Albaro Ramon Paiva Sanz

APRENDIZAGEM DE MÁQUINA PARA CLASSIFICACÃO DE ESTRUTURAS EXON E INTRON EM DADOS DE GENOMA HUMANO

Recife-PE

Fevereiro/2019

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA

APRENDIZAGEM DE MÁQUINA PARA CLASSIFICACÃO DE ESTRUTURAS EXON E INTRON EM DADOS DE GENOMA HUMANO

Tese julgada adequada para obtenção do título de Doutor em Biometria e Estatística Aplicada, defendida e aprovada em 27/02/2019 pela comissão examinadora

Área de concentração: Modelagem e Métodos Computacionais

Orientador: Dr. Tiago Alessandro Espínola Ferreira

Recife-PE

Fevereiro/2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

S238a Sanz, Albaro Ramon Paiva.
Aprendizagem de máquina para classificação de estructuras Exon e Intron em dados de genoma humano / Albaro Ramon Paiva.Sanz. – Recife, 2019.
79 f.: il.

Orientador(a): Tiago Alessandro Espínola Ferreira.
Tese (Doutorado) – Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Biometria e Estatística Aplicada, Recife, BR-PE, 2019.
Inclui referências.

1. Classificação 2. Intro-exon 3. Algoritmo de computador I. Ferreira, Tiago Alessandro Espínola, orient. II. Título

CDD 310

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOMETRIA E ESTATÍSTICA APLICADA

APRENDIZAGEM DE MÁQUINA PARA CLASSIFICACÃO DE ESTRUTURAS EXON E INTRON EM DADOS DE GENOMA HUMANO

Albaro Ramon Paiva Sanz

Tese julgada adequada para obtenção do título de Doutor em Biometria e Estatística Aplicada, defendida e aprovada em 27/02/2019 pela comissão examinadora

Orientador:

Dr. Tiago Alessandro Espínola Ferreira
Orientador

Banca examinadora:

Dr.Moacyr Cunha Filho
Universidade Federal Rural de Pernambuco

Dr.Valdir Queiroz Balbino
Universidade Federal de Pernambuco

Dr. Antônio de Pádua Santos
Universidade Federal Rural de Pernambuco

Dr.Péricles Barbosa nha de Miranda
Universidade Federal Rural de PernambuCuco

Aos meus pais.

Agradecimentos

Em primeiro lugar, agradeço a Deus e à minha família por entender minha ausência em casa.

Agradeço ao professor Dr. Tiago A. E. Ferreira pelo apoio nesse trabalho, orientação e paciência.

Agradeço aos meus colegas do PPGBEA pela acolhida, por todos os conselhos e sua amizade ao longo desses anos. A Rodrigo, Edy Jonas, Fábio, Isabelly, Kerolly, Filipe, Neidinha, Eucymara, Nathielly, Herica, Leda, Dalton, Ikaro, Luisa, Glauce, Gutemberg, Rivelino, Carlos Renato e aos demais colegas do departamento por toda convivência e ajuda.

Aos professores e funcionários do PPGBEA, em especial, aos professores Tiago, Paulo, Cláudio, Tatjana, Borko, Guilherme, Moacyr e Samuel. Aos secretários Marco, Eduardo e Edivânia. Ao pessoal dos serviços gerais.

A CAPES pelo apoio financeiro.

Agradeço a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

Resumo

As técnicas de classificação são frequentemente utilizadas na solução de diferentes problemas da bioinformática. A maioria dos genes na sequência do DNA é transcrita pelo RNA mensageiro e traduzida para proteína. O DNA contém regiões que codificam as proteínas chamadas *exons*, e regiões que não codificam as proteinas são chamadas de *introns*, os limites entre os *exons* e *introns* são chamados de *splice site*. Durante o processo de transcrição, os *introns* são "cortados", isso é conhecido como *splicing*, que coloca os *exons* de um gene um atrás do outro consecutivamente, prontos para serem traduzidos na sequência de aminoácidos que compõem a proteína. Nos *splice sites*, a transição da região codificante *exon* para a região não codificante *intron* (EI) é distinguida com os nucleótidos GT, e a transição da região não codificante *intron* para a região codificante *exon* (IE) é distinguida com os nucleótidos AG. Uma pequena porcentagem dessas combinações são *splice sites* reais. Neste estudo, é apresentada uma metodologia para o problema de classificação EI e IE que consistem em obter distribuições de probabilidades usando técnicas de aprendizagem de máquina, e a partir delas obter diferentes medidas de desempenho. Uma série de algoritmos (*Support Vector Machine* (SVM), *Neural Networks* (RNA), *Random Forest* (RF), *Naive Bayes*(NB)) foram testados e comparados para encontrar o melhor classificador. Para fazer a seleção do melhor classificador, as medidas mais conhecidas, foram aplicadas com base na matriz de confusão: Acurácia, Especificidade, Sensibilidade, dentre outros, bem como a distância de Kolgomorov-Smirnov (KS) como medida de desempenho dos modelos de classificação. Mais precisamente, a KS é uma medida do grau de separação entre as distribuições de classe de probabilidade, sendo este um indicativo de maior acurácia. Os resultados apresentados neste estudo foram iguais ou superiores em acurácia quando são comparado com os trabalhos apresentados na literatura.

Palavras-chaves: classificação. intron-exon. exon-intron. algoritmo de aprendizagem de máquinas.

Abstract

Classification techniques are often used to solve different bioinformatics problems. Most genes in the DNA sequence are transcribed by messenger RNA and translated into protein. The DNA contains regions that encode proteins (exons) and regions that do not encode proteins (introns), the boundaries between exons and introns are called the splice site. During the transcription process, the introns are "cut", this is known as splicing that puts the exons of a gene consecutively, ready to be translated into the amino acid sequence that make up the protein. In splice sites, the transition from the coding region exon to the non-coding region intron (EI) and distinguished with the nucleotides GT, and transition from the non-coding region (intron) to the coding region exon (IE) and distinguished with the nucleotides AG. A small percentage of these combinations are actual splice sites. In this study, a methodology for the classification problem EI and IE is presented, which consists in obtaining probability distributions using machine learning technique and starting from them to obtain different measures of performance. A number of algorithms (Support Vector Machine (SVM), Artificial Neural Network (RNA), Random Forest (RF), Naive Bayes (NB)) are tested and compared to find the best classifier. To make the selection of the best classifier the most known measures are applied based on the confusion matrix: Accuracy, Specificity, Sensitivity, among others, as well as the Kolgomorov distance (KS) as measured performance of the classification models. More precisely, the KS is a measure of the degree of sep aration between the distributions of probability class, which is an indication of greater accuracy. The results presented in this study are equal or superior in accuracy when compared with the papers presented in the literature Classification

Key-words: classification. intron-exon.exon-itron machine learning algorithm.

Lista de Figuras

Figura 1 – Representação do *splice sites* na sequência do DNA 1
Figura 2 – Estrutura em dupla hélice de uma molécula de DNA 6
Figura 3 – Do DNA à proteína: no núcleo, o DNA é transcrito para RNA pré-mensageiro. Este, com o processamento (em várias etapas), torna-se um RNA mensageiro maduro, e traduzida para proteínas 7
Figura 4 – No processo de *splicing*, fatores específicos (representados por tesouras) reconhecem os locais de corte do DNA, eliminando os *introns* (partes sem informação gênica – em laranja)e colando os *exons* (partes com informação gênica – em roxo) adjacentes, formando a fita de RNA mensageiro 8
Figura 5 – Classificador bayesiano ingênuo 11
Figura 6 – Hiperplanos paralelos e vetores de suporte em R^2 13
Figura 7 – Conjunto de hiperplano separador em R^2 13
Figura 8 – Modelo de um neurônio artificial. (HAYKIN, 1994) 18
Figura 9 – Rede Alimentadas Adiante com Camada Única. 19
Figura 10 – Redes Alimentadas Diretamente com Múltiplas Camadas 20
Figura 11 – Exemplo da predição de uma instância X aplicada ã cada árvore da RF 21
Figura 12 – Arquitetura da metodologia proposta. 22
Figura 13 – Exemplo do 5-fold validação cruzada 28
Figura 14 – Probabilidade condicional de classe e acumulada para duas classes . . . 29
Figura 15 – Representação do índice Youden e o ponto de corte ótimo (c) 30
Figura 16 – KS obtido a partir do escores das distribuções EI, N, para os algoritmos RF e NB 32
Figura 17 – KS obtido a partir das distribuções EI, N, para os algoritmos SVM e RNN 33
Figura 18 – Distribuções IE, N, para os algoritmos RF e SVM 38
Figura 19 – Distribuções IE, N, para os algoritmos NB e RNN 38

Lista de tabelas

Tabela 1 – Resumo do conjunto de dados usado no experimento 25
Tabela 2 – Matriz de confusão para classificação de duas classes 27
Tabela 3 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI. 34
Tabela 4 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI (Continuação). 35
Tabela 5 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI (Continuação). 36
Tabela 6 – Comparação de desempenho dos métodos propostos com outros métodos com dados UCI. 37
Tabela 7 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI. 40
Tabela 8 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI (Continuação). 41
Tabela 9 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI (Continuação). 42
Tabela 10 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D. 44
Tabela 11 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D (Continuação). 45
Tabela 12 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D (Continuação). 46
Tabela 13 – Comparação de desempenho dos métodos propostos com outros métodos EI-N com dados HS3D. 47
Tabela 14 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados. 49
Tabela 15 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados HS3D (Continuação). 50
Tabela 16 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados HS3D (Continuação). 51
Tabela 17 – Comparação de desempenho dos métodos propostos com outros métodos IE-N com dados HS3D 52

Tabela 18 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21. 54
Tabela 19 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21 (Continuação). 55
Tabela 20 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21 (Continuação). 56

Sumário

1 Introdução . . . 1
1.1 Motivação . . . 3
1.2 Objetivo geral . . . 4
1.2.1 Objetivos específicos . . . 4
1.3 Estrutura do documento . . . 4
2 Conceitos Biológicos . . . 5
2.1 Genética, gene e proteínas: um dogma central na biologia . . . 5
2.2 Expressão Gênica . . . 6
3 Revisão de Literatura . . . 9
3.1 Aprendizado de Máquina . . . 10
3.2 Naive Bayes . . . 11
3.3 Máquinas de suporte vetorial . . . 12
3.3.1 Modelos Lineares de Suporte a Vetores . . . 13
3.4 Redes Neurais . . . 16
3.4.1 O Neurônio Artificial . . . 17
3.4.2 Estruturas de RNAs . . . 18
3.4.2.1 Redes Alimentadas Adiante com Camada Única . . . 18
3.4.2.2 Redes Alimentadas Diretamente com Múltiplas Camadas . . . 19
3.5 Random Forest (Floresta Aleatória) . . . 20
4 Materiais e métodos . . . 22
4.1 Método de codificação . . . 22
4.2 Dados utilizados . . . 24
4.3 Métrica para a avaliação do classificador . . . 26
4.4 Matriz de confusão . . . 26
4.5 Validação cruzada . . . 28
4.6 Distância de Kolmogorov-Smirnov (KS) . . . 29
4.7 Índice de Youden . . . 29
4.8 Razão verossimilhança . . . 30
5 Resultados e Discussão . . . 32
5.1 Conjuto de dados UCI (Newman 1998) . . . 32
5.2 Conjuto de dados HS3D (Pollastro e Rampone 2003) . . . 43
5.3 Conjuto de dados cromossomo 21, segmento HS21C103 (CR21) . . . 53

6 Conclusões 57
REFERÊNCIAS BIBLIOGRÁFICAS 58

1 Introdução

O sequenciamento de alto rendimento dos genomas revolucionou a biologia nos últimos anos, gerando novas informações que mudaram dramaticamente nosso conhecimento sobre centenas de espécies, incluindo os seres humanos (SALZBERG, 2007). A última década foi marcada por um número significativo de realizações no campo da genética. Lawson e Falush, (2012), como um novo braço científico derivado de projetos de sequenciamento de genomas inteiros. Iniciativas como o Projeto Genoma Humano[1], estabeleceram as bases para o avanço da genética molecular para estudos globais capazes de medir sinais de milhares de genes ao mesmo tempo, mantendo a identificação de cada gene com base em sequências específicas de ácido desoxirribonucléico (DNA) (MARTINEZ, 2012). Devido ao aumento dos dados da sequência genômica, existe uma demanda urgente para melhorar a eficiência dos algoritmos computacionais (SONNENBURG et al., 2007). A medida que mais sequências do genoma inteiro são geradas, com o desenvolvimento contínuo de novos métodos para o sequenciamento de DNA, a identificação de genes torna-se uma das tarefas importantes para a biologia computacional. Para entender como o genoma funciona, precisamos identificar um conjunto de fragmentos de codificação de proteínas conhecidos como *exons*, que são separados por fragmentos intermediários não-codificadores de proteínas, conhecidos como *introns* (Figura 1). Os limites entre os exons e introns são chamados de *splice sites* (SS) (BARI; REAZ; JEONG, 2014). A identificação precisa de *splice sites* desempenha um papel fundamental nas anotações de genes em eucariotos (BATEN et al., 2007; RÄTSCH et al., 2007). A maioria dos genes codificadores de proteínas eucarióticas são genes compostos de *exons* e *introns* (BARI; REAZ; JEONG, 2014).

Figura 1 – Representação do *splice sites* na sequência do DNA (PASHAEI; OZEN; AYDIN, 2017).

[1] Projeto Genoma Humano: *http://www.ghente.org/ciencia/genoma/*

Com o uso de computadores tornou-se possível capturar, armazenar, analisar e integrar informações médicas ou biológicas de maneira mais fácil. A bioinformática representa uma crescente área de pesquisa que pode ser definida como uma área de conhecimento que busca o desenvolvimento e aplicações de técnicas e ferramentas para o armazenamento, organização e análise da informação biológica. Pertence a um campo multidisciplinar de pesquisa que inclui áreas tão diversas como ciência da computação, bioquímica, matemática e estatística (MARTINEZ, 2012).

Nos últimos anos, a bioinformática concentrou grande atenção no projeto Genoma Humano, integrando o estudo da informação genética, estruturas moleculares e funções bioquímicas. As enormes quantidades de dados usados em experimentos biológicos atuais levaram a criação de muitos bancos de dados contendo genomas, sequências de proteínas e outros tipos de dados biológicos. Em consequência, a pesquisa básica em Bioinformática é orientada para o desenho e implementação de sistemas que permitam resolver problemas de armazenamento, processamento e análise de grandes quantidades de dados relativos ao DNA (CERVANTES; LI; YU, 2009).

Um problema importante na bioinformática é a identificação de genes dentro de grandes regiões de sequências de DNA RÄTSCH et al., (2007). O reconhecimento de genes não é um desafio fácil devido a grande quantidade de sequências intergênicas e ao fato de que regiões codificadoras de proteínas (*exons*) podem ser interrompidas por regiões que não codificam proteínas (*introns*) (CERVANTES; LI; YU, 2009). Em alguns organismos, os *introns* são poucos e os locais de *splicing* estão bem caracterizados, no entanto, em algumas outras sequências, incluindo o genoma humano, um grande problema é localizar a transição correta entre as regiões codificantes e as que não codificam, os genes em muitos organismos se dividem de maneiras diferentes, dependendo do tipo de tecido ou fase de desenvolvimento, complicando consideravelmente a tarefa de reconhecimento. Para identificar genes, precisamos identificar dois tipos de *splice sites*: *Exon-Intron* (EI) e *Intron-Exon* (IE). Esses *splite sites* exibem determinados padrões de características, os *splite sites* EI começam com pares de base GT enquanto os *splite sites* IE terminam com pares com base AG. No entanto, nem todos os *splite sites* com pares de base GT ou AG são necessariamente *splice sites* (Possivelmente menos de um por cento). (CERVANTES; LI; YU, 2009; PASHAEI; OZEN; AYDIN, 2017).

Vários métodos computacionais foram desenvolvidos para o reconhecimento do *splice site*. Esses métodos são baseados na abordagem probabilística (BURGE; KARLIN, 1997; PERTEA; LIN; SALZBERG, 2001; STADEN, 1984; YEO; BURGE, 1984; ZHANG; MARR, 1984), Modelo Oculto de Markov (MAJI; GARG, ZHANG et al., 2010; PASHAEI; OZEN; AYDIN, 2010), Árvores de decisão (LOPES; LIMA; MURATA, 2007), Máquina de

Vetores de Suporte (SVM) (MEHER et al., 2016; BARI; REAZ; JEONG, 2014; GOEL; SINGH; ASERI, 2014)Rede Bayesiana (CHEN; LU; LI, 2004), Floresta Aleatória (RF) (MEHER et al., 2016;PASHAEI; OZEN; AYDIN, 2016; PASHAEI et al., 2016; MANDAL, 2015), Rede Neural Artificial (RNN) (BIN; JING, 2014; HO; RAJAPAKSE, 2003; NASSA; SINGH; GOEL, 2013) e assim por diante.

1.1 Motivação

A bioinformática é um campo emergente da pesquisa que utiliza ferramentas computacionais para o armazenamento, análise e apresentação de dados biológicos e moleculares. O envolvimento de técnicas computacionais, especialmente o desenvolvimento de algoritmos eficientes torna-se indispensável para uma boa análise dos dados gerados. A identificação de genes dentro de grandes regiões de sequências de DNA é um grande problema na bioinformática. O problema *splie site* (SS) é definido como um problema de detecção e classificação de limites entre *exon* e *introns*. A transição da região codificante (*exon*) para a região não codificante (*intron*) é distinguida com os nucleótidos GT, enquanto a transição da região não codificante (*intron*) para a região codificante (*exon*) é distinguido com os nucleótidos AG. No entanto, apenas cerca de 1% de todos os dinucleótidos GT e AG são verdadeiros *splite site* na sequência do genoma, o que levou a detecção do *splite site* a ser considerada uma das tarefas mais importantes da bioinformática (Pashaei, Ozen e Aydin 2017, Pashaei et al. 2016). A transição da região codificante para região não codificante (*exon-intron*), e da região não codificante para região codificante (*intron-exon*), pode ser abordado como um problema de classificação binária.

