Biophysical *Journal*
Computational Tool

Biophysical Society

Compartmental and Spatial Rule-Based Modeling with *Virtual Cell*

Michael L. Blinov,[1,*] James C. Schaff,[1] Dan Vasilescu,[1] Ion I. Moraru,[1] Judy E. Bloom,[1] and Leslie M. Loew[1,*]
[1]R. D. Berlin Center for Cell Analysis and Modeling, University of Connecticut School of Medicine, Farmington, Connecticut

ABSTRACT In rule-based modeling, molecular interactions are systematically specified in the form of reaction rules that serve as generators of reactions. This provides a way to account for all the potential molecular complexes and interactions among multivalent or multistate molecules. Recently, we introduced rule-based modeling into the Virtual Cell (VCell) modeling framework, permitting graphical specification of rules and merger of networks generated automatically (using the BioNetGen modeling engine) with hand-specified reaction networks. VCell provides a number of ordinary differential equation and stochastic numerical solvers for single-compartment simulations of the kinetic systems derived from these networks, and agent-based network-free simulation of the rules. In this work, compartmental and spatial modeling of rule-based models has been implemented within VCell. To enable rule-based deterministic and stochastic spatial simulations and network-free agent-based compartmental simulations, the BioNetGen and NFSim engines were each modified to support compartments. In the new rule-based formalism, every reactant and product pattern and every reaction rule are assigned locations. We also introduce the rule-based concept of molecular anchors. This assures that any species that has a molecule anchored to a predefined compartment will remain in this compartment. Importantly, in addition to formulation of compartmental models, this now permits VCell users to seamlessly connect reaction networks derived from rules to explicit geometries to automatically generate a system of reaction-diffusion equations. These may then be simulated using either the VCell partial differential equations deterministic solvers or the Smoldyn stochastic simulator.

INTRODUCTION

The specification of all molecular species and interactions is usually the first step in modeling a biomolecular interaction network. However, for the interactions of multivalent or multistate molecules, the number of species and reactions can be combinatorially large (1,2), making it impractical to specify the reaction network manually. Rule-based modeling (2,3) overcomes this limitation by accounting for the complete set of reactions and species that arise when an initial (seed) set of species is transformed using reaction rules. The reaction rules can serve either as generators of individual reactions, expanding the initial set of species into the complete network of reactions and species, or as generators of stochastic events, producing molecular complexes with nonzero population numbers.

Virtual Cell (VCell; http://vcell.org) is an open-source platform that provides powerful capabilities for kinetic modeling of cellular systems (4,5) (Fig. 1). A key focus of VCell is to allow modelers to ask how spatial features of cells affect the system behavior. At the simplest level, the relative sizes of compartments affect the concentrations of species transported between them; models that account for the surface areas of membranes and the volumes of volumetric compartments, but assume that diffusion is fast on the timescale of reaction kinetics, will be referred to as "compartmental". If diffusion and spatial localization of molecular species can affect the biology, the geometric shapes of the membrane and volumetric compartments also need to be explicitly considered, and these models are considered "spatial".

In building a VCell "BioModel", users initially describe the system "Physiology" with compartments defined as multiple volumes (e.g., extracellular space, cytosol, nucleus, endoplasmic reticulum, etc.) and surfaces (e.g., plasma membrane, mitochondrial membrane, etc.); the Physiology also encompasses reactions and fluxes taking place within and between volumes and surfaces, with respective volume- and area-based units for concentrations and kinetic rate expressions. Once a Physiology is defined in VCell, any number of "Applications" can be defined that specify the initial conditions, compartment sizes, and/or geometries, and whether the system should be treated deterministically

Submitted June 7, 2017, and accepted for publication August 11, 2017.
*Correspondence: blinov@uchc.edu or les@volt.uchc.edu
Editor: Stanislav Shvartsman.
http://dx.doi.org/10.1016/j.bpj.2017.08.022

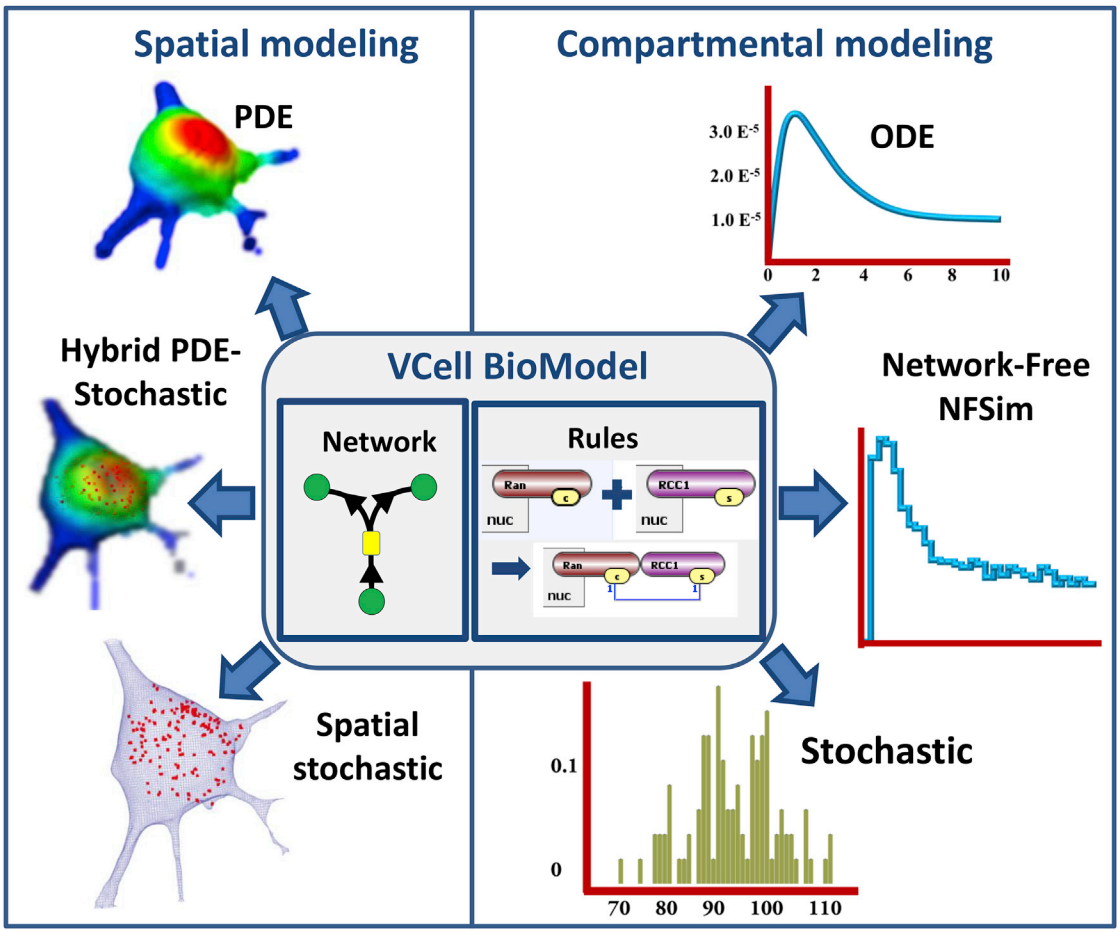

FIGURE 1 Overview of VCell capabilities and v. 6.1 enhancements. The VCell BioModel (at the *center*) can be defined as an explicit reaction network, a set of reaction rules, or as a combination of the two. Before VCell 6.1, only single-compartment models defined with reaction rules could be simulated, which also restricted rule-based modeling to nonspatial Applications. VCell 6.1 now supports rule-based modeling with all these VCell solvers. In addition, a new compartmental NFSim solver is introduced. To see this figure in color, go online.

or stochastically. An Application can be considered to be a virtual experiment and is sufficient to completely define the system's mathematical equations, which are automatically generated. VCell applications that do not include explicitly defined geometry (i.e., making the assumption that diffusion is fast on the timescale of all reaction kinetics) are called "compartmental". These can be simulated using a variety of deterministic and stochastic numerical solvers (Fig. 1) to produce time courses for concentrations and/or population numbers of species. Alternatively, an explicit geometry can be defined using a variety of methods, such as analytic equations in 1D, 2D, or 3D, constructed solid geometry in 3D, image-based (imported from various microscopy formats), mesh-based (imported from STL image), or drawn using provided drawing tools. Such VCell applications are called "spatial", and diffusion for species must be defined to simulate time courses. Diffusion can be defined in both volumetric compartments and along the surface of membranes, and reactions spanning multiple compartments account for species flux between these compartments. VCell offers several solvers for partial differential equations (PDEs) (4–7) to simulate spatial and temporal changes in concentration when the species copy numbers are large. When the copy numbers are low, a spatial stochastic simulator using the Smoldyn simulation engine (8) is available. VCell even has a solver for spatial hybrid deterministic/stochastic simulations (9), to accommodate systems containing some species at high copy number (modeled as continuous concentrations and PDEs) and others at low copy number (modeled with stochastic reaction kinetics and Brownian motion).

Earlier versions of VCell required explicit specification of species and reactions. Last year we introduced VCell 6.0, which incorporated a graphical user interface (GUI) to represent multiple sites and states within molecules and the rule-based reaction kinetics between them (10). This GUI provides a compact method for describing the key structural features of multivalent multistate molecules that control their roles in complex signaling systems. Every chemical species can be represented as structured objects

composed of molecules, with reactions that control all their modifications and changes in their connectivity. Every model can be simulated both as a reaction network (after network-generation using the BioNetGen engine), permitting both deterministic and stochastic simulations, or with the NFSim network-free algorithm, which produces stochastic simulations. However, the abstractions within the representations of molecules and rules, as well as the algorithms within the network generation and simulation engines, did not include compartments or the ability to simulate reaction diffusion equations in explicit geometries.

