Toward standards for tomorrow's whole-cell models

Dagmar Waltemath[1], Falk Schreiber[2,3], Jonathan R. Karr[4], Begum Alaybeyoglu[17], Yin Hoon Chew[18], Rafael S. Costa[19], Muhammad Haseeb[20], Vincent Knight-Schrijver[16], Sucheendra K. Palaniappan[21], Martin Scharm[1], Kieran Smallbone[22] and Markus Wolfien[1]

[1]Department of Systems Biology and Bioinformatics, University of Rostock
[2]Clayton School of Information Technology, Monash University
[3]Institute of Computer Science, Martin Luther University Halle-Wittenberg
[4]Department of Genetics & Genomic Sciences, Icahn School of Medicine at Mount Sinai
[17]Department of Chemical Engineering, Boğaziçi University
[18]Centre for Synthetic and Systems Biology, University of Edinburgh
[19]Instituto Superior Técnico, University of Lisbon
[20]Department of Bioinformatics, Mohammad Ali Jinnah University
[21]Rennes - Bretagne Atlantique Research Centre, Institute for Research in Computer Science and Automation
[22]Manchester Centre for Integrative Systems Biology, University of Manchester

Whole-cell models are a promising tool for biological research, bioengineering, and medicine. However, significant work remains to achieve fully 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 assessing the ability of SBML to represent a recently published whole-cell model. We describe three SBML extensions which are needed to support transparent, reproducible whole-cell modeling: support for multi-algorithm model composition, support for particle-based representation, and support for template reactions. In addition, we describe several new software tools which are needed to enable whole-cell modeling including user-friendly graphical model editors and parallelized simulators. We believe that together these new standards and software tools would enable researchers to realize the full potential of whole-cell modeling.

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

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 already been used to identify new metabolic genes [1], add metabolic pathways to bacteria [2], and 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 design personalized medical therapies tailored to each individual patient's unique genome. Realizing this potential requires more comprehensive and accurate computational models which are capable of predicting cellular behavior from genotype, as well as standardized methods for exchanging models, simulation experiments, and model visualizations [4], [5], [6], [7].

Recently, researchers at Stanford University developed the first whole-cell model of the gram-positive bacterium *Mycoplasma genitalium* [8]. 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 comprised 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, and is extensively documented. This has enabled other researchers to use the model for their own research and enabled educators to use the model to teach systems biology.

However, the *M. genitalium* whole-cell model software is not transparent or reusable. The *M. genitalium* whole-cell model software is also not user-friendly, computationally efficient, or easily maintainable. Consequently, significant domain expertise is required to use the model or construct new whole-cell models. New whole-cell modeling standards and simulation tools are needed to enable more researchers to develop and use their own whole-cell models. Such standards would accelerate the whole-cell modeling field. They would enable researchers to develop models more quickly, more deeply explore model predictions, and more rigorously evaluate models. Furthermore, they would enable researchers to contribute whole-cell models to model repositories such as BioModels [9] which in turn, 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*) [10] including the Systems Biology Markup Lan-

Corresponding author email: dagmar.waltemath@uni-rostock.de

guage (*SBML*) [11], the Cell Markup Language (*CellML*) [12], the Simulation Experiment Description Markup Language (*SED-ML*) [13], and the Systems Biology Graphical Notation (*SBGN*) [14]. SBML and CellML are standard 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 standard language for describing computational experiments. SED-ML enables computational scientists to reproduce simulations by describing simulation setups in detail, including the simulation algorithm and every parameter value. SBGN is a standard which provides three language for describing visual representations of models. However, 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 recode the *M. genitalium* whole-cell model using SBML to train students and 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 extensions to support whole-cell modeling. First, we summarize the organization and outcomes of the summer school. Second, we describe our progress toward encoding the *M. genitalium* whole-cell model using SBML. Lastly, we describe the SBML, SED-ML, and SBGN expansions needed to encode whole-cell models.

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 through encoding the *M. genitalium* model using only standard representation formats and open-source simulation software.

A. Organization

The 2016 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 tutors, 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; open-source software tools which support those standards; and the COMBINE initiative. In addition to the introductory lectures, we organized three discussions on three common SBML encoding problems: particle-based state representation, random number generation, and sub-model integration.

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 recode 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 standards encoding challenges and model integration and provided opportunity for groups to obtain feedback from the rest of the school.

We also organized a poster session, as well as several evening social activities to provide students opportunities to network.

B. Educational outcomes

Following the school, we surveyed the students to assess the educational outcome of the school. Most students reported gaining deeper knowledge of whole-cell modeling, increased appreciation for reproducibile science, and increased understanding of the SBML, SED-ML, and SBGN standards. Many students also reported that they learned about valuable open-source modeling software tools for their graduate research.

