Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Investigando o desempenho de métodos de Aprendizado de Máquina para predição de RNAs não-codificadores utilizando construção in silico de dados artificiais

Mirele Carolina Souza F. Costa

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientadora
Prof[a]. Dr[a]. Maria Emília Machado T. Walter

Brasília
2020

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Investigando o desempenho de métodos de Aprendizado de Máquina para predição de RNAs não-codificadores utilizando construção in silico de dados artificiais

Mirele Carolina Souza F. Costa

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Profª. Drª. Maria Emília Machado T. Walter (Orientadora)
CIC/UnB

Profª. Drª. Célia Ghedini Ralha
Universidade de Brasília

Prof. Dr. Nalvo Franco de Almeida Junior
Universidade Federal do Mato Grosso do Sul

Profª. Drª. Genaína Nunes Rodrigues
Coordenadora do Programa de Pós-graduação em Informática

Brasília, 18 de dezembro de 2020

Dedicatória

Dedico este trabalho a minha mãe Maria Cecília, que mesmo com toda dificuldade sempre apoiou os meus estudos e sempre me incentivou. Ao meu esposo Rafael Moura por sempre estar ao meu lado. Ao meu sogro Jorge Luiz e sogra Ana Martins pelo apoio e motivação. A professora Maria Emília que me orientou desde 2018 com muita paciência e dedicação. A minha amiga Francielle Marques por sempre acreditar no meu potencial e me apoiar. Ao meu amigo Guilherme Enéas que esteve ao meu lado desde o início do mestrado. E por fim, as minha amigas Ana Paula, Bianca, Fernanda e Gabrielly que mesmo a distância estiveram ao meu lado.

Agradecimentos

Agradeço primeiramente a Deus por abençoar minha jornada no mestrado. Aos professores do programa de Pós-graduação em Informática da UnB que colaboraram para minha pesquisa acadêmica, em especial a professora Maria Emília, que contribuiu para meu crescimento pessoal e profissional. Ao grupo de bioinformática da UnB, em especial o professor Waldeyr Silva, professor João Victor, Daniel Souza e Deborah Bambil pela amizade, colaboração, ensinamentos, e todo apoio nestes tempos de muito estudo. Agradeço também a colaboração do grupo de pesquisa do professor Peter Stadler, da Universidade de Leipzig, na Alemanha. A Capes pelo financiamento do projeto. E a todos que contribuíram para minha formação.

Resumo

Métodos de aprendizado de máquina (AM) são frequentemente usados para predizer diferentes classes de RNAs não-codificadores (ncRNAs), como microRNAs ou snoRNAs. Nos métodos de AM que usam o paradigma de aprendizagem supervisionada, atributos ou características (em inglês, *features*) são extraídas dos dados de entrada e usados em um classificador, nas diferentes etapas desses métodos. No entanto, os métodos de AM não foram usados com tanto sucesso quanto se esperava para busca de homologia em ncRNAs. Neste contexto, é relevante medir o desempenho de métodos de AM para verificar seu poder de predição, tanto de sequências evolutivamente próximas quanto daquelas mais distantes. Uma avaliação sistemática de métodos de AM para predição de homologia requer conjuntos de testes, grandes, controlados e conhecidos. Assim, devem ser criadas formas para construir grandes conjuntos de dados artificiais de forma que se aproxime o máximo possível dos ncRNAs reais. Nesta dissertação, inicialmente, descrevemos uma forma de gerar conjuntos arbitrariamente grandes e diversos de sequências de ncRNAs, com base em uma evolução artificial, das duas classes principais de snoRNAs, C/D box e H/ACA box snoRNAs. Em seguida, esses dados artificiais são usados para avaliar o poder de predição de snoRNAs, em um genoma de cordados, de três métodos supervisionados de AM - Máquina de Vetores de Suporte (em inglês, *Support Vector Machine - SVM*), Redes Neurais Artificiais (em inglês, *Artificial Neural Networks - ANN*) e Floresta Aleatória (em inglês, *Random Forest - RF*). Nossos resultados indicam que as abordagens de AM podem de fato ser competitivas para a busca de homologia em ncRNAs, dependendo do conhecimento de *features* biológicas, extraídas dos dados, que são a entrada desses métodos de AM. Para a mutação de substituição, os classificadores SVM e ANN obtiveram excelentes desempenhos para conjuntos de dados com mutações de bases de 10%, 20%, 30% e 40% de diferença relativamente aos snoRNAs originais. No entanto, para conjuntos de dados com mutações de 50%, os classificadores não alcançaram um desempenho tão bom. Para H/ACA box, o desempenho dos classificadores de AM foram equivalentes, tanto utilizando um número maior de *features* biológicas conhecidas quanto um número reduzido delas. Para a mutação de inserção, quanto maior a porcentagem de mutação, menor o desempenho dos três classificadores - SVM, ANN e RF. Para os dois tipos de

snoRNAs, os tamanhos das sequências mostraram ser características importantes para a predição correta. Além disso, os métodos de AM apresentaram resultados de predição melhores, quando comparados a métodos que usam diretamente as sequências primárias de ncRNAs, como BLAST.

Palavras-chave: predição de RNAs não-codificadores (ncRNAs), RNAs nucleolares pequenos (snoRNAs), genoma de cordados, aprendizado de máquina, avaliação de desempenho de métodos de aprendizado de máquina, criação de snoRNAs artificiais.

Abstract

Machine learning (ML) methods are often used to predict different classes of non-coding RNAs (ncRNAs), such as microRNAs or snoRNAs. In ML methods that use the supervised learning paradigm, attributes or features are extracted from the input data and used in a classifier, in the different steps of these methods. However, ML methods have not been used as successfully as expected to search for homology in ncRNAs. In this context, it is relevant to measure the performance of ML methods in order to verify their predictive power, both for evolutionary close sequences and those that are more distant. A systematic evaluation of ML methods for homology prediction requires large, controlled and known sets of tests. Thus, large sets of artificial data have to be created such that their stored sequences are as close as possible to real ncRNAs. In this dissertation, initially, we describe a way to generate arbitrarily large and diverse sets of ncRNA sequences, based on an artificial evolution, of the two main classes of snoRNAs, C/D box and H/ACA box. Then, these artificial data are used to evaluate the predictive power of snoRNAs, in a chordate genome, of three supervised methods of ML - Support Vector Machine (SVM), Artificial Neural Networks (ANN) and Random Forest (RF). Our results indicate that ML approaches can in fact be competitive to predict homology for ncRNAs, depending on the knowledge of biological features, extracted from the data, which are the input of these ML methods. For the substitution mutation, the SVM and ANN classifiers achieved excellent performances for data sets with base mutations of 10%, 20%, 30% and 40% distant from the original snoRNAs. However, for data sets with mutations of 50%, the classifiers did not perform so well. For H/ACA box, the performance of the ML classifiers were equivalent, using a larger number of known biological features as well as a reduced number of them. For the insertion mutation, the higher the percentage of mutation, the lower the performance of the three classifiers - SVM, ANN and RF. For both types of snoRNAs, the size of the sequences proved to be an important characteristic for correct prediction. In addition, ML methods presented better prediction results, when compared to methods that directly use primary ncRNA sequences, such as BLAST.

Keywords: Non-coding RNA (ncRNA) prediction, Small Nucleolar RNAs (snoRNAs),

chordate genome, machine learning, perfomance evaluation of machine learning methods, construction of artificial snoRNAs

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Décadas após a proposição, por Watson e Crick em 1953 [10], do modelo de dupla hélice para o DNA, a quantidade de informações genômicas cresceu exponencialmente devido aos avanços das técnicas de sequenciamento em laboratórios de Biologia Molecular. Especialmente na décaca de 90, o projeto Genoma Humano gerou grandes volumes de informações [11], obtidas a partir do sequenciamento do DNA humano.

Essas técnicas foram um fator decisivo para consolidar a Bioinformática como uma importante área do conhecimento científico. As técnicas de sequenciamento exigiram a construção de ferramentas computacionais sofisticadas, que permitiram a análise e a resolução de diversas perguntas relacionadas à estrutura do DNA [12, 13]. A Bioinformática é capaz de relacionar, armazenar dados biológicos e reconhecer padrões que seriam improváveis de serem analisados sem o auxílio dos métodos computacionais.

A Biologia Molecular [1] contribuiu para investigar substâncias orgânicas descobertas inicialmente no núcleo celular, denominadas ácidos nucléicos. Existem dois tipos de ácidos nucléicos, o DNA (ácido desoxirribonucleico) e o RNA (ácido ribonucleico). Essas macromoléculas são fundamentais, pois armazenam informações e participam de muitos processos importantes nas células, por exemplo, a síntese das proteínas, a replicação das células e mecanismos de transmissão das características hereditárias [4].

Durante muito tempo, os pesquisadores acreditavam que as funções dos RNAs eram restritas apenas à síntese de proteínas. Entretanto, estudos posteriores apontaram que algumas enzimas são capazes de utilizar o RNA para produzir DNA. Além disso, pesquisas revelaram que apenas 2% do genoma humano é constituído por regiões codificadoras de proteínas, ou seja, a maior parte do DNA não codifica proteínas. Cerca de 98% do que é transcrito pelo genoma humano é constituído de RNAs não-codificadores (em inglês, *non-coding RNA* - ncRNAs) [14, 15]. As informações que codificam transcritos estão contidas em regiões do DNA denominadas genes. Os genes que são traduzidos em proteínas são também chamados de genes codificadores de proteínas.

No entanto, existem regiões do DNA que não codificam proteínas, denominadas ncRNAs. Assim, os ncRNAs exercem papéis importantes em muitas atividades celulares, não sendo apenas intermediários na transferência de informação genética do DNA para sintetizar proteínas, como se pensava a princípio. As moléculas de ncRNAs controlam diversos processos biológicos, como iniciação da tradução, manutenção de células tronco, desenvolvimento de cérebro e músculos, além de serem usados como marcadores moleculares para métodos de diagnósticos específicos. A descoberta de ncRNAs causador de uma doença, em um organismo, pode auxiliar no desenvolvimento de tratamentos ou fármacos [16].

Existem muitas classes distintas de ncRNAs, cada qual com função específica. As funções dos ncRNAs dependem de sua estrutura espacial, composição de sequência e comprimento. Nesta dissertação, nosso foco é em um tipo específico de ncRNA, chamado de *small nucleolar RNAs* (snoRNAs). Eles formam uma grande classe de ncRNAs com comprimentos variando de 60 a 300 nucleotídeos, que se enquadram em duas subclasses funcionais e estruturalmente distintas, C/D *box* e H/ACA *box* snoRNAs [17] .

Por outro lado, métodos de AM são um recurso poderoso da Inteligência Artificial para análise de dados. Esses métodos são capazes de aprender a partir de dados e automatizam a construção de modelos utilizados para diversas aplicações. Em particular, métodos de AM são frequentemente usados para predição de diferentes classes de ncRNAs [18, 19, 3, 20, 21, 22], como microRNAs ou snoRNAs. O uso de características da estrutura secundária de ncRNAs em métodos de AM vem sendo utilizado recentemente para a predição de ncRNAs [23]. Por exemplo, a ferramenta SnoReport2.0 [3] utiliza um classificador para snoRNAs. Ele extrai determinadas subsequências primárias, além de características espaciais (sendo essas últimas previstas por dobramento termodinâmico [24]), de uma sequência de consulta e emprega uma Máquina de Vetor de Suporte para classificação das duas classes principais de snoRNAs: H/ACA *box* e C/D *box*.

No entanto, métodos de AM não obtiveram o sucesso esperado para tarefas de inferência de homologia[1] em ncRNAs.

Para uma avaliação sistemática de métodos de AM para inferência de homologia em ncRNAs deve-se construir conjuntos de teste grandes, controlados e conhecidos. Assim, deve-se propor formas de construção de grandes conjuntos de dados artificiais, que sejam o mais próximos possíveis de ncRNAs conhecidos. Nesta dissertação, inicialmente descrevemos uma forma de gerar conjuntos de sequências arbitrariamente grandes e diversas, com base em evolução artificial. Construímos árvores *in silico*, a partir de mutações em sequências de snoRNAs, cuidadosamente escolhidas. Para análise sistemática dos métodos de AM supervisionados, estudamos o desempenho de Máquinas de Vetores de Suporte

[1]Homologia é o estudo biológico das semelhanças entre estruturas de diferentes organismos que possuem a mesma origem ontogenética e filogenética

(em inglês, *Support Vector Machine - SVM*), Redes Neurais Artificiais (em inglês, *Artificial Neural Networks - ANN*) e Florestas Aleatórias (em inglês, *Random Forest - RF*) para inferência de homologia em snoRNAs.

1.1 Motivação

De forma genérica, existem duas classes de métodos para predizer ncRNAs em dados genômicos. A primeira realiza a inferência de homologia por similaridade de sequência, com uma sequência de consulta específica (um ncRNA) e um banco de dados contendo sequências de ncRNAs. Porém, para a maioria dos ncRNAs, seu desempenho é muito baixo [25]. Atualmente, sabe-se que a estrutura espacial é fundamental para a realização da função da maioria dos ncRNAs. Em geral, a estrutura espacial de uma sequência de ncRNA é mais bem conservada do que sua sequência primária. Assim, os métodos de inferência de homologia para ncRNAs consideram a similaridade espacial (também conhecida como estrutura secundária). Ferramentas como Infernal [26] realmente produzem predições melhores do que os métodos que usam apenas sequência primária, como Blast [27], por exemplo [28]. No entanto, apenas homólogos de ncRNAs conhecidos podem ser encontrados.

A segunda classe inclui certos ncRNAs, como transferRNAs, microRNAs e snoRNAs, pertencentes a famílias maiores, que compartilham função e biogênese e podem ser reconhecidos por conjuntos de características de sequência e estrutura bem conhecidos. Pode-se predizer membros de uma classe específica de ncRNAs modelando um problema de classificação, normalmente solucionado por métodos de AM [23, 29, 30].

Identificar ncRNAs em organismos é uma tarefa importante. Entretanto, no tocante à inferência de ncRNAs considerando grandes distâncias evolutivas, os métodos de AM não atingiram o sucesso esperado. Um aspecto importante para uma investigação mais sistemática sobre o desempenho dos métodos de AM, como uma alternativa à comparação direta de sequências é a necessidade de construir conjuntos de treinamento e teste suficientemente grandes, diversos e com ncRNAs conhecidos. Além disso, esses conjuntos precisam ter uma abrangência ampla que inclui distâncias evolutivas de sequências homólogas, que divergiram além do limite de predição realizado por métodos de alinhamento de sequência. Nesse sentido, a construção cuidadosa de dados artificiais é uma abordagem promissora, e pode contribuir para uma avaliação sistemática dos métodos de AM na busca de homologia para ncRNAs.

1.2 Problema

Os métodos de AM não obtiveram o sucesso esperado para tarefas de inferência de homologia, por exemplo, para predizer classes de ncRNAs considerando grandes distâncias evolutivas. Uma solução eficiente para esse problema de predição de ncRNAs forneceria uma alternativa para a busca de homologia. As tentativas de usar AM para essa tarefa, têm sido desencorajadoras, embora isso possa ser uma consequência de conjuntos de treinamento muito pequenos. A construção de grandes conjuntos de dados artificiais de ncRNAs não é uma tarefa simples, pois ainda não são conhecidos grandes volumes de ncRNAs confirmados em laboratório, que possam ser utilizados em avaliações de desempenho. Portando, são necessários métodos para construção de grandes conjuntos de dados artificiais, que sejam o mais próximos possíveis de ncRNAs conhecidos. Esses dados artificiais poderiam ser utilizados para avaliação sistemática de desempenho de métodos de AM.

1.3 Objetivos

Esta dissertação tem como objetivo geral criar conjuntos de ncRNAs artificiais, que cobrem diferentes distâncias evolutivas, e avaliar de forma sistemática os métodos de AM na busca de homologia de ncRNAs usando esses dados artificiais. Mais detalhadamente:

1. Inicialmente, será proposto um método para gerar conjuntos de dados arbitrariamente grandes e diversos de ncRNAs artificiais, abrangendo diferentes distâncias evolutivas, usando snoRNAs como exemplo. A ideia chave é simular a evolução dos ncRNAs ao longo de uma árvore filogenética artificial, gerada aleatoriamente. Para isso, será utilizada uma ferramenta de predição de ncRNAs de interesse - no nosso caso, snoRNAs. Escolhemos o SnoReport 2.0 para predição de snoRNAs. A árvore artificial conterá sequências que, apesar de mutadas, foram aceitas pela ferramenta. Portanto, a árvore conterá snoRNAs artificiais, com sequências sucessivamente mais divergentes em termos evolutivos, mas que ainda são reconhecidas como snoRNAs pelo SnoReport 2.0;

2. Em seguida, utilizando os snoRNAs artificiais criados que serão inseridos em genomas para produzir dados o mais próximo possível de snoRNAs reais, serão treinados e avaliados os seguintes métodos de AM para busca de homologia em snoRNAs - SVM, RF e ANN.

Os objetivos específicos são:

1. Construir árvores, a partir de mutações pontuais (substituição, inserção e deleção) *in silico* em sequências de snoRNAs para as duas classes de snoRNAs: C/D *box* e H/ACA *box*;
2. Selecionar os conjuntos de dados com 10%, 20%, 30%, 40% e 50% de mutação em sequências de snoRNAs artificiais criadas pela árvore de mutação;
3. Extrair das sequências dos conjuntos de dados *features* revelantes;
4. Construir bases de dados balanceadas, a serem utilizadas para avaliar o desempenho dos classificadores de AM;
5. Testar e avaliar o desempenho dos classificadores de AM - SVM, RF e ANN, para predizer snoRNAs.

1.4 Descrições dos capítulos

No Capítulo 2, abordamos conceitos básicos de Biologia Molecular, em particular, ncRNAs. Inicialmente, apresentamos conceitos relativos ao Dogma Central da Biologia Molecular. Em seguida, realizamos uma breve revisão de literatura, contendo os principais métodos de predição de ncRNAs, incluindo métodos tradicionais, e de AM, além de descrevermos os bancos de dados de ncRNAs utilizados neste trabalho.

No Capítulo 3, discutimos conceitos gerais de Aprendizado de Máquina (AM) e, especificamente, os métodos utilizados nesta dissertação - SVM, ANN e RF, além das ferramentas computacionais para implementação dos algoritmos de AM.

No Capítulo 4, primeiro, propomos o método para produzir conjuntos de dados de snoRNAs artificiais buscando simular evolução. Em seguida, descrevemos as etapas para avaliar os métodos de AM na predição de snoRNAs na pesquisa de homologia.

No Capítulo 5, são discutidos os resultados de experimentos com C/D *box* e H/ACA *box* snoRNAs.

Por fim, no Capítulo 6, concluimos esta dissertação, destacando as contribuições e sugerindo trabalhos futuros.

Capítulo 2

RNAs não-codificadores

Neste capítulo, são apresentados conhecimentos básicos da Biologia Molecular, em particular ncRNAs e os conceitos necessários para o entendimento deste trabalho. Na Seção 2.1, é descrito o Dogma Central da Biologia Molecular (síntese de proteínas), baseado em ácidos nucleicos e proteínas. Por fim, na Seção 2.2, são apresentados diferentes tipos de ncRNAs, conceitos de mutações biológicas, além de métodos de predição e bancos de dados de ncRNAs.

2.1 Dogma Central da Biologia Molecular e Bioinformática

Em 1958, Francis Crick definiu o processo de transmissão de informação genética como Dogma Central da Biologia Molecular [31]. Segundo esse Dogma, o fluxo da informação genética ocorre por meio de três processos: replicação, onde a molécula de DNA é duplicada; transcrição, onde uma porção da fita molde de DNA traz informações que permitem gerar uma fita simples de RNA; e tradução, onde o RNA gerado na transcrição é utilizado como molde para a síntese de uma proteína.

Desde então, os pesquisadores, a partir de novas descobertas, propuseram mudanças no modelo original do Dogma proposto por Crick. Por exemplo, hoje se sabe que existem enzimas denominadas de RNA-polimerases, capazes de utilizar o RNA para produzir DNA e ainda que não são todas as moléculas de RNAs que codificam proteínas. Esses RNAs são chamados de não-codificadores de proteínas (em inglês, *non-coding RNAs* - ncRNAs).

2.1.1 Ácidos nucleicos

Os ácidos nucleicos são macromoléculas, formadas por unidades menores conhecidas como nucleotídeos. Cada nucleotídeo, por sua vez, é formado por três partes: um açúcar com

cinco átomos de carbono (pentose), ligado a um grupo fosfato e uma base orgânica nitrogenada [32]. Existem dois tipos de ácidos nucléicos, o DNA (ácido desoxirribonucleico) e o RNA (ácido ribonucleico). Essas macromoléculas são fundamentais para muitos processos importantes nas células, como na sintetização das proteínas. As pentoses do DNA e do RNA podem se ligar a quatro tipos diferentes de bases nitrogenadas que são: adenina (A), citosina (C), guanina (G) e timina (T), caso seja DNA, ou A, C, G e uracila (U) no lugar da timina, caso se trate de um RNA.

Além disso, as bases nitrogenadas são classificadas em dois grupos: purinas e pirimidinas. As purinas, adenina (A) e guanina (G) são maiores, enquanto as pirimidinas, citosina (C), timina (T) e uracila (U), são menores. Ambas são combinadas da seguinte forma - adenina e timina (ou uracila) e citosina e guanina, duas purinas e duas pirimidinas, que se ligam por meio de pontes de hidrogênio, para formar o DNA (ou RNA).

O DNA e o RNA possuem diferenças tanto posicionais quanto estruturais. Como mostra a Figura 2.1, o açúcar do DNA é a desoxirribose, enquanto o RNA contém ribose, idêntica à desoxirribose exceto pela presença de um grupo OH (hidroxila) extra no carbono 2'. Além disso o RNA contém a pirimidina uracila e o DNA a timina, as outras três bases (adenina (A), guanina (G) e citosina (C)) que ocorrem no DNA e RNA são idênticas.

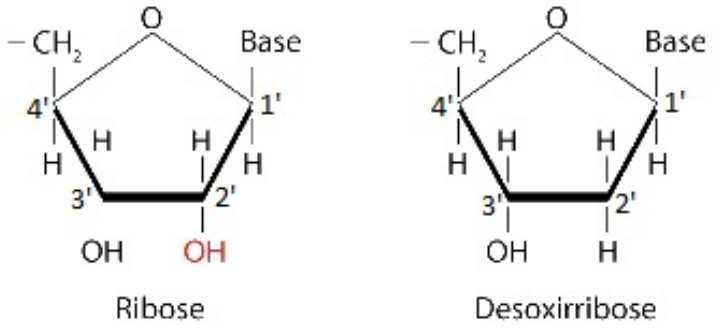

Figura 2.1: Estrutura dos nucleotídeos de DNA e RNA. Os nucleotídeos do RNA possuem o açúcar ribose, enquanto os nucleotídeos do DNA possuem a desoxirribose, que apresenta um grupo hidroxila no carbono 2' (adaptado de [1]).

Existem duas extremidades em uma cadeia de ácido nucleico: extremidade 5' e extremidade 3'. As ligações entre dois nucleotídeos é possível entre os carbonos 5' (grupo fosfato) e o carbono 3' (grupo hidroxila). Essas ligações significam que uma cadeia de DNA possui direção 5' $\longrightarrow$ 3'. O DNA é formado por uma dupla fita, disposta espacialmente em formato helicoidal (dupla hélice): uma fita com direção 5' $\longrightarrow$ 3' (fita codificadora) liga-se a uma fita na direção 3' $\longrightarrow$ 5' (fita molde). Enquanto o RNA é formado usualmente apenas por uma fita, como ilustra a Figura 2.2. Notamos que o RNA pode formar dupla fita, em certos processos celulares.

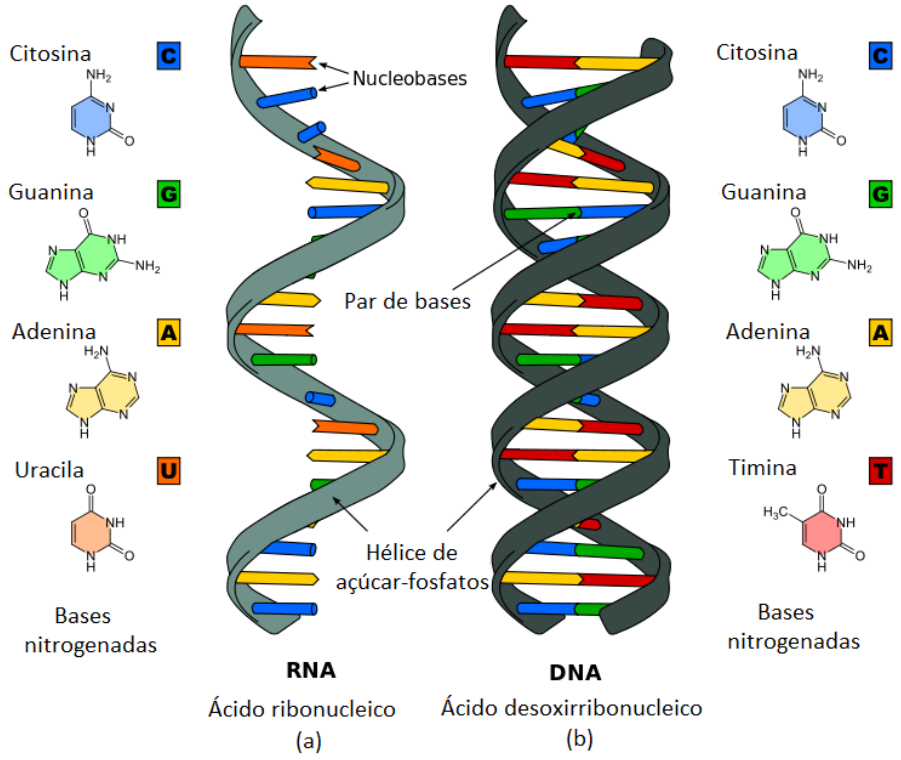

Figura 2.2: Estrutura espacial das moléculas de (a) RNA (fita simples) e (b) DNA (fita dupla), com suas bases nitrogenadas [2].

O DNA possui como função principal o armazenamento de informações, por exemplo, instruções fundamentais para a síntese de proteínas ou molécula de RNA. As informações que codificam transcritos estão contidas em regiões do DNA denominadas genes. As regiões do gene que codificam proteínas são chamadas de éxons, onde são intercaladas por regiões não codificantes, chamadas de introns, que são inicialmente transcritas em RNA [33].

2.1.2 Proteínas

Proteínas são polímeros, ou seja, moléculas que possuem várias cópias ligadas de um componente menor, sendo esses componentes chamados de aminoácidos [4]. Essas macromoléculas atuam em inúmeras funções no organismo dos seres vivos, sendo que a maioria das funções celulares necessitam de proteínas para realizá-las. Por exemplo, as proteínas participam de funções no sistema imunológico, podem atuar como enzimas catalisando reações químicas, podem transportar pequenas moléculas e participam na regulação gênica, dentre outras funções essenciais de organismos vivos [1].

Um aminoácido é constituído por um carbono central, chamado carbono alfa (C_α), que está ligado a quatro outros componentes, três em todos os aminoácidos - um grupo

amina (NH2), um grupo carboxila (COOH) e um átomo de hidrogênio (H), sendo o quarto chamado de cadeia lateral e simbolizado geralmente por R (Figura 2.3).

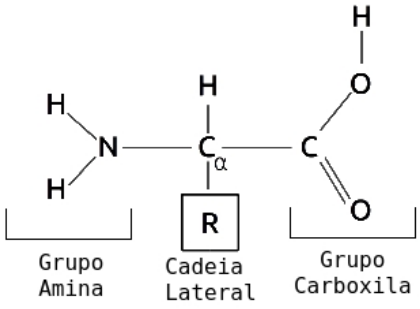

Figura 2.3: Características da estrutura química de um aminoácido (adaptado de [3]).

As propriedades da cadeia lateral (R) determinam as características específicas de um aminoácido, isto é, a cadeia lateral distingue um aminoácido de outro. Por exemplo, no caso do aminoácido mais simples, a glicina, a cadeia lateral é um átomo de hidrogênio (H), ou no caso de um aminoácido mais complexo, o triptofano, a cadeia lateral é composta por dois anéis de carbono [3, 4].

As proteínas são compostas por ligações entre os aminoácidos, denominadas ligações peptídicas. Uma ligação peptídica forma-se por uma reação de condensação, com a eliminação de uma molécula de água (H2O), em que ligações consecutivas do mesmo tipo podem gerar uma cadeia polipeptídica linear. Como a formação de cada ligação peptídica contém a eliminação de uma molécula de água, os componentes da cadeia são conhecidos como resíduos de aminoácidos.

Logo, uma proteína é uma cadeia de polipeptídeos constituída por resíduos de aminoácidos resultantes de ligações peptídicas. Existem 20 diferentes tipos de aminoácidos: alanina (A), cisteína (C), ácido aspártico (D), ácido glutâmico (E), fenilanina (F), glicina (G), histidina (H), isoleucina (I), lisina (K), leucina (L), metionina (M), asparagina (N), prolina (P), glutamina (Q), arginina (R), serina (S), treonina (T), valina (V), triptofano (W) e tirosina (Y).

2.1.3 Síntese de proteínas

Segundo o Dogma Central da Biologia Molecular, a síntese de proteínas (Figura 2.4) é iniciada por um processo de replicação, em que a molécula de DNA é duplicada. Na replicação do DNA, as fitas da dupla-hélice se dividem, ou seja, uma fita parental é dividida em duas moléculas filhas, então cada fita da dupla hélice atua como molde para a síntese de uma nova fita complementar.

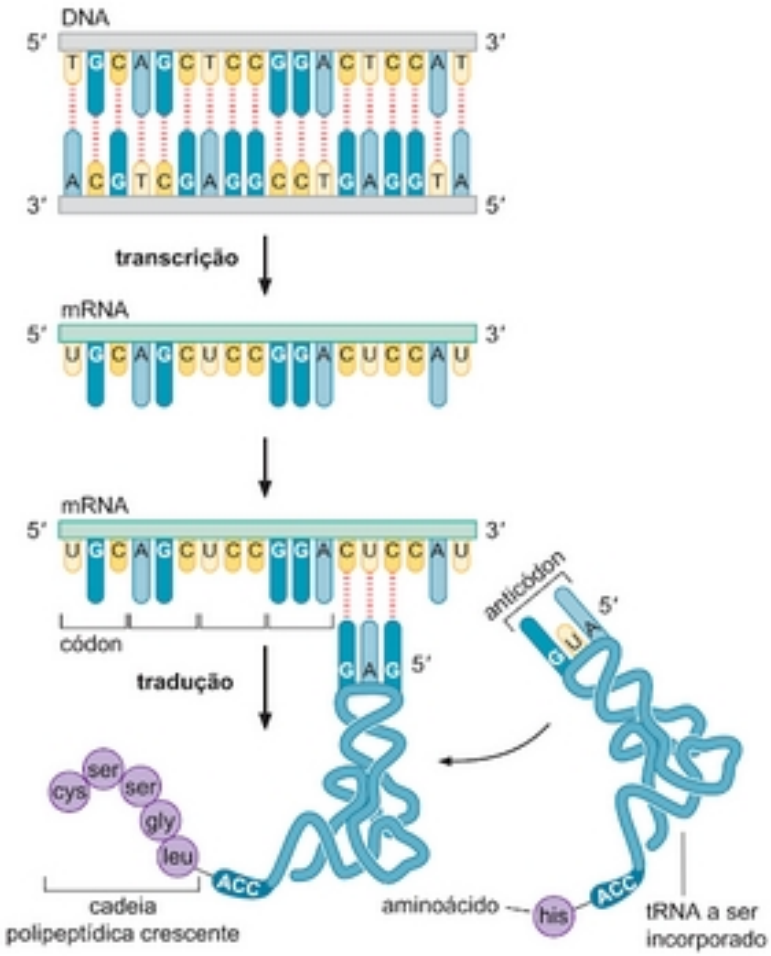

Figura 2.4: Síntese de proteínas: os nucleotídeos do pré-mRNA são unidos para formar uma cópia complementar da fita de DNA. Cada grupo de três nucleotídeos é um códon complementar a um grupo de três nucleotídeos na região do anticódon de uma molécula de tRNA. Quando ocorre o pareamente de bases, um aminoácido carregado pela outra extremidade da molécula de tRNA é adicionado à cadeia crescente de proteína [4].

