

Teaching Mathematical Modeling of Cellular Systems with the VCell MathModel

Jasia King^{1,§}, Kerbaï Saïd Eroumé^{1,§}, Roman Truckenmüller¹, Stefan Giselbrecht¹, Ann Cowan², Leslie M. Loew², Aurélie Carlier^{1,*}

¹MERLN Institute for Technology-Inspired Regenerative Medicine, Maastricht University, Universiteitssingel 40, 6229 ER, Maastricht, The Netherlands

²Richard D. Berlin Center for Cell Analysis and Modeling, University of Connecticut School of Medicine, Center for Cell Analysis and Modeling, UConn Health, 263 Farmington Avenue, Farmington, CT 06030-6406 USA

ABSTRACT Mathematical biology has emerged as a powerful approach to describe and understand biological systems. Here, we introduce an interactive teaching tool with a practical hands-on skill session plan to introduce students to the various components of a mathematical model with 4 different mathematical approaches (i.e., ordinary differential equations, partial differential equations, stochastic differential equations, and spatial stochastic differential equations) and their advantages and disadvantages. As such, we provide a didactic summary for instructors and students interested in using VCell MathModels for mathematical modeling; this work is also valuable for mathematics-savvy users who would like to exploit fully the capabilities of the VCell software.

KEY WORDS mathematical modeling; VCell software; e-learning; undergraduate biophysics; graduate biophysics

I. INTRODUCTION

Computational modeling has become a useful and increasingly popular tool in cell biology to describe biological events, predict outcomes of systems, inform experimental design, and much more. Modeling biological systems is used, among other reasons, to understand various cellular phenomena, such as the nuclear translocation of important transcription factors (1), signaling in healthy and diseased cells and tissues (2), and the effect of drugs on cell and tissue behavior (3) and on the entire organ system (4). Here, we illustrate how to use the Virtual Cell (VCell) MathModel interface to explore a cellular system with mathematical modeling (5,6). We introduce an interactive teaching tool with a practical hands-on skill session plan to introduce students to the various components of a mathematical model with 4 different mathematical approaches (i.e., ordinary differential equations [ODE], partial differential equations [PDE], stochastic processes, and spatial stochastic processes), along with their advantages and disadvantages.

“§” shared first author.

“*” corresponding author

Received: 3 April 2021

Accepted: 9 July 2021

Published: 3 December 2021

© 2022 Biophysical Society.

II. SCIENTIFIC AND PEDAGOGICAL BACKGROUND

A. Mathematical modeling

From introductory physics and chemistry classes, students often have come into contact with physical laws or mathematical models

describing the world around us (e.g., Newton's law of motion describing the relationship between the motion of a body and the forces acting upon it or basic kinetic laws that describe individual chemical reactions). However, students struggle with applying a similar formalism to highly complex systems, where multiple components interact and feed back on each other. As such, it is worthwhile for students, as a first step before embarking on establishing mathematical models for biological processes from scratch, to look under the hood of some existing models and see how the mathematical equations give rise to complex biological behavior. Moreover, by comparing various modeling formalisms and understanding their advantages and disadvantages, students can see that various models exist for the same biological process, but that depending on the posed research question, some formalisms and levels of abstraction are more appropriate to provide useful insights.

A theoretical lecture explaining the basic concepts of mathematical biology could start by introducing why biological systems need to be modeled and what a mathematical model is. By brainstorming, the students are likely able to identify some important challenges of biological systems, including their multiscale nature and their large number of components and interactions, as well as the nonlinearity of these interactions (7). To highlight the generic features of models (i.e., omission of details, abstraction, nonuniqueness), it can also be instructive to think about various types of models (e.g., physical models, maps and blueprints, chemical structures, *in vivo* models, *in vitro* models, mathematical models), which could lead to a discussion of the various components of a mathematical model: variables, parameters, geometry, and the processes or interactions that link the variables that are captured in the mathematical equations. The theoretical lecture could end by introducing different types of differential equations, what each represents, how they could be established (e.g., from mass action laws, Michaelis–Menten kinetics) and for which applications a particular formalism is most valid (see, e.g., the flowchart

in Fig 1). All this would naturally establish the value of computational tools to aid in the construction and simulation of mathematical models.

A number of references (7–12) provide a nice overview of mathematical modeling and can be used as (background) reading material.

B. An example inspired by nuclear transport

To illustrate the concepts discussed in the article, we use as an example nucleocytoplasmic transport of macromolecules, an important process to segregate the genome from the protein synthesis machinery and create spatio-temporal control over transcription and translation (13). Defects in the machinery involved in nuclear transport can lead to transport inefficiencies that affect signaling to and from the nucleus. Such disruptions in nuclear transport can lead to altered differentiation and development or impede other physiological processes resulting in (severe) diseases (14). In recent decades, much has been learned about the mechanism of transport of macromolecules, such as proteins and nucleic acids, between the nucleus and cytoplasm of the cells. In particular, molecules (and ions) less than 50 kDa can diffuse passively across the nuclear membrane via the nuclear pore complexes (NPCs), whereas large molecules require active transport to move through the NPCs (15, 16).

Briefly, large molecules (also termed “cargoes”) destined for transport contain nuclear localization sequences (NLS) for transport into the nucleus and nuclear export signals (NES) for transport out of the nucleus. The NLS and NES sequences are then recognized by members of the β -karyopherin family, which includes importins, exportins, and transportins. Importins (IMP) usher the NLS cargoes into the nucleus, exportins (EXP) deliver the NES cargoes out of the nucleus, and transportins have both import and export functionalities. Importantly, the nuclear cytoplasmic transport of macromolecules is also regulated by the gradient of the ras-related nuclear (Ran) G protein (13). The GTP-bound form of Ran is around 200 times more abundant in the nucleus, and its interac-

