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

Método baseado em aprendizado de máquina para seleção de características para distinção entre RNAs não-codificadores longos e RNAs codificadores de proteínas

Bruno Couto Kümmel

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientadora
Profa. Dra. Maria Emília Machado Telles Walter

Brasília
2017

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

Método baseado em aprendizado de máquina para seleção de características para distinção entre RNAs não-codificadores longos e RNAs codificadores de proteínas

Bruno Couto Kümmel

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

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

Prof. Dr. André Carlos P. L. F. de Carvalho
Univesidade de São Paulo

Profa. Dra. Célia Ghedini Ralha
Universidade de Brasília

Prof. Dr. Bruno Luiggi Macchiavello Espinoza
Coordenador do Programa de Pós-graduação em Informática

Brasília, 12 de dezembro de 2017

Dedicatória

Dedico este trabalho à minha família, aos meus professores e a todos aqueles que buscam na ciência a resposta para os problemas e desafios da humanidade.

Agradecimentos

À minha esposa, Alessandra, pelo apoio e paciência (e pelas correções de português). À minha família, e em especial aos meus pais, Regina e Weber, por me ensinarem o valor e a importância do estudo. À minha orientadora, Profa. Dra. Maria Emília Machado Telles Walter, pela confiança depositada em mim. Ao professor André Carlos Ponce de Leon Ferreira de Carvalho pela paciência e pelos ensinamentos. À Universidade de Brasília, ao Departamento de Ciência da Computação e em especial à secretaria de pós-graduação. Ao Instituto de Ciências Biológicas e aos amigos e colegas de curso: Lucas, Daniel, Waldeyr, Julien e Hugo. Aos amigos e colegas do Centro de Pesquisa: Otávio, Robson, Rodrigo, Rafael, Fábio e Jones. E por último ao sr. Sulu pela companhia e compreensão.

"I began to read the paper. It kept talking about extensors and flexors, the gastrocnemius muscle, and so on. This and that muscle were named, but I had not the foggiest idea of where they were located in relation to the nerves or to the cat. So I went to the librarian in the biology section and asked her if she could find me a map of the cat. "A map of the cat, sir?" she asked horrified. "You mean a zoological chart!" From then on there were rumors about a dumb biology student who was looking for "a map of the cat"."

Richard P. Feynman

Resumo

RNAs não-codificadores longos (*long non-coding* RNA - lncRNAs) constitui uma classe heterogênea de RNAs que agrega transcritos com pouca capacidade de codificar proteínas e que possuem mais de 200 nucleotídeos em sua composição. Estudos recentes apontam que essas moléculas possuem funções de regulação de processos biológicos importantes dentro das células. Sabe-se também que o nível de expressão dos lncRNAs está correlacionado com diversas doenças genéticas, tais como câncer e doenças neuro-degenerativas. Este trabalho apresenta um método para seleção das características mais relevantes para modelos de aprendizado de máquina aplicados ao problema de distinguir lncRNAs de transcritos codificadores de proteínas. O método proposto, denominado *Single Score Feature Selection* (S2FS), utilizou como características as frequências de 2-mers, 3-mers e 4-mers dos transcritos, para detectar aquelas mais relevantes para distinguir lncRNAs de transcritos codificadores de proteínas. As características identificadas pelo S2FS foram avaliadas nos *datasets* obtidos de repositórios públicos de transcritos RNAs codicadores de proteínas e de lncRNAs de *Homo sapiens*, *Mus musculus* e *Danio rerio*. Para o dataset de *H. sapiens*, também foi utilizada a característica da ORF mais longa de cada transcrito. Os resultados obtidos indicam que o S2FS identificou boas características para os modelos de predição de lncRNAs baseados em *Random Forest*. Nos modelos de classificação testados, as características selecionadas pelo S2FS possibilitaram resultados melhores do que as características selecionadas por um método de seleção univariada de características baseado no escore da função χ^2.

Palavras-chave: LncRNAs, Distinção de lncRNAs e transcritos codificadores de proteínas, Seleção de características, Aprendizado de máquina

Abstract

Long non-coding RNA (lncRNAs) constitutes a heterogeneous class of RNAs that includes RNAs with more than 200 nucleotides and poor capacity for coding proteins. Recent studies have indicated that these molecules act on critical biological processes inside the cells. However, their expression levels are also correlated with a number of complex human diseases, such as cancer, neuro-degenerative diseases and others. This work proposes a method for feature selection for machine learning methods applied to the task of distinguishing lncRNAs from protein coding transcripts. The proposed method, called Single Score Feature Selection (S2FS), used as features the 2-mer, 3-mer and 4-mer frequencies of the transcripts, in order to detect those more relevant to distinguish lncRNAs from protein coding transcripts. The features identified by S2FS were evaluated on datasets obtained from public repositories of protein coding transcripts and lncRNAs of *Homo Sapiens*, *Mus musculus* and *Danio rerio*. For the *H. sapiens* dataset, the longest ORF of each transcript was also used as a feature. The obtained results show that the S2FS identified good features for the lncRNA prediction models based on Random Forest. In the tested classification models, the selected features from S2FS enabled better performance results than the features selected by an univariate selection method based on the scores of a χ^2 function.

Keywords: LncRNA, Distinction of LncRNA and protein coding transcripts, Feature selection, Machine Learning

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AUC *Area Under the Curve.*

BLAST *Basic local alignment search tool.*

CNCI *Coding-Non-Coding Index.*

CPAT *Coding-Potential Assessment Tool.*

CPC *Coding Potential Calculator.*

DBSCAN *Density-based spatial clustering of applications with noise.*

DNA Ácido Desoxirribonucleico.

FEELnc *FlExible Extraction of LncRNAs.*

FSBE *Forward Selection-Backward Elimination.*

HOTAIR HOX antisense intergenic RNA.

KNN *K-nearest neighbors.*

LDA Análise de Discriminantes Lineares.

lincRNAs *long intergenic* RNAs.

lncRNA ncRNAs longos - (*long non-coding* RNAs).

lncRNA-MFDL *Identification of human long non-coding RNAs by fusing multiple features and using deep learning.*

miRNAs MicroRNAs.

mRNA RNA mensageiro.

ncRNAs RNAs não-codificadores.

ORF *Open Reading Frame.*

PCA *Principal Componente Analysis.*

PCTs transcritos codificadores de proteínas.

piRNAs Piwi-interacting RNAs.

PLEK *Predictor of Long non-coding RNAs and messenger RNAs based on an improved k-mer scheme.*

RF *Random Forest.*

RISC *RNA-induced silencing complex.*

RNA Ácido Ribonucleico.

rRNA RNA ribossomal.

SFS *Sequential Feature Selection.*

siRNAs Small interfering RNAs.

SOM *Self-Organizing Map.*

SVM *Support Vector Machine.*

tRNA RNA transportador.

Capítulo 1

Introdução

A publicação do artigo *Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid* [126], no qual é descrita pela primeira vez a estrutura de dupla hélice do Ácido Desoxirribonucleico (DNA), é considerada um marco importante para a Biologia Molecular, área que utiliza conceitos e métodos da Física e Química para estudar ácidos nucleicos e estruturas de proteínas, descrever a interação entre biomoléculas, além de identificar processos bioquímicos intracelulares envolvendo organelas celulares e enzimas.

Os ácidos nucleicos são biopolímeros existentes em todas as formas conhecidas de vida e são categorizados entre DNA e Ácido Ribonucleico (RNA). Esses ácidos estão envolvidos no armazenamento de informações hereditárias, na produção de proteínas e na regulação desses processos dentro do núcleo da célula [5].

O desenvolvimento das pesquisas, da publicação de Watson e Crick [126] até os dias de hoje, possibilitou um grande entendimento de como esses processos ocorrem dentro da célula. De forma geral, compreende-se que determinadas regiões do DNA, por meio do processo de transcrição, são transformadas em uma molécula de RNA mensageiro (mRNA). Em seguida, esses mRNAs, pelo processo de tradução, em conjunto com oRNA transportador (tRNA) e o RNA ribossomal (rRNA), sintetizam proteínas [104].

No entanto, ainda há muito o que se descobrir na Biologia Molecular, pois, como em diversas áreas da Biologia, é uma área dinâmica, na qual os avanços ocorrem muito rapidamente. Novas descobertas e novos métodos vão alterando ou consolidando teorias e hipóteses existentes, que não puderam ser testadas ou observadas anteriormente. Um exemplo clássico desse processo é o próprio Dogma Central da Biologia Molecular, sedimentado por Crick [26], que, à época, estabelecia que a única função do DNA era transcrever RNA para a produção de proteínas. Hoje, sabe-se que existem RNAs que não traduzem proteínas, mas que atuam na regulação gênica das células [106, 105, 63]. Estes RNAs são conhecidos como RNAs não-codificadores (ncRNAs) e estão divididos, de maneira geral, em duas classes: ncRNAs pequenos, com até 200 nucleotídeos de extensão,

e os ncRNAs longos - (*long non-coding* RNAs) (lncRNA), com mais de 200 nucleotídeos.

Pode-se considerar que o início da era do grande volume de informações na Biologia Molecular ocorreu em 2001 com o Projeto Genoma Humano [64, 119], no qual foram identificados mais de 3 bilhões de pares de bases nos 23 pares de cromossomos que compõem o DNA humano. Desde então, muito se evoluiu em relação às pesquisas de larga escala com ferramentas de alto desempenho para sequenciamento.

O Projeto Genoma Humano propiciou o surgimento de uma nova área de pesquisa, a Bioinformática, que agrega disciplinas de outras áreas para auxiliar na aquisição de novos conhecimentos, como Matemática, Estatística e Computação. Em particular, no contexto em que este trabalho foi desenvolvido, métodos computacionais auxiliam no armazenamento e no processamento de uma quantidade enorme, e crescente, de dados.

O desenvolvimento da nova geração de métodos de sequenciamento de DNA representou um grande salto na velocidade de obtenção de dados biomoleculares, não apenas pela velocidade do método em si, mas pela redução do custo de obtenção desses dados (Figura 1.1). Como exemplo de métodos *Next Generation Sequencing*-NGS, pode-se citar o pirossequenciamento, desenvolvido por Ronaghi e Nyrén em 1996 [99], ou o sequenciamento por síntese, desenvolvido a partir dos trabalhos realizados por Canard e Sarfati [14].

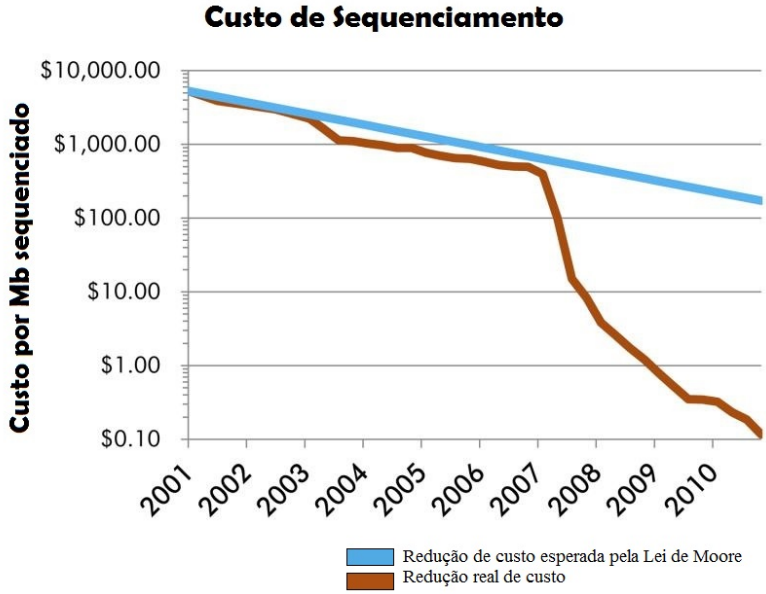

Figura 1.1: Redução de custos de sequenciamento ao longo do tempo. A curva azul representa a redução de custo esperada, se essa acompanhasse o aumento de desempenho computacional previsto na lei de Moore. A curva marrom representa a redução de custo real com o surgimento de ferramentas NGS como pirossequenciamento, sequenciamento por síntese e outros (traduzido de [88]).

A tecnologia RNA-Seq [125] também utiliza métodos NGS para revelar a presença e quantidade de RNA em uma amostra biológica qualquer [125]. O RNA-Seq permite identificar RNAs oriundos de *splicing* alternativo, modificações pós-transcrição, mutações e outras mudanças na expressão dos genes de origem dos transcritos [75].

Com a melhoria das técnicas e redução dos custos de produção de dados, torna-se possível produzir informações para estudos de diversas espécies. Atualmente, estima-se que haja um trilhão de diferentes organismos no planeta [72]. Dessa forma, é esperado que novas técnicas sejam desenvolvidas, não apenas para obter os dados, mas também para processar esse volume enorme de dados, transformando-o em informação biológica compreensível aos pesquisadores. Dentre os diversos métodos existentes para processar e extrair informações de quantidades de dados tão grandes como essas, as técnicas de Inteligência Artificial mostram-se bastante promissoras.

As contribuições da Inteligência Artificial para essas pesquisas cobrem uma vasta gama de aplicações, como análise de sequências genômicas, transcritômicas e proteômicas, representação e predição de estruturas moleculares e simulação de interação de sistemas biológicos, entre outros [54, 93, 113].

1.1 Motivação

A justificativa para este trabalho encontra-se não apenas na redução de custo e tempo de processamento como já mencionado, mas também no interesse que pesquisas recentes têm mostrado a respeito do tipo ainda não completamente conhecido de RNA, o lncRNA. Algumas de suas funções e características, no entanto, já foram mapeadas e observadas. Sabe-se, por exemplo, que algumas moléculas de lncRNAs atuam na regulação gênica, e que essa função está mais correlacionada à sua estrutura secundária do que à sequência de nucleotídeos [57].

Outra razão pelo interesse por essas moléculas está nas crescentes evidências que apontam que várias alterações no genoma que acarretam doenças como câncer ocorrem em regiões que são transcritas como lncRNAs [90]. As técnicas de sequenciamento de nova geração aplicadas a transcritomas de amostras de alguns tipos de câncer apontam para diferença significativa de expressão dessas moléculas (*upregulation* e *downregulation*) em relação a amostras de indivíduos saudáveis [70, 60].

No entanto, apesar da importância dos lncRNAs em diversos processos de regulação gênica e das crescentes evidências de sua atuação em diversas doenças genéticas, ainda não há uma ferramenta *in silico* amplamente aceita para a predição de lncRNAs, tampouco um estudo sistemático de identificação de características determinantes para a identificação desses RNAs.

Assim, existe bastante interesse no desenvolvimento de ferramentas para predição de lncRNAs. Muitas das ferramentas já desenvolvidas para a predição de ncRNAs, de maneira mais geral, utilizam conjuntos distintos de características em seus modelos. No entanto, ao contrário do senso comum, não se pode simplesmente agregar todas as características já levantadas nas diferentes pesquisas para criar um classificador melhor, devido a um conceito de aprendizado de máquina denominado de Maldição de Dimensionalidade [59].

Também existem fortes indícios de que não há um conjunto universal de características que seja útil para a predição de ncRNAs para todas as espécies [120]. Sendo assim, o pré-processamento de escolha das melhores características é uma etapa necessária para o bom desempenho das ferramentas computacionais.

1.2 Problema

Tanto quanto sabemos, não há um conjunto definido de características provenientes das sequências de transcritos que sejam capazes de distinguir lncRNAs de transcritos codificadores de proteínas (PCTs) em diferentes espécies. Devido ao conceito de Maldição de Dimensionalidade, não se pode acrescentar o número de características sem aumentar o número de amostras de treinamento; caso contrário, ocorre *overfitting* do modelo de classificação com os dados de treinamento, acarretando modelos com menor acurácia.

1.3 Objetivos

O objetivo principal deste trabalho é propor um método de seleção de características relacionadas às frequências dos k-mers das sequências de transcritos gerados por RNA-Seq. As frequências dos k-mers são características bastante utilizadas em ferramentas computacionais para a distinção de ncRNAs de PCTs.

Esse método deve ser usado no pré-processamento de um modelo que distingue lncRNAs de PCTs, para escolher o menor número de características relevantes para treinar um modelo de aprendizado supervisionado com boa acurácia.

Os objetivos específicos são:

1. Propor técnica para a seleção de características relacionadas às frequências dos k-mers das sequências utilizando três abordagens distintas: por clusterização, por escolha gulosa e pela combinação de clusterização e escolha gulosa;

2. Comparar as características selecionadas pelo método de seleção proposto com um algoritmo de seleção univariada. Cada método de seleção escolherá um conjunto

de características que será utilizado para treinar um modelo de classificação implementado com um algoritmo de *Random Forest* (RF). O algoritmo RF é bastante utilizado em ferramentas de Biologia Molecular, devido aos modelos de boa acurácia que consegue construir com dados biológicos. Outra razão para a escolha do RF como algoritmo para treinar um modelo de teste é a possibilidade de visualizar um ranqueamento das características utilizadas para a criação do modelo de distinção. Dessa forma, é possível visualizar a importância de cada característica escolhida pelos dois métodos de seleção.

1.4 Descrição do trabalho

Este trabalho está dividido em seis capítulos. No capítulo 2, estão apresentados conceitos básicos de Biologia Molecular. No capítulo 3, fundamentos de Aprendizado de Máquina. No capítulo 4, são apresentados os dados utilizados no trabalho e a proposta de métodos para a escolha de características. No capítulo 5, são discutidos os resultados obtidos nos testes com o método de seleção proposto. Finalmente, no capítulo 6, estão apresentadas as conclusões do trabalho e sugestões de trabalhos futuros.

Capítulo 2

Biologia Molecular e Bioinformática

Este capítulo descreve conceitos básicos de Biologia Molecular e de Bioinformática necessários ao entendimento do trabalho. Na Seção 2.1, são apresentados os conceitos básicos de Biologia Molecular, tais como os conceitos de DNA e RNA. Na Seção 2.2, estão apresentados o Dogma Central da Biologia Molecular e o processo de síntese de proteínas. Na Seção 2.3, descrevemos os RNAs Não-Codificadores. Na Seção 2.4, estão descritos os lncRNAs, seus diferentes tipos e o estado da arte para predição de transcritos não-codificadores.

2.1 Conceitos Básicos de Biologia Molecular

Todo organismo existente conhecido possui uma característica em comum: ele armazena suas informações hereditárias e as informações de coordenação de funcionamento de seus processos biológicos em uma longa cadeia linear química denominada DNA [5]. Além desta, existem outras moléculas, os RNAs, que participam de diversas atividades celulares importantes [5].

2.1.1 DNA

A forma como o DNA é armazenado pode variar entre os organismos, como é o caso da diferença entre organismos procariontes e eucariontes. Entretanto, sua composição é feita com os mesmos elementos básicos.

O DNA é um longo polímero formado por moléculas de açúcar desoxirribose e fosfato intercalados e unidos por ligações fosfodiéster. Em cada molécula de açúcar está ligada uma base nitrogenada, podendo essa ser uma das quatro bases: Adenina, Timina, Citosina, ou Guanina (Figura 2.1). A composição das bases nitrogenadas com o grupo fosfato é chamada de nucleotídeo e cada nucleotídeo está conectado a outro nucleotídeo quimi-

camente complementar na fita oposta [5]. As bases complementares são Adenina/Timina e Citosina/Guanina.

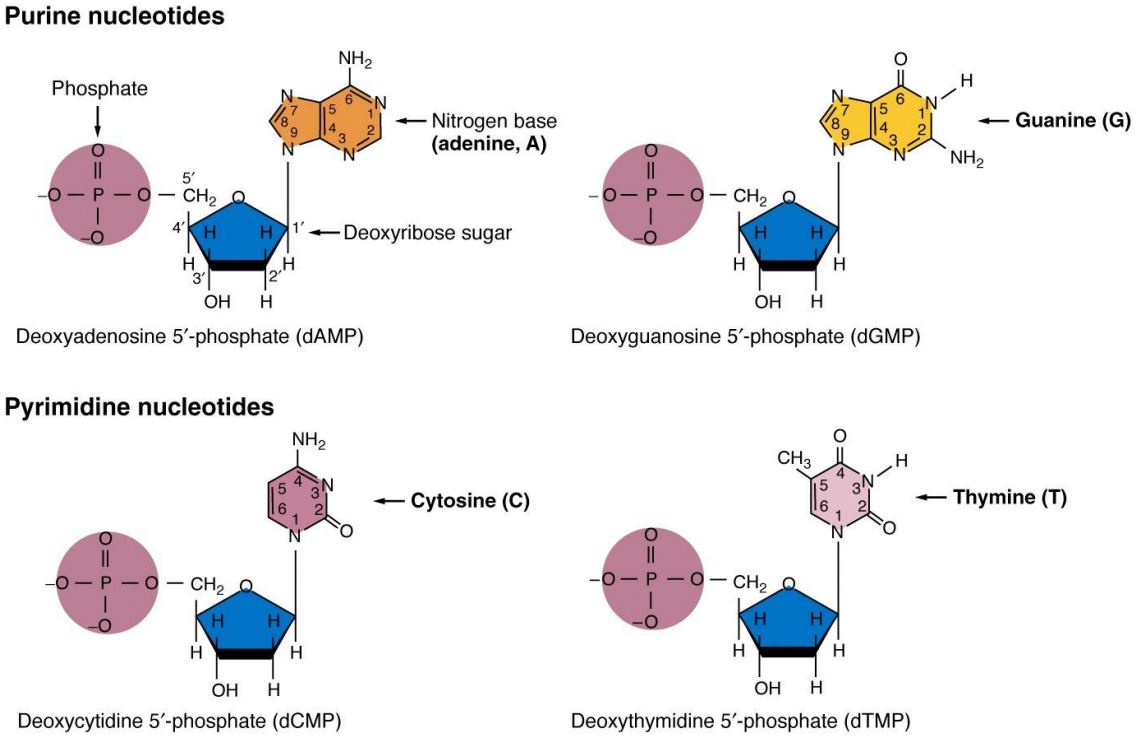

Figura 2.1: Bases nitrogenadas que compõem o DNA [16].

Essa conexão entre duas bases de nucleotídeos, denominada par de bases (bp), é feita por pontes de hidrogênio e é quimicamente mais fraca que a conexão com a cadeia de açúcar-fosfato. Essa característica é importante para os processos de replicação e transcrição do DNA. A composição desses diversos pares de bases que compõem o DNA forma uma estrutura de dupla hélice (veja Figura 2.2) que foi inicialmente apresentada por Watson e Crick [126].

Como todos os organismos possuem essa mesma estrutura fundamental, é possível, por exemplo, extrair um pedaço do DNA de uma bactéria e inseri-lo no DNA de uma célula humana, e a informação adicionada pode ser lida, interpretada e copiada pelos mesmos processos normais da célula [5]. Esse mecanismo de alteração do DNA é utilizado por retrovírus para se instalarem em um organismo, o que, em alguns casos, pode acarretar doenças como câncer [118].

