

# Reverse Engineering by Artificial Network Evolution

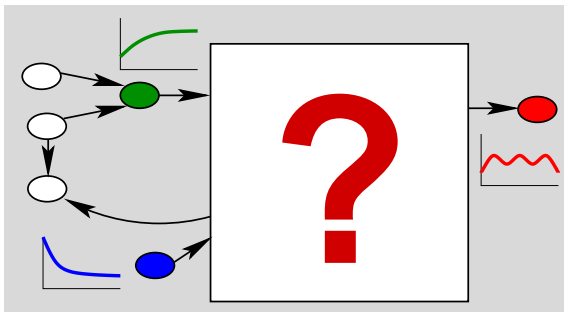
Thomas Hinze   Ines Heiland   Benedict Schau  
Thorsten Lenser   Stefan Schuster

Friedrich-Schiller University Jena  
Department of Bioinformatics at School of Biology and Pharmacy

`thomas.hinze@uni-jena.de`

October 09, 2009

# Reaction Network Reconstruction from Scratch



- Partially unknown topology
- Some behavioural data available
- Reconstruction of appropriate reaction network candidates
- Capturing ideas and inspirations for network topologies and parameterisation suitable for specific task

# Reaction Network Reconstruction: A Challenging Task

## Exhaustive candidate enumeration

- Exponential growth of search space:  
 $n$  species  $\rightarrow 2^{2n}$  possible first-order reactions

## Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

## Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

## Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

## Artificial network evolution

- Universal heuristics adopted from biological evolution

# Reaction Network Reconstruction: A Challenging Task

## Exhaustive candidate enumeration

- Exponential growth of search space:  
 $n$  species  $\rightarrow 2^{2n}$  possible first-order reactions

## Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

## Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

## Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

## Artificial network evolution

- Universal heuristics adopted from biological evolution

# Reaction Network Reconstruction: A Challenging Task

## Exhaustive candidate enumeration

- Exponential growth of search space:  
 $n$  species  $\rightarrow 2^{2n}$  possible first-order reactions

## Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

## Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

## Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

## Artificial network evolution

- Universal heuristics adopted from biological evolution

# Reaction Network Reconstruction: A Challenging Task

## Exhaustive candidate enumeration

- Exponential growth of search space:  
 $n$  species  $\rightarrow 2^{2n}$  possible first-order reactions

## Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

## Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

## Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

## Artificial network evolution

- Universal heuristics adopted from biological evolution

# Reaction Network Reconstruction: A Challenging Task

## Exhaustive candidate enumeration

- Exponential growth of search space:  
 $n$  species  $\rightarrow 2^{2n}$  possible first-order reactions

## Finding homologies

- Employ synergetic effects: known networks with similar functionality could be adapted

## Bottom-up engineering

- Provide small functional units and combine them towards entire network (constructive approach)

## Learning strategies

- Reduce a huge full network by successive weighting of reactions along with available behavioural data

## Artificial network evolution

- Universal heuristics adopted from biological evolution

## Why Artificial Evolution for Network Reverse Engineering

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.  
⇒ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.  
⇒ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.  
⇒ Artificial evolution can find unexpected, unconventional solutions.



[www.wordpress.com](http://www.wordpress.com)



## Why Artificial Evolution for Network Reverse Engineering

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.

⇒ Artificial evolution can explore network structure.

- Help in understanding emergence of biological complexity.

⇒ Evolution becomes observable.

- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.

⇒ Artificial evolution can find unexpected, unconventional solutions.



[www.wordpress.com](http://www.wordpress.com)

## Why Artificial Evolution for Network Reverse Engineering

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.  
⇒ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.  
⇒ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.  
⇒ Artificial evolution can find unexpected, unconventional solutions.



[www.wordpress.com](http://www.wordpress.com)

## Why Artificial Evolution for Network Reverse Engineering

- Systems Biology deals with interplay of biological components rather than components themselves.
- Accumulation of small modifications in component's interplay can result in a new quality of the entire network.  
⇒ Artificial evolution can explore network structure.
- Help in understanding emergence of biological complexity.  
⇒ Evolution becomes observable.
- Furthermore, bio-inspired approaches provide a flexible, fault-tolerant, reliable paradigm.  
⇒ Artificial evolution can find unexpected, unconventional solutions.

