

Alinhamento de seqüências com rearranjos

Augusto Fernandes Vellozo¹

Orientador: Alair Pereira do Lago¹

¹Departamento de Ciência da Computação - Instituto de Matemática e Estatística

Universidade Estadual de São Paulo – SP – Brasil (DCC– IME – USP)

vellozo@ime.usp.br

Abstract. *Sequence comparison done by alignment algorithms is one of the most fundamental tasks in bioinformatics. The evolutive mutations considered in these alignments are insertions, deletions and substitutions of nucleotides. This work treats of generalizations introduced in alignment algorithms in such a way that other mutations known as rearrangements are also considered, more specifically, we consider non-overlapping inversions, duplications and duplications in tandem. The new algorithms presented in this work improve asymptotically the runt time of the algorithms known to solve the same problem. This paper is an extended abstract of [Vellozo 2007].*

Resumo. *Uma das tarefas mais básicas em bioinformática é a comparação de seqüências feita por algoritmos de alinhamento, que modelam as alterações evolutivas nas seqüências biológicas através de mutações como inserção, remoção e substituições de nucleotídeos. Este trabalho trata de generalizações nos algoritmos de alinhamento que levam em consideração outras mutações conhecidas como rearranjos, mais especificamente, inversões não sobrepostas,*

duplicações e duplicações em tandem. Os novos algoritmos apresentados neste trabalho melhoram assintoticamente o tempo de execução dos algoritmos conhecidos para resolver o mesmo problema. Este artigo é um resumo estendido de [Vellozo 2007].

1. Introdução

Atualmente, devido ao grande número de projetos de seqüenciamentos desenvolvidos e finalizados, temos uma enorme quantidade de dados moleculares disponíveis, principalmente de DNA, RNA e proteína. Processar estes dados para extrair informações relevantes é um grande desafio.

Uma das maneiras de manipular, estudar e estruturar estes dados moleculares é através do uso de seqüências de símbolos. No caso do DNA os símbolos são as letras que representam as bases de nucleotídeos (A, C, T e G), no caso das proteínas os símbolos são letras que representam os 20 aminoácidos elementares.

Na história da evolução vários eventos biológicos introduzem mudanças nas seqüências do DNA. Quando duas seqüências tem alto grau de semelhança é esperado que elas se diferenciem apenas em alguns trechos. Normalmente as diferenças nestes trechos são conseqüências da ocorrência de alguns eventos biológicos. Em geral, para a visualização destes eventos ocorridos, é feito um procedimento de alinhamento entre as duas seqüências. Procedimentos de alinhamento típicos tentam identificar que partes não mudam e, nas partes que mudam, quais os eventos biológicos que ocasionaram as mudanças. Tipicamente os eventos biológicos considerados para um alinhamento são as *substituições, remoções e inserções* de nucleotídeos.

O alinhamento de seqüências biológicas é uma das principais técnicas utilizadas por biólogos para comparar seqüências biológicas.

Um alinhamento pode ser associado a um conjunto de operações de edição que transformam uma seqüência em outra. Normalmente, por motivos de performance, as únicas operações de edição consideradas para obtenção de um alinhamento ótimo são a *substituição* de um símbolo em outro, a *inserção* de um símbolo e a *remoção* de um símbolo. Se associamos custos a cada operação de edição, existe um algoritmo clássico de programação dinâmica, que em tempo de execução $O(n^2)$ [Needleman and Wunsch 1970]¹², computa um conjunto mínimo de operações de edição que tem o custo total ótimo e apresenta o alinhamento associado.

Dadas duas seqüências s e t e pontuações fixas para cada operação de edição associada a uma mutação, o problema de *obter uma transformação ótima com inversões de s e t* é um problema de otimização que procura uma transformação com inversões de s e t tal que, a soma das pontuações das operações de edição desta transformação é máxima.

Além desses eventos que resultam em alterações em um único símbolo na seqüência biológica, iremos considerar também os rearranjos para a obtenção de alinhamentos. Os rearranjos agem sobre um fragmento da seqüência. Consideraremos, além dos eventos

¹Nas análises das complexidades dos algoritmos, consideraremos neste texto que n é o comprimento da maior seqüência analisada.

