Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Dynamic Flux Balance Analysis Models in SBML

Leandro H. Watanabe [1,†] Matthias König [2,†] and Chris J. Myers [1,*]

[1]Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, 84112, USA
[2]Humboldt-University Berlin, Institute for Theoretical Biology, Invalidenstraße 43, 10115 Berlin, Germany

*To whom correspondence should be addressed.
†Equal contribution.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Systems biology models are typically simulated using a single formalism such as *ordinary differential equations* (ODE) or stochastic methods. However, more complex models require the coupling of different formalisms since different biological process have been shown to be better described using different methods. For instance, metabolism is often modeled using *flux-balance analysis* (FBA). Other dynamic changes of model components are better represented via ODEs. The coupling of FBA and ODE results in dynamic FBA models. A major challenge is how to describe such hybrid models so that they can be exchanged between tools and simulated consistently in different tools.

Results: This paper presents a scheme and implementation for encoding dynamic FBA models in the *Systems Biology Markup Language* (SBML), which allows the modeling of multi-framework computational models to be exchanged between software tools. The paper shows the feasibility of the approach using various example models and demonstrates that different tools are able to simulate and agree on the results. As part of this work, two independent implementations of the simulation method have been developed supporting such models: `iBioSim` and `sbmlutils`.

Availability: The materials discussed can be found at https://github.com/matthiaskoenig/dfba. The tools used in this project are freely available for download at: http://www.async.ece.utah.edu/ibiosim for `iBioSim`, and https://github.com/matthiaskoenig/sbmlutils/ for `sbmlutils`.

Contact: myers@ece.utah.edu

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

1 Introduction

In systems biology, mathematical modeling has been widely used to describe biological systems (Kitano, 2002). Computational models can be simulated and analyzed *i*n silico. Consequently, models allows researchers to make predictions which subsequently can be validated experimentally. Furthermore, models can give intuitions of biological systems that would be difficult to obtain in a wet lab. A key challenge, however, is ensuring that the modeling efforts are reproducible and easily exchanged between research groups such that existing models can be reused to build more complex models. To achieve this goal, standard model representation formats, such as the *Systems Biology Markup Language* (SBML) (Hucka *et al.*, 2003) or CellML (Hedley *et al.*, 2001), should be utilized. Both SBML and CellML have been successfully applied to the encoding of models using a single modeling framework, but the support of multiple modeling framework adds new challenges. This paper addresses this problem by developing a methodology and simulation tools to support such hybrid modeling efforts, and it demonstrates the successful exchange of models between two simulation tools.

1.1 Multi-framework computational models

Various simulation and analysis methods have been developed. Kinetic time-course simulation based on *ordinary differential equations* (ODE), is often employed to observe the dynamics of the entities in a model over time. Depending on the research question and biological system, such simulations can be either deterministic or non-deterministic (stochastic). Other simulation frameworks are boolean (Thomas, 1973; Kauffman, 1969) models, logical models (Morris *et al.*, 2010) and constraint-based approaches (Bordbar *et al.*, 2014), among others.

Metabolic networks, in particular, are often challenging to model dynamically using ODE approaches because kinetic parameters needed

for ODE models are often unavailable (Varma and Palsson, 1994). Hence, steady-state approaches that do not need kinetic information are often used to model metabolism, so called *flux balance analysis* (FBA) (Savinell and Palsson, 1992; Varma *et al.*, 1993) based on constraint-based optimization. This method only requires the connectivity of the reactions and metabolites along with the respective stoichiometry, an objective function (e.g. cell growth), and flux constraints. The idea is to constrain the model based on the stoichiometry of the reactions and optimize the objective function while satisfying the flux constraints. The advantages of using such method include its efficiency and not needing the kinetic information that other methods need to analyze a model.

Biological research questions often require the coupling of different model formalisms. One such recent example is the whole-cell model for the *Mycoplasma genitalium* (Karr *et al.*, 2012) that is encoded using a mixture of boolean networks, stochastic processes, differential equations, and FBA.

1.2 Dynamic flux balance analysis

One disadvantage of FBA is that it cannot express the dynamics of the metabolites since it does not update amounts or concentrations. Due to this limitation, the field of *dynamic FBA* (DFBA) (Varma and Palsson, 1994) has emerged, which couples the stationary flux distribution resulting from FBA with the kinetic update of the metabolites taken up or consumed by the FBA network.

Besides the whole-cell model, many DFBA models have been constructed for different metabolic pathways. DFBA has been applied in small-scale examples (Varma and Palsson, 1994; Mahadevan *et al.*, 2002; Luo *et al.*, 2006), over medium-size models (Pizarro *et al.*, 2007; Lequeux *et al.*, 2010; Meadows *et al.*, 2010), and up to genome-scale DFBA applications (Hanly and Henson, 2011; Hjersted *et al.*, 2007). For a recent overview, see Table 1 in (Höffner *et al.*, 2013).

