Chapter 9

Rule-Based Modeling of Signal Transduction: A Primer

John A.P. Sekar and James R. Faeder

Abstract

Biological cells accomplish their physiological functions using interconnected networks of genes, proteins, and other biomolecules. Most interactions in biological signaling networks, such as bimolecular association or covalent modification, can be modeled in a physically realistic manner using elementary reaction kinetics. However, the size and combinatorial complexity of such reaction networks have hindered such a mechanistic approach, leading many to conclude that it is premature and to adopt alternative statistical or phenomenological approaches. The recent development of rule-based modeling languages, such as BioNetGen (BNG) and Kappa, enables the precise and succinct encoding of large reaction networks. Coupled with complementary advances in simulation methods, these languages circumvent the combinatorial barrier and allow mechanistic modeling on a much larger scale than previously possible. These languages are also intuitive to the biologist and accessible to the novice modeler. In this chapter, we provide a self-contained tutorial on modeling signal transduction networks using the BNG Language and related software tools. We review the basic syntax of the language and show how biochemical knowledge can be articulated using reaction rules, which can be used to capture a broad range of biochemical and biophysical phenomena in a concise and modular way. A model of ligand-activated receptor dimerization is examined, with a detailed treatment of each step of the modeling process. Sections discussing modeling theory, implicit and explicit model assumptions, and model parameterization are included, with special focus on retaining biophysical realism and avoiding common pitfalls. We also discuss the more advanced case of compartmental modeling using the compartmental extension to BioNetGen. In addition, we provide a comprehensive set of example reaction rules that cover the various aspects of signal transduction, from signaling at the membrane to gene regulation. The reader can modify these reaction rules to model their own systems of interest.

Key words: Computational systems biology, Mathematical modeling, Combinatorial complexity, Signal transduction, Formal languages, BioNetGen, Stochastic simulation, Ordinary differential equations, Network-free simulation, Compartmental modeling, Compartmental modeling, Protein–protein interactions, Signal transduction, Kinase cascades, Signaling diagrams

1. Introduction

Biological cells have evolved complex molecular interaction networks that involve genes, proteins, and small molecules and function as information processing units. These networks are

Xuedong Liu and Meredith D. Betterton (eds.), *Computational Modeling of Signaling Networks*, Methods in Molecular Biology, vol. 880, DOI 10.1007/978-1-61779-833-7_9,

dynamic and self-modifying, responding to cues integrated from the internal and external environments. The traditional approach to understanding signal processing inside cells has been reductionist, i.e., to examine every component in isolation and characterize its microscopic behavior based on its structure, interaction partners, and physiological effect that can be observed from its overproduction or negation. Based on the relationships between components, molecules were organized into roughly linear signaling cascades that were useful in comprehending the overall role of each set of components.

However, the preponderance of experimental and structural evidence obtained in the last few decades tells a different story (1–3). Signaling cascades overlap with each other by sharing components and are not strictly delineated, resulting in significant cross talk between them (e.g., ref. 4). Given two different contexts (a broad term encompassing chemical, spatial, and temporal aspects (1–3)), the same signal can elicit different responses. Thus, in order to understand complex and nonobvious emergent phenomena, a system-wide mechanistic approach to modeling is necessary. Such an approach is complementary to the reductionist approach and builds on the available data.

When studying large networks or even large subnetworks, gross approximations are often employed that form the basis of statistical and bioinformatics approaches to network modeling (5). Explicit reaction kinetic modeling of signaling pathways may be more realistic, but has been limited by several issues (6): the high level of uncertainty in current knowledge about the key molecular players and interactions, the combinatorial complexity that arises from multiple states of each molecule and multiple molecules in a complex, the computational effort required to simulate large reaction networks using generally available methods, and the difficulty in maintaining and reusing large-scale models.

One solution to the above issues still followed by many practitioners in the field is to ignore the many combinations of complexes and many possible molecular states in order to achieve a model size that can be understood, simulated, and analyzed. Such approximations on model structure may be subjective and are usually difficult to test.

Recently, several groups have developed new modeling languages and approaches designed to enable the development of large-scale biochemical models without the need to make ad hoc assumptions. In these so-called rule-based approaches (reviewed in refs. 7, 8), molecules are treated as objects with a defined substructure and reactions are described using reaction rules. Unlike the standard reaction description where the entire reactant species is named distinctly, in a reaction rule, it suffices to represent only those parts of a reactant that actually participate in the reaction or affect the ability of the reaction to occur or its rate.

Since these substructures can be shared between many reactants, a reaction rule effectively represents multiple reactions with identical kinetic descriptions and parameters. This also results in a reduction in the number of essential parameters required by the model. By taking advantage of the modularity exhibited by protein interactions, one can effectively represent a combinatorially complex (potentially infinite) reaction network by a much smaller and finite set of reaction rules. By providing a formal language to intuitively represent detailed biophysical and biochemical information and providing algorithms to automatically mathematize and simulate such models, rule-based modeling languages (such as BioNetGen (BNG) (9, 10) and Kappa (11, 12)) form a much-needed bridge between descriptive reductionist biology and exact mathematical modeling and simulation.

BNG (9, 10) is a rule-based modeling framework and language that has previously been described in detail by Faeder et al. (9). While Faeder et al. focused on presenting a comprehensive description of the modeling language and software capabilities, we present here a self-contained tutorial on developing models following the rule-based approach. Although some aspects of the process are specific to the BNG modeling language (BNGL) and tools, much of the information presented is broadly applicable to developing detailed reaction kinetic models of signaling.

We show that the rule-based syntax and approach provide an excellent way not only to develop models and simulations, but also to encode biochemical knowledge in a concise and modular way. We also do not assume any specialized knowledge on the part of the reader. We provide suggestions about common difficulties in the creation and representation of large reaction models in general and rule-based models in particular. Most importantly, we provide a comprehensive set of scenarios that can be easily studied and modified by a novice modeler and applied to any biological system of interest.

The remainder of this chapter is structured as follows. Subheading 2 indicates where to obtain BNG and related software. Subheading 3 introduces the elements of a rule-based model and the basic syntax of BNGL. Subheading 4 illustrates the basic techniques and nuances behind writing reaction rules. Subheading 5 discusses theory and methods necessary for large-scale reaction models. Subheading 6 shows how to construct, simulate, and analyze a complete BNGL model using biological knowledge and hypotheses. Subheading 7 discusses the more advanced concept of rule-based models in a structured hierarchy of compartments. Subheading 8 talks about challenges involved in parameterizing signal transduction models. Subheading 9 builds on these technical aspects and provides examples of common biological mechanisms of signal transduction, including complex assembly at the

membrane, secondary messenger activation, kinase cascades, and gene regulation. Finally, a summary of key points is provided in Subheading 10.

2. Software

In order to follow the examples and gain hands-on experience with the software, it is highly recommended that the reader obtain access to BNG before proceeding. Two convenient interfaces are available. First, a stand-alone graphical user interface called RuleBender can be obtained free of charge from http://rulebender.org. RuleBender includes a model editor, network browser, simulation interface, and simulation results viewer. Binary distributions of RuleBender are provided free of charge for Windows, Mac, and Linux, and these also include both binaries and source code for BNG. RuleBender can be used to construct, simulate, and plot the output of all examples presented in this chapter. The latest developmental version of the BNG source code as well as additional documentation and model examples can be found at http://bionetgen.org.

Second, a Java-applet-based interface is available free of charge through the Virtual Cell modeling platform (http://vcell.org). The interface to BNG contains an editor, simulation interface, and plotting tools. Not all advanced features of BNG are available, but this is a good way to try modeling with BNGL without having to download or install any software.

3. Model Representation

3.1. Molecules and Components

In the context of signal transduction, a reaction kinetic network typically involves multiple functional complexes of biological entities, such as proteins and small molecules. In a traditional reaction kinetic model, each configuration of the complex is given a unique name and treated as a single unit that can participate in reactions. The modeler would need to identify every unique configuration and assign it a name or a label. From this list of names, the modeler would have to pick out every possible combination of reactants and products and write out the reactions by hand. Due to the combinatorial nature of association between proteins, this approach can be cumbersome and limits not only the modeling process, but also the computational memory required to store, manipulate, and simulate a model.

Rule-based modeling takes a more structured approach. Indivisible entities in the model that associate with other indivisible entities are called *molecules*. Multiple functional attributes for each

molecule type are described using a number of *components*. This mirrors the hierarchical nature of real biological entities, for example, proteins have multiple functional substructures such as domains, motifs, and binding sites. These components can be unchanging (called *stateless*) or exist in one of many different internal *states*. For example, certain binding motifs may have different behaviors depending on whether they are unphosphorylated or phosphorylated.

Consider a receptor with the following functional substructures: a ligand-binding site and a channel that is either open or closed. The BNG *molecule type* blueprint for such a molecule would be `R(lig,ch~open~closed)`. `R` is the name of the receptor molecule, `lig` is the ligand-binding site, and `ch` is the channel. `lig` is stateless, whereas `ch` can take states `open` or `closed`, indicated using the tilde symbol (~).

Multiple components within a molecule type can have the same name and will be treated identically. For example, the *molecule type* `A(x,x,y)` has two identical but independent `x` components and one `y` component. When defining the molecule type, one must ensure that identically named components have identical sets of allowed states. For example, `A(x~a~b,x~a~b,y)` is a valid molecule type, but `A(x~a~b,x~b~c,y)` is invalid since the two `x` components have been assigned different sets of allowed states.

The *molecule* itself (created from the *molecule type* blueprint) can only take one state for each component at any given time. Therefore, given the molecule type `R(lig,ch~open~closed)`, the following molecules can be created: `R(lig,ch~open)` and `R(lig,ch~closed)`. In the traditional modeling framework, we might refer to them by names, such as R_o and R_c. The order in which the components are written within the molecule is immaterial and `R(lig,ch~open)` means the same as `R(ch~open,lig)`.

3.2. Bonds and Species

A BNG molecule can associate with another molecule through a *bond* between their components. Bonds can be formed between both stateless components and components with states. The two "bonded" components are indicated by an exclamation mark followed by a unique label that marks the two ends of the bond. A component cannot participate in more than one bond at any given time.

For example, the ligand molecule `L(rec)` and receptor molecule `R(lig,ch~open)` can associate using a bond between the `rec` and `lig` components, represented as `L(rec!0).R(lig!0,ch~open)`. The bond label "`0`" after the exclamation marks indicates that `rec` and `lig` are bound. The dot between `L()` and `R()` indicates that they are in the same complex. A good convention to follow is to label bonds with numbers. In the absence of identifying information about the structural domains or motifs involved, label components based on the molecule that they are supposed to bind.

The same bond between the same pair of components can be represented using different labels in different places as long as the right pair of components is matched, for example, `L(rec!0).R(lig!0,ch~open)` is the same as `L(rec!2).R(lig!2,ch~open)`.

In the traditional modeling framework, this complex may be named R_oL. The corresponding complex with the channel closed, i.e., `L(rec!0).R(lig!0,ch~closed)`, may be referred to as R_cL. By providing a systematic way of linking molecules together, BNGL enables the systematic representation of any arbitrary complex.

For example, the following is a valid complex:

```
A(x,y!0).B(p!0,q~abc!1).C(r~def!1,s!2).A(x!2,y)
```

This complex has two `A` molecules, one each of `B` and `C` molecules. One `A` binds `B` using a `y–p` bond labeled `0`. The other `A` binds `C` using an `x–s` bond labeled `2`. `B` binds `C` using a `q–r` bond labeled `1`. `q` and `r` are in `abc` and `def` states, respectively. Dots separate the molecule names and all bonds have unique labels. The order in which the molecules are written is immaterial as long as the same pairs of components are bonded and the same internal states are assigned. The same complex can also be written as:

```
B(p!0,q~abc!1).C(r~def!1,s!2).A(x!2,y).A(x,y!0)
```

Such structured representation provides an important advantage over the traditional modeling approach. The modeler does not need to assign a unique name to every complex configuration. The name space that the modeler has to comprehend is limited to the number of molecules and components and not the possibly infinite number of complex configurations. Where finite, the construction of the full set of configurations can be easily automated.

BNG uses the term *species* to refer to any unique configuration of one or more molecules in a complex. The critical elements of a unique species definition are:

- Identifying every molecule in the species (including repeated ones)
- Identifying internal states taken by every component (including repeated ones)
- Identifying every component pair linked by a bond

3.3. Transformations and Reactions

In the BNGL, a reaction is simply one or more transformations applied simultaneously to one or more species. The following transformations are allowed:

- Forming a bond, e.g. `A(b) + B(a) -> A(b!0).B(a!0)`
- Breaking a bond, e.g. `A(b!0).B(a!0)-> A(b)+B(a)`
- Changing of component state, e.g. `X(y~0) -> X(y~p)`
- Creating a molecule, e.g. `A(b) -> A(b) + C(d)`

- Destroying a molecule, e.g. `A(b) + B(a) -> A(b)`

A single reaction may involve any number of transformations. For example,

```
A(b) + B(a!0).C(d!0) -> A(b!0).B(a!0) + C(d)
```

Here, the `a–d` bond is broken and the `b–a` bond is formed simultaneously. Another example is:

```
A(b) + B(a!0).C(d!0) -> A(b!0).B(a!0)
```

Here, the `a–d` bond is broken, the `b–a` bond is formed, and the `C(d)` molecule is destroyed simultaneously. Usually, it is advisable to restrict the number of transformations in a reaction to one or two.

4. Writing Reaction Rules

4.1. Combining Multiple Reactions

Consider the ligand and receptor defined in the Subheading 3.1. Let us assume that ligand binding and channel opening/closing are independent of each other. This means that both open and closed forms of the receptor bind the ligand with the same rate constant k.

$$\begin{aligned} R_o + L &\rightarrow R_oL \quad k \\ R_c + L &\rightarrow R_cL \quad k. \end{aligned}$$

The ordinary differential equation (ODE) description for this set of reactions is:

$$-\frac{d[R_o]}{dt} = +\frac{d[R_oL]}{dt} = k[R_o][L].$$

$$-\frac{d[R_c]}{dt} = +\frac{d[R_cL]}{dt} = k[R_c][L].$$

$$-\frac{d[L]}{dt} = k[R_o][L] + k[R_c][L] = k([R_o] + [R_c])[L].$$

Since the rate constant k is common in the first two ODEs, we can add them up.

$$-\frac{d([R_o] + [R_c])}{dt} = +\frac{d([R_oL] + [R_cL])}{dt} = k([R_o] + [R_c])[L]$$

$$-\frac{d[L]}{dt} = k([R_o] + [R_c])[L].$$

Let us create two new terms:

$$[R_{o/c}] = [R_o] + [R_c]$$

$$[R_{o/c}L] = [R_oL] + [R_cL].$$

Using these two new terms, we can compress the original set of ODEs as:

$$-\frac{\mathrm{d}[\mathrm{R_{o/c}}]}{\mathrm{d}t} = -\frac{\mathrm{d}[\mathrm{L}]}{\mathrm{d}t} = +\frac{\mathrm{d}[\mathrm{R_{o/c}L}]}{\mathrm{d}t} = k[\mathrm{R_{o/c}}][\mathrm{L}].$$

Note that we have simply summed the binding rates for open and closed receptors. The fact that the two reactions have an identical rate constant k enables us to use the summed terms in a compressed manner. Representing this compressed description using traditional labels, we might write it as:

$$\mathrm{R_{o/c}} + \mathrm{L} \rightarrow \mathrm{R_{o/c}L} \quad k.$$

BNGL provides an intuitive way to formulate such compressed representations using reaction rules. In rule-based modeling, creating a new label such as $\mathrm{R_{o/c}}$ or $\mathrm{R_{o/c}L}$ is not necessary. A *reaction rule* can be created from many reactions (that have identical kinetics) simply by omitting the components that do not influence the reactions. For example, consider the same reactions represented in BNGL:

```
R(lig,ch~open)+L(rec)->\
  R(lig!0,ch~open).L(rec!0)k
R(lig,ch~closed)+L(rec)->\
  R(lig!0,ch~closed).L(rec!0)k
```

Here, the state of the `ch` component does not affect the rate constant for the `lig-rec` binding. Therefore, we can omit the `ch` component to create a *reaction rule* that represents *both* of the above reactions:

```
R(lig)+L(rec)->R(lig!0).L(rec!0)k
```

Thus, a reaction rule can be thought of as a generator of reactions, where each generated reaction must have some substructure that corresponds exactly to the reaction rule. The generated reactions have identical kinetic descriptions at the microscopic level.

Suppose the open and closed channels had two different rate constants of association with ligand, i.e.,

$$\mathrm{R_o} + \mathrm{L} \rightarrow \mathrm{R_oL} \quad k_1$$
$$\mathrm{R_c} + \mathrm{L} \rightarrow \mathrm{R_cL} \quad k_2.$$

Since the kinetics of the two reactions are fundamentally different, we *cannot* create a single reaction rule from these two reactions. We can only create two different reaction rules (which happen to be identical to the reactions):

```
R(lig,ch~open)+L(rec)->\
  R(lig!0,ch~open).L(rec!0)k1
R(lig,ch~closed)+L(rec)->\
  R(lig!0,ch~closed).L(rec!0)k2
```

Thus, identifying independent interactions is the key to writing a rule-based model. If within a molecule every component

influences every other component and if the same is true for all molecules, then the number of reaction rules is identical to the number of reactions. As more and more independent interactions are identified, the number of reaction rules required to represent a reaction network decreases.

Bidirectional reaction rules can be combined into a single line using bidirectional arrows and two reaction rate constants. However, BNG still interprets them as two different reaction rules. For example, the bidirectional reaction rule

```
R(ch~closed) <-> R(ch~open) k_open,k_close
```

will be interpreted as two reaction rules:

```
R(ch~closed) -> R(ch~open) k_open
R(ch~open) -> R(ch~closed) k_close
```

4.2. Patterns and Species

Consider the reaction rule

```
R(lig) + L(rec) -> R(lig!0).L(rec!0) k
```

As we already saw in the previous subsection, if the molecule types were `R(lig,ch~open~closed)` and `L(rec)`, then this reaction rule would generate the following reactions:

```
R(lig,ch~open) + L(rec) ->\
  R(lig!0,ch~open).L(rec!0) k
R(lig,ch~closed) + L(rec) ->\
  R(lig!0,ch~closed).L(rec!0) k
```

Note how the complexes are fully specified in the reactions, i.e., all the components of all the molecules present and all the corresponding binding and internal states are mentioned. As mentioned in the previous section, a fully specified complex definition is called a BNG *species*. In the reaction rule, however, the complexes are not fully specified. A partially specified complex is called a BNG *pattern*.

The reactants of a reaction rule are called the *reactant patterns* and they are said to match the *reactant species* of the reactions. The reactant pattern `R(lig)` matches the reactant species `R(lig,ch~open)` and `R(lig,ch~closed)`. Similarly, the reactant pattern `L(rec)` matches the reactant species `L(rec)`. Pattern matching to generate reactions is summarized in Fig. 1.

Patterns lie at the heart of rule-based modeling. Patterns can be said to select species that possess a matching substructure. *A reaction rule is effectively a set of reactant patterns and a set of transformations applied to those patterns.* A reaction is generated by selecting species that match the reactant patterns and applying the transformation to them to get the product species.

4.3. Observables

Many biological experiments involve following the time courses of experimental outputs, often referred to as *trajectories*. Each output is usually specific to a single collection of species. For example, in a FRET experiment on receptor aggregation, one might design the

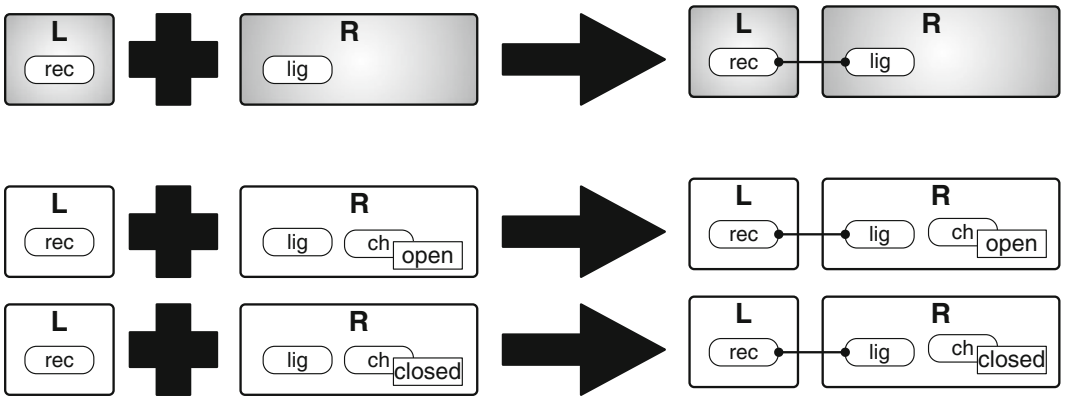

Fig. 1. Pattern matching to generate reactions. Patterns are incompletely defined species, such as those in *gray*. The pattern is said to match to a species if the species possesses a substructure that is equivalent to the pattern. In this case, the pattern R(lig) matches the species R(lig,ch~open) and R(lig,ch~closed). Reactions on patterns are called reaction rules. When the patterns match to full species, reaction rules match to reactions. The reactions all are identical with respect to their microscopic kinetics, i.e., they have identical per-site asymmetric rate constants. The symmetry and multiplicity factors are automatically detected and assigned by BioNetGen. In this case, the reaction is the formation of a bond between the lig component of R and the rec component of L.

experiment such that only dimers can produce fluorescence, and the fluorescent output is a function of the total concentration of all dimer species. Similarly, in a Western blot experiment using an antibody specific to a particular protein, the measurement at different time points is proportional to the total concentration of the protein available for binding. In a BNG model, one can similarly define specific sums of concentrations of species that are of interest and need to be monitored or tracked through the simulation of the model. Such sums are called *observables* and are of two types: molecules-observables and species-observables.

Consider the molecule types `A(b,b,c)`, `B(a)`, and `C(a)`. A can bind two B and one C. This results in the following species:

B	`B(a)`
C	`C(a)`
A	`A(b,b,c)`
AB	`A(b!0,b,c).B(a!0)`
BAB	`A(b!0,b!1,c).B(a!0).B(a!1)`
AC	`A(b,b,c!2).C(a!2)`
ABC	`A(b!0,b,c!2).B(a!0).C(a!2)`
BABC	`A(b!0,b!1,c!2).B(a!0).B(a!1).C(a!2)`

A *species-observable* is an unweighted sum of the concentrations of species matched to a pattern (or a list of patterns). For example, let us define a species-observable to be the pattern `A(b)`. This pattern matches any species that contains an `A` molecule with an unbound `b` component. The species A and AC have two unbound b components each, and the species AB and ABC have one unbound b component each. Thus, the observable would match all four of these species (but not the species BABC). The value of this

observable would be given by the sum [A] + [AB] + [AC] + [ABC], which would vary over the course of a simulation.

A *molecules-observable* weights the concentrations of the species by the number of matches to a pattern (or a list of patterns). For example, let us define a molecules-observable with the same pattern `A(b)`. This would match species AB and ABC once each and the species A and AC twice (since they have two unbound `b` sites). The value of this observable would be given by the sum $2 \times [A] + 2 \times [AC] + 1 \times [AB] + 1 \times [ABC]$.

4.4. Reaction Center and Context

Consider the reaction rule:

```
R(lig) + L(rec) -> R(lig!0).L(rec!0) k
```

The transformation applied in this rule is the creation of the `lig-rec` bond. The components that participate in the transformation(s) carried out by a reaction are collectively called the *reaction center*. Here, the reaction center is the set of components `lig` and `rec`.

Now, consider another reaction rule:

```
R(lig,ch~open) + L(rec) ->\
  R(lig!0,ch~open).L(rec!0) k1
```

Here, the same transformation is being applied, i.e., the `lig-rec` bond, and therefore it has the same reaction center (underlined for emphasis). However, the additional `ch` component in the `open` state is required to indicate that if the channel is open then the reaction rate should be k_1. The components that do not participate in the reaction, but are nevertheless required and influence the reaction rate constant, are collectively called the *reaction context*.

Finally, consider the reaction rule:

```
R(lig,ch~closed) + L(rec) ->\
  R(lig!0,ch~closed).L(rec!0) k2
```

This rule also has a reaction center identical to the previous rules (underlined for emphasis). However, the reaction context is now `ch~closed` and the rate constant is k_2.

When a reaction rule is used to generate reactions, the reaction context provides minimum compulsory conditions that need to be obeyed by the generated reactions. In the absence of any reaction context, the reaction rule is general and matches all possible reactions having the same transformation. Adding more context "specializes" the reaction rule to match fewer reactions that obey the conditions imposed by the context. The reaction itself can be considered as a highly specialized reaction rule with every component in the interacting species added as context.

Note that in bidirectional reaction rules, the context is preserved in both directions. For example, consider the bidirectional rule (with reaction center underlined for emphasis):

```
R(lig,ch~closed)+L(rec)<->\
  R(lig!0,ch~closed).L(rec!0)k2,kr
```

This is equivalent to the two reaction rules, both with `ch~closed` as context:

```
R(lig,ch~closed)+L(rec)->\
  R(lig!0,ch~closed).L(rec!0)k2
R(lig!0,ch~closed).L(rec!0)->\
  R(lig,ch~closed)+L(rec)kr
```

4.5. Synthesis and Degradation

When writing a reaction rule in BNGL, one must make sure that there is at least one reactant pattern and one product pattern. However, there are a number of situations, where the number of molecules on either the reactant side or the product side of the reaction is zero. To indicate zero stoichiometry, one can use the symbol 0 (zero).

When the reactant side of a reaction rule has zero stoichiometry, the reaction is treated as a zero-order synthesis reaction, for example:

```
0->L(rec)k_syn
```

This reaction rule should produce the ligand molecule at a constant zero-order rate, which is equal to `k_syn` (typical units M/s). Zero-order synthesis is useful to model the case, where a reacting agent is continuously flowing into the system from the outside, for example the continuous movement of growth factors from the blood to any body tissue or the controlled flow of nutrients in a bioreactor.

