Evolving Genetic Regulatory Networks for Systems Biology

Dominique Chu

Abstract— **Recently there has been significant interest in evolving genetic regulatory networks with a user-determined behaviour. It is unclear whether or not artificial evolution of biochemical networks can be of direct benefit for or biological relevance to Systems Biology. This article highlights some pitfalls when concluding from artificially evolved genetic regulatory networks to real networks. This article also gives a (brief) review of some previous attempts to evolve genetic regulatory networks with oscillatory behaviour; it also describes a new system to evolve networks and describes the networks that have been evolved. These networks seem to be very diverse sharing no apparent common motifs either with one another or with their real-life counterparts.**

I. Introduction

Systems Biology[1], [2], [3] aims to create a precise understanding of cellular processes by designing and simulating detailed models of biochemical processes (see for example [4], [5]. Most of this work includes models that are directly informed by extensive wet-lab experiments supplying interaction maps and (partially) parameters such as binding affinities, dissociation rates, transcription/translation efficiencies etc... This approach is fast producing valuable results for the biological community, allowing the detailed understanding and analysis of complex cellular processes.

An important part of Systems Biology is to create models that are as realistic as possible. One of the main difficulties to achieve this realism is the lack of quantitative information about parameters. The qualitative properties of biochemical systems will crucially depend on the various rates and affinities between the reacting pairs. Significant uncertainties in the knowledge about those parameters can result in even more significant uncertainties about the dynamical properties of the system under consideration.

There are a number of ways to overcome this problem. One relatively simple way is to fit the model to some known data. There are a number of statistical and software tools to do this [6], [7]; fitting models is under certain circumstances a very good solution, however, not useful if the available empirical data is unreliable or if there is simply very little data available. Also, model fitting will only be effective if there is not too much uncertainty about the parameters. Besides model fitting, another way to determine unknown parameters is to measure them. This method is (in principle) the best method but has the disadvantage that it is (i) relatively expensive (in terms of both money and time), (ii) possibly not feasible, and (iii) itself often plagued by large uncertainties.

While Systems Biology modelling is rapidly advancing all this shows that even very sophisticated modelling approaches are affected by a certain degree of uncertainty in particular about parameters. At present, there is no real solution to this and only gradual progress is made. Given this situation, complementary methods in Systems Biology might (in some circumstances) lead to additional insights into the processes to be modelled. In particular, various methods from computational intelligence can enhance the modelling process in Systems Biology. It would go beyond the scope of this article to review possible uses of artificial intelligence and machine learning in Systems Biology (but see for example [8]).

The focus of this contribution will be on the use of genetic algorithms[9] (and similar methods) to evolve genetic regulatory networks that display a user-defined target behavior; for example, this target behaviour might be a qualitative temporal variation pattern of the concentration of one (or several) proteins. The question is whether or not artificial evolution could give valuable insights into how real biochemical models evolve; what types of insights it can give and under what conditions.

A. Organisation of this article

This contribution focuses on evolving genetic regulatory networks with oscillating dynamics. There are no hard biological reasons why oscillations are chosen as the target behaviour (rather than any other type of dynamics). Instead, this choice was motivated by the fact that related previous work considered oscillations and because oscillations are conveniently recognised.

This article is organised as follows. The following section II will briefly review some previous work on oscillating genetic regulatory network. Section III will describe the software used to evolve genetic regulatory networks. Section IV will summarize some of the results obtained from simulations of the software. In particular, it will present some of the networks that have been evolved. Section V thereafter will discuss the results and compare them to the results obtained by previous authors; it will also argue that at present the connection between real and artificially evolved genetic regulatory networks is unclear and that more research is needed before artificial evolution of genetic regulatory networks can make a substantial contribution to concrete cases in Systems Biology. The article is then concluded by section VI.

II. Previous work

There have been several previous attempts to understand oscillatory dynamics in genetic regulatory networks[10], [11], [12]. A well know example is work by Elowitz and Leibler[13] who implemented a 3-gene network where the genes suppress one another in a cyclic fashion; this 3-gene

D. Chu is with the Computing Laboratory, University of Kent, CT2 7NF Canterbury, UK (phone: +44 1227827690; email: D.F.Chu@kent.ac.uk).

1-4244-1340-0/07$25.00 ©2007 IEEE

repressive system is known as the *repressilator*. Smith[14] theoretically predicted networks of genes that repress one another in a cyclic fashion to produce oscillations as long as the number of genes is odd. Elowitz and Leibler were the first to demonstrate this in a synthetic genetic regulatory network implemented in a live bacterium.

The idea of using artificial evolution to better understand real biological systems is not new. One early example is due to Mittenthal and co-workers[15] who attempted to understand whether or not the Krebs citric-acid metabolic cycle was optimal or whether it could be further optimized. Mittenthal and coworkers only modified the reaction rates but did not evolve the structure of the interaction network.

