Spatial Rule-based Modeling of Cellular Biochemistry with MCell/ BioNetGen/CellBlender

Jim Faeder, Ph.D.
Univ. of Pittsburgh, MMBioS

Markus Dittrich, Ph.D.
Pittsburgh Supercomputing Center,
Carnegie Mellon University,
MMBioS

GLBIO 2013 Tutorial, May 12

Goto http://bionetgen.org/index.php/GLBIO2013 for slides and other materials

Overview

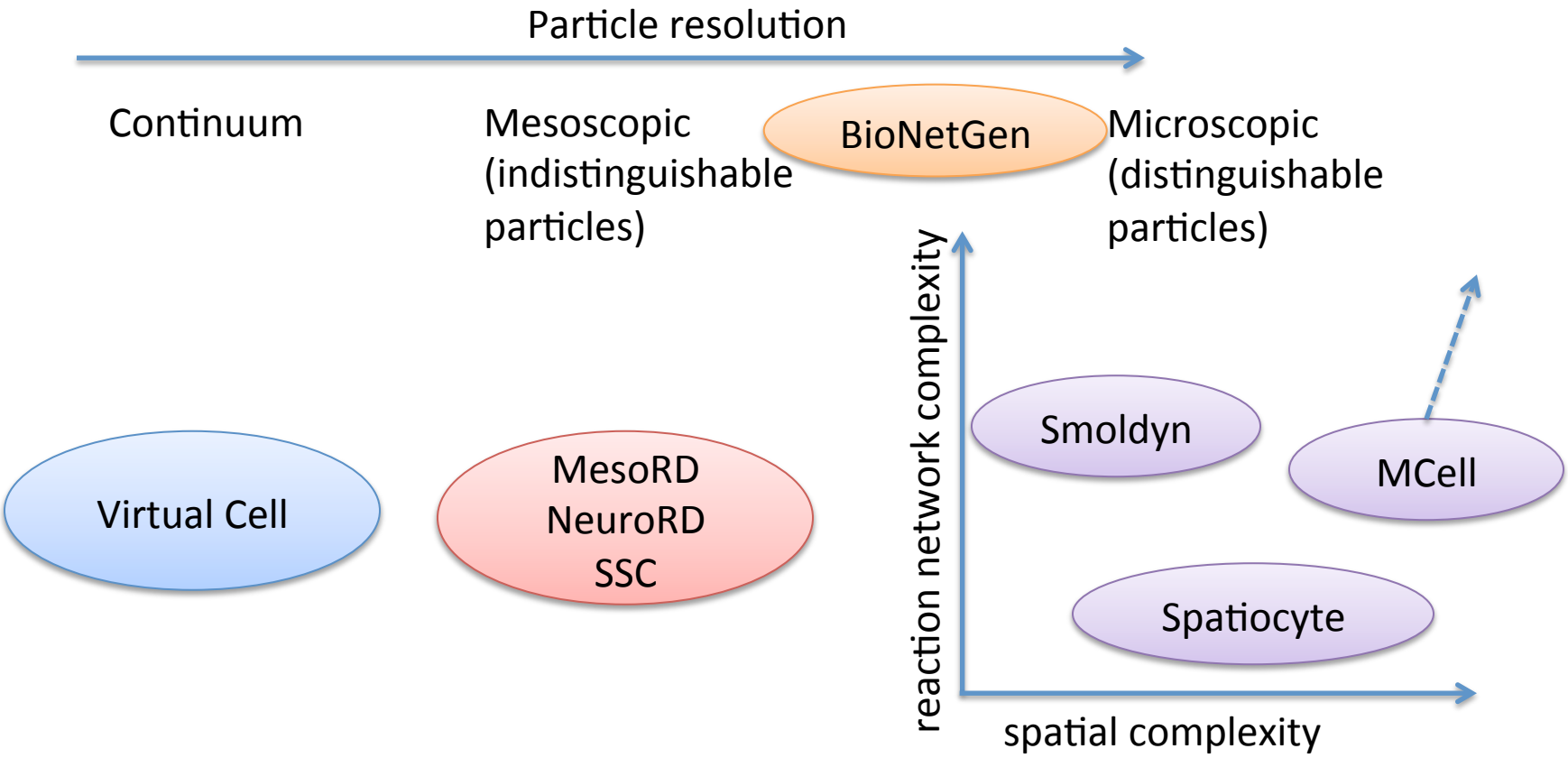
- Multiscale Challenge (Dittrich)
- Intro to National Center for Multiscale Modeling of Biological Systems (Faeder)
- BioNetGen Motivation and Intro (Faeder)
- BioNetGen/RuleBender Demo (Faeder)
- MCell Intro (Dittrich)
- Mcell/CellBlender Demo (Dittrich)
- Features in Progress and Q&A

www.nrbsc.org

The Multiscale Computational Challenge

Problem/ Method	Typical Application	Software Examples	Resolution (Scale)	Spatial Realism	Stochastic Realism	Time Step	Time Scale	Serial/ Parallel	Computer Time
Networks of Reactions/ Sets of ODEs	Metabolic or signaling pathways	Virtual Cell ECell, Gepasi XPPAUT	N/A (cell)	N/A	<none>	ms	ms - hrs	serial	minimal
Excitation/ Compartmental Circuit	Nerve signaling	NEURON GENESIS NEOSIM	μm - mm (cell - multicell)	low - medium	none	ms	ms - hrs	usually serial	usually low
Reaction Kinetics/ Stochastic	Gene regulation/ transcription	BioSpice StochSim XPPAUT MCell	N/A (cell)	N/A	high	ms	ms - hrs	serial	low
3-D Reaction Diffusion/ Finite Element	Flow models, calcium dynamics	Virtual Cell FIDAP Kaskade	<μm (cell)	medium - high	<none>	μs - ms	μs - sec	either	low - high
3-D Reaction Diffusion/ Monte Carlo	Micro-physiological processes	MCell ChemCell SmolDyn	nm – mm (subcell - cell)	high	high	ps - ms	μs - sec	either	low - high
Macromolecular Machinery/GNM	Collective dynamics	GNM ANM	Å - 100 nm (complexes)	high	none	N/A	<ns – μs>	N/A (analytic)	minimal
Diffusion in Potential Field/Poisson-Nernst-Planck	Electrostatic interactions, ion channels	UHBD Delphi CHARMM	Å - nm (membrane proteins)	high (implicit solvent)	none	N/A	<ns – μs>	parallel	low - medium
Macromolecular Motions/Brownian Dynamics (BD)	Conformational dynamics (in flow fields)	CHARMM GROMOS UHBD	Å - nm (macro-molecules)	high (implicit solvent)	high	5 - 10 fs	<ns – μs>	parallel	medium - high
Molecular Structure/ Molecular Dynamics (MD)	Conformational dynamics & free energies	NAMD AMBER CHARMM GROMOS	Å (macro-molecules)	exact (explicit solvent)	exact	1 - 2 fs	<ns – μs>	parallel	very high
Transition Dynamics/ Quantum Chem. + Mol. Mech. (QC/MM)	Enzyme reactions (make/break bonds)	pDynamo (AMBER CHARMM)	Å (molecules)	exact (explicit solvent)	exact	1 - 2 fs	<ns – μs>	parallel	very high
Molecular Structure/ *Ab initio* simulations	Solution of the Schrodinger equation	Gaussian	<Å (electrons - atoms)	exact	exact	N/A	N/A	parallel	highest

Comparison with other cell simulation tools



Biomedical Technology Research Center (BTRC)

High Performance Computing

for

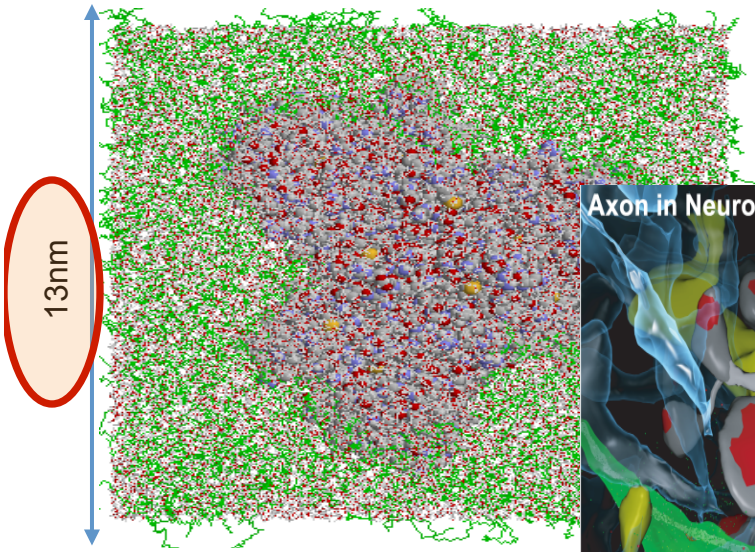
Multiscale Modeling of Biological Systems

Overarching biological theme:

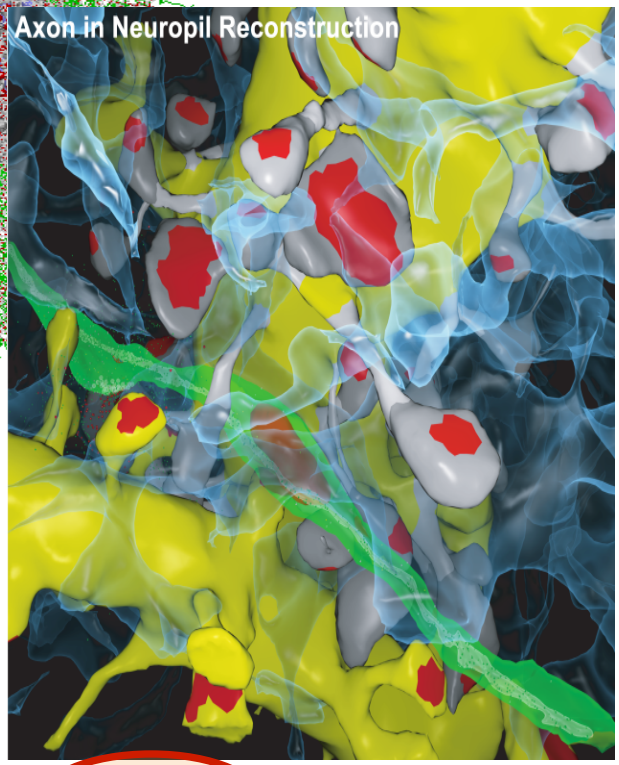
- Spatial organization
- Temporal evolution

