

# OSCILLATORY SYNTHETIC BIOLOGICAL SYSTEM CONSTRUCTION USING INTERACTIVE EVOLUTIONARY COMPUTATIONS

<sup>1</sup>Quang Huy Dinh, <sup>2</sup>Nasimul Noman and <sup>1</sup>Hitoshi Iba

<sup>1</sup>Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

<sup>2</sup> School of Electrical Engineering and Computer Science, The University of Newcastle, NSW, Australia

Received 2014-08-18; Revised 2014-10-20; Accepted 2014-10-29

## ABSTRACT

In this study, we propose an Interactive Evolutionary Computation system to design gene networks for synthetic biology, of which the objective is to construct new organisms through the artificial synthesis of gene networks. Wet laboratories typically design gene networks through trial and error methods, which makes the construction of complex networks difficult. As a solution, automation can significantly improve the efficiency of identification of useful networks. However, guidance and feedback from biologists are required for the automation of this process, primarily because models that are sufficiently accurate at precisely simulating network behavior do not exist yet. Therefore, we have implemented an Interactive Evolutionary Computation system that allows the automation of the gene network design task while incorporating expert knowledge. Our results show that the system can efficiently design oscillating networks and can successfully identify complex networks, which are difficult to generate manually.

**Keywords:** Synthetic Biology, Evolutionary Computation, Interactive, Oscillation

## 1. INTRODUCTION

Synthetic biology (Andrianantoandro *et al.*, 2006) is a new research field that is rapidly gaining attention not only from biologists but also from computer scientists and information technologists. Research in synthetic biology is based on “construction, utilization and analysis” rather than the conventional “observation and analysis” and its objective is to construct new organisms through the artificial construction of gene networks. Conventional biology is typically based on an analytical approach, in which individual organisms are observed and analyzed, whereas synthetic biology takes a bottom-up approach, in which accumulated, analyzed insight and knowledge about biological organisms are used to artificially construct new ecosystems. The artificial synthesis of a new living system from individual

components is expected to result in the development of useful new materials with potential medical applications.

One role of information science in synthetic biology is the design, simulation and analysis of artificial genetic circuits and metabolic pathways. The genetic circuits are designed by combining identified interacting biomolecules; however, there are an infinite number of combinations and identifying a circuit that behaves as intended is very difficult. Although a number of relatively simple genetic modules have been proposed, such as switches (Gardner *et al.*, 2000), oscillation circuits (Elowitz and Leibler, 1999) and logic circuits (Anderson *et al.*, 2007) - these are small modules that have been discovered through trial and error - a limitation imposed by the fact that the mathematical models used in computational design can reproduce only part of the actual phenomenon. When many modules are

**Corresponding Author:** Quang Huy Dinh, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan

brought together to create large-scale dynamic genetic circuits, inclusion of parameters that have not been precisely measured for accurate simulation and an incomplete understanding of many of the mechanisms in real organisms can result in unexpected behavior.

Guidance from biologists, who have an in depth knowledge of actual reaction systems, is crucial to designing artificial genetic circuits that contain such problems. The use of Interactive Evolutionary Computation (IEC) is proposed here as a feedback method to accurately reflect biological properties that are difficult to quantify or to predict based on experience. IEC directly applies qualitative evaluation by the user for optimization based on implicit indices such as personal preferences and emotions that are difficult to explicitly express on a computer; it is used in various situations such as creativity support systems in the graphic arts (Unemi, 2000) and musical composition (Ando and Iba, 2007), as well as for optimization of engineering problems including speech processing (Watanabe and Takagi, 1995) and data mining (Terano and Inada, 2003). The goal of this study is to propose a method that uses IEC to efficiently design practical artificial genetic circuits.

As a case study, here we present the interactive evolution of various types of oscillatory circuits. The rest of the paper is organized as follows. Section 2 describes the background of the research. Section 3 describes the materials and methods. Experiment result is given in section 4 and conclusion is given in section 5.

## 2. BACKGROUND

### 2.1. Synthetic Biology

Synthetic biology is a rapidly developing discipline involved in the construction of biological systems that do not exist in nature by combining biomolecules such as Deoxyribonucleic Acid (DNA), genes and proteins. The synthetic biology community defines synthetic biology as follows. “Synthetic biology refers to both:

- The design and construction of new biological parts, devices and systems and
- The re-design of existing, natural biological systems for useful purposes

The two definitions point to the construction of biological systems that do not exist in nature and the objectives can be categorized as either “understanding living phenomena through reconstruction” and

“construction of useful biological systems”. Approximately 10 years have passed since synthetic biology began garnering attention; however, most of the research reported to date concerns processes that have not resulted in development of novel biological systems (Dinh *et al.*, 2014).

Of the two main objectives described above, the “creation of useful biological systems” can be understood literally as enriching people’s lives by constructing new organisms that make a contribution to humankind. Examples of past research include gene networks that detect DNA damage (Kobayashi *et al.*, 2004) and bacteria that preferentially infiltrate cancer (Anderson *et al.*, 2006).

Concerning the second objective, “understanding living phenomena through reconstruction”, the process of constructing new organisms is used to identify novel properties of organisms, draw comparisons with existing knowledge gathered using conventional biology, and contribute to the development of biology. Using this process, it is possible to detect properties that cannot be identified by simple observation; therefore, artificial construction of organisms has the potential to make novel contributions to several fields.

Collaboration between wet laboratories that conduct experiments by actually building gene networks using organisms such as the bacterium *Escherichia coli* (*E. coli*), and dry laboratories that design gene networks through computer simulation, is common in synthetic biology. In wet laboratories, one method for artificially constructing organisms is the use of circular DNA plasmids of *E. coli* to build artificial gene networks and to investigate protein functions. In other words, protein concentrations are varied with time to generate artificial organisms that act as switches or oscillators. Here, computer simulations are used to design gene networks that can perform the desired functions.