Where the synthesis depends on the concentrations of specific molecules, it is better to use first-order and second-order elementary reactions. For example, the production of mRNA from a gene depends on the number of actively transcribing copies of the gene in a cell:

```
Gene()->Gene()+mRNA()k_transcription
```

When the product side of a reaction rule has zero stoichiometry, it has no bearing on the rate of the reaction since the rate is determined only by the reactant concentrations. Such reactions are typically used to model spontaneous or background degradation, which occurs in some form or the other with most biomolecules. For example, to model spontaneous degradation of the receptor:

```
R()->0k_degr
```

The internal and binding states of components can be mentioned to restrict the degradation to only certain types of molecules. For example, if the only receptors degraded were those in the closed state:

```
R(ch~closed)->0k_degr
```

It is important to note that in a degradation reaction rule, the *complete matched reactant species* is degraded and not just the molecules mentioned in the rule. The above reaction rule will generate the following reactions:

```
R(lig,ch~closed) -> 0 k_degr
R(lig!0,ch~closed).L(rec) -> 0 k_degr
```

To delete only the molecules concerned (and not the whole species), use the `DeleteMolecules` keyword:

```
R(ch~closed) -> 0 k_degr DeleteMolecules
```

In this case, the connected molecules are not deleted and this would generate the following reactions:

```
R(lig,ch~closed) -> 0 k_degr
R(lig!0,ch~closed).L(rec) -> L(rec) k_degr
```

4.6. Symmetry, Multiplicity, and Rate Addition

Consider the following situation: The A molecule can dimerize. It exists in two different isoforms A_1 and A_2 and both hetero- and homo-dimerization occur. Expressing this in the traditional framework, this implies the existence of three different reactions:

$$\begin{aligned} A_1 + A_2 &\rightarrow D_{12} \\ A_1 + A_1 &\rightarrow D_{11} \\ A_2 + A_2 &\rightarrow D_{22}. \end{aligned}$$

Even if both hetero- and homo-dimerization have identical interaction strengths (affinities), the homo-dimerization would proceed at half the rate of the hetero-dimerization due to the symmetry in picking out a homo-dimerizing molecule pair. Therefore, the rate constants are usually expressed as:

$$\begin{aligned} A_1 + A_2 &\rightarrow D_{12} \quad k \\ A_1 + A_1 &\rightarrow D_{11} \quad k/2 \\ A_2 + A_2 &\rightarrow D_{22} \quad k/2. \end{aligned}$$

Here, one can think of "k" as the asymmetric reaction rate constant and the $(1/2)$ multiplying factor as being due to the *symmetry effect*. Expressing these reactions using BNG species and representing the two isoforms as two different states:

```
A(iso~a1) + A(iso~a2) ->\
  A(iso~a1!0).A(iso~a2!0) k
A(iso~a1) + A(iso~a1) ->\
  A(iso~a1!0).A(iso~a1!0) 0.5*k
A(iso~a2) + A(iso~a2) ->\
  A(iso~a2!0).A(iso~a2!0) 0.5*k
```

BNG automatically detects reaction symmetries and applies the symmetry factor even if the reaction rules themselves are asymmetric. When writing a reaction rule, the modeler must always provide the

asymmetric reaction rate constant. All three reactions above can be generated with the correct reaction rate constants using only the single rule:

```
A(iso!0) + A(iso!0) -> A(iso!0).A(iso!0) k
```

The states of the `iso` component are left unmentioned in the reaction rule since they do not affect the asymmetric reaction rate constant. BNG detects that some of the generated reactions are symmetric and some are not. It then automatically assigns the $(1/2)$ multiplying factor to the symmetric reactions.

The *multiplicity effect* is seen when interactions are multivalent. Suppose the molecule A has two identical sites for binding the molecule B. Then, in the traditional framework, one would write the following reactions:

$$\begin{aligned} \mathrm{A} + \mathrm{A} &\rightarrow \mathrm{AB} \\ \mathrm{AB} + \mathrm{B} &\rightarrow \mathrm{BAB}. \end{aligned}$$

Even though the two binding sites have identical and independent chemical interactions, the first reaction should proceed at twice the rate of the second because A has twice the number of binding sites as AB. This is usually expressed as

$$\begin{aligned} \mathrm{A} + \mathrm{A} &\rightarrow \mathrm{AB} \quad & 2 \times k \\ \mathrm{AB} + \mathrm{B} &\rightarrow \mathrm{BAB} \quad & \mathrm{k}. \end{aligned}$$

Since BNGL allows multiple identical components in a molecule, it automatically accounts for the multiplicity factor when generating the reactions. The modeler must always provide the *per-site reaction rate constant*. If the molecule types are `A(b,b)` and `B(a)`, then it is sufficient to write the following reaction rule:

```
A(b) + B(a) -> A(b!0).B(a!0) k
```

This would automatically generate the following reactions:

```
A(b,b) + B(a) -> A(b,b!0).B(a!0) 2*k
A(b,b!0).B(a!0) + B(a) -> A(b!1,b!0).B(a!0).B(b!1) k
```

Lastly, if multiple reaction rules generate two versions of the same reaction, both are kept in the reaction network because they may have different rate constants or rate laws. During simulation, the net effect of the two identical reactions is additive, i.e., the rate of the transformations proceeds using the sum of the rates of the two copies. This is important when considering events such as dephosphorylation that might occur through general and specific mechanisms. For example, consider the following set of rules:

```
A(y~p) -> A(y~0) k_bkgrnd
A(y~p).E() -> A(y~0).E() k_enzyme
```

The second rule requires complexation with E for dephosphorylation to occur, whereas the first does not. These rules will generate two copies of the same reaction with different rate constants:

```
A(y~p,e!1).E(a!1) -> A(y~0,e!0).E(a!1) k_bkgrnd
A(y~p,e!1).E(a!1) -> A(y~0,e!0).E(a!1) k_enzyme
```

During simulation, the effect of these two reactions on the species `A(y~p,e!1).E(a!1)` is a unimolecular transformation with a net rate constant equal to the sum `k_bkgrnd+k_enzyme`.

4.7. Wildcards and Context Modification

BNGL provides syntax to tailor the context of a reaction rule to only match specific sets of reactions. The absence of an exclamation mark indicates that a component *must* be unbound. For example, the pattern `A(b,c)` will *not* match the species `A(b,c!1).C(a!0)`.

If the presence of a bond is contextually important but the identity of the binding partner is not relevant, then the + wildcard is used with the bond symbol. For example, the pattern `A(b!+,c)` will match both `A(b!0,c).B(a!0)` and `A(b!0,c).P(q!0)` and any other species with a bound `b` component and an unbound `c` component.

To match both unbound and bound components, one can use the `?` wildcard with the bond symbol. For example, the pattern `A(b!?,c)` matches both species `A(b,c)` and `A(b!0,c).B(a!0)`.

If the internal state of a component is not relevant to the context, the `~state` definition can be omitted. The presence or absence of a `!` symbol is still used for context. For example, the pattern `R(s)` matches both species `R(s~x1)` and `R(s~x2)` but not the species `R(s~x1!0).S(r!0)`.

The dot symbol can be used to indicate the presence of another molecule in a complex even if the explicit bond links are not shown. For example, the pattern `C().A(b)` will match both species `C(x!0).A(b,c!0)` and `C(x!0).A(b,c!1).X(a!1,c!0)`, even though `C` is connected to `A` directly in the first species and indirectly through `X` in the second species. This is useful when the same molecule can be linked to a complex in different ways but performs the same function, for example a kinase.

The bond and state wildcards, as well as the dot operator, can be used to tailor patterns in observables as well as reaction rules.

In reaction rules, more flexibility in defining context is provided by the `include_[reactants/products]` and `exclude_[reactants/products]` keywords. For example, consider the reaction rule

```
A(b) + B(a) -> A(b!0).B(a!0) k_ab\
  exclude_reactants(1,E,F(g))
```

The number 1 in the `exclude_reactants` directive indicates that it applies to the *first* reactant pattern. The remaining elements within the parentheses indicate the additional context that will cause exclusion. Here, it means that if the rule matches reactants with `E` or `F(g)` in the reactant 1 position, then those reactants would be discarded. Therefore, the following reactants would be *excluded*:

```
A(b,x!1).E(a!1)
A(b,x!1).F(a!1,g)
```

However, since only an unbound `F(g)` component is excluded, a reactant with a bound `F(g)` component (underlined for emphasis) would not be excluded, for example:

```
A(b,x!1).F(a!1,g!2).G(f!2)
```

The `include_reactants()` works similarly to the `exclude_reactants()`, but here the condition imposed is that *all* the additional context must be included.

For example, consider the following reaction rule:

```
A(b)+B(a)->A(b!0).B(a!0) k_ab\
  include_reactants(2,E,F,G)
```

This rule is equivalent to the reaction rule

```
A(b)+B(a).E().F().G()->A(b!0).B(a!0).E().F().G() k_ab
```

Both rules are different ways of saying that `E()`, `F()`, and `G()` molecules should be present in the second reactant species. Note that for bidirectional reaction rules `[include/exclude]_reactants()` is treated as `[include/exclude]_products` for the reaction rule in the reverse direction.

5. Modeling Background

5.1. Law of Mass Action

By default, BNG assigns rates to reactions based on the Law of Mass Action for Elementary reactions.

This means that for a unimolecular reaction

$$\mathrm{A} \rightarrow \text{Products} \quad k_{\text{uni}},$$

the rate is determined by the concentration of the reactant as follows:

$$\text{Rate } r = -\frac{\mathrm{d}[\mathrm{A}]}{\mathrm{d}t} = k_{\text{uni}}[\mathrm{A}] \text{ M/s}.$$

Similarly, for a bimolecular reaction,

$$\mathrm{A} + \mathrm{B} \rightarrow \text{Products} \qquad k_{\text{bi}}$$
$$\text{Rate } r = -\frac{\mathrm{d}[\mathrm{A}]}{\mathrm{d}t} = -\frac{\mathrm{d}[\mathrm{B}]}{\mathrm{d}t} = k_{\text{bi}}[\mathrm{A}][\mathrm{B}] \text{ M/s}.$$

The reaction rate here is expressed in volume-independent *macroscopic* terms, i.e., molar per second. But to aid modeling flexibility and simulation, it is preferable to specify rate constants such that the reaction rate is given in reaction-events/second. This *microscopic* rate is related to the macroscopic rate by a factor of the volume in which the reaction occurs times Avogadro's number.

$$r = \frac{\mathrm{d}[\mathrm{A}]}{\mathrm{d}t} = \frac{\mathrm{d}(N_{\mathrm{A}}/(VN_{\text{Avo}}))}{\mathrm{d}t} = \frac{1}{VN_{\text{Avo}}} \times \frac{\mathrm{d}N_{\mathrm{A}}}{\mathrm{d}t} = \left(\frac{1}{VN_{\text{Avo}}}\right) r'$$
$$\Rightarrow r' = r \times VN_{\text{Avo}},$$

where r = macroscopic rate constant (per M/s) and r' = microscopic rate constant (per second).

k_{uni} and k_{bi} are unimolecular and bimolecular reaction rate constants, respectively, with units per second and per molar per second. In general, any nth order volume-independent *macroscopic* reaction rate constant should have the units molar^{1-n} per second. However, since BNG treats the rate as microscopic, the modeler must convert the macroscopic rate constants to microscopic ones, as follows.

For unimolecular reactions, if N_{A} denotes the population number of the reactant molecules, then the microscopic rate is

$$r'\,(\text{s}^{-1}) = r \times VN_{\text{Avo}} = k_{\text{uni}}[\text{A}] \times VN_{\text{Avo}} = k_{\text{uni}}\left(\frac{N_{\text{A}}}{VN_{\text{Avo}}}\right) \times VN_{\text{Avo}}$$

$$r' = k_{\text{uni}} N_{\text{A}}.$$

For bimolecular reactions, if N_{A} and N_{B} denote the population numbers of the two participating reactants, then the microscopic rate is

$$\begin{aligned} r'\,(s^{-1}) &= r \times VN_{\text{Avo}} = k_{\text{bi}}[\text{A}][\text{B}] \times VN_{\text{Avo}} \\ &= k_{\text{uni}}\left(\frac{N_{\text{A}}}{VN_{\text{Avo}}}\right)\left(\frac{N_{\text{B}}}{VN_{\text{Avo}}}\right) \times VN_{\text{Avo}}. \\ r' &= \left(\frac{k_{\text{bi}}}{VN_{\text{Avo}}}\right) N_{\text{A}} N_{\text{B}}. \end{aligned}$$

If we define the microscopic reaction rate constants k'_{uni} and k'_{bi} as

$$\begin{aligned} &\text{For unimolecular reactions, } r' = k'_{\text{uni}} N_{\text{A}} \\ &\text{For bimolecular reactions, } r' = k'_{\text{bi}} N_{\text{A}} N_{\text{B}}, \end{aligned}$$

then the relationship between macroscopic and microscopic rate constants is given by

$$k'_{\text{uni}} = k_{\text{uni}}.$$

$$k'_{\text{bi}} = \frac{k_{\text{bi}}}{VN_{\text{Avo}}}.$$

In general,

$$k'_{n\text{thorder}} = \frac{k_n}{(VN_{\text{Avo}})^{n-1}}, \quad n \geq 1.$$

Thus, when writing a unimolecular reaction rule in BNGL, one does not need to worry about converting the unimolecular reaction rate constant as long as it is in per second:

```
A(x~on) -> A(x~off) k_uni
```

where `k_uni` is defined as a parameter with units per second.

When writing a bimolecular reaction rule in BNGL, however, one should convert the bimolecular reaction rate constant to per second:

```
A(b) + B(a) -> A(b!0).B(a!0) k
```

where the parameter `k` can be defined by an expression `k_bi/(V*N_Avo)` involving other parameters: `k_bi` with units molar per second, `V` with units liters, and `N_Avo` is defined as 6.022×10^{23}/mol. If bimolecular and higher order reactions occur in different volumes, then the reaction rate constant needs to be scaled differently for the reaction occurring in each volume. The modeler has to explicitly include the volume in the rate constant or use compartmental BNG (cBNG), as described in Subheading 7. Unimolecular reactions do not need to be scaled by volume.

The following assumptions are made under this paradigm: The reaction occurs in a fixed volume reactor at constant temperature and pressure and the molecules are distributed uniformly through the volume, i.e., there are no spatial gradients. Although these seem to be strong assumptions, they are often reasonable for biological models of events on the plasma membrane and in the cytosol (13). Diffusion in these compartments is often sufficiently fast that spatial effects remain minimal.

5.2. Reaction Order and Transformation Reversibility

Reaction orders higher than two are rare at the molecular level since the probability of three or more freely diffusing molecules finding each other simultaneously and in the right orientation is small. Although BNG supports reactions of any nonnegative integer order, it is prudent to restrict the model to a set of unimolecular and bimolecular reactions (i.e., first- and second-order reactions) in order to maintain physical realism and keep model assumptions to a minimum.

It is advisable to model every associative complex-forming interaction as a bimolecular association reaction between a pair of reactants. Also, in the context of protein–protein and protein–ligand associations, most association reactions are due to the cumulative effect of many weak non-covalent interactions that can be broken and re-formed. Therefore, unless an interaction is specifically known to be irreversible, it is recommended that every bimolecular association reaction be paired with the complementary unimolecular dissociation reaction. Such complementary modeling is advisable for all transformation pairs, such as synthesis/degradation and phosphorylation/dephosphorylation.

Such pairing is important to prevent system blowup or depletion. This happens when a unidirectional transformation is initiated and all the relevant molecules in the system get transformed in bulk, regardless of the magnitude of the initial stimulus. This is usually not intended or a reasonable biological behavior. For example, synthesis alone will lead to artificially high numbers and

degradation alone will lead to complete depletion of a molecule type or a set of molecule types. Similarly, modeling phosphorylation alone will lead to runaway phosphorylation of the entire system (instead of a graded response) unless paired with dephosphorylation mechanisms.

5.3. Catalysis and Sigmoidal Kinetics

5.3.1. Michaelis–Menten Kinetics

The kinetic formulation of a catalytic process dates back to Michaelis and Menten (14) and Briggs and Haldane (15). The enzyme/catalyst binds to the substrate in a reversible fashion. A fraction of the bound complex causes the substrate to be converted to the product and dissociate simultaneously in a unimolecular fashion. The canonical formulation is

$$\mathrm{E} + \mathrm{S} \underset{k_\mathrm{f}}{\overset{k_\mathrm{r}}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{K_\mathrm{cat}} \mathrm{E} + \mathrm{P}.$$

Applying reaction kinetic theory, reaction flux through ES is given by

$$\frac{\mathrm{d}[\mathrm{ES}]}{\mathrm{d}t} = k_\mathrm{f}[\mathrm{ES}] - (k_\mathrm{r} + k_\mathrm{cat}[\mathrm{E}][\mathrm{P}]).$$

Applying a quasi-steady-state approximation on [ES] and using mass conservation laws,

$$\frac{\mathrm{d}[\mathrm{ES}]}{\mathrm{d}t} \approx 0$$
$$\Rightarrow [\mathrm{ES}] \approx \frac{[\mathrm{E}]_\mathrm{T}[\mathrm{S}]}{\left(\frac{k_\mathrm{r}+k_\mathrm{cat}}{k_\mathrm{f}}\right) + [\mathrm{S}]},$$

where $[\mathrm{E}]_\mathrm{T}$ is total enzyme concentration (i.e., [E] + [ES]) which can be assumed to be constant.

The rate of conversion of substrate (and formation of product) is then given by

$$r = -\frac{\mathrm{d}[\mathrm{S}]}{\mathrm{d}t} \approx +\frac{\mathrm{d}[\mathrm{P}]}{\mathrm{d}t} \approx k_\mathrm{cat}[\mathrm{ES}] = \frac{k_\mathrm{cat}[\mathrm{E}]_\mathrm{T}[\mathrm{S}]}{\left(\frac{k_\mathrm{r}+k_\mathrm{cat}}{k_\mathrm{f}}\right) + [\mathrm{S}]} = \frac{V_\mathrm{max}[\mathrm{S}]}{K_\mathrm{M} + [\mathrm{S}]},$$

where the Michaelis constant $K_\mathrm{M} = (k_\mathrm{r} + k_\mathrm{cat})/k_\mathrm{f}$ and maximum rate $V_\mathrm{max} = k_\mathrm{cat}[\mathrm{E}]_\mathrm{T}$.

According to this formulation, the reaction rate should saturate when there is excess substrate, i.e.,

$$[\mathrm{S}] \gg K_\mathrm{M} \Rightarrow r \approx V_\mathrm{max} = k_\mathrm{cat}[\mathrm{E}]_\mathrm{T}.$$

Note that the rate is still first order with respect to enzyme concentration. The Michaelis–Menten rate law can be thought of as an extension to the elementary rate law, where the microscopic rate "constant" is not really a constant but a function that depends on the concentration of the substrate, i.e.,

$$r = \frac{k_{\text{cat}}[\text{E}]_{\text{T}}[\text{S}]}{K_{\text{M}} + [\text{S}]} = \text{f}(\text{S}) \times [\text{E}]_{\text{T}}, \quad \text{where } \text{f}(\text{S}) = \frac{k_{\text{cat}}[\text{S}]}{K_{\text{M}} + [\text{S}]}.$$

Note that the [S] term in the reaction indicates free substrate concentration which is equal to the total substrate concentration minus the substrate sequestered with the enzyme, i.e., $[\text{S}]_{\text{T}} - [\text{ES}]$. Often, the [ES] term is quite small relative to $[\text{S}]_{\text{T}}$ and can be neglected, i.e.,

$$[\text{ES}] \ll [\text{S}]_{\text{T}} \Rightarrow [\text{S}] \approx [\text{S}]_{\text{T}} \Rightarrow \text{f}(\text{S}) \approx \frac{k_{\text{cat}}[\text{S}]_{\text{T}}}{K_{\text{M}} + [\text{S}]_{\text{T}}}.$$

As a shortcut, for a single enzyme–substrate pair, the Michaelis–Menten rate law can be implemented in BNGL using the `MM` or `Sat` keywords. The substrate is given as the first reactant and the enzyme is given as the second reactant:

```
S(x~0) + E() -> S(x~p) + E() MM(kcat,KM)
```

or

```
S(x~0) + E() -> S(x~p) + E() Sat(kcat,KM)
```

The `MM` rate law calculates the free substrate concentration [S] and is more accurate than the `Sat` rate law which approximates it as the total substrate concentration $[\text{S}]_{\text{T}}$.

5.3.2. General Sigmoidal Kinetics

In general, if there are multiple reactants and it is required to saturate the rate with respect to a *single* reactant, one can use the `Sat` keyword. The saturating reactant is provided as the first reactant. The rate is elementary with respect to all the other reactants that follow. For example, if the reaction rule is

```
S(x~0) + A() + B() + C() -> S(x~p) + A() + B() + C()
  Sat(k,K)
```

then the rate of the reaction is calculated as

$$r = \text{f}(\text{S}) \times [\text{A}][\text{B}][\text{C}], \quad \text{where } \text{f}(\text{S}) = \frac{k[\text{S}]}{K + [\text{S}]}.$$

The `Sat` is a special case of the Hill rate law, another common approximation used to model cooperative phenomena (72). The Hill rate law can be employed in BNGL using the `Hill` keyword.

```
S(x~0) + A() + B() + C() -> S(x~p) + A() + B() + C()
  Hill(k,K,n)
```

The rate is calculated as follows:

$$r = \text{f}(\text{S}) \ \times \ [\text{A}][\text{B}][\text{C}], \ \text{where } \text{f}(\text{S}) = \frac{k[\text{S}]^n}{K^n + [\text{S}]^n}.$$

f(S) is a sigmoidal function which exhibits a switch-like behavior, moving from zero to maximum over a range of [S]. Figure 2 shows

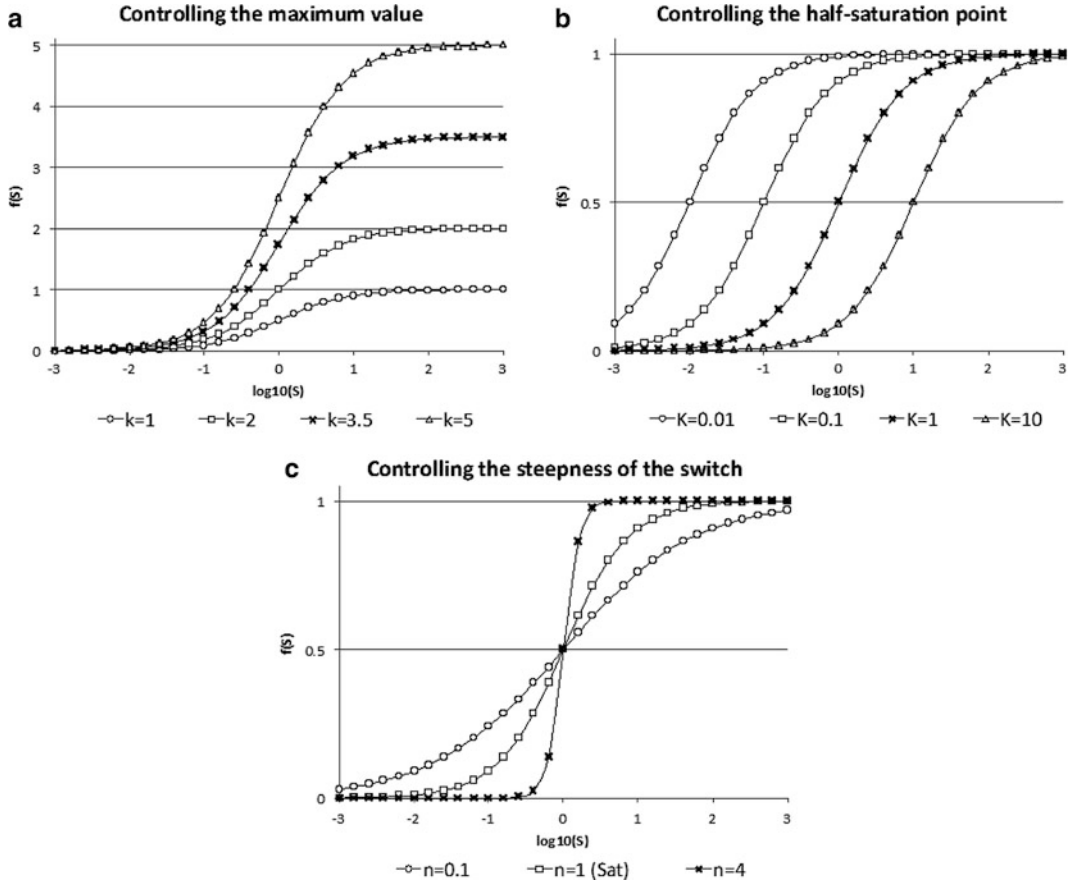

Fig. 2. Use of Hill/Sat rate laws. The Hill rate law function $f(S)$ is plotted for different parameter sets. (**a**) $K = 1$, $n = 1$. k is varied from 1 to 5 and it controls the maximum value attained at high concentrations. (**b**) $k = 1$, $n = 1$. K is varied from 0.01 to 10 and it controls the point at which half the maximum value is attained. (**c**) $K = 1$, $k = 1$. n is varied from 0.1 to 4 and it controls the steepness of the switch from zero to maximum. $n = 1$ is the special case of the Sat rate law. n is also called the Hill coefficient.

how the function f(S) is controlled by the three parameters. The k parameter controls the maximum value that f(S) tends to at high [S] values. The K parameter controls the half-saturation point, i.e., the [S] value for which f(S) is half of its maximum value. The n parameter is also called the Hill coefficient and controls the steepness of the switching behavior. The Hill coefficient should be a positive number, but does not have to be an integer.

