Análise comparativa e adaptação de algoritmos computacionais para comparação, classificação e armazenamento de estruturas moleculares extraídas de plantas do semiárido.

Tayane Leite Cerqueira[1] e Angelo Amâncio Duarte[2]

1. Ex-Bolsista PROBIC, Graduanda em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: engcomp.tayane@gmail.com

2. Angelo Amâncio Duarte, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: angeloduarte@ecomp.uefs.br

PALAVRAS-CHAVE: Redes Complexas, Análise Filogenética, Bioinformática.

INTRODUÇÃO

A Análise Filogenética é uma área da Biologia que tem crescido consideravelmente e despertado bastante interesse dos pesquisadores. Inicialmente usada para elucidar relações hierárquicas, a filogenética se expandiu, sendo usada hoje com inúmeros objetivos: criar novos grupos taxonômicos; reconstruir filogenias de organismos com representantes de diferentes áreas geográficas; investigar a evolução de espécies que interagem entre si; compreender melhor a dinâmica de populações, entre outros (SCHNEIDER, 2007).

A teoria das Redes Complexas vem sendo desenvolvida por físicos e matemáticos nas últimas décadas e é considerada uma das teorias mais modernas da ciência contemporânea. Ela é o fruto da união da Teoria dos Grafos e da Mecânica Estatística (DINIZ, 2010). O termo redes complexas refere-se a um grafo que apresenta uma estrutura topográfica não trivial, composto por um conjunto de vértices (nós) que são interligados por meio de arestas (BARBASI, 1999).

A presença de estruturas modulares locais, conhecidas como comunidades, é uma característica notável em redes complexas. Vários métodos têm sido aplicados na análise filogenética, dentre eles o de detecção de comunidades. O grupo de Física Estatística e Sistemas Complexos (FESC) da Universidade Federal da Bahia (UFBA) propôs um algoritmo para identificar a estrutura de uma comunidade com base no conceito de distância entre redes complexas e aplicá-lo ao problema específico de recuperação de informações úteis que podem ser utilizadas para inferir relações filogenéticas (ANDRADE et al., 2011). Esse algoritmo, que está descrito no artigo *Detecting Network Communities: An Application to PhylogeneticAnalysis* (ANDRADE et al., 2011) e foi publicado pela *PLoSComputationalBiology* (PLOS..., 2013), propõe a construção de uma rede complexa com base nos índices de similaridade entre as sequências de proteína, sendo essa a rede a ser submetida ao algoritmo de detecção de comunidades proposto.

MATERIAIS E MÉTODOS

Primeiramente foi realizada uma revisão bibliográfica dos conceitos teóricos importantes para o desenvolvimento do projeto. A primeira parte da pesquisa foi empenhada no estudo de como funciona o método de análise filogenética baseada na teoria das redes complexas implementado pelo FESC. Para isso, foi realizada uma leitura dos artigos

produzidos e publicados pelos autores do método. Além disso, foram realizadas algumas reuniões com os integrantes do grupo em questão com objetivo de facilitar o entendimento de assuntos pertinentes a bioinformática, de como o método foi implementado, bem como suas limitações.

Embora o método em questão tenha se mostrado uma poderosa ferramenta para a detecção de comunidades aplicado à análise filogenética, este apresenta um baixo desempenho computacional quando submetido a redes com muitos nós e vértices, sendo em alguns casos inviável a utilização. Uma vez que uma das etapas do método é o alinhamento de sequências, e, na grande maioria das vezes utiliza-se um grande número de sequências, foi iniciada uma revisão do tipo de alinhamento e o software utilizado para realizar o mesmo no método.

O software utilizado até então era o NCBI-BLASTP, cujo tipo de alinhamento utilizado é o alinhamento local. A fim de se obter um maior conhecimento a cerca de como o algoritmo estava sendo executado em um ambiente computacional, foi iniciada a execução dos programas que foram desenvolvidos pelo FESC. Uma vez que o aumento de desempenho do método é o objetivo deste projeto, foi então realizada uma pesquisa na Internet sobre outros softwares que também são utilizados no alinhamento de sequências. Os softwares encontrados foram:*ClustalW* (CLUSTALW..., 2013), *CUDA-BLASTP* (CUDA-BLASTP..., 2013) e *GPU-BLAST* (GPU-BLAST..., 2013).

Em paralelo as tentativas de instalação do GPU-BLAST, foi dado início a implementação em linguagem C do algoritmo que realiza a geração da matriz de similaridade a partir dos resultados gerados pelo NCBI-BLAST. O programa desenvolvido tem como entrada o arquivo gerado pelo NCBI-BLAST, que contém o resultado do alinhamento realizado entre sequências. Dentre outros dados, ele contém o índice de similaridade (*scores*) entre as sequências, que é o que será utilizado para a geração da matriz de similaridade. A matriz de similaridade armazena valores percentuais referentes aos *scores* entre as sequências processadas pelo NCBI-BLAST.

Após a conclusão do programa que gera a matriz de similaridade, que possui como saída um arquivo de texto que contém os valores da matriz, foi iniciada a implementação, também em linguagem C, do programa que gera as matrizes de adjacência. Esse programa foi implementado como continuação do código fonte referente à geração da matriz de similaridade, uma vez que os mesmos compartilham os mesmos dados para processamento.

Em seguida foi desenvolvido o programa que gera a matriz de cores, que é um dos produtos gerados pelo algoritmo proposto pelo método do FESC. A matriz de cores é a representação gráfica da matriz de similaridade (Figura 1). O programa foi desenvolvido em linguagem C tendo como entrada um arquivo de texto contendo a matriz de similaridade e gera como saída um arquivo de imagem no formato *bitmap*.