Synthetic biology is also closely related to electronic engineering, information engineering and chemistry and one of the objectives is to use biomolecules and biological systems in an engineering context. From an electronic engineering perspective, computers are built using standardized fundamental components to form devices and electronic circuits; these functionally differentiated modules can be combined to form a computer. In synthetic biology, biomolecules, which are regarded as fundamental components, are combined to form modules, which are reaction system networks that behave as intended and these modules are

combined to form useful biological systems. This characteristic of combining minute biomolecules connects synthetic biology to “nanotechnology”, in particular to nanobio technology.

“Construction of useful biological systems”, which is one of the goals of synthetic biology, includes research to add new functionalities to organisms that already exist in nature. The introduction of new functionalities and features to organisms by genetic modification has been attempted using “genetic engineering” technology since the 1970 s and some outcomes are already on the market, for example the genetically modified organism GloFish. The greatest differences between synthetic biology and genetic engineering are in the type of genes handled and in operability. Genetic engineering usually focuses on mutation of a single gene, whereas synthetic biology typically handles entire reaction systems that consist of many genes. Furthermore, in synthetic biology, attempts are made to standardize biomolecules to simplify joining of genes or to mutate genes to allow them to function in a targeted organism (Knight, 2003).

## 2.2. Interactive Evolutionary Computation

Interactive evolutionary computation (IEC) is a method that enables tasks such as musical composition or graphic arts, which were previously considered impossible for computers to perform, through interactions between the user and the computer. Evolutionary computation is an optimization method to achieve target specifications and performance by evolving systems such as organisms, which can efficiently find the global optimal solution by effectively utilizing the diversity of populations and neighborhood queries. The system configuration and design data are considered to be the genes (chromosomes) of organisms and the following steps are taken. (1) Systems corresponding to individual organisms are generated based on the chromosomes. (2) These organisms are evaluated based on designated criteria. (3) Crossovers between highly evaluated individual chromosomes of organisms and mutations are performed to generate the next generation of these chromosomes. Complex systems are designed and optimized by iterating this procedure. Standard or non-interactive Evolutionary Computation (EC) uses the evaluation functions in Step (2), which take well-defined values that can be processed by a computer; therefore, criteria such as human preferences and emotions cannot be completely modeled. In contrast, in IEC users iteratively evaluate each individual and generate new individuals based on highly

evaluated individuals. In other words, the evaluation step of EC is replaced by human involvement and thus optimization problems that were considered too difficult for computers to handle can be solved because no evaluation function is required. When humans attempt to search for a solution, the search often becomes local and cannot escape from a particular pattern. However, IEC uses evolutionary computation to propose solutions that users cannot conceptualize. Therefore, IEC is widely applied to creative support systems in the arts, as well as to simple optimization problems (Ando and Iba, 2007).

Because the user performs all the evaluations in IEC, one drawback is limitation by user fatigue. The number of individuals that can be presented simultaneously is also limited by the screen size (for images) and the memory capacity of the user (for music and videos). Furthermore, user fatigue limits the number of generations to be searched to approximately 10 or 20. Measures to counter these disadvantages include improvement of the user interface (Ohsaki *et al.*, 1998) and integration with EC (Ono and Nakayama, 2012).

## 3. MATERIALS AND METHODS

### 3.1. Reaction Model

One of the objectives of synthetic biology is to construct organisms that have particular desired functionalities. However, because no established method for reasonably constructing controllable artificial biomolecular networks exists, research and development is based on trial and error. Research to simulate the behavior of biomolecular networks by modeling known fundamental reactions based on DNA engineering is being undertaken to resolve this issue. One example is the following experiment based on the reaction network model by (Montagne *et al.*, 2011).

Most of the reaction networks in biomolecules combine three fundamental reactions: Activation, inhibition and destruction. Montagne *et al.* (2011) model uses oligomers for activation, 3'-mismatched oligomers for inhibition and RecJ exonuclease for destruction. The input, output and inhibitor are all short oligomers; therefore, reaction networks can be configured with arbitrary combinations, as shown in **Fig. 1**. Biological reaction networks were searched based on this model that acts as an oscillator circuit and the time change of each molecule was observed. The circuit is an oligomer oscillator and therefore Montagne *et al.* (2011) named it an “obligator”.

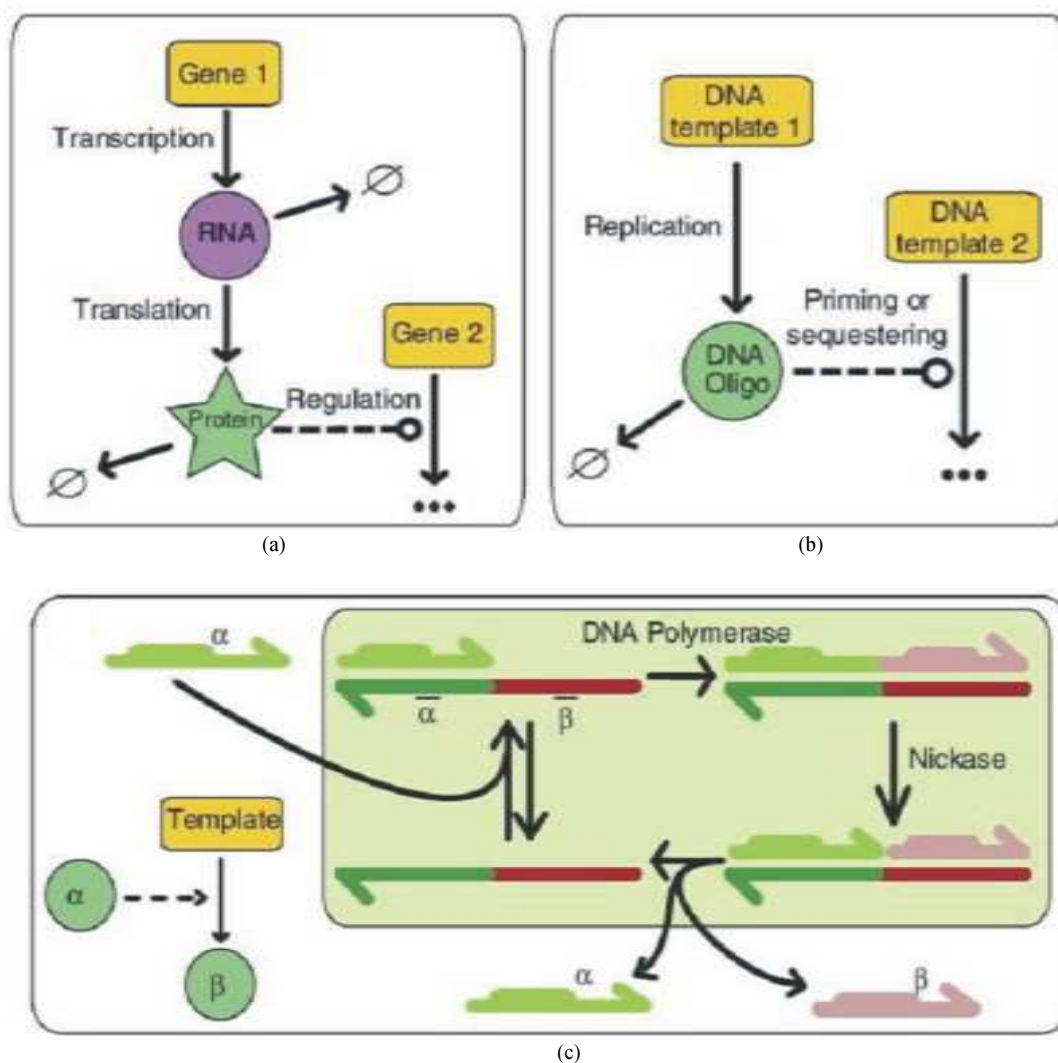
### 3.2. Gene Representation for the Reaction Network

