Algoritmos Paralelos Exatos e Otimizações para Alinhamento de Sequências Biológicas Longas em Plataformas de Alto Desempenho

Autor: Edans Flávius de Oliveira Sandes[1], **Orientadora: Alba Cristina M. A. de Melo**[1]

[1]Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Caixa Postal 4466 – 70910-900 – Brasília – DF – Brasil

{edans,albamm}@cic.unb.br

1. Introdução

A comparação de sequências biológicas é uma das operações mais básicas e relevantes na Bioinformática, sendo amplamente utilizada para determinar o grau de similaridade entre as sequências [Mount 2004] e inferir características comuns entre espécies [Durbin et al. 2002]. Essa análise é de extrema importância, pois permite, dentre outros benefícios, identificar genes que causam doenças e determinar eventos evolucionários entre organismos. O resultado de uma operação de comparação de sequências biológicas pode ser (a) um escore que indica a similaridade entre as mesmas ou (b) o escore e o alinhamento, onde uma sequência (ou parte dela) é colocada sobre a outra (ou parte dela), de maneira a evidenciar as regiões de similaridades/diferenças [Mount 2004].

O alinhamento de duas sequências biológicas é então definido como um pareamento entre os caracteres das sequências, com a possível inserção de espaços (*gaps*) entre os caracteres. Chamamos de *matches* os pareamentos entre caracteres iguais e de *mismatches* os pareamentos entre caracteres diferentes. A Figura 1 apresenta um exemplo de alinhamento entre duas sequências, onde os *matches* estão representados pelo símbolo '**:**', *mismatches* pelo símbolo '**.**' e *gaps* pelo símbolo '–'.

```
GCTCACGCCGGTAGTCCCAGCACAGAGGGAG---GAGGCGAACGTATCACCTGAGGTC----------
      :::.:::.:::::::::::.:::::   :::::.:..::::::  ::::::
------GCCTGTAATCCC-GCACTTTGGGAGGCCGAGGTGGGCGCATCAC--GAGGTCAGCGCGAAG
```

Figura 1. Exemplo de alinhamento entre duas sequências.

O problema do alinhamento ótimo de sequências visa encontrar, entre todas as possibilidades de pareamento entre duas sequências, um alinhamento cujo escore seja o maior possível de acordo com o tipo de alinhamento e os parâmetros de pontuação para os *matches*, *mismatches* e *gaps*. Dentre os tipos de alinhamento, o alinhamento *global* considera todos os caracteres de ambas as sequências, o alinhamento *semi-global* permite ignorar o início ou o final de no mínimo uma das sequências e o alinhamento *local* considera *substrings* das sequências. Para a pontuação de *gap*, os modelos mais comuns são o *linear gap* (penalidade proporcional ao comprimento do *gap*) e o *affine gap* (a abertura do gap possui uma penalidade adicional) [Durbin et al. 2002].

2. Motivação

Dentre os algoritmos exatos propostos para resolver o problema de alinhamento ótimo de duas sequências biológicas, podemos citar o Needleman-Wunsch

[Needleman and Wunsch 1970] para alinhamento global ótimo, Smith-Waterman [Smith and Waterman 1981] para alinhamento local ótimo e o Gotoh [Gotoh 1982] para alinhamento global ótimo com o modelo de *affine gap*. Os algoritmos exatos que computam o alinhamento ótimo de sequências biológicas calculam, em geral, uma ou mais matrizes de programação dinâmica de tamanho $m \times n$, onde m e n são os tamanhos das sequências comparadas. Sendo assim, esses algoritmos demandam alto poder de processamento e uma grande quantidade de memória. Por este motivo, o uso de algoritmos exatos foi considerado inviável por muito tempo e, com isso, algoritmos heurísticos surgiram para acelerar o procedimento de comparação de sequências, embora sem garantir a produção do resultado ótimo.

