

Stochastic Modelling of the Kai-based Circadian Clock

Chris Banks¹

School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Allan Clark²

SynthSys, University of Edinburgh, Edinburgh, United Kingdom

Anastasis Georgoulas³

SynthSys, University of Edinburgh, Edinburgh, United Kingdom

Stephen Gilmore⁴

School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Jane Hillston⁵

School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Dimitrios Milios⁶

School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Ian Stark⁷

School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Abstract

We present two process algebra models of a Kai-protein based circadian clock. Our models are represented in the Bio-PEPA and the continuous pi-calculus process algebras. The circadian clock is not based on transcription and has been shown to persist with a rhythmic signal when removed from a living cell. Our models allow us to speculate as to the mechanisms which allow for the rhythmic signals. We reproduce previous results based on ODE models and then use our models as the basis for stochastic simulation.

Keywords: Circadian, ODE, stochastic, temporal logic, Bio-PEPA, Continuous Pi

1 Introduction

Circadian clock systems are found in many living organisms and regulate many biological functions. Many circadian clock systems oscillate around a 24 hour period, rely upon transcription (the process of producing a complementary RNA copy of a sequence of DNA), and may be entrained by external time cues (often light). As such, circadian clock systems may be complex systems with a series of inputs and outputs. The system which we investigate here is a somewhat simple example found in cyanobacteria which produces a rhythmic signal which can be reproduced *in vitro*. The signal produced is the mean phosphorylation level of the KaiC protein and a good overview can be found here[12]. This has been studied previously with the use of Ordinary Differential Equations [5,7,6,9], and remains an interesting system of investigation due to its simplicity and the relatively low cost of performing experiments.

We present two process algebra-based models of the Kai-based circadian clock, the first written in Bio-PEPA, the second in the continuous pi-calculus. Both Bio-PEPA[4] and the continuous pi-calculus [11,10] have been designed for modelling the behaviour of biochemical systems but from different perspectives.

We base our models on those described by Van Zon *et al.* [15], who used an ODE-based model and were able to reproduce experimental observations. By producing a Bio-PEPA model for these phenomena, we can, from the same model, reproduce the ODE results of Van Zon *et al.* and investigate the role of stochasticity using stochastic simulation algorithms. In addition, we believe our process algebra model to be more amenable to *in silico* experimentation and we demonstrate this with an investigation of the role of so-called shortcut reactions which will be explained Section 4.

¹ Email: C.Banks@ed.ac.uk

² Email: a.d.clark@ed.ac.uk

³ Email: ageorgou@staffmail.ed.ac.uk

⁴ Email: stg@staffmail.ed.ac.uk

⁵ Email: Jane.Hillston@ed.ac.uk

⁶ Email: d.milios@sms.ed.ac.uk

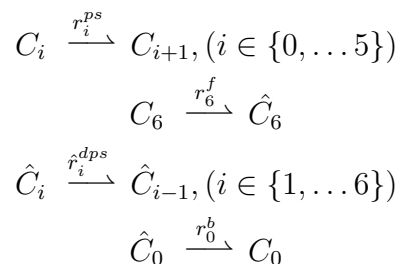
⁷ Email: Ian.Stark@ed.ac.uk

2 Background

The system under investigation is based upon three proteins KaiA, KaiB and KaiC. The rhythmic signal that is produced is the mean phosphorylation level of the KaiC protein. The phosphorylation level ultimately regulates the KaiC protein's activity as a global transcriptional repressor. However, this rhythmic signal has been shown to persist when transcription and translation have been inhibited, and in addition, the signal can be reproduced in *in vitro* experiments with only the Kai{A,B,C} proteins and Adenosine Triphosphate (ATP) present[13].

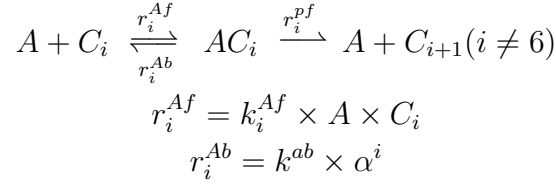
The KaiC proteins form hexamers. Each monomer within a KaiC hexamer can be individually phosphorylated (the addition of a phosphate group). Thus a single KaiC hexamer can exist in seven distinct levels of phosphorylation corresponding to the number of monomers within the hexamer which are currently phosphorylated. In addition each KaiC hexamer can exist in one of two states, the active and the inactive state. A hexamer in the active state is more readily phosphorylated than dephosphorylated and the reverse is true for a hexamer in the inactive state. Thus a single KaiC hexamer may exist in one of 14 states, corresponding to the seven distinct levels of phosphorylation in both the active and the inactive states.

A single KaiC hexamer can cycle through the fourteen possible states via the following reactions (note that by convention we use the 'r' prefix for a rate and the 'k' prefix for a constant):



Here C_i denotes a KaiC hexamer in the active state with i phosphorylated monomers and \hat{C}_i denotes the same for a KaiC hexamer in the inactive configuration. So the $C_0 \xrightarrow{r_0^{ps}} C_1$ reaction is the non-catalysed phosphorylation of a KaiC hexamer with zero phosphorylated monomers which occurs at rate r_0^{ps} . The rates r_6^f and r_0^b are associated with the reactions which flip the KaiC hexamers between the active and the inactive configurations. To achieve a macro-level signal such cycles must be somehow synchronised. This is the role of the KaiA and the KaiB proteins. It has been proposed that the KaiA protein catalyses the phosphorylation of the KaiC hexamers in the active state, but does so with a preference for those KaiC hexamers with a low level of

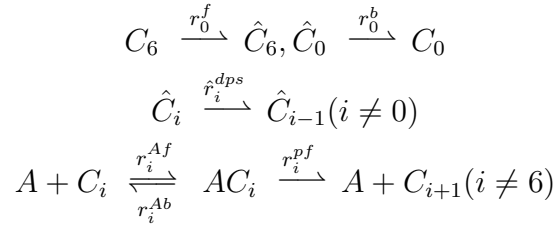
phosphorylation. The catalysed phosphorylation of active KaiC is written as:



The KaiC hexamers bind together with the KaiA protein at a rate which we initially set to be equal, that is $k_i^{Af} = k_0^{Af}$. However the rate of dissociation is greater for those KaiC hexamers with a larger level of phosphorylation which are therefore less likely to continue the catalysed addition of a further phosphate group and thereby increase their phosphorylation level. We can ensure this by setting $\alpha > 1$ since this makes $r_i^{Ab} < r_{i+1}^{Ab}$ because $\alpha^i < \alpha^{i+1}$. The role of the KaiB protein seems to be to sequester away the KaiA protein when sufficient concentration of the KaiC hexamers are in the inactive state, in a manner which will be further described in Section 3.2.

