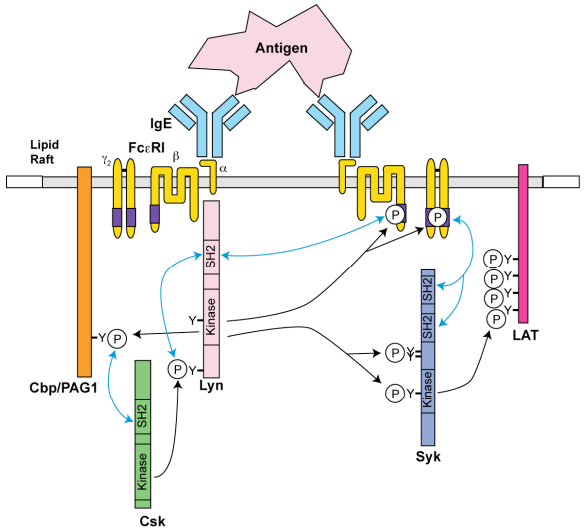
Navigating the Subway Map of the Cell

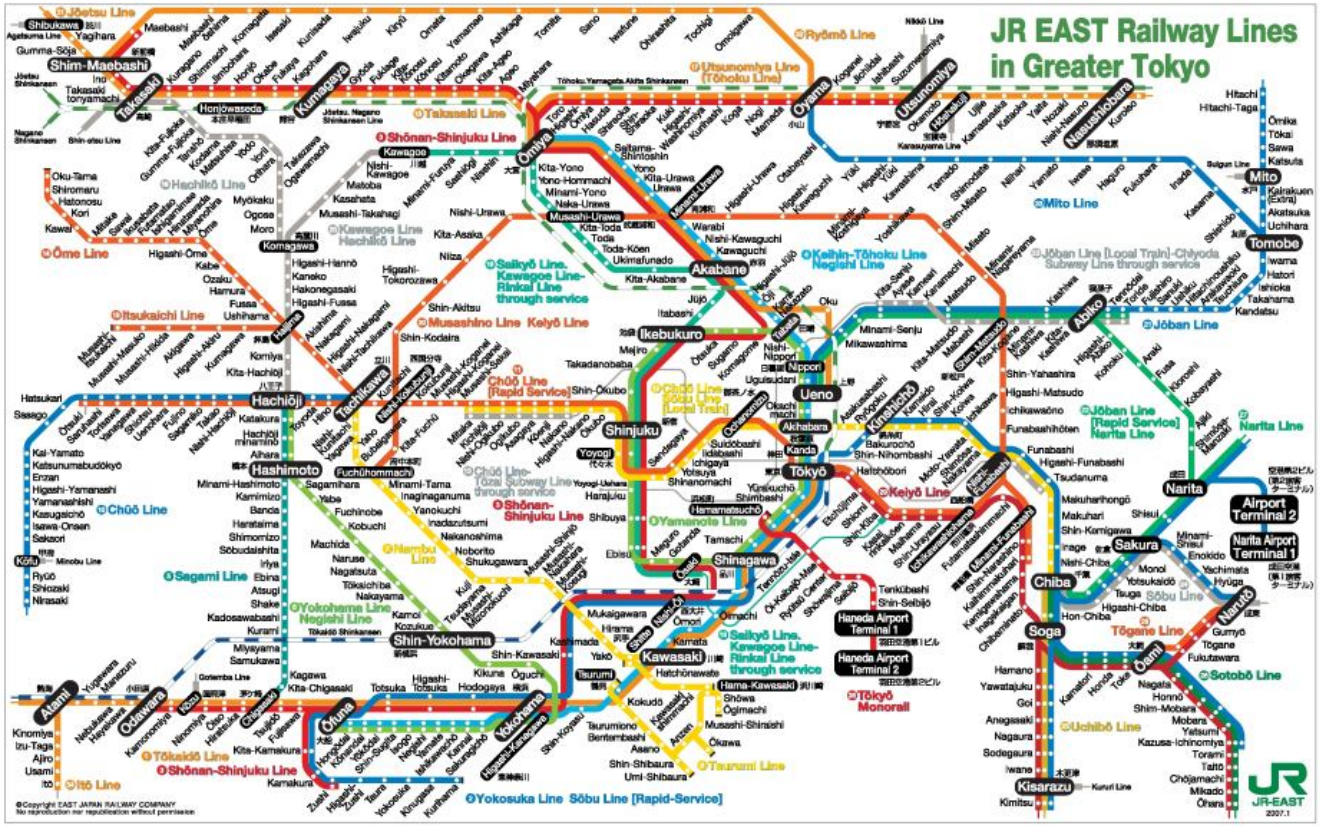


Jim Faeder
Department of Computational Biology
University of Pittsburgh School of Medicine

CMACS Expeditions Kickoff Meeting
October 31, 2009

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

Not really a map talk…



…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…

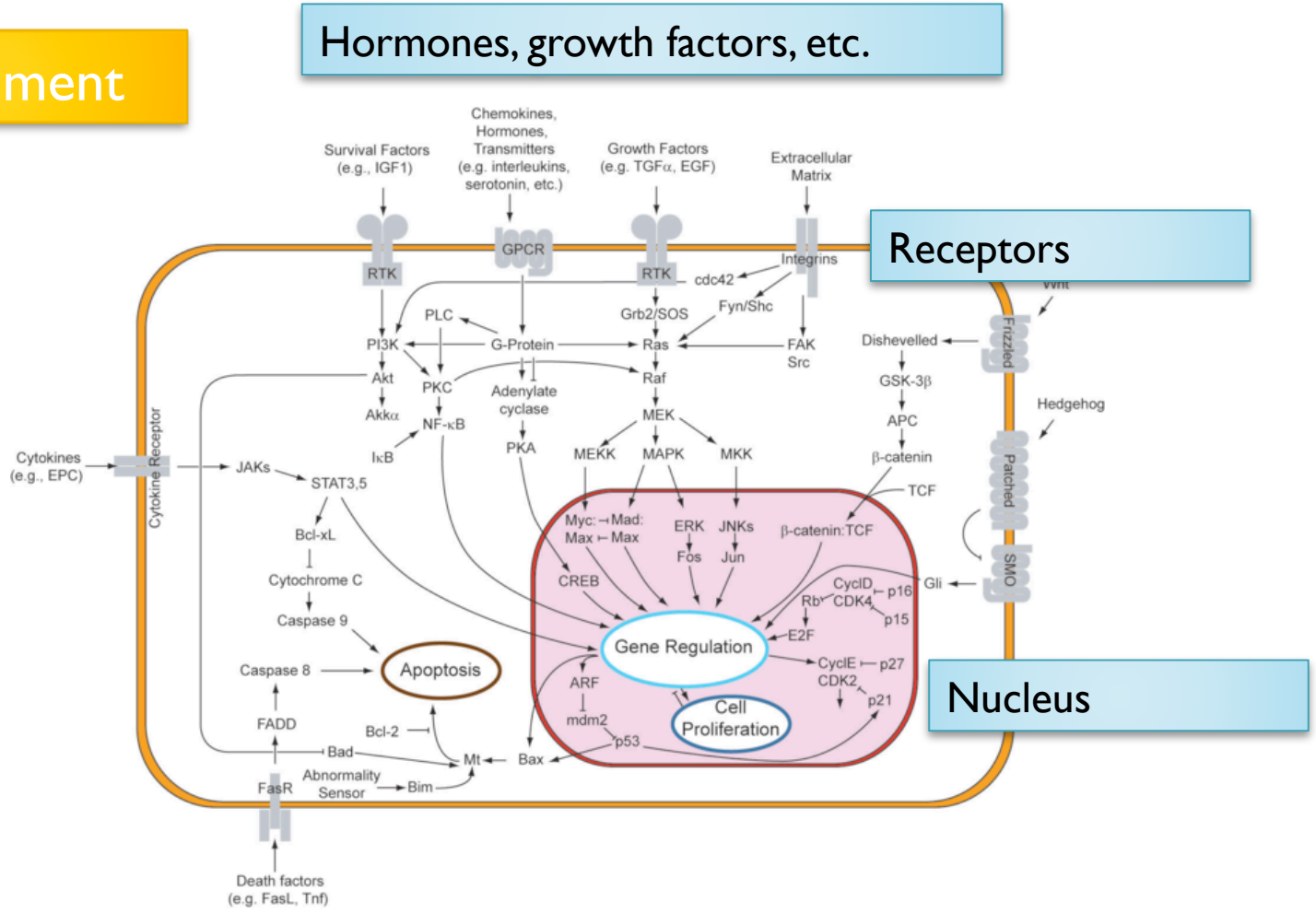
A-train

so new construction is required.

How Cells Process Information

Environment

Hormones, growth factors, etc.

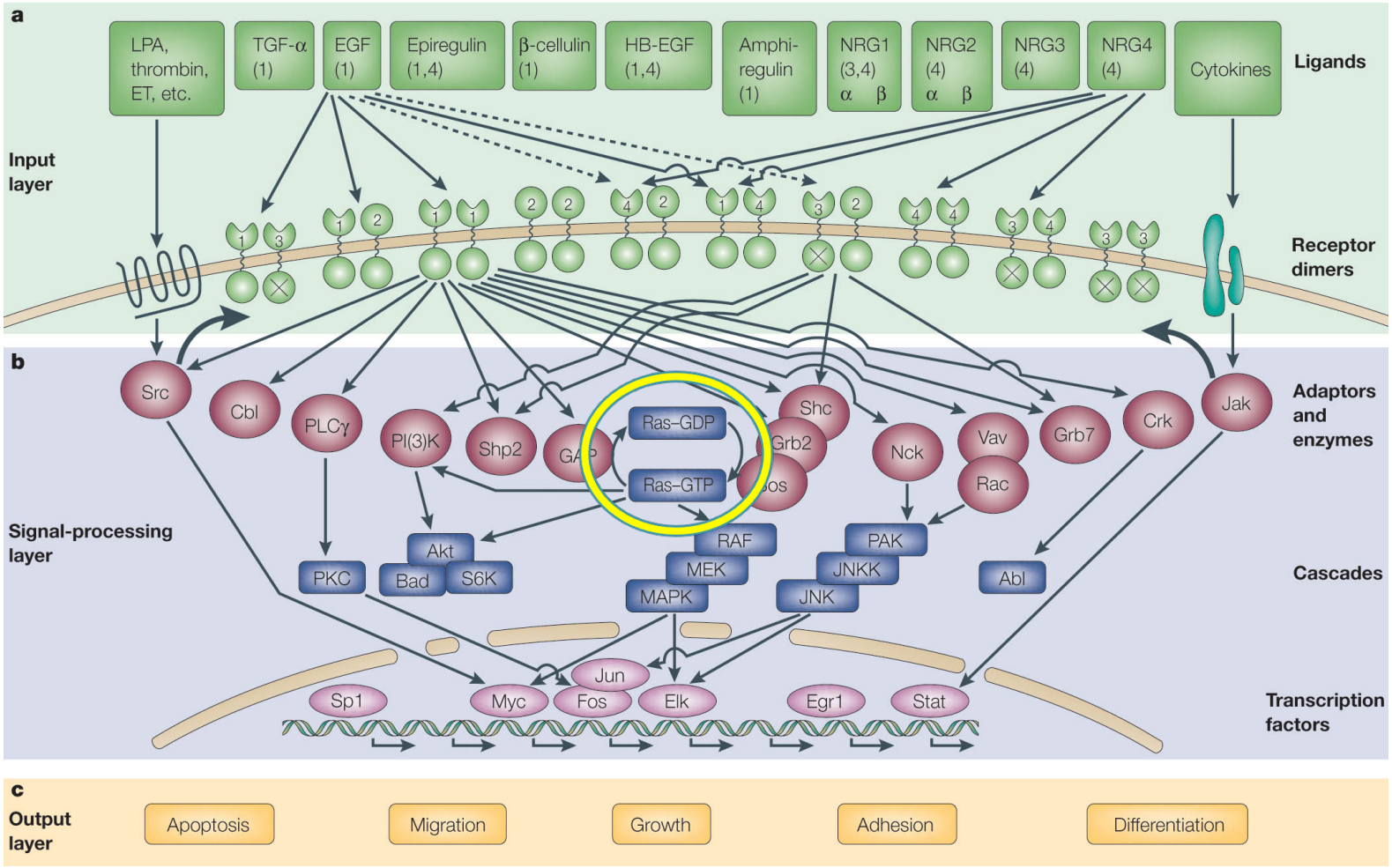


Receptors

Nucleus

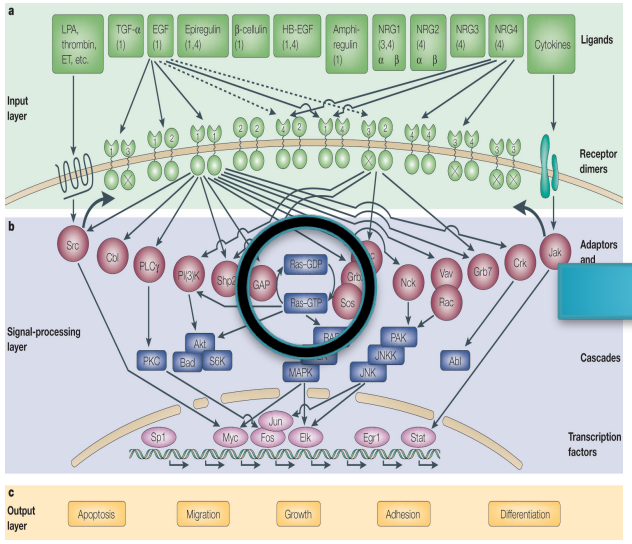
http://en.wikipedia.org/wiki/Cell_signaling

Architecture of a signaling network

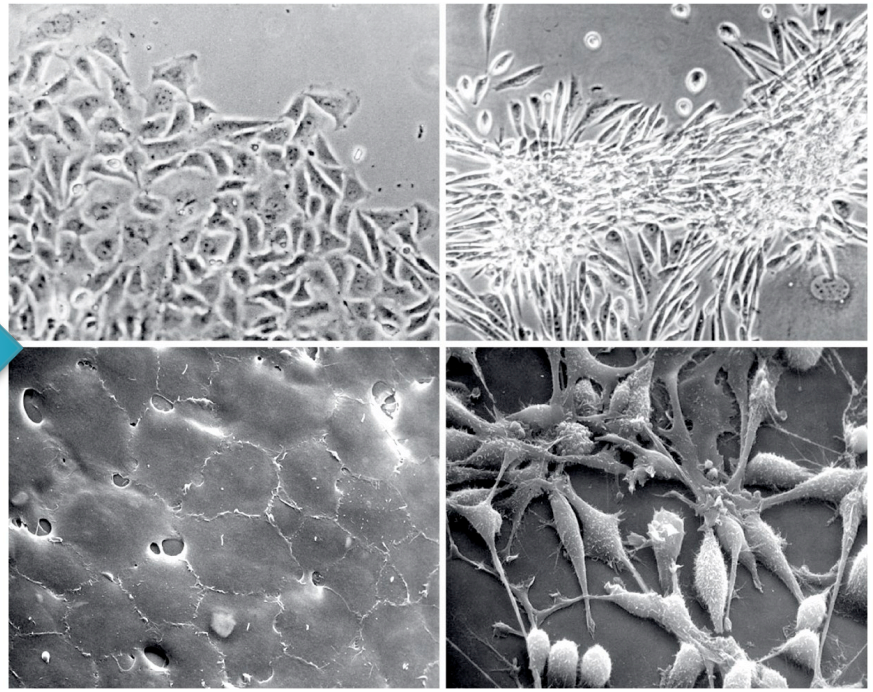


Yarden & Sliwkowski, *Nature Rev. Mol. Cell Biol.* **02**: 127-137 (2001).