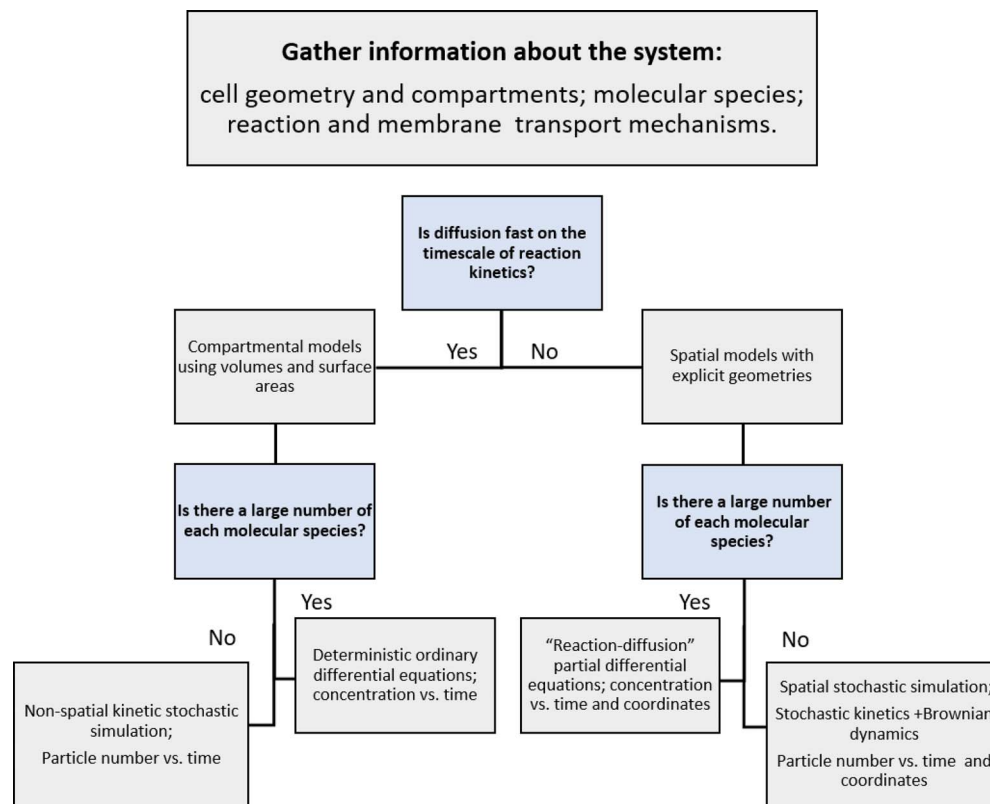


Fig 1. Flowchart to select the most appropriate equation-based modeling approach.

tion with the IMP-NLS cargo complex leads to the nuclear release of the cargo. Subsequently, the Ran-GTP-importin complex returns to the cytoplasm where Ran-GTP is hydrolyzed to Ran-GDP by SUMOylated Ran GTPase-activating protein 1 (RanGAP1) together with Ran-binding protein 1 (RanBP1) and Ran-binding protein 2 (RanBP2) (13). Nuclear export is characterized by the shuttling of the Cargo-EXP-NES-Ran-GTP complex to the cytoplasm. Release of the cargo involves transformation of Ran from the GTP- to the GDP-bound state. Ran-GDP is then recycled back to the nucleus by nuclear transport factor 2 (NTF2) (13). For more information and details, we refer the reader to an article by Kalita *et al.* (13).

Despite these recent insights, many open questions remain (13). For example, it is not well understood how the steep Ran-GTP–Ran-GDP gradient is maintained, although it is clear that adenosine triphosphate hydrolysis is required as the energy source. Considering that the dynamics of all these components are interdependent, a systems-level integration of

all the independent transport steps by a computational approach is warranted (1, 17).

The demonstration model used in this article is a simplified model of Ran nuclear transport, implemented in VCell, version 7.2 (MultiApp tutorial) (18). Note that in this simplified example, we make no attempt to model the maintenance of the Ran-GTP–Ran-GDP gradient. Only the unbound cytoplasmic Ran, unbound cargo macromolecule, and nuclear and cytoplasmic versions of the bound Ran-cargo complexes are modeled; transport across the nuclear membrane is passive, and dissociation of the Ran-cargo complex is modeled as a reversible mass action process (18). The model can be found online and accessed in the VCell BioModel Database (*BioModels > Biological models > Tutorials > Tutorial_MultiApp*). This tutorial model has all 4 basic types of modeling approaches, including ordinary and partial differential equations, as well as nonspatial stochastic and spatial stochastic equations. As such, these examples can be used to (a) understand and compare various types of

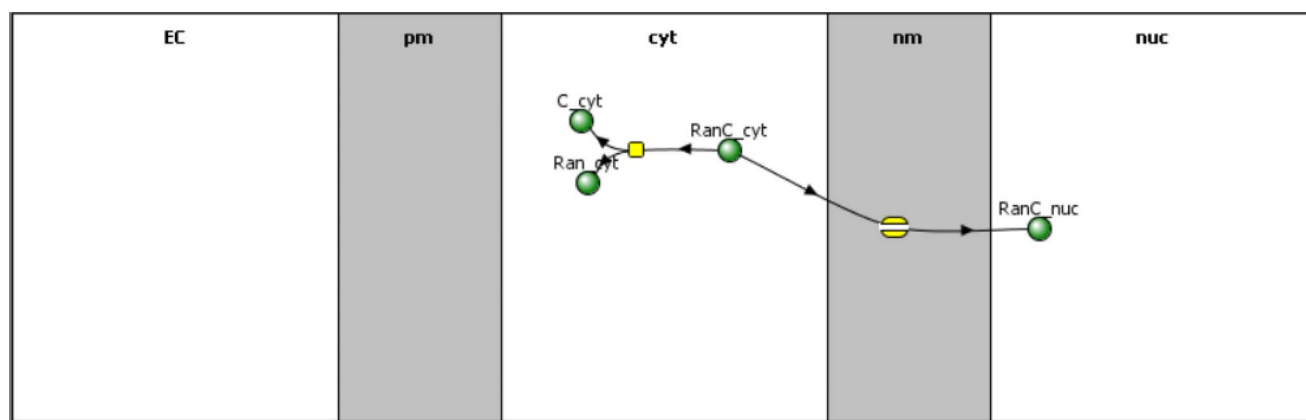


Fig 2. Physiology diagram of the simplified model of Ran nuclear transport. EC (extracellular), cyt (cytosol), and nuc (nucleus) are the 3 compartments (or subdomains) separated by the plasma (pm) and nuclear membranes (nm). C is the cargo molecule and *RanC* refers to the complex Ran GTPase and the cargo molecule.

mathematical equations; (b) illustrate that, depending on the research question, various mathematical descriptions can exist of the same biological process; and (c) explain the biological predictions and understanding that can arise from the computational models.

