

Rule-Based Modeling with BIOINETGEN

Jim Faeder

Jose Juan Tapia

Bobby Sheehan

Department of Computational and Systems Biology

University of Pittsburgh School of Medicine

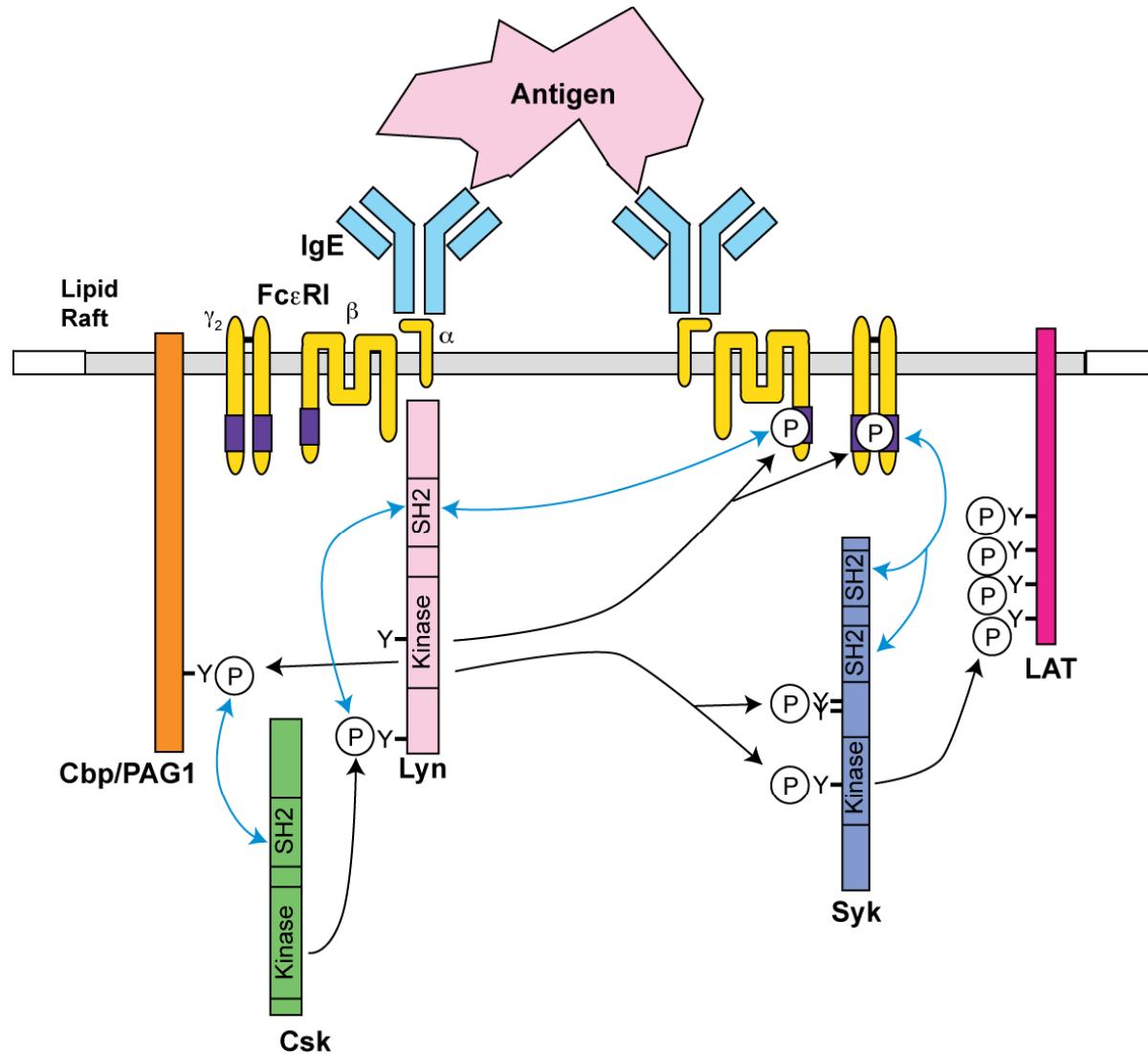
Outline

- Introduction to rule-based modeling
- Essentials
 - Specifying a model
 - Simulating a model
- Hands on tutorial (optional)

For additional information and references see
<http://bionetgen.org/index.php/q-bio-14>

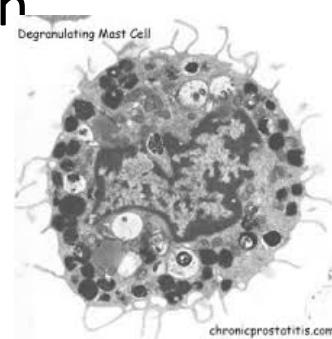
INTRODUCTION

Early events in Fc ϵ RI signaling

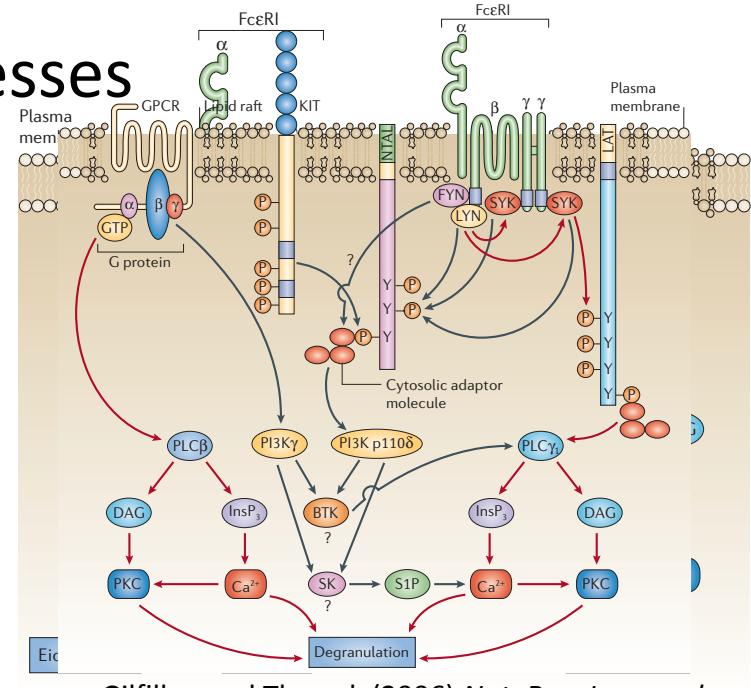


Challenges for Modeling

- Large number of components and interactions
- Rapidly evolving list of important components and interactions
 - structural uncertainty in the model
- Involvement of multiple processes
 - signaling
 - gene regulation
 - protein expression
 - (cell division)



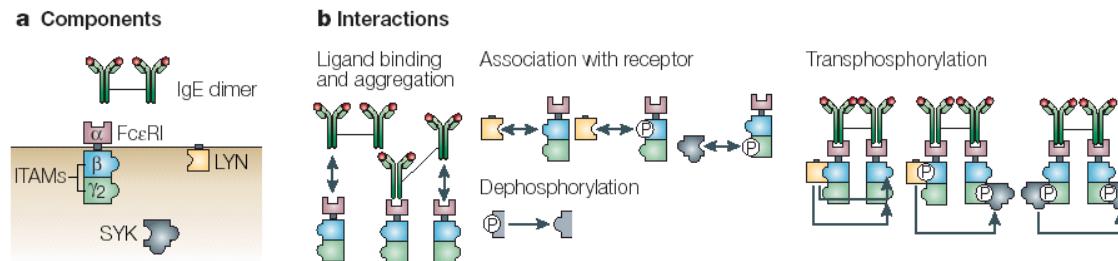
chronicprostatitis.com



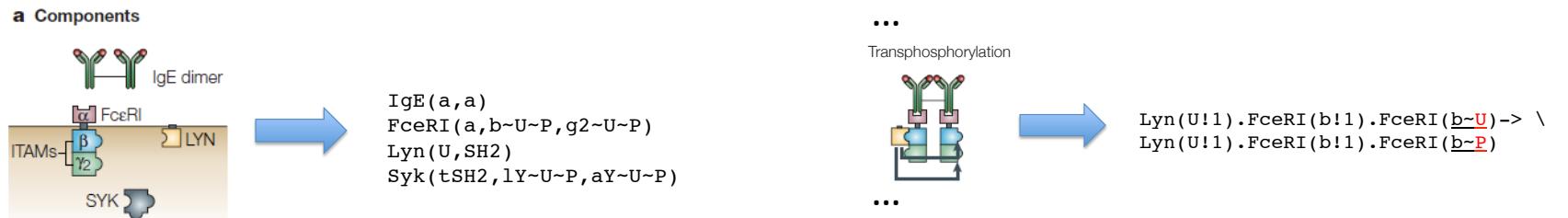
Gilfillan and Tkaczyk (2006) *Nat. Rev. Immunol.*

Rule-Based Modeling protocol

1. Identify components and interactions



2. Translate into objects (molecules) and rules



3. Determine concentrations and rate constants

4. Simulate and analyze the model

ODE's

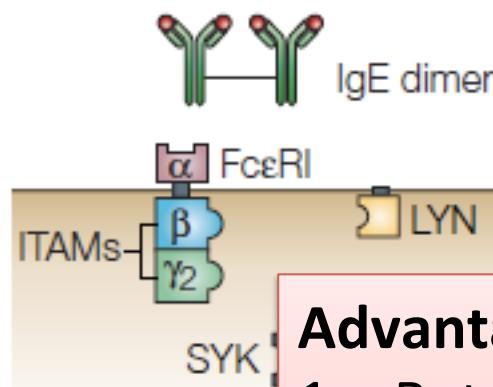
SSA

PDE's

BD

Rule-Based Modeling with BioNetGen

a Components



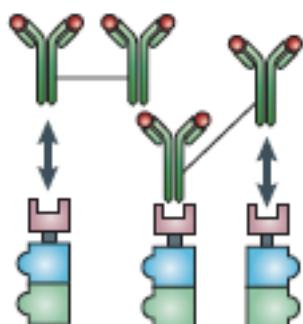
Molecules are *structured* objects

$\text{IgE}(a,a)$
 $\text{Fc}\epsilon\text{RI}(a,b\sim U\sim P,g2\sim U\sim P)$
 $\text{Lyn}(U,\text{SH2})$

