Universidade Federal de Pernambuco
Centro de Informática
Graduação em Engenharia da Computação

Representação eficiente em memória de Grafos de *de Bruijn* para uso com *Data Streams*

Aluno: Augusto Sales de Queiroz (asq@cin.ufpe.br)

Orientador: Paulo Gustavo Soares da Fonseca (paguso@cin.ufpe.br)

Área: Algoritmos e Estruturas de Dados

Recife,
22 de Fevereiro de 2022

Sumário

1 Resumo

O uso de memória causa um dos maiores *bottlenecks* nos *softwares* de reconstrução de genomas (*assemblers*). Um dos principais alvos de otimização nesse tipo de programa é a representação de um Grafo de *de Bruijn* (GdB), uma vez que essa estrutura é amplamente utilizada por esses *assemblers*, apesar de requerer um alto custo em memória (na ordem de GBs). Este trabalho se propõe a analisar soluções atuais para esse problema, e propor um novo modelo de representação de GdBs eficiente em memória. Além disso, será dado um foco em estruturas de dados baseadas em *sketches*, permitindo que as leituras não precisem estar todas disponíveis simultaneamente, reduzindo, também, os requisitos de armazenamento e, potencialmente, tempo necessário para a reconstrução do genoma.

2 Possíveis Avaliadores

- Nivan Roberto Ferreira Junior (nivan@cin.ufpe.br)
- Gustavo Henrique Porto de Carvalho (ghcp@cin.ufpe.br)

3 Cronograma

Atividade	Fev	Mar	Abr	Mai
Revisão Bibliográfica	X			
Desenvolvimento do Modelo	X	X	X	
Análise experimental comparativa de desempenho		X	X	
Escrita do Texto			X	X
Preparação da Defesa				X

4 Assinaturas

Paulo Gustavo Soares da Fonseca
Orientador

Augusto Sales de Queiroz
Orientando

5 Contexto

As tecnologias de sequenciamento de genomas de segunda geração revolucionaram o acesso a pesquisa genomica, reduzindo os custos e esforços necessários para esse tipo de estudo[6]. À medida que esse tipo de pesquisa se tornou mais acessível, surgiu, também, a importância de desenvolvimento de métodos de reconstrução de sequências de genoma, partindo dos dados gerados pelos sistemas de gerenciamento, que fossem mais eficientes. A montagem de sequências é um problema *NP-difícil* [7], de forma que tentativas de otimização do processo de reconstrução não são muito exploradas. Assim, muitos dos esforços de otimização almejaram, então, a redução no uso de memória, em particular para a representação de Grafos de *de Bruijn* (GdBs), estrutura que ganhou grande relevância nos *assemblers* baseados nas novas tecnologias de sequenciamento[3][2] graças ao fato destas produzirem uma enorme quantidade de leituras (na ordem de milhões de leituras) curtas (normalmente entre 25 e 400bp[1]).

Diversas representações para os GdBs já foram propostas[2] visando, principalmente, obter uma melhor eficiência em memória enquanto garantem uma maior exatidão da informação (pois algumas representações aceitam alguma taxa de falsos positivos em troca de um melhor aproveitamento do espaço, como é o caso da representação introduzida em [8]).

Uma perspectiva ainda pouco explorada é a de representações dos GdBs em configuração de *data stream*, na qual os dados não são todos disponibilizados simultânemanete, sendo adicionados na estrutura a medida que as leituras são realizadas. O único trabalho encontrado nesse sentido foi o *assembler FastEtch*[5], que faz uso de um *sketch CountMin*. Esse tipo de exploração tem potencial para impactar tanto o requerimento de memória do sistema, quanto o desempenho em tempo necessário para a montagem do genoma, uma vez que o sistema poderia montar o GdB a medida que as leituras são realizadas, não precisando que todas elas sejam armazenadas em disco e disponibilizadas simultâneamente.

[1] bp, ou *base pairs*, são os pares de base $\{A, C, G, T\}$ que compõem uma sequência de DNA

6 Objetivo

O objetivo deste trabalho é a explorar métodos sucintos de representação de grafos de *de Bruijn*, bem como desenvolver uma nova forma de representação eficiente em memória na forma de *sketch* para uso com *Data Streams*.

Referências

[1] R. Chikhi and G. Rizk. Space-efficient and exact de bruijn graph representation based on a bloom filter, 2013. URL `http://www.almob.org/content/8/1/22`.

[2] R. Chikhi, A. Limasset, S. Jackman, J. Simpson, and P. Medvedev. On the representation of de bruijn graphs. 1 2014. URL `http://arxiv.org/abs/1401.5383`.

[3] T. C. Conway and A. J. Bromage. Succinct data structures for assembling large genomes. *Bioinformatics*, 27:479–486, 2 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btq697.

[4] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *Journal of Algorithms*, 55:58–75, 4 2005. ISSN 01966774. doi: 10.1016/j.jalgor.2003.12.001.

[5] P. Ghosh and A. Kalyanaraman. A fast sketch-based assembler for genomes. pages 241–250. Association for Computing Machinery, Inc, 10 2016. ISBN 9781450342254. doi: 10.1145/2975167.2975192.

[6] M. Imelfort and D. Edwards. De novo sequencing of plant genomes using second-generation technologies, 2009. ISSN 14675463.

[7] P. Medvedev, K. Georgiou, G. Myers, and M. Brudno. Computability of models for sequence assembly. In R. Giancarlo and S. Hannenhalli, editors, *Algorithms in Bioinformatics*, pages 289–301, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74126-8.

[8] J. Pell, A. Hintze, R. Canino-Koning, A. Howe, J. M. Tiedje, and C. T. Brown. Scaling metagenome sequence assembly with probabilistic de bruijn graphs. *Proceedings of the National Academy of Sciences*, 109(33):13272–13277, 2012. ISSN 0027-8424. doi: 10.1073/pnas.1121464109. URL `https://www.pnas.org/content/109/33/13272`.