Combining standards for tomorrow's models: Report of the 2015 whole-cell modeling summer school

Dagmar Waltemath[1], Falk Schreiber[2], Jonathan R. Karr[3], and Other People
[1]University of Rostock
[2]Faculty of IT, Monash University
[3]Department of Genetics & Genomic Sciences, Icahn School of Medicine at Mount Sinai

Computational modeling is an increasingly powerful and important tool for biological discovery, bioengineering, and medicine. Recently, researchers developed the first whole-cell model computational model which represents every individual gene. However, significant work remains to develop fully complete and accurate models of cells. We organized the 2015 whole-cell modeling summer school to teach students the latest cell modeling methods, as well as to bring the computational systems biology community together to assess the need and limitations of our current standards for whole-cell modeling. We found that a whole-cell modeling standard would accelerate the development of whole-cell models, and that more work is needed to expand our current standards to support whole-cell models.

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

I. Introduction

RESEARCH in the life sciences is increasingly computational, and computational modeling is a promising tool for biological discovery, bioengineering, and medicine. 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 entirely new strains of bacteria optimized for industrial tasks such as chemical synthesis, biofuel production, and waste decontamination, as well as to enable clinicians to design individualized medical therapies tailored to each patient's unique genome. Realizing this potential requires more comprehensive and accurate computational models which are capable of predicting cellular behavior from genotype, but also standardized ways to exchange model description, modeling parameters as well as model visualizations [4], [5], [6].

Recently, researchers at Stanford University developed the first whole-cell model of the gram-positive bacterium *Mycoplasma genitalium* [7]. The model represents the life cycle of a single cell including the copy number of each metabolite, RNA, and protein species, and it accounts for every known gene function. The model is comprised of multiple sub-models, each of which was implemented using different mathematical representations such as ordinary differential equations, flux balance analysis, and Boolean rules 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 was extensively documented. This has enabled other researchers to expand the model and use it for their own research, as well as use it as a teaching tool in systems biology courses in universities across the world.

Although MATLAB is commonly used by academic researchers, it is proprietary and expensive. In addition, because many of the biological details of the model are intertwined with the MATLAB code, significant domain expertise is required to understand, modify, and expand the model. A more transparent, standardized implementation is needed to enable more researchers to use the existing whole-cell model, as well as to contribute to whole-cell modeling. In turn, this would enable researchers to develop faster and more efficient whole-cell model simulators, more deeply explore whole-cell model predictions, and more rigorously evaluate whole-cell models; therby accelerating the whole-cell modeling field.

The Systems Biology Markup Language (SBML) [8] and the Cell Markup Language (CellML) [9] are the most commonly used systems biology modeling standards. Both languages can be used to develop a wide variety of models including ordinary differential equation, logical, and flux balance analysis models, and both have been used to build hundreds of models. However, neither language supports several of the features needed for whole-cell modeling including large, sparsely occupied state spaces and multi-algorithm simulations. In addition, the System Biology Graphical Notation (SBGN) [10] is the language to encode the visual representation of a model, but has currently also restrictions regarding the visualization of whole-cell models. Consequently, further work is needed to expand the SBML, CellML, and SBGN languages to support whole-cell models.

We organized the 2015 whole-cell modeling summer school to train students in whole-cell modeling to expand the whole-cell modeling field, as well as to initiate the further development of the SBML and SBGN languages to support whole-cell modeling. The goals of the school were three-fold: First and foremost, the goal of the course was to train young researchers how to build whole-cell models, how to develop sub-models of individual cellular pathways using different mathematical formalisms and encode them using SBML and graphically encode them using SBGN, and how to combine sub-models into a single model. The second goal of the course was to identify the features that must be added to the SBML and

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

SBGN languages to support whole-cell models. The third goal of the course was to initiate the recoding of the *M. genitalium* whole-cell model into SBML and SBGN.

Here, we summarize the educational and scientific outcomes of the summer school. The paper is structured as follows: Section II presents the format of the summer school, the progress towards SBML and SBGN models, and the limitations of SBML and SBGN. Section III presents the lessons we learned from the summer school and Section IV discusses future directions. We finish with a conclusion in Section V.

