

Evolving Artificial Cell Signaling Networks using Molecular Classifier Systems

James Decraene, George Mitchell, Barry McMullin

Artificial Life Laboratory

Research Institute for Networks and Communications Engineering

School of Electronic Engineering

Dublin City University, Glasnevin, Dublin, Ireland

Email: james.decraene@eeng.dcu.ie

Abstract— Nature is a source of inspiration for computational techniques which have been successfully applied to a wide variety of complex application domains. In keeping with this we examine Cell Signaling Networks (CSN) which are chemical networks responsible for coordinating cell activities within their environment. Through evolution they have become highly efficient for governing critical control processes such as immunological responses, cell cycle control or homeostasis. Realising (and evolving) Artificial Cell Signaling Networks (ACSNs) may provide new computational paradigms for a variety of application areas. Our abstraction of Cell Signaling Networks focuses on four characteristic properties distinguished as follows: Computation, Evolution, Crosstalk and Robustness. These properties are also desirable for potential applications in the control systems, computation and signal processing field. These characteristics are used as a guide for the development of an ACSN evolutionary simulation platform. In this paper we present a novel evolutionary approach named Molecular Classifier System (MCS) to simulate such ACSNs. The MCS that we have designed is derived from Holland's Learning Classifier System. The research we are currently involved in is part of the multi disciplinary European funded project, ESIGNET, with the central question of the study of the computational properties of CSNs by evolving them using methods from evolutionary computation, and to re-apply this understanding in developing new ways to model and predict real CSNs.

I. INTRODUCTION

Cell Signaling networks (CSNs) are bio-chemical systems of interacting molecules in cells [1], [2]. Typically, these systems take as inputs chemical signals generated within the cell or communicated from outside. These trigger a cascade of chemical reactions that result in changes of the state of the cell and (or) generate some (chemical) output, such as prokaryotic chemotaxis, coordination of cellular division, or even to order the death of a cell (in the context of multi-cellular organisms).

As signal processing systems, CSNs can be regarded as special purpose computers [3]. In contrast to conventional silicon-based computers, the computation in CSNs is not realized by electronic circuits, but by chemically reacting molecules in the cell. The most important molecular components of CSNs are proteins and nucleic acids (DNA, RNA). There is an almost infinite variety of potential molecular species, each of which would have distinct chemical functionality and could engage in interactions with other molecules with varying degrees of specificity.

Copyright © 2006 IEEE.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of ACM's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.
1-4244-0463-0/06/\$20.00

We distinguish CSNs as being networks made up of more than one distinct cell signaling *pathway*, which interact with each other.

An example of a simple chemotaxis signaling pathway is shown in Figure 1. Chemotaxis is a phenomenon where simple organisms such as bacteria move toward higher concentrations of specific chemicals in their surroundings. In this diagram, we distinguish six intracellular proteins (denoted as A, B, R, W, Y and Z) and the membrane receptors which can bind to the corresponding stimulatory element. The input level is determined by the concentration of bound molecules. This affects the output represented by the tumbling frequency which governs the bacteria direction.

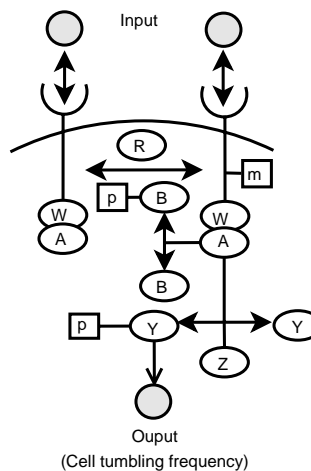


Fig. 1. Schematic representation of bacterial chemotaxis signaling pathway, adapted from [4]. The output is designated by the tumbling frequency which is determined from the input, the concentration level of ligand bound to the membrane receptors. This signal transduction is carried out by the reaction cascade depicted by the proteins A, B, R, W, X and Z. Details on chemical reactions can be found in [5].

Figure 2 shows, in schematic form, a simple Cell Signaling *Network* made up of two such interacting signal *pathways*.

