

Rule-based Modeling of Signal Transduction Systems

<http://vcell.org/bionetgen>

[http://www.santafe.edu/
Rule-Based Modeling of Biochemical
Systems](http://www.santafe.edu/Rule-Based%20Modeling%20of%20Biochemical%20Systems)



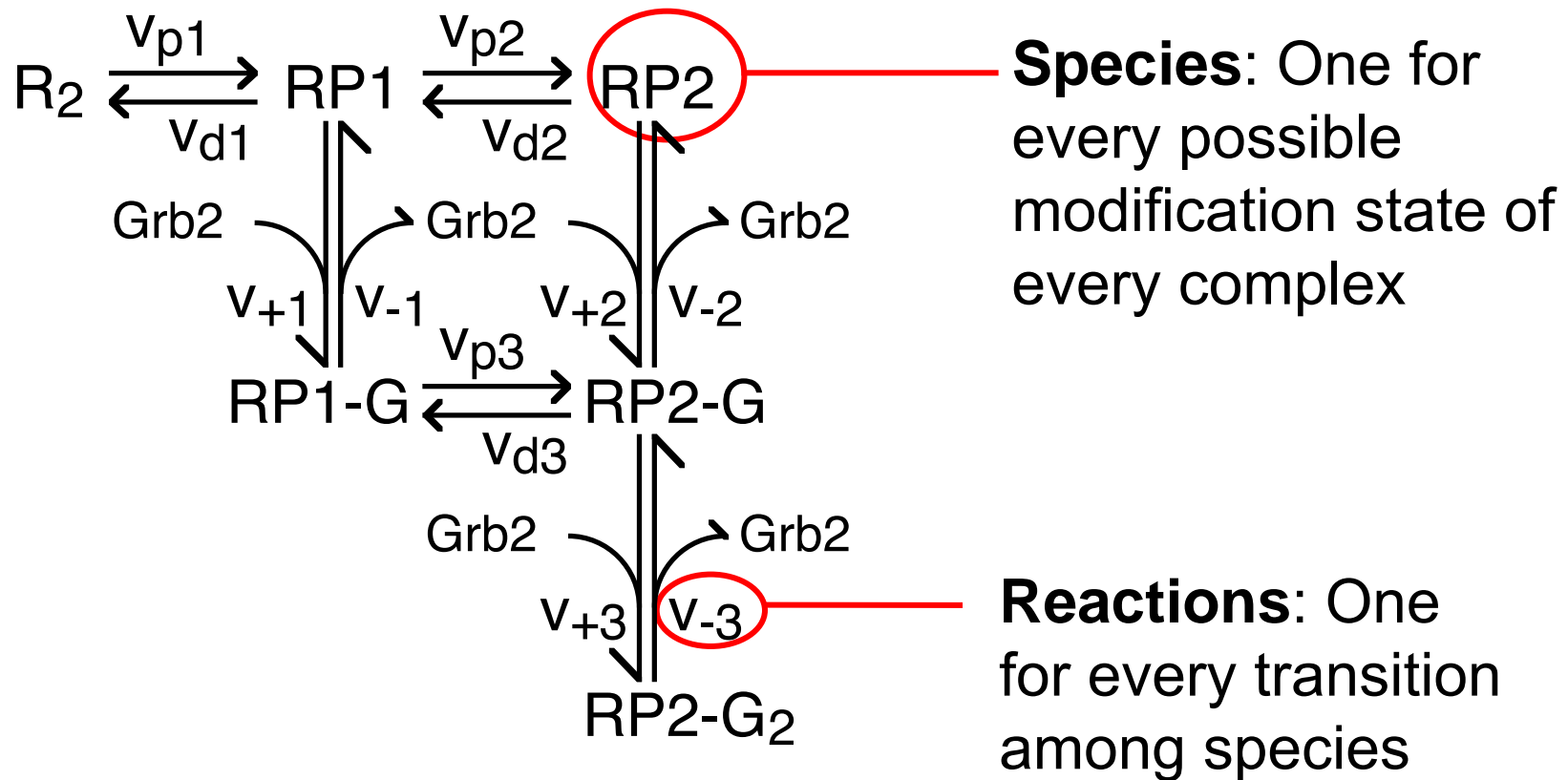
**Michael Blinov, James Schaff, Ion Moraru, Oliver Ruebenacker,
Anuradha Lakshminarayana, Fei Gao, Leslie Loew**

Center for Cell Analysis and Modeling
University of Connecticut Health Center

James Faeder¹, William Hlavacek²

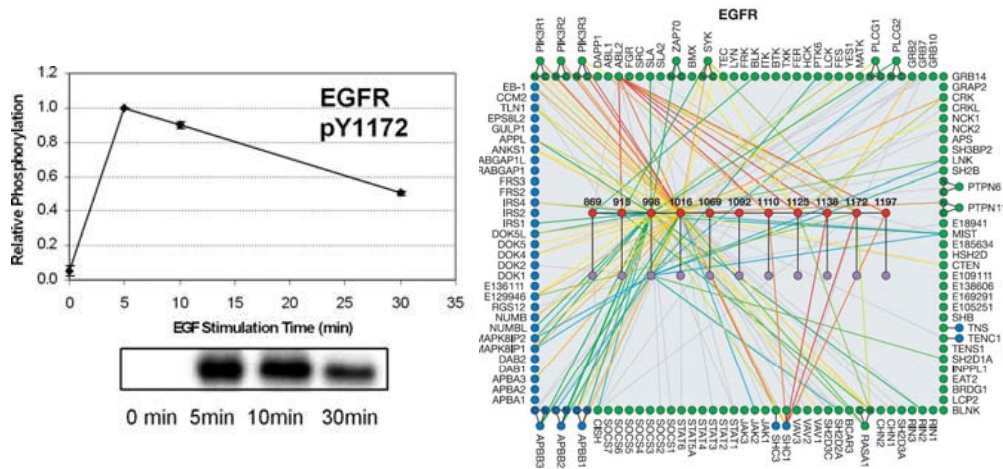
¹University of Pittsburgh, ²Los Alamos National Lab

A standard reaction scheme



Mass action kinetics gives rise to a set of ODEs, one for each species

Multiplicity of sites and binding partners gives rise to combinatorial complexity



Epidermal growth factor receptor (EGFR)