²O conhecido artigo de Needleman e Wunsch é geralmente considerado a primeira contribuição importante em procedimentos computacionais para a comparação de seqüências. O algoritmo proposto por eles neste artigo, apesar de não ter sido feita uma análise de tempo no artigo, é considerado cúbico, mas normalmente o nome "Algoritmo de Needleman e Wunsch" é utilizado para denominar o algoritmo quadrático clássico de programação dinâmica para alinhamentos [Setubal and Meidanis 1997].

de inserção, remoção e substituição de um símbolo, os eventos de inversão e de duplicação de fragmentos para a obtenção de um alinhamento ótimo de duas seqüências. Estes não são os únicos tipos de rearranjos, mas são considerados muito importantes e são largamente estudados, como por exemplo em [Shaw and Lupski 2004], inclusive na associação de doenças humanas à rearranjos no DNA, como por exemplo em [Verkerk et al. 1991]. Os novos resultados comparativos feitos por Sherer et al. [Feuk et al. 2005] nas seqüências de DNA do homem e do chimpanzé, mostram a importância do estudo da inversão ao nível da seqüência de DNA e reportaram 83 seqüências reversas que estão contidas dentro de genes.

Consideraremos que $s = s_1 s_2 \dots s_n = s[1 \dots n]$ e $t = t_1 t_2 \dots t_m = t[1 \dots m]$ são as duas seqüências a serem comparadas e alinhadas.

Chamaremos uma seqüência de operações de edição que transforma uma seqüência s em uma seqüência t de uma *transformação de s em t* . Consideraremos que cada operação de edição tem uma pontuação associada e que a pontuação de uma transformação é a soma das pontuações de suas operações de edição. Consideraremos que uma transformação de s em t é ótima se não existir outra transformação de s em t com pontuação maior.

Diremos que $\omega_{s,t}((i', j'), (i, j))$ é a pontuação de uma transformação ótima de $s[i'+1 \dots i]$ em $t[j'+1 \dots j]$ considerando as operações de substituição, inserção e remoção.

2. Alinhamento com inversões

Quando possibilitamos a operação de edição de inversão para a obtenção de uma transformação diremos que a transformação obtida é uma *transformação com inversões*

de s em t . A operação de edição de inversão substitui um fragmento da sequência pelo seu reverso complementar.

Como pode ser visto em [Chen et al. 2005], o problema de decisão associado ao problema de obter uma transformação ótima com inversões para um alfabeto ilimitado é NP-difícil

Em 1992, Schöniger e Waterman [Schöniger and Waterman 1992] introduziram uma hipótese simplificadora: todas as regiões envolvendo inversões não se sobrepõem. Com esta simplificação eles obtiveram um *alinhamento ótimo com inversões não sobrepostas* em tempo $O(n^6)$. A partir daí, outros melhoraram este tempo chegando até $O(n^4)$ [do Lago et al. 2003, Gao et al. 2003].

Propomos dois algoritmos exatos que obtêm uma transformação ótima com inversões não sobrepostas em tempo $O(n^3 \log n)$ [Alves et al. 2005] e $O(n^3)$ [Vellozo et al. 2006].

Diremos que $\overline{s[i' \dots i]}$ é o complemento reverso de $s[i' \dots i]$. Assumiremos que $ins(t[j])$, $rem(s[i])$, $sub(s[i], t[j])$ e ω_{inv} são, respectivamente, as pontuações das operações de edição de inserção de um símbolo $t[j]$, remoção de um símbolo $s[i]$, substituição de um símbolo $s[i]$ por $t[j]$ e inversão de um fragmento de s .

Os dois algoritmos computam a matriz B , onde $B[i, j]$ é a pontuação do alinhamento ótimo com inversões não sobrepostas de $s[1 \dots i]$ e $t[1 \dots j]$. O Algoritmo 1 contém o pseudo-código seguido pelos dois algoritmos.

Entre as linhas 3 a 14 do Algoritmo 1 são considerando somente as operações de edição de inserção, remoção e substituição de um símbolo. O laço da linha 18 considera

