Maintenance of Chronobiological Information by P System Mediated Assembly of Control Units for Oscillatory Waveforms and Frequency

Thomas Hinze[1,2] Benjamin Schell[2]

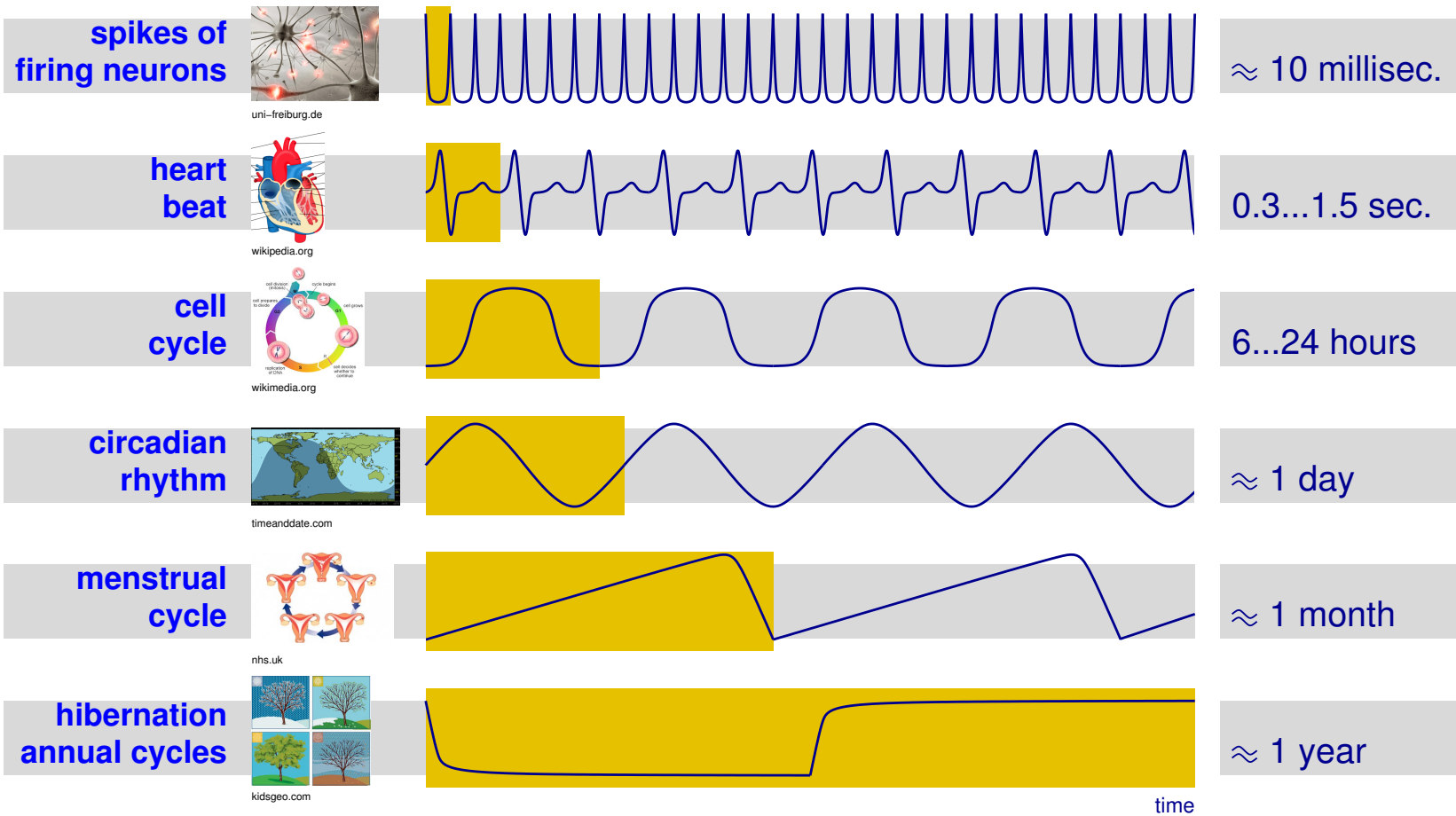
Mathias Schumann[2] Christian Bodenstein[2]

[1]Brandenburg University of Technology Cottbus
Institute of Computer Science and Information and Media Technology

[2]Friedrich Schiller University Jena
Department of Bioinformatics at School of Biology and Pharmacy

thomas.hinze@tu-cottbus.de

Oscillatory Signals in Biology: Range of Periodicities

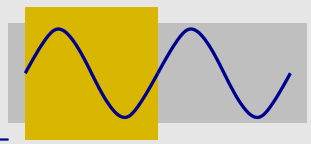


An individual organism comprises a variety of regulated oscillations.

Waveforms of Oscillatory Signals in Biology

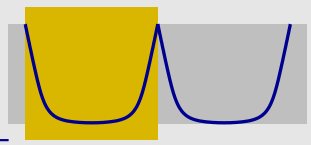
Waveform as additional information carrier beyond frequency

sinusoidal or almost sinusoidal



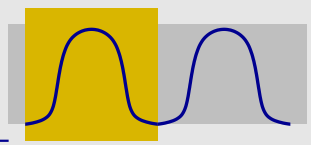
- gradual and smooth alteration
- commonly stable limit cycle
- quite robust against perturbations

spiking



- signal peak for short moment
- fast raise or fall of signal value easy to detect
- average signal level can be kept low

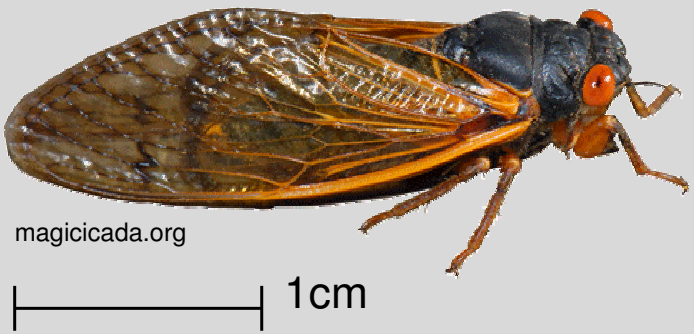
plated

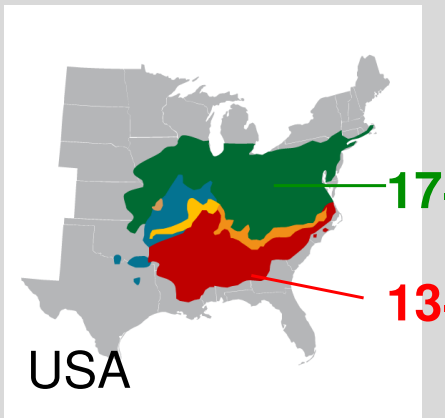


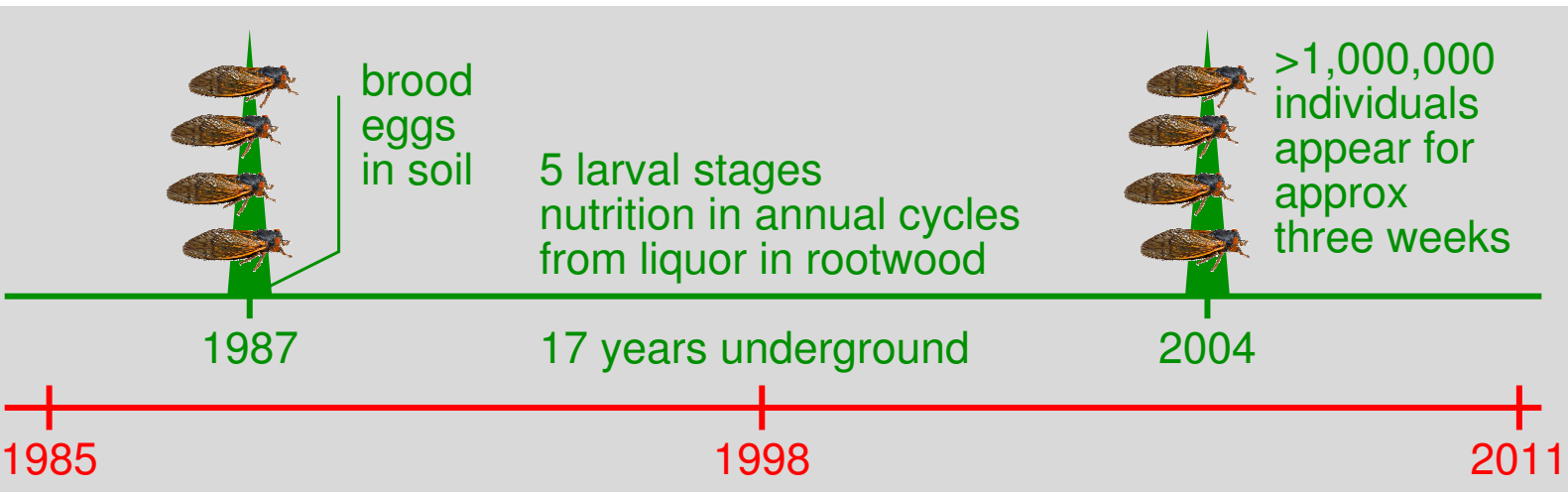
- more or less bistable oscillatory behaviour
- almost rectangular shape towards binarisation
- toggling with weightable balance btw. high/low

Corresponding oscillators for each waveform by small or medium-sized reaction networks

17-years and 13-years Periodical Cicadas with Synchronous Life Cycle







Inspiration and Motivating Questions

- No external stimulus with natural period of 17 or 13 years known up to now
- Molecular mechanism to precisely measure the passage of 17 (or 13) years?
- Need of a chemical frequency divider model, ideally configurable for distinct division ratios
- Low number of slight evolutionary changes sufficient to toggle the life cycle between a variety of years?

