ORDENAÇÃO DE FRAGMENTOS DA MONTAGEM DE GENOMAS COM BASE EM MÚLTIPLAS REFERÊNCIAS UTILIZANDO ALGORITMO GENÉTICO

MICHEL S. PEDROSO[1]; FREDERICO S. KREMER[2]; MARILTON S. DE AGUIAR[3]

[1]*Universidade Federal de Pelotas – mspedroso@inf.ufpel.edu.br*
[2]*Universidade Federal de Pelotas* – fredericok.cdtec@ufpel.edu.br
[3]*Universidade Federal de Pelotas– marilton@inf.ufpel.edu.br*

1. INTRODUÇÃO

Com o avanço da tecnologia, se tornou necessário abordar novas formas de realizar sequenciamento de DNA, como Next *Generation Sequencing* (NGS). Existem algumas plataformas de sequenciamento NGS que são capazes de gerar grandes volumes de dados por rodada, o que permite uma grande variedade de aplicações, como sequenciamento de genomas (procarióticos e eucarióticos), transcriptomas e metagenomas (LIU et al., 2012).

Tradicionalmente, a montagem de genomas é realizada com base em alguns passos, como citado por (KISAND & LETTIERI, 2013): 1) alinhamento de todos fragmentos; 2) construção de um grafo que represente as sobreposições; 3) identificação do melhor conjunto mínimo de caminhos independentes; 4) montagem das *contigs* através de um consenso das leituras. Este tipo de algoritmo, apesar de acurado, se torna inaplicável para dados de sequenciamento de nova geração (NGS), uma vez que as plataformas NGS geram milhões de fragmentos por rodada, tornando a realização dos alinhamentos extremamente lento (MILLER et al., 2010).

Devido ao grande volume de dados obtido a partir destas tecnologias, foi necessário desenvolver ferramentas e algoritmos que realizassem a análise destas informações de forma rápida e eficiente, surgindo ferramentas de ordenamento de *contigs* como ordenamento simples (ABACAS) e múltiplos (MEDUSA) (ASSEFA et al., 2009; BOSI et al., 2015).

O objetivo desta ferramenta consiste em criar um algoritmo genético para realizar o ordenamento dos fragmentos (*contigs*) obtidos através da montagem de genomas, utilizando múltiplos genomas de referência (informações consolidadas sobre o genoma alvo) onde permite o ordenamento em tempo satisfatório com uma porcentagem alta de acurácia.

Com o grande número de *contigs* (sequências não-ordenadas) e a dificuldade de gerar um conjunto satisfatório de *scaffolds* (sequências parcialmente ordenadas), as ferramentas que não utilizam abordagens inteligentes costumam realizar a tarefa em um tempo muito custo com uma acurácia boa ou em tempo satisfatório, porém com baixa acurácia. A utilização de algoritmos inteligentes (neste caso, Algoritmo Genético) possui características que permitem gerar ótimos resultados, por serem algoritmos de busca e otimização; e, utilizarem operadores baseados em princípios de evolução e genética natural (DUBITZKY & AZUAJE, 2004).

2. METODOLOGIA

Todo o desenvolvimento da ferramenta foi realizado utilizando a linguagem Python (https://www.python.org/), em conjunto com o pacote BioPython (Cock et al., 2009), permitindo um desempenho mais rápido e eficiente para a ferramenta.

O desenvolvimento deste trabalho consiste em primeiramente realizar o alinhamento (BLAST ou MUMMER) (ALTSCHUL et al., 1990; CAMACHO et al., 2009; KURTZ et al., 2004) do genoma a ser analisado com todos os arquivos de genomas *drafts*, podendo ser apenas um genoma rascunho ou múltiplos.

Após este procedimento de alinhamento é necessário realizar uma filtragem e alterações no formato do arquivo gerado, adaptando o mesmo para o formato aceito pelo algoritmo genético. É relevante ressaltar que quando a máquina utilizada permite processamento *multithreading*, tanto o alinhamento como a etapa de *parser*, será realizado em paralelo, podendo também definir a quantidade de *threads* a serem utilizadas.

