UDC 519.716.39: 519.6: 57.017

A.A. Fefelov, M.A. Taif, V.I. Lytvynenko, M.A. Voronenko,
O.M. Stepanchenko, I.V. Sokur, I.A. Lurie

REVERSE ENGINEERING OF THE GENE REGULATORY NETWORK WITH COMBINED USE OF THE CLONAL SELECTION ALGORITHM AND DIFFERENTIAL EVOLUTION

Abstract. *The development of methods for reverse engineering of gene regulatory networks is one of the important tasks in the post-genomic era. The algorithm of clonal selection is one of the popular approaches for the reconstruction of gene networks based on microchip data.*

Keywords: *Reverse engineering system reconstruction, population, gene regulatory networks, clonal selection algorithm, differential evolution algorithm, DNA microarrays.*

I. Introduction

Reverse engineering or system reconstruction is the process of deriving the structural and dynamic system characteristics under study on the observations basis of its behavior and certain knowledge in the relevant subject area.

Today, reconstruction plays an important role in biology as one of the main tools for modeling biological systems and their interactions, which is key to understanding the mechanisms of their functioning [1]. One of the most widely used applications of the methodology of reconstruction in biology is the identification of gene regulatory networks (GRN) [2]. Detection of the presence and nature of interactions between GRN genes is essential for the creation of new drugs. Reconstruction helps researchers find answers to a number of questions, among which we can distinguish the following: what processes in the body regulate the gene being studied; which genes affect the test gene; how genes interact; what genes are responsible for a particular type of disease; which drugs will have an effect in case of a disease, etc. The technology of DNA microarrays [3] facilitates the study of the behavior of GRN greatly, but the task of reconstruction remains difficult, since it contains a large number of unknowns.

A number of mathematical models are used to describe the structure and dynamics of GRS, among which the most widespread are Boolean

 ISSN 1562-9945

networks [4], linear models [5], differential equations [6], associative networks [7], Bayesian networks [8], neural networks [9, 10], the model of the state space [11]. In this paper, the classical system of ordinary differential equations (ODE) in the form of an S-system was chosen as the GRN model, for the identification of which a hybrid algorithm for clonal selection and differential evolution was proposed.

II. Formulation of the problem

For a regulatory network consisting of N genes, the S-system is represented by the ODE system of the following kind [7]:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} x_j^{h_{ij}}, i = 1 \dots N, \quad (1)$$

where $x_i(t)$ – expression's level of i-th gene at the time point t; α_i, β_i – non-negative numbers, called rate constants; g_{ij}, h_{ij} – kinetic orders that determine the direction and level of regulatory impact, that is, stimulation or inhibition.

Reconstruction of the regulatory network involves not only choosing a mathematical model, but also developing a method for identifying it. Identification of the S-system consists in finding optimal values of parameters from the set $v = \{\alpha, \beta, g, h\}$. The complexity of the task's solution is due to its high dimensionality. The number of parameters to be found is determined by the expression $2N(N + 1)$. That is, for GRN consisting of only four genes, the search space dimension is 40. For this reason, the task of reconstructing the S-system can't be solved analytically. It is known that population methods such as genetic algorithms or artificial immune systems have proved to be successful in solving such problems. But at large dimensions, the convergence rate of population methods can be very low. Thus, the aim of this study is to develop a new, fast and accurate method for optimizing the S-system's parameters, taking advantage of the population approach and hybridization technology.

III. The algorithm of clonal selection

In the research [15] immune system is considered from the view point of the mechanism of clonal selection. On the basis of clonal selection principle, an optimization algorithm CLONALG was proposed in [16], which is widely used at present as one of the varieties of IIS. In clonal algorithm, the affinity values express the proximity measure of the

individual to the optimal solution and are calculated on the basis of the objective task's function. Step-by-step algorithm description is presented below.

Step 1. $Population = 0$.

Step 2. Create an initial population of solutions randomly (Ab^0).

Step 3. Estimate populations Ab^0 based on the objective function f.

Cycle until the condition of shutdown $e = false$.

Step 4. Save the best solution in the current generation.

Step 5. Select antibodies from Ab^0 with the greatest affinity.

Step 6. Create clones Ab^c of selected antibodies in an amount $n \sim f$.

Step 7. To make a mutation of clones with intensity $p_m! \sim f$.

Step 8. Estimate populations Ab^c based on the objective function f.

Step 9. Select clones with the greatest affinity from Ab^c and transfer them to Ab^0.

Step 10. Replace the d worst antibody in Ab^0 new antibodies generated randomly.

