A Hybrid Approach to Modelling Biological Systems

Francisco J. Romero-Campero[1] Marian Gheorghe[2]
Francesco Bernardini[3] Neil Walkinshaw [2]

[1]Department of Computer Science and Artificial Intelligence, University of Seville

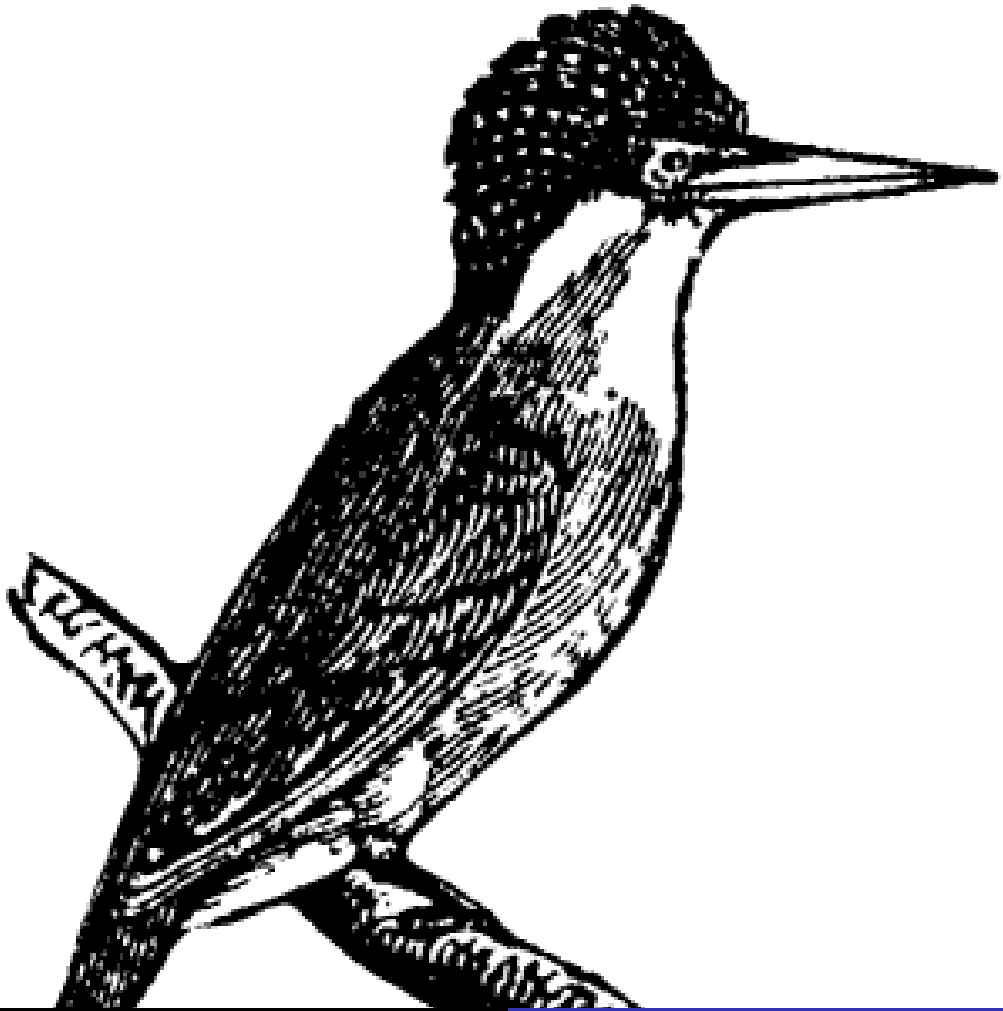
[2]Department of Computer Science, University of Sheffield

[3]Leiden Institute of Advanced Computer Science, Leiden University

WMC8, Thessaloniki 2007

What is this???

What is this???



What is this???

Models and Reality

- **Models are abstractions** of the real-world that highlight some key features while ignoring others that are assumed to be not relevant.
- The use of **models** is intrinsic to any scientific activity.
- A **model** should not be seen or presented as representations of the truth, but instead as a *statement of our current knowledge.*

Models and Reality

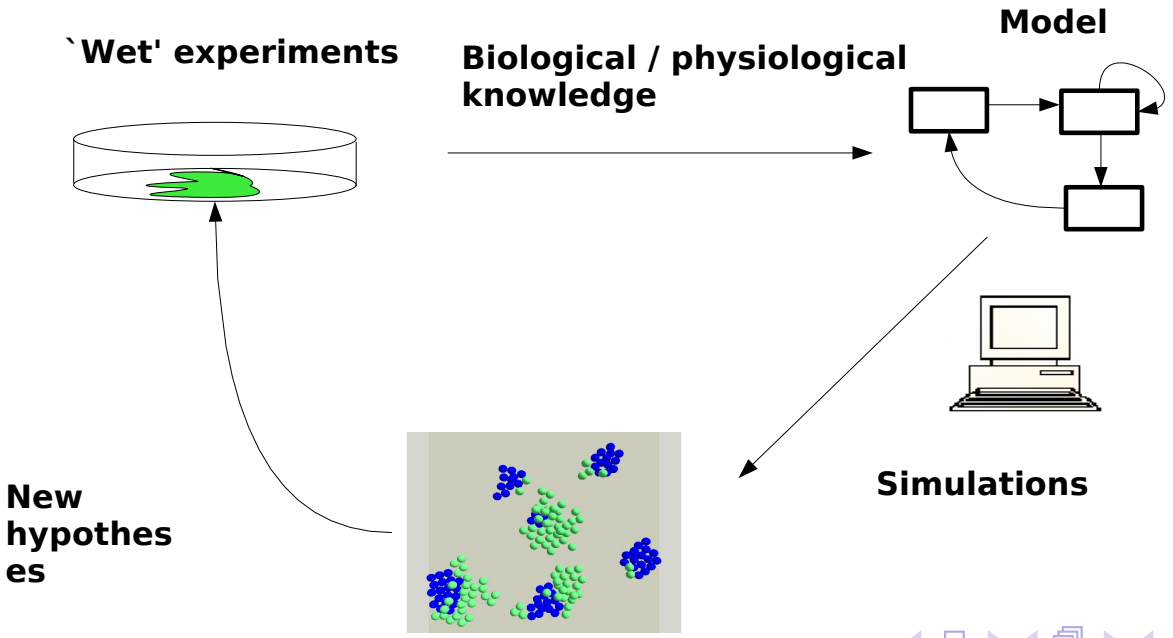
- **Models are abstractions** of the real-world that highlight some key features while ignoring others that are assumed to be not relevant.
- The use of **models** is intrinsic to any scientific activity.
- A **model** should not be seen or presented as representations of the truth, but instead as a *statement of our current knowledge.*

Models and Reality

- **Models are abstractions** of the real-world that highlight some key features while ignoring others that are assumed to be not relevant.
- The use of **models** is intrinsic to any scientific activity.
- A **model** should not be seen or presented as representations of the truth, but instead as a *statement of our current knowledge.*

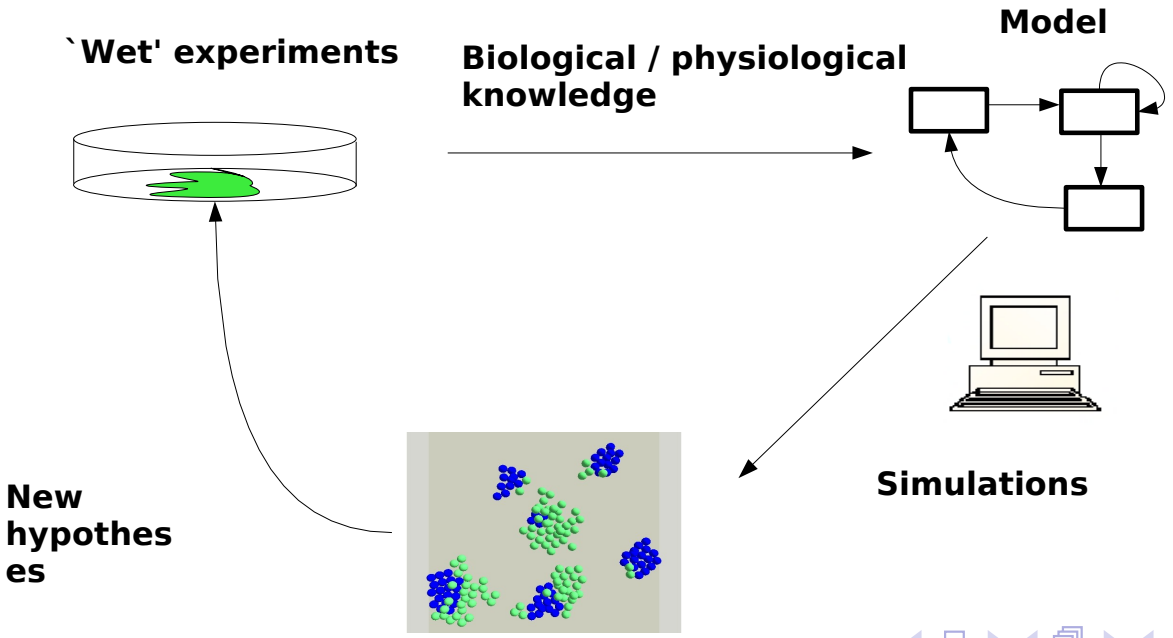
Models and Reality

- There are mainly **three goals** in modelling: understanding, prediction and control.
- A **model** should have at least four properties: relevance, understandability, extensibility and computability.



Models and Reality

- There are mainly **three goals** in modelling: understanding, prediction and control.
- A **model** should have at least four properties: relevance,understandability, extensibility and computability.



Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Modelling Approaches

There exist **many modelling approaches**, each one with its advantages and disadvantages.

- Macroscopic, Microscopic and Mesoscopic
- Quantitative and qualitative
- Discrete and Continuous
- Deterministic and Stochastic
- Top-down or Bottom-up

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The **most widely used approach** in modelling cell systems.
- Macroscopic, continuous and deterministic approach.
- Quantitative and qualitative analysis.

Two key assumptions:

- Homogeneous or well stirred volume: valid only in small volumes
- Continuous and deterministic variation of the concentration: valid only with large number of molecules

Ordinary Differential Equations

- The concentration of each chemical species is represented by a continuous variable $X_i(t)$.
- A differential equation is written to describe the concentration change of each species.
- Most used **kinetic laws**: exponential decay, mass action law and Michaelis-Menten dynamics.

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

K is the Michaelis-Menten constant

Ordinary Differential Equations

- The concentration of each chemical species is represented by a continuous variable $X_i(t)$.
- A differential equation is written to describe the concentration change of each species.
- Most used **kinetic laws**: exponential decay, mass action law and Michaelis-Menten dynamics.

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

K is the Michaelis-Menten constant

Ordinary Differential Equations

- The concentration of each chemical species is represented by a continuous variable $X_i(t)$.
- A differential equation is written to describe the concentration change of each species.
- Most used **kinetic laws**: exponential decay, mass action law and Michaelis-Menten dynamics.

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

K is the Michaelis-Menten constant

Ordinary Differential Equations

- The concentration of each chemical species is represented by a continuous variable $X_i(t)$.
- A differential equation is written to describe the concentration change of each species.
- Most used **kinetic laws**: exponential decay, mass action law and Michaelis-Menten dynamics.

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

K is the Michaelis-Menten constant

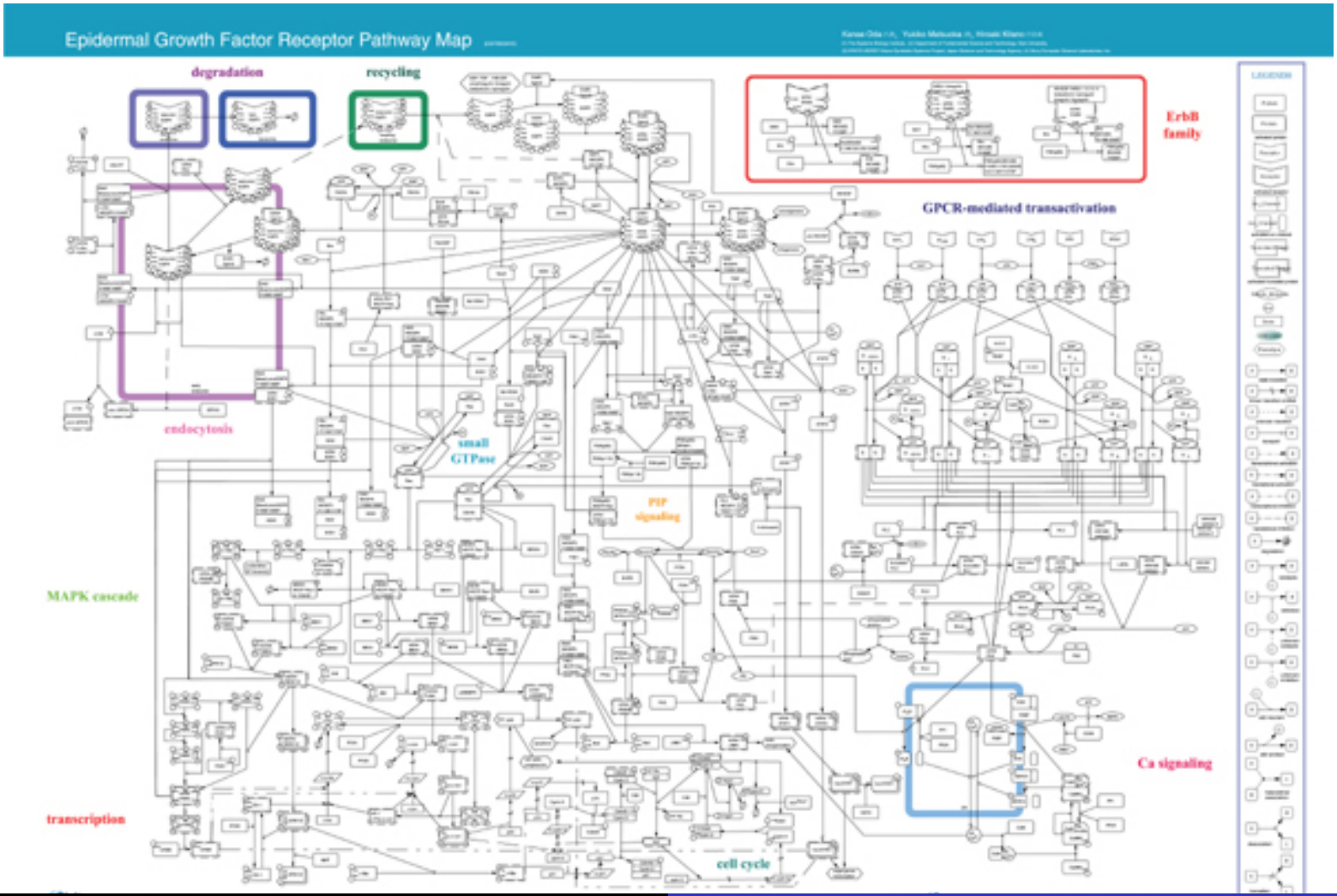
Ordinary Differential Equations

- The concentration of each chemical species is represented by a continuous variable $X_i(t)$.
- A differential equation is written to describe the concentration change of each species.
- Most used **kinetic laws**: exponential decay, mass action law and Michaelis-Menten dynamics.

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

K is the Michaelis-Menten constant

Computational Modelling



Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

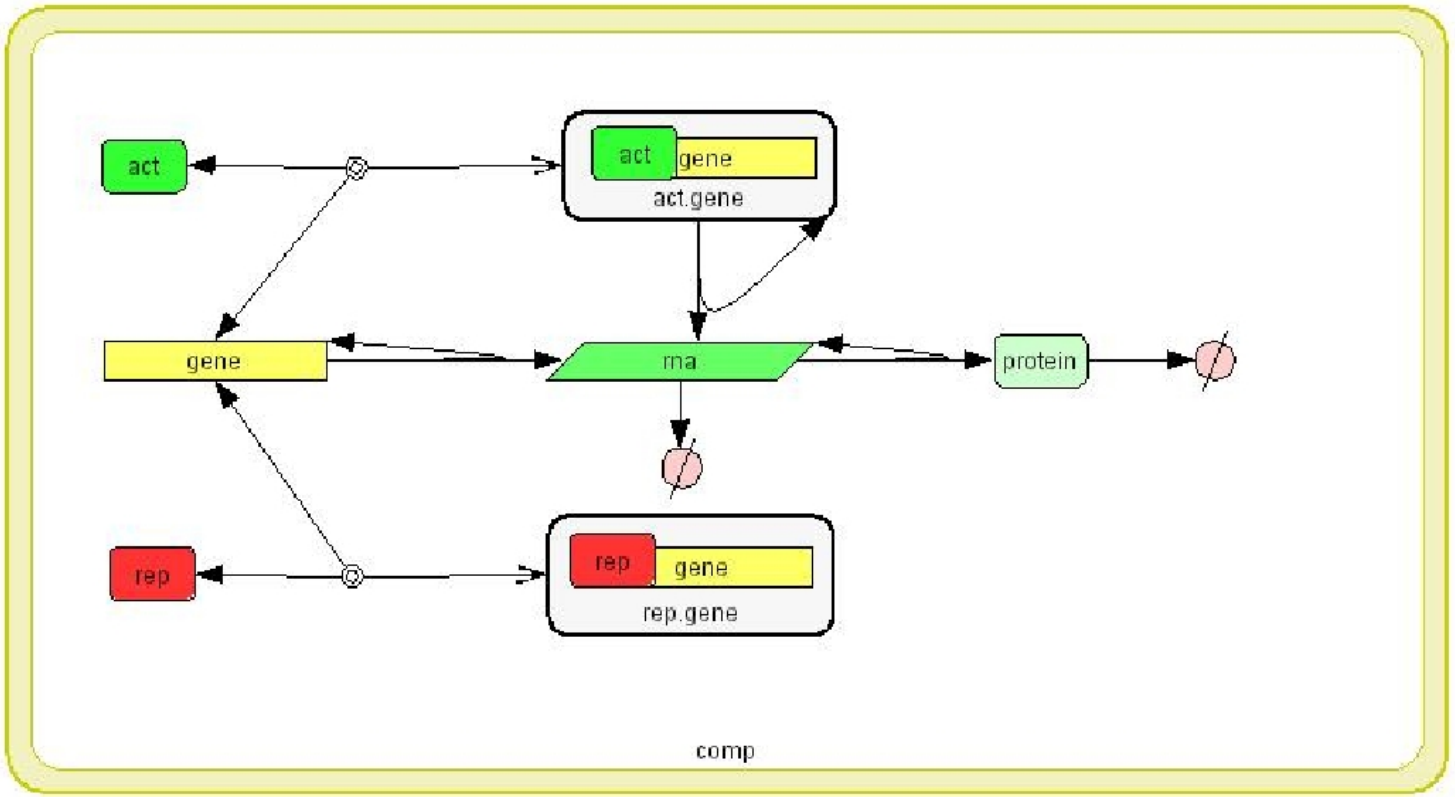
Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

Computational Modelling

- Most computational models were implemented in custom programs.
- Different computational frameworks proposed to model cellular systems:
  - Petri Nets
  - Process Algebras: π-calculus
  - P systems
  - Others: L Systems, Cellular Automata, Agent Based models, ...

