

活体生物计算模型的研究进展及展望

刘向荣^{1),2)} 赵东明^{1),2)} 郝方^{1),2)} 李菲^{1),2)}

¹⁾(北京大学信息科学技术学院 北京 100871)

²⁾(北京大学高可信软件技术教育部重点实验室 北京 100871)

摘 要 活体生物计算模型是基于生物体内各种生化分子以特定的形式互相协作、处理信息的能力而出现的一种新的计算模型. 由于其计算组成部件是直接镶嵌在生物活体里面, 并且显示具有一定的计算能力, 这可以使人们深入研究生物体信息处理能力以及获得对这种能力的有效控制. 该文介绍了近几年几类体内生物计算模型, 用于求解 NP 完全问题、基因逻辑电路、分子自动机研究状况, 并对未来的发展方向进行了展望.

关键词 生物计算; 体内; NP 问题; 基因电路; 分子自动机

中图法分类号 TP301

Research Advances and Prospect of Biomolecular Computing Models in Vivo

LIU Xiang-Rong^{1),2)} ZHAO Dong-Ming^{1),2)} XI Fang^{1),2)} LI Fei^{1),2)}

¹⁾(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

²⁾(Key Laboratory of High Confidence Software Technologies of Ministry of Education, Peking University, Beijing 100871)

Abstract Biomolecular computing models in vivo are an emerging computing model inspired from the biological phenomena that the biochemical molecular in living perform computation, communications, and signal processing collaboratively. This paper reviews some recent DNA computing models which are proposed to work at the cellular level for NP complete problem models, gene logic circuits and biomolecular automata. The future research directions on in vivo calculations are also pointed out.

Keywords bimolecular computing; in vivo; NP problem; gene circuit; biomolecular automata

1 Introduction

Computation is a map or a transform process of symbol strings based on the rules. Symbol is the special physics state. It can be the electronic circuit behavior based on transistor technology. The machine is successfully realized by digital computer, which is a finite state machine, with a vast number of states. And the digital computers excel in many areas of applications. If the interaction between the biomoleculars can be viewed as a finite number of components and the components can take on a few states, the particular biochemical reaction is bio-

molecular computing.

A physical computation in a digital computer evolves over time. Information is stored in registers and other media, while information is processed by using digital circuits. In biomolecular computing, information is stored by biomolecules and processing of information takes place by manipulating biomolecules. The concept of biomolecular computing was theoretically discussed by Head in 1987^[1], but Adleman in 1994 was the first to solve a small instance of travelling salesman problem with DNA^[2], he was the first to demonstrate by a DNA experiment that biomolecular

computations are feasible. Since it has been shown that they have powerful computational capability and potential capability in solving computational hard problems, more and more people begin to interest in this area. In 2002, Adleman's group solved a 20-variable instance of the 3-satisfiability problem by DNA computing, which is the largest instance solved by non-electronic computer as yet^[3]. Researchers realized some of the obstacles related to this incipient technology soon thereafter, efficiency and precision of biochemical reaction, and exponential explosion of solution space with respect to problem size.

Therefore, progress in biomolecular computing will depend on both novel computing concepts and biological operation technique. It was proved that some models in biomolecular have the same computing capability as Turing machines. The goal of researchers is to find biomolecular computing paradigms capable more than Turing machines, or a new application platform.

2 Biomolecular Computing in Vivo

The architecture of gene regulatory networks is reminiscent of electronic circuits. Increasing knowledge on how cell behavior is shaped by the complex regulatory transcription networks of different sets of genes, which show response characteristics as electric circuits. Cells are attractive for many programmed applications for their miniature scale, self reproduction, and capacity to manufacture biochemical products. Applications include nano-fabrication, embedded intelligence in materials, sensor arrays, patterned biomaterial manufacturing, improved pharmaceutical synthesis, and programmed therapeutics. For such applications, precise and programmed control of gene activity

can be accomplished by incorporating synthetic biochemical logic circuits into the cells.

