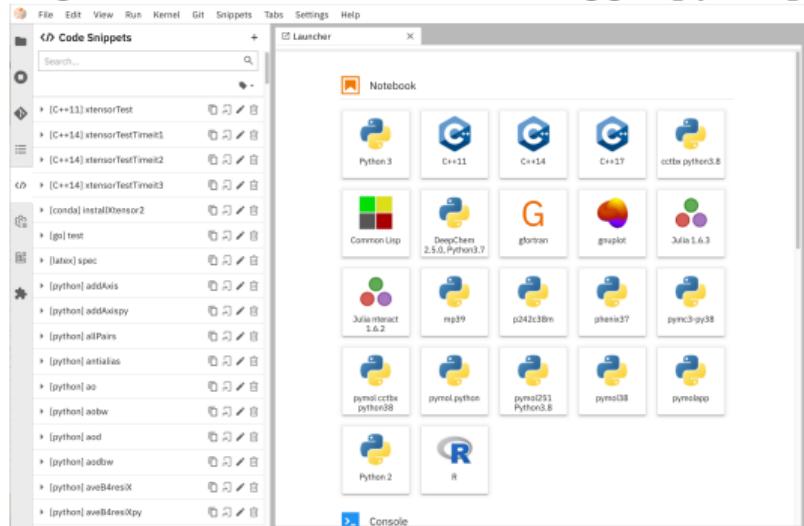
<https://github.com/MooersLab/colabcctbxsnips>

Mooers (2021) Running CCTBX and PyMOL in the same Jupyter Notebook. Computational Crystallography Newsletter 12, 26-32.

elyra-code-snippet-extension in Jupyter Lab

elyra-code-snippet-extension

<https://github.com/MooersLab/taggedpymolpysnips>



For pulldown-menu libraries for Jupyter, see Mooers (2021) A PyMOL snippet library for Jupyter to boost researcher productivity. Computing in Science and Engineering. 23(2) 47-53.

Elyra-snippet for AO: ambient occlusion

The screenshot shows the Elyra IDE interface with a code snippet for ambient occlusion. The snippet is named "[python] ao" and contains the following Python code:

```
cmd.do('set_color oxygen,  
[1.0,0.4,0.4];')  
cmd.do('set_color nitrogen,  
[0.5,0.5,1.0];')  
cmd.do('remove solvent;')  
cmd.do('as spheres;')  
cmd.do('util.cbaw;')  
cmd.do('bg white;')  
cmd.do('set light_count,10;')  
cmd.do('set spec_count,1;')  
cmd.do('set shininess, 10;')  
cmd.do('set specular,0.25;')  
cmd.do('set ambient,0;')  
cmd.do('set direct,0;')  
cmd.do('set reflect,1.5;')  
cmd.do('set ray_shadow_decay_factor,  
0.1;')  
cmd.do('set ray_shadow_decay_range,  
2;')  
cmd.do('set depth_cue, 0;')  
cmd.do('ray;')
```

GUI for making new Elyra-snippets

<https://github.com/MooersLab/taggedpymolpysnips>

The screenshot shows a Jupyter Notebook interface with a floating dialog box titled "New Code Snippet". The dialog is for "Add new Code Snippet". It contains fields for "Name*" (with a placeholder), "Description", "Tags" (with an "Add Tag +" button), and "Source" sections for "Language*" (with a placeholder) and "Code *". A vertical sidebar on the left lists recent files. At the bottom is a "SAVE & CLOSE" button.

Untitled.ipynb New Code Snippet

Add new Code Snippet

All fields marked with an asterisk are required. [Learn more ...]

Name *

Description

Tags

Add Tag +

Source

Language *

Code *

SAVE & CLOSE

Caveats to snippets in notebooks

- Only conda base env on Colab.
- No support for tab triggers.
- No support for tab stops.
- No scope for snippets.

Editing Colab cell with Neovim via GhostText

Federico Brigante: <https://github.com/fregante/GhostText>

Keyboard shortcut Shift-Cmd-k

because the prior objects will need to be deleted.

```
from pymol import cmd
from IPython.display import Image
cmd.fetch("1lw9")
cmd.orient("1lw9")
cmd.rotate("z", "270")
cmd.png("test.png")
Image(filename = "test.png", unconfined=True)
```

Neovim

```
4/4 1 Startify 2 [Scratch] 3 [Scratch]
1 from pymol import cmd
2 from IPython.display import Image
3 cmd.fetch("1lw9")
4 cmd.orient("1lw9")
5 cmd.rotate("z", "270")
6 cmd.png("test.png")
7 Image(filename = "test.png", unconfined=True)
8
```

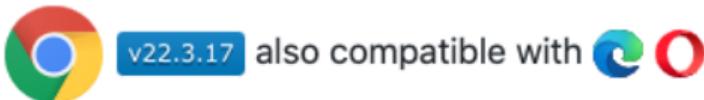
Need pymolpysnips library for UltiSnips

<https://github.com/MooersLab/pymolsnips>

<https://github.com/MooersLab/DSW22ghosttext>

GhostText is available for popular text editors

1. Install your browser extension:



2. Install your editor extension:



Works in code blocks in Jupyter Notebooks and Tex files in Jupyter Lab. Also works in Overleaf.

Summary

- Colab notebooks use snippet libraries stored on Google Drive.
- One-click installation of PyMOL via installPyMOL snippet.
- Elyra-snippets for Jupyter Lab resemble Colab snippets.
- GhostText enables editing of active code blocks.

Acknowledgements

- Billy Poon for help with cctbx
- Oklahoma Data Science Workshop

Funding:

- NIH: R01 CA242845, R01 AI088011
- NIH: P30 GM145423, P30 CA225520, P30 AG050911
- OCAST HR20-002
- PHF Team Science Grant with Ian Dunn