3 The Simple Model

The first model follows the approach taken by Van Zon *et al.* and models the preferential catalysing of the phosphorylation of active KaiC hexamers. In this model each individual KaiC hexamer may be in one of twenty states as shown in Figure 1. This gives rise to the reaction set:



In Bio-PEPA we define each biochemical entity as a distinct species and thus each state of the KaiC hexamer is a separate species. For example, a KaiC hexamer with a phosphorylation level of four in the active state is modelled by the following process definition:

$$C_4 = (r_i^{Af}, 1) \downarrow + (r_4^{Ab}, 1) \uparrow + (r_3^{pf}, 1) \uparrow$$

In Bio-PEPA each species is defined to have a number of behaviours where a behaviour is written as $(name, i)op$. The *name* part is the name of the

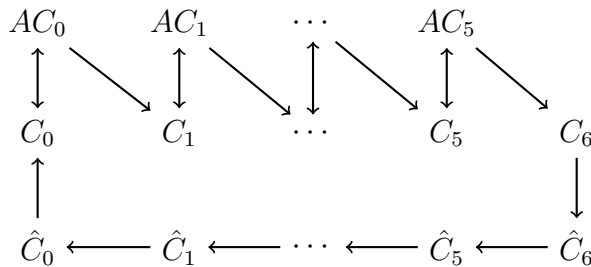


Fig. 1. Depicts the possible states of the KaiC hexamers in the simple model.

reaction for which the behaviour is being defined. The i is the stoichiometry, essentially how many parts of the species being defined are involved in the reaction. Finally op is the role which the defined species plays in the reaction $name$, it may be one of the following: \downarrow the species is a reactant, \uparrow the species is a product, \oplus the species is an activator for the reaction, \ominus the species inhibits the reaction, \odot a general modifier operator.

As described above the rates which govern the dissociation from the AC_i states are different according to the level of phosphorylation equal to the suffix i . This is achieved by setting $r_i^{Ab} = k^{ab} \times \alpha^i$, where k^{ab} is a constant. When α is greater than 1 this means that the higher the level of phosphorylation the greater the rate of dissociation. It is due to this that we believe that KaiC hexamers which are lagging behind their peers in the cycle are preferentially assisted in catalysed phosphorylation allowing them to catch up. In this way the cycles are synchronised. Note that this explains why the individual cycles are synchronised but not necessarily why those synchronised cycles would retain a steady period under stochastic conditions.

3.1 Analysis

We have first written this model in Bio-PEPA, and used the Bio-PEPA software[2] to convert the model into the SBML[8] format suitable for use with the SBSI software[14]. The SBSI software is used to perform parameter optimisation and is particularly applicable for models exhibiting oscillating behaviour. We have used SBSI to estimate parameters which allow our model to give an oscillating signal. Figure 2 shows the time course data from evaluating the model (via ODEs) both pre- and post-optimisation. This shows that the optimisation procedure has allowed us to identify parameters which allow this model to oscillate with a strong and maintained signal.

A “Tracker” component is added to the Bio-PEPA model in order to track the level of phosphorylation. The value of the Tracker is equal to the number of phosphorylated KaiC monomers (of which six comprise a single KaiC hexamer). The “Tracker” component for the simple model is defined by the

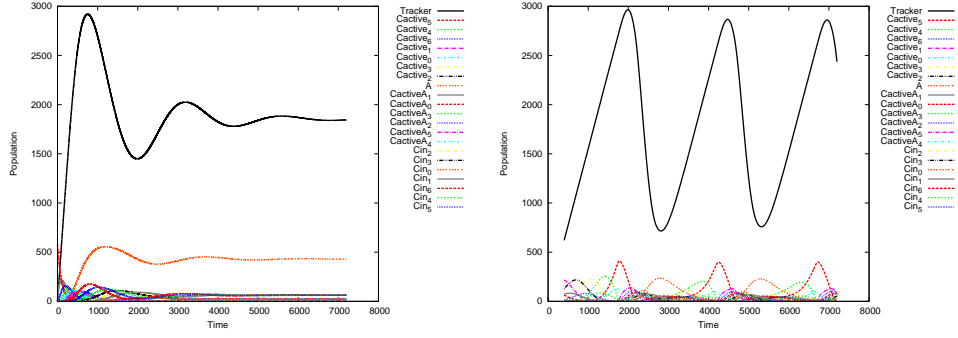


Fig. 2. The graph on the left shows the simple model with rate parameters taken from the literature. This shows an oscillating signal but with a dampened amplitude as time proceeds to the right. The graph on the right shows the same model with optimised rate parameters which gives a more maintained oscillating signal.

following component definition.

$$Tracker = (r_i^{dps}, 1) \downarrow Tracker + (r_j^{pf}, 1) \uparrow Tracker$$

for $0 < i < 7$ and $0 \leq j < 6$. The rates of each reaction remain unaffected which also means that they should not refer to the “Tracker” component even though it may be defined as a reactant. One nice feature of this Tracker component is that we need not modify any of the other process definitions.

