A Data-engineering Pipeline for Deep Learning in Structural Bioinformatics

Eli J. Draizen [1,2], Cameron Mura [1,2], and Philip E. Bourne [1,2]

[1]Department of Biomedical Engineering, University of Virginia, Charlottesville, VA [2]School of Data Science, University of Virginia, Charlottesville, VA

Introduction

Machine learning has a rich history in structural bioinformatics, and modern approaches such as deep learning are already revolutionizing our knowledge of the subtle relationships between biomolecular sequence, structure, function & evolution. The advances are enabled by large volumes of data that, ideally, are intelligible or manipulable. In structural bioinformatics, such data often relate, directly or indirectly, to protein 3D structures. A significant & recurring challenge concerns the creation of large, high-quality, openly accessible, and reproducible datasets that can be used for specific training/benchmarking tasks, e.g. in predictive modeling projects. Here, we report a protein biophysical and evolutionary featurization and data-processing pipeline that we recently developed and deployed (both in the cloud and on local HPC resources) in order to systematically and reproducibly create comprehensive, superfamily-level domain databases for deep learning tasks (e.g., for structure classification & predicting domain interactions). While motivated by specific problems, we believe this robust computational toolkit could be of broader utility for other structure-related workflows (i.e., as a community-wide resource), particularly as arise at the intersection of deep learning and structural bioinformatics.

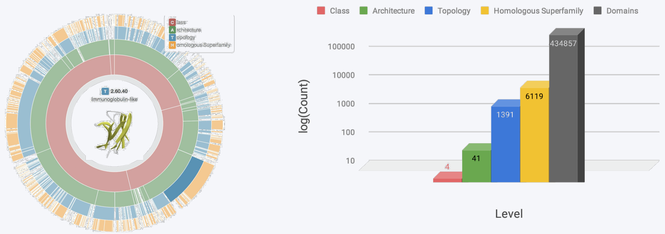
Setup & Deployment: Mapping Structural Info ⇒ Tasks

Toil [1] is a Python-based workflow management system (WMS) that enables one to auto-deploy workflows in the cloud or on local HPC resources, thus improving the reproducibility of computational pipelines. Each Toil job has child jobs and follow-on jobs, enabling the construction of complex MapReduce-like pipelines. Workflows can be built in Python or the Common Workflow Language.

A Workflow Based on PDB Files

The workflow systemically maps PDB files to jobs to run a given function.

A Workflow Based on CATH Domains

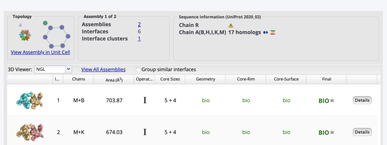


One job is created for each entry in the CATH [2] hierarchy, with child jobs spawned for subsidiary levels in the hierarchy. Once the workflow reaches a job for each individual domain (or specified level), it will run a given, user-provisioned function. Image and counts obtained from http://cathdb.info.

A Workflow Based on Protein Pairs

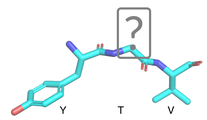
Biological Protein•Protein Interactions (PPI) are extracted from EPPIC [3] and each pair of chains is systemically mapped to jobs to run a given function.

Right: Biological PPI from 1KQ2 in EPPIC [3]

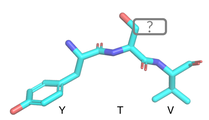


Use Case 1: Protein Structure Preparation

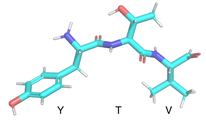
Structural bioinformatics workflows generally begin with a data-engineering task. Here, raw CATH domain structures are downloaded from the CATH API and cleaned (select first model and correct chain, remove altLocs and HETATMS), and then modified via the following stages:



1. Model missing residues using MODELLER [4] (if necessary). (PDB ID: 1KQ2)



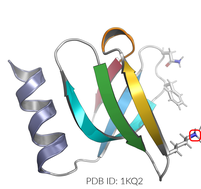
2. Correct rotamers if atoms are missing with SCWRL [5] (if necessary)



3. Add hydrogens and minimize structure with Pdb2pqr. [6]

Use Case 2: Calculate Biophysical Properties

Our toolkit enables one to efficiently and quickly compute biophysical properties for all atoms & residues in a structural dataset (e.g. from PDB or CATH). Some properties that we compute are:



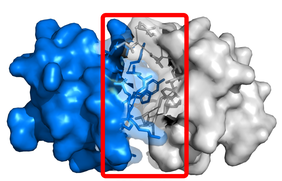
- **Atom Type** (AutoDock)
- **Partial Charge and Electrostatics** (APBS [6])
- **Concavity** (CX [7])
- **Hydrophobicity** (Kyte-Doolittle, Octanol, Biological)
- **Accessible Surface Area** (DSSP [8])
- **Secondary Structure** (DSSP [8])
- **Evolutionary Conservation** (EPPIC [3])

Some properties are computed at the residue level and mapped to each atom in the residue (e.g., hydrophobicity). For some features, residue-level values are calculated by combining atomic quantities, via various summation or averaging operations applied to the properties.