Note that both `Sat` and `Hill` rate laws are first order with respect to the other reactants that follow the first reactant. `Sat` is a special case of the `Hill` rate law with the Hill coefficient set at $n = 1$.

5.3.3. Rate Law Approximations Versus Explicit Mechanisms

The advantages of using the `Sat`, `MM`, and `Hill` keywords are that they do not introduce new species into the network for intermediate complexes. A sequence of kinetic steps is approximated into a single-step rate law using these approximations. In the traditional

framework, such approximations are typically employed to reduce the size of the network. For enzymatic interactions that are fairly independent, a significant reduction is achieved and the fast timescale of the enzyme–substrate equilibrium is removed, which can speed up model simulation. However, using a rate law approximation such as Michaelis–Menten carries the risk of modeling error, especially in densely connected catalytic networks such as signal transduction. For example, if there were two substrates that competed for binding to a single enzyme, but with identical reaction parameters, i.e.,

$$\mathrm{E} + \mathrm{S}_1 \underset{k_\mathrm{r}}{\overset{k_\mathrm{f}}{\rightleftharpoons}} \mathrm{ES}_1 \xrightarrow{k_\mathrm{cat}} \mathrm{E} + \mathrm{P}_1,$$

$$\mathrm{E} + \mathrm{S}_2 \underset{k_\mathrm{r}}{\overset{k_\mathrm{f}}{\rightleftharpoons}} \mathrm{ES}_2 \xrightarrow{k_\mathrm{cat}} \mathrm{E} + \mathrm{P}_2,$$

then the MM-like formulation for conversion for each substrate can be derived to be

$$-\frac{\mathrm{d}[\mathrm{S}_1]}{\mathrm{d}t} = \frac{k_\mathrm{cat}[\mathrm{E}]_\mathrm{T}[\mathrm{S}_1]}{K_\mathrm{M} + [\mathrm{S}_1] + [\mathrm{S}_2]}.$$

$$-\frac{\mathrm{d}[\mathrm{S}_2]}{\mathrm{d}t} = \frac{k_\mathrm{cat}[\mathrm{E}]_\mathrm{T}[\mathrm{S}_2]}{K_\mathrm{M} + [\mathrm{S}_1] + [\mathrm{S}_2]}.$$

Note how the denominator term for each conversion has both $[\mathrm{S}_1]$ and $[\mathrm{S}_2]$ terms. This cannot be expressed using a conventional `MM` keyword. We can get around this by manually deriving the rate function for such interactions (such as the relations above) and then assigning to the rate "constant" a custom global function (see Subheading 6.1). The concentration terms in the function need to be provided as molecules-observables. For example, one of the relations above can be rephrased as

$$-\frac{\mathrm{d}[\mathrm{S}_1]}{\mathrm{d}t} = \left(\frac{k_\mathrm{cat}}{K_\mathrm{M} + [\mathrm{S}_1] + [\mathrm{S}_2]}\right) \times [\mathrm{E}]_\mathrm{T} \times [\mathrm{S}_1] = k_\mathrm{func}[\mathrm{E}]_\mathrm{T}[\mathrm{S}_1].$$

k_func can be assigned a global function involving the parameters k_cat, K_M, and a molecules-observable which returns the sum $[\mathrm{S}_1] + [\mathrm{S}_2]$.

Such custom rate laws may be useful for simplifying reaction mechanisms and speeding up simulations, but they come at a significant price in terms of model clarity and modeler effort to derive or find the appropriate reduced model. Many reduced mechanisms are mathematically quite complex and may make the resultant model inaccessible to the average biological researcher. Common situations in signal transductions that can be modeled using such reduced mechanisms include (16):

- Multiple competitive substrates with different enzymatic parameters
- Inhibitors with one or more mechanisms (competitive, non-competitive, etc.)

- Combinations of inhibitors and allosteric enhancers of enzymatic activity
- Multiple enzymes acting on a common substrate
- Allosteric inhibition/enhancement caused by substrate or product feedback

More importantly, the approximations used to derive these reduced mechanisms may not hold in the context of a large reaction network. For example, the quasi-steady-state approximation may not be appropriate when changes in [ES] are of the same order of magnitude as changes in [S], i.e., when $d[ES]/dt \ll d[S]/dt$ is false (16). This may happen in signal transduction networks because the enzyme and substrate are often of similar concentrations (17). A second example where the MM approximation would fail is when the product is already present in significant amount in the system making the ES → E+P catalytic step a reversible process (16, 18). For some enzymes, the catalytic flux may even shift to flow in the opposite direction, converting product to substrate. Recent studies have shown that the repeated invocation of the Michaelis–Menten mechanism can cause a significant loss of model accuracy for dense biological networks (17, 19).

An alternative approach is simply to write out the explicit reaction mechanisms and use no higher order approximations, except the most basic Law of Mass Action (see Subheading 5.1). Although more accurate than the reduced mechanisms, the exact treatment can potentially increase the size of the reaction network by introducing a large number of intermediary complexes. In the traditional modeling framework, this approach was unfeasible because every complex had to be identified and labeled manually. *BNGL and other rule-based languages alleviate this problem by providing a structured way to create large numbers of complexes using relatively few reaction rules.*

Although rule-based languages simplify the task of coding large networks, they do not, by themselves, avoid the computational costs associated with simulating large networks using standard methods for chemical kinetics. However, theoretical advances in simulation methods for rule-based models (see Subheading 5.4) have produced network-free methods, which has computational costs that are independent of network size (20, 25, 30). Recent attempts at modeling large signal transduction systems have embraced the elementary reaction perspective (21, 22) for its accuracy. For a sufficiently large (or even infinite) network, instead of trying to compress the reactions using potentially inaccurate rate law approximations, it may be better to write down the full reaction mechanisms and use a network-free simulator (see Subheading 5.4) to simulate the system.

To illustrate, consider the example of two enzymes E1 and E2, both of which can act on two different substrates S1 and S2.

Assume that every enzyme–substrate pair has unique enzymatic parameters. Writing down the reaction rules to describe this situation takes a matter of minutes, whereas deriving the *correct* rate law approximation would take much longer, and the computational cost would likely not be reduced, especially for ODE-based simulations (see Subheading 5.4). The system can be explicitly written out as

```
E1(s) + S1(e,x~0) <-> E1(s!0).S1(e!0,x~0) kf11,kr11
E1(s) + S2(e,x~0) <-> E1(s!0).S2(e!0,x~0) kf12,kr12
E2(s) + S1(e,x~0) <-> E2(s!0).S1(e!0,x~0) kf21,kr21
E2(s) + S2(e,x~0) <-> E2(s!0).S2(e!0,x~0) kf22,kr22
E1(s!0).S1(e!0,x~0) -> E1(s) + S1(e,x~p)  kcat11
E1(s!0).S2(e!0,x~0) -> E1(s) + S2(e,x~p)  kcat12
E2(s!0).S1(e!0,x~0) -> E2(s) + S1(e,x~p)  kcat21
E2(s!0).S2(e!0,x~0) -> E2(s) + S2(e,x~p)  kcat22
```

5.4. Simulation Methods

Since chemical kinetic theory provides a formal way of defining network dynamics in terms of ODEs (such as those used in Subheading 4.1), the simplest way to simulate a chemical kinetic model is to initiate the model with certain concentrations of starting species and then use *numerical integration of ODEs* to propagate the model through time. The chemical system is assumed to exist in an isothermal reactor of constant volume with the molecules homogenously distributed throughout the system and freely diffusing. Also, the notion of the individual molecule does not exist and the concentration changes in the trajectories are "smooth," continuous, and deterministic.

The ODE integration method is fast and works well when the system is well behaved and concentrations are sufficiently large such that variations are smooth. However, when concentrations are small—on the order of tens to hundreds of molecules—stochastic noise can play a significant role. For example, turning a gene off or on can have a drastic effect on the synthesis of the corresponding protein and the timing of gene activation may be important in a gene regulation model. The continuum approximation of the ODE method is not useful in this case, since the behavior is not smooth and the notion of individual molecules becomes more important. Stochastic noise can introduce dynamic behavior that is not observable at the continuous limit, but is critical for the function at hand (e.g., see ref. 23). Another example to consider is cell differentiation, where stochastic noise could determine one of many cell fates from the same starting point. The deterministic ODE method cannot be employed in these cases.

To simulate noisy models or models where stochasticity is important, the simulation should be performed with Gillespie's stochastic simulation algorithm (SSA) (24), which provides an exact way of simulating a set of chemical ODEs in terms of such discrete random

events. The SSA updates population numbers discretely, i.e., based on the firing of a single reaction event at a time. Because of this, the SSA is also typically much slower than ODE integration. When molecule numbers are high and there are no drastic events in the model, the discrete noise becomes less noticeable and the SSA trajectories are closer to the "smooth" limit of ODE trajectories. Subheading 6.3.2 and Fig. 5 provide an example of such a case.

Both SSA and ODE integration methods suffer from a significant computational limitation. They both require the entire reaction network to be stored in memory so that the respective populations and concentrations can be updated. This limits the size of the network that can be simulated (usually about 10^4 reactions and species). But biological reaction networks are typically dense and some networks can also be infinitely sized due to oligomerization (25).

For a certain subset of reaction networks (such as scaffolded interactions and linear cascades), significant reduction in network size can be achieved by exact model reduction techniques (12, 26, 27), which can systematically synthesize a compressed set of ODEs from a rule-based model specification (similar to the approach in Subheading 4.1 but on a larger scale). This enables exact simulation of the model for relevant outputs with a reduced number of differential equations. However, some types of interactions that are common in rule-based models, e.g., cooperativity between binding sites (such as in Subheading 6.2.4), preclude exact model reduction. Current model reduction methods are not powerful enough to guarantee reduction of large-scale rule-based models, and other simulation techniques are required to overcome the combinatorial bottleneck.

Another possible solution is to truncate the size of the network arbitrarily, for example, by not allowing complexes larger than ten molecules to form. However, the artificially constrained network is not guaranteed to have the same behavior as the full network. A more accurate approach is to generate only the portion of the network that is required at a particular time to advance the simulation. This forms the basis of "on-the-fly" methods (28, 29), one of which has been implemented in BNG (29). However, its performance degrades significantly for networks larger than about a thousand species (20).

The most efficient solution developed so far is to discard generating the network altogether. The molecules can be treated as particles instead of populations and particle-based stochastic simulation methods can be applied to them instead of the network-based SSA. This forms the basis of the *network-free simulation methods*, with computational and memory requirements that scale with the number of particles, rather than the network size (25, 30). There are a number of network-free reaction kinetics simulators that have been developed for rule-based modeling. STOCHSIM (31) was the first agent-based chemical kinetics simulator, but its rule-based language is less expressive than BNGL and Kappa and it

uses a sampling method that becomes inefficient as the range of rate constant values in the model becomes large (an order of magnitude or more), which is typical for large models. DYNSTOC (32) uses an extended version of the STOCHSIM algorithm and takes BNGL models as input, but still performs poorly in comparison to SSA-based simulators. RuleMonkey (33) works on BNGL models and uses a more efficient simulation algorithm based on the Gillespie's SSA. Both DYNSTOC and RuleMonkey have been validated for a broad range of rule-based models (32, 33). Network-free simulators are also available for the Kappa language (30) (see http://kappalanguage.org).

The most efficient and flexible network-free simulator compatible with BNGL that has been developed so far is *NFsim* (20) (see http://nfsim.org). NFsim and BNGL compatibly support global functional rate laws (such as those discussed in Subheading 5.3), local functions (which can be used to make rules even more concise and powerful), and logical Boolean functions. This increased functionality combined with the efficient particle-based stochastic simulator greatly expands the range of models that can be developed and simulated. The discussion of local functions is outside the scope of this tutorial and the reader is referred to Sneddon et al. (20). Defining global functions in BNGL is discussed in Subheading 6.1.

For models with less than a few hundred species, the simulation with SSA is faster than with NFsim. As the model size increases, however, the SSA rapidly becomes less efficient, whereas NFsim displays little or no decrease in performance as model size increases (20). Thus, NFsim allows the modeler to write any number of reaction rules without worrying if the underlying set of reactions is too large or infinite. One has to pay attention to the concentrations used, however, since the simulation speed is limited by the number of particles in the system. NFsim (and other network-free simulators such as RuleMonkey and the Kappa simulator) exactly samples the full chemical master equation for the system. Therefore, the trajectories produced from these simulators are indistinguishable from and equivalent to those produced by the SSA.

6. Receptor Ligand Interactions: A BioNetGen Tutorial

In this section, we walk through the process of constructing, simulating, and analyzing a BNGL model. We consider the example of the human epidermal growth factor (EGF) ligand binding specifically to the epidermal growth factor receptor (EGFR, also known as HER1 and ERBB1). This interaction is important for mammalian cells in most tissue types and functions as a signal that initiates growth and proliferation in both healthy and cancerous cells.

The model presented here can be used to investigate how hypothesized interactions affect aggregate distributions of receptors.

We formulate the model by drawing descriptions of biochemical interactions from the existing knowledge base in the biomedical literature and representing these as BNG molecules and rules. These descriptions should not be considered established facts, but rather should be considered hypotheses about the system structure, which can be tested by comparing model predictions with experimental observations and data.

The model is written in a normal text file with the extension ".bngl". The model comprises a series of text blocks, namely, parameters, molecule types, observables, seed species, reaction rules, and actions. The blocks' functions and compartments may also be provided in more complex models. All blocks are constructed as

```
begin block-name
 [blocktext]
 [blocktext]
 [blocktext]
end block-name
```

Any text on a line following a # (hash) symbol is considered a comment and is ignored. If a line becomes too long, the \ (backslash) symbol can be used to extend a statement to the next line. Subsequent tabs and spaces are treated as a single white space.

6.1. Seeding the Model

Knowledge Base:

> The ligand EGF monovalently binds the receptor EGFR. The extracellular domain of EGFR has a binding site for EGF and a domain that mediates dimerization with other receptor molecules (34). EGFR autophosphorylates at multiple amino acid positions, including Y1068 (tyrosine at position 1068 in the amino acid sequence of human EGFR) and Y1173 (reviewed in ref. 35).

Translating this information from the literature, we can construct the molecule types as follows:

`L(rec)`—ligand with a receptor-binding site.

`R(lig,dim,y1068~0~p,y1173~0~p)`—receptor with a ligand-binding site, a dimerization domain and two tyrosines that can be in unphosphorylated or phosphorylated states.

The molecule types are defined in a molecule types block, i.e.,

```
begin molecule types
  L(rec)
  R(lig,dim,y1068~0~p,y1173~0~p)
end molecule types
```

We also need to define the starting species and their concentrations for the model. This is provided using the seed species block. The complete species specifications should be provided. In this case, the starting species are simply free ligand and monomeric unphosphorylated receptor. It is a good convention to provide population numbers for the seed species and use microscopic rate constants in the reaction rules.

```
begin seed species
  R(lig,dim,y1068~0,y1173~0) R0
  L(rec) L0
end seed species
```

`R0` and `L0` are parameters that are to be defined separately in the parameters block. In the parameters block, a parameter name is assigned a real number or a standard mathematical expression involving other parameters. For example, `R0` can be defined directly as a number and `L0` can be defined in terms of concentration converted to population number. The units of the parameters can be given in comments for clarity:

```
begin parameters
  V_ext   1.6e-9     #liters
  N_Avo   6.022e23   #molecule number per mole
  R0    1e5        #molecule number per cell
  L0    L_conc*V_ext*N_Avo   #molecule number
end parameters
```

Parameters used elsewhere in this tutorial are assumed to have been defined in the parameters block either with numerical values or as expressions of other parameters.

Finally, we need to define the observables for the model. This model treats receptor aggregation and phosphorylation. Not only are we interested in the number of dimer species, but also in the number of receptors in dimers. These can be represented using species-observables and molecules-observables, respectively, in the observables block (note the use of wildcards in observables).

```
begin observables
  Molecules BoundLigand L(rec!+)
  Molecules BoundReceptor R(lig!+)
  Species Dimer R(dim!0).R(dim!0)
  Species UnligatedSpecies R(lig,dim),\
    R(lig,dim!0).R(lig,dim!0)
  Species PhosphSpecies R(y1068~p),R(y1173~p)
end observables
```

The `Molecules/Species` keyword indicates the type of observable. This is followed by the name assigned to the observable and then the list of patterns to match the species summed up in the observable.

An optional functions block can be used to define global functions of the observables. These functions can be used to track key variables in the simulation or for modified rate laws in reaction rules (see Subheading 5.3). The observables referenced by a global function must be defined in the observables block. The function itself is defined in the functions block. The observables block must *precede* the functions block for the observables to be used in the functions.

An example function involving the species-observable `SubstrateSum` would be defined in the functions block as:

```
begin functions
  k_func kcat*E_tot/(KM + SubstrateSum)
end functions
```

Although it is not used in this particular model, `k_func` defined here can be used instead of the Sat rate law in a reaction rule.

6.2. Building the Model

In this section, we create several hypotheses about the receptor aggregation mechanism and then implement reaction rules based on each hypothesis.

The rules are enclosed in a reaction rules block as

```
begin reaction rules
 [reaction_rule]
 [reaction_rule]
end reaction rules
```

6.2.1. Dimer-Dependent Phosphorylation

Knowledge Base:

> The kinase on EGFR is inactive in monomers and is activated by dimerization (36).

If both phosphorylation sites were represented as identical (e.g., if the molecule type were `R(lig,dim,y~0~p,y~0~p)` instead), then we would only need a single rule to represent it. The dimerization is given as context and the transformation is phosphorylation.

```
R(dim!0).R(dim!0,y~0) -> R(dim!0).R(dim!0,y~p) k_ph
```

However, we have chosen to model the two phosphorylation sites distinctly. Therefore, we would need two different rules. The kinetics can still be made identical with identical rate constants or nonidentical with different rate constants.

```
R(dim!0).R(dim!0,y1068~0)   ->\
  R(dim!0).R(dim!0,y1068~p)   k_ph1
R(dim!0).R(dim!0,y1173~0)   ->\
  R(dim!0).R(dim!0,y1173~p)   k_ph2
```

Since the only available binding partner on `dim` component is another `dim` component (i.e., we are not writing any reaction rules, where `dim` binds to anything else), we can shorten the context representation using the + wildcard on the bond.

```
R(dim!+,y1068~0) -> R(dim!+,y1068~p) k_ph1
R(dim!+,y1173~0) -> R(dim!+,y1173~p) k_ph2
```

Not dephosphorylating the tyrosine sites would result in runaway phosphorylation, which is undesirable. We can use a uniform first-order background dephosphorylation:

```
R(y1068~p) -> R(y1068~0) k_deph
R(y1173~p) -> R(y1173~0) k_deph
```

This is a valid assumption to make when the phosphatases are highly active and nonspecific.

6.2.2. Ligand-Dependent Dimerization

Hypothesis:

Ligand binding initiates receptor dimerization.

Expressed as a traditional equilibrium model:

$$\mathrm{R} + \mathrm{L} \overset{K_{\mathrm{bind}}}{\Longleftrightarrow} \mathrm{RL}.$$

$$\mathrm{RL} + \mathrm{RL} \overset{K_{\mathrm{dim}}}{\Longleftrightarrow} \mathrm{LRRL}$$

Here, the reactions are represented as bidirectional equilibrium reactions defined by *equilibrium association constants*, which are ratios of the forward and reverse rate constants. BNGL requires that the model be specified in terms of individual rate constants, i.e.,

$$\mathrm{R} + \mathrm{L} \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} \mathrm{RL}, \qquad \text{where } K_{\mathrm{bind}} = \frac{k_{\mathrm{f}}}{k_{\mathrm{r}}}.$$

$$\mathrm{RL} + \mathrm{RL} \underset{k_{\mathrm{rd}}}{\overset{k_{\mathrm{fd}}/2}{\rightleftharpoons}} \mathrm{LRRL}, \quad \text{where } K_{\mathrm{dim}} = \frac{k_{\mathrm{fd}}/2}{k_{\mathrm{rd}}}.$$

The two bidirectional reactions can be directly translated into bidirectional reaction rules.

```
R(lig,dim) + L(rec) <->\
  R(lig!0,dim).L(rec!0) k_f,k_r
R(lig!+,dim) + R(lig!+,dim) <->\
  R(lig!+,dim!0).R(lig!+,dim!0)k_f_d,k_r_d
```

Note that the bimolecular association in the second rule is symmetric and in the traditional expression would require a $(1/2)$ multiplication factor. However, as mentioned earlier, BNG requires the modeler to use the *asymmetric* reaction rate constant.

On processing the rule, BNG discovers the symmetry and automatically assigns the multiplication factor.

Knowledge Base:

> Both singly ligated and unligated dimers have been discovered to exist (37).

The model as it stands represents the typical manner in which ligand-induced dimerization and phosphorylation are modeled, e.g., in refs. 21, 38, 39. However, this model is deficient because it does not capture the full nature of the interaction between ligand-binding and dimerization processes. In the current model hypothesis, the dimer state is inextricably linked to the ligand-bound state, which is not true in the light of the above evidence. The experimental evidence of existence of singly ligated and unligated dimers enables *rejecting* the current model hypothesis.

6.2.3. Ligand-Independent Dimerization

To explain the existence of singly ligated and unligated dimers, we introduce an alternate hypothesis.

Hypothesis:

> Ligand binding and receptor dimerization are independent of each other.

Expressed as a traditional equilibrium model (with equilibrium association constants):

$$\begin{array}{ccccc} \mathrm{R,R,L,L} & \overset{K_{\mathrm{bind}}}{\Longleftrightarrow} & \mathrm{R,RL,L} & \overset{K_{\mathrm{bind}}}{\Longleftrightarrow} & \mathrm{RL,RL} \\ \Updownarrow K_{\mathrm{dim}} & & \Updownarrow K_{\mathrm{dim}} & & \Updownarrow K_{\mathrm{dim}} \\ \mathrm{RR,L,L} & \overset{K_{\mathrm{bind}}}{\Longleftrightarrow} & \mathrm{RRL,L} & \overset{K_{\mathrm{bind}}}{\Longleftrightarrow} & \mathrm{LRRL} \end{array}$$

In terms of reactions:

$$\begin{array}{ccccc} \mathrm{R,R,L,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{R,RL,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{RL,RL} \\ k_{\mathrm{fd}} \downharpoonleft\upharpoonright k_{\mathrm{rd}} & & k_{\mathrm{fd}} \downharpoonleft\upharpoonright k_{\mathrm{rd}} & & k_{\mathrm{fd}} \downharpoonleft\upharpoonright k_{\mathrm{rd}} \\ \mathrm{RR,L,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{RRL,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{LRRL} \end{array},$$

where $K_{\mathrm{bind}} = \dfrac{k_{\mathrm{f}}}{k_{\mathrm{r}}}$ and $K_{\mathrm{dim}} = \dfrac{k_{\mathrm{fd}}}{k_{\mathrm{rd}}}$.

Since the dimerization and ligand binding reactions are completely independent of each other in this model, we can model this system using two context-free rules:

```
R(lig) + L(rec) <-> R(lig!0).L(rec!0) k_f,k_r
R(dim) + R(dim) <-> R(dim!2).R(dim!2) k_f_d,k_r_d
```

Note that by removing context we have enabled the reaction rule to generate additional reactions. The dimerization rule in the earlier hypothesis generated only one reaction: RL with RL. Here, it generates three dimerization reactions: R with R, R with RL, and RL with RL.

Knowledge Base:

Phosphorylation of receptor increases on ligand addition (reviewed in ref. 35).

In this model, we have delinked the dimerization from ligand binding, i.e., ligand binding does not influence the equilibrium concentrations of the dimer (and vice versa). Since phosphorylation is dimer dependent, this implies that ligand binding cannot directly influence phosphorylation. The model behavior is at odds with experimental evidence. Hence, this model hypothesis is also *rejected*.

6.2.4. Cooperative Ligand Binding and Dimerization

Two events are said to exhibit *cooperativity* if the sequence of occurrence of those two events affects the intrinsic rates at which they occur. One plausible model for dimer formation posits that the monomer, unligated dimer, and singly ligated dimer all have different affinities for the ligand (40).

Hypothesis:

Ligand binding and receptor dimerization are mutually cooperative.

Expressing this as a traditional equilibrium model (with equilibrium association constants):

$$\begin{array}{ccccc} \mathrm{R,R,L,L} & \overset{K_{\mathrm{bind},1}}{\Leftrightarrow} & \mathrm{R,RL,L} & \overset{K_{\mathrm{bind},1}}{\Longleftrightarrow} & \mathrm{RL,RL} \\ \Updownarrow K_{\mathrm{dim},1} & & \Updownarrow K_{\mathrm{dim},2} & & \Updownarrow K_{\mathrm{dim},3}\,. \\ \mathrm{RR,L,L} & \overset{K_{\mathrm{bind},2}}{\Leftrightarrow} & \mathrm{RRL,L} & \overset{K_{\mathrm{bind},3}}{\Longleftrightarrow} & \mathrm{LRRL} \end{array}$$

Cooperativity imposes thermodynamic constraints on the model. If there is no external energy source or sink, a system of reversible reactions should obey mass and energy conservation.