Advantages

1. Retain detailed information about molecular interactions.
2. Scalable both in model construction and simulation.

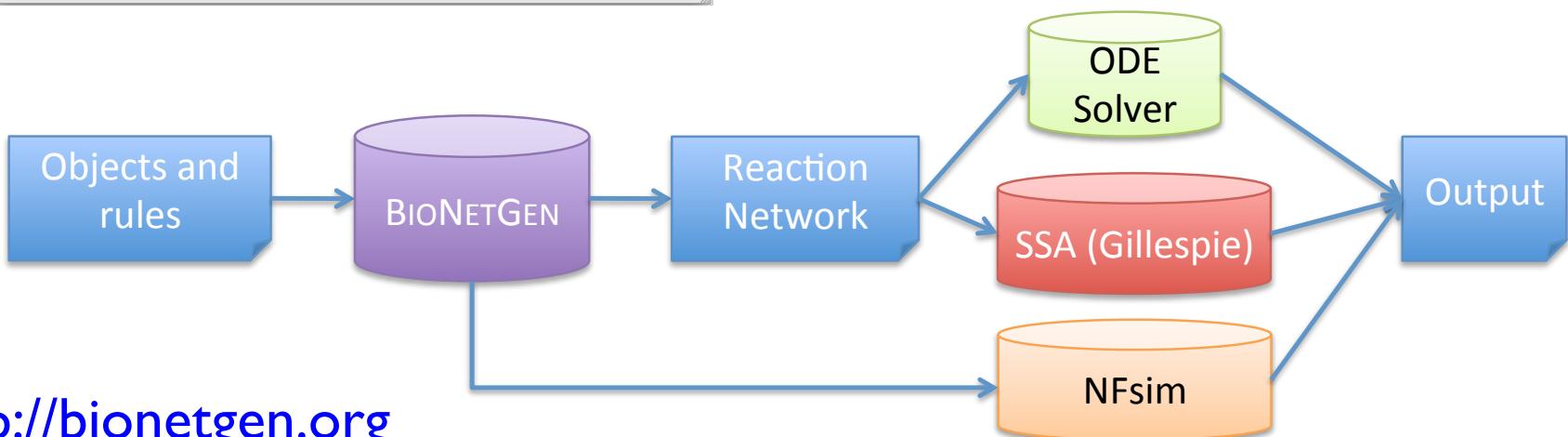
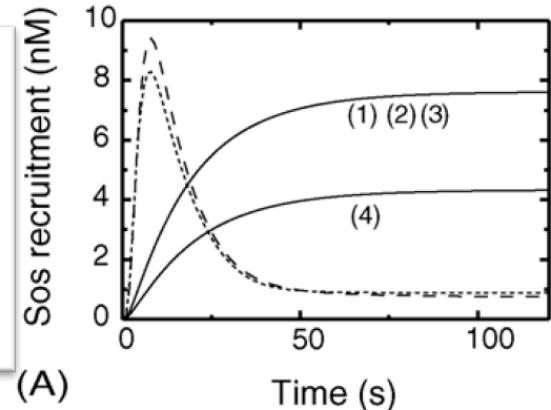
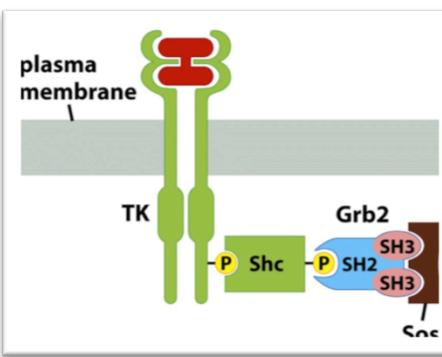
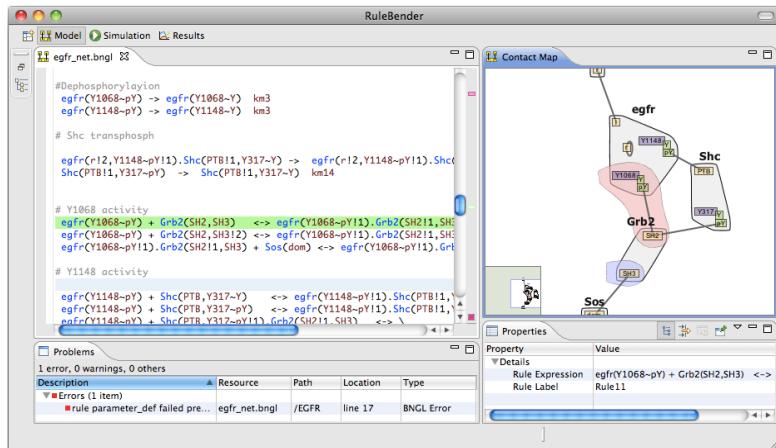
Ligand binding
and aggregation



(a!1)

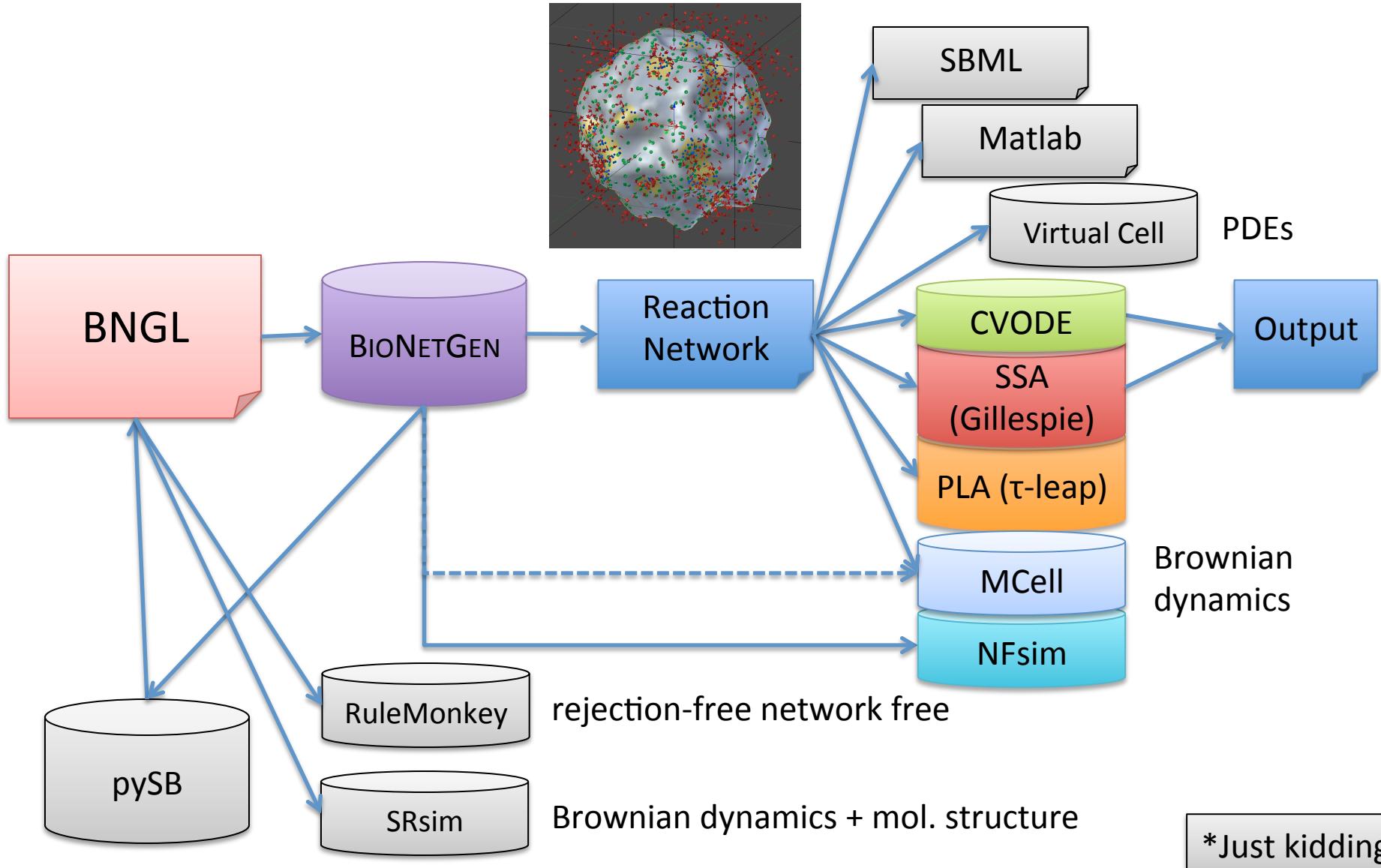
Because rules act *locally* they may generate many possible reactions

Basic RBM workflow with BioNetGen



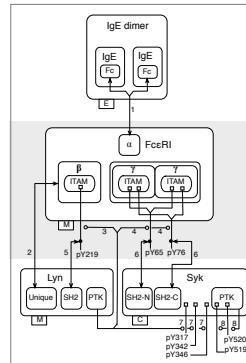
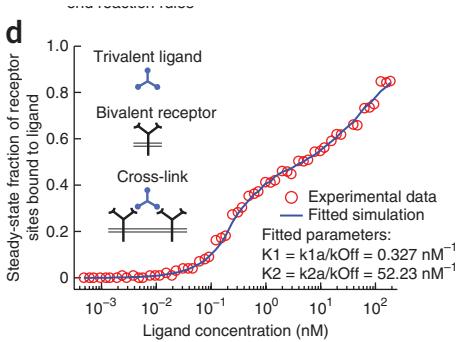
<http://bionetgen.org>
<http://rulebender.org>
<http://nfsim.org>

One language to rule them all*

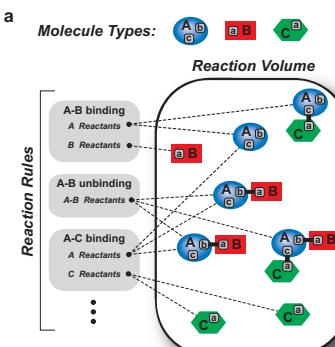


Overall Modeling Workflow

Parameter estimation
to fit experimental data



Detailed signaling
biochemistry can be
directly encoded as a *rule-based model (RBM)*.



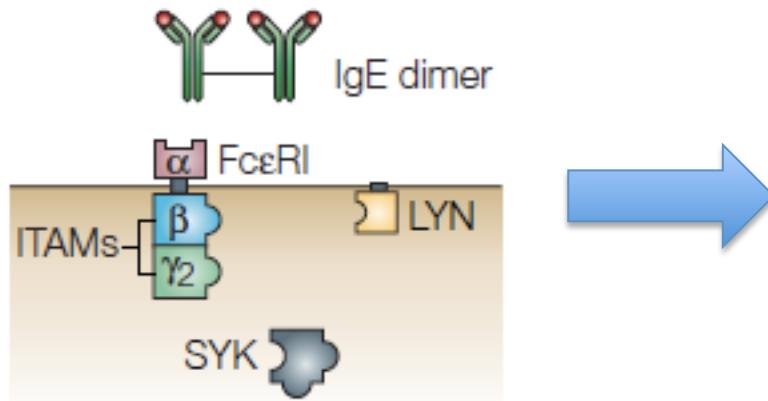
Efficient simulation of
RBM's using direct or
indirect methods

SPECIFYING A RULE-BASED MODEL

Defining Molecules

Molecules are the basic objects in a BNG model

BioNETGEN Language



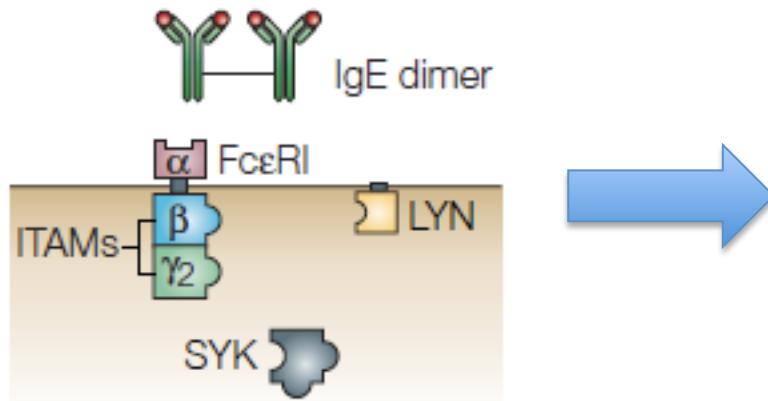
IgE(*a,a*)
Fc ϵ RI(*a,b~U~P,g2~U~P*)
Lyn(*U,SH2*)
Syk(*tSH2,1Y~U~P,aY~U~P*)

- Components** represent molecule elements
- Domains
 - Motifs
 - Properties

Defining Molecules

Molecules are the basic objects in a BNG model

BioNETGEN Language



IgE(a, a)
Fc ϵ RI($a, b-U-P, g2-U-P$)
Lyn($U, SH2$)
Syk($tSH2, lY-U-P, aY-U-P$)

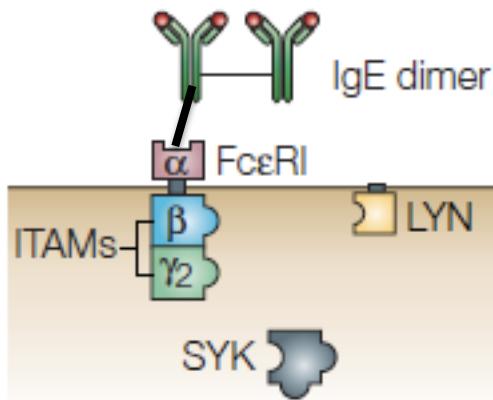
Components may have different **states** representing

- posttranslational modifications
- conformational state
- ...

Binding

Molecules bind other molecules through components

BioNETGEN Language



$\text{IgE}(a, a!1).\text{Fc}\epsilon\text{RI}(a!1, b\sim U, g2\sim U)$

Bonds are formed by linking two components. The ‘.’ indicates a set of molecules forming a complex.

$\text{Fc}\epsilon\text{RI}(a, b\sim U!1, g2\sim U).\text{Lyn}(U!1)$

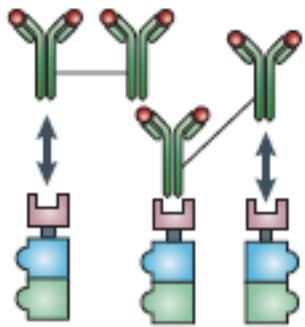
Components may have both states and bonds.