Neste sentido, a pesquisa aqui proposta abordará o problema de classificação dos *intron-exon* e exon-intron em sequências de DNA, baseados em aprendizagem estatística (Cortes e Vapnik 1995) e inteligência computacional (Duda, Hart e Stork 2012), apresentando uma metodologia, que consiste em obter distribuições de probabilidade (com base em aprendizagem estatístico), e a partir delas, obter diferentes medidas de desempenho conhecidas com base na matriz de confusão: Acurácia, Especificidade, Sensibilidade, Razão de Verossimilhança, Índice de Youden, dentre outros, bem como a distância de Kolgomorov-Smirinov (KS) como medida de desempenho dos modelos de classificação. Mais precisamente, a KS é uma medida do grau de separação entre as distribuições de classe de probabilidade.

1.2 Objetivo geral

O objetivo geral desta pesquisa é a construção de distribuições de classes em problemas binários com a utilização de algoritmo de aprendizagem de máquina, abordando o problema da classificação de *introns* e *exons* em sequências de DNA.

1.2.1 Objetivos específicos

- Avaliar algoritmos de aprendizagem de máquina, para classificação binária em sequências de DNA
- Medir o nível de discriminação da sequência de DNA.
- Selecionar o algoritmo de melhor desempenho, de acordo com sua discriminação e acurácia.

1.3 Estrutura do documento

O documento está disposto em cinco capítulos que explicam o trabalho realizado. No capítulo 1 é apresentada uma breve introdução, acompanhada da motivação e objetivos da investigação. A seguir, no capítulo 2, é apresentada uma breve descrição dos conceitos biológicos a serem usados, no capítulo 3 é apresentada a revisão da literatura, assim com uma breve descrição dos algoritmos das máquinas de aprendizagem. Posteriormente, no capítulo 4, a metodologia proposta para a construção de modelos de classificação é apresentada, e as medidas usadas para a avaliação do desempenho dos algoritmos de classificação, e nos capítulos 5 e 6 discutem-se os resultados obtidos, bem como as conclusões deste trabalho.

2 Conceitos Biológicos

2.1 Genética, gene e proteínas: um dogma central na biologia

A genética o um ramo da biologia que estuda os genes, que são as unidades orgânicas básicas, que contém informações sobre as características físicas e comportamentais de um organismo, e são passadas de geração para geração (GRIFFITHS et al., 2006).

Um nucleotídeo é uma molécula composta por um açúcar chamado pentose, um grupo fósfato e uma base nitrogenada (GRIFFITHS et al., 2006). Nucleotídeos são diferenciados pela base nitrogenada que os compõem, que podem ser: Adenina, Citosina, Guanina, Timina ou Uracila. A pentose de um nucleotídeo pode se ligar ao grupo fosfato de um outro nucleotídeo, formando, assim, uma cadeia de nucleotídeos. Uma cadeia formada por vários nucleotídeos é chamada de polinucleotídeo. Existem dois tipos de polinucleotídeos, os que armazenam informações genéticas: o DNA (ácido desoxirribonucleico) e o RNA (ácido ribonucleico)(SADAVA; PURVES, 2009). Suas estruturas são representadas na Figura 2.

De acordo com o modelo de dupla hélice proposto em 1953 por Watson e Crick, a estrutura de cada molécula de DNA consiste em duas cadeias de nucleotídeos que se unem, adquirindo uma configuração de dupla hélice. O DNA é formado por duas fitas de nucleotídeos, e a pentose que o constitui é a desoxirribose. As duas fitas do DNA são unidas através de pontes de hidrogênio formadas entre as suas quatro bases nitrogenadas. A adenina sempre forma pontes de hidrogênio com a timina, e a citosina com a guanina. As duas fitas de DNA são ditas complementares, e sempre é possível construir uma fita a partir da outra. A sequência de bases nitrogenadas ao longo da cadeia de DNA constitui a informação genética (GRIFFITHS et al., 2006).

O RNA é composto por apenas uma fita e sua pentose é a ribose. A base nitrogenada timina, exclusiva do DNA, é substituída pela uracila, exclusiva do RNA. Uma fita de RNA pode se dobrar de modo que parte de suas próprias bases nitrogenadas se pareiam umas com as outras, esse pareamento intramolecular é um fator importante no formato tridimensional do RNA, que é capaz de assumir uma variedade maior de formas complexas do que a dupla hélice de DNA (GRIFFITHS et al., 2006).

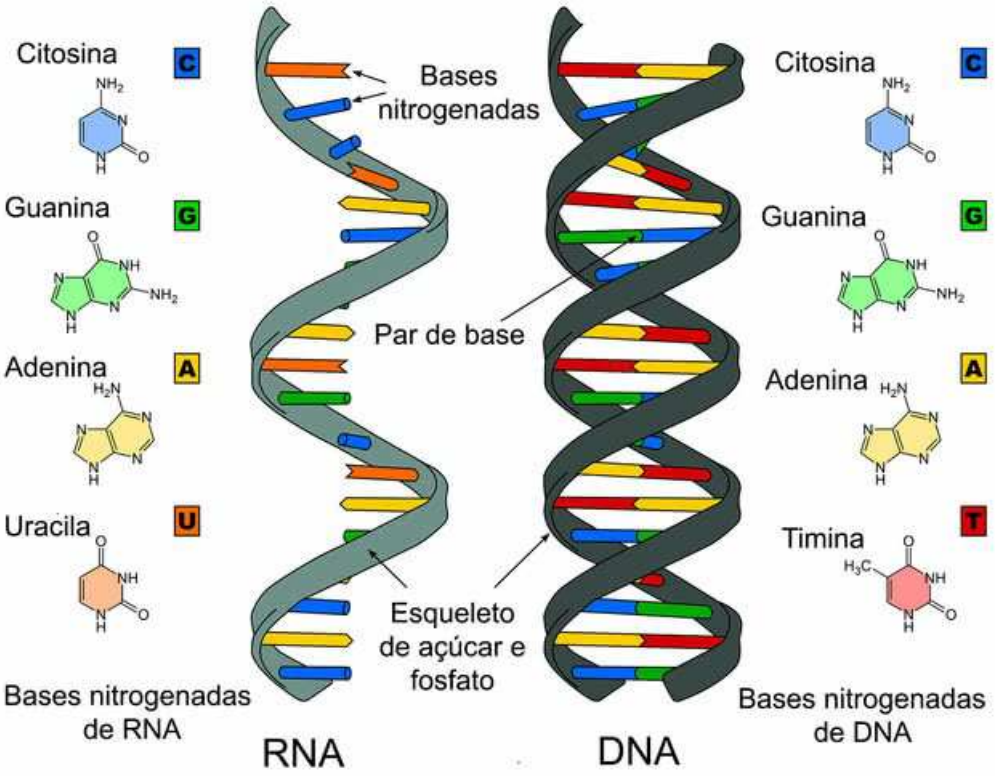

Figura 2 – Estrutura em dupla hélice de uma molécula de DNA

2.2 Expressão Gênica

Para produzir proteínas a partir do código genético, acontece uma série de transformações bioquímicas conhecidas como expressão gênica. Uma sequência de bases de DNA é lida, e os segmentos do código são traduzidos em aminoácidos que formarão uma proteína (GRIFFITHS et al., 2006).

Cada célula de um ser vivo contém um ambiente dinâmico constituído por moléculas, reações químicas e uma cópia do genoma do organismo. Embora o DNA seja exatamente o mesmo em todas as células de um dado organismo, a expressão desse código pode ser radicalmente diferente em cada uma delas, originando, desta forma, células especializadas em diferentes órgãos do organismo. É no núcleo da célula que se encontra o DNA, onde se escreve o código da vida (LEWIN, 2003). Dentro da célula, o DNA é organizado numa estrutura chamada cromossomo e o conjunto de cromossomos de uma célula forma o cariótipo. Antes da divisão celular os cromossomos são duplicados através de um processo chamado replicação. Os organismos eucariontes, tais como, animais, plantas e fungos têm o seu DNA dentro do núcleo enquanto os procariontes, como as bactérias, o tem disperso

no citoplasma (LEWIN, 2003).

Nos seres eucariontes, cujas células têm núcleo, o material genético concentra-se nesse compartimento. Toda a informação para o funcionamento do ser está contida em uma longa molécula composta por duas cadeias (ou fitas) complementares e antiparalelas, o ácido desoxirribonucléico (DNA). Examinando mais de perto, pode-se ver que o DNA contém uma sucessão de quatro tipos diferentes de bases aminadas: adenina (A), timina (T), citosina (C) e guanina (G), e que cada gene é formado por uma determinada sequência dessas bases (PENALVA; ZORIO, 2001).

Em uma regra quase geral, essas sequências (ou seja, os genes) incluem alguns segmentos chamados *exons*, com informação que as células usarão na produção de proteínas, e segmentos intercalados, os *introns*, cuja informação será descartada, não sendo usada nesse processo celular (PENALVA; ZORIO, 2001). Para que a informação contida no DNA seja transferida para as diversas funções celulares, essa molécula precisa primeiro ser copiada, em um processo denominado transcrição, que gera uma molécula-irmã, o ácido ribonucléico (RNA),mostrado Figura 3.

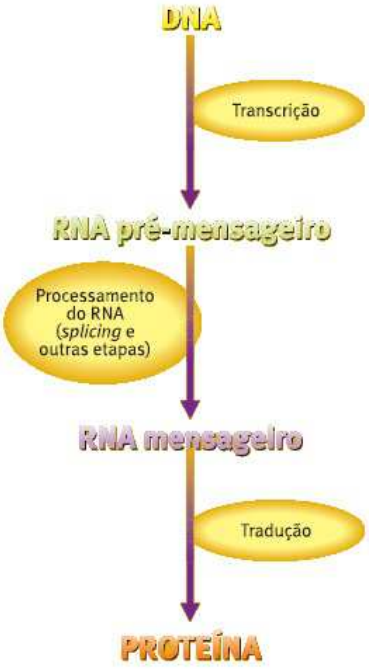

Figura 3 – Do DNA à proteína: no núcleo, o DNA é transcrito para RNA pré-mensageiro. Este, com o processamento (em várias etapas), torna-se um RNA mensageiro maduro, e traduzida para proteínas

O RNA contém a mesma informação genética presente no DNA, mas difere deste por ter uma só cadeia e por apresentar, no lugar da base timina (T), a uridina (U). No entanto, o RNA só se torna funcional depois que os *introns* são retirados, em um processo conhecido como *splicing* (esse é o termo usado internacionalmente, que pode ser traduzido, em português, por cortar e ligar.). O *splicing*, um dos fenômenos biológicos mais conservados ao longo da evolução, está presente em praticamente todos os seres vivos com núcleo (eucariontes). É um mecanismo altamente complexo e estruturado, que envolve a ligação ao RNA, de uma série de componentes celulares. Em uma explicação simplificada, pode-se dividir o *splicing* em duas etapas: a retirada dos *introns* e a ligação dos *exons*, como mostrado na Figura 4 (PENALVA; ZORIO, 2001).

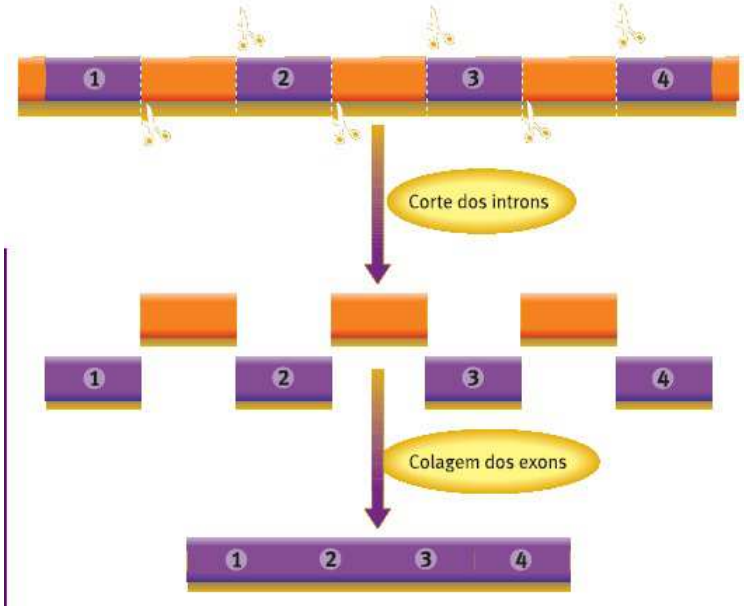

Figura 4 – No processo de *splicing*, fatores específicos (representados por tesouras) reconhecem os locais de corte do DNA, eliminando os *introns* (partes sem informação gênica – em laranja)e colando os *exons* (partes com informação gênica – em roxo) adjacentes, formando a fita de RNA mensageiro

Uma das etapas-chave no processo é o reconhecimento preciso das junções entre os *introns* e os *exons* (os locais de *splice site*). Isso é possível porque, perto dessas junções, existem sequências altamente conservadas (sequências consenso) que servem como sítios de união para componentes celulares responsáveis pelo processo de *splicing* (PENALVA; ZORIO, 2001). A regra canônica GT-AG, indica as sequências conservadas, a transição da região codificante (*exon*) para a região não codificante (*intron*) é distinguida com os nucleótidos GT, enquanto a transição da região não codificante (*intron*) para a região codificante (*exon*) é distinguido com os nucleótidos AG.

3 Revisão de Literatura

O processo de reconhecimento do *splice site* é equivalente a um problemas de classificação binária, isto é, determinação do *Exon-Intron* e determinação do *Intron-Exon*, em que o processo envolve a transformação dos dados da sequência em vetores de características numéricas, sobre os quais um classificador subjacente opera.

Para melhor acurácia e redução da complexidade do tempo das abordagens de detecção de *splice site*, Zhang et al.(2006) utilizou máquinas de suporte vetorial (SVM) com *Bayes kernel* (B-SVM).

Huang et al. (2006) propôs um método de codificação de DNA eficiente, considerando nucleotídeos pareados (PN) e calculando suas diferenças de frequência entre *true* e *false site* verdadeiro e falso splice site (FDTF). Portanto, o classificador SVM passou a ser aplicado na predição de *splice sites* (PN FDTF-SVM). Em Meher, Sahu e Rao, (2016), o método de codificação proposta Huang et al. (2006) foi combinado com Random Forest (Floreta Aleatoria)(RF), mostrando uma melhora na predição de *exon-intron splice sites*. Zhang et al., (2010) propôs um modelo de Markov com comprimento variável, no qual o modelo de Markov de segunda ordem (MM2) é utilizado para extração de atributos e algum *threshold* (limiar) para seleção e classificação, o método produz boa acurácia, encontrar o valor ótimo para os *thresholds* (limiares) é uma tarefa difícil.

Três abordagens de codificação de DNA, foram propostas em Huang et al.(2006) codificações ortogonais, codificação *codon* e informação sequencial, um SVM linear foi empregado como o classificador (ECS-LSVM) obtendo bons resultados. Em Wei et al. (2013), a distribuição de tri-nucleotídeos foi combinada com o método de codificação, modelo de Markov de primeiro ordem MM1, utilizando *true* e *false sites*(verdadeiro e falso splice site) (DM). Logo, o método *F-score* foi utilizado para destacar a seleção de atributos, e o SVM para classificação dos *splite sites* foi usado (DM-SVM).

Li et al. (2012) propuseram um método de alta taxa de transferência SVM, sendo um dos principais métodos de detecção de *splice site* em humanos, considerando sua alta acurácia. O procedimento serve como um híbrido de atributos de posição (Pos), atributos de relação de posição adjacente (APR), e atributos de sequência de componentes multiescalares (MSC) como entrada (*input*) para o SVM. No entanto, o método requer alto custo computacional devido à muitos atributos que são produzidos pela codificação MSC.

Kamath et al. (2012) propôs um algoritmo evolucionário, utilizado para gerar atributos complexos (GF-EA) (local e não-local), a partir de sequências de DNA para prever *splice sites* usando o classificador SVM. Em Maji e Garg (2013), um algoritmo de três estágios foi proposto, sendo uma combinação do método de codificação MM2, método de análise de seleção de atributos e classificador SVM para resolver o problema de *splice site* (MM2F-SVM). De maneira interessante, não foi mencionado claramente como o modelo foi construído utilizando ambos os *sites*.

Loi e Rajapakse (2002) propuseram o MCM, sendo a versão híbrida do MM1 e do MM2, como um método de codificação eficiente, alimentando seus atributos produzidos para a rede neural. Posteriormente, Goel, Singh e Aseri (2015) melhorou a performance dos trabalhos anteriores, empregando o classificador SVM (MCM-SVM).

Pashaei e Aydin (2018) propuseram o modelo de Markov de ordem tres (MM3), como um método de codificação, o classificador SVM passou a ser aplicado na predição de *splice sites*, melhorou a performance dos trabalhos anteriores com método de codificação (SVM-MM1)Goel, Singh e Aseri (2015), (SVM-MM2)(PASHAEI; OZEN; AYDIN, 2017).

A segmentação da sequência de DNA é um passo fundamental no desempenho do método de codificação MCM, constituindo numa tarefa árdua que necessita de maiores estudos (PASHAEI; OZEN; AYDIN, 2017). Inspirados pela abordagem publicada em Wei et al. (2013), desenvolvemos os métodos DMM2-SVM (MM2-SVM Duplo), que considera um novo modelo híbrido a partir do modelo MM2, utilizando o MCC e sequências falsas de DNA (PASHAEI; OZEN; AYDIN, 2016).

3.1 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial (IA) que tem como objetivo desenvolver modelos que possam aprender através de experiência. Um sistema de AM é um programa de computador que toma decisões baseado em experiências acumuladas através da solução de problemas anteriores. Os diversos sistemas de AM possuem características particulares e comuns que possibilitam sua classificação quanto a forma de aprendizado utilizado (RUSSELL; NORVIG, 2016).