A more recent attempt to use evolutionary computation to find genetic regulatory networks that show a user-defined behaviour is by Francois and Hakim[16]; they describe an algorithm to evolve genetic regulatory networks with oscillatory dynamics. Francois and Hakim use a simple genetic algorithm to evolve both the structure (i.e. number of genes and their interactions) and the reaction rates of genetic regulatory networks. Since they essentially evolve sets of differential equations the standard genetic operator of crossover is difficult to apply in their case and they only use mutations as a source of evolutionary variation. They allow a great variety of different interactions between proteins and genes but also the formation of protein dimers. In their paper Francois and Hakim claim that their oscillator is structurally similar to the network responsible for the circadian rhythm. While this might well be the case, I think, and will argue, that any similarity is merely coincidental and is not a pointer to a deeper universality among oscillating genetic regulatory networks.

Another problem of the paper by Francois and Hakim is that they only consider deterministic systems. There are two broad approaches to modelling systems of interacting molecules. Deterministic approaches are based on the assumption that the concentration of a molecular species in a solution is essentially a continuous quantity; such deterministic systems are normally modelled as a set of differential equations and solved using standard numerical integration techniques. Deterministic approaches are convenient because they allow the modeler to use very fast techniques to calculate the time-course of the concentrations; however, many bio-chemical systems are not well modelled as deterministic systems because particle numbers can be relatively low. In these cases statistical fluctuations become relevant and systems need to be modelled using stochastic approaches that take into account that molecules exist in discrete quantities. There are specialized algorithms such as the Gillespie[17] algorithm or the Gibson-Bruck[18] algorithm to simulate stochastic systems. Particularly in prokaryotic genetic regulatory systems, stochastic effects can often not be ignored. Stochastic effects potentially have a significant impact on the evolution and evolvability of a networks as well. In the context of artificial evolution it is therefore essential to consider both types of models.

Another approach to evolve genetic regulatory networks is taken by Drennan and Beer[19] who describe a genetic algorithm to evolve the repressilator. Other than Francois and Hakim these author do not evolve sets of differential equations; instead the properties of their genetic regulatory networks are directly determined by a string composed of 4 different letters mimicking the role of real-world DNA. Transcription rates and the modification of these rates through activators and repressors is directly encoded by the sequence of the pseudo-DNA. The advantage of this approach is that it allows straightforward implementation of crossover thus making the genetic algorithm more efficient. On the downside, their approach strictly limits the space of possible models. In the system of Drennan and Beer there is no scope for protein-protein interactions and it is hard to see how it could be implemented without significantly changing the system. Also different to the approach by Francois and Hakim is that Drennan and Beer use stochastic models.

A similar scheme was used by Knabe and co-workers[20] to evolve oscillators that interact with the environment. Again, in this system proteins and their properties are explicitly coded for by a single genome string. As in the scheme of Drennan and Beer this system only allows for protein-gene interaction but does not include the formation of protein compounds.

A. *Objective of this contribution*

The approaches by Drennan/Beer and Knabe and coworkers are useful for the particular questions they ask, but they rely on a very schematic representation of real cellular interactions. On the other hand, Francois and Hakim has a more general approach to evolve genetic regulatory networks, but does not contain systematic investigation of the types of oscillators evolutionary processes yield.

The objective of this contribution is twofold. Firstly, on a technical side, I will introduce a system similar to the one by Francois and Hakim to evolve oscillating genetic regulatory networks. This system will be used in order to (i) extend on the results reported by Francois and Hakim in [16] where only very little information is given about the performance of their method and the range of results they obtained. (ii) Another (minor) objective is to compare these results to the regulatory networks evolved by Knabe and coworkers and Drennan and Beer. (iii) Finally, I will also provide a discussion on how evolving genetic regulatory networks can (and cannot) contribute to a better understanding of concrete systems in Systems Biology.

III. Description of the software

A simple algorithm is used to evolve a population of models of genetic regulatory networks. Every individual of the population is a model of a genetic regulatory network; each individual/model consists of a number of genes (labeled $X1, X2, \ldots$), corresponding RNA (labeled $R1, R2, \ldots$) and protein (or product as it will be called here; labeled $P1, P2, \ldots$). Each model is defined by a set of reactions and reaction rates that specify which compounds (gene-protein and protein protein compounds are allowed) are formed, the speed of transcription/translation and association/dissociation of compounds, as well as degradation rates and other necessary parameters of the model. In the initial population every model is randomly generated. Subsequently, at every time step every member of the population is assigned a fitness and ranked accordingly; then the least fit individual is replaced by a (mutated) copy of the fittest; similarly the second unfittest is replaced by a mutated copy of the next fittest and so on (the mechanics of mutations is described below). In this way, at each time-step up to half of the population is replaced by mutated variants of fitter solutions.

