BMC
Systems Biology

SOFTWARE   **Open Access**

COBRApy: COnstraints-Based Reconstruction and Analysis for Python

Ali Ebrahim[1], Joshua A Lerman[1], Bernhard O Palsson[1] and Daniel R Hyduke[1,2*]

Abstract

Background: COnstraint-Based Reconstruction and Analysis (COBRA) methods are widely used for genome-scale modeling of metabolic networks in both prokaryotes and eukaryotes. Due to the successes with metabolism, there is an increasing effort to apply COBRA methods to reconstruct and analyze integrated models of cellular processes. The COBRA Toolbox for MATLAB is a leading software package for genome-scale analysis of metabolism; however, it was not designed to elegantly capture the complexity inherent in integrated biological networks and lacks an integration framework for the multiomics data used in systems biology. The openCOBRA Project is a community effort to promote constraints-based research through the distribution of freely available software.

Results: Here, we describe COBRA for Python (COBRApy), a Python package that provides support for basic COBRA methods. COBRApy is designed in an object-oriented fashion that facilitates the representation of the complex biological processes of metabolism and gene expression. COBRApy does not require MATLAB to function; however, it includes an interface to the COBRA Toolbox for MATLAB to facilitate use of legacy codes. For improved performance, COBRApy includes parallel processing support for computationally intensive processes.

Conclusion: COBRApy is an object-oriented framework designed to meet the computational challenges associated with the next generation of stoichiometric constraint-based models and high-density omics data sets.

Availability: http://opencobra.sourceforge.net/

Keywords: Genome-scale, Network reconstruction, Metabolism, Gene expression, Constraint-based modeling

Background

Constraint based modeling approaches have been widely applied in the field of microbial metabolic engineering [1,2] and have been employed in the analysis [3-5] and, to a lesser extent, modeling of transcriptional [6-8] and signaling [9] networks. And, we've recently developed a method for integrated modeling of gene expression and metabolism on the genome scale [10].

The popularity of these approaches is due, in part, to the fact that they facilitate analysis of biological systems in the absence of a comprehensive set of parameters. Constraints-based approaches focus on employing data-driven physicochemical and biological constraints to enumerate the set of feasible phenotypic states of a reconstructed biological network in a given condition. These constraints include

compartmentalization, mass conservation, molecular crowding [11], thermodynamic directionality [12], and transcription factor activity [13]. More recently, transcriptome data have been used to reduce the size of the set of computed feasible states [14-17]. Because constraints-based models are often underdetermined they may provide multiple mathematically-equivalent solutions to a specific question – these equivalent solutions must be assessed with experimental data for biological relevance [18].

We have previously published the COBRA Toolbox [19] for MATLAB to provide systems biology researchers with a high-level interface to a variety of methods for constraint-based modeling of genome-scale stoichiometric models of cellular biochemistry. The COBRA Toolbox is being increasingly recognized as a standard framework for constraint-based modeling of metabolism [20]. While the COBRA Toolbox has gained widespread use and become a powerful piece of software, it was not designed to cope with modeling complex biological processes

* Correspondence: hyduke@ucsd.edu
[1]Department of Bioengineering, University of California, San Diego, 9500 Gilman Drive MC0412, La Jolla, 92093-0412, CA, USA
[2]Biological Engineering Department, Utah State University, 4105 Old Main Hill, Logan, UT 84322-4105, USA