$\Longrightarrow$ Assembly of pre-defined chemical modules towards new or extended functionality

Inspiration and Motivating Questions

- No external stimulus with natural period of 17 or 13 years known up to now
- Molecular mechanism to precisely measure the passage of 17 (or 13) years?
- Need of a chemical frequency divider model, ideally configurable for distinct division ratios
- Low number of slight evolutionary changes sufficient to toggle the life cycle between a variety of years?

$\Longrightarrow$ Assembly of pre-defined chemical modules towards new or extended functionality

Inspiration and Motivating Questions

- No external stimulus with natural period of 17 or 13 years known up to now
- Molecular mechanism to precisely measure the passage of 17 (or 13) years?
- Need of a chemical frequency divider model, ideally configurable for distinct division ratios
- Low number of slight evolutionary changes sufficient to toggle the life cycle between a variety of years?

$\Longrightarrow$ Assembly of pre-defined chemical modules towards new or extended functionality

Inspiration and Motivating Questions

- No external stimulus with natural period of 17 or 13 years known up to now
- Molecular mechanism to precisely measure the passage of 17 (or 13) years?
- Need of a chemical frequency divider model, ideally configurable for distinct division ratios
- Low number of slight evolutionary changes sufficient to toggle the life cycle between a variety of years?

$\Longrightarrow$ Assembly of pre-defined chemical modules towards new or extended functionality

Inspiration and Motivating Questions

- No external stimulus with natural period of 17 or 13 years known up to now
- Molecular mechanism to precisely measure the passage of 17 (or 13) years?
- Need of a chemical frequency divider model, ideally configurable for distinct division ratios
- Low number of slight evolutionary changes sufficient to toggle the life cycle between a variety of years?

$\Longrightarrow$ Assembly of pre-defined chemical modules towards new or extended functionality

1. Biological Rhythms
2. **P Meta Framework**
   - Non-probabilistic P modules
   - Connectivity of P modules
   - Instructions for composition of P modules
3. Assembly of Frequency Dividers
4. Suprachiasmatic Nucleus

Non-probabilistic P Module

- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module

- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module

- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module

- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module

- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification
with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module

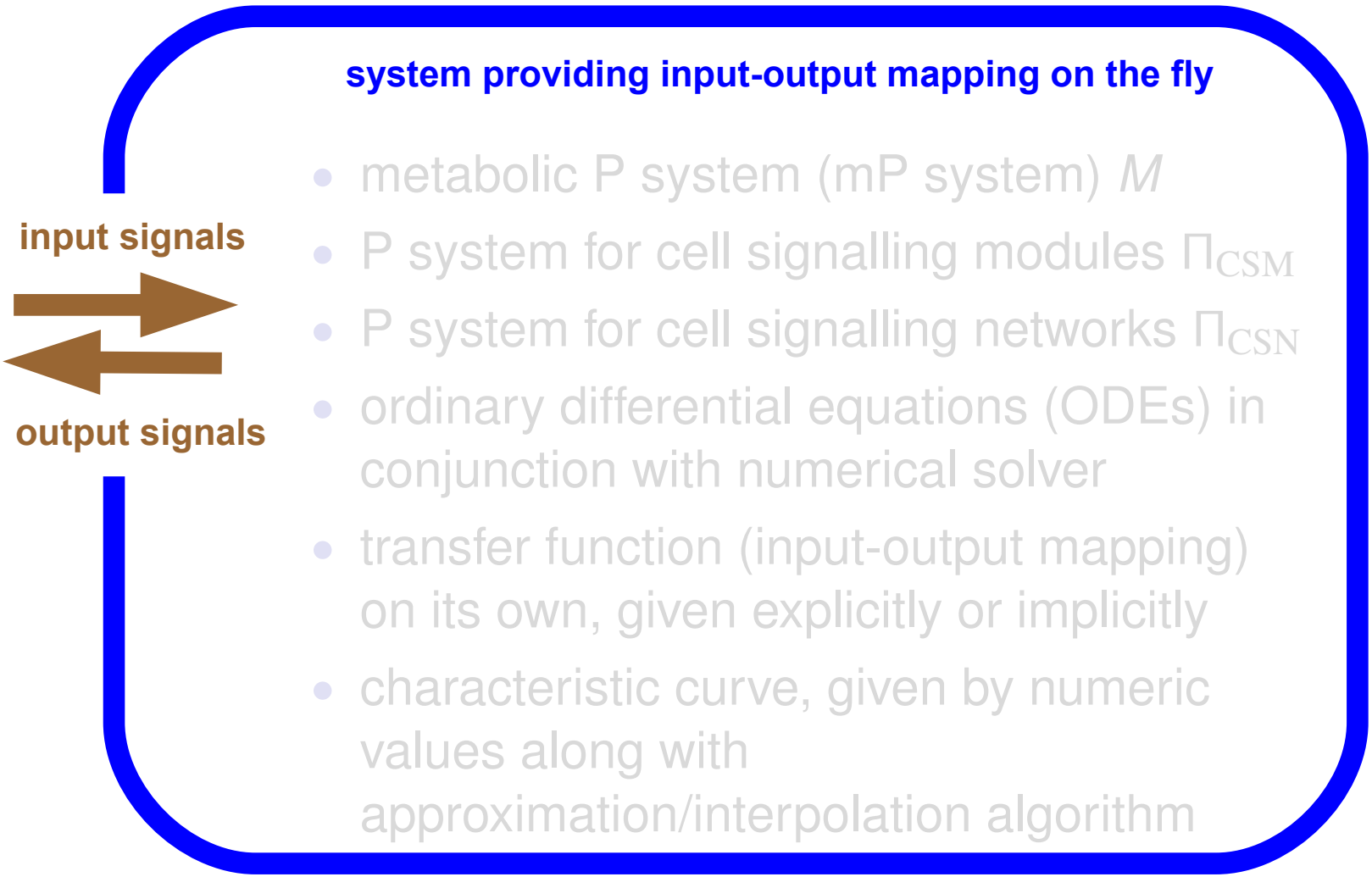
- is able to fulfill an elementary computational task on the fly
- building block of an analog computer or in a control loop
- represents a container encapsulating a formal description of its dynamical behaviour
- specifies the interface of a general real-valued system or its approximation
- aims to bridge building blocks in systems theory and membrane systems

More formally, a P module is a triple $(\downarrow, \uparrow, \square)$ where

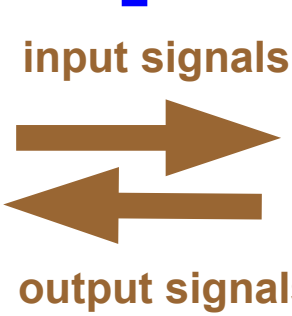
$\downarrow = (I_1, \ldots, I_i)$ ..... indicates a list of input signal identifiers
$\uparrow = (O_1, \ldots, O_o)$ ..... indicates a list of output signal identifiers
$\square$ ..... underlying system specification with or without inherent auxiliary signals

Each signal is a real-valued function over time.