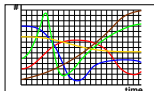
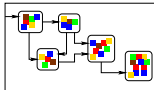
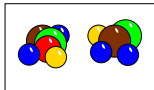
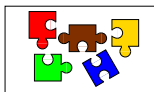


[www.wordpress.com](http://www.wordpress.com)

# Outline

## Reverse Engineering by Artificial Network Evolution

1. Artificial evolution at a glance
2. The SBMLevolver:  
a two-level evolutionary algorithm
3. Evolved networks: a selection
4. Ongoing study: control system-based  
specification of circadian oscillators
5. Perspective: hierarchical evolution



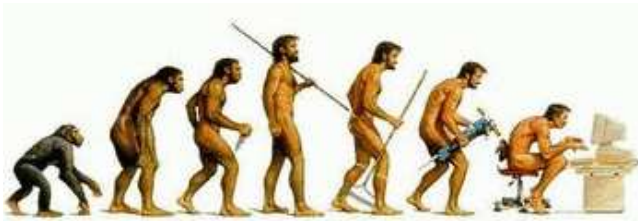
1	0	0	1	1	0	0
0	1	1	0	0	1	0
1	0	1	1	0	1	0
1	0	0	1	0	0	1

# Evolutionary Computing

- Abstraction and formalisation of evolutionary processes
- Individuals (genotype, phenotype) and population
- Evolutionary operators along with fitness evaluation
- Heuristical optimisation technique, experimentally driven

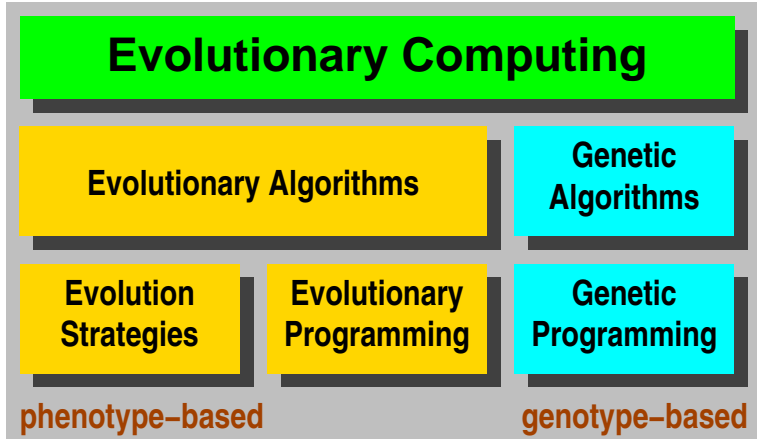
## Artificial evolution

- Initiated by Friedmann 1956
- Pioneers: Rechenberg, Schwefel, Fogel, Holland, Banzhaf, Koza, Sauro, . . .

