

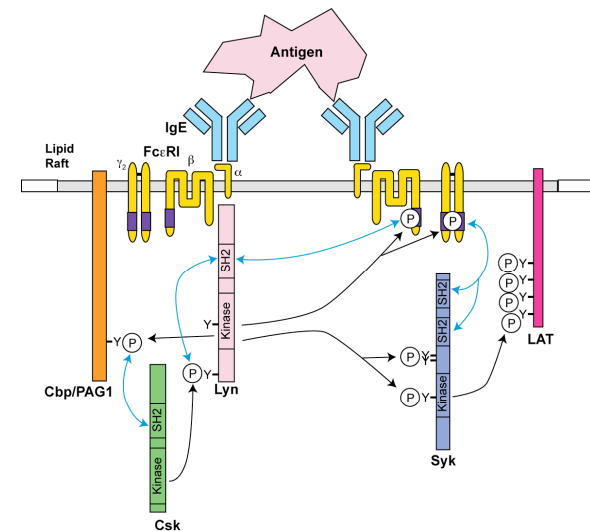
Navigating the Subway Map of the Cell

Jim Faeder

*Department of Computational Biology
University of Pittsburgh School of Medicine*

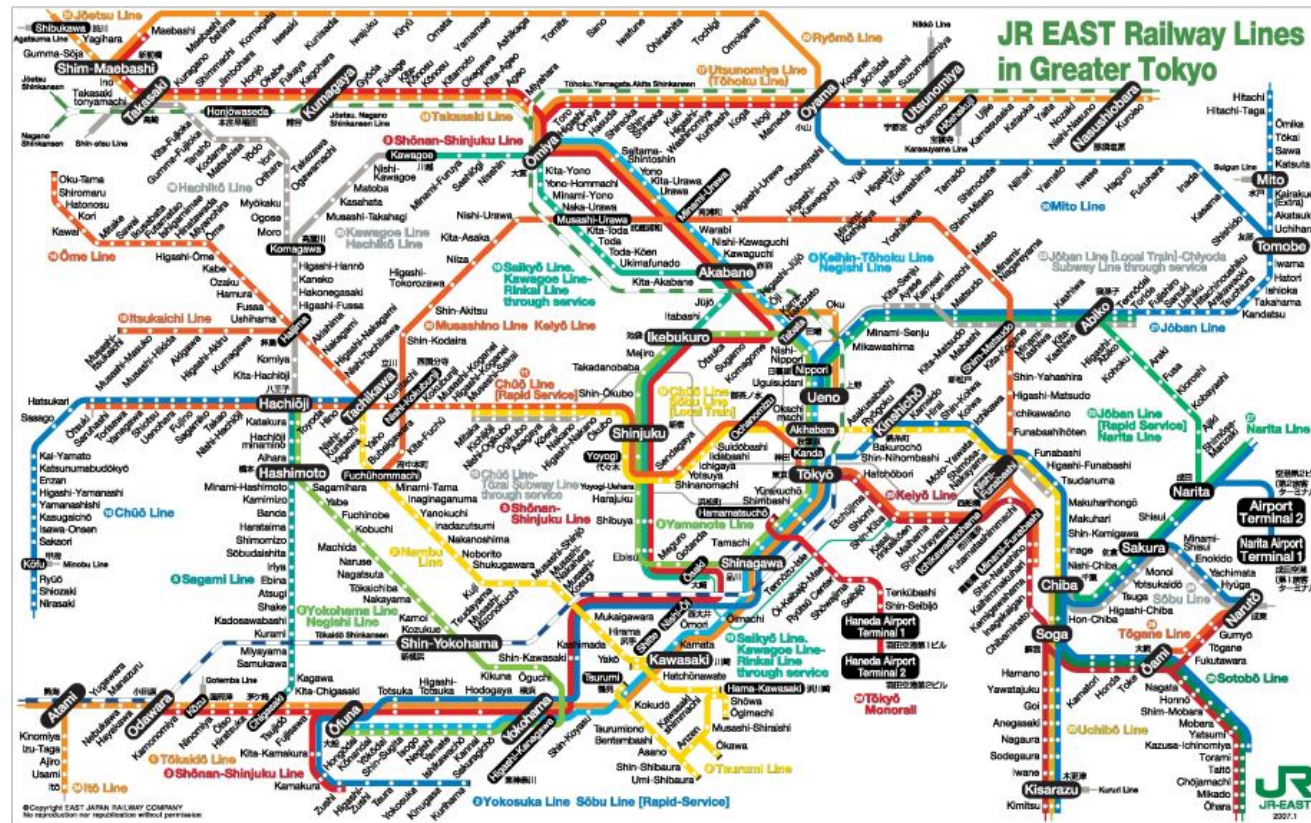
CMACS Expeditions Kickoff Meeting
October 31, 2009

faeder@pitt.edu
<http://ccbb.pitt.edu/faeder>



Department of
Computational Biology

Not really a map talk...



- Tokaido Line
- Yokosuka Line Sōbu Line [Rapid Service]
- Shōnan-Shinjuku Line
- Keihin-Tōhoku Line Negishi Line
- Sagami Line
- Yokohama Line Negishi Line
- Nambu Line
- Tsurumi Line
- Yamanote Line
- Chūō Line
- Chūō Line [Rapid Service]
- Chūō Line Tōzai Subway Line through service
- Ōme Line
- Itsukaichi Line
- Hachikō Line
- Takasaki Line
- Saikyō Line, Kawagoe Line-Rinkai Line through service
- Kawagoe Line Hachikō Line
- Jōban Line
- Jōban Line [Rapid Service] Narita Line
- Jōban Line [Local Train]-Chiyoda Subway Line through service
- Sōbu Line
- Uchibō Line
- Sotobō Line
- Narita Line
- Tōgane Line
- Kelyō Line
- Musashino Line Kelyō Line
- Jōetsu Line
- Ryōmō Line
- Mito Line
- Itō Line
- Tokyo Monorail

...it's more about trains.



(trains that aren't even really traveling on the map)

because some places are hard to get to with existing service...



A-train

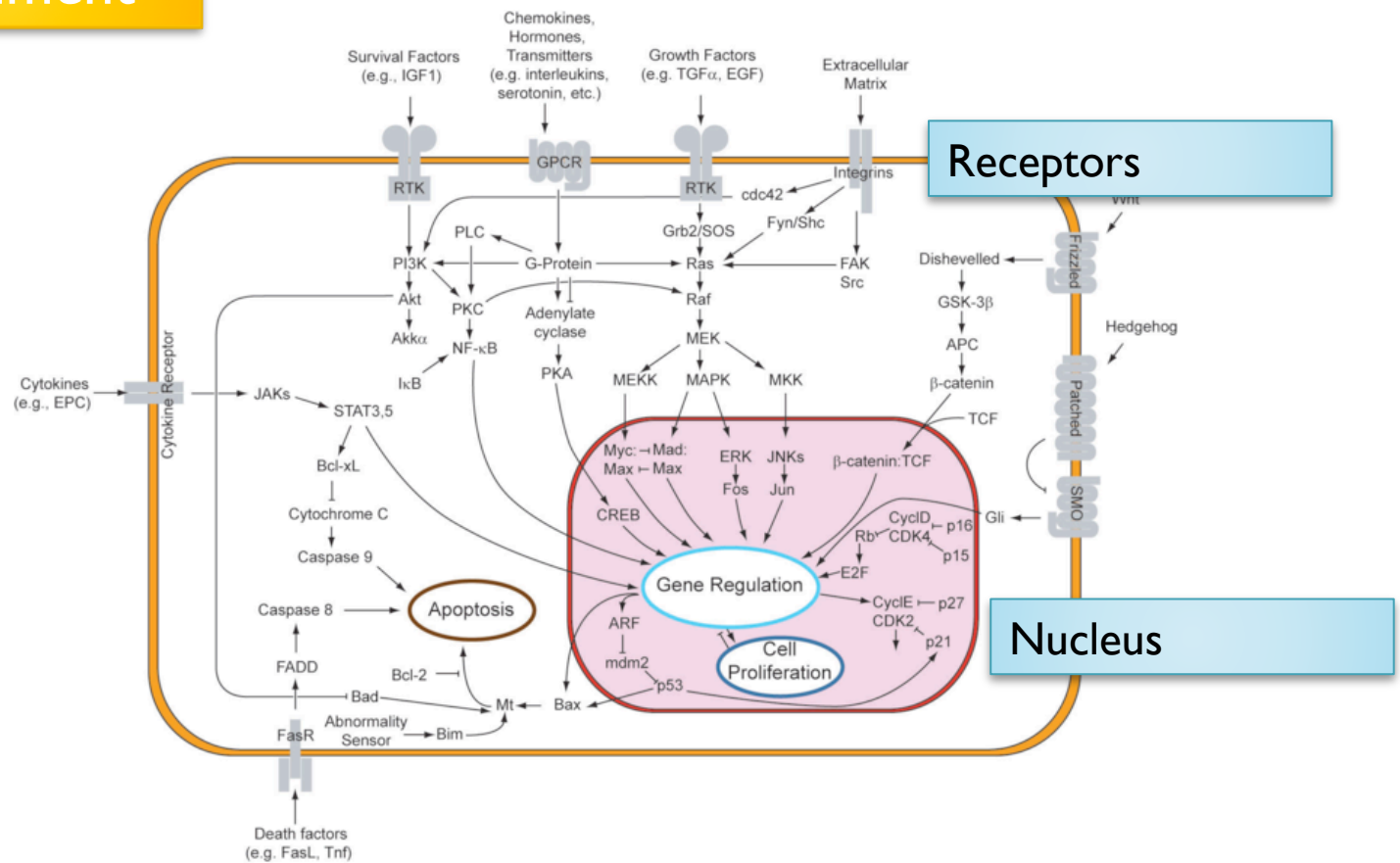
so new construction is required.



How Cells Process Information

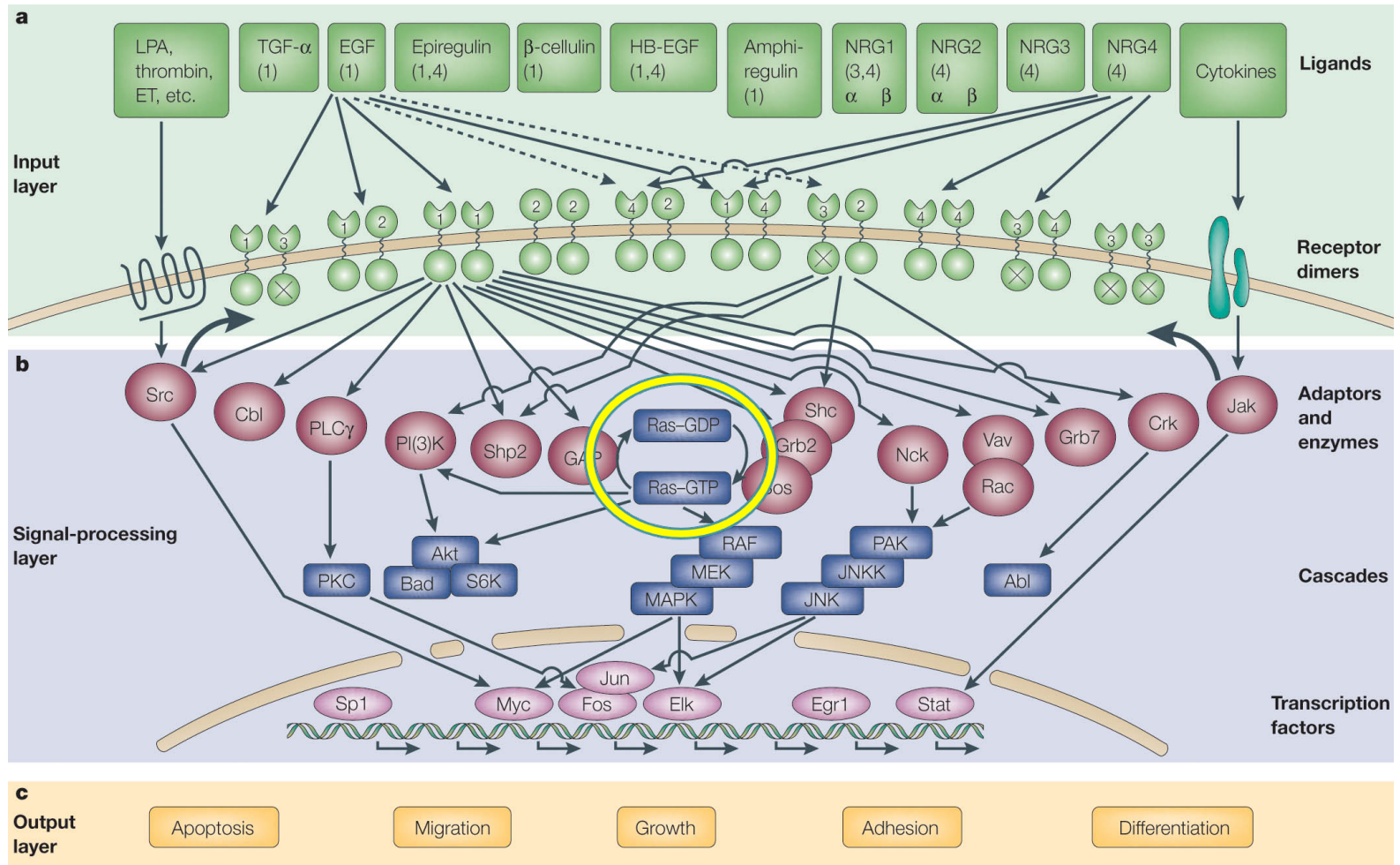
Environment

Hormones, growth factors, etc.



