Modelling biochemical signalling pathways

or.. *computing science meets the life sciences*

Muffy Calder
Department of Computing Science
University of Glasgow

Joint work with
Jane Hillston and Stephen Gilmore, LFCS, Edinburgh University
Vlad Vyshemirsky, Glasgow University

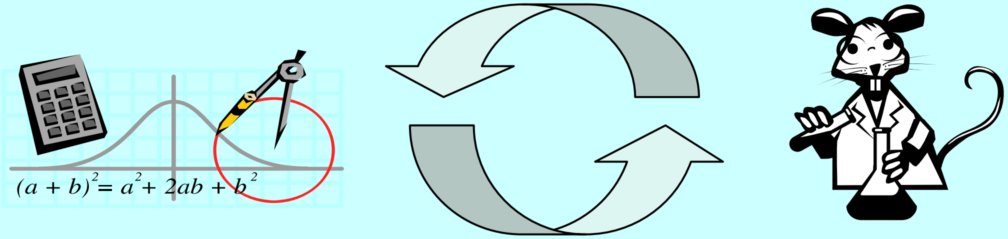
2007

The hypothesis

- Computing science can inspire the way we model and reason about biological systems, particularly ones involving communication

- Not biologically inspired computing, but
computationally inspired biology/biochemistry

- Modelling for intervention (e.g. drug targets)



what does my model do if..

try this experiment..

The talk

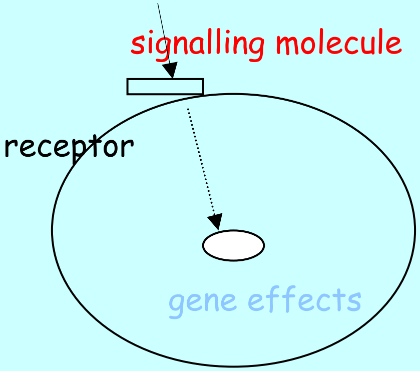
- gentle/naive biochemistry – signalling pathways
- new models of dynamic behaviour of signalling pathways based on CTMCs (continuous time Markov chains)
    - stochastic process algebra (PEPA)
    - model checking continuous stochastic logic (PRISM)
- new quantitative analysis
- relation to traditional ODE models
- Lots of Acronyms
    - RKIP/ERK
    - PEPA
    - ODE
    - PRISM
    - DIZZY

Cell signalling for dummies

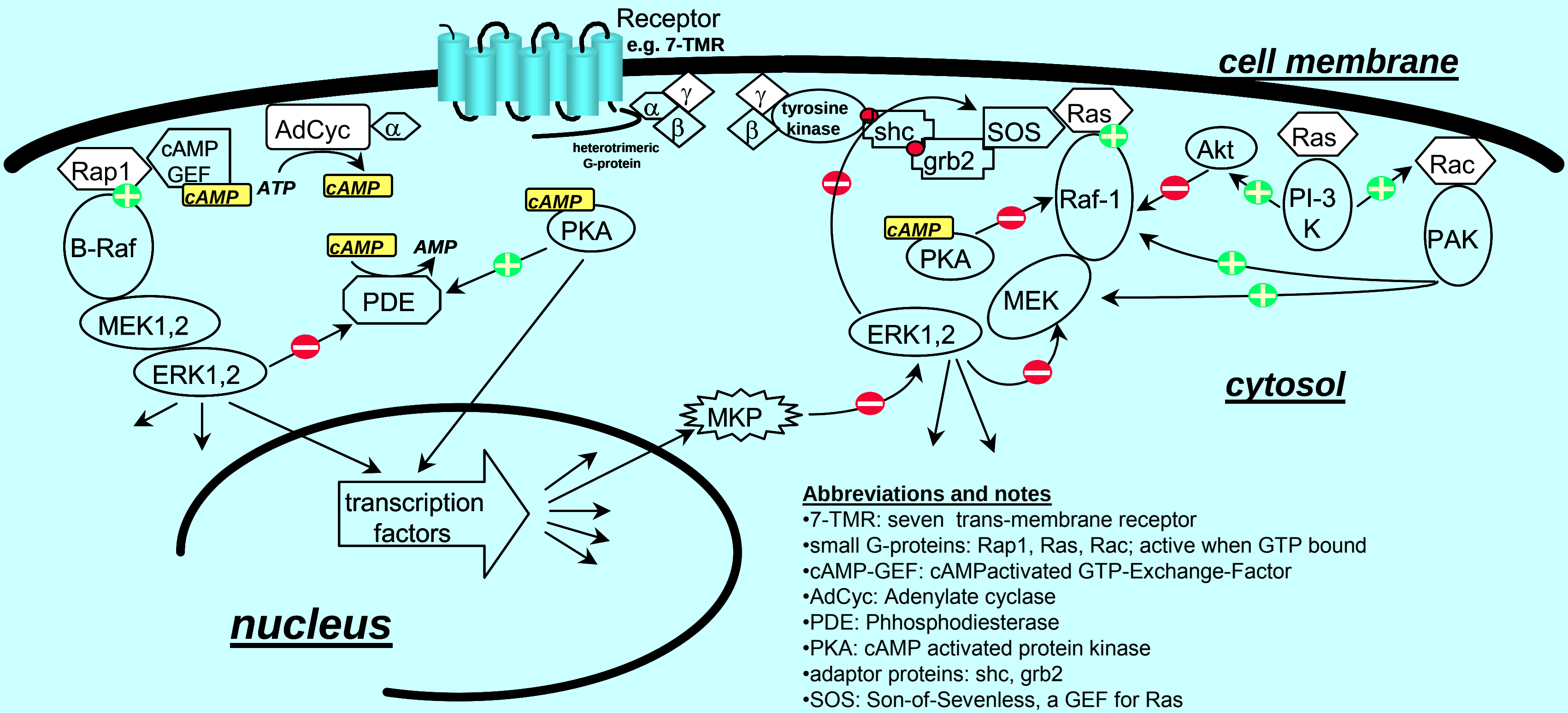
- movement of signal from outside cell to nucleus

- fundamental to cell processes (growth, division, differentiation, apoptosis)

- signalling is via membrane receptors, "signal" is *phosphorylation – accumulation of certain phosphorylated proteins*



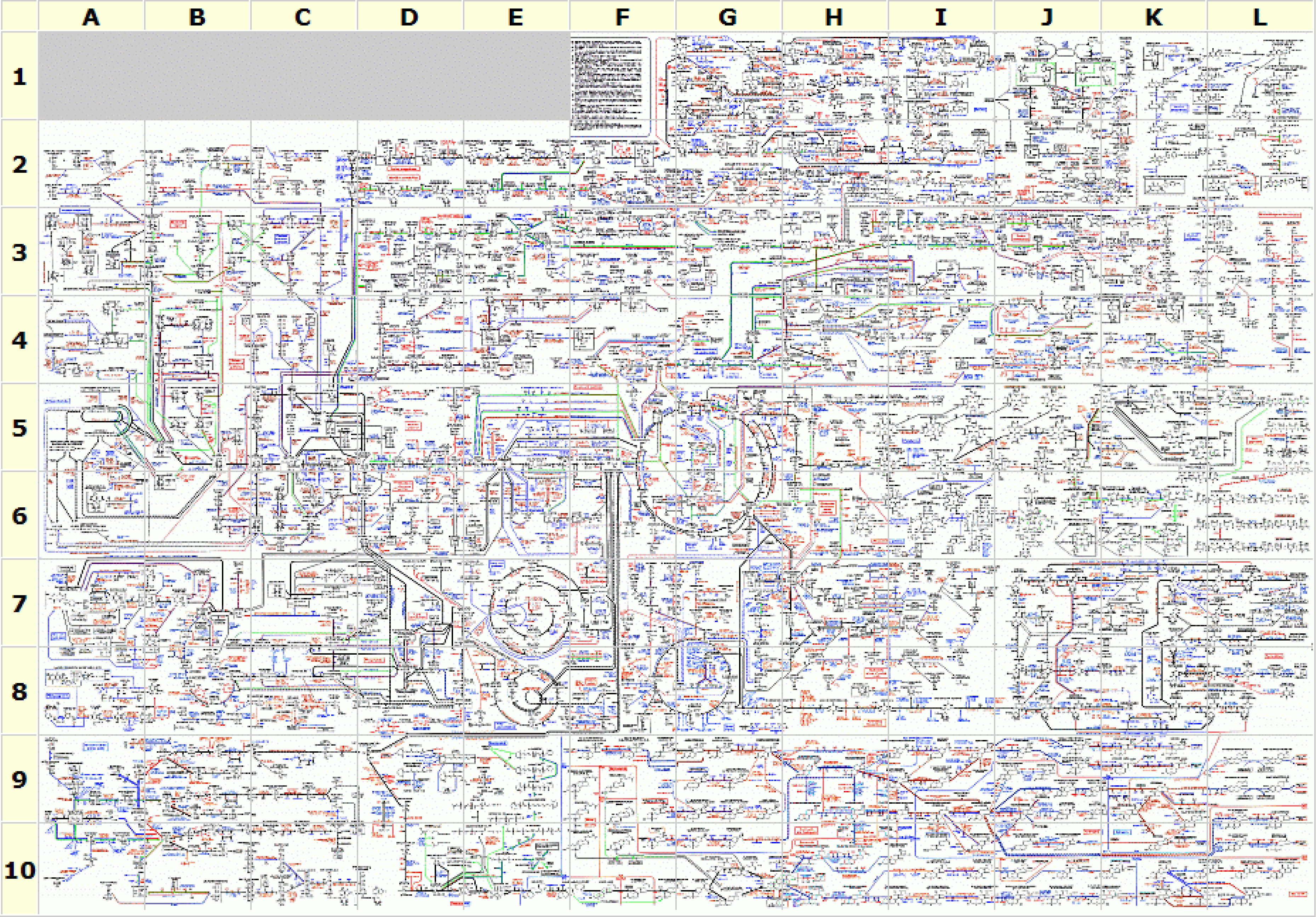
A little more complex.. pathways/networks



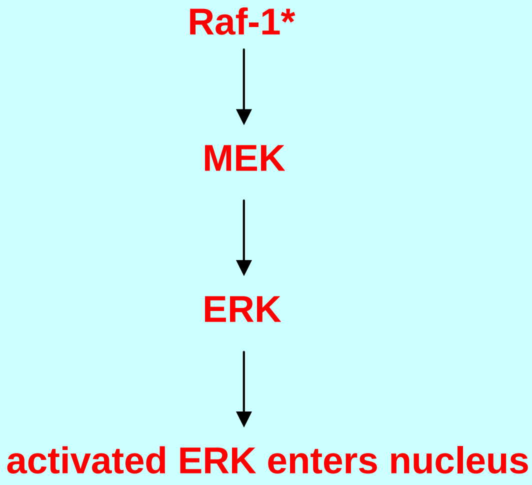
⊕ activation ⊖ inhibition ● phosphorylation

