Exploring Variation in Biochemical Pathways with the Continuous π-Calculus

Ian Stark and Marek Kwiatkowski

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

Evolutionary Ecology
Institute of Integrative Biology
Department of Environmental Sciences
EAWAG & ETH Zürich

Thursday 23 February 2012

Summary

The *continuous π-calculus* (cπ) is a process algebra for modelling behaviour and variation in biomolecular systems: *e.g.* enzyme activation and inhibition; circadian clocks; signalling pathways.

With a language of potential changes in cπ processes we systematically explore the evolutionary neighbourhoods of a specific signalling pathway, and observe instances of robustness, neutrality and evolvability.

High-level languages for biological descriptions can smooth the route from mechanism descriptions to mathematically precise models; and also help to express and test high-level hypotheses.

Marek Kwiatkowski and Ian Stark.
On Executable Models of Molecular Evolution. In *Proc. 8th International Workshop on Computational Systems Biology WCSB 2011*, pp. 105–108.

Systems Biology

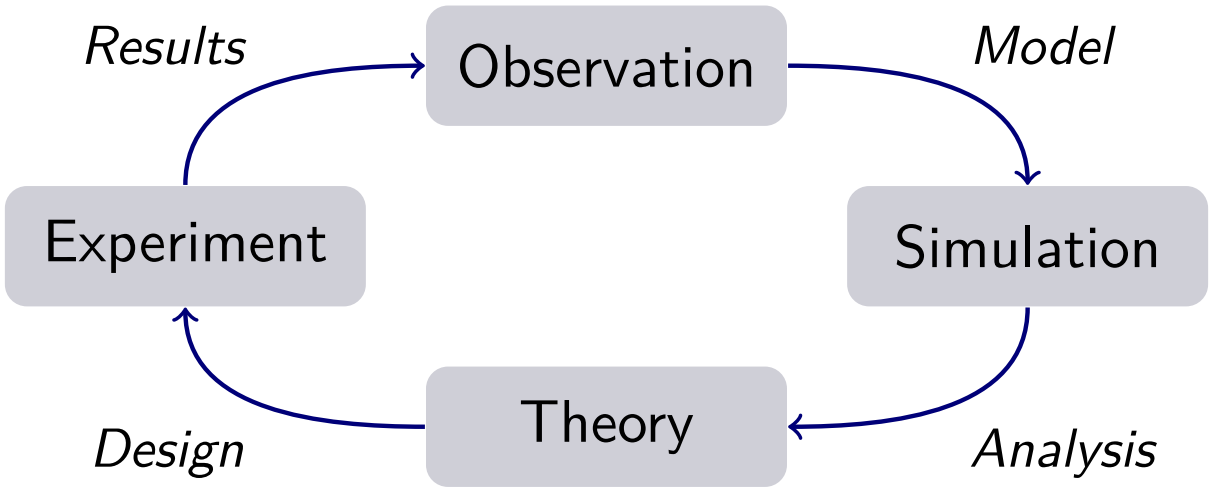
Biology is the study of living organisms; Systems Biology is the study of the dynamic processes that take place within those organisms.

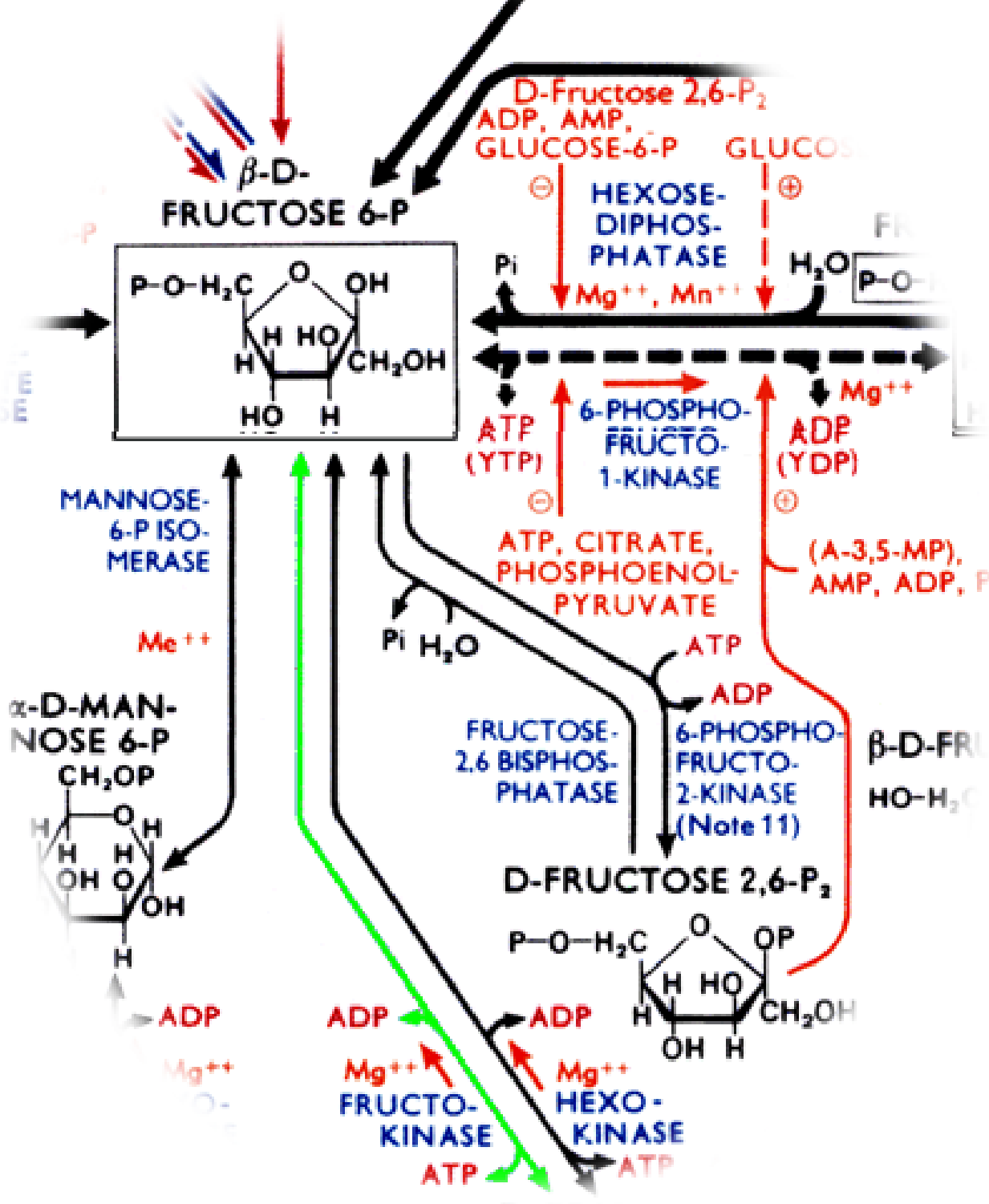
In particular:

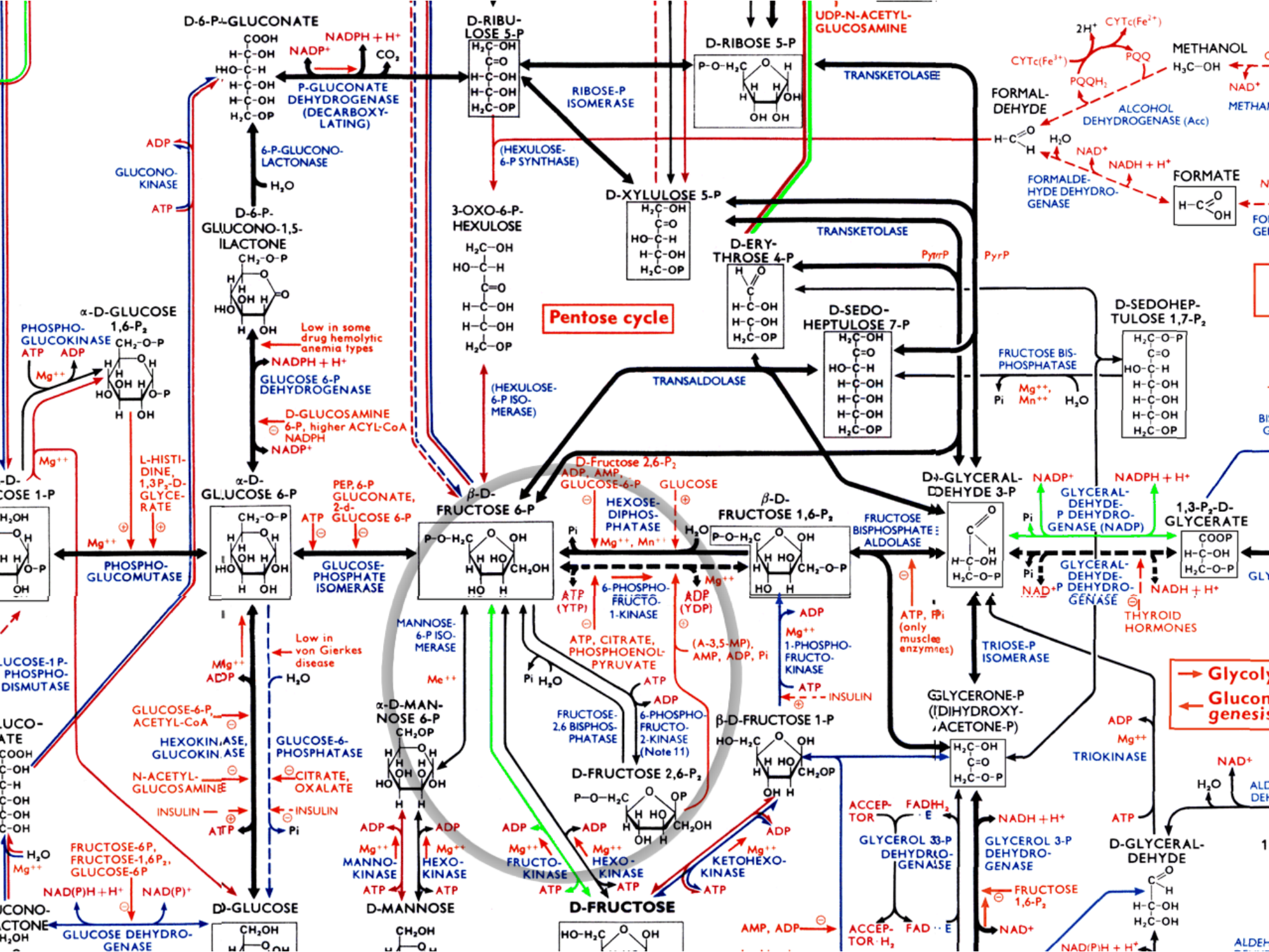
- Interaction between processes;
- Behaviour emerging from such interaction; and
- Integration of component behaviours.

