simulation

Why do we simulate?

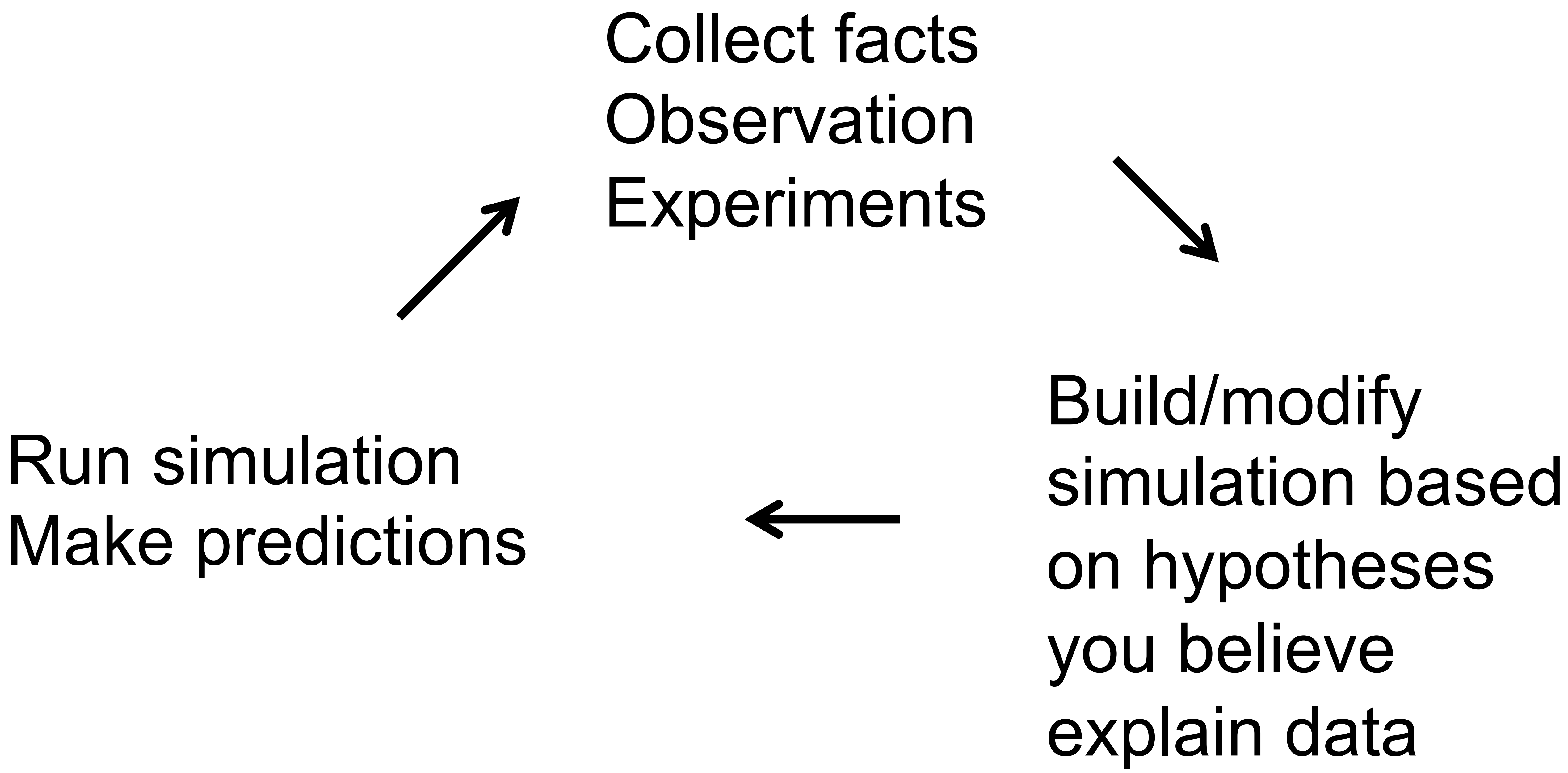
Questions biologists ask

- **What are the principles governing biological systems?**
- How is something (a gene, an organism) evolutionarily related to something else?
- How does something (molecule, protein complex, cell, organism) interact with something else?

What can we simulate?

Where do we need tools?