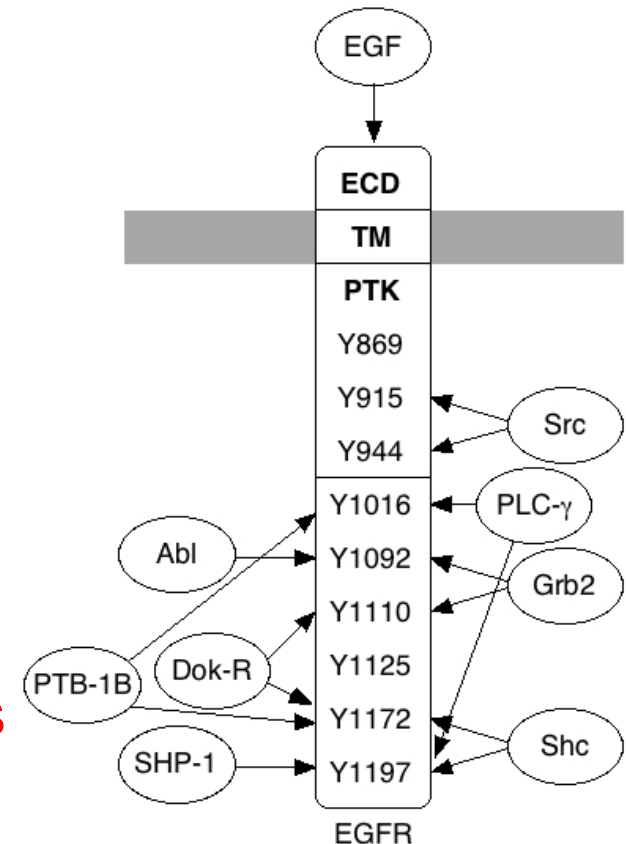
9 sites $\Rightarrow 2^9=512$ phosphorylation states

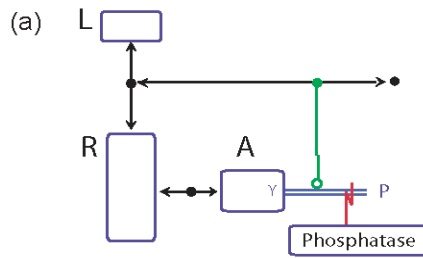
Each site has ≥ 1 binding partner

\Rightarrow more than $3^9=19,683$ total states

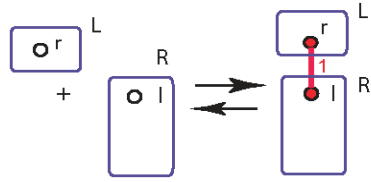
EGFR must form *dimers* to become active

\Rightarrow more than 1.9×10^8 states

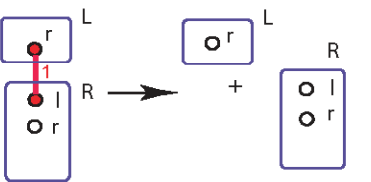
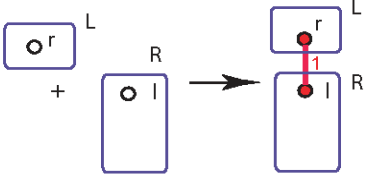




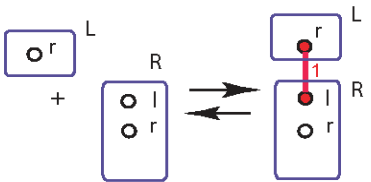
(b1) Ligand-binding independent on dimerization



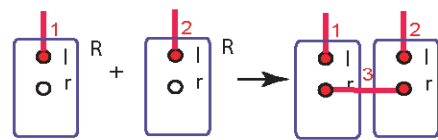
(b2) Ligand binds to any receptor, but can not dissociate in a dimer



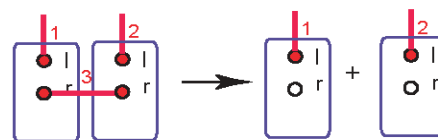
(b3) Ligand can interact with monomers only



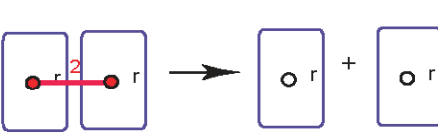
(c1) Dimer formation is ligand-induced



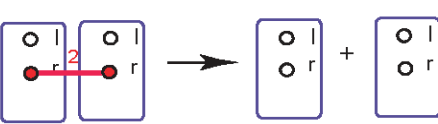
(c2) Dimer can break-up only when both ligands are present



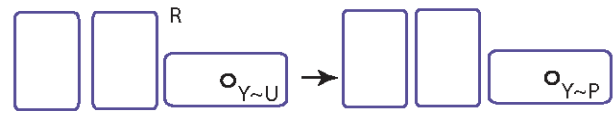
(c2) Dimer break-up is spontaneous



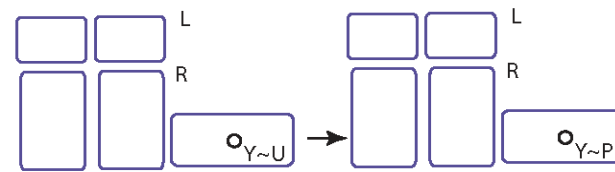
(c4) Dimer can break-up only after both ligand are gone.



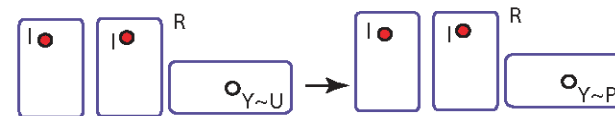
(d1) A is phosphorylated in a dimer



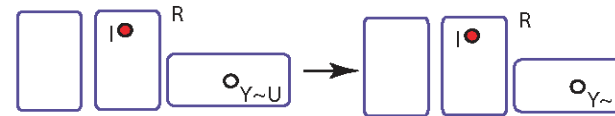
(c2) Phosphorylation requires 2 ligands L



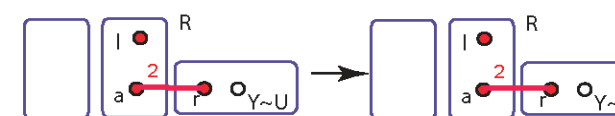
(d3) Phosphorylation requires two ligands