II. The 2015 whole-cell modeling summer school

A. *Format*

In preparation of the summer school the students were assigned to eight research groups, each led by a tutor invited prior to the summer school. There was a preparation phase before the meeting through Google hangout and code sharing through GitHub. During this phase, students got used to the MATLAB whole-cell model and started to think about their specific sub-model(s), especially the interfaces. The summer school week itself was structured in two invited talks, daily working sessions, a daily wrap-up (summary) presentation of each working group, a poster session, and a final presentation of results on the last day.

Two scientific invited talks were held at the summer school. The first speaker was Dr. Michael Hucka from the California Institute of Technology, USA. He provided an overview of the COMBINE initiative, COMBINE standards and tools supporting these standards. His talked focused on the formats most relevant to whole-cell modeling.

The second speaker was Dr. Jonathan Karr from the Icahn School of Medicine at Mount Sinai School, USA. He provided overviews of the whole-cell modeling field, of his own research toward developing and applying whole-cell models for scientific discovery and engineering, and of the *M. genitalium* whole-cell model which he and his colleagues recently developed. Dr. Karr concluded his talk by outlining several research projects which his own group is pursuing to expand the scope of whole-cell models and use whole-cell models to engineer faster growing bacteria.

The principal organisation of the course was as follows: We worked in eight teams of four to six students and one tutor. Each team focused on one part of the whole-cell model. The course included three additional “floating” tutors which were not assigned to specific teams. These tutors shared their expertise in systems modeling, model documentation, and SBGN with all of the teams.

The goal of each team was to provide a running module of their part of the model, together with the necessary inputs and outputs for the other groups. Lastly, one group, called “Integration”, coordinated the overall integration of all modules.

We chose this format deliberately to have mixed groups of standard developers (mostly the tutors) with modelers (mostly the students). At the same time we arranged students in groups so that their expertise matched the specific module, and so that the groups themselves consisted of heterogeneous scientists in terms of education. Lastly, another aspect for building the groups was that we did not want the participants to know each other before (to enhance the network experience for everybody) and internationality of the groups, whenever possible. The resulting groups thus were divergent in many aspects. However, the surrounding and the frame of the summer school led to a communicative environment. All students were open to learn new tools and methods, and they were willing to contribute with their own expertise to the overall task. Throughout the meeting, each participant presented at least once during a plenum session.

The schedule was completely free, and each group was left to organise themselves throughout the days. In the evenings, we had one plenum session to first discuss the day and then summarise the results of that day. The evening activities provided room to network, socialise and discuss about the work in a more informal setting, and specifically with participants from other groups.

At the final meeting, all tutors agreed that the goals of the summer school could not have been achieved without the interdisciplinary mix of expertises. The interdisciplinarity helped to see the task from different angles, and certainly it allowed us to try very different approaches to solving a problem (from brute force, to working with design thinking methods). In the opening to the summer school, we spoke of the effect of swarm intelligence, and this is what happened during the week of work.

The poster session enabled participants to present their own research. 27 students presented posters. Many of the students presented projects on whole-cell modeling or other computational systems biology projects.

B. *Educating young systems biologists*

One of the major intentions of this summer school was to educate young scientists. The whole-cell model is now one of the standard models in computational biology. It is therefore also important for young researchers to be informed about the model, its capabilities, the insights it gives and how it can be used and reused. Special focus was on making the participants familiar with cell biology, common modeling methods (such as ODE, FBA, and Boolean), model integration, and modeling standards (SBML, SED-ML, and SGBN).

We were glad to have Dr. Karr present his model at the summer school.e also taught students about the single submodels throughout the school. As a floating tutor, Dr. Karr was a member of the floating team. He visited the different groups and answered questions both on the modeling and on the biology.

The students were well-prepared for the course. Many had already attended the pre-course classes via Google Hangouts. Prior to the course, most of the students had already run the whole-cell model.

After the summer school, many students reported that they learned a lot about using open-source modeling software. In particular, many students reported increased understanding of SBML, and better awareness of reproducibility issues.

The course was also a great networking event for both the students and tutors. Students had opportunities to connect with other young computational systems biology researchers from around the world. The poster session provided students an opportunity to share their own work and get valuable feedback from each other, the tutors, and other scientists from the University of Rostock. Several of the tutors advertised open positions in the labs, as well as upcoming courses and meetings including the April 2016 whole-cell modeling summer school which Jonathan Karr, Luis Serrano, Maria Lluch-Senar, and Javier Carrera are organizing in Barcelona, Spain.

