Marilene Andrade Garcia

Relatório Final PIBIC
Método MEMo

São Carlos

2019

Marilene Andrade Garcia

Relatório Final PIBIC
Método MEMo

Investigação de um método computacional baseado em redes gênicas para identificação de genes significativos para o cancêr.

Universidade de São Paulo - USP

Instituto de Ciências Matemáticas e de Computação - ICMC/USP

Orientador: Prof. Dr. Adenilso da Silva Simão

Coorientador: Jorge Francisco Cutigi

São Carlos

2019

Resumo

Abordagens computacionais para detectar mutações de genes que possam ser significativos para a progressão do câncer têm sido desenvolvidas nos últimos anos [3]. Por meio de parcerias entre profissionais das áreas da biologia e da computação diversos métodos foram criados e aprimorados. Esta iniciação científica teve como objetivo o estudo do método MEMo (Mutual Exclusivity Modules in Cancer) [1], a implementação do mesmo e a validação dele em dados reais.

Palavras-chaves: abordagens computacionais, câncer, método MEMo.

Sumário

1 Introdução

Essencialmente, o câncer é uma doença genética [2]. Um elevado número de células compõem um organismo adulto, as quais sofrem processos de divisão sujeitos a uma frequência de erros [2]. Tais erros originam mutações que podem ser significativas para o desenvolvimento do câncer (*"driver mutations"*) ou serem aleatórias e ocasionais, não afetando o comportamento celular (*"passenger mutations"*) [2].

Métodos computacionais estão sendo desenvolvidos para processar os dado biológicos gerados no sequenciamento do genoma de amostras de tecidos tumorais de forma a identificar eficientemente as *"driver mutations"* e os genes significativos para o desenvolvimento do comportamento cancerígeno celular. Entre eles o método MEMo (*Mutual Exclusivity Modules in Cancer*)[1].

O propósito central do MEMo é indicar grupos de genes relacionados (módulos) com alta frequência de mutações, que pertençam ao mesmo *"pathway"*, ou seja, que estão associados a um mesmo processo biológico, e exibam padrões de exclusividade mútua [1]. Mutações em tais genes possivelmente estão relacionadas com o câncer, logo esses módulos são designados como *"drivers networks"*. [1].

No período de desenvolvimento desta iniciação científica foi estudada a oncogênese e os artigos sobre os métodos computacionais que identificam as *"driver mutations"*, com foco no método MEMo. Sendo que cada etapa do algoritmo deste método foi avaliada e implementada de forma adequada, por meio do aprendizado da linguagem de programação utilizada e da teoria de grafos, e posteriormente houveram testes do código elaborado em dados reais.

2 Descrição do Algoritmo

O método MEMo é dividido em quatro etapas, sendo a primeira relacionada com a filtragem de dados provenientes do sequenciamento do genoma de amostras de tecidos tumorais e com a construção de uma matriz binária de genes alterados em diferentes pacientes, a segunda com a obtenção dos *"pathways"*, genes possivelmente relacionados à mesma função biológica, a terceira com a definição dos módulos que serão utilizados para ponderações estatísticas sobre a característica de exclusividade mútua analisada na quarta etapa [1]. A característica de exclusividade mútua consiste da hipótese de que se houver alteração em um único gene do *pathway* é esperado que a célula na qual ele está contido adquira comportamento cancerígeno [1], contudo, se dois ou mais genes sofrerem mutações, é possível que nada aconteça à célula, ou em contrapartida, que ela morra, mas ela não irá atuar na oncogênese [1].