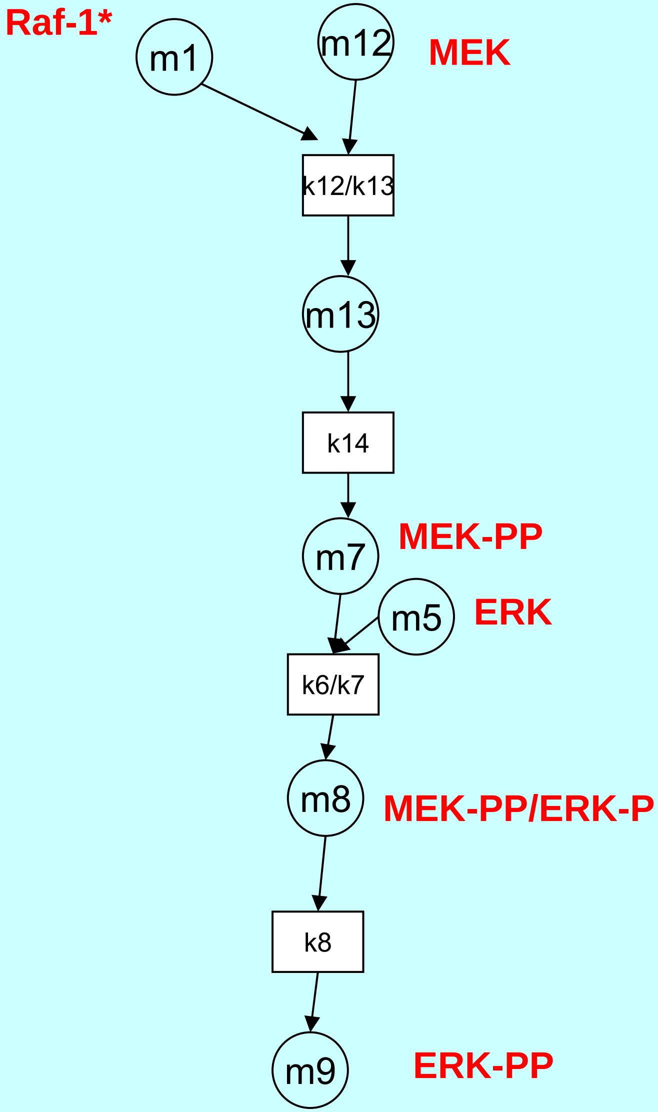
Abbreviations and notes

•7-TMR: seven trans-membrane receptor
•small G-proteins: Rap1, Ras, Rac; active when GTP bound
•cAMP-GEF: cAMPactivated GTP-Exchange-Factor
•AdCyc: Adenylate cyclase
•PDE: Phhosphodiesterase
•PKA: cAMP activated protein kinase
•adaptor proteins: shc, grb2
•SOS: Son-of-Sevenless, a GEF for Ras
•PI-3 K: Phosphatidylinositol-3 kinase
•Akt: a kinase activated by PI-3 K via PI-3 and another kinase, PDK
•PAK: a kinased activated by binding to Rac
•MKP: MAPK phosphatase, dephosphorylates MAPKs

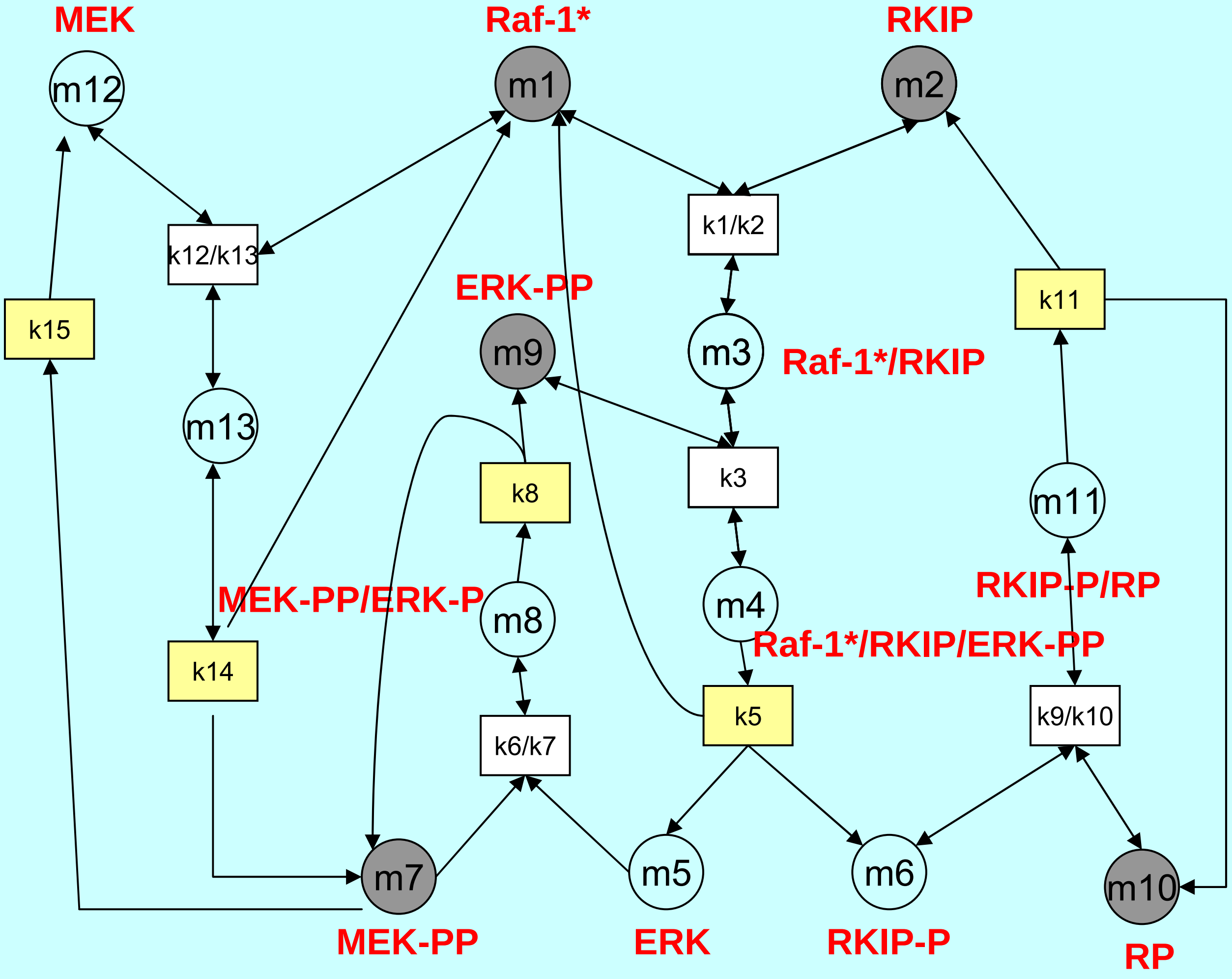

A
B
C
D
E
F
G
H
I
J
K
L
1
2
3
4
5
6
7
8
9
10

Basic ERK Pathway





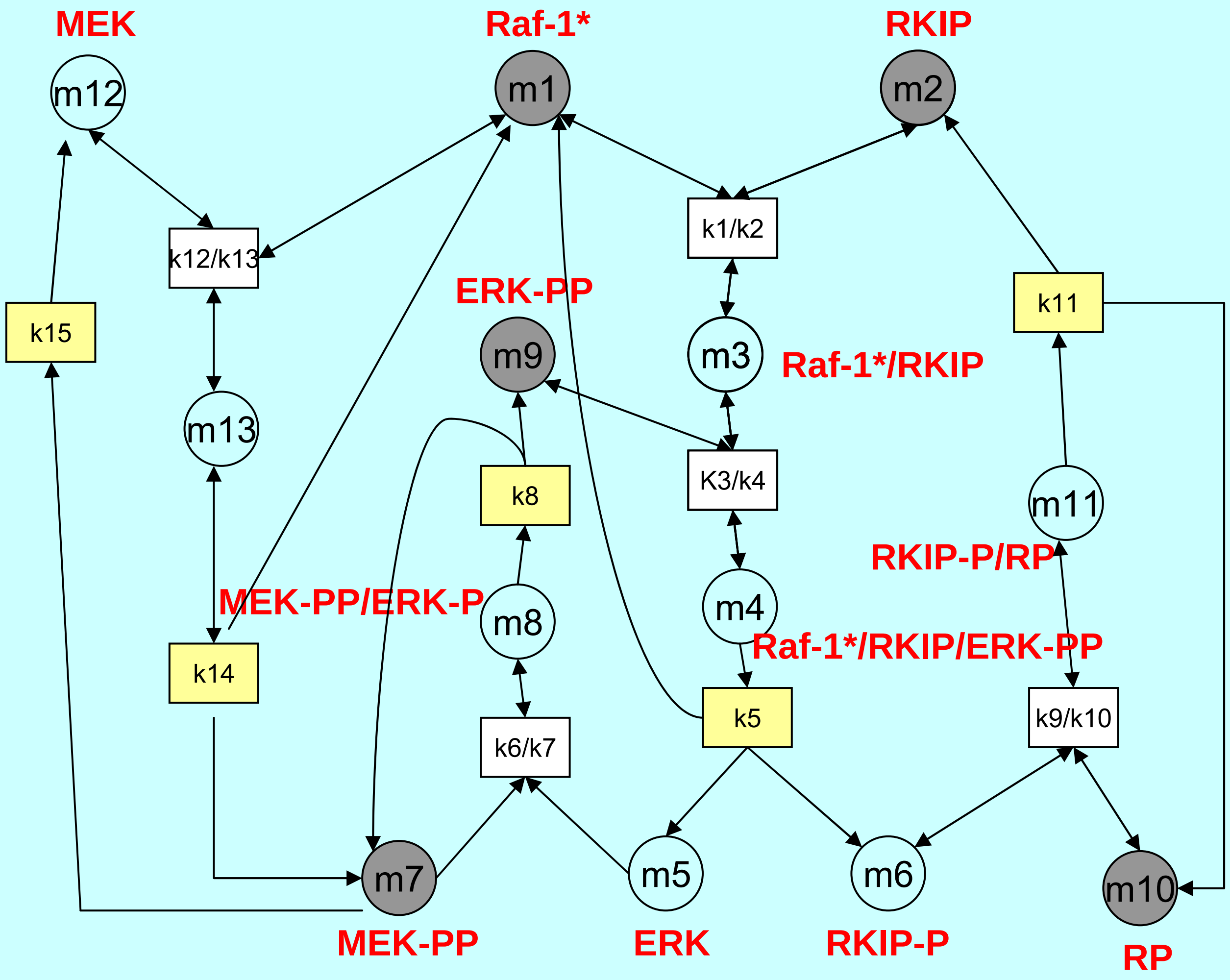
RKIP Inhibited ERK Pathway



CANCER RESEARCH UK
BEATSON LABORATORIES

From paper by Cho, Shim, Kim, Wolkenhauer, McFerran, Kolch, 2003.