O uso de memória dos algoritmos exatos também é um fator limitante, pois a complexidade quadrática de espaço ($O(mn)$) restringe bastante o tamanho das sequências comparadas. Este fato fica evidente quando comparamos sequências muito longas. Por exemplo, para alinhar sequências de um milhão de pares de bases (MBP) precisaríamos de mais de um terabyte de memória. Myers e Miller [Myers and Miller 1988] utilizaram as ideias de Hirschberg [Hirschberg 1975] para reduzir o uso de memória por meio de técnicas de dividir para conquistar, reduzindo o uso de memória para complexidade linear. Ao se utilizar esse tipo de técnica, o tempo de processamento duplica no pior caso.

Para aumentar o desempenho dos algoritmos exatos e, consequentemente, reduzir o tempo necessário para encontrar alinhamentos ótimos, utilizam-se de técnicas de paralelismo. Para comparar o desempenho das implementações paralelas de algoritmos de comparação de sequências, convencionou-se o uso da métrica CUPS (*Cells Updated Per Second*), que é calculada dividindo-se o tamanho da matriz de programação dinâmica ($m \times n$) pelo tempo de execução em segundos.

Ao iniciar a presente tese em 2011, o melhor desempenho para comparação de sequências biológicas com algoritmos exatos era de 243 GCUPS em ASIC (*Application Specific Integrated Circuit*) [Sarkar et al. 2010] e o desempenho máximo em GPUs (*Graphics Processing Units*) era de 29,7 GCUPS [Liu et al. 2010]. Com 29,7 GCUPS, é impraticável comparar sequências de DNA muito longas, tais como 230 MBP $\times$ 230 MBP, pois a comparação demoraria cerca de 20 dias. Com o desempenho de 243 GCUPS em projeto ASIC, a mesma comparação demoraria menos tempo, mas ainda levaria cerca de 2 dias e meio. Adicionalmente, a maioria das implementações impõe um limite para o tamanho da sequência de busca, o que significa que elas não são capazes de alinhar sequências muito longas. Por exemplo, a maioria das implementações não aceita sequências maiores que 60 mil caracteres, sendo que alguns cromossomos humanos possuem mais de 200 milhões de caracteres.

A principal motivação desta tese é, então, de evoluir o estado da arte de forma que o alinhamento ótimo de duas sequências longas de DNA possa ser executado em tempo viável, permitindo que cromossomos completos sejam comparados em poucas horas ou até mesmo em minutos. Visto que as ferramentas existentes não são capazes de produzir em pouco tempo os alinhamentos ótimos de sequências com mais de 200 milhões de bases, os biólogos ficam limitados ao uso de métodos heurísticos tanto para comparação como para a geração do alinhamento de sequências longas. As ferramentas propostas por meio dessa tese poderão ser utilizadas por pesquisadores para complementar as análises já efetuadas na literatura, mas considerando métodos exatos em vez de heurísticos.

3. Objetivos

A presente tese de Doutorado possui o objetivo geral de desenvolver algoritmos e otimizações que permitam que o alinhamento ótimo de sequências muito longas de DNA seja obtido em tempo reduzido em plataformas de alto desempenho. Consideramos que o processamento é realizado em tempo reduzido se duas sequências na ordem de 200 milhões de pares de base forem alinhadas em menos de duas horas, algo ainda não visto na literatura com métodos exatos.

A principal plataforma escolhida para o desenvolvimento desta tese foi a arquitetura CUDA (*Compute Unified Device Architecture*), das placas de processamento gráfico da NVIDIA. Entretanto, ao longo da pesquisa, o requisito de heterogeneidade foi levado em consideração, de forma que outras plataformas pudessem ser utilizadas para acelerar ainda mais o processamento dos algoritmos envolvidos.

