Toward standards for tomorrow's whole-cell models

Dagmar Waltemath*, Falk Schreiber, Jonathan R. Karr, Chris J. Myers, *Senior Member, IEEE,* Frank T. Bergmann, Vijayalakshmi Chelliah, Wolfram Liebermeister, Begum Alaybeyoglu, Arne T. Bittig, Paulo E. Pinto Burke, Yin Hoon Chew, Rafael S. Costa, Joseph Cursons, Muhammad Haseeb, Denis Kazakiewic, Ilya Kiselev, Vincent Knight-Schrijver, Christian Knüpfer, Matthias König, Nikita Mandrik, J. Kyle Medley, Sucheendra K. Palaniappan, Martin Scharm, Mahesh Sharma, Kieran Smallbone, Je-Hoon Song, Tom Theile, Namrata Tomar, Jannis Uhlendorf, Markus Wolfien, James T. Yurkovich, Yan Zhu, and Anna Zhukova

Manuscript received XXX XX, 2015; revised XXX XX, 201X; accepted XXX XX, 201X. Date of publication XXX XX, 201X; date of current version XXX XX, 201X. This work was supported in part by the Volkswagen Foundation (Grant to D. W. and F. S.) and the James S. McDonnell Foundation (Postdoctoral Fellowship Award in Studying Complex Systems to J. R. K.). *Asterisk indicates corresponding author.*

D. Waltemath, A. T. Bittig, M. Scharm, T. Theile, and M. Wolfien are with the Institute of Computer Science, University of Rostock, 18051 Rostock, Germany (e-mail: dagmar.waltemath@uni-rostock.de; arne.bittig@uni-rostock.de; martin.scharm@uni-rostock.de; tom.theile@uni-rostock.de; markus.wolfien@uni-rostock.de).

F. Schreiber is with the Clayton School of Information Technology, Monash University, Clayton, VIC 3800, Australia and also with the Institute of Computer Science, Martin Luther University Halle-Wittenberg, 06108 Halle, Germany (e-mail: falk.schreiber@monash.edu).

J. R. Karr is with the Department of Genetics & Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY 10029 USA (e-mail: karr@mssm.edu).

C. J. Myers is with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, Utah 84112, USA (e-mail: myers@ece.utah.edu).

F. T. Bergmann is with BioQuant, University of Heidelberg, 69120 Heidelberg, Germany (e-mail: fbergman@caltech.edu).

V. Chelliah is with the European Bioinformatics Institute, European Molecular Biology Laboratory, Cambridge CB10 1SD, UK (e-mail: viji@ebi.ac.uk).

M. Krantz and J. Uhlendorf are with the Department of Biology, Humboldt University of Berlin, 10115 Berlin, Germany (e-mail: marcus.krantz@biologie.hu-berlin.de; jannis.uhlendorf@hu-berlin.de).

W. Liebermeister and M. König are with the Institute of Biochemistry, University Medicine Charité Berlin, 10117 Berlin, Germany (e-mail: wolfram.liebermeister@gmail.com; matthias.koenig@charite.de).

P. Pir and V. Knight-Schrijver are with the Babraham Institute, Cambridge CB22 3AT, UK (e-mail: pinar.pir@babraham.ac.uk).

B. Alaybeyoglu is with the Department of Chemical Engineering, Boğaziçi University, Bebek 34342, Turkey (e-mail: begum.alaybeyoglu@boun.edu.tr).

P. E. Pinto Burke is with the Institute of Science and Technology, Federal University of So Paulo, Brazil.

Y. H. Chew is with the Centre for Synthetic and Systems Biology, University of Edinburgh, Edinburgh EH9 3BF, UK (e-mail: yinhoon.chew@ed.ac.uk).

R. S. Costa is with the Centre of Intelligent Systems-IDMEC, Instituto Superior Tcnico, University of Lisbon, 1049-001 Lisboa, Portugal (e-mail: rafael.s.costa@tecnico.ulisboa.pt).

J. Cursons is with the Department of Biomedical Engineering, School of Engineering, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: joseph.cursons@unimelb.edu.au).

M. Haseeb is with the Department of Bioinformatics, Mohammad Ali Jinnah University, Islamabad, Pakistan.

D. Kazakiewic is with the Center for Statistics, Universiteit Hasselt, Hasselt BE3500, Belgium, and also with the Center for Innovative Research, Medical University of Biaystok, Biaystok 15-089, Poland (e-mail: dzianis.kazakevich@uhasselt.be).

I. Kiselev is with the Design Technological Institute of Digital Techniques, Siberian Branch of the Russian Academy of Sciences, Novosibirsk 630090, Russia.

C. Knüpfer is with the Institut für Informatik, University of Jena, 07743 Jena, Germany (e-mail: christian.knuepfer@uni-jena.de).

Whole-cell models are promising tools for biological research, bioengineering, and medicine. However, significant work remains to achieve complete and accurate whole-cell models, including developing a strong theoretical understanding of multi-algorithm modeling, a standardized whole-cell modeling language, and an efficient general-purpose simulator. We organized the 2015 Whole-Cell Modeling Summer School to teach whole-cell modeling, as well as to evaluate the need for new whole-cell modeling standards by attempting to encode a recently published whole-cell model into SBML. We propose three SBML extensions to support transparent, reproducible whole-cell modeling: support for multi-algorithm models, support for particle-based state representation, and support for template reactions. In addition, we describe several new software tools and databases which are needed to enable researchers to encode and simulate whole-cell models including a user-friendly graphical model editor and a parallelized simulator. We also propose several new SGBN extensions. Together these new standards and software tools would accelerate whole-cell modeling.

***Index Terms*—Whole-cell modeling, Systems biology, Computational biology, Mathematical modeling, Simulation, Standards, Education**

N. Mandrik is with the Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk 630090, Russia (e-mail: manikitos@gmail.com).

J. K. Medley is with the Department of Bioengineering, University of Washington, Seattle, WA 98195, USA (e-mail: medjk@comcast.net).

S. K. Palaniappan is with the Rennes - Bretagne Atlantique Research Centre, Institute for Research in Computer Science and Automation, 35042 Rennes Cedex, France.

