

2016

PSAMM: A Portable System for the Analysis of Metabolic Models

Jon Lund Steffensen

Keith Dufault-Thompson

See next page for additional authors

Creative Commons License



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

Follow this and additional works at: https://digitalcommons.uri.edu/cmb_facpubs

Citation/Publisher Attribution

Steffensen, J. L., Dufault-Thompson, K., and Zhang, Y. (2016). PSAMM: A Portable System for the Analysis of Metabolic Models. *PLoS Computational Biology*, 12(2). doi: 10.1371/journal.pcbi.1004732

This Article is brought to you for free and open access by the Cell and Molecular Biology at DigitalCommons@URI. It has been accepted for inclusion in Cell and Molecular Biology Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons@etal.uri.edu.

Authors

Jon Lund Steffensen, Keith Dufault-Thompson, and Ying Zhang

RESEARCH ARTICLE

PSAMM: A Portable System for the Analysis of Metabolic Models

Jon Lund Steffensen, Keith Dufault-Thompson, Ying Zhang*

Department of Cell and Molecular Biology, College of the Environment and Life Sciences, University of Rhode Island, Kingston, Rhode Island, United States of America

* yingzhang@uri.edu



 OPEN ACCESS

Citation: Steffensen JL, Dufault-Thompson K, Zhang Y (2016) PSAMM: A Portable System for the Analysis of Metabolic Models. *PLoS Comput Biol* 12(2): e1004732. doi:10.1371/journal.pcbi.1004732

Editor: Thomas Dandekar, University of Wurzburg, Biocentre, GERMANY

Received: May 19, 2015

Accepted: January 5, 2016

Published: February 1, 2016

Copyright: © 2016 Steffensen et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All results generated by the methods described in this paper are contained within the Supporting Information files. The PSAMM software is available in the open source Git repository at GitHub (<https://zhanglab.github.io/psamm/>). The YAML format model files are available at <https://github.com/zhanglab/psamm-model-collection>.

Funding: This work was supported by the USDA National Institute of Food and Agriculture, Hatch fund, RI0015-H002, Accession No. 1007170, and a pilot project funded by the Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number 2 P20 GM103430. The funders

Abstract

The genome-scale models of metabolic networks have been broadly applied in phenotype prediction, evolutionary reconstruction, community functional analysis, and metabolic engineering. Despite the development of tools that support individual steps along the modeling procedure, it is still difficult to associate mathematical simulation results with the annotation and biological interpretation of metabolic models. In order to solve this problem, here we developed a Portable System for the Analysis of Metabolic Models (PSAMM), a new open-source software package that supports the integration of heterogeneous metadata in model annotations and provides a user-friendly interface for the analysis of metabolic models. PSAMM is independent of paid software environments like MATLAB, and all its dependencies are freely available for academic users. Compared to existing tools, PSAMM significantly reduced the running time of constraint-based analysis and enabled flexible settings of simulation parameters using simple one-line commands. The integration of heterogeneous, model-specific annotation information in PSAMM is achieved with a novel format of YAML-based model representation, which has several advantages, such as providing a modular organization of model components and simulation settings, enabling model version tracking, and permitting the integration of multiple simulation problems. PSAMM also includes a number of quality checking procedures to examine stoichiometric balance and to identify blocked reactions. Applying PSAMM to 57 models collected from current literature, we demonstrated how the software can be used for managing and simulating metabolic models. We identified a number of common inconsistencies in existing models and constructed an updated model repository to document the resolution of these inconsistencies.

Author Summary

The broad application of genome-scale metabolic modeling has made it a useful technique for tackling fundamental questions in biological research and engineering. Today over 100 models have been constructed for organisms that carry out a diverse array of metabolic activities spanning all three kingdoms of life. These models, however, have been curated independently following different conventions. The maintenance of model consistency has been challenging due to the lack of consensus in model representation and the absence

had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

of integrated modeling software for associating mathematical simulations with the annotation and biological interpretation of metabolic models. To solve this problem, we developed a new software package, PSAMM, and a new model format that incorporates heterogeneous, model-specific annotation information into modular representations of model definitions and simulation settings. PSAMM provides significant advances in standardizing the workflow of model annotation and consistency checking. Compared to existing tools, PSAMM supports more flexible configurations and is more efficient in running constraint-based simulations. All functions of PSAMM are freely available for academic users and can be downloaded from a public Git repository (<https://zhanglab.github.io/psamm/>) under the GNU General Public License.

Introduction

The GENome-scale Models (GEMs) of metabolic networks have broad applications in biological research and engineering [1]. Models have been developed for organisms of all three kingdoms of life [2–5] and have been used to simulate a wide variety of metabolic processes, such as photo- and chemo-autotrophic carbon fixation [6,7], fermentation [8], and the production of specific organic compounds [9]. GEMs can be applied in theoretical research to predict gene essentiality [10,11], simulate the thermo-tolerance of bacterial strains [12], and study the structural and functional evolution of metabolic networks [13]. They can also be used in practical studies to identify drug targets [14,15], illustrate the mechanism of human diseases [16], and to optimize the production of compounds of industrial significance [17–19]. By connecting genome annotations with the mathematical simulation of reaction networks, GEMs are particularly applicable for integrating heterogeneous datasets from high-throughput studies [20], such as the profiles of transcriptional regulation [21,22] and measurement of carbon isotope labeling [23].

Specialized MATLAB toolboxes have been released over the past two decades to support the mathematical simulations of metabolic networks. The COBRA Toolbox is a collection of widely used, open source tools. It includes diverse implementations of constraint-based modeling algorithms and has attracted a large number of user contributions from the modeling community [24]. This toolbox, however, is restricted to the MATLAB environment and requires users to maintain paid licenses from MathWorks Inc. The COBRAPy software is a more recent implementation of the COBRA Toolbox functions using the Python programming language and Jython, a Java implementation of Python [25]. Like the COBRA Toolbox, COBRAPy is released as a toolbox rather than a software package. Therefore, knowledge about the Python programming language is required for users to efficiently set up operations in COBRAPy.

Other tools have been developed to support the annotation and visualization of metabolic networks. ModelSEED is a web-based platform that supports automated reconstruction of metabolic models from genome annotations [26,27]. It links protein functions with an internal reaction database and is associated with the SEED functional annotation database and the RAST genome annotation pipeline [28]. In contrast to the COBRA Toolbox, this platform is focused on the reconstruction instead of the mathematical simulation of GEMs. The automated pipeline of ModelSEED permits direct construction of a draft model from genome annotation. However, the draft model still requires extensive manual curations, and manually editing the draft models is not currently supported [27]. The RAVEN Toolbox supports semi-automatic reconstruction and visualization of genome-scale models [29]. It uses information from the KEGG database [30] and, similar to the COBRA Toolbox, can only be applied under the

MATLAB environment. Finally, Pathway Tools is another software package that supports pathway annotation and visualization [31]. It uses pathway information from the MetaCyc database [32] and has recently been extended to include functions for flux balance analysis. Unlike the open source software mentioned above, Pathway Tools is released under a restricted license agreement, but it is free to use for academic users.

Despite the technological advances in supporting individual steps of network reconstruction and mathematical simulation, challenges remain in maintaining the quality of metabolic reconstructions and in associating the representation of mathematical problems with the annotation and biological interpretation of metabolic models. Human intervention and iterative manual examinations are required in various steps, such as defining biomass functions, assigning reaction directionality, establishing boundary conditions, and filling novel reaction gaps [26,33]. Frequently, these processes are carried out either manually or using customized tools that were developed by individual model curators. However, due to the absence of standardized procedures for quality checking and version tracking of iterative model annotations, the curation and manual editing of metabolic models are prone to introducing inconsistencies [34]. These inconsistencies may lead to false predictions of model viability, and as demonstrated in a recent study, may hinder the evaluation of new modeling approaches [35–37].

Another problem that prevents the effective curation and consistency checking of GEMs is the disassociations between mathematical and biological representations of model metadata. Despite a broad adoption of the SBML format, it is not designed to incorporate detailed annotations of the genes, reactions, compounds, and pathways. Two strategies have been implemented to help address this limitation. The first is to use user-defined, model-specific or toolbox-specific tags in SBML files (e.g. the <notes> tag in COBRA-compliant SBML format). These tags are not part of the standard SBML specifications and are not recognized by the standardized SBML parsers (e.g. libSBML [38]). The second strategy employs tables of annotation data. These tables are customized to represent specific models, but they are disconnected from the simulation problems defined in SBML files, and an automated parsing of these tables is impossible due to the lack of convention in table organizations. In either case, the mathematical representation of simulation problems is separated from the biological annotation of model components. As a result, the consistency between genome annotation and the definition of simulation settings have to be maintained by individual model curators. This again is prone to introducing misrepresentations in GEMs.

An additional disadvantage of the SBML format is the lack of modularity in network representations. The SBML definition of GEMs is composed of two major lists: *listOfSpecies*, which defines all of the metabolites in a GEM; and *listOfReactions*, which defines all of the reactions and their simulation settings in a single modeling state, including flux bounds, flux values, and objective coefficients. Since the definition of simulation settings interleaves with the definition of static reaction features, SBML does not support reusing the static model definitions. Anytime a new simulation is released on the same base model, information about the static features of metabolites and reactions will need to be duplicated in a new SBML file. Hence, the SBML format is not compatible with a modular organization of modeling components because it mixes the static model properties with the dynamic simulation settings.

Here, we present a Portable System for the Analysis of Metabolic Models (PSAMM), an open source software package that was implemented to support the iterative curation of GEMs by connecting model annotations with mathematical simulations. A novel model format was developed in PSAMM using the YAML language to integrate detailed annotations of model components and to provide flexibilities in both the data format (i.e. support direct referencing of annotation tables or the YAML-based list format) and the data content (i.e. support parsing of both standardized and model-specific data fields). The new YAML format assembles model

definition and simulation settings into independent and reusable modules (S1 Text). By reducing the amount of characters used for data structure definition, it streamlines the content of the real data and enables tracking and management of GEMs with conventional line-based version control systems like Git [39]. Applying PSAMM, we constructed an updated repository of 57 published models using the modular representation of the YAML format, and we further demonstrated the importance of model formatting and consistency checking through identifying and correcting potential inconsistencies in existing GEMs.

Results

Comparison of PSAMM to Existing Tools

PSAMM was developed as an integrated software package that supports calling simulation functions using simple one-line commands. This distinguishes PSAMM from existing metabolic modeling toolboxes, which require the users to go through individual steps of model loading, linear programming solver selection, model optimization, and simulation results mapping. An example of PSAMM commands is given (S1 Fig), in which a thermodynamically constrained flux balance analysis (tfBA) was performed through a simple one-line command. The result of the analysis contains metadata about the model name, model version, objective function, reaction fluxes, and reaction equations. In contrast, it takes three steps to set up a tfBA problem in COBRA, and the output is presented in a complex data structure that is dissociated from the biological interpretations of the model. The PSAMM implementations were also much more efficient than the COBRA toolbox functions. While it took only 7.15 seconds to solve a tfBA problem in PSAMM, the same process on the same model took 194.73 seconds in COBRA.

A comparison of the functions in the PSAMM software package with the functions in the COBRA and RAVEN toolboxes is provided in the S1 Table. Besides the availability of diverse functions and implementations of model import/export, model checking, and constraint-based analysis, a unique feature of PSAMM is the availability of easily accessible help information that assists the users in selecting program functions and parameter values. This feature is enabled because PSAMM integrates the various functions under a common framework of two universal programs, *psamm-model* and *psamm-import*. The help information can be accessed in PSAMM by adding the *-h* or *--help* option following any program functions listed in the S1 Table. In contrast, RAVEN and COBRA do not provide an integrated interface to the available functions but instead require the users to know the names of individual functions in order to retrieve help information. Moreover, to systematically evaluate the efficiency of PSAMM implementations, we recorded the running time of common operations in PSAMM and COBRA on the *Escherichia coli* model iJO1366 [40] (Materials and Methods). The results suggested that PSAMM is overall much more efficient than COBRA (Fig 1). For the computationally intensive functions like flux variable analysis (FVA), robustness analysis (Robustness), and thermodynamically constrained flux balance analysis (tfBA), PSAMM ranges from 9 to 25 times faster than implementations in the COBRA toolbox. For the FBA function, PSAMM appears to have an overhead of about 3 seconds on top of the time required for the problem-solving step (Fig 1 inset). This overhead is for the reading of YAML model files, which is also included in calculating the running times of FVA, Robustness, and tfBA, but solving FBA problem in PSAMM (0.03 seconds) is slightly more efficient than the same step in COBRA (0.08 seconds). The running time for RAVEN was not plotted in Fig 1 because many of the functions are unavailable in the RAVEN toolbox (S1 Table). A detailed record of the running times for available functions in COBRA, RAVEN, and PSAMM is provided in the S2 Table.