The coupling between FBA and kinetic parts can be implemented via three main approaches, i.e. *static optimization approach* (SOA), *dynamic optimization approach* (DOA), or *direct approach* (DA) Gomez *et al.* (2014). DOA approaches discretize the simulation time and optimize simultaneously over the entire time period by solving a nonlinear programming problem (NLP). The DA directly includes the LP solver in the right-hand side for the ordinary differential equations (ODEs). The SOA approach solves the LP at each time step using a Euler forward method assuming constant fluxes over the time step Gomez *et al.* (2014). Most of the published DFBA models use the SOA approach, which is relatively simple to implement and not as computationally demanding (see methods algorithm).

1.3 Exchangeability & reproducibility of models

Despite the multitude of published DFBA models, currently no standard for the exchange of such models exists. Existing implementations are written directly in programming languages (e.g., MATLAB or Python). Mathematical models in their respective formalisms are embedded in the script along with the connections between the kinetic and flux balance parts of the models. As a consequence, it is not possible to exchange existing DFBA models between different software tools. Thus, they cannot be reproduced.

While it is possible to replicate the same scripts in different programming languages, it is unpractical to do so as replication is error prone, requires unnecessary work, and needs conversions that can lead to data loss. For these reasons, script replication makes achieving reproducibility difficult and often infeasible. Losing reproducibility capabilities is especially problematic because often multiple optima can exist in FBA simulations, with one of the optima chosen based on the solver. The necessity of an exchange format for DFBA resulted from efforts to encode and reproduce the DFBA submodel of the Karr model using standards during the whole-cell workshop (Waltemath *et al.*, 2016).

1.4 Model standards

In order to achieve exchangeability and reproducibility of models, standards for the encoding of models have been created. The *de-facto* standard for systems biology models is SBML Hucka *et al.* (2003). SBML core elements include *species* (e.g. proteins, genes, etc.) and *reactions*, which are used to transform species into other species. In addition, SBML includes *compartments* for describing the membrane enclosures for species and reactions, *rules* for describing continuous dynamics, and *events* for describing discrete dynamics. SBML core elements are used to describe mathematical models of reaction-based networks. SBML uses packages for extending the functionality of the core elements. While SBML is used to encode mathematical models of biological networks, there are different standards for other purposes: the *Simulation Experiment Description Markup Language* (SED-ML) is used for describing simulations (Waltemath *et al.*, 2011), the *Systems Biology Graphical Notation* (SBGN) is used for describing visualizations (Le Novere *et al.* (2009))), and Combine Archives are used for exchanging collections of modeling files Bergmann *et al.* (2014). The main advantage of using standards over MATLAB is the ability to exchange models between research groups and reproduce results using various tools that support these standards.

SBML core in combination with the *hierarchical model composition* (comp) package (Smith *et al.*, 2015) and the *flux balance constraints* (fbc) package (Olivier and Bergmann, 2015) is used as the basis for describing the mathematical models in this work. The comp package is used to construct hierarchical models, where models can be built from submodels. The fbc package is used to encode the flux bounds for the reactions and an objective function, which allows one to perform FBA on an SBML model. Furthermore, this work uses SED-ML to describe how each SBML model should be simulated. That is, it encodes which simulation algorithm to use and its corresponding parameters. In addition, it can be used to generate plots and alter the model before simulation.

One the challenges in current SBML models is the limitation on the expression of models using different formalisms. Although there are several tools that support ODE simulation and FBA, they all support them independently. In order to overcome this challenge, this paper introduces a scheme that allows the coupling of ODE and FBA models. This paper demonstrates that this scheme provides exchangeability by encoding and simulating DFBA models in both `iBioSim` (Madsen *et al.*, 2012) and `sbmlutils` König (2017).

2 Methods

2.1 Model encoding

The DFBA models presented in this paper were created in the proposed scheme either using a graphical user interface in `iBioSim` or scripts in `sbmlutils`. For a given model, the TOP, FBA, BOUNDS, and UPDATE submodels were packaged with respective simulation files using SED-ML in combine archives for the exchange between different tools. All models and simulation results are available from https://github.com/matthiaskoenig/dfba

2.2 Stationary optimization approach (SOA)

A stationary optimization approach for DFBA was implemented as a simulation algorithm in `iBioSim` and `sbmlutils` following the simulation scheme depicted in Figure 1.

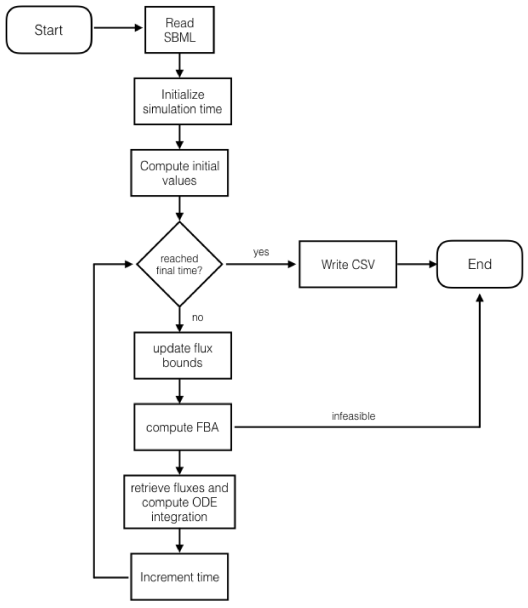


Fig. 1: Overview of the implemented SOA algorithm for DFBA. After initialization of the model, FBA and kinetic simulation are run in an iterative manner until the simulation end point. In every step, FBA is used to compute the reaction rates of the FBA network. Subsequently, based on the computed FBA rates, the values of the species are updated dynamically. In the SOA approach, FBA fluxes are assumed to be constant within a time step. For a detailed description see the methods section.