Não existe um único algoritmo no Aprendizado de Máquina que apresente um bom desempenho para todos os problemas (DIETTERICH, 1998). A seleção do melhor classificador depende do tipo de problema e a disponibilidade de dados a tratar. Diversos métodos de AM são utilizados com grande efetividade em diversas aplicações.

Nos métodos de AM existem alguns paradigmas de aprendizado. Os dois mais comuns são, o aprendizado supervisionado, onde é fornecido ao sistema de aprendizado um

conjunto de exemplos com a saída conhecida, ou seja, cada exemplo observado é descrito por um conjunto de atributos e pelo valor da classe à qual o exemplo pertence (RUSSELL; NORVIG, 2016). O algoritmo é treinado sobre um conjunto de dados pré-definidos, aprende o que precisa e toma decisões quando recebe novos dados para sua classificação. E o aprendizado não supervisionado, onde os algoritmos assumem que não se conhece a classe à qual pertencem e procuram encontrar nos valores de atributos similaridades ou diferenças que possam, respectivamente, agrupar à mesma classe (RUSSELL; NORVIG, 2016). Neste trabalho, será aplicado aprendizado supervisionado.

3.2 Naive Bayes

O classificador Naive Bayes é um dos algoritmos mais utilizado em Aprendizado de Máquina (JOHN; LANGLEY, 1995). É definido como ingênuo (naive), pois assume a hipótese de que todos os atributos são condicionalmente independentes, ou seja, a informação de um atributo não exerce influência sobre os demais. Ele é baseado na teoria de Bayes (JOHN; LANGLEY, 1995). Na Figura 5, está ilustrada a hipótese em que se baseia, indicando que cada atributo A_i, $i = 1, 2, ..., m$, influencia a classe C, mas nenhum exerce influência sobre o outro.

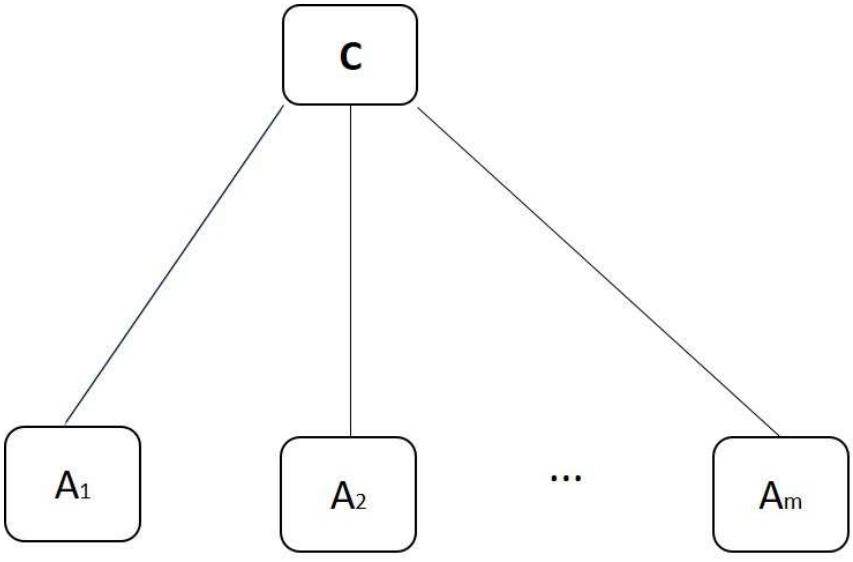

Figura 5 – Classificador bayesiano ingênuo

Naive Bayes calcula as probabilidades de uma classe C, dados os atributos A_1, A_2, $A_3, ...A_m$ conforme a equação abaixo:

$$P(C \mid A_1, A_2, A_3, ..., A_m) \tag{3.1}$$

Naive Bayes realiza a classicação de um objeto A_i, identificando a classe que possuir a maior probabilidade de conter o objeto informado. Partindo do pressuposto que um conjunto de dados possui n classes $C = (C_1, C_2, C_3, ..., C_n)$, e um determinado objeto $A = (A_1, A_2, A_3, ..., A_m)$, com classe desconhecida, o teorema de Bayes é formulado para essa hipótese da seguinte forma:

$$P(C_i \mid A_1, A_2, A_3, ..., A_m) = P(C_i)P(A_1, A_2, A_3, ..., A_n \mid C_i)P(A_1, A_2, A_3, ..., A_m) \quad (3.2)$$

O classicador Naive Bayes assume a hipótese de que todos os atributos são condicionalmente independentes, ou seja, a informação de um atributo não exerce influência sobre nenhum outro atributo, podendo essa hipótese ser representada pela equação:

$$P(X \mid C = c_i) = \prod_{i=1}^{m} P(A_i \mid C = c_i) \quad (3.3)$$

A classificação é então feita aplicando o teorema de Bayes para calcular a probabilidade de C_i dado uma particular instância de $A_1, A_2, A_3, ..., A_m$, predizendo a classe com a maior probabilidade a *posteriori*.

3.3 Máquinas de suporte vetorial

A teoria das máquinas de suporte vetorial (SVMs) foi desenvolvida inicialmente por Cortes e Vapnik (1995) no início dos anos 80, e se concentra no que é conhecido como Teoria da Aprendizagem Estatística. As máquinas de suporte vetorial (SVM), ou máquinas de vetores de suporte, configuram um conjunto de algoritmos de aprendizado supervisionados para a classificação. Eles são baseados na representação das amostras em um espaço multidimensional, com a busca pelo o hiperplano que melhor os separam. Este hiperplano tenta maximizar a distância entre as amostras da fronteira de cada classe (Figura 6). As amostras na fronteira são chamadas de vetores de suporte.

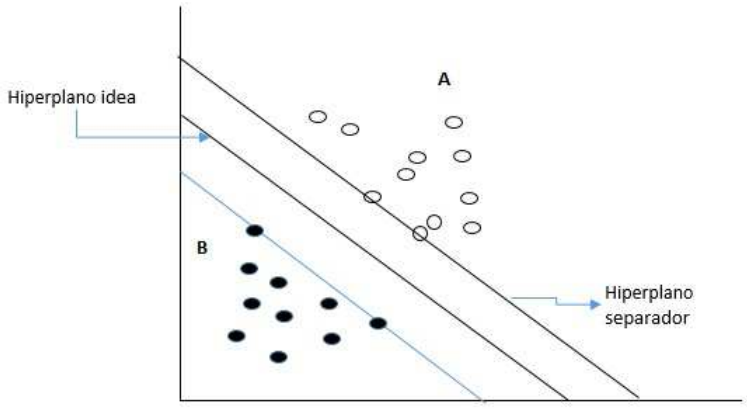

Figura 6 – Hiperplanos paralelos e vetores de suporte em R^2

3.3.1 Modelos Lineares de Suporte a Vetores

Sejam dois possíveis rótulos dados por $Y = \{-1, 1\}$, um conjunto de vetores $\{(x_1, y_1), ..., (x_n, y_n)\}$, $x_i \in R^d$ e $y_i \in \{-1, 1\}$, para $i = 1, ..., n$. Diz-se ter um problema separável se houver um hiperplano em R^d que separa os vetores com rótulo 1 daqueles que possuem rótulo -1.

Dado um conjunto separável existe (pelo menos) um hiperplano

$$\pi : \omega x + b = 0 \tag{3.4}$$

que separa os vetores $x_i, i = 1, .., n$. A busca de SVMs entre todos os hiperplanos separadores que maximizam a distância de separação entre os conjuntos $\{x_i, 1\}$ é $\{x_i, -1\}$ (Figura 7).

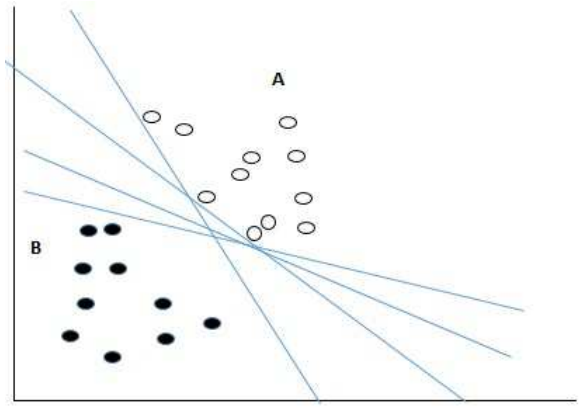

Figura 7 – Conjunto de hiperplano separador em R^2

É possível verificar o problema de otimização correspondente: ao fixar um separador de hiperplano é possível redimensionar o parâmetros ω e b de tal forma que:

$$x_i\,\omega + b \geq 1 \quad para \quad y_i = 1 \quad (região \quad A), \tag{3.5}$$

$$x_i\,\omega + b \leq -1 \quad para \quad y_i = -1 \quad (região \quad B). \tag{3.6}$$

Assim, a separação mínima entre os vetores é o hiperplano separador, e as duas desigualdades podem ser expressas em uma única forma:

$$y_i(x_i\,\omega + b) - 1 \geq 0, i = 1, ..., n. \tag{3.7}$$

Os vetores de rótulo 1, para os quais a igualdade na equação acima é atendida, pertencem a hiperplano $\pi_1 : x_i\omega + b = 1$, com vetor normal ω e distância perpendicular à origem igual a $|1 - b|/\|\omega\|$, onde $\|\omega\|$ é a norma euclidiana de ω. Da mesma forma, os vetores de rótulo -1 pertencem ao hiperplano $\pi_2 : x_i\omega + b = -1$, com vetor normal ω e distância perpendicular à origem igual a $|-1 - b|/\|\omega\|$. Portanto, temos os hiperplanos π_1 e π_2 que são paralelos. A separação entre eles é $2/\|\omega\|$ e nenhum vetor do conjunto de treinamento está entre eles.

Em relação as escolhas possíveis dos hiperplanos π_1 e π_2, parece natural escolher aquele que proporciona a maior separação entre eles, pois isso tornaria possível distinguir mais claramente as regiões onde ficam os pontos com rótulos diferentes.

$$\begin{aligned} &\min_{\omega \in R^d} \ \tfrac{1}{2}\|\omega\|^2, \\ &y_i(x_i\,\omega + b) - 1 \geq 0, \quad i = 1, .., n. \end{aligned} \tag{3.8}$$

Uma característica importante do SVM é que, se qualquer número de vetores que atenda a desigualdade estrita de (3.7) for adicionado ou eliminado, a solução do problema de otimização não será afetada. No entanto, basta adicionar um vetor que esteja entre os dois hiperplanos, para que a solução mude completamente. Para resolver o problema de otimização com restrições (3.8) são utilizados multiplicadores de Lagrange. Então a função objetivo é:

$$L_P(\omega, b, \alpha_i) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(x_i\,\omega + b) - 1). \tag{3.9}$$

A situação permanece como um problema de programação quadrática, em que a função objetivo é convexa, e vetores que satisfazem as restrições formam um conjunto convexo. Isso significa que as duas situações associadas ao problema primário podem ser resolvidas: maximizando a função $L_P(\omega, b, \alpha_i)$ em relação às variáveis duplas α_i sujeito às restrições impostas para que os gradientes de L_P em relação a ω e b sejam nulos, e sujeitos também ao conjunto de restrições

$$C_2 = \{\alpha_i \geq 0, i = 1, .., n\}.$$

A solução para este problema é expressa na seguinte forma:

$$\omega = \sum_{i=1}^{n} \alpha_i y_i x_i, \sum_{i=1}^{n} \alpha_i y_i = 0,$$

e a função de objetivo duplo:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j. \tag{3.10}$$

Os vetores do conjunto de treinamento que fornecem um multiplicador $\alpha_i > 0$ são chamados vetores de suporte. Esses vetores estão em um dos hiperplanos π_1 ou π_2. Para este tipo de modelo de aprendizagem, os vetores de suporte são os elementos críticos, pois fornecem a aproximação do problema. Uma vez que, se todos os outros elementos do conjunto de treinamento são eliminados (ou alterados por outros que não estão entre os dois hiperplanos), e o problema de otimização é repetido, os mesmos hiperplanos do separador são encontrados.

As circunstâncias do problema de otimização levam à condição a ser cumprida:

$$\alpha_i(y_i(x_i\omega + b) - 1) = 0,$$

chamada de condição complementar por Karush-Kuhn-Tucker (KKT) (QI; JIANG, 1997). Essas restrições indicam que o produto das limitações do problema primário ($y_i(x_i\omega + b) - 1 \geq 0$), e as restrições do problema duplo ($\alpha_i \geq 0$), são canceladas em todos os vetores de treinamento. Deste modo, segue-se que as condições KKT, para o problema primário definido a partir da função objetiva, são as seguintes:

$$\omega_j - \sum_{i=1}^{n} \alpha_i y_i x_{ij} = 0 \quad j = 1, ..., d,$$
$$\frac{\partial}{\partial b} L_P = -\sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$
\begin{aligned}
y_i(x_i\omega + b) - 1) = 0 \quad \forall i = 1, ..., n\,, \\
\alpha_i \geq 0 \quad \forall i = 1, ..., n\,, \\
\alpha_i(y_i(x_i\omega + b) - 1) = 0 \quad \forall i = 1, ..., n.
\end{aligned}
$$

A partir dos desenvolvimentos iniciais, não é seguida uma maneira explícita de determinar o valor b, no entanto, a condição complementar de KKT nos permite determiná-lo. Para isso, basta escolher um $\alpha_i > 0$ e achar o valor de b, obtendo $b = y_i - x_i\omega$. Embora b tenha sido determinado, é adequado realizar os cálculos com todos os $\alpha_i > 0$ e escolher como valor b um valor médio dos resultados obtidos, arrendondando os erros intrínsecos associados a qualquer método de cálculo numérico:

$$b = \frac{1}{\{\alpha_i > 0\}} \sum_{\alpha_i > 0} (y_i - x_i\omega).$$

Uma vez obtido o vetor ω e a constante b, a solução para o problema de otimização é interpretada a partir de uma função Θ da seguinte maneira:

$$\Theta(x) = \begin{cases} 1, & se \quad \pi(x) > 0; \\ -1, & se \quad \pi(x) < 0. \end{cases}$$

3.4 Redes Neurais

Redes neurais artificiais são modelos matemáticos inspirados no sistema nervoso dos seres vivos, propostas inicialmente por McCulloch e Pitts em 1943 (HAYKIN, 1994). Possuem a capacidade de adquirir e manter conhecimento (baseado em informações), podendo ser definidas como um conjunto de unidades de processamento, representadas por neurônios artificiais. Esses neurônios possuem interconexões (sinapses artificiais), implementadas por vetores e matrizes de pesos sinápticos (SILVA et al., 2017).

De acordo com Haykin (1994), uma rede neural artificial é um processador maciçamente paralelo distribuído, constituído de unidades de processamento simples, que possui propensão natural para armazenar conhecimentos experimentais e torná-los disponíveis para o uso, assemelhamdo-se ao cérebro em dois aspectos:

- O conhecimento é adquirido pela rede por meio de dados do ambiente, num processo de aprendizagem. O procedimento de treinamento ocorre por meio de algun Algoritmo de Aprendizagem, com a finalidade de ajustar os pesos sinápticos da rede
- As conexões entre os neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Uma rede neural é caracterizada por seu padrão de conexões entre os neurônios (chamado de arquitetura de rede) e seu método de determinar os pesos nas conexões (chamado de treinamento), em que os pesos são ajustados com base nos dados. Em outras palavras, as redes neurais aprendem com exemplos e exibem capacidade de generalização. Esse recurso faz com que esses modelos computacionais sejam muito atraentes em domínios de aplicação que possuem pouca ou incompleta compreensão do problema a ser resolvido, porém com os dados para o treinamento prontamente disponíveis. Redes neurais normalmente têm grande potencial para paralelismo, uma vez que os cálculos dos componentes são amplamente independentes (WU; MCLARTY, 2012).

3.4.1 O Neurônio Artificial

O neurônio é uma unidade de processamento de informação fundamental para a operação de uma rede neural. É possível identificar três elementos básicos da rede neural (HAYKIN, 1994):

- Um conjunto de sinapses ou conexões, sendo cada uma delas associada a um peso sináptico.
- Um somador para os sinais de entrada, ponderados pelas respectivas sinapses do neurônio; as operações descritas constituem um combinador linear.
- Uma função de ativação para limitar a amplitude da saída do neurônio.

O Modelo neuronal inclui também um bias (b_k), aplicado externamente, que possui o efeito de aumentar ou diminuir a entrada da função de ativação, dependendo se ele é positivo ou negativo. Em termos matemáticos podemos descrever um neurônio k, escrevendo as seguintes equações;

$$u_k = \sum_{j=1}^{n} w_{kj} x_j \tag{3.11}$$

$$y_k = \phi(u_k - \theta_k) \tag{3.12}$$

onde $x_1, x_2, x_3, ..., x_n$ são os sinais de entrada; $w_{k_k}, w_{k_2}, w_{k_3}, ..., w_k$ são os pesos sinápticos do neurônio k; u_k é a saída do combinador linear; b_k é o bias; $\phi(\cdot)$ a função de ativação; y_k é a saída do neurônio,como mostrado na Figura 8.