In this article, we describe a compartmental extension of the rule-based modeling capabilities of VCell, available in VCell 6.1. It enables specification of the locations of molecules and rules. This required us to develop new abstractions to anchor molecules explicitly to volume or surface compartments. We then modified the BioNetGen code to support network generation within the VCell compartmental formalism. This permitted us to support all the stochastic or deterministic, and nonspatial or spatial, simulators available in VCell to simulate reaction networks generated by rules (Fig. 1). Additionally, we modified the NFSim code (11) to support compartmental (albeit nonspatial) network-free simulations.

MATERIALS AND METHODS

Rule-based modeling in VCell is implemented by adapting the standalone software tools BioNetGen (3,12) and NFSim (11). They allow both deterministic and stochastic simulations after the reaction network is generated (BioNetGen) and network-free particle-based simulations (NFSim) in a single compartment. Both tools operate using the BioNetGen Language (BNGL), (12), which was originally designed to work only for a single compartment. A VCell-specific BNGL extension was developed to enable model specification in multiple compartments. The BioNetGen and NFSim tools were each modified to support multiple compartments and new VCell-specific features.

RESULTS

A previously described compartmental extension of BNGL (cBNGL), (13), enables explicit modeling of the compartment topology of the cell and its effects on system dynamics using the BioNetGen network generation engine. However, it is not suited for VCell. The major reason is that cellular topology in cBNGL is restricted to a compartment graph in which nodes represent compartments and directed edges represent containership. This graph must be a tree. A membrane may contain (and be contained in) only a single volume compartment, whereas a volume compartment may contain multiple surface compartments but be contained only by a single membrane. This cellular topology is more limited than the generalized topology available in VCell, where no restrictions are imposed on how compartments can be enclosed within each other. Likewise, cBNGL does not allow for molecular species to span several membranes. The VCell paradigm gives the modeler more flexibility and supports representation of multicellular structures with gap junctions or tight junctions. Additionally, NFSim does not support compartmental simulations using cBNGL. Therefore, to enable rule-based modeling in a generalized topology using both the BioNetGen and NFSim engines, we decided to develop our own schema based on the VCell multicompartment reaction diagram, where the locations of each reactant, product, and of the reaction itself are explicitly specified. This required new conventions for specifying spatial features within a regular BNGL file. We use the BioNetGen or NFSim engines to generate the reactions, but have added a processing step after every iteration to fix the location of each newly generated species and remove invalid species and reactions. Although the BNGL is hidden to the user, it is important to lay out the key algorithmic features that allow for merging of rule-based modeling within the VCell architecture. We describe these in Fig. 2 and Table 1, which also serve as a short glossary of rule-based modeling terminology. Note that although a species is located in a given compartment, its orientation in space (as may be required for agent-based simulations) is implicitly determined by the specification of binding reaction rules between sites on molecules with different locations. Supporting Material discusses implementation in more detail.

Let us illustrate compartmental rule-based modeling using a model of Ran-mediated nucleocytoplasmic transport (available in the VCell Database under Tutorials as "Rule-based_Ran_transport"). This is a simplified version of a published model by Smith et al. (14). In this tutorial (Fig. 3 *A*), the nuclear Ran binds to a cargo molecule, facilitating translocation into the cytosol. Ran and cargo then dissociate. The cytosolic cargo molecule may be phosphorylated on any of its three tyrosines while in the cytosol. When cargo is displaced by the Ran exchange factor RCC1, which is a component of histones, Ran stays in the nucleus. The rule-based approach provides a compact way of describing such a system, with a single transport reaction rule (Fig. 2 *E* and *rule 7* in Fig. 3 *A*) in place of 18 reversible reactions for transport of multiple types of cargo. The total number of species and reactions, if all combinations of phosphoforms are accounted for, would be 36 and 98, respectively. Such a large system is difficult to construct, visualize, and analyze without rule-based specification (Fig. 3 *B*). Moreover, model predictions have to be compared to experimental observations that often correspond to sums of multiple species. In our example, the total phosphorylation of cargo is summed over all species that have at least one site being phosphorylated, with triple-phosphorylated species counted three times. The observable concept (Fig. 1 *D* and Table 1) introduced in rule-based modeling provides an easy way to define such quantities.

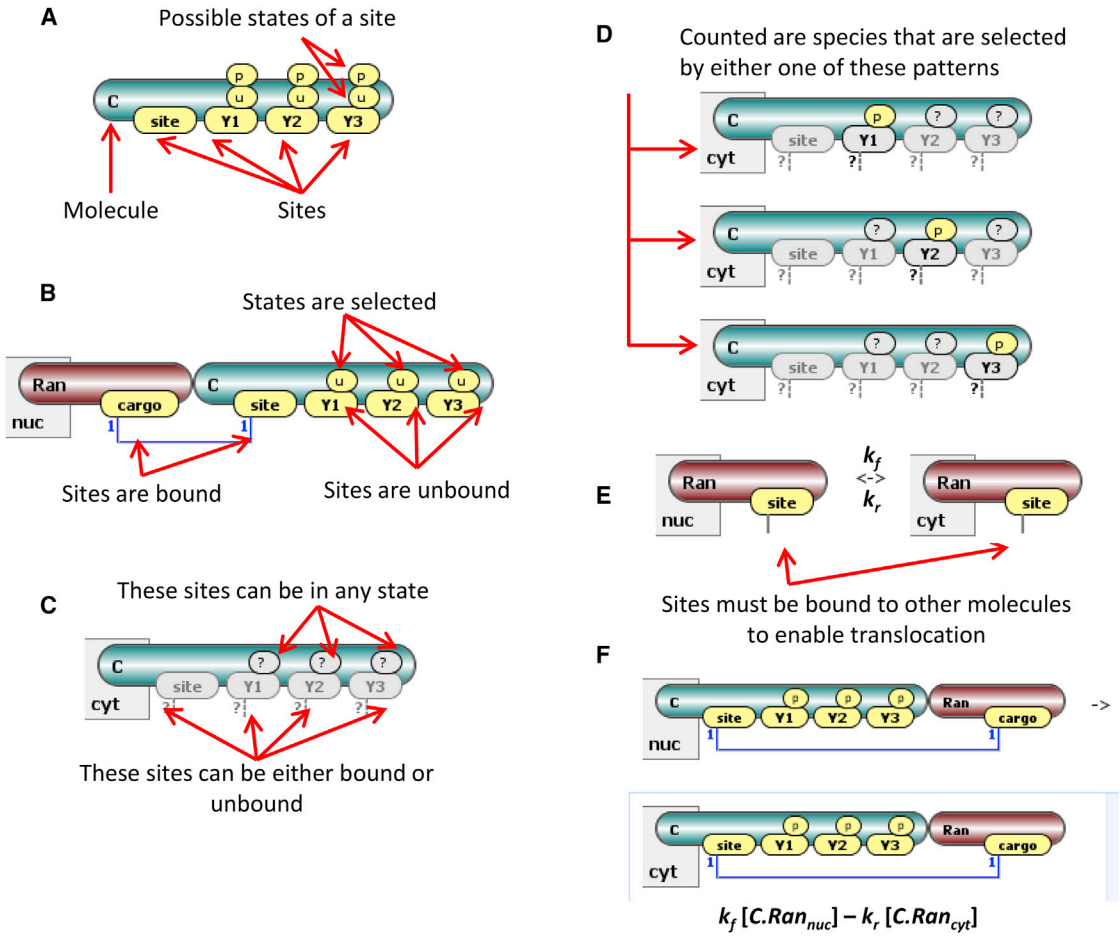

FIGURE 2 Elements of compartmental rule-based modeling. (*A*) Shown here is a cargo molecule C with three tyrosines subject to phosphorylation and one extra binding site. This molecule spawns at least eight different potential species—combinations of possible states of phosphosites. (*B*) Shown here is a species—which is a complex of Ran and C1 in the nuc compartment. All three tyrosines of C are unphosphorylated. (*C*) Here we show a pattern that includes the set of all cytosolic species that have C as its constituent. (*D*) Shown here is an observable that counts all species that have at least one phosphorylated residue of C among its constituents. (*E*) Shown here is a reaction rule for Ran translocation between nucleus and cytosol. The solid line emanating from the site specifies that Ran must be bound to other molecules to translocate. This is a convenient feature when multiple cargo molecules can be bound to this site—there is no need to enumerate all of them. The rate for all reactions generated by the rule is given by the same rate parameters k_f and k_r. (*F*) A sample reaction generated by the rule is parameterized by the same rate constants as the rule. To see this figure in color, go online.

A feature of rule-based modeling is that anything not explicitly constrained is allowed. Thus, a transport reaction rule for Ran with a pattern specifying it must have a bound site (Fig. 2 *E*), will carry with it any of the molecules that may be bound to it, including RCC1, which should remain in the nucleus according to the known biology. To address this within the generalized topology of VCell, we introduced the ability to "anchor" molecules to compartments (Fig. 4). A molecule can be anchored to one or multiple compartments. Anchoring a molecule to specific compartments constrains any species (produced through network generation or by NFSim) that contains this molecule to be in only these compartments. In spatial applications, such species are free to diffuse within compartments they are anchored to.