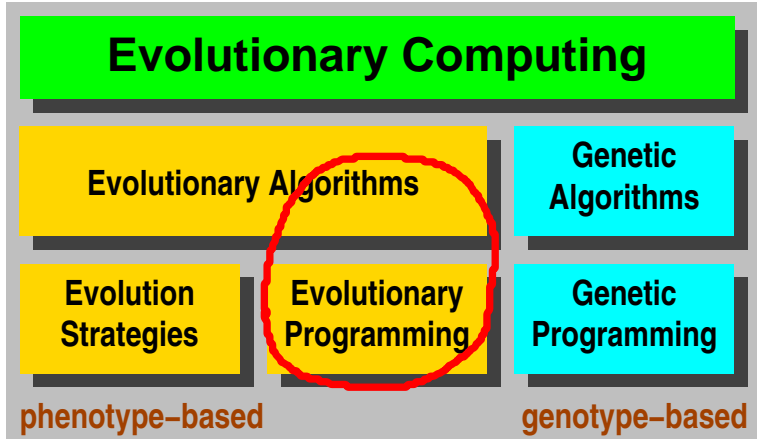


pics.goingon.com

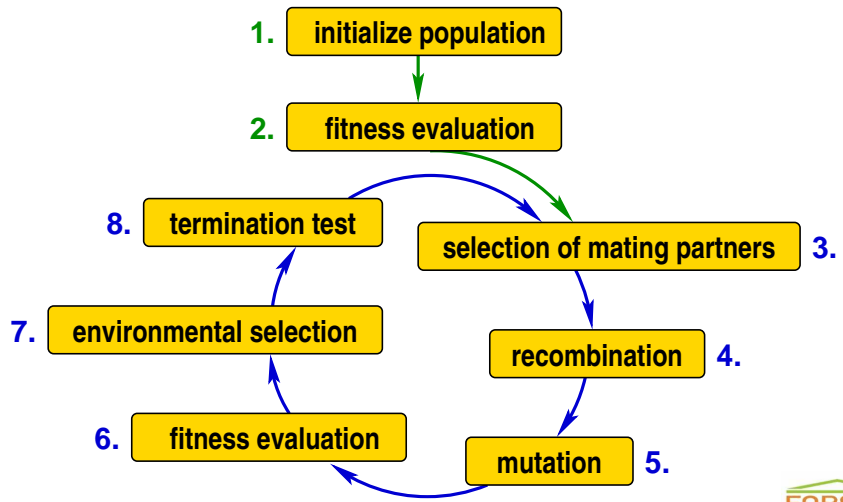
# Facets and Specialties



# Facets and Specialties



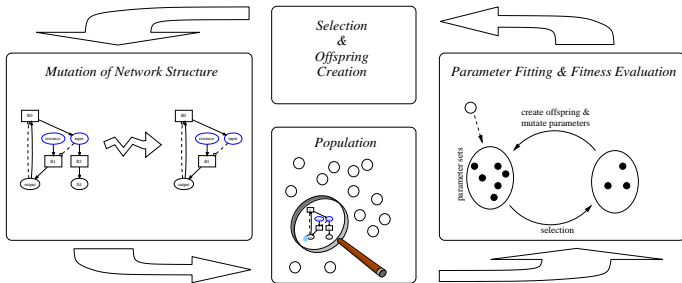
# Central Loop in Evolutionary Algorithms





# SBMLevolver: Two-Level Evolutionary Algorithm

- Separation of structural evolution from parameter fitting
- Idea: parameters can adapt to mutated network structure

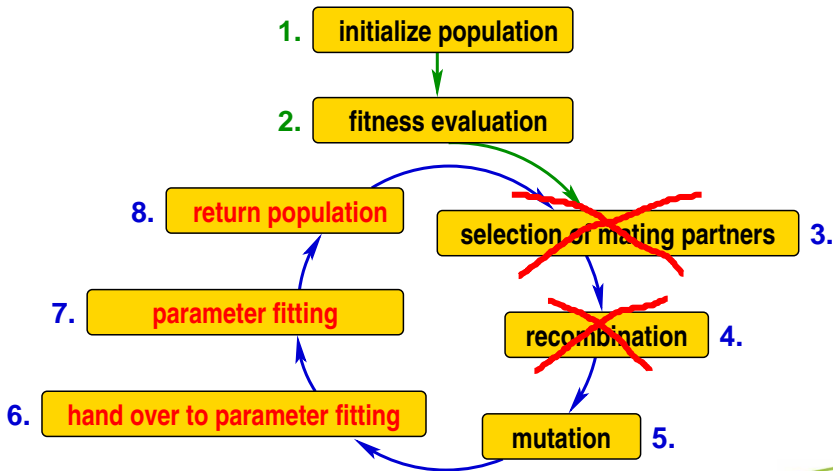


- Upper level: network structure
- Lower level: kinetic parameter fitting

⇒ open-source freeware:

<http://users.minet.uni-jena.de/~biosys/esignet>

# Structural Evolution



# Initialization of Network Population

Initial population configurable,  
typically 50 . . . 100 network individuals as SBML files