of (neuro)signaling systems/events

From small molecules, to multimeric assemblies,

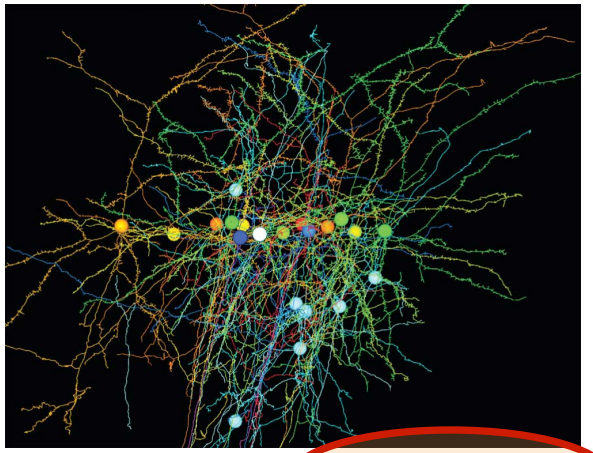


to cellular architecture,



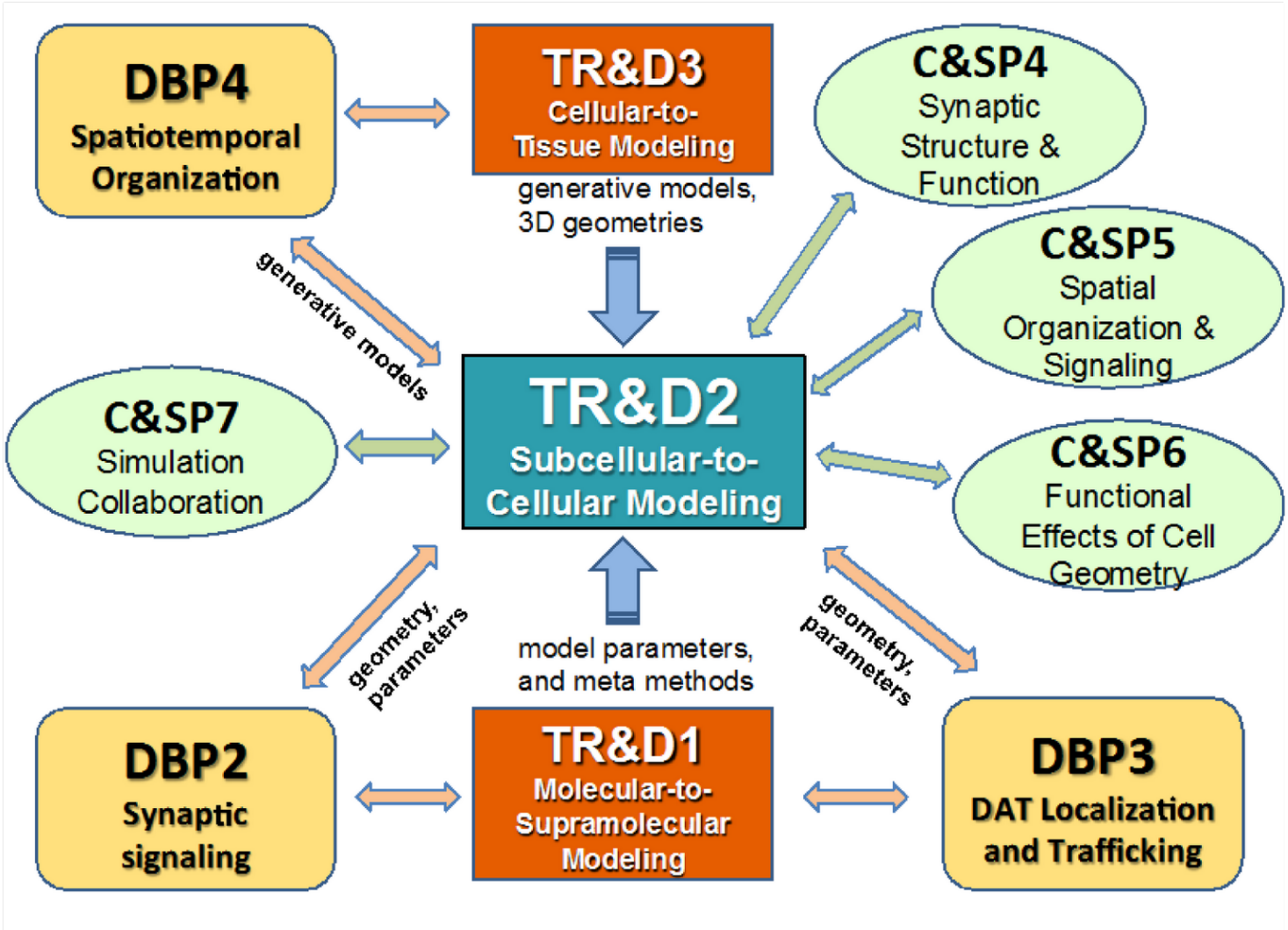
from 6 x 6 x 5 μm^3 sample of adult rat hippocampal stratum radiatum neuropil

to neural circuits

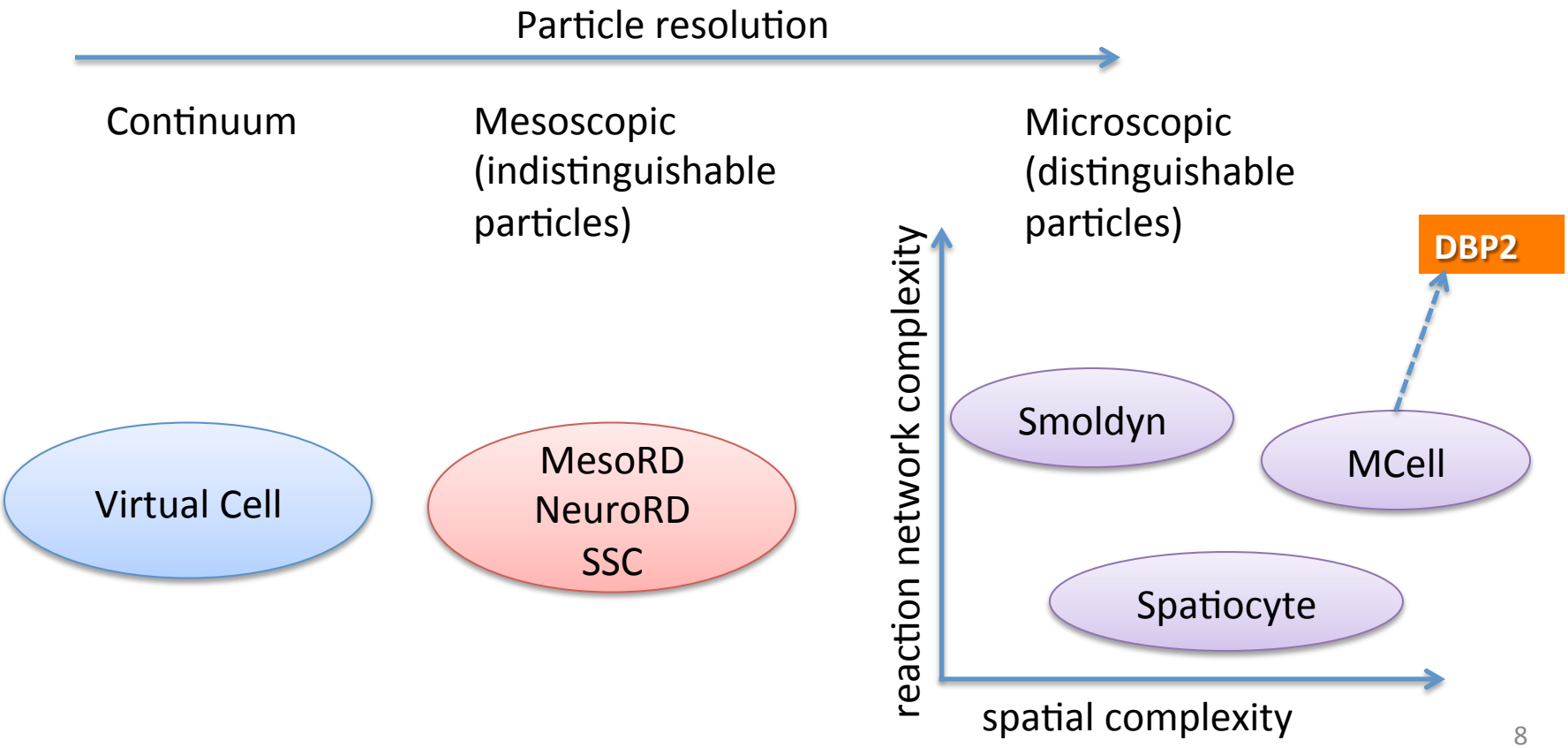


From SSEM images, 400 x 400 x 50 μm^3

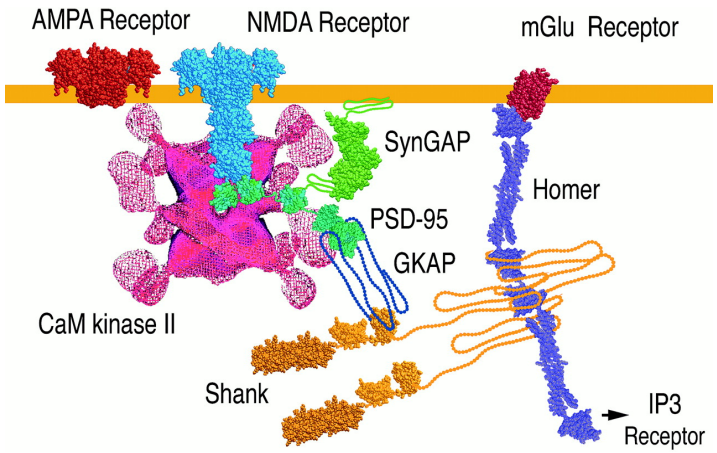
Role of MCell in the BTRC



Comparison of MCell with other tools for spatial modeling of biological systems



Motivating example for Rule-Based Modeling



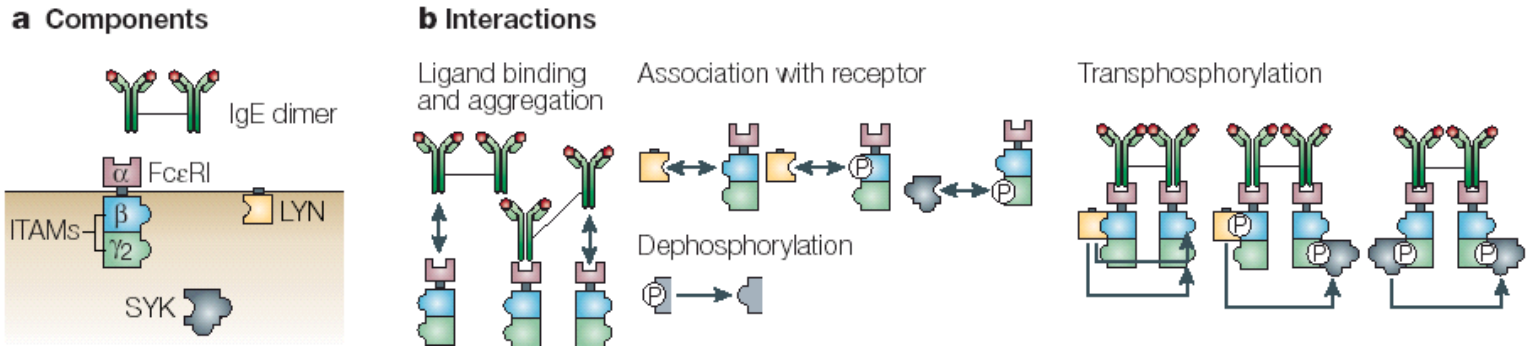
Molecular machines in the PSD

Estimated number of states of CAMKII-CaM complexes:

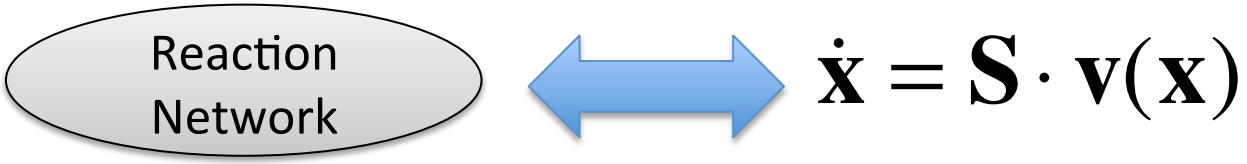
40^{12}

Standard modeling protocol

1. Identify components and interactions.

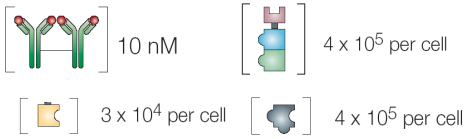


2. Write model reactions / equations



$$\dot{\mathbf{x}} = \mathbf{S} \cdot \mathbf{v}(\mathbf{x})$$

3. Determine **concentrations** and **rate constants**

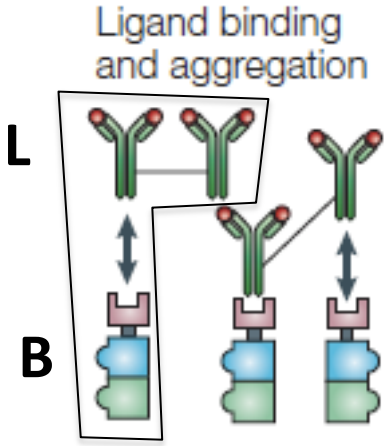


4. Simulate and analyze the model

Reactions to Differential Equations

Consider the reaction

$$R + L \xrightarrow{k_1} RL$$



The reaction rate is given by

$$v_1 = k_{1_{k_2}} R \cdot L$$

Rate of change of species concentrations (numbers) are

$$\frac{dR}{dt} = -v_1 + \cdots$$

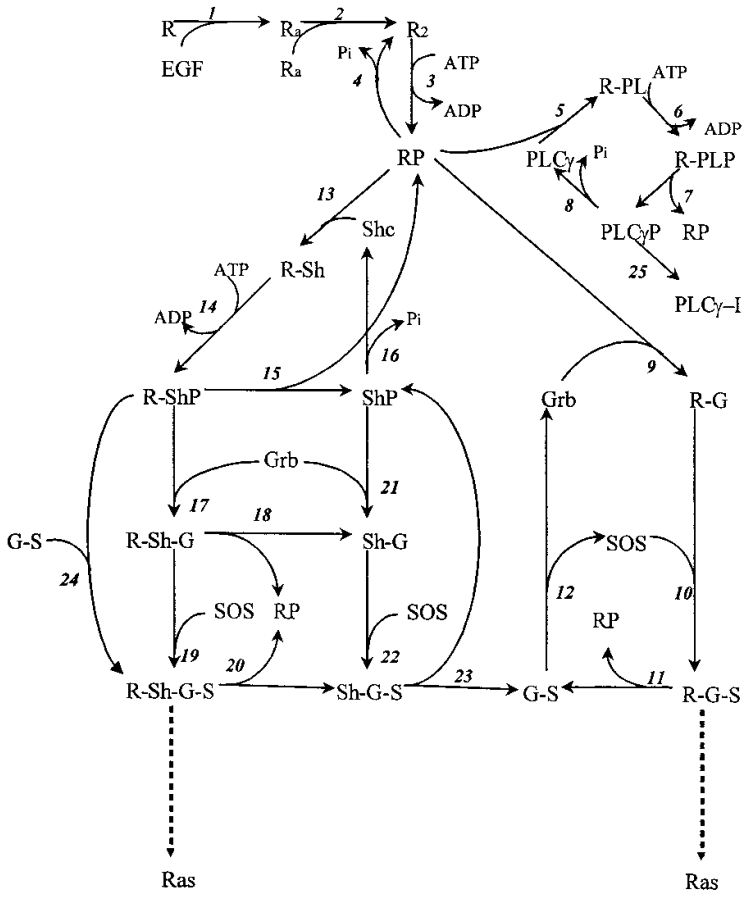
$$\frac{dL}{dt} = -v_1 + \cdots$$

$$\frac{dRL}{dt} = +v_1 + \cdots$$

Here I have indicated that there may be additional terms from other reactions in the network. Reaction fluxes combine through *addition.*