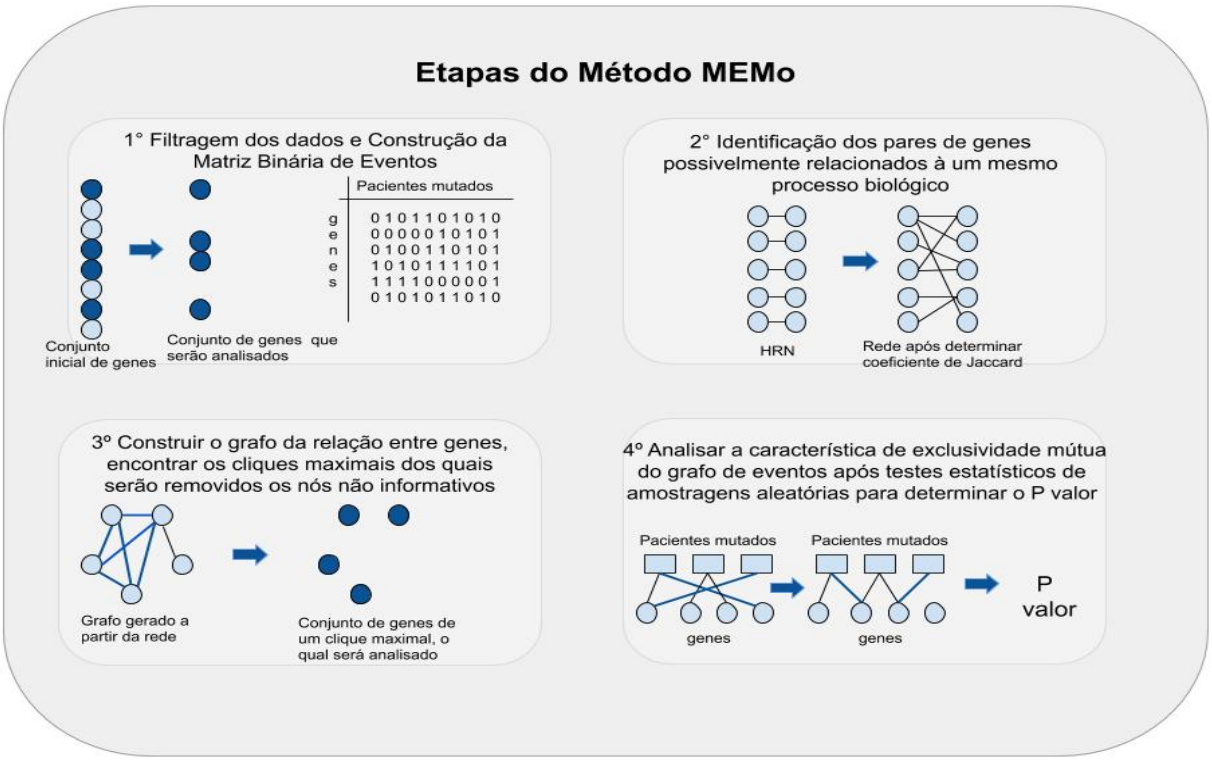

Figura 1 – Etapas do método MEMo

2.1 Primeira Etapa do Método

Nessa etapa é construída uma matriz binária de eventos na qual cada gene alterado em um paciente recebe o número um, sendo que essas alterações podem ser mutações somáticas do tipo SNVs (variação de nucleotídeo único, que ocorre quando um nucleotídeo é substituído por outro) ou CNAs (alterações do número de cópias, quando o DNA ganha ou perde um segmento

do genoma). O MEMo usa o teste padrão MutSig para identificar os genes que são mutados significativamente daqueles que são mutados por acaso, além de algoritmos como o GISTIC e o RAE para determinar os genes mutados em regiões que com maior frequência são amplificadas ou excluídas [1].

2.2 Segunda Etapa do Método

Na segunda etapa o MEMo utiliza um teste de comparação em todos os genes para determinar aqueles funcionalmente conectados baseado em uma rede de referência humana *(HRN)* já conhecida [1]. Este teste consiste em avaliar se dois genes compartilham um grande número de vizinhos comuns, mesmo se não estiverem diretamente conectados, de forma a ser determinado o coeficiente de Jaccard entre eles, o qual é calculado pela divisão da quantidade de genes no conjunto da intersecção dos vizinhos comuns pela quantidade no conjunto da união deles [1]. É fornecido um intervalo de valores que podem ser usados como o limite inferior do coeficiente quando ele for utilizado para determinar se os dois genes pertencem ao mesmo módulo funcional [1].

