Aplicação de simulated annealing para descobrir mutações drivers

Paulo Henrique Ribeiro[1], **Jorge Francisco Cutigi**[2], **Adriane Feijó Evangelista**[3], **Adenilso da Silva Simão**[4]

[1] Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP Barretos
Barretos, SP – Brasil

[2]Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP São Carlos
São Carlos, SP – Brasil

[3]Centro de Pesquisa em Oncologia Molecular – Hospital de Câncer de Barretos
Barretos, SP – Brasil

[4]Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo
São Carlos, SP – Brasil

phribeiro@ifsp.edu.br, cutigi@ifsp.edu.br,
adriane.feijo@gmail.com, adenilso@icmc.usp.br

Abstract. *Cells with driver mutations can proliferate faster than other cells, generating more daughter cells and causing the appearance of the tumor. In this context, there are computational methods for identifying possible driver mutations in a sample of cells. DriverNet is a method for classifying mutations in a sample, indicating those most likely to be driver mutations. This work presents a new proposal to identify driver mutations based on the DriverNet method. The new proposal, which requires a smaller set of input data than the DriverNet method, identified genes related to cancer from datasets from the TCGA project.*

Resumo. *As células com mutações drivers podem proliferar mais rapidamente do que outras células, gerando mais células-filhas e ocasionando o surgimento do tumor. Nesse contexto, existem métodos computacionais para identificar possíveis mutações drivers em uma amostra de células. O DriverNet é um método para classificar as mutações de uma amostra, indicando as que possuem maior probabilidade de serem mutações drivers. Este trabalho apresenta uma nova proposta para identificar mutações drivers baseado no método DriverNet. A nova proposta, que necessita de um conjunto menor de dados de entrada que o método DriverNet, identificou genes relacionados ao câncer de conjuntos de dados do projeto TCGA.*

1. Introdução

O câncer é uma doença que evolui por meio da aquisição de mutações no genoma de suas células. As mutações, que são mudanças que ocorrem nos genes das células de um organismo, podem ter como causa fatores internos ao organismo (por exemplo, erros durante algum processo biológico, como a replicação do DNA) ou fatores externos que ocasionam danos ao DNA (por exemplo, fumaça de cigarro).

A análise de mutações de uma amostra é realizada a partir do sequenciamento genético de um conjunto de células. O sequenciamento genético é uma técnica que permite identificar, na ordem correta, a sequência de nucleotídeos de uma molécula de DNA ou RNA, visando conhecer a informação genética contida nesta estrutura [Moreira 2015].

Algumas mutações fornecem à célula em que ocorreram uma vantagem evolutiva sobre outras células e são conhecidas como mutações "*drivers*", enquanto outras mutações ("*passengers*") são consideradas como tendo um efeito neutro. Uma célula tumoral com vantagem seletiva é mais adequada ao seu microambiente local e pode, portanto, proliferar mais rapidamente do que outras células e gerar mais células-filhas, ocasionando assim o surgimento do tumor.

Nesse contexto, é importante analisar as mutações existentes em uma amostra para identificar as mutações *drivers*, porém a identificação de mutações *drivers* é uma tarefa complexa na área da genômica do câncer [Hou and Ma 2013, Raphael et al. 2014, Cheng et al. 2016], pois as células de um tumor possuem muitas mutações *passengers* e poucas mutações *drivers*, sendo que não é trivial realizar a distinção entre esses dois tipos mutações [Greenman et al. 2007, Haber and Settleman 2007].

O estudo sobre métodos computacionais para identificar mutações *drivers* no câncer aumentou significativamente nos últimos anos [Hou and Ma 2013, Raphael et al. 2014, Cheng et al. 2016, Dimitrakopoulos and Beerenwinkel 2016, Pham et al. 2021, Cutigi et al. 2020, Yang et al. 2021], onde vários métodos com o objetivo de detectar mutações *drivers* a partir do sequenciamento genético de um conjunto de células foram propostos e analisados.

O DriverNet [Bashashati et al. 2012] é um método computacional para encontrar possíveis mutações *drivers*, classificando os genes de uma amostra de acordo com a probabilidade de serem mutações *drivers*. Essa classificação é realizada a partir dos três conjuntos de dados de entrada necessários para utilizar o método: matriz de mutação, matriz de expressão gênica e rede de influência. Inicialmente o DriverNet cria um grafo bipartido utilizando os dados de entrada e, após isso, um algoritmo guloso classifica os genes com base no grafo bipartido, indicando aqueles com maior probabilidade de serem mutações *drivers*.