Mutation of Ras Can Produce a Tumor Cell

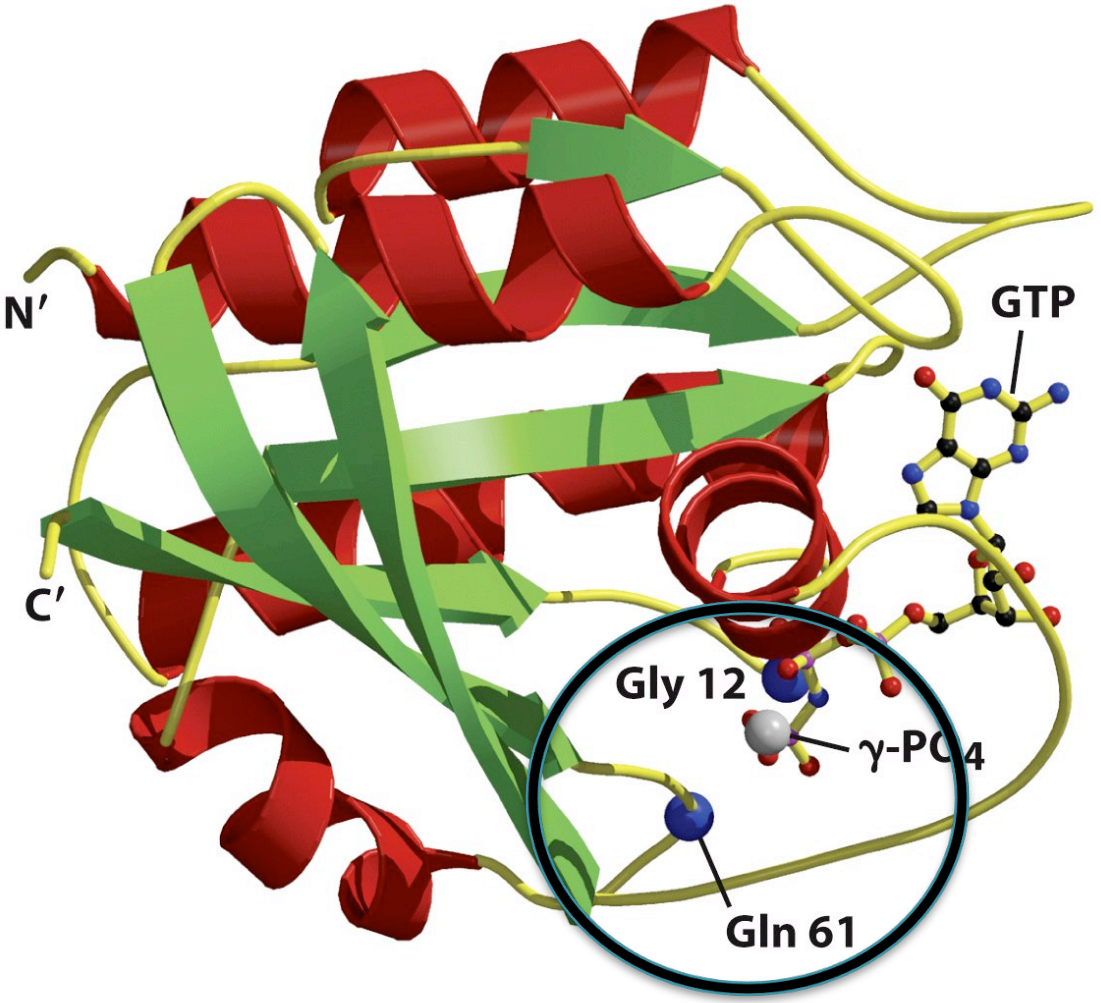


Normal **Transformed**



Ras mutations in cancer

Ras



>20% human tumors carry Ras point mutations.

>90% in *pancreatic cancer*.

The Biology of Cancer (© Garland Science 2007)

Modularity of Signaling Proteins

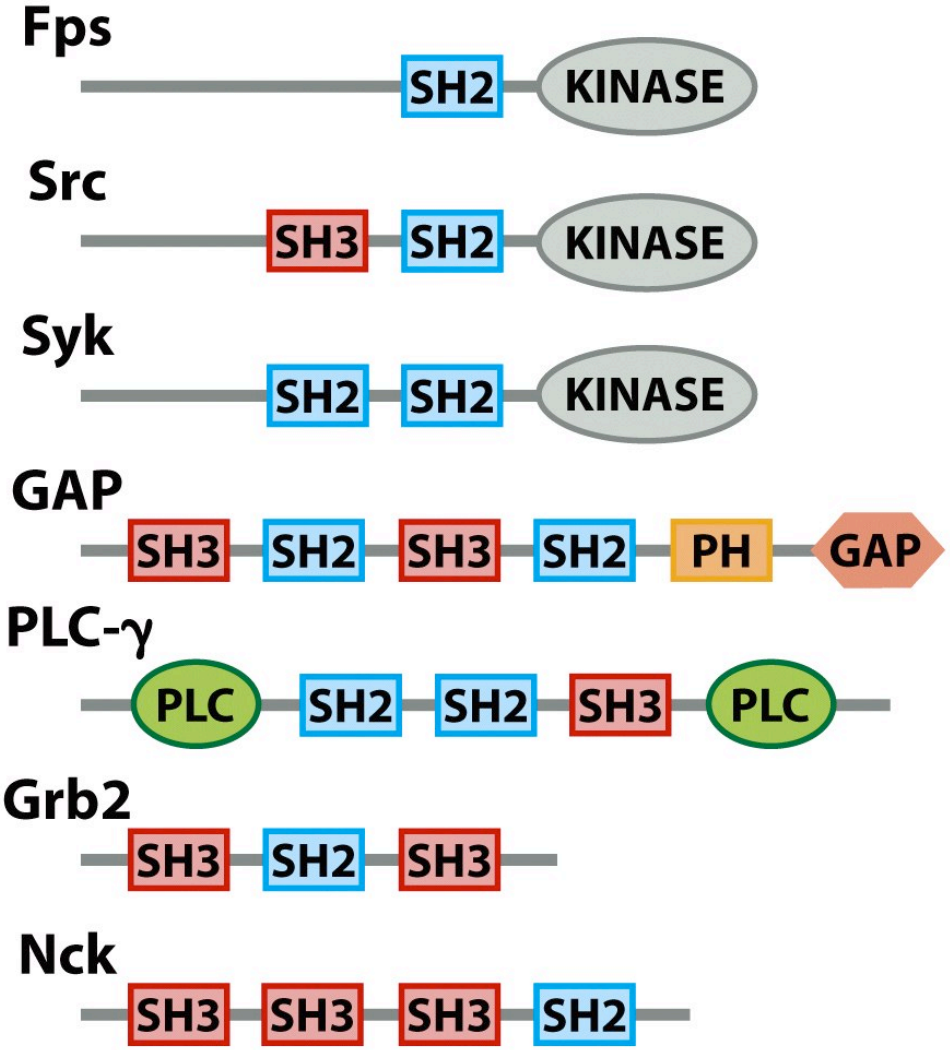
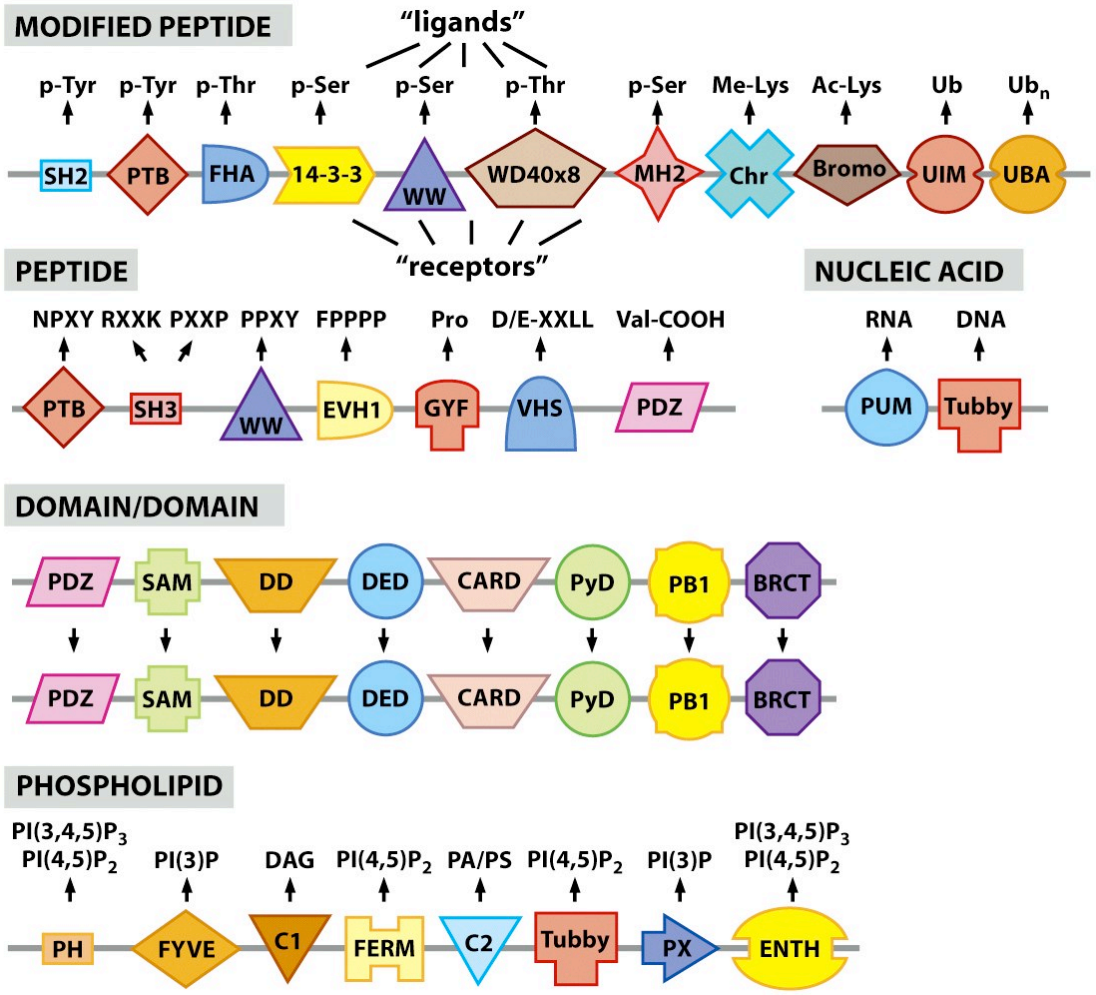


Figure 6.10a *The Biology of Cancer* (© Garland Science 2007)

Modularity produces complex wiring



Complexity of Receptor Complexes

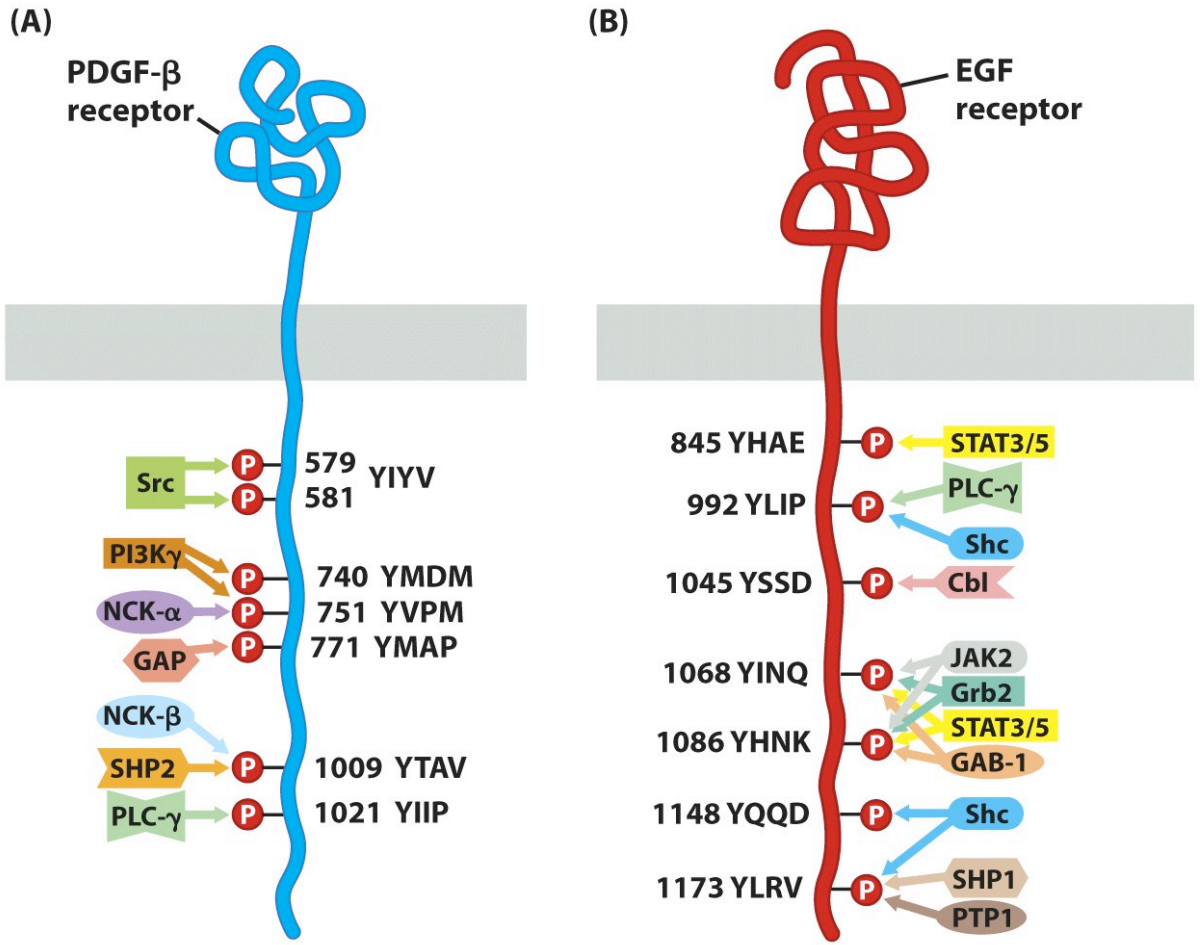
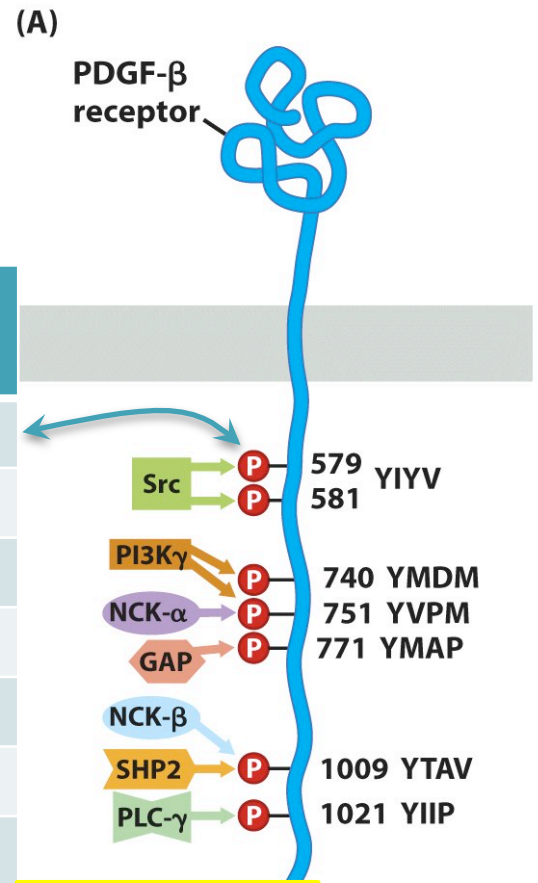
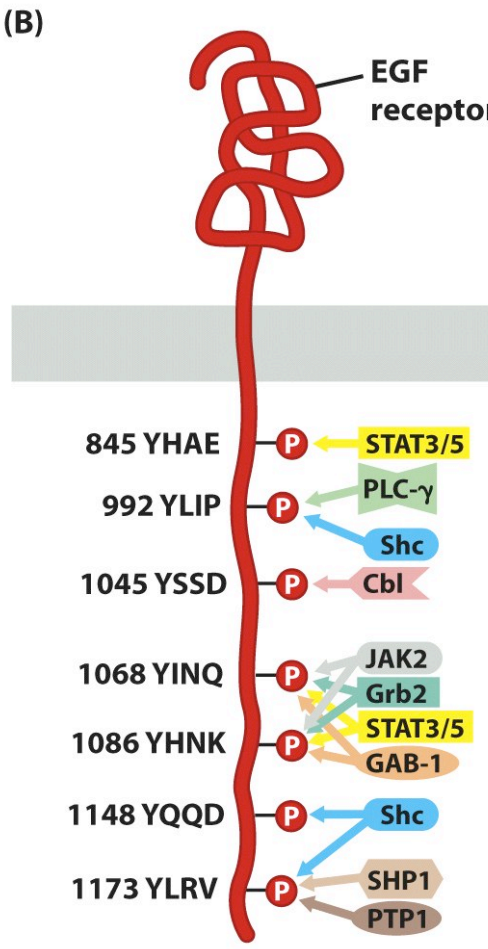