Non-probabilistic P Module as a Processing Unit



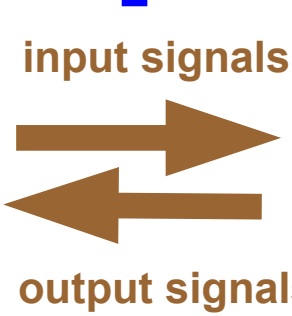
Non-probabilistic P Module as a Processing Unit



system providing input-output mapping on the fly

- metabolic P system (mP system) *M*
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

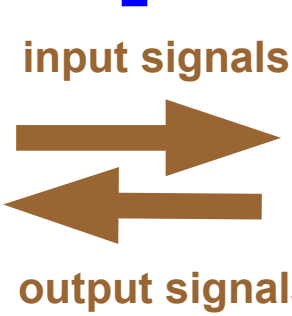
Non-probabilistic P Module as a Processing Unit



system providing input-output mapping on the fly

- metabolic P system (mP system) M
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

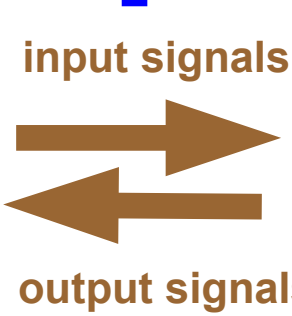
Non-probabilistic P Module as a Processing Unit



system providing input-output mapping on the fly

- metabolic P system (mP system) M
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

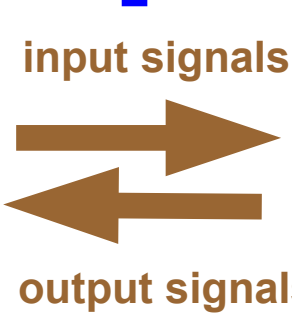
Non-probabilistic P Module as a Processing Unit



system providing input-output mapping on the fly

- metabolic P system (mP system) M
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

Non-probabilistic P Module as a Processing Unit

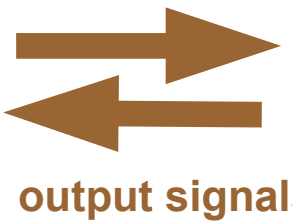


system providing input-output mapping on the fly

- metabolic P system (mP system) M
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

Non-probabilistic P Module as a Processing Unit

system providing input-output mapping on the fly



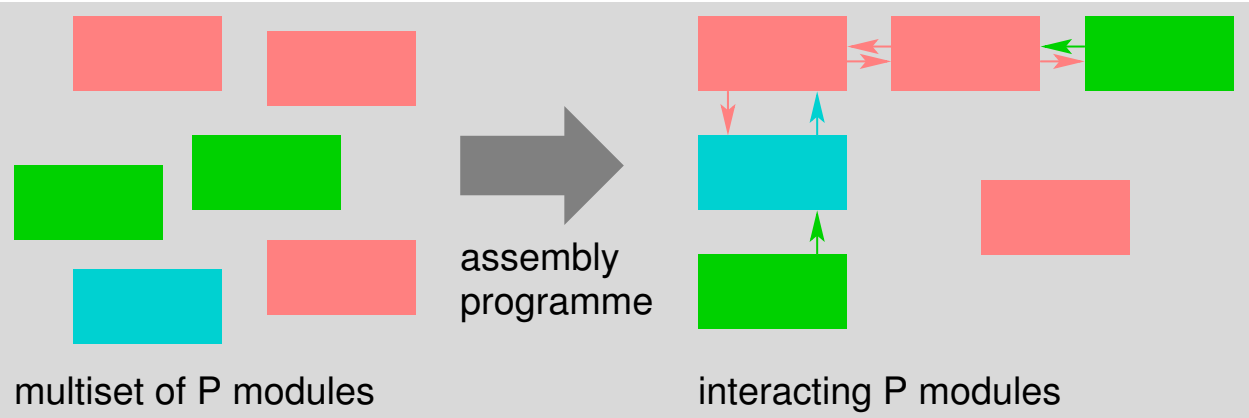
- metabolic P system (mP system) M
- P system for cell signalling modules Π_{CSM}
- P system for cell signalling networks Π_{CSN}
- ordinary differential equations (ODEs) in conjunction with numerical solver
- transfer function (input-output mapping) on its own, given explicitly or implicitly
- characteristic curve, given by numeric values along with approximation/interpolation algorithm

P Meta Framework: Connecting P Modules

Our P meta framework is a construct $\Pi_{\pi\uparrow\downarrow} = (M, P)$ where

M ..... denotes a finite multiset of non-probabilistic P modules

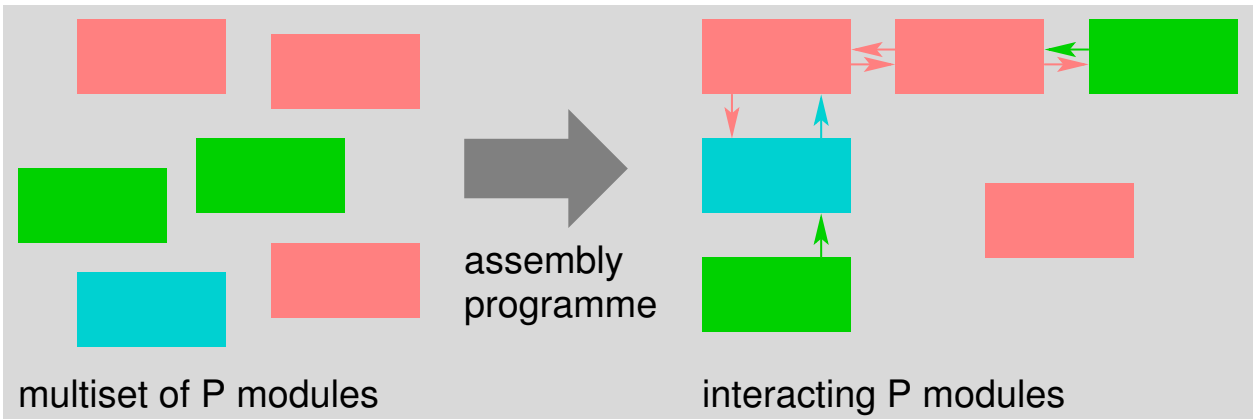
P ..... assembly programme by its connectivity instructions



P Meta Framework: Connecting P Modules

Our P meta framework is a construct $\Pi_{\pi\uparrow\downarrow} = (M, P)$ where

M .....denotes a finite multiset of non-probabilistic P modules
P .....assembly programme by its connectivity instructions



- $\mathrm{supp}(M)$ can be interpreted as the genetic potential of highly conserved reaction units
- Multiplicity of modules $\longrightarrow$ limitation of resources available for module composition

Connectivity Graph of a P Meta Framework

Each P Meta Framework $\Pi_{\pi\uparrow\downarrow} = (M, P)$ comes with a corresponding directed graph $G = (V, E)$, initially

$$\begin{aligned} V &:= \{m[i] \mid m \in \mathrm{supp}(M) \ \wedge \ i \in \{1, \ldots, M(m)\}\} \\ E &:= \emptyset \end{aligned}$$

- Indexing of all instances (copies) $m[i]$ from a P module m allows unique identification
- Let $a = (a_\downarrow, a_\uparrow, a_\square) \in \mathrm{supp}(M)$ and $b = (b_\downarrow, b_\uparrow, b_\square) \in \mathrm{supp}(M)$ be two module instances derived from M.
- An edge $(a, b, R_{a\to b}) \in E$ denotes a connection from a to b where dedicated output species of a act as input species of b.
- Each edge comes with a binary relation $R_{a\to b} \subseteq a_\uparrow \times b_\downarrow$ in which the mapping of a's output species onto b's input species is given.
- $R_{a\to b}$ handled in an injective manner since one output species is allowed to cover several downstream input species, but each input species must be supplied by at most one upstream output species.

Connectivity Graph of a P Meta Framework

Each P Meta Framework $\Pi_{\pi\uparrow\downarrow} = (M, P)$ comes with a corresponding directed graph $G = (V, E)$, initially

$$\begin{aligned} V &:= \{m[i] \mid m \in \mathrm{supp}(M) \ \wedge \ i \in \{1, \ldots, M(m)\}\} \\ E &:= \emptyset \end{aligned}$$

- Indexing of all instances (copies) $m[i]$ from a P module m allows unique identification
- Let $a = (a_\downarrow, a_\uparrow, a_\square) \in \mathrm{supp}(M)$ and $b = (b_\downarrow, b_\uparrow, b_\square) \in \mathrm{supp}(M)$ be two module instances derived from M.
- An edge $(a, b, R_{a\to b}) \in E$ denotes a connection from a to b where dedicated output species of a act as input species of b.
- Each edge comes with a binary relation $R_{a\to b} \subseteq a_\uparrow \times b_\downarrow$ in which the mapping of a's output species onto b's input species is given.
- $R_{a\to b}$ handled in an injective manner since one output species is allowed to cover several downstream input species, but each input species must be supplied by at most one upstream output species.