O objetivo deste trabalho é apresentar uma proposta para descobrir possíveis mutações *drivers* utilizando a meta-heurística *simulated annealing*. A nova abordagem, chamada DDSA (Driver Discovery Simulated Annealing), é baseada no método DriverNet. A principal diferença do DDSA em relação ao DriverNet refere-se à utilização da técnica *simulated annealing* para classificar os possíveis genes *drivers*. O método DriverNet utiliza um algoritmo guloso para realizar essa classificação, buscando os nós com maior grau na partição esquerda do grafo bipartido criado pelo método. Essa estratégia pode mascarar genes *drivers* que possuem poucas ligações com os nós da partição direita do grafo. Desse modo, esses genes não são bem classificados na lista de genes gerada pelo DriverNet. Nesse contexto, a utilização do algoritmo *simulated annealing* pode resultar em um melhor conjunto de possíveis genes *drivers*.

A técnica *simulated annealing* foi utilizada por [Li et al. 2014] para propor o método SAGA (Simulated Annealing hybrid Genetic Algorithm). Esse método, que é baseado na abordagem apresentada por [Vandin et al. 2012], tem como objetivo identi-

ficar *pathways* de mutações *drivers*. Apesar dos métodos DDSA e SAGA utilizarem a meta-heurística *simulated annealing*, os métodos possuem objetivos diferentes: enquanto o primeiro produz uma classificação de genes em ordem de significância para o câncer (genes *drivers*), o segundo produz um conjunto de genes com mutações *drivers* e com exclusividade mútua (*pathways drivers*).

Outra diferença do DDSA em relação ao DriverNet refere-se aos conjuntos de dados de entrada: em vez de três conjuntos de dados de entrada, a nova proposta necessita apenas da matriz de mutação e da rede de influência, ou seja, não há necessidade de apresentar a matriz de expressão gênica, como acontece no método DriverNet, o que possibilita a utilização da nova proposta em situações em que não há os dados de expressão gênica. Ao comparar o método DDSA com o método DriverNet, foi possível observar que a utilização de apenas dois *datasets* não prejudicou a detecção de possíveis genes *drivers* dos *datasets* utilizados. Além disso, a nova implementação também foi executada com três *datasets* reais.

Para analisar o DDSA, inicialmente foi realizada uma comparação com o método DriverNet utilizando um mesmo conjunto de dados. No experimento realizado comparou-se a precisão de cada método de acordo com o *benchmark* fornecido pelo Network of Cancer Genes (NCG). O DDSA, mesmo sem utilizar a matriz de expressão gênica, apresentou uma precisão melhor que o método DriverNet. No segundo experimento realizado, o método DDSA foi aplicado com conjuntos de dados obtidos do projeto TCGA (The Cancer Genome Atlas). Ao analisar os resultados fornecidos pelo DDSA no experimento, foi possível identificar a existência de genes classificados como *drivers* pela comunidade científica e a existência de outros genes que, por enquanto, não são reconhecidos como *drivers*, sendo que a maioria dos genes presentes nos resultados possuem citações encontradas no CancerMine [Lever et al. 2019].

Este trabalho está dividido da seguinte maneira. Na Seção 2 alguns materiais e métodos utilizados neste artigo são apresentados. Na Seção 3 são apresentados os detalhes do método DDSA. Os resultados obtidos são demonstrados na Seção 4. Por fim, as considerações finais deste artigo são descritas na Seção 5.

2. Materiais e Métodos

2.1. DriverNet

O método DriverNet [Bashashati et al. 2012] utiliza uma abordagem de grafo bipartido para classificar as mutações de uma amostra, indicando as que possuem maior probabilidade de serem mutações *drivers*. Essa classificação é realizada a partir da combinação de dados sobre mutação, níveis de expressão gênica e redes gênicas. Para estudar as propriedades e vantagens do DriverNet, os autores utilizaram diferentes conjuntos de dados para diferentes tipos de câncer: câncer de mama, câncer de ovário e glioblastoma.

O método possui três conjuntos de dados de entrada:

1. Matriz de mutação, onde as linhas são pacientes e as colunas são os genes: indica se determinado gene sofreu mutação em determinado paciente, sendo que 0 significa que o gene não sofreu mutação e 1 significa que o gene sofreu mutação;
2. Matriz de expressão gênica, onde as linhas são pacientes e as colunas são os genes: indica quais genes estão altamente expressos em cada paciente, sendo que o

valor “TRUE” significa que o gene está altamente expresso no paciente e o valor “FALSE” indica que o gene é pouco expresso no paciente;

3. Rede de influência, onde as linhas e colunas são genes: indica interação entre os genes, sendo que 1 indica que existe interação entre dois genes e 0 significa que não existe interação entre dois genes.

A partir da execução do método, um grafo bipartido é construído. Os nós da partição esquerda correspondem ao conjunto de genes existentes nos conjuntos de dados de entrada. Na partição direita temos o mesmo conjunto de genes da partição esquerda, porém para cada paciente. Um esquema demonstrando o funcionamento do DriverNet é apresentado na Figura 1.

