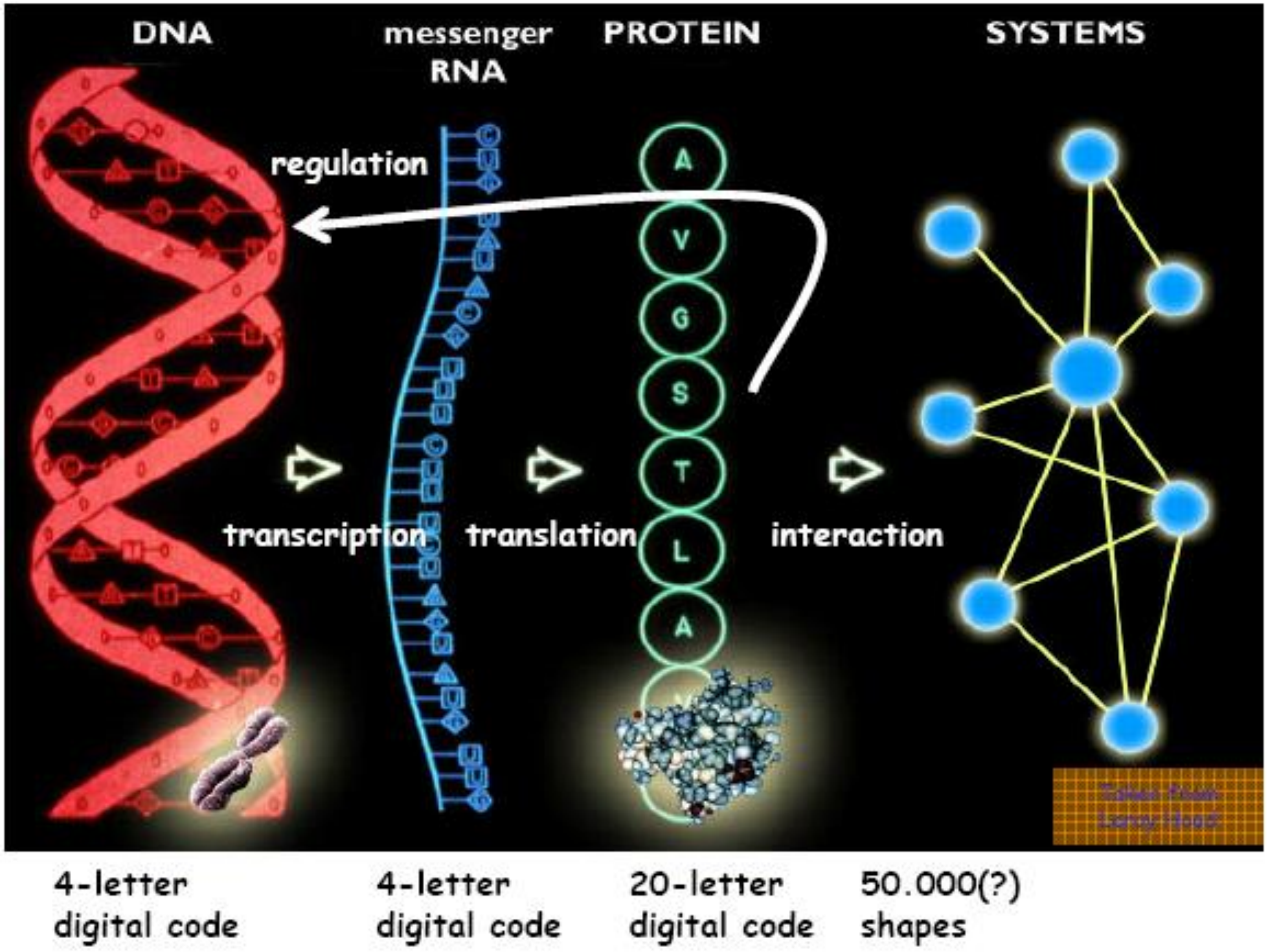
A Visual Programming Language for Biological Processes

Andrew Phillips
with
Luca Cardelli

Microsoft Research, Cambridge UK

Biological Computing

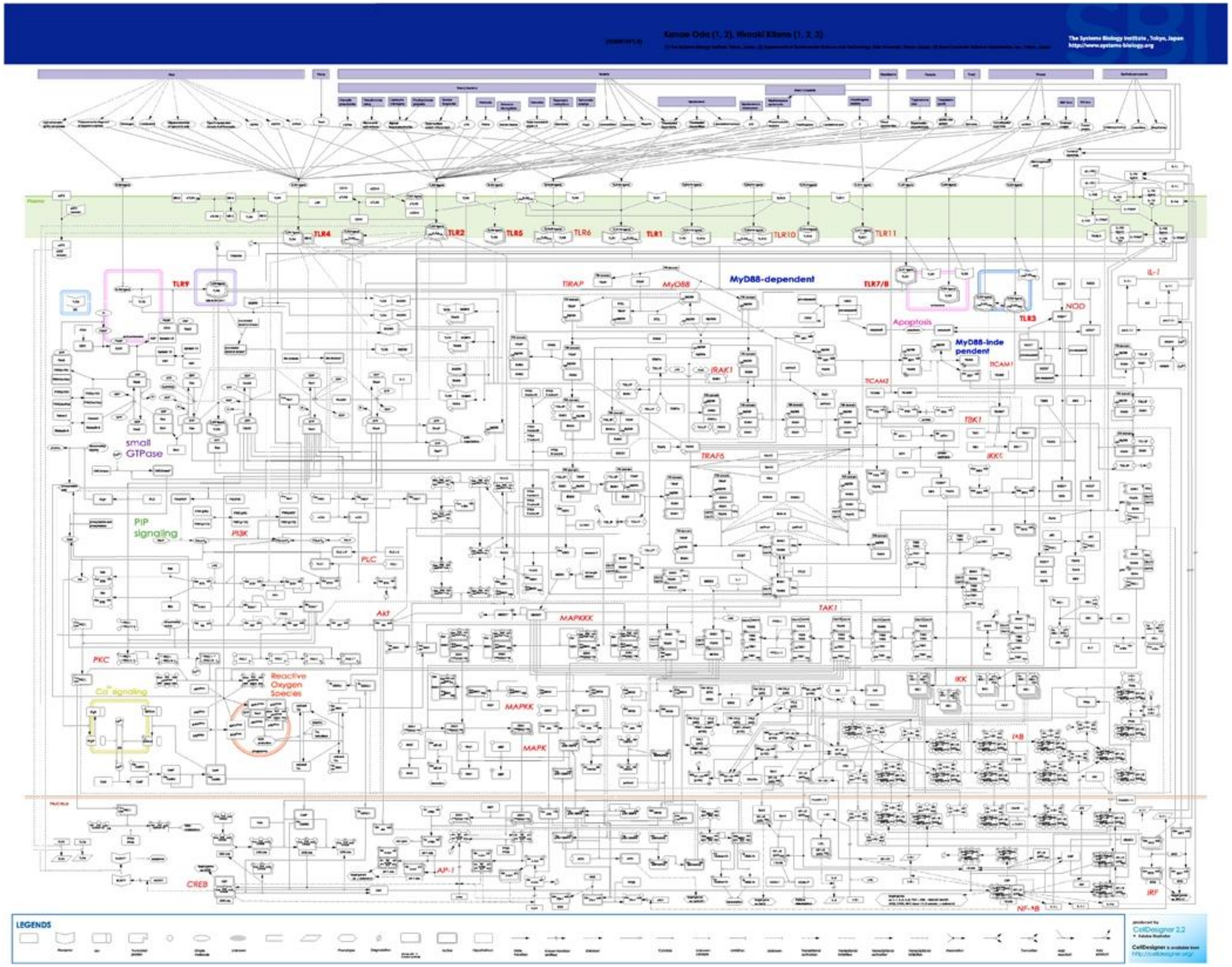


Taken From Leroy Hood

Systems Biology

- The Human Genome project:
  - Map out the complete genetic code in humans
  - To unravel the mysteries of how the human body functions
  - The code raised many more questions than answers
- Systems Biology:
  - Understand and predict the behaviour of biological systems
- Two complementary approaches:
  - Look at experimental results and infer system properties
  - Build detailed models of systems and test these in the lab
- Biological Modelling:
  - Conduct virtual experiments, saving time and resources
  - Clarify key mechanisms of how a biological system functions
  - Beginning to play a role in understanding disease

Large, Complex, Biological Models



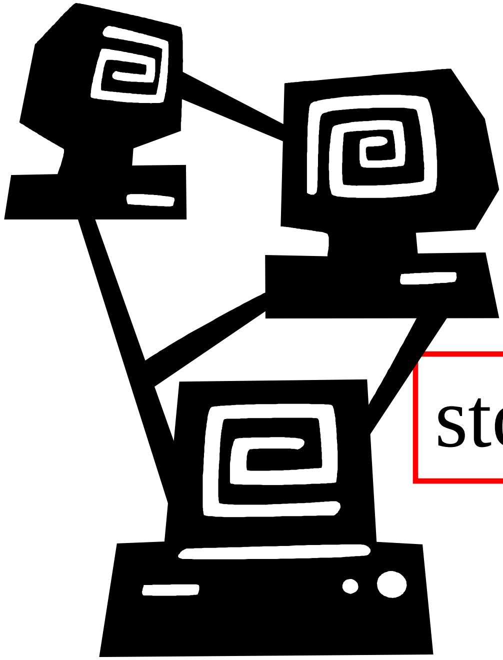
http://www.celldesigner.org

Biological Programming

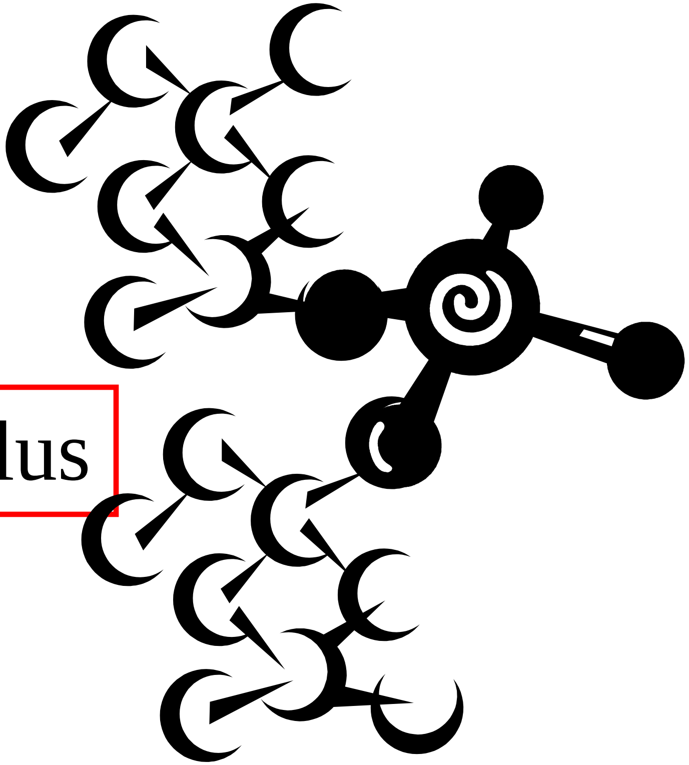
- Complex Models:
  - Difficult to understand, maintain and extend
  - Hundreds of reactions, soon to be tens of thousands
  - Would not write a program as a list of 10000 instructions
- Modularity:
  - Need a way of decomposing a model into building blocks
  - Not your average computer programs
  - Massive parallelism, each instruction has a certain probability
  - Suggests a need for a biological programming language...

Programming Languages for Biology

Languages for complex, parallel computer systems:



Languages for complex, parallel biological systems:



stochastic π-calculus

π-calculus by [Milner et al. 1989]. Stochastic version by [Priami et al. 1995]
First used in a biological context by [Regev et al. 2001]

Language Development

Exact Stochastic Simulation Algorithm

$$V,U ::= \nu x V \quad \text{Restriction} \quad (26) \qquad A,B ::= [] \quad \text{Empty} \quad (28)$$
$$\mid A \quad \text{List} \quad (27) \qquad \mid \Sigma :: A \quad \text{Summation} \quad (29)$$

Definition 7. *Syntax of* SPiM

$$n \notin fn(P) \Rightarrow P : (\nu x V) \triangleq \nu x (P : V) \quad (30)$$
$$0 : A \triangleq A \quad (31)$$
$$(P \mid Q) : A \triangleq P : Q : A \quad (32)$$
$$x \notin fn(P,A) \Rightarrow (\nu y P) : A \triangleq \nu x (P_{\{x/y\}} : A) \quad (33)$$
$$!\pi.P : A \triangleq \pi.(P \mid !\pi.P) : A \quad (34)$$
$$(\pi.P + \Sigma) : A \triangleq (\pi.P + \Sigma) :: A \quad (35)$$

Definition 8. *Construction in* SPiM

$$(P) \triangleq P : [] \quad (36)$$

Definition 9. *Encoding* SPi *to* SPiM

$$V \xrightarrow{r} V' \Rightarrow \nu x V \xrightarrow{r} \nu x V' \quad (37)$$
$$\left| \begin{array}{l} x = Next(A) \\ \wedge A \succ (x(m).P + \Sigma) :: A' \\ \wedge A' \succ (x\langle n \rangle.Q + \Sigma') :: A'' \end{array} \right. \Rightarrow A \xrightarrow{rate(x)} P_{\{n/m\}} : Q : A'' \quad (38)$$

Definition 10. *Reduction in* SPiM

$$A@\Sigma :: A' \succ \Sigma :: A@A' \quad (39)$$
$$\Sigma :: A \succ (\pi'.P' + \Sigma') :: A' \Rightarrow (\pi.P + \Sigma) :: A \succ (\pi'.P' + \pi.P + \Sigma') :: A' \quad (40)$$

Definition 11. *Selection in* SPiM

1. For all $x \in fn(A)$ calculate $a_x = \mathrm{Act}_x(A) * rate(x)$
2. Store non-zero values of a_x in a list (x_ν, a_ν), where $\nu \in 1..M$
3. Calculate $a_0 = \sum_{\nu=0}^{M} a_\nu$
4. Generate two random numbers $n_1, n_2 \in [0,1]$ and calculate τ, μ such that:

$$\tau = (1/a_0)\ln(1/n_1)$$
$$\sum_{\nu=1}^{\mu-1} a_\nu < n_2 a_0 \le \sum_{\nu=1}^{\mu} a_\nu$$

5. $Next(A) = x_\mu$ and $Delay(A) = \tau$.

Definition 12. *Calculating Next(A) and Delay(A) according to Gillespie [6].*

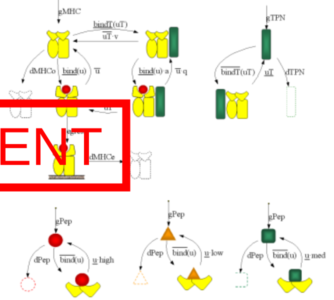
SOUND & COMPLETE

Graphical Representation and Execution Model

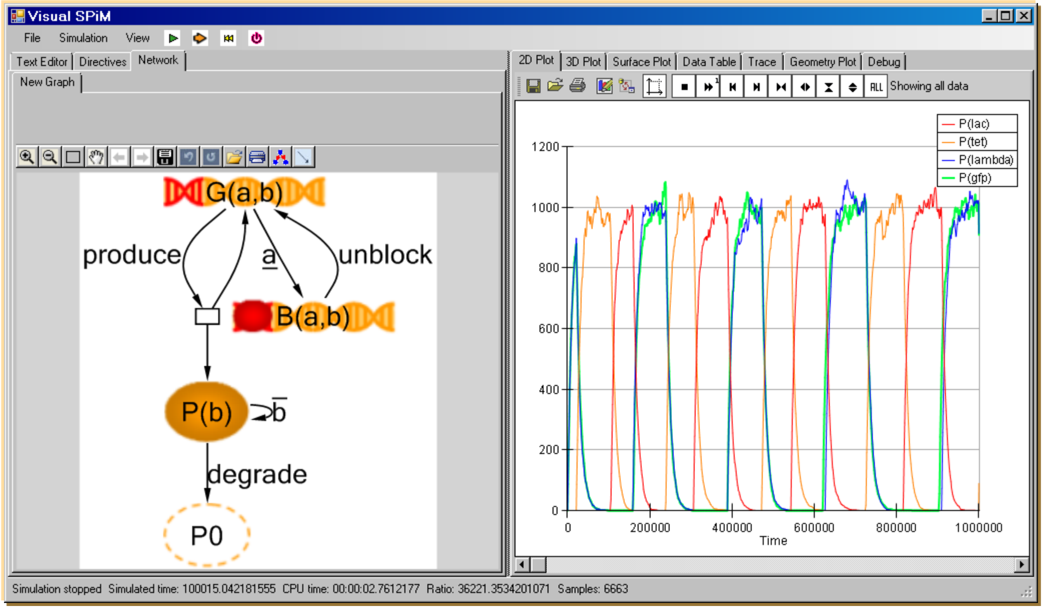
```
let gLow() = delay@g_pep; (Low() | gLow())
and Low()= (
  new low@uLow:chan
  do !pep(low); ?low; Low()
  or delay@d_pep
)
let gMed() = delay@g_pep; (Med() | gMed())
and Med()= (
  new med@uMed:chan
  do !pep(med); ?med; Med()
  or delay@d_pep
)
let gHigh() = delay@g_pep; (High() | gHigh())
and High()= (
  new high@uHigh:chan
  do !pep(high); ?high; High()
  or delay@d_pep
)
let gMHC() = delay@g_MHC;(gMHC() | MHCo() )
and gTPN() = delay@g_TPN;( gTPN() | TPN() )
```

```
and TPN()= (
  new uT@unbind_TPN:chan
  do !tpn(uT); ?uT; TPN()
  or delay@degrade_TPN
)

and MHCo()=
  do ?pep(u);MHCc_pep(u)
  or ?tpn(uT);MHCo_TPN(uT)
  or delay@degrade_MHCo
and MHCc_pep(u:chan) =
  do delay@egress; MHCe_pep(u)
  or !u; MHCo()
and MHCe_pep(u:chan) = delay@degrade_MHCe

and MHCo_TPN(uT:chan)=
  do ?pep(u)*a; MHCc_TPN_pep(u,uT)
  or !uT*v; MHCo()
and MHCc_TPN_pep(u:chan,uT:chan)=
  do !uT; MHCc_pep(u)
  or !u*q; MHCo_TPN(uT)
```

EQUIVALENT



Graphical Editor and Simulator



Simulation stopped Simulated time: 100015.042181555 CPU time: 00:00:02.7612177 Ratio: 36221.3534201071 Samples: 6663

GUI by Filippo Polo, MSR Cambridge

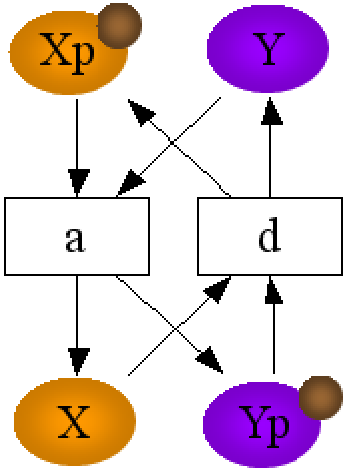
Phillips and Cardelli, 2004
Phillips, Cardelli and Castagna, 2006
Phillips and Cardelli, 2007

SPiM: Stochastic π for Biology

- A variant of stochastic π calculus
  - Supports expressive power of π
  - Graphical syntax and semantics
  - Biological constructs, e.g. complexation
  - Efficient implementation

Message-Passing Approach

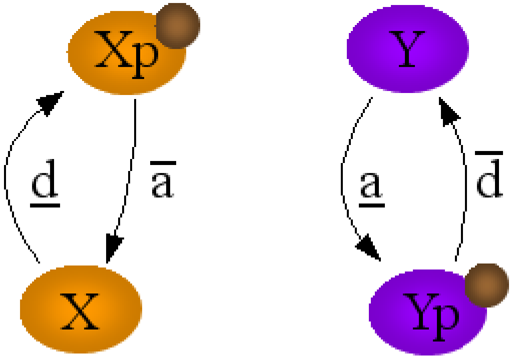
Chemical Reactions



$Xp + Y \rightarrow_{a} X + Yp$

$X + Yp \rightarrow_{d} Xp + Y$

SPiM Processes

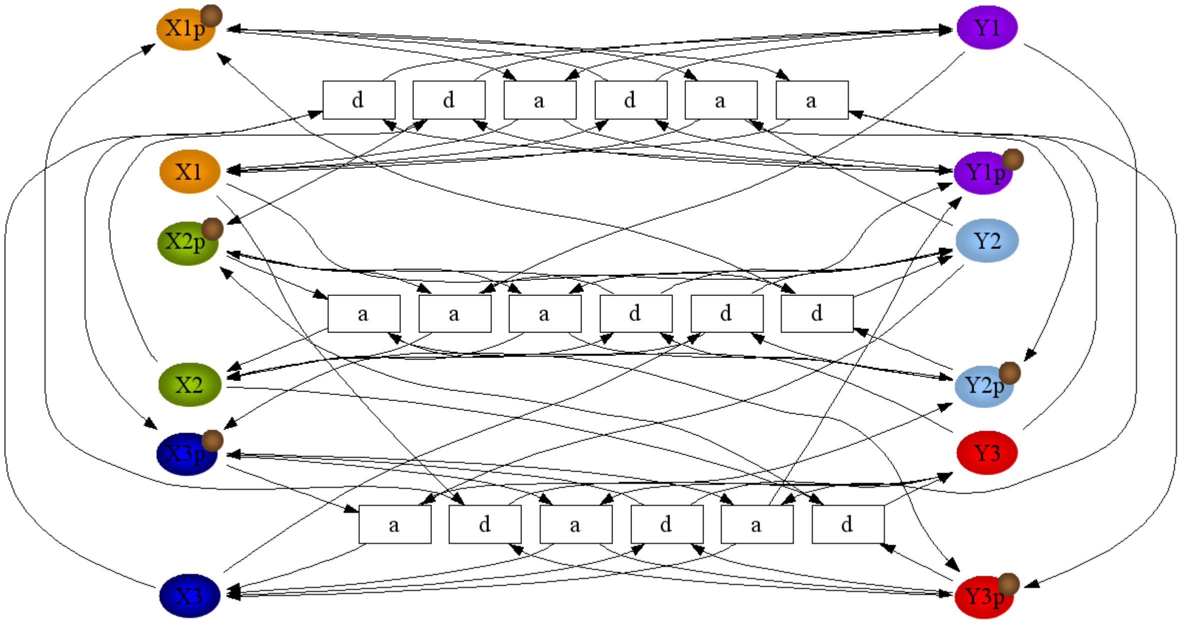


$Xp = \bar{a}.\ X$ $\qquad$ $Y = \underline{a}.\ Yp$

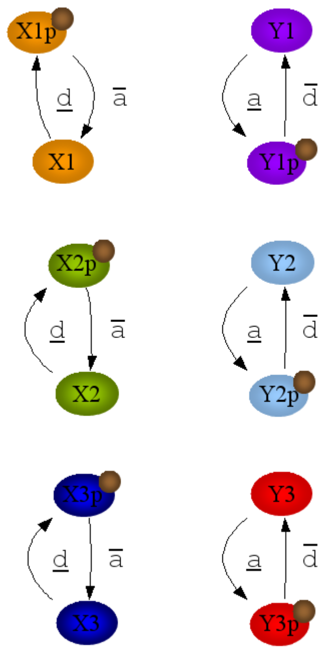
$X = \underline{d}.\ Xp$ $\qquad$ $Yp = \bar{d}.\ Y$

Compact, Modular Models

Chemical Reactions



SPiM Processes

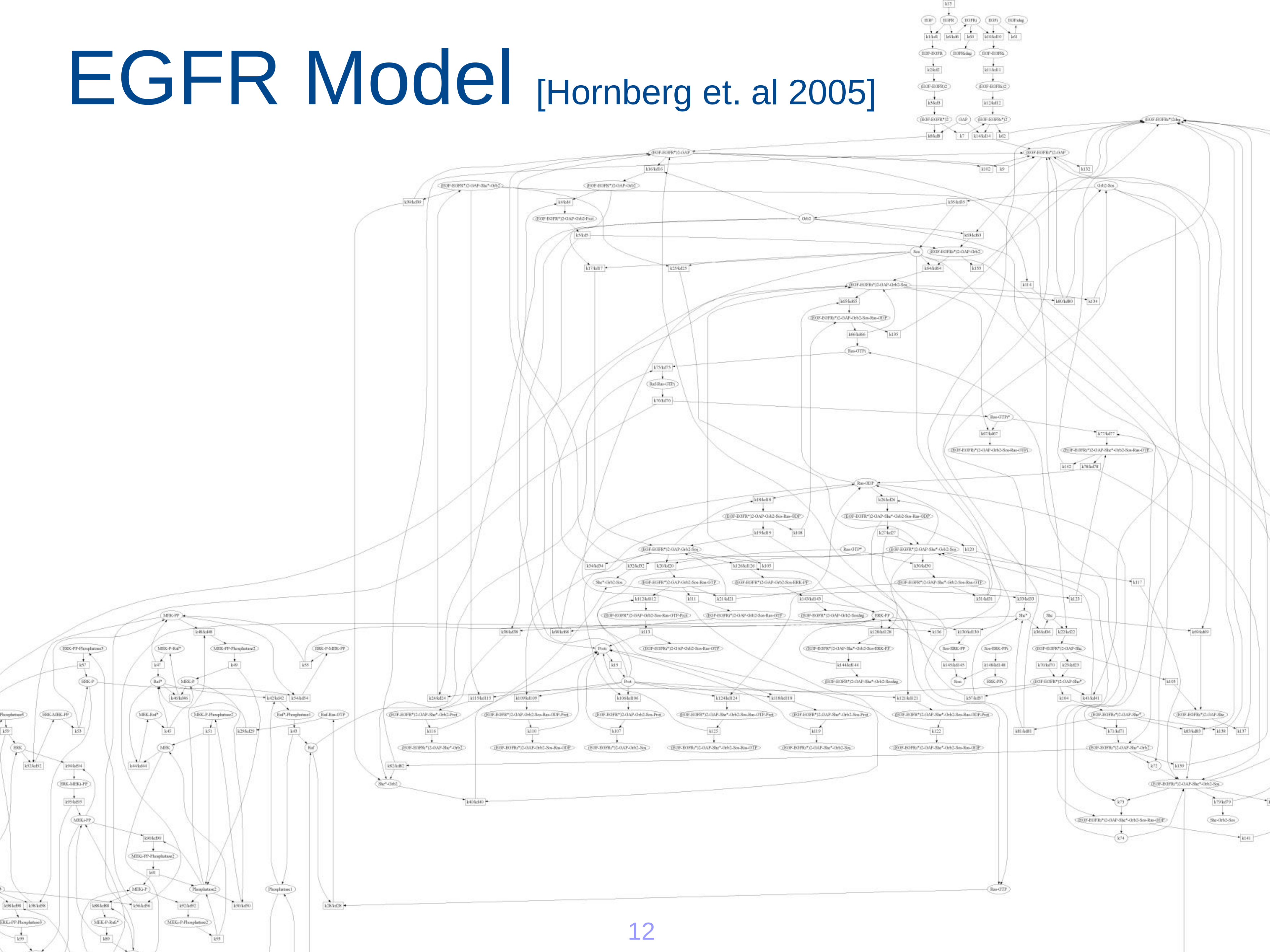


EGFR Model [Hornberg et. al 2005]

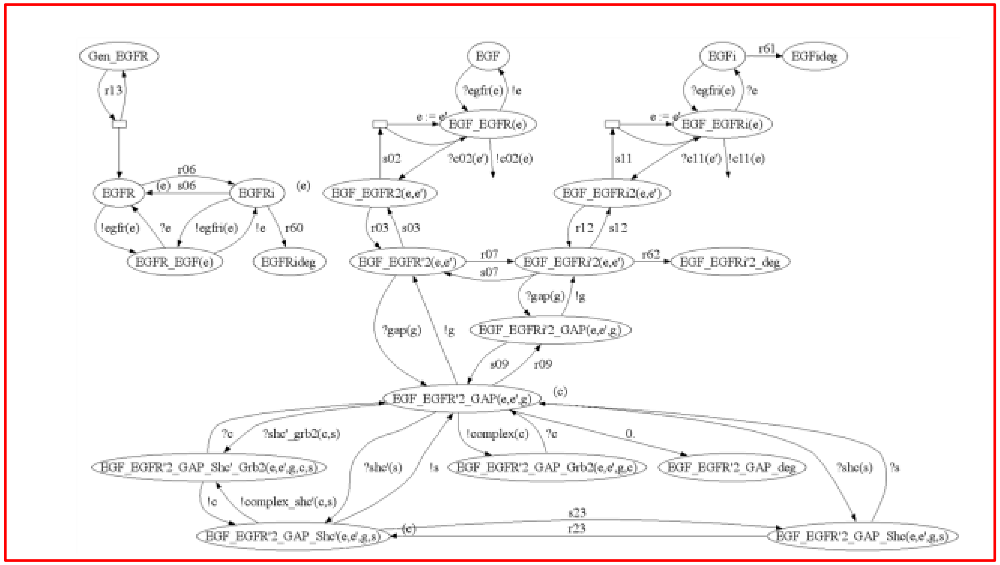
1 [EGFR]+[EGF] ↔ [EGF-EGFR]
2 [EGF-EGFR]+[EGF-EGFR] ↔ [(EGF-EGFR)2]
3 [(EGF-EGFR)2] ↔ [(EGF-EGFR*)2]
4 [(EGF-EGFR*)2-GAP-Grb2]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Grb2-Prot]
5 [(EGF-EGFR*)2-GAP-Grb2-Prot] → [(EGF-EGFRi*)2-GAP-Grb2]+[Proti]
6 [EGFR] ↔ [EGFRi]
7 [(EGF-EGFR*)2] → [(EGF-EGFRi*)2]
8 [(EGF-EGFR*)2]+[GAP] ↔ [(EGF-EGFR*)2-GAP]
9 [(EGF-EGFR*)2-GAP] → [(EGF-EGFRi*)2-GAP]
10 [EGFRi]+[EGFi] ↔ [EGF-EGFRi]
11 [EGF-EGFRi]+[EGF-EGFRi] ↔ [(EGF-EGFRi)2]
12 [(EGF-EGFRi)2] ↔ [(EGF-EGFRi*)2]
13 → [EGFR]
14 [(EGF-EGFRi*)2]+ [GAP] ↔ [(EGF-EGFRi*)2-GAP]
15 [Proti] → [Prot]
16 [(EGF-EGFR*)2-GAP]+[Grb2] ↔ [(EGF-EGFR*)2-GAP-Grb2]
17 [(EGF-EGFR*)2-GAP-Grb2]+[Sos] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]
18 [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP]
19 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GTP]
20 [Ras-GTP*]+[(EGF-EGFR*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP]
21 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GDP]
22 [(EGF-EGFR*)2-GAP]+[Shc] ↔ [(EGF-EGFR*)2-GAP-Shc]
23 [(EGF-EGFR*)2-GAP-Shc] ↔ [(EGF-EGFR*)2-GAP-Shc*]
24 [(EGF-EGFR*)2-GAP-Shc*]+[Grb2] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2]
25 [(EGF-EGFR*)2-GAP-Shc*-Grb2]+[Sos] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]
26 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]
27 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] + [Ras-GTP]
28 [Raf]+[Ras-GTP] ↔ [Raf-Ras-GTP]
29 [Raf-Ras-GTP] ↔ [Raf*]+[Ras-GTP*]
30 [Ras-GTP*]+[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]
31 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP]
32 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP]+[Shc*-Grb2-Sos]
33 [Shc*-Grb2-Sos] ↔ [Grb2-Sos]+[Shc*]
34 [(EGF-EGFR*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP]+[Grb2-Sos]
35 [Grb2-Sos] ↔ [Grb2] +[Sos]
36 [Shc*] ↔ [Shc]
37 [(EGF-EGFR*)2-GAP-Shc*] ↔ [(EGF-EGFR*)2-GAP]+[Shc*]
38 [Shc*]+[Grb2] ↔ [Shc*-Grb2]
39 [(EGF-EGFR*)2-GAP-Shc*-Grb2] ↔ [(EGF-EGFR*)2-GAP]+[Shc*-Grb2]
40 [Shc*-Grb2]+[Sos] ↔ [Shc*-Grb2-Sos]
41 [(EGF-EGFR*)2-GAP-Shc*] + [Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]
42 [Raf*]+[Phosphatase1] ↔ [Raf*-Phosphatase1]
43 [Raf*-Phosphatase1] → [Raf]+[Phosphatase1]
44 [MEK] + [Raf*] ↔ [MEK-Raf*]
45 [MEK-Raf*] → [MEK-P] +[Raf*]
46 [MEK-P]+[Raf*] ↔ [MEK-P-Raf*]
47 [MEK-P-Raf*] → [MEK-PP] + [Raf*]
48 [MEK-PP]+[Phosphatase2] ↔ [MEK-PP-Phosphatase2]
49 [MEK-PP-Phosphatase2] → [MEK-P] + [Phosphatase2]
50 [MEK-P]+[Phosphatase2] ↔ [MEK-P-Phosphatase2]
51 [MEK-P-Phosphatase2] → [MEK]+[Phosphatase2]
52 [ERK]+[MEK-PP] ↔ [ERK-MEK-PP]
53 [ERK-MEK-PP] → [ERK-P]+[MEK-PP]
54 [ERK-P]+[MEK-PP] ↔ [ERK-P-MEK-PP]
55 [ERK-P-MEK-PP] → [ERK-PP]+[MEK-PP]
56 [ERK-PP]+[Phosphatase3] ↔ [ERK-PP-Phosphatase3]
57 [ERK-PP-Phosphatase3] → [ERK-P]+[Phosphatase3]
58 [ERK-P] + [Phosphatase3] ↔ [ERK-P-Phosphatase3]
59 [ERK-P-Phosphatase3] → [ERK]+[Phosphatase3]
60 [EGFRi] → [EGFRideg]
61 [EGFi]→ [EGFideg]
62 [(EGF-EGFRi*)2] → [(EGF-EGFRi*)2deg]
63 [(EGF-EGFRi*)2-GAP]+[Grb2] ↔ [(EGF-EGFRi*)2-GAP-Grb2]
64 [(EGF-EGFRi*)2-GAP-Grb2]+[Sos] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos]
65 [(EGF-EGFRi*)2-GAP-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP]
66 [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos]+[Ras-GTPi]
67 [Ras-GTPi*]+[(EGF-EGFRi*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTPi]
68 [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos]+[Ras-GDP]
69 [(EGF-EGFRi*)2-GAP]+[Shc] ↔ [(EGF-EGFRi*)2-GAP-Shc]
70 [(EGF-EGFR*)2-GAP-Shc] ↔ [(EGF-EGFR*)2-GAP-Shc*]
71 [(EGF-EGFRi*)2-GAP-Shc*]+[Grb2] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2]
146 [(EGF-EGFRi*)2-GAP-Grb2-Sos-ERKi-PP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos]deg+[ERKi-PP]
147 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-ERKi-PP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]deg+[ERKi-PP]
148 [Sos-ERK-PPi] ↔ [Sosi]+[ERK-PPi]

