# The Continuous $\pi$-Calculus: A Process Algebra for Biochemical Modelling

Ian Stark and Marek Kwiatkowski

Laboratory for Foundations of Computer Science
School of Informatics
The University of Edinburgh

Friday 26 February 2010

http://homepages.ed.ac.uk/stark/cpi

# Overview

The *continuous $\pi$-calculus* ($c\pi$) is a process algebra for modelling behaviour and variation in molecular systems.

It has a structured operational semantics that captures system behaviour as trajectories through a continuous process space, by generating familiar differential-equation models.

We have existing biochemical systems expressed in $c\pi$; the aim is to use this to investigate evolutionary properties of biochemical pathways.

Marek Kwiatkowski and Ian Stark.

The Continuous $\pi$-Calculus: A Process Algebra for Biochemical Modelling. In *Computational Methods in Systems Biology: Proc. CMSB 2008*

Lecture Notes in Computer Science 5307, pages 103–122. Springer 2008

# Overview

## Contents

- Systems Biology and Process Algebras
- The Continuous $\pi$-Calculus
- Example: Circadian Rhythms in *Synechococcus Elongatus*

# Systems Biology

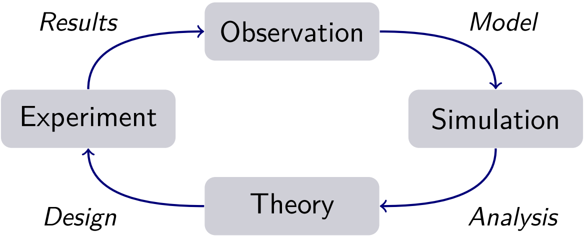
Biology is the study of living organisms; Systems Biology is the study of the dynamic processes that take place within those organisms.

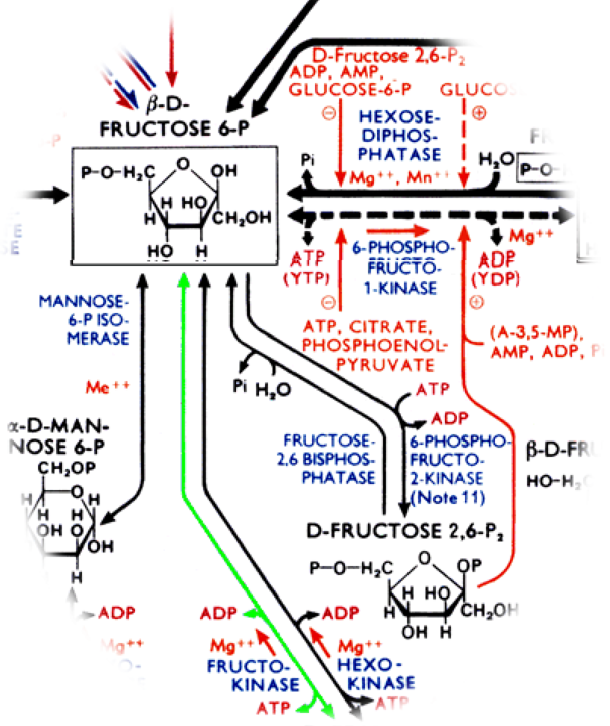
In particular:

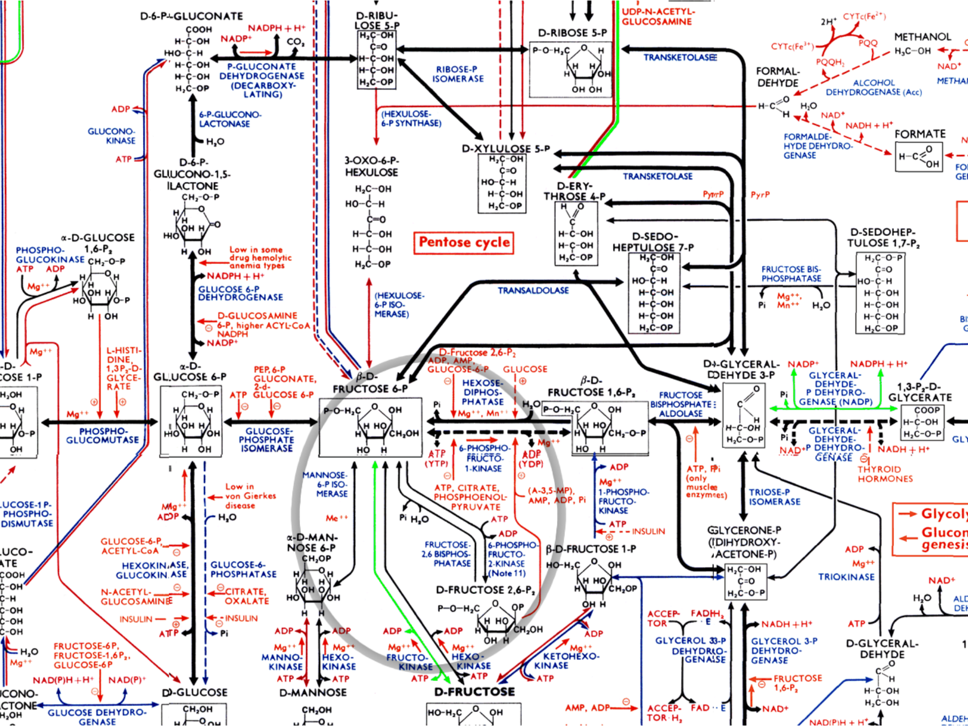
- Interaction between processes;
- Behaviour emerging from such interaction; and
- Integration of component behaviours.

