Aplicação *Web* de Processamento, *Clustering* e Visualização de Dados de Genes de *Workflows* Científicos

Alexandre A. P. Heine[1], **Eduardo P. S. Santos**[2],
Marcelo F. Lima[2], **Sérgio Lifschitz**[1]

[1]Departamento de Informática – (PUC-Rio) - Rio de Janeiro - RJ

[2]Departamento de Bioquímica - (UFRRJ) - Seropédica - RJ

{xandeaph,eduardop.dev,marcflima}@gmail.com, sergio@inf.puc-rio.br

Resumo. *Este trabalho propõe uma aplicação web de apoio a dados provenientes de workflows científicos de genes diferencialmente expressos (DEGs) que envolvem análises de sequenciamento de RNA. Assim, procura-se contribuir com formas de manuseio desses dados, permitindo a formação de um grafo de genes, por meio de dados de interação de genes, assim como a criação de subgrafos e clusters representativos, utilizando-se de informações obtidas de dados armazenados em arquivos de string. Esses arquivos são processados conforme são realizados novos pedidos na aplicação e, então, armazenados em um banco de dados relacional, possibilitando a pesquisadores visualizar e manusear grafos de genes para seus estudos.*

1. Introdução

Estudos na área de Ciências Biológicas tem envolvido uma quantidade de dados cada vez maior, sendo, assim, necessário o uso de técnicas de manuseio de dados em larga escala. Assim, *workflows* científicos são um exemplo de ferramentas que permitem trabalhar com dados nessa escala, retornando resultados importantes para análises de pesquisadores.

Apesar disso, os resultados desses *workflows* precisam ser, posteriormente, processados de forma que os dados tenham utilidade nas análises dos pesquisadores. Logo, aplicações para processamento, armazenamento e visualização desses dados são necessárias, para auxiliar os pesquisadores em seus estudos.

Dada essa necessidade, quando os dados envolvem também estruturas de dados mais complexas para se trabalhar, como grafos e árvores, também é preciso o uso de algoritmos, a fim de simplificar sua exibição, visto que o consumo da memória e do processador se tornam um desafio para a análise desses dados.

2. Contexto e Motivação

O estudo de genes em plantas permite os pesquisadores estudarem o efeito de modificações genéticas nesses organismos, tanto no sistema de defesa para prevenção de pragas, quanto na produção de alimentos [Pinto et al. 2011].

Dentro desse contexto, este trabalho propôs resolver um problema envolvendo dados de genes, contidos em arquivos de *string*, realizar seu processamento, armazenamento e, então, simplificação em *clusters* representativos. Esses *clusters*, para uma dada distância máxima escolhida por um pesquisador, são formados pelo menor conjunto de

genes que agrupam o maior número de genes diferencialmente expressos (DEGs), aqueles que demonstram uma diferença em contagem de células estatisticamente significante entre duas condições experimentais [Anjum et al. 2016]. O objetivo do estudo desses genes é identificar diferenças biológicas entre estados saudáveis e doentes de um determinado organismo [Rodriguez-Esteban and Jiang 2017].

Assim, neste trabalho, foi implementada uma ferramenta, para dar suporte a análises de grafos de genes. As informações e as interações desses genes são provenientes de dados públicos em *string* obtidos do *String-DB*[1] para diferentes organismos.

Em específico para esta pesquisa, foram processados dados da planta *Arabidopsis thaliana*, apesar de a aplicação em si ter sido pensada para qualquer organismo que utilize de arquivos no formato dessa base de dados.

Quanto aos arquivos de análises, esses são obtidos como resultado de um *workflow* científico chamado de *RNA-seq for DEs* [Anjum et al. 2016], que visa obter dados de DEGs, como contagens de cada amostra e dados que indiquem a expressividade dos genes estatística e quantitativamente, informações importantes para a análise dos pesquisadores.

3. Trabalhos Relacionados

Algumas aplicações já possibilitam a visualização de grafos de genes e obtenção de algumas informações, como o *Cytoscape*[2] e o *String-DB*.

O *Cytoscape* é uma interface gráfica bastante rebuscada de visualização de grafos de forma interativa, capaz também de criar *clusters*. Ao contrário deste trabalho, o *clustering* implementado por esta ferramenta somente realiza o agrupamento de genes que têm maior interação entre si, não resolvendo o problema proposto. Além disso, para que pudesse ser utilizada essa ferramenta diretamente, também seria necessário processar os arquivos para montagem dos grafos, além de fazer o processo de *clustering* à parte, para atender o problema.