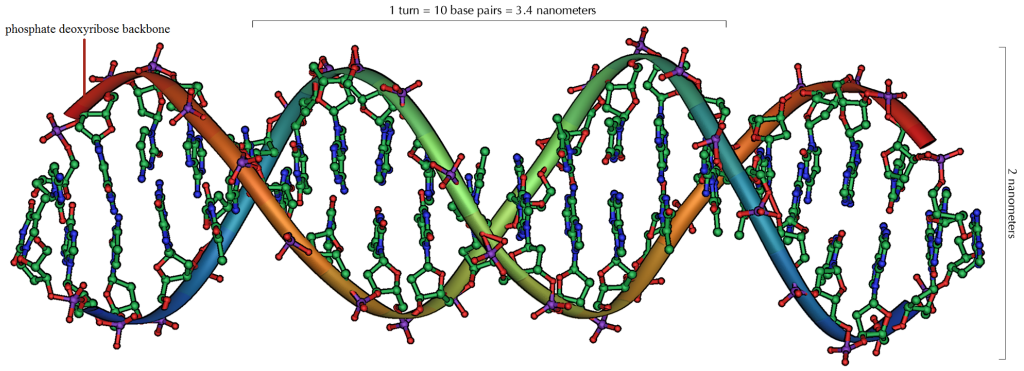

Figura 2.2: Estrutura básica em hélice do DNA com as duas fitas de açúcar desoxirribose e fosfato conectadas entre si pelos nucleotídeos com bases complementares. (Figura adaptada de [20]).

2.1.2 RNA

Além da função de replicação, que é o processo responsável por transmitir a informação hereditária do DNA para novas células, o DNA é responsável pela síntese de outras biomoléculas necessárias ao organismo. Esse processo inicia-se no DNA com uma polimerização denominada transcrição, no qual o DNA é utilizado como molde para a criação de moléculas menores chamadas de RNA [5].

O RNA difere-se do DNA em dois aspectos: na composição de sua cadeia e na estrutura da molécula. A composição do RNA também é feita por bases nitrogenadas, mas as bases que o compõem são: Adenina, Uracila, Citosina e Guanina. Em relação à estrutura, diferentemente do DNA, que possui duas fitas enroladas em dupla hélice, o RNA normalmente compõe-se de uma fita única de nucleotídeos dobrada em si mesma [128].

2.2 Dogma Central da Biologia Molecular

O Dogma Central da Biologia, proposto por Francis Crick em 1958 [126], consiste na afirmação de que o DNA codifica a produção de RNA por um processo denominado transcrição. Em seguida, o RNA codifica a produção de proteínas por tradução (Figura 2.3).

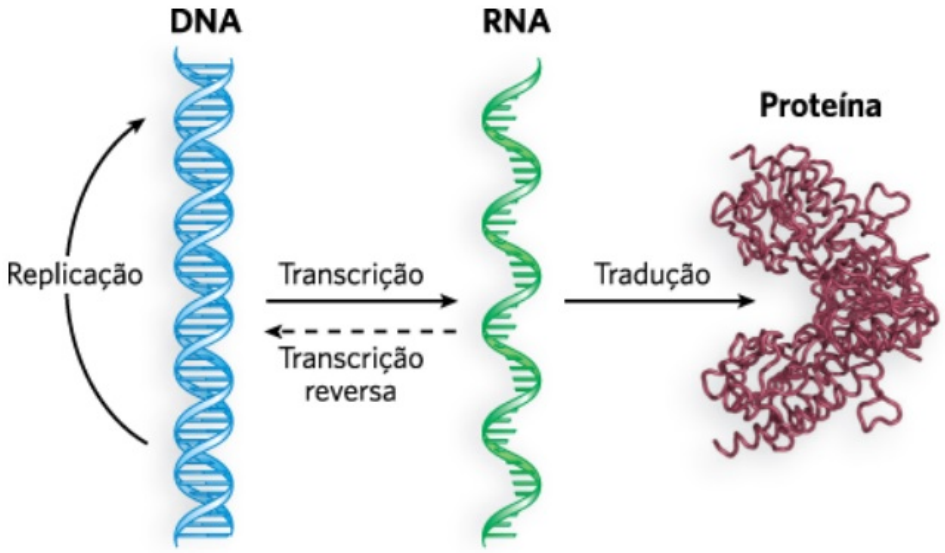

Figura 2.3: Dogma Central da Biologia Molecular: DNA → RNA → Proteína. A informação genética parte do DNA para o RNA por transcrição e do RNA para proteína por tradução. Existem casos nos quais a informação pode partir do RNA para o DNA por transcrição reversa; no entanto, não há evidência de informação transmitida a partir de proteínas para ácidos nucleicos [103].

Transcrição: DNA → RNA

O processo de transcrição começa com uma enzima chamada RNA-Polimerase em conjunto com outras proteínas denominadas fatores de transcrição. Uma vez que os fatores de transcrição conectam-se a regiões *enhancer* e *promoter* da sequência do DNA, a RNA-polimerase começa o processo de transcrição sintetizando uma molécula pré-mRNA a partir do complemento das bases presentes na fita original da molécula de DNA.

Posteriormente, este pré-mRNA é processado novamente, procedimento denominado *splicing*, para a remoção de íntrons e para a adição da cauda poli-A e do cap-5'. Assim, o pré-mRNA torna-se mRNA, RNA mensageiro, e pode sair do núcleo para atuar no processo de sintetização de proteínas, denominado de tradução.

Esse é o processo de transcrição em organismos eucariotos. Em organismos procariotos, o processo é menos complexo, pois o transcrito sofre pouca ou nenhuma alteração após sua síntese, não havendo, assim, um pré-mRNA. Em organismos procariotos, a tradução começa enquanto o mRNA ainda está sendo sintetizado [10].

Tradução: RNA → Proteína

A tradução é o último estágio de síntese de proteínas a partir da informação extraída do DNA. O mRNA resultante da transcrição e do *splicing* sai do núcleo para ser processado por outra organela, a qual lerá a informação da sequência de nucleotídeos do mRNA sequencialmente em grupos de três nucleotídeos, denominado códons.

Cada códon representa um aminoácido, que será ligado a outro aminoácido, formando uma cadeia polipeptídica que, então, será enrolada em si mesma, gerando uma proteína. Todo esse processo bioquímico ocorre nos ribossomos, organelas compostas de proteínas e RNA ribossômico (rRNA), com a interação de mais um tipo de RNA, o RNA transportador (tRNA). A Figura 2.4 mostra quais aminoácidos são formados a partir dos códons presentes no mRNA.

Standard genetic code

1st base	2nd base U		2nd base C		2nd base A		2nd base G		3rd base
U	UUU	(Phe/F) Phenylalanine	UCU	(Ser/S) Serine	UAU	(Tyr/Y) Tyrosine	UGU	(Cys/C) Cysteine	U
	UUC		UCC		UAC		UGC		C
	UUA	(Leu/L) Leucine	UCA		UAA	Stop (*Ochre*)	UGA	Stop (*Opal*)	A
	UUG		UCG		UAG	Stop (*Amber*)	UGG	(Trp/W) Tryptophan	G
C	CUU		CCU	(Pro/P) Proline	CAU	(His/H) Histidine	CGU	(Arg/R) Arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	(Ile/I) Isoleucine	ACU	(Thr/T) Threonine	AAU	(Asn/N) Asparagine	AGU	(Ser/S) Serine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	AUG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GUU	(Val/V) Valine	GCU	(Ala/A) Alanine	GAU	(Asp/D) Aspartic acid	GGU	(Gly/G) Glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GUG		GCG		GAG		GGG		G

nonpolar polar basic acidic (stop codon)

Figura 2.4: Tabela com o código genético degenerado. Dos 64 possíveis códons, 61 representam aminoácidos, e três representam sinais de parada de sintetização. O código genético é chamado de degenerado porque mais de um códon pode codificar um aminoácido (adaptada de [19]).

O tRNA é uma molécula de 76 a 90 nucleotídeos e possui uma função chave dentro do processo de transcrição. Ela traz os aminoácidos que vão compor a proteína. Sua estrutura secundária, forma espacial da molécula devido à interação de seus nucleotídeos

entre si [52], possui um anticódon que se conectará ao códon presente no mRNA. Assim, o tRNA adiciona um aminoácido conectado à sua sequência à cadeia de aminoácidos em formação no rRNA (Figura 2.5).

Nem todos os RNAs estão diretamente envolvidos na produção de proteínas. Existem diversos outros tipos de RNAs categorizados conforme suas funções dentro do organismo. Além dos RNAs codificadores, há também RNAs envolvidos na replicação do DNA [24], RNAs que atuam na modificação de outros RNAs pós-transcrição [44], RNAs que atuam em processos regulatórios do DNA [96], e até mesmo RNAs parasitas [38].

Neste trabalho, foi considerada, para fins de classificação, apenas a função desempenhada pelos RNAs com relação ao processo de síntese de proteínas, ou seja, como codificadores ou como não-codificadores de proteínas (ncRNAs).

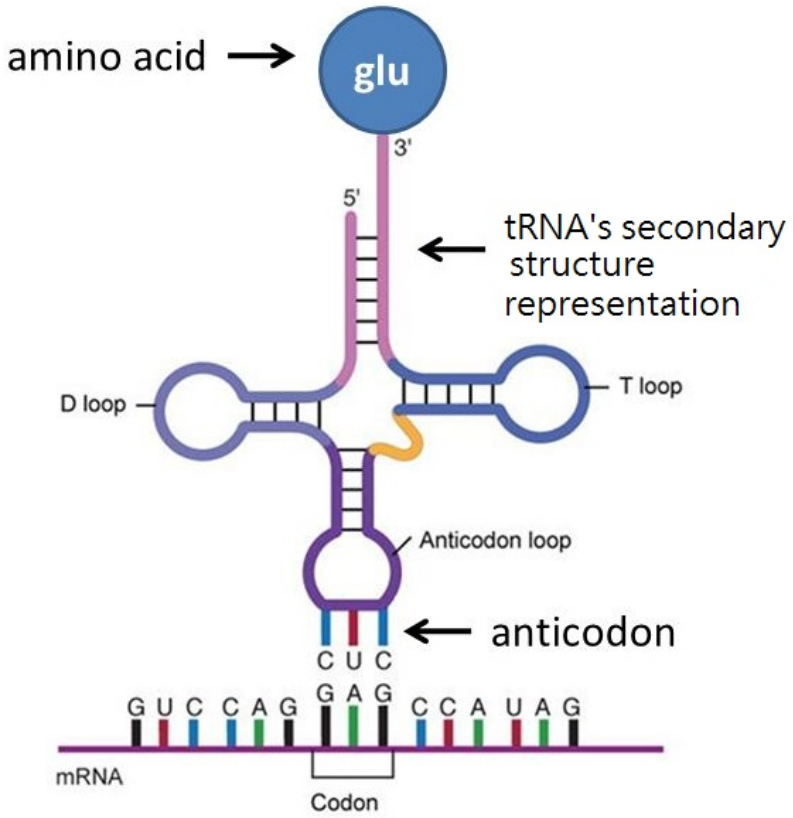

Figura 2.5: Interação entre tRNA e mRNA na síntese de proteínas [78].

2.3 RNAs Não-Codificadores - ncRNAs

Hoje, sabe-se que a maior parte do genoma humano é transcrito. No entanto, isso não implica que todo esse material genético codifica proteínas. Na verdade, menos de 2% do genoma humano encontra-se nas regiões codificadoras de proteínas [23].

Embora até há pouco tempo fossem considerados apenas como resíduos do processo de transcrição, os ncRNAs entraram em evidência por atuarem funcionalmente na célula. De fato, existem fortes evidências de que a complexidade do organismo está correlacionada com a complexidade e o tamanho dos transcritos não-codificadores e não com o tamanho do genoma, como se acreditava nos primórdios dos estudos de genes.

Em bactérias, por exemplo, 90% do genoma codifica proteínas; em leveduras essa quantidade diminui para 68%, em nematoides, 23%-24% e, em mamíferos, apenas 1,5%-2% do genoma codifica proteínas [105]. Por outro lado, se formos considerar apenas o tamanho do genoma, seres "menos complexos", como anfíbios, possuem genomas muito maiores que o genoma humano (Figura 2.6) [41].

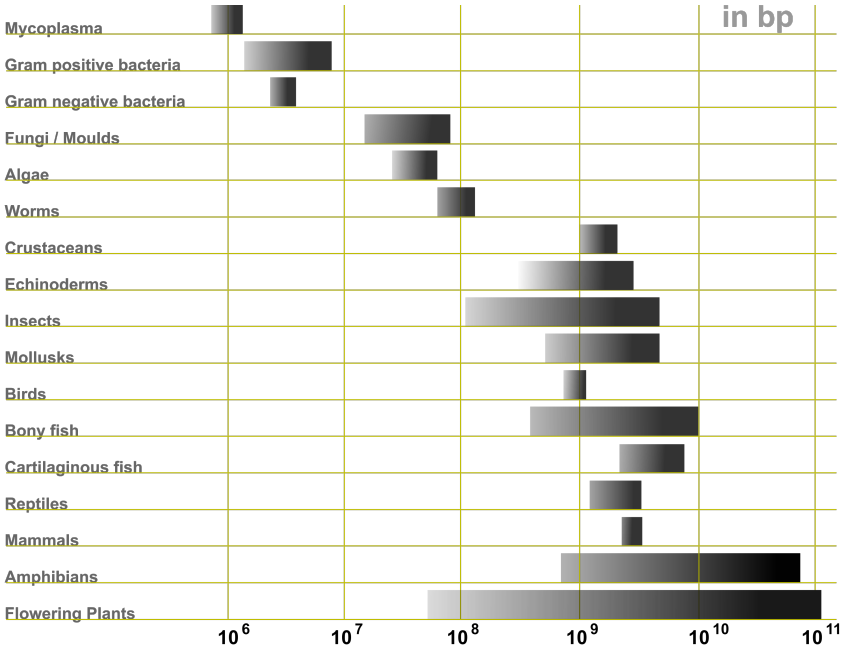

Figura 2.6: Comparação do tamanho do genoma de diversas espécies [21].

Devido ao reduzido custo de obtenção de informação dos sequenciadores NGS como o RNA-Seq [125], atualmente existem bancos de dados com centenas de milhares de informações genéticas de transcritos de diferentes espécies, alguns especializados apenas em transcritos não-codificadores de proteínas. Como exemplo, podemos citar: Ensembl [4], RefSeq [91], lncRNAdb [6] e NONCODE [13], dentre outros.

Essa grande quantidade de moléculas não codificadoras e a interação dessas com as codificadoras de proteínas são problemas em aberto da Biologia Molecular. Um melhor entendimento do funcionamento dessas moléculas pode levar a uma melhor compreensão do funcionamento de organismos com estruturas biológicas mais complexas e eventuais anomalias, que podem resultar em doenças.

Para organizar os estudos, os ncRNAs podem ser inicialmente categorizadas em dois tipos principais: os que atuam na infraestrutura da célula e os que atuam de maneira regulatória nos genes. Os que atuam na infraestrutura da célula têm funções no processo de tradução ou no processo de *splicing*, por exemplo, os tRNAS e rRNAS, já descritas. Os que atuam no processo regulatório interagem com outros RNAs e afetam a quantidade de transcritos que são produzidos, por exemplo. Dentre os mais estudados, podemos citar ncRNAs pequenos como MicroRNAs (miRNAs), Piwi-interacting RNAs (piRNAs), e Small interfering RNAs (siRNAs), descritos em seguida, e os lncRNAs descritos na Seção 2.4.

MicroRNAs (miRNAs)

Os miRNAS possuem aproximadamente 22 nucleotídeos e surgem a partir de transcritos com uma estrutura secundária de um tipo específico de dobradura *hairpin*. Esses transcritos são chamados de pré-miRNAs. Após sairem do núcleo e serem transformados em miRNA [83], esses ncRNAs integram, junto com outras proteínas, um complexo de regulação pós-transcricional denominado *RNA-induced silencing complex* (RISC).

Essa regulação ocorre com os miRNAs conectando-se a sequências complementares dos mRNAs, para causar a desestabilização e degradação desses, adequando, dessa forma, a quantidade de transcritos gerados pela célula. Os miRNAS podem ser conectados a diversos mRNAS diferentes e cada mRNA também pode ser inibido por vários miRNAs diferentes.

Piwi-interacting RNAs (piRNAs)

Os piRNAs são ncRNAs de 24 a 31 nucleotídeos, mas de estrutura secundária ainda não muito conhecida [5]. Esses ncRNAs formam complexos com proteínas Argonauta da subfamília Piwi e são bastante comuns em células germinativas. Sua função principal é o silenciamento epigenético de transposons, elementos de transposição, dessa maneira controlando mutações no genoma [132].

Small interfering RNAs (siRNAs)

Os siRNAs possuem de 20 a 24 nucleotídeos e são gerados pela ação da enzima Dicer em RNAs de dupla fita, dsRNAs. Assim como os miRNAs, os siRNAs atuam dentro do RISC silenciando alguns transcritos codificadores de proteínas se conectando a sequências complementares às das siRNAs, processo esse denominado RNA interference (RNAi) [127].

2.4 LncRNA

Atualmente, a maior categoria de ncRNAs, e também a menos conhecida, é a dos lncRNAs. Trata-se de uma categoria bastante ampla, que agrega todos os ncRNAs com mais de 200 nucleotídeos, com baixa ou nenhuma capacidade para produção de proteínas e que também está sujeita a modificações pós-transcricionais, tais como *splicing*. Os lncRNAs estão subdivididos em tipos de acordo com sua localização genômica relativa à proximidade de genes codificadores, uma vez que ainda pouco se sabe da função de muitos deles no organismo. Uma observação em relação a essa organização é que ela não fornece informação a respeito da origem evolutiva dessas moléculas [63].

2.4.1 Tipos de lncRNAs

Na Figura 2.7 estão apresentados alguns dos tipos mais comuns de lncRNAs: os *long intergenic* RNAs (lincRNAs), que se localizam entre genes codificadores de proteínas; os *intronic* lncRNAs estão localizados nos íntrons de genes codificadores; os *bidirectional* lncRNAs estão localizados na fita oposta à de genes codificadores, nas quais o ponto onde o processo de transcrição se inicia está a uma distância inferior a 1000 bp; os *enhancer* lncRNAs são os transcritos não codificadores longos que são transcritos a partir da região *enhancer* da sequência, região à qual se podem ligar proteínas que aumentam os níveis de transcrição; os *sense* lncRNAs são transcritos da fita *sense* dos genes codificadores e contêm éxons de genes codificadores, podendo até mesmo conter toda uma sequência de genes codificadores em sua sequência [73]; os *antisense* lncRNAs, por outro lado, são transcritos da fita oposta à de genes codificadores.

Existem diversas razões que justificam a dificuldade de se compreender a função dos lncRNAs. Um das dificuldades é devido à baixa conservação dessas moléculas nos diferentes organismos [57]. Além disso, as moléculas de DNA são suscetíveis a mutações em sua sequência de nucleotídeos. Essas mutações podem ser ocasionadas espontaneamente ou por agentes mutagênicos e são parte importante em processos biológicos como desenvolvimento de sistema imunológico [56], câncer [35, 90] e evolução [12].

No entanto, na parte codificadora de proteínas do genoma, de onde são transcritos os RNAs codificadores, algumas mutações podem acarretar a mudança da proteína sintetizada em uma etapa posterior, no processo de tradução. Essas mudanças na composição das proteínas podem ser fatais para o organismo, pois esse se tornaria incapaz de produzir alguma proteína necessária para a manutenção de seus processos biológicos. Assim, essas mutações tendem a não ser transmitidas para as próximas gerações.

Por outro lado, a parte que não codifica proteínas possui uma flexibilidade muito maior para mudança da sua composição primária. No entanto, a falta de conservação não implica

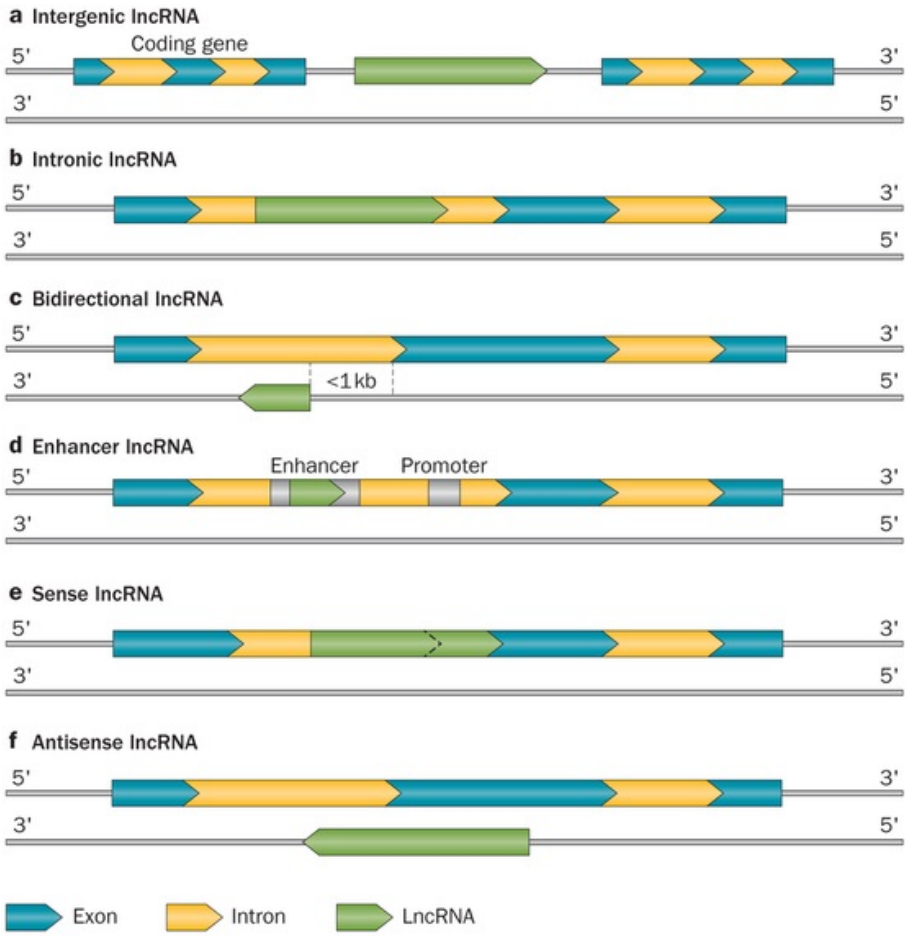

Figura 2.7: Tipos mais comuns de lncRNA classificados em relação a suas posições no genoma [28].

falta de função dos lncRNAs. Se as estruturas secundárias forem preservadas, também será mantida a função reguladora dos lncRNAs, que ocorre junto aos genes codificadores de proteínas [57].

Os lncRNAs atuam em diversos processos epigenéticos dentro da célula, mas resultados recentes demonstram que, apesar dessa diversidade de tarefas, essa atuação ocorre dentro de um mesmo modelo de ações. Um exemplo dessa atuação é a relação entre os lncRNAs H19 e Xist com o processo de *imprinting* genômico e inativação do cromossomo X que ocorre em mamíferos [89]. Apesar das evidências que apontam que essas duas moléculas estão envolvidas nesse processo, a exata forma de atuação do H19 no processo ainda é objeto de estudos.

Um fato interessante a se notar a respeito dos lncRNAs é que algumas pesquisas demonstram que existe uma conservação de função da regulação genética de diferentes lncRNAs em diferentes espécies, apesar da baixa conservação de sua estrutura primária.

