*Bioinformatics*, 34(12), 2018, 2155–2157
doi: 10.1093/bioinformatics/bty077
Advance Access Publication Date: 12 February 2018
Applications Note

OXFORD

Systems biology

# ssbio: a Python framework for structural systems biology

**Nathan Mih[1,2,*], Elizabeth Brunk[2], Ke Chen[2], Edward Catoiu[2], Anand Sastry[2], Erol Kavvas[2], Jonathan M. Monk[2], Zhen Zhang[2] and Bernhard O. Palsson[2]**

[1]Department of Bioengineering, Bioinformatics and Systems Biology Graduate Program and [2]Department of Bioengineering, University of California, San Diego, CA 92093, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on July 21, 2017; revised on January 2, 2018; editorial decision on February 7, 2018; accepted on February 9, 2018

## Abstract

**Summary:** Working with protein structures at the genome-scale has been challenging in a variety of ways. Here, we present *ssbio*, a Python package that provides a framework to easily work with structural information in the context of genome-scale network reconstructions, which can contain thousands of individual proteins. The *ssbio* package provides an automated pipeline to construct high quality genome-scale models with protein structures (GEM-PROs), wrappers to popular third-party programs to compute associated protein properties, and methods to visualize and annotate structures directly in Jupyter notebooks, thus lowering the barrier of linking 3D structural data with established systems workflows.

**Availability and implementation:** *ssbio* is implemented in Python and available to download under the MIT license at http://github.com/SBRG/ssbio. Documentation and Jupyter notebook tutorials are available at http://ssbio.readthedocs.io/en/latest/. Interactive notebooks can be launched using Binder at https://mybinder.org/v2/gh/SBRG/ssbio/master?filepath=Binder.ipynb.

**Contact:** nmih@ucsd.edu

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

## 1 Introduction

Merging the disciplines of structural and systems biology remains promising in a variety of ways, but differences in the fields present a learning curve for those looking toward this integration within their own research. Beltrao et al. stated it best, that 'apparently structural biology and systems biology look like two different universes' (Beltrao *et al.*, 2007). A great number of software tools exist within the structural bioinformatics community (Biasini *et al.*, 2010; Grünberg *et al.*, 2007; Gu and Bourne, 2009; Hamelryck and Manderick, 2003; O'Donoghue *et al.*, 2015), and with recent advances in structure determination techniques, the number of experimental structures in the Protein Data Bank (PDB) continues to steadily rise (Mizianty *et al.*, 2014). The challenges of integrating external data and software tools into systems analyses have been detailed (Ghosh *et al.*, 2011), and structural information is no exception to the norm. At the systems-level, curated network models such as genome-scale metabolic models (GEMs) provide a context for molecular interactions in a functional cell (O'Brien *et al.*, 2015). Recently, GEMs integrated with protein structures (GEM-PROs) have extended these models to explicitly utilize 3D structural data alongside modeling methods to substantiate a number of hypotheses, as we explain below. Here, we present *ssbio*, a Python package designed with the goal of lowering the learning curve associated with efforts in structural systems biology. *ssbio* directly integrates with and builds upon the COBRApy toolkit (Ebrahim *et al.*, 2013) allowing for seamless integration with existing GEMs. The core functionality of *ssbio* is additionally extended by hooks to many popular third-party structural bioinformatics algorithms, such as DSSP, MSMS, SCRATCH, I-TASSER and others (see Supplementary Tables S1 and S2 for a full list) (Cheng *et al.*, 2005; Kabsch and Sander, 1983; Roy *et al.*, 2010; Sanner *et al.*, 1996).

## 2 Functionality

### 2.1 Protein class

*ssbio* adds a Protein class as an attribute to a COBRApy Gene and is representative of the gene's translated polypeptide chain (Fig. 1A). A Protein holds related amino acid sequences and structures, and a single representative sequence and structure can be set from these. This simplifies network analyses by enabling the properties of all or a subset of proteins to be computed and subsequently queried for. For details on these properties, as well as installation and execution instructions for the third-party software used to compute them, please refer to the documentation. Additionally, proteins with multiple structures available in the PDB can be subjected to QC/QA based on set cutoffs such as sequence coverage and X-ray resolution. Proteins with no structures available can be prepared for homology modeling through platforms such as I-TASSER (Roy *et al.*, 2010). Biopython representations of sequences (SeqRecord objects) and structures (Structure objects) are utilized to allow access to analysis functions available for their respective objects (Fig. 1B) (Cock *et al.*, 2009). Finally, all information contained in a Protein (or in the context of a network model, multiple proteins) can be saved and shared as a JavaScript Object Notation (JSON) file.

### 2.2 GEM-PRO pipeline

The objectives of the GEM-PRO pipeline have previously been detailed (Brunk *et al.*, 2016). A GEM-PRO directly integrates structural information within a curated GEM (Fig. 1C), and streamlines identifier mapping, representative object selection, and property calculation for a set of proteins. The pipeline provided in *ssbio* functions with an input of a GEM, but alternatively works with a list of gene identifiers or their protein sequences if network information is unavailable.