Connectivity Graph of a P Meta Framework

Each P Meta Framework $\Pi_{\pi\uparrow\downarrow} = (M, P)$ comes with a corresponding directed graph $G = (V, E)$, initially

$$\begin{aligned} V &:= \{m[i] \mid m \in \mathrm{supp}(M) \ \wedge \ i \in \{1, \ldots, M(m)\}\} \\ E &:= \emptyset \end{aligned}$$

- Indexing of all instances (copies) $m[i]$ from a P module m allows unique identification
- Let $a = (a_\downarrow, a_\uparrow, a_\square) \in \mathrm{supp}(M)$ and $b = (b_\downarrow, b_\uparrow, b_\square) \in \mathrm{supp}(M)$ be two module instances derived from M.
- An edge $(a, b, R_{a\to b}) \in E$ denotes a connection from a to b where dedicated output species of a act as input species of b.
- Each edge comes with a binary relation $R_{a\to b} \subseteq a_\uparrow \times b_\downarrow$ in which the mapping of a's output species onto b's input species is given.
- $R_{a\to b}$ handled in an injective manner since one output species is allowed to cover several downstream input species, but each input species must be supplied by at most one upstream output species.

Connectivity Graph of a P Meta Framework

Each P Meta Framework $\Pi_{\pi\uparrow\downarrow} = (M, P)$ comes with a corresponding directed graph $G = (V, E)$, initially

$$\begin{aligned} V &:= \{m[i] \mid m \in \mathrm{supp}(M) \ \wedge \ i \in \{1, \ldots, M(m)\}\} \\ E &:= \emptyset \end{aligned}$$

- Indexing of all instances (copies) $m[i]$ from a P module m allows unique identification
- Let $a = (a_\downarrow, a_\uparrow, a_\square) \in \mathrm{supp}(M)$ and $b = (b_\downarrow, b_\uparrow, b_\square) \in \mathrm{supp}(M)$ be two module instances derived from M.
- An edge $(a, b, R_{a\to b}) \in E$ denotes a connection from a to b where dedicated output species of a act as input species of b.
- Each edge comes with a binary relation $R_{a\to b} \subseteq a_\uparrow \times b_\downarrow$ in which the mapping of a's output species onto b's input species is given.
- $R_{a\to b}$ handled in an injective manner since one output species is allowed to cover several downstream input species, but each input species must be supplied by at most one upstream output species.

Connectivity Graph of a P Meta Framework

Each P Meta Framework $\Pi_{\pi\uparrow\downarrow} = (M, P)$ comes with a corresponding directed graph $G = (V, E)$, initially

$$\begin{aligned} V &:= \{m[i] \mid m \in \mathrm{supp}(M) \ \wedge \ i \in \{1, \ldots, M(m)\}\} \\ E &:= \emptyset \end{aligned}$$

- Indexing of all instances (copies) $m[i]$ from a P module m allows unique identification
- Let $a = (a_\downarrow, a_\uparrow, a_\square) \in \mathrm{supp}(M)$ and $b = (b_\downarrow, b_\uparrow, b_\square) \in \mathrm{supp}(M)$ be two module instances derived from M.
- An edge $(a, b, R_{a\to b}) \in E$ denotes a connection from a to b where dedicated output species of a act as input species of b.
- Each edge comes with a binary relation $R_{a\to b} \subseteq a_\uparrow \times b_\downarrow$ in which the mapping of a's output species onto b's input species is given.
- $R_{a\to b}$ handled in an injective manner since one output species is allowed to cover several downstream input species, but each input species must be supplied by at most one upstream output species.

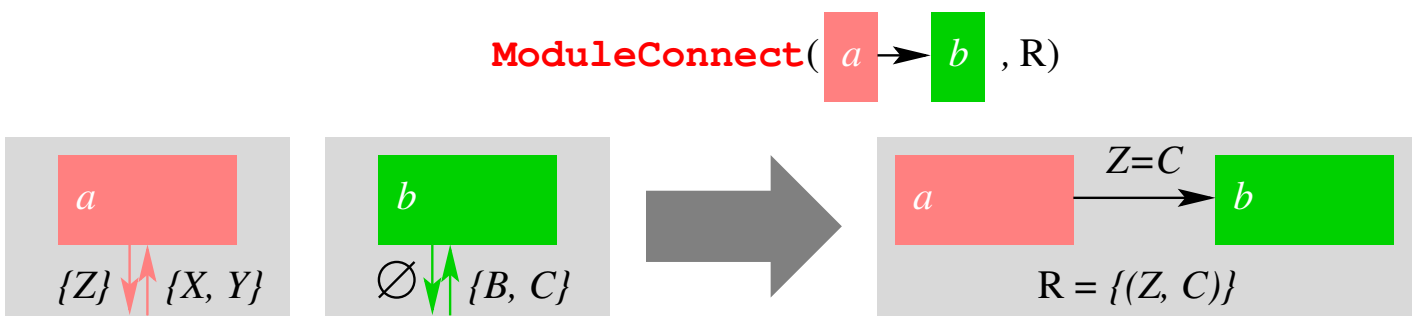
Assembly Programme Instructions (Selection)

- Time-dependent changes of module connectivity
- Time stamp t opens each instruction
- Instruction updates graph $G(V, E)$

t : `ModuleConnect`$(a \to b, R_{a\to b})$

connects some or all of module a's output species to represent b's input species by sharing species identifiers according to the injective binary relation $R_{a\to b} \subseteq a_{\uparrow} \times b_{\downarrow}$.

Edge update scheme: $E := E \cup \{(a, b, R_{a\to b})\}$



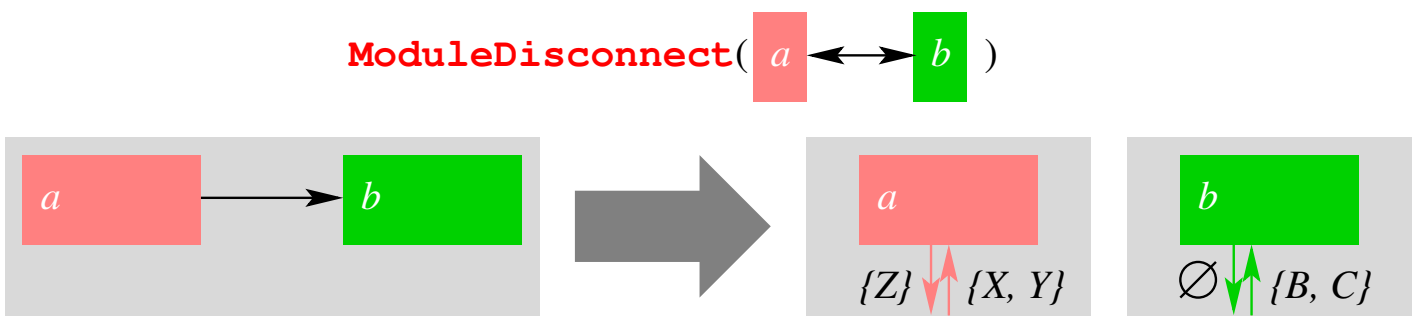
Assembly Programme Instructions (Selection)

- Time-dependent changes of module connectivity
- Time stamp t opens each instruction
- Instruction updates graph $G(V, E)$

t : `ModuleDisconnect`$(a \leftrightarrow b)$

completely disconnects modules a and b by annihilating all cross-modular species sharings. This comes along with removing $R_{a\to b}$ as well as $R_{b\to a}$, respectively.

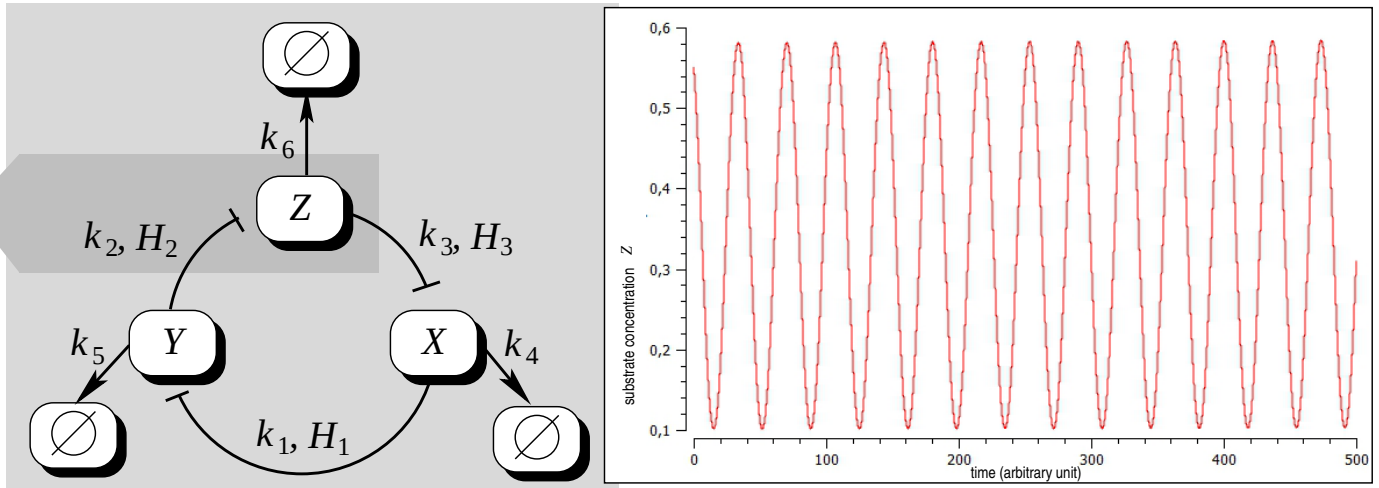
Edge update scheme: $E := E \setminus \{(a, b, R_{a\to b})\} \setminus \{(b, a, R_{b\to a})\}$