If there are multiple paths from the same set of reactants to the same set of products, then the product of the equilibrium constants along all paths should be identical. In other words, the product of equilibrium constants over a closed loop of reversible reactions should be unity. This effect is called *detailed balance* and places constraints on the parameters and rules used to model the system (e.g., in ref. 38). In this case, there are at least two closed loops, resulting in the following constraints:

$$K_{\text{bind},1} K_{\text{dim},2} = K_{\text{bind},2} K_{\text{dim},1}.$$
$$K_{\text{bind},3} K_{\text{dim},2} = K_{\text{bind},1} K_{\text{dim},3}.$$

To simplify parameters, we describe the three ligand-binding equilibrium constants using multiplicative factors α and β on the monomer ligand-binding equilibrium constant K_{bind}, i.e.,

$$K_{\text{bind},1} = K_{\text{bind}}.$$
$$K_{\text{bind},2} = \alpha K_{\text{bind},1}.$$
$$K_{\text{bind},3} = \beta K_{\text{bind},1}.$$

Using these relations and the constraints obtained previously, we can describe the dimerization equilibrium constants as multiplicative factors on the unligated dimerization equilibrium constant K_{dim}:

$$K_{\text{dim},1} = K_{\text{dim}}.$$
$$K_{\text{dim},2} = \alpha K_{\text{dim}}.$$
$$K_{\text{dim},3} = \alpha\beta K_{\text{dim}}.$$

The network can be rewritten as

$$\begin{array}{ccccc} \text{R,R,L,L} & \overset{K_{\text{bind}}}{\Longleftrightarrow} & \text{R, RL, L} & \overset{K_{\text{bind}}}{\Longleftrightarrow} & \text{RL, RL} \\ \Updownarrow K_{\text{dim}} & & \Updownarrow \alpha K_{\text{dim}} & & \Updownarrow \alpha\beta K_{\text{dim}} \\ \text{RR,L,L} & \overset{\alpha K_{\text{bind}}}{\Longleftrightarrow} & \text{RRL, L} & \overset{\beta K_{\text{bind}}}{\Longleftrightarrow} & \text{LRRL} \end{array}.$$

Note that BNG requires reaction rate constants, whereas here we have only managed to obtain multiplicative factors for the equilibrium constants. We do not know how the factor distributes between the forward reaction and the reverse reaction and must make assumptions about it.

A general way of resolving this issue would be to distribute the multiplicative factors in the following manner (41):

$$\text{If K'} = \frac{k'_{\text{f}}}{k'_{\text{r}}} = \alpha K = \alpha\left(\frac{k_{\text{f}}}{k_{\text{r}}}\right).$$
$$k'_{\text{f}} = \alpha^{\phi} k_{\text{f}}.$$
$$k'_{\text{r}} = \alpha^{\phi-1} k_{\text{r}}.$$
$$0 \leq \phi \leq 1.$$

Now, by changing ϕ, we can control the distribution of the multiplicative factor over the forward and reverse reaction rate constants in the most general way. There is no single best value for ϕ. In this tutorial, we use $\phi = 0$, causing the multiplicative factor to affect only the reverse rate and not the forward rate, i.e.,

$$k'_{\mathrm{f}} = k_{\mathrm{f}}.$$
$$k'_{\mathrm{r}} = \alpha^{-1} k_{\mathrm{r}} = \frac{k_{\mathrm{r}}}{\alpha}.$$

Rewriting the equilibrium model in terms of reactions and implementing the multiplicative factors (for the equilibrium constants) as divisive factors on the reverse rate:

$$\begin{array}{ccccc} \mathrm{R,R,L,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{R,RL,L} & \underset{k_{\mathrm{r}}}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{RL,RL} \\ k_{\mathrm{fd}} \downuparrows k_{\mathrm{rd}} & & k_{\mathrm{fd}} \downuparrows k_{\mathrm{rd}}/\alpha & & k_{\mathrm{fd}} \downuparrows k_{\mathrm{rd}}/\alpha\beta \\ \mathrm{RR,L,L} & \underset{k_{\mathrm{r}}/\alpha}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{RRL,L} & \underset{k_{\mathrm{r}}/\beta}{\overset{k_{\mathrm{f}}}{\rightleftharpoons}} & \mathrm{LRRL} \end{array},$$

where $K_{\mathrm{bind}} = \dfrac{k_{\mathrm{f}}}{k_{\mathrm{r}}}$ and $K_{\mathrm{dim}} = \dfrac{k_{\mathrm{fd}}}{k_{\mathrm{rd}}}$.

These reactions are now consistent with detailed balance. Depending on the values chosen for α and β, one can cause the system to exhibit multiple cooperative behaviors.

Knowledge Base:

> Dimerization is enhanced by ligand binding. However, the two ligand-binding sites on the dimer are negatively cooperative (42).

If the unligated dimer has a higher affinity for ligand than the monomer, the presence of ligand shifts the dimer–monomer equilibrium towards the dimer, which also increases receptor phosphorylation. In this model, ligand binding cooperates positively with dimer formation leading to an increase in dimerization when ligand is added. Another feature of this system, however, appears to be that once one receptor in a dimer binds ligand, the second receptor exhibits a considerably reduced affinity for ligand (40). In other words, ligand binding is *negatively cooperative* on the dimer. Structural evidence strongly supporting negative cooperativity has emerged recently (43).

Based on the equilibrium constants determined from experiment ((42), Fig. 3), we use the values $\alpha = 120$ and $\beta = 0.07$. These values are consistent with our previous discussion that ligand binding

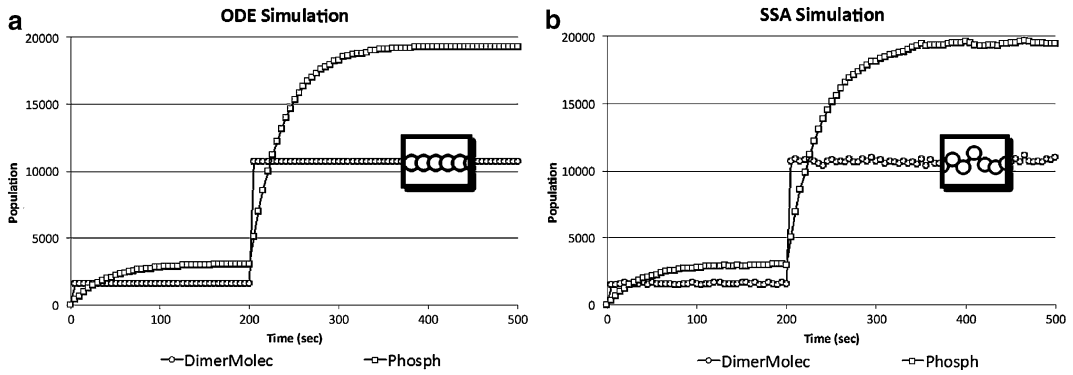

Fig. 3. Simulation of receptor dimerization model. (**a**) A model of ligand-induced dimerization and subsequent kinase activation is simulated using deterministic ODE integration. From $t = 0$ to 200 s, the system is equilibrated, and at $t = 200$ s, the system is perturbed by adding the ligand. The monomer–dimer equilibrium shifts rapidly with an increase in the number of molecules in dimers. Since kinase activity depends on dimerization, the number of phosphorylated sites also increases concurrently. Note that the trajectories are smooth (see *inset*). (**b**) The same sequence of simulations is performed using Gillespie's SSA. Note that the trajectories are noisy (see *inset*), but the noise is relatively small because of the larger population sizes and the observed behavior is similar to that of *Panel* (**a**).

enhances dimerization ($K_{\mathrm{dim},2} = 120\ K_{\mathrm{dim},1}$ and $K_{\mathrm{dim},3} = 8.4\ K_{\mathrm{dim},1}$) whereas the ligand binding to the dimer is negatively cooperative ($K_{\mathrm{bind},3}$=0.07 $K_{\mathrm{bind},2}$).

Modeling this scenario requires more reaction rules than the previous model. This is a characteristic feature of rule-based modeling. Independent processes, lacking reaction context, are the easiest to express with the fewest number of rules. Unidirectional influences, in which one process affects the rate of another but not vice versa, require additional context. Bidirectional influences, in which two processes mutually affect their rates giving rise to cooperativity, represent the most complex linkage between two components. Care must be taken in modeling such interactions to ensure that detailed balance is obeyed.

For the current hypothesis, we can write the rules as follows:

```
# Ligand association
R(lig)+L(rec) -> R(lig!1).L(rec!1) k_f
#Ligand dissociation for monomer, singly-ligated dimer and
  doubly-ligated dimer
R(dim,lig!1).L(rec!1) -> R(dim,lig)+L(rec) k_r
R(dim!2,lig).R(dim!2,lig!1).L(rec!1) -> \
R(dim!2,lig).R(dim!2,lig) + L(rec)
  k_r/alpha
R(dim!2,lig!+).R(dim!2,lig!1).L(rec!1) -> \
R(dim!2,lig!+).R(dim!2,lig) + L(rec) k_r/beta
```

```
# Dimer association
R(dim) + R(dim) -> R(dim!1).R(dim!1) k_f_d
#Dimer dissociation for unligated/singly-ligated/doubly-
  ligated dimers
R(lig,dim!1).R(lig,dim!1) -> R(lig,dim) + R(lig,dim)
  k_r_d
R(lig!+,dim!1).R(lig,dim!1) -> R(lig!+,dim) +
  R(lig,dim) k_r_d/alpha
R(lig!+,dim!1).R(lig!+,dim!1) -> R(lig!+,dim) +
  R(lig!+,dim) k_r_d/(alpha*beta)
```

This model of receptor aggregation corresponds closely with the experimental and structural evidence. Hence, we accept these rules and proceed to simulate the model. The choice of rate parameters is discussed in general in Subheading 8.

6.3. Simulating the Model

The molecule types, parameters, seed species, and reaction rules constitute the model. Various actions can then be performed on the model using the actions block. Actions are preexisting BNGL routines that can be called at will by the modeler. Typically, actions require action parameters and flags and should be terminated with the semicolon symbol.

```
begin actions
 [action];
 [action];
end actions
```

6.3.1. Network Generation

The network generation action starts with the seed species and iteratively applies the reaction rules to the species set to generate new reactions and species. This can proceed until the network size stops increasing, in which case the whole network has been generated. Or, the size of the network can be arbitrarily limited by limiting the number of iterations (using `max_iter` flag) or the maximum number of molecules (using `max_stoich`) in a complex. The network is printed out in a text file with ".net" extension. The name of the *.bngl* file is used as the base name for the *.net* file.

The different species are assigned numbers by default to compactly represent the reactions, but the full configuration can also be written out instead using a `TextReaction` flag. The `overwrite` flag is used to indicate whether to overwrite a preexisting file of the same name. A typical `generate_network` command would look like this:

```
generate_network({overwrite=>1,max_stoich=>{R,2},
  max_iter=>20,TextReaction=>1});
```

In this statement, the `generate_network` routine is called which generates the reaction network subject to the condition that the maximum number of receptor molecules is 2 in any complex and that the rules are iteratively applied at most 20 times. When the network is being written in the *.net* file, the routine is allowed to overwrite the existing files of the same name and is told to write down the full species specifications in the reactions.

For the purposes of this model, we could simply use:

```
generate_network({overwrite=>1,TextReaction=>1});
```

For large networks, it is advisable to generate the network only once using the `generate_network` command and then for subsequent simulation reuse the network using the `readFile` action or by giving the *.net* file name as a parameter in other actions (see online documentation at http://bionetgen.org for more details).

6.3.2. Equilibration and Perturbation

In any experiment, a ground state is perturbed in some defined fashion and then the ground state and perturbed states are compared using some measurable quantity. In biological systems, the ground state is typically a preexisting equilibrated control state and the perturbation is typically the addition of some chemical species or activation of some reaction. Then, after a defined time or at many time points, the two states are compared using some experimental probe.

When simulating a biological model, it is important to follow the same principle. To establish the ground state of the experiment, the model is seeded with everything, except the perturbing agent and simulated until equilibrium or steady state is achieved. This step is called the *equilibration* step. A "good" model, i.e., one which closely corresponds to reality, is one where all transformations are complementary and, in the absence of a perturbing agent, the system equilibrates. Following equilibration, the perturbing agent is added to the system and subsequent simulation is called *perturbation*, or plainly *simulation* (which can be misleading).

For the receptor aggregation model, the ground state involves receptor association and dissociation in the absence of ligand and the perturbed state involves addition of ligand and monitoring the aggregation. The monomeric receptors that seed the model would necessarily be far away from equilibrium. Using actions, we can set the ligand concentration to zero and then use numerical ODE integration to equilibrate the model.

```
simulate_ode({suffix=>ode,t_start=>0,t_end=>200,
 n_steps=>100});
```

The `suffix` option is used to append the "_" and then "ode" to the base name, which is taken from the name of the *.bngl* file. `t_start` and `t_end` indicate the total time over which the model is equilibrated and `n_steps` indicates the number of time points at which concentrations are recorded.

Following equilibration, we wish to add ligand and perturb the model. To do this, first we need to save the concentrations from the equilibration step and then add new ligand. This is accomplished using

```
saveConcentrations();
setConcentration("L(rec)","L0_temp");
```

The `continue` flag can be used to turn concatenate a new simulation with the earlier one. One must make sure that the `t_start` of the new simulation is the same as the `t_end` of the earlier simulation and that the `suffix` value is not changed. Any number of simulations can be so concatenated.

```
simulate_ode({suffix=>ode,continue=>1,t_start=>200,
  t_end=>500,n_steps=>100});
```

Invocation of the SSA simulator follows the same syntax, but the `simulate_ssa` command is used instead. For the SSA and ODE simulations, the entire network is generated (and saved in the *.net* file) and the trajectories (concentration vs. time) of all species are recorded in the *.cdat* text file. The trajectories of the observables (which are weighted sums of specific species) are recorded in the. *.gdat* text file. The NFsim method does not generate the network and, therefore, writes only a *.gdat* file. Since multiple simulations can be concatenated in a single actions block, use of appropriate suffixes is encouraged to distinguish between data files.

If the `continue` flag was set to 1 and the suffix was not changed, then the trajectory of the simulation is concatenated with the previous simulation. For a single sequence of concatenated simulations, a single *.cdat* file and a single *.gdat* file are generated. If the `continue` flag was set to 0 (which is the default value) or if the suffix used for the simulation step is changed, then each simulation output is written to a separate set of data files.

To simulate the same perturbation using SSA, we would need to reset the species concentrations to the equilibrated state. This is performed by the `resetConcentrations()` method which changes all the species concentrations to what they were during the last `saveConcentrations()` call. Following the reset, ligand is introduced using the `setConcentration()` method and SSA is used to simulate the model.

NFsim can be invoked using the `simulate_nf` command, provided the binary for NFsim exists in the bin subfolder of the BNG installation. The actions that can be used in the actions block and their respective syntax are posted with the documentation on the BNG Web site (http://bionetgen.org).

6.3.3. Execution

The actions block is used to list the sequence of commands that the modeler wishes to implement when BNG executes the model. The modeler should save the complete model as a *.bngl* file (say model. bngl), open a terminal or command prompt, and enter the directory

in which the BNGL file resides. Then, the following command should be issued:

```
perl /.../BioNetGen/Perl2/BNG2.pl model.bngl
```

where */. . ./* is a placeholder for the path to the parent folder where BNG is installed. The slash convention used (backslash, forward slash) might differ depending on the terminal used. For example, on a Windows command prompt, the command would look like:

```
perl C:\BioNetGen\Perl2\BNG2.pl model.bngl
```

On a Unix terminal such as bash (found in OS X and Linux), the command would look like:

```
perl /Users/username/BioNetGen/Perl2/BNG2.pl model.
  bngl
```

Perl is required to run BNG and might need to be separately installed on certain platforms. The command can also be embedded in terminal scripts and other scripting environments, such as R, Octave, MatLab, or Mathematica.

On execution, BNG reads the BNGL file and sequentially implements the actions in the actions block. Files written by the actions (say, network generation or simulation) are saved in the same folder.

6.3.4. Analysis

The simulation data are in table format in the *.cdat* and *.gdat* text files. They can be easily imported into different software for statistical analysis and visualization, including (but not limited to) spreadsheet software (Microsoft Excel, OpenOffice Calc), plotting programs (Gnuplot, Grace), or interactive computing environments (Matlab, Octave, R, Mathematica). The models themselves can be exported to the SBML language using `writeSBML` action, as a Matlab function file using `writeMfile` action or as a Matlab Mex file using the `writeMexfile` action. The M-file uses Matlab's inbuilt ODE15 simulation engine, whereas the Mex file is a compiled ODE model that utilizes the faster CVODE libraries used in BNG. The Mex file can be used to run computationally intensive parameter scans and analysis in Matlab.

RuleBender, the integrated development environment for BNG (see Subheading 2), also provides facilities for plotting and analyzing simulation output. In addition, it provides several different global views on models, including contact maps and influence diagrams (see http://rulebender.org).

The trajectories for both ODE and SSA simulations of the same model are shown in Fig. 3. The model file used for the simulation is shown in Fig. 4. Note how the ODE simulations are smooth, but the SSA simulations exhibit some noise. Because the model involves large populations, the SSA trajectories closely resemble the ODE trajectories.

```
begin parameters
# Universal constants
 N_Avo  6.023e23 # per mol

# Biophysical considerations
# Cell is a CHO cell, dimensions & quantities from ref 1
# Volume 1.2 pL, Cytoplasm 1 pL, Radius 6.6 microns, Sphere
# Surface area 5.5e-10 m^2 or 5.5e-8 dm^2, Membrane thickness 10 nm (or) 1e-7 dm
# External volume 1.6 nL (assuming 6.15e5 cells/mL)
 V_ext  1.6e-9 # L
 V_cell 1.2e-12 # L
 V_cyt          1.0e-12 # L
 V_nuc          V_cell - V_cyt # L
 h_mem          1e-7 # dm
 S_mem          5.5e-8 # dm^2
 V_mem          h_mem*S_mem # L, comes to 5.5 fL

# Assumed association reaction rate constant, per M per s
 k_ext  1e9 # EGF, small freely diffusing ligand, so high
 k_mem  1e6 # EGFR, large bulky protein, so low. Calculated from refs 5-7
 k_cyt  1e9 # Only small molecules such as Grb2, Sos

# Equilibrium Constants (Association Constants). Calculated from ref 2
 Kdim_2D        5.3e9 # per (mol/dm^2)
 Kdim   Kdim_2D*h_mem # per (mol/dm^3) or per M
 Kbind  4.6e9 # per M
 alpha  120
 beta   0.07

# Kinetic Rate Constants
 k_f            k_ext/(N_Avo*V_ext)
 k_f_d  k_mem/(N_Avo*V_mem)

 k_r1   k_ext/Kbind
 k_r2   k_r1/alpha
 k_r3   k_r1/beta

 k_r_d1 k_mem/Kdim
 k_r_d2 k_r_d1/alpha
 k_r_d3 k_r_d1/(alpha*beta)

# Kinase & Phosphatase
 kcat           0.025 # per s

# Initial Concentrations
 R_conc 1e5 # Receptors per cell
 L_conc 1e-3    # M
 P_conc 200e-9  # M
 R0             R_conc
 L0_new L_conc*V_ext*N_Avo
 P0             P_conc*V_cyt*N_Avo
 L0             0
end parameters

begin molecule types
 R(lig,dim,tyr~u~p,tyr~u~p)
 L(rec)
 PTase(a)
end molecule types

begin seed species
 R(lig,dim,tyr~u,tyr~u) R0
 L(rec)                                 L0
 PTase(a)                               P0
end seed species

begin reaction rules
# Ligand binding
R(lig) + L(rec) -> R(lig!0).L(rec!0) k_f
R(dim,lig!0).L(rec!0) -> R(dim,lig) + L(rec) k_r1
R(dim!1,lig).R(dim!1,lig!0).L(rec!0) -> R(dim!1,lig).R(dim!1,lig) + L(rec) k_r2
R(dim!1,lig!+).R(dim!1,lig!0).L(rec!0) -> R(dim!1,lig!+).R(dim!1,lig) + L(rec) k_r3

# Dimerization
R(dim) + R(dim) -> R(dim!0).R(dim!0)    k_f_d
R(dim!0,lig).R(dim!0,lig) -> R(dim,lig) + R(dim,lig) k_r_d1
R(dim!0,lig!+).R(dim!0,lig) -> R(dim,lig!+) + R(dim,lig) k_r_d2
R(dim!0,lig!+).R(dim!0,lig!+) -> R(dim,lig!+) + R(dim,lig!+) k_r_d3

# Phosphorylation
R(dim!+,tyr~u) -> R(dim!+,tyr~p) kcat
R(tyr~p) -> R(tyr~u)    kcat
#R(tyr~p) -> R(tyr~u)   Sat(vmax,KM)
end reaction rules

begin observables
 Molecules      BoundLig        L(rec!+)
 Molecules      Monomers        R(dim)
 Molecules      DimerMolec      R(dim!+)
 Molecules      Phosph  R(tyr~p)
end observables

begin actions
generate_network({overwrite=>1});
simulate_ssa({suffix=>ssa,t_start=>0,t_end=>200,n_steps=>200});
saveConcentrations();
setConcentration("L(rec)","L0_new");
simulate_ssa({suffix=>ssa,continue=>1,t_start=>200,t_end=>500,n_steps=>300});
end actions

#references
#1: Seewoster and Lehmann. Biotechnology and Bioengineering (1997) vol. 55 (5) pp. 793-797
#2: Macdonald and Pike. Proc Natl Acad Sci USA (2008) vol. 105 (1) pp. 112-7
#3: Gabdoulline and Wade.Current Opinion in Struct Biol (2002) vol. 12 (2) pp. 204-13
#4: Kholodenko. European Journal of Biochemistry (2000) vol. (267) pp. 1583-1588
#5: Mayawala et al. Biophys Chem (2006) vol. 121 (3) pp. 194-208
#6: Lauffenburger and Linderman, Oxford University Press, New York, 1993.
#7: Kusumi et al. Biophysical Journal (1993) vol. 65 (5) pp. 2021-40
```

Fig. 4. Model file. This is the model file used for simulation in Subheading 6, the results of which are plotted in Fig. 3.

Since BNGL models are merely text files, scripts written in scripting languages (such as Python or Perl) or scripting environments (such as Matlab) may be used to read, modify, and simulate BNGL models from the outside. This flexibility enables BNGL models to be processed for advanced model analysis, including fitting models to experimental data, scanning numerical ranges for parameters, checking local and global parameter sensitivity, using BNGL models as modules in a hierarchical framework, etc.

7. Compartmental Modeling

The default assumption in BNG is a reactor of unit volume. For modeling reactions in multiple volumes, the user can manually model the location as a separate component state and provide it as context for each reaction rule. The user would also need to include the correct volume-scaling factor for each reaction rule, however, and this quickly gets tedious as the number of compartments

increases, especially for reaction rules that perform the same transformation in different compartments. For detailed modeling with multiple compartments, a more sophisticated approach is required.

The compartmental extension to BNG seeks to alleviate these issues by accommodating within its language a compartmental hierarchy, a compartment attribute to molecules and species, an automatic detection of possible inter-compartmental transport reactions, and an automatic scaling of reaction rules based on the locations of the reactants. cBNG was introduced and discussed in detail by Harris et al. (44). In this section, we provide the technical concepts underlying compartmental modeling in cBNG and explain them using appropriate examples.

7.1. Compartment Topology

There are two types of compartments in cBNG: 3D *volumes* and 2D *surfaces*. cBNGL supports a hierarchical topology that mimics that of the cell. Every volume can enclose one or more surfaces, whereas every surface has to enclose exactly one volume. The enclosing compartment is referred to as the parent and the enclosed compartments are referred to as children. A compartment is considered *adjacent* to its parent and children. Volumes cannot enclose volumes and surfaces cannot enclose surfaces.

The parent volume that encloses a surface and the child volume enclosed by the same surface are referred to as *bridged-volumes* and the intermediary surface is referred to as the *bridging surface*. Similarly, the parent surface that encloses a volume and any child surface enclosed by the same volume are referred to as *bridged-surfaces* and the intermediary volume is referred to as the *bridging volume*.

The compartments block is used to encode the compartment hierarchy and compartment volumes. It is an optional block that triggers the use of cBNGL framework and syntax. Consider the hierarchy described in Fig. 5. This can be encoded in a compartments block as follows:

```
begin compartments
  Ext 3   V_ext          # External milieu
  Plm 2   V_plm   Ext    # Plasma Membrane, enclosed
by Ext
  Cyt 3   V_cyt   Plm    # Cytosol, enclosed by Plm
  Enm 2   V_enm   Cyt    # Endosomal Membrane,
enclosed by Cyt
  Num 2   V_num   Cyt    # Nuclear Membrane,
enclosed by Cyt
  End 3   V_end   Enm    # Endosome, enclosed by Enm
  Nuc 3   V_nuc   Num    # Nucleus, enclosed by Num
end compartments
```

Each line indicates the name of the compartment, the dimension (2D or 3D), the volume of the compartment, and the name of the parent compartment. A short description can be given as a comment for clarity.

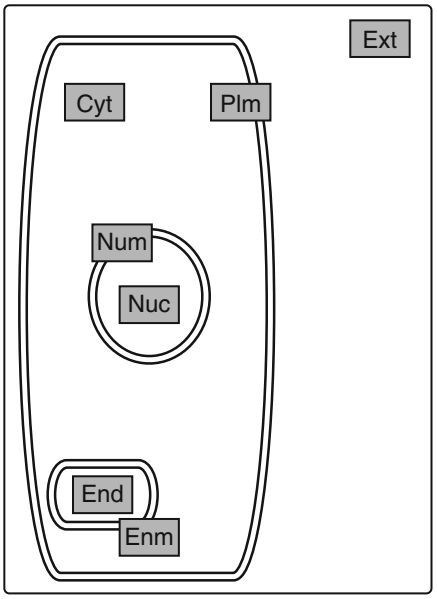

Fig. 5. Compartment hierarchy. To use cBNG, the modeler must define a compartments block with an appropriate nesting of compartments. 3D-compartments, called volumes, can only enclose 2D-compartments, called surfaces, and vice versa. A volume can enclose more than one surface, but a surface cannot enclose more than one volume. In this hierarchy, Ext, Cyt, Nuc, and End are volumes and Plm, Num, and Enm are surfaces.