Enzimas denominadas DNA polimerases são capazes de catalisar a síntese de novas fitas de DNA, percorrendo o DNA na direção 5' $\longrightarrow$ 3', assim as enzimas DNA polimerases catalisam a adição dos nucleotídeos na direção 5' $\longrightarrow$ 3'. Os mecanismos celulares não conseguem iniciar a síntese de uma proteína sem uma fita complementar. A enzima primase é responsável por produzir uma sequência curta de 15 a 30 nucleotídeos, chamado *primer* (iniciador), complementar ao segmento do DNA a ser copiado. Então, a DNA polimerase é capaz de adicionar o próximo nucleotídeo complementar na extremidade 3' do *primer*. A complementaridade entre as bases é usada, adenina com timina e vice-versa (A $\longleftrightarrow$ T), citosina com guanina e vice-versa (C$\longleftrightarrow$G).

Após a replicação, temos o processo de transcrição, que é o principal ponto de controle na expressão dos genes e na produção das proteínas. A RNA polimerase é a enzima responsável pela catalisação da transcrição do DNA, conduzindo a transcrição do DNA em um RNA mensageiro (mRNA) ou em um RNA transcrito. Ela realiza a polimerização na direção (5' $\longrightarrow$ 3'), a partir de uma fita molde de DNA, quando as fitas originais de DNA servem como moldes para a síntese das sequências complementares de RNA.

A RNA polimerase identifica na fita molde uma região promotora e, para cada nucleotídeo no molde, ela adiciona um nucleotídeo de RNA correspondente (complementar) à extremidade 3' da fita do RNA. Com o pareamento de bases complementares, adenina com uracila (A $\longrightarrow$ U), timina com adenina (T $\longrightarrow$ A), citosina com guanina e vice-versa (C $\longleftrightarrow$ G).

Em organismos procariotos, após o processo de transcrição, o mRNA produzido está pronto para concluir a síntese de proteínas. Entretanto, em organismos eucariotos, são necessárias várias reações para produzir o RNA mensageiro maduro[1]. Assim, em células eucarióticas, a produção de um RNA mensageiro é iniciada com todo o gene, incluindo introns e éxons, que é primeiramente transcrito em uma longa molécula de RNA, denominado de transcrito primário ou pré-mRNA.

Antes que o RNA deixe o núcleo, todas as sequências correspondentes aos introns são retiradas e os éxons são unidos entre si, por um processo denominado *splicing*. O resultado é uma molécula mais curta de RNA, que agora contém uma sequência codificadora. Quando o *splicing* do pré-mRNA, é completado, temos uma molécula de RNAm maduro, funcional, que pode deixar o núcleo e ser traduzido em proteínas [4].

Existe também o processo chamado de *splicing alternativo*, onde diferentes proteínas podem ser produzidas a partir do mesmo gene, denominadas isoformas proteicas. Essas modificações que ocorrem no RNA são chamadas co-transcricionais, pois ocorrem imediatamente e simultamente à transcrição do pré-mRNA.

Por fim, após a transcrição, ocorre a síntese de proteínas que é a tradução da informação contida no gene. No processo de tradução, o mRNA maduro, formado pela transcrição, é movido do núcleo para os ribossomos dentro do citoplasma, sendo transportado pelo RNA transportador (tRNA) ao RNA ribossomal (rRNA) e sintetizado em uma proteína, como descrito em seguida.

Na sequência de mRNA maduro, a informação é lida, isto é, cada aminoácido é codificado por sequências de 3 nucleotídeos chamadas de códons. Existe um único códon para iniciar todas as traduções, o *start codon* (AUG) e 3 códons para terminar, os *stop codons* (UAA, UAG e UGA).

[1] o RNA que será utilizado na transcrição

Em cada tRNA, existe a região que se ligará o aminoácido a ser transportado e a região que corresponde a uma tripla de bases complementares (chamado anticódon). Dessa forma, o tRNA transporta um aminoácido cuja sequência de bases (anticódon) será ligado a outra tripla correspondente ao códon do mRNA maduro.

2.2 NcRNAs

Avanços tecnológicos em décadas posteriores permitiram a proposição de alterações no Dogma Central da Biologia Molecular original de Crick, basicamente reavaliando papéis desempenhados pelos RNAs.

Pesquisas revelaram funções essenciais dos RNAs, além da participação na síntese de proteínas, sendo esses novos chamados de ncRNAs. Os ncRNAs participam dos mais diversos processos biológicos, por exemplo, possuem funções estruturais, catalíticas ou regulatórias nos mecanismos celulares [34, 35]. Ainda, foi identificada uma presença abundante de ncRNAs em diversos organismos, principalmente nos eucariotos [36].

As moléculas de ncRNAs podem ser classificadas por estruturas espaciais da seguinte forma:

- **Estrutura primária** é a sequência de bases nitrogenadas (A,T ou U, C e G).
- **Estrutura secundária** é representada em forma espacial 2D, corresponde às ligações feitas entre os pares de bases nitrogenadas. Resultam em uma estrutura local, por exemplo em forma de hélice.
- **Estrutura terciária** é representada em forma espacial 3D.

A diversidade funcional de cada classe de ncRNA ainda não é totalmente conhecida, principalmente porque essas moléculas possuem alta conservação de estrutura secundária (estrutura espacial) e não de estrutura primária (sequência de nucleotídes). Essa característica impossibilita o uso de métodos já conhecidos de identificação de proteínas [37].

Portanto, sob o ponto de vista computacional, as características dos ncRNAs tornam a modelagem *in silico* uma tarefa complexa e desafiadora, sendo necessário o uso de abordagens que considerem a análise da estrutura espacial dessas moléculas.

2.2.1 Classificação de ncRNAs

Atualmente existem diversas famílias de ncRNAs conhecidas [38], sendo suas funções relacionadas à sua estrutura espacial (terciária). Porém, como a estrutura terciária é extremamente complexa de ser prevista por métodos computacionais, atualmente, os pesquisadores trabalham com estrutura bidimensional (estrutura secundária), uma aproxi-

mação realizada para o estudo de funções em ncRNAs. Algumas das principais famílias de ncRNAs pequenos e longos, com suas funções, são descritas na Tabela 2.1.

Tabela 2.1: Classificação e funcionalidades de famílias importantes de ncRNAs

Sigla	**Nome**	**Função**
tRNA	RNA transportador	Transporte de aminoácidos na tradução de mRNAs
rRNA	RNA ribossomal	Catalisador no processo de tradução
snRNA	*Small nuclear RNA*	Remoção dos introns no processo de *splicing*
snoRNA	*Small nucleolar RNA*	Modificações químicas nos rRNAs, tRNAs e snRNAs
miRNA	*MicroRNA*	Família de genes com funções regulatórias na tradução
siRNA	*Small interfering RNA*	Moléculas ativas em *RNA interference*
piRNA	*Piwi-interacting RNA*	Regulação de tradução e estabilidade de mRNA
snmRNA	*Small non-messenger RNA*	Pequenos ncRNAs com função regulatória
stRNA	*Small temporal RNA*	Interrupção da tradução de mRNA
rasiRNA	*Repeat-associated siRNA*	Silenciamento da transcrição de genes com remodelagem de cromatina
lncRNA	ncRNAs longos	Regulação da expressão gênica a nível de remodelagem de cromatina

Como o foco desta dissertação será em um tipo expecífico de ncRNAs pequenos, o snoRNA, faremos uma descrição mais detalhada apenas dele.

***Small nucleolar* RNAs (snoRNAs)**

O objeto de estudo neste trabalho é uma classe específica de ncRNAs pequenos, denominada de *small nucleolar* RNAs (snoRNAs) [39], em que o comprimento varia de 60 a 300 nucleotídeos [40]. Os snoRNAs originam-se dos introns de um mRNA. Eles formam uma classe abundante, que são pequenas moléculas que exercem modificações químicas no rRNA e em outros ncRNAs, como o tRNA. Essas modificações possuem como finalidade a maturação desses ncRNAs.

Sabe-se que os snoRNAs possuem uma ampla variedade de funções celulares, além da modificação química de RNAs, que incluem manutenção de telômeros, processamento de pré-rRNA e atividades regulatórias em *splicing alternativo* [17].

Os snoRNAs estão divididos em duas classes principais - C/D *box* snoRNAs e H/ACA *box* snoRNAs, diferenciados por estruturas secundárias típicas de RNAs, além de características das sequências dos *boxes*. C/D *box* snoRNAs estão envolvidos principalmente

na metilação de rRNAs [39], enquanto H/ACA *box* snoRNAs estão envolvidos na pseudouridilação de snRNAs [41].

Os C/D *box* snoRNAs são formados por dois *boxes* conservados: C, que representa a sequência RUGAUGA, onde R é uma purina; e D, que representa a sequência CUGA. Ambos os *boxes* são encontrados perto das extremidades 5' e 3' do snoRNA, sendo separados por uma haste curta (3-10 nucleotídeos). Os H/ACA *box* snoRNAs são formados por uma estrutura contendo o *box* H (que representa a sequência ANANNA), localizado entre dois *hairpins*, e o *box* ACA seguido por 3 nucleotídeos localizados na extremidade 3' da sequência. Geralmente, os snoRNAs C/D *box* (Figura 2.5) possuem comprimento de 70 a 120 nucleotídeos, enquanto os snoRNAs H/ACA *box* (Figura 2.6) apresentam de 100 a 200 nucleotídeos [42, 19].

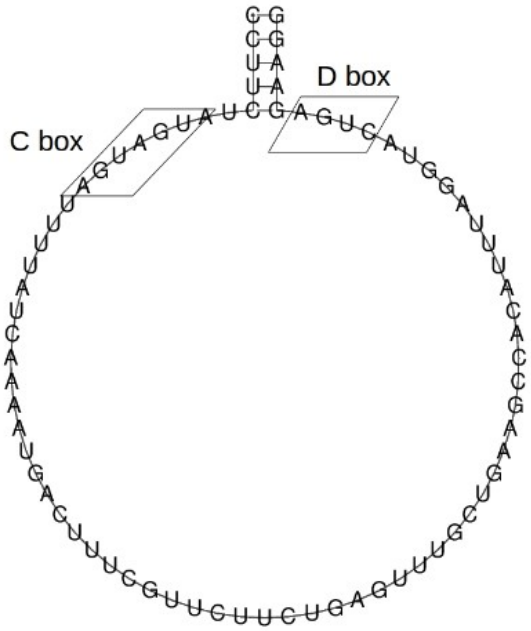

Figura 2.5: Estrutura secundária esquemática de um C/D *box* snoRNA [3].

2.2.2 Mutações

Nesta seção, apresentamos conceitos sobre mutações, sob um ponto de vista biológico, pois utilizamos esses conceitos para gerar uma árvore de sequências mutadas, de forma controlada, que simulam evolução. Essa árvore é essencial para podermos avaliar o desempenho de métodos de AM, tendo em vista que não temos disponíveis hoje um grande número de snoRNAs que possam ser utilizados para esse fim.

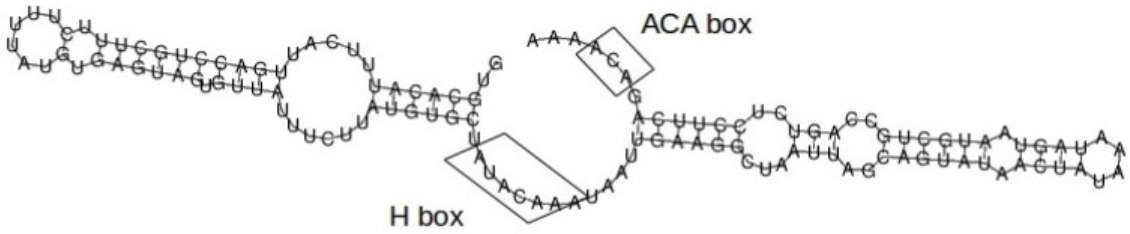

Figura 2.6: Estrutura secundária esquemática de um H/ACA *box* snoRNA [3].

O material genético[2] constitui a principal fonte de variação genética existente, assegurando que os indivíduos de uma mesma espécie apresentem DNAs diferentes. Alterações no material genético são conhecidas como mutações, garantindo a variabilidade genética.

As mutações podem ocorrer de diferentes formas, por exemplo, decorrentes de erros no processo de replicação do DNA ou de alterações por diversas causas nas moléculas que constituem o DNA, induzidas por substâncias produzidas pelo metabolismo celular. Essas alterações podem ocorrer também por meio da exposição das células a agentes que modificam o DNA, os quais são denominados de agentes mutagênicos [43].

A maioria das mutações genéticas que levam à evolução ocorrem de maneira aleatória e podem alterar as funções de uma molécula, com consequências neutras (não afetam o indivíduo), prejudiciais (podem levar à morte do indivíduo) ou ainda benéficas (que levam a um melhor desempenho do organismo na natureza) [44, 45, 46].

Em organismos multicelulares, as mutações podem ser classificadas em: mutações somáticas que ocorrem em células responsáveis pela formação de tecidos e órgãos, as quais, em animais, não são transmitidas aos descendentes; e mutações germinativas que ocorrem em células que originam gametas e podem ser transmitidas aos descendentes, além disso, podem ser transmitidas aos descendentes [47, 48].

Em nível molecular, existem dois tipos principais de mutações. Aquelas em que as alterações ocorrem em 1 ou mais bases dos genes, em nível de DNA, chamadas de gênicas. Temos ainda mutações que ocorrem por alterações em partes grandes de cromossomos, chamadas de cromossômicas [49]. Exemplos dessa últimas são reversões, que revertem porções grandes de cromossomos, e transposições, em que uma porção do cromossomo é cortada de uma localização inicial e reinserida em outra porção desse mesmo cromossomo.

Nesta dissertação, utilizamos mutação gênica, conforme descritos em seguida [50]:

[2]Material genético refere-se aos ácidos nucleicos que codificam os genes.

- Substituição: são mutações que ocorrem quando uma base nitrogenada é alterada para uma outra no DNA;
- Inserção: são mutações que ocorrem quando uma base nitrogenada é adicionada ao DNA;
- Remoção: são mutações que ocorrem quando uma base nitrogenada é removida do DNA.

Nos ncRNAs, a compreensão das variações genômicas foi fortemente motivada por constantes aprimoramentos em etapas de anotação de projetos de sequenciamento de genomas, assim como melhorias em validações funcionais de muitos ncRNAs, em laboratórios [51]. Em moléculas de snoRNAs, as mutações podem potencialmente alterar a estrutura secundária conservada, que é essencial para a funcionalidade esperada desses ncRNAs [52].

Por outro lado, diferentes métodos computacionais usam estruturas conhecidas como matrizes de substituição, que buscam refletir, da forma mais fidedigna possível, as probabilidades de ocorrência de mutações, em um período de evolução [53] . Essas matrizes contêm valores proporcionais à probabilidade que o aminoácido i sofra mutação para o aminoácido j, para todos os pares de aminoácidos. Assim, as matrizes são utilizadas como parâmetros em algoritmos de alinhamento como o BLAST [54], por exemplo.

Essas matrizes são construídas de forma cuidadosa, a partir de análises de uma grande quantidade de alinhamentos múltiplos de sequências de diferentes organismos.

Os dois tipos mais utilizados de matrizes de substituição são *Point Accepted Mutation (PAM)* e *Blocks of Amino Acid Substitution Matrix (BLOSUM)*. Na matriz BLOSUM, as frequências são obtidas diretamente das relações representadas nos blocos dos alinhamentos múltiplos, independentemente da distância evolutiva. Entretando, nas matrizes PAM, são contabilizadas mutações pontuais. Por isso foram utilizadas neste trabalho. As PAM são matrizes de substituição de aminoácidos, ou seja, codificam a mudança evolutiva esperada em nível de aminoácidos. Cada matriz PAM é projetada para comparar duas sequências separadas por um número específico de unidades PAM, sendo esse modelo mais adequado para evidenciar relacionamentos evolutivos, considerando mutações pontuais [55]. Por fim, deve-se notar que probabilidades de substituição de códons (que codificarão os aminoácidos) são usadas em muitos tipos de estudos de evolução molecular, por exemplo, na criação de sequências de DNA ancestrais [56].

2.2.3 Métodos para predição de ncRNAs

Métodos computacionais podem contribuir muito para caracterização dos ncRNAs, principalmente frente à geração crescente de dados com as tecnologias modernas de sequenci-

amento [57]. Ferramentas de detecção, análise, visualização e integração de dados passam a ser necessários e fundamentais para buscas e manuseio no enorme volume de dados hoje disponíveis [37].

As funções de ncRNAs são muito dependentes de suas estruturas secundárias, o que torna a descoberta de suas funções muito diferente dos RNAs codificadores de proteínas. Isso motivou o desenvolvimento de métodos específicos para pesquisa de ncRNAs [36]. Em seguida, são descritas algumas ferramentas para identificar e classificar ncRNAs, apresentados de acordo com os princípios seguidos pela proposta dos métodos.

Métodos baseados em homologia

- **BLAST**

A ferramente de alinhamento local *Basic Local Alignment Search Tool* (BLAST) [54] realiza buscas comparando sequências biológicas primárias contra um banco de dados que contém uma imensa quantidade de informação. A saída do Blast indica as sequências mais similares e com maiores significâncias estatísticas. É uma ferramenta importante para inferir homologia de sequências através da similaridade. O BLAST é muito utilizado por ser uma ferramenta muito rápida e existirem diversas variações, cada uma com um objetivo específico [58]:

- blastp: para comparação de sequências de aminoácidos com um banco de dados de proteínas;
- blastn: para comparação de sequências de nucleotídeos com um banco de dados de nucleotídeos;
- blastx: para comparação de sequências de nucleotídeos traduzidos em todas as *Open Reading Frame* (ORFs), com um banco de dados de proteínas;
- tblastn: para comparação de sequências de proteínas com um banco de dados de sequências de nucleotídeos traduzidos em todas as suas ORFs;
- tblastx: para comparar as ORFs de sequências de nucleotídeos com todas as ORFs de um banco de dados de nucleotídeos.

- **Infernal**

A ferramenta *INFErence of RNA ALignment* [59] é baseada em homologia de RNAs. Constrói modelos probabilísticos das sequências e das estruturas secundárias de RNA, chamados modelos de covariância (CMs), e os utiliza para pesquisar bases de dados de sequências de ácidos nucléicos para RNAs homólogos, ou para criar novos alinhamentos múltiplos de sequências baseados tanto em sequência quanto em estrutura secundária.

- **snoStrip**

A ferramenta *snoStrip* [28] é constituída de um *pipeline* de anotação automática desenvolvido para genômica comparativa de snoRNAs em genomas de fungos. Possui um método baseado em conservação de sequências e estrutura secundária, para predizer regiões de alvos putativos.

O *pipeline* abrange cinco passos: (i) um procedimento de busca baseado em homologia para acumular potenciais candidatos a snoRNA; (ii) um pós-filtro que usa a conservação de motivos de *box* e regiões de alvos putativos para aumentar a especificidade; (iii) um módulo para extrair recursos, incluindo estrutura secundária e previsões de alvos putativos; (iv) um módulo que produz um alinhamento múltiplo de todos os snoRNAs com relação à sua respectiva família; e (v) uma verificação de validação opcional.

Cada novo candidato snoRNA e suas informações derivadas de snoRNA correspondentes são subsequentemente armazenadas em um banco de dados interno. O *pipeline* do snoStrip pode ser executado com uma ou várias famílias de consultas, cada uma das quais pode conter uma ou mais sequências de consulta.

Métodos baseados em termodinâmica

- **Vienna RNA Package**

O Vienna [60] é um pacote de programas que prediz e compara estruturas secundárias de RNAs, com métodos baseados em termodinâmica para predição ou comparação de estruturas secundárias de sequências.

Alguns exemplos de programas que integram o Vienna são: o *RNAfold*, que realiza predição de estrutura secundária, dobrando em duas dimensões uma sequência de RNA, computando a estrutura espacial que apresenta uma *Minimum Free Energy* (MFE); o *RNAz*, que faz predições *de novo* de ncRNAs estruturados em alinhamento múltiplo de sequências, considerando suas estruturas secundárias; e o *RNAAlifold* calcula a MFE de múltiplas sequências de RNA, sendo a entrada um alinhamento múltiplo das sequências.

Métodos de identificação de ncRNAs

- **snoSeeker**

A ferramenta snoSeeker [61] identifica snoRNAs, incluindo os programas *CDseeker* e *ACAseeker*, que detectam C/D *box* snoRNAs e H/ACA *box* snoRNAs, respectivamente.

Possui filtragem altamente eficiente e específica de genes que modificam rRNAs e snRNAs, chamados de snoRNAs guias e snoRNAs órfãos, que não possuem complementaridade com rRNAs e snRNAs, em genomas de mamíferos.

- **snoReport 2.0**

A ferramenta snoReport 2.0 [3] identifica snoRNAs, utilizando predição de estrutura secundária de ncRNA, combinada com aprendizado de máquina, para identificar as duas principais classes de snoRNAs: os H/ACA *box* e os C/D *box*.

O snoReport 2.0 estende o método original snoReport [19], extraindo novos recursos para ambos os snoRNAs C/D *box* e H/ACA *box*, além de aplicar uma técnica mais sofisticada na fase de treinamento da *Support Vector Machine* (SVM) com dados de organismos vertebrados e uma escolha cuidadosa dos parâmetros custo C e gama γ da SVM.

- **MuStARD**

A ferramenta MuStARD [18] identifica miRNAs e snoRNAs. É um método genérico que utiliza redes neurais convolucionais, que pode ser treinado em diferentes classes de RNAs pequenos de humanos. Além disso, pode fazer a identificação interespécie de elementos funcionais, prevendo pequenos RNAs de camundongo. Identifica lócus [3] genômicos de pequenos ncRNAs.

Métodos de classificação de ncRNAs

- **SVM-Portrait**

O SVM-Portrait [62] é adequado para identicar ncRNAs de transcritomas incompletos ou de espécies cujas caracterizações não foram concluídas. Essa ferramenta utiliza métodos baseados em técnica de AM, particularmente a *Support Vector Machine* (SVM). O resultado do SVM-Portrait é a probabilidade de uma sequência não codificar uma proteína.

A Tabela 2.2 resume as ferramentas de identificação e classificação de ncRNAs.

2.2.4 Banco de dados de ncRNAs

O aumento da quantidade de dados sobre classes diferentes de ncRNAs resulta em novos bancos de dados de ncRNAs, com o objetivo de organizar informações relevantes sobre os diversos tipos de ncRNAs existentes. Alguns bancos de dados de ncRNA são apresentados em seguida.

O **NONCODE** [63] é um banco de dados de ncRNAs extraídos automaticamente da literatura e do GenBank [51], os quais foram manualmente curados. O NONCODE possui quase todos os tipos de ncRNAs com exceção de tRNAs e de rRNAs e todas as suas

[3] uma posição fixa e específica em um cromossomo, onde está localizado determinado gene ou marcador genético.

Tabela 2.2: Métodos computacionais para predição de ncRNAs

Métodos	**Descrição**
BLAST	Realiza alinhamento local de sequências primárias
Infernal	Baseado em Gramática Estocástica Livres de Contextos e Modelo de Covariância
snoStrip	*Pipeline* para análise de sequências de snoRNAs em genomas de fungos
Vienna	Compara estruturas secundárias de RNAs
snoSeeker	Identifica C/D *box* e H/ACA *box* snoRNAs homólogos procurando em alinhamentos de sequências conservadas de snoRNAs
snoReport 2.0	Usa combinação entre predição de estrutura secundária e a técnica de AM SVM para identificar C/D *box* e H/ACA *box* snoRNAs
MuStARD	Utiliza ANN para identificar lócus genômicos de pequenos ncRNAs
SVM-Portrait	Identifica ncRNAs de transcritomas incompletos

sequências e informações relacionadas foram confirmadas manualmente. Mais de 80% de suas entradas são baseadas em dados experimentais.

O **RNAdb** [64] é um banco de dados de ncRNAs de mamíferos que contém sequências e anotações de milhares de ncRNAs, mas a maioria com papéis ainda não conhecidos.

O **RFAM** [38] é uma base de dados curada (revisada e supervisionada), que contém informações sobre milhares de famílias de ncRNAs. É categorizada por sequências primárias de ncRNA e estruturas secundárias, através do uso de alinhamento múltiplo, consenso de anotações de estruturas secundárias e modelos de covariância.

O **miRBase** [65] é um banco de dados de microRNAs de diferentes organismos.

O **snoRNA Database** [66] contém snoRNAs humanos dos dois tipos H/ACA *box* e C/D *box*.

O **snOPY** [67] é um banco de dados que provê informações de snoRNAs, localização genômica de snoRNAs e RNAs alvo de diversos organimsos. Contém também sequências de ortólogos de vários organismos.

A Tabela 2.3 resume os bancos de dados de ncRNAs.

Tabela 2.3: Bancos de dados de ncRNAs

Bancos	**Descrição**
NONCODE	contém ncRNAs extraídos automaticamente da literatura e do GenBank
RNAdb	contém ncRNAs de mamíferos
RFAM	contém informações sobre milhares de famílias de ncRNAs
miRBase	contém microRNAs de diferentes organismos
snoRNA Database	contém snoRNAs de humanos
snOPY	provê informações de snoRNAs

Capítulo 3

Aprendizado de Máquina

AM é uma subárea de estudo da Inteligência Artificial responsável pela definição de algoritmos que aprendam com informações já existentes. Então, os algoritmos automaticamente se aprimoram, com a experiência, possuindo a capacidade de aprender, e também de evoluir, à medida que são expostos a novos dados. Por exemplo, eles podem utilizar reconhecimento de padrões para a aprendizagem [68].

Nos últimos anos, técnicas de AM estão auxiliando em diversas atividades como mineração de dados, processamento de linguagem natural, logísticas de empresas, negociações financeiras e muitas outras. Em um ambiente em que a velocidade das mudanças e a necessidade de adequação a essas mudanças é crescente, a análise de grandes volumes de informação, de forma mais ágil, confiável e com adaptações constantes, pode representar um diferencial competitivo significativo [69].

Na Seção 3.1, são apresentados conceitos básicos sobre AM e seus métodos. Nas Seções 3.2, 3.3 e 3.4 são descritos os métodos de AM SVM, ANN e RF, respectivamente. Por fim, na Seção 3.5 são descritos ferramentas computacionais, que foram utilizadas para implementar os métodos de AM avaliados nesta dissertação.

3.1 Conceitos básicos

Segundo Mitchell [70], no AM, um programa de computador aprende a partir de uma experiência E, atendendo a alguma classe de tarefas T e a uma medida de *performance* P, onde sua *performance* para a tarefa T, medida em P, é aprimorada com a experiência E. A experiência E normalmente é usada em algoritmos de AM, como um conjunto de treinamento t = (X_1, X_2, ..., X_i, ..., X_m), com m amostras. Cada amostra X_i, $i = 1$, 2, .., m, é um vetor da forma $X_i = (x_1, x_2, ..., x_i, ..., x_n)$, tal que x_1, x_2, ..., x_i, ..., x_n são características (do inglês, *features*) que descrevem X_i. Nossa hipótese h é a de que

exista uma função de aprendizagem capaz de gerar, a partir de uma amostra X_i, uma saída $h(X_i)$, que reflete o objetivo da aprendizagem [3].

Portanto, através de um conjunto de treinamento um algoritmo de aprendizagem busca uma função f de aprendizado, tal que, essa função f corresponda a função de hipótese h ou que seja muito próxima de h. Algoritmos de AM podem ser classificados em quatro paradigmas principais de aprendizagem: supervisionada, não-supervisionada, semi-supervisionada e por reforço [70].

3.1.1 Aprendizagem supervisionada

Na aprendizagem supervisionada, os algoritmos de aprendizagem extraem um modelo de conhecimento a partir de exemplos (amostras), em que são fornecidas como entrada classificações corretas dessas amostras. O propósito da construção desse modelo é que o algoritmo tenha a capacidade de produzir saídas corretas (desejadas) para amostras não conhecidas. Assim, os algoritmos são capazes de aprender a partir de uma etapa de treinamento, que utilizará amostras cujas classificações são conhecidas.

O método supervisionado utiliza modelos preditivos, e busca encontrar uma função, modelo ou hipótese que possa ser utilizada para predição. Assim, existe um conjunto de treinamento, onde cada amostra possui uma característica ou classe já conhecida e, com essas amostras, é gerada uma função f igual ou próxima a função hipótese h (saída esperada), que será usada para descobrir as classes às quais uma nova amostra pertence [3]. Na Figura 3.1, é apresentado um exemplo de criação de um modelo de classificação, de forma simplificada, com base em aprendizado supervisionado. Nessa figura, temos um conjunto com n dados de entrada. Cada dado x_i, $i = 1, \ldots, n$, possui m características (*features*), $x_i = (x_{i1},..., x_{im})$. As variáveis y_i representam as classes de cada x_i, $i = 1, \ldots, n$. A partir dos exemplos (amostras) e suas respectivas classes corretas, o algoritmo de AM constroi um modelo (classificador) $f(x)$.

São exemplos de métodos computacionais deste paradigma:*Naive Bayes* [29], SVM [71], ANN [7], RF [72] e *Ensemble* [73].

3.1.2 Aprendizagem não-supervisionada

Na aprendizagem não-supervisionada, existe um conjunto de treinamento, porém seus exemplos (amostras) não possuem classe de saída conhecida. Assim, os algoritmos precisam descobrir padrões e categorias extraídos dos dados para determinar uma saída com as classes de dados, criadas pelo método. Os algoritmos de aprendizagem não-supervisionada são utilizados principalmente para encontrar padrões que auxiliem no entendimento dos

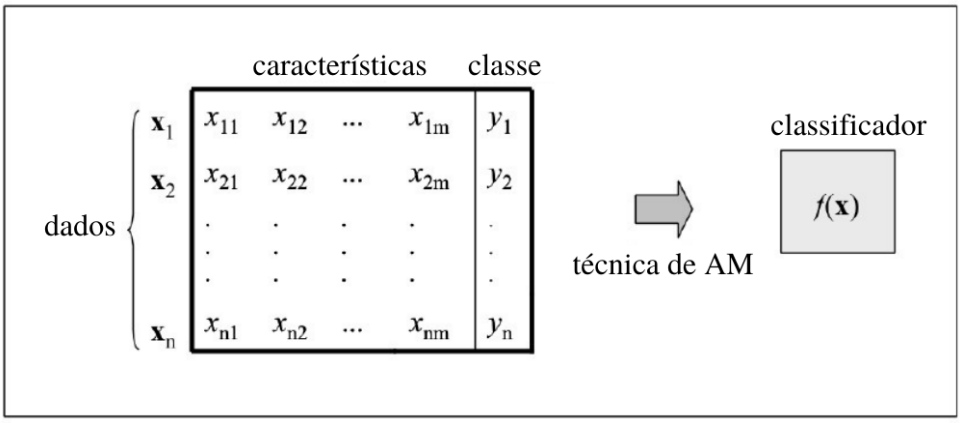

Figura 3.1: Exemplo de construção de modelo (classificador) com base no paradigma de aprendizado supervisionado (adaptado de [5]).

dados. Utilizam modelos descritivos, com o objetivo de encontrar algum padrão ou estrutura contida no conjunto de treinamento.