(d4) Phosphorylation requires at least one ligand



(d5) Explicit requirement which ligand is required

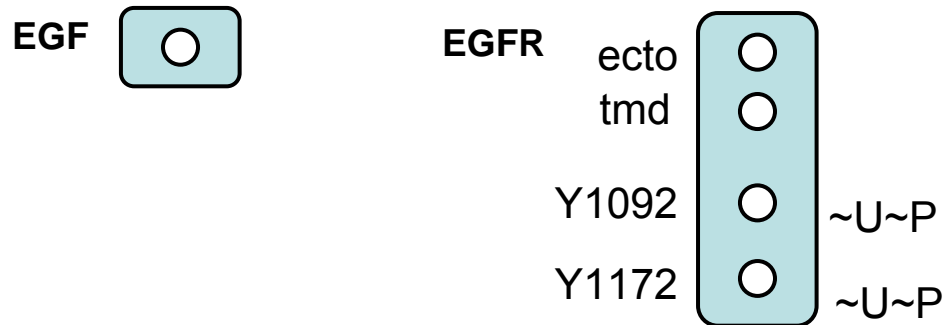


Evolution of modeling

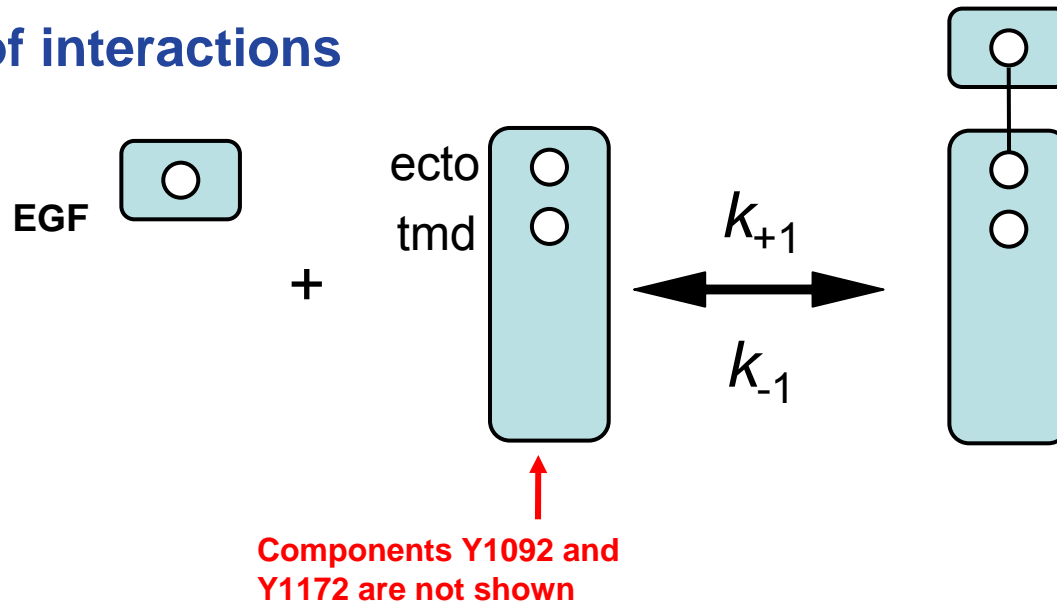
- Model variables described by mathematical equations (MathML)
- Model species and interactions described by reaction networks - can be reduced to math equations
- Model properties of the biological systems, described by rules – can be reduced to reaction networks (analogy with SBGN)

Rule-based approach

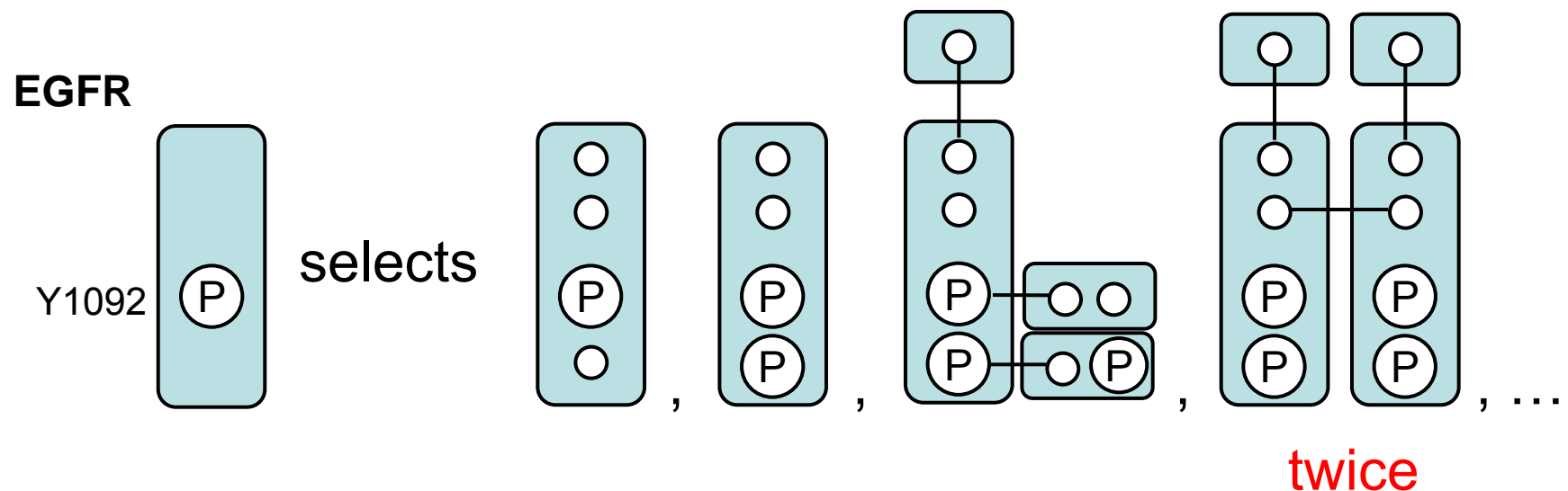
a) Biomolecules and their components



b) Rules of interactions



Property is specified via pattern



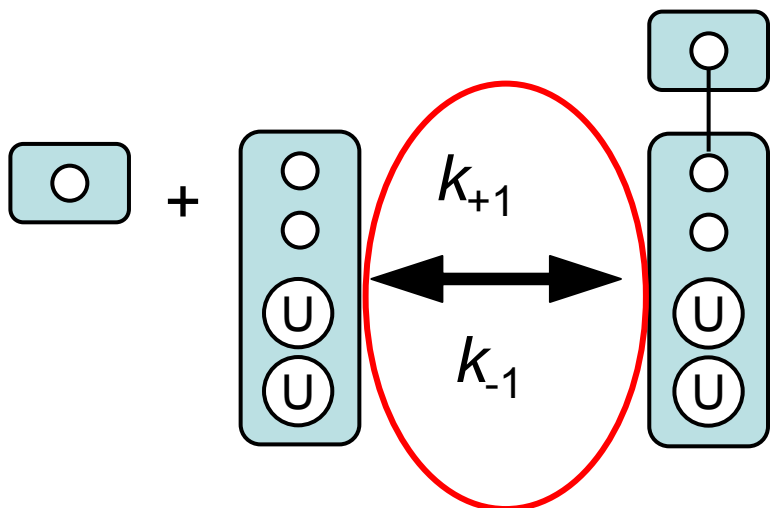
Pattern that selects EGFR phosphorylated at Y1092.