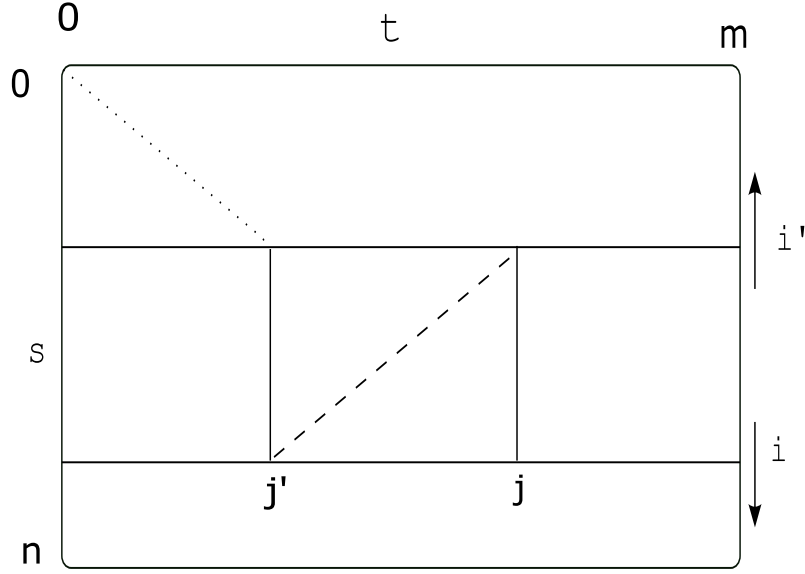


Figura 1. Ilustração da execução do Algoritmo 1. A linha pontilhada representa em um grafo de edição um alinhamento ótimo com inversões não sobrepostas de $s[1 \dots i] \times t[1 \dots j]$ e a linha tracejada o alinhamento de $\overline{s[i' + 1 \dots i]} \times t[j' + 1 \dots j]$

a possibilidade de uma operação de edição de inversão de $s[i' + 1 \dots i]$, $0 \leq i' \leq i \leq n$.

Os dois algoritmos que propomos diferem somente na forma da construção da matriz $W_{i'}^i$ e do vetor $A_{i'}^i$, onde $W_{i'}^i[j', j]$ é a pontuação do alinhamento ótimo de $\overline{s[i' + 1 \dots i]}$ e $t[j' + 1 \dots j]$ considerando somente inserção, remoção e substituição, pois não consideramos inversões sobrepostas no alinhamento de s e t , e $A_{i'}^i[j]$ é o valor máximo de $(B[i', j] + W_{i'}^i[j', j] + \omega_{inv})$, para todo j' , $0 \leq j' \leq j$.

2.1. Alinhamento com inversões não sobrepostas em tempo $O(n^3 \log n)$

Jeanette Schmidt [Schmidt 1998] desenvolveu um algoritmo que constrói uma estrutura de árvores binárias de altura $\log m$ que possibilita obter um valor de $\omega_{s,t}((i', j'), (i, j))$ em tempo $\log m$. Dada a estrutura de árvores com os valores de $\omega_{s,t}((i', j'), (i - 1, j))$ o algoritmo precisa de tempo $O(m \log m)$ para construir outras m árvores que contêm os

Algoritmo 1 Algoritmo para obtenção da matriz B tal que $B[i, j]$ é a pontuação do alinhamento ótimo com inversões não sobrepostas de $s[1 \dots i]$ e $t[1 \dots j]$

INVGERAL(s, t)

```

1  para  $i$  de 0 até  $|s|$  faça
2       $\triangleright$  Obtém  $B[i, j]$  considerando somente inserção, remoção e substituição
3      se  $i = 0$  então
4           $B[0, 0] \leftarrow 0$ 
5      senão
6           $B[i, 0] \leftarrow B[i - 1, 0] + \text{rem}(s[i])$ 
7      para  $j$  de 1 até  $|t|$  faça
8          se  $i = 0$  então
9               $B[0, j] \leftarrow B[0, j - 1] + \text{ins}(t[j])$ 
10         senão
11              $h \leftarrow B[i, j - 1] + \text{ins}(t[j])$ 
12              $v \leftarrow B[i - 1, j] + \text{rem}(s[i])$ 
13              $d \leftarrow B[i - 1, j - 1] + \text{sub}(s[i], t[j])$ 
14              $B[i, j] \leftarrow \max(h, v, d)$ 
15          $\triangleright$  A seguir, obtém  $B[i, j]$  considerando também inversões não sobrepostas
16          $\triangleright W_{i'}^i[j', j] = \text{pontuação do alinhamento ótimo de } \overline{s[i' + 1 \dots i]} \text{ e } t[j' + 1 \dots j]$ 
17         Inicializa  $W_{i+1}^i$ 
18         para  $i'$  de  $i$  descendo até 0 faça
19             Obtém  $W_{i'}^i$  a partir de  $W_{i'+1}^i$ 
20             Para cada  $j$  obtém  $A_{i'}^i[j] = \max_{j' \leq j} (B[i', j] + W_{i'}^i[j', j] + \omega_{inv})$ 
21             para  $j$  de 0 até  $|t|$  faça
22                  $B[i, j] \leftarrow \max(B[i, j], A_{i'}^i[j])$ 
23 devolva  $B$ 

```

valores de $\omega_{s,t}((i', j'), (i, j))$.

