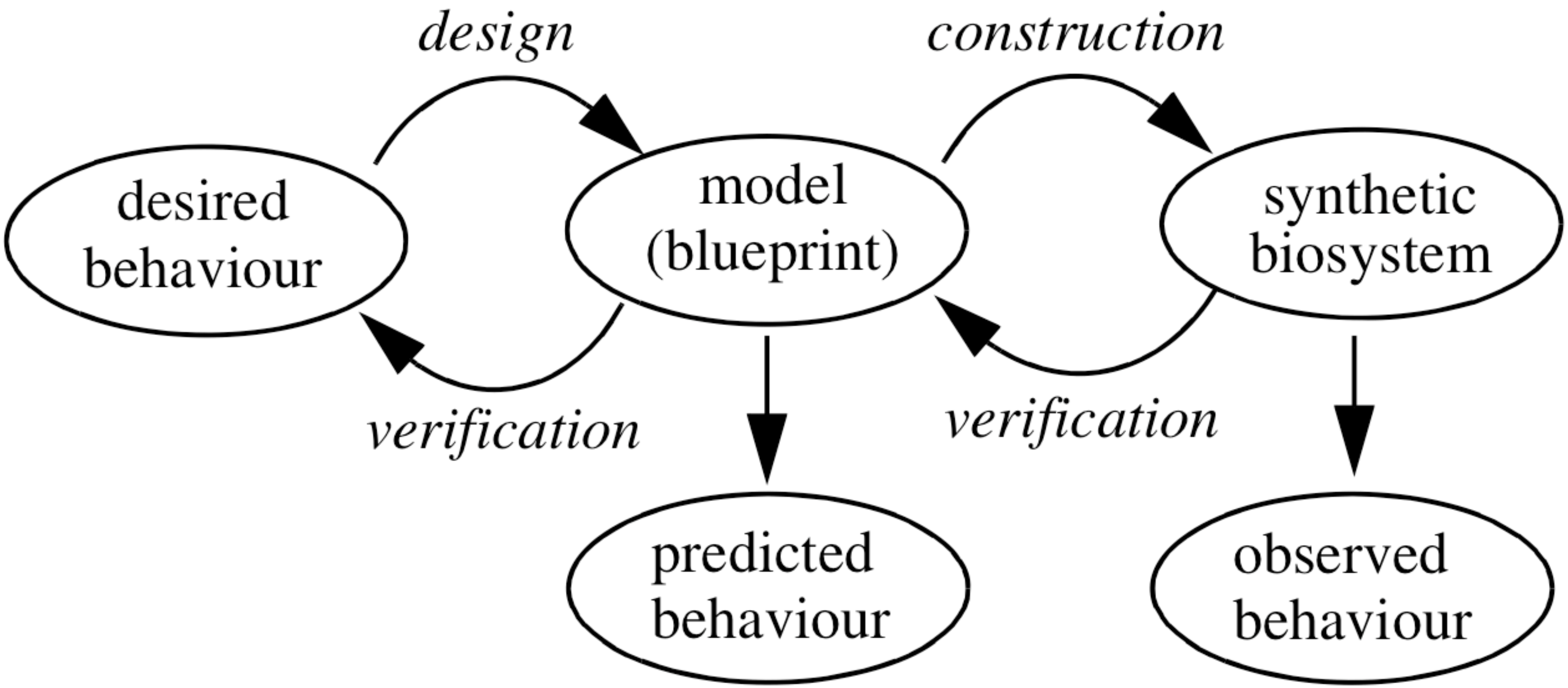
Model validation and verification

David Gilbert

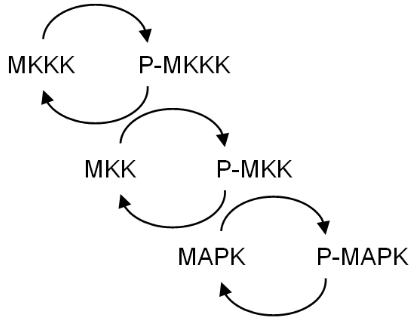
Systems Biology Research Group
School of Information Science Computing and Mathematics
Brunel University, UK

Summary

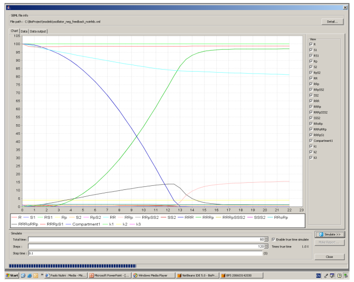
- In this tutorial we describe some uses of model checking for Synthetic Biology.
- Many of these ideas originated in the context of research in Systems Biology.
- Special thanks to Robin Donaldson, University of Glasgow, who developed the MC2 model checker (www.brc.dcs.gla.ac.uk/software/mc2/) and created the MC2 model checking slides in this presentation.


design
construction
desired behaviour
model (blueprint)
synthetic biosystem
verification
verification
predicted behaviour
observed behaviour

Model

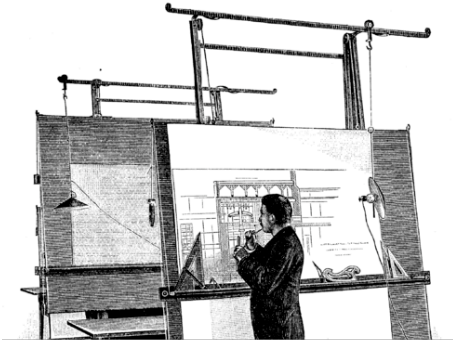


- A model
  - formal representation of the real world
  - simplified abstract view of the complex reality.



- A simulation: implementation of a model over time.

- To design: the process of originating and developing a plan for a product



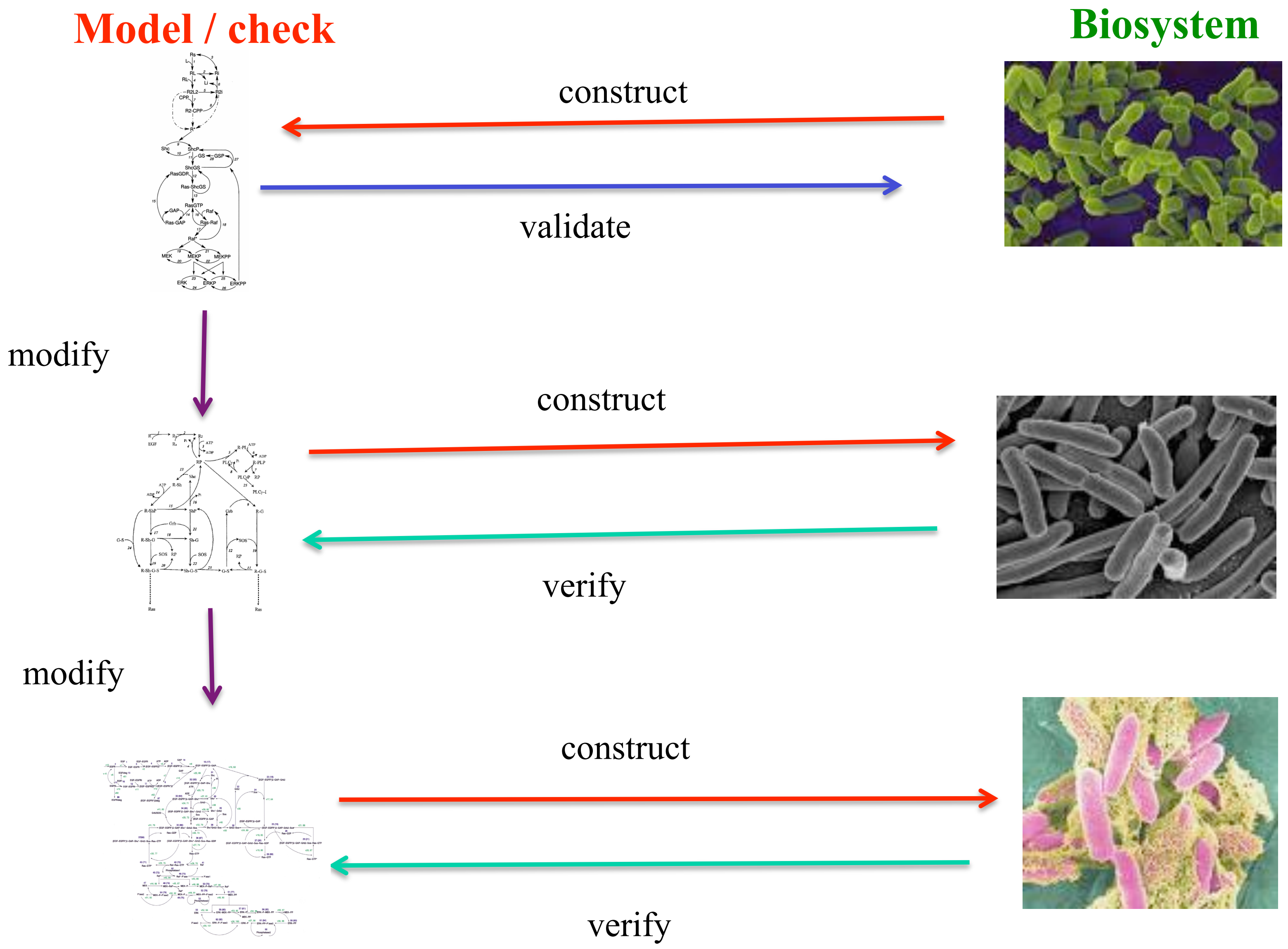
- A design: (Final) plan, e.g. model, description, for the product

Validation & verification

- **Validation** – '***You built the right product?***'.
  - Product / system accomplishes its intended requirements.
  - Model / simulation are accurate representations of the real world

- **Verification** - '***You built the product right?***'.
  - System complies with its specification
  - Model / simulation accurately represent the specifications

Synthetic biology development cycle



Model / check

Biosystem

construct

validate

modify

construct

verify

modify

construct

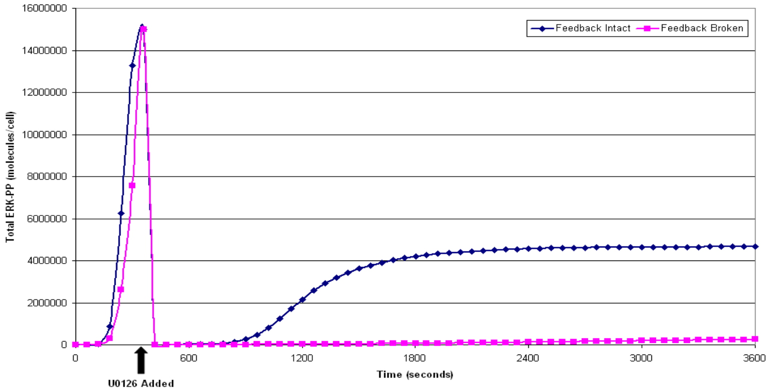
verify

Describing experimental data

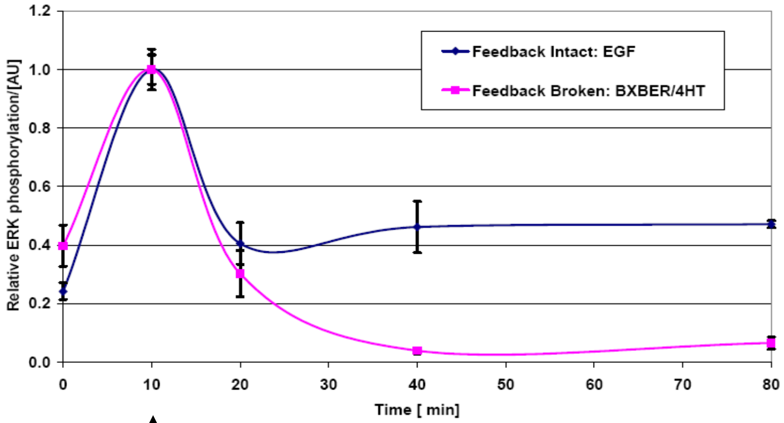
- Biologists will often talk in qualitative or semi-quantitative language (trends).
  - “this protein peaks after 5 minutes, then falls to half concentration”
  - Often quite certain about time,
  - But not about concentrations