Por exemplo, recentemente foi demonstrado que a determinação de sexo em *Drosophila* também é regulada por um lncRNA de nome Sxl, mas a forma de atuação no processo é bastante diferente de como ocorre em mamíferos [80].

Diversas doenças genéticas estão relacionadas com a expressão errática de ncRNAs, razão do crescente interesse por pesquisas relacionadas a essas biomoléculas [30].

Já havia evidência há bastante tempo da correlação entre os níveis de expressão de lncRNAs e diversas doenças genéticas, tais como o câncer. O mesmo lncRNA H19 que atua no processo de *imprinting* no cromossomo X também possui funções de supressão de tumores [131, 49]. Em experimentos com camundongos, foi verificado que esses eram muito mais suscetíveis a tumores e carcinomas quando não possuíam o H19. Por outro lado, é possível encontrar um grau elevado de expressão de H19 em diversos tipos de câncer [50]. Em outras palavras, existem evidências opostas que o lncRNA pode ser tanto oncogênico quanto supressor de tumores. Enquanto a função biológica deste lncRNA não for completamente compreendida, não há como se ter certeza se a molécula possui duas funções distintas, ou se sua função depende de um contexto externo [42].

Outro exemplo de lncRNA relacionado com câncer é o HOX antisense intergenic RNA (HOTAIR). Localizado no locus HOXC no cromossomo 12q13.13 em mamíferos [97], este transcrito com 2,2 kb possui uma atuação determinante no comportamento de metástase em câncer de mama, com um aumento de 2.000 vezes da expressão deste lncRNA em amostras com câncer [46].

Esses são apenas alguns exemplos de pesquisas correlacionando diretamente lncRNAs com doenças genéticas. Com os avanços das técnicas e tecnologias envolvidas no *profiling* do transcritoma do câncer, hoje existem bancos de dados disponíveis especializados em correlacionar doenças com lncRNAs. O lncRNADisease [18] é um exemplo de um desses repositórios de dados online.

2.4.2 Estado da Arte

Atualmente, não há ferramenta computacional amplamente aceita para a predição de lncRNAs, mas há algumas ferramentas que foram desenvolvidas ao longo dos anos para a predição de RNAs não-codificadores. As características e algoritmos para a construção dos modelos de classificação variam pouco, visto que o conhecimento a respeito dessas moléculas, suas funções e suas estruturas secundárias ainda é muito incipiente.

O *Coding Potential Calculator* (CPC), é uma ferramenta precisa e rápida desenvolvida em 2007 utilizada para calcular o potencial que um dado transcrito possui de ser codificador de proteína [62]. O CPC utiliza um modelo de aprendizado supervisionado denominado *Support Vector Machine* (SVM) [25], configurado um *kernel* radial padrão. As seis características utilizadas para o treinamento da ferramenta foram extraídas da pes-

quisa com o BLASTx [43]; duas dessas características estão relacionadas com o tamanho e a qualidade das *Open Reading Frame* (ORF).

O PORTRAIT é uma ferramenta, publicada em 2009, desenvolvida para análise de ncRNA verificando o potencial de codificação de proteína das amostras por meio de SVM. Dois modelos de SVM são utilizados: se uma proteína relacionada ao transcrito for encontrada, utiliza-se uma SVM treinada com informações relacionadas a proteínas; caso contrário, utiliza-se um modelo de SVM treinado sem as informações relacionadas a proteínas. O PORTRAIT não utiliza características relacionadas a homologia em seus modelos [7].

O *Coding-Potential Assessment Tool* (CPAT), desenvolvido em 2013, é uma ferramenta que utiliza um modelo baseado em regressão logística [124]. Entre as características utilizadas estão a qualidade da ORF, a pontuação de Fickett e pontuação de *hexamer* das amostras. A função da pontuação de Fickett é medir diferenças de frequência e distribuição da posição dos códons na amostra. Já a pontuação de *hexamer* utiliza informação de um *bias* de quais aminoácidos são mais comuns de estarem adjacentes a cada códon para formar proteínas.

O *Coding-Non-Coding Index* (CNCI), também desenvolvido em 2013, é um classificador baseado em SVM com *kernel* radial para diferenciar transcritos codificadores de não codificadores [110]. A ideia principal é explorar a distribuição desigual das trincas de nucleotídeos adjacentes das sequências. A ferramenta constrói uma matriz para avaliar as amostras. O conceito é similar à pontuação de *hexamer* utilizada pelo CPAT, mas uma análise mais abrangente é feita para classificar transcritos incompletos.

O iSeeRNA [108], publicado em 2013, é uma ferramenta baseada em SVM com kernel radial para a distinção de lincRNAs de PCTs. A ferramenta utiliza 10 características para treinamento do modelo: uma característica de conservação do transcrito, duas características relacionadas a ORF (tamanho total e proporção do tamanho da ORF pelo tamanho do transcrito) e sete características de frequência de di- e tri-nucleotídeos. Os resultados publicados da ferramenta demonstram que o iSeeRNA obteve uma acurácia de 96,1% na identificação de lincRNAs e 94,5% em PCTs no *dataset* de teste de *H. sapiens* utilizado. No *dataset* de *M. musculus* o desempenho foi de 94,2% de acurácia para a identificação de lincRNAs e 92,7% para PCTs.

O PLEK é uma ferramenta *alignment-free* para classificar transcritos em lncRNAs e mRNAs por meio de um esquema de melhoramento nas características de k-mer de tamanhos 1 a 5 e um classificador SVM. Publicado em 2014. O artigo que descreve a ferramenta apresenta como resultado acurácia acima de 95.6% em testes com *dataset* de *Homo sapiens* com mRNA obtidos do RefSeq e lncRNAs obtidos do GENCODE [66]. A ferramenta também é apresentada como sendo oito vezes mais rápida que a CNCI e 244 vezes mais rápida que a CPC executando no modo *single thread*.

O lncRNA-MFDL, publicado em 2015, é um classificador para identificar lncRNAs que utiliza algoritmos de *Deep Learning* aplicados a uma combinação (fusão) de características relacionadas a *k*-mer, *Open Reading Frame* (ORF), estrutura secundária e a sequência codificadora mais provável de um transcrito [33]. Utilizando um *dataset* de transcritos de *Homo sapiens*, a ferramenta obteve uma acurácia de 97.1%, o que, segundo os autores, é uma melhora bastante significativa quando comparada com as ferramentas CPC e CNCI.

O lncRScan-SVM é outra ferramenta que utiliza o SVM como classificador para distinguir PCTs de lncRNAs [109]. Publicado em 2015, a ferramenta utiliza três categorias de características. A primeira categoria são 14 atributos de frequência de tri-nucleotídeos: ACG, CCG, CGA, CGC, CGG, CGT, CTA, GCG, GGG, GTA, TAA, TAC, TAG e TCG, um desvio padrão entre três frames de *stop* códons (TAG, TAA e TGA) e o conteúdo GC. A segunda categoria são características calculadas pela ferramenta txCdsPredict da UCSC: txCdsPredict score, CDS *length* e CDS *percentage*. A terceira categoria de características foram extraídas da estrutura do gene do transcrito: tamanho do transcrito, contagem de exons e o tamanho médio dos exons.

O DeepLNC, publicado em dezembro de 2016, utiliza redes neurais profundas como solução para identificar lncRNAs. A solução, desenvolvida por Tripathi et al. [116], utiliza dados dos datasets LNCipedia e Ref-Seq e, segundo resultados publicados, atinge uma acurácia de 98,07%, sensibilidade de 98,8% e especificidade de 97,19% no *dataset* de teste. As características utilizadas para a construção do classificador foram baseadas na Entropia de Shannon aplicadas aos *k*-mers de tamanho 2,3,4 e 5 e as melhores características foram escolhidas com o *Forward Selection-Backward Elimination* (FSBE) [111], algoritmo de seleção de características da família do *Sequential Feature Selection* (SFS).

O FEELnc é uma solução publicada em janeiro de 2017 que utiliza um modelo treinado com Random Forest para a anotação de transcritos lncRNAs [130]. O FEELnc utiliza como características de treinamento do modelo uma definição mais flexível de ORF e de escores derivados de k-mers de tamanhos variados (k = 1, 2, 3, 6, 9 e 12). O FEELnc foi desenvolvido com dados de 20 amostras de *Canis lupus familiaris* do RNA-seq produzidos pelo consórcio europeu LUPA. O FEELnc provê além da classificação dos transcritos, a possibilidade de anotação de lncRNAS. Segundo o artigo publicado que apresenta a ferramenta, o FEELnc possibilitou a anotação de 10.374 novos lncRNAs e 58.640 mRNAs no genoma canino.

Algumas ferramentas, como o PhyloCSF [67] de 2011, que foram concebidas originalmente para predizer ncRNAs, podem ser utilizadas para ajudar na predição de lncRNAs. O PhyloCSF é um método que verifica características evolutivas em regiões codificadoras conservadas de forma a indicar se uma dada sequência de nucleotídeos, ou os sinônimos dessas sequências, são preservados em diversas espécies.

Em outros trabalhos mais recentes publicados em 2017, como em Schneider et al. [101], ocorre uma etapa de escolha das características mais relevantes para os métodos de aprendizado de uma SVM, usando a técnica de extração de características denominada Análise de Componentes Principais-*Principal Componente Analysis* (PCA). Apenas as características consideradas mais relevantes, de acordo com o PCA, são usadas para criar o modelo de distinção entre lncRNAs e PCTs.

Em Ventola et al. [120], também é proposta a combinação de informações da estrutura primária com informações de conservação, além de outras, para compor grupos de características, denominados de assinaturas, que são então utilizados para treinar os modelos de classificação.

Até onde sabemos, são poucos os trabalhos tratando de analisar sistematicamente as características das estruturas primárias das sequências de maneira isolada. Muitas das ferramentas citadas utilizam informações obtidas de uma análise anterior das sequências por outras ferramentas, como o BLASTx [43], ou do conhecimento da composição de proteínas, ou informações relacionadas a homologia para classificar as amostras entre codificadoras e não codificadoras.

Capítulo 3

Aprendizado de Máquina

Neste capítulo são descritos conceitos básicos de Aprendizado de Máquina, sendo discutidos com mais detalhes os métodos e modelos usados neste trabalho. Na Seção 3.1, apresentamos o conceito de Aprendizado de Máquina. Na Seção 3.2, descrevemos os conceitos de Aprendizado Supervisionado e apresentamos alguns exemplos de algoritmos. Na Seção 3.3, estão apresentados os conceitos de Aprendizado não Supervisionado e alguns exemplos de algoritmos utilizados neste trabalho. Na Seção 3.4, apresentamos o conceito de Maldição de Dimensionalidade e descrevemos os tipos de técnicas para redução de dimensionalidade.

3.1 Aspectos Gerais

Aprendizado de máquina é o nome que se dá à subárea da Inteligência Artificial em que um agente pode aprender e se aperfeiçoar a partir dos próprios dados que está processando, ou seja, sem ser explicitamente programado usando parâmetros necessários que melhorem seu desempenho no processamento dos dados [81].

Em uma definição mais formal de Mitchel [79], um programa de computador é considerado capaz de aprender a partir da experiência E, a tratar uma determinada classe de tarefas T com uma medida de desempenho P, se o desempenho nas tarefas de T, medidas por P, melhora com o aumento da experiência E.

Essa definição é interessante, pois define aprendizado de máquina de forma concreta, ou seja, em resultados práticos a serem obtidos, em vez de termos cognitivos, como um ser humano resolveria o problema.

Os problemas tratados por aprendizado de máquina podem ser categorizados entre os que necessitam de uma abordagem de aprendizado supervisionado e os que necessitam de uma abordagem de aprendizado não supervisionado. Além desses dois tipos de aprendizado, há também métodos de aprendizado semi-supervisionado [17] e aprendizado por

reforço [112]. No entanto, como esses dois métodos não foram utilizados no desenvolvimento deste trabalho, os detalhes de funcionamento e implementação destes dois métodos não estão descritos nesta dissertação.

De forma genérica, o Aprendizado Supervisionado é utilizado quando se tem um conhecimento *a priori* de que determinadas amostras pertencem a determinadas classes, e, com isso, queremos construir um modelo que classifique novas amostras em alguma das classes conhecidas [55].

Por outro lado, no Aprendizado Não-Supervisionado, não se sabe a qual classe as amostras pertencem. Logo, analisam-se as características das amostras entre si e se propõem grupos ou *Clusters* de amostras semelhantes [55].

3.2 Aprendizado Supervisionado

O processo de aprendizado supervisionado ocorre quando as amostras de dados possuem um rótulo, ou classe, associado a elas. Ou seja, para cada elemento de um conjunto de dados com p características, $\mathbf{x} = \{x_1, x_2, ..., x_p\}$, existe um valor associado de saída Y descrito por uma função fixa e não conhecida, conforme apresentado na Equação: 3.1:

$$Y = f(\mathbf{x}) + \epsilon \tag{3.1}$$

A tarefa a ser executada por um modelo supervisionado é tentar estimar uma função $\hat{f}$ que obtenha valores $\hat{Y}$ bastante próximos a Y, mesmo que $\hat{f}$ não se assemelhe à função f real.

Essa obtenção de $\hat{f}$ é realizada de maneira iterativa utilizando os próprios dados, representados por suas características e valores de saída, e uma medida de desempenho relativo ao erro entre Y, já conhecido das amostras, e $\hat{Y}$ estimado por $\hat{f}$.

De fato, o núcleo dos programas de aprendizado supervisionado é buscar parâmetros Θ da função $\hat{f}$ que serão aplicados às características $\mathbf{x} = \{x_1, x_2, ..., x_p\}$ de forma a minimizar o erro de predição E, conforme apresentado na Equação 3.2.

$$\underset{\Theta}{\operatorname{argmin}}\, E[f(\mathbf{x}) + \epsilon - \hat{f}(\mathbf{x})] \tag{3.2}$$

Existem diversos tipos de função de erro que podem ser utilizados no processo de treinamento do modelo. O desempenho dessas funções está fortemente relacionado com o tipo de problema que está sendo estudado e com o algoritmo que está sendo utilizado.

O resultado final desse processo de minimização é denominado de Modelo e é a parte do programa que é utilizada para processar novos dados e predizer os valores $\hat{Y}$ de novos dados até então não observados.

De maneira geral, quando o Y associado às amostras são valores quantitativos, dizemos que se trata de um problema de Regressão; quando os Y são valores qualitativos, é um problema de Classificação. Apesar de os algoritmos de obtenção dos modelos que resolvem problemas de Regressão e Classificação serem ligeiramente diferentes, o processo de minimização ilustrado na Equação 3.2 para a obtenção do modelo ótimo é o mesmo.

Os modelos obtidos por aprendizado supervisionado também podem ser utilizados para processos de inferência, nos quais se quer compreender a relação entre determinadas características das amostras e o valor de saída $\hat{Y}$ associado. Uma vez que o valor de saída das amostras de treinamento é conhecido, é possível, por exemplo, alterar de forma controlada os valores de um subconjunto de características que se queira estudar e observar o impacto na predição obtida pelo modelo.

3.2.1 Aprendizado em *Ensemble*

Dentro da categoria de aprendizado supervisionado, há uma subcategoria denominada aprendizado em *Ensemble*. O conceito de *Ensemble* diz que, quando múltiplos algoritmos simples de aprendizado são combinados, é possível obter um desempenho semelhante, ou até mesmo superior, ao desempenho de classificadores mais complexos.

Uma indicação empírica dos bons resultados dos métodos *Ensemble* pode ser atestada nos resultados de competições de *Data Mining*, nas quais existe uma variedade grande de tarefas, como detecção de intrusão em rede ou gerenciamento de relacionamento de clientes. A maior parte dos modelos vencedores utiliza alguma forma de *Ensemble* em sua implementação [133].

De acordo com Dietterich [29], existem três razões intuitivas para o bom desempenho de métodos *Ensemble*: Estatística, Computacional e Representativa (Figura 3.1).

Do ponto de vista estatístico, se o algoritmo de aprendizado for considerado com uma implementação de um processo de busca pelo melhor modelo de classificação em um espaço de busca S, e se a quantidade de dados de treinamento for pequena quando comparadas ao espaço de busca, a busca pode retornar diversos classificadores de S que possuem a mesma acurácia nos dados de treinamento, mas, de maneira geral, possuem um desempenho inferior a um classificador ótimo. Ao combinar diversos classificadores no espaço de busca, na média, é possível aproximar-se do desempenho de um classificador ótimo que esteja contido no espaço de busca S.

No aspecto computacional, a configuração dos dados utilizados pela busca, por exemplo a ordem de processamento das amostras, pode induzir o algoritmo a ficar preso na seleção de um modelo classificador ótimo local, cercado de modelos de desempenho ruim. Entretanto, esse classificador ótimo local pode ter um desempenho inferior a um classificador ótimo global que não pode ser alcançado a partir do ponto ótimo local. Ao utilizarmos

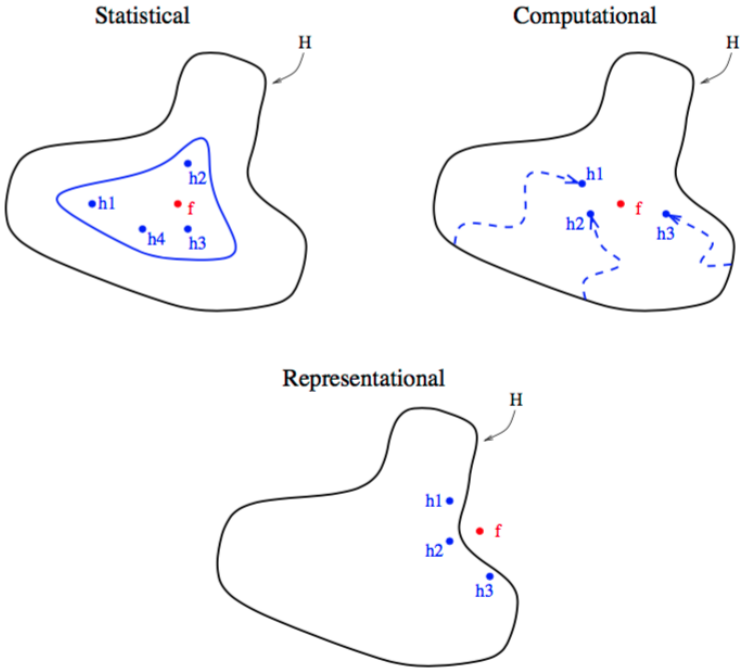

Figura 3.1: Representação visual da três razões apresentadas por Dietterich pelas quais um *Ensemble* de classificadores possui um bom desempenho [29].

diversos classificadores, com configurações de dados distintas, é possível minimizar esse risco de não encontrar o classificador ótimo global.

Por último, a razão representativa é um desdobramento da razão estatística mencionada anteriormente. Ao trabalhar com dados que devem ser representados em recursos computacionais finitos, é possível que o melhor modelo classificador não esteja presente no espaço de busca S, restrito pelo subconjunto de dados utilizados. Ainda assim, ao combinar diversos modelos da borda do espaço de busca próximos ao modelo ótimo, o desempenho da combinação desses classificadores será melhor do que qualquer classificador isolado.

Existem, entretanto, duas condições necessárias e suficientes para que o *Ensemble* possua um bom desempenho. Os classificadores integrantes do *Ensemble* devem ser precisos e, também, diversos [48]. Um classificador é considerado preciso quando sua taxa de erro é melhor que a taxa de erro de uma predição aleatória para novas amostras. E dois classificadores são considerados diversos se eles erram de maneira distinta em suas predições.

De maneira mais formal, pode-se considerar que, se a taxa de erro de cada um dos L classificadores h_L for $p < 0,5$ (escolha aleatória) e os erros forem independentes entre si, então a probabilidade que a maioria dos classificadores estará errada é a área sob a

distribuição binomial na qual mais que $\frac{L}{2}$ hipóteses estão erradas (Equação 3.3) [29]:

$$\sum_{i=\frac{L}{2}}^{L} \binom{L}{i} \varepsilon^i (1-\varepsilon)^{L-i} = \varepsilon_{ensemble} \tag{3.3}$$

A Figura 3.2 mostra um *Ensemble* simulado apresentado no artigo de Dietterich [29] com 21 classificadores, cada classificador com uma taxa de erro de $0,3$. A área sob a curva que indica a probabilidade de que 11 ou mais classificadores estejam errados é de $0,026$, ou seja, bastante inferior à taxa de erro de cada classificador integrante do *Ensemble*.

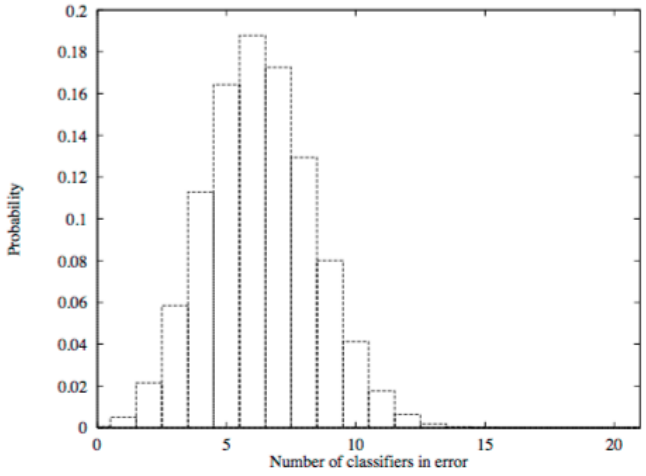

Figura 3.2: Probabilidade de erro de classificação em razão do número de classificadores [29].

3.2.2 Tipos de *Ensemble*

Existem diversos tipos de técnicas de *Ensemble*. As mais comuns entre essas são *Bootstrap Aggregating* (*Bagging*), *Boosting* e *Stacking*.

Em *Bagging*, cada modelo no *Ensemble* tem um mesmo peso no resultado, como em uma eleição. A variância dos modelos é garantida treinando cada modelo com um subconjunto aleatório dos dados de treinamento. Random Forest é um exemplo de algoritmo que implementa *Bagging* e que é bastante utilizado pelos bons resultados que consegue alcançar [11].

Boosting, por outro lado, iterativamente muda os pesos das amostras de treinamento e os pesos dos classificadores que integram o *Ensemble* durante o estágio de treinamento do modelo. A ideia é dar uma importância maior aos classificadores que predizem correta-

mente as amostras mais difíceis. Adaboost é um dos algoritmos pioneiros que implementa essa técnica de *Boosting* [39].

Stacking é a técnica que combina as predições de diversos outros algoritmos de aprendizado em um estágio posterior. O *Input* do *Stacking* é o *Output* de outros classificadores que são então combinados com outro algoritmo de aprendizado, *Logistic Regression*, por exemplo, para obter uma predição mais precisa [129].

3.2.3 Random Forest

O algoritmo Random Forest é um método de aprendizado em *Ensemble* conhecido e bastante utilizado para tarefas de classificação e regressão [51]. Este algoritmo utiliza múltiplas Árvores de Decisão durante o processo de treinamento e retorna a moda dos resultados das árvores de decisão que integram o *Ensemble* para tarefas de classificação ou a média das saídas para tarefas de regressão. Dessa forma, o *Ensemble* consegue minimizar *overfitting* com os dados de treinamento, problema comum nos métodos de Árvore de Decisão, principalmente quando as árvores geradas utilizam muitas características [40].