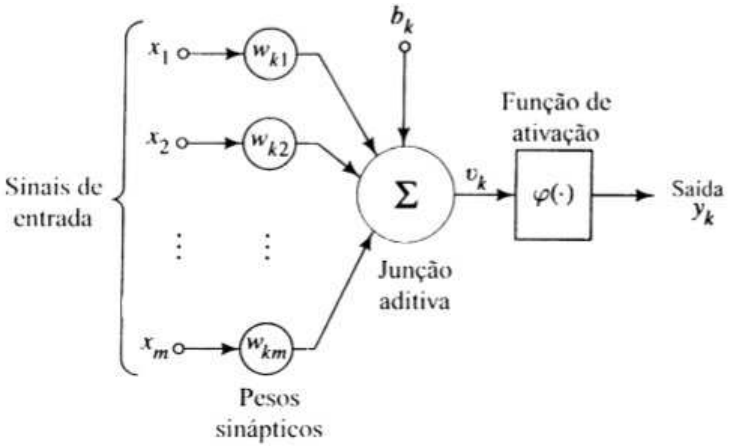

Figura 8 – Modelo de um neurônio artificial. (HAYKIN, 1994)

3.4.2 Estruturas de RNAs

A maneira pela qual os neurônios de uma rede neural são estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede (HAYKIN, 1994). Esses algoritmos de aprendizado (regras) dependem muito da estrutura da rede neural. O desempenho da rede neural não depende apenas da função de ativação, sendo também relacionado à conformação de sua estrutura. Haykin (1994), Silva et al. (2017) dividem a arquitetura de um RNA do tipo *Multileyer Perceptron* em três tipos: redes *feedforward*, de camada simples, redes *feedforward*, de camada múltiplas e redes recorrentes.

3.4.2.1 Redes Alimentadas Adiante com Camada Única

Este tipo de estrutura é composta por apenas uma camada de entrada e uma única camada de neurônios, sendo esta última a mesma camada de saída (HAYKIN, 1994). A informação sempre flui em uma única direção (portanto, unidirecional), que é da camada de entrada para a camada de saída (Figura 9). É possível observar que nas redes pertencentes à essa arquitetura, o número de saídas de rede sempre coincidirá com a quantidade de neurônios. Essas redes são geralmente empregadas na classificação de padrões e problemas de filtragem linear. Entre os principais tipos de rede pertencentes a arquitetura *feedforward* estão o Perceptron e o Adaline (SILVA et al., 2017).

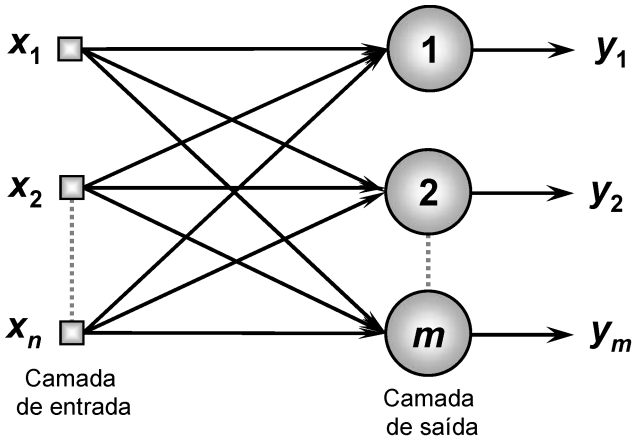

Figura 9 – Rede Alimentadas Adiante com Camada Única.

(SILVA et al.,2017)

3.4.2.2 Redes Alimentadas Diretamente com Múltiplas Camadas

Esta rede neural se distingue pela presença de uma ou mais camadas ocultas. A função dos neurônios escondidos é intervir entre a entrada externa e a saída da rede (HAYKIN, 1994). São empregados na solução de diversos problemas, como aqueles relacionados à aproximação de funções, classificação de padrões, identificação de sistemas, controle de processos, otimização, robótica, entre outros (SILVA et al., 2017).

A Figura 10 exibe uma rede *feedforward* com múltiplas camadas, composta de uma camada de entrada com n sinais de amostra, duas camadas ocultas de neurônios n_1 e n_2 respectivamente e, finalmente, uma camada de saída composta de m neurônios representando os respectivos valores de saída do problema analisado.

A quantidade de neurônios que compõem a primeira camada oculta é geralmente diferente do número de sinais que compõem a camada de entrada da rede. De fato, o número de camadas ocultas e sua respectiva quantidade de neurônios dependem da natureza e complexidade do problema que está sendo mapeado pela rede, bem como da quantidade e qualidade dos dados disponíveis sobre o problema (HAYKIN, 1994).

Entre as principais redes que utilizam arquiteturas *feedforward* de múltiplas camadas estão o *Multilayer Perceptron* (MLP) e o *Radial Basis Function* (RBF), cujos algoritmos de aprendizado utilizados em seus processos de treinamento são baseados respectivamente na regra delta generalizada e na regra delta competitiva.

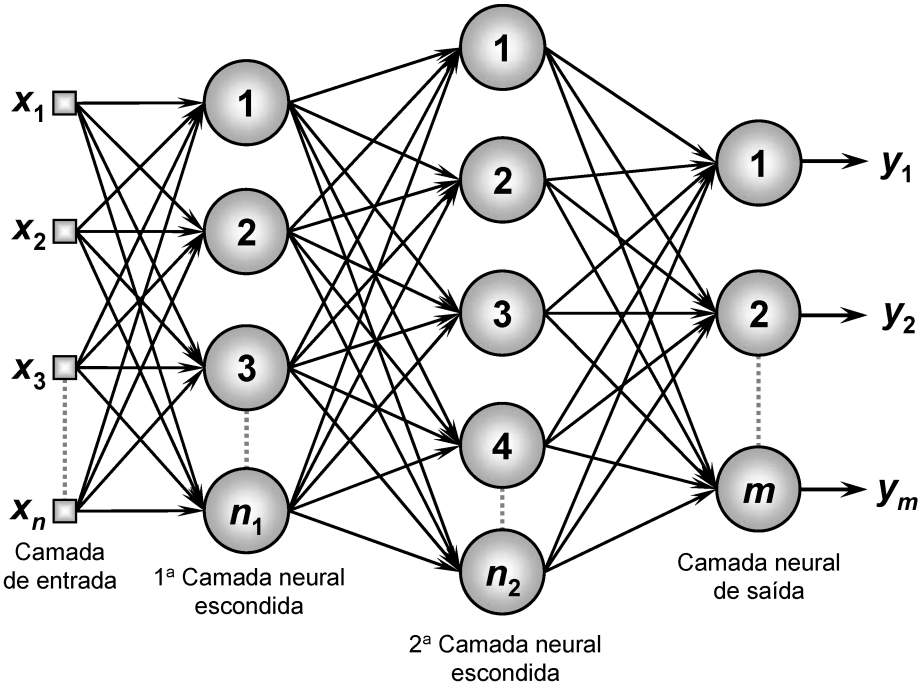

Figura 10 – Redes Alimentadas Diretamente com Múltiplas Camadas (SILVA et al., 2017).

3.5 Random Forest (Floresta Aleatória)

O algoritmo de floresta aleatória (RF), proposto por Breiman (2001), tem sido bem sucedido como um método de classificação e regressão de propósito geral (BIAU; SCORNET, 2016). Uma floresta aleatória é um classificador que consiste em uma coleção de classificadores estruturados em árvore (*ensemble methods*) $h(x, \theta_k), k = 1, ...$ sendo θ_k vetores aleatórios identicamente distribuídos, de forma que cada árvore lança um voto unitário para a classe mais popular na entrada (BREIMAN, 2001).

A construção de uma RF envolve o uso da técnica de *bootstrap* para criar subconjuntos de dados utilizados no crescimento das árvores do modelo. Cada árvore é baseada em um subconjunto aleatório diferente dos dados originais. Geralmente, os subconjuntos incluem cerca de 2/3 da amostra completa. Após a criação dos conjuntos de árvores é possível efetuar a classificação na qual possui melhor ganho de conhecimento para a solução de determinado problema. Para isto, é necessário escolher um subconjunto de árvores de decisão que possui melhor lógica e vantagens para a tomada de decisão. Para cada subconjunto é dado um voto sobre qual classe o atributo chave deve pertencer.

As RFs possuem a característica de dividir e conquistar, e isto possibilita algumas propriedades que se destacam em relação as outras técnicas, nas quais algumas delas são:

- Possui boa taxa de acerto quando testado em diferentes conjuntos de dados.
- Técnica exata.
- Evitam super ajustamento (overfitting).
- Menos sensíveis a ruídos.
- Classificação aleatória das árvores sem intervenção humana.

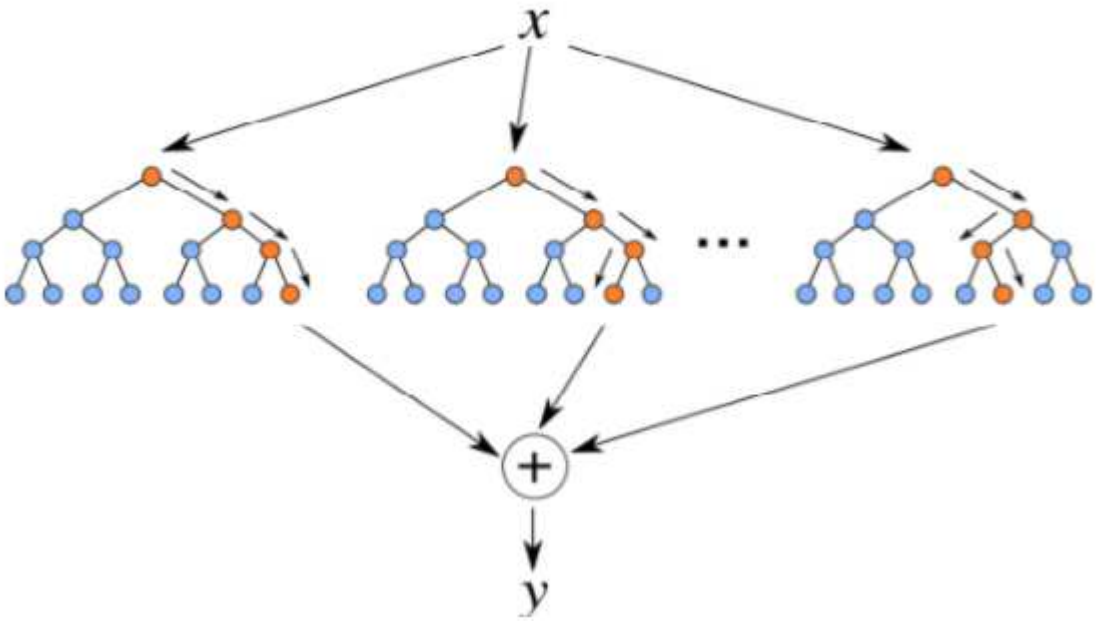

Figura 11 – Exemplo da predição de uma instância X aplicada ã cada árvore da RF (GOLDSTEIN; POLLEY; BRIGGS, 2011).

Como representado na Figura 11, não é possível verificar que partindo de uma base de dados, geram-se várias RFs. Neste ponto, cada uma produz diversas regras e nelas existe a possibilidade de descoberta de novos padrões, que poderão ser determinante na tomada de decisão. Com a escolha feita, as mesmas são aplicadas numa base de dados chegando a um resultado Y.

4 Materiais e métodos

Foi proposta uma metodologia para a classificação de *exon-intron* / *intron-exon* usando técnicas (AM). Essa abordagem pode ser dividida nas quatro etapas:

1. Sequência DNA,
2. Extrator de características
3. Aprendizado de máquina
4. Geração de sequências de probabilidades

todo o processo é ilustrado na Figura 12.

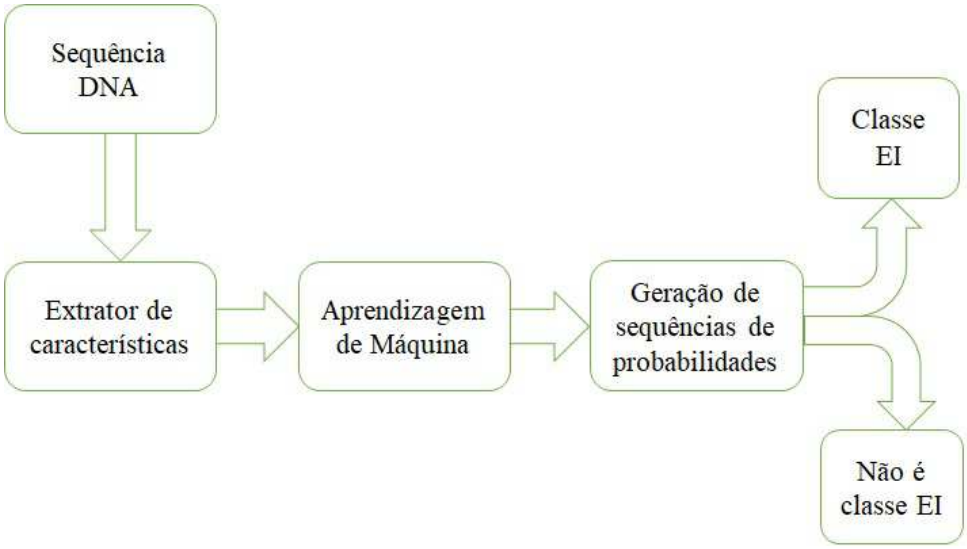

Figura 12 – Arquitetura da metodologia proposta.

4.1 Método de codificação

Os métodos baseados em aprendizado de máquina são usados para resolver o problema de classificação de *splice site* . A entrada de um classificador de aprendizado de máquina geralmente é um vetor de caractere numérico. Para converter sequências de DNA em vetores característicos, diversos métodos de codificação de DNA são empregados. Os

métodos de codificação de DNA tentam extrair tanta informação quanto as sequências de DNA, aumentando a precisão dos classificadores. Neste trabalho foi utilizada a codificação binária descrita em Damaševičius (2008).

Codificação ortogonal: Os nucleotídeos em uma sequência de DNA são representados por vetores binários ortogonais de 4 dimensões:

$A- > (0001), C- > (0010), G- > (0100), T- > (1000)$.

Exemplo: Suponha que uma sequência de DNA, **AATCGTC**, com codificação ortogonal da sequência, seja representada como:

0001000110000010010010000010

Codificação Binária (CB): Existem métodos distintos para representar os nucleotídeos de DNA usando um código binário de 2 bits. De acordo com Damaševičius 2008 (2008), existem $4! = 24$ regras; no entanto, apenas 3 regras são essencialmente diferentes, enquanto as regras restantes podem ser obtidas por inversão:

- Binária 1 (CB1): $A- > (0,0), C- > (0,1), G- > (1,0), T- > (1,1)$.

  Exemplo: AATCGT $- >$ 00001101101101

- Binária 2 (CB2): $A- > (0,0), c- > (0,1), G- > (1,1), T- > (1,0)$.

  Exemplo: AATCGTC $- >$ 00001001111001.

- Binária 3 (CB3): $A- > (0,0), C- > (1,1), G- > (0,1), T- > (1,0)$

  Exemplo: AATCGTC $- >$ 00001011011011.

Codificação de letras simples: São regras de representação de um nucleotídeo em 1, e restantes em 0. Na prática, há quatro dessas regras : regra A, regra C, regra G e regra T. Estas regras refletem a distribuição de um tipo particular de nucleotideo ao longo da sequência de DNA (DAMAŠEVIČIUS, 2008):

- regra A (RA): $(A- > 1, B- > 0, B = C, G, T)$

  Exemplo: AATCGTC $- >$ 1100000

- regra C (RC): $(C- > 1, D- > 0, D = A, G, T)$.

  Exemplo: AATCGTC $- >$ 0001001

- regra G (RG): $(G- > 1, H- > 0, H = A, C, T)$.

  Exemplo: AATCGTC $- >$ 0000100

- regra T (RT): $(T- > 1, V- > 0, V = A, C, G)$.

  Exemplo: AATCGTC $- >$ 0010010

Regra de agrupamento: São baseadas no agrupamento de 4 bases nitrogenadas do DNA em dois subconjuntos de dois nucleotídeos cada um. Existem três partições diferentes, portanto existem três regras diferentes. Cada regra representa um aspecto distinto da estrutura das moléculas de DNA (DAMAŠEVIČIUS, 2008).

A regra SW ($AT \quad versus \quad CG$) representa a diferença no número de ligações de hidrogênio na molecula de DNA. Cada nucleotídeo forte (S)(possuindo C ou G) possui 3 ligações de hidrogênio, e cada nucleotídeo fraco (W) (com a presença de A ou T) possui apenas 2 ligações de hidrogênio.

- Regra SW: $(S- > 1, W- > 0), S = A, T, W = C, G$

  Exemplo: AATCGTC$- >$ 1110010

A Regra RY ($AG \quad versus \quad TC$) descreve como as purinas (R) e pirimidinas (Y) são distribuídas ao longo da sequência de DNA.

- Regra RY: $((R- > 1, Y- > 0), R = A, G, Y = C, T)$

  Exemplo: AATCGTC $- >$ 1100100

A Regra KM ($(AC \quad versus \quad TG)$) caracteriza como as aminas (M) e as cetonas (K) estão distribuídas ao longo da sequência de DNA.

- Regra KM: $((K- > 1, M- > 0), K = A, C, M = G, T)$

  Exemplo: AATCGTC $- >$ 1101001

4.2 Dados utilizados

É feita uma distinção entre *acceptor* e *donor splices sites*, portanto os conjuntos de dados do *splice site* são divididos em conjunto *donor* e *acceptor*, e o desempenho de classificação é comparado separadamente em cada subconjunto (PASHAEI; AYDIN, 2018).

Dois conjuntos de dados, de *splice junctions* foram considerados neste trabalho. O primeiro foi obtido do *UCI Machine Learning Repository* Newman, (1998) , sendo este um banco de dados de sequências de primatas. O banco de dados contém 3.190 instâncias no total. Posteriormente, as sequências foram aleatoriamente amostradas para obter um banco de dados com o número de instâncias por classe em uma proporção específica: 25% para *donors*, 25% para *acceptors* 50% para os *false splice sites*.

A segunda base de dados utilizada foi retirada do *Homo Sapiens Splice Sites*, pelo conjunto de dados (HS3D) (POLLASTRO; RAMPONE, 2003). Esta base de dados contém originalmente 5.947 sequências de DNA humano com sítios de *splice* conhecidos (tendo presentes *donors* e *acceptors*), e 635.666 sequências com *false splice*. Para este conjunto de dados, 11.184 sequências foram escolhidas aleatoriamente, distribuídas na mesma proporção de antes: 25% para os *donors*, 25% para os *acceptors* e 50% para os *false splice sites*. Na Tabela 1 esta o resumo dos conjuntos de dados usados neste trabalho, mostrando o número total de instâncias e a distribuição de classes. As denominações EI, IE e N significam junções *Exon-Intron*, junções *Intron-Exon* e *false splice site*, respectivamente.

