UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Edwin Jahir Rueda Rojas

Identificação de candidatos a genes de referência utilizando Redes Geradoras Adversárias

Belém

2020

Edwin Jahir Rueda Rojas

Identificação de candidatos a genes de referência utilizando Redes Geradoras Adversárias

Qualificação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Universidade Federal do Pará

Orientador: Prof. Dr. Jefferson Magalhães de Morais
Coorientador: Prof. Dr. Rommel Thiago Juca Ramos

Belém
2020

Edwin Jahir Rueda Rojas

Identificação de candidatos a genes de referência utilizando Redes Geradoras Adversárias

Qualificação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Conceito: _____

Belém, <dia> de <mes> de 2019.

BANCA EXAMINADORA

Prof. Dr. Jefferson Magalhães de Morais - Orientador
UFPA

Prof. Dr. Rommel Thiago Juca Ramos - Coorientador
UFPA

Prof. Dr. Ronnie Cley de Oliveira Alves
ITV

Prof. Dr. Luis Carlos Guimarães
UFPA

Resumo

Os genes de referência (GR) são genes constitutivos necessários para a manutenção das funções celulares básicas. Diferentes tecnologias de alto desempenho são usadas para identificar esses tipos de genes, incluindo o seqüenciamento de RNA (RNA-seq), o qual permite medir os níveis de expressão gênica em um tecido específico ou em uma célula isolada. Nesta proposta de qualificação de mestrado, é apresentada uma nova abordagem baseada em Redes Geradoras Adversárias (GAN) e em Máquinas de Vetores de Suporte (SVM) para a identificação *in-silico* de candidatos a genes de referência. O método proposto é dividido em duas etapas principais. Primeiro, a rede GAN é usada para aumentar um pequeno número de genes de referência encontrados no conjunto de dados públicos de RNA-seq da *Escherichia coli*. Segundo, um classificador de uma classe baseado em SVM e em detecção de novidades é avaliado usando alguns genes de referência reais e sintéticos gerados pela arquitetura GAN na primeira etapa. Os resultados parciais mostram que o aumento do conjunto de dados empregando a arquitetura GAN proposta melhora a pontuação do classificador em 16.67%, fazendo com que o método proposto tenha uma pontuação *recall* de 83.33% nos dados de teste. Assim, a principal contribuição da metodologia proposta foi encontrar 753 genes candidatos de um total de 4170, o que permite reduzir a quantidade de genes a serem testados em laboratório em até 80%.

Palavras-chave: Redes Adversas Generativas, RNA-seq, Aprendizagem profunda, genes de referência.

Abstract

Reference genes (RG) are constitutive genes required for the maintenance of basic cellular functions. Different high-throughput technologies are used to identify these types of genes, including RNA sequencing (RNA-seq), which allows measuring gene expression levels in a specific tissue or an isolated cell. In this paper, we present a new approach based on Generative Adversarial Network (GAN) and Support Vector Machine (SVM) to identify *in-silico* candidates for reference genes. The proposed method is divided into two main steps. First, the GAN is used to increase a small number of reference genes found in the public RNA-seq dataset of *Escherichia coli*. Second, a one-class SVM based on novelty detection is evaluated using some real reference genes and synthetic ones generated by the GAN architecture in the first step. The results show that increasing the data set using the proposed GAN architecture improves the classifier score by 16.67%, making the proposed method have a recall score of 83.33% on the test data. The main contribution of the proposed methodology was to find 753 candidate genes out of a total of 4170, which allows to reduce the amount of genes to be tested in the laboratory by up to 80%.

Keywords: Generative adversarial networks, one-class svm, RNA-seq, deep learning, reference genes.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AI	Artificial Intelligence
AM	Aprendizado de Maquina
cDNA	DNA complementar
D	Rede Discriminadora
DNA	Ácido desoxirribonucleico
G	Rede Generadora
GAN	*Generative Adversarial Networks* (Redes Geradoras Adversárias)
GR	Genes de Referência
KNN	*K-Nearest Neighbors* (k-vizinhos mais próximos)
PCA	*Principal Component Analysis* (análise de componentes principais)
RNA	Ácido Ribonucleico
RNA-seq	*RNA-sequencing* (Sequenciamento de RNA)
SGD	*Stochastic Gradient Descent* (Descida de gradiente estocástico)
SVM	*Support Vector Machine* (Máquina de vetores de suporte)

Sumário

1 Introdução

A bioinformática é uma área multidisciplinar que aplica conhecimentos da informática, matemática e estatística, na busca de soluções para armazenamento, recuperação e análise de grandes quantidades de informações biológicas (GAUTHIER et al., 2019). Entre as diversas aplicações, este trabalho tem como foco a identificação *in-silico* dos chamados genes de referência (GR).

Os GR são genes constituintes necessários para a manutenção da função celular básica e eles estão expressos em todas as células de um organismo em condições normais e anormais. Assim, estes genes são empregados em controles internos em pesquisas de análises de expressão gênica (KIM; KIM; KIM, 2020). Por exemplo, no *quantitative real-time* PCR (qRT-PCR) o qual fornece uma quantificação precisa e reproduzível de cópias genéticas, a seleção dos GR é muito importante para a normalização dos resultados, pois conseguem reduzir as diferenças devido à variabilidade biológica e técnica.

Na atualidade, diferentes métodos tem sido desenvolvidos para a identificação *in-silico* de GR. Em (FRANCO et al., 2019), foi proposta uma abordagem de aprendizado de máquina (AM) baseada em algoritmos de agrupamento (empregando a distância Euclidiana como métrica de similaridade) para poder descobrir candidatos no conjunto de dados da *Corynebacterium pseudotuberculosis*. Nesta abordagem, 14 *clusters* foram escolhidos com base nos critérios de *SD validity index* e *Dunn index* (LEGÁNY; JUHÁSZ; BABOS, 2006). Após a geração dos *clusters*, e com base em 9 GR obtidos da literatura, foram considerados apenas os *clusters* que possuíam um ou mais GR, conseguindo reduzir consideravelmente o conjunto de genes iniciais. Finalmente, baseados na distância Euclidiana e no coeficiente de variação dos genes, 134 candidatos foram escolhidos.

Por outro lado, em (BERGHOFF et al., 2017) é proposta uma metodologia baseada em algoritmos de otimização, mais especificamente adaptando o problema do fluxo de custo mínimo em uma rede para a seleção dos possíveis candidatos. A ideia é encontrar o caminho de menor custo em um grafo direcionado e acíclico n_s-dimensional, onde n_s representa o número total de amostras (genes) partindo do gene com a expressão mais baixa diferente de zero (fonte) ao gene mais altamente expresso (sumidouro), dadas várias restrições. No final, com base em 6 GR os quais devem estar presentes no percurso, são encontrados 27 possíveis candidatos para o conjunto de *Escherichia coli* MG1655.

A principal desvantagem dos métodos atuais é o baixo número de GR disponíveis na literatura, e mesmo tendo uma quantidade razoável de genes, a seleção destes varia de acordo com a célula ou tecido estudado (OLSVIK; SØFTELAND; LIE, 2008). Assim, os métodos que usam a distância Euclidiana entre os genes não classificados e os GR como

métrica de similaridade para detectar possíveis candidatos, não estão dando importância a genes não classificados os quais estão distantes dos GR, mas que podem ser candidatos devido à estabilidade em sua expressão gênica.

Nesta pesquisa é proposta uma metodologia baseada em Redes Geradoras Adversárias (GAN) e classificadores de uma classe (baseados em detecção de novidades) para a identificação de genes candidatos a GR. A metodologia proposta é dividida em duas etapas. Primeiro, através de uma arquitetura GAN é aumentada sinteticamente a classe dos GR para o conjunto de dados de *Escherichia coli* MG1655, obtidos de (BERGHOFF et al., 2017). Segundo, é treinado um classificador de uma classe usando uma porcentagem dos GR e os genes sintéticos gerados na primeira etapa, para subsequentemente classificar os genes restantes como candidatos ou não. Nesta abordagem, é empregado um classificador de uma classe baseado em Máquinas de Vetores de suporte (SVM) devido a que este apresenta um bom desempenho na literatura (YAHAYA; LANGENSIEPEN; LOTFI, 2018) (AMRAEE et al., 2018).

A principal vantagem da metodologia proposta surge devido ao uso de técnicas para a geração de dados sintéticos, o que permite aumentar sinteticamente a quantidade de genes de referência e assim criar um modelo mais generalizado para a seleção de possíveis novos candidatos a GR. Por sua vez, a não implementação da distância Euclidiana como métrica de seleção de genes candidatos, permite a seleção de genes com estabilidade em sua expressão genética, mas que não se aproximam dos genes de referência empregados para a seleção.

1.1 Justificativa

Atualmente, existem diversos métodos para a identificação *in-silico* de candidatos a GR (FRANCO et al., 2019) (BERGHOFF et al., 2017) (VANDESOMPELE et al., 2002), alguns deles empregando algoritmos de agrupamento (*clustering*), ou outros algoritmos baseados principalmente na distância Euclidiana. O principal objetivo do uso dessas propostas é diminuir o teste *in-vitro* dos genes, reduzindo assim o tempo e o recurso econômico empregado. Embora esses métodos sejam úteis, a seleção dos genes candidatos é baseada em um pequeno número de GR, fazendo com que a escolha de candidatos seja viciada por um pequeno grupo de genes.

O objetivo desta pesquisa é mitigar o problema que surge ao criar um modelo de seleção de candidatos com um pequeno número de GR, para isso, é proposto um classificador de uma classe (baseado em detecção de novidades) o qual é treinado com um conjunto de GR o qual é previamente aumentado sinteticamente mediante uma rede GAN. Com esta abordagem, se constrói um modelo de seleção de candidatos mais generalizado, pois é aumentada a diversidade e quantidade dos dados de treinamento.

1.2 Objetivo Geral

O Objetivo geral deste trabalho consiste em propor uma metodologia para identificação *in-silico* de genes de referência com base no aumento de dados de RNA-seq utilizando Redes Geradoras Adversárias.

1.3 Objetivos Específicos

- Identificar os genes de referência que serão empregados para o estudo.
- Avaliar arquiteturas de rede GAN para aumentar sinteticamente os genes de referência.
- Treinar um modelo de classificação de uma classe para a identificação de candidatos a genes de referência.

1.4 Estrutura do Trabalho

Este capítulo faz uma breve introdução aos tópicos abordados neste estudo, bem como ao conteúdo abordado em cada capítulo. A seguir, apresenta-se cada capítulo com uma breve descrição do conteúdo abordado.

Capítulo 2. Fundamentação Teórica. É apresentada uma introdução aos conceitos fundamentais para o desenvolvimento do estudo. São abordados tópicos sobre genes de referencia; RNA-seq; aprendizado de máquina y redes neuronales artificiales.

Capítulo 3. Trabalhos Relacionados. Neste capítulo, é apresentada uma breve descrição dos principais métodos utilizados para a seleção de candidatos a genes de referência, como limitações encontradas nesses métodos. Finalmente, é introduzido o método proposto para identificar candidatos a genes de referência.

Capítulo 4. Método Proposto. É apresentado em detalhe o método proposto para a seleção de genes candidatos a genes de referência. Por sua vez, é abordado o processamento inicial dos genes; o processo de construção da arquitetura GAN e o modelo de aprendizado de máquina usado para a seleção de genes candidatos.

Capítulo 5. Resultados Parciais. São apresentados e discutidos os resultados obtidos nesta pesquisa, como o ganho obtido na identificação de candidatos a genes de referencia com o método proposto.