Step 11. Estimate new antibodies in Ab^0.

Step 12. $Population = Population + 1$.

Cycle until end

Step 13. Conclusion: the best solution in the current generation.

In CLONALG you can use different ways of presenting solutions, depending on the type of task. The binary and real representations are most often used. Also, the conditions and objectives of the problem are decisive when choosing a method for representing immune operators, the affinity function form , and algorithm parameters values.

When calculating the main population affinity, conditions are created for the selection of cells, that are complete (at this stage) enter into interaction with the antigen maximally, that is, they form the minimum of the objective function. During the activation process, the selected antibodies increase their representation in the solution space due to cloning. Cells, whose affinity is higher, create more clones, but this cells are less susceptible to mutation. The mutation in CLONALG is of high intensity, since it is the main driving force of evolution. In the process of replacement, cells with low affinity are removed from the main population, and new generated randomly individuals come in their place. This avoids local extremes, and explores the entire target surface.

Individuals of the population are coded by real numbers rows in the range from 0.0 to 1.0. Each line contains a complete set of parameters of the S-system (Fig. 1).

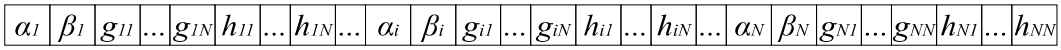

Fig. 1. Encoding of antibodies of the hybrid clonal algorithm

When evaluating the next solution variant, the values of the individual string are recalculated into new ranges, in accordance with the allowable intervals of the change of one or another S-system's parameter, which are set before the algorithm start.

The affinity measure or objective function of the task is left the same as in work [11]:

$$f = MIN \sum_{i=1}^{N} \sum_{j=1}^{T} \left(\frac{x_i^M(t_0 + j\Delta t) - x_i(t_0 + j\Delta t)}{x_i(t_0 + j\Delta t)} \right)^2, \quad (3)$$

where t_0 – time to start measuring the gene expression level; Δt – time step between successive measurements; T – number of measurements; $x_i^M(t_0 + j\Delta t)$ – values of the i-th gene expression level, obtained by means of the model, i.e., by the decision of SODU (1) (in this case by the Runge-Kutta method of the fourth order); $x_i(t_0 + j\Delta t)$ – measured values of the i-th gene expression level.

IV. Differential evolution algorithm

The differential evolution algorithm (DE) is one of the evolutionary algorithms varieties [18, 19]. With its high efficiency, DE has found application in many subject areas, as a global optimization method. There are several variants of DE, differing in details of the evolutionary operators implementation. In this paper we use the version presented in [18]. A brief description of DE method is given below.

The task of objective function minimization is considered:

$$f(\boldsymbol{x}) \to min, \boldsymbol{x} = (x_1, \ldots, x_n), \quad (3)$$

where $\boldsymbol{x}$ – the vector of task's parameters, on the basis of which the individuals of the solutions populations are built $\boldsymbol{x}_i^G, i = 1, \ldots, P$; P – размер популяции решений; G – текущее поколение.

The main difference between DE algorithm and other evolutionary algorithms is the implementation of the mutation operator. The DE mutation is as follows:

$$\boldsymbol{v}_i^{G+1} = \boldsymbol{x}_{r3}^G + F(\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G), \quad (4)$$

where $\boldsymbol{v}_i^{G+1}, i = 1,\ldots,P$ – an individual, that is the result of a mutation; $r1, r2, r3 \in \{1,\ldots,P\}$ – indexes of individuals, that are chosen from the population of solutions randomly in the current generation, such that $r1 \neq r2 \neq r3 \neq i$; F – scale coefficient ($F \geqslant 0$).

Components of individuals $\boldsymbol{x}_i^G$ are replaced by the corresponding components of the vectors partially $\boldsymbol{v}_i^{G+1}$ with the formation of candidates population $\boldsymbol{u}_i^{G+1} = (u_{i1}^{G+1},\ldots,u_{in}^{G+1})$. Formation of a vector $\boldsymbol{u}_i^{G+1}$ occurs according to the following expression:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, if\ randEvent(p_{DE}) = 1 \vee j = k \\ x_{ij}^G, otherwise \end{cases}, j = 1,2,\ldots,n, \quad (5)$$

where $k \in \{1,\ldots,n\}$ random parameter index, chosen once for each individual, the sense of which is to ensure the transition of at least one vector component $\boldsymbol{v}_i^{G+1}$ to vector $\boldsymbol{u}_i^{G+1}$; p_{DE} – the transition probability of the j-th component of the vector $\boldsymbol{v}_i^{G+1}$ to vector $\boldsymbol{u}_i^{G+1}$. Since expression (5) is associated with the crossover operator in evolutionary algorithms, p_{DE} called the crossing-over probability. It is not difficult to see that expression (5) is very similar to expression (2), therefore in this work p_{DE} is called the intensity of DE mutation.