Lab data versus simulations

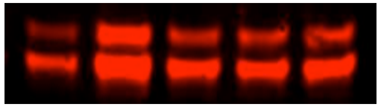
Simulation



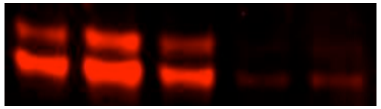
Experiment



U0126 added



pERK1/2, +EGF



pERK1/2, + BXBER/4HT

0 10 20 40 80 min stimulation

David Gilbert ECSBII 2009

Properties…

Examples:

- After 100 seconds the concentration of Protein1 is stable
- Protein1 peaks and falls
- Protein1 peaks and stays constant
- Protein1 peaks before Protein2
- Protein1 oscillates 4 times in 5,000 seconds
- Molecules of Protein2 are required for molecules of Protein1 to be created

Model Checking

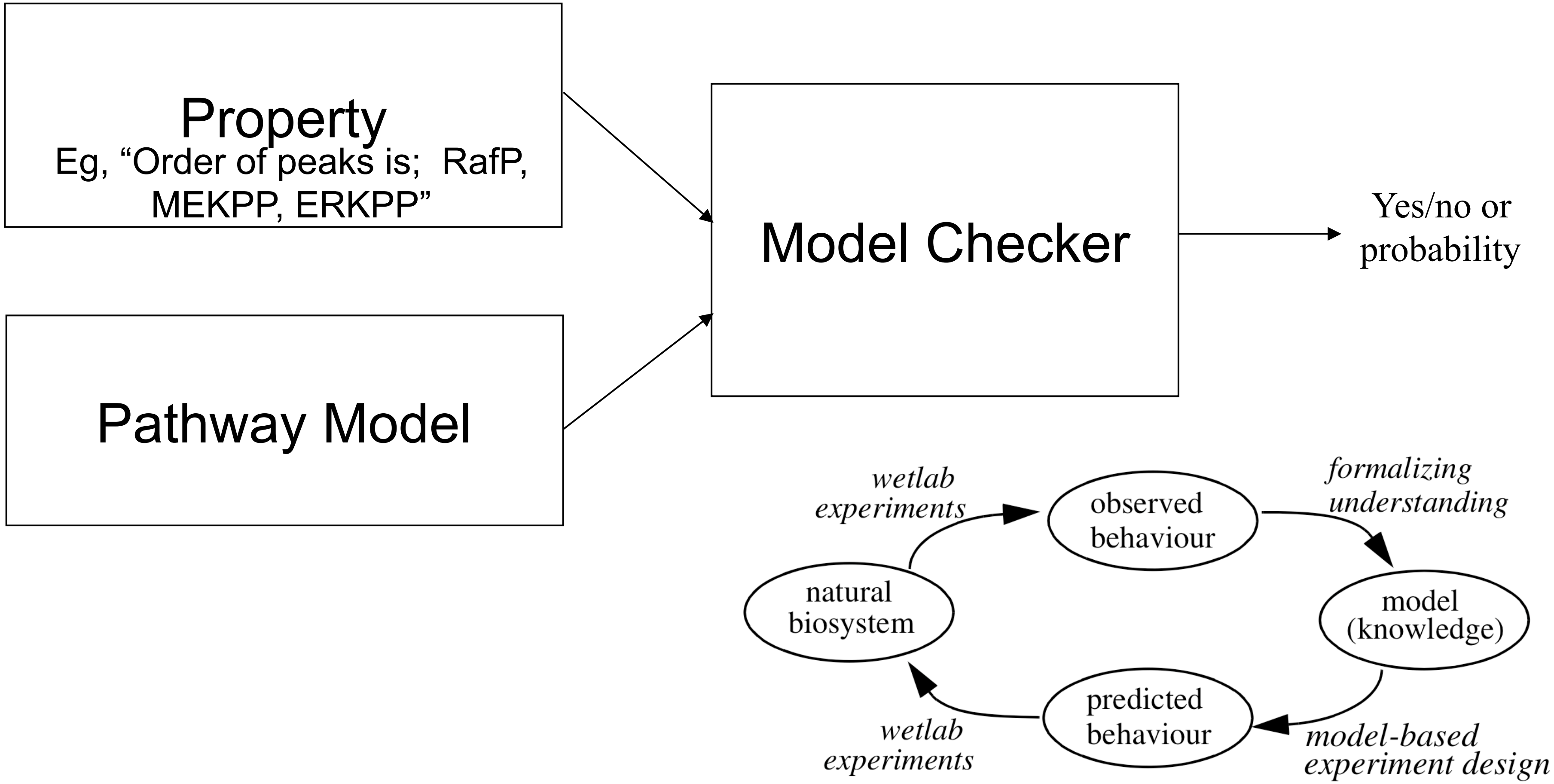
In a sentence:

- “Formally check whether a model of a biochemical system does what we want”

Components:

- A model
  - the current description of a biochemical system of interest
- A property
  - a property which we think the system should have
- A model checker
  - a program to test whether the model has the property

Model Checking
Biochemical Pathways



Property
Eg, “Order of peaks is; RafP, MEKPP, ERKPP”

Pathway Model

Model Checker

Yes/no or probability

What can we do with model checking?

- **Model validation**:
  - Show that your model of the pathway matches the lab data
  - Might have a high probability of doing what you want, but doesn't always do it!
- **Model analysis**:
  - In a collection of variants of a model (e.g., *in silico* gene knock-outs), which models show a certain behavior? (loss of oscillations…)
- **Model development**:
  - If the model doesn't do what we want, change the model automatically until it does! (parameters, structures,…)
- **Model finding**:
  - Many models in a database, can use model checking to query the database
  - "Give me all the models in the database which oscillate"
- **Biosystem verification**:
  - Does the constructed system do what we intended?

To formally express time properties we use a temporal logic

"*I am hungry*.”

"*I am always hungry*", "*I will eventually be hungry*",

"I will be hungry until I eat something”.

Linear time logics restricted to single time line.

Branching logics can reason about multiple time lines.

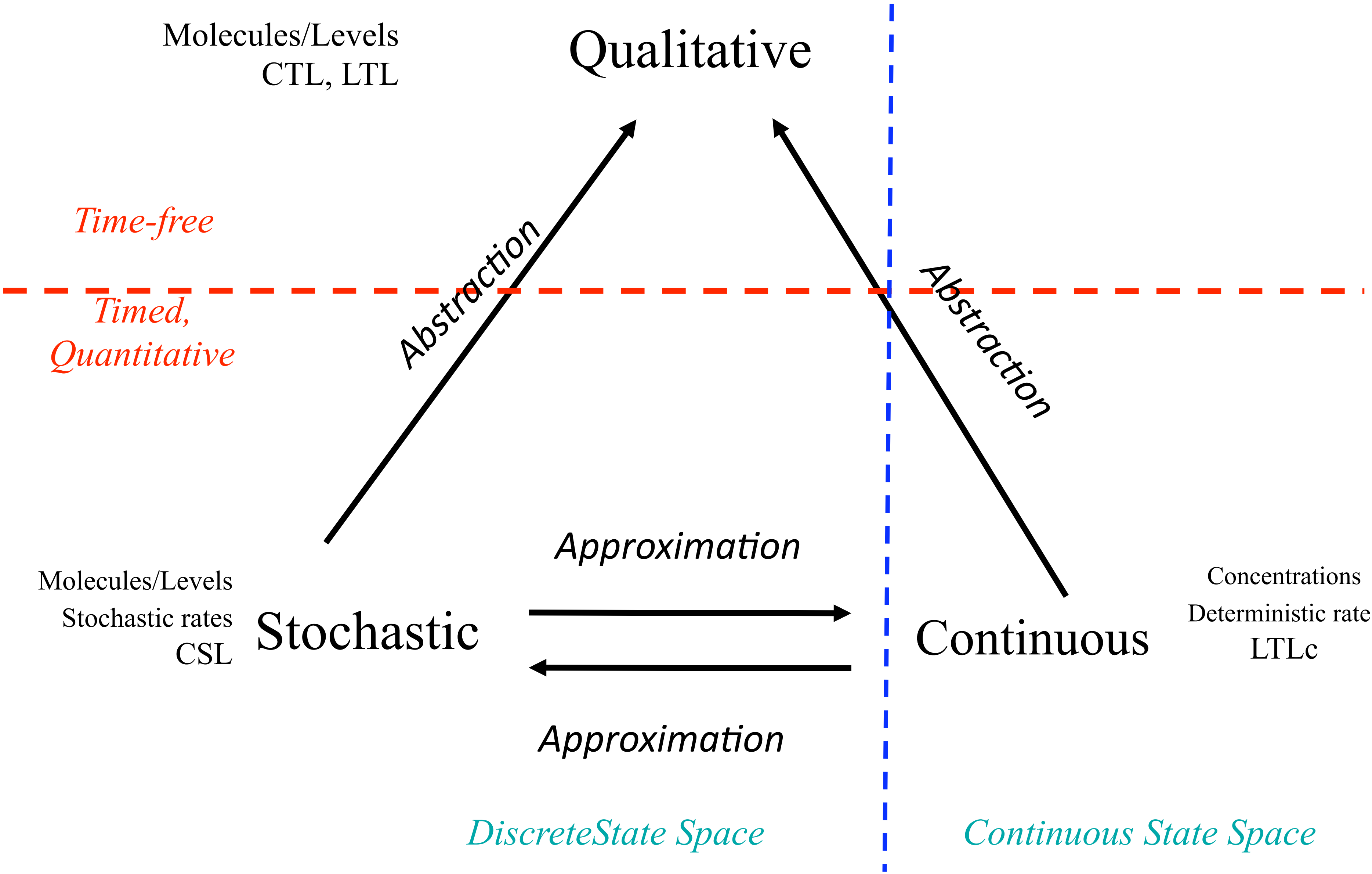
“*There is a possibility that I will stay hungry forever*.”

“*There is a possibility that eventually I am no longer hungry*.”

Various logics :

- Computational Tree Logic (CTL)
- Continuous Stochastic Logic (CSL)
- Linear-time Temporal Logic (LTL)

each with different expressivity.