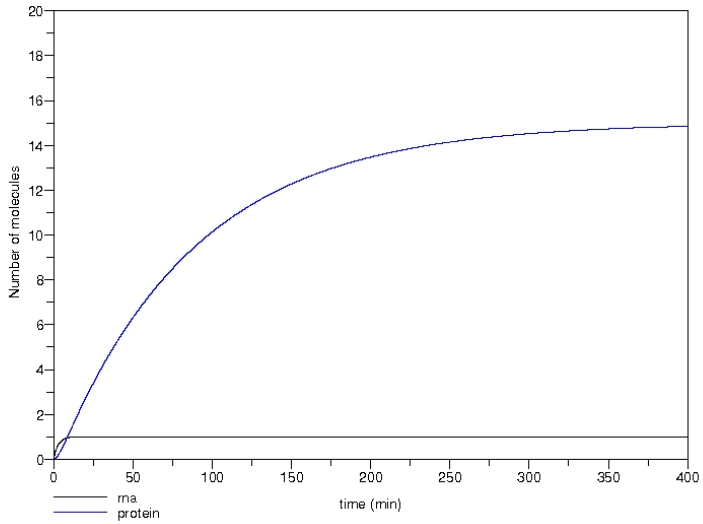
Gene regulation

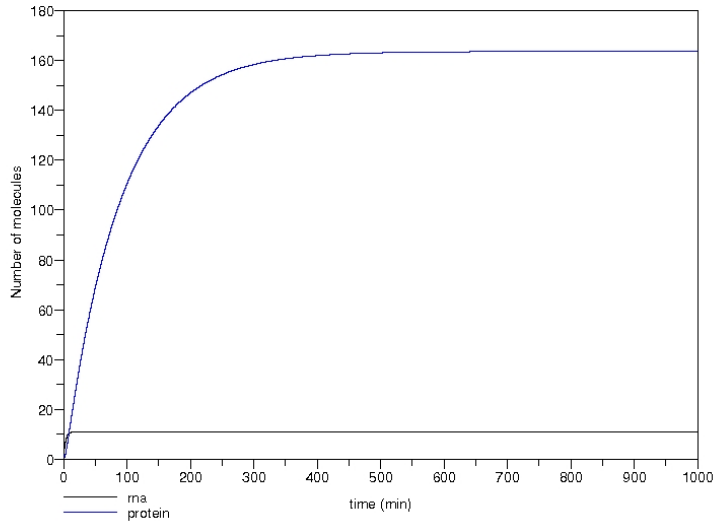


ODE Model

$$\begin{cases} \dfrac{dr}{dt} = c_1 - c_3 r + c_7 \dfrac{act}{act + K} \\ \dfrac{dp}{dt} = c_2 r - c_4 p \end{cases}$$

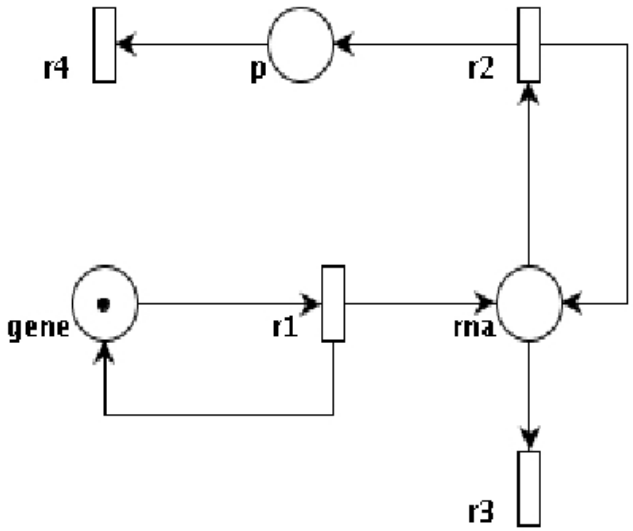
K is the Michaelis-Menten constant





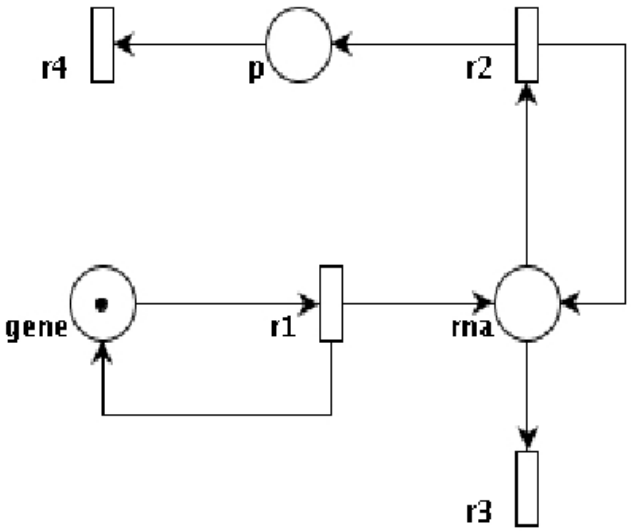
Petri Nets

- A mathematical and computational tool for modelling and analysis of discrete event systems with a concurrent behaviour.
- A formal way to represent the system structure, simulate its behaviour, and prove certain properties of the system.
- A PT-net is a directed graph with two kinds of nodes, places and transitions; arcs, connecting places and transition; and tokens.



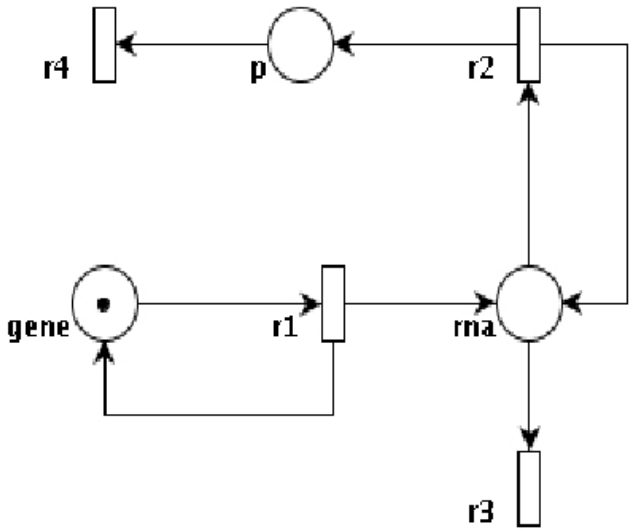
Petri Nets

- A mathematical and computational tool for modelling and analysis of discrete event systems with a concurrent behaviour.
- A formal way to represent the system structure, simulate its behaviour, and prove certain properties of the system.
- A PT-net is a directed graph with two kinds of nodes, places and transitions; arcs, connecting places and transition; and tokens.



Petri Nets

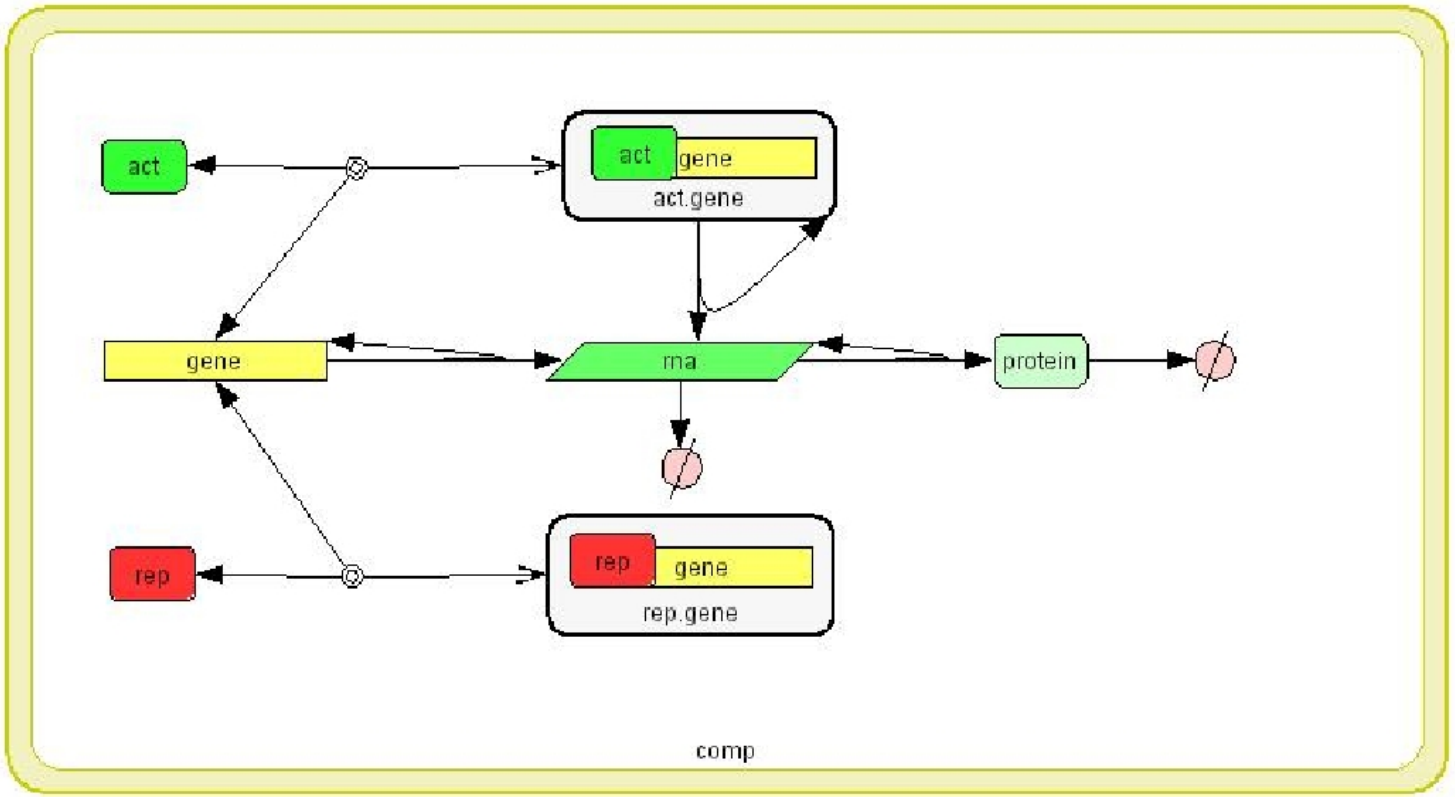
- A mathematical and computational tool for modelling and analysis of discrete event systems with a concurrent behaviour.
- A formal way to represent the system structure, simulate its behaviour, and prove certain properties of the system.
- A PT-net is a directed graph with two kinds of nodes, places and transitions; arcs, connecting places and transition; and tokens.