The first step is the initialization of the model. All of the species and parameters in the model are initialized, where each variable's initial value is computed. After the initialization step, the FBA submodel is executed. During the FBA step, reaction fluxes are computed using the initial flux bound values where the flux bounds for the reactions come from the top-level using SBML comp *replacements*. In SBML, replacements of parameters and species indicate the top-level entities are the same entity as the one being replaced. Once the fluxes are computed, they are assigned to parameters using assignment rules on the top-level. These parameters are assigned reaction rates computed as functions of the fluxes. After computing reaction fluxes, the update step is performed concurrently with the dynamic step by computing the time-evolution of every species in the UPDATE and KINETIC submodels. Species that affect any flux bound in the FBA submodel are updated in the top-level. The new bounds are used in the FBA submodel for the next time step. Simulation time is incremented at the end. If the time limit is reached, then simulation is complete. Otherwise, all of the steps are repeated.

The proposed simulation algorithm has been implemented in `iBioSim` and `sbmlutils`. The `iBioSim` tool uses the structure of (Watanabe and Myers, 2014) for simulation. The `sbmlutils` tool uses *tellurium* (Choi *et al.*, 2016) for the kinetic simulation and *cobrapy* (Ebrahim *et al.*, 2013) to solve the FBA problem. Both `iBioSim` and `sbmlutils` take an SBML file that describes a DFBA model and a SED-ML file that describes the simulation experiment. In the proposed approach, SED-ML is mainly used to indicate which simulation algorithm to use, the time points in which tools should print out the values of the variables, the initial time and the time limit. Many other features are not used. In order to compare results between different tools, simulators should execute each time increment consistently. The value of each time increment for SOA is defined as a parameter with id dt in the SBML model but can be overwritten by the SED-ML file.

All example models are available as Combine Archives with example results from the two implementations. The Combine archives include the SBML model files, a SED-ML file, and the results created by each software.

2.3 Reproducibility between tools

Reproducibility of DFBA models is challenging because there may exist several possible outcomes that satisfy the objective function and constraints of the FBA models. In order to test interoperability, models were built in both the `iBioSim` and the `sbmlutils` tools. Models built in `iBioSim` were then imported into `sbmlutils` and vice-versa to check whether models could be interpreted by both tools consistently. Once interoperability is tested, models undergo *Flux Variability Analysis* (FVA), a procedure that gives the minimal and maximal fluxes for each reaction in each step of the simulation, for the constructed models. FVA allows the verification of uniqueness of the DFBA solution. Reproducibility of the model simulations is tested by comparing the numerical SOA results between the two tools for models with unique solutions. Namely, simulators should produce the same results (within some tolerance) for models with unique solutions.

3 Results

The major result of this work is the creation of the first schema for encoding DFBA in SBML, demonstrating exchangeability and reproducibility of DFBA models between tools.

3.1 Schema for dynamic flux balance analysis

This paper proposes for the first time a scheme to encode hybrid models, such as DFBA models, in SBML. All of the interconnections are encoded in the SBML model rather than in SED-ML. The connections between components should be part of the model and not simulation. This means that this scheme requires a single SBML model and a single SED-ML file. The scheme is depicted in Figure 2.

The model is constructed hierarchically using the SBML comp package. The top-level model is composed of four submodels: (i) a submodel that computes flux bounds to ensure the model never runs out of any metabolite (BOUNDS submodel); (ii) a submodel that encodes metabolism as a FBA problem (FBA submodel); and (iii) a submodel that updates the amounts and concentrations of the metabolites used in the FBA submodel using chemical kinetics rather than reaction fluxes (UPDATE submodel); (iv) an optional submodel that represents a dynamic part with all kinetics other than the metabolic pathway, such as DNA transcription, DNA translation, and protein degradation, among others (KINETIC submodel). The top-level model ties together the four different submodels using SBML comp replacements.

