BME Tutorial – Part 5
Model checking

Rainer Breitling, Groningen, NL

David Gilbert, Brunel, UK

Monika Heiner, Cottbus, DE

Model Checking - Summary

- This presentation for users

- Analytical vs simulative model checking

- Model checking in the 3 worlds

- Break

- Model checking for BME:

  – parameter fitting
  – database searching
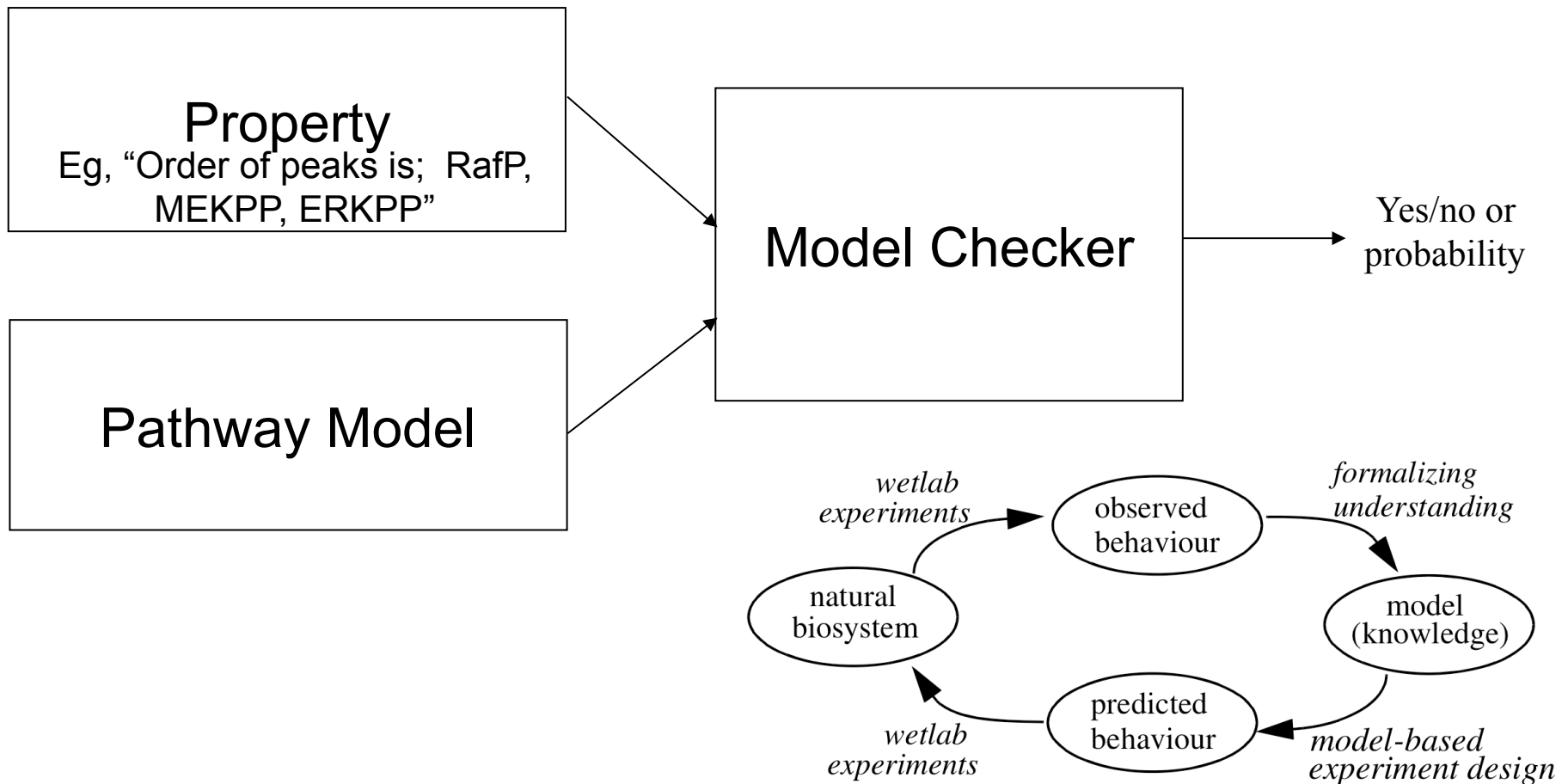 – model comparison

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

wetlab experiments

observed behaviour

formalizing understanding

natural biosystem

model (knowledge)

predicted behaviour

wetlab experiments

model-based experiment design

What can we do with Model Checking?

- **Model validation**:
    - Show that your model of the pathway matches the lab data (…stochastic…)
- **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?
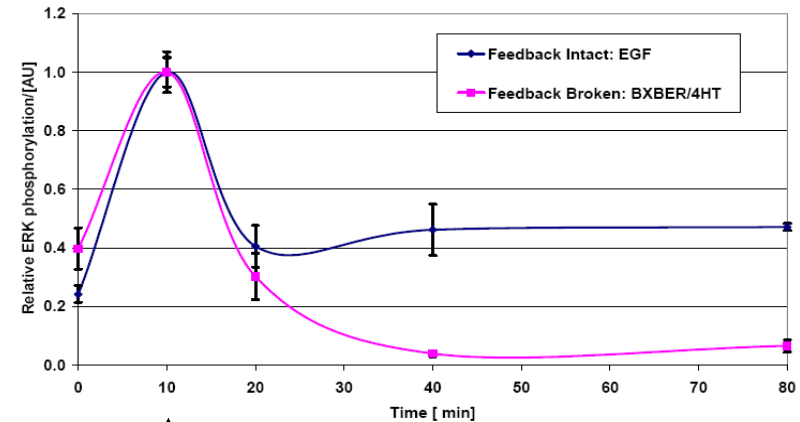
Describing experimental data

- 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
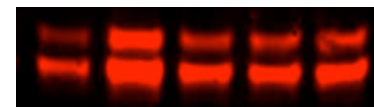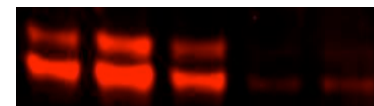
Lab data versus simulations

Simulation
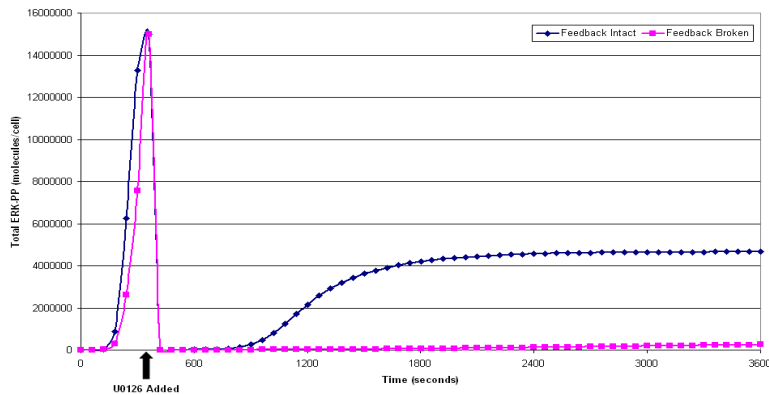
Experiment

U0126 added

pERK1/2, +EGF

pERK1/2, + BXBER/4HT

0   10   20   40   80 min stimulation

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

To formally express properties we use temporal logics

Various logics each with different expressivity

Branching logics consider all branching time lines.
  – Computational Tree Logic (CTL)
  – Continuous Stochastic Logic (CSL)

"There is a possibility that I will stay hungry forever."

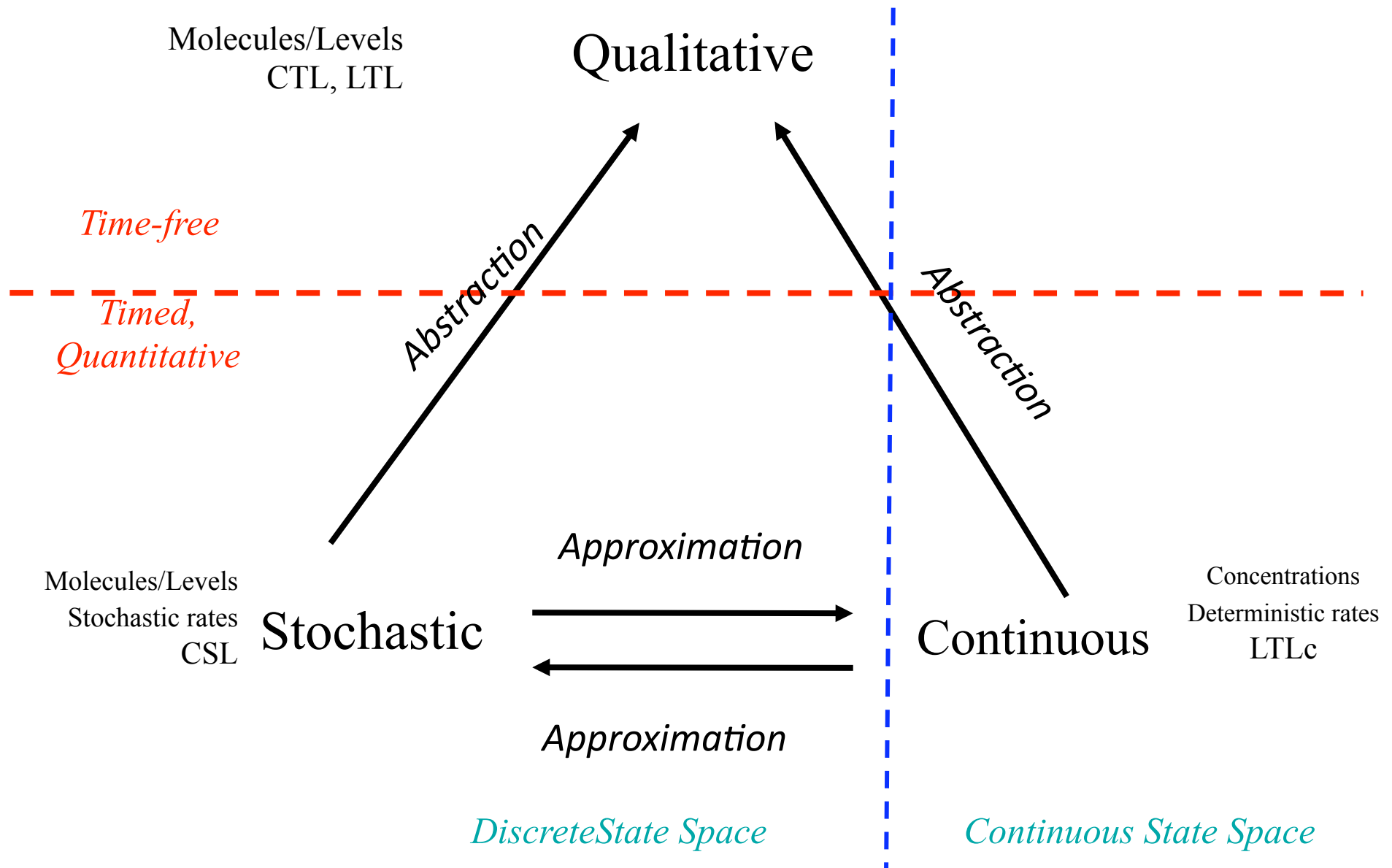*"There is a possibility that eventually I am no longer hungry."*

Linear time logics consider separately all single time lines.
  – Linear-time Temporal Logic (LTL, LTLc, PLTLc)

"I am hungry."