Quando a etapa do *parser* é finalizada, os dados retirados do alinhamento são passados para o algoritmo genético. O desenvolvimento consiste na utilização dos operadores dos AGs (população inicial, aptidão, seleção, cruzamento, mutação) cada operador foi definido da seguinte maneira: população inicial com tamanho de 350 indivíduos, aptidão calculada sobre os *matchs* do alinhamento, seleção é feita pelo método da roleta, cruzamento de recombinação de único ponto com probabilidade 20% e mutação com inversão das fitas do alinhamento com probabilidade 5%. Uma representação do fluxograma do algoritmo proposto está indicada na Figura 1.

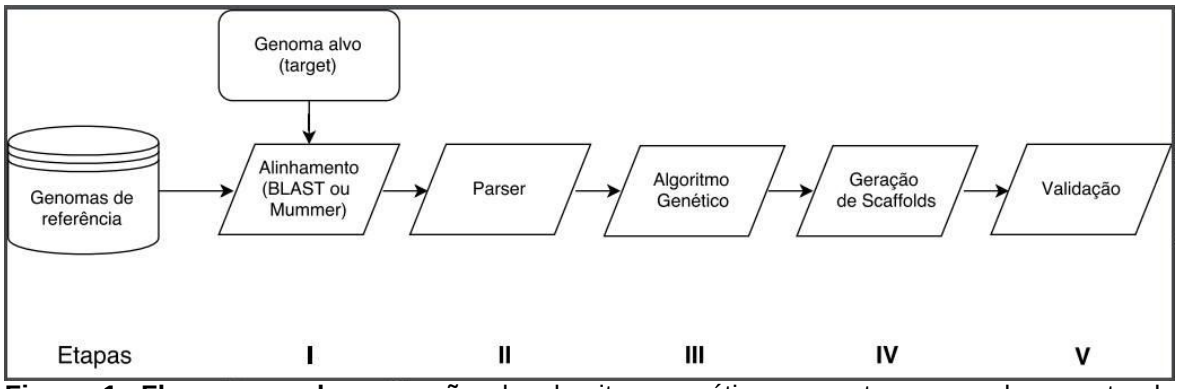

Figura 1. Fluxograma de execução do algoritmo genético proposto para ordenamento de *contigs*: (I) alinhamento de *contigs* do genoma de interesse "*target*" contra os demais genomas de interesse através das ferramentas BLAST ou Mummer; (II) processamento dos resultados do alinhamento; (III) identificação do ordenamento ótimo através do algoritmo genético; (IV) geração de *scaffolds*; (V) validação dos resultados.

Os testes da ferramenta foram realizados utilizando como genoma alvo resultados da montagem de novo de um genoma de *Leptospira borgpetersenii* cepa 4E (GenBank: CP015814.1, CP015815.1) gerados a partir das ferramentas CISA (LIN and LIAO, 2013), BWA (LI and DURBIN, 2009), Velvet (ZERBINO & BIRNEY, 2008) e Edena (HERNANDEZ et al., 2008), sendo este constituído por 541 *contigs* não ordenadas. Os genomas usados para ordenamento consistem em 21 genomas de referências do genoma de *L. Borgpetersenii* obtidos a partir do GenBank (www.ncbi.nlm.nih.gov/genbank/).

3. RESULTADOS E DISCUSSÃO

A ferramenta proposta ainda está em processo de implementação, entretanto, sua versão atual já possui integralmente as etapas de alinhamento (com suporte para as ferramentas BLAST e MUMMER) e *parsing*. O algoritmo genético ainda está sendo ajustado, uma vez que são necessários diversos testes para definir as configurações ótimas para alguns parâmetros. As próximas etapas do desenvolvimento envolvem a finalização do algoritmo genético e a comparação dos resultados gerados pelo programa com os gerados por ferramentas já disponíveis, como MeDuSa (BOSI et al. 2015).
A partir dos dados utilizados no teste, o *parsing* foi capaz de extrair 300 regiões de adjacência entre *contigs*, que posteriormente podem ser utilizadas no *fitness* do algoritmo genético para avaliar a consistência das *scaffolds* que foram geradas.

