

Pyshifts User Guide

Jingru Xie

May 30, 2017

Contents

1	Introduction	2
1.1	Overview	2
1.2	Copyright Notice	2
2	Installation	3
2.1	Requirements	3
2.2	Installation guide	3
3	Getting started with Pyshifts	4
3.1	Layout	4
3.2	Global parameter	5
3.3	Test case	5
4	Options (Setup)	8
4.1	PyMOL selection/object	8
4.2	Mode	8
4.2.1	Using predictor	8
4.2.2	Using external predicted chemical shifts file	8
5	Error Analysis	9
5.1	Error Table	9
5.1.1	Sort table: Nuclei and Metric	9
5.2	Chemical Shifts Table	10
5.2.1	Sort table	10
5.3	Save to file	10
6	Advanced options	11
6.1	Scale Differences	11
6.2	Offset	11
6.3	Accuracy File	11
6.4	Rendering options	11

1 Introduction

1.1 Overview

PyShifts is a PyMol based graphical analysis tool that utilize chemical shifts to assess the global quality of NMR structures of RNA. Pyshifts takes structure and measured chemical shifts file as input, using various predictors to predict chemical shifts for the structure (or different states in an ensemble if input is an ensemble), comparing it to measured chemical shifts, analyze and visualize difference. Or, it can take external predicted chemical shifts, comparing it to measured chemical shifts and visualize difference.

Codes and resources are freely available at [out github site](#). For more information on theoretical basis as well as our promising test results, see [manuscript](#).

Pyshifts is tested on Mac OS X El Captain, the following manual uses Mac OS X as example.

1.2 Copyright Notice

The PyMOL Plugin source code is copyrighted, but you can freely use and copy it as long as you don't change or remove any of the copyright notices.

This PyMOL Plugin is Copyright©2016 by Jingru Xie (jingrux@umich.edu) and Aaron T. Frank (afrankz@umich.edu).

All Rights Reserved.

2 Installation

2.1 Requirements

Pyshifts is a plugin in PyMOL, an open source Python-enhanced molecular graphics tool. Python of version 2.7.10 and PyMOL are REQUIRED for Pyshifts. Larmord package is also required to enable chemical shifts prediction.

2.2 Installation guide

1. Python

Python version of 2.7.10 (and 2.7.10 only), which is freely available at <https://www.python.org/downloads/release/python-2710/>.

If your current Python version is not 2.7.10, and you prefer not to change it, you can use the following commands to create a Python 2.7.10 environment temporarily for Pyshifts:

```
conda create -n pyshifts python=2.7.10
source activate pyshifts
```

After each use of Pyshifts, use

```
source deactivate pyshifts
```

to go back to your normal Python settings.

2. PyMOL

You can obtain PYMOL at [sourceforge](#).

3. Adding Pyshifts to PyMOL

Download or clone this git repository. Open PyMOL and then go to "Plugin" → "Plugin manager" → "Install new plugin", and choose the Pyshifts.py file in your local Pyshifts repository. For this step PyMOL need to be run with the Tcl/Tk interface, read more on [PyMOL wiki](#).

4. Get Lamord package

Larmord can be obtained from [University of Michigan Tech Transfer](#), free of charge if not for commercial use. You also have to set LARMORD_BIN path in your environment. For example, if the path to Larmord package is */Software/LarmorD_New - master/*, you should do

```
vi ~/.bashrc
```

in command line and insert the following line:

```
export LARMORD_BIN=/Software/LarmorD\_New-master/bin
```

3 Getting started with Pyshifts

3.1 Layout

Once you have Pyshifts successfully installed in PyMOL, you can see it under "Plugin" menubar. Click to run.

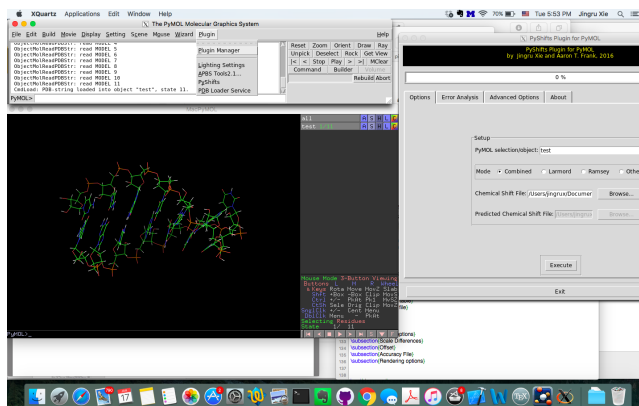


Figure 1: Run Pyshifts from PyMOL.