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

"I will be hungry until I eat something".

Molecules/Levels
CTL, LTL

Qualitative

Time-free

Abstraction

Timed,
Quantitative

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

BME Model Checking   david.gilbert@brunel.ac.uk

10

Models

- **Qualitative**:
  - Petri nets,…

- **Continuous**:
  - ODEs, Continuous Petri nets

- **Stochastic**:
  - Stochastic process algebras, Stochastic Petri nets, P-systems, (Chemical master equations)…

Levchenko Model
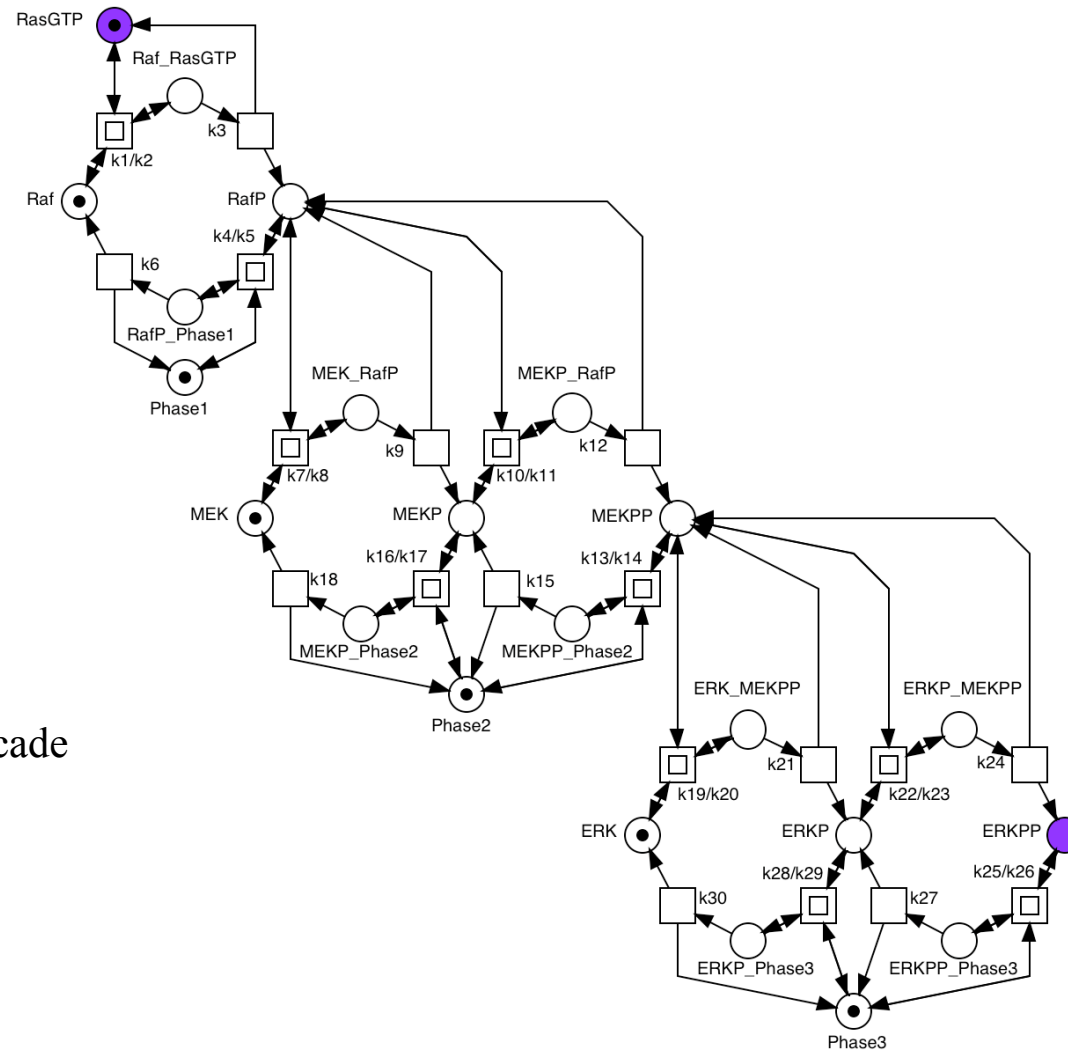
Typical signalling cascade

Input RasGTP
Output ERKPP

CTL Computational Tree Logic

- EX φ : if there is a state reachable by one step where φ holds.

- EF φ : if there is a path where φ holds finally, i.e., at some point.

1

- EG φ : if there is a path where φ holds globally, i.e., forever.

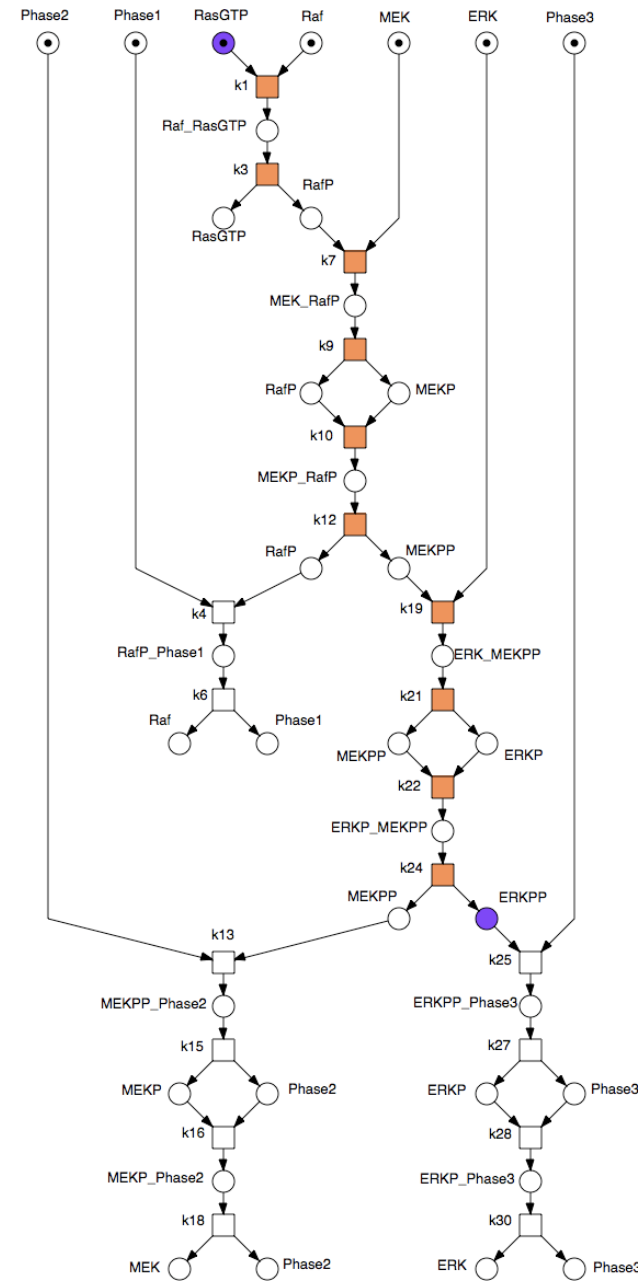- E ($φ_1$ U $φ_2$) : if there is a path where $φ_1$ holds until $φ_2$ holds

CTL Computational Tree Logic

- AX φ : if φ holds for all states which are reachable by one step.

- AF φ : if φ holds finally (at some point) for all paths.

1

- AG φ : if φ holds globally (i.e. for ever) for all paths.

- A (φ_1 U φ_2) : if φ_1 holds until φ_2 holds for all paths.

Infinite run (beginning)

Generated from the input-output T-invariant

Qualitative: CTL properties

- **property Q1**: The signal sequence predicted by the partial order run of the I/O T-invariant is the only possible one. I.e., starting at the initial state, it is necessary to pass through states RafP, MEKP, MEKPP and ERKP in order to reach ERKPP.

$$\neg\, [\, E\, (\, \neg\, RafP\, U\, MEKP\,)\, \vee$$
$$E\, (\, \neg\, MEKP\, U\, MEKPP\,)\, \vee$$
$$E\, (\, \neg\, MEKPP\, U\, ERKP\,)\, \vee$$
$$E\, (\, \neg\, ERKP\, U\, ERKPP\,)\,]$$

Qualitative: CTL properties

- **property Q2**: Dephosphorylation takes place independently. E.g., the duration of the phosphorylated state of ERK is independent of the duration of the phosphorylated states of MEK and Raf.

$$
\begin{aligned}
(\ & EF\ [\ Raf \land (\ ERKP \lor ERKPP\ )\ ]\quad \land \\
& EF\ [\ RafP \land (\ ERKP \lor ERKPP\ )\ ] \land \\
& EF\ [\ MEK \land (\ ERKP \lor ERKPP\ )\ ] \land \\
& EF\ [\ (\ MEKP \lor MEKPP\ ) \land (\ ERKP \lor ERKPP\ )\ ]\quad)
\end{aligned}
$$

- DSSZ-MC (BDD-CTL, IDD-CTL) – Cottbus
  (qualitative: pn2009 paper)

CSL Continuous Stochastic Logic

- Replaces the path quantifiers (E, A) in CTL by the probability operator $P_{\bowtie p}$, where $\bowtie p$ specifies the probability p of the formula.

- $P_{=?}$ [X φ] : prob there is a state reachable by one step where φ holds.

- $P_{=?}$ [F φ] : prob there is a path where φ holds finally, i.e., at some point.