2.3 Terceira Etapa do Método

Nesta etapa o MEMo constrói um gráfico de todos os pares de genes relacionados que foram indicados pela etapa anterior [1]. Em seguida é extraído desse gráfico os *"cliques maximais"*, ou seja, todos subgrafos totalmente conectados, de forma que cada subgrafo não possa ser contido por outro sub gráfico totalmente conectado [1]. E, por fim, são filtrados os genes que possivelmente não são relacionados ao câncer, por meio da remoção de nós não informativos, aqueles cujo número de pacientes nos quais um gene é alterado ao mesmo tempo de outros é maior que a quantidade na qual apenas aquele gene é multado, considerando um mesmo clique [1].

2.4 Quarta Etapa do Método

A quarta e última etapa consiste em analisar a característica de exclusividade mútua dos genes de cada clique identificado na etapa anterior. A matriz binária da primeira etapa será utilizada também, pois ela será considerada como de um grafo bipartido, com um conjunto de nós representando os genes e outro os pacientes, cada aresta ligando um gene a um paciente significa uma alteração genômica [1]. A fim de garantir que a característica de exclusividade mútua não foi obtida por acaso são realizadas ponderações estatísticas [1]. Ao final de todas as etapas, tem-se elencados os prováveis "drivers networks".

3 Desenvolvimento do Projeto

Para este projeto foi decidido que o algoritmo do MEMo seria implementado utilizando a linguagem de programação *"Python (versão 3.6.7)"*.

3.1 Primeira Etapa do Método

Foi determinado que apenas seriam analisadas mutações somáticas do tipo SNVs e era previsto que não seria usado o filtro *MutSig* pois ele reduz a quantidade de genes a serem analisados. Entretanto na etapa três do código é necessário encontrar os *"cliques maximais"* do grafo formado pela interação dos genes, para determinar os *"Pathways"*, e quando o conjunto total de genes foi usado percebeu-se que o algoritmo não finalizava a execução, então foi verificado que se tratava de um problema NP-completo com uma grande quantidade de dados, logo, em uma segunda análise, foi estabelecido que seria usado o *MutSig* nos dados para alguns dos testes, para outros um filtro que retém apenas genes mutados em mais de 2% dos pacientes, ou mesmo a combinação de ambos e, posteriormente, os resultados obtidos de cada um foram comparados.

O código elaborado recebe como entrada um arquivo que contém os genes mutados e os pacientes nos quais ocorreram as mutações, sendo que ele deve necessariamente conter um cabeçalho com as identificações do nome do gene *"Hugo_Symbol"*, do número associado a ele *"Entrez_Gene_Id"* e da amostra tumoral do paciente *"Tumor_Sample_Barcode"*. E outro arquivo que contém uma rede da associação de pares de genes, de forma que eles devem estar indicados pelos respectivos números e não deve haver cabeçalhos antecedendo os dados. Foram usados alguns dados reais dos cânceres glioblastoma multiforme (GBM) e de ovário para os testes, os quais são provenientes do *"The Cancer Genome Atlas"*, entretanto foram obtidos no software do próprio método, disponível para download em: http://cbio.mskcc.org/memo. Também deve ser fornecido um arquivo que contém os genes listados por um filtro de genes, para diminuir significativamente a quantidade de dados, no qual deve haver um cabeçalho *"gene"* na coluna que identifica os genes.

A partir do arquivo de genes mutados em determinados pacientes é criado outro arquivo com a matriz binária de eventos cujo nome é *"matriz_eventos.txt"*. Para manipular os dados foi utilizada a biblioteca *"Pandas (versão 0.24.2)"* e gerados *"DataFrames"*.

3.2 Segunda Etapa do Método

A rede de associação de genes recebida como entrada do algoritmo é utilizada como *HRN*. E para a realização dos testes deste projeto foram aplicados os valores 0.02, 0.025 e 0.03 para analisar os resultados em função dos limites do coeficiente de Jaccard indicado pelo método, além do valor 0.05 que não deveria ser apropriado. Após as comparações de proximidade dos genes por meio da analise dos vizinhos deles, a nova rede gerada é salva no arquivo *"rede_filtrada.txt"*.

