Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Application Note

Structural bioinformatics

Lemon: a modern C++ tool for the rapid development of structural benchmarking datasets

Jonathan Fine[1], and Gaurav Chopra[1*]

[1]Department of Chemistry, Purdue University, 720 Clinic Drive, West Lafayette, Indiana, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The protein data bank (PDB) currently holds over 140,000 biomolecular structures and continues to release new structures on a weekly basis. The PDB is an essential resource to the structural bioinformatics community to develop software that use, categorize, and analyze such data. New computational biology methods are evaluated using custom benchmarking sets derived as subsets of 3D experimentally determined structures from the PDB. Currently, these benchmarking subsets are manually curated with custom scripts developed in a non-standardized manner that results in slow distribution and updates with new experimental structures. Finally, there are scarcity of tools available to rapidly query 3D descriptors of the entire PDB to derive subsequent benchmarking sets in a standard manner.

Approach: Our solution is, Lemon, a C++11 framework that provides a methodology for selecting biomolecules based on precalculated and user derived 3D descriptors. This framework can parse and characterize the entire PDB in less than twenty minutes on modern, multithreaded hardware. The speed in parsing is obtained by using the recently developed MacroMolecule Transmission Format (MMTF) to reduce computational load of reading PDB files. We have incorporated C++ lambda functions that offer extensive flexibility to analyze and categorize the PDB by allowing the user to write custom functions to suite their objective. We think Lemon will become a one-stop-shop to quickly mine the entire PDB to generate custom benchmarking datasets for the entire scientific community. The Lemon software is available as a C++ header library along with example functions at https://github.com/chopralab/lemon.

Contact: gchopra@purdue.edu

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

1 Introduction

Experimental structures deposited in the Protein Data Bank (PDB) (Rose *et al.*, 2015) has resulted in several advances for structural and computational biology scientific and education communities. Several software packages have been developed that use and apply data available in the PDB.

Computational structural biology methods are evaluated using several benchmarking datasets mined from the PDB. As one example, for protein-ligand docking, the Astex (Hartshorn *et al.*, 2007), PDBbind (Liu *et al.*, 2017), and DUD-E (Mysinger *et al.*, 2012) sets have been used to predict the 3D coordinates of ligands, rank target activity, and discriminate binders from non-binders. The process for developing these benchmarking sets is non-standard, time consuming and computationally challenging as it requires significant computational resources to mine different 3D descriptors in the PDB to produce benchmarking datasets for other software packages.

The Macro Molecular Transmission Format (MMTF) (Bradley *et al.*, 2017) was recently introduced to significantly reduce the time required to parse text-based formats traditionally used to store crystallographic data. MMTF requires a fraction of the computation time to read

multiple files into computer memory. Hence, we developed our rapid benchmarking framework to leverage the capabilities of this new PDB format such that the entire PDB can be analyzed in minutes.

Here, we present a novel benchmarking framework that is both flexible and fast for the community to develop new benchmarking datasets quickly and in a standard manner. Such standardized functions will be distributed as modifications to Lemon or part of each investigator's software packages that enhances automation and standardization to develop unique benchmarking datasets.

2 Materials and methods

The work flow of our framework uses modern C++ guidelines and practices to offer a simple pipeline that is both modular and intuitive. The user provides a C++ lambda function that accepts two arguments: an object representing the macromolecular complex, and a string representing the current PDBID. Lemon evaluates this lambda function for all PDBs in a given subset defined by the user in a multithreaded manner. This allows one to perform any calculation on the structural information encoded by the MMTF file (**Figure S1**).

The MMTF object given to the user contains biomolecular data at the atomic, residue, and molecular level. This includes the position, name, element type, and charge of the biomolecule (protein) atoms as well as the name, chain, biologic assembly placement, chemical linkage and composition type of the biomolecular residues. These features can be used to create Lemon workflows which selects and extract the desired 3D interactions.

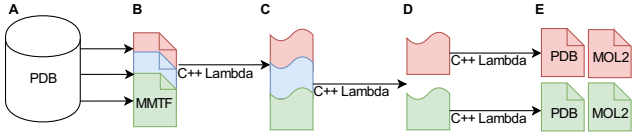

Fig. 1. Workflow for Lemon. The overall work follow for the Lemon framework is given. The user provides C++ Lambda functions which use predefined functions to query information about each complex to filter the PDB into a desired subset.

The recommended work flow is given in **Figure 1** above. It begins with either the entire PDB database or a user defined query generated on the RCSB website (**Figure 1A**). This subset is used for the 'selection' step (**Figure 1B**), where the user selects residues present in the complex for further analysis. Functions already exists in Lemon for selecting small-molecules, metals, nucleic acids, and amino acids that work on the residue level by querying the residue's size and composition type.

Given the 3D environment of the residues and their arrangement in a macromolecules, the next step in the work flow allows the user to perform queries on selected residues (**Figure 1C**). Structures that fail to contain residues of interest after the selection process are discarded. Lemon also provides functions to remove biologically identical residues, remove common residues (see **Table S1-S2**), and keep or remove residues based on their relationship to other residues arranged in 3D (**Figure 1D**).

The final step in the work flow is writing the complexes that fulfill the queries in file formats supported by other popular bioinformatics tools (**Figure 1E**). Example lambda functions for querying features of interest are given in **Listings S1-S10**.

3 Results

3.1 Querying the Protein Data Bank takes minutes

To measure our program's execution time, we ran each of the example functions provided in the supporting information. The first four listings show Lemon's ability to query residue level information (it's biologic assembly and residue name, and residue size/charge). The remaining six listings showcase Lemon's ability to find interacting residues using distance cutoffs.

Each category example was completed within 30 minutes (**Table S3**) and the summary of the queries are shown in **Figures S2-S3** and **Tables S4-S5.**

3.2 Code availability

Lemon is hosted on GitHub, which includes a CMake based build system, example programs for creating simple benchmarking sets, and unit tests to test example functions (see **'Obtaining Lemon'** in Supplementary data). File input and output is provided by the Chemfiles library.

Acknowledgements

We would like to thank Gerardo Tauriello and Daniel Farrell for developing the MMTF-CPP library and Guillaume Fraux for developing the Chemfiles library.

Funding

This work has been supported by the Purdue Instructional Innovation Award, Purdue Research Foundation and start up award to Gaurav Chopra.

Conflict of Interest: none declared.

References

Bradley,A.R. *et al.* (2017) MMTF—An efficient file format for the transmission, visualization, and analysis of macromolecular structures. *PLOS Comput. Biol.*, **13**, e1005575.

Hartshorn,M.J. *et al.* (2007) Diverse, high-quality test set for the validation of protein-ligand docking performance. *J. Med. Chem.*

Liu,Z. *et al.* (2017) Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Acc. Chem. Res.*, **50**, 302–309.

Mysinger,M.M. *et al.* (2012) Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.*, **55**, 6582–6594.

Rose,P.W. *et al.* (2015) The RCSB Protein Data Bank: Views of structural biology for basic and applied research and education. *Nucleic Acids Res.*, **43**, D345–D356.