7.2. Molecule Location

Molecule location is given by a postfix with the @ symbol in both patterns and species. For example, to represent a monomeric receptor present in the plasma membrane compartment (`Plm`), we use:

```
R(dim)@Plm
```

In patterns and species with multiple molecules, every molecule's location can be indicated by an @ postfix. For example, to represent a dimer of receptors present in the plasma membrane, we use:

```
R(dim!0)@Plm.R(dim!0)@Plm
```

Molecules can form bonds with molecules in the same compartments or in adjacent compartments. Thus, a single species can span multiple compartments. For example, to represent a receptor on the plasma membrane that binds a ligand outside the cell, we use:

```
R(lig!0)@Plm.L(rec!0)@Ext
```

A *topologically consistent species* does not span more than one surface and does not have bonds that need to pass through compartments. Examples are shown in Fig. 6.

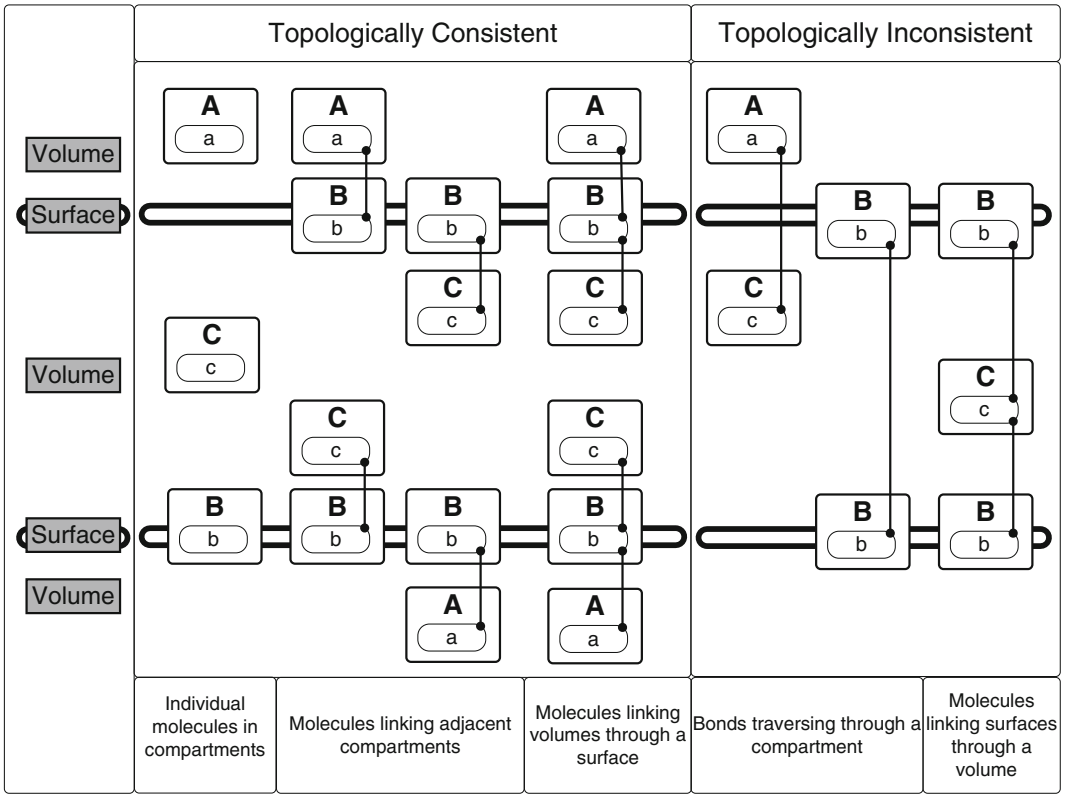

Fig. 6. Topologically consistent species. cBNG maintains topological consistency whenever a transport rule is implemented. Topologically consistent species do not have bonds traversing compartments. The species itself cannot span more than one surface. Reactions that create topologically inconsistent species are discarded.

If cBNG is invoked using the compartments block, and then the seed-species block should contain topologically consistent species with their full compartmental specification, for example,

```
begin seed species
 R(lig,dim)@Plm   R0
 L(rec)@Ext       L0
end seed species
```

7.3. Location in Patterns

cBNG provides a lot of flexibility in tailoring patterns to match species based on the location of their molecules.

The *aggregate location* of a species can be inferred from the location of its molecules.

- If all the molecules in a species are in a volume compartment, then their aggregate location is that volume compartment.
- If all the molecules in a species are in a surface compartment, then their aggregate location is that surface compartment.
- If the species spans one surface and one or two volumes, then the aggregate location is that surface compartment.

Species can be referred to by their aggregate location using an @ prefix on a pattern. For example, consider the dimer pattern (on the plasma membrane), `R(dim!0)@Plm.R(dim!0)@Plm`.

The same pattern can be referred to using the aggregate location as `@Plm:R(dim!0).R(dim!0)`.

For species that bridge compartments, the aggregate location (the surface) can still be provided as @ prefix, whereas the connected molecules in the adjacent compartments can be tagged with the @ postfix as exceptions to the rule.

For example, consider the receptor dimer on the plasma membrane connected to a single ligand molecule in the external compartment: `R(dim!0,lig!1)@Plm.L(rec!1)@Ext.R(dim!0)@Plm`.

The aggregate location is the `Plm` surface. To match this species using a pattern, we can use the `@Plm` prefix and provide the `Ext` locations as postfix (underlined for emphasis), i.e., `@Plm:R(dim!0,lig!1).R(dim!0).L(rec!1)@Ext`

Volume molecules bound to a surface can be matched by prefix location tags of the aggregate location.

For example, consider the species: `R(dim,lig!1)@Plm.L(rec!0)@Ext`.

The aggregate location of the species is `Plm`.

It can be matched by both patterns: `L(rec!+)@Ext` and `@Plm:L(rec!+)`.

The matched region is underlined for emphasis.

7.4. Reaction Rules in the Compartmental Framework

7.4.1. Automatic Volume Scaling

If cBNG is not invoked, i.e., if the compartments block is not used, the modeler must manually provide the volume factor and Avogadro number factor for the microscopic reaction rate constant, for example:

```
A(b) + B(a) -> A(b!0).B(a) k/(N_Avo*V_cyt)
```

If cBNG is invoked, i.e., the compartments and volumes are specified in the compartments block, then the volume scaling is part of the cBNG processing and the modeler needs to only provide the Avogadro number factor. For example, the same reaction rule would be modeled in the cBNG framework as:

```
A(b) + B(a) -> A(b!0).B(a) k/N_Avo
```

cBNG would automatically scale it to `k/(N_Avo*V_cyt)` if it detects that the bimolecular reaction can occur in the cytoplasm and `V_cyt` is specified as the volume of the cytoplasmic compartment in the compartments block.

In accordance with the explanation given in Subheading 5.1, unimolecular reactions are never scaled by volume, either in surfaces or volumes. Bimolecular reactions in "volumes" and "surfaces" are automatically scaled by the respective compartmental volume. Molecules in surfaces are assumed to be restricted to a

small volume enveloping the surface (i.e., the surface volume is equal to the surface area multiplied by a surface thickness) provided by the modeler. It is *not* recommended to model reaction orders higher than bimolecular in the compartmental framework.

7.4.2. Universal Reaction Rules

Biological reactions can happen between reactants as long as they are in the same or adjacent compartments. Often, the identity of the compartments could be irrelevant to the rate constant (except for the volume-scaling factor). For example, multiple cell types can produce different numbers of the same receptor on their plasma membranes, but these receptors bind with the same intrinsic affinity to an external ligand. Such universal phenomena can be written as universal reaction rules, i.e., without any compartmental context.

On processing a universal reaction rule, cBNG can identify the compartments in which the reactants can occur. Then, cBNG automatically generates reactions matching the reaction rule, but applicable to the specific compartments (or pairs of adjacent compartments) where the reactants are in proximity. For bimolecular and higher reactions, cBNG automatically scales the reaction rate by the appropriate volume.

For example, receptor dimerization can occur on both plasma membranes (`Plm`) and endosomal membranes (`Enm`). This can be represented by a universal rule:

```
R(dim) + R(dim) -> R(dim!0).R(dim!0) k
```

This rule would generate dimerization reactions in both `Plm` and `Enm` compartments.

```
R(dim)@Plm + R(dim)@Plm -> R(dim!0)@Plm.R(dim!0)@Plm
  k/V_Plm
R(dim)@Enm + R(dim)@Enm -> R(dim!0)@Enm.R(dim!0)@Enm
  k/V_Enm
```

If a specific reacting pair occurs only in one location (or one pair of adjacent locations), then it is sufficient to use a universal rule to model the reaction.

7.4.3. Scope-Restricted Reaction Rules

Scope-restricted reactions are reactions whose behavior the modeler wishes to restrict depending on where the location of the reaction is. For example, the same reactions can occur in three different compartments, but the modeler might wish to assign different microscopic rates for the reaction in two compartments and disallow the reaction in the third.

Universal rules cannot be employed for these purposes. In cBNGL, by using the @ symbol, modelers have the power to specify exactly where a particular reaction occurs. For example, suppose receptors were degraded at two different rates depending on the location, i.e., rapidly in endosomes, but only very slowly in the plasma membrane, then we can use scope-restricted rules to show this:

```
R()@End -> 0 k_degr_fast DeleteMolecules
R()@Plm -> 0 k_degr_slow DeleteMolecules
```

Appropriate volume scaling is done for bimolecular and higher order reactions.

7.4.4. Transport Rules

The cBNGL specification allows molecules and species to be moved between compartments in a number of ways as long as the resulting species is *topologically consistent*. If a transport reaction generated by a transport rule creates a topologically inconsistent species, BNG will detect it and discard the reaction.

Molecules can be moved individually from their location to any adjacent compartment (volume to surface or surface to volume) simply by changing the @-postfix. For example, moving a hypothetical A molecule from the plasma membrane to the child compartment cytoplasm:

```
A(x)@Plm -> A(x)@Cyt
```

This is *adjacent-compartment molecular transport*.

Moving a molecule between adjacent compartments does not affect the postfix location tag of other molecules in the same species. For example, if a B molecule were connected to A, then it would stay in `Plm`. The above rule would generate the reaction (with transported molecule underlined for emphasis):

```
A(x,b!0)@Plm.B(a!0)@Plm -> A(x,b!0)@Cyt.B(a!0)@Plm k
```

An example where such a transport rule would be useful is to model membrane insertion of proteins, i.e., when a protein freely diffusing in the cytoplasm is transported to a membrane compartment. Adjacent-compartment molecular transport *cannot be extended to species transport* by using the @-prefix instead of the postfix.

Individual molecules can be moved from one volume to another volume through a bridging surface. For example, they can be moved between the cytoplasm and the nucleus, which are bridged by the nuclear membrane. This is *bridged-volume molecular transport*.

```
A(x)@Cyt -> A(x)@Nuc k
```

An example where such a transport mechanism would be useful is transport through a channel in the surface, such as the flow of chloride ions from the external volume to the cytoplasm through a chloride channel in the membrane.

Bridged-volume molecular transport has the potential to create topologically inconsistent complexes. Such reactions, if they could be generated from the reaction rules, will be automatically discarded. Unlike adjacent-compartment transport, bridged-volume molecular transport *can be extended* to *bridged-volume species transport* by using the @-prefix instead of the postfix. This would simply move the entire species matched by the pattern as long as it is fully contained in the bridged-volume.

For example, consider the rule:

```
@Cyt:A(x) -> @Nuc:A(x) k
```

When this rule is applied to the `@Cyt:A(x,b!0).B(a!0)` species, the following reaction is generated:

```
A(x,b!0)@Cyt.B(a!0)@Cyt -> A(x,b!0)@Nuc.B(a!0)@Nuc k
```

As one can see, the location tags of all molecules in the species have been converted from `Cyt` to `Nuc`.

The reaction mechanism does not have to be unimolecular, for example, a valid bridged-volume transport rule would be:

```
@Cyt:A(x)+B()@Num -> @Nuc:A(x) + B()@Num k
```

Here, a molecule `B()` on the bridging surface `Num` enables the transport of some species matching `A(x)` between the bridged volumes `Cyt` and `Nuc`. The rate constant k is bimolecular rate constant scaled only by the Avogadro number.

An example where a bridged-volume transport mechanism could be useful is when target proteins bind transport proteins and the resulting complex is compact enough to penetrate the pores on the nuclear membrane. Bimolecular mechanisms are useful to model a situation, where the rate of transport depends on the number of pores or channels present.

A slightly more complicated form of bridged-volume transport is when a "fixed" molecule is on a surface and is connected to a "transporting" molecule in a volume. Now, when the transporting molecule is moved across the surface, it carries over any molecules (or "cargo") that are connected to it in the volume. To describe this transport mechanism, we use the `MoveConnected` keyword.

For example, if the fixed molecule were R on the nuclear membrane, the transporting molecule were T, and cargo molecules were A and B, one can imagine a situation where T is connected to R and is able to move between cytoplasm and nucleus. This can be described with the reaction rule:

```
T(r!0)@Cyt.R(t!0)@Num -> T(r!+)@Nuc.R(t!0)@Num k
  MoveConnected
```

The `MoveConnected` keyword would ensure that any molecules connected to T (and any molecules connected to those molecules and so on) would be moved too, i.e., it would generate the following reactions:

```
A(t!1)@Cyt.T(car!1,r!0)@Cyt.R(t!0)@Num ->\
  A(t!1)@Nuc.T(car!1,r!0)@Nuc.R(t!0)@Num k
B(t!1)@Cyt.T(car!1,r!0)@Cyt.R(t!0)@Num ->\
  B(t!1)@Nuc.T(car!1,r!0)@Nuc.R(t!0)@Num k
```

This is *bridged-volume connected transport*. In the absence of `MoveConnected` option, the A and B molecules would retain their

`Cyt` tags creating a topologically inconsistent complex. BNG would then discard the reaction.

This mechanism is useful to model situations, where the transported molecule forms relatively stable associations with the pore that is necessary for the transport process. For example, glucose transporters need to form stable complexes with the glucose molecules before they can be moved across a membrane.

The final form of transport is *bridged-surface connected transport*. In BNG, this is implemented in a manner consistent with endocytosis. Compartments cannot be dynamically created in BNG; hence, endocytosis is modeled as movement between preexisting surface compartments plasma membrane and endosomal membrane bridged by the cytoplasm. When implementing a transport between surfaces bridged by a volume compartment, cBNG does the following:

- The molecules in the rule on the starting surface are moved to the destination surface.
- The connected molecules directly on the starting surface are moved to the destination surface.
- The connected molecules in the bridging volume stay in the bridging volume.
- The connected molecules in the outer volume are moved to the inner volume and vice versa.

Consider the bridged-surface transport rule from plasma membrane to endosomal membrane:

```
R()@Plm -> R@Enm k
```

Using the hierarchy of compartments in Fig. 5, cBNG interprets the rule to mean:

`Plm`— starting surface

`Enm`—destination surface

`Ext`—outer volume (parent of `Plm`)

`Cyt`—bridging volume (child of `Plm` and parent of `Enm`)

`End`—inner volume (child of `Enm`)

Let us see how the rule affects the following species that spans the plasma membrane, i.e., it has molecules in external compartment, plasma membrane, and cytoplasm:

```
A(r!0)@Ext.R(a!0,b!1)@Plm.B(r!0)@Cyt
```

Molecules in `Plm` (starting surface) are moved to `Enm` (destination surface)
Molecules in `Ext` (outer volume) are moved to `End` (inner volume).
Molecules in `Cyt` (bridging volume) remain in `Cyt`.

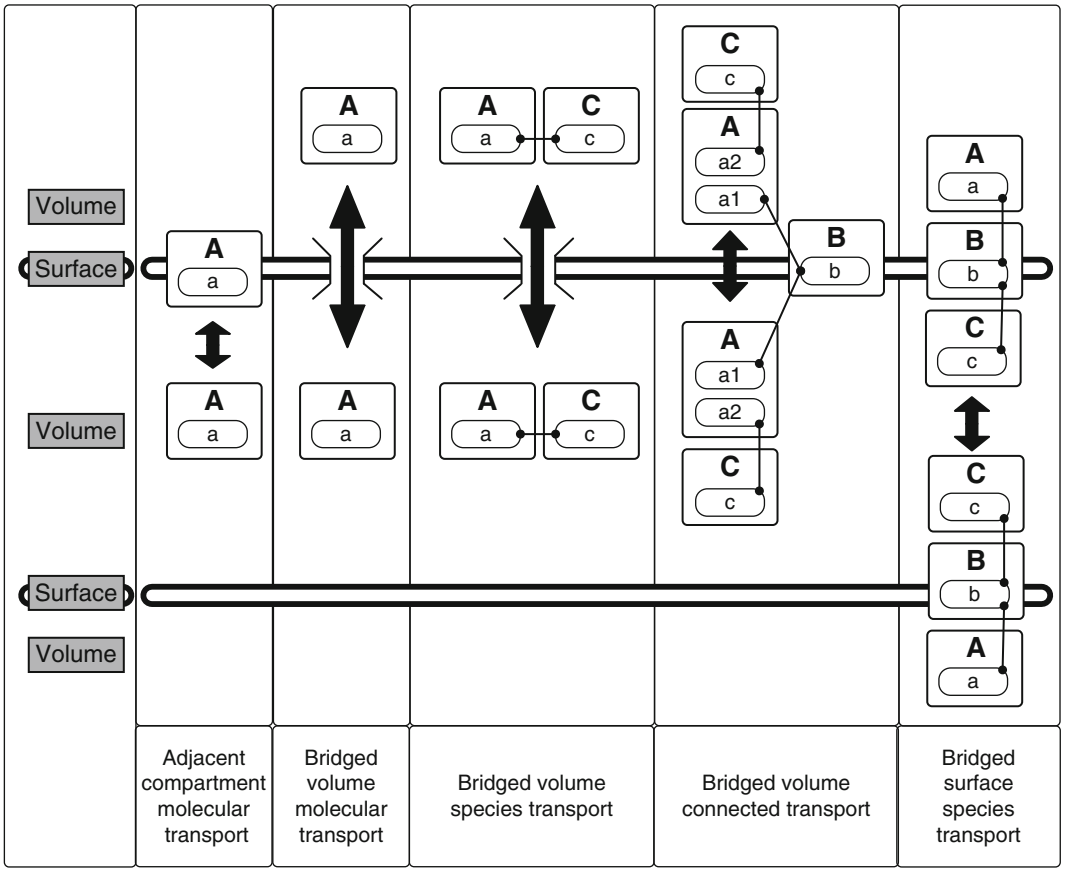

Fig. 7. Transport. The different types of transport mechanisms available in cBNG are illustrated. The details can be found in Subheading 7.4.4.

In other words, the above species would be converted to:

```
A(r!0)@End.R(a!0,b!1)@Enm.B(r!0)@Cyt
```

The different forms of allowed transport mechanisms are described in Fig. 7.

8. Biological Parameterization

One of the problems with modeling detailed biochemistry is the combinatorially large requirement of rate parameters. Rule-based modeling alleviates this requirement to a large extent by assigning identical parameters to reactions modeled by a single rule. This makes the number of required kinetic parameters proportional to the number of rules in the model. Where possible, these parameters are obtained either directly or estimated from experiments detailed in the biochemical literature.

Should the literature prove insufficient, certain limited assumptions can be made about parameters using general biophysical and biochemical knowledge. These assumptions can also be tested for their impact on a model using parameter sensitivity analysis methods (45). In this section, we demonstrate approaches to estimate, calculate, and represent important parameters of signal transduction models, such as cell geometry, concentrations, reaction rate constants, and equilibrium constants. This section can be skipped by the experienced modeler familiar with biological parameters and their relevance to reaction kinetic modeling.

8.1. Volumes

As mentioned in the Subheading 5.1, the volume of the compartment in which a reaction occurs directly affects the microscopic reaction rate constant. Therefore, before modeling a system, one must specify the volumes of the compartments in which the reactions occur. Consider an example parameterization used for the model in the Subheading 6.

We are studying the effects of an external growth factor on a eukaryotic cell line. For modeling purposes, we assume a single cell of volume V_{cell} suspended in a liquid milieu of volume V_{ext}. For Chinese Hamster Ovary cells, an experiment determining the cell size distribution (46) indicates that the mean cell volume (V_{cell}) is 1.2 picoliters (pl) and that the cell suspension has a cell density of 6.15×10^5 cells/ml. Inverting this value, we can estimate V_{ext} as 1.6 nl. The cell is assumed to be divided into the cytosol and nucleus in a 5:1 ratio (from ref. 47 similar to assumptions in ref. 48). This enables us to calculate the compartment volumes V_{cyt} as 1 pl and V_{nuc} as 0.2 pl.

Secondly, we wish to model reactions on the cell membrane, which can be assumed to be a flat sheet containing freely diffusing molecules. These surface interactions are scaled by the 2D surface area rather than the 3D volume. To arrive at these values, we can assume that the cell is a sphere.

$$\text{Volume } V_{\text{cell}} = 1.2\,\text{pl} = 1.2 \times 10^{-12}\text{L} = 1.2 \times 10^{-15}\text{m}^3$$

$$\Rightarrow \text{Radius } r = \left(\frac{V_{\text{cell}}}{4\pi/3}\right)^{(1/3)} = 6.6 \times 10^{-6}\text{m} = 6.6\,\mu\text{m}$$

$$\Rightarrow \text{Surface Area } S_{\text{mem}} = 4\pi r^2 = 5.5 \times 10^{-10}\,\text{m}^2 = 5.5 \times 10^{-8}\,\text{dm}^2.$$

To calculate a surface "volume," one can approximate membrane thickness h, say 10 nm (or 10^{-7} dm), encompassing the membrane and all the proteins attached to the membrane on either side.

$$h = 10\,\text{nm} = 10\ \times 10^{-9}\,\text{m} = 10^{-7}\,\text{dm}$$

$$\Rightarrow \text{Membrane Volume } V_{\text{mem}} = S_{\text{mem}}\ h = 5.5 \times 10^{-15}\,\text{dm}^3$$

$$= 5.5\ \times 10^{-15}\,\text{L} = 5.5\,\text{fl}.$$

Although they are 2D interactions, surface reaction parameters are often approximated from their solution-phase values, which are reported in 3D units (i.e., on a per-volume basis). The membrane thickness approximation is useful to interconvert the 2- and 3D units. Note that the units used for each of the parameters have to be consistent and interconvertible. We considered membrane thickness in dm and surface area in dm^2. By simply multiplying these values, we directly get the volume in dm^3, which is equivalent to liters.

The ratio of V_{cyt} to V_{mem} (here, ~182) is an important consideration when modeling membrane reactions because it provides a measure of the acceleration of membrane reactions versus cytosolic reactions (13). This acceleration is purely a concentration effect, which arises from membrane reactions being restricted to a small volume enveloping the membrane, whereas cytosolic reactions are distributed throughout the large cytosolic volume.

8.2. Concentrations

Understanding the relationship between concentration and population number is of paramount importance in signal transduction models. A realistic model is one, where the concentrations of the molecules and complexes stay within the biologically acceptable ranges. These ranges can be quite wide or narrow, depending on the type of molecule. Even within a biological context, concentrations of specific molecular species can change dramatically during signaling, often over an order of magnitude. To ensure biological relevance, the modeler must provide starting concentrations that reflect the acceptable concentration ranges.

Concentration is traditionally defined as the number of moles of a molecule (or complex) per unit volume of the container (expressed in liters). A mole equals 6.02214×10^{23} individual entities.

$$\text{Concentration (M)}\ C = \frac{N}{N_{\text{Avo}}\, V},$$

where N is number of entities (no unit), N_{Avo} is Avogadro number $= 6.02214 \times 10^{23}$/mol, V is volume (L).

To provide a sense of scale, one can keep in mind that in a volume of 1 pl, 1 nM concentration equals 602 molecules, but 1 μM equals 602,214 molecules.

Similarly, for the same number of molecules, the concentration can be large or small depending on the volume of the compartment. For example, 10^4 molecules per cell is 16.6 nM in the cytosol of volume 1 pl, but 3.02 μM in the membrane compartment of volume 5.5 fl.

Depending on the biology being modeled, different molecule types in the model can have drastically different concentration ranges. A global protein profiling study of yeast cells (49) shows a log-normal distribution with some proteins being as few as 50

molecules per cell and some as high as a million per cell. Most proteins are found at levels of many hundreds to a few thousands of molecules per cell. Eukaryotic cells are typically larger than yeast cells, but they also exhibit high variability in size and shape.

Typical concentration/population ranges for signal transduction models (from low to high) include the following:

1–10 molecules	Genes
10–100 molecules	mRNA transcripts
nM concentrations	Typical signal transduction proteins
μM concentrations	Typical secreted molecules
mM concentrations	Free triphosphate nucleotides (ATP and GTP), Ca^{2+}

Another effect of extremely small volumes (or extremely high concentrations) is molecular crowding. This can cause a breakdown of the well-mixed infinite dilution assumption, especially for bulky proteins. Crowding tends to reduce reaction probabilities and increase spatial heterogeneity because the reaction probability becomes more dependent on local conditions. Crowding can also affect assumptions made about other parameters, such as rate constants and equilibrium constants. The problem of identifying crowding effects in kinetic models is an active research area (50, 51). For a BNGL model, the modeler should be aware of the caveat that significant crowding can affect the rate constants that apply inside cells in comparison to those that are measured using purified proteins or cell extracts.

8.3. Equilibrium Constants

Consider a typical reversible bimolecular reaction. The ratio of the forward and reverse rate constants is referred to as the *equilibrium association constant* K_a and the reciprocal is referred to as the *equilibrium dissociation constant* K_d.

$$A + B \underset{k_r}{\overset{k_f}{\rightleftharpoons}} AB.$$

$$K_a = \frac{1}{K_d} = \frac{k_f}{k_r}.$$

At equilibrium, the flux in both directions is equal.

$$k_f[A]_{eq}[B]_{eq} = k_r[AB]_{eq} \Rightarrow K_a = \frac{1}{K_d} = \frac{k_f}{k_r} = \frac{[AB]_{eq}}{[A]_{eq}[B]_{eq}}.$$

Thus, the equilibrium constant is the ratio of the concentration of products and reactants at equilibrium. Suppose at equilibrium, bound and unbound concentrations of reactant A are equal (i.e., half of A is bound and half of A is free). This would restrict the value of unbound concentration of reactant B.

$$[A]_{eq} = [AB]_{eq} \Rightarrow K_d = [B]_{eq}.$$

Similarly,

$$[B]_{eq} = [AB]_{eq} \Rightarrow K_d = [A]_{eq}.$$

Thus, the dissociation equilibrium constant is also the concentration of one reactant for which the other reactant is half-bound at equilibrium. The dissociation equilibrium constant is also that concentration near which the changes in concentration of the bound complex are the largest.