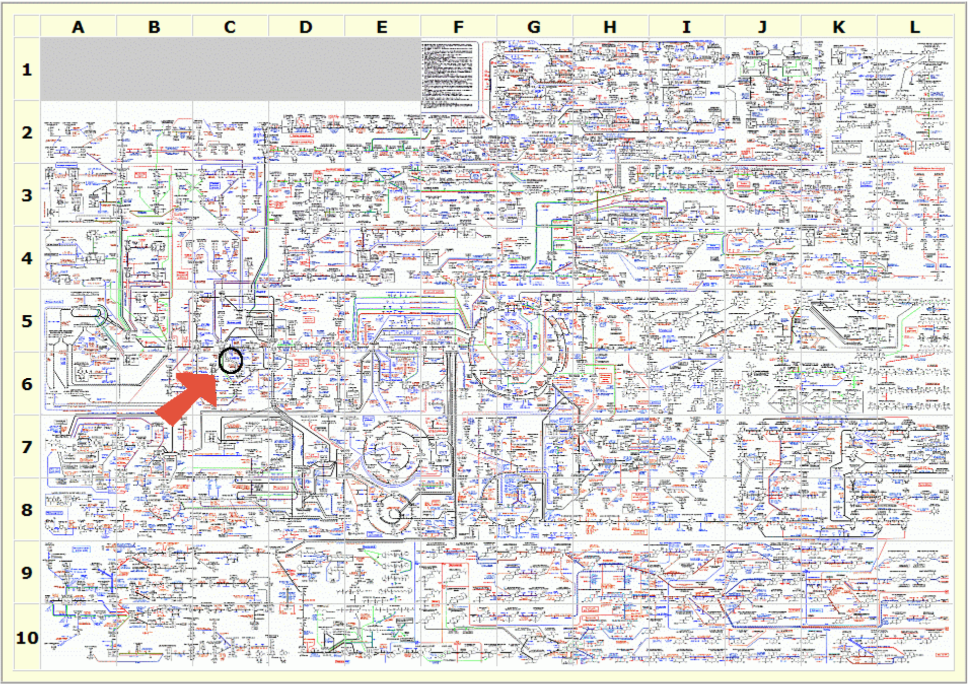
# Systems Biology

Biology is the study of living organisms; Systems Biology is the study of the dynamic processes that take place within those organisms.




β-D-
FRUCTOSE 6-P
P–O–H₂C
O
OH
H HO
H
CH₂OH
HO
H
D-Fructose 2,6-P₂
ADP, AMP,
GLUCOSE-6-P
GLUCOS
⊖
⊕
HEXOSE-
DIPHOS-
PHATASE
Pi
H₂O
Mg⁺⁺, Mn⁺⁺
P-O-
Mg⁺⁺
ATP
(YTP)
6-PHOSPHO-
FRUCTO-
1-KINASE
ADP
(YDP)
⊖
⊕
ATP, CITRATE,
PHOSPHOENOL-
PYRUVATE
(A-3,5-MP),
AMP, ADP, P
MANNOSE-
6-P ISO-
MERASE
Me⁺⁺
Pi
H₂O
ATP
ADP
α-D-MAN-
NOSE 6-P
CH₂OP
H
O
H
H
H
OH
O
OH
H
FRUCTOSE-
2,6 BISPHOS-
PHATASE
6-PHOSPHO-
FRUCTO-
2-KINASE
(Note 11)
β-D-FR
HO–H₂
D-FRUCTOSE 2,6-P₂
P–O–H₂C
O
OP
H HO
H
CH₂OH
OH
H
ADP
Mg⁺⁺
ADP
Mg⁺⁺
FRUCTO-
KINASE
ATP
ADP
Mg⁺⁺
HEXO-
KINASE
ATP


D-6-P-GLUCONATE
NADPH + H⁺
NADP⁺
CO₂
P-GLUCONATE DEHYDROGENASE (DECARBOXYLATING)
D-RIBULOSE 5-P
D-RIBOSE 5-P
UDP-N-ACETYL-GLUCOSAMINE
TRANSKETOLASE
RIBOSE-P ISOMERASE
2H⁺
CYTc(Fe²⁺)
CYTc(Fe³⁺)
PQQ
PQQH₂
METHANOL
NAD⁺
ALCOHOL DEHYDROGENASE (Acc)
FORMALDEHYDE
H₂O
NAD⁺
NADH + H⁺
FORMALDEHYDE DEHYDROGENASE
FORMATE
ADP
GLUCONOKINASE
ATP
6-P-GLUCONOLACTONASE
H₂O
D-6-P-GLUCONO-1,5-LACTONE
(HEXULOSE-6-P SYNTHASE)
3-OXO-6-P-HEXULOSE
D-XYLULOSE 5-P
TRANSKETOLASE
D-ERYTHROSE 4-P
PyrP
PyrP
Pentose cycle
D-SEDOHEPTULOSE 7-P
D-SEDOHEPTULOSE 1,7-P₂
FRUCTOSE BISPHOSPHATASE
Mg⁺⁺, Mn⁺⁺
Pi
H₂O
TRANSALDOLASE
α-D-GLUCOSE 1,6-P₂
PHOSPHOGLUCOKINASE
ATP
ADP
Mg⁺⁺
Low in some drug hemolytic anemia types
NADPH + H⁺
GLUCOSE 6-P DEHYDROGENASE
D-GLUCOSAMINE 6-P, higher ACYL-CoA NADPH
NADP⁺
(HEXULOSE-6-P ISOMERASE)
α-D-GLUCOSE 1-P
Mg⁺⁺
L-HISTIDINE, 1,3P₂-D-GLYCERATE
Mg⁺⁺
PHOSPHOGLUCOMUTASE
α-D-GLUCOSE 6-P
PEP, 6-P GLUCONATE, 2-d-GLUCOSE 6-P
ATP
GLUCOSEPHOSPHATE ISOMERASE
β-D-FRUCTOSE 6-P
D-Fructose 2,6-P₂
ADP, AMP, GLUCOSE-6-P
GLUCOSE
HEXOSEDIPHOSPHATASE
Mg⁺⁺, Mn⁺⁺
Pi
H₂O
β-D-FRUCTOSE 1,6-P₂
FRUCTOSE BISPHOSPHATE ALDOLASE
D-GLYCERALDEHYDE 3-P
NADP⁺
NADPH + H⁺
GLYCERALDEHYDE-P DEHYDROGENASE (NADP)
Pi
1,3-P₂-D-GLYCERATE
GLYCERALDEHYDE-P DEHYDROGENASE
Pi
NAD⁺
NADH + H⁺
THYROID HORMONES
ATP (YTP)
6-PHOSPHOFRUCTO-1-KINASE
ADP (YDP)
Mg⁺⁺
ATP, CITRATE, PHOSPHOENOLPYRUVATE
(A-3,5-MP), AMP, ADP, Pi
ATP, Pi (only muscle enzymes)
GLUCO-1P-PHOSPHODISMUTASE
Mg⁺⁺
ADP
Low in von Gierkes disease
H₂O
MANNOSE-6-P ISOMERASE
Me⁺⁺
Pi
H₂O
ATP
ADP
6-PHOSPHOFRUCTO-2-KINASE (Note 11)
FRUCTOSE-2,6 BISPHOSPHATASE
D-FRUCTOSE 2,6-P₂
ADP
Mg⁺⁺
1-PHOSPHOFRUCTOKINASE
ATP
INSULIN
TRIOSE-P ISOMERASE
GLYCERONE-P (DIHYDROXYACETONE-P)
Glycolysis
Gluconeogenesis
GLUCOSE-6-P, ACETYL-CoA
HEXOKINASE, GLUCOKINASE
GLUCOSE-6-PHOSPHATASE
N-ACETYL-GLUCOSAMINE
CITRATE, OXALATE
INSULIN
INSULIN
ATP
Pi
α-D-MANNOSE 6-P
β-D-FRUCTOSE 1-P
ADP
Mg⁺⁺
TRIOKINASE
NAD⁺
H₂O
ALDEHYDE DEHYDROGENASE
GLUCONATE
H₂O
Mg⁺⁺
FRUCTOSE-6P, FRUCTOSE-1,6P₂, GLUCOSE-6P
NAD(P)H + H⁺
NAD(P)⁺
GLUCOSE DEHYDROGENASE
GLUCONOLACTONE
D-GLUCOSE
ADP
Mg⁺⁺
MANNOKINASE
ATP
ADP
Mg⁺⁺
HEXOKINASE
ATP
D-MANNOSE
ADP
Mg⁺⁺
FRUCTOKINASE
ATP
ADP
Mg⁺⁺
HEXOKINASE
ATP
D-FRUCTOSE
ADP
Mg⁺⁺
KETOHEXOKINASE
ATP
ACCEPTOR
FADH₂
E
GLYCEROL 3-P DEHYDROGENASE
ACCEPTOR·H₂
FAD·E
AMP, ADP
NADH + H⁺
GLYCEROL 3-P DEHYDROGENASE
FRUCTOSE 1,6-P₂
NAD⁺
ATP
D-GLYCERALDEHYDE
NAD(P)H + H⁺
ALDEHYDE