The unoptimised version clearly shows an oscillating signal but one which is dampening, however the optimised version maintains a strong rhythmic signal. The rates are mass action except that where “Tracker” is a reactant the rate will be unaffected by the population of “Tracker”. As noted before by convention we use the prefix ‘r’ for a rate law and ‘k’ for a rate constant parameter, so in general the rate $r_i^X = k_i^X \times \text{reactant populations}$. Except for the unbinding rate laws r_i^{Ab} which we have noted to depend upon $k^{Ab} \times \alpha^i$. Additionally, all the k_i^{pf} are equal as are all the \hat{k}_i^{dps} . Here are the optimised parameter values:

$\alpha = 3.341$	$k^{ab} = 1$	$k_0^{af} = 1720000$
$k_6^f = 1.155$	$\hat{k}^{dps} = 1.3083345$	$k_1^{af} = 1720000$
$r_0^b = 0.255$	$k^{pf} = 1.0157632$	$k_2^{af} = 1720000$
$\text{init}_A = 153$		$k_3^{af} = 1720000$
$\text{init}_{C.0} = 580$		$k_4^{af} = 1170734.1$
		$k_5^{af} = 17200$

It should be noted that although we conjectured that the constant α which controls the difference in the dissociation rates should be greater than 1 to

allow those KaiC hexamers which are lagging behind in the cycle to catch up, we did not constrain it to be so. It was also possible that setting α to a number less than one would allow those KaiC hexamers ahead to more quickly move to the inactive state and thus complete a full extra cycle. However our optimised α parameter was greater than one.

Figure 3 shows the results of performing stochastic simulation over the simple model with the optimised parameter values. Each graph depicts the time course for the number of phosphorylated monomers within KaiC hexamers and the number of KaiA dimers which are not bound to any KaiC. The first thing to notice is that the stochastic simulation does produce results which agree with the ODE results in the first graph of the figure. Secondly the subsequent graphs depict simulations with varying amounts of initial KaiA dimers in the system. As we increase the amount of initial KaiA dimers we reduce the ability for the model to reproduce the rhythmic signal suggesting that the method of synchronisation for this simple model does indeed depend upon a low concentration of the KaiA protein. Since this is unsupported by experimental evidence, we seek a model to explain the rhythmic signal even in the presence of a large quantity of the KaiA protein.

3.2 *The Full Model*

To account for the continued observation of sustained oscillation even in the presence of higher concentrations of the KaiA protein we extend our simple model. As before we follow the route taken by Van Zon *et al.* and allow the KaiB dimers to bind with a KaiC hexamer in the inactive configuration (at any level of phosphorylation). Once bound, the complex may then further bind with free KaiA dimers and in this way sequester away KaiA dimers to allow the preferential catalysation of the phosphorylation of the active KaiC hexamers to once again synchronise the individual cycles.

The reactions for the full model are:

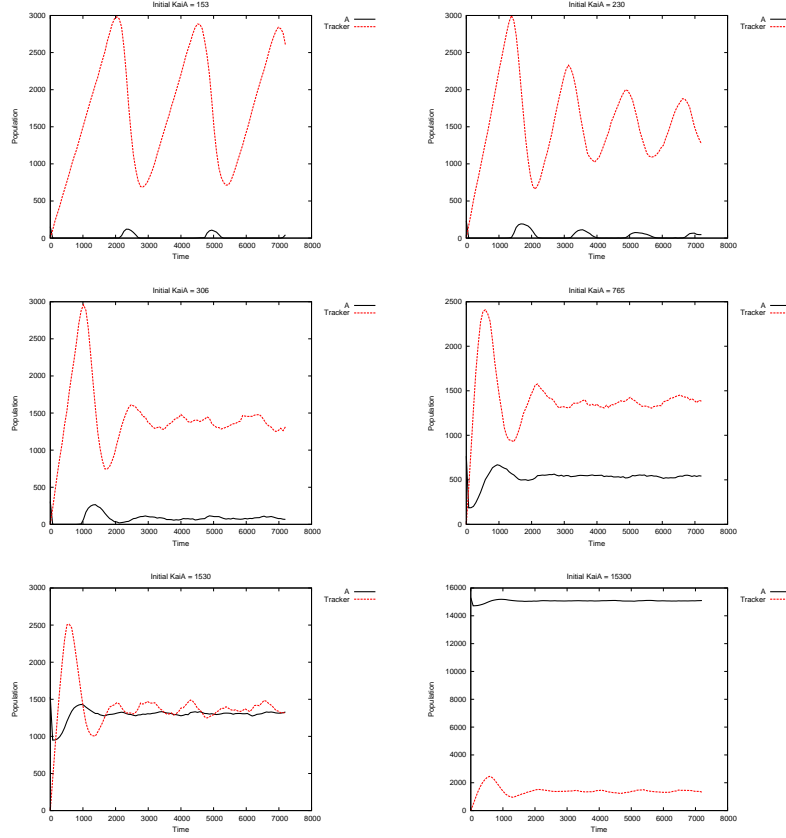


Fig. 3. The results of stochastic simulation using the optimised parameter values for the simple model. Here we have changed the initial population of the KaiA protein by multiplying it by increasing factors — the title of each plot gives the actual initial population of KaiA. The results support the hypothesis that the differential affinity can only account for the sustained oscillations if the population of the KaiA protein is sufficiently small.