Para verificar a eficácia do nosso método, foi retirada do *Homo sapiens chromosome 21 segment HS21C103*, pelo *https://www.ncbi.nlm.nih.gov*. O banco de dados contem 4.860 sequências de DNA humano. Para este conjunto de dados, 2.536 sequências foram escolhidas aleatoriamente, distribuídas na mesma proporção de antes: 25% para os *donors*, 25% para os *acceptors* e 50% para os *false splice sites*

Os parâmetros dos algoritmos para as bases de dados utilizadas neste trabalho, eles foram obtidos de estudos anteriores. Para o SVM (utilizando *kernel Radial*), C (150) e sigma ([0,0001]) (ZHANG et al., 2006). Para o Random Forest, testes foram realizados para os seguintes valores do parâmetro mtry: 4 o número de atributos preditores, *ntree* = 500 número de árvores (MEHER; SAHU; RAO, 2016).

Tabela 1 – Resumo do conjunto de dados usado no experimento

Dados	Instâncias	Classe(%)		
		EI (25%)	IE(25%)	N(50%)
UCI	3190	767	768	1655
HS3D	11184	2796	2796	5592

4.3 Métrica para a avaliação do classificador

Os modelos de classificação são desenvolvidos para serem aplicados a dados diferentes daqueles que foram usados para os construir, tornando-se essencial avaliá-los para medir até que ponto eles são capazes de realizar as classificações corretamente. Essa avaliação é tipicamente feita dividindo o conjunto de dados disponíveis em 2 partes: o conjunto de treino e o conjunto de teste. O modelo é construído com base no conjunto de treino. Posteriormente, é aplicado aos dados de testes, comparando-se o valor da classe de cada exemplo neste conjunto com o que se obtém na previsão. Esta técnica serve para avaliar o modelo e estimar a incerteza das suas previsões. Existem variantes desse método, como por exemplo, a validação cruzada (MINING, 2006).

O desempenho de um algoritmo de classificação é geralmente descrito por medidas estatísticas como *Accuracy* (Acurácia), *Presicion* (Precisão), *Specificity* (Especificidade), *Sensitivity* (Sensibilidade) e área sob a curva (ROC); calculado sobre a matriz confusão (ou matriz de contingência), obtida pelo classificador durante o teste.

Nesta seção, serão apresentadas as principais métricas que poderão ser utilizadas para se avaliar um classificador binário. Para isto, as seguintes siglas serão consideradas:

- Sequências de DNA que contêm *exon/intron* (EI);
- Sequências de DNA que contêm *intron/exon* (IE);
- Sequências de DNA que não contêm EI nem IE (N).

4.4 Matriz de confusão

A Matriz de Confusão é uma tabela para visualização dos resultados tipicamente utilizada em problemas de classificação. Cada linha da matriz representa as instâncias previstas de uma classe, enquanto cada coluna da matriz representa as instâncias reais de uma classe (VISA et al., 2011).

Uma matriz de confusão, de tamanho $2x2$, é utilizada para classificação binária. Desta forma, podemos considerar que a matriz de confusão é uma tabela, que registra neste estudo o número de *true positive* (ou verdadeiro positivo, que ocorre quando um EI é corretamente classificado como EI (TP); *false positive* (ou falso positivo), que ocorre quando um N é incorretamente classificado como EI (FP); *false negative* (ou falso negativo), que ocorre quando um EI é incorretamente classificado como N(FN); e *true negative* (ou verdadeiro negativo), que ocorre quando um N é corretamente classificado como N (TN). Na Tabela 2 representa a matriz de confusão para uma classificação de duas classes.

Tabela 2 – Matriz de confusão para classificação de duas classes

	(EI) Atual Positivo	(N) Atual Negativo
(EI) Previsto positivo	TP	FN
(N) Previsto negativo	FP	TN

- TP (*true positive* ou positivos verdadeiros): número de instâncias positivas que foram corretamente classificadas;
- FP (*false positive* ou falsos positivos): número de instâncias negativas classificadas como positivos;
- FN (*false negative* ou falsos negativos): número de instâncias positivas classificadas como negativas;
- TN (*true negative* ou negativos verdadeiros): número de instâncias negativas corretamente classificadas;

A partir de uma tabela de confusão, mais especificamente, quatro métricas de avaliação são observadas: Acurácia (*Accuracy*), Precisão (*Presicion*), Especificidade (*Specificity*) e Sensibilidade (*Sensitivity*).

Acurácia (*Accuracy*): porcentagem de instâncias positivas e negativas, classificadas corretamente sobre a soma de amostras positivas e negativas, ou seja, trata-se da proporção de classificações corretas e seu cálculo é descrito pela equação:

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precisão (*Precision*): proporção de instâncias positivas que foram classificadas corretamente. Em problemas de classificação binária, a precisão pode ser obtida pela seguinte equação:

$$Precisão = \frac{TP}{TP + FP} \tag{4.2}$$

Especificidade (*Recall ou Specificity*): descreve a proporção que foram classificadas corretamente como instâncias negativas. Esta medida pode ser estimada pela seguinte equação:

$$Especificidade = \frac{FP}{TN + FP} \tag{4.3}$$

Sensibilidade (*Sensitivity*): descreve a porção que foi classificada corretamente como exemplos positivos. Esta medida pode ser estimada pela seguinte equação:

$$Sensibilidade = \frac{TP}{TP + FN} \tag{4.4}$$

4.5 Validação cruzada

A validação cruzada é um método empregado para avaliação e posterior comparação entre modelos de classificação ou de regressão por meio da estimação da acurácia dos mesmos, a qual é definida por uma medida especificada a priori. Esse método divide o conjunto de instâncias ou observações com n elementos em k partes iguais quando o n é divisível por k, ou em k - *1* partes iguais e a k-ésima parte com o número de elementos igual ao resto da divisão entre n e k. Nessa abordagem, cada observação é usada um mesmo número de vezes para treinamento e exatamente uma única vez para teste (KOHAVI et al., 1995).

Na Figura 13 é exemplificado o uso de *k-fold* com $k = 5$. Primeiramente, o conjunto de dados inicial é dividido em 5 partes, onde a primeira parte em cinza é separada para teste e as outras 4 partes em branco são usadas para o treinamento do classificador. Esse procedimento é replicado k vezes e o erro de predição é calculado a partir dos k conjuntos de teste. Cada subconjunto é usado somente uma vez no teste e 4 vezes no treinamento, isto é, todas as observações são usadas ora para treinamento, ora para teste.

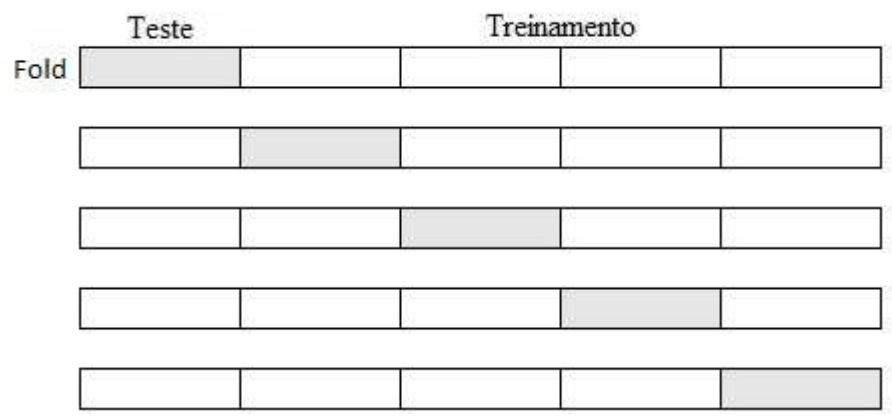

Figura 13 – Exemplo do 5-fold validação cruzada

4.6 Distância de Kolmogorov-Smirnov (KS)

O critério de divisão de *Kolmogorov-Smirnov* foi usado por Friedman, (1977) para uma partição binária em algoritmos de regras de decisão. A distância de Kolmogorov-Smirnov mede a separação de duas funções de distribuição. Naturalmente permite separar uma população em dois grupos homogêneos.

A distribuição de *Kolmogorov-Smirnov* foi originalmente concebida como um teste de hipótese para medir a aderência de uma distribuição aos dados. Em problemas de classificação binária, ela tem sido usada como medida de dissimilaridade para avaliar o poder discriminante do classificador medindo a distância que a sua pontuação produz entre as funções de distribuição acumulada (FDA) das duas classes de dados. A métrica comum para ambos os fins é a diferença vertical máxima entre as FDAs (Max(KS) (ADEODATO; MELO, 2016). Considere as funções de distribuição acumulada $F_A(x)$ e $F_B(x)$ que correspondem a $f_A(x)$ e $f_B(x)$. Na Figura 14, um ponto de corte ótimo α é aquele que maximiza $|F_{A(\alpha)} - F_{B(\alpha)}|$ e esse valor máximo é a distância do Kolmogorov-Smirnov (KS) (UTGO; CLOUSE, 1996).

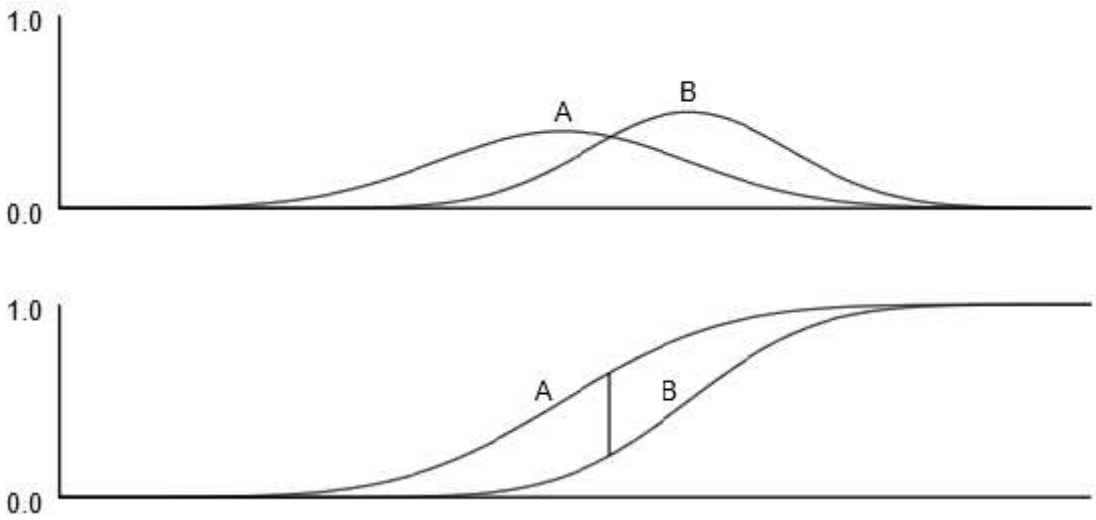

Figura 14 – Probabilidade condicional de classe e acumulada para duas classes

4.7 Índice de Youden

O índice de Youden γ é uma das medidas mais antigas proposta em 1950 por Youden como uma solução prática para relacionar a sensibilidade e a especificidade em testes diagnósticos, este índice tem como objetivo medir a performance geral de testes diagnósticos. Também configura boa opção utilizado para determinar ponto de corte

entre sensibilidade e especificidade (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ 2006, BEKKAR; DJEMAA; ALITOUCHE, 2013).

$\gamma = Sensibilidade(1 - Especificidade)$ O índice de Youden tem sido tradicionalmente usado para comparar habilidades diagnósticas de dois testes (BIGGERSTAFF, 2000).

O Índice de Youden possui valores entre -1 e 1, sendo que 1 representa o classificador perfeito e 0 representa um classificador que fornece a mesma proporção de resultados positivos para ambos os grupos (sabidamente positivos e sabidamente negativos), ou seja, não possui capacidade de discriminação, e -1, um classificador cuja predição é perfeitamente invertida.

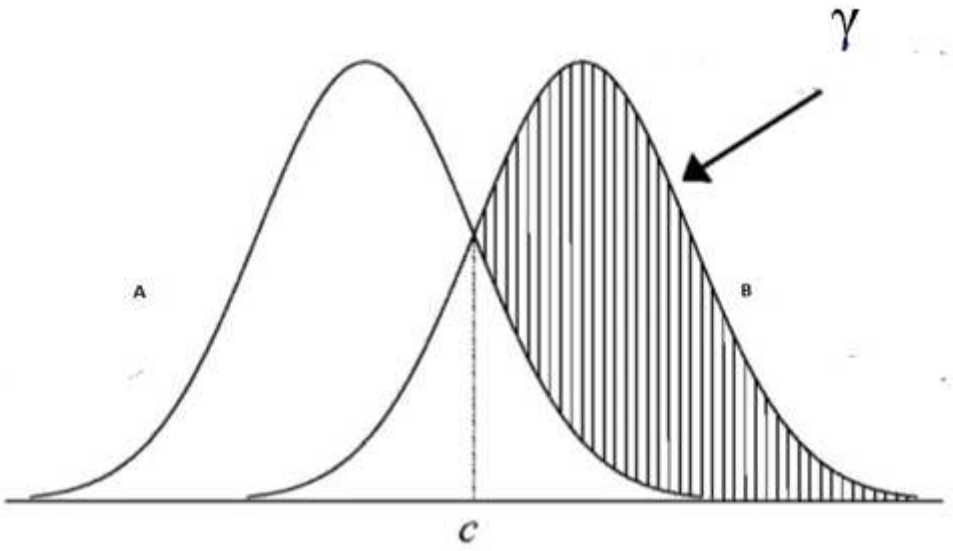

Figura 15 – Representação do índice Youden e o ponto de corte ótimo (c)

4.8 Razão verossimilhança

Razão de Verossimilhança (*Likelihood Ratio–*ρ): A ρ faz uso da sensibilidade e da especificidade para avaliar o desempenho do classificador (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006, BEKKAR; DJEMAA; ALITOUCHE, 2013). Existem dois índices:

- Razão de Verossimilhança Positiva(*Positive Likelihood Ratio* – ρ_+);
- Razão de Verossimilhança Negativa (*Positive Likelihood Negative* – ρ_-).

A razão de verossimilhança positiva(*Positive Likelihood Ratio* – ρ_+) pode ser entendida como uma medida da probabilidade de um EI ser classificado como EI dividida pela probabilidade de um N também ser classificado como EI. Pode ser obtida pela fórmula apresentada pela equação que segue:

$$\rho_+ = \frac{sensibilidade}{1 - especificidade} \tag{4.5}$$

A razão de verossimilhança negativa (*Negative Likelihood Ratio* – ρ_-) pode ser entendida como uma medida da probabilidade de um EI ser classificado como N dividida pela probabilidade de um N também ser classificado como N. Pode ser obtida pela equação que segue:

$$\rho_- = \frac{1 - sensibilidade}{especificidade} \tag{4.6}$$

Maior verossimilhança positiva e menor verossimilhança negativa significam melhor desempenho nas classes positivas e negativas, respectivamente.

5 Resultados e Discussão

A avaliação de desempenho do classificador foi realizada de acordo com as medidas de avaliação da matriz de confusão já apresentadas no Capítulo 4. Os resultados da classificação para diferentes dados, regras e algoritmos estão resumidos a seguir.

5.1 Conjuto de dados UCI (Newman 1998)

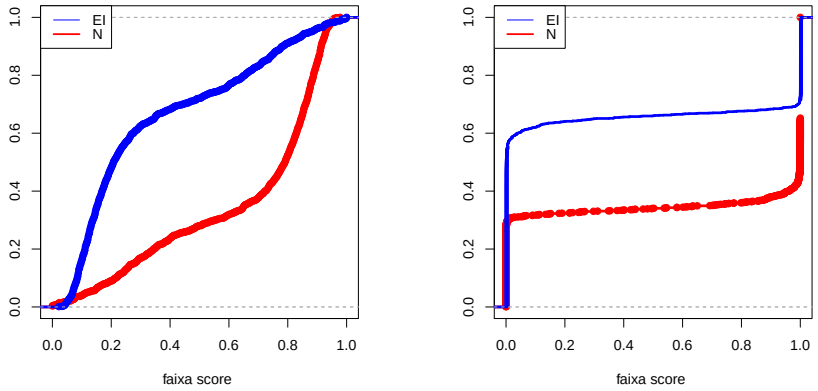

Figura 16 – KS obtido a partir do escores das distribuções EI, N, para os algoritmos RF e NB

As linhas azul e vermelha correspondem a funções de distribuição empírica de EI e N. KS é a diferença máxima entre duas funções de distribuição empírica, e uma medida de quão boa é a discriminação dos algoritmos, isto é, como os modelos conseguem separar as distribuições de EI e N. Pode-se observar, de acordo com o gráfico, que RF separa mais as classes do que NB.

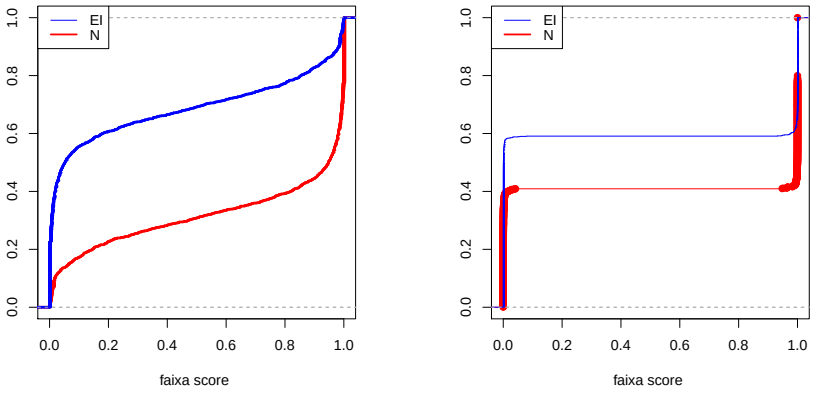

Figura 17 – KS obtido a partir das distribuções EI, N, para os algoritmos SVM e RNN

Na Figura 17 KS é a diferença máxima entre duas funções de distribuição empírica, as linhas azul e vermelha correspondem a funções de distribuição empírica de EI e N.