## Empty

- Network reconstruction from scratch

## Randomly chosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

## Taken from imported SBML file

- Generate a number of file copies
- Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution

# Initialization of Network Population

Initial population configurable,  
typically 50 . . . 100 network individuals as SBML files

## Empty

- Network reconstruction from scratch

## Randomly chosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

## Taken from imported SBML file

- Generate a number of file copies
- Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution

# Initialization of Network Population

Initial population configurable,  
typically 50 . . . 100 network individuals as SBML files

## Empty

- Network reconstruction from scratch

## Randomly chosen

- Individual networks randomly chosen, upper/lower limits for numbers of species, reactions, and kinetic parameter values

## Taken from imported SBML file

- Generate a number of file copies
- **Dedicated species, reactions, and kinetic parameters can be marked as fixed during evolution**

# Fitness Evaluation

## Specification of dynamical behaviour

- Input/output table: desired course of input and output species at discrete points in time
- Distinction between finite number of cases (runs) in input/output table
- Penalties can be set

## Fitness evaluation

- Numerical integration of reaction network using ODE solver (SOSlib)
- Currently, mass-action kinetics
- Fitness measure given by weighted squared distance to target time course (output species)
- Minimisation of fitness value (!)

```
# Initial input concentrations
# Starting with * sets the concentr
# Only one number means the concent
# Case 0
* 0
* 0
# Case 1
* 0
* 10
# Case 2
* 10
* 0
# Case 3
* 10
* 10

# Now the output data comes
# Case 0
0
# Case 1
10
# Case 2
```

# Fitness Evaluation

## Specification of dynamical behaviour

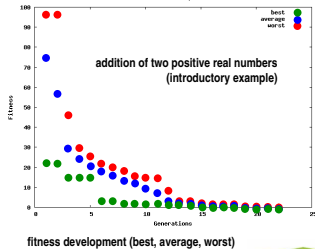
- Input/output table: desired course of input and output species at discrete points in time
- Distinction between finite number of cases (runs) in input/output table
- Penalties can be set

## Fitness evaluation

- Numerical integration of reaction network using ODE solver (SOSlib)
- Currently, mass-action kinetics
- Fitness measure given by weighted squared distance to target time course (output species)
- Minimisation of fitness value (!)

```
# Initial input concentrations
# Starting with * sets the concentr
# Only one number means the concent
# Case 0
* 0
* 0
# Case 1
* 0
* 10
# Case 2
* 10
* 0
# Case 3
* 10
* 10

# Now the output data comes
# Case 0
0
# Case 1
10
# Case 2
```



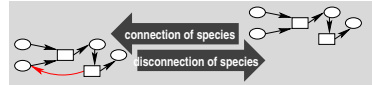
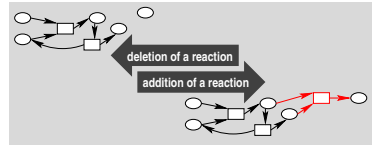
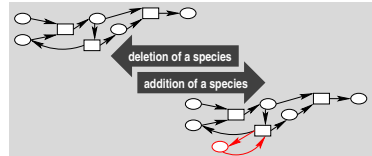
# Mutation Operators

Seven mutations available,  
randomly selected

- Addition/deletion of a species
- Addition/deletion of a reaction
- Connection/removal of existing species to/from a reaction
- Duplication of a species with all its reactions

Network size can be limited.

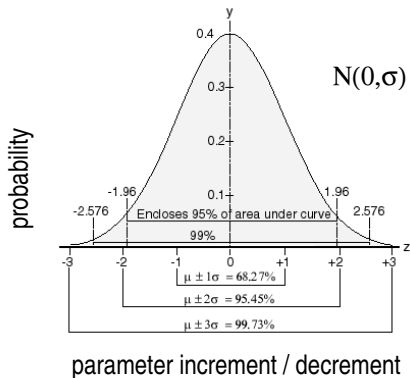
⇒ **One or several mutations per turn**





# Parameter Fitting

- Adaptation of networks after structural mutation(s)
- Separate evolutionary algorithm
- Generate copies of networks resulted from structural mutation(s)
- Random selection of one or several kinetic parameters
- Mutation: addition of **Gauss variable**
- Plausibility check
- No recombination
- Environmental selection