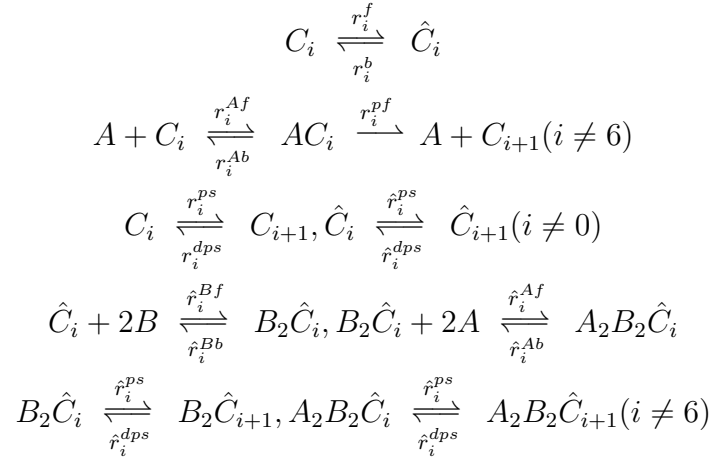


Figure 4 as before shows the time course data the model produces via

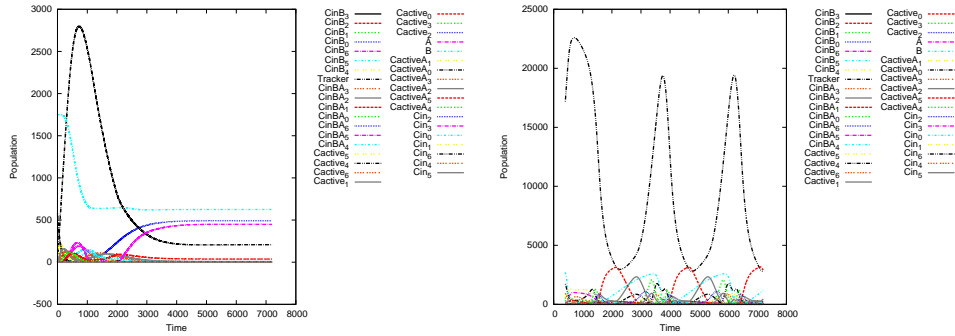


Fig. 4. The graph on the left shows the time course obtained by ODE solution from the full version of the model prior to optimisation of the rate constant parameters. No rhythmic cycle at all is detected. However, once we optimise the rate parameters, as the graph on the right reveals, we can obtain a sustained oscillating signal.

ODEs. Again we depict the time courses for both the unoptimised and the optimised versions of the model. We see that we are once again able to identify parameters which give us sustained oscillations.

We then attempted the same kind of experiment for the full model as we have performed for the simple model, that is using stochastic simulation to first check that we get similar behaviour and then increasing the initial population of the KaiA protein to garner the effect on the oscillation of the signal. However the first graph in Figure 5 depicts the time course produced from a stochastic simulation for the same model for which ODE analysis is producing an oscillating signal. As we can see the oscillating signal has been lost. We are still attempting to verify and explain this result. To add further to the mystery we have still performed our experiment of increasing the initial population of the KaiA protein, but using ODEs. As shown in the final two graphs of the same figure even a very modest increase in this initial population removes the oscillation from the signal produced. This is counter to our expectation that the full model is more resilient to increases in the population of KaiA protein. As before, it would be useful to force this initial population to be high and then perform parameter estimation.

4 The Role of ‘Shortcut’ Reactions

As mentioned before this circadian clock system has been studied before numerous times and one reason may be that because of the simplicity it is possible to investigate which features of the model are necessary for producing the oscillating signal and under which conditions. We are interested in exploring several of these, including whether oscillations may be possible with a protein that forms polymers other than hexamers. In this particular study we investigate the nature of the so-called “shortcut reactions”. The shortcut reactions are those which flip the state of a KaiC hexamer between the active and the

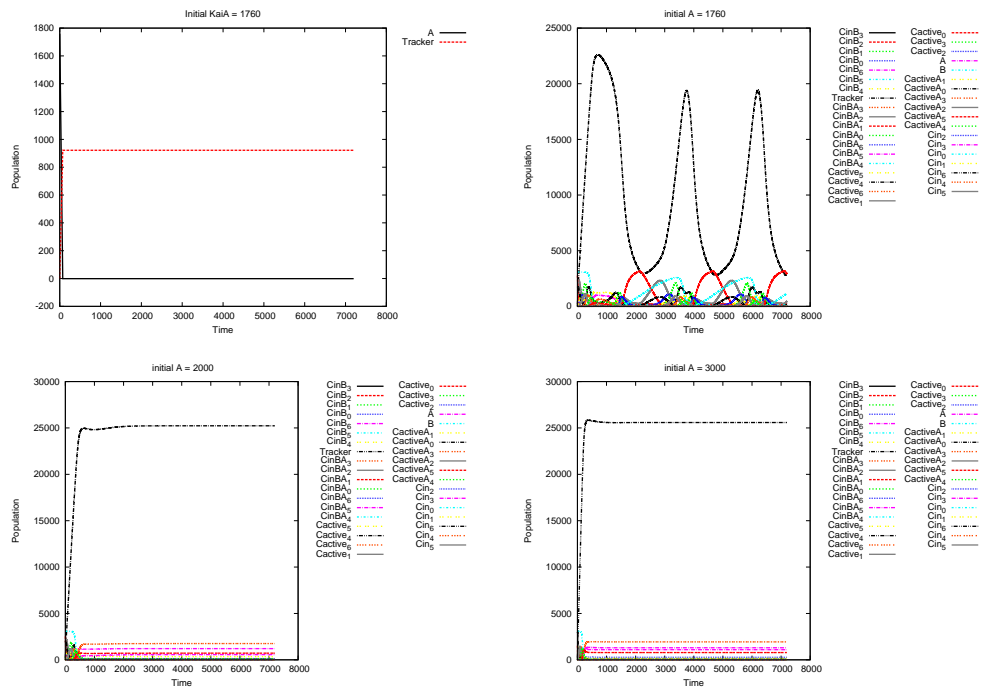
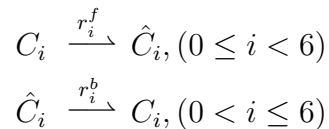


Fig. 5. The graph on the top left shows the full model with optimised parameters evaluated using stochastic simulation and the graph on the top right is the exact same model evaluated via ODEs. We see a startling difference. The graphs on the bottom increase the initial population of the KaiA protein, we see that even a very modest increase removes the oscillation from the signal produced.

inactive states whilst not at either end of the loop.



These are called the shortcut reactions since it is possible that they allow those KaiC hexamers which are behind the others in the phosphorylation cycle to catch up through bypassing either end of the loop. We are interested in evaluating whether these reactions are necessary to sustain the oscillations or whether they are simply a chemical reality that must be overcome in order to produce the oscillating signal. In other words, are the shortcut reactions, necessary, helpful or detrimental to the rhythmic signal. The simple version of the model containing no shortcuts gives some evidence that the shortcut reactions are not necessary. However, it may be that the phosphorylation level — which is the rhythmic signal produced — does not cycle around the loop but rather sways back and forth, conceptually like a tide rather than a loop.