O funcionamento do algoritmo, como já mencionado, baseia-se em *Bagging* de Árvores de Decisão pouco profundas, ou seja, de um que utiliza poucas características. Dado um conjunto de treinamento $X = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ definido por um conjunto de características $F = f_1, f_2, ..., f_m$ e as respostas associadas $y = y_1, y_2, ..., y_n$, o algoritmo cria B diferentes árvores utilizando aleatoriamente amostras subconjunto randômico de X,F e y.

Após a fase de treinamento do *Ensemble*, novos dados x' são classificados pela combinação das árvores geradas durante o treinamento (Equação 3.4):

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \tag{3.4}$$

O fato de o algoritmo Random Forest também escolher aleatoriamente um subconjunto de características, processo também conhecido como *Feature Bagging*, diminui a possibilidade da criação de árvores fortemente correlacionadas, caso haja uma ou mais características que sejam fortes preditores para a resposta associada y. Nesses casos, essas caraterísticas seriam bastante utilizadas por todas as árvores, e, ao criar árvores correlacionadas, a tendência é que haja *overfitting* nos dados de treinamento.

3.3 Aprendizado Não-Supervisionado

Por outro lado, no processo de aprendizado não-supervisionado, as amostras $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ ou não possuem um valor de saída, ou rótulo, associado a elas, ou não se deseja utilizar seus rótulos para a construção de um modelo de associação ou agrupamento.

Sendo assim, os métodos aplicados buscam encontrar padrões para agrupar amostras semelhantes entre si em grupos distintos. Um tipo comum desses modelos de agrupamento é denominado modelos de *clustering*. Esse modelos podem ser utilizados tanto para fornecer uma visão geral dos dados, quanto para extrair características inicialmente desconhecidas em modelos supervisionados para Regressão ou Classificação.

Os modelos não-supervisionados não são tecnicamente mais complexos que os modelos de aprendizado supervisionado, mas a interpretação dos resultados dos modelos pode exigir conhecimento do domínio ao qual o modelo está sendo aplicado, sobretudo quando se trabalha com dados de alta dimensionalidade nos quais o relacionamento dessas características não é claro. Em geral, esses modelos assemelham-se bastante a problemas estatísticos de estimativa de densidade [117].

Existem diversos métodos para a construção de modelos não-supervisionados. Dentre os mais comuns, podemos citar: K-Means [74], K-medoids [58], [31] e Mixture Models para clusterização de dados [77]; SVM de classe única [102] e Isolation Forests [69] para detecção de anomalias; e Principal Component Analysis [1], Independent Component Analysis [22] para problemas de redução de dimensionalidade.

K-means

O algoritmo K-means é bastante utilizado em mineração de dados para análise de agrupamentos, ou *clusters*, nos dados. A ideia principal do algoritmo é dividir as n amostras de dados em k *clusters* de forma que cada amostra passa a pertencer ao *cluster* em relação ao qual a distância dessa amostra até o centro do *cluster* é a menor. A Figura 3.3 ilustra as iterações do processo com diagramas de Voronoi [8]). O diagrama de Voronoi (ou Tesselação de Voronoi) é uma técnica de visualização de *clusters* que também pode ser utilizado para auxiliar o processo de escolha inicial dos centros dos *clusters*, segundo trabalho publicado por Reddy et al. [95].

O algoritmo utilizado por K-means, também conhecido por algoritmo de Lloyd [71], funciona da seguinte forma. Dado um conjunto de amostras $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, no qual cada amostra é um vetor d-dimensional de valores reais, o algoritmo busca iterativamente particionar as n amostras em k *clusters* $S = \{S_1, S_2, ...S_k\}$ de forma a minimizar a soma do quadrado das distâncias entre os pontos de cada *cluster* com o respectivo centro do *cluster* (equação 3.5). Em seguida as médias das amostras de cada *cluster* são recalculadas a ajustar o centro do *cluster* antes da próxima iteração. O processo se repete até que se atinja uma precisão pré-determinada.

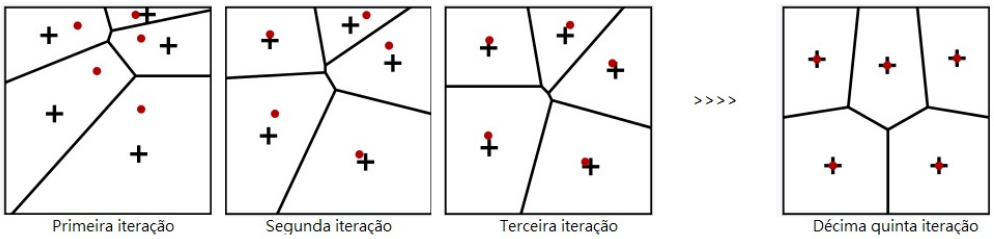

Figura 3.3: Exemplo do algoritmo de Lloyd. Estão ilustrados a tesselação de Voronoi [8] dos pontos a cada iteração. O sinal de "+" denota os centróides de cada célula Voronoi.

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} ||x - \mu_i||^2 \tag{3.5}$$

Onde μ_i é a média dos pontos em S_i.

Dessa forma, o K-means tem como peculiaridade a criação de *clusters* hiperesféricos de extensão espacial semelhantes, de modo a distribuir as amostras de forma equilibrada entre os *clusters*, mesmo que o agrupamento real dos dados não corresponda a esse tipo de agrupamento. Um efeito não desejado dessa forma de *clustering* é que amostras consideradas bastante distintas em relação a outras amostras, denominadas de *outliers*, acabam influenciando de maneira considerável a formação dos *clusters*.

O número ideal de *clusters* não é automaticamente descoberto pelo algoritmo, como acontece, por exemplo, no algoritmo DBSCAN. O número de *clusters* deve ser informado ao algoritmo. Existem algumas heurísticas para determinar a quantidade ideal de *clusters*. Por exemplo, com um gráfico do número de *clusters* em relação à porcentagem da variância explicada, é possível visualizar que, nos primeiros *clusters* acrescentados, a quantidade de variância explicada aumenta rapidamente. No entanto, passado um determinado limite, o ganho da variância explicada passa a ser menor, gerando, dessa forma, um ângulo no gráfico. Este ponto é um indicativo do número ideal de *clusters* para o conjunto de dados.

Entretanto, essa heurística não é muito precisa para determinar o número exato de *clusters* em todos os casos. Uma alternativa é utilizar o método *gap statistic* desenvolvido por Tibshirani et al. [115], no qual se mede a mudança de dispersão dentro de cada *cluster* com uma distribuição nula de referência apropriada.

K-medoids

O K-medoids é um algoritmo de clusterização semelhante ao K-means, pois ambos utilizam uma estratégia de particionar os dados em grupos menores e, em seguida, minimizar as distâncias das amostras de cada grupo em relação ao centro do *cluster*. A diferença entre

os dois algoritmos é que o K-medoids escolhe pontos do próprio *dataset* como centro do *cluster* em vez das médias calculadas pelo K-means. O ponto escolhido, também denominado de *medoid*, é o que possui a menor dissimilaridade média entre as amostras de seu *cluster*. Por minimizar a soma das dissimilaridades dos pares de amostras em vez da soma dos quadrados das distâncias, o K-medoids é menos suscetível à influência de *outliers* do que o K-means. Por outro lado, o K-medoids é mais computacionalmente intensivo que o K-means [94].

Clustering ou Agrupamento Hierárquico

Diferente do K-means, o método hierárquico busca construir os *clusters* recursivamente, particionando as amostras de maneira *top-down* ou *bottom-up*, estabelecendo uma hierarquia relacionada à distância entre os pontos a serem particionados [76].

Na abordagem *bottom-up*, denominada aglomerativa, cada ponto, ou amostra, começa em seu próprio *cluster*, e pares de *clusters* próximos entre si são fundidos em *clusters* maiores em cada iteração.

Já a abordagem *top-down*, denominada de agrupamento ou *clustering* por divisão, parte do princípio de que todas as amostras encontram-se em um mesmo *cluster*, e, a cada iteração, ocorre uma divisão separando as amostras em grupos distintos.

Ambas as abordagens utilizam uma estratégia gulosa para a decisão de unir ou cindir *clusters*. Este tipo de algoritmo é menos suscetível a *outliers* como o K-means, gerando *clusters* mais precisos. Por outro lado, a complexidade de um algoritmo hierárquico aglomerativo é $O(n^2 \log(n))$ e do algoritmo por divisão é $O(2^n)$, o que os torna não adequados para uma grande quantidade de dados [98].

O critério avaliado pelo algoritmo é a medida de dissimilaridade entre dois conjuntos de pontos. Essa dissimilaridade é verificada por uma métrica entre os pontos e um critério de enlace, *linkage*, entre os conjuntos de pontos. A escolha da métrica (Tabela 3.1) e do critério de enlace (Tabela 3.2) está relacionada com o tipo de separação que se quer obter dos dados e, sendo assim, vão influenciar o tamanho e formato dos *clusters*.

Tabela 3.1: Alguns exemplos de métricas normalmente utilizadas em clusterização.

Métrica de Distância	Fórmula
Euclidiana	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Euclidiana quadrática	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Coseno	$\cos(a,b) = \frac{a \cdot b}{\|a\|\|b\|}$

Tabela 3.2: Alguns exemplos de enlance comumente utilizados em algoritmos de clusterização hierárquico. A função d é a métrica utilizada pelo algoritmo.

Critério de Enlace	Fórmula
Máximo, ou Enlace completo	$max\{d(a,b) : a \in A, b \in B\}$
Mínimo, ou Enlace único	$min\{d(a,b) : a \in A, b \in B\}$
Enlace Médio, ou UPGMA	$\frac{1}{\|A\|\|B\|} \sum_{a \in A} \sum_{b \in B} d(a,b)$

Determinando a qualidade dos *clusters*

Existem algumas métricas de avaliação para determinar a qualidade da clusterização proposta pelos algoritmos. Essas métricas podem ser de duas formas: supervisionada, quando cada dado já possui uma classe que se deseja comparar com a classe proposta, ou seja, o *cluster* designado pelo algoritmo. Por outro lado, na métrica de avaliação não-supervisionada, como os dados não possuem uma classe como referência, avalia-se como os *clusters* estão dispostos entre si e, consequentemente, a divisão das amostras entre os *clusters*.

Entre exemplos de métricas supervisionadas, podemos citar a homogeneidade e a completude. Uma clusterização com uma pontuação alta na métrica de homogeneidade significa que todas as amostras presentes em um mesmo *cluster* possuem a mesma classe verdade.

Completude, por outro lado, é uma métrica de clusterização que avalia se todas as amostras de uma mesma classe verdade estão no mesmo *cluster*. A completude garante que uma clusterização que utilize apenas a homogeneidade como critério não tenha um viés de criar um *cluster* para cada amostra do conjunto de dados a ser clusterizado. Uma boa clusterização atende aos dois critérios de completude e homogeneidade, criando um número mínimo de *clusters*, de forma que as amostras semelhantes fiquem agrupadas em um mesmo *cluster*.

Um exemplo de métrica não supervisionada é o escore de Silhouette, que visa medir o grau de separação dos *clusters*. O escore de Silhouette é calculado a partir da média das distâncias intra-*cluster* e a média das distâncias a partir do *cluster* mais próximo.

Por exemplo, para cada dado i, utilizamos $\bar{d}_i(i)$ como a dissimilaridade média entre todos os dados pertencentes ao mesmo *cluster* de i. Nesse contexto, a dissimilaridade média entre um ponto i e um *cluster* c é a distância média entre i e todos os pontos do *cluster* c. Essa dissimilaridade média avalia o quão adequada está a atribuição de i ao *cluster* para o qual foi designado por um algoritmo de clusterização.

Outro valor importante a ser considerado no escore de Silhouette é a dissimilaridade entre i e os *clusters* dos quais i não faz parte. Assim, definindo $d_e(i)$ como a dissimilaridade entre i e seu *cluster* vizinho, o *cluster* com a menor dissimilaridade média de i, define-se

o escore de Silhouette, conforme apresentado na Equação 3.6:

$$s(i) = \frac{\bar{d}_e(i) - \bar{d}_i(i)}{max\{\bar{d}_e(i), \bar{d}_i(i)\}} \tag{3.6}$$

A Equação 3.6 pode ser expressa conforme o sistema de equação apresentado em 3.7.

$$s(i) = \begin{cases} 1 - \frac{\bar{d}_i(i)}{\bar{d}_e(i)}, \; se \quad \bar{d}_e(i) < \bar{d}_i(i) \\ 0, \; se \quad \bar{d}_i(i) = \bar{d}_e(i) \\ \frac{\bar{d}_e(i)}{\bar{d}_i(i)} - 1, \; se \quad \bar{d}_e(i) > \bar{d}_i(i) \end{cases} \tag{3.7}$$

Da Equação 3.7 temos que $-1 \leq s(i) \leq 1$, em que, se $s(i)$ está próximo de -1, isso indica que a atribuição do dado i ao *cluster* C para o qual ele foi atribuído não está correta. Quanto mais próximo de um, mais correta está a atribuição.

3.4 Maldição de Dimensionalidade

No contexto de aprendizado de máquina, o conceito de Maldição de Dimensionalidade afirma que, ao contrário do que seria intuitivo, ao aumentar sem critério o número de características para um modelo de classificação sem aumentar o número de amostras de treinamento, o desempenho do modelo aumenta apenas nos dados de treinamento, mas tende a ser pior de maneira geral em dados novos, ou seja, ocorre *overfitting* do classificador. Este conceito também é denominado de fenômeno de Hughes [53] (Figura 3.4).

Ao aumentar a dimensionalidade, ou seja, ao aumentar o número de características, sem aumentar a quantidade de dados de treinamento, a distribuição dos dados existentes no hiperespaço se torna mais esparsa, de maneira não uniforme, e tende a se concentrar nos cantos do hiperespaço.

Como consequência desse fenômeno, torna-se difícil a estimativa precisa dos coeficientes e parâmetros que definem a fronteira de decisão da função que se quer construir. Sem esses valores bem definidos, não é possível obter um classificador generalizável que funciona tão bem tanto com os dados de treinamento quanto com dados novos.

De fato, se a dimensionalidade do espaço de características for aumentada de maneira indefinida, tendendo ao infinito, a razão entre a diferença da distância Euclidiana mínima e máxima de uma amostra até o centroide e a distância mínima vai para zero (equação 3.8).

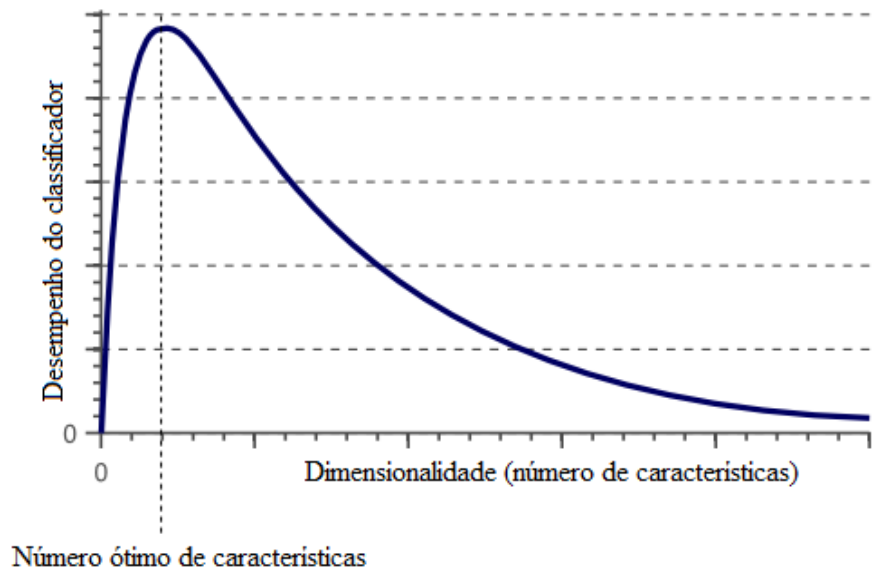

Figura 3.4: Representação gráfica da queda de desempenho de um classificador com *overfitting* devido ao aumento do número de características (adaptado de [107]).

$$\lim_{d \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} = 0 \tag{3.8}$$

Isso significa que a métrica de distância perde efetividade para medir dissimilaridade em espaços de alta dimensionalidade, o que prejudica o desempenho dos classificadores que dependem dessas métricas. Assim, é importante equilibrar o número de características a serem utilizadas para criar um classificador, devendo ser o menor número possível para que se atinja a precisão adequada em dados de teste.

Além de diminuir os efeitos de *overfitting* com os dados de treinamento, reduzir o tamanho da dimensionalidade dos dados utilizados tem como vantagem a redução do tempo e custo de treinamento do modelo, além de facilitar a interpretação do modelo construído [55].

Existem duas estratégias para reduzir o número de características de um conjunto de dados: a seleção de características e a extração de características.

Seleção de características

A seleção de características é uma estratégia de redução de dimensionalidade que objetiva escolher um subconjunto das características mais relevantes para a construção de um modelo de classificação ou regressão [27].

O conceito da seleção de características baseia-se na ideia de que há a possibilidade de que algumas das características presentes nos dados utilizados no treinamento sejam irrelevantes ou redundantes com uma ou mais características do conjunto de dados. Assim, essas características podem ser removidas sem causar prejuízo à acurácia do modelo.

Tradicionalmente, algoritmos de seleção de características são implementados como uma combinação de algoritmos de busca por um subconjunto de características com uma avaliação da acurácia alcançada por um modelo com esse subconjunto. Intuitivamente, pode-se imaginar que a solução ótima possa ser encontrada testando todos os subconjuntos possíveis e escolhendo o que resulta no menor erro. Entretanto, essa abordagem é computacionalmente intratável para conjuntos com grande número de características.

A métrica de avaliação do subconjunto de características influencia o comportamento do algoritmo de busca. Por isso, a métrica é utilizada como critério para diferenciar três categorias de algoritmos de seleção de características: *wrappers*, métodos de filtragem e métodos integrados ou *embedded* [47].

Métodos *wrappers* são métodos de busca nos quais diferentes subconjuntos de características são utilizados para treinar um modelo. Em seguida, algumas inferências são extraídas dos resultados e, então, o algoritmo decide por adicionar ou remover características de um subconjunto de características até encontrar o subconjunto com melhor desempenho (Figura 3.5). Os métodos *wrappers* são essencialmente algoritmos de busca e representam um custo computacional elevado.

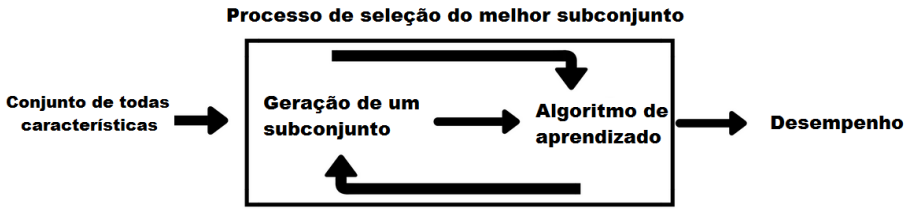

Figura 3.5: Esquema de funcionamento de um método *wrapper* para seleção de características.

Métodos de filtragem, diferente dos métodos *wrapper*, ocorrem como um pré-processamento, antes do treinamento dos modelos e de forma independente de algoritmo de aprendizado de máquina (Figura 3.6). A seleção de características é feita baseada em pontuações oriundas de testes estatísticos aplicados aos dados, de forma a medir a correlação entre cada característica e a classe resposta conhecida dos dados de treinamento. Dentre os testes estatísticos mais comuns utilizados para esta finalidade, podemos citar a correlação de Pearson [84], Análise de Discriminantes Lineares (LDA) [37, 9], ANOVA [36] e teste χ-quadrado [85].

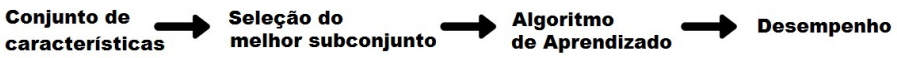

Figura 3.6: Pipeline de funcionamento de um método de filtragem para seleção de características.

Normalmente, os métodos de filtragem resultam em subconjuntos de características que alcançam menor desempenho, quando comparados com os subconjuntos obtidos por métodos *wrappers*. No entanto, o subconjunto de características gerado por métodos de filtragem não depende de um modelo de predição específico e, por isso, torna mais evidentes as relações entre características.

Métodos integrados ou *embedded* de seleção combinam características de métodos de filtragem e métodos *wrappers*. Esses métodos estão implementados dentro de alguns algoritmos de aprendizado de máquina que possuem um mecanismo próprio de seleção de características durante o construção do modelo de classificação com os dados de treinamento. Dois exemplos de algoritmos de aprendizado de máquina conhecidos que possuem métodos de seleção de características integrados para reduzir *overfitting* são o LASSO [114] e RIDGE [45].

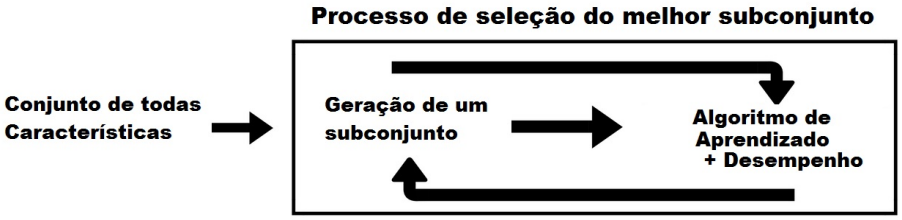

Figura 3.7: Representação do processo de seleção de características por métodos integrados.

Extração de características

A estratégia denominada de extração de características difere da seleção de características, pois, na primeira, novas características são geradas a partir das características originais. Essas transformações podem ser lineares, como ocorre com a utilização de *Principal Component Analysis* (PCA) [86], ou não-lineares, como ocorre com *Self-Organizing Map* (SOM) [61].

Uma propriedade dessa estratégia de redução de dimensionalidade, em contraste com a seleção de características, é que as novas características geradas por estes métodos não podem ser diretamente interpretadas pelo detentor do conhecimento do domínio para o

qual está sendo aplicado o modelo de aprendizado de máquina. Cada nova característica gerada pelas transformações agrega duas ou mais características originais, formando uma nova característica numérica para ser utilizada na construção de um modelo de predição.

Apesar de essa estratégia de redução obter bons resultados na redução de dimensionalidade, levando a bons modelos de predição, esses possuem duas peculiaridades que devem ser levadas em consideração quando utilizados.

A primeira é que, como geralmente combinam duas ou mais características originais do conjunto de dados, se houver uma correlação significativa entre duas ou mais características, a nova característica gerada, a partir de características correlacionadas, vai ser afetada por esta correlação. A atribuição de importância das características originais que irão compor as novas características é fortemente afetada por correlação existente nos dados.

A outra peculiaridade da extração de características é que a interpretação da importância de cada nova característica feita pelo detentor do conhecimento de domínio fica mais difícil. Uma vez que cada nova característica representa uma mistura das características originais que se complementam matematicamente, na aplicação do modelo no domínio para o qual o modelo está sendo criado, as características agrupadas podem não fazer sentido do ponto de vista do pesquisador da área de domínio.