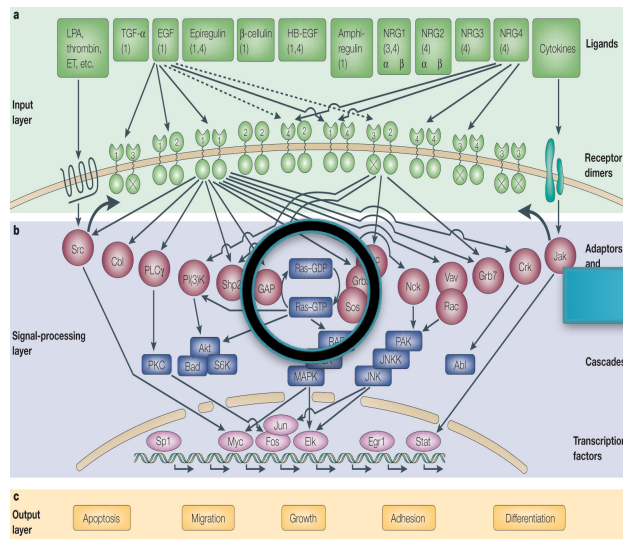
http://en.wikipedia.org/wiki/Cell_signaling

Architecture of a signaling network



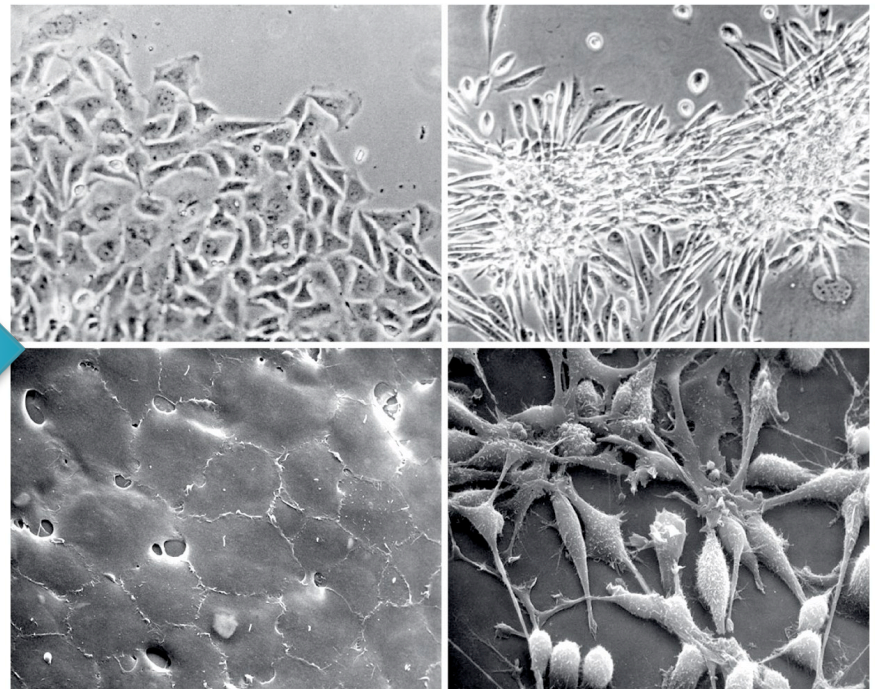
Yarden & Sliwkowski, *Nature Rev. Mol. Cell Biol.* **02**: 127-137 (2001).

Mutation of Ras Can Produce a Tumor Cell



Normal

Transformed

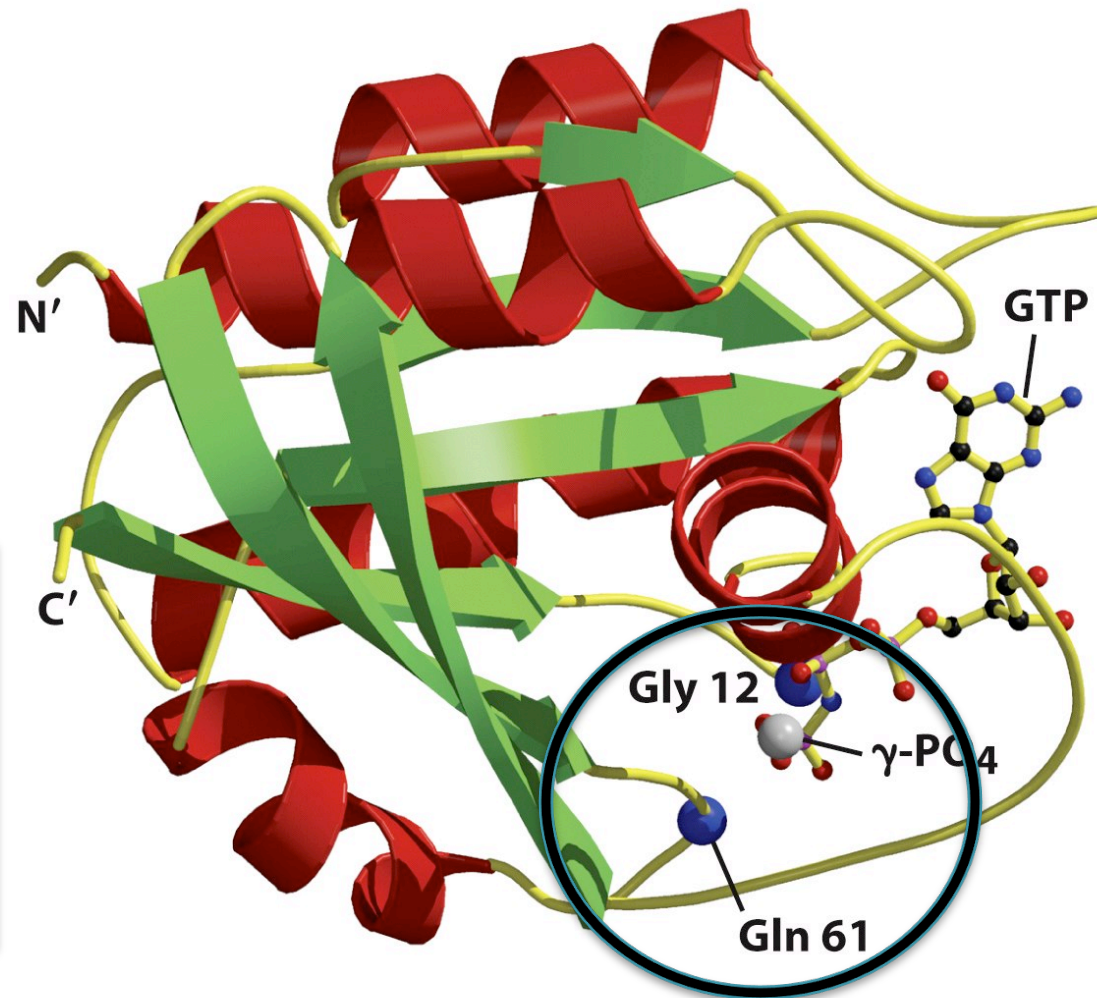


Ras mutations in cancer

Ras

>20% human tumors carry Ras point mutations.

>90% in *pancreatic cancer*.



Modularity of Signaling Proteins

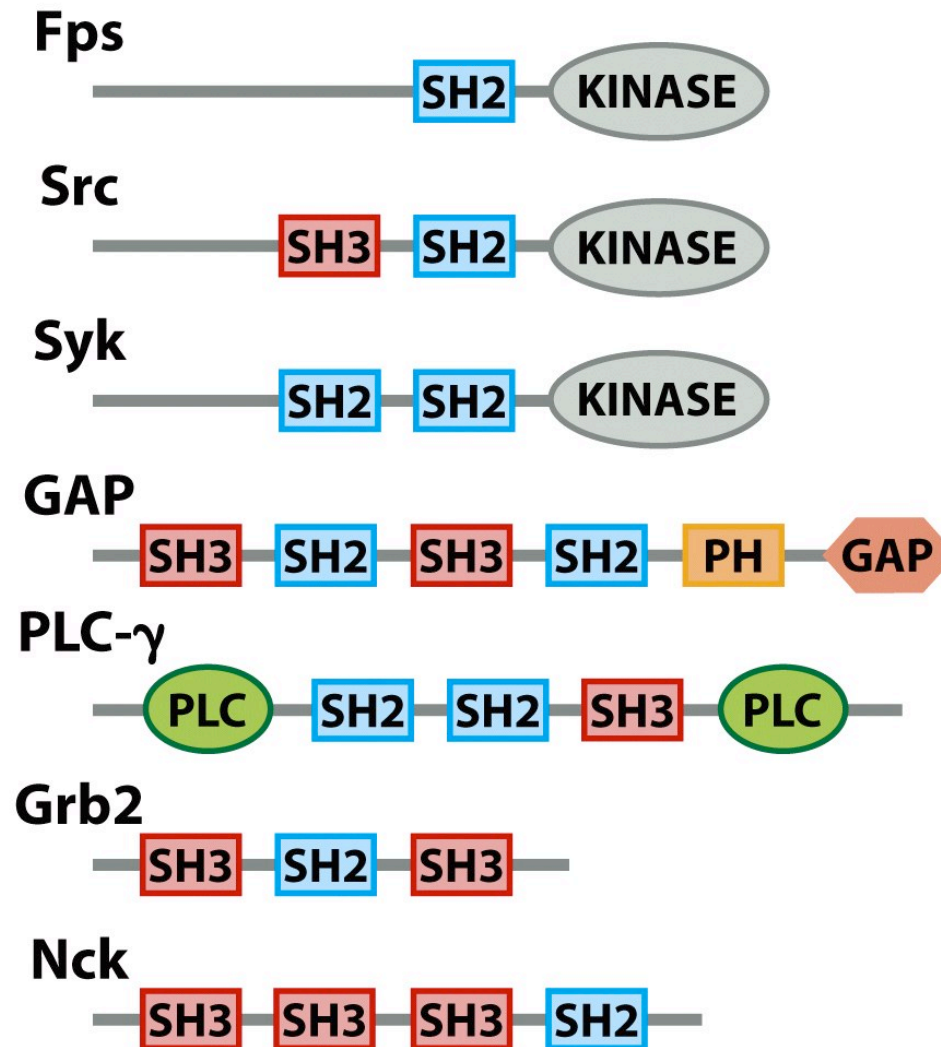
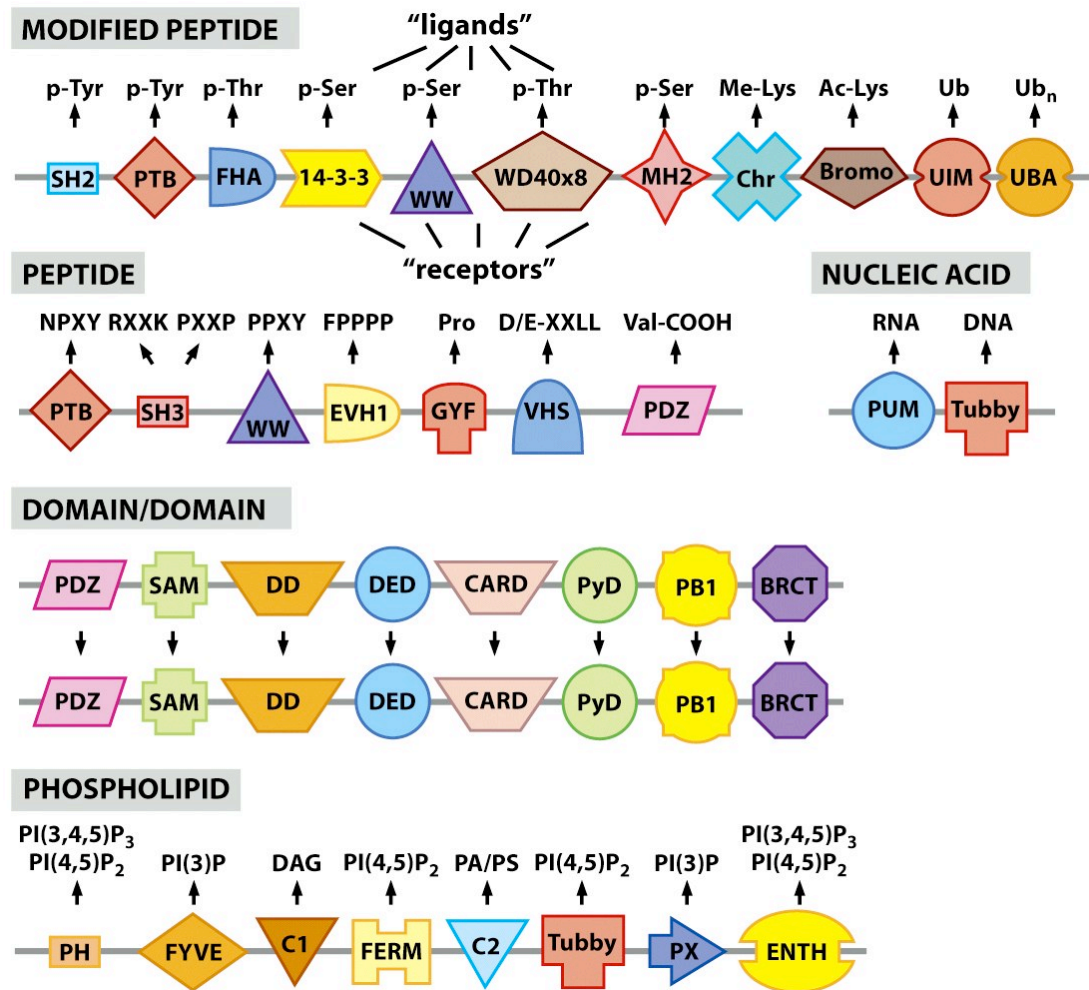


Figure 6.10a *The Biology of Cancer* (© Garland Science 2007)