- $P_{=?}$ [G φ] : prob there is a path where φ holds globally, i.e., forever

- $P_{=?}$ [(φU φ)] : prob there is a path where φ holds until φ holds
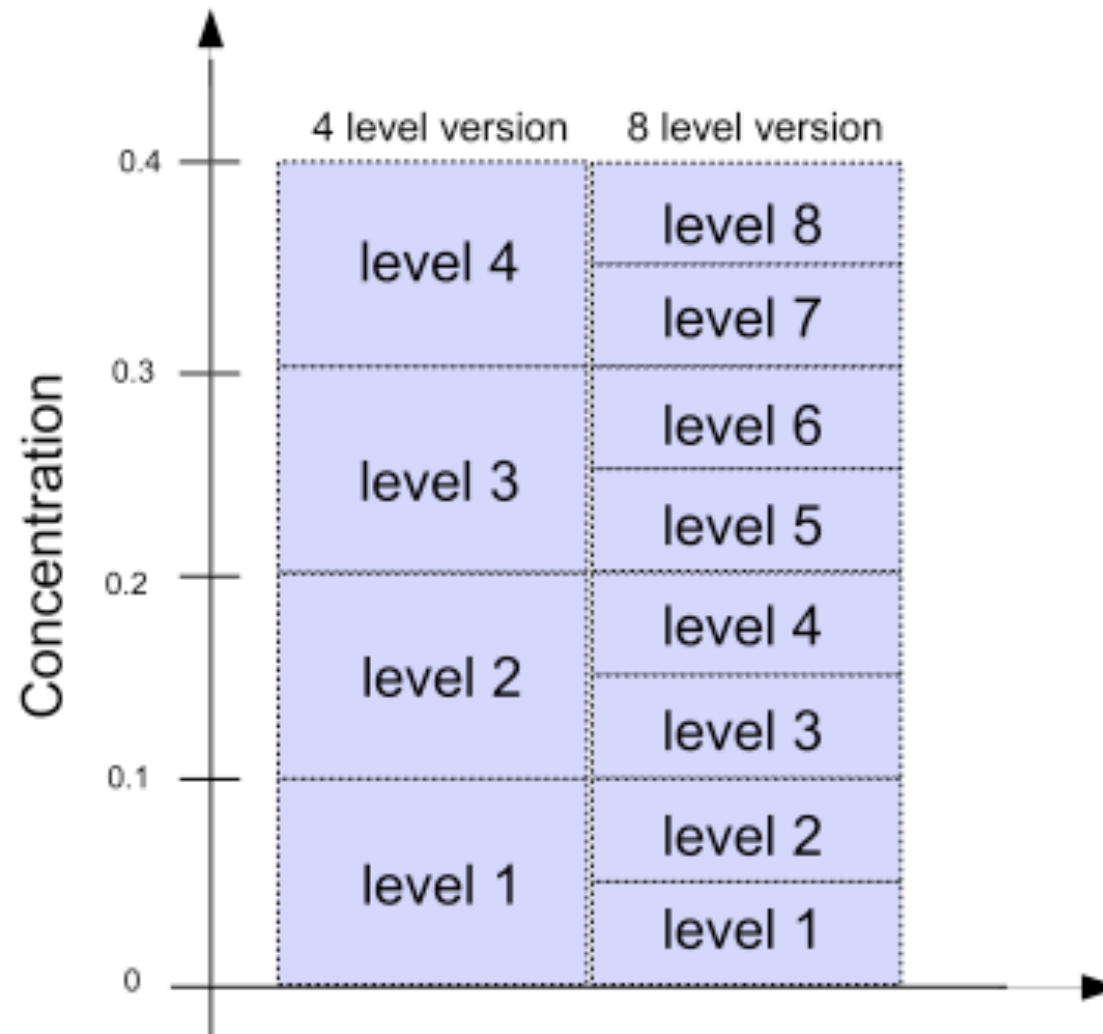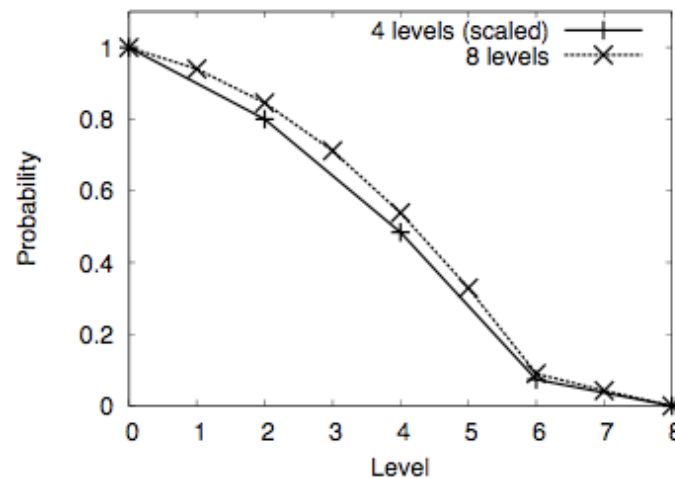
Levels

Stochastic: CSL properties

- **property S1**: What is the probability of the concentration of RafP increasing, when starting in a state where the level is already at L (the latter side condition is specified by the filter given in braces)?
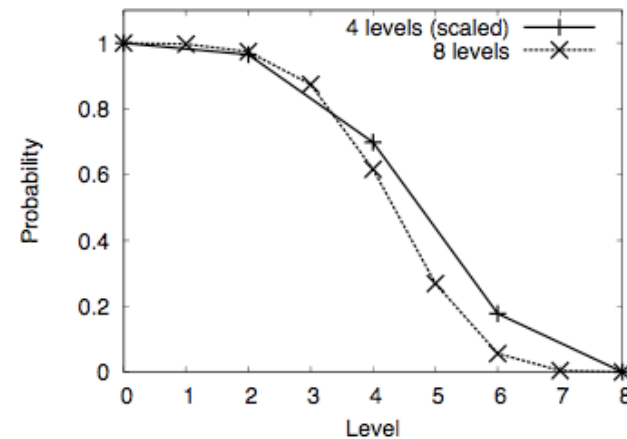
$$P_{=?} [(RafP = L) \; U^{<=100} \; (RafP > L) \; \{ RafP = L \}]$$

CSL properties

- **property S2**: What is the probability that, given the initial concentrations of RafP, MEKPP and ERKPP being zero, the concentration of RafP rises above some level L while the concentrations of MEKPP and ERKPP remain at zero, i.e. RafP is the first species to react?

$P_{=?}$ [((MEKPP = 0) $\wedge$ (ERKPP = 0)) U $^{<=100}$
(RafP > L) { (MEKPP = 0) $\wedge$ (ERKPP = 0) $\wedge$ (RafP = 0) }]

LTL Linear Temporal Logic

- $G(\varphi)$: φ always happens
- $F(\varphi)$: φ happens at some time
- $X(\varphi)$: φ happens in the next time point
- $\varphi_1 \ U \ \varphi_2$: φ_1 happens until φ_2 happens

Syntactic sugar (also for CSL)

- $\varphi_1 \ \{\varphi_2\}$: φ_1 happens from the first time φ_2 happens, where no temporal operators in φ_2

Time-series (continuous)
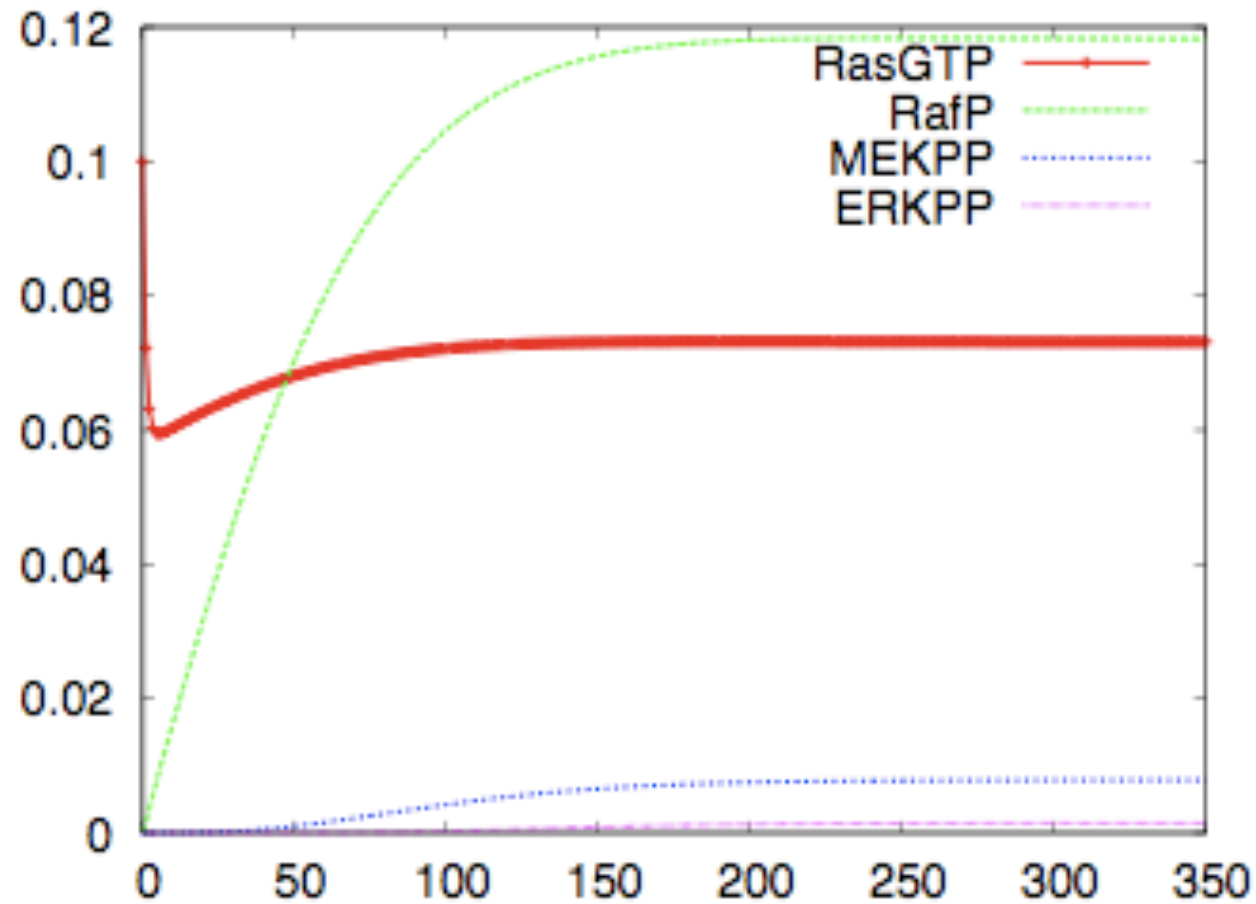
Continuous: LTLc properties