The above, simplified model of Ran nuclear transport can serve to motivate students to explore a full model for nucleocytoplasmic transport and maintenance of the Ran-GTP–Ran-GDP gradient, which can also be accessed in the VCell BioModel Database (BioModel user *les*, model name *Smith et al 2002 Systems Analysis of Ran Transport*) and is described in detail by Smith *et al.* (1).

III. VCELL

A. Starting with a BioModel

VCell is an open source, freely available computational biology platform developed at the University of Connecticut, Farmington, to model and simulate cellular signaling dynamics (5,6). VCell offers a range of solvers to implement deterministic, stochastic, or hybrid deterministic–stochastic models (19, 20).

The VCell platform provides an easy-to-use interface for students with limited programming experience to navigate model building under the *BioModel* option. In the BioModel graphical user interface environment, separately organized steps define the model and create simulations based on the model; the steps

include: defining *Physiology*, creating *Applications*, and running *Simulations*. *Physiology* is a conceptual (graphical) representation of the biological system in terms of cellular structures (e.g., cytosol, nucleus, plasma membrane), species, and biochemical reactions occurring within or outside the cell. The user can build the model reactions by dragging and dropping reaction arrows that connect the species to define reactions (Fig 2). Each element of the model can be associated with annotated biological information. *Applications* define the quantitative conditions needed to simulate a virtual experiment, such as the type of physical approximation to be used (e.g., stochastic, deterministic) initial conditions, diffusion coefficients, geometries (i.e., well-mixed or spatial, 1-, 2-, or 3-dimensional [1D, 2D, or 3D]). A single *BioModel* may have numerous *Applications*, in which the physical approximations, geometries, parameters, or a combination of factors have been changed. Each application, in turn can have multiple *Simulations*, in which different parameters, numerical methods, or conditions are used.

Importantly, the student does not need to write lines of code to describe the system of reactions and solve it, but only specifies the rate expressions and the parameter values. For each application, VCell automatically converts the graphical, biological description into a corresponding mathematical system of ordinary or partial differential equations, or both, or

stochastic probabilities on behalf of the user. This automatic conversion significantly reduces the possibility of scripting errors. Importantly, the mathematical representation that is automatically generated from a BioModel application, coded in the VCell Math Description Language (VCMDL), can be viewed, edited, and run independently as a *MathModel*. In the MathModel interface, the students can directly see the underlying equations, variables, and functions and use them to interpret the simulation results. Furthermore, it is possible to include additional abstractions within the MathModel interface that cannot be included directly in the BioModel interface.

In the next section, we describe how a MathModel can be created from a BioModel and explain the VCMDL syntax with snippets of code. Understanding the VCMDL syntax will help instructors and students understand the mathematical equations and relate them to the simulation outcomes. The entire MathModel VCMDL script can be found in the Supplemental Material.

B. MathModel from BioModel

As an introductory project, students could start from a BioModel and convert it into a MathModel, which would allow students to look under the hood of some existing models and see how the mathematical equations give rise to complex biological behavior. In a more advanced project, students could start implementing their mathematical equations from scratch (see below), although in practice it is easier to start from a BioModel and convert it into a MathModel. In VCell, a MathModel can be generated from an application of a BioModel with *Open BioModel > Applications > Simulations > Generated Math > Create MathModel*.

Within the BioModel, the student may select which application in the BioModel will be copied as a new MathModel. Note, each application has a separate MathModel associated with it! The new MathModel is editable as a VCMDL document and is completely independent from the original BioModel application. Any changes in the new MathModel will not affect the original BioModel. In the VCMDL

Editor you can find the MathModel code. In the Supplemental Material, the most common features of the VCMDL syntax are explained in detail (Fig 3).

C. Creating a de novo MathModel

To create a new MathModel de novo, click *File > New > MathModel*. Four options are provided in the next level of the popup menu when starting a new MathModel:

- *Non-Spatial* creates a nonspatial (with ordinary differential equations) MathModel.
- *Spatial from database geometry* creates a spatial MathModel with the spatial geometry selected from VCell geometries stored in the VCell database (any geometry that belongs to the current user or that is shared by other users).
- *Spatial from new geometry* creates a spatial MathModel from a new geometry. In a new dialog, the student is prompted to create a new geometry. The geometry can be analytical (1D, 2D, or 3D); image-based (from the VCell database or imported from an image file and processed); mesh-based (from a stereolithography [STL] CAD file); copied from a BioModel application or another MathModel or a saved Geometry from the VCell database; or created from scratch.
- *From BioModel* was discussed in section III.B.

Once a geometry is selected, VCell opens to a new page in the VCMDL Editor:

```
MathDescription {
  CompartmentSubDomain subdomain0 {
    BoundaryXm Value
    BoundaryXp Value
    BoundaryYm Value
    BoundaryYp Value
    BoundaryZm Value
    BoundaryZp Value
  }
} <end>
```

Now the student can initialize the constants to be called in the equations, define the membrane and compartment variables, add the equations in the function section, and finally, define all the discontinuities as jump conditions. To this end, the student can draw