Vale ressaltar que há uma filtragem de dados para garantir que tanto a matriz de eventos da primeira etapa quanto a rede gerada na segunda possuem os mesmos genes.

3.3 Terceira Etapa do Método

Por meio da biblioteca *"igraph (versão 0.7.1)"* o arquivo *"rede_filtrada.txt"* é convertido em um grafo não direcionado, do qual são extraídos os *"cliques maximais"* que serão filtrados com a remoção dos nós não informativos. É necessário evidenciar que encontrar os *"cliques maximais"* é um problema NP-completo, logo deve-se ponderar sobre o tamanho do grafo criado para que seja possível executar essa etapa em tempo polinomial.

3.4 Quarta Etapa do Método

Novamente a biblioteca *"igraph (versão 0.7.1)"* é utilizada para criar um grafo, desta vez o bipartido a partir do arquivo *"matriz_eventos.txt"* e também durante as ponderações estatísticas quando aleatoriamente algumas arestas são excluídas ou adicionadas ao grafo.

É possível notar que um clique é como classificado como um *"driver network"* com base em um P valor proveniente da observação da quantidade de pacientes em que cada gene dele é mutado mais de uma vez, após serem aplicadas modificações aleatórias nas arestas grafo bipartido dos eventos, em relação a essa mesma quantia em dados reais, e caso não haja relevância estatística quanto a essa característica de exclusividade mútua analisada, ou seja um p valor igual ou maior que 0.05, são verificados os sub-cliques dele de tamanho superior à dois nós. Para o cálculo desse p valor é considerado sempre que o clique adquire uma significância estatística igual ou superior aos dados reais.

A implementação desse projeto visou gerar o arquivo *"resultados.txt"* como saída do programa, o qual contém os p valor encontrado para cada módulo analisado, até o momento em que ele se tornou significativo ou que possuísse apenas dois genes.

4 Resultados

O primeiro teste do algoritmo foi direcionado aos dados do câncer glioblastoma multiforme (GBM), sendo que foram analisados apenas os genes mutados em mais de 2% dos pacientes. Os *"drivers networks"*, ou seja, os módulos cujo p valor foi inferior a 0.05, para os diferentes valores de coeficiente de Jaccard analisados, seguem abaixo:

- Para 0.02: (PSMD13 EGFR), (NF1 BRCA2), (PTEN PIK3CA PIK3R1 EGFR), (PDGFRA PIK3CA PIK3R1 EGFR PTEN), (PDGFRA PIK3CA PIK3R1 EGFR PTEN ERBB2) e (PIK3CA PIK3CG PIK3R1 EGFR ERBB2).
- Para 0.025: (PSMD13, EGFR), (EGFR TP53), (PTEN EGFR), (PIK3CA PIK3R1 EGFR PTEN), (PDGFRA PIK3CA PIK3CG PIK3R1 EGFR), (PTEN PIK3CA ERBB2 PIK3R1 EGFR) e (PIK3CA PIK3CG ERBB2 PIK3R1 EGFR).
- Para 0.03: (TP53 PRKDC), (PTEN EGFR), (PIK3CA PIK3R1 EGFR PTEN), (PDGFRA PIK3CA PIK3CG PIK3R1 EGFR), (PTEN PIK3CA ERBB2 PIK3R1 EGFR) e (PDGFRA PIK3CA PIK3CG ERBB2 PIK3R1 EGFR).
- Para 0.05: (PSMD13 EGFR), (COL1A1 FN1), (PTEN EGFR PDGFRA PIK3CA PIK3R1), (PIK3CA PIK3R1) e (PIK3R1 PDGFRA PIK3CA PIK3CG ERBB2 EGFR)

Em seguida o mesmo teste foi aplicado, mas apenas para os genes selecionados pelo *MutSig*, e os resultados seguem abaixo:;