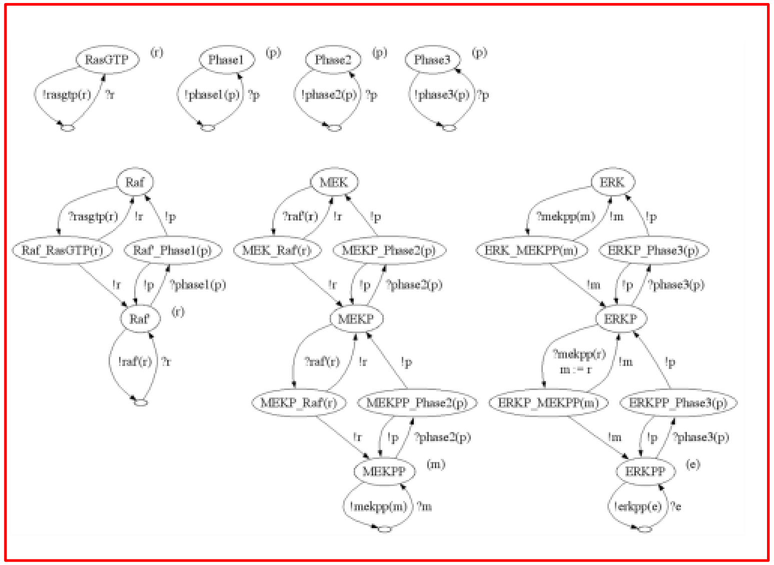
72 [(EGF-EGFRi*)2-GAP-Shc*-Grb2]+[Sos] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]
73 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]
74 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos] + [Ras-GTPi]
75 [Raf]+[Ras-GTPi] ↔ [Raf-Ras-GTPi]
76 [Raf-Ras-GTPi] ↔ [Rafi*]+[Ras-GTPi*]
77 [Ras-GTPi*]+[(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]
78 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP]
79 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP]+[Shc-Grb2-Sos]
80 [(EGF-EGFRi*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP]+[Grb2-Sos]
81 [(EGF-EGFRi*)2-GAP-Shc*] ↔ [(EGF-EGFRi*)2-GAP]+[Shc*]
82 [(EGF-EGFRi*)2-GAP-Shc*-Grb2] ↔ [(EGF-EGFRi*)2-GAP]+[Shc*-Grb2]
83 [(EGF-EGFRi*)2-GAP-Shc*] + [Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]
84 [Rafi*]+[Phosphatase1] ↔ [Rafi*-Phosphatase1]
85 [Rafi*-Phosphatase1] → [Raf]+[Phosphatase1]
86 [MEK] + [Rafi*] ↔ [MEK-Rafi*]
87 [MEK-Rafi*] → [MEKi-P] +[Rafi*]
88 [MEKi-P]+[Rafi*] ↔ [MEK-P-Rafi*]
89 [MEK-P-Rafi*] → [MEKi-PP] + [Rafi*]
90 [MEKi-PP]+[Phosphatase2] ↔ [MEKi-PP-Phosphatase2]
91 [MEKi-PP-Phosphatase2] → [MEKi-P] + [Phosphatase2]
92 [MEKi-P]+[Phosphatase2] ↔ [MEKi-P-Phosphatase2]
93 [MEKi-P-Phosphatase2] → [MEK]+[Phosphatase2]
94 [ERK]+[MEKi-PP] ↔ [ERK-MEKi-PP]
95 [ERK-MEKi-PP] → [ERKi-P]+[MEKi-PP]
96 [ERKi-P]+[MEKi-PP] ↔ [ERKi-P-MEKi-PP]
97 [ERKi-P-MEKi-PP] → [ERKi-PP]+[MEKi-PP]
98 [ERKi-PP]+[Phosphatase3] ↔ [ERKi-PP-Phosphatase3]
99 [ERKi-PP-Phosphatase3] → [ERKi-P]+[Phosphatase3]
100 [ERKi-P] + [Phosphatase3] ↔ [ERKi-P-Phosphatase3]
101 [ERKi-P-Phosphatase3] → [ERK]+[Phosphatase3]
102 [(EGF-EGFR*)2-GAP] → [(EGF-EGFRi*)2-GAP]
103 [(EGF-EGFR*)2-GAP-Shc] → [(EGF-EGFRi*)2-GAP-Shc]
104 [(EGF-EGFR*)2-GAP-Shc*] → [(EGF-EGFRi*)2-GAP-Shc*]
105 [(EGF-EGFR*)2-GAP-Grb2-Sos] → [(EGF-EGFRi*)2-GAP-Grb2-Sos]
106 [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Prot]
107 [(EGF-EGFR*)2-GAP-Grb2-Sos-Prot] → [(EGF-EGFRi*)2-GAP-Grb2-Sos]+[Proti]
108 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP] → [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP]
109 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP-Prot]
110 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP-Prot] → [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP]+[Proti]
111 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP] → [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP]
112 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP-Prot]
113 [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP-Prot] → [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP]+[Proti]
114 [(EGF-EGFR*)2-GAP-Shc*-Grb2] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2]
115 [(EGF-EGFR*)2-GAP-Shc*-Grb2]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Prot]
116 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Prot] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2]+[Proti]
117 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]
118 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Prot]
119 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Prot] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]+[Proti]
120 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]
121 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP-Prot]
122 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP-Prot] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]+[Proti]
123 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]
124 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP-Prot]
125 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP-Prot] → [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]+[Proti]
126 [(EGF-EGFR*)2-GAP-Grb2-Sos]+[ERK-PP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-ERK-PP]
127 [(EGF-EGFRi*)2-GAP-Grb2-Sos]+[ERKi-PP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos-ERKi-PP]
128 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[ERK-PP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-ERK-PP]
129 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]+[ERKi-PP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-ERKi-PP]
130 [Sos]+[ERK-PP] ↔ [Sos-ERK-PP]
131 [Sos]+[ERKi-PP] ↔ [Sos-ERKi-PP]
132 [(EGF-EGFRi*)2-GAP] → [(EGF-EGFRi*)2deg]
133 [(EGF-EGFRi*)2-GAP-Grb2] → [(EGF-EGFRi*)2deg]
134 [(EGF-EGFRi*)2-GAP-Grb2-Sos] → [(EGF-EGFRi*)2deg]
135 [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP] → [(EGF-EGFRi*)2deg]
136 [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP] → [(EGF-EGFRi*)2deg]
137 [(EGF-EGFRi*)2-GAP-Shc] → [(EGF-EGFRi*)2deg]
138 [(EGF-EGFRi*)2-GAP-Shc*] → [(EGF-EGFRi*)2deg]
139 [(EGF-EGFRi*)2-GAP-Shc*-Grb2] → [(EGF-EGFRi*)2deg]
140 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos] → [(EGF-EGFRi*)2deg]
141 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] → [(EGF-EGFRi*)2deg]
142 [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] → [(EGF-EGFRi*)2deg]
143 [(EGF-EGFR*)2-GAP-Grb2-Sos-ERK-PP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]deg+[ERK-PP]
144 [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-ERK-PP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]deg+[ERK-PP]
145 [Sos-ERK-PP] ↔ [Sosi]+[ERK-PP]

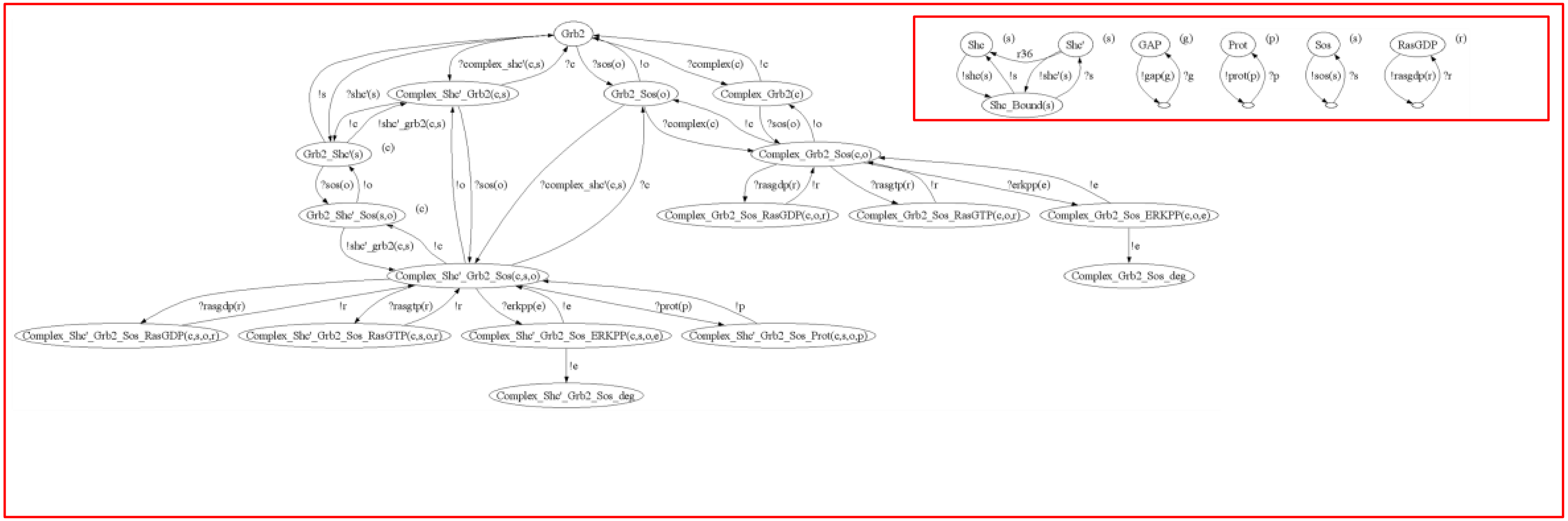
EGFR Model [Hornberg et. al 2005]



Modular EGFR







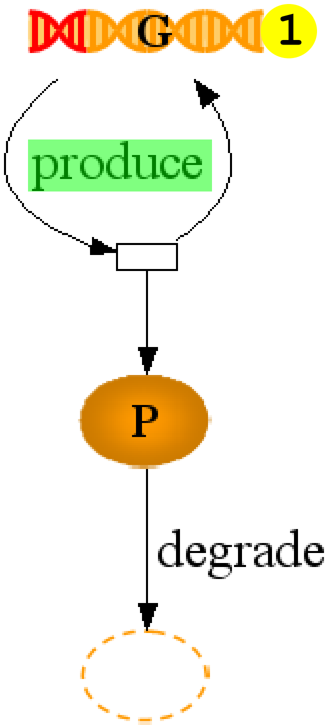
Outline

- Basic Examples
- Gene Networks
- *C. elegans* Development
- Immune System Modelling

Basic SPiM Examples

Protein Production
Protein Interaction
Protein Binding

Production: $G \xrightarrow{\text{produce}} G + P \qquad P \xrightarrow{\text{degrade}} \varnothing$

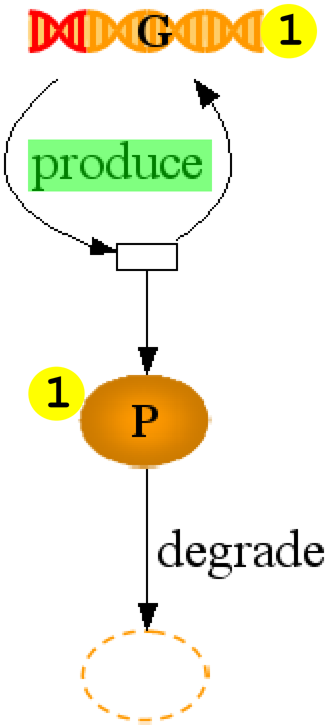


reaction	rate	propensity (s^{-1})
produce	0.1	0.1·1
degrade	0.001	0.001·0

G = produce.(P | G)
P = degrade.0

- A protein *P* can be produced with propensity 0.1
- Probability of a reaction depends on propensity
- Exact simulation: what happens next?

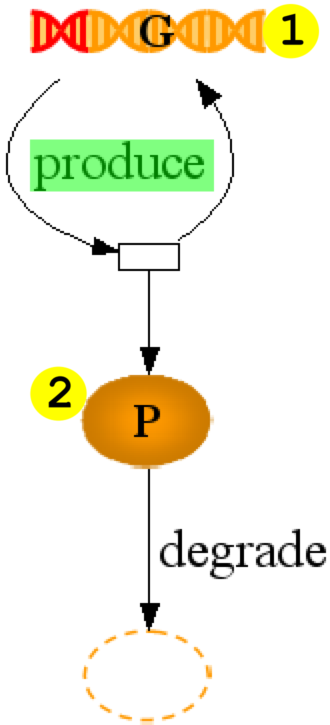
Production: $G \xrightarrow{\text{produce}} G + P \quad P \xrightarrow{\text{degrade}} \varnothing$



reaction	propensity (s^{-1})
produce	0.1
degrade	0.001

- Another protein *P* can be produced
- 100 times more likely to produce than degrade

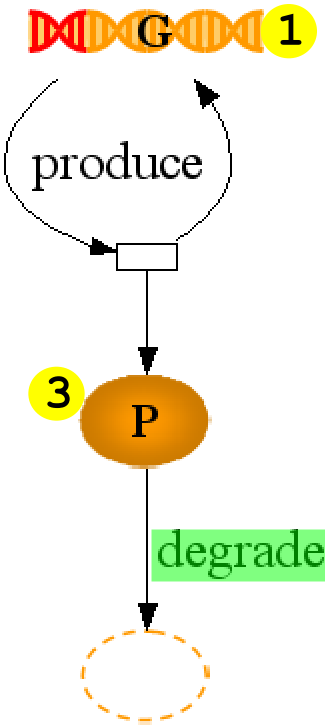
Production: $G \xrightarrow{\text{produce}} G + P \quad P \xrightarrow{\text{degrade}} \varnothing$



reaction	propensity (s^{-1})
produce	0.1
degrade	0.001·2

- And another...

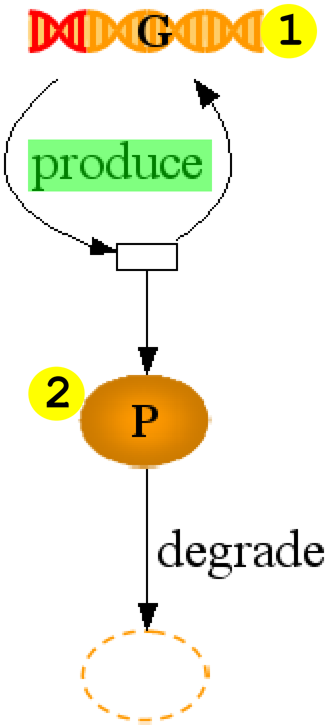
Production: $G \xrightarrow{\text{produce}} G + P \quad P \xrightarrow{\text{degrade}} \varnothing$



reaction	**propensity (s^{-1})**
produce	0.1
degrade	0.001·3

- A protein *b* can be degraded at rate 0.001
- Low probability, but still possible

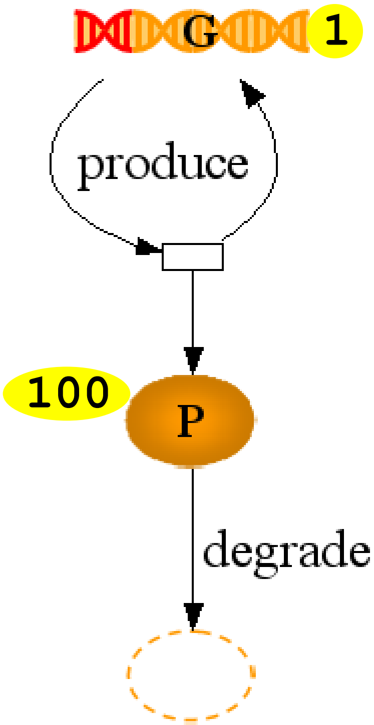
Production: $G \xrightarrow{\text{produce}} G + P \quad P \xrightarrow{\text{degrade}} \varnothing$



reaction	**propensity (s^{-1})**
produce	0.1
degrade	0.001·2

- Eventually...

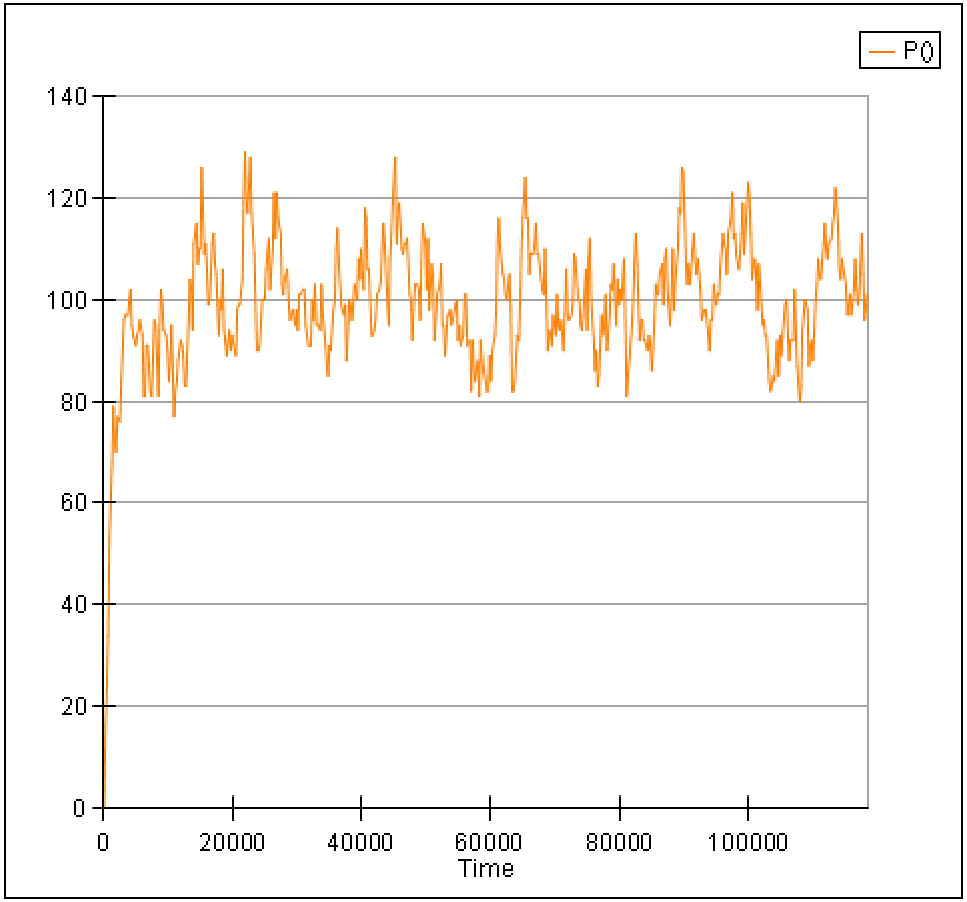
Production: $G \xrightarrow{\text{produce}} G + P \quad P \xrightarrow{\text{degrade}} \varnothing$



reaction	**propensity (s^{-1})**
produce	0.1
degrade	0.001·100

- Equilibrium at about 100 proteins.
- Propensities of both reactions are equal.

Gene Simulation

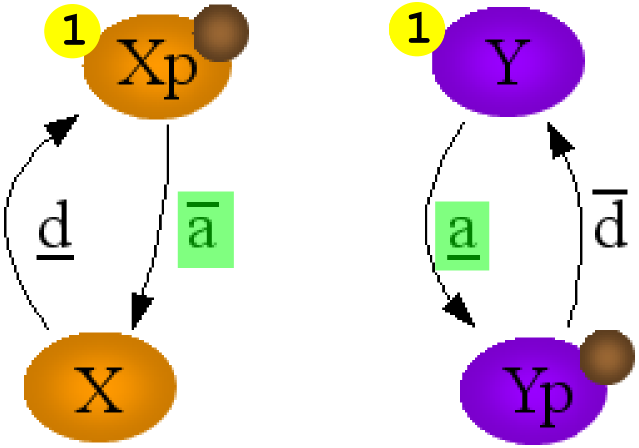


- Simulation results show evolution over time
- Level of protein *P* fluctuates around 100

Interaction: $Xp + Y \; {}_{d}\!\leftrightarrow^{a} \; X + Yp$

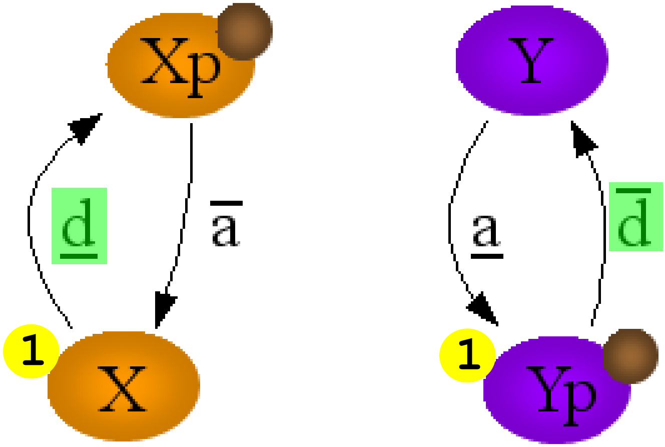
$Xp = \bar{a}.\ X$ $\qquad$ $Y = \underline{a}.\ Yp$

$X = \underline{d}.\ Xp$ $\qquad$ $Yp = \bar{d}.\ Y$



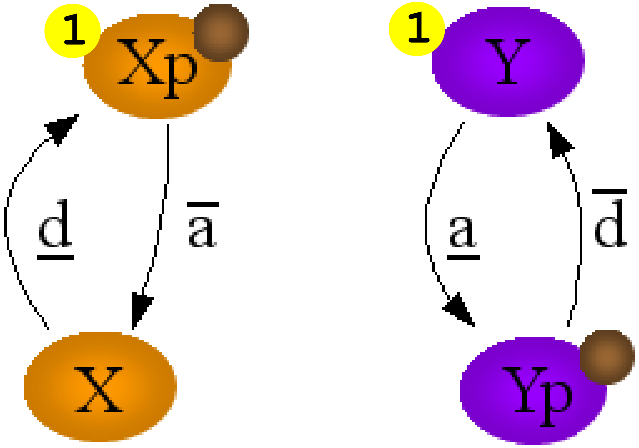
- *Xp* and *Y* can interact on channel *a*
- *Xp* activates *Y* by sending its phosphate group

Interaction: $Xp + Y \; {}_{d}{\leftrightarrow}^{a} \; X + Yp$



- *X* and *Yp* can interact on channel *d*

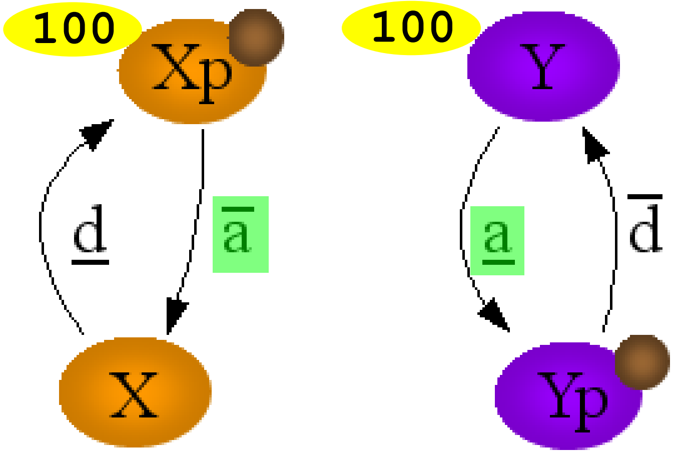
Interaction: $Xp + Y \;{}_{d}\!\leftrightarrow^{a}\; X + Yp$



- Interactions can continue indefinitely...

Interaction: $Xp + Y \;{}_{d}\!\leftrightarrow^{a}\; X + Yp$

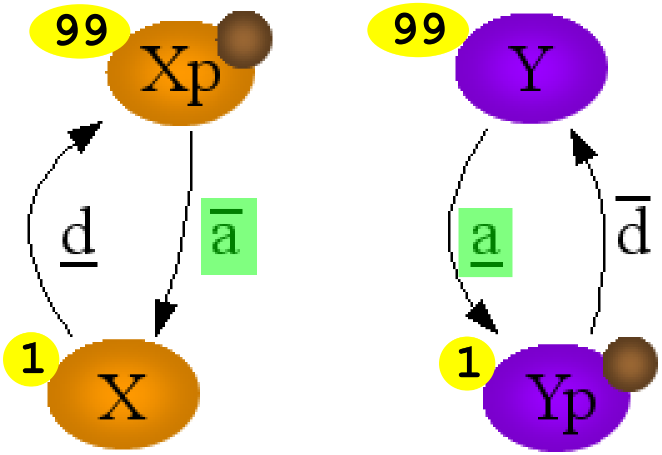
reaction	propensity (s^{-1})
a	100·100·100
d	0



- What happens if we mix 100·*Xp* and 100·*Y* ?
- Assume *rate*(*a*) = 100s^{-1} and *rate*(*d*) = 10s^{-1}
- An *Xp* and *Y* protein can interact on channel *a*.

Interaction: $Xp + Y \; {}_{d}\!\leftrightarrow^{a} \; X + Yp$

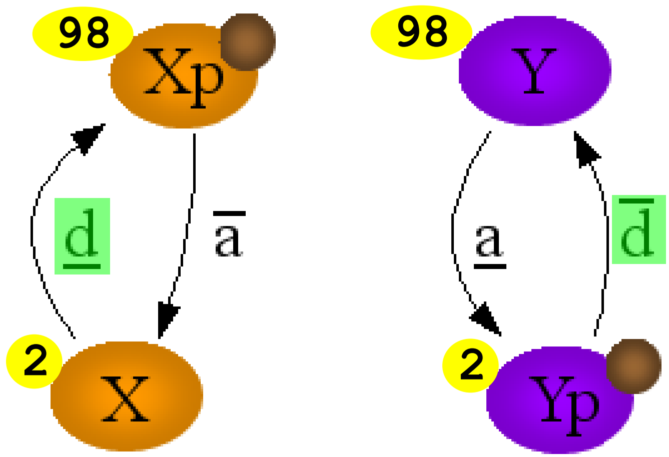
reaction	propensity (s^{-1})
a	100·99·99
d	10·1·1



- An additional *Xp* and *Y* protein can interact.

Interaction: $Xp + Y \ {}_{d}\!\leftrightarrow^{a}\ X + Yp$

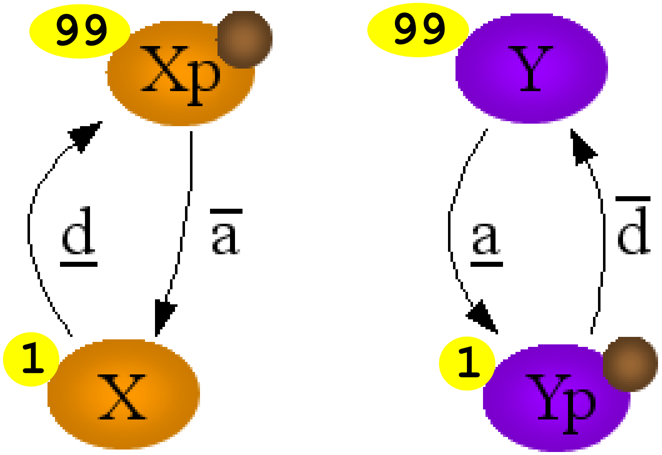
reaction	propensity (s^{-1})
a	100·98·98
d	10·2·2



- An *X* and *Yp* protein can interact

Interaction: $Xp + Y \;{}_{d}\!\leftrightarrow^{a}\; X + Yp$

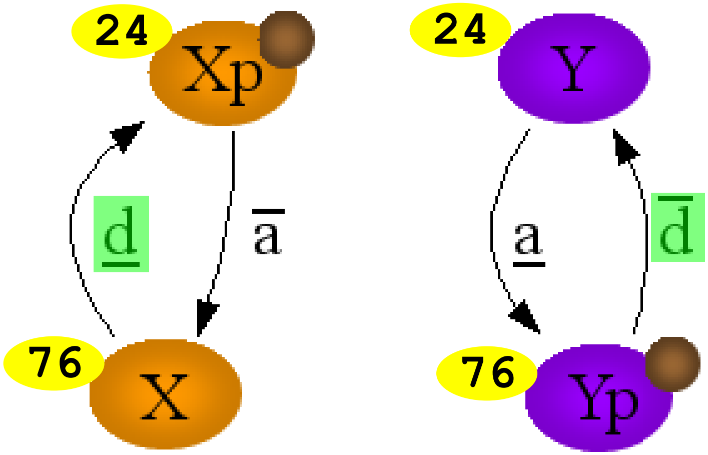
reaction	propensity (s^{-1})
a	100·99·99
d	10·1·1



- Eventually an equilibrium is reached...