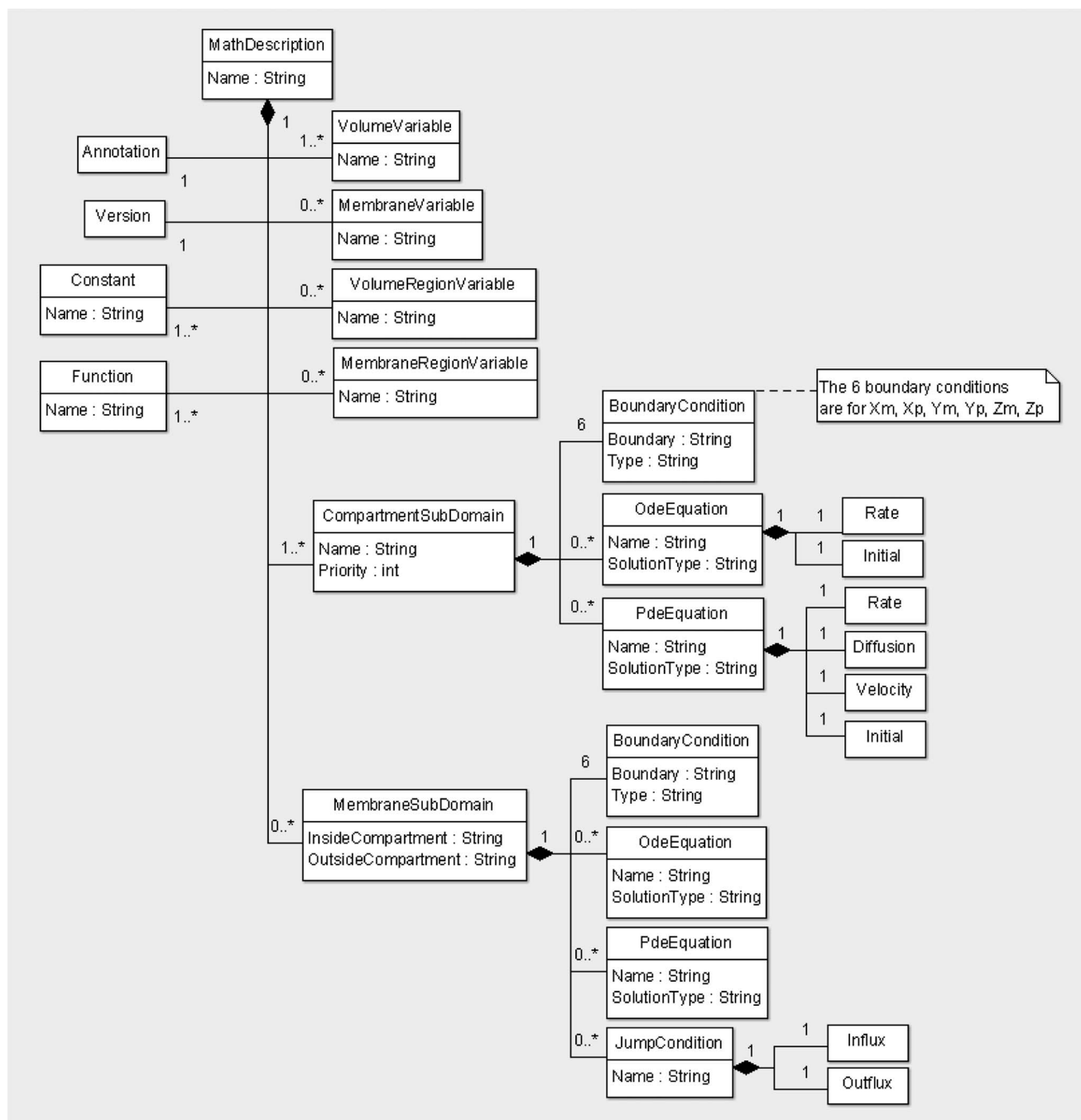


Fig 3. Unified modeling language diagram of a MathDescription. Each end of an association indicates the number of objects participating in the relationship. * Indicates “or more.”

inspiration from the snippets of code described in the previous section.

D. Troubleshooting

After pressing *apply changes* in the VCMDL Editor, warnings or errors may pop up. Warnings indicate potential problems that VCell has identified in the model but do not prevent the

student from saving the file, generating the math, and running the model. Errors must be corrected before a MathModel can be saved to the database and run. (A MathModel can always be saved as a text file on the student’s storage device).

To further troubleshoot, it might be interesting to compare with previous versions of the

code, which can be facilitated by copy-pasting the generated math description into word-processor files (DOC or TXT) so that an automatic comparison tool can be used to detect differences. Alternatively, VCell also provides the *File > Compare with Saved* tool for comparing VCell models in the central VCell database.

Additional information can also be found in the Help menu bar option, which links to the VCell website, the VCell Open Discussion forum, and VCell Support contact details.

To help troubleshooting, it might be convenient to grant access to specific students or instructors with *File > Permissions*. Three options allow sharing of models: *Public*, *Private*, or *Grant Access to Specific Users*. By default, a student's model is *Private*, which means that only the student has permission to view the model. To make a model public to everyone with VCell, select *Public*. Other users can save a copy of the current user's model as their own and then have complete access to their own copy. No modifications can be made to the user's original copy. If the user wants to share their model with specific VCell users, it can be accomplished by selecting *Grant Access to Specific Users*. These users will be able to access the shared model the same way they access any other public model. To all other users, this model would be invisible (like a private model). Sharing models is particularly attractive because instructors can help students and use the shared, finalized code to grade a project. When the *VCell Support* checkbox is checked, the VCell Support team gets permission to view the user's model if the user is having any difficulties with the model.

IV. SKILL SESSION PLAN

The skill session presented here has the following learning objectives: (a) students are able to recognize the various components of a mathematical model (i.e., equations, parameter values, variables, geometry, initial conditions, boundary conditions); (b) students are able to recognize 4 different mathematical approaches (i.e., ordinary differential equations, partial differential equations, stochastic processes, and spatial stochastic processes); (c) students under-

stand the advantage and disadvantages of every approach; (d) students are able to select the proper approach for a particular biological application; and (e) students are able to interpret the mathematical modeling results and link them to the modeled biological process—here, Ran nucleocytoplasmic transport. This practical hands-on lesson should be preceded by an introduction to mathematical modeling and differential equations (see section II).

To prepare for the class, the students are asked to install the VCell software and to read through the quick start guide (21). The students start by opening the *Tutorial_MultiApp* and studying the BioModels. The students can check by hand whether the reaction units are consistent. The students can also play with the kinetic type of the reaction and see how this would alter the mathematical formulation. The instructor can guide the self-directed learning of the students by asking some of the questions listed in Table 1.

Next, the students look at the MathModel of the different applications and derive the corresponding differential equations. During this part of the skill session, the instructor should emphasize the following concepts: (a) the differences between spatial and nonspatial models: nonspatial models have only well-mixed compartments, no diffusion coefficients, and diffusive terms; for spatial models, other elements are found back in the MathModel: *BoundaryXm*, *PDEequation*, *Diffusion*; (b) the differences between stochastic and nonstochastic: the stochastic equations have probabilities and count the number of molecules; the nonstochastic equations approximate the number of molecules by a concentration that can vary as a function of space, time, or both.

After the more theoretical exploration, the students should run the different simulations and compare the results. Students can play with different settings of initial conditions, parameter values (i.e., spatially dependent diffusion coefficients), and number of molecules. Playing with the settings of the simulations allows students to explore hands-on the differences between spatial and nonspatial simulations, as well as the differences between stochastic and non-

Table 1. Potential questions for the VCell skill session.