Reaction Network Models

Reaction Network Scheme ⟺ Mathematical Formulation



Rate Equations

$k_1 \cdot [R] \cdot [EGF] - k_{-1} \cdot [R_a]$
$k_2 \cdot [R_a] \cdot [R_a] - k_{-2} \cdot [R_2]$
$k_3 \cdot [R_2] - k_{-3} \cdot [RP]$
$V_4 \cdot [RP]/(K_4 + [RP])$
$k_5 \cdot [RP] \cdot [PLC\gamma] - k_{-5} \cdot [R\text{-}PL]$
$k_6 \cdot [R\text{-}PL] - k_{-6} \cdot [R\text{-}PLP]$
$k_7 \cdot [R\text{-}PLP] - k_{-7} \cdot [RP] \cdot [PLC\gamma P]$
$V_8 \cdot [PLC\gamma P]/(K_8 + [PLC\gamma P])$
$k_9 \cdot [RP] \cdot [Grb] - k_{-9} \cdot [R\text{-}G]$
$k_{10} \cdot [R\text{-}G] \cdot [SOS] - k_{-10} \cdot [R\text{-}G\text{-}S]$
$k_{11} \cdot [R\text{-}G\text{-}S] - k_{-11} \cdot [RP] \cdot [G\text{-}S]$
$k_{12} \cdot [G\text{-}S] - k_{-12} \cdot [Grb] \cdot [SOS]$
$k_{13} \cdot [RP] \cdot [Shc] - k_{-13} \cdot [R\text{-}Sh]$
$k_{14} \cdot [R\text{-}Sh] - k_{-14} \cdot [R\text{-}ShP]$
$k_{15} \cdot [R\text{-}ShP] - k_{-15} \cdot [ShP] \cdot [RP]$
$V_{16} \cdot [ShP]/(K_{16} + [ShP])$
$k_{17} \cdot [R\text{-}ShP] \cdot [Grb] - k_{-17} \cdot [R\text{-}Sh\text{-}G]$
$k_{18} \cdot [R\text{-}Sh\text{-}G] - k_{-18} [RP] \cdot [Sh\text{-}G]$
$k_{19} \cdot [R\text{-}Sh\text{-}G] \cdot [SOS] - k_{-19} \cdot [R\text{-}Sh\text{-}GS]$
$k_{20} \cdot [R\text{-}Sh\text{-}G\text{-}S] - k_{-20} \cdot [Sh\text{-}G\text{-}S] \cdot [RP]$
$k_{21} \cdot [ShP] \cdot [Grb] - k_{-21} \cdot [Sh\text{-}G]$
$k_{22} \cdot [Sh\text{-}G] \cdot [SOS] - k_{-22} \cdot [Sh\text{-}G\text{-}S]$
$k_{23} \cdot [Sh\text{-}G\text{-}S] - k_{-23} \cdot [ShP] \cdot [G\text{-}S]$
$k_{24} \cdot [R\text{-}ShP] \cdot [G\text{-}S] - k_{-24} \cdot [R\text{-}Sh\text{-}G\text{-}S]$
$k_{25} \cdot [PLC\gamma P] - k_{-25} \cdot [PLC\gamma P\text{-}I]$

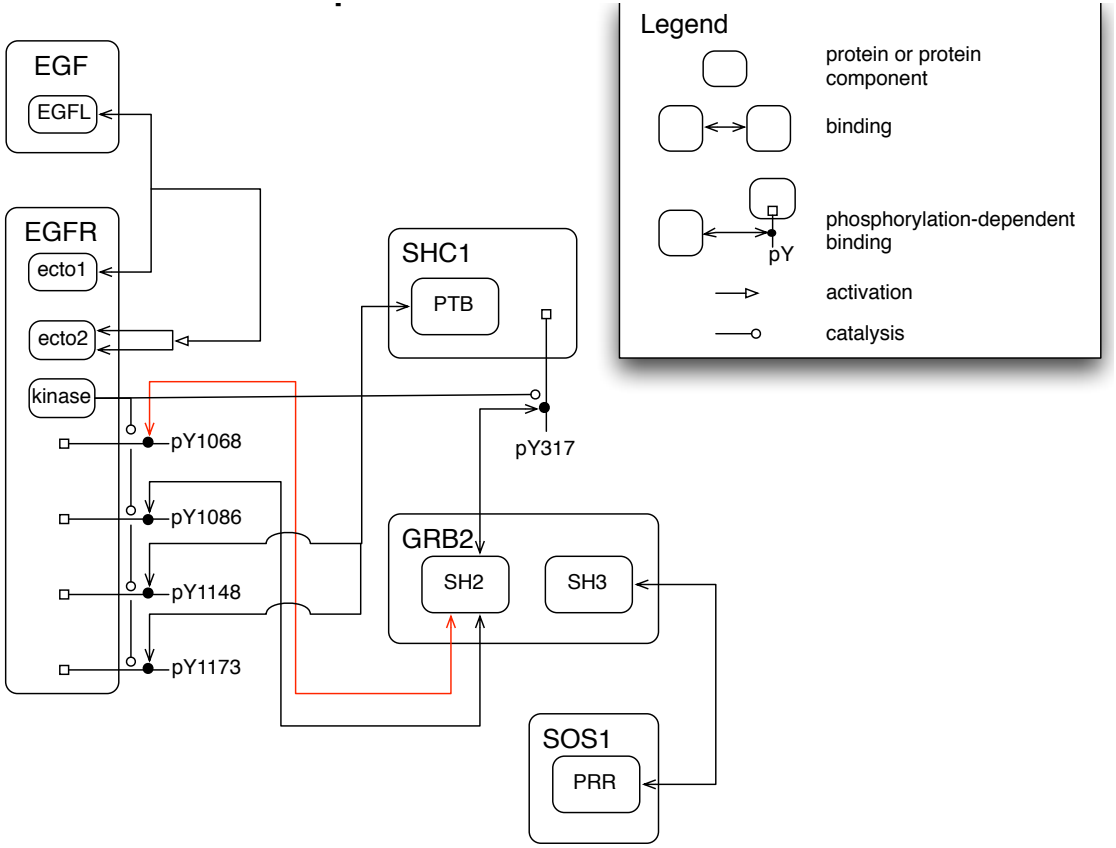
Differential Equations

$d[EGF]/dt = -v_1$
$d[R]/dt = -v_1$
$d[R_a]/dt = v_1 - 2v_2$
$d[R_2]/dt = v_2 + v_4 - v_3$
$dRP/dt = v_3 + v_7 + v_{11} + v_{15} + v_{18} + v_{20} - v_4 - v_5 - v_9$
$d[R\text{-}PL]/dt = v_5 - v_6$
$d[R\text{-}PLP]/dt = v_6 - v_7$
$d[R\text{-}G]/dt = v_9 - v_{10}$
$d[R\text{-}G\text{-}S]/dt = v_{10} - v_{11}$
$d[R\text{-}Sh]/dt = v_{13} - v_{14}$
$d[R\text{-}ShP]/dt = v_{14} - v_{24} - v_{15} - v_{17}$
$d[R\text{-}Sh\text{-}G]/dt = v_{17} - v_{18} - v_{19}$
$d[R\text{-}Sh\text{-}G\text{-}S]/dt = v_{19} - v_{20} + v_{24}$
$d[G\text{-}S]/dt = v_{11} + v_{23} - v_{12} - v_{24}$
$d[ShP]/dt = v_{15} + v_{23} - v_{21} - v_{16}$
$d[Sh\text{-}G]/dt = v_{18} + v_{21} - v_{22}$
$d[PLC\gamma]/dt = v_8 - v_5$
$d[PLC\gamma P]/dt = v_7 - v_8 - v_{25}$
$d[PLC\gamma P\text{-}I]/dt = v_{25}$
$d[Grb]/dt = v_{12} - v_9 - v_{17} - v_{21}$
$d[Shc]/dt = v_{16} - v_{13}$
$d[SOS]/dt = v_{12} - v_{10} - v_{19} - v_{22}$

22 species / 25 reactions

Kholodenko et al., *J. Biol. Chem.* (1999)

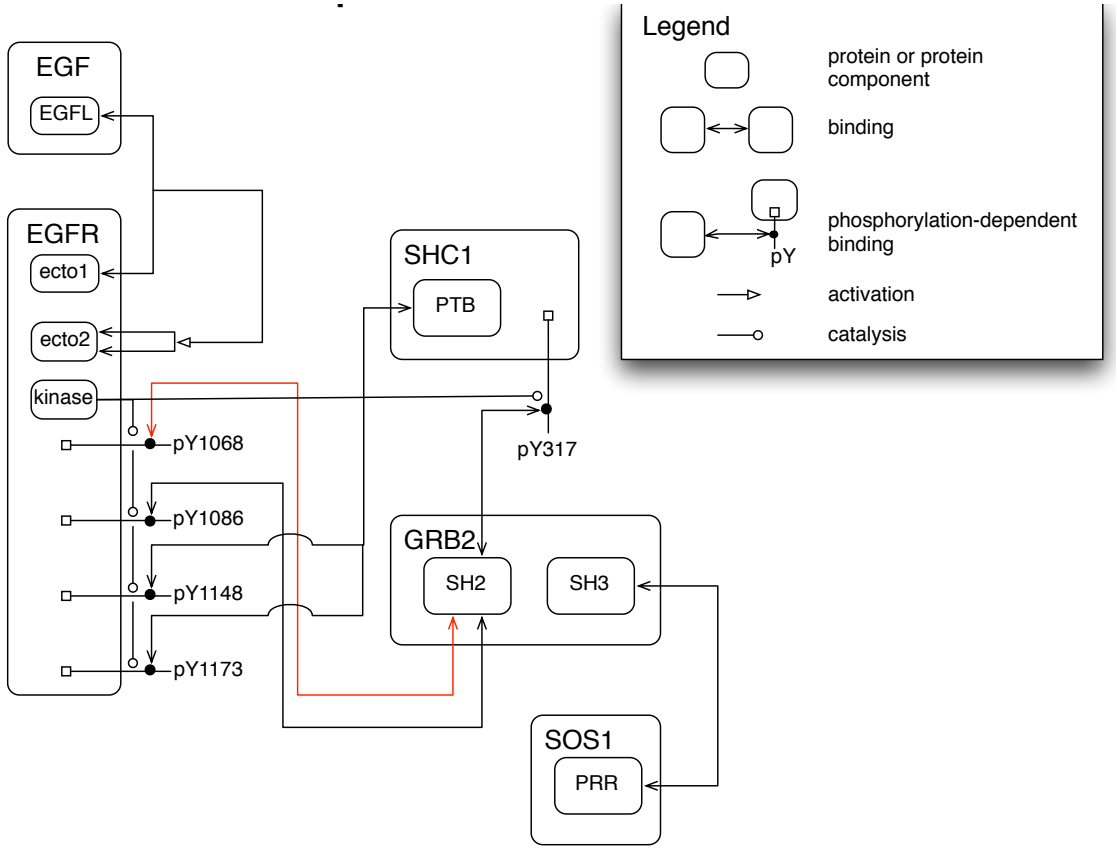
Combinatorial complexity in a prototypical signaling module



5 proteins, 20 interactions → 170,000 species

… but only 20 (or so) **rules**.