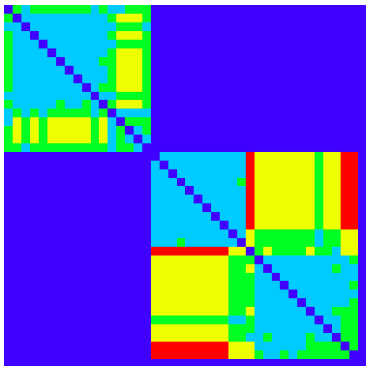

Figura 1. Matriz de cores gerada pelo programa desenvolvido.

RESULTADOS E DISCUSSÃO

O CUDA-BLASTP, apesar de se mostrar bastante promissor em relação ao poder de desempenho no alinhamento de sequências, apresentou diversos erros durante as tentativas de instalação. Além disso, apesar de diversas tentativas de contato com o seu desenvolvedor, nenhuma resposta foi recebida o que levou a interromper as atividades aplicadas no mesmo. Por outro lado, a instalação GPU-BLAST foi realizada com êxito. Dessa maneira, foram realizados teste de comparação de tempo de execução entre o mesmo e o NCBI-BLAST.

Nos testes realizados no GPU-BLAST, para um arquivo de entrada contendo 827 sequências, o tempo de execução obtido foi de 5m7.875s. Utilizando o mesmo ambiente computacional e o mesmo arquivo de entrada, obteve-se um tempo de execução de 4m8.575s no NCBI-BLAST. Visto que o NCBI-BLAST apresentou um tempo de execução menor em relação ao GPU-BLAST, as atividades do projeto foram prosseguidas utilizando este software.

O programa que gera a matriz de similaridade foi implementado com sucesso para ambiente Linux, utilizando como entrada um arquivo de texto gerado após a execução do NCBI-BLAST. O programa é executado através de linhas de comando, onde deve ser informado na mesma: o tipo do arquivo (no formato FASTA), a localização do mesmo, a localização do arquivo resultante do NCBI-BLAST e por fim o nome e a localização do arquivo de texto que será gerado contendo a matriz de similaridade.

O programa que gera as matrizes de adjacência foi implementado com sucesso e utiliza como entrada a matriz de similaridade. Essas matrizes são utilizadas por etapas subsequentes no método do FESC.

O programa que gera a matrizes de similaridade e as matrizes de adjacência se mostrou mais eficiente e de maior facilidade de execução quando comparado ao que foi implementado pelo FESC. Em vez de executar uma série de scripts utilizando bancos de dados (além de um sistema de gerenciamento do mesmo) e pedaços de programas escritos em outras linguagens, agora é possível utilizar somente um programa, sem necessidade de programas auxiliares e ferramentas adicionais, para realizar essa tarefa.

O programa que gera a matriz de cores também foi implementado com sucesso e o arquivo de imagem gerado exibe a matriz de cores de maneira satisfatória, porém é necessário que o mesmo seja aperfeiçoado de tal forma que, juntamente com a matriz de cores, seja exibida uma escala de cores com o objetivo de informar ao usuário os valores de similaridade que cada cor da matriz representa.

Desta maneira, pode-se dar continuidade a implementação dos demais passos que constituem o método do FESC onde, ao final, pode-se obter um único programa que execute o método com eficiência e que ofereça uma interface amigável ao usuário.

CONCLUSÃO

Com o intuito de oferecer melhorias no método de Análise Filogenética com base na Teoria de Redes Complexas desenvolvido pelo FESC, os resultados obtidos através da pesquisa foram satisfatórios para as primeiras etapas do método. Entretanto, para que haja uma melhoria significada na execução do método, é necessário que haja uma revisão do mesmo em sua totalidade.

Dessa maneira, espera-se que as atividades seguintes implementem os passos subsequentes do método objetivando otimizar o sistema desenvolvido pelo FESC. Para isso, sugere-se a implementação do sistema utilizando a linguagem CUDA para que o método possa ser executado utilizando-se o poder de processamento da GPU e o problema de

desempenho enfrentado no estudo de grandes comunidades seja solucionado, tornando-o assim, viável para o processamento de redes com grande número de nós e vértices.

REFERÊNCIAS

ANDRADE, R. F. S.; ROCHA-NETO, I. C.; SANTOS, L. B. L.; SANTANA, C. N.; DINIZ, M. V. C. et al. *Detecting Network Communities: An application to Phylogenetic Analysis. PLoS Computational Biology*, v. 7, n. 5. doi:10.1371/journal.pcbi.1001131. 2011.

SCHNEIDER, H. Métodos de Análise Filogenética - Um Guia Prático. [S.l.]: Holos, 2007.

DINIZ, M. V. C. Análise Computacional de Sintases da Quitina de FungsBasidiomicetos. Dissertação de Mestrado — Universidade Estadual de Feira de Santana, Feira de Santana, 2010.

BARABASI, A. L. and Albert, R. (1999a). *Emergence of scaling in random networks.Science*, pág. 286–509.

PLOS – *Computacional Biology*. Último acesso em 07 de Maio de 2013. Disponível em: <www.ploscompbiol.org>.

CLUSTALW. Último acesso em 07 de Maio de 2013. Disponível em <http://www.clustal.org/clustal2/>

CUDA-BLASTP. Último acesso em 08 de Maio de 2013. Disponível em <https://sites.google.com/site/liuweiguohome/software>

GPU-BLAST Homepage. Último acesso em 08 de Maio de 2013. Disponível em <http://eudoxus.cheme.cmu.edu/gpublast/gpublast.html>