Because of the above relations, the equilibrium constant (in either definition) is a powerful way to quantify the strength of a reversible association. The dissociation equilibrium constant has units of concentration and is often referred to as the *binding affinity*. The lower the value of K_d, the higher the strength of the interaction. Reversible protein–protein and protein–small molecule interactions have very wide ranges of affinities:

$K_d \approx$ (fM)	Extremely strong protein–inhibitor interactions
$K_d \approx$ (pM)	Typical strong interactions
$K_d \approx$ (nM)	Typical moderate interactions
$K_d \approx$ (μM)	Typical weak interactions
$K_d \approx$ (mM)	Extremely weak nonspecific interactions

Most functional protein–protein interactions in signal transduction are in the nM–μM range. Equilibrium constants are some of the most widely available types of experimental data and are often the first target of biophysical experiments if an interaction is hypothesized between two proteins. When using available literature values, the modeler must be aware of experimental caveats, such as use of truncated or chimeric proteins, non-physiological experimental conditions, etc.

8.4. Rate Constants

BNG requires individual microscopic rate constants for each interaction, i.e., *per-site asymmetric* rate constants. The conversion from macroscopic to microscopic rate constants is explained in the Subheading 5.1. In this section, we focus on how to arrive at the macroscopic constants themselves.

Often, these parameters are not explicitly determined or available. They are usually available indirectly in the form of equilibrium constants or Michaelis–Menten fits. Due to the unconstrained nature of the relationships between them, assumptions must often be made when modeling explicit reactions.

Consider the bimolecular association reaction described earlier.

$$A + B \underset{k_r}{\overset{k_f}{\rightleftharpoons}} AB.$$

The "forward" rate constant k_f, i.e., the bimolecular rate constant of association, is often estimated by theoretical derivation or Brownian dynamics simulations (reviewed in ref. 52). To simplify model parameterization in the face of insufficient data, a weak assumption can be employed on bimolecular association rate constants: similar sized molecules have similar diffusivities and therefore similar association rate constants under well-mixed conditions. The diffusion coefficient for small molecules in water places an upper limit on the possible diffusion coefficients for proteins and, consequently, the association rate constants. Typical ranges of values for the association rate constant (high to low) include the following (52):

10^{10} $M^{-1}s^{-1}$	Diffusion limit in water
10^{9}–10^{10} $M^{-1}s^{-1}$	Small molecule interactions
10^{7}–10^{9} $M^{-1}s^{-1}$	Protein–protein interactions involving long-range electrostatics
10^{6}–10^{7} $M^{-1}s^{-1}$	Typical protein–protein interactions
10^{3}–10^{6} $M^{-1}s^{-1}$	Interactions of bulky slowly diffusing proteins

Dissociation rate constants (k_r) are much harder to theoretically derive or estimate using simple assumptions. More often than not, they need to be experimentally verified. Dissociation rate constants can also vary over a much wider range than association rate constants for similarly sized molecules. If equilibrium constants are the only available data, a consistent approach to estimating dissociation rates is to assume a fixed association rate constant and compute the dissociation rate constant:

$$K_a = \frac{1}{K_d} = \frac{k_f}{k_r} \Rightarrow k_r = \frac{k_f}{K_a} = k_f K_d.$$

In the case of enzymatic catalysis, interactions are typically reported using the implicit enzyme parameters k_{cat} and K_M which are estimated under controlled conditions. However, as explained in the Subheading 5.3, in large and dense biological networks, it is preferable to model the reactions explicitly using the rate parameters k_f, k_r, and k_{cat}. The three explicit rate parameters can be derived from the two implicit enzyme parameters by making an additional assumption. To be consistent with our previous approach, the forward rate constant k_f can be fixed and the reverse rate estimated:

$$K_M = \frac{k_r + k_{cat}}{k_f} \Rightarrow k_r = K_M k_f - k_{cat}.$$

When determined experimentally, the dissociation rate k_r (U/s) is often expressed in terms of the stability of the formed complex. A typical quantification is the half-life $\tau_{1/2}$ which denotes the time taken (s) for half the complexes to dissociate or the exponential decay constant τ_e (s) which denotes the time taken for the concentration of the complex to decay to $1/e$ of the original value (Euler number $e = 2.71828$). The dissociation rate is inversely proportional to these decay constants.

$$k_r = \frac{\ln 2}{\tau_{1/2}} = \frac{1}{\tau_e}.$$

9. Downstream Signaling from the Membrane

In Subheading 6, we created and simulated a detailed model of receptor–ligand interactions. In this section, we broadly treat the rest of the signal transduction paradigm (see Fig. 8).

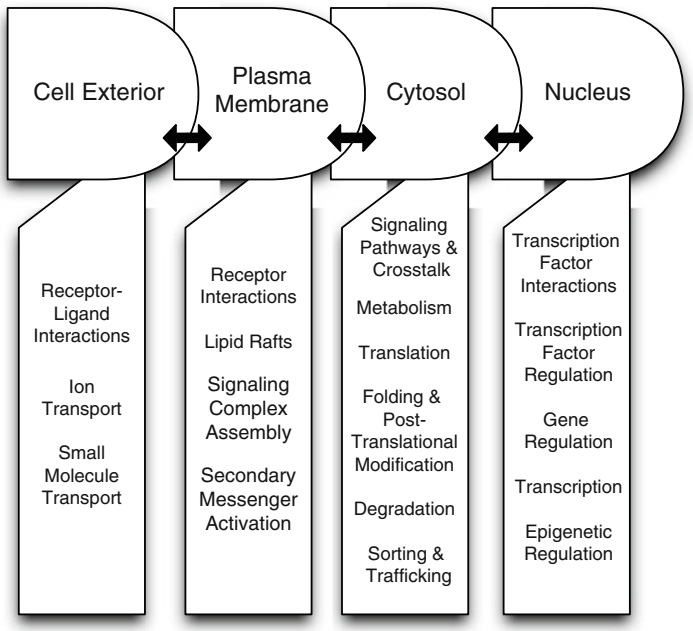

Fig. 8. General paradigm of signal transduction. At each compartmentalization of the cell, numerous processes of various types occur. These processes interact with each other and also with processes in adjacent compartments through means of transport. In general, a signal from the exterior is detected at the membrane, transduced through the cytoplasm, and concludes in gene regulation. This changes the transcription profile, subsequently affecting the protein distribution and can lead to a phenotypic change of state for the entire cell.

Signaling at the membrane causes the recruitment of molecules into an active signaling complex which possesses or modifies catalytic activities to change the distribution of messenger molecules in the cytosol and membrane. These messengers subsequently recruit and initiate signaling cascades. The signaling cascades interact with each other through shared molecules and usually terminate with the activation/attenuation of transcription factors in the nucleus. These transcription factors initiate/terminate synthesis of new molecules which can then modify the network structure to adapt to the incoming stimulus.

Here, we focus only on the EGFR (ErbB1) and the signals recruited by its homo-dimerization. One should keep in mind that ErbB1 is only one member of the growth factor-activated ErbB kinases and that homo- and hetero-dimerization result in a typically dense and intricate network that is still under experimental scrutiny. We have attempted to capture many of the essential mechanisms here that can be easily extended or modified.

9.1. Assembling the Signaling Complex

The assembly of the signaling complex at the membrane is a critical step linking the ligand binding that happens outside the cell to the signaling inside the cell. The EGFR is activated by ligand-induced enhancement of dimerization that is modeled in Subheading 6. The dimer formation introduces a conformational change that activates the tyrosine kinase domains on the receptor. These kinase domains then phosphorylate specific tyrosines on the receptor.

These phosphorylated tyrosine sites then form specific docking sites for several proteins that contain SH2, SH3, and PTB domains. Some recruited proteins, such as Grb2, Gab1, and Shc1, function as adaptors and scaffolds to facilitate recruitment of other proteins to the signaling complex. The receptor tyrosine kinase can also activate binding sites on these adaptors and scaffolds. Among the recruited proteins are several enzymes that perform one of the following two functions: (a) activate/deactivate other binding sites in the complex (typically through phosphorylation/dephosphorylation (53)) or (b) being themselves activated by the recruitment process, and change the distributions of messenger molecules in the cytosol and membrane.

While utilizing biochemical literature to formalize the recruitment interactions, it is important to distinguish sequential (processive) and parallel (distributive) interactions (54). *Sequential* interactions are those where every interaction is a prerequisite for the next interaction. *Parallel* interactions are those that are independent of each other, occurring on different regions of the same molecule. The typical phenomenological description of the recruitment process is often misleading, making it seem sequential, when in fact some interactions can be independent and parallel.

For example, a typical phenomenological statement might be "Phosphotyrosines on the receptor recruit Grb2 which in turn

recruits Sos1." This makes it seem as if the Sos1–Grb2 interaction is dependent on the Grb2–phosphotyrosine interaction. A closer look at the structure and function of Grb2 shows that this implied linearization is false and that the Grb2–Sos1 and Grb2–phosphotyrosine interactions are independent of each other. The statement is then an observation of just one outcome of these two parallel interactions, which is Grb2 bound to both Sos1 and a receptor phosphotyrosine. It ignores the other two outcomes, where Grb2 is bound to only one or the other.

The older modeling literature is replete with such subjective interpretations and rule-based modeling was developed so that such arbitrary choices can be avoided (54). Structural information on different molecules is available in several databases, such as PDB (55) and UniProt (56), that can be exploited to model the interactions rigorously and accurately. An intimate knowledge of the structural nature of the interactions helps in determining which interactions are sequential and which are not.

A good way of distinguishing a sequential binding interaction is to ask the question: "Does a new binding site need to be created/activated for this interaction to occur?" If the answer is yes, then the interaction is sequential to the previous interaction. If the answer is no, then it is independent of the previous interaction. An intimate knowledge of the structural aspects of the proteins is very useful in this aspect. In the absence of conclusive evidence, it is better to assume that interactions are distributive/parallel rather than processive/sequential.

Knowledge Base:

> The binding of EGF ligand to EGFR receptor is cooperative with the homo-dimerization of EGFR. Ligand binding enhances dimerization, but ligand-binding sites on the dimer are negatively cooperative.

The modeling of this information is treated in the Subheading 6.

Knowledge Base:

> Dimerized EGFR autophosphorylates on several tyrosines in its amino acid sequence, including Y998, Y1016, Y1092, Y1110, Y1138, Y1172, and Y1197 (57). 2030

We model the phosphorylation with identical kinetics, although we could potentially use a different rate constant for each phosphorylation.

```
R(dim!+,Y998~0)  -> R(dim!+,Y998~p)    k_ph
R(dim!+,Y1016~0)  -> R(dim!+,Y1016~p)  k_ph
R(dim!+,Y1092~0)  -> R(dim!+,Y1092~p)  k_ph
R(dim!+,Y1110~0)  -> R(dim!+,Y1110~p)  k_ph
R(dim!+,Y1138~0)  -> R(dim!+,Y1138~p)  k_ph
R(dim!+,Y1172~0)  -> R(dim!+,Y1172~p)  k_ph
R(dim!+,Y1197~0)  -> R(dim!+,Y1197~p)  k_ph
```

Knowledge Base:

Dimerized EGFR kinases are also responsible for activation of phosphorylation sites on Gab1 and Shc1 if they are present in the same complex. These include Y627 on Gab1 (58), tandem YXXP and YXXM motifs on Gab1 (58), and Y317 on Shc1 (59).

We can use a dot operator to show that these molecules are in the same complex when they are modified, even though there can be several ways in which the molecules could be recruited.

```
R(dim!+).Gab1(YXXP~0) -> R(dim!+).Gab1(YXXP~p) k_ph
R(dim!+).Gab1(YXXM~0) -> R(dim!+).Gab1(YXXM~p) k_ph
R(dim!+).Gab1(Y627~0) -> R(dim!+).Gab1(Y627~p) k_ph
R(dim!+).Shc1(Y317~0) -> R(dim!+).Shc1(Y317~p) k_ph
```

Knowledge Base:

If the molecule Shp2 is recruited to the complex, it opposes the phosphorylation activity of the receptor tyrosine kinases (60).

In the absence of a dephosphorylation event, the phosphorylation sites will be saturated to phosphorylation even with a tiny amount of dimerization, as explained in Subheading 5.2. Shp2 recruited to the complex can dephosphorylate any of the kinase substrates:

```
Shp2().R(Y998~p) -> Shp2().R(Y998~0)       k_deph
Shp2().R(Y1016~p) -> Shp2().R(Y1016~0)     k_deph
Shp2().R(Y1110~p) -> Shp2().R(Y1110~0)     k_deph
Shp2().R(Y1138~p) -> Shp2().R(Y1138~0)     k_deph
Shp2().R(Y1172~p) -> Shp2().R(Y1172~0)     k_deph
Shp2().R(Y1197~p) -> Shp2().R(Y1197~0)     k_deph
Shp2().Gab1(YXXP~p) -> Shp2().Gab1(YXXP~0) k_deph
Shp2().Gab1(YXXM~p) -> Shp2().Gab1(YXXM~0) k_deph
Shp2().Gab1(Y627~p) -> Shp2().Gab1(Y627~0) k_deph
Shp2().Shc1(Y317~p) -> Shp2().Shc1(Y317~0) k_deph
```

Knowledge Base:

The receptor phosphotyrosines recruit proteins in the following manner: Y1110, Y1138, Y1172, and Y1197 bind both Grb2 and Shc1. Y1092 binds only Grb2. Y1016 binds only Shp2. Y998 binds both Shp2 and Shc1 (57). All binding interactions are reversible.

The interaction with the receptor is localized to the SH2 domain on Grb2. The N-terminal regions of Shc1, Shp2, and Rasa1 all contain SH2/SH3/PTB domains in some combination that enables binding to specific phosphotyrosines. They are modeled as generic "n" domains, referring to their N-terminal position.

```
# Binding of Y1110, Y1138, Y1172 and Y1197
R(Y1110~p) + Grb2(sh2) <-> R(Y1110~p!1).Grb2(sh2!1)
  kf_pY_grb2,kr_pY_grb2
R(Y1138~p) + Grb2(sh2) <-> R(Y1138~p!1).Grb2(sh2!1)
  kf_pY_grb2,kr_pY_grb2
R(Y1172~p) + Grb2(sh2) <-> R(Y1172~p!1).Grb2(sh2!1)
  kf_pY_grb2,kr_pY_grb2
R(Y1197~p) + Grb2(sh2) <-> R(Y1197~p!1).Grb2(sh2!1)
  kf_pY_grb2,kr_pY_grb2
R(Y1110~p) + Shc1(n) <-> R(Y1110~p!1).Shc1(n!1)
  kf_pY_shc1,kr_pY_shc1
R(Y1138~p) + Shc1(n) <-> R(Y1138~p!1).Shc1(n!1)
  kf_pY_shc1,kr_pY_shc1
R(Y1172~p) + Shc1(n) <-> R(Y1172~p!1).Shc1(n!1)
  kf_pY_shc1,kr_pY_shc1
R(Y1197~p) + Shc1(n) <-> R(Y1197~p!1).Shc1(n!1)
  kf_pY_shc1,kr_pY_shc1

# Binding of Y1092
R(Y1092~p) + Grb2(sh2) <-> R(Y1092~p!1).Grb2(sh2!1)
  kf_pY_grb2,kr_pY_grb2

# Binding of Y1016
R(Y1016~p) + Shp2(n) <-> R(Y1016~p!1).Shp2(n!1)
  kf_pY_shp2,kr_pY_shp2

# Binding of Y998
R(Y998~p) + Shc1(n) <-> R(Y998~p!1).Shc1(n!1)
  kf_pY_shc1,kr_pY_shc1
R(Y998~p) + Shp2(n) <-> R(Y998~p!1).Shp2(n!1)
  kf_pY_shp2,kr_pY_shp2
```

A contact map with defined symbols for components, interactions, and influences is useful in summarizing these reaction rules. The conventions followed for the contact maps in this tutorial are

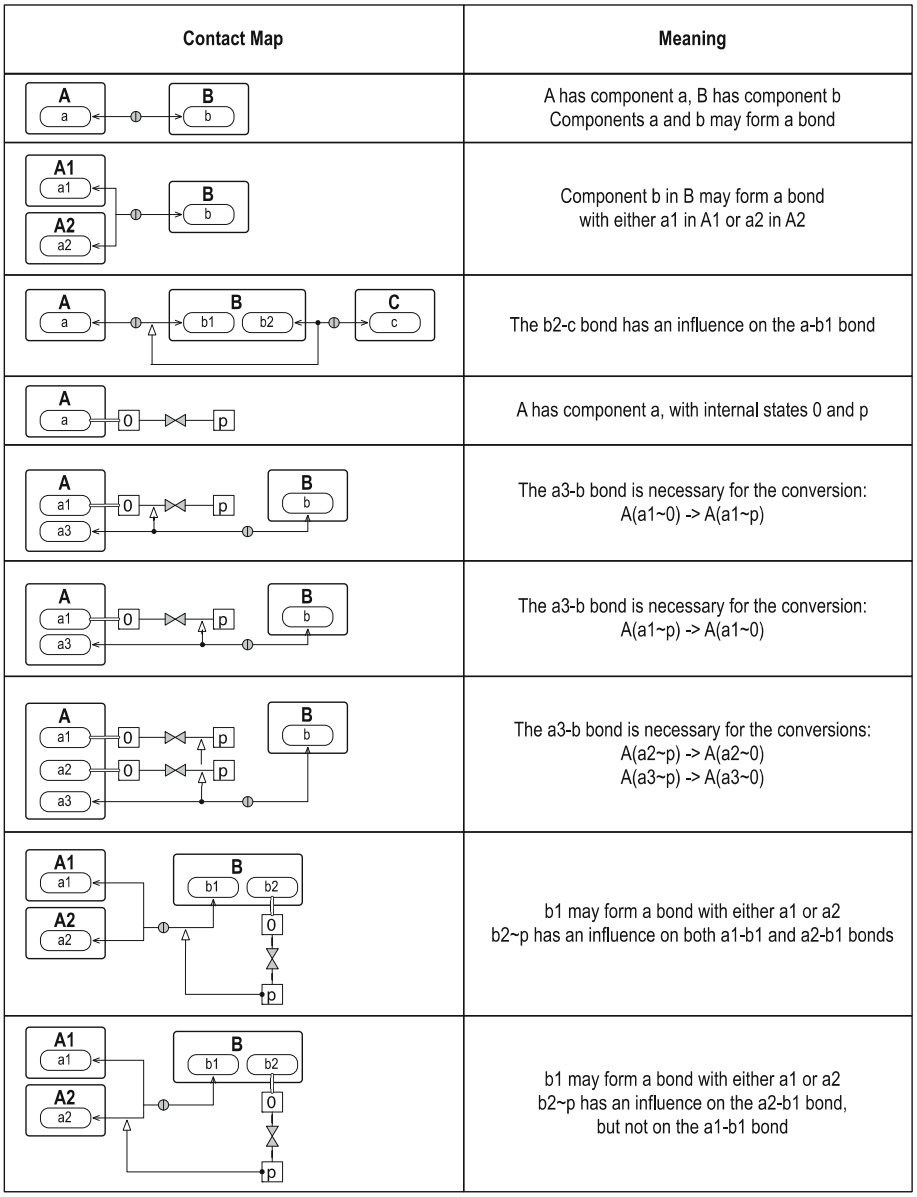

Fig. 9. Contact map explanation. *Square boxes* are molecules and *curved boxes* are components. The *bond symbol* is a *crossed out circle* with *arrows* on both sides and represents a potential reversible bond between components. The different permutations of bonds can be expressed concisely by combining the *arrows* of the *bond symbol*. Influences are shown by *unidirectional* or *bidirectional arrows* with *white solid arrowheads*. Unidirectional influences always begin on a *solid dot*. The *gray bowtie symbol* indicates a reversible state change. An influence *arrow* terminating on one side of the bowtie is a catalyst to convert the state to the other side of the bowtie. Influences can terminate on bond interactions or on other influences.

shown in Fig. 9. A contact map of the kinase and phosphatase activities in the signaling complex and the primary binding interactions with the receptor are shown in Fig. 10.

The molecules recruited to the primary phosphotyrosines on the receptor have complex interactions among themselves. Grb2 and Shc1 are adaptor proteins that bind each other as well as other proteins. Gab1, a protein recruited via Grb2, is a scaffold protein with multiple binding sites.

Knowledge Base:

Grb2 has three domains SH2, SH3N, and SH3C. In addition to receptor phosphotyrosines binding, the SH2 domain is capable of binding phosphorylated Y317 of Shc1 (59). The SH3N domain recruits Sos1 through the N-terminal domain of Sos1 (61). The SH3C domain recruits Gab1 through a proline-rich region on Gab1 (58). Certain serines on Sos1 (62) and certain serines and threonines on Gab1 (58) need to be unphosphorylated for the Grb2 binding to be effective.

```
Grb2(sh2) + Shc1(Y317~p) <->
Grb2(sh2!1).Shc1(Y317~p!1) kf_grb2shc1,kr_grb2shc1
Grb2(sh3n) + Sos1(S~0,n) <-> Grb2(sh3n!1).Sos1(S~0,n!1)
  kf_grb2sos1,kr_grb2sos1
Grb2(sh3c) + Gab1(ST~0,pro)<->Grb2(sh3c!1).Gab1
  (ST~0,pro!1) kf_grb2gab1,kr_grb2gab1
```

Knowledge Base:

The Gab1 scaffold protein contains tandem YXXP motifs that bind the N-terminal domain of Rasa1 on phosphorylation. Similarly, a phosphorylated tandem YXXM motif binds the regulatory subunit p85 of PI3K. Phosphorylated Y627 recruits the Shp2 phosphatase. All of these binding events require that certain serines and threonines on Gab1 remain unphosphorylated (58).

```
Gab1(ST~0,YXXP~p)+Rasa1(n)<->Gab1(ST~0,YXXP~p!1).
  Rasa1(n!1) kf_gab1rasa1,kr_gab1rasa1
Gab1(ST~0,YXXM~p)+PI3K(p85)<->Gab1(ST~0,YXXP~p!1).
  PI3K(p85!1) kf_gab1pi3k,kr_gab1pi3k
Gab1(ST~0,Y627~p)+Shp2(n)<->Gab1(ST~0,YXXP~p!1).
  Shp2(n!1) kf_gab1shp2,kr_gab1shp2
```

The secondary binding interactions between the molecules recruited to the receptor are summarized in Fig. 11. The contact

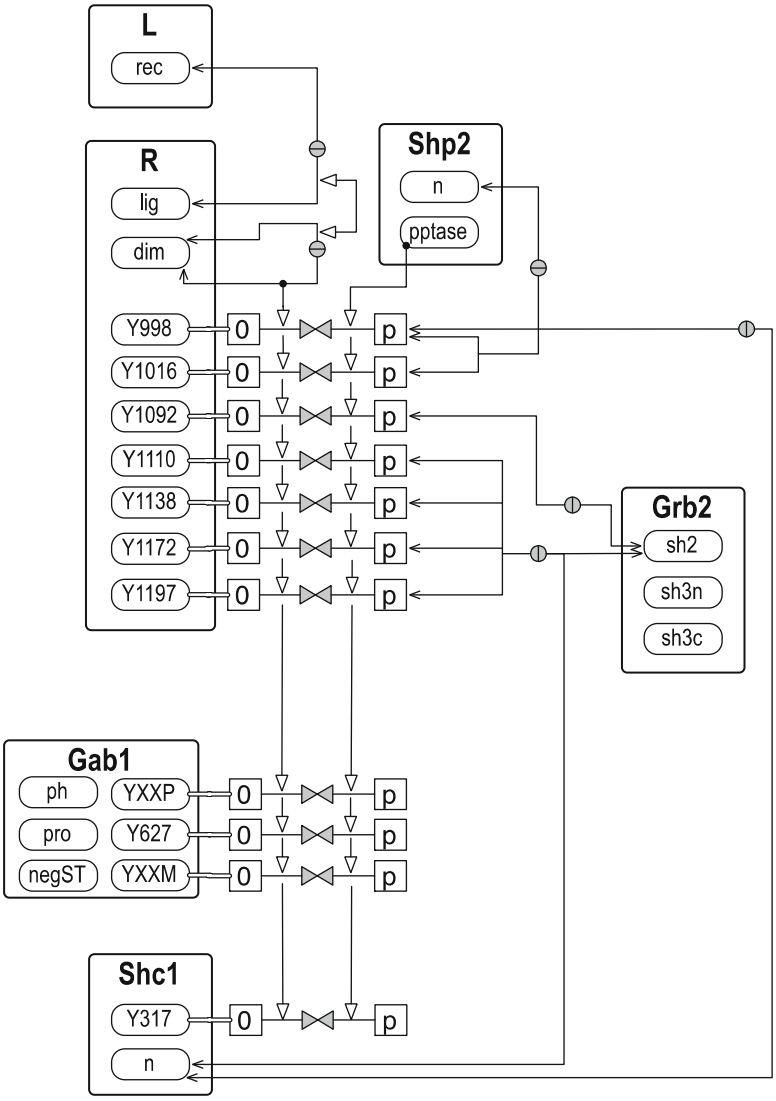

Fig. 10. Signal complex assembly, receptor interactions. This encapsulates the direct interactions of the receptor with other molecules. The receptor can bind the ligand, dimerize with another receptor, and the dimerized state activates the kinase activity of the receptor, resulting in phosphorylation of the many tyrosines on its tail. The kinase activity is limited to not only its own tyrosines, but also tyrosines on other molecules that may be in the complex, such as Gab1 and Shc1. The phosphotyrosines can bind and recruit other molecules. One of the recruits is a phosphatase Shp2 that dephosphorylates the tyrosines in the same complex.

map in Fig. 11 is to be considered complementary to the contact map in Fig. 10 and not exclusive.