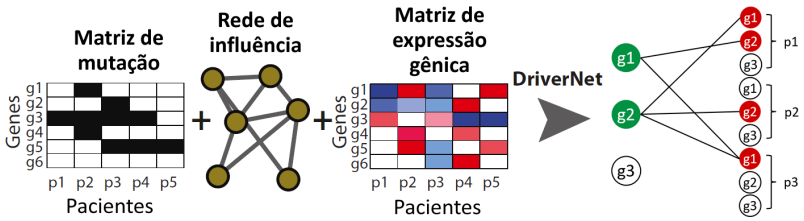

Figura 1. Representação esquemática do método DriverNet. A partir dos dados de entrada, o grafo bipartido é construído. Fonte: Adaptado de [Bashashati et al. 2012].

No grafo bipartido construído pelo método DriverNet, uma aresta é definida entre os nós nas partições esquerda e direita do grafo se as seguintes condições forem satisfeitas:

- O gene g_i está mutado no paciente p da população (nós verdes na partição esquerda do grafo bipartido);
- O gene g_j possui uma alta expressão gênica no paciente p (nós vermelhos na partição direita do grafo bipartido);
- Os genes g_i e g_j possuem uma ligação direta na rede de influência.

Após a construção do grafo bipartido, o DriverNet possui um algoritmo guloso para classificar os genes que possuem uma maior probabilidade de serem genes *drivers*. Esse algoritmo realiza uma busca na partição esquerda do grafo bipartido pelos nós com maior grau, ou seja, pelos genes que estão mais ligados aos genes da partição direita. Quando o nó de maior grau da partição esquerda é encontrado, ele é inserido na lista de possíveis genes *drivers* e os seus vizinhos da partição direita do grafo bipartido são removidos. O resultado do algoritmo é uma lista de possíveis genes *drivers*, sendo que o primeiro gene dessa lista é o que possui uma maior probabilidade de ser um gene *driver*.

2.2. Simulated Annealing

Simulated annealing (SA) [Kirkpatrick et al. 1983, Černý 1985] é uma meta-heurística que pode ser aplicada para diferentes problemas de otimização, possibilitando encontrar ótimos globais mesmo na presença de ótimos locais. O temo “*annealing*” possui como referência a termodinâmica, especificamente a maneira como os metais resfriam e se reaquecem.

O *simulated annealing*, que é baseado em técnicas de busca local probabilística, utiliza uma função objetivo para resolver um problema de otimização. Ao implementar o *simulated annealing*, em cada iteração do algoritmo procura-se aleatoriamente a

próxima solução do problema proposto que otimiza a função objetivo do algoritmo, porém é possível aceitar com uma determinada probabilidade soluções que pioram o valor da função objetivo para evitar ótimos locais.

O algoritmo *simulated annealing* pode ser descrito da seguinte forma. Partindo de uma solução inicial, uma sequência de iterações é executada, onde em cada iteração é gerada uma solução vizinha da solução atual. Se essa solução vizinha tiver um valor da função objetivo melhor, a solução atual é substituída por essa solução vizinha, caso contrário, verifica-se com uma certa probabilidade a substituição da solução atual pela vizinha com um valor da função objetivo pior.

A probabilidade de aceitar uma solução com um valor da função objetivo pior é definida como "função de aceite" e é normalmente representada por $e^{\frac{-\Delta E}{T}}$, onde ΔE é a diferença entre os valores da função objetivo da solução atual e da solução vizinha e T é o parâmetro "temperatura". Esta função implica que pequenas diferenças da função objetivo são mais aceitas que grandes mudanças. Além disso, quando o parâmetro "temperatura" é alto, soluções que pioram o valor da função objetivo são mais aceitas. Quando esse parâmetro se aproxima de zero, soluções com valores piores da função objetivo são mais rejeitadas.

Nesse contexto, o algoritmo *simulated annealing* é iniciado com uma temperatura alta para permitir que mais mudanças que piorem o valor da função objetivo sejam aceitas para, assim, evitar que o algoritmo fique preso um uma solução que representa um ótimo local. A cada iteração do algoritmo, o valor do parâmetro "temperatura atual" diminui utilizando o parâmetro conhecido como "fator de resfriamento".

Dessa maneira, os parâmetros a serem definidos para um algoritmo *simulated annealing* é: temperatura inicial, número de iterações e fator de resfriamento. Além disso, é necessário definir uma solução inicial, que será utilizada para encontrar uma solução vizinha na primeira iteração do algoritmo.

2.3. Genes conhecidamente relacionados ao câncer

A avaliação de métodos computacionais que identificam possíveis genes *drivers* é uma tarefa desafiadora. A inexistência de um padrão conhecido para indicar quais mutações podem ser classificadas como mutações *drivers* dificulta a comparação e a análise dos métodos dessa área, sendo que em muitos casos é necessário realizar a análise dos genes em laboratórios biológicos. Porém essa técnica, além de apresentar um alto custo, requer um tempo considerável de análise.