- Classificação (EI-N)

Os resultados experimentais com os dados UCI para a classificação EI-N, são mostrados nas Tabelas 3, 4, 5. A Tabela 3 mostra os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (ortogonal, CB1, CB2, CB3) como entrada de dados para o algoritmos, através de validação cruzada, os valores de desempenho para as diferentes métricas são obtidos. Pode-se observar que para a regra ortogonal o algoritmo RF possui a maior acurácia (97,8%), sendo este o melhor classificador, com um KS de (51,92%), ou seja, o RF possui uma boa discriminação dos dados classificados.

A Tabela 4 mostra os resultados de acordo com o tipo de regra utilizada (RA, RC, RG, RT) como dados de entrada pelos algoritmos, pode-se observar que RF possui o melhor desempenho com uma acurácia (88,2%), para o regra RA, com um KS de (64,74%).

Os resultados da Tabela 5 mostram o tipo de regra utilizada (SW,KM,RY) como entrada e a métrica de desempenho, observar-se que o algoritmo como a maior precisão (90%) é RF para a regra SW, com um KS de (62,46%).

O algoritmo RF possui o maior KS, representa o nível de discriminação entre as classes, e é um indicativo de um bom desempenho do classificador.

Tabela 3 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI.

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
Ortogonal	**RF**	**0,978**	0,94	0,974	0,979	0,996	0,954	0,992	0,953	49,104	0,026	**51,923**
	NB	0,922	0,762	0,978	0,904	0,986	0,856	0,972	0,882	10,742	0,024	32,223
	SVM	0,909	0,756	0,912	0,909	0,968	0,826	0,937	0,820	10,099	0,097	36,662
	RNN	0,878	0,853	0,887	0,739	0,791	0,870	0,740	0,740	7,815	0,037	35,703
CB1	**RF**	**0,986**	0,955	0,988	0,986	0,998	0,971	0,995	0,973	72,194	0,013	**45,954**
	NB	0,974	0,929	0,965	0,977	0,996	0,947	0,991	0,942	42,941	0,036	26,215
	SVM	0,972	0,909	0,979	0,969	0,995	0,942	0,991	0,948	33,581	0,022	27,277
	RNN	0,879	0,857	0,888	0,739	0,793	0,872	0,744	0,744	7,637	0,035	35,587
CB2	**RF**	**0,987**	0,957	0,989	0,986	0,998	0,973	0,996	0,975	74,976	0,012	**45,808**
	NB	0,977	0,937	0,969	0,980	0,996	0,953	0,993	0,949	49,196	0,031	26,062
	SVM	0,973	0,915	0,978	0,972	0,996	0,945	0,991	0,950	35,796	0,023	27,308
	RNN	0,901	0,849	0,921	0,799	0,823	0,885	0,769	0,769	10,886	0,079	35,509
CB3	**RF**	**0,988**	0,963	0,985	0,988	0,998	0,974	0,996	0,974	87,267	0,015	**44,323**
	NB	0,976	0,932	0,968	0,978	0,996	0,950	0,992	0,946	45,084	0,033	26,054
	SVM	0,975	0,915	0,985	0,972	0,996	0,949	0,992	0,956	35,174	0,016	27,154
	RNN	0,885	0,816	0,910	0,771	0,793	0,863	0,726	0,726	9,153	0,103	35,587

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 4 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
RA	RF	**0,882**	0,873	0,706	0,956	0,953	0,780	0,906	0,662	16,536	0,308	**64,741**
	NB	0,678	0,473	0,991	0,550	0,907	0,753	0,814	0,541	2,200	0,016	63,379
	SVM	0,876	0,787	0,799	0,908	0,939	0,793	0,878	0,707	8,814	0,221	39,726
	RNN	0,850	0,795	0,871	0,695	0,742	0,833	0,666	0,666	6,173	0,087	35,703
RC	RF	**0,834**	0,813	0,565	0,946	0,917	0,666	0,835	0,511	10,549	0,460	**72,057**
	NB	0,754	0,550	0,913	0,688	0,885	0,686	0,769	0,601	2,951	0,125	51,696
	SVM	0,826	0,697	0,725	0,869	0,896	0,710	0,793	0,593	5,538	0,317	44,745
	RNN	0,821	0,749	0,848	0,646	0,694	0,799	0,597	0,597	4,923	0,116	35,781
RG	RF	**0,943**	0,937	0,866	0,975	0,988	0,900	0,976	0,841	37,763	0,138	**51,104**
	NB	0,872	0,702	0,997	0,819	0,981	0,823	0,962	0,816	5,802	0,004	43,931
	SVM	0,943	0,889	0,922	0,951	0,985	0,905	0,970	0,874	19,483	0,081	34,663
	RNN	0,909	0,885	0,918	0,800	0,840	0,901	0,803	0,803	10,070	0,036	35,587
RT	RF	**0,916**	0,857	0,859	0,940	0,972	0,858	0,944	0,799	14,607	0,150	**52,421**
	NB	0,892	0,748	0,956	0,866	0,959	0,839	0,918	0,822	7,206	0,051	38,523
	SVM	0,910	0,889	0,918	0,820	0,965	0,852	0,930	0,807	11,049	0,121	37,323
	RNN	0,895	0,845	0,913	0,783	0,813	0,879	0,758	0,758	9,731	0,074	35,820

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 5 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados UCI (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
SW	RF	**0,905**	0,897	0,766	0,963	0,971	0,826	0,943	0,729	21,269	0,243	**62,460**
	NB	0,802	0,602	0,993	0,721	0,952	0,749	0,903	0,714	3,633	0,010	50,653
	SVM	0,904	0,828	0,875	0,923	0,969	0,851	0,938	0,798	11,704	0,135	38,074
	RNN	0,882	0,858	0,891	0,746	0,798	0,875	0,750	0,750	7,929	0,037	35,859
KM	RF	**0,901**	0,885	0,767	0,958	0,971	0,822	0,942	0,725	18,354	0,243	**62,356**
	NB	0,859	0,684	0,985	0,805	0,961	0,807	0,921	0,790	5,179	0,018	45,551
	SVM	0,905	0,821	0,873	0,919	0,964	0,846	0,928	0,792	10,815	0,138	38,295
	RNN	0,915	0,796	0,861	0,932	0,896	0,827	0,793	0,793	12,893	0,149	35,826
RY	RF	**0,939**	0,925	0,865	0,971	0,986	0,894	0,972	0,836	29,690	0,139	**56,529**
	NB	0,914	0,792	0,975	0,888	0,979	0,872	0,958	0,863	9,844	0,028	38,579
	SVM	0,941	0,872	0,940	0,942	0,984	0,905	0,967	0,882	16,265	0,063	35,189
	RNN	0,885	0,839	0,902	0,759	0,797	0,870	0,740	0,740	8,546	0,070	35,703

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 6 – Comparação de desempenho dos métodos propostos com outros métodos com dados UCI.

Referência	Ano	Método	Acurácia maximo
Towell et al	1993	Neural networks	87%
Towell et al	1994	KBANN	90%
Ali et. al	1996	Multiple Descriptions	85,30%
Dietterich. T	1999	Decision tress	91,50%
Zupana et al	1999	Suboptimal heuristic algorithm (HINT)	93%
Meila et al.	2001	Mixture of trees	95%
Lumini et al	2006	Linear SVM	92,35%
Liu et al.	2007	Markov chain and Neural network	93,05%
Nasibov et al	2010	WPSSM and GA	95,69%
Damasevicius, R	2010	Grammar inference and SVM	92,10%
Romero et al.	2012	Kernel methodo	93%
Ferles et al	2013	SOHMMM	91%
Mandal, I.	2015	Adaptive Boosting	98,48%
	Neste estudo	**RF**	**98,88%**

A Tabela 6, se realiza uma comparação com estudos encontrados na literatura com os dados da UCI, e pode-se observar que nossa metodologia é igual ou superior a estudos anteriores com uma acurácia 98%.

- Classificação IE-N

Kolmogorov-Smirnov (KS) é uma medida de quão boa é a discriminação dos modelos, isto é, quão "fácil"os modelos conseguem separar as distribuições de bons e maus. Como normalmente estamos tentando construir função de score que possua um threshold ótimo, o KS fornece um valor de quão distantes está a CDF (cumulative distribution function) empíricas.

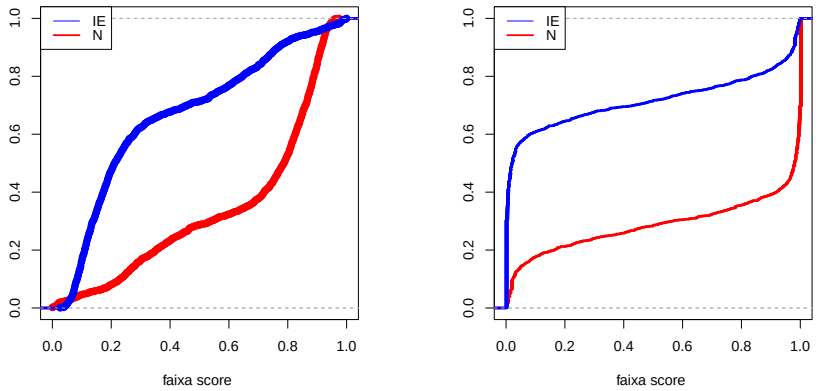

Figura 18 – Distribuções IE, N, para os algoritmos RF e SVM

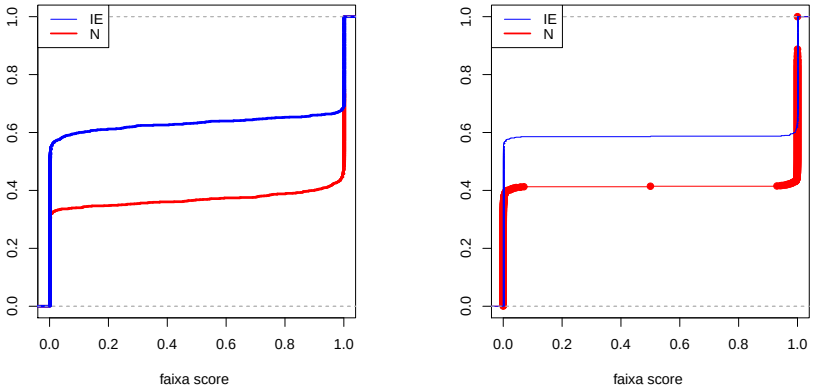

Figura 19 – Distribuções IE, N, para os algoritmos NB e RNN

Na Figura 19 KS é a diferença máxima entre duas funções de distribuição empírica, as linhas azul e vermelha correspondem a funções de distribuição empírica de EI e N respectivamente.

Os resultados com os dados UCI para classificação IE-N, são mostrados nas tabelas 7, 8, 9. A Tabela 7 mostra os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) empregados, esta comparação de acordo com o tipo de regra utilizada (ortogonal, CB1, CB2, CB3) como entrada de dados para o algoritmos, através de validação cruzada, os valores de desempenho para as diferentes métricas são obtidos. Pode notar-se que, pela regra ortogonal o algoritmo RF possui a maior acurácia (97%), sendo o melhor classificador, com un KS (52,75%), ou seja RF tem uma boa discriminação de dados classificados.

Na Tabela 8 são mostrados os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) empregados, esta comparação de acordo com o tipo de regra utilizada (RA, RC, RG, RT) como entrada para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Pode observar-se que, pela regra RA o algoritmo RF possui a maior acurácia (93%), sendo o melhor classificador, com un KS (47.90%).

A Tabela 9 mostra os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) empregados, esta comparação de acordo com o tipo de regra utilizada (RW, KM, RY,) como entrada para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Pode ver-se que, pela regra SW,KM,RY o algoritmo RF possui a maior acurácia, sendo o melhor classificador.

A Tabela 6, se realiza uma comparação com estudos encontrados na literatura com os dados da UCI, e pode-se observar que nossa metodologia é igual ou superior a estudos anteriores com uma acurácia 98%.

Tabela 7 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI.

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
Ortogonal	RF	**0,970**	0,911	0,968	0,971	0,994	0,939	0,988	0,939	33,686	0,033	**52,759**
	NB	0,936	0,808	0,984	0,920	0,989	0,884	0,977	0,905	15,580	0,017	30,661
	SVM	0,913	0,762	0,924	0,910	0,972	0,835	0,944	0,834	10,396	0,083	34,604
	RNN	0,886	0,849	0,900	0,761	0,802	0,875	0,749	0,749	8,655	0,056	35,625
CB1	RF	**0,982**	0,950	0,975	0,984	0,996	0,962	0,993	0,959	65,006	0,026	**44,696**
	NB	0,976	0,940	0,961	0,981	0,995	0,950	0,990	0,942	51,775	0,039	25,434
	SVM	0,965	0,888	0,975	0,962	0,993	0,930	0,985	0,937	26,094	0,026	29,101
	RNN	0,889	0,872	0,895	0,755	0,809	0,883	0,767	0,767	8,360	0,026	35,742
CB2	RF	**0,983**	0,958	0,971	0,987	0,995	0,964	0,990	0,957	76,203	0,030	**44,112**
	NB	0,977	0,945	0,960	0,982	0,994	0,952	0,988	0,943	56,043	0,041	25,165
	SVM	0,964	0,892	0,965	0,964	0,991	0,927	0,983	0,928	27,045	0,036	28,447
	RNN	0,910	0,901	0,914	0,795	0,845	0,907	0,815	0,815	10,435	0,014	35,703
CB3	RF	**0,983**	0,952	0,977	0,985	0,996	0,964	0,992	0,962	66,619	0,023	**44,435**
	NB	0,978	0,941	0,968	0,981	0,995	0,954	0,989	0,949	53,429	0,033	25,550
	SVM	0,964	0,886	0,971	0,961	0,992	0,927	0,984	0,933	25,231	0,030	29,008
	RNN	0,920	0,907	0,925	0,819	0,861	0,916	0,833	0,833	12,168	0,019	35,742

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 8 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
RA	RF	**0,930**	0,889	0,871	0,955	0,981	0,880	0,962	0,826	19,460	0,135	**47,960**
	NB	0,912	0,815	0,919	0,909	0,975	0,861	0,951	0,828	11,780	0,087	34,520
	SVM	0,927	0,849	0,914	0,932	0,976	0,880	0,952	0,847	13,638	0,092	35,861
	RNN	**0,916**	0,911	0,918	0,806	0,855	0,915	0,829	0,829	11,248	0,008	35,664
RC	RF	0,843	0,913	0,586	0,972	0,911	0,713	0,822	0,558	21,949	0,426	**64,680**
	NB	0,736	0,565	0,922	0,643	0,904	0,670	0,809	0,565	2,581	0,121	63,619
	SVM	0,844	0,786	0,729	0,901	0,901	0,756	0,803	0,630	7,370	0,300	43,338
	RNN	0,797	0,727	0,823	0,604	0,660	0,775	0,550	0,550	4,115	0,117	35,237
RG	RF	**0,930**	0,930	0,850	0,969	0,983	0,888	0,967	0,819	27,503	0,155	**48,878**
	NB	0,902	0,793	0,972	0,868	0,976	0,871	0,953	0,840	9,245	0,032	42,051
	SVM	0,927	0,893	0,882	0,948	0,976	0,887	0,952	0,830	17,077	0,124	36,486
	RNN	0,909	0,885	0,918	0,800	0,840	0,901	0,803	0,803	10,807	0,036	35,587
RT	RF	**0,879**	0,822	0,718	0,941	0,927	0,766	0,855	0,658	12,351	0,300	**81,457**
	NB	0,534	0,373	0,998	0,357	0,908	0,542	0,815	0,355	1,568	0,005	77,718
	SVM	0,859	0,721	0,801	0,882	0,919	0,759	0,839	0,683	6,808	0,226	44,797
	RNN	0,827	0,763	0,851	0,656	0,705	0,807	0,614	0,614	5,182	0,103	35,703

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 9 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados UCI (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
SW	RF	**0,806**	0,901	0,458	0,975	0,943	0,607	0,886	0,434	18,603	0,555	**65,887**
	NB	0,734	0,554	0,994	0,607	0,904	0,578	0,809	0,601	2,527	0,010	65,772
	SVM	0,842	0,762	0,762	0,882	0,930	0,762	0,859	0,644	6,526	0,270	44,211
	RNN	0,822	0,758	0,846	0,647	0,698	0,802	0,604	0,604	4,935	0,103	35,625
KM	RF	**0,831**	0,819	0,557	0,947	0,935	0,662	0,869	0,504	10,889	0,467	**73,001**
	NB	0,812	0,618	0,995	0,734	0,919	0,761	0,837	0,729	3,909	0,007	51,415
	SVM	0,830	0,674	0,828	0,831	0,924	0,743	0,849	0,659	4,899	0,206	43,621
	RNN	0,822	0,758	0,846	0,647	0,698	0,802	0,604	0,604	4,935	0,103	35,625
RY	RF	**0,958**	0,948	0,908	0,979	0,989	0,928	0,978	0,888	45,686	0,094	**48,797**
	NB	0,942	0,859	0,963	0,934	0,989	0,907	0,977	0,896	15,589	0,040	33,814
	SVM	0,957	0,913	0,945	0,962	0,987	0,929	0,974	0,908	25,316	0,057	33,280
	RNN	0,940	0,912	0,950	0,873	0,892	0,931	0,863	0,863	18,428	0,040	35,548

Valores em negrito refletem critérios com pontuações mais altas.

5.2 Conjuto de dados HS3D (Pollastro e Rampone 2003)

Para avaliar o desempenho, usaremos Sensibilidade, Especificidade KS como medida de avaliação.