We distinguish this work from previous work on real CSNs [1], [2] by focusing purely on Artificial Cell Signaling Networks (ACSNs). Through the use of evolutionary computing techniques we allow ACSNs to spontaneously emerge and adapt to the environment. Potentially of interest for the Biolo-

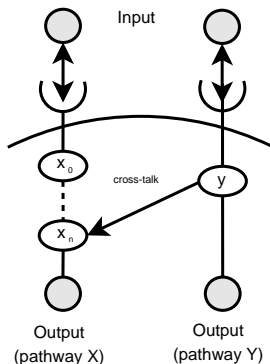


Fig. 2. Schematic illustration of a CSN composed from two distinct cell signaling pathways with unique input and output, an interaction between pathways occurs as molecule y interacts with x_n , this modulates the output of pathway X.

gist may be the insight that ACSNs gives as to how real CSNs evolved and how they operate. This synthetic biology approach allows us to incorporate the present knowledge of real CSNs into ACSNs. This biological understanding provided guiding points that directed the design of the MCS, these points also guide the evolution of ACSNs in silico. This may, for example, facilitate the prediction of missing signaling pathway information in real CSNs [6].

Given our motivation to maintain the biological plausibility of ACSNs, we are interested in investigating the use of ACSNs to implement computation, signal processing and (or) control functionality. This is motivated by preliminary studies which demonstrated that real CSNs could be considered for computational and engineering purposes:

- In [7], Lauffenburger presents his approach to cell signaling pathways which could be thought of and modelled as control modules in living systems.
- Yi et al. [8] demonstrated that CSNs may have some of the essential properties of an integral feedback control. This is a basic engineering strategy to ensure that a system outputs desired values independent of internal and external perturbations.
- Deckard and Saura [9] used and evolved artificial biochemical networks capable of certain simple forms of mathematical computation such as a square root function.

One way to design ACNs to carry out such complex operations is to use artificial evolutionary techniques. A significant insight related to the evolution of signaling networks in silico, was suggested by Holland [6]. Holland proposed examining a simple agent-based model where the agents' behavior and adaptation was determined by the use of Learning Classifier System [10], [11]. Based on this machine learning approach Holland suggested that signaling networks could be modeled with LCS in a *top-down* fashion. We show how this work does not lend itself as a method for addressing our project.

We propose a variation of the LCS called the Molecular Classifier System which allows the emergence and evolution of signaling networks. This approach may be considered as

viewing the evolution and design of signaling networks from a *bottom-up* manner complementing Holland's approach.

In section II we examine the nature of ACSNs by examining the critical issues that these raise. In section III we first present and discuss Holland's approach and following this we propose the structure of our ACSN evolutionary simulation platform, the Molecular Classifier System.

II. ARTIFICIAL CELL SIGNALING NETWORKS

As an abstraction of real CSNs, ACSNs are differentiated and simplified by some key properties. The selection of these particular characteristics is motivated by the will to employ Artificial Cell Signaling Networks for computational and control engineering purposes. Four issues are distinguished and presented: Computation, Evolution, Crosstalk and Robustness.

A. Computation

In the simplest cases, CSNs can be approximately modelled by systems of continuous differential equations, where the state variables are the concentrations of the distinct species of interacting molecules. As a "computational" device, this is most naturally compared to a traditional *analog* computer. Analog computers are precisely designed to model the operation of a target dynamical system, by creating an "analogous" system which shares (approximately) the same dynamics. Electronic analog computers (based on the "operational amplifier" as the core computational device) have long been displaced by digital computers, programmed to numerically solve the relevant dynamical equations, due to their much greater ease of programming and stability.

Nonetheless, there may be applications where a molecular level analog computer, in the form of a CSN, may have distinct advantages. Specifically, CSNs may offer capabilities of high speed and small size that cannot be realised with solid state electronic technology. More critically, where it is required to interface computation with chemical interaction, a CSN may bypass difficult stages of signal transduction that would otherwise be required. This could have direct application in so-called "smart drugs" and other bio-medical interventions.

While CSNs are typically treated in this "aggregate" manner, where the signal or information is carried by molecular concentration, one can also consider the finer grained behaviours of individual molecules are computational in nature. Thus a single enzyme molecule can be regarded as carrying out pattern matching to identify and bind target substrates, and then executing a discrete computational operation in transforming these into the product molecule(s). This has clear parallels with a wide variety of so-called "rewriting systems" in computational theory.

However, it also clearly differs in important ways, such as:

- Operation is stochastic rather than deterministic.
- Operation is intrinsically *reflexive* in that all molecules can, in principle, function as both "rules" (enzymes) and "strings" (substrates/products).

Dittrich [12] provides a more extended discussion of the potential of such “chemical computing”.