As summarized in the Introduction, VCell has a hierarchical architecture whereby the system "Physiology" (exemplified by the network in Fig. 3 and all its underlying details) can be associated with several Applications. Applications are used to specify initial conditions, geometries, and the physical and mathematical approaches by which the system should be simulated. New Application types have been developed to provide control for network generation and to support a network-free simulation with NFSim. All the simulation methods previously available to manually constructed reaction networks are thereby now available to networks generated automatically from rules. Fig. 5 shows how the Physiology summarized in Fig. 3 can be used to produce compartmental deterministic (ordinary differential equations), compartmental stochastic, compartmental network-free, spatial deterministic (PDE), spatial stochastic, and spatial hybrid deterministic/stochastic simulations. We believe that VCell is unique in making all these approaches available within one unified software

TABLE 1 Concepts and Definitions for Compartmental Rule-Based Modeling in VCell

Term	Purpose	Composition	Compartmental/Spatial Features
Compartment	cellular structure containing species and reactions	a volume or a surface, corresponding to extracellular regions, cytosol, and organelles and their associated membranes; no specification of the relationship between compartments (such as enclosures or adjacency) is required	for Applications without explicit geometries, compartmental sizes are specified as surface areas or volumes; for spatial Applications, compartments may be explicitly associated with regions within a geometry
Molecule (Fig. 2 *A*)	a building block for species	composed of sites that can bind other sites between or within molecules; a site may also have multiple states; in this way, a molecule spawns a collection of chemical species—one per every combination of site occupancy and/or state	can be, optionally, "anchored" to one or more compartments; a species containing an anchored molecule can be located only in one of these compartments
Species (Fig. 2 *B*)	an individual chemical species that may occur in the model	composed of molecules that are connected through bonds between binding sites; all modification sites must be explicitly defined; a species can be a "seed species" (defined as initial condition and used as a seed for reaction rules application) or a generated species (a result of reaction rule application)	every species is located in a unique compartment; seed species are assigned to a compartment by the user; it may not contain molecules that are excluded from that compartment by anchoring to other compartments
Pattern (Fig. 2 *C*)	specifies a set of possible states of species to be selected as participants in reaction rules and in observables	composed of molecules; the states of sites may be left unspecified; thus a pattern may select multiple species; moreover, binding sites may have "implicit binding status" (has external bond or may be bound) where its binding partner is not explicitly defined; such patterns may be inclusive of species that contain molecules not explicitly specified in a pattern but being possibly bound to molecules within it	defined in a single compartment; all molecules comprising a pattern must be permitted to be located in this compartment
Observable (Fig. 2 *D*)	specifies simulation outputs of interest	consists of one or more patterns that define features of species; the result is the total population (concentration or count) of multiple species	defined in a single compartment; all molecules comprising an observable must be permitted to be located in this compartment
Reaction rule (Fig. 2 *E*)	defines transformation of multiple species at once, generating multiple reactions	species to be transformed are selected by reactant pattern(s); product pattern(s) define the end result of transformation; product may differ from reactant by reassigning molecules, adding or deleting bonds and changing site states; a kinetic expression is also a component of a reaction rule	reaction rule is defined in a single compartment; each reactant pattern and each product pattern are assigned a specific compartment, which may be different from compartment for reactants or products

environment; we are also unaware of any other systems biology software offering a spatial hybrid deterministic/stochastic solver (9).

But VCell does not have pretensions of being a solution for every modeling problem. Accordingly, we have devoted significant effort to interoperability—in particular through the SBML standard (15). Specifically, for the case of rule-based modeling, a model defined in VCell can be exported to cBNGL (13) for simulation with the standalone BioNetGen engine. The generated cBNGL file has a new "anchors block" specifying molecules anchored to compartments. This block is ignored by the BioNetGen compiler. The exported cBNGL has no information about enclosing compartments in the compartment block. Thus,

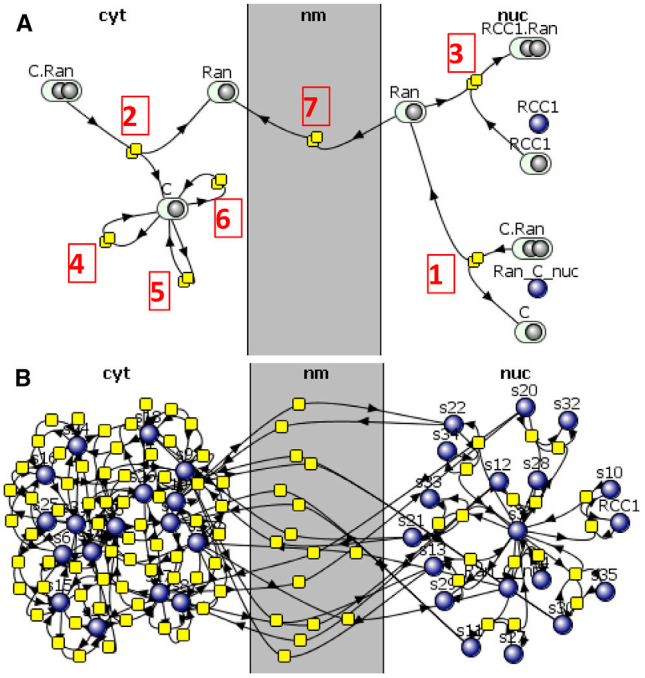

FIGURE 3 Graphical representation of reaction rules and reaction networks. (*A*) Shown here is a collapsed VCell reaction diagram of a rule-based model for Ran transport across nuclear membrane. Ran can bind cargo molecule C in both nucleus (rule 1) and cytoplasm (2). In nucleus, Ran can bind RCC1 (3), whereas in cytoplasm, cargo protein may undergo phosphorylation on all their phosphosites (4–6). Ran can undergo transport across the nuclear membrane (7). The full mechanisms (such as what should be bound to Ran in rule 7) are hidden in this view, but can be seen when the user clicks on a reaction rule node. (*B*) The reaction network generated within VCell from this set of rules contains 36 species and 98 reactions. To see this figure in color, go online.

to simulate an exported model with the standalone BioNetGen engine, a modeler needs to specify a compartmental tree manually. Also, a model specified in cBNGL can be imported into VCell. Not every file can be seamlessly imported; errors will be displayed when compartment specification is done at a level of individual molecules in seed species and patterns. VCell provides a BNGL import editor where all inconsistencies are displayed and explained, so an experienced BioNetGen user should be able to fix all issues during the import process. Supporting Material provides more details on comparison between cBNGL and VCell representation.

DISCUSSION

We have described a major enhancement of the VCell software to enable rule-based modeling in multiple compartments. This enhancement gives users with combinatorially complex biochemical systems the ability to specify all interactions and their dependencies in terms of molecular features such as cellular locations, sites for binding, modification states, or conformations. To achieve this, we built a rich GUI that also serves to help visualize the details of these complex systems (Fig. 3). The VCell "classic" manual network generation functionality and GUI are still available and the implementation actually supports mixing of automatically generated rule-based networks with reaction networks generated manually. Such networks can then be modeled with all the VCell compartmental and spatial simulation methods (Fig. 5). For network-free simulations, we have also modified the NFSim engine to support compartments.

There are limitations to rule-based modeling that users should appreciate. One is the restriction of reaction kinetics to mass action; however, the VCell user may be able to overcome this restriction for deterministic models by judiciously mixing rules and explicit network reactions. A second restriction is that the network-free simulations can only be run for nonspatial models. Additionally, for spatial models based on reaction networks generated by rules, care needs to be exercised not to allow a combinatorial explosion of species and reactions; the resultant large system of PDEs could produce prohibitively expensive computations.

Several rule-based modeling tools that can operate in multiple compartments or perform spatial simulations are

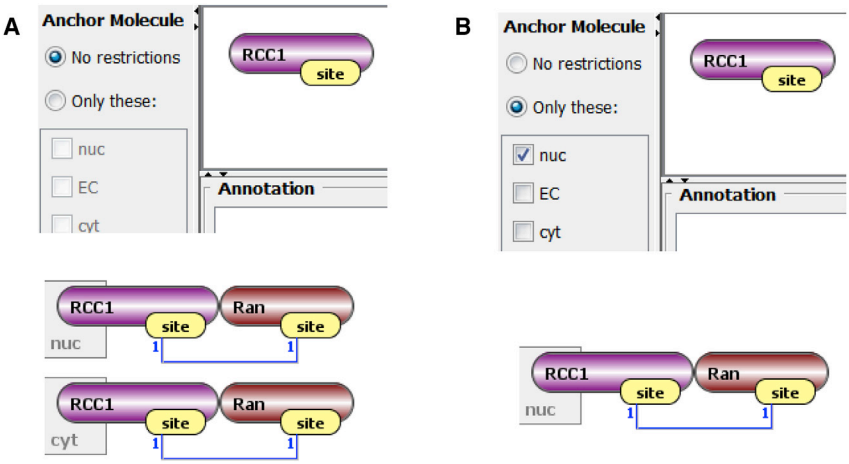

FIGURE 4 Effect of molecule anchoring. (*A*) If "No restrictions" is chosen (*top*), the rule in Fig. 2 *E* will transport RCC1 to the cytosol, producing both species shown (*bottom*). (*B*) Anchoring will prevent this interaction and the only generated Ran-RCC1 species will be in the nucleus. To see this figure in color, go online.