Molecules/Levels
CTL, LTL

Qualitative

Time-free

Timed,
Quantitative

Abstraction

Abstraction

Approximation

Molecules/Levels
Stochastic rates
CSL

Stochastic

Continuous

Concentrations
Deterministic rates
LTLc

Approximation

DiscreteState Space

Continuous State Space

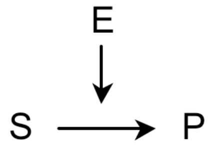
Gilbert, Heiner and Lehrack. ``A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets." Proc CMSB 2007

Models

- Qualitative: Petri nets,…
- Continuous: ODEs, Continuous Petri nets
- Stochastic: Stochastic process algebras, Stochastic Petri nets, Chemical master equations, P-systems, (Gillespie)…

Enzymatic reaction models

A



B

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} E + P$$

C

$$\frac{d[P]}{dt} = \frac{k_{cat}[E][S]}{K_m + [S]}$$

$$K_m = \frac{k_1 + k_2}{k_3}$$

$$k_{cat} = k_3$$

D

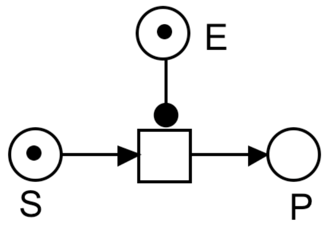
$$\frac{d[S]}{dt} = -k_1[S][E] + k_2[ES]$$

$$\frac{d[ES]}{dt} = k_1[S][E] - k_2[ES] - k_3[ES]$$

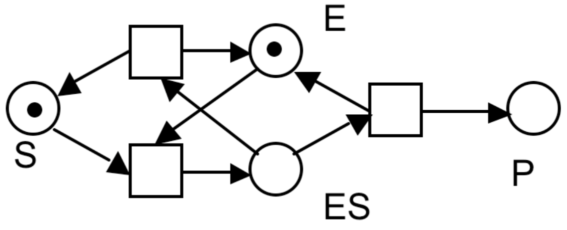
$$\frac{d[E]}{dt} = -k_1[S][E] + k_2[ES] + k_3[ES]$$

$$\frac{d[P]}{dt} = k_3[ES]$$

E



F



G. PEPA

$$E_H = (r_1, k_1).E_L$$

$$E_L = (r_2, k_2).E_H + (r_3, 1).E_H$$

$$S_H = (r_1, k_1).S_L$$

$$S_L = (r_2, k_2).S_H$$

$$SE_H = (r_2, k_2).SE_L + (r_3, k_3).SE_L$$

$$SE_L = (r_1, k_1).SE_H$$

$$P_L = (r_3, k_3).P_H$$

$$P_H = (\text{stop}, 1).P_H$$

$$S_H \underset{\{r_1, r_2\}}{\bowtie} (SE_L \underset{\{r_1, r_2, r_3\}}{\bowtie} E_H) \underset{\{r_3\}}{\bowtie} P_L$$

H. Stochastic π-calculus

$$E(k_1) \triangleq \nu k_2 \nu k_3 ! k_1(k_2, k_3).(? k_2.E(k_1) + ? k_3.E(k_1))$$

$$S(k_1) \triangleq ? k_1(k_2, k_3).(! k_2.K(k_1) + ! k_3.P())$$

$$\text{run100of}\, E(a) \mid S(a)$$

A,B: Conventional notation of the chemical reactions and kinetic constants. **C,D**: ODE. **E,F**: Discrete Petri net description. **G:** Stochastic process algebra PEPA. **H:** Stochastic π-calculus

David Gilbert ECSBII 2009

PLTL language

- Probabilistic logic called Probabilistic Linear-time Temporal Logic (PLTL)

- Main PLTL operators:

G (P) : P always happens

F (P) : P happens at some time

X (P) : P happens in the next time point

(P1) U (P2) : P1 happens until P2 happens

P1 { P2 } : P1 happens from the first time P2 happens

Qualitative to quantitative descriptions in PLTL

- **Qualitative**:
  Protein rises then falls
  $P=?\ [\ (\ d(\text{Protein}) > 0\ )\ U\ (\ G(\ d(\text{Protein}) < 0\ )\ )\ ]$

- **Semi-qualitative**:
  Protein rises then falls to less than 50% of peak concentration
  $P=?\ [\ (\ d(\text{Protein}) > 0\ )\ U\ (\ G(\ d(\text{Protein}) < 0\ )\ \wedge\ F\ (\ [\text{Protein}] < 0.5 \ast \max[\text{Protein}]\ )\ )\ ]$

- **Semi-quantitative**:
  Protein rises then falls to less than 50% of peak concentration by 60 minutes
  $P=?\ [\ (\ d(\text{Protein}) > 0\ )\ U\ (\ G(\ d(\text{Protein}) < 0\ )\ \wedge\ F\ (\ \text{time} = 60\ \wedge\ \text{Protein} < 0.5 \ast \max(\text{Protein})\ )\ )\ ]$

- **Quantitative**:
  Protein rises then falls to less than 100μMol by 60 minutes
  $P=?\ [\ (\ d(\text{Protein}) > 0\ )\ U\ (\ G(\ d(\text{Protein}) < 0\ )\ \wedge\ F\ (\ \text{time} = 60\ \wedge\ \text{Protein} < 100\ )\ )\ ]$

PRISM

Probablistic model checker

- Specification language: PRISM language (based on reactive modules language)
- Supports 3 types of models
  - Discrete time Markov chains (DTMCs)
  - Continuous time Markov chains (CTMCs)
  - Markov decision processes (MDPs)
- Logic: PCTL or CSL
- Symbolic
- Communication via shared events
- Synchronous execution (apart from MDPs)

Quantitative analysis using costs/rewards
Can run experiments

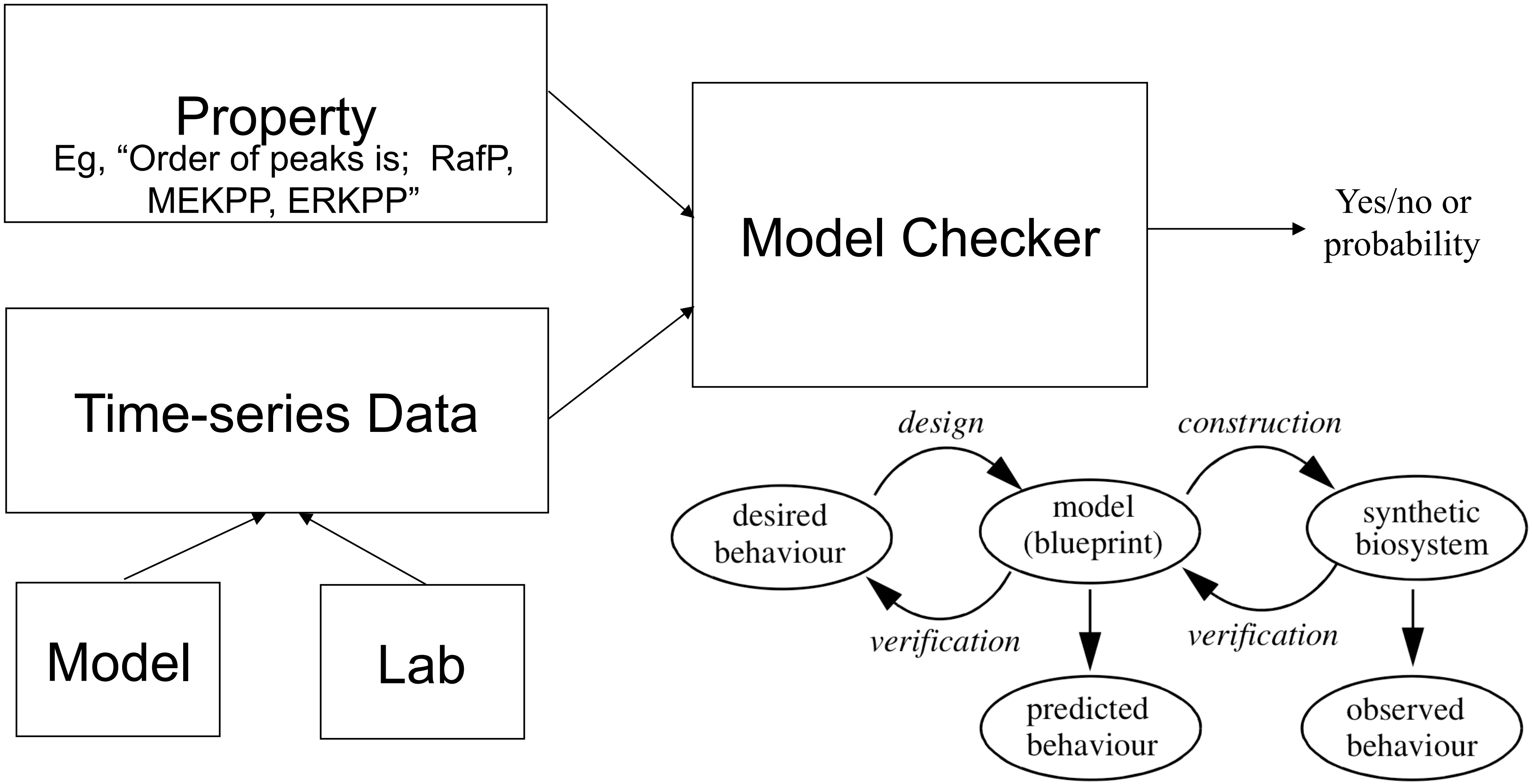
Has been used to verify: signalling pathways in systems biology, PIN block attacks, communications protocols (e.g. bluetooth, SMAC), aviation security procedures

Analytical vs Simulative Model Checking

- **Analytical**:
  - Exact probabilities & prove properties
  - A model state is an association of #molecules to each of the species
    - **Protein1** has 10 molecules & **Protein2** has 20 molecules
  - Analytical assesses every state that the model can be in (reachable states)
  - State space can grow even worse than exponentially with increasing molecules
  - Stochastic model checking with even as little as 12 molecules can be impossible with today's technology

- **Simulative**:
  - Instead of analysing the constructed state space, analyse simulation outputs
  - Simulate the model X times and check these simulations
  - Simulation run = finite path through the state space
  - Can't prove probabilities

Simulation-based Model Checking Biochemical Pathways



Property

Eg, “Order of peaks is; RafP, MEKPP, ERKPP”

Time-series Data

Model

Lab

Model Checker

Yes/no or probability

design

construction

desired behaviour

model (blueprint)

synthetic biosystem

verification

verification

predicted behaviour

observed behaviour

Simulative model checking

- **In-line**: check the observations as they arrive
  - Requires complex computational machinery: ‘combine’ simulator & model checker
  - Good for biochemical observations
  - Don’t always need to finish the experimental run

- **Off-line**: check the observations after all have been generated
  - Easier to implement computationally (simulate then check)
  - Need to always define when to ‘stop’ generating observations

Monte Carlo Model Checker

- Developed an ***offline*** Monte Carlo Model Checker for PLTLc properties, MC2(PLTLc) for short.
- Operates on a finite set of simulations – simulative approach
- Typically, many stochastic simulations to approximate probabilities
  - Approximate probability = fraction of simulations which satisfy the property over the #simulations
- Monte Carlo approximation – 2 approximations made:
  - finite number of simulations
  - Simulations of finite length

Monte Carlo Model Checker

- The set of time series data can be:
  - Set of stochastic runs
  - A single continuous run
  - A parameter scan
  - Lab data!

