PyRosetta Jupyter Notebooks Teach Biomolecular Structure Prediction and Design

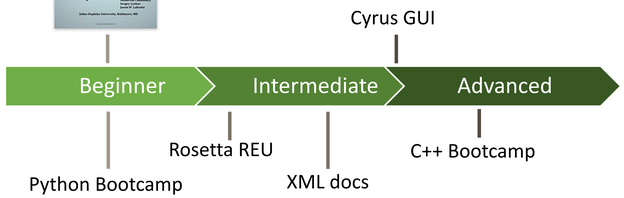
Kathy H. Le[1], Jared Adolf-Bryfogle[2], Jason C. Klima[3,4], Sergey Lyskov[1], Jason Labonte[1,5], Steven Bertolani[6], Shourya S. Roy Burman[1], Andrew Leaver-Fay[7], Brian Weitzner[3,4], Jack Maguire[7], Ramya Rangan[8], Matt A. Adrianowycz[8], Rebecca F. Alford[1], Aleexsan Adal[1], Morgan L. Nance[1], Yuanhan Wu[8], Jordan Willis[2], Daniel Kulp[8], Rhiju Das[9], Roland L. Dunbrack, Jr.[10], William Schief[2], Brian Kuhlman[7], Justin B. Siegel[6], Jeffrey J. Gray[1*]

[1]Johns Hopkins University, [2]The Scripps Research Institute, [3]University of Washington, [4]Lyell Immunopharma, [5]Franklin & Marshall College, [6]University of California, Davis, [7]University of North Carolina at Chapel Hill, [8]Wistar Institute, [9]Stanford University, [10]Fox Chase Cancer Center, *Corresponding author: jgray@jhu.edu

Need for Interactive Teaching Material

Finding ways to train people in molecular modeling is a longstanding problem, but certain challenges remain.

100,000+ downloads of print material
12,000+ lecture video views in ~4 years



Pedagogical Challenges

- Need for multimodal engagement (Sankey, et al., 2010)
- Need material for active learning (Berge, 2002)
- Self-paced learning vs. classroom setting

Technical Challenges

- Need for constantly-updated material
- Personal computer incompatibility
- Difficult for students to independently set up software

Solution: Hands-on learning with PyRosetta

Goal: to add compelling and relevant multimedia to the set of Jupyter Notebook workshops

Links:
GitHub Repository
Preprint

Jupyter Notebooks:
Shareable web application that supports live code, equations, visualization, and text

Google Colaboratory:
Free Jupyter Notebook environment that requires **no setup** and runs entirely **on the cloud**

Students can access the PyRosetta workshops in Google Colaboratory

1. Download PyRosetta package and workshops from GitHub Repository.
2. Follow instructions in Chapter 1 to configure PyRosetta with Google Colaboratory (~12 min).
3. Access PyRosetta Notebooks in Google Colaboratory:

Topics in The New Notebooks

Part I: Introduction

1. How to Get Started
2. Intro to PyRosetta
3. Rosetta Energy Score Function
4. Intro to Folding
5. Structure Refinement
6. Intro to Packing & Design
7. Protein Docking
8. Ligand Docking
9. Loop Modeling

Part II: Advanced

10. Working with Symmetry
11. Working with Density
12. Working with Antibodies
13. Carbohydrates
14. RNA Basics
15. Membrane Modeling
16. Running PyRosetta in Parallel

Keyword Index and ToC are automatically generated with nbpages tool

nbpages

Jupyter Notebook Workshops

Pose Basics

Keywords: pose_from_pdb(), sequence(), cleanATOM, annotated_sequence()

Keyword Index

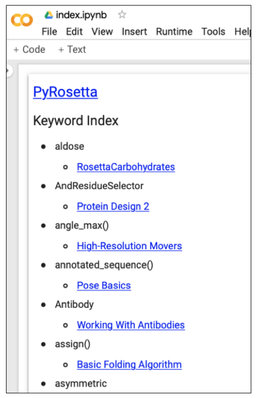
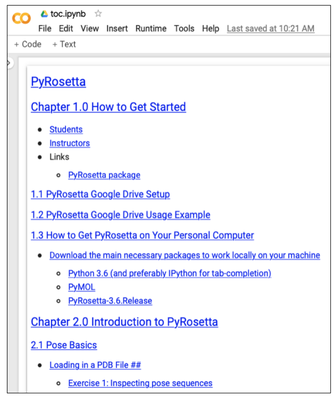


