The Journal of Open Source Software

dfba: Software for efficient simulation of dynamic flux-balance analysis models in Python

David S. Tourigny[1], **Jorge Carrasco Muriel**[2], **and Moritz E. Beber**[2]

1 Columbia University Irving Medical Center, 630 West 168th Street, New York, NY 10032 USA **2** Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, Building 220, Kemitorvet, 2800 Kongens Lyngby, Denmark

DOI: 10.21105/joss.02342

Software

- Review
- Repository
- Archive

Editor: Marcos Vital

Reviewers:

- @jdbrunner
- @pstjohn
- @synchon

Submitted: 20 May 2020
Published: 31 August 2020

Summary

Flux-balance analysis (FBA) is a computational method based on linear programming (LP) that has had enormous success modeling the metabolic behaviour of organisms and cellular systems existing in steady state with their environment (Orth, Thiele, & Palsson, 2010; Varma & Palsson, 1994). Representing the corresponding biological model as an LP problem means that FBA can be used to study metabolism at genome-scale. Unfortunately, the underlying assumption of an unchanging environment means that FBA is not immediately applicable to systems with dynamics where, for example, environmental conditions may vary in time. Extensions of FBA, including dynamic FBA (DFBA) (Mahadevan, Edwards, & Doyle, 2002), have therefore been developed in order to accommodate temporal dynamics into the framework of genome-scale metabolic modeling.

Although DFBA is well-defined mathematically as an LP problem embedded in a system of ordinary differential equations (ODEs), numerical simulation of DFBA models proves particularly challenging (as described in (Harwood, Höffner, & Barton, 2016)). Consequently, Harwood et al. (2016) proposed an algorithm for efficiently simulating DFBA models based on reformulating the ODE system as a differential algebraic equation (DAE) with root detection and representing solutions of the LP problem using an optimal basis formulation. An initial implementation of this algorithm has been provided in the software package DFBAlab (Gomez, Höffner, & Barton, 2014) written in MATLAB and using commercial LP solvers.

Increasingly, researchers engaged in metabolic modeling prefer open source software. Python is quickly becoming their platform of choice (Carey, Dräger, Beber, Papin, & Yurkovich, 2020). Among other packages, open source resources for building and simulating FBA models using Python can be found in the COBRApy (Ebrahim, Lerman, Palsson, & Hyduke, 2013) package which is part of the openCOBRA organization (openCOBRA, n.d.). Until now, COBRApy lacked an efficient implementation of DFBA using the DAE formulation. **Statement of need**: *researchers wanting to build and simulate specific models of interest often lack the background in numerical analysis or high-performance computing required to overcome the numerical challenges of DFBA*. We have solved this issue by developing a software package based on open source libraries GLPK (Makhorin, n.d.) and SUNDIALS (Hindmarsh et al., 2005) that implements the most efficient algorithms in a compiled programming language that is made accessible to users through a simple and intuitive pybind11 (Wenzel, Rhinelander, & Moldovan, n.d.) interface with pandas (McKinney, 2011) and the openCOBRA Python module COBRApy (Ebrahim et al., 2013). ODEs are constructed using the symbolic expression enabled by optlang (Jensen, Cardoso, & Sonnenschein, 2016), SymPy, and SymEngine (Meurer et al., 2017).

Acknowledgements

DST is a Simons Foundation Fellow of the Life Sciences Research Foundation. MEB received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 686070 (DD-DeCaF). We thank Peter St. John and Christian Diener for discussions and suggestions.

References

Carey, M. A., Dräger, A., Beber, M. E., Papin, J. A., & Yurkovich, J. T. (2020). Community standards to facilitate development and address challenges in metabolic modeling. *Molecular Systems Biology*, *16*(8), e9235. doi:10.15252/msb.20199235

Ebrahim, A., Lerman, J. A., Palsson, B. Ø., & Hyduke, D. R. (2013). COBRApy: COnstraints-based reconstruction and analysis for python. *BMC Syst Biol.*, *7*, 74. doi:10.1186/1752-0509-7-74

Gomez, J. A., Höffner, K., & Barton, P. I. (2014). DFBAlab: A fast and reliable matlab code for dynamic flux balance analysis. *BMC Bioinform.*, *15*, 409. doi:10.1186/s12859-014-0409-8

Harwood, S. M., Höffner, K., & Barton, P. I. (2016). Efficient solution of ordinary differential equations with a parametric lexicographic linear program embedded. *Numerische Mathematik*, *133*, 623–653. doi:10.1007/s00211-015-0760-3

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, *31*, 363–396. doi:10.1145/1089014.1089020

Jensen, K., Cardoso, J., & Sonnenschein, N. (2016). Optlang: An algebraic modeling language for mathematical optimization. *J. Open Source Softw.*, *7*, 139. doi:10.21105/joss.00139

Mahadevan, R., Edwards, J. S., & Doyle, F. J. (2002). Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophys. J.*, *83*, 1331–1340. doi:10.1016/S0006-3495(02)73903-9

Makhorin, A. (n.d.). GNU linear programming kit. Retrieved from https://www.gnu.org/software/glpk/glpk.html

McKinney, W. (2011). Pandas: A foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, *14*.

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., et al. (2017). SymPy: Symbolic computing in python. *PeerJ Computer Science*, *3*, e103.

openCOBRA. (n.d.). OpenCOBRA: Open-source, community-developed code base for constraint-based reconstruction and analysis. Retrieved from https://opencobra.github.io/

Orth, J., Thiele, I., & Palsson, B. Ø. (2010). What is flux balance analysis? *Nat. Biotechnol.*, *28*, 245–248. doi:10.1038/nbt.1614

Varma, A., & Palsson, B. Ø. (1994). Metabolic flux balancing: Basic concepts, scientific and practical use. *Nat. Biotechnol.*, *12*, 994–998. doi:10.1038/nbt1094-994

Wenzel, J., Rhinelander, J., & Moldovan, D. (n.d.). Pybind11 - seamless operability between c++11 and python. Retrieved from https://github.com/pybind/pybind11