Pode-se provar que a matriz $W_{i'}^i$ é uma matriz de Monge inversa triangular superior e portanto totalmente monotônica, o que possibilita que utilizemos o algoritmo proposto por [Aggarwal et al. 1987] para obtermos $A_{i'}^i[j]$ com $O(m)$ comparações.

Utilizando a estrutura de árvores proposta em [Schmidt 1998] para armazenar os valores de $\omega_{s,t}((i', j'), (i, j))$, executamos a linha 19 do Algoritmo 1 em tempo $O(m \log m)$ e, utilizando o algoritmo proposto por [Aggarwal et al. 1987], obtemos $A_{i'}^i[j]$ na linha 20 em tempo $O(m \log m)$. Como essas linhas são executadas para todo i e i' , $0 \leq i' \leq i \leq n$, o Algoritmo 1 é executado em tempo $O(n^2 m \log m)$. Além disto, pode-se verificar que o Algoritmo 1 precisa de $O(\max(nm, m^2))$ de memória.

2.2. Alinhamento com inversões não sobrepostas em tempo $O(n^3)$

3. Alinhamento com duplicações

Uma duplicação é um evento biológico que ocorre frequentemente e que insere uma cópia de um fragmento da sequência na própria sequência. Existe um tipo de duplicação muito estudado que são as duplicações encadeadas (ou duplicações *tandem*). Neste tipo de duplicação a cópia é inserida imediatamente após (ou antes) do fragmento copiado. Diremos que repetição é a cópia inserida e sequência base é a sequência de onde a repetição foi copiada.

Em [Benson 1997] é proposto um modelo para o alinhamento de sequências que considera duplicações em *tandem* de mesmo tamanho. São propostos dois algoritmos exatos para obter um tal alinhamento ótimo que considera estas duplicações. O primeiro algoritmo proposto executa em tempo $O(n^5)$ e espaço $O(n^2)$. O segundo algoritmo pro-

posto executa em tempo $O(n^4)$ e espaço $O(n^3)$. Propomos algoritmos que executam em tempo $O(n^3)$ e memória $O(n^2)$.

Propomos um modelo para obter um alinhamento ótimo com duplicações que é parecido com o modelo proposto por Benson, porém com algumas restrições diferentes:

1. Benson propõe somente duplicações em *tandem*. Não mantemos a restrição de ser em *tandem*. Um modelo com esta restrição também pode ser considerado sem alterar as complexidades do algoritmo que propomos.
2. No modelo proposto por Benson, se um trecho de s é uma repetição então a sequência base a ser comparada (ou alinhada) com esta repetição deve ser um trecho de t . Portanto ele não considera que a sequência base da repetição pode estar na própria sequência. Nós consideramos que a sequência base pode estar na outra sequência. Com isto, se estamos interessados em obter uma transformação ótima com duplicações de s em t , o modelo proposto por Benson considera que as operações de edição pontuais (inserção, remoção e substituição) em um trecho duplicado devem ocorrer depois da duplicação na transformação. Nós consideramos que as operações de edição pontuais podem ocorrer antes ou depois da duplicação.
3. No modelo proposto por Benson, a sequência base deve ser a mesma para repetições contíguas. Nós consideramos que a sequência base pode ser qualquer trecho das sequências s ou t .

Se estamos obtendo uma transformação ótima de s em t com duplicações então consideramos que a sequência base de uma repetição $s[i + 1 \dots i']$ está em $s[1 \dots i]$ ou $t[1 \dots j]$, onde $s[1 \dots i]$ se transforma em $t[1 \dots j]$ na transformação ótima sendo obtida. Além disto, em uma transformação com duplicações consideramos a existência da ope-

ração de edição de *excisão* que é a operação de edição dual à duplicação, ou seja, uma excisão remove um trecho da sequência (repetição), tal que a sequência de símbolos do trecho removido ainda aparece como um fragmento (sequência base) da sequência gerada após a excisão. Numa transformação de s em t , podemos associar uma operação de edição de excisão ao evento biológico de uma duplicação na sequência s que não ocorreu em t ou ao evento biológico de remoção de uma repetição em t que não ocorreu em s .