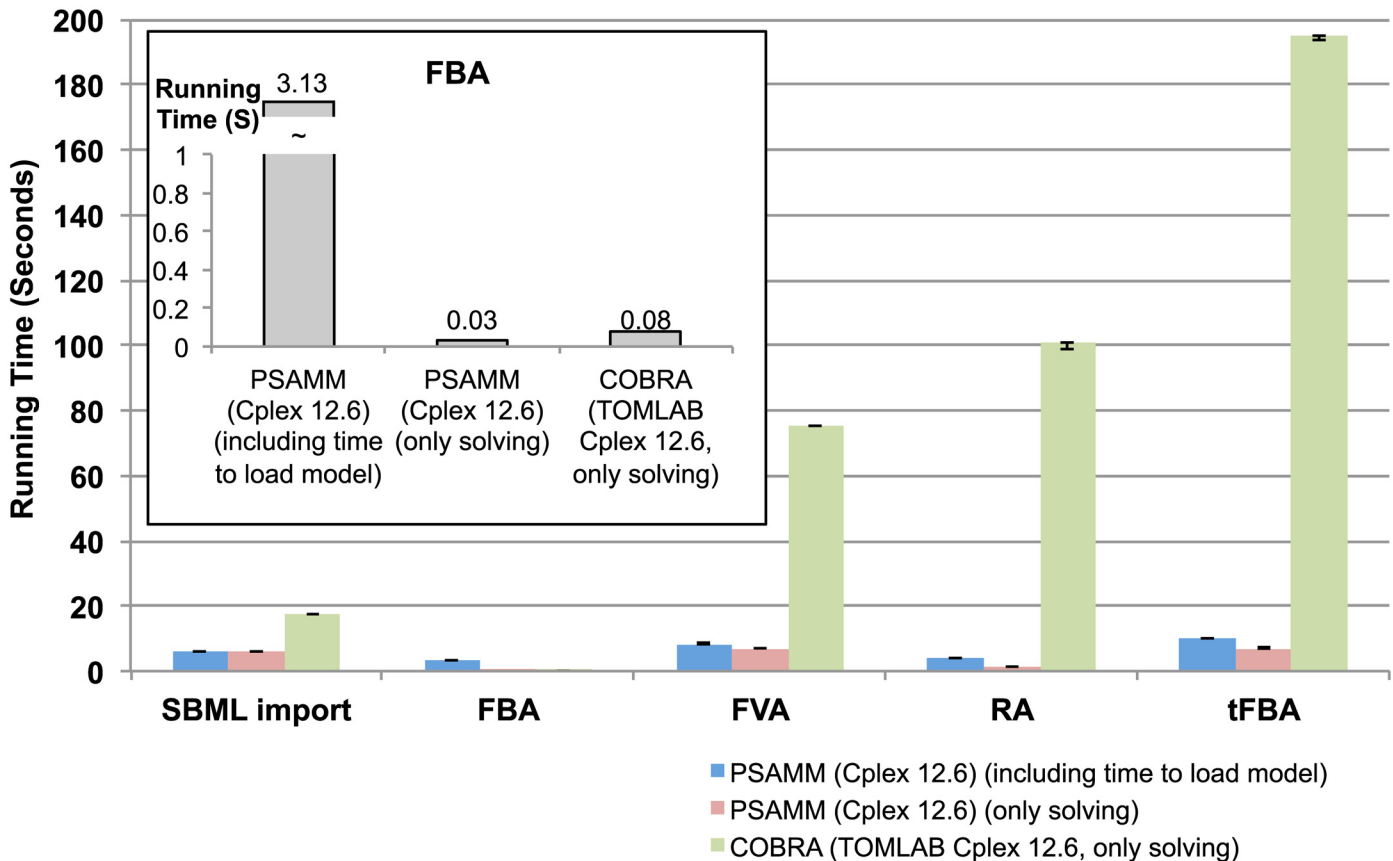


Fig 1. Comparing the running time of different functions in PSAMM and COBRA. The running time for both the PSAMM overall function (blue) and the PSAMM problem-solving steps (red) were calculated, and the running time for COBRA only included the problem-solving step (green). Each value represents a median of seven simulation runs using the same specifications, and the error bars indicate the 25th and the 75th percentiles.

doi:10.1371/journal.pcbi.1004732.g001

Overall, PSAMM is more efficient than the COBRA and RAVEN toolboxes in carrying out constraint-based analysis of GEMs.

PSAMM Implementation

The PSAMM software package includes five main components: user interface, model input/output, internal model representations, model checking/simulation, and linear programming utilities (Fig 2). These components are interconnected with one another to form the internal workflow of model importing and model optimizations in PSAMM. Below, we provided a detailed description of these functions.

User interface. The user interface was implemented as two command-line programs that provide entry points to PSAMM functions. The first program, named *psamm-model*, is the main interface for model consistency checking, constraint-based simulation and automatic gap analysis. The commands that can be specified with *psamm-model* encapsulate both commonly used metabolic modeling procedures (e.g. *fa*, *fva*, and *gapfill*) and PSAMM-specific procedures, such as *masscheck*, *fluxcheck*, *randomsparse*, *search*, and *console* (Table 1 and S1 Table). The second program, named *psamm-import*, facilitates the recognition and conversion of multiple GEM formats, including the SBML format that is most widely used as a convention for model distributions in current literature, the Microsoft Excel format that is frequently used to store the annotation of compound and reaction features, and the JSON format that is

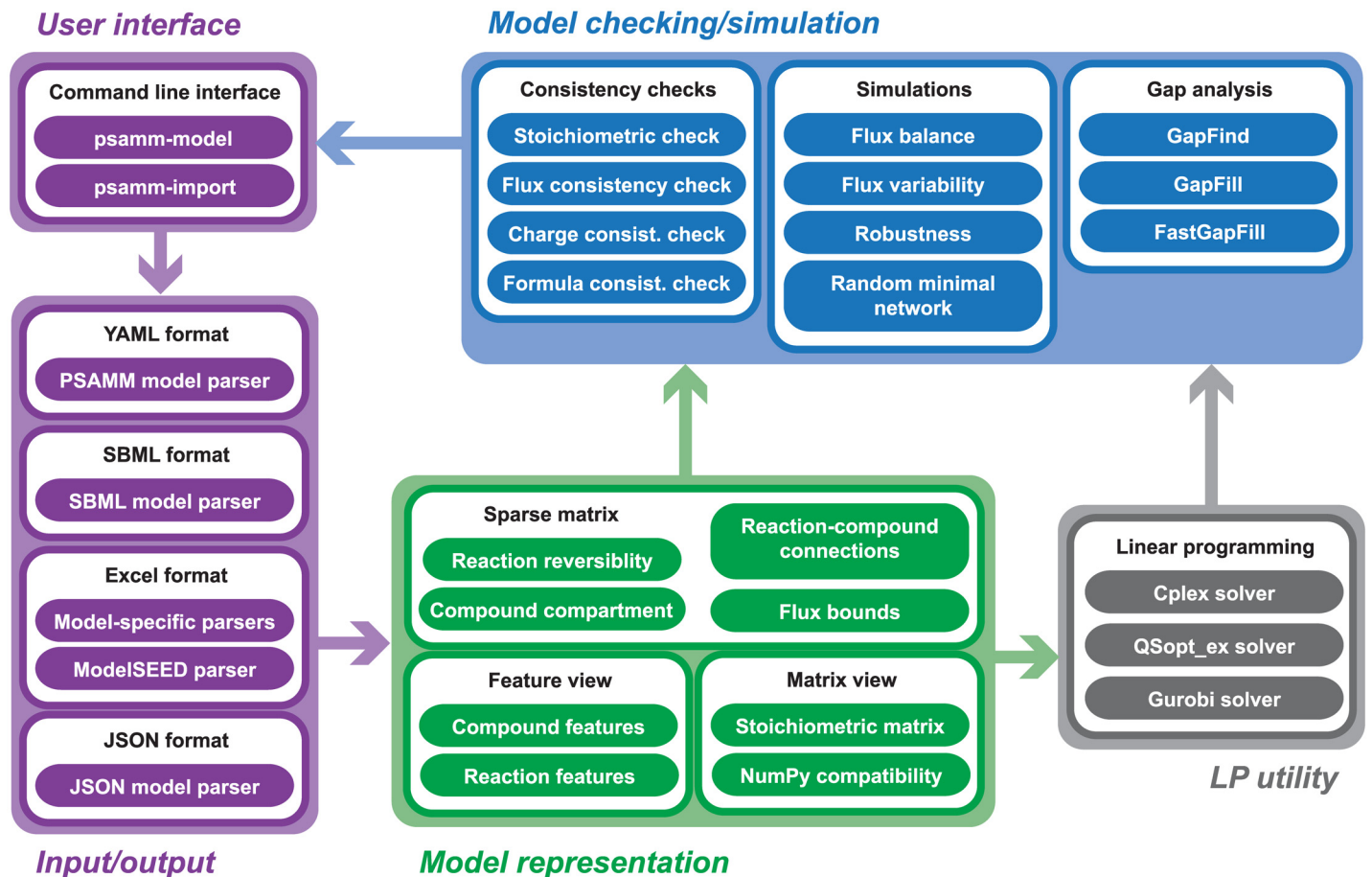


Fig 2. Overview of the internal workflow in PSAMM. The five main components include: (1) user interface, (2) model input/output, (3) model representation, (4) linear programming utilities, and (5) model checking/simulation. Connections among these components form the internal workflow of PSAMM.

doi:10.1371/journal.pcbi.1004732.g002

developed within the openCOBRA project. The *psamm-import* program converts the SBML, Excel, and JSON files into a YAML-based format that was designed as a new convention in PSAMM for integrating user-defined annotations with mathematical simulations of metabolic models (S1 Text). The *psamm-model* and *psamm-import* programs interface directly with the model input/output module, which permits the import of model files into internal model representations and the export of model information into standard SBML, YAML, and Excel files.

Model input/output. The model input/output module includes functions for importing model files in the SBML, Excel, JSON and YAML formats, and exporting models in the YAML, Excel, and SBML formats. PSAMM supports importing both the standard SBML specifications and the COBRA-compliant SBML format [24], and the parsing of model files was based on a built-in SBML parser instead of relying on external dependencies like the *libSBML* [38]. This reduction of external dependencies can simplify the installation process and hence increase the portability of PSAMM. Moreover, the development of a built-in parser enables PSAMM to recognize model-specific data entries that are not part of the standard SBML specification. Using the built-in parser, users can check if a model is strictly compliant with the SBML specifications and import it by setting the *sbml-strict* option in the *psamm-import* program. The *sbml-strict* option will only check for parameters defined in the standardized SBML specifications [41].

Table 1. List of commands supported in the *psamm-model* program. Additional parameters can be specified for the commands, for example to select a specific implementation among multiple algorithms, enable/disable thermodynamic constraints, or select linear programming solvers, etc. Details about these parameters are available through the *-h* or *--help* options for each command.

	Command	Function
Model Consistency Checking	<i>masscheck</i>	Stoichiometric (mass) consistency check
	<i>fluxcheck</i>	Check for blocked reactions
	<i>formulacheck</i>	Chemical formula balance check
	<i>chargecheck</i>	Charge balance check
Constraint-based Simulations	<i>fba</i>	Flux balance analysis
	<i>fva</i>	Flux variability analysis
	<i>robustness</i>	Robustness analysis
	<i>randomsparse</i>	Find random minimal network
Gap Analysis	<i>gapfill</i>	Gap analysis using the GapFind and GapFill algorithms
	<i>fastgapfill</i>	Fast gap analysis using the fastGapFill algorithm
Model Interfacing	<i>search</i>	Looking up the detailed features of specific compounds or reactions.
	<i>console</i>	Opening an interactive Python or iPython session with the model loaded in memory. Providing a programming interface for the customized inspection and analysis of GEMs.
	<i>sbmlexport</i>	Export the model to SBML format.
	<i>excelexport</i>	Export the model to Excel format.

doi:10.1371/journal.pcbi.1004732.t001

However, to load constraint-based modeling parameters, such as the reaction bound parameters defined in the COBRA-compliant format, the users can use the more flexible *sbml* option. This later option is also extended to automatically resolve some common inconsistencies in existing models, such as permitting decimal numbers in a SBML level that supports only integers in stoichiometry or recognizing a special suffix “*_b*” that indicates exchange compounds (Table 2). When a potential inconsistency is resolved automatically, *psamm-import* will print a warning message so that users can examine if the SBML parsing was performed correctly.

Besides SBML files, many GEMs were published in the format of Microsoft Excel tables that contain the definition of compound and reaction features. However, so far no standard has been specified for the Excel format. To incorporate these models, PSAMM provides a customized interface that enables importing of model-specific Excel tables and the Excel tables generated from automated model reconstruction pipeline in ModelSEED [26]. Additionally, PSAMM supports the import of JSON models either from local files or from the BiGG database [42]. The import functions for both JSON and Excel formats were implemented in the *psamm-import* program, which converts models into the PSAMM-specific YAML format (S1 Text).

Internal model representations. The PSAMM internal model representations provide three distinct data structures for the organization of model information. At the hub is a sparse matrix, which stores a global mapping between compounds and reactions. The sparse matrix can include information regarding stoichiometry, compound compartments, reaction reversibility, and flux limits. It provides a central framework for the management and analysis of metabolic models and can be queried into two data structures that support the implementation of PSAMM functions. In the first data structure, a feature view can be created to incorporate additional information about the model metadata, including compound features (e.g. formulas, charges, etc.) and reaction features (e.g. names, subsystem classifications, equations, etc.). In the second data structure, a matrix view can be initiated to represent a full-scale stoichiometric matrix. The diverse data structures in PSAMM enables the integration of heterogeneous data regarding the user-defined model annotations as well as the mathematical simulations of

Table 2. List of models that had inconsistencies in SBML syntax.

Model	Organism	Inconsistencies
AbyMBEL891	<i>Acinetobacter baumannii</i>	Decimals in stoichiometric values is not supported in Level 1 SBML specification
SpoMBEL1693	<i>Schizosaccharomyces pombe</i>	Decimals in stoichiometric values is not supported in Level 1 SBML specification
iJN746	<i>Pseudomonas putida</i>	Undefined compound references in the biomass reaction
iNJ661m	<i>Mycobacterium tuberculosis</i>	COBRA-compliant, using a special suffix “_b” instead of defining “boundaryCondition” based on SBML standard
iNJ661v	<i>Mycobacterium tuberculosis</i>	COBRA-compliant, using a special suffix “_b” instead of defining “boundaryCondition” based on SBML standard
iSR432	<i>Clostridium thermocellum</i>	COBRA-compliant, using a special suffix “_b” instead of defining “boundaryCondition” based on SBML standard
iTZ479	<i>Thermotoga maritima</i>	COBRA-compliant, using a special suffix “_b” instead of defining “boundaryCondition” based on SBML standard
iTH366	<i>Plasmodium falciparum</i>	COBRA-compliant, using a special suffix “_b” instead of defining “boundaryCondition” based on SBML standard
iRC1080	<i>Chlamydomonas reinhardtii</i>	Inconsistencies between reversibility and “LOWER_BOUND” settings
iRS1563	<i>Zea mays</i>	Inconsistencies between reversibility and “LOWER_BOUND” settings
iRS1597	<i>Arabidopsis thaliana</i>	Inconsistencies between reversibility and “LOWER_BOUND” settings
iRsp1095	<i>Rhodobacter sphaeroides</i>	Nutrient not provided in the simulation environment
IMA871	<i>Aspergillus Niger</i>	Accumulation of an artificial <i>M_BIOMASS</i> compound
IMA945	<i>Salmonella enterica</i>	Mislabeling of compound compartments
iSyn731	<i>Synechocystis sp. PCC 6803</i>	Potential inconsistencies in model conversions

doi:10.1371/journal.pcbi.1004732.t002

GEMs. These internal model representations allow for the formulation of modeling problems by interfacing with the linear programming (LP) utility and the model checking/simulation modules.