RKIP Inhibited ERK Pathway



○ proteins/complexes

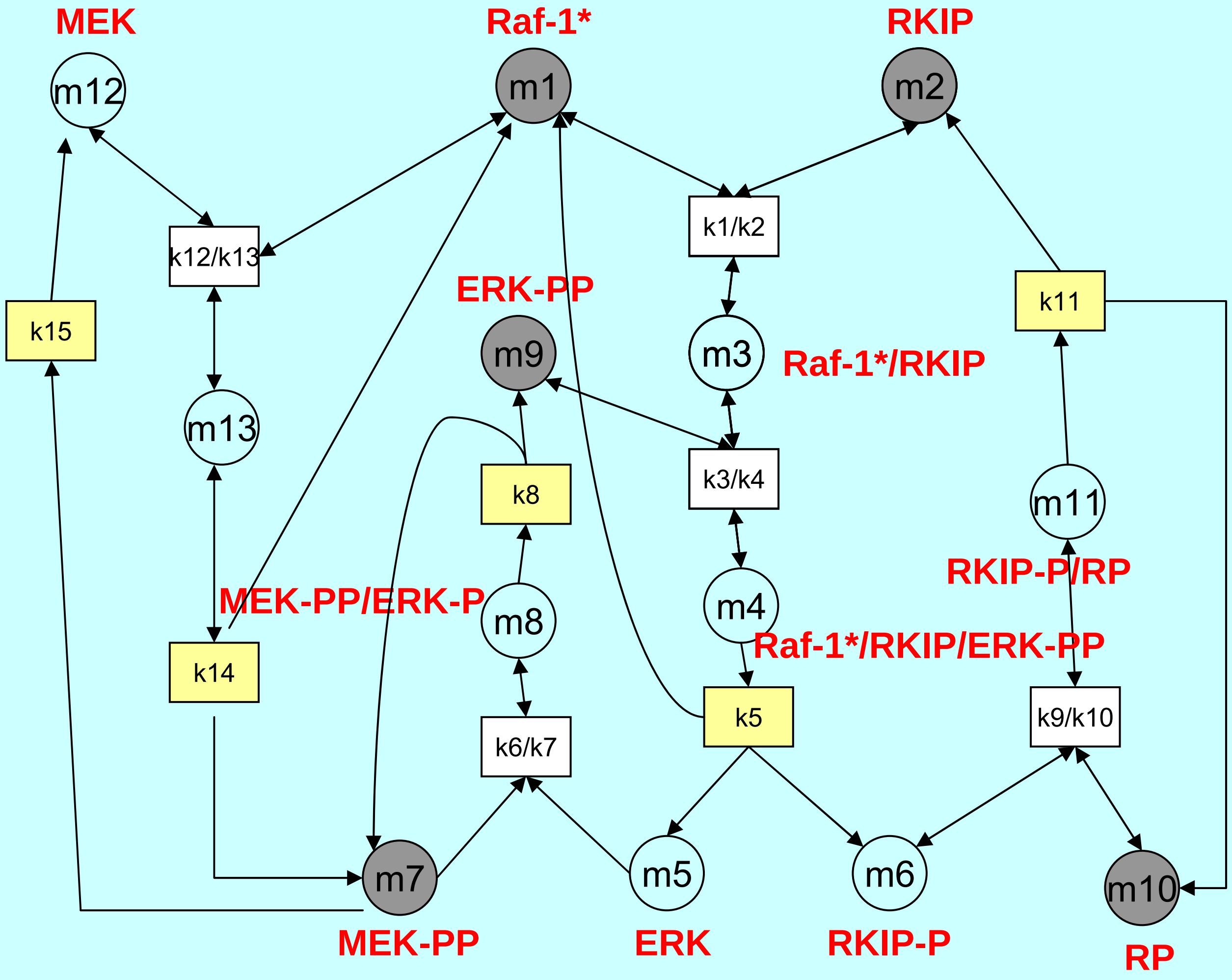
□ forward /backward reactions (associations/disassociations)

🟨 products (disassociations)

m1, m2 .. concentrations of proteins

k1,k2 ..: rate (performance) coefficients

RKIP Inhibited ERK Pathway

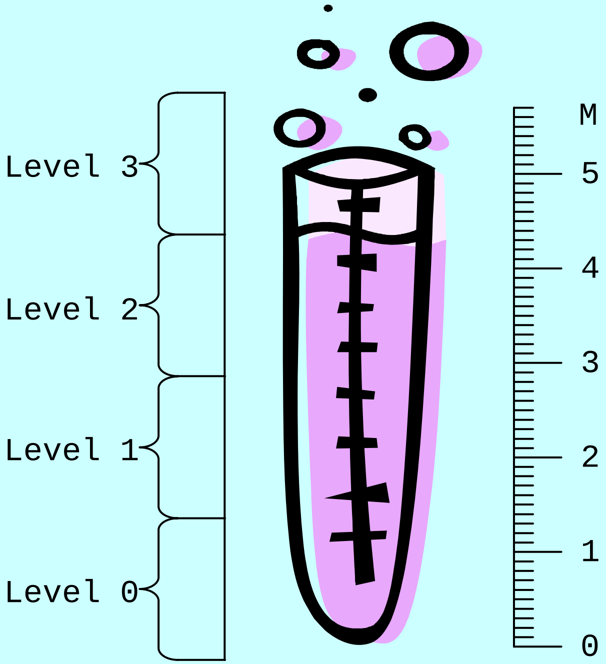


producer/consumer behaviour

- computation
- stochastic
- concurrent
- message passing

Modelling *concentrations* not molecules

Abstract Discrete Values



Continuous Values

levels 0..N *represent* [0,1*M/N), [1*M/N,2*M/N), … , [N-1*M/N,N*M/N]

nb. time is real, concentration is discrete.

PEPA

Process algebra with *performance,* invented by Jane Hillston

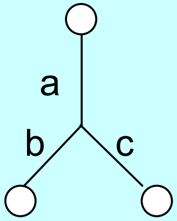
Prefix	*(a,r)*	
Choice	*P1 + P2*	*competition between components (race)*
Cooperation/	*P1 |I| P1*	*a ∉ I independent concurrent (interleaved) actions*
Synchronisation		*a ε I shared action, at rate of slowest* *multiway*
Constant	*A = P*	*assign names to components*

$$P ::= S \mid P\ |I|\ P$$
$$S ::= (\alpha,r).S \mid S+S \mid A$$

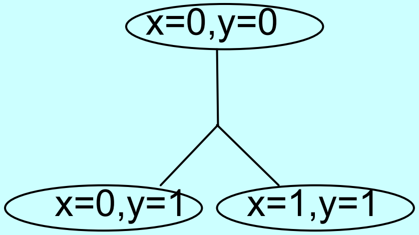
Synchronisation is associative, commutative.
T *is the passive rate.*

Process algebra semantics

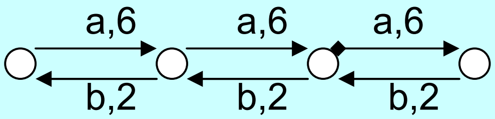
Standard process algebra descriptions denote a form of labelled transition systems or Kripke structures.



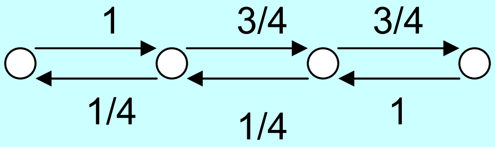
or



Performance evaluation process algebra descriptions denote Continuous Time Markov chains (probabilistic transition systems).

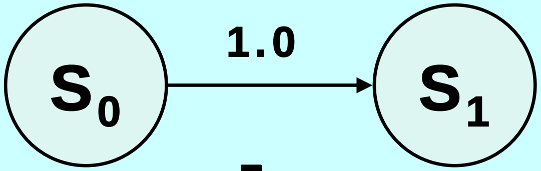


CTMC

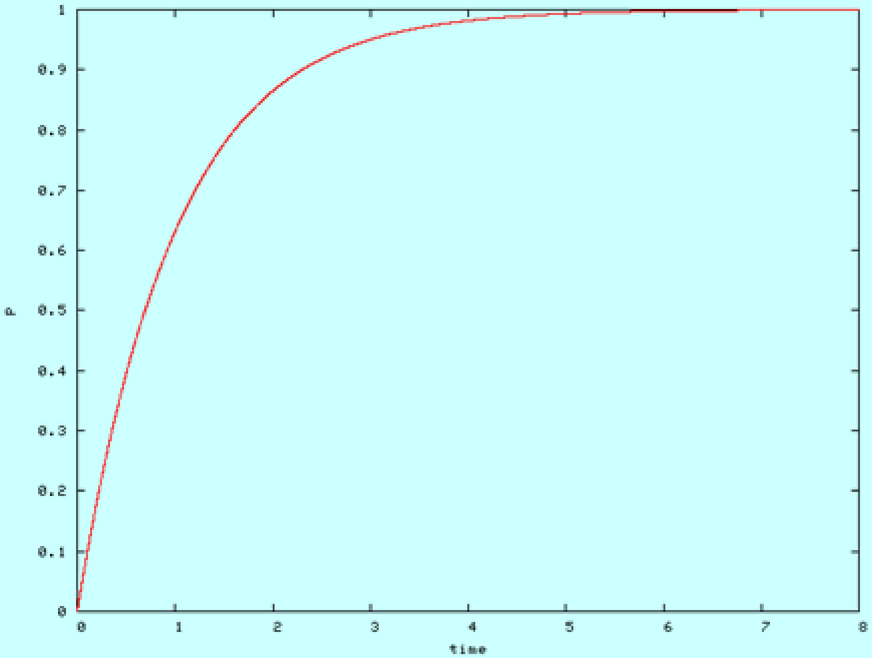


DTMC

Continuous Time Markov Chains



$$P(t) = 1 - e^{-\lambda t}$$
$$\lambda = 1.0$$



State based models for *dynamic, stochastic* behaviour.

- performance coefficients associated with all transitions
- from rates we derive probabilities
- "memoryless" property.

Modelling the ERK Pathway in PEPA

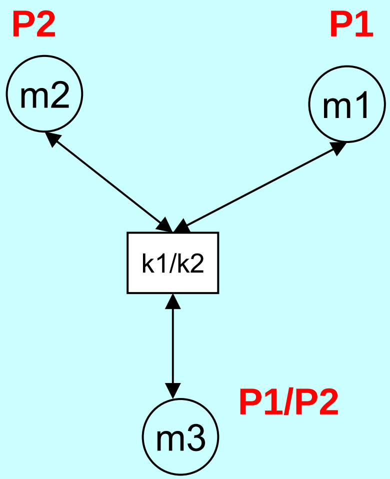
- Each *reaction* is modelled by an event, which has a *performance* coefficient.
- Each *protein* is modelled by a process which synchronises others involved in a reaction. A fine-grained distributed view.

  (reagent-centric view)
- Each *sub-pathway* is modelled by a process which synchronises with other sub-pathways. A coarser grained view.

  (pathway-centric view)

Take the simplest concentration abstraction, N=1, the "high/low" approach.

Modelling reactions

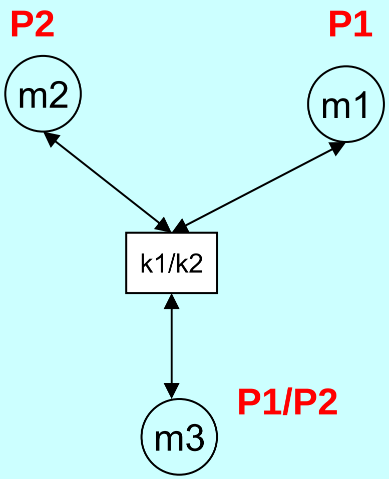


Reaction	**Producer(s)**	**Consumer(s)**
k1react	{P2,P1}	{P1/P2}
k2react	{P1/P2}	{P2,P1}

k1react and k2react will be a *3-way* synchronisations.

(Multiway synch is essential!)

Modelling reactions



Reagent view: models whether or not a reagent can participate in a reaction (observable/unobservable).

Each reagent gives rise to a pair of definitions.

Model equations

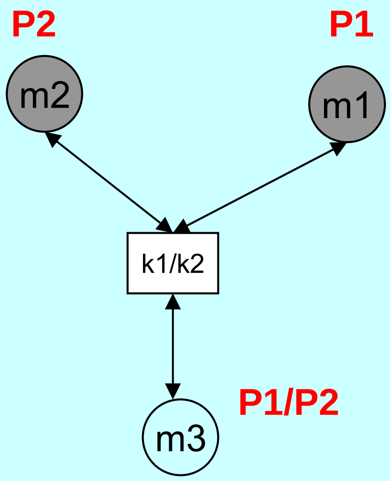
$P1_H$ = (k1react,k1). $P1_L$
$P1_L$ = (k2react,T). $P1_H$

$P2_H$ = (k1react,k1). $P2_L$
$P2_L$ = (k2react,T). $P2_H$

$P1/P2_H$ = (k2react,k2). $P1/P2_L$
$P1/P2_L$ = (k1react,T). $P1/P2_H$

(consumers have passive rate)

Modelling reactions



Reagent view: models whether or not a reagent can participate in a reaction (observable/unobservable).

Each reagent gives rise to a pair of definitions.