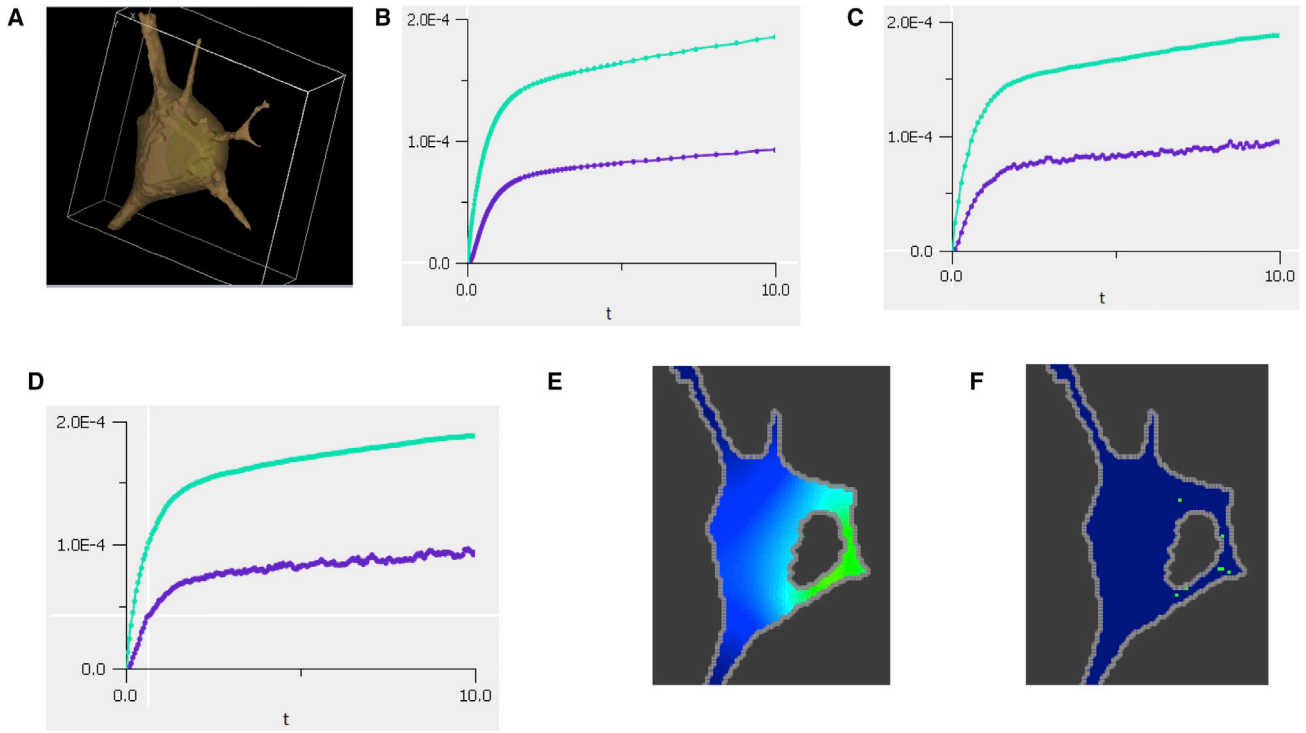

FIGURE 5 Visualization and simulation capabilities of VCell. (*A*) Shown here is an example of a 3D image-based geometry (neuroblastoma cell) that can be imported into VCell from a confocal microscope. A 3D segmentation within VCell identifies regions corresponding to cytoplasm and nucleus (*oval shape* inside cytoplasm). (*B–D*) Shown here are simulation results for the time course of cargo concentration in cytosol of total (*upper curve*) and phosphorylated (*bottom curve*) cargo for deterministic (*B*), stochastic (*C*), and network-free (*D*) simulations. (*E*) Shown here are PDE simulation result for one slice displaying the cytosolic cargo concentration gradient at 1 s. Brighter colors correspond to higher concentration; see also Movie S1. (*F*) Shown here is the same slice for spatial stochastic simulations for 1000 molecules of Ran. See also Movie S1, which shows side-by-side simulation results for spatial deterministic and stochastic applications for a single *XY* slice. To see this figure in color, go online.

available: Simmune (16,17), KaSim (18), Smoldyn (8), Meredys (19), SRSim (20), SSC (21), and SpringSaLaD (22). All of them are exclusively stochastic, whereas VCell offers deterministic spatial and hybrid deterministic/stochastic simulation capabilities. The VCell spatial stochastic solver is based on Smoldyn (23), adapted to permit users to incorporate experimental 3D image-based geometries in simulations; analytical geometries and constructive geometries can also be used. A variety of nonspatial compartmental simulators, both stochastic and deterministic, are also in VCell for quick answers when diffusion is fast on the timescale of reaction kinetics. Additionally, only Simmune and SpringSaLaD have biology oriented GUIs, as in VCell, whereas all the other simulators are based on scripting. But the other more specialized simulators have important strengths that might be needed for certain classes of problems. SSC and Smoldyn both have implementations to employ high performance computing or graphics processing units for computationally intensive simulations. Simmune is specialized for complex signaling in immunology. Meredys, SRSim, and SpringSaLaD are all designed to account for molecular-excluded volume effects and are therefore well suited for simulations where molecular crowding might be important. To facilitate interoperability with such other simulators, VCell supports the SBML standard (15) by enabling export and import of SBML models; it also supports cBNGL export, although some manual editing will be required.

Model sharing is also facilitated within VCell through the VCell database. All models can be stored in the database along with simulations results that were run on the VCell server farm (although users may opt to save models and run simulations on their local machines). Access control is implemented to permit sharing of models with individual collaborators or to make a model openly accessible. Users may annotate model components to connect them to the primary literature sources as well as to ontologies and pathway databases. This is particularly valuable for molecules in rule-based models, where the localization and sites within a molecule can be directly related to both molecular structure and pathway data. Importantly, proper annotation can assure reusability of not just the entire model, but the individual molecules and rules.

Software availability

This software is available as VCell (versions 6.1 and later) at the Virtual Cell web site (http://vcell.org/). The application

installs and runs on all major platforms and does not require registration for use on the user's computer. Tutorials are available at the Virtual Cell website and Help is provided within the software. Source code is available at https://github.com/virtualcell.

SUPPORTING MATERIAL

Supporting Materials and Methods, fourteen figures, one movie, and one QuickStart Guide are available at http://www.biophysj.org/biophysj/supplemental/S0006-3495(17)30915-3.

ACKNOWLEDGMENTS

The authors gratefully acknowledge useful discussions with James Faeder at the University of Pittsburgh and Ann Cowan at UConn Health.

The project is supported by the National Institute of General Medical Science (NIGMS) through grants P41 GM103313 and R01 GM095485.

REFERENCES

1. Mayer, B. J., M. L. Blinov, and L. M. Loew. 2009. Molecular machines or pleiomorphic ensembles: signaling complexes revisited. *J. Biol.* 8:81.
2. Faeder, J. R., M. L. Blinov, …, W. S. Hlavacek. 2005. Combinatorial complexity and dynamical restriction of network flows in signal transduction. *Syst. Biol. (Stevenage).* 2:5–15.
3. Blinov, M. L., J. R. Faeder, …, W. S. Hlavacek. 2004. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics.* 20:3289–3291.
4. Cowan, A. E., I. I. Moraru, …, L. M. Loew. 2012. Spatial modeling of cell signaling networks. *Methods Cell Biol.* 110:195–221.
5. Schaff, J., C. C. Fink, …, L. M. Loew. 1997. A general computational framework for modeling cellular structure and function. *Biophys. J.* 73:1135–1146.
6. Schaff, J. C., B. M. Slepchenko, …, L. M. Loew. 2001. Analysis of nonlinear dynamics on arbitrary geometries with the virtual cell. *Chaos.* 11:115–131.
7. Resasco, D. C., F. Gao, …, B. M. Slepchenko. 2012. Virtual Cell: computational tools for modeling in cell biology. *Wiley Interdiscip. Rev. Syst. Biol. Med.* 4:129–140.
8. Andrews, S. S. 2017. Smoldyn: particle-based simulation with rule-based modeling, improved molecular interaction and a library interface. *Bioinformatics.* 33:710–717.
9. Schaff, J. C., F. Gao, …, B. M. Slepchenko. 2016. Numerical approach to spatial deterministic-stochastic models arising in cell Biology. *PLOS Comput. Biol.* 12:e1005236.
10. Schaff, J. C., D. Vasilescu, …, M. L. Blinov. 2016. Rule-based modeling with virtual cell. *Bioinformatics.* 32:2880–2882.
11. Sneddon, M. W., J. R. Faeder, and T. Emonet. 2011. Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. *Nat. Methods.* 8:177–183.
12. Faeder, J. R., M. L. Blinov, and W. S. Hlavacek. 2009. Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol. Biol.* 500:113–167.
13. Harris, L. A., J. S. Hogg, and J. R. Faeder. 2009. Compartmental rule-based modeling of biochemical systems. *In* Winter Simulation Conference, Austin, TX. 2009:908–919.
14. Smith, A. E., B. M. Slepchenko, …, I. G. Macara. 2002. Systems analysis of Ran transport. *Science.* 295:488–491.
15. Hucka, M., A. Finney, …, J. Wang; SBML Forum. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics.* 19:524–531.
16. Angermann, B. R., F. Klauschen, …, M. Meier-Schellersheim. 2012. Computational modeling of cellular signaling processes embedded into dynamic spatial contexts. *Nat. Methods.* 9:283–289.
17. Zhang, F., B. R. Angermann, and M. Meier-Schellersheim. 2013. The Simmune Modeler visual interface for creating signaling networks based on bi-molecular interactions. *Bioinformatics.* 29:1229–1230.
18. Sorokina, O., A. Sorokin, …, V. Danos. 2013. A simulator for spatially extended κ models. *Bioinformatics.* 29:3105–3106.
19. Tolle, D. P., and N. Le Novère. 2010. Meredys, a multi-compartment reaction-diffusion simulator using multistate realistic molecular complexes. *BMC Syst. Biol.* 4:24.
20. Gruenert, G., B. Ibrahim, …, P. Dittrich. 2010. Rule-based spatial modeling with diffusing, geometrically constrained molecules. *BMC Bioinformatics.* 11:307.
21. Lis, M., M. N. Artyomov, …, A. K. Chakraborty. 2009. Efficient stochastic simulation of reaction-diffusion processes via direct compilation. *Bioinformatics.* 25:2289–2291.
22. Michalski, P. J., and L. M. Loew. 2016. SpringSaLaD: a spatial, particle-based biochemical simulation platform with excluded volume. *Biophys. J.* 110:523–529.
23. Andrews, S. S., N. J. Addy, …, A. P. Arkin. 2010. Detailed simulations of cell biology with Smoldyn 2.1. *PLOS Comput. Biol.* 6:e1000705.