In order to describe the different formalisms in each submodel, the *Systems Biology Ontology* (SBO) is used (Courtot *et al.*, 2011). The SBO defines controlled vocabulary terms used in the systems biology field. The SBO terms are arranged in a taxonomic hierarchy using a tree structure. This allows the grouping of terms that are related to one another. The modeling formalisms are described using terms on the *modeling framework* branch, where FBA models are described using the *flux balance framework* term, stochastic processes are described using the *non-spatial discrete framework* term, and differential equations are described using the *non-spatial discrete framework* term. Although the terms for stochastic processes and differential equations can be used for describing either stochastic or deterministic simulation methods, these terms were selected

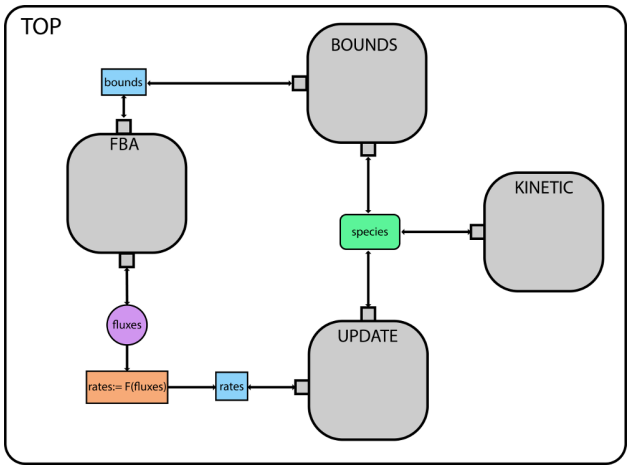


Fig. 2: Schema for encoding DFBA models in SBML. The model is composed of a top-level model with three submodels: FBA, UPDATE, and KINETIC. The FBA submodel computes the reaction fluxes of the metabolic fbc model. The UPDATE submodel calculates the dynamic update of the metabolites in the metabolic pathway. The dynamic submodel includes all of the other processes in the model, which may affect or be affected by entities in metabolism. The top-level model ties together the three different submodels using SBML comp replacements.

because they are the ones that best describes these two formalisms. The detailed guidelines of encoding are provided in Supplement [?]

3.2 Minimal Example

In order to illustrate the proposed scheme, a simplified example of a whole-cell model is shown in Figure 3. This example is not a representation of a real model, but rather a toy model that is used for testing purposes. This model is constructed hierarchically where a top-level model is created to instantiate different submodels (BOUNDS, UPDATE, and FBA) and make the necessary connections between them. The figure illustrates the structure of each submodel and how each submodel ties in with each other in a flat version of the model once all of the connections are established.

In order to validate the model, we simulate using the simulation algorithm described in Figure 1. The results are shown in Figure 4. The two tools produce curves the comparable results (within some tolerance level).

In addition to the presented minimal model, a second model of a simplified DFBA glycolysis is available in the supplementary material.

3.3 Diauxic growth in *E. coli*

The next example is an encoding and reproduction of results from a published DFBA model of diauxic growth of *E. coli* (Mahadevan *et al.*, 2002) consisting of four reactions between four metabolites, i.e. glucose ($Glcxt$), oxygen (O_2), acetate (A_c) and biomass (X). The model can grow either aerobically on acetate ($v1$), aerobically on glucose ($v2$ or $v3$) or anaerobically convert glucose to acetate:

$$v1 : 39.43A_c + 35O_2 \rightarrow X$$
$$v2 : 9.46Glcxt + 12.92O_2 \rightarrow X$$
$$v3 : 9.84Glcxt + 12.73O_2 \rightarrow 1.24A_c + X$$
$$v4 : 19.23Glcxt \rightarrow 12.12A_c + X$$

The kinetic part of the model is described by the following differential equations:

$$\frac{dGlcxt}{dt} = A^{Glcxt}\nu X$$
$$\frac{dA_c}{dt} = A^{A_c}\nu X$$
$$\frac{dO_2}{dt} = A^{O_2}\nu X + k_L a(0.21 - O_2)$$
$$\frac{dX}{dt} = (v1 + v2 + v3 + v4)X$$

where A^{Glcxt}, A^{A_c}, A^{O_2} are the respective rows of each variable in the stoichiometry matrix and $k_L a$ is the mass transfer coefficient of oxygen. For a detailed description see (Mahadevan *et al.*, 2002).

The results in Figure 5 show that the cell has an exponential growth until the cell runs out of glucose, which at this point the cell grows linearly due to oxygen. When both oxygen and glucose run out, the cell growth stagnates. Experimental data from (Varma and Palsson, 1994) is plotted alongside the simulation results. The model is able to capture similar behavior from the experimental data. The results are equivalent to the models in (Mahadevan *et al.*, 2002).

3.4 *E. coli* core

To demonstrate the feasibility of the proposed scheme and method for real-world examples of DFBAs, a larger metabolic network for the core metabolism of *E. coli* is encoded in the proposed schema and simulated as shown in Figure **??**. The encoding of larger scale examples demonstrates the scalability of the proposed approach. While `sbmlUtils` is able to find a solution for the model, `iBioSim` cannot as it runs into an unfeasible solution in the middle of simulation. This captures the well-known problem of DFBA with multiple solutions. Depending on the solutions the simulator dynamically picks, different solutions may arise. Even though multiple solutions may exist, tools typically pick solutions deterministically. Hence, tools can reproduce their own results, but results will vary if different tools use different solvers. Without the use of standards, this could never be demonstrated because variations in results could be due to discrepancies in the model, and not in the tool.