São exemplos de métodos computacionais deste paradigma: *Redes de Kohonem* ou *Self-Organizing Map (SOM)* [74] e *Cocktail Party* [75].

3.1.3 Aprendizagem semi-supervisionada

Na aprendizagem semi-supervisionada, os algoritmos de aprendizagem são capazes de aprender a partir de dados supervisionados e não-supervisionados. Nesse tipo de aprendizagem o treinamento utiliza exemplos (amostras) já classificados e amostras que não possuem classes conhecidas. Esse método geralmente possui uma grande quantidade de dados de treino mas apenas alguns são supervisionados. Logo, é possível usar esse tipo de aprendizado quando se deseja que o algoritmo aprenda informações nos dados mas também aprenda utilizando alguns dados supervisionados.

São exemplos de métodos computacionais deste paradigma: *COP–k-means* [76] e *SEEDED–k-means* [77].

3.1.4 Aprendizagem por reforço

Na aprendizagem por reforço, não existe um conjunto de treinamento, e o algoritmo descobre por tentativa e erro quais ações geram as maiores recompensas. Esses métodos baseiam-se em tomadas de decisões, a partir do ambiente no qual o algoritmo interage e das ações que o algoritmo pode tomar. Dessa forma, o objetivo do aprendizado por reforço é que o agente escolha ações que maximizem a recompensa esperada e minimizem

a função de custo, definida como a expectativa do custo cumulativo das ações tomadas em uma sequência de etapas, em vez de considerar simplesmente um único custo [70].

São exemplos de métodos computacionais deste paradigma: *Q-learning* [78], *Sarsa* [79] e *Dyna* [80].

As seções seguintes trazem detalhes dos métodos comumente usados para predição de ncRNAs [3, 18, 81, 82] a serem utilizados neste trabalho - SVM, ANN e RF, respectivamente.

3.2 Máquinas de Vetores de Suporte

O método denominado de Máquinas de Vetores de Suporte (do inglês, *Support Vector Machine* - SVM) pode ser utilizado tanto para classificação quanto para regressão. É uma técnica que possui um conjunto de métodos para analisar dados e reconhecer padrões. SVM tem como principal objetivo, no contexto de problemas de classificação de padrões, construir uma superfície de decisão, ou um hiperplano, de tal modo que a margem de separação entre as amostras de entrada positivas e negativas é maximizada. A Figura 3.2 mostra um exemplo de hiperplano de separação de amostras de entrada.

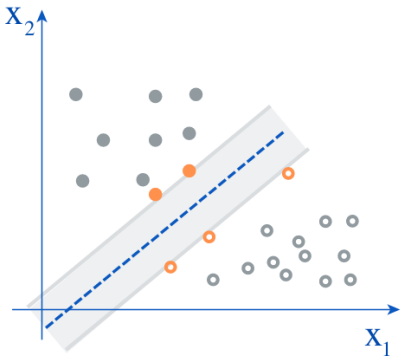

Figura 3.2: Exemplo de um hiperplano ótimo para padrões linearmente separáveis. As amostras de dados de cor laranja representam os vetores de suporte, separando dois conjuntos de amostras.

O hiperplano usado para separar as classes busca separar dois conjuntos de amostras, maximizando a distância das entradas mais próximas entre as classes. Essas amostras mais próximas caracterizam o hiperplano e são denominadas de vetores de suporte. Existem duas margens, demarcando o hiperplano, as quais consistem na distância entre as amostras mais próximas de cada uma das duas classes.

SVMs lineares são eficazes na classificação de conjuntos de amostras linearmente separáveis ou que possuam uma distribuição aproximadamente linear. Porém, há muitos casos

em que não é possível separar corretamente as amostras de entrada por um hiperplano de separação, como mostra a Figura 3.3.

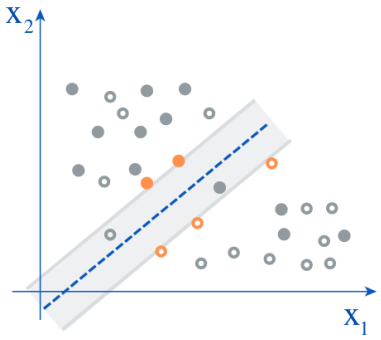

Figura 3.3: Exemplo de um hiperplano contendo amostras de entrada que não são linearmente separáveis.

Para solucionar o problema das amostras de entrada que não são linearmente separáveis, é necessário fazer um mapeamento não linear do vetor de entrada dentro de um espaço de características de dimensão maior [70]. Para isso, é realizada uma operação chamada de função *kernel*, que visa encontrar um hiperplano mais adequado para separar as classes, mapeando por exemplo, as amostras do plano R^2 para o R^3. Assim, o conjunto de amostras não linearmente separável em R^2 torna-se linearmente separável em R^3. A Figura 3.4 mostra uma função *kernel*.

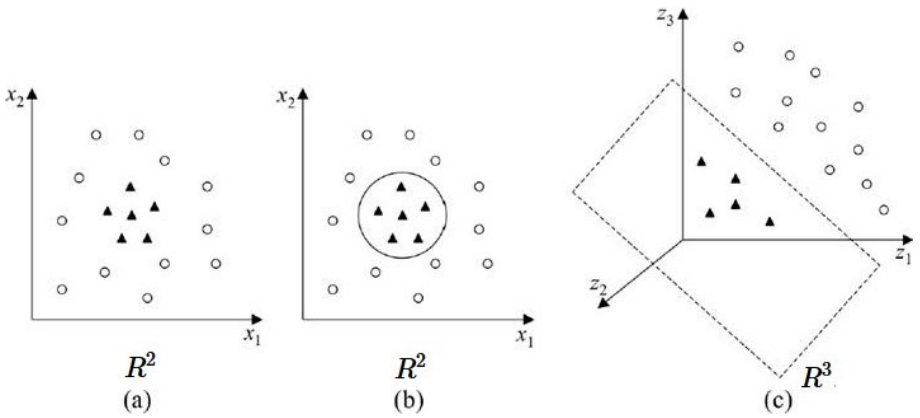

Figura 3.4: Exemplo de transformação do hiperplano em R^2 para um em R^3 usando uma função *kernel*. (a) Conjunto de dados não linearmente separável. (b) Fronteira não linear no espaço de entrada,em R^2. (c) Fronteira linear no espaço de características, em R^3 (adaptado de [5])

Função *kernel*

Como dito antes, a função *kernel* faz um mapeamento não linear das amostras do espaço de entrada para um novo espaço de características, onde os padrões são linearmente

separáveis com alta probabilidade, se forem satisfeitas duas condições: a transformação não é linear; e a dimensão do espaço de características é grande o suficiente [3]. A Figura 3.5 ilustra esse mapeamento.

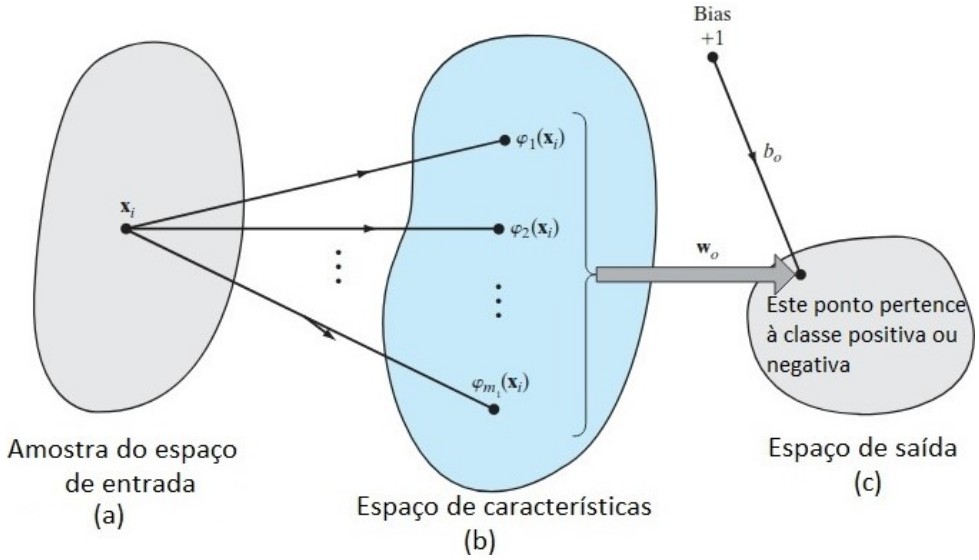

Figura 3.5: Exemplo de dois mapeamentos em uma SVM para classificação de padrões. (a) Mapeamento não linear do espaço de entrada para o espaço de características. (b) Mapeamento linear do espaço de características para o espaço de saída. (c) Espaço de saída (adaptado de [5]).

Alguns exemplos de função kernel usados na SVM são apresentados na Tabela 3.1 - Polinomial, *Radial basis function kernel (RBF)* e *Perceptron* de duas camadas [7].

Tabela 3.1: Definição de funções *kernel*.

Função *kernel*	**Definição**
Polinomial	$(x^T x_i + 1)^p$
RBF	$exp(-\frac{1}{2\sigma^2}\|\|x - x_i\|\|^2)$
Perceptron de duas camadas	$tanh(\beta_0 x^T x_i + \beta_1)$

A SVM padrão é um classificador linear binário, ou seja, toma como entrada um conjunto de dados e prediz, para cada entrada, a qual das duas possíveis classes a entrada pertence. Isso ocorre porque a SVM faz um treinamento que recebe como entrada amostras e cria um modelo que, ao ser testado, realiza a separação em duas classes [5].

Na etapa de treinamento, a SVM realiza uma busca em grade (do inglês, *grid*) para identificar bons valores para os meta-parâmetros custo (C) e gama (γ), para otimizar os

critérios de *performance*, por exemplo, acurácia. C é uma espécie de tolerância a erros existentes em uma classificação durante o treinamento. Os valores de C e γ podem ser ajustados para valores altos e baixos, dependendo do problema a ser resolvido [68].

3.3 Redes Neurais Artificiais

Redes Neurais Artificiais (do inglês, *Artificial Neural Networks - ANN*) representam um modelo matemático inspirado na estrutura neural do cérebro de organismos inteligentes, sobretudo de seres humanos. As ANN adquirem conhecimento através da experiência, possuindo capacidade de reconhecer padrões complexos, utilizando uma função de aprendizagem implícita na própria rede.

ANN foram projetadas para reproduzir o aprendizado, por meio de sistemas que aprendem a partir de exemplos, em uma etapa de treinamento.

Sua estrutura é composta por camadas - uma de entrada, uma de saída e pelo menos uma camada oculta, que transformam os dados de entrada em informações que serão utilizadas em classificadores e informações de saída. As camadas são compostas de neurônios [7]. As ANN podem classificar dados com base em padrões aprendidos. Devido ao seu bom desempenho, esse modelo matemático têm sido amplamente aplicado na identificação de ncRNAs. UM exemplo é a ferramenta MuStARD [18], já mencionada em capítulo anterior.

ANNs são modeladas basicamente a partir do sistema nervoso humano, que é formado por um conjunto extremamente complexo de células, os neurônios. O cérebro humano possui cerca de 10^{11} neurônios e mais de 10^{14} sinapses[1], possibilitando a formação de redes muito complexas. Neurônios biológicos têm papel essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio. Os neurônios possuem dendritos, que são um conjunto de terminais de entrada, um corpo central e axônios, que são longos terminais de saída. A Figura 3.6 mostra uma representação esquemática de um neurônio biológico com seus componentes básicos.

3.3.1 Modelo básico de ANN

As ANNs são caracterizadas por um modelo baseado em neurônios artificiais e na arquitetura de rede utilizada, a qual define a conectividade entre os neurônios artificiais e também pela estratégia adotada no ajuste dos pesos sinápticos, conhecida como algoritmo de treinamento ou aprendizado [7]. Portanto, uma ANN é composta por um conjunto de neurônios artificiais interconectados, através dos quais os dados de entrada percorrem um

[1]Sinapse é a região de contato entre dois neurônios através da qual os impulsos nervosos são transmitidos entre eles.

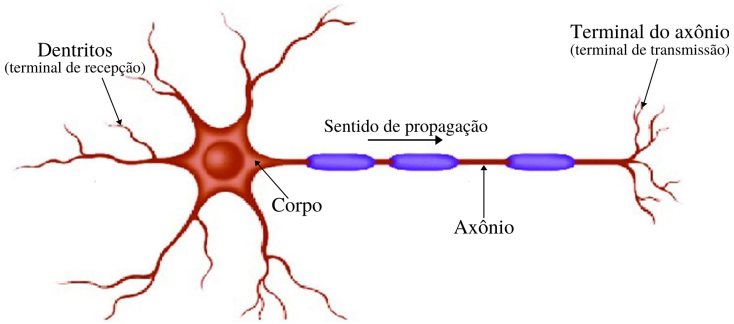

Figura 3.6: Representação esquemática de um neurônio biológico (adaptado de [6]).

caminho iniciando nas camadas de entrada, passando pelas camadas intermediárias (que podem ser ocultas) e finalizando na camada de saída [70].

A Figura 3.7 mostra o modelo computacional de um neurônio artificial.

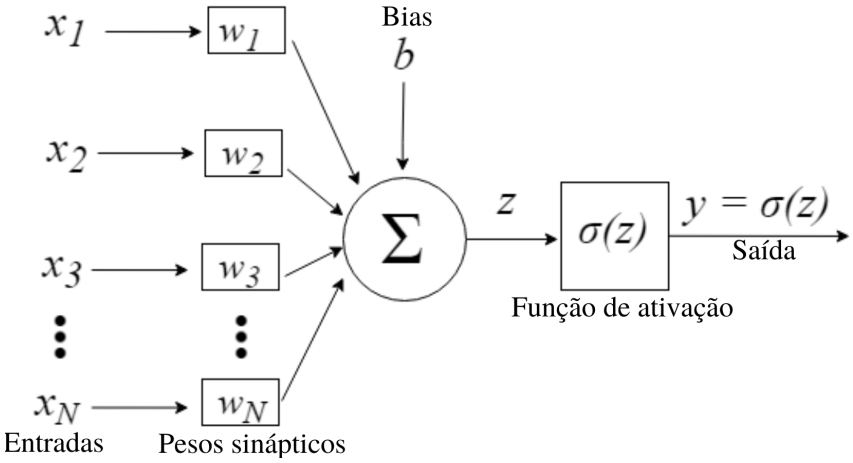

Figura 3.7: Modelo não-linear de um neurônio artificial (adaptado de [7]).

As entradas no neurônio artificial são representadas pelo vetor $x = [x_1, x_2, x_3, \ldots, x_N]$. Ao chegarem ao neurônio, são multiplicados pelos respectivos pesos sinápticos, que são os elementos do vetor $w = [w_1, w_2, w_3, \ldots, w_N]$, gerando o valor z. O valor z é comumente denominado potencial de ativação, sendo representado pela Equação 3.1, em que b é o *bias*.

$$z = \sum_{i=1}^{n} x_i w_i \tag{3.1}$$

Então, o valor z passa por uma função de ativação σ e produz a saída y, conforme a Equação 3.2 do neurônio.

$$y = \sigma(z + b), \quad com \quad y = \begin{cases} 1, \ se \quad z \geq 0 \\ 0, \ se \quad z < 0 \end{cases} \tag{3.2}$$

3.3.2 Funções de ativação

Um elemento extremamente importante das ANN é a função de ativação, denotada por σ, que define o valor da saída de um neurônio. Essa função decide se um neurônio deve ser ativado ou não, dependendo se a informação que o neurônio está recebendo é relevante para a classificação dos dados de entrada ou deve ser ignorada. Além disso, pode conter vários comportamentos, dependendo da aplicação e do modelo da rede neural utilizada. Em seguida, descrevemos alguns tipos básicos de funções de ativação [7].

Função sigmóide

A sigmóide é uma das funções de ativação mais comuns na construção de ANN. É definida como uma função estritamente crescente, que exibe balanceamento adequado entre comportamento linear e não-linear. Um exemplo de função sigmóide é a função logística, dada pela Equação 3.3, onde a é o parâmetro de inclinação da função sigmóide.

$$\sigma(z) = \frac{1}{1 + exp(-az)} \tag{3.3}$$

A variação do parâmetro a permite obter funções sigmóides de diferentes inclinações, conforme ilustrado na Figura 3.8. No limite, à medida que o parâmetro de inclinação se aproxima do infinito, a função sigmóide torna-se simplesmente uma função de limiar. A principal vantagem uma função sigmóide é a capacidade de assumir um intervalo contínuo de valores entre 0 a 1, além de ser uma função diferenciável.

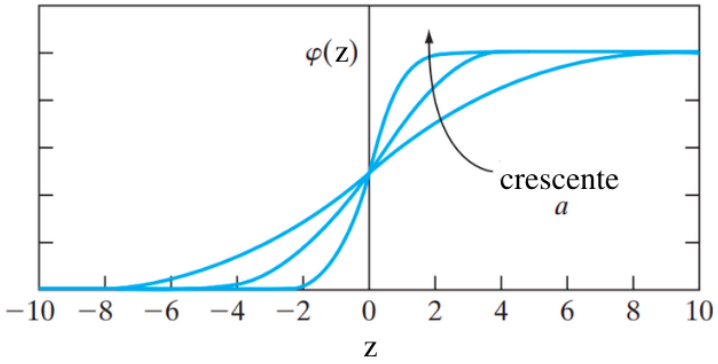

Figura 3.8: Gráfico da função sigmóide para parâmetro de inclinação variável a (adaptado de [7]).

Função tangente hiperbólica

A função de ativação sigmóide descrita acima assume valores de 0 a 1. Entretanto, algumas vezes, é desejável que a função de ativação se estenda de -1 a 1. Uma forma especial da função sigmóide corresponde a função de ativação tangente hiperbólica, definida pela Equação 3.4.

$$\sigma(z) = tanh(z) \tag{3.4}$$

Função ReLU

A Unidade Linear Retificada (do inglês *rectified linear unit* - ReLU) é uma função de ativação genérica, usada na maioria dos casos atualmente [83]. Essa função é inspirada nos neurônios que retornam o máximo entre um valor positivo especificado e 0, conforme definido na Equação 3.5.

$$\sigma(z) = max(0, z) \tag{3.5}$$

Uma ANN com a função de ativação ReLU (veja Figura 3.9) é computacionalmente simples, pois a propagação de uma entrada através de qualquer rede neural requer essencialmente apenas multiplicações de matrizes, que é um procedimento altamente otimizado e facilmente paralelizável.

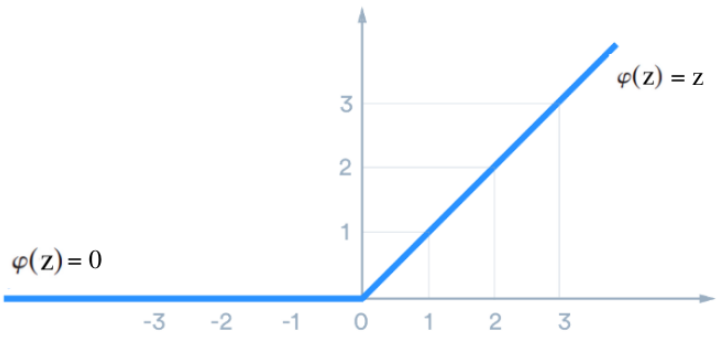

Figura 3.9: Gráfico da função de ativação ReLU (adaptado de [8]).

3.3.3 Classificações das ANNs

Dependendo da natureza da aplicação, o modelo de rede neural a ser empregado deverá seguir um dos dois paradigmas descritos a seguir [7]:

1. Em situações onde as características que determinam a que classe um dado pertence não sejam conhecidas *a priori*, ou seja, a ANN deve descobrir padrões de classificação

dos dados de entrada, deve-se usar modelos de **ANNs de aprendizagem não-supervisionada**. Exemplos deste paradigma é chamado de Mapa Auto-Organizável (em inglês, *Self Organizing Map* - SOM) [74] e *AutoWiSARD* [84].

2. Em situações onde as classes para os dados de entrada são bem conhecidos, ou seja, quando as classes dos dados de entrada são conhecidas, na fase de treinamento, deve-se usar modelos de **ANNs de aprendizagem supervisionada**. O treinamento desse tipo de rede neural consiste na apresentação de padrões cujas classes são conhecidas, até que a rede neural seja capaz de identificar essas classes de forma correta. Exemplos dessa categoria de ANNs são o *Perceptron* [7], *ADALINE* [85] e o modelo *WiSARD* [86].

3.4 Floresta Aleatória

Floresta Aleatória (do inglês, *Random Forest* - *RF*) é um algoritmo capaz de realizar tarefas de regressão e classificação. O algoritmo RF cria uma floresta aleatória contendo um conjunto de árvores de decisão, como mostra a Figura 3.10. As árvores de decisão, que formam a floresta aleatória, são construídas de forma aleatória na fase de treinamento, sendo que cada árvore será utilizada para escolher a classificação final [87].

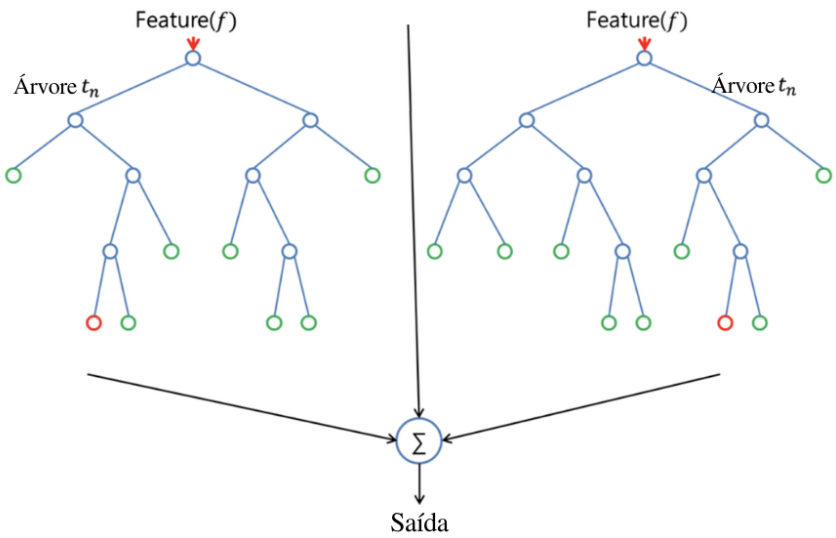

Figura 3.10: Exemplo de uma floresta aleatória com duas árvores de decisão, onde $Features(f)$ representa as características das amostras de entrada (adaptado de [9]).

Conceitos básicos de árvores de decisão

Uma árvore de decisão pode ser definida recursivamente da seguinte maneira:

- Toda árvore T possui um *nó* chamado raiz, que possui ligações para outros *nós*, denominados ramos ou filhos;
- Suponha que t seja um *nó* e $T_1, T_2, ..., T_n$ sejam árvores com *nós* raízes $t_1, t_2, ..., t_n$, respectivamente. Podemos construir uma nova árvore transformando t no pai dos *nós* $t_1, t_2, ..., t_n$. Logo na árvore T, t será o *nó* raíz e $T_1, T_2, ..., T_n$ serão as sub-árvores ou ramos da raiz. Os *nós* $t_1, t_2, ..., t_n$ são chamados filhos do *nó* t;
- Todos os *nós* que não possuem filhos são chamados *nós* terminais ou folhas. Os nós que têm filhos são chamados *nós* não terminais ou nós internos.

Uma árvore de decisão possui a seguinte estrutura:

- *Nós* internos (ou nós de decisão) são rotulados com amostras. O *nó* interno que contém o nome de uma amostra, para cada possível valor da amostra, corresponde a um ramo para uma outra árvore de decisão;
- *Nós* folhas (ou *nós* resposta) são rotuladas com classes. O *nó* folha contém o nome de uma classe ou o símbolo nulo, onde *nulo* indica que não é possível atribuir nenhuma classe ao nó por não haver nenhum exemplo que corresponda a esse *nó*;
- Ramos são rotulados com valores (amostras categóricos) ou com intervalos (amostras numéricos).

Na RF, cada árvore de decisão recebe um subconjunto de amostras aleatórias do conjunto de dados. O algoritmo RF utiliza *Bootstrap aggregating (Bagging)*[2] para o treinamento de cada árvore de decisão, como mostra a Figura 3.11.

Portanto, uma RF é um classificador que consiste em uma coleção de classificadores estruturados em árvores de decisão $h(x, \Theta_k)$, $k = 1, \ldots, k = n$, onde Θ_k são amostras aleatórias distribuídas de forma idêntica, e cada árvore fornece um voto unitário para a classe na entrada x. A partir das árvores de decisão, a RF escolhe a classificação que obteve mais votos (de todas as árvores de decisão da floresta aleatória), isto é, realiza a predição com base na maioria dos votos, ou, em caso de regressão, considera a média das saídas por árvores de decisão diferentes [87].

3.5 Ferramentas computacionais

Nesta seção, apresentamos duas ferramentas computacionais, utilizadas para os experimentos realizados nesta dissertação, *Keras* e *Scikit-learn*, do ambiente *Jupyter-notebook*

[2] *Bagging*, ou ensacamento, é um meta-algoritmo de aprendizado de máquina, projetado para melhorar a estabilidade e a precisão dos algoritmos de aprendizado de máquina usados na classificação e regressão estatística.

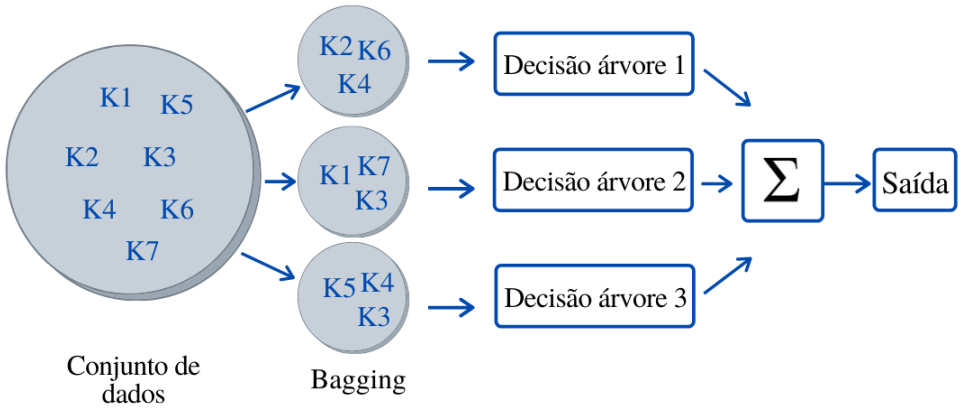

Figura 3.11: Exempo de uma RF com um conjunto de dados de treinamento $[k_1, K_2, \ldots, k_7]$, distribuído em três árvores de decisão.

O *Jupyter-notebook* [88] é um ambiente computacional web para criação de documentos na plataforma Jupyter, que permite unir código de linguagem de programação e texto. Utilizamos essa ferramenta para implementar os algoritmos de aprendizagem supervisionada avaliados nesta dissertação. A implementação foi realizada utilizando pacotes em linguagem *Python*, (*Keras* e *Scikit-learn*).

Keras

Keras [89] é uma API[3] de alto nível em linguagem *Python*, para ANNs. É capaz de executar as bibliotecas de tensores, por exemplo, TensorFlow[4]. Essa API provê uma estrutura que permite compilar ANNs combinando camadas de diferentes dimensões e funções de ativação, tornando o ciclo de desenvolvimento de novos modelos de aprendizado de máquina muito mais rápido.

Utilizamos o *Keras* para implementar o algoritmo com aprendizagem supervisionada ANN avaliado nesta dissertação. Seu desenvolvimento foi realizado da forma descrita a seguir.

Aplicamos o modelo sequencial de rede neural disponível no *Keras*, que permite inserir camadas em série, onde a saída da primeira camada serve como entrada da segunda camada, e assim por diante. Adicionamos camadas do tipo *Dense*, que tem como objetivo calcular uma função de ativação em um conjunto de dados de entrada e seus pesos. A camada de entrada recebeu a matriz com a dimensão seguindo o número de *features*

[3] *Application Programming Interface* - API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.

[4] Tensorflow é a biblioteca mais utilizada hoje para modelos de aprendizado profundo (em inglês, *deep learning*). Oferece diferenciação automática para realizar a retropropagação corretamente, permitindo que seja construído qualquer modelo de aprendizado de máquina.

selecionadas a partir dos conjuntos de dados de entrada. As camadas de entrada e a oculta receberam como paramêtro a função de ativação *ReLU*.

Nosso estudo de caso tem classificação binária e apenas uma dimensão de saída. Para classificadores binários, o interesse está na probabilidade de dado de entrada pertencer a uma classe ou outra. Para isso, alteramos a função de ativação da camada de saída para *Sigmoide*. Por fim, utilizamos *Adam* [90] como a função que define como os pesos da rede neural são atualizados e *binary crossentropy* como a função que calcula a diferença entre os dados de teste e os dados de validação. Um trecho de código em linguagem *Python* com as definições da rede neural é descritro abaixo:

```
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu',))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(out_dim, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='Adam')
```

Tanto para o treinamento quanto para o teste, utilizamos a técnica de validação cruzada[5] com *10-Fold*.

Scikit-learn

Scikit-learn é um pacote de aprendizado de máquina de código aberto em linguagem *Python* mais abrangente e bastante utilizado atualmente [91]. Inclui uma coleção de métodos de aprendizado de máquina implementados com eficiência.

Utilizamos o *Scikit-learn* para implementar os algoritmos de aprendizagem supervisionada SVM e RF, avaliados nesta dissertação.

Para o SVM, utilizamos um modelo baseado no pacote *Library for SVMs (libSVM)*[6], chamado *C-Support Vector Classification (SVC)* [92]. O trecho de código em linguagem *Python* com essa definição da SVM é descritro abaixo:

```
clf = svm.SVC()
clf.fit(X_train, y_train)
```

[5]Validação cruzada - é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados de entrada.

[6]LibSVM é um software que implementa a SVM, tanto para classificação quanto para regressão, que suporta classificação múltipla

Por padrão, esse modelo utiliza a função *kernel* RBF, e identifica os valores para os meta-parâmetros C e γ da SVM através de uma busca em grade.

Para a RF, utilizamos o modelo *RandomForestClassifier*. Por padrão, esse modelo constrói as árvores de decisão usando a técnica *Bagging*. O número padrão de árvores na floresta é 100 e como critério usamos entropia.

```
clf = RandomForestClassifier(criterion="entropy", n_estimators=100,
bootstrap=True)
clf.fit(X_train, y_train)
```

Para o treinamento e teste dos dois algoritmos (SVM e RF) utilizamos a técnica de validação cruzada com *10-Fold*.

Capítulo 4

Construção dos snoRNAs artificiais e avaliação de algoritmos de AM

Neste capítulo, descrevemos dados e apresentamos os métodos usados para realizar a avaliação de desempenho de algoritmos de AM para predição de snoRNAs. Começamos a Seção 4.1 com uma descrição dos dados biológicos usados como ponto de partida para criação dos snoRNAs artificiais e depois mostramos o tratamento de dados realizado para obter sequências intrônicas com snoRNAs biológicos. Em seguida, na Seção 4.2, o método proposto nesta dissertação é apresentado em duas partes. Primeiro, na Seção 4.2.1, descrevemos a construção dos conjuntos de dados com snoRNAs artificiais. Depois, na Seção 4.2.2, detalhamos como foi proposta a avaliação dos métodos de AM (SVM, ANN e RF), incluindo uma avaliação de um método de comparação de sequências primárias, utilizando o BLAST [27].

4.1 Dados biológicos e artificiais

Os dados biológicos utilizados nos experimentos foram extraídos da espécie marinha *Ciona intestinalis*, obtidos no *RefSeq* do National Center for Biotechnology Information (NCBI) [93]. Utilizamos o genoma e as sequências de transcrição no formato fasta, assim como a anotação do genoma no formato GFF3. A Tabela 4.1 descreve os detalhes dos arquivos utilizados.