In all simulation runs considered here the fitness of a model is determined by how well the time course of the concentration of (the arbitrary) product $P1$ conforms to a given target behaviour. In this article the considered target behaviour is an oscillation given by $200\cos(\frac{x}{20})$. Since this article is not concerned with matching the model to a particular phase or amplitude during fitness evaluation the output of the model is not directly compared to the target function, but instead the respective autocorrelations are compared. One disadvantage of using the autocorrelation rather than an absolute time course of the concentration as benchmark is that the fitness will strongly depend on the sampling frequency of the simulator; if for example 10000 samples are taken for every time unit of the simulator, then a given oscillation will appear of much lower frequency than if only 10 samples are taken. This is not a very serious problem, however, because in the current context “real” time units are not of particular interest. The results are kept comparable by simply retaining the same sampling frequency for all experiments.

The simulator used during fitness evaluation of the models was the freely available “Dizzy” simulator[21]. Dizzy implements several stochastic and deterministic algorithms; throughout this article the Gibson-Bruck algorithm was used for stochastic simulations and the adaptive Runge-Kutta algorithm for deterministic solutions. In the case of stochastic simulations each gene X_i is represented as a single molecule; in the case of deterministic simulations genes are represented as being present in a concentration of 0.1. This choice is arbitrary beside the fact that the concentration of genes should be low.

Evolutionary computation methods of the sort described above require the entire space of allowed models to be specified (at least implicitly). In practice this is done by specifying a range of allowed reactions. Computational experiments of the sort described here normally require the experimenter to limit herself to a subset of all biologically and chemically possible interactions in order to maintain a manageable code and interpretable results. The overarching principle guiding the choice of interactions is of course biological plausibility, yet there is a certain degree of arbitrariness guiding the details of this choice, i.e. there are a number of different choices that would be equally plausible. The particular choice for the system introduced in this contribution is as follows:

- Every model has at least two genes.
- The genes are transcribed into RNA at specific rates and the RNA is translated into the product of the gene.
- Gene products can bind to genes; this alters the transcription rate of the gene. If the compound is transcribed at a higher than leak rate (i.e. the transcription rate of the unbound gene), then the respective product is an activator, otherwise it is a repressor. The leak rate can be zero.
- Genes can simultaneously bind to up to two products; the species of the first and the second product can be equal and can even be from the same gene. So, there could be models where gene $X1$ forms compounds $X1P1$ and $X1P1P1$ where $P1$ is the product of gene $X1$.
- Products can form compounds by homo- or heterodimerisation.
- Genes can bind to compounds of products; product compounds act as perfect inhibitors of gene expression, i.e. while a gene is bound to a compound it will not transcribe any RNA (not even at the leak rate).
- Products will decay at a specific rate, although this rate is allowed to be zero.
- Products can catalyze each others decay. So, for example, a molecule of $P1$ might catalyze the breakdown of a molecule of $P3$. Products might catalyze their own decay.

Every model will typically have only a subset of all possible features. That is, a given gene might, for example, only bind to one rather than two compounds (or none). Only some of the possible product-product compounds will form and only some of the genes will be inhibited by a product compound. The source of variation in the software are mutations (as described above). A mutation involves one of the following actions:

- Mutate any of the rates (i.e. translation, transcription, binding or dissociation rates).
- Add/remove gene. Whenever a gene is removed then all the reactions involving its product are removed as well. If a new gene is created then initially it will not be connected to the system through interactions. The maximal number of genes allowed in the system is determined by the user.
- Add/remove interactions (i.e. new gene-product or product-product compounds).
- For a chosen product P_i add/remove a breakdown catalyst. Note that the decay of a product can be catalyzed by at most one other product.

Mutations of reaction rates are mostly gradual, i.e. only small changes to rates are made during any mutation. However, the rate of breakdown of a product by another one can be turned on/off in a single mutation.

IV. Results

Evolving a genetic network is computationally very costly. Evaluating the fitness of one generation requires (in the current settings) about 50 simulations of different genetic network models. Finding low fitness candidates can take up to 20 thousand time steps. So, altogether this then requires of the order of 1 million of models to be evaluated. As it turns out, however, the success rate of the evolutionary algorithm is very low in the sense that most runs never converge to acceptable solutions. Hence it is necessary to perform many runs in order to find just a few successful ones. Depending on the parameters, there will be between 1 and 3 successful runs in every 30 runs. Determining the exact success rate would require significant computational resources and is beyond the scope of this contribution. However, there is significant scope for improving the performance of the algorithm; this is not the focus of this contribution and will be left for future research.