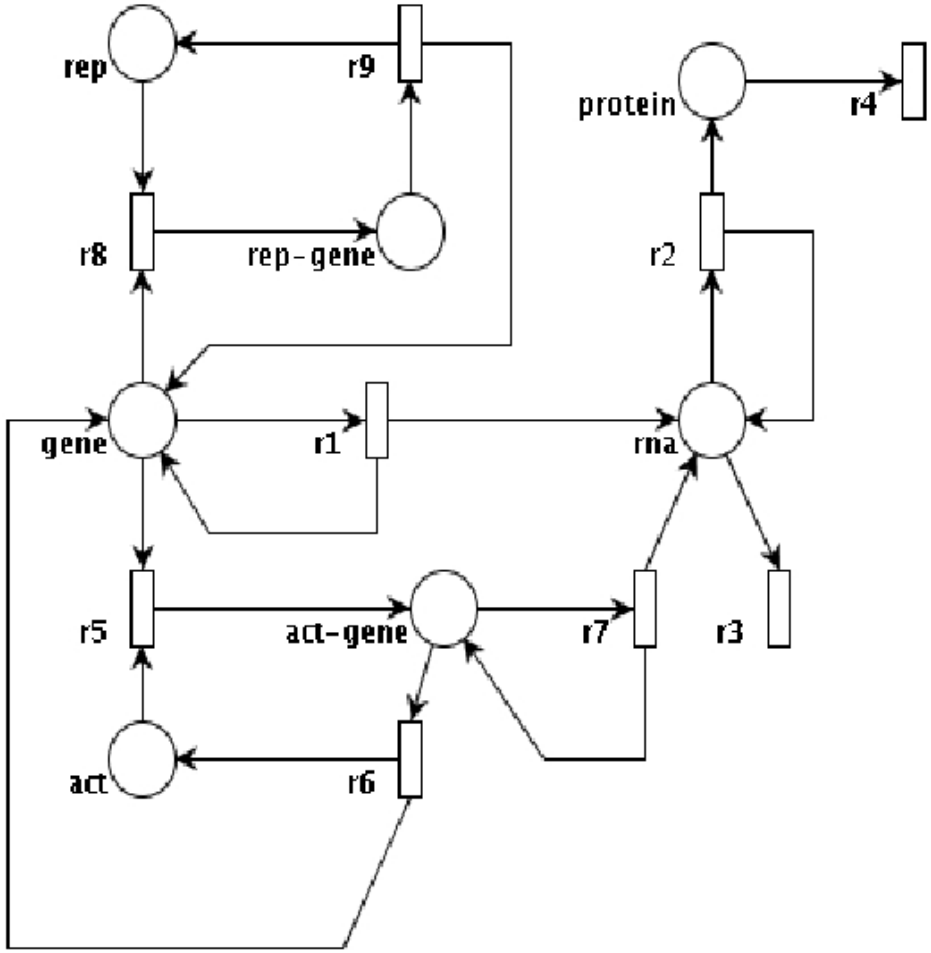
Petri Net Modelling Principles

Biochemistry	**PT-net**
Molecule	Place
Molecular Population	Marking
Biochemical Transformation	Transition

Goss, P.J.E., Peccoud, J. (1998) Quantitative Modeling of Stochastic Systems in Molecular Biology using Stochastic Petri Nets., *Proc. Natl. Acad. Sci. USA*, **95** 6750–6755.


act
act
gene
act.gene
gene
ma
protein
rep
rep
gene
rep.gene
comp

Petri Net Model



Process Algebras: π-calculus

- Introduced as a formal language to describe mobile concurrent processes and systems with dynamically evolving communication topology.
- The π-calculus has a **simple semantics** and a **tractable algebraic theory**.
- The process expressions are defined by **guarded processes**, parallel composition $P|Q$, nondeterministic choice $P + Q$, replication $!P$, and a restriction operator $(\nu x)P$ creating a local fresh channel x for a process P.

Process Algebras: π-calculus

- Introduced as a formal language to describe mobile concurrent processes and systems with dynamically evolving communication topology.
- The π-calculus has a **simple semantics** and a **tractable algebraic theory**.
- The process expressions are defined by **guarded processes**, parallel composition $P|Q$, nondeterministic choice $P + Q$, replication $!P$, and a restriction operator $(\nu x)P$ creating a local fresh channel x for a process P.

Process Algebras: π-calculus

- Introduced as a formal language to describe mobile concurrent processes and systems with dynamically evolving communication topology.
- The π-calculus has a **simple semantics** and a **tractable algebraic theory**.
- The process expressions are defined by **guarded processes**, parallel composition $P|Q$, nondeterministic choice $P + Q$, replication $!P$, and a restriction operator $(\nu x)P$ creating a local fresh channel x for a process P.

π-calculus Modelling Principles

Biochemistry	**π-calculus**
Molecule	Process
Molecular Population	Systems of communicating processes
Biochemical Transformation	Communication channel
Compartment	Private communication channel
Compartment Translocation	Extrusion of a private channel's scope

Regev, A., Shapiro, E. (2004) The π-calculus as an Abstraction for Biomolecular Systems. In Ciobanu, G., Rozenberg, G. (eds.) *Modelling in Molecular Biology*. Springer Verlag.

π-calculus model

Initial processes: $S_{0,1} = gene$; $S_{0,2} = gene \mid act \mid \ldots \mid act$ and $S_{0,3} = gene \mid rep \mid \ldots \mid rep$

Processes:

$gene := \tau_{c_1}.(\, gene \mid rna \,) + a_{c_5}?.act\text{-}gene + r_{c_8}?.rep\text{-}gene$

$rna := \tau_{c_2}.(\, rna \mid protein \,) + \tau_{c_3}.0$

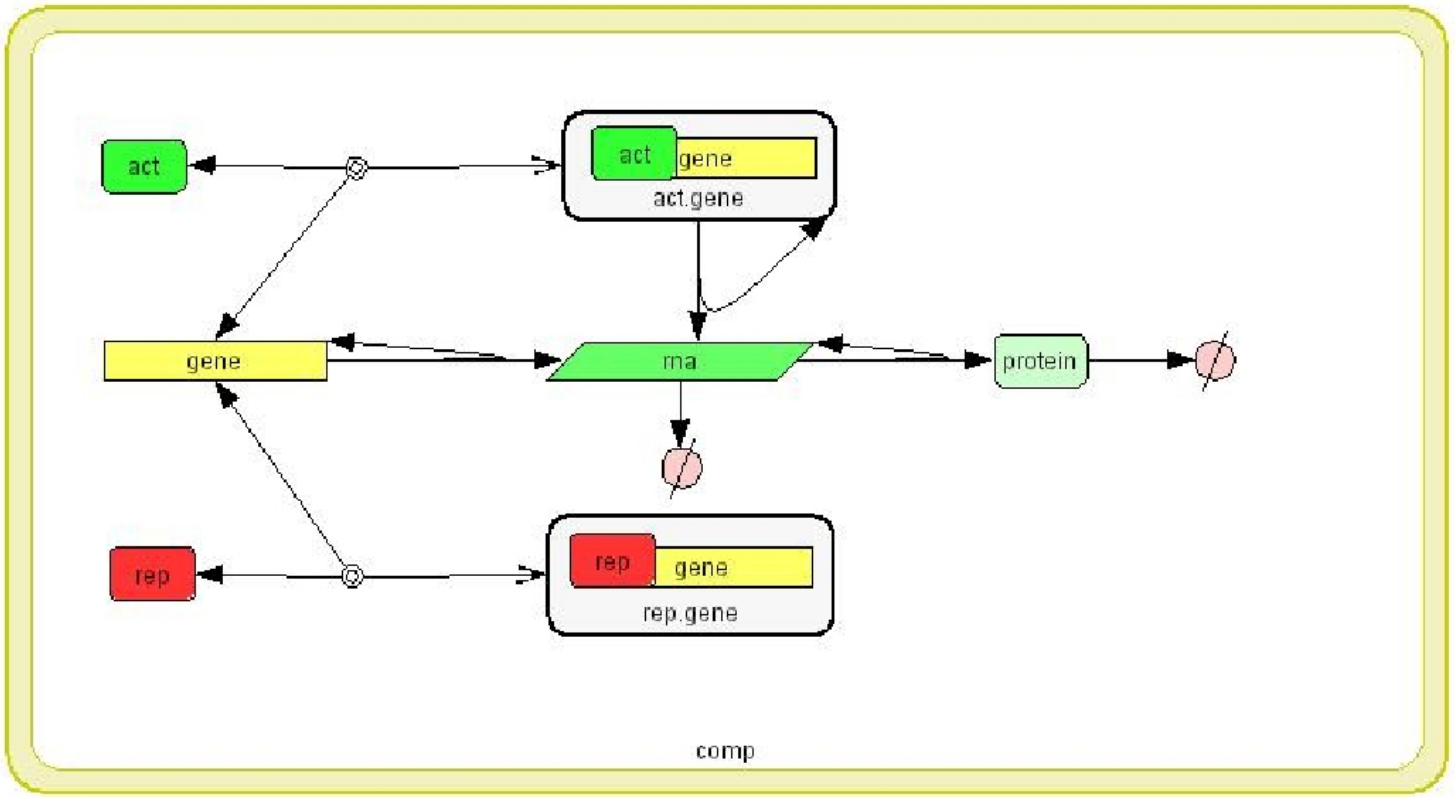
$protein := \tau_{c_4}.0$

$act := a_{c_5}!.0$

$act\text{-}gene := \tau_{c_6}.(\, act \mid gene \,) + \tau_{c_7}.(\, act\text{-}gene \mid rna \,)$

