

Um Modelo Generalizado de Computação Molecular baseado em Regulação de Circuitos Gênicos em Plasmídeos

Fernando Pellon de Miranda¹ José Wagner Garcia¹ Luciano Silva^{1,2}

¹Grupo de Computação Líquida, Projeto Cognitus II, CENPES-PETROBRÁS

²Departamento de Ciência da Computação, FCI, Universidade Presbiteriana Mackenzie

e-mail: fmiranda@petrobras.br, automata@uninet.com.br, lucianosilva@mackenzie.com.br

Resumo

Computação Molecular representa um modelo poderoso de computação não-convencional, cujas características principais são não-determinismo e paralelismo, importantes para abordar certas classes de problemas computacionalmente difíceis. Este trabalho apresenta um modelo generalizado de computação molecular via regulação de circuitos gênicos em plasmídeos que permite, em particular, a computação de funções de várias variáveis booleanas.

Palavras-chave: Circuitos Gênicos, Computação baseada em Plasmídeos, Computação Molecular, Computação Não-Standard.

1 Introdução

A Computação Molecular apresenta-se como um novo paradigma de computação, que utiliza processos moleculares para realizar computação. Desde o surgimento do paradigma, através do trabalho pioneiro de Adleman[2], muitos modelos que exploram alternativas computacionais dos ácidos nucleicos RNA e DNA, membranas, plasmídeos e química supra-molecular têm sido propostos. Adicionalmente, vários problemas computacionalmente difíceis tais como o Problema do Circuito Hamiltoniano Orientado ou o Problema da Satisfazibilidade Booleana(SAT) já possuem algoritmos eficientes no contexto molecular. Não-determinismo, paralelismo e, principalmente, baixo consumo de energia são algumas das atrações naturais do ambiente molecular e é de grande interesse computacional o desenvolvimento de modelos e algoritmos que alcancem grandes classes de problemas para o paradigma.

Recentemente, surgiu o conceito de circuito gênico em Computação Molecular, que disponibiliza, a grosso modo, um suporte de *hardware* molecular para operações booleanas básicas como AND, OR e NOT. Várias construções bioquímicas, como os operons, tornaram-se candidatas em potencial

para implementação destes circuitos tanto *in vivo* quanto *in vitro*. Em particular, já existem redes de operons disponíveis para se calcular funções booleanas de duas variáveis a um valor booleano.

Este trabalho apresenta um modelo de computação molecular generalizado, que utiliza seqüências de circuitos gênicos separados por enzimas de restrição e alojados em estruturas de plasmídeos, que permite computações seqüenciais e paralelas. Em particular, é apresentado um resultado que torna possível a computação molecular de funções de várias variáveis booleanas a vários valores booleanos.

2 Bases da Biologia Molecular

Biologia Molecular representa um dos ramos mais progressistas da Biologia Celular, onde as dimensões dos objetos de estudo são menores que 1 μ m. Neste contexto, a abordagem das estruturas é cerceada por ferramentas bioquímicas, físico-químicas e advindas das químicas macromoleculares e coloidais.

Além do meio aquoso, rico em propriedades físico-químicas, os ácidos nucleicos DNA e RNA representam alguns dos principais objetos de interesse da Biologia Molecular. Tais ácidos são macromoléculas formadas a partir do encadeamento de moléculas menores, denominadas aminoácidos que, por sua vez, são formadas por unidades ainda menores chamados nucleotídeos. Um conjunto bastante reduzido de bases—Adenina(A), Timina(T), Citosina(C), Guanina(G) e Uracila(U)—forma o alicerce primordial de construção dos ácidos nucleicos.