O *String-DB*, por outro lado, contém dados de genes e monta subgrafos próprios a partir de um gene inicial pesquisado. Ele permite a obtenção de *clusters* de forma idêntica ao *Cytoscape*, não resolvendo o problema proposto. É importante mencionar que o *String-DB* mostra dados de análises de diversas proveniências, mas não contém dados de genes diferencialmente expressos específicos, pois isso varia de acordo com a pesquisa.

Assim, optou-se por fazer um sistema próprio que atendesse às demandas levantadas, integrando *scripts* de pré-processamento e *clustering* a uma aplicação *web*.

4. Projeto e Implementação

A partir do contexto e da motivação, foi planejado um sistema contendo: um banco de dados, responsável por integrar as informações e as requisições das partes do sistema; uma aplicação *web* para a requisição do pré-processamento de arquivos de genes e de análises, assim como a requisição da criação de *clusters* e exibição desses em um painel interativo; um *script* de pré-processamento dos dados de genes; um *script* de pré-processamento dos dados das análises; e um *script* para realização do processo de *clustering*.

[1] https://string-db.org/
[2] https://cytoscape.org/

Foi utilizada a linguagem de programação Python[3] como padrão para todo o sistema. Assim, os *scripts* de pré-processamento utilizaram a biblioteca *Pandas*[4] para manuseio e tratamento dos dados, fazendo uso de conceitos similares ao de tabelas do banco de dados. O *script* de *clustering* utilizou da biblioteca *NetworkX*[5] para manuseio de maneira otimizada de grafos e árvores, a partir do uso de um tipo de dado chave-valor. Por último, a aplicação *web* foi feita com a *framework* Django[6], com sua *frontend* em Javascript[7], usando os pacotes DataTables[8], para exibição de tabelas com funcionalidades de filtro e paginação; e Cytoscape.js[9]. O Cytoscape.js é um pacote feito à parte do Cytoscape, que permite exibir grafos de forma interativa em aplicações *web*, apesar de não ter todas as ferramentas do outro *software* integradas, sendo responsabilidade do desenvolvedor disponibilizá-las por programação. Como banco de dados, utilizou-se o Sistema Gerenciador de Banco de Dados Relacional PostgreSQL[10].

4.1. Banco de Dados

Conforme na Figura 1, o sistema funciona da seguinte forma: pesquisadores realizam análises, que identificam determinados genes do sistema como diferencialmente expressos (relacionamento entre "*Analysis*" e "*Gene*"). Esses genes analisados realizam interações com outros genes, formando, assim, um grafo de genes (autorrelacionamento da entidade *Gene*). A partir das análises podem ser feitos *clusters* representativos (subgrafos), os quais são determinados pela distância máxima procurada e o conjunto de genes que agrupam.