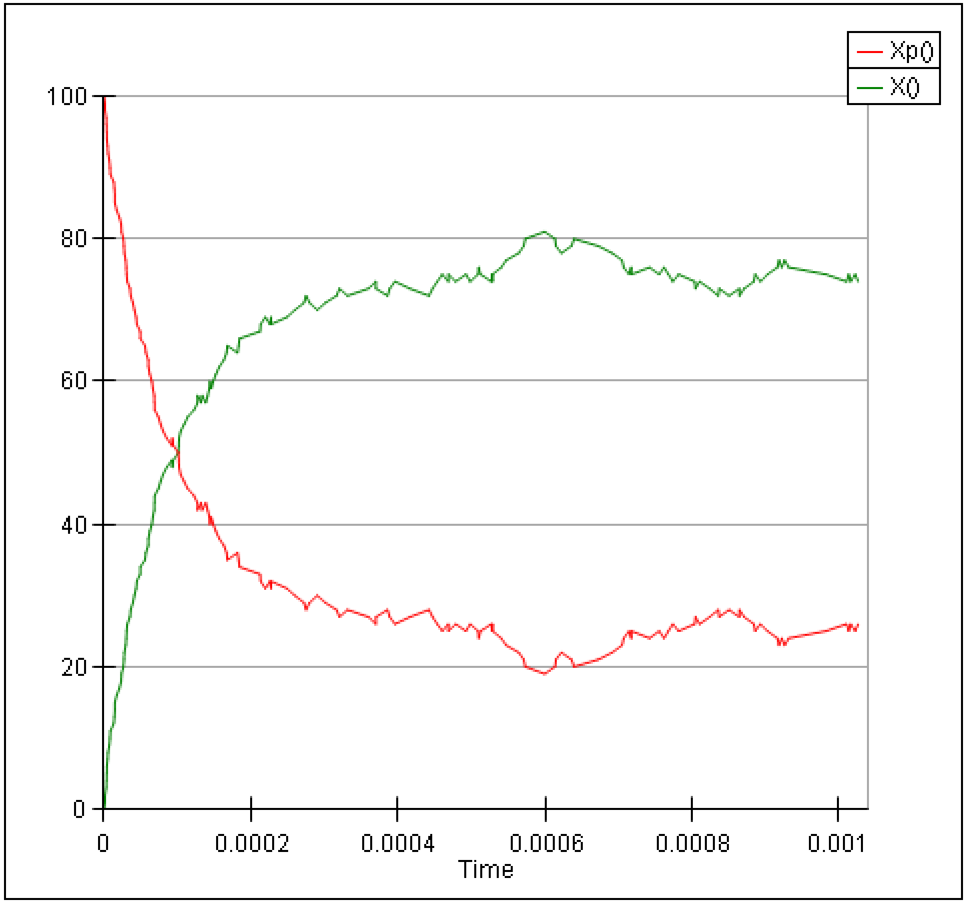
Interaction: $Xp + Y \;{}_{d}\!\leftrightarrow^{\mathrm{a}}\; X + Yp$

reaction	propensity (s^{-1})
a	100·24·24
d	10·76·76



- At equilibrium when rate(*a*)·[*Xp*][*Y*] ≈ rate(*d*)·[*X*][*Yp*]

Interaction: $Xp + Y \;{}_{d}\!\leftrightarrow^{a}\; X + Yp$

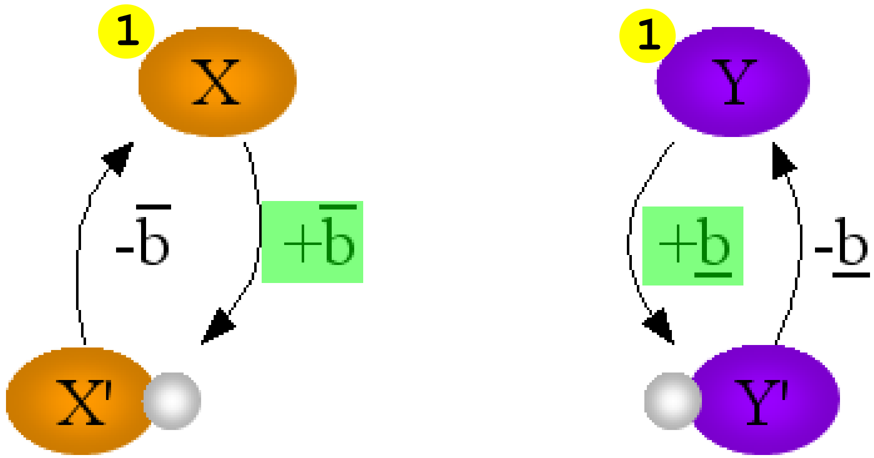


- At equilibrium: $100\text{s}^{-1}\cdot[Xp][Y] \approx 10\text{s}^{-1}\cdot[X][Yp]$
- Approximately $24\cdot Xp$ and $76\cdot X$

Binding: $X + Y \;{}_{-b}{\leftrightarrow}^{+b}\; X'Y'$

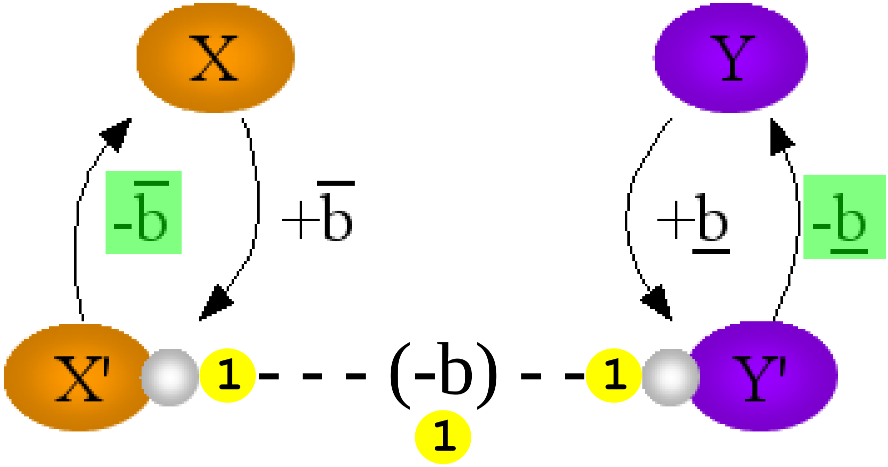
$X = {}^{+}\overline{b}.X'$ $\qquad$ $Y = {}^{+}\underline{b}.Y'$

$X' = {}^{-}\overline{b}.X$ $\qquad$ $Y' = {}^{-}\underline{b}.Y$



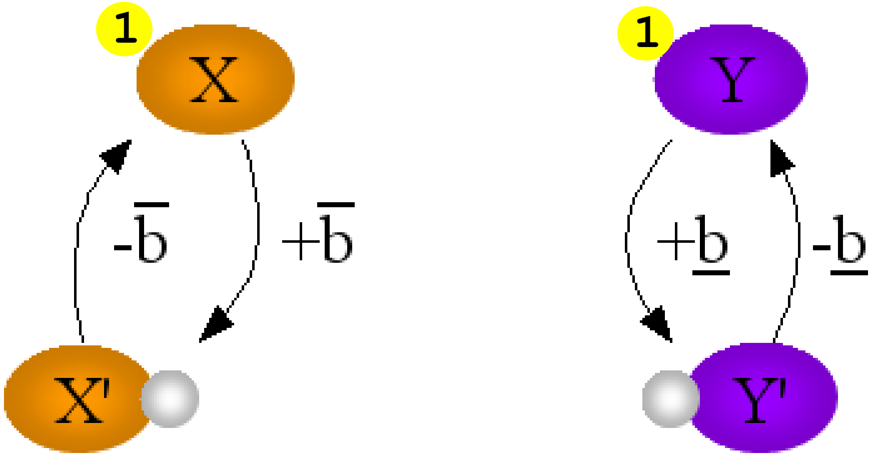
- *X* and *Y* can bind on channel ${}^{+}b$

Binding: $X + Y \;{}_{-b}\!\leftrightarrow^{+b}\; X'Y'$



- *X′* and *Y′* can unbind on channel ^{-}b

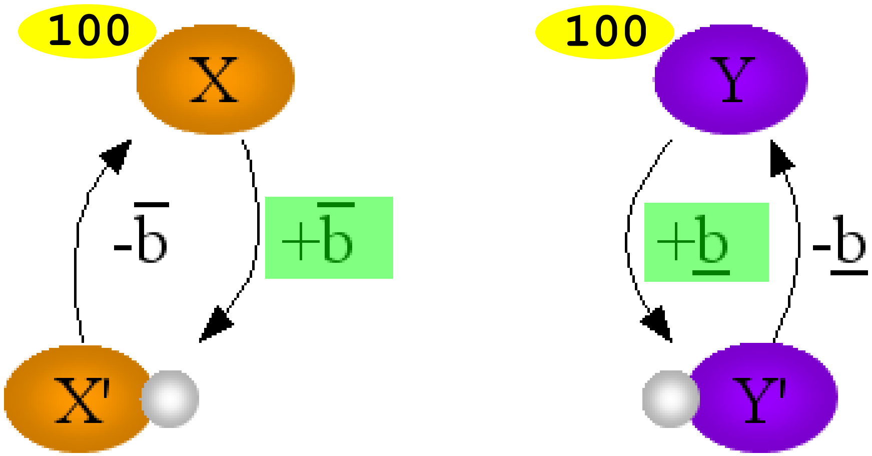
Binding: $X + Y \;{}_{-b}\!\leftrightarrow^{+b}\; X'Y'$



- Binding and unbinding can continue indefinitely...

Binding: $X + Y \ {}_{\text{-b}}\!\leftrightarrow^{\text{+b}} \ X'Y'$

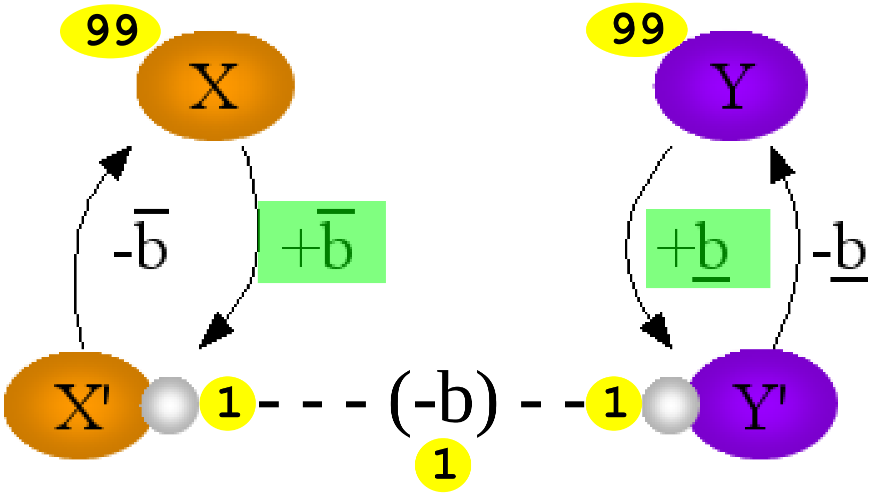
reaction	propensity (s^{-1})
^{+}b	100·100·100
^{-}b	0



- What happens if we mix 100·*X* and 100·*Y* ?
- Assume *rate*(^{+}b) = 100s^{-1} and *rate*(^{-}b) = 10s^{-1}
- An *X* and *Y* protein can bind on channel ^{+}b.

Binding: $X + Y \;_{-b}\!\leftrightarrow^{+b}\; X'Y'$

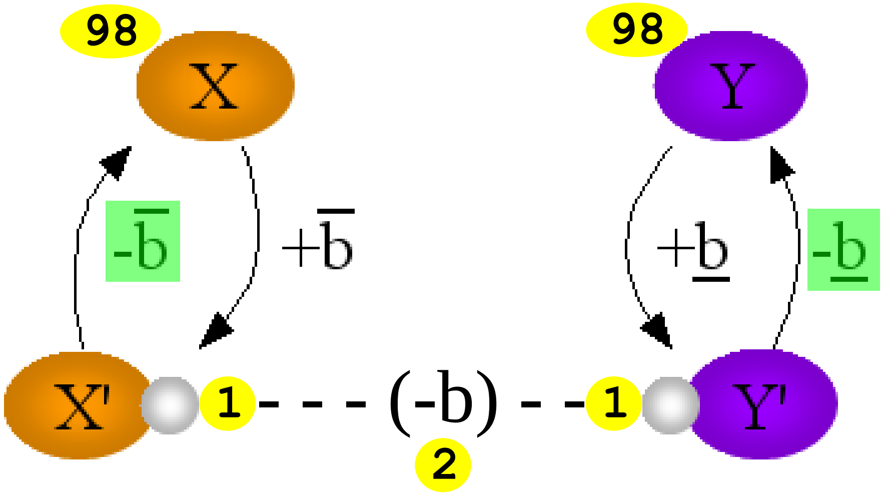
reaction	propensity (s^{-1})
^{+}b	100·99·99
^{-}b	10·1



- An additional *X* and *Y* protein can bind.

Binding: $X + Y \ {}_{-b}\!\leftrightarrow^{+b}\ X'Y'$

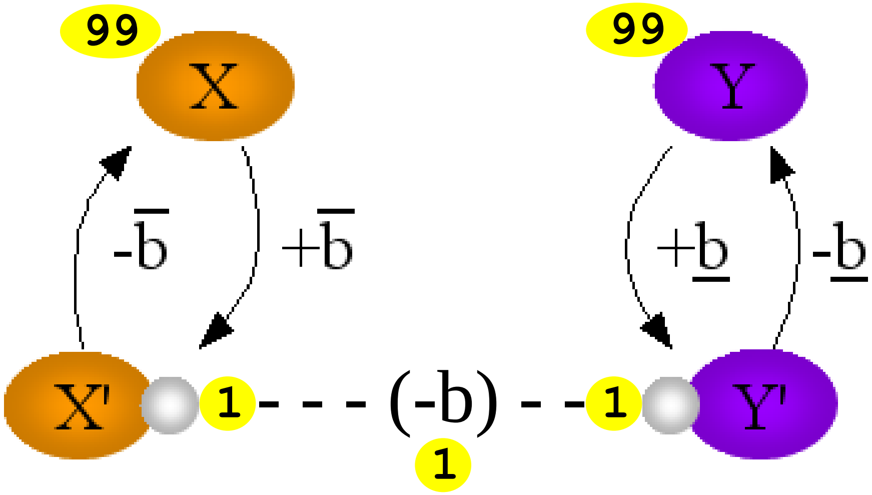
reaction	**propensity (s^{-1})**
$^{+}$b	100·98·98
$^{-}$b	10·2



- An *X'* and *Y'* protein can unbind on channel ^{-}b

Binding: $X + Y \;{}_{-b}\!\leftrightarrow^{+b}\; X'Y'$

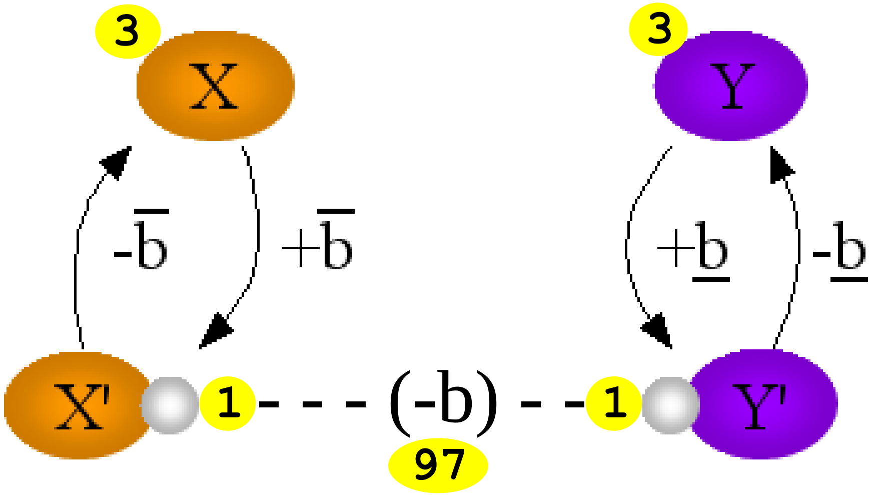
reaction	propensity (s^{-1})
$^{+}$b	100·99·99
$^{-}$b	10·1



- Eventually...

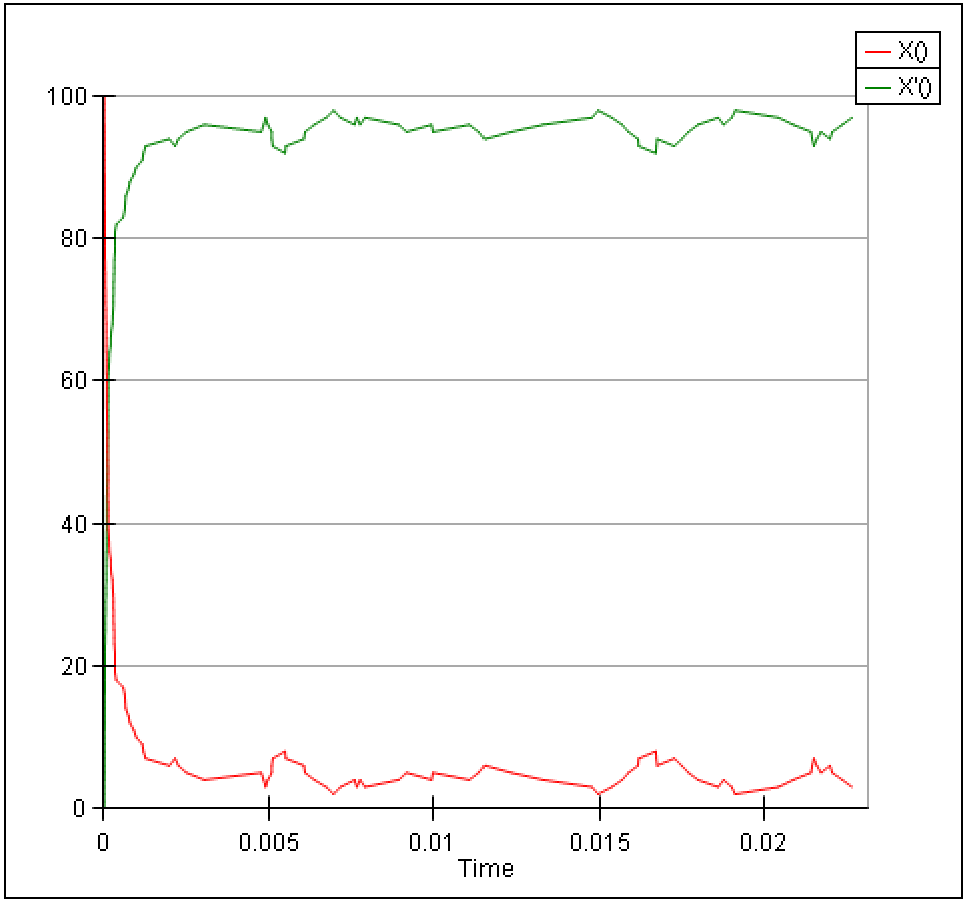
Binding: $X + Y \ {}_{\text{-b}}\!\leftrightarrow^{\text{+b}} \ X'Y'$

reaction	propensity (s^{-1})
^{+}b	100·3·3
^{-}b	10·97



- At equilibrium when rate(^{+}b)·[X][Y] ≈ rate(^{-}b)·(^{-}b) ([X'] [Y'])

Binding: $X + Y \;{}_{-b}{\leftrightarrow}^{+b}\; X'Y'$



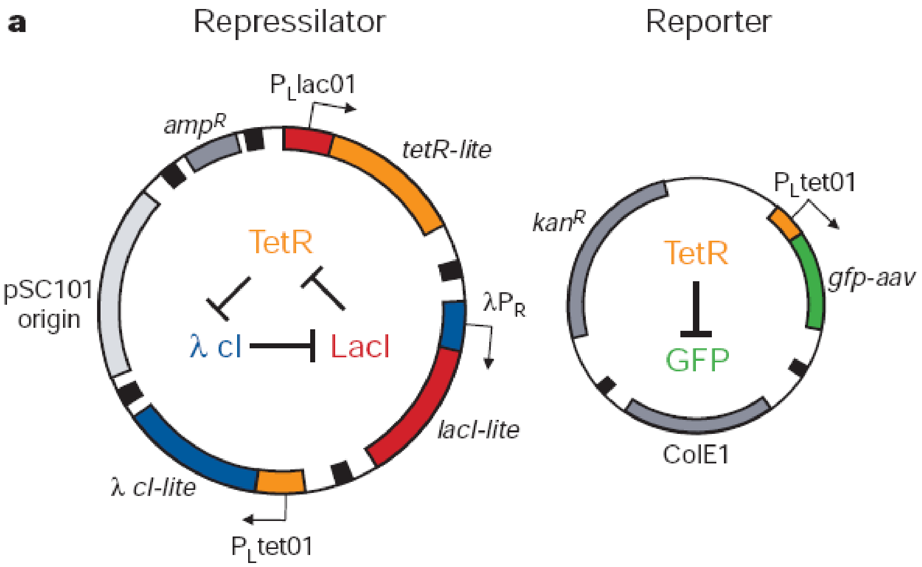
- At equilibrium: $100\text{s}^{-1}\cdot[X][Y] = 10\text{s}^{-1}\cdot[X'Y']$
- Approximately $3\cdot X$ and $97\cdot X'Y'$

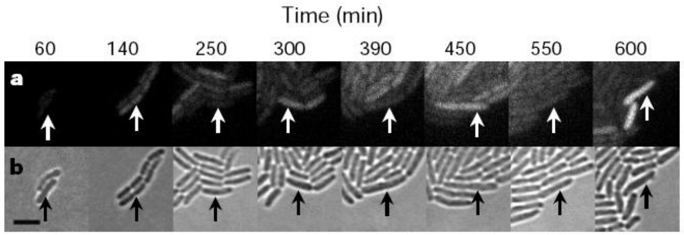
Programming Gene Networks

with
Luca Cardelli (MSR Cambridge)
Ralf Blossey (IRI Lille)

Repressilator [Elowitz and Leibler, 2000]

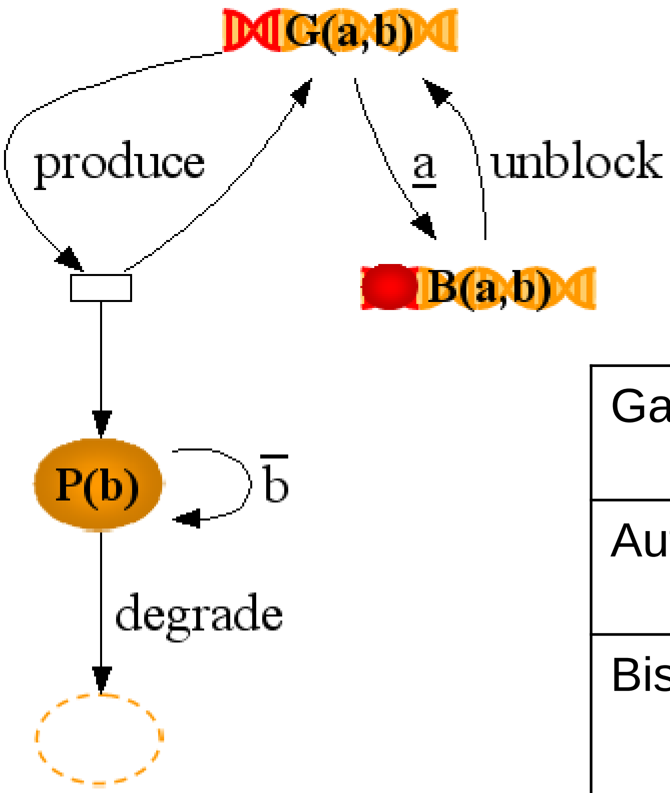
- A gene network engineered in live bacteria.



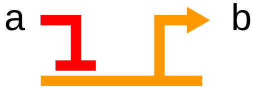
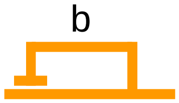
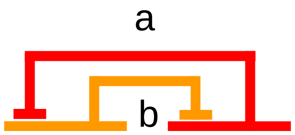
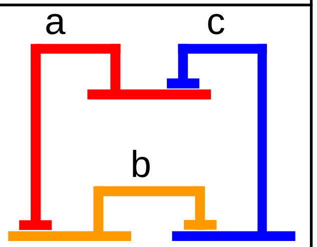


© 2000 Elowitz, M.B., Leibler. S. A Synthetic Oscillatory Network of Transcriptional Regulators. Nature 403:335-338.

Parameterised Gene Gate

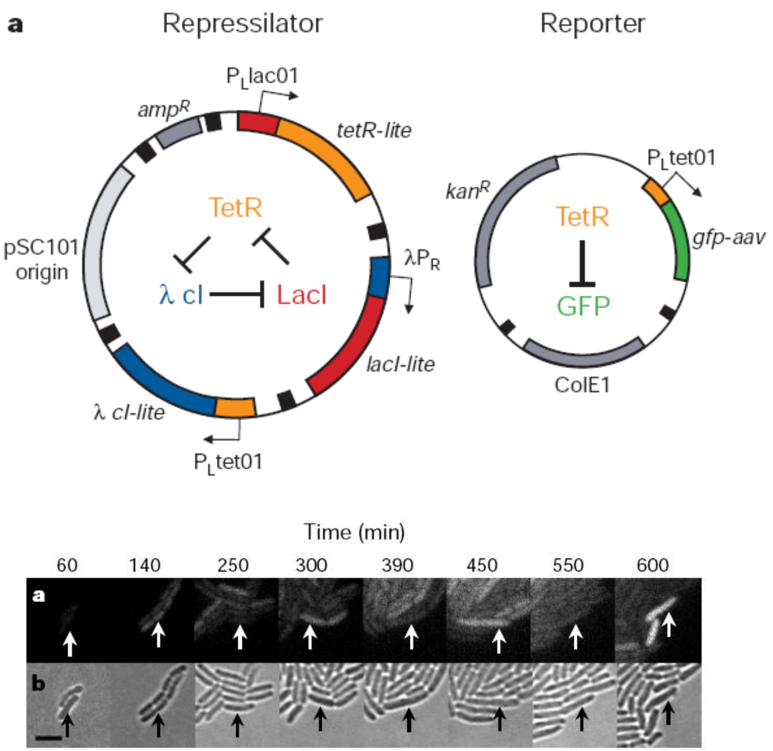


G(a,b) = $\underline{a}$.B(a,b) + produce.(P(b) | G(a,b))
B(a,b) = unblock.G(a,b)
P(b) = $\overline{b}$.P(b) + degrade.0

Gate	G(a,b)	
Autoinhibitory	G(b,b)	
Bistable	G(a,b) | G(b,a)	
Oscillator	G(a,b) | G(b,c) | G(c,a)	

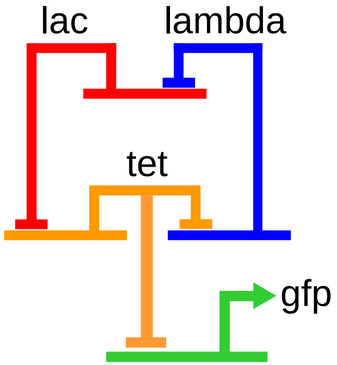
Repressilator [Elowitz and Leibler, 2000]

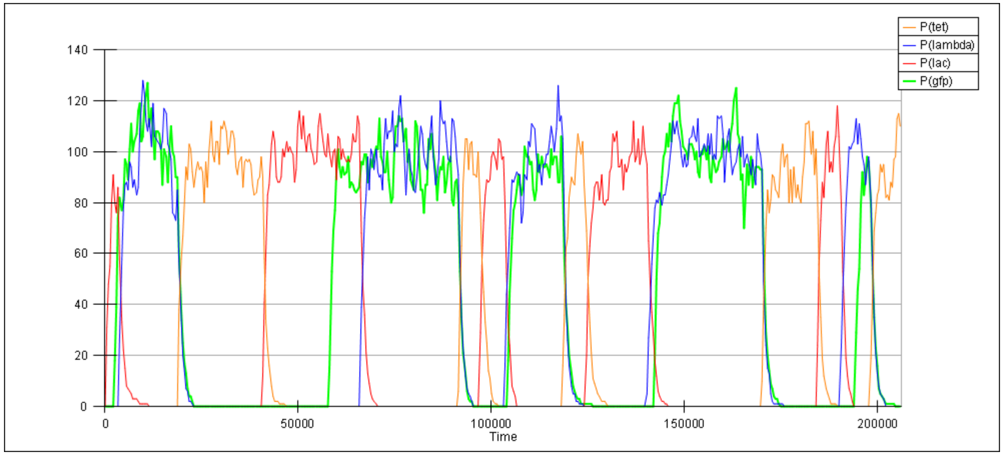
- Modelled as a simple combination of gene gates:



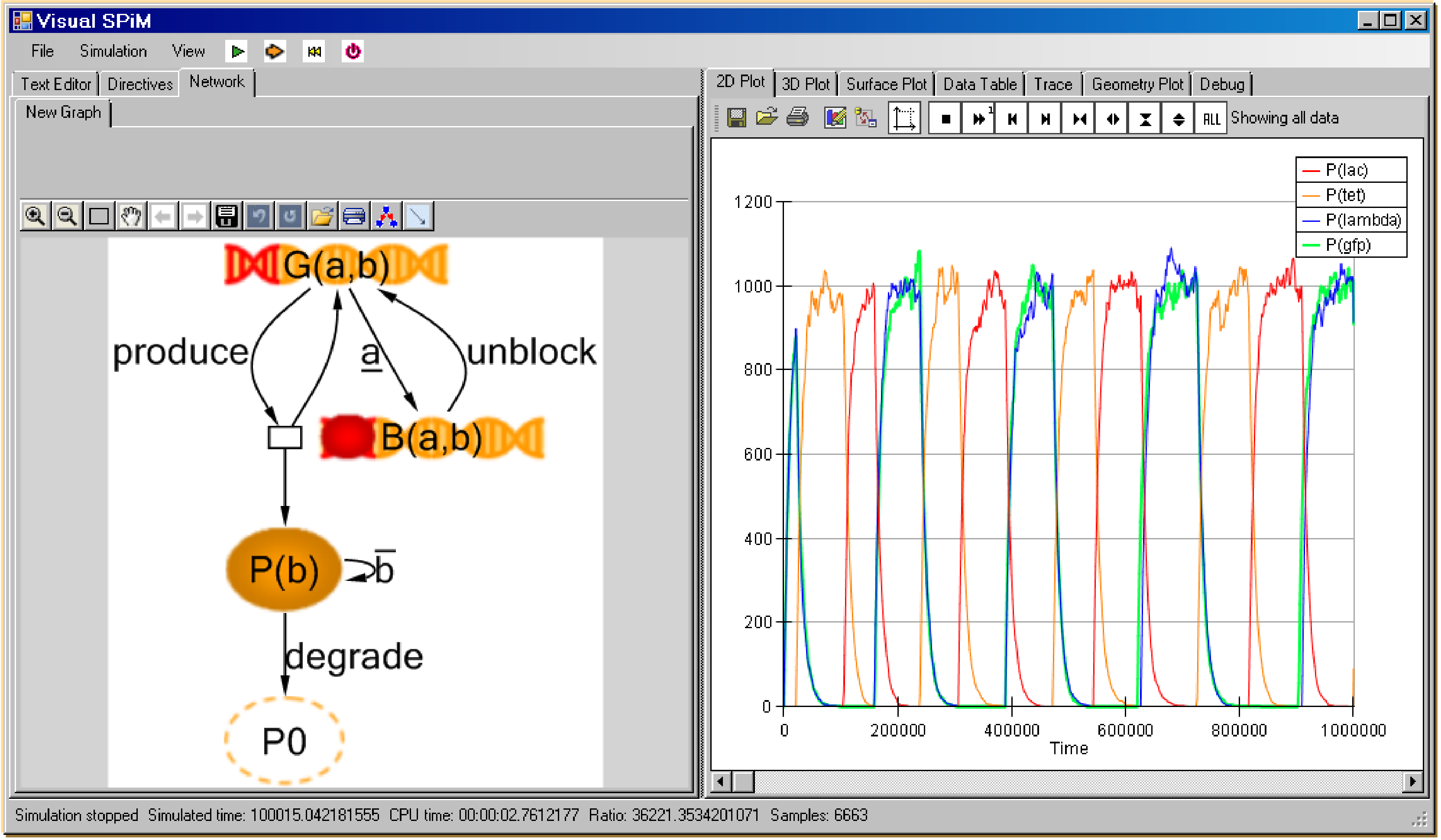
© 2000 Elowitz, M.B., Leibler. S. A Synthetic Oscillatory Network of Transcriptional Regulators. Nature 403:335-338.

G(lac,tet)
| G(tet,lambda)
| G(lambda,lac)
| G(tet,gfp)





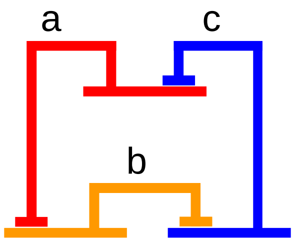
Graphical Programming

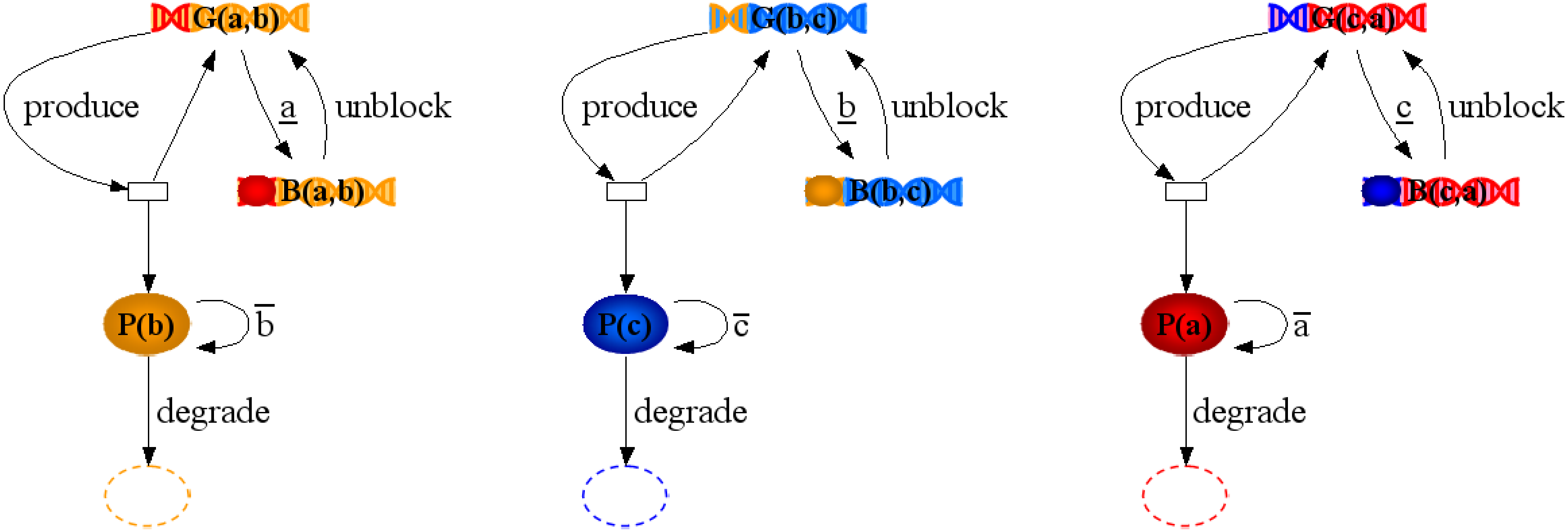


```
Simulation stopped  Simulated time: 100015.042181555  CPU time: 00:00:02.7612177  Ratio: 36221.3534201071  Samples: 6663
```

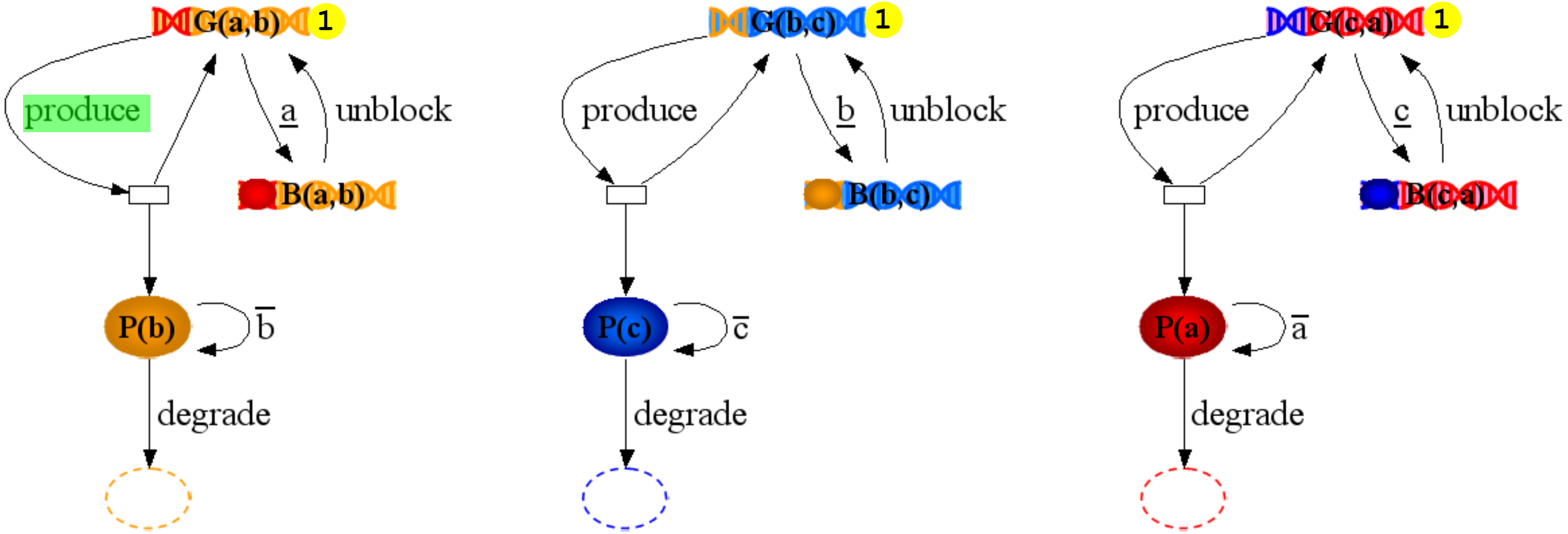
How does the oscillator work?

