



ALGORITMOS GENÉTICOS APLICADOS A MAPEAMENTO FÍSICO DE DNA

Fábio Ribeiro Cerqueira e Renarde Bergamo Paiva Stelzer

Departamento de Ciência da Computação e Informática
Faculdade de de Aracruz (UNIARACRUZ)
{renarde,fabio}@fsjb.edu.br

RESUMO

Neste artigo, propõe-se uma solução para o problema de mapeamento físico de DNA (PMP). O PMP aqui é visto como o clássico problema do caixeiro viajante. Sendo assim, foi necessária a criação de heurísticas para a obtenção de soluções em tempo polinomial. As heurísticas implementadas aqui foram baseadas em algoritmos genéticos que é uma técnica largamente utilizada nos últimos anos.

Palavras-chave: Mapeamento físico de DNA. Problema do caixeiro viajante. Propriedade dos “uns” consecutivos. Algoritmos Genéticos

ABSTRACT

In this paper we propose a solution to the problem of DNA physical mapping (PMP). The PMP is viewed here as the classic traveling salesman problem. So it was necessary to create heuristics in order to obtain solutions in polynomial time. The heuristics implemented here were based on genetic algorithms which is a wide used technique in recent researches.

Keywords: Physical mapping of DNA. Traveling salesman problem. Consecutive ones property. Genetic algorithms

INTRODUÇÃO

Neste trabalho, é apresentado um problema na área de análise de genomas, denominado mapeamento físico de DNA. Mostra-se aqui a definição desse problema, um modelo para ele, como resolvê-lo utilizando a abordagem de algoritmos genéticos e, finalmente, os testes e conclusões.

MAPEAMENTO FÍSICO DE DNA

A motivação para este trabalho reside no problema do **mapeamento físico de DNA** (PMP). Para se entender do que se trata, alguns conceitos se fazem necessários. Todos esses conceitos podem ser encontrados em Cerqueira (2000) e Setubal e Meidanis (1997).

Em um projeto genoma, a primeira pesquisa realizada se chama **seqüenciamento**, que é voltada para a descoberta da seqüência de bases (Adenina-A, Citosina-C, Guanina-G, Timina-T) que compõem o DNA do ser vivo em análise.

É sabido que os genomas são extremamente grandes, ou seja, possuem um número alto de bases componentes. As tecnologias atuais não permitem que os laboratórios seqüenciem a molécula inteira, pois as máquinas para esse fim têm um limite de até mil bases para o DNA em questão.

Considerando o exposto, o que se faz é a quebra do DNA em pequenos pedaços de tamanho até mil para que esses possam ter suas bases decifradas pelas máquinas seqüenciadoras e sejam remontados posteriormente na ordem correta, gerando a seqüência final (**consenso**) da molécula de origem de onde provieram. Essa remontagem é o problema conhecido como **montagem de fragmentos de DNA**.

O que se faz na prática é uma primeira quebra do DNA em pedaços menores, mas ainda grandes, denominados **clones**. Cada pedaço gerado da primeira quebra é novamente quebrado em fragmentos de mil bases, que são tratados na fase de montagem de fragmentos.

Uma vez tendo os consensos dos clones (resultado da fase de montagem de fragmentos), basta remontá-los em sua ordem original para que juntos dêem o consenso da molécula de DNA-alvo. Se fosse sabido de que região, na molécula

original, cada clone vem, essa ordenação seria fácil. O mapeamento de cada clone, em sua respectiva região de origem no DNA-alvo, é conhecido como o problema do mapeamento físico de DNA.

Uma importante técnica no PMP é o **mapeamento físico por hibridização**. Nesse caso, identifica-se a ocorrência de pequeninos trechos de DNA na molécula-alvo. Esses trechos se chamam **sondas**. Uma vez definidas essas sondas, verifica-se a sua ocorrência das mesmas também nos clones. Essa ocorrência é chamada de **hibridização**. O resultado disso é que, se houver hibridização de uma sonda y nos clones A e B, provavelmente estes se sobrepõem e, portanto, devem vir de regiões subseqüentes na molécula original.

Em todo esse processo explicitado até agora, é comum o acontecimento de uma série de problemas que dificultam o mapeamento. No que se refere ao PMP, pode-se destacar a situação das sondas não hibridizarem onde deveriam ou hibridizarem onde não deveriam, criando problemas chamados de **falsos negativos** e **falsos positivos**, respectivamente. Há, ainda, um fenômeno que ocorre no processo de quebra do DNA, em que algumas partes que foram quebradas se juntam e formam o que se chama de **clone quimérico**, que será formado por regiões não contíguas do DNA original. Será mostrado, mais à frente, como tratar esses problemas.

