

PyMOL in RStudio

Blaine Mooers. Dept. of Biochemistry and Molecular Biology, University of Oklahoma Health Sciences Center, Oklahoma City, OK, USA 73104
blaine-mooers@ouhsc.edu

Introduction

- Literature programming weaves prose, computer code, and output from that code to tell a story.
- **Literature programming** is used in scientific computing and data science to support **reproducible computations** in research.
- Electronic notebooks allows readers of literate programming documents to re-run the code.
- The snippet library *pymolpysnips* is used with PyMOL's Python API to run PyMOL in electronic notebooks.
- RMarkdown are easy to assemble in RStudio, an IDE for R.
- Snippet libraries support the assembly of literate programming documents.

Methods

- Rewrote the *pymolpysnips* library (Mooers and Brown 2021) in Python.
- Added tab stops at sites of parameters values that may need to change.
- Formatted the snippet libraries for RMarkdown electronic notebooks.
- Describe the library on a GitHub webpage (<https://mooerslab.github.io/rstudio/pymolpysnips/>).

- Share the library at <https://github.com/MooersLab/rstudio/pymolpysnips>

Results

Technology Stack

Technology	Version	Description
PyMOL	2.5.1	Molecular graphics program
Python from Anaconda	3.6–3.9	Programming language
conda from Anaconda	4.10.1	Program used to create Python environments.
R	4.0.5	R is required to host Rstudio.
RStudio	1.4.1717	The Integrated Development Environment for R.
reticulate	1.20	Provides the interface between Python and R.
git	2.25.1	Eases the downloading and updating of the libraries.

Snippet library in Rstudio.

```
1 # reticulate
2 snippet reticulate
3 ``{r}
4 library(reticulate)
5 use_condaenv("r-reticulate")
6 ``$0
7 # python code block
8 snippet pythonCodeBlock
9 ``{python}
10 ``${1:enter code here}
11 ...
12 # Apply the ambient occlusion effect to get the pi
13 snippet ao
14 cmd.do("set_color oxygen, [1.0,0.4,0.4];")
15 cmd.do("set_color nitrogen, [0.5,0.5,1.0];")
16 cmd.do("remove solvent;")
17 cmd.do("as spheres;")
18 cmd.do("# the \"as\" command is a shortcut for si
19 cmd.do("util.cbaw;")
20 cmd.do("# cbaw" represents "color by atom".
21 cmd.do("# The last letter represents the colore
22 cmd.do("bg white;")
23 cmd.do("bg is an alias for bg_color or backgro
24 cmd.do("set light_count,10;")
25 cmd.do("# light_count is the number of light sou
26 cmd.do("# The max is 10. The default is 10.")
27 cmd.do("set spec_count,1;")
28 cmd.do("# Not documented on Wiki.")
29 cmd.do("set shininess, 10;")
30 cmd.do("# sets the shininess of the object.")
```

Conclusions

The *rstudiotypmolpysnips* library supports molecular graphics in electronic notebooks. This is a step towards literate programming in structural biology.

Selection of the ao snippet in-line.

```
43
44 45 ``{python}
46 ao
47 aobw {snippet}
48 aod {snippet}
49 aodbw {snippet}
50 ao {snippet}
51 ``{python}
52 ``{r}
53 ``{codebraid}
```

The screenshot shows the RStudio interface with a Python code chunk containing PyMOL commands. A yellow selection box highlights the line 'ao' in the code, indicating it is being selected or edited. The RStudio environment pane on the right shows variables like 'cmd', 'pymol', and 'pymol2'.

ao snippet in Python code chunk.

```
44
45 ``{python}
46 cmd.do("set_color oxygen, [1.0,0.4,0.4];")
47 cmd.do("set_color nitrogen, [0.5,0.5,1.0];")
48 cmd.do("remove solvent;")
49 cmd.do("as spheres;")
50 cmd.do("util.cbaw;")
51 cmd.do("bg white;")
52 cmd.do("set light_count,10;")
53 cmd.do("set spec_count,1;")
54 cmd.do("set shininess, 10;")
55 cmd.do("set specular,0.25;")
56 cmd.do("set ambient,0;")
57 cmd.do("set direct,0;")
58 cmd.do("set reflect,1.5;")
59 cmd.do("set ray_shadow_decay_factor, 0.1;")
60 cmd.do("set ray_shadow_decay_range, 2;")
61 cmd.do("set denth_cue 0;")
```

The screenshot shows the RStudio interface with the same Python code chunk. A circular cursor icon is over the 'Run' button, indicating the code is ready to be executed. The RStudio environment pane shows the variable 'cmd'.