outside of metabolism or for integrated analyses of omics data, and requires proprietary software to function. To drive COBRA research through this avalanche of omics and model increasingly complex biological processes [10], we have developed an object-oriented implementation of core COBRA Toolbox functions using the Python programming language. COBRA for Python (COBRApy) provides access to commonly used COBRA methods in a MATLAB-free fashion.
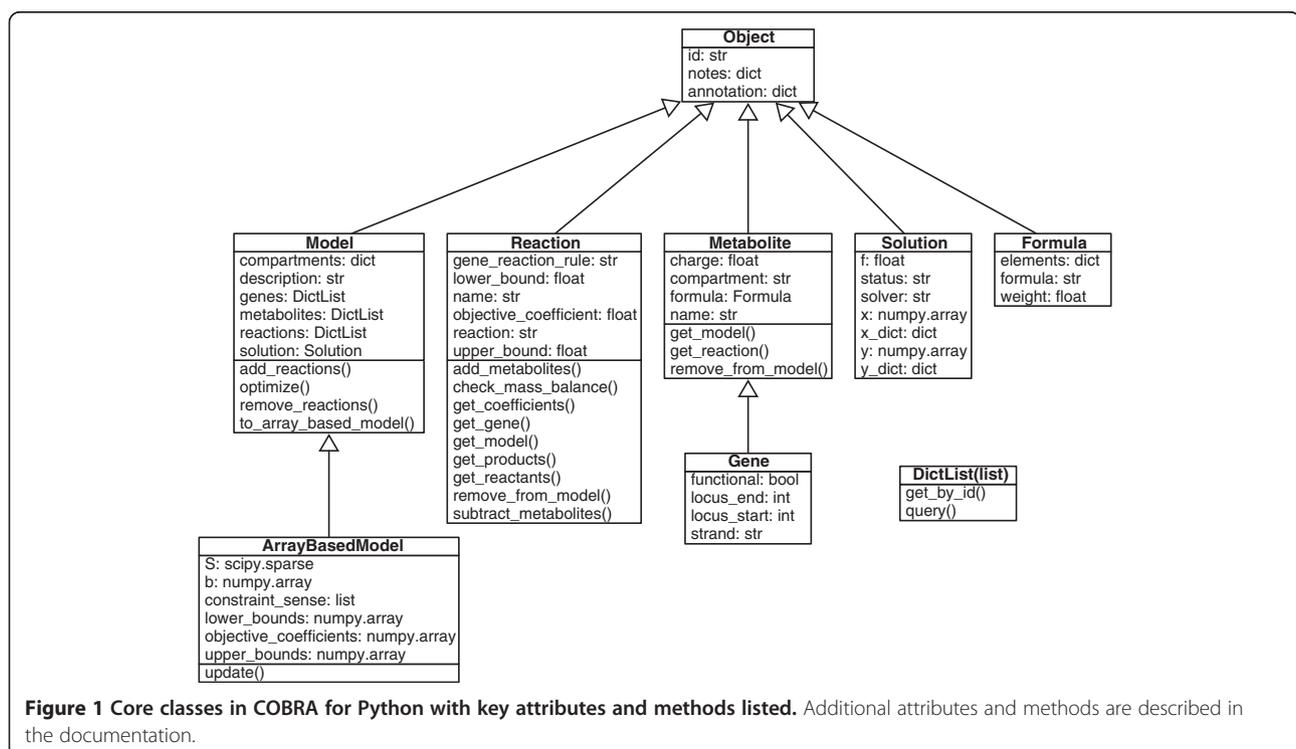
Implementation

The core capabilities of COBRApy are enabled by a set of classes (Figure 1) that represent organisms (Model), biochemical reactions (Reaction), and biomolecules (Metabolite and Gene). The core code is accessible through either Python or Jython (Python for Java). COBRApy contains: (1) cobra.io: an input/output package for reading / writing SBML [21] models and reading / writing COBRA Toolbox MATLAB structures. (2) cobra.flux_analysis: a package for performing common FBA operations, including gene deletion and flux variability analysis [18]. (3) cobra.topology: a package for performing structural analysis – the current version contains the reporter metabolites algorithm of Patil & Nielsen [22]. (4) cobra. test: a suite of unit tests and test data. (5) cobra.solvers: interfaces to linear optimization packages. And, (6) cobra.mlab: an interface to the COBRA Toolbox for MATLAB.

Results and discussion

COBRApy is a software package for constraints-based modeling that is designed to accommodate the increasing complexity of biological processes represented with COBRA methods. Like the COBRA Toolbox, COBRApy provides core COBRA modeling capabilities in an extendible and accessible fashion. However, COBRApy employs an object oriented programming approach that is more amenable to representing increasingly complex models of biological networks. Moreover, COBRApy inherits numerous benefits from the Python language, and allows the integration of models with databases and other sources of high-throughput data. Additionally, COBRApy does not require commercial software for commonly used COBRA operations whereas the COBRA Toolbox depends on MATLAB. As the COBRA Toolbox is in wide use, it will likely be used as a development and analysis platform for years to come. To take advantage of legacy and future modules written for the COBRA Toolbox, COBRApy includes a module for directly interacting with the COBRA Toolbox (cobra.mlab) and support for reading and writing COBRA Toolbox MATLAB structures (cobra.io.mat).