4 Discussion

Modularity in computational biology is necessary for encoding more complex models since models can be constructed. Our approach allows a clear separation of the different modeling frameworks via comp submodels and defining the interfaces between the submodels. This paper proposes a new scheme for encoding DFBA models in a standard way. This scheme has been implemented in two different tools, demonstrating the exchangeability and reproducibility of our approach on various examples models like diauxic growth in *E.coli*. `iBioSim` and `sbmlUtils` are freely available for download and offer the necessary infrastructure for anyone to develop DFBA models using the proposed scheme.

Currently, the proposed approach supports the modeling of DFBA models. However, many additional frameworks are not currently supported. Different types of hybrid model, such as a mixture of differential equations, stochastic processes, and boolean models still need further study. This approach can be potentially used as a mixture of different standards, such as mixing SBML and CellML models. This approach may be challenging since a community consensus is needed for combining different model representations. Another limitation of the proposed

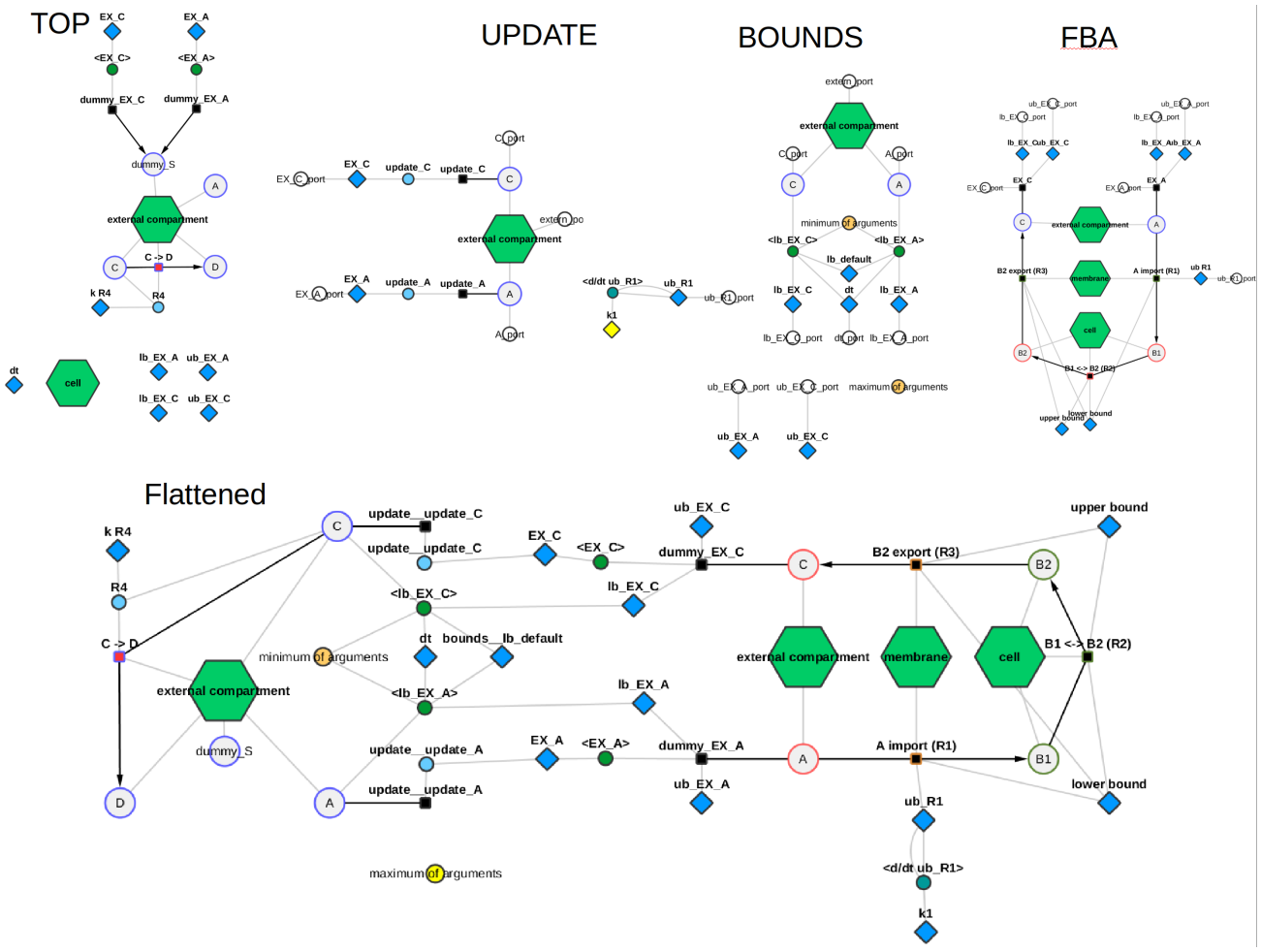
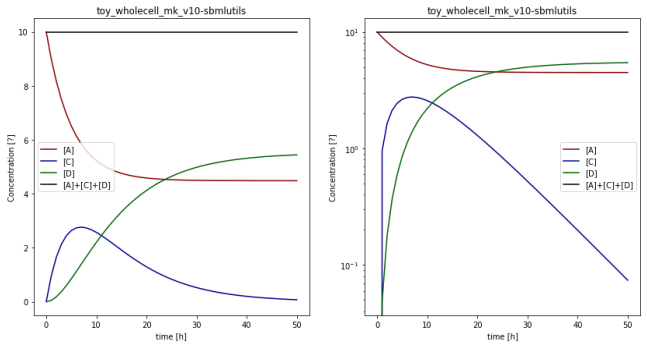
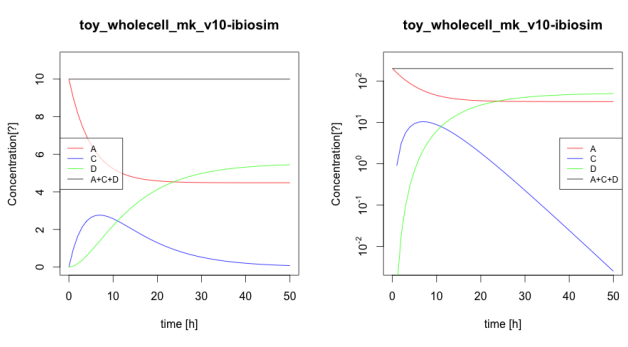