The Simulation Cycle

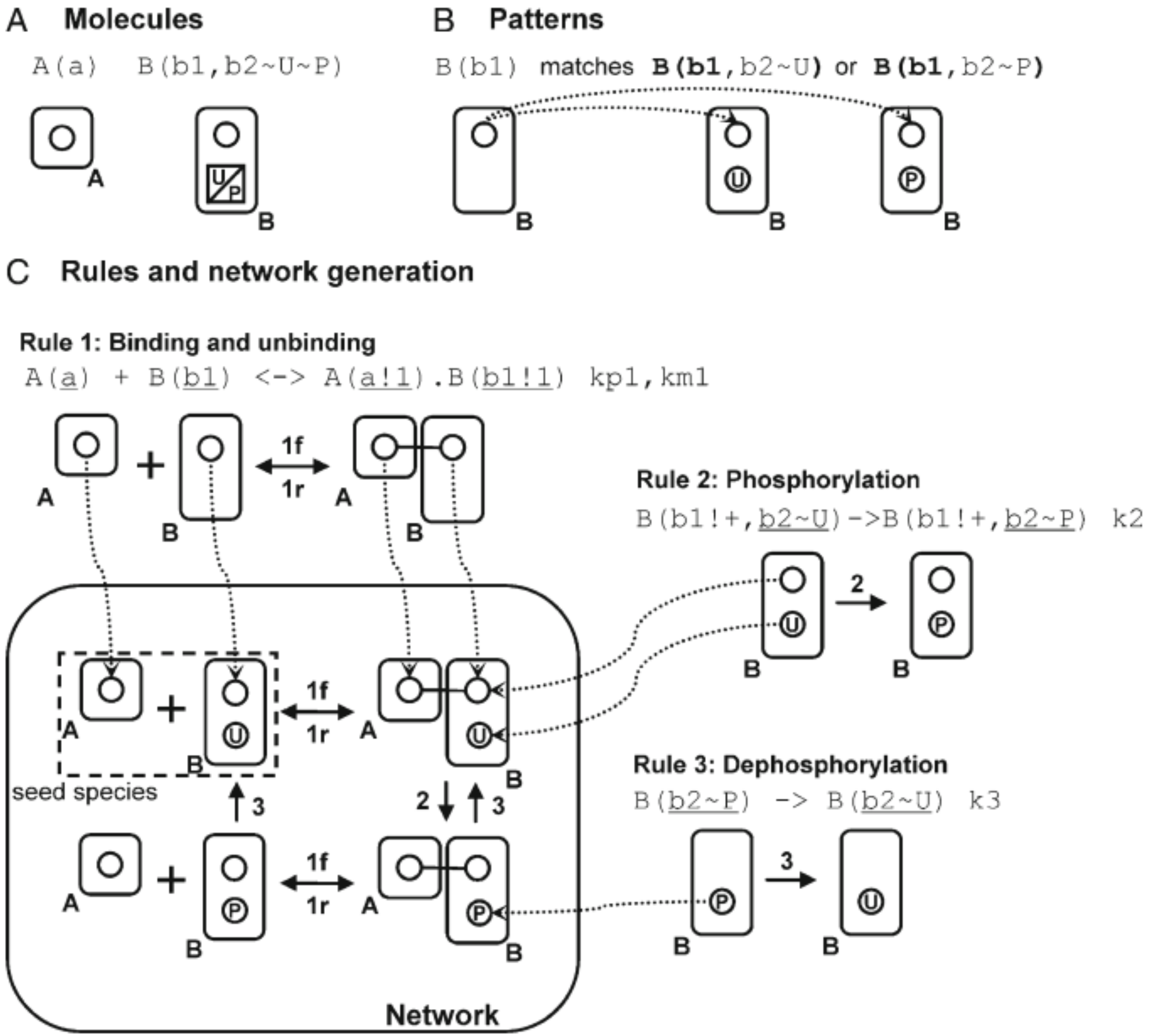


Collect facts
Observation
Experiments

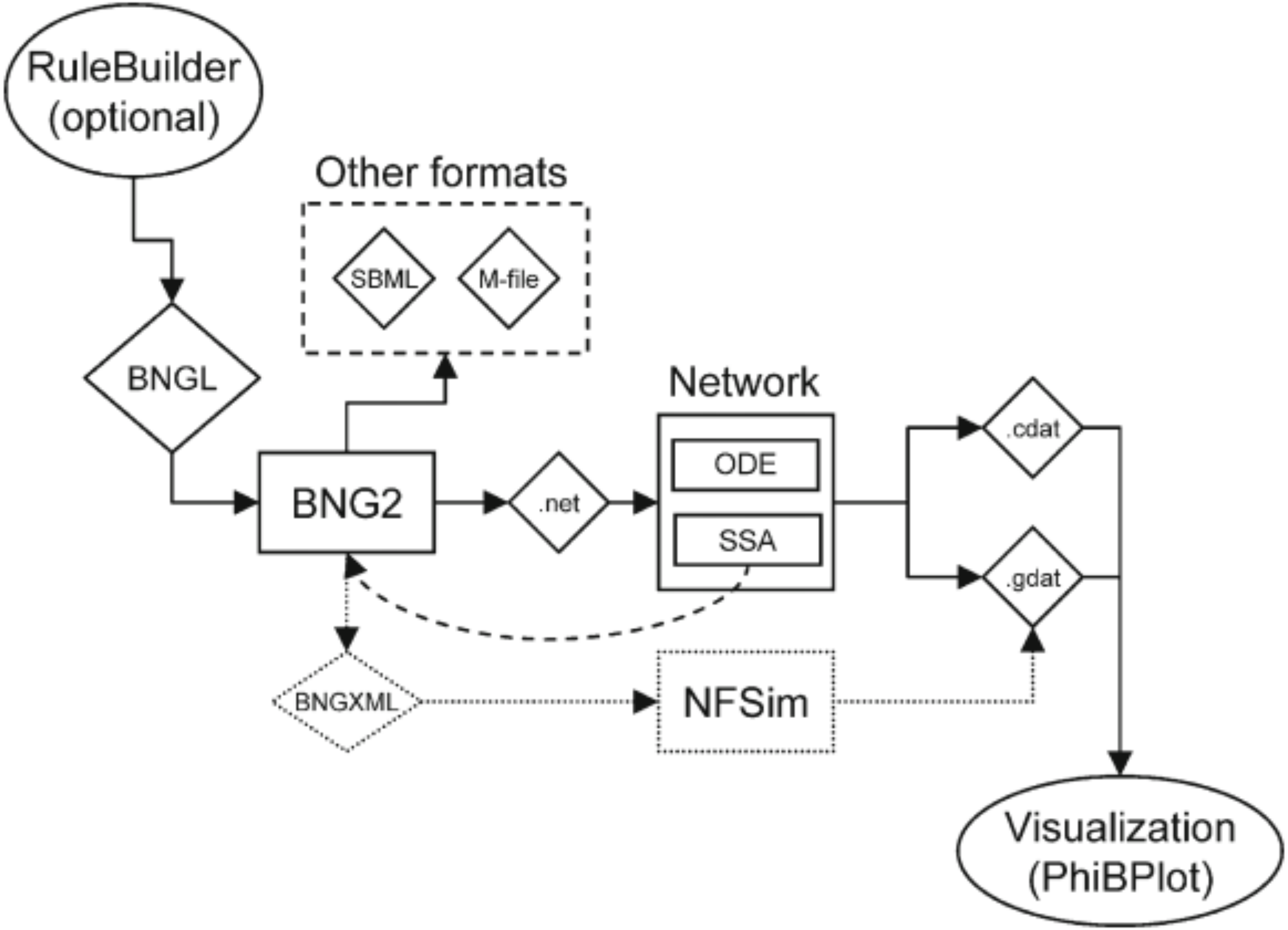
Build/modify
simulation based
on hypotheses
you believe
explain data