The added context of manually curated network interactions to protein structures enables different scales of analyses. For instance, from the top-down, global non-variant properties of protein structures such as the distribution of fold types can be compared within or between organisms (Brunk *et al.*, 2016; Monk *et al.*, 2017; Zhang *et al.*, 2009). From the bottom-up, structural properties predicted from sequence or calculated from structure can be utilized to guide a metabolic reconstruction (Broddrick *et al.*, 2016) or to enhance model predictive capabilities (Chang *et al.*, 2010, 2013; Chen *et al.*, 2017; Mih *et al.*, 2016). Looking forward, applications to multi-strain modelling techniques (Bosi *et al.*, 2016; Monk *et al.*, 2016; Ong *et al.*, 2014) would allow strain-specific changes to be investigated at the molecular level, potentially explaining phenotypic differences or strain adaptations to certain environments.

### 2.3 Scientific analysis environment

We provide a number of Jupyter notebook tutorials to demonstrate analyses at different scales (i.e. for a single protein sequence or structure, set of proteins, or network model). These notebooks can be launched in a virtual environment through the Binder project (https://mybinder.org/), with most third-party software pre-installed so users can immediately run through tutorials and experiment with them. Certain data can be represented as Pandas DataFrames (McKinney, 2012), enabling quick data manipulation and graphical visualization. These notebooks are further extended by visualization tools such as NGLview for interacting with and annotating 3D structures (Nguyen *et al.*, 2017; Rose and Hildebrand, 2015), and Escher for constructing and viewing biological pathways (King *et al.*, 2015) (Supplementary Fig. S1). Module organization and directory organization for cached files is further described in the Supplementary Material.

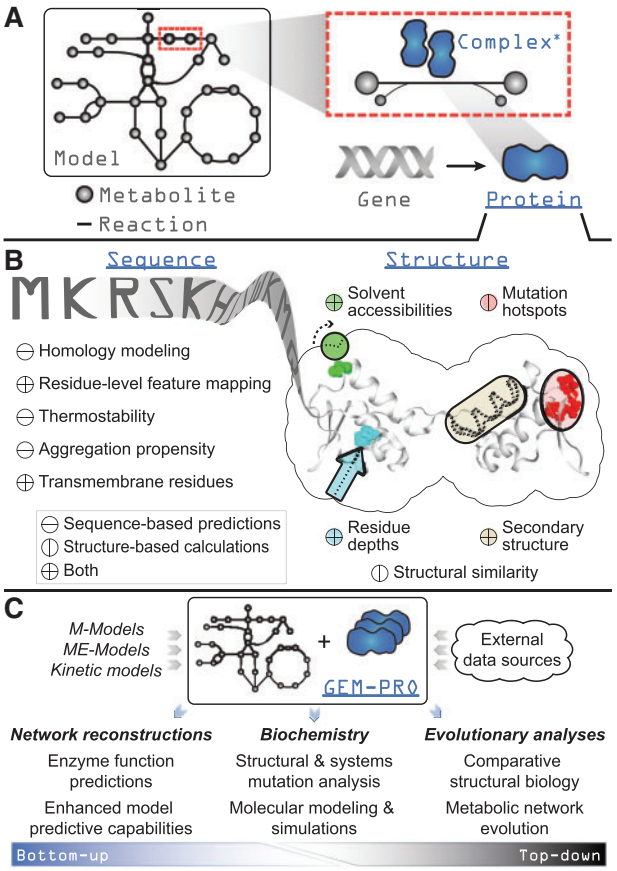


**Fig. 1.** Overview of the design and functionality of *ssbio*. Underlined fixed-width text indicates added functionality to COBRApy for a genome-scale model loaded using *ssbio*. (**A**) A simplified schematic showing the addition of a Protein to the core objects of COBRApy (non-underlined fixed-width text). A gene is directly associated with a protein, which can act as a monomeric enzyme or form an active complex with itself or other proteins (the asterisk denotes that methods for complexes are currently under development). (**B**) Summary of functions available for computing properties on a protein sequence or structure. (**C**) Uses of a GEM-PRO, from the bottom-up and the top-down. Once all protein sequences and structures are mapped to a genome-scale model, the resulting GEM-PRO has uses in multiple areas of study, as noted in the main text

## 3 Conclusion

*ssbio* provides a Python framework for systems biologists to start thinking about detailed molecular interactions and how they impact their models, and enables structural biologists to scale up and apply their expertise to multiple enzymes working together in a system. Towards a vision of whole-cell *in silico* models, structural information provides invaluable molecular-level details, and integration remains crucial.

## Acknowledgements

We would like to thank Dr. Zachary King, Patrick Phaneuf, Marta Matos and Colton Lloyd for valuable discussions in software development, and Dr. Laurence Yang, Yara Seif, J.C. Lachance and Jared Broddrick for insight into desired functionalities of the package. We would also like to thank Marc Abrams for proofreading of the manuscript.