$\text{Lyn}(\text{SH2}!1, \text{Cterm}\sim P!1)$

Bonds may occur within a molecule.

Defining Interaction Rules

Ligand binding
and aggregation



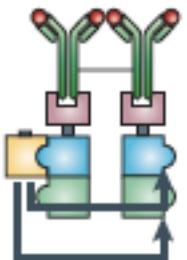
BioNetGEN Language

$\text{IgE}(a, \underline{a}) + \text{Fc}\epsilon\text{RI}(\underline{a}) \rightleftharpoons \text{IgE}(a, \underline{a!1}).\text{Fc}\epsilon\text{RI}(\underline{a!1})$

...

binding and dissociation

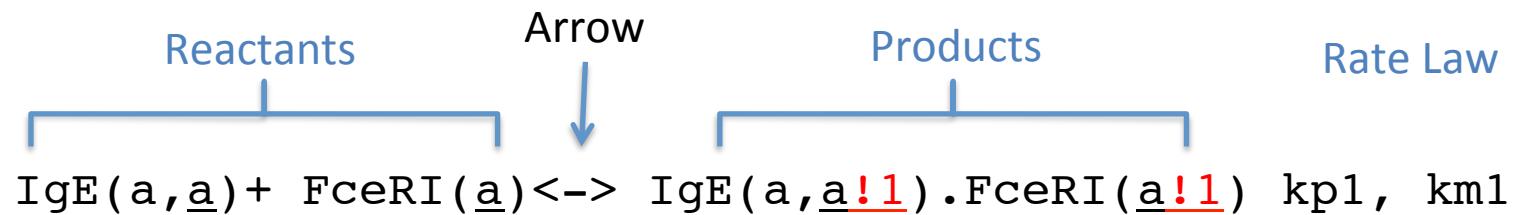
Transphosphorylation



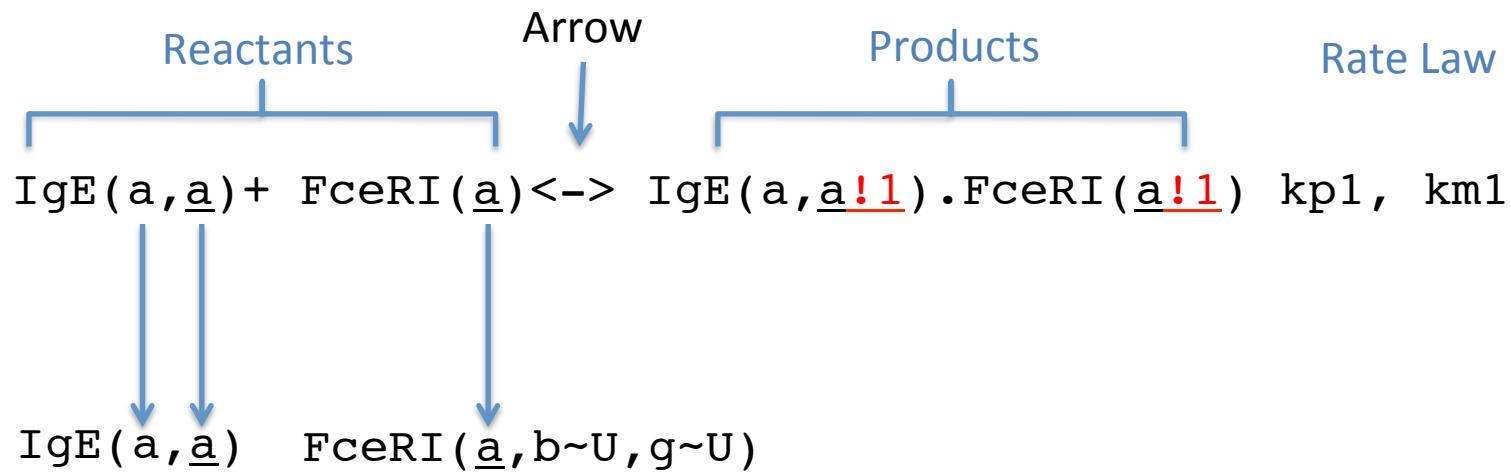
$\text{Lyn}(U!1).\text{Fc}\epsilon\text{RI}(b!1).\text{Fc}\epsilon\text{RI}(b\sim U) \rightarrow \\ \text{Lyn}(U!1).\text{Fc}\epsilon\text{RI}(b!1).\text{Fc}\epsilon\text{RI}(b\sim P)$

component state change

Parts of a rule

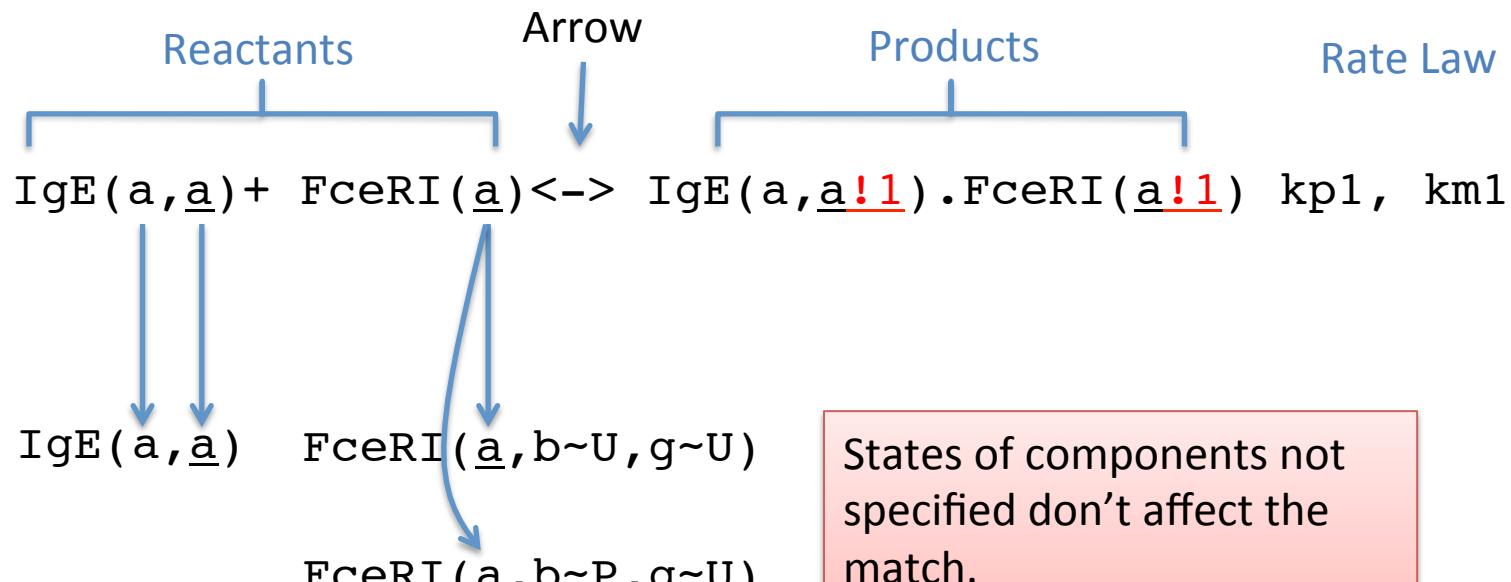


Parts of a rule



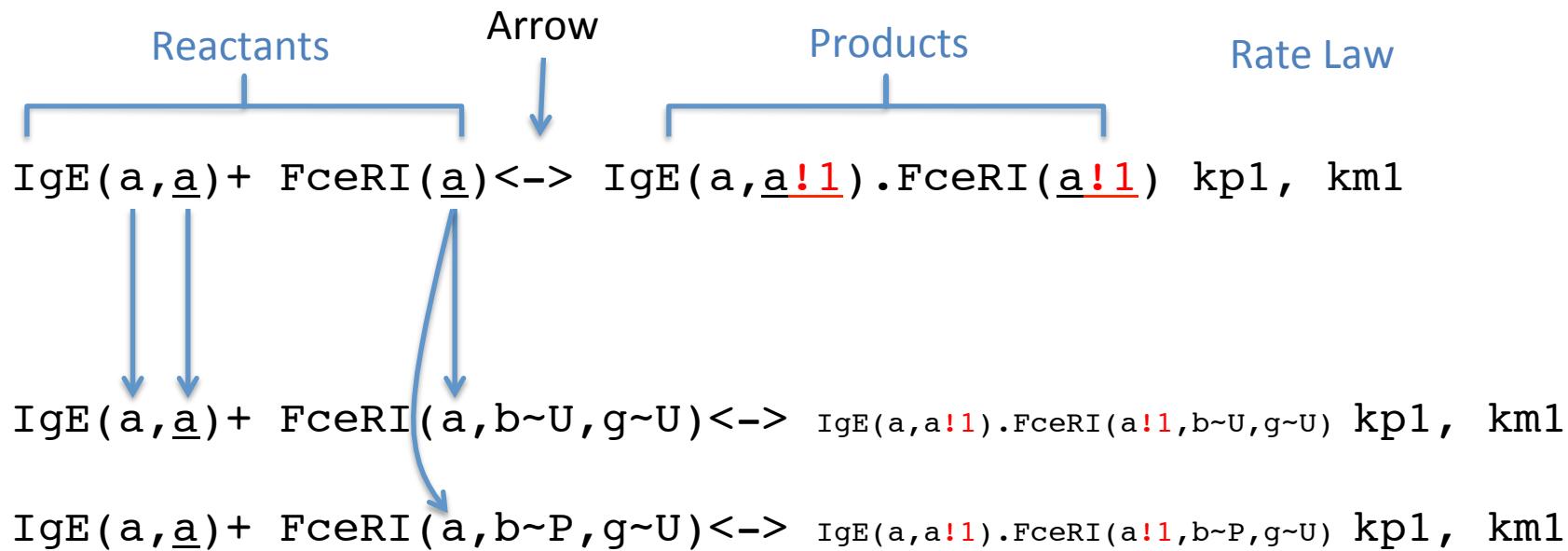
Reactant patterns
select properties of
each reactant
molecule.

Parts of a rule



Reactant patterns
select properties of each reactant molecule.

Parts of a rule

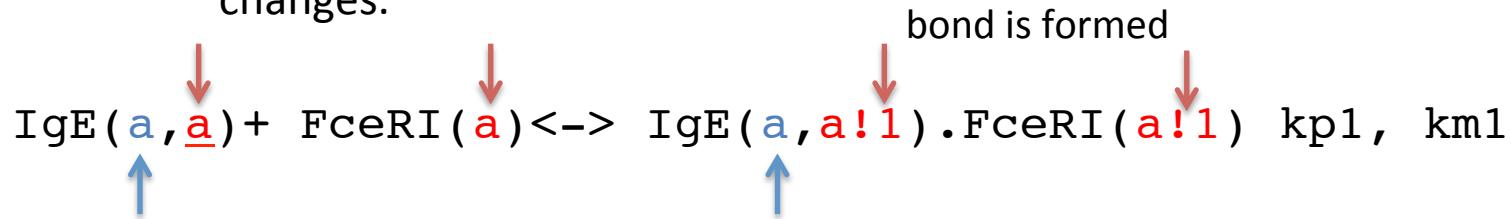


Reactant patterns
select properties of
each reactant
molecule.

Because patterns can match
many different species,
each rule can generate
many reactions.

Center and context

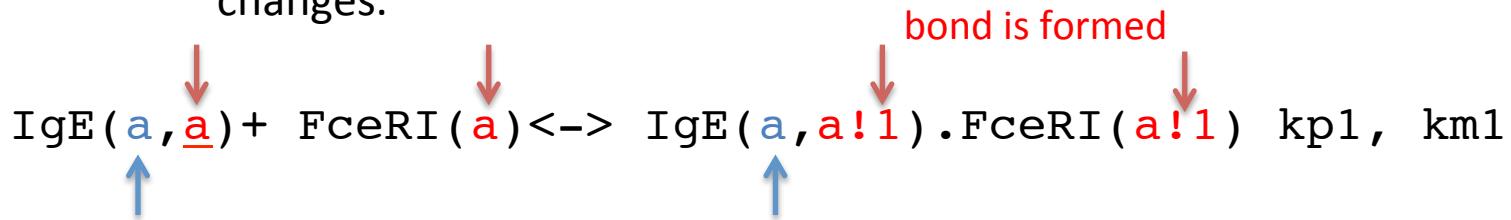
The **center** of a rule is the part that the rule changes.