Run simulation
Make predictions

BioNetGen



```
A(a)  B(b1,b2~U~P)
B(b1) matches B(b1,b2~U) or B(b1,b2~P)
A(a) + B(b1) <-> A(a!1).B(b1!1) kp1,km1
B(b1!+,b2~U)->B(b1!+,b2~P) k2
B(b2~P) -> B(b2~U) k3
```


RuleBuilder (optional)
Other formats
SBML
M-file
BNGL
Network
.cdat
ODE
BNG2
.net
SSA
.gdat
BNGXML
NFSim
Visualization (PhiBPlot)

```
begin parameters
  NA 6.02e23 # Avogadro's number (molecues/mol)
  f  1       # Fraction of the cell to simulate
  Vo f*1.0e-10 # Extracellular volume=1/cell_density (L)
  V  f*3.0e-12 # Cytoplasmic volume (L)
  # Inital amount of ligand (20 nM)
  EGF_init 20*1e-9*NA*Vo # convert to copies per cell
  # Initial amounts of cellular components (copies per cell)
  EGFR_init     f*1.8e5
  Grb2_init     f*1.5e5
  Sos1_init     f*6.2e4
  # Rate constants
  # Divide by NA*V to convert bimolecular rate constants
  # from /M/sec to /(molecule/cell)/sec
  kp1 9.0e7/(NA*Vo) # ligand-monomer binding
  km1 0.06         # ligand-monomer dissociation
  kp2 1.0e7/(NA*V) # aggregation of bound monomers
  km2 0.1          # dissociation of bound monomers
  kp3 0.5          # dimer transphosphorylation
  km3 4.505        # dimer dephosphorylation
  kp4 1.5e6/(NA*V) # binding of Grb2 to receptor
  km4 0.05         # dissociation of Grb2 from receptor
  kp5 1.0e7/(NA*V) # binding of Grb2 to Sos1
  km5 0.06         # dissociation of Grb2 from Sos1
  deg 0.01         # degradation of receptor dimers
end parameters
```

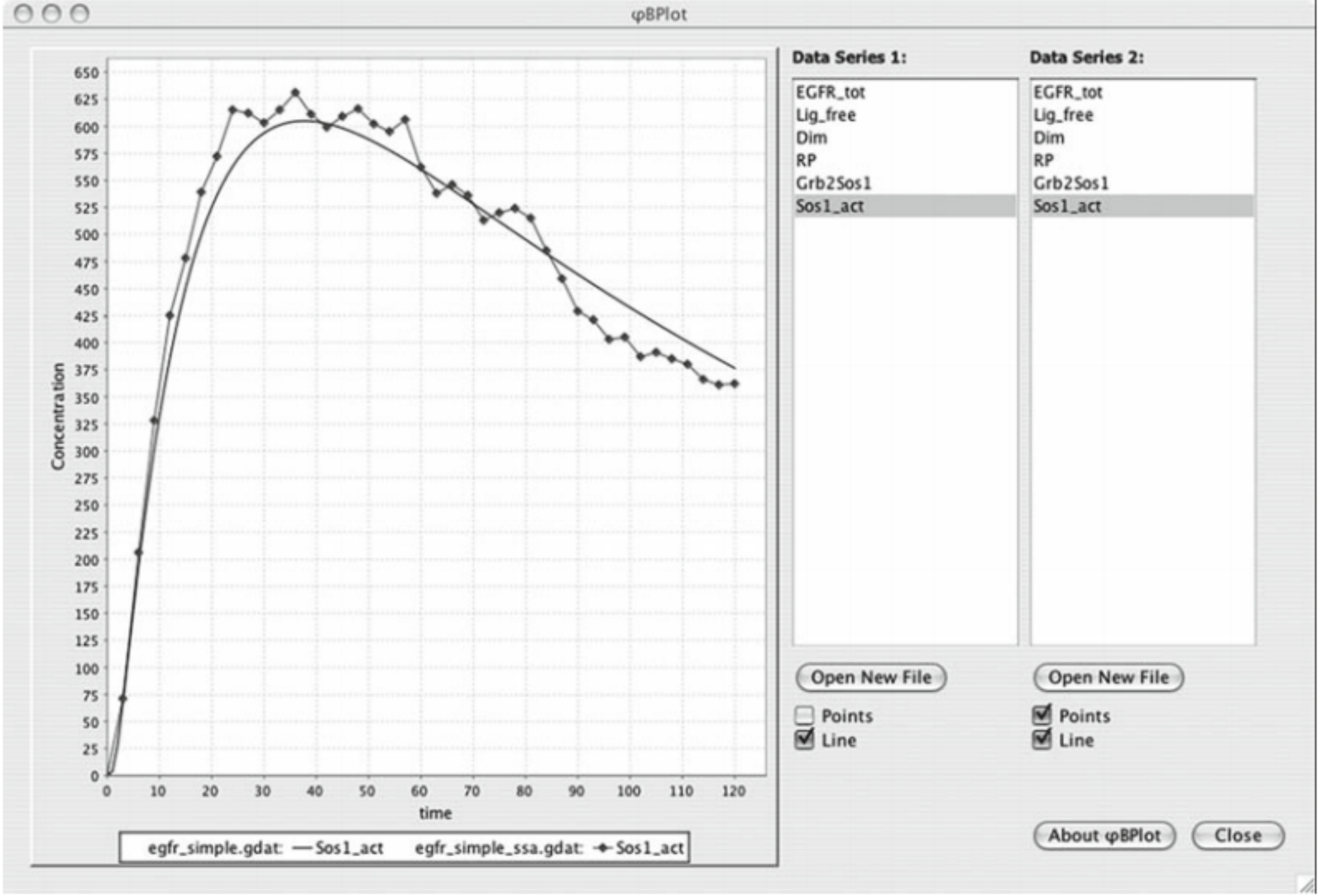
```
begin molecule types
  EGF(R)
  EGFR(L,CR1,Y1068~U~P)
  Grb2(SH2,SH3)
  Sos1(PxxP)
  Trash()
end  molecule types

begin seed species