$rep := r_{c_8}!.0$

$rep\text{-}gene := \tau_{c_9}.(\, rep \mid gene \,)$


act
act
gene
act.gene
gene
ma
protein
rep
rep
gene
rep.gene
comp

Membrane Computing: P Systems

- **Inspired directly** from the structure and functioning of the living cells.
- **Distributed and parallel rewritting systems** *compartmentalised hierarchical structures*.
- A **formal framework** for the specification and simulation of cellular systems which integrates structural and dynamic aspects in a comprehensive and relevant way while keeping mathematical and computational tractability.

Membrane Computing: P Systems

- **Inspired directly** from the structure and functioning of the living cells.
- **Distributed and parallel rewritting systems** *compartmentalised hierarchical structures*.
- A **formal framework** for the specification and simulation of cellular systems which integrates structural and dynamic aspects in a comprehensive and relevant way while keeping mathematical and computational tractability.

Membrane Computing: P Systems

- **Inspired directly** from the structure and functioning of the living cells.
- **Distributed and parallel rewritting systems** *compartmentalised hierarchical structures*.
- A **formal framework** for the specification and simulation of cellular systems which integrates structural and dynamic aspects in a comprehensive and relevant way while keeping mathematical and computational tractability.

P System Modelling Principles

Biochemistry	**P System**
Compartment	Region defined by a membrane
Molecule	Object
Molecular Population	Multiset of objects
Biochemical Transformation	Rewriting rule
Compartment Translocation	Boundary rule

Pérez-Jiménez, M.J., Romero-Campero, F.J. Modelling Gene Expression Control Using P Systems: The Lac Operon, A Case Study, BioSystems, to appear.

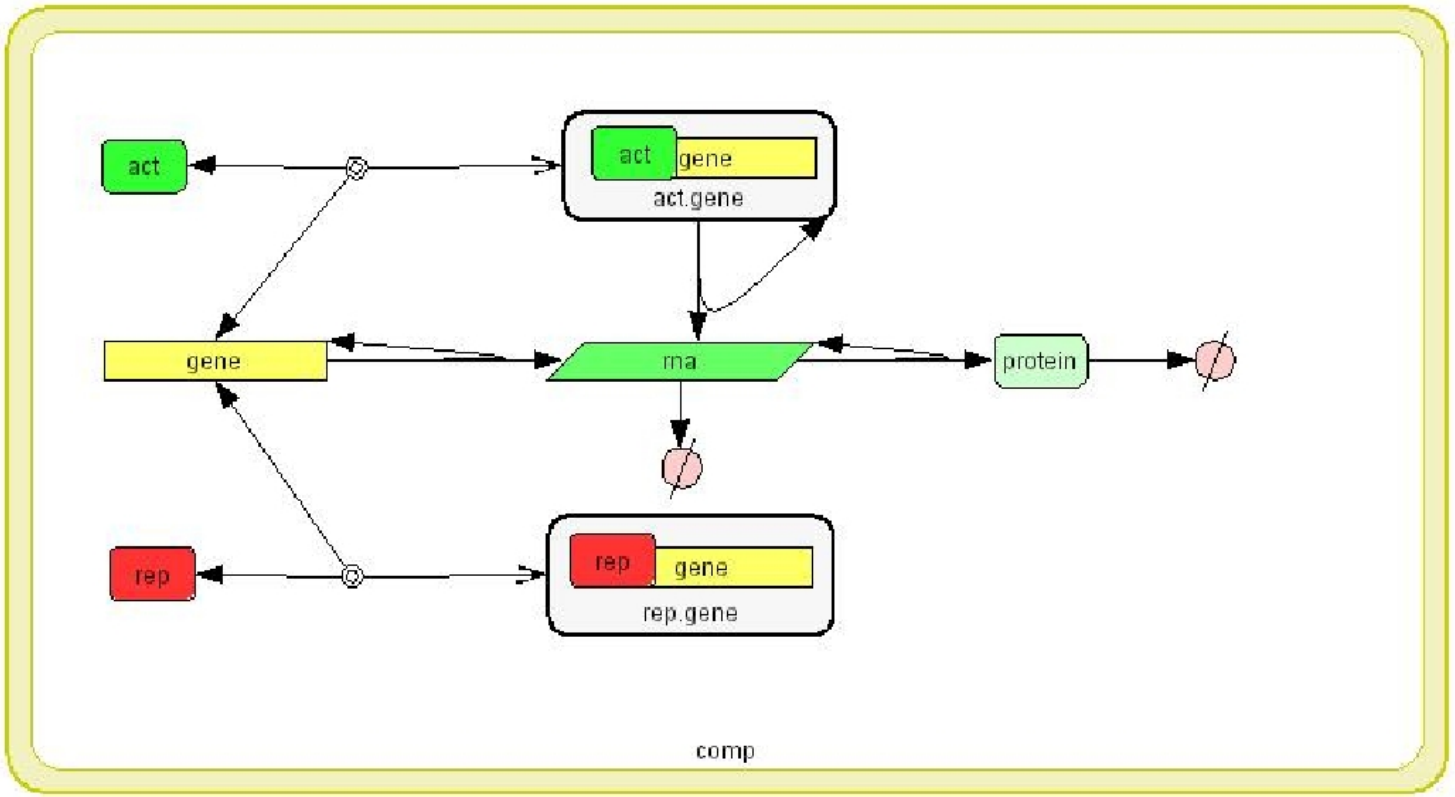
P System Model

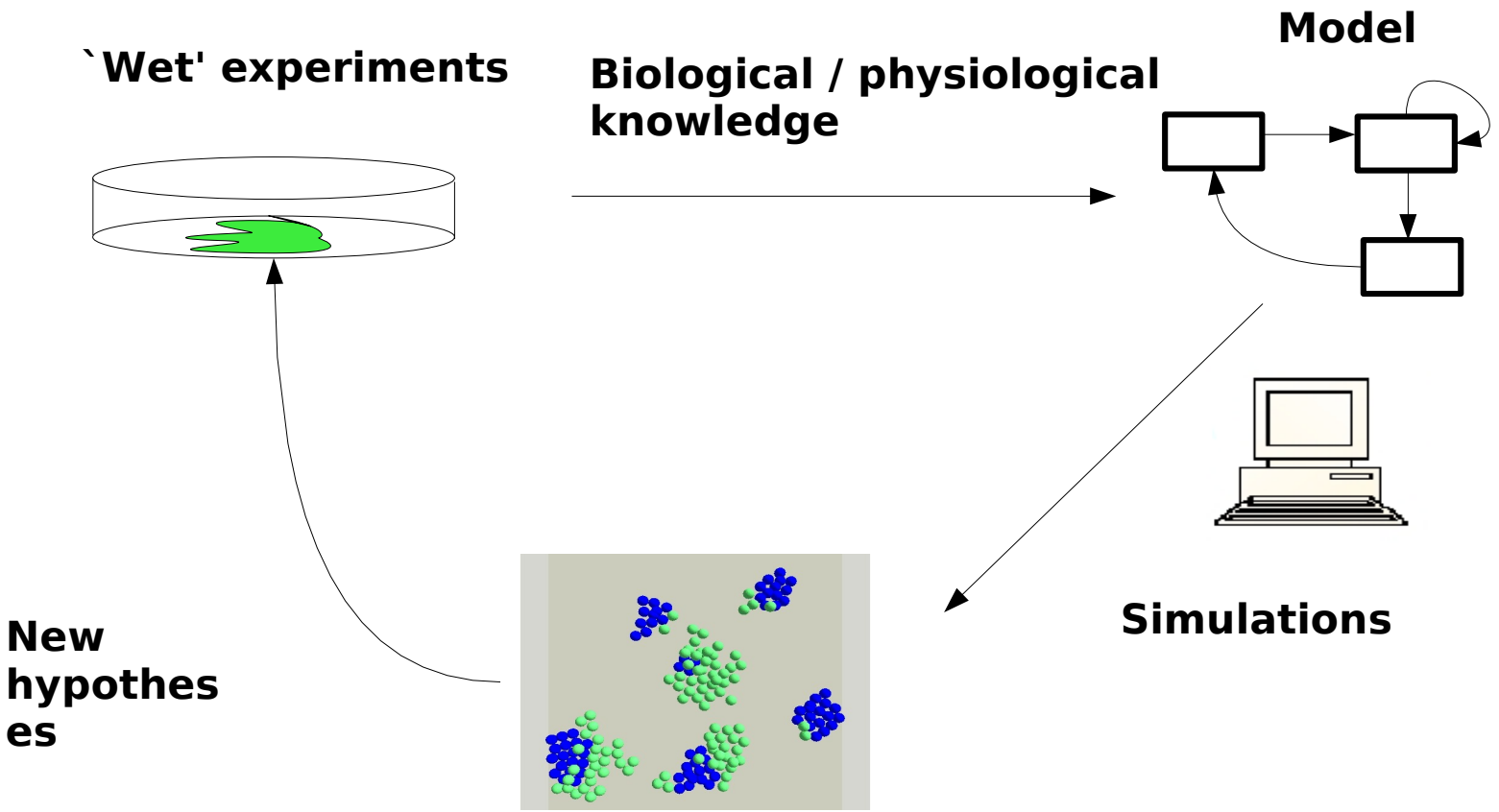
$\Pi = (\{gene, rna, protein, act, rep, act\text{-}gene, rep\text{-}gene\}, \{b\}, [\ ]_b, (b, M_i, \emptyset), \{r_1, \ldots, r_9\})$