Modularity produces complex wiring



Complexity of Receptor Complexes

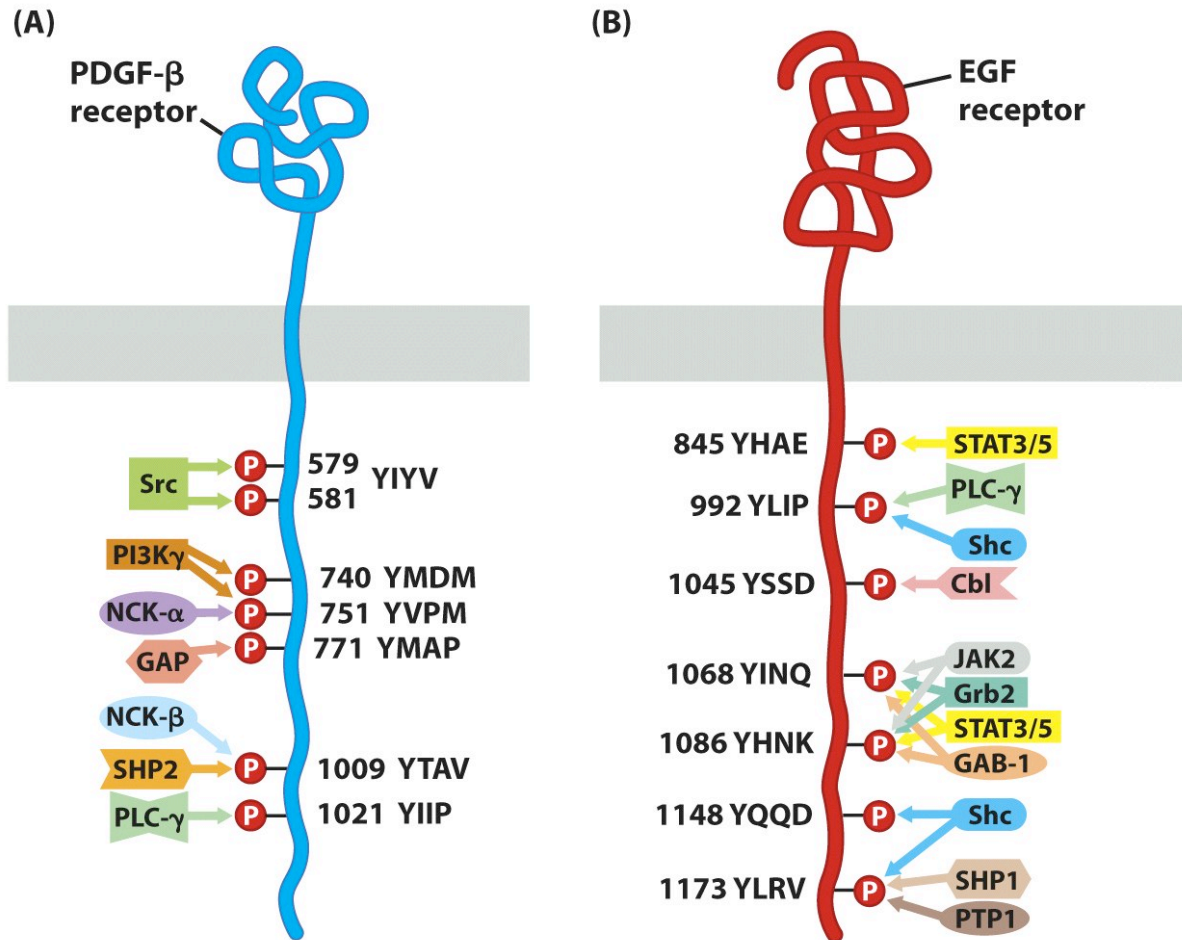
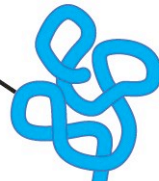


Figure 6.9 *The Biology of Cancer* (© Garland Science 2007)

The “curse” of complexity

(A)

PDGF- β receptor



Number of States

3

3

3

4

3

4

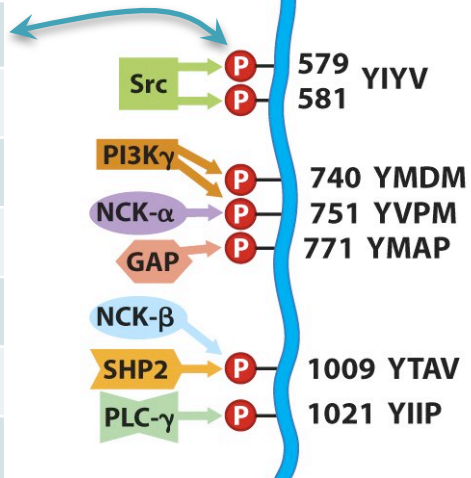
3

3,888

Monomers

7,560,216

Dimers



(B)

EGF receptor



Number of States

3

4

3

6

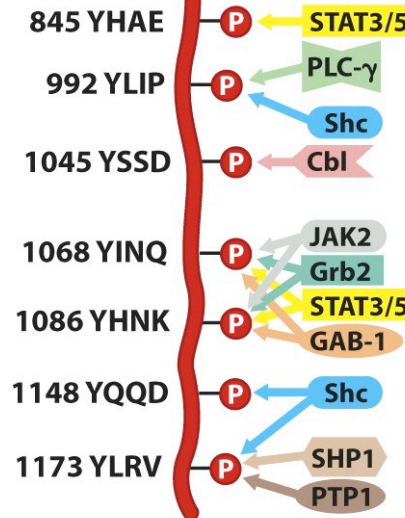
6

3

5

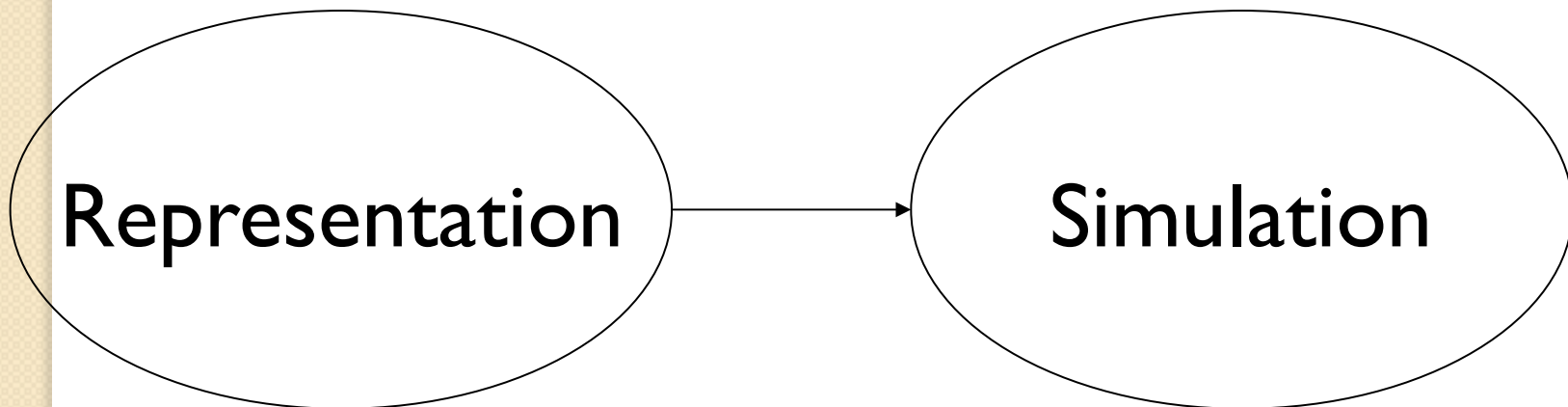
19,440

188,966,520



Modeling cell signaling

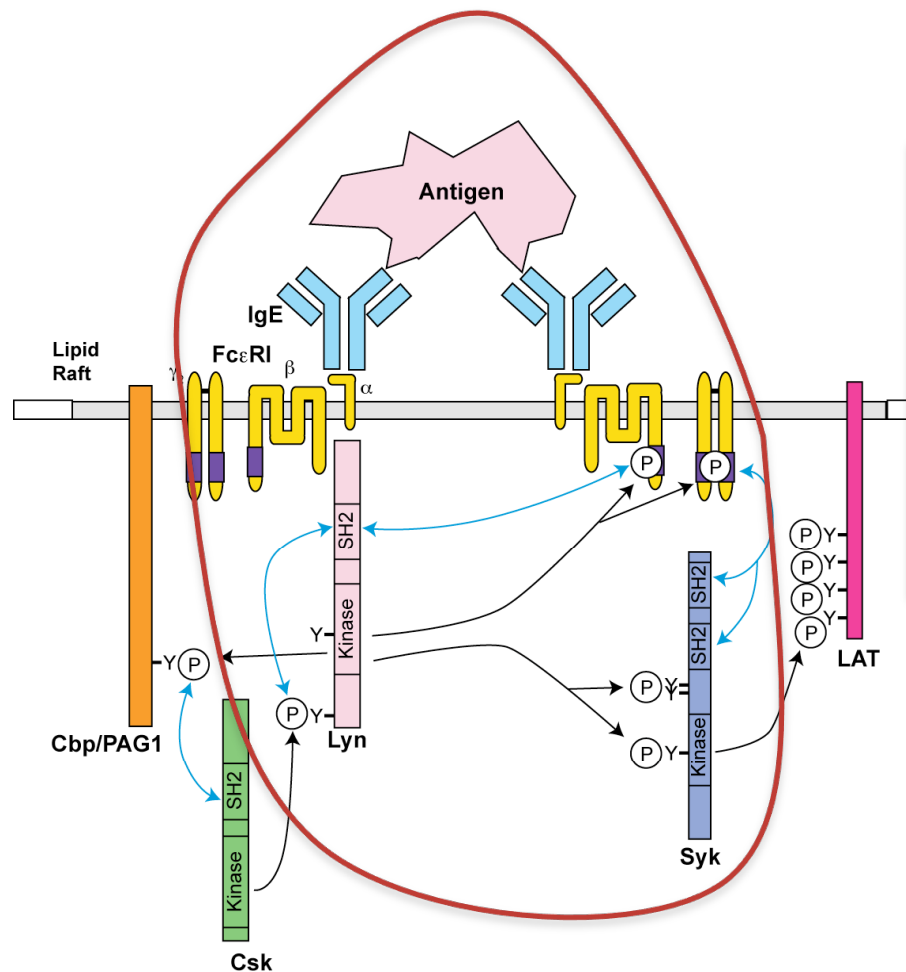
AIM: Model the biochemical machinery by which cells process information (and respond to it).



BIONETGEN Language
kappa
etc.

ODE, PDE
Stochastic Simulation Algorithm
Kinetic Monte Carlo
Brownian dynamics

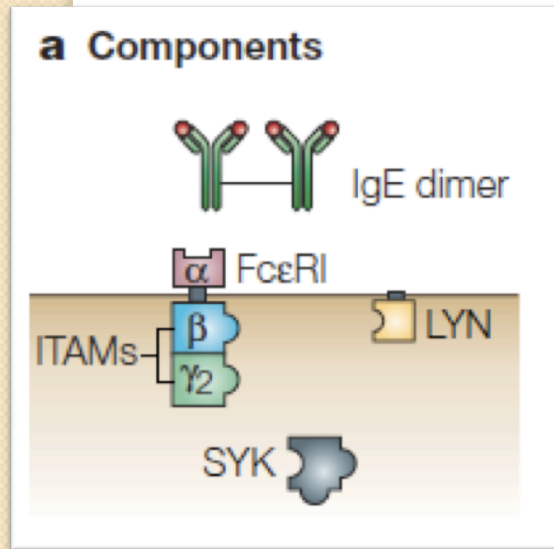
Syk activation model



Key variables

- ligand properties
- protein expression levels
- multiple Lyn-FcεRI interactions
- transphosphorylation

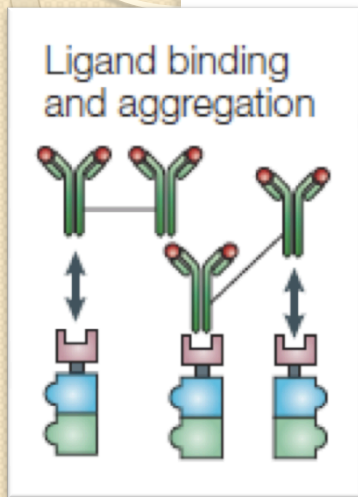
Defining Molecules



BIONETGEN Language

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

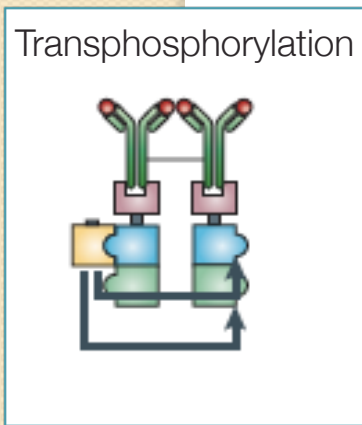
Defining Interaction Rules



BIONETGEN Language

$\text{IgE}(a, \underline{a}) + \text{FceRI}(\underline{a}) \leftrightarrow \text{IgE}(a, \underline{a!1}) \cdot \text{FceRI}(\underline{a!1})$
...

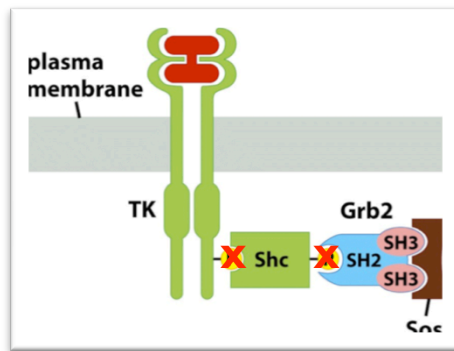
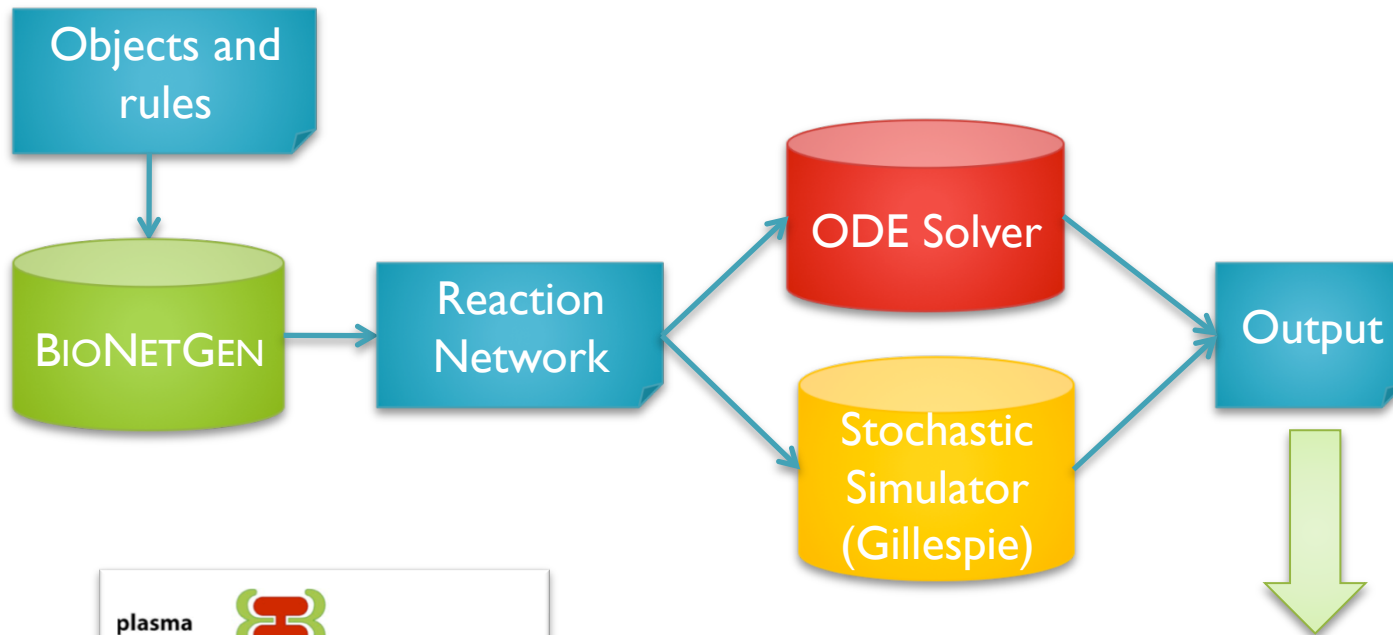
binding and dissociation