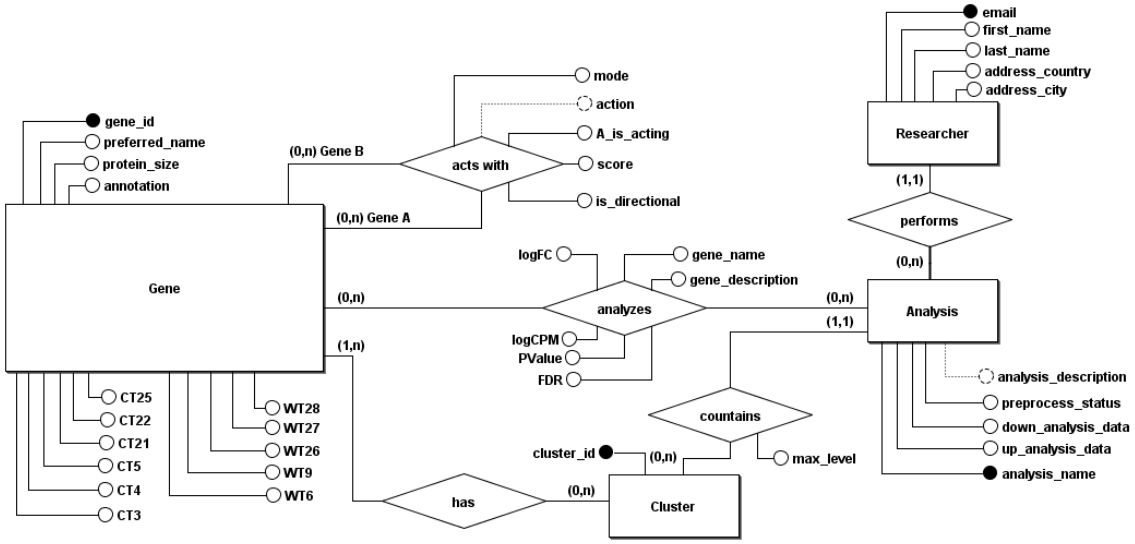

Figura 1. Esquema conceitual (entidade-relacionamento)

Desse modo, as tabelas do banco foram formadas de acordo com o esquema lógico simplificado, presente na Figura 2. Como ilustrado, no banco também foram represen-

[3]https://www.python.org/
[4]https://pandas.pydata.org/
[5]https://networkx.org/
[6]https://www.djangoproject.com/
[7]https://www.javascript.com/
[8]https://datatables.net/
[9]https://js.cytoscape.org/
[10]https://www.postgresql.org/

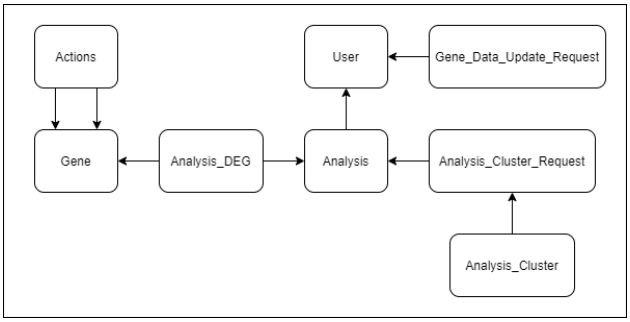

Figura 2. Modelagem relacional simplificada

tadas as requisições feitas pelos usuários ao sistema. Usuários administradores têm permissão de atualizar a base de genes com dados no formato daqueles obtidos do *String-DB*, por meio de uma requisição, armazenada em "*Gene_Data_Update_Request*". Além disso, os DEGs também são representados como uma tabela que referencia "*Analysis*" e "*Gene*". Por uma razão de desempenho na formação do grafo de genes no painel, decidiu-se armazenar os genes dos *cluster* como um vetor de genes na tabela "Analysis_Cluster", do mesmo modo como foi armazenado um JSON[11], já estruturado para exibição do *cluster* em um painel. Da mesma forma que a requisição de atualização de genes, a requisição de *clusters* foi representada como uma tabela no banco ("Analysis_Cluster_Request").

4.2. *Script* de Pré-processamento de Dados de Genes

O *script* de pré-processamento de dados de genes tem por objetivo obter dados disponibilizados no sistema por usuários administradores, para atualizar a tabela de genes e interações entre eles.

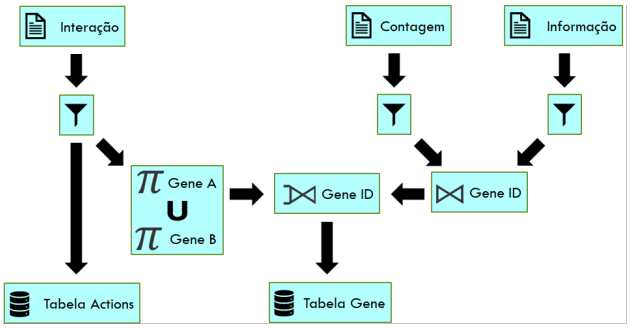

Figura 3. Esquema do pré-processamento dos arquivos de dados de genes

Como ilustrado na Figura 3, primeiro é feito um tratamento dos dados dos arquivos, substituindo valores de colunas "*NaN*" por *string* vazia, removendo caracteres desnecessários do ID dos genes e tratando os diversos tipos de dados, para que fiquem em um formato aceito pelo banco de dados. Então, para inserção na tabela "*Actions*", se utiliza o arquivo de interação pré-processado. Já para a tabela "Gene", os genes presentes no arquivo de interações são utilizados como base e, logo, são complementados com as informações obtidas dos demais arquivos pré-processados. Com isso, tem-se a base de genes, informações e interações que pesquisadores precisam, a fim de complementar os estudos feitos nas análises que realizam.