Many proteins in living cells appear to have as their primary function the transfer and processing of information, rather than the chemical transformation of metabolic intermediates or the building of cellular structures. Such proteins are functionally linked through allosteric or other mechanisms into biochemical circuits that perform a variety of simple computational tasks including amplification, integration and information storage^[4].

2.1 NP Problems Models

Bacterial activity illustrates other common features of biomolecular computing in vivo, such as their ability to integrate multiple inputs. Plasmid is a circular, double-stranded DNA molecular, which contain an origin for replication and allows the production in bacteria. A method of computing using DNA plasmids is introduced by Head in 2000^[5]. It is illustrated by reporting a laboratory computation of an instance of the NP-complete algorithmic problem of computing the cardinal number of a maximal independent subset of the vertex set of a graph. This computational plasmid contains a specially inserted series of DNA sequence segments, each of which is bordered by a characteristic pair of restriction enzyme sites. By applying a scheme of enzymatic treatments to the computational plasmids, modified plasmids were generated from which the solution of the computational problem was selected. In the computation, let P be a plasmid, k be a positive integer and s_1, s_2, \dots, s_k be k pair wised non-overlapping sub segments of P . For each i , the nucleotide sequence of s_i occurs nowhere else in the plasmid P . Subsegments chosen in this way will be called 'stations' of plasmid. Fig. 1 gives a sketch of the computational plasmid.

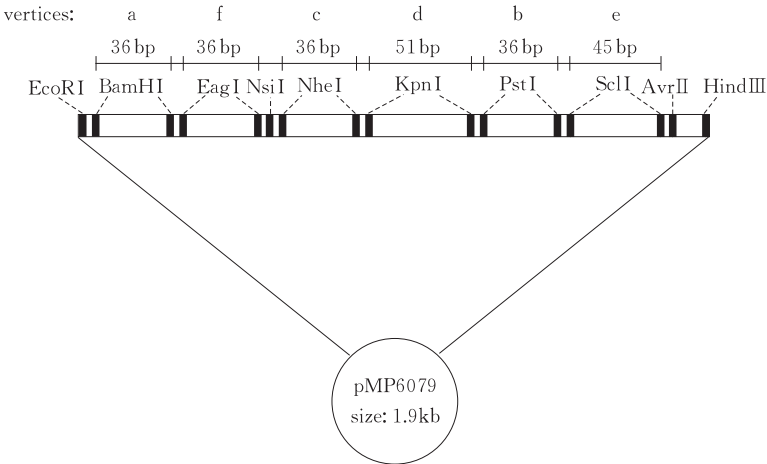


Fig. 1 The plasmid used as template for a six-station problem^[5]

The DNA plasmid is constructed that segment of 297 base pairs was inserted before recircularizing to form the final plasmid. The six stations are all included in the specially designed insert. A unique restriction enzyme (EcoRI, BamHI, EagI ect.) is associated with each of the stations. Each station is bounded by a pair of sites for its associated restriction enzyme. In computations, one is free to consider the initial state of each station of the computational plasmid to represent either the bit 0 or the bit 1, whichever is convenient.

Plasmids can serve to perform computation at the molecular level^[6]. Plasmid computing has been successfully applied to small test instance of several computationally hard optimization problems. On the basis of Head, the DNA solution to the Maximum Weight Clique Problem of an undirected graph^[7] and maximum matching^[8] based on the plasmid were presented. Henkel extended plasmid computing with protein expression by the construction of the whole computing region of a plasmid as part of an open reading frame^[9]. After library generation, the library was expressed into a protein representation, and this was in turn used to select a solution. A potential advantage of this translation of the solution into protein is smaller molecules and consequently higher information densities. In 2007, several instances of knapsack problems using plasmids were presented^[10]. These problems ask for the best way to pack a knapsack of limited capacity with items of different size and weight. DNA computing seems very well suited for this family of problems, as both the encoding and the algorithm are relatively straightforward: Formal size or weight can be linked directly to physical properties of DNA.