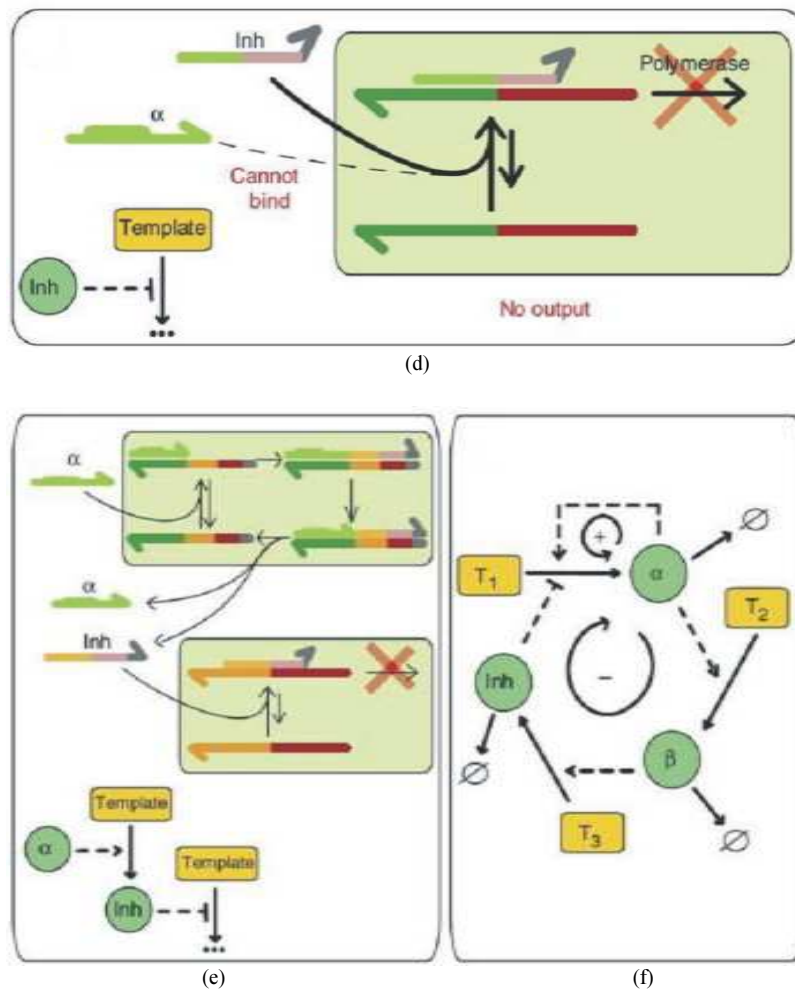
Figure 2 shows the gene representations used in our system. The left side of this figure shows an example of a reaction network and the two matrices on the right represent the connections in the reaction network. The blue matrix is for activation and each element contains 1 or 0. If oligomer  $i$  activates oligomer  $j$ , the element in the  $i$ th row and  $j$ th column becomes 1 and if there is no activation, it becomes 0.

The red matrix is for inhibition and each element contains a 0 or the name of an oligomer. The element 0 indicates that there is no inhibition of the corresponding activation. If oligomer  $k$  inhibits the activation from oligomer  $i$  to oligomer  $j$ , the element

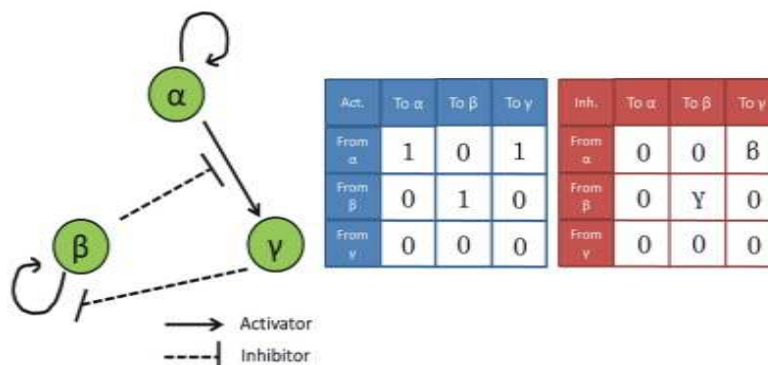
in the  $i$ th row and  $j$ th column becomes  $k$ . However, if there is no activation, the corresponding element must be 0, which is a constraint for inhibition matrices.