M. Sharma is with the Department of Pharmacoinformatics, National Institute of Pharmaceutical Education and Research, Punjab 160062, India.

K. Smallbone is with the Manchester Centre for Integrative Systems Biology, University of Manchester, Manchester M1 7DN, UK (e-mail: kieran.smallbone@manchester.ac.uk).

J.-H. Song is with the Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Republic of Korea (e-mail: song.je-hoon@kaist.ac.kr).

N. Tomar is with the Department of Dermatology, University Medicine, Friedrich-Alexander University of Erlangen-Nürnberg, 91052 Erlangen, Germany.

J. T. Yurkovich is with the Department of Bioengineering, University of California, San Diego, La Jolla, CA 92093, USA (e-mail: jyurkovich@ucsd.edu).

Y. Zhu is with Monash Institute of Pharmaceutical Sciences, Monash University, Parkville, VIC 3052, Australia.

A. Zhukova is with the Institut de Biochimie et Gntique Cellulaires, National Center for Scientific Research, and also with the University of Bordeaux, France, 33077 Bordeaux Cedex, France (e-mail: zhutchok@gmail.com).

Digital Object Identifier 10.1109/TBME.XXXX.XXXXXXX

I. Introduction

OVER the past twenty years, computational modeling has become an essential and powerful tool for biological research, bioengineering, and medicine to analyze high-throughput molecular measurements and understand the molecular details of complex biological systems. Computational modeling has been used to identify new metabolic genes [1], to add metabolic pathways to bacteria [2], and to identify potential new antimicrobial drug targets [3]. Computational models also have the potential to enable bioengineers to design new microorganisms for industrial applications such as chemical synthesis, biofuel production, and waste decontamination, as well as to enable clinicians to tailor therapy to individual patients. Realizing this potential requires more comprehensive and accurate computational models which are capable of predicting cellular behavior from genotype and improved simulation tools, as well as standardized methods for exchanging models, simulation experiments, and model visualizations [4], [5], [6], [7], [8].

Recently, researchers at Stanford University developed the first whole-cell model of the gram-positive bacterium *Mycoplasma genitalium* [9]. The model represents the life cycle of a single Mycoplasma cell including the copy number dynamics of each metabolite, RNA, and protein species and accounts for every known gene function. The model is composed of 28 sub-models, each of which is implemented using different mathematical representations including ordinary differential equations (*ODE*s), flux balance analysis (*FBA*), and Boolean rules (*BR*s), and trained using different experimental data.

The *M. genitalium* whole-cell model was implemented in MATLAB, is available open-source under the MIT license [10], and is extensively documented. This has enabled other researchers to use the model for their own research.

However, the *M. genitalium* whole-cell model simulation software is not transparent or reusable. The *M. genitalium* whole-cell model simulation software is also not user-friendly, computationally efficient, or easily maintainable. Covert and colleagues have developed several software programs including WholeCellKB [11], WholeCellSimDB [12], and WholeCellViz [13] to provide user-friendly interfaces on top of their whole-cell simulation software software. However, significant domain expertise is still required to use the *M. genitalium* model or construct new whole-cell models. New whole-cell modeling standards and simulation tools are needed to enable more researchers to develop and simulate their own whole-cell models. Such standards and software tools would accelerate whole-cell modeling. They would enable researchers to develop models more quickly, to explore models more deeply, and to evaluate models more rigorously. Furthermore, standards would simplify the submission process to model repositories such as BioModels [14], [15]. In turn, this would make models more searchable, retrievable, reusable, and comparable.

Several systems biology standards have already been developed by the COmputational Modeling in BIology NEtwork (*COMBINE*) [16] including the Systems Biology Markup Language (*SBML*) [17], the Cell Markup Language (*CellML*) [18], the Simulation Experiment Description Markup Language (*SED-ML*) [19], and the Systems Biology Graphical Notation (*SBGN*) [20]. SBML and CellML are languages for describing mathematical models including ODE, logical, and FBA models. Both have been used to build thousands of models of various intracellular pathways. SED-ML is a language for describing computational experiments. SED-ML enables scientists to reproduce a simulations by completely describing simulation setups, including the simulation algorithm and every parameter value. SBGN includes three languages for describing visual representations of models. None of these standards have been used to construct, simulate, or visualize models as complex as the *M. genitalium* whole-cell model.

We organized the 2015 Whole-Cell Modeling Summer School to train students in whole-cell modeling, as well as to evaluate the need for new standards for whole-cell modeling. The majority of the school was focused on trying to encode the *M. genitalium* whole-cell model using SBML to train students, as well as to evaluate the ability of SBML to encode whole-cell models. The ultimate scientific goal of the school was to develop an open-source whole-cell model encoded in SBML and simulated using SED-ML.

Here, we describe the summer school, outline our progress toward encoding the *M. genitalium* model using SBML, and propose several SBML and SGBN extensions to support whole-cell modeling. First, we summarize the summer school. Second, we describe our progress toward encoding the *M. genitalium* whole-cell model using SBML. Lastly, we describe the SBML and SBGN expansions and software tools needed to support whole-cell modeling.

II. The 2015 Whole-Cell Modeling Summer School

We organized the summer school to teach students how to build and encode models using COMBINE standards by attempting to encode the *M. genitalium* model using only standard representation formats.

A. Organization

The Whole-Cell Modeling Summer School was held March 9-13, 2015 at the University of Rostock in Rostock, Germany. The school was organized by Dagmar Waltemath and Falk Schreiber and supported by the Volkswagen Foundation. 45 students, nine instructors, and two organizers participated in the five-day school.

The school began with two lectures which introduced whole-cell modeling and the existing systems biology standards. Jonathan Karr from the Icahn School of Medicine at Mount Sinai, USA presented an overview of whole-cell and multi-algorithm modeling. Michael Hucka from the California Institute of Technology, USA presented an overview of the SBML, SED-ML, and SBGN standards; several open-source software tools which support these standards; and the COMBINE initiative. We also organized three discussions on multi-algorithm model composition, particle-based state representation, and random number generation.