2.2 Gene Logic Circuits

The genetic and biochemical networks which underlie such things as homeostasis in metabolism and the developmental programs of living cells, must withstand considerable variations and random perturbations of biochemical parameters. These occur as transient changes in, for example, transcription, translation, and RNA and protein degradation. Genetic circuits are collections of basic elements that interact to produce a particular behavior. By constructing biochemical logic circuits and embedding them in cells, one can extend or modify the behavior of cells. To date, several small synthetic gene networks have been built that accomplish specific genetic regulatory functions in vivo.

The autoregulatory is a repressor regulates its

own production to reduce noise in gene expression. Savageau proposed but not demonstrated that the negative feedback loops in gene circuits provide stability^[11].

Becskei and Serrano have designed and constructed simple gene circuits consisting of a regulator and transcriptional repressor modules in *Escherichia coli* and show the gain of stability produced by negative feedback^[12]. To test the role of negative feedback in the stability of gene networks, they first designed simple gene circuits based on simple control systems. To construct the autoregulatory system, the tetracycline repressor (TetR) of the transposon Tn10 was fused to the green fluorescent protein (EGFP) (TetR-EGFP) and placed downstream of the lambda promoter containing two tetracycline operators (Fig. 2). As controls, the unregulated counterparts were obtained by slight modification of this system. In this way, differences among the systems could be neglected, while the feedback was eliminated.

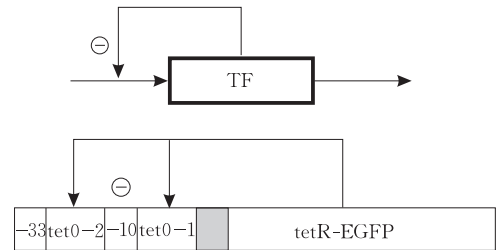


Fig. 2 Gene circuits^[12]

Gardner presented the construction of a genetic toggle switch, a synthetic, bistable gene-regulatory network, in *Escherichia coli* and provided a simple theory that predicts the conditions necessary for bistability^[13]. The toggle is constructed from any two repressible promoters arranged in a mutually inhibitory network. It is flipped between stable states using transient chemical or thermal induction and exhibits a nearly ideal switching threshold. As a practical device, the toggle switch forms a synthetic, addressable cellular memory unit and has implications for biotechnology, biomolecular computing and gene therapy.

The toggle switch is composed of two repressors and two constitutive promoters (Fig. 3). Each promoter is inhibited by the repressor that is transcribed by the opposing promoter. The toggle switch requires the fewest genes and cis-regulatory elements to achieve robust bistable behavior. The bistability of the toggle arises from the mutually inhibitory arrangement of the repressor genes. In the absence of inducers, two stable states are pos-

sible: one in which promoter 1 transcribes repressor 2, and one in which promoter 2 transcribes repressor 1. Switching is accomplished by transiently introducing an inducer of the currently active repressor. The inducer permits the opposing repressor to be maximally transcribed until it stably represses the originally active promoter.

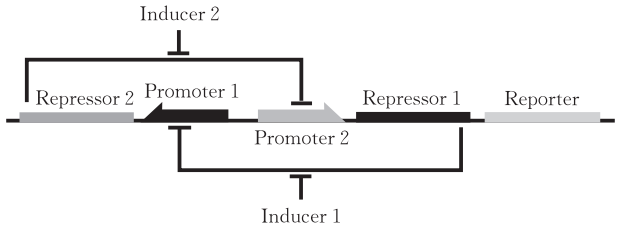


Fig. 3 Toggle switch design^[13]

Elowitz present the design and construction of a synthetic network to implement a particular function, three transcriptional repressor systems that are not part of any natural biological clock to build an oscillating network in *Escherichia coli*^[14]. The network periodically induces the synthesis of green fluorescent protein as readout of its state in individual cells. The resulting oscillations, with typical periods of hours, are slower than the cell-division cycle, so the state of the oscillator has to be transmitted from generation to generation. This artificial clock displays noisy behavior, possibly because of stochastic fluctuations of its components. Such rational network design may lead both to the engineering of new cellular behaviors and to an improved understanding of naturally occurring networks.

