

UNIVERSIDADE FEDERAL DO PARANÁ

AMANDA WILCZEK

IDENTIFICAÇÃO *IN SILICO* DE SÍTIOS DE LIGAÇÃO À PROTEÍNA
REGULATÓRIA NTRC EM SEQUÊNCIAS GENÔMICAS

CURITIBA

2019

AMANDA WILCZEK

IDENTIFICAÇÃO *IN SILICO* DE SÍTIOS DE LIGAÇÃO À PROTEÍNA REGULATÓRIA NTRC
EM SEQUÊNCIAS GENÔMICAS

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Bioinformática, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, área de concentração Bioinformática.

Orientador: Profº Drº Roberto Tadeu Raittz

Co-orientador: Profº Drº Luciano Fernandes Huergo

CURITIBA

2019

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

W667i Wilczek, Amanda

Identificação in silico de sítios de ligação à proteína regulatória NTRC em sequências genômicas / Amanda Wilczek. – dados eletrônicos. – Curitiba, 2019.

1 arquivo (93 f.) : PDF.

Requisitos do Sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Orientador: Prof. Dr. Roberto Tadeu Raittz.

Co-orientador: Prof. Dr. Luciano Fernandes Huergo

Dissertação (mestrado) – Universidade Federal do Paraná, Setor de Educação Profissional e Tecnológica, Programa de Pós-graduação em Bioinformática.

1. Bioinformática. 2. Redes neurais artificiais. 3. Nitrogênio - Fixação. 4. Software - Desenvolvimento. I. Raittz, Roberto Tadeu. II. Huergo, Luciano Fernandes. III. Universidade Federal do Paraná. Programa de Pós-Graduação em Bioinformática. IV. Título.

CDD : 570.285

Bibliotecária: Thays Luciana Barbosa de Farias CRB-9/1995



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Pós-Graduação em Bioinformática WWW.BIOINFO.UFPR.BR
E-mail: bioinfo@ufpr.br Tel: 41 33614906

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em BIOINFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **AMANDA WILCZEK** intitulada: “**Identificação in silico de sítios de ligação à proteína regulatória NtrC em sequências genômicas**”, após terem inquirido a aluna e realizado a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 30 de maio de 2019.

Dr. Roberto Tadeu Raittz
Presidente/Programa de Pós-graduação em Bioinformática – UFPR

Dr. Vinicius Almir Weiss
Avaliador Externo/Bolsista Pos-Doc-Programa de Pós-graduação
em Microbiologia, Parasitologia e Patologia – UFPR)

Dr^a. Ana Claudia Bonatto
Avaliadora Externa/Programa de Pós-graduação em Genética-UFPR

AGRADECIMENTOS

Agradeço imensamente meu orientador Prof. Dr. Roberto Tadeu Raittz, por todos os ensinamentos passados desde meu tempo na iniciação científica (que, de repente parece tão distante..) e pela paciência que teve comigo. Ao meu co-orientador Prof. Dr. Luciano Fernandes Huergo, por sempre arranjar um tempo para responder minhas perguntas. Agradeço a maravilhosa equipe que tive o prazer de participar, pelos momentos dentro e fora do laboratório.

Também sou grata aos professores do programa de Pós-Graduação que se dedicam para transmitir conhecimento aos jovens pesquisadores. Agradeço a secretária Suzana pelas conversas nas tardes chuvosas e por todo o auxílio prestado de forma exemplar.

Agradeço a minha família, principalmente meus pais e meu irmão, por todo o apoio e carinho que me forneceram independentemente da ocasião. Aos meus amigos, que compreenderam meu afastamento em certos momentos, mas não desistiram de me convidar para sair.

Agradeço aos demais funcionários, alunos e egressos da Bioinformática, por terem feito parte dos meus dias nesse período tão importante.

Por fim agradeço a CAPES por fornecer minha bolsa de estudos, e ao Programa de Pós-Graduação em Bioinformática, que me concedeu a oportunidade de aprender, pesquisar, e ensinar. Certamente foi uma experiência intensa, mas muito satisfatória.

RESUMO

O nitrogênio é um elemento essencial para a manutenção da vida na Terra. Apesar disso sua maior concentração está presente na atmosfera. Algumas proteobactérias possuem o sistema Ntr, que é responsável pelo processo de regulação do metabolismo do nitrogênio. Dentro do Ntr, encontra-se o sistema NtrBC, que regula a expressão de genes envolvidos com a utilização de fontes alternativas de nitrogênio. Nele encontramos a proteína NtrC, que atua como um ativador de transcrição se ligando a sítios específicos no DNA e ativando promotores dependentes do fator sigma 54 (σ_{54}), tipicamente relacionados à transcrição de genes ligados ao metabolismo de nitrogênio. Os métodos mais comuns para detectar regiões de ligação da proteína NtrC ao DNA consiste em análises experimentais em laboratório, o que pode ser um processo caro e demorado. Para auxiliar nessa tarefa propomos uma ferramenta preditora de regiões relacionadas ao sítio de ligação da proteína NtrC a partir de um arquivo de genoma completo. A ferramenta contém uma rede neuronal artificial que passou pelo processo de treinamento supervisionado. Referente aos dados para o treinamento, utilizamos conjuntos de regiões promotoras de NtrC já confirmadas anteriormente e disponibilizadas em bancos de dados abertos para compor o conjunto de dados verdadeiros. Para compor o conjunto de regiões falsas utilizamos regiões geradas aleatoriamente, regiões retiradas de organismos modelo, e regiões provenientes de outros fatores de transcrição. A fim de selecionar qual é o melhor processo de extração de características e o modelo de rede neuronal mais adequado para solucionar o problema utilizamos janelas móvel e bases ortonormais de tamanhos variados. Esses conjuntos previamente classificados foram agrupados e embaralhados e passaram pelos modelos de classificadores MLP, SVM, RBF, DT, KNN, NB, RF (sendo os 3 primeiros utilizados no software MATLAB e o restante utilizando a biblioteca sklearn em Python 3), e por um modelo baseado em FAN com o software EasyFan. Após efetuar testes com arquivos de genoma da base de dados do NCBI e comparar com dados disponíveis em bancos de dados voltados à fatores de transcrição, a rede foi disponibilizada em uma ferramenta web para que possa ser utilizada pelo público.

Palavras-chave: NtrC, Redes Neurais Artificiais, TF, TFBS

ABSTRACT

Nitrogen is an essential element for the maintenance of life on Earth. However its greater concentration is present in the atmosphere. Some proteobacteria have the Ntr system, which is responsible for the regulation process of nitrogen metabolism. Within the Ntr, is the NtrBC system, which regulates the expression of genes involved with the use of alternative sources of nitrogen. In it we find the NtrC protein, which acts as a transcriptional activator binding to specific sites in the DNA and activating promoters dependent on the sigma factor 54 (σ_{54}), typically related to the transcription of genes linked to nitrogen metabolism. The most common methods for detecting binding regions of NtrC protein to DNA are experimental laboratory analyzes, which can be an expensive and time-consuming process. To assist in this task, we propose a predictor tool for regions related to the NtrC protein binding site from whole-genome. The tool contains an artificial neural network that has gone through the supervised training process. About the training data, we used sets of NtrC promoter regions previously confirmed and available in open databases to compose the true data set. To compose the set of false regions we use randomly generated regions, regions taken from model organisms, and regions from other transcription factors. In order to select which is the best feature extraction process and the most appropriate neural network model to solve the problem we use different and sliding windows and orthonormal bases. These previously classified sets were grouped and shuffled and went through the classification models MLP, SVM, RBF, DT, KNN, NB, and RF using the sklearn package (Python 3) and software MATLAB, and a FAN based model with EasyFan software. After testing NCBI database genomes and comparing it with data available in transcription factor databases, the network is available in a web tool so it could be used by the public

Keywords: NtrC, Artificial Neural Networks, Transcription Factor, Transcription Factor Binding Sites

LISTA DE SIGLAS E ABREVIATURAS

ANN – Redes Neurais Artificiais

CNN – Convolutional Neural Networks

DT – Decision Trees

FAN – Free Associative Neurons

KNN – K Nearest Neighbors

MLP – Multi Layer Perceptron

NB – Naive Bayes

PWM - Position Weight Matrix

RBF – Radial Basis Function

RF – Random Forest

SVM – Support Vector Machine

TF – Fator de Transcrição

TFBS – Sítios de Ligação a Fatores de Transcrição

LISTA DE FIGURAS

FIGURA 1 - REPRESENTAÇÃO DE MOTIF ATRAVÉS DE SEQUÊNCIAS LOGO.....	18
FIGURA 2 - WORKFLOW.....	23
FIGURA 3 - REPRESENTAÇÃO DO NEURÔNIO BIOLÓGICO E ARTIFICIAL.....	25
FIGURA 4 - REPRESENTAÇÃO DE UMA ÁRVORE DE DECISÃO.....	28
FIGURA 5 - INTERFACE DO EASYFAN.....	30
FIGURA 6 - REPRESENTAÇÃO DA CLASSIFICAÇÃO REALIZADA PELO KNN.....	31
FIGURA 7 - REDE MLP.....	32
FIGURA 8 - REDE RBF.....	34
FIGURA 9 - CLASSIFICAÇÃO RANDOM FOREST.....	35
FIGURA 10 - DIVISÃO DE HIPERPLANOS.....	36
FIGURA 11 - SOLUÇÕES COM SVM PARA A DIVISÃO DE DADOS NO HIPERPLANO.....	36
FIGURA 12 - REPRESENTAÇÃO ADOTADA PARA A MATRIZ DE CONFUSÃO.....	39
FIGURA 13 - MÉTRICAS DE AVALIAÇÃO.....	40
FIGURA 14 - ETAPAS DO DESENVOLVIMENTO.....	46
FIGURA 15 - COMPONENTES DO CONJUNTO DE DADOS.....	47
FIGURA 16 - REPRESENTAÇÃO VETORIAL BINÁRIA DA SEQUÊNCIA DE NUCLEOTÍDEOS	50
FIGURA 17 - PROJEÇÕES DO VETOR BINÁRIO.....	51
FIGURA 18 - EXEMPLO DE CONVERSÃO DE NUCLEOTÍDEOS.....	52
FIGURA 19 - SEQUÊNCIA CONSENSO DA NTRC.....	53
FIGURA 20 - SEQUÊNCIA ANTI-CONSENSO DA NTRC.....	53
FIGURA 21 - PROCESSOS DO NTRC FINDER.....	62
FIGURA 22 - VISUALIZAÇÃO DE TFBS ATRAVÉS DO ARTEMIS.....	63
FIGURA 23 - ANOTAÇÃO DAS INFORMAÇÕES DO TFBS EM ARQUIVO.....	64
FIGURA 24 - REDE REGULATÓRIA DOS GENES.....	71
FIGURA 25 - TELA INICIAL NTRC FINDER.....	73
FIGURA 26 - CONSULTA AO BANCO DE DADOS NTRC FINDER.....	74
FIGURA 27 - UPLOAD NO NTRC FINDER.....	75

LISTA DE TABELAS

TABELA 1 - CONVERSÃO DE NUCLEOTÍDEOS	52
TABELA 2 - AVALIAÇÃO DOS CLASSIFICADORES	57
TABELA 3 - TAXA DE RECUPERAÇÃO DE TFBS VERDADEIROS	60
TABELA 4 - SÍTIOS DE LIGAÇÃO À NTRC EM E. COLI MG1655 (NC_000913).....	66
TABELA 5 - ORGANISMOS E GENES ONDE OCORRE LIGAÇÃO À NTRC	83
TABELA 6 - FATORES DE TRANSCRIÇÃO (TF) UTILIZADOS NO CONJUNTO DE FALSO-NTRC.....	88
TABELA 7 - ORGANISMOS E GENES ONDE NÃO OCORRE LIGAÇÃO À NTRC.....	88
TABELA 8 - ANÁLISE DOS ENRIQUECIMENTOS ENCONTRADOS PELO STRING: PROCESSOS BIOLÓGICOS (GO).....	93

SUMÁRIO

1. INTRODUÇÃO.....	13
2. OBJETIVOS.....	14
2.1 OBJETIVOS ESPECÍFICOS.....	14
3. FUNDAMENTAÇÃO TEÓRICA	15
3.1 FATORES DE TRANSCRIÇÃO (TF)	15
3.2 UMA INTRODUÇÃO À PROTEÍNA NTRC: FUNÇÕES E CARACTERÍSTICAS	16
3.3 MÉTODOS ATUAIS DE PREDIÇÃO DE SÍTIOS DE LIGAÇÃO A FATORES DE TRANSCRIÇÃO (TFBS)	17
3.4 APRENDIZADO DE MÁQUINA	21
3.4.1 ALGORITMOS BASEADOS EM LÓGICA.....	24
3.4.2 ALGORITMOS BASEADOS EM MÉTODOS ESTATÍSTICOS	24
3.4.3 ALGORITMOS COM REDES NEURONAIAS ARTIFICIAIS.....	25
3.4.4 ALGORITMOS COM SVM	26
3.5 ARQUITETURAS DE CLASSIFICADORES	26
3.5.1 DECISION TREE (DT)	28
3.5.2 FREE ASSOCIATIVE NEURONS (FAN)	29
3.5.3 K-NEAREST NEIGHBOR (K-NN)	30
3.5.4 MULTILAYER PERCEPTRON (MLP).....	31
3.5.5 NAIVE BAYES (NB)	32
3.5.6 RADIAL BASIS FUNCTION (RBF).....	33
3.5.7 RANDOM FOREST (RF)	34
3.5.8 SUPPORT VECTOR MACHINE (SVM)	35
3.6 MÉTRICAS PARA AVALIAÇÃO DOS CLASSIFICADORES	38
3.7 Bancos de Dados.....	41
3.7.1 EcoCyc.....	41
3.7.2 Genome NCBI.....	41
3.7.3 RegPrecise	41
3.7.4 RegulonDB.....	41
4. MATERIAL E MÉTODOS.....	43
5. MODELO PROPOSTO	45
5.1 CONJUNTO DE DADOS PARA OBTENÇÃO DE PADRÕES	46
5.1.1 Sequências Para Padrões Corretos.....	47
5.1.2 Sequências Para Padrões Incorretos.....	47
5.1.3 Sequências Aleatórias Para Padrões Incorretos.....	48

5.1.4 Sequências Adicionais Para Padrões Incorretos	48
5.2 EXTRAÇÃO DE CARACTERÍSTICAS.....	49
5.2.1 Características 1 a 15	49
5.2.2 Características 16 a 32	52
5.2.3 Característica 33	52
5.2.4 Característica 34	53
5.2.5 Característica 35	54
5.3 DIVISÃO DOS CONJUNTOS	55
5.4 PARÂMETROS DOS MODELOS CLASSIFICADORES.....	55
6. RESULTADOS E DISCUSSÃO	56
6.1 COMPARAÇÃO DE MODELOS CLASSIFICADORES.....	56
6.2 NtrC FINDER	61
6.3 GBK2TABLE	63
6.4 ESTUDO DE CASO COM ESCHERICHIA COLI.....	64
6.5 FERRAMENTA WEB DO NtrC FINDER	73
7. CONCLUSÃO	76
8. REFERÊNCIAS BIBLIOGRÁFICAS.....	78
ANEXO I – CONJUNTO VERDADEIRO.....	83
ANEXO II – CONJUNTO FALSO.....	88
ANEXO III – ANÁLISE COM STRING	93

1. INTRODUÇÃO

Dentre os objetivos da bioinformática estão organizar dados de forma que permita aos pesquisadores criar e acessar informações, desenvolver ferramentas que facilitam a gestão de dados, e usar dados biológicos para analisar e interpretar os resultados de maneira biologicamente significativa (HAPUDENIYA, 2010).

No ramo da bioinformática existem métodos para prever onde proteínas se ligam no DNA. Dentre esses métodos destacam-se a utilização de PWM (matriz de peso), ANN (redes neuronais artificiais), e *Phylogenetic footprinting* (comparação entre espécies taxonomicamente próximas) (CHEN; KURGAN, 2012; GONZALEZ *et al.*, 2004).

Algoritmos preditores que utilizam esses métodos não funcionam bem para múltiplas regiões, visto que seu resultado depende do conjunto de dados de treinamento, seja para calcular o peso do nucleotídeo para cada posição (no caso da utilização de PWM), ou para a generalização do padrão dos nucleotídeos e suas exceções pela rede (no caso da utilização de ANN) (CARMACK *et al.*, 2007; OLIVER *et al.*, 2016; STORMO, 2000).

Os preditores atuais tentam englobar o maior número possível de sítios de ligação de diferentes fatores de transcrição, o que resulta em sítios falsos-positivos e falha ao detectar alguns sítios de ligação importantes (OLIVER *et al.*, 2016). Por isso propomos o desenvolvimento de uma ferramenta web que foque na predição de apenas uma proteína.

Escolhemos a proteína NtrC por ser uma proteína já reconhecida e estudada previamente por diversos autores. A proteína NtrC possui múltiplos sítios de ligação no DNA e ativa promotores dependentes do fator sigma 54 (σ_{54}), sendo essencial em sistemas que regulam o metabolismo de nitrogênio em bactérias (MERRICK; EDWARDS, 1995; TRENTINI, 2010). Como o nitrogênio é essencial para a célula, o sistema de adaptação à escassez de nitrogênio é um tema importante que vem sendo amplamente explorado nas últimas décadas.

Métodos da Inteligência Artificial oferecem uma abordagem poderosa e eficiente para resolver problemas básicos porém difíceis da bioinformática, por exemplo: alinhamento múltiplo de sequências, inferências filogenéticas, identificação de redes regulatórias de genes, predição de estrutura de proteína, entre outros. Dentre os métodos mais utilizados destacam-se Redes Neuronais Artificiais (ANN), Lógica difusa (*fuzzy*), e algoritmos genéticos (HAPUDENIYA, 2010).

Utilizamos redes neuronais artificiais para fazer um preditor de regiões relacionadas à proteína NtrC em genomas completos. Para isso foram feitos testes utilizando diferentes formas de extrair características e diferentes modelos de aprendizado de máquina (DT, FAN, KNN, MLP, NB, RBF, RF, e SVM) com a biblioteca sklearn (Python 3) e os *softwares* MATLAB, e EasyFan.

Ao final obtivemos uma rede com alta taxa de acerto que foi validada através da comparação dos resultados obtidos para a bactéria *Escherichia coli* com os dados disponíveis na base EcoCyc.

2. OBJETIVOS

O objetivo desse trabalho é utilizar redes neuronais artificiais para identificar corretamente regiões relacionadas à proteína NtrC dentro de genomas completos e com isso desenvolver uma ferramenta preditora de sítios relacionados à proteína NtrC.

2.1 OBJETIVOS ESPECÍFICOS

1. Selecionar informações verdadeiras e falsas para criar o conjunto de dados
2. Efetuar o treinamento dos modelos classificadores de diferentes arquiteturas
3. Testar, validar o resultado, e fazer alterações visando o aperfeiçoamento dos classificadores
4. Comparar o desempenho dos classificadores
5. Fazer estudo de caso com genomas fechados
6. Implementar uma ferramenta que receba o genoma completo através de um arquivo GenBank, extraia a sequência de DNA, e identifique possíveis regiões de ligação à NtrC utilizando o classificador que atingiu as melhores métricas
7. Disponibilizar a ferramenta

3. FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo apresentamos os conceitos e breves discussões dos temas pertinentes para a compreensão desse trabalho. O capítulo começa com uma introdução biológica a fatores de transcrição e visões gerais sobre a proteína NtrC, passa para uma explicação sobre os métodos atuais de algoritmos preditores de fatores de transcrição, e por modelos de aprendizado de máquina, incluindo redes neuronais artificiais que é o método proposto para a resolução do problema.

3.1 FATORES DE TRANSCRIÇÃO (TF)

A regulação da transcrição de genes como resposta a sinais extracelulares e intracelulares é um importante mecanismo para a adaptação bem-sucedida de microorganismos às mudanças das condições ambientais (NOVICHKOV *et al.*, 2013).

A regulação gênica ocorre no início da transcrição por Fatores de Transcrição (TF's) de ligação ao DNA que reconhecem uma ou mais regiões específicas próximo de promotores e resultam na ativação ou repressão da transcrição de genes próximos (OLIVER *et al.*, 2016).

Essas regiões específicas do DNA seguem um determinado padrão de nucleotídeos (GAO *et al.*, 2018), possuem de 6 a 20bp (MEHTA; SCHWAB; SENGUPTA, 2011), e são chamadas de sítios de ligação ao fator de transcrição (TFBS) (GAO *et al.*, 2018; NOVICHKOV *et al.*, 2013).

Esses sítios de ligação geralmente são pequenos e degenerados, ou seja, algumas posições podem ter múltiplas alternativas possíveis (WANG; ALHAMDOOSH; PEDRYCZ, 2016). Delinear essas posições específicas nas quais os TFs se ligam ao DNA é de grande importância para decifrar a regulação gênica no nível transcricional (MATHELIER; WASSERMAN, 2013).

Um grupo de operons regulados pelo mesmo TF forma um regulon. O regulon geralmente inclui genes de um subsistema celular em comum (RODIONOV, 2007). Todos os regulons juntos operados no mesmo genoma formam uma rede regulatória transcricional (TRN) de uma célula (NOVICHKOV *et al.*, 2013).

3.2 UMA INTRODUÇÃO À PROTEÍNA NTRC: FUNÇÕES E CARACTERÍSTICAS

O nitrogênio é um componente essencial para proteínas, ácidos nucleicos e parede celular da célula da bactéria (SWITZER; BROWN; WIGNESHWERARAJ, 2018). Na ausência de nitrogênio fixado, procariotos como *E. coli* param seu crescimento imediatamente (SANCHUKI *et al.*, 2017). O Ntr, sistema global de regulação do nitrogênio, é responsável pelo metabolismo de nitrogênio em diversas bactérias (MERRICK; EDWARDS, 1995).

Quando ocorre a falta de nitrogênio na bactéria, esse sistema procura por fontes alternativas de obtenção do elemento (SWITZER; BROWN; WIGNESHWERARAJ, 2018). Essas fontes podem ser nitrato, dinitrogênio, e uma variedade de aminoácidos e outros compostos orgânicos nitrogenados (LEIGH; DODSWORTH, 2007). A amônia é quase sempre a fonte preferida de nitrogênio para o crescimento bacteriano porque ela suporta ritmo de crescimento mais avançado do que qualquer outra fonte alternativa (MERRICK; EDWARDS, 1995). Em enterobactérias o principal regulador de transcrição de resposta à insuficiência de N é a proteína NtrC, do sistema de dois componentes NtrBC (BROWN *et al.*, 2014; SWITZER; BROWN; WIGNESHWERARAJ, 2018).

De acordo com o KEGG:

“Um sistema de dois componentes de tradução de sinal permite a bactéria detectar, responder e se adaptar a mudanças no ambiente ou em seu estado intracelular. Cada sistema de dois componentes consiste em um sensor de proteína-histidina kinase (HK) e um regulador de resposta (RR)” (KEGG Pathway, entry KO02020. 2018).

Nesse caso o sensor é a proteína NtrB e o regulador de resposta é a proteína NtrC. Sob condições de limitação de nitrogênio, NtrB catalisa a fosforilação e consequente ativação de seu regulador de resposta parceiro, NtrC (MERRICK; EDWARDS, 1995).

A proteína NtrC, por sua vez, ativa promotores dependentes do fator sigma 54 que estão relacionados à transcrição de genes ligados ao metabolismo do nitrogênio (MERRICK; EDWARDS, 1995; TRENTINI, 2010).

Vários genes ativados por NtrC codificam sistema de transporte de compostos nitrogenados (ZIMMER *et al.*, 2000). NtrC é responsável, por exemplo, por ativar a transcrição de genes que codificam a proteína NaC (*switch* para genes de adaptação a carência de N) e do gene *relA*, cujo produto é responsável pela síntese do maior nucleotídeo sinalizador de stress, durante a falta de aminoácidos (SWITZER; BROWN; WIGNESHWERARAJ, 2018).

A proteína NtrC é capaz de se ligar a sequências específicas do DNA caracterizadas pelo consenso TGCAC-N5-TGGTGCA (REITZER & MAGASANIG¹, 1985 citado por HUERGO, 2006).

Existem múltiplos locais de ligação à NtrC no DNA no genoma de bactérias (SHIAU *et al.*, 1992). Os métodos mais conhecidos para a detecção desses locais são a utilização de cepas mutantes para NtrC e/ou superexpressando a proteína NtrC acoplados as técnicas de análise de transcriptoma como DNA *microarrays* e RNAseq (BROWN *et al.*, 2014; ZIMMER *et al.*, 2000), análise de ChIP-seq, análise de PCR quantitativo em tempo real (BROWN *et al.*, 2014), entre outros. A predição de sítios de ligação entre DNA e proteínas como fatores de transcrição (TF) é uma tarefa útil devido ao fato de que “métodos experimentais (como o ChIP-seq) conseguem determinar os sítios de ligação verdadeiros de um tipo de proteína sob uma condição (tecido, célula, tratamento/doença, etc) por vez. É impossível fazer o perfil das combinações de todos os TF's e condições celulares experimentalmente” (XU *et al.*, 2018).

Com isso a predição computacional se tornou uma solução popular. Nela pode-se usar dados existentes para aprender as regras de ligação do TF e então atribuir o perfil de ligação sob uma nova condição biológica sem fazer os experimentos de fato (XU *et al.*, 2018). Técnicas computacionais visando auxiliar o processo de predição vem sendo desenvolvidas desde o início da década de 90. Essas técnicas são o tema do próximo tópico.

3.3 MÉTODOS ATUAIS DE PREDIÇÃO DE SÍTIOS DE LIGAÇÃO A FATORES DE TRANSCRIÇÃO (TFBS)

Atualmente uma das metodologias mais utilizadas para encontrar sequências de TFBS em bactérias consiste em selecionar regiões a montante de genes de interesse. O pesquisador então compara visualmente a região completa com a sequência consenso do TFBS e marca as possíveis regiões. Para auxiliar nessa tarefa e conferir se as regiões marcadas estão corretas podem ser utilizados ensaios experimentais diversos.

Métodos para auxiliar na detecção de TFBS vem sendo desenvolvidos há algumas décadas. Em seu artigo, Staden (1989) apresenta formas de calcular a probabilidade de encontrar padrões em sequências e, para isso, utiliza operadores lógicos e equações para efetuar os cálculos probabilísticos para 9 formas diferentes de definir *motifs*.

¹ Reitzer LJ, Magasanik B. **Expression of glnA in Escherichia coli is regulated at tandem promoters.** *Proc Natl Acad Sci U S A.* 1985;82(7):1979–1983. doi:10.1073/pnas.82.7.1979

Sequências *motif* são descritas como “sequências curtas, padrões recorrentes no DNA que presumidamente possuem uma função biológica, e muitas vezes indicam locais de ligação específicos para sequências de proteínas, como nucleases e fatores de transcrição (TF)” (D’HAESELEER, 2006).

Motifs são frequentemente representados em sequências *logo*, uma representação gráfica do padrão encontrado a partir do alinhamento múltiplo de N sequências. A *logo* consiste em uma pilha de letras para cada posição na sequência. A altura da pilha reflete o grau de conservação para essa posição (medida em *bits*) e a altura das letras em cada posição reflete a frequência relativa do aminoácido ou nucleotídeo correspondente nessa posição (CROOKS *et al.*, 2004).

Na Figura 1 (a) temos 10 sequências de nucleotídeo aleatórias com 20bp cada. Em (b) geramos a sequência *logo* utilizando a ferramenta WebLogo. Através dessa representação conseguimos fazer constatações mais facilmente: Pela altura da letra nota-se que os nucleotídeos nas posições 4, 5, 16, 17, e 18 são bem conservados, enquanto na posição 11, por exemplo, pode ocorrer os 4 nucleotídeos.

FIGURA 1 - REPRESENTAÇÃO DE MOTIF ATRAVÉS DE SEQUÊNCIAS LOGO

1. AACTGTATATAAATACAGTT
2. TATTGGCTGTTTATACAGTA
3. TCCTGTTAATCCATACAGCA
4. ACCTGTATAAATAACCGTA
5. TGCTGTATATACTCACAGCA
6. AACTGTATATACACCCAGGG
7. GACTGTATAAAACACAGCC
8. TACTGTATGAGCATAACAGTA
9. TACTGTATATAAAACAGTT
10. TACTGTACACAATAACAGTA

(a)



(b)

FONTE: A autora (2019).