B. Evolution

Evolutionary Algorithms (EAs) are non-deterministic search and optimisation algorithms inspired by the principles of neo-Darwinism. They have been applied successfully in a variety of fields [13], [14], [15]. Generally based on genetic operations such as crossover and mutation, EAs initially generate a wide range of candidate solutions. Over time, through selection, this can be reduced to an optimized set. Evolutionary computation can therefore deliver useful results without requiring a priori knowledge of the entire search space [14], [15].

Such techniques are relevant to the study of ACSNs because:

- The complex, and unpredictable, interactions between different components of CSNs, make it very difficult to design them “by hand” to meet specific performance objectives.
- However, natural evolution shows that in suitable circumstances, effective CSNs functionality can be achieved through evolutionary processes.

For example, Deckard and Saura [9] used such evolutionary techniques to construct (simulated) biochemical networks capable of certain simple forms of signal-processing. In this model (called Lakhesis), computational “nodes” represent molecule species with an attribute for concentration. Connections between nodes designate reactions defined by the type and rate of the reaction. The algorithm employed is depicted in Figure 3.

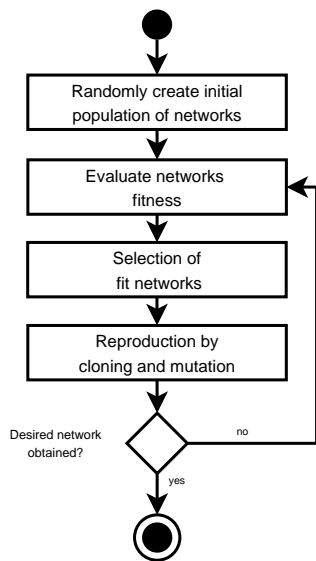


Fig. 3. Flowchart of the Lakhesis evolutionary algorithm adapted from [9], the initial population of networks is randomly generated referring to genetic diversity. Fitness computation is obtained by: solving ODEs which allows one to compute steady states of the network, then the deviation of each node is calculated according to the mathematical computation we want the network to perform. Selection implies removal of unfit networks and reproduction correspond to asexual reproduction where mutation operators are applied. This process is repeated until we get a correct network performing the desired mathematical function.

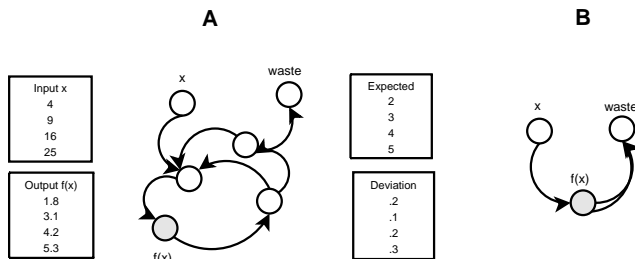


Fig. 4. **A**: Deviation calculation is performed via a series of different inputs. **B**: An evolved network performing exact square-root computation [9]

Each reaction is described by a set of Ordinary Differential Equations (ODE). Solutions to ODEs are generated by using a 4th order Runge-Kutta solver and describe the changes in molecular concentration of the network in time. “Fit” networks are determined by their ability to sustain steady states which correspond to molecular species reaching a constant concentration, with a desired relationship between input and output. Figure 4 **A** shows a random network with series of different inputs/outputs, an evolved network performing exact square-root computation is presented in Figure 4 **B**.

C. Crosstalk

“Crosstalk” phenomena happen when signals from different pathways become mixed together. This arises very naturally in CSNs due to the fact that the molecules from all pathways may share the same physical reaction space (the cell). Depending on the relative specificities of the reactions there is then an automatic potential for any given molecular species to contribute to signal levels in multiple pathways. An example is shown in Fig. 2.

In traditional communications and signal processing engineering, crosstalk is regarded as a defect—an *unintended* or *undesigned* interaction between signals, that therefore has the potential to cause system malfunction. This can also clearly be the case of crosstalk in CSNs. However, in the specific case of CSN’s, crosstalk also has additional potential functionality, which may actually be constructive:

- Even where an interfering signal is, in effect, adding uncorrelated “noise” to a functional signal, this may sometimes improve overall system behaviour. This is well known in conventional control systems engineering in the form of so-called “dither”. Compare also, [16], [17] on constructive biological roles of noise.
- The crosstalk mechanism provides a very generic way of creating a large space of possible modifications or interactions between signaling pathways. Thus, although many cases of crosstalk may be immediately negative in their impact, crosstalk may still be a key mechanism in enabling incremental evolutionary search for more elaborate or complex cell signaling networks.