  EGF(R)                 0
  EGFR(L,CR1,Y1068~U) EGFR_init
  Grb2(SH2,SH3)         Grb2_init
  Sos1(PxxP)            Sos1_init
end seed species

begin observables
  1 Molecules  EGFR_tot   EGFR()
  2 Molecules  Lig_free   EGF(R)
  3 Species    Dim        EGFR(CR1!+)
  4 Molecules  RP EGFR(Y1068~P!?)
  # Cytosolic Grb2-Sos1
  5 Molecules  Grb2Sos1  Grb2(SH2,SH3!1).Sos1(PxxP!1)
  6 Molecules  Sos1_act
EGFR(Y1068!1).Grb2(SH2!1,SH3!2).Sos1(PxxP!2)
end observables
```

```
begin reaction rules
  # Ligand-receptor binding
  1 EGFR(L,CR1) + EGF(R) <-> EGFR(L!1,CR1).EGF(R!1) kp1, km1
  # Receptor-aggregation
  2 EGFR(L!+,CR1) + EGFR(L!+,CR1) <-> EGFR(L!+,CR1!1).EGFR(L!+,CR1!1) kp2,km2
  # Transphosphorylation of EGFR by RTK
  3 EGFR(CR1!+,Y1068~U) -> EGFR(CR1!+,Y1068~P)  kp3
  # Dephosphorylation
  4 EGFR(Y1068~P) -> EGFR(Y1068~U) km3
  # Grb2 binding to pY1068
  5 EGFR(Y1068~P) + Grb2(SH2) <-> EGFR(Y1068~P!1).Grb2(SH2!1) kp4,km4
  # Grb2 binding to Sos1
  6 Grb2(SH3) + Sos1(PxxP) <-> Grb2(SH3!1).Sos1(PxxP!1) kp5,km5
  # Receptor dimer internalization/degradation
  7 EGF(R!1).EGF(R!2).EGFR(L!1,CR1!3).EGFR(L!2,CR1!3) -> Trash() deg\
  DeleteMolecules
end reaction rules
```

```
#actions
generate_network({overwrite=>1});
# Equilibration
simulate_ode({suffix=>equil,t_end=>100000,n_steps=>10,sparse=>1,\
    steady_state=>1});
setConcentration(“EGF(R)”,”EGF_init”);
saveConcentrations(); # Saves concentrations for future reset
# Kinetics
writeSBML({});
simulate_ode({t_end=>120,n_steps=>120});
resetConcentrations(); # reverts to saved Concentrations
simulate_ssa({suffix=>ssa,t_end=>120,n_steps=>120});
```