Capítulo 6. Considerações Parciais. Neste capítulo é apresentada a conclusão parcial da pesquisa, as contribuições do projeto e as próximas atividades a serem desenvolvidas.

2 Fundamentação Teórica

Neste capítulo são apresentados os conceitos fundamentais relacionados ao desenvolvimento deste estudo, assim como, estudos já desenvolvidos na área.

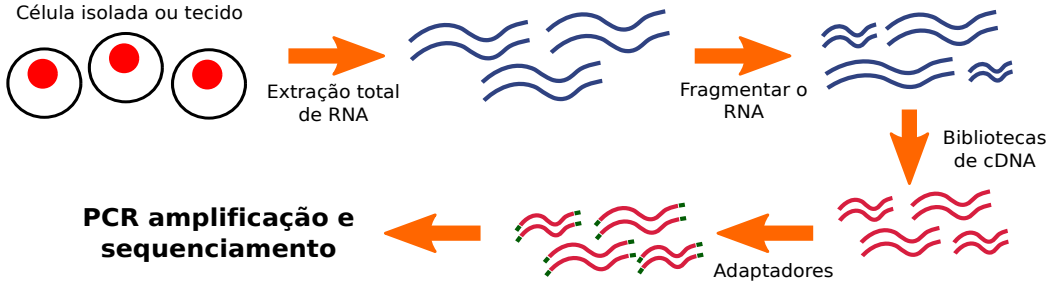
2.1 Genes de Referência

Os GR podem ser definidos como genes constituintes necessários para a manutenção da função celular básica (são genes essenciais para a existência de uma célula), estes genes estão expressos em todas as células de um organismo em condições normais ou de estresse, e o que os diferencia de outros tipos de genes é que sua expressão gênica varia pouco em diferentes condições de estresse. Por isso, estes tipos de genes são necessários nos controles internos de análise de expressão gênica (KIM; KIM; KIM, 2020), e sua escolha incorreta em técnicas que permitem a quantificação do RNA mensageiro (mRNA) como *real-time reverse transcription-polymerase chain reaction* (RT-qPCR), *Northern blotting* e *RNase protection assay* pode levar à interpretações errôneas dos dados de expressão gênica obtidos (LONG et al., 2020) (DHEDA et al., 2005).

2.2 RNA Sequencing (RNA-seq)

O RNA-seq é uma técnica de sequenciamento a qual é empregada para revelar a presença e a quantidade de RNA em uma amostra biológica em um determinado momento, o que permite observar mudanças na expressão gênica ao longo do tempo (CHU; COREY, 2012). A Figura 1 ilustra o funcionamento da técnica, onde a partir de uma amostra de RNA é obtida uma biblioteca de cDNA com adaptadores ligados nas duas extremidades, para posteriormente mapear o genoma de referência e quantificar os níveis de expressão dos genes em estudo em diversas condições.

Figura 1. Visão geral de um experimento de RNA-seq



Fonte: Adaptado de (KUKURBA; MONTGOMERY, 2015)

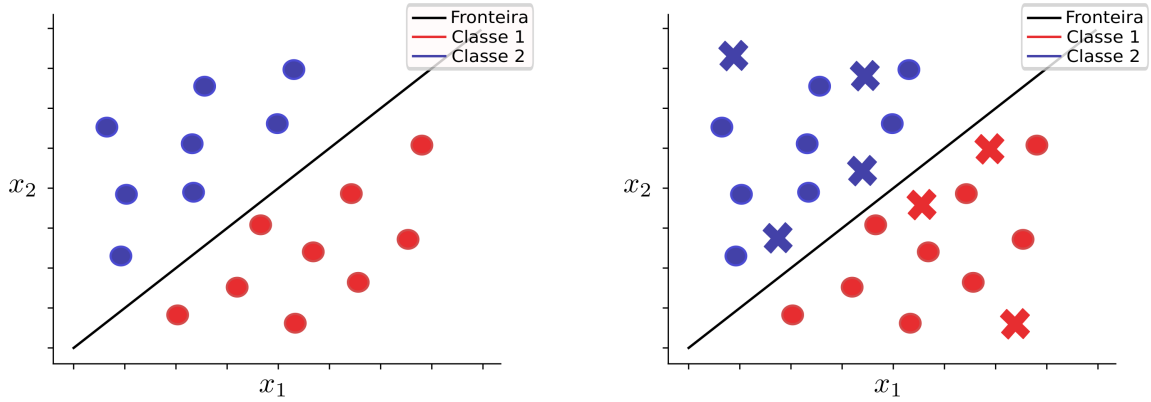
2.3 Aprendizado de Máquina

Na área da inteligência artificial, o Aprendizado de Máquina (AM) refere-se ao fato de construir algoritmos os quais melhoram com a experiência, ou seja, se diz que um computador aprende de uma experiência **E** com respeito à umas tarefas **T** com medida de desempenho **P**, se seu desempenho nas tarefas **T**, medido por **P**, melhora com a experiência **E** (JORDAN; MITCHELL, 2015). Neste ramo, os diferentes problemas são classificados de acordo com a natureza dos dados e a tarefa a ser desenvolvida. Existindo dois algoritmos principais: os algoritmos supervisionados os quais são empregados quando os dados são rotulados (dados distinguíveis), e os algoritmos não supervisionados (empregados nesta pesquisa) para quando os dados não são rotulados (dados indistinguíveis).

2.3.1 Aprendizado Supervisionado

O aprendizado supervisionado parte do fato de que um conjunto de dados é distinguível (tem-se um conhecimento a priori dos dados), o que permite diferenciar cada exemplo do conjunto com base em uma classe ou rótulo. Este tipo de aprendizado é comumente empregado em problemas de classificação, onde o objetivo é treinar um modelo para criar uma fronteira de decisão a qual mais tarde servirá para classificar novos exemplos como pertencentes a uma determinada classe/rótulo (CARUANA; NICULESCU-MIZIL, 2006). A Figura 2a ilustra a criação de uma fronteira de decisão (linha preta) com base em um conjunto de dados que possui duas características (x_1 e x_2) e duas classes (Classe 1 e Classe 2). Por sua vez, a Figura 2b ilustra o processo de classificação de novos exemplos (cruzes) não contemplados no conjunto de treinamento, os quais são classificados como pertencentes à Classe 1 se estiverem abaixo da fronteira de decisão, caso contrário, são classificados como pertencentes à Classe 2.

Figura 2. Aprendizado Supervisionado



(a) Fronteira de decisão

(b) Clasificação de novos exemplos

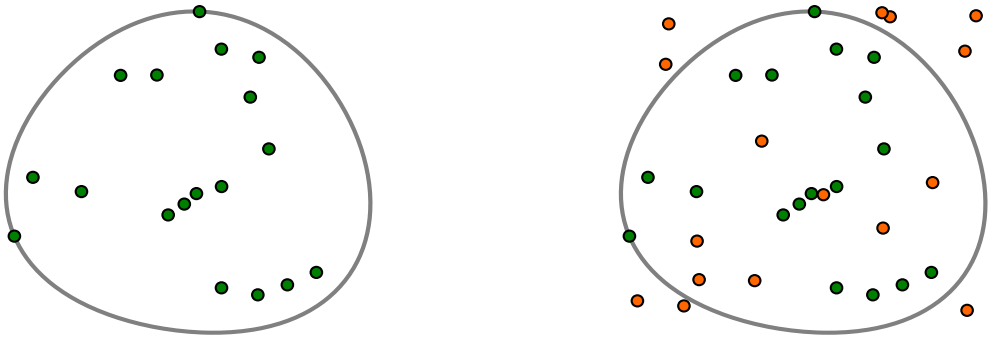
Fonte: Autor

2.3.2 Aprendizado não Supervisionado

O aprendizado não supervisionado é um tipo de AM no qual os dados empregados para treinar um determinado modelo não possuem rótulos (são dados indistinguíveis). Este tipo de aprendizado é comumente empregado em pesquisas onde o objetivo é encontrar agrupamentos (*clusters*) em conjuntos de dados nos quais não há conhecimento a priori (não há uma relação previa conhecida nos dados) (ZHUL et al., 2018). Atualmente, existem diversos algoritmos não supervisionados que são empregados dependendo do tipo de problema a ser resolvido. Algoritmos de agrupamento (*clustering*) são utilizados quando necessário encontrar relações nos dados; algoritmos de detecção de anomalias são empregados para encontrar *outliers* nos dados, e algoritmos de detecção de novidades são treinados para detectar se uma nova observação é um *outlier* (novidade) ou não. Este último algoritmo também é considerado na literatura como semi-supervisionado, uma vez que o conjunto de treinamento não possui *outliers*, o que o faz provir de uma única distribuição (classe/rótulo).

Nesta pesquisa, com base no fato de que apenas o rótulo dos GR está disponível, e com o objetivo de encontrar o rótulo dos outros genes (se são candidatos ou não), é implementado um algoritmo de detecção de novidades. Este tipo de algoritmo é treinado para criar uma fronteira de decisão com base em um conjunto de dados que segue uma certa distribuição p. O objetivo é classificar uma nova observação como pertencente (dentro da fronteira) ou não pertencente (fora da fronteira) à essa distribuição. A Figura 3a ilustra a criação da fronteira de decisão com base no conjunto de treinamento (círculos verdes) e a Figura 3b ilustra a classificação de novas obervações (círculos laranjas), destacando que as observações fora da fronteira (novidades) são consideradas não provenientes da distribuição p.

Figura 3. Detecção de novidades



(a) Fronteira de decisão

(b) Clasificação de novas observações

Fonte: Autor

2.3.3 Métricas de Avaliação

Existem inúmeras métricas para validar o desempenho de algoritmos de aprendizado semi-supervisionados (empregados nesta pesquisa), a maioria destas focadas no tipo de algoritmo e no tipo de problema abordado. Para validar o desempenho dos algoritmos de detecção de novidades, é comumente empregada a métrica *Recall* (também chamada de sensibilidade). Nesta métrica, representada na Equação 2.1, é calculada a proporção de sucesso de um modelo com base em sua classe positiva, em outras palavras, é calculada a proporção de classificar como positivos todos os dados pertencentes a essa classe, onde TP (*True positives*) representa a quantidade de dados que o algoritmo classificou corretamente como pertencentes à classe positiva, e FN (*False Negatives*) representa a quantidade de dados classificados como não pertencentes à classe positiva, sendo essa sua classe.

$$Recall = \frac{TP}{TP + FN} \tag{2.1}$$

Finalmente, para algoritmos de detecção de novidades, um *recall* próximo a 1.0 indica que o modelo está reconhecendo corretamente a maioria dos dados pertencentes à distribuição de treinamento. Observe que, na implementação destes algoritmos, seu treinamento é realizado com apenas uma classe (classe positiva); portanto, novos dados que não pertencem a essa classe são considerados novidades (classe negativa).