- Classificação (EI-N)

Os resultados experimentais com os dados HS3D para classificação EI-N, são mostrados nas tabelas 10, 11, 12. Na Tabela 10 são mostrados os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (ortogonal, CB1, CB2, CB3) como entrada de dados para o algoritmos, através de validação cruzada, os valores de desempenho para as diferentes métricas são obtidos. Perceber-se que, para a regra ortogonal, o algoritmo RF possui a maior sensibilidades e especificidade, sendo este o melhor classificador, com um KS de (53.83%), RF possui uma boa discriminação de dados classificados.

Na Tabela 11 são mostrados os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (RA, RC, RG, RT) como entrada de dados para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Verificar-se que, para a regra RA,RC,RG,RT, o algoritmo NB possui a maior especificidade, sensibilidsde , sendo este o melhor classificador, para estas regras.

Na Tabela 12 são mostrados os resultados de comparação de desempenho dos algoritmos (SW, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (SW, KM, RY) como entrada de dados para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Constata-se que, para a regra SW,KM,RY, o algoritmo NB possui a maior especificidade e sensibilidade, sendo este o melhor classificador para estas regras.

Na Tabela 13, faz-se uma comparação com estudos encontrados na literatura com os dados HS3D, observar-se que neste estudo a Sensibilidade, Especificidade são iguais ou superiores com os estudos encontrados na literatura com 96% e 98% respectivamente.

Tabela 10 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D.

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
Ortogonal	**RF**	0,977	0,948	**0,968**	**0,980**	0,996	0,958	0,991	0,947	49,819	0,033	**53,839**
	NB	0,922	0,789	0,978	0,900	0,986	0,873	0,971	0,879	10,143	0,024	36,031
	SVM	0,908	0,796	0,894	0,913	0,966	0,842	0,932	0,807	10,445	0,116	38,945
	RNN	0,779	0,732	0,793	0,526	0,612	0,763	0,525	0,525	3,554	0,078	41,313
CB1	**RF**	0,984	0,969	**0,971**	**0,988**	0,997	0,970	0,995	0,959	83,645	0,029	**48,564**
	NB	0,975	0,948	0,962	0,980	0,996	0,983	0,992	0,942	48,445	0,039	29,450
	SVM	0,967	0,918	0,967	0,967	0,993	0,942	0,987	0,934	29,615	0,035	30,683
	RNN	0,768	0,708	0,787	0,510	0,593	0,748	0,496	0,496	3,333	0,100	41,323
CB2	**RF**	0,984	0,954	**0,991**	**0,982**	0,998	0,972	0,996	0,973	55,223	0,010	**48,635**
	NB	0,974	0,943	0,965	0,977	0,996	0,970	0,992	0,943	42,848	0,036	30,409
	SVM	0,910	0,805	0,890	0,917	0,965	0,845	0,931	0,807	10,799	0,120	37,833
	RNN	0,765	0,703	0,785	0,505	0,588	0,744	0,487	0,487	3,266	0,104	41,190
CB3	**RF**	0,980	0,942	**0,986**	**0,978**	0,998	0,963	0,996	0,963	43,954	0,015	**48,528**
	NB	0,974	0,932	0,979	0,973	0,996	0,973	0,992	0,951	36,173	0,022	30,628
	SVM	0,915	0,811	0,902	0,919	0,970	0,854	0,940	0,821	11,200	0,107	38,119
	RNN	0,762	0,693	0,784	0,502	0,582	0,738	0,477	0,477	3,226	0,116	41,167

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 11 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
RA	RF	0,902	0,815	0,835	0,928	0,960	0,825	0,921	0,763	11,589	0,178	37,901
	NB	0,687	0,474	**0,993**	0,567	0,913	0,776	0,826	0,560	2,295	0,012	**62,459**
	SVM	0,834	0,683	0,748	0,867	0,897	0,714	0,793	0,615	5,618	0,290	46,729
	RNN	0,792	0,735	0,809	0,547	0,627	0,772	0,545	0,545	3,858	0,091	41,362
RC	RF	0,784	0,749	0,336	0,957	0,836	0,464	0,672	0,293	7,791	0,694	36,117
	NB	0,457	0,337	**0,987**	0,252	0,825	0,502	0,650	0,239	1,320	0,051	**64,583**
	SVM	0,772	0,575	0,695	0,802	0,840	0,629	0,680	0,497	3,515	0,380	55,391
	RNN	0,724	0,645	0,749	0,446	0,527	0,697	0,394	0,394	2,569	0,138	41,049
RG	RF	0,855	0,812	0,619	0,945	0,916	0,702	0,833	0,564	11,272	0,403	34,570
	NB	0,480	0,343	**0,970**	0,294	0,810	0,470	0,620	0,264	1,374	0,103	**60,773**
	SVM	0,869	0,730	0,839	0,881	0,937	0,781	0,874	0,720	7,064	0,182	42,484
	RNN	0,806	0,750	0,823	0,571	0,648	0,787	0,573	0,573	4,252	0,089	41,214
RT	RF	0,831	0,729	0,619	0,912	0,904	0,670	0,808	0,531	7,040	0,418	34,102
	NB	0,632	0,428	**0,972**	0,502	0,883	0,594	0,766	0,473	1,951	0,056	**65,542**
	SVM	0,830	0,662	0,793	0,845	0,907	0,721	0,815	0,638	5,114	0,245	46,154
	RNN	0,796	0,747	0,812	0,554	0,636	0,779	0,558	0,558	3,965	0,080	41,520

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 12 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados HS3D (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
SW	RF	0,834	0,757	0,585	0,929	0,880	0,660	0,761	0,514	8,216	0,447	44,335
	NB	0,522	0,365	**0,992**	0,343	0,860	0,534	0,719	0,335	1,511	0,023	**58,309**
	SVM	0,856	0,709	0,811	0,873	0,927	0,756	0,853	0,684	6,409	0,217	43,771
	RNN	0,781	0,724	0,799	0,530	0,612	0,762	0,523	0,523	3,613	0,094	41,414
KM	RF	0,798	0,787	0,372	0,961	0,892	0,504	0,784	0,333	9,761	0,653	38,446
	NB	0,524	0,370	**0,999**	0,340	0,920	0,680	79,131	0,839	0,339	79,131	**53,636**
	SVM	0,824	0,652	0,778	0,841	0,896	0,710	0,792	0,619	4,906	0,264	48,149
	RNN	0,776	0,710	0,796	0,522	0,601	0,753	0,506	0,506	3,487	0,108	41,389
RY	RF	0,848	0,809	0,592	0,946	0,921	0,684	0,843	0,538	11,131	0,431	45,963
	NB	0,558	0,386	**0,998**	0,389	0,915	0,556	0,831	0,388	1,638	0,004	**56,708**
	SVM	0,864	0,714	0,851	0,869	0,932	0,776	0,864	0,720	6,516	0,172	43,648
	RNN	0,792	0,750	0,805	0,546	0,632	0,777	0,555	0,555	3,855	0,067	41,217

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 13 – Comparação de desempenho dos métodos propostos com outros métodos EI-N com dados HS3D.

Referência	Ano	Método	Sensibilidade	Especificidade
(Wei et al. 2013)	2013	DM-SVM	0,9517	0,9213
(Goel, Singh e Aseri 2015)	2015	MCM-SVM	0,9639	0,9346
(Meher et al. 2016)	2016	P1-RF	0,9556	0,9206
(Pashaei, Yilmaz e Aydin 2016)	2016	DMM2-SVM	0,9596	0,9406
(Pashaei et al. 2016)	2016	SCMM1-AdaBoost	0,9703	0,9428
	Neste estudo	**RF**	**0,968**	**0.980**

A Tabela 13, se realiza uma comparação com estudos encontrados na literatura com os dados da HS3D, e pode-se observar que nossa metodologia é igual ou superior a estudos anteriores com uma Sensibilidade e Especificidade 96%, 98%.

- Classificação (IE-N)

Os resultados experimentais com os dados HS3D para classificação IE-N, são mostrados nas tabelas 14, 15 e 16. A Tabela 14 são mostrados os resultados de comparação dos algoritmos (RF, NB, SVM, RNN) utilizados, esta comparação é feita de acordo com o tipo de regra utilizada (ortogonal, CB1, CB2, CB3) como entrada de dados para os algoritmos, através de validação cruzada os valores de desempenho para as diferentes métricas são obtidos. Perceber-se que, para a regra ortogonal, CB1,CB2,CB3, o algoritmo RF possui a maior sensibilidade (95%) e especificidade (97%), sendo este o melhor classificador, com um KS de (53,83%), RF possui uma boa discriminação de dados classificados.

A Tabela 15 são mostrados os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) utilizados, de acordo com o tipo de regra utilizada (RA, RC, RG, RT) como entrada de dados para o algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Verificar-se que, para a regra RA,RC,RG,RT, o algoritmo NB possui a maior especificidade, sensibilidade, sendo este o melhor classificador, para estas regras. NB possui uma boa discriminação de dados classificados conforme os valores de KS.

Os resultados na Tabela 16 mostram a comparação de desempenho dos algoritmos (SW, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (SW, KM, RY) como entrada de dados para o algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Constata-se que, para a regra SW,KM,RY, o algoritmo NB possui a maior especificidade e sensibilidade, sendo este o melhor classificador, para estas regras. NB possui uma boa discriminação de dados classificados em conformidade com os valores de KS.

Na Tabela 17, faz-se uma comparação com estudos encontrados na literatura com os dados HS3D, observar-se que neste estudo a Sensibilidade e Especificidade são iguais ou superiores com os estudos encontrados na literatura com 96% e 98% respectivamente.

Tabela 14 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados.

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
Ortogonal	RF	0,967	0,925	**0,957**	**0,970**	0,993	0,941	0,986	0,928	33,676	0,044	**54,365**
	NB	0,942	0,845	0,967	0,933	0,986	0,935	0,972	0,901	14,440	0,035	32,912
	SVM	0,912	0,812	0,888	0,921	0,967	0,849	0,935	0,809	11,212	0,122	35,714
	RNN	0,720	0,669	0,737	0,450	0,538	0,703	0,406	0,406	2,547	0,093	42,092
CB1	RF	0,979	0,959	0,964	**0,985**	0,997	0,961	0,993	0,948	62,928	0,037	**46,785**
	NB	0,969	0,937	0,956	0,974	0,992	0,967	0,984	0,931	37,501	0,045	29,485
	SVM	0,913	0,813	0,901	0,918	0,855	0,970	0,940	0,819	11,036	0,108	38,266
	RNN	0,723	0,670	0,741	0,455	0,542	0,705	0,411	0,411	2,588	0,095	42,070
CB2	RF	0,980	0,964	**0,961**	**0,987**	0,995	0,962	0,989	0,948	71,902	0,039	**46,515**
	NB	0,874	0,741	0,856	0,881	0,794	0,948	0,896	0,737	7,219	0,164	37,313
	SVM	0,915	0,818	0,901	0,920	0,971	0,858	0,941	0,822	11,343	0,107	37,949
	RNN	0,715	0,661	0,733	0,444	0,531	0,697	0,394	0,394	2,488	0,098	41,987
CB3	RF	0,979	0,967	**0,958**	**0,987**	0,994	0,963	0,987	0,946	75,898	0,042	**46,069**
	NB	0,873	0,740	0,852	0,881	0,948	0,792	0,895	0,733	7,179	0,168	36,933
	SVM	0,917	0,822	0,902	0,923	0,971	0,860	0,942	0,824	11,666	0,106	37,539
	RNN	0,826	0,799	0,835	0,610	0,692	0,817	0,634	0,634	4,864	0,043	42,090

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 15 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados HS3D (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
RA	RF	0,849	0,754	0,692	0,911	0,915	0,721	0,831	0,602	7,760	0,338	49,677
	NB	0,673	0,464	**0,974**	**0,554**	0,904	0,628	0,807	0,528	2,186	0,046	**62,450**
	SVM	0,839	0,684	0,805	0,853	0,913	0,740	0,827	0,658	5,484	0,228	46,525
	RNN	0,801	0,755	0,816	0,569	0,649	0,786	0,571	0,571	4,103	0,074	42,131
RC	RF	0,819	0,762	0,524	0,935	0,841	0,621	0,682	0,459	8,116	0,509	49,635
	NB	0,365	0,308	**0,996**	**0,116**	0,797	0,471	0,595	0,113	1,128	0,030	**60,112**
	SVM	0,800	0,633	0,697	0,840	0,851	0,663	0,701	0,537	4,364	0,361	52,865
	RNN	0,739	0,668	0,762	0,475	0,555	0,715	0,429	0,429	2,807	0,123	41,610
RG	RF	0,846	0,791	0,622	0,935	0,923	0,696	0,847	0,557	9,603	0,405	41,346
	NB	0,619	0,426	**0,990**	**0,473**	0,912	0,596	0,823	0,462	1,877	0,021	**58,315**
	SVM	0,851	0,704	0,820	0,863	0,925	0,758	0,851	0,683	6,015	0,208	44,771
	RNN	0,793	0,751	0,806	0,555	0,639	0,779	0,557	0,557	3,879	0,068	42,034
RT	RF	0,822	0,779	0,517	0,942	0,890	0,622	0,780	0,459	9,001	0,512	47,295
	NB	0,617	0,428	**0,991**	0,466	0,914	0,683	0,828	0,457	1,856	0,020	**58,813**
	SVM	0,826	0,672	0,749	0,856	0,894	0,709	0,788	0,605	5,212	0,293	49,096
	RNN	0,770	0,716	0,787	0,520	0,602	0,751	0,503	0,503	3,367	0,090	42,160

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 16 – Comparação de desempenho dos algoritmos na classificação IE-N de conjuntos de dados HS3D (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
SW	RF	0,783	0,658	0,483	0,901	0,833	0,556	0,666	0,384	4,898	0,574	45,601
	NB	0,523	0,366	**0,994**	0,343	0,858	0,654	0,717	0,338	1,514	0,017	**58,469**
	SVM	0,790	0,604	0,736	0,811	0,868	0,663	0,736	0,546	3,889	0,326	53,639
	RNN	0,731	0,677	0,749	0,465	0,551	0,713	0,426	0,426	2,697	0,096	42,148
KM	RF	0,746	0,687	0,189	0,966	0,792	0,296	0,584	0,155	5,607	0,840	49,432
	NB	0,521	0,366	**0,996**	0,338	0,865	0,655	0,730	0,334	1,505	0,012	**53,886**
	SVM	0,727	0,514	0,634	0,764	0,795	0,567	0,590	0,397	2,682	0,480	41,910
	RNN	0,674	0,591	0,701	0,389	0,469	0,646	0,292	0,292	1,976	0,157	41,987
RY	RF	0,912	0,867	0,816	0,950	0,966	0,841	0,933	0,766	16,406	0,193	58,055
	NB	0,690	0,479	**0,997**	0,569	0,961	0,647	0,921	0,565	2,316	0,006	**62,596**
	SVM	0,912	0,811	0,900	0,917	0,968	0,854	0,936	0,817	10,829	0,109	38,339
	RNN	0,871	0,840	0,881	0,694	0,760	0,861	0,721	0,721	7,054	0,046	42,107

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 17 – Comparação de desempenho dos métodos propostos com outros métodos IE-N com dados HS3D

Referência	Ano	Método	Sensibilidade	Especificidade
(Wei et al. 2013)	2013	DM-SVM	0,9264	0,9073
(Goel, Singh e Aseri 2015)	2015	MCM-SVM	0,9246	0,908
(Pashaei, Yilmaz e Aydin 2016)	2016	DMM2-SVM	0,9416	0,9191
(Pashaei et al. 2016)	2016	SCMM1-AdaBoost	0,9507	0.926
	Neste estudo	**RF**	**0,961**	**0,987**

A Tabela 17, se realiza uma comparação com estudos encontrados na literatura com os dados da HS3D, e pode-se observar que nossa metodologia é igual ou superior a estudos anteriores com uma Sensibilidade e Especificidade 96%, 98%.

5.3 Conjuto de dados cromossomo 21, segmento HS21C103 (CR21)

- Classificação (EI-N)

Os resultados experimentais com os dados HS3D para classificação EI-N, são mostrados nas tabelas 18, 19 e 20. Na Tabela 18 são mostrados os resultados de comparação dos algoritmos (RF, NB, SVM, RNN) utilizados, esta comparação é feita de acordo com o tipo de regra utilizada (ortogonal, CB1, CB2, CB3) como entrada de dados para os algoritmos, através de validação cruzada os valores de desempenho para as diferentes métricas são obtidos. Perceber-se que, para a regra ortogonal, CB1,CB2,CB3, para a regra ortogonal o algoritmo RF possui a maior sensibilidades (92%) e especificidade (99%), sendo este o melhor classificador, com um KS de (52,16%). RF possui uma boa discriminação de dados classificados.

Na Tabela 19 são mostrados os resultados de comparação de desempenho dos algoritmos (RF, NB, SVM, RNN) utilizados, de acordo com o tipo de regra utilizada (RA, RC, RG, RT) como entrada de dados para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Verifica-se que, para a regra RA,RC,RG,RT, o algoritmo NB possui a maior especificidade e sensibilidsde, sendo este o melhor classificador, para estas regras. NB possui uma boa discriminação de dados classificados conforme os valores de KS.

Os resultados na Tabela 20 mostram a comparação de desempenho dos algoritmos (SW, NB, SVM, RNN) utilizados, esta comparação de acordo com o tipo de regra utilizada (SW, KM, RY) como entrada de dados para os algoritmos, os valores de desempenho para as diferentes métricas são obtidos. Constata-se que, para a regra SW,KM,RY, o algoritmo RF possui a maior especificidade e sensibilidade, sendo este o melhor classificador, para estas regras. RF possui uma boa discriminação de dados classificados em conformidade com os valores de KS.

O algoritmo RF possui o maior KS, representa o nível de discriminação entre as classes, e é um indicativo de um bom desempenho do classificador. Nenhum trabalho foi obtido na literatura para comparar os resultados deste banco de dados.