- G(a,b) | G(b,c) | G(c,a)





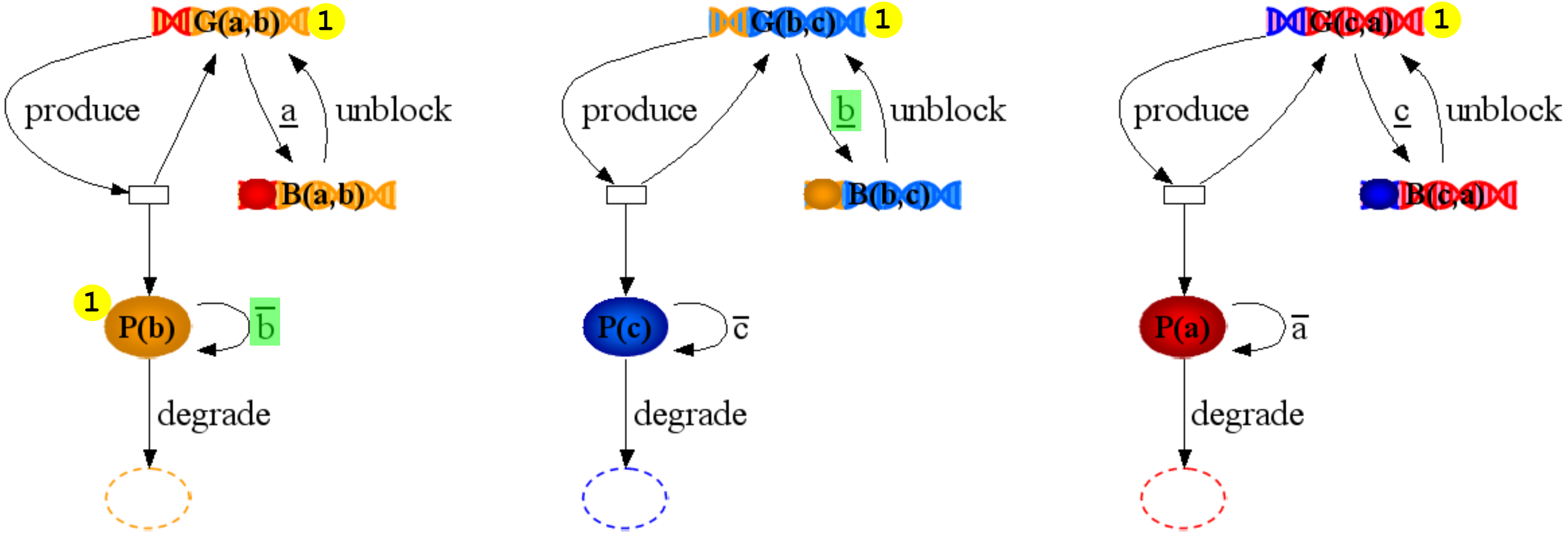
Oscillator: 0s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1*3	0	0	0	0	0	0.3

- Initially there is one copy of each gene
- Any one of the proteins can be produced at rate 0.1

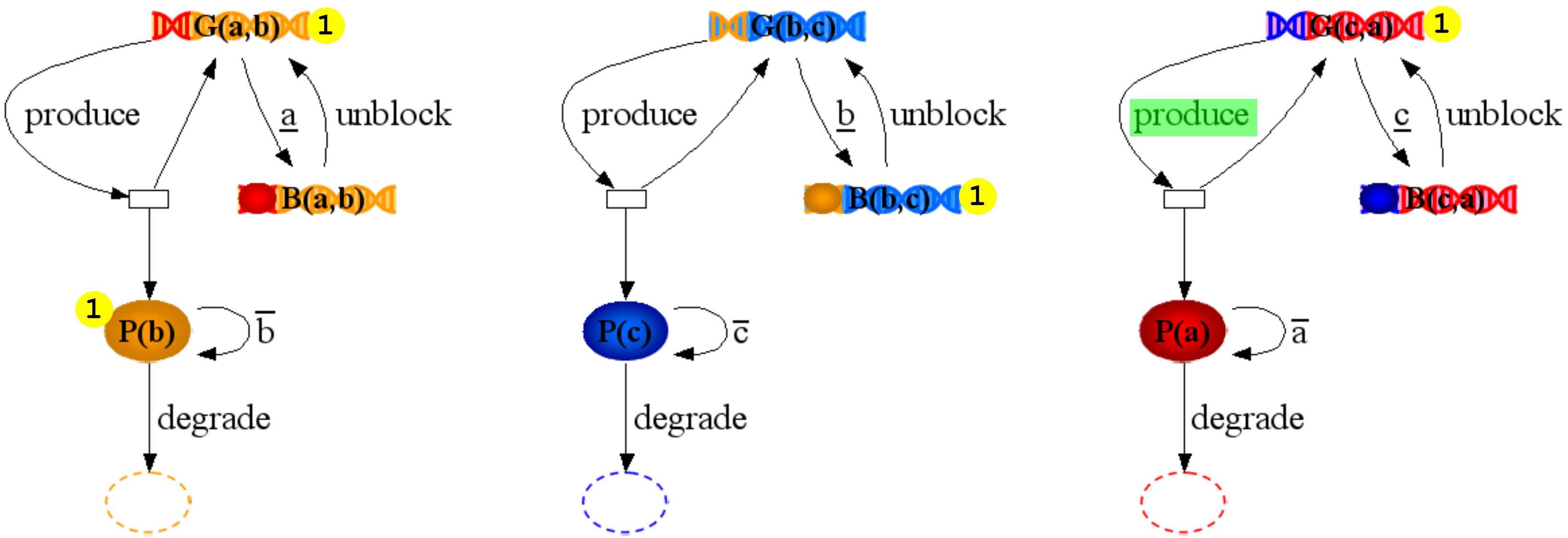
Oscillator: 5.568177s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1*3	0	0.001	1	0	0	1.301

- The *b* protein can block the *c* gene at rate 1

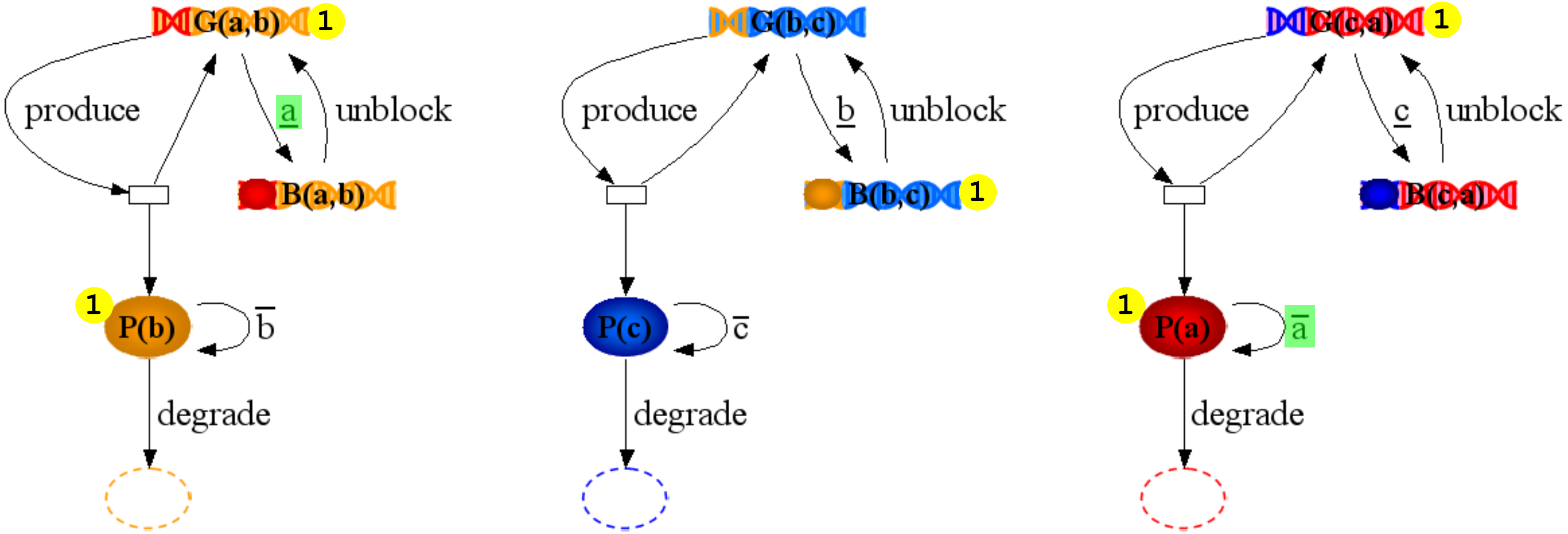
Oscillator: 6.329912s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1*2	0.0001	0.001	0	0	0	0.2011

- Now no *c* protein can be produced.
- But an *a* protein can still be produced at rate 0.1

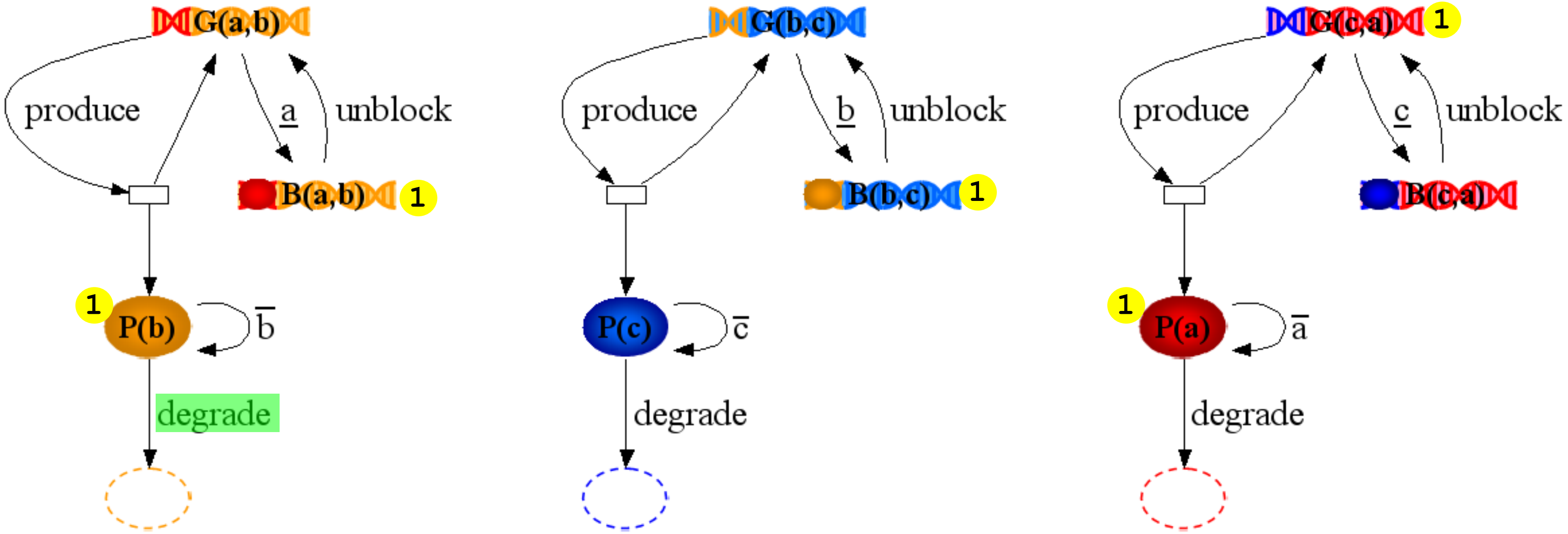
Oscillator: 11.62149s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1*2	0.0001	0.001*2	0	0	1.0	1.2021

- The *a* protein can block the *b* gene at rate 1

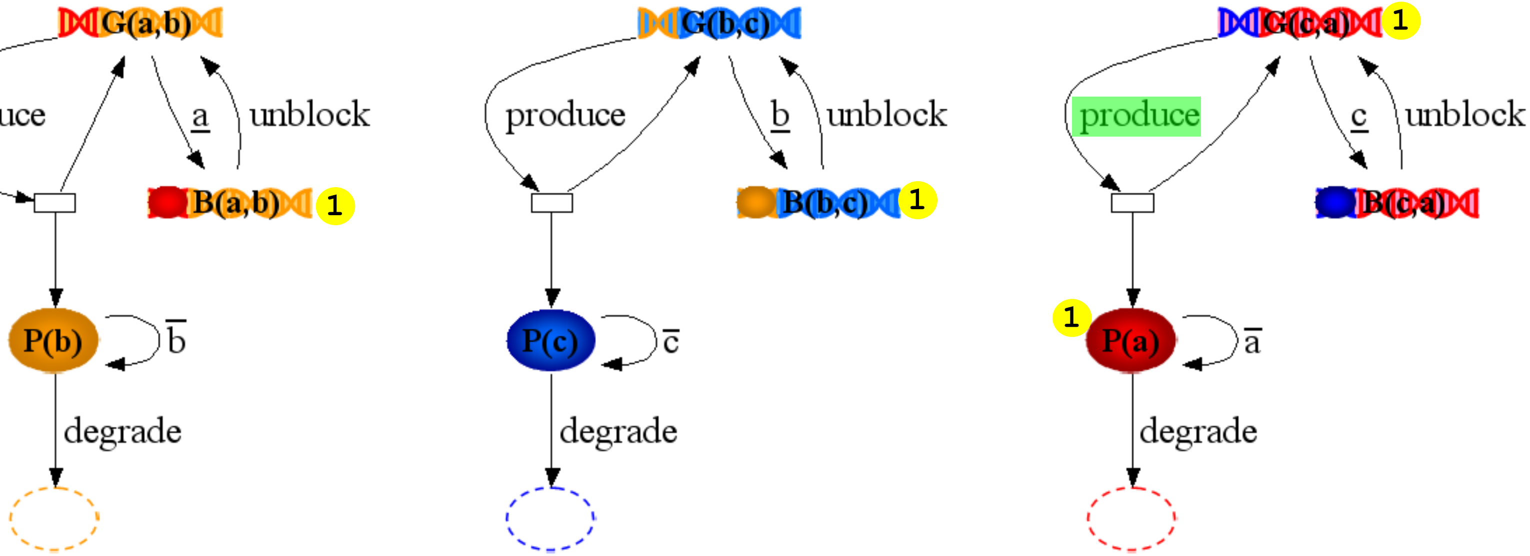
Oscillator: 13.21617s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1	0.0001*2	0.001*2	0	0	0	0.1022

- Now no *b* or *c* protein can be produced.
- A *b* protein can degrade at rate 0.001

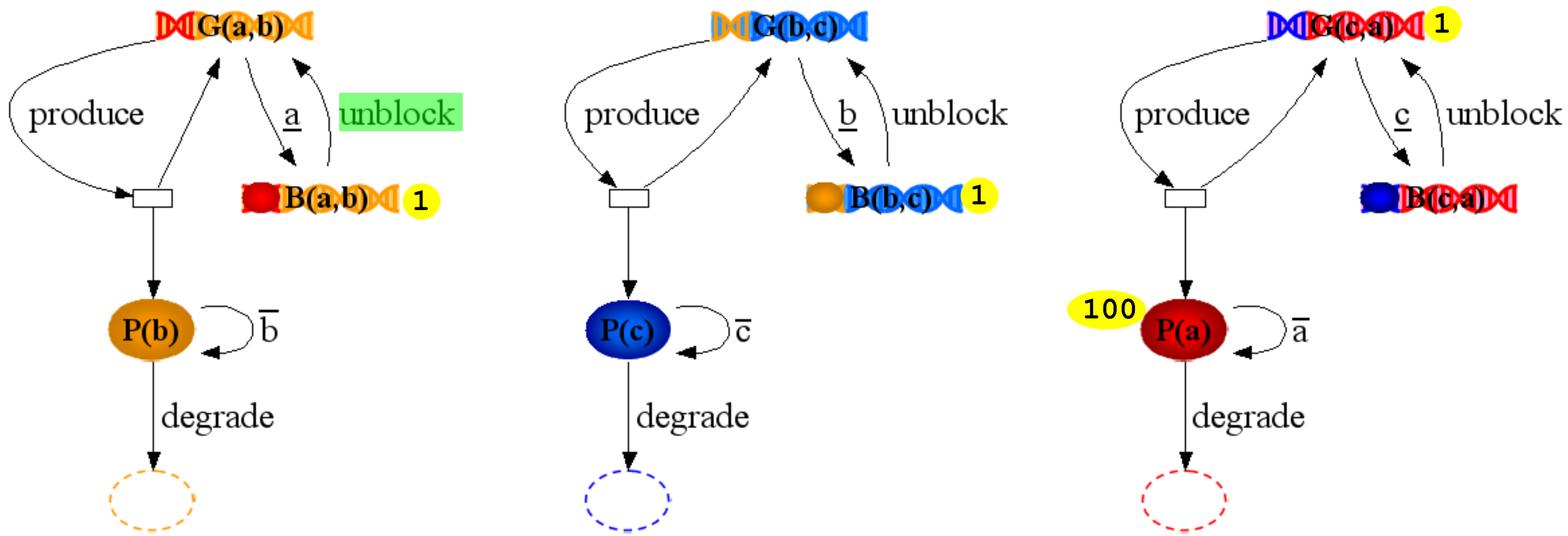
Oscillator



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1	0.0001*2	0.001	0	0	0	0.1012

- Meanwhile, lots of *a* protein is produced

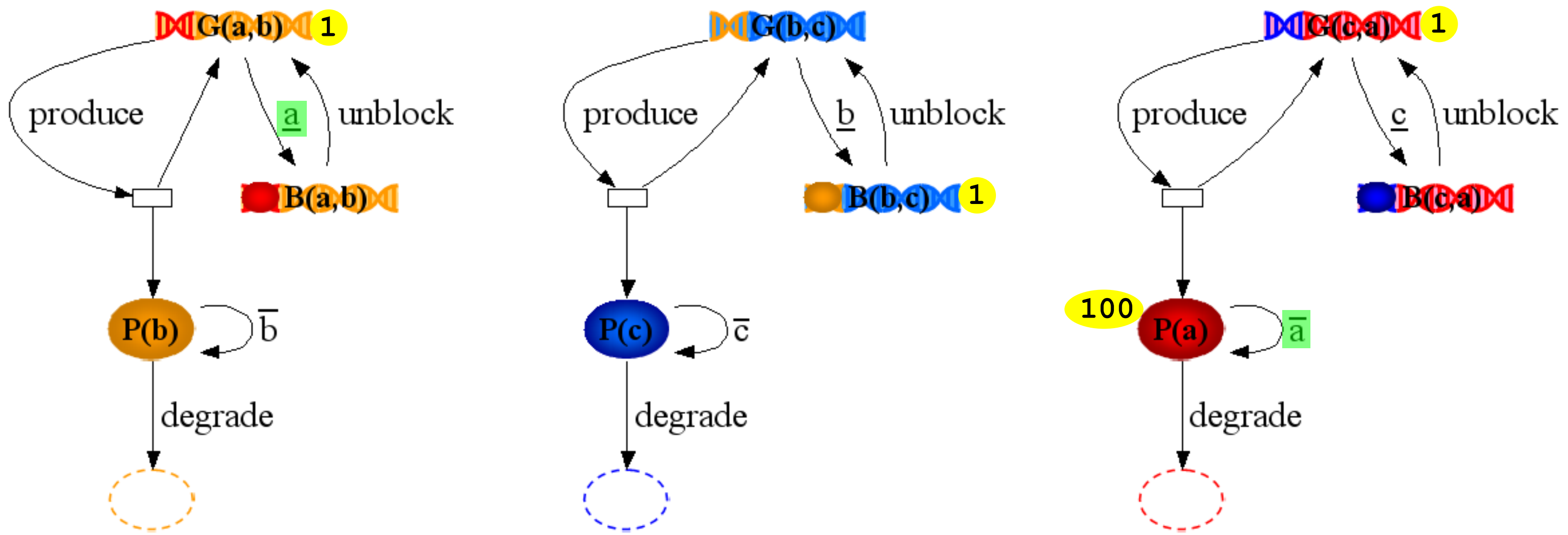
Oscillator



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1	0.0001*2	0.001*100	0	0	0	0.2002

- The *a* protein dominates
- Equilibrium between transcription and degradation
- Eventually, the *c* or *a* gene unblocks at rate 0.0001

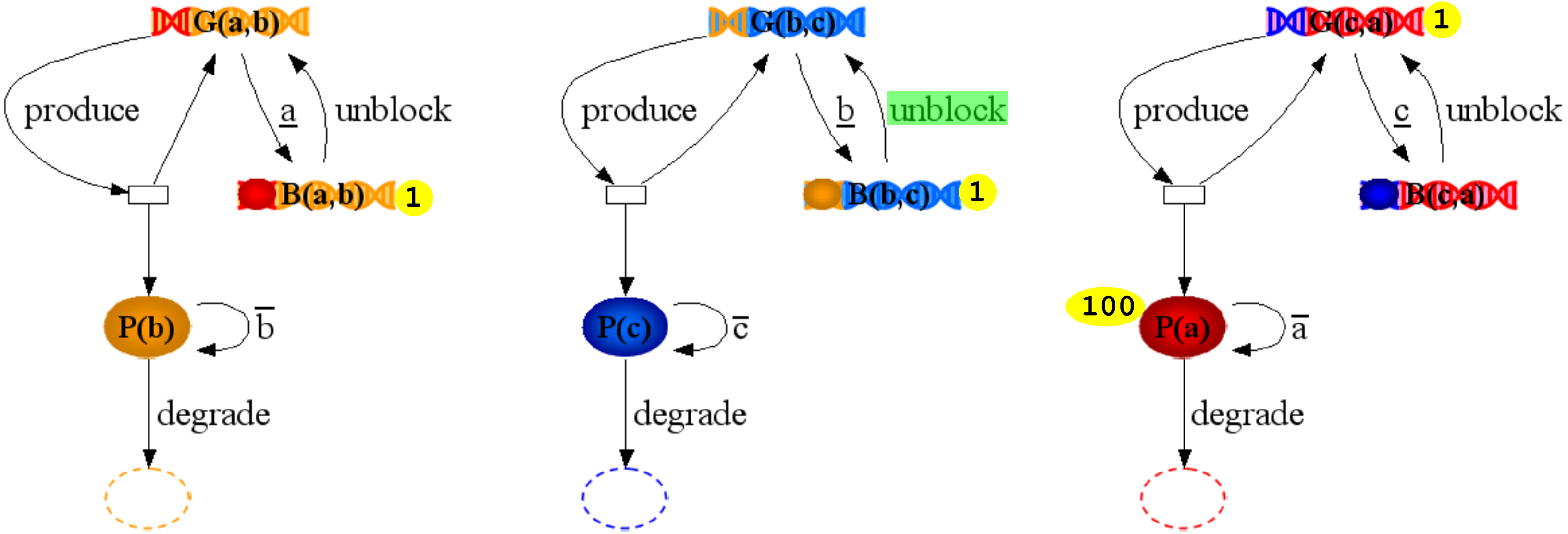
Oscillator



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1*2	0.0001	0.001*100	0	0	1.0*100	100.3001

- Suppose the *a* gene unblocks
- There is a high probability that it will block immediately

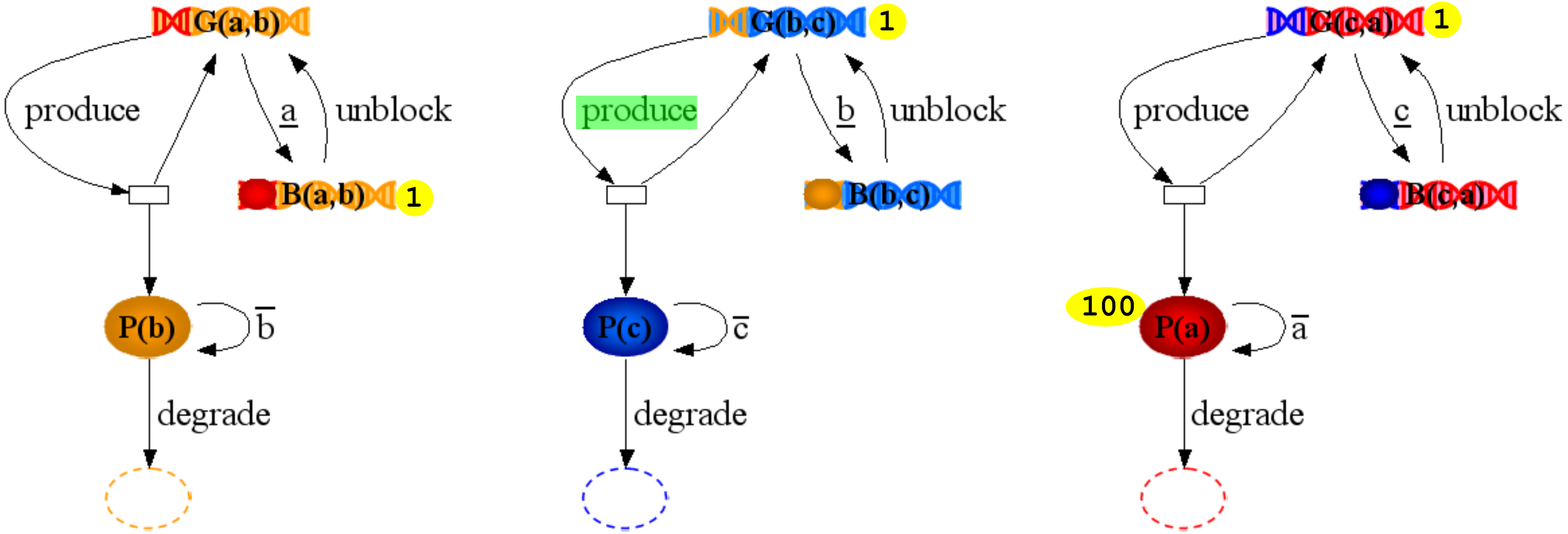
Oscillator: 11039.31s



reaction	produce	unblock	degrade	b	c	a	total
propensity (s^{-1})	0.1	0.0001*2	0.001*100	0	0	0	0.2002

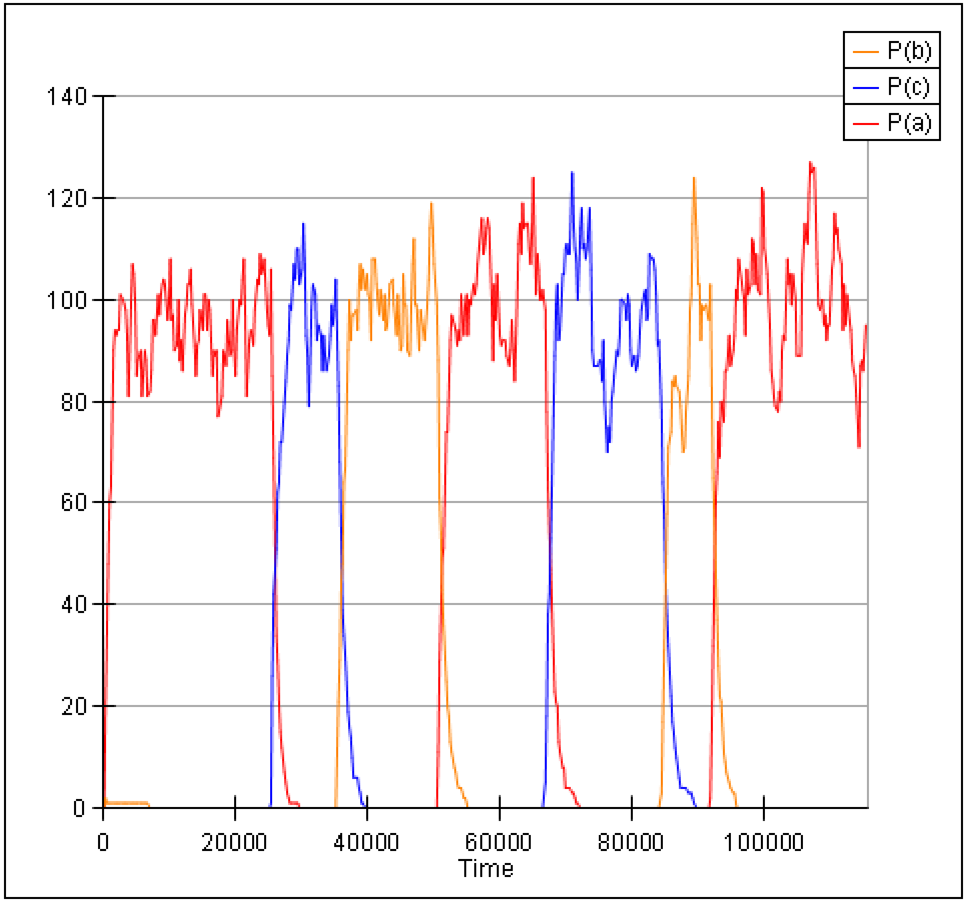
- Eventually, the *c* gene unblocks at rate 0.0001

Oscillator: 11039.77s



- There is nothing to block the *c* gene.
- The *c* protein can now take over...
- Eventually...