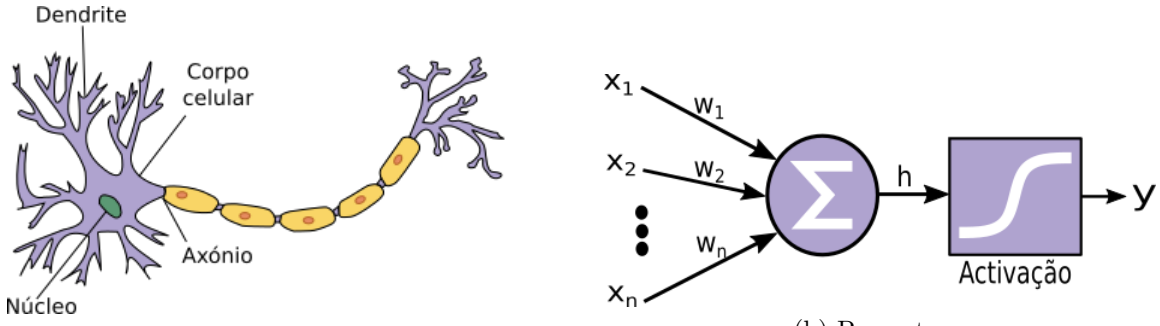
2.4 Redes Neurais Artificiais

As redes neurais artificiais, como seu nome indica, são redes computacionais que tentam simular aproximadamente os processos de decisão executados pelas redes de células nervosas (neurônios) do sistema nervoso central humano. A principal vantagem destas redes é o uso de operações matemáticas simples (adição, subtração, multiplicação e operações lógicas fundamentais) para resolver poblemas complexos, como problemas não-lineares ou problemas estocásticos (DANIEL, 2013).

O perceptron, introduzido por (ROSENBLATT, 1960), é uma unidade básica (neurônio artificial) a qual se assemelha ao funcionamento de um neurônio biológico, podendo aprender a discriminar e reconhecer diferentes padrões de percepção. A Figura 4 ilustra a equivalência entre um neurônio biológico (Figura 4a) e um perceptron (Figura 4b), onde os dendritos do neurônio biológico são equivalentes à entrada $x_1, \ldots, x_n$ do perceptron, e o axônio é equivalente à saida h (hipótese) do referido perceptron, a qual é dada pela soma da multiplicação de suas entradas $x_1, \ldots, x_n$ com seus pesos $w_1, \ldots, w_n$. O objetivo é encontrar os valores $w_1, \ldots, w_n$ que satisfazem com o menor erro possível um determinado problema. Deve-se destacar que o axônio artificial pode ser conectado a outros axônios mediante os dendritos para poder formar uma rede neural artificial capaz de resolver

problemas mais complexos.

Figura 4. Neurônio biológico vs Perceptron



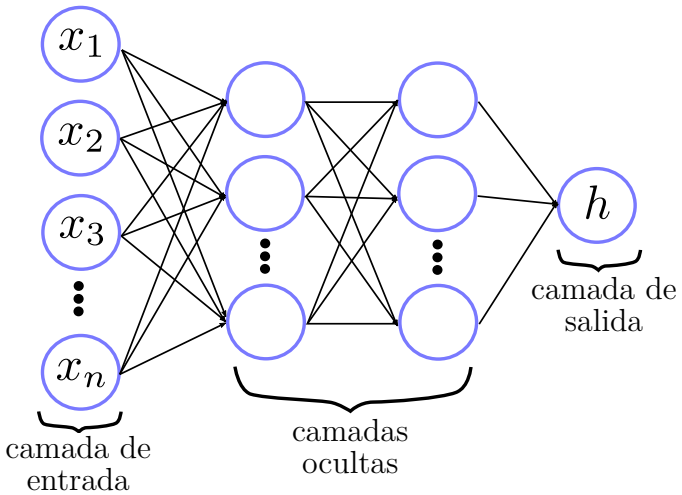
(a) Neurônio biológico

(b) Perceptron

Fonte: Autor

Atualmente, as redes neurais estão compostas por múltiplos neurônios e múltiplas camadas, o que facilita a resolução de problemas mais complexos. A Figura 5 ilustra uma arquitetura de rede neural com duas camadas ocultas, onde cada neurônio em cada camada se conecta a todos os outros neurônios na próxima camada, destacando que a camada que recebe os dados de entrada $x_1, \ldots, x_n$ é conhecida como camada de entrada (*input layer*), as camadas intermediárias são conhecidas como camadas ocultas (*hidden layers*) e a última camada é chamada de camada de saída (*output layer*). Este tipo de arquitetura de rede neural é conhecida como totalmente conectada.

Figura 5. Arquitetura de uma rede neural totalmente conectada

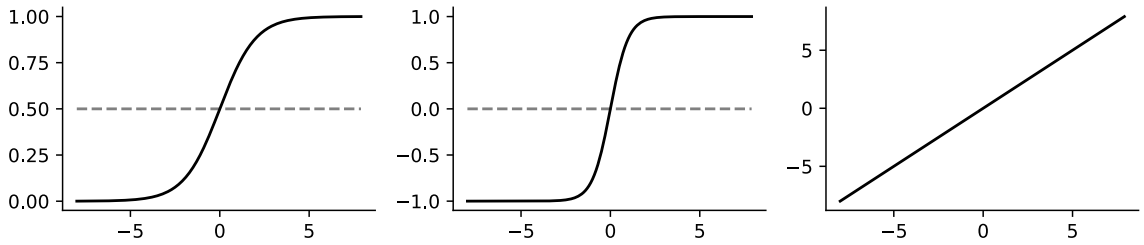


Fonte: Autor

Para facilitar a resolução de diferentes problemas empregando redes neurais artificiais e para facilitar a convergência destas arquiteturas, são empregadas funções de ativação as quais são computadas com o valor de sáida de cada neurônio artificial. Estas funções são escolhidas dependendo do tipo de arquitetura de rede empregada e do tipo de saída

esperada. A Figura 6 ilustra as funções de ativação mais comuns, onde a função sigmóide (Figura 6a) é mais empregada na camada de saída para problemas de classificação, a função tangente hiperbólica (Figura 6b) é mais comum em camadas ocultas e na camada de saída das redes geradoras (em arquiteturas GAN), e a função linear é mais empregada na camada de saída para problemas de regressão.

Figura 6. Principais funções de ativação



(a) Função sigmoide (b) Função tangente hiperbólica (c) Função linear

Fonte: Autor

O processo de aprendizado neste tipo de redes neurais ocorre através da atualização dos seus pesos $w_1, \ldots, w_n$, a fim de minimizar uma função de custo $J(w_1, \ldots, w_n)$ responsável por medir o quão bem a rede está aprendendo sobre um determinado problema. O *stochastic gradient descent* (SGD) é o método mais empregado para resolver este problema de minimização, no qual, os pesos são atualizados com base em seus gradientes de descida. A Equação 2.2 mostra a atualizaçõ destes pesos, levando em consideração uma taxa de aprendizado α, responsável por controlar a magnitude do salto com o qual os pesos da rede são atualizados para atingir um mínimo de $J(w_1, \ldots, w_n)$.

$$w_i = w_i - \alpha \frac{\partial J(w_1, \ldots, w_n)}{\partial w_i} \tag{2.2}$$

Atualmente, existem diferentes funções de custo as quais são implementadas dependendo do tipo de problema a ser resolvido. Por exemplo, em problemas de classificação binária (empregado nas redes GAN), é utilizada uma função de custo chamada entropia cruzada binária (ECB). A Ecuação 2.3 mostra o cálculo desta função de custo, onde y representa o valor alvo (0 ou 1 para problemas de classificação binária) e p_x representa a probabilidade de um exemplo x pertencer à classe positiva ($y = 1$) e não à classe negativa ($y = 0$).

$$ECB = -\left(y \log(p_x) + (1 - y) \log(1 - p_x)\right) \tag{2.3}$$

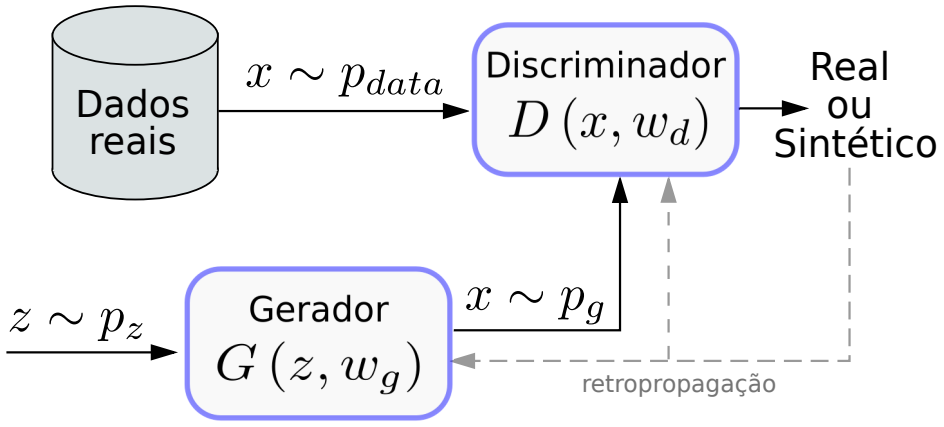
Finalmente, espera-se que, ao repetir um certo número de vezes (épocas) a atualização dos pesos em uma rede, seja atingido um mínimo local ou o mínimo global da função

de custo $J(w_1, \ldots, w_n)$, o que garantiria uma arquitetura de rede neural suficientemente robusta para um determinado problema.

2.5 Redes Geradoras Adversárias

As Redes Geradoras Adversárias foram introduzidas por (GOODFELLOW et al., 2014). O conceito foi desenvolvido para a estimativa de modelos generativos baseados em processos adversários. Nestes procesos, duas arquiteturas de redes neurais são treinadas; a primeira, uma rede geradora $G(z, w_g)$ que recebe como entrada algumas variáveis de ruído provenientes de uma distribuição $p_z(z)$ e tem como objetivo aprender uma distribuição p_g sobre os dados de treinamento x, e uma segunda rede discriminadora $D(x, w_d)$, que tem como saída um único escalar o qual representa a probabilidade de que x venha da distribuição dos dados reais p_{data} e não de p_g. A Figura 7 ilustra a arquitetura geral de uma rede GAN, sendo que esta representa um jogo *min-max* onde existe uma única solução a qual ocorre quando a rede G recupera a distribuição dos dados reais $p_g = p_{data}$.

Figura 7. Arquitetura geral de uma rede GAN



Fonte: Autor

O processo de aprendizado neste tipo de arquitetura GAN é semelhante ao processo de aprendizagem visto acima para redes neurais artificiais (empregando a entropia cruzada binária como função de custo), com a diferença de que neste processo os pesos w_g e w_d das redes G e D são atualizados com base em duas funções de custo diferentes. O Algoritmo 1 representa a atualização dos pesos através do uso do SGD. Espera-se que, se as arquiteturas das redes G e D forem suficientemente robustas e equilibradas, a rede G será capaz de gerar dados sintéticos similares aos reais, tornando a rede D incapaz de distinguir entre dados reais e sintéticos, $D(x) = \frac{1}{2}$.

Atualmente, este tipo de arquitetura de rede adversária é amplamente utilizada em problemas onde se deseja aumentar um pequeno conjunto de dados (geralmente imagens), a fim de treinar um modelo de AM com a maior diversidade possível de dados, aumentando

Algorithm 1 *Minibatch Stochastic Gradient Descent* para treinar uma rede generadora adversária. Os pesos das duas redes são atualizados com base em m exemplos (Minibatch).

for número de iterações **do**

- *Minibatch* de $m/2$ exemplos sintéticos $G\left(z, w_g\right) \rightarrow \left\{x'_1, x'_2, \ldots, x'_{m/2}\right\} \in p_g$.
- *Minibatch* de $m/2$ exemplos reais $\left\{x_1, x_2, \ldots, x_{m/2}\right\} \in p_{data}\left(x\right)$.