Capítulo 4

Dados e Métodos

Neste capítulo, são apresentados os bancos de dados dos quais foram obtidos os *datasets* utilizados no trabalho e o método proposto para seleção de características. Na Seção 4.1, estão apresentados os dados utilizados no trabalho e o processo de filtragem e seleção das sequências de transcritos que foram utilizadas. Na Seção 4.2, está descrito o processo de seleção de características proposto neste trabalho.

4.1 Dados

Os dados utilizados para o trabalho foram obtidos de três bases de dados amplamente utilizadas: a LNCipedia [123], a RefSeq [92] e o Ensembl [4].

O *dataset* de *Homo sapiens* é composto de 111.145 sequências de transcritos codificadores da base RefSeq e 100.849 sequências de transcritos lncRNA da base LNCipedia versão 4.0, ambos referentes à montagem GRCh38. O *dataset* de *Mus musculus*, obtido da montagem GRCm38 da base de dados Ensembl, é composto de 90.854 transcritos codificadores e 12.528 lncRNAs. O *dataset* de *Danio rerio*, obtido da montagem GRCz10 da base de dados Ensembl, é composto de 50.731 transcritos codificadores e 3.976 transcritos lncRNAs.

Todos os três *datasets* passaram por um processo de filtragem para escolha do conjunto de sequências de treinamento e do conjunto de teste. Essa etapa é necessária para que as classes tenham o mesmo número de transcritos, visto que os *datasets* são desbalanceados, e também para remover sequências cujas funções não foram confirmadas. O processo de filtragem também é responsável por remover sequências muito diferentes do resto do conjunto, também denominada *outliers*, uma vez que os *outliers* podem prejudicar o desempenho de um modelo de classificação.

4.1.1 Filtragem

Na etapa de filtragem das sequências, foram excluídos do conjunto de dados dos transcritos codificadores, mRNAs, todas as sequências marcadas como *PREDICTED*, ou seja, as sequências que, apesar de possuírem uma composição que indique que elas sejam mRNAs, não foram confirmadas ainda como codificadoras de proteínas.

Do conjunto de sequências não-codificadoras, foram excluídas as que possuíam N no lugar de qualquer uma da bases A, C, G, U, ou seja, as sequências que ainda não tiveram sua composição totalmente confirmada. A justificativa prática para remover todas as sequências que tenham ao menos um N é que, como as características utilizadas neste trabalho são as extraídas das porcentagens dos k-mers das sequências, a inserção de uma nova letra, no caso N, geraria novas características de frequências com N em sua composição, o que na prática não seria interessante, tendo em vista que as bases de RNA são quatro.

4.1.2 Escolha de sequências

Em seguida ao processo de filtragem, optou-se por uma estratégia *alignment-free* semelhante à utilizada por Tripathi et al. [116] para a escolha das sequências, de forma a equilibrar as classes.

Para isso, foram computadas a quantidade dos k-mers de tamanhos 2, 3, 4 e 5 para cada sequência de ambas as classes. Em seguida, para cada conjunto de k-mers de mesmo tamanho de cada sequência, calculou-se a Entropia de Shannon (Equação 4.1), na qual p_i é a probabilidade de ocorrência do k-mer i na sequência. Dessa forma, foram geradas quatro características para todas as sequências: $H2, H3, H4, H5$.

$$H = -\sum_{i} p_i \log p_i \tag{4.1}$$

A Entropia de Shannon é um conceito da Teoria da Informação que mede a quantidade de informação esperada em uma mensagem, ou em qualquer fluxo de informação. É um conceito que vem sendo aplicado em diversos trabalhos relacionados a análise de sequenciamentos [122]. No caso deste trabalho, o conceito foi utilizado para escolher sequências das classes distintas que fossem as mais diferentes entre si, sem ter que realizar o alinhamento tradicional de todas as sequências entre si.

Uma vez calculadas as características de entropia dos k-mers de diferentes tamanhos, foi possível escolher pares de sequências de classes distintas que estivessem mais distantes entre si para a composição do *dataset*.

Uma possibilidade para isso seria calcular a matriz de distâncias entre as sequências de PCTs e de lncRNAs e escolher as sequências com as maiores distâncias. No entanto,

essa alternativa é demorada e exigiria muitos recursos de memória, disco e tempo de processamento.

A alternativa escolhida foi agrupar os dados de ambas as classes com um algoritmo K-means configurado para gerar dois *clusters*. O *cluster* com a maior parte das sequências de lncRNAs foi denominado de lncRNA-Cluster e o outro PCT-Cluster. As sequências da classe minoritária de cada *cluster* foram descartadas.

Uma vez criados os *clusters*, foram escolhidas as sequências com maiores distâncias euclidianas do centro do *cluster* oposto (Figura 4.1).

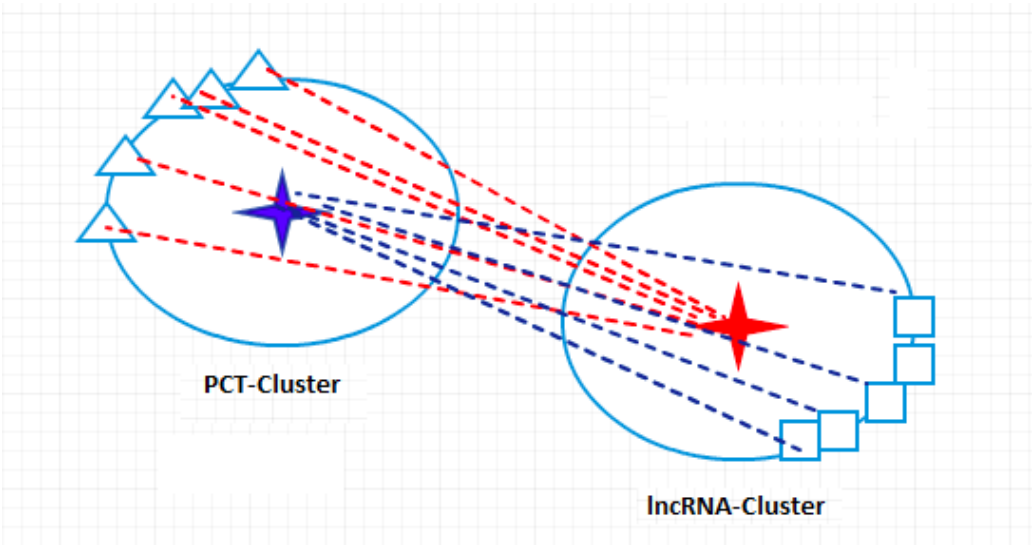

Figura 4.1: Processo de escolha das sequências para compor os *datasets* utilizados neste trabalho. As estrelas indicam o centro de cada *cluster*, e as linhas tracejadas entre as estrelas e as sequências nas bordas dos *clusters*, representadas por triângulos e quadrados, representam as distâncias entre as sequências de uma classe até o centro do *cluster* da classe oposta.

Com esse método, foram escolhidas 30 mil sequências de cada classe para o *dataset* de *Homo sapiens*, 13.800 sequências de cada classe para o *dataset* de *Mus musculus*, e 6 mil sequências de cada classe para o *dataset* de *Danio rerio*.

Depois de selecionadas as sequências dos *datasets*, as características utilizadas para a escolha de sequências foram descartadas para evitar que qualquer viés proveniente da etapa de seleção de sequências influenciasse no método proposto para a seleção de características. Com isso, foram geradas as características de frequências dos k-mers de tamanho 2, 3 e 4 que compõem as sequências dos transcritos.

A justificativa para a escolha dessas características de frequência é que elas são bastante utilizadas por diversas ferramentas de classificação [124, 66, 33], direta ou indireta-

mente. Logo, um método para escolher as melhores características pode auxiliar tanto na acurácia quanto no tempo para treinar as ferramentas e classificar novas sequências.

Dados de Treinamento e Dados de Teste

O último passo de pré-processamento foi separar, de maneira aleatória, os dados de ambas as classes em dois conjuntos com uma quantidade aproximadamente igual de sequências em cada: um conjunto de dados de treinamento e um conjunto de dados de teste.

O conjunto de dados de treinamento foi utilizado tanto para a escolha das características mais relevantes quanto para treinar um modelo de classificação. Essa etapa de desenvolvimento do modelo de classificação foi implementada com o algoritmo *Random Forest* para classificação. Os dados de teste foram então utilizados para verificar o desempenho de acurácia do modelo de classificação, ou classificador, treinado com as características selecionadas.

A justificativa para o uso de *Random Forest* está no fato de ser um algoritmo *Ensemble*, do tipo *Bagging*, que é relativamente simples de utilizar, pois são poucos parâmetros que devem ser ajustados para se obter a melhor configuração possível para o classificador.

Existem outros classificadores aplicados a diversas áreas de pesquisa, inclusive Biologia Molecular, que obtêm desempenhos ligeiramente melhores que o RF. No entanto, são mais complexos, possuindo diversos parâmetros de ajuste para obter melhor desempenho e necessitam de muito mais tempo de processamento para treinamento dos classificadores. Como exemplo, pode-se citar as redes neurais profundas [100], que obtêm resultados bastante expressivos até mesmo para predição de lncRNAs [116].

Neste trabalho, o objetivo não está na obtenção do melhor desempenho de acurácia para a tarefa de distinção entre PCTs e lncRNAs, e sim no processo de seleção das melhores características de maneira independente do classificador utilizado. Dessa forma, o RF é utilizado como uma ferramenta de comparação de desempenho entre as características selecionadas pelo S2FS e as características escolhidas por outro método de seleção. Para esse propósito, o RF é simples, rápido e preciso. Além disso, o RF obtém ótimos desempenhos de acurácia em dados provenientes de diversas outras pesquisas na área de Biologia Molecular, sendo por isso utilizado em algumas ferramentas de detecção e identificação de transcritos ncRNAs [34, 65, 2].

O RF também tem como vantagem a possibilidade de descobrir a importância de cada característica utilizada para o treinamento do modelo, pois é um algoritmo que também seleciona características no processo de treinamento do modelo. Assim, é possível verificar, ao final do treinamento do modelo criado por RF, qual a importância de cada $kmer_f$ pré-selecionado pelo método de seleção proposto.

Os resultados obtidos com as características selecionadas foram então comparados com o resultado de classificação com o mesmo algoritmo utilizando os mesmos parâmetros, mas utilizando características selecionadas por um método de seleção de características univariada por meio de uma função de teste χ-quadrado. Para simplificar as referências que serão feitas a esses dois métodos, o método de seleção de características univariada foi denominado de algoritmo de referência. O método de seleção proposto neste trabalho foi denominado de *Single Score Feature Selection - S2FS*.

4.2 *Single Score Feature Selection - S2FS*

A ideia principal do S2FS é escolher um subconjunto de características baseado no desempenho que cada característica obtém isoladamente ao classificar o conjunto de sequências de treinamento. O fluxo de trabalho do método proposto está ilustrado na Figura 4.2. Nela estão apresentados as etapas de obtenção dos transcritos para o trabalho, a filtragem para escolha das sequências, a seleção de características com o S2FS e o teste das características selecionadas.

Para isso, cada característica das sequências de treinamento, ou seja, cada frequência de k-mer de tamanho 2, 3 e 4 foi utilizada para treinar um classificador simples implementado com regressão logística testando dois algoritmos de otimização diferentes para a obtenção do maior valor de *Area Under the Curve* (AUC) para as características. As funções testadas foram a L-BFGS [68] e Liblinear [32] com regularização L2.

A escolha da função de otimização e dos hiperparâmetros desse classificador foram ajustados por *Gridsearch* com um fator de *cross-validation* de 10 (Figura 4.3).

Um fator importante desse processo é utilizar um parâmetro de regularização forte adequado para forçar a separação das melhores características. Caso o parâmetro de regularização seja fraco, muitas das características de frequência acertam muitas sequências de ambas as classes. Por outro lado, se o parâmetro de regularização for forte demais, muitas características de frequência vão errar demasiadamente na classificação das sequências das duas classes. Ambos os resultados são indesejáveis para atingir o objetivo de separar as melhores características.

Para facilitar a compreensão, uma vez que novas características derivadas deste passo inicial são utilizadas em outras etapas do trabalho, essas características originais, as quais estamos interessados em selecionar, serão referenciadas como $kmer_f$ e não pela palavra característica.

Na Tabela A da Figura 4.3 estão as porcentagens dos k-mers de tamanho 2, 3 e 4 que compõem cada transcrito do *dataset* de treinamento. Os $kmer_f$ estão ilustrados como os nomes das colunas da Tabela A. Por exemplo, na Tabela A da Figura 4.3 o

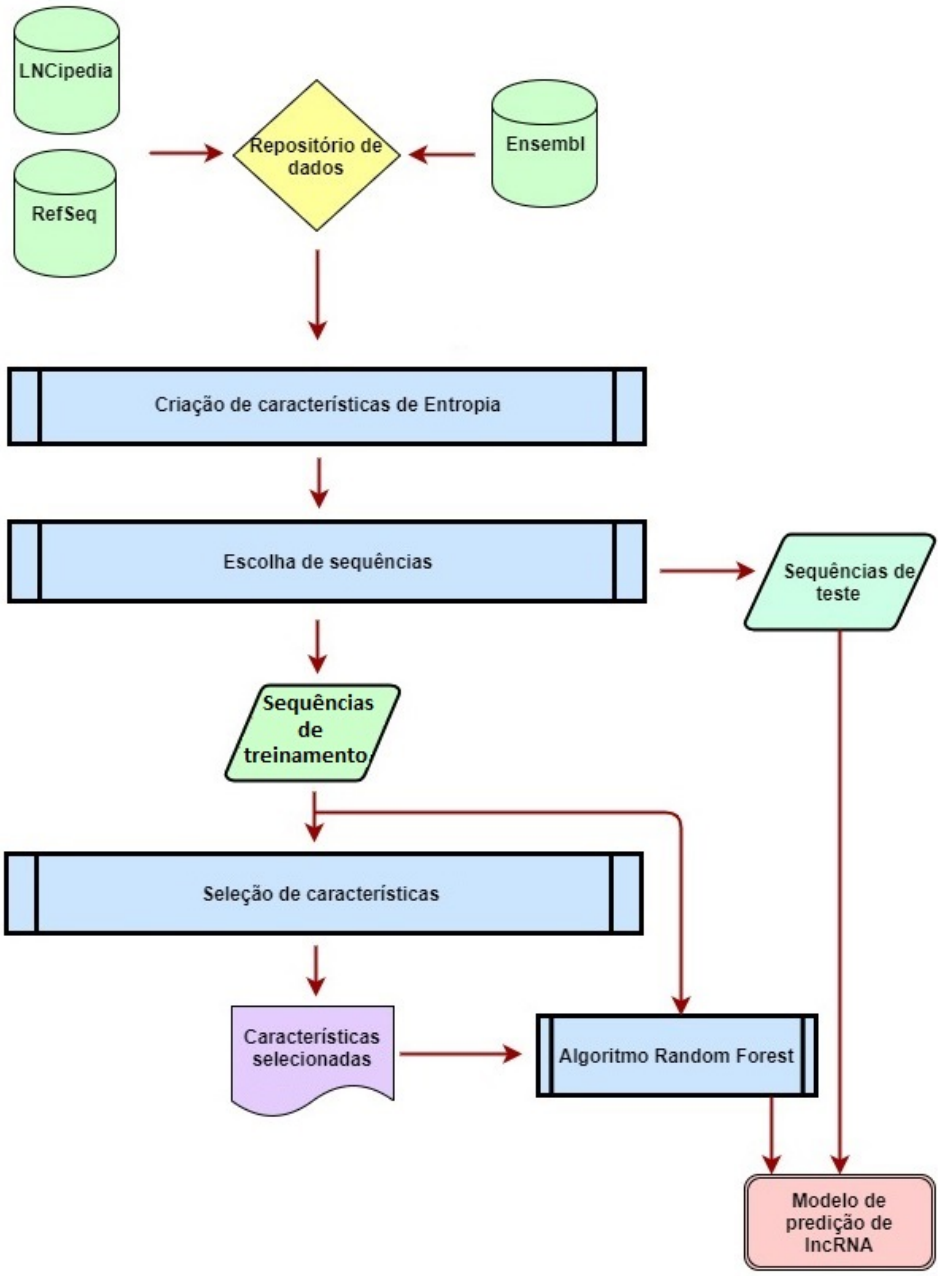

Figura 4.2: Fluxo de trabalho completo do método proposto para seleção de características.

%AC representa qual a porcentagem do 2-mer AC que compõe o transcrito, o %ACG representa a porcentagem do 3-mer ACG que compõe o transcrito e o %TTTT representa a porcentagem do 4-mer TTTT que compõe o transcrito. No total são 336 características das combinações das características 2-mer, 3-mer e 4-mer.

Em seguida, o desempenho de cada $kmer_f$ é verificado, descartando aqueles que não obtiveram um desempenho, ou escore de área sob a curva de ROC, maior que $0,5$, ou

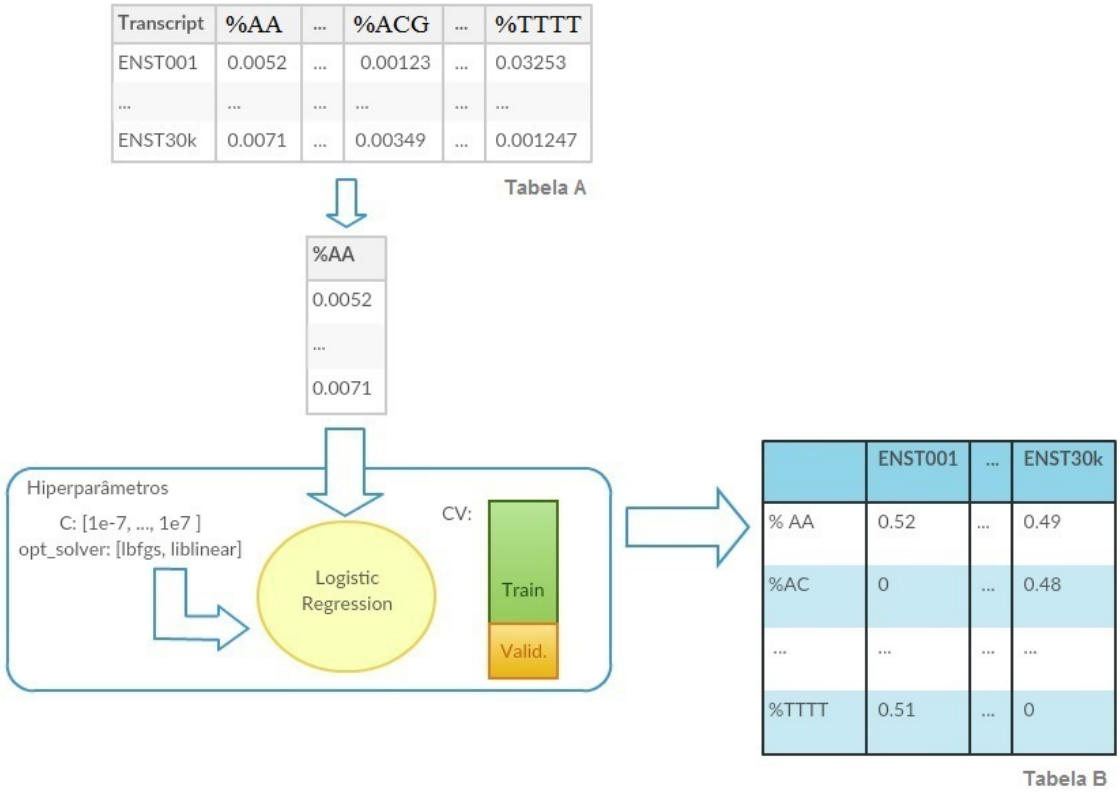

Figura 4.3: Processo de classificação das sequências com Regressão Logística utilizando apenas uma característica por vez. Os resultados gerados são agrupados em uma nova tabela de dados, que é utilizada como critério de seleção das características.

seja, os $kmer_f$ que possuem um desempenho pior do que classificar aleatoriamente as sequências entre PCTs e lncRNAs. Dessa forma, um pequeno subconjunto de $kmer_f$ já é descartado inicialmente.

Com os $kmer_f$ remanescentes, é construída uma tabela transposta da tabela inicial, na qual cada linha é um $kmer_f$ e cada coluna representa uma sequência de treinamento (Figura 4.3:Tabela B). O valor de cada entrada na tabela B é a porcentagem de certeza que aquela sequência é um PCT, utilizando apenas o dado k-mer como característica da regressão logística. Se o valor for acima de $0,5$, o classificador acertou a sequência como PCT. Se for menor que $0,5$, o classificador acertou a sequência como lncRNA. Caso o classificador não tenha acertado a classificação de uma sequência, quando comparados com a classificação real das sequências, o valor zero é atribuído à entrada na tabela referente ao $kmer_f$/sequência. Desta forma, obtemos uma tabela, na qual os valores não-zero representam os acertos de cada $kmer_f$ para a classificação das sequências.

O passo seguinte é contabilizar quantas sequências de cada classe cada $kmer_f$ classificou corretamente gerando duas novas características: nc_hits e pc_hits. Ao se utilizar

duas características para a clusterização dos dados em vez das 30 mil características da Tabela B (referente aos 30 mil transcritos), o algoritmo de clusterização executa mais rápido, e o resultado do agrupamento dos $kmer_f$ é mais facilmente interpretável.

Essas duas caraterísticas então, *nc_hits* e *pc_hits*, são utilizadas para a criação de mais uma característica, que é a distância euclidiana a partir do ponto $(0, 0)$ até (*nc_hits*, *pc_hits*) de cada $kmer_f$. A ideia dessa característica é determinar um critério de escolha para a seleção dos melhores $kmer_f$, pois apenas com a clusterização não é possível determinar qual o melhor *cluster* de $kmer_f$. O critério que foi estabelecido é que os melhores $kmer_f$ são aqueles que classificam corretamente o maior número de sequências de ambas as classes.

No entanto, apenas a distância euclidiana não é suficiente, pois podem ocorrer situações na qual um $kmer_f$ classifica corretamente todas as sequências de uma classe e erra quase todas ou todas as sequências da outra classe obtendo, ainda assim, uma distância euclidiana considerável em relação ao ponto $(0, 0)$ (Figura 4.4). Logo, aplicou-se um peso trigonométrico a esta característica de distância para priorizar os $kmer_f$ que classificam corretamente ambas as classes.