Initial multisets: $M_{0,1} = gene$; $M_{0,2} = gene + act... + act$ and $M_{0,3} = gene + rep... + rep$

Rules:

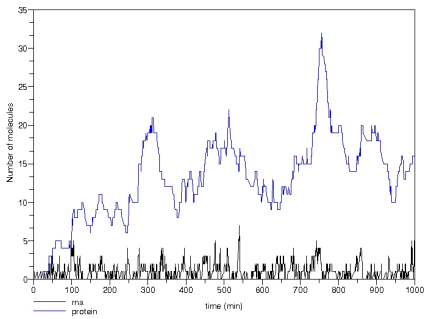
$$r_1 : [\ gene\ ]_b \xrightarrow{c_1} [\ gene + rna\ ]_b \quad c_1 = 0.347\ min^{-1}$$
$$r_2 : [\ rna\ ]_b \xrightarrow{c_2} [\ rna + protein\ ]_b \quad c_2 = 0.174\ min^{-1}$$
$$r_3 : [\ rna\ ]_b \xrightarrow{c_3} [\ ]_b \quad c_3 = 0.347\ min^{-1}$$
$$r_4 : [\ protein\ ]_b \xrightarrow{c_4} [\ ]_b \quad c_4 = 0.0116\ min^{-1}$$
$$r_5 : [\ act + gene\ ]_b \xrightarrow{c_5} [\ act\text{-}gene\ ]_b \quad c_5 = 6.6412087\ molec^{-1}\, min^{-1}$$
$$r_6 : [\ act\text{-}gene\ ]_b \xrightarrow{c_6} [\ act + gene\ ]_b \quad c_6 = 0.6\ s^{-1}$$
$$r_7 : [\ act\text{-}gene\ ]_b \xrightarrow{c_7} [\ act\text{-}gene + rna\ ]_b \quad c_7 = 3.47\ min^{-1}$$
$$r_8 : [\ rep + gene\ ]_b \xrightarrow{c_8} [\ rep\text{-}gene\ ]_b \quad c_8 = 6.6412087\ molec^{-1}\, min^{-1}$$
$$r_9 : [\ rep\text{-}gene\ ]_b \xrightarrow{c_9} [\ rep + gene\ ]_b \quad c_9 = 0.6\ min^{-1}$$

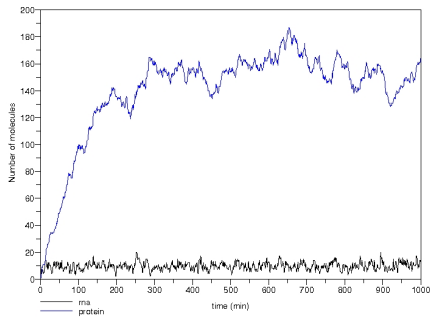

act
act
gene
act.gene
gene
ma
protein
rep
rep
gene
rep.gene
comp

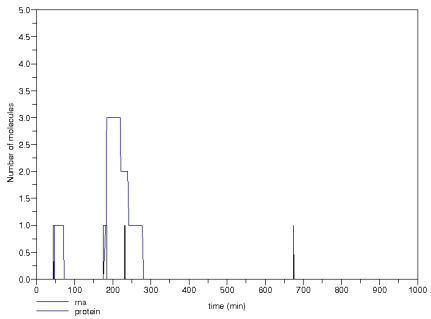

`Wet' experiments
Biological / physiological knowledge
Model
Simulations
New hypothes es

Stochastic Variants

- In the original approaches the transitions / communications / rules are applied in a nondeterministic parallel way. This produces two inaccuracies:
  - **Reactions** do not take place at the correct **rate**
  - All time steps are equal and do not represent the **time evolution** of the real system
- To get around this problem, Stochastic Simulation Algorithm (SSA) or Gillespie algorithm is used.
- A stochastic constant is associated with each transition / communication channel / rule.

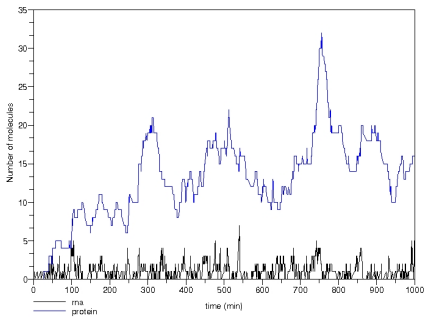


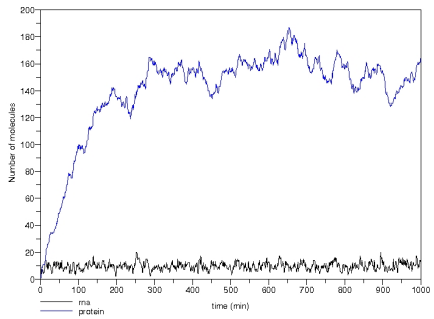


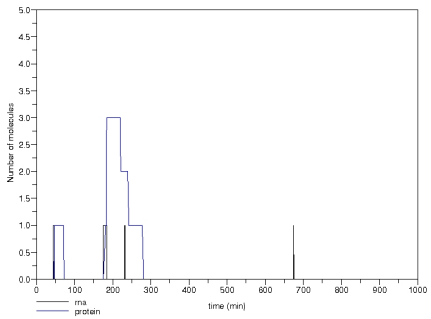


Stochastic Variants

- In the original approaches the transitions / communications / rules are applied in a nondeterministic parallel way. This produces two inaccuracies:
  - **Reactions** do not take place at the correct **rate**
  - All time steps are equal and do not represent the **time evolution** of the real system
- To get around this problem, Stochastic Simulation Algorithm (SSA) or Gillespie algorithm is used.
- A stochastic constant is associated with each transition / communication channel / rule.

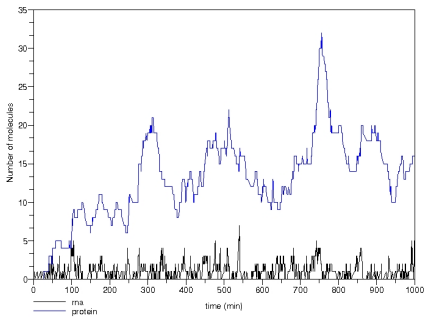


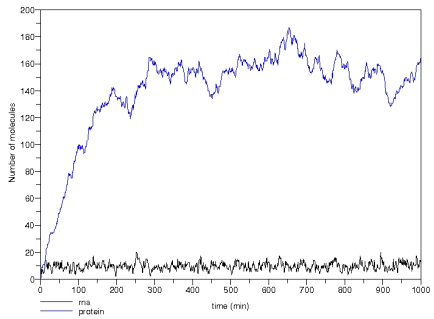


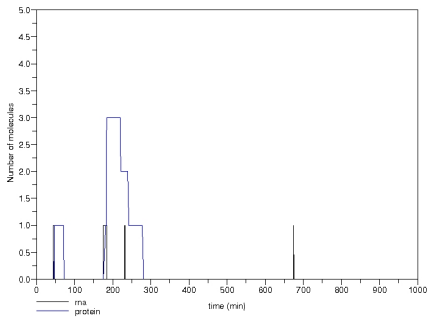


Stochastic Variants

- In the original approaches the transitions / communications / rules are applied in a nondeterministic parallel way. This produces two inaccuracies:
  - **Reactions** do not take place at the correct **rate**
  - All time steps are equal and do not represent the **time evolution** of the real system
- To get around this problem, Stochastic Simulation Algorithm (SSA) or Gillespie algorithm is used.
- A stochastic constant is associated with each transition / communication channel / rule.

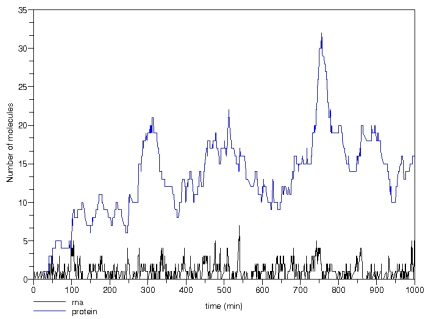


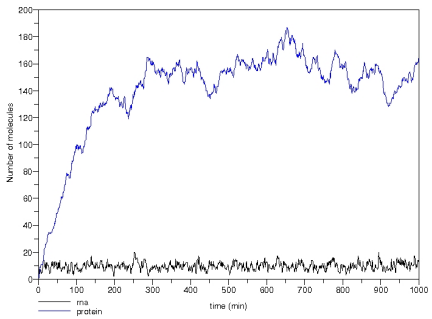


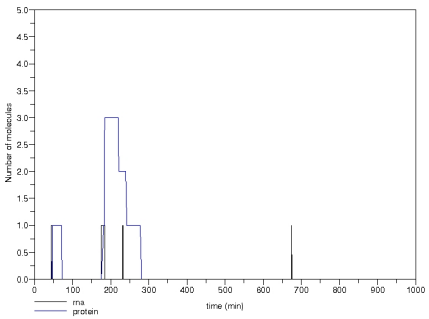


Stochastic Variants

- In the original approaches the transitions / communications / rules are applied in a nondeterministic parallel way. This produces two inaccuracies:
  - **Reactions** do not take place at the correct **rate**
  - All time steps are equal and do not represent the **time evolution** of the real system
- To get around this problem, Stochastic Simulation Algorithm (SSA) or Gillespie algorithm is used.
- A stochastic constant is associated with each transition / communication channel / rule.