The **context** is the part that is necessary for the rule to happen but is unchanged.

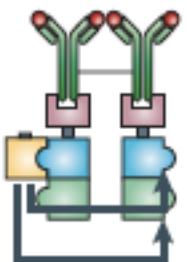
Center and context

The **center** of a rule is the part that the rule changes.



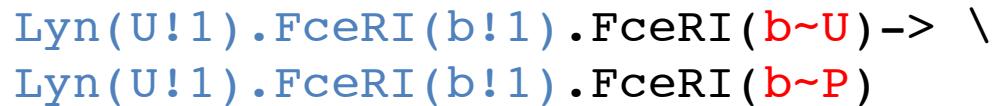
The **context** is the part that is necessary for the rule to happen but is unchanged.

Transphosphorylation



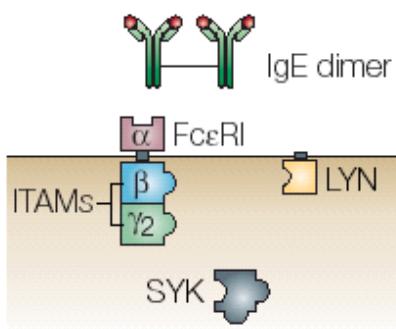
Context can represent complex
biochemistry.

component state is
changed

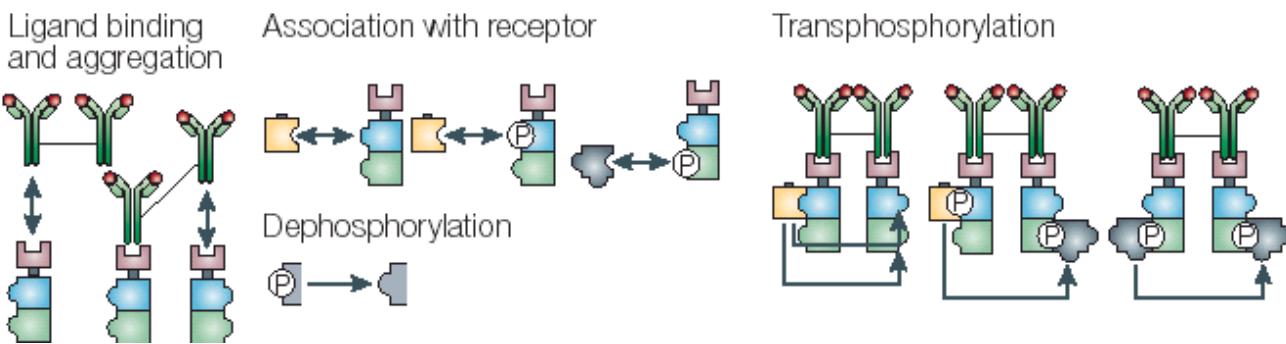


Composition of a Rule-Based Model

a Components



b Interactions



Molecules

```
begin molecules
Lig(1,1)
Lyn(U,SH2)
Syk(tSH2,1~U~P,a~U~P)
Rec(a,b~U~P,g~U~P)
end molecules
```

Reaction Rules

```
begin reaction_rules
# Ligand-receptor binding
1 Rec(a) + Lig(1,1) <-> Rec(a!1).Lig(1!1,1) kp1, km1
    Rec(a) + Lig(1,1) <-> Rec(a!1).Lig(1!1,1) kp1, km1

# Receptor-aggregation
2 Rec(a) + Lig(1,1!1) <-> Rec(a!2).Lig(1!2,1!1) kp2, km2

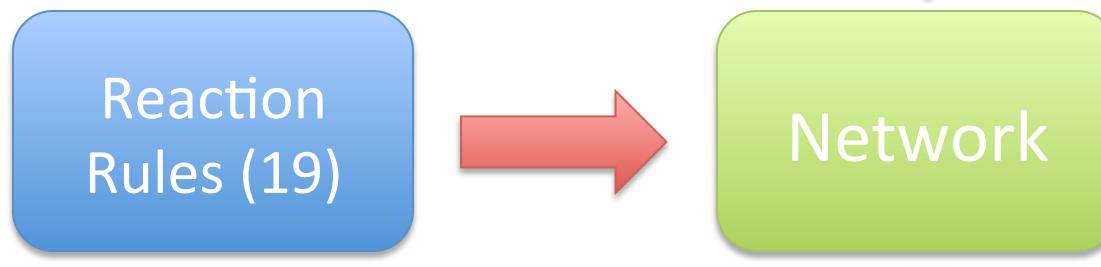
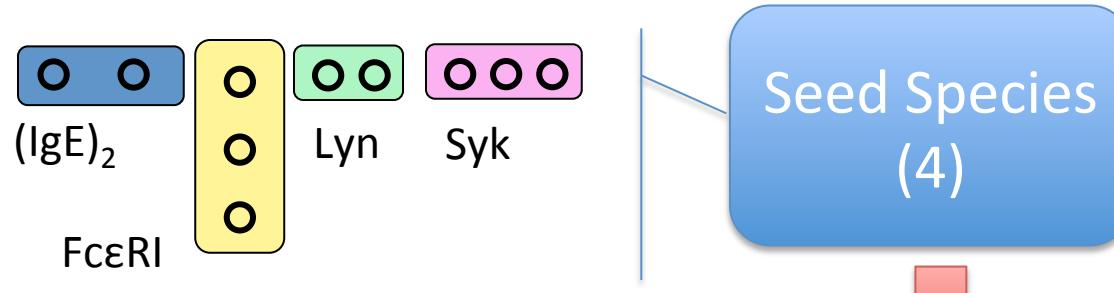
# Constitutive Lyn-receptor binding
3 Rec(b~Y) + Lyn(U,SH2) <-> Rec(b~Y!1).Lyn(U!1,SH2) kpL, kmL
...
```

BioNetGen language

SIMULATING A RULE-BASED MODEL

Automatic Network Generation

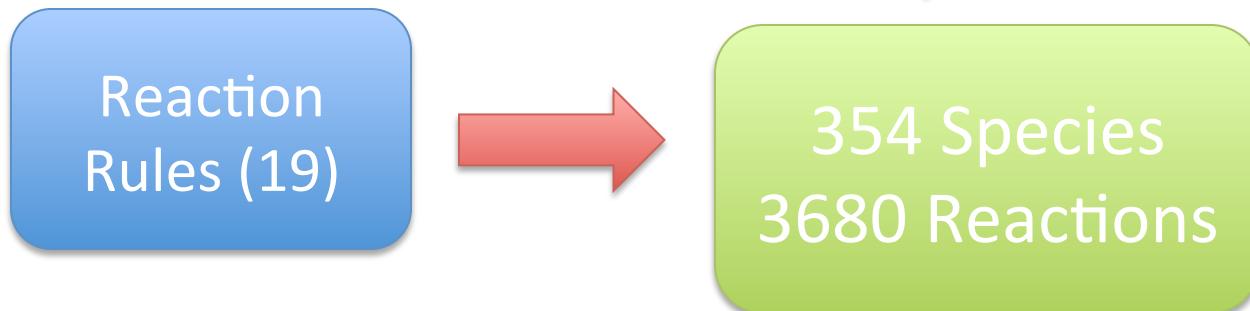
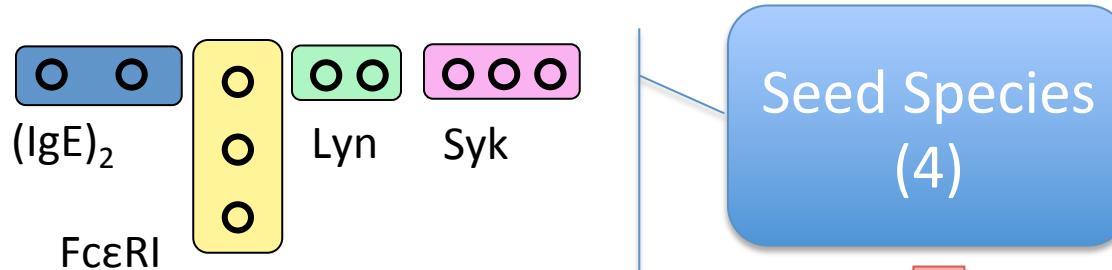
Fc ϵ RI Model



New
Reactions &
Species

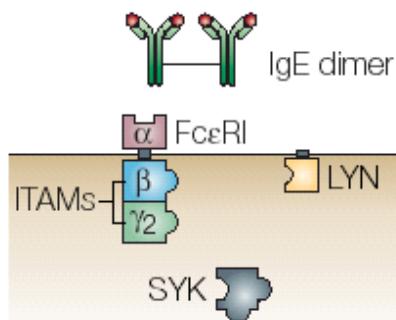
Automatic Network Generation

Fc ϵ RI Model



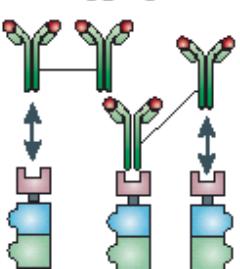
Combinatorial complexity

a Components

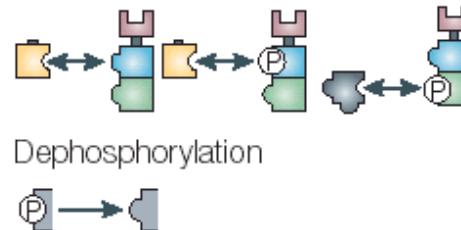


b Interactions

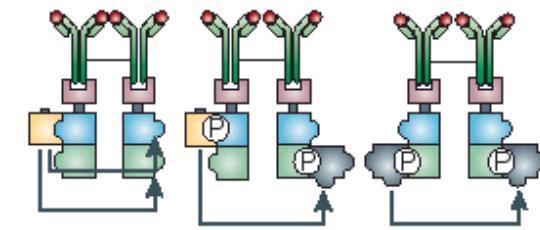
Ligand binding and aggregation



Association with receptor

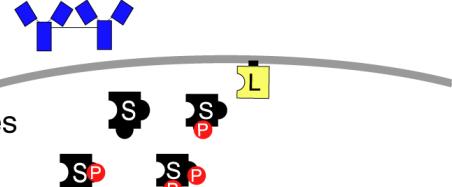


Transphosphorylation



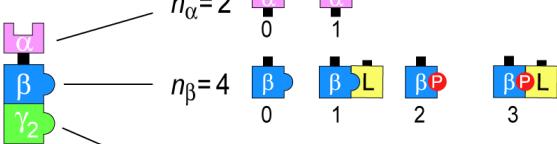
States of the Model

6 free nonreceptor states



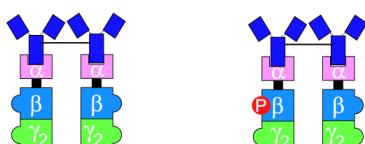
48 monomer states:

$$n_\alpha n_\beta n_\gamma$$



300 dimer states:

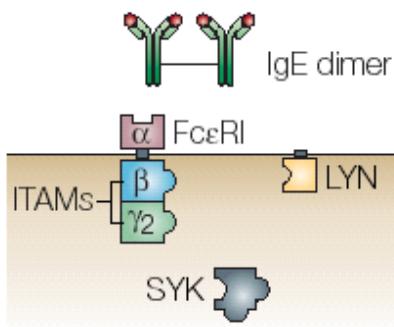
$$n_\beta n_\gamma (n_\beta n_\gamma + 1)/2$$



The model has 354 states (2954 if the ligand was a trimer)