The majority of the school was dedicated to hands-on active learning sessions in which students learned about whole-cell

modeling and the COMBINE standards by trying to encode parts of the *M. genitalium* whole-cell model using SBML. Students were divided into ten groups of four to five students, each of which was challenged to encode one or more sub-models using SBML. Each group was led by an experienced instructor.

Each day concluded with brief progress reports from each group. This facilitated discussion on common encoding challenges and model integration and provided an opportunity for groups to obtain feedback from each other.

We also organized a poster session, as well as several evening social activities to provide the students opportunities to network with each other and the instructors.

B. Educational outcomes

We surveyed the students to assess the educational outcome of the school. Most students reported gaining deep knowledge of whole-cell modeling, increased appreciation for reproducible science, and increased understanding of the SBML, SED-ML, and SBGN standards. Many students also reported learning about open-source modeling software tools relevant to their own research.

In addition, many of the students reported that the school expanded their scientific network. Several students commented that the school introduced them to potential postdoctoral positions and next year's whole-cell modeling summer school (http://www.wholecell.org/school-2016).

C. Lessons learned for organizing research-based schools

We learned several valuable lessons about how to best organize an open-ended, research-based school. First, we conclude that research-based schools should clearly outline the expected background knowledge and learning objectives and have well-organized learning activities. This helps students make informed decisions about whether to participate in the school, know how to prepare for the school, and learn efficiently. Second, we conclude that students greatly enjoy learning through open research problems rather than through prescribed training exercises. This makes students feel engaged, challenged, and connected to research. This also helps students build practical skills to complement their foundational undergraduate and graduate training. Third, we conclude that open-ended project-based schools require a high teacher-to-student ratio, a flexible schedule, and multidisciplinary project teams. A high teacher to student ratio allows students to get feedback and iterate through potential solutions quickly. A flexible schedule enables impromptu lectures and discussions. Multidisciplinary teams enable students to work through difficult problems by drawing on perspectives from multiple fields.

III. Toward an SBML-encoded whole-cell model

In addition to training young computational systems biology researchers, the second goal of the school was to attempt to encode the *M. genitalium* whole-cell model (https://github.com/CovertLab/WholeCell/releases/tag/v1.1) into SBML. To achieve this goal, most of the course was devoted to active learning sessions in which students were challenged to encode sub-models of the *M. genitalium* into SBML, integrate sub-models into a single model, and simulate models using SED-ML. During these sessions, the students and instructors were divided into ten groups. Eight of the groups were tasked with encoding one or more sub-models. The ninth group was tasked with developing a standards-compliant scheme to integrate the sub-models into a single model. This group was responsible for defining the global state variables and sub-model interfaces and developing a SED-ML scheme to simulate the integrated model. The tenth group was responsible for annotating the model and helping the other groups visualize their sub-models. Table SI lists the ten groups and all of the students and instructors.

A. Sub-model encoding

The eight sub-model encoding groups pursued various strategies to encode the sub-models using SBML. Several of the groups encoded sub-models by first reading the sub-model documentation, then drawing pathway maps using software tools such as Cell Designer [21] and VANTED [22], and finally writing scripts to generate SBML models from their maps using libSBML [23]. Other groups used modeling software tools such as BioUML [24], COPASI [25], and iBioSim [26] to encode sub-models based on their documentation. The metabolism and transcription groups also used simulation libraries such as libRoadRunner [27] and COBRApy [28]. A few of the groups encoded sub-models by converting the MATLAB code to SBML. These groups then generated SBGN maps from their SBML to better understand their sub-models.

The groups encountered several challenges to encoding the *M. genitalium* sub-models into SBML. First, most of the groups had to spend a significant amount of time reading the MATLAB code and documentation to understand the details of the *M. genitalium* sub-models because the connection between the sub-models and the associated pathway/genome database is not transparent, many of the sub-models details are implemented directly in MATLAB code rather than in a transparent language such as SBML, and the documentation only provides overviews of the sub-models. Fortunately, one of the authors of the *M. genitalium* model was available to answer questions about the model.

A second challenge to encoding the sub-models in SBML was encoding serially executed MATLAB sub-models into SBML which, because it is not a programming language, does not expose control over the order of simulation execution. This fundamental difference between programming languages and SBML makes quantitatively reproducing the *M. genitalium* model impossible. Most of the groups decided to tackle this problem by formalizing MATLAB sub-models as discrete stochastic models and simulating them using the Gillespie [29] or other approximate algorithms. For several of the sub-models, this conversion imposed an explicit internal sub-model timescale which was not present in the original MATLAB sub-model due to the lack of kinetic data for the corresponding pathway.

The fact that SBML is not a programming language and does not expose methods for arbitrary random number generation also made it challenging for groups to encode the random algorithms used by the MATLAB sub-models into SBML. For example, the MATLAB translation sub-model includes a random algorithm which assigns amino acids to individual polypeptides. Because this algorithm is not equivalent to the Gillespie algorithm, the algorithm cannot easily be encoded into SBML. Most of the groups also solved this problem by formalizing sub-models as stochastic models. Even if it were possible to transcode the MATLAB sub-models directly into SBML, it would still be difficult to quantitatively reproduce the MATLAB simulations because SBML does not expose control over the random number generator algorithm or seed. Consequently, it would only be feasible to compare the first two moments of the MATLAB and transcoded model simulations.

To encode many of the sub-models into SBML, the groups also had to either enumerate the hybrid population/particle-based state representation used by the MATLAB sub-models or approximate the MATLAB sub-models. The groups responsible for the transcription and translation sub-models chose to approximate the MATLAB sub-models by eliminating the internal dynamics of the polymerization of each RNA and polypeptide. Consequently, these sub-models no longer track the progress of individual RNA polymerases and ribosomes, account for base-specific transcription or translation rates, or predict RNA polymerase collisions. The groups responsible for the DNA sub-models including replication, replication initiation, and transcriptional regulation, chose to enumerate the sparse chromosome representation used by the MATLAB model by creating Boolean indicator variables to represent the existence and protein-binding status of each chromosome base. This enumerated representation requires millions of variables. Consequently, the corresponding SBML XML files are impractical to read and computationally expensive to simulate. Enumerating the rules which govern the joint values of the enumerated variables, such as the rules which represent the steric effects of DNA-bound proteins by preventing proteins from binding neighboring bases, is also impractical. Furthermore, the enumerated SBML files are impractical for humans to read, edit, or maintain.