- Para 0.02: (PTEN EGFR PIK3R1 PIK3CA ERBB2).
- Para 0.025: (PTEN EGFR PIK3R1 PIK3CA ERBB2).
- Para 0.03: (PTEN EGFR PIK3R1 PIK3CA PDGFRA ERBB2).
- Para 0.05: (PTEN EGFR PIK3R1 PIK3CA)

Por fim, este teste foi aplicado ao conjunto da união dos genes analisados nos testes anteriores, e os *"drivers networks"* encontrados foram:

- Para 0.02: (BRCA2 NF1), (CHEK1 PRKDC), (PIM1 RB1), (TNFRSF10A PRKCZ TNFRSF1B), (TP53 EGFR), (PTEN PIK3CA PIK3R1 EGFR), (PTEN PDGFRA PIK3CA PIK3R1 EGFR), (PTEN PDGFRA PIK3CA PIK3R1 EGFR ERBB2), (PDGFRA PIK3CA PIK3CG PIK3R1 EGFR) e (EGFR PDGFRA PIK3CA PIK3CG PIK3R1 ERBB2).

- Para 0.025: (BRCA2 NF1), (CHEK1 PRKDC), (PIM1 RB1), (TNFRSF10A PRKCZ TNFRSF1B), (TP53 PRKDC), (PIK3CA PIK3R1 EGFR PTEN), (PTEN PIK3CA PIK3R1 ERBB2 EGFR), (ERBB2 PDGFRA PIK3CA PIK3CG PIK3R1 TEK EGFR) e (PTEN EGFR).
- Para 0.03: (PTEN PIK3CA PIK3R1 EGFR), (PTEN PIK3CA PIK3R1 EGFR ERBB2) e (EGFR PDGFRA PIK3CA PIK3CG PIK3R1 ERBB2).
- Para 0.05: (PTEN EGFR PIK3CA PIK3R1) e (EGFR PIK3CA PDGFRA PIK3CG PIK3R1 ERBB2).

Para uma melhor análise alguns resultados foram plotados em dois gráficos, o primeiro representa o número de *"drivers networks"* de acordo com o filtro aplicado e o segundo representa o tempo de execução de cada conjunto de dados.