A
B
C
D
E
F
G
H
I
J
K
L
1
2
3
4
5
6
7
8
9
10

# What can Computer Science do for Systems Biology?

## Machines

Large Databases: Semistructured data; data integration; data mining

Large Simulations: Experiments *in silico*; parameter scans; folding search

## Ideas

Language: Abstraction; modularity; semantics; formal models

Reasoning: Logics; behavioural description; model checking

# Scope of Study

## Processes

- Metabolic networks
- Regulatory systems: promotion, inhibition
- Signalling pathways
- Gene expression: translation, transcription

## Models

- Discrete time, continuous time
- Discrete space, continuous space
- Deterministic, nondeterministic, probabilistic
- Qualitative, quantitative

# Biochemical Simulation

Biologists routinely use one of two alternative approaches to computational modelling of biochemical systems:

- Stochastic simulation
  - Discrete behaviour: tracking individual molecules
  - Randomized: Gillespie's algorithm
- Ordinary Differential Equations
  - Continuous behaviour: chemical concentrations
  - Deterministic: Numerical ODE solutions

The classical approach is to use the mathematics directly as the target formal system. However, experience in Computer Science suggests the value of an intermediate *language* to describe a system. An expression in this language can then be analysed as it stands, or further mapped into (one or more) mathematical representations.

# Process Algebras in Systems Biology

- Petri nets
- $\pi$-calculus; stochastic $\pi$; BioSPI; SPiM
- Beta binders; BlenX
- Ambients, bioAmbients
- Brane calculi; Bitonal systems
- PEPA, bioPEPA
- Kappa
- PRISM
- Pathway Logic
- ...

# The Continuous $\pi$-Calculus

The *Continuous $\pi$-Calculus* ($\mathrm{c}\pi$) is a process algebra for modelling behaviour and variation in molecular systems.

Based on the $\pi$-calculus, it introduces continuous variability in:

- rates of reaction;
- affinity between interacting names; and
- quantities of processes.

while retaining classic process-algebra features of:

- compositional semantics (modular, not monolithic);
- abstraction (separating language and semantics);
- specifying interaction (taking behaviour as it emerges).

Motivated by Fontana's work on evolutionary change, neutral spaces and the "topology of the possible".

# Basics of $c\pi$

Continuous $\pi$ has two levels of system description:

- Species
  - Individual molecules (proteins)
  - Transition system semantics
- Processes
  - Bulk population (concentration)
  - Differential equations

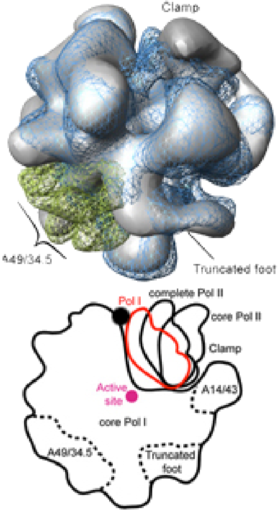
*Process space* arises as a real-valued vector space over species, with each point the state of a system and behaviours as trajectories through that.

# Names in $c\pi$

As in standard $\pi$-calculus, *names* indicate a potential for interaction: for example, the docking sites on an enzyme where other molecules may attach.

These sites may interact with many different other sites, to different degrees.

This variation is captured by an *affinity network*: a graph setting out the interaction potential between different names.

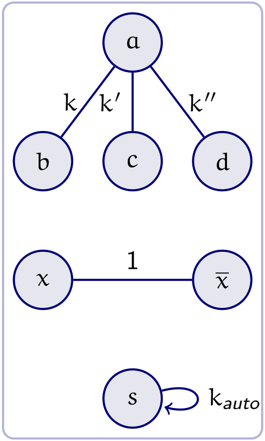


# Names in $c\pi$

As in standard $\pi$-calculus, *names* indicate a potential for interaction: for example, the docking sites on an enzyme where other molecules may attach.

These sites may interact with many different other sites, to different degrees.

This variation is captured by an *affinity network*: a graph setting out the interaction potential between different names.

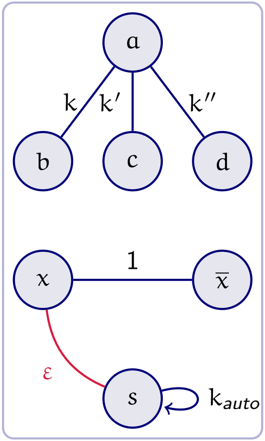


# Names in $c\pi$

As in standard $\pi$-calculus, *names* indicate a potential for interaction: for example, the docking sites on an enzyme where other molecules may attach.