Figure 6.9 *The Biology of Cancer* (© Garland Science 2007)

The "curse" of complexity

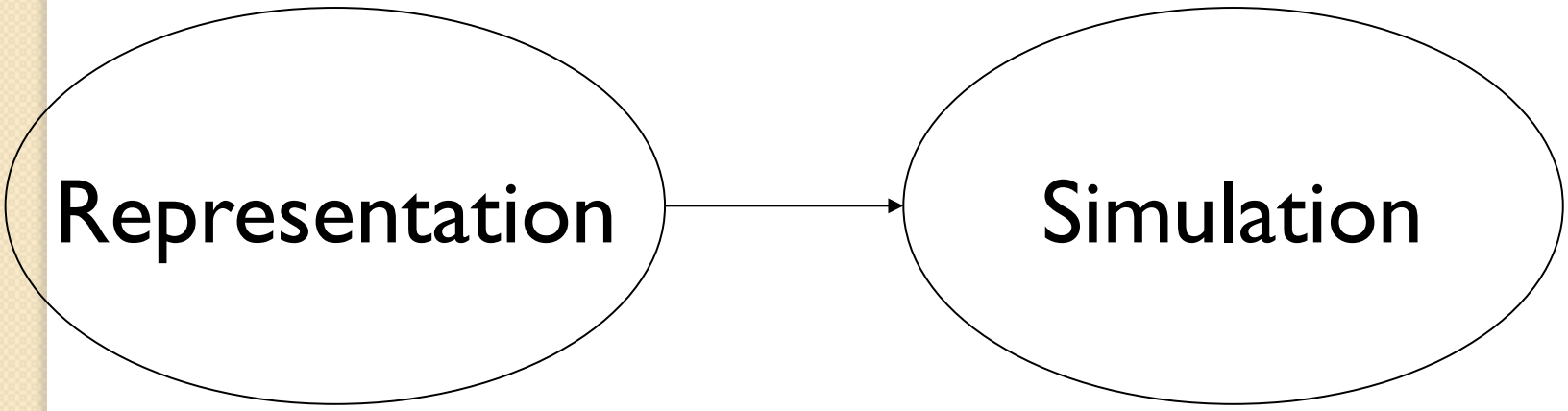




Number of States		Number of States
3		3
3		4
3		3
4		6
3		6
4		3
3		5
3,888	Monomers	19,440
7,560,216	Dimers	188,966,520

Modeling cell signaling

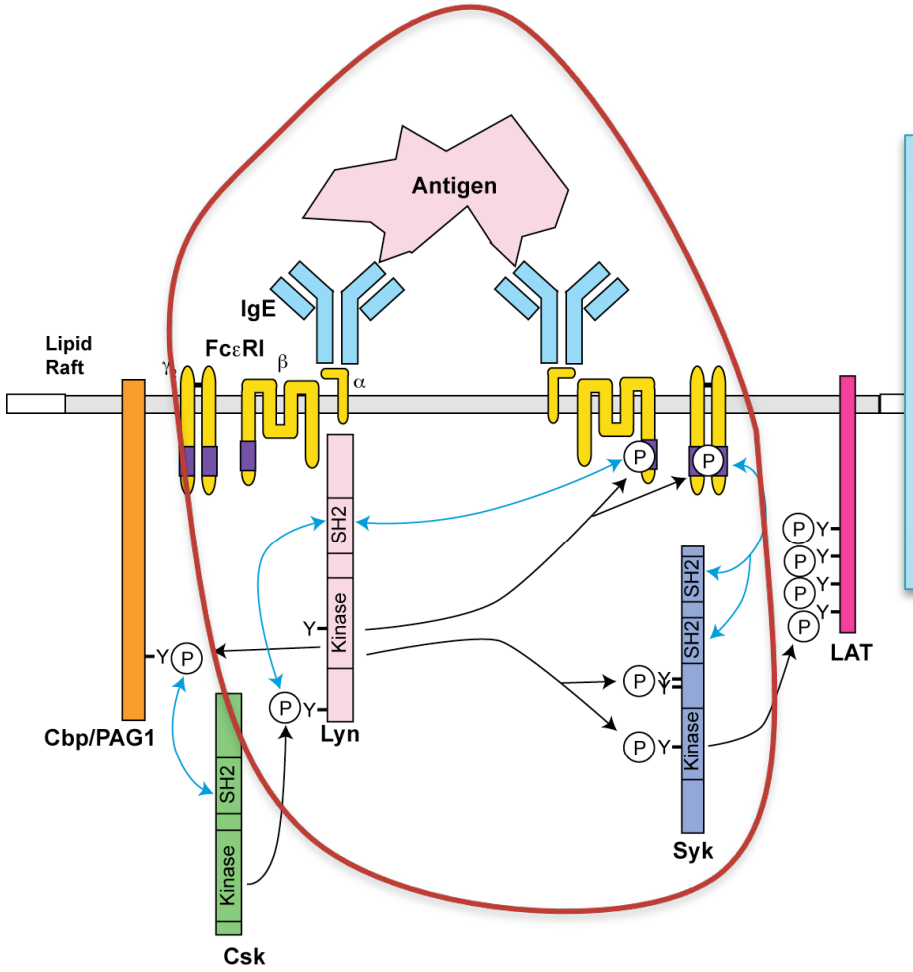
AIM: Model the biochemical machinery by which cells process information (and respond to it).



BIONETGEN Language
kappa
etc.

ODE, PDE
Stochastic Simulation Algorithm
Kinetic Monte Carlo
Brownian dynamics

Syk activation model

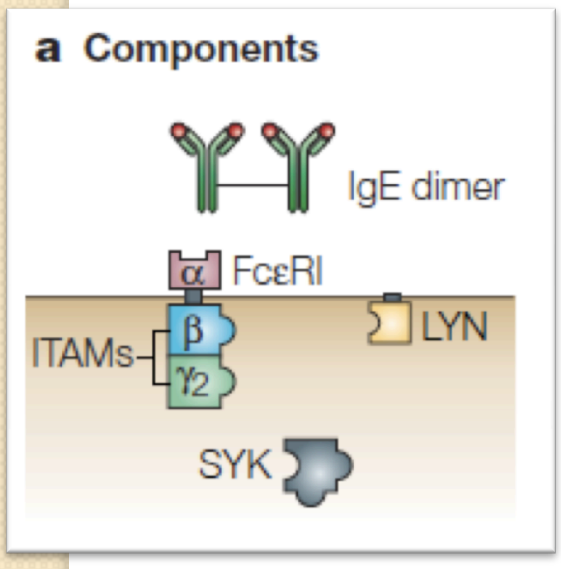


Key variables

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

Mol. Immunol.,2002
J. Immunol., 2003

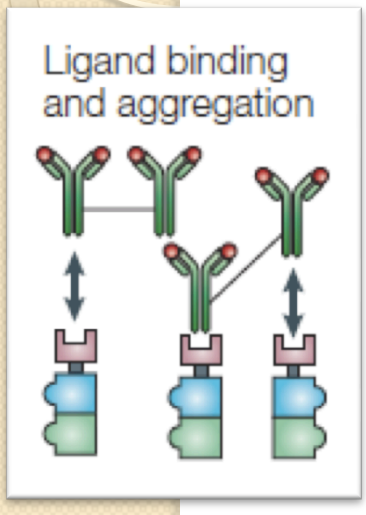
Defining Molecules



BIONETGEN Language

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

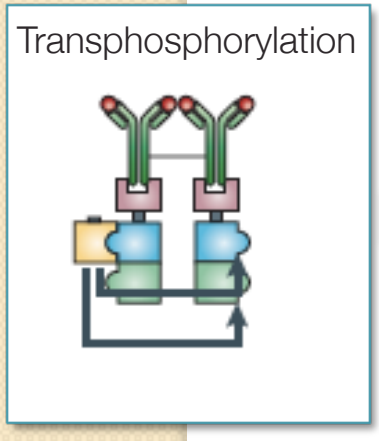
Defining Interaction Rules



BioNetGen Language

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

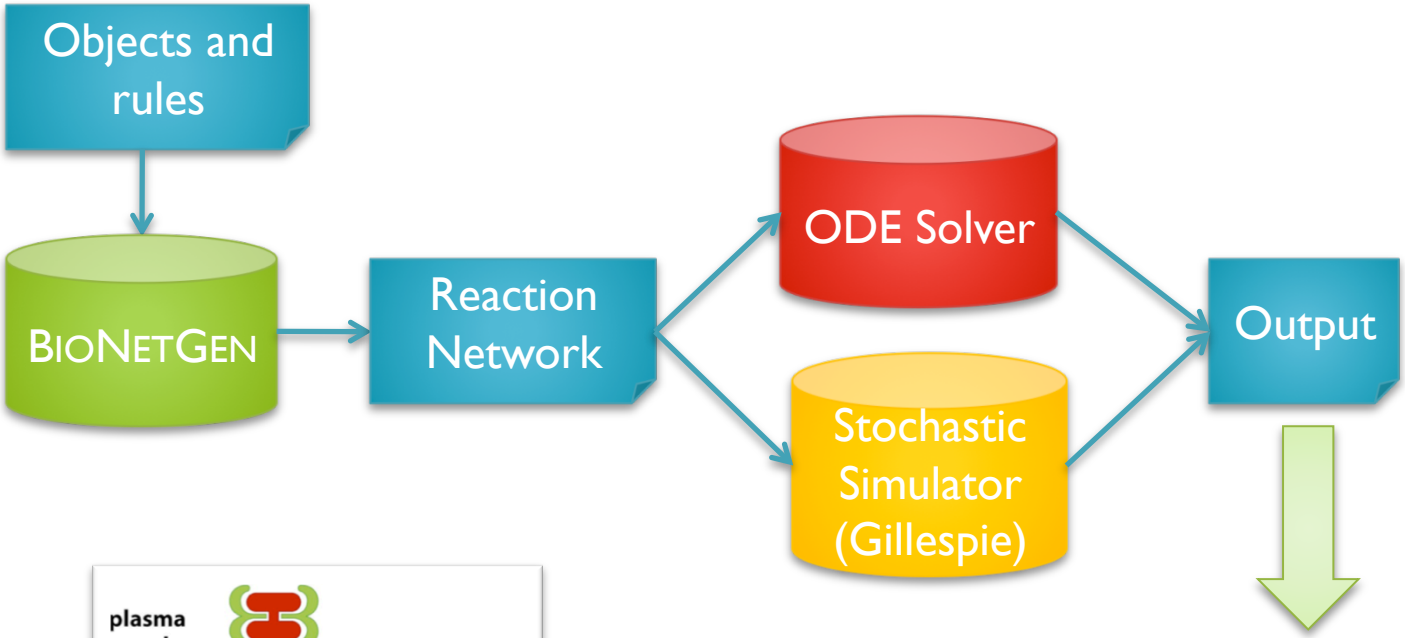
binding and dissociation

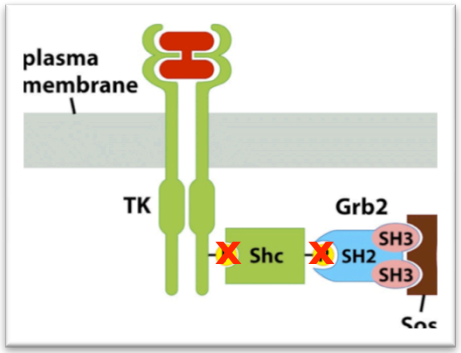


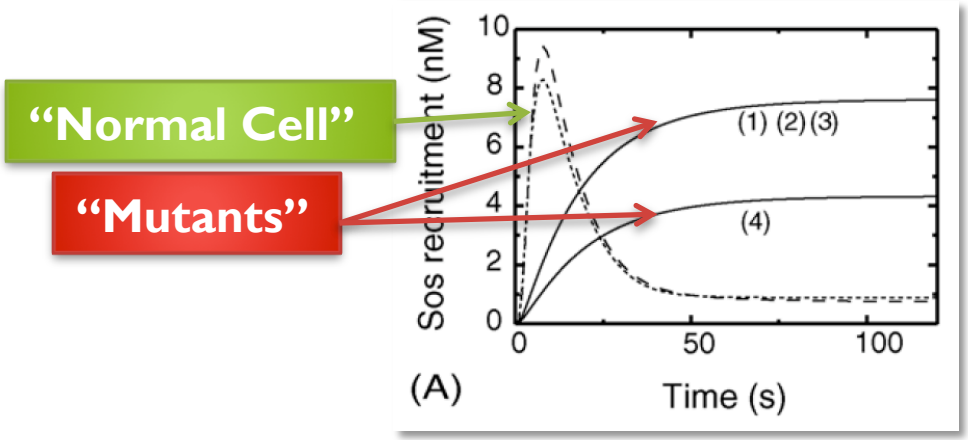

```
Lyn(U!1).FceRI(b!1).FceRI(b~U)-> \
Lyn(U!1).FceRI(b!1).FceRI(b~P)
```