- **property C1**: The concentration of RafP rises to a significant level, while the concentrations of MEKPP and ERKPP remain close to zero; i.e. RafP is really the first species to react.

$$((MEKPP < 0.001) \land (ERKPP < 0.0002)) \ U \ (RafP > 0.06)$$

Continuous: LTLc properties

- **property C2**: if the concentration of RafP is at a significant concentration level and that of ERKPP is close to zero, then both species remain in these states until the concentration of MEKPP becomes significant; i.e. MEKPP is the second species to react.

$$(\ (RafP > 0.06) \wedge (ERKPP < 0.0002) \ ) \Rightarrow$$
$$(\ (RafP > 0.06) \wedge (ERKPP < 0.0002) \ ) \ U \ (MEKPP > 0.004)$$

Continuous: LTLc properties

- **property C3**: if the concentrations of RafP and MEKPP are significant, they remain so, until the concentration of ERKPP becomes significant; i.e. ERKPP is the third species to react.

$$(\, (RafP > 0.06) \wedge (MEKPP > 0.004) \,) \Rightarrow$$
$$(\, (RafP > 0.06) \wedge (MEKPP > 0.004) \,) \, U \, (ERKPP > 0.0005)$$

Analytical vs Simulative Model Checking

- **Analytical**:
  - Exact probabilities & prove properties
  - A model state is an association of #molecules/levels 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, or even be infinite!
  - Stochastic model checking with even as little as 12 molecules/levels 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

Probabilistic analytical model checkers

PRISM

- Specification 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)

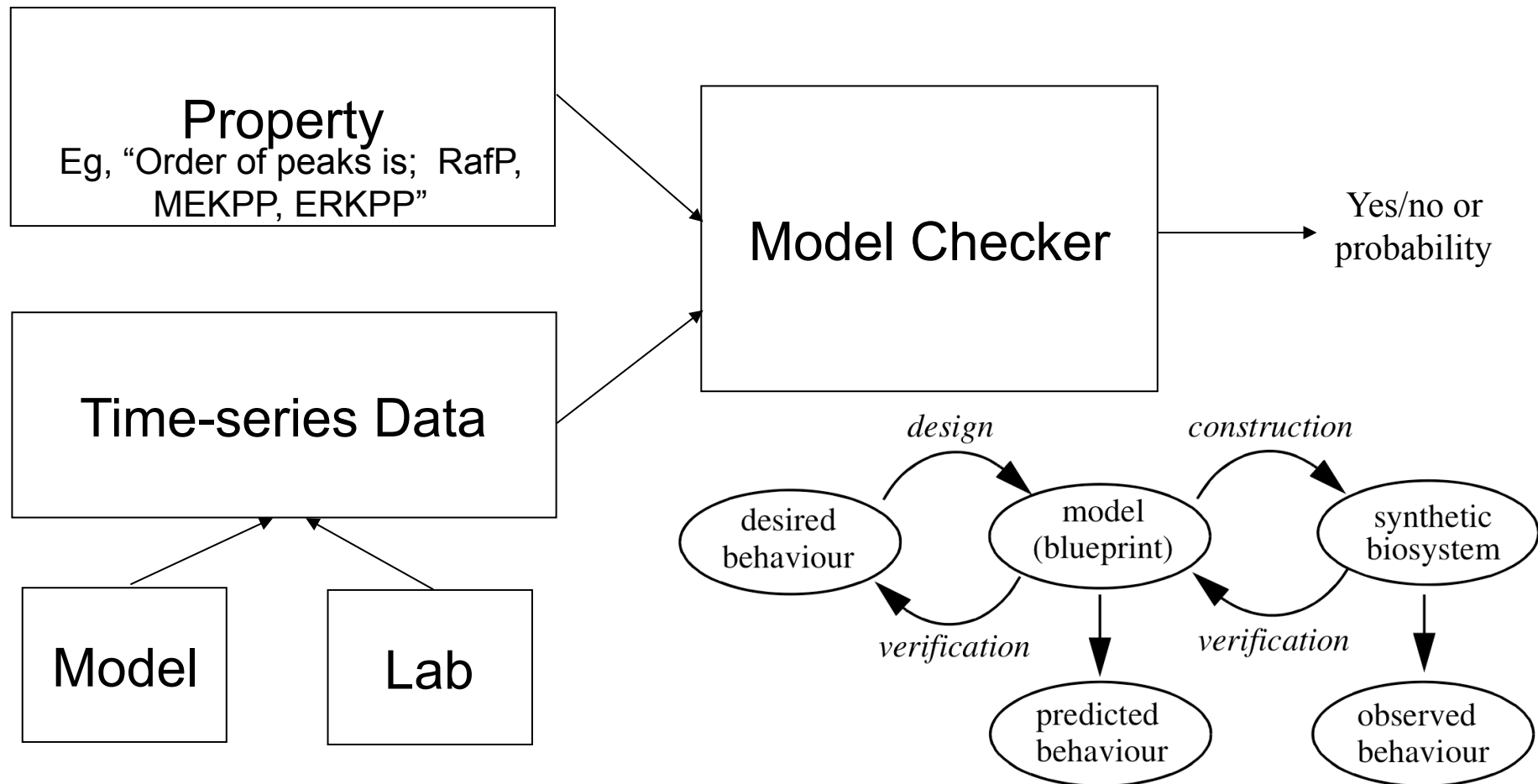Quantitative analysis using costs/rewards
Can run experiments

IDD-CSL

- New system under development from Cottbus (stochastic 1st prototype AWPN 2008, CMSB2009)

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

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

MC2: Monte Carlo Model Checker

- *Offline* Monte Carlo Model Checker for PLTLc properties.