These sites may interact with many different other sites, to different degrees.

This variation is captured by an *affinity network*: a graph setting out the interaction potential between different names.



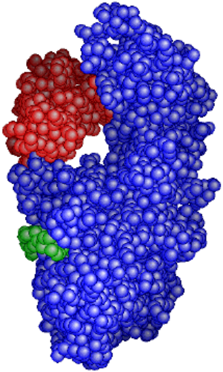
# Restriction in $c\pi$

Name restriction $\nu x(A \mid B)$ captures molecular *complexes*, with local name $x$ mediating further internal modification, or decomplexation.

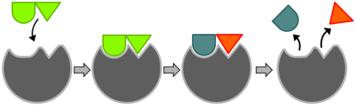
The binder can be a single local name $(\nu x.-)$, or several names with their own affinity network $(\nu M.-)$.

As in the classic $\pi$-calculus "cocktail party" model, interacting names can communicate further names, allowing further interactions.

In particular, we use name *extrusion* to model complex formation.



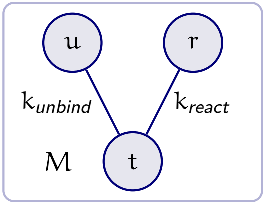
# Example Species: Enzyme Catalysis

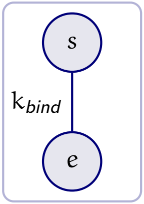


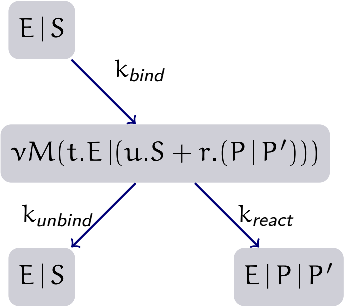
$$S = s(x, y).(x.S + y.(P|P'))$$

$$E = \nu(u, r, t{:}M).(e\langle u, r\rangle.t.E)$$

$$P = P' = \tau@k_{degrade}.0$$







$E \,|\, S$

$k_{bind}$

$\nu M(t.E \,|(u.S + r.(P \,|\, P')))$

$k_{unbind}$

$k_{react}$

$E \,|\, S$

$E \,|\, P \,|\, P'$

# Species

Species $A, B ::= 0 \mid S(\vec{a}) \mid \Sigma a(\vec{b}; \vec{y}).A \mid \tau@k.A \mid A|B \mid \nu M.A$

| | |
|---|---|
| Symmetric prefix | $a(b, c; x, y).A$ for two-way communication. |
| Guarded sums | $\Sigma_i \alpha_i.A$ or $\alpha.A + \alpha'.A'$ for alternative choices. |
| Silent transition | $\tau@k.A$ for constitutive reactions at rate $k \in \mathbb{R}_{\geqslant 0}$. |
| Parallel composition | $A|B$ within complexes. |
| Restriction | $\nu M.A$ for intra-complex reaction. |
| Recursion | via guarded species definitions $S(\vec{x}) = \dots$ |

Set $\mathcal{S}$ of species up to structural congruence, and $\mathcal{S}^{\#}$ of *prime* species.

# Operational Semantics for Species

The behaviour of a species is given by transitions:

$A \xrightarrow{a} (\vec{b};\vec{y})B$ Potential interaction

$A \xrightarrow{\tau@k} B$ Immediate action (fixed rate)

$A \xrightarrow{\tau\langle x,y\rangle} B$ Internal action (rate tbd)

Here $(\vec{b};\vec{y})B$ is a *concretion* representing potential interaction; the result of actual interaction is given by pseudo application:

$$(\vec{a};\vec{x})A \circ (\vec{b};\vec{y})B \;=\; A\{\vec{b}/\vec{x}\} \mid B\{\vec{a}/\vec{y}\}$$

Rules for deriving transitions give a structural operational semantics:

$$\frac{A \xrightarrow{a} F \quad B \xrightarrow{b} G}{A \mid B \xrightarrow{\tau\langle a,b\rangle} F \circ G} \qquad \frac{A \xrightarrow{\tau\langle a,b\rangle} B \quad a,b \in M}{\nu M.A \xrightarrow{\tau@M(a,b)} B} \qquad \ldots$$

# Processes

| | |
|---|---|
| Processes | $\mathrm{P}, \mathrm{Q} \ ::= \ 0 \mid c \cdot A \mid \mathrm{P} \parallel \mathrm{Q}$ |
| Component | species $c \cdot A$ at concentration $c \in \mathbb{R}_{\geqslant 0}$. |
| Mixture | of processes $\mathrm{P} \parallel \mathrm{Q}$. |

We can identify processes, up to structural congruence, with elements of *process space* $\mathcal{P} = \mathbb{R}^{\mathcal{S}^{\#}}$.

Species embed in process space $\langle - \rangle : \mathcal{S} \to \mathcal{P}$ at unit concentration.

# Operational Semantics for Processes

The behaviour of a process over time is a trajectory through process space.

Immediate behaviour $\frac{dP}{dt} \in \mathbb{R}^{\mathcal{S}^{\#}}$ vector in process space

Interaction potential $\partial P \in \mathbb{R}^{\mathcal{S} \times \mathcal{N} \times \mathcal{C}} = \mathcal{D}$ interaction space

Space $\mathcal{D}$ has basis $\langle A \xrightarrow{a} F \rangle$ for species $A$, name $a$, concretion $F$.