component state change

Rule-based modeling protocol

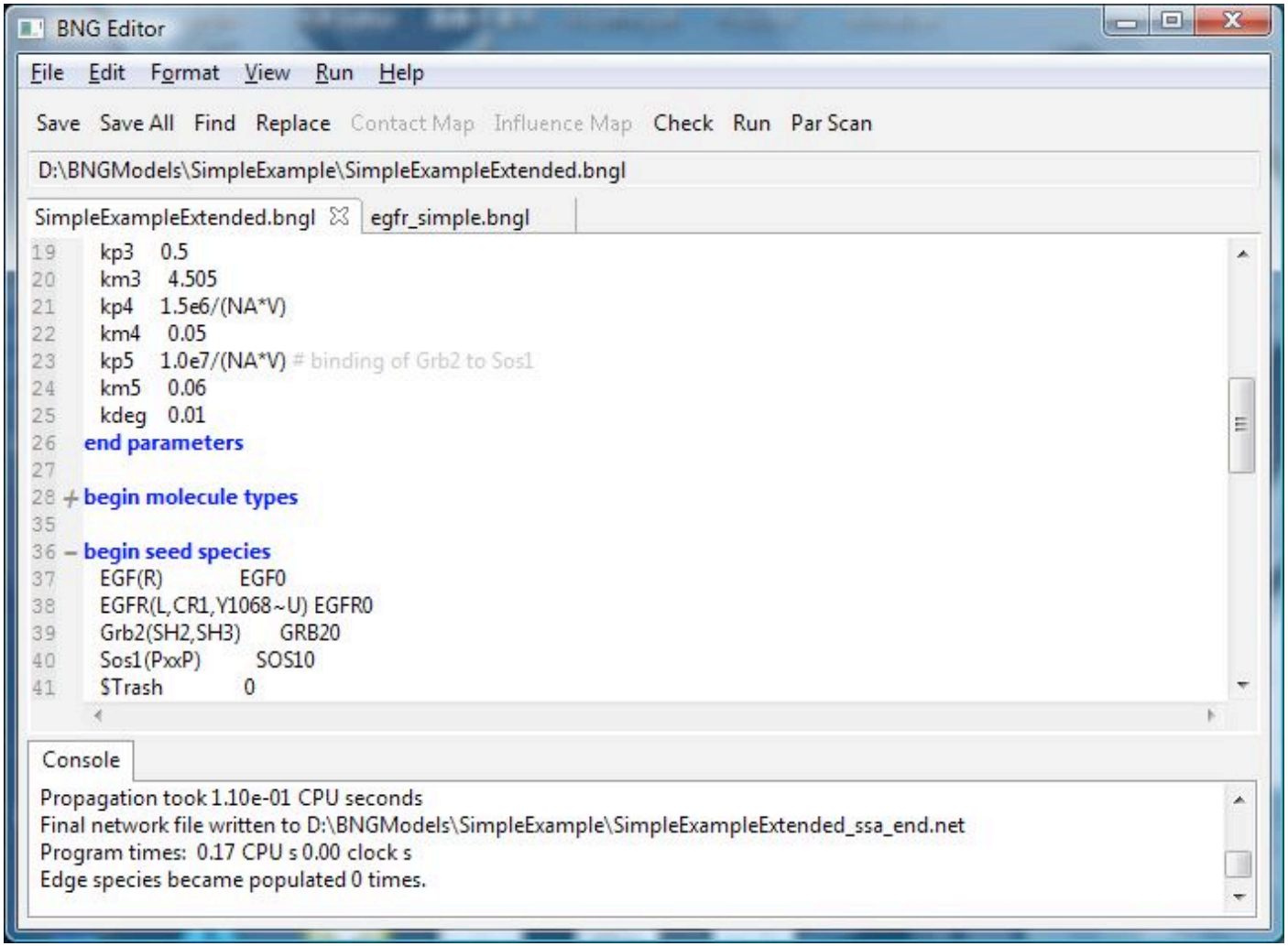






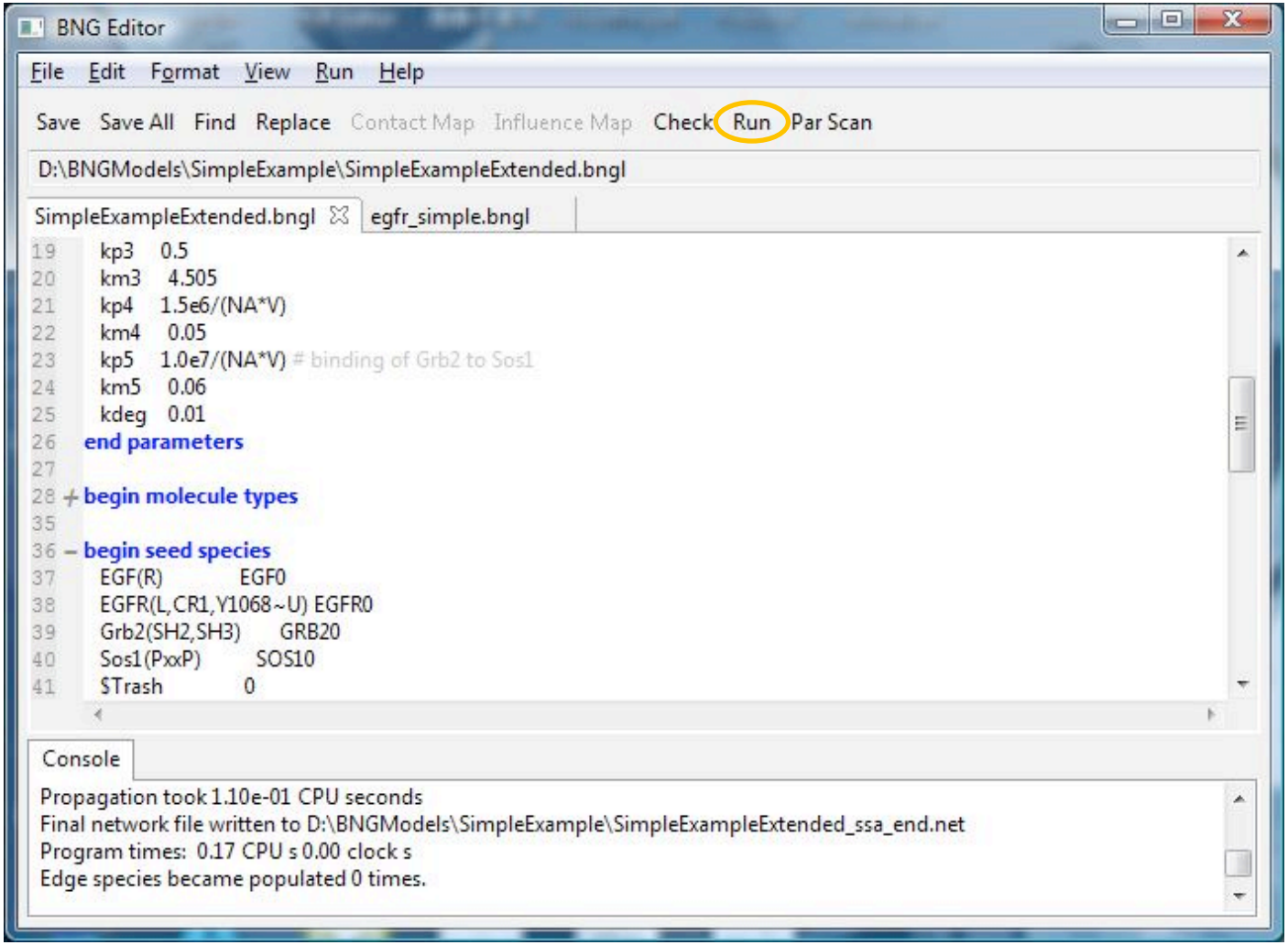
http://bionetgen.org

BioNetGen Editor - BiNGE

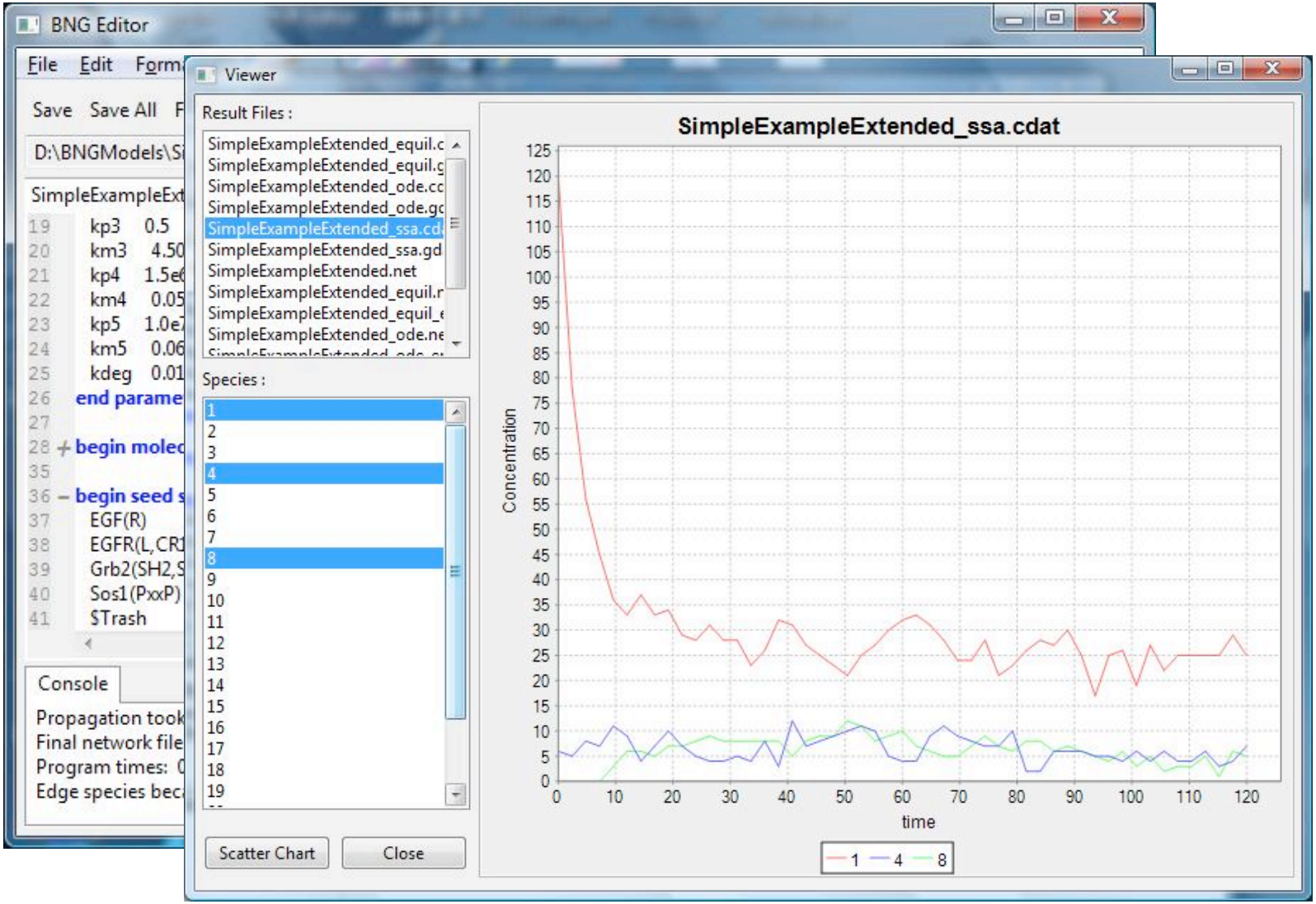


Yao Sun and Liz Marai, U. Pitt Computer Science

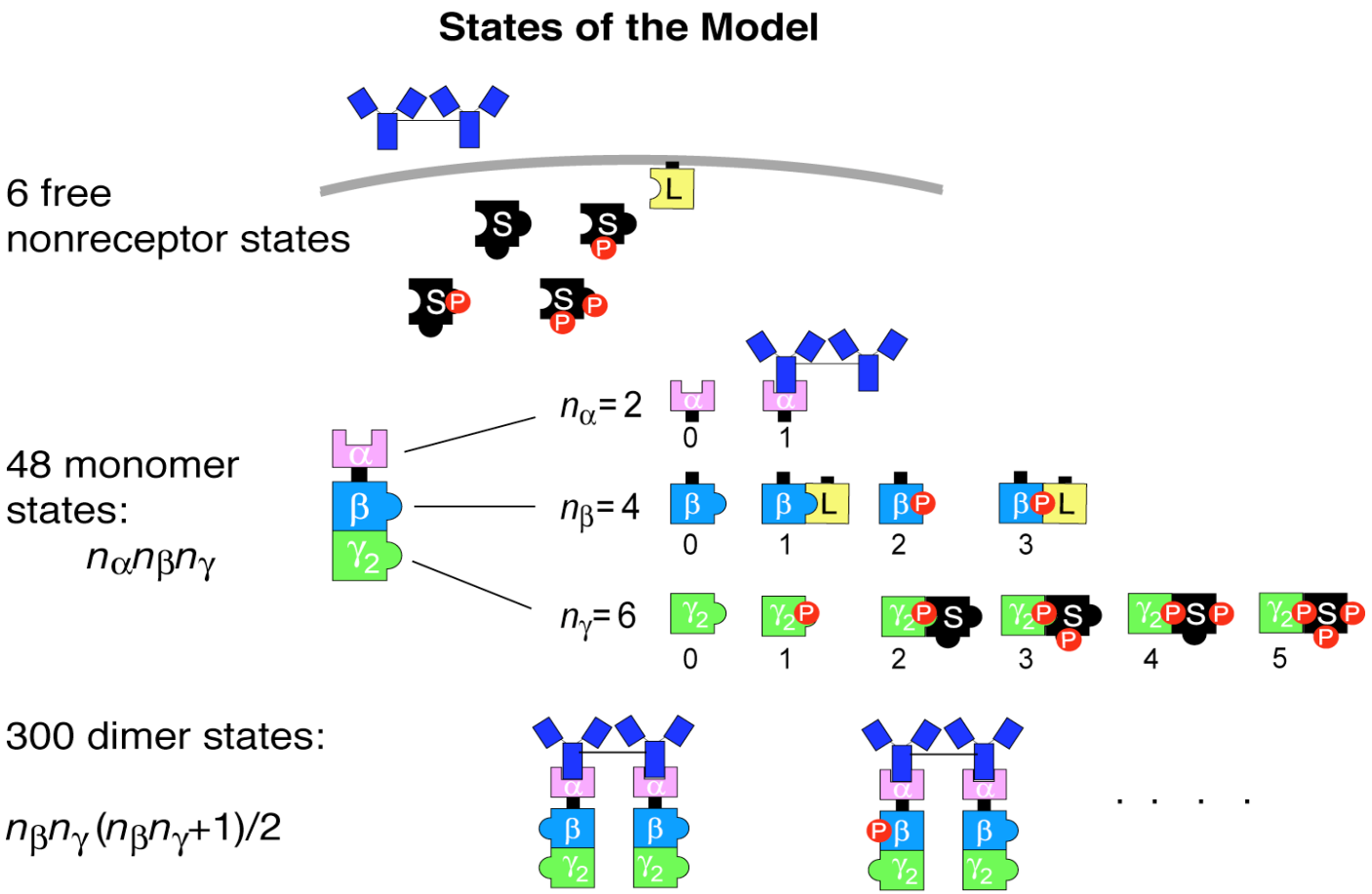
BioNetGen Editor - BiNGE



BioNetGen Editor - BiNGE



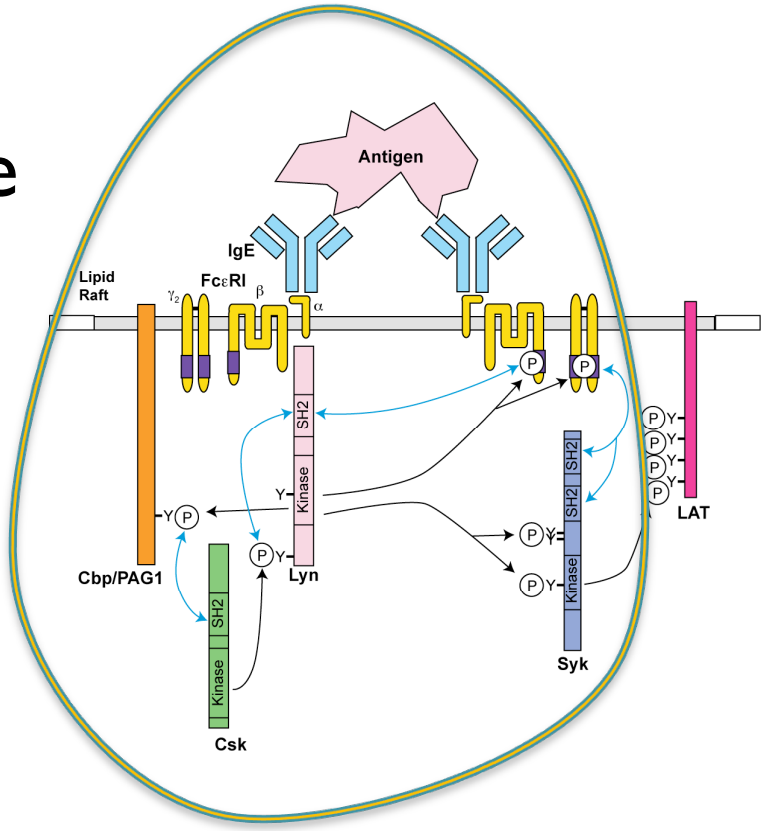
Enumeration of States, aka "Species"