$\text{Lyn}(U!1) \cdot \text{FceRI}(b!1) \cdot \text{FceRI}(b \sim U) \rightarrow \backslash$
 $\text{Lyn}(U!1) \cdot \text{FceRI}(b!1) \cdot \text{FceRI}(b \sim P)$

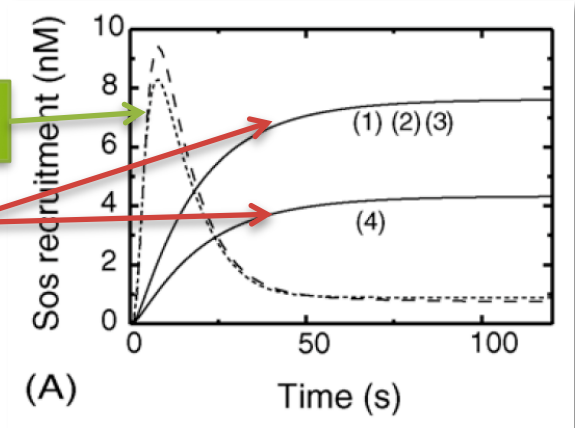
component state change

Rule-based modeling protocol



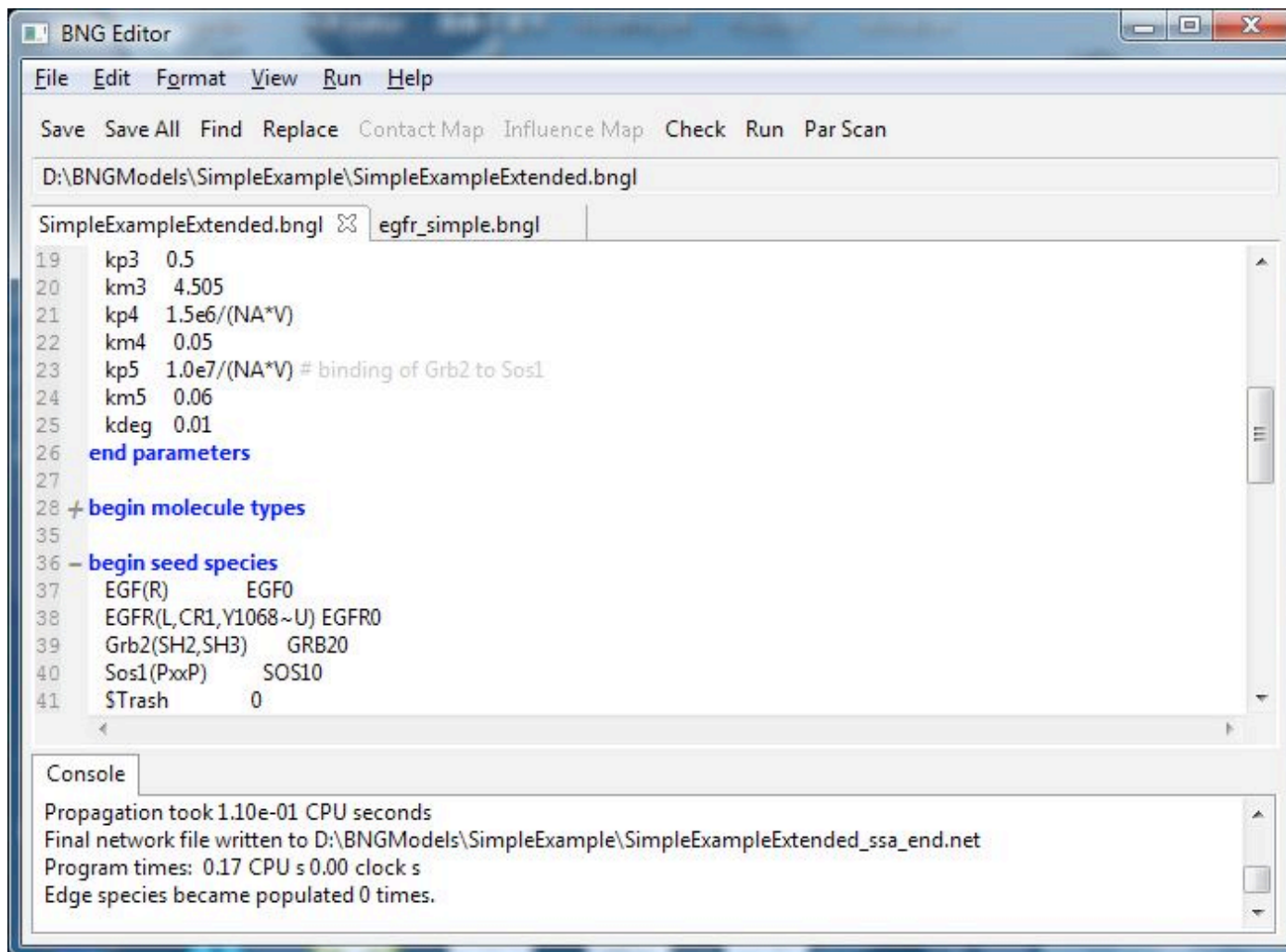
“Normal Cell”

“Mutants”



<http://bionetgen.org>

BIONETGEN Editor - BiNGE



The screenshot shows the BNG Editor window with the following content:

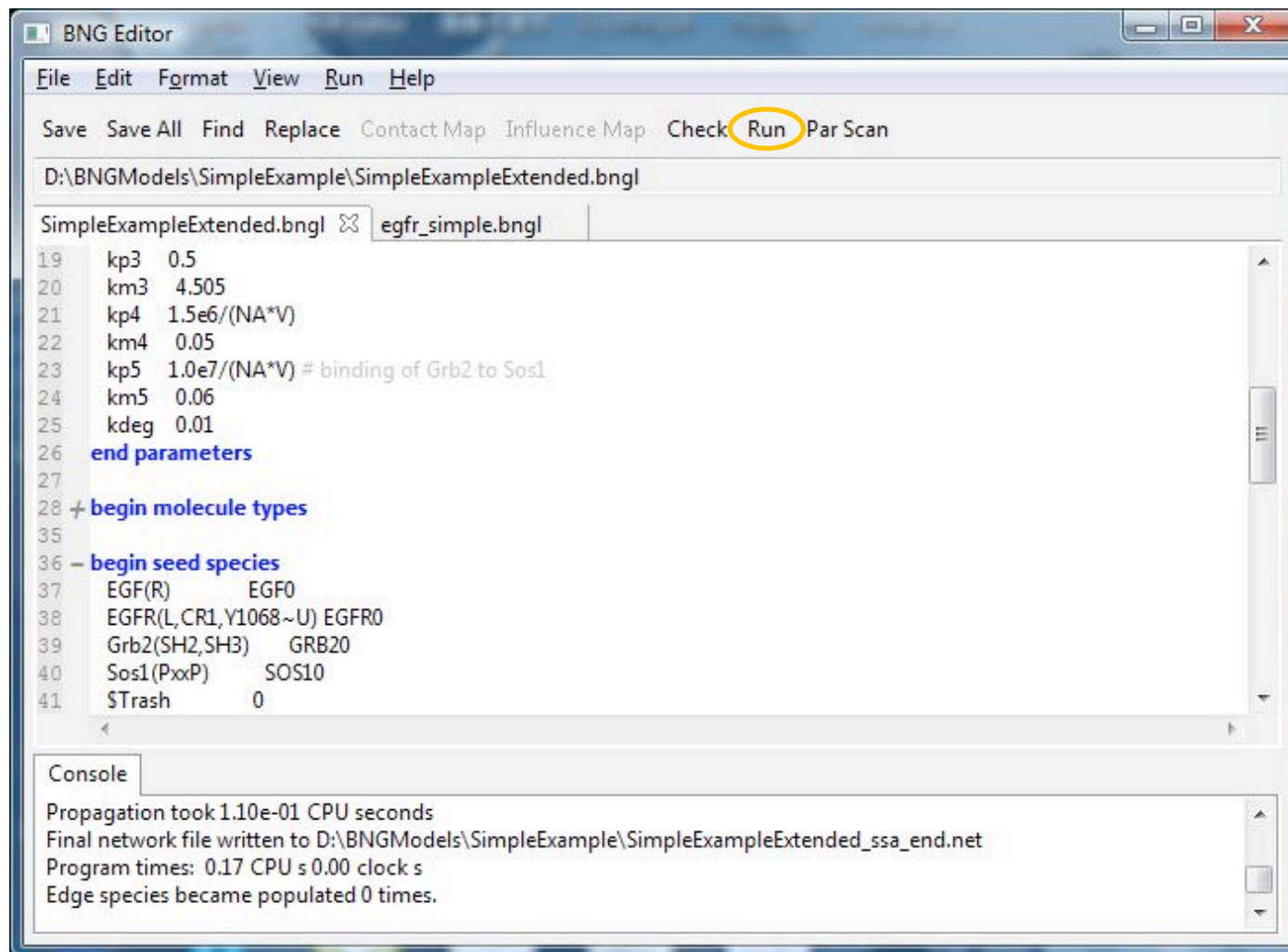
```
File Edit Format View Run Help
Save Save All Find Replace Contact Map Influence Map Check Run Par Scan
D:\BNGModels\SimpleExample\SimpleExampleExtended.bngl
SimpleExampleExtended.bngl egfr_simple.bngl
19 kp3 0.5
20 km3 4.505
21 kp4 1.5e6/(NA*V)
22 km4 0.05
23 kp5 1.0e7/(NA*V) # binding of Grb2 to Sos1
24 km5 0.06
25 kdeg 0.01
26 end parameters
27
28 + begin molecule types
35
36 - begin seed species
37 EGF(R) EGF0
38 EGFR(L,CRI,Y1068~U) EGFR0
39 Grb2(SH2,SH3) GRB20
40 Sos1(PxxP) SOS10
41 STrash 0
```

Console

```
Propagation took 1.10e-01 CPU seconds
Final network file written to D:\BNGModels\SimpleExample\SimpleExampleExtended_ssa_end.net
Program times: 0.17 CPU s 0.00 clock s
Edge species became populated 0 times.
```

Yao Sun and Liz Marai, U. Pitt Computer Science

BIONETGEN Editor - BiNGE



The screenshot shows the BNG Editor window with the following content:

File Edit Format View Run Help

Save Save All Find Replace Contact Map Influence Map Check **Run** Par Scan

D:\BNGModels\SimpleExample\SimpleExampleExtended.bngl

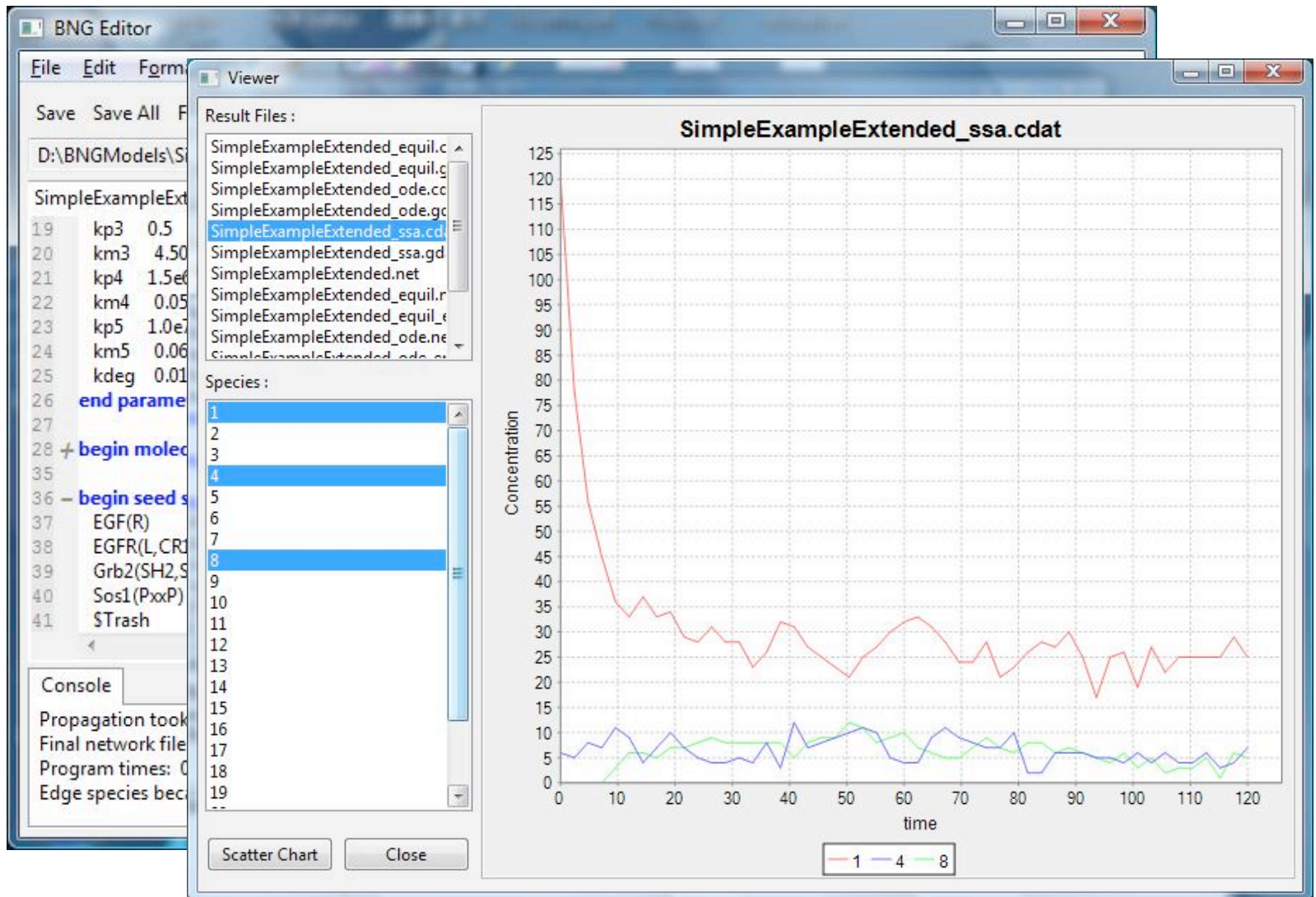
SimpleExampleExtended.bngl egfr_simple.bngl

```
19 kp3 0.5
20 km3 4.505
21 kp4 1.5e6/(NA*V)
22 km4 0.05
23 kp5 1.0e7/(NA*V) # binding of Grb2 to Sos1
24 km5 0.06
25 kdeg 0.01
26 end parameters
27
28 +begin molecule types
35
36 -begin seed species
37 EGF(R) EGF0
38 EGFR(L,CR1,Y1068~U) EGFR0
39 Grb2(SH2,SH3) GRB20
40 Sos1(PxxP) SOS10
41 $Trash 0
```