1. Biological Rhythms
2. P Meta Framework
3. **Assembly of Frequency Dividers**
   - Chemical frequency divider model 1:17
   - Frequency divider 1:5 by separator removal
   - Frequency divider 1:6 by Repressilator
   - Frequency divider 1:3 by Goodwin module
4. Suprachiasmatic Nucleus

The Repressilator Module

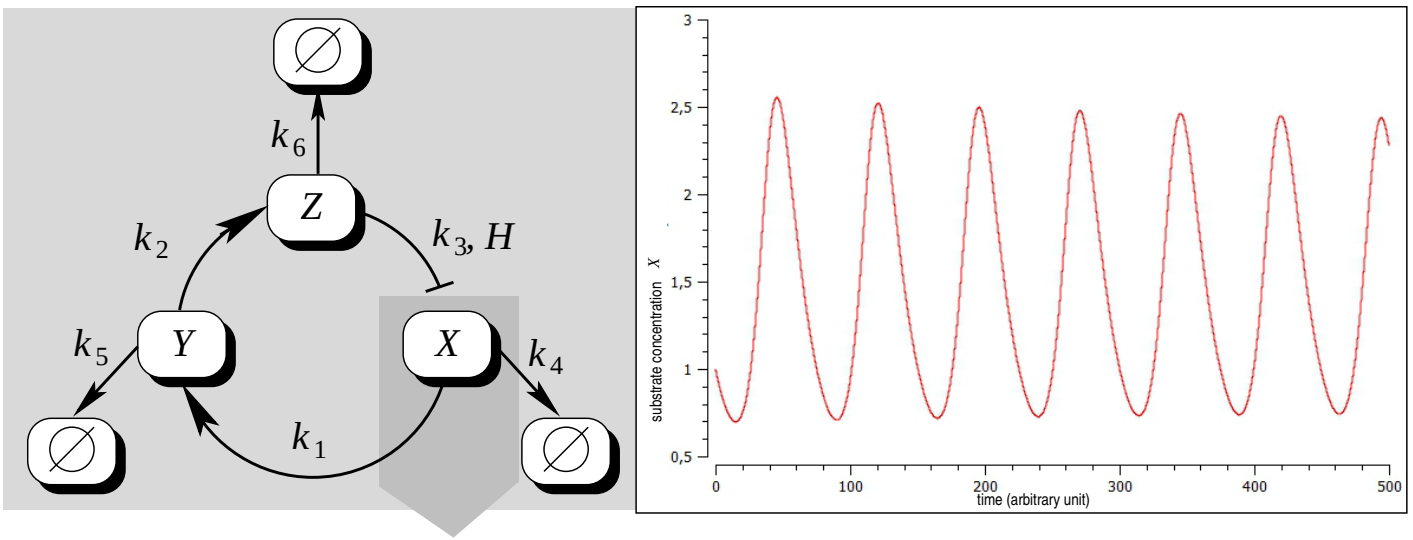
$$\text{repressilator} = (\emptyset, \{Z\}, F)$$



- Oscillation by progressional inhibition in gene regulatory network
- Involves higher-order Hill kinetics
- Velocity of decay reactions mainly determines frequency
- Inherently almost sinusoidal waveform

The Goodwin Module

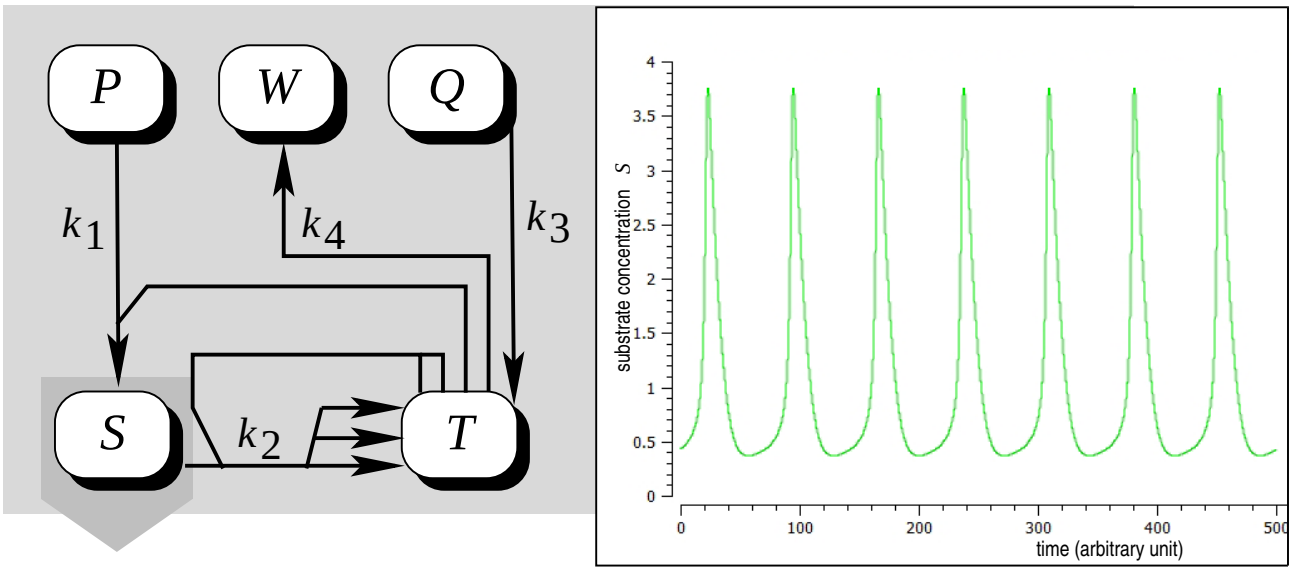
$$\text{goodwin} = (\emptyset, \{X\}, F)$$



- Oscillation by mutual activation and inhibition in gene regulatory network
- Involves higher-order Hill kinetics
- Velocity of decay reactions mainly determines frequency
- Configurable to exhibit almost sinusoidal or plated waveforms

The Brusselator Module

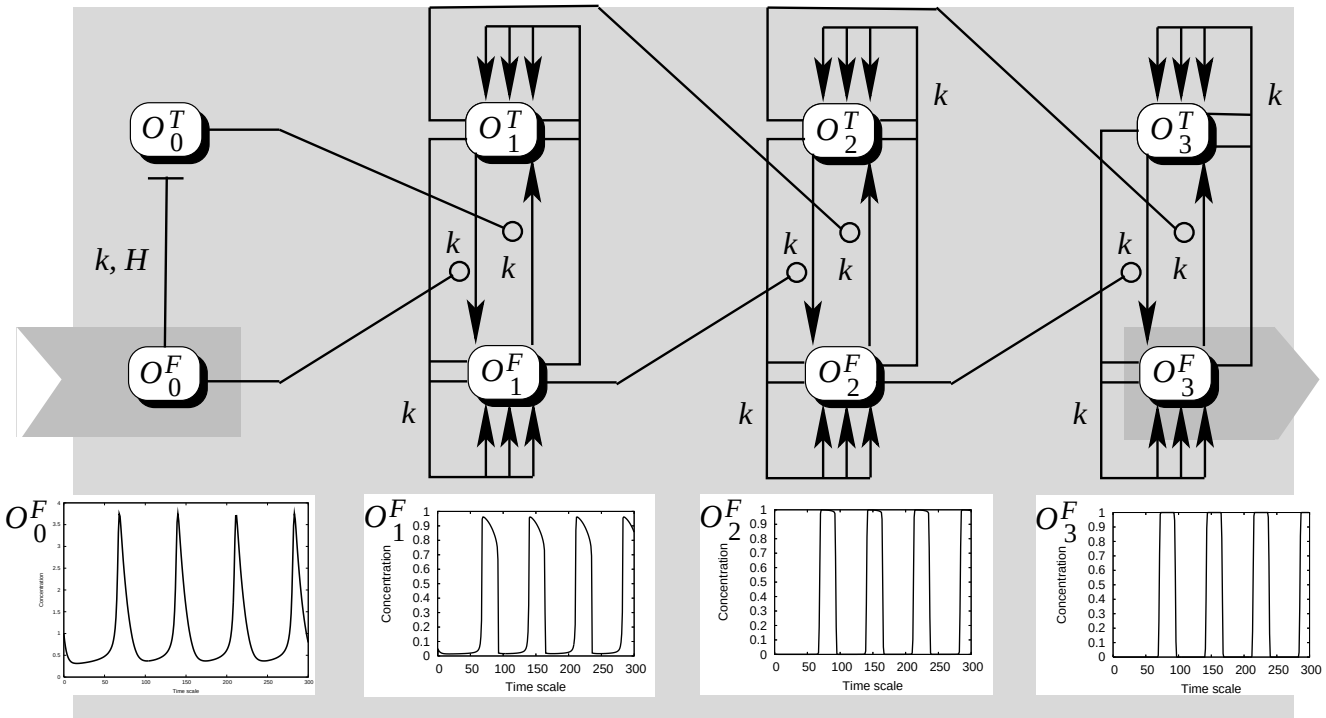
$$\text{brusselator} = (\emptyset, \{S\}, F)$$