In addition, many of the students reported that the school expanded their scientific network. Several students reported 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 found 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 found that graduate students greatly enjoy learning through tackling open research problems rather than through prescribed toy training exercises. This makes students feel engaged, challenged, and connected to the forefront of research. This also helps students build practical skills which complement their theoretical training from their undergraduate and graduate coursework. Third, we found that open-ended project-based schools are most successful with 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 allows organizers to arrange impromptu lectures and discussions to provide additional background information as needed to solve the research problem. Multidisciplinary teams help students work through difficult problems by drawing on multiple perspectives and backgrounds 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 develop an standard-encoded version the *M. genitalium* whole-cell model. To achieve this goal, most of the course was devoted to active learning sessions in which students were challenged to recode sub-models of the *M. genitalium* model using SBML and integrate and simulate the recoded sub-models into a single model using SBML and SED-ML. During these sessions the students and tutors were divided into ten groups. Eight of the groups were tasked with recoding one or more sub-models. The ninth group was tasked with developing a standards-compliant scheme to integrate the recoded SBML 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 developing an annotation scheme and helping the other groups document and visualize their sub-models. Table SI lists the ten groups and all of the students and tutors.

A. Sub-model encoding

The eight sub-model recoding 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 [15] and VANTED [16], and finally writing scripts to automatically generate SBML representations from their maps. Other groups used modeling software tools such as BioUML [18], COPASI [17], and iBioSim [19] to encode sub-models based on their documentation. A few of the groups encoded sub-models by converting the original MATLAB code to SBML. These groups then automatically 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, because many of the sub-models details are implemented directly in MATLAB code rather than in a transparent declarative modeling language such as SBML or in the pathway/genome database, and because the documentation only provides overviews of the sub-models and does not describe all of their quantitative details. Fortunately, one of the authors of the *M. genitalium* model participated in the school and was available to answer questions about the model and its MATLAB implementation.

A second challenge to encoding sub-models in SBML was to encode the 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 in SBML 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 or other approximate algorithms. For several of the sub-models this conversion imposes an internal sub-model timescale which was not part of the original MATLAB model due to the lack of kinetic data describing that cellular 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. Most of the groups also solved this problem by formalizing sub-models as stochastic models which are simulated using random numbers. 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 to read, edit, and 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 simply enumerating the individual elements of the MATLAB arrays and enumerating all matrix algebra computations. However, this creates more 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 must 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 interfaces for each of the sub-models to expose to be able to read and write to the global state variables, and developing a scheme for managing simultaneous writing of shared state variables. The integration group chose define the global state variables as the union of all state variables shared by at least two sub-models rather than explicitly define a set of global state variables as implemented 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 disadvantage of this approach are that the total set of variables is less transparent and that it requires users to become familiar with all of the naming conventions of all of the sub-models in order to retrieve simulation data for analysis.

The integration group standardized the interfaces exposed by the individual sub-models by asking the other sub-model encoding groups to adhere to a common variable naming convention. This naming convention ensures, for example, that the copy numbers of each protein species are represented by the variables with the same names in each of the sub-models. The main feature of this convention is that it 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 whose names contained additional suffixes which indicate their identity.

The primary challenged faced by the integration group was how to handle the writing of state variables shared by multiple sub-models. The integration group explored several potential strategies to manage variable writing. First, they explored sequentially simulating the sub-models and updating the global state variables. The advantage of this approach is that it sub-models edit the global variables sequentially, avoiding other more complex strategies for merging variable changes. The disadvantage of this approaches is that within the same timestep the sub-models are simulated using different variable values. As a result, the simulation predictions are sensitive to the sub-model execution order.

Second, the integration group several more complex strategies in which within a single timestep all of the sub-models are simulated with the same variable values. These strategies included reducing the sub-model integration timestep such that sub-models would 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 has a high computational cost. The second strategy is simple to implement and computationally efficient for independent variables, but difficult to implement for sets of variables such as those which represent the chromosome protein occupancy with constraints on their joint values. 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 the SBML packages needed to simulate the integrated model including hierarchical model composition (*comp*), arrays, and flux balance constraints (*fbc*).

The second major challenged faced by the integration group was that no SBML simulator supports mutli-algorithm model composition. The group plans to overcome this limitation by adding support for multi-algorithm to iBioSim.

C. 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 combined model. Complete drafts are only available for a few of the simplest sub-models. The drafts of most of the more complex sub-models are 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 chromosome protein occupancy. 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.

D. Future steps

We hope to completely recode each of 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, BioUML, and iBioSim. To continue to progress toward this goal, several students are continuing to meet online to finish encoding, testing, and documenting sub-models. Several students and tutors also plan to participate in a second meeting in October 2015 which will be held in Salt Lake City, 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 the BioModels database, along with SED-ML tests, SBGN maps, and textual documentations. This will make the sub-models searchable, retrievable, and reusable by other scientists. We believe this will be a valuable community resource which will enable other researchers to build upon the *M. genitalium* sub-models.