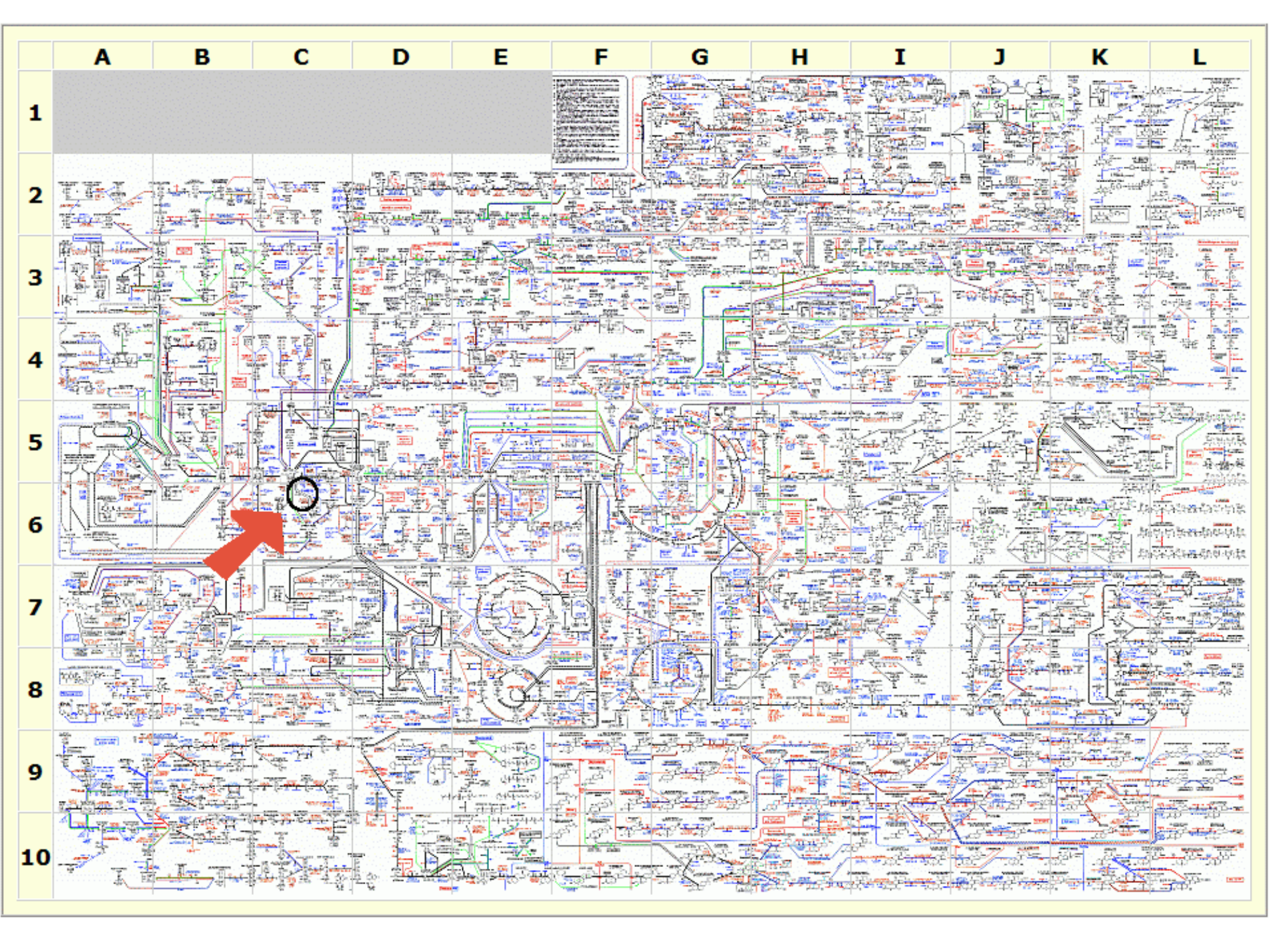
Systems Biology

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
D-Fructose 2,6-P2
ADP, AMP,
GLUCOSE-6-P
GLUCOSE
HEXOSE-
DIPHOS-
PHATASE
Mg++, Mn++
Pi
H2O
ATP
(YTP)
6-PHOSPHO-
FRUCTO-
1-KINASE
ADP
(YDP)
Mg++
ATP, CITRATE,
PHOSPHOENOL-
PYRUVATE
(A-3,5-MP),
AMP, ADP,
MANNOSE-
6-P ISO-
MERASE
Me++
α-D-MAN-
NOSE 6-P
Pi H2O
ATP
ADP
FRUCTOSE-
2,6 BISPHOS-
PHATASE
6-PHOSPHO-
FRUCTO-
2-KINASE
(Note 11)
β-D-FR
D-FRUCTOSE 2,6-P2
ADP
ADP
ADP
Mg++
Mg++
Mg++
FRUCTO-
KINASE
HEXO-
KINASE
ATP
ATP


Pentose cycle
D-6-P-GLUCONATE
NADP+
NADPH + H+
CO2
P-GLUCONATE DEHYDROGENASE (DECARBOXY-LATING)
D-RIBULOSE 5-P
D-RIBOSE 5-P
RIBOSE-P ISOMERASE
TRANSKETOLASE
UDP-N-ACETYL-GLUCOSAMINE
METHANOL
FORMALDEHYDE
ALCOHOL DEHYDROGENASE (Acc)
FORMALDEHYDE DEHYDROGENASE
FORMATE
6-P-GLUCONO-LACTONASE
GLUCONO-KINASE
ADP
ATP
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
D-SEDOHEPTULOSE 7-P
D-SEDOHEPTULOSE 1,7-P2
FRUCTOSE BISPHOSPHATASE
Mg++, Mn++
Pi
TRANSALDOLASE
(HEXULOSE-6-P ISOMERASE)
α-D-GLUCOSE 1,6-P2
PHOSPHOGLUCOKINASE
Low in some drug hemolytic anemia types
NADPH + H+
GLUCOSE 6-P DEHYDROGENASE
D-GLUCOSAMINE 6-P, higher ACYL-CoA
NADP+
L-HISTIDINE, 1,3P2-D-GLYCERATE
α-D-GLUCOSE 6-P
β-D-FRUCTOSE 6-P
PEP, 6-P GLUCONATE, 2-d-GLUCOSE 6-P
D-Fructose 2,6-P2
ADP, AMP, GLUCOSE-6-P
GLUCOSE
HEXOSE-DIPHOSPHATASE
β-D-FRUCTOSE 1,6-P2
FRUCTOSE BISPHOSPHATE ALDOLASE
D-GLYCERALDEHYDE 3-P
GLYCERALDEHYDE-P DEHYDROGENASE (NADP)
1,3-P2-D-GLYCERATE
PHOSPHOGLUCOMUTASE
GLUCOSE-PHOSPHATE ISOMERASE
6-PHOSPHOFRUCTO-1-KINASE
ATP (YTP)
ADP (YDP)
ATP, CITRATE, PHOSPHOENOLPYRUVATE
(A-3,5-MP), AMP, ADP, Pi
1-PHOSPHOFRUCTOKINASE
INSULIN
GLYCERALDEHYDE-P DEHYDROGENASE
THYROID HORMONES
ATP, Pi (only muscle enzymes)
TRIOSE-P ISOMERASE
GLYCERONE-P (DIHYDROXYACETONE-P)
Low in von Gierkes disease
MANNOSE-6-P ISOMERASE
α-D-MANNOSE 6-P
FRUCTOSE-2,6 BISPHOSPHATASE
6-PHOSPHOFRUCTO-2-KINASE (Note 11)
D-FRUCTOSE 2,6-P2
β-D-FRUCTOSE 1-P
TRIOKINASE
GLUCOSE-6-P, ACETYL-CoA
HEXOKINASE, GLUCOKINASE
GLUCOSE-6-PHOSPHATASE
N-ACETYL-GLUCOSAMINE
CITRATE, OXALATE
INSULIN
FRUCTOSE-6P, FRUCTOSE-1,6P2, GLUCOSE-6P
NAD(P)H + H+
NAD(P)+
D-GLUCOSE
D-MANNOSE
D-FRUCTOSE
MANNOKINASE
HEXOKINASE
FRUCTOKINASE
KETOHEXOKINASE
ACCEPTOR
FADH2
GLYCEROL 3-P DEHYDROGENASE
FRUCTOSE 1,6-P2
AMP, ADP
D-GLYCERALDEHYDE
GLUCOSE DEHYDROGENASE
Glycoly
Glucon genesis


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?

Machinery

Large Data: Semistructured data; data integration; data mining; learning

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

Ideas

Language: Abstraction; modularity; semantics; formal models

Reasoning: Logics; behavioural description; model checking

Biochemical Simulation

Systems 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
- π-calculus; stochastic π; BioSPI; SPiM
- Beta binders; BlenX
- Ambients, bioAmbients
- P-models
- Brane calculi; Bitonal systems
- PEPA, bioPEPA
- Kappa
- PRISM
- Pathway Logic
- ...

Development and Evolution

Development is the process by which genetic information (genotype) is translated to a functional biological object (phenotype).

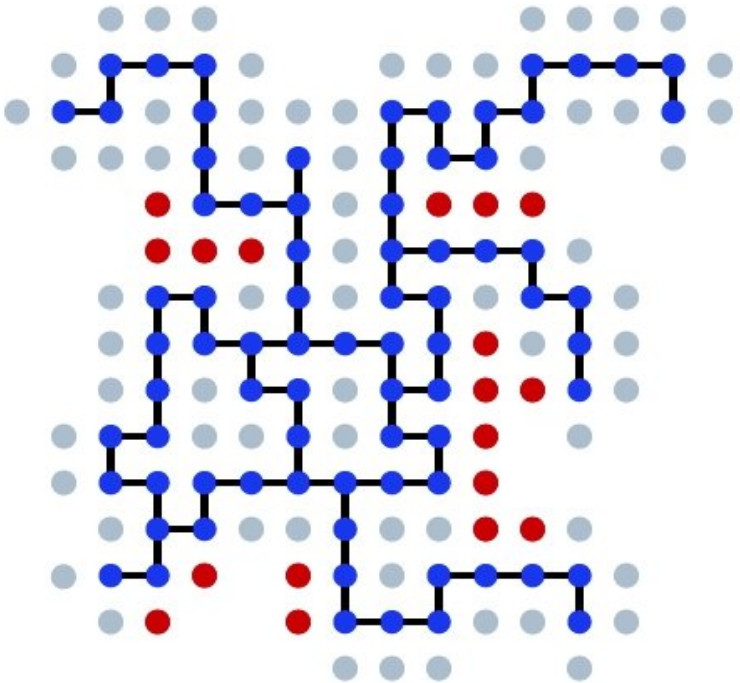
In most settings of interest, development is notoriously complex. For example, an embryo becoming an organism or a peptide chain folding into a protein.

Evolutionary developmental biology (evo-devo) is concerned with evolution-related properties of development, such as *evolvability*, *robustness*, *canalisation* and *plasticity*.

Mathematical abstractions and simple instances of development help to illuminate generic features of this process.

Neutral Spaces and Neighbours

The neutral space of a phenotype is the collection of all genotypes giving rise to that phenotype.



- ✓ robustness
- ✓ evolvability
- ✓ neutral evolution
- ? recombination
- ? horizontal gene transfer
- ✗ phenotype plasticity
- ✗ variable development

A. Wagner Robustness and Evolvability in Living Systems Princeton University Press, 2005

The Continuous π-Calculus

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

Based on Milner's π-calculus, it introduces continuous variability in:

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

while retaining classic process-algebra features of:

Formality:	Unambiguous description
Parsimony:	Few primitives
Compositionality:	Behaviour of the whole arises from that of its parts
Abstraction:	System description distinct from system dynamics
Intermediation:	Many analyses techniques for a single description

Basics of $c\pi$

Continuous π has two levels of system description:

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

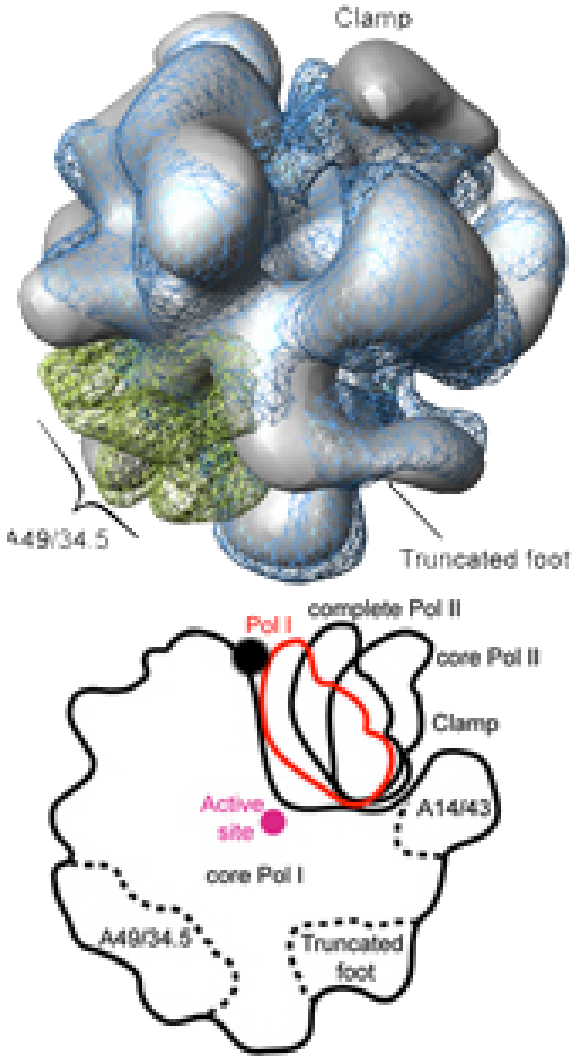
Process space arises as a real-valued vector space over species, with each point the state of a system and behaviours as trajectories through that.

Names in $c\pi$

As in standard π-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.

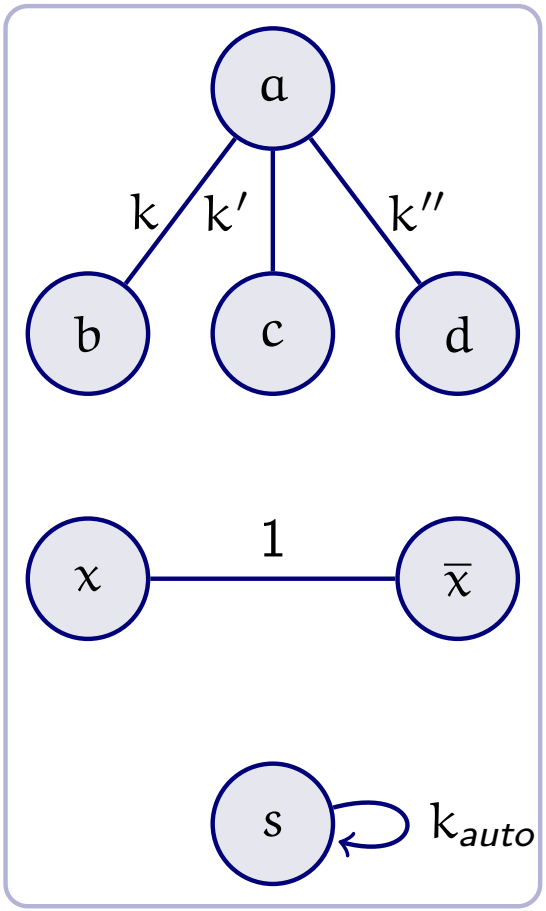


Names in $c\pi$

As in standard π-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.