D. Robustness

It is argued that key properties in biochemical networks are to be robust, this is so as to ensure their correct functioning

[18]. Similar works include research carried out at the Santa Fe institute in studying Cytokine signaling networks to design distributed autonomous networks, that are robust to small perturbations and responsive to larger ones [19]. Potential applications are distributed intelligent systems such as large fleets of robots working together, for automated response in computer security, for mobile computing networks, etc.

Alon et al. have demonstrated from studying *Escherichia coli* chemotaxis that molecular interactions can exhibit robustness [4], [7]. In this case it means that after a change in the stimulus concentration (input), the tumbling frequency (output) managed to reach a steady state that is equivalent to the pre-stimulus level. This is illustrated in Figure 5.

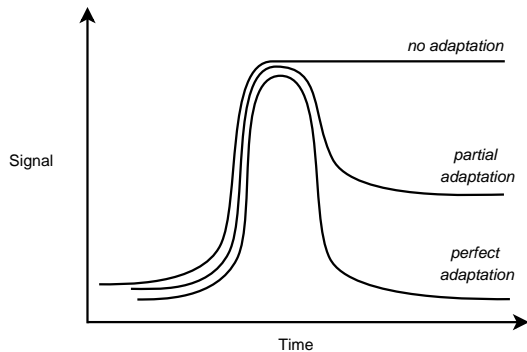


Fig. 5. Representation of dynamic responses of a system to a stimuli adapted from [7]. No adaptation is observed when the system response attains a new steady state following the change in input. Partial adaptation describe a partial recovery, the difference between the initial state and the new state is lower than the one observed in the previous case. Perfect adaptation is met when the system is able to come back to its initial state

Such properties are highly desirable in dynamic engineered systems when subjected to internal and external uncertainty and perturbation.

III. AN EVOLUTIONARY APPROACH TO IMPLEMENT ACSNS

In the following, we first examine a specific class of Evolutionary Algorithm called Learning Classifier System (LCS) devised by Holland in 1976 [10]. In 2001 Holland [6] identified the possibility of using LCS to implement signaling networks (biochemical circuits). However Holland’s work was never actually implemented. We use Holland’s proposition as the seminal point for the development of the first Classifier System based ACSN implementation, our Molecular Classifier System (MCS).

A. Learning Classifier Systems

Learning Classifier Systems are systems constructed from condition-action rules called *classifiers*. The classifiers can be viewed as IF/THEN statements in the form IF “rule” THEN “action”. The condition section of the classifier examines all of the messages in the system and identifies those that satisfy the *rules* conditions. Once this is accomplished the action part instructs that a message is to be sent. Holland’s initial work

was modified a number of times and at present many different varieties of learning classifier systems are available [20].

In Holland’s LCS the system receives an input from its environment as a binary encoded data. This is then stored in an internal data store termed the *message list*, see Figure 6. The LCS then evaluates the input and determines an appropriate response, indicated by the action. This action typically alters the current state of the environment. Any desired behaviour that is exhibited is then rewarded through a scalar reinforcement. The system iterates the cycle of response, reinforcement and discovery for each discrete time-step.

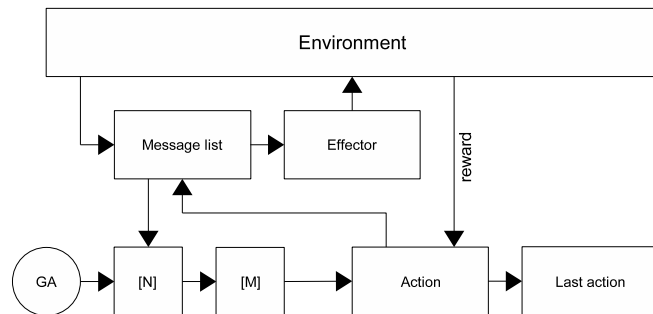


Fig. 6. Schematic of Holland’s Learning Classifier System

The rule-base consists of a population of N classifiers. Both parts of the classifier are randomly initialized. The rule conditions and actions (the classifiers) can be characterized by strings formed from a ternary alphabet 0,1,#. The use of the # provides a single character wildcard which allows for the potential matching of a greater number of strings e.g. 10# would match two potential inputs 100 or 101. The use of the wildcard character also provides for string processing at the action stage, for example: in responding to the input 110, the rule IF 1#0 THEN 0#1 would produce the action 011. Each classifier also has an associated fitness measure, quantifying the *usefulness* of a rule in attracting external reward.