Fig. 3: Detailed schema of the toy wholecell model. The figure shows the components in the TOP, UPDATE, BOUNDS and FBC submodels and their connection via the respective ports. The flattened model shows the connections between the different submodels. The figures are available as interactive graphs in cytoscape as Supplementary Material ?. The figure was created with cy3sbml using the SBML models (König *et al.*, 2012).



(a) This shows the simulation results from sbmlutils for the minimal example.

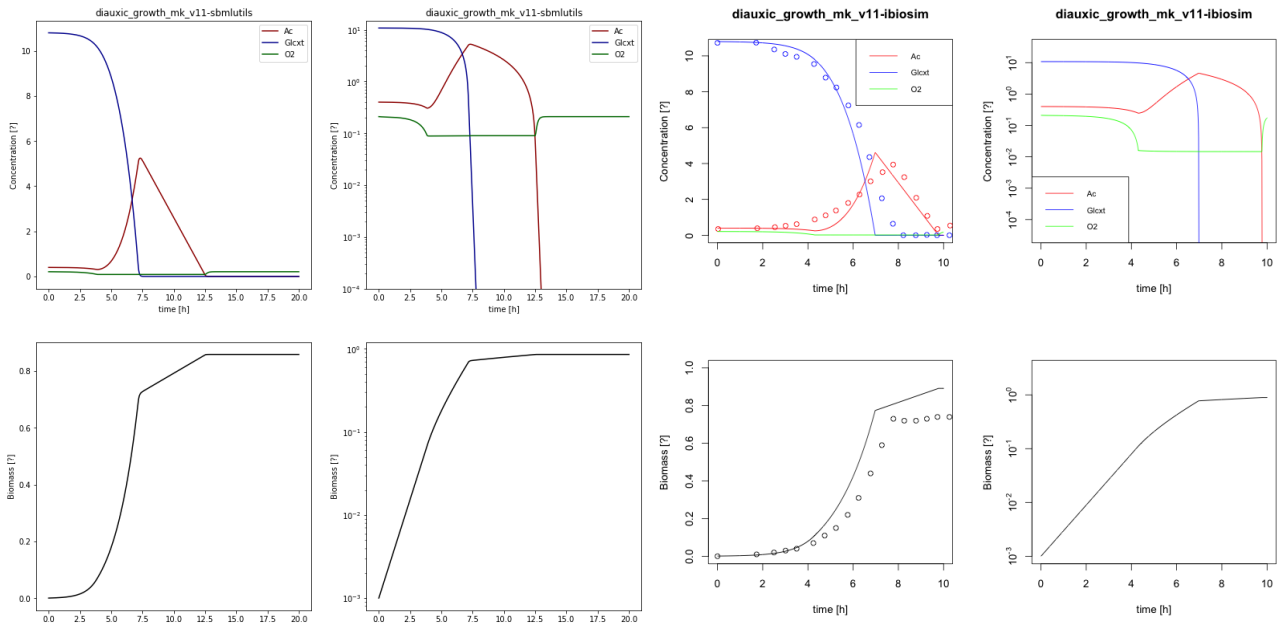


(b) This shows the simulation results from iBioSim for the minimal example.

Fig. 4: These figures show the simulation results for the same model in two different tools. This demonstrates that models can be exchanged by different tools using standards and the results can be reproduced when using the same simulation algorithm.

approach is that the reaction fluxes are updated in a fixed rate using the SOA algorithm.