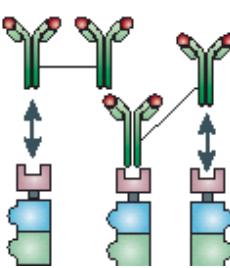
Addressing combinatorial complexity

a Components

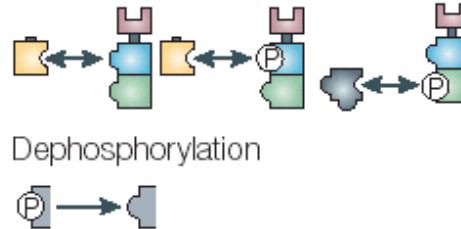


b Interactions

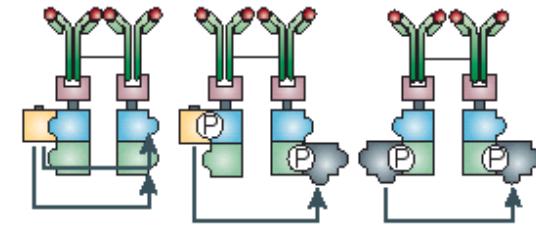
Ligand binding
and aggregation



Association with receptor



Transphosphorylation

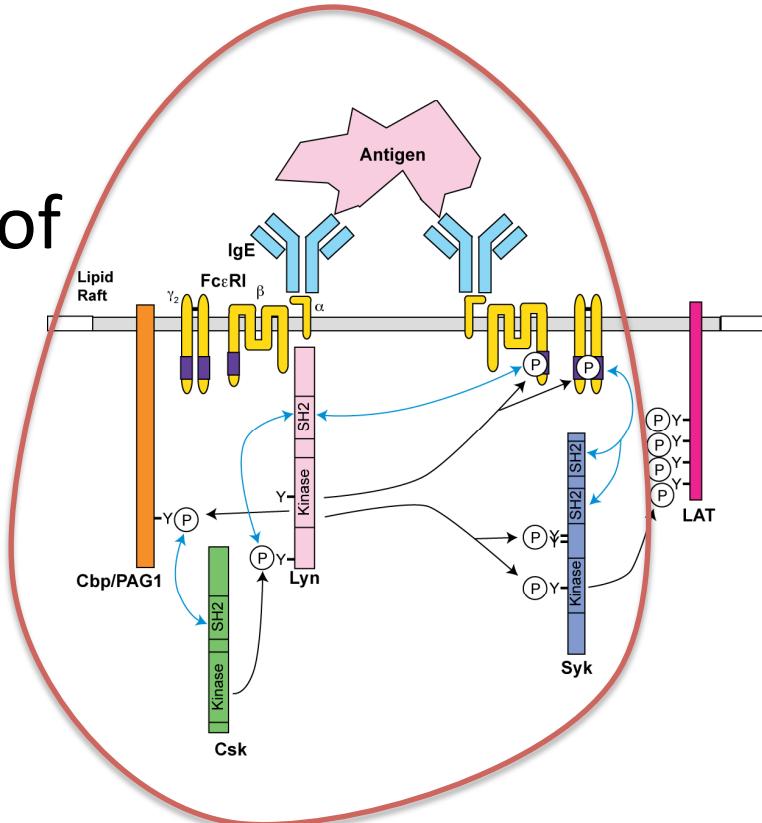


354 species / 3680 reactions

- Standard approach – writing equations by hand – won't work!
- New approach
 - Write model by describing interactions.
 - Automatically generate the equations.

Limits of Network Generation

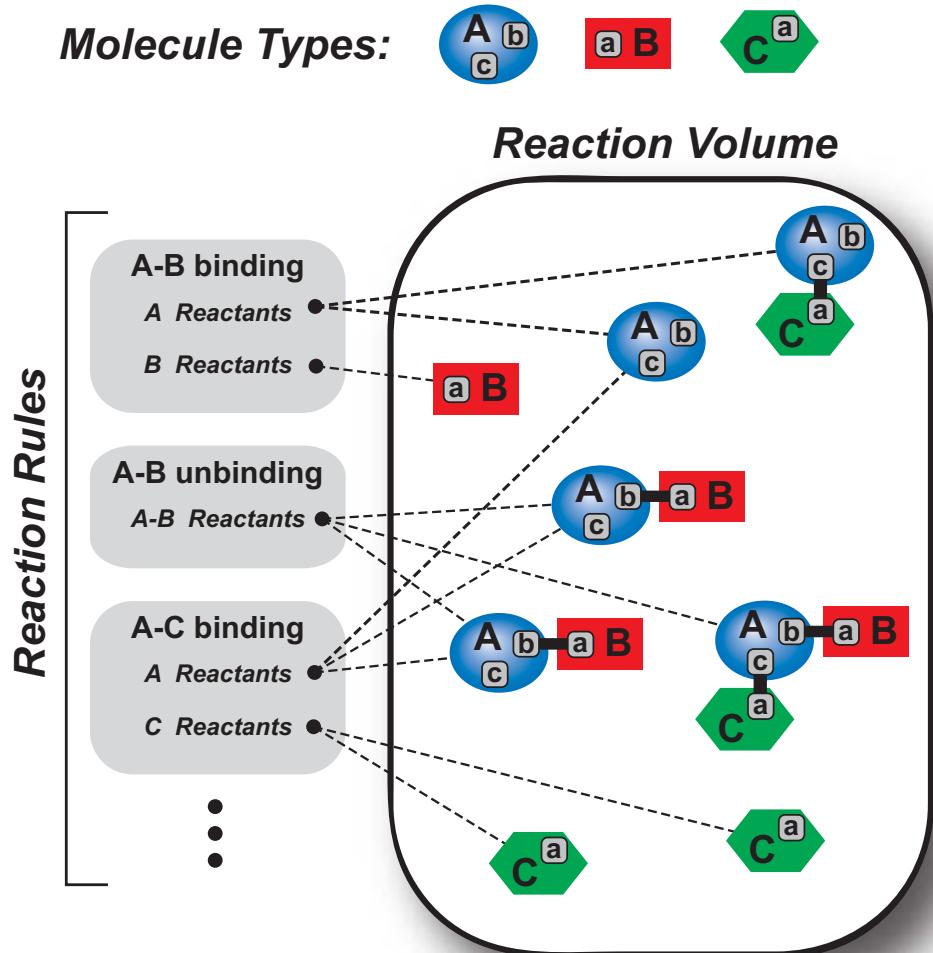
- Combinatorial complexity leads to exponential growth of reaction network size
 - Models with 10^4 - 10^5 reactions can easily arise.
- For such large models, ODE simulation may become inefficient.
- Simulations may have impractical CPU and/or memory requirements.



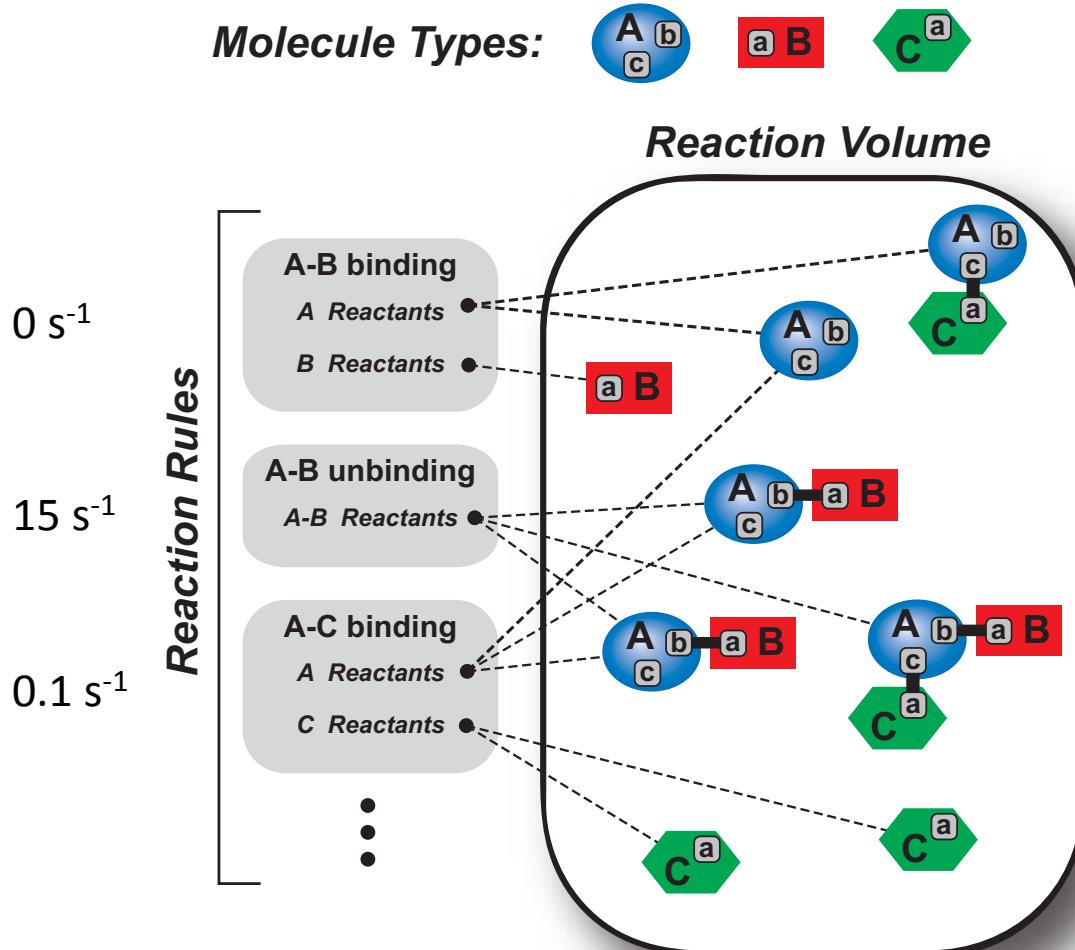
NFSIM*

Network-Free Stochastic Simulator

- Generalization of rule-based kMC method of Yang et al.
- Uses Gillespie (direct) algorithm to sample over *reaction rules*.
- Like BKL ‘*n*-fold method’:
 - sites are instantiated
 - rule-based
 - transformations may affect reactivity of other sites in a complex.