[11]https://datatracker.ietf.org/doc/html/rfc4627

4.3. *Script* de Pré-processamento de Dados de Análises

Quanto ao *script* de pré-processamento de dados de análises, esse é o momento em que os DEGs dos arquivos da análise (Análise *Up* e Análise *Down*) são inseridos no sistema.

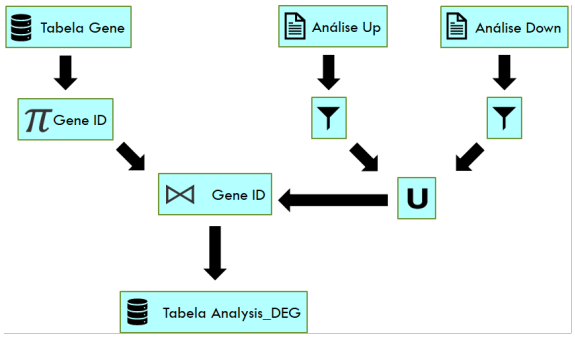

Figura 4. Esquema do pré-processamento dos arquivos de análises

Para isso, como ilustrado na Figura 4, é feito um tratamento dos arquivos de análise, semelhante ao explicado no script de pré-processamento de dados de genes. Após isso, é feita a união dos dados dos dois arquivos e, então, a uma junção interna com os IDs de genes da tabela de "*Gene*". O resultado disso são os DEGs da análise que pesquisadores podem utilizar para montar subgrafos por um painel do site ou por meio da funcionalidade de *clustering*.

4.4. *Script* de Busca de *Clusters* Representativos

O algoritmo de *clusters* representativos utiliza os DEGs de uma análise e o valor da distância máxima, entre quaisquer DEGs, ambos escolhidos pelo pesquisador na solicitação feita pelo *site*.

Com isso, o algoritmo funciona em duas etapas: na primeira, busca por *clusters* que abrangem a maior quantidade de DEGs para a distância máxima dada e gera árvores de *cluster*; e na segunda, elimina todos os nós que não fazem parte do caminho entre os DEGs, ou seja, as folhas da árvore que não são DEGs. Um exemplo dessas etapas é mostrado na Figura 5, em que a parte mais à esquerda é o grafo inicial e a mais à direita são os *clusters* formados pelo algoritmo.

Para a primeira parte do algoritmo, foi adaptado o algoritmo de busca em largura (BFS), para que a busca fosse limitada ao valor máximo de distância escolhido em relação a qualquer dos DEGs visitados durante o percurso. O motivo para o uso do algoritmo de BFS é a realização de um percurso por níveis, assim percorrendo primeiro os nós mais próximos ao nó inicial [Kleinberg and Tardos 2014]. O algoritmo realiza esse BFS adaptado sobre todos os DEGs do grafo, até que todos estejam marcados com nível zero (visitados). Enquanto é feita essa iteração, os demais genes são marcados com o nível do vértice atual somado a 1, caso esse valor seja inferior ao último valor de marcação desses genes. Logo, são obtidas as árvores de *cluster*, como na parte central da Figura 5.

Na segunda parte do algoritmo, é aplicada uma busca em profundidade (DFS) para remoção dos nós das árvores que são folhas, mas não DEGs. O uso do DFS é justificado, pois, caso todos os filhos de um nó fossem eliminados, esse nó também se tornaria uma folha, precisando ser eliminado também, o que não seria possível fazer com a mesma

facilidade, ao utilizar um algoritmo de BFS [Kleinberg and Tardos 2014]. Com o término dessa remoção, obtém-se os *clusters* representativos para a solicitação do pesquisador, os quais são persistidos no banco de dados para posterior visualização na aplicação *web*.