9.2. Secondary Messenger Activation

As mentioned earlier, certain catalytic activities are activated on recruitment to the signaling complex. These catalytic activities rapidly change the distributions of certain effector molecules in

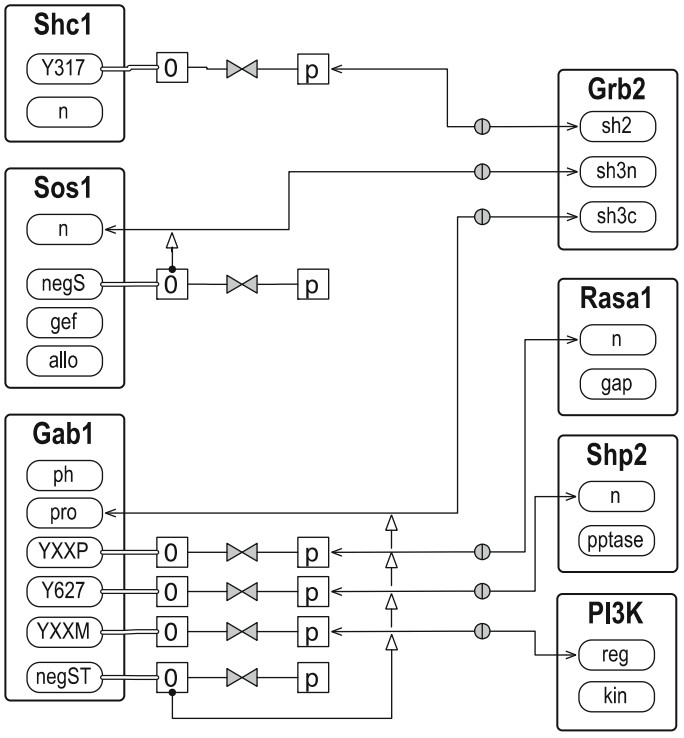

Fig. 11. Signal complex assembly, recruit interactions. The recruited molecules in the EGFR themselves have complicated binding properties with respect to each other. Certain serines on Sos1 and serines and threonines on Gab1 have a negative effect on the binding properties of Sos1 and Gab1, and this is utilized to create a feedback loop by molecules downstream of the signal.

the plasma membrane and cytosol. This manifests as a large phenotypic switch causing wholesale activation of corresponding cytosolic signaling pathways. In this section, we consider the activation of Ras, a GTPase switch, and PIP3, a phosphoinositide messenger.

9.2.1. PIP3 Activation

Phosphoinositides are small lipid molecules that function as secondary messengers in several eukaryotic pathways. Phosphoinositides can be interconverted by adding and removing phosphates onto the 3', 4', and 5' positions on the inositol moiety. The ratio and spatial distribution of the different phosphoinositides are exploited for signaling purposes. In growth factor receptor activation, the interconversion of phosphatidylinositol (4,5)-bisphosphate, also known as PIP2, and phosphatidylinositol (3,4,5)-trisphosphate is especially critical. PIP3 is maintained at very low levels by the activity of cytosolic PTEN phosphatase. PIP3 levels are drastically increased by recruitment of PI3K to the membrane. PIP3, thus, serves as a secondary messenger for activating several PH domain-containing proteins, an important one being Akt, which triggers the Akt/mTor pathway critical for

cell proliferation and differentiation. The information here can be found in ref. 63. The rules describing change of state have the appropriate component underlined for emphasis.

Knowledge Base:

PIP3 is maintained at low levels by conversion to PIP2 by PTEN phosphatase.

Here, we model PIP3 as a phosphoinositide with a 3' site targeted by the phosphatase.

```
PTEN(c2) + PI(3p~p) <-> PTEN(c2!1).PI(3p~0!1)
  kf_pip3_pten,kr_pip3_pten
PTEN(c2!1).PI(3p~0!1) -> PTEN(c2) + PI(3p~p) kcat_pten
R().PI3K(p110) + PI(3p~0) <->
```

Knowledge Base:

The p110 domain of PI3K catalyzes conversion of PIP2 to PIP3 when recruited to the membrane.

```
  R().PI3K(p110!1).PI(3p~0!1) kf_pip2_pi3k,kr_pip2_pi3k
PI3K(p110!1).PI(3p~0!1) -> PI3K(p110) + PI(3p~p)
  kcat_pi3k
Gab1(ph) + PI(3p~p) <-> Gab1(ph!1).PI(3p~p!1)
  kf_pip3_gab1,kr_pip3_gab1
```

Knowledge Base:

PIP3 recruits PH domain-containing proteins like Gab1 (58) and Akt to the membrane.

```
Akt(ph) + PI(3p~p) <-> Akt(ph!1).PI(3p~p!1)
  kf_pip3_akt, kr_pip3_akt

Gab1(ph!1,pro).PI(3p~p!1) + R().Grb2(sh3c)->
Gab1(ph,pro!2).Grb2(sh3c!2) + PI(3p~p) \
```

Knowledge Base:

PIP3-recruited Gab1 at the membrane can be recruited to the signaling complex easily.

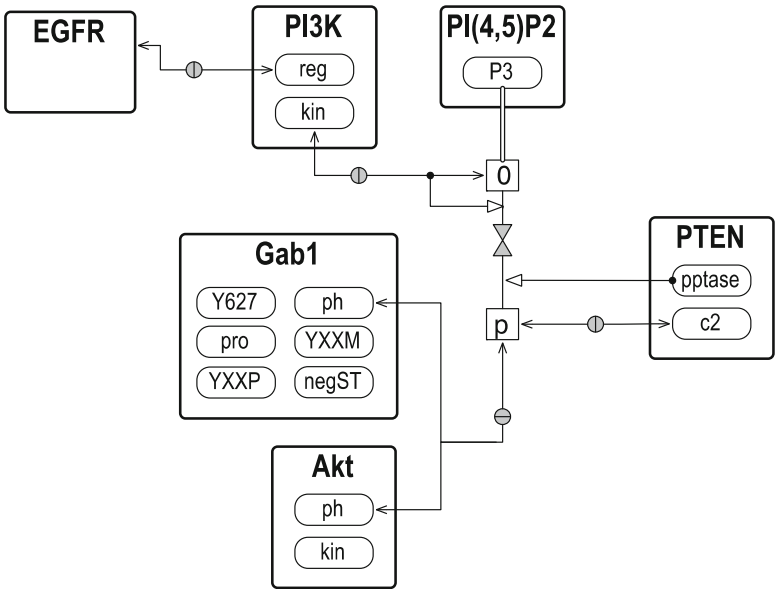

Fig. 12. PIP3 activation. PI(3,4,5)P2 or PIP3 is an important secondary messenger whose levels are kept at low basal levels by the phosphatase PTEN which catalyzes it to PI(4,5)P2 or PIP2. The third position on the inositol moiety is modeled as a separate component which can either be phosphorylated or unphosphorylated. PI3K is a kinase which can convert PIP2 to PIP3, but it is activated only by recruitment to the membrane by recruitment to the EGFR-signaling complex. PIP3 can then activate many other signaling cascades by recruitment, an important one being the kinase Akt. It can also strengthen the existing signals since Gab1 possesses a PH domain that can bind PIP3.

```
k_gab1grb2_mem
Akt(ph!+,kin) + mTor(S2448~0) <->
Akt(ph!+,kin!1).mTor(S2448~0!1) \
  kf_akt_mtor,kr_akt_mtor
Akt(ph!+,kin!1).mTor(S2448~0!1) -> Akt(ph!+,kin) +
  mTor(S2448~p) kcat_akt
```

Knowledge Base:

PIP3-recruited Akt is an active kinase with downstream substrates, such as mTOR.

The PIP3 activation and downstream interactions modeled here are illustrated in Fig. 12.

9.2.2. Ras Activation

Ras is a small GTPase protein that is tethered to the membrane and binds with high affinity to free guanidine nucleotides (GTP and GDP) in the cytosol. Ras, in its basal state, slowly catalyzes the dephosphorylation of GTP into GDP. In the absence of ligand-induced signal, the majority of Ras molecules are bound to GDP.

When bound by a guanidine nucleotide exchange factor (GEF), Ras releases the GDP nucleotide and binds a free GTP nucleotide. Since free GTP is in excess over free GDP under normal conditions, we can assume that the binding event after GDP release is always with a GTP. The Ras population switches from a GDP-bound majority to a GTP-bound majority.

Ras-GTP, by binding to other molecules, initiates many signaling cascades, including the MAPK cascade. The intrinsic GTPase activity can be enhanced by a GTPase-activating protein (GAP) to provide the opposite effect, i.e., to quickly revert a Ras-GTP population to Ras-GDP. EGFR recruits both Sos1 (a GEF) and Rasa1 (a GAP) to the membrane through adaptors and scaffolds.

Ras has three subtypes: HRas, NRas, and KRas with many common interactions among them. Here, we model HRas as an example. Instead of modeling the nucleotide as a separate molecule, we model it as simply a state on an HRas component. This is because the binding of GDP/GTP to HRas is very strong and does not reverse spontaneously. Any free HRas quickly binds to excess free nucleotide. The nucleotide-binding site is underlined for emphasis whenever a GTP–GDP or GDP–GTP transition occurs.

The following information is taken from ref. 61. The rules describing change of state have the appropriate component underlined for emphasis.

Knowledge Base:

HRas has a protein-binding site and a nucleotide-binding site. The nucleotide-binding site can be assumed to bind GTP or GDP strongly. HRas has innate GTPase activity.

```
HRas(pbs,nbs~gtp) -> HRas(pbs,nbs~gdp) k_hras_gtpase
```

Knowledge Base:

Receptor-bound Rasa1 is a GAP, i.e., it enhances the GTPase activity of HRas.

```
R().Rasa1(gap)+HRas(pbs,nbs~gtp) ->
R().Rasa1(gap!1).HRas(pbs!1,nbs~gtp) kf_rasa1_hras
Rasa1(gap!1).HRas(pbs!1,nbs~gtp) -> Rasa1(gap) +
HRas(pbs,nbs~gtp) kr_rasa1_hras
Rasa1(gap!1).HRas(pbs!1,nbs~gtp) -> Rasa1(gap) +
HRas(pbs,nbs~gdp) kcat_rasa1
```

Knowledge Base:

Sos1 recruited to the membrane can bind HRas at two places, an allosteric site and a catalytic site. Sos1 possesses GEF activity, i.e., it catalyzes the release of GDP from the HRas bound at the catalytic site. The GEF activity is enhanced by the presence of another HRas bound at the allosteric site. When GDP is released, GTP is assumed to bind immediately since it is in excess over GDP.

```
# HRas-GDP binding at GEF site
R().Sos1(gef)+HRas(pbs,nbs~gdp)->R().Sos1(gef!1).
  HRas(pbs!1,nbs~gdp) kf_sos1gef_hras Sos1(gef!1).
  HRas(pbs!1,nbs~gdp)-> Sos1(gef) + HRas(pbs,nbs~gdp)
  kr_sos1gef_hras

# HRas (both GTP and GDP) binding at allosteric site
R().Sos1(allo) + HRas(pbs) -> R().Sos1(allo!1).HRas
  (pbs!1) kf_sos1allo_hras
Sos1(allo!1).HRas(pbs!1) -> Sos1(allo) + HRas(pbs)
  kr_sos1allo_hras

# GEF activity in presence/absence of GTP at allo-site
HRas(pbs!1,nbs~gdp).Sos1(gef!1,allo)->  HRas(pbs,
nbs~gtp)+Sos1(gef,allo) kcat_sos1_1
HRas(pbs!1,nbs~gdp).Sos1(gef!1,allo!+)->HRas(pbs,
nbs~gtp)+Sos1(gef,allo!+) kcat_sos1_2
```

The modeled HRas activation is illustrated in Fig. 13.

9.3. Cytosolic Signaling—MAPK Cascade

The Ras proteins, on activation by GTP binding, go on to activate several pathways. One of the most widely studied Ras-activated pathways is the classical mitogen-activated protein kinase (MAPK) cascade. MAPKs are kinases that are heavily involved in growth-related transcription factor regulation. Previously, the paradigm of activation was that MAPKs were phosphorylated by MAP2Ks and MAP2Ks were phosphorylated by MAP3Ks, which in turn were activated by binding of Ras-GTP. However, as evidence accumulated, it became fairly obvious that the situation is not so simple. The presence of multiple subtypes, oligomerization, scaffolds, feedback mechanisms, and cell-specific conditions confound the study of this complex pathway.

As mentioned earlier, there are three subtypes of Ras proteins: HRas, KRas, and MRas. MAP3Ks are of three types too: Raf1, Raf2, and Raf3. They are subject to complex regulation from other signaling pathways as well as regulation by each other. Mek1 and Mek2 are MAP2K subtypes. MAPKs themselves have multiple families, such as the Erk (Erk1 and Erk2) and Jnk. Erk2

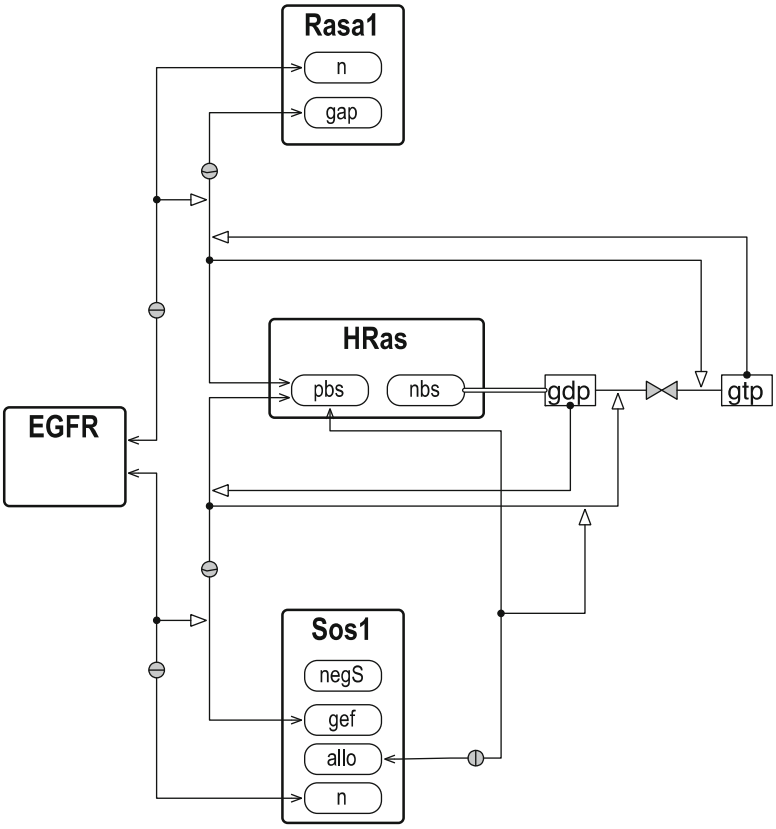

Fig. 13. Ras activation. Using HRas as an example, the complex interactions involved in Ras activation are shown. Rasa1 and Sos1 are recruited to the receptor in multiple ways (see Figs. 11 and 12). HRas is distributed on the membrane and comes into contact with Sos1 and Rasa1. Both Sos1 and Rasa1 have antagonistic activity towards HRas. Sos1 enables the quick release of bound HRas-GDP, letting HRas bind ubiquitous GTP and become activated. Rasa1 enhances the GTPase activity of HRas, speeding up the conversion of HRas-GTP to HRas-GDP.

can cause negative feedback by deactivating Gab1 and Sos1, early participants in Ras activation. The primary function of the MAPKs seems to be in the nucleus though, causing activation of growth-related transcription factors such as those of the AP1 family.

With these caveats, the model presented here omits many of the known details of the MAPK cascade and is provided to illustrate how the basic biochemistry can be encoded. The rules describing change of state have the appropriate component underlined for emphasis.

Knowledge Base:

HRas-GTP binds Raf1 and causes a conformational change (64). Raf1 is now an active kinase that can phosphorylate Mek1 (65) on S218 and S222. The phosphorylation is antagonized by the phosphatase PP2A (66).

```
# HRas-GTP binds Raf1
HRas(nbs~gtp,pbs) + Raf1(rbd) ->
HRas(nbs~gtp,pbs!1).Raf1(rbd!1) kf_hras_raf1
HRas(pbs!1).Raf1(rbd!1) -> HRas(pbs!1) + Raf1(rbd!1)
  kr_hras_raf1

# Ras-bound Raf1 is a kinase of Mek1
Raf1(rbd!+,kin) + Mek1(S218~0) ->
Raf1(rbd!+,kin!1).Mek1(S218~0!1) kf_raf1_mek1
Raf1(rbd!+,kin) + Mek1(S222~0) ->
Raf1(rbd!+,kin!1).Mek1(S222~0!1) kf_raf1_mek1
Raf1(kin!1).Mek1(S218~0!1) -> Raf1(kin!1) + Mek1
  (S218~0) kr_raf1_mek1
Raf1(kin!1).Mek1(S222~0!1) -> Raf1(kin!1) + Mek1
  (S222~0) kr_raf1_mek1
Raf1(kin!1).Mek1(S218~0!1) -> Raf1(kin) + Mek1
  (S218~p) kcat_raf1
Raf1(kin!1).Mek1(S222~0!1) -> Raf1(kin) + Mek1
  (S222~p) kcat_raf1

# PP2A is a phosphatase of Mek1
PP2A(pptase) + Mek1(S218~p) ->
PP2A(pptase!1).Mek1(S218~p!1) kf_pp2a_mek1
PP2A(pptase) + Mek1(S222~p) ->
PP2A(pptase!1).Mek1(S222~p!1) kf_pp2a_mek1
PP2A(pptase!1).Mek1(S218~p!1) -> PP2A(pptase) +
Mek1(S218~p) kr_pp2a_mek1
PP2A(pptase!1).Mek1(S222~p!1) -> PP2A(pptase) +
Mek1(S222~p) kr_pp2a_mek1
PP2A(pptase!1).Mek1(S218~p!1) -> PP2A(pptase) +
Mek1(S218~0) kcat_pp2a
PP2A(pptase!1).Mek1(S222~p!1) -> PP2A(pptase) +
Mek1(S222~0) kcat_pp2a
```

Knowledge Base:

Biphosphorylated Mek 1 is an active kinase that can phosphorylate Erk2 (65) on T185 and Y187. The phosphorylation is antagonized by the phosphatase Dusp1 (68).

```
# Biphosphorylated Mek1 is a kinase of Erk2
Mek1(S218~p,S222~p,kin) + Erk2(T185~0)->
Mek1(S218~p,S222~p,kin!1).Er2(T185~0!1) \
  kf_mek1_erk2
Mek1(S218~p,S222~p,kin) + Erk2(Y187~0)->
Mek1(S218~p,S222~p,kin!1).Er2(Y187~0!1) \
  kf_mek1_erk2
```

```
Mek1(kin!1).Erk2(T185~0!1) -> Mek1(kin!1) + Erk2
  (T185~0) kr_mek1_erk2
Mek1(kin!1).Erk2(Y187~0!1) -> Mek1(kin!1) + Erk2
  (Y187~0) kr_mek1_erk2
Mek1(kin!1).Erk2(T185~0!1) -> Mek1(kin) + Erk2
  (T185~p) kcat_mek1
Mek1(kin!1).Erk2(Y187~0!1) -> Mek1(kin) + Erk2
  (Y187~p) kcat_mek1

# Dusp1 is a phosphatase of Erk2
Dusp1(pptase) + Erk2(T185~p) ->
Dusp1(pptase!1).Erk2(T185~p!1) kf_dusp1_erk2
Dusp1(pptase) + Erk2(Y187~p) ->
Dusp1(pptase!1).Erk2(Y187~p!1) kf_dusp1_erk2
Dusp1(pptase!1).Erk2(T185~p!1) -> Dusp1(pptase) +
Erk2(T185~p) kr_dusp1_erk2
Dusp1(pptase!1).Erk2(Y187~p!1) -> Dusp1(pptase) +
Erk2(Y187~p) kr_dusp1_erk2
Dusp1(pptase!1).Erk2(T185~p!1) -> Dusp1(pptase) +
Erk2(T185~0) kcat_dusp1
Dusp1(pptase!1).Erk2(Y187~p!1) -> Dusp1(pptase) +
Erk2(Y187~0) kcat_dusp1
```

Signal transduction networks exhibit significant *positive and negative feedback*, i.e., downstream effectors doubling back and modifying (enhancing or attenuating) the properties of upstream signal proteins. This enables ultra-sensitivity in regulation and adaptive responses to incoming signals. Interesting dynamic phenomena, such as oscillations and travelling waves, are also observed in some cases due to such feedback (73).

Knowledge Base:

Biphosphorylated Erk2 can dimerize with other Erk2 (phosphorylated or unphosphorylated). The dimer is catalytically active on many substrates (53), including negative regulatory sites on Gab1 (58) and Sos1 (67).

```
Erk2(T185~p,Y187~p,dim) + Erk(dim) ->
Erk2(T185^p,Y187^p,dim!1) + Erk2(dim!1) kf_erk2_dim
Erk2(dim!1).Erk(dim!1) -> Erk2(dim) + Erk(dim)
  kr_erk2_dim
Erk2(dim!+,kin) + Gab1(ST^0) ->
Erk2(dim!+,kin!1).Gab1(ST^0!1) kf_erk2_gab1
Erk2(kin!1).Gab1(ST^0!1) -> Erk2(kin) +
Gab1(ST^0) kr_erk2_gab1
Erk2(kin!1).Gab1(ST^0!1) -> Erk2(kin) +
Gab1(ST^p) kcat_erk2_gab1
```

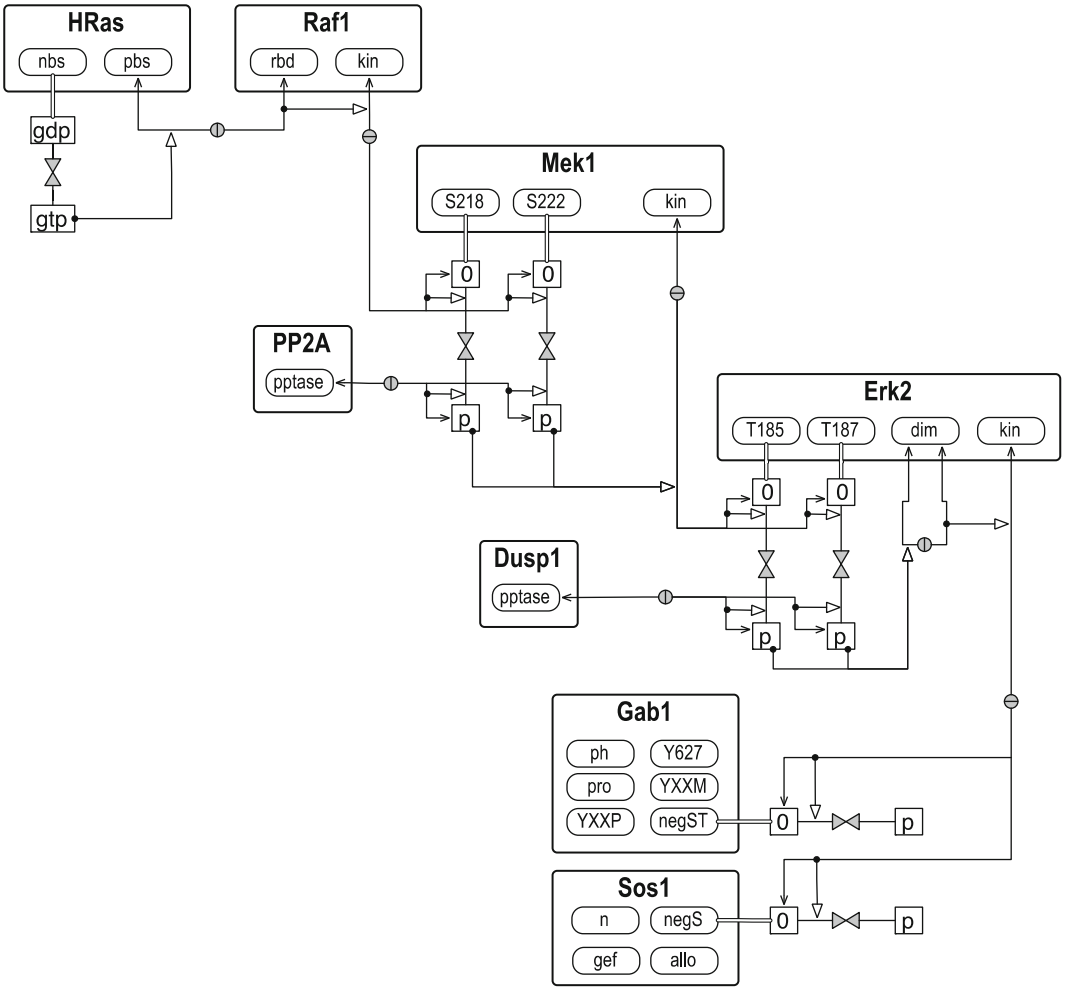

Fig. 14. MAPK cascade. The canonical MAPK cascade is shown for illustration purposes. In reality, the pathway is much more complicated. Here, the example G-protein HRas activates Raf1 by binding, when HRas is in GTP-bound state. The HRas-GTP-bound Raf1 undergoes a conformational change and becomes an active kinase. It binds to and phosphorylates serines on Mek1 and this is opposed by an example phosphatase PP2A. Phosphorylated Mek1 is an active kinase and phosphorylates Erk2 on its threonines, and this is opposed by a dual-specificity phosphatase, Dusp1, shown as an example. Phosphorylated Erk2 can dimerize and this positively influences its kinase activity. Erk2 has several system-wide substrates, two important ones being Gab1 and Sos1 which allow negative feedback control over the transduced signal.

```
Erk2(dim!+,kin) + Sos1(S~0) -> Erk2(dim!+,kin!1).
  Sos1(S~0!1) kf_erk_sos1
Erk2(kin!1).Sos1(S~0!1) -> Erk2(kin) + Sos1(S~0)
  kr_erk_sos1
Erk2(kin!1).Sos1(S~0!1) -> Erk2(kin) + Sos1(S~p)
  kcat_erk2_sos1
```

The modeled rules are illustrated in Fig. 14.