4. CONCLUSÕES

No presente trabalho foi apresentada a implementação preliminar de uma ferramenta baseada em algoritmo genético para ordenamento de *contigs* a partir de alinhamento contra múltiplos genomas de referência. Apesar de ainda não finalizado, o presente projeto demonstra-se como promissor por aplicar uma abordagem de computação evolutivo no contexto do sequenciamento de DNA de nova geração.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ALTSCHUL, S. F; GISH, W; MILLER, W; MYERS, E. W; LIPMAN, D. J. *Basic local alignment search tool*. **Journal of Molecular Biology**, V. 215, p. 403-410, 1990.

ASSEFA, S; KEANE, T. M; OTTO, T. D; NEWBOLD, C; BERRIMAN, M. ABACAS: *algorithm-based automatic contiguation of assembled sequences*. Bioinformatics, V. 25, 1968–1969, 2009.

BOSI, E.; DONATI, B.; GALARDINI, M.; BRUNETTI, S.,;SAGOT, M.-F.; LIÓ, P., CRESCENZI, P.; FANI, R.; FONDI, M. *MeDuSa: a multi-draft based scaffolder*. **Bioinformatics**, V. 31, p2443–2451, 2015.

CAMACHO, C.; COULOURIS, G.; AVAGYAN, V.; MA, N.; PAPADOPOULOS, J.; BEALER, K.; MADDEN, T.L. *BLAST+: architecture and applications*. **BMC Bioinformatics** V. 10, 2009.

COCK, P.J.A. ANTAO, T.; CHANG, J.T., CHAPMAN, B.A.; COX, C.J.; DALKE, A.; FRIEDBERG, I.; HAMELRYCK, T.; KAUFF, F.; WILCZYNSKI, B.; DE HOON, M.J.L. *Biopython: freely available Python tools for computational molecular biology and bioinformatics*. **Bioinformatics**, V. 25, p1422-1423, 2009.

DUBITZKY, W.; AZUAJE, F. **Artificial Intelligence Methods And Tools For Systems Biology, Computational Biology**. *Springer* Netherlands, Dordrecht, 2004.

HERNANDEZ, D.; FRANÇOIS, P.; FARINELLI, L.; OSTERÁS, M. SCHRENZEL, J. *De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer***. Genome Research**, V. 18, p802-809, 2008.

KISAND, V.; LETTIERI, T; *Genome sequencing of bacteria: sequencing, de novo assembly and rapid analysis using open source tools*. . **BMC Genomics**, V. 14, 2013.

KURTZ, S.; PHILLIPPY, A.; DELCHER, A.L.; SMOOT, M.; SHUMWAY, M.; ANTONESCU, C.; SALZBERG, S.L. *Versatile and open software for comparing large genomes*. **Genome Biology**, V. 5, 2004.

LIN, H; DURBIN, R.. F*ast and accurate short read alignment with Burrows-Wheeler transform*. **Bioinformatics**, V. 25, 1754-60, 2009.

LIN, S.H; LIAO, Y.C. *CISA: contig integrator for sequence assembly of bacterial genomes.* **PLoS One**, V. 8, 2013.

LIU, L; LI, Y; LI, S; HU, N; HE, Y; PONG, R; LIN, D; LU, L; LAW, M. *Comparison of next-generation sequencing systems***. Journal of Biomedical Biotechnology**, V. 2012, 2012.

MILLER, J.R; KOREN, S; SUTTON, G. *Assembly algorithms for next-generation sequencing data*. **Genomics**, V. 95, p315,327, 2010.

ZERBINO, D.R; BIRNEY, E. *Velvet: algorithms for de novo short read assembly using de Bruijn graphs*. **Genome Research**, V. 18, p821-829, 2008.