- Oscillation by autocatalytic feedback loops
- Exclusively mass-action kinetics
- Velocity of decay in concert with autocatalytic loop determines frequency
- Inherently spiking waveform

The Binary Signal Separator Module

$$\text{separator} = (\{O_0^F\}, \{O_3^F\}, F)$$



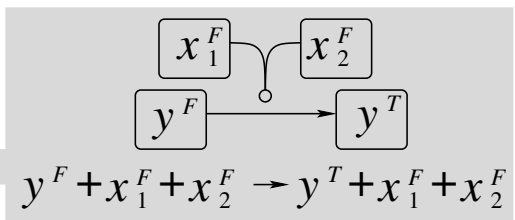
- Converts spiking or sinusoidal oscillations into plated waveform
- Successive binarisation by enzymatically controlled cascade
- Exclusively mass-action kinetics

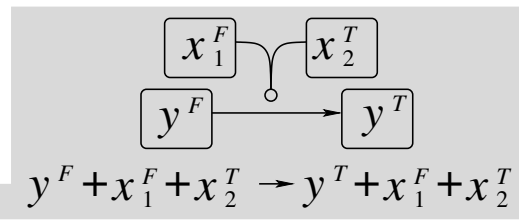
Binary Counter Modulo 17 – Chemical Logic gate

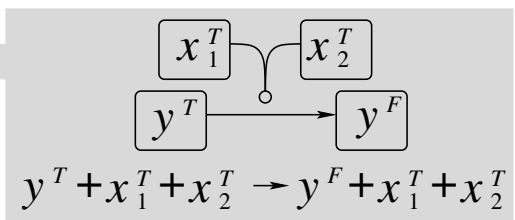
Boolean variable z represented by two correlated species Z^T and Z^F

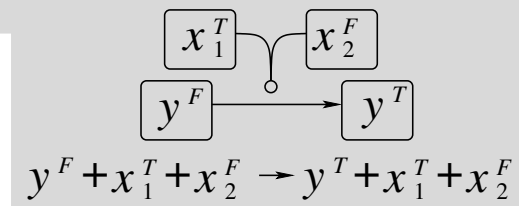
Chemical reaction network for NAND

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0





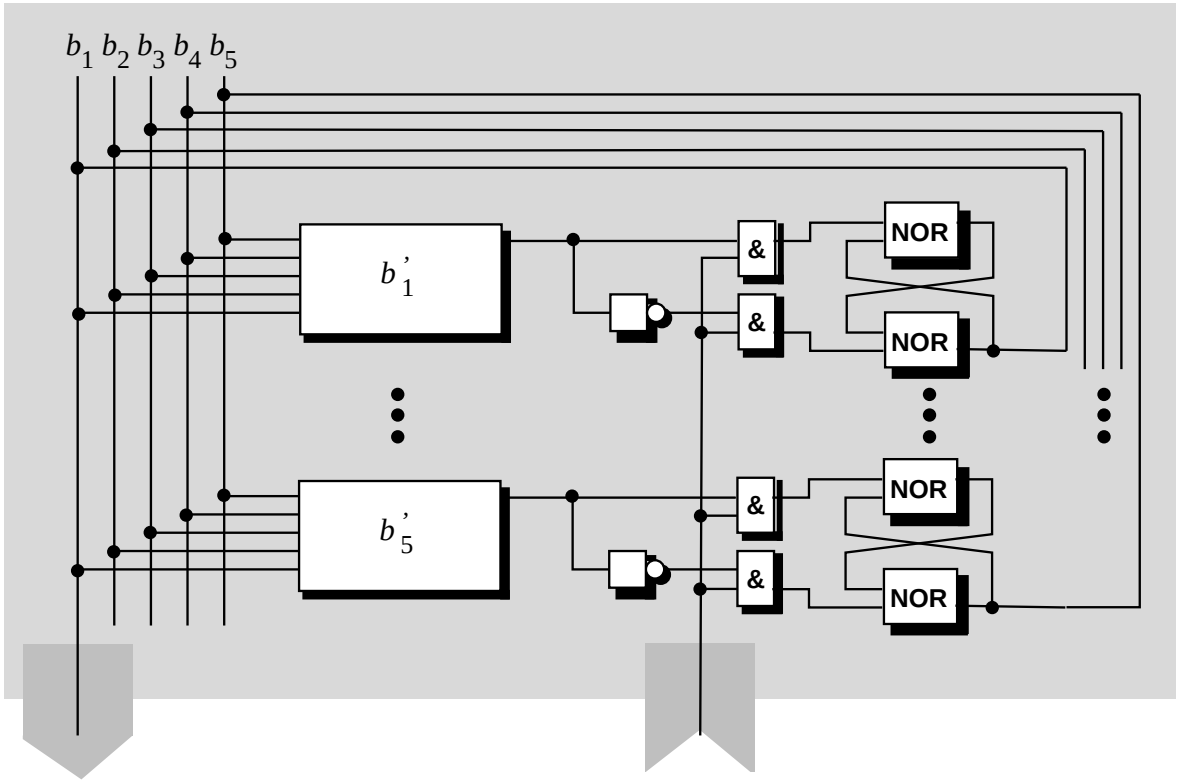




Analogously implementation of other boolean functions

The Binary Counter Modulo 17

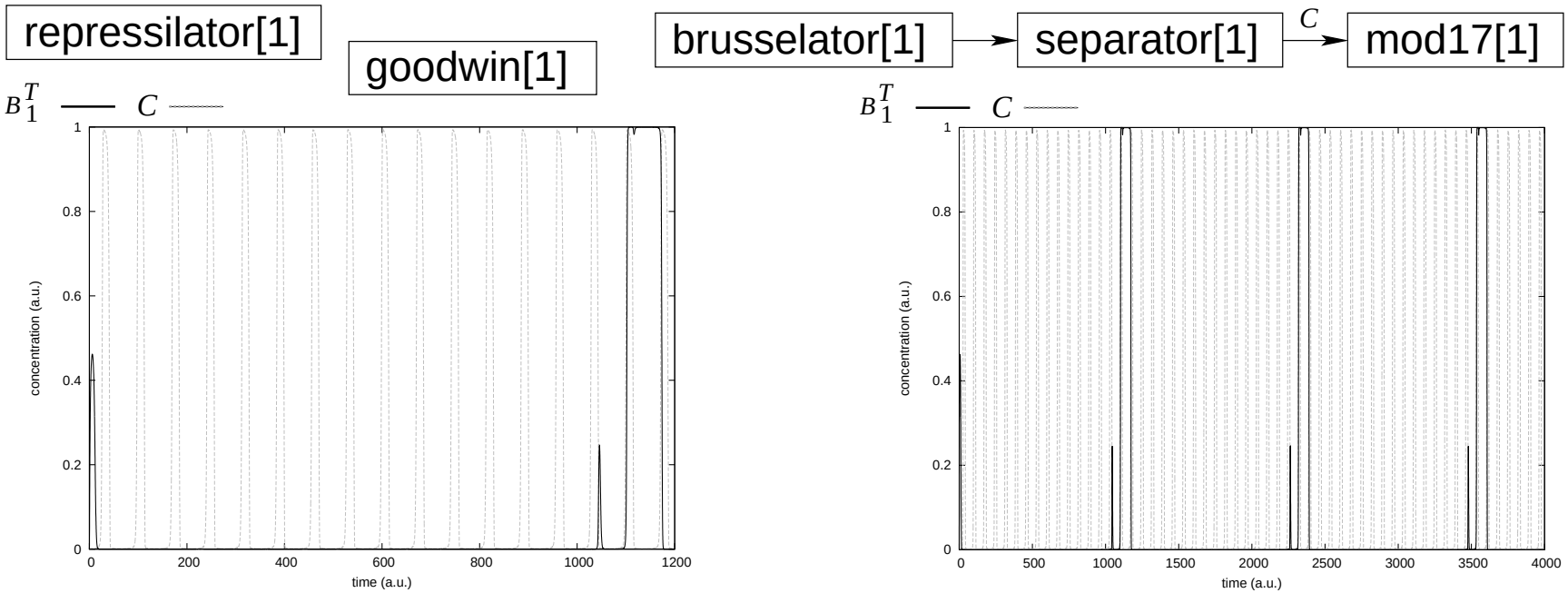
$$\mathrm{mod}17 = (\{C\}, \{B_1^T\}, F)$$