O DNA é o *container* fundamental das informações genéticas. Estas informações podem ser propagadas através de um mecanismo de cópia, chamado *replicação* de DNA, ou induzir um processo denominado *transcrição*, que consiste na formação de seqüências de RNA a partir de seqüências-molde de DNA. A partir destas seqüências de RNA, ocorre o processo da *tradução*, que

um modelo específico de plasmídeo foi construído e era composto de determinadas seqüências de DNA (*stations*), delimitadas por pares de sítios enzimáticos de restrição. Após procedimentos bioquímicos de cortes e junções, novos plasmídeos eram formados, dos quais a solução para o problema foi obtida. As seqüências de DNA utilizadas por Head tinham o propósito exclusivo de representar informações do grafo e não expressavam poder computacional. O modelo do presente trabalho propõe algo mais poderoso: utilizar como seqüências operons generalizados, separados por sítios enzimáticos de restrição.

4 Operons Generalizados

O modelo *operon* foi proposto por Jacob e Monod[9] e formaliza os mecanismos básicos de regulação de expressão gênica. Um operon simples é formado por uma seqüência específica composta de um promotor, um operador e um gene que define a expressão. O promotor tem a função de indicar o início do processo de transcrição, enquanto que o operador está vinculado aos mecanismos de regulação. Normalmente, os três componentes do operon são arranjados conforme o esquema abaixo:

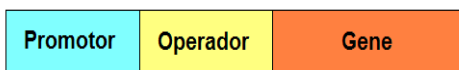


Figura 2: Modelo básico de um operon

A expressão do gene é realizada por mediação de uma enzima denominada *RNA polimerase* (RNAP). Esta enzima acopla-se ao promotor e, se o operador não estiver inibido, realiza a transcrição do gene para um *RNA mensageiro* que, posteriormente, será traduzido para uma proteína específica. Um operon pode funcionar como uma porta lógica devido à atuação de indutores e repressores. O indutor normalmente funciona como um controle positivo, isto é, na sua presença o gene é expresso. Já o repressor, normalmente inibe a expressão do gene, pois liga-se ao operador impedindo a transcrição do gene, funcionando como um controle negativo. Os controles positivo e negativo podem se inverter em alguns tipos de operons.

A tabela abaixo ilustra um exemplo de operon, onde está representada a ação de indutores e repressores na expressão gênica:

Repressor(R)	Indutor(I)	Expressão Gênica
Não	Não	Sim
Não	Sim	Sim
Sim	Não	Não
Sim	Sim	Sim

Não é difícil observar que a tabela anterior calcula a função booleana IMPLICA, ou seja, define uma porta lógica, denominada de *circuito gênico*. De um ponto de vista mais formal, um operon pode ser visto como uma função booleana de duas variáveis booleanas $\sigma : Bool \times Bool \rightarrow Bool$, onde $Bool = \{0, 1\}$. Matematicamente, existem oito operons possíveis com esta visão, mas nem todas estas combinações formais possuem um mapeamento direto para um único operon. Por outro lado, Weiss[19] mostrou um resultado bastante forte acerca de circuitos gênicos:

Teorema 1 [Weiss] *Toda função de duas variáveis booleanas a uma variável booleana pode ser computada através de circuitos gênicos.*

Este resultado é bastante importante, pois muitos problemas computacionais podem ser reduzidos a cálculos envolvendo funções booleanas. No final deste trabalho, o resultado de Weiss será estendido para funções com várias variáveis booleanas a várias variáveis booleanas.

A partir da definição básica de operon, constrói-se a noção de operon generalizado:

Definição 1 *Um operon generalizado de tipo (r, k) , denotado por $\sigma_{r,k}$, com r níveis de regulação e k genes de expressão, é qualquer função booleana $\sigma_{r,k} : Bool^r \rightarrow Bool^k$.*

É fácil observar que existem 2^{r+k} operons generalizados possíveis. Porém, ressalta-se novamente que nem todas as possibilidades são passíveis de implementação bioquímica. Neste modelo generalizado, o operon lac estudado por Jacob e Monod[9] seria um operon de tipo $(2, 3)$, pois existem dois níveis de controle, a lactose(indutor) e o repressor lac, e três genes de expressão (enzimas β -galactosidase(LacZ), permease(LacY) e a transacetilase(LacA)). O operon triptofano (Voet e Voet[16]), outro exemplo importante em mecanismos de regulação, seria um operon do tipo $(2, 5)$, pois é induzido pelo ácido 3- β -indolacrilico, reprimido pelo triptofano e existem cinco genes de expressão.