Question	Answer
What are the variables in the nuclear transport model?	$RanC_{cyt}$, $RanC_{nuc}$, C_{Cyt} , Ran_{cyt} , time (independent variable)
What are the parameters in the nuclear transport model?	Kf , kf , kr , initial conditions
What are the initial conditions in the nuclear transport model?	All 0 except $RanC_{nuc} = 4.49 \times 10^{-4} \mu\text{M}$
What are the diffusion coefficients of the variables in the 3D PDE model?	All $10 \mu\text{m}^2/\text{s}$
What is the boundary condition at the plasma membrane for the 3D PDE model?	No flux boundary condition (nothing can leave the cell)
What is the difference between the initialization of the stochastic and nonstochastic spatial application?	Stochastic: number of molecules; nonstochastic: concentration (continuum approximation)
Are these open or closed models?	Closed; nothing leaves the system, no production or degradation
Are these dynamic or static models?	Dynamic, the variables change as a function of time

stochastic simulations. In particular, to illustrate the difference between spatial and nonspatial simulations, the students can vary the diffusion coefficients and see how this influences the distribution of variables in the model. Here, the instructor can highlight that, because of local spatial variations, the average concentration is not a good measure for system behavior and that spatial plots provide important insight. Increasing the diffusion coefficients will approximate a well-mixed system, and students can explore whether the PDE results indeed match the ODE results. To illustrate the difference between stochastic and nonstochastic simulations, the students can play with the number of molecules. Here, the instructor can highlight that, by increasing the number of molecules, the continuum approximation becomes more valid, and as such, the results should approach those of the ODE model. The exploration of the differences between spatial and nonspatial and stochastic and nonstochastic simulations can be done in small student groups. Each group then explains their findings to the front of the classroom. The class is asked whether they agree or not with the group's interpretation, followed, if necessary, by a discussion. This setting not only allows students to apply their theoretic knowledge from, for example, a preceding lecture, but also to all be engaged, including those who are otherwise more timid or reluctant to participate.

The entire skill session should last approximately 2 h. By combining a theoretical lecture with a hands-on skill session, students should

be able to achieve a good understanding of the basics of mathematical modeling.

A. Spatial versus non-spatial simulations

To illustrate how some of the questions for the skill session can be discussed with VCell simulation results, Figure 4A–C shows the 3D PDE simulation results of $RanC_{cyt}$ for 3 diffusion coefficients of $RanC_{cyt}$. From Figure 4A, it can be concluded that higher gradients of $RanC$ exist in the cytosol when the diffusion coefficient is lower and can be confirmed by looking at the postprocessing tab. When the diffusion coefficient is high, approaching a well-mixed system, the maximal and average concentration almost overlap (Fig 4B), whereas for low diffusion coefficients, large differences persists between the maximal and average concentrations (Fig 4C). Instructively, when the diffusion coefficient is high (i.e., $1000 \mu\text{m}^2/\text{s}$), the $RanC$ concentration in the cytosol is very similar for the ODE ($8.936 \times 10^{-5} \mu\text{M}$) and PDE models ($8.973 \times 10^{-5} \mu\text{M}$) (see Fig 4B).

B. Stochastic versus nonstochastic simulations

Figure 5 shows the temporal plots of the $RanC_{cyt}$ concentration for various runs of 1002 molecules of initial nuclear $RanC$ (corresponding to the initial ODE concentration), as well as the deterministic ODE solution. From these plots and the insert, it can be concluded that stochastic solutions fluctuate around the deterministic solution and that every stochastic run gives rise to a different temporal plot.