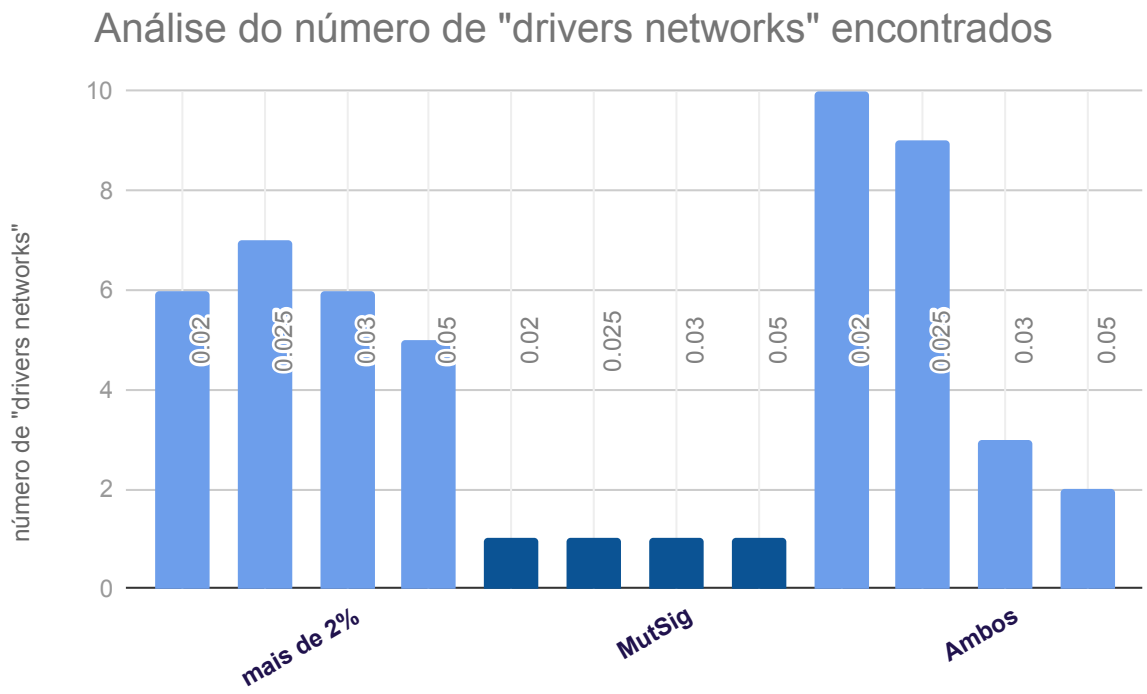

Figura 2 – Gráfico do número de "drivers networks"encontrados

Vale ressaltar que o códigos que usam apenas os dados dos genes mutados em mais de %2 dos pacientes e os que usam ambos os filtros foram executados em um *"Intel Core i5-7300HQ @ 3.50GHz, quad-core, 8192MB RAM, LinuxMint 18.3 operating system"*, já os códigos que usam os dados provenientes do *'MutSig"* foram executado em um *"Processor: Intel Core i7-5500U @ 2.30GHz, quad-core, 8192MB RAM, Ubuntu 18.04 operating system"*

Pode-se reparar que quando o código foi executado nos genes indicados pelo "MutSig"houve uma menor quantidade de *'Drivers Networks"* encontrados e um maior tempo de

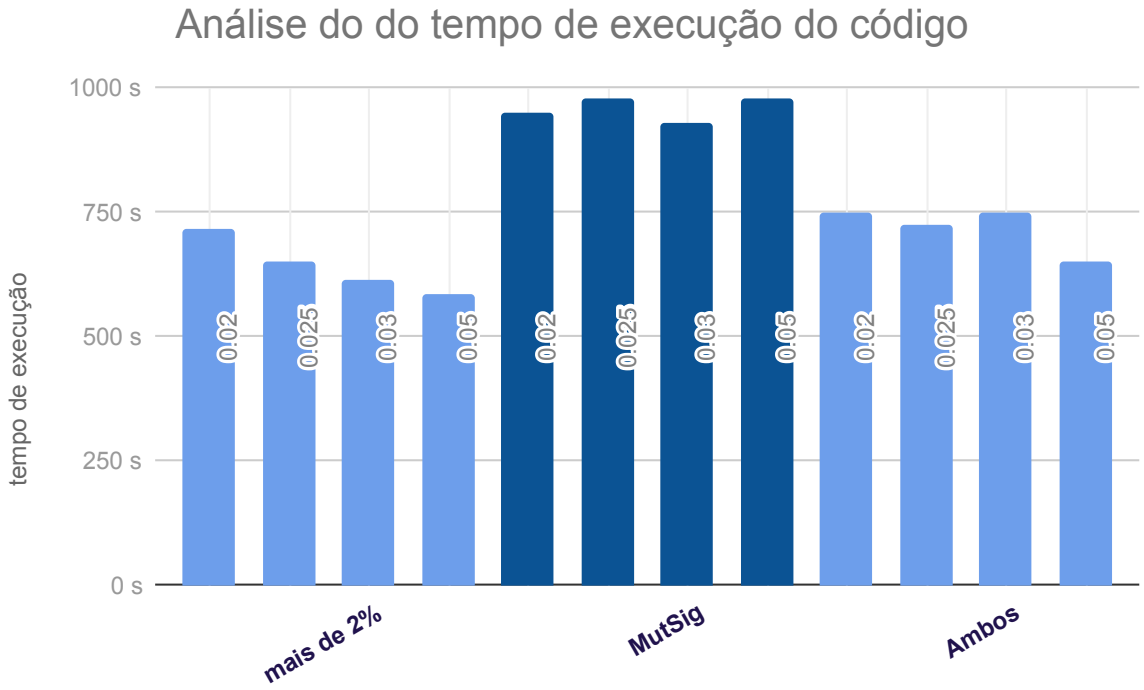

Figura 3 – Análise do tempo de execução do código para cada dado

execução, possivelmente pelo fato o algoritmo deve ter sido aplicado a muitos sub-cliques, de forma recursiva nos laços de repetição, para tentar encontrar um p valor adequado.