Nesse cenário, existem *benchmarks* com a relação dos genes já classificados como *drivers* pela comunidade científica. Um *benchmark* amplamente utilizado é o Network of Cancer Genes (NCG), que recentemente disponibilizou a versão 7.0 com a relação de 591 genes *drivers* conhecidos atualmente. O repositório do NCG pode ser acessado a partir do endereço: `http://ncg.kcl.ac.uk/`.

Com o auxílio de repositórios como do NCG é possível verificar se um método computacional, como é o caso do DriverNet, consegue detectar os genes já reconhecidamente como *drivers*, ou seja, genes que fazem parte de *benchmarks* como o NCG.

3. Método Driver Discovery Simulated Annealing

Para analisar melhor as suas características, o método DriverNet [Bashashati et al. 2012] foi implementado utilizando a linguagem de programação Python. O código fonte do método implementado está disponível em: `https://github.com/paulo-ribeiro/DriverNet/`. A verificação e validação da implementação foram realizadas a partir da comparação dos resultados fornecidos pelo método implementado na linguagem Python com os resultados indicados pelo pacote do DriverNet disponibilizado pelos autores do método para a linguagem R, disponível em: `https://github.com/shahcompbio/drivernet`.

Uma das características do método DriverNet é a necessidade de apresentar três conjuntos de dados para a execução do método: matriz de mutação, matriz de expressão gênica e rede de influência entre os genes. Considerando que, em algumas situações, é difícil obter dados da matriz de mutação e da matriz de expressão gênica do mesmo conjunto de pacientes, a necessidade de apresentar esses dados para utilizar o método DriverNet pode impossibilitar o seu uso em uma circunstância em que não há a matriz de mutação e a matriz de expressão gênica dos pacientes.

Outra característica do DriverNet refere-se ao algoritmo guloso para definir a lista dos genes que possuem uma maior probabilidade de serem genes *drivers*. Esse algoritmo, que utiliza a partição esquerda do grafo bipartido, realiza uma busca pelos nós com maior grau, que representa os genes que possuem mais ligações com os genes da partição direita. O resultado é uma lista de genes classificados de acordo com a sua cobertura da partição direita do grafo bipartido. Essa abordagem pode mascarar genes *drivers* existentes na partição esquerda do grafo bipartido que não possuem muitas ligações com os nós da partição direita. Consequentemente, esses genes não são bem classificados na lista de genes gerada pelo DriverNet.

Nesse contexto, baseado no método DriverNet [Bashashati et al. 2012], propomos o método Driver Discovery Simulated Annealing (DDSA) para a classificação de possíveis genes *drivers* de um conjunto de dados. A primeira modificação em relação ao método DriverNet é referente à matriz de expressão gênica, que não é necessária para a utilização do método DDSA. Dessa maneira, os dados de entrada necessários para a execução do método DDSA são a matriz de mutação e a rede de influência entre os genes, permitindo assim utilizar o método DDSA em situações que não há informações sobre a matriz de expressão gênica dos pacientes. Para implementar essa modificação, as condições para a criação de arestas no grafo bipartido foram modificadas, retirando a verificação relacionada à expressão gênica. O Algoritmo 1 representa o processo de construção do grafo bipartido do método DDSA.

A segunda modificação do método DDSA refere-se ao algoritmo para determinar a classificação dos possíveis genes *drivers*. Para gerar essa classificação, foi utilizada a meta-heurística *simulated annealing*, que possibilita encontrar ótimos globais mesmo na presença de ótimos locais, permitindo resultar um conjunto melhor de possíveis genes *drivers*. Os passos do novo algoritmo podem ser descritos da seguinte maneira:

- Seleciona-se aleatoriamente N genes da partição esquerda do grafo bipartido, inserindo no conjunto C;
- O algoritmo *simulated annealing* realiza I iterações, sendo que em cada iteração

verifica-se a possibilidade de realizar a troca do gene $G1$, selecionado aleatoriamente do conjunto C, pelo gene $G2$, selecionado aleatoriamente da partição esquerda do grafo bipartido;
- Após o término das iterações do algoritmo *simulated annealing*, ordena-se os genes do conjunto C de acordo com o grau de cada nó da partição esquerda.

O objetivo do algoritmo *simulated annealing* é encontrar um conjunto de N genes da partição esquerda do grafo bipartido que possui a maior cobertura dos genes da partição direita do grafo, ou seja, que possui uma maior quantidade de nós vizinhos. Para analisar a possibilidade da troca de genes em cada iteração do algoritmo *simulated annealing* é calculada a cobertura dos nós da partição direita. Considerando que os N genes existentes no conjunto C representam N nós da partição esquerda do grafo bipartido, verifica-se a quantidade de nós da partição direita do grafo bipartido ligados aos genes dos N nós da partição esquerda que fazem parte do conjunto C. Se a troca aumenta a cobertura da partição direita do grafo, a troca é efetivada. Caso contrário, através da técnica do *simulated annealing*, verifica-se com uma determinada probabilidade a realização da mudança, mesmo que a mudança diminua a cobertura da partição direita. O Algoritmo 2 apresenta a técnica *simulated annealing* implementada no método DDSA.