5 Redes de Simulação

A cada sistema bioquímico, é possível associar uma equação diferencial ordinária. Normalmente, são utilizados os chamados Sistemas S (Voit[18]), que permite descrever mudanças temporais em um sistema bioquímico através de uma diferença de produtórias.

Sejam (X_1, \dots, X_n) e $(X_{n+1}, \dots, X_{n+m})$ os conjuntos de variáveis dependentes e independentes, respectivamente, de um sistema bioquímico. Então, a variação temporal de uma determinada variável

X_i pode ser dada pela seguinte equação diferencial exibida abaixo:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n+m} X_i^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_i^{h_{ij}}$$

A primeira parcela do lado direito da equação correspondente à produção da variável X_i , enquanto que a segunda parcela representa, normalmente, processos de consumo ou degradação. As constantes α_i , g_{ij} , β_i e h_{ij} dependem de cada ambiente bioquímico que está sendo modelado e, ao conjunto formado pelas variáveis X_i e equações $\frac{dX_i}{dt}$, é dada a denominação de Sistema S associado ao processo bioquímico. Por exemplo, o Sistema S parcial abaixo descreve as variações de concentração temporal do repressor e das enzimas β -galactosidase e permease, vinculados ao operon lac (Wong *et al.*[20]):

$$\frac{d[\text{Rep}]}{dt} = V_{\text{Rep}} - (k_{d,\text{Rep}} + \mu)[\text{Rep}]$$

$$\frac{d[\beta\text{Gal}]}{dt} = V_{\beta\text{Gal}} - (k_d + \mu)[\beta\text{Gal}]$$

$$\frac{d[\text{Perm}]}{dt} = V_{\text{Perm}} - (k_d + \mu)[\text{Perm}]$$

Para simular de forma discreta ou contínua um Sistema S, pode-se utilizar as *redes bioquímicas* (Voit[18]). Uma rede bioquímica é um grafo orientado que descreve o fluxo de reações bioquímicas vinculadas a um Sistema S. A figura mostrada a seguir ilustra uma rede bioquímica parcial para o operon lac, construída com a ferramenta Cell Illustrator:

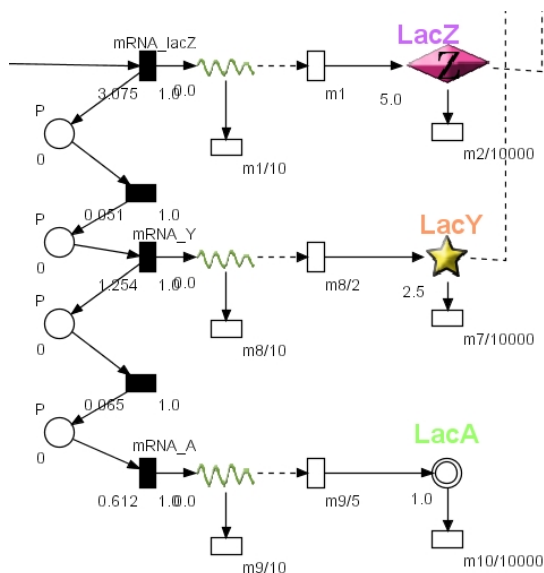


Figura 3: Diagrama parcial para a rede bioquímica vinculada ao operon lac, onde estão destacadas as enzimas LacZ, LacY e LacA.

Os elementos principais da rede ilustrada acima são os círculos e retângulos. Círculos representam

entidades ou variáveis da rede, enquanto que retângulos representam reações bioquímicas, governadas pelas equações diferenciais dos Sistemas S. Cada retângulo pode ter várias entradas (reagentes) e várias saídas (produtos da reação). A simulação de

uma rede bioquímica, como a mostrada anteriormente, normalmente produz uma série de gráficos sobre a evolução temporal das variáveis.