LEGENDA: Em (a) Sequências aleatórias de 20bp cada. Em (b) Representação das sequências com WebLogo, evidenciando a frequência (ou o grau de conservação) dos nucleotídeos para cada posição.

O maior desafio do bioinformata é ter um bom conjunto de dados para analisar. Muitas vezes o tamanho pequeno de uma amostra ou o baixo grau de conservação impedem a construção de regras de reconhecimento confiáveis (MIRONOV *et al.*, 1999).

Assim apenas um número limitado de sítios de ligação são conhecidos para um TF, e, por essa razão, os algoritmos criados devem construir um classificador geral baseado nos dados limitados para treinamento (MEHTA; SCHWAB; SENGUPTA, 2011).

Segundo McCue e colaboradores (2002), mesmo em um grupo filogenético pequeno (como as gama-proteobactérias, por exemplo) existe alta variedade de espécies e por isso o tamanho do genoma, a distância filogenética e um habitat em comum (ou similar) são fatores que devem ser levados em consideração ao escolher as espécies que irão compor um estudo.

Ainda com pequena distância filogenética não é raro que existam diferenças significativas entre as bactérias. A bactéria *Haemophilus influenzae* é descrita como o único genoma bacteriano completo parecido com *Escherichia coli* (MIRONOV *et al.*, 1999). Mesmo assim após comparar os dois genomas verificou-se que apesar da similaridade, a *H. influenzae* não possui a subunidade de RNA polimerase sigma 54 e vias regulatórias relacionadas à NtrC, enquanto essas são encontradas em *E. coli* (TATUSOV *et al.*, 1996). Ou seja, apesar de que as duas bactérias sejam parecidas (pequena distância filogenética), a *E. coli* possui genes dependentes de NtrC e *H. influenzae* não. Em seu artigo, Tatusov e colaboradores (1996) consideram que esse é um indicativo de que *H. influenzae* é adaptada para crescer em ambientes ricos em nitrogênio e, por essa razão, não possui o sistema de resposta para a limitação do elemento.

A disponibilidade de dados de genoma completo de vários procariotos abriram a oportunidade de identificar prováveis TFBS pela comparação cruzada entre espécies (do inglês, *cross-species comparison* ou *phylogenetic footprinting*) sem a necessidade de identificar experimentalmente o mesmo conjunto de genes coregulados por um TF em comum entre duas espécies próximas (MCCUE, 2002).

De acordo com Gonzalez e colaboradores (2004), métodos computacionais para reconhecer TFBS em genomas bacterianos baseiam-se em PWM, na impressão filogenética (*phylogenetic footprint*), e em busca por “super-representação” estatística de oligo-nucleotídeos. Desses, o modelo de predição de TFBS mais conhecido e utilizado atualmente é o PWM, devido a sua simplicidade (KHAMIS *et al.*, 2018).

A matriz de posição de pesos (PWM, do inglês *Position Weight Matrix*) é o modelo mais comum para representar a preferência de ligação do TF ao DNA. Uma PWM é uma matriz de $4 \times k$ onde k é o comprimento do sítio de ligação e cada linha da matriz se refere a um nucleotídeo de DNA (A, C, G, T). As entradas da PWM representam a probabilidade do nucleotídeo de aparecer em cada posição e quanto maior o valor mais conservada é a base de DNA nessa

posição (WANG; ALHAMDOOSH; PEDRYCZ, 2016). Apesar de ser simples possui suas desvantagens, como a alta taxa de falso-positivos em seus resultados, isso porque “PWM convencionais não modelam dependências entre posições individuais entre os TFBS” (STORMO, 2000).

Desde a abordagem de Staden (1989) já se destaca a utilização de matrizes de posição de peso (PWM). Desde então outros métodos (probabilísticos ou não) começaram a se popularizar.

Como uma tentativa de melhorar o modelo foram criadas novas abordagens. Uma delas é utilizar k-mer em relações entre nucleotídeos (KEILWAGEN; GRAU, 2015). Também já foram criadas abordagens baseadas em bayesian networks, HMM, e *deep learning* em redes neuronais. Esses modelos são mais flexíveis, pois conseguem verificar a independência da posição do nucleotídeo (INUKAI; KOCK; BULYK, 2017).

De acordo com a revisão de Gao e colaboradores:

“Os algoritmos para procurar TFBS candidatos são divididos em duas categorias: Os index-based algoritmos, utilizam e constroem estruturas de índice como árvores e arrays de sufixos para acessar mais rapidamente as localizações dos candidatos na sequência, sendo uma opção custosa em relação ao tempo e espaço utilizado, porém com eficiência aprimorada para buscas.

O método online é considerado tradicional e consiste em ‘escanear’ uma sequência do início ao fim utilizando janela deslizante com o tamanho de acordo com o de motifs conhecidos de TF e apresenta os possíveis candidatos. Esse método também custa tempo” (GAO *et al.*, 2018).

Apesar da existência e da ampla utilização de programas desenvolvidos para encontrar TFBS, a acurácia alcançada por esses permanece baixa, sendo que não é raro que TFBS's importantes de um sistema regulatório não seja identificado pelo programa (OLIVER *et al.*, 2016).

Quando conhecemos sítios de ligação para um fator de transcrição em particular é possível construir um modelo de motif que pode ser usado para encontrar sítios de ligação adicionais (CARMACK *et al.*, 2007). O problema dessa abordagem é que quando a busca é feita em escala genômica, retorna também muitos sítios insignificantes que se encaixaram no motif.

Para tentar diminuir o número de sítios o habitual é restringir a área de busca na região a montante dos genes (FERREIRA *et al.*, 2018; GONZALEZ *et al.*, 2004) e utilizar abordagens que levam em conta a filogenia (CARMACK *et al.*, 2007; GONZALEZ *et al.*, 2004; NOVICHKOV *et al.*, 2013).

Para encontrar novos sítios de ligação (TFBS), Novichkov e colaboradores (2013) selecionam um fator de transcrição (TF) com sítios de ligação já conhecidos em um genoma e

buscam por sítios de ligação candidatos na região a montante (de -350 a +50 pb em relação ao códon inicial, excluindo as regiões codificantes dos genes a montante) em genes ortólogos.

Já Gonzalez e colaboradores (2004) optaram por fazer a predição para 17 genomas filogeneticamente próximos de *E. coli*. Em seguida dividiram os genomas em grupos conforme a porcentagem de genes ortólogos com *E. coli*, construíram matrizes de posição de peso (PWM) para TFs desses genes, e utilizaram dois modelos estatísticos (CONSENSUS e Gibbs-SAMPLER) para cada TF para encontrar TFBS nas regiões promotoras.

Em alguns casos pesquisadores utilizaram a conservação como um filtro adicional, e também já foram vistas abordagens usando dados de expressão de mRNA, rede bayesiana, regressão linear, entre outros (ERNST *et al.*, 2008).

No caso de eucariotos passaram a ser utilizadas abordagens com o classificador Random Forest para predizer genes alvo a partir dos dados de ligação de fatores de transcrição (ESSEBIER *et al.*, 2017).

Para otimizar tempo novas técnicas estão sendo integradas no método tradicional. Gao e colaboradores (2018) apresentam como exemplos de técnicas a utilização de “matrix partitioning”, “fast fourier transform”, “data compression”, entre outros. Porém o problema pode persistir, já que a maioria dos preditores se preocupa em cobrir todos os TF, obtendo um resultado razoável, onde se destacam a presença de mais falsos-positivos (OLIVER *et al.*, 2016).

3.4 APRENDIZADO DE MÁQUINA

Os algoritmos de aprendizado de máquina (do inglês, *machine learning*) usam dados de treinamento para descobrir padrões ocultos, construir modelos, e fazer predições baseadas no melhor modelo (MIN; LEE; YOON, 2016).

Existem dois tipos principais de aprendizado de máquina: O supervisionado, onde é construído o modelo a partir do aprendizado de dados com classe conhecida, e o não supervisionado, no qual os dados não possuem classe atribuída e o método aprende a separar os dados pelas suas características em comum (HUANG *et al.*, 2018).

Nesse trabalho utilizamos modelos de classificadores com treinamento supervisionado, ou seja, em que o resultado esperado já é conhecido e o modelo aprende a classificar novos dados a partir de dados previamente classificados utilizados no treinamento. Uma forma mais simples de conceituar o treinamento supervisionado é como se o classificador recebesse uma

prova e o gabarito já preenchido e a partir disso tivesse que deduzir a forma correta de resolver as questões da prova.

O algoritmo utilizado no aprendizado supervisionado deve ser capaz de raciocinar com instâncias fornecidas externamente para produzir hipóteses gerais e criar um modelo que faz previsões sobre novas instâncias (KOTSIANTIS, 2007).

Alguns algoritmos bem conhecidos, como SVM, Random Forests, HMM, Bayesian Networks, e Gaussian Networks, vem sendo aplicados na genômica, proteômica, biologia de sistemas, e outros domínios (MIN; LEE; YOON, 2016).

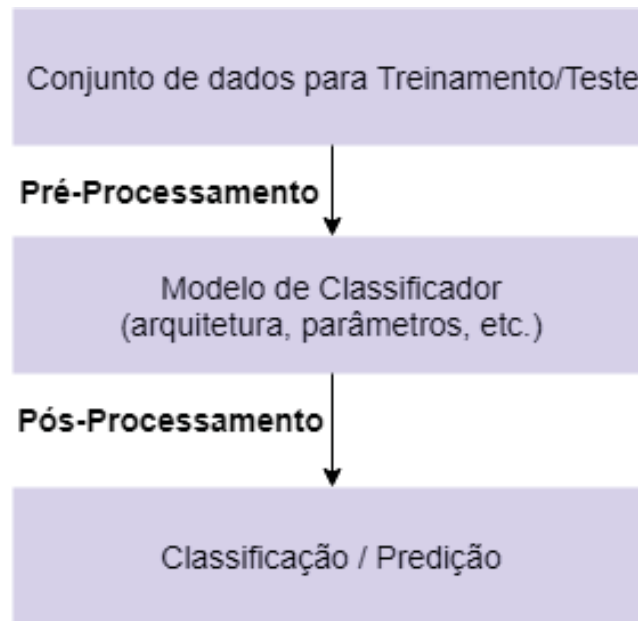
No que se refere a sua utilização em problemas relacionados à detecção de regiões específicas em genomas, podemos citar o uso de aprendizado de máquina em regiões específicas de leveduras (HOLLOWAY; KON; DELISI, 2007), regiões dependentes do fator $\sigma 70$ (FREIRE, 2014), predição de sítios de ligação ao fator $\sigma 54$ (FERREIRA *et al.*, 2018), predição de expressão gênica (SINGH, RITAMBHARA *et al.*, 2016), predição de sequências promotoras em micobactérias (KALATE *et al.*², 2003 citado por HAPUDENIYA, 2010), predição de sítios de *splicing* (REESE e EECKMAN³, 1995 citado por HAPUDENIYA, 2010), predição de regiões cis-regulatórias em genoma humano (LI, YIFENG; SHI; WASSERMAN, 2016), predição de sítios de ligação no DNA (CHEN; KURGAN, 2012), predição de ligação de proteína ao DNA (SHAO *et al.*, 2009; ZHANG, YAN-PING *et al.*, 2016; ZHANG, YANPING *et al.*, 2014), predição de interações regulatórias entre gene e fator de transcrição (ERNST *et al.*, 2008), entre outros. Nessas soluções as mais populares foram MLP, SVM, e CNN, mas outras arquiteturas também foram utilizadas.

Para utilizar uma solução de classificação supervisionada precisamos primeiro definir algumas etapas. A FIGURA 2 a seguir, adaptada de (HAPUDENIYA, 2010), apresenta o workflow das escolhas necessárias.

² Kalate RN, Tambe SS, Kulkarni BD. **Artificial Neural Networks For Prediction Of Mycobacterial Promoter Sequences**. *Comput Biol Chem*, 2003. 27(6): p. 555-64.

³ Reese MG, Eeckman FH. (1995) **Novel Neural Network Prediction Systems For Human Promoters And Splice Sites**. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.5678>

FIGURA 2 - WORKFLOW



FONTE: Adaptado de (HAPUDENIYA, 2010)

O Pré-processamento de dados envolve a representação de características e codificação de entrada (HAPUDENIYA, 2010). Falamos sobre o conjunto de dados utilizado e seu pré-processamento no tópico 5.1.

Dependendo da escolha do classificador há vários hiperparâmetros, número de camadas, número de camadas ocultas, valor de inicialização dos pesos, número de iterações, e até mesmo a taxa de aprendizado que são necessários definir, e cada um deles influencia notavelmente os resultados (MIN; LEE; YOON, 2016). Os parâmetros definidos são descritos no tópico 5.4.

Existem diversas técnicas de aprendizado que são usadas pelos algoritmos capazes de realizar a classificação supervisionada. Essas técnicas são divididas entre aprendizado baseado em lógica, baseado em estatística, utilizando redes neurais artificiais, e SVM (KOTSIANTIS, 2007).

Geralmente, SVMs e redes neurais tendem a ter um desempenho muito melhor ao lidar com multidimensões e características contínuas. Por outro lado, os sistemas baseados em lógica tendem a ter um desempenho melhor ao lidar com características categóricas. Para modelos de redes neurais e SVMs, é necessário um grande tamanho de amostra para atingir sua precisão máxima de previsão, enquanto o Naive Bayes precisa de um conjunto de dados relativamente pequeno (KOTSIANTIS, 2007).

Segundo Chen e Kurgan (2012), modelos de redes neurais artificiais são utilizados principalmente pela sua capacidade de predição, como a predição de estrutura de proteínas, e a predição de sítios de ligação e ligantes.

A seguir explicamos a definição das técnicas de aprendizado mais populares.

3.4.1 ALGORITMOS BASEADOS EM LÓGICA

O modelo lógico pode ser visualizado como um conjunto de regras "se-então" (MAXWELL; WARNER; FANG, 2018). O modelo é treinado para conseguir classificar uma nova instância a partir das regras criadas com base no conjunto de treinamento.

O mais conhecido é a árvore de decisão (DT) que utiliza dados categorizados e, uma vez que o modelo tenha sido desenvolvido, a classificação é extremamente rápida já que não é necessário efetuar cálculos matemáticos complexos (MAXWELL; WARNER; FANG, 2018).

3.4.2 ALGORITMOS BASEADOS EM MÉTODOS ESTATÍSTICOS

As abordagens estatísticas são caracterizadas por ter um modelo de probabilidade subjacente explícito, que fornece uma probabilidade de que uma instância pertença a cada classe ao invés de classificar a instância (KOTSIANTIS, 2007). Ou seja, são utilizados cálculos estatísticos para verificar as chances que a instância tem para ser atribuída a cada uma das classes.

O número de classificadores que utilizam métodos estatísticos é variado tanto na quantidade como na abordagem. Dentre eles temos desde classificadores que utilizam cálculos probabilísticos (como o Naive Bayes) até classificadores baseados em instâncias (como o k-NN) (KOTSIANTIS, 2007; MAXWELL; WARNER; FANG, 2018;).

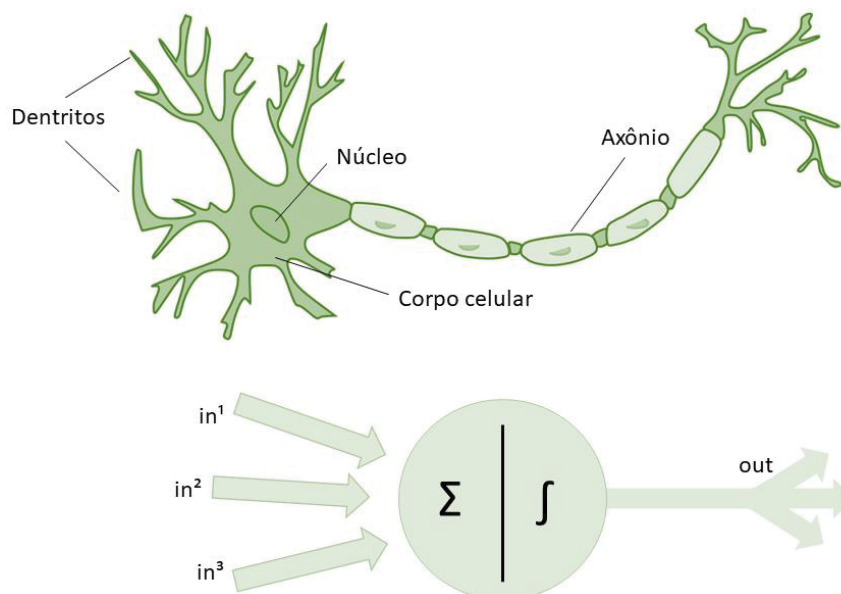
O aprendizado baseado em instância requer pouco tempo durante a fase de treinamento porém possui a etapa de classificação mais lenta do que outros modelos de aprendizado. Esses também são chamados de algoritmos de aprendizado preguiçoso, pois atrasam o processo de generalização até o momento de classificação (KOTSIANTIS, 2007).

3.4.3 ALGORITMOS COM REDES NEURONAIS ARTIFICIAIS

O conceito de rede neuronal é inspirado pela forma em que o cérebro recebe e processa informações (KUBAT, 1999), ou seja, aprende por meio de exemplos. Assim, redes neuronais artificiais (ou ANN, do inglês *Artificial Neural Networks*) reúnem conhecimento por meio da detecção de padrões e relacionamentos em conjuntos de dados que recebem como entrada (JHA, 2007) e aprendem (ou são treinadas) a partir da experiência e não da programação (AGATONOVIC-KUSTRIN; BERESFORD, 2000).

A forma que o neurônio artificial é apresentado se assemelha ao neurônio biológico (FIGURA 3): O neurônio tem um corpo celular, vários dentritos e um axônio. Os neurônios se ligam através da conexão entre dentritos e axônio. Dentritos recebem um sinal do neurônio anterior que funciona como a entrada. Essa entrada aumenta ou diminui os potenciais elétricos do corpo celular e, se atinge um limite, um pulso elétrico é enviado para o axônio. Esse pulso é chamado de resultado e será utilizado como entrada para o próximo neurônio (HAPUDENIYA, 2010).

FIGURA 3 - REPRESENTAÇÃO DO NEURÔNIO BIOLÓGICO E ARTIFICIAL



FONTE: Adaptado de Richárd (2018)

Em uma rede neuronal artificial temos várias unidades computacionais chamadas de neurônios que se conectam por links e cada link tem um peso associado a ele. O neurônio recebe dados pelos links de entrada (in), os pesos são somados (Σ) e uma função (\int) os transforma no

valor de saída do neurônio (out) (HAPUDENIYA, 2010). Existem várias funções de ativação. As mais conhecidas são step function, sigmoid/logistic function, e gaussian function.

Técnicas de ANN são consideradas eficientes devido ao seu processamento rápido e resultados satisfatórios. Além disso elas conseguem lidar com dados incompletos, com ruídos, e com problemas não-lineares, e, uma vez treinadas, conseguem fazer previsões rapidamente. (KALOGIROU, 2000). Por esses motivos são frequentemente usadas em problemas complexos, quando não há soluções computacionais mais leves (GHRITLAHRE; PRASAD, 2018).

Vale lembrar que existem bem mais arquiteturas de redes neuronais. CNN (Convolutional Neural Networks), por exemplo, é um modelo de deep learning que domina as áreas de reconhecimento de imagens, detecção de objetos, imagem *inpainting*, e super resolução (LI, YU *et al.*, 2019), assim cada arquitetura tem uma área de utilidade mas pode ser mais indicada conforme os tipos de dados e o processo de seleção de características que será usado.

Ao selecionar a arquitetura de rede neuronal também devemos prestar atenção aos parâmetros utilizados. Sobre o número de camadas deve-se levar em conta que quanto mais camadas, maior a complexidade o modelo terá. Se tiver camadas insuficientes, o modelo será muito simples e não conseguirá explicar a relação complexa entre a entrada e a saída, resultando em *underfitting*. Se ocorrer excesso, o modelo fica tão complexo que se torna sensível a ruídos nos dados e *overfitting* (LI, YU *et al.*, 2019).

3.4.4 ALGORITMOS COM SVM

O método SVM é baseado no conceito de maximizar a distância mínima do hiperplano até o ponto do dado mais próximo (SINGH, A; THAKUR; SHARMA, 2016). Explicamos o conceito de forma mais detalhada no tópico 3.5.7.

O classificador SVM é uma forma de aprendizado de máquina. Entretanto uma questão recorrente na área é a categoria em que o SVM se encontra: Enquanto alguns autores o definem como método estatístico (GIRALDI *et al.*, 2008; PENNSTATE, 2019) outros o consideram otimização pura (HOANG, 2017).

3.5 ARQUITETURAS DE CLASSIFICADORES

Como não há um consenso claro na literatura sobre o melhor algoritmo de aprendizado de máquina, o algoritmo ideal é provavelmente específico para cada caso, dependendo das

classes mapeadas, da natureza dos dados de treinamento e das variáveis preditoras (MAXWELL; WARNER; FANG, 2018).

Por essa razão utilizamos vários classificadores a fim de determinar qual modelo oferece a classificação ideal para a identificação de sítios de ligação. Seleccionamos arquiteturas que de classificadores supervisionados que utilizam diferentes abordagens de aprendizado de máquina. Essas arquiteturas são apresentadas a seguir.

3.5.1 DECISION TREE (DT)

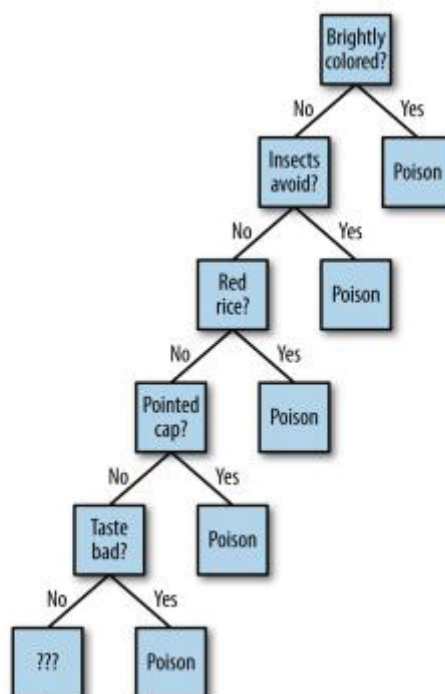
Árvores de decisão (do inglês Decision Tree, ou DT) são árvores que classificam instâncias ordenando-as com base nos valores das características. Cada nó em uma árvore de decisão representa uma característica de uma instância a ser classificada, e cada ramificação representa um valor que o nó pode assumir (KOTSIANTIS, 2007).

A analogia com árvore é usada para descrever o padrão geral de divisões repetidas, com ramos representando o caminho entre as divisões e folhas representando as classes. Por exemplo, o dado pode ser dividido dependendo se o valor em uma determinada banda está acima ou abaixo de um limite (MAXWELL; WARNER; FANG, 2018).

A característica que melhor divide os dados de treinamento será o nó raiz da árvore. Existem inúmeros métodos para encontrar a característica que melhor divide os dados de treinamento, como ganho de informação e índice de gini. (KOTSIANTIS, 2007).

O classificador com árvore de decisão é considerado o mais simples e intuitivo (MAXWELL; WARNER; FANG, 2018). Na figura a seguir temos uma representação de uma árvore de decisão.

FIGURA 4 - REPRESENTAÇÃO DE UMA ÁRVORE DE DECISÃO



FONTE: Kirk (2017)

LEGENDA: Árvore de decisão para classificação de cogumelos utilizando conhecimento popular. Esse modelo tem o objetivo de descobrir se um cogumelo é venenoso ou não a partir de suas características.

O modelo também possui suas desvantagens: Problemas com árvores de decisão incluem a possibilidade de gerar solução não ideal e *overfitting*. Uma alternativa para corrigir o *overfitting* é remover uma ou mais camadas de ramos. Esse processo é chamado de poda da árvore. A poda reduz a acurácia da classificação dos dados de treinamento enquanto aumenta a acurácia de dados desconhecidos (MAXWELL; WARNER; FANG, 2018).

3.5.2 FREE ASSOCIATIVE NEURONS (FAN)

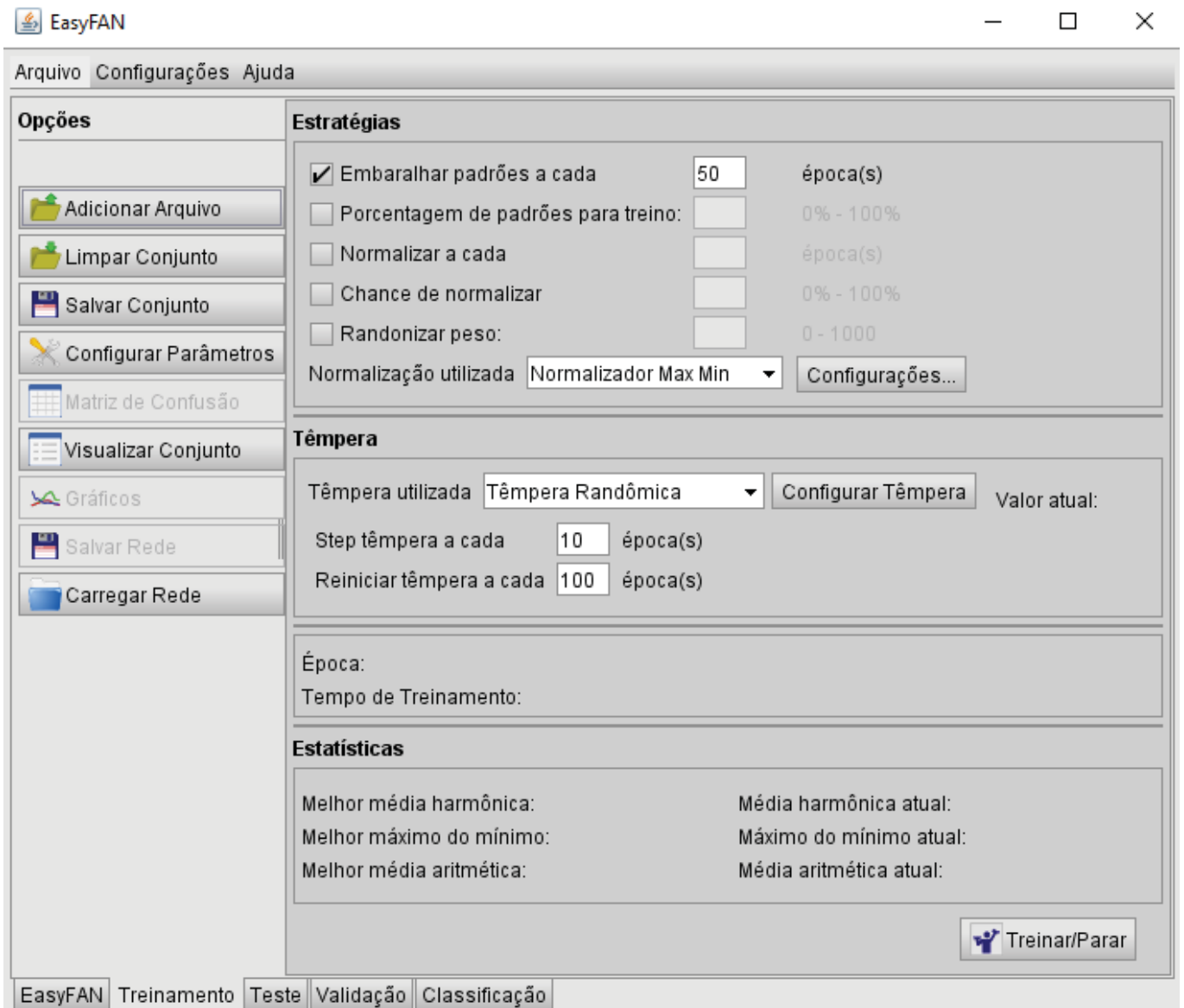
O modelo FAN é constituído por neurônios independentes com capacidade autônoma de aprendizado. O poder de aprendizado de FAN está baseado na granularidade na representação da informação (RAITZ, 1997). Na abordagem usando FAN, cada padrão de entrada do sistema é expandido em uma vizinhança nebulosa (COELHO; RAITZ; TREZUB, 2016).