A new window will pop up. This is the main page of Pyshifts. Basic options will show in main page. At the top of the window is a progress bar. When running, progress bar will show the progress of your current operation. Under progress bar is the main menu bar. By clicking on those tabs you can switch back and forth among the four tabs: "Options" (default), "Error Analysis", "Advanced Options" and "About".

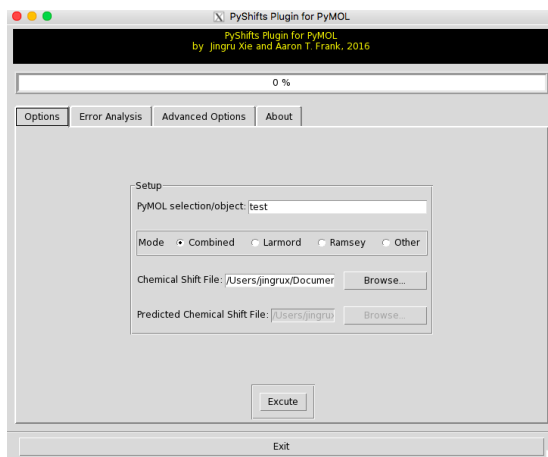


Figure 2: Pyshifts default window.

3.2 Global parameter

Pyshifts use LarmorD as one of its predictor. Before we start working on test case, first you have to set file path to LarmorD parameters correctly. To do this, create a file "pymolrc" under your root folder

```
vi ~/.pymolrc
```

Under insert mode, add a line to that file

```
cd /Users/jingrux/Documents/GitSoftware/Pyshifts
```

3.3 Test case

To start with, let's first have a try on our test case. Test case is an NMR solution structure of a 14-mer hairpin RNA with cUUCGg tetraloop [PDB: 2KOC](#). The structure coordinates (2KOC_test.pdb), reference chemical shifts (measured_shifts_2KOC.dat) and an example predicted chemical shifts file (predicted_shifts_2KOC.dat) are all included in "test" folder which comes with the original package. Or you can always download the "test" folder from github site <https://github.com/atfrank/PyShifts>.

First, load structure into PyMOL as "test". In PyMOL command line, type in

```
load test/2KOC_test.pdb, test
```

A structure will show in PyMOL window. This is frame of 2KOC, our test case.

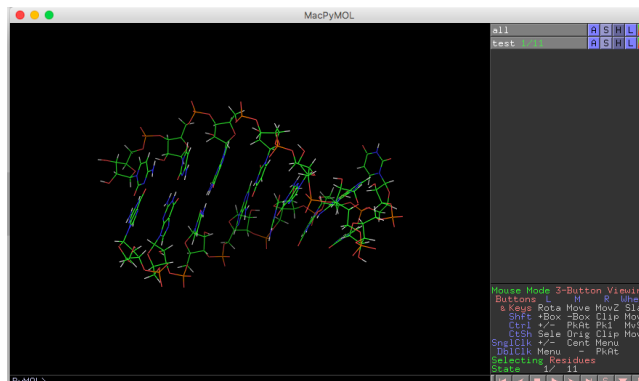


Figure 3: Test ensemble in PyMOL.

In Pyshifts window, find the corresponding file and enter in "Chemical Shifts File" entry. The measured chemical shifts for 2KOC is provided as measured_shifts_2KOC.dat in "test" folder.

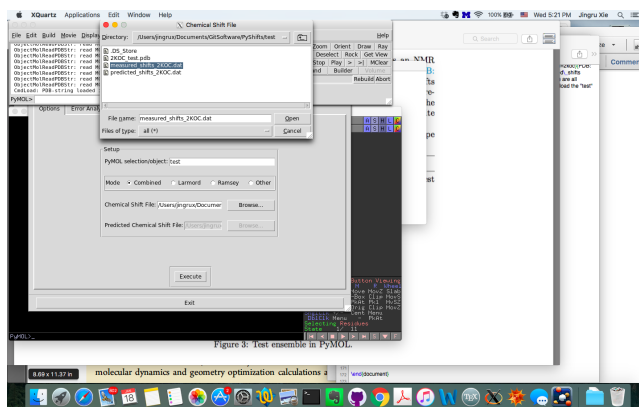


Figure 4: Test measured chemical shifts file

Click on 'execute' button. It may take some time for Pyshifts to complete computing. The progress bar will show your progress in percentage.