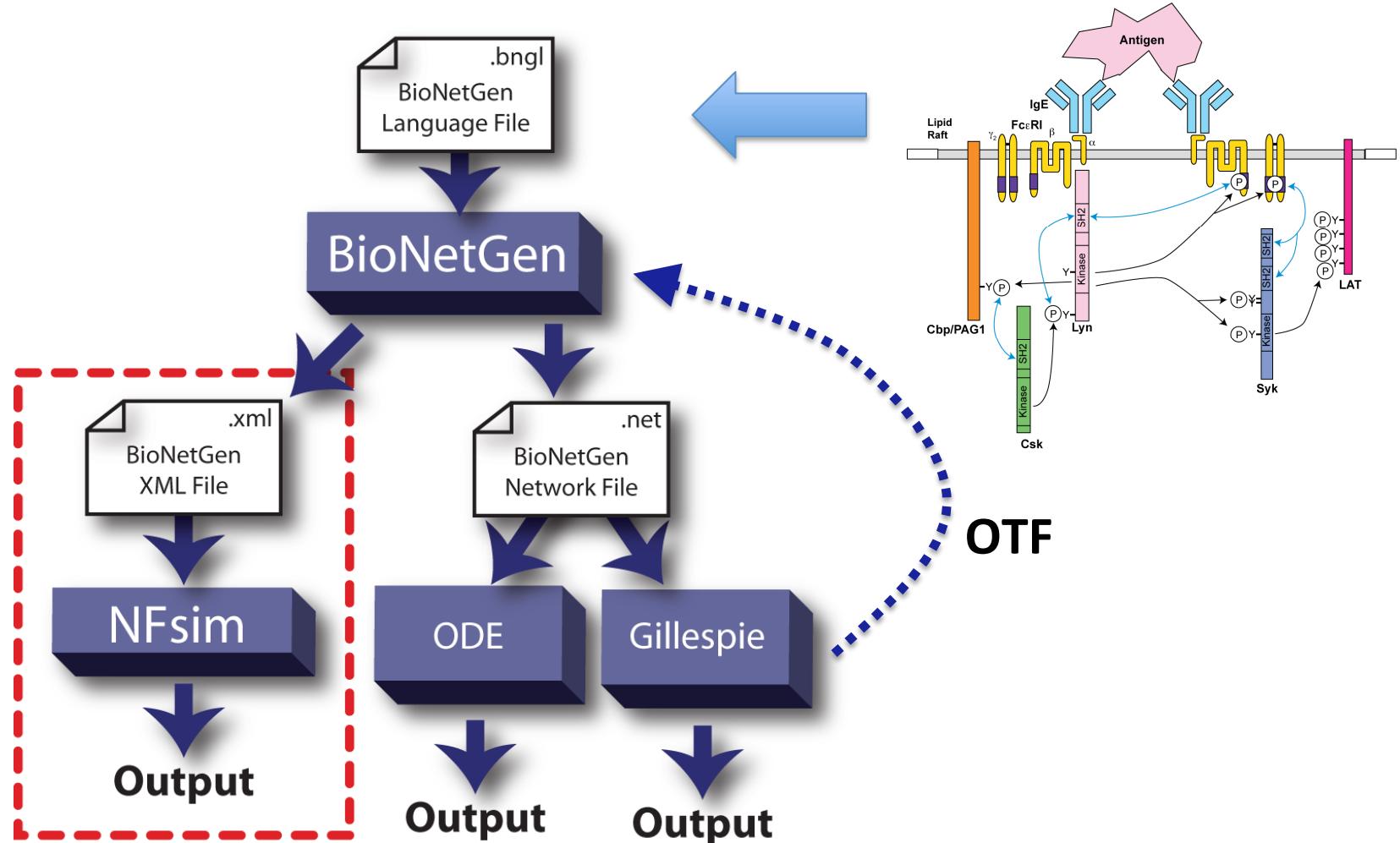


Network Free (NF) Algorithm



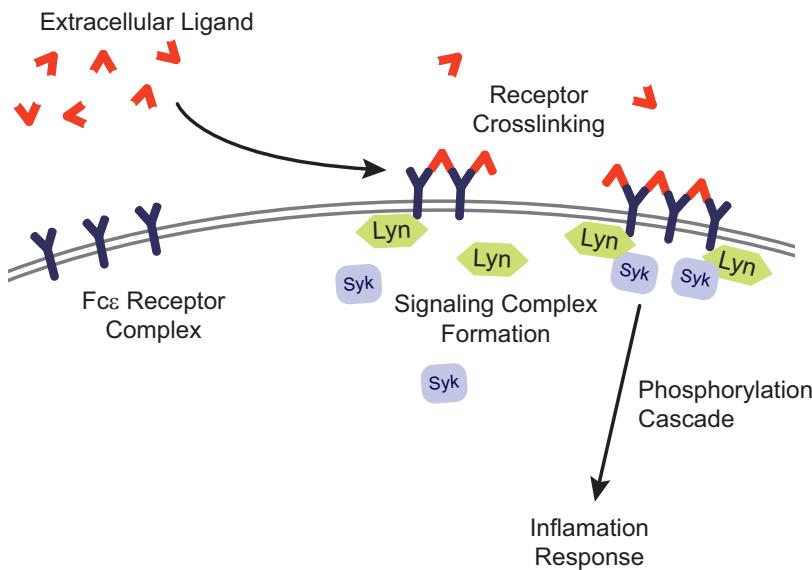
0. Initialize *reactant lists* and calculate *rule propensities*.
1. Select next reaction time and next *rule*.
2. Select molecules and sites to react.
 - a. Check any application condition(s).
3. Apply operation specified by rule.
4. Update reactant lists and propensities.
5. **Increment time.**

Integration with BioNetGEN

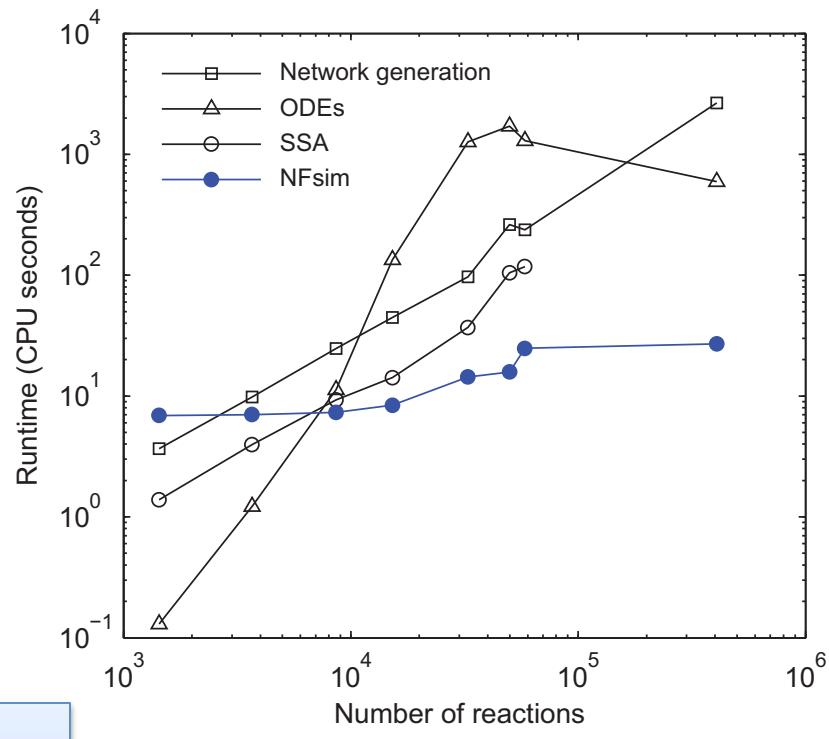


Fc ϵ RI signaling models

a



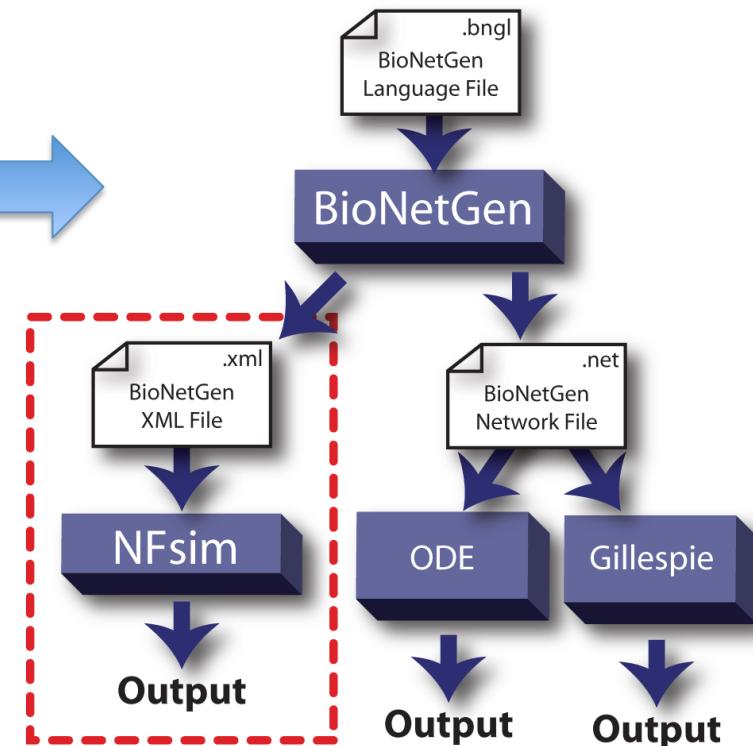
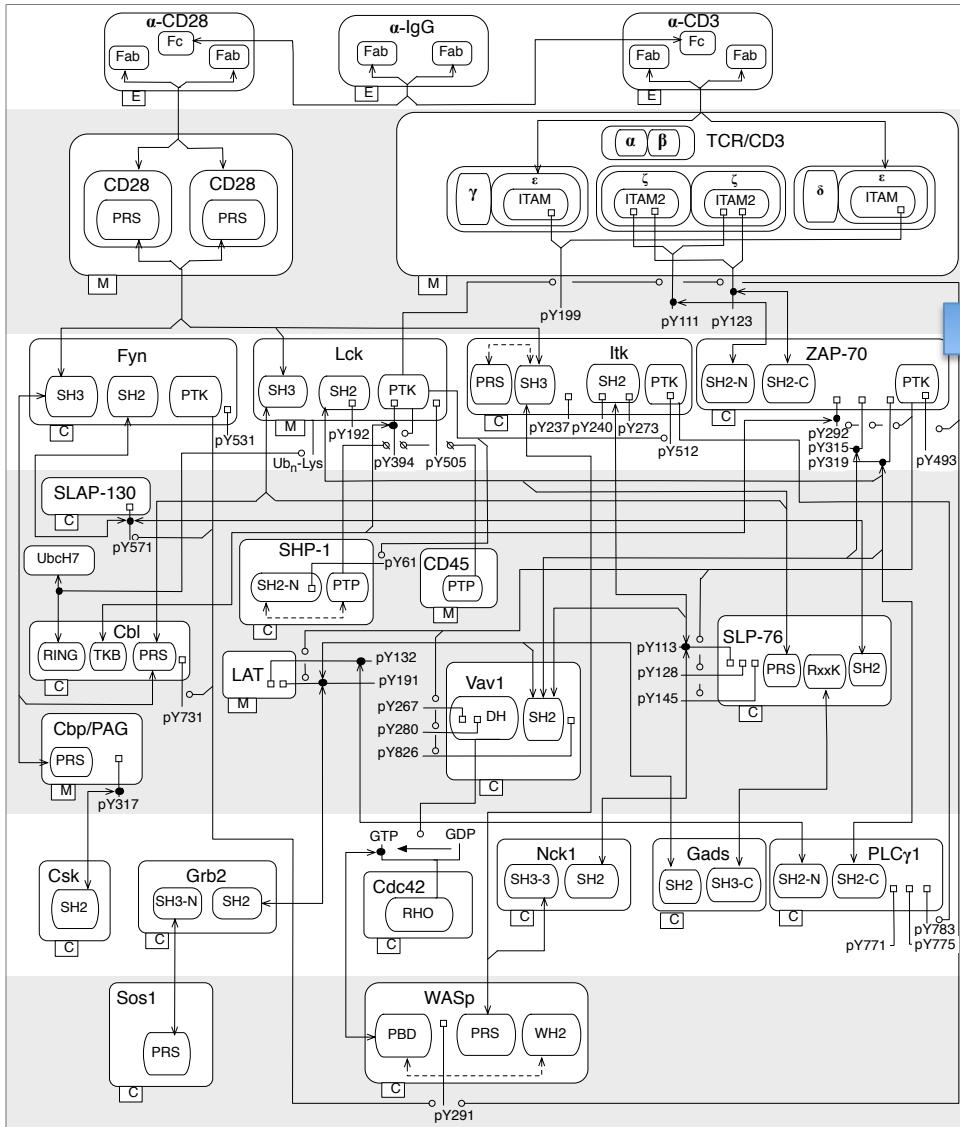
b



NFsim can simulate models of greatly increased complexity with manageable increase in cost.

Increasing complexity →

Large Scale TCR Signaling Model



Large-scale RBM Studies

- Interactome
 - Deeds et al. (2010?)
- Yeast Pheromone
 - Thomson et al.
 - Tiger et al.
 - Deeds lab
- Bacterial chemotaxis
 - Emonet lab
- EGFR
 - Creamer et al. (2012)
- TCR
 - Chylek et al. (2014)

Some Rule-Based Modeling Tools

Well-mixed / Compartmental Systems

- BioNetGen / RuleBender / NFsim
- Kappa

Spatially-resolved Systems

- MCell
- Simmune
- Smoldyn / Moleculizer
- BioNetGen@Virtual Cell
- Stochastic Simulation Compiler
- SRsim
- ML-rules / ML-space
- Meredys

Modeling Frameworks

- pySB (see tutorial by Carlos Lopez)
- rxncon
- LBS (Language for Biological Systems)

Recent BNG-related development

See <http://bionetgen.org> for more details.