In recent years, a number of software packages have been developed that employ stoichiometric constraint-based modeling approaches [23], such as Cell Net Analyzer [24], FASIMU [25], PySCeS-CBM [26], the Raven Toolbox [27], and the Systems Biology Research Tool [28].

Figure 1 Core classes in COBRA for Python with key attributes and methods listed. Additional attributes and methods are described in the documentation.

While there is overlap in functionality between some of packages and COBRApy (Table 1), the other packages do not currently support the next generation models of metabolism and expression (ME-Models) [10] nor integration with the COBRA Toolbox for MATLAB. It is worth noting that the other software packages often contain a rich variety of functionality that is targeted towards other research topics, such as modeling signaling networks [24]. COBRApy continues the COBRA Toolbox's tradition of providing an interactive / programmable framework for constraints-based modeling and is a new initiative of The openCOBRA Project [29]. Software downloads, tutorials, forums, and detailed documentation are available at http://opencobra.sourceforge.net.

Core classes: model, metabolite, reaction, & gene

The core classes of COBRApy are Model, Metabolite, Reaction, and Gene. The Model class serves as a container for a set of chemical Reactions, including associated Metabolites and Gene products (Figure 2a). Within a Model, Metabolites are modified by one or more Reactions that may be spontaneous or catalyzed by one or more Genes (Figure 2b). The underlying genetic requirements for a Reaction to be active in a Model are supplied as a Boolean relationship [19], where each gene is referred to by a unique identifier. During the construction of a Model, the Model and the Reactions, Metabolites, and Genes are explicitly aware of each other. For example, given a Metabolite, it is possible to use the get_reaction() method to determine in which Reactions this Metabolite participates. Then the genes associated with these Reactions may be accessed by the Reaction.get_gene() method.

The object-based design of COBRApy provides the user with the ability to directly access attributes for each object (Figure 1), whereas with the COBRA Toolbox for MATLAB biological entities and their attributes are each contained within separate lists. For example, with COBRApy, a Metabolite object provides information about its chemical Formula and associated biochemical Reactions, whereas, with the COBRA Toolbox for MATLAB, one must query multiple tables to access these values and modify multiple tables to update these values.

Key capabilities

COBRApy comes with variants of the published metabolic network models (M-Models) for *Salmonella enterica* Typhimurium LT2 [30] and Escherichia coli K-12 MG1655 [31]. These models can be loaded with the cobra.test.create_test_model function; with *S.* Typhimurium LT2 being the default model. Additionally, COBRApy can read SBML-formatted models [32] downloaded from a variety of sources, such as the Model SEED [33] and the BioModels database [34].

A common operation performed with M-Models is to optimize for the maximum flux through a specific reaction in a defined growth medium [35]. The *S.* Typhimurium LT2 model comes with a variety of media whose compositions are specified in the model's media_compositions attribute. Here, we initialize the Model's boundary conditions to mimic the minimal MgM medium [36] and then perform a linear optimization to calculate the maximal flux through the Reaction biomass_iRR1083_metals. Biomass_iRR1083_metals is a reaction that approximates the materials required to support *S.* Typhimurium LT2 growth in a minimal medium where approximately 0.3 grams dry weight *S.* Typhimurium LT2 are produced per hour. It is important to note that cellular composition can vary as a function of growth rate [37], therefore, for biological accuracy it may be necessary to construct a new biomass reaction if the simulated, or experimentally-observed, growth rate is substantially different [10,38].

Flux balance analysis of M-Models has enjoyed substantial success in qualitative analyses of gene essentiality [30]. These studies used simulations to identify which genes or synthetic lethal gene-pairs are essential for biomass production in a given condition. The lists of essential genes and synthetic lethal gene-pairs may then be targeted to inhibit microbial growth or excluded from manipulation when constructing designer strains [39]. COBRApy provides functions for automating single and double gene deletion studies in the cobra.flux_analysis module.