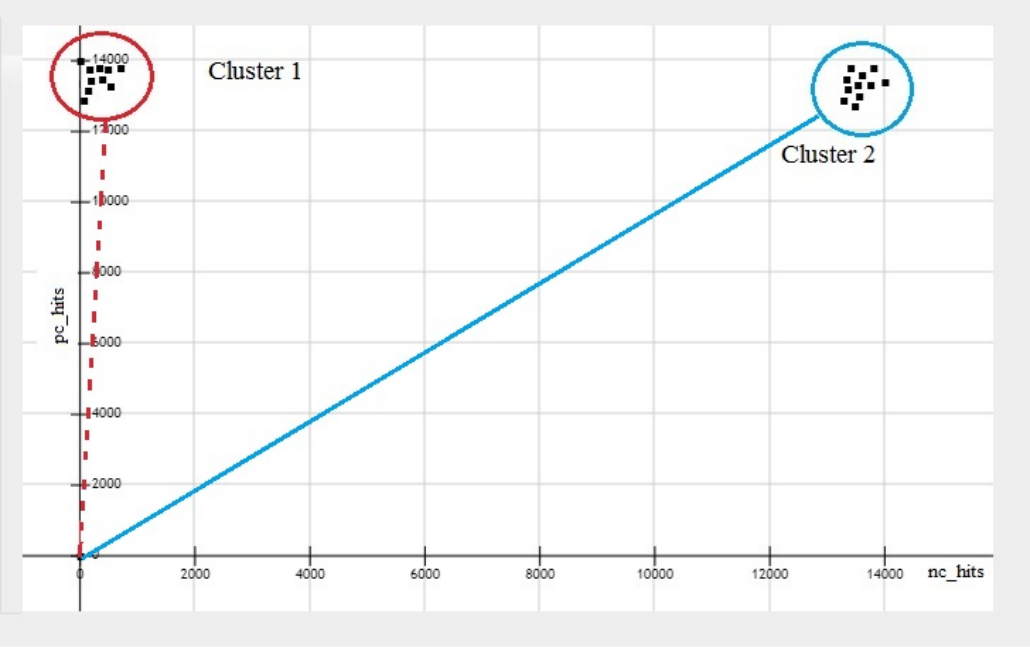

Figura 4.4: O gráfico mostra que o os $kmer_f$ do cluster 2 são melhores candidatos a características importantes, quando comparados aos $kmer_f$ do cluster 1, pois possuem um maior número de acertos de classificação individual nas duas classes.

O peso aplicado foi a função $\sin(\theta) * \cos(\theta)$, na qual θ é o ângulo formado entre os pontos (*nc_hits*, *pc_hits*), $(0, 0)$ e projeção do ponto (*nc_hits*, *pc_hits*) no eixo *nc_hits*, ou seja, (nc_hits, 0). Esta característica já ponderada foi denominada *w_dist*. A Tabela

Tabela 4.1: Características geradas a partir do desempenho de cada $kmer_f$.

kmer	pc_hits	nc_hits	w_dist
AA	302	15103	3.019396e+2
...	...	...	...
TGT	495	15103	4.947344e+2
...	...	...	...
GCAT	9567	11	1.099999e+1

4.1 mostra três exemplos das características *nc_hits*, *pc_hits* e *w_dist* geradas para cada $kmer_f$.

Estratégia gulosa e Estratégia de clusterização

Com as três novas características geradas, *nc_hits*, *pc_hits* e *w_dist*, é possível adotar três estratégias diferentes para a seleção de características: uma estratégia gulosa, uma estratégia de clusterização, ou a combinação das duas.

Com a estratégia gulosa, é feita uma ordenação dos $kmer_f$ em relação à característica *w_dist*, e, em seguida, escolhem-se os $kmer_f$ com os maiores valores de *w_dist*, ou seja, os $kmer_f$ que acertam o maior número de sequências de ambas as classes. Essa estratégia é mais rápida do que clusterizar os dados e é ideal para quando já se sabe o número de características que se quer escolher para treinar o classificador.

A estratégia gulosa pode ser implementada por duas abordagens distintas. A primeira abordagem seria simplesmente acrescentar os $kmer_f$ com maior valor *w_dist* até se alcançar o número de características desejado. No entanto, essa abordagem, apesar de ser mais rápida, pode acarretar a escolha de $kmer_f$ correlacionados entre si, uma vez que a primeira etapa de classificação com o algoritmo de regressão logística é feita com cada característica individualmente.

A segunda abordagem seria fazer uma filtragem no momento de se acrescentar novos $kmer_f$ ao conjunto de características que se quer obter. Essa filtragem seria uma verificação de correlação dos valores, nos transcritos de treinamento, referentes aos $kmer_f$ que já foram acrescentadas ao conjunto de características relevantes com o próximo $kmer_f$ de maior *w_dist* a ser acrescentado a esse conjunto. O processo de seleção gulosa com verificação de correlação está descrito no Algoritmo 1.

A possibilidade de utilizar as duas abordagens é interessante, pois alguns métodos de Aprendizado de Máquina são mais sensíveis à correlação de características do que outros no momento do treinamento dos modelos. Modelos de classificação treinados com SVM, por exemplo, tendem a sofrer com deterioração de capacidade de generalização se características irrelevantes ou correlacionadas forem utilizadas para treinar os modelos [15].

Algorithm 1 Escolha gulosa com filtro de correlação

```
1: procedure GREEDY SELECTION
2:     S ← ∅
3:     Let Y[1...n] be list of kmer_f sorted by the w_dist
4:     addFeature ← TRUE
5:     for kmer_f ∈ Y do
6:         if S ≠ ∅ then
7:             for feature ∈ S do
8:                 if |ρ(feature, kmer_f)| > Threshold then
9:                     addFeature ← FALSE
10:                end if
11:            end for
12:        end if
13:        if addFeature = TRUE then
14:            S ← S + {kmer_f}
15:        end if
16:        addFeature ← TRUE
17:    end for
18: end procedure
```

Neste trabalho, utilizou-se o coeficiente de Pearson [87] para a verificar a correlação entre as características. O controle de correlação é feito pelo valor da variável $Threshold$ na linha 8 do Algoritmo 1. Dessa forma, é possível filtrar o grau desejado de correlação entre os $kmer_f$ que estão no conjunto de características selecionadas. Definimos $0,3 \leq |\rho(feature_1, feature_2)| < 0,5$ como correlação fraca, $0,5 \leq |\rho(feature_1, feature_2)| < 0,7$ como correlação média, e $|\rho(feature_1, feature_2)| \geq 0,7$ como correlação forte entre as características.

Na estratégia de clusterização, utilizou-se um tipo de clusterização hierárquica *bottom-up* denominado clusterização aglomerativa. A escolha do número ideal de *clusters*, da métrica de distância e do critério de *linkage* também foi realizada por uma busca em grade para obter a maior pontuação de Silhouette, ou seja, qual o melhor número de *clusters* para separar os $kmer_f$, utilizando como características nc_hits e pc_hits.

Uma vez determinado o número ideal de *clusters* C, os $kmer_f$ são separados em C *clusters*, utilizando a melhor conjunto de parâmetros de clusterização com a clusterização hierárquica. Para a quantidade pequena $kmer_f$, a clusterização hierárquica é uma opção viável, uma vez que, para *clusters* de tamanhos e formatos distintos, a clusterização hierárquica obtém uma separação melhor do que o k-means.

Se a regularização escolhida na etapa de classificação inicial com cada $kmer_f$ for adequada, a etapa de clusterização deve gerar ao menos dois *clusters*, um com os $kmer_f$ que obtiveram o melhor desempenho no classificador do primeiro estágio e outro com os $kmer_f$ de pior desempenho.

No entanto, o resultado da clusterização sozinho não é suficiente para determinar qual o melhor conjunto de características. Para selecionar o melhor *cluster*, verifica-se então a média dos *w_dist* dos $kmer_f$ de cada *cluster*. O *cluster* com a maior média é o melhor conjunto de $kmer_f$.

A ideia de se utilizar clusterização para a escolha dos melhores $kmer_f$, ou características em geral para problemas de outra área de domínio, é interessante para quando não se tem certeza da quantidade de características que se quer utilizar para o treinamento de um modelo de classificação. A clusterização auxilia na remoção das características consideradas de pior desempenho na classificação realizada no primeiro estágio de seleção dos $kmer_f$ com o algoritmo de Regressão Logística.

A abordagem híbrida, que combina a estratégia gulosa com a estratégia de clusterização, surge da necessidade eventual de reduzir ainda mais o número de $kmer_f$ após a clusterização, pois é possível que o *cluster* selecionado como melhor conjunto tenha mais $kmer_f$ do que o adequado para construir um classificador sem *overfitting*. Sendo assim, uma vez escolhido o melhor *cluster* utiliza-se a estratégia gulosa para escolher os $kmer_f$ com os maiores *w_dist* até o número desejado de características.

Capítulo 5

Resultados e Discussão

Neste capítulo, estão descritos os resultados obtidos com o método proposto de seleção de características relacionadas às frequências de 2-mer, 3-mer e 4-mer. Na Seção 5.1, estão descritos os resultados obtidos com o *dataset* de *Homo sapiens*. Na Seção 5.2, estão os resultados obtidos com um subconjunto do *dataset* de *Homo Sapiens*, no qual estão presentes apenas os lincRNAs como transcritos não-codificadores. Na Seção 5.3, estão apresentados os resultados obtidos com o *dataset* de *Mus musculus*. Na Seção 5.4, estão os resultados com o *dataset* de *Danio rerio*. Por fim, na Seção 5.6, os resultados obtidos são discutidos.

Todos os resultados de acurácia apresentados foram obtidos utilizando um modelo de classificação implementado com um algoritmo *Random Forest*, utilizando as características escolhidas pelo método proposto e também as características escolhidas por um algoritmo de escolha univariada de características.

Os testes foram realizados inicialmente apenas com as características de mesmo número de pares de bases, ou seja, apenas com 2-mer, 3-mer e 4-mer. Em seguida, todos os três conjuntos de características foram utilizados para verificar o desempenho do método proposto aplicado a um conjunto maior e mais diversificado de dados.

Os resultados obtidos com os primeiros testes no *dataset* de *H. sapiens* sugeriram que seria interessante também realizar testes utilizando lincRNAS, um subconjunto de lncRNAs do *dataset* de *Homo sapiens*.

No caso dos testes realizados com o *dataset* de *H. sapiens*, também utilizou-se como característica o tamanho da maior ORF gerada pela ferramenta CPAT [124], uma vez que esta característica é bastante utilizada em métodos de classificação de transcritos.

A característica ORF também foi utilizada para testar a correlação entre o desempenho de cada $kmer_f$ na etapa inicial do método proposto e o desempenho do $kmer_f$ junto com a ORF no modelo de classificação implementado com *Random Forest*.

Todos os testes foram realizados na mesma máquina, com as mesmas configurações de hardware e software. Foi utilizado um computador com um processador intel i3 3.3GHz, 8GB de memória RAM e 100G de disco rígido e o sistema operacional utilizado foi o Linux Debian 3.16.36. Todos os testes seguiram rigorosamente os mesmos passos do *pipeline* para seleção das sequências e para geração de características para serem escolhidas pelos dois métodos de seleção testados.

5.1 *Homo sapiens*

5.1.1 Amostras caracterizadas por 2-mers

Para o *dataset H. sapiens*, representado pela decomposição dos transcritos em frequências de ocorrências de 2-mers, estão apresentados os resultados de comparação do desempenho de um classificador treinado com apenas sete características escolhidas por cada um dos métodos. Isso é justificado pois, no estágio de clusterização do S2FS, em média, restam apenas 7 características consideradas relevantes das 16 características possíveis.

Um ponto a se considerar com a caracterização dos transcritos apenas por 2-mer é que, além do número pequeno de características, existe uma forte correlação entre elas. Logo, os resultados obtidos para 2-mer não foram tão expressivos quanto os resultados obtidos com os transcritos representados por frequências de ocorrência dos 3-mer e dos 4-mer.

Em todos os testes, a acurácia do modelo nos dados de teste aumenta à medida que aumentamos o número de características utilizadas. É possível constatar estes resultados na Figura 5.1, que apresenta o desempenho dos classificadores treinados com as características escolhidas pelo algoritmo de referência.

Os pontos azuis representam o teste de desempenho dos modelos classificadores com os próprios dados de treinamento, enquanto que os pontos verdes representam o desempenho dos classificadores com os dados de teste. As linhas azul e verde representam funções de regressão linear apenas para facilitar a visualização do aumento da acurácia nos dados de treino e teste.

As características selecionadas pelo algoritmo de referência estão apresentadas na Tabela 5.1. As características estão ordenadas pelas suas importâncias para o algoritmo *Random Forest* na construção do classificador utilizando os dados de treinamento

Tabela 5.1: Características 2-mer mais importantes escolhidas pelo algoritmo de escolha univariada, ordenadas pela importância da característica para o classificador.

Características
CG, CA, AG, TT, GG, AT, TA

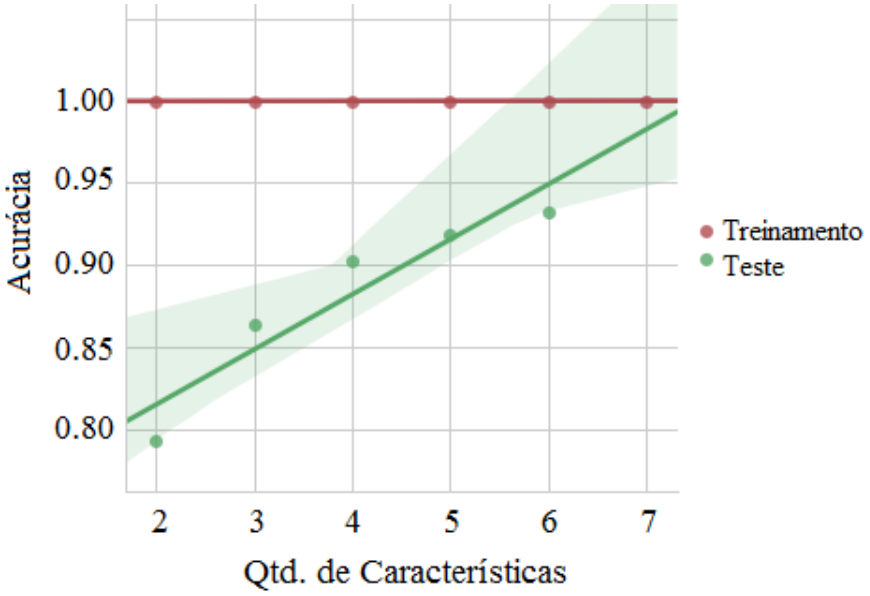

Figura 5.1: Resultado das classificações de um conjunto de dados de teste com os transcritos de *H. sapiens* caracterizados por 2-mer. Os classificadores foram treinados com as características escolhidas pelo algoritmo de referência .

Na Figura 5.2 estão apresentados os resultados de desempenho obtidos com as características escolhidas pelo S2FS. De forma semelhante ao algoritmo de referência, o S2FS seleciona características que aumentam o desempenho dos classificadores de maneira monotônica.

As características selecionadas pelo S2FS estão apresentadas na Tabela 5.2. Assim como na Tabela 5.1, as características estão ordenadas de acordo com suas importâncias para o classificador.

Tabela 5.2: Características 2-mer mais importantes escolhidas pelo S2FS.

Características
CG, GT, TG, TT, AA, AT, TA

Ao comparar o desempenho dos dois métodos de seleção (Figura 5.3), é possível ver que o S2FS obtém pequeno ganho de desempenho em relação ao algoritmo de referência a partir de cinco características.

Um ponto a se observar é que os desempenhos estão bastante altos considerando o baixo número de características utilizadas. No entanto, devido ao processo inicial de seleção de sequências, não se pode generalizar que este pequeno número de características seja suficiente para obter acurácias tão altas. O foco do resultado apresentado é a comparação do desempenho de classificação utilizando dois métodos de seleção de características, e não o desempenho do classificador, uma vez que este está utilizando apenas um subconjunto

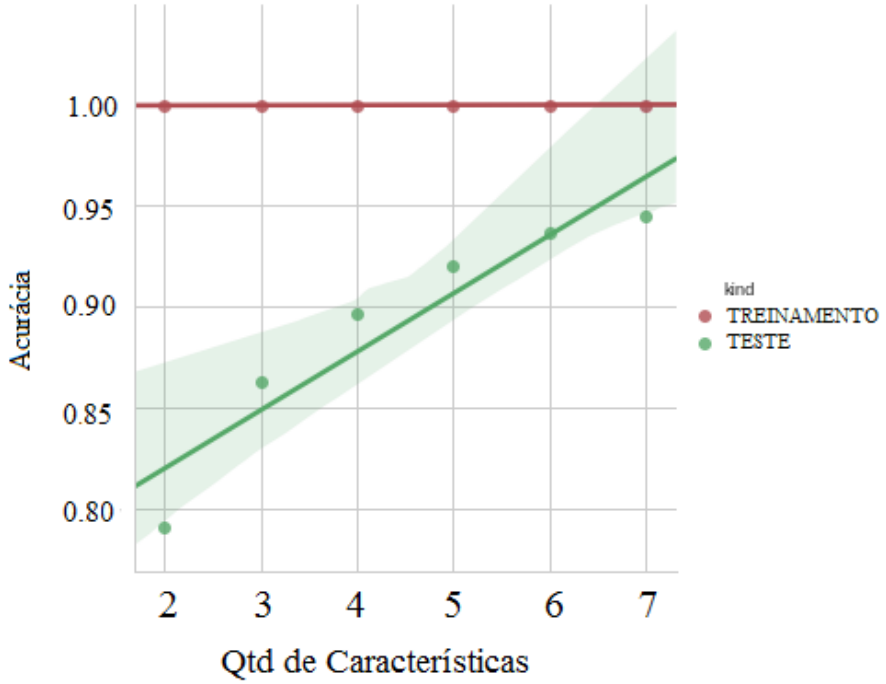

Figura 5.2: Resultado das classificações com o conjunto de dados de teste utilizando as características escolhidas pelo S2FS. Os transcritos são do *dataset* de *H. sapiens* utilizando características 2-mers.

do total de transcritos disponíveis devido ao processo de seleção inicial de sequências para balancear as classes do *dataset*.

Com o pequeno número de características disponíveis, quando as amostras estão caracterizadas por 2-mer, a diferença entre os métodos não é muito evidente. Mais da metade das características escolhidas pelo algoritmo de referência também está presente no conjunto de características escolhido pelo S2FS.

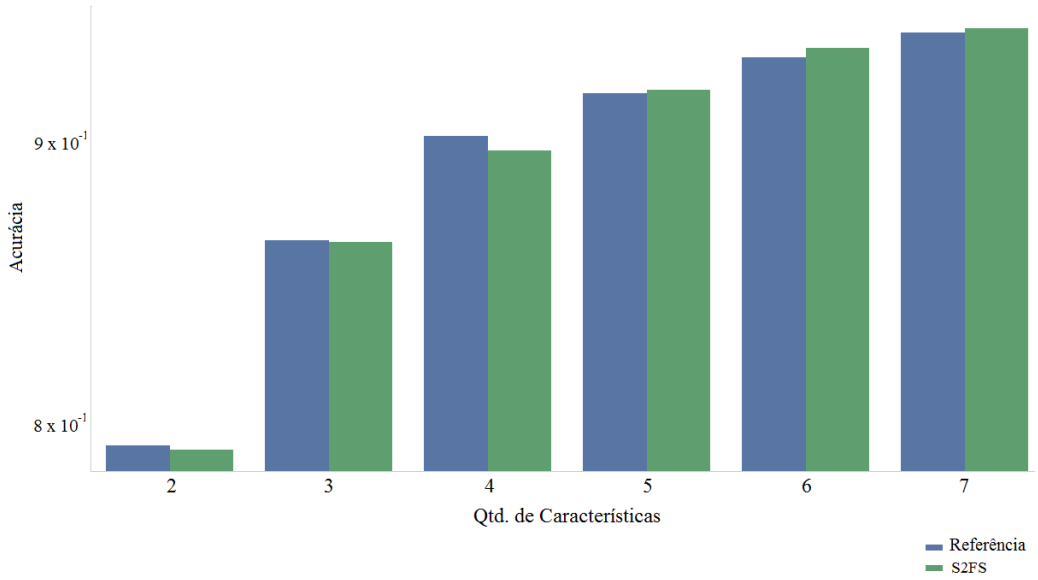

Figura 5.3: Comparação de desempenho dos dois métodos de seleção de características.

5.1.2 Amostras caracterizadas por 3-mers

Em seguida, o S2FS foi testado com as sequências representadas por 3-mers. Como era esperado, o desempenho dos dois métodos de seleção testados foi melhor do que o desempenho obtido com os classificadores treinados com as sequências representadas por 2-mers. Isto se deve ao fato de que o número de características disponíveis para o classificador é consideravelmente maior. Na Figura 5.4, é possível ver o aumento do desempenho com o aumento do número de características escolhidas pelo algoritmo de referência.

Os resultados dos classificadores treinados com as características escolhidas pelo S2FS (Figura 5.5) apresentaram resultados que se aproximam do desempenho das classificações com os dados de treinamento mais rapidamente, ou seja, com menos características quando comparadas aos resultados obtidos com as características escolhidas pelo algoritmo de referência. Na Figura 5.6, essa diferença de desempenho pode ser mais facilmente visualizada. Uma observação interessante da Figura 5.6 é que, a partir de 18 características, o ganho de desempenho dos dois métodos de seleção de característica, ao acrescentar mais características, é muito pouco significativo quando comparado ao ganho de desempenho com menos características. Esse pode ser um indicativo do limite de características que devem ser consideradas para o *dataset* para evitar *overfitting*.

Da mesma forma que com 2-mer, foram identificadas as características mais importantes utilizando os dois métodos de seleção de características. As características listadas são as 18 mais relevantes, em ordem de importância, para um modelo de classificação

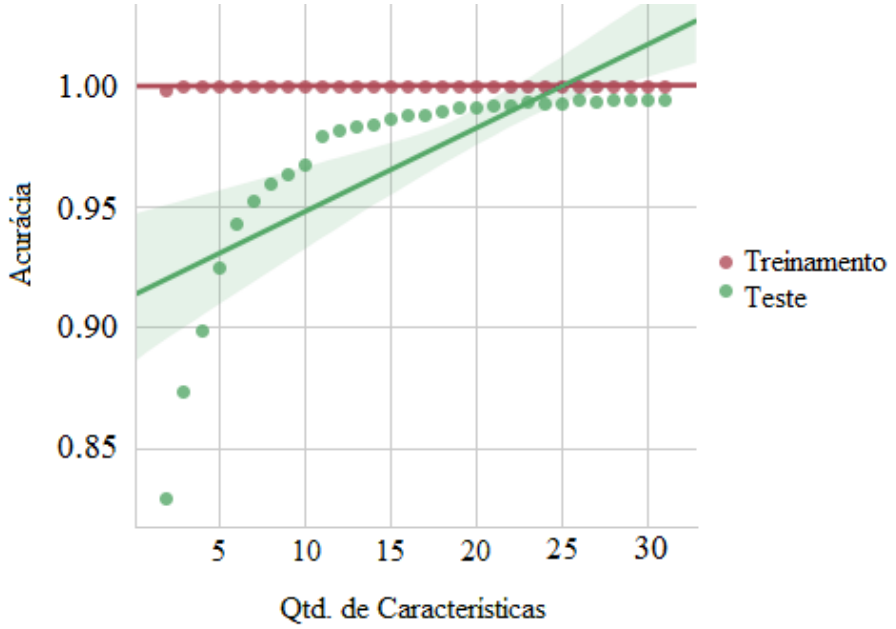

Figura 5.4: Resultado das classificações em um conjunto de dados de teste com os transcritos de *H. sapiens* caracterizados por 3-mers. Os classificadores foram treinados com as características escolhidas pelo algoritmo de referência.