The overall picture emerging from all simulations is as follows:

- Stochastic oscillators seem to be relatively common in the (considered) space of all models; this observation is consistent with theoretical predictions that stochastic oscillators are more common than deterministic ones. Furthermore, these stochastic oscillators do not readily generalise to deterministic oscillators, that is the average behaviour taken over many runs will be non-oscillating (usually tending to a steady state average behaviour); again, this behavior is consistent with theoretical results.
- Stochastic oscillators are often of poor quality in the sense that both frequency and amplitude is variable over time. This effect is due to stochastic effects and therefore stronger in systems with low particle numbers. High quality stochastic oscillators are difficult to evolve in the present framework.
- The evolved stochastic systems tend to be small (often only 2 genes). Deterministic systems are normally larger and evolve to a size of about 4 or 5. By restricting the allowed maximal size of a model it is possible to evolve deterministic oscillators with only 2 genes; this suggests that the density of oscillators among 4-gene systems is higher than in 2 gene systems. No example of oscillators involving more than 5 genes was observed. It is likely that this is due to an inherent bias of the evolutionary algorithm for smaller systems, rather than a reflection of the density of oscillators in larger systems: Any newly added gene is initially unconnected to the rest of the system and will as such not contribute to any oscillations. It is therefore likely to be removed again.

A. *Description a specific runs*

In this section a few models that have been evolved will be described. This will give the reader an intuition of the type of result the evolutionary process leads to. It should be noted, however, that there is a great variance between the various evolved networks; in fact one of the results of the experiments presented here is that there appears to be no typical network pattern that consistently evolved throughout a large proportion of the experiments.

1) A deterministic example: In this subsection a specific (deterministic) oscillator will be described. It consists of 4 genes. The interactions of the system are depicted in figure IV-A.1. The most important features are:

- The genes $X1$ and $X2$ are controlled by the products of genes $X3$ and $X4$: Transcription of $X1$ and $X2$ is suppressed by $P3$ and by the product compound $P4P4$.
- Transcription of $X3$ is suppressed by $P4P4$ but activated by $P2$. Furthermore breakdown of $X3$ is catalyzed by $P4$.
- Transcription of $P4$ is strongly suppressed by $P1$ and the compound $P2P3$.

In this system the product $P4$ plays a pivotal role in that it suppresses every other gene and is itself suppressed by every other gene. This antagonistic relation between $P4$ and the other products is necessary for the oscillatory behavior yet by no means sufficient. The formation of compounds of products is, in this particular system, also necessary for the observed oscillations. This applies both to the product compounds that repress the genes $X1$-$X3$ but also other product compounds that do not, such as $P1P3$ that are not involved in suppressing other genes. This is illustrated in figure IV-A.1: As the association rate of $P1P3$ is reduced the amplitudes of the oscillations become smaller to the point where oscillations effectively cease to exist. While $P1P3$ does not itself act on any of the genes, it functions as a sink for the products $P1P3$ and as such introduces a delay into the system that is crucial to the oscillatory behavior.

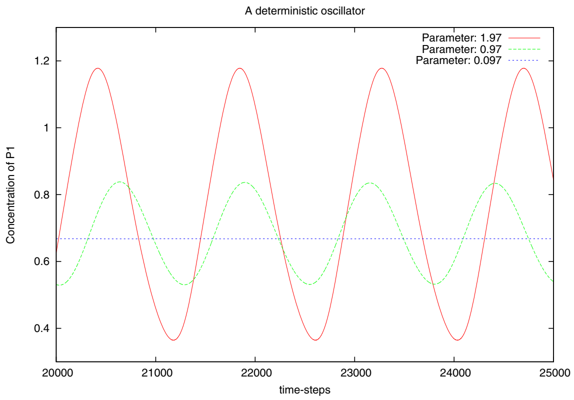

Fig. 1. This figure shows the oscillator for 3 different association rates of the $P1P3$ compound. Note that this compound is by itself not repressing any of the genes, yet nonetheless it appears to be crucial for the oscillatory behavior. The rate that was found by the evolutionary process is 0.9749. As the figure shows, increasing this value (to 1.9749) leads to a larger amplitude, whereas decreasing the value leads to smaller amplitude. In the present case the value was decreased to 0.09749 which led to the disapearance of the oscillations.

2) A stochastic example: The second example is a stochastic oscillator. This is an example of a system that evolved to a size of 4 genes, but, as it turns out two of its genes are not necessary for the oscillation. The main features of the system are as follows:

- The product $P1$ activates both itself and $X3$.