Data Series 1: Sos1_act

☐ Points

☑ Line

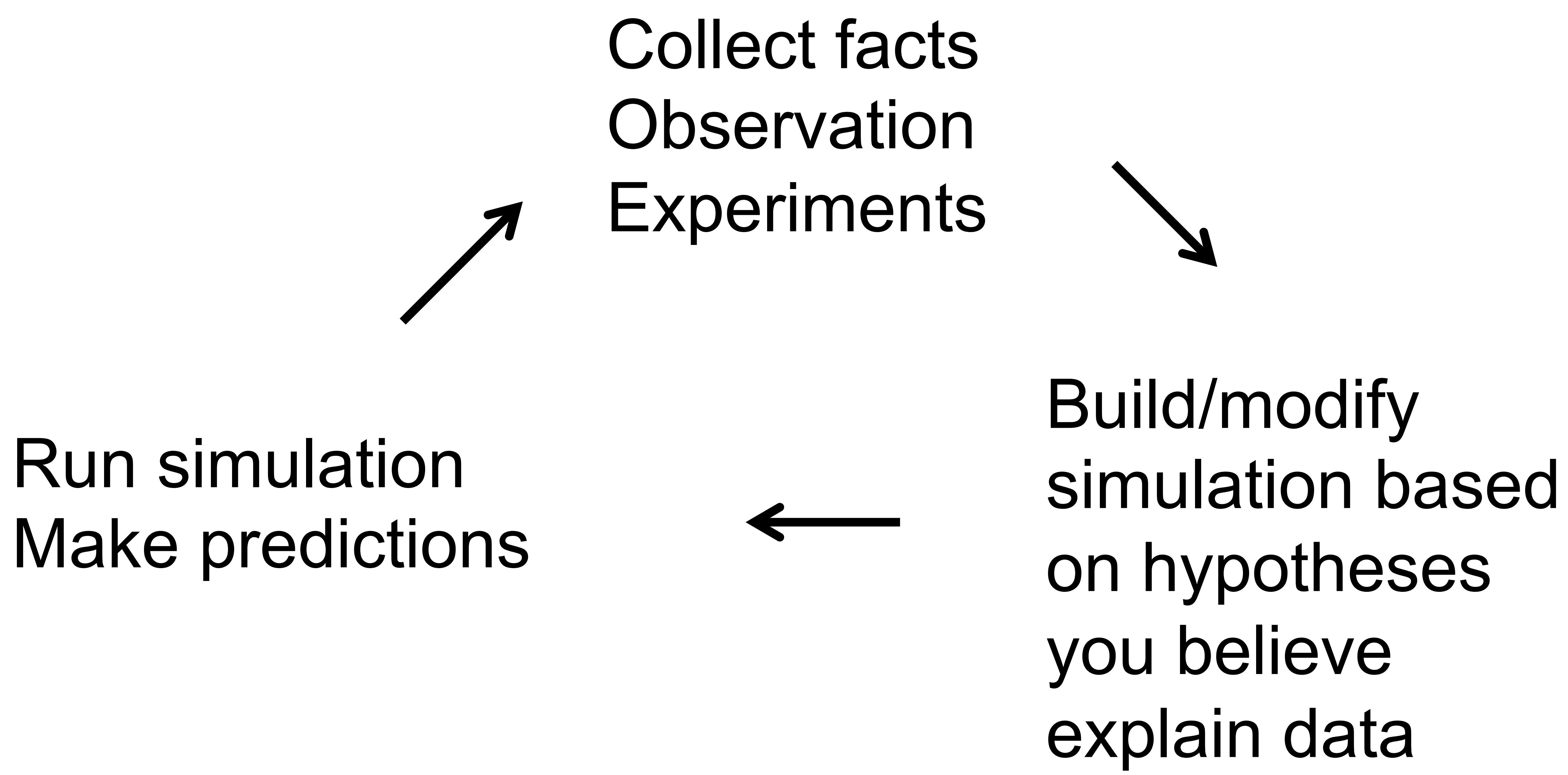
Data Series 2: Sos1_act

☑ Points

☑ Line

critique

The Simulation Cycle



Collect facts
Observation
Experiments

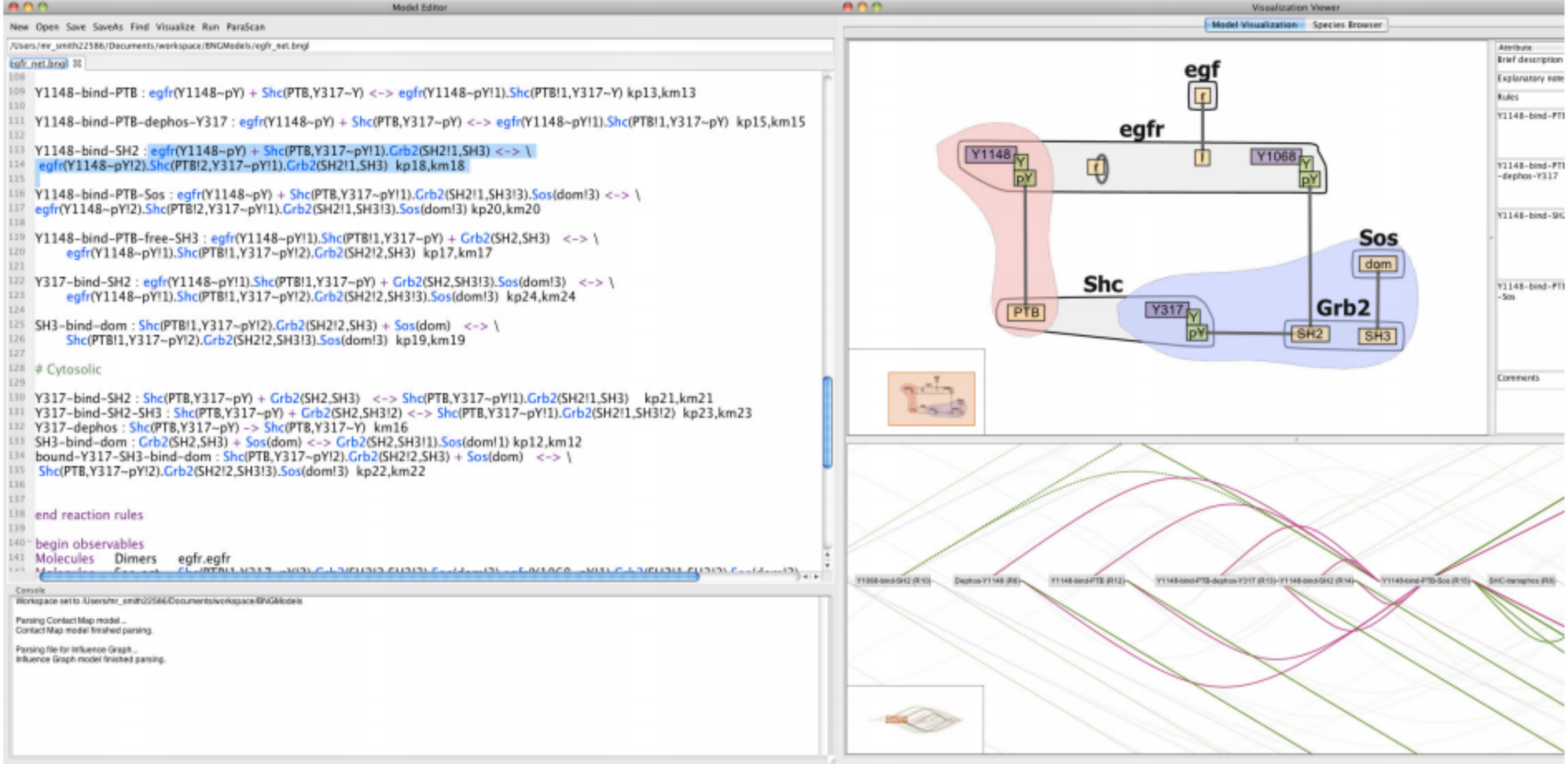
Build/modify
simulation based
on hypotheses
you believe
explain data

