Estimação da Distância de Reversão de Genomas Baseada em Técnicas de *Machine Learning*

Nathalia Menini, Sergio Arnosti e Jorge da Silva

Instituto de Computação da Unicamp,
Av. Albert Einstein, 1251 - Cidade Universitária, Campinas - SP
{nathmenini,serza.arnosti,jorge.inatel}@gmail.com

Resumo FAZER

Keywords: fazer.

1 Introdução

Comparar dois genomas é uma tarefa fundamental para se estudar as relações e a evolução entre os genes [5,2]. Nesse caminho, a Biologia Computacional tem se apresentado como uma ótima aliada para os pesquisadores da área, tornando possível descobrir relações entre os genes que, para a percepção do ser humano, poderia ser muito difícil de detectar. [2]

O *rearranjo de genomas* é uma mutação que ocorre nos genomas mitocondrias [1], de modo que a ordem dos genes no genoma está em constante rearranjo. Desse modo, através da estimação da distância de rearranjo entre dois genes, a relação entre eles pode ser estimada [4].

Quando um fragmento do filamento de DNA é revertido na replica final, temos o que chamamos de *reversão*, que é uma das mutações mais comumente vistas em genomas (uma ilustração dessa mutação pode ser vista na Figura 1, em que ilustra a transformação do *Tabacco* em *Lobelia fervens*). Por outro lado, se dois fragmentos de DNA trocam de posições durante o processo de replicação (mas não sofrem reversão), temos a mutação chamada de *transposição* [2]. Neste trabalho, o nosso foco será apenas na mutação conhecida por reversão.

A área da Filogenia, que estuda a história evolutiva das relações entre espécies, depende fortemente do Principio da Parcimônia. Basicamente, dado um conjunto de possíveis explicações para um fato, a explicação mais simples é a mais provável de estar certa. Como as mutações são relativamente raras de acontecer, quando os pesquisadores tentam construir uma árvore filogenética, eles tentam fazer com que as espécies tenham o menor número de antecessores possíveis, ou seja, é muito mais provável que duas espécies que possuem uma determinada característica tenham evoluído do mesmo ancestral comum que desenvolveu essa característica, em vez de acreditar que a característica evoluiu duas vezes, de espécies diferentes [2].

Considere a permutação identidade $\iota = (1, 2, ..., n)$ de tamanho n como o genoma original. Dada qualquer permutação π de tamanho n, o problema consiste

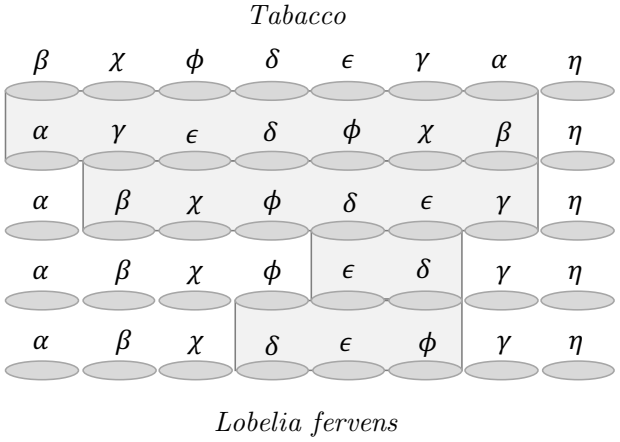

Figura 1. Transformação do *Tabacco* em *Lobelia fervens* através de operações de reversão. Imagem adaptada de Bafna e Pevzner [8].

em construir um modelo capaz de calcular a distância de reversão de ι e π, em que são permitidos apenas operações de reversão.

Uma outra possível representação de genomas é utilizando a *orientação* dos genes. Nesse caso, representamos o genoma com os símbolos + ou − para cada gene como, por exemplo, na permutação $(+1, -2, \cdots, -n)$. Na literatura, essa abordagem é conhecida como *permutações com orientação de genes conhecida*. Por outro lado, se não temos informações sobre a orientação dos genes, os sinais são omitidos e a abordagem passa a ser chamada de *permutação sem orientação de genes conhecida*. Neste trabalho, focaremos apenas nas permutações sem orientação.