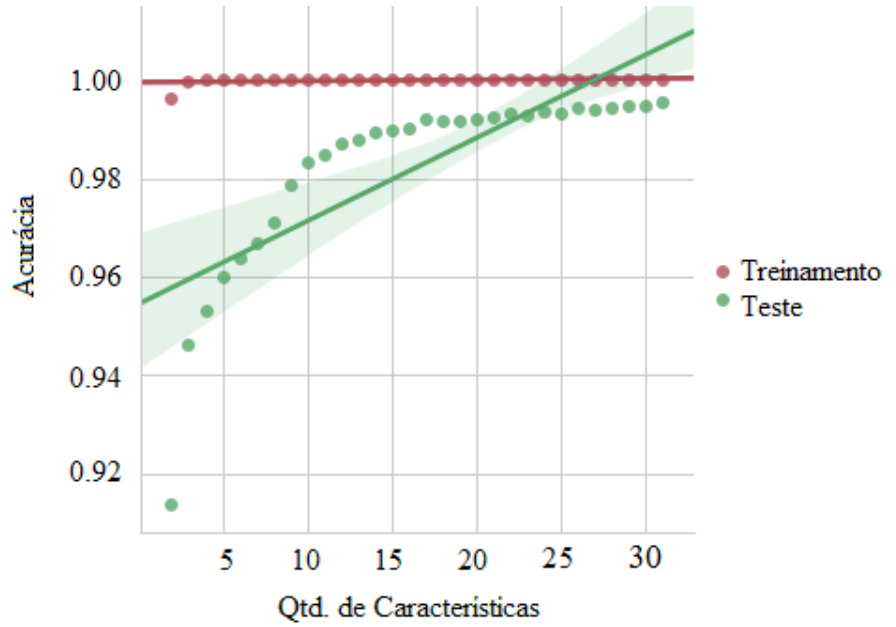

Figura 5.5: Resultado das classificações em um conjunto de dados de teste com os transcritos de *H. sapiens* caracterizados por 3-mers. Os modelos de classificação foram treinados com as características escolhidas pelo S2FS.

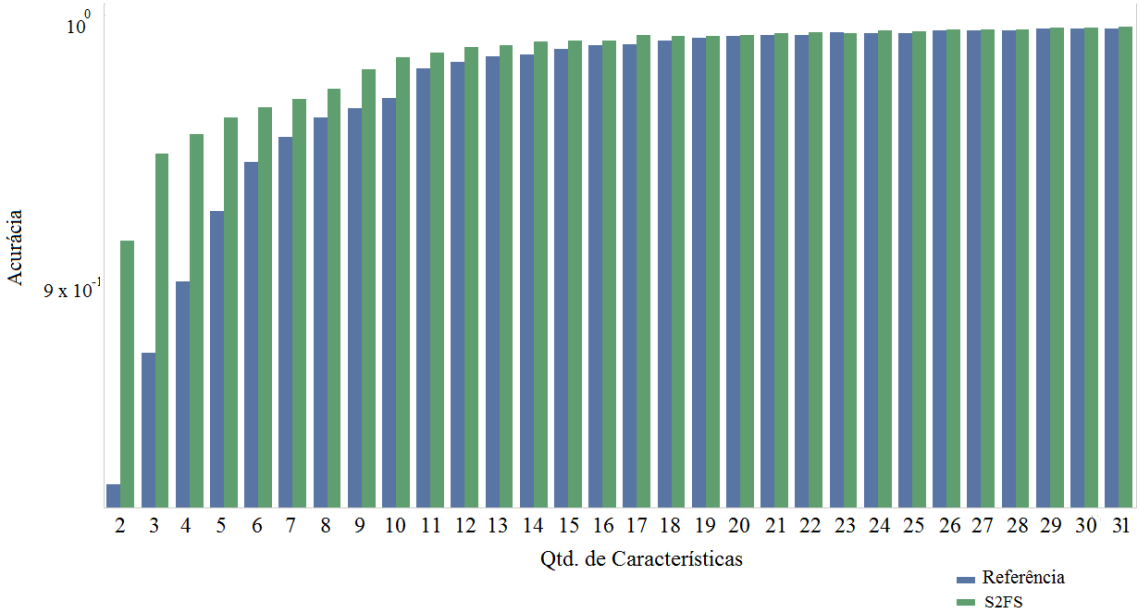

Figura 5.6: Comparação de desempenho dos dois métodos de seleção de características para as sequências caracterizadas por 3-mers.

treinado com as características escolhidas por cada um dos métodos de seleção.

Tabela 5.3: Características 3-mers mais importantes escolhidas pelo algoritmo de referência.

Características
GCG, TCG, CGC, TAC, CGA, CCG, CTT, CGG TAT, TGT, TTT, GGT, GTA, ACA, AAA, GTT CTC, GGG

Tabela 5.4: Características 3-mers mais importantes escolhidas pelo S2FS.

Características
CGC, GCG, TCG, CGA, TAC, CGT, CCG, CTT CGG, ACG, TGT, TAT, CAT, CTA, TTT, TTC GTA, ATC

Nas duas tabelas (Tabela 5.3 e Tabela 5.4), foram elencadas apenas as 18 características mais relevantes, uma vez que os gráficos mostram que não há ganhos significativos de desempenho para um número de características maior que esse limite. Além disso, uma das principais justificativas para desenvolver métodos de seleção de características é trabalhar com o menor número possível destas para reduzir a possibilidade de *overfitting* do modelo de classificação.

É possível perceber que as características elencadas pelos dois métodos são bastante semelhantes, porém a importância de cada característica para o modelo de classificação é ligeiramente diferente.

Do conjunto de características selecionadas pelo algoritmo de referência, o k-mer GCG é a característica mais importante, enquanto que do S2FS o k-mer CGC é o mais importante. Como mencionado na fundamentação teórica na Seção de seleção de características(Seção 3.4, página 31), a vantagem dessa estratégia de redução de dimensionalidade em contraste com a estratégia de extração de características é preservar as características para que possam ser avaliadas por pesquisadores do domínio para o qual as ferramentas de Aprendizado de Máquina estão sendo aplicadas.

Dessa forma, comparamos as características elencadas pelos dois métodos com as características mencionadas como relevantes e utilizadas no trabalho publicado em 2017 por Ventola et al [120].

Das características escolhidas pelo S2FS e pelo algoritmo de referência como mais relevantes, o S2FS seleciona um conjunto de características que possui mais características citadas em Ventola et al. [120] do que o conjunto de características selecionadas pelo algoritmo de referência.

Vale ressaltar que nenhum dos dois métodos de seleção testados encontraram todos as características elencadas no artigo. A hipótese é que o número de transcritos selecionados na fase inicial deste trabalho limita o número de características que podem ser utilizadas antes que comece a ocorrer *overfitting*. Dessa forma, os dois métodos de seleção tiveram que trabalhar com um número menor de características quando comparados com o número de características utilizada em Ventola et al [120]. Esses autores, por outro lado, utilizam outra abordagem e, por isso, não se preocupam com essa limitação do número de características.

5.1.3 Amostras caracterizadas por 4-mers

À medida que aumentamos o número de características, a função de um algoritmo de escolha de características passa a ser mais relevante para um bom desempenho de um classificador.

Pelo que foi observado com as sequências representadas por frequências de 3-mers, o ganho de desempenho a partir de 18 características é pouco significativo. Assim, ao caracterizar as amostras por 4-mers, torna-se necessário escolher um subconjunto de aproximadamente 20 melhores características, de um conjunto de 256 possíveis combinações de nucleotídeos. Tanto o algoritmo de referência como o S2FS obtiveram bons resultados na escolha das características.

Nos resultados apresentados nas Figuras 5.7 e 5.8, é possível perceber que, ao aumentar o número de características de frequência, a diferença entre o S2FS e o algoritmo de referência fica mais destacada.

De forma análoga às sequências caracterizadas por 3-mers, o S2FS se aproxima mais rapidamente do desempenho de treinamento ao aumentarmos o número de características, quando comparamos com o desempenho obtido com as características escolhidas pelo algoritmo de referência.

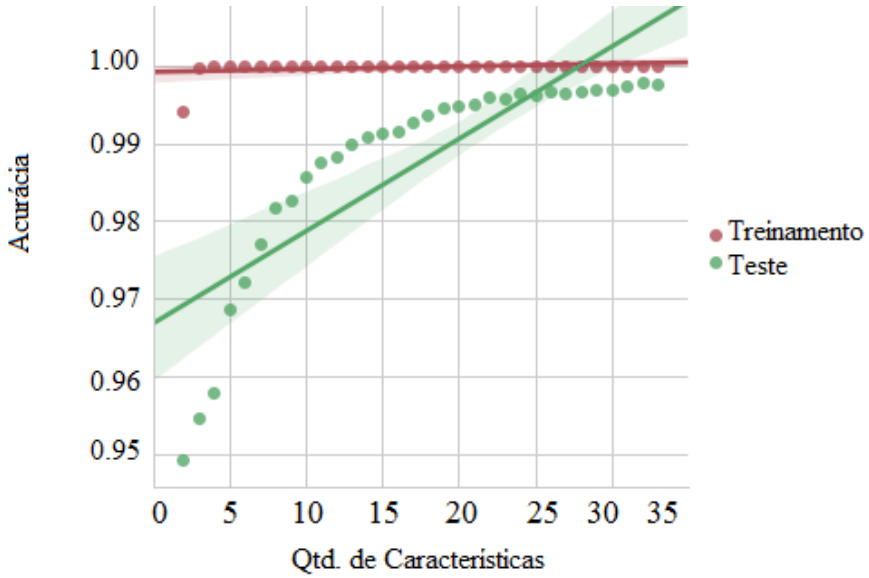

Figura 5.7: Resultado das classificações em um conjunto de dados de teste com os transcritos de *H. sapiens* caracterizados por 4-mers. Os classificadores foram treinados com as características escolhidas pelo algoritmo de referência.

A diferença entre os dois métodos fica mais aparente quando colocamos os desempenhos juntos (Figura 5.9).

Os resultados obtidos pelo S2FS com as sequências caracterizadas por 3-mer e 4-mer são bastante interessantes, pois aparentam obter um desempenho superior ao obtido pelo algoritmo de referência.

O passo seguinte foi verificar o desempenho do S2FS para selecionar características de um conjunto maior de características, ou seja, utilizando um conjunto com todas as representações dos transcritos: 2-mers, 3-mers e 4-mers e mais uma característica bastante utilizada por ferramentas de predição de transcritos ncRNAs, o tamanho da ORF mais longa de cada sequência.

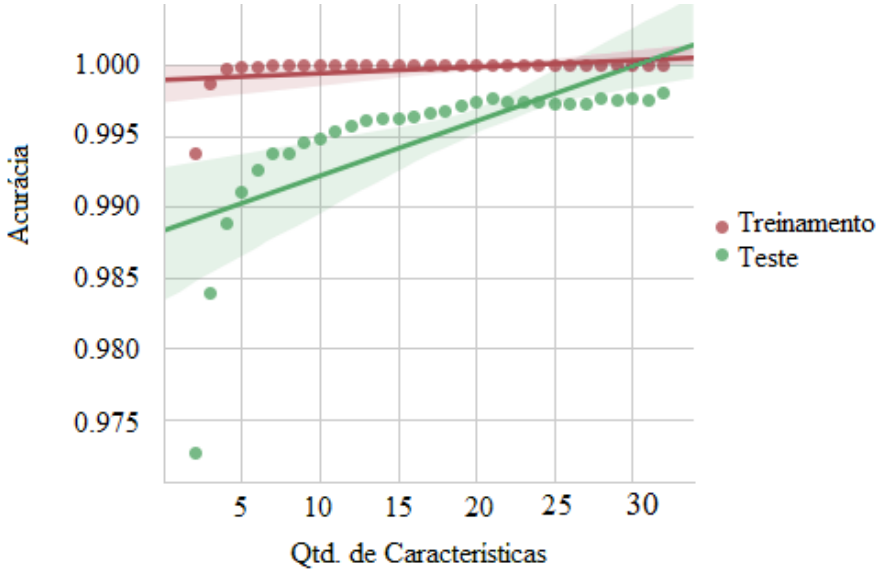

Figura 5.8: Resultado das classificações em um conjunto de dados de teste com os transcritos de *H. sapiens* caracterizados por 4-mers. Os modelos de classificação foram treinados com as características escolhidas pelo S2FS.

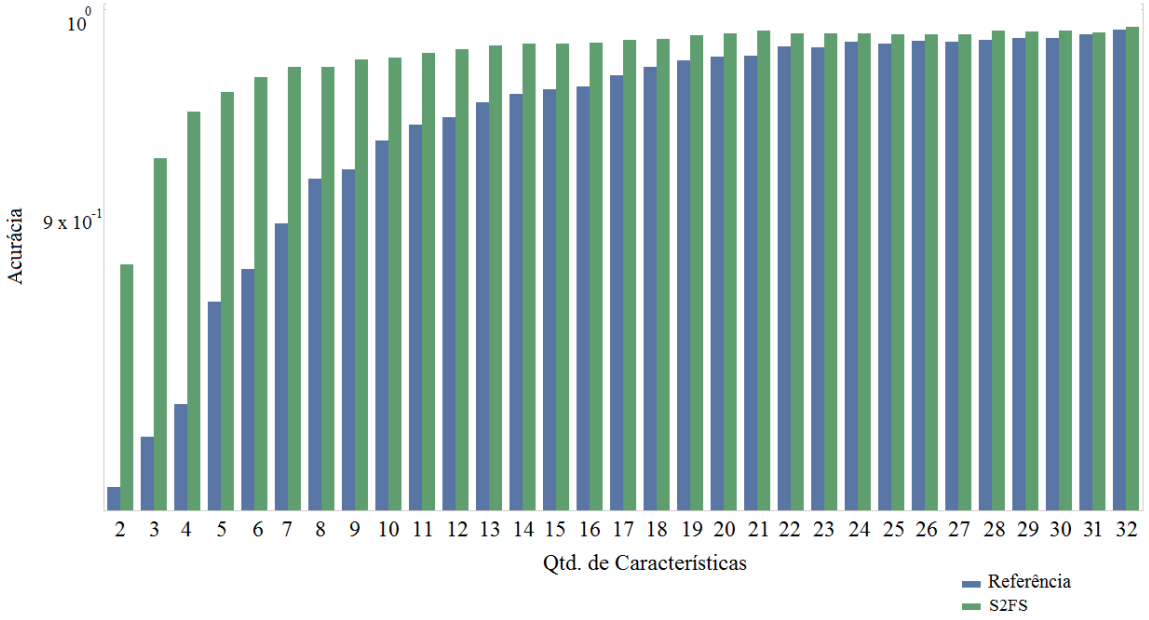

Figura 5.9: Comparação de desempenho dos dois métodos de seleção de características.

5.1.4 Amostras caracterizadas por 2-mers, 3-mers, 4-mers e tamanho de ORF

Ao utilizar o tamanho da ORF mais longa com as características de tamanho variado, os resultados de diferença de desempenho entre o S2FS e o algoritmo de referência variaram mais do que os testes realizados apenas com as características de frequência. Nas Figuras 5.10 e 5.11 nota-se que o desempenho do S2FS é melhor que a do algoritmo de referência em todos os modelos de classificação com diferentes números de características. As Tabelas 5.5 e 5.6 apresentam as características mais relevantes escolhidas do conjunto de características 2,3 e 4-mer pelo algoritmo de referência e pelo S2FS.

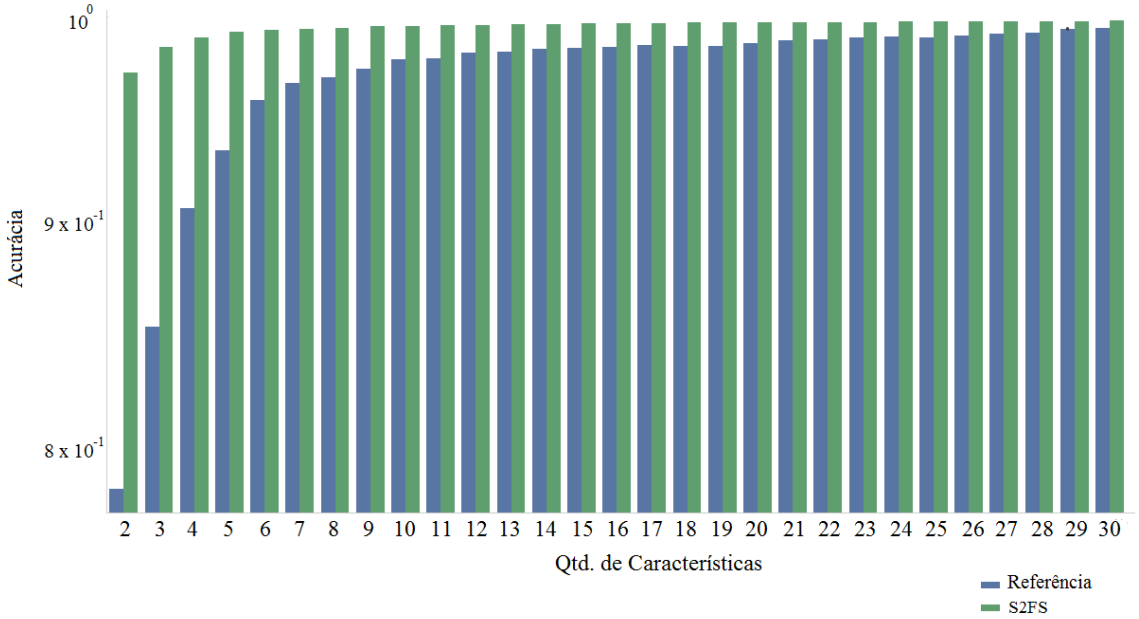

Figura 5.10: Desempenho de ambos os métodos de seleção de características para um conjunto de amostras representadas por 2-mers, 3-mers e 4-mers.

Tabela 5.5: Características 2, 3, 4-mers mais importantes escolhidas pelo algoritmo de referência ordenadas por importância para o RF.

Características
GCG, TATT, ACAC, CCG, TTTA, GTTT, TTTT, AAAA, CACA, CG, ATTT, CAGG, TTTG, ACA, AGG, GAG, CA, AG, GTT, GG, TTT, CAC, TAT, AAA, TTG, TT, TTA, TA, ATT

Por outro lado, na Figura 5.12 ocorre um comportamento diferente. Ao utilizar o tamanho da ORF mais longa como característica junto com as características 2-mer, 3-mer e 4-mer, o desempenho do algoritmo de referência é melhor do que o desempenho

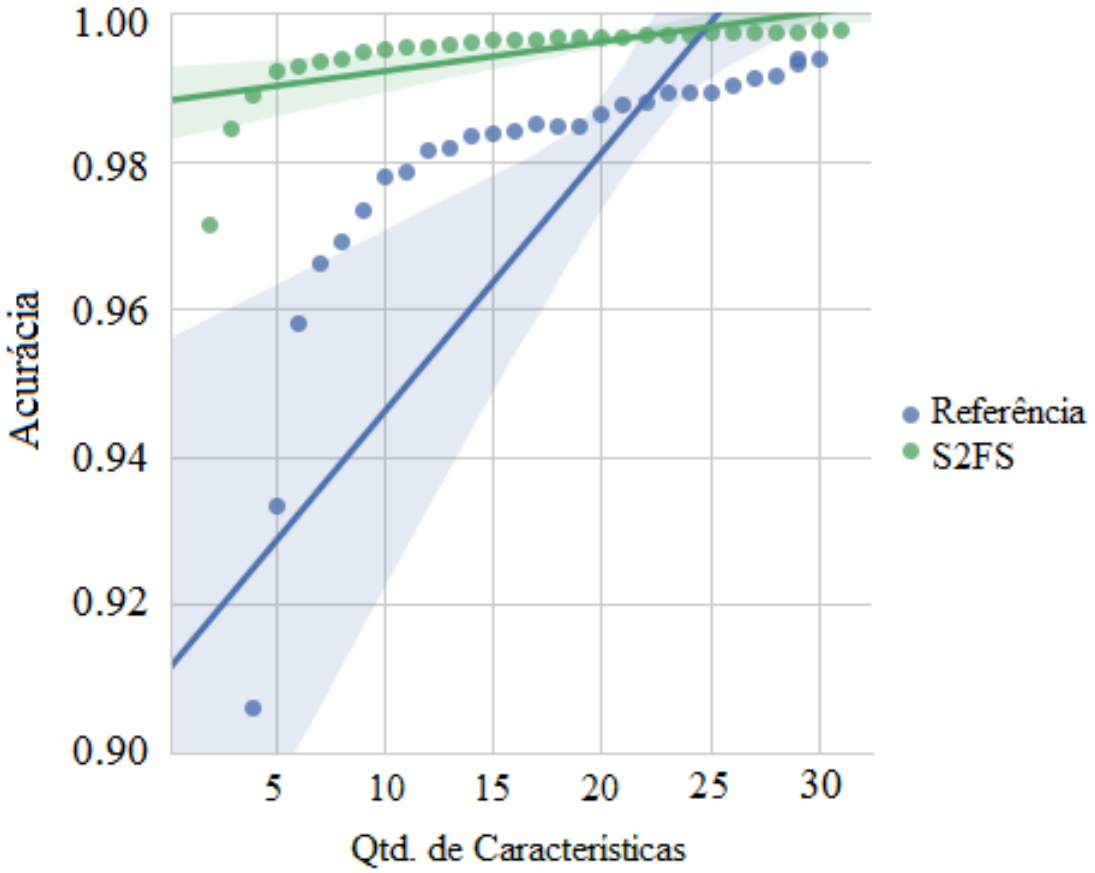

Figura 5.11: Outra representação da diferença de desempenho entre SBFS e o algoritmo de referência em modelos de classificação treinados com o conjunto de sequências de treinamento representado por 2-mer, 3-mer e 4-mer.

Tabela 5.6: Características 2, 3, 4-mers mais importantes escolhidas pelo S2FS ordenadas por importância para o RF.

Características
CGTG, GTAT, GTTA, CGC, ATAT, GCG, TTAT, TATT, CGG, TTTA, CCG, GTTT, TTTT, AAAA, CG, CTT, GT, TGT, GTT, TAT, TTT, TTG, AAA, TTA, AA, TT, AT, ATT, TA

do S2FS para os testes realizados com modelos de classificação treinados com poucas características.

A respeito desses resultados, algumas observações foram feitas. A primeira é que a etapa de pré-processamento, a qual escolheu as sequências de PCT e lncRNA, selecionou transcritos cujos valores do tamanho da ORF são muito diferentes. Enquanto que nas amostras de PCT a média do tamanho das ORFs é $1934,099 \pm 1684.11$, nas amostras lncRNAs a média dos valores para essa mesma característica é 140.20 ± 114.91. Essa

diferença tão grande impacta fortemente na primeira etapa do S2FS, na qual a pontuação do desempenho de classificação da característica ORF sozinha é muito maior que a das outras características.

A segunda observação é que ambos os métodos selecionam a ORF mais longa como característica mais importante, ou seja, a diferença de desempenho está na combinação da ORF com as características de frequência. Isso poderia ser a indicação de que a combinação de características que representam diferentes informações dos transcritos, como o tamanho da ORF e frequência de um dado k-mer quando combinadas, diminuiria a eficácia do S2FS quando aplicada a um pequeno número de características.