Seu método é baseado em lógica difusa (do inglês *fuzzy*) e em noções de redes neuronais. Sendo um método para representação de ambientes complexos, FAN pode ser aplicado no reconhecimento de padrões, classificação e diagnóstico (RAITZ, 1997).

Redes FAN são utilizadas na detecção de fraudes em operações de comércio, onde atinge mais de 90% de precisão na classificação tanto de operações legais quanto fraudulentas (COELHO; RAITZ; TREZUB, 2016), para identificar sítios de ligação ao fator sigma 54 em genomas completos de bactérias (FERREIRA *et al.*, 2018), entre outros.

Nesse trabalho utilizamos o modelo FAN já implementado pelo EasyFan. O EasyFan (FIGURA 5) é um software desenvolvido em JAVA para reconhecimento de padrões voltado tanto para iniciantes na área de Inteligência Artificial quanto para pesquisadores mais experientes. EasyFan utiliza técnicas de FAN (Free Associative Neurons). O algoritmo original do modelo FAN foi aperfeiçoado na ferramenta e modelado conforme princípios da orientação a objetos (KUSTER *et al.*, 2016).

FIGURA 5 - INTERFACE DO EASYFAN



FONTE: A autora (2019)

LEGENDA: O *software* permite carregar redes já treinadas ou fazer um novo treinamento. É possível configurar os parâmetros de treinamento para a rede, além de selecionar os conjuntos de treinamento e testes através da barra na parte inferior da janela.

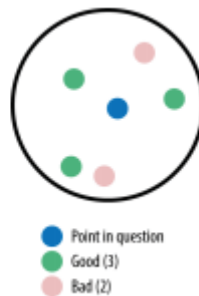
3.5.3 K-NEAREST NEIGHBOR (K-NN)

O classificador k-NN (k-ésimo vizinho mais próximo, do inglês *k-Nearest Neighbour*) é baseado no princípio de que instâncias em um conjunto de dados geralmente existirão muito próximas a outras instâncias que têm propriedades semelhantes. Se as instâncias estão classificadas, o valor da classe de uma instância não classificada pode ser determinado pela observação da classe de seus vizinhos mais próximos (KOTSIANTIS, 2007).

O k-NN é diferente dos outros classificadores: Ao invés de ser treinado para produzir um modelo classificador, esse algoritmo compara diretamente cada dado desconhecido com os dados originais de treinamento. O dado desconhecido é atribuído à classe mais comum dos k dados de treinamento que estão mais próximos do dado desconhecido. Um baixo k produz decisão complexa enquanto um alto k resulta em maior generalização. Como o modelo de treinamento não é produzido, é esperado que a classificação utilizando k-NN exija mais recursos à medida que o conjunto de treinamento aumenta (MAXWELL; WARNER; FANG, 2018).

A figura a seguir ilustra como é feita a classificação de uma instância desconhecida (em azul) a partir de seus vizinhos mais próximos.

FIGURA 6 - REPRESENTAÇÃO DA CLASSIFICAÇÃO REALIZADA PELO KNN



FONTE: Kirk (2017)

LEGENDA: Representação simples da classificação com $k=5$. Consideramos apenas as k instâncias mais próximas (vizinhas) do ponto em questão (azul). Contabilizamos o número de instâncias vizinhas para cada classe (bom e ruim). A classe com o maior número de instâncias vizinhas é atribuída ao ponto em questão. Nesse caso o ponto azul recebe a classe "bom" (verde).

A posição absoluta das instâncias no espaço não é tão significativa quanto a distância relativa entre instâncias. Essa distância relativa é determinada usando uma métrica de distância. Idealmente, a métrica de distância deve minimizar a distância entre duas instâncias classificadas da mesma forma, enquanto maximiza a distância entre instâncias de classes diferentes. Dentre as métricas mais conhecidas estão distância de Minkowski, distância de Manhattan, distância Euclidiana, distância de Chebyshev, e correlação de Kendall (KOTSIANTIS, 2007).

3.5.4 MULTILAYER PERCEPTRON (MLP)

Uma rede perceptron tem apenas duas camadas (entrada e saída) e é usada para classificar padrões em uma ou duas classes apenas. A rede MLP consiste em perceptrons com mais de duas camadas de neurônios, tendo uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída (HAPUDENIYA, 2010). Devido ao fato de ter múltiplas camadas não lineares alguns autores (LI, YIFENG; SHI; WASSERMAN, 2016; MIN; LEE; YOON, 2016)

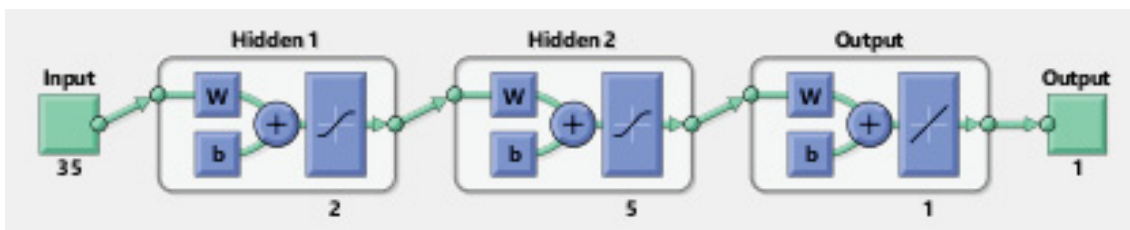
consideram MLP uma arquitetura de deep learning, entrando na categoria de Deep Neural Network (DNN).

Segundo o teorema de Kolmogorov-Nielsen, uma rede neural com três camadas, é um aproximador universal de funções contínuas e limitadas sobre um domínio compacto (KOVÁCS, 2002). Ou seja, uma rede MLP com apenas 3 camadas já consegue resolver problemas de classificação. Apesar disso, dependendo do problema a ser solucionado, a MLP permite que sejam trocadas a quantidade de neurônios de entrada, de saída, e de neurônios na camada oculta.

MLP é treinada de forma supervisionada e usa apenas dados já rotulados/classificados. Como o método de treinamento é um processo de otimização de hiper-planos, MLP é tipicamente usado quando um grande número de dados classificados está disponível (HAPUDENIYA, 2010; MIN; LEE; YOON, 2016).

A figura abaixo mostra o processo de treinamento de uma rede MLP no *software* MATLAB. A rede recebe 35 características através das suas duas camadas de entrada. As características passam por 5 camadas ocultas e o resultado do classificador é apresentado na camada de saída.

FIGURA 7 - REDE MLP



FONTE: A autora (2019)

O modelo MLP é considerado um dos mais populares para efetuar predições (H. KHALAFI; M., 2011). Dentre outras aplicações, redes MLP atualmente são aplicadas em pesquisas de estrutura de proteína, regulação da expressão gênica, classificação de proteína, e classificação de anomalias (MIN; LEE; YOON, 2016).

3.5.5 NAIVE BAYES (NB)

Naive Bayes (NB) é a forma mais simples de redes bayesianas, compostas de gráficos acíclicos direcionados com apenas um pai (representando o nó não observado) e vários filhos (correspondentes aos nós observados) com uma forte suposição de independência entre os nós filhos (KOTSIANTIS, 2007; SINGH, A; THAKUR; SHARMA, 2016).

Nesse modelo todos os atributos são independentes dado o valor da variável da classe. Isso é chamado de independência condicional. A suposição de independência condicional raramente é verdadeira na maioria das aplicações do mundo real (ZHANG, HARRY; ZHANG, 2004). Esse fato explica a nomenclatura do classificador, que significa “ingênuo” (do inglês, *naïve*).

A aprendizagem bayesiana utiliza abordagem probabilística. O algoritmo é baseado no teorema de Bayes: O modelo gera uma tabela de probabilidades para cada característica dos registros. Para um novo registro é calculada as possibilidades para todas as classes utilizando a probabilidade em cada registro. O registro novo é atribuído à classe com maior probabilidade (PEDREGOSA *et al.*, 2011).

Existem casos em que a probabilidade para um atributo é zero, pois os registros de treinamento não tem essa informação. Por ser feita a multiplicação dos valores basta um atributo nulo para que a probabilidade em uma característica se torne nula. Para evitar esse problema é feita a correção laplaciana (KOTSIANTIS, 2007).

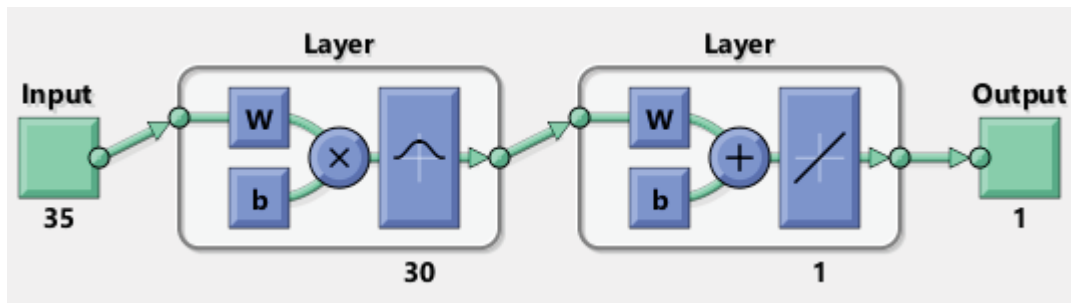
Ao contrário de redes neurais ou SVM, não há parâmetros livres a serem definidos o que simplifica a aplicação do classificador em uma ampla variedade de tarefas (KOTSIANTIS, 2007). Os classificadores Naive Bayes são famosos por serem utilizados na classificação de documentos e em filtros de *spam* (PEDREGOSA *et al.*, 2011).

3.5.6 RADIAL BASIS FUNCTION (RBF)

A arquitetura do RBF é parecida com a MLP, mas o princípio de ação e treinamento é diferente. A função RBF pode agrupar dados em números finitos de áreas elipsoides. As funções utilizadas por padrão são a Gaussian, spline function, ou várias funções quadráticas. Cada neurônio da rede age como o centro da região. Ao invés da soma de pesos da entrada é medida a distância (a mais comum é a euclidiana). O neurônio calcula a saída como uma função para o vetor de entrada e o seu centro. Tem 3 camadas: a de entrada, a oculta – com uma função de ativação não-linear -, e a camada linear de saída (CHEN; KURGAN, 2012).

A figura a seguir apresenta o treinamento de uma rede RBF utilizando MATLAB.

FIGURA 8 - REDE RBF



FONTE: A autora (2019)

LEGENDA: A rede recebe 35 características de entrada. Possui 30 centros e uma camada de saída

Esse modelo é tradicionalmente associado às funções radiais em redes com apenas uma camada (ORR, 1996). Em 2012, Chen e colaboradores verificaram que redes RBF foram utilizadas em aplicações como previsão de mapas de contato entre resíduos (Zhang & Huang, 2004) predição de locais de clivagem de protease (Yang e Thomson, 2005), predição de alvos para compostos direcionados à proteína (NIWA⁴, 2004 citado por CHEN e KURGAN, 2012).

3.5.7 RANDOM FOREST (RF)

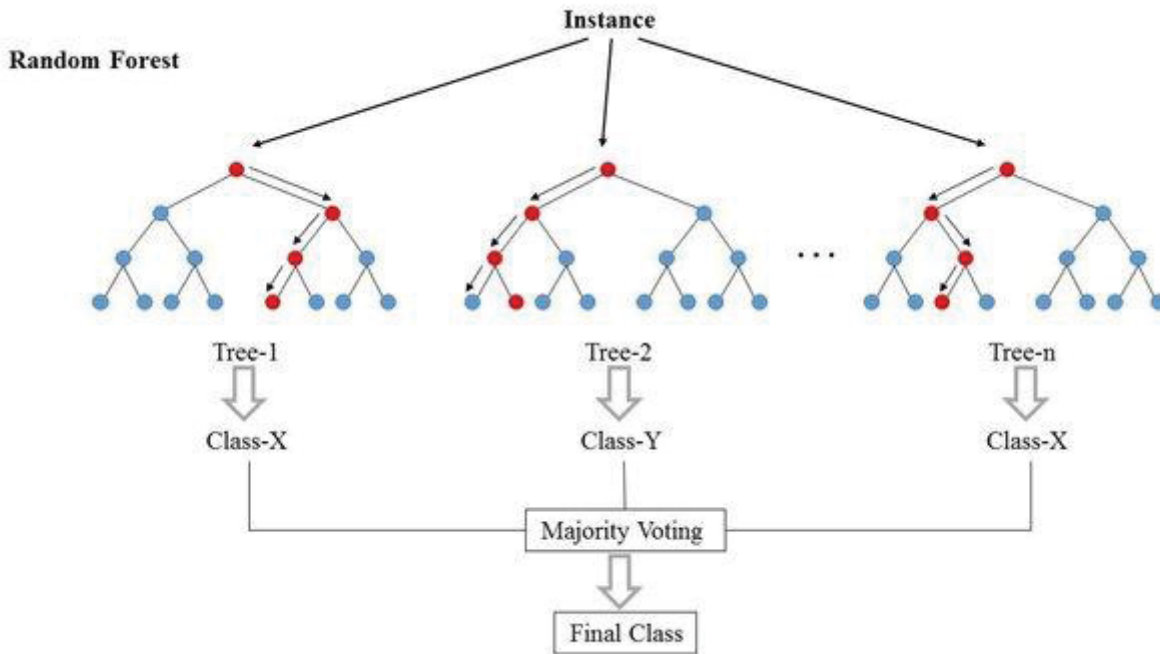
O RF (Floresta Aleatória, do inglês *Random Forest*) é um método de conjunto que opera treinando várias árvores de decisão (DT) e retornando a classe indicada pela maioria sobre todas as árvores do conjunto (KOTSIANTIS, 2007; SINGH, A; THAKUR; SHARMA, 2016).

A maioria dos "votos" de todas as árvores é usado para atribuir uma classe para cada dado desconhecido. Isso supera diretamente o problema de que qualquer uma das muitas árvores não é ótima, mas incorporando muitas árvores, deve-se obter uma ótima global. Essa ideia é estendida ainda mais, treinando cada árvore com seu próprio subconjunto gerado aleatoriamente dos dados de treinamento e também usando apenas um subconjunto das variáveis para essa árvore. A combinação de dados de treinamento reduzidos e número reduzido de variáveis significa que as árvores serão individualmente menos precisas, mas também serão menos correlacionadas, tornando o conjunto como um todo mais confiável (MAXWELL; WARNER; FANG, 2018).

A figura a seguir ilustra o processo de classificação Random Forest.

⁴ Niwa T (2004) Prediction of biological targets using probabilistic neural networks and atom-type descriptors. J Med Chem 47:2645–2650

FIGURA 9 - CLASSIFICAÇÃO RANDOM FOREST



FONTE: DIMITRIADIS; LIPARAS (2018)

LEGENDA: A nova instância passa por todas as árvores de decisão e recebe uma classificação em cada uma delas. Ao final a floresta calcula qual classe obteve mais votos e atribui a classe vencedora para a nova instância.

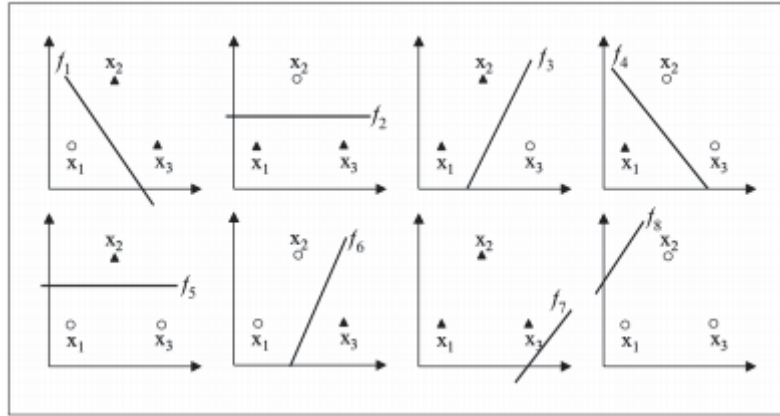
3.5.8 SUPPORT VECTOR MACHINE (SVM)

Aprendizado SVM (Máquinas de Vetores de Suporte, do inglês *Support Vector Machine*) é um de vários métodos de aprendizado de máquina. Comparado com outros métodos de aprendizado de máquina, SVM é muito poderoso no reconhecimento de padrões sutis em conjuntos de dados complexos (HUANG *et al.*, 2018).

Support Vector Machine (SVM) é um classificador originalmente criado visando a classificação binária (LIN & LIN, 2003). É um algoritmo complexo, que geralmente oferece alta acurácia. Também possui garantias teóricas que previnem o *overfitting* (SINGH, A; THAKUR; SHARMA, 2016).

O SVM representa os dados em um hiperplano e procura criar uma representação em que seja possível criar uma linha dividindo os dados dos dois rótulos, como na Figura 10.

FIGURA 10 - DIVISÃO DE HIPERPLANOS



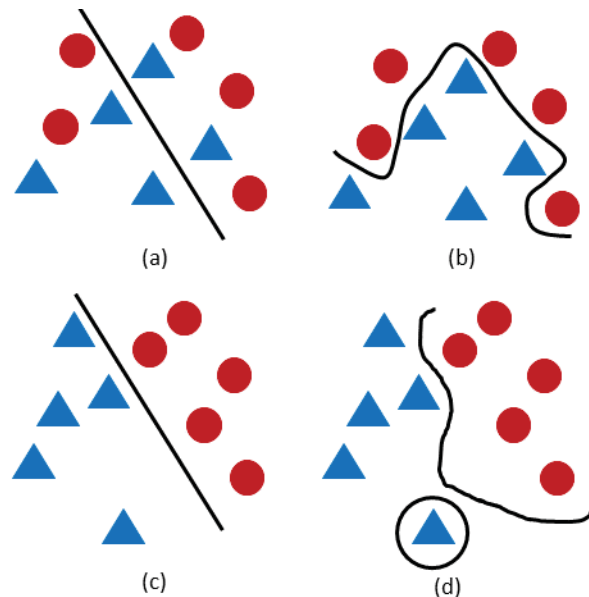
FONTE: LORENA et al. (2007)

LEGENDA: Divisão de hiperplanos para separar as instâncias em triângulo das instâncias em círculo

Como SVM utiliza métodos kernel, a escolha de diferentes funções kernel irá resultar em diferentes SVM com diferentes performances. Entre as funções kernel mais conhecidas estão linear, polynomial, radial basis function, e gaussian (ZANATY, 2012).

Dependendo do kernel escolhido a solução do problema será mais adequada conforme vemos na Figura 11, onde um kernel linear não consegue solucionar o problema de dividir as formas entre triângulos azuis e círculos vermelhos mostrado em (a), mas que é resolvido em (b) com um não linear. Acredita-se que a melhor performance seja utilizando kernel não linear, porém há casos em que os dados podem ser separados com uma única linha (c), além disso o método linear também é mais rápido em relação ao processamento.

FIGURA 11 - SOLUÇÕES COM SVM PARA A DIVISÃO DE DADOS NO HIPERPLANO



FONTE: A autora (2019)

Várias áreas da pesquisa utilizam de abordagens com SVM, como, por exemplo, reconhecimento de manuscritos (STUDHOLME; DIXON, 2003), reconhecer cartões de crédito fraudulentos, identificar voz, e detectar rostos (HUANG *et al.*, 2018). Já aplicações biológicas de SVM envolvem a classificação de proteínas, sequências de DNA, perfis de expressão de microarray, tipos de câncer, entre outros (HUANG *et al.*, 2018; NOBLE, 2006).

3.6 MÉTRICAS PARA AVALIAÇÃO DOS CLASSIFICADORES

O objetivo final dos modelos é generalização, ou seja, a habilidade de categorizar corretamente novas instâncias que diferem dos dados de treinamento (MEHTA; SCHWAB; SENGUPTA, 2011).

Uma preocupação especial ao usar classificadores poderosos, como métodos de aprendizado de máquina, é o *overfitting*. Isso ocorre quando o classificador mapeia os dados de treinamento de maneira tão precisa que não é capaz de generalizar bem (MAXWELL; WARNER; FANG, 2018). Ou seja, a rede pode “decorar” as respostas desejadas ao invés de aprender de fato e assim ela não conseguirá prever corretamente novos dados, mesmo que tenha atingido 100% de acerto no treinamento.

Por esse motivo apenas a classificação correta não é sinal de que o melhor modelo foi encontrado. É aconselhável efetuar a classificação com um conjunto de dados desconhecido pelo modelo (teste e validação) e utilizar métricas de avaliação de performance.

As métricas amplamente utilizadas na área são a matriz de confusão (GHRITLAHRE; PRASAD, 2018), a acurácia, a sensibilidade (*recall*), a precisão, e o F1-Score (MIN; LEE; YOON, 2016; ZHANG, YANPING *et al.*, 2014).

Algumas métricas utilizadas no aprendizado de máquina podem não funcionar para dados limitados e não balanceados, que é o caso dos dados da bioinformática. Enquanto a acurácia mostra com frequência resultados enganosos, o F1-Score, a média harmônica da precisão e *recall* fornece valores de performance mais compreensíveis. (MIN; LEE; YOON, 2016)

A matriz de confusão contabiliza as instâncias classificadas pelo modelo e as apresenta da seguinte forma:

- Verdadeiros Negativos (TN): instâncias falsas que foram classificadas como falsas
- Falsos Positivos (FP): instâncias falsas que foram classificadas como verdadeiras
- Falsos Negativos (FN): instâncias verdadeiras que foram classificadas como falsas
- Verdadeiros Positivos (TP): instâncias verdadeiras que foram classificadas como verdadeiras

A figura a seguir contém a representação adotada para a matriz de confusão:

FIGURA 12 - REPRESENTAÇÃO ADOTADA PARA A MATRIZ DE CONFUSÃO

$$\text{Matriz de Confusão} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

FONTE: A autora (2019)

LEGENDA: A matriz de confusão contém TN: Verdadeiros negativos; FP: Falso-positivos; FN: Falso-negativos; TP: Verdadeiros positivos;

A partir da matriz de confusão conseguimos calcular as métricas de avaliação. Seleccionamos as métricas de acurácia, precisão, recall, e F1-score.

A acurácia é simplesmente uma taxa de erro do modelo. O quanto ele classifica o conjunto (KIRK, 2017).

Precisão é uma medida de quão pontual é a classificação. A pergunta feita é “de todas as correspondências positivas encontradas pelo modelo, quantas estavam corretas?” (KIRK, 2017).

O recall pode ser considerado como a sensibilidade do modelo. É uma medida de se todas as instâncias relevantes foram realmente analisadas (KIRK, 2017). Se refere à quantidade de vezes em que o resultado esperado é verdadeiro e o modelo o classifica como verdadeiro.

O F1-Score combina precisão e *recall* de modo a trazer um número único que indique a qualidade geral do modelo.

As fórmulas gerais para o cálculo das métricas são apresentadas na figura a seguir.

FIGURA 13 - MÉTRICAS DE AVALIAÇÃO

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \times 100$$

$$Precisão = \frac{TP}{TP + FP} \times 100$$

$$Sensibilidade (Recall) = \frac{TP}{TP + FN} \times 100$$

$$F1 - Score = \frac{2 * Precisão * Recall}{Precisão + Recall} \times 100$$

LEGENDA: Fórmulas gerais para o cálculo das métricas

3.7 BANCOS DE DADOS

Segundo Chen e Kurgan (2012), nas últimas décadas houve o “rápido crescimento de bancos de dados biológicos que armazenam conteúdos relacionados à sequência de DNA e RNA, sequência e estrutura de proteínas, e outras estruturas macromoleculares”.

Atualmente existem muitos bancos de dados biológicos. Em nossas análises selecionamos bancos de dados públicos e consolidados com informações sobre sítios de ligação à proteína NtrC para efetuar o treinamento e os testes dos modelos classificadores. A seguir falaremos sobre os bancos de dados utilizados.

3.7.1 EcoCyc

A base de dados é descrita como:

“Uma base de dados científica para a bactéria *Escherichia coli* K-12 MG1655. O projeto EcoCyc realiza curadoria baseada na literatura de todo o genoma e da regulação transcricional, transportadores e vias metabólicas.”

EcoCyc(2017)

3.7.2 Genome NCBI

O Genome NCBI (2019) é um dos bancos de dados mais conhecidos na bioinformática. Esse organiza informações sobre genomas, incluindo sequências, mapas, cromossomos, montagens e anotações. O acesso dessas informações é feito de forma gratuita e o banco de dados é atualizado constantemente.

3.7.3 RegPrecise

O RegPrecise é um banco de dados para captura, visualização e análise de regulons de diversos fatores de transcrição.

Ela é descrita como um recurso da web para coleta, visualização e análise de regulons transcricionais reconstruídos por genômica comparativa, contendo uma coleção de regulons de referência com curadoria manual. A base ainda fornece acesso a interações regulatórias inferidas organizadas por propriedades filogenéticas, estruturais e funcionais (NOVICHKOV *et al.*, 2013).

3.7.4 RegulonDB

Segundo o site oficial (2019), o RegulonDB é um “banco de dados sobre regulação transcricional no *Escherichia coli* K-12, contendo conhecimento curado manualmente a partir de

publicações científicas originais, complementado com conjuntos de dados de alto rendimento e previsões computacionais abrangentes”.

RegulonDB é uma base de dados gratuita que permite a visualização e o *download* de conjuntos de dados curados experimentalmente e previsões computacionais.

4. MATERIAL E MÉTODOS

Obtivemos da base de dados curada do RegPrecise, disponível através do link <http://regprecise.lbl.gov/RegPrecise/>, sítios de ligação a fatores de transcrição (TFBS) para compor o conjunto de dados utilizado.

A partir da página web do Genome NCBI, cujo acesso é feito através do link <http://www.ncbi.nlm.nih.gov/genome/>, fizemos o *download* dos genomas de referência utilizados durante as etapas.

Utilizamos o software EasyFan para a implementação da rede FAN. O programa foi desenvolvido de forma a facilitar o treinamento e usabilidade de redes FAN e está disponível para download através do link <http://sourceforge.net/projects/easyfan/>.

Utilizamos o MATLAB na versão R2012b para fazer as implementações das redes MLP, SVM e RBF e para desenvolver o NtrC Finder. Posteriormente refizemos as análises utilizando a versão R2018a para verificar se os algoritmos classificadores permanecem com a mesma implementação nas versões mais recentes.

Utilizamos o ambiente de desenvolvimento (IDE) Spyder 3.3.3 com a linguagem Python 3.7 e a biblioteca *sklearn* para fazer as implementações dos métodos de aprendizagem Decision Tree, K-Nearest Neighbor, Naive Bayes, e Random Forest. Todos os pacotes foram instalados automaticamente com Anaconda Navigator, uma interface gráfica para gerenciar distribuições de Python, módulos, e ambientes de trabalho, que pode ser baixada através do link <http://docs.anaconda.com/anaconda/navigator/>.

Coletamos do RegulonDB, disponível através do link <http://regulondb.ccg.unam.mx/>, todos os sítios de ligação à NtrC encontrados para *Escherichia coli* k-12 para fazer a comparação de sítios encontrados pelos diferentes classificadores implementados.

Utilizamos a linguagem de programação Python 3.6 e a biblioteca BioPython no desenvolvimento da ferramenta GBK2TABLE para disponibilizar o resultado do NtrC Finder em formatos adicionais. O Python é gratuito e disponível para download para todos os sistemas operacionais através do link <http://www.python.org/downloads/>. A biblioteca BioPython possui ferramentas para biologia computacional e seu site oficial (<http://biopython.org/>) contém o link para download da biblioteca, além de tutoriais.

Para o desenvolvimento da ferramenta web utilizamos a linguagem de programação PHP e o *framework* Foundation 6 pela sua capacidade de facilitar a criação de páginas web

responsivas e agradáveis ao usuário. A página oficial do Foundation (<http://foundation.zurb.com/>) oferece tutoriais para instalação, utilização e até alguns componentes prontos.