$\overline{x} = \left\{x_1, x_2, \ldots, x_{m/2}, x'_1, x'_2, \ldots, x'_{m/2}\right\}$

$\overline{y} = \left\{1^{m/2}, 0^{m/2}\right\}$ ▷ $1^{m/2}$ representa "$m/2$" vezes 1

- Função de custo para regressão logística (rede D):

$$J_d\left(w_d\right) = -\frac{1}{m}\sum_{i=1}^{m}\left(\overline{y}_i \log D\left(\overline{x}_i\right) + \left(1 - \overline{y}_i\right)\log\left(1 - D\left(\overline{x}_i\right)\right)\right)$$

- Actulização dos pesos do Discriminador:

$$w_d = w_d - \alpha\frac{\partial J_d\left(w_d\right)}{\partial w_d}$$

- *Minibatch* de m exemplos de ruído $\left\{z_1, z_2, \ldots, z_m\right\} \in p_g\left(z\right)$.

$\hat{y} = \left\{1^m\right\}$ ▷ 0^m representa "m" vezes 0

- Função de custo para regressão logística (rede G):

$$J_g\left(w_g\right) = -\frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i \log\left(D\left(G(z_i)\right)\right) + \left(1 - \hat{y}_i\right)\log\left(1 - D\left(G(z_i)\right)\right)\right)$$

- Atualização dos pesos do Gerador:

$$w_g = w_g - \alpha\frac{\partial J_g\left(w_g\right)}{\partial w_g}$$

o desempenho desse modelo em um determinado problema (TANG et al., 2018) (ZHU et al., 2018). A Figura 8 ilustra os dígitos gerados por uma arquitetura GAN treinada com um conjunto de dados de dígitos manuscritos, onde a coluna mais à direita (coluna amarela) representa o vizinho maís próximo da penúltima coluna, mostrando que a arquitetura GAN não memoriza o conjunto de treinamento (causando sobreajuste), pelo contrário, gera novos exemplos semelhantes aos da distribuição original (dados de treinamento).

Figura 8. Dígitos sintéticos gerados por uma arquitetura GAN

Fonte: Extraido de (GOODFELLOW et al., 2014)

Não entanto, o desempenho das arquiteturas GAN pode ser avaliado em base a

diversas métricas. Em (BORJI, 2019) são listadas 24 métricas que validam o desempenho da GAN quantitativamente e outras 5 métricas que a validam qualitativamente. Destacando que algumas destas métricas estão relacionadas à medição da qualidade das imagens sintéticas geradas pela GAN, o que não é aplicável nesta pesquisa, uma vez que os dados a serem aumentados não representam imagens.

Atualmente, duas métricas principais para validar modelos de AM são empregadas em redes GAN. Uma primeira métrica chamada *Recall* (Equação 2.1), a qual permite calcular a proporção de sucesso da rede D com base na classe positiva (dados reais) e uma segunda métrica chamada precisão dada pela Equação 2.4, onde FP (*False Positives*) representa a quantidade de dados pertencentes à classe negativa (dados sintéticos) que a rede D considera como pertencentes à classe positiva. Esta última métrica permite calcular a proporção da rede D de não classificar um exemplo negativo como pertencente à classe positiva. Enfatizando que neste tipo de arquiteturas, o objetivo é que estas duas métricas sejam o mais próximo possível de 0.5, o que ocorre quando $p_g = p_{data}$.

$$Precisão = \frac{TP}{TP + FP} \tag{2.4}$$

Além das métricas vistas acima, outra métrica de validação a ser empregada é o cálculo da divergência de *Kullback-Leibler* ($D_{KL}(P, Q)$) (ERVEN; HARREMOS, 2014), dada pela Equação 2.5. Esta métrica calcula a semelhança ou diferença entre duas funções de distribuição de probabilidade $P(x)$ e $Q(x)$, cujo resultado é igual a 1 para funções de probabilidade iguais e diferente de 1 caso contrário.

$$D_{KL}(P, Q) = \frac{1}{m} \sum_{i}^{m} P(x_i) \ln \frac{P(x_i)}{Q(x_i)} \tag{2.5}$$

Finalmente, outras métricas de validação baseadas na distancia (Euclidiana, Manhattan, entre outras) são implementadas para calcular a proximidade dos dados sintéticos gerados pela rede G em relação aos dados reais utilizados no treinamento da arquitetura GAN (BORJI, 2019).

3 TRABALHOS RELACIONADOS

O objetivo da seleção *in-silico* de candidatos a genes de referência empregando aprendizado de máquina é reduzir o tempo e os recursos utilizados no laboratório. Para isso, é necessário reduzir o número de candidatos iniciais, partindo da hipótese de que, em um conjunto de genes, todos os genes com nível de expressão diferente de zero e que não estão validados na literatura como GR, são considerados genes candidatos.

Atualmente, diversas abordagens são empregadas para a identificação *in-silico* de GR, a maioria delas empregando a distância Euclidiana como parâmetro de seleção. Em (BERGHOFF et al., 2017) é proposta uma metodologia adaptando o problema do fluxo de custo mínimo em uma rede para a seleção dos possíveis candidatos. A ideia é encontrar a rota de menor custo em um grafo direcionado e acíclico n_s-dimensional (onde n_s representa o número total de genes), desde o gene menos expresso diferente de zero (fonte) até o gene mais altamente expresso (sumidouro), considerando as seguintes restrições: i) qualquer aresta que conecte dois nós (genes) no grafo, deve ir na direção do nó menos ranqueado para o mais ranqueado, onde o *ranking* é o nível médio de expressão de cada gene; ii) o custo de percorrer cada aresta é calculado com base na distância Euclidiana, além de outros parâmetros de ajuste os quais garantem a inclusão de certos GR no percurso; iii) no final, com base em 6 GR os quais estão presentes no percurso, são encontrados 27 possíveis candidatos no conjunto de dados da *Escherichia coli* MG1655.

Outra abordagem empregando a distância Euclidiana para a seleção dos genes candidatos é proposta em (FRANCO et al., 2019). Nesta abordagem, é construído um algoritmo de *clustering* baseado no algoritmo *k-means*, onde o número k de *clusters* é definido pelo índice de validade SD e pelo índice Dunn (LEGÁNY; JUHÁSZ; BABOS, 2006). No final, com base em 9 GR encontrados na literatura e com o objetivo de reduzir a quantidade de genes candidatos, apenas são levados em consideração os *clusters* em que existe pelo menos um gene de referência. Assim, com base nos *clusters* restantes e tendo o coeficiente de variação e o desvio padrão como critérios de seleção, 76 possíveis candidatos são selecionados.

Em (VANDESOMPELE et al., 2002) é proposta uma estratégia chamada *geNorm*, a qual identifica os genes com a expressão gênica mais estável baseado na hipótese de que a razão de expressão de dois genes de controle é idêntica em todas as amostras, independentemente da condição experimental ou tipo de célula. Assim, para cada gene, a variação em pares com todos os outros genes é determinada como o desvio padrão das razões de expressão transformadas logaritmicamente. Em seguida, a medida de estabilidade M do gene é definida como a variação média em pares de um gene em particular com

todos os outros. Assim, os genes com o menor valor de M são os genes mais estáveis, sendo os melhores candidatos.

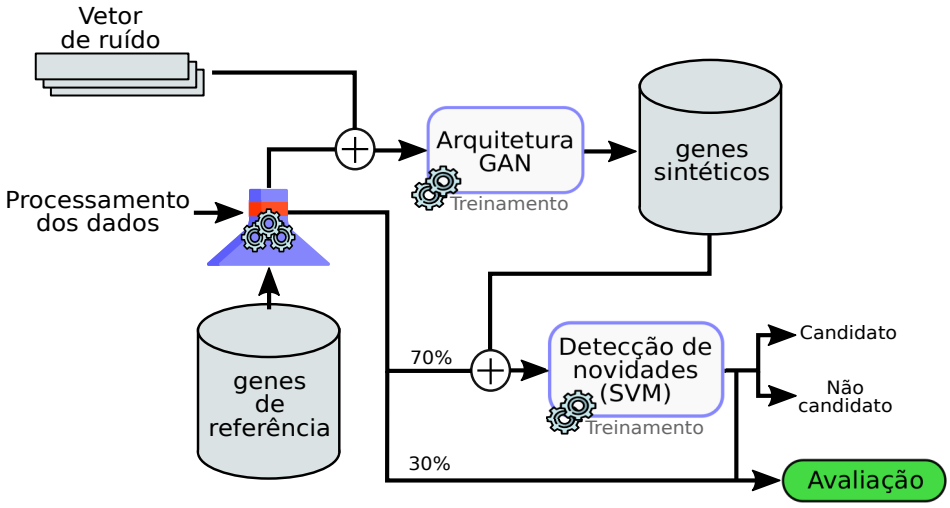
Conforme apresentado, existem diversos tipos de abordagens para a seleção *in-silico* de candidatos a GR. No entanto, os métodos atualmente propostos são baseados em um pequeno número de GR, limitando a seleção de candidatos à relação que esses genes têm com os GR, e priorizando os genes candidatos que apresentam uma maior similitude com os GR. Portanto, quanto maior seja a quantidade e diversidade dos GR utilizados, maior poderia ser a diversidade de candidatos encontrados.

Para mitigar a limitação gerada pelo uso de poucos GR na seleção dos genes candidatos, esta pesquisa propõe uma metodologia baseada no aumento do conjunto de GR a serem utilizados através de modelos adversários, mais precisamente, mediante a criação e treinamento de uma Rede Geradora Adversária (GAN). Finalmente, com os dados aumentados pela arquitetura GAN, é treinado um modelo de classificação de uma classe baseado em *support vector machine* (SVM) o qual servirá para identificar novos candidatos a GR no conjunto de dados *Escherichia coli* MG1655.

4 Método Proposto

Este capítulo tem como objetivo mostrar todos os detalhes do método proposto para a seleção de genes candidatos a genes de referência. A Figura 9 resume o método proposto, que consiste em duas etapas principais. Primeiro, o número de genes de referência disponíveis no conjunto de dados é aumentado empregando uma arquitetura GAN proposta. Segundo, é treinado um classificador baseado em detecção de novidades e máquinas de vetores de suporte (SVM) usando 70% dos genes de referência e alguns genes sintéticos gerados pela arquitetura GAN na primeira etapa. O objetivo é identificar, por meio do modelo treinado na segunda etapa, se um novo gene pode ou não ser um gene de referência. Observe que o modelo de detecção de novidades é avaliado com 30% dos genes de referência não empregados no treinamento.

Figura 9. Metodologia proposta



Fonte: Autor

As subseções a seguir explicam o conjunto de dados empregado nesta pesquisa, o pré-processamento realizado nos dados para posterior aumento através da arquitetura GAN proposta. Finalmente, é explicada a operação do detector de novidades baseado em SVM para a identificação de genes candidatos.

4.1 Base de dados

Esta pesquisa usa a base de dados RNA-seq de *Esquerichia coli* MG1655 disponível em (BERGHOFF et al., 2017). Este conjunto de dados consiste em 4293 genes que foram submetidos a condições de estresse. Três amostras diferentes foram removidas aos 0, 30 e 90 minutos e imediatamente misturadas com 0,25 vol de *RNA stop solution* (95% de etanol,

5% de fenol) em gelo. Observe que o conjunto de dados possui 9 características, 3 para cada tempo de amostragem mecionado acima, sendo que cada característica representa o nível de expressão gênica de cada gene em cada intervalo de tempo.

Para a identificação do maior número de GR neste conjunto de dados, foi necessária uma revisão da literatura, que consistiu em coletar os genes que foram pelo menos testados em três estudos diferentes. Encontrando 21 genes de referência no total, 15 em (ROCHA; SANTOS; PACHECO, 2015) e 6 em (BERGHOFF et al., 2017).