The next generation population is formed from the current generation population and the candidates population by means of DE-selection:

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^{G+1}, if\ f(\boldsymbol{u}_i^{G+1}) \leq f(\boldsymbol{x}_i^G) \\ \boldsymbol{x}_i^G, otherwise \end{cases}, \quad (6)$$

i.e. each individual from the candidate population is compared with the corresponding individual from the current population. If the candidate has a smaller value of the objective function, he passes into a new generation. Otherwise, the current individual passes into the new generation.

V. The proposed hybrid algorithm

The idea of combining different computational methods and the formation of hybrids is based on the assumption that the new computational method obtained as a result of combining should have a

higher productivity than the constituent components. Being embodied in life, this idea led to the creation of hybridization technology, which allows synthesizing whole algorithms classes capable of solving complex tasks at a qualitatively new level. According to this technology, the strengths of the methods involved in hybridization form a combined result that characterizes the main advantages of the hybrid approach, such as: obtaining better solutions; getting solutions in less time; solving problems of large dimension.

In this paper, a hybrid clonal algorithm is proposed in which the mutation phase is extended by operators taken from the algorithm of differential evolution. A step-by-step description of the algorithm is presented below.

Step 1. $Population = 0$.

Step 2. Create an initial population of solutions randomly (Ab^0).

Step 3. Estimate populations Ab^0 based on the objective function f.

Cycle until the condition of shutdown $e = false$.

Step 4. Save the best solution in the current generation.

Step 5. Select antibodies from Ab^0 with the greatest affinity.

Step 6. Create clones Ab^c of selected antibodies in an amount $n \sim f$.

Step 7. To make a mutation of clones with probability $p_c! \sim f$ and intensity $p_m! \sim f$ according to the formula (2).

Step 8. To make DE-mutation of clones with intensity $p_{DE}! \sim f$ according to the formula (4) и (5) and place the results into the population of candidates Ab^t.

Step 9. Estimate populations Ab^t based on the objective function f.

Step 10. To carry out DE-selection of candidates according to the formula (6).

Step 11. Select the candidates with the highest affinity from Ab^t and transfer them to Ab^0.

Step 12. Replace the d worst antibody in Ab^0 new generated randomly antibodies.

Step 13. Estimate new antibodies into Ab^0.

Step 14. $Population = Population + 1$.

Cycle until end

Step 15. Conclusion: the best solution in the current generation.

Step 7 is left to prevent premature convergence of the algorithm. Unlike the classical CLONALG, in this case, mutations are not all clones, but only a part of them, which is controlled by the parameter p_c. In this case, the probability values p_c and intensity p_m are subject to investigation. In addition, the authors studied the effect of the intensity of the DE-mutation p_{DE}, the scale factor F and DE-selection on the quality of the algorithm. The results of the research are presented in the next section.

VI. Experimental research

In this research work, the efficiency of the algorithm was evaluated both on artificial and on real gene networks, and the experimental results were compared with other existing methods found in the modern literature. Now the performance of the S-system model was measured in terms of its sensitivity (Sn) and specificity (Sp), which were defined as follows:

$$S_n = \frac{TP}{TP + FN}, \tag{4}$$

$$S_p = \frac{TN}{TN + FP}, \tag{5}$$

where True Positive (TP) denotes the number of correctly predicted rules, and True Negative (TN) is the number of correctly predicted uncontrollable output algorithms. False Positive (FP) indicates the number of incorrectly predicted rules, and False Negative (FN) represents the number of false-predicted uncontrollable output algorithms.

To test the effectiveness of modeling the S-system, a reference small artificial network was chosen. It contains five genes with simple regulatory dynamics. In the studies of other authors this network was already used to test the effectiveness of the proposed algorithms. Therefore, in this article, the same network was used to test our methodology, and also to compare its effectiveness with earlier work.

The proposed methodology was applied to synthetic noiseless data having the parameters shown in Table 1.

Time series data were obtained by solving the system of differential equations (1). For training, we used datasets containing only positive values. In our work, only 70 data points were used for each of the genes. Twelve identifiable parameters were also used. The search interval is

 ISSN 1562-9945

selected in the range [-3; 3] for kinetic orders $g_{i,j}$ и $h_{i,j}$ in the range [0,12] for rate constants, α_i и β_i.