Tabela 4.1: Dados da *C. intestinalis* obtidos no NCBI.

Dado	*Accession prefix RefSeq*	Arquivo
Genoma	NC_	GCF_000224145.3_KH_genomic.fna
Transcrito	XR_	GCF_000224145.3_KH_rna.fna
GFF	NC_	GCF_000224145.3_KH_genomic.gff

A *C. intestinalis* é um urucordado (tunicados)[1], possui um modelo interessante para estudar as origens e evolução dos cordados[2], uma vez que possui um genoma compacto, o que é vantajoso para estudos evolutivos do desenvolvimento [94]. Além disso, os tunicados são os parentes mais próximos dos vertebrados e têm sido escolhidos com sucesso como modelos para estudar a evolução dos cordados [95].

Para analisar o desempenho dos métodos de AM para prever snoRNAs, da forma mais rigorosa possível, dados artificiais foram cuidadosamente preparados, para serem bastante similares aos dados biológicos.

Primeiro, para obter snoRNAs biológicos sem repetições e sem correlação entre eles, propusemos etapas, que estão descritas na Figura 4.1. Na etapa A usamos a ferramenta GFF-Ex [96] para recuperar sequências intrônicas do arquivo GFF3 o comando utilizado é descrito abaixo:

```
gffex -in GCF_000224145.3_KH_genomic.gff -db GCF_000224145.3_KH_genomic.fna
```

Além disso, na etapa A filtramos as sequências de snoRNAs do arquivo de transcrição. Em seguida, na etapa B, a ferramenta Blastn foi executada para localizar sequências intrônicas similares aos snoRNAs filtrados, os comandos utilizados são descritos abaixo:

```
makeblastdb -in introns.fa -dbtype nucl
blastn -in snoRNAs.fa -db introns.fa
```

A fim de obter sequências intrônicas não correlacionados, na etapa C, usamos o Blastn para calcular a similaridade entre pares de sequências intrônicas. Foi definido um corte de 80% de *identities*[3] (identidade de sequência), ou seja, duas sequências com 80% ou mais de identidade eram consideradas similares, e uma delas descartada. Ao final dessa etapa, foram obtidas sequências intrônicas contendo snoRNAs, sem correlação entre cada par.

4.2 Descrição do método

O método desenvolvido nesta dissertação está dividido em duas partes principais: a criação de snoRNAs artificiais e a avaliação dos métodos de AM. O método proposto tem seis etapas, iniciando com os dados biológicos, conforme descrito na Seção 4.1 (veja Figura 4.2):

[1] Urucordado é um subfilo de animais marinhos, que pertencem ao filo Cordados.

[2] Cordados (em inglês, *Chordata*) constituem um filo dentro do reino Animalia que inclui os vertebrados, os anfioxos e os urocordados.

[3] *Identities* constituem um parâmetro do Blast que indica o percentual de correspondências idênticas entre duas sequências.

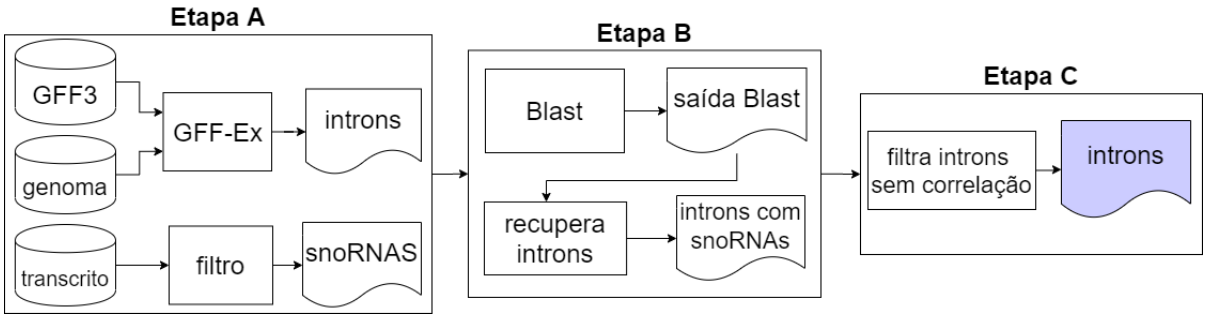

Figura 4.1: Etapas do tratamento de dados para snoRNAs biológicos da *C. intestinalis*, de modo a obter um conjunto de snoRNAs, sem correlação entre cada par deles.

1. Executa snoReport 2.0 com as sequências de introns selecionadas na etapa anterior para identificar as classes de snoRNAs - C/D *box* ou H/ACA box;
2. Escolhe sequências de snoRNAs representativas da saída do snoReport 2.0 para aplicar mutações;
3. Constrói a árvore de mutações usando as sequências de saída do snoReport 2.0, criando conjuntos com porcentagens cumulativas de mutações;
4. Extrai *features* para cada sequência, considerando os conjuntos positivos e negativos obtidos da árvore de mutação;
5. Constrói bases de dados para os algoritmos de AM com o mesmo número de vetores de amostras (*features*) para os conjuntos positivos (1) e negativos (0);
6. Executa os algoritmos de AM e analisa seus resultados.

4.2.1 Criação dos snoRNAs artificiais

O snoReport 2.0 foi usado para identificar sequências das duas classes de snoRNAs (C/D *box* ou H/ACA box) conforme descrito na Etapa 1 da Figura 4.2, a partir das sequências intrônicas cuidadosamente selecionadas a partir dos snoRNAs biológicos de *C. intestinalis*, como descrito na Seção 4.1. O comando utilizado na ferramenta snoReport 2.0 é descrito abaixo:

```
snoreport_2 -i introns.fa -HACA -CD -o snoRNAs_identificados.fa
```

A partir dessas sequências, dois representantes foram escolhidos de forma aleatória para gerar as árvores de mutação. Um C/D *box* snoRNA e um H/ACA *box* snoRNA (Figura 4.2 - Etapa 2). O comprimento total de nucleotídeos do intron e do snoRNA biológico é 833 e 96, para C/D *box*, e 1.577 e 173, para H/ACA *box*, respectivamente.

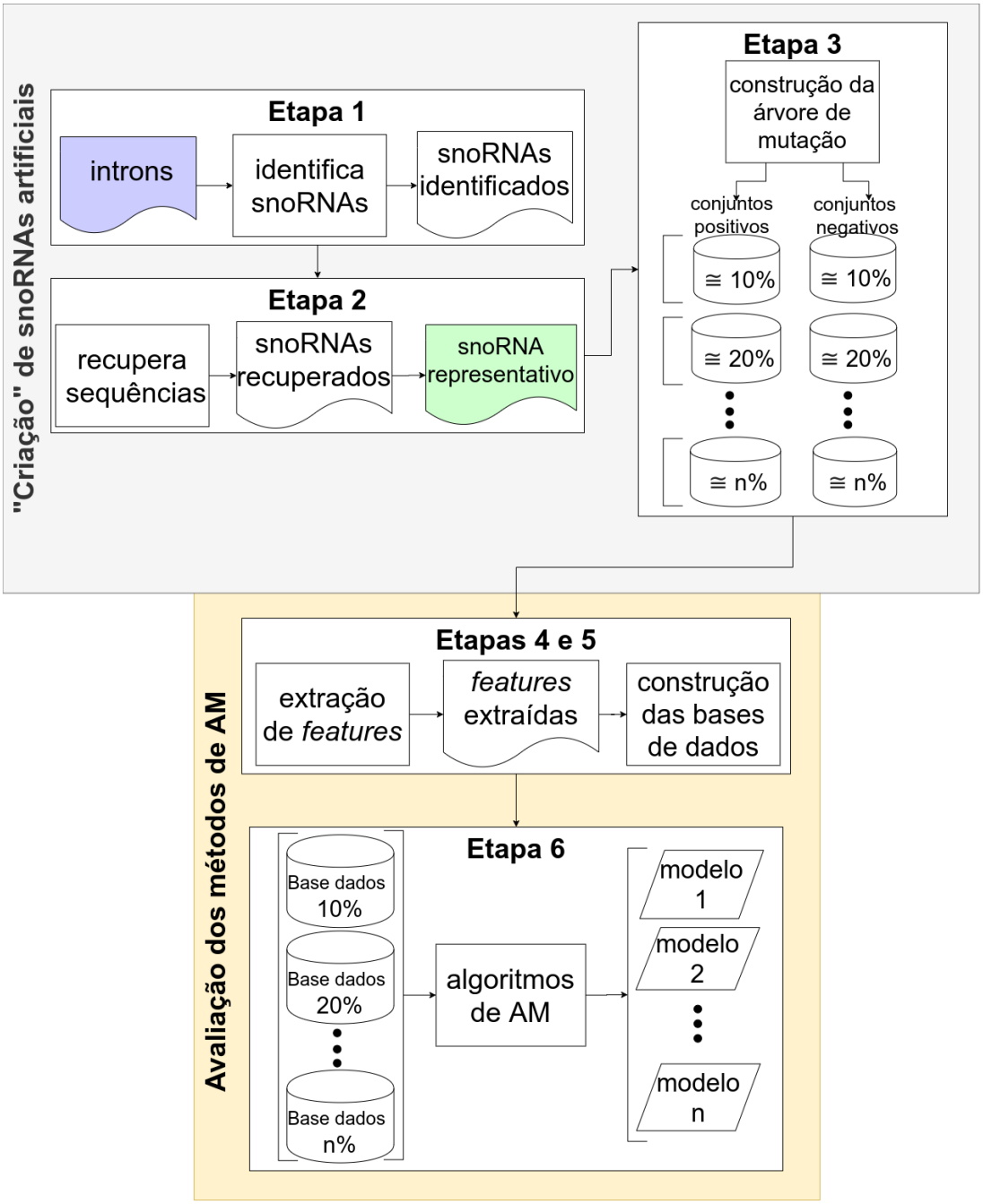

Figura 4.2: Etapas do método para construir os conjuntos de snoRNAs artificiais e, com esses dados, avaliar os métodos de AM (SVM, ANN e RF).

Com esses dados, o Algoritmo 1 descreve a construção da árvore de mutação contendo os snoRNAs artificiais, com suas bases mutadas de forma cumulativa. Os conjuntos positivos e negativos para a avaliação dos métodos de classificação, como descrito na Etapa 3 da Figura 4.2, foram construídos a partir do reconhecimento (ou não), pelo snoReport 2.0, dos snoRNAs mutados (armazenados na variável $sMutated$) como snoRNAs.

Algorithm 1: Construção da árvore de mutação.

Data: s// `sequência de snoRNA representativo` $intron$// `intron contendo a sequência de snoRNA`
Result: T// `árvore de mutação`

```
1 faça mutações em s: sMutated substitui s no intron
2 if intron tem um snoRNA then
3     if sMutated é identificada como snoRNA no mesmo lócus do snoRNA
        original then
4         sMutated é inserida em T, para ser sucessivamente mutada;
5         sMutated é armazenado no conjunto positivo correspondente ao nível de
            mutações de bases (10% ... 50%) na árvore;
6     else
7         sMutated não é inserida em T, mas é armazenada em um conjunto
            separado;
8     sMutated não é inserida em T e é armazenada em um conjunto negativo;
```

A raiz da árvore corresponde a uma das sequências representativas de snoRNAs biológicos s, descritas antes. Em cada etapa, as mutações em bases (substituição, deleção e inserção) são aplicadas a s em posições aleatórias. Para obter mutações cumulativas, as posições que foram mutadas no ramo equivalente da árvore são respeitadas, isto é, não são mutadas novamente. A sequência mutada resultante $sMutated$ é então reinserida nas sequências intrônicas. Definimos $sMutated$ como um snoRNA artificial, ou seja, como um verdadeiro positivo, se for reconhecido no intron, no mesmo *locus* que o snoRNA orginal, como um snoRNA, pela ferramenta snoReport 2.0. As árvores de mutação são construídas de forma independente, para cada um dos snoRNAs biológicos inicialmente identificados. O processo de mutação imita uma população, tendo sido definida por um tamanho fixado, tendo sido cada nível da árvore definido com um número máximo N de sequências, de acordo com o custo computacional de tempo para geração da árvore utilizando um servidor com as seguintes características:

```
RAM: 4 TB
Cores CPU: 10
width:  64 bits
```

A sequência a ser mutada é escolhida aleatoriamente nesta população. Os *nós* (rotulados por $sMutated$) da árvore de mutação são gerados com um número F, também fixado, de filhos.

O **conjunto positivo** consiste nas sequências mutadas $sMutated$ inseridas nos introns que ainda são reconhecidas como snoRNAs. Como as mutações são cumulativas, o conjunto positivo compreende sequências com aproximadamente a mesma porcentagem de mutação em posições mutadas, com relação à sequência de snoRNA biológico, esco-

lhido inicialmente. Teremos então porcentagem de mutações de 10%, 20%, 30%, 40% e 50%. O **conjunto negativo** é formado pelas sequências mutadas $sMutated$ inseridas nos introns que não puderam mais ser identificadas como snoRNAs. Cada conjunto negativo é composto por sequências com aproximadamente a mesma porcentagem de mutação, de forma semelhante ao conjunto positivo.

Observamos que, além dos conjuntos positivos e negativos, foi gerado um terceiro conjunto separado com sequências de introns que ainda são reconhecidos como snoRNAs, mas cujas sequências mutadas ($sMutated$) foram identificados em *loci* diferentes daqueles das sequências de snoRNAs originais. Para uma melhor compreensão dos conjuntos positivos e negativos, ilustramos um exemplo com 10% das posições mutadas na Figura 4.3.

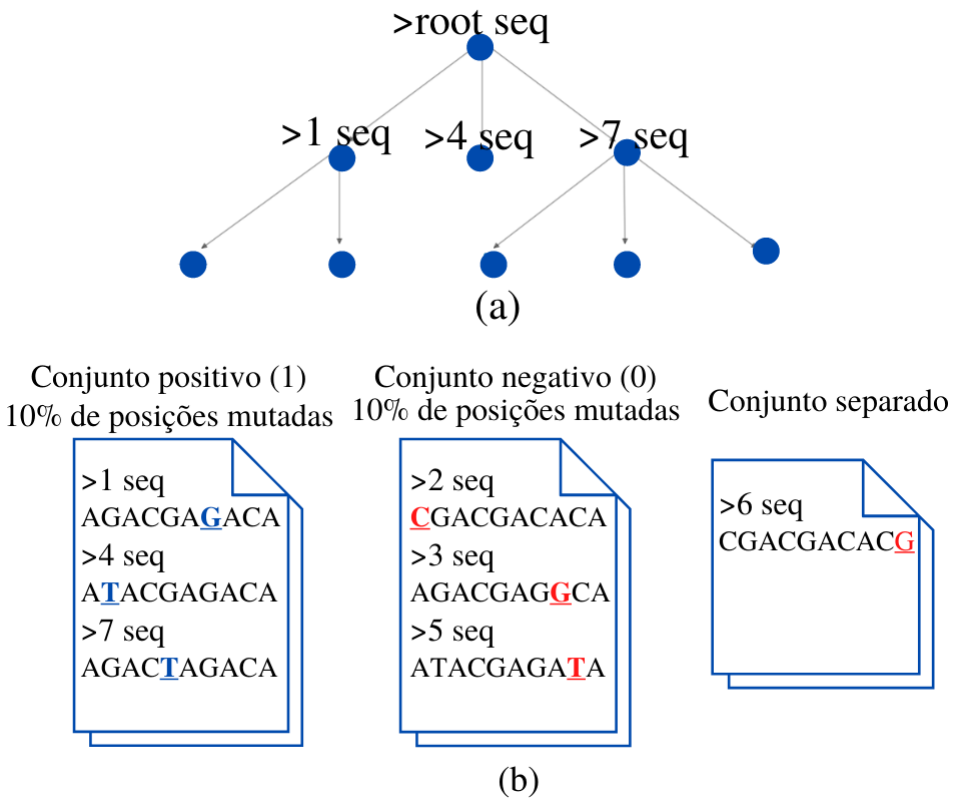

Figura 4.3: (a) Exemplo de uma árvore de mutação com 3 filhos e um máximo de 5 nós por nível. As sequências que rotulam os nós no nível 1 da árvore são armazenadas no conjunto positivo. (b) Exemplo de um conjunto positivo e um conjunto negativo com 10% de posições mutadas. No caso deste exemplo, cada sequência tem tamanho de 10 nucleotídeos e por fim, o conjunto separado.

4.2.2 Avaliação dos métodos de AM

Inicialmente, foram extraídas *features* para cada sequência armazenada nos conjuntos positivos e negativos, obtidos a partir de cada árvore de mutação (Figura 4.2 - Etapa 4). As

features extraídas para sequências C/D *box* e H/ACA *box*, são descritas nas Tabelas 4.2 e 4.3, respectivamente. Essas *features* possuem diversas propriedades características de candidatos a snoRNAs, tendo sido cuidadosamente selecionadas por Oliveira e co-autores [3]. Nesta avaliação, usamos exatamente as mesmas *features*.

Tabela 4.2: *Features* extraídas de um candidato a C/D *box* snoRNA.

Feature **(característica)**	**Descrição**
*mfe*C	MFE da estrutura secundária com restrições em RNAfold
mfe	MFE da estrutura secundária sem restrições em RNAfold
E_{avg}	Média da MFE
E_{stdv}	Desvio padrão da MFE
ls	Comprimento do *stem* terminal
Dcd	Distância entre os *boxes* C a D
C_{score}	Escore do *box* C
D_{score}	Escore do *box* D
GC	Conteúdo *GC*
zscore	*zscore* obtido por RNAz
bpStem	Número de pares de bases no terminal
lu5	Número de nucleotídeos não pareados dentro do *stem* antes do *box* C
lu3	Número de nucleotídeos não pareados dentro do *stem* depois do *box* D
stemU npCbox	Número de nucleotídeos não pareados entre o *stem* e o *box* C
stemU npDbox	Número de nucleotídeos não pareados entre o *box* D e o *stem*

As bases de dados utilizadas para avaliação dos classificadores de AM (SVM, ANN e RF) são arquivos no formato *Comma-separated values (CSV)*. Essas bases de dados são compostas pelo mesmo número de vetores de *features* dos conjuntos positivo (1) e negativo (0). Como sequências diferentes podem produzir os mesmos valores de *features*, aqueles duplicados são removidos dos dois conjuntos, positivo e negativo.

Para cada árvore de mutação, identificamos o conjunto positivo com o menor número de vetores de *features* $= n$. Construímos bases de dados compreendendo diferentes números para n, de forma que n vetores de *features* dos dois conjuntos (positivo e negativo) são selecionados aleatoriamente de acordo com a porcentagem de mutação. Por exemplo, em uma árvore o conjunto de dados com menor número $n = 1.000$ de vetores de *features* é aquele com 30% de mutação. Portanto, as bases de dados dessa árvore são formadas por $n = 1.000$ vetores de *features* do conjunto positivo e $n = 1.000$ do negativo (Figura 4.2 - Etapa 5).

Em relação à normalização de dados, usamos a linear, também chamada de normalização por interpolação linear. Ela consiste em considerar os valores mínimo e máximo

Tabela 4.3: *Features* extraídas de um candidato a H/ACA *box* snoRNA.

Feature **(característica)**	**Descrição**
*mfe*C	Energia livre mínima (MFE) da estrutura secundária com restrições em RNAfold
AC, GU e GC	Conteúdo AC, GU e GC
zscore	Pontuação *zscore* computada por RNAz
Hscore	Escore do *box* H calculado a partir da matriz de pesos de posição específica
ACAscore	Escore do *box* ACA calculado a partir da matriz de pesos de posição específica
LseqSize	Número de nucleotídeos antes do *box* H
RseqSize	Número de nucleotídeos entre os *boxes* H e ACA
LloopSC	Comprimento do *loop*, onde encontra o *pocket* contendo a região alvo, mais próximo do *box* H
RloopSC	Comprimento do *loop*, onde encontra o *pocket* contendo a região alvo, mais próximo do *box* ACA
LloopY C	Simetria do *loop* encontrado perto do *box* H
RloopY C	Simetria do *loop* encontrado perto do *box* ACA
LloopSym	Simetria de todos os loops antes do *box* H
RloopSym	Simetria de todos os loops antes do *box* ACA

de cada *feature* e mapear os valores de uma amostra no intervalo fechado de 0 a 1 [97]. A normalização linear preserva as distâncias proporcionais entre os dados normalizados com as distâncias entre os dados originais. A definição é dada pela Equação 4.1, onde v é o valor a ser normalizado, min representa o valor mínimo da amostra, neste caso da *feature*, e max é o valor máximo. Finalmente, V' corresponde ao valor normalizado.

$$V' = \frac{v - min}{max - min} \tag{4.1}$$

As métricas para avaliar o valor preditivo dos classificadores de AM utilizam taxas de erros e acertos, as quais são obtidas através dos resultados de uma matriz de confusão. A matriz de confusão de uma hipótese h oferece uma medida efetiva do modelo de classificação, ao comparar o número de classificações corretas com as classificações preditas, para cada classe, sobre um conjunto de amostras. O número de acertos, para cada classe, localiza-se na diagonal principal da matriz e os demais elementos representam erros na classificação.

A Figura 4.4 apresenta um exemplo de uma matriz de confusão para classificação binária, composta pelos seguintes resultados:

- Verdadeiro positivo (do inglês, *true positive — TP*): ocorre quando, no conjunto verdadeiro, a classe que estamos buscando predizer foi prevista corretamente;
- Falso positivo (do inglês, *false positive — FP*): ocorre quando, no conjunto verdadeiro, a classe que estamos buscando predizer foi prevista incorretamente;
- Verdadeiro negativo (do inglês, *true positive — TN*): ocorre quando, no conjunto verdadeiro, a classe que não estamos buscando predizer foi prevista corretamente;
- Falso negativo (do inglês, *false positive — FN*): ocorre, quando no conjunto verdadeiro, a classe que não estamos buscando predizer foi prevista incorretamente.

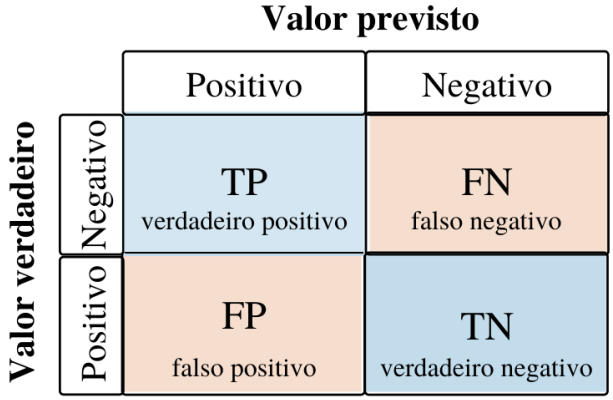

Figura 4.4: Exemplo de uma matriz de confusão para classificação binária, com classes positiva e negativa.

Dentre as métricas comumente utilizadas para avaliar o valor preditivo dos classificadores de AM, pode-se citar: Sensibilidade, Especificidade, Acurácia, Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall*, Precisão e Curva de Característica Operacional do Receptor (do inglês, *Receptor Operational Curve* - curva ROC) [98, 99], definidas na Tabela 4.4.

Para avaliar o desempenho dos algoritmos de AM (SVM, ANN e RF), implementados conforme descrito na Seção 3.5, utilizamos as seguintes métricas de avaliação (Figura 4.2 - Etapa 6) - AUC, MCC, *Recall*, Precisão e curva ROC: AUC avalia a qualidade das previsões do modelo; MCC a medida da qualidade da classificação binária; *Recall* verifica com que frequência o classificador está encontrando amostras de uma classe; precisão constata, das amostras classificadas como certas, quantas realmente são; e a curva ROC é usada para gerar uma estatística de resumo.

Para comparar os resultados do desempenho dos algoritmos de AM com métodos que usam comparações de sequências primárias para predição de snoRNAs, utilizamos o Blastn [54, 27]. E definimos uma taxa de acerto média M para os alinhamentos com

Tabela 4.4: Métricas de avaliação dos algoritmos de AM.

Métrica	**Definição**
Acurácia	$AC = \frac{TP+TN}{P+N}$
MCC	$MCC = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
Recall/Sensibilidade	$R = \frac{TP}{(TP+FN)}$
Precisão	$P = \frac{TP}{(TP+FP)}$
Especificidade	$E = \frac{TN}{(TN+FP)}$
Curva ROC	$TPR = \frac{TP}{(TP+FN)}$ e $FPR = \frac{FP}{(FP+TN)}$
AUC	$AUC = \int_{x=0}^{1} TPR(FPR^{-1}(x))dx$

Blastn. M é definido pela Equação 4.2, onde S é o número de sequências alinhadas e N é o número total de sequências (comprimento do arquivo de consulta).

$$M = \frac{S}{N} \tag{4.2}$$

Capítulo 5

Resultados e Discussão

Neste capítulo, discutimos os resultados da execução do método proposto nesta dissertação. Inicialmente, apresentaremos na Seção 5.1 os resultados de um estudo para verificar o impacto das mutações em determinadas posições dos snoRNAs, o que é útil para o desenvolvimento de algoritmos para predição desses ncRNAs. Em seguida, discutimos os experimentos realizados, divididos em dois grandes grupos. Na Seção 5.2, são apresentados tanto os resultados obtidos com o Blast quanto com os algoritmos de AM (SVM, ANN e RF) com árvores de mutação contendo 10 filhos e $N = 3000$ (número de sequências dos conjuntos positivos) para C/D *box* e $N = 2000$ (número de sequências dos conjuntos negativos) para H/ACA *box*. Em seguida, na Seção 5.3, são mostrados os resultados do Blast e dos algoritmos de AM com árvores de mutação com 100 filhos e $N = 5000$ para ambos os snoRNAs, C/D *box* e H/ACA *box*.

5.1 Impacto de mutações na predição de snoRNAs

Nesta seção, discutimos o impacto de mutações em determinadas posições de um snoRNA. Nesse estudo, testamos, para todas as posições da sequência do snoRNA, substituições dos nucleotídeos (A, C, T e G) por nucleotídeos diferentes. Essas substituições não foram cumulativas, tendo sido mutada apenas uma posição da sequência.

Cada um dos nucleotídeos de cada posição de uma sequência de snoRNA foi mutado para todas as outras possibilidades. Por exemplo, supondo que a posição 10 de uma sequência de snoRNA possui o nucleotídeo A, todas as outras possibilidades serão sucessivamente testadas, no caso, C, T e G. A cada substituição, a sequência foi testada no snoReport 2.0 [3] para verificar se a sequência mutada ainda era predita como snoRNA.

Esse método permitiu identificar posições dos snoRNAs que, ao sofrerem a substituição por qualquer nucleotídeo diferente do original, impediam que a sequência fosse reconhecida como snoRNA.

Este estudo inicial foi interessante para verificar que os *boxes* dos snoRNAs H/ACA e C/D são importantes para sua predição. A Figura 5.1 mostra o exemplo de posições de uma sequência de H/ACA *box* que, se sofrerem uma substituição, impedem que a sequência seja reconhecida como snoRNA. É possível identificar essas posições nos *boxes* H e ACA da sequência. O estudo completo está no Anexo II, realizado em sequências aleatórias do transcrito da C. *intestinalis* para sequências H/ACA *box* e para C/D *box*.

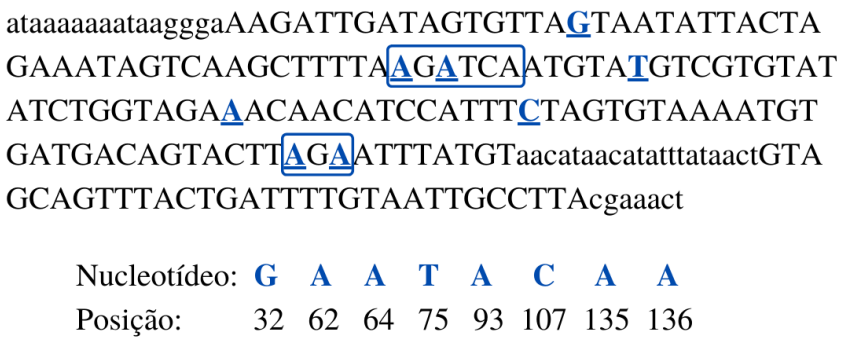

Figura 5.1: Exemplo de uma sequência de H/ACA *box* snoRNA, com posições importantes marcadas em azul. Essas posições, se sofrerem uma substituição, impedem que a sequência seja reconhecida como snoRNA. Os *boxes* H e ACA são destacados no retângulo.

5.2 Primeiro grupo de experimentos

Os experimentos foram executados para ambas as classes de snoRNAs (C/D *box* e H/ACA box), considerando mutações nas bases - substituição, inserção, remoção. Os *nós* das árvores de mutação geradas possuem 10 filhos, enquanto cada nível apresenta um máximo de $N = 3000$ *nós* (sequências mutadas), para C/D *box*, e $N = 2000$ para H/ACA *box*. O valor de N foi fundamentado através do custo computacional Seção 4.2.1 para a geração de cada árvore. As árvores com H/ACA *box* snoRNA demandam mais tempo para criação, por conseguinte o valor de N é menor.

Experimento com Blast

Inicialmente, realizamos uma comparação Blastn com parâmetros *default*, usando como consulta as sequências de snoRNA biológicos (raízes de árvore) e como bancos de dados cada um dos conjuntos de dados positivos gerados pelas árvores de mutação (com taxas de mutação de 10%, 20%, 30%, 40% e 50% e $N = 3.000$ para C/D *box* e $N = 2.000$ para H/ACA box). As sequências, os snoRNAs biológicos e os mutados, foram testados dentro dos introns e considerando também apenas as próprias sequências. Para quantificar o sucesso desta busca de homologia baseada em sequência, calculamos uma taxa de acerto média M.

Com os snoRNAs dentro do intron, Blastn sempre encontrou *matches*, então neste caso $M = 100\%$. No entanto, na maioria dos casos, eles não correspondiam apenas ao snoRNA alvo, mas eram acertos espúrios em outras partes do intron. Desconsiderando as sequências mutadas, obtivemos essencialmente resultados similares com Blastn, independente do modelo de mutação. Esses resultados representam falsos positivos, além disso, como a região do snoRNA é muito pequena quando comparado ao comprimento total da sequência intrônica, a probabilidade de ter *matches* aumenta.

Para obter uma comparação mais justa do método Blast com os algoritmos de AM, utilizamos o Blastn com os próprios snoRNAs. Observamos que os resultados desse experimento são bastante similares, independentemente do tipo de mutação. Assim, discutimos em detalhes apenas os resultados derivados da substituição, como exemplo.

Os resultados mostraram uma baixa similaridade de sequência. Como mostra a Tabela 5.1 Blastn encontrou *matches* apenas para os conjuntos com taxa de mutação de 10% e 20%. Os conjuntos com taxas de mutação de 30%, 40% e 50%, respectivamente, não foram encontradas similaridades de sequências em nehuma classe C/D *box* e H/ACA *box*.

Tabela 5.1: Média de sequências que alcançaram alinhamento no Blastn.

snoRNA	Conjunto de dados	M
C/D	10%	18,9%
	20%	0,5%
H/ACA	10%	73,6%
	20%	2,9%

As tabelas 5.2 e 5.3 mostram para C/D *box* e H/ACA *box*, respectivamente, o número de sequências dos conjuntos positivos (Figura 4.2 - Etapa 3), $n : (x)$ o número de instâncias com *features* extraídas dos conjuntos positivos ($x = 1$)) e negativos ($x = 0$) (Figura 4.2 - Etapa 4) e o número total (soma) de instâncias das duas bases de dados (Figura 4.2 - Etapa 5).