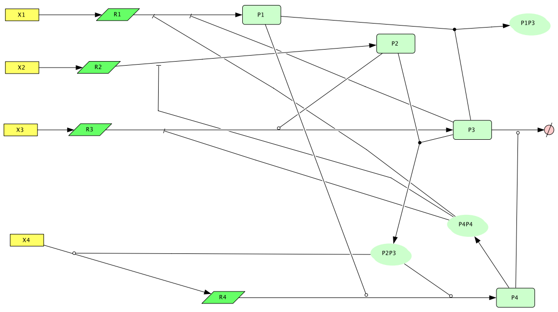

Fig. 2. A schematic representation of the most important interactions in the deterministic oscillator described in section IV-A.1. Note that this diagram is only a schematic representation of the interactions, but does not contain any reaction rates; in order to simplify the diagram the breakdown of proteins is also not represented unless it is catalyzed by another product. As such it does not provide enough information to repeat the simulation. The complete Dizzy source code of the model can be obtained from the author upon request.

- $X1$ is suppressed by $P1P2$.
- The breakdown of $P1$, $P4$ and $P2$ is catalyzed by $P3$.
- $P2$ and $P4$ suppress their own respective genes and $P4$ is also suppressed by $P1P4$.

A closer inspection of this system revealed that removing genes $X2$ and $X4$ will not significantly impact on the system's ability to oscillate. So, even though the network evolved to a size of 4, half of the genes are not required for the oscillatory behaviour. That only $X1$ and $X3$ are required for the oscillation is rather surprising and not immediately clear from pure visual inspection of the interaction map of the system.

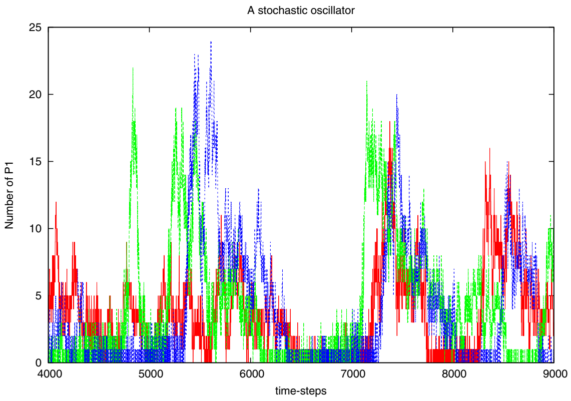

Fig. 3. This figure shows a 5000 time-step window of 3 runs of the same stochastic oscillator. Clearly, the 3 runs are not in phase but clearly show a periodic surge in the particle number of $P1$. Comparison with figure IV-A.1 show that the period is approximately equal, i.e. there are 3 maxima in a given window of 5000 time steps for all three runs.

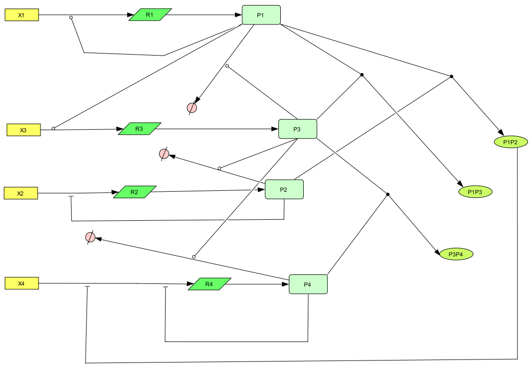

Fig. 4. A schematic representation of the most important interactions in the stochastic oscillator described in section IV-A.2.

B. A maximally simple (stochastic) oscillator

This oscillator is perhaps the simplest type of oscillator that can be evolved. It is so simple that the oscillatory dynamics can be intuitively understood; furthermore, it is clear that this oscillator will not work in a deterministic version but strictly relies on the fact that particles are discrete. The system consists of only 2 genes. In this particular example there was no intermediate RNA, so products were directly produced from genes without the delay of producing RNA. Also, in this test case there was only one activator/repressor allowed per gene.

The two products in this system repress one another's genes totally; so, as long as the compound $P1X2$ exists no new $P2$ will be produced; the same holds for $P2X1$. Furthermore, the product $P2$ catalyzes both its own breakdown and the breakdown of $P1$. Since $P2$ has a high affinity for $X1$ no new $P1$ will be produced as long as $P2$ is around. At the same time, $P2$ efficiently catalyzes its own breakdown; as a consequence there will normally only be one (or very few) molecule of $P2$ in the system (see below for an explanation). This molecule will decay at some point, but in the meantime it will catalyze the breakdown of $P1$ and hence drive down its number. Once it has decayed the amount of $P1$ will increase again, until, at some point another $P2$ is expressed. The resulting behaviour is that there are spikes of $P1$, whenever there is no $P2$ in the system. The evolved parameters work out such that these spikes occur with (approximately) the given frequency. Finally, there was another interaction that turned out not to be essential for the oscillatory behaviour: Formation of the $P1P2$ compound. This dimerisation acted as a temporary sink for $P1$ and $P2$ and resulted overall in smoother oscillations. Yet, other than in the case discussed above, here the formation of the product compounds was not essential for the oscillatory behaviour.