Our first evidence that the shortcut reactions are not required comes from the results from our optimisation of the full model. These have returned very

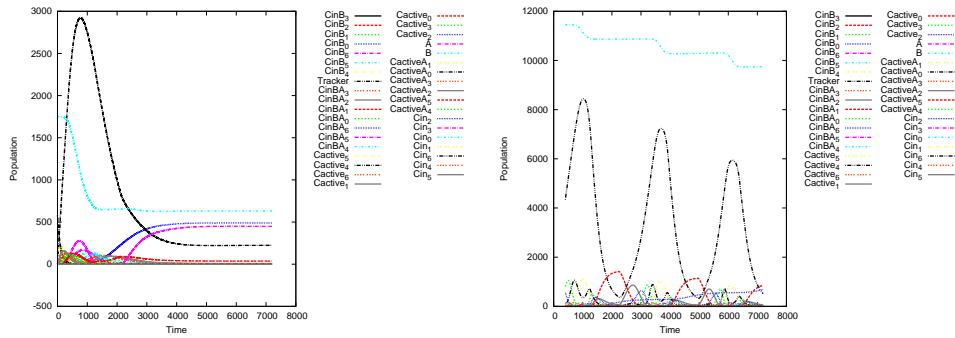


Fig. 6. ODE analysis of the version of the model which has no shortcut reactions. The graph on the left is prior to the optimisation of parameters while the graph on the right is after the optimisation of parameters. Once again parameter optimisation is able to find a configuration in which the model produces an oscillating signal but here the signal appears to dampen.

small values for the rate parameters associated with the shortcut reactions which flip the active state to the inactive state. However, there is a single rate parameter associated with the shortcut reactions which flips the inactive KaiC hexamers to the active configuration. In the above reaction the rate r_i^b is set to $C_i \times b$ where b is a constant which may be optimised. The results of optimisation give a relatively large value to the shared rate parameter b of 703.

This is an interesting result and highlights an advantage of performing parameter estimation. By performing parameter estimation we gain insight into the model in ways that we may not have otherwise thought to investigate. Our initial thought was to investigate the role of the shortcut reactions, but we had assumed that they were either all required, or unnecessary or perhaps even an impediment. We had not considered that the forward and backward shortcut reactions should be considered separately.

4.1 Removing the Shortcut Reactions

Despite the results of optimisation on the full model we went ahead with the experiment of removing the shortcut reactions altogether. We did so and once again performed parameter optimisation to see if we could find parameter values which would give us sustained oscillations. Figure 6 shows the time course produced by numerical evaluation of the full model with the shortcut reactions disabled after parameter optimisation. As we can see we can still produce an oscillating signal but this now dampens.

5 Modelling with the Continuous Pi-Calculus

The final model we present is the model of the full system using the *continuous pi-calculus* (c-pi) and a temporal logic of behaviour in context. The c-pi model generates differential equations for numerical simulation, while the

logic allows us to specify and check conjectures about the system’s behaviour over time and under varying conditions.

Our model exhibits the 24-hour cycling of phosphorylation levels in KaiC hexamers, and we plot the change in concentrations of the various species present. We formulate statements in the logic to capture the oscillation of mean phosphorylation level across the KaiC population and its dependence on the presence of KaiA. Model-checking these confirms that our model has this behaviour.

5.1 Method

The continuous pi-calculus is a reagent-centric process algebra. It provides a high-level language for describing the behaviour of individual chemical species, from which the behaviour of complexes, mixtures and derived species are automatically computed. The primary mode of execution for c-pi systems is to generate ordinary differential equations (ODEs) suitable for conventional numerical simulation.

In recent work we have developed a temporal logic of behaviour in context to capture properties of c-pi expressions. The basic observations in the logic are real-valued concentrations and rates of change in individual species. Temporal modalities F and G capture behaviour over time as usual in a continuous-time linear temporal logic: process P has behaviour $F(b)$ if at some time in the future it has behaviour b ; and it has behaviour $G(b)$ if it has behaviour b at all future times. Variants F_t and G_t look only at behaviour up to time t . Finally, we introduce a modality[3] for *behaviour in context*, based on the *guarantee* operator from spatial logics: a process P has behaviour $Q \triangleright b$ if, when P and Q are mixed, then the resulting system $P \parallel Q$ has behaviour b .

Purely temporal expressions in the logic describe features of behaviour that can be directly observed from a single execution trace. Expressions involving behaviour in context describe features involving multiple traces, with yet more expressive complexity when nesting the two kinds of modality.

In previous work, Kwiatkowski and Stark built a c-pi model for the *S. elongatus* circadian clock, based on the mechanism proposed by van Zon, and demonstrated its periodic behaviour [11]. Here we have reused some of their parameters and methods, but with a new tool and new analyses. The tool, the *continuous pi-calculus workbench* (CPiWB)[1] provides access to alternative ODE solvers and a model-checker for the logic of behaviour in context.

5.2 Model

We use a single expression in c-pi for each of KaiA and KaiB stating that they can bind and unbind at sites **a** and **b** respectively.

species A(a) = a(x).x.A(a);

species B(b) = b(x).x.B(b);

The rates of these bindings are determined by a global *affinity network* stating binding affinity between a, b and sites on other species.

...
a – a0 @ 1.72e5,
a – a1 @ 1.72e5,
a – a2 @ 1.72e5,
...

For KaiC we use a mutually recursive collection of declarations to capture all possible states: phosphorylation states from zero to six; active and inactive conformations; bound to KaiB dimers, and then to KaiA dimers too.