Utilizando as técnicas utilizadas em [Landau and Ziv-Ukelson 2001], podemos obter um alinhamento ótimo com duplicações em *tandem* segundo o mesmo modelo proposto por Benson, em tempo $O(n^3)$ e com memória $O(n^2)$. Porém, acreditamos que o modelo que propomos está mais próximo à realidade, pois considera mais casos de duplicações.

Dados i e i' tais que $0 \leq i' < i \leq n$, dizemos que $X_s^{i',i}$ é o conjunto de todas as possíveis sequências que podem ser a sequência base de $s[i' + 1 \dots i]$. Definimos a matriz R_s , tal que $R_s(i', i) = \omega_{s,x}((i', i), (0, |x|)) + \text{dup}(s[i' \dots i])$ se $0 \leq i' < i \leq n$ e 0 caso contrário, onde $x \in X_s^{i',i}$ e não existe um $x' \mid x' \in X_s^{i',i}$ e $\omega_{s,x'}((i', i), (0, |x'|)) > \omega_{s,x}((i', i), (0, |x|))$ e $\text{dup}(s[i' \dots i])$ é a pontuação da operação de edição de duplicação de $s[i' + 1 \dots i]$. Dizemos que $R_s(i', i)$ é a pontuação da inserção da repetição $s[i' + 1 \dots i]$, $0 \leq i' < i \leq n$. No nosso modelo consideraremos que $X_s^{i',i}$ são todos os possíveis fragmentos de $s[1 \dots i]s[i' + 1 \dots n]$ e todos os possíveis fragmentos de $t[1 \dots m]$, onde $s[1 \dots i]s[i' + 1 \dots n]$ é a concatenação das sequências $s[1 \dots i]$ e $s[i' + 1 \dots n]$. De forma análoga, definimos a matriz R_t . Utilizando-se apenas de programação dinâmica conseguimos construir as matrizes R_s e R_t em tempo $O(n^3)$ e memória $O(n^2)$.

Seja M uma matriz $n \times m$ tal que

$$\begin{aligned}
M[0, 0] &= 0 \\
M[i, 0] &= \max \left\{ \begin{array}{l} M[i-1, 0] + \text{rem}(s[i]), \\ \max_{\forall i' | 0 \leq i' < i} (M[i', 0] + R_s(i', i)) \end{array} \right\}, i > 0 \\
M[0, j] &= \max \left\{ \begin{array}{l} M[0, j-1] + \text{ins}(t[j]), \\ \max_{\forall j' | 0 \leq j' < j} (M[0, j'] + R_t(j', j)) \end{array} \right\}, j > 0 \\
M[i, j] &= \max \left\{ \begin{array}{l} M[i, j-1] + \text{ins}(t[j]), \\ M[i-1, j] + \text{rem}(s[i]), \\ M[i-1, j-1] + \text{sub}(s[i], t[j]), \\ \max_{\forall i' | 0 \leq i' < i} (M[i', j] + R_s(i', i)), \\ \max_{\forall j' | 0 \leq j' < j} (M[i, j'] + R_t(j', j)) \end{array} \right\}, i > 0 \text{ e } j > 0
\end{aligned} \tag{1}$$

Diremos que $M[i, j]$ é a pontuação de um alinhamento ótimo com duplicações de $s[1 \dots i]$ e $t[1 \dots j]$. Dadas as matrizes R_s e R_t existe um algoritmo que com programação dinâmica constrói a matriz M em tempo $O(n^3)$ e memória $O(n^2)$.

4. Conclusão

Para a comparação de duas seqüências, desenvolvemos e apresentamos algoritmos exatos e inéditos, que obtêm alinhamentos que consideram a possibilidade da ocorrência de outros importantes eventos biológicos (inversão e duplicação) além dos eventos de inserção, remoção e substituição, comumente utilizados em algoritmos usuais para obtenção de alinhamentos ótimos. Esperamos com isto, que o alinhamento obtido esteja mais próximo de mostrar o que realmente ocorreu na evolução.

Apesar do problema da obtenção de um alinhamento ótimo com inversões não sobrepostas ser um problema já bem estudado e com alguns algoritmos já publicados, con-

seguimos desenvolver algoritmos com complexidade de tempo significativamente melhor que os já existentes.