Tabela 5.2: C/D box

Árvore	Sequências	n:(1)	n:(0)	Base de dados
substituição	3000	2574	2574	5148
inserção	3000	1795	1795	3590
remoção	3000	2232	2232	4464

Tabela 5.3: H/ACA box

Árvore	Sequências	n:(1)	n:(0)	Base de dados
substituição	2000	1835	1835	3670
inserção	2000	1009	1009	2018
remoção	2000	832	832	1664

5.2.1 C/D box

Nesta seção, discutimos os resultados dos experimentos para a classe dos C/D *box* snoRNAs.

Experimentos com métodos de AM

Como dito antes, realizamos experimentos com os algoritmos ANN, SVM e RF para mutações (substituição, inserção e remoção) de bases com taxas de mutação de 10%, 20%, 30%, 40% e 10%. A seguir, discutimos os resultados obtidos para cada uma das mutações.

- **Substituição**

A Tabela 5.4 mostra os resultados dos três algoritmos de AM para a mutação do tipo substituição. Os classificadores ANN e SVM apresentaram resultados muito semelhantes em todas as métricas de avaliação. Podemos observar que os classificadores ANN e o SVM apresentaram valores decrescentes. Com 10% de mutação, ANN obteve $MCC = 98,84\%$ e SVM $MCC = 96,97\%$, enquanto que, com 50% de mutação, ANN e SVM obtiveram $MCC = 63,36\%$ e $MCC = 44,44\%$, respectivamente.

A Figura 5.2, mostra as curvas ROC para todos os classificadores. Os modelos ANN e SVM alcançaram uma boa medida de separabilidade para conjuntos de dados com mutação de 10%, 20%, 30% e 40%. No entanto, para conjuntos de dados com 50% de mutação, eles mostraram menos capacidade de separação de classes. RF obteve bons resultados em todas as métricas de avaliação para 20% e 40% de mutação. Diferente do resultado dos outros classificadores ANN e SVM, para os conjuntos de dados 10% e 30% os resultados não foram bons, observamos para 10%, 30% e 50%,s - predição com $MCC = 56,61\%$, $MCC = 47,40\%$ e $MCC = 57,63\%$ respectivamente.

Para investigar casos onde as características biológicas não são conhecidas, também testamos os algoritmos de AM com um número reduzido de *features*, neste caso - zscore, ls, Dcd, GC, lu5 e lu3. A Tabela 5.5 apresenta os resultados dos três algoritmos de AM.

Como pode ser visto na Figura 5.5, os três classificadores não obtiveram bons resultados, para todos os conjuntos de dados (de 10% a 50%).

Tabela 5.4: Resultados dos algoritmos de AM para a mutação do tipo substituição com todas as *features*, para predição de C/D *box* snoRNA - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	99.42	98.84	99.18	99.65
	20%	97.79	95.64	96.16	99.40
	30%	93.93	88.65	88.35	99.43
	40%	93.96	88.55	90.17	97.56
	50%	80.32	63.36	72.27	86.16
SVM	10%	98.47	96.97	97.71	99.21
	20%	96.14	92.35	94.49	97.71
	30%	93.55	87.31	93.55	93.55
	40%	88.53	77.45	89.59	87.72
	50%	71.47	44.44	77.90	69.03
RF	10%	77.21	56.61	57.32	95.23
	20%	91.19	82.96	83.30	98.89
	30%	72.98	47.40	47.38	97.13
	40%	93.03	86.52	90.80	95.04
	50%	78.40	57.63	71.15	83.23

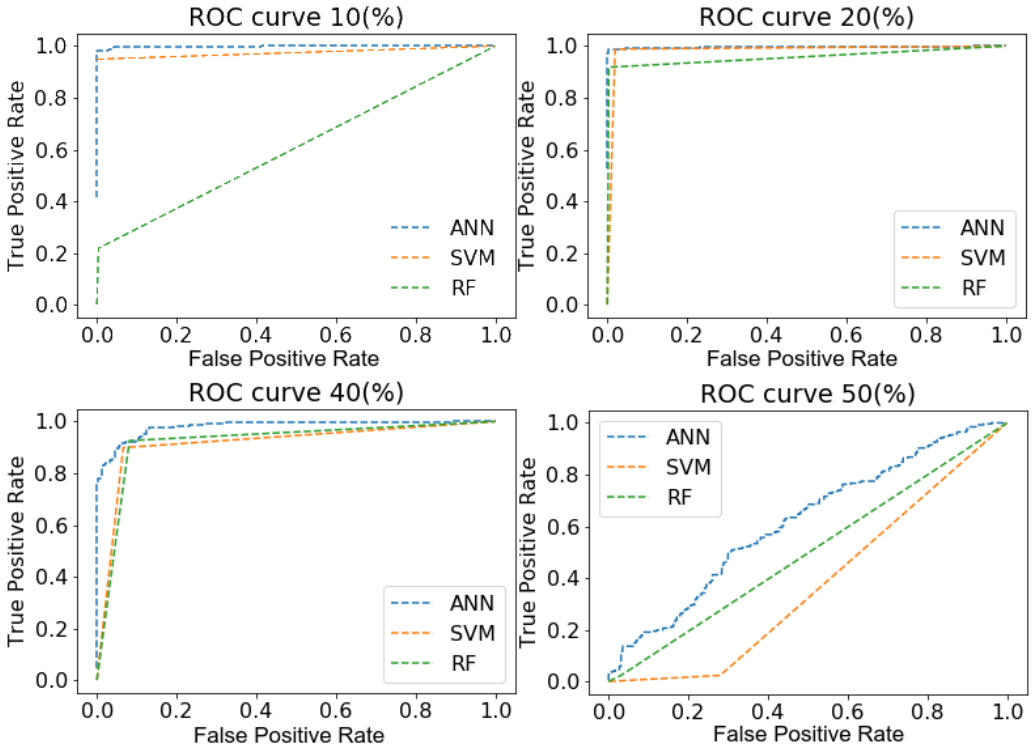

Figura 5.2: Curva ROC dos conjuntos de dados com 10%, 20%, 40% e 50% de taxas de mutação, para a mutação do tipo substituição, utilizando todas as *features*.

Tabela 5.5: Resultados dos algoritmos de AM para substituição, com número de *features* reduzidas, para predição de C/D *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	68.31	36.97	57.62	73.26
	20%	57.13	8.42	35.39	62.73
	30%	61.64	20.49	44.85	67.56
	40%	68.94	37.01	54.83	76.44
	50%	63.46	23.45	51.14	67.88
SVM	10%	71.94	45.97	82.15	68.23
	20%	61.59	21.94	62.39	61.45
	30%	64.98	29.15	67.22	64.39
	40%	70.88	42.76	78.40	68.18
	50%	65.46	30.62	68.93	64.51
RF	10%	57.15	11.21	22.74	72.91
	20%	52.38	2.76	16.71	58.31
	30%	52.34	1.02	24.59	55.29
	40%	54.40	4.65	23.19	61.76
	50%	58.10	13.51	41.93	61.97

Na Figura 5.3), as curvas ROC dos conjuntos de dados com taxas de mutação de 10%, 20%, 40% e 50% são mostradas. Observamos que esses modelos de fato não alcançaram uma boa capacidade preditiva de C/D *box*.

Desses dois experimentos com a substituição, podemos perceber que o conjunto de *features* é bastante importante para predição do C/D *box*. Se não são conhecidas características biológicas relevantes, o desempenho dos classificadores fica bastante ruim, piorando para as sequências homólogas mais distantes.

- **Inserção**

A Tabela 5.6 mostra os resultados dos três algoritmos de AM para a mutação do tipo inserção.

Os classificadores ANN e SVM apresentaram *Recall* acima de 97% para todos os conjuntos de dados. O classificador ANN apresenta uma boa predição para os conjuntos de dados, $AUC = 91,73\%$ para 40% e $AUC = 92,26\%$ para 50%, como mostrado pelas curvas ROC da Figura 5.4. O classificador SVM apresentou uma boa predição em todos os conjuntos de dados, com 10% de mutação $AUC = 97,71\%$, enquanto que para 50% de mutações, $AUC = 94,01\%$, como também mostrado pelas curvas ROC da Figura 5.4. O modelo de classificador RF não conseguiu realizar a separação de classes em todos os

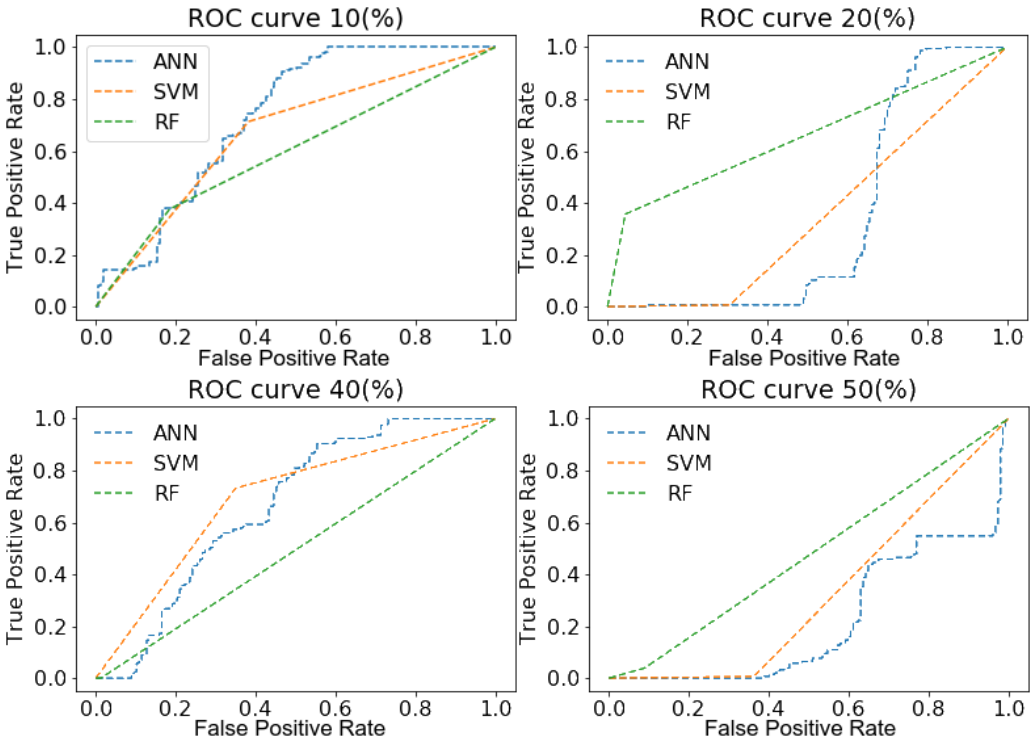

Figura 5.3: Curva ROC dos conjuntos de dados com taxas de mutação de 10%, 20%, 40% e 50%, para substituição, com um número de *features* reduzidas.

Tabela 5.6: Desempenho dos algoritmos de AM para a mutação do tipo inserção, com todas as *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	82.17	65.42	64.87	99.23
	20%	81.70	68.21	63.77	99.48
	30%	78.55	59.77	57.41	99.52
	40%	91.73	84.58	84.97	98.26
	50%	92.26	85.20	86.97	97.26
SVM	10%	97.71	95.53	95.94	99.48
	20%	96.71	93.58	94.16	99.24
	30%	93.76	88.17	88.75	98.64
	40%	94.27	88.97	89.64	98.77
	50%	94.01	88.48	89.37	98.53
RF	10%	53.83	9.61	9.69	82.86
	20%	49.3	-3.63	1.17	31.34
	30%	49.72	0.53	2.67	45.28
	40%	50.11	1.16	0.56	62.50
	50%	54.3	4.88	9.52	90.96

conjuntos de dados, com $AUC = 53,83\%$ para 10% de mutação e $AUC = 54,30\%$ para 50% de mutação, como mostrado nas curvas ROC da Figura 5.4.

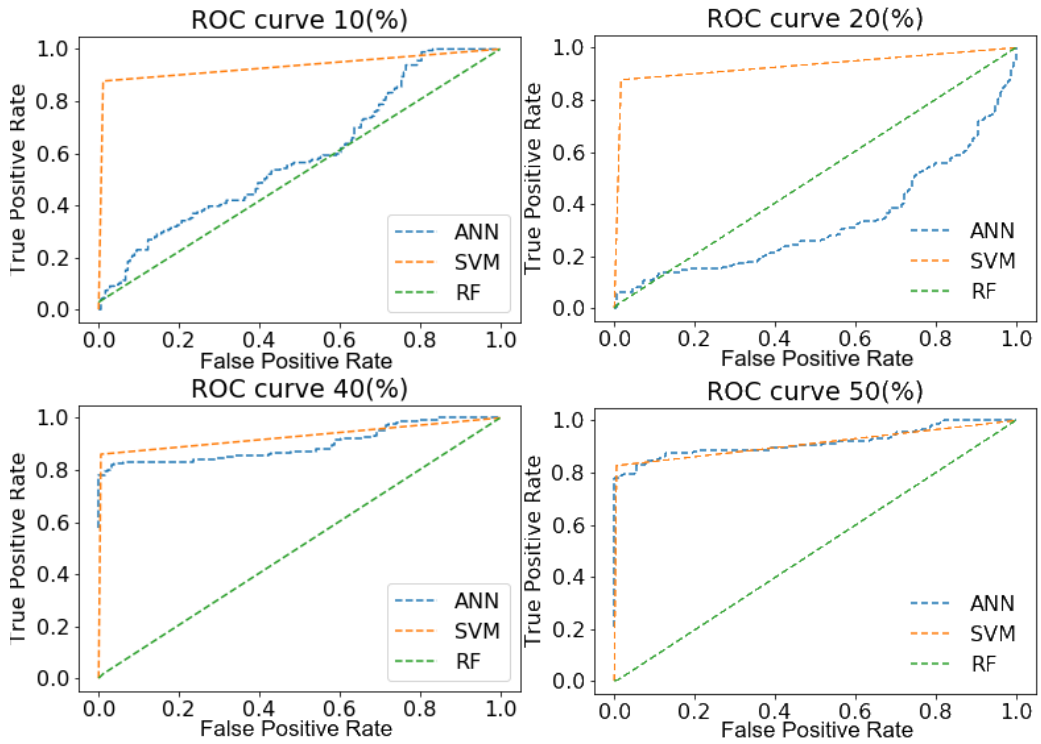

Figura 5.4: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50% de mutação, considerando a inserção, com todas as *features*.

Para investigar os casos em que as características biológicas não são conhecidas, também testamos a mutação de inserção com um número reduzido de *features*, neste caso - zscore, ls, Dcd, GC, lu5 e lu3. A Tabela 5.7 apresenta os resultados.

Quando comparado com os resultados de desempenho considerando todas as *features*, a AUC do RF foi maior, mas ainda não atingiu a capacidade de separação de classes para alguns conjuntos de dados. As AUCs dos classificadores ANN e SVM foram menores, mas permaneceram com bom desempenho $AUC > 70\%$, conforme mostrado pelas curvas ROC da Figura 5.5.

- **Remoção**

A Tabela 5.8 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo remoção. Neste experimento, a árvore de mutação gerou sequências reconhecidas como C/D *box* snoRNAs apenas até 20%. Sequências com mais de 20% de mutação do tipo remoção não foram mais reconhecidas como snoRNAs pelo snoReport 2.0 [100]. Podemos observar que o tamanho da sequência de um snoRNA é uma característica relevante para sua predição.

Tabela 5.7: Desempenho dos algoritmos de AM para inserção com número de *features* reduzidas, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	77.51	55.38	61.29	91.10
	20%	83.34	66.96	71.89	92.86
	30%	81.09	63.31	67.74	91.88
	40%	88.58	78.70	81.11	95.65
	50%	76.71	52.80	60.37	90.34
SVM	10%	84.40	71.19	86.18	83.48
	20%	83.65	69.33	85.25	82.59
	30%	81.42	64.23	79.72	82.38
	40%	90.04	81.56	98.62	84.25
	50%	82.12	67.85	95.85	75.36
RF	10%	71.99	42.91	47.93	93.69
	20%	82.65	65.93	72.81	90.80
	30%	68.08	32.26	45.62	83.19
	40%	78.63	57.31	58.53	97.69
	50%	58.11	9.83	23.04	76.92

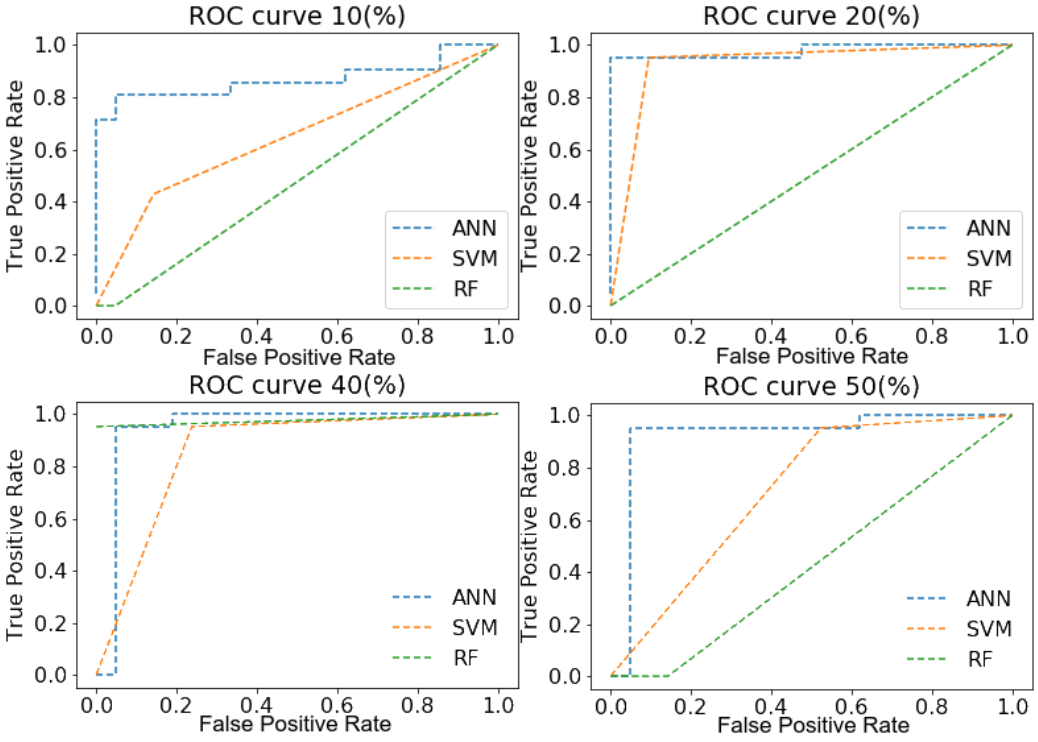

Figura 5.5: Curva ROC dos conjuntos de dados 10%, 20%, 40% e 50%, para inserção com *features* reduzidas.

Conforme a Tabela 5.8 os três classificadores apresentaram valores decrescentes para as métricas de avaliação AUC, MCC e *Recall*. Por exemplo, Com 10% de mutação, as ANN obtiveram $MCC = 82,68\%$, SVM $MCC = 74,54\%$ e a RF $MCC = 68,54\%$, enquanto que, com 20% de mutação, ANN, SVM e RF obtiveram $MCC = 59,97\%$, $MCC = 49,95\%$ e $MCC = 16,44\%$, respectivamente. É possível visualizar esse resultado através das curvas ROC da Figura 5.6.

Tabela 5.8: Desempenho dos algoritmos de AM para a mutação do tipo remoção, com todas as *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	90.62	82.68	95.3	87.14
	20%	78.92	59.97	84.82	75.88
SVM	10%	85.49	74.54	96.64	79.02
	20%	72.53	49.95	85.4	67.94
RF	10%	82.76	68.54	88.36	79.46
	20%	56.63	16.44	13.84	95.96

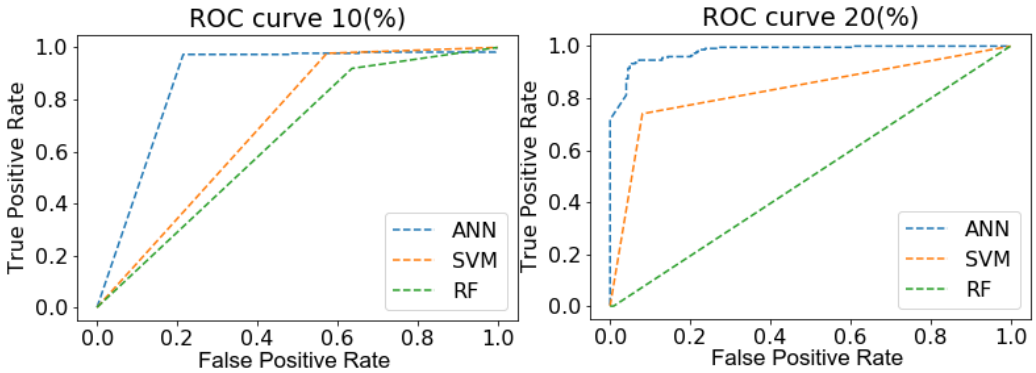

Figura 5.6: Curvas ROC dos conjuntos de dados com taxas de mutação de 10% e 20%, para mutação do tipo remoção, utilizando todas *features*.

Além disso, testamos os desempenhos dos algoritmos de AM com um número reduzido de *features* para a remoção, neste caso - zscore, ls, Dcd, GC, lu5 e lu3. A Tabela 5.9 apresenta os resultados, nos quais os três classificadores não obtiveram bons resultados, para todos os conjuntos de dados (de 10% e 20%).

É possível visualizar esse resultado através das curvas ROC da Figura 5.7.

Assim como na substituição, esses dois experimentos com a remoção mostraram que o conjunto de *features* é bastante importante para predição do C/D *box* por parte de classificadores de AM. Se características biológicas relevantes não são conhecidas, o de-

Tabela 5.9: Desempenho dos algoritmos de AM para a mutação do tipo remoção com um número de *features* reduzidas, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	68.23	40.48	85.83	63.58
	20%	52.06	-0.39	18.75	56.25
SVM	10%	63.22	33.13	87.92	58.94
	20%	61.2	23.29	65.0	60.47
RF	10%	49.16	-9.57	10.83	0.0
	20%	46.66	-15.31	46.43	0.0

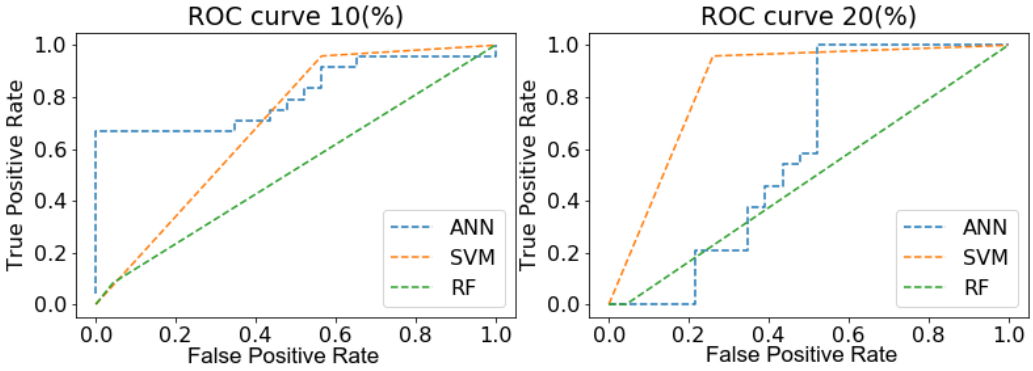

Figura 5.7: Curva ROC dos conjuntos de dados 10%, 20%, para remoção com *features* reduzidas.

sempenho dos classificadores fica bastante ruim, piorando para as sequências homólogas mais distantes.

5.2.2 H/ACA box

Nesta seção, discutimos os resultados dos experimentos para a classe dos H/ACA *box* snoRNAs.

Experimentos com métodos de AM

Da mesma forma que para os C/D/*box* snoRNAs, realizamos os experimentos com os algoritmos ANN, SVM e RF, para cada uma das mutações de bases - substituição, inserção e remoção, como descrito em seguida:

- **Substituição**

A Tabela 5.10 mostra os desempenhos dos três algoritmos de AM (SVM, ANN e RF) para a mutação de substituição. Como pode ser observado nessa tabela, os classificadores

ANN, SVM e RF apresentaram valores decrescentes para todas as métricas de avaliação. Com mutação de 10%, as ANN obtiveram $AUC = 74,20\%$, a SVM obteve $AUC = 67,76\%$ e a RF obteve $AUC = 62,03\%$. Com 50% de mutação, os resultados foram ainda piores, com ANN, SVM e RF obtendo $AUC = 59,08\%$, $AUC = 61,36\%$ e $AUC = 50,57\%$, respectivamente.

Tabela 5.10: Desempenhos dos algoritmos de AM para a mutação do tipo substituição, utilizando todas as *features* para predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	74.20	50.03	71.08	75.83
	20%	63.22	30.02	40.41	74.35
	30%	64.66	32.98	51.53	69.92
	40%	58.14	22.71	21.41	80.70
	50%	59.08	23.15	29.41	72.29
SVM	10%	67.76	36.37	55.12	73.76
	20%	66.20	33.29	54.9	70.99
	30%	64.33	29.23	65.63	63.99
	40%	61.34	24.17	70.53	59.60
	50%	61.36	22.78	54.85	63.06
RF	10%	62.03	28.86	27.51	88.75
	20%	52.75	2.38	17.43	59.37
	30%	60.64	21.59	29.47	78.29
	40%	52.42	7.19	6.81	77.64
	50%	50.57	2.92	4.52	57.24

As curvas ROC são mostradas na Figura 5.8.

Para investigar casos em que as características biológicas não são conhecidas, também testamos os algoritmos de AM com um número reduzido de *features* para predição de H/ACA *box* snoRNAs, neste caso - zscore, zscore, AC, GU, GC, LloopSC, RloopSC, LloopYC e RloopYC. A Tabela 5.11 mostra os resultados dos três algoritmos de AM.

Os três classificadores não obtiveram bons resultados para todos os conjuntos de dados. Todos os resultados da AUC foram inferiores a 70%, um desempenho abaixo do ideal. Além disso, a maioria dos conjuntos de dados mostrou AUC próxima de 50%, correspondendo a uma predição aleatória. As curvas ROC podem ser observadas na Figura 5.9.

- **Inserção**

A Tabela 5.12 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo inserção. Neste experimento, na árvore de mutação, sequências geradas foram reconhecidas como H/ACA *box* snoRNA apenas até 20%. Sequências com mais de 20%

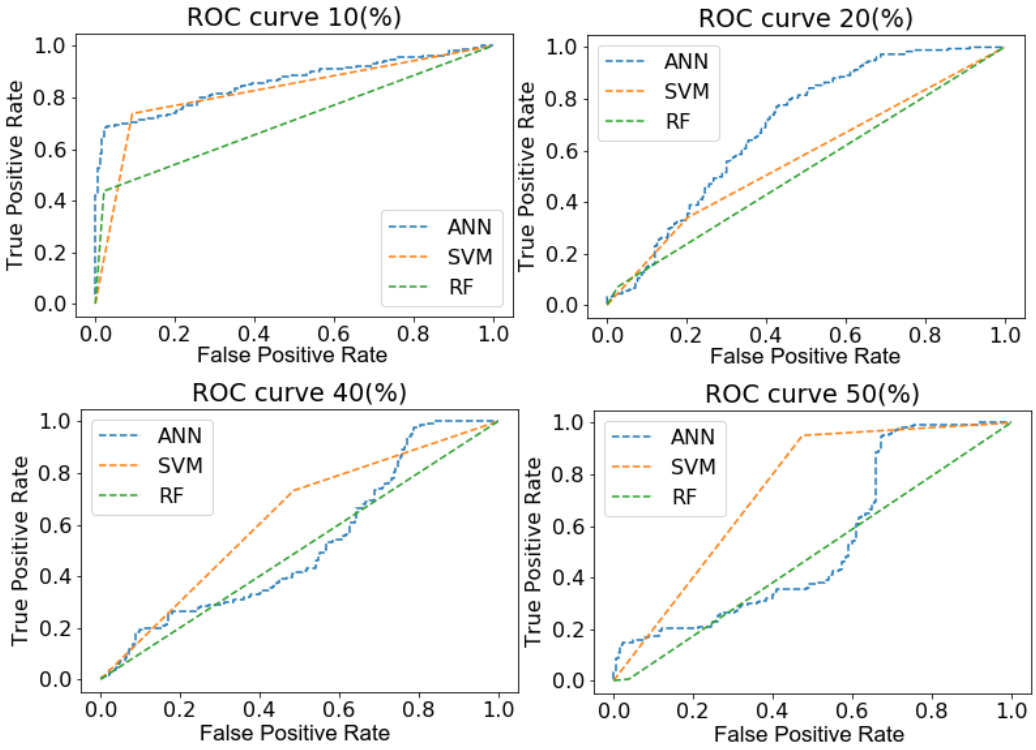

Figura 5.8: Curvas ROC dos conjuntos de dados com taxas de mutação de 10%, 20%, 40% e 50%, para a substituição, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs.

Tabela 5.11: Desempenhos dos algoritmos de AM para a substituição, com um número reduzido de *features*, para a predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	61.28	24.86	35.93	72.91
	20%	64.77	31.39	48.30	72.05
	30%	62.34	24.19	55.38	64.37
	40%	59.0	19.18	39.67	64.70
	50%	61.51	23.27	61.70	61.50
SVM	10%	62.78	26.95	59.29	63.77
	20%	61.71	24.75	51.92	64.59
	30%	57.59	14.53	48.24	59.36
	40%	53.83	6.46	34.56	56.26
	50%	58.63	18.38	60.93	58.28
RF	10%	57.52	17.34	26.48	69.86
	20%	62.93	25.55	39.95	73.96
	30%	60.38	21.17	37.09	69.44
	40%	53.40	6.43	20.0	60.26
	50%	57.98	15.05	43.74	61.18

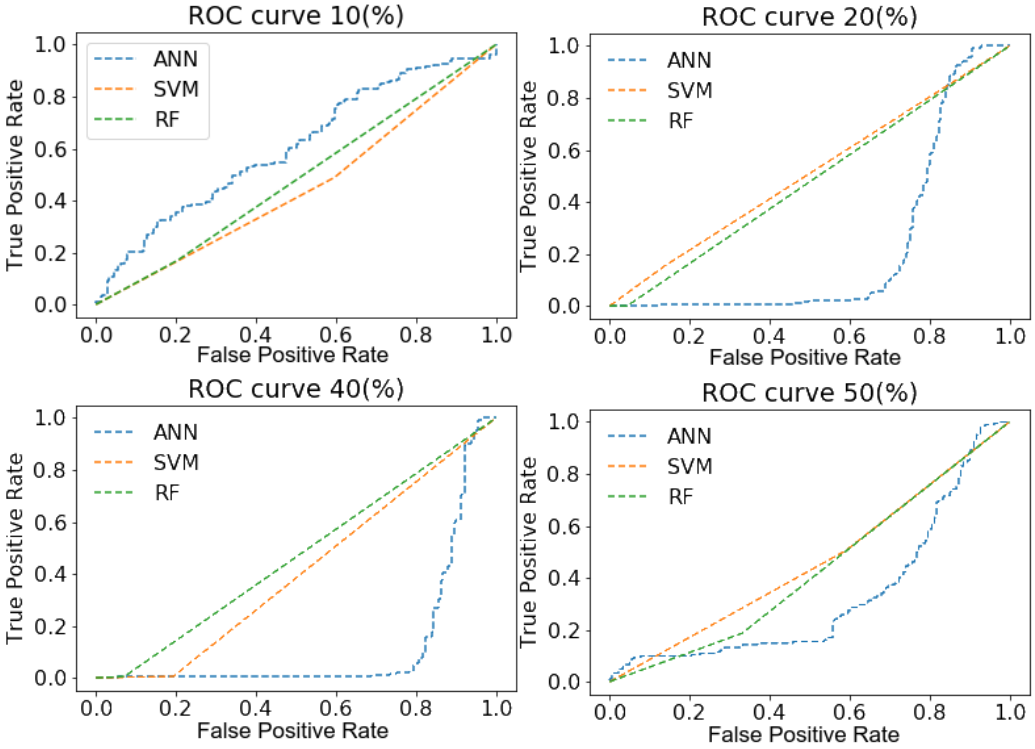

Figura 5.9: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo substituição, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs.

de mutação não foram mais reconhecidas como snoRNAs, pelo snoReport 2.0 [3]. Podemos confirmar que o tamanho da sequência de um snoRNA é uma característica importante para sua predição. Os três classificadores apresentaram melhor desempenho com 10% de mutação do que com 20% mutação, sendo os desempenhos decrescentes. No entanto, todas as métricas estão próximas ou abaixo de 70%, conforme a curva ROC mostrada na Figura 5.10. Apenas ANN e SVM mostraram AUC e *Recall* maiores que 70%, apenas com 10% de mutação. Os desempenhos dos três classificadores com 20% de mutação correspondem a um desempenho inferior ao ideal.