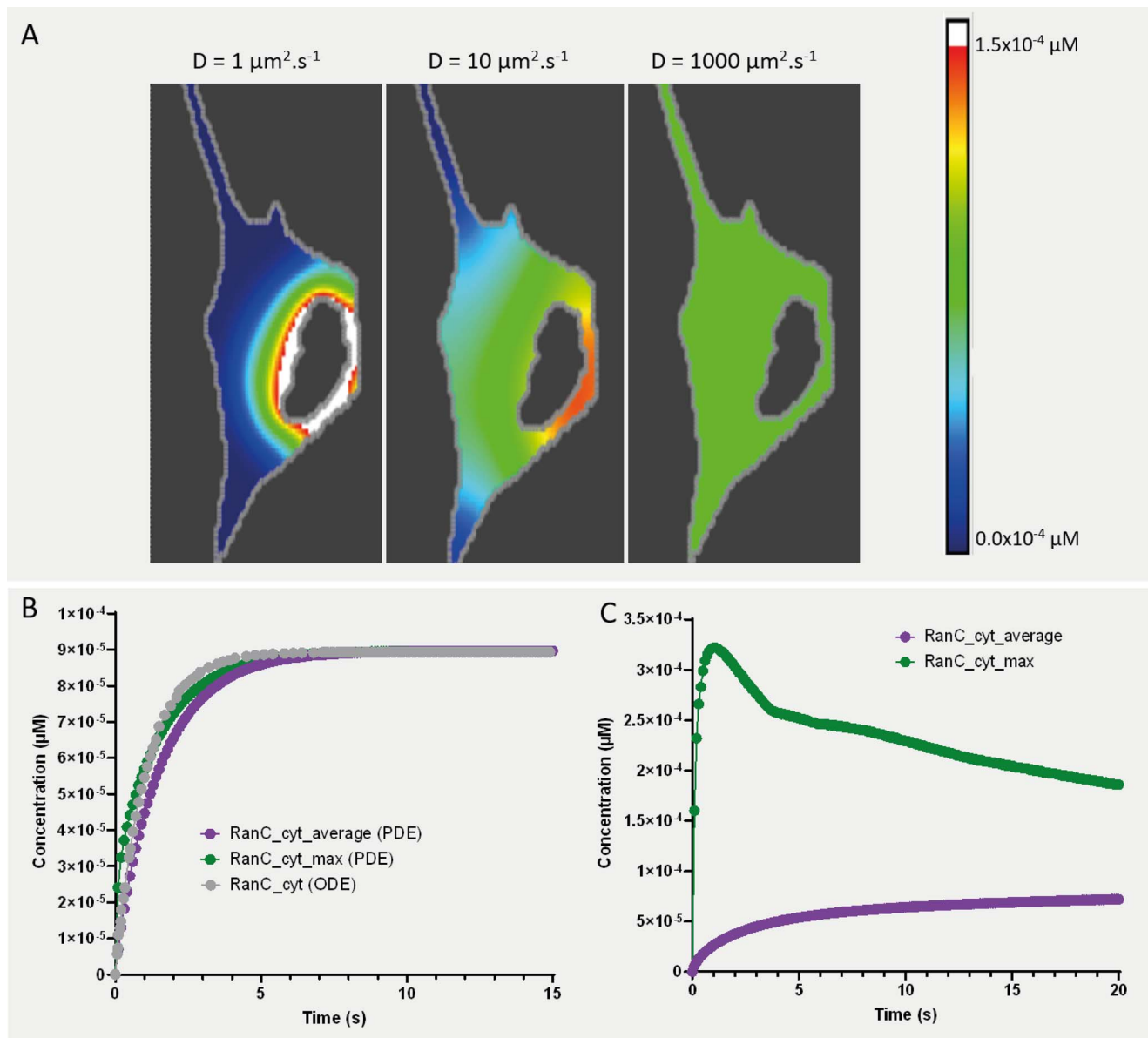


Fig 4. Comparison of the spatial and nonspatial model of Ran nuclear transport. (A) Spatial plots of the 3D PDE simulation results of *RanC_cyt* for 3 diffusion coefficients of *RanC_cyt*. The plots are taken at 9 s and $z\text{-stack}[14] = 10.4 \mu\text{m}$. The regions that are colored white have a concentration $> 1.5 \times 10^{-4} \mu\text{M}$. (B) Temporal plots of the average and maximal *RanC_cyt* concentration for diffusion coefficient $D = 1000 \mu\text{m}^2/\text{s}$ and of the *RanC_cyt* concentration in the ODE model. (C) Temporal plots of the average and maximal *RanC_cyt* concentration for $D = 1 \mu\text{m}^2/\text{s}$.

C. Spatial stochastic simulations

Similar to the deterministic application, Figure 6 shows the 3D spatial stochastic simulation results of *RanC_cyt* for 3 diffusion coefficients of *RanC_cyt*. Because of the limited number of particles (higher numbers are possible in VCell but computationally very intensive and, as such, not practical for a skills lab), it is difficult to conclude whether any cytosolic *RanC* gradients exist and how this

differs for the 3 diffusion coefficients. Note, as well, that because of the stochastic nature, ideally every simulation should be run multiple times. From Figure 6B it can be concluded that *RanC* in the cytosol reaches a steady state more quickly when its cytosolic diffusion coefficient is higher. Indeed, when the *RanC* molecules can diffuse faster in the cytosol, they do not accumulate near the nucleus after export and, as such, increase the kinetic driving force for

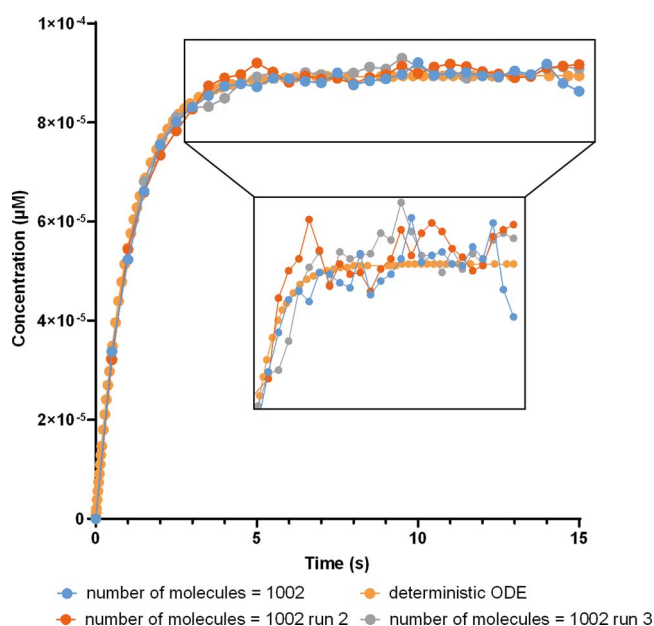


Fig 5. Comparison of the stochastic and nonstochastic model of Ran nuclear transport. Temporal plots of the $RanC_{cyt}$ concentration for various runs with 1002 molecules of initial nuclear $RanC$ as well as the ODE solution (with $4.5 \times 10^{-4} \mu\text{M}$ initial nuclear $RanC$). The inset shows that every stochastic run gives rise to a different temporal plot.

nuclear export. These conclusions are similar to the spatial deterministic results of Figure 4.

D. Biophysical interpretation

Next to understanding the various modeling formalisms and their limitations, the simplified

model of Ran nuclear transport can also be used to explore the underlying biology (and whether it is adequately captured by the mathematical model). As described in the previous sections, a higher diffusion coefficient of $RanC$ in the cytosol results not only in faster $RanC$ nuclear export (Fig 6B), but also in reduced cytosolic gradients (Fig 4A). The simplified model could also be used to explore whether the export rate is dependent on the initial concentration of nuclear $RanC$. The results shown in Figure 7 indicate that this is the case because of a higher initial concentration gradient that drives the transport in the simplified model.

In all our simulations with this passive transport model, the steady state distribution of $RanC$ is uniform between the nucleus and cytosol, in contradiction to experimental data, which shows that stable gradients exist (1). This result indicates that energy-dependent mechanisms are at play and will motivate the student to dig deeper into the details of this important system. Indeed, it is important to emphasize that the presented mathematical model of Ran transport is just a first step and that there are many additional components that will be needed to model the actual biology properly. For example, it does not distinguish between Ran-GTP and Ran-GDP. It also does not account for the various proteins that can hydrolyze Ran-GTP to Ran-GDP (e.g., RanGAP1, RanBP1, RanBP2) or