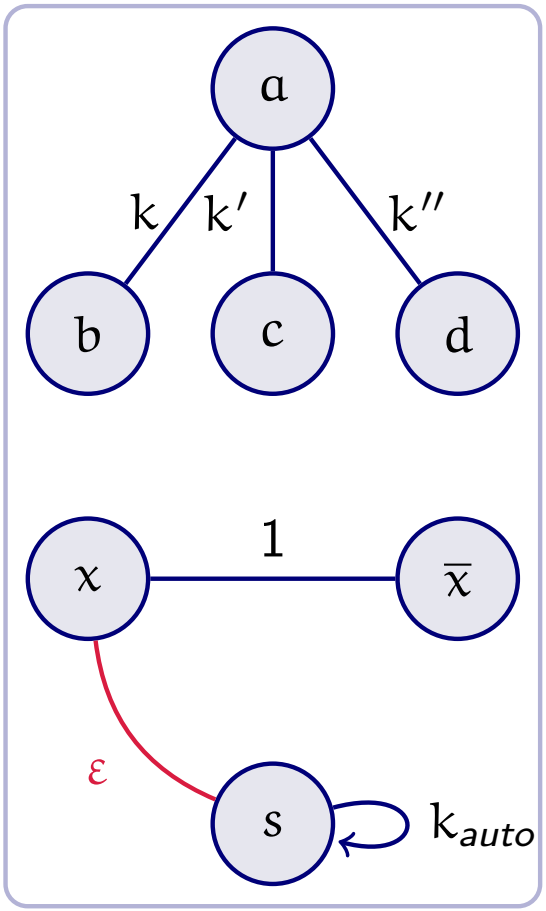


Names in $c\pi$

As in standard π-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.



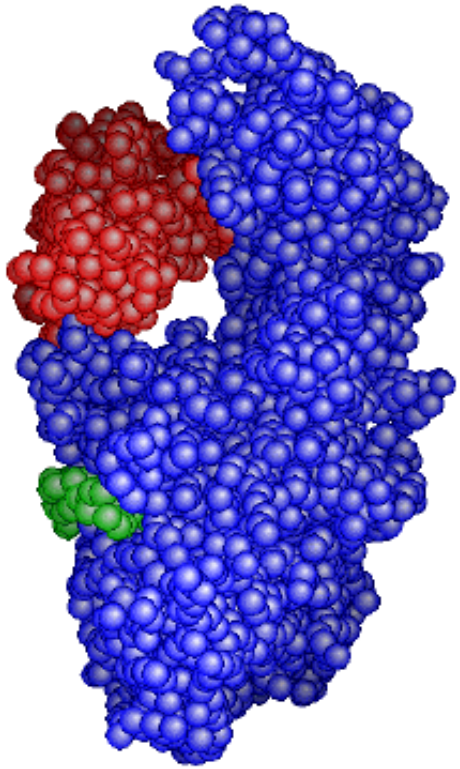
Restriction in $c\pi$

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

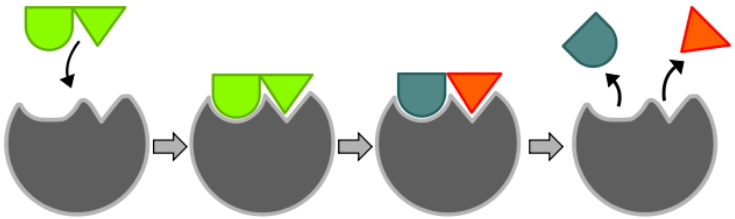
The binder can be a single local name $(\nu x.-)$, or several names with their own affinity network $(\nu M.-)$.

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

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



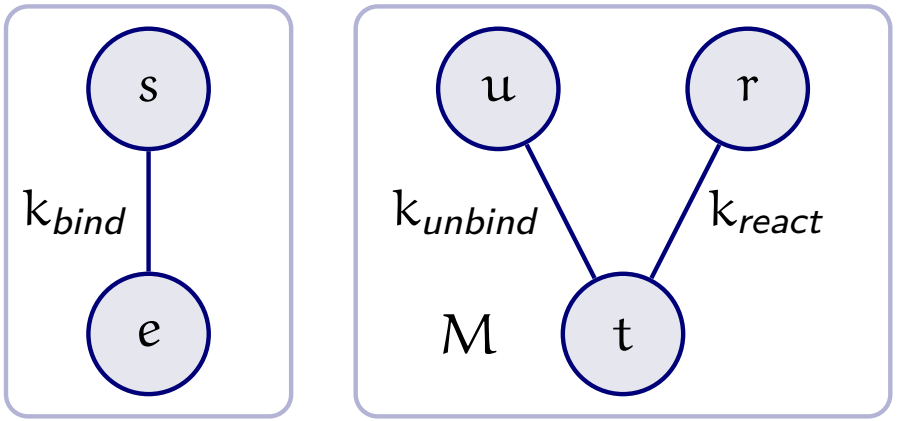
Example: Enzyme Catalysis

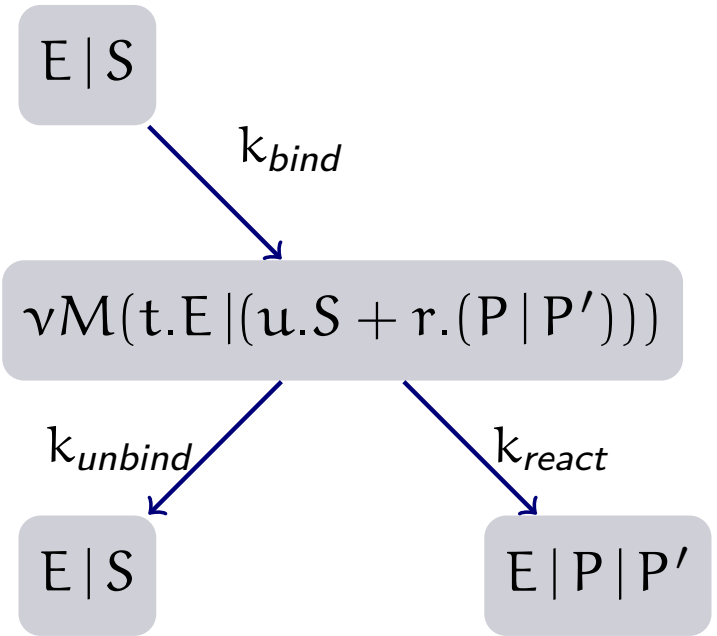


$$S = s(x, y).(x.S + y.(P|P'))$$

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

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





Formalities: Species and Processes

Species $A, B ::= \Sigma\alpha.A \mid A \mid B \mid \nu M.A \mid \ldots$

Processes $P, Q ::= 0 \mid c \cdot A \mid P \parallel Q$

Component $c \cdot A$ of species A at concentration $c \in \mathbb{R}_{\geqslant 0}$.

Mixture of processes $P \parallel Q$.

Species transitions $A \xrightarrow{\alpha} B$ are given by structural operational rules.

We can identify processes with elements of *process space* $\mathcal{P} = \mathbb{R}^{\mathcal{S}}$, where $\mathcal{S}$ is the set of species (up to structural congruence)

Process Semantics

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

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

∂P: Interaction potential

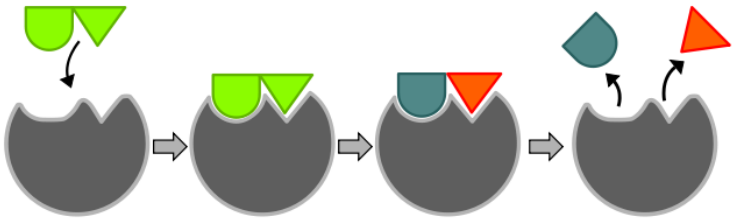
- Captures available reactivity
- Element of $\mathbb{R}^{\mathcal{N}\times\mathcal{S}\times\mathcal{C}}$

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

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

Both $\frac{dP}{dt}$ and ∂P are defined by induction on the structure of processes; and beneath that, from the transitions of component species $c \cdot A$.

Example: Enzyme Catalysis



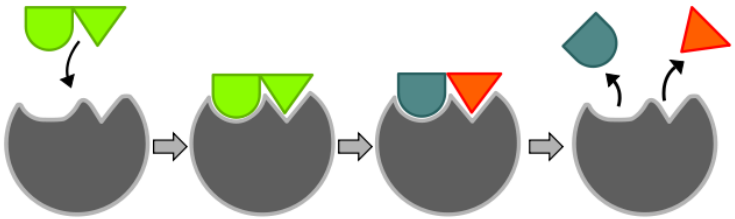
$$S = s(x, y).(x.S + y.(P|P'))$$

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

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

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

Example: Enzyme Catalysis



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

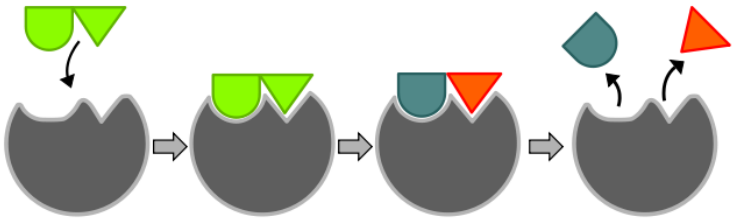
$$E = \nu(u, r, t : M).(e\langle u, r\rangle.t.E)$$

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

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

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

Example: 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

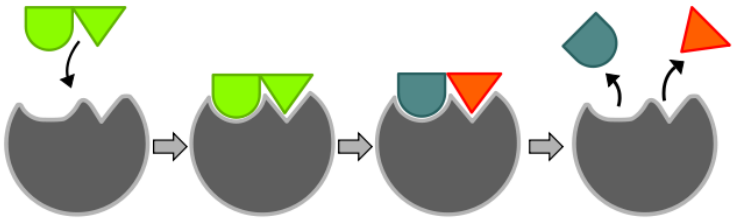
```
enzyme.cpi
...
species E() = {
  site u, r, t;
...
```