- 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

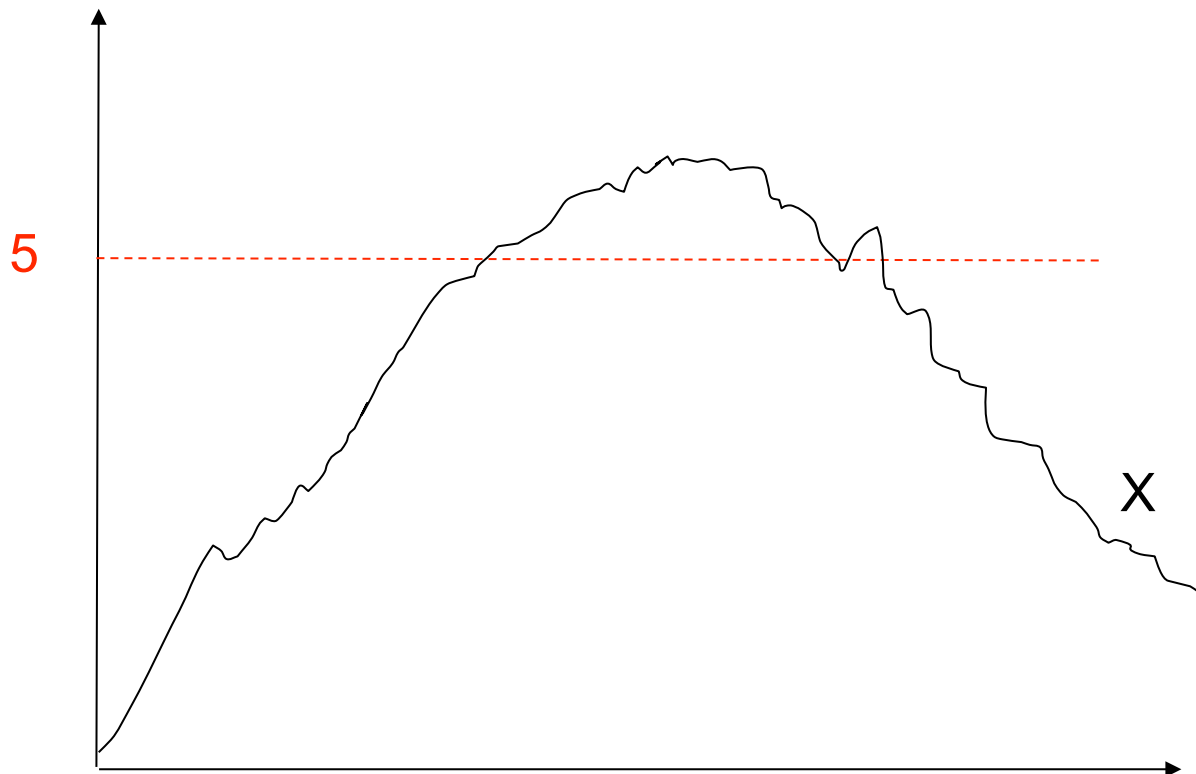# MC2 with ODE Output

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

=> P = 1

5

X

david.gilbert@brunel.ac.uk
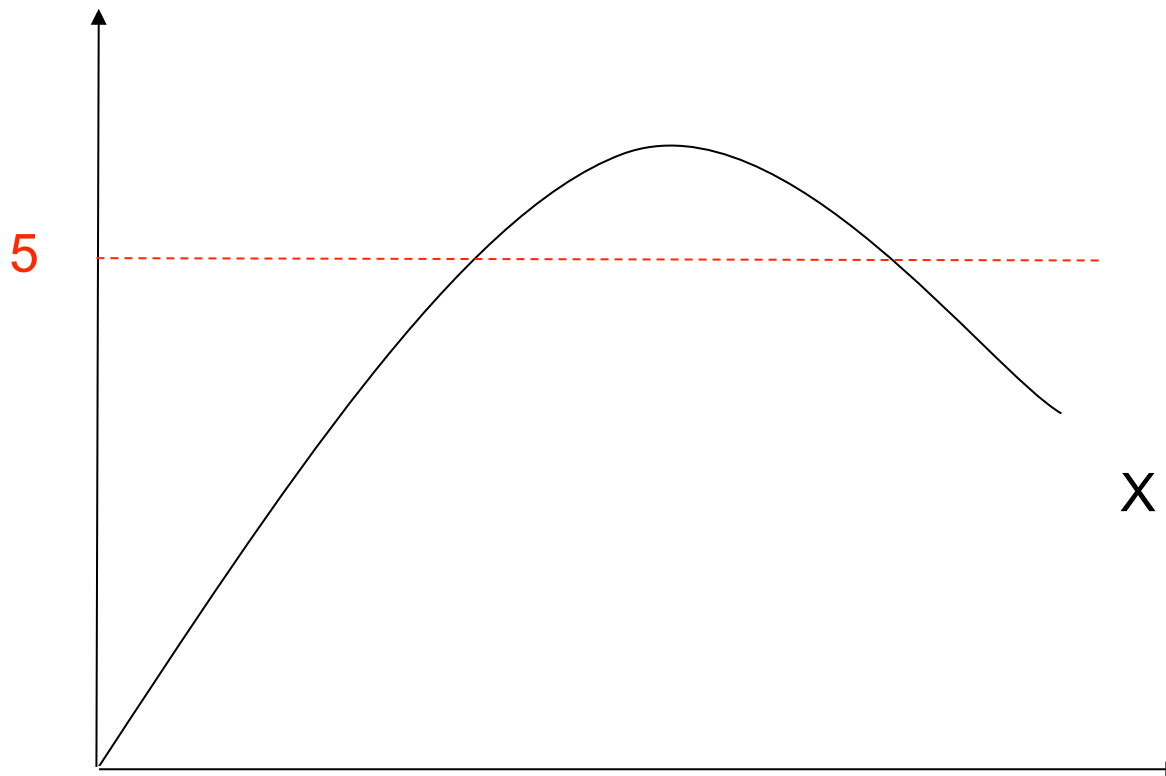
MC2 with Gillespie Output

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

=> P = 1

5 ⌇ ⌇ ⌇ ⌇ ⌇ ⌇ ⌇ ⌇ ⌇ ⌇

X

MC2 with Gillespie Output

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

=> P = 4/6

5

X
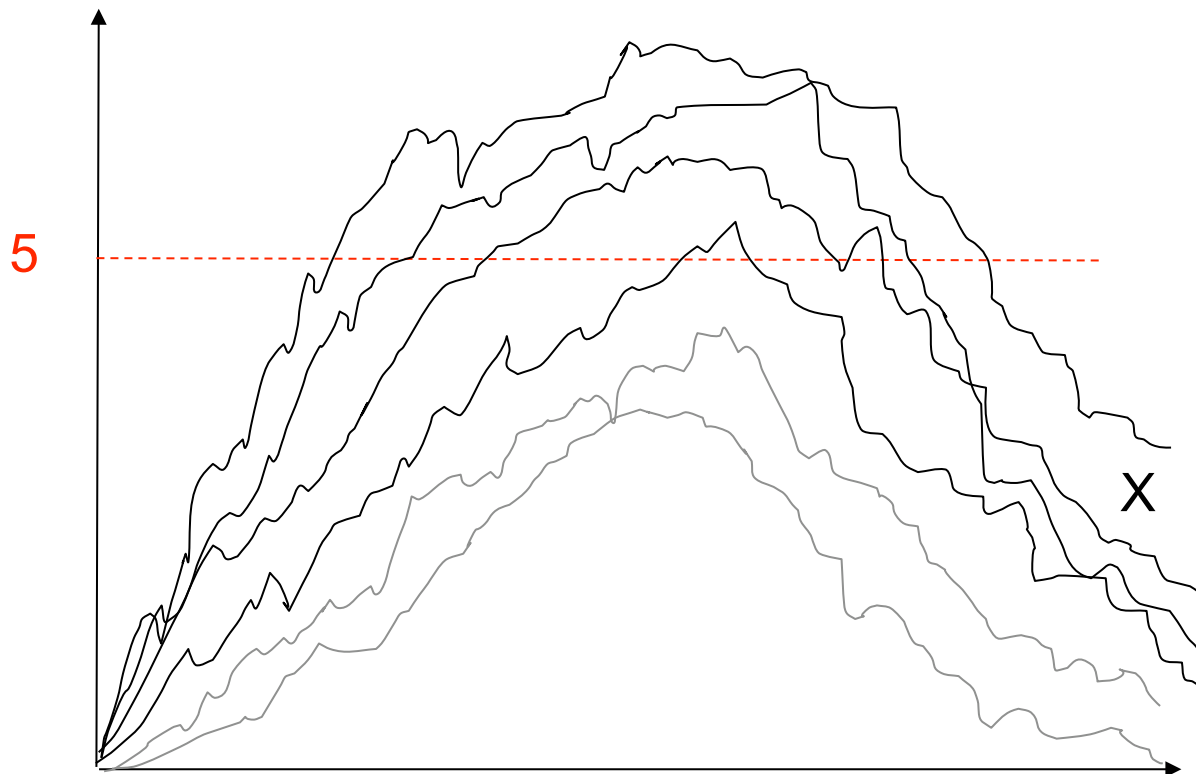
MC2 versus Analytical Checkers

- 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 worse than exponential or even infinite 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.
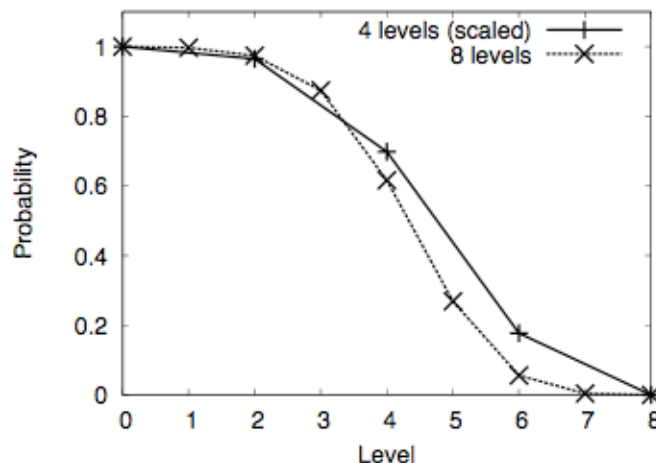
Gilbert *et al.* (2007), "A unifying framework for modelling and analysing biochemical pathways using Petri nets".
In Proc. CMSB 2007, pages 200-216.

MC2 versus Analytical checkers

- Can easily perform model checking on a cluster - Fast!

- Fast, but it is an approximation…

- No need to *explicitly* impose time-bounds in formulae on operators for efficiency as done in Gilbert *et al.* (2007)

  $P_{=?}[\ (RafP = X)\ U_{<=100}\ (RafP > X)\ \{\ RafP = X\ \}\ ]$

  – but implicit time bounds due to finite simulation runs…

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

Stochastic Analysis

Check the property S2:

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

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

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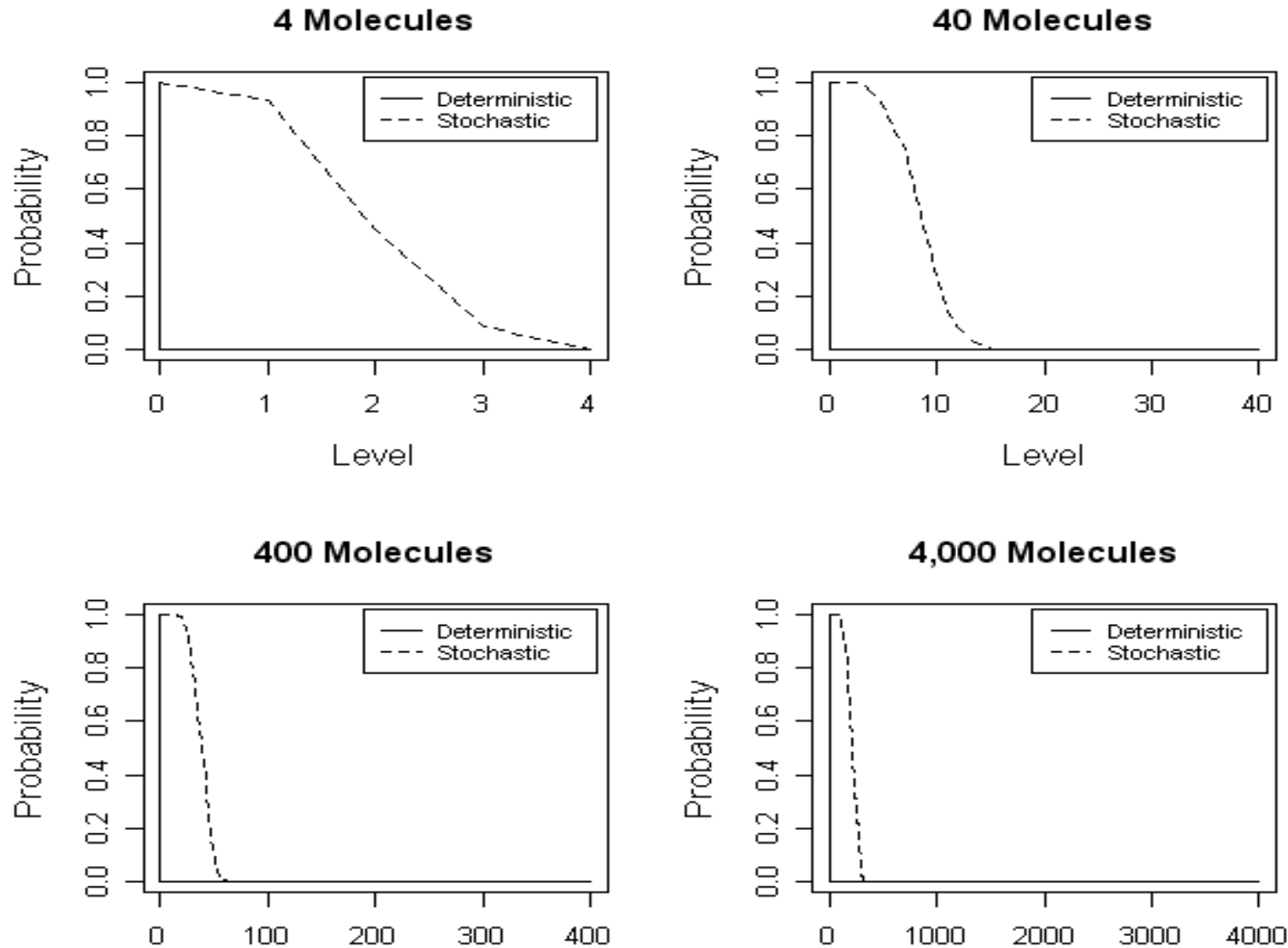
Monika Heiner, David Gilbert, and Robin Donaldson (2008), Petri Nets for Systems and Synthetic Biology. In M Bernardo, P Degano, and G Zavattaro (Eds.): Formal Methods for Systems Biology SFM 2008, Springer LNCS 5016