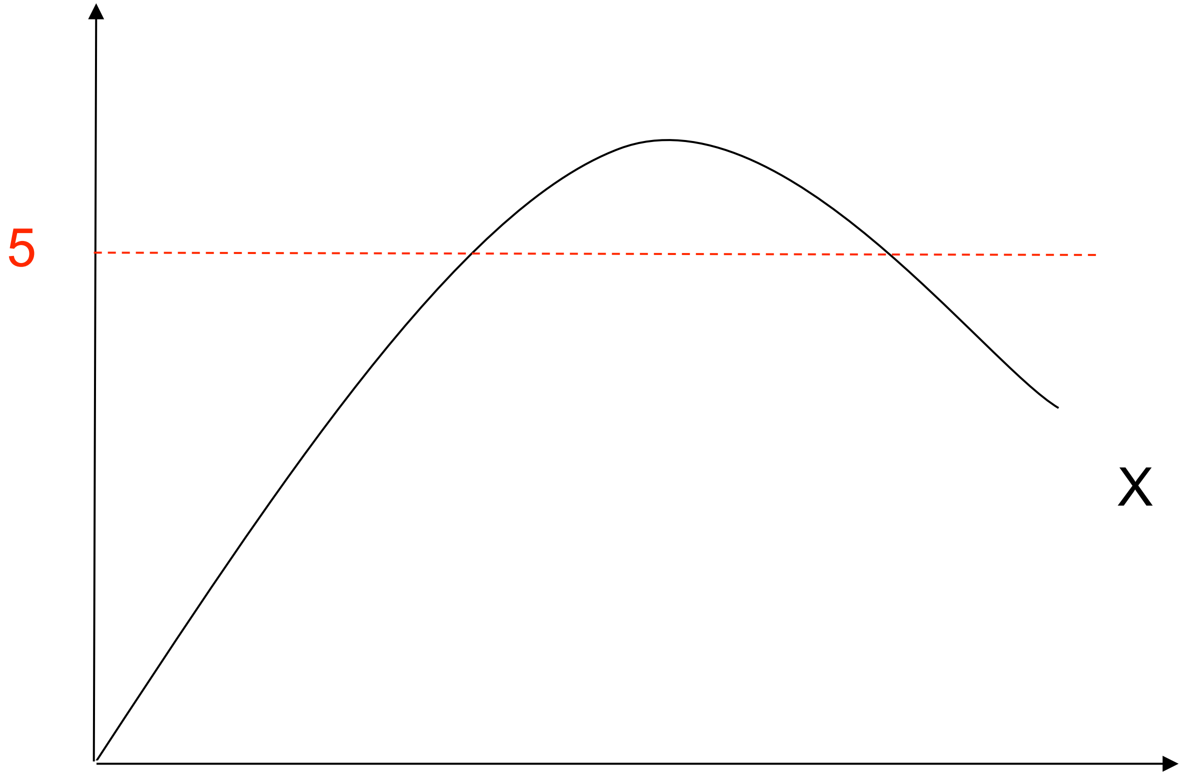
- We could use simulation output from;
  - ODE, SDE, CTMC, Gillespie, hybrid approaches, multi-cellular simulation, open models

- Or experimental data from the wet lab

MC2 with ODE Output

$P_{=?}[\ F(\ X > 5\ )\ ]$

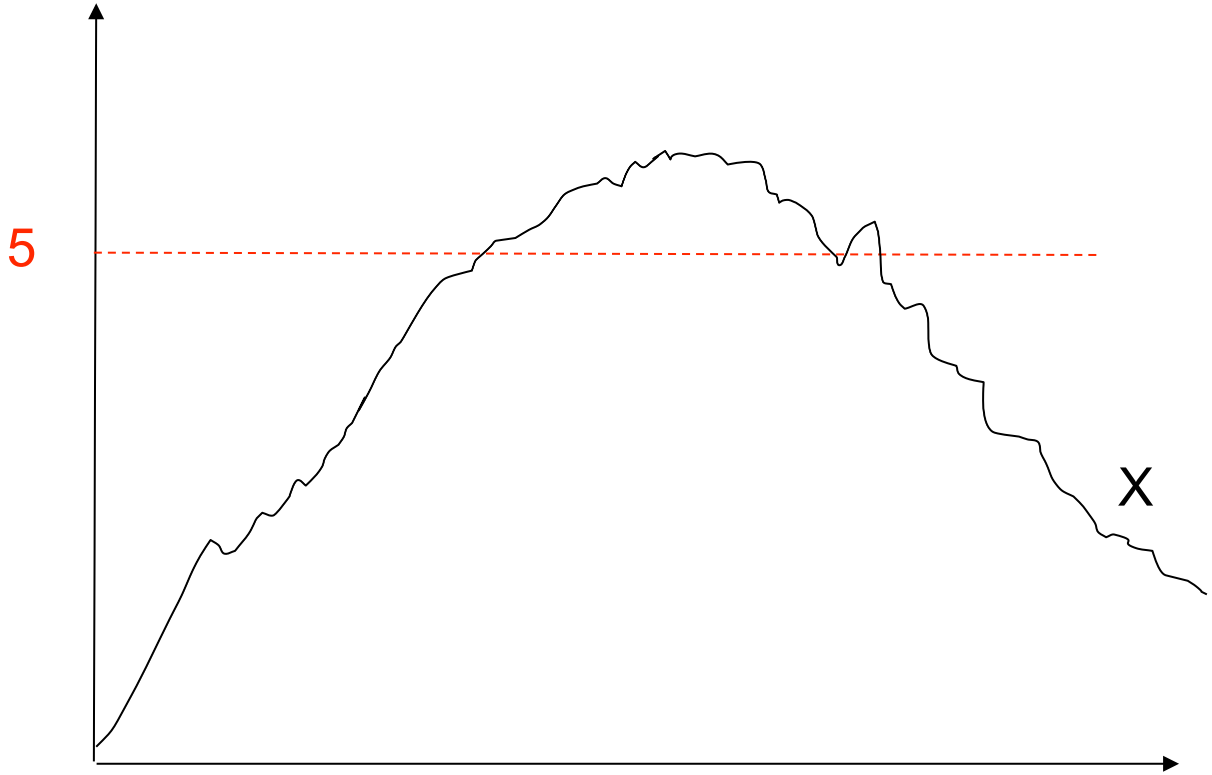
=> P = 1



MC2 with Gillespie Output

$P_{=?}[\ F(\ X > 5\ )\ ]$

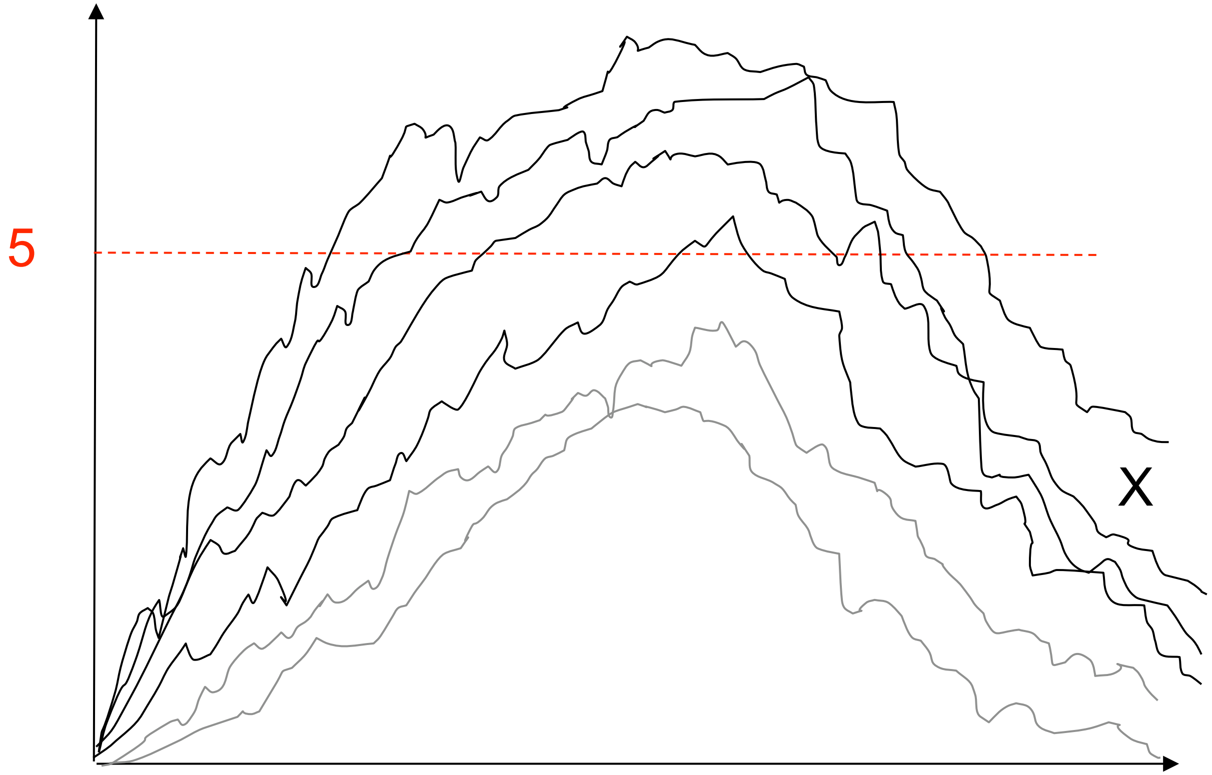
=> P = 1



MC2 with Gillespie Output

$P_{=?}[\ F(\ X > 5\ )\ ]$

=> P = 4/6



Why MC2?

- MC2
  - Probabilities are derived from fraction of #finite simulation outputs satisfying property
  - Provides an approximation of the probability because
    - Finite set of simulation outputs and simulation outputs are finite size

- PRISM
  - Exact probabilities are produced but much higher cost
  - Exact, need to construct the state-space
  - State-space is exponential in #levels and #protein types.
  - Gilbert *et al.* (2007) can only use up to 8 concentration levels in PRISM – with MC2 we can use easily 4,000.

Gilbert *et al*. (2007), “A unifying framework for modelling and analysing biochemical pathways using Petri nets”. In Proc. CMSB 2007, pages 200-216.

Why MC2?

- PLTL (MC2) is more theoretically correct than CSL (PRISM) for model checking on simulation outputs.

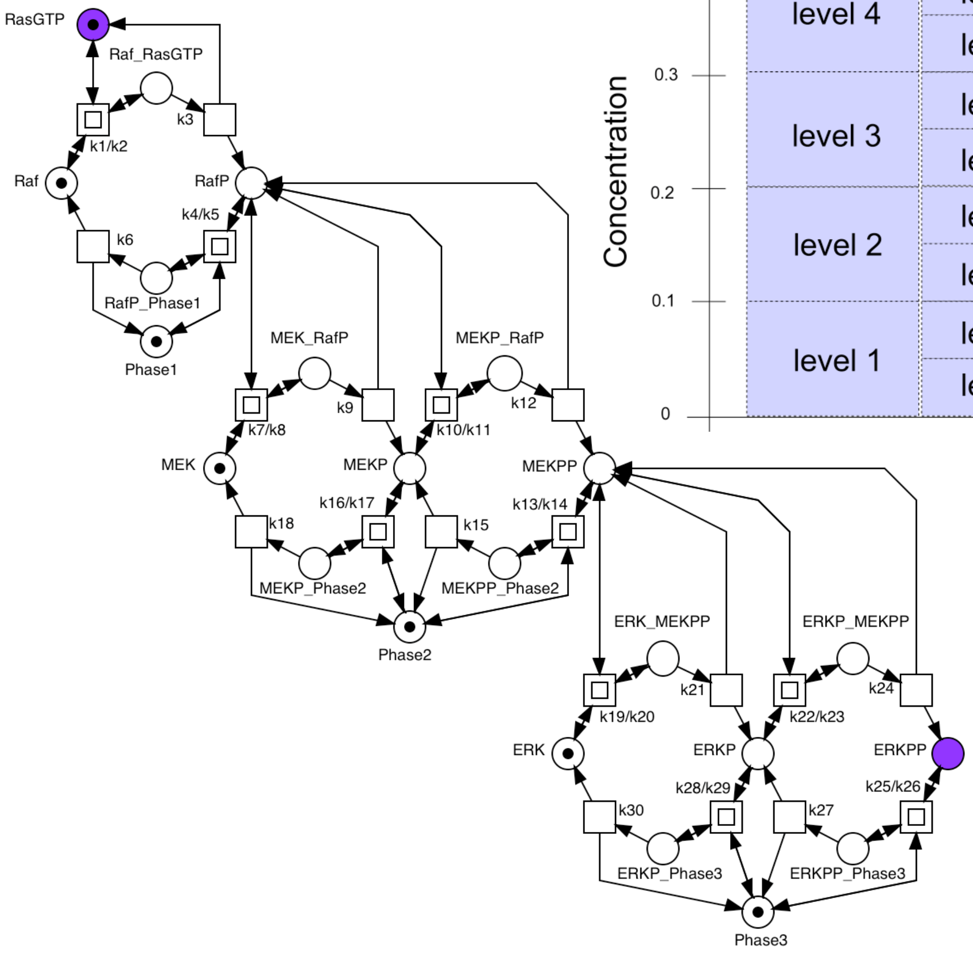
- No need to impose time-bounds on operators for efficiency as done in Gilbert *et al.* (2007):
- $P_{=?}$[(RafP = X) $U_{<=100}$ (RafP > X) { RafP = X }]