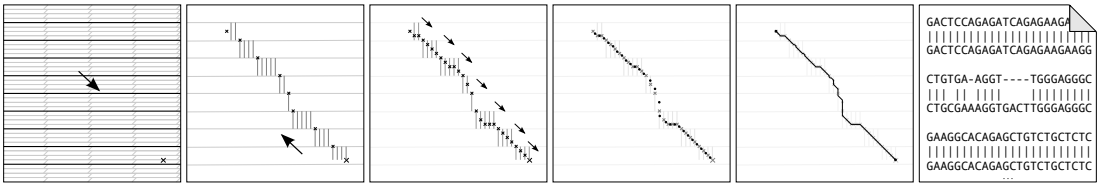
Os objetivos específicos da tese estão elencados a seguir:

- Desenvolver e avaliar de algoritmos paralelos que permitam a recuperação eficiente de alinhamentos ótimos de sequências de DNA longas em GPUs;
- Propor e avaliar teoricamente de método para redução do espaço de busca dos algoritmos propostos, sem prejuízo do resultado ótimo;
- Propor, implementar e avaliar estratégia para execução com múltiplas GPUs dos algoritmos propostos;
- Propor uma arquitetura de *software* que permita a execução dos algoritmos propostos em plataformas homogêneas ou heterogêneas compostas por *multicores* ou GPUs, recuperando alinhamentos ótimos locais, globais e semiglobais;

4. Contribuições

Como contribuição desta tese, algoritmos paralelos e otimizações foram desenvolvidos para permitir a recuperação de alinhamos ótimos entre sequências de DNA. Esses novos algoritmos foram implementados no *software* CUDAlign, que foi evoluído em diversas versões incrementais. A primeira versão, CUDAlign 1.0, foi desenvolvida na dissertação de mestrado [Sandes 2011] do mesmo autor desta tese. A seguir listamos as evoluções de cada uma das versões propostas no escopo desta tese:

- CUDAlign 2.0 [Sandes and Melo 2011]: versão capaz de recuperar alinhamentos locais ótimos entre cromossomos completos em uma GPU utilizando seis estágios (Figura 2). Isso foi possível devido às otimizações propostas nessa tese (*matching* baseado em objetivo, execução ortogonal e divisão balanceada). Atingiu-se 23,1 GCUPS em uma GPU;
- CUDAlign 2.1 [Sandes and Melo 2013]: versão que propõe a otimização *Block Pruning* para alinhamentos locais ótimos, permitindo acelerar o cálculo da matriz de programação dinâmica em mais de 50%, para sequências similares. Nesta otimização, descartam-se áreas da matriz de programação dinâmica que não contribuem para a obtenção do alinhamento ótimo, sem prejuízo do resultado ótimo. Atingiu-se 50,7 GCUPS em uma GPU;
- CUDAlign 3.0 [Sandes et al. 2014b][Sandes et al. 2014a]: versão capaz de comparar sequências longas de DNA em *clusters* com múltiplas GPUs (homogêneas e heterogêneas), permitindo comparar sequências de até 249 milhões de pares de base (MBP). Atingiu-se 1,73 TCUPS (Trilhões de CUPS) com 64 GPUs;



(a) Estágio 1 (b) Estágio 2 (c) Estágio 3 (d) Estágio 4 (e) Estágio 5 (f) Estágio 6

Figura 2. Execução do CUDAlign 2.0 dividida em seis estágios.

- CUDAlign 4.0 [Sandes et al. 2016b]: versão que obtém o alinhamento completo em múltiplas GPUs de maneira eficiente. Por meio do mecanismo de *traceback* especulativo proposto, foi possível obter o alinhamento completo de todos os cromossomos homólogos entre o homem e o chimpanzé, totalizando mais de 500 trilhões de células processadas. Atingiu-se 10,37 TCUPS com 384 GPUs.

  Adicionalmente, as seguintes contribuições foram apresentadas nesta tese.