Para o alinhamento com duplicações, não conhecemos nenhum outro trabalho que utilize os modelos de duplicações que apresentamos. O modelo mais próximo que conhecemos do modelo que propomos para alinhamentos com duplicações em *tandem* é o modelo proposto por Benson [Benson 1997]. Porém, os algoritmos propostos por Benson têm complexidades de tempo e memória muito piores que as dos algoritmos que propomos, apesar do nosso espaço de busca das duplicações em *tandem* ser maior que o do Benson.

Referências

- [Aggarwal et al. 1987] Aggarwal, A., Klawe, M. M., Moran, S., Shor, P., and Wilber, R. (1987). Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(2):195–208.
- [Alves et al. 2005] Alves, C. E. R., do Lago, A. P., and Vellozo, A. F. (2005). Alignment with non-overlapping inversions in $O(n^3 \log n)$ -time. *Electronic Notes in Discrete Mathematics*, 19:365–371.
- [Benson 1997] Benson, G. (1997). Sequence alignment with tandem duplication. In *RECOMB '97: Proceedings of the first annual international conference on Computational molecular biology*, pages 27–36, New York, NY, USA. ACM Press.
- [Chen et al. 2005] Chen, X., Zheng, J., Fu, Z., Nan, P., Zhong, Y., Lonardi, S., and Jiang, T. (2005). Assignment of orthologous genes via genome rearrangement. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(4):302–315.

- [do Lago et al. 2003] do Lago, A. P., Muchnik, I., and Kulikowski, C. (2003). An $O(n^4)$ algorithm for alignment with non-overlapping inversions. In *Second Brazilian Workshop on Bioinformatics, WOB 2003*, Macaé, RJ, Brazil. <http://www.ime.usp.br/~alair/wob03.pdf>.
- [Feuk et al. 2005] Feuk, MacDonald, Tang, Carson, Li, Rao, Khaja, and Scherer (2005). Discovery of human inversion polymorphisms by comparative analysis of human and chimpanzee DNA sequence assemblies. *PLoS Genet*, 1(4):e56.
- [Gao et al. 2003] Gao, Y., Wu, J., Niewiadomski, R., Wang, Y., Chen, Z.-Z., and Lin, G. (2003). A space efficient algorithm for sequence alignment with inversions. In *Computing and Combinatorics, 9th Annual International Conference, COCOON 2003*, volume 2697 of *Lecture Notes in Computer Science*, pages 57–67. Springer-Verlag.
- [Landau and Ziv-Ukelson 2001] Landau, G. M. and Ziv-Ukelson, M. (2001). On the common substring alignment problem. *J. Algorithms*, 41(2):338–359.
- [Needleman and Wunsch 1970] Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.
- [Schmidt 1998] Schmidt, J. P. (1998). All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings. *SIAM J. Comput.*, 27(4):972–992 (electronic).
- [Schöniger and Waterman 1992] Schöniger, M. and Waterman, M. S. (1992). A local algorithm for DNA sequence alignment with inversions. *Bulletin of Mathematical Biology*, 54(4):521–536.

- [Setubal and Meidanis 1997] Setubal, J. and Meidanis, J. (1997). *Introduction to computational molecular biology*. PWS Publishing Company.
- [Shaw and Lupski 2004] Shaw, C. and Lupski, J. (2004). Implications of human genome architecture for rearrangement-based disorders: the genomic basis of disease. *Hum. Mol. Genet.*, 13 Spec No 1:57–64.
- [Vellozo 2007] Vellozo, A. F. (2007). *Alinhamento de seqüências com rearranjos*. PhD thesis, Universidade de São Paulo - DCC - IME- USP, <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04052007-185842>.
- [Vellozo et al. 2006] Vellozo, A. F., Alves, C. E. R., and do Lago, A. P. (2006). Alignment with non-overlapping inversions in $o(n^3)$ -time. In *6th Workshop on Algorithms in Bioinformatics*. Springer. Lecture Notes in Bioinformatics 4175.
- [Verkerk et al. 1991] Verkerk, Pieretti, Sutcliffe, Fu, Kuhl, Pizzuti, Reiner, Richards, Victoria, and Zhang (1991). Identification of a gene (FMR-1) containing a CGG repeat coincident with a breakpoint cluster region exhibiting length variation in fragile X syndrome. *Cell*, 65(5):905–914.