Para disponibilizar o GBK2TABLE utilizamos o *framework* Flask no desenvolvimento do webserver e o Bootstrap 4 para customização da interface. A documentação do Flask está disponível em <http://flask.palletsprojects.com/en/1.1.x/installation/> e a página oficial do Bootstrap é <http://getbootstrap.com/>.

Para a análise do resultado obtido pelo NtrC Finder com o genoma *Escherichia coli* str. K-12 substr. MG1655 utilizamos a base de dados EcoCyc e a ferramenta Sigma54 Finder . A base EcoCyc, disponível através do link <https://ecocyc.org/>, contém curadoria baseada na literatura de todo o genoma de *Escherichia coli* K-12 MG1655, incluindo regulação transcricional, transportadores e vias metabólicas. A ferramenta web Sigma54 Finder identifica sítios de ligação ao sigma 54 a partir de um arquivo GenBank (FERREIRA *et al.*, 2018) e está disponível através do endereço <http://200.236.3.16/s54.php>.

Também fizemos análises referentes a interações entre genes com a ferramenta STRING versão 11, disponível através do link <https://string-db.org/>, que oferece a visualização da rede regulatória entre um grupo de proteínas.

Para a visualização dos genomas completos utilizamos o *software* Artemis, uma ferramenta gratuita para anotação de genomas que permite a visualização de sequências (CARVER *et al.*, 2012). A página oficial da ferramenta é <https://www.sanger.ac.uk/science/tools/artemis>.

5. MODELO PROPOSTO

A identificação experimental dos sítios de ligação aos fatores transcrição (TFBS) em um genoma é um trabalho complexo e geralmente de alto custo. Nosso método automatizado propõe a utilização de redes neurais para focar apenas em um fator de transcrição e seus múltiplos sítios de ligação ao longo do DNA.

A FIGURA 14 apresenta as etapas necessárias para o desenvolvimento do projeto. Primeiramente recuperamos sequências de TFBS de bancos de dados públicos. Em seguida iniciamos o pré processamento, que inclui as etapas de extração de características das sequências, a divisão das sequências em dois conjuntos, e a escolha dos classificadores e seus parâmetros.

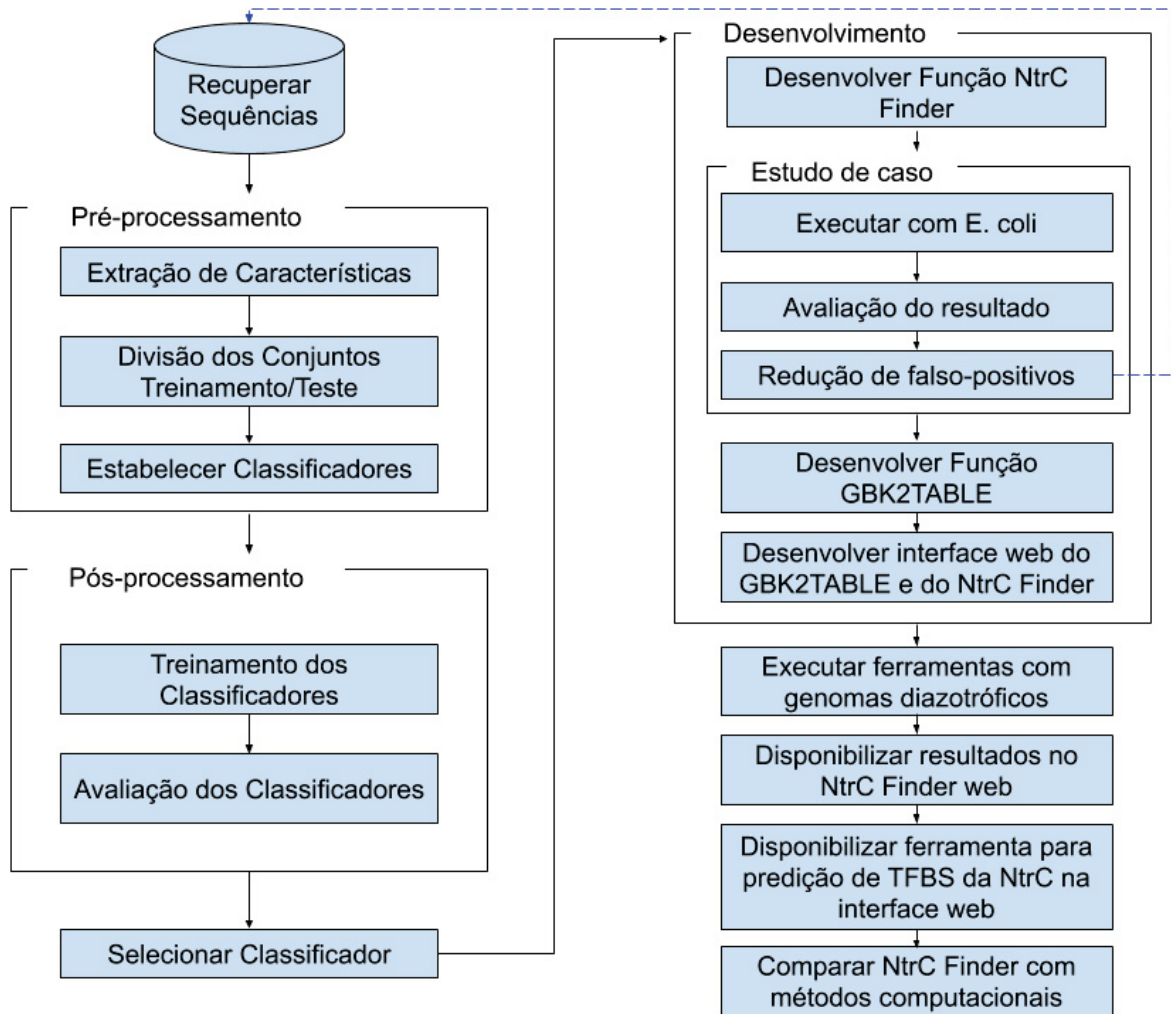
No pós-processamento efetuamos o treinamento dos classificadores, testamos os classificadores treinados e os avaliamos utilizando métricas reconhecidas. Selecionamos o classificador que obteve as melhores métricas para ser utilizado no desenvolvimento da função NtrC Finder.

Com a ferramenta pronta, fizemos um estudo de caso utilizando o genoma completo de *Escherichia coli* str. K-12 substr. MG1655 e ao avaliar o resultado verificamos a alta quantidade de falso-positivos. Desenvolvemos uma técnica para reduzir os falso-positivos que consiste em utilizar sequências de DNA selecionadas aleatoriamente de três genomas bacterianos. Essas novas sequências foram adicionadas ao conjunto de dados inicial (representado pela seta tracejada azul) e todos os processos foram refeitos. Quando verificamos que o número de sítios encontrados estava aceitável, demos continuidade nas etapas. Ao executar novamente o estudo de caso analisamos os resultados obtidos pela rede classificadora para o genoma de *E. coli* e comparamos com dados da literatura e bancos de dados especializados.

Desenvolvemos o GBK2TABLE, uma função adicional que recebe a anotação gerada pela ferramenta e cria uma tabela. Após a etapa de desenvolvimento do GBK2TABLE e sua interface web, executamos a ferramenta NtrC Finder com genomas diazotróficos, utilizamos o GBK2TABLE para padronizar as anotações, e disponibilizamos os resultados no banco de dados do sistema. Disponibilizamos as ferramentas de predição NtrC Finder e de conversão GBK2TABLE na interface web.

Por fim, comparamos a performance do NtrC Finder com outros métodos computacionais de predição.

FIGURA 14 - ETAPAS DO DESENVOLVIMENTO



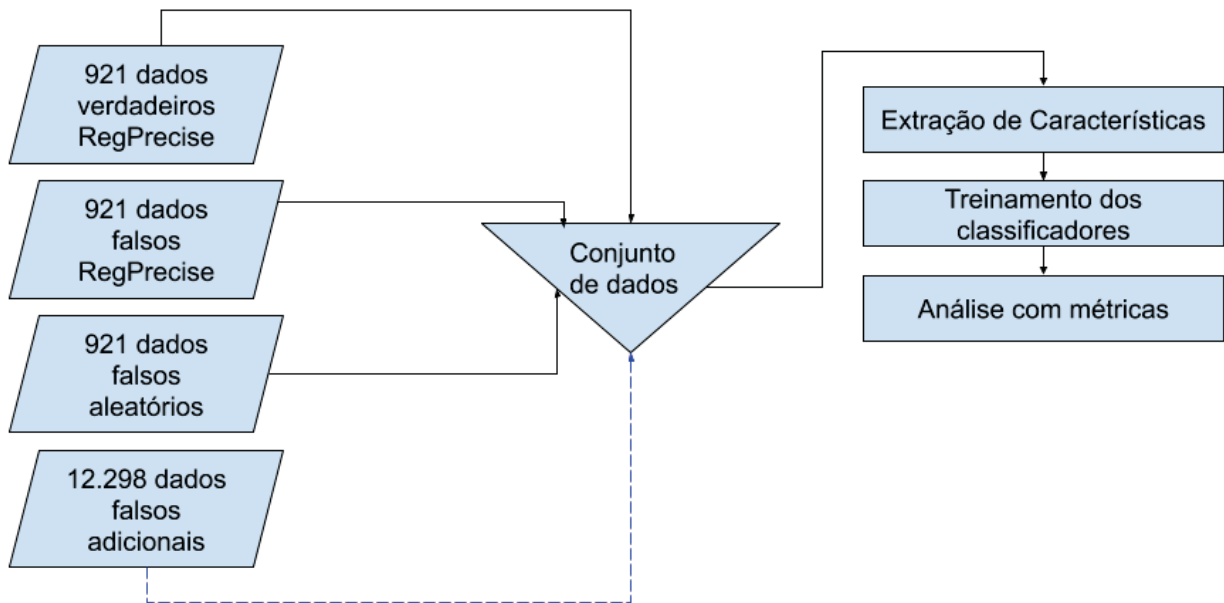
FONTE: A autora (2019)

5.1 CONJUNTO DE DADOS PARA OBTENÇÃO DE PADRÕES

A qualidade dos dados interfere na qualidade do resultado. Visando obter a maior quantidade de dados com boa qualidade optamos por utilizar a base de dados curada RegPrecise.

A partir do RegPrecise obtivemos sequências utilizadas para “integrar” o conjunto de dados utilizados para o treinamento e teste dos classificadores. Isso é contextualizado na Figura 15 e explicado detalhadamente nos subtópicos a seguir.

FIGURA 15 - COMPONENTES DO CONJUNTO DE DADOS



FONTE: A autora (2019)

LEGENDA: Os dados, provenientes de diferentes fontes e representados por paralelogramas, foram agrupados em um único conjunto de dados, representado pelo triângulo no centro da figura. Esse conjunto de dados foi utilizado nas etapas seguintes.

5.1.1 Sequências Para Padrões Corretos

Depois de uma interação inicial percebemos que entre os dados disponibilizados na base encontra-se regulons cujo o fator de transcrição é a proteína NtrC. Selecionamos do RegPrecise todos os regulons cujo fator de transcrição é o NtrC o que contabilizou 921 sítios de ligação em 169 genomas diferentes e salvamos esses dados em um arquivo multifasta. A lista completa dos genomas e dos genes regulados por NtrC está disponível no Anexo I.

5.1.2 Sequências Para Padrões Incorretos

A fim de constituir o conjunto de dados classificados como falsos (ou seja, a região do sítio de ligação não remete a um sítio de ligação à NtrC), fizemos uma nova consulta no RegPrecise procurando por regulons de outros fatores de transcrição, mas cujo o sítio de ligação tenha o mesmo tamanho (17bp). Também foi dada preferência para fatores de transcrição que atuam nas mesmas bactérias que o NtrC.

Selecionamos 921 sequências dos fatores de transcrição FadR, LldR, MetR, NagQ, GguR, GlcC, Fur, HexR, e FixJ. A lista de genomas utilizados, genes, e uma breve descrição dos TFs está disponível no Anexo II.

5.1.3 Sequências Aleatórias Para Padrões Incorretos

Geramos 921 sítios de ligação aleatórios. Esse processo, utilizado anteriormente por Ferreira e colaboradores (2018), consiste em criar um arquivo multifasta com sequências de 17 caracteres que representam os nucleotídeos (A, T, C, e G) através do software MATLAB. Com isso visamos incluir regiões sem representatividade biológica real classificados como incorretos no conjunto de dados

5.1.4 Sequências Adicionais Para Padrões Incorretos

Após treinar as redes com os primeiros 3 conjuntos de dados e fazer o teste com genoma completo de *E. coli* observamos que o modelo retornou 6720 sítios de ligação, o que já era esperado após a revisão de literatura sobre o histórico de altos falso-positivos para preditores de motifs.

Optamos por desenvolver um novo conjunto de sequências incorretas porém provenientes de genomas reais. A partir dos genomas de *Escherichia coli str. K-12 substr. MG1655*, *Pseudomonas aeruginosa PAO1*, e *Vibrio cholerae str. N16961* retiramos pedaços de sequências de dentro de genes, sequências com direção oposta de leitura, regiões distantes de coding sequences (CDs), e sequências de anotações anteriores incorretas e conseguimos 12.298 sequências. Esse conjunto adicional foi classificado como falso e “integrado” no conjunto de sequências, então todos os processos posteriores foram refeitos.

Com essa abordagem procuramos enriquecer o conjunto para treinamento e melhorar a sensibilidade dos classificadores: Como o classificador terá de lidar com uma grande quantidade de sequências para classificar, aumentamos notavelmente o conjunto de sequências falsas, a fim de fazer com que a rede aprenda as várias aparências de sequências incorretas (uma sequencia aparentemente correta pode não ser um sítio verdadeiro por diversos motivos).

Antes da inclusão do conjunto adicional o NtrC Finder identificava 6270 TFBS para *E. coli*. Após a integração o número de TFBS encontrado caiu para 112 sequências.

5.2 EXTRAÇÃO DE CARACTERÍSTICAS

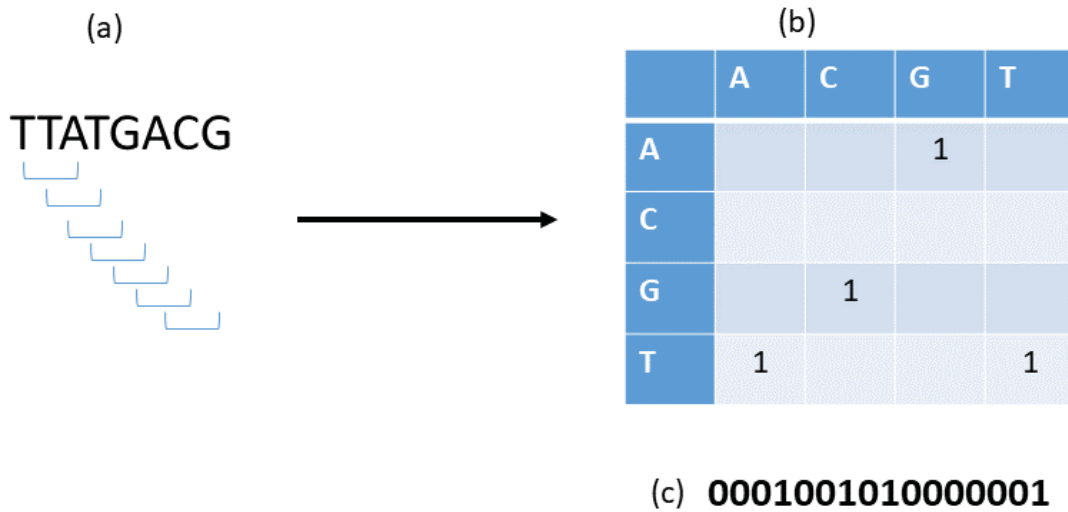
A performance dos algoritmos de aprendizado de máquina em geral depende fortemente de representações dos dados, chamadas de características. Essas características são definidas pelo pesquisador, que utiliza todo o seu conhecimento para traçar quais características são mais apropriadas (MIN; LEE; YOON, 2016).

Nesse tópico apresentamos como foram obtidas as 35 características utilizadas como entrada para as redes.

5.2.1 Características 1 a 15

Para testar combinações entre os nucleotídeos de cada sequência adotamos uma estratégia de representação vetorial utilizada por De Pierri e colaboradores (2020) no desenvolvimento da ferramenta SWeeP, cujo objetivo é representar conjuntos de dados de grandes seqüências biológicas em vetores compactos. A estratégia se baseia em *k-mers* espaçados (BODEN *et al.*, 2013) com janela deslizante cujo tamanho é de 3 nucleotídeos com 1 descartado. Assim combinamos o primeiro e o terceiro nucleotídeo e ignoramos o segundo nucleotídeo (do meio). Resta então uma lista de N pares de nucleotídeos. É criada uma matriz 4x4 para registrar cada possibilidade de combinação de bases. Marcamos 1 bit nas posições referentes a ocorrência de cada par de nucleotídeos. As posições onde não houve combinação permanecem desligadas (bit 0). Por fim a matriz é linearizada, dando origem a um vetor binário de tamanho 16. Esse processo é resumido na Figura 16 porém explicado integralmente por De Pierri e colaboradores (2020).

FIGURA 16 - REPRESENTAÇÃO VETORIAL BINÁRIA DA SEQUÊNCIA DE NUCLEOTÍDEOS



FONTE: A autora (2019)

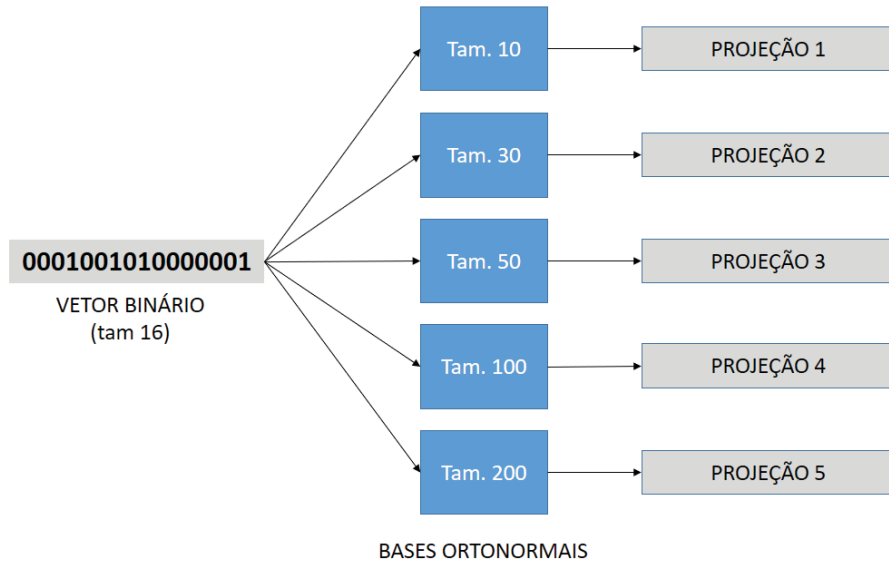
LEGENDA: (a) A sequência é dividida para cada 3 nucleotídeos. Para formar os pares de nucleotídeos utiliza-se o primeiro e o terceiro nucleotídeo e ignora-se o segundo. Esse processo é feito até o fim da sequência. (b) Os pares de nucleotídeos tem sua ocorrência marcada na matriz binária (c) A matriz é linearizada por colunas. As posições vazias da matriz são consideradas 0.

A segunda etapa dessa abordagem também é baseada no trabalho de De Pierri e colaboradores (2020). Enquanto a abordagem do SWeeP projeta a matriz binária em uma base ortonormal para diminuir o tamanho da representação dos aminoácidos sem perder informações, a nossa abordagem consiste em projetar o vetor binário em uma base ortonormal para modificar a projeção dos atributos a fim de melhorá-los.

Sendo assim o vetor binário obtido durante a primeira etapa é projetado em uma base ortonormal. Fizemos testes utilizando bases de diversos tamanhos.

Esse processo, caracterizado na Figura 17, foi feito da seguinte forma: Geramos bases ortonormais com dimensões de tamanho 10, 30, 50, 100, e 200, e projetamos o vetor binário para essas bases.

FIGURA 17 - PROJEÇÕES DO VETOR BINÁRIO



FONTE: A autora (2019)

Testamos o comportamento dos modelos classificadores com todas as projeções e escolhemos a base ortonormal de tamanho 30, pois essa se manteve consistente e forneceu resultados bons em todos os modelos testados. A base ortonormal de tamanho 30 resulta em uma matriz com 16 linhas e 15 colunas. A projeção do vetor binário na base de tamanho 30 resulta em um vetor com 15 colunas. Essas 15 colunas são estabelecidas como as 15 primeiras características da sequência.

5.2.2 Características 16 a 32

Para obter mais características que representem as sequências candidatas utilizamos a estratégia apresentada por Ferreira e colaboradores (2018). Essa consiste em utilizar como características para cada sequência candidata: sua conversão para números, seu alinhamento com a sequência consenso, e seu alinhamento com a sequência anticonsenso. A seguir explicamos como foi feita a extração de cada uma dessas características.

Codificamos a sequência dos 17 nucleotídeos para números através da função `dna2num` que recebe os nucleotídeos e converte cada um deles em um número conforme a tabela a seguir.

TABELA 1 - CONVERSÃO DE NUCLEOTÍDEOS

Nucleotídeo	Número
A	0
C	1
G	2
T	3

Assim, conforme a FIGURA 18 mostra, uma sequência qualquer “AGTCTAAGC” em (A) se torna uma sequência de números (B). Com esse processo o conjunto de 17 nucleotídeos passam a ser as novas 17 características.

FIGURA 18 - EXEMPLO DE CONVERSÃO DE NUCLEOTÍDEOS

AGTCTAAGC  023130021
(A) (B)

FONTE: A autora (2019)

LEGENDA: Conversão de uma sequência aleatória de nucleotídeos (A) para uma string de números (B)

5.2.3 Característica 33

Utilizamos as sequências verdadeiras obtidas através do `RegPrecise` e a função `seqconsensus` do `MATLAB` para gerar a sequência consenso do `NtrC`. Essa sequência, apresentada na figura a seguir, é confirmada pela literatura (MERRICK & EDWARDS, 1995;

RegulonDB, 2019) e não representa grau de conservação dos nucleotídeo para cada posição da sequência, visto que o objetivo é apenas obter ilustrar de forma melhor a sequência consenso de nucleotídeos.

FIGURA 19 - SEQUÊNCIA CONSENSO DA NTRC



FONTE: A autora (2019)

LEGENDA: Sequência consenso dos sítios de ligação à NtrC utilizando 921 sítios confirmados.

Para cada sequência candidata fazemos o alinhamento com a sequência consenso e contabilizamos os nucleotídeos que se encaixam no consenso, conforme o método utilizado por Ferreira e colaboradores (2018). O resultado é utilizado como característica.

5.2.4 Característica 34

Assim como Ferreira e colaboradores (2018), fizemos o mesmo processo descrito no subtópico anterior porém utilizando a sequência anticonsenso. A sequência anti consenso (FIGURA 20) foi gerada através das sequências verdadeiras e da função *anticonsensus* desenvolvida no MATLAB, e contém os nucleotídeos menos frequentes para cada posição.

FIGURA 20 - SEQUÊNCIA ANTI-CONSENSO DA NTRC



FONTE: A autora (2019)

LEGENDA: Sequência anti-consenso dos sítios de ligação à NtrC utilizando 921 sítios confirmados.

5.2.5 Característica 35

Contabilizamos também o número de vezes em que a subsequência 'TGCA' ocorre na sequência candidata. Essa subsequência é bem conservada nos sítios de ligação à NtrC e isso é evidenciado tanto pela literatura, quanto pela sequência consenso obtida (FIGURA 19).

5.3 DIVISÃO DOS CONJUNTOS

Para os procedimentos de treinamento e teste dos modelos foram escolhidas, de forma aleatória, 50% do total de sequências. Assim tanto o conjunto de treinamento quanto o de testes possui 7530 sequências.

5.4 PARÂMETROS DOS MODELOS CLASSIFICADORES

Para o modelo DT utilizamos os critérios de impureza de entropia e de gini index para verificar qual classificador é superior.

Para o modelo FAN fizemos o treinamento contínuo e o treinamento limitado a 5000 épocas. Os resultados foram muito semelhantes, então optamos por utilizar 5000 épocas para tornar a etapa de treinamento mais rápida.

Para o modelo KNN utilizamos vizinhança de tamanho 5 e métrica de distância euclidiana.

Para o modelo MLP, utilizamos valores diferentes de camadas de entrada e ocultas a fim de verificar qual era a melhor configuração para o conjunto de dados utilizados. Fizemos combinações alternando entre maior número de camadas de entrada e maior número de camadas ocultas. Os testes foram feitos com: 1 camada de entrada e 2 ocultas, 3 camadas de entrada e 5 ocultas, 5 camadas de entrada e 2 ocultas, e 5 camadas de entrada e 7 ocultas.

Para o modelo NB não foi necessário informar parâmetros.

Para o modelo RBF utilizamos os seguintes números de centro: 10, 20, 30, 40, e 500.

Ambos os classificadores do modelo RF contém 40 árvores de decisão. Geramos classificadores com os critérios de impureza de entropia e de gini index.

Para o modelo SVM não foi necessário passar parâmetros extras além do conjunto de dados. Utilizamos kernel com função linear, função RBF, e função polinomial.

6. RESULTADOS E DISCUSSÃO

Nessa seção apresentamos, seguindo a ordem cronológica de obtenção do resultado, cada resultado e suas discussões pertinentes.

6.1 COMPARAÇÃO DE MODELOS CLASSIFICADORES

Utilizamos as métricas apresentadas no Tópico 3.6 para avaliar os classificadores.

A TABELA 2 apresenta os modelos e os parâmetros de cada um dos classificadores treinados, seguido pelos valores calculados em cada métrica. Com essa tabela conseguimos analisar a performance de cada classificador.

O primeiro método de classificação apresentado na tabela é o Decision Tree (DT), que foi treinado com os critérios de impureza Gini Index e Entropia e atingiu F1-Score de 72,18% e 71,20%, respectivamente. Na matriz de confusão verificamos que das 7067 sequências falsas, 6935 foram classificadas corretamente (TN), e 132 se tornaram falso-positivos (FP). Por outro lado das 463 sequências verdadeiras, 127 foram classificadas como falsas (FN) e 336 foram classificadas corretamente (TP).

A rede FAN atingiu resultados acima da média em todas as métricas, com destaque para as métricas acurácia e F1-Score, onde obteve 98,63% e 88,72% respectivamente. Essas métricas possuem resultados melhores ao ser comparado com as outras redes. Ao voltar a análise para a matriz de confusão verificamos que das 7067 sequências falsas, 7022 foram classificadas corretamente (TN), e apenas 45 se tornaram falso-positivos (FP). Por outro lado das 463 sequências verdadeiras, 58 foram classificadas como falsas (FN) e 405 foram classificadas corretamente (TP). Com esses dados comprovamos que a alta precisão da rede FAN a torna mais resistente à falso-positivos em comparação com os outros modelos testados para o problema proposto.

Utilizamos o algoritmo KNN com vizinhança de tamanho 5 e métrica euclidiana. A execução utilizando o conjunto de teste atingiu 97,09% de acurácia e 77,63% de F1-Score. Como o método não efetua treinamento, um novo modelo é criado para cada conjunto de dados recebido, o que torna essa solução lenta ao utilizar grandes conjuntos de dados.

Verificamos que para a rede neuronal MLP o resultado com poucas camadas se mostra satisfatório em relação a acurácia, mas o melhor resultado para esse modelo foi com 3 camadas de entrada e 5 ocultas, onde atingiu o F1-Score de 77,85%. Apesar disso seus resultados foram superados pelos classificadores FAN, RF, e SVM.

Naive Bayes (NB) é um modelo muito conhecido que atingiu 95,64% de acurácia e 71,58% de F1-Score. O resultado obtido demonstra que o problema apresentado pode ser resolvido com esse classificador, mas conforme o esperado verificamos que outros métodos obtiveram resultados superiores.