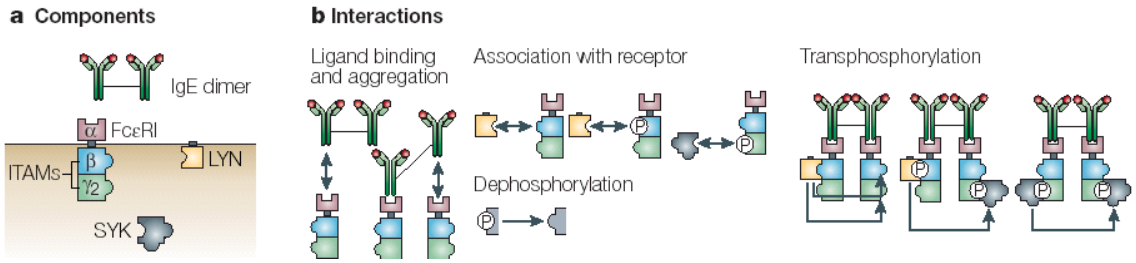
Rules provide a scalable way to model molecular interactions



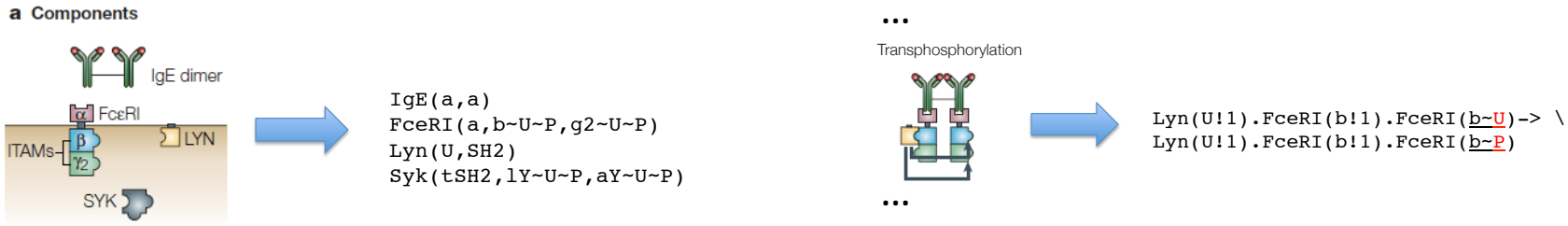
Rules ~ number of interactions << number of species

Rule-Based Modeling protocol

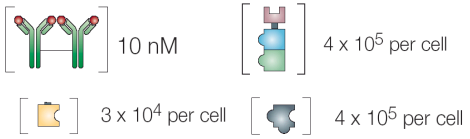
1. Identify components and interactions.



2. Translate into objects (molecules) and rules



3. Determine **concentrations** and **rate constants**

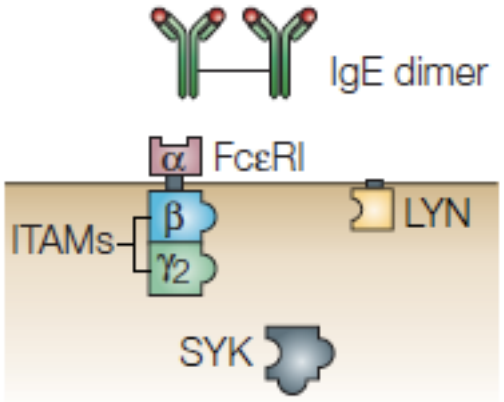


4. Simulate and analyze the model

SPECIFYING A RULE-BASED MODEL

Defining Molecules

Molecules are the basic objects in a BNG model



BioNetGen Language

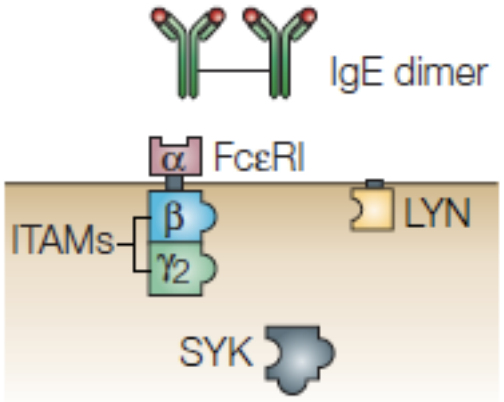
```
IgE(a,a)
FceRI(a,b~U~P,g2~U~P)
Lyn(U,SH2)
Syk(tSH2,lY~U~P,aY~U~P)
```

Components represent molecule elements

- Domains
- Motifs
- Properties

Defining Molecules

Molecules are the basic objects in a BNG model



BioNetGen Language

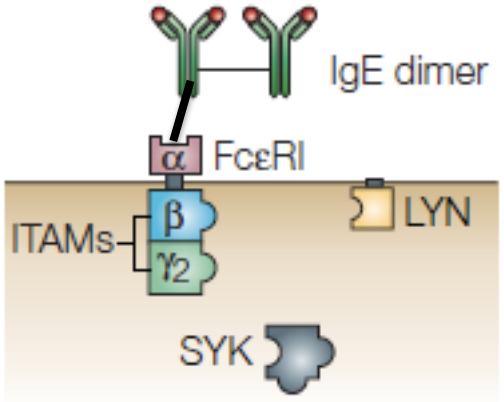
```
IgE(a,a)
FceRI(a,b~U~P,g2~U~P)
Lyn(U,SH2)
Syk(tSH2,lY~U~P,aY~U~P)
```

Components may have different states representing

- posttranslational modifications
- conformational state
- …

Binding

Molecules bind other molecules through components



BIONETGEN Language

```
IgE(a,a!1).FceRI(a!1,b~U,g2~U)
```

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

```
FceRI(a,b~U!1,g2~U).Lyn(U!1)
```

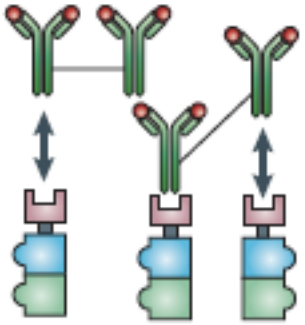
Components may have both states and bonds.

```
Lyn(SH2!1,Cterm~P!1)
```

Bonds may occur within a molecule.

Defining Interaction Rules

Ligand binding and aggregation

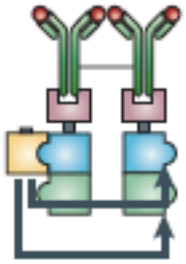


BioNetGen Language

```
IgE(a,a)+ FceRI(a)<-> IgE(a,a!1).FceRI(a!1)
…
```

binding and dissociation

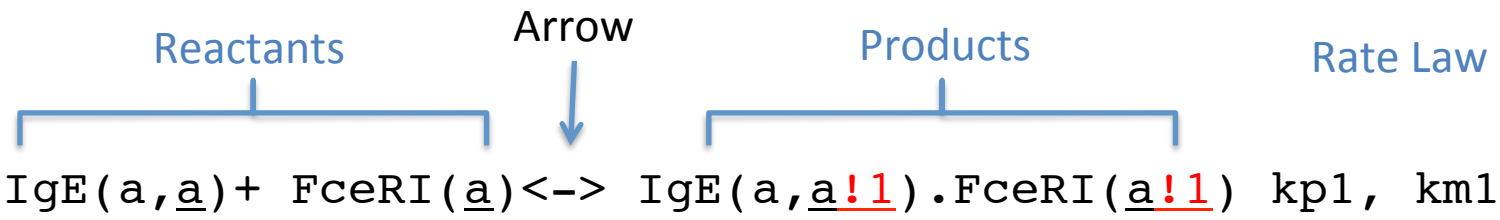
Transphosphorylation



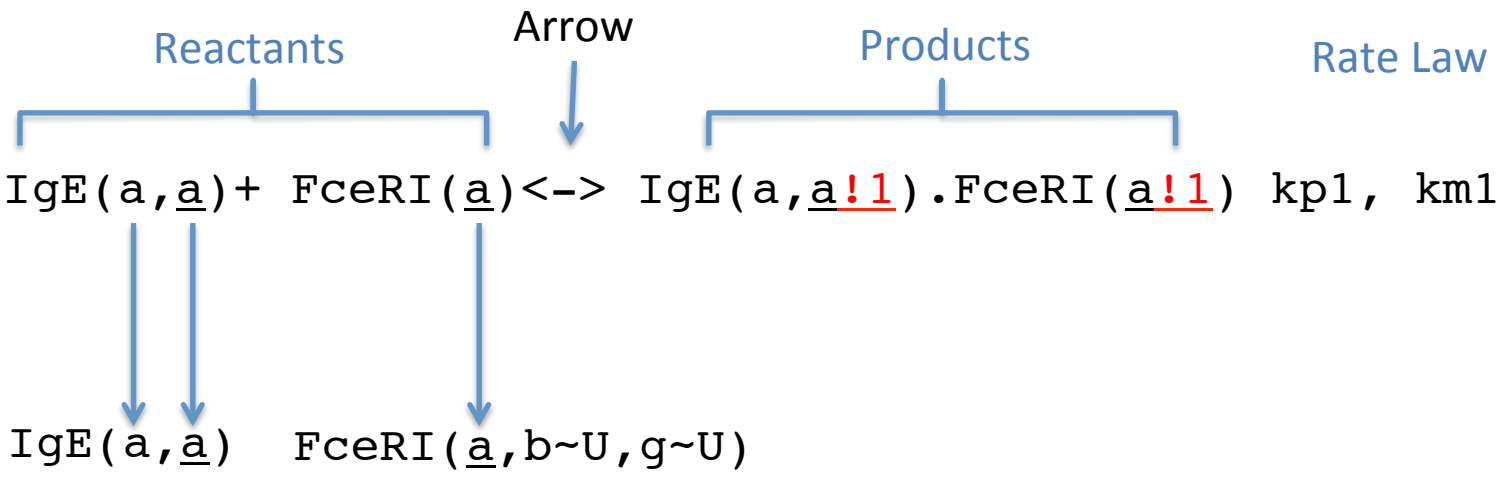
```
Lyn(U!1).FceRI(b!1).FceRI(b~U)-> \
Lyn(U!1).FceRI(b!1).FceRI(b~P)
```

component state change

Parts of a rule

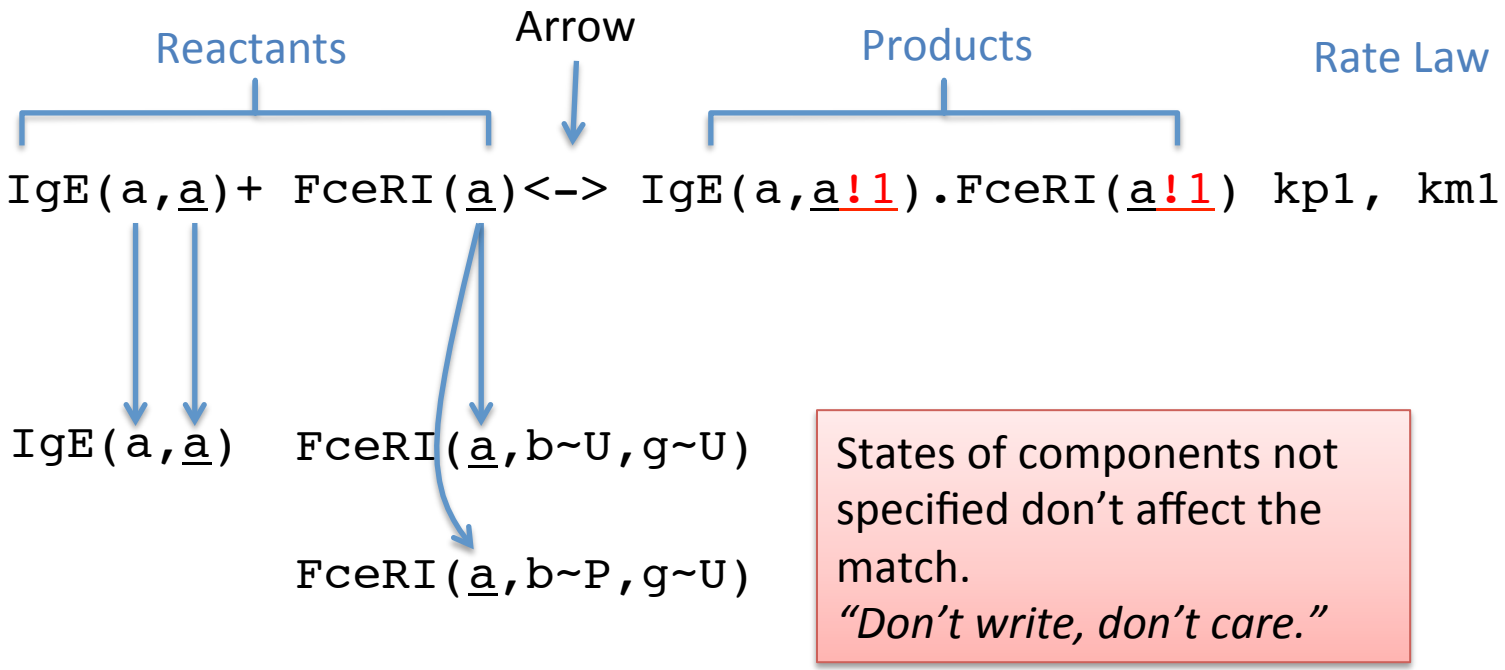


Parts of a rule



Reactant patterns select properties of each reactant molecule.