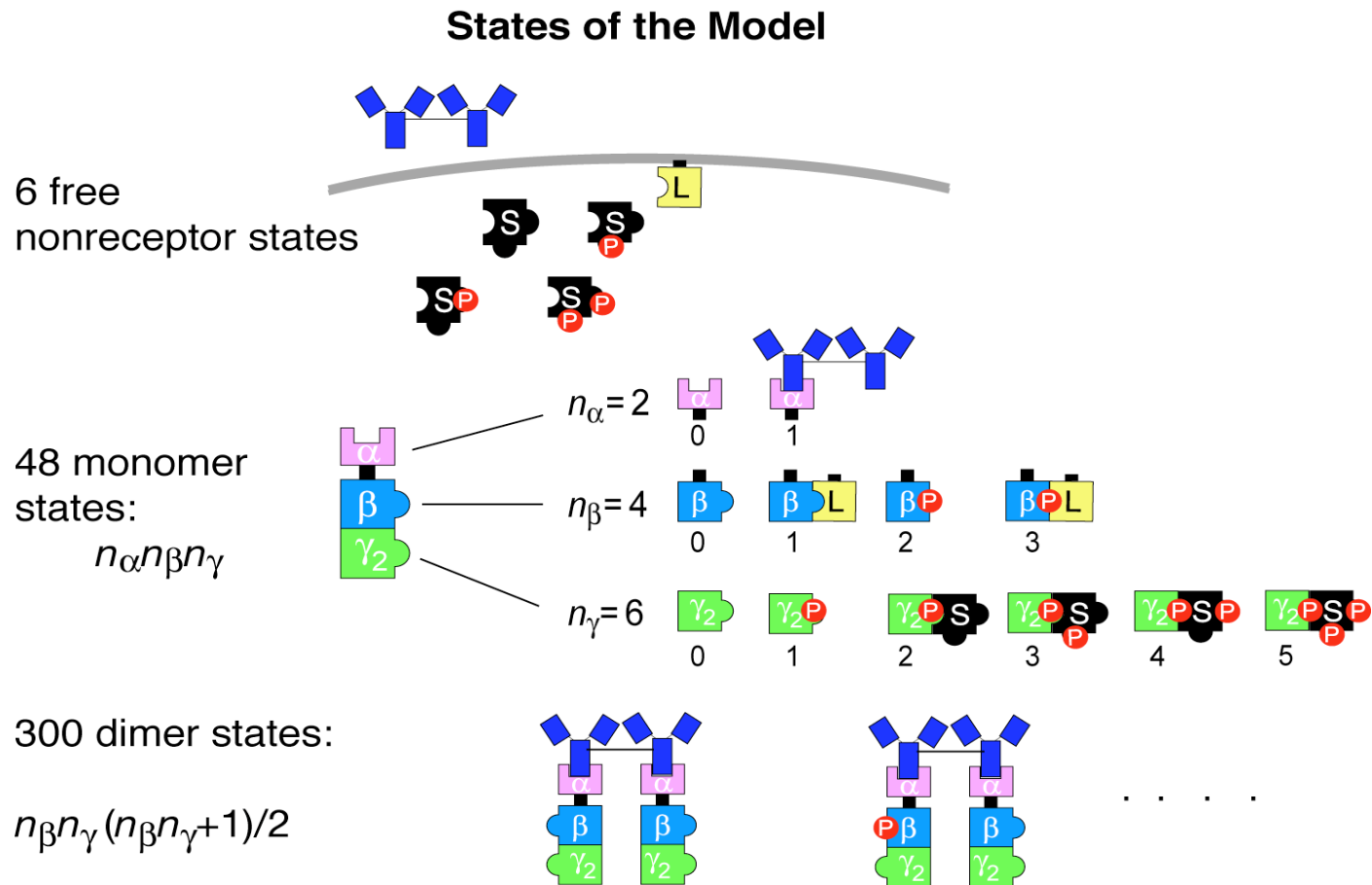
Console

Propagation took 1.10e-01 CPU seconds
Final network file written to D:\BNGModels\SimpleExample\SimpleExampleExtended_ssa_end.net
Program times: 0.17 CPU s 0.00 clock s
Edge species became populated 0 times.

BIONETGEN Editor - BiNGE



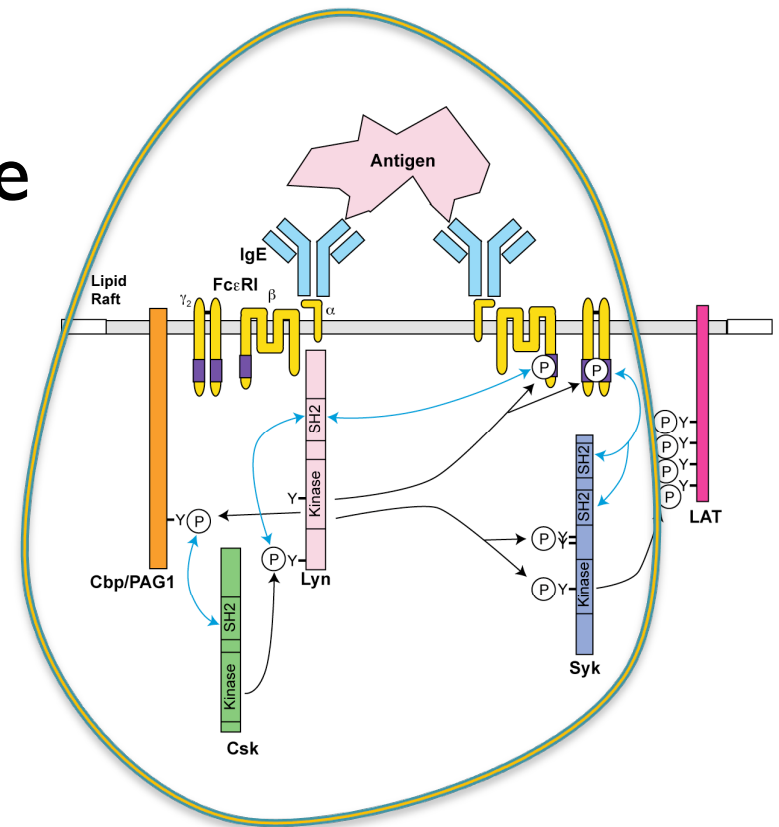
Enumeration of States, aka “Species”



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

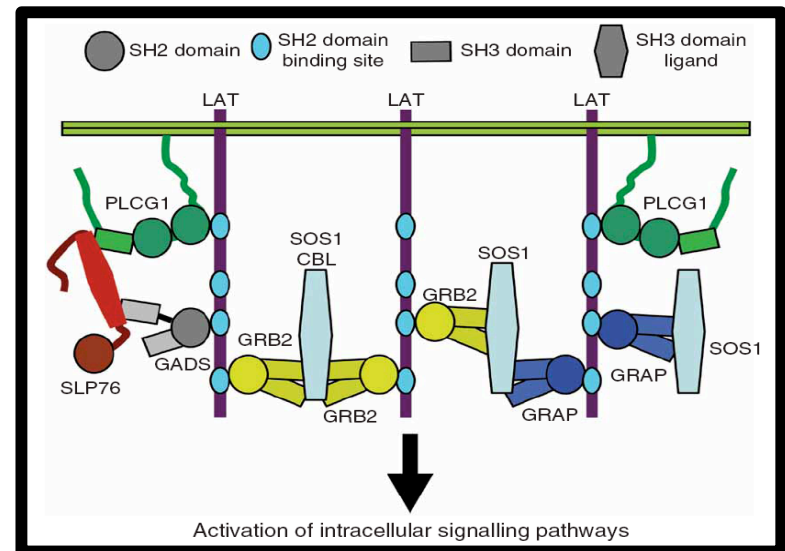
Limits of the network generation approach

- Extending model to include Lyn regulation results in >20,000 species.



Limits of the network generation approach

- Extending model to include Lyn regulation results in >20,000 species.
- LAT may form large oligomers under physiological conditions.

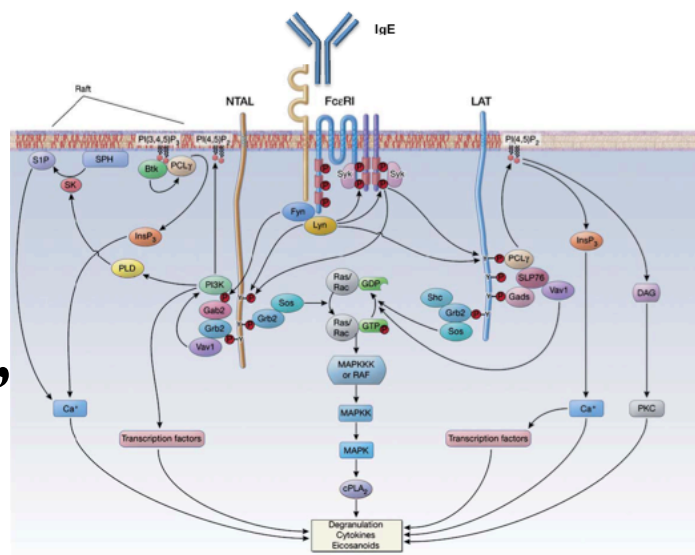


Houtman et al., *Nat. Struct. Mol. Biol.* (2006)

Nag et al., *Biophys. J.* (2009)

Limits of the network generation approach

- Extending model to include Lyn regulation results in >20,000 species.
- LAT may form large oligomers under physiological conditions.
- Many more components are still missing. Networks can easily reach “Avogadro limit”



Population- vs. Particle-Based Approaches to Simulation

Population

- Each species is enumerated

1. A
2. B
3. C
4. AB
5. BC
6. ABC

Particles

- Molecules are instantiated

1. A
2. A
3. B
4. B
5. B
6. C
7. C
8. C

Population- vs. Particle-Based Approaches to Simulation

Population

- Each species is enumerated
- Configuration is vector of populations


1.	A	1
2.	B	2
3.	C	3
4.	AB	1
5.	BC	0
6.	ABC	0

Particles

- Molecules are instantiated
- Configuration is complex data struct

1.	A	
2.	A	4
3.	B	
4.	B	2
5.	B	
6.	C	
7.	C	
8.	C	

bond




Population- vs. Particle-Based Approaches to Simulation

Population

- Each species is enumerated
- Configuration is vector of populations
- Update dependencies can be precomputed

Particles

- Molecules are instantiated
- Configuration is complex data struct
- Update dependencies computed on-the-fly



Population- vs. Particle-Based Approaches to Simulation

Population

- Each species is enumerated
- Configuration is vector of populations
- Update dependencies can be precomputed
- Single particles cannot be tracked

Particles

- Molecules are instantiated
- Configuration is complex data struct
- Update dependencies computed on-the-fly
- Single particles can be tracked

Population- vs. Particle-Based Approaches to Simulation

Population

Combinatorial complexity can make population-based simulations intractable!

vector of populations

- Update dependencies can be precomputed
- Single particles cannot be tracked

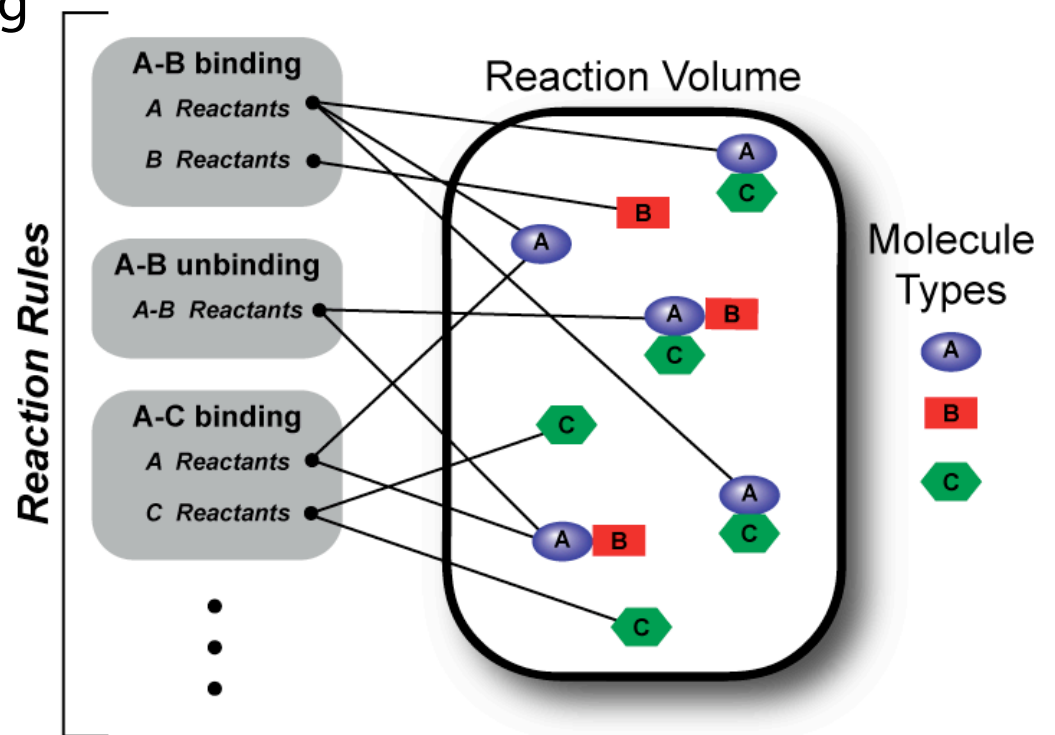
Particles

- Molecules are instantiated
- Configuration is complex data struct
- Update dependencies computed on-the-fly
- Single particles can be tracked

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 neighbor sites (*in Gillespie, updates are static*)



Sneddon, Faeder, and Emonet, in preparation.