A maneira mais comum de simulação de uma rede bioquímica é via Redes de Petri (Resig[12]), que permitem simulações tanto discretas quanto contínuas. Estas redes permitirão a construção da seguinte definição:

Definição 2 *Uma rede bioquímica computacional generalizada (RBCG) é uma Rede de Petri que simula um operon generalizado, onde os lugares e transições correspondam às variáveis e equações diferenciais, respectivamente, de um Sistema S que descreva o operon.*

É importante ressaltar que esta definição permite vincular, a um mesmo operon, várias RBCG, pois pode-se ter Sistemas S em diferentes resoluções de simulação para o operon. Além disso, as RBCGs formarão um suporte bastante importante para simulação do modelo computacional proposto nas próximas seções.

6 Plasmídeos Generalizados

A unidade computacional proposta neste trabalho é formada por uma seqüência de operons generalizados, separados por enzimas de restrição e organizados numa estrutura circular de plasmídeo. Esta estrutura será denominada de plasmídeo generalizado:

Definição 3 *Um plasmídeo generalizado P de ordem q é uma seqüência circular finita $P = (e_1\sigma_{r_1,k_1}^1 e'_1 e_2\sigma_{r_2,k_2}^2 e'_2 \dots e_q\sigma_{r_q,k_q}^q e'_q)$, onde cada 2-upla (e_i, e'_i) representa as enzimas de restrição que limitam o operon generalizado σ_{r_i,k_i}^i .*

Os pares de enzimas (e_i, e'_i) podem conter, eventualmente, enzimas de naturezas diferentes e têm o objetivo de possibilitar a seleção de um operon generalizado específico σ_{r_i,k_i}^i , ou seja, permitem operar seletivamente nos circuitos gênicos imersos no plasmídeo.

Embora o conceito de plasmídeo generalizado seja formalmente consistente, seu mapeamento para o ambiente bioquímico pode apresentar alguns problemas. Por exemplo, grandes plasmídeos podem apresentar instabilidades moleculares durante as reações. Isto pode ser contornado dividindo-se os plasmídeos em sub-plasmídeos, isto é, plasmídeos

de ordem menor. A simulação direta de um plasmídeo *in vivo* pode, também, apresentar problemas de rejeição pelo hospedeiro do plasmídeo. Porém, a simulação não precisa ser necessariamente efetuada *in vivo* e pode recorrer aos mecanismos de tubosteste proposto por Adleman[2], em meio líquido bioquimicamente controlado.

A cada plasmídeo generalizado $P = (e_1\sigma_{r_1,k_1}^1 e'_1 e_2\sigma_{r_2,k_2}^2 e'_2 \dots e_q\sigma_{r_q,k_q}^q e'_q)$, associa-se uma seqüência de RBCGs $R_P = (R_1, R_2, \dots, R_q)$, onde cada R_i corresponde à rede de simulação do operon generalizado σ_{r_i,k_i}^i . O objetivo da seqüência de redes R_P é fornecer um ambiente de simulação, *in silico*, para os processos computacionais.

7 Processo Computacional

Com os circuitos gênicos organizados nos plasmídeos generalizados, existem duas formas de se operar computacionalmente com tais plasmídeos: sequencial e paralelamente. Em ambos os casos, existem três procedimentos básicos:

- seleção dos circuitos gênicos;
- re-circularização do plasmídeo;
- regulação e expressão gênicas, colocados por conveniência num único procedimento.

A seleção dos circuitos gênicos é realizada através de procedimentos enzimáticos de recorte e remoção de um circuito gênico específico. Estes procedimentos poderiam utilizar, por exemplo, os mecanismos expostos por Vieira e Messing[17], que construíram um plasmídeo bastante controlável, com diferentes enzimas de marcação de segmentos e origens de replicação. Este procedimentos deixariam vários circuitos gênicos imersos no ambiente líquido e, também, quebrariam a estrutura circular do plasmídeo. Eventualmente, os plasmídeos podem ser removidos temporariamente do ambiente líquido para não sofrer interferência dos mecanismos regulatórios envolvidos nos circuitos removidos.