4.2 Pré-processamento dos genes

O conjunto de dados RNA-seq foi normalizado por meio de RPKM (*Reads per kilobase per million mapped reads*) e aplicando $(log_2 + 1)$ para mitigar o número de outliers e aumentar a proximidade das características dos dados. Além destas normalizações, e com base no fato de que os genes de referência têm um nível mínimo de variação em sua expressão gênica (DIE et al., 2010), é realizado um teste de estabilidade com base no coeficiente de variação CV dado pela Equação 4.1, onde σ representa o desvio padrão do conjunto de características de cada gene e μ representa sua média.

$$CV = \frac{\sigma}{\mu} \tag{4.1}$$

O teste de estabilidade consiste na seleção dos genes para os quais seu valor de CV está dentro de um intervalo interquartil dado por $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q3 - Q1)]$ com $k = 1,5$. Observe que os genes com um valor de CV fora desse intervalo não são considerados nesta pesquisa. Finalmente, os dados são normalizados entre $[-1, 1]$ empregando a Equação 4.2, onde X_t representa os dados transformados, x_i representa o *i-th* gene, e X_{max} e X_{min} representam os valores máximo e mínimo para cada característica do conjunto de dados. Este tipo de normalização entre $[-1, 1]$ foi adotado porque a última camada da rede geradora G na arquitetura GAN foi proposta com uma função de ativação *tanh*.

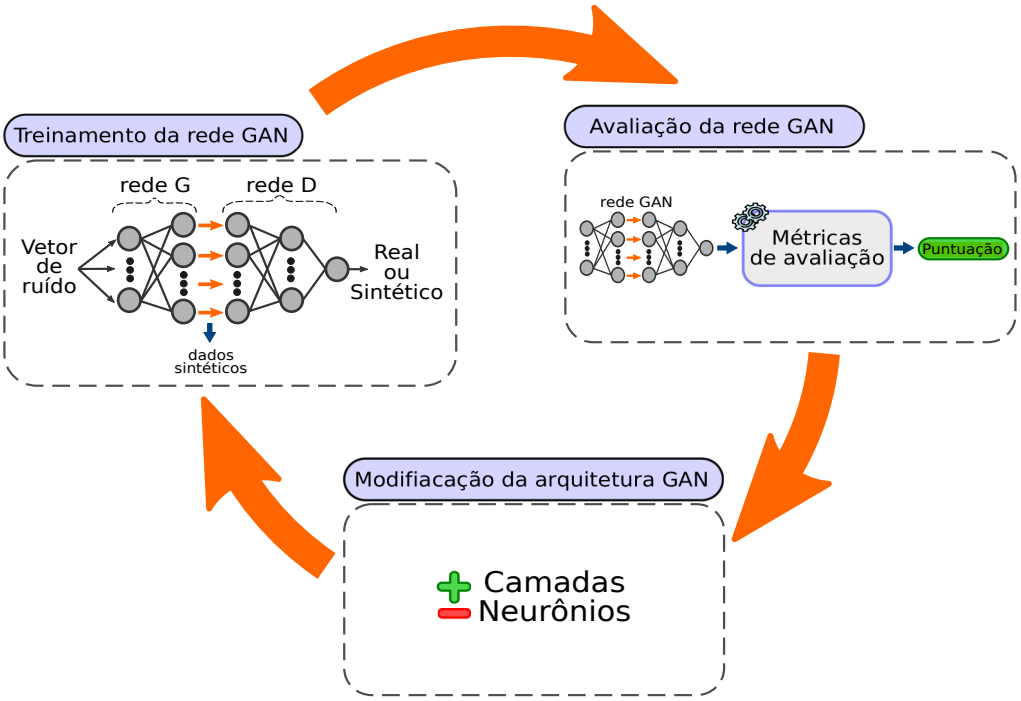
$$X_t = \frac{2(x_i - X_{min})}{X_{max} - X_{min}} - 1 \tag{4.2}$$

4.3 Construção da aquitetura GAN

Com os genes já processados, o próximo passo é construir uma arquitetura GAN para aumentar a quantidade de genes de referência. Para isso, a seleção do número de camadas e neurônios em cada camada é feita com base nas métricas de avaliação propostas anteriormente. A Figura 10 ilustra o processo de construção da arquitetura GAN a ser

implementada, onde o ciclo é repetido até que se obtenha uma arquitetura estável. Note que a seleção de camadas e neurônios, embora baseada nas métricas propostas, também possui um pouco de heurística.

Figura 10. Construção da arquitetura GAN



Fonte: Autor

Finalmente, a partir da arquitetura GAN mais estável, são gerados genes de referência sintéticos para aumentar a quantidade de genes de referência a serem empregados no treinamento do classificador. Esta quantidade de genes sintéticos a serem gerados é baseada no aumento do desempenho do classificador.

4.3.1 Avaliação da arquitetura GAN

Para avaliar o desempenho da arquitetura GAN proposta, serão empregadas as métricas de validação mencionadas no Capítulo 2, além de uma nova métrica de similaridade $S(x, x')$ dada pela Equação 4.3, onde x' representa o conjunto de dados sintéticos gerados pela rede G e $\hat{y}$ representa a classe prevista pela rede D para os dados sintéticos, a qual toma valores iguais a 1 para dados reais (GR) e 0 para dados sintéticos (gerados pela rede G). Os parâmetros n_f, n_g e m representam o número de características de cada gene, o número de genes sintéticos e o número de genes de referência no conjunto de treinamento, respectivamente. Observe que $x_i^{(k)}$ representa a característica k no i-th gene.

$$S\left(x, x'\right) = \sum_{i}^{m} \sum_{j}^{n_g} \sum_{k}^{n_f} \frac{|x_i^{(k)} - x'^{(k)}_j|}{n_f n_g m} + |0.5 - \frac{1}{n_g} \sum_{j}^{n_g} \hat{y}_j| \tag{4.3}$$

Esta métrica de similaridade proposta permite combinar dois fatores importantes para validar a qualidade dos genes sintéticos gerados. O primeiro termo da métrica permite calcular, em média, quão próximos os genes sintéticos estão dos reais, por sua vez, o segundo termo da métrica permite calcular o quão "confusa" a rede D está. Observe que o segundo termo da métrica tende a zero, o que ocorre quando a rede D falha ao distinguir dados reais de dados sintéticos ($D(x) = \frac{1}{2}$). Note também, que um valor $S(x, x')$ proximo de zero indica que os dados sintéticos são mais semelhantes aos reais.

4.4 Identificação dos genes candidatos

Para a identificação dos genes candidatos, é empregado um detector de novidades baseado em Máquinas de vetores de suporte (SCHÖLKOPF et al., 2000). O objetivo do detector é estimar uma função f, com $f \geq 0$ para exemplos positivos e $f < 0$ para qualquer outro exemplo. Nesta estratégia, os dados são mapeados com uma função Φ em um novo espaço característico F e separados da origem com uma margem máxima.

Considere o conjunto de treinamento $x_1, \ldots, x_\ell \in \mathcal{X}$, onde $\ell \in \mathbb{N}$ é o número de observações e $\mathcal{X}$ é um conjunto $\in \mathbb{R}^N$. Para separar o conjunto de treinamento da origem, o seguinte problema quadrático é resolvido:

$$\min_{w \in F, \xi \in \mathbb{R}^\ell, \rho \in \mathbb{R}} \frac{1}{2}\|w\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho \tag{4.4}$$

$$s.t. \quad f(x) = (w \cdot \Phi(x_i) - \rho) \geq -\xi_i,\ \xi_i \geq 0,\ i = 1, \ldots, \ell \tag{4.5}$$

Derivando o problema duplo, pode-se mostrar que a solução possui uma expansão de vetor de suporte:

$$f(x) = sgn\left(\sum_i \alpha_i k(x_i, x) - \rho\right) \tag{4.6}$$

onde a função sgn é igual a 1 para valores maiores ou iguais a zero e -1 caso contrário. Os padrões x_i com α_i diferentes de zero, são chamados vetores de suporte, e os coeficientes são encontrados ao resolver o problema duplo:

$$\min_\alpha \frac{1}{2}\sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \quad s.t\ \ 0 \leq \alpha_i \leq \frac{1}{\nu\ell},\ \sum_i \alpha_i = 1. \tag{4.7}$$

Nesta abordagem, o parâmetro ν é um *trade-off* entre os dados normais e anormais no conjunto de dados, o qual será ajustado com base na pontuação *recall* (Equação (2.1)). A função de mapeamento Φ é representada pelo kernel gaussiano fornecido na Equação (4.8), onde γ é igual a $1/n_{fe}$ e n_{fe} é o número de caracteristicas do conjunto de treinamento.

$$k(x, y) = e^{-\gamma\|x-y\|^2} \tag{4.8}$$

Se o problema duplo proposto for resolvido, a função de decisão $f(x)$ terá valores positivos para o maior número de amostras x_i contidas no conjunto de treinamento. Observe que nesta abordagem, o detector de novidades é treinado com 70% dos genes de referência encontrados na literatura mais os genes sintéticos gerados a partir da arquitetura GAN proposta. Finalmente, com o detector de novidades treinado, os genes restantes no conjunto de dados são rotulados como genes candidatos (se $f \geq 0$) ou genes não candidatos (se $f < 0$).

Para validar o detector de novidades, e com base no fato de que apenas a classe positiva está disponível (genes de referência), é implementada a métrica *recall* (Equação (2.1)), como a única métrica de validação do detector. Com esta métrica, pode-se avaliar a capacidade do detector de novidades em rotular exemplos positivos como pertencentes a essa classe.

5 Resultados Parciais

Esta seção apresenta os resultados parciais e as tecnologias utilizadas para a identificação de candidatos a GR com base no uso da GAN para aumentar sinteticamente o conjunto de treinamento e, com base nesse conjunto, criar um detector de novidades baseado en SVM. O foco desta pesquisa foi baseada na criação da arquitetura GAN e no detector de novidades para a identificação de possíveis candidatos a GR no conjunto de dados *Escherichia coli* MG1655 obtido de (BERGHOFF et al., 2017).

5.1 Tecnologias Utilizadas

Este estudo foi desenvolvido inteiramente em Python, em sua versão 3.6.5, e seu código fonte está disponível publicamente para a comunidade no GitHub (RUEDA, 2020). A seguir, são apresentadas as bibliotecas que foram usadas no desenvolvimento da pesquisa, e uma breve descrição do papel que desempenharam.

Tensorflow. Na sua versão 2.2.0. Esta biblioteca foi usada para a construção e o treinamento da arquitetura GAN proposta.

Scikit-learn. Esta biblioteca de aprendizado de máquina em Python foi usada para a criação e o treinamento do detector de novidades, bem como para a decomposição do conjunto de dados em 2 componentes principais. Foi utilizada dua versão 0.21.3.

Numpy. Na versão 1.18.1, esta biblioteca foi utilizada para o processamento inicial dos dados.

Pandas. Esta biblioteca, em sua versão 0.25.1, foi usada para carregar o conjunto de dados em python, bem como para parte do processamento inicial.

Bokeh. Na versão 1.4.0, foi utilizada para representar graficamente os resultados obtidos nesta pesquisa.

5.2 Processamento inicial dos genes

O primeiro paso no pré-processamento dos genes foi remover todos os genes cujo valor de expressão era igual a zero (genes não expressos) para depois desconsiderar os genes para os quais o CV (Equação 4.1) não estava dentro da faixa interquartil (considerados *outliers*). A Tabela 1 mostra o processamento de dados para obter o conjunto final de dados. Lembrando que o conjunto final de dados foi normalizado entre [-1, 1] (Equação 4.2),

devido a que foi implementada a função *tanh* (Figura 6b) como a função de ativação da última camada da rede geradora G.