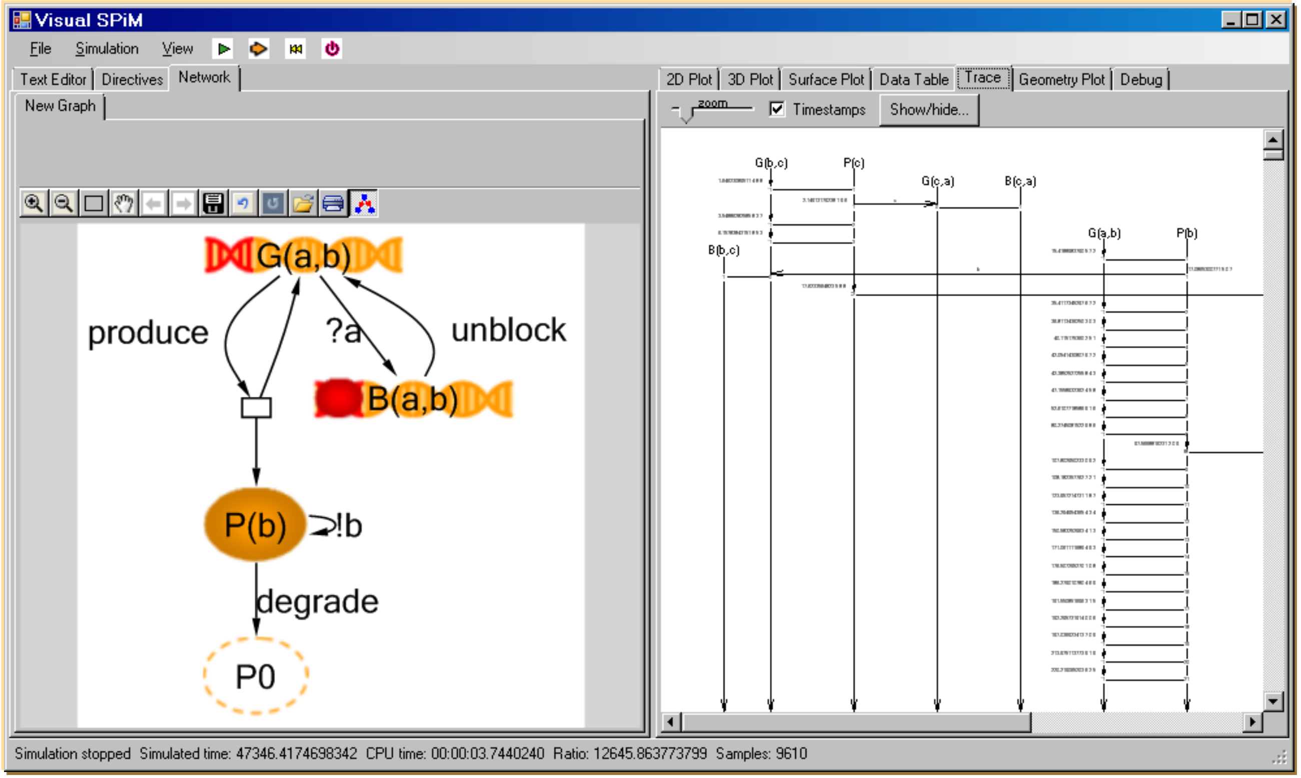
Oscillator Simulation



- Alternate oscillation of proteins: *b*, *c*, *a*, *b*
- Oscillations in a particular order

Analysing Simulation Traces

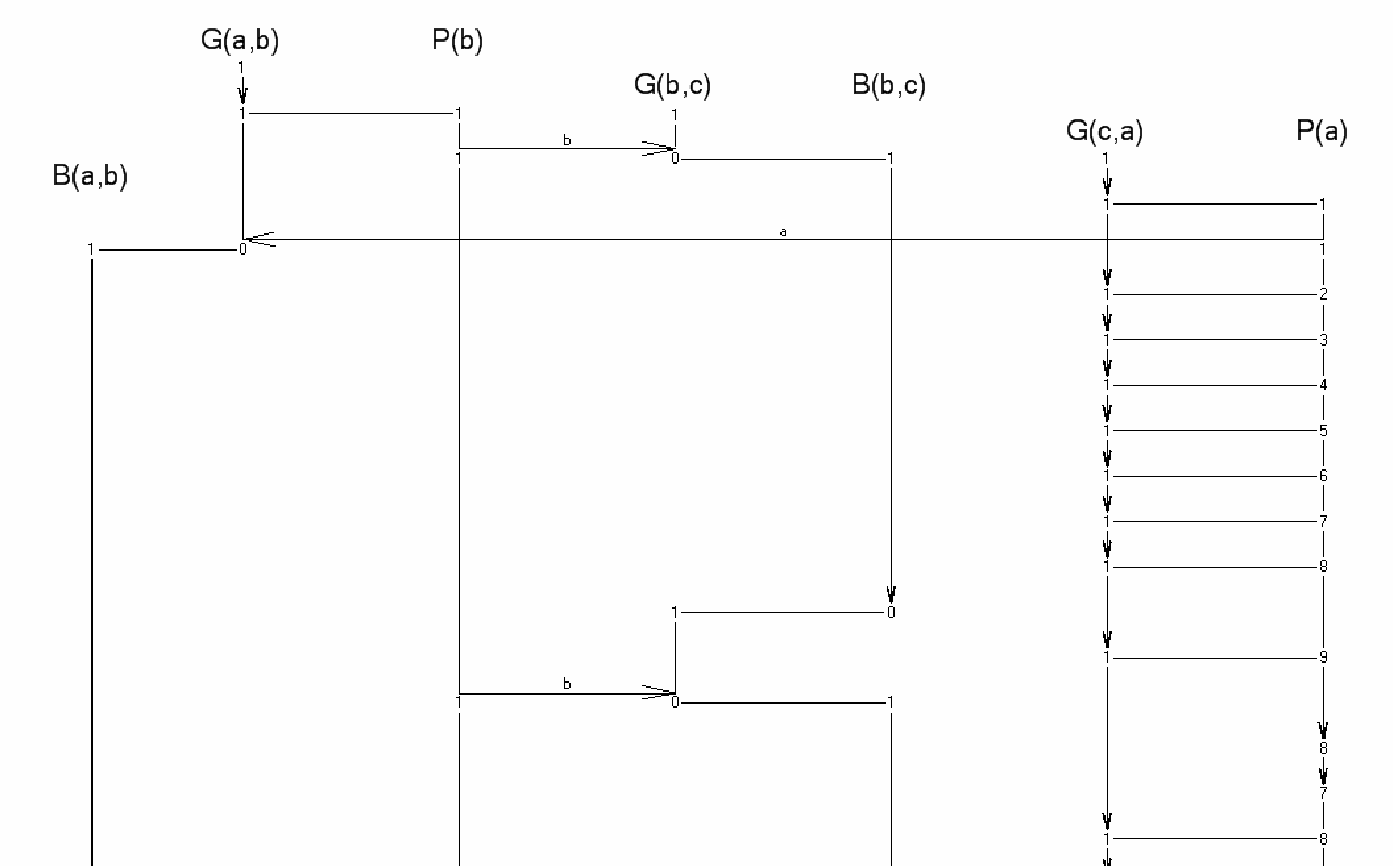
- A simulation trace can be visualised as a sequence of messages exchanged between parallel processes.
- Can debug a biological system in a similar way to a communication protocol. Causality, critical paths...



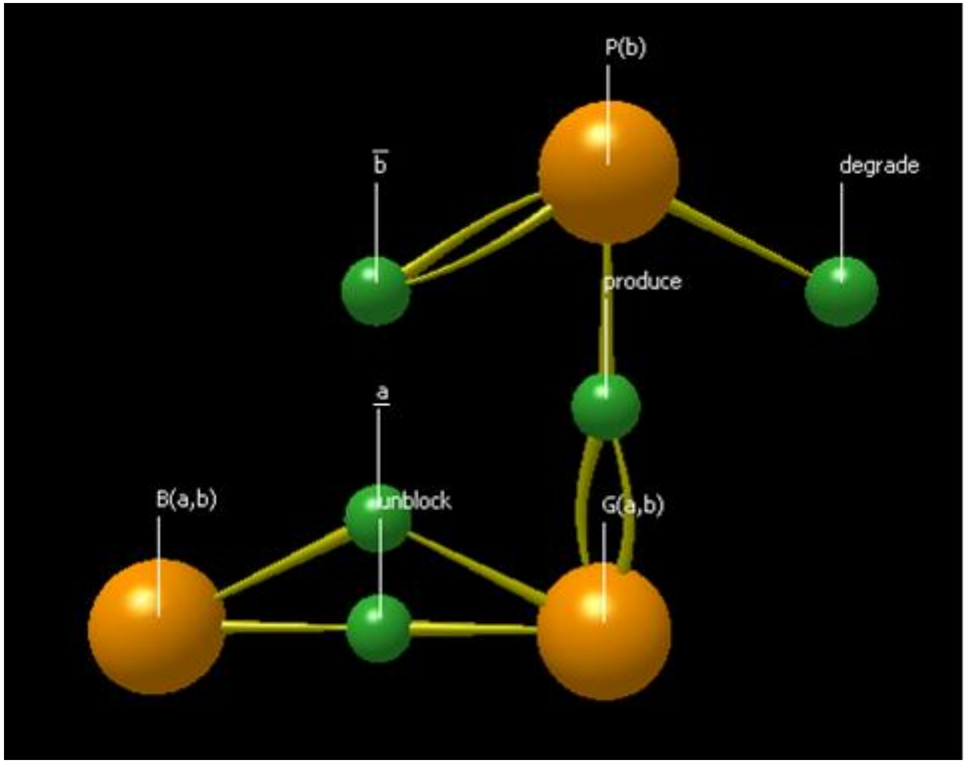
Simulation stopped Simulated time: 47346.4174698342 CPU time: 00:00:03.7440240 Ratio: 12645.863773799 Samples: 9610

Borgstroem, Gordon, Phillips, 2007

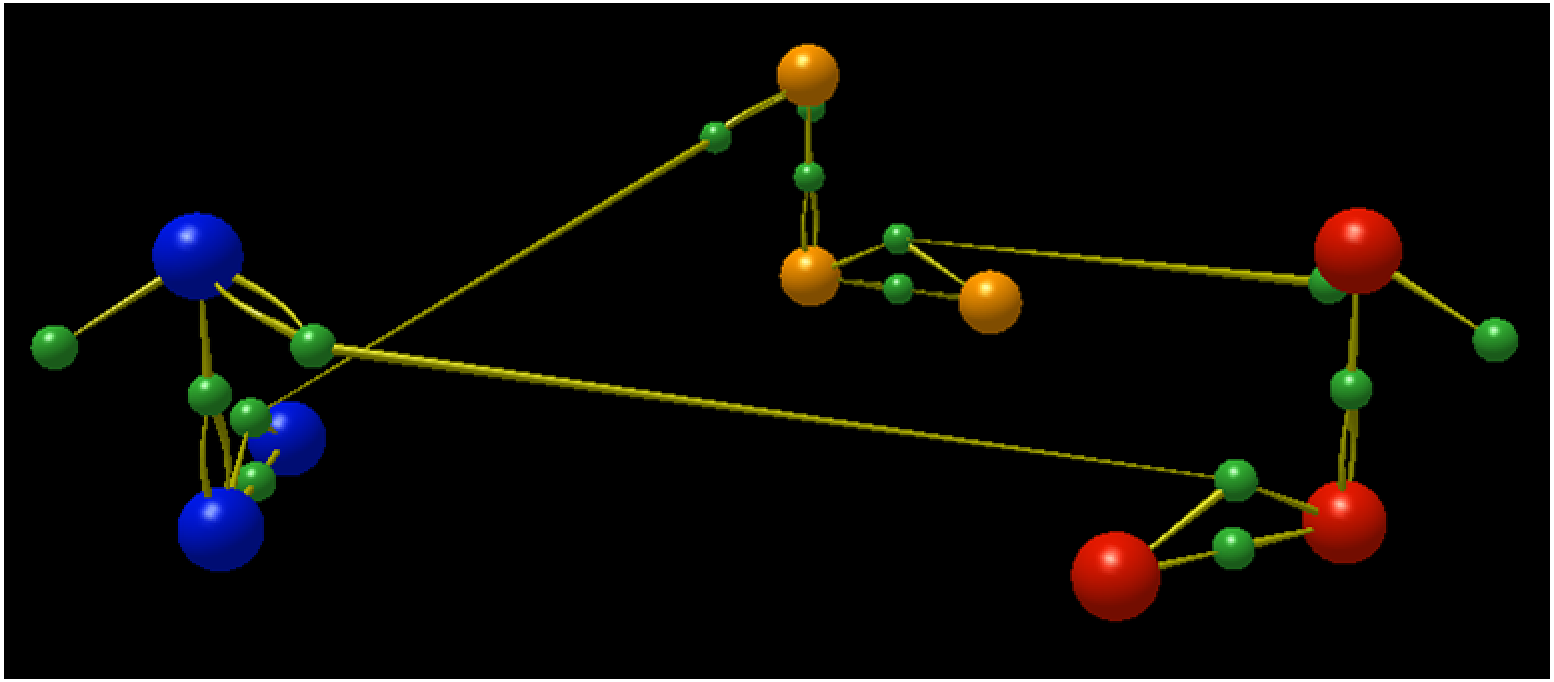
Repressilator Trace



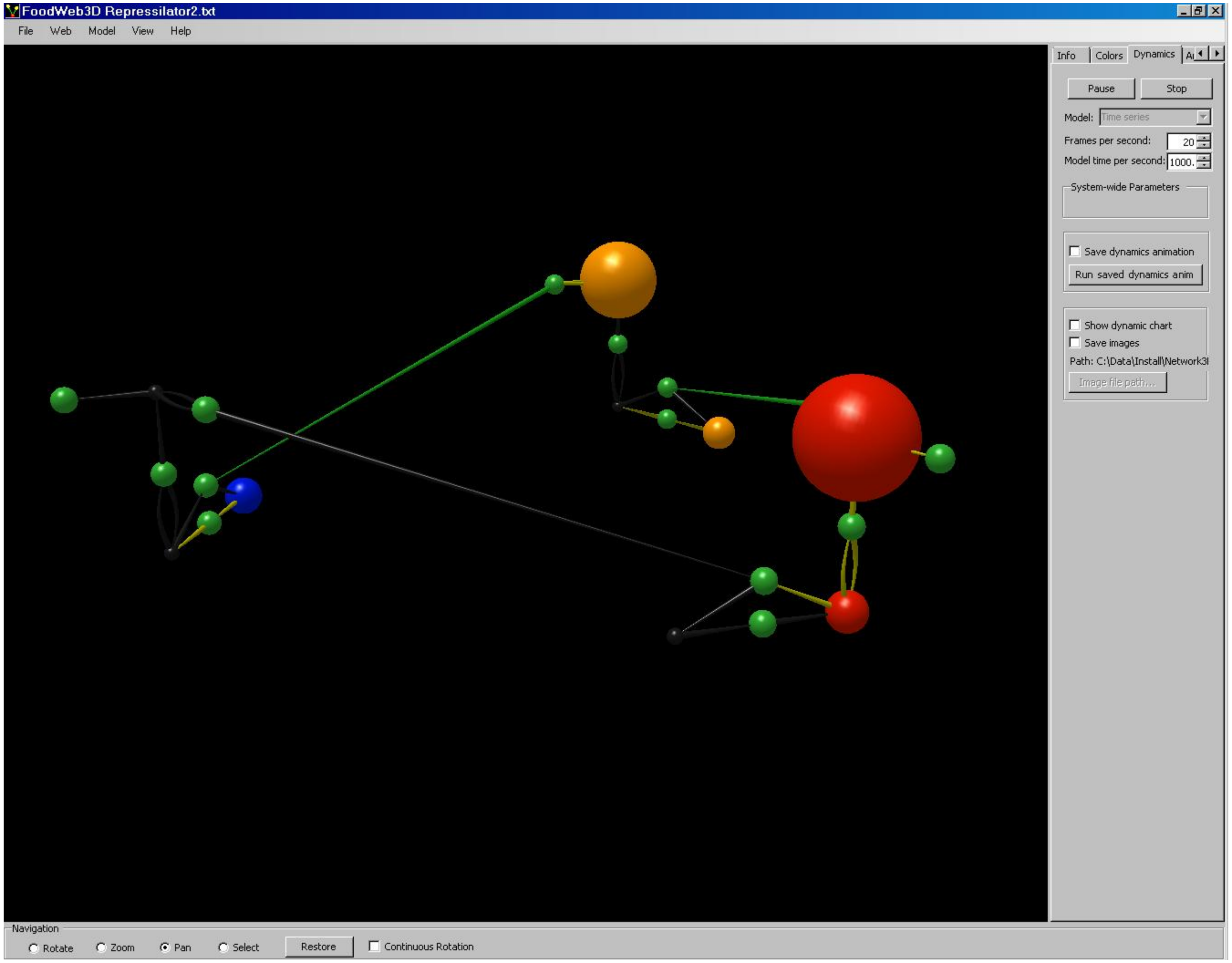
A Gene Gate in 3D



The Repressilator in 3D



Graphical Debugging



Model: Time series

Frames per second: 20

Model time per second: 1000.

□ Save dynamics animation

□ Show dynamic chart

□ Save images

Path: C:\Data\Install\Network3I

○ Rotate ○ Zoom ● Pan ○ Select

□ Continuous Rotation

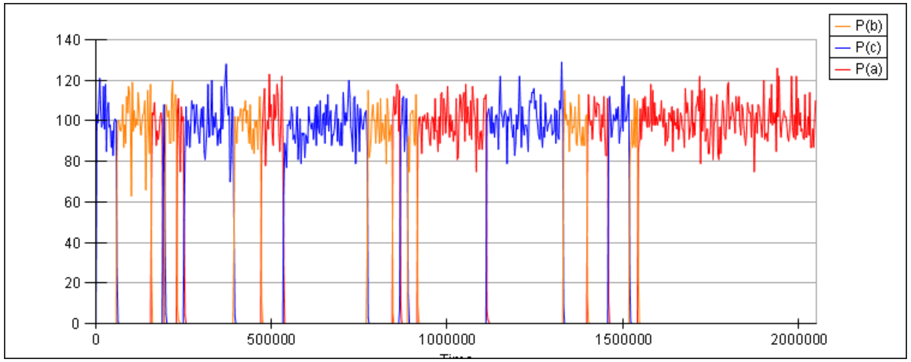
GUI by Rich Williams, MSR Cambridge

Parameter Analysis

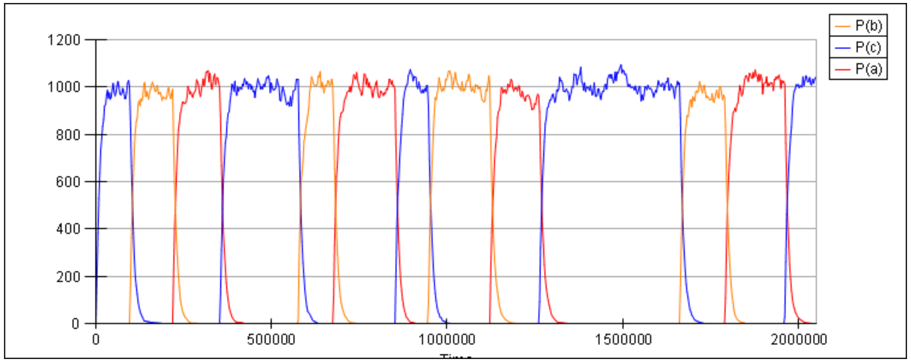
Range of parameters for good oscillations (produce=0.1):

produce/degrade > 1000, unblock > degrade, block > 100*produce

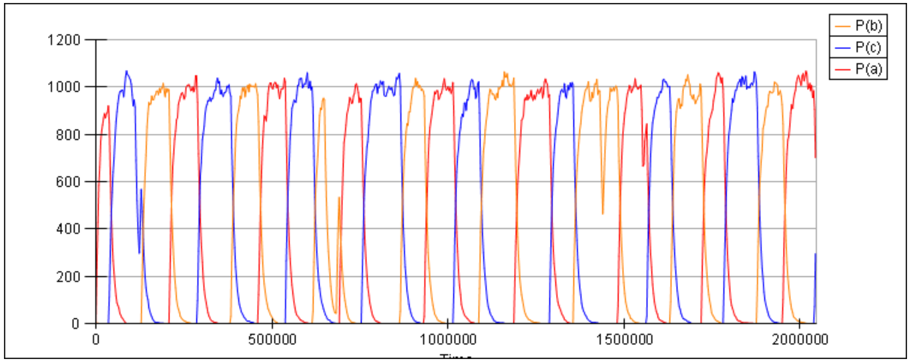
degrade = 0.001, unblock = 0.00001, block = 1.0



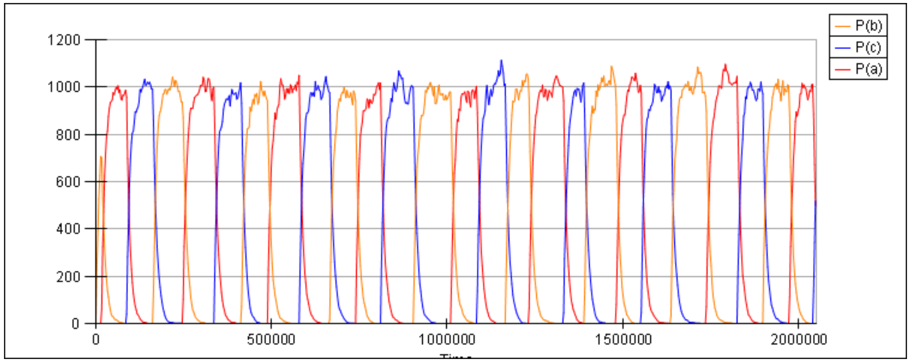
degrade = 0.0001, unblock = 0.00001, block = 1.0



degrade = 0.0001, unblock = 0.0001, block = 1.0



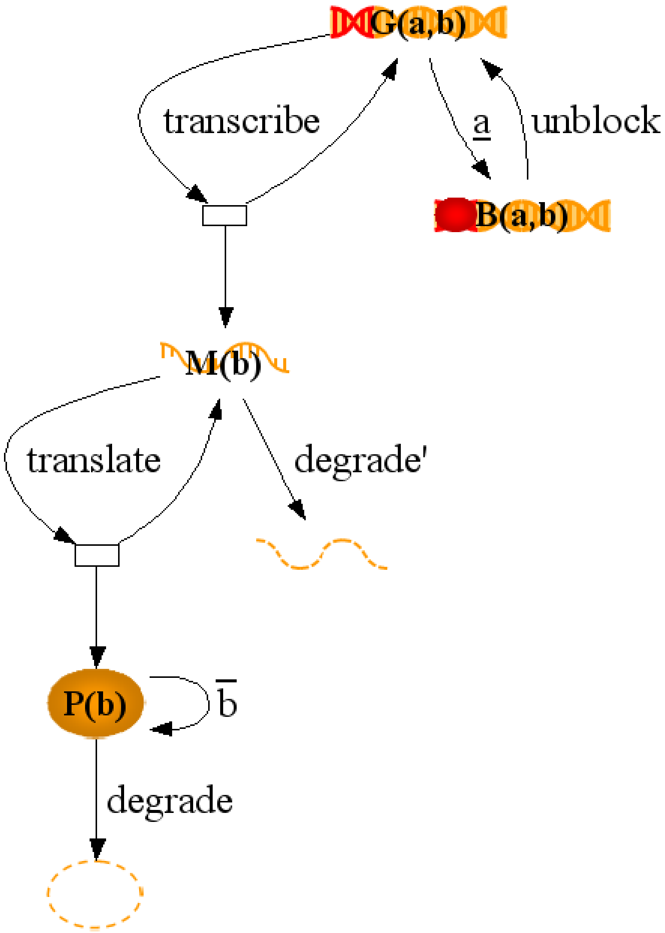
degrade = 0.0001, unblock = 0.0001, block = 10.0

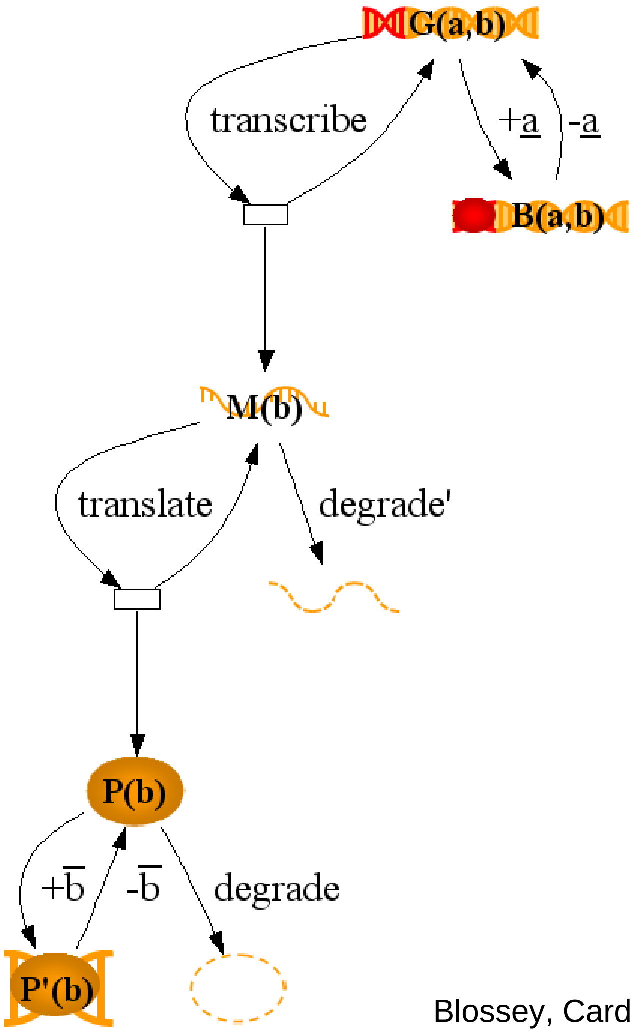


Blossey, Cardelli, Phillips, 2007

Model Refinement:

- Different behaviour, same network G(a,b) | G(b,c) | G(c,a)

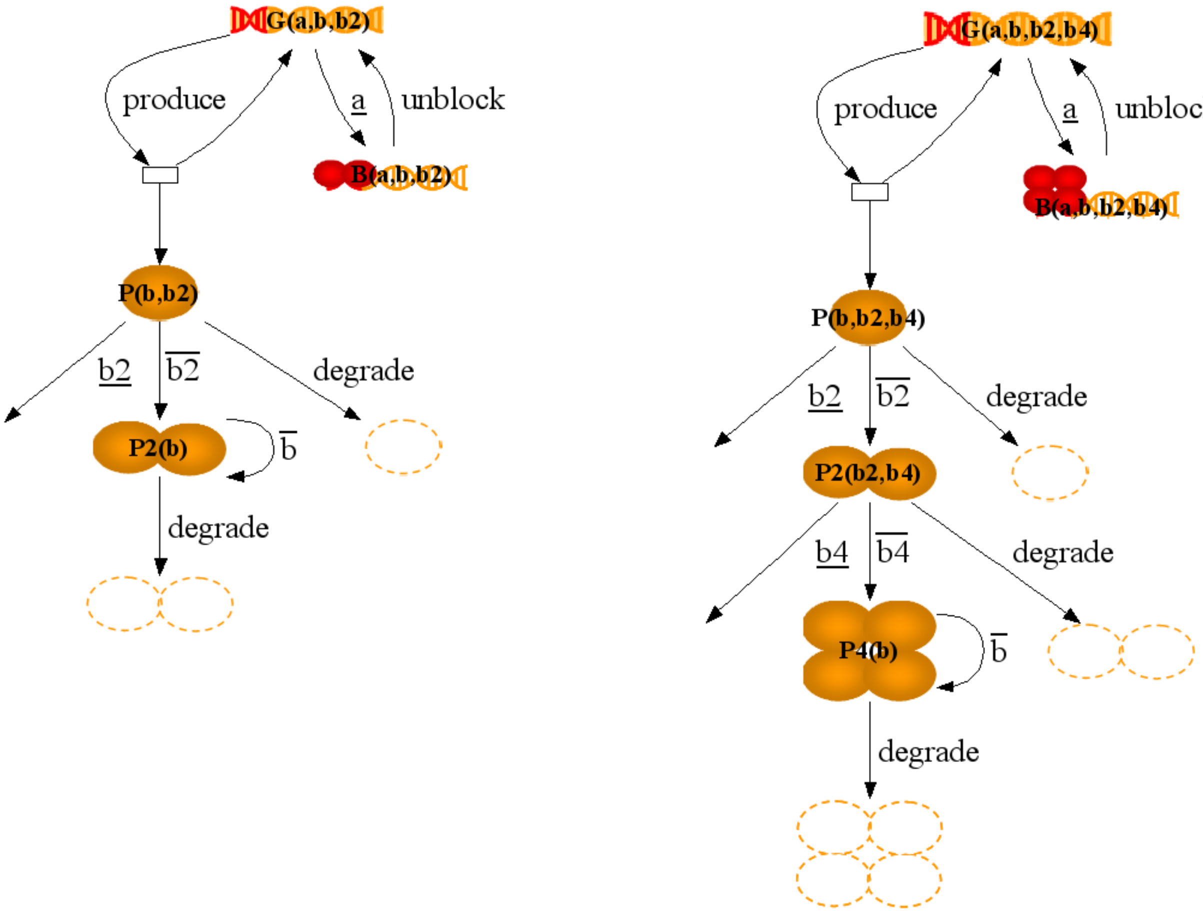




Blossey, Cardelli, Phillips, 2007

Cooperativity improves robustness

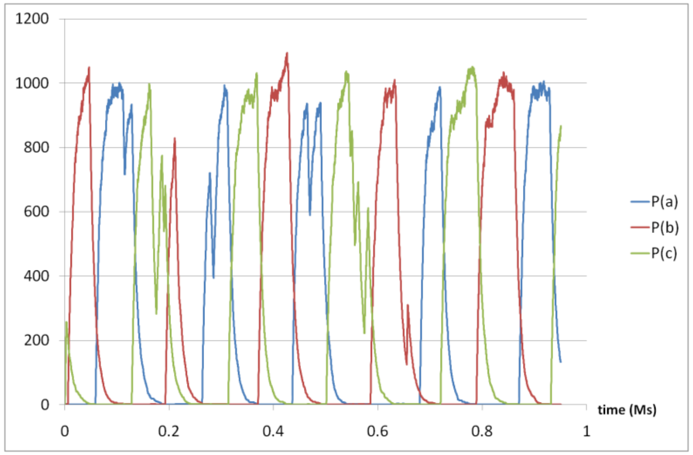
- Proteins form complexes before repressing



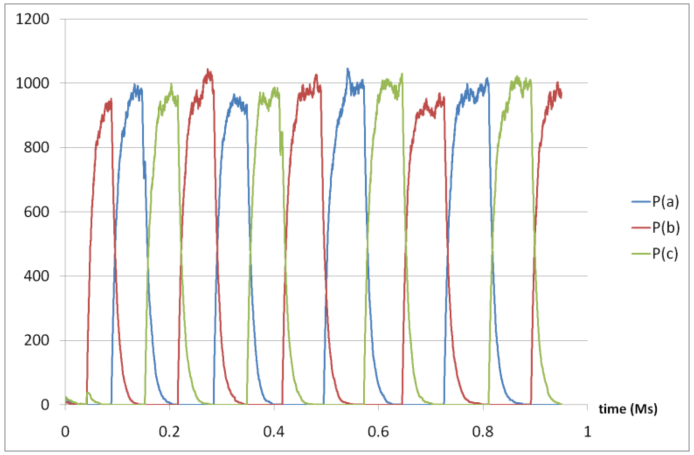
Blossey, Cardelli, Phillips, 2007

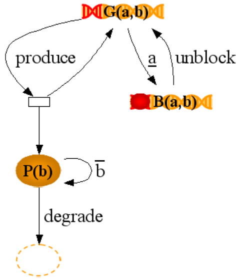
Cooperativity Simulations

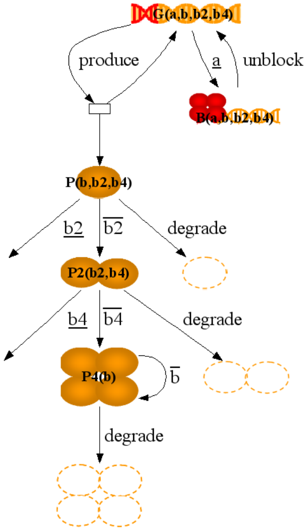
- Monomers



- Tetramers

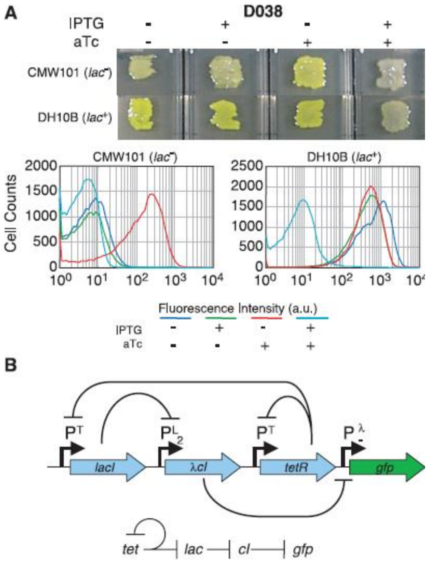






Blossey, Cardelli, Phillips, 2007

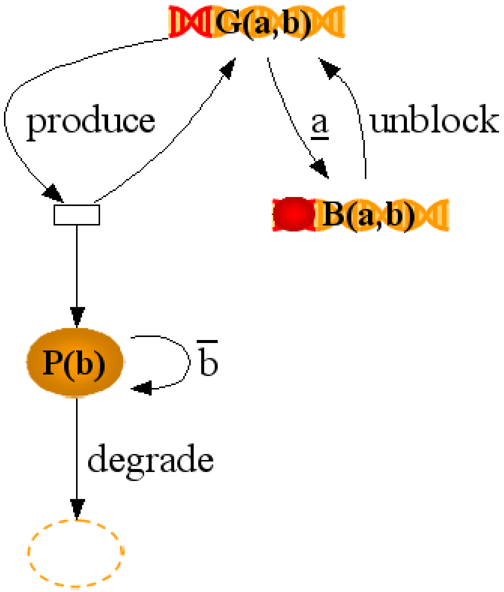
Bacteria Logic Gates [Guet et al., 2002]

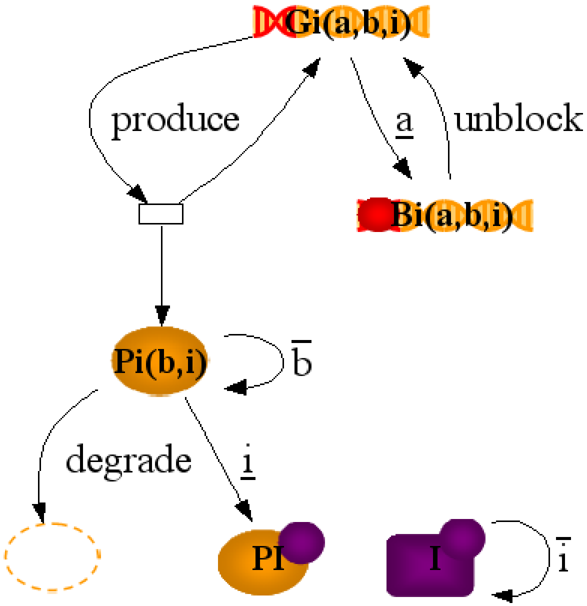


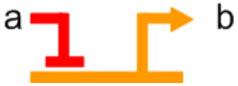
- 3 genes: tetR, lacI, λcI
- 5 promoters: PL1, PL2, PT, Pλ-, Pλ+
- 125 possible networks consisting of 3 promoter-gene units
- 2 inputs: IPTG (represses Lac), aTc (represses Tet)
- 1 output: GFP (linked to Pλ-)

Blossey, Cardelli, Phillips, 2006

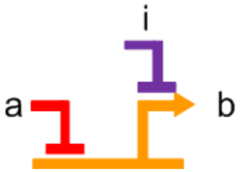
Gene with protein inhibitor







G(a,b)



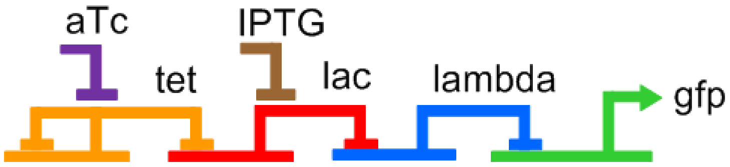
Gi(a,b,i)

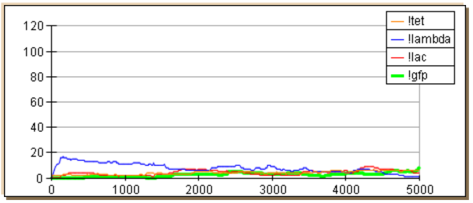
Blossey, Cardelli, Phillips, 2006

Bacteria Logic Gates

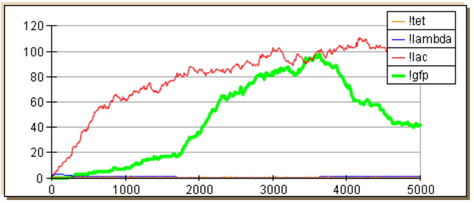
- Model 125 networks using just 2 modules:
- Enables modular simulation and analysis
- Can easily refine the modules without rewiring the networks.