ODEs

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

$$\vdots$$

Example: Enzyme Catalysis



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

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

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

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

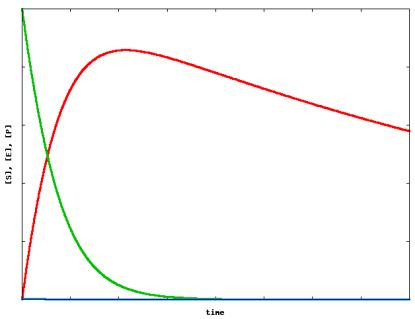
Cpi tool *Octave*

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

ODEs

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

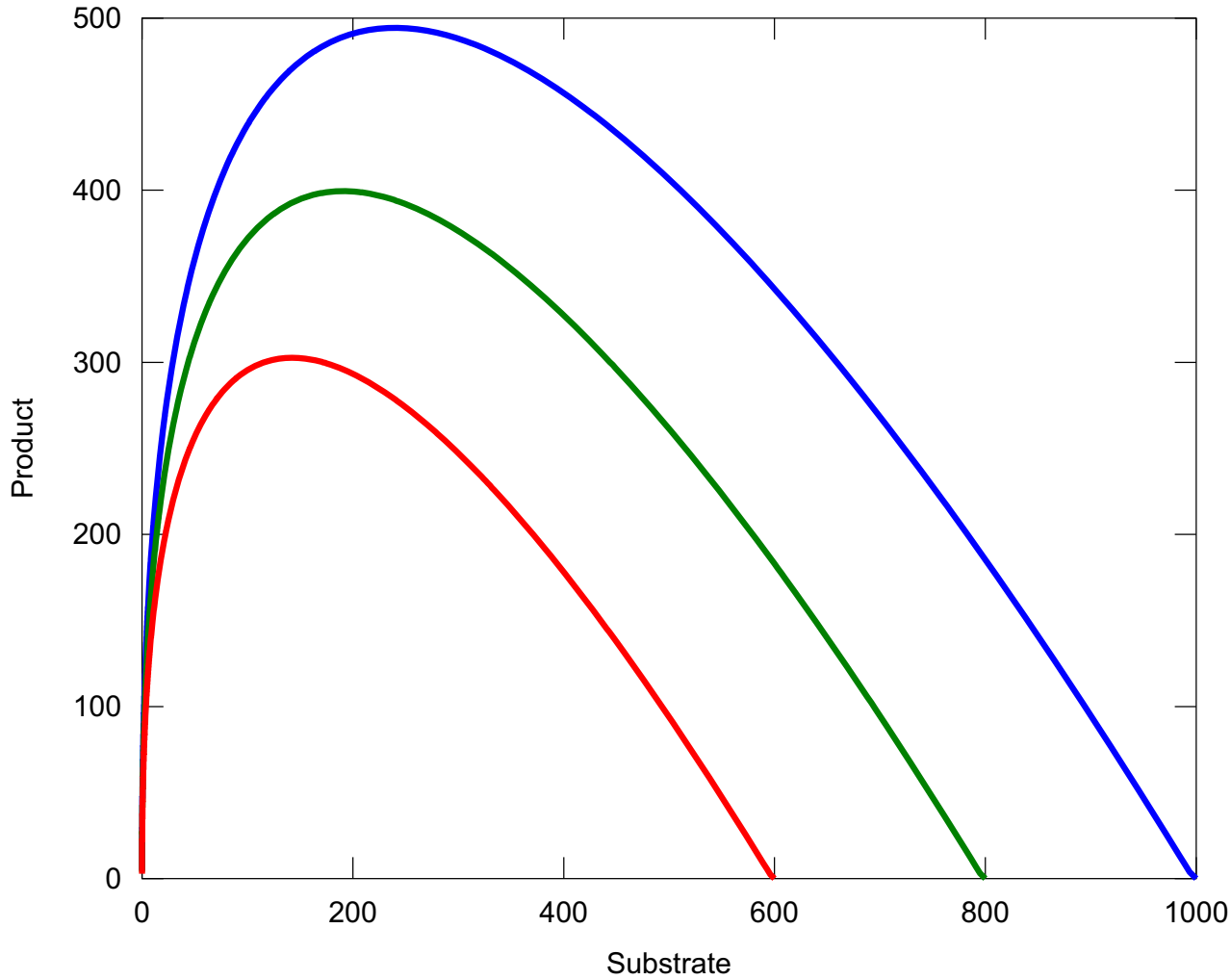
$$\vdots$$



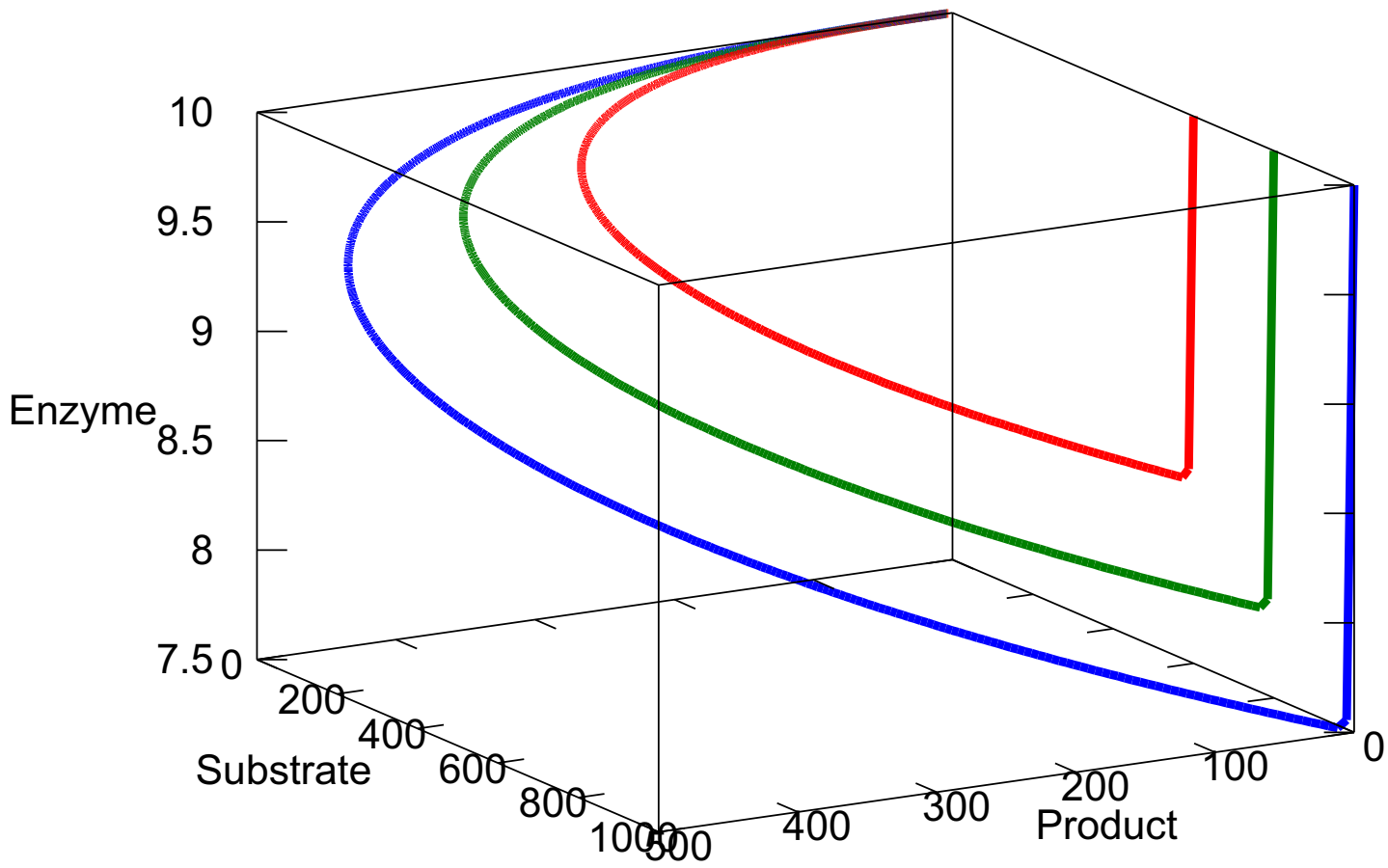
Tool Syntax

```
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



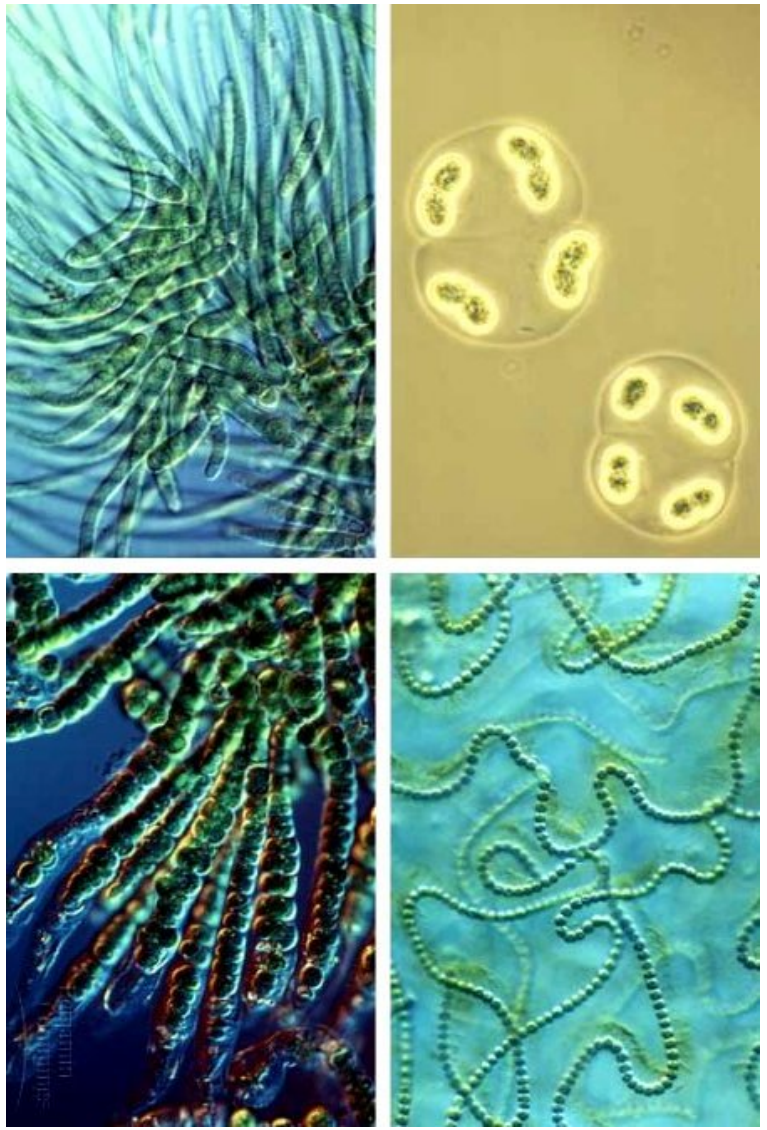
Process Space: Substrate & Product & Enzyme



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.

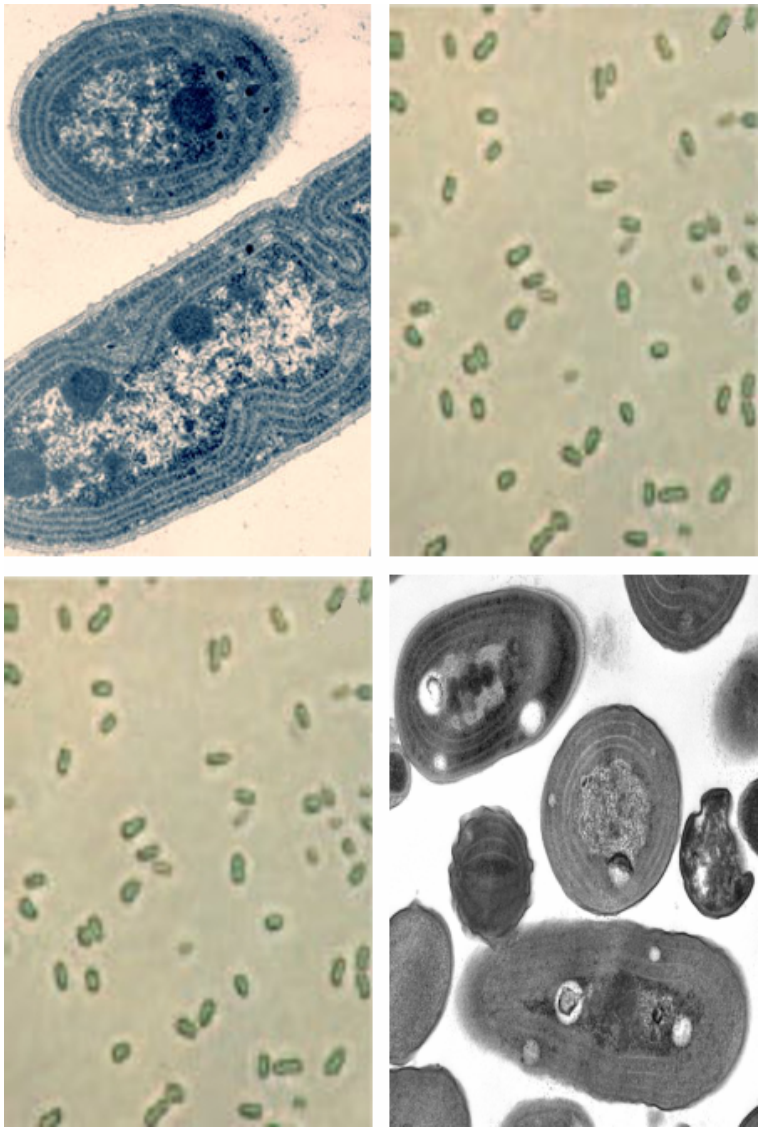


Circadian Clock in S. Elongatus

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.



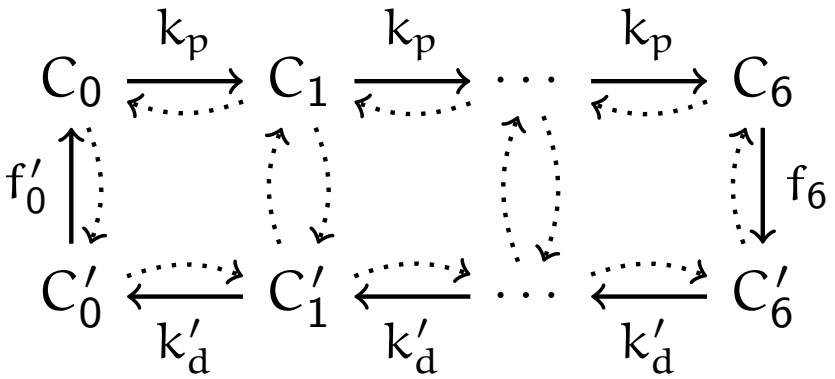
Tomita, Nakajima, Kondo, Iwasaki.
No transcription-translation feedback in circadian rhythm of KaiC phosphorylation.
Science **307**(5707):251–254, 2005.

Proposed Mechanism

The S. Elongatus clock requires three proteins: KaiA, KaiB and KaiC.

Their structure is known, and several different mechanisms have been proposed for how they interact to coordinate circadian rhythms.

For example, van Zon et al. suggest cyclic six-fold phosphorylation of KaiC hexamers in two alternative conformations, moderated by KaiA and KaiB.

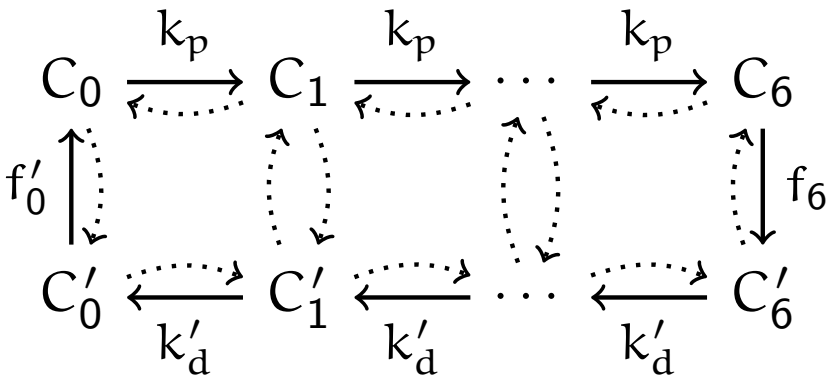


Active forms

Inactive forms

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

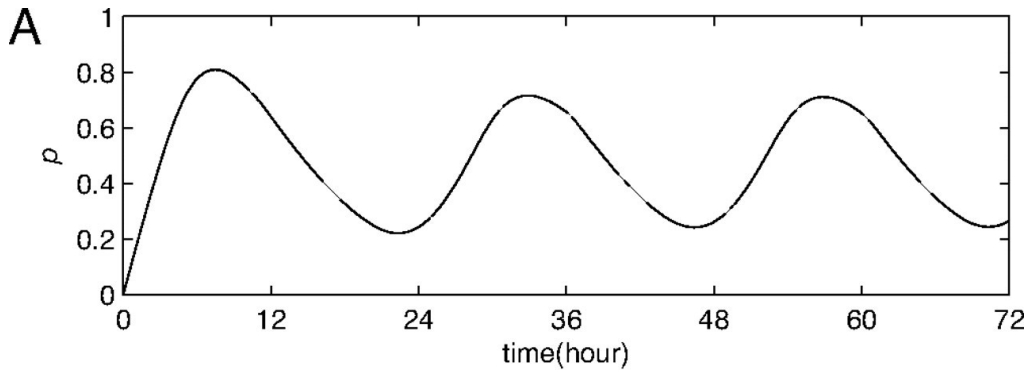
ODE Model



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 π Model

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

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

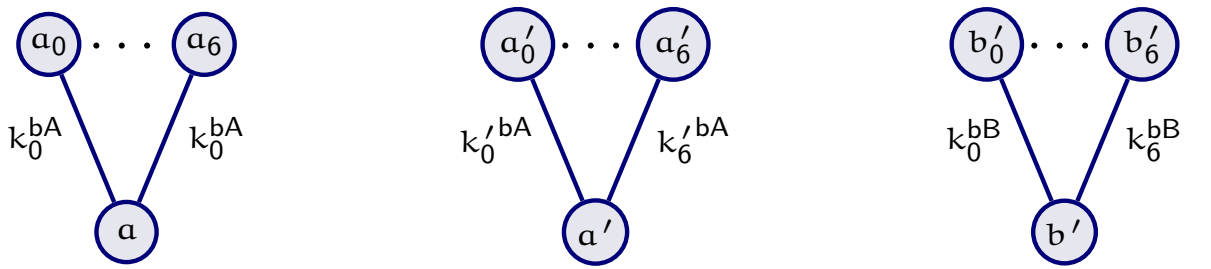