In the network shown in Fig. 4, the first repressor protein, lacI from *E. coli*, inhibits the transcription of the second repressor gene, tetR from the tetracycline-resistance transposon Tn10, whose protein product in turn inhibits the expression of a third gene, cI from λ phage. Finally, cI inhibits lacI expression, completing the cycle. That such a negative feedback loop can lead to temporal oscillations in the concentrations of each of its components can be seen from a simple model of transcriptional regulation, which can be used to design the repressilator and study its possible behaviors. In this model, the action of the network depends on several factors, including the dependence of transcription rate on repressor concentration, the translation rate, and the decay rates of the protein and messenger RNA. Depending on the values of these parameters, at least two types of solutions are possible: The system may converge toward a stable steady state, or the steady state

may become unstable, leading to sustained limit-cycle oscillations.

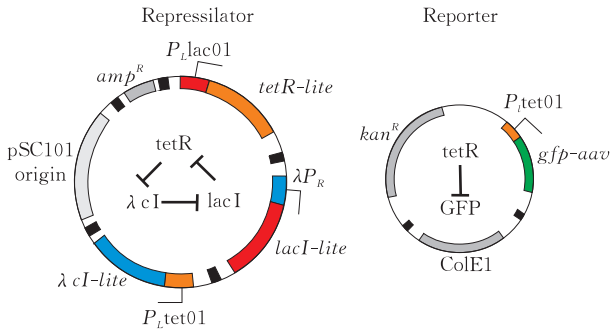


Fig. 4 Construction and design of the repressilator^[14]

These gene circuits consisted of 1~3 repression systems and constructed the simple digital logic circuits. Increasing with more circuits cascade, there is more complexity and difficulty in the construction of gene circuits. The key reason is that the gene circuits run in the living cell and many parameters of biochemical reaction in cellular isn't clear yet. Only several known gene network can be used in the circuit design. Another reason is that the gene circuit will influence the normal physiological activity of cell. Similar to the digital logic circuits, gene circuit should be new design principle to overcome the shortcoming and challenge for the design and construction of more sophisticated genetic circuitry in the future.

A combined rational and evolutionary design strategy for constructing genetic regulatory circuits is proposed by Yokobayashi et al.^[15] The approach allows the engineer to fine-tune the biochemical parameters of the networks experimentally in vivo. By applying directed evolution to genes comprising a simple genetic circuit, they demonstrate that a nonfunctional circuit containing improperly matched components can evolve rapidly into a functional one. In the process, they generated a library of genetic devices with a range of behaviors that can be used to construct more complex circuits.

Weiss presented a genetic component library and a gene circuit design methodology for assembling these components into compound circuits^[16]. The main challenge in gene circuit design lies in selecting well-matched genetic components that when coupled, reliably produce the desired behavior. They used simulation tools to guide circuit design, a process that consists of selecting the appropriate components and genetically modifying existing components until the desired behavior is achieved. In addition to such rational design, they also employ directed evolution to optimize genetic circuit

behavior. The integration of all the above capabilities in future synthetic gene networks will enable cells to perform sophisticated digital and analog computation, both as individual entities and as part of larger cell communities. This engineering discipline and its associated tools will advance the capabilities of genetic engineering, and allow us to harness cells for a myriad of applications not previously achievable.