Each child in the next generation is generated by two parents selected by tournament selection. At the beginning of the process of generating a child, the system executes a crossover of the activation matrices of the parents. In the crossover operation, each element in the child's activation matrix inherits the corresponding elements from the matrix of one of the two parents. In the next step, the system executes a mutation, which randomly alters a part of an element in the child's matrix. After generating the activation matrix, the system generates the inhibition matrix randomly, considering the constraints.





**Fig. 1.** Model of a reaction network (Montagne *et al.*, 2011). (A) Schematic of the reaction from adjustment of genes. (B) Model in which genes are replaced by single-strand DNA. (C) Activation reaction. (D) Inhibition reaction. (E) Cascading reaction. (F) Oscillator model with a positive-feedback loop (+) and negative-feedback loop (-)



**Fig. 2.** Gene representation of the reaction network

### 3.3. Evaluation Score

In our search system, the reaction network search is performed with the initial concentration and rate constant fixed to the standard values in the reaction model (Montagne *et al.*, 2011). Each individual is evaluated using a linear weighted average of the qualitative evaluation score given by the user and a quantitative computational evaluation score, as follows Equation 1:

$$F(x) = w.P(x) + (1 - w).Q'(x) \quad (1)$$

Here,  $P(x)$  is the evaluation score given by the user and  $Q'(x)$  is the quantitative evaluation score. Users evaluate each displayed graph as “good”, “default”, or “bad” in IEC. The evaluation score is determined by  $P(x) = 1$  for “good” individuals and  $P(x) = 0$  for “default” individuals and “bad” individuals are never chosen as parents when generating the next generation. We used 0.5 for the parameter  $w$  in this study.

However,  $Q(x)$  was defined as follows, to become larger when the number and amplitude of oscillations increase, because the objective of this experiment was to search for an oscillator network Equation 2 and 3:

$$Q(x) = \sum \{(A_{max} - A_{min}) / (Maxvalue - Minvalue)\} . n^2 \quad (2)$$

Here,  $A_{max}$  represents the local maximum,  $A_{min}$  represents the corresponding nearest local minimum,  $n$  is the number of oscillations and  $Maxvalue$  and  $MinValue$  represent the maximum and minimum values in the system. This quantity is calculated for each existing oligomer system and then summed.  $Q'(x)$  is defined as shown below for normalization before taking a linear sum with  $P(x)$ :

$$Q'(x) = \begin{cases} 1, & (Q(x) \geq 300) \\ \frac{Q(x)}{300}, & (Q(x) < 300) \end{cases} \quad (3)$$

$Q(x)$  in Equation 2 usually does not have a maximum value; however, the above normalization was performed because sufficient oscillation was observed in the resulting time-series graph when  $Q(x)$  was greater than 300 in this experiment.

We used a population size of 30 for this experiment. To minimize user fatigue, the system first calculates

quantitative scores for the whole population and then asks the user to evaluate the top 6 individuals.

## 4. RESULTS

### 4.1. Oscillator with 2 Oligomers

We conducted an experiment to design an oscillator with 2 oligomers. The system obtained an existing oligator (**Fig. 3**) proposed by Montagne *et al.* (2011). For visualization of the network in the system, intermediary nodes are used to express inhibiting connections. For the example in **Fig. 3**, the inhibition (from oligomer  $\beta$  to the activation from oligomer  $\alpha$  to  $\alpha$ ) is expressed with intermediary nodes.

The time of calculation to find this network, including human evaluation, is approximately 30 min for a PC with an Intel Core i7 Q720 CPU and 4 GB of memory. This result shows that the system can design a simple module for oscillation in a reasonable calculation time. Unknown networks can also be found in the same experiment. One example is shown in **Fig. 4**. Networks having both a positive-feedback loop and a negative-feedback loop are known to be robust for the noise in the initial concentration and rate constant. This network includes both types of feedback loop.

### 4.2. Utilization of the Expert's Experience and Knowledge

If the number of oligomers increases, the number of possible networks increases exponentially and in this case, manual design of networks through trial and error is difficult and inefficient. Therefore, the contribution of automatic network design increases when seeking networks containing many oligomers. However, as already stated, because of user fatigue, the system should reduce the number of evaluations as much as possible. To identify useful networks more efficiently, it will be helpful to evaluate not just phenotypes but also the entire network configuration. Although it is difficult to convert expert experience and knowledge into data, reflecting judgments such as “this section may be useful” or “this part of the circuit has a relatively robust configuration”, in IEC should result in faster and easier convergence into more realistic networks. For this purpose, we added the following two functions to the system.

- Manual fixation for a part of the elements in the activation and inhibition matrices
- Switching from evolutionary computation to a local search

A local search is achieved by selecting a single individual and populating the next generation with mutations of the selected individual only. Using these functions, a user can directly guide the search scheme. Manual fixation of the gene has been shown to be effective in Japanese anagram sentence generation (Ono and Nakayama, 2012).

### 4.3. Oscillator with 3 Oligomers

We compared our IEC system with an EC that only uses quantitative scores, mentioned in the previous section. In this experiment, the search performances of the two systems for design of an oligator with 3 oligomers were compared. Population size and other settings for the experiment were the same as in the experiment with 2 oligomers. **Figure 5** shows the fitness transition after 1 run with IEC and 3 runs with EC. The vertical axis indicates the quantitative score before normalization and the horizontal axis indicates the number of generations, up to 10. The pink line shows the results of IEC and the other lines show the results of EC.