Table 1.

The actual parameters of the S-system for an artificial gene network.

	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$	α_i	β_i
1	0	0.0	1.0	0.0	-1.0	2.0	0.0	0.0	0.0	0.0	5.0	10.0
2	2	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	10.0	10.0
3	0.0	-1.0	0.0	0.0	0.0	0.0	-1.0	2.0	0.0	0.0	10.0	10.0
4	0.0	0.0	2.0	0.0	-1.0	0.0	0.0	0.0	2.0	0.0	8.0	10.0
5	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	2.0	10.0	10.0

For the hybrid clonal selection algorithm (HACS), 8000 iterations were used. The number of the maximum iteration and the initial population remains high for solving the nonlinearity of the S-system. Initial parameter values α_i and β_i were established in 0,95. The choice of parameters requires certain experiments. The frequency boundary and the step size were initialized to [0,1], while the random movement was fixed to 0.001.

Table 2 shows the expected parameters for the experiment. The values of kinetic orders less than 0.1 were ignored. The S-system model based on the hybrid clonal algorithm produced satisfactory results for silent data, since almost all the parameters were predicted to be correct. In addition, HACS also accurately determined the correct sign and position of the rules (regulated and unregulated). However, the predicted parameter values for gene No. 3 were somewhat less accurate, but still quite satisfactory in terms of predicting TP and FP and the nature of their regulation.

Table 2.

Calculated values of parameters of the S-system for an artificial network

	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$	α_i	β_i
1	0.2	0.0	1.2	0.0	-1.0	2.1	0.0	0.0	0.0	0.2	5.0	10.0
2	2.4	0.2	0.0	0.2	0.0	0.2	2.0	0.2	0.0	0.0	9.0	9.0
3	0.0	-1.1	-0.2	-0.2	-0.2	0.0	-1.1	0.0	0.0	0.0	7.0	8.0
4	0.3	0.0	2.1	0.0	-1.1	0.0	0.0	0.0	2.0	0.0	7.6	9.5
5	0.2	0.0	0.0	2.1	0.0	0.0	0.0	0.0	0.0	2.1	9.5	9.4

To fully evaluate the proposed HACS algorithm, it was tested on noisy artificial data. The approach proposed in [19] was used, where 5%, 15% and 25% were added to the data.

Table 3 presents comparative studies of the effectiveness of various reverse engineering algorithms. It can be noted that in the presence of different noise levels, the HACS-based S-system model is better than the algorithms based on differential evolution (DE) and the algorithm for clonal selection.

Table 3.

Comparative study of HACS-based S-system with the other techniques for noisy artificial data

Methods	HACS		ACS		DE	
Noise(%)	S_n	S_p	S_n	S_p	S_n	S_p
5	1.0	0.96	1.0	087	1.0	0.75
15	1.0	0.79	0.81	0.71	0.93	0.73
25	1.0	0.68	0.65	0.68	0.89	0.74

VII. Conclusions

The report describes the application of the S-system to search for gene networks from microchip's data using the combined algorithms of clonal selection and differential evolution. The algorithm is implemented on the microchip data which, as shown, is able to determine and predict the dynamics of the studied data on which the training was conducted. The obtained results show the high accuracy of the algorithm. One of the problems with the reconstruction of gene networks from real data is the lack of final network models to compare the methods being developed.

REFERENCES

1. Csete M. Reverse Engineering of Biological Complexity / M. Csete, J. Doyle // Science. – 2002. – Vol. 295. – P. 1664-1669.
2. Cho K.H. Reverse engineering of gene regulatory networks / K.H. Cho, S.M. Choo, S.H. Jung [et al.] // IET System Biology. – 2007. – Vol. 1. – P. 149-163.
3. DeRisi J.L. Exploring the metabolic and genetic control of gene expression on a genomic scale / J.L. DeRisi, V.R. Lyer, P.O. Brown // Science. – 1997. – Vol. 278. – P. 680-686.
4. Liang S. Reveal, a general reverse engineering algorithm for inference of genetic network architectures / S. Liang, S. Fuhrman,