# Environmental Selection

## Small population size

- Due to high computational costs of fitness evaluation

## Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

## Non-overlapping generations

- Comma-selection supports self adaptation

## Parameter settings copied from parent to offspring

- Incremental parameter fitting

## Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

# Environmental Selection

## Small population size

- Due to high computational costs of fitness evaluation

## Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

## Non-overlapping generations

- Comma-selection supports self adaptation

## Parameter settings copied from parent to offspring

- Incremental parameter fitting

## Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

# Environmental Selection

## Small population size

- Due to high computational costs of fitness evaluation

## Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

## Non-overlapping generations

- Comma-selection supports self adaptation

## Parameter settings copied from parent to offspring

- Incremental parameter fitting

## Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

# Environmental Selection

## Small population size

- Due to high computational costs of fitness evaluation

## Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

## Non-overlapping generations

- Comma-selection supports self adaptation

## Parameter settings copied from parent to offspring

- Incremental parameter fitting

## Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

# Environmental Selection

## Small population size

- Due to high computational costs of fitness evaluation

## Self-adaptation of strategy parameters

(Gaussian distribution)

- Balancing between exploration of search space and fine-tuning

## Non-overlapping generations

- Comma-selection supports self adaptation

## Parameter settings copied from parent to offspring

- Incremental parameter fitting

## Fitness proportional selection

- Combines survival of the fittest with ability to leave local optima and keeps diversity of population

# Termination and Final Network Simplification

## Termination

- Best fitness below configurable threshold (ideally = 0)
- After configurable number of generations
- After configurable number of fitness evaluations

## Final network simplification

- Optional, only deletion of species keeping minimal fitness

## Challenges and insufficiencies

- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)

# Termination and Final Network Simplification

## Termination

- Best fitness below configurable threshold (ideally = 0)
- After configurable number of generations
- After configurable number of fitness evaluations

## Final network simplification

- Optional, only deletion of species keeping minimal fitness

## Challenges and insufficiencies

- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)



# Termination and Final Network Simplification

## Termination

- Best fitness below configurable threshold (ideally = 0)
- After configurable number of generations
- After configurable number of fitness evaluations

## Final network simplification

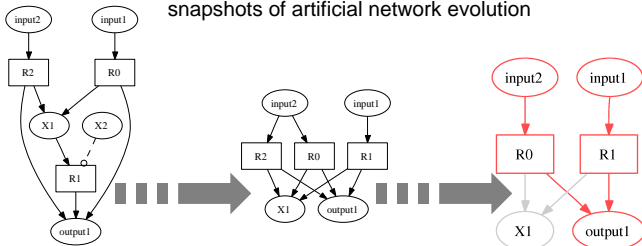
- Optional, only deletion of species keeping minimal fitness

## Challenges and insufficiencies

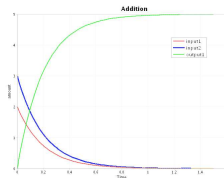
- Premature convergence along with low diversity of population
- Overfitting (perfect replication of test cases but no further functionality of network)

# Introductory Example: Arithmetic Addition

Task: addition of two positive real numbers  
snapshots of artificial network evolution

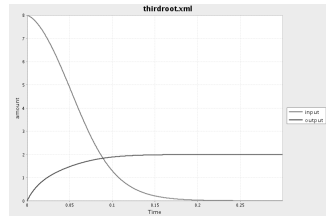
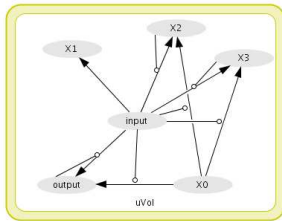
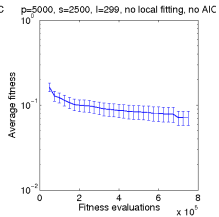
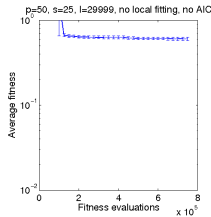
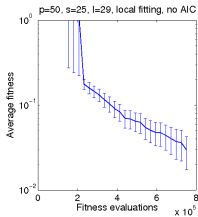