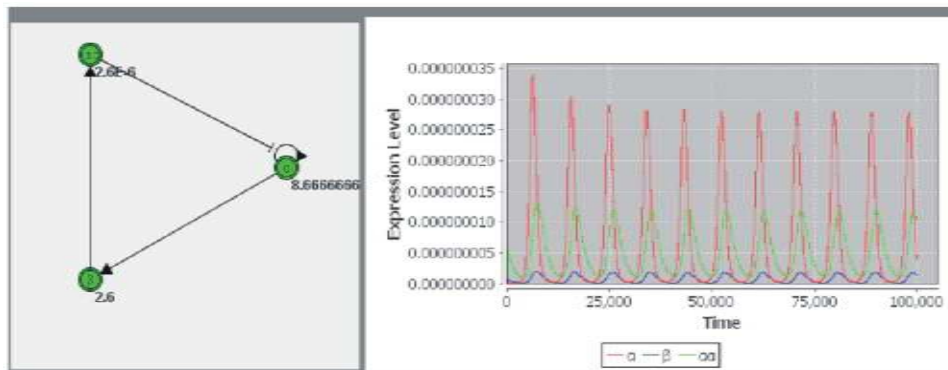
The best networks obtained from the 4 total runs are shown in **Fig. 6 to 9**. Examining the fitness transition shows that IEC found a potentially useful network more rapidly than EC. A human user can judge if the network has the potential for oscillation, whereas the quantitative score used in EC can only judge whether or not the network already contains oscillation. This property affects search performance at the beginning of the search. Examining the networks obtained in the experiment, it can be observed that although the networks obtained by EC actually contain oscillation, oligomer  $\gamma$  contains no oscillation in all 3 cases. This problem originates in the definition of the quantitative score, which uses the total value of each oligomer's oscillation, with the result that these networks are assigned high scores even though they do not possess the desired qualities. However, in an IEC system, users can avoid problems caused by the definition of the quantitative score by scoring unintended networks as "bad." In **Fig. 5**, a reduction of the quantitative score occurs in generation 5, using IEC, indicating user avoidance of unintended networks. IEC successfully

identified a network in which all 3 oligomers were oscillating and that ultimately obtained a higher quantitative score than the other identified networks.

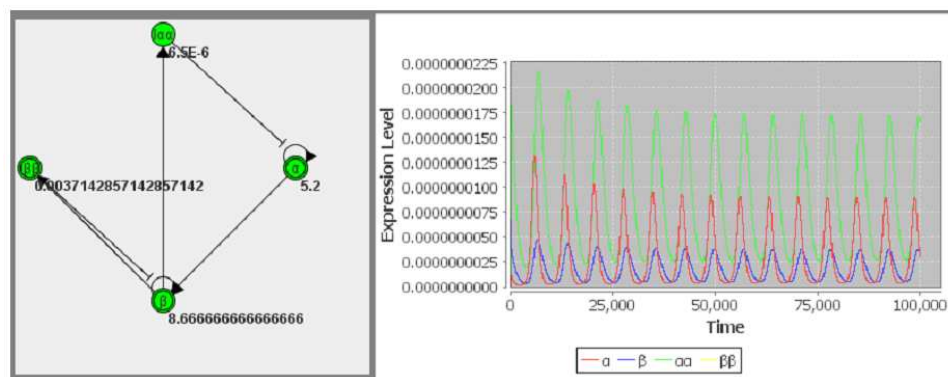
### 4.4. Complex Oscillators

In the previous section, the efficiency of IEC for the design of oligomers was demonstrated from the view point of search performance. Next, we attempted to design complex oscillators that had not previously been identified by biologists. Data for oscillators that display complex and interesting behavior is useful for the development of synthetic biology. "Complex" or "interesting" behavior cannot easily be defined, making our method using human evaluation particularly useful. Chaotic oscillation is an "interesting" behavior. **Figure 10** shows an example of chaotic oscillation called a Duffing oscillator, which is generated when an iron pendulum and a magnet are on a surface that is oscillated by external forces. This type of chaotic oscillation is known to exist in living organisms. For example, our brain waves contain chaotic elements.

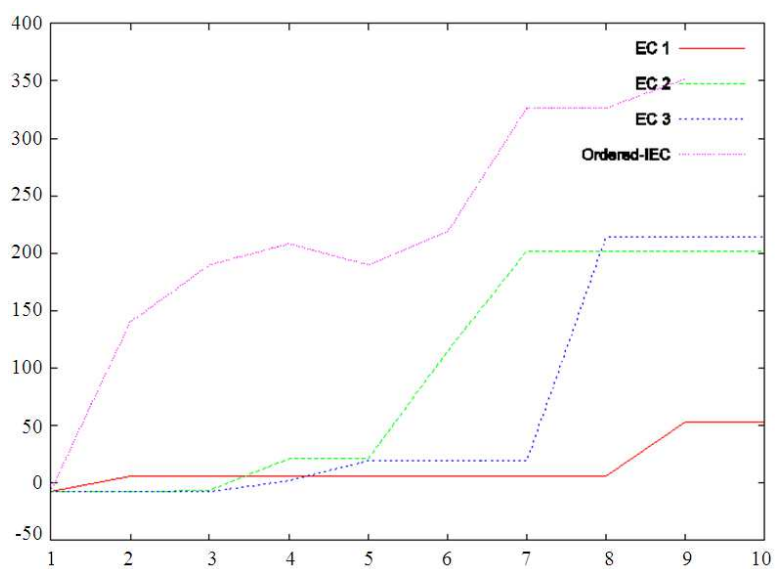
For this problem, we changed the strategy of the search system. First the system generates simple oscillators using EC, as described in the previous section. Since we used a population size of 20, the system prepared 20 oscillators. Then, using the 20 oscillators as the initial population, the system evolved them into complex networks using IEC. Quantitative scores cannot be applied to this phase of IEC because the purpose has been shifted to generate "complex" or "interesting" networks. However, without quantitative scores, the instability of human evaluation becomes a significant problem, because using this method, the only dependable information available for improving networks is human evaluation. A human evaluation score can differ, even for the same network, after evaluating other networks. Therefore, we implemented another interface and algorithm which use relative comparisons between individuals. Takagi and Pallez (2009) noted that, when using IEC, information obtained from relative comparisons is more stable than that obtained from absolute comparisons. **Figure 11** shows the GUI of an IEC system that employs relative comparisons, in which the user selects one network from three candidates. This GUI also performs the function of user guidance, mentioned in the section 4.2. In this system, the user selects one network from three, using an A, B, or C button located to the right of each network. The panel to the right of the A button is for user guidance.