As redes RBF tiveram problemas para classificar corretamente os dados em todos os números de centro testados, pois consideraram quase todas as sequências verdadeiras como incorretas. Essa dificuldade de classificação torna-se evidente com as métricas de precisão e de recall utilizadas. A precisão de todas as redes RBF chegaram em 100% porque as poucas sequências que foram classificadas como verdadeiras eram realmente verdadeiras e nenhuma sequência falsa foi classificada como verdadeira. Já o recall de todas as redes RBF se mostra baixo, variando de 0,86% a 8,21%, isso porque a métrica verifica como foram classificadas as sequências verdadeiras e conclui que a maioria foi classificada erroneamente como falsa. Por essa razão o F1-Score, que leva em consideração os valores obtidos nas métricas de precisão e de recall, também obteve valor baixo.

Random Forest utiliza mais de uma árvore de decisão (DT). Treinamos dois modelos utilizando 40 árvores de decisão e diferentes critérios de impureza, onde entropia foi considerado o melhor critério para esse caso por atingir o F1-Score de 82,31%.

O modelo SVM que utiliza função linear obteve acurácia e F1-Score satisfatórios, chegando a atingir 97,45% de acurácia e 77,93% de F1-Score. O melhor classificador do modelo SVM utilizou função polinomial como kernel e obteve acurácia de 97,86% e F1-Score de 82,82%.

TABELA 2 - AVALIAÇÃO DOS CLASSIFICADORES

Modelo	Parâmetros	Acurácia	Precisão	Recall	F1-Score	Matriz de Confusão	
DT	Critério de Impureza: Gini Index	96,56	71,79	72,57	72,18	6935	132
						127	336
DT	Critério de Impureza: Entropia	96,35	69,11	73,43	71,20	6915	152
						123	340
FAN	5000 épocas	98,63	90,00	87,47	88,72	7022	45
						58	405
KNN	k=5 de vizinhança (default), métrica: standard Euclidean metric	97,09	73,64	82,07	77,63	6931	136
						83	380
MLP	1 camada de entrada e 2 ocultas	97,42	87,47	67,82	76,40	7022	45
						149	314

MLP	3 camadas de entrada e 5 ocultas	97,66	82,54	78,62	80,53	6990	77
						99	364
MLP	5 camadas de entrada e 2 ocultas	97,48	85,45	71,06	77,59	7011	56
						134	329
MLP	5 camadas de entrada e 7 ocultas	97,45	90,45	65,44	75,94	7035	32
						160	303
NB	-	95,64	59,77	89,20	71,58	6789	278
						50	413
RBF	10 centros	93,90	100,00	0,86	1,71	7067	0
						459	4
RBF	20 centros	93,92	100,00	1,08	2,14	7067	0
						458	5
RBF	30 centros	93,93	100,00	1,30	2,56	7067	0
						457	6
RBF	40 centros	93,93	100,00	1,30	2,56	7067	0
						457	6
RBF	500 centros	94,36	100,00	8,21	15,17	7067	0
						425	38
RF	40 árvores e Critério de Impureza: Gini Index	97,78	92,29	69,76	79,46	7040	27
						140	323
RF	40 árvores e Critério de Impureza: Entropia	98,05	92,93	73,87	82,31	7041	26
						121	342
SVM	Função linear	97,61	83,93	75,59	79,55	7000	67
						113	350
SVM	Função RBF	94,30	100,00	7,34	13,68	7067	0
						429	34
SVM	Função Polinomial	97,86	81,86	83,80	82,82	6981	86
						75	388

FONTE: A autora (2019)

LEGENDA: Nas duas primeiras colunas identificamos os modelos treinados e seus parâmetros. Nas colunas seguintes temos as métricas de avaliação calculadas: A acurácia corresponde a taxa de sequências classificadas corretamente. A precisão calcula quantas sequências classificadas como verdadeiras são realmente verdadeiras. O recall verifica como as sequências verdadeiras foram classificadas. O F1-Score combina os resultados obtidos nas métricas de precisão e recall. Na matriz de confusão são apresentados na primeira linha os verdadeiros negativos (TN) e os falso-positivos (FP) e na segunda linha os falso-negativos (FN) e os verdadeiros positivos (TP). O conjunto de testes utilizado contém 7067 sequências falsas e 463 sequências verdadeiras. Consideramos o F1-Score como critério de desempate portanto o classificador com avaliação mais alta é a rede FAN, que obteve 88,72%

Para descobrir a taxa de perda para cada modelo, isso é, quantas sequências são perdidas por cada classificador, submetemos as 921 sequências verdadeiras para todos os classificadores treinados.

Na TABELA 3 apresentamos os modelos seguidos pelo número de sequências perdidas (FN), número de sequências recuperadas (TP), e porcentagem de acerto (Acerto).

Os modelos baseados em árvores (Decision Trees e Random Forest) chegam a acertar entre 84 e 86% das sequências verdadeiras.

A rede FAN obteve a terceira menor perda de sequências, com taxa de acerto de 87,62%. Alinhando essa taxa de acerto com a análise feita na tabela 2, concluímos que essa rede neuronal consegue recuperar a maioria das sequências verdadeiras enquanto mantém a taxa de falso-positivos baixa, sendo assim a mais indicada para a solução do problema proposto. Por essa razão a rede neuronal FAN foi escolhida para integrar a ferramenta NtrC Finder.

O classificador KNN também obteve um bom resultado, visto que sua porcentagem de acerto é de 85,34%.

Entre as redes MLP a que atingiu o maior acerto (81%) utiliza 3 camadas de entrada e 5 camadas ocultas.

O classificador NB obteve o segundo melhor resultado, chegando a recuperar 89,69% das sequências verdadeiras.

As redes RBF continuaram a apresentar problemas em sua classificação entretando o teste com 500 números de centro chegou a 74,59%.

O modelo SVM utilizando como kernel a função polinomial perdeu o menor número de sequências verdadeiras, chegando a 91,8% de acerto na classificação.

Para decidir o classificador utilizado pelo NtrC Finder fizemos a análise utilizando os dados de teste (Tabela 2) e os dados verdadeiros (Tabela 3). Assim, como apresentado na TABELA 2, os modelos SVM e Naive Bayes tendem a gerar falsos-positivos, ou seja, classificar sequências incorretas como verdadeiras. Esse fator diminuiu seu F1-Score e por isso optamos pela utilização da rede FAN.

TABELA 3 - TAXA DE RECUPERAÇÃO DE TFBS VERDADEIROS

Modelo	Parâmetros	FN	TP	Acerto
DT	Gini Index	127	794	86,21%
DT	Entropia	123	798	86,64%
FAN	5000 épocas	114	807	87,62%
KNN	k=5, euclidean metric	135	786	85,34%
MLP	1 camada de entrada e 2 ocultas	295	626	67,97%
MLP	3 camadas de entrada e 5 ocultas	175	746	81,00%
MLP	5 camadas de entrada e 2 ocultas	265	656	71,23%
MLP	5 camadas de entrada e 7 ocultas	331	590	64,06%
NB	-	95	826	89,69%
RBF	10 centros	902	19	2,06%
RBF	20 centros	889	32	3,47%
RBF	30 centros	878	43	4,67%
RBF	40 centros	868	53	5,75%
RBF	500 centros	234	687	74,59%
RF	40 árvores, Gini Index	141	780	84,69%
RF	40 árvores, Entropia	121	800	86,86%
SVM	Função linear	429	492	53,42%
SVM	Função RBF	425	496	53,85%
SVM	Função Polinomial	75	846	91,86%

FONTE: A autora (2019)

LEGENDA: Nas duas primeiras colunas identificamos os modelos treinados e seus parâmetros. Na coluna seguinte temos a porcentagem de acerto do classificador. Em seguida temos a matriz de confusão parcial onde são apresentados na primeira linha os falso-negativos (FN) e os verdadeiros positivos (TP). O conjunto de testes utilizado contém as 921 sequências verdadeiras. Após comparar os resultados verificamos que o classificador que conseguiu recuperar a maior quantidade de sequências verdadeiras é o SVM utilizando função polinomial.

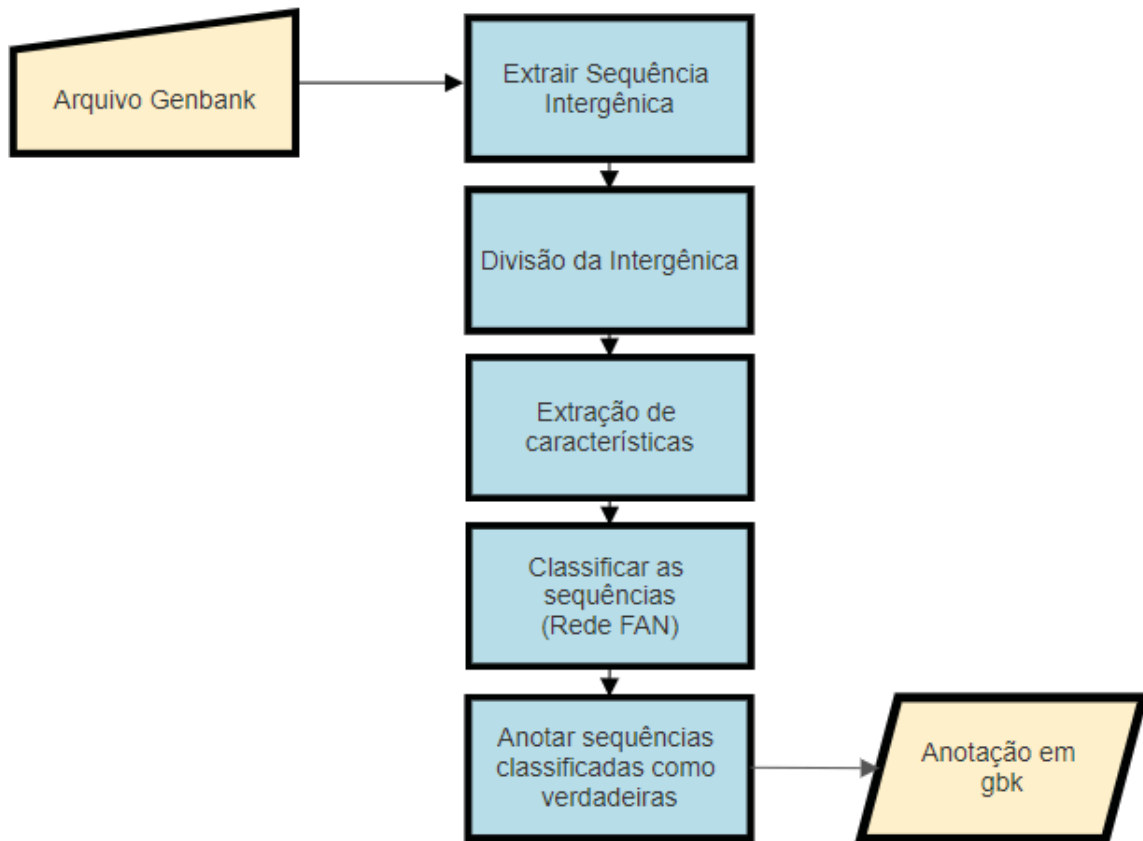
6.2 NTRC FINDER

De acordo com a definição de Gao e colaboradores (2018) apresentada anteriormente, o método de preditor que utilizamos nesse trabalho é considerado tradicional. Assim, com uma janela deslizante obtemos todas as sequências candidatas em uma sequência de DNA e verificamos se cada uma delas equivale a um possível sítio de ligação à NtrC. Para fazer essa verificação utilizamos uma rede neuronal artificial. A seguir explicamos o NtrC Finder:

Construímos o NtrC Finder utilizando o MATLAB. A função abre o arquivo Genbank contendo o genoma da bactéria e extrai a sequência intergênica completa. A seguir utiliza janela deslizante para testar todas as possibilidades para cada sequência de 17bp. Por estarmos interessados nos resultados que encontraríamos, não restringimos a busca para apenas regiões promotoras de genes.

As sequências obtidas então passam pelo mesmo processo de extração de características usado no treinamento e essas são passadas para a rede. O NtrC Finder anota as sequências classificadas como positivo em um arquivo *.gbk*, o qual pode ser aberto em *softwares* de visualização de genoma. Essas etapas estão resumidas na Figura 21.

FIGURA 21 - PROCESSOS DO NTRC FINDER

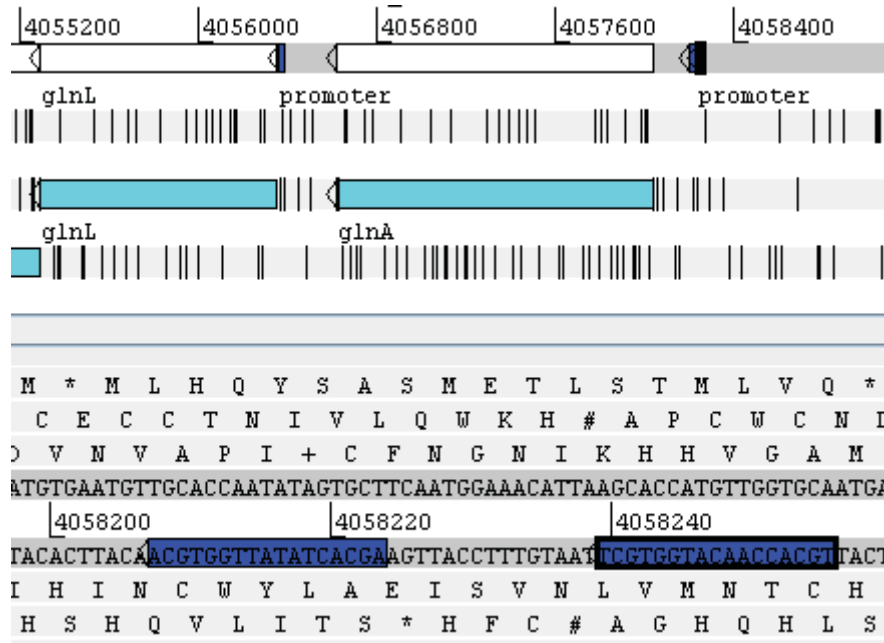


FONTE: A autora (2019)

LEGENDA: O NtrC Finder recebe um arquivo GenBank como parâmetro de entrada, realiza os processos indicados pelos retângulos verdes, e cria um arquivo GenBank para anotar os resultados encontrados.

Para confeccionar a Figura 22 abrimos o arquivo de anotação dos sítios apontados pelo NtrC Finder com o visualizador de genomas Artemis. Na figura é possível verificar dois sítios de ligação (azul escuro) próximo ao gene *glnA* em *E. coli*. Esses sítios foram preditos pela rede e confirmados pela literatura.

FIGURA 22 - VISUALIZAÇÃO DE TFBS ATRAVÉS DO ARTEMIS



FONTE: A autora (2019)

LEGENDA: Os trechos em azul claro indicam genes. Os trechos em azul escuro indicam TFBS preditos pelo NtrC Finder

6.3 GBK2TABLE

O GBK2TABLE é uma ferramenta adicional que permite salvar as informações dos TFBS em um arquivo texto, eliminando a necessidade de utilizar um visualizador de genoma para entender o contexto dos sítos preditos.

Desenvolvemos o script do GBK2TABLE utilizando Python 3 e BioPython. A ferramenta recebe o genoma da bactéria (arquivo *.gbff*, *.gb*, ou *.gbk*) e a anotação resultante da rede (arquivo *.gbk*) e gera um arquivo texto (*.txt*) contendo as informações do operon em que o TFBS se encontra.

O conteúdo do arquivo de saída segue formato tabular e possui as informações do sítio de ligação (sequência, posições de início, de fim, direção de leitura do DNA) e do gene mais próximo (distância até o gene, posição de início, de fim, e nome do gene).

O arquivo texto utiliza o caractere de vírgula (,) como separador, tornando mais fácil a conversão para arquivos de tabela, como os utilizados para visualizar planilhas. Na imagem a seguir (FIGURA 23) evidenciamos no arquivo texto gerado pelo GBK2TABLE os mesmos sítios de ligação vistos na

FIGURA 22.

FIGURA 23 - ANOTAÇÃO DAS INFORMAÇÕES DO TFBS EM ARQUIVO

```

['CGCACCAAAGGGGAGCG', '3853656', '3853672', 1, 250, 3853922, 3855106, 'emrD']
['TGCCTTAATTTCTGCA', '3915358', '3915374', -1, 158, 3913830, 3915200, 'glmU']
['AGCTCACAATATGTGCA', '3944052', '3944068', -1, 2725, 3940635, 3941327, 'yieP']
['TGCGCTTCTTTAGCGCA', '4046609', '4046625', -1, 7, 4045670, 4046602, 'yihG']
['TGCACTAAAATGGTGCA', '4056361', '4056377', -1, 22, 4055290, 4056339, 'glnL']
['AGCACTATATTGGTGCA', '4058207', '4058223', -1, 173, 4056625, 4058034, 'glnA']
['TGCACCAACATGGTGCT', '4058239', '4058255', -1, 205, 4056625, 4058034, 'glnA']
['TGCCTGAATTTTGGTCG', '4127267', '4127283', 1, 1389, 4128672, 4129832, 'metB']
['TGCTCCAGTATTGTGAA', '4161043', '4161059', -1, 253, 4159390, 4160790, 'sthA']
['TGCGCACTAAAAGGGCA', '4177151', '4177167', 1, 191, 4177358, 4177741, 'secE']

```

FONTE: A autora (2019)

LEGENDA: Cada linha traz informações de um TFBS diferente. As informações são separadas por vírgula. Seguindo a ordem das colunas temos a sequência do sítio de ligação, a posição de início do TFBS, a posição final do TFBS, o sentido de leitura, a distância (em bp) até o gene mais próximo, a posição inicial do gene, a posição final do gene, e o nome do gene.

Atualmente utilizamos o GBK2TABLE para disponibilizar as anotações do NtrC Finder na ferramenta web. Em breve pretendemos oferecer acesso ao GBK2TABLE, tornando-se assim mais uma ferramenta auxiliar de bioinformática.

6.4 ESTUDO DE CASO COM ESCHERICHIA COLI

Executamos o NtrC Finder com o genoma completo de *Escherichia coli* str. K-12 substr. MG1655 (NC_000913) e obtivemos 112 sítios de ligação à NtrC.

Utilizamos a ferramenta *GBK2TABLE* para converter a anotação para o formato tabular e recuperar as informações relacionadas à cada gene. Em seguida complementamos a tabela com as informações sobre o operon e comparamos os sítios de ligação candidatos com os sítios de ligação disponíveis na base EcoCyc. Com esse processo validamos uma parte dos resultados através das evidências que o EcoCyc reúne.

Como forma de validar o resultado para genes em que ainda não é conhecida a relação com NtrC, e sabendo que NtrC atua dependentes do fator Sigma 54, decidimos procurar por interações com esse fator. Utilizamos então as informações relacionadas ao Sigma 54 advindas do EcoCyc e fizemos uma busca por sítios de ligação ao Sigma 54 através da ferramenta Sigma54 Finder.

Com o resultado dessas análises desenvolvemos a TABELA 4 - SÍTIOS DE LIGAÇÃO À NTRC EM *E. COLI* MG1655 (NC_000913). Nessa tabela reunimos as informações relacionadas ao sítio de ligação candidato (a sequência em “TFBS Sequence”, a posição inicial em “TFBS start”, a posição final em “TFBS end”, o sentido de leitura em “strand”, a distância até o gene em “distance”), ao gene (posição inicial em “gene start”, posição final em “gene end”, o nome do gene em “gene name”, o nome do operon em “Transcription Units”), verificamos se o sítio de ligação é confirmado pelo EcoCyc (coluna “NtrC Ecocyc”), e se existe presença do fator de transcrição Sigma 54 no operon (utilizando confirmação através do EcoCyc na coluna “s54 Ecocyc” e busca com a ferramenta preditora Sigma54 Finder na coluna “s54 Finder”).

Ao concluir a tabela pudemos fazer as análises a seguir.

Alguns sítios de ligação candidatos contam com a presença de sequências promotoras reconhecidas de sigma 70 e sigma 32 em seus operons. Em *Azospirillum brasilense* o NtrC regula a expressão do gene *glnB* ao ativar ou suprimir os promotores *glnBp2* - σ^N e *glnBp1* - σ^{70} (HUERGO, 2006). Levando isso em consideração abre-se a hipótese de que NtrC não dependa exclusivamente do fator de transcrição sigma 54 e possa regular também promotores tipo sigma70 de maneira análoga ao promotor *glnBp1* de *A. brasilense*.

O NtrC Finder costuma apontar sítios de ligação em que não há confirmações pela literatura, mas ao comparar seus resultados com os sítios encontrados pela ferramenta Sigma54 Finder verifica-se que existem evidências apontadas pelo segundo preditor que dizem que o operon em questão também contém a presença do fator sigma 54. A união dessas duas evidências indicam possíveis locais promissores porém ainda não estudados. Para o caso de *E. coli* os genes com sítios não confirmados são *osmF*, *prfF*, *ptrA*, *rmf*, e *tesB* (conforme é apresentado na TABELA 4 a seguir).

TABELA 4 - SÍTIOS DE LIGAÇÃO À NTRC EM E. COLI MG1655 (NC_000913)

LEGENDA: Para as três últimas colunas: N = Não, Y = Sim. Informações obtidas com o banco de dados Ecocyc e com a ferramenta web S54 Finder

TFBS Sequence	TFBS start	TFBS end	strand	gene distance	gene start	gene end	gene name	Transcription Units	NtrC Ecocyc	s54 Ecocyc	s54 Finder
TGCCTGATTTTGGGCA	3219257	3219273	-1	181	3217556	3219076	aer	aer	N	N	N
TGCTCCTTTATTGGGCC	3281631	3281647	1	329	3281976	3283130	agaS	agaSYBCDI	N	N	N
TGCATAAAGCGGGTGCA	121707	121723	-1	156	120178	121551	aroP	aroP	N	N	N
TGCGTCAGAATGGCGCA	1832269	1832285	-1	287	1830762	1831982	astC	astCADBE	Y	Y	Y
TGCCCGCTTTTGGTGCG	1832289	1832305	-1	307	1830762	1831982	astC	astCADBE	Y	Y	Y
TGCTTCAAAAACGAGTCA	34248	34264	1	36	34300	34695	caIF	caIF	N	N	N
TGCACAAAATGTTGATCA	1563187	1563203	-1	111	1562495	1563076	ddpX	ddpXABCFD	Y	Y	Y
TGCATAAAAACTTAATCA	3708047	3708063	-1	342	3706098	3707705	dppA	dppABCFD	N	N	N
CGCTCATTTTAAATGCA	3410155	3410171	1	109	3410280	3411245	dusB	dusB-fis	N	N	N
CGCACCAAGGGGAGCG	3853656	3853672	1	250	3853922	3855106	emrD	emrD	N	Y	N
TGCATCATAGAGATGCA	658361	658377	-1	423	657555	657938	Fic	crcB	N	N	N
TGCATAAAAACCATGCG	954569	954585	-1	103	953609	954466	focA	focA-pfIB	N	N	N
CGCACAAATAATCAGGCT	2243720	2243736	-1	68	2242984	2243652	folE	folE-yeiB	N	N	N
TGCCCTGTTTTGAATCA	2933703	2933719	-1	15	2933041	2933688	fucA	fucAO	N	N	N
AGCACCCGCAATTAGGCG	2942878	2942894	-1	311	2941650	2942567	gcvA	gcvA	N	N	N
TGCACAAAAGAATGGGCA	4352434	4352450	1	134	4352584	4352880	ghoS	ghoST	N	N	N
TGCCTTAATTTCTGCA	3915358	3915374	-1	158	3913830	3915200	glmU	glmUS	N	N	N
AGCACTATATTGGTGCA	4058207	4058223	-1	173	4056625	4058034	glnA	glnALG	Y	Y	Y
TGCACCAACATGGTGCT	4058239	4058255	-1	205	4056625	4058034	glnA	glnALG	Y	Y	Y
TGCCCCAGAAATGGTGCA	848148	848164	-1	144	847258	848004	glnH	glnHPQ	Y	Y	N
CGCACCCAGATTGGTGCC	848161	848177	-1	157	847258	848004	glnH	glnHPQ	Y	Y	N
TGCACAAATTTAGCGCA	848174	848190	-1	170	847258	848004	glnH	glnHPQ	Y	Y	N
CGCACTATTTAGTGCA	848306	848322	-1	302	847258	848004	glnH	glnHPQ	Y	Y	N
TGCACTGTCATAGTGCG	472461	472477	1	121	472598	472936	glnK	glnK-amtB	Y	Y	Y

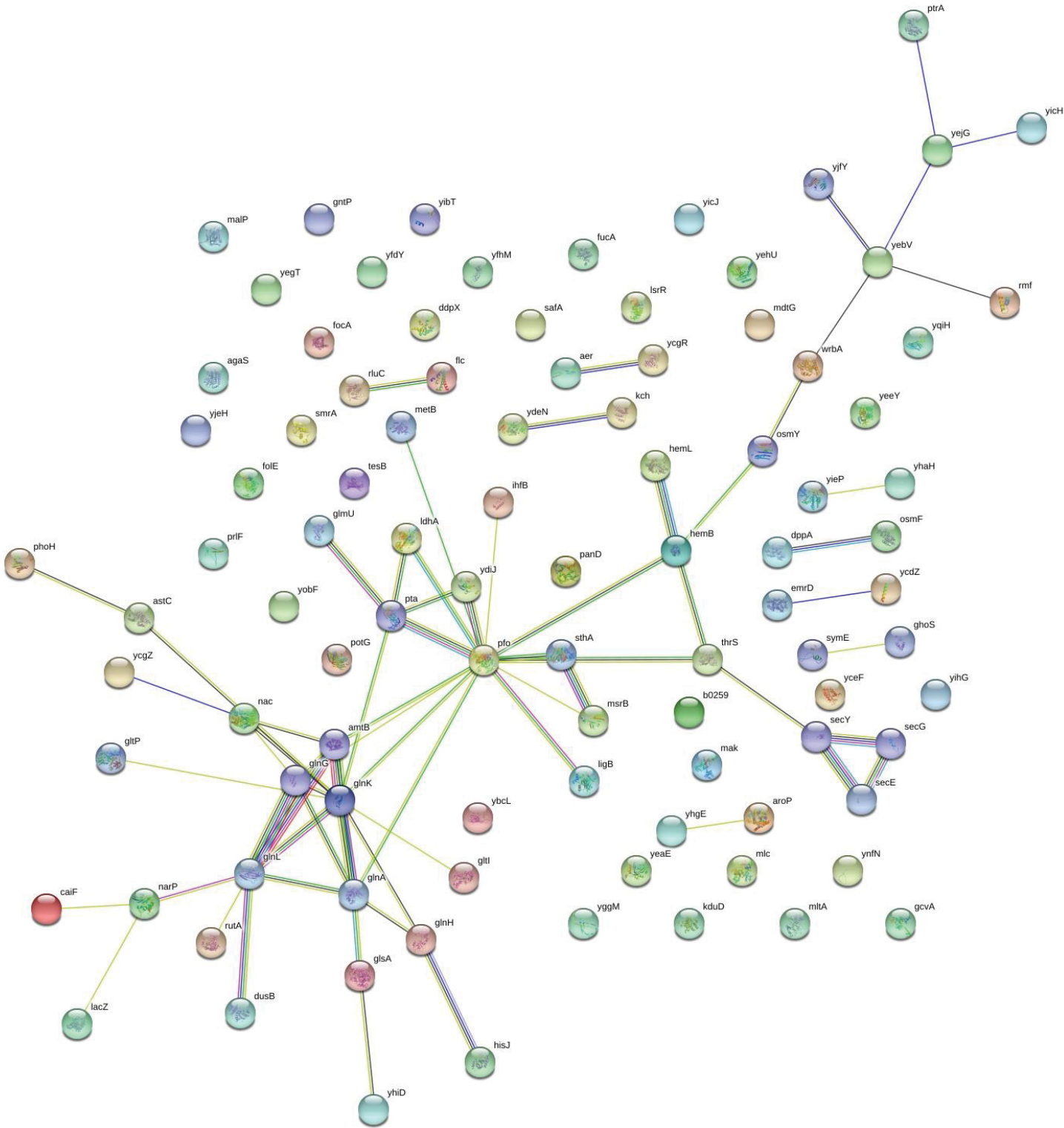
TGCACTAAAATGGTGCA	4056361	4056377	-1	22	4055290	4056339	glnL	glnALG	Y	Y	N
TGGCCACCCCTAAAGCA	508838	508854	1	2787	511641	512573	glsA	glsA-ybaT	N	N	N
TGCACAATAAAGTTGCA	687818	687834	-1	71	686839	687747	gltI	gltIJKL-sroC	N	Y	N
TGCATAAAAAATAATCG	4294171	4294187	1	294	4294481	4295794	gltP	gltP	N	N	N
TGCTGCAAAATTCAGCA	4551414	4551430	-1	118	4549953	4551296	gntP	gntP	N	N	N
TGCATAATCATGGTTCA	390525	390541	-1	798	388753	389727	hemB	hemB	N	N	N
TGATCACATTATTGGTCA	175086	175102	-1	204	173602	174882	hemL	hemL	N	N	N
TGCACTATCGTGGTGCA	2426934	2426950	-1	146	2426006	2426788	hisJ	argT-hisJQMP	Y	Y	N
CGCCCTTAATCAATGCA	963767	963783	1	45	963828	964112	infB	rpsA-infB	N	N	N
TGCTTCGGAATGGTGGC	689115	689131	-1	102	687997	689013	insH1	insH-1	N	N	N
AGCACTAAAAGAGTGCA	689144	689160	-1	131	687997	689013	insH1	insH-1	N	N	N
TGTTCAGAAAATGTGCA	1310311	1310327	-1	42	1309016	1310269	kch	Kch	N	N	N
TGGCTAGTTGTGGGCA	2983279	2983295	-1	21	2982497	2983258	kduD	kduD	N	N	N
AGTCACCTCATTAGGCA	366388	366404	-1	83	363231	366305	lacZ	lacZYA	N	N	N
TGGCCTACACTAAAGCA	1442953	1442969	-1	110	1441854	1442843	ldhA	ldhA	N	N	N
TGGCAAAAGTTGTGGC	3821317	3821333	-1	147	3819488	3821170	ligB	ligB	N	N	N
TGCTTTTAATTTGTTC	1601426	1601442	-1	185	1600288	1601241	lsrR	lsrRK	N	N	N
CGCATAATAATTGCTGCG	408984	409000	1	1144	410144	411052	mak	mak	N	N	N
TGCATTGATTTGATGCT	3552637	3552653	-1	164	3550080	3552473	malP	malPQ	N	N	N
CGCATAATTAGTGTGCT	1115542	1115558	-1	52	1114264	1115490	mdtG	mdtG	N	N	N
TGCCTGAATTTTGGTCG	4127267	4127283	1	1389	4128672	4129832	metB	metBL	N	N	N
AGCCCGAAAAAATGTGCT	1668664	1668680	-1	100	1667344	1668564	mlc	mlc-ynfK	N	N	N
AGCCTATTTTTTGTGCA	2947271	2947287	-1	93	2946081	2947178	mitA	mitA	N	N	N
ATCACGATTTAGGTGCA	1862537	1862553	-1	108	1862016	1862429	msrB	msrB	N	N	N
TGCCTAAGCATTACGCG	1862646	1862662	-1	217	1862016	1862429	msrB	msrB	N	N	N
TGAACCATCGTGGTGCA	2062122	2062138	-1	189	2061016	2061933	nac	nac	Y	Y	Y
TGCCCCATTACTCATCT	2286864	2286880	1	3620	2290500	2291147	narP	narP	N	N	N
TGCCCAACGTAAATGCA	2287394	2287410	1	3090	2290500	2291147	narP	narP	N	N	N
TGCACAAATATCAGTTCC	575964	575980	-1	-861	575786	576825	nmpC	nmpC	N	N	N
TGGCCCTTTGTAGGCC	2219652	2219668	-1	171	2218564	2219481	osmF	osmF-yehYXW	N	N	Y