There are two features of this system that deserve attention: Firstly, the oscillating behaviour crucially depends on the fact that molecules are discrete entities. Since $P2$ efficiently catalyzes its own breakdown, in a deterministic model the concentration of $P2$ would quickly approach zero;

in the stochastic model on the other hand the breakdown of $P2$ by itself will stop when there is only a single molecule left in the system.

Secondly, this system is similar to the repressilator in that the dynamics of the system depends on the mutual suppression of the genes. Yet, again theoretical results predict that a 2-gene deterministic repressilator does not oscillate, whereas this stochastic system does.

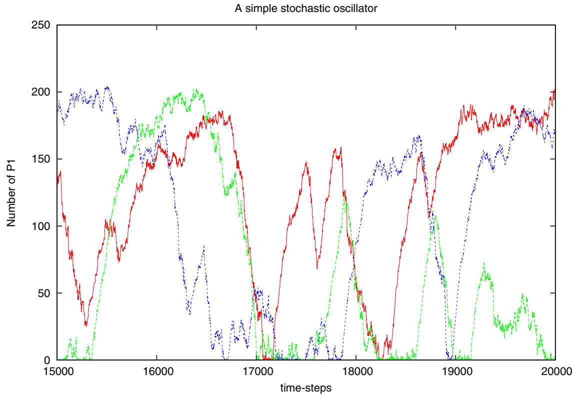

Fig. 5. This figure shows three runs of the simple oscillator described in section IV-B. From the figure it is clear that the quality of the oscillator is not extremely high, but there are about 3 periods for every 5000 time steps.

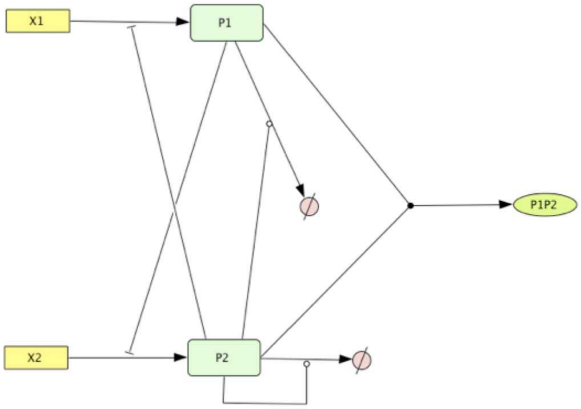

Fig. 6. A schematic representation of the maximally simple oscillator described in section IV-B.

V. Discussion

1) Relation between evolved and real networks: Artificial evolution certainly can play a role in computational Systems Biology. However, the relation between real and evolved networks is a subtle one and one needs to tread carefully when inferring conclusions about real networks based on a few artificially evolved genetic regulatory networks. Francois and Hakim claimed that their evolved oscillator is structurally similar to the networks responsible for the circadian rhythm. Is this a significant observation?

Simulations with the algorithm described here indicate that there are no typical networks that appear as a result of the simulation, at least in the sense that there are no obvious patterns that consistently show up across all oscillating networks that have been evolved. To be fair, the computational costs of evolving networks is currently too high to allow a proper statistical analysis of oscillating networks and a comparison with random networks. This must be left to future research. Yet, the absence of consistent circadian rhythm like genetic regulatory networks in the system described here indicates the following: Any similarity of Francois and Hakim's results to any real oscillator was either coincidental or due to some special design features of their system. If the former is the case, then this similarity is of course of no relevance; if the latter is the case, then Francois and Hakim need to provide an argument as to why their system is more correct than ours (or the one of other authors). Only then can the similarity be considered significant. Bearing in mind, though, that their system and ours are very similar in terms of the interactions they allow, the second possibility seems unlikely.

The conclusion from this is that any similarity between the networks evolved by Francois and Hakim and any real genetic regulatory network is likely to be coincidental rather than an indication of a deep universality of oscillators. It should also be noted at this point that there are no particular reasons to expect such a universality. Seen from the perspective of dynamical systems, oscillatory behaviour can be realized by a great variety of different interactions, and will not only depend on the structure of a genetic regulatory network but also on its parameters, i.e. the reaction/transcription/translation rates.