**Fig. 3.** Well-known oscillating network



**Fig. 4.** Unknown oscillating network



**Fig. 5.** Fitness transition for the design of oscillator with 3 oligomers

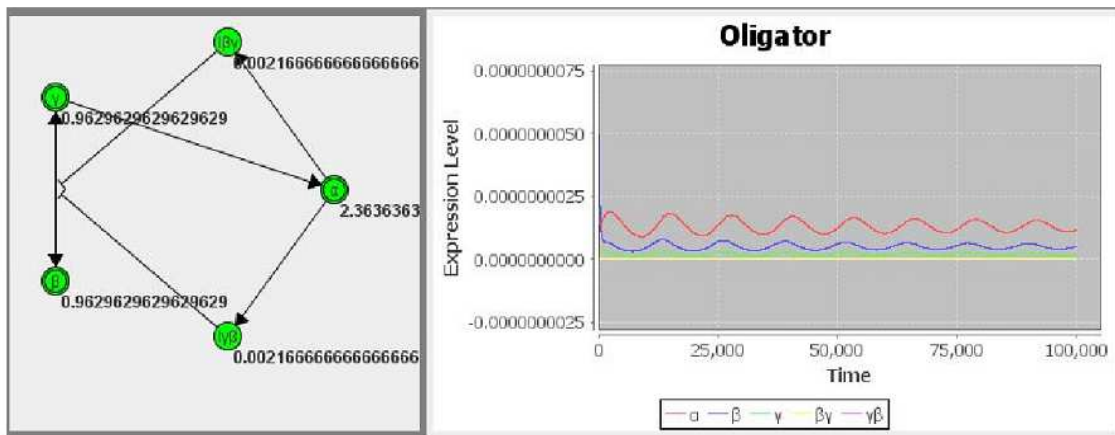


Fig. 6. Best network obtained by EC 1

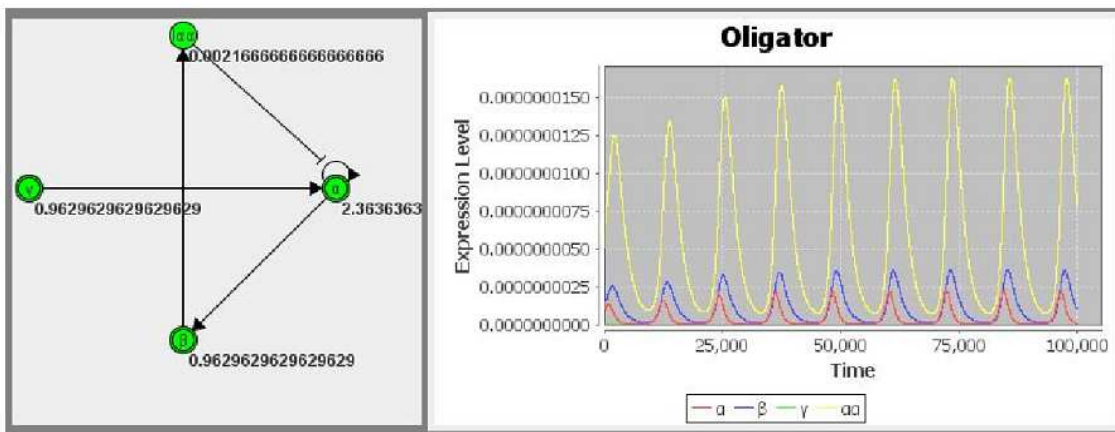


Fig. 7. Best network obtained by EC 2

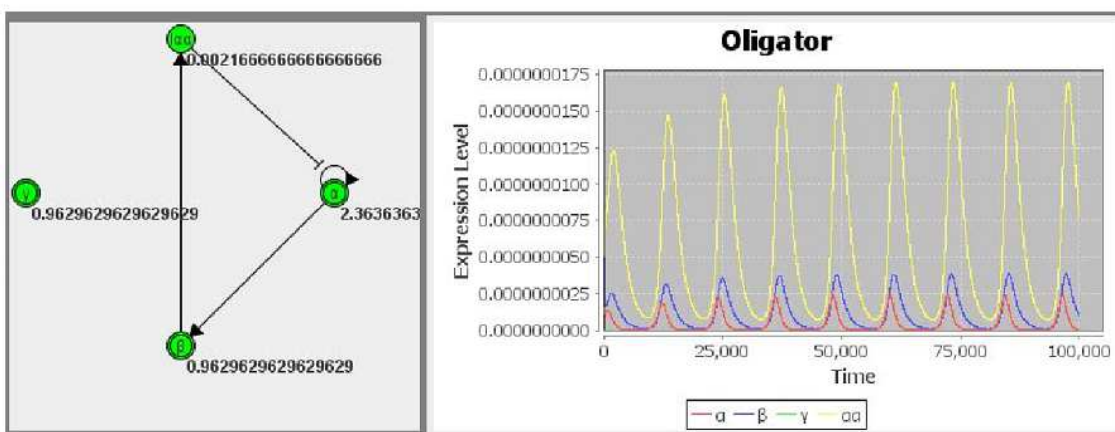
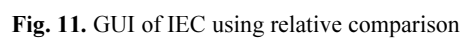
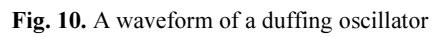
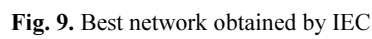


Fig. 8. Best network obtained by EC 3



The three candidate networks are generated by the algorithm shown in **Fig. 12**. For each individual, the system randomly chooses another individual as a parent and generates a child via crossover and mutation. Repeating this process, the system generates 2 children and asks the user which of the networks, of the original and the 2 children, is best. Only the selected network remains in the population and the others are discarded.