9.4. Gene Regulation

Extracellular signals have their furthest reaching effects when the transduced signals culminate in the nucleus. This is usually accomplished by changing the distribution of transcription factors in the nucleus causing repression or enhancement of mRNA transcript production from multiple genes.

The central dogma of biology, i.e., transcription followed by translation, can be explicitly implemented using rules. Both involve creation of new molecules (mRNA transcripts and proteins).

```
Gene(a) -> Gene(a) + Transcript(a) k_transcription
Transcript(a) -> Transcript(a) + Protein(a)
  k_translation
```

If stochastic fluctuations (due to genes turning on and off) are important to the system, then it is important to restrict the number of molecules of the `Gene` to the actual copy number to replicate real behavior.

A simplifying assumption can be made, where all the transcription, translation, and post-synthetic modification steps can be compressed into a single step. This amounts to assuming that the delay caused by sequence of steps in creating the product is negligible when compared to the rates at which the protein is produced (see ref. 48 supplementary material). The effective reaction would be:

```
Gene(a) -> Gene(a) + Protein(a) k_syn
```

Protein synthesis can also be made dependent on transcription-factor binding.

```
Gene(tf)+ TF(gene) <-> Gene(tf!0).TF(gene!0)
  kf_gene_tf,kr_gene_tf
Gene(tf!+) -> Gene(tf!+) + Protein(a) k_syn
```

Often, the binding can be assumed to be in quasi-equilibrium on the timescale on which genes are expressed so that the expression rate becomes a function of the transcription factor concentration. An example is shown here using a Hill rate law (see Subheading 5.3.2 for details).

```
TF()+Gene() -> TF()+Gene()+Protein() Hill(k,K,n)
```

In the rest of this section, we implement reaction rules pertaining to gene regulation by a cytosolic signal. The rules will be modeled in the compartmental context, retaining the same compartmental hierarchy in Subheading 7.4 and Fig. 5. Specifically, we model the transduction of signal from two parallel MAPK pathways in the form of Erk2 and Jnk1 into the nucleus and activation of the AP1 transcription factor. The information presented here is reviewed in refs. 69–71.

Knowledge Base:

Biphosphorylated Erk2 and Jnk1 can reversibly homodimerize in both cytosolic and nuclear compartments. It is sufficient that one partner is phosphorylated.

```
Erk2(T185~p,Y187~p,dim)+Erk2(dim) ->\
 Erk2(S218~p,S222~p,dim!1)+Erk2(dim!1)\
 kf_erk2_dim
Erk2(dim!1).Erk2(dim!1) -> Erk2(dim) + Erk2(dim)\
 kr_erk2_dim
Jnk1(T183~p,Y185~p,dim)+Jnk1(dim) ->\
 Jnk1(T183~p,Y185~p,dim!1)+Jnk1(dim!1)\
 kf_jnk1_dim
Jnk1(dim!1).Jnk1(dim!1) -> Jnk1(dim) + Jnk1(dim)\
 kr_jnk1_dim
```

Since homo-dimerization occurs in both cytosolic and nuclear compartments, it is sufficient to model them with universal reaction rules.

Knowledge Base:

Both Erk2 and Jnk1 monomers passively diffuse into the nucleus. Erk2 and Jnk1 dimers also translocate to the nucleus at a faster rate.

Since the nucleus and cytosol are bridged by the nuclear membrane, we use bridged-volume molecule transport rules. We also need to allocate separate microscopic rates for monomers and dimers.

```
# Erk2 and Jnk1 monomer transport
Erk2(T185,Y187,dim,kin)@Cyt <-> Erk2(T185,Y187,\
 dim,kin)@Nuc k_tr_erk2_m,k_tr_erk2_m
Jnk1(T183,Y185,dim,kin)@Cyt <-> Jnk1(T183,Y185,\
 dim,kin)@Nuc k_tr_jnk1_m,k_tr_jnk1_m

# Erk2 and Jnk1 dimer transport
Erk2(T185,Y187,dim!1,kin)@Cyt.Erk2(T185,Y187,\
  dim!1,kin)@Cyt<->Erk2(T185,Y187,dim!1,kin)@Nuc.\
Erk2(T185,Y187,dim!1,kin)@Nuc k_tr_erk2_d,\
 k_tr_erk2_d
Jnk1(T183,Y185,dim!1,kin)@Cyt.Jnk1(T183,Y185,\
  dim!1,kin)@Cyt<->Jnk1(T183,Y185,dim!1,kin)@Nuc.\
Jnk1(T183,Y185,dim!1,kin)@Nuc k_tr_jnk1_d,\
 k_tr_jnk1_d
```

Note that the other components on Erk2 and Jnk1 need to be in unbound state to allow transport.

Knowledge Base:

Fos and Jun are transcription factors synthesized and maintained at a certain level in the nucleus.

```
FosGene()@Nuc -> FosGene()@Nuc + Fos(y~0)@Nuc
  k_syn_fos
JunGene()@Nuc -> JunGene()@Nuc + Jun(y~0)@Nuc
  k_syn_jun
Fos()@Nuc -> 0 k_del_fos
Jun()@Nuc -> 0 k_del_jun
```

Instead of simply creating a preexisting number of Fos and Jun molecules, we now have a dynamic equilibrium of synthesis/degradation events that leads to a steady-state number of Fos and Jun.

Knowledge Base:

Activated Erk2 and Jnk1 dimers can phosphorylate Fos and Jun, respectively.

```
Erk2(dim!+,fos) + Fos(y~0)<-> Erk2(dim!+,fos!0).\
  Fos(y~0!0) kf_erk_fos,kr_erk_fos
Erk2(dim!+,fos!0).Fos(y~0!0)-> Erk2(dim!+,fos) +\
   Fos(y~p) k_ph_erk
Jnk1(dim!+,kin) + Jun(y~0)<-> Jnk1(dim!+,kin!0).\
  Jun(y~0!0) kf_jnk_kin,kr_jnk_kin
Jnk1(dim!+,kin!0).Jun(y~0!0)-> Jnk1(dim!+,kin) +\
  Jun (y~p) k_ph_jnk
```

Since there are no direct or indirect transport mechanisms for Fos and Jun, universal rules are sufficient to model these interactions. Activated Fos and Jun can dimerize to form the transcription factor AP1, which shows binding affinity for other transcription factors and certain promoter DNA sequences. The binding sites are shared across both Fos and Jun subunits. Since the Fos–Jun dimer seems to behave as an independent molecule type with binding sites, we create a new molecule type called AP1.

Knowledge Base:

Activated Fos and Jun can hetero-dimerize to form AP1 (Fos–Jun). AP1 can be degraded.

```
Fos(y~p) + Jun(y~p) <-> AP1(g,nfat) kf_kin_kin,
  kr_kin_kin
AP1()@Nuc -> 0 k_del_ap1
```

Knowledge Base:

AP1 can bind to promoter sequences and activate target genes on its own or in combination with other transcription factors (NFAT). AP1-bound genes enable synthesis of target proteins in the cytosol.

```
AP1(nfat) + NFAT(ap1) <-> AP1(nfat!0).NFAT(ap1!0)\
  kf_ap1_nfat,kr_ap1_nfat
AP1(g) + TargetGene(prm) <-> AP1(g!0).TargetGene\
  (prm!0) kf_ap1_gene,kr_ap1_gene
TargetGene(prm!+)@Nuc -> TargetGene(prm!+)@Nuc +\
  TargetProtein@Cyt k_syn_ap1
```

The gene regulation model is illustrated in Fig. 15.

10. Good Modeling Practice

This section is intended as a summary of the critical points from the previous sections, which taken together provide guidelines for the creation of comprehensible, intuitive and biologically realistic models using BNGL.

- Name molecules after important indivisible entities in the model.
- Name components after biological substructures (domains, motifs, amino acid sequence position, etc.), their binding partner, or their designated function.
- The fewer the transformations in a reaction rule, the more realistic it is.
- Unimolecular and bimolecular reaction orders are realistic. Higher orders are less so.
- The rate constant used should be the *asymmetric, per-site rate constant*, which would require scaling by volume and Avogadro number in BNG and only by Avogadro number in cBNG.
- When writing reaction rules, omit components that do not influence the rate of the reaction.
- When two (or more) reactions have the same reaction center but different kinetics, use multiple reaction rules to model them, unless the difference is due to symmetry and multiplicity effects.

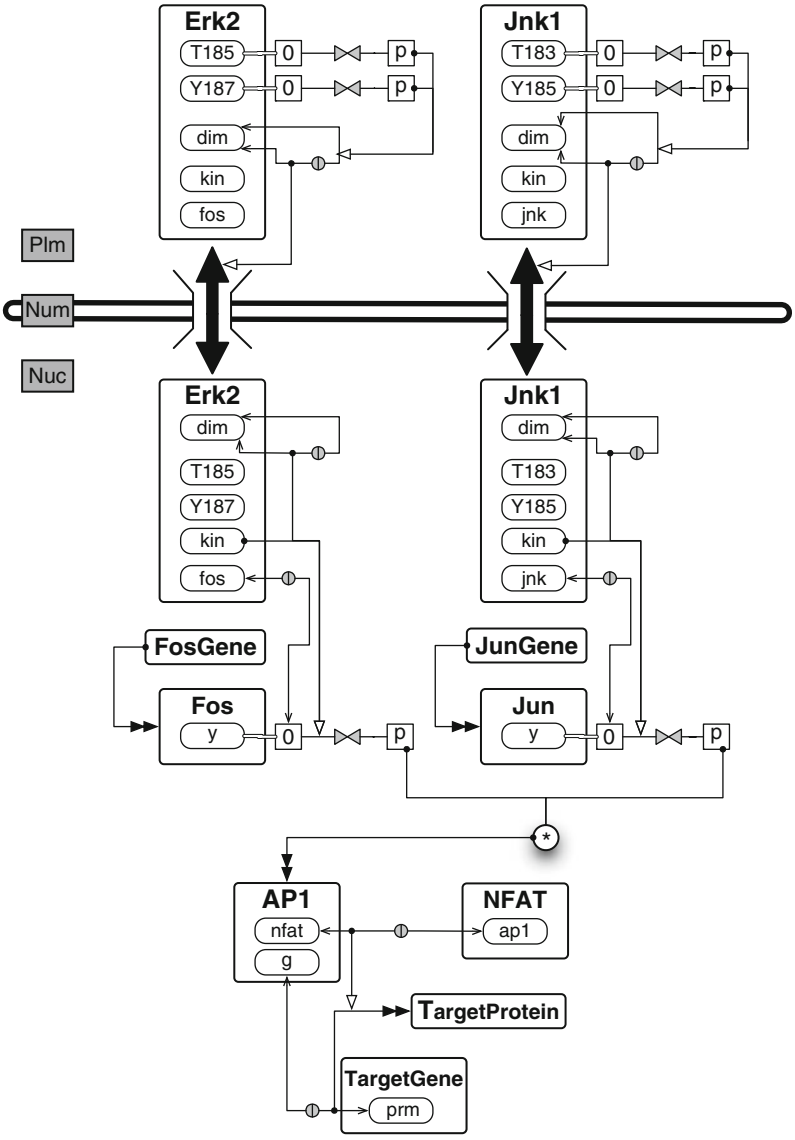

Fig. 15. AP1 activation. Erk2 and Jnk1 are MAP kinases that are activated through many pathways, especially growth factor signals. Both Erk2 and Jnk1 can translocate to the nucleus, with the transport being enhanced by dimerization. In the nucleus, they can activate Fos and Jun transcription factors by phosphorylation. Fos and Jun are actively synthesized (as shown by the *black double arrowhead*) and maintained at equilibrium. Activated Fos and Jun can bind to each other to form the AP1 transcription factor. Since AP1 has many distinct activities on its own, it is modeled as a separate molecule. The *circle* with the *star* indicates that one molecule of Fos and one molecule of Jun are destroyed to synthesize one molecule of AP1. AP1 can bind to other transcription factors, such as NFAT, and the complex can initiate transcription at several target genes. Here, we have also subsumed transcription, translation, and posttranslational modification into a single process.

- Cooperative interactions must obey detailed balance.
- Use wildcards in reaction rules and observables to finely tailor matching conditions in patterns.
- When synthesizing a new species in a reaction rule, provide the full species specification.
- Most transformations are reversible in a realistic model. Unless there is specific information on irreversibility, there should exist rules that:
  - Dephosphorylate all phosphorylation sites.
  - Break all bonds that can be formed.
  - Degrade all molecules that can be synthesized.
- Deletion rules delete the whole species by default. Use `DeleteMolecules` keyword to delete only the respective molecules and not the whole species.
- When transporting molecules in cBNG, check to see if you are transporting molecules or species.
- Avoid rate law approximations. Prefer explicit mechanisms.
- Use structural knowledge to infer how interactions affect one another. In the absence of information assume they are independent (i.e. distributive or parallel).
- Model interactions as sequential (i.e. processive) only if there is reasonable structural information that such is the case.
- If small concentration changes can cause drastic effects, use a stochastic simulator.
- If the network is too big, use a network-free simulator.
- To simulate a typical biological experiment, always equilibrate before perturbing.
- Avoid having to generate the same network repeatedly, especially with large networks.

Acknowledgments

We gratefully acknowledge support from NIH grant GM76570, NSF grant CCF-0829788, and NSF Expeditions in Computing Grant 0926181. We would also like to thank Arshi Arora, Leonard A. Harris, William S. Hlavacek, and Justin S. Hogg for their helpful comments on the manuscript.

References

1. Kholodenko BN, Hancock JF, Kolch W (2010) Signalling ballet in space and time. Nat Rev Mol Cell Biol 11:414–426
2. Kholodenko BN (2006) Cell-signalling dynamics in time and space. Nat Rev Mol Cell Biol 7:165–176
3. Scott JD, Pawson T (2009) Cell signaling in space and time: where proteins come together and when they're apart. Science (New York, NY) 326:1220–1224
4. Natarajan M, Lin K-M, Hsueh RC, Sternweis PC, Ranganathan R (2006) A global analysis of cross-talk in a mammalian cellular signalling network. Nat Cell Biol 8:571–580
5. Hecker M, Lambeck S, Toepfer S, van Someren E, Guthke R (2009) Gene regulatory network inference: data integration in dynamic models – a review. Biosystems 96:86–103
6. Hlavacek WS, Faeder JR (2009) The complexity of cell signaling and the need for a new mechanics. Sci Signal 2:46
7. Hlavacek W, Faeder J, Blinov M (2006) Rules for modeling signal-transduction systems. Sci STKE 344:6
8. Bachman JA, Sorger P (2011) New approaches to modeling complex biochemistry. Nat Methods 8:130–131
9. Faeder J, Blinov M, Hlavacek W (2009) Rule-based modeling of biochemical systems with BioNetGen. Methods Mol Biol 500:113–167
10. Blinov ML, Faeder JR, Goldstein B, Hlavacek WS (2004) BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. Bioinformatics 20:3289–3291
11. Danos V, Laneve C (2004) Formal molecular biology. Theor Comput Sci 325:69–110
12. Feret J, Danos V, Krivine J, Harmer R, Fontana W (2009) Internal coarse-graining of molecular systems. Proc Natl Acad Sci USA 106:6453–6458
13. Haugh JM, Lauffenburger DA (1997) Physical modulation of intracellular signaling processes by locational regulation. Biophys J 72:2014–2031
14. Michaelis L, Menten M (1913) Die kinetik der invertinwirkung. Biochem Z 49:333–369
15. Briggs G, Haldane J (1925) A note on the kinetics of enzyme action. Biochem J 19:338
16. Cornish-Bowden A (2004) Fundamentals of enzyme kinetics, 3rd edn. Portland Press, London
17. Chen WW, Niepel M, Sorger PK (2010) Classic and contemporary approaches to modeling biochemical reactions. Gene Dev 24:1861–1875
18. Tracy TS, Hummel MA (2004) Modeling kinetic data from in vitro drug metabolism enzyme experiments. Drug Metab Rev 36:231–242
19. Sabouri-Ghomi M, Ciliberto A, Kar S, Novak B, Tyson JJ (2008) Antagonism and bistability in protein interaction networks. J Theor Biol 250:209–218
20. Sneddon MW, Faeder JR, Emonet T (2011) Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. Nat Methods 8:177–183
21. Chen WW, Schoeberl B, Jasper PJ, Niepel M, Nielsen UB, Lauffenburger DA, Sorger PK (2009) Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. Mol Syst Biol 5:239
22. Barik D, Baumann WT, Paul MR, Novak B, Tyson JJ (2010) A model of yeast cell-cycle regulation based on multisite phosphorylation. Mol Syst Biol 6:405
23. Hänggi P (2002) Stochastic resonance in biology how noise can enhance detection of weak signals and help improve biological information processing. Chemphyschem 3:285–290
24. Gillespie D (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 22:403–434
25. Yang J, Monine MI, Faeder JR, Hlavacek WS (2008) Kinetic Monte Carlo method for rule-based modeling of biochemical networks. Phys Rev E 78:031910
26. Borisov N, Chistopolsky A, Faeder J, Kholodenko B (2008) Domain-oriented reduction of rule-based network models. IET Syst Biol 2:342–351
27. Conzelmann H, Saez-Rodriguez J, Sauter T, Kholodenko BN, Gilles ED (2006) A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. BMC Bioinformatics 7:34
28. Lok L, Brent R (2005) Automatic generation of cellular reaction networks with Moleculizer 1.0. Nat Biotechnol 23:131–136
29. Faeder JR, Blinov M, Goldstein B, Hlavacek W (2005) Rule-based modeling of biochemical networks. Complexity 10:22–41
30. Danos V, Feret J, Fontana W, Krivine J (2007) Scalable simulation of cellular signaling networks. Lect Notes Comput Sci 4807:139–157

31. Le Novére N, Shimizu TS (2001) STOCHSIM: modelling of stochastic biomolecular processes. Bioinformatics 17:575–576
32. Colvin J, Monine MI, Faeder JR, Hlavacek WS, Hoff DDV, Posner RG (2009) Simulation of large-scale rule-based models. Bioinformatics 25:910–917
33. Colvin J, Monine MI, Gutenkunst RN, Hlavacek WS, Hoff DDV, Posner RG (2010) RuleMonkey: software for stochastic simulation of rule-based models. BMC Bioinformatics 11:404
34. Walker F, Orchard SG, Jorissen RN, Hall NE, Zhang H-H, Hoyne PA, Adams TE, Johns TG, Ward C, Garrett TPJ, Zhu H-J, Nerrie M, Scott AM, Nice EC, Burgess AW (2004) CR1/CR2 interactions modulate the functions of the cell surface epidermal growth factor receptor. J Biol Chem 279:22387–22398
35. Jorissen RN, Walker F, Pouliot N, Garrett TPJ, Ward CW, Burgess AW (2003) Epidermal growth factor receptor: mechanisms of activation and signalling. Exp Cell Res 284:31–53
36. Zhang X, Gureasko J, Shen K, Cole PA, Kuriyan J (2006) An allosteric mechanism for activation of the kinase domain of epidermal growth factor receptor. Cell 125:1137–1149
37. Tao R, Maruyama I (2008) All EGF (ErbB) receptors have preformed homo-and heterodimeric structures in living cells. J Cell Sci 121:3207–3217
38. Kholodenko BN, Demin OV, Moehren G, Hoek JB (1999) Quantification of short term signaling by the epidermal growth factor receptor. J Biol Chem 274:30169–30181
39. Schoeberl B, Eichler-Jonsson C, Gilles ED, Müller G (2002) Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. Nat Biotechnol 20:370–375
40. Wofsy C, Goldstein B, Lund K, Wiley HS (1992) Implications of epidermal growth factor (EGF) induced egf receptor aggregation. Biophys J 63:98–110
41. Ollivier JF, Shahrezaei V, Swain PS (2010) Scalable rule-based modelling of allosteric proteins and biochemical networks. PLoS Comput Biol 6:e1000975
42. Macdonald JL, Pike LJ (2008) Heterogeneity in EGF-binding affinities arises from negative cooperativity in an aggregating system. Proc Natl Acad Sci USA 105:112–117
43. Alvarado D, Klein DE, Lemmon MA (2010) Structural basis for negative cooperativity in growth factor binding to an EGF receptor. Cell 142:568–579
44. Harris LA, Hogg JS, Faeder JR (2009) Compartmental rule-based modeling of biochemical systems. Proceedings of the 2009 winter simulation conference, 13–16 Dec 2009, Austin TX, USA
45. Saltelli A, Ratto M, Tarantola S, Campolongo F (2005) Sensitivity analysis for chemical models. Chem Rev 105:2811–2827
46. Seewöster T, Lehmann J (1997) Cell size distribution as a parameter for the predetermination of exponential growth during repeated batch cultivation of CHO cells. Biotechnol Bioeng 55:793–797
47. Swanson JA, Lee M, Knapp PE (1991) Cellular dimensions affecting the nucleocytoplasmic volume ratio. J Cell Biol 115:941–948
48. Lipniacki T, Puszynski K, Paszek P, Brasier AR, Kimmel M (2007) Single TNFalpha trimers mediating NF-kappaB activation: stochastic robustness of NF-kappaB signaling. BMC Bioinformatics 8:376
49. Ghaemmaghami S, Huh W-K, Bower K, Howson RW, Belle A, Dephoure N, O'Shea EK, Weissman JS (2003) Global analysis of protein expression in yeast. Nature 425:737–741
50. Lee B, LeDuc PR, Schwartz R (2008) Stochastic off-lattice modeling of molecular self-assembly in crowded environments by Green's function reaction dynamics. Phys Rev E Stat 78:031911
51. Kim JS, Yethiraj A (2010) Crowding effects on association reactions at membranes. Biophys J 98:951–958
52. Gabdoulline RR, Wade RC (2002) Biomolecular diffusional association. Curr Opin Struct Biol 12:204–213
53. Alberts B, Bray D, Lewis J, Raff M, Roberts K, Watson JD (1994) Molecular biology of the cell, 3rd edn. Garland Publishing, New York
54. Hlavacek WS, Faeder JR, Blinov ML, Perelson AS, Goldstein B (2003) The complexity of complexes in signal transduction. Biotechnol Bioeng 84:783–794
55. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucleic Acids Res 28:235–242
56. Consortium U (2010) The universal protein resource (UniProt) in 2010. Nucleic Acids Res 38:D142–D148
57. Schulze WX, Deng L, Mann M (2005) Phosphotyrosine interactome of the ErbB-receptor kinase family. Mol Syst Biol 1:2005.0008
58. Gu H, Neel BG (2003) The "Gab" in signal transduction. Trends Cell Biol 13:122–130

59. Ravichandran KS (2001) Signaling via Shc family adapter proteins. Oncogene 20:6322–6330
60. Neel B, Gu H, Pao L (2003) The 'Shp'ing news: SH2 domain-containing tyrosine phosphatases in cell signaling. Trends Biochem Sci 28:284–293
61. Das J, Ho M, Zikherman J, Govern C, Yang M, Weiss A, Chakraborty AK, Roose JP (2009) Digital signaling and hysteresis characterize ras activation in lymphoid cells. Cell 136:337–351
62. Langlois WJ, Sasaoka T, Saltiel AR, Olefsky JM (1995) Negative feedback regulation and desensitization of insulin- and epidermal growth factor-stimulated p21ras activation. J Biol Chem 270:25320–25323
63. Seminario M-C, Precht P, Bunnell SC, Warren SE, Morris CM, Taub D, Wange RL (2004) PTEN permits acute increases in D3-phosphoinositide levels following TCR stimulation but inhibits distal signaling events by reducing the basal activity of Akt. Eur J Immunol 34:3165–3175
64. Terai K, Matsuda M (2005) Ras binding opens c-Raf to expose the docking site for mitogen-activated protein kinase kinase. EMBO Rep 6:251–255
65. Huang W, Kessler DS, Erikson RL (1995) Biochemical and biological analysis of Mek1 phosphorylation site mutants. Mol Biol Cell 6:237–245
66. Grethe S, Pцrn-Ares MI (2006) p38 MAPK regulates phosphorylation of Bad via PP2A-dependent suppression of the MEK1/2-ERK1/2 survival pathway in TNF-alpha induced endothelial apoptosis. Cell Signal 18:531–540
67. Kamioka Y, Yasuda S, Fujita Y, Aoki K, Matsuda M (2010) Multiple decisive phosphorylation sites for the negative feedback regulation of SOS1 via ERK. J Biol Chem 285:33540–33548
68. Caunt CJ, Rivers CA, Conway-Campbell BL, Norman MR, McArdle CA (2008) Epidermal growth factor receptor and protein kinase C signaling to ERK2: spatiotemporal regulation of ERK2 by dual specificity phosphatases. J Biol Chem 283:6241–6252
69. Cobb MH, Goldsmith EJ (2000) Dimerization in MAP-kinase signaling. Trends Biochem Sci 25:7–9
70. Chang L, Karin M (2001) Mammalian MAP kinase signalling cascades. Nature 410:37–40
71. Shaulian E, Karin M (2002) AP-1 as a regulator of cell life and death. Nat Cell Biol 4: E131–E136
72. Weiss J (1997) The Hill equation revisited: uses and misuses. FASEB J 11:835–841
73. Sauro HM, Kholodenko BN (2004) Quantitative analysis of signaling networks. Prog Biophys Mol Biol 86:5–43