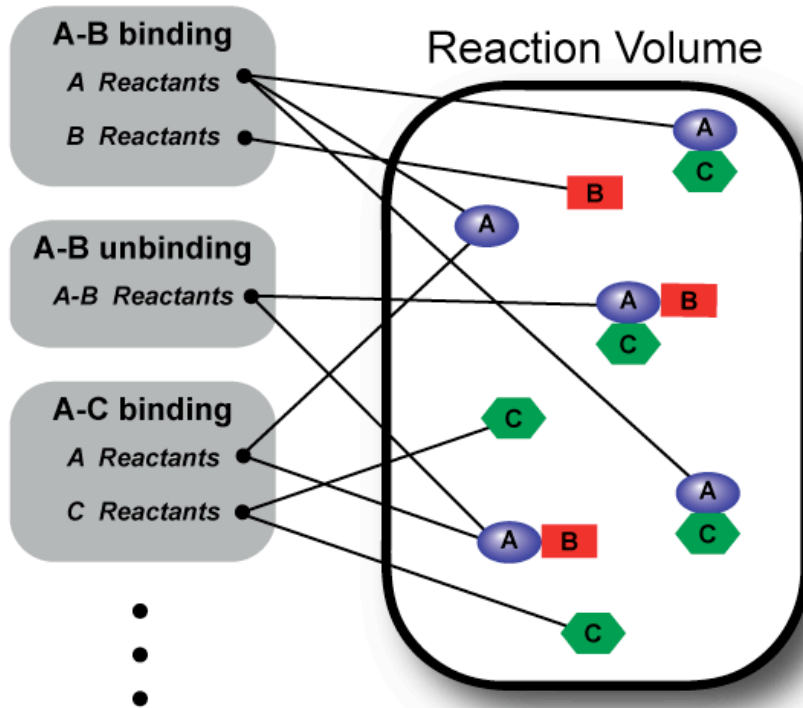
NFSIM Algorithm

Rates

25 s⁻¹

10 s⁻¹

0.1 s⁻¹



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.

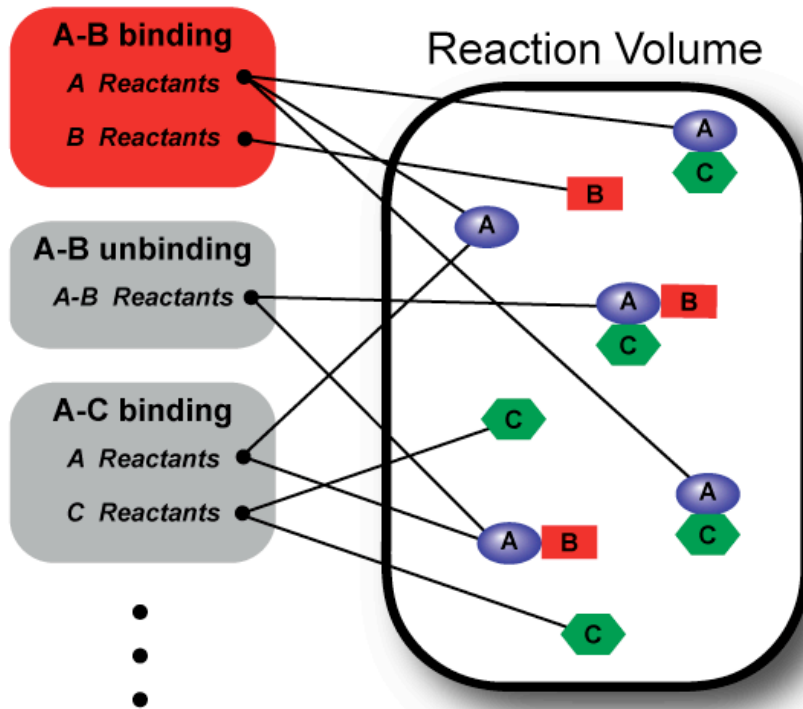
NFSIM Algorithm

Rates

25 s⁻¹

10 s⁻¹

0.1 s⁻¹



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.

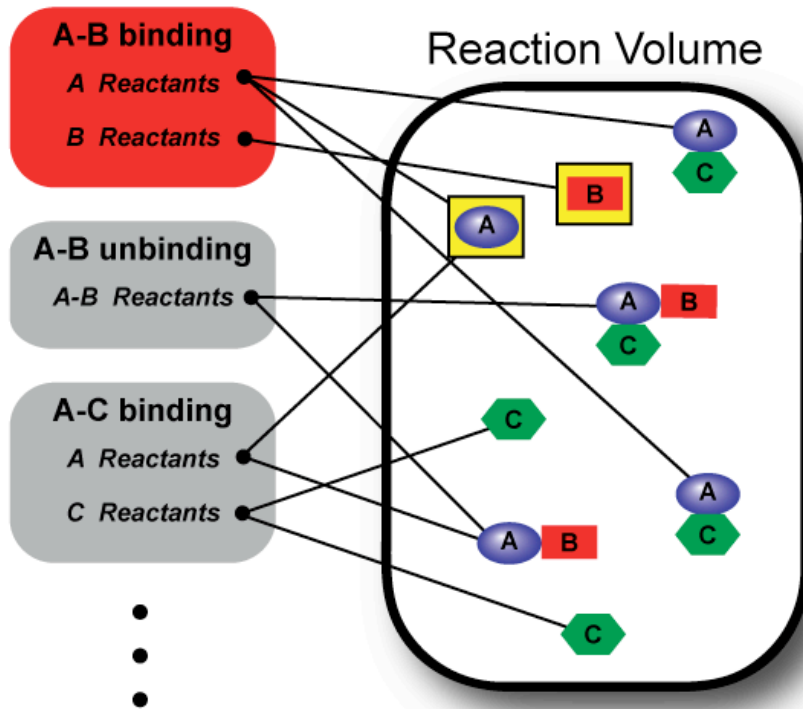
NFSIM Algorithm

Rates

25 s⁻¹

10 s⁻¹

0.1 s⁻¹



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.

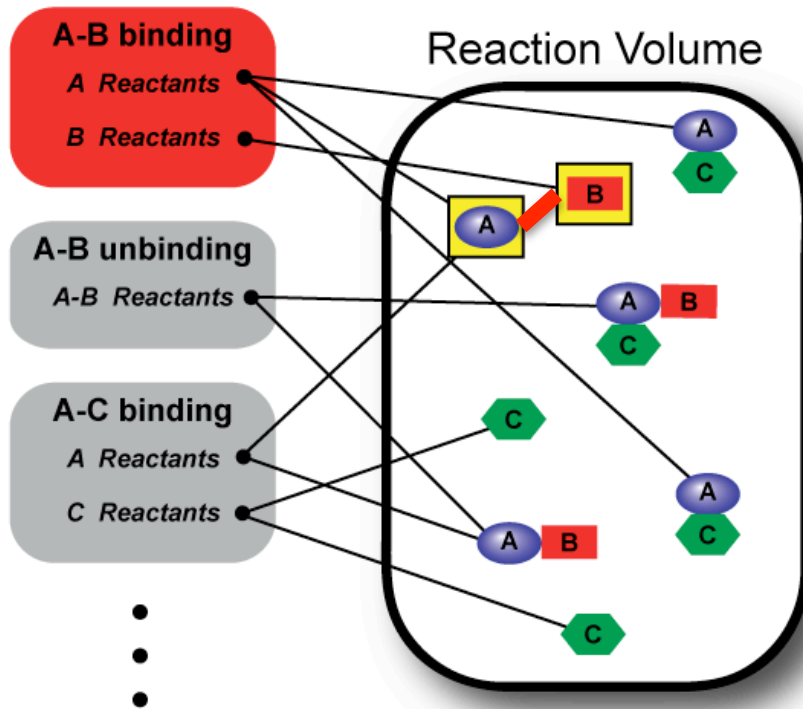
NFSIM Algorithm

Rates

25 s⁻¹

10 s⁻¹

0.1 s⁻¹



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.

NFSIM Algorithm

Rates

0 s⁻¹

A-B binding

A Reactants

B Reactants

15 s⁻¹

A-B unbinding

A-B Reactants

0.1 s⁻¹

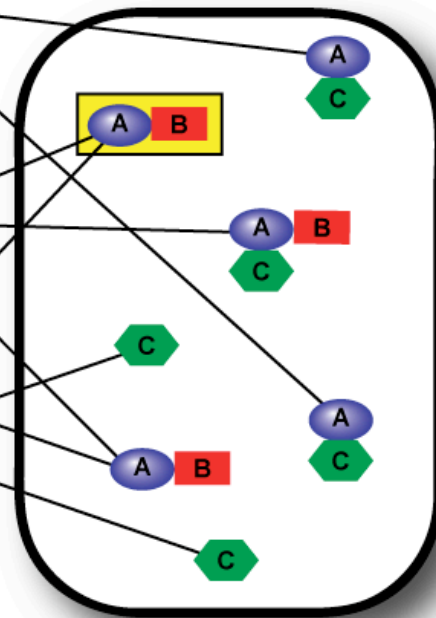
A-C binding

A Reactants

C Reactants

•
•
•

Reaction Volume



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.

NFSIM Algorithm

Rates

0 s^{-1}

A-B binding

A Reactants

B Reactants

15 s^{-1}

A-B unbinding

A-B Reactants

0.1 s^{-1}

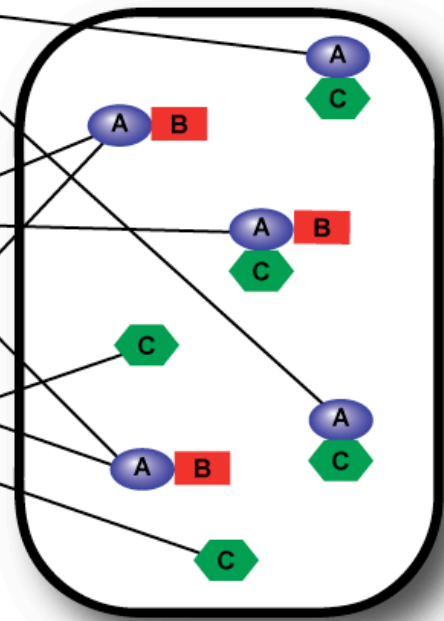
A-C binding

A Reactants

C Reactants

•
•
•

Reaction Volume

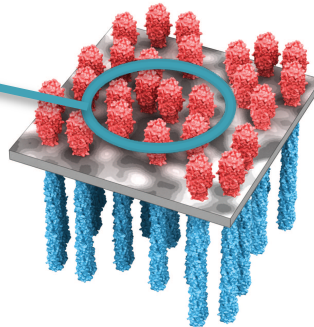


0. Initialize *reactant lists* and calculate *rule propensities*.
1. Select next reaction time and next *rule*.
2. Select molecules and sites to react.
 - a. Check any application condition(s).
3. Apply operation specified by rule.
4. Update reactant lists and propensities.
5. **Increment time.**

NFSIM Core Simulator Features

- 1) Modular C++ code base and highly efficient implementation
- 2) Operates seamlessly with BIONETGEN
- 3) Extended BIONETGEN Language handles
 - 1) Spatial compartments
 - 2) System variables in rate law expressions

cooperative
receptor
interactions



$$\text{MethLevel}(x) = 1 * R1(x) + 2 * R2(x) + 3 * R3(x) + 4 * R4(x) + 5 * R5(x) + 6 * R6(x) + 7 * R7(x) + 8 * R8(x)$$

Multi-site Phosphorylation

BioNetGen Language [2]

begin molecule types

Kinase(s)

Phosphatase(s)

Prot(p~U~P,p~U~P,p~U~P)

end molecule types

begin reaction rules

Kinase(s) + Prot(p~U) <-> Kinase(s!1).Prot(p~U!1)

Kinase(s!1).Prot(p~U!1) -> Kinase(s) + Prot(p~P)

...

end reaction rules

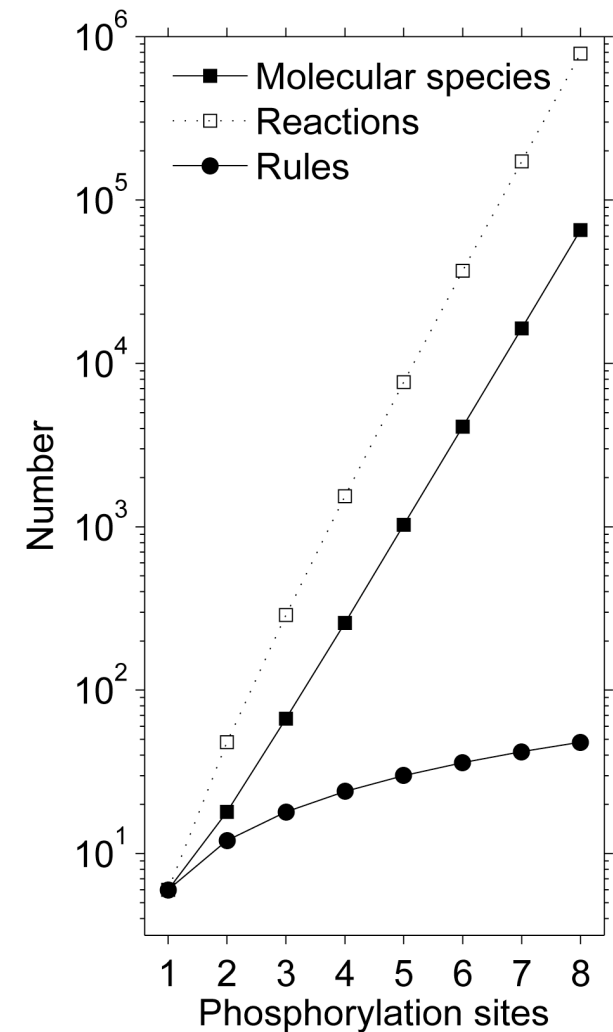
begin observables

Molecules Prot-P Prot(p~P,p~U,p~U)