D038 = Gi(tet,tet,aTc)
| Gi(tet,lac,IPTG)
| G(lac,lambda)
| G(lambda,gfp)



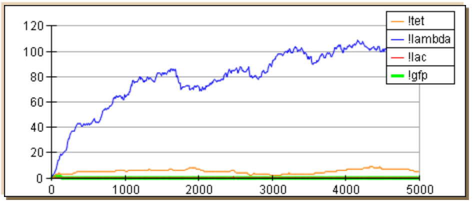


D038

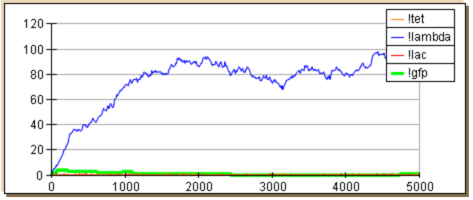


D038 | Inh(aTc)

Experiment:	
aTc	0101
IPTG	0011
GFP	0100



D038 | Inh(IPTG)



D038 | Inh(aTc) | Inh(IPTG)

Blossey, Cardelli , Phillips, 2006

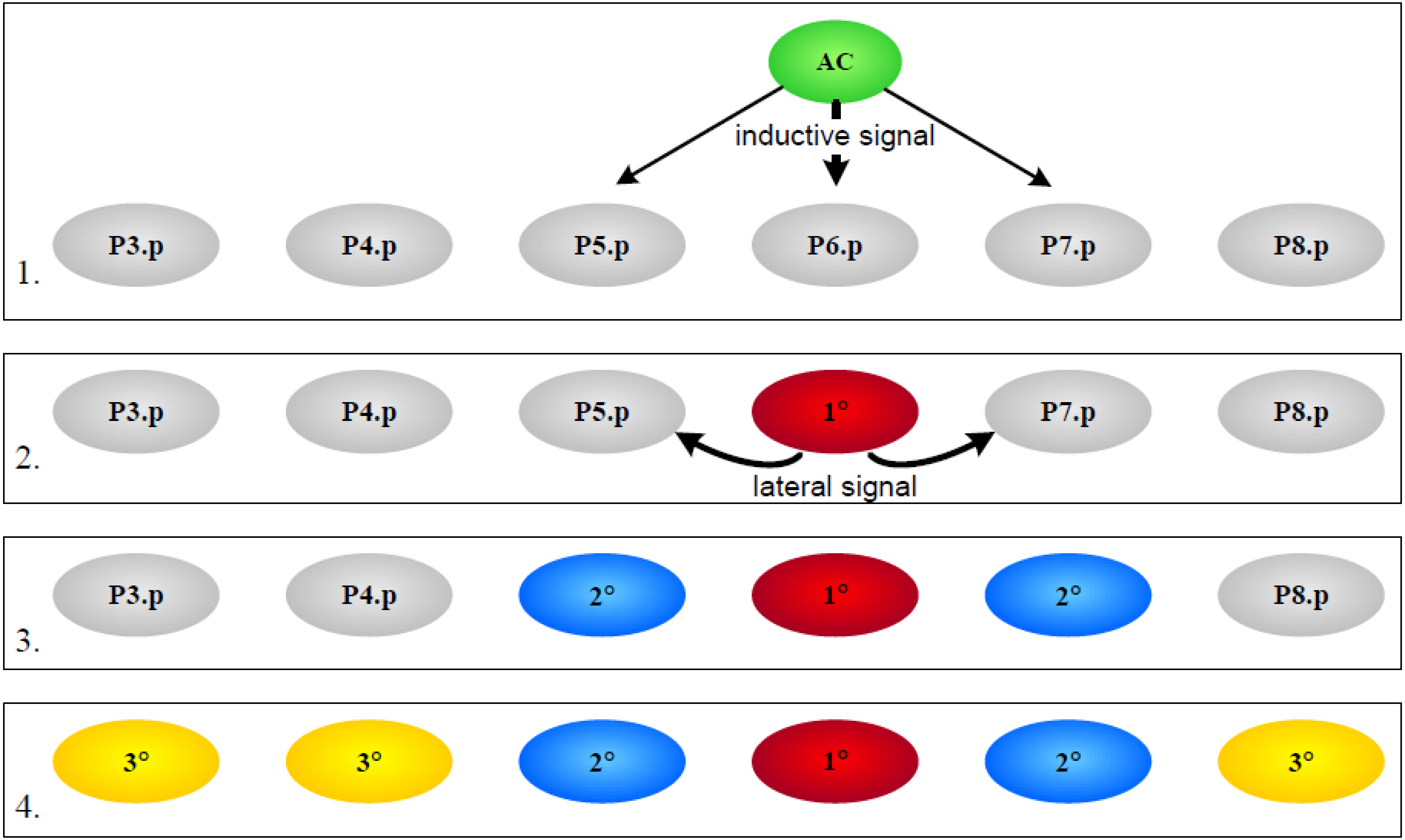
A Computational Model of C. elegans Vulval Development

with
Rosie Bloxsom (Cambridge University)
Tim Labeeuw (Cambridge University)
Jasmin Fisher (MSR Cambridge)
Hillel Kugler (MSR Cambridge)

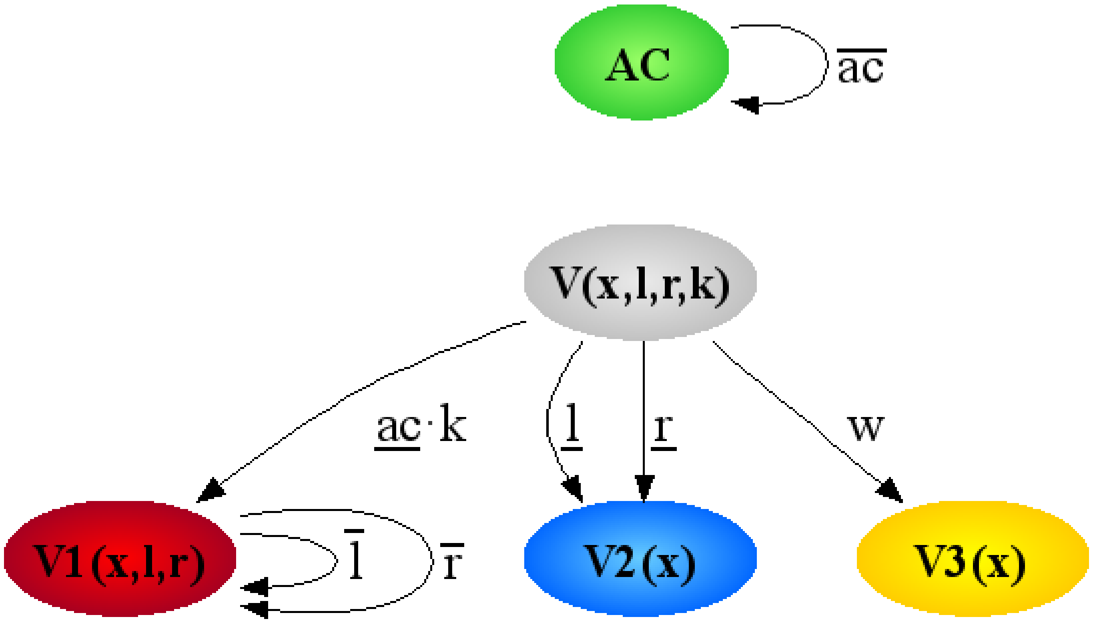
C. elegans Nematode Worm

- Model organism for development
- 1mm long, about 1000 cells
- Completely transparent, can observe growth

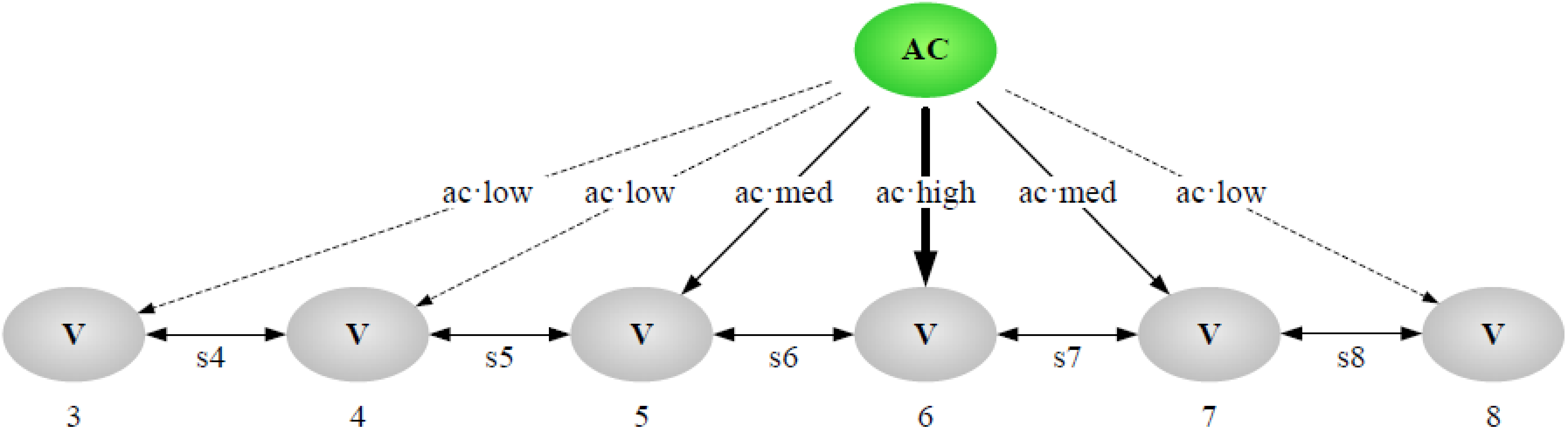
C. elegans VPC Differentiation



Simplified Model of AC and VPC



Network of AC and 6 VPC

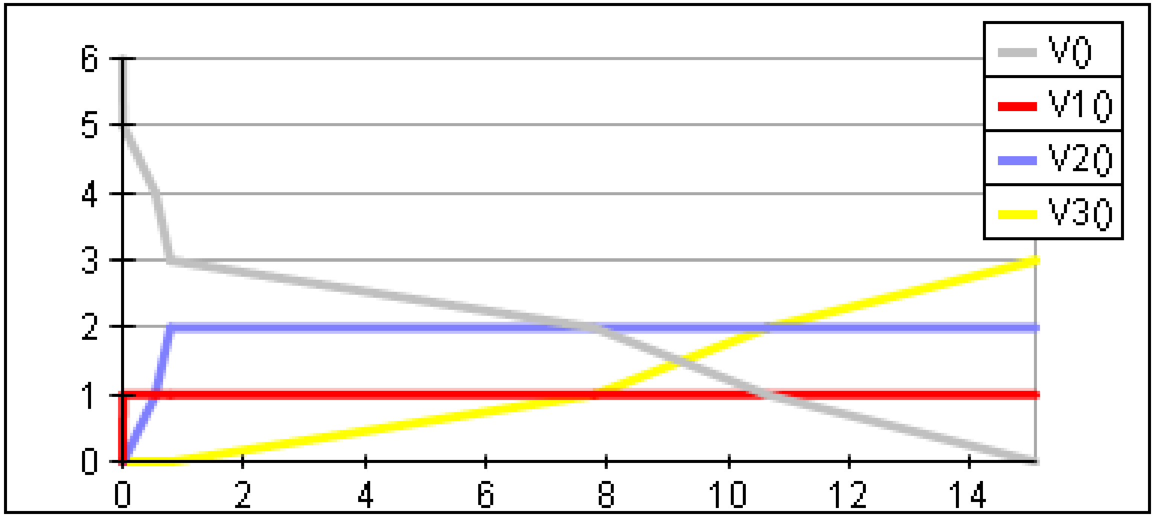


AC |

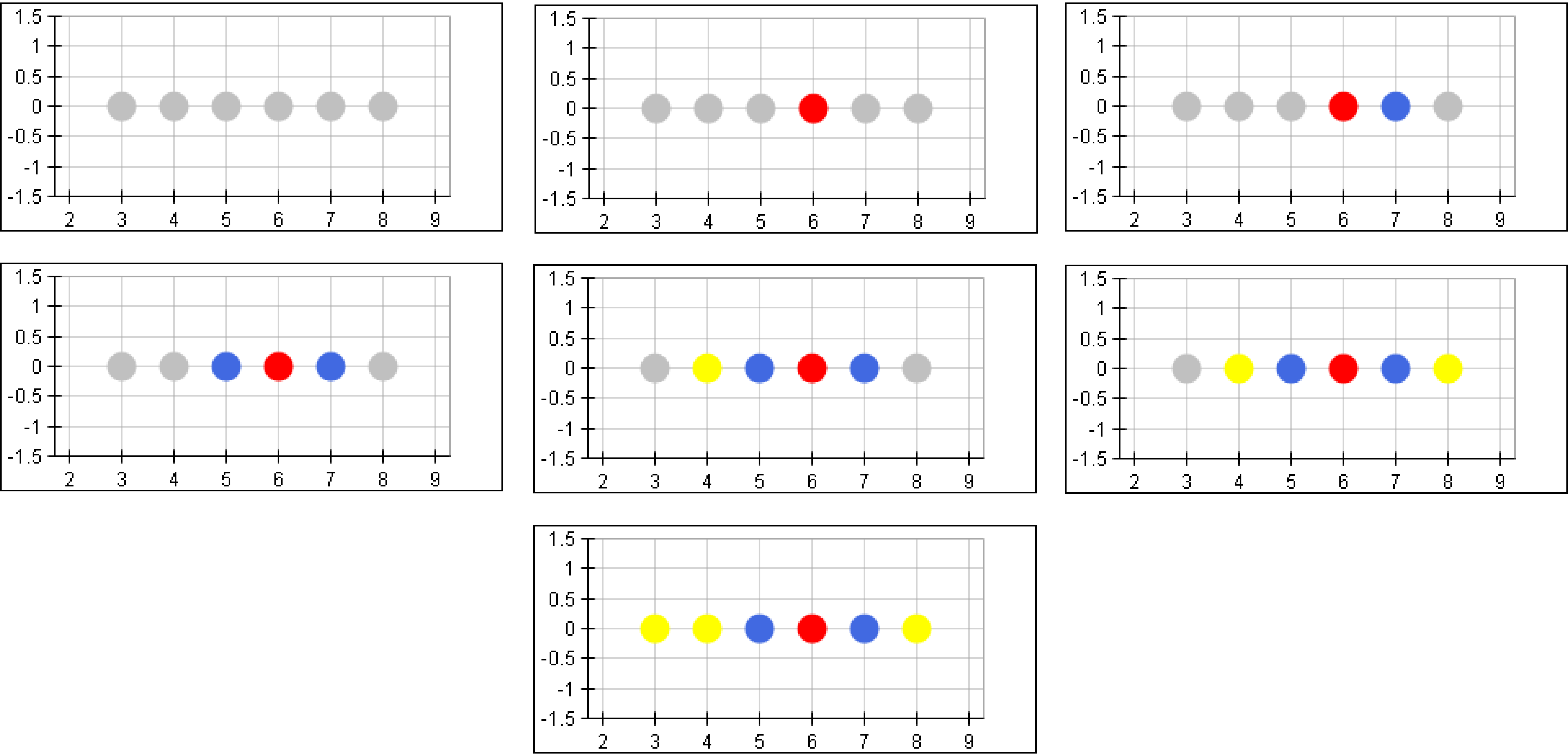
V(3,s3,s4,low) | V(4,s4,s5,low) | V(5,s5,s6,med) |

V(6,s6,s7,high) | V(7,s7,s8,med) | V(8,s8,s9,low)

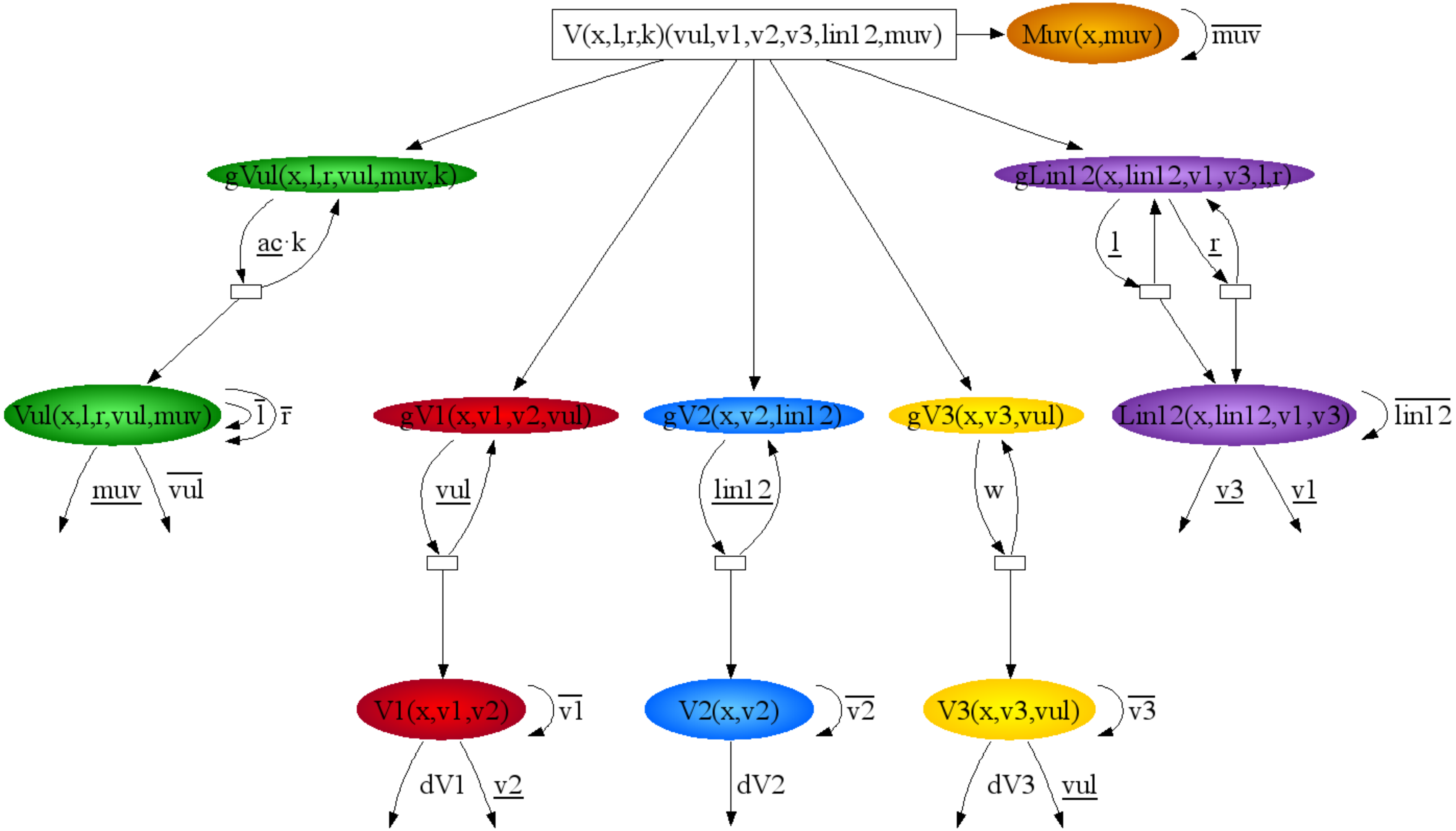
Population Plot



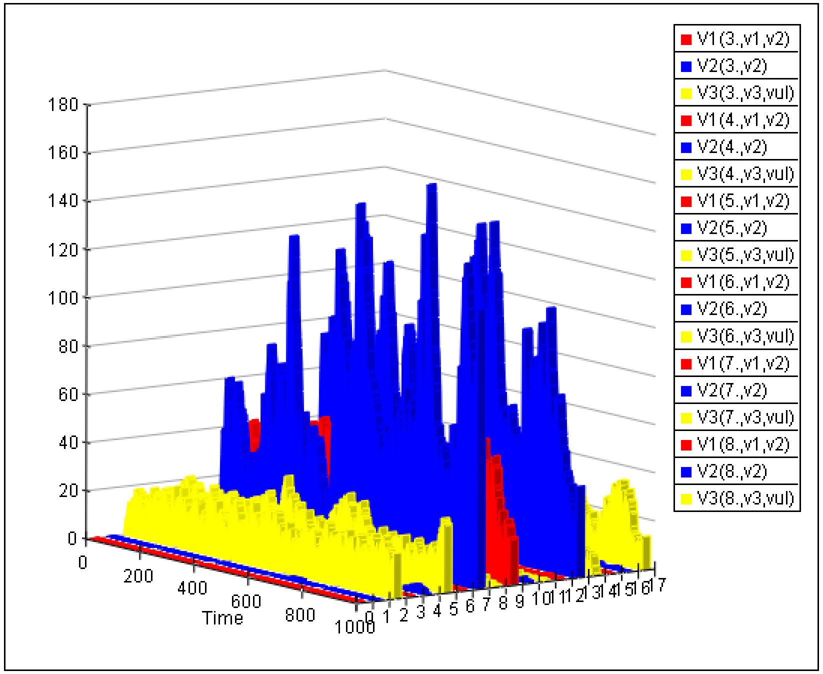
Geometric Plot

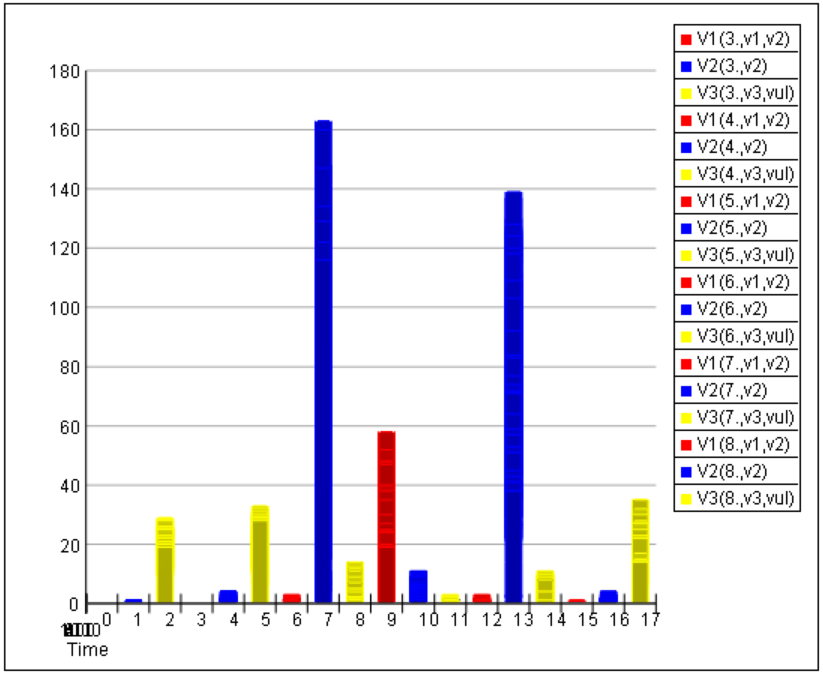


Refined VPC Model



Simulation Results

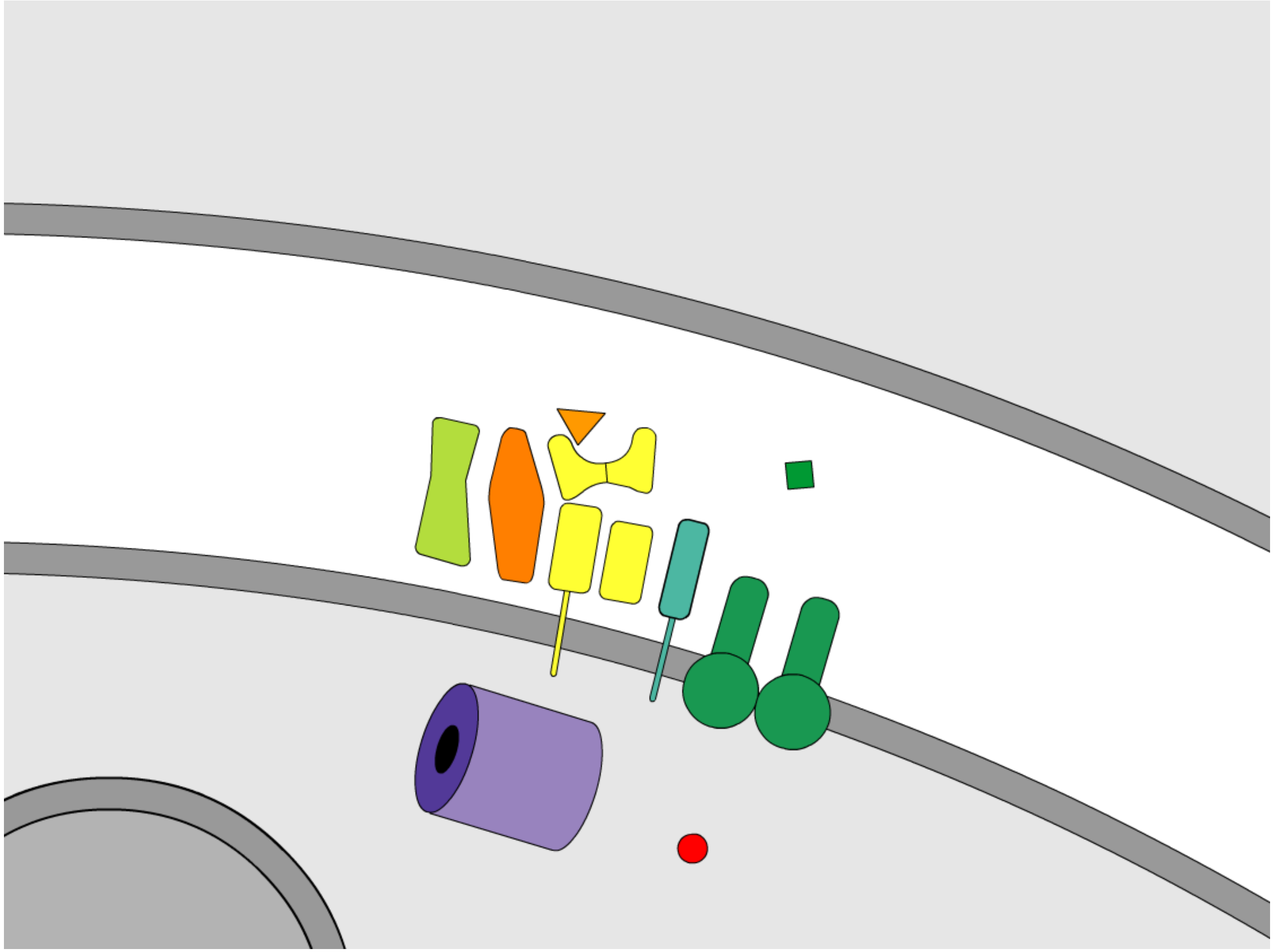




A Computational Model of MHC class I Antigen Presentation

with
Luca Cardelli (MSR Cambridge)
Leonard Goldstein (Cambridge University)
Tim Elliott (Southampton University)
Joern Werner (Southampton University)

MHC: A Biological Virus Scanner



©2005 from Immunobiology, Sixth Edition by Janeway et al.

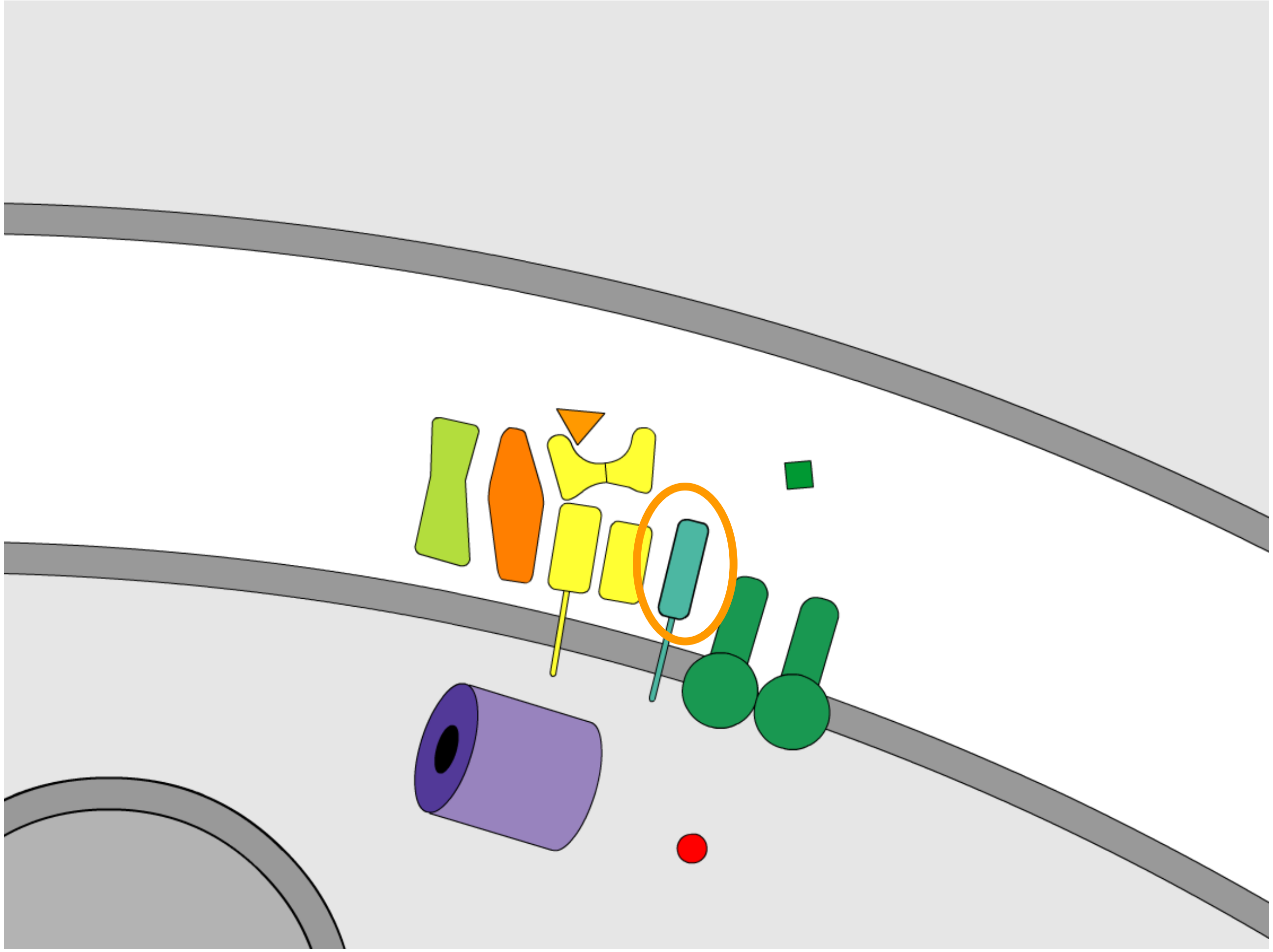
Reproduced by permission of Garland Science/Taylor & Francis LLC.

MHC: A Biological Virus Scanner

©2005 from Immunobiology, Sixth Edition by Janeway et al.

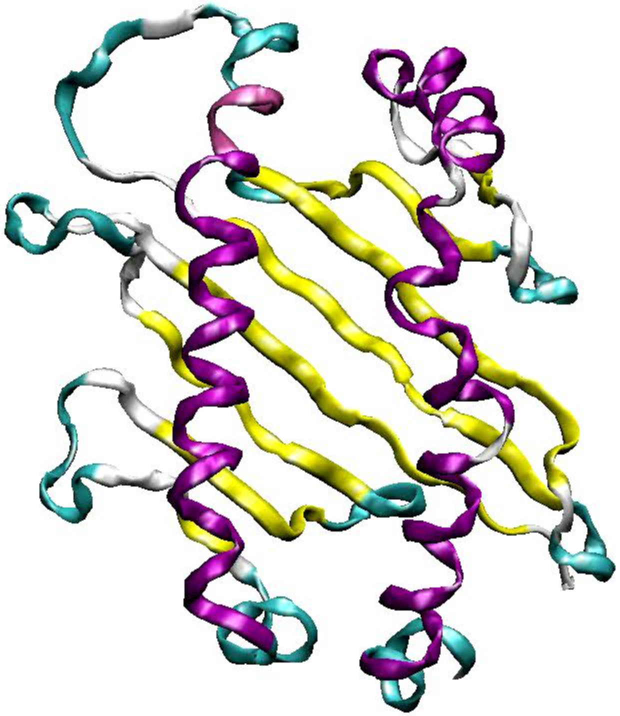
Reproduced by permission of Garland Science/Taylor & Francis LLC.