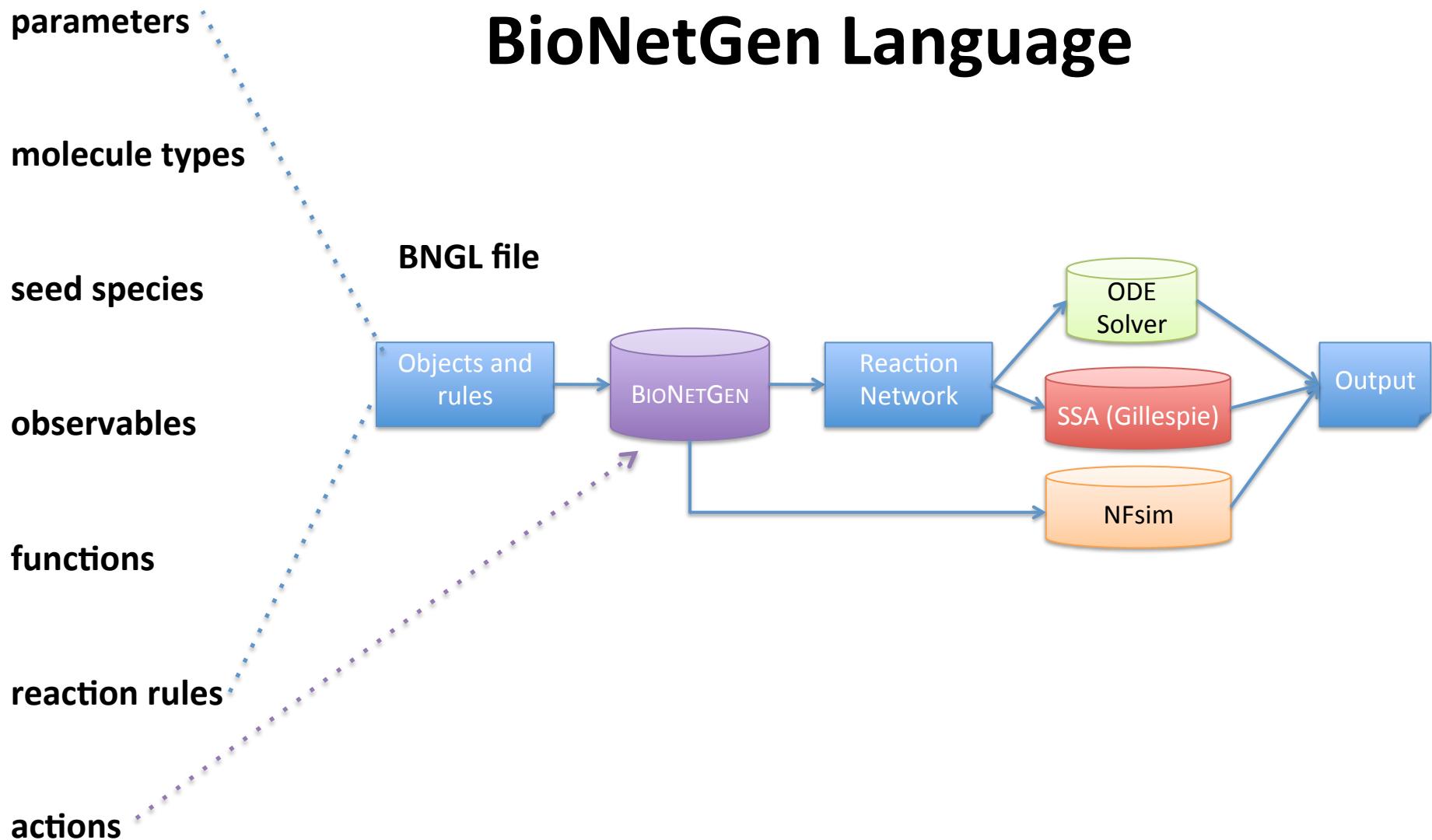
- Modeling resources
 - BioNetGen wiki
 - Mast cell modeling library ([Chylek et al. 2014](#))
 - [RuleHub](#) ([Tapia and Faeder, q-bio 2014](#))
- Compartmental modeling (core)
 - Different spatial domains are treated as well-mixed compartments ([Harris et al. 2009](#))
- Model export (core)
 - Matlab: M-file / Mex-file
 - [SBML](#) (including compartments)
 - [SSC](#) (spatial stochastic simulator)
 - [MCell](#) (native MDL format)
 - [Smoldyn](#) (externally developed)
 - SBML Multi – under development
- Import
 - [SBML/BioModels](#)
 - [Atomizer](#) - extract molecules / components from any SBML file ([Tapia and Faeder 2013](#))
- Hybrid simulation
 - Mix populations and agents (core / NFsim) ([Hogg et al. 2014](#))
- Accelerated stochastic simulations
 - Partitioned leaping algorithm (core) ([Harris and Clancy 2006](#))
- Rare event sampling
 - Weighted Ensemble ([Donovan et al. 2014](#))
- Visualization
 - Extended contact maps ([Chylek et al. 2011](#))
 - Contact and process maps (core BNG)
- Energy-based modeling (eBNGL)
 - Compact model description – interactions and cooperativity ([Hogg 2013](#))
- Parameter estimation
 - Ptempest (<http://code.google.com/p/ptempest>)
- Boolean modeling
 - 3 different Boolean update schemes (<http://code.google.com/p/bionetgen/boolean>)

References

- L. A. Chylek, D. A. Holowka, B. A. Baird, and W. S. Hlavacek (2014) “An interaction library for the Fc ϵ RI signaling network,” *Frontiers Immunol.*, 42pp.
- L. A. Chylek, B. Hu, M. L. Blinov, T. Emonet, J. R. Faeder, B. Goldstein, R. N. Gutenkunst, J. M. Haugh, T. Lipniacki, R. G. Posner, J. Yang, and W. S. Hlavacek (2011) “Guidelines for visualizing and annotating rule-based models.,” *Mol. Biosyst.*, **7**, 2779–2795.
- L. A. Chylek, L. A. Harris, C.-S. Tung, J. R. Faeder, C. F. Lopez, and W. S. Hlavacek (2013) “Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems.,” *Wiley Interdiscip. Rev. Syst. Biol. Med.*, **6**, 13–36.
- R. M. Donovan, A. J. Sedgewick, J. R. Faeder, and D. M. Zuckerman (2013) “Efficient Stochastic Simulation of Chemical Kinetics Networks using a Weighted Ensemble of Trajectories.” *J. Chem. Phys.*, **139**, 115105.
- Harris LA, Clancy P (2006) “A ‘partitioned leaping’ approach for multiscale modeling of chemical reaction dynamics.” *J. Chem. Phys.*, **125**, 144107.
- L. A. Harris, J. S. Hogg, and J. R. Faeder (2009) “Compartmental rule-based modeling of biochemical systems,” in Proceedings of the 2009 Winter Simulation Conference (WSC09), pp. 908–919.
- J. S. Hogg (2013) *Advances in Rule-Based Modeling: Compartments, Energy, and Hybrid Simulation, with Application to Sepsis and Cell Signaling*, PhD Thesis, University of Pittsburgh School of Medicine.
- J. S. Hogg, L. A Harris, L. J. Stover, N. S. Nair, and J. R. Faeder (2014) “Exact hybrid particle/population simulation of rule-based models of biochemical systems.,” *PLOS Comput. Biol.*, **10**, e1003544.
- J. J. Tapia and J. R. Faeder (2013) “The Atomizer: Extracting Implicit Molecular Structure from Reaction Network Models,” in *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics (BCB’13)*, pp. 726–727.

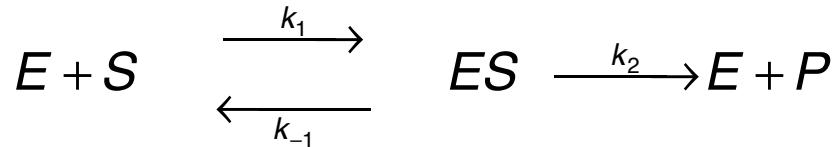
HANDS-ON TUTORIAL

Technical Overview of the BioNetGen Language



Example 1: MM Mechanism

parameters



molecule types

seed species

A BioNetGen model consists of a set of blocks, each beginning and ending with begin <blockname> / end <blockname> respectively.

observables

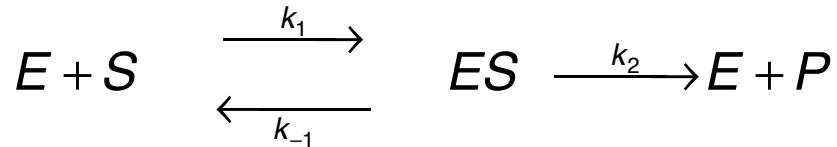
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

parameters – model constants are defined here. *The user is responsible for using a consistent set of units, which should be indicated in the associated comments.*

observables

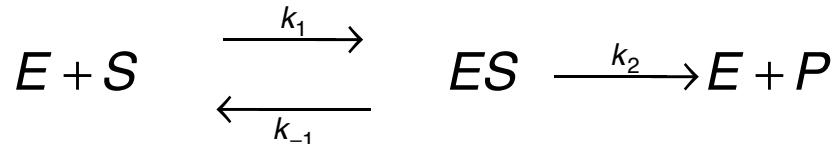
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

```
begin parameters
# Avogadro's number- scaled for umol
NA 6.02e23/1e6
```

observables

```
# Cell volume
V 1e-12 # liters - typical for eukaryote
```

functions

```
# Rate constants
kp1 1.0/(NA*V) # 1/uM 1/s-> 1/molec 1/s
km1 1.0e-1 # 1/s
k2 1.0e-2 # 1/s
```

reaction rules

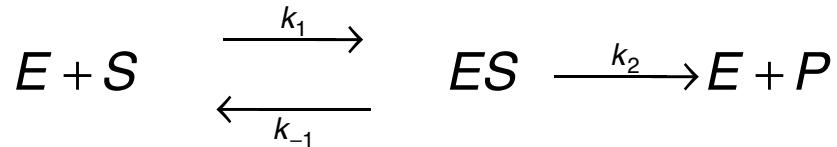
```
# Initial concentrations
E0 0.01*NA*V # uM -> molec / cell
S0 1.0*NA*V # uM -> molec / cell
```

actions

```
end parameters
```

Example 1: MM Mechanism

parameters



molecule types

molecule types—molecules, their components, and their allowed component states are declared here.

seed species

observables

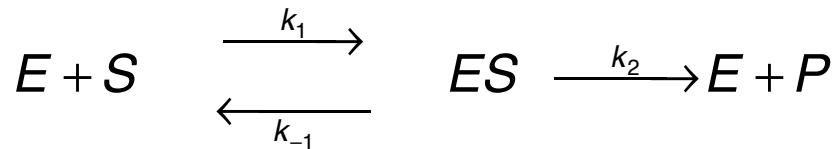
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

```
begin molecule types
  E(s)
  S(Y~0~P)
end molecule types
```

seed species

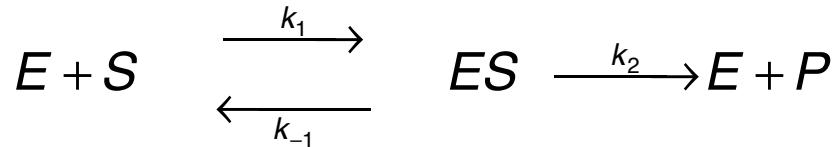
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

seed species— species initially present in the system at time t=0 followed by their initial concentration. Standard is all molecule types in their “ground state” with basal expression levels. May include complexes. All components of molecules that have states must be in a specified state. All complexes must be connected.

observables

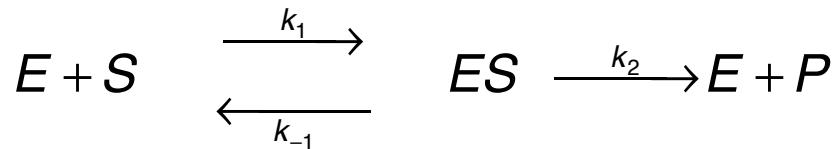
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

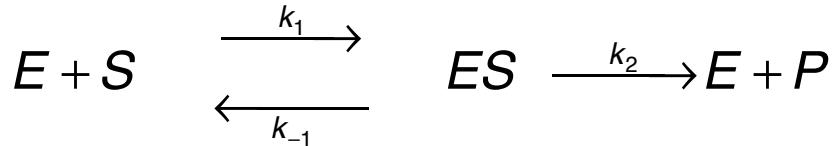
```
begin seed species
    E(s)    E0
    S(Y~0)  S0
end seed species
```

functions

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

observables— Defined sums of concentrations of species with specified properties. Syntax is <type> <name> <pattern>. Types considered here are Molecules and Species, which indicate weighted and unweighted sums respectively. These are used to define model outputs and are used as to make the default plot in RuleBender.