Run simulation
Make predictions

RuleBender

Table 1 RBM Tasks and RuleBender Scores

Index	Task	Score (1 to 5)
T1	Compose a model from scratch.	4.2
T2	Find and correct an error in a model.	4.8
T3	Understand relationships between rules in the model - do they have overlapping reactants, products, etc.?	4.4
T4	Modify an existing model and run simulations to compare results with those of the original.	4.2
T5	Generate a network; examine species and reactions.	4.4
T6	Run a parameter scan. Examine overall results and look at results for individual trajectories.	4.8
T7	Compare results of scanning a parameter in two different models.	4.4
T8	Find a set of parameters that makes the model behave in a specific way.	3.4



```
Y1148-bind-PTB : egfr(Y1148~pY) + Shc(PTB,Y317~Y) <-> egfr(Y1148~pY!1).Shc(PTB!1,Y317~Y) kp13,km13

Y1148-bind-PTB-dephos-Y317 : egfr(Y1148~pY) + Shc(PTB,Y317~pY) <-> egfr(Y1148~pY!1).Shc(PTB!1,Y317~pY)  kp15,km15

Y1148-bind-SH2 : egfr(Y1148~pY) + Shc(PTB,Y317~pY!1).Grb2(SH2!1,SH3) <-> \
 egfr(Y1148~pY!2).Shc(PTB!2,Y317~pY!1).Grb2(SH2!1,SH3)  kp18,km18

Y1148-bind-PTB-Sos : egfr(Y1148~pY) + Shc(PTB,Y317~pY!1).Grb2(SH2!1,SH3!3).Sos(dom!3) <-> \
egfr(Y1148~pY!2).Shc(PTB!2,Y317~pY!1).Grb2(SH2!1,SH3!3).Sos(dom!3) kp20,km20

Y1148-bind-PTB-free-SH3 : egfr(Y1148~pY!1).Shc(PTB!1,Y317~pY) + Grb2(SH2,SH3)   <-> \
      egfr(Y1148~pY!1).Shc(PTB!1,Y317~pY!2).Grb2(SH2!2,SH3)  kp17,km17

Y317-bind-SH2 : egfr(Y1148~pY!1).Shc(PTB!1,Y317~pY) + Grb2(SH2,SH3!3).Sos(dom!3)   <-> \
      egfr(Y1148~pY!1).Shc(PTB!1,Y317~pY!2).Grb2(SH2!2,SH3!3).Sos(dom!3)  kp24,km24

SH3-bind-dom : Shc(PTB!1,Y317~pY!2).Grb2(SH2!2,SH3) + Sos(dom)   <-> \
      Shc(PTB!1,Y317~pY!2).Grb2(SH2!2,SH3!3).Sos(dom!3)  kp19,km19

# Cytosolic

Y317-bind-SH2 : Shc(PTB,Y317~pY) + Grb2(SH2,SH3)   <-> Shc(PTB,Y317~pY!1).Grb2(SH2!1,SH3)    kp21,km21
Y317-bind-SH2-SH3 : Shc(PTB,Y317~pY) + Grb2(SH2,SH3!2) <-> Shc(PTB,Y317~pY!1).Grb2(SH2!1,SH3!2)  kp23,km23
Y317-dephos : Shc(PTB,Y317~pY) -> Shc(PTB,Y317~Y)  km16
SH3-bind-dom : Grb2(SH2,SH3) + Sos(dom) <-> Grb2(SH2,SH3!1).Sos(dom!1) kp12,km12
bound-Y317-SH3-bind-dom : Shc(PTB,Y317~pY!2).Grb2(SH2!2,SH3) + Sos(dom)   <-> \
 Shc(PTB,Y317~pY!2).Grb2(SH2!2,SH3!3).Sos(dom!3)  kp22,km22


end reaction rules

begin observables
Molecules    Dimers    egfr.egfr
```

```
Workspace set to /Users/mr_smith22586/Documents/workspace/BNGModels

Parsing Contact Map model ...
Contact Map model finished parsing.

Parsing file for Influence Graph ...
Influence Graph model finished parsing.
```

Figure 1 The RuleBender interface. Shown are the Model Editor pane including console for text output (left) and the Visualization Viewer pane (right). The Visualization Viewer shows two complementary visual encodings corresponding to the text model in the Editor: the interactive contact map (top), and part of the influence graph for this model (bottom). RuleBender's main features include syntax checking, syntax highlighting, visual global model exploration with linked views, integrated execution, support for multiple simulation modules, simulation journaling, interactive plotting including comparison of multiple datasets, and parameter scanning.

Shc
Y317
Y
pY
PTB
egfr
Y1148
Y
pY
I
r
Y1068
Y
pY
egf
r
Grb2
SH2
SH3
Sos
dom
CD4
I
Null
HIV
0
L
A
HIV

```
egfr(r!2,Y1148~pY!1).Shc(PTB!1,Y317~Y) ->  egfr(r!2,Y1148~pY!1).Shc(PTB!1,Y317~pY)  kp14
```

```
CD4(HIV~A) -> CD4(HIV~A) + HIV() pi
```

Figure 2 The Contact Map. Contact Maps without (left) and with (right) hub nodes. Molecules are represented as larger nodes (light gray) while domains and domain states (yellow, orange and purple) are represented as smaller sub-nodes in the molecules. State nodes (green and dark gray) are adjacent to the domain sites to which they apply. Reaction rules are mapped to edges (rules that indicate the creation or destruction of a bond between these two domains) and state nodes (rules that indicate state changes). Selecting a state-node (red boundary on the left) lists all rules that indicate that state change. Similarly, selecting an edge (not shown) lists all rules that create or destroy bonds between the linked domains. Selecting one rule from such a list marks the reaction context in blue and the reaction center in pink. Hub nodes are associated with rules that define molecular level interactions without domains involved, such as the degradation of proteins. Selecting a hub node lists all rules involving the linked molecules as shown on the right.