GGCTCAAATTACGAGCA	4611360	4611376	1	20	4611396	4612001	osmY	osmY	N	N	N	N
TAAACAAAAATCGGGCA	146766	146782	-1	72	146314	146694	panD	panD	N	N	N	N
CGCCCTCATTTGCGCA	1440800	1440816	-1	16	1437260	1440784	pfo	ydbK-ompN	N	N	N	N
CGCAGCAATTCGTGCG	1440815	1440831	-1	31	1437260	1440784	pfo	ydbK-ompN	N	N	N	N
TGCCACAAATCAGTGCG	1084711	1084727	1	265	1084992	1086056	phoH	phoH	N	N	N	N
CGCACCAATTATGGTGCG	894944	894960	1	31	894991	896124	potG	potFGHI	Y	Y	Y	N
AGCTTATAATTTGAGCA	3276956	3276972	1	30	3277002	3277337	prlF	sohA-yhaV	N	N	N	Y
TGTATAAAAAATTCGCGCA	2959039	2959055	-1	155	2955996	2958884	ptrA	ptrA-recBD	N	N	N	Y
TGAAGTAATAAGGTGCA	1144858	1144874	1	66	1144940	1145899	rluC	rluC	N	N	N	N
TGCACATTTAGTAATCA	1015655	1015671	1	44	1015715	1015882	rnf	rnf	N	N	N	Y
CGCATCATTTGAAGTGCA	3815948	3815964	-1	-601	3815881	3816549	rph	rph-pyrE	N	N	N	N
TGCATGTTTTATGTGCA	1074142	1074158	-1	131	1072863	1074011	rutA	rutABCDEFG	Y	Y	Y	Y
TGCACTCTCATCGCGCA	1074163	1074179	-1	152	1072863	1074011	rutA	rutABCDEFG	Y	Y	Y	Y
CGCTCAGTACTGAAGCA	1584050	1584066	-1	91	1583762	1583959	safA	safA-ydeO	N	N	N	N
TGCGCACTAAAAGGGCA	4177151	4177167	1	191	4177358	4177741	secE	secE-nusG	N	N	N	N
TGCTTAAAATATCGGCA	1405866	1405882	1	97	1405979	1406542	smrA	ydaL	N	N	N	N
TGCTCCAGTATTGTGAA	4161043	4161059	-1	253	4159390	4160790	sthA	sthA	N	N	N	N
CGCACCTTTCGGTGCG	4580037	4580053	-1	197	4579499	4579840	symE	symE	N	N	N	N
TGCACGAGTTTCATTCA	475376	475392	-1	215	474301	475161	tesB	tesB	N	N	N	Y
TGCATATCTCTTGTGCG	1805138	1805154	-1	2568	1800642	1802570	thrS	thrS-infC	N	N	N	N
TGCATAGAAATTAACGCG	1068032	1068048	-1	324	1067112	1067708	wrbA	wrbA-yccJ	N	N	N	N
TGCATGAACATTGCGCC	570811	570827	1	66	570893	571444	ybcL	ybcLM	N	N	N	N
TGCATAAAAATGTGTGCT	1100238	1100254	1	42	1100296	1100787	ycdZ	ycdZ	N	N	N	N
TGCTTTATTTTCGTTCA	1146682	1146698	-1	87	1146011	1146595	yceF	yceF	N	N	N	N
CGCTCTAAGTATAGGCA	1244662	1244678	-1	135	1243793	1244527	ycgR	ycgR	N	N	N	N
AGCCCAATTAATTGAGCC	1211674	1211690	1	4099	1215789	1216025	ycgZ	ycgZ-ymgA-ariR-ymgC	N	N	N	N
CGCGTTGTTTTAGGCG	1582841	1582857	-1	317	1580842	1582524	ydeN	ydeNM	N	N	N	N
TTCACCTTTTTGTGCG	1769049	1769065	-1	364	1765629	1768685	ydiJ	ydiJ-meniI-ydiH	N	N	N	N
TGCATAAAAACAGGGCG	1865684	1865700	-1	48	1864782	1865636	yeaE	yeaE	N	N	N	N
TGCATCTTTCAGGGCA	1865706	1865722	-1	70	1864782	1865636	yeaE	yeaE	N	N	N	N

CGCACAAAAAAGCGCA	1921666	1921682	1	98	1921780	1922016	yebV	yebV	N	N	N
GGCTGAAAAATGGTGCG	1921716	1921732	1	48	1921780	1922016	yebV	yebV	N	N	N
TGCACTGCAAGGGGGCG	2088262	2088278	-1	4	2087329	2088258	yeeY	yeeY	N	N	N
TGCACAAATACCCGGCCA	2171795	2171811	1	7010	2178821	2180098	yegTUV	yegTUV	N	N	N
TGCCCGTTTTTTGTGCA	2214784	2214800	-1	140	2212959	2214644	yehU	bisSR	N	N	N
TGCACAAACGAGGAAGCT	2278548	2278564	-1	311	2277893	2278237	yejG	yejG	N	N	N
ATCACAAATACITGTGCA	2290284	2290300	-1	104	2290114	2290180	yejO	yejO	N	N	N
TGATTATTAATCTGTGCA	2495439	2495455	-1	147	2495050	2495292	yfdY	yfdY	N	N	N
CGCATAATAATTATTCCG	2652323	2652339	-1	36	2647326	2652287	yfhM	yfhM	N	N	N
CGCTTAAATACAGAGCA	2777784	2777800	1	8597	2786397	2788649	ygaQ	ygaQ	N	N	N
CGCCCCGGTATCGTGCC	3099659	3099675	-1	94	3098558	3099565	yggM	yggM	N	N	N
TGAGCAAAAATTGAGGCA	3252284	3252300	1	4	3252304	3252669	yhaH	yhaH	N	N	N
CGCACCAATTGCGGGCG	3418868	3418884	1	158	3419042	3419107	yhdW	yhdWXYZ	Y	Y	N
TGAATGAAATTGATGCA	3532560	3532576	-1	121	3530715	3532439	yhgE	yhgE	N	N	N
CGAATTAATGAGGTGCA	3655957	3655973	-1	55	3655255	3655902	yhiD	hdeAB-yhiD	N	N	N
TGCTCAAAAACACTGCGCA	3776542	3776558	-1	162	3776171	3776380	yibT	yibT	N	N	N
TGATTCTATAAAGTGCA	3769756	3769772	1	176	3769948	3770034	yibU	yibU	N	N	N
TGCGCGTAAATCGTGCA	3830404	3830420	1	37	3830457	3832166	yich	yich	N	N	N
CGATTGAAATATTGAGCA	3836377	3836393	-1	448	3834547	3835929	yicJ	yicJ	N	N	N
AGCTCACAAATATGTGCA	3944052	3944068	-1	2725	3940635	3941327	yieP	yieP-hsrA	N	N	N
TGCGCTTCTTTAGCGCA	4046609	4046625	-1	7	4045670	4046602	yihG	yihG	N	N	N
TGCCGCAATCTTAAGCA	4362692	4362708	-1	339	4362270	4362353	yjdQ	-	N	N	N
TGCCCCAAAATTTGGCGG	4370430	4370446	-1	18	4369156	4370412	yjeH	yjeH	N	N	N
TGCATCAGAAAATGGTCA	4424879	4424895	-1	88	4424516	4424791	yjfY	yjfY	N	N	N
TGGTTTATATTGGTGCG	1590772	1590788	-1	236	1590426	1590536	yneL	yneL	N	N	N
TGAATCCTTTCCGGGCGCA	1638349	1638365	-1	240	1637954	1638109	yfnN	yfnN	N	N	N
CGCACTGTCATGGTGCA	1908184	1908200	-1	593	1907448	1907591	yobF	yobF-cspC	N	N	N
TGCACCACATCAGGGCG	3187903	3187919	1	1962	3189881	3190630	yqIH	yqIG-insC-5D-5-yqIH-insCD-5	N	N	N

Submetemos a lista dos genes da TABELA 4 para a ferramenta web STRING v11, que gerou uma rede regulatória a partir deles. Essa rede é mostrada na Figura 24.

FIGURA 24 - REDE REGULATÓRIA DOS GENES



FONTE: A autora (2019)

LEGENDA: As linhas que ligam dois ou mais genes representam as interações. Linha azul claro: interações conhecidas de bases de dados curadas; Linha rosa: interações conhecidas determinadas experimentalmente; Linha verde: interações previstas de vizinhança de gene; Linha vermelha: interações previstas de fusão de gene; Linha azul: interações previstas de co-ocorrência de gene; Linha amarela: interações previstas por text-mining; Linha preta: interações previstas por co-expressão; Linha lilás: interações previstas por homologia de proteínas;

A ferramenta STRING identificou o enriquecimento no nosso conjunto de genes para Proteínas periplasmáticas de ligação ao substrato. Nesse conjunto encontram-se os genes *glnH*, que resulta na proteína de ligação periplasmática de um sistema de transporte de L glutamina ABC, *gltI*, que resulta na proteína de ligação periplasmática de um sistema de transporte de glutamato / aspartato ABC, e *hisJ*, que resulta na proteína de ligação periplasmática de um sistema de captação de histidina dependente de ATP (EcoCyc, 2019).

Além dessa constatação, ampliamos o alcance da análise do STRING e agrupamos os genes de acordo com seus processos biológicos. Isso resultou na TABELA 8 - ANÁLISE DOS ENRIQUECIMENTOS ENCONTRADOS PELO STRING: PROCESSOS BIOLÓGICOS (GO), disponível no Anexo III.

6.5 FERRAMENTA WEB DO NTRC FINDER

A ferramenta web do NtrC Finder está disponível através do endereço http://200.236.3.16/bioinfo_apps/index.html. Ao acessar a página o usuário é apresentado a uma tela como a da FIGURA 25. Nessa tela escrevemos um breve texto de introdução explicando o que é a página e o que pode ser acessado através dela. No NtrC Finder disponibilizamos a consulta ao banco de dados com sítios de ligação à NtrC previstos computacionalmente para 310 replicons; Oferecemos acesso a ferramenta de predição NtrC Finder para que o usuário faça novas predições para organismos de seu interesse; Apresentamos informações de outras ferramentas de bioinformática desenvolvidas no laboratório.

FIGURA 25 - TELA INICIAL NTRC FINDER

NtrC FINDER

Home Browse Run NtrC Finder See results Other tools ▾

Welcome to NtrC Finder

NtrC Finder is a web application to predict NtrC transcription factor binding sites (TFBS) in bacteria.

The NtrC Finder uses our Neural Network and searches for possible TFBS in whole intergenic area in Genbank files and annotates in a `.gbk` file.

We predicted binding sites for 685 organisms and the results can be accessed on the browse section.

You can also run the prediction tool for other files at the Run NtrC Finder section.

All the results are available at the See results section. If you want a `.csv` table file try [this tool](#) to convert the genbank annotation to a table format.

Our lab is responsible for other bioinformatic tools and you can learn more about them at the Other Tools section

Citation:
If you use please cite NtrC Finder as NtrC Finder: Prediction Tool for NtrC binding sites in prokaryotes.

BROWSE NTRC DB
TRANSCRIPTION FACTOR BINDING SITES DATABASE

Browse predicted NtrC
You can browse, visualize and download data from our database

[Read More](#)

FONTE: A autora (2019)

A página de acesso ao banco de dados é mostrada na FIGURA 26. Nela carregamos uma lista com as informações do nome completo do genoma, o número de acesso, o número de sítios de ligação encontrados, um link para visualização dos TFBS e a opção de *download* das informações.

FIGURA 26 - CONSULTA AO BANCO DE DADOS NTRC FINDER

The screenshot shows the NtrC FINDER web application. The header includes the logo and navigation menu. The main heading is 'Browse Predicted NtrC Binding Sites in Bacteria'. Below this is a table with the following data:

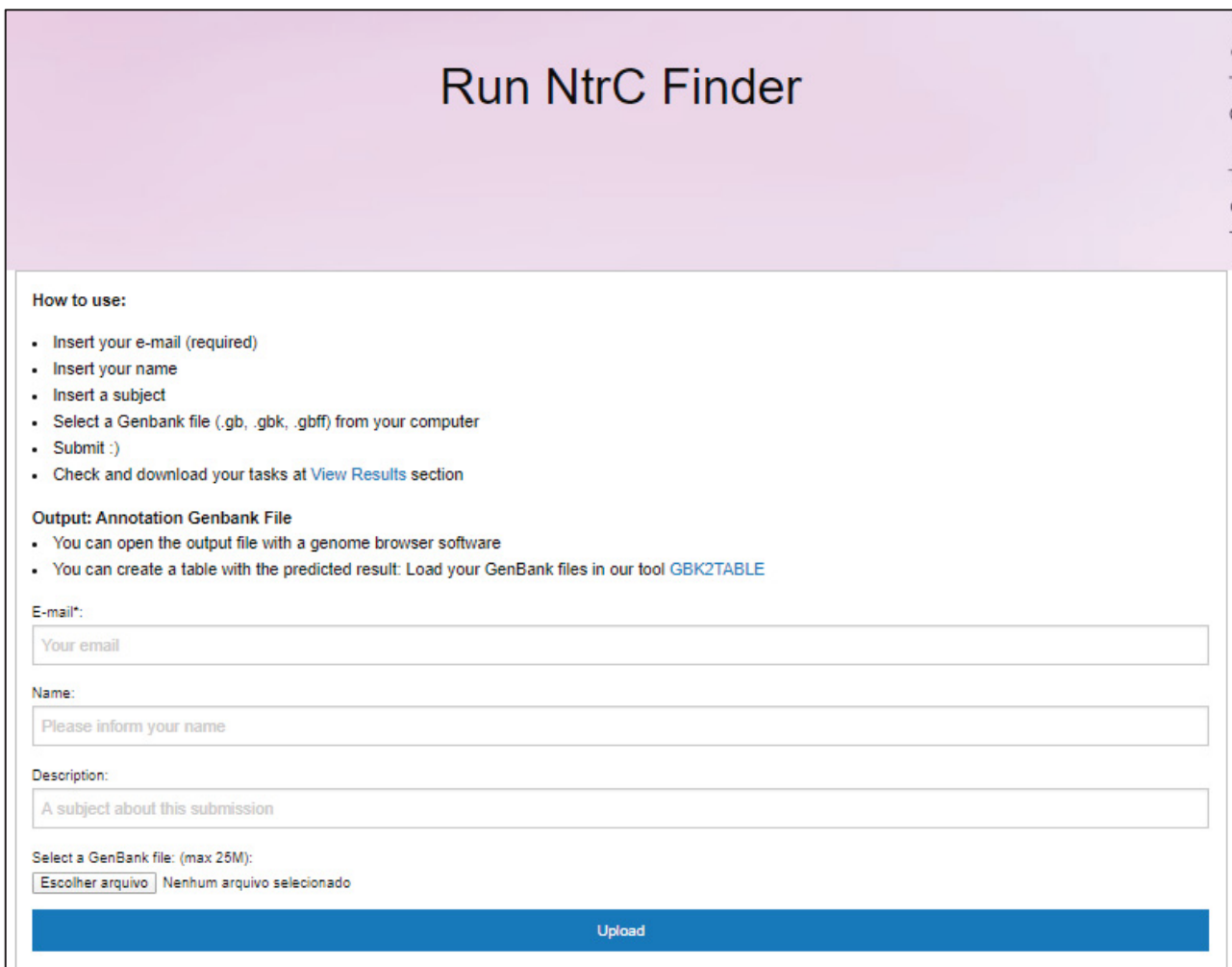
Organism	Accession	TFBS Found	View	Download
Acidithiobacillus ferrovarans SS3, complete genome	NC_015942.1	137	View	Download
Allochrocatium vinosum DSM 180 chromosome, complete genome	NC_013851.1	71	View	Download
Allochrocatium vinosum DSM 180 plasmid pALVIN01, complete sequence	NC_013852.1	6	View	Download
Anabaena cylindrica PCC 7122 plasmid pANACY.06, complete sequence	NC_019775.1	118	View	Download
Anabaena variabilis ATCC 29413 plasmid A, complete sequence	NC_007410.1	10	View	Download

On the right side of the page, a DNA sequence is shown with a predicted binding site highlighted in blue: `ATAAACCCAGCCTCGG`, `TTATCCGCCCGCCTAG`, `TGCACAATTTTAGCGCA`, `TCTTTAGTCCCGTAAT`, `CTTAATCGCCACGCATC`, `TGAGTTCGACATCTCCG`, `AGAACTTAAAGAAGTTC`, `TCAGGCCCCCAAGTA`, `CATTAGCGTCGCTGCTA`, `TGAGTCGACATCTCCG`, `AACAGCTTAGTCAACC`.

FONTE: A autora (2019)

Na FIGURA 27 apresentamos a página da ferramenta preditora NtrC Finder. Nela o usuário precisa adicionar um e-mail, seu nome, um título para o processo, e um arquivo Genbank. O usuário é avisado via e-mail sobre o status da predição. Os status e os resultados são acessíveis através da opção “See results” no menu superior. Vale notar que não é necessário fazer cadastro, login, ou baixar a ferramenta. Acreditamos que com isso facilite a experiência do usuário na página.

FIGURA 27 - UPLOAD NO NTRC FINDER



The screenshot displays the 'Run NtrC Finder' web interface. At the top, there is a purple header with the title 'Run NtrC Finder'. Below the header, the page is divided into several sections:

- How to use:** A list of instructions: 'Insert your e-mail (required)', 'Insert your name', 'Insert a subject', 'Select a Genbank file (.gb, .gbk, .gbff) from your computer', 'Submit :)', and 'Check and download your tasks at [View Results](#) section'.
- Output: Annotation Genbank File**: A list of instructions: 'You can open the output file with a genome browser software' and 'You can create a table with the predicted result: Load your GenBank files in our tool [GBK2TABLE](#)'.
- E-mail*:** A text input field with the placeholder 'Your email'.
- Name:** A text input field with the placeholder 'Please inform your name'.
- Description:** A text input field with the placeholder 'A subject about this submission'.
- Select a GenBank file: (max 25M):** A file selection area with a button labeled 'Escolher arquivo' and the text 'Nenhum arquivo selecionado'.

At the bottom of the form is a large blue button labeled 'Upload'.

FONTE: A autora (2019)

LEGENDA: Captura de tela da área do formulário de inclusão de arquivo GenBank

7. CONCLUSÃO

A proteína NtrC participa de um importante sistema de regulação em bactérias diazotróficas responsável pela ativação de fontes alternativas de nitrogênio. Essa proteína se liga a regiões específicas do DNA, que são chamadas de sítios de ligação ao fator de transcrição (ou TFBS, do inglês *Transcription Factor Binding Sites*). Os métodos tradicionais incluem diversos testes em laboratório, o que o torna custoso caso seja aplicado para genomas completos, e os métodos computacionais identificam esses sítios através da localização de motifs, porém essa abordagem resulta em uma grande quantidade de falso-positivos. Visando solucionar esses problemas desenvolvemos o NtrC Finder.

Fizemos testes utilizando diferentes formas de extrair características e diferentes modelos de classificadores (DT, FAN, KNN, MLP, NB, RBF, RF, e SVM) utilizando a biblioteca sklearn (Python 3) e os softwares MATLAB e EasyFan. Os classificadores foram testados utilizando métricas de avaliação bem estabelecidas. A rede neuronal artificial FAN revelou ser o classificador mais eficiente dentre os testados para integrar a ferramenta, recuperando 87,62% dos TFBS verdadeiros confirmados de *E. coli*, e apresentando o maior F1-Score entre os classificadores (88,72% com o conjunto de testes).

Em nosso modelo proposto criamos uma técnica para diminuir a taxa de falso-positivos na predição de TFBS para o genoma completo. Essa técnica consiste na utilização de sequências incorretas provenientes de bactérias modelo (*E. coli*, *P. aeruginosa*, e *V. cholerae*) para aumentar o conjunto de sequências classificadas como falso e conseqüentemente diminuir a taxa de falso-positivos. O resultado foi significativo: O número de sítios de ligação preditos pelo NtrC Finder para *E. coli* diminuiu de 6270 sequências para 112.

NtrC Finder é uma ferramenta preditora de sítios de ligação à proteína NtrC em genomas completos de bactérias que utiliza uma rede neuronal treinada especificamente para identificar as sequências, que são anotadas em um arquivo .gbk e disponibilizadas ao usuário para utilizá-lo em softwares de visualização de genoma.

Testamos o NtrC Finder com o genoma modelo *E. coli* e encontramos relações esperadas e sítios promissores. Essa ferramenta pode ajudar pesquisadores a encontrar genes candidatos para regulação por NtrC em genomas de interesse, para explicar fenômenos que ocorrem, ou como ponto de partida para novas pesquisas. Os sítios de ligação apontados parecem promissores e bem relacionados entre si conforme vimos na rede regulatória gerada no STRING.

Executamos o NtrC Finder foi executado para 310 replicons de bactérias diazotróficas (entre cromossomos e plasmídeos) e disponibilizamos os TFBS preditos através da página web da ferramenta.

Implementamos o NtrC Finder como um sistema web. Para fazer uma nova predição, o usuário carrega um arquivo no formato Genbank e informa seu e-mail sem a necessidade de fazer cadastro no site. Nesse sistema é possível consultar o andamento das predições, baixar o arquivo de anotação dos sítios identificados, e acessar o banco de dados contendo resultados das predições realizadas pelo NtrC Finder para 310 replicons bacterianos, disponibilizado para o usuário. NtrC Finder está disponível através do endereço http://200.236.3.16/bioinfo_apps/.

Como perspectivas futuras pretendemos fazer melhorias na ferramenta web, disponibilizando ainda mais ferramentas auxiliares e mantendo atualizadas as já existentes, explorar os sítios de ligação encontrados, e realizar mais testes com o método. Para melhorar a sensibilidade da rede neuronal utilizada podemos fazer o retreinamento incluindo no conjunto de dados incorretos os TFBS preditos que se encontram distantes de regiões codificantes, pois apesar de se encaixar no padrão encontrado o TFBS não corresponde a um sítio de ligação à NtrC verdadeiro.

8. REFERÊNCIAS

- AGATONOVIC-KUSTRIN, S; BERESFORD, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, v. 22, n. 5, p. 717–727, 1 jun. 2000. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0731708599002721>>.
- ARAGHINEJAD, Shahab. *Artificial Neural Networks*. . [S.l.: s.n.], 2013. Disponível em: <http://cabgrid.res.in/cabin/publication/smfa/Module IV/3. Artificial Neural Networks_GK Jha.pdf>.
- BODEN, Marcus *et al.* Alignment-free sequence comparison with spaced k-mers. 2013, [S.l.: s.n.], 2013. p. 24–34.
- BROWN, Daniel R. *et al.* Nitrogen stress response and stringent response are coupled in *Escherichia coli*. *Nature Communications*, v. 5, n. 1, p. 4115, 20 set. 2014. Disponível em: <<http://www.nature.com/articles/ncomms5115>>.
- CARMACK, C Steven *et al.* PhyloScan: identification of transcription factor binding sites using cross-species evidence. *Algorithms for Molecular Biology*, v. 2, p. 1, 2007. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1794230/>>.
- CARVER, T. *et al.* Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics*, v. 28, n. 4, p. 464–469, 15 fev. 2012. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/22199388>>.
- CHEN, Ke; KURGAN, Lukasz A. Neural Networks in Bioinformatics. *Handbook of Natural Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 565–583. Disponível em: <http://link.springer.com/10.1007/978-3-540-92910-9_18>.
- COELHO, Leandro dos Santos; RAITTZ, Roberto Tadeu; TREZUB, Maurício. FControl: Sistema Neuro-Nebuloso-Evolutivo Aplicado à Detecção de Fraudes em Operações de Comércio Eletrônico. 29 ago. 2016, [S.l.]: SBRN, 29 ago. 2016. p. 1–6. Disponível em: <http://abricom.org.br/eventos/cbrn_2005/CBRN2005_059>.
- CROOKS, Gavin E *et al.* WebLogo: a sequence logo generator. *Genome research*, v. 14, n. 6, p. 1188–90, jun. 2004. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/15173120>>.
- De Pierri, C.R., Voyceik, R., Santos de Mattos, L.G.C. *et al.* SWeeP: representing large biological sequences datasets in compact vectors. *Sci Rep* **10**, 91 (2020). <https://doi.org/10.1038/s41598-019-55627-4>
- D'HAESELEER, Patrik. What are DNA sequence motifs? *Nature Biotechnology*, v. 24, n. 4, p. 423–425, abr. 2006. Disponível em: <<http://www.nature.com/articles/nbt0406-423>>.
- DIMITRIADIS, Stavros I.; LIPARAS, Dimitris. *How random is the random forest? Random forest algorithm on the service of structural imaging biomarkers for Alzheimer's disease: From Alzheimer's disease neuroimaging initiative (ADNI) database*. *Neural Regeneration Research*. [S.l.]: Wolters Kluwer Medknow Publications. , 1 jun. 2018
- ERNST, Jason *et al.* A Semi-Supervised Method for Predicting Transcription Factor–Gene Interactions in *Escherichia coli*. *PLoS Computational Biology*, v. 4, n. 3, p. e1000044, 28 mar. 2008. Disponível em: <<https://dx.plos.org/10.1371/journal.pcbi.1000044>>.
- ESSEBIER, Alexandra *et al.* Bioinformatics approaches to predict target genes from transcription factor binding data. *Methods*, v. 131, p. 111–119, 1 dez. 2017. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1046202317302207?via%3Dihub>>.