Antes da regulação do circuitos, é necessário reconstruir a estrutura circular do plasmídeo, que é realizada através de enzimas especiais denominadas *ligases*. É importante observar que estas enzimas não devem participar dos mecanismos de regulação dos operons removidos nem daqueles ainda presentes no plasmídeo.

Finalmente, os circuitos removidos e imersos no meio líquido podem ser induzidos ou inibidos através dos seus mecanismos regulatórios específicos. Os produtos expressos nestes circuitos podem ser observados ou participar de mecanismos regulatórios de outros circuitos gênicos.

Estes três procedimentos são executados de forma ciclica até que os plasmídeos não contenham mais circuitos gênicos. Como os tamanhos dos

plasmídeos são finitos, garante-se *a priori* que os ciclos sempre terminam. Cada ciclo destes três procedimentos (seleção, re-circularização e regulação/expressão) configura um passo computacional do modelo.

A característica sequencial ou paralela de cada passo computacional depende das relações existentes entre os circuitos selecionados em cada passo. Se o produto de um determinado circuito gênico não participa do mecanismo regulatório de outro circuito, então a execução dos dois circuitos é dita paralela. Porém, se o produto de um deles interferir na regulação do outro, então a execução dos dois circuitos é dita sequencial. Ainda mais, na execução sequencial é possível uma relação circular entre os circuitos. Adicionalmente, é possível também uma dependência sequencial entre dois passos computacionais, ou seja, produtos resultantes de determinados circuitos de um passo podem ser regulatórios dos circuitos selecionados nos próximos passos.

A simulação do processo computacional descrito anteriormente pode ser facilmente realizada *in silico* através de ligações das redes R_p de forma sequencial ou paralela. Inicialmente, determina-se quantos passos terá o processo computacional, seleciona-se as redes de cada passo (correspondentes aos circuitos escolhidos em cada passo), alocando-as de forma paralela ou sequencial e, posteriormente, realizando as ligações correspondentes às dependências entre os passos.

Neste ponto, existe uma indagação natural: o que pode ser computado neste modelo? Inicialmente, objetos computacionalmente importantes são as funções de várias variáveis booleanas a vários valores booleanos e que podem ser computados pelo modelo proposto através do resultado a seguir, cuja prova não é realizada neste artigo:

Teorema 2 *Toda função de várias variáveis booleanas a vários valores booleanos pode ser decomposta como uma seqüência finita de disjunções finitas de funções de duas variáveis booleanas a um valor booleano.*

Não é difícil observar que, para cada função de duas variáveis, o resultado do Teorema 1 garante que existe um circuito gênico que a computa. Assim, inicialmente constrói-se um plasmídeo generalizado (ou grupos de plasmídeos generalizados) contendo os circuitos gênicos. Para computar as disjunções, são necessários procedimentos sequenciais e paralelos. Ainda mais, como existe um número finito de disjunções e cada uma tem tamanho também finito, é possível organizar a sua computação em um número finito de passos. É importante observar que, quando existir uma dependência entre dois circuitos, o produto(ou produtos) de um seja um dos mecanismos regulatórios do outro.

8 Conclusões e Trabalhos Futuros

Computação Molecular representa um novo paradigma de computação, com ramificações para várias áreas como computação com DNA, RNA, membranas e plasmídeos. A área de circuitos gênicos mapeados em DNA, RNA ou plasmídeos tem recebido muitas contribuições recentes e apresenta-se com uma tecnologia promissora efetiva para realização de Computação Molecular em ambientes líquidos.

Este trabalho apresentou um modelo de Computação Molecular generalizado baseado na utilização de seqüências finitas de circuitos gênicos, separados por enzimas de restrição e organizados em estruturas de plasmídeos. Como resultado inicial, o modelo permitiu estender o poder computacional dos plasmídeos para funções de várias variáveis booleanas a vários valores booleanos.