For example, the following clause for singly-phosphorylated KaiC in the active conformation indicates that it can: bind on a1 to form a complex which subsequently unbinds or reacts to phosphorylate the KaiC; constitutively phosphorylate or dephosphorylate; or switch to the inactive conformation. Here act, r and u are local names that mediate the internal behaviour of the KaiC-KaiA complex.

species C1() = {act-r@1.0, act-u@30}
a1<act>.(u.C1() + r.C2()) + tau<2.5e-2>.C2() +
tau<0.4>.C0()+ tau<1e-5>.CC1();

Similar clauses capture all of the other KaiC states. Finally, a single clause expresses the initial state of the system and its affinity network.

process Kai = [0.56] A(a) || [1.78] B(b) || [0.58] C0() : { a – a0 @ 1.72e5, ... };

5.3 Numerical Simulation

The source for the KaiA/B/C model includes 23 explicit species descriptions. The tool computes a further 41 species and complexes arising from their reactions, and generates 64 ODEs describing their behaviour.

The tool CPiWB can send these ODEs either to an internal solver, using the GNU Scientific Library (GSL), or out to GNU Octave, a numerical software toolkit similar to Matlab. In this case the ODEs appear too stiff for the internal solver to handle successfully so we applied the Octave *lsode* solver, making use of the system Jacobian computed symbolically by CPiWB.

Figure 7 shows the resulting time series of species concentrations. These clearly show oscillation over approximately a 24-hour period. The traces here only include unbound KaiC; there are also complexes with KaiA and KaiB. We can immediately see from the plot that there is ample KaiB available

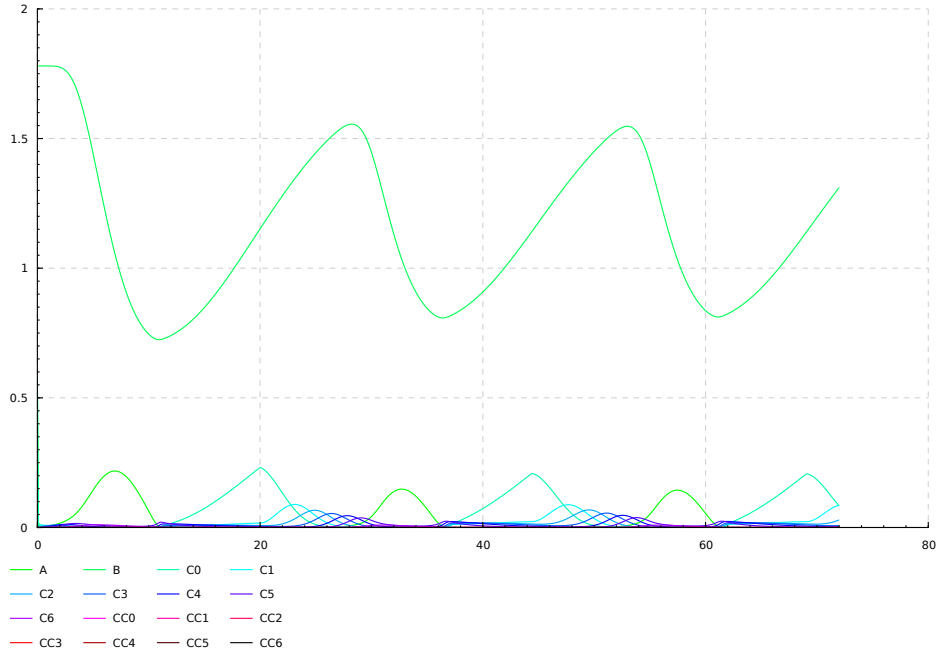


Fig. 7. The time series produced by using CPiWB to generate the ODEs for the Kai system and solving those ODEs using the *lsode* solver from Gnu Octave. There is a clear oscillating signal.

throughout the reaction, while for long periods there is little free KaiA, it being sequestered in the KaiA-KaiB-KaiC complex. Note also that only a small proportion of the KaiC becomes fully phosphorylated.

5.4 Temporal Logic

As well as generating direct plots of process behaviour, CPiWB can check that behaviour against properties specified in the temporal logic described earlier. In some cases that requires only a single time series, computed as before; more complex properties may require several time series and the solution of different ODEs under varying initial conditions.

For example, a simple query like “is there ever any fully phosphorylated KaiC?” is formulated as

$$F([C6]+[CC6] > 0.01)$$

We can also assess the mean phosphorylation level of KaiC in the system:

$$\text{phos} = (([C0]+[CC0])*0 + \dots + ([C6]+[CC6])*6) / ([C0]+[CC0]+\dots+[C6])$$

and then ask the following queries:

- $F(\text{phos} \geq x)$ Does the mean phosphorylation level ever exceed x ?
- $F(G(\text{phos} \geq x))$ Does the mean phosphorylation level eventually reach and then stay above x ?

By nesting temporal modalities we can ask whether behaviour repeats.

- $G\{t\}(F(\text{phos} > x) \text{ AND } F(\text{phos} < x))$ Does the mean phosphorylation level repeatedly pass above and below the value x for at least time t ?
- $G\{t\}(F([S] > y) \text{ AND } F([S] < y))$ Does the concentration of species S oscillate around y for at least time t ?

Here we are forced to use the G_t version of the always operator. This is because although conceptually we would like to make our G queries run to time infinity, in practice we are bound by the length of the simulation trace. So when we say $F(\text{phos} > x)$ we mean “Will the mean phosphorylation level ever rise above x ” but what we actually calculate is “Does the mean phosphorylation level rise above x before the end of the simulation trace”. The middle part of our query then: $F(\text{phos} > x) \text{ AND } F(\text{phos} < x)$ asks will the mean phosphorylation level both rise above x and fall below x by the end of the simulation trace. Clearly there will be time points towards the end of the simulation trace for which at least one of these is untrue. Therefore we have to bound our G operator to only look at points so far along the trace.

We may then define our own operators as composites of those provided by the CPiWB software. Here we make use of an operator for detecting oscillations of a given quantity around a given value.

$$\text{OscS}(q, v) = G\{t\}(F(q > v) \text{ AND } F(q < v))$$

5.5 Model Checking Results

For the purely temporal formulae above, it takes a minute to test all of the assertions. This includes generating all species transitions, creating the ODEs, numerically solving them, and carrying out model-checking on the resultant time series. This is single-threaded, on an Intel Core2 Duo T7500 running at 2.2GHz. Around 75% of CPU time is taken up with compiling the model and solving the ODEs, with the latter being the real bottleneck. The results are given in the table in Figure 8.