Tabela 1. Etapas do processamento dos dados

Número inicial de genes	Número de genes expressos	*Outliers* baseados no CV	Dataset Final
4293	4191	3	4188

Dos 21 GR identificados na literatura para *Escherichia coli*, o gene *idnT* não foi levado em consideração devido a que seu CV estava fora do intervalo interquartil calculado para os genes de referência. Observe que foram calculadas duas faixas interquartis, uma baseada nos genes de referência e a outra baseada nos demais genes. A Tabela 2 mostra a lista dos 20 GR usados nesta pesquisa.

Tabela 2. Lista dos genes de referência usados nesta pesquisa

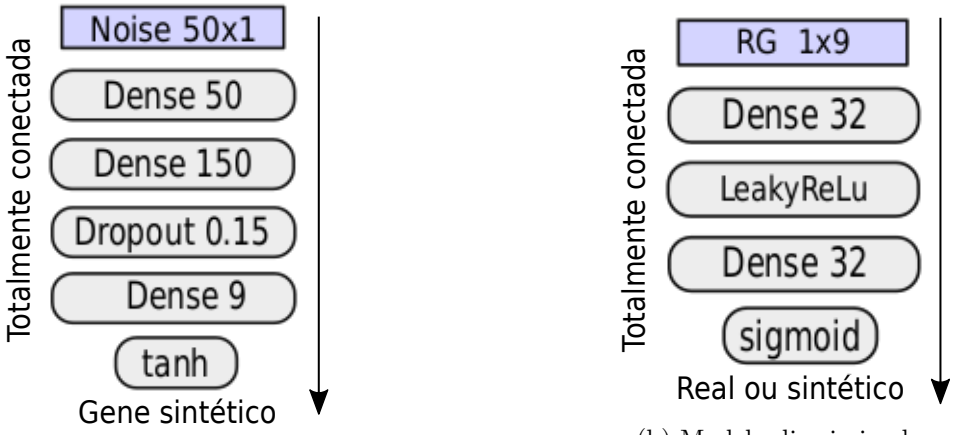
Gene	Produto
adk	Adenylate kinase
cysG	Siroheme synthase
dnaG	DNA primase
ftsZ	Cell division protein ftsZ
gmK	Guanylate kinase
glnA	Glutamine synthetase
gyrA	DNA gyrase subunit A
gyrB	DNA gyrase subunit B
hcaT	Putative transport protein
ihfB	Integration host factor subunit beta
rho	Transcription termination factor
recA	Protein RecA
recF	DNA replication and repair protein RecF
rpoA	RNA polymerase subunit alpha
rpoB	RNA polymerase subunit beta
rpoC	RNA polymerase subunit beta
rpoD	RNA polymerase sigma factor RpoD
rrSA	16S ribosomal RNA (rrsA)
secA	Protein translocase subunit SecA
ssrA	tmRNA

5.3 Construção e treinamento da arquitetura GAN

Com base no processo de construção da arquitetura GAN (Figura 10), foi proposta uma arquitetura baseada em camadas densas totalmente conectadas. A Figura 11 ilustra

a arquitetura GAN proposta, onde o modelo gerador G (Figura 11a) recebe como entrada um vetor de ruído com base em uma distribuição normal $N(0, 1)$ e produz como saída um gene de referência sintético. Por outro lado, o modelo discriminador D (Figura 11b) toma como entrada genes de referência reais (distribuição real) e genes de referência sintéticos (gerados pela rede G) e sua saída representa a probabilidade de um gene pertencer à distribuição dos dados reais.

Figura 11. Arquitetura GAN proposta.



(a) Modelo gerador

(b) Modelo discriminador

Fonte: Autor

Para o treinamento da arquitetura GAN proposta (Figura 11), foi empregado o algoritmo *stochastic gradient descent* SGD (Algoritmo 1), usando diferente taxa de aprendizado α, para o modelo gerador como para o modelo discriminador. Essas taxas de aprendizado diminuíram em cada época por um fator igual a $\frac{\alpha}{épocas}$ (*decay rate*) para melhorar a convergência da arquitetura GAN. Por sua vez, os pesos iniciais de cada camada em cada modelo foram inicializados com base no *Xavier uniform initializer* (GLOROT; BENGIO, 2010), o qual inicializa os pesos com base em uma distribuição uniforme $U[-t, t]$, onde t é igual a $\sqrt{6/\left(n_j + n_{j+1}\right)}$ sendo n_j o número de neurônios de entrada e n_{j+1} o número de neurônios de saída na camada j.

Finalmente, foi feito um ajuste de parâmetros nas redes G e D para atingir o óptimo global ($D(x) = \frac{1}{2}$). Por este motivo, os pesos das duas redes (θ_g e θ_d) foram atualizados com base em todos os genes de referência ($minibacth = 20$), pois isso permitiu que a rede convergisse mais rapidamente. Observe que os pesos da rede D foram atualizados com base em 40 genes (20 genes de referência e 20 genes sintéticos gerados pela rede G). No final, foram necessárias 1700 épocas para treinar a arquitetura GAN proposta, sabendo que uma época ocorre quando todos os genes no conjunto de treinamento são iterados (um *minibacth*). A Tabela 3 mostra os parâmetros empregados para treinar a arquitetura GAN.

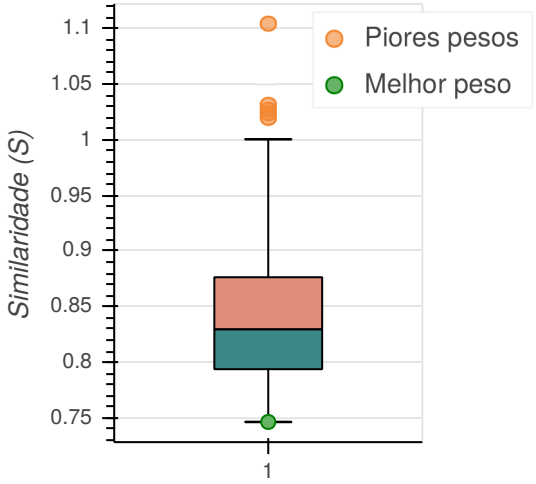
Com o ajuste dos parâmetros feito, a próxima etapa foi escolher o melhor conjunto

Tabela 3. Parâmetros empregados para o treinamento da arquitetura GAN.

Parâmetro	Gerador	Discriminador
Optimizer	SGD	SGD
Taxa de aprendizado	0.00015	0.001
Decay rate	0.00015/1700	0.001/1700
Momentum	0.92	0.9
Épocas	1700	1700

de pesos iniciais para a arquitetura GAN. Para isso, baseados na métrica de similaridade S (Equação 4.3), foi treinada a arquitetura GAN em 100 ocasiões diferentes e, para cada uma dessas ocasiões, foi calculada a similaridade média S com base em 30 repetições, gerando 300 genes sintéticos em cada repetição. A Figura 12 ilustra um *boxplot* com os diferentes valores de similaridade calculados, sabendo que o melhor conjunto de pesos iniciais é aquele com o menor valor de S (círculo verde).

Figura 12. Similaridade (S) para cada conjunto de pesos inicias.



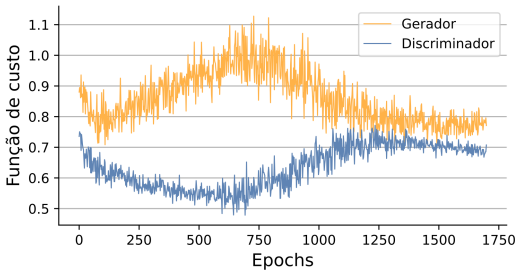
Fonte: Autor

Uma vez selecionado o melhor conjunto de pesos iniciais, o próximo passo foi treinar a arquitetura GAN. A Figura 13 ilustra o processo de convergência da arquitetura GAN proposta, onde a Figura 13a ilustra como as funções de custo $J_d(w_d)$ e $J_g(w_g)$ de ambas redes convergem à medida que o número de épocas aumenta, até atingir um mínimo local próximo a 0.7. Por sua vez, a Figura 13b ilustra como a precisão do discriminador D diminui, tendendo a 0.5, o ótimo global. Estes resultados indicam que a rede G está gerando genes sintéticos semelhantes aos genes reais, tornando a rede D incapaz de distinguir entre genes reais e genes sintéticos.

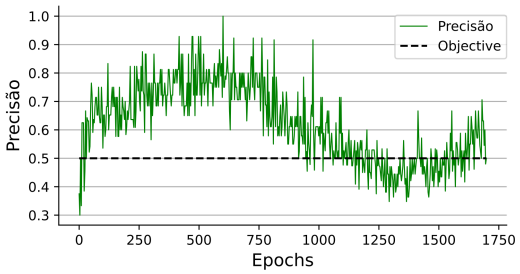
Para retificar que a arquitetura GAN proposta não está apenas convergindo, mas

também está gerando dados sintéticos semelhantes aos reais, é calculada em cada época a divergência de *Kullback-Leibler* (Equação 2.5). A Figura 13c ilustra como, à medida que a quantidade de épocas aumenta, a divergência diminui (tende a zero), o que indica que a rede G está conseguindo recuperar a distribuição de probabilidade dos dados reais.

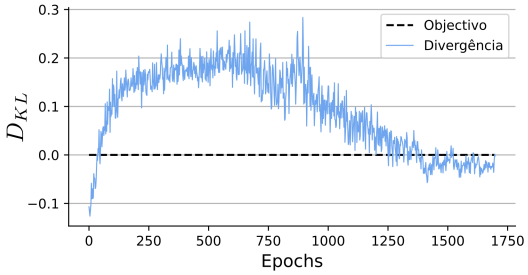
Figura 13. Processo de convergência no treinamento da arquitetura GAN.



(a) Função de custo da GAN.



(b) Precisão da GAN.



(c) Divergência de *Kullback-Leibler*.

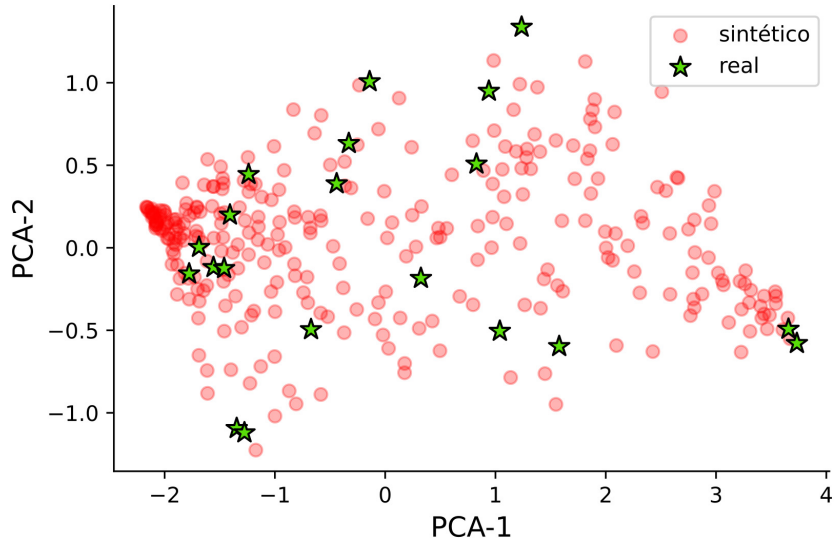
Fonte: Autor

Finalmente, para usar uma representação gráfica dos genes de referência e dos genes sintéticos, é utilizada a análise de componentes principais PCA (WOLD; ESBENSEN; GELADI, 1987), a qual permite observar visualmente a semelhança entre os genes reais e os genes sintéticos gerados pela rede G. A Figura 14 ilustra uma representação 2-PCA dos genes de referência (estrelas verdes) e de 300 genes sintéticos (círculos vermelhos) gerados pela rede G. Observe que os genes sintéticos não estão distantes dos genes reais, o que era de se esperar devido aos resultados obtidos no processo de convergência da arquitetura proposta (Figura 13).