Table 1 Features of available constraints-based programming packages

Software package	GUI	FBA	FVA	M-models	ME-models	SBML	Strain design	Language
Cell net analyzer	+	+	+	+		+	+	MATLAB
COBRA toolbox		+	+	+		+	+	MATLAB
COBRApy		+	+	+	+	+	*	Python
fasimu		+	+	+		+		bash
PySCeS-CBM		+	+	+		+		Python
Raven	+	+		+		+		MATLAB
Systems biology research tool		+	+	+				Java

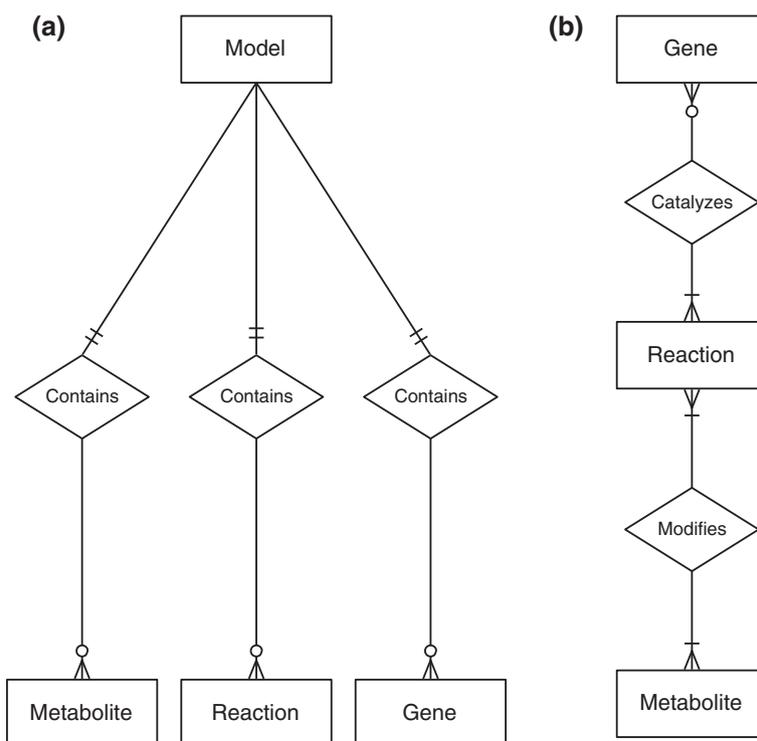+: feature is available; *: feature is accessible through the COBRA Toolbox.

Figure 2 Entity relationship diagrams for core classes in COBRApy. (a) A Model contains Metabolites, Reactions, and Genes. **(b)** A Reaction may be catalyzed by 0 or more Genes. Reactions catalyzed by 0 Genes are spontaneous. A Reaction may be catalyzed by different sets of Genes. Reactions modify 1 or more Metabolites. A Reaction that modifies only 1 metabolite is an external boundary condition. A Metabolite may be modified by many different Reactions.

Because of the presence of equivalent alternative optima in constraint based-simulations of metabolism [18], many reactions may theoretically be able to carry a wide range of flux for a given simulation objective. Flux variability analysis (FVA) is often used to calculate the amount of flux a reaction can carry while still simulating the maximum flux through the objective function subject to a specified tolerance. Flux variability analyses can be used to identify problems in model structure [40] or 'pinch-points' in a metabolic network. COBRApy provides automated functions for FVA in the cobra. flux_analysis.variability module.

Advanced capabilities

Because whole genome double deletion and FVA simulations can be time intensive with a single CPU, we have provided a function that uses Parallel Python [41] to split the simulation across multiple CPUs for multicore machines. Additionally, there are a wide range of legacy operations that are present in the COBRA Toolbox that can be accessed using mlabwrap [42]. MATLAB is only necessary for accessing codes written in the COBRA Toolbox for MATLAB; it is not necessary to run the majority of COBRApy functions.