5.6 Formal Experiments

The context operator “ $- \triangleright -$ ” of the logic makes it possible to perform experiments upon the formal model: we can introduce new species and test the resulting behaviour.

As a very simple example, we can define an additional species **Inhib** which binds to the existing interaction site **a** of species **A**. This models an agent which can remove KaiA from the reaction. We can then write the following query:

- $!([0.5] \text{Inhib} \triangleright \text{OscS}(\text{phos}, 3))$ Does the introduction of the binding agent

Query	Meaning	Truth	CPU-Time(s)
$F([C6]+[CC6]>1.0e-2)$	Some fully phosphorylated KaiC does appear	True	0.48
$F(\text{phos} \geq 6)$	Mean phosphorylation rises to 6	False	0.22
$F(\text{phos} \geq 5)$	It rises to 5	True	0.00
$F(G(\text{phos} \geq 5))$	It remains above 5	False	0.23
$\text{OscS}(\text{phos}, 3)$	It oscillates continually rising above and falling below 3	True	0.00
$\text{OscS}([B], 1)$	Concentration of KaiB oscillates	True	0.00
$\text{OscS}([A], 0.1)$	Concentration of KaiA oscillates	True	0.00
$!([0.5]\text{Inhib} \mid > \text{OscS}(\text{phos}, 3))$	Introducing an additional binder to KaiA inhibits oscillation	True	60.0

Fig. 8. This table captures the results of model checking temporal logic formulae over our continuous pi-calculus model. Included are the execution times of each individual query. The purely temporal formulae are solved within a second whilst formulae using the context operator requires further numerical evaluation and takes significantly longer.

Inhib suppress oscillation in the concentration of KaiB? Here the inner part of the formula checks if the oscillations persist after the introduction of the binding agent and the ‘!’ mark is logical not.

Looking at the final entry in the CPiWB log above, we see that indeed it does. This is also visible if we plot the trace of the system with Inhib , as in Figure 9.

The table of model checking results also shows that checking this formula takes a further minute of CPU time. This is because testing behaviour in context involves calculating all transitions for the new c-pi terms, generating ODEs, and computing a new trace for the system. Nesting temporal and contextual modalities is potentially even more expensive:

- $G([0.5]\text{Inhib} \mid > !\text{OscS}(\text{phos}, 3))$ Does the introduction of the binding agent Inhib at any future time suppress oscillation in the concentration of KaiB from that point?

This requires generating a suite of time series solutions starting at each point in the original execution trace. This has a large impact on the computation time and indeed this query took nine and a half hours to complete on the

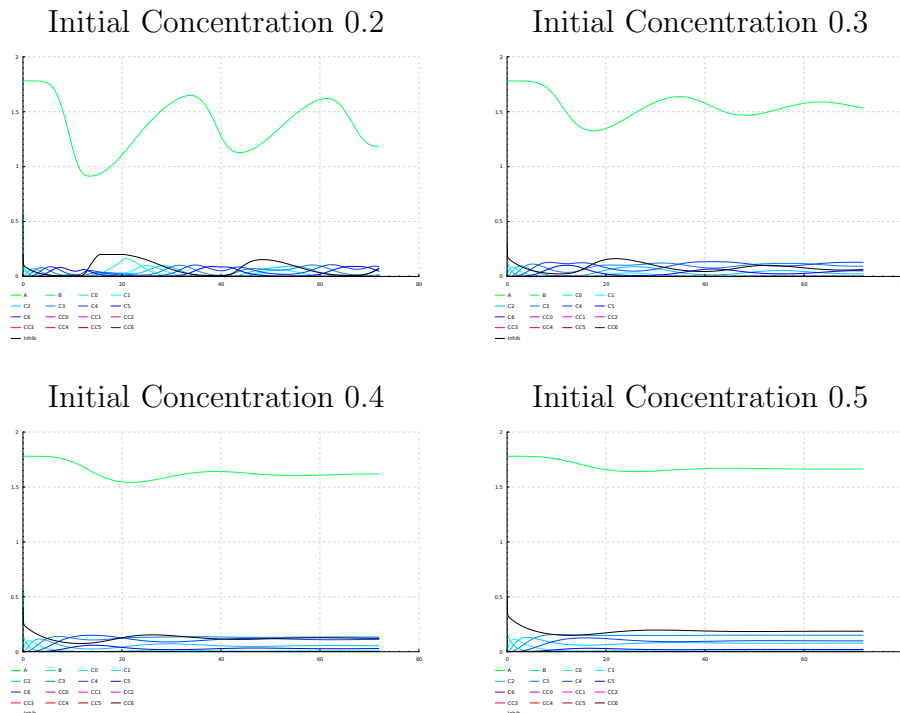


Fig. 9. CPiWB plots of the clock system with the addition of an inhibitor which is an additional agent binding to the KaiA protein and thus preventing the catalysed phosphorylation of the KaiC protein. For each plot we have a different initial concentration of the inhibitor showing that as we increase that initial concentration of the inhibitor the strength of the oscillating signal diminishes.

same desktop computer as all the other queries have been run.

6 Conclusions and Future Work

We have successfully reproduced a model of a biochemical circadian clock originally given as a series of ordinary differential equations in two process algebras. The process algebra models have allowed us to conduct some more experimentation on the model to garner further insight. In particular we were able to perform stochastic simulation of the models, perform parameter optimisation and test our hypotheses about the role of the shortcut reactions and the KaiB protein. The testing of hypotheses involved modifying the models in question.

We have also demonstrated here a novel software package, the CPiWB capable of analysing models in continuous-pi. Above this, the software implements a temporal logic for analysis of the time series generated from the numerical analysis of the model. The CPiWB is open source software and free to download. The temporal logic is powerful enough to include operators which allow us to perform *in silico* experiments on the model, and indeed experiments which involve changing the model part-way through a numerical

evaluation. In the current implementation such queries rely on the model being authored in the continuous pi-calculus, but could conceptually be applied to any kind of model with process composition, including Bio-PEPA. Additionally queries which do not make use of the context operator to modify the model may be made over time series produced from any kind of model such as one written in Bio-PEPA.