On receiving an input message, a typical LCS processes as follows: initially the input message rule-base is scanned and all rules whose condition matches the external message are added to the “match set” denoted as $[M]$, see Figure 6. Secondly any other rules matching messages in the message list are also added to $[M]$.

Then through a bidding technique a rule is selected from the rules in $[M]$. Once the rule is selected, it then becomes the system’s external action. Following this, the message list is purged and the action string is posted in preparation for the next cycle. Any remaining spaces on the internal message list can then be filled by of other rules selected via the bidding technique. This bidding technique can be a simple stochastic roulette wheel selection method. The rules’ bids are formed from two components: the rules fitness and the rules specificity, that is the proportion of literal characters that the string contains. The following formula is typically used:

$$\text{Bid}(C, t) = \beta \cdot \text{specificity}(C) \cdot \text{fitness}(C, t)$$

The reinforcement stage sets about redistributing bids across rules to be chosen later in the cycle. At each time step, winners placed their bids in a *bucket*. Each of the previous winners are then equally given a portion of the bucket, the fitness being shared amongst the activated rules. When a reward is received from the environment, the winning rule that produced the last output obtains this reward.

Holland terms this technique the *bucket brigade algorithm* and this can be viewed as an economical analogy to the middle man in a financial arrangement. A steady state genetic algorithm is utilized over the whole rule-set. After a defined number of time-steps the GA uses a roulette wheel technique to select parent rules based upon the fitness of the rules. The parent rules produce offspring through the use of a mutation and crossover operator and the new offspring replace an existing rule from the population.

For a comprehensive introduction to Learning Classifier System, see [11].

B. Implementing signaling networks with LCS

In [6] Holland proposed an agent-based model where the agents' behavior and adaptation are determined by the use of Learning Classifier System. This work provided an existence proof that LCS could be used to evolve a simple repertoire of condition-action rules to a more complex goal directed set of rules. In typical biochemical networks, interactions between molecules follow the same condition-action mechanisms. Thus Holland suggested that this approach could be used to simulate and evolve signaling networks.

His proposition to design signaling networks was to start with a LCS-based "over-general" model of a biological phenomenon (e.g. transformation of a healthy cell to a cancer cell, see Table I). Then he refined this general phenomenon through several iterations. At each iteration, the details of the occurring interactions were refined, see Table II. These iterations were continued until the desired CSN level was reached, where the biomolecular elements are specified (e.g. protein ligand, receptor, ions etc.), see Table III for an example of such a rule. This refining process clearly shows the top-down methodology to design signaling networks.

(1)	IF healthy cell and DNA damage	THEN apoptosis or immortality
(2)	IF immortality	THEN stable existence or genetic instability
(3)	IF genetic instability	THEN ephemeral clonal expansion or robust clonal expansion
(4)	robust clonal expansion	THEN cancer mass

TABLE I

OVER-GENERAL MODEL OF THE TRANSFORMATION OF HEALTHY CELL TO CANCER CELL

Despite this, the LCS-based approach to specify CSNs sounded promising, actual implementation was never performed. Importantly, this approach does not meet the requirements of our project. First, we do not distinguish a demarcation between rules and messages, in our context, the chemical

(1.1)	IF healthy cell and DNA damage	THEN apoptosis or mutation for resistance to apoptosis
(1.2)	IF resistance to apoptosis	THEN susceptibility to growth inhibitory signals or mutation for loss of susceptibility to growth inhibitory
(1.3)	IF loss of susceptibility to immortality	THEN selective growth advantage and growth inhibitory signals

TABLE II

REFINEMENT OF RULE I

IF apropos growth factor	THEN gf receptor activated
--------------------------	----------------------------

TABLE III

A BIOMOLECULAR LEVEL RULE

operations are reflexives. Secondly, Holland's suggestion was to initially model known real CSNs, however from our bottom-up perspective, we require the ACSNs to evolve from very simple networks to more complex networks that exhibit the known real CSNs properties. As a consequence we propose a variation of Holland's LCS to fulfill the requirements of our project.

C. The Molecular Classifier System and ACSNs

Holland distinguished a demarcation between *rules* and *messages*, however operations in a biochemical networks are intrinsically *reflexive* in the sense that all molecules can function as both rules (enzymes) and messages (substrates/products). In the MCS, we address this through the use of a dual-level encoding of the artificial molecules. The first level encoding referring to primary structure (messages) as a string of variable length and the second level corresponding to secondary structure (rules).