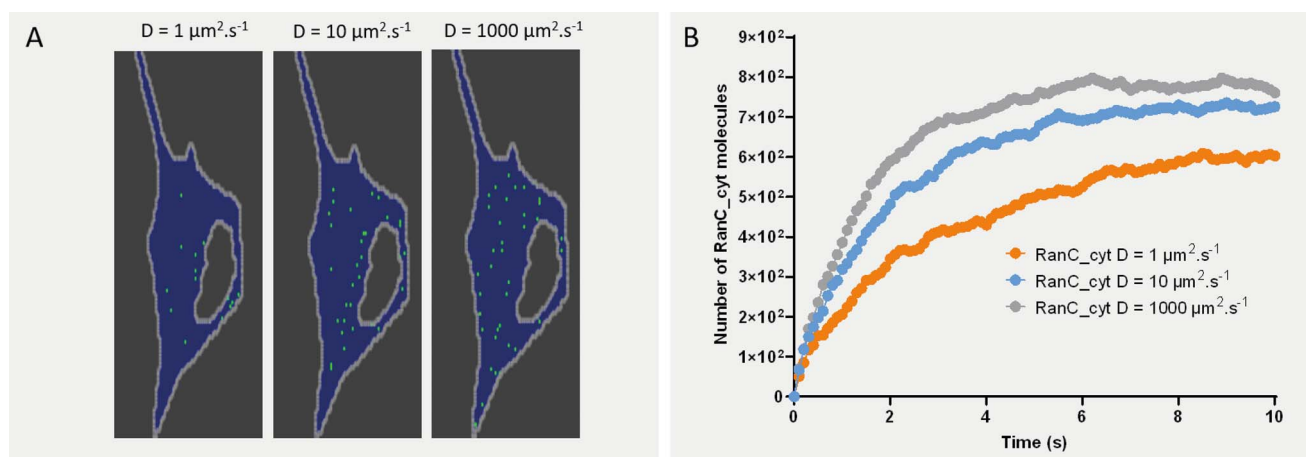


Fig 6. Comparison of spatial stochastic simulations of Ran nuclear transport. (A) Spatial plots of the 3D spatial stochastic simulation results of $RanC_{cyt}$ for 3 diffusion coefficients of $RanC_{cyt}$. The plots are taken at 9 s and $z\text{-stack}[14] = 10.4 \mu\text{m}$. (B) Temporal plots of the number of $RanC_{cyt}$ molecules for 3 diffusion coefficients of $RanC_{cyt}$ $D = 1, 10, \text{ and } 1000 \mu\text{m}^2/\text{s}$.

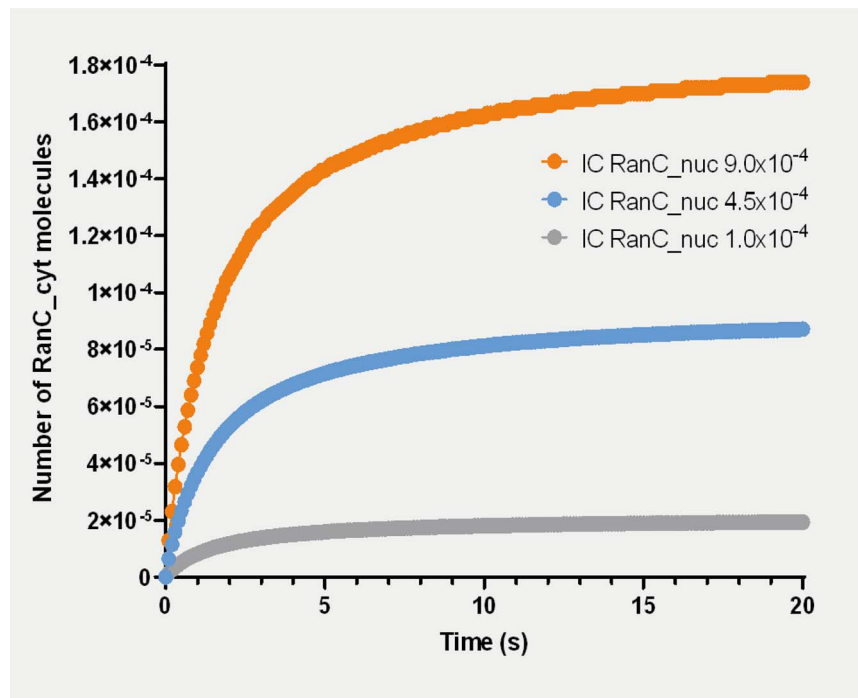


Fig 7. Comparison of Ran transport for different initial *RanC* nuclear concentrations. The figure shows the average *RanC* concentration in the cytosol as a function of time for different initial conditions of the *RanC* nuclear concentration. These results are calculated with the standard settings of the spatial deterministic model (3D PDE).

transport Ran-GTP back into the nucleus (e.g., NTF2). Moreover, transport across the nuclear membrane is passive, whereas it has been shown that passive diffusion makes little contribution to the net flux of Ran through the NPC (1).

In a follow-up session, the students could extend the nuclear transport model with additional variables or explore the more extended Ran transport model in the VCell database (BioModels user *les*, model name *Smith et al 2002 Systems Analysis of Ran Transport*). Although a detailed description of this extended model falls out of the scope of this publication, the model predictions of this extended Ran transport model have shown, among other things, that (a) Ran transport is robust for fluctuations in concentrations of transport factors; (b) the free Ran-GTP concentration is orders of magnitude lower than the dissociation constant for Ran binding to importins, indicating that additional factors may play a role for cargo unloading from importins; (c) cytosolic Ran-GTP gradients exist because of RanGAP activity near the nucleus, potentially providing positional information on the loca-

tion of the nucleus (1). Importantly, these quantitative predictions allow designing dedicated wet lab experiments to further investigate and elucidate the complex processes involved in the nucleocytoplasmic transport of macromolecules.

V. CONCLUSION

In this study, we have described how mathematical modeling or reaction kinetics and diffusion can be introduced to students with the help of VCell, through an existing tutorial to illustrate some basic concepts of mathematical biology. We emphasized the VCell MathModel interface because it allows instructors and students to see and modify the underlying mathematics. The BioModel and MathModel interfaces each have their own advantages and disadvantages. In the MathModel interface, the students can directly (a) see the underlying equations, variables, and functions and (b) write their own customized code in VCMDL, bypassing the graphical user interface, including, for example, the definition of new variables to store values temporarily or

scale them. As such, by using a well-defined syntax instead of drawing a physiology diagram, one has more flexibility in the handling and modification of the computational model. In the BioModel interface, one can (a) perform parameter estimation; (b) from a single physiology, rapidly use different types of solvers to identify how solutions differ among simulations strategies; (c) reduce scripting errors from the automatic conversion; (d) reduce errors from checking for unit and equation consistency; and (e) readily communicate the elements of the model in a user-friendly visual format.