Hwa et al. explored theoretically the potentials and limitations of combinatorial signal integration at the level of cisregulatory transcription control^[17]. Their analysis suggested that many complex transcription-control functions of the type encountered in higher eukaryotes are already implementable within the much simpler bacterial transcription system. Using a quantitative model of bacterial transcription and invoking only specific protein—DNA interaction and weak glue-like interaction between regulatory proteins, they showed explicit schemes to implement regulatory logic functions of increasing complexity by appropriately selecting the strengths and arranging the relative positions of the relevant protein-binding DNA sequences in the cis-regulatory region. The architectures emerged are naturally modular and evolvable. Their results suggested that the transcription regulatory apparatus is a “programmable” computing machine, belonging formally to the class of Boltzmann machines. Crucial to their results is the ability to regulate gene expression at a distance. In bacteria, this can be achieved for isolated genes via DNA looping controlled by the dimerization of DNA-bound proteins. However, if adopted extensively in the genome, long-distance interaction can cause unintentional intergenic cross talk, a detrimental side effect difficult to be overcome by the known bacterial transcription-regulation systems. This may be a key factor limiting the genome-wide adoption of complex transcription control in bacteria. Implications of their findings for combinatorial transcription control in eukaryotes are discussed.

In 2004, Kramer pioneerly presented a variety of different two- and three-input biologic gates in mammalian cells by combining several compatible heterologous gene control units responsive to tetracycline, streptogramin, macrolide, and butyrolactones^[18]. In combination with modern transduction technologies, the biologic gates could serve as versatile tools for regulated gene expression and as building blocks for complex artificial gene regulatory networks for applications in gene therapy, tis-

sue engineering, and biotechnology.

In 2007, Weiss and Benenson used RNA interference (RNAi) in human kidney cells to construct a molecular computing core that implements general Boolean logic to make decisions based on endogenous molecular inputs^[19]. The state of an endogenous input is encoded by the presence or absence of mediator small interfering RNAs (siRNAs). The encoding rules, combined with a specific arrangement of the siRNA targets in a synthetic gene network, allow direct evaluation of any Boolean expression in standard forms using siRNAs and indirect evaluation using endogenous inputs. They demonstrated direct evaluation of expressions with up to five logic variables. Implementation of the encoding rules through sensory up- and down-regulatory links between the inputs and siRNA mediators will allow arbitrary Boolean decision-making using these inputs.

2.3 Biomolecular Automata

Finite state automata operating autonomously at the molecular scale can be sued conceptually for applications in the living cell.

Shapiro built a small finite state automaton from DNA strands and enzymes^[20]. This automaton uses anti-sense technology to carry out molecular diagnosis and therapy. The molecular computer at least in vitro logically analyses the levels of messenger RNA species, and in response produces a molecule capable of affecting levels of gene expression. The computer operates at a concentration of close to a trillion computers per microlitre and consists of three programmable modules: a computation module, that is, a stochastic molecular automaton; an input module, by which specific mRNA levels or point mutations regulate software molecule concentrations, and enhance automaton transition probabilities; and an output module, capable of controlled release of a short single-stranded DNA molecule. This approach might be applied in vivo to biochemical sensing, genetic engineering and even medical diagnosis and treatment.

The first autonomous finite state machine working in a living cell was proposed by Sakakibara and Coworkers^[21]. This approach is based on the length-encoding automaton model and was tested in *E. colicells*. The success of a computation in this model crucially depends on the concentration of available tRNA molecules. The experiment showed that a computation by a single *E. colicell* is not effective and accurate, while a colony of *E. colicells* provides more reliable computations. Since bacterial cells can multiply to over a million cells

overnight, these in vivo computation might offer a massive amount of parallelism.

Computational genes provide another type of finite state automata, which autonomously operate at the molecular scale^[22]. Computational genes invented by the authors are able to detect and correct aberrant molecular phenotype given by mutated genetic transcripts.

3 Conclusion

Biomolecular computing models for understanding and manipulating cellular behavior are generally invaluable. However, several issues need to be addressed before cellular DNA models can be implemented in vivo. First, the DNA material must be safely internalized into the cell, specifically into the nucleus. Second, the DNA complexes should have low immunogenicity to guarantee their integrity in the cell and their resistance to cellular nucleases. Third, similar to other drugs, DNA complexes could cause non-specific and toxic side effects. Undoubtedly, progress in gene knowledge and technology would also result in a direct benefit to cellular DNA computing.