Biophysical Journal, Volume *113*

Supplemental Information

Compartmental and Spatial Rule-Based Modeling with *Virtual Cell*

Michael L. Blinov, James C. Schaff, Dan Vasilescu, Ion I. Moraru, Judy E. Bloom, and Leslie M. Loew

S1. Internal representation of compartments in VCell and use of BioNetGen and NFSim engines

We use BioNetGen and NFSim engines with augmented BNGL files to generate reaction network (BioNetGen) or perform simulations (NFSim) in multiple compartments. Fig S1 illustrates major steps of compartmental network generation:

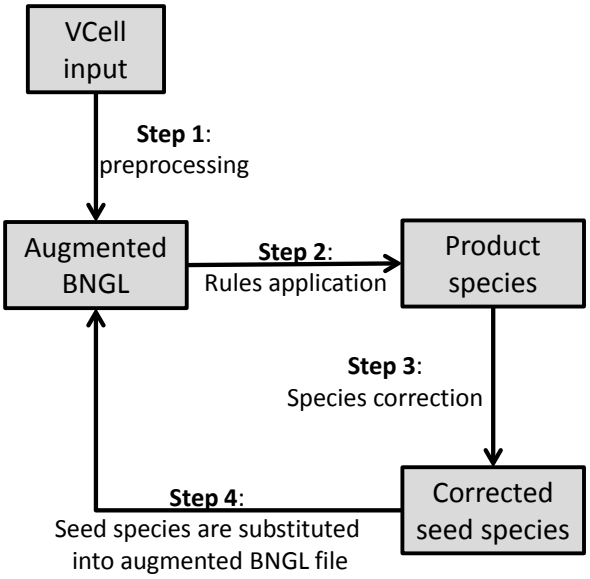

Fig S1. Block diagram for compartmental network generation.

Step 1: An augmented BNGL file is created.

Step 2: A single step of network generation (application of rules to the current set of seed species) is performed using BioNetGen engine.

Step 3: Generated set of product species is corrected to properly locate them in designated compartments.

Step 4: A corrected set of seed species is substituted into the augmented BNGL file.

Steps 2 -4 are repeated until no new species are generated or predefined termination condition is satisfied. Observables are not generated during these steps to speed up the process.

The last step of network generation is performed to generate a full set of reactions and observables. These sets are fixed as in step 3.

In Figure S2 we describe these steps in more detail.

First we describe the augmented BNGL file generated from VCell rule-based model (Box A). We assign two additional functional components as new sites to every BNGL molecular definition (step 1). These sites are used for internal processing only and are not visible to a user. The first added component is called “location”. Its states correspond to possible locations of a molecule. Using anchoring, a modeler can optionally limit the allowable compartments for each molecule. The second added

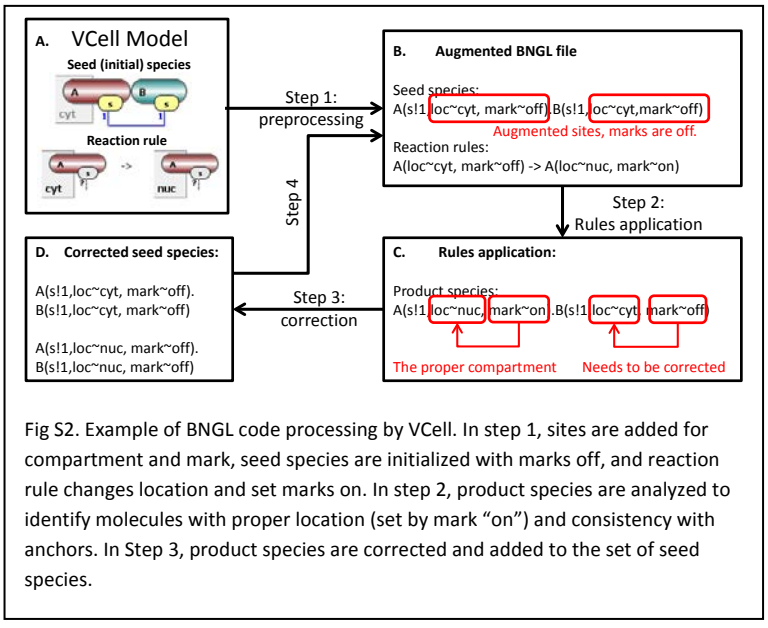

Fig S2. Example of BNGL code processing by VCell. In step 1, sites are added for compartment and mark, seed species are initialized with marks off, and reaction rule changes location and set marks on. In step 2, product species are analyzed to identify molecules with proper location (set by mark “on”) and consistency with anchors. In Step 3, product species are corrected and added to the set of seed species.

component is called “mark”. It defines whether this molecule is explicitly mentioned in a rule and is used to mark molecules that are located in the intended compartment.

User-defined reaction rules are converted into BNGL language for running with BioNetGen and/or NFSim engines. This way, the file can be processed by both BioNetGen and NFSim (NFSim currently does not support cBNGL). In the augmented BNGL file, changes in “location” and “mark” sites from reactant to product species patterns is defined in every reaction rule (Box B).

VCell defines each reactant and product species patterns to be located in a given compartment. Thus, a “location” site is changed from reactant pattern’s compartment to product pattern’s compartment for all molecules explicitly mentioned in a rule definition (Box B).

A “mark” site is changed for all molecules explicitly mentioned in a rule definition from “off” to “on” (Box B). This site is required because a species generated by the rule may contain multiple molecules, including those not explicitly mentioned in the rule definition but are bound to a molecule explicitly present in a reactant or product patterns. These molecules may have contradictory location sites, because a rule does not have any information to change location for molecules not explicitly defined in the rule definition. Thus, the component “mark” is used to resolve this conflict by identifying those molecules (with mark site in “on” state) that are located in the intended compartment. This compartment is then assigned to all molecules in a product species (Box C).

Below we summarize steps of the compartmental network generation using BioNetGen in greater detail. All processing but step 3 is done using Java code in VCell.

1. An augmented BNGL file is created:
    a. All seed species are initialized with “mark” site in “off” state.
    b. Every reversible reaction rule is broken into two irreversible rules.
    c. Every irreversible reaction rule is augmented to change location site and the state of the “mark” site from “off” to “on” in molecules explicitly mentioned in product pattern.
2. A single step of network generation (application of rules to the current set of seed species) is performed using BioNetGen engine.
3. Generate a new set of seed species:
    a. Product species are corrected by changing “location” site of all molecules to the intended location site specified in the rule (identified as a location site of molecules that have a “mark” site in “on” state).
    b. Product species that have molecules in anchored compartments override location in a rule, so such species are located in anchored compartment. VCell user interface assures that there will be no conflict in anchors during model specification.
    c. Duplicated product species are removed.
    d. A new set of seed species is compiled consisting of the previous set of seed species and product species not isomorphic to existing species.
    e. All “mark” sites in the seed species are set to “off”.
4. A new set of seed species is substituted into the augmented BNGL file.

5. Steps 2 -4 are repeated until no new species are generated or predefined termination condition is satisfied. Observables are not generated during these steps to speed up the process.
6. The last step of network generation is performed to generate a full set of reactions and observables. These sets are fixed as in step 3.

To use NFSim for network-free simulation, the same augmented BNGL file is generated (Step 1 above) and converted to XML input for NFSim. NFSim code is updated to correct product molecular constructs and remove duplicates during each simulation time step. Post-processing of species generated by iterations of BioNetGen (Step 2 above) is replaced by post-processing of molecular complexes after each rule firing by NFSim. The NFSim *MoleculeType* class was extended with *bool bHasAnchors* and a vector of anchor names. VCell uses NFSim with bookkeeping enabled, thus complexes are processed at each time point using the same steps as for network generation:

a. After each reaction event, each product complex is searched to identify molecules with “mark” site state set to "on".
b. Find location site state for marked molecules (this is the desired location) and any anchors present in that complex.
c. Reconcile locations of “marked” molecules with allowed locations defined by anchors.
d. For all molecules in the complex, set location site state to the desired location and set all marked site states to "off."

When exporting as BNGL, we use standard cBNGL conventions, as described in supplemental material S2.

S2. Rule-based model specification in VCell vs cBNGL

VCell can import and export cBNGL. However, mapping is not one-to-one.