We have presented an interactive lesson plan to develop mathematical biology skills and intuition with a simple biological model of passive nuclear transport. The framework can serve as a stepping stone for more complex simulations of other biological processes, where students can move beyond inspecting the mathematical equations toward extending and implementing mathematical biology models from scratch. In summary, this paper provided a didactic summary for instructors and students interested in using VCell Math-Models for mathematical modeling or for (seasoned) mathematics-savvy users that would like to exploit fully the capabilities of the VCell software.

SUPPLEMENTAL MATERIAL

Supplemental material for the MathModel VCMDL script and the most common features of the VCMDL syntax are available at: <https://doi.org/10.35459/tp.2021.000198.s1>.

AUTHOR CONTRIBUTIONS

JK and SE conceptualized, visualized, and wrote the original draft. RT and SG wrote, reviewed, and edited; provided supervision; and acquired funding. LL and ACo conceptualized and wrote, reviewed, and edited the manuscript. ACA conceptualized and wrote the original draft, reviewed and edited the manuscript, provided supervision, and acquired funding.

ACKNOWLEDGMENTS

We kindly acknowledge the Dutch province of Limburg in the LINK (FCL67723—SAS-2014-00837 and SAS-2018-02477) (“Limburg INvesteert in haar Kenniseconomie”) knowledge economy project, a VENI grant (15075) from the Dutch Science Foundation (NWO), and support by the partners of RegMed XB, powered by Health~Holland, Top Sector Life Sciences & Health. The research was also supported by National Institute for

General Medical Sciences of the National Institutes of Health under grant R24 GM137787. The funders did not have any role in the design and conduct of the study; in the collection, analysis, and interpretation of the data; and in the preparation, review, or approval of the manuscript. We also thank Boris Slepchenko for discussion and suggestions that greatly improved the manuscript.

REFERENCES

- Smith, A. E., B. M. Slepchenko, J. C. Schaff, L. M. Loew, and I. G. Macara. 2002. Systems analysis of ran transport. *Science* 295:488–491.
- Ji, Z., K. Yan, W. Li, H. Hu, and X. Zhu. 2017. Mathematical and computational modeling in complex biological systems. *Biomed Res Int* 2017:5958321. <https://doi.org/10.1155/2017/5958321>.
- Derendorf, H., and B. Meibohm. 1999. Modeling of pharmacokinetic/ pharmacodynamic (PK/PD) relationships: concepts and perspectives. *Pharm Res* 16:176–185.
- Holzütter, H. G., D. Drasdo, T. Preusser, J. Lippert, and A. M. Henney. 2012. The virtual liver: a multidisciplinary, multilevel challenge for systems biology. *Wires Syst Biol Med* 4:221–235.
- Slepchenko, B. M., and L. M. Loew. Use of virtual cell in studies of cellular dynamics. In *International Review of Cell and Molecular Biology*. Elsevier/North Holland, Amsterdam, pp. 1–56.
- Center for Cell Analysis and Modeling. VCell: a modeling environment for the simulation of cellular events. University of Connecticut Health Center, Farmington, CT. Accessed 4 August 2021. <https://vcell.org/>.
- Voit, E. O., Z. Qi, and G. W. Miller. 2008. Steps of modeling complex biological systems. *Pharmacopsychiatry* 41: S78–S84. <https://doi.org/10.1055/s-2008-1080911>.
- Daun, S., J. Rubin, Y. Vodovotz, and G. Clermont. 2008. Equation-based models of dynamic biological systems. *J Crit Care* 23:585–594.
- Xiong, F., and S. G. Megason. 2015. Abstracting the principles of development using imaging and modeling. *Integr Biol* 7:633–642.
- Scholma, J., S. Schivo, R. A. Urquidí Camacho, J. van de Pol, M. Karperien, and J. N. Post. 2014. Biological networks 101: computational modeling for molecular biologists. *Gene* 533:379–384.
- Brodland, G. W. 2015. How computational models can help unlock biological systems. *Semin Cell Dev Biol* 47–48:62–73.
- Mogilner, A., R. Wollman, and W. F. Marshall. 2006. Quantitative modeling in cell biology: what is it good for? *Dev Cell* 11:279–287.
- Kalita, J., L. E. Kapinos, and R. Y. H. Lim. 2021. On the asymmetric partitioning of nucleocytoplasmic transport—recent insights and open questions. *J Cell Sci* 134:jcs240382. <https://doi.org/10.1242/jcs.240382>.
- Poon, I. K. H., and D. A. Jans. 2005. Regulation of nuclear transport: central role in development and transformation? *Traffic* 6:173–186.
- Beck, M., V. Lúí, F. Förster, W. Baumeister, and O. Medalia. 2007. Snapshots of nuclear pore complexes in action captured by cryo-electron tomography. *Nature* 449:611–615.
- Eibauer, M., M. Pellanda, Y. Turgay, A. Dubrovsky, A. Wild, and O. Medalia. 2015. Structure and gating of the nuclear pore complex. *Nat Commun* 6:1–9.
- Görllich, D., M. J. Seewald, and K. Ribbeck. 2003. Characterization of Ran-driven cargo transport and the RanGTPase system by kinetic measurements and computer simulation. *EMBO J* 22:1088–1100.
- Center for Cell Analysis and Modeling, University of Connecticut Health Center. VCell tutorial: multiple application of a nuclear transport (neuroblastoma stack for tutorial), version 7.2. Accessed 4 August 2021. https://vcell.org/webstart/VCell_Tutorials/MultiAppTransport_7.2.pdf.
- Schaff, J., C. C. Fink, B. Slepchenko, J. H. Carson, and L. M. Loew. 1997. A general computational framework for modeling cellular structure and function. *Biophys J* 73:1135–1146.
- Cowan, A. E., I. I. Moraru, J. C. Schaff, B. M. Slepchenko, and L. M. Loew. 2012. Spatial modeling of cell signaling networks. In *Methods in Cell Biology*. Academic Press, Cambridge, MA, pp. 195–221.
- Center for Cell Analysis and Modeling, University of Connecticut Health Center. VCell quickstart. Accessed 4 August 2021. https://vcell.org/webstart/VCell_Tutorials/VCell_Quickstart_6.pdf.