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

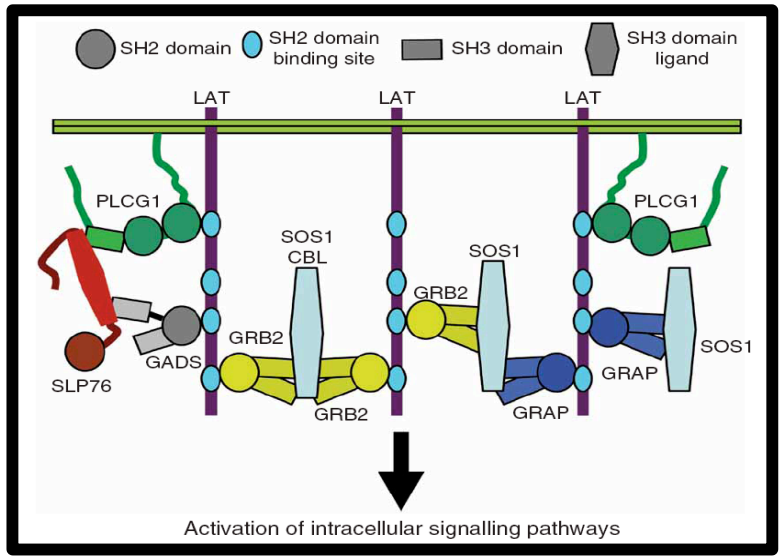
Limits of the network generation approach

- 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.

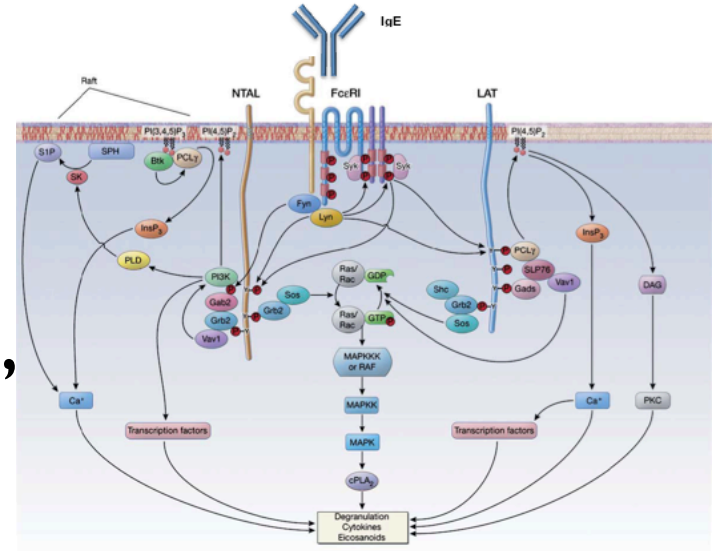


Houtman et al., *Nat. Struct. Mol. Biol.* (2006)

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

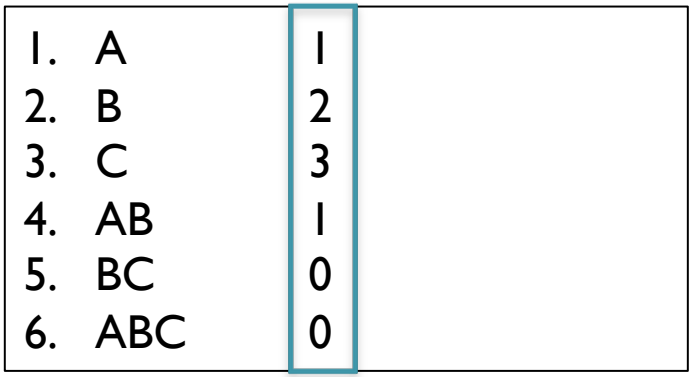
Population- vs. Particle-Based Approaches to Simulation

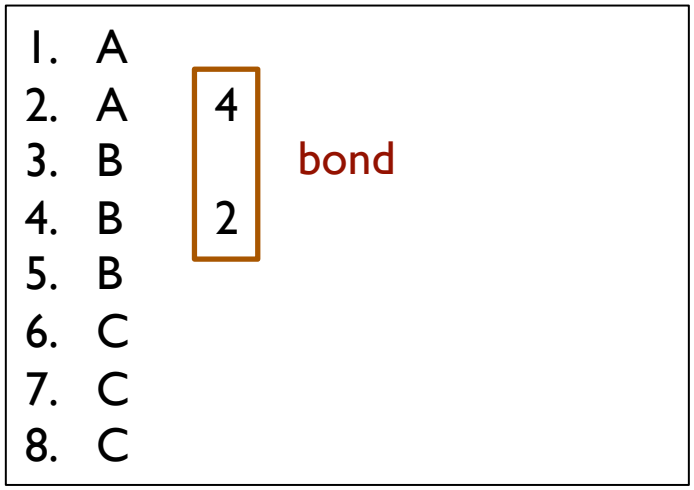
Population

- Each species is enumerated
- Configuration is vector of populations

Particles

- Molecules are instantiated
- Configuration is complex data struct





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!

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

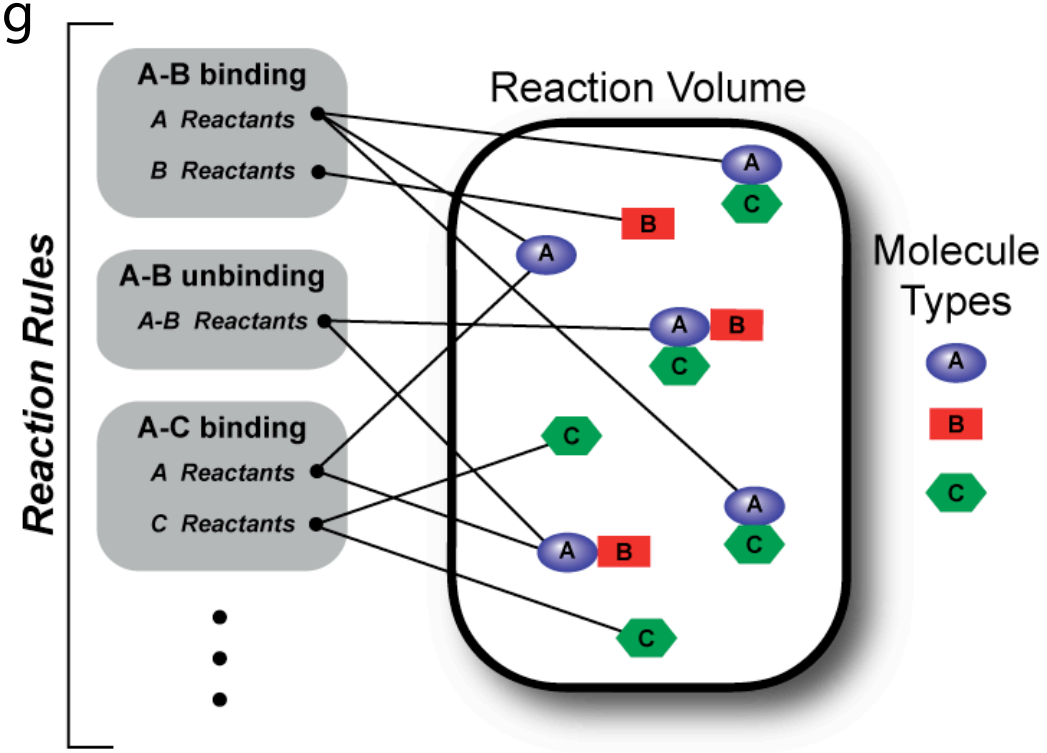
Particles

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

NFSIM

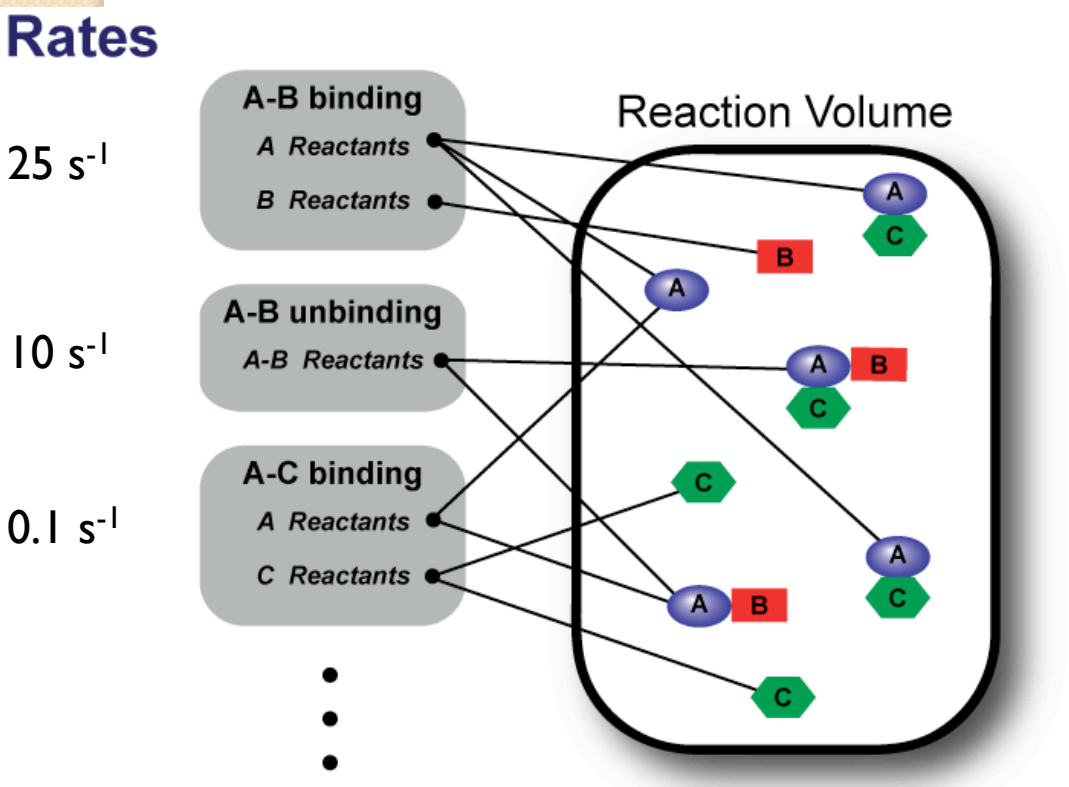
"Network-Free" Stochastic Simulator

- 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)*



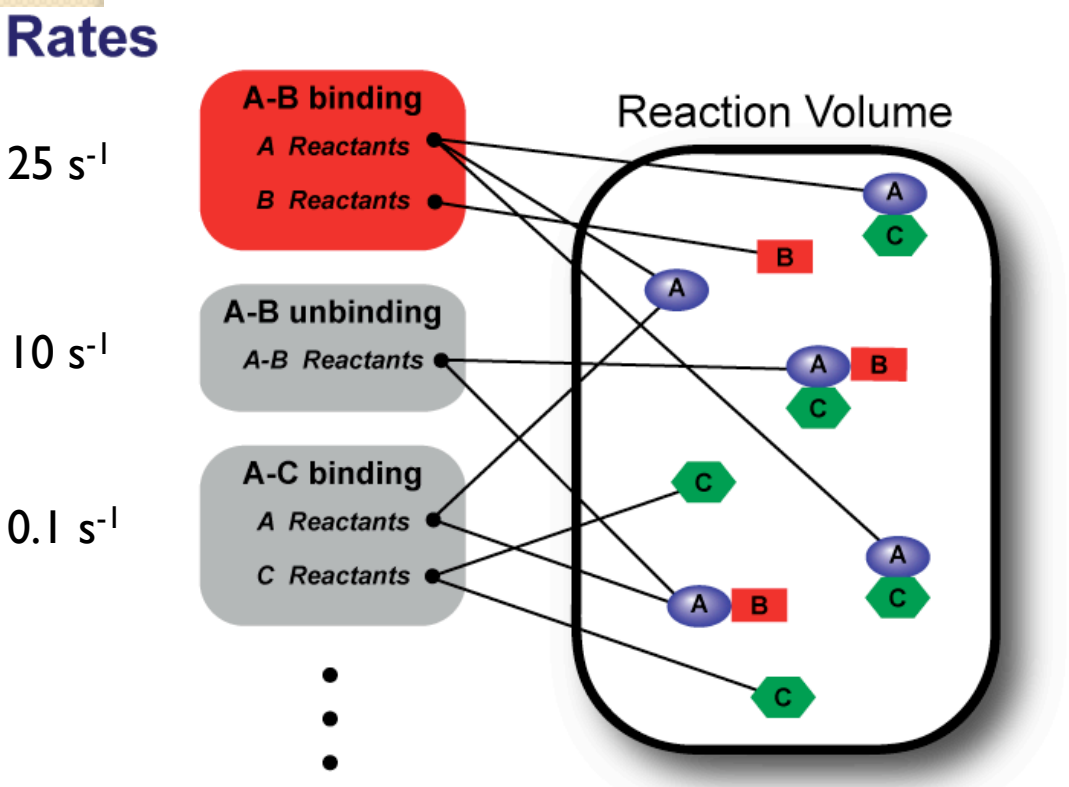
Sneddon, Faeder, and Emonet, in preparation.