Below is cBNGL export of the Ran_tutorial_RB model used in the paper.

```
begin model

begin compartments
EC      3       1
cyt     3       1
nuc     3       1
pm      2       1
nm      2       1
end compartments

begin parameters
end parameters

begin molecule types
Ran(cargo)
C(site,Y1~u~p,Y2~u~p,Y3~u~p)
RCC1(site)
```

```
end molecule types

begin anchors
RCC1(nuc)
end anchors

begin seed species
@nuc:Ran(cargo!1).C(site!1,Y1~u,Y2~u,Y3~u) 4.5E-4
@nuc:RCC1(site) 4.5E-4
end seed species

begin observables
Molecules Ran_cyt @cyt:Ran()
Molecules Cargo_cyt @cyt:C()
Molecules RCC1_nuc @nuc:RCC1()
Molecules Cargo_phosp_cyt_total @nuc:C(Y1~p!?) @nuc:C(Y2~p!?) @nuc:C(Y3~p!?)
Molecules Cargo_nuc @nuc:C()
Molecules Cargo_phosp_cyt @cyt:C(Y1~p!?,Y2~p!?,Y3~p!?)
Molecules Ran_bound_cyt @cyt:Ran(cargo!+)
end observables

begin reaction rules
Transport: @nuc:Ran(cargo!+) <-> @cyt:Ran(cargo!+)                    2.0 * 602.0, 0.0
Ran_C_bind_cyt:    @cyt:Ran(cargo!1).C(site!1) <-> @cyt:Ran(cargo) + @cyt:C(site)                    1.0, 100.0
C_p1:    @cyt:C(Y3~u!?) <-> @cyt:C(Y3~p!?)                    10.0, 1.0
C_p2:    @cyt:C(Y2~u!?) <-> @cyt:C(Y2~p!?)                    10.0, 1.0
C_p3:    @cyt:C(Y1~u!?) <-> @cyt:C(Y1~p!?)                    10.0, 1.0
Ran_RCC1_bind:    @nuc:Ran(cargo) + @nuc:RCC1(site) <-> @nuc:Ran(cargo!1).RCC1(site!1)                    1.0, 100.0
Ran_C_bind_nuc:    @nuc:Ran(cargo!1).C(site!1) <-> @nuc:Ran(cargo) + @nuc:C(site)                    1.0, 100.0
end reaction rules

end model

generate_network({max_iter=>10,max_agg=>10,overwrite=>1})
```

To import it into BioNetGen, the following corrections are needed:

1. A compartments block has to be modified to specify enclosing compartments and the hierarchy EC-> pm -> cyt -> nm -> nuc:

   ```
   begin compartments
   EC      3       1
   pm      2       1       EC
   cyt     3       1       pm
   nm      2       1       cyt
   nuc     3       1       nm
   end compartments
   ```

2. Anchors block to be removed.
3. Check units consistency. BioNetGen does not import units, and only requires that they are consistent. VCell operates with units that may be distinct on membrane and in volumetric compartments. In particular, in the example above rates for mass-action kinetics for all volumetric reactions are expressed in per second (first order) and per second per micromolar

(for the second order reactions), while a transport reactions on a membrane are expressed surface densities, so the rate is in molecules per um^2 per second per micromolar.

After these modifications, the file can be processed and generates 37 species and 100 reactions, as if VCell model when RCC1 is not anchored. To make a cBNGL model identical to the VCell, additional specifications are needed that are beyond the scope of this paper.

The import of cBNGL files into VCell requires few modifications (errors are displayed in a pop-up window while importing):

1. In species block, change Molecule @compartment to @compartment:molecule syntax
2. By default, universal rules (rules without explicit compartments defined) will be created in the first compartment (by specification sequence in the BNGL file). Such rules need to be manually duplicated among other required compartments (VCell provides a button for a duplication operation).
3. In reaction rules and observable blocks, change species_pattern @compartment to @compartment:species_pattern syntax, having one compartment per species pattern. Molecules in species patterns cannot be assigned to different compartments, as admissible in cBNGL.

Quick Start Guide:
Create & Simulate Rule-Based Models in VCell 6.1

VCell installation and set-up

Go to http://www.vcell.org/ , select "DOWNLOAD VCELL" from the top menu, choose the proper OS and click the link to VCell **6.1**.

Internet-dependence: After installation, you can run the local VCell application without an Internet connection and without logging in to the model database; in this case, however, you will only be able to work with local files being saved by using File > Export. Whenever you are connected to the Internet, VCell will automatically check for and download any updated version.

Navigation through VCell Modeling Framework

The Virtual Cell opens with four panels.

- The **upper left panel** of VCell provides navigation links to different steps of the VCell modeling process that become active in the **top right panel**.
- Select any element in the top right panel and you can view/edit its **Object Properties** in the **bottom right panel**.
- The **bottom left panel** shows models stored in VCell databases and other resources.

Quick tour of rule-based modeling

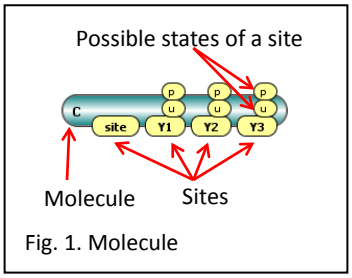

Fig. 1. Molecule

Rule-based modeling involves the representation of **species** as structured objects consisting of **molecules** and molecular interactions as **reaction rules** for transforming the attributes of these objects. It allows one to systematically incorporate site-specific details about molecular interactions into a model.

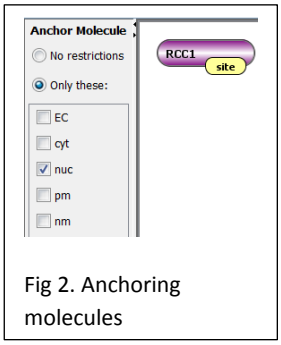

Fig 2. Anchoring molecules

Molecules (Fig. 1) are the principal component of rule-based model specification. Molecules are comprised of sites that can bind to each other, both within a molecule and between molecules. Sites typically represent physical parts of proteins, such as extracellular and trans-membrane domains or phosphotyrosines of a receptor. Sites may also be associated with a list of states, intended to represent states or properties of the site, e.g. phosphorylation status.

Molecule can be anchored to specific locations (Fig. 2), so all species that contain this molecule will be located in these locations only.

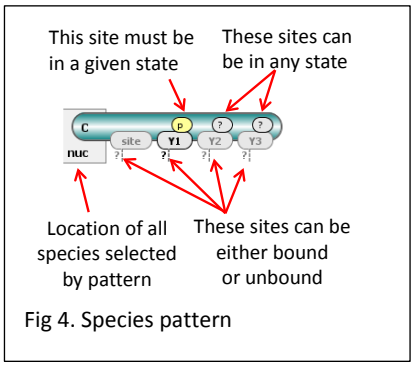

Fig 4. Species pattern

Species (Fig. 3) are composed of molecules with bonds connecting binding sites. If a Molecule has multiple states, each state must be specified - otherwise the species will present a pool of different chemical entities and will not be valid.

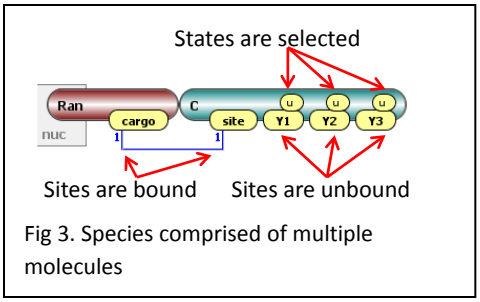

Fig 3. Species comprised of multiple molecules

Species Patterns (Fig. 4) specify a set of possible species to be selected as participants in reaction rules and in observables. Patterns are comprised of molecules. The states of sites may be left unspecified; thus a pattern may select multiple species. Moreover, binding

sites may have implicit binding status (has external bond or may be bound) where its binding partner is not explicitly defined. Such patterns may be inclusive of species that contain molecules not explicitly specified in a pattern but being possibly bound to molecules within it.

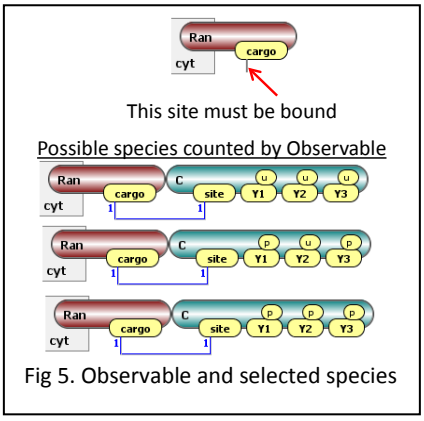

Fig 5. Observable and selected species

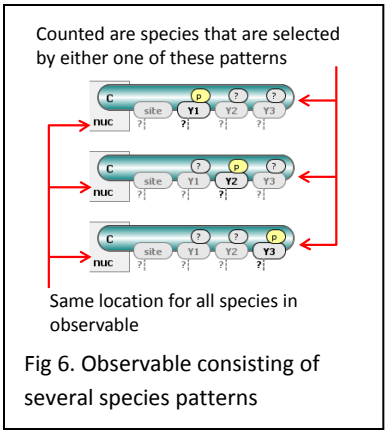

Fig 6. Observable consisting of several species patterns

Observables (Fig. 5) are used to specify model outputs, which are functions of the population levels of multiple chemical species that share a set of properties. For example, to simulate the tyrosine phosphorylation level of a particular protein, one needs to specify an Observable that is a function for the total amount of all chemical species containing the phosphorylated form of this protein. Observables are computed over a set of chemical species (Fig. 6) that share similar properties.

Reaction rules define transformation of multiple species at once (Fig 7, 8). Species to be transformed are selected by reactant pattern(s). Product pattern(s) define the end result of transformation. Product pattern may differ from reactant by adding, deleting or re-assigning of molecules, adding or deleting bonds and changing site states.