$$BC'_i = (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \mid B \mid B)) + (a'_i.a'_i.ABC'_i)$$

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

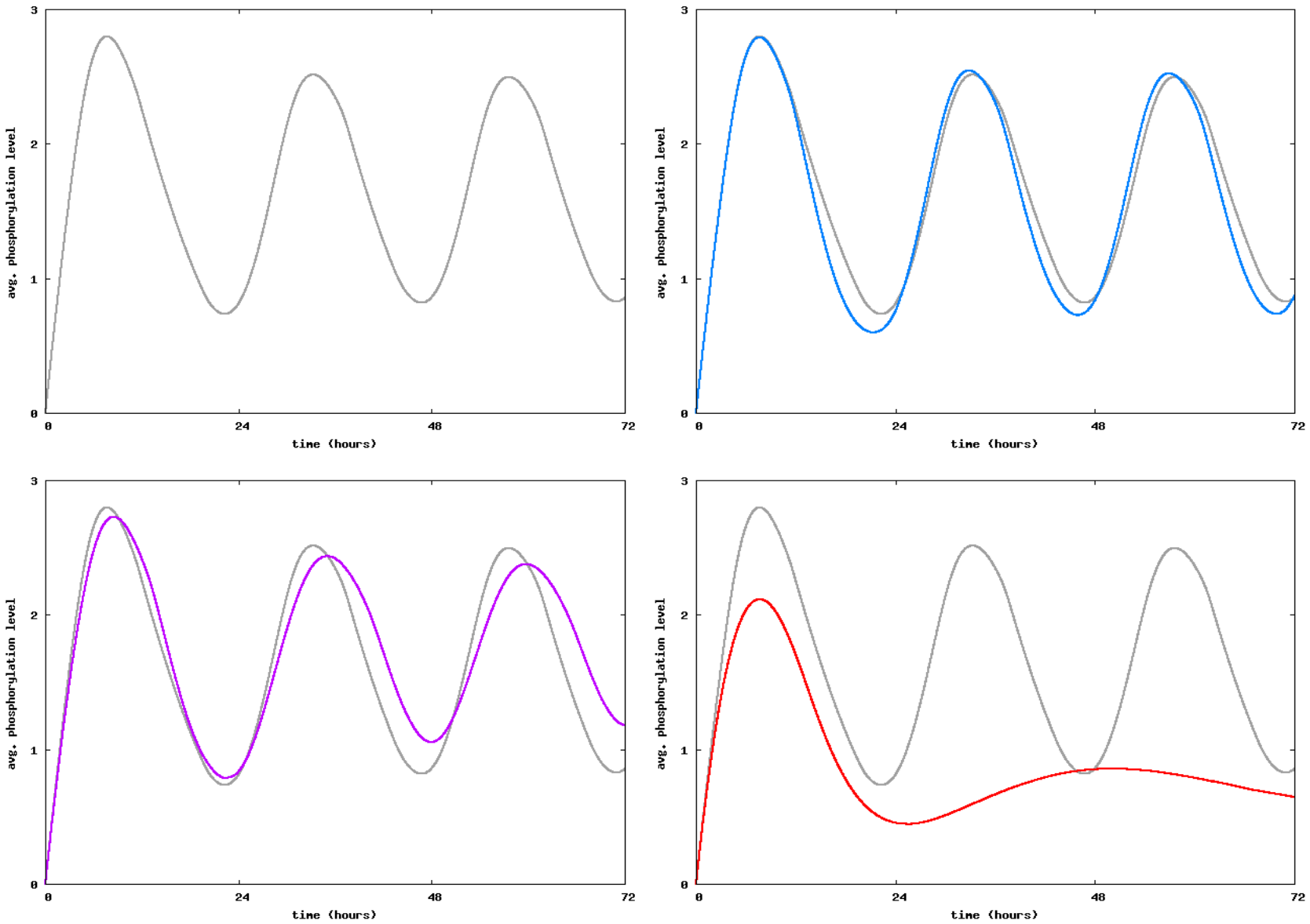
$$A = a(x).x.A + a'.0$$

$$B = b'.0$$

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



Execution and Modification



Modification: KaiA-KaiB dimers

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

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

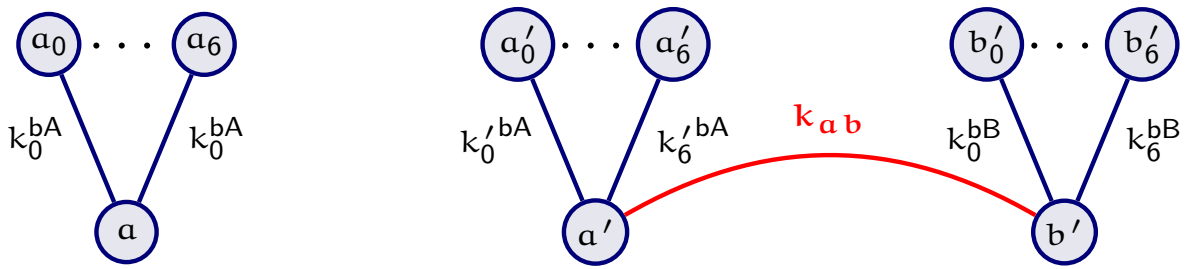
$$BC'_i = (\tau@k'_p.BC'_{i+1}) + (\tau@k'_d.BC'_{i-1}) + (\tau@k_i^{uB}.(C'_i \mid B \mid B)) + (a'_i.a'_i.ABC'_i)$$

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

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

$$B = b'.0$$

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



Process Algebras for Molecular Evolution

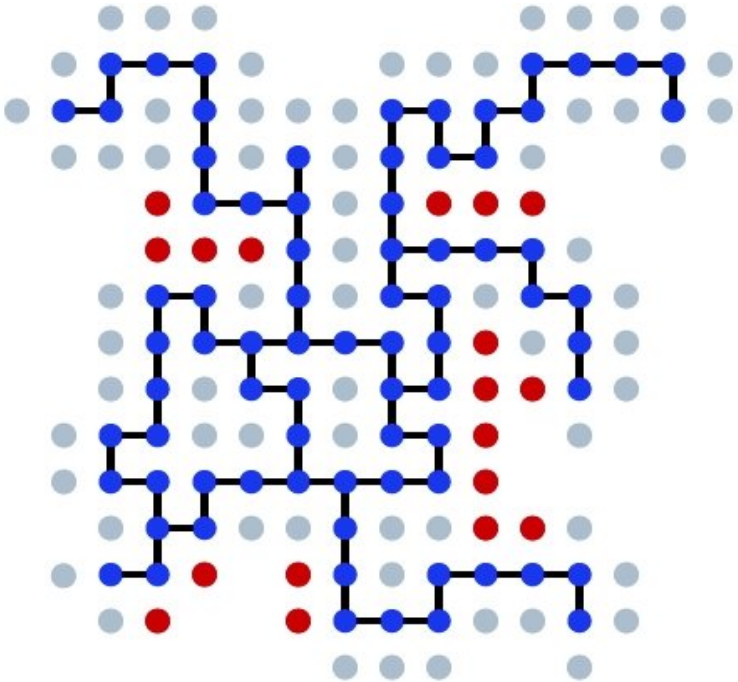
Process algebras, and languages in general, offer a framework for exploring molecular evolution beyond that of individual concrete mathematical models.

$$\text{Process} \sim \text{Genotype}$$
$$\text{Execution} \sim \text{Development}$$
$$\text{Behaviour} \sim \text{Phenotype}$$

Relevant features of models like continuous π include:

- Agent-based models to match genetic variation
- Free formation of new terms, particularly novel complexes
- Compute behaviour of created components (combinatorial explosion)

Remember Neutral Spaces?



We need:

1. genotype space
   (done: $c\pi$ models)
2. phenotype space
   (done: model dynamics)
3. a mapping between the two
   (done: ODE extraction)
4. accessibility relation

Variation Operators

Variation operators are transformations of $c\pi$ models which correspond to evolutionary events.

$$(\mathcal{D}ef, \mathcal{A}ff, \mathcal{P}) \longrightarrow (\mathcal{D}ef', \mathcal{A}ff', \mathcal{P}')$$

Ideally, a suite of such operations should:

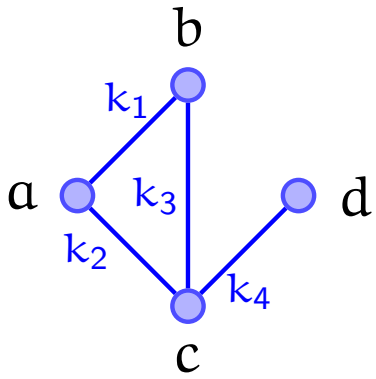
- Maintain the biological idiom;
- Be biologically meaningful;