Linear programming (LP) utilities. The LP utilities were implemented as an internal component that supports the definition of LP problems and the parsing of LP solutions. Three LP solvers are supported in the PSAMM utilities: the IBM ILOG Cplex Optimizer, the Gurobi Optimizer, and the QSOpt_ex Rational Solver ([Materials and Methods](#)). While Cplex and Gurobi provide approximated solutions for both LP and mixed integer LP (MILP) problems, QSOpt_ex computes exact solutions only for LP problems. The QSOpt_ex Rational Solver is supported because it has recently been suggested to provide a more accurate solution for constraint-based modeling problems [35]. Applying both the approximated solvers and the exact solvers under the same framework of PSAMM model representations will enable a standardized comparison of these two approaches. The LP utilities form a backend module in PSAMM that communicates with the Application Programming Interface (API) of external solvers and provides LP solutions to the model checking/simulation functions.

Model checking/simulation. The model checking/simulation module includes diverse user functions for model consistency checking, constraint-based simulation, and gap analysis ([Fig 2](#)). It was developed based on the internal feature representation of GEMs to support the listing of simulation results along with the biological annotations. Thus, mathematical simulation results can be analyzed under a user-friendly interface that permits the direct association of the numeric values with the biologically meaningful representations of biochemical equations ([S1 Fig](#)).

The constraint-based simulation functions include flux balance analysis (FBA) [43,44], flux variability analysis (FVA) [45], robustness analysis [46], and random minimal network analysis [47] ([Table 1](#) and [S1 Table](#)). Each function was implemented to support two universal options:

`--loop-removal`, which determines whether thermodynamic constraints should be applied to the LP problem, and `--solver`, which allows the users to select an LP solver while running the simulation problem. The random minimal network analysis (*randomsparse*) is a unique function in PSAMM that is not available in any existing toolboxes. Each round of the *randomsparse* performs a series of random gene deletions, which result in a minimal metabolic network where all genes are essential for biomass production. The median running time for 100 rounds of *randomsparse* analysis on iJO1366 [40] is 1,664 seconds (S2 Table). In case gene association information is not available, the command can also be applied directly to reactions, which results in a minimal network where all reactions are essential. The median running time for 100 rounds of the reaction-based *randomsparse* analysis on iJO1366 is 3,524 seconds (S2 Table). Hence, each round of the successive random deletions takes less than or close to 30 seconds.

The consistency checking and gap analysis functions were set up to assist with the model curation and to examine stoichiometric, flux, charge, and formula consistencies of individual reactions (S1 Table). While the stoichiometric check can provide a global estimation of whether all reactions in a model are mass balanced (Materials and Methods), the charge and formula checks can examine the compound element and charge balances in individual reactions and can only be applied when the compound charge and formula is defined in the GEM. The flux check function was implemented to identify reactions that are unable to carry a non-zero flux under any modeling constraints. It can be used to reveal disconnected areas of the model and to ensure unbiased interpretation of the model simulations (Materials and Methods). The gap analysis functions were implemented based on two existing algorithms, GapFind/GapFill [48] and fastGapFill [49]. Both functions can recommend a list of new reactions that should be added into a given model. Depending on the scope of information included in the YAML reaction database (S1 Text), the prediction of gap reactions may be based on a list of enzymatic reactions in the broader reaction database, or artificial exchange reactions that point out the missing links between gap compounds and the modeling environment.

The PSAMM YAML Format of Model Representation

A new model format was implemented in PSAMM based on YAML, a serialization language that is designed to support both computational parsing and human readability of complex data structures [50]. The YAML format has several new features that distinguish it from the SBML format of constraint-based metabolic models. First, while SBML strictly couples the model definition with a single simulation condition, YAML allows the users to freely combine model definitions with simulation setups in a modular form. Hence, a model component can be reused in multiple simulation setups simply by using the *include* function (Fig 3 and S1 Text). In contrast, to publish the simulation results of multiple modeling conditions in the SBML format, one has to build duplicated instances of the entire model definition to introduce variability in simulation setups. Second, while SBML is suboptimal in working with the widely-used version control systems like Git [51], the YAML format is fully compatible and supports tracking of model edits with these systems. This is because YAML, by applying line breaks and whitespaces as meaningful structural information, can be tracked in a way similar to programming scripts using the longest common sequence (LCS) algorithm that is commonly used by Git [52]. In contrast, SBML does not have a convention of breaking lines, and therefore the LCS line-based version tracking will have difficulties in pinpointing the exact location of changes in the SBML files. Despite the development of new algorithms to support the comparison of SBML model files, the efficient and reliable tracking of changes in the SBML format is still challenging [34,51]. Third, while SBML relies on the verbose representation of markup tags, the YAML

(a) model.yaml

```
name: Escherichia coli model
biomass: Biomass_Ecoli

reactions:
- include: glycolysis.yaml (I)
- include: tca-cycle.tsv (II)
- include: biomass.yaml (III)
# ...

compounds:
- include: compounds.yaml (IV)

model:
- include: model_rxn.tsv (V)

media:
- include: media.yaml (VI)

limits:
- include: limits.yaml (VII)
```

(I) glycolysis.yaml

```
- id: ENO
name: Enolase
genes:
- b2779
equation: '|2pg[c]| <=> |h2o[c]| + |pep[c]|'
subsystem: Glycolysis/Gluconeogenesis
ec: 4.2.1.11

- id: F6PA
# ...
```

(II) tca-cycle.tsv

```
id      equation
ACONTa  |cit[c]| <=> |acon-C[c]| + |h2o[c]|
ACONTb  |acon-C[c]| + |h2o[c]| <=> |icit[c]|
AKGDH   |akg[c]| + |coa[c]| + |nad[c]| => ...
CITL    |cit[c]| => |ac[c]| + |oaa[c]|
CS      |accoa[c]| + |h2o[c]| + |oaa[c]| => ...
FRD2    |fum[c]| + |mq18[c]| => |mqn8[c]| + ...
FRD3    |2dmmq18[c]| + |fum[c]| => |2dmmq8[c]| ...
```

(III) biomass.yaml

```
- id: Biomass_Ecoli
equation:
reversible: no
left:
- id: ala-L
value: 0.49915
- id: arg-L
value: 0.28742
- id: asn-L
# ...
```

(IV) compounds.yaml

```
- id: ac
name: Acetate
formula: C2H3O2
charge: -1

- id: glc-D
name: D-Glucose
formula: C6H12O6
charge: 0
# ...
```

(V) model_rxn.tsv

```
ACONTa
ACONTb
AKGDH
CITL
CS
FRD2
FRD3
ENO
Biomass_Ecoli
# ...
```

(VI) media.yaml

```
name: Default medium
compartment: e
compounds:
- id: ac

- id: akgl
lower: -10
upper: 100
# ...
```

(VII) limits.yaml

```
- reaction: ENO
upper: 10

- reaction: FRD2
lower: -50
upper: 50
# ...
```

(b)

```
$ git diff
diff --git a/biomass.yaml b/biomass.yaml
index 4f03f36..f8017c8 100644
--- a/biomass.yaml
+++ b/biomass.yaml
@@ -20,7 +20,7 @@
- id: ala-L
value: 0.49915
- id: arg-L
- value: 0.28742
+ value: 0.31944
- id: asn-L
value: 0.234232
- id: atp
```

Fig 3. An illustration of the PSAMM YAML format. (a) This diagram shows an example of the YAML model format, which includes a central model definition (model.yaml) and multiple annotation files. Each box indicates a file with a possible filename indicated above the box, and the text within is a snapshot of the file content. (b) An example showing how changes can be tracked in a PSAMM YAML file (biomass.yaml) using the Git version control system in command line. The text highlighted in red indicates the stoichiometry of the compound arg-L in an old version of the biomass function, while the text highlighted in green indicates the updated value in a new version of the model. Additional examples of applying Git version control on the YAML format are provided in the supplemental materials (S2–S5 Texts).

doi:10.1371/journal.pcbi.1004732.g003

format minimizes the amount of characters used for marking up the structure of the data to streamline the integration of complex data (Fig 3 and S1 Text). Finally, while the parsing of SBML files requires knowledge of predefined markup tags to capture specific data fields, the parsing of YAML files can be achieved by the recognition of line-based data structure rather than relying on the identification of specific markup tags. For example, some features like the compound formula and reaction EC numbers are not an integral part of standardized SBML specifications [41], so they will be invisible to standard SBML parsers (e.g. libSBML [38] or JSBML [53]) even when they are included in a SBML file with user-defined markup tags. However, the same information can be easily represented in YAML files. A standard YAML parser will have no problem recognizing user-defined, non-standard data domains and subsequently integrate it into the presentation of metabolic models. Overall, the YAML format is an optimized solution for connecting manual curation with the mathematical simulation of GEMs.

The PSAMM YAML-based representation of GEMs includes the model name, biomass function, compound/reaction database(s), model reaction lists, reaction flux limits, and growth media. A flexible infrastructure was designed that permits the integration of model definitions either within a centralized file (e.g. model.yaml) or in additional files using the *include* function (Fig 3a). In an example shown in Fig 3a, the reaction database was divided into multiple files based on pathway classification, and both the YAML format (e.g. glycolysis.yaml and biomass.yaml) and the tab-delimited table format (e.g. tca-cycle.tsv) were illustrated. Similarly, the compound annotations can also be represented as YAML files (e.g. compound.yaml) or tab-delimited tables. The reaction and compound files can be used to define a broader database of all possible reactions with additional entries beyond the scope of a given model. For example, when running the *gapfill* or *fastgapfill* functions, users may include a broader reaction database that contains enzymatic reactions that can be potentially applied to close network gaps. In such cases, the model file (e.g. model_rxn.tsv) is used to identify a subset of reactions that are included in the model definition. The other reactions in the broader database will be ignored in model simulations and will only be probed in *gapfill* or *fastgapfill* to predict potential gap filling reactions. Finally, to define the constraints for metabolic simulations, the media file (e.g. media.yaml) is used to identify the exchange reactions, and the limits file (e.g. limits.yaml) is used to set the flux limits for internal reactions. The limits file is defined only when the flux bounds of an internal reaction deviate from the defaults in PSAMM, which automatically assigns a conventional boundary of $[-x, x]$ for reversible reactions and $[0, x]$ for irreversible reactions, where the value x is assigned using an option named *default_flux_limit* in the model.yaml file. When *default_flux_limit* is not assigned, PSAMM will use a default value of $x = 1000$ to define reaction bounds (S1 Text). This feature permits further streamlining of the YAML format to reveal model specific simulation setups. Although PSAMM supports references to model files with user-defined file names that are saved in distributed locations in the file system, it is recommended that users save all files of an individual model within a dedicated directory to facilitate model organization and maintenance. The users may also choose to combine all the model information into a single YAML file by replacing the *include* functions with the actual content of the data files. However, the *include* functions are recommended to maintain modularity of model definition and simulation settings in the YAML representation.

Using the YAML format, annotations of compound and reaction features can be further divided into multiple files that represent logical divisions of cellular compartments or metabolic pathways. Moreover, The annotations can be represented in either a YAML format or a tab-delimited table format. These flexibilities can enable the manual curation of model files and make the model components portable among different simulation conditions. The line-based text representation of YAML files also makes it compatible with broadly-used version control systems like Git, and the YAML format is especially efficient at tracking changes in

large reaction equations, such as the biomass function and the protein, RNA, and lipid synthesis functions (Fig 3b). In contrast, the Excel format is not compatible with version control due to the binary form, and the lack of a standard model definition in Excel files also detracts from its suitability as a GEM format. The SBML files, in spite of a well-defined data structure, do not use line breaks or white spaces as a part of the data structure and hence are suboptimal in working with Git [34]. For example, each list item in the SBML file is frequently represented as a single line that contains multiple data entries describing this item. This will render the version control in Git ineffective because changes on any one entry will cause the version tracking system to highlight an entire line of multiple data entries, and it is difficult for a user to pinpoint the exact location of the change within a long line of data. Moreover, since the markup tags are included as a part of the data presentation, changes in the markup tag definitions during the periodical updates of the SBML specifications will show up in the tracking even though there is no change in the model data.

Examining Model Consistencies Using PSAMM

To test the application of PSAMM in supporting model annotations, we applied the software to 57 published GEMs collected from current literature or open-source model releases (S3 Table). These included the RECON2.04, which was released in the format of a MATLAB data file. All models were first converted to the PSAMM-specific YAML format and then analyzed individually using the stoichiometric and flux checking functions. FBA simulations were also performed using designated biomass function to verify if the published descriptions of model viability can be replicated. When inconsistencies occurred, the causes were identified by manually inspecting the corresponding models. Overall, PSAMM revealed several types of inconsistencies among existing models. These included inconsistencies in the syntax of SBML files, the annotation contents of Excel tables, the stoichiometric balance of metabolic reactions, and the connectivity of metabolic reactions. Below, we provide an overview of the existing inconsistencies identified by PSAMM.

Inconsistencies in SBML syntax. The Systems Biology Markup Language (SBML) community has published three different specifications (levels) of the SBML format. Of the 57 GEMs we analyzed, 56 were released as SBML files, including two level-1 models (AbyMBEL891 [54] and SpoMBEL1693 [3]), three level-3 models (iMM1415 [55], iMO1056 [56], and RECON1 [57]), and 51 level-2 models as defined in their SBML file headings. The SBML models were converted to YAML format using the *psamm-import* program. First, the *sbml-strict* option was used to check if the model follows strict SBML specifications. Then, the *sbml* option was used to incorporate COBRA-compliant variations in SBML syntax. Finally, an FBA simulation was performed using biomass production as the objective function to examine the viability of each model.