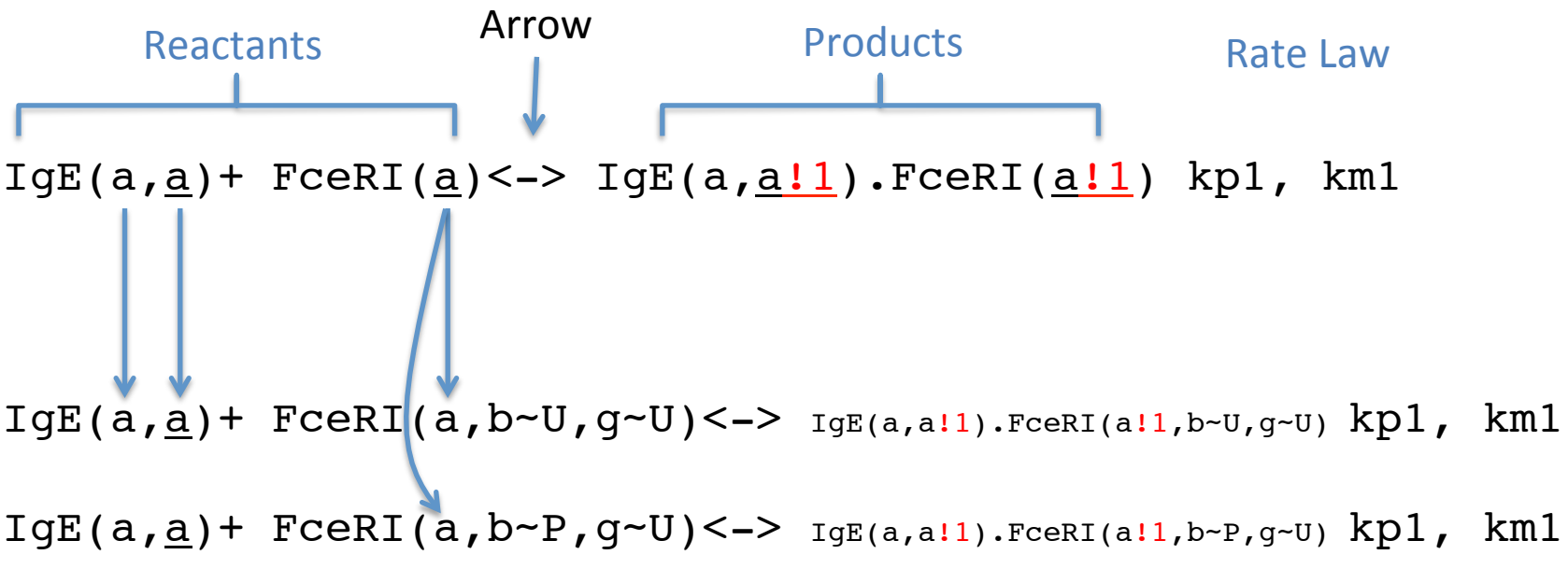
Parts of a rule



States of components not specified don't affect the match.
"Don't write, don't care."

Reactant patterns select properties of each reactant molecule.

Parts of a rule

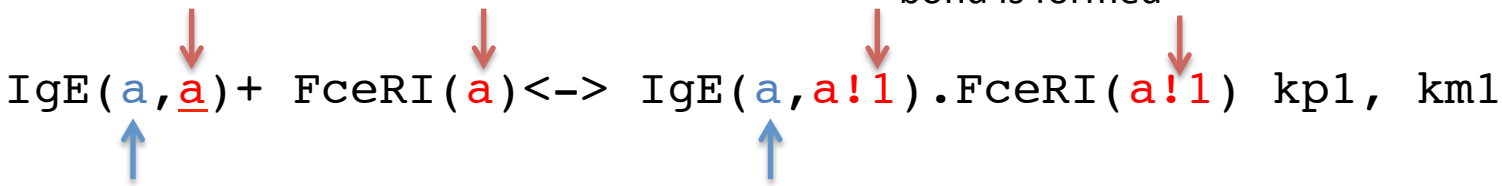


Reactant patterns select properties of each reactant molecule.

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

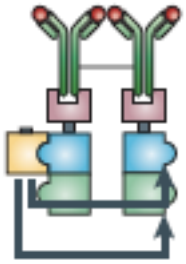
The **center** of a rule is the part that the rule changes.

bond is formed

```
IgE(a,a)+ FceRI(a)<-> IgE(a,a!1).FceRI(a!1) kp1, km1
```

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

Transphosphorylation

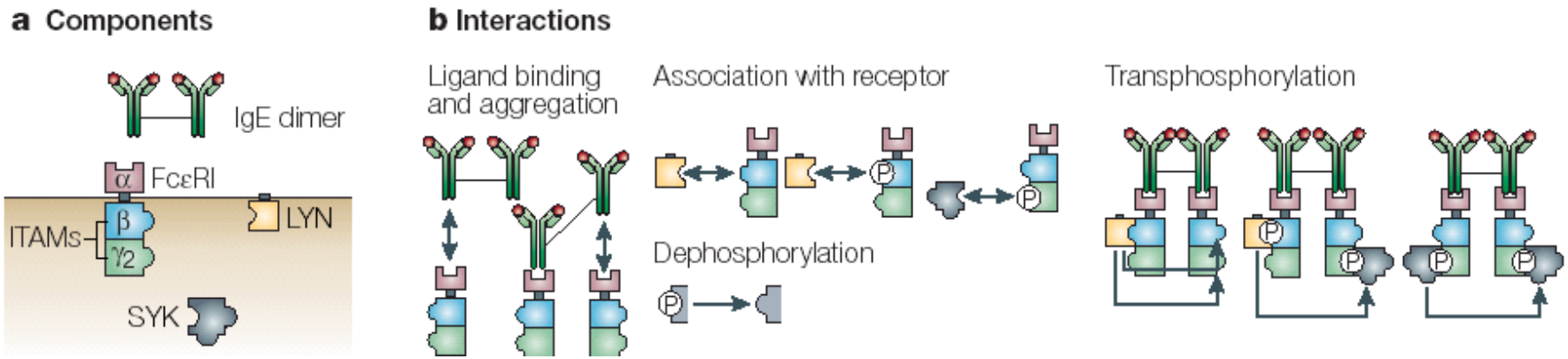


Context can represent complex biochemistry.

component state is changed

```
Lyn(U!1).FceRI(b!1).FceRI(b~U)-> \
Lyn(U!1).FceRI(b!1).FceRI(b~P)
```