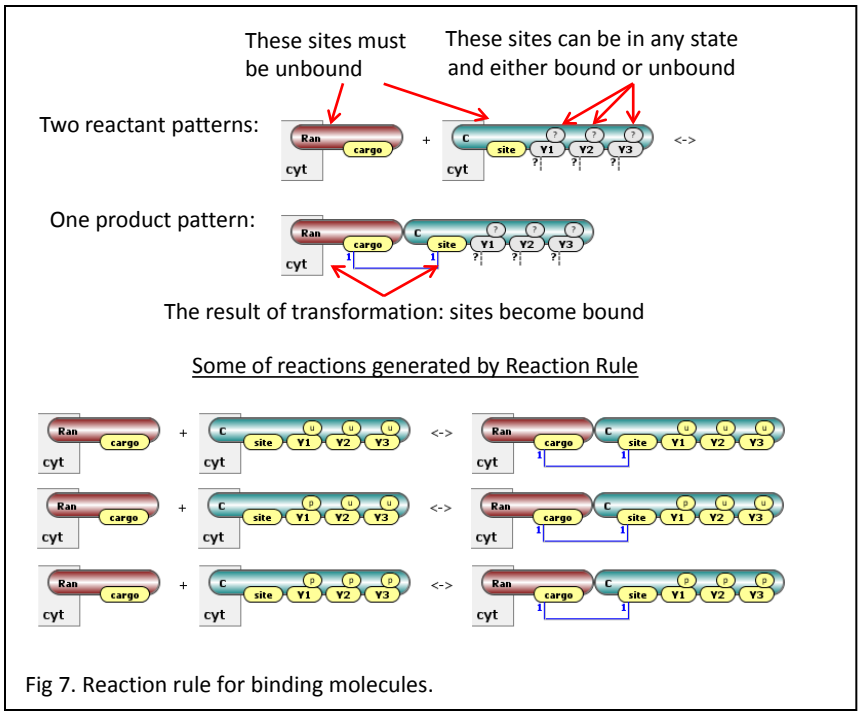

Fig 7. Reaction rule for binding molecules.

Network generation: given an initial set of species with non-zero amounts (**seed species**) and a set of reaction rules, every rule in the set is applied to those species among the seed species that can be selected as reactants, generating proper well-defined reactions and an extended set of species (seed species and species that are products of these reactions). The set of rules is again applied to this extended set of species to generate even more reactions and more species. This process is iteratively repeated until no new species and reactions are generated.

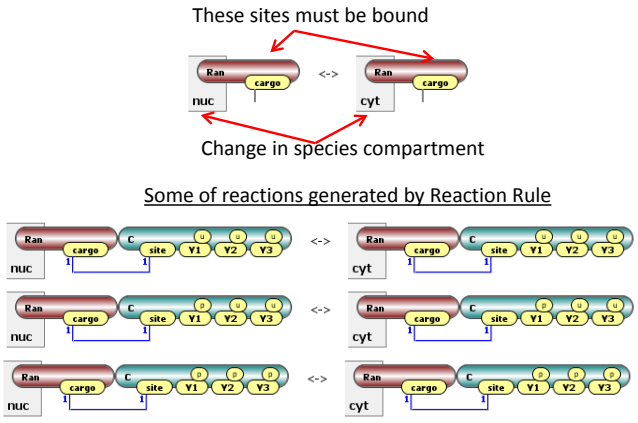

Fig 8. Reaction rule for transport.

BioNetGen (**Bio**logical **Net**works **Gen**erator, http://pubmed.org/19399430) and **NFSim** (**N**etwork-**F**ree **Sim**ulator, http://pubmed.org/21186362) are two back-end engines that are used in VCell 6.0 to work with rule-based models.

Load an existing rule-based model

Public rule-based models are located in **Tutorial VCell 6.1 (Rule-Based)** in bottom left panel under **VCellDB -> BioModels**. These models are accessible when users log in to VCell from the Internet.

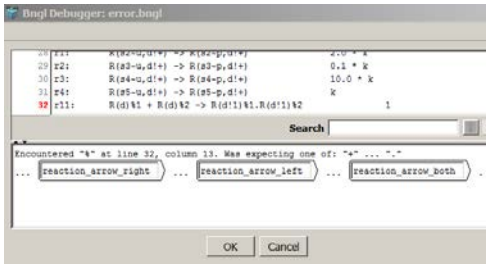

Fig 9. Correcting errors in VCell import

If you have a BNGL code, you can choose **File -> Import**. The code will be imported and a rule-based model will be created with two applications: Network-Free called NFSim, and deterministic network application called BioNetGen. If some features of BNGL file are not supported by VCell, a pop-up window is launched with a suggestion to correct unsupported features (Fig 9). When some features in BNGL file (such as fixed value of concentration) are supported by BioNetGen but not NFSim, only a single deterministic rule-based application will be created. You will be asked about units and simulation volume during import.

A model can be saved to the database (**File > Save As)** or locally **(File->Save As Local)**.

Create a new rule-based model

Please note that **Help->Help** menu provides a very detailed searchable description of all steps in modeling, definitions and specifications. This guide is intended to give just a short overview of modeling process.

1. **Create one or more Molecules** (Fig. 10).
2. **Create one or more Species** (Fig 11).

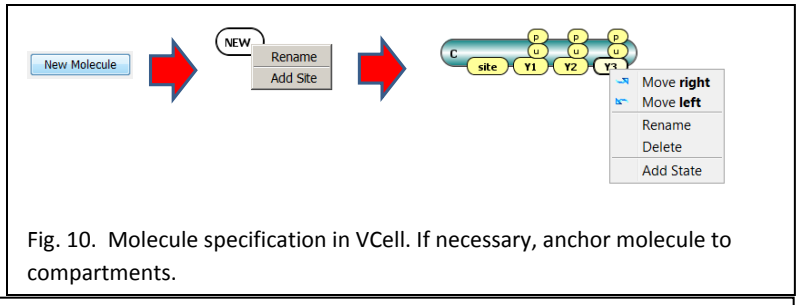

Fig. 10. Molecule specification in VCell. If necessary, anchor molecule to compartments.

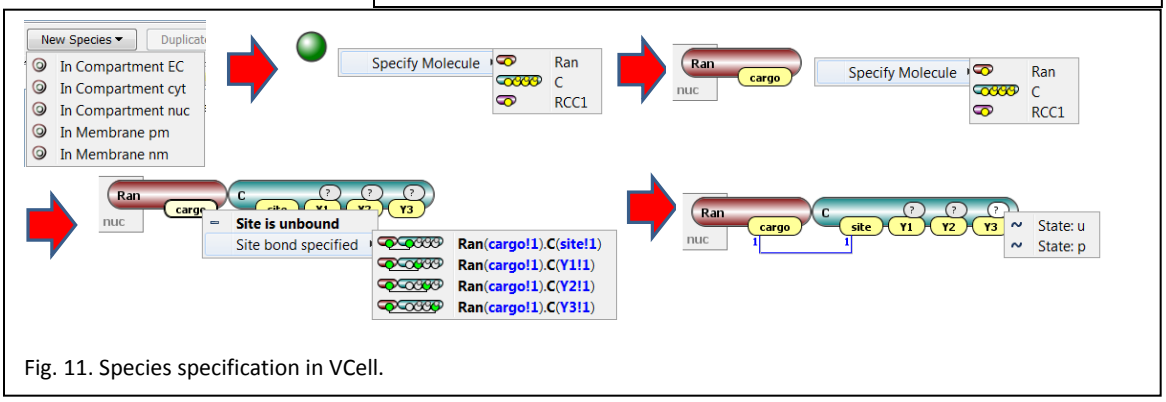

Fig. 11. Species specification in VCell.

3. **Create one or more Observables** (Fig 12).

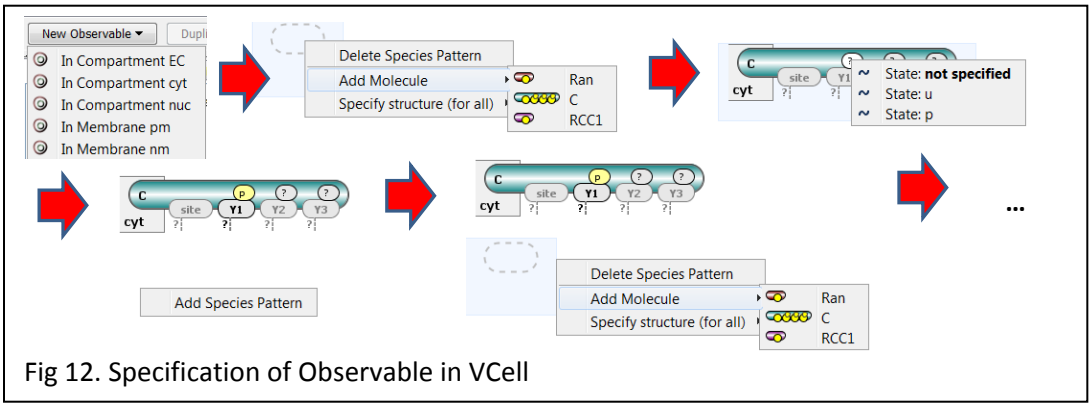

Fig 12. Specification of Observable in VCell

4. Create one or more Reaction Rules (using **Editor** panel in **Reactions**, Fig. 13).

After reaction rule is specified, click on **Kinetics** tab and set whether the rule is reversible, forward and reverse microscopic rate constants for default mass action kinetic law. By default, all reactions generated by a reaction rule have a mass-action kinetic law with forward and reverse rate constants specified here.