5.4 Seleção dos genes sintéticos

Para treinar o detector de novidades, o qual mais tarde servirá para identificar genes candidatos à GR, precisa-se selecionar o melhor conjunto de genes sintéticos gerados pela rede G e, por sua vez, encontrar o melhor número desses genes, já que o resultado do detector de novidades pode variar de acordo com o tamanho do conjunto de treinamento. Para isso, é proposta uma métrica E (Equação 5.1), a qual permite medir a qualidade de um conjunto de genes sintéticos com base em seu coeficiente de variação (Equação 4.1) e

Figura 14. Análise de componentes principais emtre os genes reais e sintéticos.



Fonte: Autor

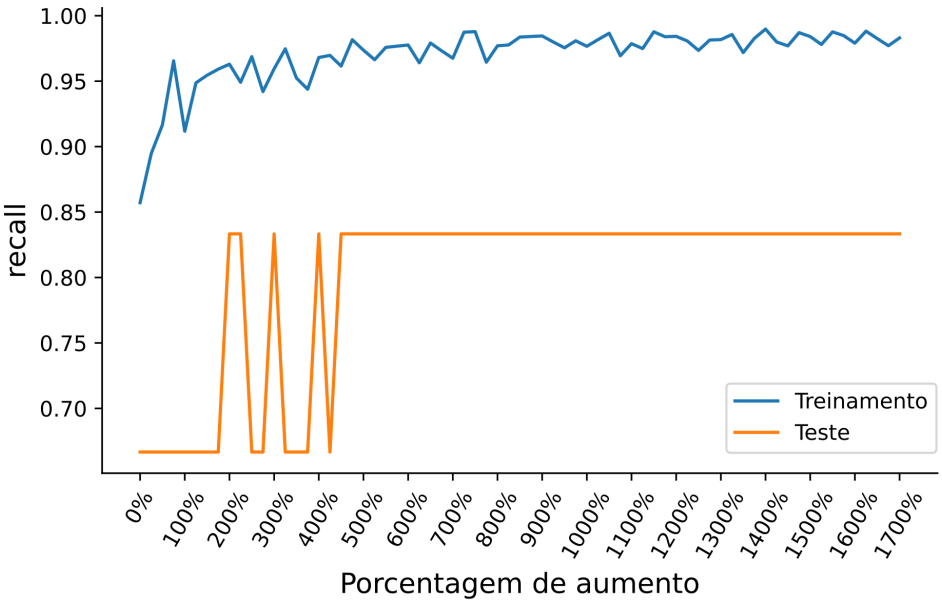
na probabilidade calculada pela rede D para cada um deles, assim, com base nesta métrica, se pode selecionar o melhor conjunto de genes sintéticos, assumindo que se conhece a quantidade de genes sintéticos a serem gerados. Observe que x'_j representa o j-ésimo gene sintético e n_g o número de genes sintéticos gerados.

$$E(x') = \frac{1}{n_g} \sum_{j}^{n_g} \left[CV(x'_j) + \frac{1 - D(x'_j)}{D(x'_j)} \right] \tag{5.1}$$

Para selecionar o número de genes sintéticos necessários, a escolha foi baseada no desempenho do detector de novidades, tanto no conjunto de treinamento quanto no conjunto de teste. Para fazer isso, foi treinado o detector de novidades proposto aumentando gradualmente os dados de treinamento, começando com um aumento do 25%, até atingir um aumento do 1700%. Para cada um desses conjuntos de dados aumentados, foi treinado o detector de novidades ajustando seu parâmetro ν (Equação 4.7) com base na pontuação *recall* (Equação 2.1). A Figura 15 ilustra como a pontuação *recall* se comporta à medida que o conjunto de treinamento é aumentado, o que permite observar que, ao aumentar os dados sinteticamente, se consegue aumentar a pontuaçaõ *recall* do modelo, embora, nos dados do teste, esta pontuação atinja um máximo local de 83.33%, devido a que o detector é avaliado apenas com 30% dos genes de referência reais (Figura 9).

Finalmente, da Figura 15, pode-se deduzir que aumentar o conjunto de genes iniciais em mais de 500% não melhorará o desempenho do detector de novidades; portanto, o conjunto de datos foi aumentado com base nessa porcetagem, e usando a métrica E (Equação 6.1) foi selecionado o melhor conjunto de genes sintéticos. A Figura 16 ilustra o conjunto de dados aumentados, onde as estrelas verdes representam os GR reais, e os

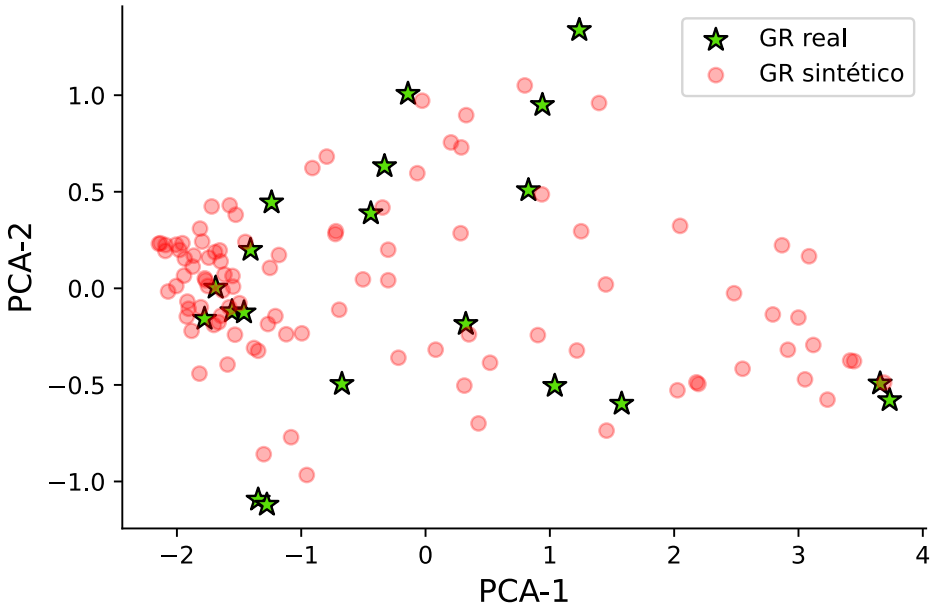
Figura 15. Pontuação recall vs. aumento do conjunto de treinamento.



Fonte: Autor

círculos vermelhos representam os GR sintéticos. Observe que nesta representação foi usado o 100% dos genes de referência.

Figura 16. Representação 2-PCA do conjunto de dados aumentado.



Fonte: Autor

5.4.1 Seleção dos genes candidatos

Para selecionar os genes candidatos, foi treinado o detector de novidades proposto (Equação 4.6) com 70% dos genes de referência (Tabela 2) e com 100 GR sintéticos gerados na seção anterior, o que permitiu aumentar a diversidade do conjunto de treinamento e, assim, aumentar a precisão do detector de novidades.

A validação do detector de novidades foi realizada mediante a Equação (2.1) com quatro conjuntos de dados diferentes. A Tabela 4 mostra os resultados do detector proposto para cada um desses conjuntos de dados. Onde no conjunto de treinamento, foram calculadas duas pontuações *recall* diferentes, a primeira para o conjunto de dados aumentado (empregado no treinamento) e a segunda com base no 70% dos GR escolhidos para o treinamento do detector (GR reais). Por sua vez, para a validação do classificador nos dados de teste, os dados foram divididos em dois conjuntos, o primeiro conjunto com o 30% dos GR restantes, e o segundo conjunto baseado apenas em novos GR sintéticos gerado pela rede G.

Tabela 4. Resultado da pontuação *recall* nos diferentes conjuntos de dados.

Métrica	Dados de treinamento		Dados de teste	
	Dados aumentados	Genes de referência (70%)	Genes de referência (30%)	Genes sintéticos
Recall	97.36%	92.85%	83.33%	92.73%

Para observar como o aumento do conjunto de dados através da arquitetura GAN proposta melhora o desempenho do detector de novidades, o detector proposto é treinado com base só nos genes de referência (70% para o treinamento e 30% para o teste), e sua pontuação *recall* obtida é comparada com a pontuação *recall* do detector treinado com os dados aumentados. A Tabela 5 mostra a comparação do desempenho dos dois detectores, sendo que o detector que apresenta a maior pontuação é aquele que foi treinado com o conjunto de dados aumentado.

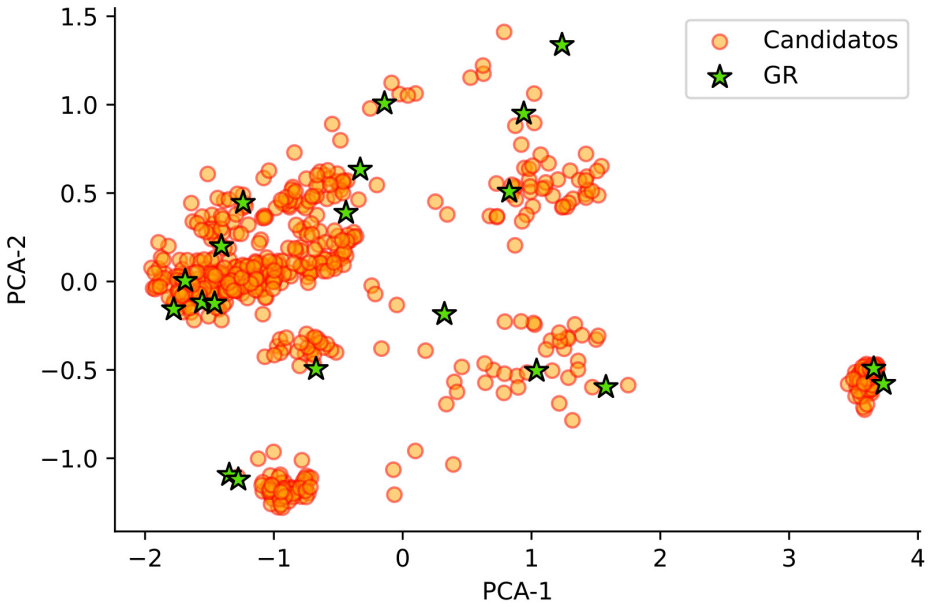
Tabela 5. Pontuação *recall* nos dados aumentados e sem aumentar.

Genes de referência	Pontuação *recall*	
	Dados de treinamento	Dados de teste
Aumentados	97.36%	83.33%
Sem aumentar	85.71%	66.66%

Depois de treinar o detector de novidades com os dados aumentados, e com base nos 4168 genes não classificados no conjunto de dados de *Escherichia coli*, foram considerados como possíveis candidatos a GR os genes para os quais a função $f(x)$ (Equação (4.6)) foi igual a 1. Observe que esta função assume valores iguais a 1 para dados dentro da fronteira de decisão (classe positiva) e valores iguais a -1 caso contrário. A Figura 17

ilustra a representação 2-PCA dos GR e dos genes candidatos selecionados pelo detector proposto. Nesta representação, pode-se perceber que o detector proposto gera candidatos próximos aos GR conforme o esperado, o que indica que o espaço mapeado criado pelo detector está sendo consistente com os dados do treinamento.