MODELOS E CONSIDERAÇÕES PARA A SOLUÇÃO DO PMP

Aqui o PMP foi abstraído como o problema: encontrar “uns” consecutivos em uma matriz binária, como visto a seguir.

A PROPRIEDADE DOS “UNS” CONSECUTIVOS (CONSECUTIVE ONES PROPERTY - C1P)

Uma matriz binária M , da forma mostrada na Figura 1(a), tem a **propriedade dos “uns” consecutivos** (SETÚBAL; MEIDANIS, 1997) se for possível permutar as colunas de tal modo que todas as linhas tenham seus “uns” de forma consecutiva.

Se M for tal que as linhas representem clones e as colunas representem sondas, a entrada $M_{i,j}$ será 1, se o clone i foi hibridizado pela sonda j e será 0, caso contrário.

Além disso, a permutação das colunas para gerar os “uns” consecutivos é equivalente a resolver a questão das sobreposições entre os clones, ou seja, o PMP.

Na Figura 1(b), é mostrada a matriz com os “uns” consecutivos, gerada a partir da matriz em 1(a). A Figura 2 mostra como interpretar essa matriz resultante em termos do mapeamento físico, clarificando que tal permutação realmente é o que se procura.

Clones x Sondas	S1	S2	S3	S4	S5	S6	S7	S8	S9
C1	1	1	0	1	1	0	1	0	1
C2	0	1	1	1	1	1	1	1	1
C3	0	1	0	1	1	1	1	0	1
C4	0	0	1	0	0	0	0	1	0
C5	0	0	1	0	0	1	0	0	0
C6	0	0	0	1	0	0	1	0	0
C7	0	1	0	0	0	0	1	0	0
C8	0	0	0	1	1	0	0	0	1

(a)

Clones x Sondas	S1	S2	S7	S4	S9	S5	S6	S3	S8
C1	1	1	1	1	1	1	0	0	0
C2	0	1	1	1	1	1	1	1	1
C3	0	1	1	1	1	1	1	0	0
C4	0	0	0	0	0	0	0	1	1
C5	0	0	0	0	0	0	1	1	0
C6	0	0	1	1	0	0	0	0	0
C7	0	1	1	0	0	0	0	0	0
C8	0	0	0	1	1	1	0	0	0

(b)

Figura 1. Em (a) uma matriz M com a C1P, que indica quais sondas hibridizam quais clones; em (b) a permutação que gerou os “uns” consecutivos

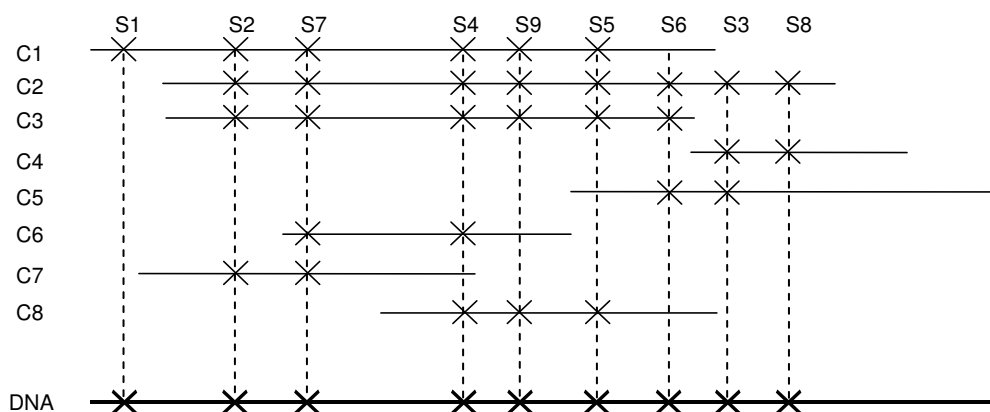


Figura 2. Alinhamento entre os clones, obtido pela matriz permutada (Fig. 1b)

PMP SOBRE A PERSPECTIVA DO PROBLEMA DO CAIXEIRO VIAJANTE

O grande problema de tudo que foi visto até o momento é que, na prática, os entraves citados, tais como: fragmentos quiméricos, falsos negativos e falsos positivos, fazem com que a matriz não tenha a C1P e, portanto, deve-se partir para uma permutação aproximada.