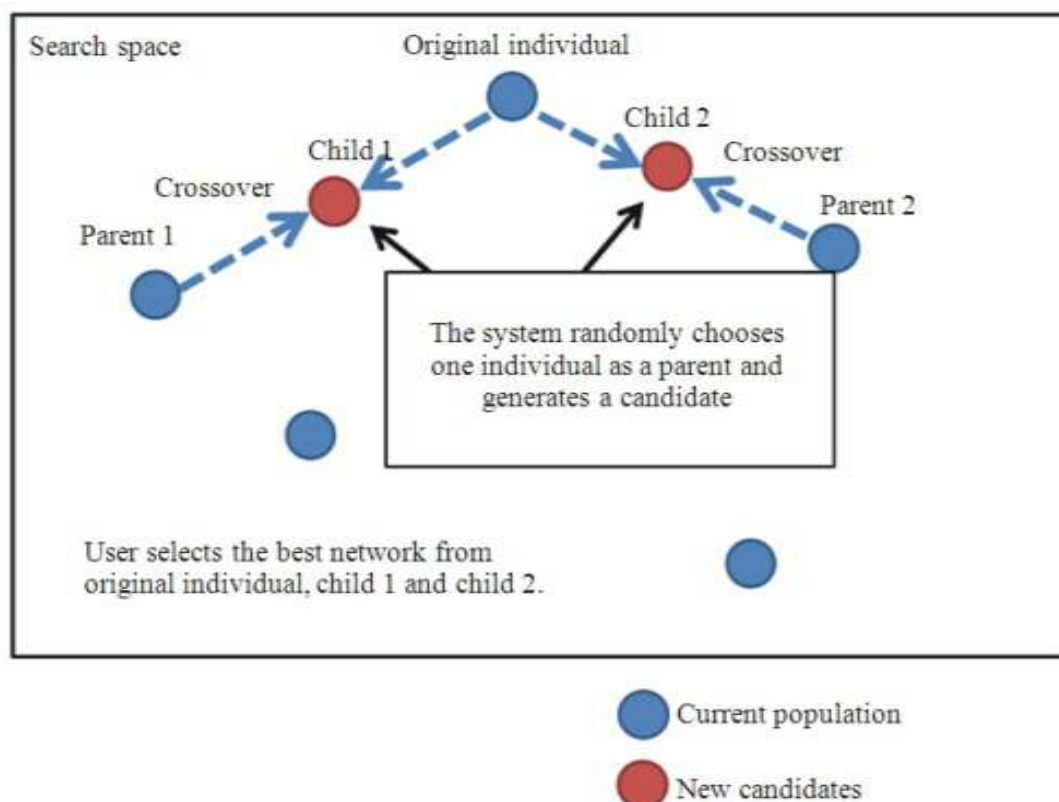
The system that included the GUI and the algorithm that ameliorates the instability of human evaluation obtained several networks that displayed complex behavior. We showed these networks to biologists and asked for their comments. The network shown in **Fig. 13** contains an oscillation which gradually increases in amplitude. Biologists commented that this network contains a number of structures which are important for oscillation and this makes the behavior of the network complex. Chaotic oscillation also requires these multiple oscillating structures.

In **Fig. 14**, it can be seen that the waveform of the oscillation is distorted. Biologists commented that this

study has the structure of a “double inhibitor” for the oligomers  $\alpha$ ,  $\beta$  and  $\gamma$ . One example of a simple “double inhibitor” structure is shown in **Fig. 15**. A “double inhibitor” represents a special relationship between oligomers, in which activation of each oligomer is inhibited by the other oligomer. For example, in **Fig. 15**, the oligomer  $\alpha$  activates itself and this activation is inhibited by the oligomer  $\beta$ . Conversely, self-activation of the oligomer  $\beta$  is inhibited by the oligomer  $\alpha$ .

The next network shown in **Fig. 16** also has the structure of a “double inhibitor” for the oligomers  $\alpha$ ,  $\beta$ ,  $\delta$  and this network displays time-lag oscillation. The expression level of each oligomer becomes stable at the beginning and after a certain period of time, the network begins oscillating.

In this experiment, we showed that our IEC system can be used to obtain networks that display complex behavior. If we are able to implement the operations required to properly combine these networks, the IEC system will be able to identify a network that possesses more complex behavior, such as Duffing oscillation.



**Fig. 12.** An algorithm for finding complex oscillator

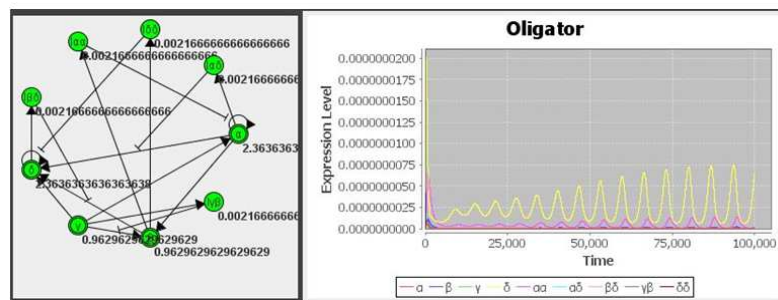


Fig. 13. A network with an amplified oscillation

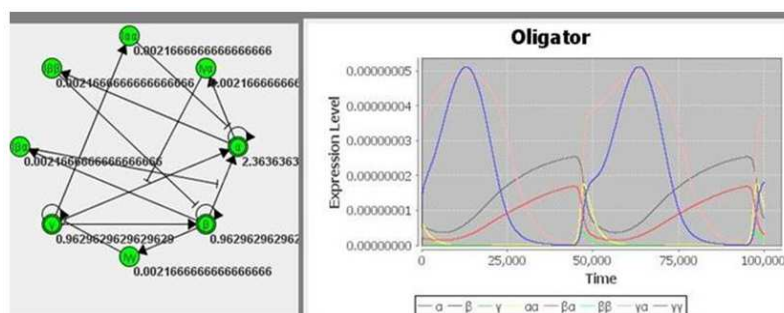


Fig. 14. Oscillation with a distorted waveform

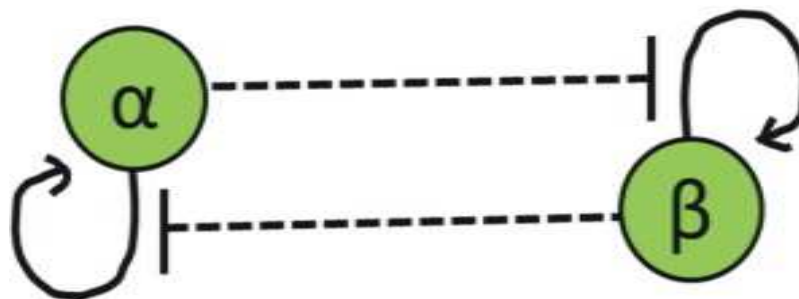


Fig. 15. A simple example of a “double inhibitor”