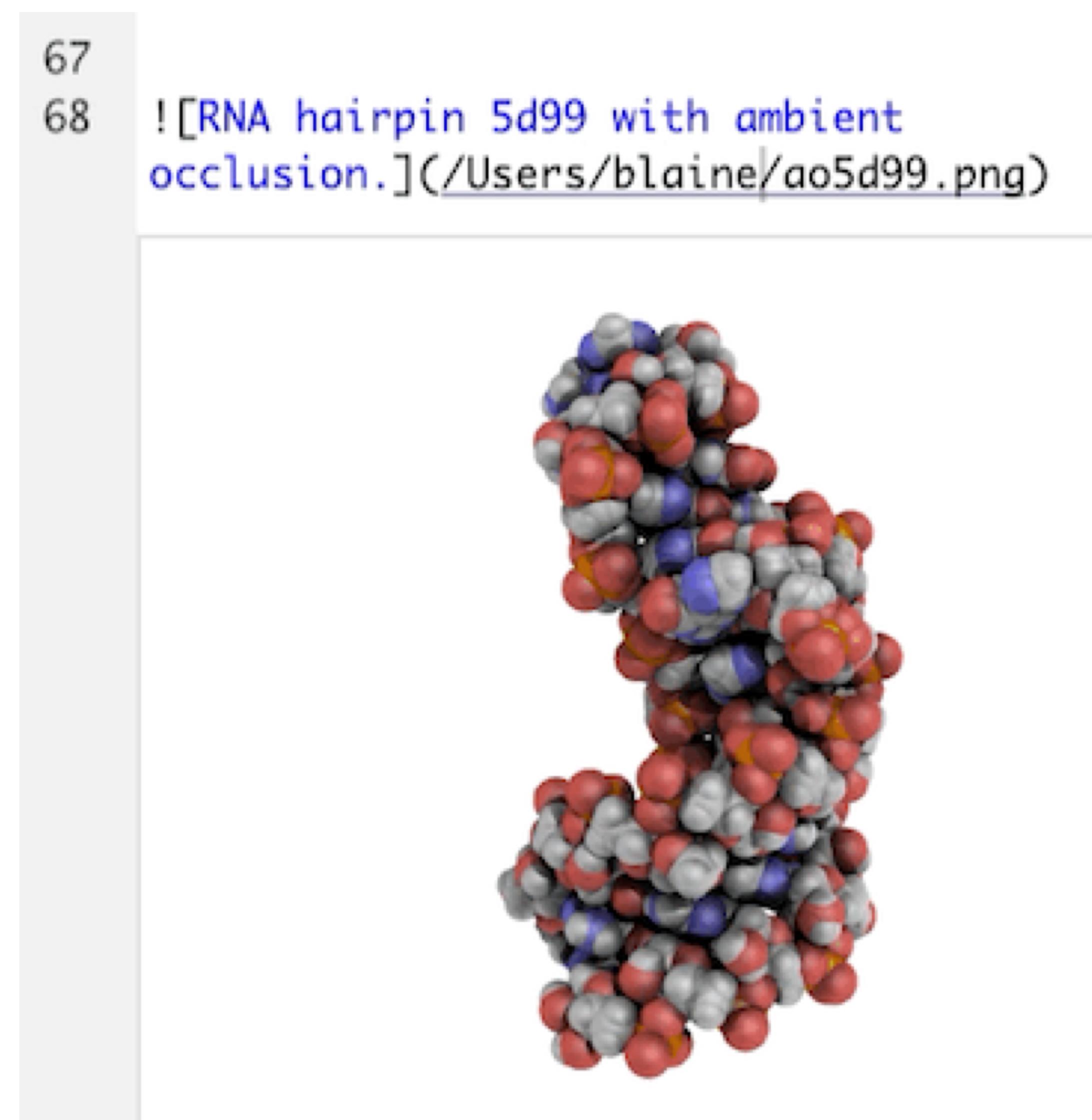
ao code in Python REPL.

```
RETICULATE 1.20 REPL -- A PYTHON INTERPRETER IN R.
PyMOL>set_color oxygen, [1.0,0.4,0.4];
Color: "oxygen" defined as [ 1.000, 0.400, 0.400 ].
PyMOL>set_color nitrogen, [0.5,0.5,1.0];
Color: "nitrogen" defined as [ 0.500, 0.500, 1.000 ].
PyMOL>remove solvent;
Remove: eliminated 183 atoms in model "5d99".
PyMOL>as spheres;
PyMOL>util.cbaw;
PyMOL>bg white;
```