A similar multi-level encoding approach was introduced by Hofstadter [22] and is called Typogenetics. The aim of Hofstadter's system was to describe some of the basic principles of molecular biology using a typographic system. In this system, the primary structure alphabet includes the four amino acids ACGT, secondary structure is extracted by splicing strands of amino acids from the primary structure. These strands of amino acids correspond to their enzymatic functions identified by the use of a lookup table.

Similarly the MCS uses this primary-secondary structure mapping, the primary structure alphabet used in the MCS contains 1s and 0s. The secondary structure ("folded") is determined by extracting *codons* of n bits, e.g. if we are using codons of 3 bits in length then the secondary structure alphabet would comprise to a maximum of 8 different symbols or *enzymatic functions*, see Table V.

The variety of operations is limited by the codon length. Codons having longer bit length allow a greater number of computational functions. For example, Table V presents a chemical language comprising 6 functional operations $\{H, L, :, \#, ?, , :\}$. By using a codon length of 3, we allow a set

		Encoding Structure											
Primary		1	0	1	0	1	1	0	0	0	1	0	1
Secondary		H			#			:			H		

TABLE IV

A SIMPLE ARTIFICIAL MOLECULE USING THE MCS ENCODING

of 8 distinct computational functions to be defined. In this definition, redundant functions may also be specified within this set of functions.

000	001	010	011	100	101	110	111
:	H	*	#	L	H	L	?

TABLE V

AN EXAMPLE OF A PRIMARY/SECONDARY STRUCTURE ALPHABET MAPPING

The behavior of the condition/binding properties and action/enzymatic functions is specified by a “chemical” language defined in the MCS. The more computational functions a chemical language defines, the more complex chemical behavior can be observed in a MCS simulation.

Before describing the nature of the enzymatic functions (action part of a molecule), the binding properties of the molecules must be identified. We have thus far identified the following potential properties: in the LCS approach, a reaction between two molecules may only occur if one of the molecules (message) satisfies the conditional part of a candidate molecule (self binding may also occur). In the MCS we are not using the message/rule demarcation, instead we are using the following:

- The message is encoded in the primary structure
- The condition/action rule is encoded in the secondary structure

The condition part of a rule refers to the binding properties of a molecule whereas action refers to the computational (“enzymatic”) function. Thus, for a reaction to occur in the MCS between two molecules, the primary structure of a molecule needs to satisfy the secondary structure condition part/binding rule of a candidate molecule.

Following this description of the binding rule used in the MCS, we now present the enzymatic functions that may occur. When two molecules can bind to each other and consequently can react with other, the secondary structure action part of one of the molecules is used to carry out the enzymatic operations upon the binding molecule (substrate). This operation results in producing another artificial molecule (product). This is analogous to the action part of a LCS rule used by Holland [6].

As shown in Table V, we have a limited number of symbols/operations in the secondary structure alphabet according to the codon length used. When a reaction occurs, the symbols contained in the secondary structure are processed in a sequential order, also the secondary structure symbols are parsed from the left. The outcome (product) of the reaction depends on the nature of the symbols’ functionality. The definitive set

of operations is still under investigation as we are trying to understand what are the minimal operational requirements to allow a primitive ACSN to spontaneously emerge. This would facilitate more complex ACSNs mimicking natural CSNs to be built/evolved upon this *generic* ACSNs.

A *potential* set of operations to be used in the secondary structure is defined as follows. Two classes of operation can be discriminated: the condition/binding pattern language and action/enzymatic functions. A punctuation mark “:” is used to distinguish the condition/action part in the secondary structure. This punctuation mark may or may not be present in the secondary structure, a molecule having no punctuation mark in the secondary structure would have no enzymatic function. We define the pattern language used for binding properties as follows:

- “H” matches 1
- “L” matches 0
- “#” single symbol wildcard matches exactly 0 or 1
- “*” wildcard matches substring of any length

Pattern matching implies a notion of *specificity* or “binding strength”. A molecule having a high specificity would have less chance to react with another one. Whereas a molecule having a low specificity is likely to bind to another more often. Therefore we could translate this into an effect on reaction rate/kinetics.

When a molecule *A* can react with a molecule *B*, the action statement of the secondary structure of molecule *A* is “executed” on the primary structure of the binding molecule *B*. *A* is viewed as an enzyme and *B* as a substrate, thus *A*’s structure is not affected by the reaction whereas *B*’s structure is degraded and a product *P* is generated. *A*’s secondary structure operators take as inputs the symbols of *B*’s primary structure. An offspring molecule *P* is generated as a result of these operations, see Figure 7.