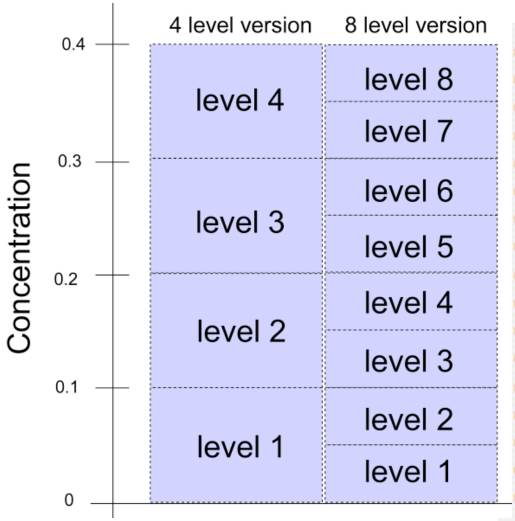
- We perform model checking on our cluster. Fast!

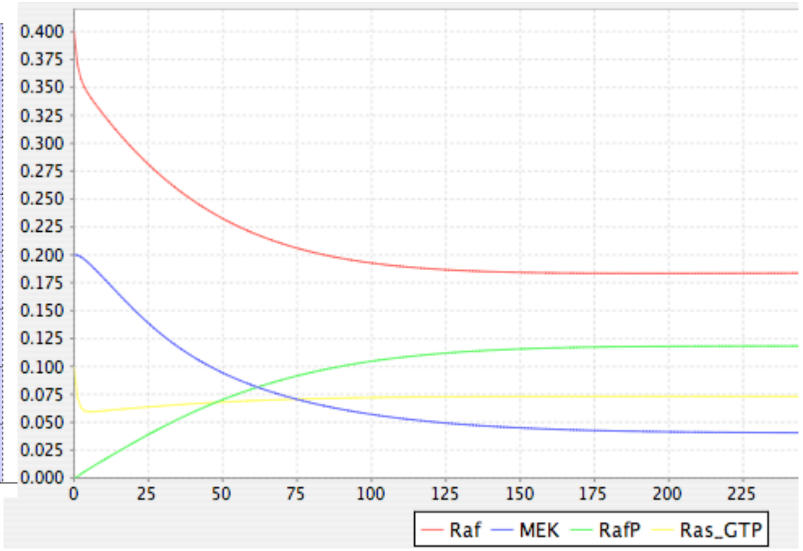
- Fast, but it is an approximation…

Gilbert *et al*. (2007), “A unifying framework for modelling and analysing biochemical pathways using Petri nets”. In Proc. CMSB 2007, pages 200-216.

Levchenko Model





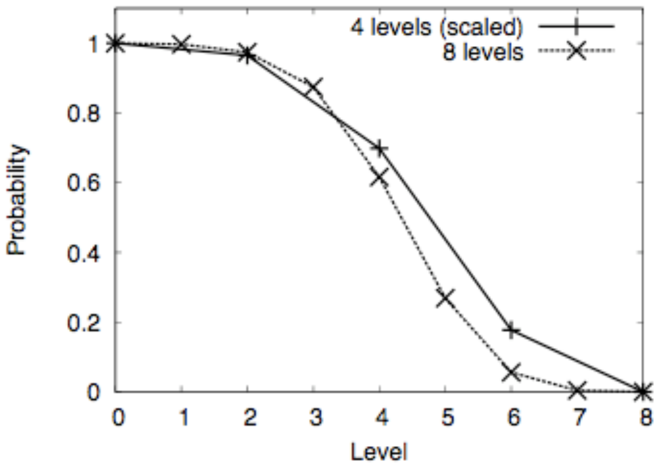


Stochastic Analysis

Check the property:

"*What is the probability that RafP will reach concentration X while MEKPP and ERKPP remain at 0?*"

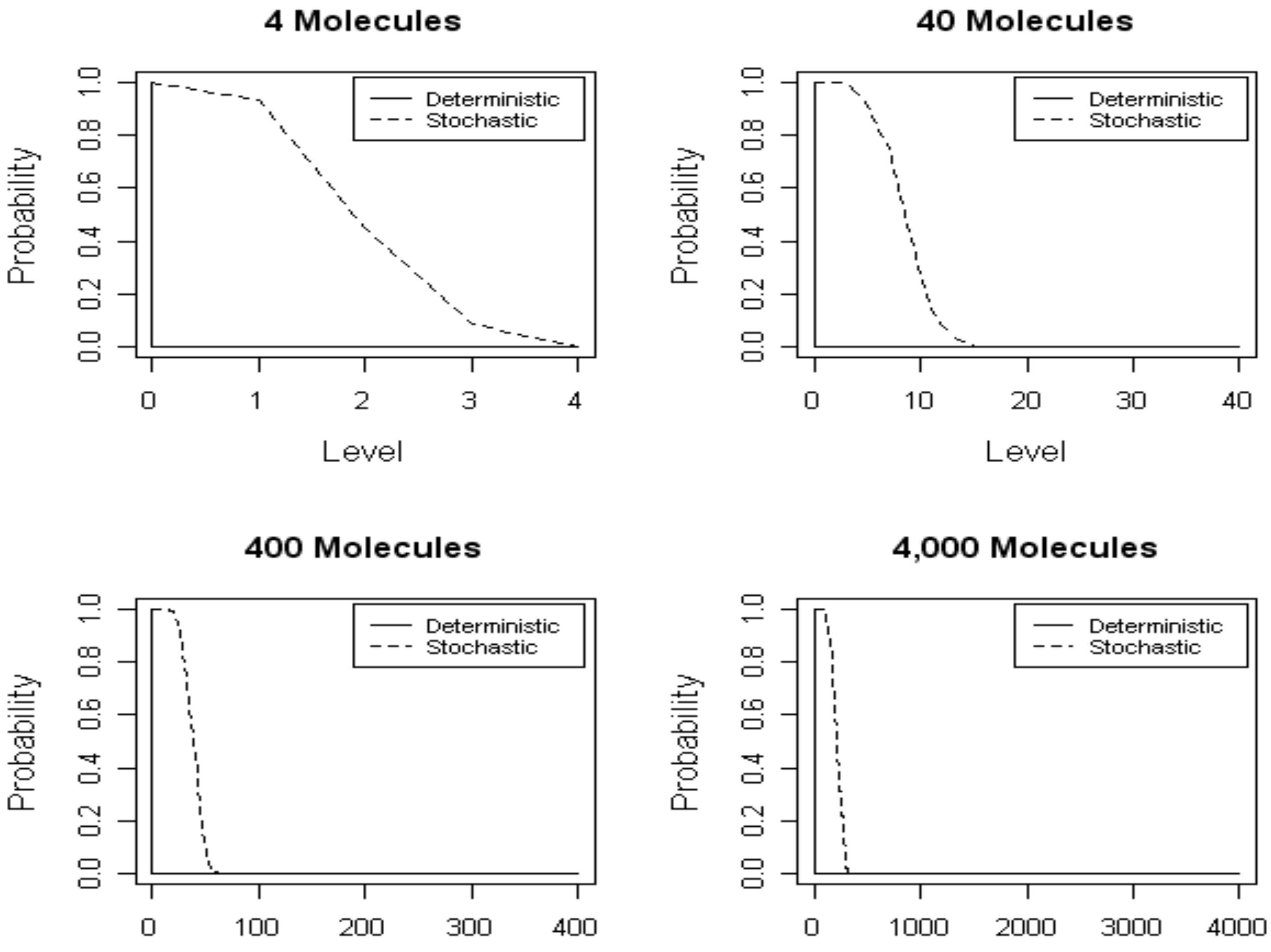
$$P_{=?}\ [\ (\text{MEKPP} = 0 \wedge \text{ERKPP} = 0)\ \text{U}\ (\text{RafP} > X)\ \{\text{RafP} = 0 \wedge \text{MEKPP} = 0 \wedge \text{ERKPP} = 0\}\ ]$$



4 LEVELS : 4 HOURS
8 LEVELS : 24 HOURS
16 LEVELS: ??

David Gilbert, Monika Heiner and Sebastian Lehrack (2007). A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets. Proc CMSB 2007

Checking the property at varying levels



Monika Heiner, David Gilbert, and Robin Donaldson (2008), Petri Nets for Systems and Synthetic Biology. In M Bernardo, P Degano, and G Zavattaro (Eds.): Formal Methods for Systems Biology SFM 2008, Springer LNCS 5016

PLTLc language, specifics

LTLc [Fages et al.] - extension of LTL with numerical constraints.

We extend Probabilistic LTL with numerical constraints : PLTLc.

E.g.

- Free variable $X always greater than the concentration of Protein.
  $P_{=?}$ [G($X > [Protein])]

Model checking of PLTLc properties returns:

- Probability of behaviour:
  some value is always greater than concentration of protein
- The domain of free variable $X for which the behaviour holds true

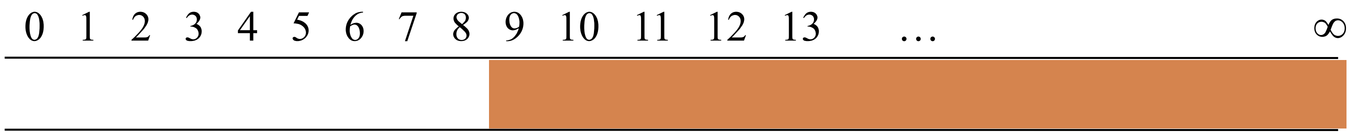
Fages et al. “On the Analysis of Numerical Data Time Series in Temporal Logic.” In *Proc. CMSB2007, pages48–63. LNCS/LNBI4695, Springer,2007.*

PLTLc language

- Continuous model with [Protein] between 0…8 in simulation output

- $P_{=?}$ [G($X > [Protein])]

- Probability =1 : there is a value always greater than [Protein]
- Domain of $X is 9…inf : these are the values always greater than [Protein] :



- Probabilistic language, interpret this as a probabilistic domain:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	…	∞
0	0	0	0	0	0	0	0	0	1	1	1	1	1		1

PLTLc language

- Stochastic model has varied behaviour, so probabilistic domain ranges 0…1.

