ssbio: A Python Framework for Structural Systems Biology

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

* Correspondence should be addressed to: B.O.P. (palsson@ucsd.edu)

[a] Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, CA 92093

[b] Department of Bioengineering, University of California, San Diego, CA 92093

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 which provides a framework to easily work with structural information in the context of genome-scale network reconstructions. The *ssbio* package provides an automated pipeline to construct high quality genome-scale models with protein structures (GEM-PROs), intuitively 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/.

Contact

nmih@ucsd.edu

Supplementary Information

Supplementary data are available at *bioRxiv*.

Introduction

Genome-scale models (GEMs), commonly stored using the Systems Biology Markup Language (SBML) (Hucka et al. 2003), are curated network models that provide a context for molecular interactions in a functional cell (O'Brien et al. 2015). A number of methods have been developed to simulate and analyze these models *in silico*, notably constraint-based modeling methods (i.e., COBRA) to study cell metabolism (Schellenberger et al. 2011). Recently, genome-scale models 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. A researcher interested in integrating structural information with their systems analyses and experimental datasets may encounter questions such as: how can I zoom in and visualize the interactions happening in the cell at the molecular level; how do structural properties correlate with my experimental datasets; or, how can I improve the contents of my model with structural data?

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 towards 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 (Gu & Bourne 2009), and with recent advances in structure determination techniques, the number of experimental protein 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-level analyses has been detailed (Ghosh et al. 2011), and structural information is no exception to the norm.

Here, we present *ssbio*, a Python package designed with the goal of lowering the learning curve associated with structural systems biology to directly address this challenge. *ssbio* directly integrates with and builds upon the COBRApy toolkit (Ebrahim et al. 2013) allowing for seamless integration with existing genome-scale models.