Model equations

$P1_H$ = (k1react,k1). $P1_L$
$P1_L$ = (k2react,T). $P1_H$

$P2_H$ = (k1react,k1). $P2_L$
$P2_L$ = (k2react,T). $P2_H$

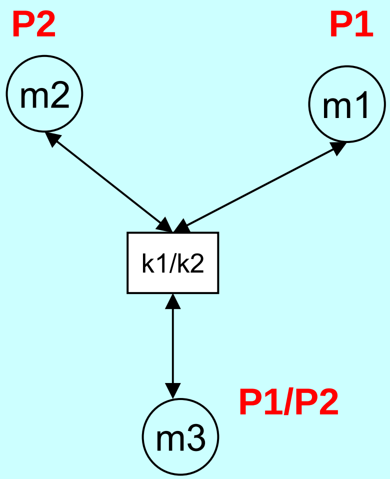
$P1/P2_H$ = (k2react,k2). $P1/P2_L$
$P1/P2_L$ = (k1react,T). $P1/P2_H$

Model configuration

$P1_H$ |k1react,k2react| $P1_H$ |k1react,k2react| $P1/P2_L$

(assuming initial concentrations of m1 and m2)

Modelling reactions



N levels:

Model equations

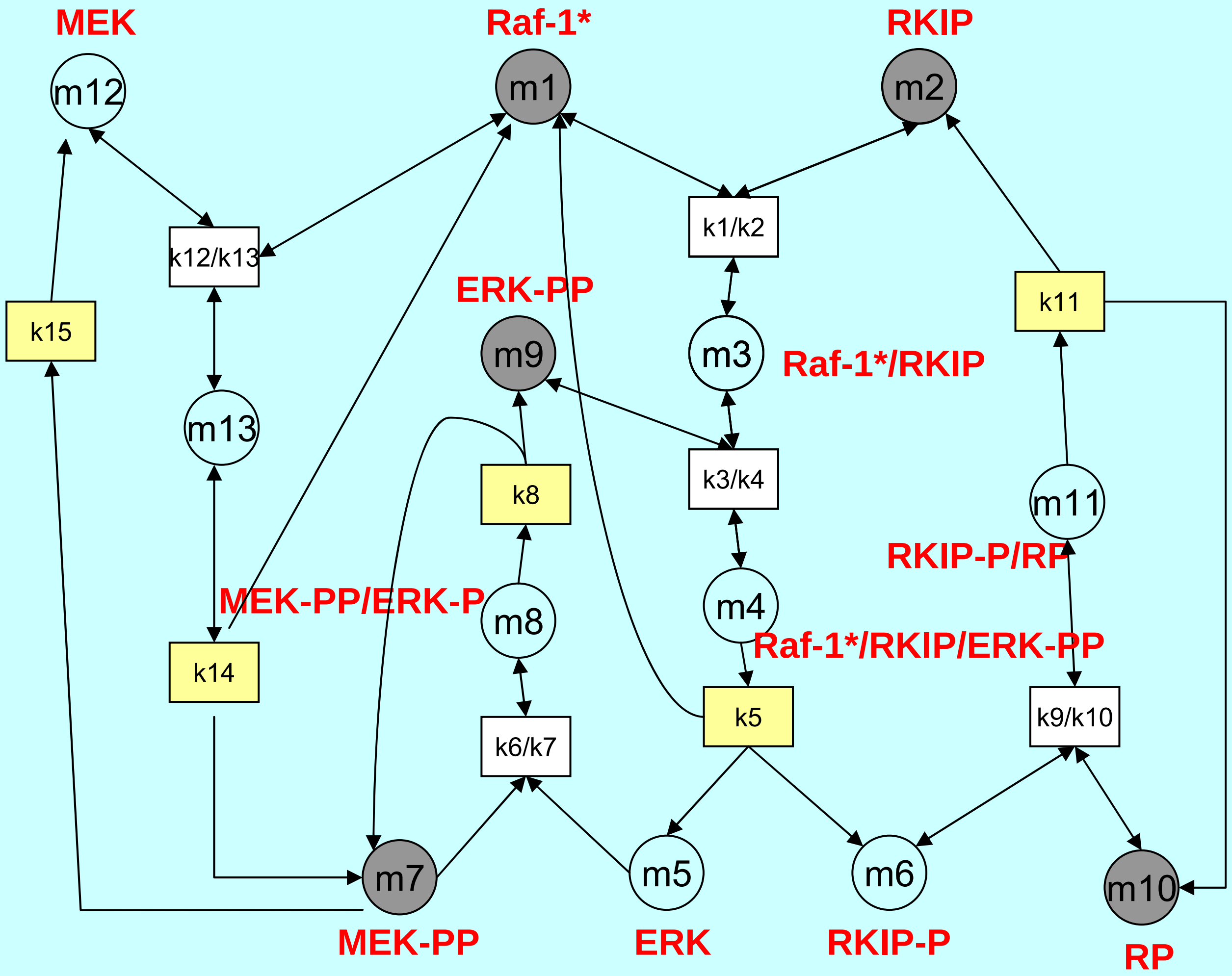
$P1_N$ = (k1react,N*k1). $P1_{N-1}$
$P1_{N-1}$ = (k1react,(N-1)*k1). $P1_{N-2}$ + (k2react,T).$P1_N$

…

$P1_1$ = (k1react,k1). $P1_0$ + (k2react,T).$P1_2$

$P1_0$ = (k2react,T). $P1_1$

….



Reagent view:

Raf-1*$_H$ = (k1react,k1). Raf-1*$_L$ + (k12react,k12). Raf-1*$_L$

Raf-1*$_L$ = (k5product,T). Raf-1*$_H$ +(k2react,T). Raf-1*$_H$ + (k13react,T). Raf-1*$_H$ + (k14product,T). Raf-1*$_H$

…

(26 equations)

Signalling Dynamics

Reagent view: model configuration

Raf-1*$_H$ |k1react,k12react,k13react,k5product,k14product|

RKIP$_H$ | k1react,k2react,k11product |

Raf-1*$_H$/RKIP$_L$ |k3react,k4react|

Raf-1*/RKIP/ERK-PP$_L$ |k3react,k4react,k5product|

ERK-P$_L$ |k5product,k6react,k7react|

RKIP-P$_L$ |k9react,k10react|

RKIP-P/RP$_L$|k9react,k10react,k11product|

RP$_H$||

MEK$_L$|k12react,k13react,k15product|

MEK/Raf-1*$_L$|k14product|

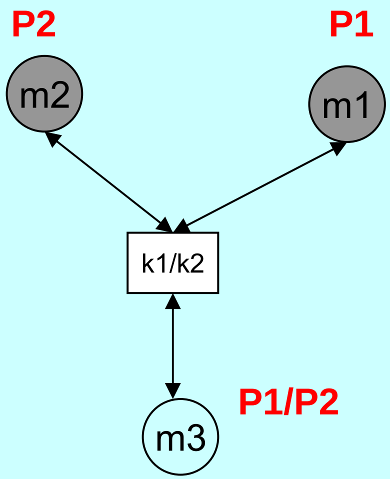
MEK-PP$_H$ |k8product,k6react,k7react|

MEK-PP/ERK$_L$|k8product|

MEK-PP$_H$|k8product|

ERK-PP$_H$

Modelling reactions



Pathway view: model *chains* of behaviour flow

Model equations

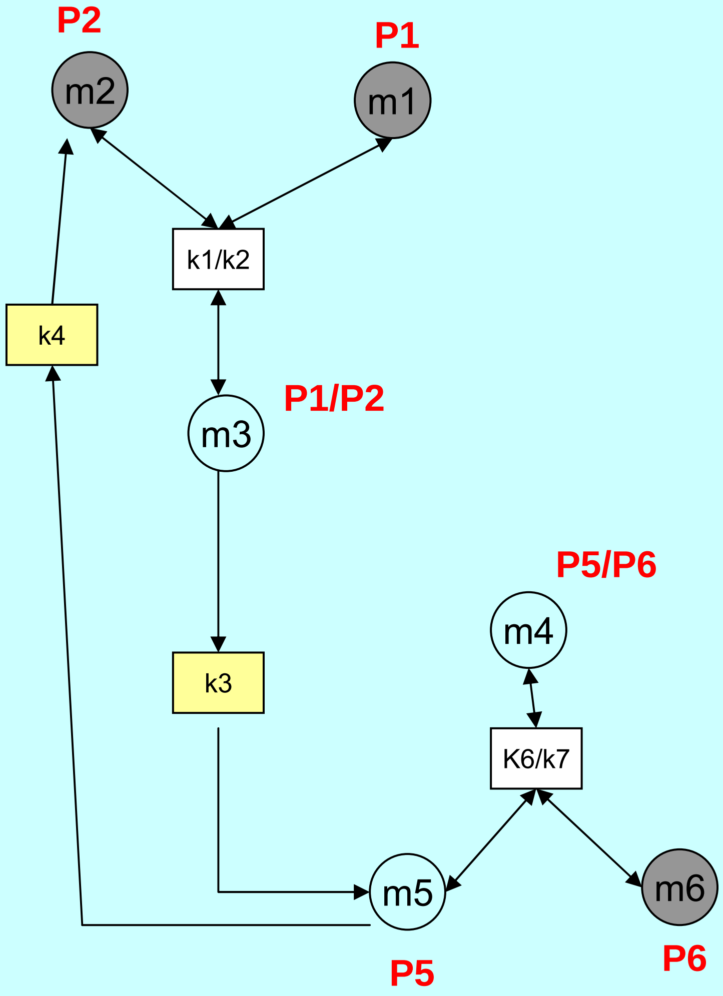
$\text{Pathway}_1 = (\text{k1react},\text{k1}).\ (\text{k2react},\text{k2}).\text{Pathway}_1$

Model configuration

Pathway_1

(assuming initial concentrations of m1 and m2)

Note: only one component!



Pathway view: model chains of behaviour flow.

Two pathways, corresponding to initial concentrations:

Path10 = (k1react,k1). Path11

Path11 = (k2react).Path10 + (k3product,k3).Path12

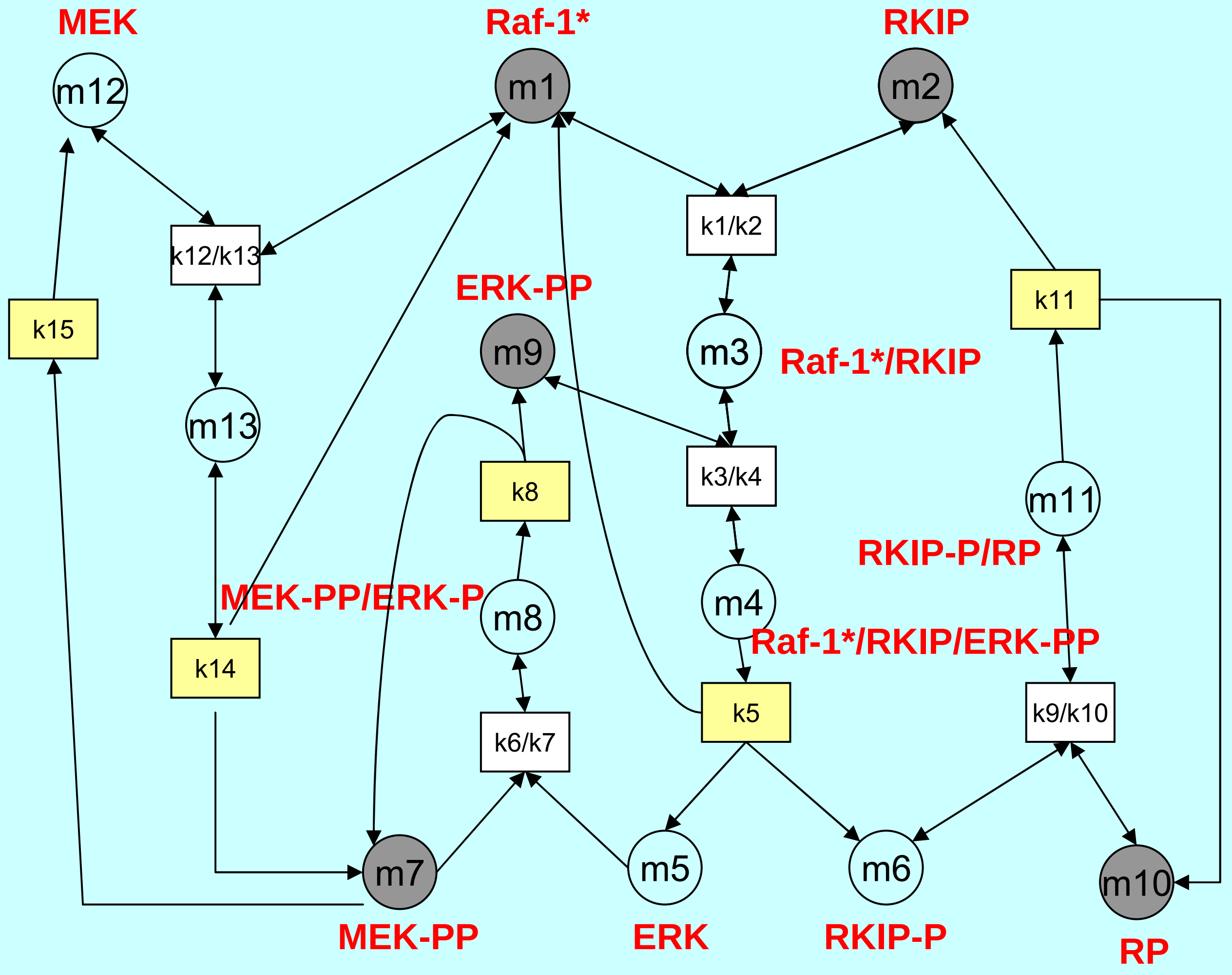
Path12 = (k4product,k4).Path10 + (k6react,k6).Path13

Path13 = (k7react,k7).Path12

Path2 = (k6react,k6). (k7react,k6).Path2

Pathway view: model configuration

Path10 | k6react,k7react | Path2



Pathway view:

Pathway10 = (k9react,k9). ((k11product,k11). Pathway10 + (k10react,k10). Pathway10)

…

(5 pathways)

Pathway view: model configuration

Pathway10 |k12react,k13react,k14product| Pathway40
|k3react,k4react,k5product,k6react,k7react,k8product| Pathway30
|k1react,k2react,k3react,k4react,k5product| Pathway20
|k9react,k10react,k11product| Pathway50

simpler!

What is the difference?

- reagent-centric view is a *fine* grained view
- pathway-centric view is a *coarse* grained view
  - reagent-centric is easier to derive from data
  - pathway-centric allows one to build up networks from already known components

The two models are equivalent!

The equivalence proof, based on *bisimulation* between steady state solutions, unites two views of the same biochemical pathway.

Steady State Solution

1	0.04135079004156481
2	0.020806115102310632
3	0.07346775929692899
4	0.006935371700770152
5	0.06516104016641672
6	0.03737546622097119
7	0.011336715749471194
8	0.036048205933593286
9	0.004639841577167708
10	0.005691394350960237
11	0.04138456618620803
12	0.0025828089820320505
13	0.004807783620797024
14	0.04817123798507296
15	0.018640671069835055
16	0.016743539619515142
17	0.02162874351056745
18	0.0028912552492803816
19	0.004970238100423158
20	0.02076780718322302
21	0.1840054851485999
22	0.008846052672337585
23	0.01413218356459678
24	0.0030482221649047224
25	0.0020844704151460223
26	0.20477329233182312
27	0.09642576891046874
28	0.0012831731450123965

Reagent view

1	0.04135079004156353
2	0.020806115102310604
3	0.07346775929692419
4	0.006935371700769834
5	0.06516104016641262
6	0.03737546622096783
7	0.011336715749470889
8	0.03604820593359156
9	0.005691394350959787
10	0.004639841577167543
11	0.04138456618620752
12	0.04817123798507505
13	0.0025828089820318246
14	0.01864067106983504
15	0.004807783620796737
16	0.01674353961951507
17	0.020767807183224345
18	0.021628743510568222
19	0.18400548514860549
20	0.002891255249280038
21	0.008846052672337464
22	0.004970238100423424
23	0.014132183564597499
24	0.20477329233182964
25	0.09642576891047139
26	0.0030482221649046053
27	0.0020844704151453983
28	0.0012831731450119671

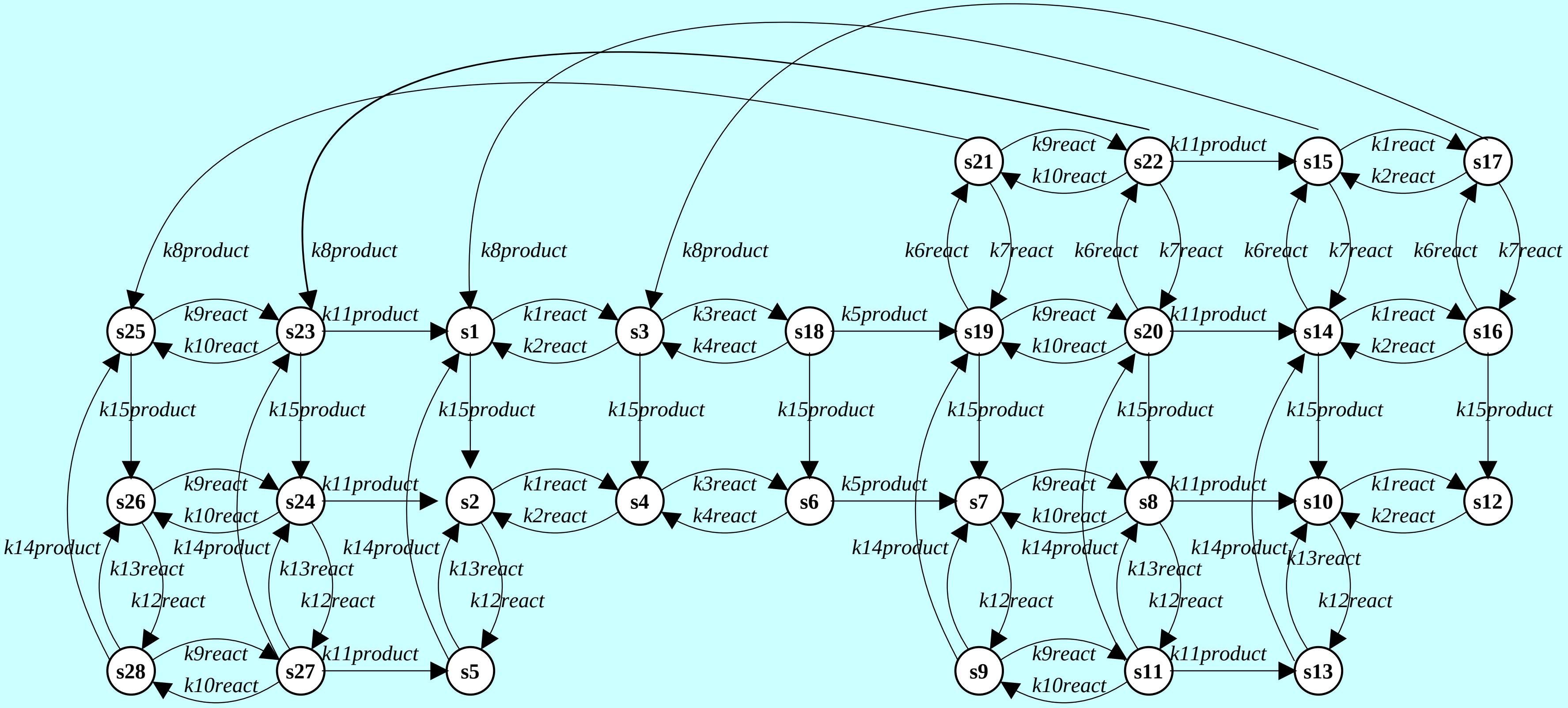
Pathway view

State space of reagent and pathway model

state s1	reagent-view Raf-1*$_H$, RKIP$_H$,Raf-1*/RKIP$_L$,Raf-1*/RKIP/ERK-PP$_L$, ERK$_L$,RKIP-P$_L$, RKIP-P/RP$_L$, RP$_H$, MEK$_L$,MEK/Raf-1*$_L$,MEK-PP$_H$,MEK-PP/ERK$_L$,ERK-PP$_H$ pathway view Pathway50,Pathway40,Pathway20,Pathway10
s2 . . . s28	…

(28 states)

State space of reagent and pathway model



What do you do with these two models?

-investigate properties of the underlying Markov model.

Transient analysis

e.g. analysis to determine whether a state will be reached.

Steady state analysis

e.g. analysis of the steady state solution – behaviour in the long run.

Note: there isn't one steady state, but a very large "cycle"!

Static Analysis

Check for deadlocks, livelocks

- something “wrong” with model
- e.g. an incompleteness   (discovered problem with RKIP model)

Quantitative Analysis

Generate steady-state probability distribution (using linear algebra).

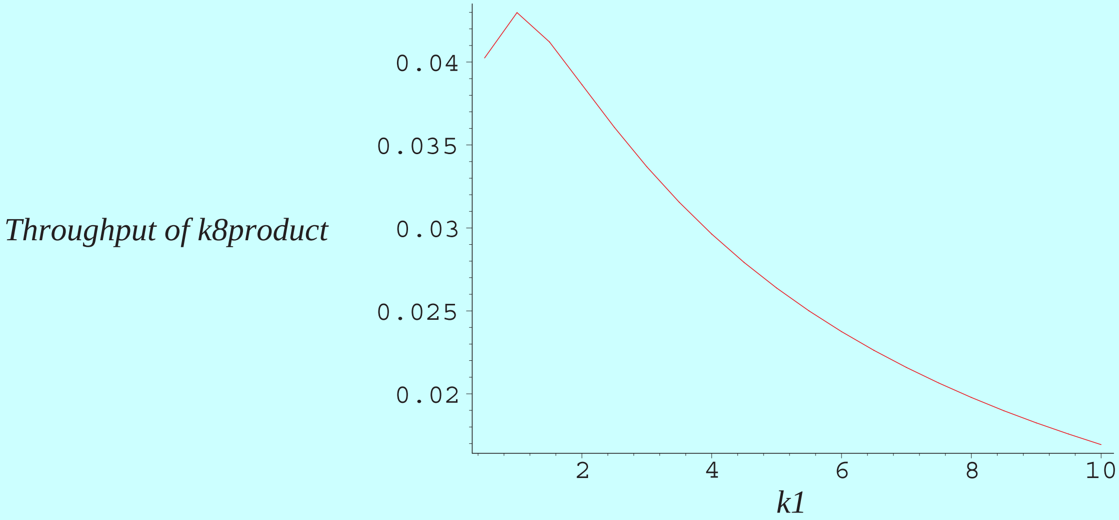
1. Use state finder (in reagent model) to aggregate probabilities and look for proteins high or low.