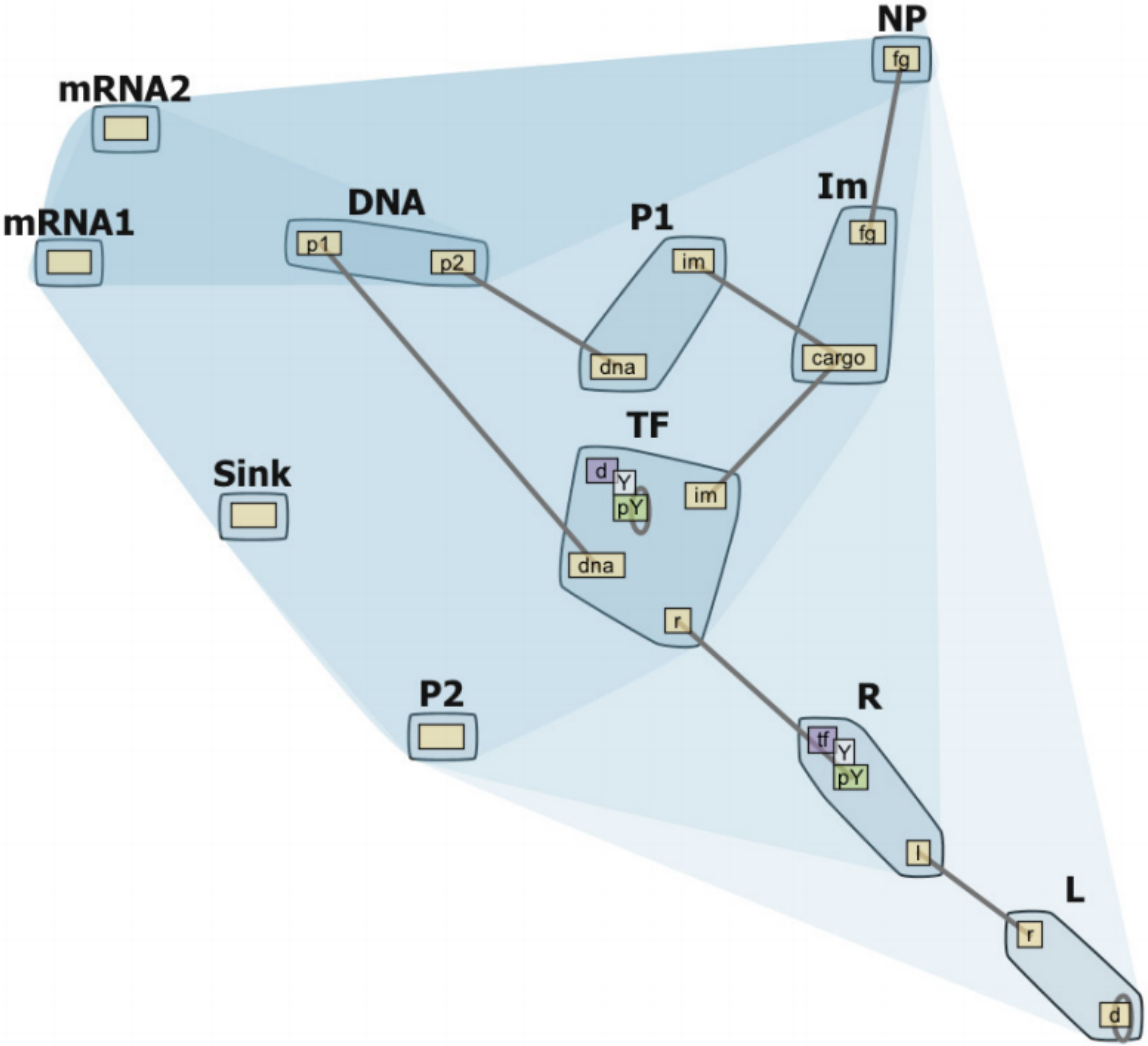


Figure 3 Compartmental Contact Map. Contact Map with molecule compartment hierarchy (extracellular, cytoplasmic, nucleus etc). The saturation of the convex hull encompassing a compartment indicates the hierarchical structure of the compartments; the outermost compartment is colored the lightest blue. All the members of a compartment can be moved as a whole unit to get a clear view of the hierarchical structure.

LRbind(R1)
LRbind(R1')
Rdim(R2)
Rdim(R2')
AbindR(R3)
AbindR(R3')
dephos_mem(R7)
AbindK(R4)
AbindK(R4')
KtransP_inact(R5)
KtransP_act(R6)
dephos_cyt(R8)

Figure 4 The influence graph. Nodes represent reaction rules while arcs represent influence between rules. Green/Magenta solid arcs represent fully activation/inhibition, and Green/Magenta dash arcs represent partial activation/inhibition. Filter options that show or hide activation/ inhibition are provided through pop-up menus. Two separate groups of rule nodes (group1: the first four nodes, group2: the rest of the nodes). can indicate that the model is not complete.