Suponha que M tenha a C1P e não esteja sujeita aos problemas citados. Seja M após a permutação. Em todas as linhas, seus 1s estarão consecutivos. No entanto, se uma dada linha corresponde a um clone quimérico, em que dois trechos tenham sido acidentalmente unidos no processo de quebra, então serão visualizados dois blocos de 1s separados por algum número de 0s. Esse bloco consecutivo de 0s, limitado por 1s, será chamado de **buraco**. Se, em uma outra linha, houver um falso negativo, o correspondente 0 separará dois blocos de 1s, criando outro buraco. Finalmente, um falso positivo pode quebrar um bloco de 0s em dois, possibilitando a criação de um outro buraco. Sendo assim, fica claro que há uma estreita relação entre erros e buracos na matriz. Uma interessante abordagem, então, seria encontrar uma permutação que minimize o número de buracos na matriz. Tal tarefa é NP-Difícil (SETÚBAL; MEIDANIS, 1997). Dessa forma, propostas para construção de heurísticas são obrigatórias.

Seja um grafo completo G não direcionado, com pesos, onde os vértices correspondam às colunas de M e o peso de cada aresta de G é dado pelo número de linhas onde as colunas correspondentes se diferem. Na Figura 3 é mostrado um exemplo.

Matriz M

	P1	P2	P3	P4	P5	P6
C1	1	1	1	0	0	0
C2	0	1	1	1	0	0
C3	1	0	0	1	1	0
C4	1	1	1	1	0	0

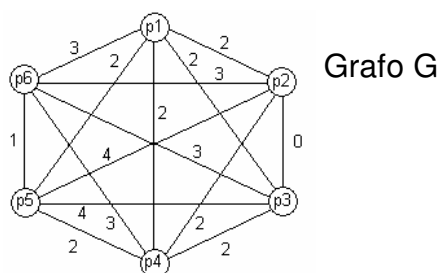


Figura 3. Uma matriz Clones x Sondas e o grafo correspondente

Agora se deve notar que o ciclo hamiltoniano de peso mínimo em G (Problema do Caixeiro Viajante-TSP) corresponde ao número mínimo de buracos em M (SETÚBAL; MEIDANIS, 1997). Para enxergar isso, note que, dada uma permutação das colunas, um buraco em uma linha significa que, em um certo ponto, há uma transição de 1 para 0 e mais tarde de 0 para 1. Então, para cada buraco, há duas transições e cada buraco contribui exatamente com dois para o peso do ciclo correspondente à dada permutação.

A conclusão é que basta então propor alguma heurística para o TSP.

PROPOSTA DE UM ALGORITMO GENÉTICO (AG) PARA O TSP

Todos os conceitos sobre AGs aqui podem ser vistos em Michalewicz (1996).

A proposta deste trabalho baseia-se em AGs e considera um **cromossomo** como o ciclo hamiltoniano e cada aresta do ciclo como um **gene**.

A **população inicial** é gerada baseando-se em um algoritmo guloso, que age de modo construtivo. Esse algoritmo cria n ciclos (cromossomos) iniciais, onde n é o número de vértices do grafo, partindo-se a cada vez de um vértice diferente do grafo.

Quanto ao cruzamento de cromossomos i_1 e i_2 , usa-se o **multiponto**, pegando ora um gene de i_1 ora um de i_2 . Na Figura 4, para gerar f_1 , um gene é pego de i_1 , por exemplo, 2-1. Procura-se, então, o gene em i_2 , cujo primeiro vértice coincida com o segundo vértice do gene 2-1. No caso, pega-se o 1-5. Agora, procura-se em i_1 o gene que começa com 5. Será o 5-4. Ao se procurar em i_2 um gene que comece com 4, será encontrado o 4-1, mas a solução até o momento já possui o vértice 1. É feita, então, uma **mutação** nesse gene, alterando o vértice de 1 para algum outro que ainda não tenha sido usado. Segue-se o processo, gerando-se f_1 . Depois, repete-se o procedimento, mas, desta vez, com o primeiro gene vindo de i_2 , que, no exemplo da Figura 4, foi escolhido o 4-1.

i_1 :	4-2 2-1 1-6 6-3 3-5 5-4	f_1 :	2-1 1-5 5-4 4-3 3-6 6-2
i_2 :	5-3 3-6 6-2 2-4 4-1 1-5	f_2 :	4-1 1-6 6-2 2-3 3-5 5-4

Figura 4. À esquerda, cromossomos que se cruzam e geram os filhos à direita

Outro tipo de mutação, que se tenta aplicar a todo novo cromossomo gerado, é a substituição de um par de genes por outro par, desde que isso cause uma melhora na qualidade do cromossomo (peso do ciclo diminua). Na Figura 5 troca-se o par: 1-2 e 3-4, por: 1-3 e 2-4. Note que o par escolhido para tentar a troca será aleatório.