Then go to "Error Analysis" tab, click on "Compare shifts". Error Table on the left will be filled with state number and their corresponding MAE (mean absolute error), Pearson coefficient, RMSE (Root mean square error). We will talk about it later in "Error Analysis" section. The table is by default sorted in MAE ascending order.

At the meantime the first few states on top of error table will show in PyMOL. Error for each residues are shown in spheres of different size and color.

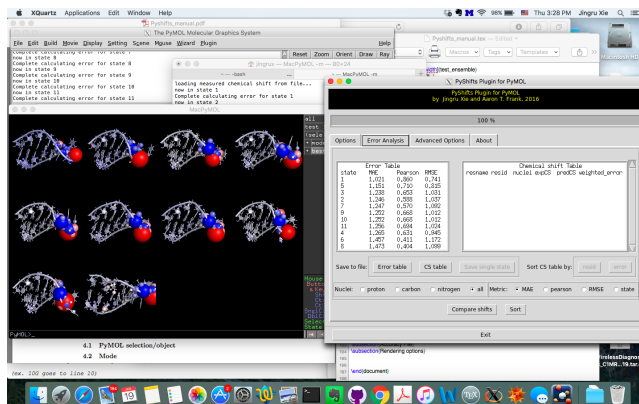


Figure 5: Test error analysis overview

Then double click on one of the state in "Error Table", "Chemical shift Table" will be filled with all the individual residue in your selected state. Each residual is specified by residue name (resname), residue id (resid), nuclei, experimental chemical shifts (expCS) given by the reference file, predicted chemical

shifts (predCS) given by built-in predictor, and weighted error (weighted_error) between them.

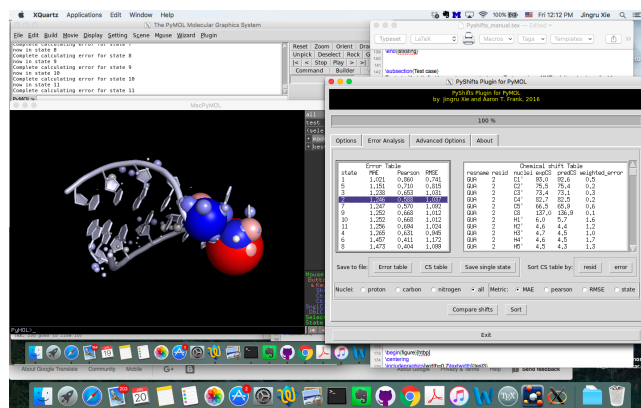


Figure 6: Test Chemical shift table and a single state in PyMOL

At the meantime the state you chose will be enlarged in PyMOL window, showing details of that state.

If you double click on one row in chemical shift, that specific residual will also be highlighted in PyMOL.

In the following three sections, detailed configuration and usage will be introduced.

4 Options (Setup)

4.1 PyMOL selection/object

PyMOL selection/object should be a valid name of a Pymol object that has been loaded. For example if you load a 2LPS.pdb file into PyMOL and name the object as "batman", then you should use "batman" in PyMOL selection/object entry (rather than 2LPS).

4.2 Mode

4.2.1 Using predictor

Pyshifts provided you with two predictors and three modes: LarmorD, Ramsey or a combination of them. By default Pyshifts use combined mode to realize a high accuracy of prediction. In this mode, both LarmorD and Ramsey are used for prediction, and an average of them will be taken for analysis.

In each of the three modes, only one chemical shifts file is needed. The bottom line "Predicted Chemical Shift File" will be disabled.

4.2.2 Using external predicted chemical shifts file

The last mode in Mode selection, "Other" is somehow different from all other modes. In this mode, Pyshifts will not run predictor, rather, it will use the predicted chemical shifts data you provided.