The lack of universal SBML simulator support for arrays was another challenge to encoding the MATLAB sub-models into SBML. All of the groups overcame the lack of array support by enumerating individual array elements of the MATLAB and all matrix algebra computations. This creates verbose SBML files which are more difficult to interpret, maintain, and edit. Enumerating the matrix algebra computations also increases the computational cost of simulation.

Together, these five challenges made it very difficult for the groups to encode most of the MATLAB sub-models into SBML. Going forward, SBML and the SBML simulators should be expanded to provide support for random number generation, particle-based state representation, and arrays.

B. Model integration

The integration group was responsible for assembling the individual sub-models into a single model including defining the global state variables, defining the interfaces exposed by the sub-models to the global state variables, and developing a scheme for managing simultaneous writing of shared state variables. The integration group defined the global state variables as the union of all state variables shared by at least two sub-models rather than by explicitly defining a set of global state variables as done by the MATLAB simulator. The advantages of this approach are that sub-model developers are not also required to develop global state variables and that it minimizes the number of global state variables. The disadvantages of this approach are that the total set of variables is less transparent and that it requires users to learn all of the sub-models and their naming conventions to analyze model simulations.

The integration group standardized the interfaces exposed by the individual sub-models by defining a variable naming convention. This naming convention ensures, for example, that the copy numbers of each protein species are represented by variables with the same names in each of the sub-models. This convention makes it clear how multiple local sub-model variables map onto the same global variable. Specifically, the integration group chose to use the same variable names as those used by the MATLAB implementation. Matrix and particle-based variables were enumerated by creating multiple variables with names containing additional suffixes to indicate their identity.

The primary challenge faced by the integration group was how to handle concurrent editing of shared state variables by multiple sub-models. The integration group explored several potential strategies to manage concurrent writing. First, they explored sequentially simulating the sub-models and updating the global state variables. This avoids the need for more complex strategies to merge variable changes. However, under this approach sub-models are simulated with different variable values within each timestep. Consequently, simulation predictions are sensitive to the sub-model execution order.

The integration group also explored several more complex sub-model integration strategies which enable all of the sub-models to be simulated with the same variable values within each timestep. These strategies included reducing the sub-model integration timestep such that sub-models do not request conflicting variable changes; dividing each of the shared state variables into separate, independent sub-variables for each sub-model, simulating the sub-models in parallel, and merging the sub-variables to compute the update global values; and using semaphores to manage concurrent variable changes whereby at each timestep sub-models request sets of atomic state variables changes and a controller decides which change sets are processed. Each of these strategies has different advantages and disadvantages. The first strategy is the simplest to understand and implement, but is computationally expensive. The second strategy is simple to implement and computationally efficient for independent variables, but is difficult to implement for sets of non-independent variables such as those which represent the protein occupancy of the chromosome. The third strategy is the most complex to implement, but is more general than the second strategy and more computationally efficient than the first. The integration group tested their sub-model integration

strategies using iBioSim because iBioSim is one of the only SBML simulators to support all of the existing needed SBML packages including hierarchical model composition (*comp*), arrays, and flux balance constraints (*fbc*).

The lack of an SBML simulator which supports multi-algorithm model composition was another challenge to integrating the sub-models. The integration group plans to overcome this limitation by adding support for multi-algorithm to iBioSim.

C. Annotation, documentation, and visualization

The documentation group was responsible for annotating the model. The goal was to annotate every model entity with a cross reference to an external database such as ChEBI [30] as in the case of small molecules and/or in terms of other model entities as in the case of chemical reactions. The group wrote scripts to search molecular biology databases such as ChEBI for every entity contained in the *M. genitalium* model. The main problem faced by the documentation group was that many model entities are not currently represented by any molecular biology database. This shortcoming can easily be overcome by proposing new ontology terms to the databases. This problem highlights the need for expanded molecular biology databases to aggregate data on more biological molecules and interactions.

The documentation group also helped the other groups map their sub-models by providing advice on SBGN and SBGN-compatible drawing tools such as VANTED [22]. The main visualization problem encountered by the documentation group was that whole-cell models require large maps which must be manually arranged to produce intuitive diagrams.

D. Progress

The students produced preliminary SBML and SBGN-ML versions of many of the *M. genitalium* whole-cell model sub-models. However, significant work remains to finish encoding the sub-models, integrate the sub-models into a single model, and test the sub-models and the combined model. Complete drafts are only available for a few of the simplest sub-models. The current drafts of most of the more complex sub-models are also greatly simplified compared to their MATLAB versions. For example, the transcription and translation sub-models drafts do not represent the polymerization of individual bases and the DNA sub-models do not account for the protein occupancy of the chromosome. In addition, none of the SBML-encoded sub-models have been thoroughly tested by replicating the unit tests from the MATLAB implementation. Furthermore, none of the SBML sub-models have been thoroughly documented and complete SBGN maps are not yet available for any of the sub-models.

E. Future steps

We hope to finish encoding the *M. genitalium* sub-models into SBML and integrate the SBML-encoded sub-models into a single model simulatable by open-source software tools such as COPASI, libRoadRunner, BioUML, and iBioSim. Several students and instructors have continued to keep working and meeting online. Several students and instructors also plan to participate in a second meeting in October 2015 which will be held at the University of Utah, USA immediately prior to the 2015 COMBINE Forum.

Going forward, we plan to publish SBML-encoded versions of each of the *M. genitalium* sub-models to BioModels, along with SED-ML tests, SBGN maps, and textual documentations. This would make the sub-models searchable, retrievable, and reusable by other scientists. We believe this would be a valuable community resource. It would demonstrate how to build a whole-cell model and enable other researchers to build upon the *M. genitalium* sub-models. Ultimately, SBML, SBGN, the SBML and SBGN software tools also need to be extended to facilitate whole-cell modeling.