- Análise teórica da otimização *Block Pruning*: a análise matemática do *Block Pruning* foi feita para identificar quais os aspectos que contribuem ou prejudicam a eficácia deste método;
- Método de balanceamento dinâmico de carga [Sandes et al. 2014c]: para que ambientes distribuídos e não dedicados sejam utilizados para execução do CUDAlign, é necessário que haja um balanceamento dinâmico de carga eficiente. Nesta tese, propusemos um método de balanceamento de carga baseado em agentes, sem necessidade de um elemento central coordenador. Este método foi testado em ambiente simulado, mostrando ser bastante efetivo para o uso proposto;
- Arquitetura MASA [Sandes et al. 2016c]: A arquitetura MASA (*Multi-platform Architecture for Sequence Aligners*) foi projetada para simplificar a portabilidade do CUDAlign para outras arquiteturas. O MASA suporta alinhamentos ótimos locais, globais e semi-globais. Nesta tese, a arquitetura MASA foi aplicada em 4 plataformas distintas: GPU (CUDA), CPU (OpenMP e OmpSs) e Intel Phi (OpenMP). Como contribuição indireta desta tese, a arquitetura MASA foi aplicada na arquitetura OpenCL no escopo da dissertação de mestrado do aluno Marco Antônio C. de Figueiredo Jr. [de Figueiredo Jr. 2015], do mesmo grupo de trabalho do autor desta tese. Neste trabalho, atingiu-se 179,2 GCUPS em uma única GPU da AMD [de Figueiredo Jr. et al. 2015].

A Tabela 1 apresenta um resumo com o desempenho máximo obtido nos testes de cada uma das versões. Com os resultados apresentados, conseguiu-se evoluir o estado da arte em dois aspectos: a) o maior desempenho conhecido para alinhamento ótimo de sequências biológicas tornou-se 10,37 TCUPS com 384 GPUs; b) até onde sabemos, a maior sequência alinhada com métodos exatos passou a ser de 249 MBP.

4.1. Artigos publicados ou aceitos para publicação

O autor desta tese publicou 8 artigos dentro do escopo da tese, todos eles como autor principal. Destes, 5 foram artigos completos em periódicos internacionais, 2 artigos em conferências internacionais e 1 artigo resumido em conferência internacional. As qualificações dos artigos publicados encontram-se abaixo.

Tabela 1. Desempenho das versões do CUDAlign propostas nesta tese

Versão	Saída	Desempenho	Ambiente	Tam. Máx.
CUDAlign 2.0	Alinhamento	23,1 GCUPS	1 × GTX 285	47 MBP
CUDAlign 2.1	Alinhamento	50,7 GCUPS	1 × GTX 560	59 MBP
CUDAlign 3.0	Escore	1,73 TCUPS	64 × M2090	249 MBP
CUDAlign 4.0	Alinhamento	10,37 TCUPS	384 × M2090	249 MBP

Periódicos internacionais:

- **[Sandes and Melo 2013]**: IEEE Transactions on Parallel and Distributed Systems (TPDS) – Capes Qualis CC - A1 (JCR 2,173).
- **[Sandes and Melo 2013]**: Expert Systems with Applications (ESWA) – Capes Qualis CC - A1 (JCR 1,965)
- **[Sandes et al. 2016c]**: ACM Transactions on Parallel Computing (TOPC) – (periódico novo, ainda sem JCR)
- **[Sandes et al. 2016b]**: IEEE Transactions on Parallel and Distributed Systems (TPDS) – Capes Qualis CC - A1 (JCR 2,173): *Paper aceito, DOI=10.1109/TPDS.2016.2515597, aguardando publicação.*
- **[Sandes et al. 2016a]**: ACM Computing Surveys: Capes Qualis CC - A1 (JCR 4,043): *Paper aceito, aguardando publicação.*

Conferências internacionais:

- **[Sandes and Melo 2011]**: IEEE International Parallel & Distributed Processing Symposium (IPDPS) – Capes Qualis CC - A1.
- **[Sandes et al. 2014b]**: IEEE/ACM Symposium on Cluster, Cloud and Grid Computing (CCGrid) – Capes Qualis CC - A1.
- **[Sandes et al. 2014a]**: Symposium on Principles and Practice of Parallel Programming (PPoPP) – Capes Qualis CC - A2: *(artigo resumido)*.