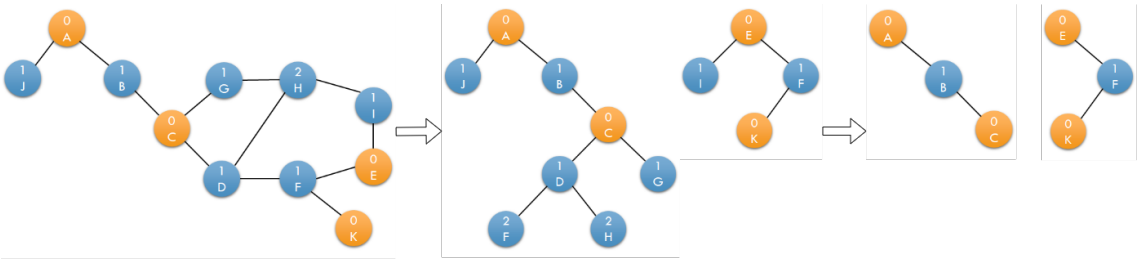

Figura 5. Etapas do algoritmo de busca de *clusters* representativos (DEGs em laranja e demais genes em azul) para uma distância máxima de 2 nós.

4.5. Aplicação *Web*

Como meio de realizar solicitações de pré-processamento de dados de genes e análises, solicitações de busca por *clusters* representativos e a visualização dos resultados dessas solicitações foi criada uma aplicação *web*. Logo, pode-se observar, por meio da Figura 6, um exemplo de *cluster* resultante da solicitação feita em uma análise. Na *navtab*, logo acima da tabela, é possível mudar a tabela que está sendo visualizada, permitindo adicionar ou remover DEGs ou outros genes do grafo; ver suas informações; e buscar quais genes interagem entre si. No painel à direita, é possível interagir com o grafo de genes, reorganizando os nós conforme necessário.

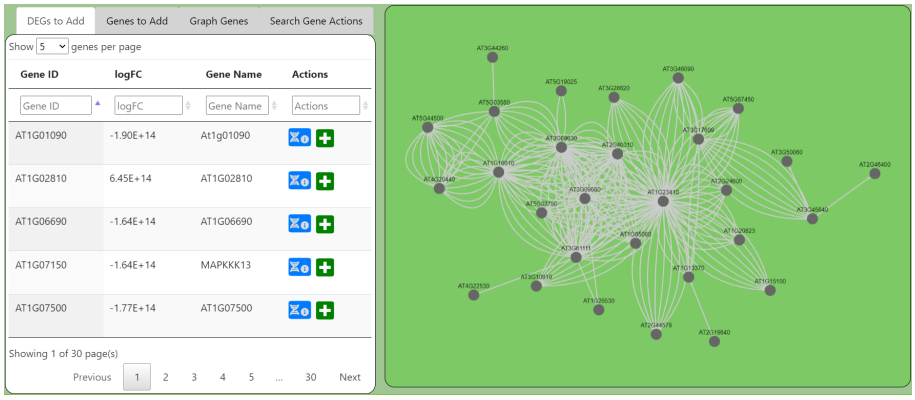

Figura 6. Uso da aplicação *web* para visualização de *clusters* representativos

5. Considerações Finais

Neste trabalho, por meio do uso de *scripts*, um banco relacional e uma aplicação *web*, foi possível montar e exibir subgrafos de genes, que possibilitem pesquisadores fazerem análises relativas a genes específicos, chamados de DEGs, para organismos infectados, e estudar o comportamento deles em relação à rede de interação de genes.

Ainda assim, a ferramenta tem muitas possibilidades de crescimento. Algumas dessas opções são: uma comparação entre o desempenho de um banco de dados relacional e outro em grafo para esse esquema; uma opção para comparar o resultado de análises distintas; a realização de sintonia fina para aumento no desempenho das buscas da aplicação; e o aprimoramento da interface de interação com o grafo de genes.

Referências

Anjum, A., Jaggi, S., Varghese, E., Lall, S., Bhowmik, A., and Rai, A. (2016). Identification of differentially expressed genes in rna-seq data of arabidopsis thaliana: A compound distribution approach. *Journal of Computational Biology*, 23(4):239–247.

Kleinberg, J. and Tardos, E. (2014). *Algorithm design*. Pearson Education Limited, 1st edition.

Pinto, M. d. S. T., Ribeiro, J. M., and de Oliveira, E. A. G. (2011). O estudo de genes e proteínas de defesa em plantas. *Revista Brasileira de Biociências*, 9(2).

Rodriguez-Esteban, R. and Jiang, X. (2017). Differential gene expression in disease: a comparison between high-throughput studies and the literature. *BMC medical genomics*, 10(1):1–10.