Os testes seguintes foram aplicados aos dados do câncer de ovário, sendo que os genes testados foram apenas os mutados em mais de 2% dos pacientes e os *"drivers networks"* encontrados para os diferentes valores de coeficiente de Jaccard analisados, seguem abaixo:

- Para 0.02: (BRCA2 BRCA1).
- Para 0.025: (TTN MYH4) e (BRCA2 BRCA1).
- Para 0.03: (LAMA3 DST), (HSPG2 APOB) e (BRCA2 BRCA1).
- Para 0.05: ().

Foram plotados dois gráficos com os mesmos parâmetros dos testes anteriores.

Por meio do primeiro gráfico pode-se analisar que o valor 0.05 para o coeficiente de Jaccard realmente foi inadequado, visto que nenhum "drivers networks"foi localizado.

Desta vez todos os códigos foram executados em um *"Processor: Intel Core i7-5500U @ 2.30GHz, quad-core, 8192MB RAM, Ubuntu 18.04 operating system"*

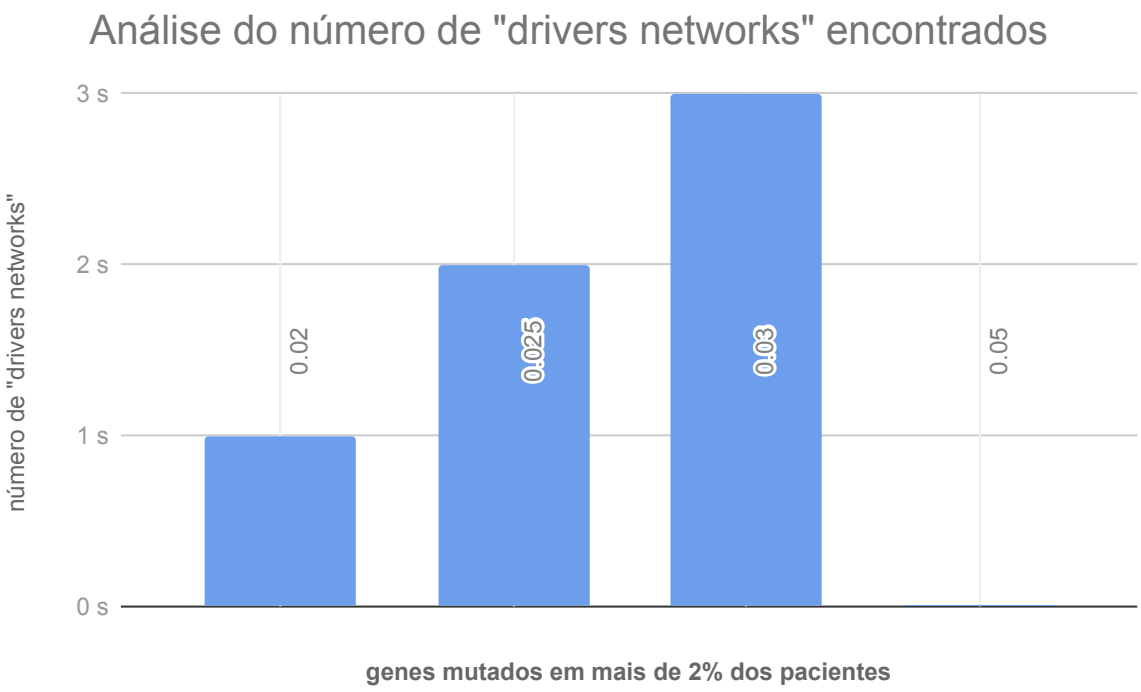

Figura 4 – Gráfico do número de "drivers networks"encontrados

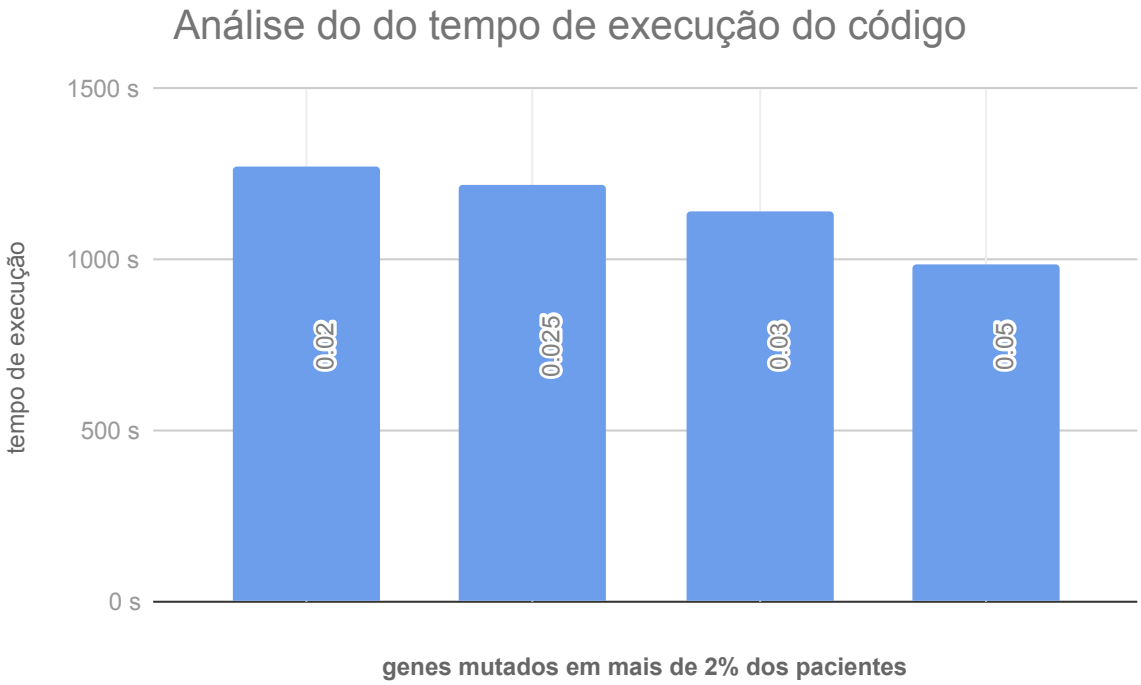

Figura 5 – Análise do tempo de execução do código para cada dado

5 Conclusões

Neste projeto foi proposto a replicação do método MEMo (Mutual Exclusivity Modulos in Cancer). Foram realizados testes com os dados reais do câncer GBM (glioblastoma multiforme) e do câncer de ovário que demonstraram o funcionamento do código desenvolvido, e a partir da análise dele é possível sugerir questões importantes que podem motivar projetos futuros:

- Criar threads para melhorar o desempenho dos laços de repetição.
- Incluir o uso de mutações CNAs (alterações do número de cópias) nos dados analisados.
- Verificar a eficiência do código em outros tipos de tumor.
- Utilizar algoritmos evolutivos para setar os parâmetros da forma mais eficiente para os dados testados, tais como os limites de valores do coeficiente de Jaccard e as variáveis *"Q" e "N"* utilizadas para definir a quantidade de iterações dos laços de repetição da etapa quatro, os quais são definidas por valores empíricos.
- Criar verificações dos arquivos de entrada do algoritmo, para garantir que todos sigam os modelos necessários.
- Analisar melhor o problema NP-completo de encontrar os *"cliques maximais"* e inferir possíveis soluções para quando forem usados grandes volumes de dados iniciais.

Bibliografia

[1] Giovanni Ciriello et al. “Mutual exclusivity analysis identifies oncogenic network modules”. Em: *Genome research* 22.2 (2012), pp. 398–406.

[2] Roderick R McInnes, Huntington F Willard e Robert Nussbaum. *Thompson & Thompson Genética Médica*. Elsevier Brasil, 2016.

[3] Benjamin J Raphael et al. “Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine”. Em: *Genome medicine* 6.1 (2014), p. 5.