Rules generate reactions and new chemical species

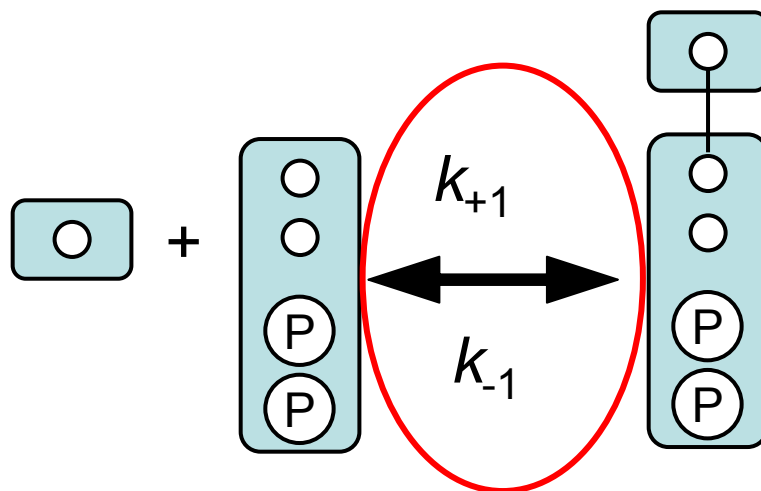
Set of species



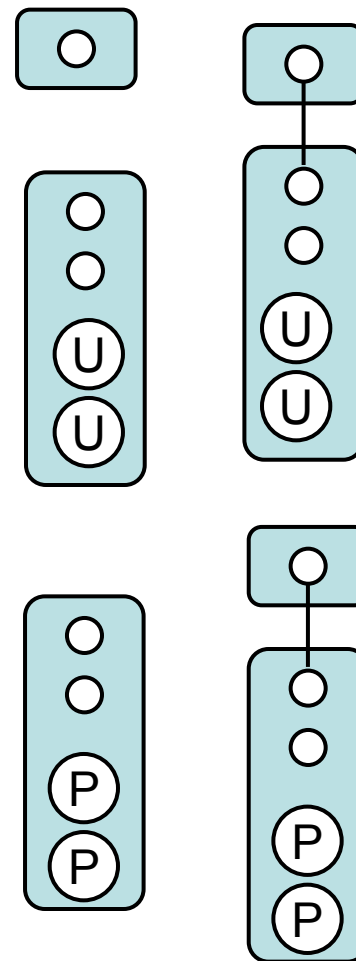
Rule application: reactions



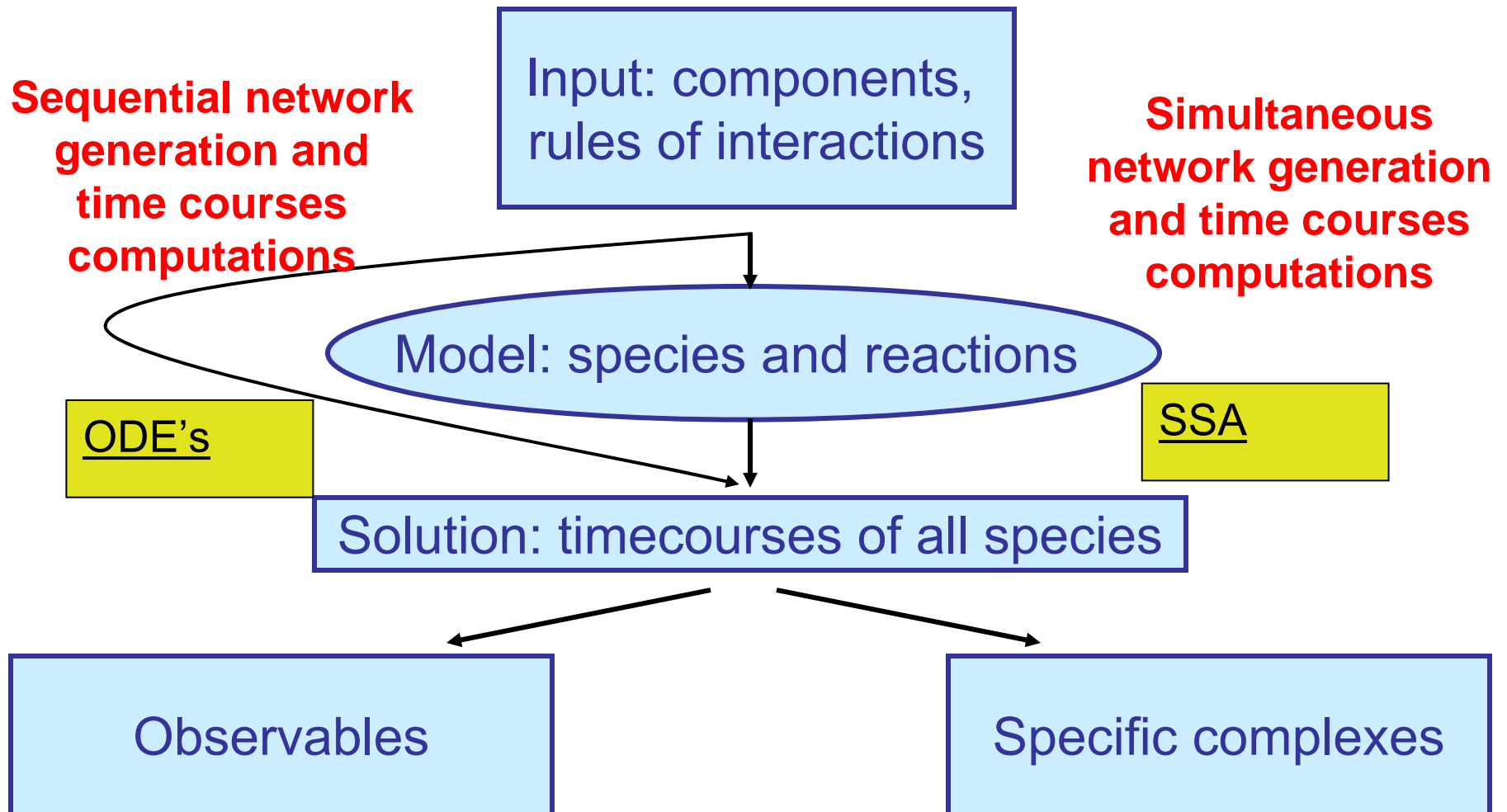
All reactions inherit the same rate law.