$$
\begin{aligned}
b'_1 &= \bar{b}_1 b_2 \bar{b}_3 \bar{b}_4 \bar{b}_5 \\
b'_2 &= \bar{b}_1 b_2 \vee \bar{b}_1 b_2 \bar{b}_3 \bar{b}_4 \bar{b}_5 \\
b'_3 &= \bar{b}_1 \bar{b}_2 b_3 \vee \bar{b}_1 b_3 \bar{b}_4 \vee \\
&\quad \bar{b}_1 b_2 b_3 b_5 \vee \bar{b}_1 \bar{b}_2 b_4 \bar{b}_5 \\
b'_4 &= \bar{b}_1 b_4 \bar{b}_5 \vee \bar{b}_1 b_2 b_3 b_5 \vee \\
&\quad \bar{b}_1 \bar{b}_2 \bar{b}_3 b_5 \vee \bar{b}_1 \bar{b}_2 \bar{b}_3 b_4 \vee \\
&\quad \bar{b}_1 b_2 b_3 b_4 \\
b'_5 &= \bar{b}_1 \bar{b}_2 \bar{b}_3 \bar{b}_4 \vee \bar{b}_1 \bar{b}_2 b_3 b_4 \vee \\
&\quad \bar{b}_1 b_2 b_3 \bar{b}_4 \vee \bar{b}_1 b_2 \bar{b}_3 b_4
\end{aligned}
$$

- Based on 5-bit Gray code
- Clock signal C increments counter
- Exclusively mass-action kinetics

Composing the Original Frequency Divider 1:17

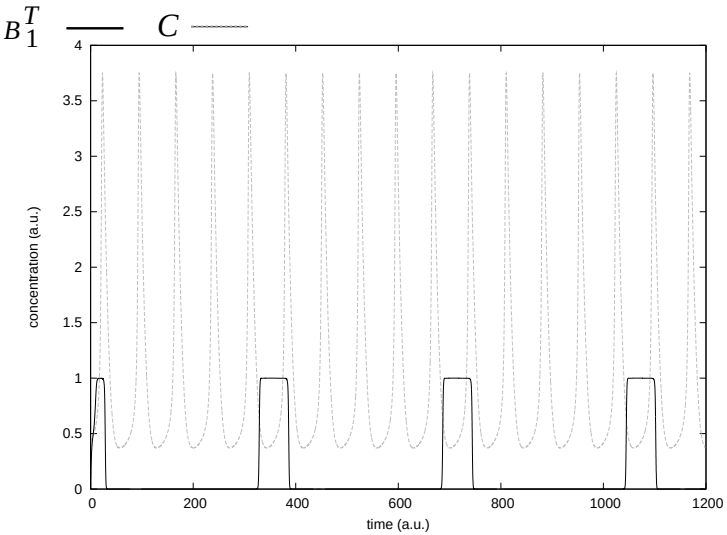


$$
\begin{aligned}
\Pi_{\mathrm{FD17}} &= (M, P) \quad \text{with} \\
M &= \{(\text{brusselator}, 1), (\text{repressilator}, 1), (\text{goodwin}, 1), (\text{separator}, 1), (\text{mod17}, 1)\} \\
P &= \{0 : \texttt{ModuleConnect}(\text{brusselator}[1] \rightarrow \text{separator}[1], \{(S, O_0^F)\}), \\
&\quad\ 0 : \texttt{ModuleConnect}(\text{separator}[1] \rightarrow \text{mod17}[1], \{(O_3^F, C)\})\}
\end{aligned}
$$

Frequency Divider 1:5 by Removal of Signal Separator

repressilator[1] goodwin[1] separator[1] brusselator[1] $\xrightarrow{C}$ mod17[1]

B_1^T	B_2^T	B_3^T	B_4^T	B_5^T
0	0	0	1	1
0	0	1	1	1
0	1	1	1	1
0	1	0	1	1
1	1	0	0	0



$$\Pi_{FD5} = (M, P) \quad \text{with}$$

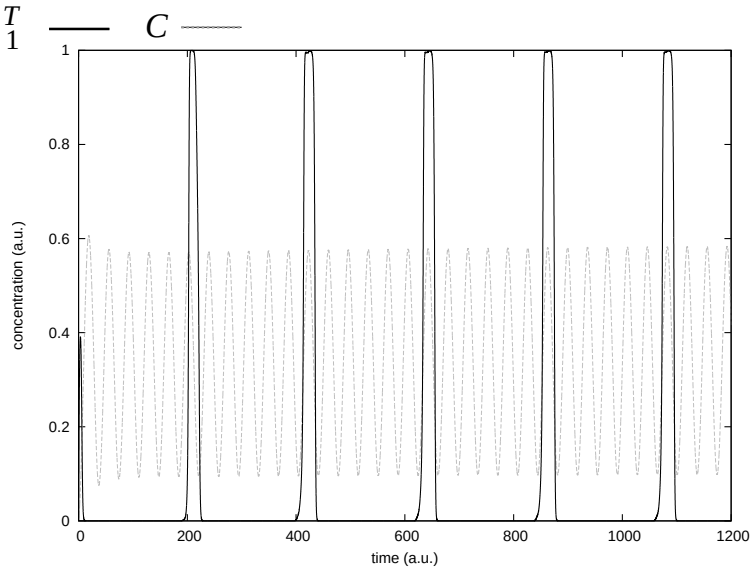
$$M = \{(\text{brusselator}, 1), (\text{repressilator}, 1), (\text{goodwin}, 1), (\text{separator}, 1), (\text{mod17}, 1)\}$$

$$\begin{aligned} P = \{ & 0 : \texttt{ModuleConnect}(\text{brusselator}[1] \rightarrow \text{separator}[1], \{(S, O_0^F)\}), \\ & 0 : \texttt{ModuleConnect}(\text{separator}[1] \rightarrow \text{mod17}[1], \{(O_3^F, C)\}), \\ & 200 : \texttt{ModuleDisconnect}(\text{brusselator}[1] \leftrightarrow \text{separator}[1]), \\ & 200 : \texttt{ModuleConnect}(\text{brusselator}[1] \rightarrow \text{mod17}[1], \{(S, C)\})\} \end{aligned}$$

Frequency Divider 1:6 by Repressilator

goodwin[1] brusselator[1] separator[1] repressilator[1] $\xrightarrow{C}$ mod17[1]

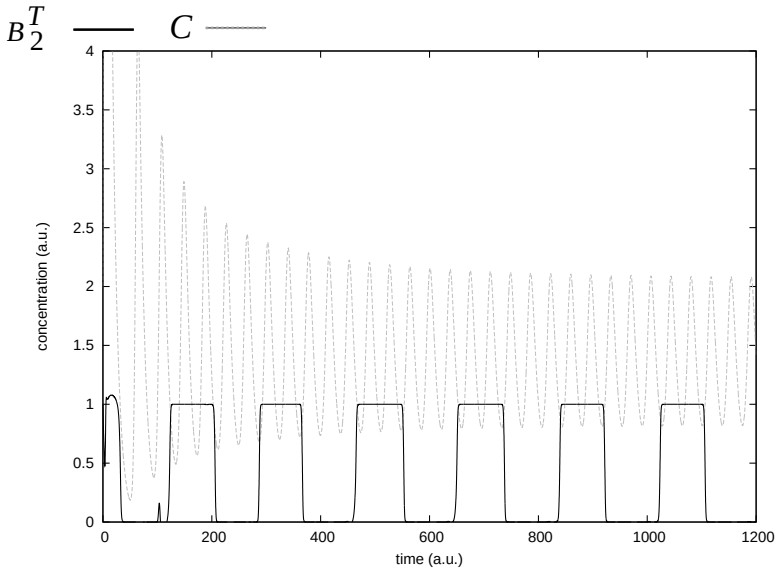
B_1^T	B_2^T	B_3^T	B_4^T	B_5^T
0	0	0	1	1
0	0	1	1	1
0	1	1	0	0
0	1	1	1	0
0	1	0	0	1
1	0	0	0	0



$$\Pi_{\mathrm{FD6}} = (M, P) \quad \text{with}$$
$$M = \{(\text{brusselator}, 1), (\text{repressilator}, 1), (\text{goodwin}, 1), (\text{separator}, 1), (\text{mod17}, 1)\}$$
$$P = \{0 : \texttt{ModuleConnect}(\text{repressilator}[1] \rightarrow \text{mod17}[1], \{(Z, C)\})\}$$