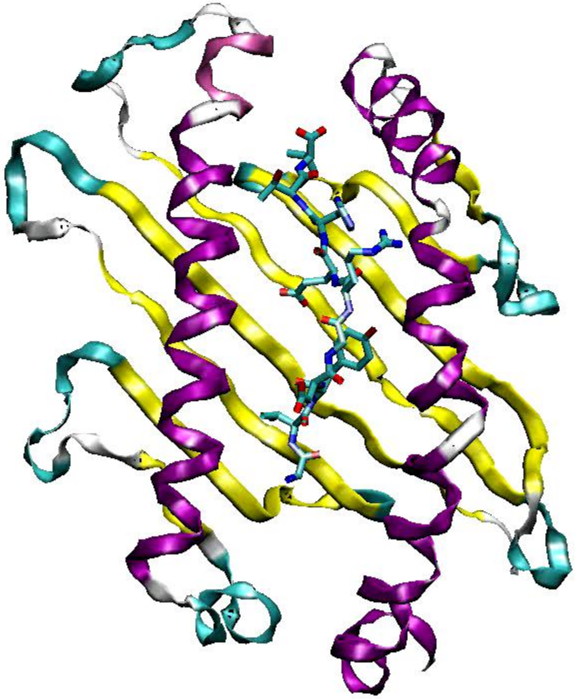
Investigate the Role of Tapasin



MHC I Structure

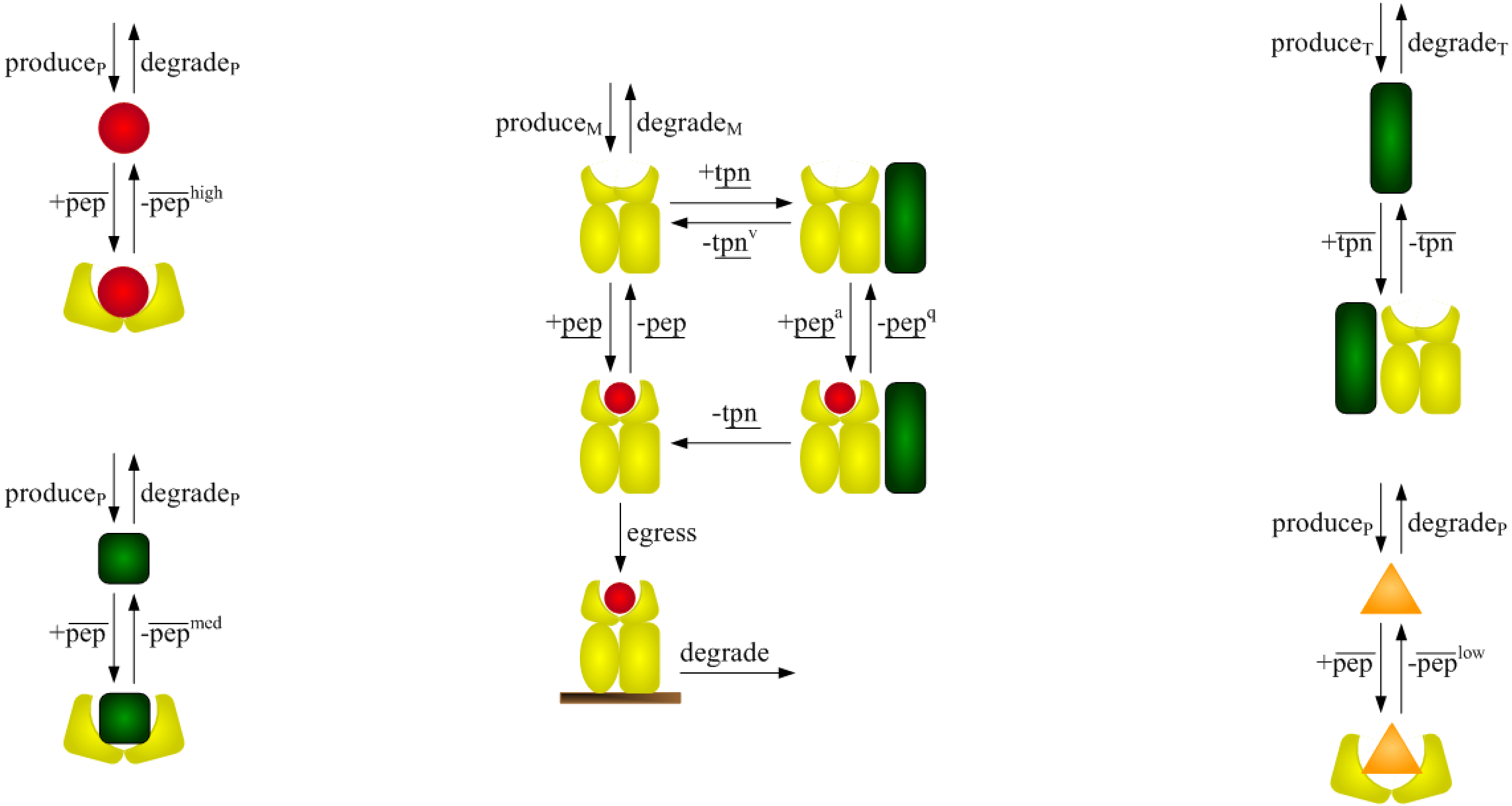
- Interaction of MHC I with peptide





SPiM Peptide Editing Model

- Graphs describe the behaviour components
- Assume low, medium and high affinity peptides



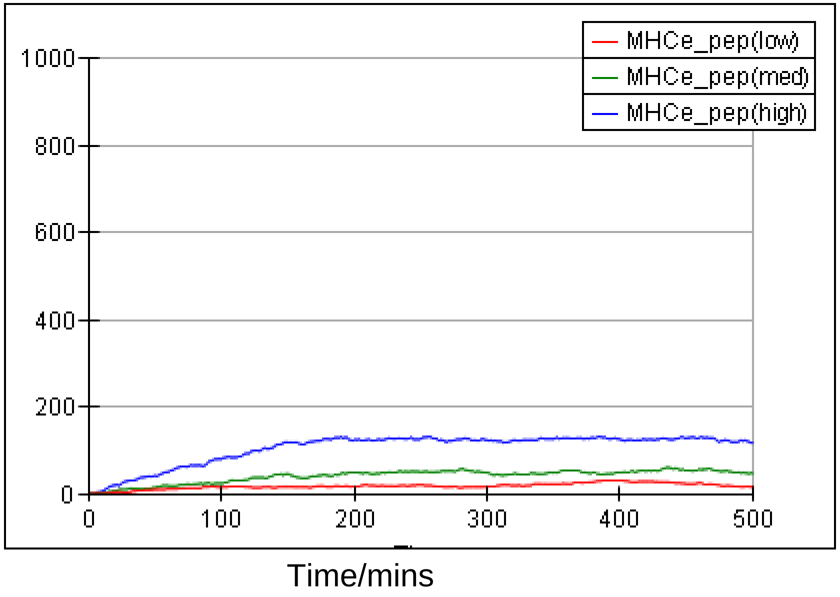
Model Parameters

- MHC spends < 2h on average in the ER.

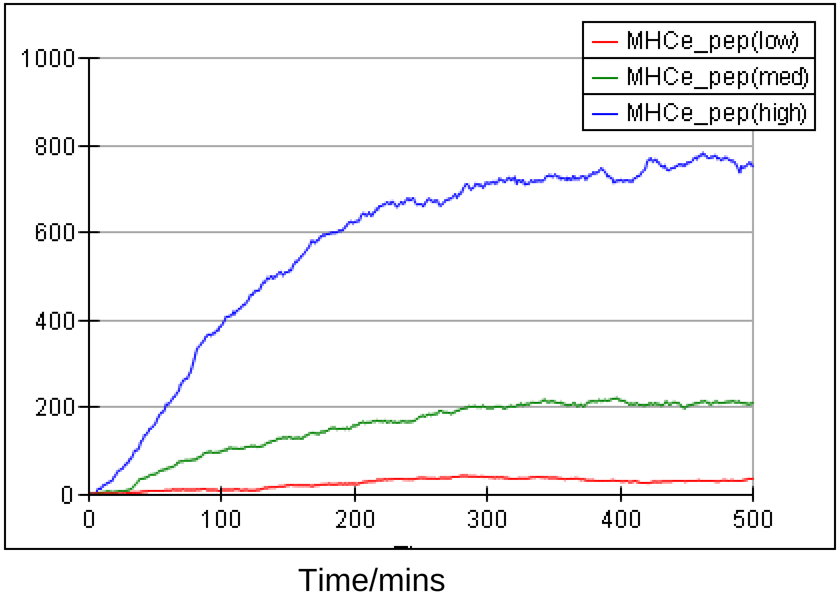
Name	**Rate min^{-1}**	**Time min**	**Range min^{-1}**	**Description**
gPep	50	0.02		Active transport of peptides into the ER
dPep	10	0.1		Degradation of free peptides inside the ER
bind	1	1		Binding of peptides to MHC (per molecule)
low	3	0.33		Unbinding of low affinity peptides from MHC
med	1.2	0.83		Unbinding of medium affinity peptides from MHC
high	0.5	2		Unbinding of high affinity peptides from MHC
gMHC	10	0.1		Assembly of MHC complexes inside the ER
dMHCo	0.01	100	0.01 - 100	Degradation of free MHC inside the ER
dMHCe	0.01	100		Degradation of loaded MHC at the cell surface
egress	1	1	0.01 - 1	Egression of loaded MHC from the ER
gTPN	10	0.1		Production of tapasin inside the ER
dTPN	0.01	100		Degradation of free tapasin inside the ER
bindT	100	0.01	1 - 1000	Binding of tapasin to MHC (per molecule)
uT	1	1	0.01 - 1	Unbinding of tapasin from loaded MHC

Simulations Match Experiments

- MHC needs to present stable peptides
- Improved selection with tapasin. How?



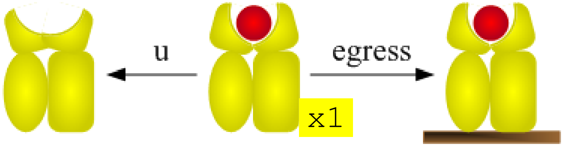
No Tapasin ✘



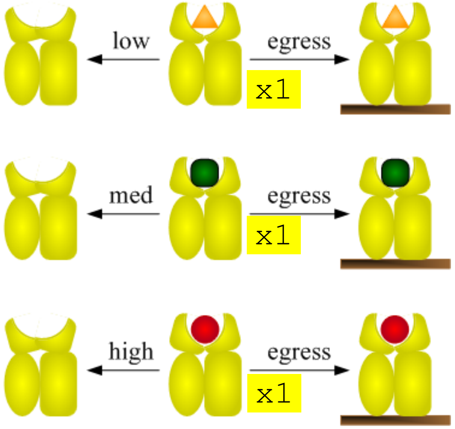
Tapasin ✓

Peptide Discrimination

- Consider a loaded peptide with unbinding rate *u*
- Competition between unbinding and egression
- Egression probability determined by off-rate
- Maximal discrimination as egress tends to 0



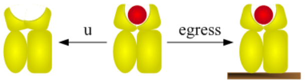
$$P(u) = \frac{egress}{u + egress} \rightarrow \frac{egress}{u}$$

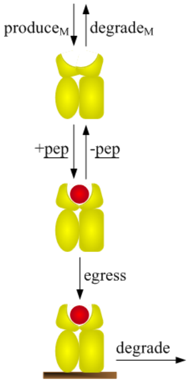


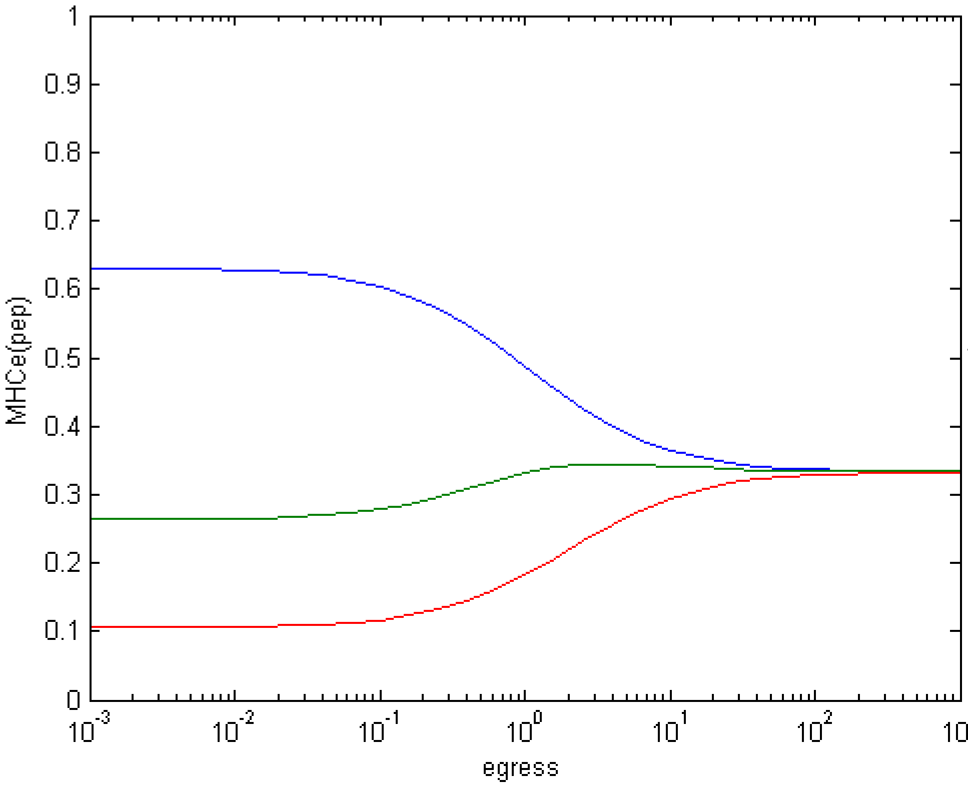
$$P(u_i|E) = \frac{P(u_i)}{\sum_k P(u_k)} \rightarrow \frac{1/u_i}{\sum_k 1/u_k}$$

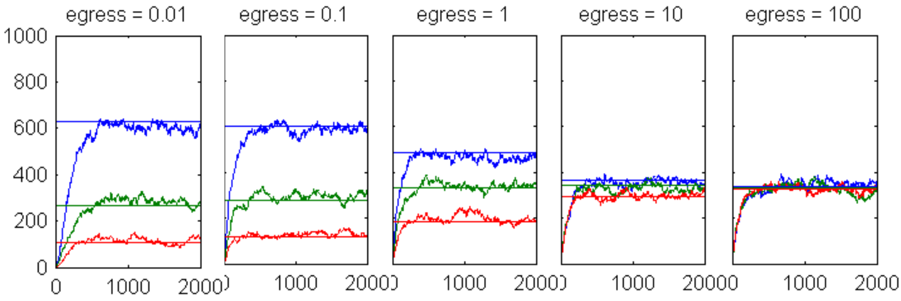
Parameter Space

- Maximal discrimination determined by off-rate
- High peptide turnover is a key factor



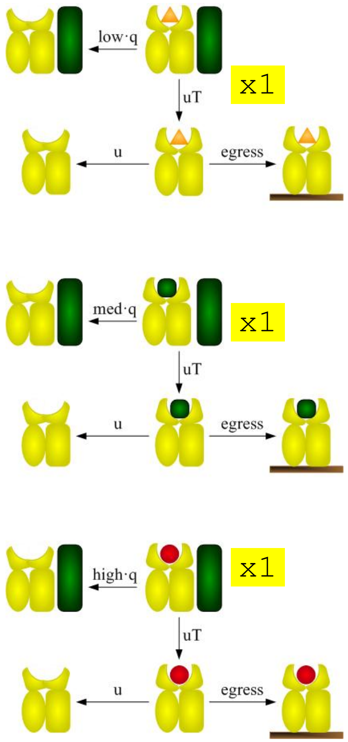


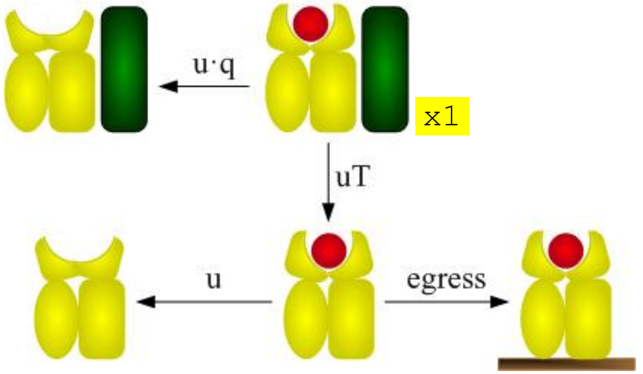




Peptide Discrimination

- Tapasin adds a second filtering stage
- Egression probability determined by off-rate^2



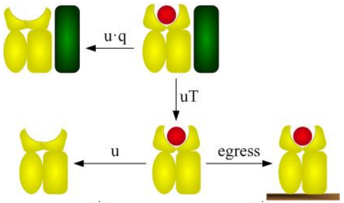


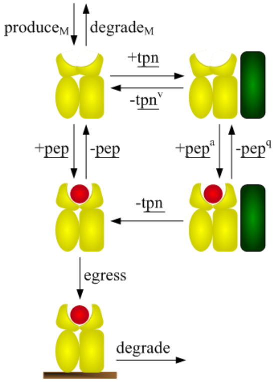
$$P_2(u) = \frac{uT}{q\cdot u + uT}\cdot P(u) \rightarrow \frac{uT\cdot egress}{q\cdot u^2}$$

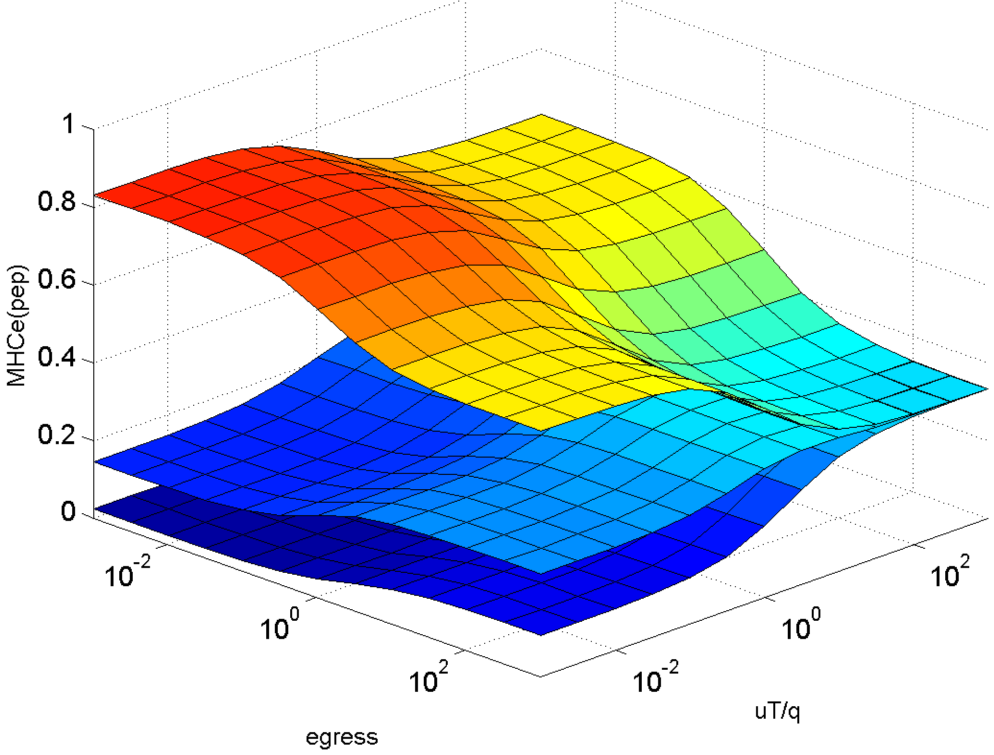
$$P_2(u_i|E) = \frac{P_2(u_i)}{\sum_k P_2(u_k)} \rightarrow \frac{1/u_i^2}{\sum_k 1/u_k^2}$$

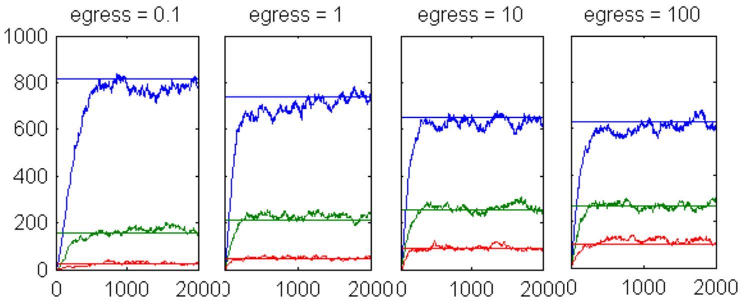
Parameter Space

- Tapasin improves upper bound on discrimination
- Peptide editing is a 2-stage filter process



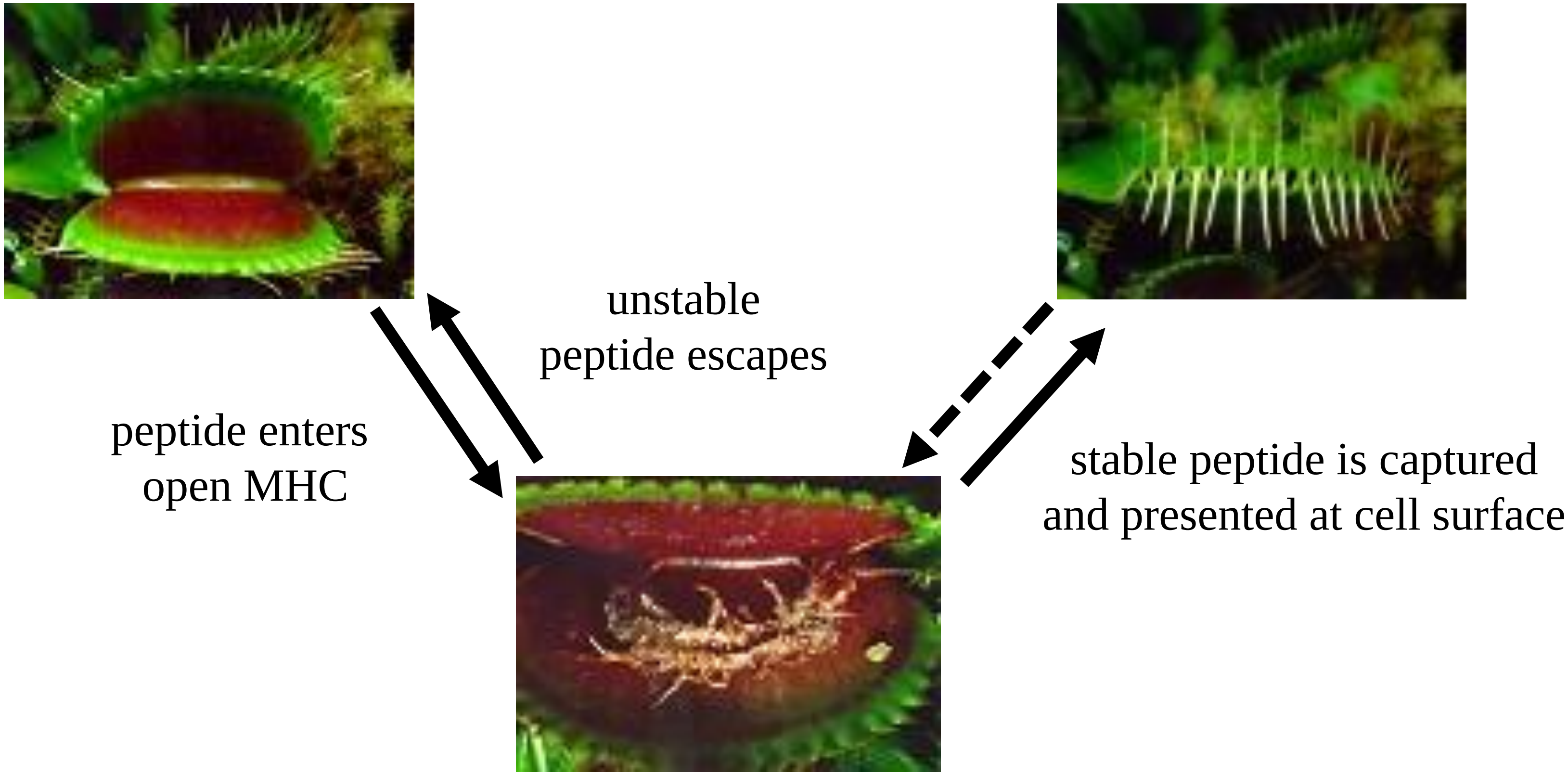






Peptide Loading: Flytrap Model

- MHC I captures peptides like a Venus Flytrap.



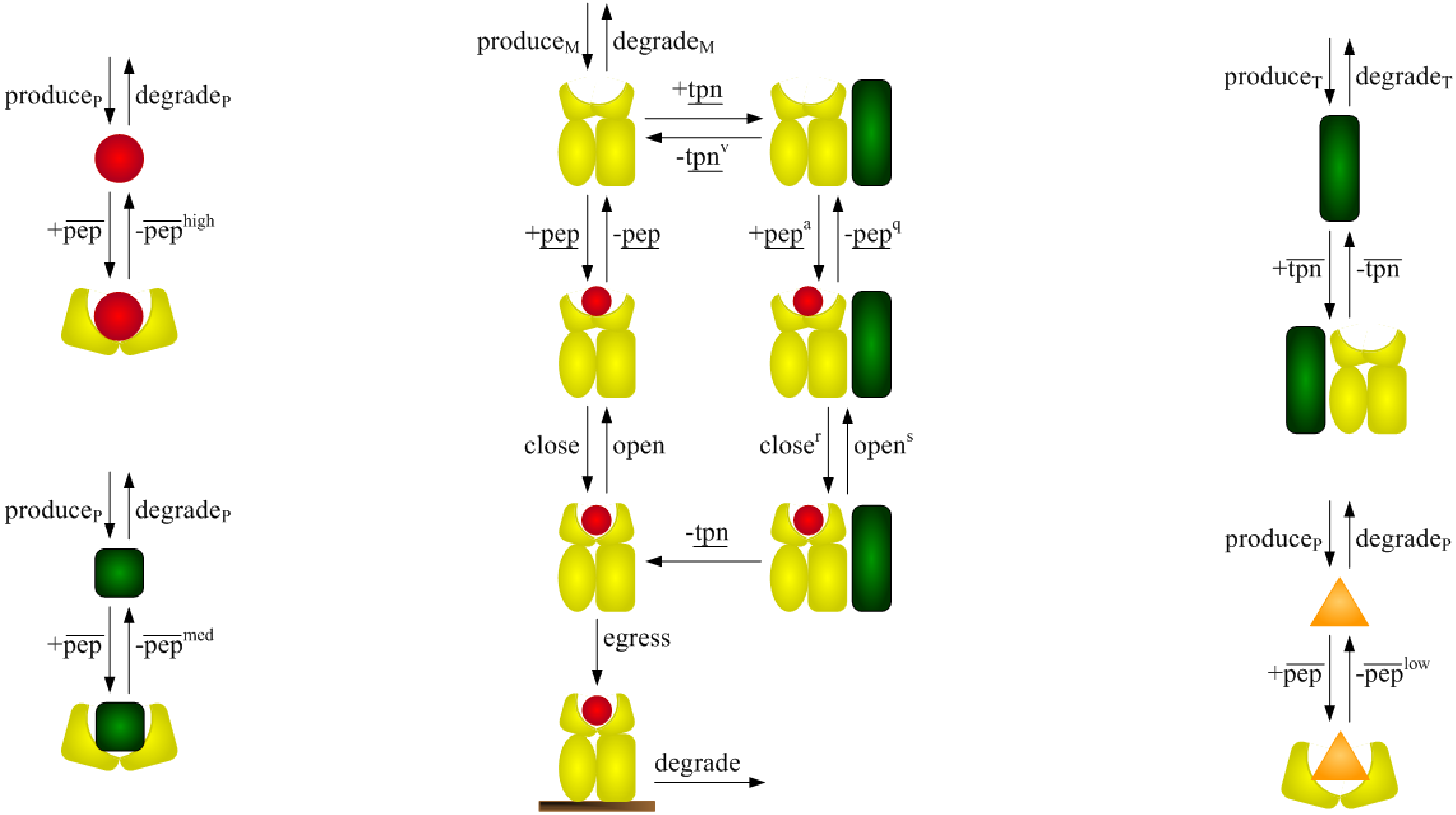
peptide enters
open MHC

unstable
peptide escapes

stable peptide is captured
and presented at cell surface

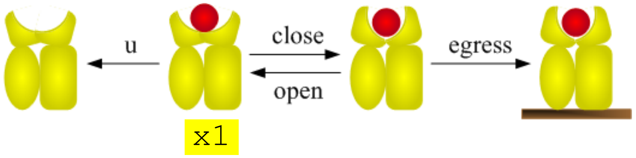
Flytrap Peptide Editing Model

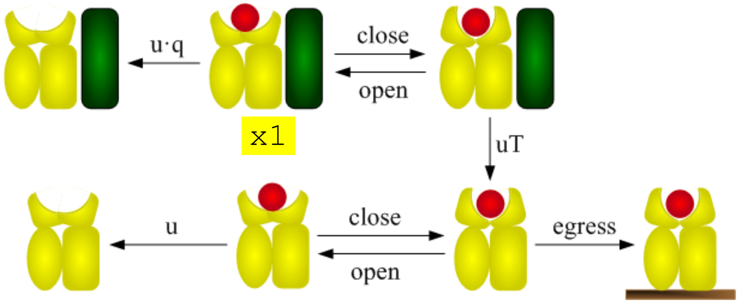
- Extend the model with conformational change of MHC



Peptide Discrimination: Flytrap

- MHC can open and close several times
- But same upper bound on discrimination





$$P(u) = \frac{c}{u+c}\cdot\frac{e}{o+e}\cdot\sum_{n=0}^{\infty}\left(\frac{c}{u+c}\cdot\frac{o}{o+e}\right)^{n}$$

$$= \frac{c}{u+c}\cdot\frac{e}{o+e}\cdot\frac{1}{1-co/(u+c)\cdot(o+e)}$$

$$= \frac{c\cdot e}{u\cdot e+u\cdot o+c\cdot e} = \frac{c\cdot e/(e+o)}{u+c\cdot e/(e+o)}$$

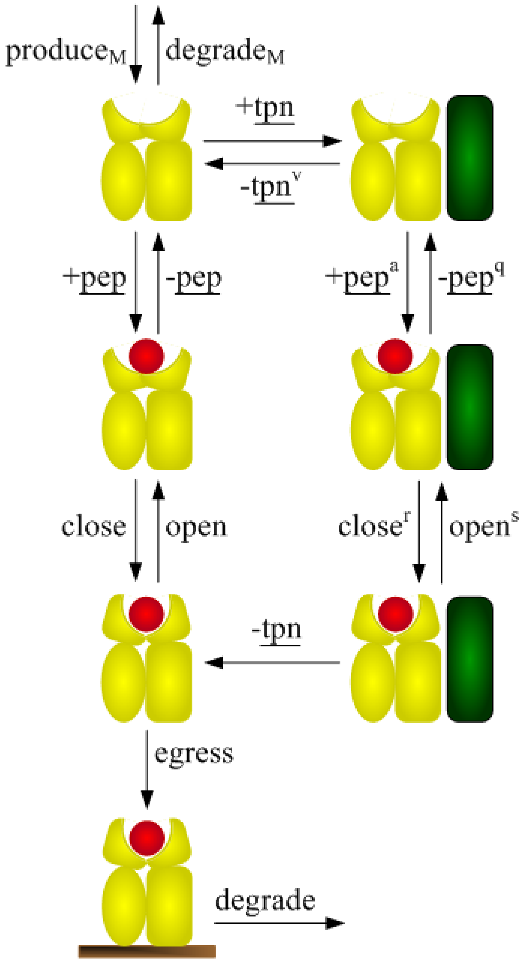
$$\rightarrow \frac{c\cdot e/(e+o)}{u}$$

$$P_2(u) = \frac{c\cdot r\cdot uT/(uT+o\cdot s)}{q\cdot u+c\cdot r\cdot uT/(uT+o\cdot s)}\cdot\frac{o\cdot P(u)}{o+e}$$

$$\rightarrow \frac{c\cdot r\cdot uT\cdot o\cdot c\cdot e/(uT+o\cdot s)(e+o)}{q\cdot(o+e)\cdot u^2}$$

Key mechanisms identified

- MHC can delay egression to sample a wider range of peptides
- High peptide turnover is needed to maintain peptide distribution inside ER
- Tapasin holds open MHC and increases peptide off-rate to quickly select high affinity peptides
- Tapasin stabilises MHC to prevent degradation and increase presentation
- Tapasin increases peptide on-rate by anchoring MHC at entrance to ER?
- Tapasin shifts equilibrium to open conformation as a way of delaying egression?