IV. Toward SBML-, SED-ML-, and SBGN-based standards for whole-cell modeling

Prior to the school, SBML, SED-ML, and SBGN had never been used to represent whole-cell or similarly complex models. Consequently, not surprisingly, the summer school revealed several limitations of SBML, SBGN, and the existing simulation software for large models. Importantly, the summer school facilitated discussions among modelers, tool developers, and model curators about how to best expand the existing standards and software tools to overcome these limitations.

A. Current limitations

As discussed above, SBML does not support multi-algorithm modeling, particle-based state representation, or arbitrary random number generation and the current SBML simulators have limited support for arrays. Consequently, SBML cannot efficiently simulate models with large, combinatorial state spaces; represent arbitrary stochastic models; or efficiently simulate arbitrary mathematical models involving matrix algebra operations. State spaces must be enumerated, stochastic models must be described using a ratio time scale and simulated using the Gillespie or other approximate algorithm, and matrix algebra operations must be expanded. For the *M. genitalium* whole-cell model this would result in large, unmanageable XML files and millions of variables.

Furthermore, no general-purpose SBML simulation software is currently capable of efficiently simulating models involving millions of variables and no SBML simulation software supports multi-algorithm modeling. In addition, no SBML editing tool provides researchers a user-friendly interface for editing SBML files containing millions of variables. Consequently, currently, whole-cell model SBML files must be generated by scripts and cannot be easily edited.

The summer school also revealed several limitations of the graphical representation scheme SBGN and the existing SBGN viewers for whole-cell models. First, whole-cell modeling requires hybrid SBGN maps which contain all three types of nodes: Process Descriptions, Entity Relationships, and Activity Flows. Currently, SBGN maps can only contain one type of node. Second, whole-cell modeling requires new automatic layout algorithms suitable for large maps. The existing layout

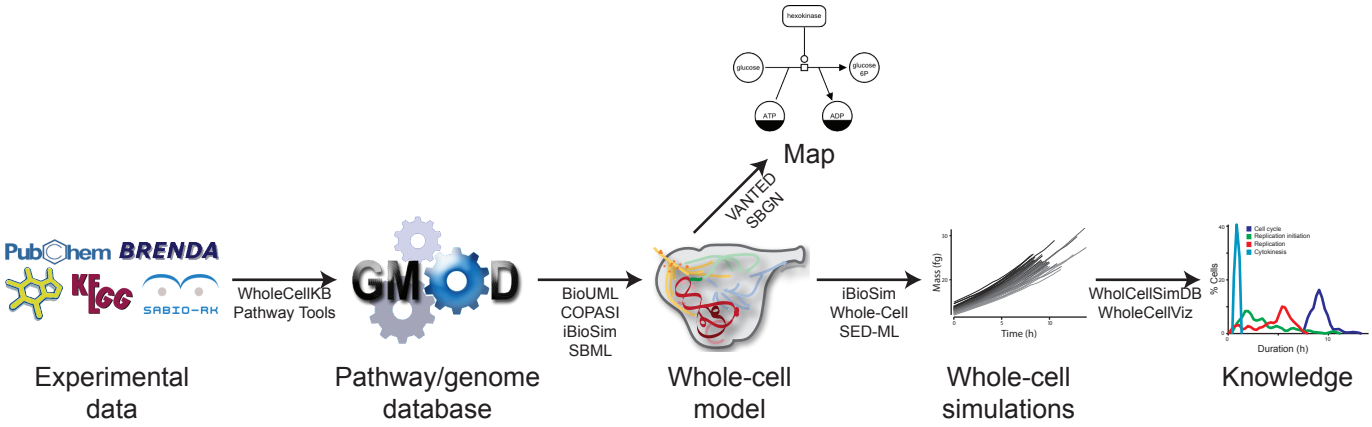

Figure 1. Whole-cell modeling pipeline. First, researchers will assemble experimental data into pathway/genome databases. Second, researchers will use pathway/genome databases to construct sub-models. Third, researchers will use multi-algorithm simulators to conduct in silico experiments. Lastly, researchers will analyze in silico experiments to learn new biology.

algorithms are unable to construct intuitive maps of many of the pathways that need to be in whole-cell models. Third, in order to effectively visualize complex SGBN maps of whole-cell models, SBGN viewers must be able to display maps at different levels of granularity by automatically reducing maps.

The summer school did not reveal any limitations of SED-ML for describing whole-cell model simulations.

B. Standard extensions

Taken together, expanded standards, simulation software tools, and databases are needed to facilitate accurate, reproducible whole-cell modeling. We propose three SBML expansions to facilitate whole-cell modeling. First, the SBML comp package must be expanded to support models composed of sub-models implemented with different simulation algorithms. Currently, the comp package only supports models composed of sub-models each implemented using the same simulation algorithm. In addition, the existing SBML simulators must be expanded to support multi-algorithm simulations and/or new simulators must be developed which support the expanded package. This requires significantly more research to determine the best ways to integrate heterogeneous sub-models, including rigorously evaluating the advantages and disadvantages of each of the schemes proposed by the integration group. Significant effort will also be needed to develop a parallelized simulator which is capable of quickly simulating complex whole-cell models.

Second, a new SBML package must be created to support hybrid population/particle-based state representations such as those used by BioNetGen [31], [32] and NFSim [33]. In parallel, the existing SBML simulators must be expanded to support this new package and/or new SBML simulators must be developed. This would enable modelers to compactly describe and efficiently simulate models with large, combinatorial state spaces. The compact descriptions enabled by this package would also make models more transparent and easier to maintain and expand.

A new SBML package must also be created to support reaction templates so that, for example, translation could be described using a single reaction template and arrays of mRNA-specific translation initiation rates and codon-specific elongation rates. Such reaction templates would enable whole-cell and other large models to be compactly described, and consequently easily interpretable, maintainable, and editable. By separating the mathematical descriptions and quantitative parameter values, reaction templates would also make the connection between dynamical models and the experimental data used to inform their parameter values more transparent. The new reaction templates could be expanded for backward compatibility with older SBML simulators.