Os parâmetros utilizados no algoritmo *simulated annealing* do método DDSA foram: temperatura inicial $T_0 = 10^3$, fator de resfriamento de 10^{-2}, quantidade de iterações igual a 10^5. Os parâmetros foram definidos empiricamente a partir dos testes realizados.

Algoritmo 1: algoritmo do DDSA para a construção do grafo bipartido.
Fonte: Elaborado pelos autores.

```
Data: Matriz de mutação M; rede de interação entre os genes I
Result: Um grafo bipartido BP
G ← conjunto de genes de M;
P ← conjunto de pacientes de M;
Partição esquerda de BP ← G;
forall paciente p_k ∈ P do
    Partição direita de BP de p_k ← G;
end
forall par (g_i, p) do gene g_i ∈ partição esquerda de BP e p ∈ partição direita de BP do
    if g_i possui mutação em p then
        forall gene g_j do paciente p ∈ partição direita de BP do
            if g_i e g_j interagir em I then
                Coloque uma aresta entre g_i e g_j do paciente p;
            end
        end
    end
end
return BP;
```

4. Resultados

Para estudar as propriedades e vantagens do método Driver Discovery Simulated Annealing (DDSA), dois experimentos foram realizados. Primeiramente o método DDSA foi comparado com o método DriverNet utilizando um mesmo conjunto de dados. O segundo experimento consiste em executar o DDSA com outros conjuntos de dados, obtidos do

Algoritmo 2: algoritmo *simulated annealing* do DDSA para classificação de genes. Fonte: Elaborado pelos autores.

Data: Um grafo bipartido $BG(V, E)$
Result: Uma lista de classificação D dos genes *drivers*

$LP \leftarrow$ partição esquerda de BG ;
$T \leftarrow$ temperatura inicial ;
$bestSet \leftarrow$ conjunto de N genes selecionados aleatoriamente de LP ;
$bestSetCoverage \leftarrow$ quantidade de vizinhos dos genes $\in bestSet$
for $i \leftarrow 1$ *to* N **do**
    $g_i \leftarrow$ gene selecionado aleatoriamente de $bestSet$;
    $g_j \leftarrow$ gene selecionado aleatoriamente de LP ;
    $currentSet \leftarrow bestSet$;
    Remover g_i de $currentSet$;
    Inserir g_j em $currentSet$;
    $currentSetCoverage \leftarrow$ quantidade de vizinhos dos genes $\in currentSet$;
    if $currentSetCoverage > bestSetCoverage$ **then**
        $bestSet \leftarrow currentSet$
    else
        $diff \leftarrow currentSetCoverage - bestSetCoverage$;
        $n \leftarrow$ número aleatório de 0 a 1 ;
        if $n < exp(-diff/T)$ **then**
            $bestSet \leftarrow currentSet$;
        end
    end
    $T \leftarrow coolingFactor * T$;
end
$D \leftarrow$ genes $\in bestSet$ ordenados pelo grau do nó em LP ;
return D;

projeto TCGA (The Cancer Genome Atlas) [TCGA data portal 2018], para verificar se o método identifica *drivers* bem conhecidos e genes que, por enquanto, não são classificados como *drivers*.

4.1. Comparação com o DriverNet

Inicialmente o método DDSA foi analisado com o conjunto de dados disponibilizado com o método DriverNet [Bashashati et al. 2012], disponível em: `https://github.com/shahcompbio/drivernet`. Esse conjunto de dados possui 120 pacientes e 1255 genes e é constituído pela matriz de mutação, matriz de expressão gênica e rede de influência entre os entres. Para analisar a eficiência do método DDSA, o mesmo conjunto de dados também foi aplicado com o método DriverNet, sendo que o método DDSA necessita apenas da matriz da mutação e da rede de influência e o método DriverNet, além desses dois conjuntos de dados, necessita também da matriz de expressão gênica. Dessa maneira é possível comparar os resultados do DriverNet e do DDSA para o mesmo conjunto de dados.

Para analisar o resultado dos dois métodos, foi calculada a precisão de cada método. Dada uma lista de genes, a precisão é a fração desses genes que são classificados como genes *drivers* de acordo com algum *benchmark*. Conforme apresentado na Seção 2.3, neste trabalho foi utilizada a relação fornecida pelo Network of Cancer Genes (NCG), que possui 591 genes classificados como *drivers* pela comunidade científica.