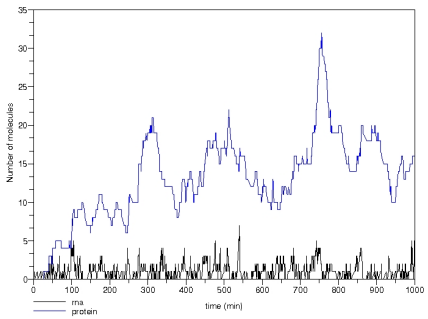


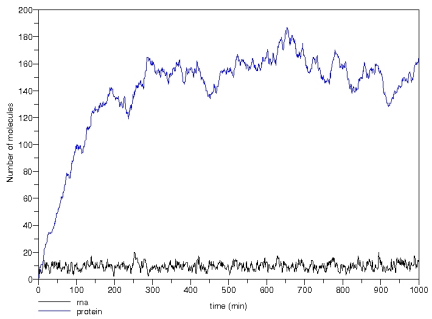


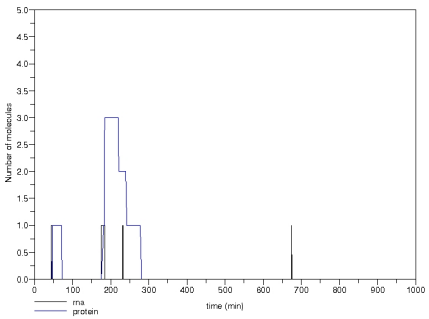


Stochastic Variants

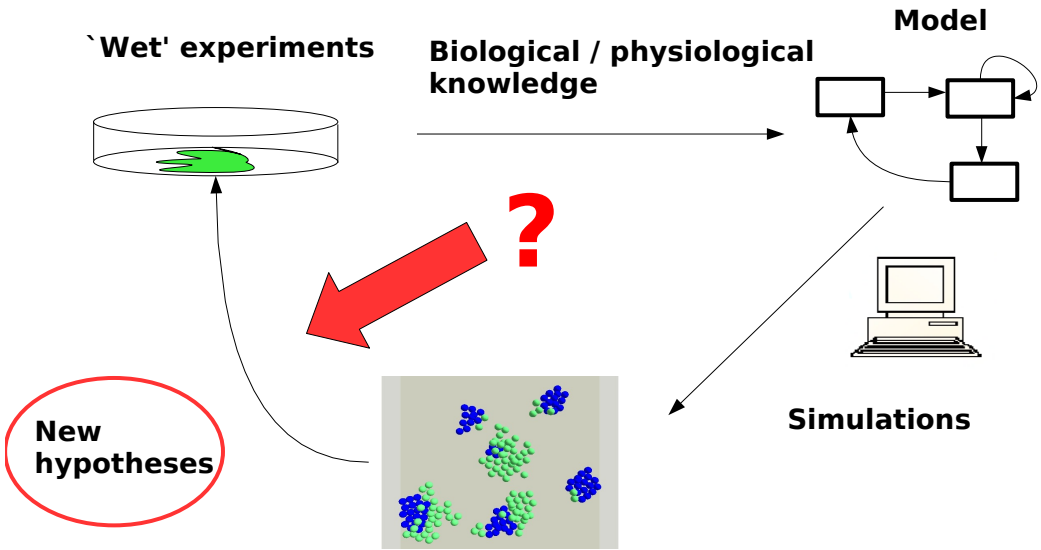
- In the original approaches the transitions / communications / rules are applied in a nondeterministic parallel way. This produces two inaccuracies:
  - **Reactions** do not take place at the correct **rate**
  - All time steps are equal and do not represent the **time evolution** of the real system
- To get around this problem, Stochastic Simulation Algorithm (SSA) or Gillespie algorithm is used.
- A stochastic constant is associated with each transition / communication channel / rule.





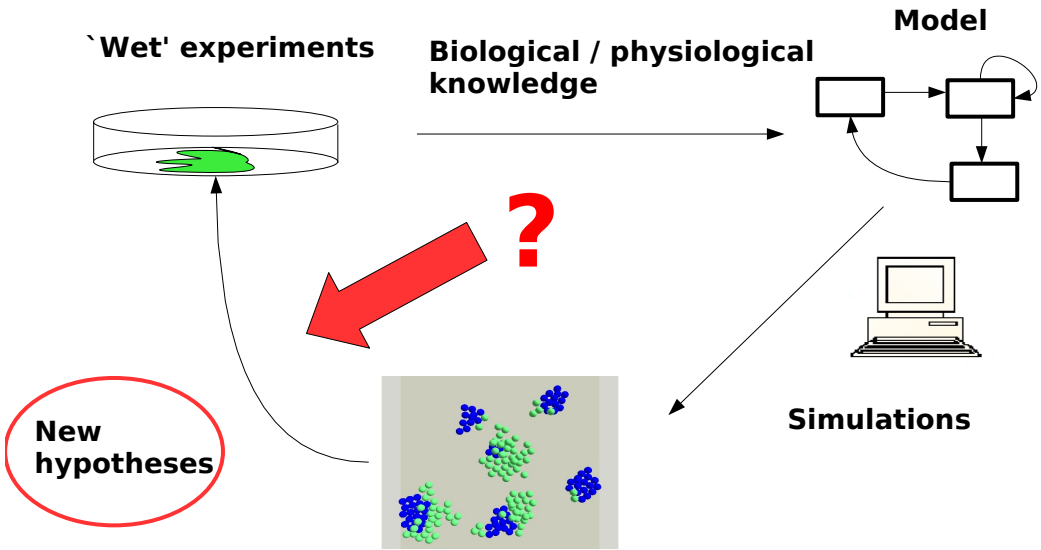


Analysing Computational Models



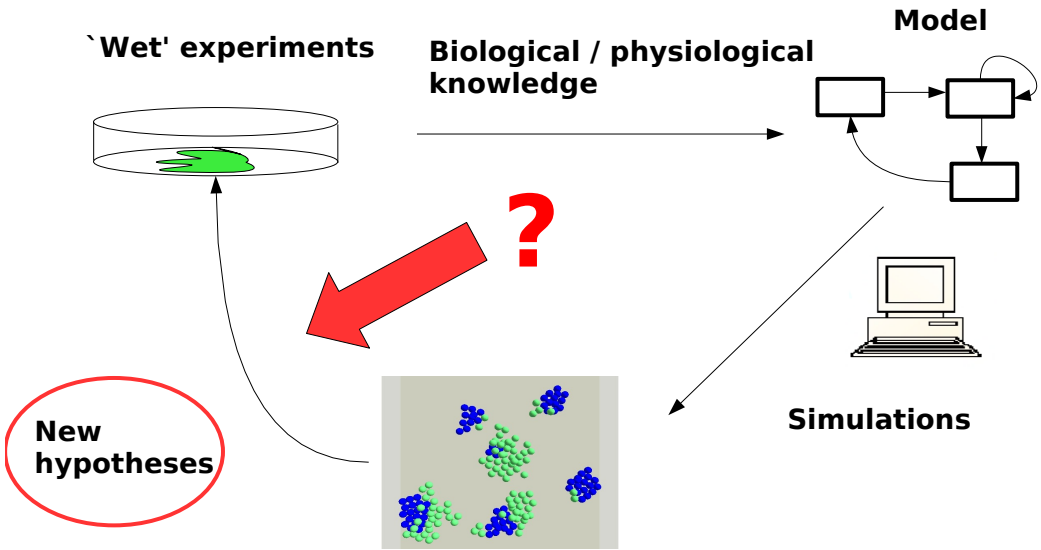
- Models are difficult to analyse
- Similar to the problem faced by software engineers when analysing software systems
- Use existing software analysis techniques to discover behaviour rules by analysing model simulations: **PIPE**, **Daikon** and **PRISM**.

Analysing Computational Models



- Models are difficult to analyse
- Similar to the problem faced by software engineers when analysing software systems
- Use existing software analysis techniques to discover behaviour rules by analysing model simulations: **PIPE**, **Daikon** and **PRISM**.

Analysing Computational Models



- Models are difficult to analyse
- Similar to the problem faced by software engineers when analysing software systems
- Use existing software analysis techniques to discover behaviour rules by analysing model simulations: **PIPE**, **Daikon** and **PRISM**.