- Be expressive enough to build new reaction networks from scratch.

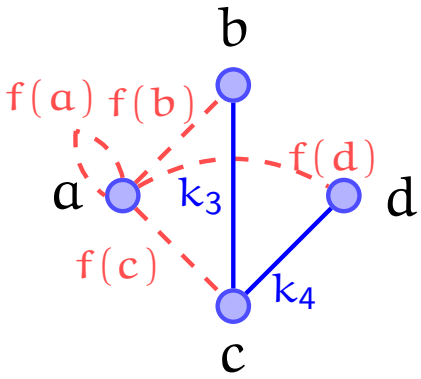
Variation Operators

Variation operators are transformations of $c\pi$ models which correspond to evolutionary events.

$$(\mathcal{D}ef, \mathcal{A}ff, \mathcal{P}) \longrightarrow (\mathcal{D}ef', \mathcal{A}ff', \mathcal{P}')$$

For example: site reconfiguration



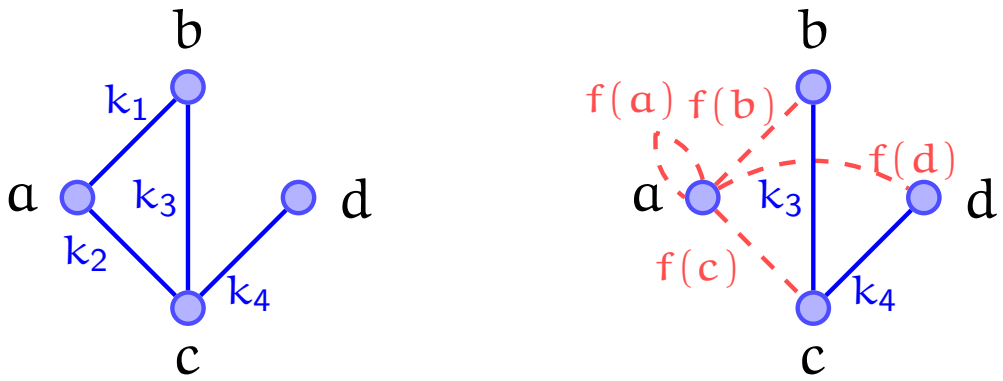


Variation Operators

Variation operators are transformations of $c\pi$ models which correspond to evolutionary events.

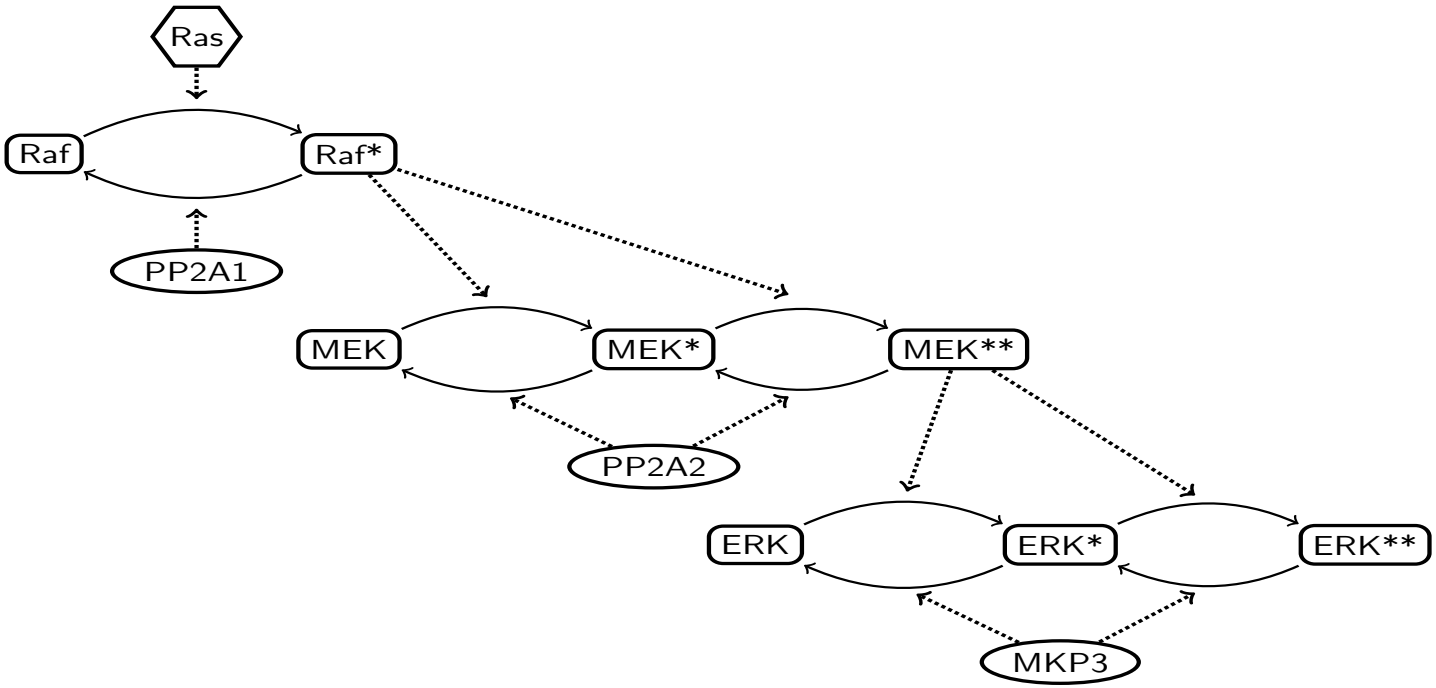
$$(\mathcal{D}ef, \mathcal{A}ff, \mathcal{P}) \longrightarrow (\mathcal{D}ef', \mathcal{A}ff', \mathcal{P}')$$

For example: site reconfiguration



We have defined a dozen operators modelling gene duplications, gene knockouts, changes in complex activity rates, and more.

The MAPK Cascade



MAPK: Mitogen-activated protein kinase cascades

- Functionally conserved across all eukaryotes
- Crucial component of many signalling pathways
- Relays and amplifies a signal
- Benchmark for new modelling techniques

MAPK in cπ

$$Ras = (\boldsymbol{\nu} x \frown \overline{x})\, ras(x;y).(\overline{x}.Ras + y.Ras)$$

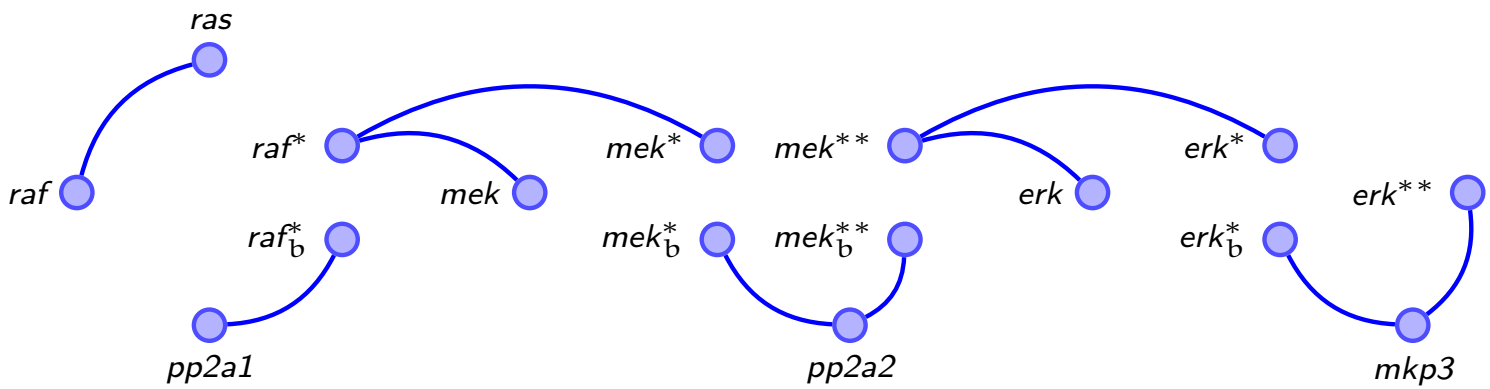
$$Raf = (\boldsymbol{\nu} x \frown \overline{x})\, raf(x;y).(\overline{x}.Raf + y.Raf^{*})$$

$$\ldots$$

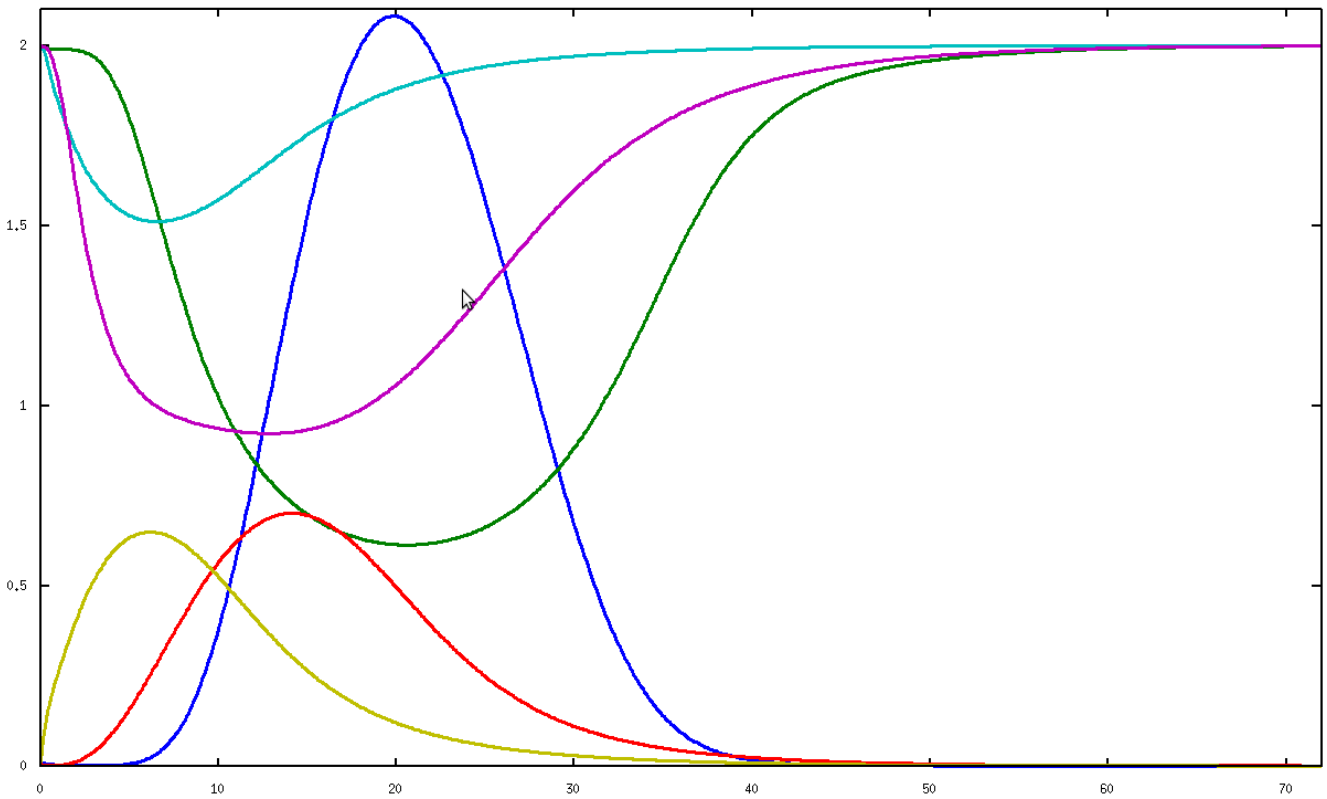
$$ERK^{**} = (\boldsymbol{\nu} x \frown \overline{x})\, erk^{**}_{b}(x;y).(\overline{x}.ERK^{**} + y.ERK^{*})$$

$$MKP3 = (\boldsymbol{\nu} x \frown \overline{x})\, mkp3(x;y).(\overline{x}.MKP3 + y.MKP3)$$

$$\Pi = \mathbf{c_1} \cdot \textbf{\textit{Raf}} \parallel \mathbf{c_2} \cdot \textbf{\textit{Ras}} \parallel \ldots \parallel \mathbf{c_4} \cdot \textbf{\textit{ERK}} \parallel \mathbf{c_7} \cdot \textbf{\textit{MKP3}}$$

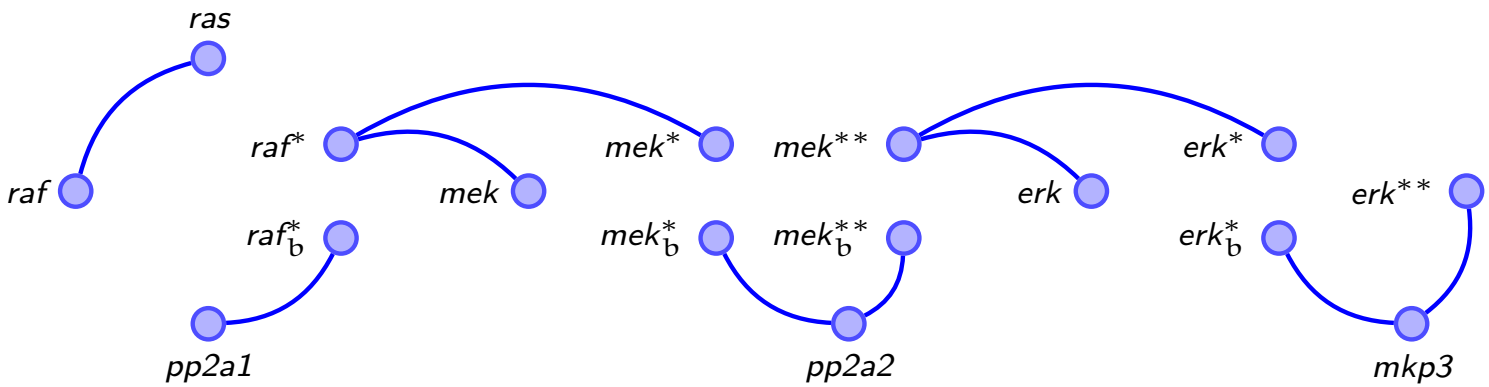


MAPK Behaviour