Example

increase k1 from 1 to 100 and the probability of being in a state with ERK-PP_H drops from .257 to .005

2. Perform throughput analysis (in pathway model).

Quantitative Analysis



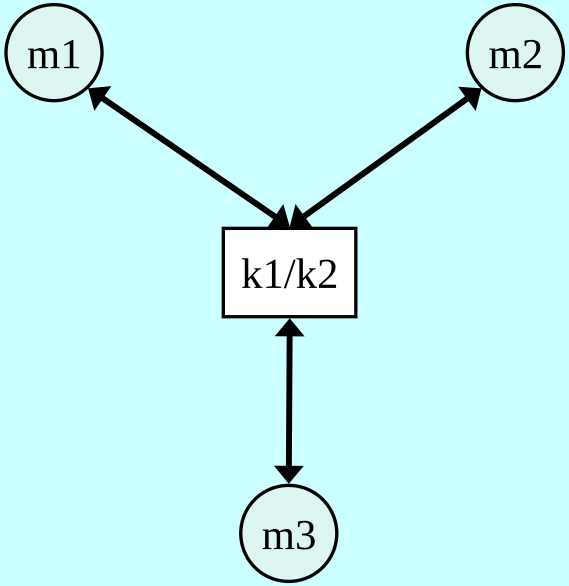
Effect of binding of RKIP to Raf-1* on ERK-PP

(increasing the rate of k1 on k8product throughput (rate x probability))

We can see the effect of RKIP, but is this only indicative?
How accurate is the model?

Ordinary differential equations

Mass action kinetics (semantics):

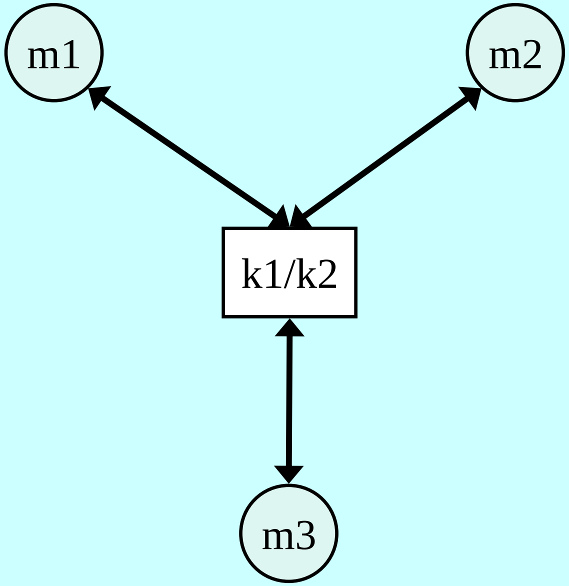


$$\begin{cases} \dfrac{dm_1}{dt} = -k_1 m_1 m_2 + k_2 m_3 \\ \dfrac{dm_2}{dt} = -k_1 m_1 m_2 + k_2 m_3 \\ \dfrac{dm_3}{dt} = k_1 m_1 m_2 - k_2 m_3 \end{cases}$$

note, nonlinear ODEs (second order reactions)

more accurate rates

Mass action kinetics (semantics):



Consider forward only:

$$\frac{dm_3}{dt} = k_1 m_1 m_2$$

But the PEPA model has rate k1.
A consequence of synchronisation (slowest).

More accurate rates, and model checking capabilities will be obtained by moving to a state-based PRISM model.

PRISM reagant model

PRISM – probabilistic model checker developed by Marta Kwiatkowska et al

Temporal reasoning (experiments, open formulae) for

- DTMCs
- MDPs
- CTMCs

- Prism modelling language – processes and synchronising transitions. Rates are *multiplied*.

```
const int N = 7;
const double R = 2.5/N  /* 2.5 is initial concentration */

module RAF1
        RAF1: [0..N] init N;
    [r1] (RAF1 > 0) -> RAF1*R: (RAF1' = RAF1 - 1);
    [r2] (RAF1 < N) -> 1: (RAF1' = RAF1 + 1);
    [r5] (RAF1 < N) -> 1: (RAF1' = RAF1 + 1);
    endmodule

module RKIP
        RKIP: [0..N] init N;
    [r1] (RKIP > 0) -> RKIP*R: (RKIP' = RKIP - 1);
    [r2] (RKIP < N) -> 1: (RKIP' = RKIP + 1);
    [r11] (RKIP < N) -> 1: (RKIP' = RKIP + 1);
    endmodule


…


module Constants
             x: bool init true;
    [r1] (x) -> 0.53/R: (x' = true);
    [r2] (x) -> 0.0072/R: (x' = true);
    [r3] (x) -> 0.625/R: (x' = true);
    [r4] (x) -> 0.00245/R: (x' = true);


 system   RAF1 || RKIP || RAF1/RKIP …   || Constants      endsystem
```

```
const int N = 7;
const double R = 2.5/N  /* 2.5 is initial concentration */

module RAF1
        RAF1: [0..N] init N;
     [r1] (RAF1 > 0) -> RAF1*R: (RAF1' = RAF1 - 1);
     [r2] (RAF1 < N) -> 1: (RAF1' = RAF1 + 1);
     [r5] (RAF1 < N) -> 1: (RAF1' = RAF1 + 1);
     endmodule

module RKIP
        RKIP: [0..N] init N;
     [r1] (RKIP > 0) -> RKIP*R: (RKIP' = RKIP - 1);
     [r2] (RKIP < N) -> 1: (RKIP' = RKIP + 1);
     [r11] (RKIP < N) -> 1: (RKIP' = RKIP + 1);
     endmodule


…


module Constants
              x: bool init true;
     [r1] (x) -> 0.53/R: (x' = true);
     [r2] (x) -> 0.0072/R: (x' = true);
     [r3] (x) -> 0.625/R: (x' = true);
     [r4] (x) -> 0.00245/R: (x' = true);



 system   RAF1 || RKIP || RAF1/RKIP …   || Constants      endsystem
```

rates

Example rate of r1: (N*R) * (N*R) * 1 * 0.53/R = 7*0.53*2.5

*RAF1   *   RKIP   * RAP1/RKIP * Constants*

Temporal properties

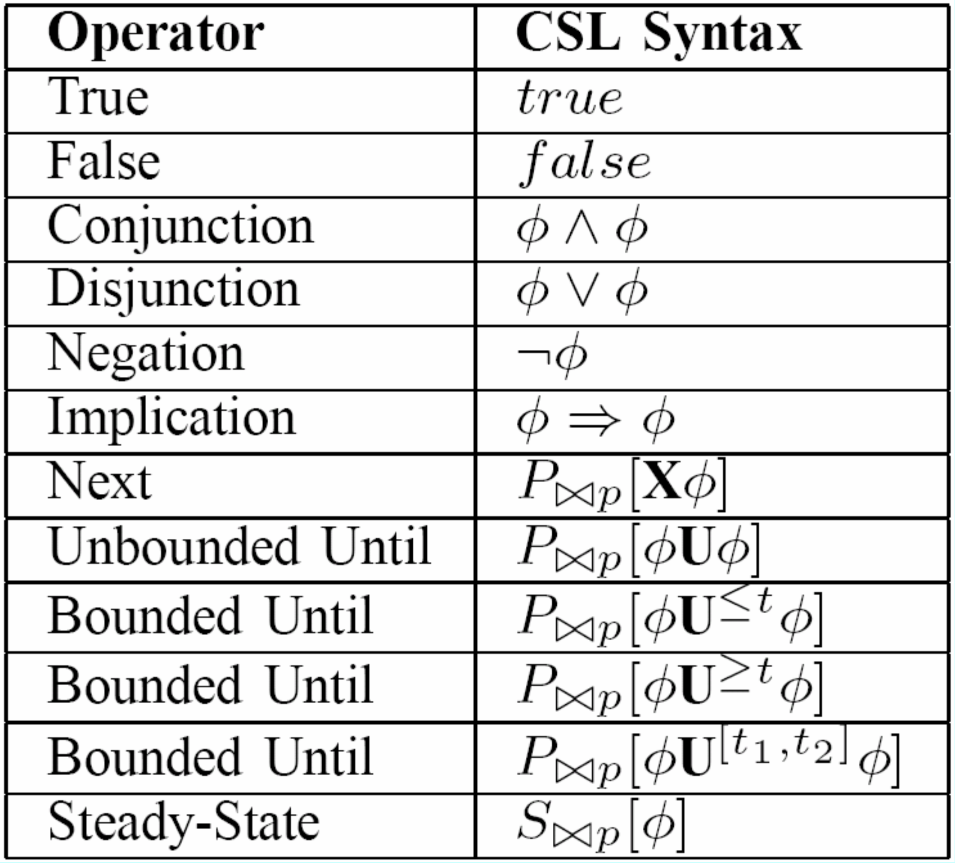
1. stability of protein (steady state)

2. monotonic decrease of protein (transient)

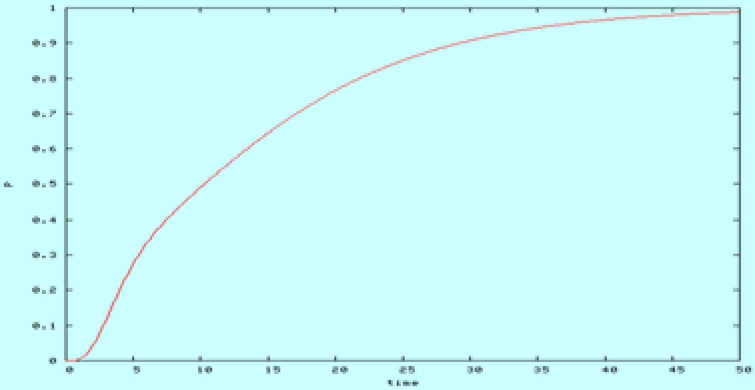
3. protein stability with varying rates (steady state)

4. protein activation order (transient)

CSL (Continuous Stochastic Logic)



Operator	**CSL Syntax**
True	$true$
False	$false$
Conjunction	$\phi \wedge \phi$
Disjunction	$\phi \vee \phi$
Negation	$\neg\phi$
Implication	$\phi \Rightarrow \phi$
Next	$P_{\bowtie p}[\mathbf{X}\phi]$
Unbounded Until	$P_{\bowtie p}[\phi\mathbf{U}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{\leq t}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{\geq t}\phi]$
Bounded Until	$P_{\bowtie p}[\phi\mathbf{U}^{[t_1,t_2]}\phi]$
Steady-State	$S_{\bowtie p}[\phi]$



$$P_{=?}[(true)U_{\leq 150}(protein = N)]$$

$$S_{=?}[(protein > 10)]$$

$$P_{\geq 1}[(protein1 < C)U(protein1 > C)]$$

Temporal properties

1. stability of protein (steady state)

   $S{=}_{?}$ [(RAF1>+ C-1) ^ (RAF1<= C+1)]

2. monotonic decrease of protein (transient)

   $P{=}_{>=1}$ [(true) U ((Protein = C) ^ $P{=}_{>=0.95}$ [X(Protein = C-1)])]

   $P{=}_{?}$ [(true) $U^{<=120}$ (Protein > C){RAF1=C}]

3. stability with varying rates (steady state)