C. *Setting the goal*

The goal to encode the whole-cell model was ambitious from the beginning, but the expectations have been more than met. All participants dedicated their full time to working on the project, preparing initial results weeks beforehand, and even worked long hours. The achieved results are of high quality and the spirit at the summer school was very positive over the whole week. With 3 more days on the same project, we would have been able to test and then publish the 28 modules. The final state after one week is that all material is there, but the modules need to be tested and integrated. Several weeks after the summer school, the activity on the Git project is still high, and it can be expected that we will finish these two tasks before summer. The modules shall be published in BioModels Database by autumn 2015.

D. *Progress toward an SBML whole-cell model*

In addition to training young systems biology researchers, the course produced preliminary SBML encoded versions of each of the whole-cell models sub-models.

We taught standard formats using open standards (COMBINE standards) and open software (COMBINE-compliant software). COMBINE, the COmputational Modeling in BIology NEtwork [**?**] is the umbrella organisation for various standardisation initiatives, including SBML [8], CellML [9], SED-ML [11], and SBGN [10]. All formats are *de facto* standards. SBML represents networks in biology; CellML represents networks in physiology; SED-ML encodes simulation descriptions; and SBGN encodes the graphical representation of networks. Together, these standards allow for the encoding of virtual experiments in biology. Many so-called COMBINE-compliant software exist to run these experiments. Popular software, used during the summer school, are COPASI [12], BioUML [13], VANTED [14], and iBioSim [15].

We enhanced the exchange of information and developed 28 new modules in COMBINE formats that represent the majority of the whole-cell model, and which can be run in open-source software.

An overview of progress in the single modules is shown in Table 1. The sub-models are available open-source at http://github.com/dagwa and https://github.com/whole-cell-tutors/wholecell. The scientific community will benefit from these open-access, reusable versions of the sub-models that make up the model.

Further work is needed to integrate these sub-models into a single model. We hope to achieve this over the next several months.

E. *Limitations of SBML and SBGN for whole-cell modeling*

SBML and SBGN has never been applied before to such a large and complex model, and therefore using those standards for the whole-cell model provided interesting insights. Specific issues included the representation of large, combinatorial state spaces (such as different states of the binding sites) summing up to over one million in the SBML file, the lack of representation of arrays in core SBML, the generation of random numbers, and the sharing of variables across modules. In SBGN the size and (automatic) layout of some diagrams provided challenges and there arose a need to have mixed diagrams containing more than one SBGN language (SBGN has three sub-languages: Process description, Entity Relationship and Activity Flow).

The second reason for the delay in encoding is the lack of concepts to represent large arrays of data in SBML. We faced serious problems in representing, for example, variations in DNA binding sites. While MATLAB does have concepts for the representation of arrays and vectors, SBML lacks comprehensive ways of encoding these. Here, the discussions between modelers, standard developers and tool developers were particularly interesting, as arguments were provided from very different view points.

The positive effect on the standardisation community became immediately visible: The summer school offered a great opportunity for standard developers to receive feedback on the usability of COMBINE standards and associated software. This connection, between tool developers, standard developers and modelers is essential and yet usually missing. The single groups do have their own meetings and it happens that the communication between them is rather sparse. This summer school, however, offered a new channel of communication, through a very concrete task. The fact that modelers openly and directly pointed at lacks in existing standards, and that this criticism was converted into positive action on the developers site is for us the biggest achievement of the summer school. We hope that the summer school will inspire more events like this, where modelers and standard developers work together to solve a biological problem.

III. Lessons Learned

From the feedback that we received, we conclude that the summer school was well perceived. Most participants thought that the format was unusual, but gave them the opportunity to learn a lot. However, two participants mentioned that they had not been satisfied with the format, they said that they would have wished for a tighter schedule with more lectures. To meet this need, we introduced two break-out sessions with discussions on SBML-specific topics.

IV. Future directions

The goal is to construct SBML-encoded versions of all 28 sub-models and to publish them in the BioModels database. This will guarantee a long-time availability of the sub-models.