Ultimately, SBML needs to be extended to support multi-algorithm modeling, particle-based state representation, arbitrary random number generation, and arrays to facilitate

whole-cell modeling. In addition, new SBML simulators must be developed which are capable of efficiently simulating large multi-algorithm models.

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. Most importantly, the summer school facilitated discussions among modelers, software tools 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 the 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 results in large, unmanageable XML files containing 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 tools 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 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 improved support for automatic layout of large maps. The existing automatic layout algorithms are unable to construct intuitive maps of many of the pathways described by the *M. genitalium* model. 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 while retaining meaningful labels.

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

B. Standard extensions

Taken together, expanded standards and supporting software tools are needed to facilitate accurate, reproducible whole-cell and other large-scale dynamical modeling. Here, 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 increasingly complex whole-cell models eventually including models of human cells. Together, the expanded package and software tools would enable teams of researchers to collaboratively build whole-cell models without needing to know the details of how every other sub-model is implemented.

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

Third, a new SBML package must be created to support reaction templates so that, for example, translation could be described using a single reaction template and arrays of the translation initiation rates of each mRNA and the codon-specific elongation rates. 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 description and quantitative parameter values, reaction templates would also make the connection between dynamical models and the experimental data from pathway/genome databases used to inform their parameter values more transparent. The new reaction templates could be expanded for backwards 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, as discussed above, SBGN must be expanded to support hybrid diagrams and the SBML viewers must be expanded to support more automatic layout algorithms, automatic model reduction, and contextual zooming. These

features will enable researchers to use SBGN to map whole-cell and other large models.

Together, these SBML, SBGN, and software 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 building more comprehensive and more accurate models, including of human cells. Ultimately, these new standards and software will enable whole-cell modeling to support rational biological design and the design of personalized medical therapies.

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 recode the *M. genitalium* whole-cell model into SBML. Additional summer schools and courses are needed to provide students 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 recoding 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 recode the *M. genitalium* model, and several of the students and tutors are participating in a second working meeting prior to 2015 COMBINE Forum at the University of Utah in Salt Lake City, USA. Ultimately, we hope to publish an SBML-recoded model to the BioModels database.

However, significant work remains to complete, test, integrate, simulate, and document the SBML-encoded version of the *M. genitalium* model. SBML currently cannot represent whole-cell models, no simulation software program exists to efficiently simulate an SBML-encoded multi-algorithm whole-cell model, and no graphical editor software program exists to construct, edit, or visualize SBML-encoded whole-cell models. SBML must be expanded to support multi-algorithm modeling, template reactions, particle-based state representation, and arrays. In addition, an efficient simulation software program must be developed. Going forward, new user-friendly model editors must be developed to enable modelers to more easily construct whole-cell models and connect whole-cell models to pathway/genome databases. New parameter estimation, modeling testing, and simulation visualization testing tools must also be developed to enable researchers to effectively use SBML-encoded whole-cell models. 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 transparently communicating whole-cell models and simulations, as well as general-purpose simulation software for reproducibly simulating whole-cell models. In turn, this requires expanding the whole-cell modeling field including training young researchers.

Acknowledgments

The school was supported by a grant from the Volkswagen Foundation to DW and FS. JRK was supported by a James S. McDonnell Foundation Postdoctoral Fellowship Award in Studying Complex Systems.

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] 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.

[3] 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.

[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] 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.

[7] 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.

[8] 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.

[9] C. Li, M. Donizelli, N. Rodriguez, H. Dharuri, L. Endler, V. Chelliah, L. Li, E. He, A. Henry, M. Stefan, J. Snoep, M. Hucka, N. Le Novere, and C. Laibe, "BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models," *BMC Systems Biology*, vol. 4, no. 1, p. 92, 2010.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

Table SI
2015 WHOLE-CELL MODELING SUMMER SCHOOL CONSORTIUM MEMBERS.

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	California Institute of Technology, USA
	Rafeal 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	KAIST, South Korea
Replication	Vincent Knight-Schrijver	Babraham Institute, UK
	Je-Hoon Song	KAIST, South Korea
	Jannis Uhlendorf	Humboldt University of Berlin, Germany
	Dagmar Waltemath	University of Rostock, Germany
	James Yurkovich	University of California, San Diego, USA
	Anna Zhukova	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
	Jesse 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
	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
Visualization and documentation	Christian Knüpfer	University of Jena, Germany
	Falk Schreiber	Monash University, Australia
	Tom Theile	University of Rostock, Germany
Modeling tutor	Jonathan R. Karr	Icahn School of Medicine at Mount Sinai, USA