Functionality

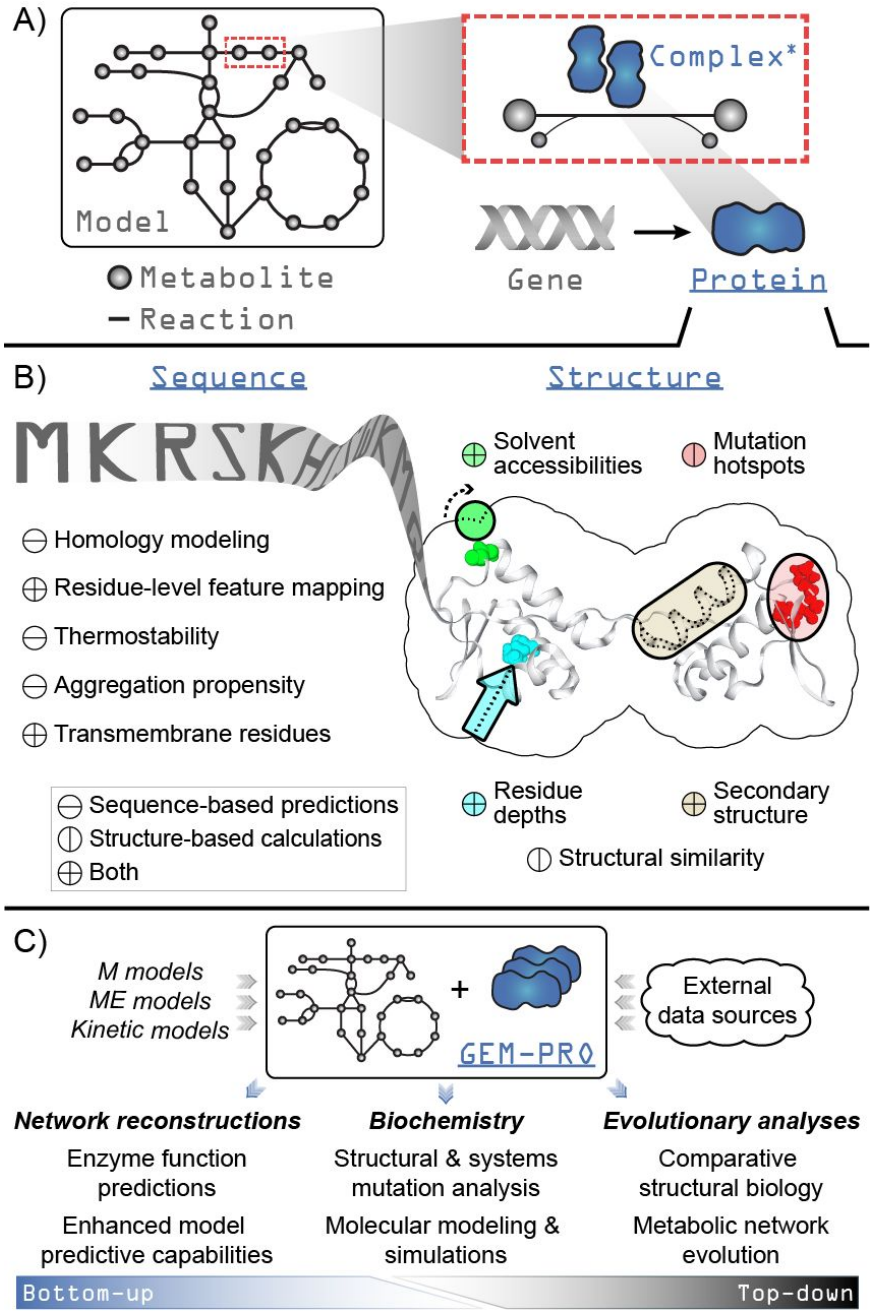

Fig. 1. Overview of the design and functionality of *ssbio.* Underlined fixed-width text in blue indicates added functionality to COBRApy for a genome-scale model loaded using *ssbio*. A) A simplified schematic showing the addition of a `Protein` to the core objects of COBRApy (`Model`, `Reaction`, `Metabolite`, and `Gene`). 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 protein complexes are currently under development). B) Summary of properties and functions available for a protein sequence and structure. For all details including software packages used and additional properties calculated, see Supplementary Table S1. Many properties can be predicted from the amino acid sequence, or computed directly from an available 3D structure. *ssbio* provides wrappers and output parsers for numerous popular packages to carry out these computations, either through the Biopython project (Cock et al. 2009) or through newly developed code. Residue-level properties from structures are further linked to the correct residue numbering schemes in mapped sequences, or vice-versa. 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. External database connections through the Bioservices project (Cokelaer et al. 2013) allow for even further extensions.

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). In the context of a COBRApy `Model` (`my_model` in the example below), the protein can be accessed simply by:

```
my_gene_id = 'Gene123'
my_protein = my_model.genes.get_by_id(my_gene_id).protein
```

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 queried for at once. 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 the I-TASSER platform (Roy et al. 2010).

Biopython representations of sequences (`SeqRecord` objects) and structures (`Structure` objects (Hamelryck & Manderick 2003)) are utilized to allow access to prediction and analysis functions available for their respective objects (Fig. 1B) (Cock et al. 2009). A full table of current integrations can be found in Supplementary Table S1. 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.

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* requires an input of a GEM in supported formats (SBML, JSON, or MAT), 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; 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; Liu et al. 2014; Mih et al. 2016; O'Brien et al. 2013). Looking forward, applications to multi-strain modelling techniques (Bosi et al. 2016; Monk et al. 2013, 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.

Scientific analysis environment

We provide a number of Jupyter notebook tutorials available with the documentation (Kluyver et al. 2016) to demonstrate analyses at different scales (i.e. for a single protein sequence or structure, set of proteins, or network model). Certain data can be represented as Pandas DataFrames (McKinney 2012), enabling quick data manipulation and graphical visualization. These notebooks demonstrate fully-featured Python scientific analysis environments which are further extended by visualization tools such as the NGL viewer for visualizing 3D structures (Rose & Hildebrand 2015), and Escher for visualizing biological pathways (King et al. 2015) (Supplementary Figure S1). Module organization and directory organization for cached files is further described in the Supplementary Text.