Most DFBA models are stiff and small time steps are required for stability, making the SOA approach computationally expensive. Another disadvantage of the SOA approach is that it requires a fixed time step that has to be small enough to give accurate results. Future directions are to explore adaptive time steps for executing the DFBA with SOA and



(a) This is the simulation results for the diauxic growth of *E. Coli* in `sbmlutils`.

(b) This is the simulation results for the diauxic growth of *E. Coli* simulated in `iBioSim`.

Fig. 5: This plot shows the results for the model representing diauxic growth in *E. coli*. The model is able to reproduce the general behavior captured from experiment data. There is an exponential cell growth while glucose is present in the model, but when the cell runs out of glucose, growth slows down and is affected mostly by oxygen. However, when the cell runs out of glucose and oxygen, growth diminishes significantly.

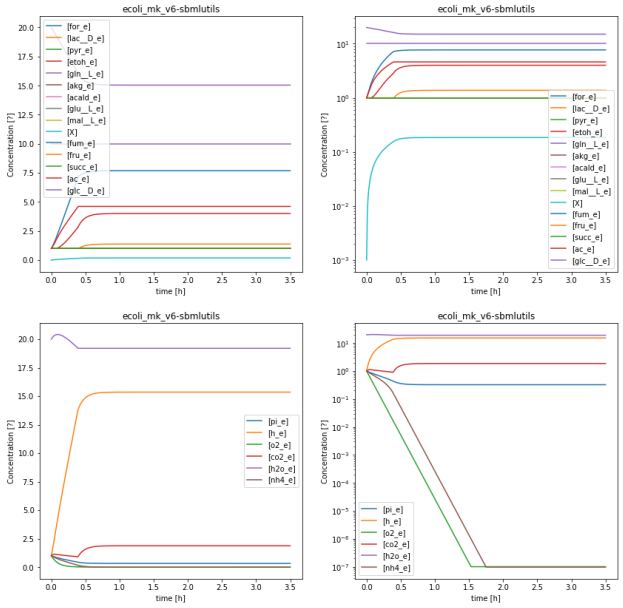


Fig. 6: We demonstrate that the proposed approached can be used in larger models, such as the *E. coli* model described in the paper. This model shows one of the problems of DFBA, where multiple solutions that realize an objection function is obtained. While `sbmlutils` can find a solution for the model, `iBioSim` runs into an unfeasible solution.

alternative DFBA approaches, such as DOA or DA and extending our scheme to encode such models.

So far, only small to medium-size DFBA models have been encoded in our proposed approach. For future work, we will encode genome-scale metabolic models. This would allow us to assess the scalability and performance of the proposed approach.

Acknowledgements

The authors would like to thank Frank Bergmann and Ilya Kiselev for participating in the discussions for this work.

Funding

The authors of this work are supported by the National Science Foundation under Grants CCF-1218095 and CCF-1748200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. MK is supported by the Federal Ministry of Education and Research (BMBF, Germany) within the research network Systems Medicine of the Liver (LiSyM) (grant number 031L0054).

- Scheme for DFBA encoding in SBML

References

Bergmann, F. T., Adams, R., Moodie, S., Cooper, J., Glont, M., Golebiewski, M., Hucka, M., Laibe, C., Miller, A. K., Nickerson, D. P., *et al.* (2014). Combine archive and omex format: one file to share all information to reproduce a modeling project. *BMC bioinformatics*, **15**(1), 369.

Bordbar, A., Monk, J. M., King, Z. A., and Palsson, B. O. (2014). Constraint-based models predict metabolic and associated cellular functions. *Nature reviews. Genetics*, **15**(2), 107.

Choi, K., Medley, J. K., Cannistra, C., Konig, M., Smith, L., Stocking, K., and Sauro, H. M. (2016). Tellurium: A python based modeling and reproducibility platform for systems biology. *bioRxiv*.

Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., Hoops, S., Keating, S., Kell, D. B., Kerrien, S., Lawson, J., Lister, A., Lu, J., Machne, R., Mendes, P., Pocock, M., Rodriguez, N., Villeger, A., Wilkinson, D. J., Wimalaratne, S., Laibe, C., Hucka, M., and Le Novère, N. (2011). Controlled vocabularies and semantics in systems biology. *Molecular Systems Biology*, **7**(1).

Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013). Cobrapy: Constraints-based reconstruction and analysis for python. *BMC systems biology*, **7**(1), 74.

Gomez, J. A., HÃ¶ffner, K., and Barton, P. I. (2014). Dfbalab: a fast and reliable matlab code for dynamic flux balance analysis. *BMC bioinformatics*, **15**, 409.

Hanly, T. J. and Henson, M. A. (2011). Dynamic flux balance modeling of microbial co-cultures for efficient batch fermentation of glucose and xylose mixtures. *Biotechnology and bioengineering*, **108**(2), 376–385.

Hedley, W. J., Nelson, M. R., Bellivant, D., and Nielsen, P. F. (2001). A short introduction to CellML. *Philos. T. Roy. Soc. A*, **359**(1783), 1073–1089.