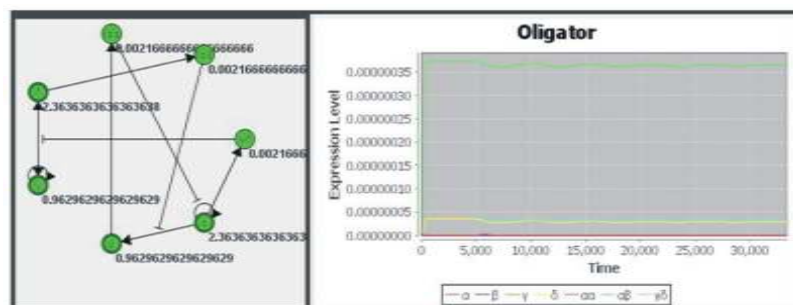


Fig. 16. A network with a time lag before oscillation begins

## 5. CONCLUSION

In this study, we applied the principles of synthetic biology and the IEC technique to design oscillating networks. We succeeded in automation of a portion of this task and demonstrated the efficiency of using human evaluation, which greatly reduced the computational time required. Our system also succeeded in identifying several complex networks with useful structures, which are difficult to design manually. Data on networks such as these that possess “complex” and “interesting” behaviors are useful for the study of synthetic biochemical systems.

### 5.1. Author's Contributions

All authors equally contributed in this work.

### 5.2. Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

## 6. REFERENCES

- Anderson, J.C., C.A. Voigt and A.P. Arkin, 2007. Environmental signal integration by a modular and gate. *Molecular Syst. Biol.* DOI: 10.1038/msb4100173
- Anderson, J.C., E.J. Clarke, A.P. Arkin and C.A. Voigt, 2006. Environmentally controlled invasion of cancer cells by engineered bacteria. *J. Molecular Biol.*, 355: 619-627. DOI: 10.1016/j.jmb.2005.10.076
- Ando, D. and H. Iba, 2007. Interactive composition aid system by means of tree representation of musical phrase. *Proceedings of the IEEE Congress on Evolutionary Computation*, Sept. 25-28, IEEE Xplore Press, Singapore, pp: 4258-4265. DOI: 10.1109/CEC.2007.4425027
- Andrianantoandro, E., S. Basu, D.K. Karig and R. Weiss, 2006. Synthetic biology: new engineering rules for an emerging discipline. *Molecular Syst. Biol.* DOI: 10.1038/msb4100073
- Dinh, Q.H., N. Aubert, N. Noman, T. Fujii and H. Iba *et al.*, 2014. An effective method for evolving reaction networks in synthetic biochemical systems. *IEEE Trans. Evolut. Computat.* DOI: 10.1109/TEVC.2014.2326863
- Elowitz, M.B. and S. Leibler, 1999. A synthetic oscillatory network of transcriptional regulators. *J. Biol. Chem.*, 274: 6074-6079. DOI: 10.1038/35002125
- Gardner, T.S., C.R. Cantor and J.J. Collins, 2000. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403: 339-342. DOI: 10.1038/35002131
- Knight, T., 2003. Idempotent vector design for standard assembly of biobricks. MIT Artificial Intelligence Laboratory, MIT Synthetic Biology Working Group.
- Kobayashi, H., M. Kaern, M. Araki, K. Chung and J.J. Collins *et al.*, 2004. Programmable cells: Interfacing natural and engineered gene networks. *Proceedings of the National Academy of Sciences of the United States of America*, April 26-26, PANAS eTOC Alerts, pp: 8414-8419. DOI: 10.1073/pnas.0402940101
- Montagne, K., R. Plasson, Y. Sakai, T. Fujii and Y. Rondelez, 2011. Programming an in vitro DNA oscillator using a molecular networking strategy. *Molecular Syst. Biol.*, 7: 466-466. DOI: 10.1038/msb.2010.120
- Ohsaki, M., H. Takagi and K. Ohya, 1998. An input method using discrete fitness values for interactive GA. *J. Intelli. Fuzzy Syst.*, 6: 131-145.
- Ono, S. and S. Nakayama, 2012. User-system cooperative evolution for Japanese anagram sentence generation. *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, Jul. 7-11, Philadelphia, USA, pp: 1529-1530. DOI: 10.1145/2330784.2331030
- Takagi, H. and D. Pallez, 2009. Paired comparison-based interactive differential evolution. *Proceedings of the World Congress on Nature and Biologically Inspired Computing*, Dec. 9-11, IEEE Xplore Press, Coimbatore, pp: 475-480. DOI: 10.1109/NABIC.2009.5393359
- Terano, T. and M. Inada, 2003. Data Mining from Clinical Data using Interactive Evolutionary Computation. In: *Advances in Evolutionary Computing*, Ghosh, A. and S. Tsutsui (Eds.), Springer Berlin Heidelberg, ISBN-10: 978-3-642-62386-8, pp: 847-861.
- Unemi, T., 2000. SBART 2.4: An IEC tool for creating 2D images, movies and collage. *Proceedings of Genetic and Evolutionary Computational Conference workshop program*.
- Watanabe, T. and H. Takagi, 1995. Recovering system of the distorted speech using interactive genetic algorithms. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Oct. 22-25, IEEE Xplore Press, Vancouver, BC., pp: 684-689. DOI: 10.1109/ICSMC.1995.537843