PLTLc language, specifics

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

(2) PLTLc [Donaldson et al]: Add probabilistic operator & free variables

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, pp48–63. LNCS/ LNBI4695, Springer,2007.*

Donaldson and Gilbert. 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

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] :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	…	∞
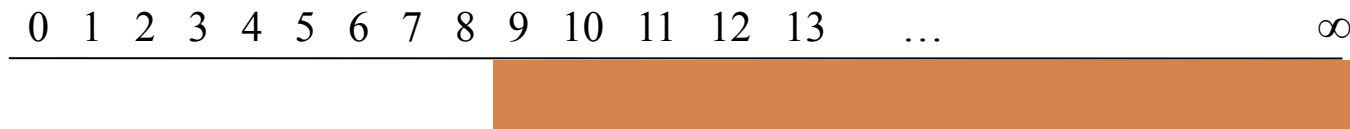
- Probabilistic language, interpret this as a probabilistic domain:

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

Qualitative to quantitative descriptions in PLTL

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

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

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

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

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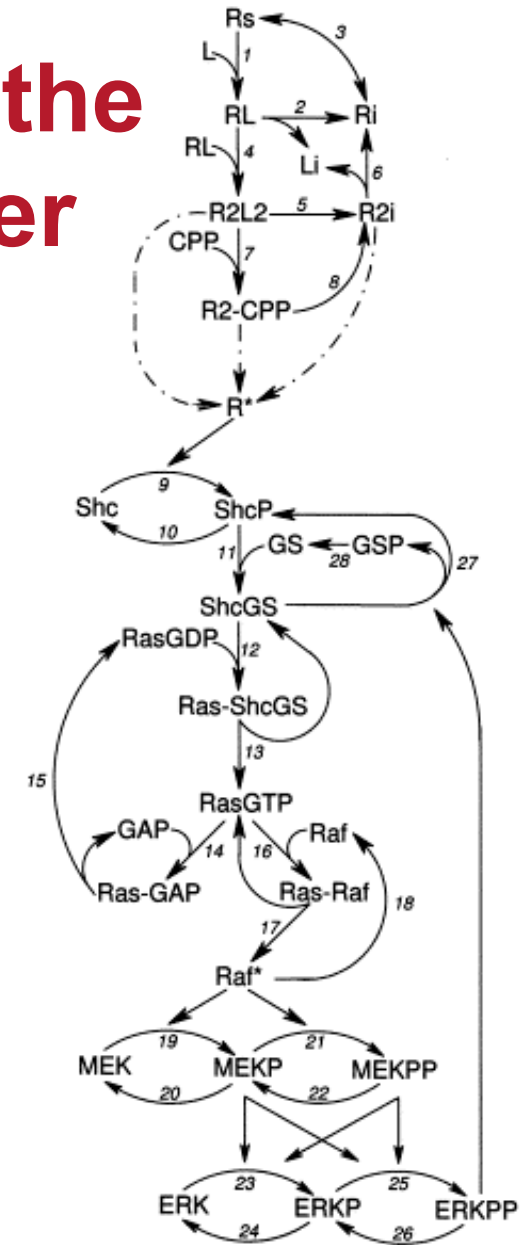
Brightman & Fell, FEBS Lett 2000. "Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells"

BME Model Checking   david.gilbert@brunel.ac.uk

44

Response with EGF vs. NGF signal

+ EGF
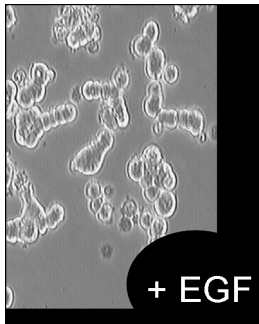
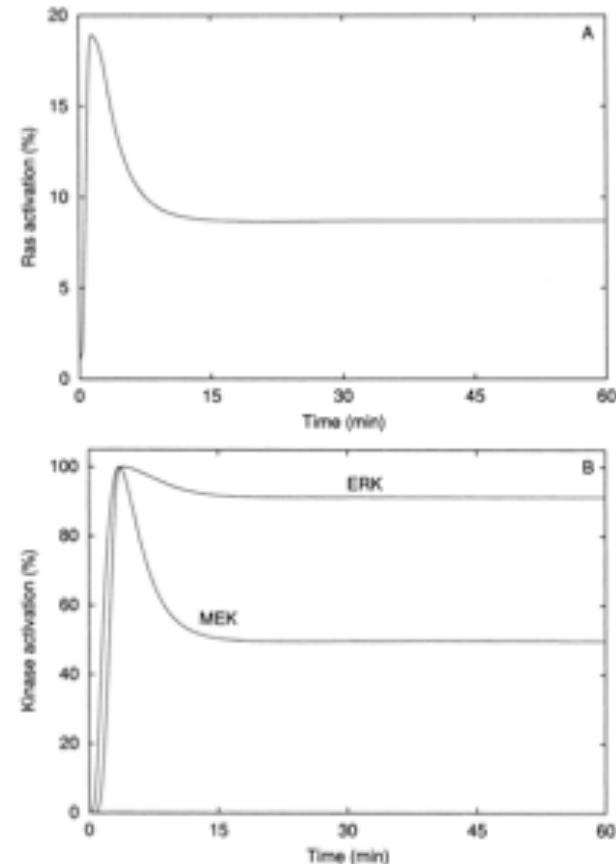Proliferation
(cell division)
PC12 cells

EGF
**Transient activation of
Ras, MEK and ERK**
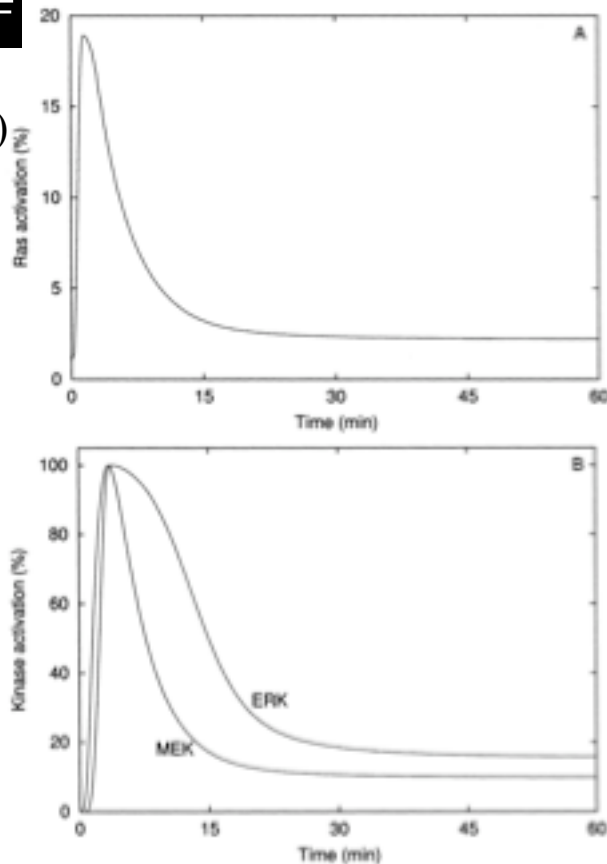
NGF
**Sustained activation
of Ras, MEK and ERK**

+ NGF
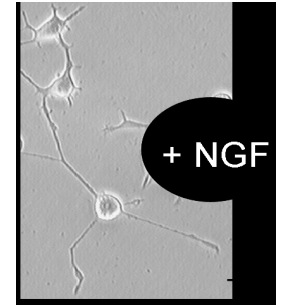
Differentiation
(neurite
outgrowth)

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(active Ras) > 0 U (time ≤ 2 ∧ active Ras ≥ 0.15∗total Ras
∧ active Ras ≤ 0.2∗total Ras ∧ (d(active Ras) < 0)
U (G(active Ras ≥ 0.05∗total Ras ∧ active Ras ≤ 0.10∗total Ras)))]

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

$P_{=?}$ [d(MEKPP) > 0 U (time ≥ 2 ∧ time ≤ 5 ∧ d(MEKPP) < 0
U (G(MEKPP ≥ 0.40∗max(MEKPP) ∧ MEKPP ≤ 0.50∗max(MEKPP))))]

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

$P_{=?}$ [(d(ERKPP) > 0) U (time ≥ 2 ∧ time ≤ 5 ∧ d(ERKPP) < 0
U (G(ERKPP ≥ 0.85 ∗ max(ERKPP))))]

BME Model Checking   david.gilbert@brunel.ac.uk

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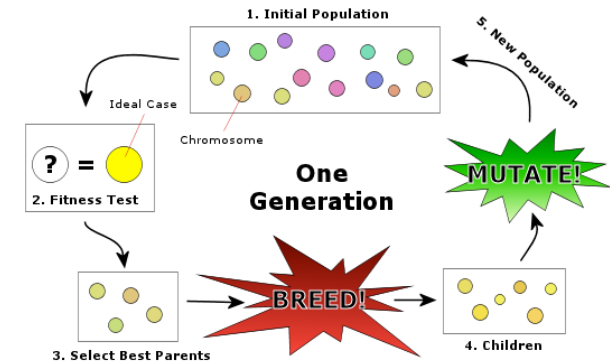
Starting with the EGF (transient) model, construct the
NGF (sustained) model by varying the values of the *critical* kinetic rate constants.
I.e. minimise the distance of the model to sustained behaviour.