Using the *sbml-strict* option of *psamm-import*, inconsistencies were found in the SBML syntax of 15 GEMs (Table 2). These included inconsistencies both in the model specification and in the definition of simulation setups. Three models were not strictly compliant with the standardized SBML specifications, including two models that contain decimal numbers in a SBML level that supports only integers in stoichiometry (AbyMBEL891, SpoMBEL1693) and one model that contains undefined compound species (i.e. *cardiolipin[c]*) in the biomass function (iJN746 [58]). With the *sbml* option, the former inconsistencies were resolved by permitting decimals in stoichiometry for all SBML levels, and the latter was resolved by automatically removing undefined compound species in reaction equations. In both cases, warning messages were given by *psamm-import* so the users can be aware of these modifications. To ensure a more accurate parsing of iJN746, we also manually examined the model to look for potential

production pathways of the undefined cytosolic *cardiolipin[c]* compounds. This manual curation revealed that the missing compounds were actually mislabeled in their compartment settings: while the synthesis functions occurred in periplasmic space, the biomass function referred to a non-existing cytosolic copy. Since experimental results suggest that the synthesis of cardiolipin compounds occurs in periplasm [59,60], we decided to manually modify the biomass function of iJN746 by changing the compartment definition of the *cardiolipin[c]* compounds into periplasm (i.e. *cardiolipin[p]*) (S2 Text). After the manual correction, iJN746 was able to produce a non-zero biomass flux.

Another five models, iNJ661m [61], iNJ661v [61], iSR432 [62], iTZ479 [13], and iT366 [63], initially failed to generate a non-zero biomass flux under FBA when loaded with the *sbml-strict* option. A close examination of these non-viable models revealed additional inconsistencies in SBML syntax, where an attribute named “*boundaryCondition*” was missing. This was fixed by using the *sbml* option of *psamm-import*, which is compatible with the COBRA convention and can automatically create exchange reactions for compounds with a special suffix (i.e. the “*_b*” suffix that indicates boundary compounds).

Additionally, five models had inconsistencies that needed to be manually examined before a viable simulation could be reproduced. The models iRC1080 [64], iRS1563 and iRS1597 [65] contained reactions that were marked irreversible (meaning that a reaction should only progress in the forward direction) but at the same time had the “*LOWER_BOUND*” parameter set to a negative value (meaning that the reaction should be able to progress in the reverse direction). This reflected inconsistencies between the COBRA-compliant format and the SBML standard specifications. To ensure that users can be aware of such inconsistencies, we set up the *sbml* option of *psamm-import* to produce a warning message and by default allowed reactions that had inconsistent reversibility settings to carry fluxes according to the flux bounds specified using the COBRA-compliant format. The model iRsp1095 [66] contained no exchange reaction that would supply nutrients to the simulation environment. This was due to the model defining all exchange reactions in the SBML file as sinks but not sources of boundary compounds. To resolve such inconsistency, we manually set the flux limits of all exchange reactions according to the minimal medium reported in the original publication [66] to permit uptakes of nutrient compounds that lead to the production of non-zero biomass flux (S3 Text). The model iMA871 [67] was not viable because a symbolic compound of the biomass, *M_BIOMASS*, was accumulated as a result of the biomass reaction, which deviated from a steady-state assumption of the FBA simulation. This was resolved by connecting the *M_BIOMASS* compound with a biomass sink flux, which simulates the biomass production of the model (S4 Text).

Finally, two SBML models, iSyn731 [65] and iMA945 [68], were not viable and could not be fixed by simply introducing flexibilities in SBML parsing. The iMA945 model contains a large number of reaction equations that carry duplicated compounds on the opposite sides, which were likely introduced to the model due to the mislabeling of compound compartments. This inconsistency was corrected by importing the model from an Excel file released in the original publication of iMA945. In the Excel file, compound compartments were assigned correctly and there were no duplicated compounds in any reactions. The iSyn731 model was corrected using an updated version of the SBML release downloaded from the original authors’ website (S5 Text). A comparison of the updated version with the original version of iSyn731 revealed a number of changes in reaction directionality and stoichiometric balance, which might be the result of unintentional inconsistencies in the original conversion of the author’s internal model annotations into the SBML format. Hence, this further illustrated the importance of standardized model version tracking and integrated model representations. The YAML files resulting from importing all the above-mentioned SBML models and all the manual corrections were

documented in an open-source Git repository at <https://github.com/zhanglab/psamm-model-collection>.

Inconsistencies in excel files. As an important part of a standard model release, most of the published GEMs were supplemented with Excel files. Compared to the SBML format, the Excel tables usually include more extensive metadata, such as the compound and reaction features, into the model definition. However, so far there is no standardized definition of the Excel format. Hence, the Excel import function was implemented individually for specific models. In the PSAMM input/output module, customized Excel parsers were created based on a supervised process, in which the content of Excel tables were examined and model specific scripts were developed to parse out the complete information that defines the nine Excel models we analyzed (S3 Table). FBA simulations were also performed on the biomass functions to examine the viability of each Excel model. In cases where FBA resulted in a zero biomass flux, we further examined the non-viable models using a combination of manual curations and comparisons to the corresponding SBML releases. This analysis uncovered a number of inconsistencies in the Excel files. For example, in the reaction worksheet for model iMA945 [68], the columns of reaction equations and gene identities were merged for a subset of entries, causing a column shift in these entries. By correcting the alignment of table columns, the model was parsed successfully using the customized Excel parser for iMA945. Another example of inconsistencies involved the misspelling of compound identifiers. The reaction worksheet of model iMR1_799 [69] contained an invalid compound “*aaacoa*” in the reaction equation of acetyl-CoA C-acetyltransferase, which as revealed by comparing to the SBML release of the same model, is a misspelled identifier of the compound Acetoacetyl-CoA (“*aacoa*”). Similar inconsistencies were found in other models and were specifically corrected in the construction of model-specific parsers to map the misspelled compounds to corrected names. These inconsistencies have highlighted the difficulty in preventing unintended edits to annotation tables since the observed column shifts and misspellings are likely a result of the manual insertion of compound or reaction entries in the Excel format. This problem can be resolved with the PSAMM YAML format, which permits the direct incorporation of annotation tables into the model using a simple *include* function (Fig 3 and S1 Text). By submitting YAML models to the Git version control, all edits to the table will be recorded and changes that render the model non-viable can then be pinpointed for correction. Moreover, by maintaining the independence of individual *include* functions, the YAML format effectively incorporates the diverse model metadata that is traditionally included in a scattered set of Excel worksheets into an integrated documentation of GEMs.

Stoichiometric inconsistencies. The stoichiometric consistency of all GEMs was analyzed using the stoichiometric consistency checking approaches implemented in the PSAMM *masscheck* function (Materials and Methods). Surprisingly, only 18 out of the 57 models passed the criteria of stoichiometric consistency check, suggesting that most models included reactions that are not mass balanced (S3 Table). This is alarming especially in the context of constraint-based analysis because unbalanced reactions can potentially generate infinite source of materials. For example, the model iKF1028 [70] contains two mutually exclusive reactions, RR00610 and RR08939, which differ only by the presence or absence of two additional *H+* compounds on one side of the reaction equations (Table 3, reactions marked with *). Both reactions cannot occur simultaneously in the same model, otherwise the combination of the two can result in a physiologically impossible overall function that generates or consumes an infinite amount of *H+* compounds at no cost. Running a compound-based stoichiometric check on iKF1028 (S1 Table, command “*psamm-model masscheck*”), we identified two compounds, *H+* and *H+[e]*, as the sources of stoichiometric inconsistency. However, this information is not sufficient for model correction because the compounds *H+* and *H+[e]* occurred in

Table 3. Stoichiometric inconsistencies in iKF1028 [70]. Reaction: the reaction identifiers in the GEM; Equation: the reaction equations; H left/right: the total number of H atoms at the left/right side of the equations; H residue: the differences between the number of H atoms at the left versus the right side of the equations. Two reactions, RR08939 and IR01815, are shown at the bottom of the table, which correspond to the balanced version of the inconsistent reactions RR00610 and IR04287, respectively. Both pairs (marked with * and **, respectively) were present in iKF1028, rendering the overall model stoichiometrically inconsistent.

	Reaction	Equation	H left	H right	H residue
Unbalanced reactions	IR00136	[c]: 2-Oxoglutarate --> Succinate semialdehyde + CO2	4	5	-1
	IR00159	[c]: H2O + Acetyl phosphate --> Orthophosphate + Acetate	5	4	1
	IR00373	[c]: L-Ornithine --> CO2 + Putrescine	13	14	-1
	IR00765	[c]: NADP+ + H2O + L-Valine --> NH3 + NADPH + 3-Methyl-2-oxobutanoate	38	36	2
	IR01585	[c]: D-Glucose 6-phosphate + UDPglucose --> alpha,alpha'-Trehalose 6-phosphate + UDP	33	32	1
	IR04287**	[c]: Uroporphyrinogen III + (2) S-Adenosyl-L-methionine --> Precorrin 2 + (2) S-Adenosyl-L-homocysteine + H+	82	89	-7
	IR04586	[c]: NADH + NH3 + H+ + Pyruvate --> L-Alanine + H2O + NAD+	34	35	-1
	IR09357	[c]: (S)-3-Hydroxy-3-methylglutaryl-CoA --> Acetyl-CoA + Acetoacetate + (5) H+	39	44	-5
	IR10056	[c]: Precorrin 2 + S-Adenosyl-L-methionine --> Precorrin 3A + S-Adenosyl-L-homocysteine + H+	71	64	7
	IR10481	[c]: Aminoimidazole ribotide + HCO3- + ATP --> 5-Carboxyamino-1-(5-phospho-D-ribosyl)imidazole + ADP + Orthophosphate	25	27	-2
	RR00067	[c]: NAD+ + (2) L-Glutamate <=> L-Glutamine + NADH + 2-Oxoglutarate	42	41	1
	RR00610*	[c]: D-Glucose 1-phosphate + (2) H+ <=> D-Glucose 6-phosphate	13	11	2
	RR00692	[c]: D-Fructose 1,6-bisphosphate <=> Dihydroxyacetone phosphate + (4) H+ + D-Glyceraldehyde 3-phosphate	10	14	-4
	RR10479	[c]: 5-Carboxyamino-1-(5-phospho-D-ribosyl)imidazole <=> 1-(5-Phospho-D-ribosyl)-5-amino-4-imidazolecarboxylate	14	11	3
Balanced reactions	IR01815**	[c]: Uroporphyrinogen III + (6) H+ + (2) S-Adenosyl-L-methionine --> Precorrin 2 + (2) S-Adenosyl-L-homocysteine	88	88	0
	RR08939*	[c]: D-Glucose 1-phosphate <=> D-Glucose 6-phosphate	11	11	0

doi:10.1371/journal.pcbi.1004732.t003

460 reactions in the model and only a small subset of those were unbalanced. Applying a new algorithm in the PSAMM *masscheck* function ([Materials and Methods](#) and [S1 Table](#), command “*psamm-model masscheck --type=reaction*”), a list of 14 unbalanced reactions was identified ([Table 3](#)), which greatly reduced the number of reactions that required manual examinations. By comparing the number of hydrogen elements on both sides of the 14 reaction equations, the model was revised to fulfill the requirements of stoichiometric consistency. The *masscheck* function can be applied for the identification of unbalanced reactions in other models following the same procedure as demonstrated in the analysis of iKF1028. Unlike the *formulacheck* function, which relies on the definition of compound formulas, *masscheck* can be applied even when compound formulas are not available. This feature is especially useful in the annotation of existing SBML models, as many of the SBML releases do not contain annotation of the compound formulas. When using *masscheck*, users could make expert decisions to exclude reactions (using the *--exclude* option) that are expected to be unbalanced. These may include the exchange and biomass reactions, and the reactions for which the biochemistry is undetermined (i.e. biosynthesis of polymer molecules). Overall, the PSAMM *masscheck* function is optimized to pinpoint the sources of inconsistency among hundreds and thousands of reactions, and it provides flexibility for identifying stoichiometric inconsistencies in both compounds and reactions even when the mass and formula annotations are missing.

Flux inconsistencies. The PSAMM *fluxcheck* function was applied to all the GEMs to search for flux inconsistent reactions—the reactions that are blocked and unable to carry a non-

zero flux under any simulation conditions (Materials and Methods). On average, 30% of all internal reactions in the GEMs were blocked, suggesting that these reactions were topologically disconnected from the rest of the model. We looked into the distribution of blocked reactions among functional subsystems by analyzing a subset of 19 GEMs that contained subsystem annotations in the SBML format (S3 Table). First, subsystem annotations in the GEMs were obtained from SBML files using *psamm-import*, which identified 436 distinct subsystem names from the 19 GEMs. Then, the subsystem names were manually curated and classified into ten metabolic pathways (S4 Table). Finally, the fraction of blocked reactions was calculated within each metabolic pathway, and the median of fractions was identified across all analyzed GEMs (Fig 4).

As a result, most of the metabolic pathways demonstrated high level of flux consistency, and the fraction of blocked reactions had a median lower than 0.2 for seven out of the ten pathways, including four pathways in the central metabolism, two pathways for the metabolism of lipid and glycan, and one pathway of transport reactions. Not surprisingly, the four central metabolic pathways, carbon, energy, amino acid, and nucleic acid metabolism, had the lowest fraction of blocked reactions and were the least variable among different GEMs. This not only reflected their essential roles in biomass production, but also indicated the extent of research that contributed to the complete reconstruction of these pathways. One outlier, however, was the energy metabolism pathway in iAI549 [71], which had 23 blocked reactions out of the 41 total reactions. This was largely due to the lack of certain nutrient sources that drive the energy functions. For example, the model contained two nitrite reductase reactions, but no nitrite source was provided either from boundary conditions or from other pathways.