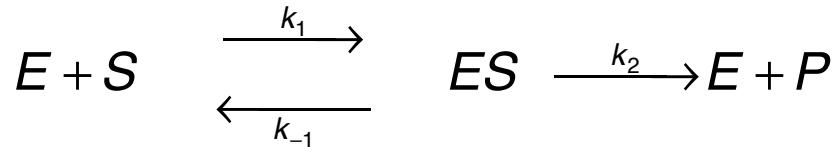
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

```
begin observables
Molecules SU S(Y~0)
Molecules SP S(Y~P)
Molecules ES E(s!1).S(Y!1)
end observables
```

seed species

observables

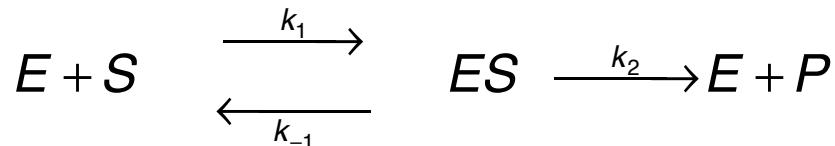
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

```
begin observables
Molecules SU S(Y~0)
Molecules SP S(Y~P)
Molecules ES E(s!1).S(Y!1)
end observables
```

observables

functions

reaction rules

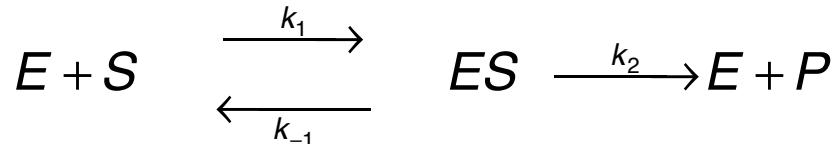
actions

observable	SU	S(Y~0)		S(Y~0)
matches			not	
species		S(Y~0)	E(s!1).S(Y~0!1)	

$SU = \text{sum of concentration of matches} = [S(Y~0)]$

Example 1: MM Mechanism

parameters



molecule types

seed species

```
begin observables
Molecules SU S(Y~0)
Molecules SP S(Y~P)
Molecules ES E(s!1).S(Y!1)
end observables
```

observables

functions

reaction rules

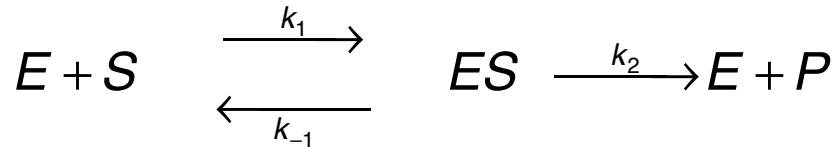
actions

observable ES $E(s!1).S(Y!1)$
matches
species $E(s!1).S(Y~0!1)$

$$ES = \text{sum of concentration of matches} = [E(s!1).S(Y~0!1)]$$

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

functions

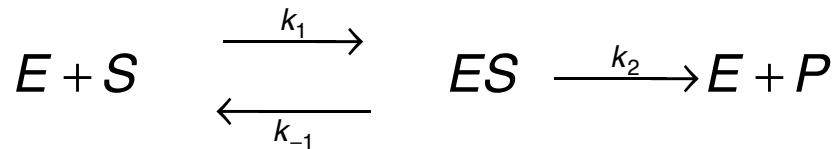
reaction rules

actions

reaction rules— Rules that generate reactions based on selecting reactants with specified properties and transforming them in a specified way with the specified rate law. Syntax is <name>: <reactants> <arrow> <products> <rate law>. Name is optional but useful.

Example 1: MM Mechanism

parameters



molecule types

seed species

```
begin reaction rules
```

observables

```
ESbind: \
E(s) + S(Y~0) <-> E(s!1).S(Y~0!1) kp1, km1
```

functions

```
ESconvert: \
E(s!1).S(Y~0!1) -> E(s) + S(Y~P) k2
```

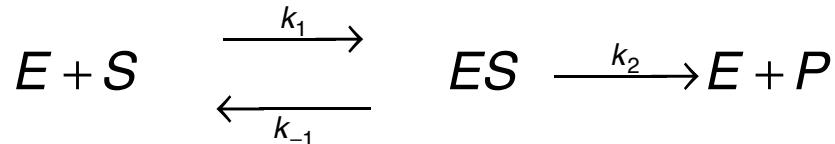
reaction rules

```
end reaction rules
```

actions

Example 1: MM Mechanism

parameters



molecule types

actions– Need not be enclosed in block. Come after model definition and specify simulation protocol for a model.

seed species

```
generate_network({});  
simulate_ode({t_end=>1000,n_steps=>100});
```

observables

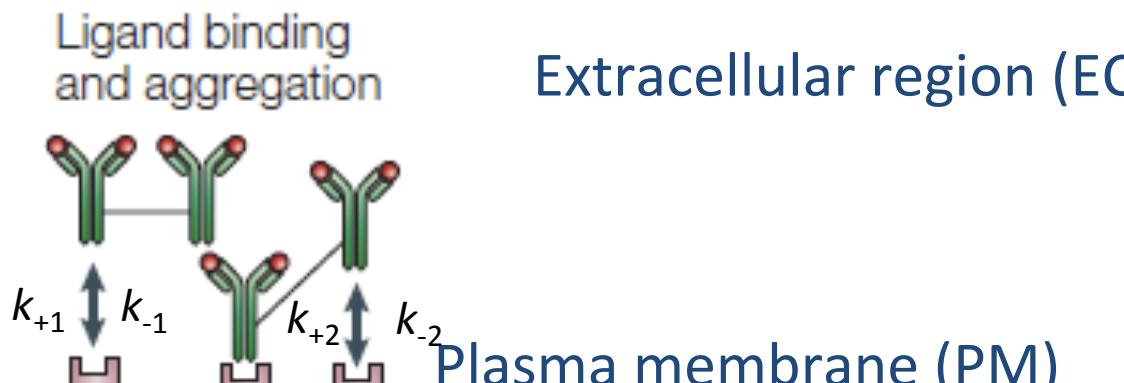
functions

reaction rules

actions

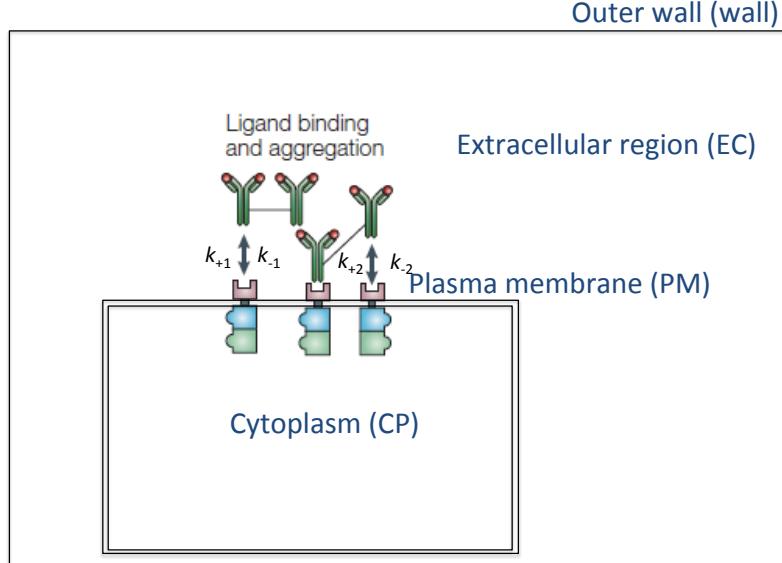
Dimerization Model

Outer wall (wall)



Cytoplasm (CP)

Compartment Specification



```
begin compartments
wall 2 vol_wall
EC   3 vol_EC
PM   2 vol_PM
CP   3 vol_CP
end compartments
```

wall
EC
PM

Volume of surface compartment = Area*thickness
thickness = 10 nm = 0.01 μm

Example 2: Synthesis and Degradation

So far we have covered three of the five elementary types of transformations that can be carried out in a BNG rule: bond addition, bond removal, and component state change. The other two types are molecule creation and molecule deletion. These five elementary types are the building blocks of all transformations.

To illustrate these two additional elementary transformations, we will return to the synthesis and degradation model we solved in the notes on Chemical Kinetics.



To model this system in BNG, we need to define two molecule types:

```
begin molecule types
  I()
  A(b)
end molecule types
```

where `I()` is a dummy molecule that will be used as a placeholder in the rules and `A` is the molecule that will be synthesized and degraded. We give it a component `b` in anticipation of future extension of the model. The reaction rules are

```
begin reaction rules
  syn: I() -> I() + A(b) beta
  deg: A() + I() -> I() alpha
end reaction rules
```

Example 2: Synthesis and Degradation

So far we have covered three of the five elementary types of transformations that can be carried out in a BNG rule: bond addition, bond removal, and component state change. The other two types are molecule creation and molecule deletion. These five elementary types are the building blocks of all transformations.

To illustrate these two additional elementary transformations, we will return to the synthesis and degradation model we solved in the notes on Chemical Kinetics.



To model this system in BNG, we need to define two molecule types:

```
begin molecule types  
I()  
A(b)  
end molecule types
```

where `I()` is a dummy molecule that will be used as a placeholder in the rules and `A` is the molecule that will be synthesized and degraded. We give it a component `b` in anticipation of future extension of the model. The reaction rules are

```
begin reaction rules  
syn: I() -> I() + A(b) beta  
deg: A() + I() -> I() alpha  
end reaction rules
```

The molecule that is produced must be a species.

I is not consumed or produced by either rule.

If A is part of a complex, the entire complex will be destroyed by the reaction.

Example 2: Synthesis and Degradation



Here is the complete BNGL file for this example:

```
begin model
begin parameters
  alpha 1
  beta 10
  A0 0
end parameters

begin molecule types
  I()
  A(b)
end molecule types

begin seed species
  I() 1
  A(b) A0
end seed species
```

```
begin observables
  Molecules Atot A()
end observables

begin reaction rules
  I() -> I() + A(b) beta
  A() + I() -> I() alpha
end reaction rules

end model

#ACTIONS
generate_network({});
simulate_ode({t_end=>5,n_steps=>100});
```

Units: Bridging Continuum and Discrete Limits

In general, continuum chemical kinetics is formulated using *intrinsic* parameters – they do not depend on volume

$$[\text{intrinsic}] \text{ rate (M/s)} = k_2 (1/\text{M 1/s}) [\text{A}] (\text{M}) [\text{B}] (\text{M})$$

In stochastic chemical kinetics, a finite volume V is assumed, and in general the rate constants depend on V – they are *extrinsic*

$$[\text{extrinsic}] \text{ rate (#/s)} = c_2 (1/\#\ 1/\text{s}) n_A (\#) n_B (\#)$$

To determine how c_2 and k_2 are related we divide the extrinsic rate by $N_A V$ to get

$$\text{rate (M/s)} = c_2 / N_A V \times n_A n_B = c_2 / N_A V \times (N_A V)^2 n_A / N_A V n_B / N_A V$$

$$\text{rate (M/s)} = c_2 \times (N_A V) [\text{A}] [\text{B}] (\text{M/s})$$



$$c_2 = k_2 / N_A V$$

Units: Bridging Continuum and Discrete Limits

In general, for a reaction with molecularity m we have

$$\text{rate (M/s)} = c_m / N_A V \times n_1 \dots n_m = c_m / N_A V \times (N_A V)^m n_1 / N_A V \dots n_m / N_A V$$



$$c_m = k_m \times (N_A V)^{-(m-1)}$$

(zeroth order)

$$c_0 = k_0 \times (N_A V)$$

(first order)

$$c_1 = k_1$$

(second order)

$$c_2 = k_2 / N_A V$$

Faeder Lab

current

John Sekar
Jose Juan Tapia
Robert Sheehan
Cihan Kaya

alumni

Dipak Barua
Qingyang Ding
Leonard Harris
Justin Hogg
Jintao Liu
Natasa Miskov-Zivanov
Yang Yang

Liz Marai

John Wenskovitch
Adam Smith
Wen Xu
Yao Sun

Dan Zuckerman

Rory Donovan

rulebender.org

bionetgen.org

nfsim.org

mcell.org

Thierry Emonet
Michael Sneddon

Jason Haugh

Penelope Morel
Larry Kane
Michael Turner
William Hawse

Sarah Gaffen
Abhishek Garg

Markus Dittrich
Jacob Czech
Tom Bartol
Terry Sejnowski
Ivet Bahar
Bing Liu
Bob Murphy
Devin Sullivan

William Hlavacek
Lily Chylek

Byron Goldstein
Ambarish Nag
Michael Monine

Michael Blinov

Edmund Clarke
Haijun Gong
Paolo Zuliani

Peter Setlow
Barbara Setlow
Xuan Yi

NIH-NIGMS P41, P50, R01
NIH-NIAID R01
NSF Expeditions, EMT
NIH-NCRR CTSI
DOD MURI