## Funding

This work was supported by the Novo Nordisk Foundation Center for Biosustainability [NNF10CC1016517 to N.M., K.C., E.C. and A.S.]; the Swiss National Science Foundation [p2elp2_148961 to E.B.]; and the National Institute of General Medical Sciences of the National Institutes of Health [U01-GM102098 to B.O.P., 1-U01-AI124316-01 to J.M.M. and E.K.].

*Conflict of Interest:* none declared.

## References

Beltrao,P. *et al.* (2007) Structures in systems biology. *Curr. Opin. Struct. Biol.*, **17**, 378–384.

Biasini,M. *et al.* (2010) OpenStructure: a flexible software framework for computational structural biology. *Bioinformatics*, **26**, 2626–2628.

Bosi,E. *et al.* (2016). Comparative genome-scale modelling of staphylococcus aureus strains identifies strain-specific metabolic capabilities linked to pathogenicity. *Proc. Natl. Acad. Sci. USA*, **113**(26), E3801–9.

Broddrick,J.T. *et al.* (2016) Unique attributes of cyanobacterial metabolism revealed by improved genome-scale metabolic modeling and essential gene analysis. *Proc. Natl. Acad. Sci. USA*, **113**, E8344–E8353.

Brunk,E. *et al.* (2016) Systems biology of the structural proteome. *BMC Syst. Biol.*, **10**, 26.

Chang,R.L. *et al.* (2010) Drug off-target effects predicted using structural analysis in the context of a metabolic network model. *PLoS Comput. Biol.*, **6**, e1000938.

Chang,R.L. *et al.* (2013). Structural systems biology evaluation of metabolic thermotolerance in *Escherichia coli*. *Science*, **340**(6137), 1220–1223.

Chen,K. *et al.* (2017) Thermosensitivity of growth is determined by chaperone-mediated proteome reallocation. *Proc. Natl. Acad. Sci. USA*, **114**, 11548–11553.

Cheng,J. *et al.* (2005) SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Res*, **33** (Web Server issue), W72–W76.

Cock,P.J.A. *et al.* (2009) Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Ebrahim,A. *et al.* (2013) COBRApy: constraints-based reconstruction and analysis for python. *BMC Syst. Biol.*, **7**, 74.

Ghosh,S. *et al.* (2011) Software for systems biology: from tools to integrated platforms. *Nat. Rev. Genet.*, **12**, 821–832.

Grünberg,R. *et al.* (2007) Biskit—a software platform for structural bioinformatics. *Bioinformatics*, **23**, 769–770.

Gu,J. and Bourne,P.E. (2009). *Structural Bioinformatics*. John Wiley and Sons, New York.

Hamelryck,T. and Manderick,B. (2003) PDB file parser and structure class implemented in python. *Bioinformatics*, **19**, 2308–2310.

Kabsch,W. and Sander,C. (1983) DSSP: definition of secondary structure of proteins given a set of 3D coordinates. *Biopolymers*, **22**, 2577–2637.

King,Z.A. *et al.* (2015) Escher: a web application for building, sharing, and embedding Data-Rich visualizations of biological pathways. *PLoS Comput. Biol.*, **11**, e1004321.

McKinney,W. (2012). *Python for data analysis: data wrangling with Pandas, NumPy, and IPython*. O'Reilly Media Inc.

Mih,N. *et al.* (2016) A multi-scale computational platform to mechanistically assess the effect of genetic variation on drug responses in human erythrocyte metabolism. *PLoS Comput. Biol.*, **12**, e1005039.

Mizianty,M.J. *et al.* (2014) Covering complete proteomes with x-ray structures: a current snapshot. *Acta Crystallogr. D Biol. Crystallogr.*, **70**, 2781–2793.

Monk,J.M. *et al.* (2016) Multi-omics quantification of species variation of *Escherichia coli* links molecular features with strain phenotypes. *Cell Syst.*, **3**, 238–251.e12.

Monk,J.M. *et al.* (2017) iML1515, a knowledgebase that computes *Escherichia coli* traits. *Nat. Biotechnol.*, **35**, 904–908.

Nguyen,H. *et al.* (2017). NGLview: interactive molecular graphics for jupyter notebooks. *Bioinformatics*, **34**, 1241–1242.

O'Brien,E.J. *et al.* (2015). Using genome-scale models to predict biological capabilities. *Cell*, **161**(5), 971–987.

O'Donoghue,S.I. *et al.* (2015). Aquaria: simplifying discovery and insight from protein structures. *Nat. Methods*, **12**(2), 98–99.

Ong,W.K. *et al.* (2014) Comparisons of shewanella strains based on genome annotations, modeling, and experiments. *BMC Syst. Biol.*, **8**, 31.

Rose,A.S. and Hildebrand,P.W. (2015) NGL viewer: a web application for molecular visualization. *Nucleic Acids Res.*, **43**, W576–W579.

Roy,A. *et al.* (2010) I-TASSER: a unified platform for automated protein structure and function prediction. *Nat. Protoc.*, **5**, 725–738.

Sanner,M.F. *et al.* (1996) Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, **38**, 305–320.

Zhang,Y. *et al.* (2009). Three-dimensional structural view of the central metabolic network of thermotoga maritima. *Science*, **325**(5947), 1544–1549.