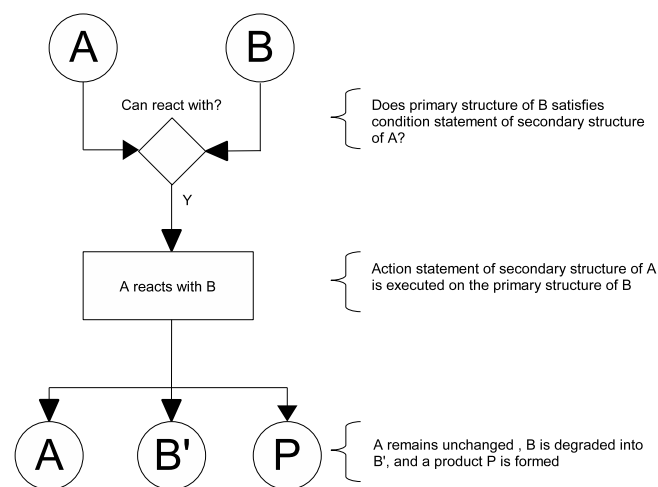


Fig. 7. Schematic of a reaction in the MCS

Regarding the potential “action” language, several operators can be distinguished:

- “-” cleavage operator

- “+” ligation operator
- “c” copy operator

The copy operator can be subjected to random mutation determined by a probability P_m . Also the symbol segment length to be copied is variable. Potential mutations that may occur during copy are as follows:

- point mutation: the offspring gets the opposite symbol to the parent symbol
- point deletion: the symbol is not replicated
- point insertion: the symbol is copied again

In addition to mutation operators, the evolutionary process is carried through the use of several common genetic operations such as crossover, selection routines, and appropriate evaluation function for the domain. Defining the MCS is a natural step to Holland’s work on employing evolutionary techniques to investigate signaling networks.

IV. FUTURE WORK

In keeping with the four presented ACSN characteristic properties, we will focus on the following areas:

A. Computation

Concerning the language implementation, we are also investigating in parallel other potential languages such as the broadcast language proposed by Holland [10]. The broadcast language was the precursor of LCS. Holland envisaged the modeling of Genetic Regulatory Networks using this approach, however this was never performed. A motivation to investigate this language is that it may be Turing Complete as suggested by Holland.

We are investigating the computational power of the broadcast language, this will also us to analyze and demonstrate whether or not the broadcast language is Turing Complete. As one of our project goals is to evolve ACSNs for computational purposes, incorporating completeness may be regarded as a crucial issue.

B. Evolution

An ACSN implies several cell signaling pathways interacting with each other. In order to evolve such a system of signaling networks *controlling* each other, it will be necessary to evaluate different Evolutionary Computational (EC) techniques. Because from biology it is natural to have a *hierarchical* system it may prove beneficial to investigate multi-level EC systems e.g. Hierarchical Genetic Algorithms [23].

C. Crosstalk

To obtain a better understanding of the crosstalk phenomenon and more specifically about the positive and negative effects of crosstalk. We will would like to see if it is possible to specify a network topology that allows optimal control of crosstalk effects.

A small world topology [24] may be of interest, as we may observe an analogy between CSNs and small world

networks. This class of network, and more specifically scale-free networks are characterized by possessing nodes acting as “highly connected hubs”. Although most nodes in these networks are of low degree. For example, a highly connected node could be referring to an ATP molecule that shares the same high degree of connectivity in real biochemical networks.

D. Robustness

We will investigate the ability of ACSNs to create and sustain specific internal conditions such as homeostasis. We would like to exhibit such robust behavior in simulated ACSNs, and how through evolutionary changes, robustness can be refined. Another consequent issue is to quantify the robustness of such systems to external shocks and changes of conditions.

V. CONCLUSION

In this paper we have introduced a novel mechanism: the MCS for evolving ACSNs from a bottom-up approach. This technique is inspired by the work of Holland and Hofstadter utilizing Classifier Systems and Multi-level encodings. We discussed Holland’s approach to implement biological circuits, in which known real CSNs could be specified in a top-down fashion using LCS. However this remains an unimplemented sketch to model real CSNs. The requirements of our project contradict his proposed method on several points. Principally, we want to evolve signaling networks from a bottom-up approach. Secondly we are examining *artificial* signaling networks, and these were characterized as potentially providing four properties distinguished as follows: Computation, Evolution, Crosstalk and Robustness. We indicated how these attributes can be highly desirable properties for potential applications in the control systems, computation and signal processing field. We completed the paper by examining further work that is required to conclusively validate our approach.

