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...
so new construction is required.
How Cells Process Information

Environment

Hormones, growth factors, etc.

Receptors

Nucleus

Architecture of a signaling network

Mutation of Ras Can Produce a Tumor Cell
Ras mutations in cancer

>20% human tumors carry Ras point mutations.

>90% in pancreatic cancer.
Modularity of Signaling Proteins

- Fps
  - SH2
  - KINASE

- Src
  - SH3
  - SH2
  - KINASE

- Syk
  - SH2
  - SH2
  - KINASE

- GAP
  - SH3
  - SH2
  - SH3
  - SH2
  - PH
  - GAP

- PLC-γ
  - PLC
  - SH2
  - SH2
  - SH3
  - PLC

- Grb2
  - SH3
  - SH2
  - SH3

- Nck
  - SH3
  - SH3
  - SH3
  - SH2

Figure 6.10a  The Biology of Cancer (© Garland Science 2007)
Modularity produces complex wiring
Complexity of Receptor Complexes

(A) PDGF-β receptor

(B) EGF receptor

Figure 6.9 *The Biology of Cancer* (© Garland Science 2007)
The “curse” of complexity

<table>
<thead>
<tr>
<th>Number of States</th>
<th>Monomers</th>
<th>Dimers</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3,888</td>
<td>7,560,216</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(A) PDGF-β receptor

(B) EGF receptor

<table>
<thead>
<tr>
<th>Number of States</th>
<th>Monomers</th>
<th>Dimers</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>19,440</td>
<td>188,966,520</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of States</th>
<th>Monomers</th>
<th>Dimers</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>845 YHAE</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>992 YLIP</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1045 YSSD</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1068 YINQ</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1086 YHNK</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1148 YQOD</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1173 YLRV</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(A) PDGF-β receptor

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

**Representation**
- BIO.NETGEN Language
- kappa
- etc.

**Simulation**
- 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(tSH2,1Y~U~P,aY~U~P)
Defining Interaction Rules

**BIONETGEN Language**

$IgE(a,a) + FceRI(a) \leftrightarrow IgE(a,a1).FceRI(a1) \ldots$

**binding and dissociation**

$Lyn(U1).FceRI(b1).FceRI(b_{-U}) \rightarrow Lyn(U1).FceRI(b1).FceRI(b_{-P})$

**component state change**
Rule-based modeling protocol

Objects and rules

**BIONETGEN**

Reaction Network

ODE Solver

Stochastic Simulator (Gillespie)

Output

“Normal Cell”

“Mutants”

http://bionetgen.org
BIONETGEN Editor - BiNGE

Yao Sun and Liz Marai, U. Pitt Computer Science
BIONETGEN Editor - BiNGE

```plaintext
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  ×  egrf_simple.bngl

13  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
28

+ begin molecule types
29

+ begin seed species
36  EGF(R)
37  EGF0
38  EGFR(L1,CR1,Y1,068,-U)  EGFR0
39  Grb2(SH2,SH3)  GRB20
40  Sos1(ProP)  SOS10
41  STriple  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”

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)
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.

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

Particles
- Molecules are instantiated

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AB</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>BC</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ABC</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Population- vs. Particle-Based Approaches to Simulation

<table>
<thead>
<tr>
<th>Population</th>
<th>Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Each species is enumerated</td>
<td>• Molecules are instantiated</td>
</tr>
<tr>
<td>• Configuration is vector of populations</td>
<td>• Configuration is complex data struct</td>
</tr>
<tr>
<td>• Update dependencies can be precomputed</td>
<td>• Update dependencies computed on-the-fly</td>
</tr>
</tbody>
</table>
### Population- vs. Particle-Based Approaches to Simulation

<table>
<thead>
<tr>
<th>Population</th>
<th>Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Each species is enumerated</td>
<td>- Molecules are instantiated</td>
</tr>
<tr>
<td>- Configuration is vector of populations</td>
<td>- Configuration is complex data struct</td>
</tr>
<tr>
<td>- Update dependencies can be precomputed</td>
<td>- Update dependencies computed on-the-fly</td>
</tr>
<tr>
<td>- Single particles cannot be tracked</td>
<td>- Single particles can be tracked</td>
</tr>
</tbody>
</table>
### Population- vs. Particle-Based Approaches to Simulation

<table>
<thead>
<tr>
<th>Population</th>
<th>Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combinatorial complexity can make population-based simulations intractable!</td>
<td></td>
</tr>
<tr>
<td>vector of populations</td>
<td>Molecules are instantiated</td>
</tr>
<tr>
<td>Update dependencies can be precomputed</td>
<td>Configuration is complex data struct</td>
</tr>
<tr>
<td>Single particles cannot be tracked</td>
<td>Update dependencies computed on-the-fly</td>
</tr>
<tr>
<td>Single particles can be tracked</td>
<td></td>
</tr>
</tbody>
</table>
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.
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.
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.
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.
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 BIO.NETGEN

3) Extended BIO.NETGEN Language handles
   1) Spatial compartments
   2) System variables in rate law expressions

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

BioNetGen

- .bngl
  BioNetGen Language File

- .xml
  BioNetGen XML File

- .net
  BioNetGen Network File

NFsim

Output

ODE

Output

Gillespie

Output

Diagram of antigen binding and signal transduction pathways.

Legend:
- IgE: Immunoglobulin E
- FcRI: High-affinity IgE receptor
- Lyn: Lyn tyrosine kinase
- Syk: Syk tyrosine kinase
- Csk: C-terminal Src kinase
- LAT: Linker for Activation of T-Cells
Subway Map of Cell Signaling

Hanahan and Weinberg, 2000
Rule-based Model of EGFR Signaling

Preliminary Model: 20 molecules / 532 rules / 496 parameters

Matt Creamer and Rich Posner
Stats

<table>
<thead>
<tr>
<th>Model</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 20 Molecule Types</td>
<td>1500 sim sec</td>
</tr>
<tr>
<td>– 4 Receptors</td>
<td>• ~10-18 million events</td>
</tr>
<tr>
<td>– 3 Ligands</td>
<td>• ~1060 real sec</td>
</tr>
<tr>
<td>• 536 Parameters</td>
<td>~ 6e-5 CPU seconds/event</td>
</tr>
<tr>
<td>• 547 Reaction Rules</td>
<td>(On a 2.4 GHz Intel Core2Duo on iMac with 4 GB RAM)</td>
</tr>
</tbody>
</table>
Model Validation

Model Steady State Response to EGF

John Sekar
Model Validation

Model Steady State Response to EGF

Basal activity is too high!
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)
Thank You!

Photo by John Sekar