Literature Cited

Mooers, B.H.M. and Brown, M. E. (2021) Templates for writing PyMOL scripts. Protein Sci. 30(1):262-269. doi:10.1002/pro.3997. PubMed PMID: 33179363.

Image of RNA hairpin loaded.



PDF after running knitr on R Notebook.

Running PyMOL from R Notebook

```
library(reticulate)
use_condaenv("r-reticulate")
from pymol import cmd
cmd.do("reinitialize")

## PyMOL not running, entering library mode (experimental)
cmd.do("fetch 6VXX")
cmd.do("zoom (resi 614 and chain A)")
cmd.label(selection="chain A and resi 614 and name CB", expression="%s-%s % (resn,resi)")
cmd.do("set label_color, black; set label_size, 48")
cmd.do("set stick_radius, 0.12")
cmd.do("hide cartoon; show sticks")
cmd.do("set ray_shadows, 0")
cmd.do("draw")
cmd.do("png /Users/blaine/D614Greticulate.png, 800, 500, dpi=300")
```

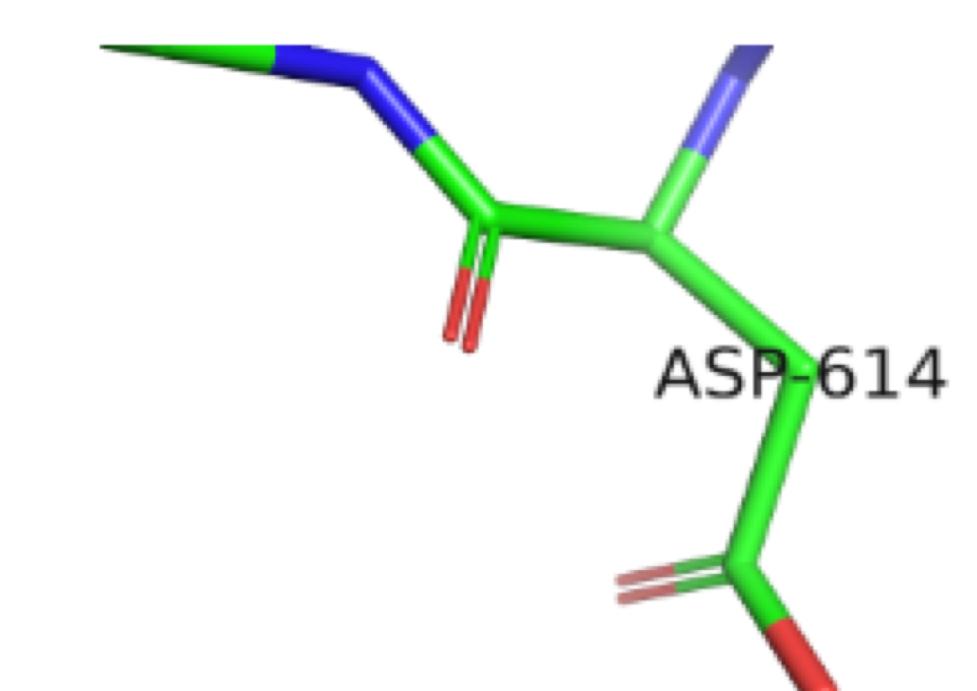


Figure 1: Coronavirus SARS2 Spike Protein site 61.

Acknowledgements

- Warren Delano Memorial Open Source Software Fellowship
- NSF MCB-1616845
- NIH R01 CA242845
- NIH P20 GM103640