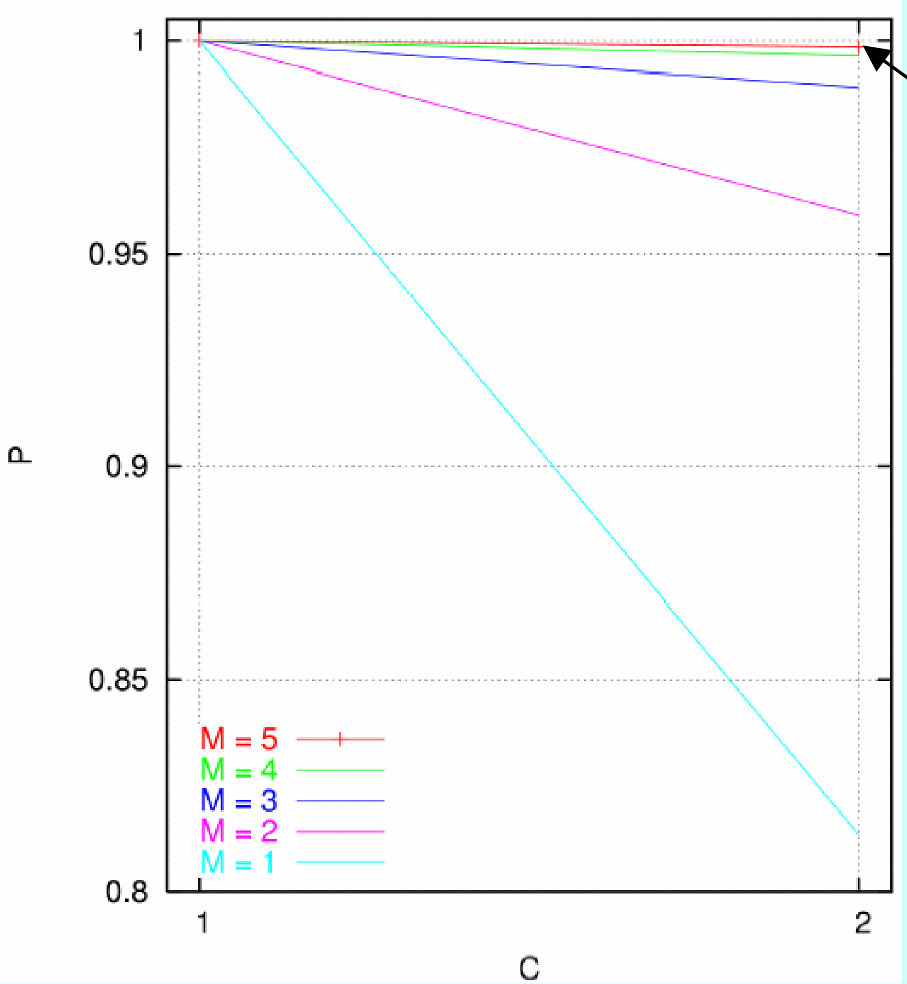
   $S{=}_{?}$ [(RAF1>=2) ^ (RAF1<=3)]

   $S{=}_{?}$ [(RAF1>=0) ^ (RAF1<=1)] (vary k1)

4. protein activation order (transient)

   $P{=}_{?}$ [(RAF1/RKIP/ERK-PP < M) U (RAF1/RKIP = C)]

4. protein activation order



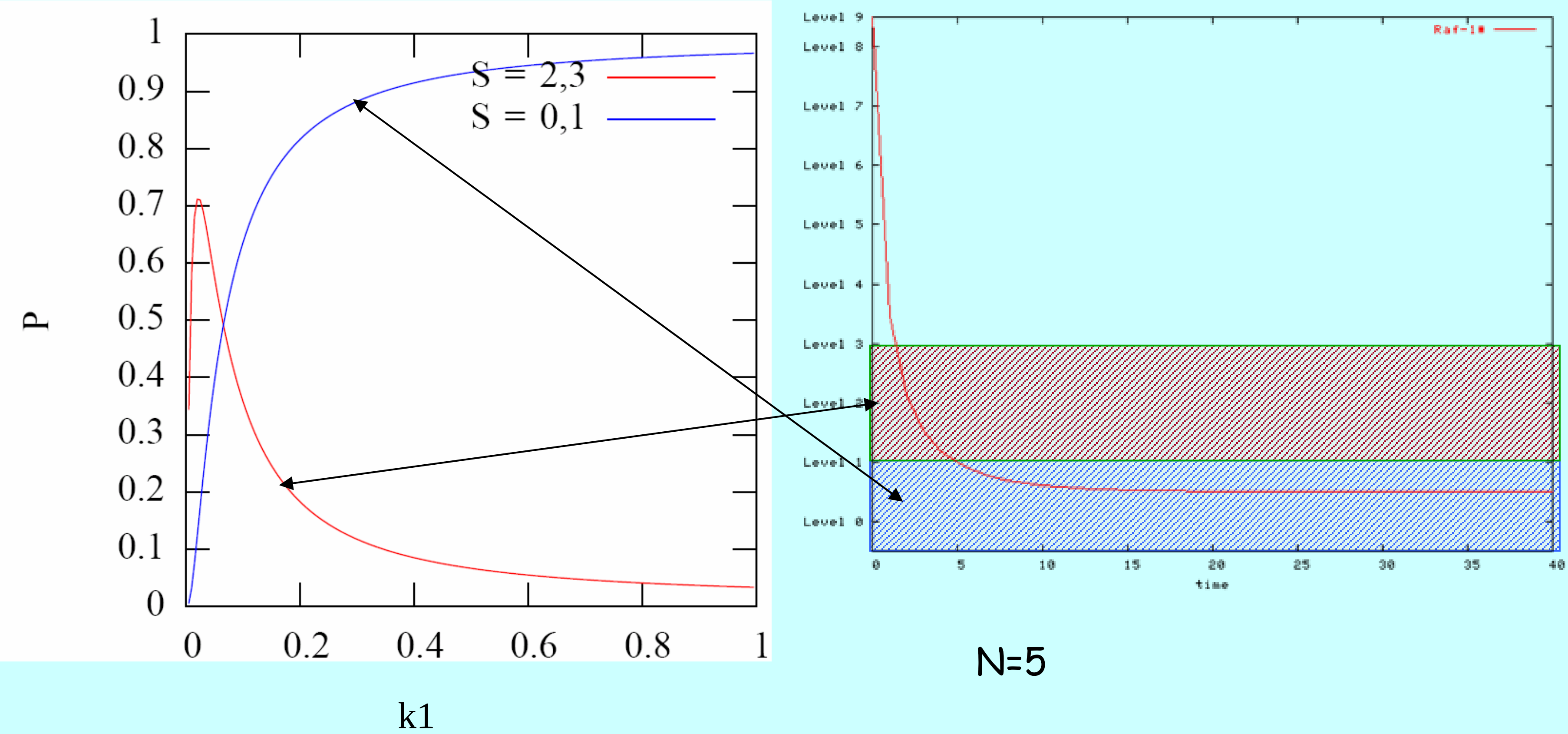
Probability that

RAF1/RKIP/ERK-PP reaches 5 before

RAF1/RKIP reaches 2 > 99%

Property 4:
protein activation order (transient)
P=?
[(RAF1/RKIP/ERK-PP < M) U (RAF1/RKIP = C)]

3. Stability when varying rate



When the binding rate of RKIP and RAF1 (k1) is increased, the probability that RAF1 stabilises on levels 2 or 3 (red square) falls, and the probability that RAF1 stabilises on levels 0 or 1 (blue square) rises.

Much more – allows us to conclude RKIP dampens down the ERK pathway.

Soundness

why is rate for r1 sound?

(RAF1*R)*(RKIP*R) *(k1*N)

relate discrete and continuous variables: m = m*R=m/N

Recall $$\frac{dm_3}{dt} = k_1 m_1 m_2$$

m_3'= m_3 +(k1* m_1 * m_2 * Δt). But abstract levels increase by 1, so

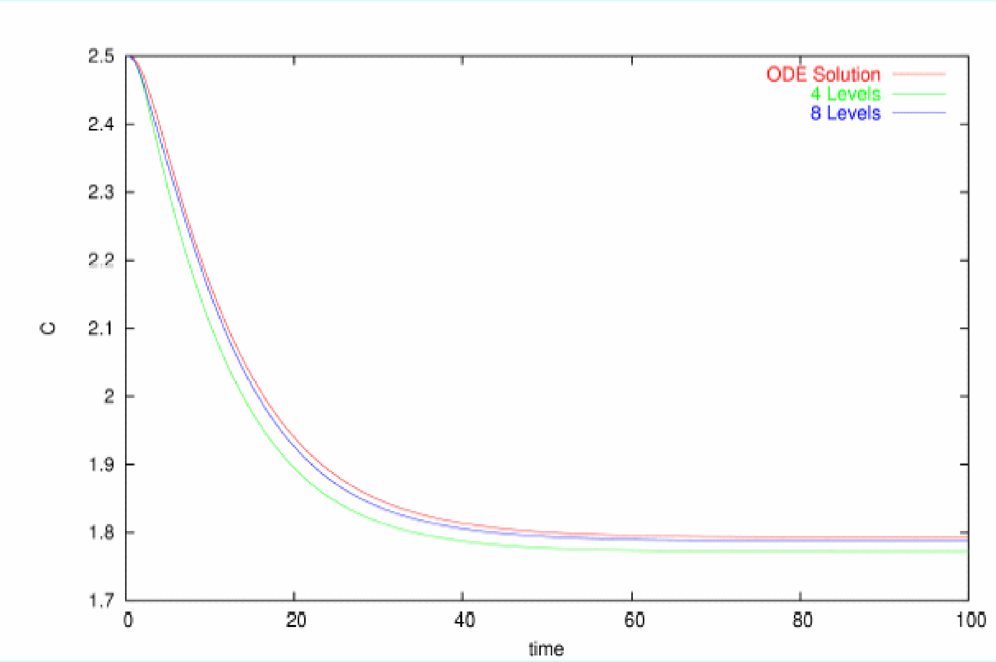
Δt = 1/(k1* m_1 * m_2* N)

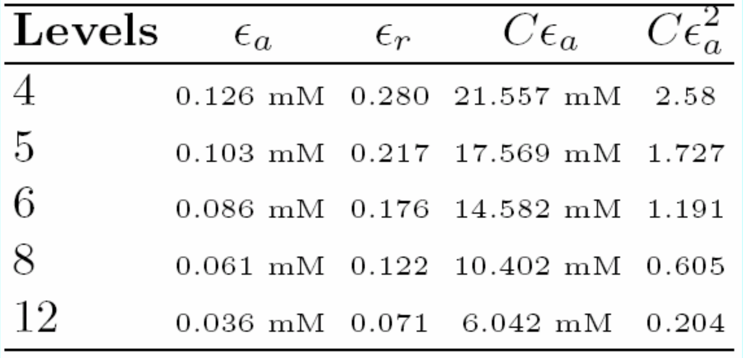
Let λ = k* m_1 * m_2* N
= k* (m_1*R) (m_2*R) *N
= k1* (RAF1*R)*(RKIP*R) *N

(We have, in effect, encoded Euler's method of numerical integration!)

Simulation

With simple formulae (*rewards*) it is possible to reproduce a simulation trace of the system. Compare with ODE model:

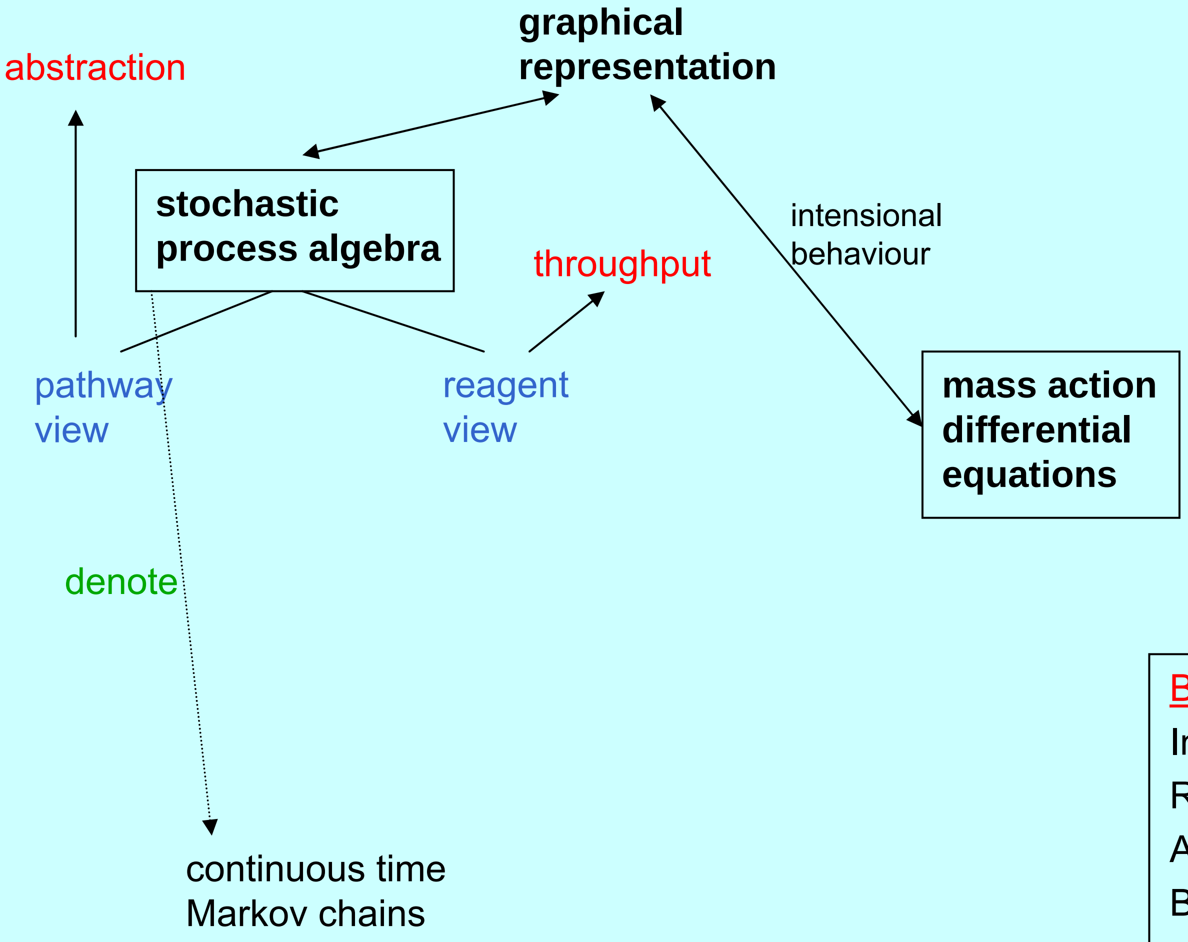




Levels	ϵ_a	ϵ_r	$C\epsilon_a$	$C\epsilon_a^2$
4	0.126 mM	0.280	21.557 mM	2.58
5	0.103 mM	0.217	17.569 mM	1.727
6	0.086 mM	0.176	14.582 mM	1.191
8	0.061 mM	0.122	10.402 mM	0.605
12	0.036 mM	0.071	6.042 mM	0.204

7-9 levels are sufficient! Extremely tractable!

Big Picture



Benefits

Interactions

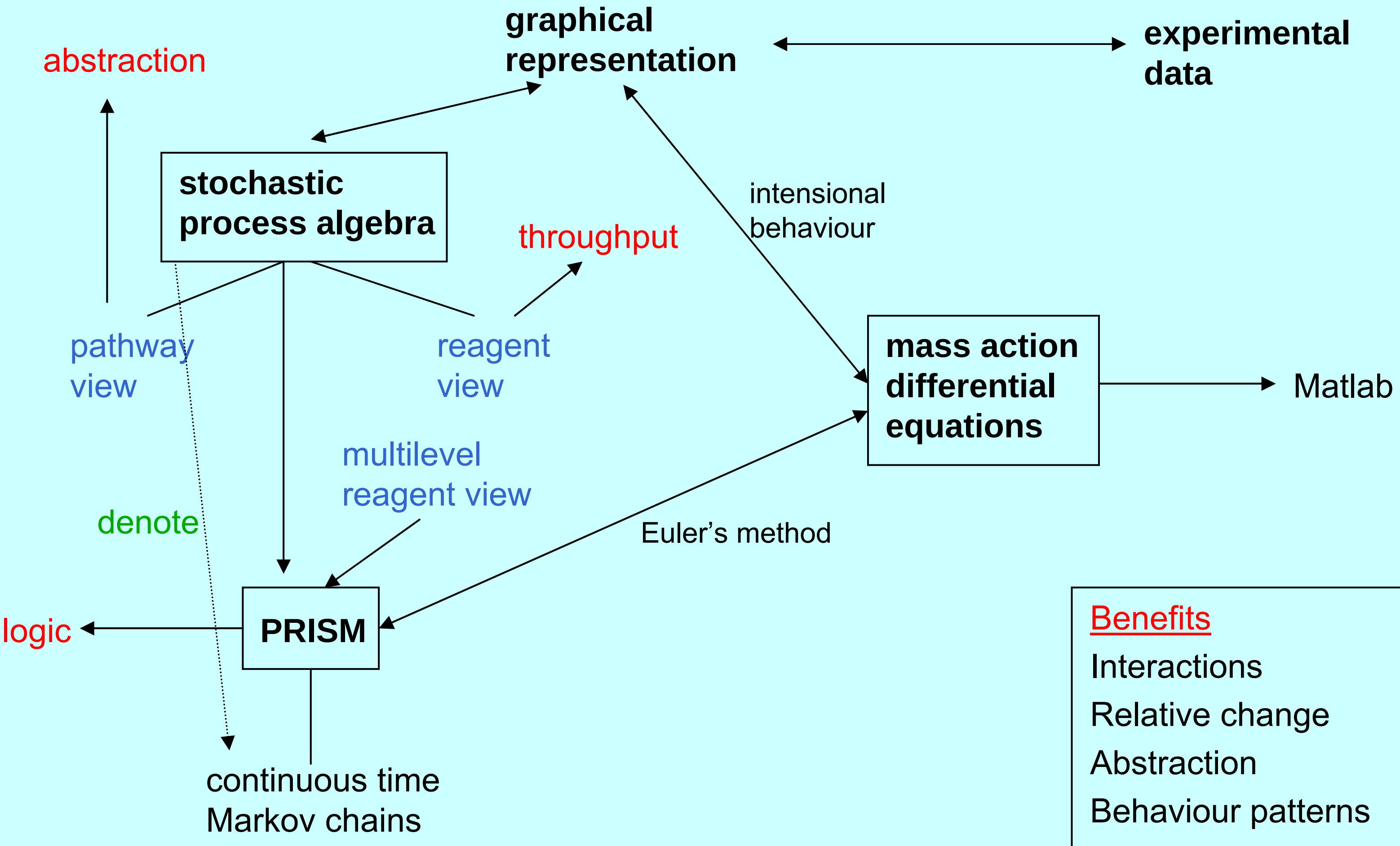
Relative change

Abstraction

Behaviour patterns

Applicability to other dynamics

Bigger Picture



graphical representation

experimental data

abstraction

stochastic process algebra

throughput

intensional behaviour

pathway view

reagent view

mass action differential equations

Matlab

multilevel reagent view

denote

Euler's method

logic

PRISM

continuous time Markov chains

<u>Benefits</u>

Interactions

Relative change

Abstraction

Behaviour patterns

Applicability to other dynamics

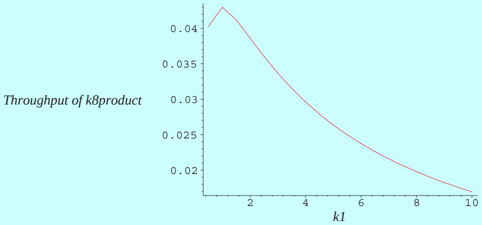
Modelling

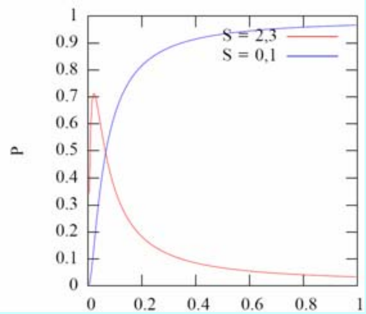
Each process molecular *species* is modelled by a stochastic, concurrent process

Expressed using high level languages from computer science!

⇒ new kinds of analysis
new ways of relating traditional and non-standard models

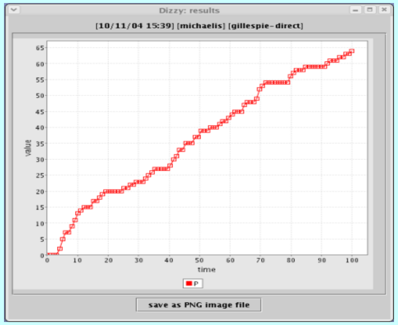
Process algebra models (reagent and pathway)





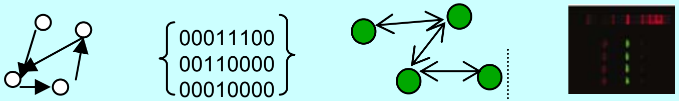
S=?[(Raf-1*≥2) /\ Raf-1*≤3]

$$\frac{dm_3}{dt} = k_1 m_1 m_2$$



The really big picture

topology & parameters + stochiometry + traffic + data



rates

stochastic process algebra

Raf-1*$_H$ = (k1react,k1). Raf-1*$_L$ + …

stochastic simulation
Dizzy

approximation

Approximation
model-fitting
(Bayesian inference)

abstraction

throughput

pathway view

reagent view

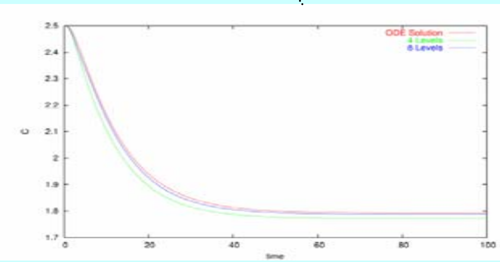
Continuous time Markov chains

N levels

solve

mass action differential equations

$$\frac{dm1}{dt} = - k1*m1*m2 + k2*m3$$



simulation

approximation

continuous stochastic logic

PRISM

probabilities

(other dynamics)

***continuous* time**
***discrete* concentrations**

***continuous* time**
***continuous* concentrations**

***discrete* time**
***continuous* concentrations**

Discussion & Conclusions

- Regent-centric view
  - probabilities of states (H/L)
  - differential equations
  - fit with data
  - PEPA -> PRISM
- Pathway-centric view
  - simpler model
  - building blocks, modularity approach
  - no further information is gained from having multiple levels

- Life science
  - see potential of an *interaction* approach which affords two views; static analysis and modelling language

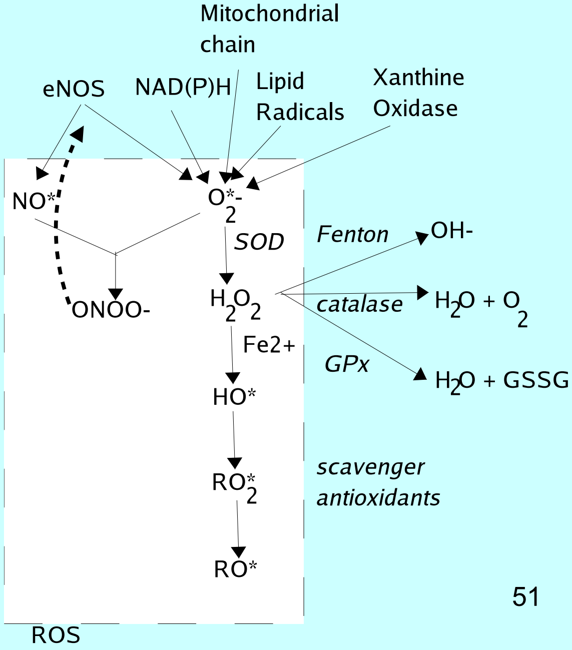
- Computing science
  - individual/population view
  - continuous, traditional mathematics and simulation

- theorem: as n->∞, ODE and CTMC semantics converge (1st, 2nd order eqns)

Further Challenges

- Quantification of abstraction over networks
  - "chop" off bits of network
- Model spatial dynamics (vesicles) and scaffolds
- Model other stochiometries
- Relate individual to molar concentration models
- Prove CTMC and ODE equivalence
- Relate data to high level operators (topology)

- Further applications: e.g. oxidative stress
  cardiovascular medicine



The End

Thank you.