Referências

de Figueiredo Jr., M. A. C. (2015). MASA-OpenCL: Comparação Paralela de Sequências Biológicas Longas em GPU. Master's thesis, Universidade de Brasília, Brasília, Brasil.

de Figueiredo Jr., M. A. C., Sandes, E. F. O., and Melo, A. C. M. A. (2015). Parallel megabase dna sequence comparison with opencl. In *22st International Conference on High Performance Computing, HiPC*, pages 436–445.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (2002). *Biological sequence analysis*. Cambridge University Press.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162(3):705–708.

Hirschberg, D. S. (1975). A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343.

Liu, Y., Schmidt, B., and Maskell, D. (2010). CUDASW++2.0: enhanced Smith-Waterman protein database search on CUDA-enabled GPUs based on SIMT and virtualized simd abstractions. *BMC Research Notes*, 3(1):93.

Mount, D. M. (2004). *Bioinformatics - sequence and genome analysis (2. ed.)*. Cold Spring Harbor Laboratory Press.

Myers, E. W. and Miller, W. (1988). Optimal alignments in linear space. *Computer applications in the Biosciences*, 4(1):11–17.

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.

Sandes, E. F. O. (2011). Comparação Paralela de Sequências Biológicas Longas utilizando Unidades de Processamento Gráfico (GPUs). Master's thesis, Universidade de Brasília, Brasília, Brasil.

Sandes, E. F. O., Boukerche, A., and Melo, A. C. M. A. (2016a). Parallel Exact Pairwise Biological Sequence Comparison: Algorithms, Platforms and Classification. *ACM Computing Surveys (accepted)*.

Sandes, E. F. O. and Melo, A. C. M. A. (2011). Smith-Waterman alignment of huge sequences with GPU in linear space. In *IEEE International Parallel Distributed Processing Symposium*, pages 1199–1211.

Sandes, E. F. O. and Melo, A. C. M. A. (2013). Retrieving smith-waterman alignments with optimizations for megabase biological sequences using gpu. *IEEE Transactions on Parallel and Distributed Systems*, 24(5):1009–1021.

Sandes, E. F. O., Miranda, G., , Melo, A. C. M. A., Martorell, X., and Ayguadé, E. (2014a). Fine-grain parallel megabase sequence comparison with multiple heterogeneous GPUs. In *Proceedings of the 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '14, pages 383–384 (short paper).

Sandes, E. F. O., Miranda, G., Martorell, X., Ayguadé, E., Teodoro, G., and Melo, A. C. M. A. (2016b). CUDAlign 4.0: Incremental Speculative Traceback for Exact Chromosome-Wide Alignment in GPU Clusters. *IEEE Transactions on Parallel and Distributed Systems*, PP(99):1–1.

Sandes, E. F. O., Miranda, G., Martorell, X., Ayguadé, E., Teodoro, G., and Melo, A. C. M. A. (2016c). MASA: a multiplatform architecture for sequence aligners with block pruning. *ACM Transactions on Parallel Computing*, 2(4):28:1–28:31.

Sandes, E. F. O., Miranda, G., Melo, A. C. M. A., Martorell, X., and Ayguade, E. (2014b). CUDAlign 3.0: Parallel Biological Sequence Comparison in Large GPU Clusters. In *IEEE/ACM Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 160–169.

Sandes, E. F. O., Ralha, C. G., and Melo, A. C. M. A. (2014c). An agent-based solution for dynamic multi-node wavefront balancing in biological sequence comparison. *Expert Systems with Applications*, 41(10):4929 – 4938.

Sarkar, S., Kulkarni, G., Pande, P., and Kalyanaraman, A. (2010). Network-on-chip hardware accelerators for biological sequence alignment. *IEEE Transactions on Computers*, 59(1):29–41.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.