Molecules Prot-P Prot(p~P,p~P,p~U)

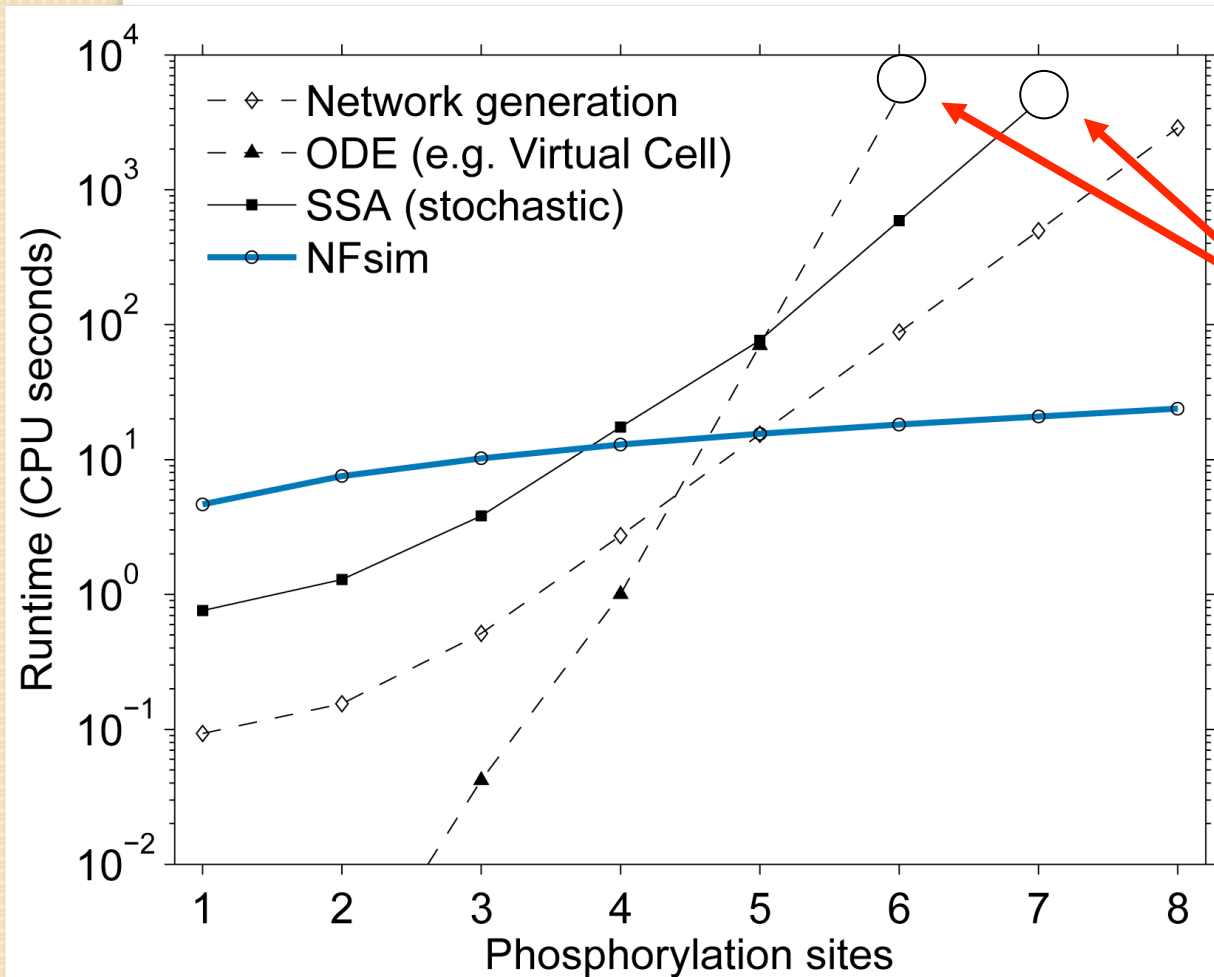
Molecules Prot-P Prot(p~P,p~P,p~P)

end observables



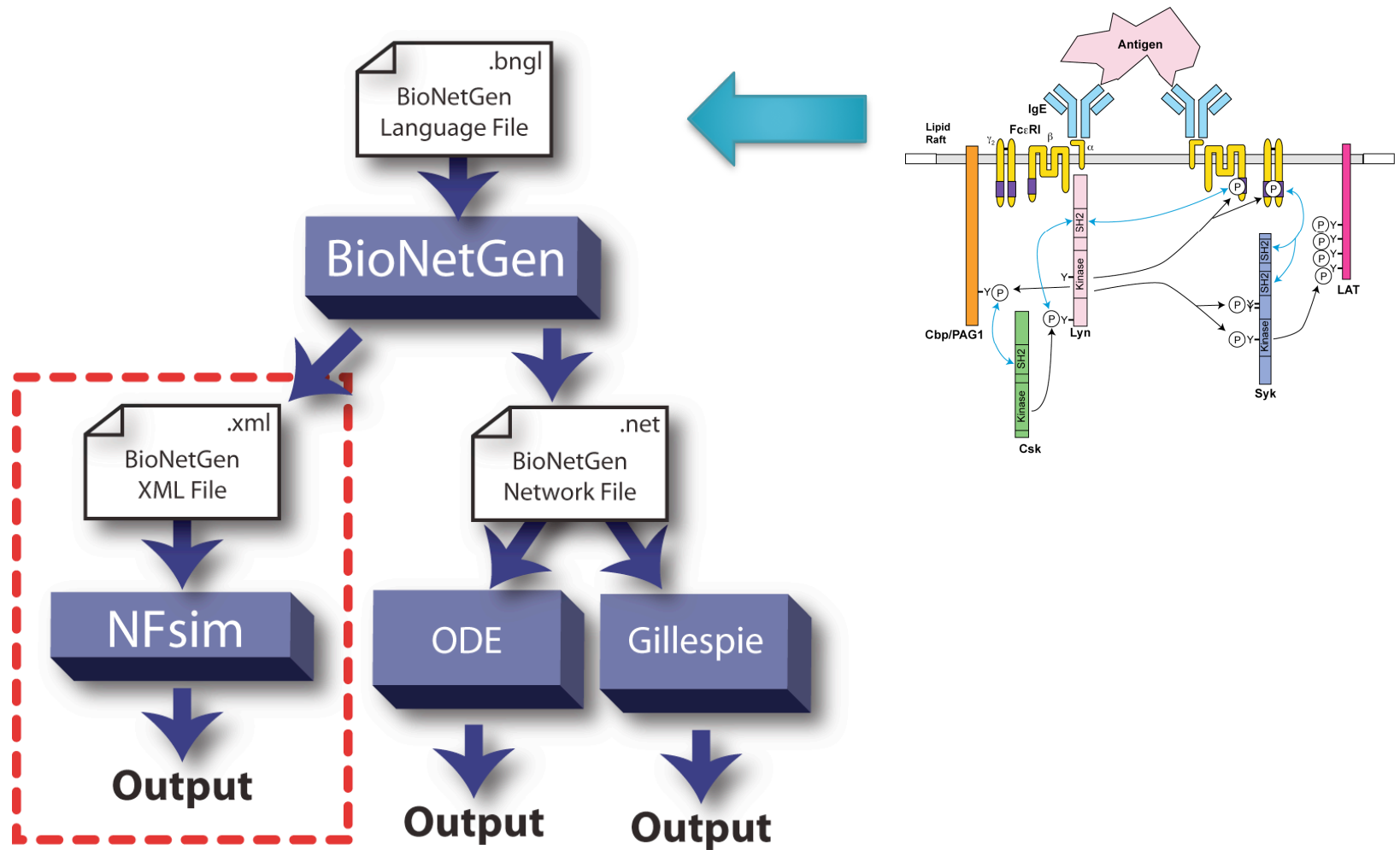
Michael Sneddon and Thierry Emonet

Multi-site Phosphorylation

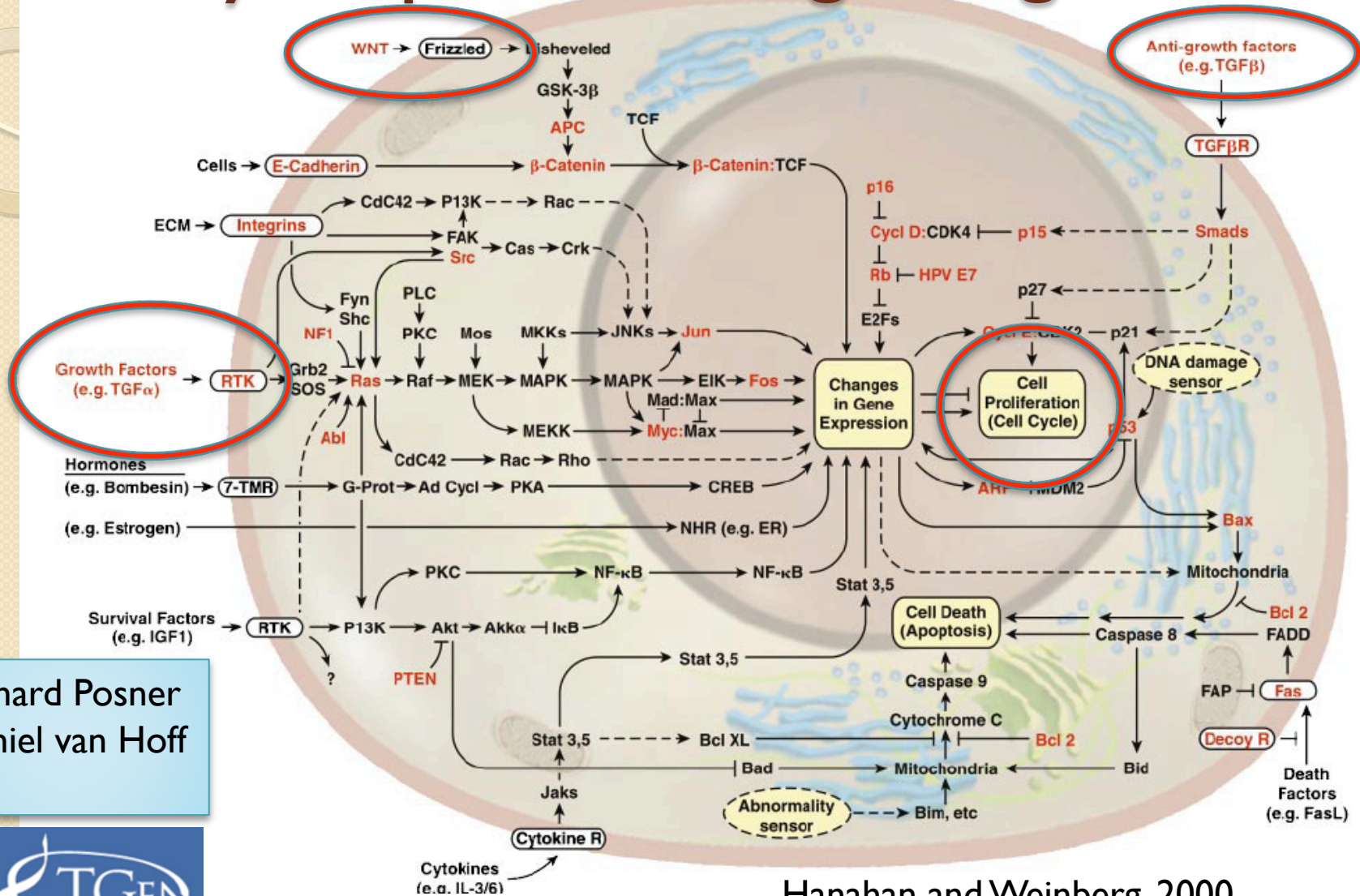


Not possible with
ODEs or SSA!

Integration with BIO NET GEN



Subway Map of Cell Signaling



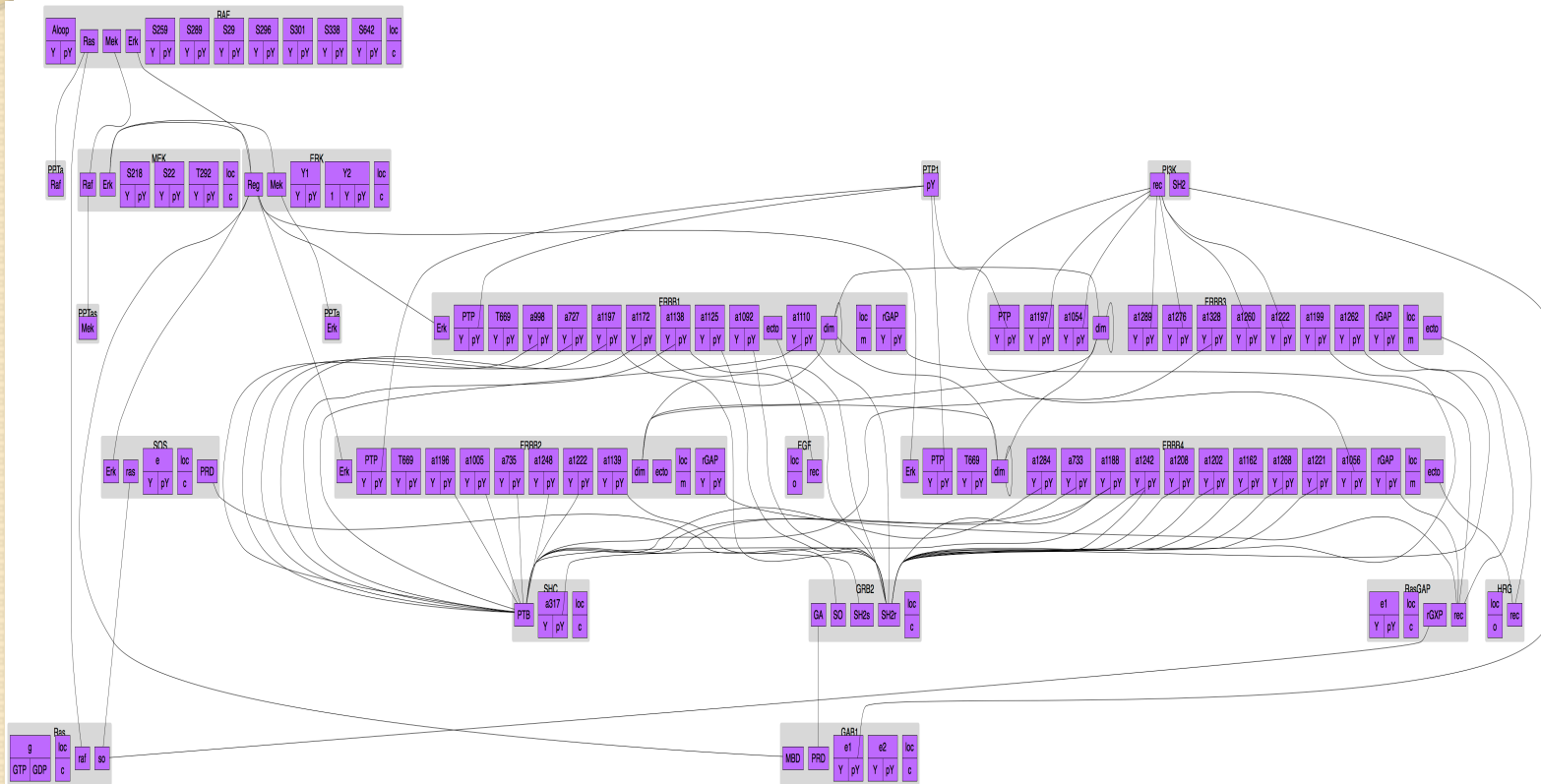
Richard Posner
Daniel van Hoff
...