Use Case 3: Create & Validate Protein Complexes

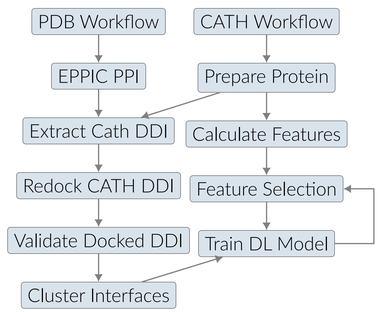
With over 430,000 domains in the CATH database, it becomes challenging to perform all vs. all domain docking in reasonable time. Our workflow toolkit allows for hundreds of simultaneous docking & validation runs. PPIs can be mapped to Domain•Domain Interactions (DDIs) based on CATH, which can be created or refined using HADDOCK & CNS. Finally, we validate the DDI complexes via the following various metrics:

Single Complex	Original vs. Refined
Bured Surface Area	Interface & Ligand RMSD
Radius of Gyration	TMScore (MMAlign)
ZRank Score [9]	Fraction of Common Contacts
CNS Energy	
HADDOCK Score [10]	
Binding Affinity (PRODIGY [11])	



We can also use this workflow to **cluster** interfaces, e.g. with tools such as MaxCluster, to create train/test splits. **Binding sites** can be similar even without structural similarity across a full 3D domain, so we cluster on binding site residues rather than the entire domain structure.

Illustrative Workflows



Domain•Domain Binding Site Prediction **Superfamily Reconstruction**

For both workflows, we need to 'clean' CATH domain PDB files, calculate biophysical features for each atom, and split into clusters to avoid train/test overlap. Feature selection is first performed using Pearson correlations (filter) and then performed again after model training by comparing AUC scores for each feature (embed).

Because our toolkit runs executable code via virtualization (as Docker containers), workflows are readily extensible. New tools can be easily added to the workflow as Docker containers. We have also dockerized many common bioinformatics tools (https://hub.docker.com/u/edraizen).

Conclusion

- We created a structural bioinformatics toolkit to perform **large-scale, reproducible analyses of thousands of structures simultaneously** that runs on any cloud resource or HPC
- Using the workflows, we **prepared all CATH domain structures** and **calculated biophysical features for all atoms** for use in deep learning models
- Biological **PPI have been mapped to CATH domains**; DDI validation still in progress
- We plan to make **datasets available as an API** and **accessible via PyTorch DataLoaders**

References and Acknowledgements

We thank M. Jaiswal, S. Saleem, K. Yonghyeon, Z. Zhao, and S. Veretnik for support, testing, and useful discussions. We also acknowledge the School of Data Science Presidential Fellows Program and NSF Career award MCB-1350957 for funding, and the UVA Rivanna team for HPC resources.

[1] J Vivian, A Rao, et al. *Nat Biotechnol*, 35(4):314--316, apr 2017.
[2] N Dawson, T Lewis, et al. *Nucleic Acids Res*, 45(D1):D289--D295, jan 2017.
[3] S Bliven, A Lafita, et al. *PLoS Comput Biol*, 14(4):e1006104, apr 2018.
[4] N Eswar, B Webb, et al. *Curr Protoc Bioinformatics*, Chapter 5:Unit 5.6, oct 2006.
[5] G Krivov, M Shapovalov, and R Dunbrack. *Proteins*, 77(4):778--795, dec 2009.
[6] T Dolinsky, P Czodrowski, et al. *Nucleic Acids Res*, 35(Web Server issue):W522--5, jul 2007.
[7] A Pintar, O Carugo, and S Pongor. *Bioinformatics*, 18(7):980--984, jul 2002.
[8] W Kabsch and C Sander. *Biopolymers*, 22(12):2577--2637, dec 1983.
[9] B Pierce, K Wiehe, et al. *Bioinformatics*, 30(12):1771--1773, jun 2014.
[10] C Dominguez, R Boelens, and A Bonvin. *J Am Chem Soc*, 125(7):1731--1737, feb 2003.
[11] L Xue, J Rodrigues, et al. *Bioinformatics*, 32(23):3676--3678, dec 2016.

doi://10.5281/zenodo.3914449

edraizen.github.com ed4bu@virginia.edu EliDraizen edraizen