Interaction tensor $\oplus : \mathcal{D} \times \mathcal{D} \to \mathcal{P}$

Bilinear function generated by

$$\langle A \xrightarrow{a} F \rangle \oplus \langle B \xrightarrow{b} G \rangle = Aff(a, b)(\langle F \circ G \rangle - \langle A \rangle - \langle B \rangle)$$

# Process Semantics

$\frac{dP}{dt}$: Immediate behaviour

- Vector field $\frac{d}{dt}$ over process space $\mathcal{P}$
- Equivalent to an ODE system

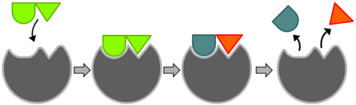
$\partial P$: Interaction potential

- Element of $\mathbb{R}^{\mathcal{S} \times \mathcal{N} \times \mathcal{C}}$
- Equivalent to transition system

$$\partial(P \parallel Q) = \partial P + \partial Q$$

$$\frac{d(P \parallel Q)}{dt} = \frac{dP}{dt} + \frac{dQ}{dt} + \partial P \obar \partial Q$$

# Example Process: Enzyme Catalysis



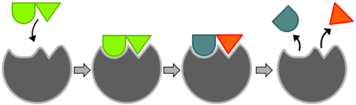
$$S = s(x, y).(x.S + y.(P|P'))$$

$$E = \nu(u, r, t{:}M).(e\langle u, r\rangle.t.E)$$

$$P = P' = \tau@k_{degrade}.0$$

$$c_S \cdot S \parallel c_E \cdot E$$

# Example Process: Enzyme Catalysis



$$S = s(x, y).(x.S + y.(P|P'))$$

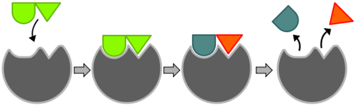
$$E = \nu(u, r, t{:}M).(e\langle u, r\rangle.t.E)$$

$$P = P' = \tau@k_{\mathit{degrade}}.0$$

$$c_S \cdot S \parallel c_E \cdot E$$

```
enzyme.cpi
...
species E() = {
  site t, u, r;
...
```

# Example Process: Enzyme Catalysis

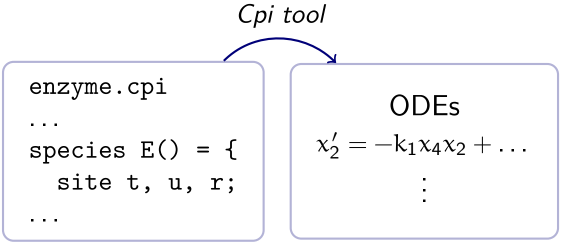


$$S = s(x, y).(x.S + y.(P|P'))$$

$$E = \nu(u, r, t{:}M).(e\langle u, r\rangle.t.E)$$

$$P = P' = \tau@k_{degrade}.0$$

$$c_S \cdot S \parallel c_E \cdot E$$



*Cpi tool*

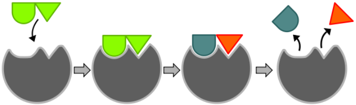
```
enzyme.cpi
...
species E() = {
  site t, u, r;
...
```

ODEs

$$x_2' = -k_1 x_4 x_2 + \ldots$$

$$\vdots$$

# Example Process: Enzyme Catalysis



$$S = s(x, y).(x.S + y.(P|P'))$$

$$E = \nu(u, r, t{:}M).(e\langle u, r\rangle.t.E)$$

$$P = P' = \tau @ k_{degrade}.0$$

$$c_S \cdot S \parallel c_E \cdot E$$

*Cpi tool*

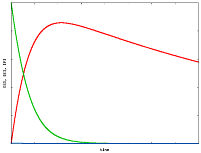
*Octave*

```
enzyme.cpi
...
species E() = {
  site t, u, r;
...
```

ODEs

$$x_2' = -k_1 x_4 x_2 + \ldots$$

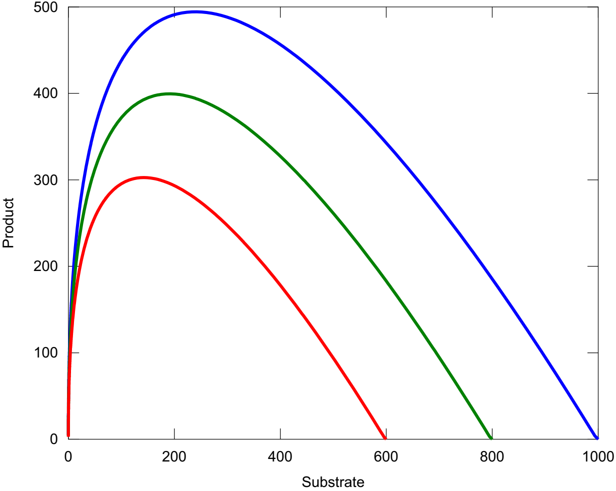
$$\vdots$$



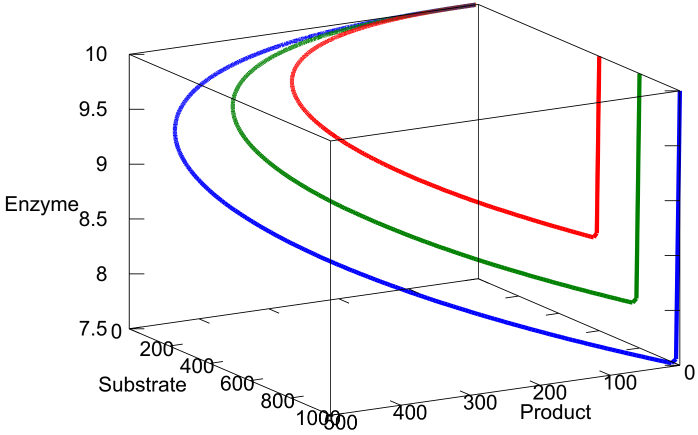
# Cpi tool input syntax

```
const kbind=1e−3;     const kreact=2.0;
const kunbind=1.0;    const kdegrade=3e−4;

site e,s;  react (e,s)@kbind;

species S() = { body s(;x,y).(x(;).S() + y(;).P());   init 1000.0; }

species E() = { site u,r,t;
                react (u,t)@kunbind;
                react (r,t)@kreact;
                body e(u,r;).act(;).E();
                init 10.0; }

species P() = { body tau<kdegrade>.0; init 0.0; }
```

# Process Space: Substrate & Product



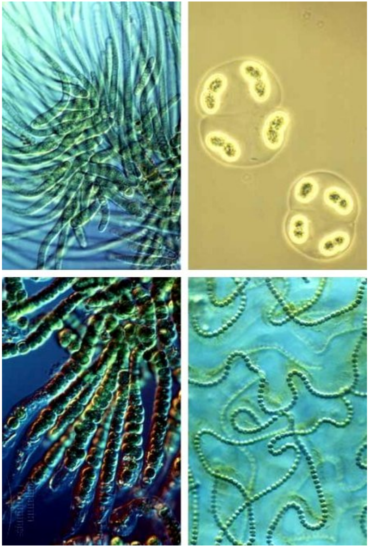
# Process Space: Substrate & Product & Enzyme



# Example: Synechococcus Elongatus

*Synechococcus* is a genus of cyanobacteria (blue-green algae): single-celled photosynthesising plankton that provide a foundation for the aquatic food chain.