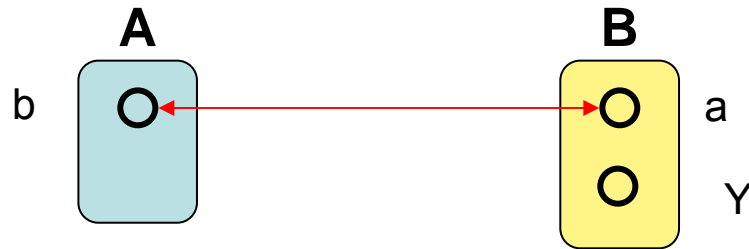
New set of species



BioNetGen modeling



BNGL



Molecules

A(b)

B(a,Y~U~P,location~Cyt~Nuc)

Patterns

B_tot

B()

B_unbound

B(a)

B_bound

B(a!+)

B_phospho_all

B(Y~P!?)

B_phospho_unbound

B(Y~P)

B_phospho_bound

B(Y~P!+)

A_B_complex

A().B()

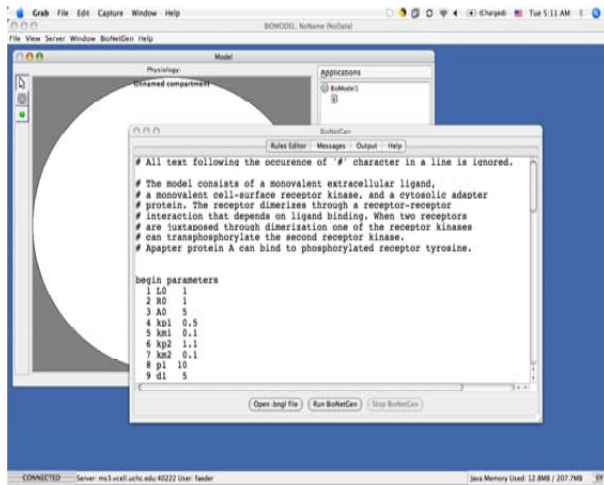
Reaction rules

A(b) + B(a) -> A(b!1).B(a!1) p

a bond between two components

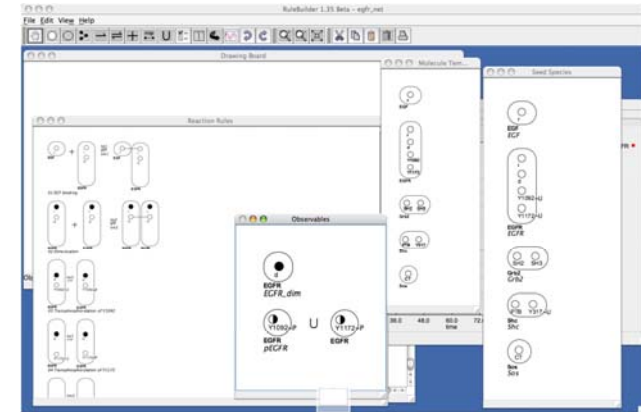
B(Y~P) -> B(Y~U) d

Three interfaces to BNG



Web interface
(text-based input

<http://vcell.org/bionetgen>)



RuleBuilder GUI

<http://bionetgen.org>

```
xterm  
ntal:~/shared/Conferences/RTX-trainingcourse2006/faeder$ BNG2 RB.bngl  
/Users/faeder/BioNetGen_2.0.40/Per12/BNG2.pl  
BioNetGen version 2.0.40  
Reading from file RB.bngl  
Read 1 parameters.  
Read 2 species.  
Read 1 reaction rule(s).  
WARNING: Removing old network file RB.net.  
Iteration 0: 2 species 0 runs 0.00e+00 CPU s  
Iteration 1: 3 species 1 runs 0.00e+00 CPU s  
Iteration 2: 3 species 1 runs 0.00e+00 CPU s  
Cumulative CPU time for each rule  
Rule 1: 1 reactions 0.00e+00 CPU s 0.00e+00 CPU s/rxn  
Total : 1 reactions 0.00e+00 CPU s 0.00e+00 CPU s/rxn  
Wrote network to RB.net.  
CPU TIME: generate_network 0.0 s.  
Network simulation using ODEs  
Running run_network on ntal.local  
full command: "/Users/faeder/BioNetGen_2.0.40/bin/run_network_sac" -o "RB" -p cvode -s 1e-08 -r 1e-08 -g "RB.net" "RB.net"  
- 0.5 2  
Read 1 parameters  
Read 3 species  
Read 1 reaction(s)  
1 reaction(s) have nonzero rate  
Read 0 prop(s) from RB.net  
Initialization took 0.00 CPU seconds  
Propagating with cvode using dense LU  
time n_steps n_deriv_calls  
0.50 308 355  
1.00 352 404  
Time course of concentrations written to file RB.cdat.  
Propagation took 0.00 CPU seconds  
Program times: 0.00 CPU s 0.00 clock s  
Updating species concentrations from RB.cdat  
CPU TIME: simulate_ode 0.0 s.  
Finished processing file RB.bngl  
CPU TIME: total 0.3 s.  
ntal:~/shared/Conferences/RTX-trainingcourse2006/faeder$
```

Terminal interface
(text-based input

<http://bionetgen.org>)

General SBML L3RBM structure

- Create a consistent and self-contained set of abstractions that have a related mathematical formalism (graph theory, rewriting logic?)
- Include support for “rule-aware” simulators and analysis tools.
- Enable “rule-unaware” tools (L2) to read sbml L3.
- Avoid changes to the SBML core described in L2V3.