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.

Funding

This work was supported by the Novo Nordisk Foundation Center for Biosustainability at the Technical University of Denmark [NNF10CC1016517 to N.M., E.C., K.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.].

Acknowledgements

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

References

Beltrao, P., Kiel, C., & Serrano, L. (2007). 'Structures in systems biology', *Current opinion in structural biology*, 17/3: 378–84. Elsevier.

Bosi, E., Monk, J. M., Aziz, R. K., Fondi, M., Nizet, V., & Palsson, B. Ø. (2016). 'Comparative genome-scale modelling of Staphylococcus aureus strains identifies strain-specific metabolic capabilities linked to pathogenicity', *Proceedings of the National Academy of Sciences of the United States of America*, 113/26: E3801–9. DOI: 10.1073/pnas.1523199113

Broddrick, J. T., Rubin, B. E., Welkie, D. G., Du, N., Mih, N., Diamond, S., Lee, J. J., et al. (2016). 'Unique attributes of cyanobacterial metabolism revealed by improved genome-scale metabolic modeling and essential gene analysis', *Proceedings of the National Academy of Sciences of the United States of America*, 113/51: E8344–53. DOI: 10.1073/pnas.1613446113

Brunk, E., Mih, N., Monk, J., Zhang, Z., O'Brien, E. J., Bliven, S. E., Chen, K., et al. (2016). 'Systems biology of the structural proteome', *BMC systems biology*, 10/1: 26. bmcsystbiol.biomedcentral.com. DOI: 10.1186/s12918-016-0271-6

Chang, R. L., Andrews, K., Kim, D., Li, Z., Godzik, A., & Palsson, B. O. (2013). 'Structural systems biology evaluation of metabolic thermotolerance in Escherichia coli', *Science*, 340/6137: 1220–3. American Association for the Advancement of Science.

Chang, R. L., Xie, L., Xie, L., Bourne, P. E., & Palsson, B. Ø. (2010). 'Drug off-target effects predicted using structural analysis in the context of a metabolic network model', *PLoS computational biology*, 6/9: e1000938. Public Library of Science.

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., et al. (2009). 'Biopython: freely available Python tools for computational molecular biology and bioinformatics', *Bioinformatics* , 25/11: 1422–3. Oxford Univ Press.

Cokelaer, T., Pultz, D., Harder, L. M., Serra-Musach, J., & Saez-Rodriguez, J. (2013). 'BioServices: a common Python package to access biological Web Services programmatically.', *Bioinformatics* , 29/24: 3241–2. DOI: 10.1093/bioinformatics/btt547

Ebrahim, A., Lerman, J. A., Palsson, B. O., & Hyduke, D. R. (2013). 'COBRApy: COnstraints-Based Reconstruction and Analysis for Python', *BMC systems biology*, 7: 74. DOI: 10.1186/1752-0509-7-74

Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.-Y., & Kitano, H. (2011). 'Software for systems biology: from tools to integrated platforms', *Nature reviews. Genetics*, 12/12: 821–32. DOI: 10.1038/nrg3096

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

Hamelryck, T., & Manderick, B. (2003). 'PDB file parser and structure class implemented in Python', *Bioinformatics* , 19/17: 2308–10.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., et al. (2003). 'The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models', *Bioinformatics* , 19/4: 524–31. DOI: 10.1093/bioinformatics/btg015

King, Z. A., Dräger, A., Ebrahim, A., Sonnenschein, N., Lewis, N. E., & Palsson, B. O. (2015). 'Escher: A Web Application for Building, Sharing, and Embedding Data-Rich Visualizations of Biological Pathways', *PLoS computational biology*, 11/8: e1004321. DOI: 10.1371/journal.pcbi.1004321

Kluyver, T., Ragan-Kelley, B., & Pérez, F. (2016). 'Jupyter Notebooks—a publishing format for reproducible computational workflows', *and Power in …*. microblogging.infodocs.eu.