Entretanto, determinar o número mínimo de reversões necessárias para um genoma se transformar no outro, no contexto de permutação sem orientação, não é um problema fácil de se resolver, uma vez que foi provado ser um problema NP-difícil [6]. Muitas abordagens já foram proposta para atacar esse problema como, por exemplo, Kececioglu e Sankoff [7], em que propuseram o algoritmo de 2-aproximação que remove *breakpoints* por reversão (o conceito de *breakpoint* será descrito posteriormente) e Bafna e Pevzner [8] que criaram o algoritmo de 1.75-aproximação utilizando grafos de reversão. O atual estado da arte foi desenvolvido por Barman et al. [9], que construiu um algoritmo de 1.375-aproximação. Abordagens utilizando Algoritmos Genéticos e técnicas de *Machine Learning* (ML) também já foram propostas em Auyeung e Abraham [10] e da Silva et al [2], respectivamente.

Entretanto, a maioria dos algoritmos na literatura buscam, além de encontrar a distância de reversão, encontrar também as reversões que transformam uma permutação em outra. Neste trabalho, propomos a utilização de técnica de ML (como Regressão Linear e *Neural Networks*) com o objetivo de estimar a

distância de reversão baseando-se em *features* obtidas a partir da permutação. Em um primeiro momento, focaremos esforços apenas em estimar a distância e não obter as reversões em si, uma vez que, em áreas como filogenia, deseja-se, apenas, uma métrica de distância entre objetos e, quanto mais precisa a métrica for, melhor será a análise. Além disso, devido a dificuldade de obter dados da distância de reversão exata para permutações grandes, utilizaremos como treinamento dos nossos algoritmos um *dataset* com permutações de tamanhos reduzidos e, posteriormente, verificaremos o seu desempenho em permutações de vários tamanhos.

2 Definições e Conceitos Básicos

Nessa seção explicaremos a operação de reversão no contexto de permutações sem orientação e, também, alguns conceitos utilizados para o protocolo de extração das *features* utilizadas nos algoritmos de ML.

Os genomas podem ser representados por permutações (n-tuplas, em que n é a quantidade de genes), em que cada gene é representado por um número e que todos os genes são diferentes um dos outros. Ou seja, uma permutação π de tamanho n é representada por $\pi = (\pi_1\ \pi_2\ \cdots\ \pi_n)$, com $\pi_i \in \{1, 2, \cdots n\}$ e $\pi_i \neq \pi_j \iff i \neq j$. A *permutação identidade*, como mencionada anteriormente, é definida como $\iota = (1\ 2\ \cdots n)$. A *composição* entre duas permutações π e σ é dada por $\pi\sigma = (\pi_{\sigma_1}\ \pi_{\sigma_2} \cdots \pi_{\sigma_3})$. Definimos também o *inverso* de uma permutação π como π^{-1}, em que $\pi^{-1}\pi = \iota$.

A *permutação extendida* de uma permutação $\pi = (\pi_1\ \pi_2\ \cdots\ \pi_n)$, denotada por π_e, é definida como $\pi_e = (0\ \pi_1\ \pi_2\ \cdots\ \pi_n\ n+1)$. A definição de permutação extendida é necessária para a descrição de operações que serão abordadas no decorrer do trabalho.

Dadas duas permutações π e σ e um conjunto de operações $\Sigma = \{\rho_1, \rho_2, \cdots, \rho_k\}$, o problema de transformar π em σ consiste em obter no menor número possível de operações de Σ aplicada em π, tal que *pi* seja transformada em σ. O número de operações necessárias é chamado de distância entre π e σ e é representado por $d(\pi, \sigma)$. Para fins de notação, utilizaremos que $d(\pi, \iota) = d(\pi)$.