- **R0, R1, R2** identify reactions
- **input1, input2, output1**: dedicated species
- **X1, X2**: auxiliary species
- Stepwise modification of network structure and kinetic parameters



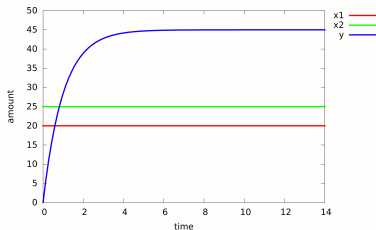
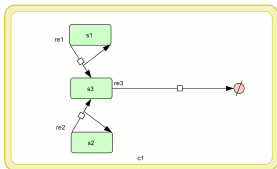
# Third Root Network

initial conc. of input species  $\mapsto$  steady state conc. of output species



T. Lenser, T. Hinze, B. Ibrahim, P. Dittrich. Towards Evolutionary Network Reconstruction Tools for Systems Biology. In E. Marchiori, J.H. Moore, J.C. Rajapakse (Eds.), Proceedings Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer LNCS 4447:132-142, 2007

# Addition



$$\frac{dx_1}{dt} = 0 \quad \frac{dx_2}{dt} = 0 \quad \frac{dy}{dt} = k_1 x_1 + k_2 x_2 - k_3 y$$

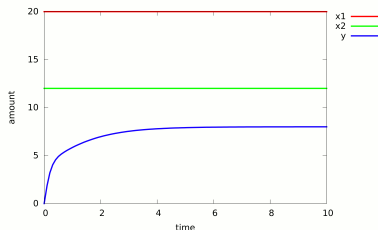
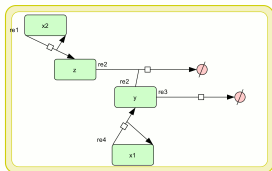
Let  $k_1 = k_2 = k_3 > 0$ .

Steady state:

$$y = \lim_{t \rightarrow \infty} (1 - e^{-k_1 t}) \cdot (x_1 + x_2) = x_1 + x_2$$

B. Schau, T. Hinze, T. Lenser, I. Heiland, S. Schuster. Control System-Based Reverse Engineering of Circadian Oscillators. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, P. Stadler (Eds.), Proceedings German Conference on Bioinformatics (GCB2009), p. 126-127, Martin-Luther University Halle-Wittenberg, 2009

# Non-Negative Subtraction



$$\frac{dx_1}{dt} = 0$$

$$\frac{dx_2}{dt} = 0$$

$$\frac{dy}{dt} = -k_2 y z - k_1 y + k_1 x_1$$

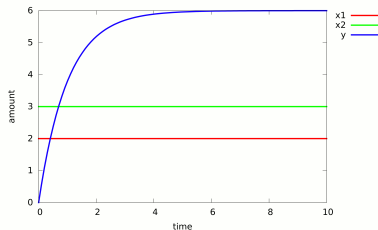
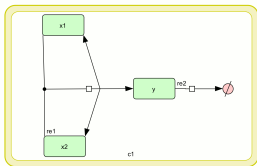
$$\frac{dz}{dt} = k_1 x_2 - k_2 y z$$

Let  $k_1 > 0$  and  $k_2 > 0$ .

Steady state:

$$y = \begin{cases} x_1 - x_2 & \text{iff } x_1 > x_2 \\ 0 & \text{otherwise} \end{cases}$$

# Multiplication



$$\frac{dx_1}{dt} = 0 \quad \frac{dx_2}{dt} = 0 \quad \frac{dy}{dt} = k_1 x_1 x_2 - k_2 y$$

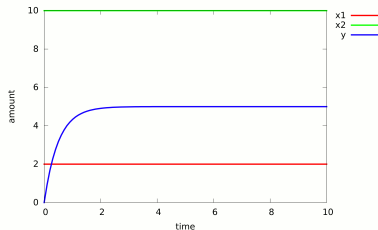
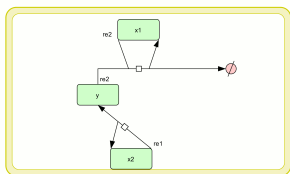
Let  $k_1 = k_2 > 0$ .