Composition of a Rule-Based Model



Molecules

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

Reaction Rules

BioNetGen language

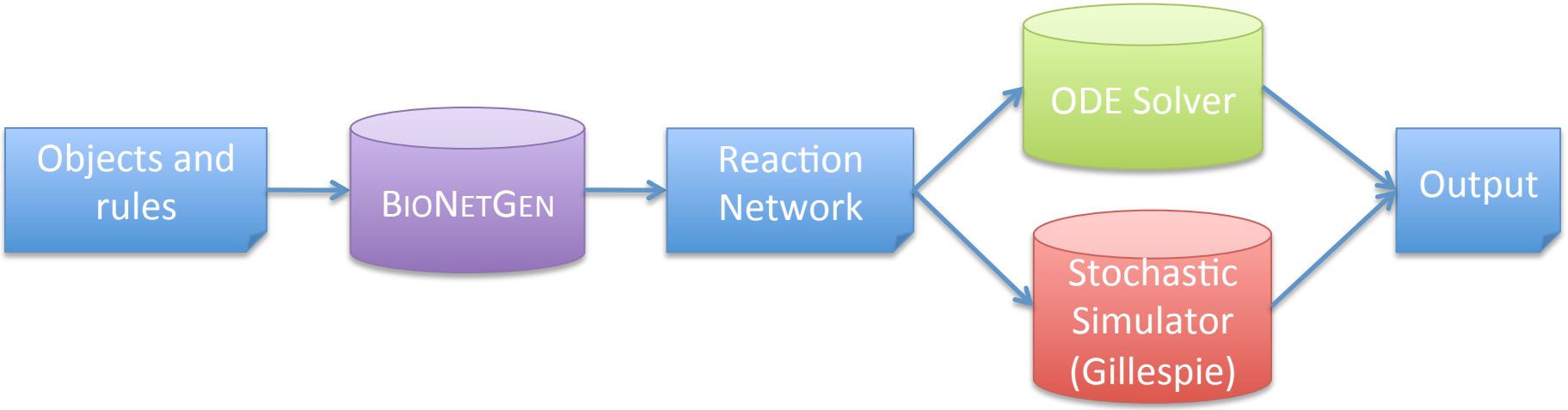
```
begin reaction_rules
# Ligand-receptor binding
1 Rec(a) + Lig(l,l) <-> Rec(a!1).Lig(l!1,l) kp1, km1
 Rec(a) + Lig(l,l) <-> Rec(a!1).Lig(l!1,l) kp1, km1

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

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

Applications

- Immunoreceptor Signaling
- Growth factor receptor signaling
- Multivalent binding
- Scaffold effects
- Yeast pheromone signaling
- For a complete list of BioNetGen Applications see http://bionetgen.org/Model_Examples.

SIMULATING A RULE-BASED MODEL

Basic RBM workflow with BioNetGen

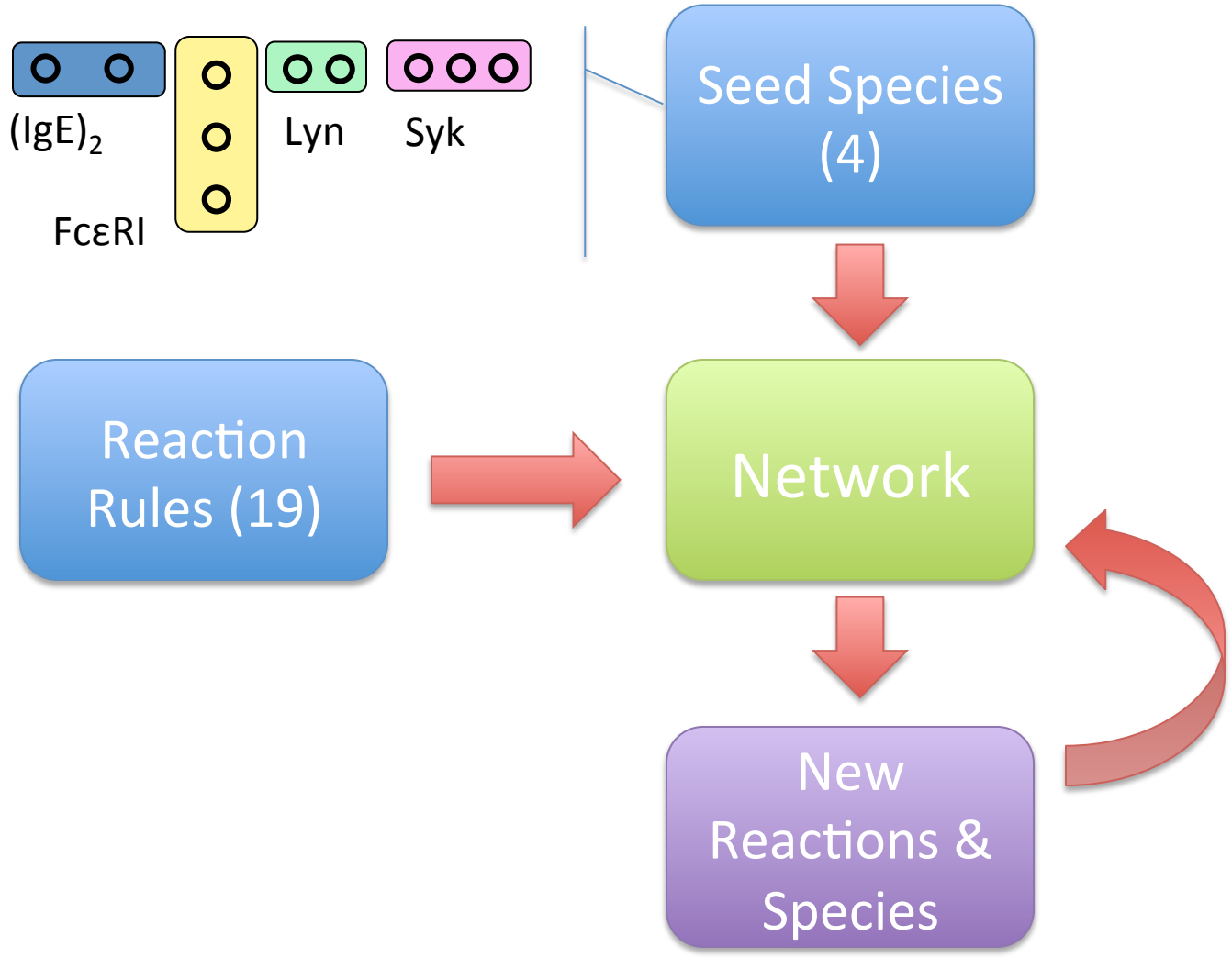


http://bionetgen.org

Faeder, Blinov, and Hlavacek, *Methods Mol. Biol.* (2009)

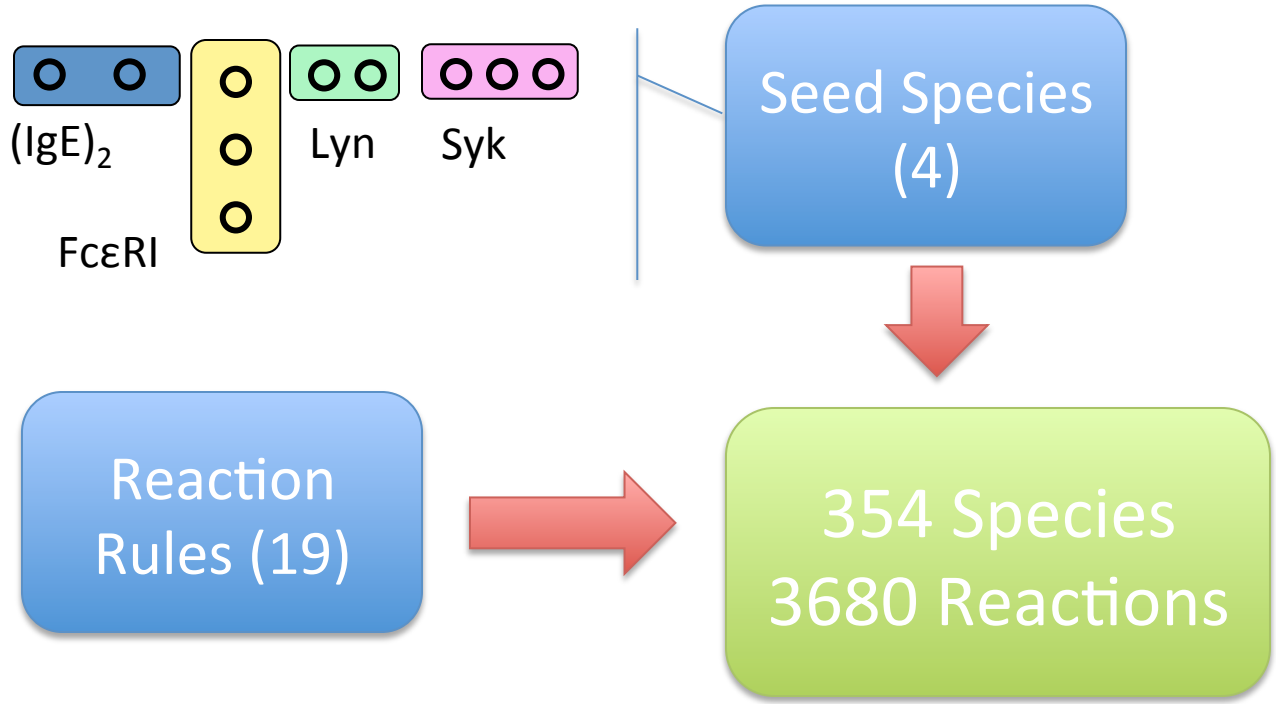
Automatic Network Generation

FcεRI Model



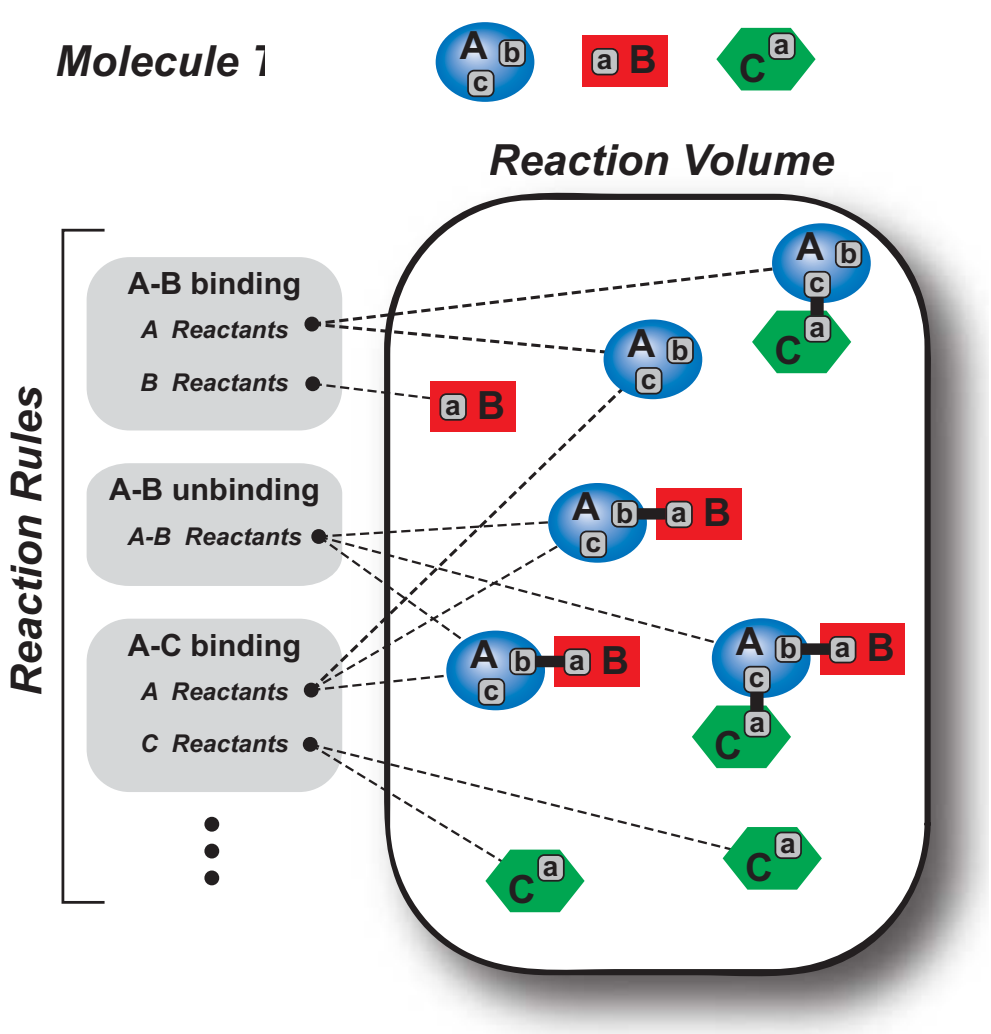
Automatic Network Generation

FcεRI Model



NFSIM*

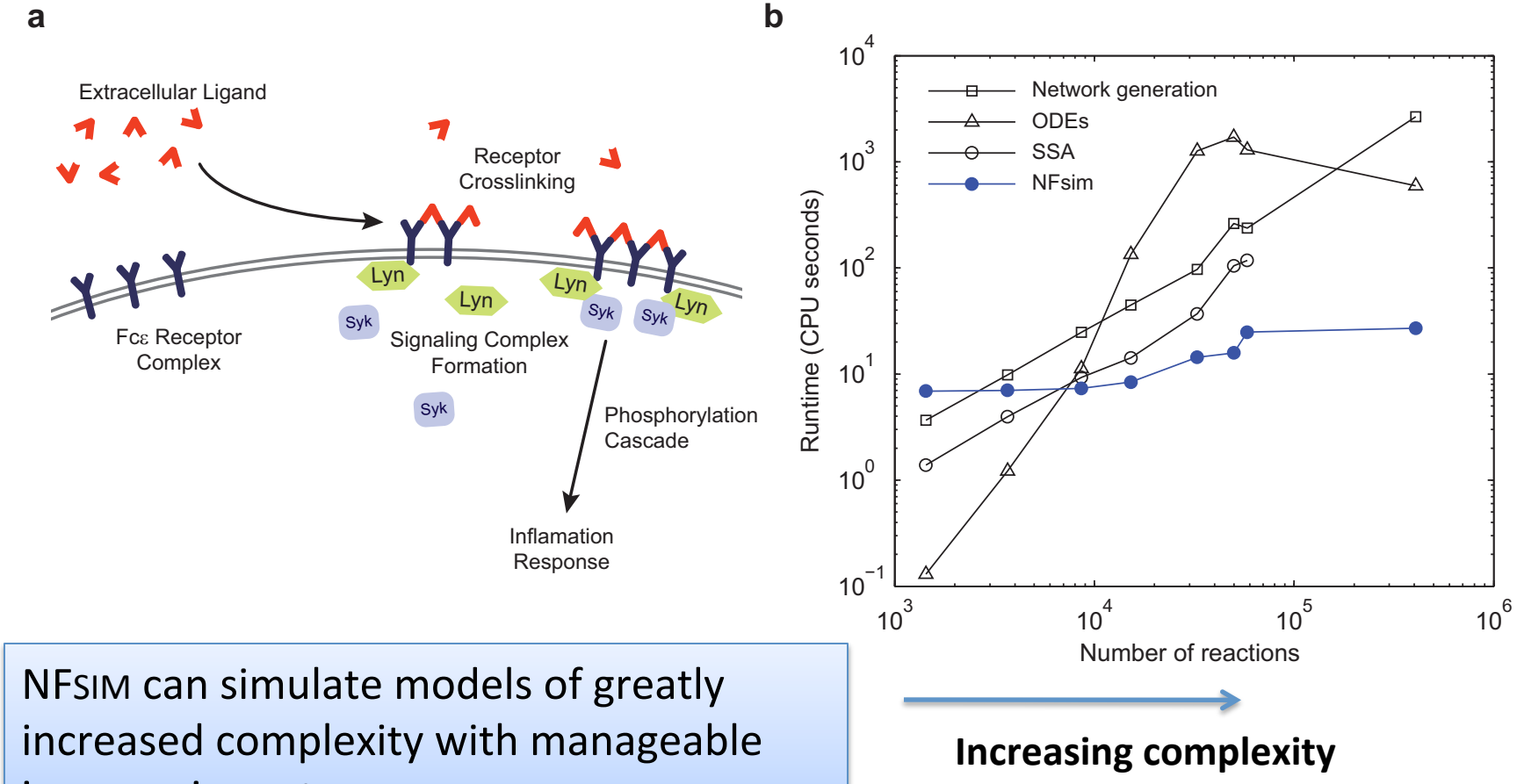
Network-Free Stochastic Simulator



Sneddon et al. (2011) *Nat. Methods*, **8**, 177

http://emonet.biology.yale.edu/nfsim/

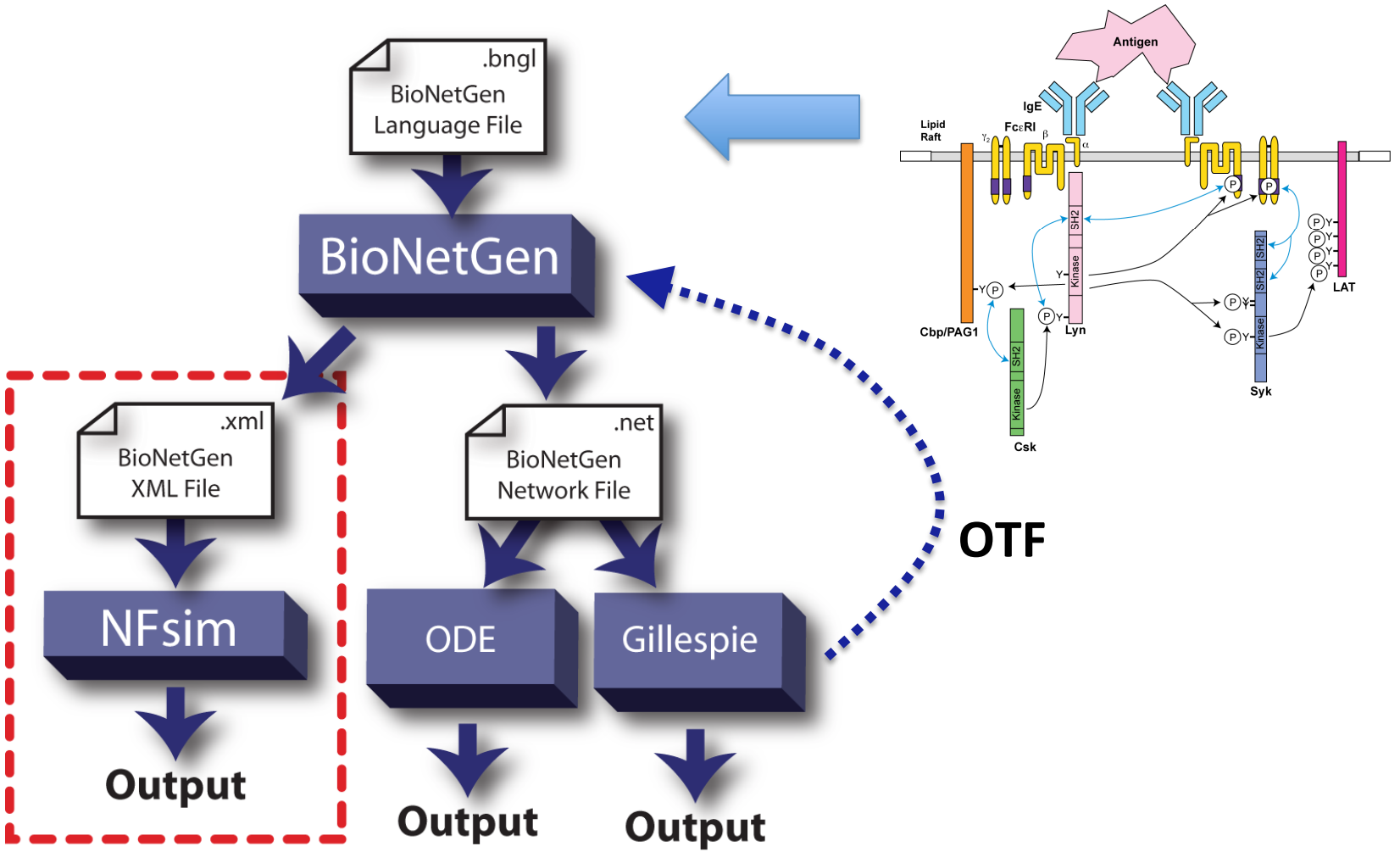