New user-friendly graphical editors must also be developed to enable researchers to easily build SBML files which take advantage of these new features. These graphics editors must also allow researchers to transparently map model parameters onto experimental data organized in pathway/genome databases.

In addition, our molecular biology databases such as ChEBI must be expanded to enable researchers to rigorously define whole-cell models in terms of external entities. SBGN must also be expanded to support hybrid diagrams and the SBGN viewers must be expanded to support more automatic layout algorithms, automatic model reduction, and contextual zooming. These features would enable researchers to use SBGN to map whole-cell and other large models.

Together, these SBML, SBGN, software, and database expansions would enable more researchers to more easily build, manage, simulate, and reproduce whole-cell models and simulations. These new standards and software tools would also enable researchers to build more comprehensive and more accurate models, including of human cells. Ultimately, these new standards and software would enable whole-cell modeling to support rational biological design and personalized medicine.

C. The whole-cell modeling pipeline

We anticipate that such expanded standards and tools will enable a four step approach to whole-cell model-driven discovery (Figure 1). First, researchers will assemble experimental data from numerous sources including databases such

as SABIO-RK [34] and UniProt [35] into pathway/genome databases using software tools such as Pathway Tools [36] and WholeCellKB. Second, researchers will use pathway/genome databases and graphical modeling tools such as BioUML, COPASI, and iBioSim to build sub-models and encode them using transparent languages such as SBML. Third, multi-algorithm simulators will be used to conduct in silico experiments. Lastly, software tools such as WholeCellSimDB and WholeCellViz will be used to discover new biology through exploring, visualizing, and analyzing in silico experiments.

V. Conclusion

The 2016 Whole-Cell Modeling Summer School provided 45 young scientists hands-on training in whole-cell and multi-algorithm modeling through attempting to encode the *M. genitalium* whole-cell model into SBML. Additional summer schools and courses are needed to provide students with deeper theoretical training in dynamical modeling, multi-algorithm modeling, model reduction, and parameter estimation, as well as practical training in model construction including data curation, model building, numerical optimization, model testing, and model analysis.

The summer school also made significant strides toward encoding the *M. genitalium* whole-cell model using SBML for simulation by open-source software. The students developed preliminary SBML versions of all of the sub-models of the *M. genitalium* model. Since the summer school, several students have continued to encode the *M. genitalium* model, and several of the students and instructors are participating in a second meeting prior to 2015 COMBINE Forum at the University of Utah in Salt Lake City, USA. Ultimately, we hope to publish an SBML-encoded version of the model to BioModels.

However, significant work remains to complete, test, integrate, simulate, and document the SBML-encoded version of the *M. genitalium* model. Currently, SBML cannot represent whole-cell models, there is no simulation software program which could efficiently simulate an SBML-encoded whole-cell model, and there is no graphical editor which is capable of constructing, editing, or visualizing an SBML-encoded whole-cell model. SBML must be expanded to support multi-algorithm modeling, template reactions, particle-based state representation, and arrays and an efficient simulation software program and a user-friendly model editor must be developed to enable modelers to easily construct and simulate whole-cell models. New parameter estimation, model testing, and visual analysis tools must also be developed to enable researchers to effectively use SBML-encoded whole-cell models for scientific research. In addition, our molecular biology databases must be expanded to facilitate whole-cell model annotation. Furthermore, SBGN and the SBGN viewers must also be expanded to support hybrid diagrams, automatic graph layout, automatic graph reduction, and contextual zooming.

In summary, we believe that whole-cell modeling has the potential to be an important tool for biological discovery, bioengineering, and medicine. Achieving this potential requires improved standards for describing whole-cell models, as well as new general-purpose simulation software for reproducibly simulating whole-cell models. In turn, this requires expanding the whole-cell modeling field including training additional young researchers.

References

[1] J. L. Reed, T. R. Patel, K. H. Chen, A. R. Joyce, M. K. Applebee, C. D. Herring, O. T. Bui, E. M. Knight, S. S. Fong, and B. O. Palsson, "Systems approach to refining genome annotation," *PNAS*, vol. 103, no. 46, pp. 17 480–17 484, 2006.

[2] J. W. Lee, D. Na, J. M. Park, J. Lee, S. Choi, and S. Y. Lee, "Systems metabolic engineering of microorganisms for natural and non-natural chemicals," *Nature Chemical Biology*, vol. 8, no. 6, pp. 536–546, 2012.

[3] D. S. Lee, H. Burd, J. Liu, E. Almaas, O. Wiest, A. L. Barabási, Z. N. Oltvai, and V. Kapatral, "Comparative genome-scale metabolic reconstruction and flux balance analysis of multiple *Staphylococcus aureus* genomes identify novel antimicrobial drug targets," *Journal of Bacteriology*, vol. 191, no. 12, pp. 4015–4024, 2009.

[4] D. N. Macklin, N. A. Ruggero, and M. W. Covert, "The future of whole-cell modeling," *Current Opinion in Biotechnology*, vol. 28, pp. 111–115, 2014.

[5] J. R. Karr, K. Takahashi, and A. Funahashi, "The principles of whole-cell modeling," *Current Opinion in Microbiology*, vol. (in press), 2015.

[6] J. R. Karr, A. H. Williams, J. D. Zucker, A. Raue, B. Steiert, J. Timmer, C. Kreutz, DREAM8 Parameter Estimation Challenge Consortium, S. Wilkinson, B. A. Allgood, B. M. Bot, B. R. Hoff, M. R. Kellen, M. W. Covert, G. A. Stolovitzky, and P. Meyer, "Summary of the DREAM8 Parameter Estimation Challenge: Toward Parameter Identification for Whole-Cell Models," *PLoS Comput Biol*, vol. 11, no. 5, p. e1004096, 2015.

[7] M. Hucka, D. P. Nickerson, G. D. Bader, F. T. Bergmann, J. Cooper, E. Demir, A. Garny, M. Golebiewski, C. J. Myers, F. Schreiber *et al.*, "Promoting coordinated development of community-based information standards for modeling in biology: the COMBINE initiative," *Frontiers in Bioengineering and Biotechnology*, vol. 3, 2015.