Hanahan and Weinberg, 2000

Rule-based Model of EGFR Signaling

Preliminary Model: 20 molecules / 532 rules / 496 parameters



Matt Creamer and Rich Posner

Stats

Model

- 20 Molecule Types
 - 4 Receptors
 - 3 Ligands
- 536 Parameters
- 547 Reaction Rules

Simulation

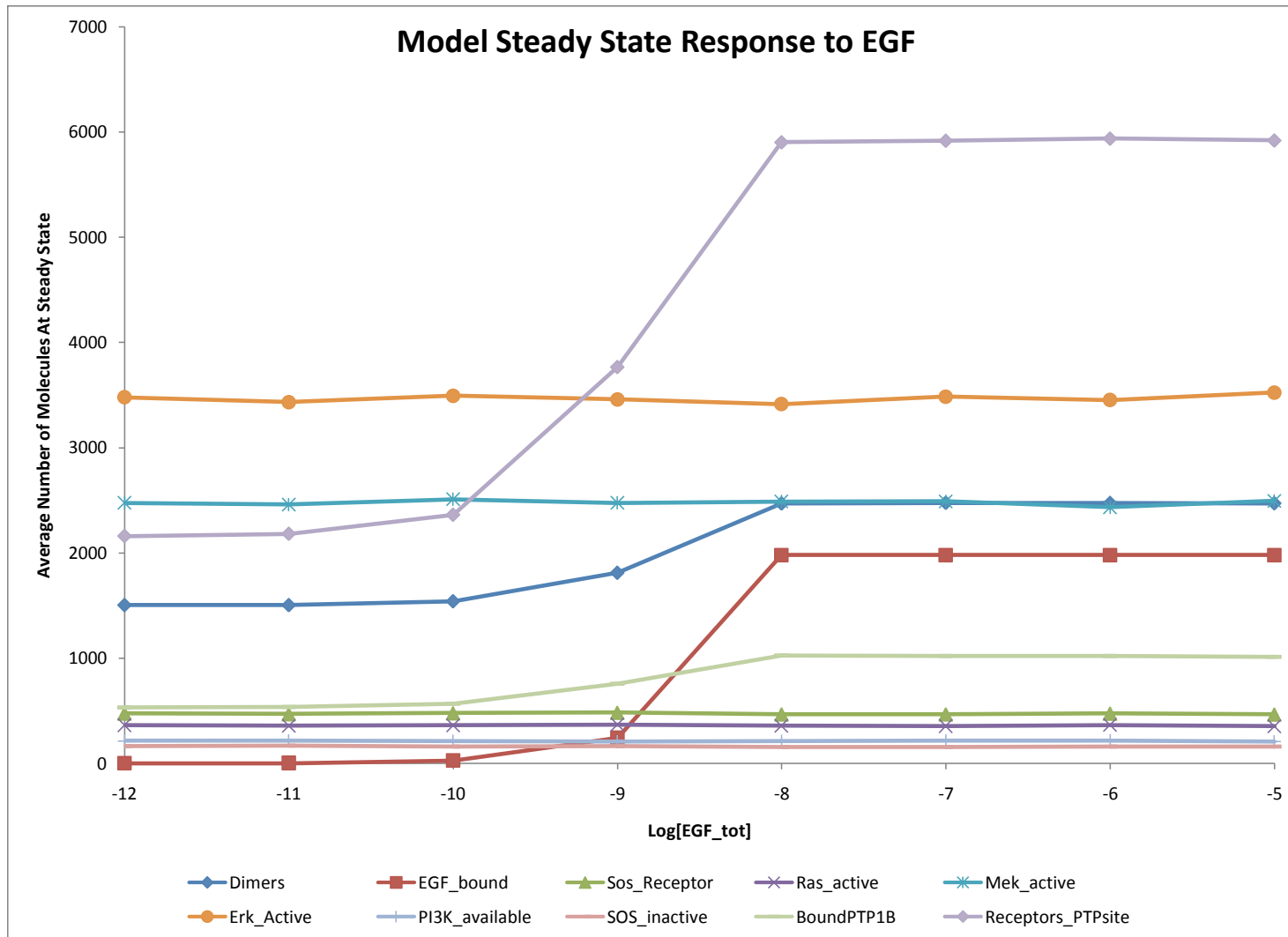
1500 sim sec

- ~10-18 million events
- ~ 1060 real sec

~ 6e-5 CPU seconds/event

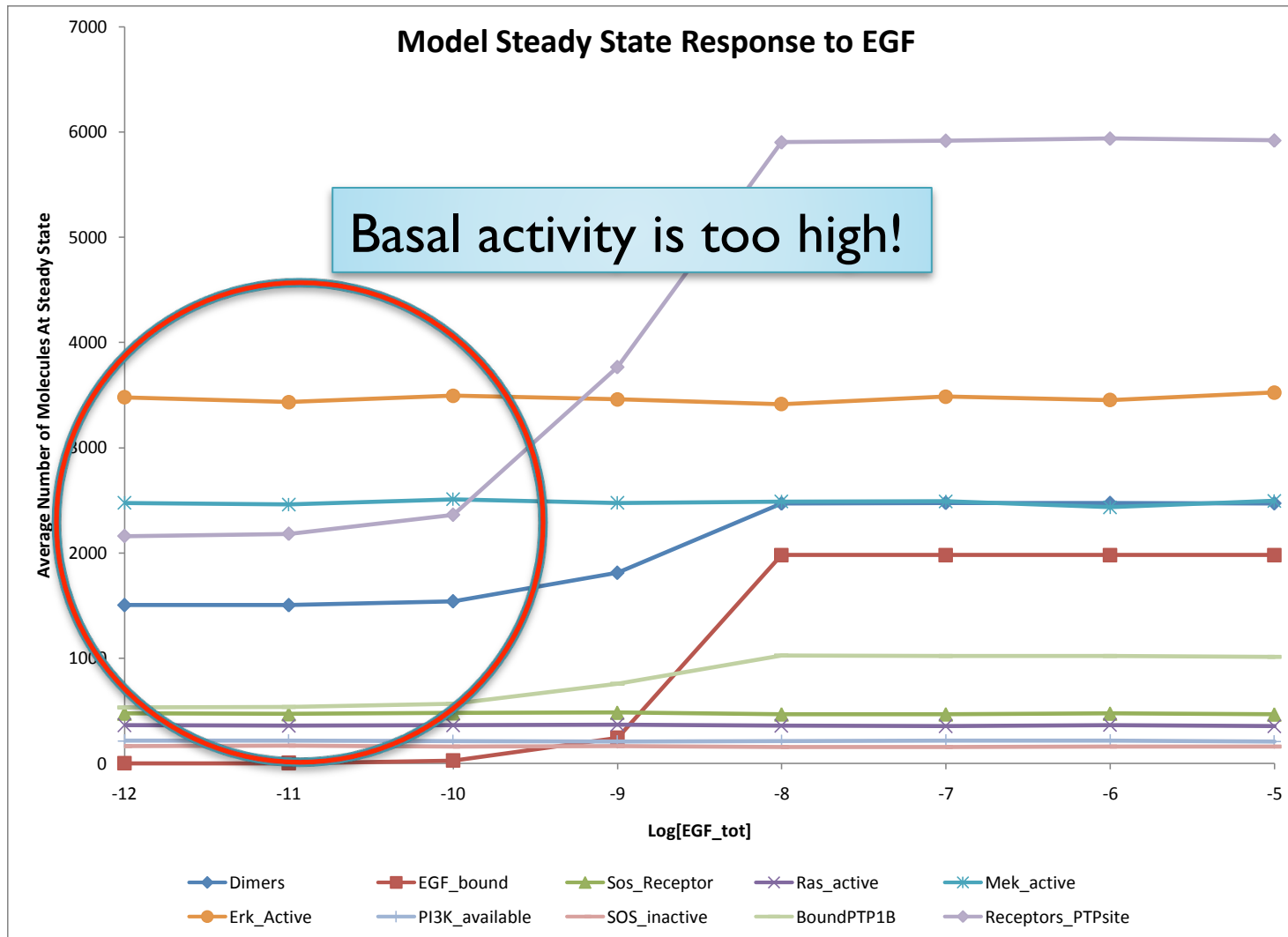
(On a 2.4 GHz Intel Core2Duo on
iMac with 4 GB RAM)

Model Validation

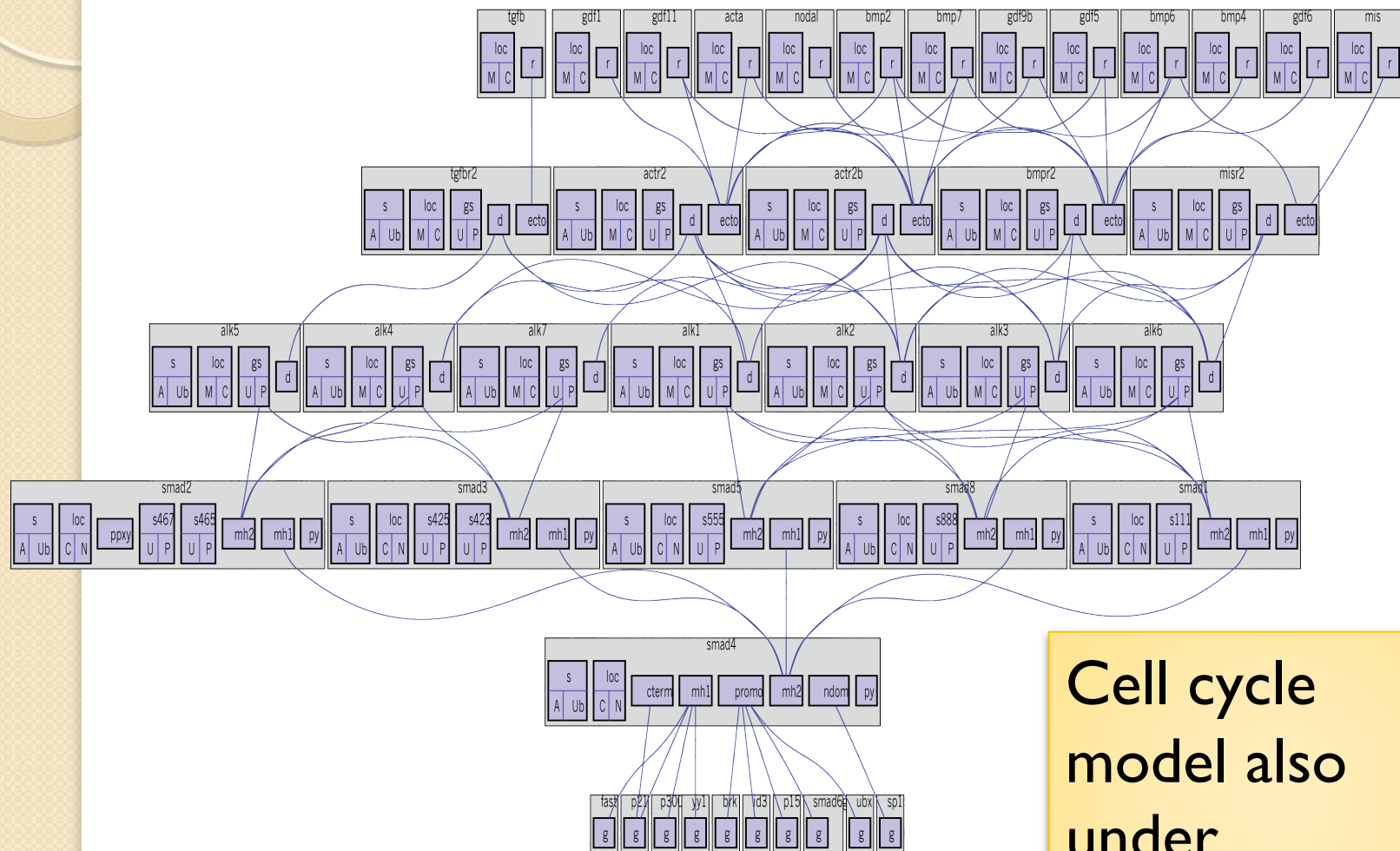


John Sekar

Model Validation



Stop # 2: TGF- β Pathway



Cell cycle model also under development



The Path Ahead

- Continue to build and analyze models of key pathways
- Systematic investigation of models using
 - Statistical and Bayesian Model Checking
 - Global parameter sensitivity analysis
 - Parameter estimation and synthesis
- Integration of pathway models
- Model reduction
 - Coarse-graining of detailed models (bottom up)
 - Comparison / Mapping to logical models (top down)

Collaborators



Thank You!



Photo by John
Sekar