- Could look something like:

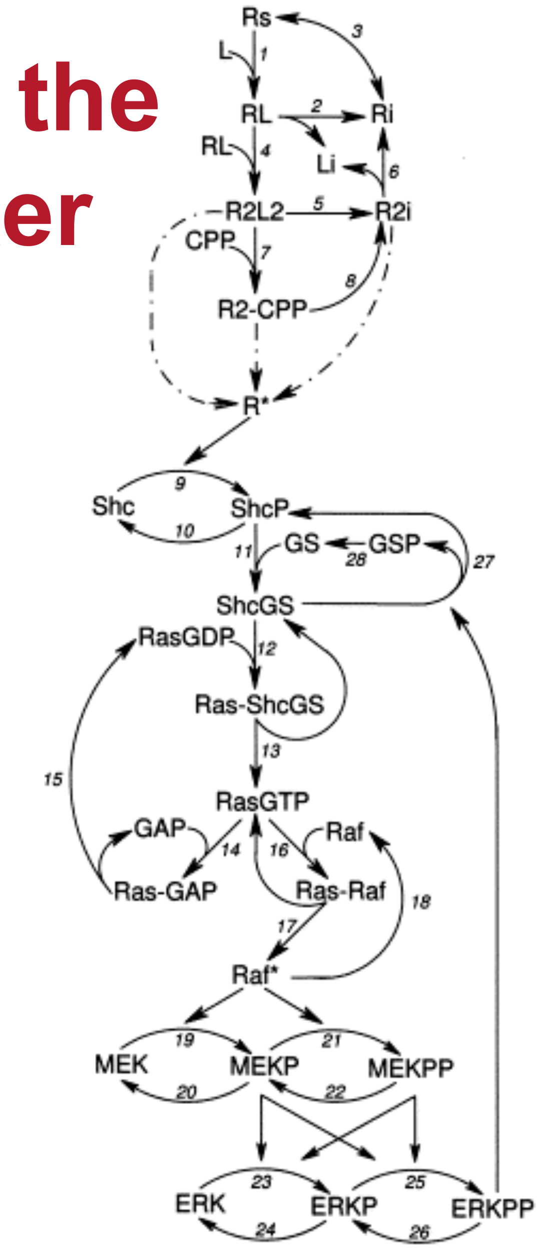
0	1	2	3	4	5	6	7	8	9	10	11	12	13	…	∞
0	0	0	0	0	0	0	0	0	0.1	0.2	0.67	0.78	1	…	1

Parameter estimation using the Monte Carlo Model Checker

Continuous Brightman & Fell model:

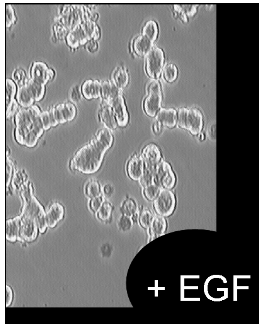
- The EGF signal transduction pathway produces transient Ras, MEK and ERK activation whereas NGF stimulation produces sustained activation.

- Parameter V28 has the the highest probability of generating the desired behaviour, but requires 40-fold increase in value



Brightman & Fell, FEBS Lett 2000. "Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells"

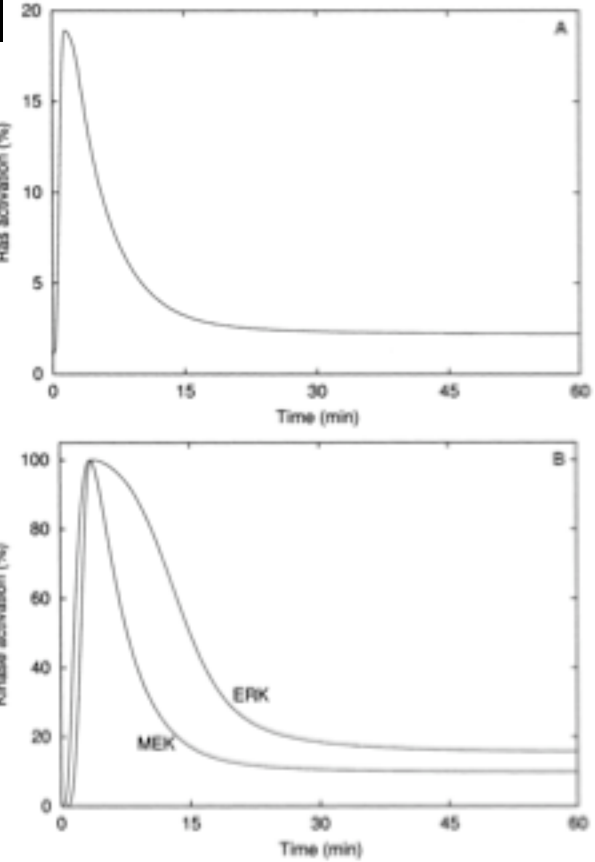
Response with EGF vs. NGF signal



Proliferation (cell division) PC12 cells

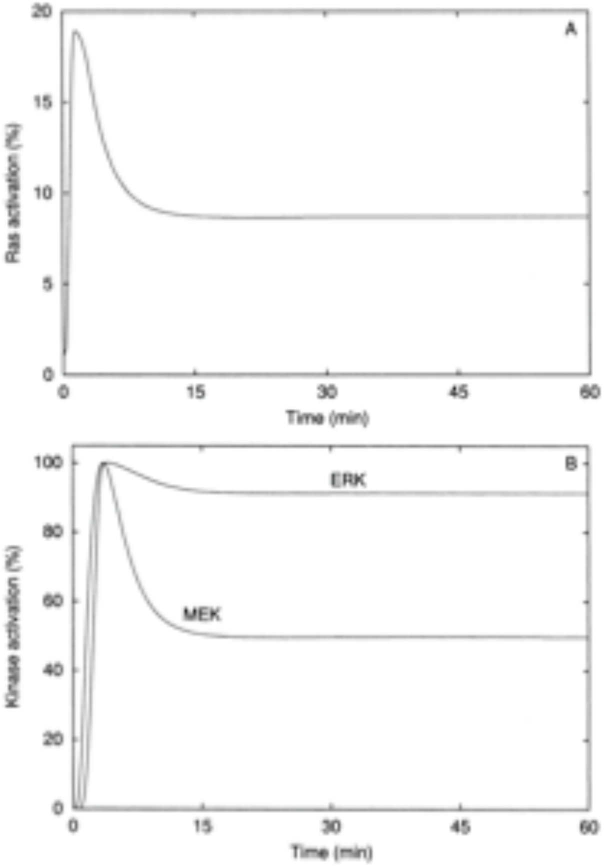
EGF

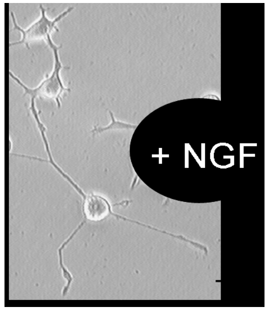
Transient activation of Ras, MEK and ERK



NGF

Sustained activation of Ras, MEK and ERK





Differentiation (neurite outgrowth)

David Gilbert ECSBII 2009

Desired Behaviour in PLTLc

The desired (sustained) NGF behaviour of the pathway was written informally in the original paper.

We can formalise it in PLTLc as:

Sustained Ras: Active Ras peaks within 2 min to max 20% total Ras and stable between 5%..10%

$$P_{=?} [\ d(\text{active Ras}) > 0\ \ U\ (\ \text{time} \leq 2\ \wedge\ \text{active Ras} \geq 0.15 * \text{total Ras}$$
$$\wedge\ \text{active Ras} \leq 0.2 * \text{total Ras}\ \wedge\ (\ d(\text{active Ras}) < 0)$$
$$U\ (\ G(\ \text{active Ras} \geq 0.05 * \text{total Ras}\ \wedge\ \text{active Ras} \leq 0.10 * \text{total Ras}\ )\ )\ )\ ]$$

Sustained MEK: Active MEK peaks in 2 to 5 min and is stable between 40%..50% of peak value

$$P_{=?} [\ d(\text{MEKPP}) > 0\ U\ (\ \text{time} \geq 2\ \wedge\ \text{time} \leq 5\ \wedge\ d(\text{MEKPP}) < 0$$
$$U\ (\ G(\ \text{MEKPP} \geq 0.40 * \max(\text{MEKPP})\ \wedge\ \text{MEKPP} \leq 0.50 * \max(\text{MEKPP})\ \ )\ )\ )\ ]$$

Sustained ERK: Active ERK peaks in 2 to 5 min and is stable between 85%..100% of peak value

$$P_{=?} [\ (\ d(\text{ERKPP}) > 0\ )\ U\ (\ \text{time} \geq 2\ \wedge\ \text{time} \leq 5\ \wedge\ d(\text{ERKPP}) < 0$$
$$U\ (\ G(\ \text{ERKPP} \geq 0.85 * \max(\text{ERKPP})\ )\ )\ )\ ]$$

Robin Donaldson and David Gilbert (2008). A Model Checking Approach to the Parameter Estimation of Biochemical Pathways In proceedings CMSB 2008 (Computational Methods in Systems Biology). LNCS 5307/2008, pp269-287

Critical parameters

Critical parameters can produce sustained activation of Ras, MEK or ERK.

Used to give an idea which parameters to vary

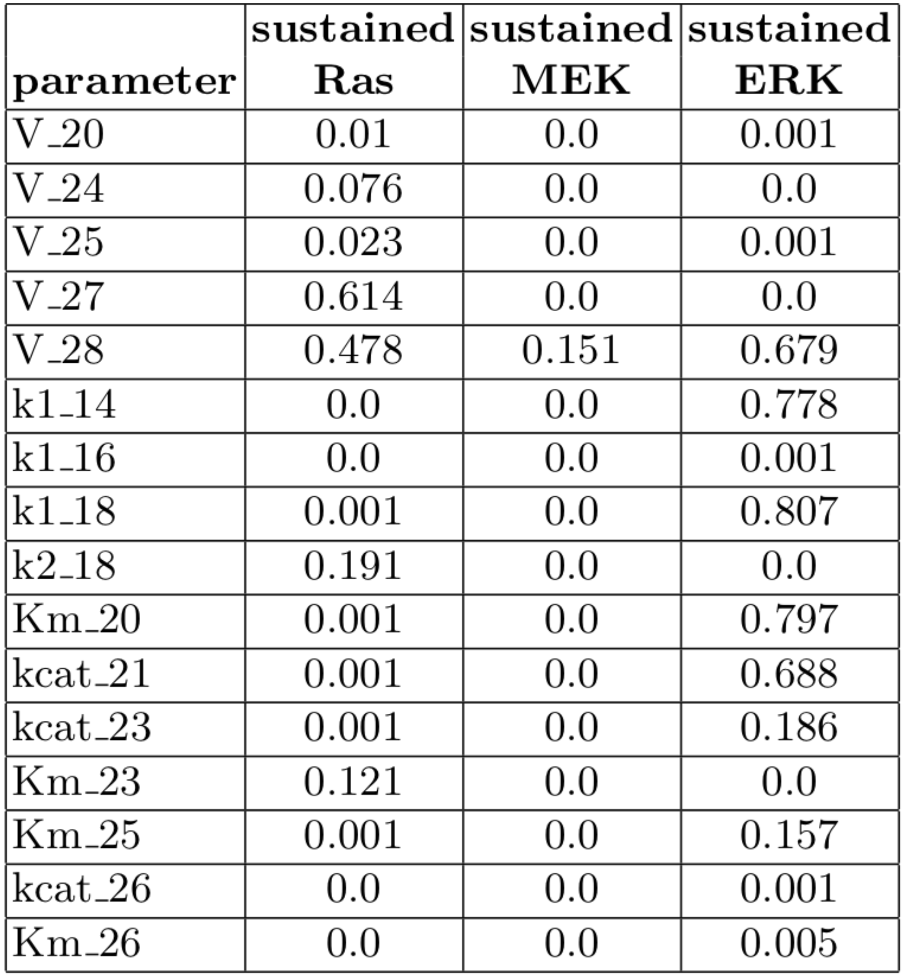
Method:

- Vary the kinetic rate constant parameters in range ± 2 orders of magnitude from original value.