Os classificadores ANN e SVM apresentaram resultados muito semelhantes para AUC. Com 10% de mutação ANN $AUC = 71,92\%$ e SVM $AUC = 74,76\%$, e com 20% de mutação ANN $AUC = 59,30\%$ e SVM $AUC = 62,10\%$

Novamente, testamos os classificadores de AM com um número reduzido de *features* - zscore, AC, GU, GC, LloopSC, RloopSC, LloopYC e RloopYC. A Tabela 5.13 mostra que os resultados dos três algoritmos foram semelhantes aos resultados com todas as *features*.

Os três classificadores apresentaram melhor desempenho com 10% de mutação do que com 20% mutação, sendo os desempenhos decrescentes. Novamente, todas as métricas estão próximas ou abaixo de 70%. Os desempenhos dos três classificadores com 20% de

Tabela 5.12: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, utilizando todas as *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	71.92	47.90	59.31	79.3
	20%	59.30	19.80	45.05	63.02
SVM	10%	74.76	51.45	80.4	72.31
	20%	62.1	25.13	69.01	60.66
RF	10%	59.79	22.83	28.91	75.65
	20%	51.21	1.81	28.32	52.29

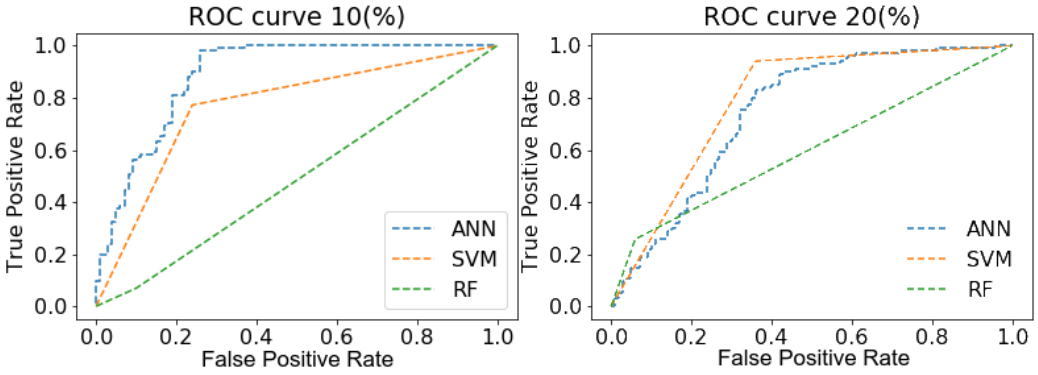

Figura 5.10: Curvas ROC dos conjuntos de dados com taxas de mutação de 10% e 20%, para a mutação do tipo inserção, utilizando todas as *features*, para a predição de H/ACA *box*.

mutação foram semelhantes, correspondem a um desempenho inferior ao ideal, para ANN $AUC = 52,02\%$, SVM $AUC = 51,60\%$ e RF $AUC = 52,25\%$. As curvas ROC podem ser observadas na Figura 5.11.

- **Remoção**

A Tabela 5.14 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo remoção. Neste experimento, a árvore de mutação gerou sequências reconhecidas como H/ACA *box* snoRNAs apenas até 10% de mutações. Sequências com mais de 10% de remoções não foram mais reconhecidas como snoRNAs pelo snoReport 2.0 [100]. ANN e SVM obtiveram *Recall* com valores superiores ao RF, $R = 94,72\%$, $R = 94,60\%$ e $R = 67,95\%$, respectivamente. Os três classificadores apresentaram valores de AUC acima de 70%, entretanto abaixo de 80%.

A curva ROC é mostrada na Figura 5.12.

Novamente, testamos com 10% de mutação os desempenhos dos algoritmos de AM com um número reduzido de *features* - zscore, AC, GU, GC, LloopSC, RloopSC, Llo-

Tabela 5.13: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, utilizando um número reduzido de *features*, para predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	72.62	47.31	73.87	72.12
	20%	52.02	1.80	32.35	53.39
SVM	10%	71.13	45.31	78.71	68.41
	20%	51.6	4.97	57.64	51.46
RF	10%	61.59	25.22	34.04	75.82
	20%	52.25	5.55	12.96	60.59

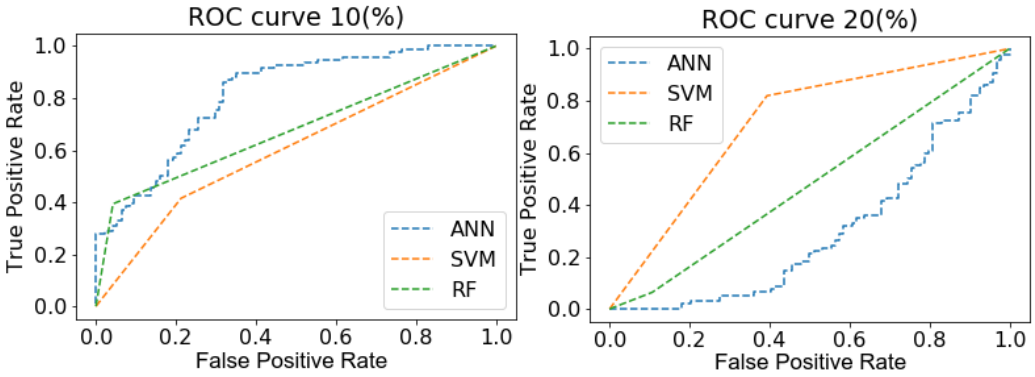

Figura 5.11: Curvas ROC dos conjuntos de dados com taxas de mutação de 10% e 20%, para a inserção, com um número reduzido de *features*, para a predição de H/ACA *box* snoRNAs.

opYC e RloopYC. Os três classificadores apresentaram um desempenho muito similar ao experimento com todas as *features*, como mostrado na Tabela 5.15. Os três classificadores também apresentaram valores de AUC acima de 70%, entretanto abaixo de 80%. Para ANN, SVM, RF temos $AUC = 76,80\%$, $AUC = 75,86\%$ e $AUC = 75,45\%$, respectivamente. A curva ROC é mostrada na Figura 5.13.

5.3 Segundo grupo de experimentos

Considerando que, no primeiro grupo de experimentos, as árvores de mutação do tipo substituição alcançaram níveis com maior porcentagem de mutação, para ambas as classes de snoRNAs, neste segundo grupo de experimentos, consideramos a mutação de substituição, tanto em bases quanto em codons.

Os *nós* das árvores de mutação geradas neste grupo de experimentos possuem 100 filhos, sendo que cada nível apresenta um máximo de $N = 5.000$ *nós* (sequências mutadas)

Tabela 5.14: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, utilizando todas *features*, para a predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	78.88	61.83	94.72	71.99
SVM	10%	77.38	59.53	94.6	70.42
RF	10%	71.77	43.75	67.95	73.6

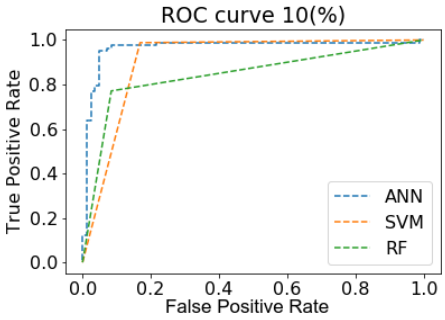

Figura 5.12: Curva ROC do conjunto de dados 10% , para remoção com todas as features.

Tabela 5.15: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), *Recall* (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	76.80	56.76	87.99	71.94
SVM	10%	75.86	55.81	91.30	69.82
RF	10%	75.45	52.23	73.4	76.60

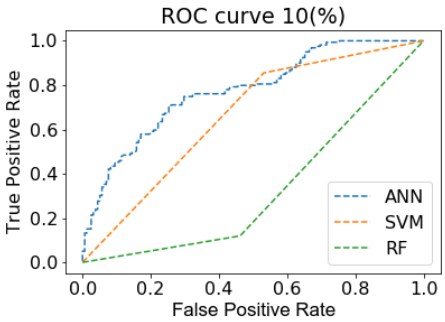

Figura 5.13: Curva ROC do conjunto de dados gerado para a taxa de mutação de 10%, para a mutação do tipo remoção, utilizando um número reduzido de *features*, para a predição de H/ACA *box* snoRNAs.

para C/D *box*, e $N = 5.000$ para H/ACA *box*. O foco é construir conjuntos de dados maiores, para verificar se há influência dos tamanhos dos conjuntos positivos e negativos para a predição de snoRNAs.

Para C/D *box*, as árvores geradas possuem no máximo 10.000 sequências por nível, então os conjuntos positivos e negativos foram construídos com 5.000 sequências, escolhidas aleatoriamente. Porém, o custo computacional de gerar as árvores com H/ACA *box*, considerando que essas sequências são maiores do que as de C/D *box*, nos levaram a escolher sequências para formar os conjuntos positivos e negativos mantendo $N = 5.000$, de uma forma diferente, como descrita a seguir. Para H/ACA *box*, as árvores geradas possuem no máximo 2.000 sequências por nível. Os conjuntos positivos e negativos foram construídos utilizando também sequências com uma diferença de ($\stackrel{+}{-}1$) relativamente à porcentagem da mutação. Por exemplo, os conjuntos com 10% de mutação contêm sequências com 9%, 10% e 11% de mutações. Então, para obter $N = 5.000$ sequências para os conjuntos positivos e negativos do H/ACA *box*, escolhemos aleatoriamente sequências com exatamente o número de mutações relativas aos conjuntos (10%, 20%, 30%, 40% e 50%) e acrescentamos outras (também escolhidas aleatoriamente) com ($\stackrel{+}{-}1$) de mutação com relação à porcentagem da mutação.

Experimento com Blast

De forma similar ao primeiro grupo de experimentos, para obter uma comparação mais justa do método Blastn com relação aos algoritmos de AM, utilizamos no Blastn como consulta as sequências de snoRNA biológico real (raiz da árvore) e como banco de dados cada um dos conjuntos de dados positivos, com taxas de mutação de 10%, 20%, 30%, 40% e 50% gerados a partir das árvores de mutação (substituição de bases, substituição com codons, inserção de bases e remoção de bases). Cada conjunto positivo é formado por $N = 5.000$ sequências para C/D *box* e H/ACA *box*.

Observamos que os resultados deste experimento são semelhantes aos resultados da Seção 5.2. Portanto, discutimos em detalhes apenas os resultados derivados de substituições. Os resultados mostraram baixas similaridades de sequência. Como mostra a Tabela 5.16 para C/D *box* apenas o conjunto com taxa de mutação de 10% encontrou *matches* e os conjuntos com taxas de mutação de 20%, 30%, 40% e 50%, nenhuma similarida foi encontrada, em nenhuma das comparações. Novamente para H/ACA *box* resultados mostraram baixas similaridades de sequência para o conjunto com taxa de mutação de 10% e 20% e nenhuma similarida para os outros conjuntos de dados.

A Tabela 5.17 mostra para C/D *box* e H/ACA *box*, o número de sequências dos conjuntos positivos (Figura 4.2 - Etapa 3), $n : (x)$ o número de instâncias com *features* extraídas

Tabela 5.16: Média de sequências que alcançaram alinhamento no Blastn.

snoRNA	Conjunto de dados	M
C/D	10%	27,13%
	10%	68,38%
H/ACA	20%	2,26%

dos conjuntos positivos ($x = 1$) e negativos ($x = 0$) (Figura 4.2 - Etapa 4) e o número total (soma) de instâncias das duas bases de dados (Figura 4.2 - Etapa 5).

Tabela 5.17: C/D *box* e H/ACA box

Árvore	C/D | H/ACA	n:(1)	n:(0)	Base de dados
substituição	10.000 | 6.000	5.000	5.000	10.000
substituição codons	10.000 | 6.000	5.000	5.000	10.000
inserção	10.000 | 6.000	5.000	5.000	10.000
remoção	10.000 | 6.000	5.000	5.000	10.000

5.3.1 C/D box

Nesta seção, discutimos os resultados dos experimentos para a classe dos C/D *box* snoRNAs.

Experimentos com métodos de AM

Assim como na Seção 5.2, realizamos experimentos com os algoritmos ANN, SVM e RF para mutações (substituição de bases, substituição com codons, inserção de bases e remoção de bases), considerando taxas de mutação de 10%, 20%, 30%, 40% e 50%. A seguir, discutimos os resultados obtidos para cada uma das mutações.

- **Substituição**

A Tabela 5.18 mostra os resultados dos três algoritmos de AM executados em sequências com mutações do tipo substituição, para a predição de C/D *box* snoRNAs. Podemos observar que os classificadores ANN e o SVM apresentaram bons desempenhos para conjuntos de dados com mutação de 10%, 20%, 30%, 40%, 50%, no entanto apresentaram valores decrescentes. Com 10% de mutação, ANN obteve $MCC = 99,48\%$ e SVM $MCC = 99,30\%$, enquanto que, com 50% de mutação, ANN e SVM obtiveram $MCC = 85,31\%$ e $MCC = 82,13\%$, respectivamente. Os modelos ANN e SVM alcançaram uma boa medida de separabilidade para todos os conjuntos de dados com mutações.

No entanto, para conjuntos de dados com 50% de mutação, eles mostraram menos capacidade de separação de classes. As curvas ROC são mostradas na Figura 5.14. RF obteve Precisão com valores decrescentes, entretando as outras métricas apresentaram os melhores resultados para 20% e 30%. Diferente do resultado dos outros classificadores ANN e SVM, para os conjunto de dados de mutação 50% os resultados não foram bons, observamos - predição com $MCC = 21,11\%$, e $AUC = 60,06\%$.

Tabela 5.18: Desempenhos dos algoritmos de AM para a mutação do tipo substituição, utilizando todas as *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	99.74	99.48	99.66	99.82
	20%	99.22	98.46	98.66	99.78
	30%	98.38	96.78	98.00	98.75
	40%	92.29	85.41	88.58	95.68
	50%	92.25	85.31	90.72	93.58
SVM	10%	99.65	99.30	99.66	99.64
	20%	97.75	95.59	96.86	98.62
	30%	97.05	94.12	97.64	96.50
	40%	91.41	83.19	95.20	88.49
	50%	90.74	82.13	95.24	87.38
RF	10%	89.40	78.20	79.28	99.40
	20%	99.26	98.54	98.60	99.92
	30%	91.32	83.95	83.12	99.40
	40%	84.51	69.36	70.61	97.81
	50%	60.06	21.11	24.82	84.08

A Figura 5.14 mostra as curvas ROC, utilizando todas as *features*.

Para investigar casos onde as características biológicas não são conhecidas, também testamos os algoritmos de AM com um número reduzido de *features*, neste caso - zscore, ls, Dcd, GC, lu5 e lu3. A Tabela 5.19 apresenta os resultados dos três algoritmos de AM.

Assim como para o experimento com $N = 3.000$ na Seção 5.2, utilizando todas as *features*, pode ser observado na Tabela 5.19, que os três classificadores não alcançaram bons resultados, para todos os conjuntos de dados (de 10% a 50%). RF apresentou o menor desempenho para todos os conjuntos de mutação.

Na Figura 5.15, as curvas ROC dos conjuntos de dados com taxas de mutação de 10%, 20%, 40% e 50% são mostradas. Observamos que esses modelos também não alcançaram uma boa capacidade preditiva de C/D *box*.

Como na Seção 5.2, com esses dois experimentos com a substituição, podemos perceber que o conjunto de *features* é bastante importante para predição do C/D *box*. Se não

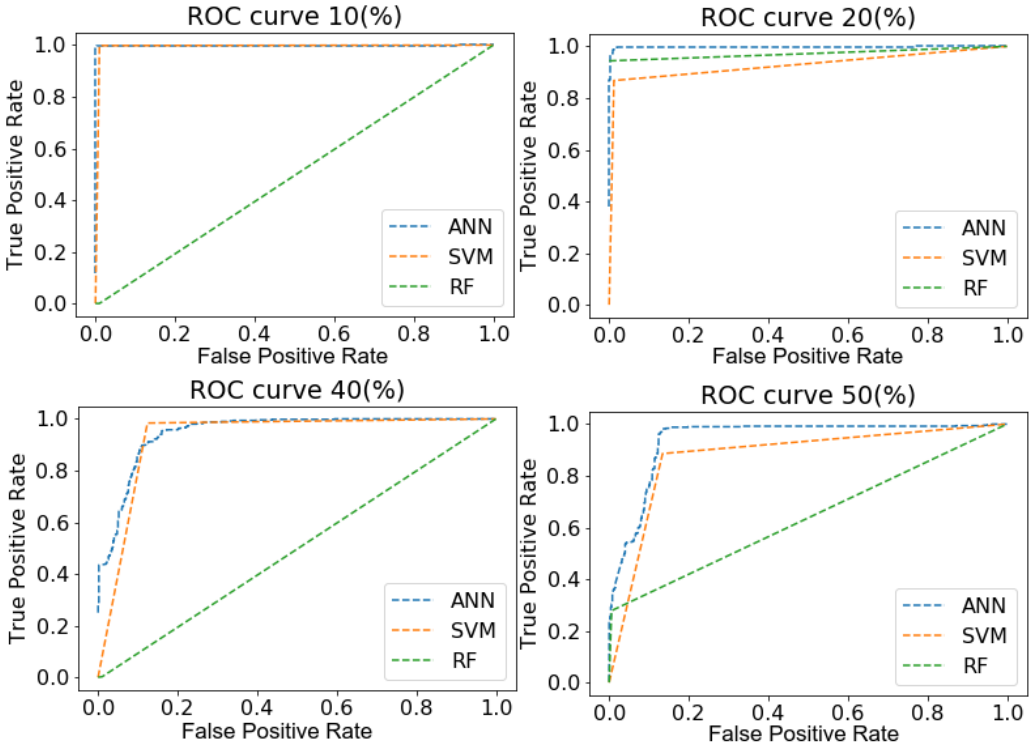

Figura 5.14: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo substituição, utilizando todas as *features*, para predição de C/D *box* snoRNAs.

Tabela 5.19: Desempenhos dos algoritmos de AM para a mutação do tipo substituição, com um número reduzido de *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	74.24	50.11	69.54	76.76
	20%	61.95	23.61	54.48	64.09
	30%	58.38	14.24	42.50	62.26
	40%	56.37	11.92	44.43	58.38
	50%	67.87	35.25	56.15	73.34
SVM	10%	73.26	51.78	91.63	67.03
	20%	65.86	37.83	89.63	60.79
	30%	64.94	35.41	85.38	60.63
	40%	70.54	44.56	82.94	66.46
	50%	73.89	48.90	71.60	75.03
RF	10%	57.47	9.37	25.76	70.42
	20%	53.30	3.57	19.51	60.24
	30%	51.27	-1.55	19.38	53.46
	40%	59.33	18.26	32.84	69.86
	50%	59.55	19.02	29.36	74.03

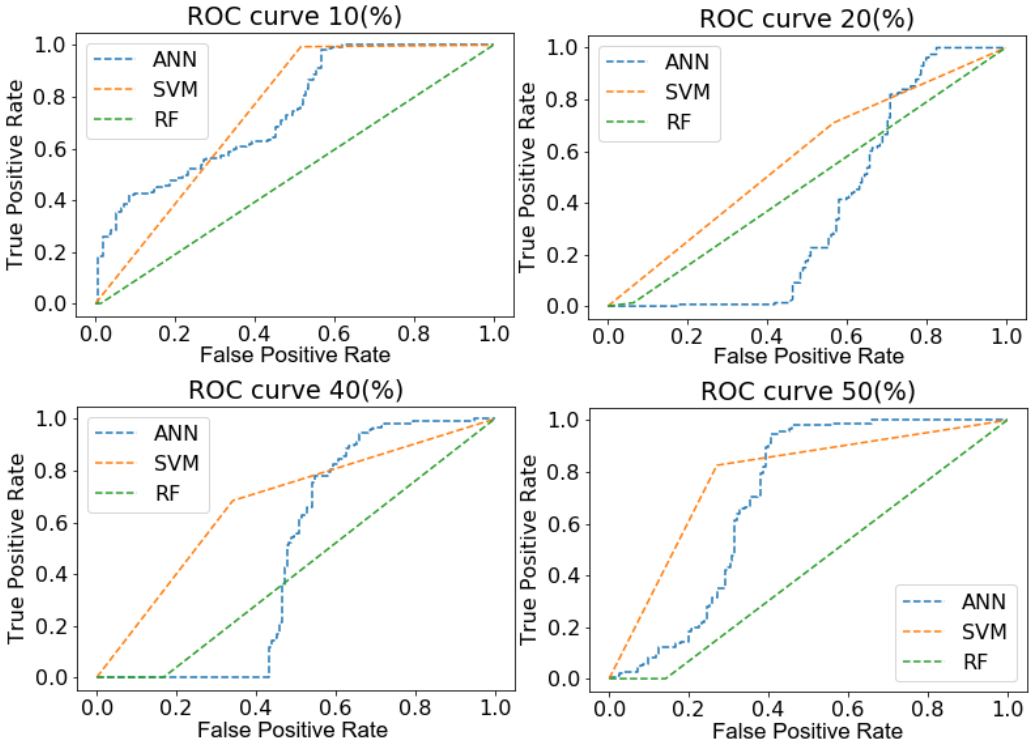

Figura 5.15: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo substituição, com um número reduzido de *features*, para a predição de H/ACA *box* snoRNAs.

são conhecidas características biológicas relevantes, o desempenho dos classificadores fica bastante ruim, piorando para as sequências homólogas mais distantes.

- **Substituição por codons**

Neste experimento, utilizamos a substituição por codons, com a matriz de substituição de aminoácidos descrita na Seção 2.2.2. Como a substituição foi realizada em três bases nitrogenadas (por códon), a árvore de mutação gerou sequências C/D *box* snoRNAs apenas até 10% de mutações.

A Tabela 5.20 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com códons. Podemos observar que a predição para snoRNAs com 10% de mutações com códons obtiveram valores altos de AUC, de 99, 09%, 98, 38% e 97, 96% para ANN, SVM e RF, respectivamente, e decrescentes com relação a 20% de mutação AUC, de 86, 62%, 84, 60% e 79, 81% para ANN, SVM e RF, respectivamente.

Tabela 5.20: Desempenhos dos algoritmos de AM para a mutação do tipo substituição por codons, utilizando todas as *features* para a predição de C/D *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	99.09	98.20	98.44	99.74
	20%	86.62	75.62	75.76	96.78
SVM	10%	98.38	96.78	98.60	98.17
	20%	84.60	71.11	74.91	92.93
RF	10%	97.96	95.98	98.76	97.21
	20%	79.81	62.62	93.88	73.27

As curvas ROC são mostradas na Figura 5.16.

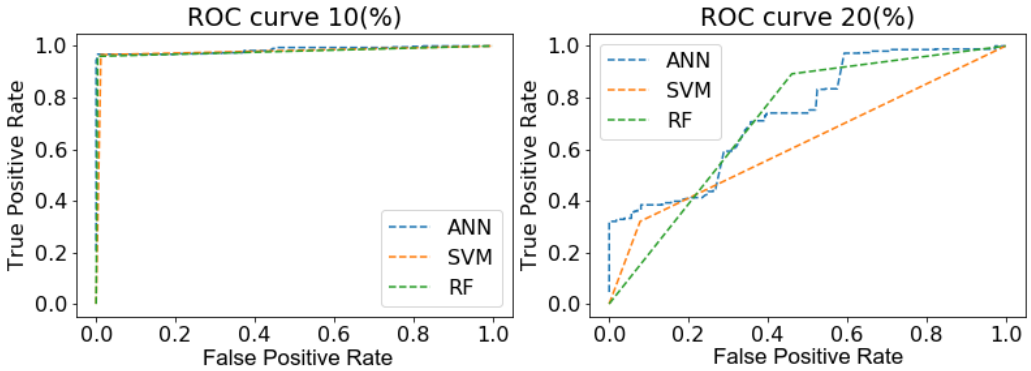

Figura 5.16: Curvas ROC dos conjuntos de dados com 10%, 20%, relativas à mutação do tipo substituição com codons, utilizando todas as *features*, para predição de C/D *box* snoRNAs.

A Tabela 5.21 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com códons, para um número reduzido de *features*. Podemos observar que a predição para snoRNAs com 10% de mutações com códons obteve valores ruins de AUC, de 62.93%, 79.70% e 54.62% para ANN, SVM e RF, respectivamente, o mesmo ocorrendo para 20% de mutação, de 65.77%, 73.46% e 56.63% para ANN, SVM e RF, respectivamente.

Tabela 5.21: Desempenhos dos algoritmos de AM para a mutação do tipo substituição com codons, com um número reduzido de *features* para a predição de C/D *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	62.93	26.32	39.43	74.44
	20%	65.77	32.33	53.77	70.77
SVM	10%	79.70	61.25	86.46	76.15
	20%	73.46	49.12	83.59	69.50
RF	10%	54.62	8.68	37.43	57.03
	20%	56.63	13.37	47.04	58.22

As curvas ROC são mostradas na Figura 5.17.

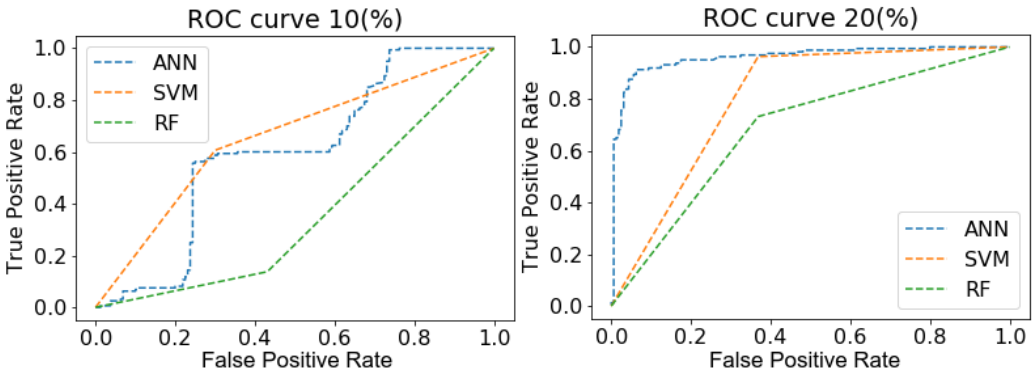

Figura 5.17: Curvas ROC dos conjuntos de dados com 10%, 20%, relativas à mutação do tipo substituição por codons, com um número reduzido de *features*, para predição de C/D *box* snoRNAs.

- **Inserção**

A Tabela 5.22 mostra os resultados dos três algoritmos de AM para a mutação do tipo inserção. Os classificadores ANN e SVM apresentaram AUC acima de 90% para todos os conjuntos de dados. O classificador ANN obteve $AUC = 99,87\%$ para 10% e $AUC = 90,99\%$ para 50%, como mostrado pelas curvas ROC da Figura 5.18. O

classificador SVM também apresentou uma boa predição em todos os conjuntos de dados - com 10% de mutação, $AUC = 99,52\%$ e para 50% de mutações, $AUC = 93,60\%$. As curvas ROC são mostradas na Figura 5.18. O modelo RF alcançou AUC acima de 90% apenas para o conjunto de dados com 30% de mutação, como mostra a Tabela 5.22.

Tabela 5.22: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, utilizando todas as *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	99.87	99.75	99.74	100.0
	20%	99.85	99.70	99.74	99.95
	30%	99.23	98.51	98.47	100.0
	40%	96.34	93.54	93.42	99.21
	50%	90.99	83.22	86.72	94.79
SVM	10%	99.52	99.06	99.14	99.91
	20%	99.87	99.74	99.77	99.98
	30%	99.78	99.56	99.60	99.95
	40%	95.78	92.39	92.72	98.76
	50%	93.60	87.72	89.47	97.52
RF	10%	65.80	34.51	31.67	99.78
	20%	60.56	24.23	22.11	95.67
	30%	97.55	95.23	95.28	99.81
	40%	60.43	14.19	33.78	72.36
	50%	84.01	68.03	77.40	89.18

A Tabela 5.23 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com inserção, para um número reduzido de *features*. Assim como para o experimento com $N = 3.000$ na Seção 5.2, podemos observar que o desempenho dos três classificadores foi menor quando comparado ao experimento com todas as *features*.

As curvas ROC são mostradas na Figura 5.19.

- **Remoção**

A Tabela 5.24 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo remoção. Neste experimento, a árvore de mutação gerou sequências reconhecidas como C/D *box* snoRNAs até 40%. Sequências com mais de 40% de mutação do tipo remoção não foram mais reconhecidas como snoRNAs pelo snoReport 2.0 [100]. Assim como na Seção 5.2, podemos observar que o tamanho da sequência de um snoRNA é uma característica relevante para sua predição. Ainda, os três classificadores obtiveram o melhor desempenho para 10% de mutação e o pior desempenho para 40% de mutação. Com 10% de mutação, ANN, SVM e RF apresentaram $AUC = 97,32\%$, $AUC = 96,89\%$

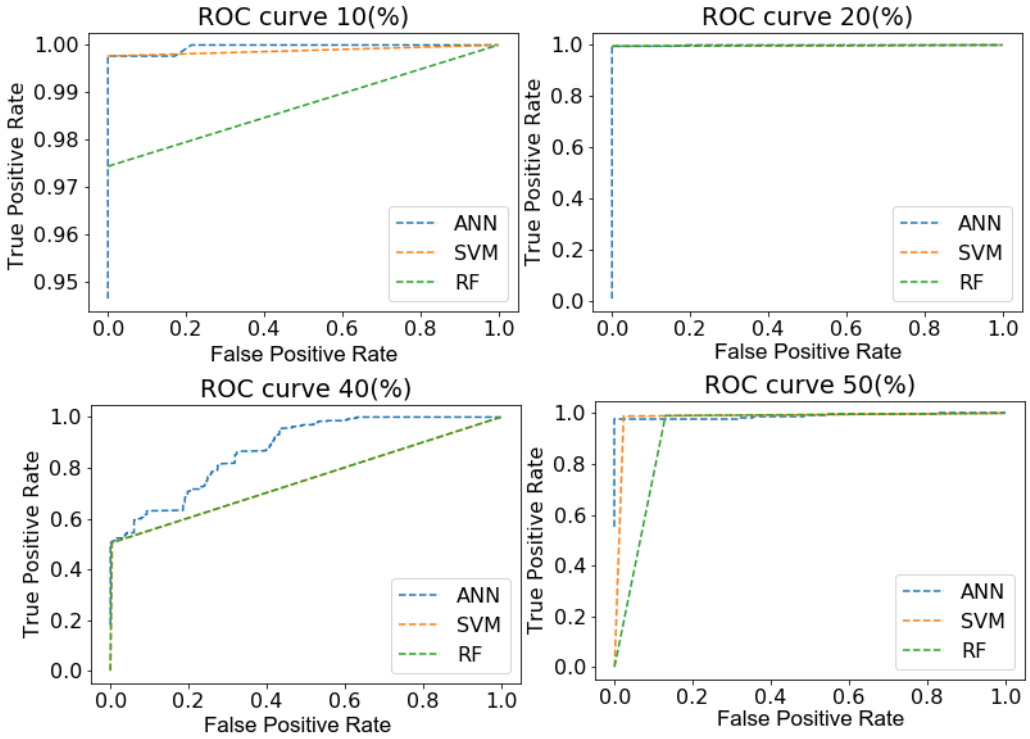

Figura 5.18: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo inserção, utilizando todas as *features*, para predição de C/D *box* snoRNAs.