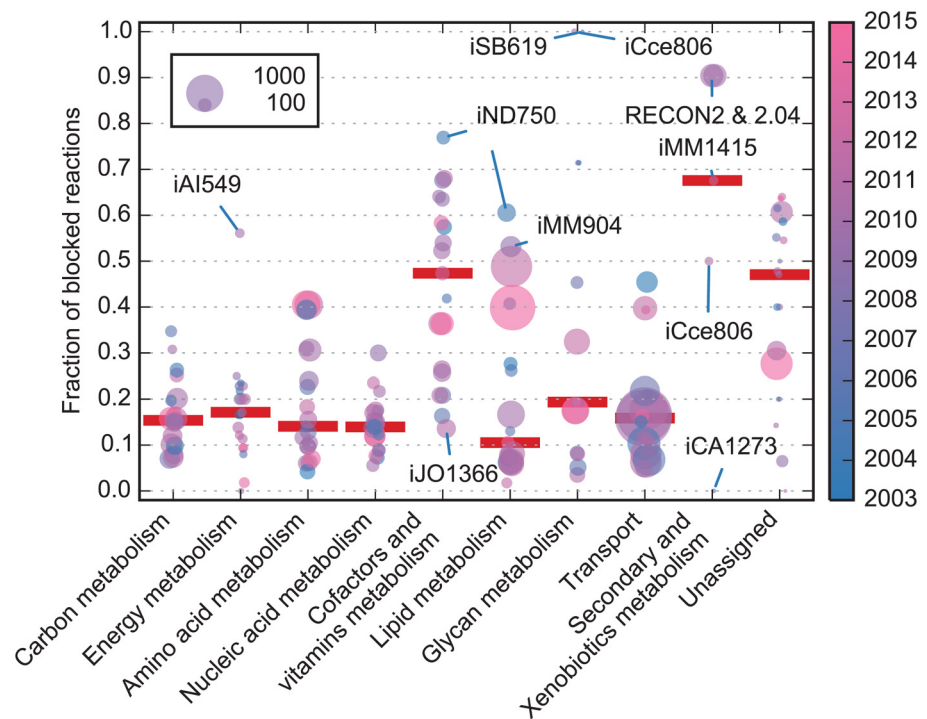


Fig 4. Distribution of blocked reactions in metabolic pathways. The GEMs were represented in each metabolic pathway as a solid circle. The color of the circles corresponds to the year in which a GEM was published (color legend was shown on the right, and the year of publication ranges from 2003 to 2014). The area of the circle is proportional to the total number of reactions in the pathway, and its vertical position indicates the fraction of reactions that are blocked. The median fractions were indicated by a red mark for each pathway, and models discussed in the main text were highlighted.

doi:10.1371/journal.pcbi.1004732.g004

In contrast to the central metabolic pathways, the lipid and glycan metabolism had higher variations among different GEMs despite having low median fractions of reactions that were blocked. For example, the glycan metabolism varies from almost fully consistent (STM_v1.0 [72]) to completely blocked (iSB619 [73] and iCce806 [74]). The STM_v1.0 model contained 143 reactions in glycan metabolism. Out of these only five were blocked reactions, which was attributed to a dead-end compound in the “*Murein LPP Biosynthesis*” subsystem. The iSB619 and iCce806, however, contained two and eleven reactions in glycan metabolism, respectively. All of these reactions were blocked and appeared as dead ends of the metabolic models. Similarly, some models (e.g. iND750 [75] and iMM904 [76]) had a large fraction of blocked reactions in lipid metabolism, indicating that the lipid metabolic pathways were incomplete in these models.

The cofactors and vitamins metabolism pathway had slightly higher fractions of blocked reactions (median close to 0.5), and the level of flux inconsistency varies among different GEMs. For example, the model iND750 [75] had a largely blocked cofactors and vitamins metabolism pathway, with 80 of the 104 reactions (77%) unable to produce non-zero fluxes due to the missing connections between metabolites and the biomass function. In contrast, the model iJO1366 [40] had 228 reactions in the cofactors and vitamins metabolism pathway, and only 26 reactions (11%) were blocked.

The secondary and xenobiotics metabolic pathway, while having variable fractions of blocked reactions, only occurred in five GEMs due to their inessentiality to biomass production. Reactions in this pathway can be used to analyze the relationship between biomass production and the metabolism of secondary metabolites. For example, the iCA1273 [77] model has three reactions associated with penicillin breakdown. These reactions are not connected to the biomass production but they are connected appropriately with exchange reactions. Hence, the model can be used to simulate the effect of penicillin on the growth of the organism. Conversely, the four other models (iMM1415 [55], iCce806 [74], RECON2 [78], and RECON2.04) appear to contain a large fraction (0.67 and 0.5, respectively, for iMM1415 and iCce806 and 0.9 for the two RECON models) of blocked reactions in the secondary and xenobiotics metabolism and would require the addition of transport or exchange reactions to simulate functions in this pathway.

It is also worth noting that the total number of reactions was more consistent in the central metabolism than in the pathways of lipid metabolism, glycan metabolism, and transport reactions (Fig 4). The variations among the latter pathways can be attributed to the modeling decisions or the large variability of these pathways among different organisms. For example, some models had experienced an expansion of the glycan and lipid metabolism pathways in later releases compared to the earlier releases. This expansion is apparent when comparing the 2003 *E. coli* model iJR904 [79] and the 2007 model iAF1260 [80], and when comparing the 2004 *S. cerevisiae* model iND750 [75] with the 2009 model iMM904 [76] (S2 Fig). As the models have evolved through successive updates, a more extensive annotation of glycan and lipid metabolism has been applied, and the lipid and glycan compounds have been included into the biomass function. In contrast to the other pathways, the total number of transport reactions was influenced by the number of cellular compartments in the model. While the GEMs of some eukaryotic organisms contained over 1,000 transport reactions that account for the metabolite exchange among different cellular compartments, GEMs with only two compartments had on average less than 100 transport reactions.

Discussion

The reconstruction and analysis of GEMs have broad applications in understanding genotype-phenotype connections, studying evolutionary processes, and predicting the outcomes of

metabolic engineering [13,81]. Despite the development of tools that support individual steps along the modeling procedure [24,26,29], it is still challenging to associate mathematical simulation with the biological interpretation of GEMs. Often times, model files need to be curated iteratively to set up exchange reactions, assign reaction directionality, define biomass functions, and fill reaction gaps, and the results of mathematical simulations need to be mapped to the annotated data of model definitions. However, due to the absence of integrated data representations and the lack of standardized model checking procedures, the curation of GEMs are prone to introducing inconsistencies, which in turn may lead to errors in modeling and in interpreting the biological meaning of modeling outcomes.

The PSAMM software package was developed to solve the above challenges by integrating the annotation of metabolic pathways with the consistency checking and the constraint-based analyses of mathematical models (Fig 2). It offers a novel YAML format of model representation that integrates the heterogeneous annotation of model components with the formulation of mathematical simulation problems. PSAMM takes advantage of several useful features commonly supported in the Python programming language: (1) it is highly portable and can be easily installed using the pip package management system; (2) it is open source and does not rely on paid software like MATLAB; (3) it has extensive library support that permits both matrix-based operations and dictionary-based feature representations; (4) it is an integrated package that supports calling simulation functions using simple one-line commands (S1 Fig) and permits adjustments to modeling parameters using command options (S1 Table). PSAMM supports the import and export of diverse model formats and is applicable for the curation of draft models generated from model reconstruction pipelines like ModelSEED [26], RAVEN [29], and Pathway Tools [31]. It provides extensive help information (e.g. through the command-line `-h` or `--help` options) and an online tutorial that demonstrates the main functions and workflow of PSAMM (<https://psamm.readthedocs.org/en/latest/tutorial.html>). Additionally, the PSAMM functionality is easily expandable by the user community through an Application Programming Interface (API).

PSAMM integrates heterogeneous metadata in model annotations with the simulation of metabolic networks, so that users can perform model curation and mathematical simulations under the integrated framework of a unified software package. While COBRA and RAVEN are released as toolboxes and are dependent on the MATLAB software, PSAMM is an independent software package that is freely available for all academic users. Calling the simulation functions is largely simplified in PSAMM because the individual steps of model reading, solver selection, model simulation, and simulation results mapping are integrated into a single one-line command. In contrast, setting up a simulation problem in COBRA would require calling multiple functions in the MATLAB interface (S1 Fig). Moreover, PSAMM allows users to document their modifications to the model definition as well as the simulation conditions during model curations. In contrast, changes made through the MATLAB interface will be lost when the computing session is closed. Even if users write out their changed models into an SBML file, it will be difficult for the users to track the exact location of the changes due to the incompatibility of SBML files with line-based version control systems [51,82]. Finally, PSAMM is configured to perform with high efficiency a number of computationally intensive simulations, such as tFBA, FVA, robustness analysis and random minimal networks (Fig 1 and S2 Table). As an example, we recorded the running time of robustness analysis on iJO1366, which is one of the larger models in the collection we analyzed. Using a single computing thread, the median runtime of robustness analysis with 1000 steps on the EX_o2(e) exchange reaction using Cplex solver was 4.07 seconds in PSAMM (including the reading of model files), whereas COBRA takes 100.7 seconds under the same settings for just solving the simulation problem. This is because PSAMM modifies the simulation problem directly through an optimized LP utility

interface (Fig 2), while COBRA requires the simulation problem to be redefined for every step of the robustness analysis.

The PSAMM YAML-based model representation provides additional support to model curation by offering a human-readable interface for the organization of heterogeneous data (Fig 3 and S1 Text). The YAML format incorporates information about both model annotations (e.g. in the *reactions*, *compounds*, and *model* definitions) and model simulations (e.g. in the *biomass*, *media*, and *limits* definitions), and it supports a flexible interface for storing annotation information either within a centralized model file or as independent feature files. The YAML format streamlines model representation by minimizing the use of markup tags. The use of line breaks and whitespaces as markups of data structure not only enables the parsing and association of user-defined, model-specific model annotations, but also enhances the compatibility of YAML with line-based version control systems. This compatibility with version tracking is especially useful in collaborative projects that involve multiple curators working on the same model since modifications made by different curators can be documented in parallel and later reconciled and merged into a new model release that is more comprehensive than what could be achieved by a single curator. Additionally, YAML supports the modular representation of diverse model components and the free combination of model definitions with diverse simulation conditions (Fig 5). Hence, instead of being restricted to a single simulation condition, YAML provides additional flexibilities for the documentation of multiple modeling conditions in a single model release.

Applying PSAMM to existing GEMs in public literature, a number of inconsistencies were identified. These included inconsistencies in the formatting of model files (e.g. SBML or Excel files), the definition of model stoichiometry, and the presence of blocked reactions. In general, the representation of SBML and Excel files varies among different GEMs both in the formatting of model annotations and in the completeness of information provided. For example, only a small fraction of the collected SBML models included complete metadata regarding protein-coding genes, functional annotations, Enzyme Commission (EC) numbers, and subsystem classifications. This was largely due to the lack of standardized definition of such information in SBML specifications and to some extent has prevented the SBML format from integrating heterogeneous annotation data. The Excel files, although frequently used by model curators for the annotation of pathway information, also lacks standardization and data integration. These limitations have caused a number of common inconsistencies in the representation of both SBML and Excel models. Using the annotation framework of PSAMM and the integrated YAML representation, we have corrected these inconsistencies and documented the changes in an online Git repository at <https://github.com/zhanglab/psamm-model-collection> (S2–S5 Text).

To ensure that the GEMs correctly represented the function and connectivity of metabolic networks, the PSAMM stoichiometric and flux consistency checking were applied to search for unbalanced or disconnected reactions. Surprisingly, only 30% of the analyzed models were stoichiometrically balanced, and the unbalanced reactions can be attributed to the inconsistent representation of compounds in different reaction equations. The analysis of flux consistency revealed that while the central metabolic pathways contain only a small fraction of blocked reactions across all GEMs, the pathways of lipid metabolism, glycan metabolism, and cofactors and vitamins metabolism were largely variable among different GEMs and were frequently disconnected from biomass production. Interestingly, the pathway distribution of metabolic reactions varied among different GEMs (S2 Fig). In the successive models of certain organisms (e.g. in *E. coli*), it appeared that the initial reconstructions have focused on central metabolic pathways, while the later iterations have contributed to the inclusion and completion of peripheral pathways like the transport reactions and the glycan and lipid metabolism. While it is not