In this mode, two chemical shifts files should be provided: one measured chemical shifts, one set of chemical shifts for all states as reference, same as before; and another predicted measured chemical shifts file, which contains the predicted chemical shifts for each residue in each mode. The bottom line "Predicted Chemical Shift File" will be put into use.

An example predicted chemical shifts file for 2KOC is also given in "test" folder. You can play around with "Other" mode in our test case using this file.

5 Error Analysis

5.1 Error Table

Error Table shows the difference between measured predicted chemical shifts of each state based on three metrics: MAE, Pearson matrix coefficient and RMSE. For a structure with N nuclei, MAE and RMSE are computed as follows:

$$MAE = \frac{1}{N} \sum_{n=1}^N |measuredCS[i] - predictedCS[i]|$$

$$RMSE = \frac{1}{N} \sum_{n=1}^N (measuredCS[i] - predictedCS[i])^2$$

Where *measuredCS[i]* (*predictedCS[i]*) denotes measured (predicted) chemical shift for the *i*th nucleus. Pearson coefficient is computed using Python function *scipy.stats.pearsonr*.

By default the summation goes over all kinds of nuclei in the structure (provided that nucleus has chemical shifts data given). You can also make Pyshifts only calculate the error of either carbons, hydrogens or nitrogens. This can be achieved using Nuclei selection.

5.1.1 Sort table: Nuclei and Metric

Error table can be sorted based on a combination of different nuclei and metric by changing selections in the lower row. When Nuclei selection in selections

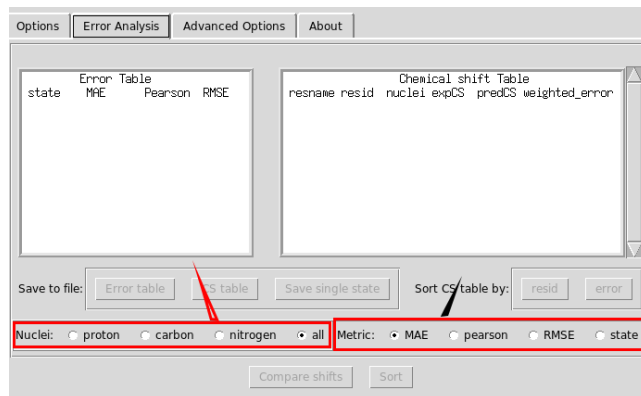


Figure 7: Sort error table based on different nuclei and metrics

other than "all", only selected group of residues will be used to calculate error and only shown in chemical shift table and Pymol.

Note after you change your selection of nuclei or metric, use "sort" button to realize change.

5.2 Chemical Shifts Table

Chemical shifts (both predicted and measured) and "weighted error" are displayed in chemical shifts table for each individual nuclei in your selected state.

Weighted error is calculated based on the difference between predicted and measured chemical shifts and predictor accuracy:

$$weighted_error[i] = \frac{predictedCS - measuredCS}{accuracy[i]}$$

Predictor accuracy can be customized by providing an accuracy file, or you can choose not to weight the errors in advanced options. (See next chapter)

5.2.1 Sort table

Chemical shift Table can be sorted by residue id ascended or in error descending order.

5.3 Save to file

Both error table and chemical shift table can be saved to a .dat file. Tables should have been filled in Pyshifts before they can be saved. If you only want to save the chemical shifts data of your selected state, use "Save single state".

6 Advanced options

6.1 Scale Differences

Predictor yield different errors for different nuclei. "Scale Differences" take predictor error into account. When in "Yes", errors used for analysis will be weighted error (absolute error divided by weight, which is given as accuracy file, or default setting in code).

6.2 Offset

"Offset" option can shift the error of a certain kind of nuclei, to account for chemical shifts offset that always exist in NMR measurements.

6.3 Accuracy File

Optional. Weight used to scale difference is given in accuracy file. When not given, a default LarmorD accuracy list will be used.

6.4 Rendering options

1. Colors

Colors used to render error spheres in Pymol. Default setting is "blue_white_red", which will show negative error as blue, positive as red, while zero error is white.

2. Sphere size

A scale of how large the error sphere will show in Pymol. Depending on the weighted error of you specific case, this option should be adjusted to give the best visualizing performance. Default setting is 0.4.

3. No. models to display

Number of models to display. How many models to display in Pymol after each Comparing shifts and sorting behavior. Default is 10.