R. Somogyi // Pacific Symposium on Biocomputing (PSB '98), January 1998: proceedings. – Maui, Hawaii, USA, 1998. – P. 18-29.
5. D'haeseleer P. Linear modeling of mRNA expression levels during CNS development and injury / P. D'haeseleer, X. Wen, S. Fuhrman [et al.] // 4th Pacific Symposium on Biocomputing (PSB '99), January 1999: proceedings. – Maui, Hawaii, USA, 1999. – P. 41-52.
6. Chen T. Modeling gene expression with differential equations / T. Chen, H. L. He, G. M. Church // 4th Pacific Symposium on Biocomputing (PSB '99), January 1999: proceedings. – Big Island of Hawaii, Hawaii, USA, 1999. – P. 29-40.
7. Schafer J. An empirical Bayes approach to inferring large-scale gene association networks / J. Schafer, K. Strimmer // Bioinformatics. – 2005. – Vol. 21(6). – P. 754-764.
8. Friedman N. Using Bayesian networks to analyze expression data / N. Friedman, M. Linial, I. Nachman, D. Pe'er // Journal of Computational Biology. – 2000. – Vol. 7(3-4). – P. 601-620.
9. Hache H. Reconstruction and validation of gene regulatory networks with neural networks / H. Hache, C. Wierling, H. Lehrach [et al.] // 2nd Foundations of Systems Biology in Engineering Conference (FOSBE '07), September 2007: proceedings. – Stuttgart, Germany, 2007. – P. 319-324.
10. Фефелов А.А. Гибридный подход при реконструкции генных регуляторных сетей / А.А. Фефелов, В.И. Литвиненко, М.А. Таиф, И.А. Лурье // Управляющие системы и машины. – Киев, 2017. – № 3 – С. 63-72.
11. Rangel C. Modeling T-cell activation using gene expression profiling and state-space models / C. Rangel, J. Angus, Z. Ghahramani [et al.] // Bioinformatics. – 2004. – Vol. 20(9). – P. 1361-1372.
12. Storn R. Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces / R. Storn, K.V. Price // Journal of Global Optimization. – 1997. – Vol. 11(4). – P. 341-359.
13. Storn R. Minimizing the real function of the ICEC'96 contest by differential evolution / R. Storn, K. Price // IEEE International Conference on Evolutionary Computation, May 1996: proceedings. – Nagoya, Japan, 1996. – P. 842-844.
14. DeRisi J.L. Exploring the metabolic and genetic control of gene expression on a genomic scale / J.L. DeRisi, V.R. Lyer, P.O. Brown // Science. – 1997. – Vol. 278. – P. 680-686.

15. Butte A. J. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements / A. J. Butte, I. S. Kohane // Pacific Symposium on Biocomputing (PSB '00), January 2000: proceedings. – Maui, Hawaii, USA, 2000. – P. 415-426.
16. Friedman N. Using Bayesian networks to analyze expression data / N. Friedman, M. Linial, I. Nachman, D. Pe'er // Journal of Computational Biology. – 2000. – Vol. 7(3-4). – P. 601-620.
17. Yu J. Advances to Bayesian network inference for generating causal networks from observational biological data / J. Yu, V. A. Smith, P. P. Wang [et al.] // Bioinformatics. – 2004. – Vol. 20(18). – P. 3594-3603.
18. Gardner T.S. Inferring genetic networks and identifying compound mode of action via expression profiling / T.S. Gardner, D. di Bernardo, D. Lorenz [et al.] // Science. – 2003. – Vol. 301. – P. 102-105.
19. Chen T. Modeling gene expression with differential equations / T. Chen, H. L. He, G. M. Church // 4th Pacific Symposium on Biocomputing (PSB '99), January 1999: proceedings. – Big Island of Hawaii, Hawaii, USA, 1999. – P. 29-40.
20. Savageau M. A. Introduction to S-systems and the underlying power-law formalism / M.A. Savageau // Mathematical and Computer Modelling. – 1988. – Vol. 11. – P. 546-551.
21. Burnet F.M. A modification of jerne's theory of antibody production using the concept of clonal selection / F.M. Burnet // CA: a cancer journal for clinicians. – 1976. – Vol. 26(2). – P. 119-121.
22. De Castro L.N. Learning and optimization using the clonal selection principle / L.N. De Castro, F.J. Von Zuben // IEEE Transactions on Evolutionary Computation. – 2002. – Vol. 6(3). – P. 239-251.
23. Фефелов А.А. Параметрическая идентификация S-системы с применением модифицированного алгоритма клонального отбора / А.А. Фефелов, В.И. Литвиненко, М.А. Таиф, И.А. Лурье // Управляющие системы и машины. – Киев, 2017. – № 5 – С. 43-53.
24. Фефелов А.А. Реконструкция S-системы гибридным алгоритмом клонального отбора и дифференциальной эволюции / А.А. Фефелов, В.И. Литвиненко, М.А. Таиф, М.А. Вороненко // Управляющие системы и машины. – Киев, 2017. – № 6 – С. 43-53.