NFsim Algorithm



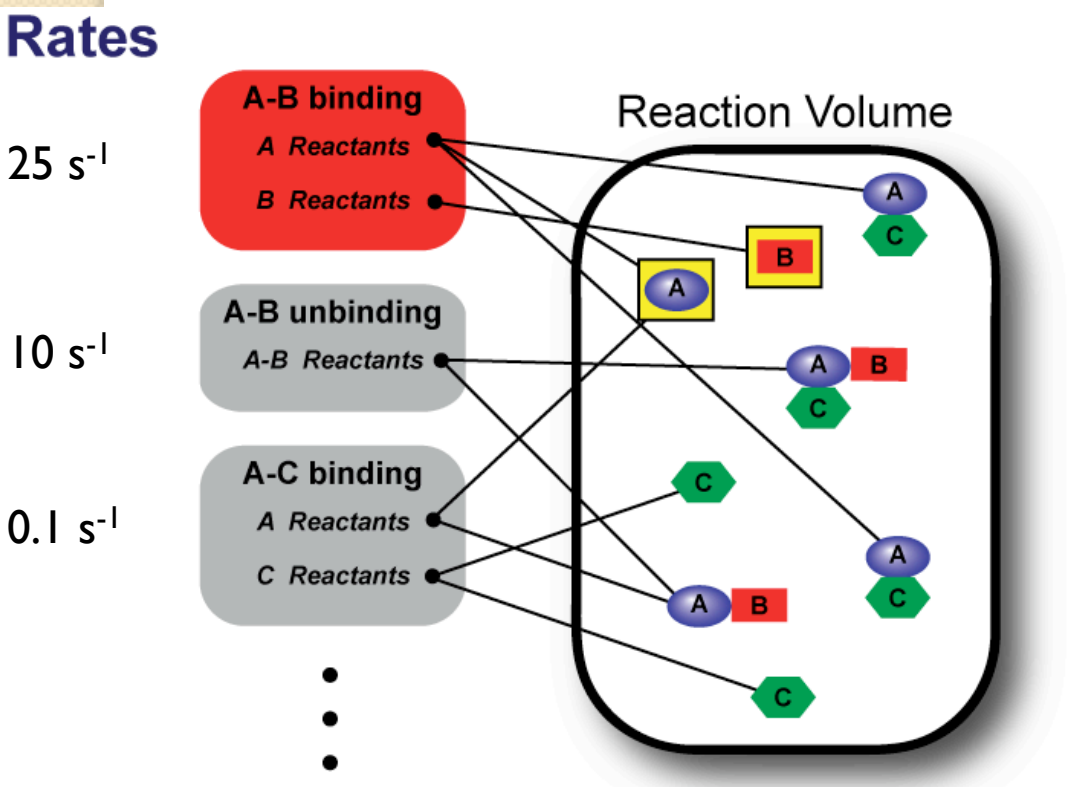
0. **Initialize *reactant lists* and calculate *rule propensities.***
1. Select next reaction time and next *rule*.
2. Select molecules and sites to react.
   a. Check any application condition(s).
3. Apply operation specified by rule.
4. Update reactant lists and propensities.
5. Increment time.

NFsim Algorithm



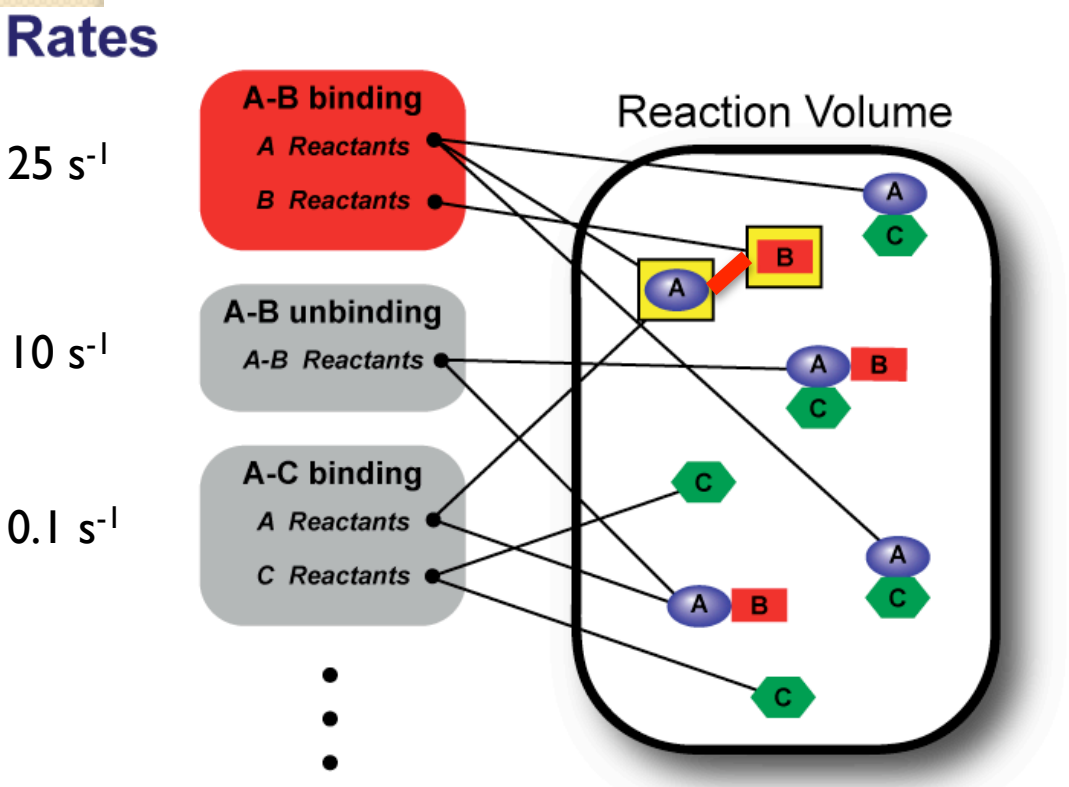
0. Initialize *reactant lists* and calculate *rule propensities.*
1. **Select next reaction time and next *rule*.**
2. Select molecules and sites to react.
   a. Check any application condition(s).
3. Apply operation specified by rule.
4. Update reactant lists and propensities.
5. Increment time.

NFsim Algorithm



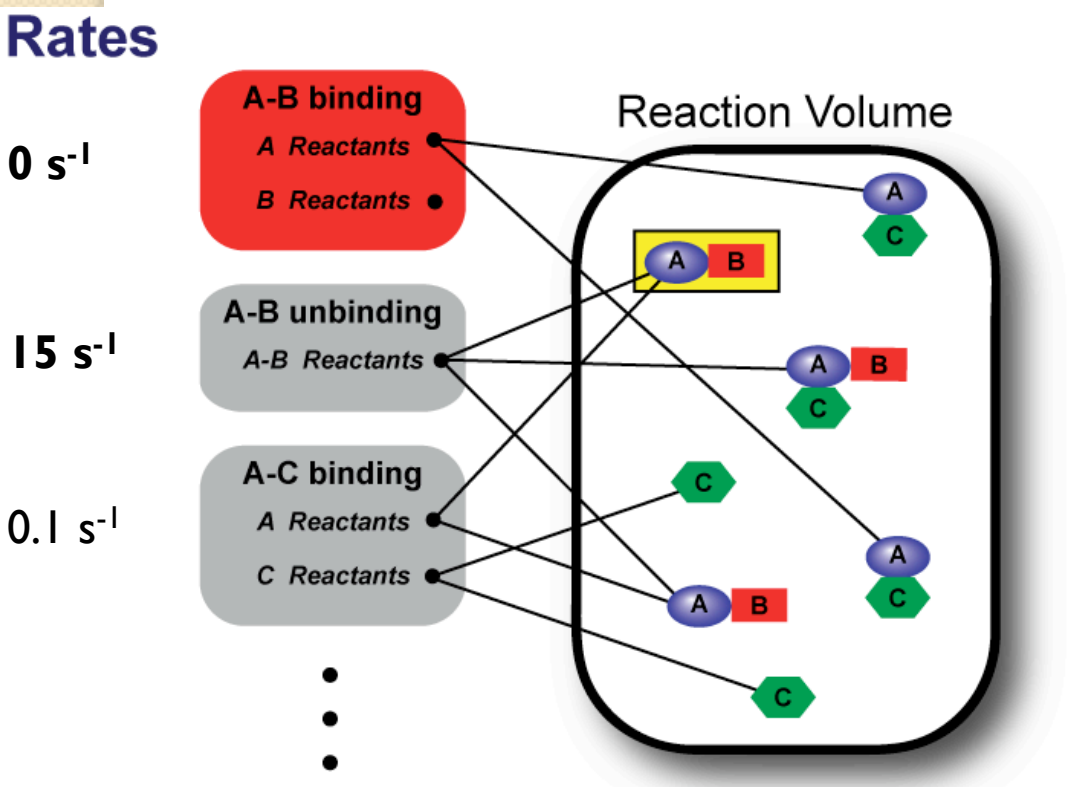
0. Initialize *reactant lists* and calculate *rule propensities.*
1. Select next reaction time and next *rule*.
2. **Select molecules and sites to react.**
   a. **Check any application condition(s).**
3. Apply operation specified by rule.
4. Update reactant lists and propensities.
5. Increment time.

NFsim Algorithm



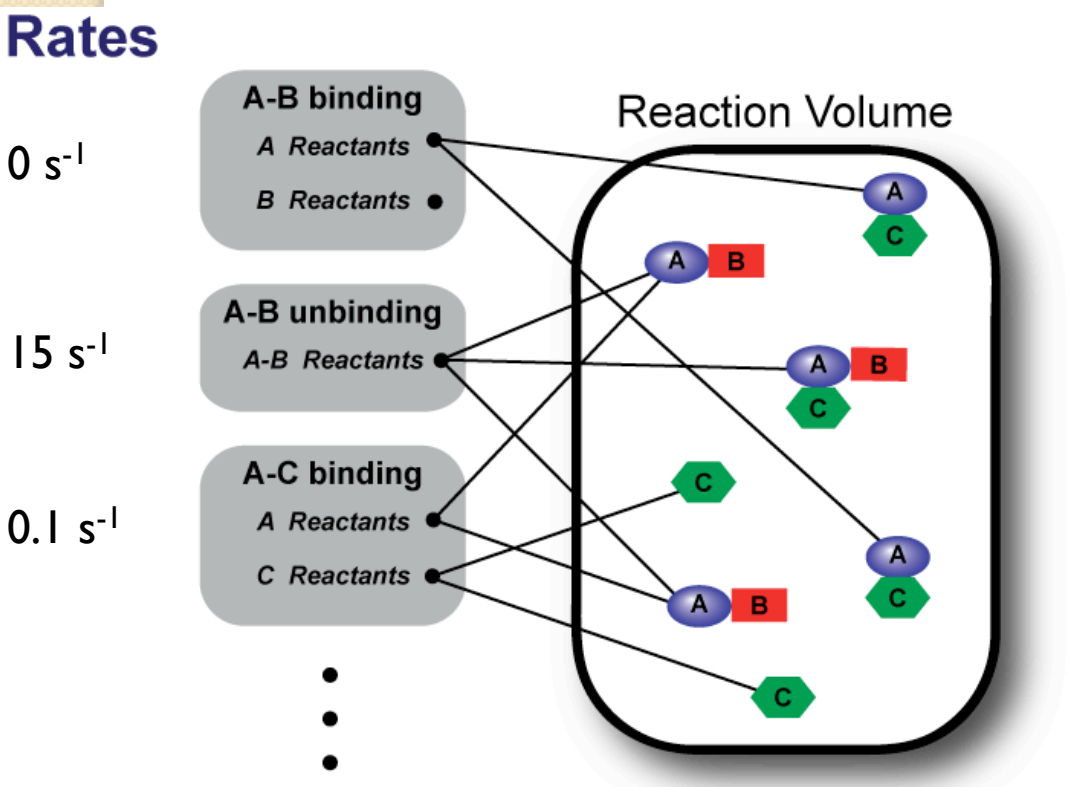
0. Initialize *reactant lists* and calculate *rule propensities.*
1. Select next reaction time and next *rule*.
2. Select molecules and sites to react.
   a. Check any application condition(s).
3. **Apply operation specified by rule.**
4. Update reactant lists and propensities.
5. Increment time.

NFsim Algorithm



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



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.**

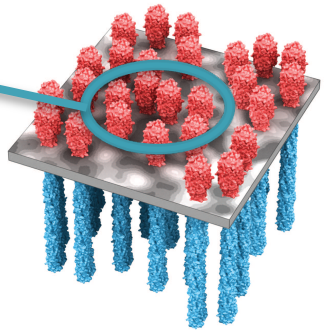
NFsim Core Simulator Features

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

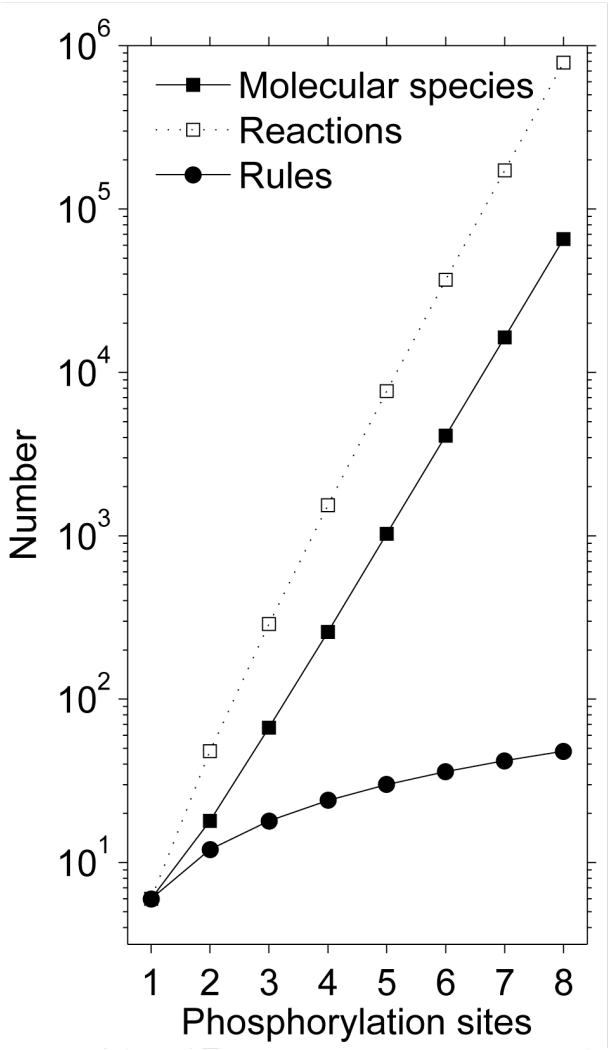


```
MethLevel(x) = 1*R1(x)+2*R2(x)+
     3*R3(x)+4*R4(x)+5*R5(x)+6*R6(x)+
     7*R7(x)+8*R8(x)
```