*S. Elongatus* is a species of Synechococcus that is particularly abundant: some estimates suggest that it contributes 25% of marine nutrient primary production.



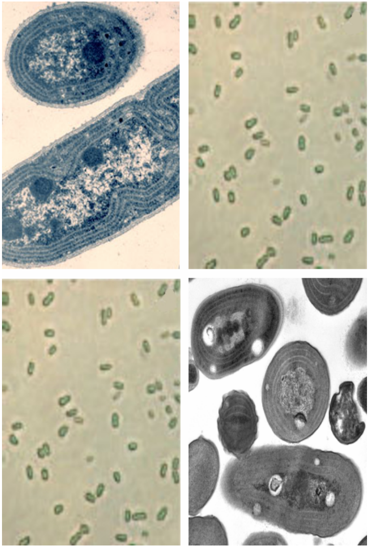
# Circadian Clock in S. Elongatus

S. Elongatus has an internal clock, that turns genes on and off through day and night.

The cycling mechanism does not require gene transcription, and will operate in a test tube (*in vitro*).

Although it is entrained by light, it will also run for weeks without external stimulus.

Tomita, Nakajima, Kondo, Iwasaki.
No transcription-translation feedback in circadian rhythm of KaiC phosphorylation.
Science **307**(5707) (2005) 251–254



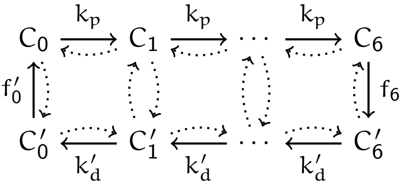
# Proposed Mechanism

The S. Elongatus clock requires three proteins: KaiA, KaiB and KaiC (for *kaiten*). One proposed mechanism is the following:

- KaiC forms hexamers, with six phosphorylation sites.
- KaiC also has two conformations; it preferentially phosphorylates in one and dephosphorylates in the other,
- KaiA catalyses phosphorylation of the first (active) conformation.
- KaiB dimers stabilise the second (inactive) conformation.
- A KaiB dimer bound to KaiC will bind a further two KaiA, removing them from other possible interactions.
- Cyclic phosphorylation of individual KaiC gives the basic mechanism; interaction with varying KaiA and KaiB coordinates this across the cell.

van Zon, Lubensky, Altena, ten Wolde.
An allosteric model of circadian KaiC phosphorylation.
PNAS **104**(18) (2007) 7420–7425

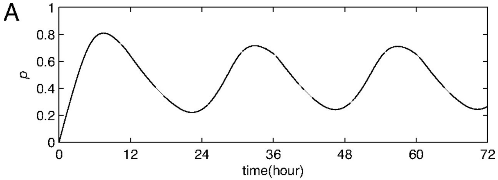
# ODE Model



Active forms

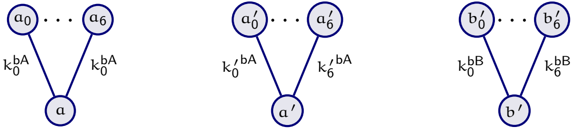
Inactive forms

van Zon et al. give an ODE model of this mechanism, and show that it cycles. They conjecture that *differential affinities* are a key feature.

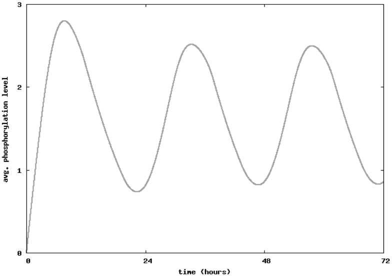


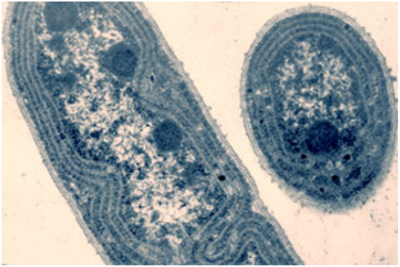
# Continuous $\pi$ Model

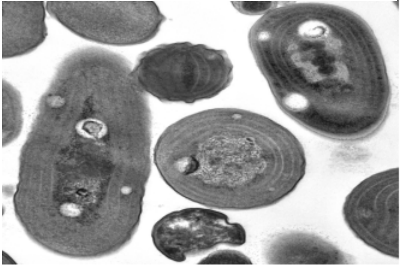
$$
\begin{aligned}
C_i &= \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))\\
C'_i &= (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)\\
BC'_i &= (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \,|\, B \,|\, B)) + (a'_i.a'_i.ABC'_i)\\
ABC'_i &= (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \,|\, A \,|\, A)) + (\tau@k'_d.ABC'_{i-1})\\
A &= a(x).x.A + a'.0\\
B &= b'.0\\
\\
P &= 0.58 \cdot A \;\|\; 0.58 \cdot B \;\|\; 1.72 \cdot C_0
\end{aligned}
$$

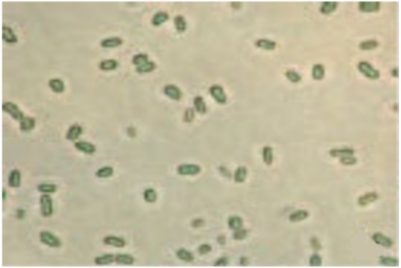


# Running $\pi$



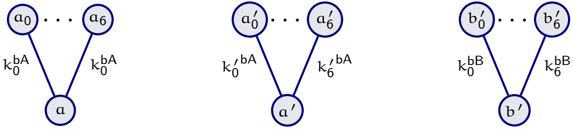






# Modification: Remove autonomous phosphorylation

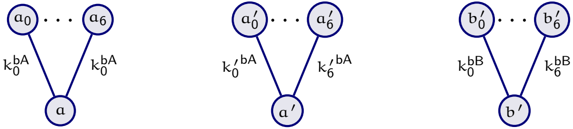
$$
\begin{aligned}
C_i &= \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))\\
C'_i &= (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)\\
BC'_i &= (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \,|\, B \,|\, B)) + (a'_i.a'_i.ABC'_i)\\
ABC'_i &= (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \,|\, A \,|\, A)) + (\tau@k'_d.ABC'_{i-1})\\
A &= a(x).x.A + a'.0\\
B &= b'.0\\
\\
P &= 0.58 \cdot A \parallel 0.58 \cdot B \parallel 1.72 \cdot C_0
\end{aligned}
$$