+ components
+ physicalEntity
+ bonds
Species
+?? ComplexSpecies
reaction
+?? ReactionRule
+ speciesTemplate
+ ReactionTemplate
+ observable

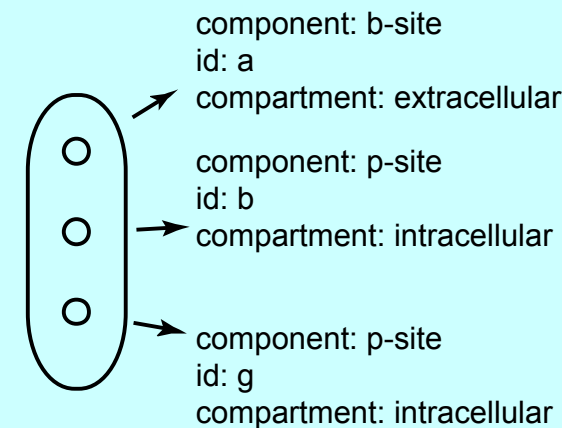
Components and physical entities

R(a,b~u~p,g~u~p)

```
<componentType id="p-site" name="phosphosite" defaultStateValue="u" >  
  <listOfComponentTypeStates>  
    <componentTypeState value="u" name="unphosphor"/>  
    <componentTypeState value="p" name="phosphor"/>  
  </listOfComponentTypeStates>  
</componentType>
```

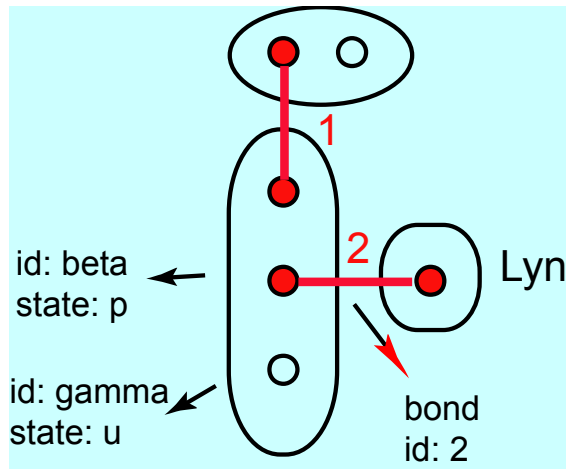
```
<physicalEntity id="R" compartment="m"/>  
  <listOfStates>  
    <state value="u" name="unfolded"/>  
    <state value="f" name="folded"/>  
  </listOfStates>  
  <listOfComponents>  
    <component id="a" componentType="b-site"/>  
    <component id="b" componentType="p-site"/>  
    <component id="g" componentType="p-site" compartment="ic"/>  
  </listOfComponents>  
</physicalEntity>
```

physicalEntity
id: R
states: u,f
compartment: membrane



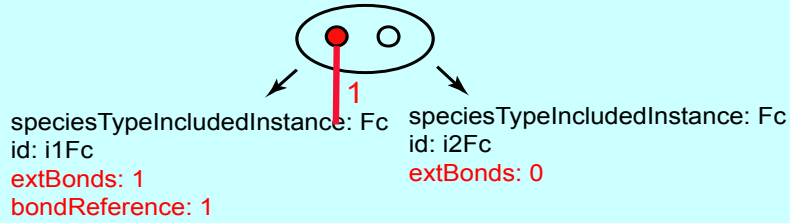
Species or ComplexSpecies

R(a!1,b~p!2,g~u).L(Fc!1,Fc).Lyn(SH2!2)

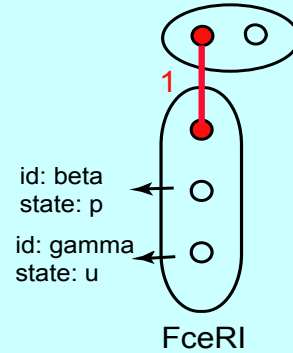


```
<species id="S_lig_monomer_Lyn ">
  <listOfPhysicalEntityInstances>
    <physicalEntityInstance physicalEntity="L" id="iL">
      <listOfComponentInstances>
        <componentInstance id="i1Fc" component="Fc"/>
          <listOfBondReferences>
            <bondReference bond="1"/>
          </listOfBondReferences>
        </componentInstance>
        <componentInstance id="i2Fc" component="Fc"/>
      </listOfComponentInstances>
    </physicalEntityInstance>
    .....
  </listOfPhysicalEntityInstances>
  <listOfBonds>
    <bond id="1" type="external"/>
    <bond id="2" type="external"/>
  </listOfBonds>
</species>
```

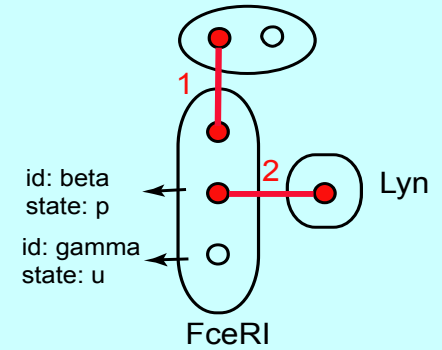
1a SpeciesPattern: P_lig_bound_once
 speciesTypeInstance: Lig
 id: iLig



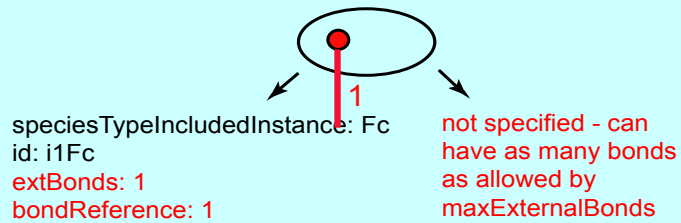
1b Species: S_lig_bound_once



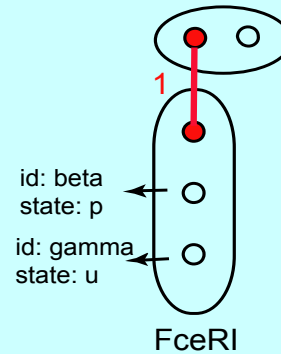
1c Species



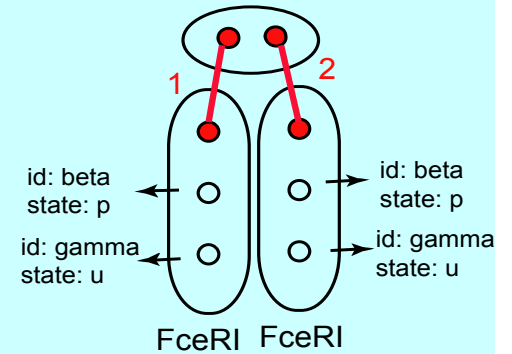
2a SpeciesPattern: P_lig_bound_one_or_more
 speciesTypeInstance: Lig
 id: iLig