Figura 17. Representação 2-PCA dos genes de referência e dos genes candidatos.



Fonte: Autor

Finalmente, com o método proposto, consegue-se identificar 753 possíveis candidatos a genes de referência, reduzindo o número de possíveis genes a serem avaliados em laboratório em 81,93%. A Tabela 6 mostra os 11 genes candidatos comuns encontrados nesta pesquisa e, em (BERGHOFF et al., 2017), os quais são considerados mais relevantes, pois são selecionados por dois métodos propostos diferentes.

Tabela 6. Os genes candidatos mais relevantes.

Gene	Product
ftsX	Cell division protein ftsX
ftsY	Cell division protein ftsY
glyY	$tRNA_{glyY}$
mutY	Adenine DNA glycosylase
ndk	Nucleoside diphosphate kinase
nfuA	iron-sulfur cluster scaffold protein
rrsE	16S ribosomal RNA(rrsE)
rrsG	16S ribosomal RNA (rrsG)
spoT	(p)ppGpp synthase
thrW	$tRNA_{thrW}$
zupT	heavy metal divalent cation transporter ZupT

6 Considerações Parciais

Esta pesquisa apresentou uma nova abordagem para a identificação *in-silico* de genes candidatos a genes de referência no conjunto de dados *Escherichia coli* MG1655. Esta abordagem consistiu em duas etapas principais. Primeiro, uma arquitetura GAN proposta foi treinada para aumentar sinteticamente o conjunto de genes de referência. Segundo, com o conjunto de dados aumentado foi treinado um classificador SVM de uma classe o qual obteve uma pontuação *recall* de 83.33% nos dados do teste. Esta nova abordagem permitiu a identificação de 753 genes candidatos a genes de referência de um total de 4170 genes expressos, o que reduz o número de genes a serem analisados em laboratório em até um 81%.

Por outro lado, por meio do treinamento da arquitetura GAN proposta, pode-se observar que, embora a arquitetura convergisse, ela não atingiu seu ideal global, o que pode ser devido a vários fatores, entre eles a baixa quantidade de dados de treinamento, a seleção dos parâmetros, ou até a própria arquitetura proposta, o que gera um erro nos dados gerados.

Um fato importante é que 11 dos genes candidatos selecionados nesta pesquisa já foram selecionados em outra pesquisa como genes candidatos (BERGHOFF et al., 2017), o que permite supor que o método proposto tem alguma concordância com os métodos já propostos na literatura.

Finalmente, a abordagem demonstrou, como esperado, que o aumento sintético do número de genes de referência melhora a pontuação do classificador, tendo nesta pesquisa um aumento do 16.67% em relação ao classificador treinado sem os dados aumentados, fazendo a seleção dos genes candidatos mais confiável.

6.1 Publicações

O método proposto e os resultados obtidos neste trabalho foram publicados como artigo completo na *International Conference on Computational Science and its Applications* (ICCSA 2020) sob o título: "*One-class SVM to identify candidates to Reference Genes based on the augment of RNA-seq data with Generative Adversarial Networks*".

6.2 Próximas Atividades

A seguinte etapa, planejada para a dissertação, é tentar aproximar a arquitetura GAN do seu ótimo global $D(x) = \frac{1}{2}$, para isso, a intenção é tentar com outros ajustes de

parâmetros e, por sua vez, tentar com outras arquiteturas GAN, isto com o fim de gerar dados sintéticos mais próximos dos reais.

Além disso, de acordo com a literatura, procurar-se-á reduzir o conjunto de candidatos seleccionados pelo método proposto com base no coeficiente de variação CV, uma vez que uma das principais características dos GR é a pouca variação na sua expressão gênica.

Por último, vai se tentar implementar o método proposto em outros conjuntos de dados já utilizados em outras pesquisas, o que servirá para comparar/validar a eficácia do método proposto, e por sua vez, outros classificadores de uma classe serão implementados com o fim de poder comparar os resultados do detector de novidades proposto até o momento.

6.3 Cronograma

A seguir, a Tabela 7 mostra o cronograma de actividades a desenvolver até à defesa da dissertação.

Tabela 7. Cronograma das atividades.

Atividades	**Período**					
	2020					
	Jul.	Ago.	Set.	Out.	Nov.	Dez.
1	×	×	×	×	×	×
2	×	×				
3		×	×			
4			×	×		
5			×	×		
6				×	×	
7					×	×
8						×

Definição das atividades a serem desenvolvidas:

1. Revisão da literatura sobre tópicos relacionados à pesquisa;
2. Procura de outros conjuntos de dados.
3. Novo ajuste de parâmetros e desenvolvimento de outras arquiteturas GAN;
4. Implementação e testagem de outros detectores de novidades;
5. Análise dos resultados;

6. Escrita de artigo;
7. Escrita da dissertação;
8. Defesa da dissertação.

Referências

AMRAEE, S.; VAFAEI, A.; JAMSHIDI, K.; ADIBI, P. Abnormal event detection in crowded scenes using one-class svm. *Signal, Image and Video Processing*, Springer, v. 12, n. 6, p. 1115–1123, 2018. Citado na página 11.

BERGHOFF, B. A.; KARLSSON, T.; KÄLLMAN, T.; WAGNER, E. G. H.; GRABHERR, M. G. Rna-sequence data normalization through in silico prediction of reference genes: the bacterial response to dna damage as case study. *BioData mining*, BioMed Central, v. 10, n. 1, p. 30, 2017. Citado 8 vezes nas páginas 10, 11, 22, 24, 25, 29, 37 e 38.

BORJI, A. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, Elsevier, v. 179, p. 41–65, 2019. Citado na página 21.

CARUANA, R.; NICULESCU-MIZIL, A. An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd international conference on Machine learning*. [S.l.: s.n.], 2006. p. 161–168. Citado na página 14.

CHU, Y.; COREY, D. R. Rna sequencing: platform selection, experimental design, and data interpretation. *Nucleic acid therapeutics*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 22, n. 4, p. 271–274, 2012. Citado na página 13.

DANIEL, G. *Principles of artificial neural networks*. [S.l.]: World Scientific, 2013. v. 7. Citado na página 16.

DHEDA, K.; HUGGETT, J.; CHANG, J.; KIM, L.; BUSTIN, S.; JOHNSON, M.; ROOK, G.; ZUMLA, A. The implications of using an inappropriate reference gene for real-time reverse transcription pcr data normalization. *Analytical biochemistry*, Academic Press, v. 344, n. 1, p. 141–143, 2005. Citado na página 13.

DIE, J. V.; ROMÁN, B.; NADAL, S.; GONZÁLEZ-VERDEJO, C. I. Evaluation of candidate reference genes for expression studies in pisumsativum under different experimental conditions. *Planta*, Springer, v. 232, n. 1, p. 145–153, 2010. Citado na página 25.

ERVEN, T. V.; HARREMOS, P. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, IEEE, v. 60, n. 7, p. 3797–3820, 2014. Citado na página 21.

FRANCO, E. F.; MAUÉS, D.; ALVES, R.; GUIMARÃES, L.; AZEVEDO, V.; SILVA, A.; GHOSH, P.; MORAIS, J.; RAMOS, R. T. A clustering approach to identify candidates to housekeeping genes based on rna-seq data. In: SPRINGER. *Brazilian Symposium on Bioinformatics*. [S.l.], 2019. p. 83–95. Citado 3 vezes nas páginas 10, 11 e 22.

GAUTHIER, J.; VINCENT, A. T.; CHARETTE, S. J.; DEROME, N. A brief history of bioinformatics. *Briefings in bioinformatics*, Oxford University Press, v. 20, n. 6, p. 1981–1996, 2019. Citado na página 10.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. [S.l.: s.n.], 2010. p. 249–256. Citado na página 31.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 2672–2680. Citado 2 vezes nas páginas 19 e 20.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015. Citado na página 14.

KIM, Y.; KIM, Y.; KIM, Y. H. Evaluation of reference genes for gene expression studies using quantitative real-time pcr in drosophila melanogaster after chemical exposures. *Journal of Asia-Pacific Entomology*, Elsevier, 2020. Citado 2 vezes nas páginas 10 e 13.

KUKURBA, K. R.; MONTGOMERY, S. B. Rna sequencing and analysis. *Cold Spring Harbor Protocols*, Cold Spring Harbor Laboratory Press, v. 2015, n. 11, p. pdb–top084970, 2015. Citado na página 13.

LEGÁNY, C.; JUHÁSZ, S.; BABOS, A. Cluster validity measurement techniques. In: WORLD SCIENTIFIC AND ENGINEERING ACADEMY AND SOCIETY (WSEAS) STEVENS POINT .... *Proceedings of the 5th WSEAS international conference on artificial intelligence, knowledge engineering and data bases*. [S.l.], 2006. p. 388–393. Citado 2 vezes nas páginas 10 e 22.

LONG, X.; LU, J.; KAV, N. N.; QIN, Y.; FANG, Y. Identification and evaluation of suitable reference genes for gene expression analysis in rubber tree leaf. *Molecular Biology Reports*, Springer, p. 1–13, 2020. Citado na página 13.

OLSVIK, P. A.; SØFTELAND, L.; LIE, K. K. Selection of reference genes for qrt-pcr examination of wild populations of atlantic cod gadus morhua. *BMC Research Notes*, Springer, v. 1, n. 1, p. 47, 2008. Citado na página 10.

ROCHA, D. J.; SANTOS, C. S.; PACHECO, L. G. Bacterial reference genes for gene expression studies by rt-qpcr: survey and analysis. *Antonie Van Leeuwenhoek*, Springer, v. 108, n. 3, p. 685–693, 2015. Citado na página 25.

ROSENBLATT, F. Perceptron simulation experiments. *Proceedings of the IRE*, IEEE, v. 48, n. 3, p. 301–309, 1960. Citado na página 16.

RUEDA, E. *ejrueda/20_ICCSA_RG_GAN: zenodo release*. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.3934376>. Citado na página 29.

SCHÖLKOPF, B.; WILLIAMSON, R. C.; SMOLA, A. J.; SHAWE-TAYLOR, J.; PLATT, J. C. Support vector method for novelty detection. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2000. p. 582–588. Citado na página 27.

TANG, B.; TU, Y.; ZHANG, Z.; LIN, Y. Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access*, IEEE, v. 6, p. 15713–15722, 2018. Citado na página 20.

VANDESOMPELE, J.; PRETER, K. D.; PATTYN, F.; POPPE, B.; ROY, N. V.; PAEPE, A. D.; SPELEMAN, F. Accurate normalization of real-time quantitative rt-pcr data by geometric averaging of multiple internal control genes. *Genome biology*, BioMed Central, v. 3, n. 7, p. research0034–1, 2002. Citado 2 vezes nas páginas 11 e 22.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, Elsevier, v. 2, n. 1-3, p. 37–52, 1987. Citado na página 33.

YAHAYA, S. W.; LANGENSIEPEN, C.; LOTFI, A. Anomaly detection in activities of daily living using one-class support vector machine. In: SPRINGER. *UK Workshop on Computational Intelligence*. [S.l.], 2018. p. 362–371. Citado na página 11.

ZHU, X.; LIU, Y.; LI, J.; WAN, T.; QIN, Z. Emotion classification with data augmentation using generative adversarial networks. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. [S.l.], 2018. p. 349–360. Citado na página 20.

ZHUL, R.; LI, G.; LIU, J.-X.; DAI, L.-Y.; YUAN, S.; GUO, Y. A fast quantum clustering approach for cancer gene clustering. In: IEEE. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. [S.l.], 2018. p. 1610–1613. Citado na página 15.