Apart from this degeneracy of oscillators with respect to network topologies there is another problem when trying to interpret results from artificial evolution: The range of allowed interactions that the evolutionary algorithms can use. In real biological systems there are many different types of molecular interactions: Steric effects or methylation can modify binding affinities/dissociation rates, the concentration of global regulators can change over time, there are various types of protein-protein interactions that can act back on the properties of individual proteins, RNA decay can be modified in various ways, and so on and so forth. In practice, the designer of artificial evolutionary algorithms will have to concentrate on a rather small subset of these interactions. This choice will by itself limit the kinds of networks that can evolve and how fast they will evolve. For example, in the case of the model of Drennan and Beer, the repressilator readily evolves, whereas in the present system, among all the evolved networks there was not one repressilator. Part of the explanation for this is of course that the system described in this contribution searches for oscillators in a different space of allowed networks than Drennan and Beer's system. In the space considered here repressilator type oscillators are presumably rare compared to non-repressilator oscillators. Seen in a more general context, the bias introduced by the design of the system becomes clear.

Finally, real genetic regulatory networks are likely to be subjected to a number of constraints that are hard to reproduce in artificial evolution. Firstly, the oscillator (or any other system) likely needed to evolve in a gradual fashion; it also needs to remain functional at every step the system in the sense that it is fit to perform at least some task; this restricts the possible pathways evolution can take. Secondly, the cell does not start from a random genome but has existing genetic and proteomic material at its disposition; again, this could limit the possible pathways of evolution. Thirdly, any regulatory network in a cell must avoid interactions that interfere with other regulatory networks in the cell thus potentially disrupting cell function.

All these constraints are not taken into account, either in the system presented here or any other system attempting to evolve genetic regulatory networks *in silico*. Whether or not this materially affects the plausibility of the models is a question that needs to be clarified before artificially evolved genetic regulatory networks can be of significant relevance in Systems Biology.

Taken all of this into account it seems unlikely that a similarity between Francois and Hakim's evolved oscillator indicates a deep universality of oscillators. None of the oscillators evolved in this contribution are particularly similar to Francois and Hakim's system, or indeed to one another. This indicates that the similarity reported by Francois and Hakim is likely to be coincidental.

2) Efficiency: Possibly relevant with respect to real genetic regulatory networks is another observation (that was not reported by Francois and Hakim (yet confirmed in a private communication)): It is very difficult for the evolutionary algorithm to find oscillators in the following sense: Firstly many runs fail to find any oscillator and those that do find one take a very long time. Notably, the approaches by Knabe and coworkers and Drennan and Beer seem unaffected by this problem. This difference in efficiency might be due to a number of reasons:

- Francois and Hakim's systems and the present system use a very simple algorithm without crossover. This might be partially responsible for the difference in efficiency. Experiments (not reported here) with crossover and with various variations of the selection regime did however not make a clear difference to the performance of the evolutionary algorithm; this, again, is not surprising because there is no natural way to define a crossover operator in the present scheme. Clearly more research needs to be done to resolve this question and/or improve the performance of the evolutionary system.
- Drennan and Beer's system operated on a much smaller search space than the system presented here. This might be a contributing factor for the difference in performance but is unlikely to be very important. In the system discussed here the search space can be restricted, but experiments (data not shown) indicate that this does not lead to significant improvements.
- Likely to be important is the fitness function itself. Drennan and Beer selected for a wide range of oscillating systems; Francois and Hakim on the other hand selected for a specific oscillating function with given phase and amplitude. In this contribution I use an in-between approach selecting for a given frequency based on the autocorrelation function. It is plausible to assume that a fitness function such as Drennan's leads to an easier fitness function with clearer pathways towards a good solution.

Amongst biological systems, oscillating behaviour is not particularly important, although there are some systems that do show oscillations (again an example is the circadian rhythm). While, as discussed above, artificially evolved genetic regulatory networks are not necessarily very instructive with respect to real networks, attempting to evolve artificial genetic networks can give some insights into the challenges that cells face themselves in their evolution. If the variation mechanisms cells employ are similar to the mechanisms described in this contribution (and one can plausibly argue so) then the cell will face similar difficulties in finding oscillators as the evolutionary algorithm does. If this is so, then there are restrictions regarding the likely evolutionary routes that lead to oscillators in cells: Oscillating systems are unlikely to have evolved in a direct response to adaptive pressures from the environment because finding oscillating behaviour (at least with a specific frequency) seems to be difficult (at least this is what the experiments by Francois and Hakim and the results shown here indicate). Instead oscillations will have evolved via (what evolutionary biologists call) a pre-adaptation route. According to this model oscillations have emerged as a coincidental by-product of a genetic regulatory network that did not require oscillatory behaviour in order to function. Drennan and Beer's work suggest that as long as one does not look for a specific frequency, oscillation readily emerge. Once the oscillator was more or less functional it would then be much easier for the cell to fine tune the frequency and network and use it for a specific purpose.