References

- [1] Head T. Formal language theory and DNA: An analysis of the generative capacity of specific recombination behaviors. *Bulletin of Mathematical Biology*, 1987, 47: 737-759
- [2] Adleman L M. Molecular computation of solution to combinatorial problems. *Science*, 1994, 266: 1021-1024
- [3] Braich R S, Chelyapov N, Johnson C, Rothmund P W K, Adleman L M. Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, 2002, 296: 499-502
- [4] Bray D. Protein molecules as computational elements in living cells. *Nature*, 1995, 376: 307-312
- [5] Head T, Rozenberg G, Bladergroen R B, Breek C K D, Lommerse P H M, Spaink H P. Computing with DNA by operating on plasmids. *BioSystems*, 2000, 57: 87-93
- [6] Paun G, Rozenberg G, Salomaa A. *DNA Computing—New Computing Paradigms*. Berlin: Springer-Verlag, 2005
- [7] Gao Lin, Ma Run-Nian, Xu Jin. The molecular algorithm of the matching problem based on plasmid DNA. *Progress in Biochemistry and Biophysics*, 2002, 29(5): 820-823(in Chinese)
- [8] Ma Run-Nian, Zhang Qiang, Gao Lin, Xu Jin. Using DNA to solve the maximum weight clique of graphs. *Acta Electronica Sinica*, 2004, 32(1): 1-16(in Chinese)
- [9] Henkel C V, Bladergroen R S, Balog C I A, Deelder A M, Head T, Rozenberg G, Spaink H P. Protein output for DNA computing. *Natural Computing*, 2005, 4: 1-10
- [10] Henkel C V, Back T, Kok J N, Rozenberg G, Spaink H P. DNA computing of solutions to knapsack problems. *BioSystems*, 2007, 88: 156-162
- [11] Savageau M A. Comparison of classical and autogenous systems of regulation in inducible operons. *Nature*, 1974, 252: 546-549
- [12] Becskei A, Serrano L. Engineering stability in gene networks by autoregulation. *Nature*, 2000, 405: 590-593
- [13] Gardner T S, Cantor C R, Collins J J. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 2000, 403: 339-342
- [14] Elowitz M B, Leibler S. A synthetic oscillatory network of transcriptional regulators. *Nature*, 2000, 403: 335-338
- [15] Yokobayashi Y, Weiss R, Arnold F H. Directed evolution of a genetic circuit. *Proceedings of the National Academy of Science*, 2002, 99: 16587-16591
- [16] Weiss R, Basu S, Hooshangi S et al. Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing*, 2003, 6(2): 1-40
- [17] Buchler N E, Gerland U, Hwa T. On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences*, 2003, 100: 5136-5141
- [18] Kramer B P, Fischer C, Fussenegger M. Biologic gates enable logical transcription control in mammalian cells. *Biotechnology and Bioengineering*, 2004, 87: 478-484
- [19] Keller R, Leonidas B, Rohan M et al. A universal RNAi-based logic evaluator that operates in mammalian cells. *Nature Biotechnology*, 2007, 25(7): 795-801
- [20] Benenson Y, Gil B, Ben-Dor U, Adar R et al. An autonomous molecular computer for logical control of gene expression. *Nature*, 2004, 429: 423-429
- [21] Nakagawa H, Sakamoto K, Sakakibara Y. Development of an in vivo computer based on *Escherichia coli*//*Lecture Notes in Computer Science* 2892, Berlin: Springer, 2006: 203-212
- [22] Martinez-Perez I, Ignatova Z, Gong Z, Zimmermann K H. Computational genes: A tool for molecular diagnosis and therapy of aberrant mutational phenotype. *BMC Bioinformatics*, 2007, 8: 365

LIU Xiang-Rong, born in 1978, Ph.D.. His main research interests focus on biomolecular computing in vivo.



ZHAO Dong-Ming, born in 1977, Ph.D.. His main research interests focus on DNA computing.

XI Fang, born in 1986, Ph.D. candidate. His main research interests focus on DNA computing device.

LI Fei, born in 1983, Ph.D. candidate. Her main research interests focus on theories of bio-inspired computing.