Liu, J. K., O'Brien, E. J., Lerman, J. A., Zengler, K., Palsson, B. O., & Feist, A. M. (2014). 'Reconstruction and modeling protein translocation and compartmentalization in Escherichia coli at the genome-scale', *BMC systems biology*, 8: 110. DOI: 10.1186/s12918-014-0110-6

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

Mih, N., Brunk, E., Bordbar, A., & Palsson, B. O. (2016). 'A Multi-scale Computational Platform to Mechanistically Assess the Effect of Genetic Variation on Drug Responses in Human Erythrocyte Metabolism', *PLoS computational biology*, 12/7: e1005039. DOI: 10.1371/journal.pcbi.1005039

Mizianty, M. J., Fan, X., Yan, J., Chalmers, E., Woloschuk, C., Joachimiak, A., & Kurgan, L. (2014). 'Covering complete proteomes with X-ray structures: a current snapshot', *Acta crystallographica. Section D, Biological crystallography*, 70/Pt 11: 2781–93. DOI: 10.1107/S1399004714019427

Monk, J. M., Charusanti, P., Aziz, R. K., Lerman, J. A., Premyodhin, N., Orth, J. D., Feist, A. M., et al. (2013). 'Genome-scale metabolic reconstructions of multiple Escherichia coli strains highlight strain-specific adaptations to nutritional environments', *Proceedings of the National Academy of Sciences of the United States of America*, 110/50: 20338–43. DOI: 10.1073/pnas.1307797110

Monk, J. M., Koza, A., Campodonico, M. A., Machado, D., Seoane, J. M., Palsson, B. O., Herrgård, M. J., et al. (2016). 'Multi-omics Quantification of Species Variation of Escherichia coli Links Molecular Features with Strain Phenotypes', *Cell systems*, 3/3: 238–51.e12. DOI: 10.1016/j.cels.2016.08.013

O'Brien, E. J., Lerman, J. A., Chang, R. L., Hyduke, D. R., & Palsson, B. Ø. (2013). 'Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction', *Molecular systems biology*, 9/1. Wiley Online Library.

O'Brien, E. J., Monk, J. M., & Palsson, B. O. (2015). 'Using Genome-scale Models to Predict Biological Capabilities', *Cell*, 161/5: 971–87. DOI: 10.1016/j.cell.2015.05.019

Ong, W. K., Vu, T. T., Lovendahl, K. N., Llull, J. M., Serres, M. H., Romine, M. F., & Reed, J. L. (2014). 'Comparisons of Shewanella strains based on genome annotations, modeling, and experiments', *BMC systems biology*, 8: 31. DOI: 10.1186/1752-0509-8-31

Rose, A. S., & Hildebrand, P. W. (2015). 'NGL Viewer: a web application for molecular visualization', *Nucleic acids research*, 43/W1: W576–9. DOI: 10.1093/nar/gkv402

Roy, A., Kucukural, A., & Zhang, Y. (2010). 'I-TASSER: a unified platform for automated protein structure and function prediction', *Nature protocols*, 5/4: 725–38. Nature Publishing Group.

Schellenberger, J., Que, R., Fleming, R. M. T., Thiele, I., Orth, J. D., Feist, A. M., Zielinski, D. C., et al. (2011). 'Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2. 0', *Nature protocols*, 6/9: 1290–307. Nature Publishing Group.

Zhang, Y., Thiele, I., Weekes, D., Li, Z., Jaroszewski, L., Ginalski, K., Deacon, A. M., et al. (2009). 'Three-dimensional structural view of the central metabolic network of Thermotoga maritima', *Science*, 325/5947: 1544–9. American Association for the Advancement of Science.