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….
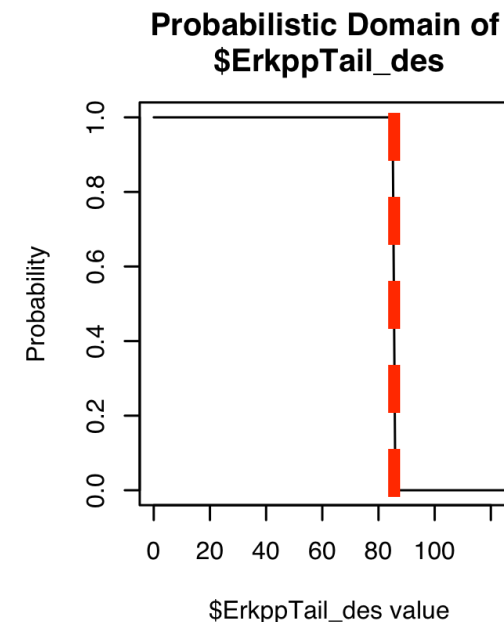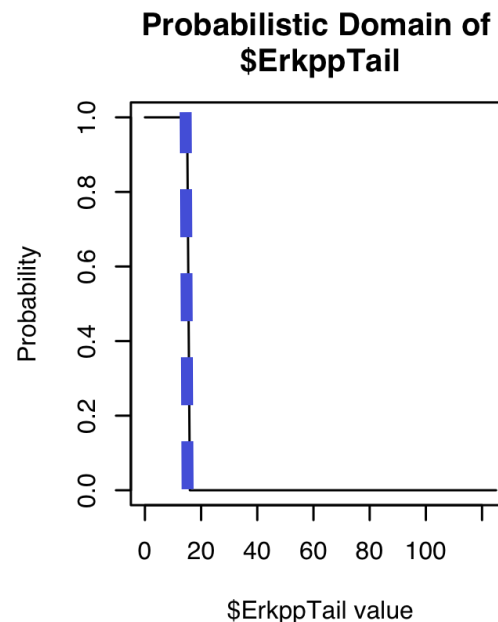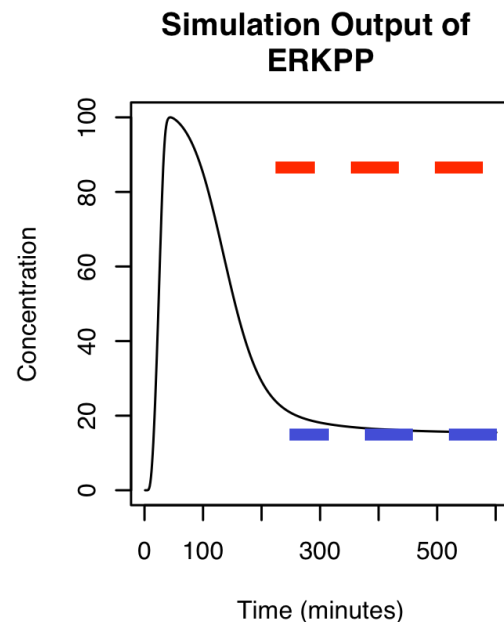- the behaviour in PLTLc enforces a peak at the right position.

Fitness function using PLTLc

Fitness function for sustained ERK

Enforce a peak between time 2 and 5 and characterise the tail in $ErkppTail:

$P=? [(d(ERKPP) > 0) U (time \geq 2 \wedge time \leq 5 \wedge d(ERKPP) < 0$
$\wedge G(ERKPP \geq \$ErkppTail))]$

- Fitness function is distance between tail and 85% of the peak height (sustained activation), calculated using the probabilistic domains:

Simulation Output of ERKPP — Concentration vs Time (minutes)

Probabilistic Domain of $ErkppTail — Probability vs $ErkppTail value

Probabilistic Domain of $ErkppTail_des — Probability vs $ErkppTail_des value

Distance metric

Uses Residual Sum of Squares

Over probabilistic domains in range m,n
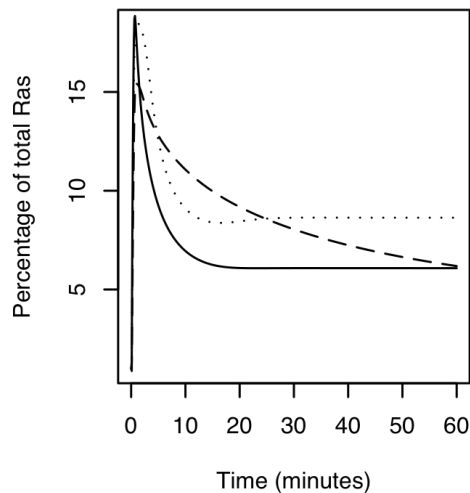
$$RSS(X,X',m,n) = \sum_{i=m}^{n} |\$X(i) - \$X'(i)|^2$$

Dist(Model,Model')=

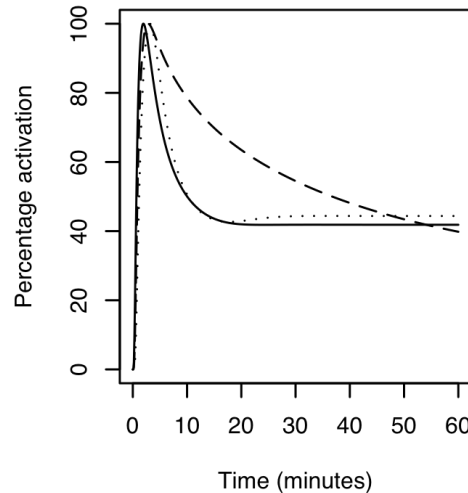   (dist(Ras)+dist(MEKpp)+dist(ERKpp))/3

Results

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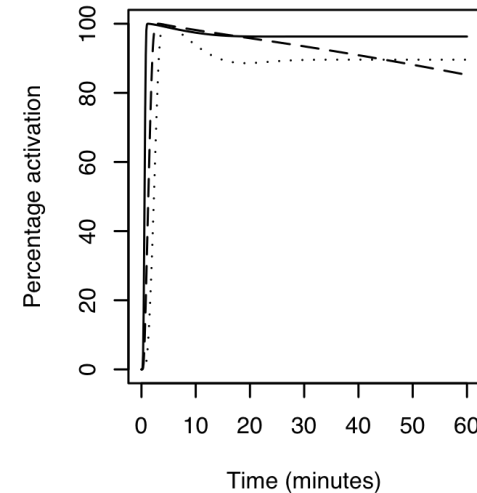- Ran the genetic algorithm with 100 generations with results:

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

BME Model Checking   david.gilbert@brunel.ac.uk

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?"
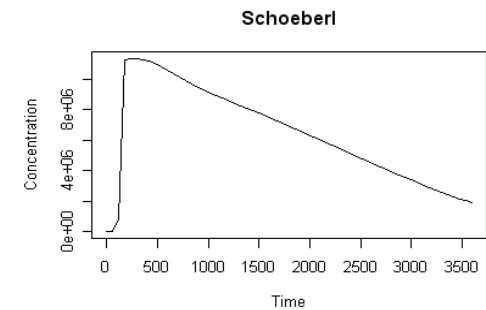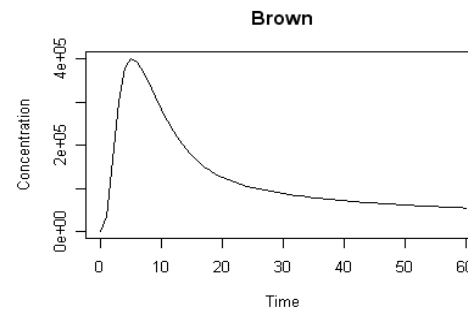 - "What are the behaviours of the pathway not backed up by lab data?"

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*
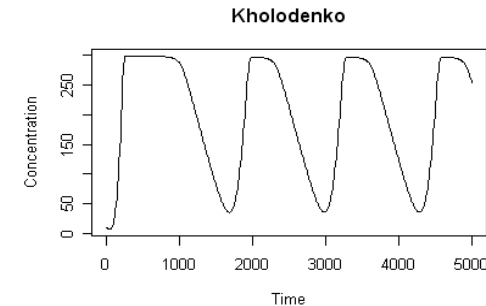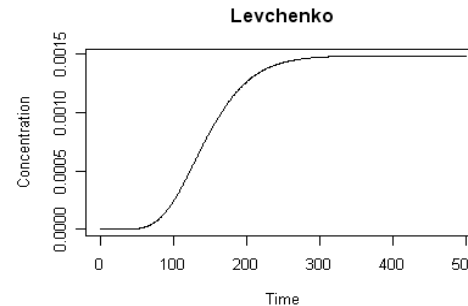
Oscillates at least 4 times

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

- **Kholodenko**

BME Model Checking   david.gilbert@brunel.ac.uk   55

Database Web Interface

MAPK Model Database

Home | MAPK | Models | Search | Links

Menu Home

The MAPK Model Database was developed as part of the SIMAP (Simulation modelling of the MAPK kinase pathway) project, funded by a European Union Framework 6 grant, at the Bioinformatics Research Centre, University of Glasgow.

MAPK Model Database: Originally designed by Richard Orton
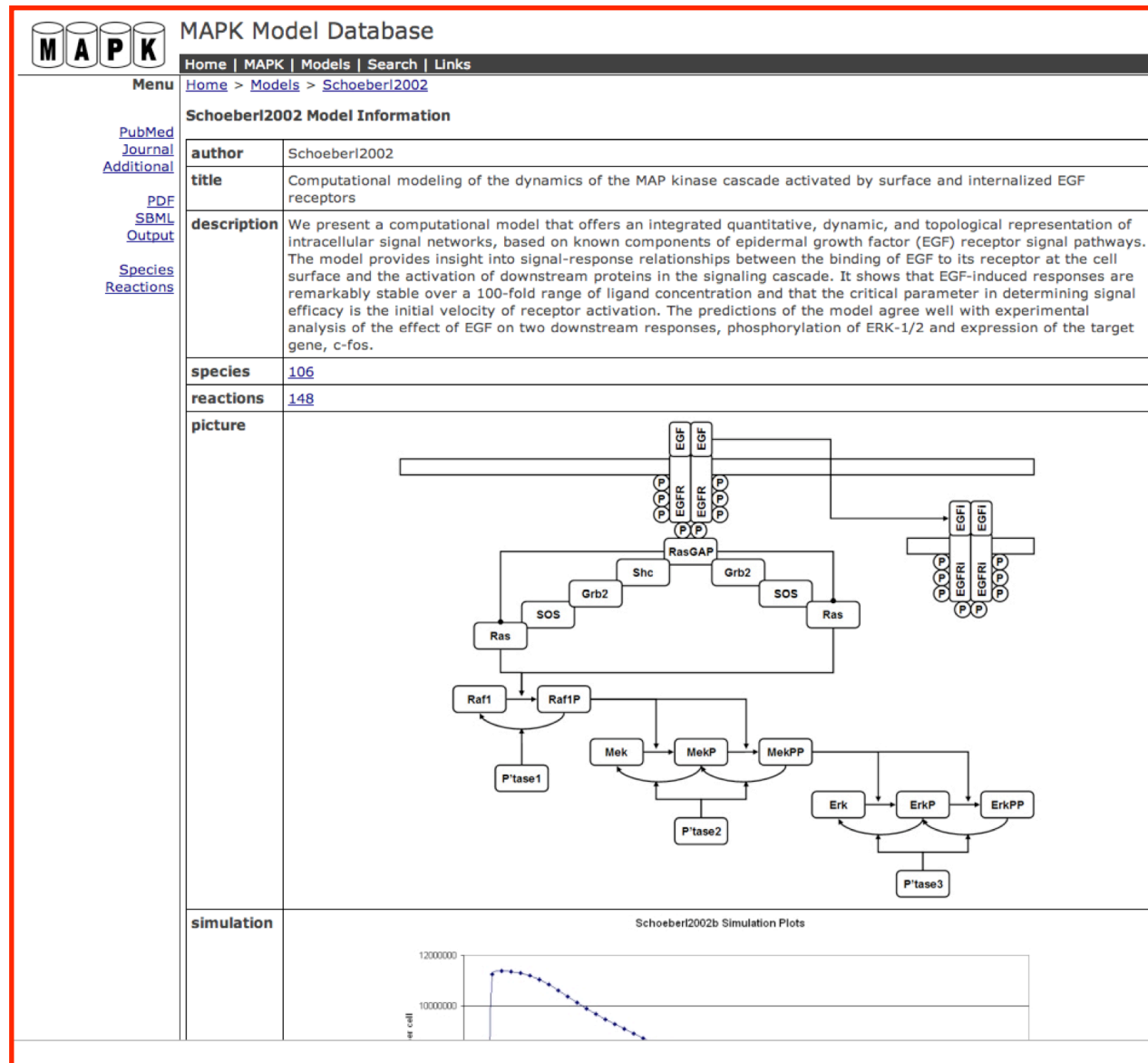
developed and maintained by Qian (Pam) Gao