- Perform 1,000 simulations using different values for each parameter, linearly spaced in the range

- The 'significance values' are the fraction of values in the range which give rise to sustained behaviour for each protein

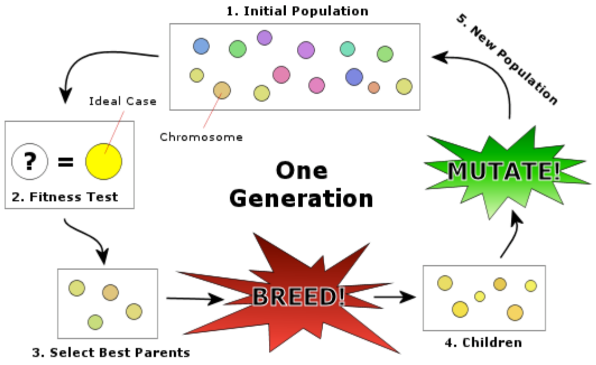
- Found through model checking



parameter	sustained Ras	sustained MEK	sustained ERK
V_20	0.01	0.0	0.001
V_24	0.076	0.0	0.0
V_25	0.023	0.0	0.001
V_27	0.614	0.0	0.0
V_28	0.478	0.151	0.679
k1_14	0.0	0.0	0.778
k1_16	0.0	0.0	0.001
k1_18	0.001	0.0	0.807
k2_18	0.191	0.0	0.0
Km_20	0.001	0.0	0.797
kcat_21	0.001	0.0	0.688
kcat_23	0.001	0.0	0.186
Km_23	0.121	0.0	0.0
Km_25	0.001	0.0	0.157
kcat_26	0.0	0.0	0.001
Km_26	0.0	0.0	0.005

2000 models, 100 generations, 2.10^5 simulations/checks

Model construction



Starting with the EGF (transient) model, construct the NGF (sustained) model by varying the values of the ***critical*** kinetic rate constants. I.e. minimise the distance of the model to sustained behaviour.

Could vary the initial concentrations, or topologies (ongoing research).

Genetic algorithm:

- Define a parameter space (±2 orders of magnitude for each parameter)
- Initialise a population of models randomly throughout parameter space
- For each generation of the algorithm:
  - Perform genetic operations on binary representations of the models' parameter values (crossover, mutation, reproduction)
  - Evaluate all models' fitness values using model checking – fitness is the distance to sustained behaviour
  - Probabilistically select models to survive to the next generation based on their fitness value

Fitness function using PLTLc

Probability:

- Can optimise the probability of a behaviour, which works fine on stochastic models.
  - On continuous models, the probability is boolean so not good in a fitness function – no gradient

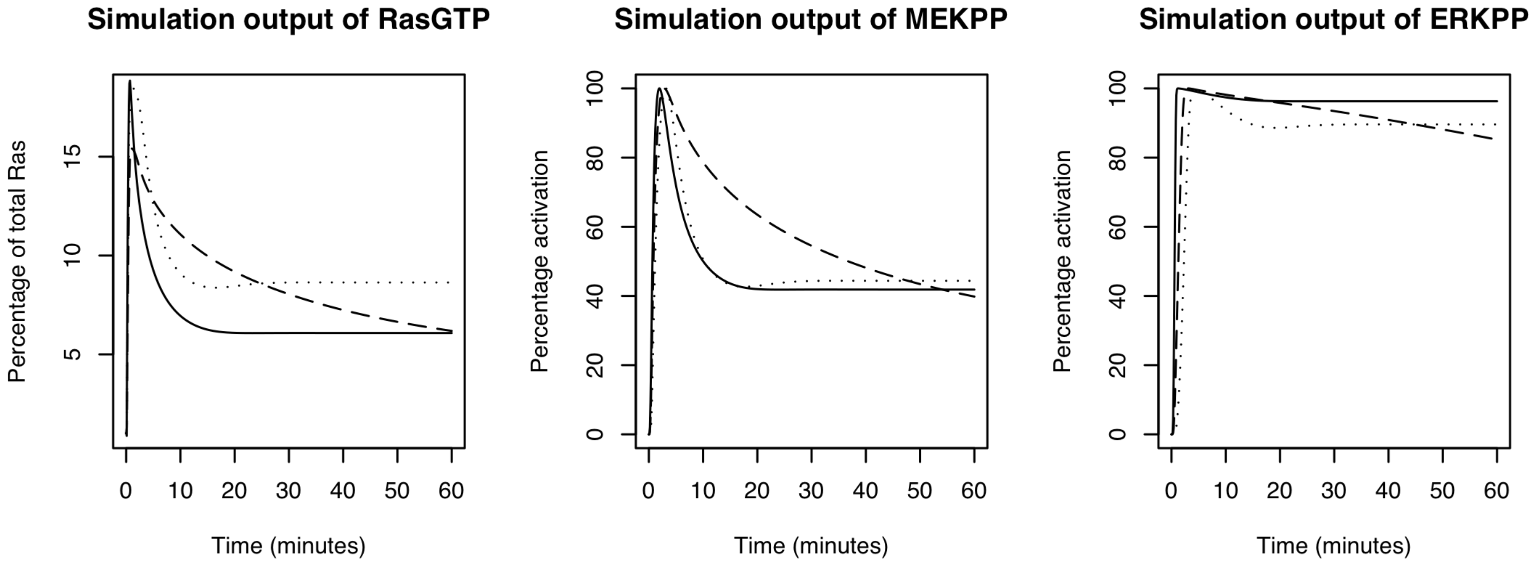
Free variables:

- Can use the free variables in a PLTLc behaviour, works for continuous or stochastic models.
  - Can always get a numerical value for the fitness function, even in continuous models – good for search algorithm

- We specify the behaviour in PLTLc and at the same time characterise the 'tail' of the peak in a free variable.
- We have an idea of the desired behaviour of the tail and can calculate the distance, using the free variables, to give us a numerical value for the fitness function, whilst….
- the behaviour in PLTLc enforces a peak at the right position.

Results

- Built a fitness function for sustained Ras, MEK and ERK
- Ran the genetic algorithm with 100 generations with results:



Simulation output of RasGTP

Simulation output of MEKPP

Simulation output of ERKPP

- Original model of the NGF signalling pathway varying V28 (dotted)
- Best model returned when varying the critical parameters (solid)
- Critical parameters without V28 (dashed).

The best model returned when varying the critical parameters only required a ***16-fold*** increase in V28 (compared with 40-fold in original paper)

Even possible to get similar behaviour **without** varying V28

Robin Donaldson and David Gilbert (2008). A Model Checking Approach to the Parameter Estimation of Biochemical Pathways In proceedings CMSB 2008 (Computational Methods in Systems Biology). LNCS 5307/2008, pp269-287

Related work

- Francois Fages, Biocham.
  - **On a Continuous Degree of Satisfaction of Temporal Logic Formulae with Applications to Systems Biology** Aurélien Rizk, Grégory Batt, François Fages and Syvain Soliman. Comptational Methods in Systems Biology CMSB'08

Model searching

- Query the database on the behaviour of the models
- SIMAP project has a database of MAPK pathway models
- We are using MC2(PLTLc) as a SQL for models of biochemical systems
- Also want to use it as a SQL for lab data
- For example, these questions
  - “What are the models where ERK oscillates?”
  - “Under what conditions does ERK not behave as we observe?”
  - “What are the behaviours of the pathway not backed up by lab data?”

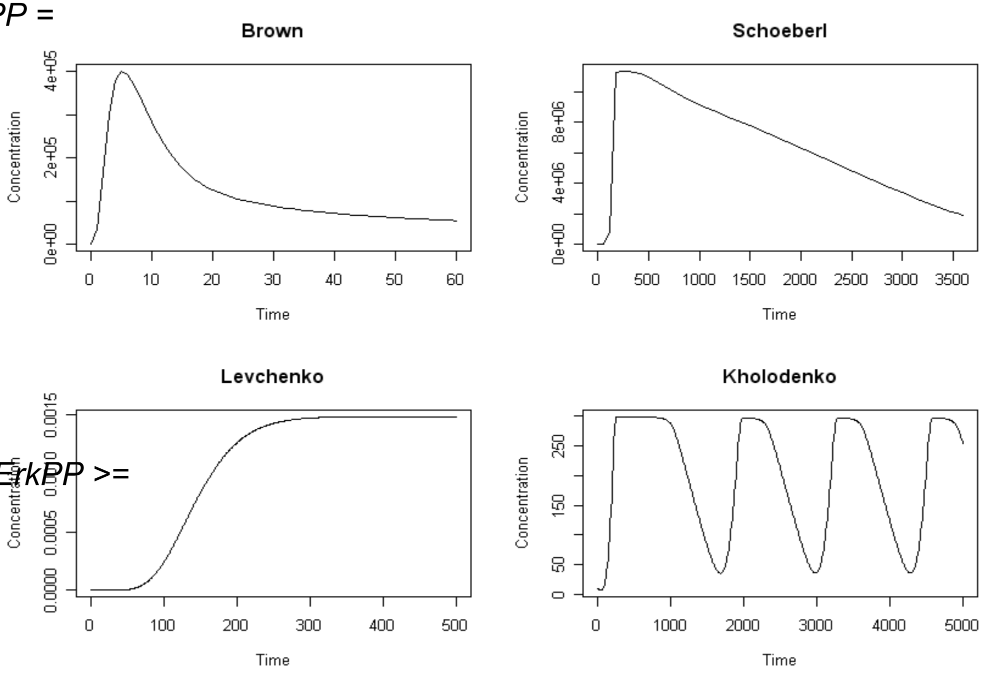
Model searching

Peaks at least once

(rises then falls below 50% max concentration)

$P_{>=1}[\ ErkPP <= 0.50*max(ErkPP)\ \wedge\ d(ErkPP) > 0\ U\ (\ ErkPP = max(ErkPP)\ \wedge\ F(\ ErkPP <= 0.50*max(ErkPP)\ )\ )\ \ ]$

- ***Brown***
- ***Kholodenko***
- ***Schoeberl***



Rises and remains constant

(99% max concentration)

$P_{>=1}[ErkPP <= 0.50*max(ErkPP)\ \wedge\ (\ d(ErkPP) > 0\ )\ U\ (\ G(ErkPP >= 0.99*max(ErkPP))\ )\ \ ]$

- ***Levchenko***

Oscillates at least 4 times

$P_{>=1}[\ \ F(\ d(ErkPP) > 0\ \wedge\ F(\ d(ErkPP) < 0\ \wedge\ \ldots\ )\ )\ \ ]$

- **Kholodenko**

BioModel Engineering

- The science of designing, constructing and analyzing computational models of biological systems
- A systematic and powerful extension of earlier mathematical modeling approaches
- Applied in systems biology and synthetic biology.