[8] E. Klipp, W. Liebermeister, A. Helbig, A. Kowald, and J. Schaber, "Systems biology standards - the community speaks," *Nature Biotechnology*, vol. 25, no. 4, pp. 390–391, 2007.

[9] J. R. Karr, J. C. Sanghvi, D. N. Macklin, M. V. Gutschow, J. M. Jacobs, B. Bolival, Jr, N. Assad-Garcia, J. I. Glass, and M. W. Covert, "A whole-cell computational model predicts phenotype from genotype," *Cell*, vol. 150, no. 2, pp. 389–401, 2012.

[10] J. R. Karr, J. C. Sanghvi, D. N. Macklin, J. M. Jacobs, and M. W. Covert. Whole Cell Model. [Online]. Available: https://github.com/CovertLab/WholeCell/releases/tag/v1.1

[11] J. R. Karr, J. C. Sanghvi, D. N. Macklin, A. Arora, and M. W. Covert, "WholeCellKB: model organism databases for comprehensive whole-cell models," *Nucleic Acids Res*, vol. 41, no. Database issue, pp. D787–92, 2013.

[12] J. R. Karr, N. C. Phillips, and M. W. Covert, "WholeCellSimDB: a hybrid relational/HDF database for whole-cell model predictions," *Database*, vol. 2014, no. pii, p. bau095, 2014.

[13] R. Lee, J. R. Karr, and M. W. Covert, "WholeCellViz: data visualization for whole-cell models," *BMC Bioinformatics*, vol. 14, p. 253, 2013.

[14] N. Juty, R. Ali, M. Glont, S. Keating, N. Rodriguez, M. Swat, S. Wimalaratne, H. Hermjakob, N. Le Novère, C. Laibe *et al.*, "Biomodels: Content, features, functionality, and use," *CPT: Pharmacometrics & Systems Pharmacology*, vol. 4, no. 2, pp. 1–14, 2015.

[15] V. Chelliah, N. Juty, I. Ajmera, R. Ali, M. Dumousseau, M. Glont, M. Hucka, G. Jalowicki, S. Keating, V. Knight-Schrijver *et al.*, "Biomodels: ten-year anniversary," *Nucleic acids research*, vol. 43, no. D1, pp. D542–D548, 2015.

[16] N. Le Novère, M. Hucka, N. Anwar, G. D. Bader, E. Demir, S. Moodie, and A. Sorokin, "Meeting report from the first meetings of the Computational Modeling in Biology Network (COMBINE)," *Standards in Genomic Sciences*, vol. 5, no. 2, p. 230, 2011.

[17] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E.-D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. L. Novère, L. M. Loew, D. Lucio, P. Mendes, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang,

"The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[18] W. J. Hedley, M. R. Nelson, D. P. Bullivant, and P. F. Nielson, "A short introduction to CellML," *Philosophical Transactions of the Royal Society of London A*, vol. 359, pp. 1073–1089, 2001.

[19] D. Waltemath, R. Adams, F. Bergmann, M. Hucka, F. Kolpakov, A. Miller, I. Moraru, D. Nickerson, S. Sahle, J. Snoep, and N. Le Novère, "Reproducible computational biology experiments with SED-ML—the Simulation Experiment Description Markup Language," *BMC Systems Biology*, vol. 5, no. 1, p. 198, 2011.

[20] N. Le Novère, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M. Aladjem, S. M. Wimalaratne, F. T. Bergman, R. Gauges, P. Ghazal, H. Kawaji, L. Li, Y. Matsuoka, A. Villéger, S. E. Boyd, L. Calzone, M. Courtot, U. Dogrusoz, T. Freeman, A. Funahashi, S. Ghosh, A. Jouraku, S. Kim, F. Kolpakov, A. Luna, S. Sahle, E. Schmidt, S. Watterson, G. Wu, I. Goryanin, D. B. Kell, C. Sander, H. Sauro, J. L. Snoep, K. Kohn, and H. Kitano, "The systems biology graphical notation," *Nature Biotechnology*, vol. 27, pp. 735–741, 2009.

[21] A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi, and H. Kitano, "Celldesigner 3.5: a versatile modeling tool for biochemical networks," *Proceedings of the IEEE*, vol. 96, no. 8, pp. 1254–1265, 2008.

[22] H. Rohn, A. Junker, A. Hartmann, E. Grafahrend-Belau, H. Treutler, M. Klapperstück, T. Czauderna, C. Klukas, and F. Schreiber, "VANTED v2: a framework for systems biology applications," *BMC Systems Biology*, vol. 6, p. 139, 2012.

[23] B. J. Bornstein, S. M. Keating, A. Jouraku, and M. Hucka, "LibSBML: an API library for SBML," *Bioinformatics*, vol. 24, no. 6, pp. 880–881, 2008.

[24] F. Kolpakov, "BioUML: visual modeling, automated code generation and simulation of biological systems," *Proceedings BGRS*, vol. 3, pp. 281–285, 2006.

[25] P. Mendes, S. Hoops, S. Sahle, R. Gauges, J. Dada, and U. Kummer, "Computational modeling of biochemical networks using COPASI," *Methods in Molecular Biology*, vol. 500, pp. 17–59, 2009.

[26] C. Madsen, C. J. Myers, T. Patterson, N. Roehner, J. T. Stevens, and C. Winstead, "Design and test of genetic circuits using iBioSim," *IEEE Des Test Comput*, vol. 29, no. 3, 2012.

[27] E. T. Somogyi, J.-M. Bouteiller, J. A. Glazier, M. Knig, J. K. Medley, M. H. Swat, and H. M. Sauro, "libroadrunner: a high performance sbml simulation and analysis library," *Bioinformatics*, 2015. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/early/2015/07/13/bioinformatics.btv363.abstract

[28] A. Ebrahim, J. A. Lerman, B. O. Palsson, and D. R. Hyduke, "Cobrapy: constraints-based reconstruction and analysis for python," *BMC systems biology*, vol. 7, no. 1, p. 74, 2013.