ACKNOWLEDGEMENT

This work was funded by ESIGNET (Evolving Cell Signaling Networks in Silico), an European Integrated Project in the EU FP6 NEST Initiative (contract no. 12789).

REFERENCES

- [1] E. J. M. Helmreich, *The Biochemistry of Cell Signalling*. USA: Oxford University Press, 2001.
- [2] G. Krauss, *Biochemistry of Signal Transduction and Regulation*. John Wiley & Sons, 2003.
- [3] D. Bray, “Protein molecules as computational elements in living cells,” *Nature*, vol. 376, no. 6538, pp. 307–312, Jul 1995.
- [4] U. Alon, M. G. Surette, N. Barkai, and S. Leibler, “Robustness in bacterial chemotaxis,” *Nature*, vol. 397, no. 6715, pp. 168–171, January 1999.
- [5] R. C. Stewart and F. W. Dahlquist, “Molecular components of bacterial chemotaxis,” *Chem. Rev.*, vol. 87, pp. 997–1025, 1987.
- [6] J. Holland, “Exploring the evolution of complexity in signaling networks,” *Complexity*, vol. 7, no. 2, pp. 34–45, 2001.
- [7] D. Lauffenburger, “Cell signaling pathways as control modules: complexity for simplicity?” *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 10, pp. 5031–3, 2000.
- [8] T. M. Yi, Y. Huang, M. I. Simon, and J. Doyle, “Robust perfect adaptation in bacterial chemotaxis through integral feedback control!” *Proc Natl Acad Sci U S A*, vol. 97, no. 9, pp. 4649–4653, April 2000.

- [9] A. Deckard and H. M. Sauro, "Preliminary studies on the in silico evolution of biochemical networks." *Chembiochem*, vol. 5, no. 10, pp. 1423–1431, October 2004.
- [10] J. Holland, "Adaptation," *Progress in theoretical biology*, vol. 4, pp. 263–293, 1976.
- [11] L. Bull and T. Kovacs, "Foundations of Learning Classifier Systems: An Introduction," *Foundations of Learning Classifier Systems*, 2005.
- [12] P. Dittrich, "Chemical computing." in *UPP*, ser. Lecture Notes in Computer Science, J.-P. Banâtre, P. Fradet, J.-L. Giavitto, and O. Michel, Eds., vol. 3566. Springer, 2004, pp. 19–32.
- [13] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.
- [14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989. [Online]. Available: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/0201157675>
- [15] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, December 1992. [Online]. Available: <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/0262111705>
- [16] A. M. Arias and P. Hayward, "Filtering transcriptional noise during development: concepts and mechanisms," *Nature Reviews Genetics*, vol. 7, no. 1, pp. 34–44. [Online]. Available: <http://dx.doi.org/10.1038/nrg1750>
- [17] D. Volfson, J. Marciniak, W. J. Blake, N. Ostroff, L. S. Tsimring, and J. Hasty, "Origins of extrinsic variability in eukaryotic gene expression," *Nature*, December 2005. [Online]. Available: <http://dx.doi.org/10.1038/nature04281>
- [18] N. Barkai and S. Leibler, "Robustness in simple biochemical networks." *Nature*, vol. 387, no. 6636, pp. 913–917, June 1997. [Online]. Available: <http://dx.doi.org/10.1038/43199>
- [19] S. F. Lee Segel, "Robustness of cytokine signalling networks. <http://www.santafe.edu/research/signallingnetworks.php>."
- [20] P. L. Lanzi, W. Stolzmann, and S. W. Wilson, Eds. Springer-Verlag, April 2001.
- [21] L. Bull, *Learning Classifier Systems: A Brief Introduction*. Springer-Verlag, 2004, pp. 3–14.
- [22] D. Hofstadter, *Gödel, Escher, Bach: an eternal golden braid*. Basic Books, 1999.
- [23] B. Freisleben, "Metaevolutionary approaches," in *Handbook of Evolutionary Computation*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. Bristol, New York: Institute of Physics Publishing and Oxford University Press, 1997, pp. C7.2:1–8.
- [24] M. E. J. Newman, "Models of the small world: A review," May 2000. [Online]. Available: <http://arxiv.org/abs/cond-mat/0001118>