VI. Conclusion

Artificially evolving biochemical networks with a specific function is certainly of intrinsic interest. Does it also have valuable contributions to make for concrete modelling tasks in Systems Biology? Possibly, but at present it is not sufficiently developed for this task. At present there are a number of open problems regarding the methodology of evolving genetic regulatory (or other biochemical) networks. The main problem is a lack of understanding of the constraints that the evolution of real networks is subject to. When this and similar methodological problems are resolved, evolving artificial biochemical networks certainly could contribute to a better understanding of concrete real networks. Once developed the main benefit will possibly not be the estimation of unknown parameters, but rather broad explorations of evolutionary pathways leading to specific types of networks. For example statistical features of evolved networks could be compared to real networks, leading to a better insight into the conditions and constraints under which biochemical networks evolved.

Artificial evolution will probably not (at least not at first) be a method of choice to answer specific "how questions," i.e. to find out about the mechanistic details of particular biochemical networks. However, it has the potential to be a tool for biologists to ask "why questions" about the origin and current function of systems.

References

[1] H. Kitano, "Computational Systems Biology," *Nature*, vol. 420, pp. 206–210, 2002.

[2] ——, "Systems Biology: A Brief Overview," *Science*, vol. 295, pp. 1662–1664, 2002.

[3] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall, 2006.

[4] H. Sauro, A. Uhrmacher, D. Harel, M. Hucka, M. Kwiatkowska, P. Mendes, C. Shaffer, L. Strömback, and J. Tyson, "Challenges for modeling and simulation methods in systems biology," in *Proc. 2006 Winter Simulation Conference*, L. Perrone, F. Wieland, J. Liu, B. Lawson, D. Nicol, and R. Fujimoto, Eds., 2006, to appear.

[5] M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, J. Heath, and E. Gaffney, "Simulation and verification for computational modelling of signalling pathways," in *Proc. Winter Simulation Conference*, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, Eds. Omnipress, 2006, pp. 1666–1675.

[6] Z. Zi and E. Klipp, "Sbml-pet: a systems biology markup language-based parameter estimation tool," *Bioinformatics*, vol. 22, no. 21, pp. 2704–2705, November 2006.

[7] M. Vass, C. Shaffer, N. Ramakrishnan, L. Watson, and J. Tyson, "The jigcell model builder: A spreadsheet interface for creating biochemical reaction network models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 03, no. 2, pp. 155–164, 2006.

[8] F. Dubitzky, Werner; Azuaje, Ed., *Artificial Intelligence Methods and Tools for Systems Biology*. Springer, 2005.

[9] M. Mitchell, *An Introduction to Genetic Algorithms*, 3rd ed. A Bradford Book, MIT Press, 1997.

[10] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, S. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, pp. 824–827, 2002.

[11] ——, "Network motifs in the transcriptional regulation network of Escherichia coli." *Nature Genetics*, vol. 31, pp. 64–68, 2002.

[12] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Scheffer, and U. Alon, "Superfamilies of evolved and designed networks," *Science*, vol. 303, pp. 1538–1542, 2004.

[13] M. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, pp. 335–3338, 2000.

[14] H. Smith, "Oscillations and multiple steady stated in a cyclic gene model with repression," *Journal of Mathematical Biology*, vol. 25, pp. 169–190, 1987.

[15] J. Mittenthal, B. Clarke, T. Waddell, and G. Fawcett, "A New Method for Assembling Metabolic Networks with Application to the Krebs Citric Acid Cycle," *Journal of Theoretical Biology*, vol. 208, pp. 361–382, 2001.

[16] P. Francois and V. Hakim, "Design of genetic networks with specified functions by evolution in silico," *Proceedings of the National Academy of Science USA*, vol. 101, no. 2, pp. 580–585, 2004. [Online]. Available: http://www.pnas.org/cgi/content/abstract/101/2/580

[17] D. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, pp. 2340–2361, 1972.

[18] M. Gibson and J. Bruck, "An efficient algorithm for generating trajectories of stochastic gene regulation reactions," California Institute of Technology, Tech. Rep. CaltechPARADISE:1998.ETR026, 1998. [Online]. Available: http://resolver.caltech.edu/CaltechPARADISE:1998.ETR026

[19] B. Drennan and R. Beer, "Evolution of repressilators using a biologically-motivated model of gene expression," in *Artificial Life X : Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, Eds., International Society for Artificial Life. The MIT Press (Bradford Books), August 2006, pp. 22–27.

[20] J. Knabe, C. Nehaniv, M. Schilstra, and T. Quick, "Evolving biological clocks using genetic regulatory networks," in *Proceedings of the Artificial Life X Conference (Alife 10)*, L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, Eds. MIT Press, 2006, pp. 15–21. [Online]. Available: http://homepages.feis.herts.ac.uk/ kj6an/GRNclocks

[21] S. Ramsey, D. Orrell, and H. Bolouri, "Dizzy: stochastic simulation of large-scale genetic regulatory networks," *Journal of Bioinformatics and Computational Biology*, vol. 3, pp. 415–436, 2005.