The participants of the summer school plan to continue working in their single groups even after the close of the school. Specifically, the groups plan to meet virtually via

Google Hangouts, to finalise the representation of the modules, the annotations, and the graphical maps.

We finished the last day with a long session on "How to move on", asking each group to make a plan on how to continue after the course. Here, it proved valuable to have all code available in a Git repository that is accessible to everyone. All participants of the course said that they would like to finalise the project. We hope to publish a scientific paper on the COMBINE-compliant model in autumn with all contributors as co-authors.

In addition, lessons learned from the summer school and weaknesses of COMBINE standards will be discussed during the next standardisation meeting (HARMONY) in April this year. It will therefore foster further standard development in systems biology.

V. Conclusion

Computational modeling is increasingly important for biological discovery, bioengineering, and medicine, and whole-cell models are expected to become a standard in the future. The whole-cell modelling summer school targeted this critical area and provided training to young researchers how to build whole-cell models (including how to develop sub-models, encode them using SBML and SBGN, and combine them into a single model), identified features that must be added to the SBML and SBGN languages to support whole-cell models, and initiated the recoding of the *M. genitalium* whole-cell model into SBML and SBGN.

The experiences of this summer school will be reported at this year's HARMONY and COMBINE meetings (as invited talks, opening the two major standardisation meetings for systems biology). We also aim to give feedback through our European networks, such as EraNet SysBio [16], CaSYM [17] and ISBE [18]. Jonathan Karr furthermore announced yet another summer school on whole-cell models, taking place in Barcelona in 2016, but with a focus on the theory behind modeling whole cells. We believe that this summer school set the path for a new series of meetings related to whole-cell modeling. Lastly, we hope that this summer school will become a prime example for modern educational events, showcasing how standard development can happen in close interaction with the end-users.

Acknowledgment

The course was supported by a grant from the Volkswagen Foundation to DW and FS.

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," *Proc Natl Acad Sci U S A*, 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," *J Bacteriol*, 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," *Nat Chem Biol*, 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," *Curr Opin Biotechnol*, vol. 28, pp. 111–115, 2014.

[5] J. R. Karr, K. Takahashi, and A. Funahashi, "The principles of whole-cell modeling," *Curr Opin Microbiol*, 2015.

[6] E. Klipp, W. Liebermeister, A. Helbig, A. Kowald, and J. Schaber, "Systems biology standards - the community speaks," *Nat Biotech*, vol. 25, no. 4, pp. 390–391, 04 2007.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

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

[14] 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 Syst Biol*, vol. 6, p. 139, 2012.

[15] J. T. Stevens and C. J. Myers, "Dynamic modeling of cellular populations within iBioSim," *ACS Synth Biol*, vol. 2, no. 5, pp. 223–229, 2013.

[16] ERASysBio consortium. ERA-NET for systems biology. [Online]. Available: http://www.erasysbio.net

[17] CaSYM consortium. Coordinating Action Systems Medicine: Implementation of Systems Medicine Across Europe. [Online]. Available: https://www.casym.eu

[18] O. Wolkenhauer, D. Fell, P. De Meyts, N. Blüthgen, H. Herzel, N. Le Novère, T. Höfer, K. Schürrle, and I. van Leeuwen, "SysBioMed report: advancing systems biology for medical applications," *IET Syst Biol*, vol. 3, no. 3, pp. 131–136, 2009.

Dagmar Waltemath

Falk Schreiber is a professor at Monash University's Faculty of IT and an adjunct professor at Martin Luther University Halle-Wittenberg's Institute of Computer Science. His interests are visual computing and visual analytics of biological data, analysis of structure and dynamics of biological networks, integration of multimodal data, standards for systems biology, as well as modelling and analysis of metabolism. Contact him at falk.schreiber@monash.edu.

Jonathan R. Karr received the Ph.D. degree in Biophysics and the M.S. degree in Medicine from Stanford University, Stanford, CA, USA in 2014 and the S.B. degrees in Physics and Brain & Cognitive Sciences from the Massachusetts Institute of Technology, Cambridge, MA, USA in 2006.

Dr. Karr is currently a Fellow at the Icahn School of Medicine at Mount Sinai in New York, NY, USA. His research focuses on the development of comprehensive whole-cell computational models and their applications to bioengineering and medicine. Contact him at jkarr@stanford.edu.