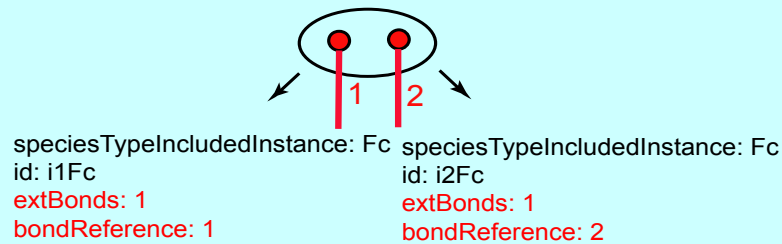
2b Species



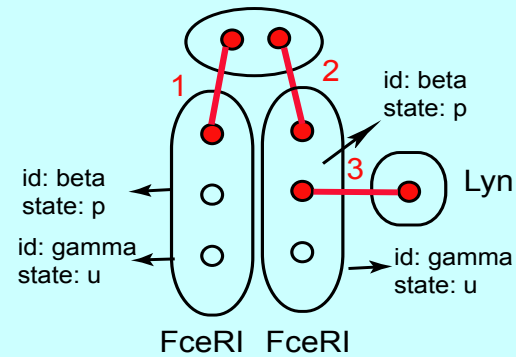
2c Species



3a SpeciesPattern: P_ligand_bound_two_times
 speciesTypeInstance: Lig
 id: iLig



3b Species



speciesTemplate

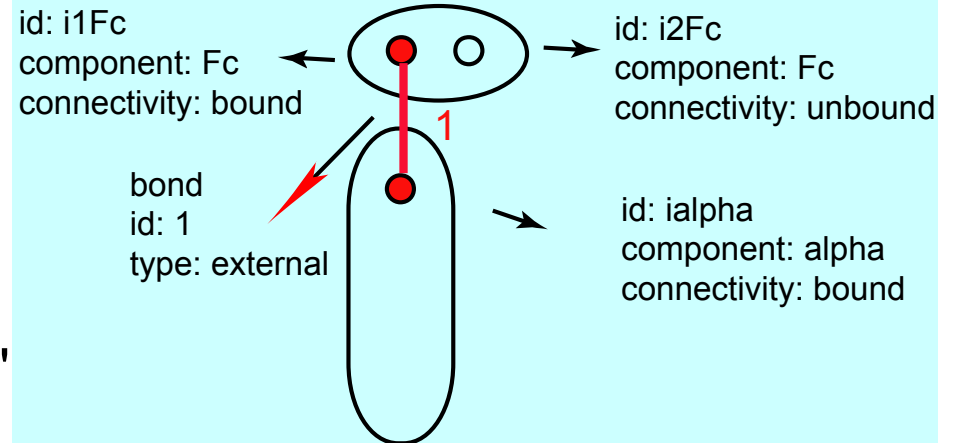
R(a!1).L(Fc!1,Fc)

R(a,g~p!?)

```
<speciesTemplate id="T_L_monomer"
  <listOfPhysicalEntitiesIncluded>
    ....
    <listOfComponentInstances>
      ....
      ...
    <listOfPhysicalEntitiesIncluded>
  <listOfBonds>
</speciesTemplate>
```

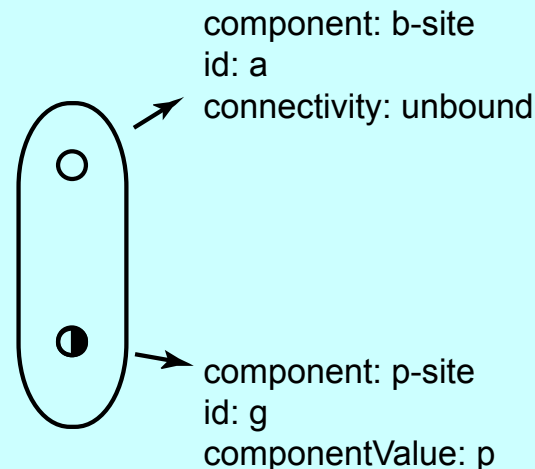
speciesType: T_lig_monomer

physicalEntityIncluded: Lig



physicalEntityIncluded: FceRI

speciesType
state: u



Reactions and reaction rules

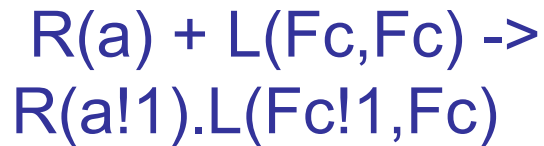
Option 1:

```
<reaction>
  <listOfReactants>
    <speciesTemplate id="R"/>
    <speciesReference species="L"/>
  </listOfReactants>
  <listOfProducts>
    <speciesTemplate id="R-L"/>
  </listOfProrducts>
</reactionRule>
```

Option 2:

```
<reactionRule>
  <listOfReactants>
    <speciesTemplate id="T_monomer"/>
    <speciesReference species="T_free_L"/>
  </listOfReactants>
  <reactionTemplate>
    <SpeciesTemplateChange>
      <bond ...>
        <kineticLaw>
          .....
        </kineticLaw>
      </SpeciesTemplateChange>
    </reactionTemplate>
  </reactionRule>
```

Reaction or ReactionRule

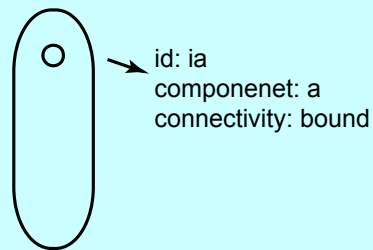


```

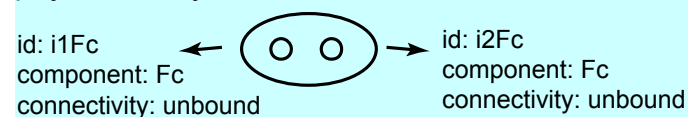
<reaction id="Ligand_bind" reversible="true">
  <listOfReactants>
    <speciesTemplate id="T_monomer"/>
    <speciesTemplate species="T_free_L"/>
  </listOfReactants>
  <reactionTemplate>
    <SpeciesTemplateChange>
      <bond ...>
        <kineticLaw>
          .....
        </kineticLaw>
      </SpeciesTemplateChange>
    </reactionTemplate>
  </reactionRule>