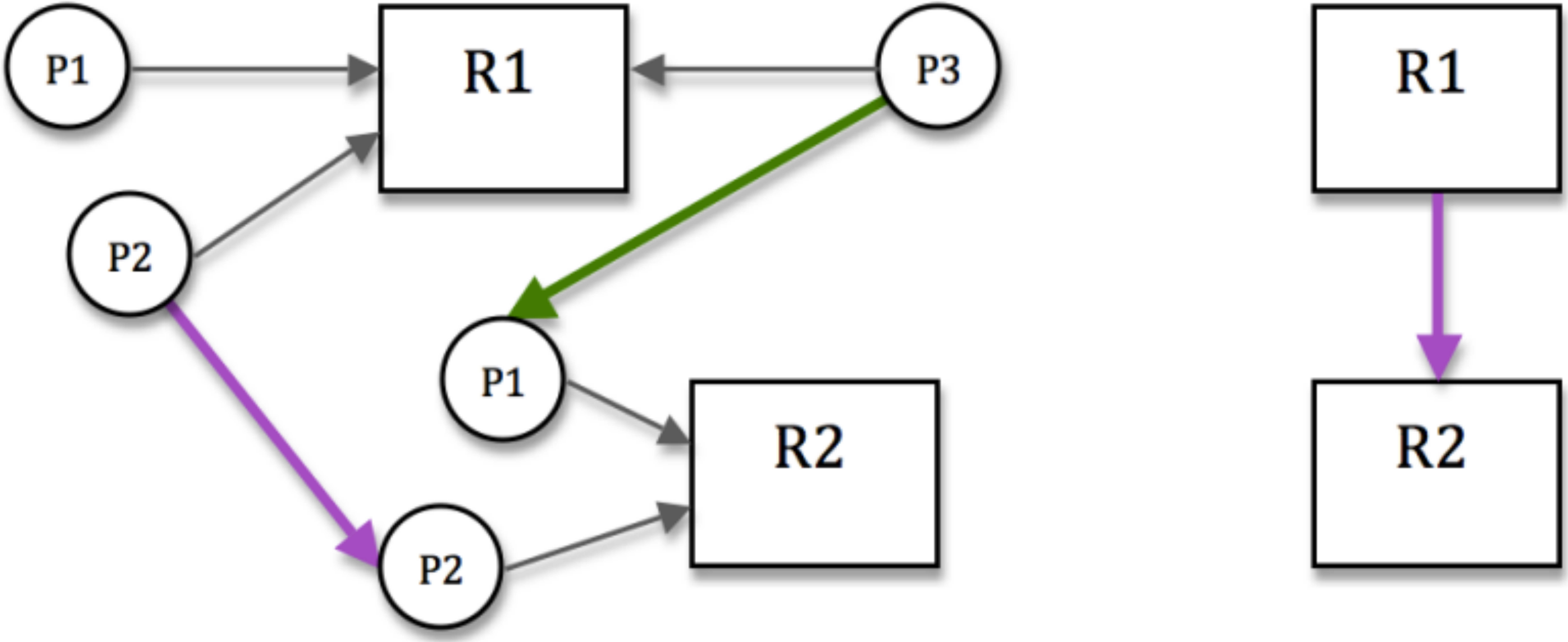


Figure 5 Influence graph definition. Prototype pattern relations (P) and rule relations (R) used to determine influence graphs: an intermediate graph (Left) is ultimately reduced to the simplified, final influence graph (Right). An arrow from P to R means that P is a reactant pattern of the R; for the reverse direction P is a product pattern of R. Green edges show activation relations and red ones show inhibition.

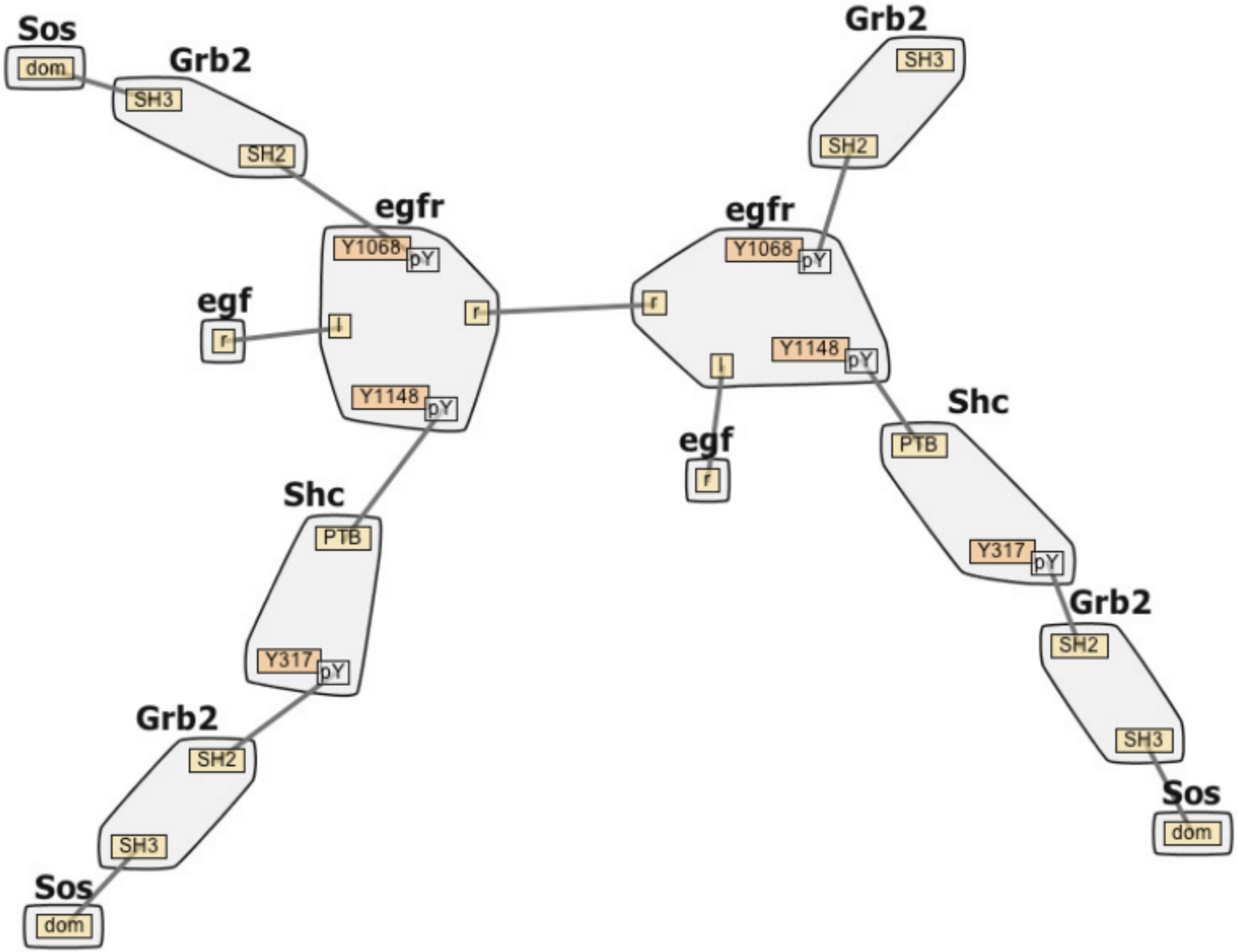


Figure 7 The species graph. The species graph is constructed similarly to the Contact Map. Shown is an example of a complex species containing thirteen molecules which is difficult to grasp from the text representation only.

L
R
Lyn
r
b
SH2
r
l
U
(a)
L
R
Lyn
r
b
SH2
r
l
U
(b)
...
Lyn
R
L
R
Lyn
SH2
b
r
b
SH2
U
l
r
l
U
...
(c)

Figure 9 Lyn-binding debugging. Reduced view: Ligand notation shortened to L and Rec shortened to R. If the user programs the rule tha binds Lyn to Rec incorrectly (see Table 2), the corresponding contact map in (a) is missing the rule context information. The correct binding leads instead to the visualization in (b); the presence of the blue bubble set alerted the researcher to the difference and allowed them to debug their RBM. The incorrect formulation would allow at run time for the creation of the infinitely binding chain shown in (c).

Mosbie

Assignment 1

Many design options…

Show grid?

Mark types/ attributes?

Initial state?

Rules?

# Triggers?

Stopping conditions

Interactions?

2d grid,
3d, nd?

animation?