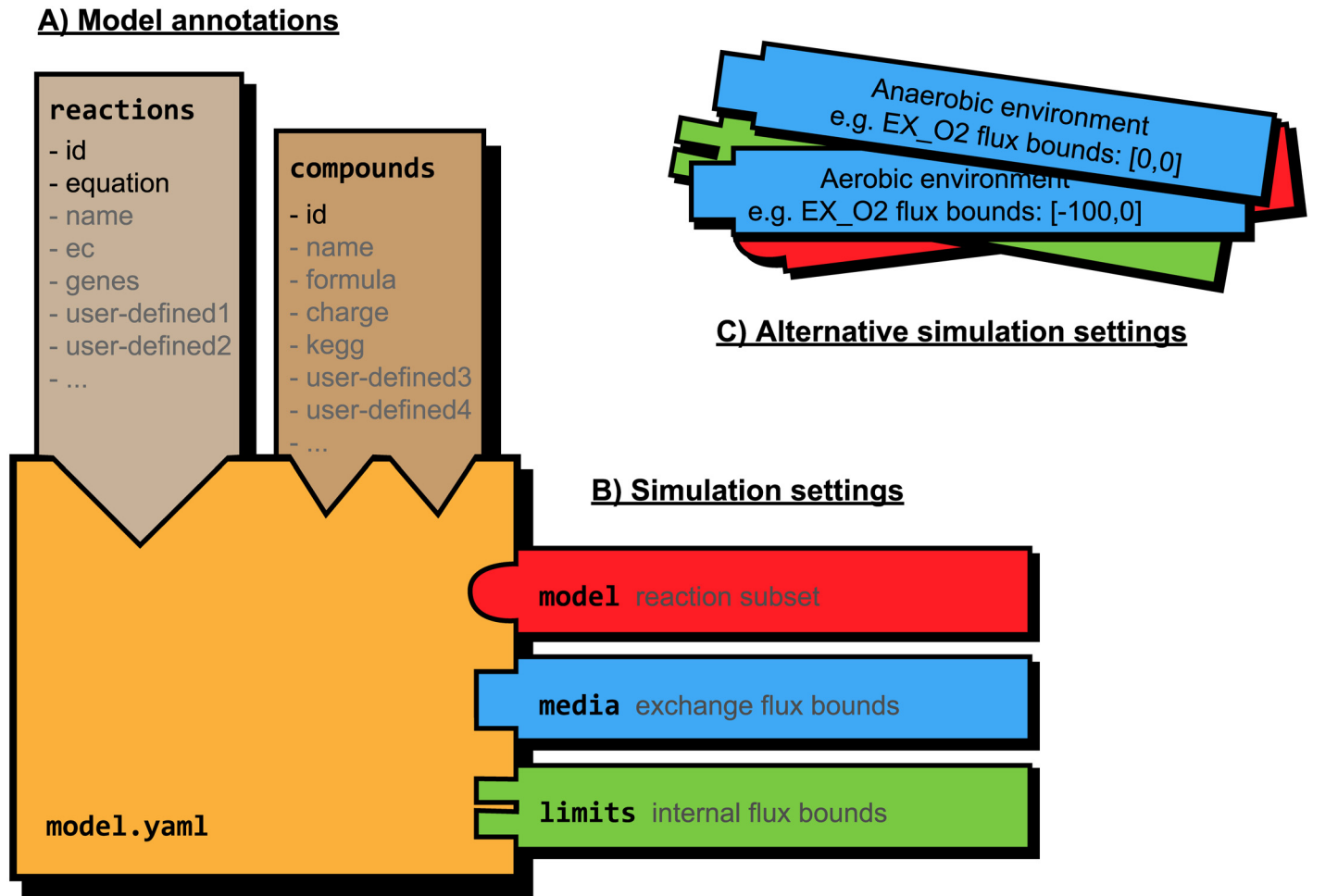


Fig 5. A diagram illustrating the modular representation of model components in the YAML format. The data structure is divided into the static components of model annotation (A) and the dynamic components of simulation settings (B). The reaction and compound annotation databases are associated with a number of required (highlighted in black, e.g. “- id” and “- equation” for reactions) and optional (gray) data entries, and user-defined, model-specific data entries are permitted in the annotation databases. The simulation settings can be represented with various combinations of the model, limits, and media files. Alternative conditions may be defined using a number of alternative modules that can be switched with one another.

doi:10.1371/journal.pcbi.1004732.g005

uncommon to have blocked reactions in GEMs due to the limited knowledge in current literature about certain metabolic pathways, the analysis of flux inconsistency should be a standard step especially in interpreting the biological significance of gene or reaction deletion simulations. Since the blocked reactions do not contribute to the production of biomass, they will always be predicted as non-essential reactions. However, such prediction is not biologically meaningful because it reflects an artifact caused by the disconnection of metabolic pathways.

Materials and Methods

Comparisons of Program Running Time

A list of functions (marked in [S1 Table](#)) was examined in the PSAMM package as well as the COBRA and RAVEN toolboxes to compare the efficiency of the different tools. The simulations were carried out on the model iJO1366 [40] with the Cplex linear programming solver for PSAMM (version 0.17) and COBRA (version 12.6, IBM academic release for PSAMM and TOMLAB release for COBRA), and with the MOSEK solver for RAVEN (version 7.1.0.36).

The specific simulation parameters were set according to the conditions specified for each function in the [S2 Table](#). Outputs from different tools were manually examined to ensure that the same results were generated from the same simulation settings. The running time was recorded in a CentOS operating system with a single processor core allocated to each program (the solver was not restricted in this way and was able to use up to 20 cores). Each simulation was carried out at least seven times, from which the median was plotted in [Fig 1](#) with the 75th and the 25th percentile values as the upper and the lower limits of the error bar.

Dependencies

PSAMM has the following dependencies, some of which are required and some of which are optional but limit the functionality of PSAMM in their absence. The dependencies PyYAML, xlrd, xlswriter, and NumPy can be automatically installed through the Python package manager pip. The linear programming solvers need to be installed by the user following instructions from corresponding releases.

- PyYAML (required, for parsing the PSAMM YAML format)
- IBM ILOG Cplex Optimizer (required, for solving linear programming problems)
- GUROBI Optimizer (optional, an alternative to the Cplex linear programming solver)
- QSOPT_ex rational solver [83] (optional, for obtaining exact solutions to the LP problem)
- xlrd (optional; for reading models from Excel sheet files)
- xlswriter (optional; for exporting YAML models as Excel sheet files)
- NumPy (optional; for converting the sparse matrix representation to the NumPy matrix)

Collection of Metabolic Models

A list of 56 GEMs was collected from current literature and a public model collection [84] as SBML files. Additionally, the RECON2.04 model was downloaded in the format of a MATLAB data file from a public release at <https://vmh.uni.lu/#downloadview>. Compound and reaction features of nine models were downloaded in the Microsoft Excel format ([S3 Table](#)). These models were used to test the application of PSAMM for model management and model curation. For constraint-based simulations, a biomass reaction was identified for each model by examining the objective coefficient defined under the SBML “kineticLaw” section. The reaction with a non-zero objective coefficient was treated as the biomass reaction. If the objective function was not specified in SBML file, the reaction identifiers were searched and the ones containing the string “biomass” were marked as biomass reactions. Information was also obtained from original publications of the collected models to identify biomass reactions and to select the main biomass reaction when more than one was present. The biomass reactions of each model were listed in [S3 Table](#).

Stoichiometric Consistency Checking

The stoichiometric consistency check was implemented in the PSAMM *masscheck* function, which supports both compound-based and reaction-based checking of the stoichiometric consistency. By default, the *masscheck* function excludes the exchange and biomass reactions from consideration because by design, stoichiometric balance is not required in these reactions. The

compound-based stoichiometric consistency checking was implemented based on Thiele et al. [49] and was formulated as:

Maximize $\sum_i z_i$

Subject to

- $S^T \mathbf{m} = 0$
- For all m_i in \mathbf{m} : $m_i \geq z_i$
- For all z_i in \mathbf{z} : $z_i \in [0;1]$

This problem looks for compounds that failed to obtain a positive mass under optimizing conditions. By maximizing the values in \mathbf{z} up to 1 for as many compounds as possible, it forced the mass values (\mathbf{m}) to be at least 1 for the same compounds. The sum of the values in \mathbf{z} can be used to approximate the number of compounds that were properly balanced.

Although the compound-based checking was useful for identifying compounds that may cause stoichiometric inconsistency, it provides no insights into the stoichiometric balance of individual reactions. Therefore, PSAMM implements a new algorithm to directly search for unbalanced reactions. This approach is performed using an LP problem that was modified from the above stoichiometric consistency check. A mass residual variable (\mathbf{r}) is introduced into every reaction and bounded by the residual bound variable (\mathbf{z}) such that $r_j \in [-z_j; z_j]$. The residual bound variables are then minimized, and the reactions that carry a non-zero residual value r_j (i.e. bounded by a positive z_j) are reported as a candidate of inconsistency. The mathematical formulation of this approach is described in the following LP problem:

Maximize $\sum_j z_j$

Subject to

- $S^T \mathbf{m} = \mathbf{r}$
- For all m_i in \mathbf{m} : $m_i \geq 1$
- For all z_j in \mathbf{z} : $z_j \geq 0$
- For all r_i in \mathbf{r} : $r_i \in [-z_j; z_j]$

This approach can be applied iteratively to assist with the correction of stoichiometric inconsistencies in metabolic models. The residual variables indicate a mass that is missing from identified reactions, and the sign of the residual can be used to determine whether the left or right side of an equation has a missing mass. Additionally, PSAMM provides a “--checked” option in the *masscheck* function, which allows fixing certain residual values at zero when the corresponding reactions have been confirmed to be balanced, making the program able to converge on the remaining set of unbalanced reactions. When the stoichiometric consistency check was performed on the collection of models, the biomass and exchange reactions were excluded from consideration and the requirement of a positive mass was removed from compounds that were used to model photons or electrons.

Flux Consistency Checking

The flux consistency check was implemented in the PSAMM *fluxcheck* function. A *flux inconsistent reaction* is a reaction that cannot take a non-zero flux under any simulation conditions. In other words, given the stoichiometric matrix S , the vector of fluxes \mathbf{v} , and the reaction j , if no solution to $S\mathbf{v} = \mathbf{0}$ exists, where $v_j \neq 0$ the reaction is considered to be flux inconsistent. Conversely, if a solution exists the reaction is considered to be flux consistent. A consistent model is a model that only contains consistent reactions [85]. The flux consistency check in PSAMM

was implemented based on two independent approaches. The first approach was based on the definitions in [86]. This was achieved by applying a modified version of the flux variability analysis (FVA) on each reaction while using default constraints on other reactions. For reversible reactions, the flux is first maximized and then minimized, and for irreversible reactions, only the maximization is necessary. A significant speedup was possible in PSAMM (S2 Table) by avoiding regenerating new problem definitions throughout the procedure. In previous implementations (e.g. COBRA) the LP problem was regenerated for optimizing each reaction. However, in PSAMM the LP problem definition was instead reused. This was feasible since the LP problems in this approach are all equivalent except for using a different objective function for each reaction. Reactions that carried non-zero fluxes were considered to be flux consistent, and vice versa. Alternatively, PSAMM provides another way of checking reaction flux consistency based on the fast consistency check (FASTCC) algorithm, which was designed to provide a more efficient solution in evaluating reaction flux consistency [87]. When analyzing the models in the collection for flux inconsistencies, the constraints on the flux limits of the exchange reactions were removed prior to the analysis to eliminate the influences of external settings and to provide a lower estimate for the fraction of inconsistent reactions (Fig 4).

Supporting Information

S1 Fig. Comparison of the user interfaces in PSAMM versus COBRA. While PSAMM operates as a command-line tool, COBRA is a toolbox dependent on the MATLAB environment. The red font indicates commands typed in by users, and the black font indicates outputs from running a given command.

(TIF)

S2 Fig. Distribution of model reactions by metabolic pathways in 19 models. The RECON2 and RECON2.04 were plotted in a different panel, with iMM1415 in both panels as a reference to show the differences in scaling y-axis.

(TIF)

S1 Table. Comparison of the common constraint-based modeling functions in PSAMM, RAVEN, and COBRA. The marked rows in the column "Time" indicate commands for which the running time was recorded.

(PDF)

S2 Table. Comparisons of the time needed to use various functions in PSAMM, COBRA, and RAVEN. The function is listed in the first column along with information on the parameters used in the simulation. The specific commands, running times for seven independent experiments, along with the 25 and 75 percentile times and the median time taken to run the commands are listed in the remaining columns.

(PDF)

S3 Table. List of metabolic models analyzed. The table shows the model identifiers (column 1); the scientific names of the modeled organisms (column 2); the domain of life that each organism belongs to (column 3); the journal, authors and year of the model publications (columns 4–6); whether the Excel format of a model was analyzed (column 7); the SBML level used to format each model's SBML release (column 8); whether the subsystem information was available for model comparison (column 9); the main biomass reaction used to simulate model growth (column 10); additional biomass reactions included in the model (column 11); whether the model is stoichiometrically consistent (column 12); and the percentage of internal reactions

in the model that were blocked (column 13).
(PDF)

S4 Table. Mapping of model subsystems to metabolic pathways. The model-specific subsystem names in the first column are mapped to the metabolic pathways in the second column.
(PDF)

S1 Text. Description of the PSAMM YAML model format.
(PDF)

S2 Text. Comparison of the original version and the fixed version of iJN746 [58] using the *git diff* function in the Git version control system. The original version cannot provide a non-zero flux in FBA simulations because the periplasmic cardiolipin compounds (M_clpn120_p, M_clpn160_p, M_clpn161_p, M_clpn180_p, M_clpn181_p) were mislabeled as cytosolic compounds in the SBML model. The updated model correctly incorporates the mislabeled compounds and is able to produce a non-zero flux on biomass. The lines are color coded to highlight manual corrections made to the model. Highlighted in red are the lines removed, and highlighted in green are the lines added from the original version to the new version.
(PDF)

S3 Text. Comparison of the original version and the fixed version of the model iRsp1095 [66] model using the *git diff* function in the Git version control system. The original version cannot provide a non-zero flux in FBA simulations, while the fixed version can. This was determined to be due to all of the exchange reactions being set to have a lower bound of zero thus preventing uptake of any nutrients. The exchange reactions were altered to reflect the boundary conditions stated in the supplemental materials of the original paper [66]. The lines are color coded to highlight the changes between the two versions. Highlighted in red are the lines removed, and highlighted in green are the lines added from the original version to the new version.
(PDF)

S4 Text. Comparison of the original version and the fixed version of the model iMA871 [67] model using the *git diff* function in the Git version control system. The original version cannot provide a non-zero flux in FBA simulations, while the fixed version can. The problem was determined to be that there was no sink for the artificial biomass compound, *M_BIO-MASS*, meaning it could accumulate but could not be counted towards biomass production. A sink was added for the compound allowing the model to generate a non-zero biomass flux. The lines are color coded to highlight the changes between the two versions. Highlighted in red are the lines removed, and highlighted in green are the lines added from the original version to the new version.
(PDF)

S5 Text. Comparison of the original version and the updated version (downloaded on September 09, 2015 from <http://www.maranasgroup.com/models.htm>) of the model iSyn731 [65] using the *git diff* function in the Git version control system. The original version cannot provide a non-zero flux in FBA simulations. The new version of the model contained several changes including reaction directionality changes, stoichiometric changes, and the addition and deletion of reactions. The lines are color coded to highlight the changes between the two versions. Highlighted in red are the lines removed, and highlighted in green are the lines added from the original version to the new version.
(PDF)

Acknowledgments

We thank Kyle Sexton and Kelsey Wright for their contributions to programming and software testing.

Author Contributions

Conceived and designed the experiments: YZ JLS. Performed the experiments: JLS KDT YZ. Analyzed the data: YZ JLS KDT. Contributed reagents/materials/analysis tools: JLS KDT YZ. Wrote the paper: YZ JLS KDT. Designed the software: YZ JLS. Implemented software programs: JLS KDT.