```

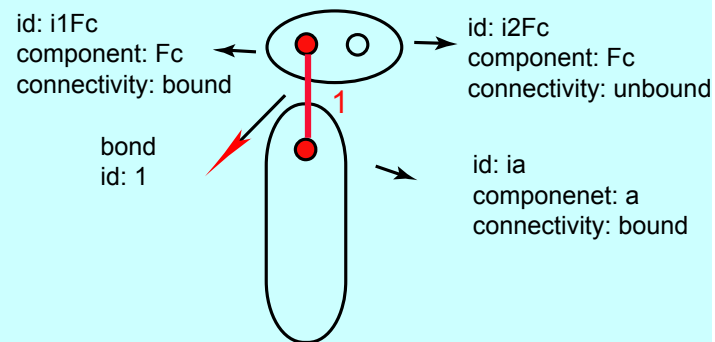
Reactant: speciesType T_monomer
physicalEntityIncluded: R



Reactant: speciesType T_free_ligand
physicalEntityIncluded: L



Product: speciesType T_L_monomer
physicalEntityIncluded: FceRI
physicalEntityIncluded: Lig



Logic and Range (BioPAX)

```
<physicalEntity id="R" >
  <listOfComponents>
    <component id="a">
      <listOfStates>
        <state id="sta1"/>
          .....
        <state id="sta6"/>
      </listOfStates>
    </component>
  </listOfComponents>
</physicalEntity>
```

```
<speciesTemplate id="T_R">
  <listOfPhysicalEntityInstances>
    <physicalEntityInstance physicalEntity="R" id="iR">
      <listOfComponentInstances>
        <componentInstance value="ia" component="a" state=" NOT sta1"/>
        <componentInstance value="ia" component="a" state="sta1 OR "sta2"/>
        <componentInstance value="ia" component="a" state="[sta3... sta5]"/>
      </listOfComponentInstances>
    </physicalEntityInstance>
  </listOfPhysicalEntityInstances>
</speciesTemplate>
```

Arbitrary level of hierarchy

- Why not specify components and physicalEntities as speciesTypes?
- Advantage: generality
- Cost: complexity of connectivity

speciesType

speciesTypeIncluded

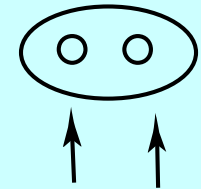
speciesType
class: "component"
id: p-site
states: u,p



speciesType
class: "component"
id: b-site



speciesType
class: physicalEntity
id: Lig



speciesTypeIncluded: b-site
id: Fc
multiplicity:2
compartment: extracellular

```
<speciesType id="p-site" defaultStateValue="u">
```

```
  <listOfSpeciesTypeStates>
```

```
    <speciesTypeState value="u" name="unphosphorylated"/>
```

```
    <speciesTypeState value="p" name="phosphorylated"/>
```

```
  </listOfSpeciesTypeStates>
```

```
</speciesType>
```

```
<speciesType id="Lig">
```

```
  <listOfSpeciesTypesIncluded>
```

```
    <speciesTypeIncluded id="Fc" speciesType="b-site" multiplicity="2">
```

```
  </listOfSpeciesTypesIncluded>
```

```
</speciesType>
```

speciesTemplate

```
id : SId {use="optional"}
name : string {use="optional"}
speciesTypeInstance: speciesType[0..*]
compartment: Sid {use="optional"}
```

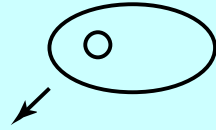
speciesTypeInstance

```
id : SId
speciesTypeValue: speciesTypeValue {use="optional"}
name : string {use="optional"}
multiplicity: int { minInclusive="0" use="optional" default="1"}
extBonds: int { minInclusive="0" maxInclusive="maxExtBonds" use="optional"}
intBonds: int { minInclusive="0" maxInclusive="maxIntBonds" use="optional"}
```

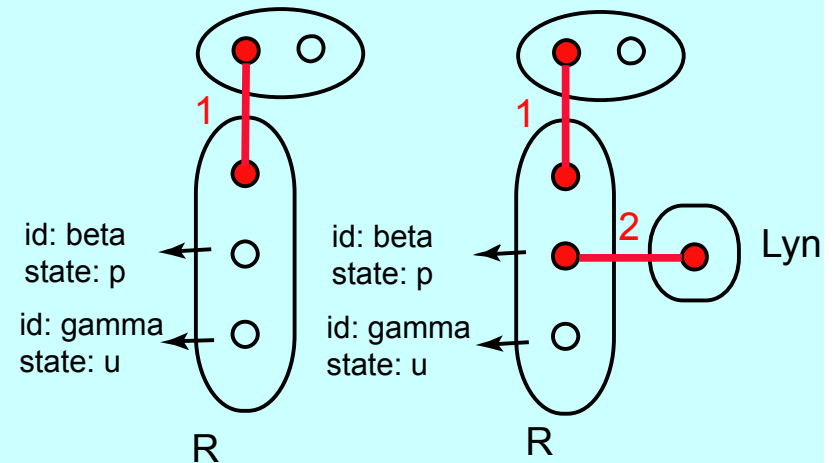
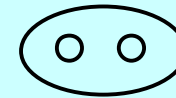
SpeciesPattern: P_L_bound_at_most_one

speciesTypeInstance: Lig
id iLig

speciesTypeIncludedInstance: Fc
id: i1Fc
extBonds: 0



species



```
<speciesTemplate id="P_ligand_free_on_one_site">  
  <listOfSpeciesTypeInstances>  
    <speciesTypeInstance speciesType="Lig" id="iLig">  
      <listOfSpeciesTypeIncludedInstances>  
        <speciesTypeIncludedInstance id="i1Fc"  
          speciesTypeIncluded="Fc" extBonds="0"/>  
      </listOfSpeciesTypeIncludedInstances>  
    </speciesTypeInstance >  
  </listOfSpeciesTypeInstances>  
</speciesTemplate>
```