Table of Contents



Student version Notebooks are automatically generated with nbgrader tool

Instructor version → nbgrader → **Student version**

Step 1: Random Move

For the **random** trial move, write a subroutine to choose one residue at random using `random.randint()` and then randomly perturb either the φ or ψ angles by a random number chosen from a Gaussian distribution. Use the Python built-in function `random.gauss()` from the `random` library with a mean of the current angle and a standard deviation of 25°. After changing the torsion angle, use `pmm.apply(polyA)` to update the structure in PyMOL.

nbgrader

```
[ ] import math
    import random

    def randTrial(your_pose):
    ### BEGIN SOLUTION
        randNum = random.randint(2, your_pose.total_residue())
        phiorpsi = random.randint(1)
        if random.randint(0,1) == 0:
            currPhi = your_pose.phi(randNum)
            newPhi = random.gauss(currPhi, 25)
            your_pose.set_phi(randNum,newPhi)
        else:
            currPsi = your_pose.psi(randNum)
            newPsi = random.gauss(currPsi, 25)
            your_pose.set_psi(randNum,newPsi)
        pmm.apply(your_pose)
    ### END SOLUTION
        return your_pose
```

Written solution

Step 1: Random Move

For the **random** trial move, write a subroutine to choose one residue at random using `random.randint()` and then randomly perturb either the φ or ψ angles by a random number chosen from a Gaussian distribution. Use the Python built-in function `random.gauss()` from the `random` library with a mean of the current angle and a standard deviation of 25°. After changing the torsion angle, use `pmm.apply(polyA)` to update the structure in PyMOL.

```
[ ] import math
    import random

    def randTrial(your_pose):
    # YOUR CODE HERE
    raise NotImplementedError()
        return your_pose
```

Omitted solution

Step 2: Scoring Move

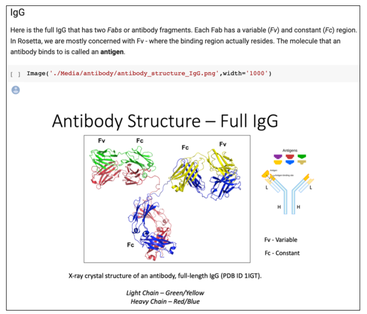
For the **scoring** step, we need to create a scoring function, which we will use to obtain the numerical energy score of the pose.

```
[ ] #sfxn = ??
    # YOUR CODE HERE
    raise NotImplementedError()
```

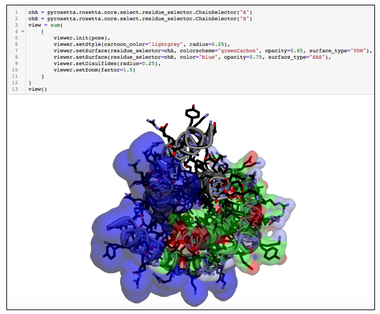
Multimedia examples in PyRosetta Notebooks

Moving GIF from Chapter 5.01: High-Resolution Movers

Schematic from Chapter 12: Working with Antibodies



Schematic from Chapter 2.08: Visualization and pyrosetta.distributed.viewer



Summary

Hands-on learning with PyRosetta:

1. Google Drive-based
   - Familiar to most students
   - No local machine setup necessary
2. Adaptable for class or independent learning
3. Interactive exercises and multimedia
4. Easily expandable by adding more notebooks

Links:
GitHub Repository
Preprint

Acknowledgements

We would like to thank the JHU Center for Educational Resources, NIH, NSF, and Rosetta Commons for funding and support. We would also like to thank Jeffrey Kantor for his advice and insight.