References

1. Oberhardt MA, Palsson BØ, Papin JA (2009) Applications of genome-scale metabolic reconstructions. *Mol Syst Biol* 5: 320. doi: [10.1038/msb.2009.77](https://doi.org/10.1038/msb.2009.77) PMID: [19888215](https://pubmed.ncbi.nlm.nih.gov/19888215/)
2. Feist AM, Palsson BØ (2008) The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*. *Nat Biotechnol* 26: 659–667. doi: [10.1038/nbt1401](https://doi.org/10.1038/nbt1401) PMID: [18536691](https://pubmed.ncbi.nlm.nih.gov/18536691/)
3. Sohn SB, Kim TY, Lee JH, Lee SY (2012) Genome-scale metabolic model of the fission yeast *Schizosaccharomyces pombe* and the reconciliation of in silico/in vivo mutant growth. Available: doi: [10.1186/1752-0509-6-49](https://doi.org/10.1186/1752-0509-6-49)
4. Gonnerman MC, Benedict MN, Feist AM, Metcalf WW, Price ND (2013) Genomically and biochemically accurate metabolic reconstruction of *Methanosarcina barkeri* Fusaro, IMG746. *Biotechnol J* 8: 1070–1079. doi: [10.1002/biot.201200266](https://doi.org/10.1002/biot.201200266) PMID: [23420771](https://pubmed.ncbi.nlm.nih.gov/23420771/)
5. Pitkänen E, Jouhten P, Hou J, Syed MF, Blomberg P, et al. (2014) Comparative Genome-Scale Reconstruction of Gapless Metabolic Networks for Present and Ancestral Species. *PLoS Comput Biol* 10.
6. Nogales J, Gudmundsson S, Knight EM, Palsson BO, Thiele I (2012) Detailing the optimality of photosynthesis in cyanobacteria through systems biology analysis. *Proc Natl Acad Sci U S A* 109: 2678–2683. Available: doi: [10.1073/pnas.1117907109](https://doi.org/10.1073/pnas.1117907109) PMID: [22308420](https://pubmed.ncbi.nlm.nih.gov/22308420/)
7. Feist AM, Nagarajan H, Rotaru AE, Tremblay PL, Zhang T, et al. (2014) Constraint-Based Modeling of Carbon Fixation and the Energetics of Electron Transfer in *Geobacter metallireducens*. *PLoS Comput Biol* 10: e1003575. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3998878&tool=pmcentrez&rendertype=abstract>. doi: [10.1371/journal.pcbi.1003575](https://doi.org/10.1371/journal.pcbi.1003575) PMID: [24762737](https://pubmed.ncbi.nlm.nih.gov/24762737/)
8. Nishio Y, Ogishima S, Ichikawa M (2013) Analysis of L-glutamic acid fermentation by using a dynamic metabolic simulation model of *Escherichia coli*. Available: doi: [10.1186/1752-0509-7-92](https://doi.org/10.1186/1752-0509-7-92)
9. Yim H, Haselbeck R, Niu W, C P-B (2011) Metabolic engineering of *Escherichia coli* for direct production of 1, 4-butanediol. Available: doi: [10.1038/nchembio.580](https://doi.org/10.1038/nchembio.580)
10. Segrè D, Vitkup D, Church GM (2002) Analysis of optimality in natural and perturbed metabolic networks. *Proc Natl Acad Sci U S A* 99: 15112–15117. PMID: [12415116](https://pubmed.ncbi.nlm.nih.gov/12415116/)
11. Joyce AR, Palsson BØ (2008) Predicting gene essentiality using genome-scale in silico models. *Methods Mol Biol* 416: 433–457. doi: [10.1007/978-1-59745-321-9_30](https://doi.org/10.1007/978-1-59745-321-9_30) PMID: [18392986](https://pubmed.ncbi.nlm.nih.gov/18392986/)
12. Chang R, Andrews K, Kim D, Li Z, Godzik A, et al. (2013) Structural systems biology evaluation of metabolic thermotolerance in *Escherichia coli*. *Science* (80-) 340: 1220–1223. Available: doi: [10.1126/science.1234012](https://doi.org/10.1126/science.1234012)
13. Zhang Y, Thiele I, Weekes D, Li Z, Jaroszewski L, et al. (2009) Three-dimensional structural view of the central metabolic network of *Thermotoga maritima*. *Science* 325: 1544–1549. doi: [10.1126/science.1174671](https://doi.org/10.1126/science.1174671) PMID: [19762644](https://pubmed.ncbi.nlm.nih.gov/19762644/)
14. Pinney JW, Papp B, Hyland C, Wambua L, Westhead DR, et al. (2007) Metabolic reconstruction and analysis for parasite genomes. *Trends Parasitol* 23: 548–554. PMID: [17950669](https://pubmed.ncbi.nlm.nih.gov/17950669/)
15. Lee SY, Kim HU, Park JH, Park JM, Kim TY (2009) Metabolic engineering of microorganisms: general strategies and drug production. *Drug Discov Today* 14: 78–88. doi: [10.1016/j.drudis.2008.08.004](https://doi.org/10.1016/j.drudis.2008.08.004) PMID: [18775509](https://pubmed.ncbi.nlm.nih.gov/18775509/)
16. Shlomi T, Cabili MN, Ruppin E (2009) Predicting metabolic biomarkers of human inborn errors of metabolism. *Mol Syst Biol* 5: 263. doi: [10.1038/msb.2009.22](https://doi.org/10.1038/msb.2009.22) PMID: [19401675](https://pubmed.ncbi.nlm.nih.gov/19401675/)
17. Lee KH, Park JH, Kim TY, Kim HU, Lee SY (2007) Systems metabolic engineering of *Escherichia coli* for L-threonine production. *Mol Syst Biol* 3: 149. PMID: [18059444](https://pubmed.ncbi.nlm.nih.gov/18059444/)

18. Kim TY, Kim HU, Lee SY (2010) Metabolite-centric approaches for the discovery of antibacterials using genome-scale metabolic networks. *Metab Eng* 12: 105–111. doi: [10.1016/j.ymben.2009.05.004](https://doi.org/10.1016/j.ymben.2009.05.004) PMID: [19481614](https://pubmed.ncbi.nlm.nih.gov/19481614/)
19. Feist AM, Zielinski DC, Orth JD, Schellenberger J, Herrgard MJ, et al. (2010) Model-driven evaluation of the production potential for growth-coupled products of *Escherichia coli*. *Metab Eng* 12: 173–186. doi: [10.1016/j.ymben.2009.10.003](https://doi.org/10.1016/j.ymben.2009.10.003) PMID: [19840862](https://pubmed.ncbi.nlm.nih.gov/19840862/)
20. Hyduke DR, Lewis NE, Palsson BØ (2013) Analysis of omics data with genome-scale models of metabolism. *Mol Biosyst* 9: 167–174. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3594511&tool=pmcentrez&rendertype=abstract>. doi: [10.1039/c2mb25453k](https://doi.org/10.1039/c2mb25453k) PMID: [23247105](https://pubmed.ncbi.nlm.nih.gov/23247105/)
21. Barua D, Kim J, Reed JL (2010) An automated phenotype-driven approach (GeneForce) for refining metabolic and regulatory models. *PLoS Comput Biol* 6.
22. Faria JP, Overbeek R, Xia F, Rocha M, Rocha I, et al. (2013) Genome-scale bacterial transcriptional regulatory networks: reconstruction and integrated analysis with metabolic models. *Brief Bioinform* 15: 592–611. Available: <http://bib.oxfordjournals.org/content/early/2013/02/18/bib.bbs071.short>.
23. Young JD (2014) INCA: A computational platform for isotopically non-stationary metabolic flux analysis. *Bioinformatics* 30: 1333–1335. Available: <http://www.ncbi.nlm.nih.gov/pubmed/24413674>. doi: [10.1093/bioinformatics/btu015](https://doi.org/10.1093/bioinformatics/btu015) PMID: [24413674](https://pubmed.ncbi.nlm.nih.gov/24413674/)
24. Hyduke D, Schellenberger J, Que R, Fleming R, Thiele I, et al. (2011) COBRA Toolbox 2.0. *Protoc Exch*.
25. Ebrahim A, Lerman JA, Palsson BO, Hyduke DR (2013) COBRApy: COstraints-Based Reconstruction and Analysis for Python. *BMC Syst Biol* 7: 74. Available: <http://www.biomedcentral.com/1752-0509/7/74>. doi: [10.1186/1752-0509-7-74](https://doi.org/10.1186/1752-0509-7-74) PMID: [23927696](https://pubmed.ncbi.nlm.nih.gov/23927696/)
26. Henry CS, DeJongh M, Best A a, Frybarger PM, Linsay B, et al. (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat Biotechnol* 28: 977–982. Available: <http://www.ncbi.nlm.nih.gov/pubmed/20802497>. Accessed 29 October 2012. doi: [10.1038/nbt.1672](https://doi.org/10.1038/nbt.1672) PMID: [20802497](https://pubmed.ncbi.nlm.nih.gov/20802497/)
27. Devoid S, Overbeek R, DeJongh M, Vonstein V, Best AA, et al. (2013) Automated genome annotation and metabolic model reconstruction in the SEED and model SEED. *Methods Mol Biol* 985: 17–45. doi: [10.1007/978-1-62703-299-5_2](https://doi.org/10.1007/978-1-62703-299-5_2) PMID: [23417797](https://pubmed.ncbi.nlm.nih.gov/23417797/)
28. Overbeek R, Olson R, Pusch GD, Olsen GJ, Davis JJ, et al. (2014) The SEED and the Rapid Annotation of microbial genomes using Subsystems Technology (RAST). *Nucleic Acids Res* 42.
29. Agren R, Liu L, Shoaie S, Vongsangnak W, Nookaew I, et al. (2013) The RAVEN Toolbox and Its Use for Generating a Genome-scale Metabolic Model for *Penicillium chrysogenum*. *PLoS Comput Biol* 9.
30. Tanabe M, Kanehisa M (2012) Using the KEGG database resource. *Curr Protoc Bioinforma*.
31. Karp PD, Paley SM, Krummenacker M, Latendresse M, Dale JM, et al. (2010) Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology. *Brief Bioinform* 11: 40–79. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2810111&tool=pmcentrez&rendertype=abstract>. doi: [10.1093/bib/bbp043](https://doi.org/10.1093/bib/bbp043) PMID: [19955237](https://pubmed.ncbi.nlm.nih.gov/19955237/)
32. Caspi R, Altman T, Billington R, Dreher K, Foerster H, et al. (2014) The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* 42.
33. Thiele I, Palsson BØ (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat Protoc* 5: 93–121. doi: [10.1038/nprot.2009.203](https://doi.org/10.1038/nprot.2009.203) PMID: [20057383](https://pubmed.ncbi.nlm.nih.gov/20057383/)
34. Ravikrishnan A, Raman K (2015) Critical assessment of genome-scale metabolic networks: the need for a unified standard. *Brief Bioinform*. Available: doi: [10.1093/bib/bbv003](https://doi.org/10.1093/bib/bbv003)
35. Chindelevitch L, Trigg J, Regev A, Berger B (2014) An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. *Nat Commun* 5. Available: doi: [10.1038/ncomms5893](https://doi.org/10.1038/ncomms5893)
36. Chindelevitch L, Trigg J, Regev A, Berger B (2015) MONGOOSE Homepage. Available: <https://groups.csail.mit.edu/cb/mongoose/>.
37. Ebrahim A (2015) Solving models with rational and floating point. Available: http://nbviewer.ipython.org/github/opencvobra/m_model_collection/blob/master/exact_solving_models.ipynb.
38. Bornstein BJ, Keating SM, Jouraku A, Hucka M (2008) LibSBML: An API library for SBML. *Bioinformatics* 24: 880–881. doi: [10.1093/bioinformatics/btn051](https://doi.org/10.1093/bioinformatics/btn051) PMID: [18252737](https://pubmed.ncbi.nlm.nih.gov/18252737/)
39. Git (n.d.). Available: <https://git-scm.com/>.
40. Orth JD, Conrad TM, Na J, Lerman JA, Nam H, et al. (2011) A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism—2011. *Mol Syst Biol* 7.
41. The Systems Biology Markup Language (2010). Available: <http://sbml.org/Documents/Specifications>.