O problema de encontrar a distância entre as permutações π e σ é equivalente ao problema de encontrar a distância entre alguma permutação π' e ι, em que $\pi = \sigma^{-1}\pi$. Ou seja, se queremos encontrar a distância entre π e σ, equivale ao problema de ordenação de $\sigma^{-1}\pi$. Por exemplo, na Figura 2, se considerarmos $\pi = (5\ 3\ 1\ 6\ 7\ 4\ 2)$ e $\sigma = (6\ 3\ 2\ 7\ 5\ 1\ 4)$, temos que $\sigma^{-1} = (6\ 3\ 2\ 7\ 5\ 1\ 4)$, o que faz com que $\pi = \sigma^{-1}\pi = (5\ 2\ 6\ 1\ 4\ 7\ 3)$. Então, a distância entre π e σ equivale a distância de ordenação de π'.

2.1 Reversões

Uma operação de reversão $\rho_r(i, j)$, com $1 \leq i < j \leq n$ reverte um fragmento da permutação, ou seja, $\rho_r(i, j)$ aplicado em $\pi = (\pi_1\ \cdots\ \pi_{i-1}\ \underline{\pi_i \cdots \pi_j}\ \pi_{j+1}\ \cdots\ \pi_n)$ gera $\pi\rho_r(i, j) = (\pi_1\ \cdots\ \pi_{i-1}\ \underline{\pi_j \cdots \pi_i}\ \pi_{j+1}\ \cdots\ \pi_n)$.

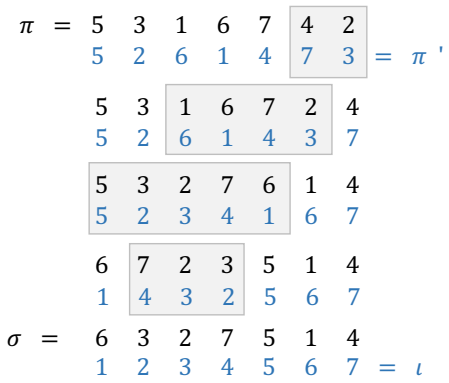

Figura 2. Equivalência na distância entre π e σ com a distância entre π' e ι.

Considere, por exemplo, $\pi = (1\ 4\ 3\ 2\ 6\ 7\ 5)$ e a operação de reversão $\rho_r(2,4)$. O resultado após aplicar essa operação é $\pi\rho_r(2,4) = (1\ \underline{2\ 3\ 4}\ 6\ 7\ 5)$.

2.2 *Breakpoints*

2.3 *Strips*

2.4 Maior e Menor *Strip* Crescente

2.5 Ciclos

3 Metodologia

4 Resultados e Avaliação

5 Conclusões

Referências

1. Russell, P. J.: iGenetics. Benjamim Cummings (2002)
2. da Silva, M., Oliveira, A., Dias, Z.: Machine Learning Applied to Sorting Permutations by Reversal and Transpositions. Technical report, Unicamp (2017)
3. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
4. Pevzner, P.: Computational Molecular Biology An Algorithmic Approach. MIT Press (2001)

5. Lou, X.W., Zhu, D.M.: Sorting Unsigned Permutations by Weighted Reversals, Transpositions, and Transversals. Journal of Computer Science and Technology (2009)
6. Caprara, A.: Sorting by Reversals is Difficult. Proceedings of the First International Conference on Comptational Molecular Biology (1997)
7. Kececioglu, J., Sankoff, D.: Exact and approximation algorithms for sorting by reversals, with application to genome rearreangement. Algorithmica (1995)
8. Bafna, V., Pevzner, P.: Genome rearragements and sorting by reversals. SIAM Journal on Computing (1996)
9. Berman, P., Hannenhalli, S., Karpinski, M.: 1.375-approximation algorithm for sorting by reversals. In 10th Annual European Symposium on Algorithms (2002)
10. Auyeung, A., Abraham, A.: Estimating Genome Reversal Distance by Genetic Algorithm. ?