FcεRI signaling models



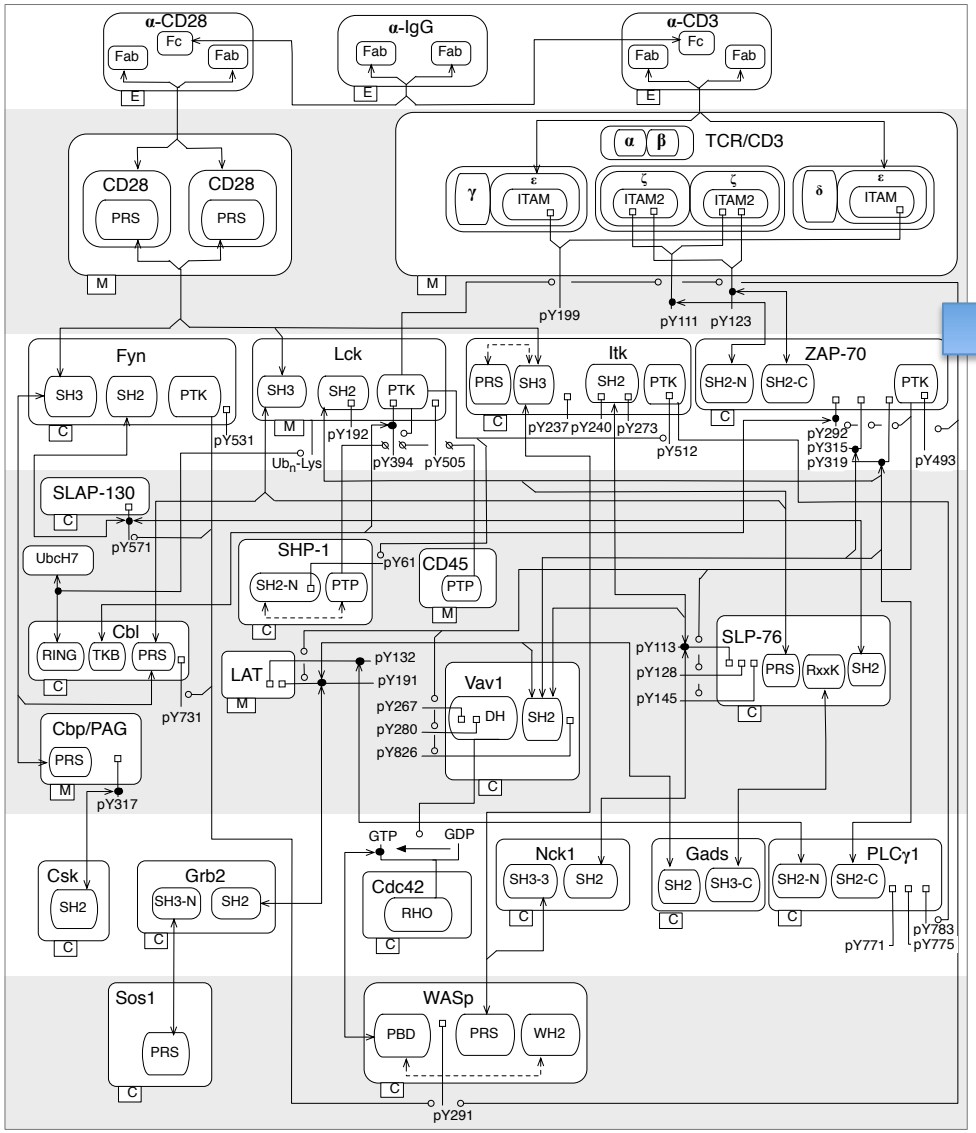
NFSIM can simulate models of greatly increased complexity with manageable increase in cost.

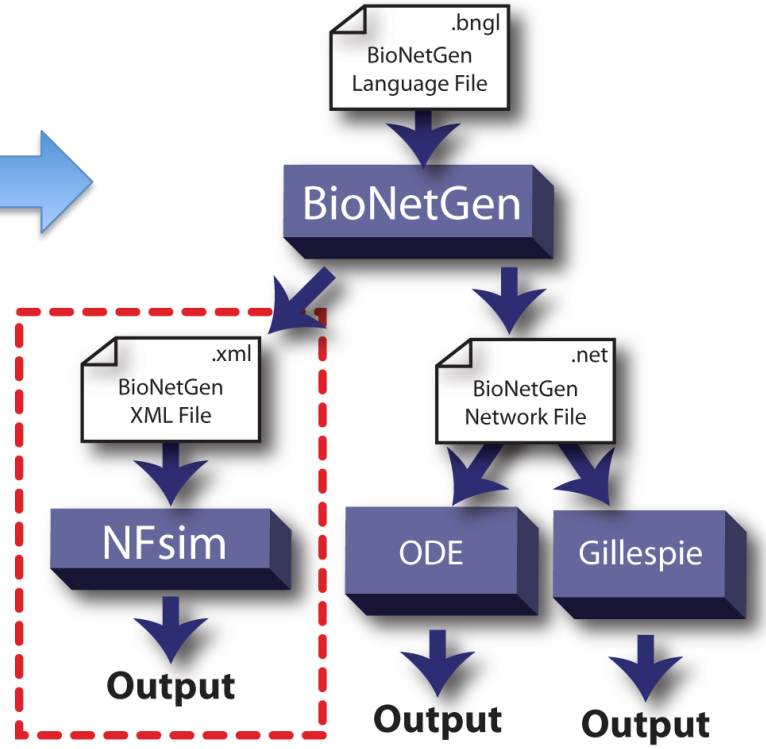
Increasing complexity

Integration with BioNetGen



Large Scale TCR Signaling Model



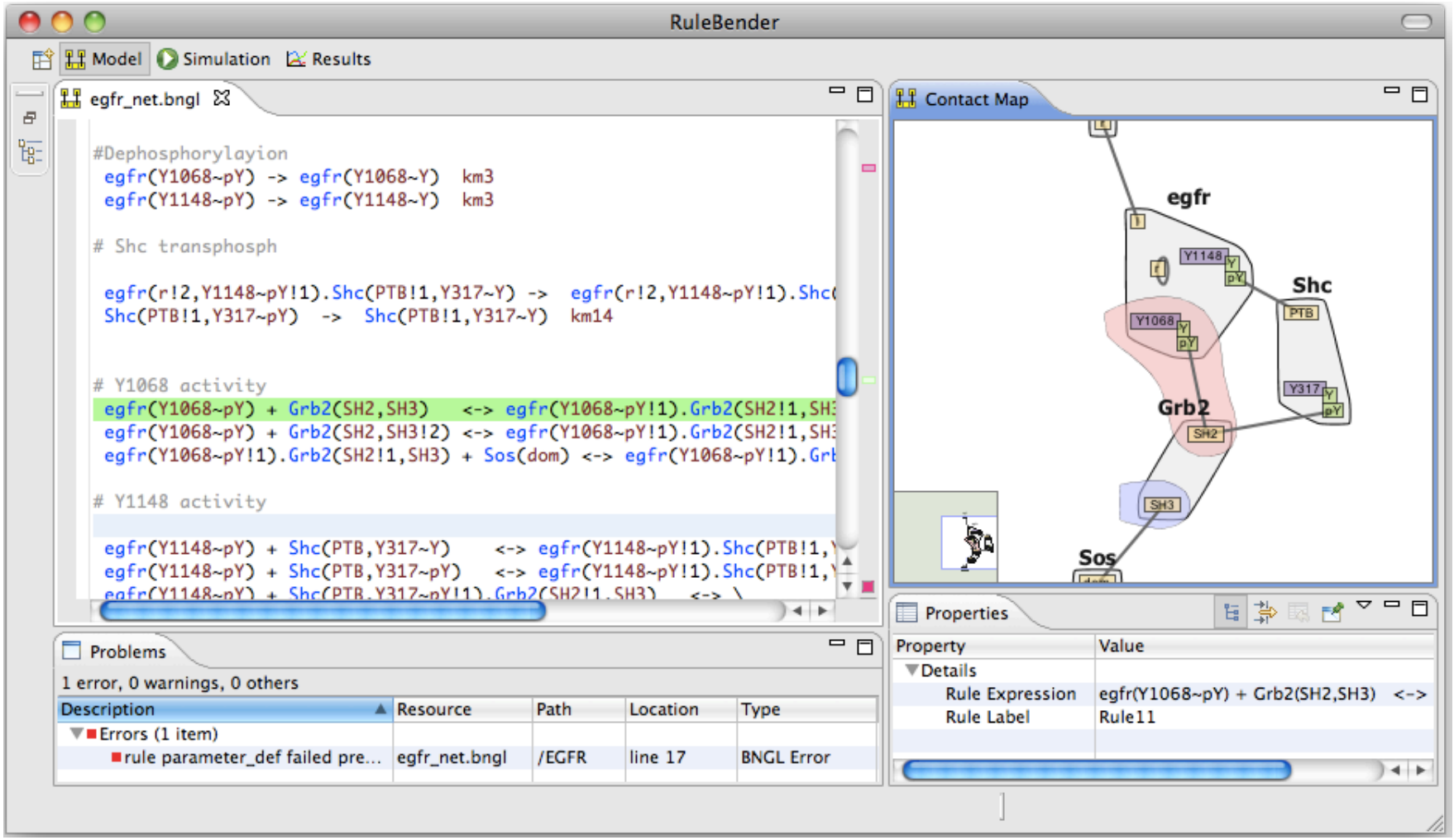


Hu, Chylek, and Hlavacek, unpublished.

RuleBender

Built in Eclipse RCP

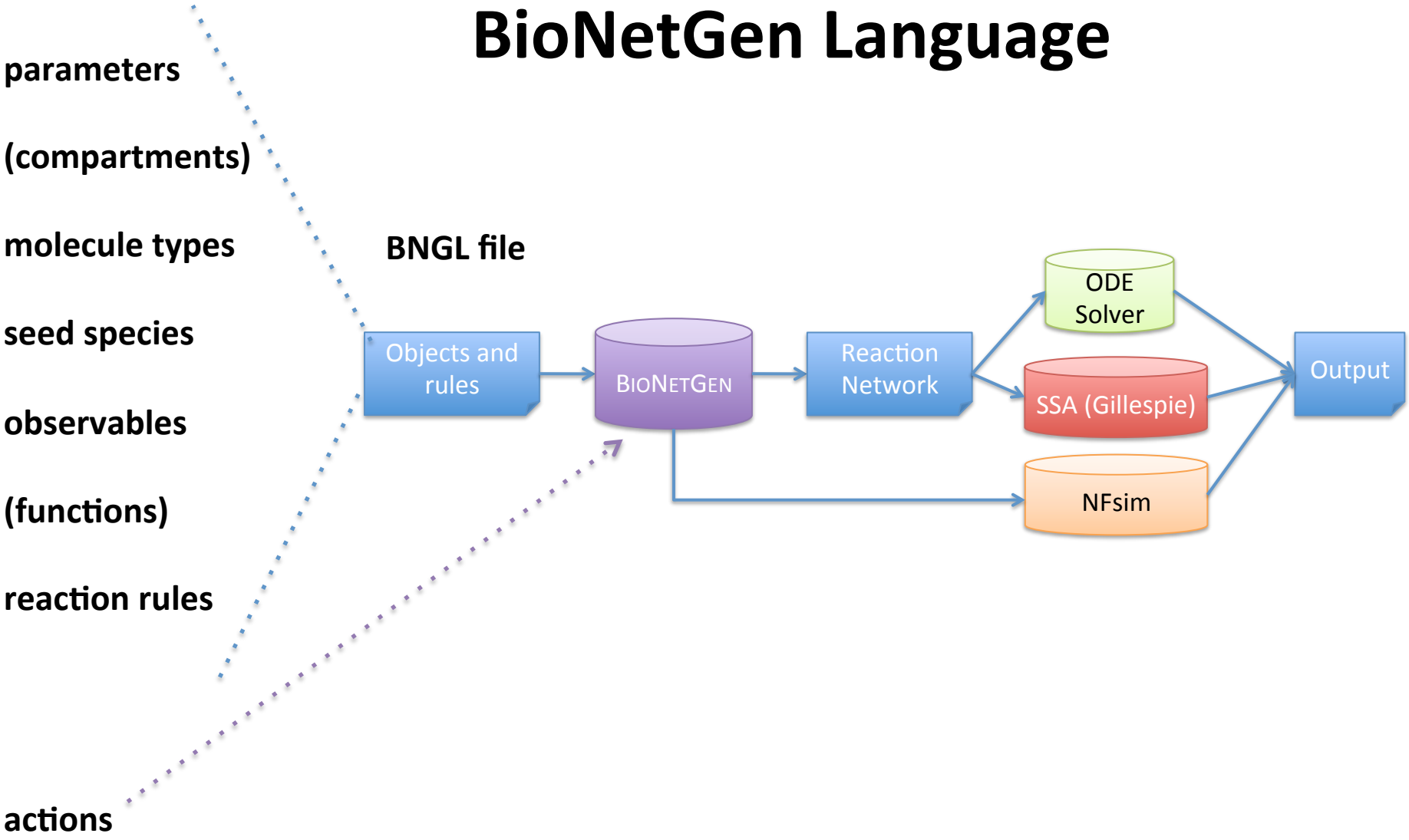
http://rulebender.org



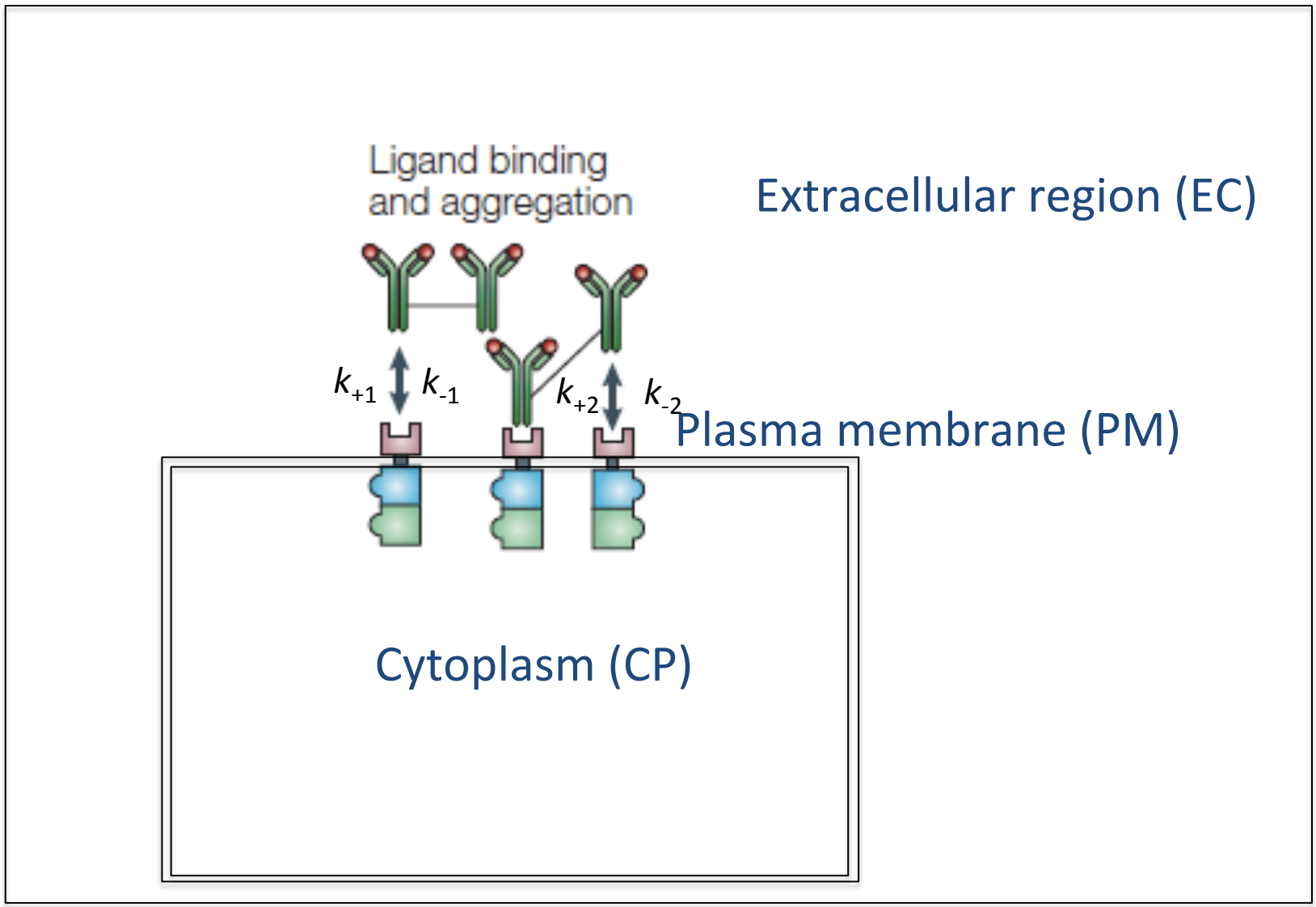
Xu et al. *Bioinformatics* (2011); Smith et al. *BioVis12 (Best Paper)*; *BMC Bioinformatics* (2012)

HANDS-ON TUTORIAL

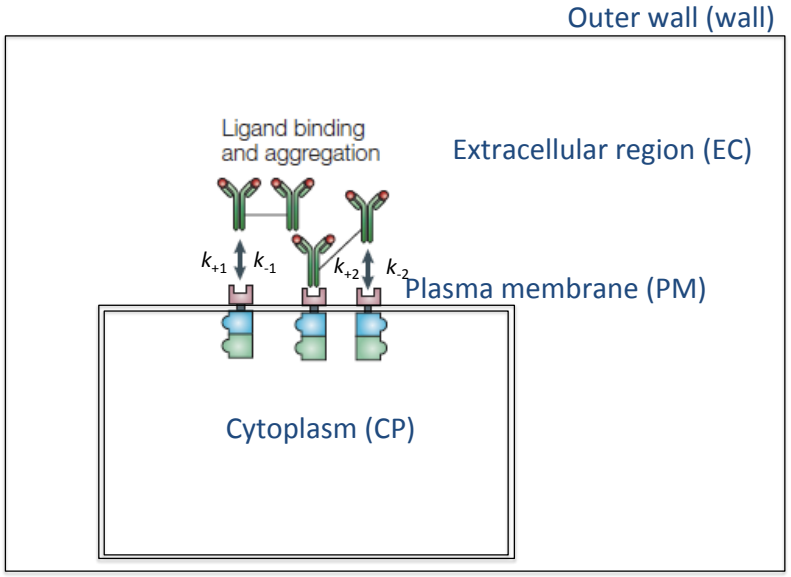
Technical Overview of the BioNetGen Language



Dimerization Model



Compartment Specification




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

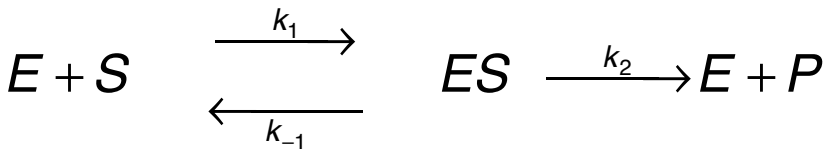
Volume of surface compartment = Area*thickness

thickness = 10 nm = 0.01 µm

BACKUP EXAMPLE

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

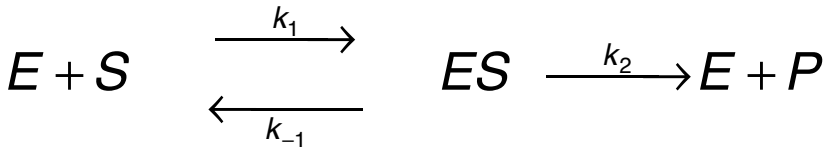
functions

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

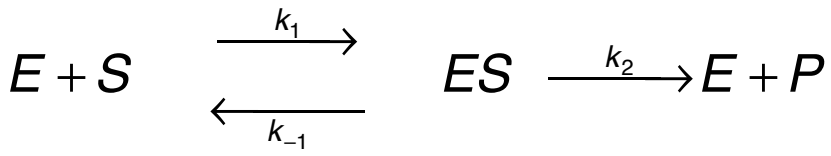
seed species

observables

functions

reaction rules

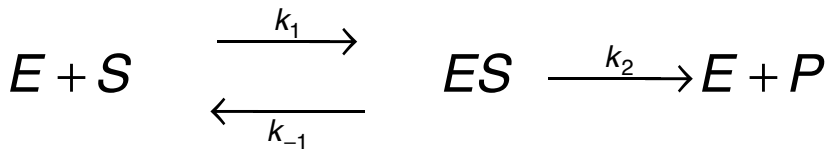
actions