1-2 2-3 3-4 4-5 5-6 6-1	1-3 3-2 2-4 4-5 5-6 6-1
---	---

Figura 5. À esquerda, cromossomo antes da retirada do par 1-2 e 3-4; à direita, depois da substituição pelo par 1-3 e 2-4

Os melhores cromossomos (baixo peso) serão selecionados para a população seguinte. O tamanho da população, devido à memória e tempo de processamento, foi limitado em $2n$. Outro ponto é que podem ocorrer casos em que as gerações que vão sendo sucessivamente produzidas não mais conseguem incrementar a melhor solução atual, sugerindo a presença de um mínimo local. Para esses casos, quando se atinge o máximo de iterações em que não há mudança, faz-se uma mutação em vários (e não apenas em um) pares de genes de vários cromossomos (escolha aleatória) da população atual.

COMENTÁRIOS E TESTES

O algoritmo proposto e implementado aqui possui complexidade $O(n^2)$, onde n é o número de vértices do grafo. Essa complexidade vem dos cruzamentos. Faz-se $2n$ cruzamentos e para cada cruzamento manipulam-se $2n$ genes.

Os testes a seguir foram feitos em um computador PENTIUM 3 550MHZ, com 128MB de RAM. Utilizou-se uma base de problemas do tipo TSP (TSPLIB), onde há várias instâncias e suas soluções ótimas. Na Tabela 6, pode-se notar que há uma tendência de melhora, quando se aumenta o número de iterações.

Tabela 6. Resultados experimentais

Instâncias	Valor da Sol. Ótima	Valor do AG (tempo entre parênteses)			Valor Alg. Guloso
		1000 iter.	2000 iter.	5000 iter.	
eil51	426	447.61 (18s)	440.24 (37s)	437.29 (1m37s)	505.8
u159	42080	43138.68 (3m9s)	43038.21 (6m8s)	42982.4(17m2s)	48586.7
u574	36905	43215.68 (37m37s)	43190.4(1h29m5s)	43162.56(3h33m22s)	45141.5

Obs.: Nos nomes das instâncias, pode-se identificar o número de vértices do grafo (51, 159 e 574)

CONCLUSÃO

O resultado do trabalho foi satisfatório. Com um algoritmo da ordem $O(n^2)$, conseguiu-se atingir resultados melhores do que o algoritmo guloso e que vão se aproximando do valor ótimo. Como os testes foram feitos em um PC doméstico, não foi possível realizar testes com um número de iterações mais elevado, mas tudo indica que, aumentando consideravelmente o número de iterações, as soluções chegarão bem próximas ao ótimo.

PRINCIPAIS REFERÊNCIAS

CERQUEIRA, F. **Montagem de fragmentos de DNA**. 2000. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Campinas, 2000.

SETÚBAL, J.; MEIDANIS, J. **Introduction to computational molecular biology**. Pacific Grove: PWS, 1997.

MICHALEWICZ, Z. **Genetic algorithms + data structures = evolution programs**, 3. ed. Charlotte: Springer, 1996.

RAMOS, N.; SALVADO, S. **Jogo de xadrez por algoritmos genéticos evolutivos**. 2001. Monografia (Curso de Ciência da Computação) - Instituto Superior Técnico, 2001.

REEVES, C. R. **Modern heuristic techniques for combinatorial problems**. New York: Blackwell Scientific Publications, 1993.

WATERMAN, M. S. **Introduction to computational biology**. Boca Raton: Chapman and Hall, 1995.

TELES, G. P. **A Propriedade dos “uns” consecutivos e aplicações**. Campinas: Universidade Estadual de Campinas, 1997.

BONDY, J. A.; MURTY, U. S. R. **Graph theory With applications**. Ontario: The Macmillan Press LTD, 1977.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. **Introduction to algorithms**. New Jersey: Prantice Hall, 1998.