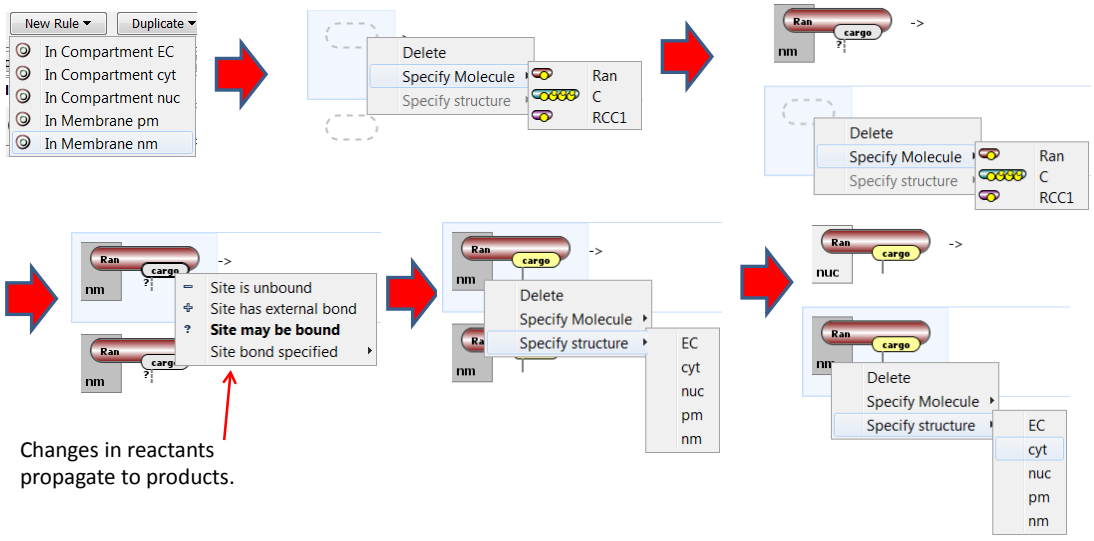

Fig 13. Specification of reaction rule for a transport reaction rule

However, these constants can be adjusted by statistical and symmetry factors (see "Expert Options" section below).

Simulate rule-based model

Once rule-based features are specified in **Physiology**, a user has multiple choices of how to simulate. As standard with Virtual Cell, a Rule-Based BioModel may have numerous **Applications**, including **Deterministic**, **Stochastic** and **Network-Free** applications. Each Application, in turn can have multiple **Simulations**, in which different numerical values or simulation conditions are used.

- **Deterministic** and **Stochastic applications** rely on BioNetGen to create the reaction network. Network specification parameters (**Maximal Number of Iterations** to apply rules and **Maximal Number of Molecules per Species** in generated network) can be used to control the size of the network, which otherwise can become infinite or too large. After the network is generated, regular deterministic or stochastic simulators can be run.
- **Network-free Application** simulation avoids network generation and directly simulates the amounts of model observables using NFSim engine. Rather than generating and tracking all possible chemical species, network-free approach follows only the molecular configurations that exist at a given time.

Deterministic Non-spatial Applications: d/dt

- Select Application in the upper left panel, right click **-> New Application -> Deterministic**. By default, new Application is a single compartment.
- Go to **Specifications -> Species** to specify initial values of species and fixed values (clamping).
- Go to **Specifications -> Reactions** to enable/disable some of reaction rules.
- Go to **Specifications -> Network** to check how a reaction network looks like for different constraints on network generation. **Apply** constraints after you're happy with the network.
- Go to **Specification->Simulations** to set simulation parameters and run.
    - Create and manipulate Simulations using the icons on the left to **Add New**, **Copy**, **Edit**, **Delete**.

- Click **Edit** to check solver settings. Use the **Parameter**s tab in the edit simulation window to create Simulations with specific parameter values being overridden.
- **Green triangle** on the right sends simulation to the server. Simulation results will be available after simulation is completed. Logging in is required. After green button is pressed, any other work with VCell may be continued, or a user can close VCell completely. Simulation results will be saved in the database.
- **Blue triangle** on the right performs simulations on a client's computer. No other VCell operations are possible till simulation is completed.
- Simulation results can display Observables, Species (that were specified in Physiology), Generated Species (those generated in the network – their structure can be seen in **Specification -> Network**) and Other – which include reaction fluxes and user-defined functions.
- Ctrl and Shift can be used to select multiple outputs.
- Simulation Results for "green" run can be viewed in a separate window invoked by clicking on the **Results** icon on the left. Results can be viewed while a Simulation is still running; the data displayed will update automatically at the same time with the Simulation status.

Deterministic Spatial Applications: 

- Most settings are the same as for deterministic application, plus add extra specifications:
- Go to **Specifications -> Geometry** to specify 2D or 3D geometry and map it to compartments.
- In **Specifications -> Species** define diffusion constant for seed species. All newly generated species will have the same diffusion constants.
- Go to **Specification->Simulations** to set simulation parameters and run.
  - All parameters same as in deterministic non-spatial.
  - Under Edit -> Mesh, define Mesh size for spatial simulations, under Edit -> Solver specify solver settings.

Stochastic Non-spatial Applications: 

- Most settings are the same as for deterministic non-spatial.
- **Specifications -> Species** will give a value of either concentrations, or particle numbers. Simulations will be performed in particles. Conversion is done using compartment volume that is defined in **Specifications->Geometry**. Make sure simulation volume is appropriate to avoid very large number of particles.
- Simulation results can be seen in concentrations and number of particles (name_Count).

Network-Free Applications: 

- Most settings are the same as for stochastic non-spatial application.
- This is a stochastic application, so compartment volume must be defined in **Specifications->Geometry**
- **Specifications -> Network-Free** can be used to generate a rule-based model from a regular reaction network. For a purely rule-based model pressing this button will generate a new rule-based model identical to the existing. Check "Expert Options" section for more details.
- NFSim simulator has special settings – check them carefully under **Simulation -> Edit->Solver**.

Stochastic spatial Applications: 

- Same as for stochastic non-spatial.
- Go to **Specifications -> Geometry** to specify 2D or 3D geometry and map it to compartments.
- In **Specifications -> Species** define diffusion constant for seed species. All newly generated species will have the same diffusion constants.

- **Specifications -> Species** will give a value of either concentrations, or particle numbers. Simulations will be performed in particles. Conversion is done using compartment volume that is defined in **Specifications->Geometry**. Make sure simulation volume is appropriate to avoid very large number of particles.
- Simulation results can be seen in concentrations and number of particles (name_Count).

Working with polymers

- If a reaction rule has multiple identical molecules, they are enumerated to provide one-to-one mapping from reactants to products. The match is established automatically, but can be reassigned with a right click on a molecular shape.
- Observables can be specified as multimolecular complexes as in Figs 5 & 6. Observables that count species comprising of multiple identical molecules are defined as polymers (Fig. 14).

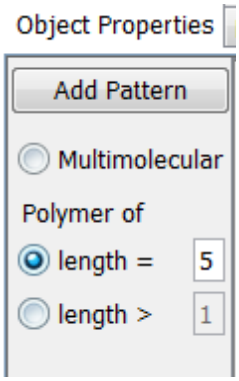

Fig 14. Defining polymeric observable that consist of exactly 5 molecules of a given type.

Expert Options

- Every VCell BioModel can be enhanced by creating Molecules and assigning molecular composition to species.
- One can mix reactions and reaction rules (see *Mix_Reactions_Rules* model in Tutorial folder). In particular, species that are used as initial seed species for rule-based models can be used as species in reaction networks.
- Only mass-action kinetics is supported in reaction rules. In deterministic applications, one can put expressions depending on Species as forward and reverse rates (see *RB_Enzyme_Kinetics* model in Tutorial folder). In Stochastic and Network-Free Applications, only numerical expressions are allowed.
- Using Network-Free Application, please be aware:
    - While it can be used for every VCell BioModel obeying the limitations above, it is designed to be used to simulate rule-based models.
    - This is a stochastic simulation, so simulation results will vary by run.
    - It operates with particles, so using “copy as” to convert a deterministic application to a Network-Free application may lead to a very large number of particles. This number can be adjusted by changing simulation volume in **Specifications -> Geometry**. The default maximal number of particles is 200,000 (it may be increased in **Simulations -> Edit -> Solver**)
    - The user is unable to see what species are populated during a simulation. To test if reaction rules produce expected species and reactions, the user is advised to generate a deterministic rule-based application and test network properties.
- Using Deterministic and Stochastic Applications, one can create a new VCell BioModel that consists of all generated species and reactions; species have molecular details, while reactions carry rule name under reaction name. Note that reaction rates in reaction network are adjusted for symmetry factors and statistical factors:
    - The rate law associated with a rule is a microscopic rate for reactions of the form A+B -> products(). If a rule generates a reaction of the form A+A -> products(s), a reaction rate for such symmetric reaction is multiplied by a **symmetry factor** of 1/2.
    - In some cases, a reaction generated by a rule can occur in multiple ways that are indistinguishable. For example, a rule A.A -> A.A’ can be applied in two different ways: either the first A wil be modified, or the second A will be modified. In these cases, the single-site rule rate law is multiplied by a statistical factor 2 to obtain the rate of the reaction.

- For any VCell BioModel that satisfies above constraints (mass-action only) one can create a rule-based model by creating Network-Free Application and going to **Specifications -> Network-Free -> Create new Rule-Based VCell BioModel.** If no rule-based elements were present, new molecule will be created for every species and trivial rules corresponding reactions will be done. If species have molecular structures, they will be used in rule-based model. Note that the same reaction expression treated as a reaction and as a reaction rule will have different rates:
    - VCell will take care of reactions of the form A + A -> product(s), and the rate of generated reaction rules will be multiplied by 2 to account for symmetry factors.
    - VCell does not take care of reactions where reactants have symmetric sites and reaction rule will have a statistical factor. If this is not properly treated by a modeler, the resultant rule-based model may show different simulation results than its network precursor.

Limitations compared to stand-alone tools (temporary, will be lifted in future releases of VCell). (for BioNetGen/NFSim experts)

- Only mass-action kinetic laws are supported. No functions are supported (only expressions in forward and reverse rates). Total rate is not supported.
- Molecules cannot have identical sites.
- No exclude/include operations are supported.
- All reaction rules with identical molecules are automatically assigned matches between reactant and product patterns.
- Matches between molecules in reactants and products in BNGL file are not imported, but can be reassigned within VCell interface.
- To speed-up simulations, the number of generated species cannot exceed 800 and the number of reactions should be less than 2,000. If network constraints provide larger numbers, they will not be accepted for deterministic and stochastic simulations.