A Figura 2 apresenta um gráfico com a precisão dos métodos DriverNet e DDSA considerando o conjunto de dados disponibilizado junto com o método DriverNet. Em ambos os métodos a precisão foi a mesma para os quinze primeiros genes. A partir do 16º gene o DDSA apresentou uma precisão um pouco melhor. Lembrando que, para a execução do DDSA, não foi necessária a utilização da matriz de expressão gênica, o que não prejudicou a precisão do método quando comparado com o DriverNet, que necessita desse conjunto de dados.

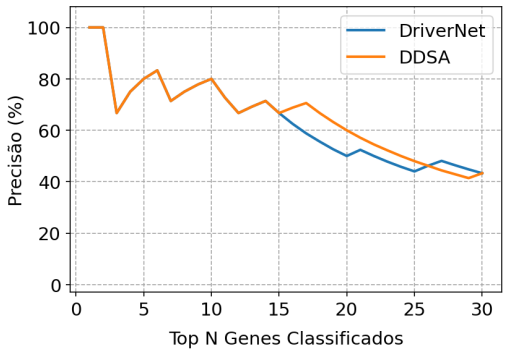

Figura 2. Precisão dos métodos DriverNet e DDSA considerando o conjunto de dados disponibilizado junto com o método DriverNet.

Outra comparação realizada entre os métodos DriverNet e DDSA refere-se à quantidade de citações indicada pelo CancerMine [Lever et al. 2019] para cada gene pertencente no resultado de cada método. A Figura 3 mostra a quantidade de citações encontrada no CancerMine para cada um dos 30 primeiros genes de ambos os métodos.

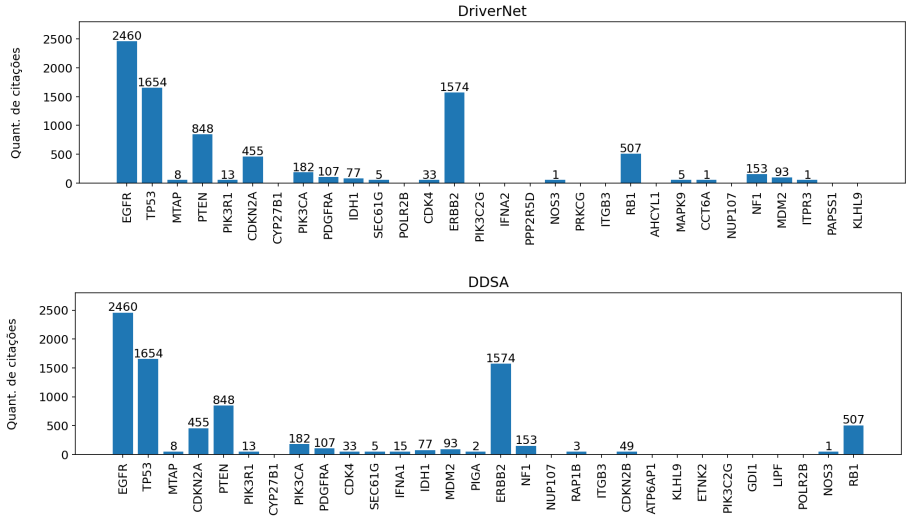

Figura 3. Quantidade de citações indicada pelo CancerMine.

Apesar de existir algumas diferenças entre os gráficos, aqui também é possível observar que a não utilização da matriz de expressão gênica no método DDSA não prejudicou o seu resultado quando comparado com o resultado do DriverNet.

4.2. Aplicações com conjuntos de dados do TCGA

Nesta subseção o método DDSA foi aplicado com conjuntos de dados de glioblastoma multiforme (GBM), ovariano seroso e adenocarcinoma da próstata. Os conjuntos de dados

utilizados foram obtidos do projeto TCGA (The Cancer Genome Atlas), que disponibiliza dados de mutação de vários tipos de câncer. A rede de interação de genes utilizada foi obtida da base de dados Reactome [Croft et al. 2014], que disponibiliza dados de interações funcionais gênicas. Os conjuntos de dados utilizados nesse experimento, selecionados com base nos *datasets* utilizados por [Bashashati et al. 2012], estão descritos na Tabela 1.

Tabela 1. Descrição dos datasets.

Dataset	Tipo do câncer	Nº Pacientes	Nº Genes
GBM	Glioblastoma multiforme	395	16.454
OV	Ovariano seroso	409	15.366
PRAD	Adenocarcinoma da próstata	493	12.825

Para cada tipo de câncer indicado na Tabela 1, foram criados dois gráficos. O primeiro tipo de gráfico refere-se à precisão do método DDSA quando utilizados os *datasets* do TCGA. Nesse gráfico é possível visualizar a fração dos N primeiros genes fornecidos pelo DDSA que são classificados como genes *drivers*. A Figura 4 apresenta a precisão do DDSA quando utilizados os *datasets* GBM, OV e PRAD do TCGA.