Conclusions

COBRApy is a constraint-based modeling package that is designed to accommodate the biological complexity of the next generation of COBRA models [10] and provides access to commonly used COBRA methods, such as flux balance analysis [35], flux variability analysis [18], and gene deletion analyses [43]. Through the mlabwrap module it is possible to use COBRApy to call many additional COBRA methods present in the COBRA Toolbox for MATLAB [19]. As part of The openCOBRA Project, COBRApy serves as an enabling framework for which the community can develop and contribute application specific modules.

Availability and requirements

Project name: COBRApy version 0.2.1
Project home page: http://opencobra.sourceforge.net
Operating systems: Platform independent, including Java
Programming language: Python (≥ 2.6) / Jython (≥ 2.5)
Other requirements:
***Python*:** libSBML $\geq 5.5.0$ [32]. Currently supported linear programming solvers: GLPK [44] through PyGLPK 0.3 [45], IBM ILOG/CPLEX Optimization Studio ≥ 12.4 (IBM Corporation, Armonk, New York), and Gurobi ≥ 5.0 (Gurobi Optimization, Inc., Houston, TX, USA).

- [Optional] Numpy ≥ 1.6.1 & Scipy ≥ 0.10.1 [46] for ArrayBasedModel, MoMA, and double_deletion analysis.
- [Optional] Parallel python [41] for parallel processing.
- [Optional] To directly interface with the COBRA Toolbox for MATLAB it is necessary to install mlabwrap [42], the COBRA Toolbox [29], and a version of MATLAB (Mathworks, Natick, Massachusetts, U.S.A.) that is compatible with the COBRA Toolbox.

Jython: JSBML ≥ 0.8 [32,47].

- Currently supported linear programming solvers: GLPK for Java 1.0.22 [48], IBM ILOG/CPLEX Optimization Studio ≥ 12.4, and Gurobi ≥ 5.0.
- The COBRA Toolbox for MATLAB and ArrayBasedModel are not currently accessible from Jython.

License: GNU GPL version 3 or later.

Abbreviations
COBRA: COnstraint-Based Reconstruction and Analysis; FBA: Flux balance analysis; FVA: Flux variability analysis; M-Model: Metabolic network model.

Competing interests
This software was used by DRH, JAL, and BOP to develop the method that is the subject of a provisional patent application U.S. Provisional Application Serial No. 61/644,924 filed on May 9, 2012 entitled "Method for in silico modeling of gene product expression and metabolism".

Authors' contributions
DRH conceived COBRA for Python. AE, JAL, and DRH contributed to various aspects of development and testing. All authors read and approved the final manuscript.