Tabela 5.23: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, com número reduziddo de *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	78.23	59.29	59.31	95.44
	20%	65.89	35.78	34.10	93.60
	30%	76.87	58.58	53.97	99.58
	40%	52.78	1.97	7.84	77.48
	50%	82.05	63.70	76.87	85.77
SVM	10%	95.05	90.39	97.09	93.28
	20%	91.29	83.10	94.54	88.78
	30%	89.75	80.05	91.66	88.29
	40%	74.24	53.6	51.14	95.06
	50%	84.03	70.43	72.56	94.18
RF	10%	62.44	19.20	27.46	91.49
	20%	61.24	17.13	25.53	89.31
	30%	89.92	79.99	84.34	94.95
	40%	78.66	58.67	62.94	91.81
	50%	76.80	52.28	65.87	84.29

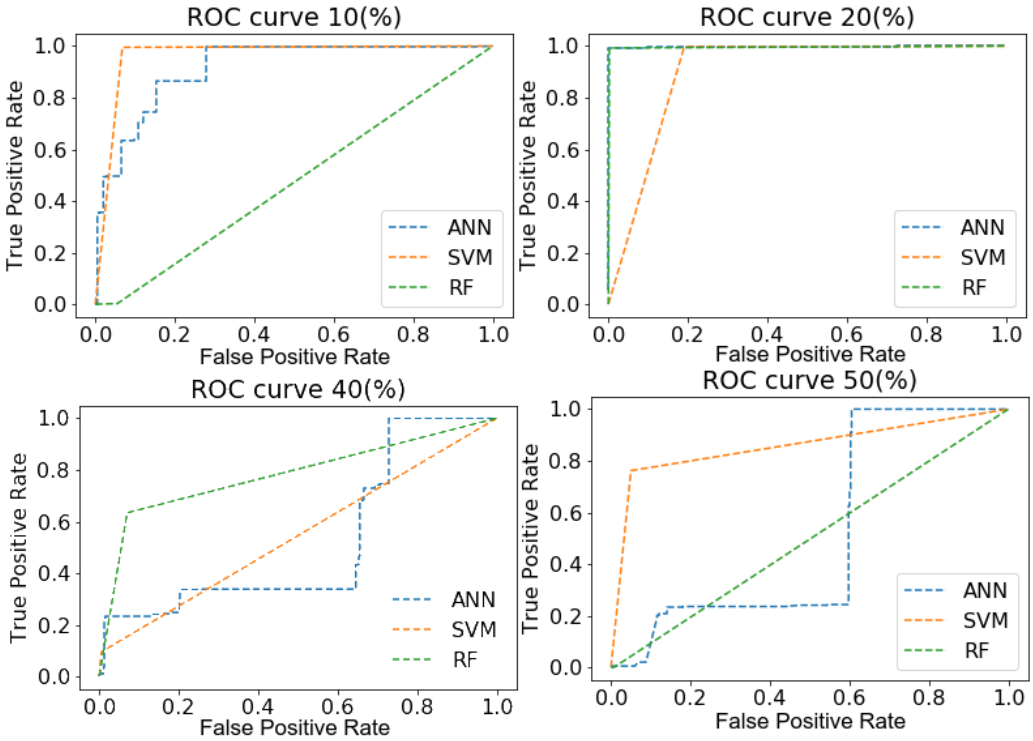

Figura 5.19: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo inserção, com um número reduzido de *features*, para predição de C/D *box* snoRNAs.

e $AUC = 95,48\%$, respectivamente. No entanto, com conjunto de 40% de mutação, ANN, SVM e RF apresentaram $AUC = 61,21\%$, $AUC = 76,26\%$ e $AUC = 63,03\%$, respectivamente.

As curvas ROC são mostradas na Figura 5.20.

Além disso, testamos os desempenhos dos algoritmos de AM com um número reduzido de *features* para a remoção, neste caso - zscore, ls, Dcd, GC, lu5 e lu3. A Tabela 5.25 apresenta os resultados, nos quais os três classificadores obtiveram resultados piores do que os resultados do experimento com todas as *features*, para todos os conjuntos de dados (de 10% a 40%).

As curvas ROC são mostradas na Figura 5.21.

Assim como na substituição e na inserção, os dois experimentos com a remoção mostraram que o conjunto de *features* é bastante importante para predição do C/D *box* por parte de classificadores de AM. Se características biológicas relevantes não são conhecidas, o desempenho dos classificadores fica bastante ruim, piorando para as sequências homólogas mais distantes.

Tabela 5.24: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, utilizando todas as *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	97.32	94.93	95.98	98.62
	20%	93.86	88.35	89.74	97.80
	30%	83.42	69.93	93.28	77.92
	40%	61.21	29.80	24.62	91.87
SVM	10%	96.89	93.88	99.06	94.94
	20%	90.09	80.81	90.12	90.07
	30%	94.08	88.64	91.06	96.91
	40%	76.26	56.98	64.31	84.52
RF	10%	95.48	91.65	91.08	99.87
	20%	68.49	37.50	40.79	91.48
	30%	90.28	81.27	92.58	88.51
	40%	63.03	33.35	32.75	83.02

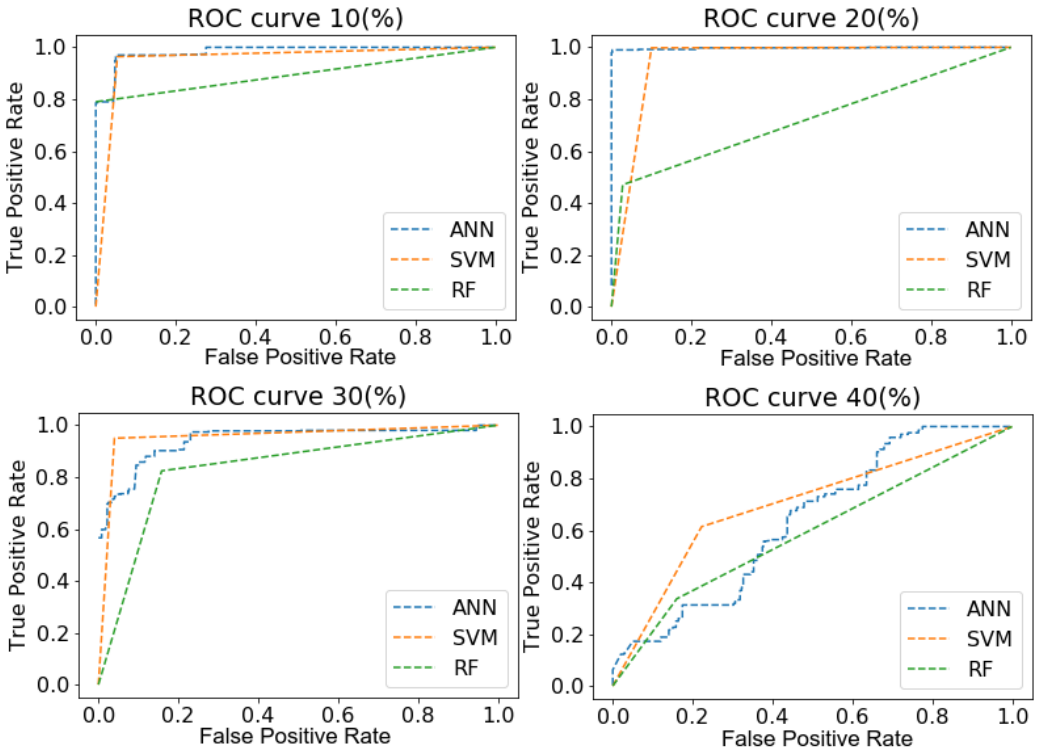

Figura 5.20: Curvas ROC dos conjuntos de dados com 10%, 20%, 30% e 40%, relativas à mutação do tipo remoção, utilizando todas as *features*, para predição de C/D *box* snoRNAs.

Tabela 5.25: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, com número reduziddo de *features*, para predição de C/D *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	96.55	93.26	98.62	94.70
	20%	90.88	83.30	82.92	98.62
	30%	53.69	8.45	46.97	54.27
	40%	60.35	19.71	42.97	65.86
SVM	10%	93.62	87.87	99.56	88.99
	20%	90.49	81.54	94.40	87.56
	30%	59.05	18.90	65.27	58.06
	40%	66.03	39.21	89.74	60.88
RF	10%	54.08	1.87	9.80	85.66
	20%	66.91	27.86	37.93	90.25
	30%	54.88	11.49	31.61	59.15
	40%	63.03	26.44	54.63	65.66

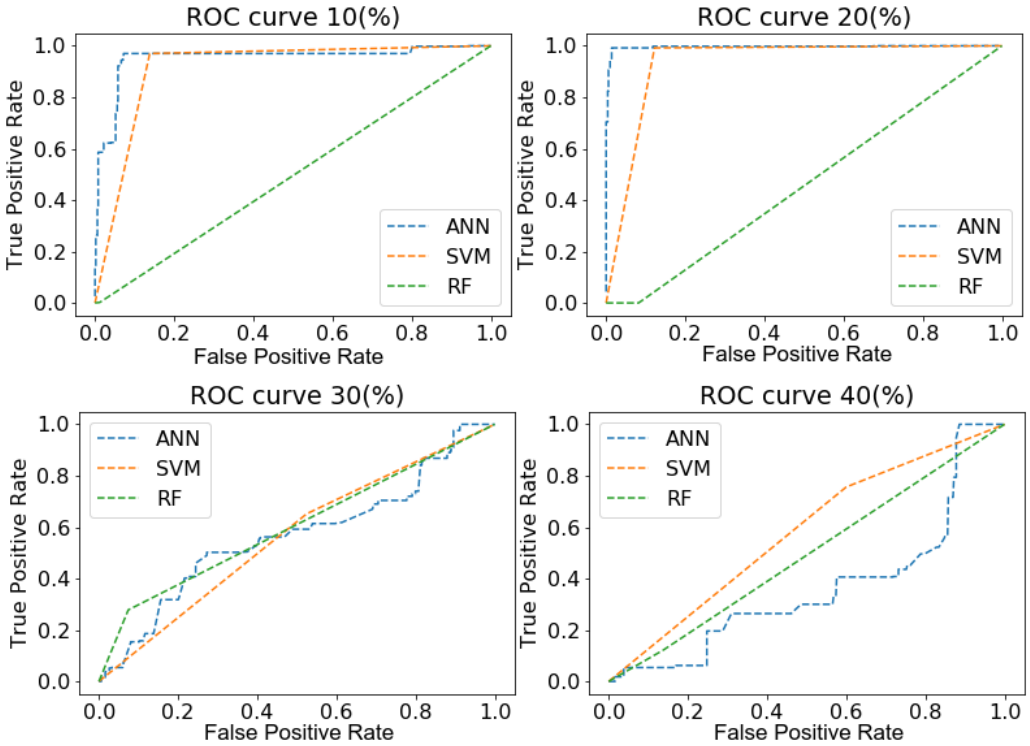

Figura 5.21: Curvas ROC dos conjuntos de dados com 10%, 20%, 30% e 40%, relativas à mutação do tipo remoção, com um número reduzido de *features*, para predição de C/D *box* snoRNAs.

5.3.2 H/ACA box

Nesta seção, discutimos os resultados dos experimentos para a classe dos H/ACA *box* snoRNAs.

Experimentos com métodos de AM

- **Substituição**

A Tabela 5.26 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com substituições, utilizando todas as *features*. Pode-se observar que as predições, das taxas de mutação de 10% até as de 50% tiveram valores ruins de AUC, abaixo de 70%, para os três métodos - ANN, SVM e RF.

Tabela 5.26: Desempenhos dos algoritmos de AM para a mutação do tipo substituição, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	68.67	39.79	55.32	75.47
	20%	62.89	29.51	45.67	69.67
	30%	61.35	27.77	66.29	60.35
	40%	59.77	22.71	36.64	68.20
	50%	59.34	23.41	34.61	68.51
SVM	10%	66.19	33.09	59.80	68.57
	20%	66.80	35.07	57.09	70.85
	30%	62.74	28.74	76.92	59.94
	40%	62.83	28.48	57.96	64.21
	50%	64.01	28.83	52.65	68.14
RF	10%	55.33	14.17	14.50	79.05
	20%	54.99	12.52	13.79	78.30
	30%	59.33	19.00	30.66	71.87
	40%	51.04	1.04	4.53	65.03
	50%	57.15	14.46	21.41	75.07

As curvas ROC são mostradas na Figura 5.22.

A Tabela 5.27 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com substituições, com número reduzido de *features*. Pode-se observar que as predições, das taxas de mutação de 10% até as de 50% também tiveram valores ruins de AUC, abaixo de 65%, para os três métodos - ANN, SVM e RF.

As curvas ROC são mostradas na Figura 5.23.

- **Substituição por codons**

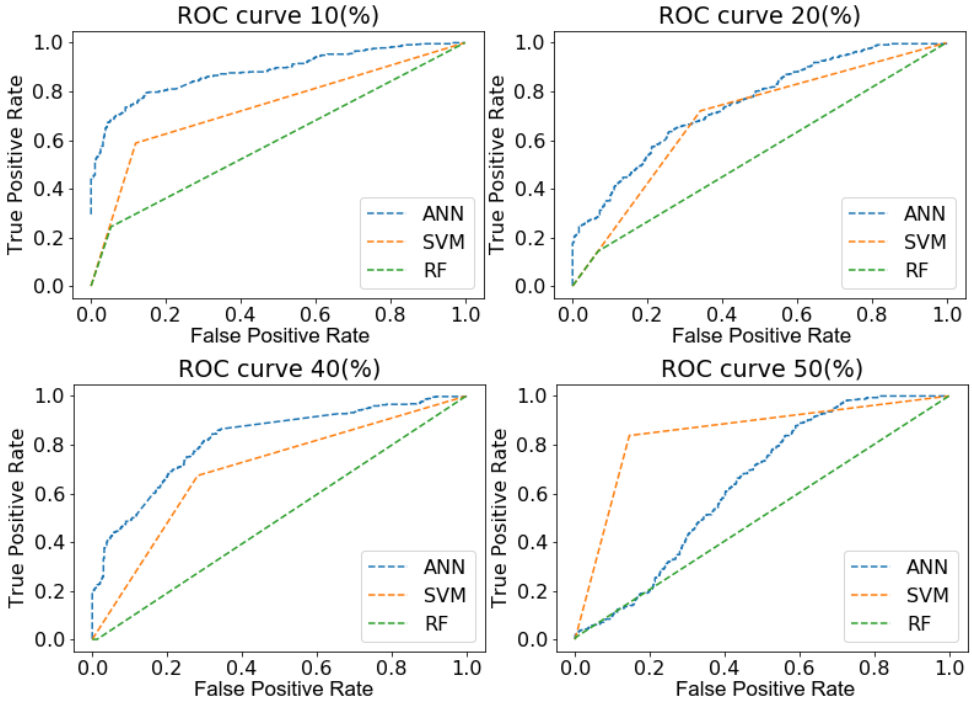

Figura 5.22: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo substituição, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs.

Neste experimento, utilizamos a substituição por codons, com a matriz de substituição de aminoácidos descrita na Seção 2.2.2. Como a substituição foi realizada em três bases nitrogenadas (por códon), a árvore de mutação gerou sequências H/ACA *box* snoRNAs apenas até 10% de mutações.

A Tabela 5.28 mostra os resultados dos três métodos de AM, executados com sequências mutadas com substituições por códons. Os classificadores ANN e SVM apresentaram AUC acima de 75%, no entanto novamente o RF apresentou um desempenho menor, com $AUC = 57,68\%$.

As curvas ROC são mostradas na Figura 5.24.

A Tabela 5.29 mostra os resultados dos três algoritmos de AM executados com sequências mutadas com substituições por códons, para um número reduzido de *features*. Neste experimento, ANN e SVM obtiveram um desempenho menor, quando comparado com os obtidos com todas as *features*, $AUC = 67,85\%$ e $AUC = 75,64\%$, respectivamente. O classificador RF obteve uma melhora no desempenho, quando comparado com os resultados com todas as *features*, no entanto, obteve um desempenho abaixo do ideal, com $AUC = 65,54\%$.

As curvas ROC são mostradas na Figura 5.25.

- **Inserção**

Tabela 5.27: Desempenhos dos algoritmos de AM para a mutação do tipo substituição, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Bases de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	61.66	26.13	34.47	75.58
	20%	64.57	30.15	49.14	71.11
	30%	62.50	23.73	55.81	64.45
	40%	54.44	9.81	46.34	55.33
	50%	61.19	23.02	61.43	61.16
SVM	10%	60.43	22.37	47.10	64.24
	20%	61.77	25.71	48.53	66.04
	30%	57.95	15.66	54.46	58.58
	40%	54.10	8.93	53.25	54.19
	50%	58.47	17.42	61.91	57.94
RF	10%	58.59	19.52	26.33,	74.24
	20%	55.85	12.10	24.05	66.11
	30%	58.42	15.84	31.08	68.58
	40%	54.34	8.92	31.02	58.16
	50%	53.63	7.56	35.25	55.75

A Tabela 5.30 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo inserção. Assim como no experimento com inserção para H/ACA na Seção 5.2, na árvore de mutação, sequências geradas foram reconhecidas como H/ACA *box* snoRNA apenas até 20%. Sequências com mais de 20% de mutação não foram mais reconhecidas como snoRNAs, pelo snoReport 2.0 [3]. Podemos confirmar novamente que o tamanho da sequência de um snoRNA é uma característica importante para sua predição. Os três classificadores também apresentaram melhor desempenho com 10% de mutação do que com 20% mutação, sendo os desempenhos decrescentes. ANN e SVM mostraram AUC e *Recall* maiores que 75%, apenas com 10% de mutação. Os desempenhos dos três

Tabela 5.28: Desempenhos dos algoritmos de AM para a mutação do tipo substituição por codons, utilizando todas as *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	79.09	59.90	73.89	82.49
SVM	10%	77.60	55.81	73.81	79.87
RF	10%	57.68	14.87	37.64	62.85

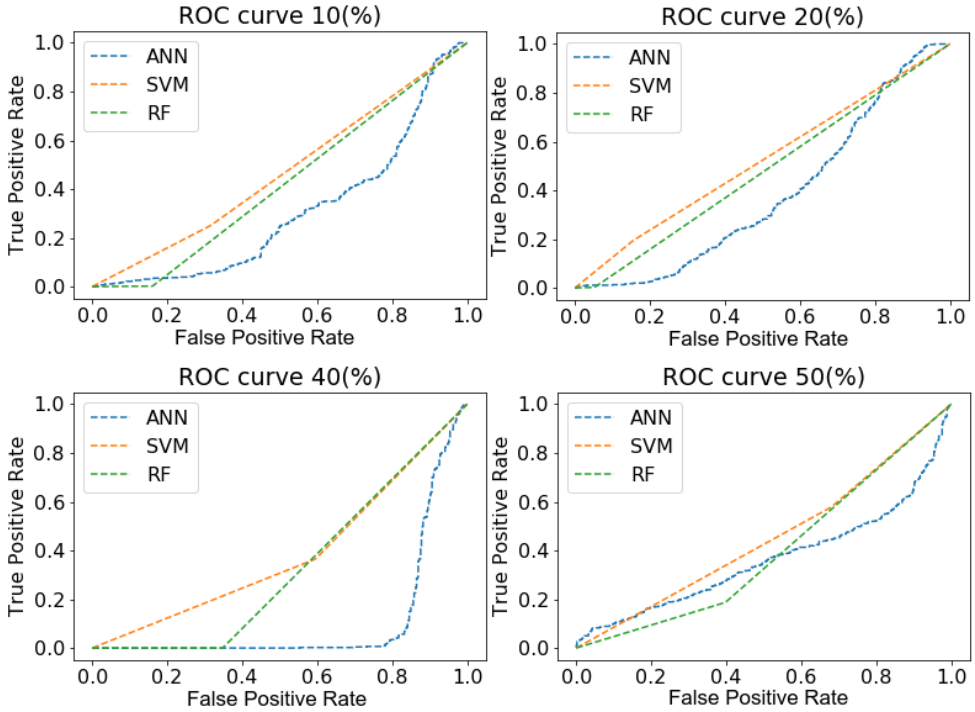

Figura 5.23: Curvas ROC dos conjuntos de dados com 10%, 20%, 40% e 50%, relativas à mutação do tipo substituição, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs.

classificadores com 20% de mutação correspondem a um desempenho inferior ao ideal.

As curvas ROC são mostradas na Figura 5.26.

Novamente, testamos os desempenhos dos algoritmos de AM com um número reduzido de *features* para a inserção. A Tabela 5.31 apresenta os resultados, nos quais os classificadores ANN, SVM e RF apresentaram resultados muito semelhantes para AUC, sendo os desempenhos decrescentes. Com 10% de mutação ANN $AUC = 76,31\%$, SVM $AUC = 71,35\%$ e RF $AUC = 72,14\%$, e com 20% de mutação, os três classificadores apresentaram AUC na faixa de 51% a 58%.

As curvas ROC são mostradas na Figura 5.27.

Tabela 5.29: Desempenhos dos algoritmos de AM para a mutação do tipo substituição por codons, número reduzido de *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	67.85	37.48	57.70	72.42
SVM	10%	75.64	53.84	84.45	71.84
RF	10%	65.54	31.28	68.46	64.73

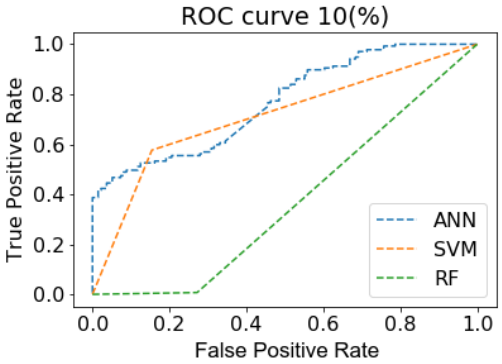

Figura 5.24: A curva ROC do conjunto de dados com 10% de mutações do tipo substituição com codons, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs.

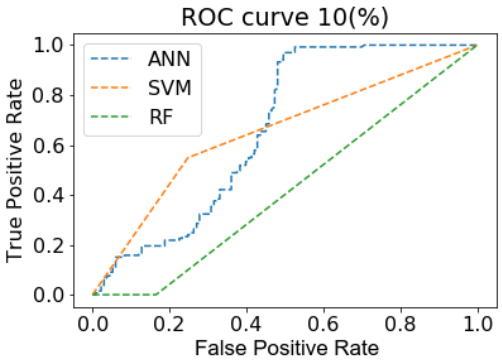

Figura 5.25: A curva ROC do conjunto de dados com 10% de mutação de substituição por códons, com número reduzido de *features*, para predição de H/ACA *box* snoRNAs.

Tabela 5.30: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, utilizando todas as *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	75.54	53.30	75.22	75.71
	20%	58.47	18.94	54.43	59.21
SVM	10%	78.02	58.19	88.50	73.18
	20%	56.87	15.32	64.79	55.94
RF	10%	69.48	41.65	49.33	82.65
	20%	62.72	26.32	56.25	64.62

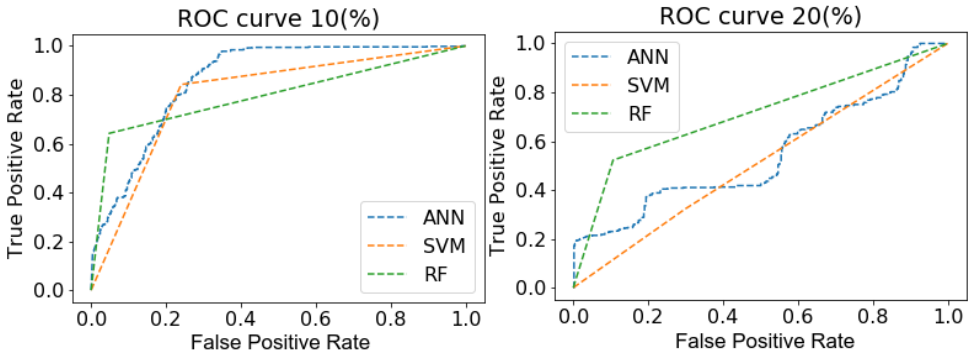

Figura 5.26: Curvas ROC dos conjuntos de dados com 10% e 20%, relativas à mutação do tipo inserção, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs.

Tabela 5.31: Desempenhos dos algoritmos de AM para a mutação do tipo inserção, com número reduzido de *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	76.31	52.53	67.79	81.71
	20%	58.23	17.47	29.91	69.0
SVM	10%	71.35	44.52	70.74	71.64
	20%	51.76	4.48	53.14	51.73
RF	10%	72.14	46.76	56.37	82.33
	20%	56.59	14.72	26.10	66.91

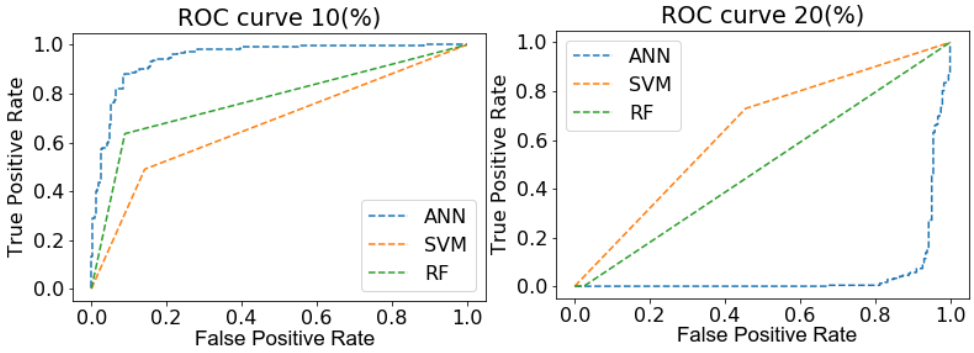

Figura 5.27: Curvas ROC dos conjuntos de dados com 10% e 20%, relativas à mutação do tipo inserção, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs.

- **Remoção**

A Tabela 5.32 mostra os desempenhos dos três algoritmos de AM para a mutação do tipo remoção. Assim como na Seção 5.2, neste experimento, a árvore de mutação gerou sequências reconhecidas como H/ACA *box* snoRNAs apenas até 10% de mutações.

Sequências com mais de 10% de remoções não foram mais reconhecidas como snoRNAs pelo snoReport 2.0 [100]. ANN e SVM obtiveram em todas as métricas valores superiores ao RF. No entanto, os desempenhos deste experimento foram abaixo dos resultados obtidos no experimento de remoção com H/ACA *box* na Seção 5.2. Apenas ANN obteve *AUC* próxima de 70%, enquanto os outros classificadores obtiveram *AUC* abaixo de 70%.

Tabela 5.32: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, utilizando todas as *features*, para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	70.19	45.92	92.90	63.92
SVM	10%	66.64	38.12	86.94	61.87
RF	10%	52.76	4.32	41.84	53.54

A curva ROC é mostrada na Figura 5.28.

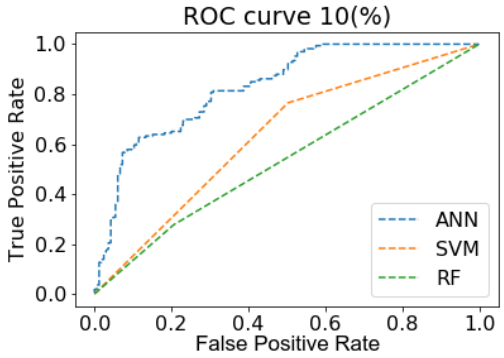

Figura 5.28: A curva ROC do conjunto de dados com 10% de mutação do tipo remoção, utilizando todas as *features*, para predição de H/ACA *box* snoRNAs.

Por fim, testamos os desempenhos dos algoritmos de AM com um número reduzido de *features* para a remoção. A Tabela 5.33 apresenta os resultados, nos quais os três classificadores obtiveram resultados ainda piores que os resultados do experimento com todas as *features*.

A curva ROC é mostrada na Figura 5.29.

Tabela 5.33: Desempenhos dos algoritmos de AM para a mutação do tipo remoção, com número reduzido de *features* para a predição de H/ACA *box* - Área Sob a Curva (AUC), Coeficiente de Correlação de Matthews (MCC), Recall (R) e Precisão (P).

AM	Base de dados	AUC(%)	MCC(%)	R(%)	P(%)
ANN	10%	63.45	28.88	76.68	60.67
SVM	10%	62.98	27.38	69.39	61.54
RF	10%	50.48	-1.03	38.21	50.67

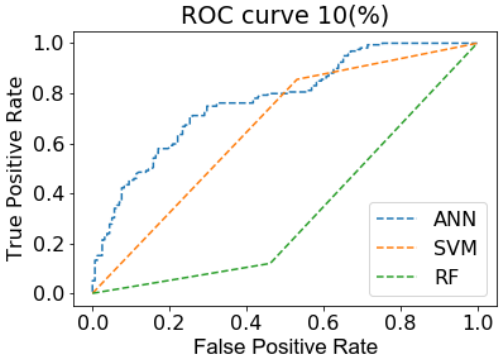

Figura 5.29: A curva ROC do conjunto de dados com 10% de mutação do tipo remoção, com um número reduzido de *features*, para predição de H/ACA *box* snoRNAs.

Capítulo 6

Conclusões

Nesta dissertação, realizamos uma avaliação sistemática de métodos de AM para predição de ncRNAs, do tipo snoRNAs.

Para isso, inicialmente desenvolvemos um método para construção de conjuntos de snoRNAs artificiais, cobrindo diferentes distâncias evolutivas, de forma cuidadosa e controlada. Esses conjuntos de dados foram extraídos de árvores de mutação, contendo na sua raiz uma sequência de ambas as classes de snoRNAs (C/D *box* ou H/ACA box) e nos seus nós internos, sequências de snoRNAs artificiais, construídas *in silico*, a partir de mutações pontuais em bases nitrogenadas (substituição, inserção e remoção).

As sequências de snoRNAs artificiais geradas nessas árvores foram extraídas aleatoriamente, para construir os conjuntos de dados com 10%, 20%, 30%, 40% e 50% de mutações. Assim, foi possível obter conjuntos arbitrariamente grandes, diversos e próximos de snoRNAs reais.

Para testar os métodos de AM, extraímos *features* conhecidas e relevantes das sequências armazenadas nos conjuntos positivos e negativos, que por sua vez foram construídos de forma balanceada. Os classificadores testados de AM foram SVM, RF e ANN, escolhidos pois vêm sendo utilizados para predição de ncRNAs. Assim, esses métodos foram avaliados quanto as suas capacidades de predizer snoRNAs, utilizando conjuntos de dados grandes, diversos e com snoRNAs artificiais construídos a partir das árvores de mutação descritas anteriormente.

Mesmo com limitações de tempo e espaços computacionais para gerar árvores de mutações com um grande número de filhos e níveis, observamos que os métodos de AM prediziam melhor snoRNAs do que os métodos de inferência de homologia baseados em sequência primária, como o Blast. Como esperado, os resultados com Blast mostraram muitos falsos negativos.