The tool compiles MAPK into 23 differential equations, which are then solved with Octave. The signalling cascade correctly transmits initial presence of **Ras** into a peak of **ERK**** via **Raf*** and **MEK****.

Evolutionary Analysis of MAPK



- Reconfigure every site in every way possible ($16 \times 2^{16} \approx 10^6$).
- Generate ODEs and thus behaviour traces for every variant.

Qualitative analysis

- Classify phenotypes with LTL model-checking
- Find evolutionarily fragile and robust sites

Quantitative analysis

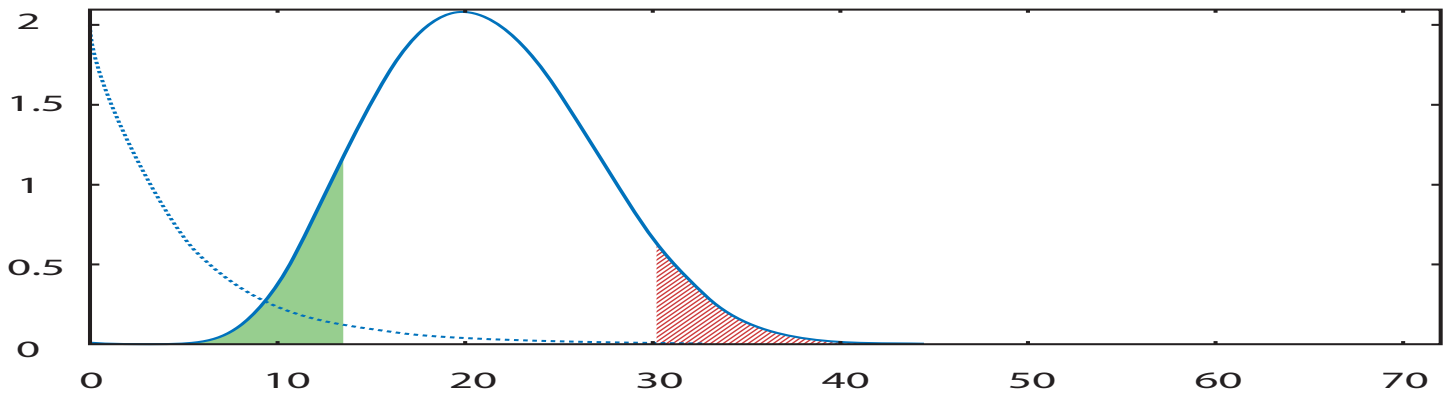
- Compute the fitness of every variant using signal integration
- Find the distribution of mutation effects on fitness

Phenotype Classes and Fitness

Phenotype classes

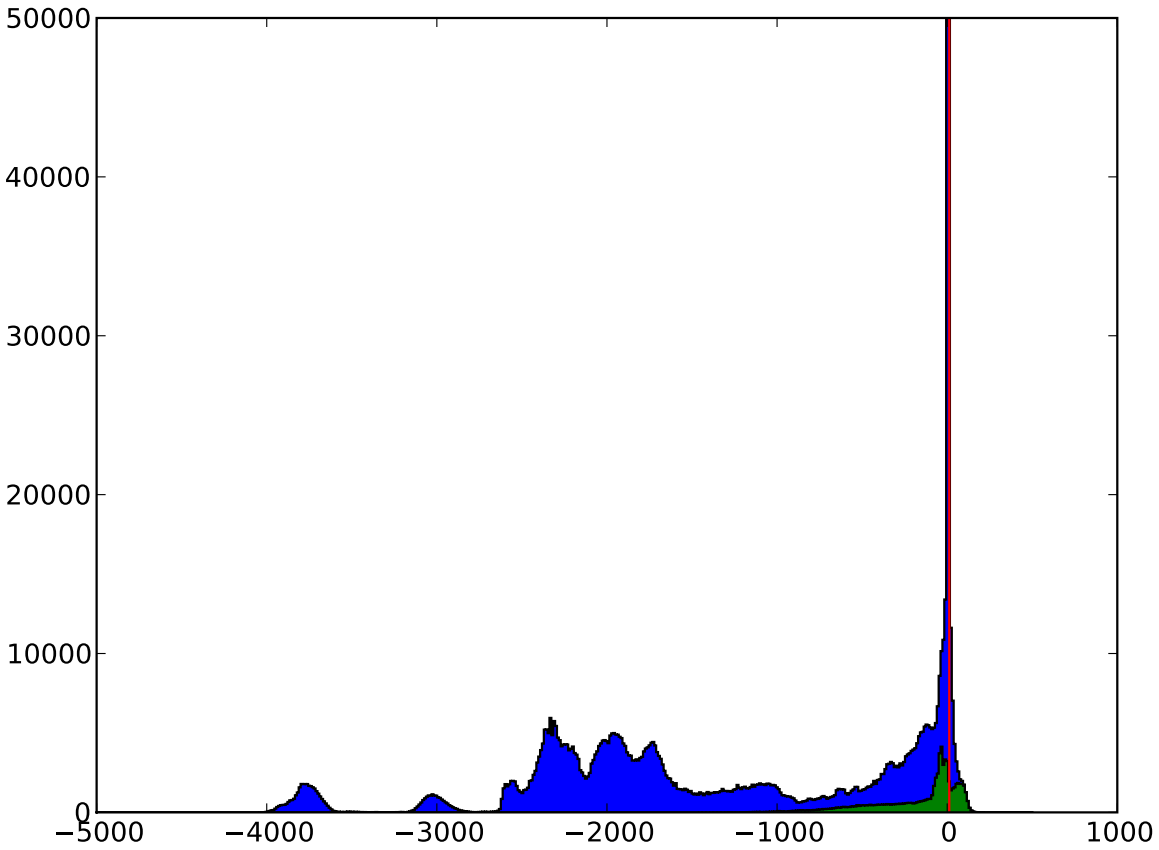
- Four categories: peak, switch, oscillatory, noise.
- Automatically identified using LTL checking.
- Results: peak 7.0%; switch 45.2%; oscillatory 0.0; noise 47.8%.

Fitness



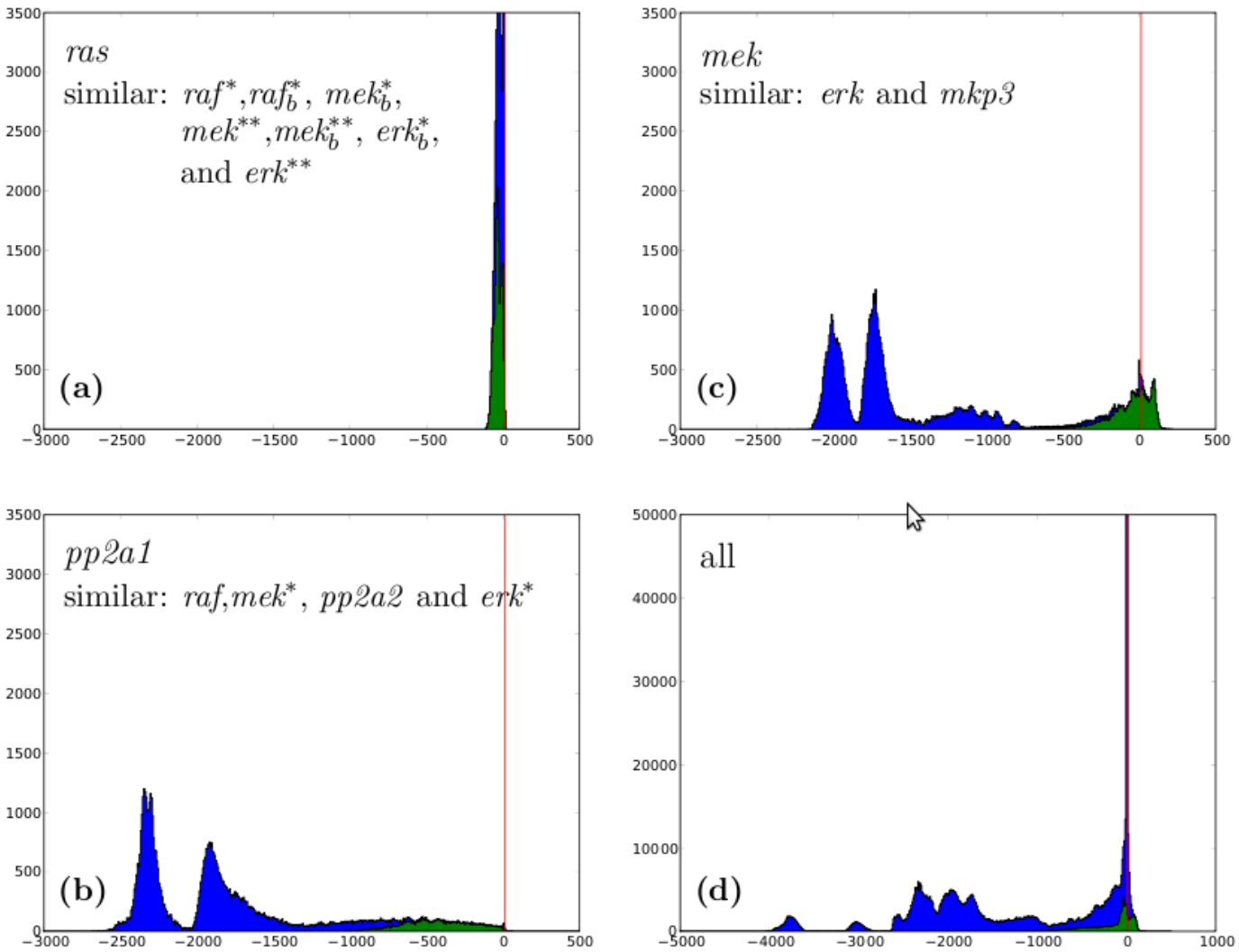
Fitness is the area marked green minus the area marked red.

Fitness Distribution

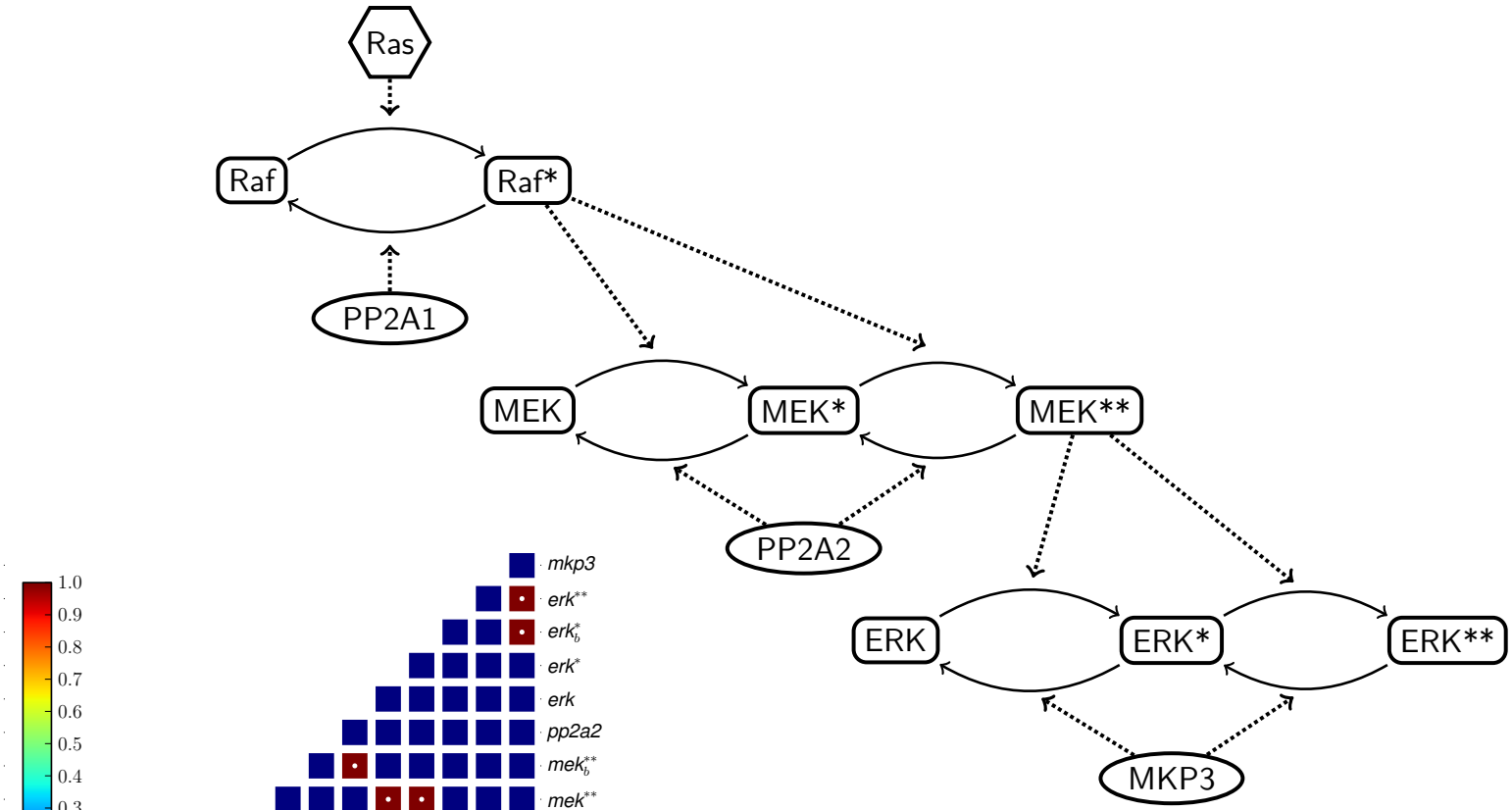


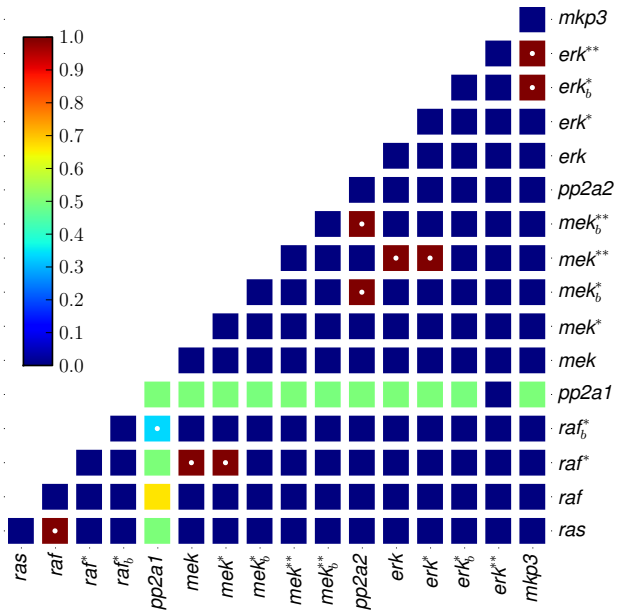
Histogram with 500 evenly-sized bins; green sections are *peak* variants; red vertical line shows initial model.

Fitness Distributions by Site Modified

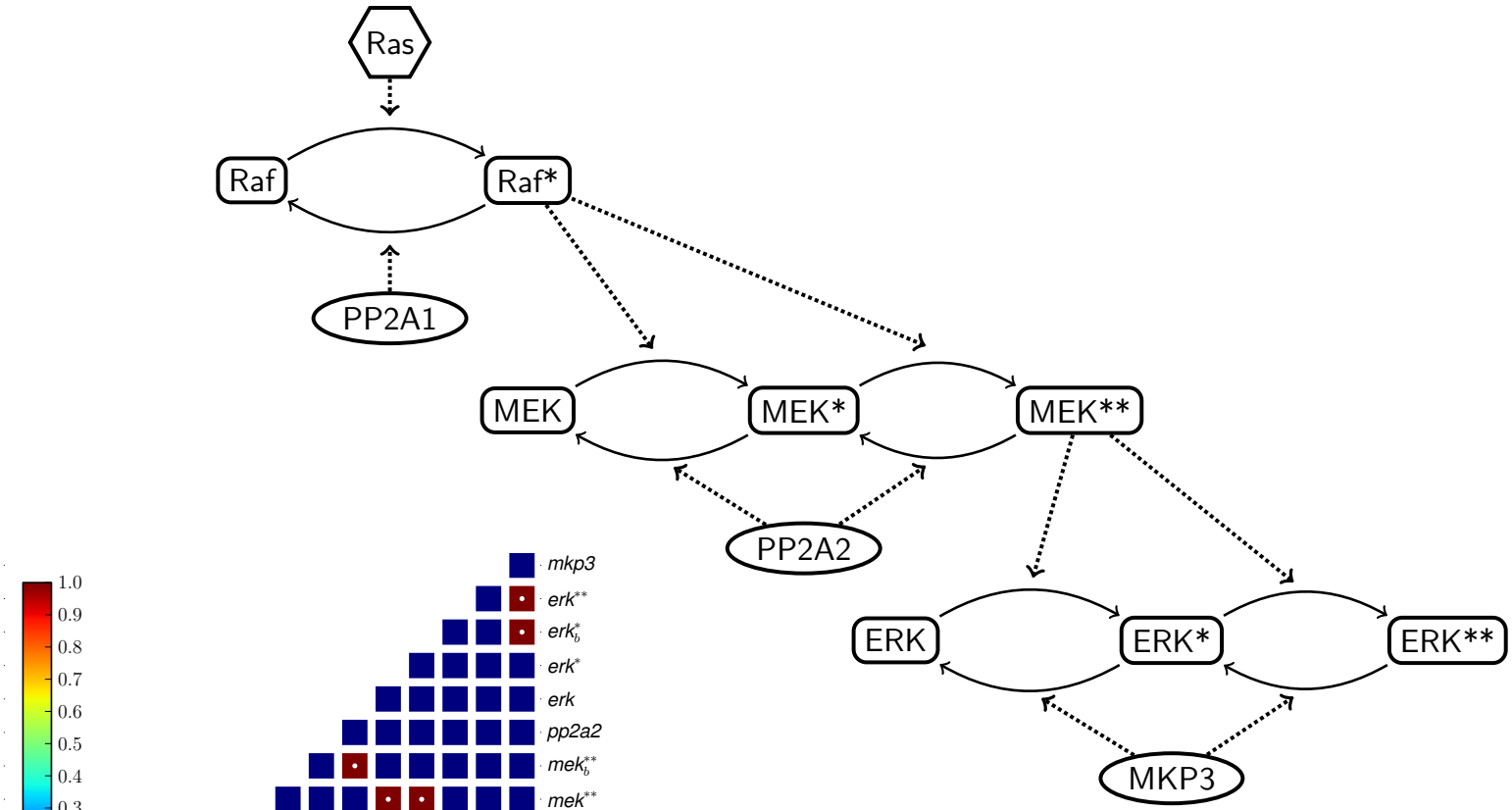


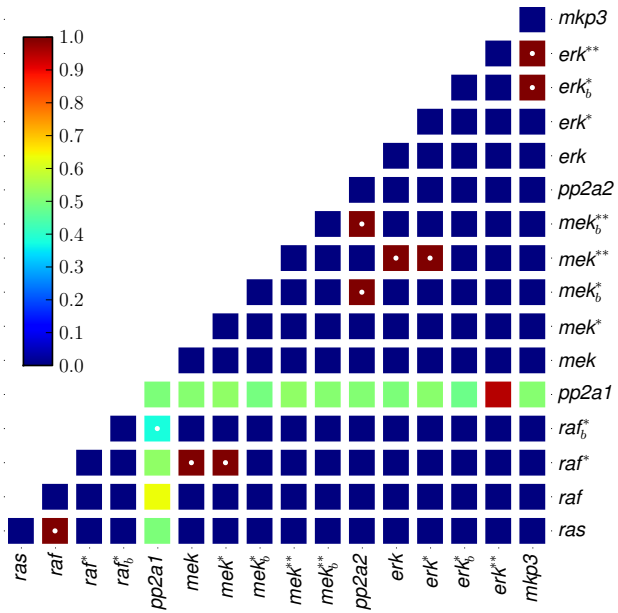
Less Fit Peaks (Left)



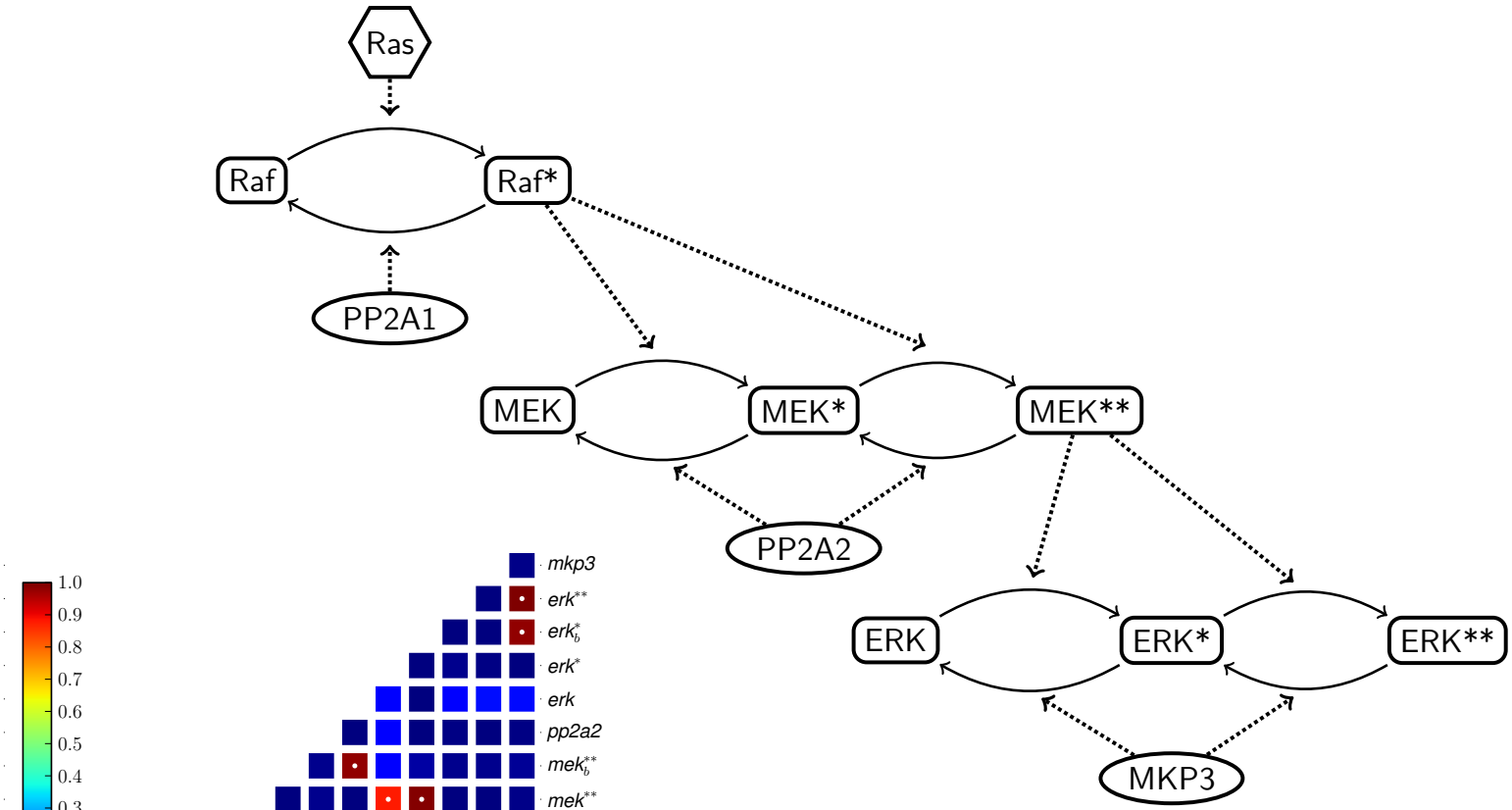


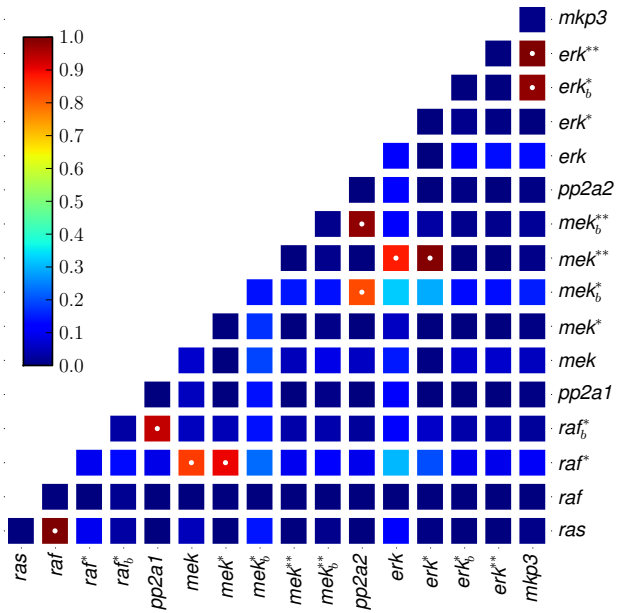
Less Fit Peaks (Right)





Advantageous Mutations





Observations

We have been able to explore the complete one-step evolutionary neighbourhood of a MAPK cascade under modifications of site activity.

For this model, we observe:

- Signal transmission has some robustness.
- Switch behaviour is readily accessible.
- Almost all mutations reduce fitness, although many only slightly so.
- A few give improvement against the chosen fitness measure.

Summary