- FERREIRA, L. *et al.* The identification of DNA binding regions of the σ 54 factor using artificial neural network. *bioRxiv*, p. 393736, 17 ago. 2018. Disponível em: <<https://www.biorxiv.org/content/10.1101/393736v2>>.
- FREIRE, Rodnei Damaceno. Identificação e análise de promotores Sigma 70 no genoma de *Herbaspirillum seropedicae* SmR1 utilizando métodos de inteligência artificial. 2014. Disponível em: <<https://acervodigital.ufpr.br/handle/1884/36878>>.
- GAO, Liangxin *et al.* Fast sequence analysis based on diamond sampling. *PLOS ONE*, v. 13, n. 6, p. e0198922, 28 jun. 2018. Disponível em: <<https://dx.plos.org/10.1371/journal.pone.0198922>>.
- GHRITLAHRE, Harish Kumar; PRASAD, Radha Krishna. Exergetic performance prediction of solar air heater using MLP, GRNN and RBF models of artificial neural network technique. *Journal of environmental management*, v. 223, p. 566–575, 1 out. 2018. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/29975883>>.
- GIRALDI, Gilson A. *et al.* Statistical learning approaches for discriminant features selection. *Journal of the Brazilian Computer Society*, v. 14, n. 2, p. 7–22, jun. 2008. Disponível em: <<https://journal-bcs.springeropen.com/articles/10.1007/BF03192556>>.
- GONZALEZ, A. D. *et al.* TRACTOR_DB: a database of regulatory networks in gamma-proteobacterial genomes. *Nucleic Acids Research*, v. 33, n. Database issue, p. D98–D102, 17 dez. 2004. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/15608293>>.
- H. KHALAFI, Farshad Faghihi; M., S. A Literature Survey of Neutronics and Thermal-Hydraulics Codes for Investigating Reactor Core Parameters; Artificial Neural Networks as the VVER-1000 Core Predictor. *Nuclear Power - System Simulations and Operation*. [S.l.]: InTech, 2011. . Disponível em: <<http://www.intechopen.com/books/nuclear-power-system-simulations-and-operation/a-literature-survey-of-neutronics-and-thermal-hydraulics-codes-for-investigating-reactor-core-parame>>. Acesso em: 24 maio 2019.
- HAPUDENIYA, Muditha. Artificial Neural Networks in Bioinformatics. *Sri Lanka Journal of Bio-Medical Informatics*, v. 1, n. 2, p. 104, 7 abr. 2010. Disponível em: <<https://sljbmi.sljol.info/article/10.4038/sljbmi.v1i2.1719/>>.
- HOANG, Duc. What is the statistical model behind the SVM algorithm? *Quora*, 2017. Disponível em: <<https://www.quora.com/What-is-the-statistical-model-behind-the-SVM-algorithm>>.
- HOLLOWAY, Dustin T.; KON, Mark; DELISI, Charles. Machine learning for regulatory analysis and transcription factor target prediction in yeast. *Systems and Synthetic Biology*, v. 1, n. 1, p. 25–46, 21 fev. 2007. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/19003435>>.
- HUANG, Shujun *et al.* *Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genomics and Proteomics*. [S.l.]: International Institute of Anticancer Research. , 1 jan. 2018
- INUKAI, Sachi; KOCK, Kian Hong; BULYK, Martha L. Transcription factor–DNA binding: beyond binding site motifs. *Current Opinion in Genetics & Development*, v. 43, p. 110–119, abr. 2017. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/28359978>>.
- KALOGIROU, Soteris A. Long-term performance prediction of forced circulation solar domestic water heating systems using artificial neural networks. *Applied Energy*, v. 66, n. 1, p. 63–74, maio 2000. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0306261999000422>>.
- KEILWAGEN, Jens; GRAU, Jan. Varying levels of complexity in transcription factor binding motifs. *Nucleic acids research*, v. 43, n. 18, p. e119, 15 out. 2015. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/26116565>>.

- KHAMIS, Abdullah M *et al.* A novel method for improved accuracy of transcription factor binding site prediction. *Nucleic Acids Research*, v. 46, n. 12, p. e72–e72, 6 jul. 2018. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/29617876>>.
- KIRK, M. *Thoughtful Machine Learning with Python: A Test-driven Approach*. [S.l.]: O'Reilly, 2017. Disponível em: <<https://books.google.com.br/books?id=DCd4rgEACAAJ>>.
- KOTSIANTIS, S B. Supervised Machine Learning: A Review of Classification Techniques. 2007, Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 3–24. Disponível em: <<http://dl.acm.org/citation.cfm?id=1566770.1566773>>.
- KOVÁCS, Zsolt László. *Redes neurais artificiais : fundamentos e aplicações*. [S.l.]: Ed. Livraria da Física, 2002. Disponível em: <<https://www.estantevirtual.com.br/livros/zsolt-l-kovacs/redes-neurais-artificiais-fundamentos-e-aplicacoes/2614607246>>. Acesso em: 24 maio 2019.
- KUBAT, Miroslav. Neural Networks: A Comprehensive Foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. *Knowl. Eng. Rev.*, v. 13, n. 4, p. 409–412, 1999. Disponível em: <<http://dl.acm.org/citation.cfm?id=975792.975796>>.
- KUSTER, Claiton Werner, 1967- *et al.* EasyFan. 2016. Disponível em: <<https://acervodigital.ufpr.br/handle/1884/41156>>.
- LEIGH, John A.; DODSWORTH, Jeremy A. Nitrogen Regulation in Bacteria and Archaea. *Annual Review of Microbiology*, v. 61, n. 1, p. 349–377, out. 2007. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/17506680>>.
- LI, Yifeng; SHI, Wenqiang; WASSERMAN, Wyeth W. Genome-Wide Prediction of cis-Regulatory Regions Using Supervised Deep Learning Methods. *bioRxiv*, p. 041616, 28 fev. 2016. Disponível em: <<https://www.biorxiv.org/content/10.1101/041616v1>>.
- LI, Yu *et al.* Deep learning in bioinformatics: introduction, application, and perspective in big data era. 2019.
- LORENA, Ana *et al.* Uma Introdução às Support Vector Machines. *Revista de Informática Teórica e Aplicada; Vol. 14, No 2 (2007); 43-67*, v. 14, 2007.
- MATHELIER, Anthony; WASSERMAN, Wyeth W. The Next Generation of Transcription Factor Binding Site Prediction. *PLoS Computational Biology*, v. 9, n. 9, p. e1003214, 5 set. 2013. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/24039567>>.
- MAXWELL, Aaron E.; WARNER, Timothy A.; FANG, Fang. *Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing*. [S.l.]: Taylor and Francis Ltd. , 3 maio 2018
- MCCUE, L. A. Factors Influencing the Identification of Transcription Factor Binding Sites by Cross-Species Comparison. *Genome Research*, v. 12, n. 10, p. 1523–1532, 1 out. 2002. Disponível em: <<http://www.genome.org/cgi/doi/10.1101/gr.323602>>.
- MEHTA, Pankaj; SCHWAB, David J.; SENGUPTA, Anirvan M. Statistical Mechanics of Transcription-Factor Binding Site Discovery Using Hidden Markov Models. *Journal of statistical physics*, v. 142, n. 6, p. 1187, 2011. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3407691/>>.
- MERRICK, M J; EDWARDS, R a. Nitrogen control in bacteria. *Microbiological reviews*, v. 59, n. 4, p. 604–622, 1995.
- MIN, Seonwoo; LEE, Byunghan; YOON, Sungroh. Deep learning in bioinformatics. *Briefings in Bioinformatics*, v. 18, n. 5, p. bbw068, 29 jul. 2016. Disponível em: <<https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbw068>>.

- MIRONOV, A A *et al.* Computer analysis of transcription regulatory patterns in completely sequenced bacterial genomes. *Nucleic acids research*, v. 27, n. 14, p. 2981–9, 15 jul. 1999. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/10390542>>.
- NOBLE, William S. What is a support vector machine? *Nature Biotechnology*, v. 24, n. 12, p. 1565–1567, dez. 2006. Disponível em: <<http://www.nature.com/articles/nbt1206-1565>>.
- NOVICHKOV, Pavel S. *et al.* RegPrecise 3.0 - A resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC Genomics*, v. 14, n. 1, p. 1, 2013a. Disponível em: <BMC Genomics>.
- NOVICHKOV, Pavel S *et al.* RegPrecise 3.0--a resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC genomics*, v. 14, p. 745, nov. 2013b.
- OLIVER, Patricia *et al.* Molecular and structural considerations of TF-DNA binding for the generation of biologically meaningful and accurate phylogenetic footprinting analysis: the LysR-type transcriptional regulator family as a study model. *BMC Genomics*, v. 17, n. 1, p. 686, 27 dez. 2016. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/27567672>>.
- ORR, Mark J L. *Introduction to Radial Basis Function Networks*. . [S.l.: s.n.], 1996. Disponível em: <www.anc.ed.ac.uk/mjo/software/rbf.zip>. Acesso em: 24 maio 2019.
- PEDREGOSA, F *et al.* Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.
- PENNSYLVANIA STATE UNIVERSITY: STATISTICS ONLINE COURSES. *Lesson 10: Support Vector Machines | STAT 508*.
- RAITZ, Roberto Tadeu. Free Associative Neurons - FAN : uma abordagem para reconhecimento de padrões /. 1997. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/77282>>.
- RODIONOV, Dmitry A. Comparative Genomic Reconstruction of Transcription Regulatory Networks in Bacteria. *Chemical Reviews*, v. 107, n. 8, p. 3467–3497, 2007.
- SANCHUKI, Heloisa B.S. *et al.* Dynamics of the Escherichia coli proteome in response to nitrogen starvation and entry into the stationary phase. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, v. 1865, n. 3, p. 344–352, mar. 2017. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/27939605>>.
- SHAO, Xiaojian *et al.* Predicting DNA- and RNA-binding proteins from sequences with kernel methods. *Journal of Theoretical Biology*, v. 258, n. 2, p. 289–293, 21 maio 2009. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022519309000289?via%3Dihub#!>>.
- SHIAU, S P *et al.* Role of nitrogen regulator I (NtrC), the transcriptional activator of glnA in enteric bacteria, in reducing expression of glnA during nitrogen-limited growth. *Journal of bacteriology*, v. 174, n. 1, p. 179–85, jan. 1992. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/1345910>>.
- SINGH, A; THAKUR, N; SHARMA, A. A review of supervised machine learning algorithms. 2016, [S.l.: s.n.], 2016. p. 1310–1315.
- SINGH, Ritambhara *et al.* DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, v. 32, n. 17, p. i639–i648, 1 set. 2016. Disponível em: <<https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btw427>>.
- STADEN, Rodger. Methods for calculating the probabilities of finding patterns in sequences. *Bioinformatics*, v. 5, n. 2, p. 89–96, 1 abr. 1989. Disponível em: <<https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/5.2.89>>.

- STORMO, G. D. DNA binding sites: representation and discovery. *Bioinformatics*, v. 16, n. 1, p. 16–23, 1 jan. 2000. Disponível em: <<https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/16.1.16>>.
- STUDHOLME, David J; DIXON, Ray. Domain architectures of sigma54-dependent transcriptional activators. *Journal of bacteriology*, v. 185, n. 6, p. 1757–1767, 2003.
- SWITZER, Amy; BROWN, Daniel R.; WIGNESHWERARAJ, Sivaramesh. New insights into the adaptive transcriptional response to nitrogen starvation in *Escherichia coli*. *Biochemical Society Transactions*, v. 46, n. 6, p. 1721–1728, 17 dez. 2018. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/30514772>>.
- TATUSOV, Roman L. *et al.* Metabolism and evolution of Haemophilus influenzae deduced from a whole-genome comparison with Escherichia coli. *Current Biology*, v. 6, n. 3, p. 279–291, 1 mar. 1996. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0960982202004785>>.
- TRENTINI, Débora Broch. Identificação dos alvos celulares das proteínas de transdução de sinal PII do diazotrófico de vida livre Azospirillum amazonense. 2010. Disponível em: <<https://lume.ufrgs.br/handle/10183/21431>>.
- WANG, Dianhui; ALHAMDOOSH, Monther; PEDRYCZ, Witold. ANFIS-based fuzzy systems for searching dna-protein binding sites. *bioRxiv*, p. 058800, 15 jun. 2016. Disponível em: <<https://www.biorxiv.org/content/10.1101/058800v1.full>>.
- XU, Tianlei *et al.* A comprehensive review of computational prediction of genome-wide features. *Briefings in Bioinformatics*, 16 nov. 2018. Disponível em: <<https://academic.oup.com/bib/advance-article/doi/10.1093/bib/bby110/5177808>>.
- ZANATY, E.A. Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification. *Egyptian Informatics Journal*, v. 13, n. 3, p. 177–183, 1 nov. 2012. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1110866512000345>>.
- ZHANG, Harry; ZHANG, Harry. The Optimality of Naïve Bayes. *IN FLAIRS2004 CONFERENCE*, 2004. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.483.2183>>.
- ZHANG, Yan-ping *et al.* gDNA-Prot: Predict DNA-binding proteins by employing support vector machine and a novel numerical characterization of protein sequence. *Journal of Theoretical Biology*, v. 406, p. 8–16, 7 out. 2016. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0022519316301308>>.
- ZHANG, Yanping *et al.* newDNA-Prot: Prediction of DNA-binding proteins by employing support vector machine and a comprehensive sequence representation. *Computational Biology and Chemistry*, v. 52, p. 51–59, 1 out. 2014. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1476927114001054>>.
- ZIMMER, D P *et al.* Nitrogen regulatory protein C-controlled genes of Escherichia coli: scavenging as a defense against nitrogen limitation. *Proceedings of the National Academy of Sciences of the United States of America*, v. 97, n. 26, p. 14674–14679, dez. 2000.

ANEXO I – CONJUNTO VERDADEIRO

Para integrar o conjunto de dados verdadeiros obtivemos do banco de dados RegPrecise 921 sítios de ligação à NtrC. Os organismos e genes referentes a esses sítios estão listados na tabela a seguir.

TABELA 5 - ORGANISMOS E GENES ONDE OCORRE LIGAÇÃO À NTRC

Organismo	Genes
Acetobacter pasteurianus IFO 3283-01	nifR3
Acidovorax avenae subsp. citrulli AAC00-1	glnA, glnA, CHP02001, CHP02001, nrtA, nasD
Acidovorax sp. JS42	glnA, glnA, CHP02001, nasD, ybiB, nrtA
Acinetobacter baumannii AB0057	ntrB, gltB, gdhA, gdhA, gdhA, glnA, glnA
Acinetobacter sp. ADP1	gltB, gltB, rutR, ntrB, gdhA, gdhA, gdhA, rutA, rutR, rutA, glnA, glnA
Aeromonas hydrophila subsp. hydrophila ATCC 7966	glnA, glnA, amtB2, glnK2, ntrB
Aeromonas salmonicida subsp. salmonicida A449	glnA, glnA, amtB2, glnK, glnK2, ntrB
Agrobacterium tumefaciens str. C58 (Cereon)	nrtA, nifR3, glnB, glnB
Alcanivorax borkumensis SK2	glnK, glnK, PF09694, glnA, glnA, ntrB, ureD
Alteromonadales bacterium TW-7	glnK, ntrB, glnA
Alteromonas macleodii 'Deep ecotype'	glnK, glnA, ntrB
Azoarcus sp. EbN1	ntrC, CHP02001, glnA, amtB2, gltB
Azorhizobium caulinodans ORS 571	dppA, nrtA, glnK, urtA, uctA, glnB
Azospirillum sp. B510	potA, potA, amtB, amtB, glnK, glnK, amtB2, nrtA, glnB
Azotobacter vinelandii AvOP	ybdK, glnA, glnA, ntrB, glnK
Bradyrhizobium japonicum USDA 110	urtA, urtA, nirA, narK, glnK, dppA, glnB
Bradyrhizobium sp. BTAi1	dppA, urtA, urtA, urtA, nirA, nrtA, glnK, uctA, nifR3, glnB
Brucella melitensis 16M	amtB, nifR3, nifR3, glnB
Burkholderia cepacia AMMD	amtB, narK, narK, glnA
Burkholderia glumae BGR1	glnA, glnA, narK
Burkholderia mallei ATCC 23344	glnA, narK, narK, glnA
Burkholderia phymatum STM815	narK, glnA
Burkholderia pseudomallei K96243	narK, narK, glnA, glnA
Burkholderia sp. 383	amtB, narK, narK, glnA
Burkholderia vietnamiensis G4	amtB, narK, narK, glnA
Burkholderia xenovorans LB400	glnA, nasD
Caulobacter crescentus CB15	glnK, glnB, nifR3, narK, narK
Caulobacter segnis ATCC 21756	glnK, nifR3
Caulobacter sp. K31	narK, glnK, glnB, narK, nifR3

<i>Cellvibrio japonicus</i> Ueda107	<i>glnA</i> , <i>ntrB</i> , <i>glnK</i>
<i>Chromobacterium violaceum</i> ATCC 12472	<i>ntrB</i> , <i>glnK</i> , <i>glnA</i> , <i>glnA</i>
<i>Chromohalobacter salexigens</i> DSM 3043	<i>glnK</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>urtA</i> , <i>urtA</i>
<i>Citrobacter koseri</i> ATCC BAA-895	<i>ntrB</i> , <i>glnK</i> , <i>glnA</i> , <i>glnA</i> , <i>CKO_01526</i> , <i>gltI</i> , <i>gltI</i> , <i>hisJ</i> , <i>hisJ</i> , <i>glnH</i> , <i>glnH</i> , <i>nac</i> , <i>potG</i> , <i>ygjG</i>
<i>Colwellia psychrerythraea</i> 34H	<i>glnK</i> , <i>glnA</i> , <i>ntrB</i>
<i>Comamonas testosteroni</i> KF-1	<i>glnA</i> , <i>glnA</i> , <i>CHP02001</i> , <i>CHP02001</i> , <i>nrtA</i> , <i>nasD</i>
<i>Cupriavidus taiwanensis</i>	<i>dppA</i> , <i>amtB</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrZ</i>
<i>Dechloromonas aromatica</i> RCB	<i>glnA</i> , <i>CHP02001</i> , <i>glnA</i> , <i>amtB2</i> , <i>amtB2</i> , <i>nasD</i> , <i>ntrB</i>
<i>Delftia acidovorans</i> SPH-1	<i>CHP02001</i> , <i>CHP02001</i> , <i>glnA</i> , <i>glnA</i> , <i>glnA</i> , <i>nrtA</i> , <i>nasD</i>
<i>Desulfuromonas acetoxidans</i> DSM 684	<i>nifR</i> , <i>gdhA</i> , <i>nifEN</i> , <i>amtB</i> , <i>glnB</i> , <i>glnB</i>
<i>Edwardsiella tarda</i> EIB202	<i>glnK</i> , <i>ntrB</i> , <i>glnA</i> , <i>glnA</i> , <i>gltI</i> , <i>gltI</i>
<i>Enterobacter</i> sp. 638	<i>ntrB</i> , <i>glnA</i> , <i>glnA</i> , <i>astC</i> , <i>astC</i> , <i>rutA</i> , <i>glnK</i> , <i>gltI</i> , <i>gltI</i> , <i>hisJ</i> , <i>hisJ</i> , <i>glnH</i> , <i>glnH</i> , <i>nac</i> , <i>potG</i> , <i>ygjG</i>
<i>Erwinia amylovora</i> ATCC 49946	<i>ntrB</i> , <i>glnA</i> , <i>glnA</i> , <i>astC</i> , <i>astC</i> , <i>glnK</i> , <i>glnK</i> , <i>glnH</i> , <i>nac</i> , <i>EAM_0872</i> , <i>EAM_0872</i> , <i>gltI</i>
<i>Erwinia carotovora</i> subsp. <i>atroseptica</i> SCRI1043	<i>ntrB</i> , <i>glnA</i> , <i>glnA</i> , <i>glnK</i> , <i>nac</i> , <i>ygjG</i> , <i>gltI</i>
<i>Erythrobacter litoralis</i> HTCC2594	<i>glnK</i> , <i>glnK</i> , <i>glnB</i> , <i>nifR3</i>
<i>Erythrobacter</i> sp. NAP1	<i>glnB</i> , <i>glnK</i> , <i>glnK</i> , <i>nifR3</i>
<i>Escherichia coli</i> str. K-12 substr. MG1655	<i>ntrB</i> , <i>glnA</i> , <i>glnA</i> , <i>rutA</i> , <i>astC</i> , <i>ddpX</i> , <i>ddpX</i> , <i>astC</i> , <i>glnK</i> , <i>gltI</i> , <i>hisJ</i> , <i>glnH</i> , <i>glnH</i> , <i>nac</i> , <i>potG</i> , <i>ygjG</i>
<i>Geobacter lovleyi</i> SZ	<i>gdhA</i> , <i>gdhA</i> , <i>gdhA</i> , <i>Gmet_0693</i> , <i>Gmet_0693</i> , <i>glnB</i>
<i>Geobacter metallireducens</i> GS-15	<i>nifR</i> , <i>gdhA</i> , <i>nifEN</i> , <i>Gmet_0693</i> , <i>glnB</i>
<i>Geobacter</i> sp. FRC-32	<i>nifR</i> , <i>gdhA</i> , <i>gdhA</i> , <i>gdhA</i> , <i>nifEN</i> , <i>Gmet_0693</i> , <i>glnB</i> , <i>glnB</i>
<i>Geobacter</i> sp. M21	<i>nifR</i> , <i>gdhA</i> , <i>gdhA</i> , <i>Gmet_0693</i> , <i>glnB</i>
<i>Geobacter sulfurreducens</i> PCA	<i>nifR</i> , <i>gdhA</i> , <i>gdhA</i> , <i>gdhA</i> , <i>nifEN</i> , <i>Gmet_0693</i> , <i>glnB</i> , <i>glnB</i>
<i>Geobacter uraniumreducens</i> Rf4	<i>nifR</i> , <i>gdhA</i> , <i>gdhA</i> , <i>nifEN</i> , <i>glnB</i> , <i>glnB</i>
<i>Glaciecola</i> sp. HTCC2999	<i>glnK</i> , <i>glnA</i> , <i>ntrB</i>
<i>Gluconacetobacter diazotrophicus</i> PAI 5	<i>nifR3</i> , <i>glnK</i>
<i>Granulibacter bethesdensis</i> CGDNIH1	<i>glnK</i> , <i>nifR3</i>
<i>Hahella chejuensis</i> KCTC 2396	<i>glnK</i> , <i>glnK</i> , <i>glnA</i> , <i>ntrB</i> , <i>urtA</i>
<i>Hyphomonas neptunium</i> ATCC 15444	<i>nifR3</i> , <i>glnB</i> , <i>glnB</i>
<i>Idiomarina baltica</i> OS145	<i>ntrB</i> , <i>glnA</i>
<i>Idiomarina loihiensis</i> L2TR	<i>glnK</i> , <i>ntrB</i> , <i>glnA</i>
<i>Jannaschia</i> sp. CCS1	<i>nifR3</i> , <i>amtB2</i> , <i>glnK</i> , <i>ureD</i> , <i>glnB</i> , <i>nasT</i> , <i>gdhA</i> , <i>nrtA</i> , <i>nrtB</i>

<i>Klebsiella pneumoniae</i> subsp. <i>pneumoniae</i> MGH 78578	ntrB, glnA, glnA, rutA, rutA, astC, glnK, gltI, hisJ, hisJ, glnH, glnH, nac, EAM_0872, EAM_0872, potG
<i>Laribacter hongkongensis</i> HLHK9	glnA, glnA
<i>Leptothrix cholodnii</i> SP-6	glnA, glnA, glnK, nrtA
<i>Loktanella vestfoldensis</i> SKA53	glnK, glnK, nifR3, ureD, glnB, urtA
<i>Magnetospirillum magneticum</i> AMB-1	amtB, amtB, glnK, amtB2, amtB2, amtB2, nrtA, nrtA2, glnB, nifR3
<i>Magnetospirillum magnetotacticum</i> MS-1	glnK, amtB2, nrtA3, glnB, glnB, nifR3
<i>Marinobacter aqueolei</i>	glnK, glnA, glnA, glnA, ntrB, urtA, urtA
<i>Marinobacter</i> sp. ELB17	glnK, glnA, ntrB, urtA
<i>Marinomonas</i> sp. MWYL1	ntrB, glnA, glnA, glnK
<i>Mesorhizobium loti</i> MAFF303099	glnK, glnB
<i>Mesorhizobium</i> sp. BNC1	narK, glnK
<i>Methylibium petroleiphilum</i> PM1	glnA
<i>Methylobacillus flagellatus</i> KT	glnK, glnA, glnA, glnK
<i>Methylophilales bacterium</i> HTCC2181	CHP02001, amtB2, glnA, glnA
<i>Methylotenera mobilis</i> JLW8	glnA, ntrB, ntrB, glnA, CHP02001, amtB2, amtB2, nasD
<i>Moritella</i> sp. PE36	glnA, glnK, ntrB
<i>Nitrobacter winogradskyi</i> Nb-255	nifR3, glnB
<i>Novosphingobium aromaticivorans</i> DSM 12444	glnB, glnB, glnK, glnK, nifR3
<i>Oceanicola batsensis</i> HTCC2597	ureD, nifR3, nifR3, glnK, glnK, urtA, urtA
<i>Oceanicola granulosus</i> HTCC2516	nifR3, nifR3, glnB, urtA, ureD
<i>Oceanobacter</i> sp. RED65	glnA, glnA, ntrB, ureD, PF09694, PF09694
<i>Oceanospirillum</i> sp. MED92	glnA, glnA, ntrB, glnK, glnK
<i>Paracoccus denitrificans</i> PD1222	amtB2, nifR3, glnB, glnB, nasT
<i>Pelobacter carbinolicus</i> str. DSM 2380	amtB, amtB, glnK2, glnB, glnB
<i>Pelobacter propionicus</i> DSM 2379	nifR, gdhA, nifEN, Gmet_0693, Gmet_0693, glnB, glnB
<i>Phenylobacterium zucineum</i> HLK1	glnK, glnB, nifR3
<i>Photobacterium profundum</i> SS9	glnA, glnK, ntrB
<i>Photorhabdus luminescens</i> subsp. <i>laumondii</i> TTO1	glnA, ntrB, glnA, glnK, gltI
<i>Polaromonas naphthalenivorans</i> CJ2	glnA, glnA, glnK, glnK, nrtA
<i>Polaromonas</i> sp. JS666	glnA, glnA, CHP02001, narK, narK
<i>Proteus mirabilis</i> HI4320	ntrB, glnA, glnA, glnK
<i>Pseudoalteromonas atlantica</i> T6c	glnK, glnA, ntrB
<i>Pseudoalteromonas haloplanktis</i> TAC125	glnK, ntrB, glnA
<i>Pseudoalteromonas tunicata</i> D2	glnK, ntrB, glnA
<i>Pseudomonas aeruginosa</i> PAO1	ybdK, ntrB, glnK, glnA, glnA
<i>Pseudomonas entomophila</i> L48	alsT, amtB2, glnA, ntrB, ntrB, glnK
<i>Pseudomonas fluorescens</i> Pf-5	alsT, amtB2, ybdK, glnA, ntrB, glnK, glnK