Frequency Divider 1:3 by Goodwin Module

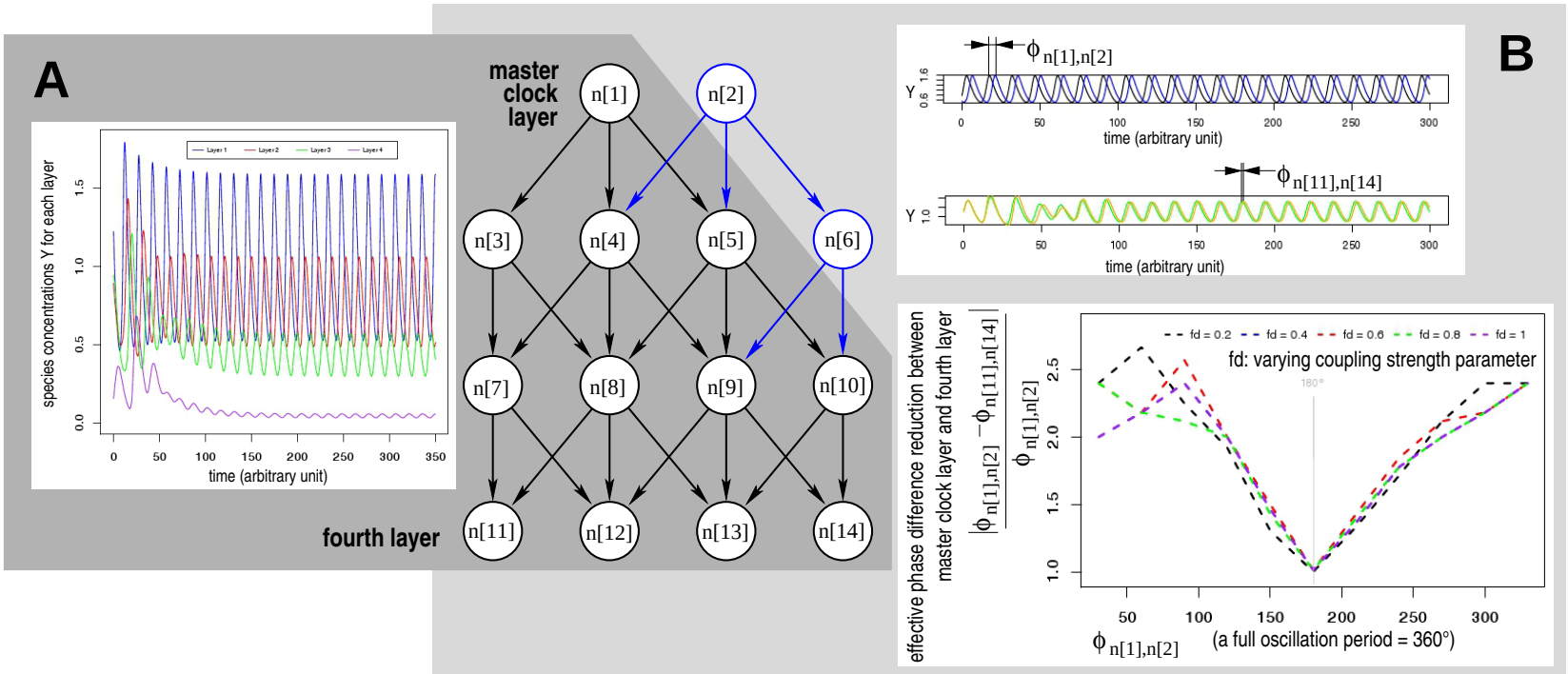
B_1^T	B_2^T	B_3^T	B_4^T	B_5^T
0	0	1	1	0
0	1	1	1	1
1	1	0	0	0



$$\Pi_{FD3} = (M, P) \quad \text{with}$$
$$M = \{(\text{brusselator}, 1), (\text{repressilator}, 1), (\text{goodwin}, 1), (\text{separator}, 1), (\text{mod17}, 1)\}$$
$$P = \{0 : \texttt{ModuleConnect}(\text{goodwin}[1] \rightarrow \text{mod17}[1], \{(X, C)\})\}$$

1. Biological Rhythms
2. P Meta Framework
3. Assembly of Frequency Dividers
4. **Suprachiasmatic Nucleus**
   - Core Oscillator's Interplay in Suprachiasmatic Nucleus
   - Prospectives
   - Acknowledgement

Core Oscillator's Interplay in Suprachiasmatic Nucleus



$$
\begin{aligned}
\Pi_{\mathrm{SCN}} &= (M, P) \quad \text{with} \\
M &= \{(\mathrm{n}, 14)\} \\
P &= \{0 : \texttt{ModuleConnect}(\mathrm{n}[1] \to \mathrm{n}[3], \{(Y, X)\}), \\
&\quad\ 0 : \texttt{ModuleConnect}(\mathrm{n}[1] \to \mathrm{n}[4], \{(Y, X)\}), \ldots, \\
&\quad\ 0 : \texttt{ModuleConnect}(\mathrm{n}[10] \to \mathrm{n}[14], \{(Y, X)\}) \\
&\quad\ 300 : \texttt{ModuleConnect}(\mathrm{n}[2] \to \mathrm{n}[4], \{(Y, X)\}), \ldots, \\
&\quad\ 300 : \texttt{ModuleConnect}(\mathrm{n}[6] \to \mathrm{n}[10], \{(Y, X)\})\}
\end{aligned}
$$

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Further work

- Incorporation of Infobiotics workbench
- Extension of P module library
- Combining advantages of P system-based approaches with analytical examination tools using ODEs

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Further work

- Incorporation of Infobiotics workbench
- Extension of P module library
- Combining advantages of P system-based approaches with analytical examination tools using ODEs

Prospectives

Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Further work

- Incorporation of Infobiotics workbench
- Extension of P module library
- Combining advantages of P system-based approaches with analytical examination tools using ODEs

Prospectives

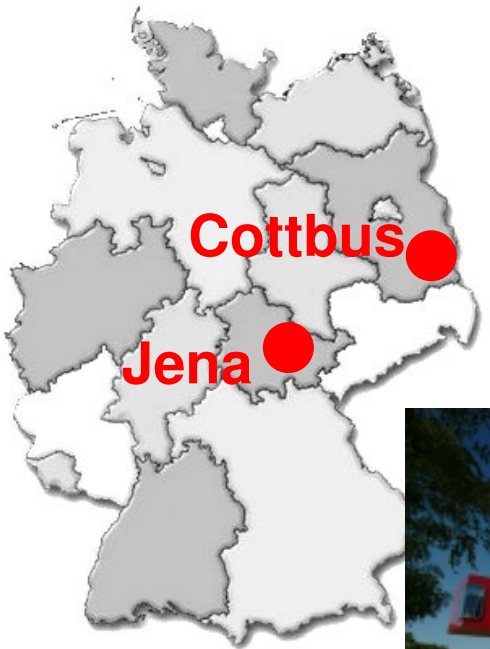
Conclusions

- Assembly and reassembly as well as composition and decomposition of pre-defined reaction network modules on the fly $\longrightarrow$ promising strategy in order to achieve complex systems capable of new, extended, or unexpected functionality
- Framework for an artificial evolution at a granularity of highly conserved genetic ensembles
- P system-based approach features by coping with dynamical structures at a modular level rather than a molecular level
- Simulation studies carried out using CoPaSi along with Gepasi Model Extractor

Further work

- Incorporation of Infobiotics workbench
- Extension of P module library
- Combining advantages of P system-based approaches with analytical examination tools using ODEs

Acknowledgements



JCB Jena Centre for Bioinformatics

Jena Centre for Bioinformatics

b-tu

Brandenburg University of Technology Cottbus

DFG

Deutsche Forschungs-gemeinschaft, grant Hi801/3-1