Bioinformatics, 34(10), 2018, 1784–1785
doi: 10.1093/bioinformatics/bty021
Advance Access Publication Date: 11 January 2018
Applications Note

OXFORD

Structural bioinformatics

PyChimera: use UCSF Chimera modules in any Python 2.7 project

Jaime Rodríguez-Guerra Pedregal* and Jean-Didier Maréchal*

InsiliChem, Departament de Química, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on September 25, 2017; revised on December 11, 2017; editorial decision on January 8, 2018; accepted on January 9, 2018

Abstract

Motivation: UCSF Chimera is a powerful visualization tool remarkably present in the computational chemistry and structural biology communities. Built on a C++ core wrapped under a Python 2.7 environment, one could expect to easily import UCSF Chimera's arsenal of resources in custom scripts or software projects. Nonetheless, this is not readily possible if the script is not executed within UCSF Chimera due to the isolation of the platform. UCSF ChimeraX, successor to the original Chimera, partially solves the problem but yet major upgrades need to be undergone so that this updated version can offer all UCSF Chimera features.

Results: PyChimera has been developed to overcome these limitations and provide access to the UCSF Chimera codebase from any Python 2.7 interpreter, including interactive programming with tools like IPython and Jupyter Notebooks, making it easier to use with additional third-party software.

Availability and implementation: PyChimera is LGPL-licensed and available at https://github.com/insilichem/pychimera.

Contact: jaime.rodriguezguerra@uab.cat or jeandidier.marechal@uab.cat

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Python is one of the most used programming languages in science due to its readability, smooth learning curve and excellent availability of packages dedicated to data analysis and numeric computing. A lot of scientific software is being either developed in, converted to or wrapped under Python, most of the time as open-source projects, which leads to higher reproducibility and collaborative efforts.

This is also true in molecular modeling, either for structural biology, computational chemistry or anything in between like visualization tools [PyMol (Sanner, 1999)], protein-ligand docking software [AutoDock Vina (Trott and Olson, 2010)], molecular dynamics simulations [OpenMM (Eastman *et al.*, 2013)] or trajectory analysis [MDTraj (McGibbon *et al.*, 2015)]. UCSF Chimera (Pettersen *et al.*, 2004) is one peculiar project that falls within several categories at the same time: a powerful visualization engine that also doubles as a versatile molecular modeling toolkit thanks to advanced features like multiformat input parsers and output writers, hydrogen bonds detection, analysis of steric clashes or hydrophobic interactions, construction of peptides, energy minimization or trajectory analysis. UCSF Chimera is written in both C++ and Python and distributed as a stand-alone application that ships its own versions of Python and other packages to achieve a robust, fail-proof behavior in every major platform. Although this ensures that every user will get the same experience no matter their system environment, it also creates a barrier for those who want to call UCSF Chimera functions from their Python scripts by forcing the user to run it from within UCSF Chimera's process.

In bigger projects, it can also slow down the development process. For example, configuring an Integrated Development Environment or text editor can represent a real *tour de force* if you want to use autocompletion, automated testing or inline documentation. In addition, most modern Python development tools are not distributed by default with UCSF Chimera, and installing them with the proper version can be struggling. For example, latest UCSF Chimera version (1.12) ships its own version of NumPy (van der Walt *et al.*, 2011), but is not