[29] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977. [Online]. Available: http://dx.doi.org/10.1021/j100540a008

[30] J. Hastings, P. de Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, V. Muthukrishnan, G. Owen, S. Turner, M. Williams, and C. Steinbeck, "The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013," *Nucleic Acids Res*, vol. 41, no. Database issue, pp. D456–D463, 2013.

[31] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana, "Rules for modeling signal-transduction systems," *Sci STKE*, vol. 2006, no. 344, p. re6, 2006.

[32] J. S. Hogg, L. A. Harris, L. J. Stover, N. S. Nair, and J. R. Faeder, "Exact hybrid particle/population simulation of rule-based models of biochemical systems," *PLOS Comp Biol*, vol. 10, no. 4, p. e1003544, 2014.

[33] M. W. Sneddon, J. R. Faeder, and T. Emonet, "Efficient modeling, simulation and coarse-graining of biological complexity with NFsim," *Nat Methods*, vol. 8, no. 2, pp. 177–183, 2011.

[34] U. Wittig, R. Kania, M. Golebiewski, M. Rey, L. Shi, L. Jong, E. Algaa, A. Weidemann, H. Sauer-Danzwith, S. Mir, O. Krebs, M. Bittkowski, E. Wetsch, I. Rojas, and W. Mller, "SABIO-RK–database for biochemical reaction kinetics," *Nucleic Acids Res*, vol. 40, no. Database issue, pp. D790—D796, 2012.

[35] UniProt Consortium, "UniProt: a hub for protein information," *Nucleic Acids Res*, vol. 43, no. Database issue, pp. D204–D212, 2015.

[36] P. D. Karp, S. M. Paley, M. Krummenacker, M. Latendresse, J. M. Dale, T. J. Lee, P. Kaipa, F. Gilham, A. Spaulding, L. Popescu, T. Altman, I. Paulsen, I. M. Keseler, and R. Caspi, "Pathway Tools version 13.0: integrated software for pathway/genome informatics and systems biology," *Brief Bioinform*, vol. 11, no. 1, pp. 40–79, 2010.

2015 Whole-Cell Modeling Summer School included the 56 researchers listed in Table SI.

Table SI
2015 WHOLE-CELL MODELING SUMMER SCHOOL PARTICIPANTS.

Group	Participant	Affiliation
Cytokinesis	Naveen Kumar Aranganathan	University Paris-Sud, France
	Daniel Alejandro Priego Espinosa	National Autonomous University of Mexico, Mexico
	Ilya Kiselev	Siberian Branch of the Russian Academy of Sciences Novosibirsk, Russia
	Wolfram Liebermeister	Charité Medical University of Berlin, Germany
	Yan Zhu	Monash University, Australia
DNA repair	Arne Bittig	University of Rostock, Germany
	Vijayalakshmi Chelliah	European Bioinformatics Institute, UK
	Audald Lloret-Vilas	European Bioinformatics Institute, UK
	Mahesh Sharma	National Institute of Pharmaceutical Education and Research, India
	Namrata Tomar	Friedrich-Alexander University of Erlangen-Nürnberg, Germany
Metabolism	Kambiz Baghalian	University of Oxford, UK
	Frank T. Bergmann	University of Heidelberg, Germany
	Rafael Sousa Costa	University of Lisbon, Portugal
	Matthias König	Charité Medical University of Berlin, Germany
	Kieran Smallbone	University of Manchester, UK
	Milenko Tokic	Swiss Federal Institute of Technology in Lausanne, Switzerland
Protein	Begum Alaybeyoglu	Boğaziçi University, Turkey
	Matteo Cantarelli	OpenWorm, UK
	Yin Hoon Chew	University of Edinburgh, UK
	Marcus Krantz	Humboldt University of Berlin, Germany
	Daewon Lee	Korea Advanced Institute of Science and Technology, Republic of Korea
Replication	Vincent Knight-Schrijver	Babraham Institute, UK
	Je-Hoon Song	Korea Advanced Institute of Science and Technology, Republic of Korea
	Jannis Uhlendorf	Humboldt University of Berlin, Germany
	Dagmar Waltemath	University of Rostock, Germany
	James T. Yurkovich	University of California, San Diego, USA
	Anna Zhukova	National Center for Scientific Research and University of Bordeaux, France
Replication initiation	Harold Gomez	Boston University, USA
	Jens Hahn	Humboldt University of Berlin, Germany
	Michael Hucka	California Institute of Technology, USA
	Nikita Mandrik	Siberian Branch of the Russian Academy of Sciences Novosibirsk, Russia
	Martin Scharm	University of Rostock, Germany
	Florian Wendland	University of Rostock, Germany
RNA	Tuure Hameri	Swiss Federal Institute of Technology in Lausanne, Switzerland
	J. Kyle Medley	University of Washington, USA
	Sucheendra Kumar Palaniappan	Institute for Research in Computer Science and Automation, France
	Pinar Pir	Babraham Institute, UK
	Natalie Stanford	University of Manchester, UK
	Markus Wolfien	University of Rostock, Germany
Translation	Joseph Cursons	University of Melbourne, Australia
	Muhammad Haseeb	Mohammad Ali Jinnah University, Pakistan
	Daniel Hernandez	Swiss Federal Institute of Technology in Lausanne, Switzerland
	Denis Kazakiewicz	University of Hasselt, Belgium and Medical University of Biaystok, Poland
	Pedro Mendes	University of Manchester, UK
	Hojjat Naderi Meshkin	Academic Center for Education, Culture and Research, Iran
Integration	Paulo Eduardo Pinto Burke	Federal University of São Paulo, Brazil
	Tobias Czauderna	Monash University, Australia
	Bertrand Moreau	CoSMo Company, France
	Chris J. Myers	University of Utah, USA
	Thawfeek Mohamed Varusai	University College Dublin, Ireland
	Argyris Zardilis	University of Edinburgh, UK
Annotation, documentation and visualization	Christian Knüpfer	University of Jena, Germany
	Falk Schreiber	Monash University, Australia
	Tom Theile	University of Rostock, Germany
Modeling instructor	Jonathan R. Karr	Icahn School of Medicine at Mount Sinai, USA