Tabela 18 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21.

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
Ortogonal	**RF**	0,967	0,985	**0,921**	**0,992**	0,996	0,952	0,991	0,913	160,439	0,080	**52,168**
	NB	0,930	0,857	0,967	0,909	0,984	0,909	0,968	0,877	10,952	0,036	42,223
	SVM	0,903	0,877	0,849	0,933	0,964	0,862	0,928	0,782	12,937	0,162	43,743
	RNN	0,864	0,680	0,861	0,864	0,863	0,760	0,726	0,726	6,401	0,003	42,868
CB1	**RF**	0,984	0,990	**0,964**	**0,995**	0,998	0,977	0,996	0,959	223,214	0,036	**49,278**
	NB	0,969	0,963	0,948	0,980	0,995	0,955	0,991	0,928	53,332	0,053	36,895
	SVM	0,972	0,954	0,969	0,974	0,995	0,961	0,991	0,943	40,228	0,032	37,955
	RNN	0,935	0,837	0,918	0,940	0,929	0,876	0,858	0,858	15,624	0,024	42,687
CB2	**RF**	0,984	0,989	**0,967**	**0,994**	0,998	0,978	0,997	0,961	211,170	0,034	**49,576**
	NB	0,970	0,966	0,951	0,981	0,996	0,958	0,992	0,932	54,782	0,050	37,147
	SVM	0,975	0,960	0,972	0,977	0,996	0,966	0,992	0,949	46,041	0,029	38,303
	RNN	0,877	0,709	0,868	0,881	0,874	0,780	0,748	0,748	7,458	0,014	42,789
CB3	**RF**	0,982	0,988	**0,961**	**0,993**	0,998	0,974	0,995	0,954	168,277	0,039	**48,246**
	NB	0,965	0,963	0,939	0,980	0,995	0,951	0,989	0,919	51,369	0,062	36,901
	SVM	0,971	0,952	0,969	0,973	0,995	0,960	0,990	0,941	36,431	0,032	38,399
	RNN	0,827	0,620	0,806	0,834	0,820	0,700	0,640	0,640	4,924	0,034	42,846

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 19 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21 (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
RA	RF	0,850	0,918	0,636	0,968	0,947	0,751	0,895	0,604	21,315	0,376	64,895 [t]
	NB	0,652	0,511	**0,977**	**0,473**	0,899	0,669	0,798	0,450	1,920	0,043	**72,136**
	SVM	0,867	0,842	0,773	0,919	0,937	0,805	0,873	0,693	9,875	0,246	43,459
	RNN	0,859	0,678	0,835	0,867	0,851	0,748	0,703	0,703	6,374	0,037	42,800
RC	RF	0,778	0,905	**0,427**	**0,974**	0,915	0,579	0,829	0,402	18,979	0,587	**72,011**
	NB	0,759	0,610	0,928	0,665	0,889	0,735	0,778	0,593	2,834	0,107	57,152
	SVM	0,812	0,765	0,688	0,882	0,892	0,724	0,784	0,570	5,929	0,354	47,010
	RNN	0,802	0,579	0,767	0,814	0,790	0,660	0,581	0,581	4,139	0,057	42,880
RG	RF	0,921	0,956	**0,816**	**0,979**	0,987	0,880	0,974	0,795	43,648	0,188	**54,954**
	NB	0,871	0,741	0,996	0,802	0,980	0,848	0,960	0,798	5,714	0,005	50,181
	SVM	0,933	0,906	0,906	0,948	0,982	0,906	0,965	0,854	17,944	0,099	40,341
	RNN	0,918	0,773	0,952	0,906	0,929	0,853	0,859	0,859	10,316	-0,051	42,891
RT	RF	0,890	0,895	**0,787**	**0,948**	0,969	0,837	0,939	0,735	15,389	0,225	**55,483**
	NB	0,901	0,808	0,951	0,872	0,957	0,873	0,914	0,823	7,518	0,056	43,270
	SVM	0,903	0,862	0,870	0,921	0,960	0,866	0,921	0,791	11,158	0,141	42,867
	RNN	0,896	0,742	0,896	0,896	0,896	0,812	0,792	0,792	8,703	-0,001	42,925

Valores em negrito refletem critérios com pontuações mais altas.

Tabela 20 – Comparação de desempenho dos algoritmos na classificação EI-N de conjuntos de dados CR21 (Continuação).

Rules	Método	Acurácia	Precisão	Sensibilidade	Especificidade	AUC	Fcore	G	γ	ρ_+	ρ_-	KS
SW	RF	0,872	0,930	**0,695**	**0,970**	0,969	0,795	0,937	0,665	25,856	0,314	**64,425**
	NB	0,823	0,671	0,995	0,728	0,952	0,801	0,905	0,723	3,669	0,007	54,386
	SVM	0,900	0,869	0,851	0,928	0,965	0,860	0,929	0,779	12,028	0,161	43,209
	RNN	0,885	0,726	0,870	0,890	0,880	0,791	0,760	0,760	7,989	0,023	42,868
KM	RF	0,864	0,918	**0,680**	**0,966**	0,970	0,781	0,939	0,646	20,405	0,331	**64,946**
	NB	0,873	0,749	0,977	0,815	0,960	0,847	0,920	0,793	5,615	0,027	49,549
	SVM	0,896	0,855	0,855	0,919	0,961	0,855	0,923	0,774	10,818	0,158	43,988
	RNN	0,885	0,720	0,887	0,885	0,886	0,794	0,771	0,771	7,791	-0,003	42,891
RY	RF	0,920	0,950	**0,821**	**0,976**	0,985	0,881	0,970	0,796	36,411	0,184	**59,359**
	NB	0,911	0,827	0,963	0,882	0,977	0,888	0,953	0,845	9,885	0,041	43,909
	SVM	0,936	0,903	0,922	0,944	0,982	0,912	0,964	0,866	16,881	0,082	40,986
	RNN	0,906	0,755	0,924	0,900	0,912	0,831	0,824	0,824	9,314	0,027	42,823

Valores em negrito refletem critérios com pontuações mais altas.

6 Conclusões

O desempenho destes algoritmos foi avaliado a partir de métricas decorrentes da matriz de confusão, que registra em suas linhas e colunas os erros e acertos da classificação. No capítulo 4 foram apresentados os indicadores estatísticos, calculados sobre as matrizes de confusão dos algoritmos de classificação, para avaliar o desempenho e a intensidade da discriminação na classificação de *Introns-Exons*, bem como a capacidade de distinguir as classes.

Os resultados obtidos pelos modelos de classificação com as diferentes transformações binárias mostraram a eficiência metodológica aplicada a este trabalho, resultados estes que nos permitem fazer afirmações que confirmam investigações prévias. Os experimentos realizados nos conjuntos de dados mostram que a metodologia proposta possui a capacidade de caracterizar e discriminar dados em problemas biológicos.

O algoritmo mais apropriado para a classificação das estruturas de Exon e Itrons na cadeia DNA, de acordo com a metodologia proposta é RF, devido ao seu bom desempenho e nível de discriminação dos dados, A precisão média alcançada é superior a 96% em comparação com as abordagens na literatura.

A distância de Kolgomorov-Smirnov foi apresentada como um indicador de discriminaąão para o problemas de classificação binário, obtendo resultados favoráveis. A metodologia proposta dá bons resultados com as transformações ortogonais e binárias, isto é, KS, é um indicador de discriminação da classificação, e ao mesmo tempo é um indicativo de boa acurácia do classificador. Altos valores da distância de Kolgomorov, são indicativos do bom desempenho de um algoritmo de classificação.

Referências Bibliográficas

ADEODATO, P. J.; MELO, S. B. On the equivalence between kolmogorov-smirnov and roc curve metrics for binary classification. **arXiv preprint arXiv:1606.00496**, 2016.

BARI, A.; REAZ, M. R.; JEONG, B.-S. Effective dna encoding for splice site prediction using svm. **MATCH Commun. Math. Comput. Chem**, v. 71, p. 241–258, 2014.

BATEN, A. K. et al. Biological sequence data preprocessing for classification: A case study in splice site identification. In: SPRINGER. **International Symposium on Neural Networks**. [S.l.], 2007. p. 1221–1230.

BEKKAR, M.; DJEMAA, H. K.; ALITOUCHE, T. A. Evaluation measures for models assessment over imbalanced datasets. **Iournal Of Information Engineering and Applications**, v. 3, n. 10, 2013.

BIAU, G.; SCORNET, E. A random forest guided tour. **Test**, Springer, v. 25, n. 2, p. 197–227, 2016.

BIGGERSTAFF, B. J. Comparing diagnostic tests: a simple graphic using likelihood ratios. **Statistics in medicine**, Wiley Online Library, v. 19, n. 5, p. 649–663, 2000.

BIN, W.; JING, Z. A novel artificial neural network and an improved particle swarm optimization used in splice site prediction. **J Appl Computat Math**, n. 166, 2014.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BURGE, C.; KARLIN, S. Prediction of complete gene structures in human genomic dna1. **Journal of molecular biology**, Elsevier, v. 268, n. 1, p. 78–94, 1997.

CERVANTES, J.; LI, X.; YU, W. Splice site detection in dna sequences using a fast classification algorithm. In: IEEE. **2009 IEEE International Conference on Systems, Man and Cybernetics**. [S.l.], 2009. p. 2683–2688.

CHEN, T.-M.; LU, C.-C.; LI, W.-H. Prediction of splice sites with dependency graphs and their expanded bayesian networks. **Bioinformatics**, Oxford University Press, v. 21, n. 4, p. 471–482, 2004.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

DAMAŠEVIČIUS, R. Feature representation of dna sequences for machine learning tasks. In: **Proc. of Fifth Int. Workshop on Computational Systems Biology (WCSB 2008)**. [S.l.: s.n.], 2008. p. 11–13.

DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. **Neural computation**, MIT Press, v. 10, n. 7, p. 1895–1923, 1998.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. [S.l.]: John Wiley & Sons, 2012.

FRIEDMAN, J. H. A recursive partitioning decision rule for nonparametric classification. **IEEE Transactions on Computers**, IEEE, n. 4, p. 404–408, 1977.

GOEL, N.; SINGH, S.; ASERI, T. C. An improved method for splice site prediction in dna sequences using support vector machines. **Procedia Computer Science**, Elsevier, v. 57, p. 358–367, 2015.

GOLDSTEIN, B. A.; POLLEY, E. C.; BRIGGS, F. B. Random forests for genetic association studies. **Statistical applications in genetics and molecular biology**, De Gruyter, v. 10, n. 1, 2011.

GRIFFITHS, A. J. et al. Introdução à genética. In: **Introdução à genética**. [S.l.: s.n.], 2006.

HAYKIN, S. **Neural networks: a comprehensive foundation**. [S.l.]: Prentice Hall PTR, 1994.

HO, L. S.; RAJAPAKSE, J. C. Splice site detection with a higher-order markov model implemented on a neural network. **Genome Informatics**, Japanese Society for Bioinformatics, v. 14, p. 64–72, 2003.

HUANG, J. et al. An approach of encoding for prediction of splice sites using svm. **Biochimie**, Elsevier, v. 88, n. 7, p. 923–929, 2006.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the Eleventh conference on Uncertainty in artificial intelligence**. [S.l.], 1995. p. 338–345.

KAMATH, U. et al. An evolutionary algorithm approach for feature generation from sequence data and its application to dna splice site prediction. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 9, n. 5, p. 1387–1398, 2012.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. **Ijcai**. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.

LAWSON, D. J.; FALUSH, D. Population identification using genetic data. **Annual review of genomics and human genetics**, Annual Reviews, v. 13, p. 337–361, 2012.

LEWIN, B. **Genes VIII**. [S.l.]: Oxford University Press, New York, 2003.

LI, J. et al. High-accuracy splice sites prediction based on sequence component and position features. **Genetics and Molecular Research**, v. 11, n. 3, p. 3432–3451, 2012.

LOI, H. S.; RAJAPAKSE, J. Splice site detection with neural networks/markov model hybrids. In: IEEE. **Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on**. [S.l.], 2002. v. 5, p. 2249–2253.

LOPES, H. S.; LIMA, C. R. E.; MURATA, N. J. A configware approach for high-speed parallel analysis of genomic data. **Journal of Circuits, Systems, and Computers**, World Scientific, v. 16, n. 04, p. 527–540, 2007.

MAJI, S.; GARG, D. Hidden markov model for splicing junction sites identification in dna sequences. **Current Bioinformatics**, Bentham Science Publishers, v. 8, n. 3, p. 369–379, 2013.

MANDAL, I. A novel approach for predicting dna splice junctions using hybrid machine learning algorithms. **Soft Computing**, Springer, v. 19, n. 12, p. 3431–3444, 2015.

MARTINEZ, I. F. **Tratamiento y resolución de las descripciones definidas y su aplicación en sistemas de extracción de información**. Tese — Departamento de Estadística e Investigación Operativa. Universidad de Valladolid, Junio 2012.

MEHER, P. K. et al. Identification of donor splice sites using support vector machine: a computational approach based on positional, compositional and dependency features. **Algorithms for molecular biology**, BioMed Central, v. 11, n. 1, p. 16, 2016.

MEHER, P. K.; SAHU, T. K.; RAO, A. R. Prediction of donor splice sites using random forest with a new sequence encoding approach. **BioData mining**, BioMed Central, v. 9, n. 1, p. 4, 2016.

MINING, W. I. D. Data mining: Concepts and techniques. **Morgan Kaufinann**, 2006.

NASSA, T.; SINGH, S.; GOEL, N. Splice site detection in dna sequences using probabilistic neural network. **International Journal of Computer Applications**, Citeseer, v. 76, n. 4, 2013.

NEWMAN, D. J. Uci repository of machine learning database. **http://www. ics. uci. edu/˜ mlearn/MLRepository. html**, 1998.

PASHAEI, E.; AYDIN, N. Markovian encoding models in human splice site recognition using svm. **Computational biology and chemistry**, Elsevier, v. 73, p. 159–170, 2018.

PASHAEI, E.; OZEN, M.; AYDIN, N. Random forest in splice site prediction of human genome. In: SPRINGER. **XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016**. [S.l.], 2016. p. 518–523.

PASHAEI, E.; OZEN, M.; AYDIN, N. Splice site identification in human genome using random forest. **Health and Technology**, Springer, v. 7, n. 1, p. 141–152, 2017.

PASHAEI, E.; YILMAZ, A.; AYDIN, N. A combined svm and markov model approach for splice site identification. In: IEEE. **Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on**. [S.l.], 2016. p. 200–204.

PASHAEI, E. et al. A novel method for splice sites prediction using sequence component and hidden markov model. In: IEEE. **Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the**. [S.l.], 2016. p. 3076–3079.

PENALVA, L.; ZORIO, D. A leitura do dna como é processada a informação dos genes. **Ciência Hoje**, SOCIEDADE BRASILEIRA PARA O PROGRESSO DA CIENCIA, p. 34–39, 2001.

PERTEA, M.; LIN, X.; SALZBERG, S. L. Genesplicer: a new computational method for splice site prediction. **Nucleic acids research**, Oxford University Press, v. 29, n. 5, p. 1185–1190, 2001.

POLLASTRO, P.; RAMPONE, S. Hs3d: homosapiens splice site data set. **Nucleic Acids Research**, n. Annual Database, 2003.

QI, L.; JIANG, H. Semismooth karush-kuhn-tucker equations and convergence analysis of newton and quasi-newton methods for solving these equations. **Mathematics of Operations Research**, INFORMS, v. 22, n. 2, p. 301–325, 1997.

RÄTSCH, G. et al. Improving the caenorhabditis elegans genome annotation using machine learning. **PLoS Computational Biology**, Public Library of Science, v. 3, n. 2, p. e20, 2007.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Malaysia; Pearson Education Limited„ 2016.

SADAVA, D.; PURVES, W. H. **Vida/Life: La ciencia de la biologia/The Science of Biology**. [S.l.]: Ed. Médica Panamericana, 2009.

SALZBERG, S. L. Genome re-annotation: a wiki solution? **Genome biology**, BioMed Central, v. 8, n. 1, p. 102, 2007.

SILVA, I. N. D. et al. Artificial neural networks. **Cham: Springer International Publishing**, Springer, 2017.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: SPRINGER. **Australasian joint conference on artificial intelligence**. [S.l.], 2006. p. 1015–1021.

SONNENBURG, S. et al. Accurate splice site prediction using support vector machines. In: BIOMED CENTRAL. **BMC bioinformatics**. [S.l.], 2007. v. 8, n. 10, p. S7.

STADEN, R. Computer methods to locate signals in nucleic acid sequences. Oxford University Press, 1984.

UTGO, P. E.; CLOUSE, J. A kolmogorov-smirnoff metric for decision tree induction. **Amherst, Massachusetts: University of Massachusetts, Department of Computer Science**, Citeseer, 1996.

VISA, S. et al. Confusion matrix-based feature selection. In: . [S.l.: s.n.], 2011.

WEI, D. et al. A novel splice site prediction method using support vector machine. **Journal of Computational Information Systems**, v. 9, n. 20, p. 8053–8060, 2013.

WU, C. H.; MCLARTY, J. W. **Neural networks and genome informatics**. [S.l.]: Elsevier, 2012. v. 1.

YEO, G.; BURGE, C. B. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. **Journal of computational biology**, Mary Ann Liebert, Inc., v. 11, n. 2-3, p. 377–394, 2004.

ZHANG, M.; MARR, T. A weight array method for splicing signal analysis. **Bioinformatics**, Oxford University Press, v. 9, n. 5, p. 499–509, 1993.

ZHANG, Q. et al. Splice sites prediction of human genome using length-variable markov model and feature selection. **Expert Systems with Applications**, Elsevier, v. 37, n. 4, p. 2771–2782, 2010.

ZHANG, Y. et al. Splice site prediction using support vector machines with a bayes kernel. **Expert Systems with Applications**, Elsevier, v. 30, n. 1, p. 73–81, 2006.