Platform Independent Preti net Editor (PIPE)

- Platform independent tool for creating and analysing Petri nets including Generalised Stochastic Petri nets
- PIPE allows to check properties of boundedness and liveness (i.e, absence of deadlock)

T-invariants and P-invariants computed by PIPE for some initial marking contains one token in *gene n* in *act* and *m* in *rep*, with $n, m \geq 0$:

$r1$	1	0	0	0	0
$r2$	0	1	0	0	0
$r4$	0	1	0	0	0
$r3$	1	0	0	1	0
$r5$	0	0	1	0	0
$r7$	0	0	0	1	0
$r6$	0	0	1	0	0
$r8$	0	0	0	0	1
$r9$	0	0	0	0	1

gene	1	0	0	
rna	0	0	0	$M(gene) + M(act.gene) + M(rep.gene) = 1$
protein	0	0	0	$M(act) + M(act - gene) = n$
act	0	1	0	$M(rep) + M(rep - gene) = m$
$act - gene$	1	1	0	
rep	0	0	1	
$rep - gene$	1	0	1	

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Daikon

- Reverse engineers software specifications
- Analyses traces of software behaviour
- Uses machine learning techniques to extract rules relating program variables
- Can be used as basis for **interpreting output from biological models**
  - Prevent modeller from erroneously altering model behaviour
  - Precisely characterise specific model properties
  - Documenting or discovering novel aspects of model behaviour

Using Daikon with Biological Models

- Model aims to investigate effect of repressors and activators on expression activity
- Executed model 30 times
  - **10 times** with activators and no repressors (positve)
  - **10 times** with both repressors and activators (negative)
  - **10 times** with no repressors and no activators (constitutive)

Using Daikon with Biological Models

- Model aims to investigate effect of repressors and activators on expression activity
- Executed model 30 times
  - **10 times** with activators and no repressors (positve)
  - **10 times** with both repressors and activators (negative)
  - **10 times** with no repressors and no activators (constitutive)

Using Daikon with Biological Models

- Model aims to investigate effect of repressors and activators on expression activity
- Executed model 30 times
  - **10 times** with activators and no repressors (positve)
  - **10 times** with both repressors and activators (negative)
  - **10 times** with no repressors and no activators (constitutive)

Invariants Discovered by Daikon

Positive	Negative	Constitutive
$gene = one\ of\ \{0, 1\}$ $rep = rep\text{-}gene = 0$ $0 \leq rna \leq 24$ $0 \leq prot \leq 205$ $act = one\ of\ \{9, 10\}$ $act\text{-}gene = one\ of\ \{0, 1\}$ $(gene \wedge act\text{-}gene) = 0$ $(rna = 0) \rightarrow (prot = 0)$	$gene = one\ of\ \{0, 1\}$ $act = act\text{-}gene = 0$ $rna = one\ of\ \{0, 1\}$ $rep = one\ of\ \{9, 10\}$ $rep\text{-}gene = one\ of\ \{0, 1\}$ $prot = one\ of\ \{0, 1, 2, 3\}$ $(gene \wedge rep\text{-}gene) = 0$ $rna < rep$ $rep > prot$	$rep = rep\text{-}gene$ $= act$ $= act\text{-}gene$ $= 0$ $gene = 1$ $0 \leq rna \leq 7$ $0 \leq prot \leq 32$ $rna \geq rep$

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

Probabilistic and Symbolic Model Checking (PRISM)

- Probabilistic model checking is a formal verification technique.
- Construction of a precise mathematical model of a system which is to be analysed.
- Properties are expressed formally using temporal logic and analysed.
- The fundamental components of the **PRISM language** are modules, variables and commands.
  - A model is composed of a number of modules which can interact with each other.
  - A module contains a number of local variables and commands.

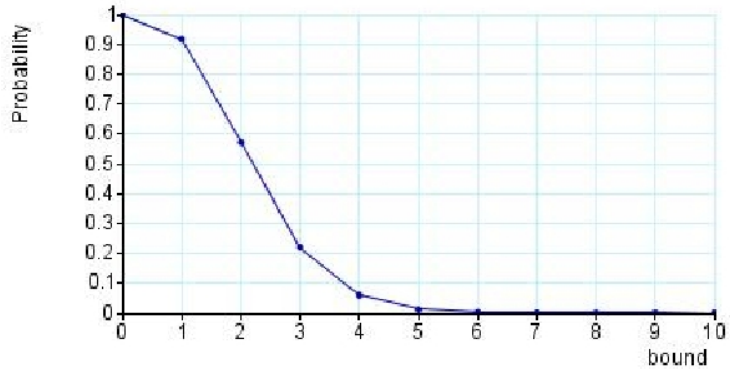
PRISM Model

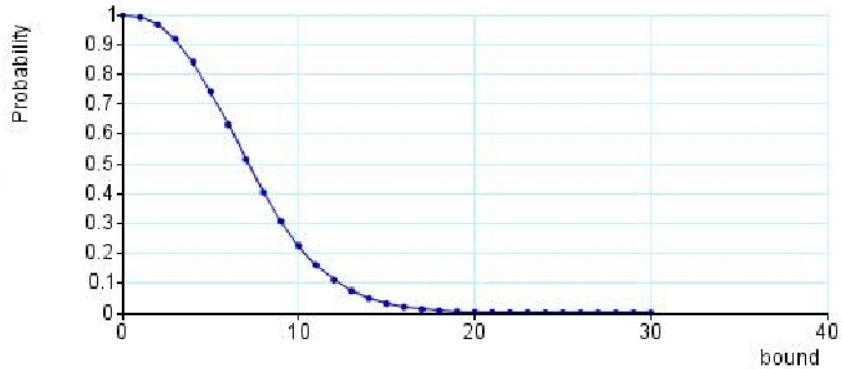
```
module bacterium
    gene : [ 0 .. 1 ] init 1;
    rna : [ 0 .. rnabound ] init 0;
    protein : [ 0 .. proteinbound ] init 0;
    // [ gene ]b -c1-> [ gene + rna ]b
    [ ] gene = 1 & rna < rnabound -> c1 : (rna' = rna + 1);
    // [ rna ]b -c2-> [ rna + protein ]b
    [ ] rna > 0 & protein < proteinbound -> c2*rna :(protein' = protein + 1);
    // [ rna ]b -c3-> [ ]b
    [ ] rna > 0 -> c3*rna : (rna' = rna - 1);
    // [ protein ]b -c4-> [ ]b
    [ ] protein > 0 -> c4*protein : (protein' = protein - 1);
endmodule
```

PRISM Results: Ranges of molecules in Constitutive Expression

$0 \leq rna \leq 7$ $0 \leq prot \leq 32$

P = ? [true U <= T rna > bound]
P = ? [true U <= T protein > bound]

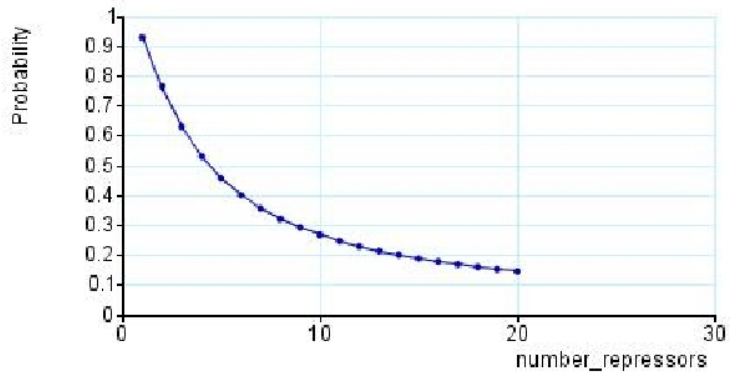


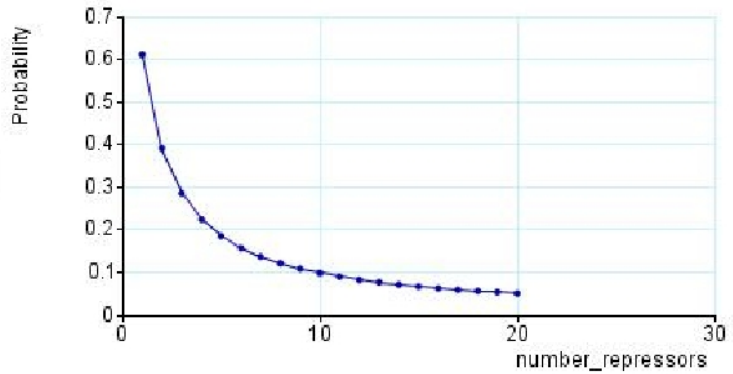


PRISM Results: Relationship between rna/protein and repressors

rna < *rep*	*rep* > *prot*

P = ? [true U<=T rna > rep]
P = ? [true U<=T protein > rep]





Take Home Message

- Modelling is intrinsic to the scientific activity.
- Limitations of the Conventional Macroscopic and deterministic approach (ODE).
- Computational Modelling: Not only custom programs but formal frameworks (Petri Nets, Process Algebra, Membrane Computing).
- Simulation is not enough. Analysis on computational models is necessary.
- PIPE, Daikon and PRISM are good candidates to solve this problem.

Take Home Message

- Modelling is intrinsic to the scientific activity.
- Limitations of the Conventional Macroscopic and deterministic approach (ODE).
- Computational Modelling: Not only custom programs but formal frameworks (Petri Nets, Process Algebra, Membrane Computing).
- Simulation is not enough. Analysis on computational models is necessary.
- PIPE, Daikon and PRISM are good candidates to solve this problem.

Take Home Message

- Modelling is intrinsic to the scientific activity.
- Limitations of the Conventional Macroscopic and deterministic approach (ODE).
- Computational Modelling: Not only custom programs but formal frameworks (Petri Nets, Process Algebra, Membrane Computing).
- Simulation is not enough. Analysis on computational models is necessary.
- PIPE, Daikon and PRISM are good candidates to solve this problem.

Take Home Message

- Modelling is intrinsic to the scientific activity.
- Limitations of the Conventional Macroscopic and deterministic approach (ODE).
- Computational Modelling: Not only custom programs but formal frameworks (Petri Nets, Process Algebra, Membrane Computing).
- Simulation is not enough. Analysis on computational models is necessary.
- PIPE, Daikon and PRISM are good candidates to solve this problem.

Take Home Message

- Modelling is intrinsic to the scientific activity.
- Limitations of the Conventional Macroscopic and deterministic approach (ODE).
- Computational Modelling: Not only custom programs but formal frameworks (Petri Nets, Process Algebra, Membrane Computing).
- Simulation is not enough. Analysis on computational models is necessary.
- PIPE, Daikon and PRISM are good candidates to solve this problem.

$Ενχαριστω \quad πολν$

Thank You