Hjersted, J. L., Henson, M. A., and Mahadevan, R. (2007). Genome-scale analysis of saccharomyces cerevisiae metabolism and ethanol production in fed-batch culture. *Biotechnology and bioengineering*, **97**(5), 1190–1204.

Höffner, K., Harwood, S. M., and Barton, P. I. (2013). A reliable simulator for dynamic flux balance analysis. *Biotechnology and bioengineering*, **110**(3), 792–802.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., and et. al. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4), 524–531.

Karr, J. R., Sanghvi, J. C., Macklin, D. N., Gutschow, M. V., Jacobs, J. M., Bolival, Benjamin, J., Assad-Garcia, N., Glass, J. I., and Covert, M. W. (2012). A whole-cell computational model predicts phenotype from genotype. *Cell*, (2), 389–401.

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, **22**(3), 437 – 467.

Kitano, H. (2002). Computational systems biology. *Nature*, **420**(6912), 206–210.

König, M. (2017). sbmlutils: python utilities for sbml.

König, M., Dräger, A., and Holzhütter, H.-G. (2012). Cysbml: a cytoscape plugin for sbml. *Bioinformatics*, **28**(18), 2402–2403.

Le Novere, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M. I., Wimalaratne, S. M., *et al.* (2009). The systems biology graphical notation. *Nature biotechnology*, **27**(8), 735–741.

Lequeux, G., Beauprez, J., Maertens, J., Van Horen, E., Soetaert, W., Vandamme, E., and Vanrolleghem, P. A. (2010). Dynamic metabolic flux analysis demonstrated on cultures where the limiting substrate is changed from carbon to nitrogen and vice versa. *BioMed Research International*, **2010**.

Luo, R.-Y., Liao, S., Tao, G.-Y., Li, Y.-Y., Zeng, S., Li, Y.-X., and Luo, Q. (2006). Dynamic analysis of optimality in myocardial energy metabolism under normal and ischemic conditions. *Molecular systems biology*, **2**(1).

Madsen, C., Myers, C. J., Patterson, T., Roehner, N., Stevens, J. T., and Winstead, C. (2012). Design and test of genetic circuits using iBioSim. *Design and Test of Computers, IEEE*, **29**(3), 32–39.

Mahadevan, R., Edwards, J. S., and Doyle, F. J. (2002). Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, **83**(3), 1331–1340.

Meadows, A. L., Karnik, R., Lam, H., Forestell, S., and Snedecor, B. (2010). Application of dynamic flux balance analysis to an industrial escherichia coli fermentation. *Metabolic engineering*, **12**(2), 150–160.

Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., and Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry*, **49**(15), 3216–3224. PMID: 20225868.

Olivier, B. G. and Bergmann, F. T. (2015). The systems biology markup language (sbml) level 3 package: Flux balance constraints. *Journal of Integrative Bioinformatics*, **12**(2), 269.

Pizarro, F., Varela, C., Martabit, C., Bruno, C., Pérez-Correa, J. R., and Agosin, E. (2007). Coupling kinetic expressions and metabolic networks for predicting wine fermentations. *Biotechnology and bioengineering*, **98**(5), 986–998.

Savinell, J. M. and Palsson, B. O. (1992). Network analysis of intermediary metabolism using linear optimization. i. development of mathematical formalism. *Journal of theoretical biology*, **154**(4), 421–454.

Smith, L. P., Hucka, M., Hoops, S., Finney, A., Ginkel, M., Myers, C. J., Moraru, I., and Liebermeister, W. (2015). Sbml level 3 package: Hierarchical model composition, version 1 release 3. *Journal of Integrative Bioinformatics*, **12**(2), 268.

Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, **42**(3), 563 – 585.

Varma, A. and Palsson, B. O. (1994). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Applied and environmental microbiology*, **60**(10), 3724–3731.

Varma, A., Boesch, B. W., and Palsson, B. O. (1993). Biochemical production capabilities of escherichia coli. *Biotechnology and bioengineering*, **42**(1), 59–73.

Waltemath, D., Adams, R., Bergmann, F. T., Hucka, M., Kolpakov, F., Miller, A. K., Moraru, I. I., Nickerson, D., Sahle, S., Snoep, J. L., *et al.* (2011). Reproducible computational biology experiments with sed-ml-the simulation experiment description markup language. *BMC systems biology*, **5**(1), 1.

Waltemath, D., Karr, J. R., Bergmann, F. T., Chelliah, V., Hucka, M., Krantz, M., Liebermeister, W., Mendes, P., Myers, C. J., Pir, P., *et al.* (2016). Toward community standards and software for whole-cell modeling. *IEEE Transactions on Biomedical Engineering*, **63**(10), 2007–2014.

Watanabe, L. H. and Myers, C. J. (2014). Hierarchical stochastic simulation algorithm for sbml models of genetic circuits. *Frontiers in Bioengineering and Biotechnology*, **2**, 55.