MAPK

Models – Schoeberl Model (2000)

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

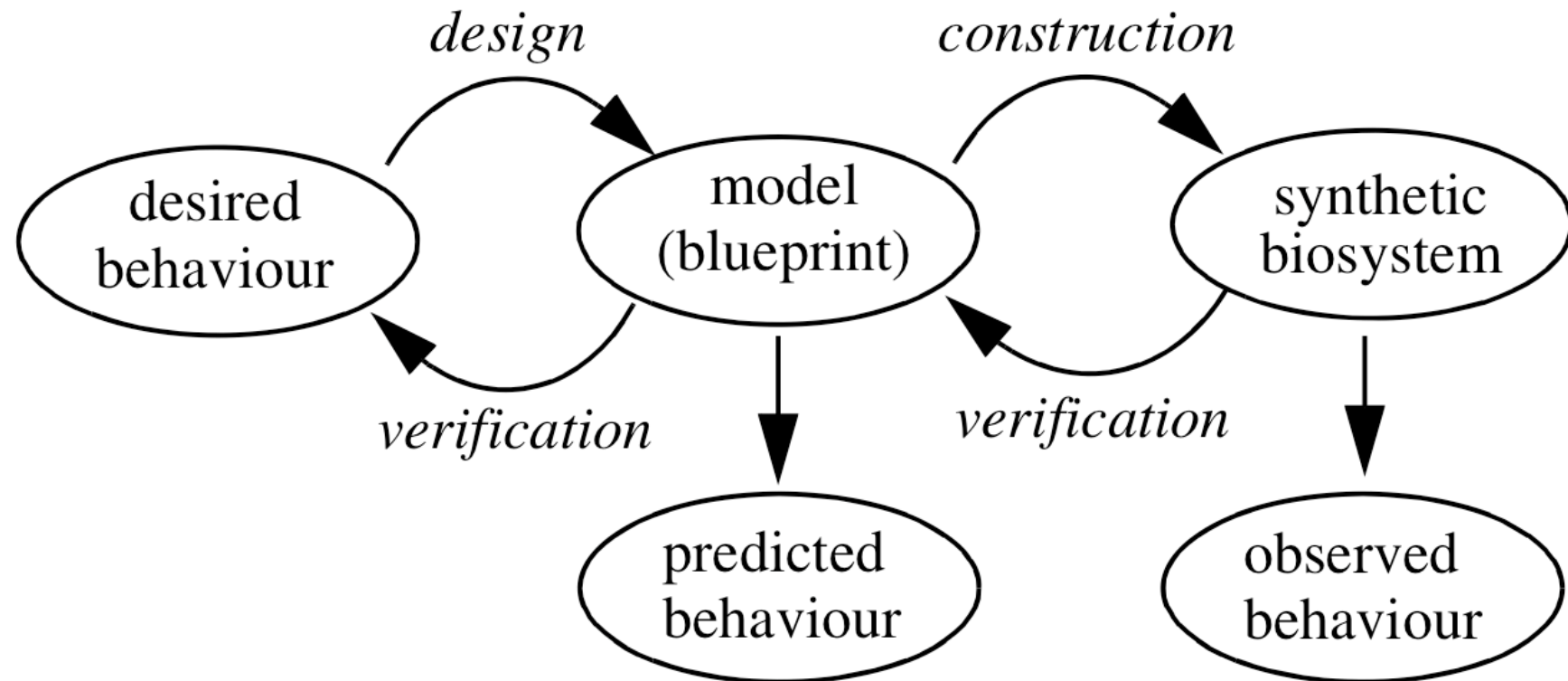 – Model / simulation accurately represent the specifications

Synthetic biology development cycle

Model / check　　　　　　　　　　　　　　　　　**Biosystem**

construct

validate

modify

construct

verify

modify

construct

verify
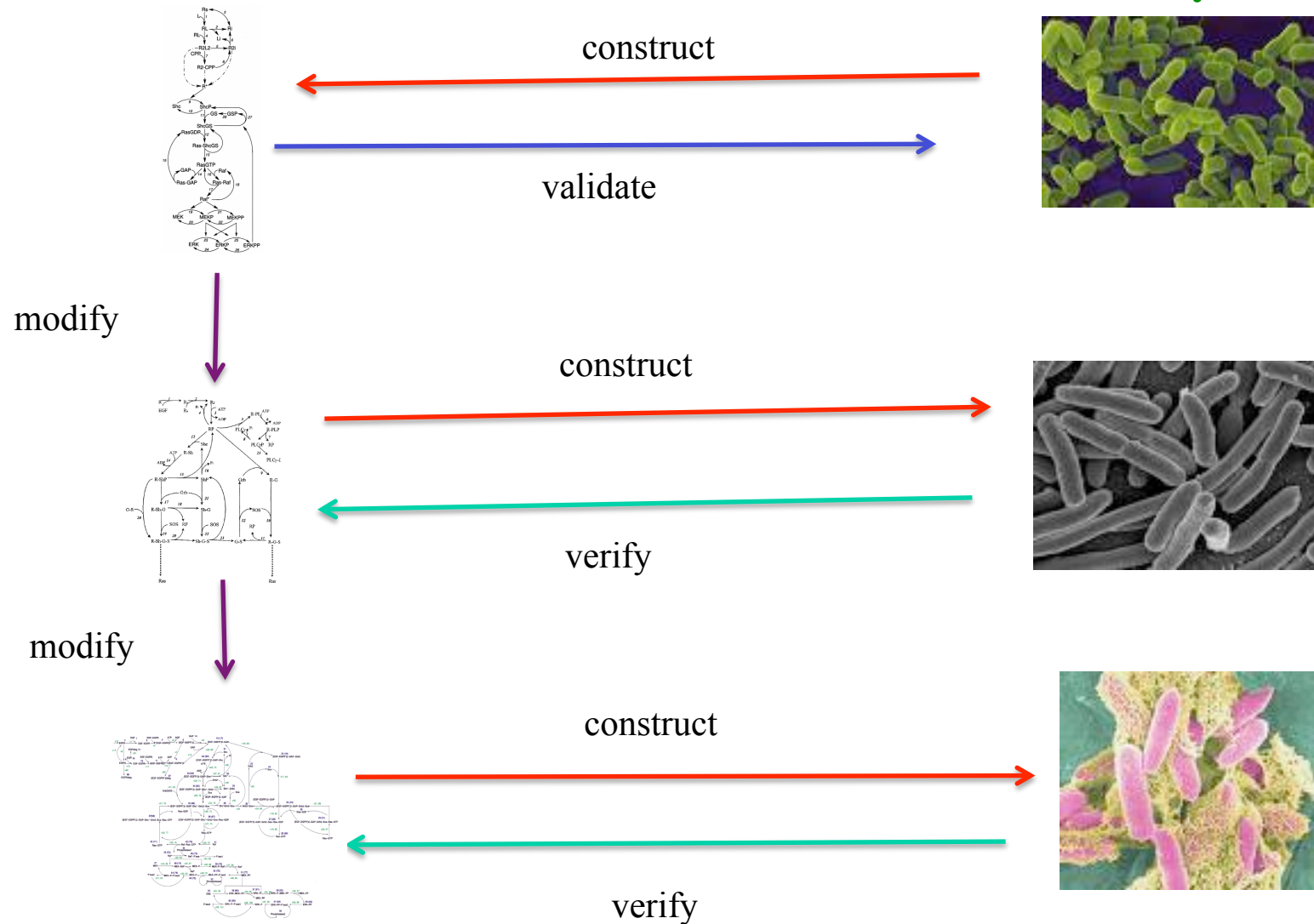
Model checking for Synthetic Biology

Acknowledgements

MC2: Robin Donaldson – University of Glasgow

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

BioNessie(+MC2): Xuan Liu – Brunel University

www.bionessie.org

Some references

Robin Donaldson and David Gilbert (2008). **A Model Checking Approach to the Parameter Estimation of Biochemical Pathways**. CMSB 2008, LNCS 5307/2008, pp269-287.

Monika Heiner, David Gilbert, and Robin Donaldson (2008), **Petri Nets for Systems and Synthetic Biology**. Formal Methods for Systems Biology SFM 2008, LNCS 5016, pp. 215-264

David Gilbert, Monika Heiner and Sebastian Lehrack (2007). **A Unifying Framework for Modelling and Analysing Biochemical Pathways Using Petri Nets** CMSB 2007, LNCS/LNBI 4695, pp. 200-216.

David Gilbert and Monika Heiner, (2006). **From Petri Nets to Differential Equations - an Integrative Approach for Biochemical Network Analysis,** (ATPN06), LNCS 4024 / 2006, pp. 181-200