Acknowledgements

References
1. Feist AM, Palsson BO: **The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli.** *Nat Biotechnol* 2008, **26**:659–667.
2. Kim IK, Roldao A, Siewers V, Nielsen J: **A systems-level approach for metabolic engineering of yeast cell factories.** *FEMS Yeast Res* 2012, **12**:228–248.
3. Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP: **Network component analysis: reconstruction of regulatory signals in biological systems.** *Proc Natl Acad Sci USA* 2003, **100**:15522–15527.
4. Hyduke DR, Jarboe LR, Tran LM, Chou KJ, Liao JC: **Integrated network analysis identifies nitric oxide response networks and dihydroxyacid dehydratase as a crucial target in Escherichia coli.** *Proc Natl Acad Sci USA* 2007, **104**:8484–8489.
5. Tran LM, Hyduke DR, Liao JC: **Trimming of mammalian transcriptional networks using network component analysis.** *BMC Bioinforma* 2010, **11**:511.
6. Covert MW, Palsson BO: **Transcriptional regulation in constraints-based metabolic models of Escherichia coli.** *J Biol Chem* 2002, **277**:28058–28064.
7. Gianchandani EP, Joyce AR, Palsson BO, Papin JA: **Functional states of the genome-scale Escherichia coli transcriptional regulatory system.** *PLoS Comput Biol* 2009, **5**:e1000403.
8. Thiele I, Jamshidi N, Fleming RM, Palsson BO: **Genome-scale reconstruction of Escherichia coli's transcriptional and translational machinery: a knowledge base, its mathematical formulation, and its functional characterization.** *PLoS Comput Biol* 2009, **5**:e1000312.
9. Hyduke DR, Palsson BO: **Towards genome-scale signalling-network reconstructions.** *Nat Rev Genet* 2010, **11**:297–307.
10. Lerman JA, Hyduke DR, Latif H, Portnoy VA, Lewis NE, Orth JD, Schrimpe-Rutledge AC, Smith RD, Adkins JN, Zengler K, Palsson BO: **In silico method for modelling metabolism and gene product expression at genome scale.** *Nat Commun* 2012, **3**:929.
11. Vazquez A, Beg QK, Demenezes MA, Ernst J, Bar-Joseph Z, Barabasi AL, Barabasi AL, Boros LG, Oltvai ZN: **Impact of the solvent capacity constraint on E. coli metabolism.** *BMC Syst Biol* 2008, **2**:7.
12. Henry CS, Broadbelt LJ, Hatzimanikatis V: **Thermodynamics-based metabolic flux analysis.** *Biophys J* 2007, **92**:1792–1805.
13. Gama-Castro S, Jimenez-Jacinto V, Peralta-Gil M, Santos-Zavaleta A, Penaloza-Spinola MI, Contreras-Moreira B, Segura-Salazar J, Muniz-Rascado L, Martinez-Flores I, Salgado H, Bonavides-Martinez C, Abreu-Goodger C, Rodriguez-Penagos C, Miranda-Rios J, Morett E, Merino E, Huerta AM, Trevino-Quintanilla L, Collado-Vides J: **RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation.** *Nucleic Acids Res* 2008, **36**:D120–D124.
14. Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng TY, Moody DB, Murray M, Galagan JE: **Interpreting expression data with metabolic flux models: predicting Mycobacterium tuberculosis mycolic acid production.** *PLoS Comput Biol* 2009, **5**:e1000489.
15. Frezza C, Zheng L, Folger O, Rajagopalan KN, MacKenzie ED, Jerby L, Micaroni M, Chaneton B, Adam J, Hedley A, Kalna G, Tomlinson IP, Pollard PJ, Watson DG, Deberardinis RJ, Shlomi T, Ruppin E, Gottlieb E: **Haem oxygenase is synthetically lethal with the tumour suppressor fumarate hydratase.** *Nature* 2011, **477**:225–228.
16. Bordbar A, Mo ML, Nakayasu ES, Schrimpe-Rutledge AC, Kim YM, Metz TO, Jones MB, Frank BC, Smith RD, Peterson SN, Hyduke DR, Adkins JN, Palsson BO: **Model-driven multi-omic data analysis elucidates metabolic immunomodulators of macrophage activation.** *Mol Syst Biol* 2012, **8**:558.
17. Hyduke DR, Lewis NE, Palsson BO: **Analysis of omics data with genome-scale models of metabolism.** *Mol Biosyst* 2013, **9**:167–174.
18. Mahadevan R, Schilling CH: **The effects of alternate optimal solutions in constraint-based genome-scale metabolic models.** *Metab Eng* 2003, **5**:264–276.
19. Schellenberger J, Que R, Fleming RM, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, Kang J, Hyduke DR, Palsson BO: **Quantitative prediction of cellular metabolism with constraint-based models: the COBRA toolbox v2.0.** *Nat Protoc* 2011, **6**:1290–1307.
20. Medema MH, van Raaphorst R, Takano E, Breitling R: **Computational tools for the synthetic design of biochemical pathways.** *Nat Rev Microbiol* 2012, **10**:191–202.
21. Hucka M, Finney A, Bornstein BJ, Keating SM, Shapiro BE, Matthews J, Kovitz BL, Schilstra MJ, Funahashi A, Doyle JC, Kitano H: **Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project.** *Syst Biol (Stevenage)* 2004, **1**:41–53.
22. Patil KR, Nielsen J: **Uncovering transcriptional regulation of metabolism by using metabolic network topology.** *Proc Natl Acad Sci USA* 2005, **102**:2685–2689.
23. Lakshmanan M, Koh G, Chung BK, Lee DY: **Software applications for flux balance analysis.** *Brief Bioinform* 2012.