Realizamos dois grupos de experimentos, com base nos tamanhos dos conjuntos de sequências extraídas das árvores de mutação. O primeiro grupo continha conjuntos con-

tendo 3.000 sequências de C/D *box* e 2.000 sequências de H/ACA *box*. Para C/D *box*, observamos que, para um número grande de *features* biológicas importantes, quanto maior a porcentagem de mutações, pior o desempenho dos classificadores de AM. Para a mutação de substituição, os classificadores SVM e ANN obtiveram excelente desempenho para conjuntos de dados com mutação de 10%, 20%, 30% e 40%. Mas para conjuntos de dados com 50% de mutações, os classificadores não alcançaram um desempenho tão bom. Assim como para substituição, os experimentos com inserção e remoção mostraram que o conjunto de *features* é bastante importante para predição correta do C/D *box*, por classificadores de AM. Para as três mutações, Se características biológicas relevantes não são conhecidas, o desempenho dos classificadores fica bastante ruim, piorando para as sequências homólogas mais distantes. No entanto, para H/ACA *box*, o desempenho dos classificadores de AM foram equivalentes, tanto utilizando um número grande de *features* biológicas conhecidas quanto um número reduzido delas. Para a mutação de inserção, quanto maior a porcentagem de mutação, pior o desempenho dos três classificadores.

O segundo grupo de experimentos continha conjuntos contendo 5.000 sequências de C/D *box* e 5.000 sequências de H/ACA *box*. Para C/D *box*, destacamos o experimento com mutação de substituição por códons, que apresentou desempenhos descrescentes para os três classificadores. Além disso, a comparação com um número de *features* reduzidas mostrou o quanto as características biológicas relevantes conhecidas são importantes para os algoritmos de AM alcançarem bons desempenhos. Os resultados dos classificadores ANN e SVM para mutações com substituição, inserção e remoção mostraram que, quanto maior a porcentagem de mutação, menor será o desempenho dos algoritmos de AM. Porém, para substituição e inserção, os classificadores ANN e AVM apresentaram excelente desempenho com *AUC* acima de 90% para todos os conjuntos de dados (10%, 20%, 30%, 40% e 50% de mutação). Para H/ACA *box* o expermimento com mutação de substituição por códons, mostrou AUC acima de 77% para os classificadores ANN e SVM utilizando um número grande de *features* biológicas. Com a substituição, o desempenho dos classificadores de AM foram equivalentes, tanto utilizando um número maior de *features* biológicas conhecidas quanto um número reduzido delas. Para inserção, os três classificadores apresentaram melhor desempenho com 10% de mutação do que com 20%, independentemente do número de *features*. Além disso, os experimentos com a mutação de remoção mostraram que o tamanho da sequência de um snoRNA é uma característica muito importante para sua predição.

Em resumo, os resultados desses dois grupos de experimentos mostraram que os métodos de AM podem ser competitivos para tarefas de inferência de homologia, se conjuntos suficientemente grandes de sequências independentes puderem ser construídos e um número maior de *features* biológicas forem conhecidas para os ncRNAs que devem ser

preditos, para os conjuntos de treinamento e teste.

6.1 Contribuições

Este trabalho já teve como primeira contribuição o artigo *Construction of artificial non-coding RNA training data for machine learning studies*, aceito para a conferência *12th International Conference on Bioinformatics Models, Methods and Algorithms*, a ser realizado em Vienna, na Áustria. Esse artigo foi escrito com base nos resultados obtidos com os conjuntos de tamanhos $N = 3.000$ para C/D *box* snoRNA e $N = 2.000$ para H/ACA *box* snoRNA, com a colaboração do grupo do Prof. Peter Stadler, da Universidade de Leipzig, na Alemanha.

Além disso, com os novos resultados obtidos com os conjuntos de tamanhos $N = 5.000$, pretendemos alterar e melhorar substancialmente esse primeiro artigo e submeter a periódico internacional de bom nível.

6.2 Trabalhos futuros

As perspectivas deste trabalho são:

- Comparar o desempenho dos métodos de AM com métodos mais sofisticados de pesquisa de homologia. Justificamos essa linha de pesquisa pois permanece em aberto se os métodos de AM também podem competir com métodos mais sofisticados de inferência de homologia em ncRNAs, como modelos de covariância (CMs) [101, 26] e modelos termodinâmicos [24];
- Aplicar o método de construção dos conjuntos de dados artificiais para outras famílias de ncRNAs, como miRNAS, e avaliar o desempenho dos métodos de AM para miRNAs;
- Investigar sistematicamente o método de AM - Floresta Aleatória, pois esse classificador apresentou, em alguns experimentos, resultados distintos dos outros algoritmos avaliados - ANN e SVM.

Referências

[1] Lodish, H. e et.al: *Molecular Cell Biology*. W. H. Freeman, 4th edição, 2000, ISBN 9780716731368. xi, 1, 7, 8

[2] *Archivo:Difference DNA RNA-ES.svg*. https://es.wikipedia.org/wiki/Archivo:Difference_DNA_RNA-ES.svg, acesso em 2019-08-15. xi, 8

[3] Oliveira, J. e et.al: *SnoReport 2.0: new features and a refined Support Vector Machine to improve snoRNA identification*. BMC Bioinformatics, 17(18):464, dezembro 2016, ISSN 1471-2105. https://doi.org/10.1186/s12859-016-1345-6, acesso em 2019-05-10. xi, 2, 9, 14, 15, 19, 22, 24, 26, 42, 46, 59, 77

[4] Watson, J. e et.al: *Biologia Molecular do Gene - 7ed.* Artmed Editora, abril 2015, ISBN 9788582712092. Google-Books-ID: wMHxBwAAQBAJ. xi, 1, 8, 9, 10, 11

[5] Lorena, A. e A. Carvalho: *Uma Introdução às Support Vector Machines*. Revista de Informática Teórica e Aplicada, 14(2):43–67, 2007, ISSN 21752745. https://seer.ufrgs.br/rita/article/view/rita_v14_n2_p43-67, acesso em 2019-04-15. xi, 23, 25, 26

[6] ResearchGate, Scientific Figure on: *Sincronização de disparos em redes neuronais com plasticidade sináptica - scientific figure on researchgate.* https://www.researchgate.net/figure/Figura-1-Componentes-basicos-de-um-neuronio_fig1_282962512, acesso em 2020-10-15. xii, 28

[7] Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 1999. xii, 22, 26, 27, 28, 29, 30, 31

[8] Liu, D.: *A Practical Guide to ReLU*, novembro 2017. https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7, acesso em 2019-06-26. xii, 30

[9] *Random Forest Parameter Tuning | Tuning Random Forest*, junho 2015. https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/, acesso em 2019-02-17. xii, 31

[10] Watson, J. e Crick F.: *The Structure of Dna*. Cold Spring Harbor Symposia on Quantitative Biology, 18:123–131, janeiro 1953, ISSN 0091-7451, 1943-4456. http://symposium.cshlp.org/content/18/123, acesso em 2019-03-22. 1

[11] Watson, J. D.: *The human genome project: past, present, and future.* Science, 248(4951):44–49, abril 1990, ISSN 0036-8075, 1095-9203. https://science.sciencemag.org/content/248/4951/44, acesso em 2020-11-10. 1

[12] Rehm, B.: *Bioinformatic tools for DNA/protein sequence analysis, functional assignment of genes and protein classification.* Applied Microbiology and Biotechnology, 57(5-6):579–592, dezembro 2001, ISSN 0175-7598. 1

[13] Araújo, N. e et.al: *A ERA DA BIOINFORMÁTICA: SEU POTENCIAL E SUAS IMPLICAÇÕES PARA AS CIÊNCIAS DA SAÚDE.* Estudos de Biologia, 30(70/72), novembro 2008, ISSN 1980-590X. https://periodicos.pucpr.br/index.php/estudosdebiologia/article/view/22819, acesso em 2019-08-12. 1

[14] Mattick, J.: *Non-coding RNAs: the architects of eukaryotic complexity.* EMBO reports, 2(11):986–991, novembro 2001, ISSN 1469-221X. 1

[15] Veneziano, D., G. Nigita e A. Ferro: *Computational Approaches for the Analysis of ncRNA through Deep Sequencing Techniques.* Frontiers in Bioengineering and Biotechnology, 3, 2015, ISSN 2296-4185. https://www.frontiersin.org/articles/10.3389/fbioe.2015.00077/full, acesso em 2019-03-22. 1

[16] Zou, Q. e et. al: *Prediction of MicroRNA-Disease Associations Based on Social Network Analysis Methods*, 2015. https://www.hindawi.com/journals/bmri/2015/810514/, acesso em 2019-09-30. 2

[17] Falaleeva, M. e S. Stamm: *Processing of snoRNAs as a new source of regulatory noncoding RNAs: snoRNA fragments form a new class of functional RNAs.* BioEssays: News and Reviews in Molecular, Cellular and Developmental Biology, 35(1):46–54, janeiro 2013, ISSN 1521-1878. 2, 13

[18] Georgakilas, G. e et.al: *Multi-branch Convolutional Neural Network for Identification of Small Non-coding RNA genomic loci.* Scientific Reports, 10(1):9486, junho 2020, ISSN 2045-2322. https://www.nature.com/articles/s41598-020-66454-3, acesso em 2020-11-02. 2, 19, 24, 27

[19] Hertel, J., I. Hofacker e P. Stadler: *SnoReport: computational identification of snoRNAs with unknown targets.* Bioinformatics (Oxford, England), 24(2):158–164, janeiro 2008, ISSN 1367-4811. 2, 14, 19

[20] Navarin, N. e F. Costa: *An efficient graph kernel method for non-coding RNA functional prediction.* Bioinformatics, 33(17):2642–2650, setembro 2017, ISSN 1367-4803. https://academic.oup.com/bioinformatics/article/33/17/2642/3798629, acesso em 2019-08-12. 2

[21] Yi, H. e et.al: *A Deep Learning Framework for Robust and Accurate Prediction of ncRNA-Protein Interactions Using Evolutionary Information.* Molecular Therapy - Nucleic Acids, 11:337–344, junho 2018, ISSN 2162-2531. http://www.sciencedirect.com/science/article/pii/S2162253118300313, acesso em 2019-08-12. 2

[22] Panwar, B. e et.al: *Prediction and classification of ncRNAs using structural information.* BMC Genomics, 15(1):127, fevereiro 2014, ISSN 1471-2164. https://doi.org/10.1186/1471-2164-15-127, acesso em 2019-08-15. 2

[23] Zhang, Y. e et.al: *A Review on Recent Computational Methods for Predicting Noncoding RNAs*, 2017. https://www.hindawi.com/journals/bmri/2017/9139504/abs/, acesso em 2019-09-30. 2, 3

[24] Lorenz, R. e et.al: *ViennaRNA Package 2.0.* Alg. Mol. Biol., 6:26, 2011. 2, 85

[25] Freyhult, E., J. Bollback e P. Gardner: *Exploring genomic dark matter: A critical assessment of the performance of homology search methods on noncoding RNA.* Genome Research, 17(1):117–125, janeiro 2007, ISSN 1088-9051, 1549-5469. http://genome.cshlp.org/content/17/1/117, acesso em 2020-11-24. 3

[26] P., Nawrocki e Eddy R.: *Infernal 1.1: 100-fold faster RNA homology searches.* Bioinformatics, 29(22):2933–2935, novembro 2013, ISSN 1367-4803. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3810854/, acesso em 2019-10-30. 3, 85

[27] Altschul, S. e et.al: *Basic local alignment search tool.* Journal of molecular biology, 215(3):403–410, 1990. 3, 36, 44

[28] Bartschat, S. e et.al: *snoStrip: a snoRNA annotation pipeline.* Bioinformatics, 30(1):115–116, janeiro 2014, ISSN 1367-4811. 3, 18

[29] Barber, D.: *Bayesian Reasoning and Machine Learning.* Cambridge University Press, Cambridge, UK, 2012. 3, 22

[30] Zhang, Y. e C. Rajapakse: *Machine Learning in Bioinformatics.* John Wiley & Sons, fevereiro 2009, ISBN 9780470397411. 3

[31] Crick, F.: *Central dogma of molecular biology.* Nature, 227(5258):561–563, agosto 1970, ISSN 0028-0836. 6

[32] Watson, J. e F. Crick: *Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid.* Nature, 171(4356):737–738, abril 1953, ISSN 1476-4687. https://www.nature.com/articles/171737a0, acesso em 2019-08-15. 7

[33] Sinden, R.: *DNA Structure and Function.* Gulf Professional Publishing, novembro 1994, ISBN 9780126457506. 8

[34] Matera, A., R. Terns e M. Terns: *Non-coding RNAs: lessons from the small nuclear and small nucleolar RNAs.* Nature Reviews Molecular Cell Biology, 8(3):209–220, março 2007, ISSN 1471-0080. https://www.nature.com/articles/nrm2124, acesso em 2019-07-08. 12

[35] Hombach, S. e M. Kretz: *Non-coding RNAs: Classification, Biology and Functioning.* Em Slaby, Ondrej e George A. Calin (editores): *Non-coding RNAs in Colorectal Cancer*, Advances in Experimental Medicine and Biology, páginas 3–17. Springer International Publishing, Cham, 2016, ISBN 9783319420592. https://doi.org/10.1007/978-3-319-42059-2_1, acesso em 2019-08-15. 12

[36] Machado, L., H. Portillo e A. Durham: *Computational methods in noncoding RNA research*. Journal of Mathematical Biology, 56(1-2):15–49, janeiro 2008, ISSN 0303-6812. 12, 17

[37] Paschoal, A.: *Bioinformática aplicada em RNomics: estratégias computacionais para caracterização de RNAs não-codificadores*. text, Universidade de São Paulo, abril 2012. http://www.teses.usp.br/teses/disponiveis/95/95131/tde-24092013-211625/, acesso em 2019-07-08. 12, 17

[38] Burge, S. e et.al: *Rfam 11.0: 10 years of RNA families*. Nucleic Acids Research, 41(D1):D226–D232, janeiro 2013, ISSN 0305-1048. https://academic.oup.com/nar/article/41/D1/D226/1050811, acesso em 2019-07-08. 12, 20

[39] Bachellerie, J., J. Cavaillé e A. Hüttenhofer: *The expanding snoRNA world*. Biochimie, 84(8):775–790, agosto 2002, ISSN 0300-9084. http://www.sciencedirect.com/science/article/pii/S0300908402014025, acesso em 2019-11-04. 13, 14

[40] Mcmahon, M., A. Contreras e D. Ruggero: *Small RNAs with big implications: new insights into H/ACA snoRNA function and their role in human disease*. Wiley Interdisciplinary Reviews: RNA, 6(2):173–189, 2015, ISSN 1757-7012. https://onlinelibrary.wiley.com/doi/abs/10.1002/wrna.1266, acesso em 2019-03-24. 13

[41] Samarsky, A., J. Fournier, H. Singer e E. Bertrand: *The snoRNA box C/D motif directs nucleolar targeting and also couples snoRNA synthesis and localization*. The EMBO Journal, 17(13):3747–3757, julho 1998, ISSN 0261-4189. https://www.embopress.org/doi/full/10.1093/emboj/17.13.3747, acesso em 2020-06-04. 14

[42] Taft, R. e et.al: *Small RNAs derived from snoRNAs*. RNA, 15(7):1233–1240, julho 2009, ISSN 1355-8382. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2704076/, acesso em 2019-03-24. 14

[43] Alberts, B. e et.al: *Molecular Biology of the Cell*. Garland Science, 4th edição, 2002, ISBN 9780815332183 9780815340720. 15

[44] Kimura, M.: *The role of compensatory neutral mutations in molecular evolution*. Journal of Genetics, 64(1):7, julho 1985, ISSN 0973-7731. https://doi.org/10.1007/BF02923549, acesso em 2019-11-04. 15

[45] Kimura, Motoo: *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge, 1983, ISBN 9780521317931. https://www.cambridge.org/core/books/neutral-theory-of-molecular-evolution/0FF60E9F47915B17FFA2620C49400632, acesso em 2019-12-04. 15

[46] Ofria, C., C. Adami e T. Collier: *Selective pressures on genomes in molecular evolution*. Journal of Theoretical Biology, 222(4):477–483, junho 2003, ISSN 0022-5193. http://www.sciencedirect.com/science/article/pii/S0022519303000626, acesso em 2020-09-10. 15

[47] Milholland, B. e et.al: *Differences between germline and somatic mutation rates in humans and mice*. Nature Communications, 8(1):15183, maio 2017, ISSN 2041-1723. https://www.nature.com/articles/ncomms15183, acesso em 2019-12-04. 15

[48] Schoen, Daniel J. e Stewart T. Schultz: *Somatic Mutation and Evolution in Plants*. Annual Review of Ecology, Evolution, and Systematics, 50(1):49–73, 2019. https://doi.org/10.1146/annurev-ecolsys-110218-024955, acesso em 2020-10-20. 15

[49] Moore, A.: *Science as a Way of Knowing—Genetics*. Integrative and Comparative Biology, 26(3):583–747, agosto 1986, ISSN 1540-7063. https://academic.oup.com/icb/article/26/3/583/264293, acesso em 2019-10-14. 15

[50] Loewe, L. e W. Hill: *The population genetics of mutations: good, bad and indifferent*. Philosophical Transactions of the Royal Society B: Biological Sciences, 365(1544):1153–1167, abril 2010. https://royalsocietypublishing.org/doi/10.1098/rstb.2009.0317, acesso em 2020-06-19. 15

[51] Bhartiya, D. e V. Scaria: *Genomic variations in non-coding RNAs: Structure, function and regulation*. Genomics, 107(2):59–68, março 2016, ISSN 0888-7543. http://www.sciencedirect.com/science/article/pii/S0888754316300052, acesso em 202-12-04. 16

[52] Ono, Motoharu, Kayo Yamada, Fabio Avolio, Michelle S. Scott, Silvana van Koningsbruggen, Geoffrey J. Barton e Angus I. Lamond: *Analysis of Human Small Nucleolar RNAs (snoRNA) and the Development of snoRNA Modulator of Gene Expression Vectors*. Molecular Biology of the Cell, 21(9):1569–1584, maio 2010, ISSN 1059-1524. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2861615/, acesso em 2019-11-04. 16

[53] *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*. Elsevier, agosto 2018, ISBN 9780128114322. Google-Books-ID: rs51DwAAQBAJ. 16

[54] Altschul, S. e et.al: *Basic local alignment search tool*. Journal of Molecular Biology, 215(3):403–410, outubro 1990, ISSN 0022-2836. 16, 17, 44

[55] Wilbur, W.: *On the PAM matrix model of protein evolution*. Molecular Biology and Evolution, 2(5):434–447, setembro 1985, ISSN 0737-4038. https://academic.oup.com/mbe/article/2/5/434/974047, acesso em 2019-08-24. 16

[56] Schneider, A., G. Cannarozzi e G. Gonnet: *Empirical codon substitution matrix*. BMC Bioinformatics, 6(1):134, junho 2005, ISSN 1471-2105. https://doi.org/10.1186/1471-2105-6-134, acesso em 2020-09-24. 16

[57] Pareek, C., R. Smoczynski e A. Tretyn: *Sequencing technologies and genome sequencing*. Journal of Applied Genetics, 52(4):413–435, novembro 2011, ISSN 2190-3883. https://doi.org/10.1007/s13353-011-0057-x, acesso em 2019-08-12. 17

[58] *» BLAST GeneBio – Genética e Bioinformática*. http://www.genebio.ufba.br/?page_id=260, acesso em 2019-08-12. 17

[59] Nawrocki, E., D. Kolbe e S. Eddy: *Infernal 1.0: inference of RNA alignments*. Bioinformatics, 25(10):1335–1337, maio 2009, ISSN 1367-4803, 1460-2059. https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp157, acesso em 2019-03-25. 17

[60] Hofacker, I. e et.al: *Fast folding and comparison of RNA secondary structures*. Monatshefte für Chemie / Chemical Monthly, 125(2):167–188, fevereiro 1994, ISSN 1434-4475. https://doi.org/10.1007/BF00818163, acesso em 2019-06-06. 18

[61] Yang, J. e et.al: *snoSeeker: an advanced computational package for screening of guide and orphan snoRNA genes in the human genome*. Nucleic Acids Research, 34(18):5112–5123, 2006, ISSN 1362-4962. 18

[62] Arrial, R., R. Togawa e M. Brigido: *Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus Paracoccidioides brasiliensis*. BMC Bioinformatics, 10(1):239, agosto 2009, ISSN 1471-2105. https://doi.org/10.1186/1471-2105-10-239, acesso em 2019-06-06. 19

[63] *NONCODE*. http://www.noncode.org/, acesso em 2019-08-12. 19

[64] Pang, K. e et.al: *RNAdb 2.0—an expanded database of mammalian non-coding RNAs*. Nucleic Acids Research, 35(Database issue):D178–D182, janeiro 2007, ISSN 0305-1048. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1751534/, acesso em 2019-08-12. 20

[65] *miRBase*. http://www.mirbase.org/, acesso em 2019-08-12. 20

[66] Lestrade, L. e M. Weber: *snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs*. Nucleic Acids Research, 34(suppl_1):D158–D162, janeiro 2006, ISSN 0305-1048. https://academic.oup.com/nar/article/34/suppl_1/D158/1132186, acesso em 2019-08-12. 20

[67] Yoshihama, M., A. Nakao e N. Kenmochi: *snOPY: a small nucleolar RNA orthological gene database*. BMC Research Notes, 6(1):426, outubro 2013, ISSN 1756-0500. https://doi.org/10.1186/1756-0500-6-426, acesso em 2019-08-12. 20

[68] Lopes, R.: *Desenvolvimento de ferramentas para a identificação de marcadores moleculares e imunológicos a partir de dados genômicos como alvo para o diagnóstico de doenças parasitárias*, agosto 2015. http://www.bibliotecadigital.ufmg.br/dspace/handle/1843/BUBD-A2PGE4, acesso em 2019-04-15. 21, 27

[69] Faceli, K. e et.al: *Inteligência artificial: uma abordagem de aprendizado de máquina*. 2011. https://bdpi.usp.br/item/002208293, acesso em 2019-08-26. 21

[70] Mitchell, T.: *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1ª edição, 1997, ISBN 9780070428072. 21, 22, 24, 25, 28

[71] S., Russell e Norvig P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, 3nd edição, 2010. 22

[72] Breiman, L.: *Random forests*. Machine Learning, 45:5–32, 2001. 22

[73] Dietterich, T.: *Ensemble Methods in Machine Learning*. Em *Multiple Classifier Systems*, Lecture Notes in Computer Science, páginas 1–15, Berlin, Heidelberg, 2000. Springer, ISBN 9783540450146. 22

[74] Oja, E. e S. Kaski: *Kohonen Maps*. Elsevier, julho 1999, ISBN 9780080535296. Google-Books-ID: JsRXtwWjJPsC. 23, 31

[75] Haykin, S. e Z. Chen: *The Cocktail Party Problem*. Neural Computation, 17(9):1875–1902, setembro 2005, ISSN 0899-7667. https://doi.org/10.1162/0899766054322964, acesso em 2019-10-21. 23

[76] Zhen-feng, H. e X. Fan-lun: *A Constrained Partition Model and K-Means Algorithm*. 2005. 23

[77] Basu, S., A. Banerjee e R. Mooney: *Semi-supervised Clustering by Seeding*. Em *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*, 2002. 23

[78] Peng, J. e R. Williams: *Technical Note*. Em Kaelbling, Leslie Pack (editor): *Recent Advances in Reinforcement Learning*, páginas 283–290. Springer US, Boston, MA, 1996, ISBN 9780585336565. https://doi.org/10.1007/978-0-585-33656-5_12, acesso em 2019-1-04. 24

[79] Singh, S. e R. Sutton: *Reinforcement learning with replacing eligibility traces*. Machine Learning, 22(1):123–158, março 1996, ISSN 1573-0565. https://doi.org/10.1007/BF00114726, acesso em 2020-06-10. 24

[80] Sutton, R. e A. Barto: *Reinforcement Learning, second edition: An Introduction*. MIT Press, novembro 2018, ISBN 9780262352703. Google-Books-ID: uWV0DwAAQBAJ. 24

[81] Pan, X. e el.al: *Analysis of Expression Pattern of snoRNAs in Different Cancer Types with Machine Learning Algorithms*. International Journal of Molecular Sciences, 20(9):2185, janeiro 2019. https://www.mdpi.com/1422-0067/20/9/2185, acesso em 2020-05-23. 24

[82] Achawanantakun, R.: *LncRNA-ID: Long non-coding RNA IDentification using balanced random forests*. Bioinformatics, 31(24):3897–3905, dezembro 2015, ISSN 1367-4803. https://academic.oup.com/bioinformatics/article/31/24/3897/196877, acesso em 2019-05-23. 24

[83] Petrosyan, A. e et.al: *Neural network integral representations with the ReLU activation function*. Em *Mathematical and Scientific Machine Learning*, páginas 128–143. PMLR, agosto 2020. http://proceedings.mlr.press/v107/petrosyan20a.html, acesso em 2020-11-26. 30

[84] Wickert, I. e F. França: *AUTOWISARD: Unsupervised Modes for the WISARD*. Em Mira, José e Alberto Prieto (editores): *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*, Lecture Notes in Computer Science, páginas 435–441, Berlin, Heidelberg, 2001. Springer, ISBN 9783540457206. 31

[85] Frieß, T. e R. Harrison: *A kernel-based Adaline for function approximation*. Intelligent Data Analysis, 3(4):307–313, outubro 1999, ISSN 1088-467X. `http://www.sciencedirect.com/science/article/pii/S1088467X99000256`, acesso em 2020-05-04. 31

[86] Austin, J.: *RAM-based Neural Networks*. World Scientific, 1998, ISBN 9789810232535. Google-Books-ID: oftamBSoA1IC. 31

[87] Breiman, L.: *Random forests*. Machine Learning, 45:5–32, 2001. 31, 32

[88] *Project Jupyter*. `https://www.jupyter.org`, acesso em 2019-05-12. 33

[89] Gulli, A. e S. Pal: *Deep Learning with Keras*. Packt Publishing Ltd, abril 2017, ISBN 9781787129030. Google-Books-ID: 20EwDwAAQBAJ. 33

[90] Kingma, D. e J. Ba: *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980 [cs], janeiro 2017. `http://arxiv.org/abs/1412.6980`, acesso em 2020-07-25, arXiv: 1412.6980. 34

[91] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jacob T Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot e Édouard Duchesnay: *Scikit-learn: Machine learning in python*. J. Machine Learning Res., 12:2825–2830, 2011. 34

[92] *LIBSVM – A Library for Support Vector Machines*. `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`, acesso em 2019-05-10. 34

[93] Information, National Center for Biotechnology, U. S. National Library of Medicine 8600 Rockville Pike, Bethesda MD e 20894 Usa: *National Center for Biotechnology Information*. `https://www.ncbi.nlm.nih.gov/`, acesso em 2019-08-15. 36

[94] Satoh, N.: *The ascidian tadpole larva: comparative molecular development and genomics*. Nature Reviews Genetics, 4(4):285–295, abril 2003, ISSN 1471-0064. `https://www.nature.com/articles/nrg1042`, acesso em 2020-09-12. 37

[95] Rudolf, J. e et.al: *Automated behavioural analysis reveals the basic behavioural repertoire of the urochordate Ciona intestinalis*. Scientific Reports, 9(1):2416, fevereiro 2019, ISSN 2045-2322. `https://www.nature.com/articles/s41598-019-38791-5`, acesso em 2020-10-29. 37

[96] Rastogi, A. e D. Gupta: *Gff-ex: a genome feature extraction package*. BMC research notes, 7(1):1–3, 2014. 37

[97] Goldschmidt, R. e E Passos: *Data mining: a Practical guide*. Gulf Professional Publishing, 2005. 43

[98] Witten, I. e et.al: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, outubro 2016, ISBN 9780128043578. Google-Books-ID: 1SylCgAAQBAJ. 44

[99] Fawcett, T.: *ROC Graphs: Notes and Practical Considerations for Researchers*. Relatório Técnico, 2004. 44

[100] Oliveira, J. e et.al: *SnoReport 2.0: new features and a refined Support Vector Machine to improve snoRNA identification*. BMC Bioinformatics, 17 Suppl. 18:464, 2016. 53, 60, 70, 81

[101] Eddy, R.: *Hidden markov models*. Current Op. Struct. Biol., 6:361–365, 1996. 85

Anexo I

Impacto de mutações na predição de snoRNAs

>XR_003396989.1 PREDICTED: Ciona intestinalis small nucleolar RNA U3 (LOC113475245), ncRNA
XR_003396989.1_13

Nucleotídeo	Posição
A	67
A	69
A	98
A	139
A	141
T	80
C	112
G	37

aagtaataaaaaaataagggaAAGATTGATAGTGTTAGTAATATTACTAGAAATAGTCAAGCTTTTAAGA
TCAATGTATGTCGTGTATATCTGGTAGAAACAACATCCATTTCTAGTGTAAAATGTGATGACAGT
ACTTAGAATTTATGTaacataacatatttataactGTAGCAGTTTACTGATTTTGTAATTGCCTTAcgaaact

>XR_003396992.1 PREDICTED: Ciona intestinalis small nucleolar RNA U3 (LOC113475247), ncRNA
XR_003396992.1_13

Nucleotídeo	Posição
A	86
A	88
A	117
A	158
A	160
T	99
C	131
G	46

TTAACTTGGGTAACGtagtaagtaataaaaaaataagggaAAGATTGATAGTGTTAGTAATATTACTAG
AAATAGTCAAGCTTTTAAGATCAATGTATGTCGTGTATATCTGGTAGAAACAACATCCATTTC
TAGTGTAAAATGTGATGACAGTACTTAGAATTTATGTaacataacatatttataactGTAGCAGTTTAC
TGATTTTGTAATTGCCTTAcgaaact

>XR_003396993.1 PREDICTED: Ciona intestinalis small nucleolar RNA U3 (LOC113475248), ncRNA
XR_003396993.1_14

Nucleotídeo	Posição
A	50
A	52
A	81
A	122
A	124
T	63
C	95
G	20

gggaAAGATTGATAGTGTTAGTAATATTACTAGAAATAGTCAAGCTTTTAAGATCAATGTATGTCG
TGTATATCTGGTAGAAACAACATCCATTTCTAGTGTAAAATGTGATGACAGTACTTAGAATTTAT
GtaacataacatatttataactGTAGCAGTTTACTGATTTTGTAATTGCCTTAcgaaact

>XR_003396997.1 PREDICTED: Ciona intestinalis small nucleolar RNA U3 (LOC113475252), ncRNA
XR_003396997.1_13

Nucleotídeo	Posição
A	62
A	64
A	93
A	134
A	136
T	75
C	107
G	32

ataaaaaaataagggaAAGATTGATAGTGTTAGTAATATTACTAGAAATAGTCAAGCTTTTAAGATC
AATGTATGTCGTGTATATCTGGTAGAAACAACATCCATTTCTAGTGTAAAATGTGATGACAGT
ACTTAGAATTTATGTaacataacatatttataactGTAGCAGTTTACTGATTTTGTAATTGCCTTAcgaaact

>XR_003397000.1 PREDICTED: Ciona intestinalis small nucleolar RNA U3 (LOC113475255), ncRNA
XR_003397000.1_13

Nucleotídeo	Posição
A	5
A	44
A	46
A	72
A	74
A	75
A	116
A	118
T	24
T	57
T	71
T	88
T	90
C	23
C	58
C	89
G	14
G	73

GATTGATAGTGTTAGTAATATTACTAGAAATAGTCAAGCTTTTAAGATCAATGTATGTCGTG
TATATCTGGTAGAAACAACATCCATTTCTAGTGTAAAATGTGATGACAGTACTTAGAATTTA
TGTaacataacatatttataactGTAGCAGTTTACTGATTTTGTAATTGCCTTAcgaaact