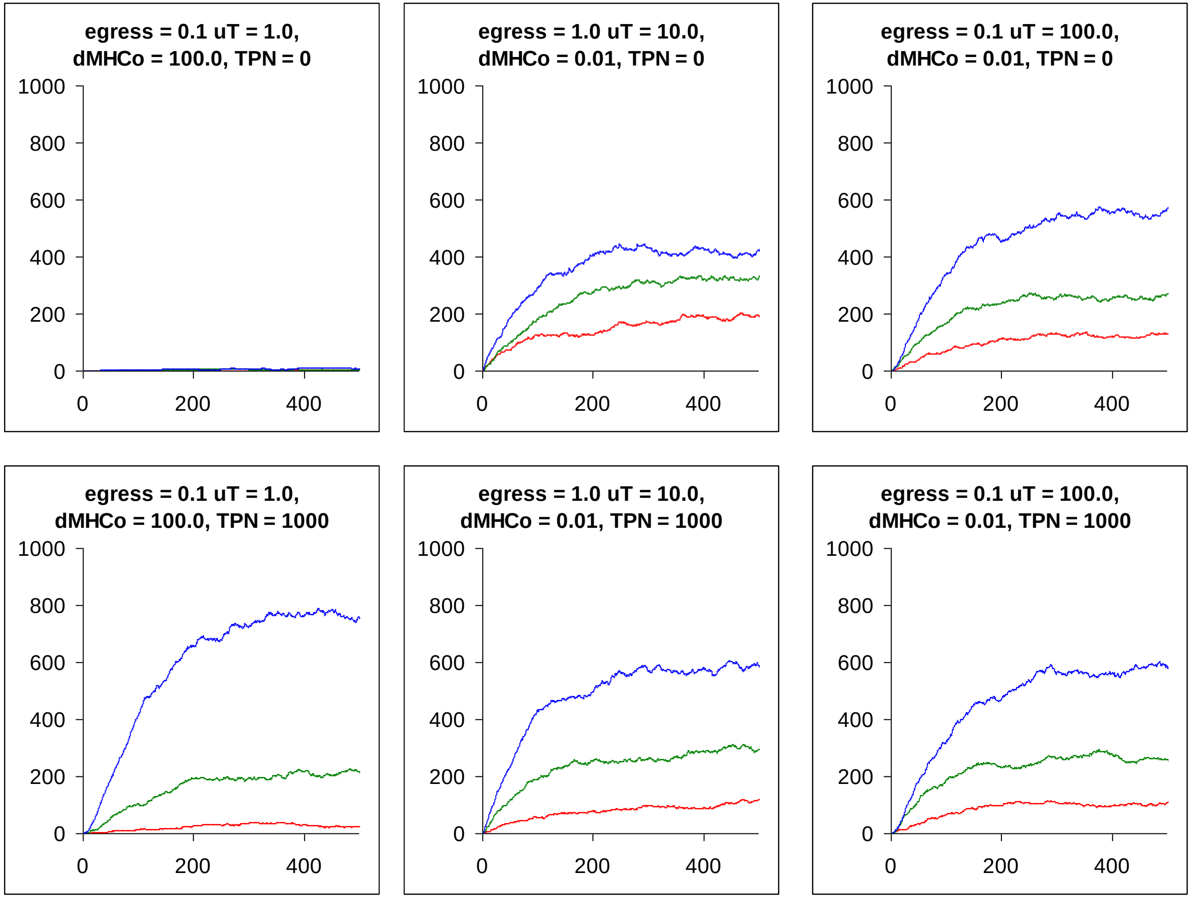
MHC Alleles: Model Predictions

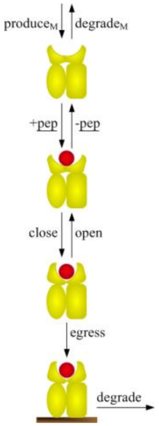
- Explanation for immune system variability

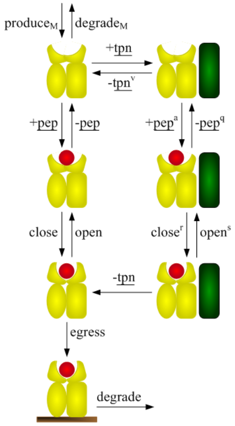
B4402 (Dependent) B2705 (Partially) B4405 (Independent)

No TPN

1000 TPN

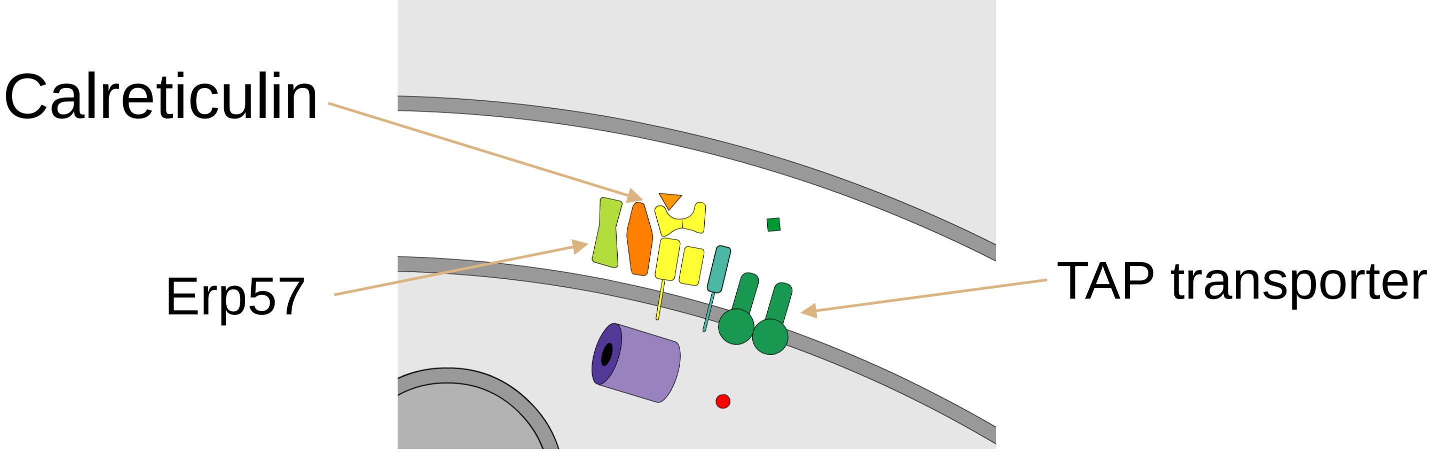






Extending the Model

- Include function of additional chaperones.



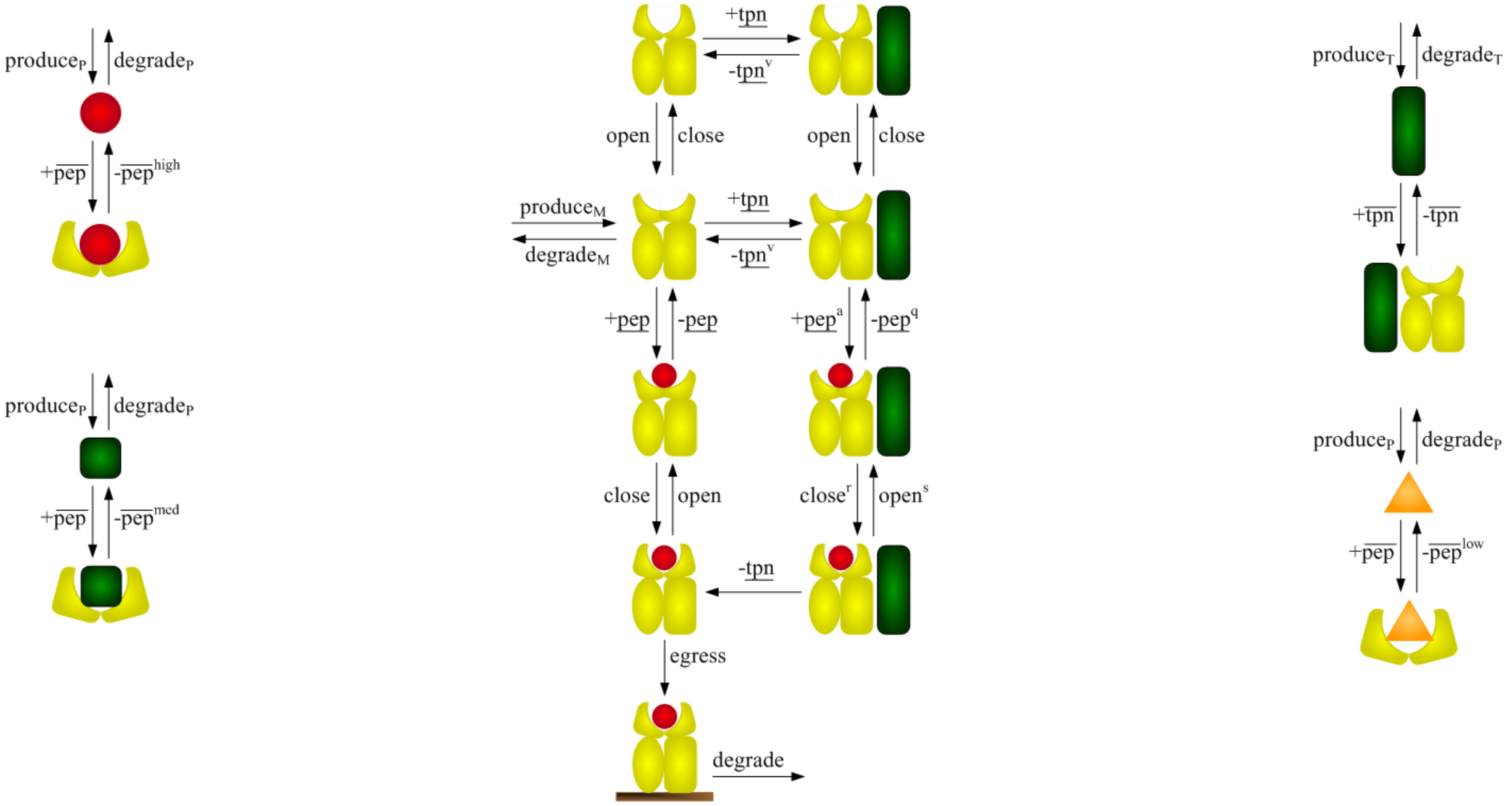
Calreticulin

Erp57

TAP transporter

Extendable,Maintainable Models

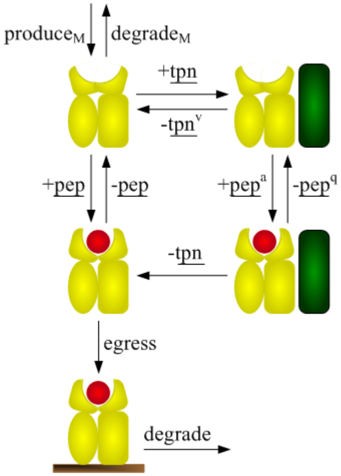
Build complex models by composing simpler components.
The models are easier to extend and maintain.

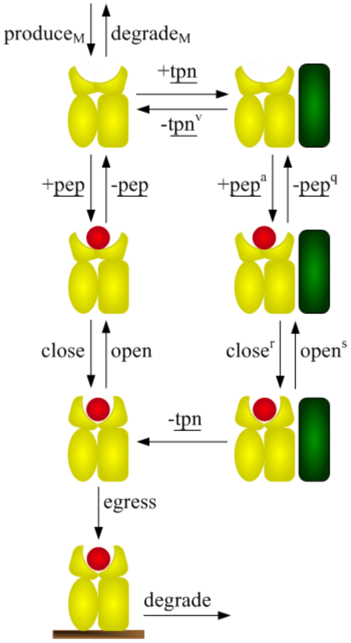


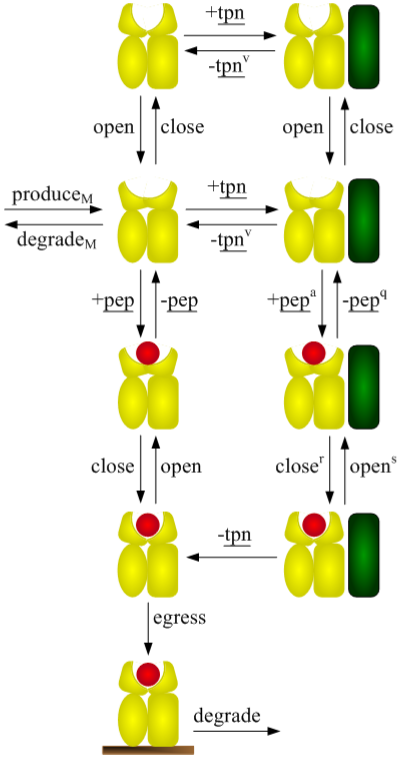
Verifying Biological Models

Mhc.txt

Can we replace one model with another?



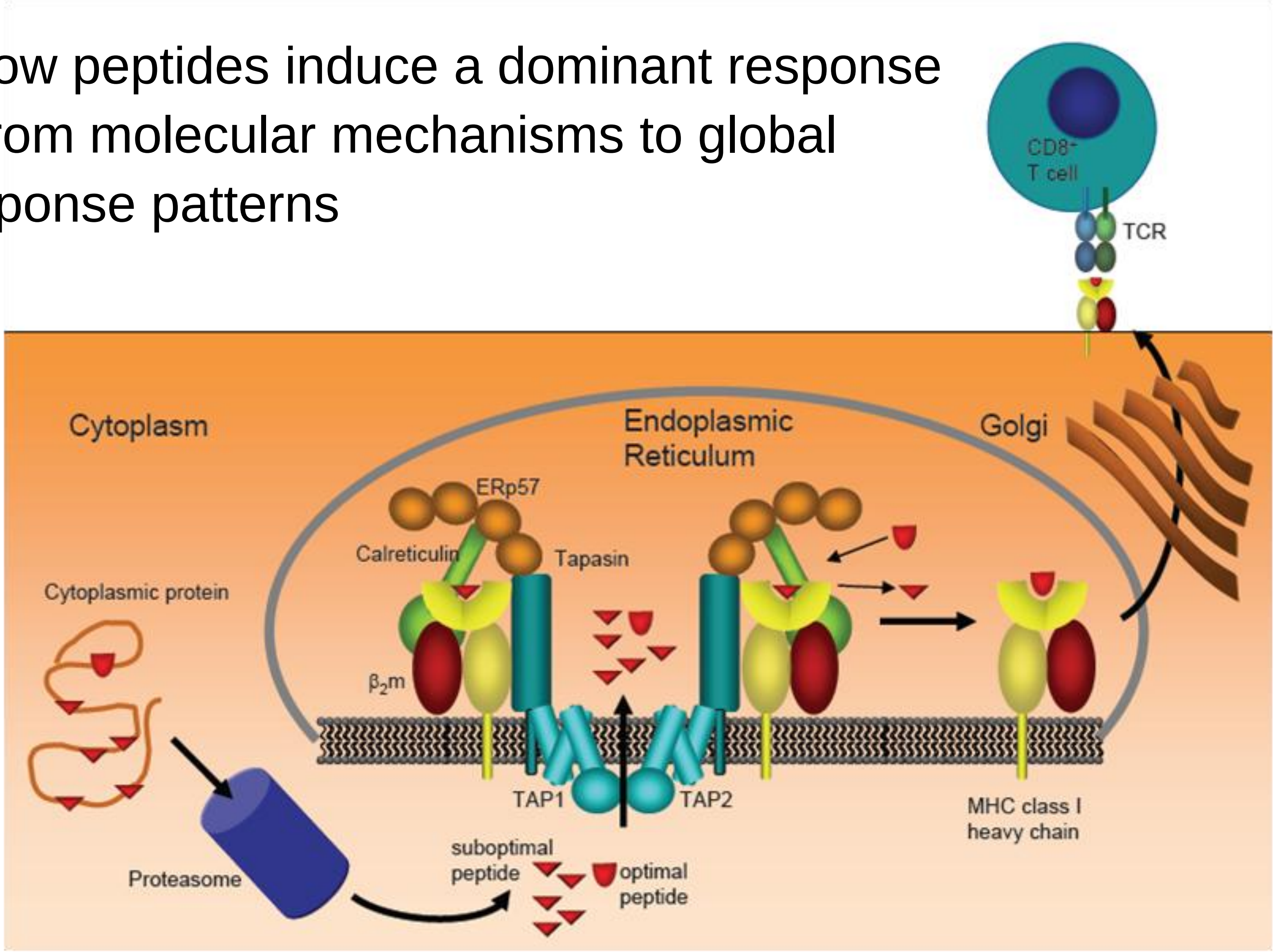




MHC(bind,bindT)

Modelling Immunodominance

- How peptides induce a dominant response
- From molecular mechanisms to global response patterns



References

- [Blossey et al., 2007] Blossey, R., Cardelli, L., and Phillips, A. (2007). Compositionality, Stochasticity and Cooperativity in Dynamic Models of Gene Regulation. HFSP Journal.
- [Blossey et al., 2006] Blossey, R., Cardelli, L., and Phillips, A. (2006). A compositional approach to the stochastic dynamics of gene networks. Transactions in Computational Systems Biology, 3939:99–122.
- [Borgstroem et al., 2007] Borgstroem, J. Gordon, A., and Phillips, A. (2007). A Chart Semantics for the Pi-calculus. In Proceedings of Expressiveness in Concurrency.
- [Gillespie, 1977] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem., 81(25):2340–2361.
- [Guet et al., 2002] Guet, C.C., Elowitz, M.B., Hsing, W. & Leibler, S. (2002) Combinatorial synthesis of genetic networks. Science 296 1466-1470.
- [Huang and Ferrel, 1996] Huang, C.-Y. F. and Ferrel, J. E. (1996). Ultrasensitivity of the mitogen-activated protein kinase cascade. PNAS, 93:10078–10083.
- [Lecca and Priami, 2003] Lecca, P. and Priami, C. (2003). Cell cycle control in eukaryotes: a biospi model. In proceedings of Concurrent Models in Molecular Biology.
- [Phillips and Cardelli, 2007] Phillips, A. and Cardelli, L. (2007). Efficient, Correct Simulation of Biological Processes in the Stochastic Pi-calculus. In Proceedings of Computational Methods in Systems Biology, 4695:184–199.
- [Phillips et al., 2006] Phillips, A., Cardelli, L., and Castagna, G. (2006). A graphical representation for biological processes in the stochastic pi-calculus. Transactions in Computational Systems Biology, 4230:123–152.

References

- [Priami, 1995] Priami, C. (1995). Stochastic π-calculus. The Computer Journal, 38(6):578–589. Proceedings of PAPM’95.
- [Priami et al., 2001] Priami, C., Regev, A., Shapiro, E., and Silverman, W. (2001). Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Information Processing Letters, 80:25–31.
- [Regev et al., 2001] Regev, A., Silverman, W., and Shapiro, E. (2001). Representation and simulation of biochemical processes using the pi- calculus process algebra. In Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E., editors, Pacific Symposium on Biocomputing, volume 6, pages 459–470, Singapore. World Scientific Press.
- [Sangiorgi and Walker, 2001] Sangiorgi, D. and Walker, D. (2001). The pi-calculus: a Theory of Mobile Processes. Cambridge University Press.
- [Silverman et al., 1987] Silverman, W., Hirsch, M., Houri, A., and Shapiro, E. (1987). The logix system user manual, version 1.21. In Shapiro, E., editor, Concurrent Prolog: Collected Papers (Volume II), pages 46–77. MIT Press, London.
- Mark Howarth, Anthony Williams, Anne B. Tolstrup, and Tim Elliott. Tapasin enhances mhc class i peptide presentation according to peptide half-life. PNAS, 101(32):11737–11742, 2004.
- Sebastian Springer, Klaus Doring, Jonathan C. A. Skipper, Alain R. M. Townsend, and Vincenzo Cerundolo. Fast association rates suggest a conformational change in the mhc class i molecule h-2db upon peptide binding. Biochemistry, 37:3001–3012, 1998.
- A. Williams, C. Au Peh, and T. Elliott. The cell biology of mhc class i antigen presentation. Tissue Antigens, 59:3–17, 2002.

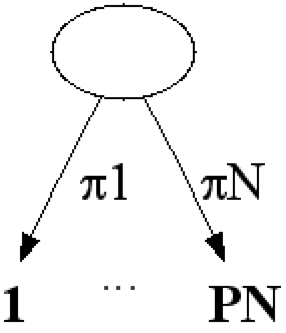
SPiM Definition

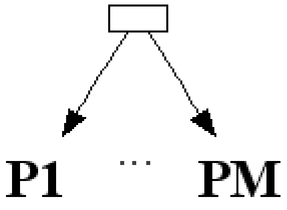
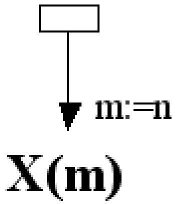
Syntax
Semantics
Graphics

SPiM Syntax

π::=	$\underline{x}(m)$	**Receive** value m on channel x
	$\bar{x}\langle n\rangle$	**Send** value n on channel x
	$\bar{x}(m)$	**Send** restricted value m on channel x
	r	**Delay** at rate r
M::=	$\pi_1.P_1 + \ldots + \pi_N.P_N$	**Choice** between actions
P::=	$P_1 \mid \ldots \mid P_M$	**Parallel** composition of processes
	$X(n)$	**Species** X with parameters n
	$(x_1,\ldots,x_\mathrm{N})\ P$	**Restriction** of channels $x_1,\ldots,x_\mathrm{N}$ to P
D::=	P	**Definition** of a process
	M	**Definition** of a choice
E::=	$X_1\,(m_1) = D_1\ ,\ldots,$	**Definitions** for X_i with parameters m_i
	$X_N\,(m_N) = D_N$	
S::=	E,P	**System** of E and P

Graphical Syntax: Environment *E*

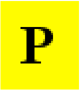
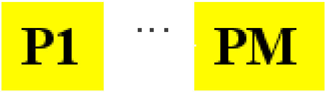
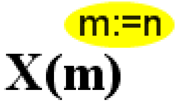
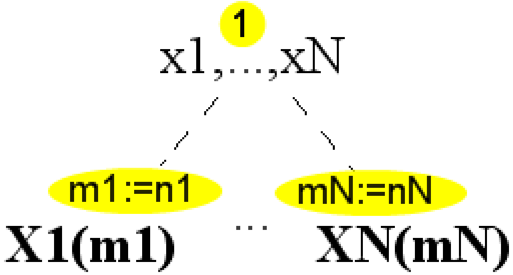
	Choice
M	$\pi_1.P_1 + \ldots + \pi_N.P_N$
M	

	Parallel	Species	Restriction
P	$P_1 \mid \ldots \mid P_M$	$X(n)$, if $X(m) = D$	$(x_1,\ldots x_N)\ P$
P			(x1,...,xN) **P**

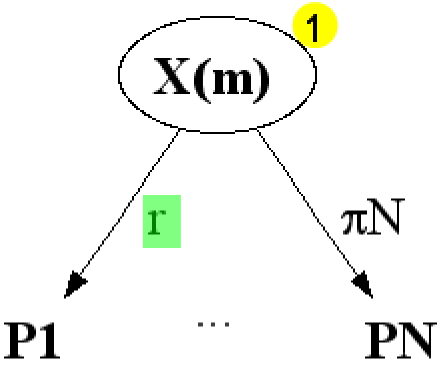
	Definitions
E	$X_1(m_1) = D_1$,..., $X_N(m_N) = D_N$
E	**D1*X1(m1)** ... **DN*XN(mN)**

Graphical Syntax: Process P

	Parallel	Species	Restriction
P	$P_1 \mid \ldots \mid P_M$	$X(n)$, if $X(m) = D$	$(x_1,\ldots x_N)\ (X_1\ (n_1) \mid \ldots \mid X_N\ (n_N))$

 P	 P1 … PM	 m:=n
		X(m)
			 1
			x1,…,xN
			m1:=n1
			X1(m1)
			…
			mN:=nN
			XN(mN)

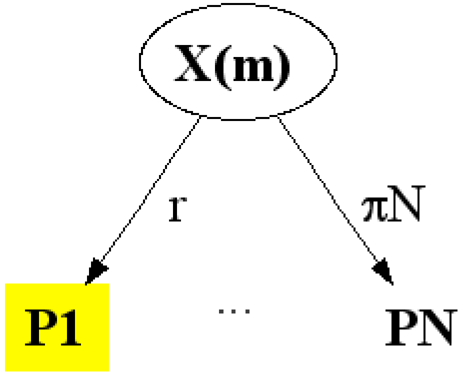
Graphical Semantics: Delay



$$X(m) = r.P_1 + \ldots + \pi_N.P_N$$

$$X(m)$$

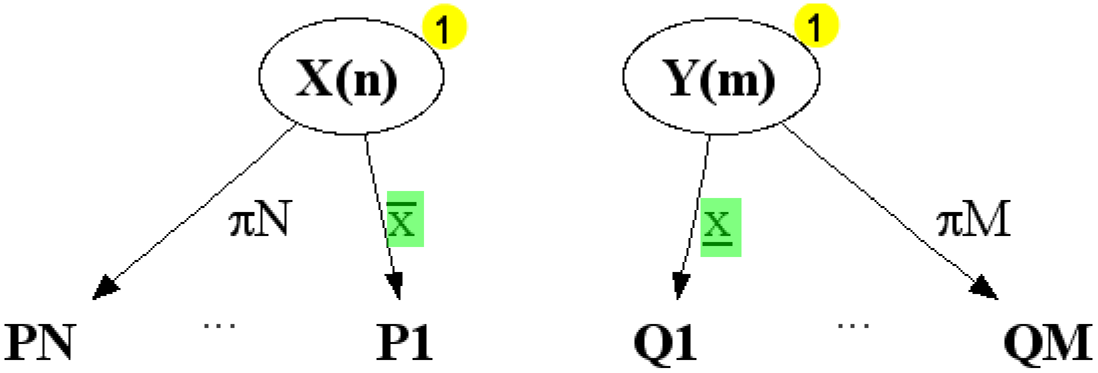
Graphical Semantics: Delay



$$X(m) = r.P_1 + \ldots + \pi_N.P_N$$

$$X(m) \longrightarrow P_1$$

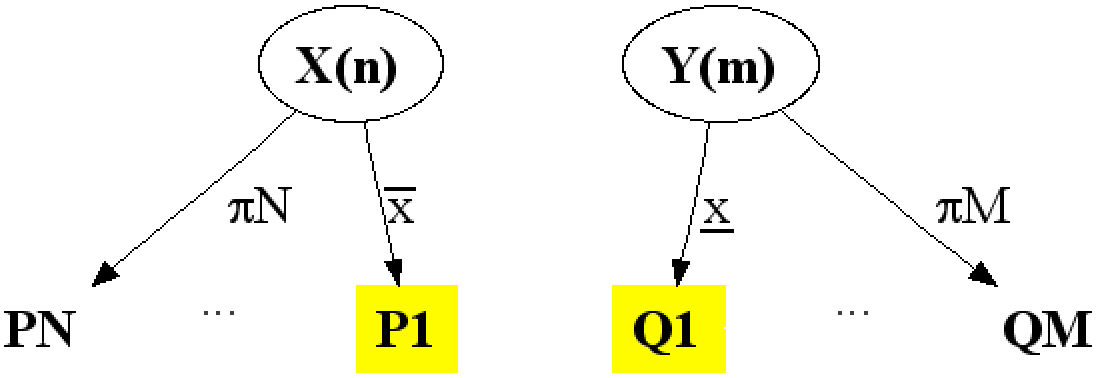
Graphical Semantics: Interaction



$$X(n) = \bar{x}.P_1 + \ldots + \pi_N.P_N \quad , \quad Y(m) = \underline{x}.Q_1 + \ldots + \pi_M.Q_M$$

$$X(n) \mid Y(m)$$

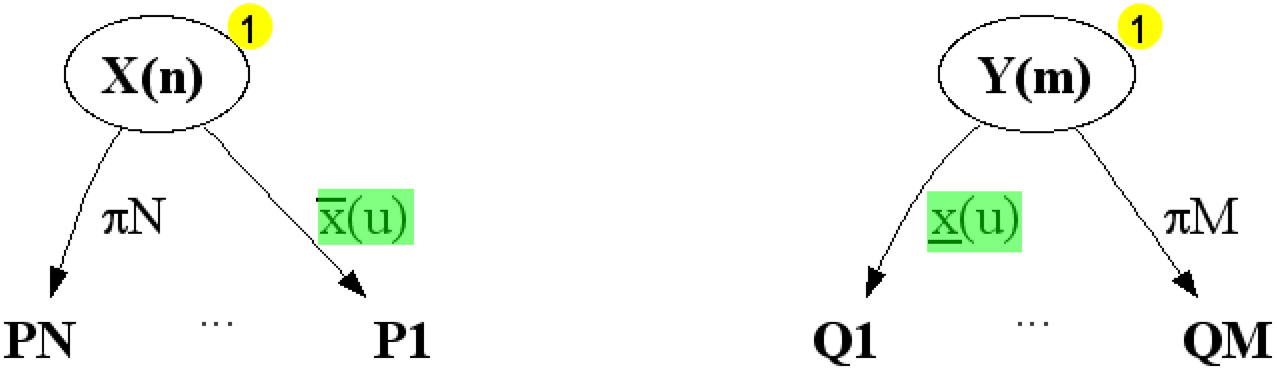
Graphical Semantics: Interaction



$$X(n) = \bar{x}.P_1 + \ldots + \pi_N.P_N \quad , \quad Y(m) = \underline{x}.Q_1 + \ldots + \pi_M.Q_M$$

$$X(n) \mid Y(m) \longrightarrow \quad P_1 \mid Q_1$$

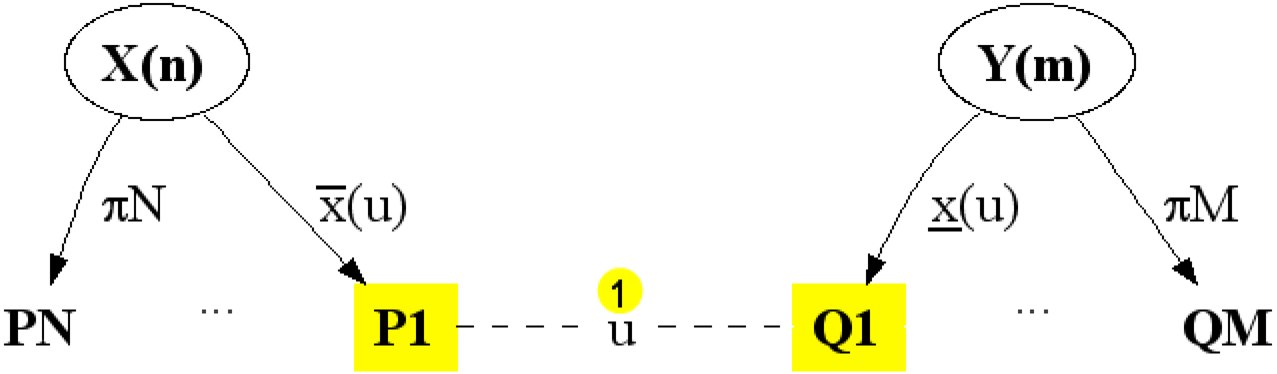
Graphical Semantics: Binding



$$X(n) = \bar{x}(u).P_1 + \ldots + \pi_N.P_N \;\;,\;\; Y(m) = \underline{x}(u).Q_1 + \ldots + \pi_M.Q_M$$

$$X(n) \mid Y(m)$$

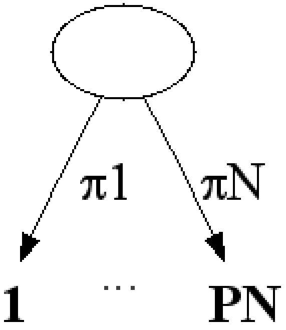
Graphical Semantics: Binding

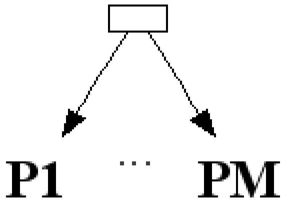
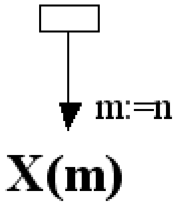


$$X(n) = \bar{x}(u).P_1 + \ldots + \pi_N.P_N \ , \ Y(m) = \underline{x}(u).Q_1 + \ldots + \pi_M.Q_M$$

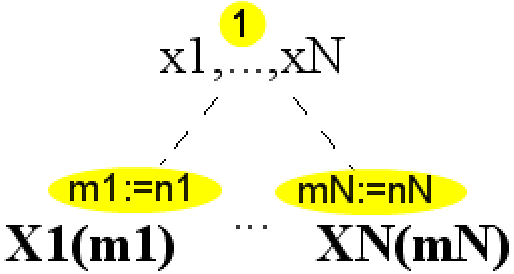
$$X(n) \mid Y(m) \longrightarrow (u)\,(P_1 \mid Q_1)$$

Graphical Syntax

	Choice
M	$\pi_1.P_1 + \ldots + \pi_N.P_N$
M	

	Parallel	Species	Restriction
P	$P_1 \mid \ldots \mid P_M$	$X(n)$, if $X(m) = D$	$(x_1,\ldots x_N)\ P$
P			(x1,...,xN) **P**

	Species	Restriction
P	$X(n)$, if $X(m) = D$	$(x_1,\ldots x_N)\ (X_1\ (n_1) \mid \ldots \mid X_N\ (n_N))$

P	m:=n
	X(m)
		 1
		x1,...,xN
		m1:=n1
		mN:=nN
		X1(m1)
		...
		XN(mN)