Trabalhos futuros irão investigar os limites computacionais do modelo proposto, com busca em aplicações a problemas NP-Completo e NP-Difíceis. Adicionalmente, um protocolo mais efetivo para mapeamento em ambientes líquidos (*in vivo* e *in vitro*) ainda necessita ser desenvolvido, assim como uma biblioteca de circuitos gênicos mais extensa que a proposta por Weiss, para simplificar o processo de mapeamento.

Referências

- [1] R. Adar, Y. Benenson, G. Linshiz, A. Rosner, N. Tishby e E. Shapiro, Stochastic computing with biomolecular automata, *PNAS*, 101(27), pp. 9960 - 9965, 2004.
- [2] L. Adleman, Molecular computation of solutions to combinatorial problems, *Science*, 266, pp. 1021–1024, 1994.
- [3] F. Bernardini, M. Gheorghe, N. Krasnogor e G. Terrazas, Membrane computing. Current results and future problems, *New Computational Paradigms. First Conf. on Computability in Europe*, CiE2005, Amsterdam (S. Barry Cooper, B. Lowe, L. Torenvliet, eds.), LNCS 3536, pp. 49–53, Springer, Berlin, 2005.
- [4] D. Faulhammer, A.R. Cukras, R.J. Lipton e L. F. Landweber, Molecular Computation: RNA Solutions to Chess Problems, *Proc. Natl. Acad. Sci. USA*, 97, pp. 1385–1389, 2000.
- [5] F. Freund, R. Freund e M. Oswald, Splicing test tube systems and their relation to splicing membrane systems, *Aspects of Molecular Computing*, Lecture Notes in Computer Science, pp. 139–151, Springer–Verlag, Berlin, 2004.
- [6] M.R. Garey e D.S. Johnson, "Computers and Intractability", Freeman, New York, 1979.
- [7] M. Hagiya, Perspectives on molecular computing, *New Generat. Comput.*, 17, pp. 131–151, 1999.
- [8] T. Head, G. Rosenberg, R.S. Bladergroen, C.K.D Breek, P.H.M. Lommerse e H.P. Spaink, Computing with DNA by operating on plasmids, *BioSystems*, 57, pp. 87–93, 2000.
- [9] F. Jacob e J. Monod, on the regulation of gene activity, *Cold Spring Harb. Symp. Quant.*, 26, pp. 193–211, 1961.
- [10] Q. Ouyang, P.D. Kaplan, S. Liu e A. Libchaber, DNA solution of the maximal clique problem, *Science*, 278, pp. 446–449, 1997.
- [11] Gh. Paun, S. Rozenberg e A. Salomaa, "DNA Computing – New Computing Paradigms", Springer–Verlag, Berlin, 1998.
- [12] W. Reisig, "Petri Nets", EATCS Monographs on Theoretical Computer Science, Springer–Verlag, Berlin, 1995.
- [13] H. Rubin e D. Wood, "DNA Based Computers IIP", American Mathematical Society, Providence, Rhode Island, 1999.
- [14] X. Su e L. M. Smith, Demonstration of a universal surface DNA computer, *Nucleic Acids Res.*, **32** (10), pp. 3115 - 3123, 2004.
- [15] F. Tanaka, A. Kameda, M. Yamamoto e A. Ohuchi, Design of nucleic acid sequences for DNA computing based on a thermodynamic approach, *Nucleic Acids Res.*, **33**(3), pp. 903 - 911, 2005.
- [16] D. Voet e J.G. Voet, "Biochemistry", 3rd. Ed., John Wiley & Sons Inc., New Jersey, 2004.
- [17] J. Vieira e J. Messing, New pUC-derived cloning vectors with different selectable marks and DNA replication origins, *Gene*, 100, pp. 189–194, 1991.
- [18] E.O. Voit, "Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists", Cambridge University Press, Cambridge, 2000.
- [19] R. Weiss, "Cellular Computation and Communications using Engineered Genetic Regulatory Networks", Tese de Doutorado, Massachusetts Institute of Technology, 2001.
- [20] P. Wong, S. Gladney e J.D. Keasling, Mathematical model of the lac operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose, *Biotechnol. Prog.*, pp. 132–143, 1997.