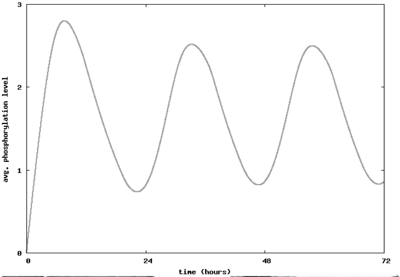
# Modification: Remove autonomous phosphorylation

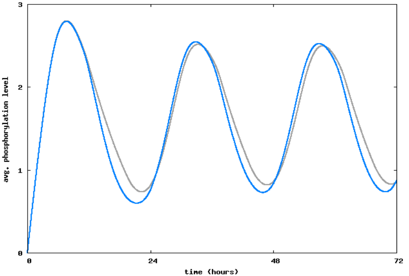
$$
\begin{aligned}
C_i &= \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))\\
C'_i &= (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)\\
BC'_i &= (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \,|\, B \,|\, B)) + (a'_i.a'_i.ABC'_i)\\
ABC'_i &= (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \,|\, A \,|\, A)) + (\tau@k'_d.ABC'_{i-1})\\
A &= a(x).x.A + a'.0\\
B &= b'.0
\end{aligned}
$$

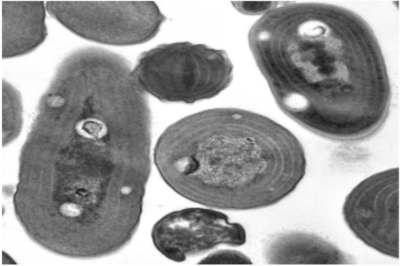
$$P = 0.58 \cdot A \parallel 0.58 \cdot B \parallel 1.72 \cdot C_0$$

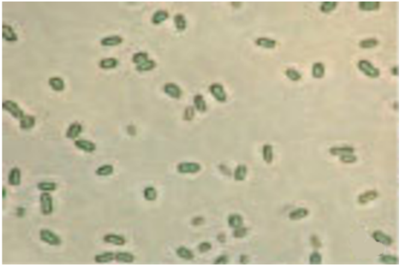


# Modification: Remove autonomous phosphorylation









# Modification: Weaken KaiA binding

$$C_i = \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))$$

$$C'_i = (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)$$

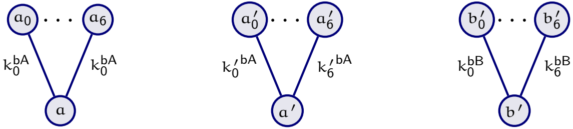
$$BC'_i = (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \mid B \mid B)) + (a'_i.a'_i.ABC'_i)$$

$$ABC'_i = (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \mid A \mid A)) + (\tau@k'_d.ABC'_{i-1})$$

$$A = a(x).x.A + a'.0$$

$$B = b'.0$$

$$P = 0.58 \cdot A \parallel 0.58 \cdot B \parallel 1.72 \cdot C_0$$



# Modification: Weaken KaiA binding

$$C_i = \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))$$

$$C'_i = (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)$$

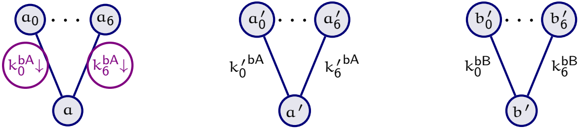
$$BC'_i = (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \,|\, B \,|\, B)) + (a'_i.a'_i.ABC'_i)$$

$$ABC'_i = (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \,|\, A \,|\, A)) + (\tau@k'_d.ABC'_{i-1})$$

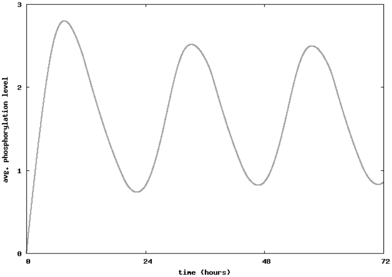
$$A = a(x).x.A + a'.0$$

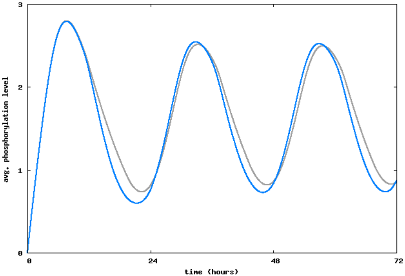
$$B = b'.0$$

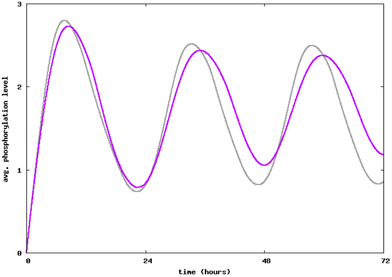
$$P = 0.58 \cdot A \;\|\; 0.58 \cdot B \;\|\; 1.72 \cdot C_0$$

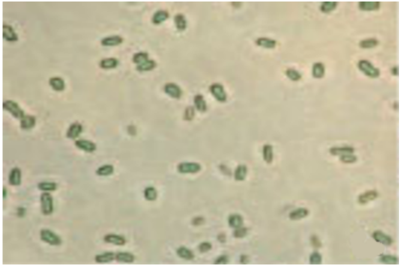


# Modification: Weaken KaiA binding









# Modification: KaiA-KaiB dimers

$$C_i = \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))$$

$$C'_i = (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)$$

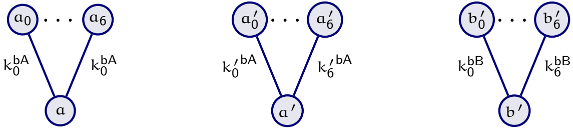
$$BC'_i = (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \mid B \mid B)) + (a'_i.a'_i.ABC'_i)$$

$$ABC'_i = (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \mid A \mid A)) + (\tau@k'_d.ABC'_{i-1})$$