24. Klamt S, von Kamp A: **An application programming interface for Cell NetAnalyzer.** *Biosystems* 2011, **105:**162–168.
25. Hoppe A, Hoffmann S, Gerasch A, Gille C, Holzhutter HG: **FASIMU: flexible software for flux-balance computation series in large metabolic networks.** *BMC Bioinforma* 2011, **12:**28.
26. Olivier BG, Rohwer JM, Hofmeyr JH: **Modelling cellular systems with PySCeS.** *Bioinformatics* 2005, **21:**560–561.
27. Agren R, Liu L, Shoaie S, Vongsangnak W, Nookaew I, Nielsen J: **The RAVEN toolbox and its use for generating a genome-scale metabolic model for penicillium chrysogenum.** *PLoS Comput Biol* 2013, **9:**e1002980.
28. Wright J, Wagner A: **The systems biology research tool: evolvable open-source software.** *BMC Syst Biol* 2008, **2:**55.
29. *The openCOBRA Project.* http://opencobra.sourceforge.net
30. Thiele I, Hyduke DR, Steeb B, Fankam G, Allen DK, Bazzani S, Charusanti P, Chen FC, Fleming RM, Hsiung CA, De Keersmaecker SC, Liao YC, Marchal K, Mo ML, Ozdemir E, Raghunathan A, Reed JL, Shin SI, Sigurbjornsdottir S, Steinmann J, Sudarsan S, Swainston N, Thijs IM, Zengler K, Palsson BO, Adkins JN, Bumann D: **A community effort towards a knowledge-base and mathematical model of the human pathogen Salmonella Typhimurium LT2.** *BMC Syst Biol* 2011, **5:**8.
31. Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, Palsson BO: **A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011.** *Mol Syst Biol* 2011, **7:**535.
32. Bornstein BJ, Keating SM, Jouraku A, Hucka M: **LibSBML: an API library for SBML.** *Bioinformatics* 2008, **24:**880–881.
33. Henry CS, Dejongh M, Best AA, Frybarger PM, Linsay B, Stevens RL: **High-throughput generation, optimization and analysis of genome-scale metabolic models.** *Nat Biotechnol* 2010, **28:**977–982.
34. Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, Chelliah V, Li L, He E, Henry A, Stefan MI, Snoep JL, Hucka M, Le Novere N, Laibe C: **BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models.** *BMC Syst Biol* 2010, **4:**92.
35. Orth JD, Thiele I, Palsson BO: **What is flux balance analysis?** *Nat Biotechnol* 2010, **28:**245–248.
36. Beuzon CR, Banks G, Deiwick J, Hensel M, Holden DW: **pH-dependent secretion of SseB, a product of the SPI-2 type III secretion system of Salmonella typhimurium.** *Mol Microbiol* 1999, **33:**806–816.
37. Schaechter M, Maaloe O, Kjeldgaard NO: **Dependency on medium and temperature of cell size and chemical composition during balanced grown of Salmonella typhimurium.** *J Gen Microbiol* 1958, **19:**592–606.
38. Pramanik J, Keasling JD: **Stoichiometric model of Escherichia coli metabolism: incorporation of growth-rate dependent biomass composition and mechanistic energy requirements.** *Biotechnol Bioeng* 1997, **56:**398–421.
39. Burgard AP, Pharkya P, Maranas CD: **Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization.** *Biotechnol Bioeng* 2003, **84:**647–657.
40. Schellenberger J, Lewis NE, Palsson BO: **Elimination of thermodynamically infeasible loops in steady-state metabolic models.** *Biophys J* 2011, **100:**544–553.
41. *Parallel Python.* http://parallelpython.com
42. *mlabwrap.* http://mlabwrap.sourceforge.net
43. Lewis NE, Nagarajan H, Palsson BO: **Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods.** *Nat Rev Microbiol* 2012, **10:**291–305.
44. *GLPK.* http://www.gnu.org/software/glpk
45. *PyGLPK (not python-glpk).* http://www.tfinley.net/software/pyglpk
46. *SciPy / NumPy.* http://scipy.org
47. Drager A, Rodriguez N, Dumousseau M, Dorr A, Wrzodek C, Le Novere N, Zell A, Hucka M: **JSBML: a flexible Java library for working with SBML.** *Bioinformatics* 2011, **27:**2167–2168.
48. *GLPK for Java.* http://glpk-java.sourceforge.net