Steady state:

$$y = \lim_{t \rightarrow \infty} (1 - e^{-k_1 t}) \cdot x_1 \cdot x_2 = x_1 \cdot x_2$$

B. Schau, T. Hinze, T. Lenser, I. Heiland, S. Schuster. Control System-Based Reverse Engineering of Circadian Oscillators. In I. Grosse, S. Neumann, S. Posch, F. Schreiber, P. Stadler (Eds.), Proceedings German Conference on Bioinformatics (GCB2009), p. 126-127, Martin-Luther University Halle-Wittenberg, 2009

# Division



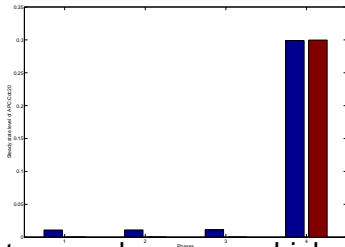
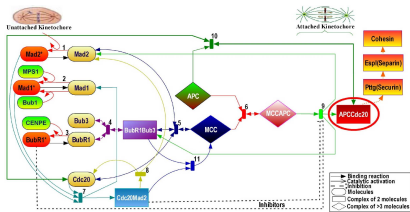
$$\frac{dx_1}{dt} = 0 \quad \frac{dx_2}{dt} = 0 \quad \frac{dy}{dt} = k_2 x_2 - k_1 x_1 y$$

Let  $k_1 = k_2 > 0$ . Steady state:

$$y = \begin{cases} \lim_{t \rightarrow \infty} (1 - e^{-k_1 t}) \cdot \frac{x_2}{x_1} & \text{iff } x_1 > 0 \\ \lim_{t \rightarrow \infty} \int k_2 x_2 dt & \text{otherwise} \end{cases}$$

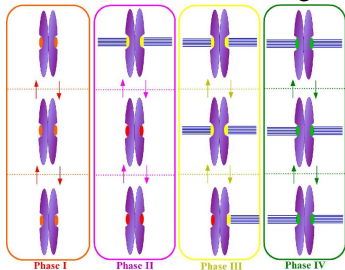
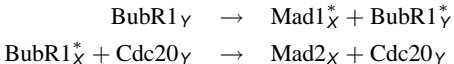
$$= \begin{cases} \frac{x_2}{x_1} & \text{iff } x_1 > 0 \\ \rightarrow \infty & \text{iff } x_1 = 0 \text{ and } x_2 > 0 \\ 0 & \text{iff } x_1 = 0 \text{ and } x_2 = 0 \end{cases}$$

# Case Study: Human Spindle Assembly Checkpoint



- 17 species, 11 reactions
- Compartments represent chromosomes X, Y
- Structural evolution adds two (unrealistic) reactions:

target: ————— low ————— high —————



T. Lenser, T. Hinze, B. Ibrahim, P. Dittrich. Springer LNCS 4447:132-142, 2007



# Circadian Systems

## Characteristics

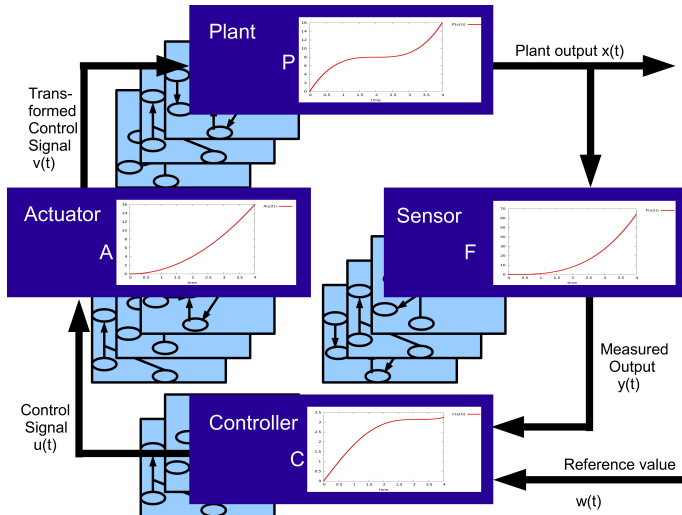
- Self-sustained biochemical oscillators
- Period of approx. 24 hours persisting under constant environmental conditions (e.g. constant darkness)
- Temperature compensation within physiological range
- Capability of entrainment by external stimuli (e.g. light/dark or temperature cycles)
- Reaction system with at least one feedback loop

## High scientific impact because . . .

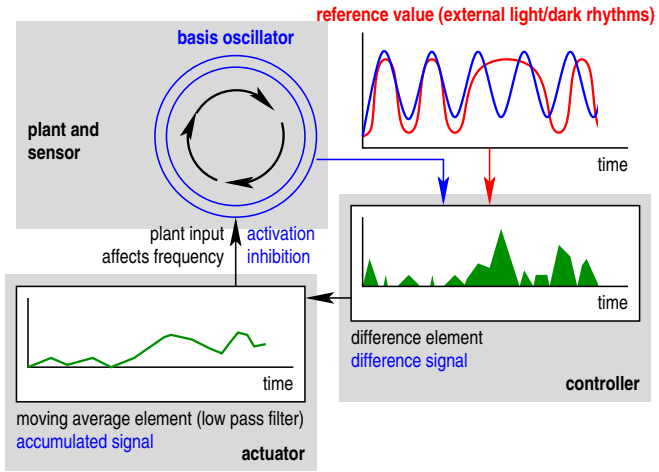
- Circadian clock as a potential universal property of life
- Self-sustainability and high precision of bio-oscillators
- Chronobiological control systems for manifold processes
- Several independent evolutionary origins assumed

# Representation as Control System

Separation of the system into smaller functional components

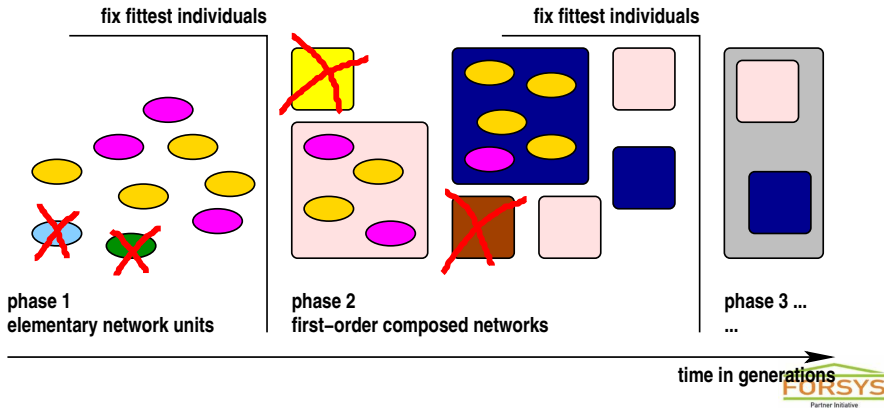


# Circadian Entrainment as Phase Locking Loop



# Perspective: Hierarchical Evolution

- Consecutive phases of evolution
- Fix fittest individuals after each phase and compose them
- Emerging complex systems within reduced search space



## Special Thanks go to . . .

### ... my coworkers

**Benedict Schau**

Department Bioinformatics, FSU Jena



**Ines Heiland**

Department Bioinformatics, FSU Jena



**Thorsten Lenser**

Bio Systems Analysis Group, FSU Jena



**Stefan Schuster**

Department Bioinformatics, FSU Jena



### ... the funding organization

German Federal Ministry of Education  
and Research, project 0315260A within  
Research Initiative in Systems Biology



Bundesministerium  
für Bildung  
und Forschung

### ... you for your attention. Questions?