$$A = a(x).x.A + a'.0$$

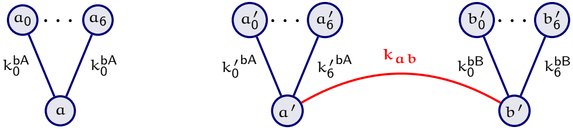
$$B = b'.0$$

$$P = 0.58 \cdot A \parallel 0.58 \cdot B \parallel 1.72 \cdot C_0$$

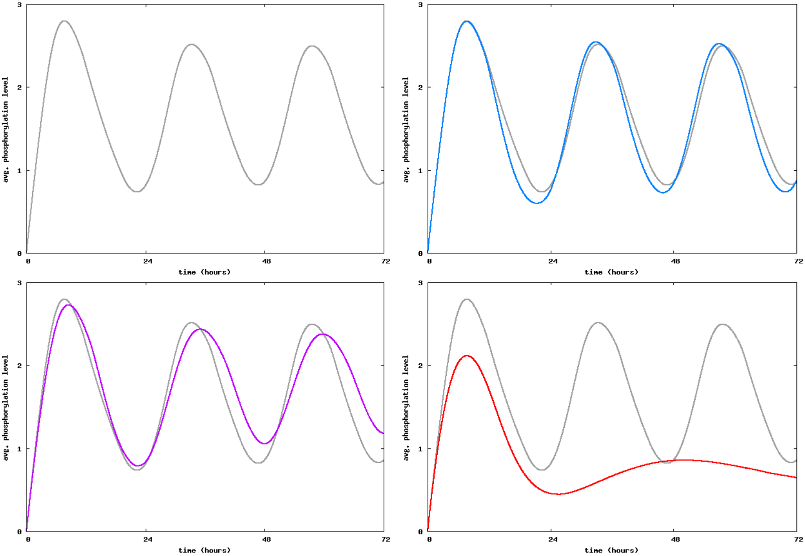


# Modification: KaiA-KaiB dimers

$$
\begin{aligned}
C_i &= \nu(u, r, t{:}M_i).((\tau@k_p.C_{i+1}) + (\tau@f_i.C'_i) + (\tau@k_d.C_{i-1}) + (a_i\langle t\rangle.(u.C_i + r.C_{i+1})))\\
C'_i &= (\tau@k'_p.C'_{i+1}) + (\tau@f'_i.C_i) + (\tau@k'_d.C'_{i-1}) + (b'_i.b'_i.BC'_i)\\
BC'_i &= (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \,|\, B \,|\, B)) + (a'_i.a'_i.ABC'_i)\\
ABC'_i &= (\tau@k'_p.ABC'_{i+1}) + (\tau@k_i^{uA}.(BC'_i \,|\, A \,|\, A)) + (\tau@k'_d.ABC'_{i-1})\\
A &= a(x).x.A + a'.0\\
B &= b'.0\\
\\
P &= 0.58 \cdot A \parallel 0.58 \cdot B \parallel 1.72 \cdot C_0
\end{aligned}
$$



# Modification: KaiA-KaiB dimers



# Variation

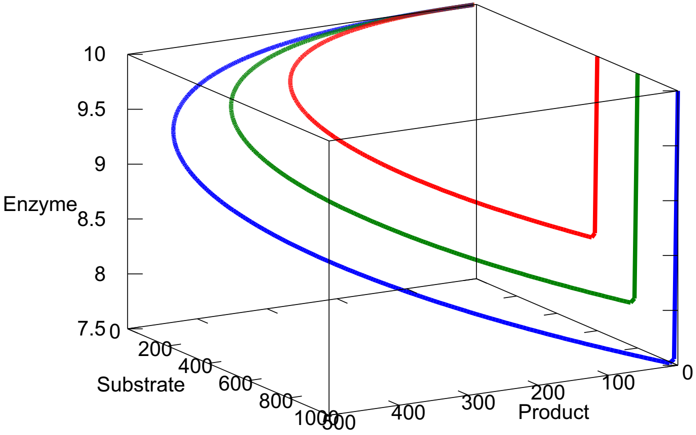
Given a system as a sequence of $c\pi$ species definitions, we can consider perturbing it in various ways.

- Duplication of species definitions
- Duplication of names
- Modification of affinities
- Modification of explicit rates
- Introduction of local names (complexation)

Larger changes can be built up by composing an appropriate basis of variation operators.

While clearly $c\pi$ syntax does not correspond directly to genetics, it is possible to identify syntactically simple variations that are also biologically plausible.

# Process Space: Substrate & Product & Enzyme



# Further Work: Analysis

To analyse the behaviour of a $c\pi$ system, we can graph its trajectory.

However, one strength of having an intermediate language is the possibility of multiple alternative mathematical models and routes to analysis:

- Numerical solution of ODE simulation; inspection of trajectory.
- Linear (metric) temporal logic over traces from ODE simulation [e.g. Fages & Rizk 2007]
- Temporal logic on $c\pi$ processes. Immediate behaviour: $P \vdash G_{\leqslant t}(\phi)$
- Contextual logic for $c\pi$ processes. Interaction potential: $Q \vdash F^{c:a}_{\leqslant t} G(\psi)$

With both variation (genotype) and model-checking (phenotype) we would have the means to explore evolvability, robustness, and neutrality.

# Review

- Continuous $\pi$-calculus
  - Modular description of biomolecular systems
  - Compositional semantics in real vector spaces
  - Flexible interaction structure
- S. Elongatus circadian clock
  - Protein-protein interaction *in vitro*
  - Candidate mechanism oscillates
  - Behaviour under system variation

# Future Work

- Behavioural analysis
  - Temporal logic on numerical simulation traces
  - Model-checking on $c\pi$ terms $P \vdash G_{\leqslant t}(\phi)$
  - Compositional logic of interaction potential $Q \vdash F^{c \cdot a}_{\leqslant t} G(\psi)$
- System Evolution
  - Evolutionary trajectories
  - Variation, evolvability
  - Robustness and neutrality
- Alternative Semantics
  - Markov chains
  - Stochastic simulation
  - Hybrid models, protein/DNA interaction

# References

Marek Kwiatkowski and Ian Stark.
The Continuous $\pi$-Calculus: A Process Algebra for Biochemical Modelling. In *Computational Methods in Systems Biology: Proc. CMSB 2008*
Lecture Notes in Computer Science 5307, pages 103–122. Springer 2008

Tomita, Nakajima, Kondo, Iwasaki.
No transcription-translation feedback in circadian rhythm of KaiC phosphorylation.
Science **307**(5707) (2005) 251–254

van Zon, Lubensky, Altena, ten Wolde.
An allosteric model of circadian KaiC phosphorylation.
PNAS **104**(18) (2007) 7420–7425