The *continuous* π*-calculus* ($c\pi$) is a process algebra for modelling behaviour and variation in biomolecular systems: *e.g.* enzyme activation and inhibition; circadian clocks; signalling pathways.

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

High-level languages for biological descriptions can smooth the route from mechanism descriptions to mathematically precise models; and also help to express and test high-level hypotheses.

With a language of potential changes in $c\pi$ processes we systematically explore the evolutionary neighbourhoods of a specific signalling pathway, and observe instances of robustness, neutrality and evolvability.

Conclusion

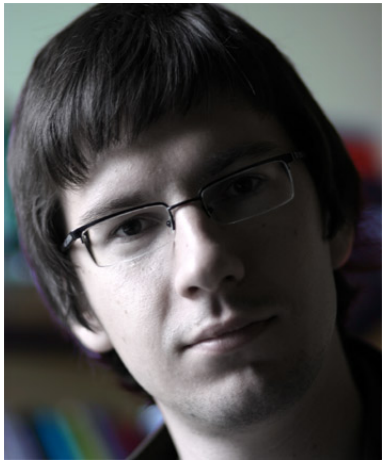
Limitations

- Challenge of $c\pi$ expressiveness: stay within the biology
- Artificiality of behaviour modelling within complexes
- Low-count species (DNA) and discrete state transitions

Further Directions

- Temporal logic to describe system behaviour $P \models G_{\leqslant t}(\phi)$
- Guarantee for behaviour-in-context $P \models F(Q \triangleright \phi)$
- Other non-transcriptional clocks; bistable systems

The Continuous π-makers



Marek Kwiatkowski
Institute of Integrative Biology
Department of Environmental Sciences
EAWAG & ETH Zürich
http://mareklab.org

Seeking a job in evolutionary aspects of theoretical/ computational/systems biology. Hire him, he's excellent.

Chris Banks
Laboratory for Foundations of Computer Science
School of Informatics
The University of Edinburgh
http://banks.ac

PhD student 2010–

References


Kwiatkowski and Stark.
On Executable Models of Molecular Evolution. In *Proc. 8th International Workshop on Computational Systems Biology WCSB 2011*, pp. 105–108.


Kwiatkowski and Stark.
The Continuous π-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


Kwiatkowski.
A Formal Computational Framework for the Study of Molecular Evolution
PhD Dissertation, University of Edinburgh, December 2010.


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