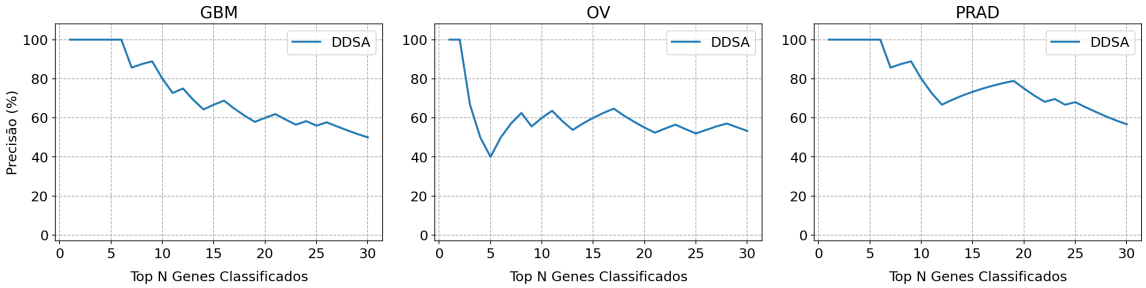

Figura 4. Precisão do DDSA.

O segundo tipo de gráfico apresenta a quantidade de citações encontradas no CancerMine para os 30 primeiros genes dos resultados fornecidos pelo DDSA para cada tipo de câncer utilizado nesse experimento. A Figura 5 apresenta essa quantidade de citações.

5. Conclusão

Nesse artigo foi apresentado o método DDSA, uma proposta para descobrir mutações *drivers* baseado no método DriverNet [Bashashati et al. 2012]. Uma mudança no algoritmo de construção do grafo bipartido foi proposta para que o novo método necessite como dados de entrada apenas a matriz de mutação e a rede de influência entre os genes, dispensando a necessidade de fornecer ao método DDSA a matriz de expressão gênica. Para classificar os possíveis genes *drivers*, ao invés de utilizar um algoritmo guloso, como acontece com o método DriverNet, foi utilizado um algoritmo *simulated annealing*.

Ao comparar o método DDSA com o método DriverNet utilizando o mesmo conjunto de dados, é possível verificar que o resultado do DDSA é próximo do resultado do DriverNet, indicando que dispensar o uso da matriz de expressão gênica não prejudicou na classificação dos genes do DDSA. Por fim, o método DDSA foi aplicado com

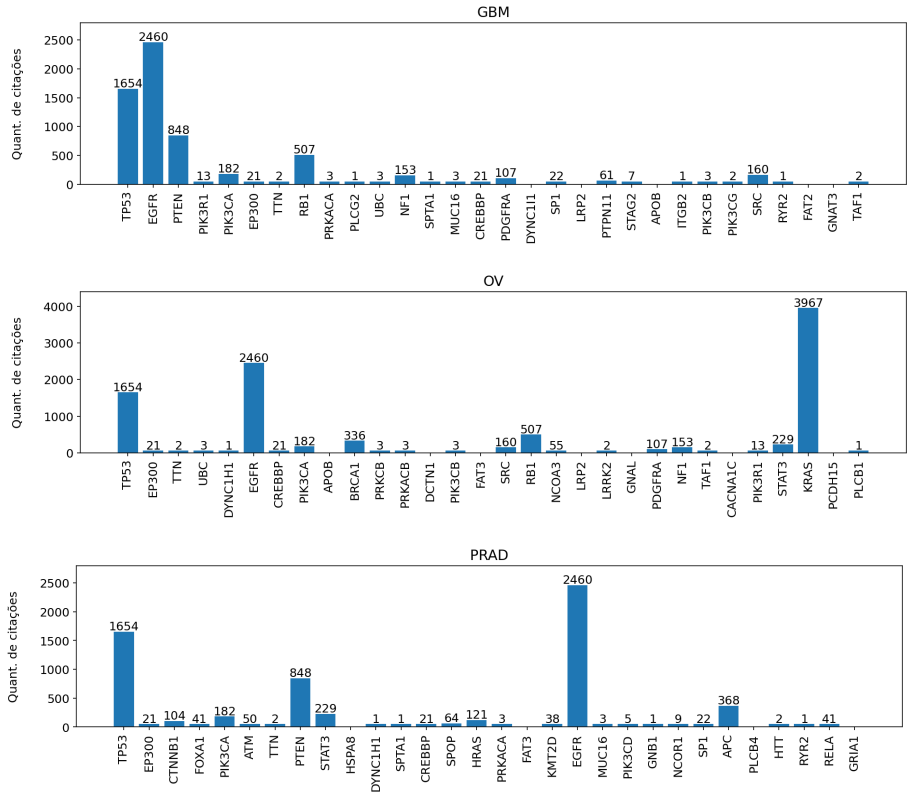

Figura 5. Quantidade de citações indicada pelo CancerMine.

conjuntos de dados obtidos do projeto TCGA (The Cancer Genome Atlas). Nesse experimento, nos 30 primeiros genes classificados pelo DDSA para os diferentes conjuntos de dados é possível observar a existência de genes classificados como *drivers* pela comunidade científica e a existência de outros genes que, por enquanto, não são reconhecidos como *drivers*, sendo que a maioria dos genes presentes nos resultados possuem citações encontradas no CancerMine. Aplicar o método DDSA com mais tipos de câncer e comparar o DDSA com outros métodos de identificação de mutações *drivers* são continuações naturais deste trabalho.