Rainer Breitling, David Gilbert, Monika Heiner, Richard Orton (2008). A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. Briefings in Bioinformatics

David Gilbert, Rainer Breitling, Monika Heiner, and Robin Donaldson (2009). An introduction to BioModel Engineering, illustrated for signal transduction pathways, 9th International Workshop, WMC 2008, Edinburgh, UK LNCS Volume 539, pp13-28

BioModel Engineering

- Takes place at the interface of computing science, mathematics, engineering and biology.
- A systematic approach for **designing**, **constructing** and **analyzing** computational models of biological systems.
- Some inspiration from efficient software engineering strategies.
- Not engineering biological systems *per se,* but
  - describes their structure and behaviour,
  - in particular at the level of intracellular molecular processes,
  - using computational tools and techniques in a principled way.

Management & development

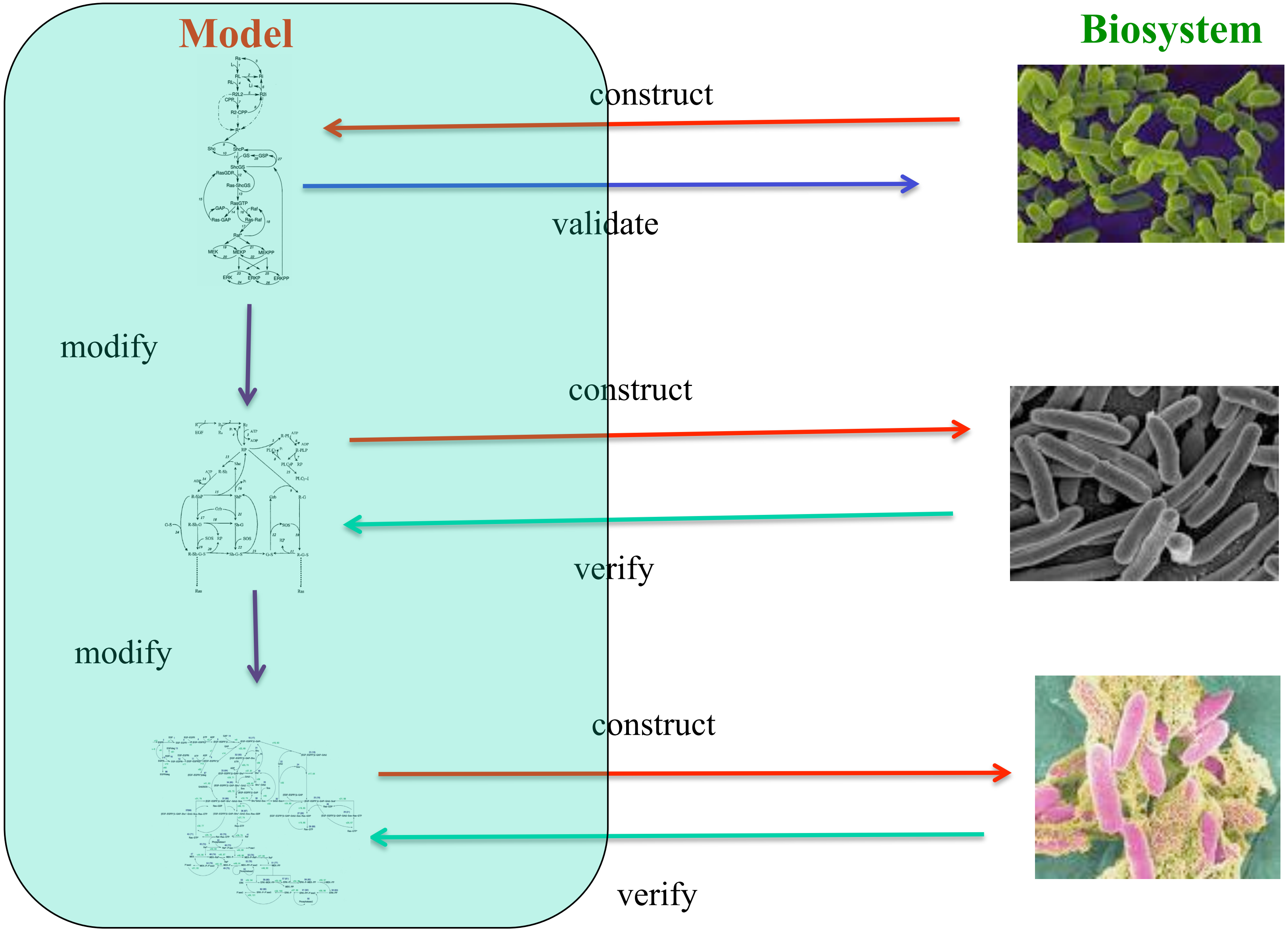
- **Database of models, behaviours, properties, …**
- **Model Version control system**
- **Model checking** – properties
- **Model behaviour checking**:
  Do the model results match the experimental data?
  - Yes: validation
  - No: back to definition and check for errors
    - Simple typos
    - Wrong kinetics
    - Over simplifications of processes
    - Missing components from the model
    - Incorrect parameter data
- Further validation – **checking if system behaves correctly when things are varied**:
  - Over-express or knockout a component
  - The model should be able to recreate this behaviour

BioNessie –BioModel Engineering environment Xuan Liu

- SBML (Systems Biology Markup Language) enabled.
- Intuitive easy-to-use interface for biochemists & modellers. Input biochemical equations.
- File storage in XML, SBML, text & graphics
- Platform Independent – Java
- Parallel processing - Efficient exploitation of available compute resources – multiple core and multiple CPUs, as well as Grid computing
- Editor, simulator, and analyser
- Model version control
- Kinetic law library creation & management
- Fast efficient ODE solver (stiff & non-stiff)
- (Stochastic solver)
- Parameter scanning
- Sensitivity analysis
- Parameter estimation using a genetic algorithm
- Advanced model checking (MC2 using PLTL)
- Module composition
- Relational database connectivity

www.BioNessie.org

Synthetic biology development cycle



Model

Biosystem

construct

validate

modify

construct

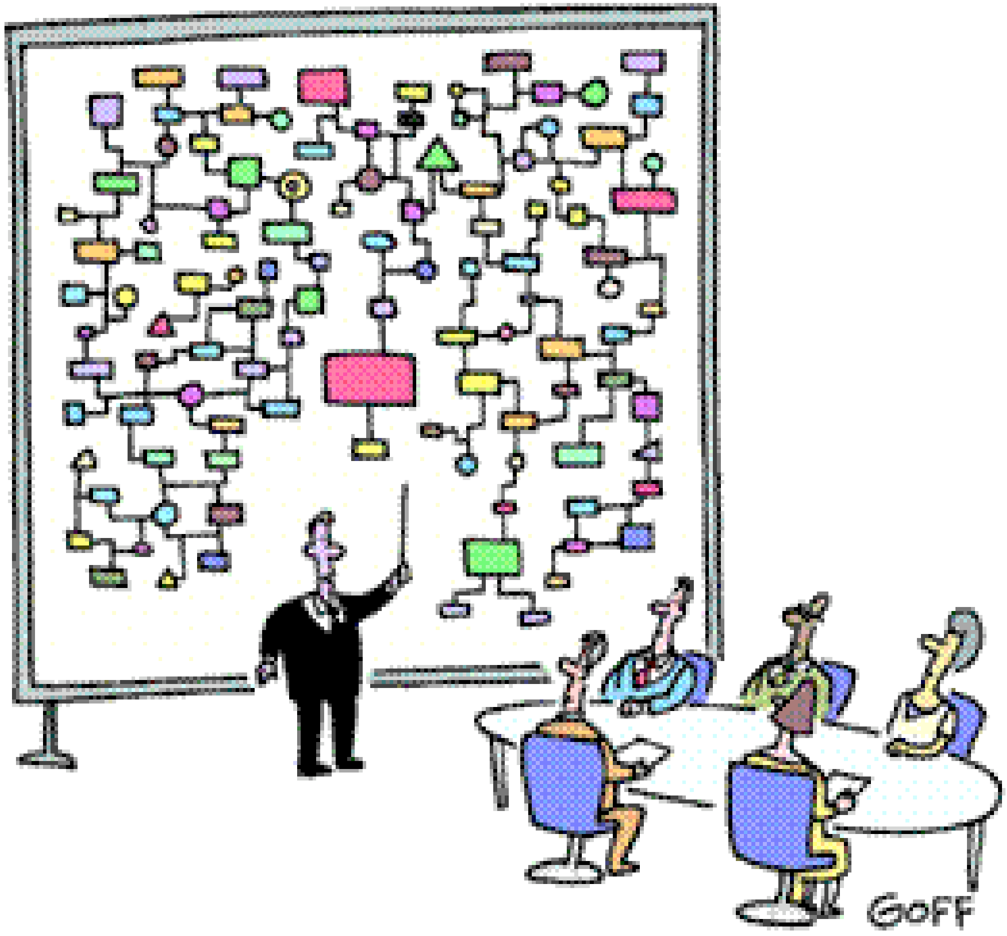
verify

modify

construct

verify

David Gilbert ECSBII 2009



"And that's why we need a computer."

Acknowledgements

MC2: **Robin Donaldson** – University of Glasgow

MC2(PLTLc) website (software + examples)

- www.brc.dcs.gla.ac.uk/software/mc2/

BioNessie(+MC2): Xuan Liu – Brunel University

- www.bionessie.org

Biomodel engineering:
Rainer Breitling (Gronigen), Monika Heiner (Cottbus)

Funding: EU FP6 “Simulation Modelling of the Map Kinase Pathway” (SIMAP) project.

Some references

David Gilbert, Rainer Breitling, Monika Heiner, Robin Donaldson (2008), An introduction to BioModel Engineering, illustrated for signal transduction pathways, Proc WMC9, LNCS Volume 5391, pp13-28

Robin Donaldson and David Gilbert (2008). **A Model Checking Approach to the Parameter Estimation of Biochemical Pathways**. In proceedings CMSB 2008 (Computational Methods in Systems Biology). Heiner, Monika; Uhrmacher, Adelinde M. (Eds.) LNCS 5307/2008, pp269-287.

David Gilbert, Monika Heiner, Richard Orton (2008). **A structured approach for the engineering of biochemical network models, illustrated for signalling pathways**. Briefings in Bioinformatics, 2008 9(5): 404-42

Monika Heiner, David Gilbert, and Robin Donaldson (2008), **Petri Nets for Systems and Synthetic Biology**. In M Bernardo, P Degano, and G Zavattaro (Eds.): Formal Methods for Systems Biology SFM 2008, Springer LNCS 5016, pp. 215-264

Xuan Liu, Jipu Jiang, Oluwafemi Ajayi, Xu Gu, David Gilbert, Richard Sinnott (2008), '**BioNessie(G) - A Grid Enabled Biochemical Networks Simulation Environment**'. Stud Health Technol Inform. IOS Press, 2008, 138: 147-157

David Gilbert, Monika Heiner, Susan Rosser, Rachael Fulton, Xu Gu and Maciej Trybilo (2008), **A Case Study in Model-driven Synthetic Biology. In Biologically Inspired Cooperative Computing**: BICC 2008. IFIP

David Gilbert, Monika Heiner and Sebastian Lehrack (2007). **A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets** In proceedings CMSB 2007 (Computational Methods in Systems Biology), Editors: M.Calder and S.Gilmore, Springer LNCS/LNBI 4695, pp. 200-216.

David Gilbert, Hendrik Fuß, Xu Gu, Richard Orton, Steve Robinson, Vladislav Vyshemirsky, Mary Jo Kurth, C. Stephen Downes and Werner Dubitzky. (2006**) Computational methodologies for modelling, analysis and simulation of signalling networks**, Briefings in Bioinformatics 2006 7(4):339-353; doi: 10.1093/bib/bbl043 Special Issue: Computational Methodologies for Systems Biology.

Available from http://people.brunel.ac.uk/~csstdrg