<i>Pseudomonas mendocina</i> ymp	ybdK, glnA, ntrB, glnK, glnA, glnK
<i>Pseudomonas putida</i> KT2440	amtB2, ybdK, ybdK, glnA, glnA, ntrB, glnK, glnK
<i>Pseudomonas stutzeri</i> A1501	glnA, ntrB, glnK, glnK, glnA
<i>Pseudomonas syringae</i> pv. tomato str. DC3000	alsT, ybdK, glnA, ntrB, ntrB, glnK, glnK
<i>Psychrobacter arcticum</i> 273-4	COG0733, glnA
<i>Psychrobacter</i> sp. PRwf-1	ntrB, ntrB, ntrB, COG0733, glnA, glnA
<i>Psychromonas ingrahamii</i> 37	glnA, glnA, glnK
<i>Psychromonas</i> sp. CNPT3	glnA, glnA, glnK
<i>Ralstonia eutropha</i> H16	dppA, amtB, amtB, glnA, ntrZ
<i>Ralstonia eutropha</i> JMP134	narK, dppA, amtB, glnA, ntrZ, amtB
<i>Ralstonia metallidurans</i> CH34	narK, dppA, dppA, glnA, ntrZ, amtB
<i>Ralstonia pickettii</i> 12J	narK, amtB, glnA
<i>Ralstonia solanacearum</i> GMI1000	dppA, narK, glnA
<i>Reinekea</i> sp. MED297	glnK, glnK, glnA, glnA, ntrB
<i>Rhizobium etli</i> CFN 42	nrtA, nrtA, narK, glnK, uctA, nifR3, glnB, glnB
<i>Rhizobium leguminosarum</i> bv. <i>viciae</i> 3841	nrtA, glnK, uctA, uctA, nifR3, glnB
<i>Rhizobium</i> sp. NGR234	narK, narK, nrtC, nrtC, nrtA, nrtA, glnK, uctA, uctA, nifR3, glnB, glnB
<i>Rhodobacter sphaeroides</i> 2.4.1	glnK, nifR3, nifR3, glnB, ureD
<i>Rhodobacterales bacterium</i> HTCC2654	glnK, glnK, nifR3, nifR3, glnB, nasT, urtA, urtA, nrtA
<i>Rhodoferax ferrireducens</i> DSM 15236	glnA, glnA, glnK, nrtA
<i>Rhodopseudomonas palustris</i> CGA009	dppA, urtA, glnK, uctA, nifR3, urtA, glnB
<i>Rhodospirillum centenum</i> SW	amtB, amtB, glnK, nifR3, glnB
<i>Rhodospirillum rubrum</i> ATCC 11170	potA, potA, amtB, glnB, nifR3
<i>Roseobacter</i> sp. MED193	nifR3, nifR3, amtB2, glnK, ureD, glnB, glnB, nasT, urtA, nrtA
<i>Roseovarius nubinhibens</i> ISM	glnK, nifR3, nifR3, glnK, glnB, glnB
<i>Roseovarius</i> sp. 217	glnK, nifR3, ureD, nifR3, glnK, glnB, nasT
<i>Saccharophagus degradans</i> 2-40	glnA, ntrB, ureD, ureD, PF09694
<i>Salmonella typhimurium</i> LT2	ntrB, glnA, glnA, astC, astC, glnK, gtlI, gtlI, hisJ, hisJ, glnH, glnH
<i>Serratia proteamaculans</i> 568	ntrB, glnA, glnA, rutA, CKO_01526, glnK, gtlI, gtlI, hisJ, hisJ, glnH, glnH, EAM_0872, EAM_0872
<i>Shewanella amazonensis</i> SB2B	glnB, glnB, glnA, glnA, glnA, ntrB, ntrB
<i>Shewanella baltica</i> OS155	glnB, glnK2, glnA, ntrB, ntrB
<i>Shewanella denitrificans</i> OS217	glnA, glnA, glnB, glnB, ntrB, ntrB
<i>Shewanella frigidimarina</i> NCIMB 400	glnB, glnA, ntrB, ntrB
<i>Shewanella halifaxensis</i> HAW-EB4	glnB, glnA, ntrB, ntrB
<i>Shewanella loihica</i> PV-4	glnB, glnB, glnA, glnA, ntrB, ntrB

<i>Shewanella oneidensis</i> MR-1	<i>glnB</i> , <i>glnK2</i> , <i>glnA</i> , <i>glnA</i> , <i>glnK2</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella pealeana</i> ATCC 700345	<i>glnB</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i>
<i>Shewanella piezotolerans</i> WP3	<i>glnB</i> , <i>glnA</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella putrefaciens</i> CN-32	<i>glnB</i> , <i>glnA</i> , <i>glnK2</i> , <i>glnK2</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella sediminis</i> HAW-EB3	<i>glnB</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella</i> sp ANA-3	<i>glnB</i> , <i>glnA</i> , <i>glnA</i> , <i>glnK2</i> , <i>glnK2</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella</i> sp MR-4	<i>glnB</i> , <i>glnK2</i> , <i>glnK2</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella</i> sp MR-7	<i>glnB</i> , <i>glnA</i> , <i>glnK2</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella</i> sp W3-18-1	<i>glnK2</i> , <i>glnK2</i> , <i>glnA</i> , <i>glnB</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Shewanella woodyi</i> ATCC 51908	<i>glnK2</i> , <i>glnA</i> , <i>glnA</i> , <i>glnA</i> , <i>glnB</i> , <i>glnB</i> , <i>ntrB</i> , <i>ntrB</i>
<i>Silicibacter pomeroyi</i> DSS-3	<i>nifR3</i> , <i>nifR3</i> , <i>glnK</i> , <i>glnB</i> , <i>glnB</i>
<i>Silicibacter</i> TM1040	<i>ureD</i> , <i>nifR3</i> , <i>nifR3</i> , <i>glnK</i> , <i>glnB</i> , <i>gdhA</i>
<i>Sinorhizobium meliloti</i> 1021	<i>narK</i> , <i>narK</i> , <i>glnK</i> , <i>nifR3</i> , <i>glnB</i> , <i>glnB</i>
<i>Sphingobium japonicum</i> UT26S	<i>glnK</i> , <i>glnK</i> , <i>glnB</i> , <i>glnB</i> , <i>nifR3</i>
<i>Sphingomonas wittichii</i> RW1	<i>glnK</i> , <i>glnK</i> , <i>glnB</i> , <i>nifR3</i>
<i>Sphingopyxis alaskensis</i> RB2256	<i>glnB</i> , <i>glnK</i> , <i>nifR3</i>
<i>Stenotrophomonas maltophilia</i> K279a	<i>glnA</i> , <i>glnA</i> , <i>ntrB</i>
<i>Sulfitobacter</i> sp. EE-36	<i>ureD</i> , <i>nifR3</i> , <i>nifR3</i> , <i>glnK</i> , <i>glnK</i> , <i>glnB</i> , <i>nasT</i> , <i>gdhA</i> , <i>urtA</i>
<i>Teredinibacter turnerae</i> T7901	<i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>ureD</i> , <i>urtA</i> , PF09694
<i>Thauera</i> sp. MZ1T	CHP02001, <i>ntrC</i> , <i>glnA</i> , <i>glnA</i> , <i>amtB2</i> , <i>amtB2</i> , <i>gltB</i> , <i>nasD</i> , <i>nasD</i>
<i>Thiobacillus denitrificans</i>	<i>glnA</i> , <i>glnA</i> , <i>amtB2</i> , <i>nasA</i>
<i>Tolomonas auensis</i> DSM 9187	<i>amtB2</i> , <i>glnK2</i> , <i>glnA</i> , <i>glnA</i> , <i>ntrB</i>
<i>Variovorax paradoxus</i> S110	<i>glnA</i> , CHP02001, <i>nasD</i> , <i>nrtA</i> , <i>narK</i>
<i>Verminephrobacter eiseniae</i> EF01-2	CHP02001, <i>glnA</i> , <i>glnA</i>
<i>Vibrio angustum</i> S14	<i>glnA</i> , <i>glnK</i> , <i>ntrB</i>
<i>Vibrio cholerae</i> O1 biovar eltor str. N16961	<i>ntrB</i> , <i>glnK</i> , <i>glnA</i>
<i>Vibrio fischeri</i> ES114	<i>glnK</i> , <i>ntrB</i> , <i>glnA</i>
<i>Vibrio harveyi</i> ATCC BAA-1116	<i>ntrB</i> , <i>glnK</i> , <i>glnA</i>
<i>Vibrio parahaemolyticus</i> RIMD 2210633	<i>ntrB</i> , <i>glnK</i> , <i>amtB2</i> , <i>glnA</i>
<i>Vibrio shilonii</i> AK1	<i>amtB2</i> , <i>ntrB</i> , <i>glnA</i>
<i>Vibrio splendidus</i> LGP32	<i>ntrB</i> , <i>glnK</i> , <i>amtB2</i> , <i>glnA</i>
<i>Vibrio vulnificus</i> CMCP6	<i>ntrB</i> , <i>glnK</i> , <i>glnA</i>
<i>Xanthobacter autotrophicus</i> Py2	<i>dppA</i> , <i>nrtA</i> , <i>nrtA</i> , <i>nrtA</i> , <i>nrtA</i> , <i>urtA</i> , <i>glnK</i> , <i>uctA</i> , <i>nifR3</i> , <i>glnB</i>
<i>Xanthomonas axonopodis</i> pv. <i>citri</i> str. 306	<i>ntrB</i> , <i>glnA</i> , <i>glnA</i>
<i>Xanthomonas campestris</i> pv. <i>campestris</i> str. ATCC 33913	<i>glnA</i> , <i>ntrB</i>
<i>Xylella fastidiosa</i> 9a5c	<i>glnA</i> , <i>ntrB</i>
<i>Yersinia pestis</i> KIM	<i>glnA</i> , <i>glnA</i> , <i>ntrB</i> , <i>astC</i> , <i>glnK</i> , <i>gltI</i> , <i>hisJ</i> , EAM_0872

ANEXO II – CONJUNTO FALSO

Para integrar o conjunto de falsos-sítios de ligação à NtrC utilizamos 921 TFBS originados de diferentes fatores de transcrição. A TABELA 6, preenchida com informações do RegPrecise, reúne uma breve descrição e o número de sítios de ligação (TFBS) obtido para cada TF. Em seguida, a TABELA 7 reúne os organismos e genes do qual foram obtidos os sítios de ligação.

TABELA 6 - FATORES DE TRANSCRIÇÃO (TF) UTILIZADOS NO CONJUNTO DE FALSO-NTRC

Fator de Transcrição	Função	Qt. De TFBS Selecionados
FadR	Regula os genes de utilização de ácidos graxos em gamaproteobactérias	347
LldR	Regula os genes de utilização de lactato em Proteobactérias	75
MetR	Regula o metabolismo da metionina em Proteobactérias	71
NagQ	Regula os genes de utilização de quitina e N-acetilglucosamina em várias Proteobactérias	6
GguR	Regula genes envolvidos na captação e utilização de hexuronatos (glucuronato e galacturonato) e hexaratos (glucarato e galactarato) em várias Proteobactérias	155
GlcC	Regula os genes de utilização de glicolatos em Proteobactérias	28
Fur	Controla a homeostase do ferro em diversos grupos taxonômicos de bactérias	41
HexR	Foi implicado na regulação dos genes do metabolismo da glicose em <i>Pseudomonas putida</i>	188
FixJ	Controla uma adaptação às condições microaeróbicas	10

TABELA 7 - ORGANISMOS E GENES ONDE NÃO OCORRE LIGAÇÃO À NTRC

Organismo	Genes
Acidovorax avenae subsp. citrulli AAC00-1	gguR, tctC1, tctC1, udh, pykA, zwf
Acidovorax sp. JS42	gguR, tctC1, tctC1, gudD2, zwf, pykA
Alteromonadales bacterium TW-7	fadE, COG0596, fadI, tesB, fadE2, fadH, metR, metA, metF-II, glyA
Alteromonas macleodii 'Deep ecotype'	tesB, fadB, fadE, fadI, fadD, fadE2, hexR, hexR, zwf, zwf, glgP, ppsA, pykA, ppc, bkdA1, cpsA, pckA, gapA
Azospirillum sp. B510	murQ, nagQ, murQ, wecA, nagB2, nagB2
Azotobacter vinelandii AvOP	kdgD, gguR, gguR, glcD, glcD, glcC, glcC, lldR, lldR, lldP, lldP, metE, metR
Burkholderia cepacia AMMD	kdgD, kdgD, gudP, gguT, uxuP, gudD, kgsD, gguR, garD, garD, pgl, kgsD, exuT

<i>Burkholderia glumae</i> BGR1	uxuP, gudD, kdgD, uxuF, garD, udh, garD, gguR, kgsD, pgl, exuT, exuT, gudP, kdgD
<i>Burkholderia phymatum</i> STM815	garL, garL, gguR, garD, garP, garP, kgsD, kgsD
<i>Burkholderia</i> sp. 383	kdgD, kdgD, gudD, gudP, gguT, uxuP, garD, garD, kgsD, gguR, exuT
<i>Burkholderia vietnamiensis</i> G4	gudD, kdgD, kdgD, gudP, uxuF, mcp, uxuP, gguR, garD, garD, kgsD, pgl
<i>Burkholderia xenovorans</i> LB400	gudD, uxuP, kdgD, kdgD, gguR, kgsD, kgsD, garD, garD
<i>Chromobacterium violaceum</i> ATCC 12472	edd, zwf, edd, zwf, ptsHI, lldE, lldR
<i>Chromohalobacter salexigens</i> DSM 3043	edd, aceE, pykA, gapA, zwf, ppsA, aceA, hexR
<i>Citrobacter koseri</i> ATCC BAA-895	fadE, fadH, fadI, fadB, fabA, fadM, fadR, fabB, yebV, iclR, fadL, fadL, zwf, hexR, ybfA, pckA, lldP, lldP
<i>Colwellia psychrerythraea</i> 34H	fadI, acdB, fadH, zwf, zwf, hexR, gapA, fba, glyA, metR
<i>Comamonas testosteroni</i> KF-1	gguR, gudD2, tctC1, pgk, pykA
<i>Cupriavidus taiwanensis</i>	gudD, garD, garD, kdgD, gguR, edd, edd, hexR, hexR, lldR, lldE, metR, metE
<i>Dechloromonas aromatica</i> RCB	lldE, lldE, lldE, lldR, lldR, lldR
<i>Delftia acidovorans</i> SPH-1	gguR, tctC1, gudD2, pgk, pykA, zwf, pgi, pgi
<i>Desulfuromonas acetoxidans</i> DSM 684	fur, PF10087, feoA
<i>Edwardsiella tarda</i> EIB202	fabA, fabB, fadR, fadL, zwf, ppc, hexR
<i>Enterobacter</i> sp. 638	fadH, fadE, fadI, fadB, fabA, fadD, fadM, fadR, fabB, yebV, iclR, fadL, fadL, zwf, hexR, ybfA, pckA, lldP, lldP
<i>Erwinia amylovora</i> ATCC 49946	fadE, fadH, fabA, fadB, fadD, fabB, fadL, fadL
<i>Erwinia carotovora</i> subsp. <i>atroseptica</i> SCRI1043	fadE, fadH, fadI, fabA, fadD, fadB, fadM, fabB, fadL, zwf, ppc, ppc, hexR, ybfA
<i>Escherichia coli</i> str. K-12 substr. MG1655	fadE, fadH, fadI, fadB, fabA, iclR, fadM, fadR, fabB, yebV, fadL, fadL, fadD, glcC, glcD, glcC, glcD, zwf, hexR, ybfA, lldP, lldP
<i>Geobacter lovleyi</i> SZ	PF10087, feoB1, fur
<i>Geobacter metallireducens</i> GS-15	fur, Gmet_2833, feoB1, psp, psp
<i>Geobacter</i> sp. FRC-32	Gmet_2833, fur, feoB1, psp, psp
<i>Geobacter</i> sp. M21	fur, feoB1, PF04966, feoA1, Gmet_2833, psp, lldG, lldG, lldP, lldP
<i>Geobacter sulfurreducens</i> PCA	PF04966, GSU3274, fur, feoB1, lldP, lldP, lldG, lldG
<i>Geobacter uraniumreducens</i> Rf4	fur, feoB1, psp, psp, lldG, lldP, lldP, lldP, lldP, lldG

Glaciecola sp. HTCC2999	fadB, fadE, fadI, fadE2, gapA, hexR, gapA, ppsA, zwf, GHTCC_010100006532, pykA, pykA
Idiomarina baltica OS145	fadE, fadI, zwf, zwf, hexR, hexR
Idiomarina loihiensis L2TR	fadD, fadE, fadI
Klebsiella pneumoniae subsp. pneumoniae MGH 78578	fadE, fadH, fadI, fadD, fabA, fadB, fadM, iclR, fadR, fabB, fadL, fadL, yebV, zwf, hexR, ybfA, pckA, lldP, lldP
Laribacter hongkongensis HLHK9	lldE, lldR
Leptothrix cholodnii SP-6	garD, garD, kdgD, kdgD, udh, gguR, garD, zwf, edd, tal
Marinomonas sp. MWYL1	edd, edd, gapA, gapA, gapA, glk, glk, gpmM, gapB, ppsA, prpB, gltA, pckA, pckA, aceE
Methylbium petroleiphilum PM1	tctC1, tctC1, gguR
Neisseria meningitidis MC58	tal, zwf, zwf, zwf, edd, edd, edd, gntU, lldP, lldP, lldP, lldR
Pelobacter carbinolicus str. DSM 2380	fur, PF10087, Gmet_2833, feoA, feoA
Pelobacter propionicus DSM 2379	Gmet_2833, feoA2, PF10087, feoB1, fur, feoB1, glcD, lldR, lldP, lldP, lldG, lldG
Photobacterium profundum SS9	fabA, fadB, fadE2, fadH, fadL, fadI
Photorhabdus luminescens subsp. laumondii TTO1	fadE, fadH, fadI, fabA, fadD, fadB, iclR, fabB, yebV, fadR, fadL, fadL, zwf, hexR, ybfA
Polaromonas naphthalenivorans CJ2	gguR, udh, pykA, zwf, edd
Polaromonas sp. JS666	gguR, udh, zwf, pykA, edd, tal
Proteus mirabilis HI4320	fadE, fadH, fadB, fadI, fadD, fabB, fadR, fadL, fabA, zwf, ppc, ppc, hexR, ybfA
Pseudoalteromonas atlantica T6c	fadB, fadE, fadI, fadE2, acdB, hexR, zwf, ppsA, pykA, gapA, glgP, cpsA, gapB, tal
Pseudoalteromonas haloplanktis TAC125	fadE, fadI, tesB, fadH, fadE2, metR, metF-II, glyA, metA
Pseudoalteromonas tunicata D2	fadI, fadE, fadH, fadE2, tesB, glyA, metR, metR, metF-II, metF-II
Pseudomonas aeruginosa PAO1	glcC, glcD, glcD, glcC, lldR, lldP, meth, meth, metE, metR
Pseudomonas entomophila L48	kdgD, gguT, gguR, gguR, lldR, lldP, metF2, metF2, metF2, PF08908, PF08908, metE, metE, metR, metR, meth, dsbC
Pseudomonas fluorescens Pf-5	glcD, glcD, glcC, glcC, lldP, lldR, meth, PF08908, PF08908, metE, metR, metR
Pseudomonas mendocina ymp	kdgD, kdgD, udh, udh, aldE, gguR, gudD2, gudD2, glcD, glcD, glcC, glcC, lldR, lldP, lldP, lldR, meth, metE, metE, metR, metR

<i>Pseudomonas putida</i> KT2440	gguT, kdgD, udh, gguR, gguR, glcD, glcD, glcC, glcC, lldP, lldR, metF2, metE2, metF2, metE2, dsbC, metR, PF08908, PF08908, metR, metH
<i>Pseudomonas stutzeri</i> A1501	glcC, glcD, glcD, glcC, lldP, lldP, lldR, lldR, metH, metE, metR
<i>Pseudomonas syringae</i> pv. <i>tomato</i> str. DC3000	kdgD, kgsD, udh, exuT, exuT, metE, metR, metH
<i>Ralstonia eutropha</i> H16	gudD, garD, garD, kdgD, kdgD, gguR, edd, edd, hexR, hexR, lldR, lldE, metR, metE
<i>Ralstonia eutropha</i> JMP134	gudD, tctC1, tctC1, gudP1, gudP1, gguR, kdgD, kgsD, edd, edd, hexR, hexR, lldR, lldE, metR, metE
<i>Ralstonia metallidurans</i> CH34	gudD, tctC1, tctC1, gudP1, kgsD, kgsD, gguR, kdgD, edd, edd, hexR, hexR, lldR, lldE, metR, metE, gudD, gguR, edd, edd, zwf, zwf, metR, metE
<i>Ralstonia solanacearum</i> GMI1000	gudD, gguR, edd, edd, zwf, zwf, metR, metE
<i>Reinekea</i> sp. MED297	oxIT, hexR, glk, glk, gpmM, gapB, ugpC, pgk, eno, pflA, pykF, adhE, gapA, gapA, gltA, pckA, pckA
<i>Rhizobium etli</i> CFN 42	fixKf, ccoN
<i>Rhizobium leguminosarum</i> bv. <i>viciae</i> 3841	fixKf, fixJf, ccoN, fixKf, fixKf, fixJf
<i>Rhodoferax ferrireducens</i> DSM 15236	pykA, zwf, tal, edd
<i>Salmonella typhimurium</i> LT2	fadE, fadH, fadI, fadB, fadM, fabA, iclR, fadR, fabB, yebV, fadL, fadL, fadD, zwf, hexR, ybfA, pckA, lldP, lldP
<i>Serratia proteamaculans</i> 568	fadE, fadH, fadI, fadB, iclR, fadD, fadM, fadR, fabB, yebV, fadL, fadL, ppc, zwf, ppc, hexR, ybfA
<i>Shewanella amazonensis</i> SB2B	fadL, fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella baltica</i> OS155	fadL, fadL, fadI, SO4716, SO0572, fadR, fade
<i>Shewanella denitrificans</i> OS217	fadL, fadI, fadR, fadE, SO0572
<i>Shewanella frigidimarina</i> NCIMB 400	fadL, fadI, SO0572, fadR, fade
<i>Shewanella halifaxensis</i> HAW-EB4	fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella loihica</i> PV-4	fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella oneidensis</i> MR-1	fadL, fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella pealeana</i> ATCC 700345	fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella piezotolerans</i> WP3	fadL, fadI, SO4716, fadR, fadE, SO0572
<i>Shewanella putrefaciens</i> CN-32	fadL, fadL, fadI, SO4716, SO0572, fadR, fade
<i>Shewanella sediminis</i> HAW-EB3	fadL, fadI, fadR, fadE, SO0572

Shewanella sp ANA-3	fadL, fadL, fadI, SO4716, SO0572, fadR, fadE
Shewanella sp MR-4	fadL, fadL, fadI, SO4716, SO0572, fadR, fadE
Shewanella sp MR-7	fadL, fadL, fadI, SO4716, SO0572, fadR, fadE
Shewanella sp W3-18-1	fadL, fadL, fadI, SO4716, SO0572, fadR, fadE
Shewanella woodyi ATCC 51908	fadL, fadI, SO0572, SO4716, fadR, fadE
Sinorhizobium meliloti 1021	fixJf, fixKf
Stenotrophomonas maltophilia K279a	lldP, lldP
Thauera sp. MZ1T	Tmz1t_1714
Variovorax paradoxus S110	gguR, tctC1, tctC1, udh
Verminephrobacter eiseniae EF01-2	hexR, zwf, pgi
Vibrio angustum S14	fabA, fabA, fadB, fadE2, fadH, fadL, fadI
Vibrio cholerae O1 biovar eltor str. N16961	fadL, fabA, fadB, plsB, fadE2, VC2105, fadH
Vibrio fischeri ES114	fadL, fadB, plsB, fadE2, VC2105, fadH, fadI
Vibrio harveyi ATCC BAA-1116	fabA, fabA, fadB, plsB, fadE2, VC2105, fadH, fadL, fadI
Vibrio parahaemolyticus RIMD 2210633	fabA, fabA, fadB, plsB, fadE2, VC2105, fadH, fadI
Vibrio salmonicida LF11238	fabA, fabA, fadB, plsB, fadE2, VC2105, fadH, fadL, fadI
Vibrio shilonii AK1	fadB, plsB, fadE2, VC2105, fadH, fadL, fadI
Vibrio splendidus LGP32	fabA, fabA, fadB, plsB, fadE2, fadH, fadL, fadI
Vibrio vulnificus CMCP6	fabA, fabA, fadB, plsB, fadE2, fadE2, fadH, fadL, fadI
Yersinia pestis KIM	fadE, fadH, fadI, fadD, fadB, fadM, fabB, yebV, fadR, iclR, fadL, fadL, zwf, zwf, hexR, hexR, ybfA

ANEXO III – ANÁLISE COM STRING

TABELA 8 - ANÁLISE DOS ENRIQUECIMENTOS ENCONTRADOS PELO STRING: PROCESSOS BIOLÓGICOS (GO)

#term ID	term description	observed gene count	background gene count	false discovery rate	matching proteins
GO:0043604	amide biosynthetic process	16	171	0.0178	foIE,panC,panD,pta,rplA,rplC,rplE,rplF,rplJ,rplK,rplL,rplM, rplN,rpsC,rpsK,thrS
GO:0043603	cellular amide metabolic process	17	206	0.0218	foIE,panC,panD,pta,rplA,rplC,rplE,rplF,rplJ,rplK,rplL,rplM, rplN,rpsC,rpsK,tesB,thrS
GO:0006412	translation	12	116	0.0279	rplA,rplC,rplE,rplF,rplJ,rplK,rplL,rplM,rplN,rpsC,rpsK,thrS
GO:0000027	ribosomal large subunit assembly	5	27	0.0460	rplA,rplC,rplE,rplF,rplK
GO:0006518	peptide metabolic process	12	139	0.0460	rplA,rplC,rplE,rplF,rplJ,rplK,rplL,rplM,rplN,rpsC,rpsK,thrS
GO:0006616	SRP-dependent cotranslational protein targeting to membrane, translocation	3	3	0.0460	secE,secG,secY
GO:0006820	anion transport	13	190	0.0460	aroP,flc,focA,glnH,glnP,glti,glfP,gntP,hisJ,osmF,yehW, yehY,yjeH
GO:0006865	amino acid transport	10	94	0.0460	aroP,glnH,glnP,glti,glfP,hisJ,osmF,yehW,yehY,yjeH
GO:0009399	nitrogen fixation	3	3	0.0460	glnA,glnG,glnL
GO:0015696	ammonium transport	5	30	0.0460	amtB,osmF,potG,yehW,yehY
GO:0017148	negative regulation of translation	5	25	0.0460	rmf,rplA,rplJ,rplM,thrS
GO:0032268	regulation of cellular protein metabolic process	7	59	0.0460	ihfB,rmf,rplA,rplJ,rplM,thrS,yfhM
GO:0032269	negative regulation of cellular protein metabolic process	6	30	0.0460	rmf,rplA,rplJ,rplM,thrS,yfhM
GO:0032978	protein insertion into membrane from inner side	3	4	0.0460	secE,secG,secY
GO:0042254	ribosome biogenesis	9	94	0.0460	rluC,rplA,rplC,rplE,rplF,rplJ,rplK,rpsC,rpsK
GO:0042255	ribosome assembly	7	49	0.0460	rplA,rplC,rplE,rplF,rplK,rpsC,rpsK
GO:0043952	protein transport by the Sec complex	3	5	0.0460	secE,secG,secY
GO:0046942	carboxylic acid transport	12	132	0.0460	aroP,focA,glnH,glnP,glti,glfP,gntP,hisJ,osmF,yehW,yehY,yjeH
GO:0071705	nitrogen compound transport	17	261	0.0460	amtB,aroP,dppA,glnH,glnP,glti,glfP,hisJ,osmF,potG,secE, secG,secY,yegT,yehW,yehY,yjeH

GO:0071941	nitrogen cycle metabolic process	5	31	0.0460	glnA,glnG,glnL,nac,narP
GO:0006417	regulation of translation	6	48	0.0471	ihfB,rmf,rplA,rplJ,rpIM,thrS
GO:0071229	cellular response to acid chemical	3	8	0.0471	ycgZ,yjeH,yobF