```
begin parameters
  # Avogadro's number- scaled for umol
  NA 6.02e23/1e6
  # Cell volume
  V 1e-12 # liters - typical for eukaryote
  # 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

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

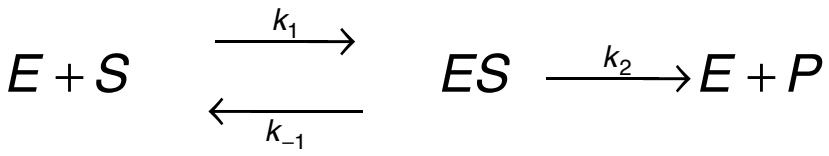
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

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

seed species

observables

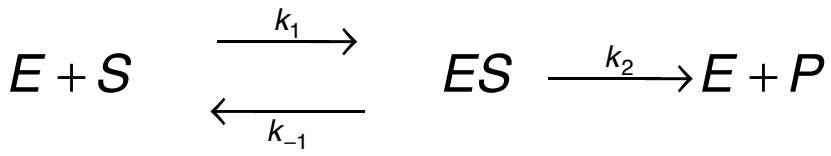
functions

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

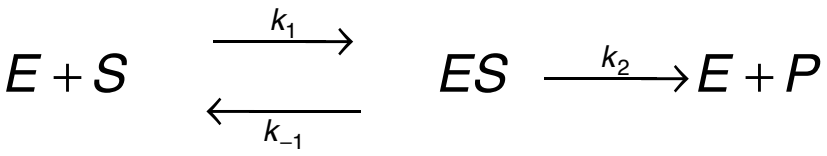
functions

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

functions

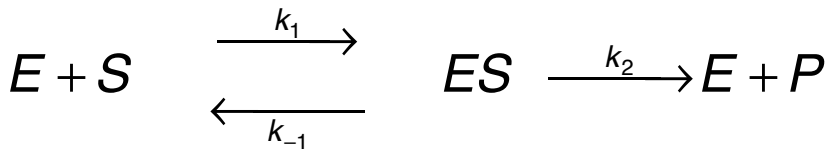
```
begin seed species
  E(s)    E0
  S(Y~0)  S0
end seed species
```

reaction rules

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

functions

reaction rules

actions

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.

Example 1: MM Mechanism

parameters

molecule types

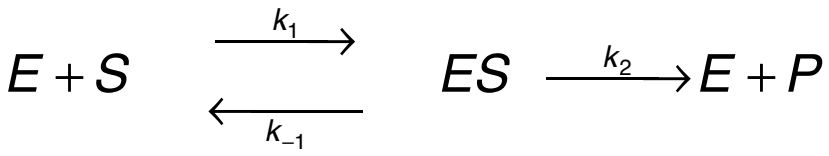
seed species

observables

functions

reaction rules

actions




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

Example 1: MM Mechanism

parameters

molecule types

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_2} E + P$$

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

observable SU S(Y~0)

matches S(Y~0)

species S(Y~0)

not S(Y~0)

E(s!1).S(Y~0!1)

reaction rules

SU = sum of concentration of matches = [S(Y~0)]

actions

Example 1: MM Mechanism

parameters

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \xrightarrow{k_2} E + P$$

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

```
observable ES   E(s!1).S(Y!1)
```

matches

species

```
E(s!1).S(Y~0!1)
```

functions

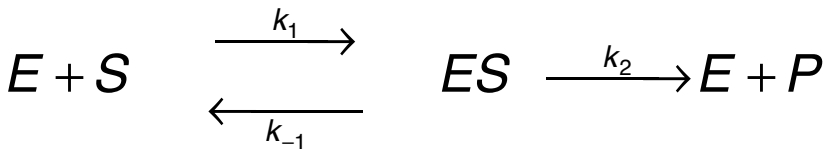
reaction rules

ES = sum of concentration of matches = [E(s!1).S(Y~0!1)]

actions

Example 1: MM Mechanism

parameters



molecule types

seed species

observables

functions

reaction rules

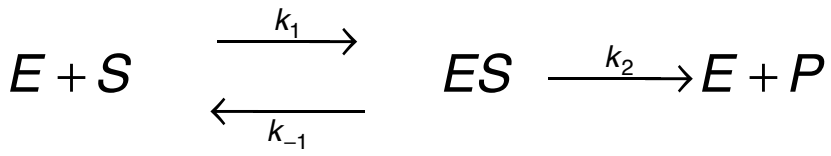
actions

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

observables

functions

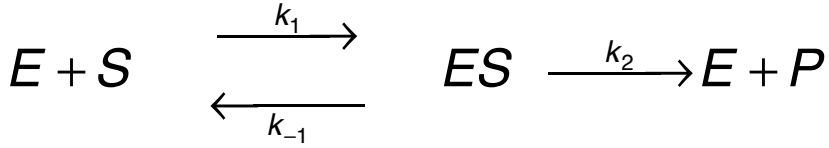
reaction rules

actions

```
begin reaction rules

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

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

end reaction rules
```

Example 1: MM Mechanism

parameters



molecule types

<u>actions</u>– 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