42. Schellenberger J, Park JO, Conrad TM, Palsson BØ (2010) BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions. *BMC Bioinformatics* 11: 213. doi: [10.1186/1471-2105-11-213](https://doi.org/10.1186/1471-2105-11-213) PMID: [20426874](https://pubmed.ncbi.nlm.nih.gov/20426874/)
43. Schellenberger J, Lewis NE, Palsson B (2011) Elimination of thermodynamically infeasible loops in steady-state metabolic models. *Biophys J* 100: 544–553. doi: [10.1016/j.bpj.2010.12.3707](https://doi.org/10.1016/j.bpj.2010.12.3707) PMID: [21281568](https://pubmed.ncbi.nlm.nih.gov/21281568/)
44. Müller AC, Bockmayr A (2013) Fast thermodynamically constrained flux variability analysis. *Bioinformatics* 29: 903–909. doi: [10.1093/bioinformatics/btt059](https://doi.org/10.1093/bioinformatics/btt059) PMID: [23390138](https://pubmed.ncbi.nlm.nih.gov/23390138/)
45. Gudmundsson S, Thiele I (2010) Computationally efficient flux variability analysis. *BMC Bioinformatics* 11: 489. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2963619&tool=pmcentrez&rendertype=abstract>. Accessed 24 March 2011. doi: [10.1186/1471-2105-11-489](https://doi.org/10.1186/1471-2105-11-489) PMID: [20920235](https://pubmed.ncbi.nlm.nih.gov/20920235/)
46. Edwards JS, Palsson BO (2000) Robustness analysis of the Escherichia coli metabolic network. *Bio-technol Prog* 16: 927–939. PMID: [11101318](https://pubmed.ncbi.nlm.nih.gov/11101318/)
47. Pál C, Papp B, Lercher MJ, Csermely P, Oliver SG, et al. (2006) Chance and necessity in the evolution of minimal metabolic networks. *Nature* 440: 667–670. PMID: [16572170](https://pubmed.ncbi.nlm.nih.gov/16572170/)
48. Kumar VS, Dasika MS, Maranas CD, Satish Kumar V, Dasika MS, et al. (2007) Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics* 8: 212. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1933441&tool=pmcentrez&rendertype=abstract>. PMID: [17584497](https://pubmed.ncbi.nlm.nih.gov/17584497/)
49. Thiele I, Vlassis N, Fleming RMT (2014) FASTGAPFILL : Efficient gap filling in metabolic networks. *Bioinformatics*: 15–17. Available: <http://www.ncbi.nlm.nih.gov/pubmed/24812336>.
50. Ben-Kiki O, Evans C, Net I döt (2009) YAML Ain't Markup Language.
51. Rönnau S, Philipp G, Borghoff UM (2009) Efficient change control of XML documents. *Proceedings of the 9th ACM symposium on Document engineering—DocEng '09*. p. 3. Available: <http://dl.acm.org/prox.lib.ncsu.edu/citation.cfm?id=1600193.1600197>.
52. Hirschberg DS (1977) Algorithms for the Longest Common Subsequence Problem. *J ACM* 24: 664–675.
53. Dräger A, Rodriguez N, Dumousseau M, Dörr A, Wrzodek C, et al. (2011) JSBML: A flexible java library for working with SBML. *Bioinformatics* 27: 2167–2168. doi: [10.1093/bioinformatics/btr361](https://doi.org/10.1093/bioinformatics/btr361) PMID: [21697129](https://pubmed.ncbi.nlm.nih.gov/21697129/)
54. Kim HU, Kim TY, Lee SY (2010) Genome-scale metabolic network analysis and drug targeting of multi-drug resistant pathogen *Acinetobacter baumannii* AYE. *Mol Biosyst* 6: 339–348. doi: [10.1039/b916446d](https://doi.org/10.1039/b916446d) PMID: [20094653](https://pubmed.ncbi.nlm.nih.gov/20094653/)
55. Sigurdsson MI, Jamshidi N, Steingrimsson E, Thiele I, Palsson BØ (2010) A detailed genome-wide reconstruction of mouse metabolism based on human Recon 1. *BMC Syst Biol* 4: 140. doi: [10.1186/1752-0509-4-140](https://doi.org/10.1186/1752-0509-4-140) PMID: [20959003](https://pubmed.ncbi.nlm.nih.gov/20959003/)
56. Puchałka J, Oberhardt MA, Godinho M, Bielecka A, Regenhardt D, et al. (2008) Genome-scale reconstruction and analysis of the *Pseudomonas putida* KT2440 metabolic network facilitates applications in biotechnology. *PLoS Comput Biol* 4.
57. Duarte NC, Becker S a, Jamshidi N, Thiele I, Mo ML, et al. (2007) Global reconstruction of the human metabolic network based on genomic and bibliomic data. *Proc Natl Acad Sci U S A* 104: 1777–1782. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1794290&tool=pmcentrez&rendertype=abstract>. PMID: [17267599](https://pubmed.ncbi.nlm.nih.gov/17267599/)
58. Nogales J, Palsson BØ, Thiele I (2008) A genome-scale metabolic reconstruction of *Pseudomonas putida* KT2440: iJN746 as a cell factory. *BMC Syst Biol* 2: 79. doi: [10.1186/1752-0509-2-79](https://doi.org/10.1186/1752-0509-2-79) PMID: [18793442](https://pubmed.ncbi.nlm.nih.gov/18793442/)
59. Shibuya I, Yamagoe S, Miyazaki C, Matsuzaki H, Ohta A (1985) Biosynthesis of novel acidic phospholipid analogs in *Escherichia coli*. *J Bacteriol* 161: 473–477. PMID: [3918012](https://pubmed.ncbi.nlm.nih.gov/3918012/)
60. Quigley BR, Tropp BE (2009) *E. coli* cardiolipin synthase: Function of N-terminal conserved residues. *Biochim Biophys Acta—Biomembr* 1788: 2107–2113.
61. Fang X, Wallqvist A, Reifman J (2010) Development and analysis of an in vivo-compatible metabolic network of *Mycobacterium tuberculosis*. *BMC Syst Biol* 4: 160. doi: [10.1186/1752-0509-4-160](https://doi.org/10.1186/1752-0509-4-160) PMID: [21092312](https://pubmed.ncbi.nlm.nih.gov/21092312/)
62. Roberts SB, Gowen CM, Brooks JP, Fong SS (2010) Genome-scale metabolic analysis of *Clostridium thermocellum* for bioethanol production. *BMC Syst Biol* 4: 31. doi: [10.1186/1752-0509-4-31](https://doi.org/10.1186/1752-0509-4-31) PMID: [20307315](https://pubmed.ncbi.nlm.nih.gov/20307315/)
63. Plata G, Hsiao T-L, Olszewski KL, Llinás M, Vitkup D (2010) Reconstruction and flux-balance analysis of the *Plasmodium falciparum* metabolic network. *Mol Syst Biol* 6: 408. doi: [10.1038/msb.2010.60](https://doi.org/10.1038/msb.2010.60) PMID: [20823846](https://pubmed.ncbi.nlm.nih.gov/20823846/)

64. Chang RL, Ghamsari L, Manichaikul A, Hom EFY, Balaji S, et al. (2011) Metabolic network reconstruction of *Chlamydomonas* offers insight into light-driven algal metabolism. *Mol Syst Biol* 7: 518. doi: [10.1038/msb.2011.52](https://doi.org/10.1038/msb.2011.52) PMID: [21811229](https://pubmed.ncbi.nlm.nih.gov/21811229/)
65. Saha R, Verseput AT, Berla BM, Mueller TJ, Pakrasi HB, et al. (2012) Reconstruction and Comparison of the Metabolic Potential of Cyanobacteria *Cyanothece* sp. ATCC 51142 and *Synechocystis* sp. PCC 6803. *PLoS One* 7.
66. Imam S, Yilmaz S, Sohmen U, Gorzalski A, Reed J, et al. (2011) iRsp1095: A genome-scale reconstruction of the *Rhodobacter sphaeroides* metabolic network. *BMC Syst Biol* 5: 116. Available: <http://www.biomedcentral.com/1752-0509/5/116>. doi: [10.1186/1752-0509-5-116](https://doi.org/10.1186/1752-0509-5-116) PMID: [21777427](https://pubmed.ncbi.nlm.nih.gov/21777427/)
67. Andersen MR, Nielsen ML, Nielsen J (2008) Metabolic model integration of the bibliome, genome, metabolome and reactome of *Aspergillus niger*. *Mol Syst Biol* 4: 178. doi: [10.1038/msb.2008.12](https://doi.org/10.1038/msb.2008.12) PMID: [18364712](https://pubmed.ncbi.nlm.nih.gov/18364712/)
68. AbuOun M, Suthers PF, Jones GI, Carter BR, Saunders MP, et al. (2009) Genome scale reconstruction of a salmonella metabolic model: Comparison of similarity and differences with a commensal *Escherichia coli* strain. *J Biol Chem* 284: 29480–29488. doi: [10.1074/jbc.M109.005868](https://doi.org/10.1074/jbc.M109.005868) PMID: [19690172](https://pubmed.ncbi.nlm.nih.gov/19690172/)
69. Ong WK, Vu TT, Lovendahl KN, Llull JM, Serres MH, et al. (2014) Comparisons of *Shewanella* strains based on genome annotations, modeling, and experiments. *BMC Syst Biol* 8: 1–11. Available: <http://www.biomedcentral.com/1752-0509/8/31/abstract> <http://www.biomedcentral.com/content/pdf/1752-0509-8-31.pdf> <http://www.ncbi.nlm.nih.gov/pubmed/24621294>.
70. Fang K, Zhao H, Sun C, Lam CMC, Chang S, et al. (2011) Exploring the metabolic network of the epidemic pathogen *Burkholderia cenocepacia* J2315 via genome-scale reconstruction. *BMC Syst Biol* 5: 83. Available: <http://www.biomedcentral.com/1752-0509/5/83>. doi: [10.1186/1752-0509-5-83](https://doi.org/10.1186/1752-0509-5-83) PMID: [21609491](https://pubmed.ncbi.nlm.nih.gov/21609491/)
71. Ahsanul Islam M, Edwards EA, Mahadevan R (2010) Characterizing the metabolism of *Dehalococcoides* with a constraint-based model. *PLoS Comput Biol* 6.
72. Thiele I, Hyduke DR, Steeb B, Fankam G, Allen DK, et al. (2011) A community effort towards a knowledge-base and mathematical model of the human pathogen *Salmonella Typhimurium* LT2. *BMC Syst Biol* 5: 8. doi: [10.1186/1752-0509-5-8](https://doi.org/10.1186/1752-0509-5-8) PMID: [21244678](https://pubmed.ncbi.nlm.nih.gov/21244678/)
73. Becker SA, Palsson BO (2008) Context-specific metabolic networks are consistent with experiments. *PLoS Comput Biol* 4: e1000082. Available: <http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1000082#pcbi-1000082-g011>. doi: [10.1371/journal.pcbi.1000082](https://doi.org/10.1371/journal.pcbi.1000082) PMID: [18483554](https://pubmed.ncbi.nlm.nih.gov/18483554/)
74. Vu TT, Stolyar SM, Pinchuk GE, Hill EA, Kucek LA, et al. (2012) Genome-scale modeling of light-driven reductant partitioning and carbon fluxes in diazotrophic unicellular cyanobacterium *Cyanothece* sp. ATCC 51142. *PLoS Comput Biol* 8.
75. Duarte NC, Herrgård MJ, Palsson BØ (2004) Reconstruction and validation of *Saccharomyces cerevisiae* iND750, a fully compartmentalized genome-scale metabolic model. *Genome Res* 14: 1298–1309. PMID: [15197165](https://pubmed.ncbi.nlm.nih.gov/15197165/)
76. Mo ML, Palsson BO, Herrgård MJ (2009) Connecting extracellular metabolomic measurements to intracellular flux states in yeast. *BMC Syst Biol* 3: 37. doi: [10.1186/1752-0509-3-37](https://doi.org/10.1186/1752-0509-3-37) PMID: [19321003](https://pubmed.ncbi.nlm.nih.gov/19321003/)
77. Archer CT, Kim JF, Jeong H, Park JH, Vickers CE, et al. (2011) The genome sequence of *E. coli* W (ATCC 9637): comparative genome analysis and an improved genome-scale reconstruction of *E. coli*. *BMC Genomics* 12: 9. doi: [10.1186/1471-2164-12-9](https://doi.org/10.1186/1471-2164-12-9) PMID: [21208457](https://pubmed.ncbi.nlm.nih.gov/21208457/)
78. Thiele I, Swainston N, Fleming RMT, Hoppe A, Sahoo S, et al. (2013) A community-driven global reconstruction of human metabolism. *Nat Biotechnol* 31: 419–425. Available: <http://www.ncbi.nlm.nih.gov/pubmed/23455439> <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3856361&tool=pmcentrez&rendertype=abstract>. doi: [10.1038/nbt.2488](https://doi.org/10.1038/nbt.2488) PMID: [23455439](https://pubmed.ncbi.nlm.nih.gov/23455439/)
79. Reed JL, Vo TD, Schilling CH, Palsson BO (2003) An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR). *Genome Biol* 4: R54. doi: [10.1186/gb-2003-4-9-r54](https://doi.org/10.1186/gb-2003-4-9-r54) PMID: [12952533](https://pubmed.ncbi.nlm.nih.gov/12952533/)
80. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, et al. (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol Syst Biol* 3: 121. doi: [10.1038/msb4100155](https://doi.org/10.1038/msb4100155) PMID: [17593909](https://pubmed.ncbi.nlm.nih.gov/17593909/)
81. McCloskey D, Palsson BØ, Feist AM (2013) Basic and applied uses of genome-scale metabolic network reconstructions of *Escherichia coli*. *Mol Syst Biol* 9: 661. Available: <http://msb.embopress.org/content/9/1/661.abstract>. doi: [10.1038/msb.2013.18](https://doi.org/10.1038/msb.2013.18) PMID: [23632383](https://pubmed.ncbi.nlm.nih.gov/23632383/)
82. Waltemath D, Henkel R, Hälke R, Scharm M, Wolkenhauer O (2013) Improving the reuse of computational models through version control. *Bioinformatics* 29: 742–748. doi: [10.1093/bioinformatics/btt018](https://doi.org/10.1093/bioinformatics/btt018) PMID: [23335018](https://pubmed.ncbi.nlm.nih.gov/23335018/)
83. Applegate DL, Cook W, Dash S, Espinoza DG (2007) Exact solutions to linear programming problems. *Oper Res Lett* 35: 693–699. Available: <http://www.sciencedirect.com/science/article/pii/S0167637707000211>.

84. Ebrahim A (2015) SBML model collection. Available: https://github.com/opencobra/m_model_collection.
85. Acuña V, Chierichetti F, Lacroix V, Marchetti-Spaccamela A, Sagot M-FF, et al. (2009) Modes and cuts in metabolic networks: Complexity and algorithms. *BioSystems* 95: 51–60. Available: <http://www.ncbi.nlm.nih.gov/pubmed/18722501>. doi: [10.1016/j.biosystems.2008.06.015](https://doi.org/10.1016/j.biosystems.2008.06.015) PMID: [18722501](https://pubmed.ncbi.nlm.nih.gov/18722501/)
86. Yousofshahi M, Ullah E, Stern R, Hassoun S (2013) MC3: a steady-state model and constraint consistency checker for biochemical networks. *BMC Syst Biol* 7: 129. Available: <http://www.biomedcentral.com/1752-0509/7/129>. doi: [10.1186/1752-0509-7-129](https://doi.org/10.1186/1752-0509-7-129) PMID: [24261865](https://pubmed.ncbi.nlm.nih.gov/24261865/)
87. Vlassis N, Pacheco MP, Sauter T (2014) Fast Reconstruction of Compact Context-Specific Metabolic Network Models. *PLoS Comput Biol* 10: e1003424. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3894152&tool=pmcentrez&rendertype=abstract>. doi: [10.1371/journal.pcbi.1003424](https://doi.org/10.1371/journal.pcbi.1003424) PMID: [24453953](https://pubmed.ncbi.nlm.nih.gov/24453953/)