Our use of parameter estimation here is worth commenting upon. One way in which parameter estimation is used, is to calibrate a model of existing known behaviour in an effort to obtain a predictive model which can then be used for *in silico* experimentation. One problem that comes with this is determining how robust the results of parameter estimation are. In particular you may have identified parameters which allow your model to account for the observed data, but they may not be the only such set of parameters and may therefore be inappropriate for use in a modified model. Here we have used parameter estimation to gain further insight into the current model under investigation. Firstly, we have allowed the parameter estimation routine to return values that are not within our expectation. For example, the α parameter which governs the difference in the rates involved in the catalysed phosphorylation of the active KaiC hexamers, we had assumed must be greater than one, to give preference to those hexamers with a lower level of phosphorylation, but we did not constrain it to be such. In fact a future optimisation could constrain it to be less than one and determine whether this destroys any attempt to find a configuration of parameters which would allow a sustained oscillating signal.

In addition rather than performing parameter estimation and then modifying the model, we have modified the model and then performed parameter estimation, in order to gain insight on the role of the parts of the model which we have modified (in this case deleted). In particular this allows us to answer the question of whether or not the modification to the model has removed the ability to identify any parameter configuration which could allow the model to explain the observed phenomena.

We would still like to verify and explain our strange result for the full Bio-PEPA model which allows for sustained oscillations when numerically evaluated via ODEs but not when analysed with stochastic simulation. We should also perform some further parameter optimisations, in particular we have performed a range experiment on our models increasing the initial population of the KaiA protein. In particular we have done this for the simple model and then concluded that the simple model is not enough to explain the oscillations observed since they cannot be maintained in the presence of larger quantities of the KaiA protein. However it could be that the optimised parameters we have found are not well related to the physical parameters we are modelling, and that those true parameters would continue to allow oscil-

lations even in the presence of large amounts of KaiA. By performing further parameter optimisations on the simple model whilst constraining the initial population of KaiA to be high, we could give greater evidence that this is not the case.

There remain many more analyses which could be performed. In particular we would like to investigate the role of the number six. Is it important that the KaiC protein forms hexamers and thus gives rise to seven levels of phosphorylation, or are rhythmic signals equally obtainable if we change this number, either lower or higher? Further stochastic simulation analysis might illuminate the question of whether the synchronisation apparatus involved really gives an oscillating signal of a constant period or whether the signal is merely synchronised.

References

- [1] C. Banks. CPiWB Software Home Page, <http://banks.ac/software/>.
- [2] Bio-PEPA Home Page. <http://www.biopepa.org/>.
- [3] L. Cardelli and A. D. Gordon. Anytime, Anywhere: Modal Logics for Mobile Ambients. In M. N. Wegman and T. W. Reps, editors, *POPL*, pages 365–377. ACM, 2000.
- [4] F. Ciocchetta and J. Hillston. Bio-PEPA: a Framework for the Modelling and Analysis of Biological Systems. *Theoretical Computer Science*, 410(33–34):3065–3084, 2009.
- [5] S. Clodong, U. Duhring, L. Kronk, A. Wilde, I. Axmann, H. Herzel, and M. Kollmann. Functioning and robustness of a bacterial circadian clock. *Mol. Syst. Biol.*, 3:90, 2007.
- [6] K. Eguchi, M. Yoda, T. P. Terada, and M. Sasai. Mechanism of robust circadian oscillation of KaiC phosphorylation in vitro. *Biophys. J.*, 95(4):1773–1784, Aug 2008.
- [7] E. Emberly and N. S. Wingreen. Hourglass model for a protein-based circadian oscillator. *Phys. Rev. Lett.*, 96(3):038303, Jan 2006.
- [8] M. Hucka, A. Finney, H. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray, A. Cornish-Bowden, A. Cuellar, S. Dronov, E. Gilles, M. Ginkel, V. Gor, I. Goryanin, W. Hedley, T. Hodgman, J.-H. Hofmeyr, P. Hunter, N. Juty, J. Kasberger, A. Kremling, J. Kummer, N. Le Novere, L. Loew, D. Lucio, P. Mendes, E. Minch, E. Mjolsness, Y. Nakayama, M. Nelson, P. Nielsen, T. Sakurada, J. Schaff, B. Shapiro, T. Shimizu, H. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. *Bioinformatics*, 19(4):524–531, 2003.
- [9] G. Kurosawa, K. Aihara, and Y. Iwasa. A model for the circadian rhythm of cyanobacteria that maintains oscillation without gene expression. *Biophys. J.*, 91(6):2015–2023, Sep 2006.
- [10] M. Kwiatkowski. *A Formal Computational Framework for the Study of Molecular Evolution*. PhD thesis, University of Edinburgh, 2010.
- [11] M. Kwiatkowski and I. Stark. The continuous π -calculus: A process algebra for biochemical modelling. In *Computational Methods in Systems Biology: Proceedings of the Sixth International Conference CMSB 2008*, number 5307 in Lecture Notes in Computer Science, pages 103–122. Springer-Verlag, 2008.
- [12] S. R. Mackey, S. S. Golden, and J. L. Ditty. The itty-bitty time machine genetics of the cyanobacterial circadian clock. *Adv. Genet.*, 74:13–53, 2011.
- [13] M. Nakajima, K. Imai, H. Ito, T. Nishiwaki, Y. Murayama, H. Iwasaki, T. Oyama, and T. Kondo. Reconstitution of circadian oscillation of cyanobacterial KaiC phosphorylation in vitro. *Science*, 308(5720):414–415, Apr 2005.

- [14] Systems Biology Software Infrastructure Home Page. www.sbsi.ed.ac.uk.
- [15] J. S. van Zon, D. K. Lubensky, P. R. H. Altena, and P. R. ten Wolde. An allosteric model of circadian KaiC phosphorylation. *Proceedings of the National Academy of Sciences*, 104(18):7420–7425, 2007.