Referências

[Bashashati et al. 2012] Bashashati, A., Haffari, G., Ding, J., Ha, G., Lui, K., Rosner, J., Huntsman, D., Caldas, C., Aparicio, S., and Shah, S. (2012). Drivernet: Uncovering the impact of somatic driver mutations on transcriptional networks in cancer. *Genome Biology*, 13(12):1 – 14.

[Cheng et al. 2016] Cheng, F., Zhao, J., and Zhao, Z. (2016). Advances in computational approaches for prioritizing driver mutations and significantly mutated genes in cancer genomes. *Briefings in bioinformatics*, 17 4:642–56.

[Croft et al. 2014] Croft, D., Mundo, A. F., Haw, R., Milacic, M., Weiser, J., Wu, G., Caudy, M., Garapati, P., Gillespie, M., Kamdar, M. R., Jassal, B., Jupe, S., Matthews, L., May, B., Palatnik, S., Rothfels, K., Shamovsky, V., Song, H., Williams, M., Birney, E., Hermjakob, H., Stein, L., and D'Eustachio, P. (2014). The reactome pathway knowledgebase. *Nucleic acids research*, 42(D1):D472–D477.

[Cutigi et al. 2020] Cutigi, J. F., Evangelista, A. F., and Simão, A. d. S. (2020). Approaches for the identification of driver mutations in cancer: a tutorial from a computational perspective. *Journal of Bioinformatics and Computational Biology*, 18(3):2050016:1–2050016:32.

[Dimitrakopoulos and Beerenwinkel 2016] Dimitrakopoulos, C. and Beerenwinkel, N. (2016). Computational approaches for the identification of cancer genes and pathways. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 9.

[Greenman et al. 2007] Greenman, C., Stephens, P., Smith, R., Dalgliesh, G., Hunter, C., Bignell, G., Davies, H., Teague, J., Butler, A., Stevens, C., Edkins, S., O'Meara, S., Vastrik, I., Schmidt, E., Avis, T., Barthorpe, S., Bhamra, G., Buck, G., Choudhury, B., and Stratton, M. (2007). Patterns of somatic mutation in human cancer genomes. *Nature*, 446:153–8.

[Haber and Settleman 2007] Haber, D. and Settleman, J. (2007). Cancer: Drivers and passengers. *Nature*, 446:145–6.

[Hou and Ma 2013] Hou, J. and Ma, J. (2013). *Identifying Driver Mutations in Cancer*, volume 4, pages 33–56. Genome Medicine.

[Kirkpatrick et al. 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

[Lever et al. 2019] Lever, J., Zhao, E. Y., Grewal, J., Jones, M. R., and Jones, S. J. M. (2019). Cancermine: a literature-mined resource for drivers, oncogenes and tumor suppressors in cancer. *Nature Methods*, 16(6):505–507.

[Li et al. 2014] Li, H. T., Zhang, Y. L., Zheng, C. H., and Wang, H. Q. (2014). Simulated annealing based algorithm for identifying mutated driver pathways in cancer. *BioMed Research International*, 2014.

[Moreira 2015] Moreira, L. M. (2015). *Ciências genômicas: fundamentos e aplicações*. Sociedade Brasileira de Genética.

[Pham et al. 2021] Pham, V., Liu, L., Bracken, C., Goodall, G., Li, J., and le, T. (2021). Computational methods for cancer driver discovery: A survey. *Theranostics*, 11:5553–5568.

[Raphael et al. 2014] Raphael, B. J., Dobson, J., Oesper, L., and Vandin, F. (2014). Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine. *Genome Medicine*, 6:5 – 5.

[TCGA data portal 2018] TCGA data portal (2018). *The Cancer Genome Atlas Program*.

[Vandin et al. 2012] Vandin, F., Upfal, E., and Raphael, B. J. (2012). De novo discovery of mutated driver pathways in cancer. *Genome research*, 22(2):375–385.

[Yang et al. 2021] Yang, L., Chen, R., Goodison, S., and Sun, Y. (2021). An efficient and effective method to identify significantly perturbed subnetworks in cancer. *Nature Computational Science*, 1:79–88.

[Černý 1985] Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51.