Multi-site Phosphorylation

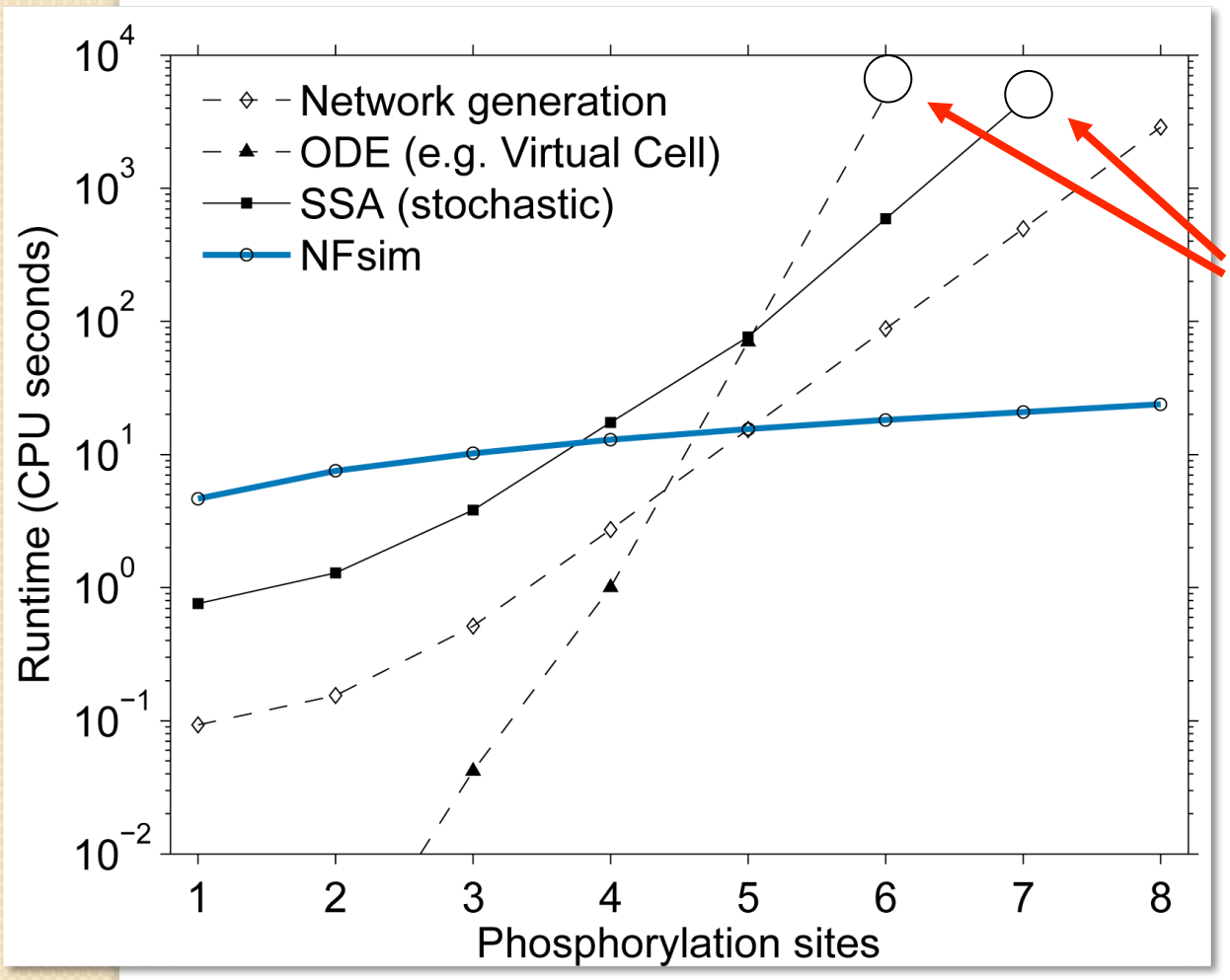
BioNetGen Language [2]

```
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

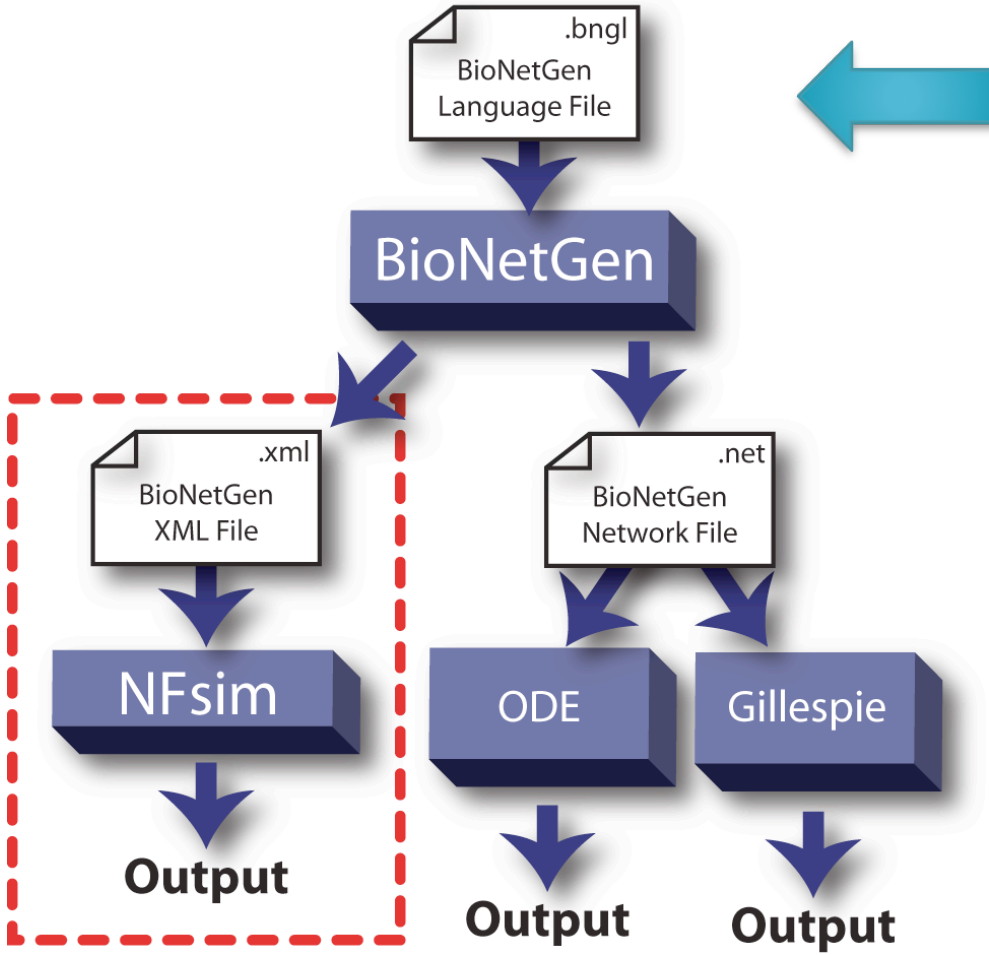
Multi-site Phosphorylation

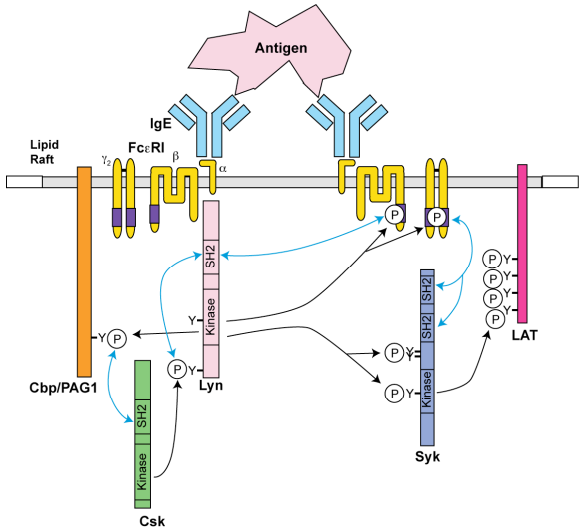


Not possible with ODEs or SSA!

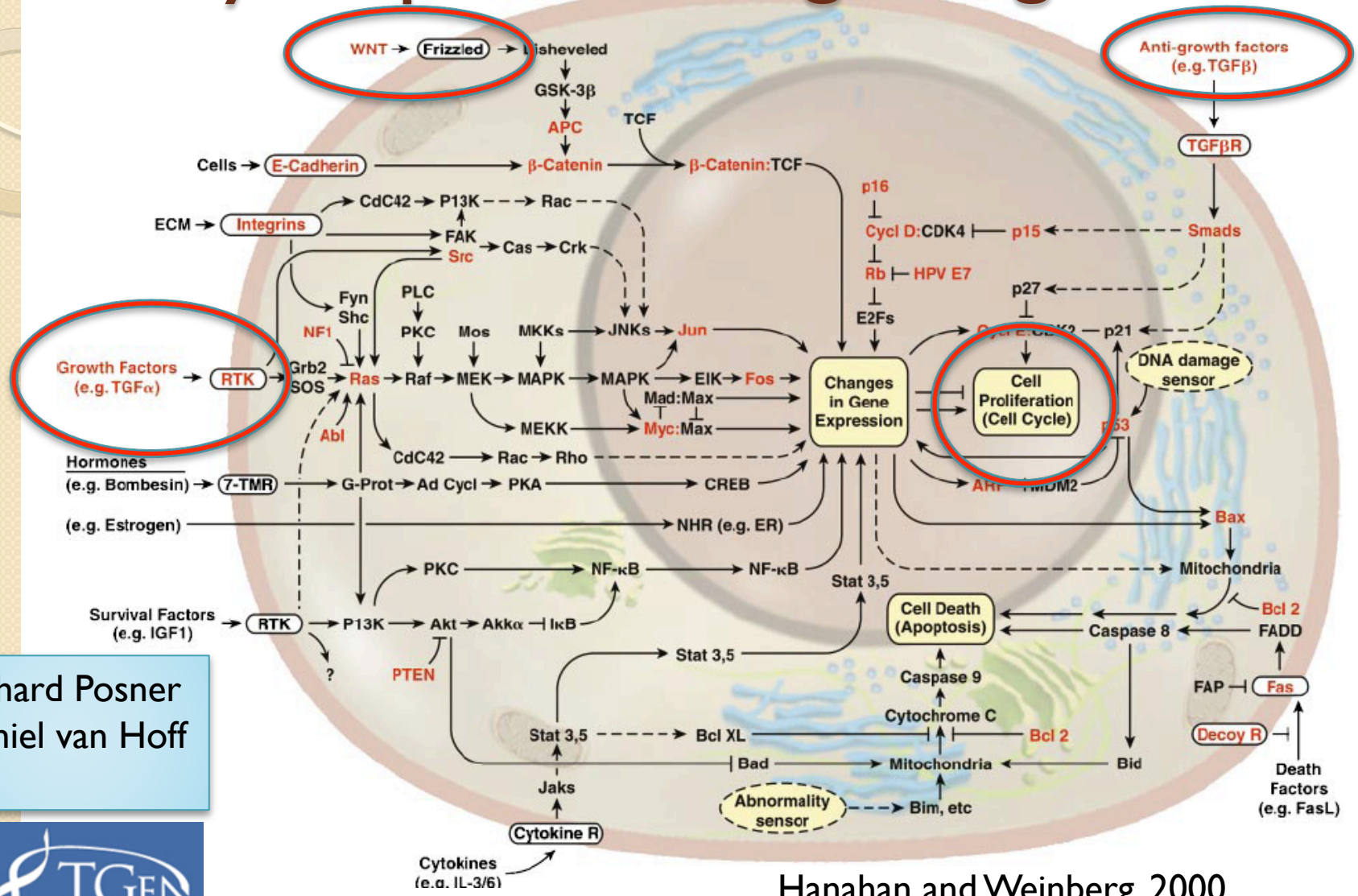
Michael Sneddon and Thierry Emonet

Integration with BioNetGen





Subway Map of Cell Signaling

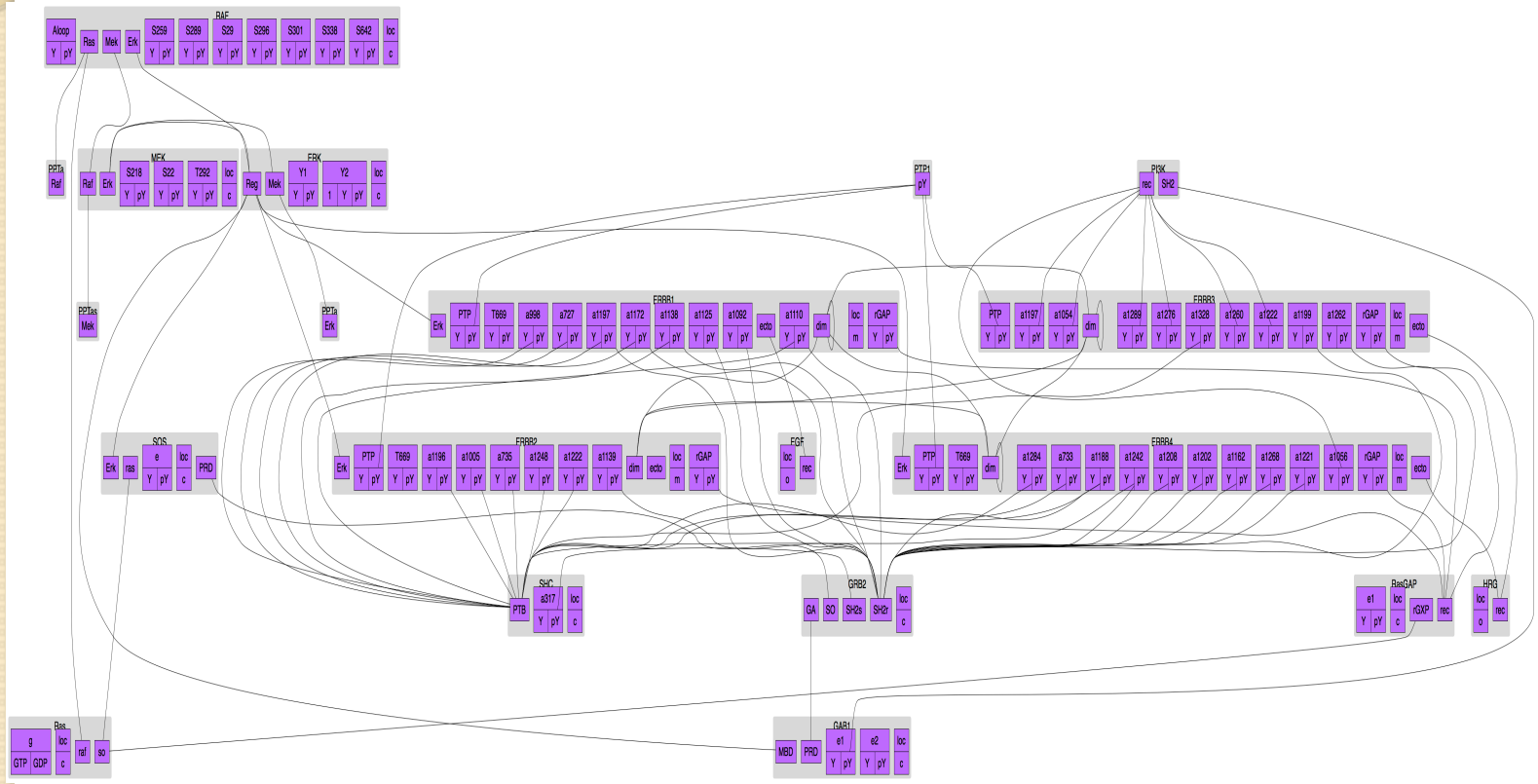


Richard Posner
Daniel van Hoff
…

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

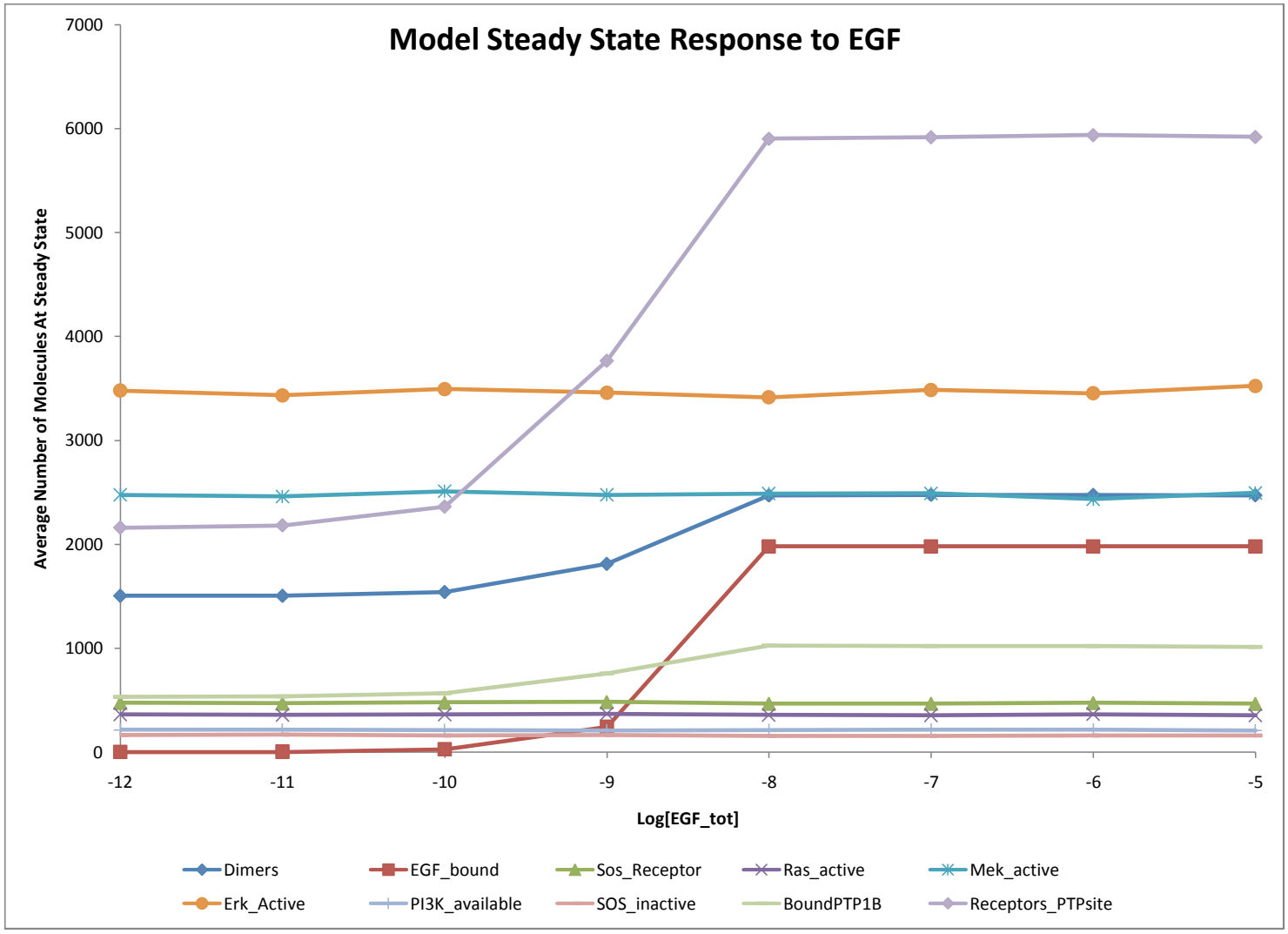
1500 sim sec

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

~ 6e-5 CPU seconds/event

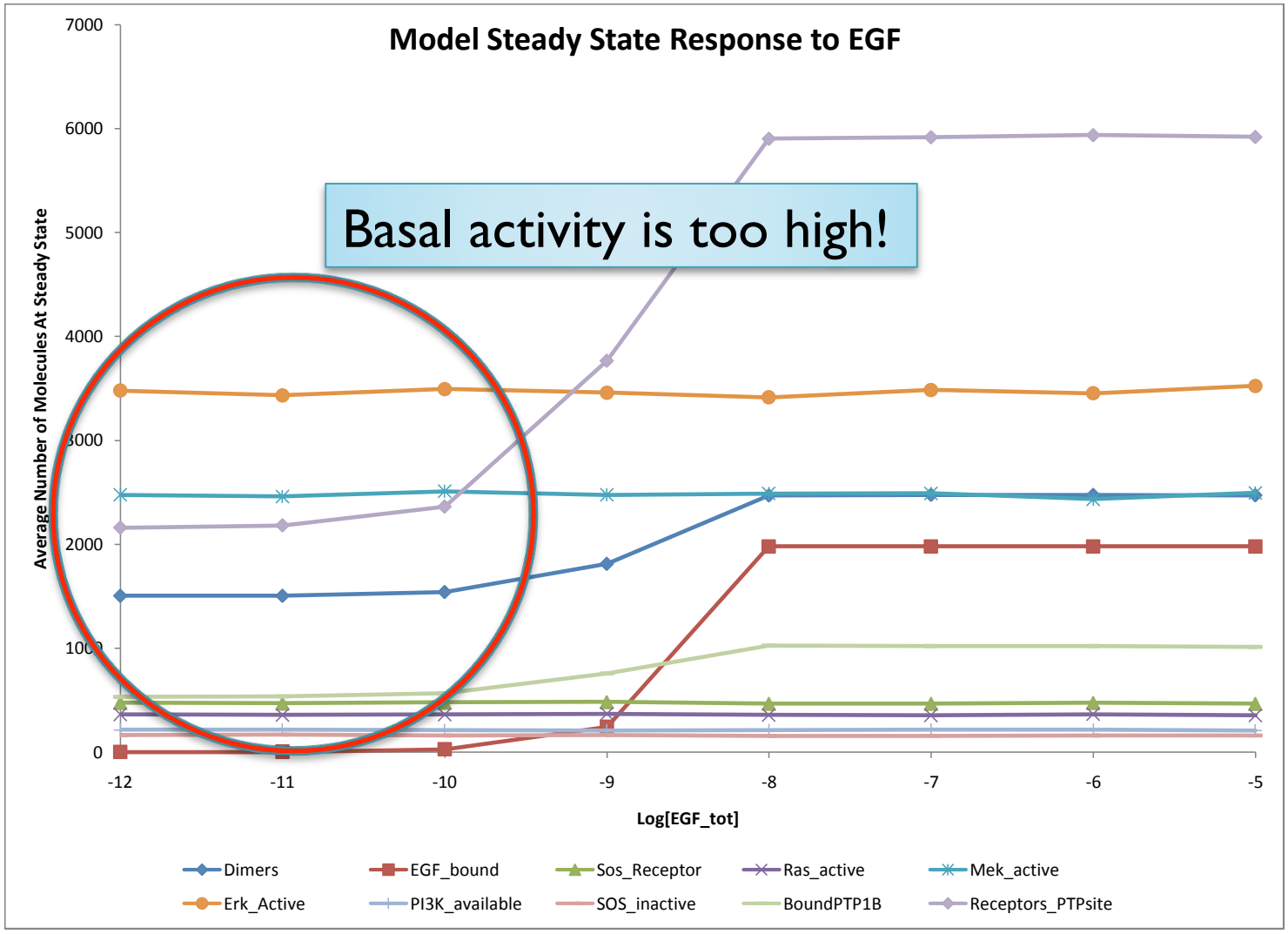
(On a 2.4 GHz Intel Core2Duo on iMac with 4 GB RAM)

Model Validation



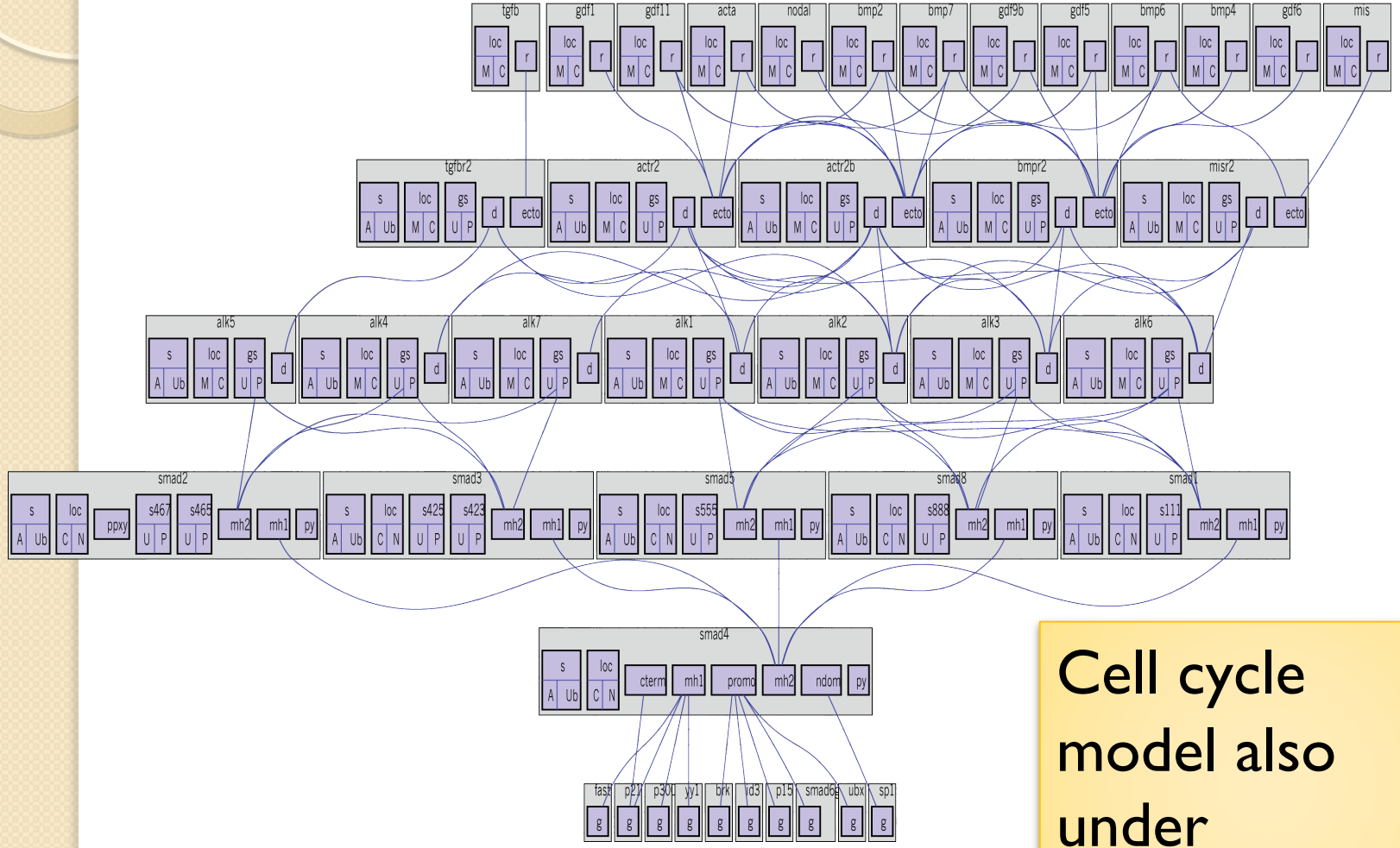
John Sekar

Model Validation



John Sekar

Stop # 2: TGF- β Pathway

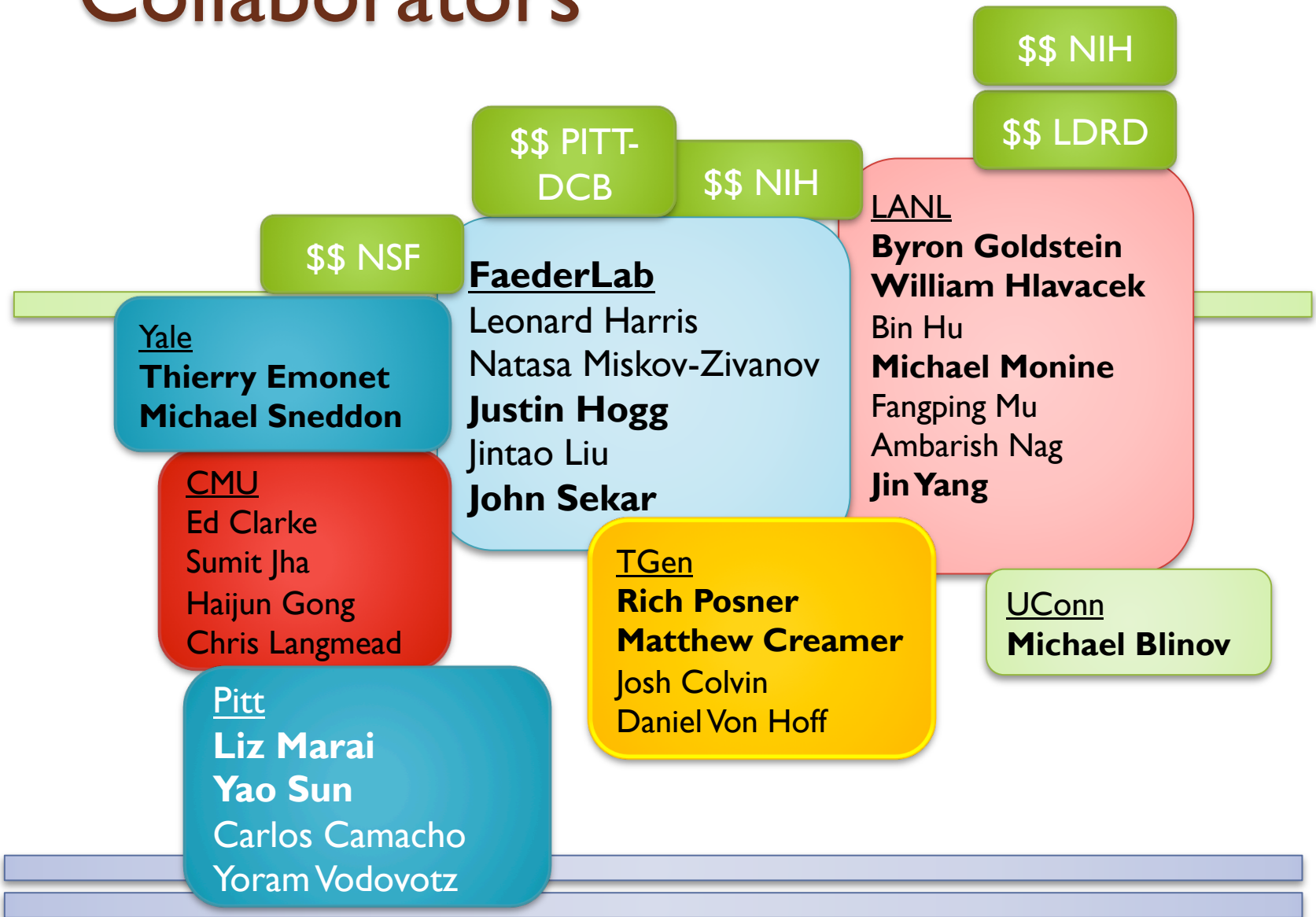


Cell cycle model also under development

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



http://bionetgen.org

Thank You!

Photo by John Sekar