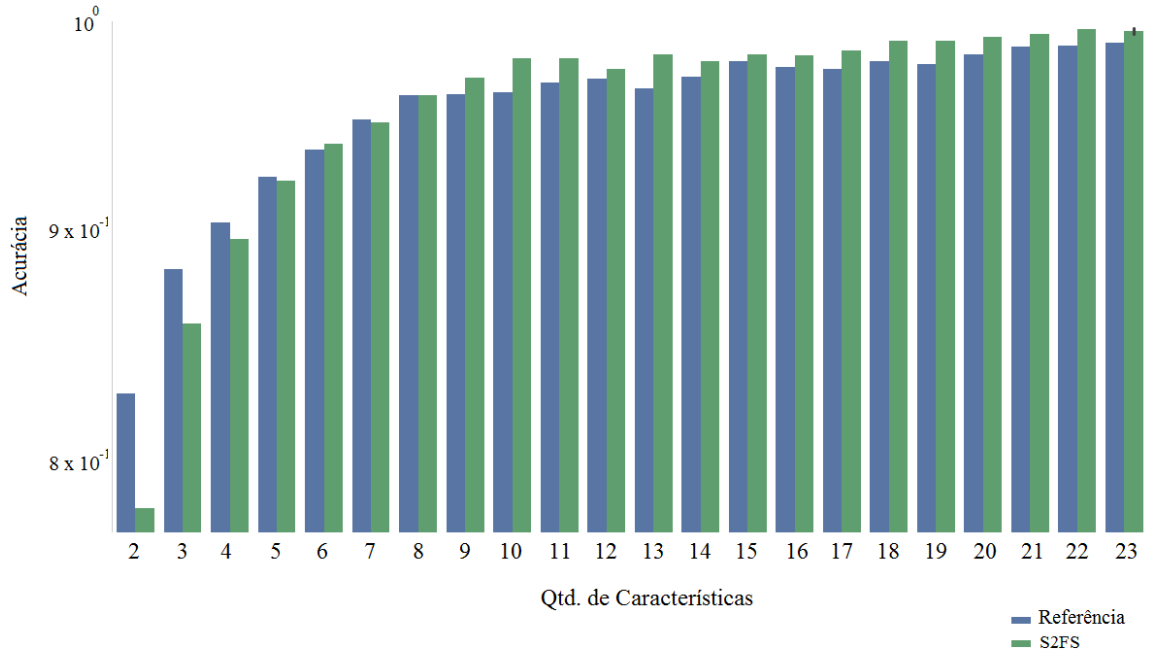

Figura 5.12: Diferença dos modelos de classificação ao utilizar a característica ORF com as características selecionadas pelo algoritmo de referência e pelo S2FS.

Para assegurar que os resultados obtidos pelo S2FS são coerentes, foi realizado um teste de correlação de Pearson com o escore de desempenho individual dos $kmer_f$ na etapa inicial de regressão logística, com o desempenho de acurácia de um classificador de duas características: tamanho da ORF e cada $kmer_f$. O objetivo do teste foi verificar se houve relação entre os resultados obtidos entre as duas etapas do método de seleção proposto.

O resultado obtido foi uma correlação de $0,6$ (Figura 5.13), indicando que existe uma correlação significativa entre os desempenhos de acurácia obtidos na fase inicial, quando utilizamos cada $kmer_f$ separadamente, e quando utilizamos este $kmer_f$ e o tamanho de

ORF como características para de um modelo de classificação. Em outras palavras, o desempenho individual é um bom critério para escolha das melhores características.

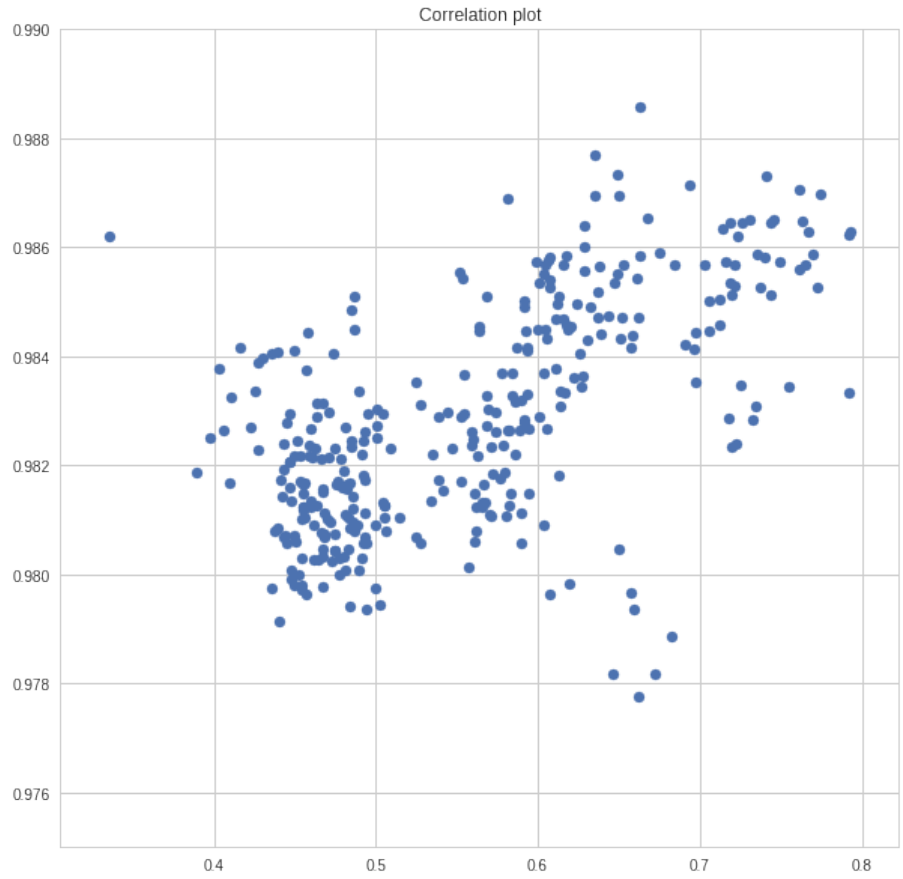

Figura 5.13: Correlação entre a acurácia de cada $kmer_f$ e a acurácia de um modelo que combina cada $kmer_f$ com o tamanho da ORF mais longa. É possível observar que existe uma correlação significativa entre as acurácias obtidas.

Para verificar se o S2FS estava escolhendo características que treinaria modelos com melhor desempenho não apenas com o algoritmo RF, utilizaram-se também os algoritmos SVM e *K-nearest neighbors* (KNN) com 30 características de frequência 2, 3 e 4-mer escolhidas pelo S2FS e pelo algoritmo de referência. Os resultados desses testes estão apresentados nas Figuras 5.14 e 5.15.

Nas Figuras 5.14 e 5.15, é possível verificar que o S2FS escolheu um conjunto de características que possibilitou um desempenho ligeiramente melhor tanto com o classificador KNN quanto com o classificador SVM. Esses dois testes dão um indicativo de que o método proposto de seleção funcionaria de maneira independente do classificador que se quer utilizar.

Univariate Feature Selection

```
Test
Accuracy:Test 0.868466666667
Confusion Matrix
             precision    recall  f1-score   support

    class 0       1.00      0.74      0.85     15103
    class 1       0.79      1.00      0.88     14897

avg / total       0.90      0.87      0.87     30000

          TRUE LABEL
         PC       NC
PC       14882    3931
NC       15       11172
```

S2FS

```
Test
Accuracy:Test 0.878633333333
Confusion Matrix
             precision    recall  f1-score   support

    class 0       1.00      0.76      0.86     15103
    class 1       0.80      1.00      0.89     14897

avg / total       0.90      0.88      0.88     30000

          TRUE LABEL
         PC       NC
PC       14882    3626
NC       15       11477
```

Figura 5.14: Comparação de desempenho dos dois métodos de seleção de características, seleção univariada e S2FS, com 30 características utilizando um classificador KNN.

Univariate Feature Selection

```
Test
Accuracy:Test 0.807866666667
Confusion Matrix
             precision    recall  f1-score   support

    class 0       0.90      0.70      0.79     15103
    class 1       0.75      0.92      0.83     14897

avg / total       0.82      0.81      0.81     30000

          TRUE LABEL
         PC      NC
PC       13683   4550
NC       1214    10553
```

S2FS

```
Test
Accuracy:Test 0.811766666667
Confusion Matrix
             precision    recall  f1-score   support

    class 0       0.88      0.72      0.79     15103
    class 1       0.76      0.90      0.83     14897

avg / total       0.82      0.81      0.81     30000

          TRUE LABEL
         PC      NC
PC       13424   4174
NC       1473    10929
```

Figura 5.15: Comparação de desempenho dos dois métodos de seleção de características, seleção univariada e S2FS, com 30 características utilizando um classificador SVM.

5.2 LincRNAs em *H. sapiens*

Ao verificar as características 3-mers mais relevantes listadas pelo modelo de classificação na Seção anterior, foi possível observar uma predominância de características como ACG, CCG, CGA, CGG, CGT, GCG, TAA, TAC, TCG e TAG, que são trinucleotídeos fortemente relacionados a estudos de classificação de lincRNAs [108, 82]. Logo, levantou-se a hipótese que uma parte considerável dos lncRNAs escolhidos na etapa de seleção das sequências fossem de lincRNAs.

Assim, de forma a verificarmos essa hipótese, o *pipeline* de seleção de sequências e características relevantes foi novamente aplicado ao *dataset* de sequências de *H. sapiens*, mas filtrando as sequências de lncRNAs, para que fossem utilizadas apenas as classificadas como lincRNAs validadas pelo banco de dados HAVANA.

Comparando os resultados de desempenho dos classificadores com as características escolhidas com o algoritmo de referência e o S2FS, para o *dataset* composto de PCTs e lincRNAs, é possível observar que os resultados foram semelhantes àqueles obtidos com os outros *datasets*. O S2FS, conforme pode ser observado na Figura 5.16, obteve um desempenho levemente superior ao algoritmo de referência, quando utilizado um subconjunto de todo o conjunto de características possíveis (2-mers, 3-mers e 4-mers).

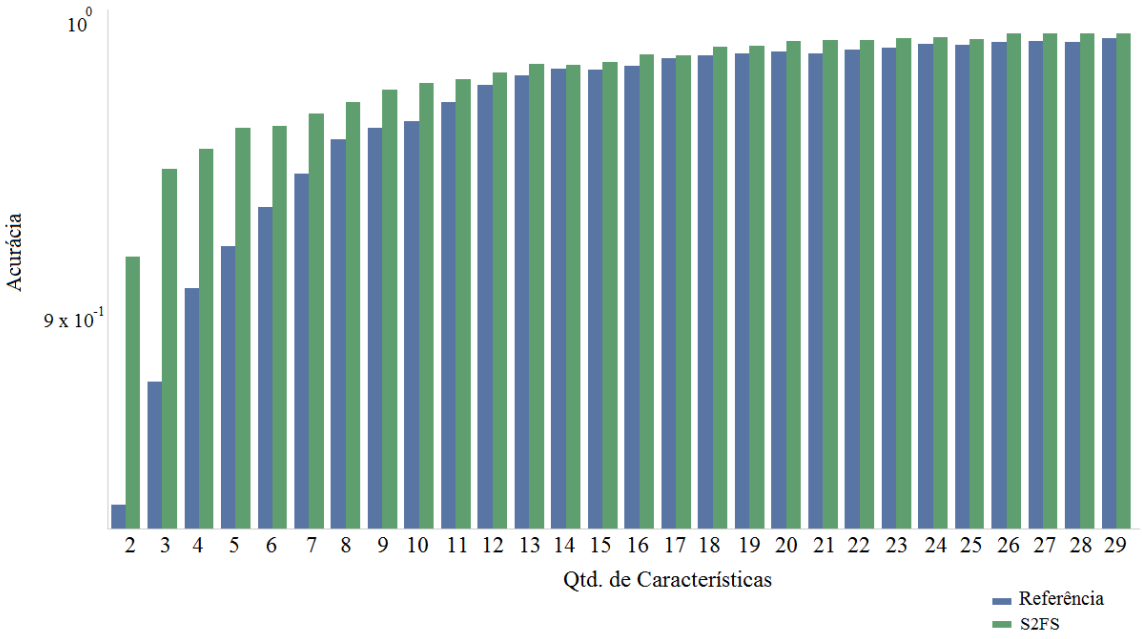

Figura 5.16: Comparação de desempenho dos dois métodos de seleção de características para amostras caracterizadas por 2-mer, 3-mer e 4-mer para o *dataset* composto de PCTs e lincRNAs de *H. sapiens*.

Um fato que pode ser observado com este *dataset* menor de lincRNAs é que o número de características limite para o qual há ganho de desempenho ao acrescentar novas características é menor do que o número limite para o *dataset* com as sequências de PCTs e lncRNAs. Esse é um resultado esperado, uma vez que, se o número de sequências disponíveis para treinamento e teste é menor, o número de características necessárias para a criação de um modelo sem *overfitting* também deve ser menor, segundo os princípios de dimensionalidade de dados vistos na Seção 3.4.

A Tabela 5.7 apresenta as características mais relevantes escolhidas do conjunto de características 2, 3 e 4-mer pelo S2FS para o *dataset* composto de PCTs e lincRNAs.

Tabela 5.7: Características 2, 3 e 4-mers mais relevantes escolhidas pelo S2FS.

Características
GCGG, CGGC, GCG, TTTT, CCG, CGG, CGC, TCG, AAAA, TTTA, CGA, ACG, TTGT, CG, TAT, TGTT, TTA, GTA, TTT, TAC, ATT, GTT, GTG, TGT, TTG, AGT, TA, TT, AT, GC

5.3 *M. musculus*

Conforme apresentado no capítulo de dados e métodos, o *dataset* de *Mus musculus*, obtido da montagem GRCm38 da base de dados Ensembl, é composto de 90.854 transcritos codificadores e 12.528 ncRNAs, ou seja, é significativamente menor que o *dataset* de *Homo sapiens*.

Por se tratar de um *dataset* menor, o número máximo de características que devem ser utilizadas sem causar *overfitting* do modelo também deve ser menor.

Os melhores resultados do S2FS foram obtidos ao selecionar as características dos dois maiores conjuntos de frequências possíveis, ou seja, conjunto de características 4-mer e o conjunto de que combina as frequências 2-mer, 3-mer e 4-mer.

5.3.1 Amostras caracterizadas por 2-mers

De forma semelhante aos resultados dos testes realizados com amostras caracterizadas por 2-mers para *H. sapiens*, com o *dataset* de *M. musculus*, a diferença de desempenho obtida com as características selecionadas pelo S2FS e pelo algoritmo de referência oscila, ou seja, para alguns números de características o S2FS escolhe características que obtêm melhor desempenho quando testadas no modelo de classificação. Para outros, o algoritmo de referência seleciona um melhor conjunto de características (Figuras 5.17, 5.18 e 5.19).

Esse resultado também ocorre com o *dataset* de *Danio rerio*, conforme apresentado na próxima Seção. Podemos levantar como hipótese que o S2FS não garante a obtenção de resultados de desempenho melhor que o algoritmo de referência para *datasets* caracterizados apenas por 2-mer.

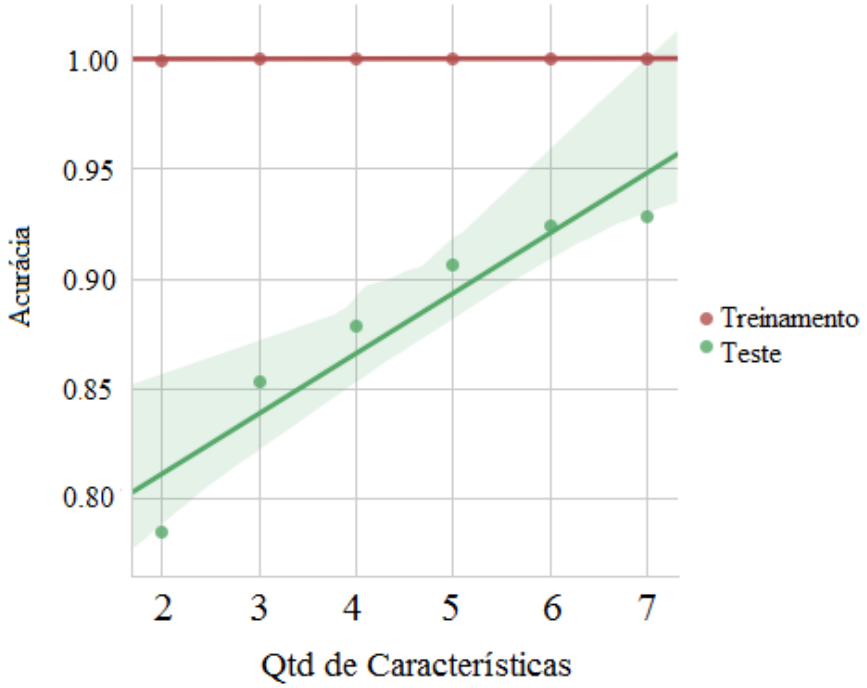

Figura 5.17: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência aplicados ao conjunto de dados de teste. *Dataset* de *M. musculus* representado por características 2-mers.

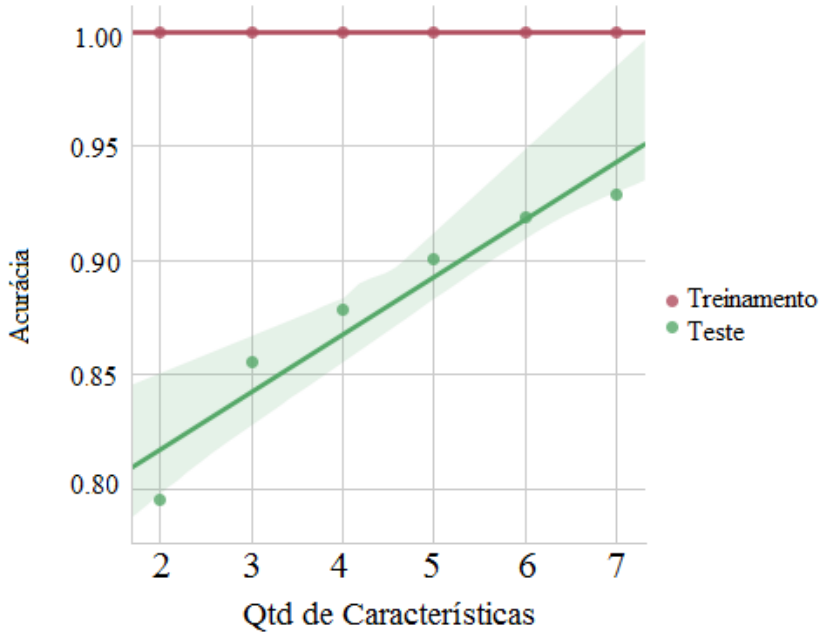

Figura 5.18: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS aplicados ao conjunto de dados de teste. *Dataset* de *M. musculus* representado por características 2-mers.

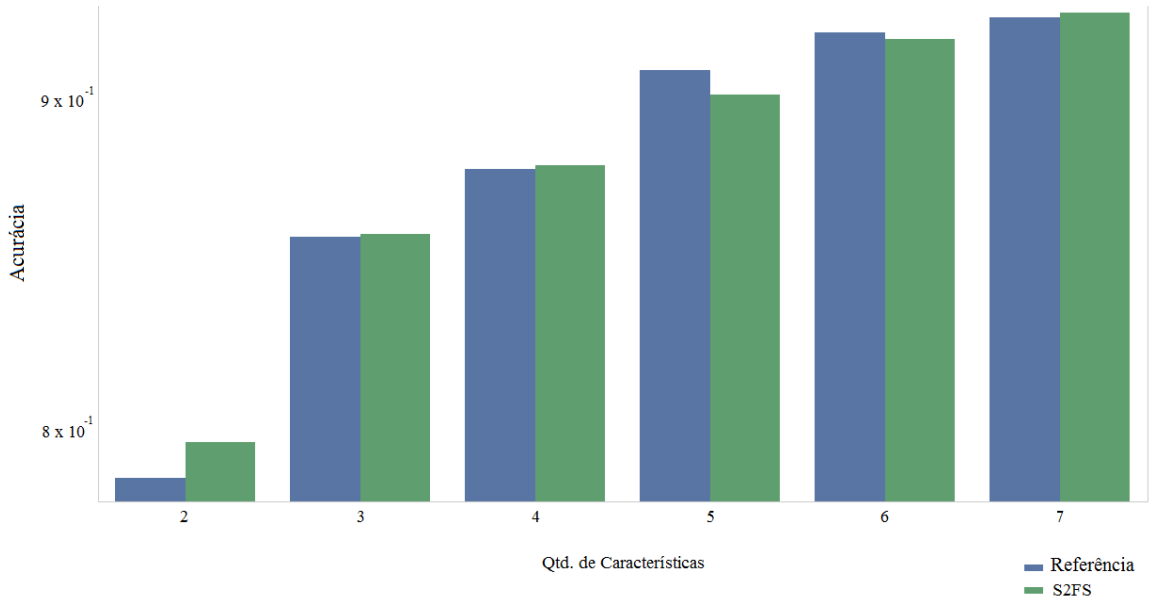

Figura 5.19: Comparação de desempenho dos modelos de classificação utilizando as características escolhidas pelos dois métodos de seleção com o *Dataset* de *M. musculus*.

De qualquer forma, o número de características possíveis utilizando apenas 2-mers não seria suficiente para construir modelos de classificação de transcritos com uma acurácia

adequada.

No entanto, os resultados ainda são interessantes, uma vez que, com poucas características, é possível observar que a importância que um algoritmo como o *Random Forest* estabelece para cada característica varia ao permutar algumas das características. Nas Tabelas 5.8 e 5.9, é possível observar que 4 das 7 características coincidem, mas a ordem delas varia. Então, mesmo que o método de seleção de características selecione uma característica por vez, o desempenho do modelo de classificação utilizado depende da combinação das características.

Tabela 5.8: Características 2-mer do *dataset Mus musculus* mais importantes escolhidas pelo algoritmo de referência. A ordem das características determina a importância que o algoritmo *Random Forest* atribui a cada característica. Nesta tabela, por exemplo, CG é a característica mais importante e AG a menos importante.

Características
CG, TT, GT , GA, TA, CT, AG

Tabela 5.9: Características 2-mer do *dataset Mus musculus* mais importantes escolhidas pelo S2FS. A ordem das características determina a importância que o algoritmo *Random Forest* atribui a cada característica. Nesta tabela, por exemplo, CG é a característica mais importante e TG a menos imporante.

Características
CG, TA, TT, AT, GC, GT, TG

Uma última observação pode ser feita em relação às características selecionadas pelos dois métodos de seleção de características. Ao comparar com as características relacionadas por Ventola et al. [120], para o *dataset* em questão, todas as características selecionadas pelo S2FS estão citadas no trabalho. Das características selecionadas pelo algoritmo de referência, cinco das sete características estão citadas como características relevantes.

O artigo citado acima considera 12 características de 2-mers, das 16 possíveis, como relevantes na composição de suas diversas assinaturas. Como o número de características foi fixado em sete para ambos os métodos, devido à etapa de clusterização que removeu algumas características que não obtiveram pontuação suficiente, os dois métodos, para 2-mer, não relacionaram todas as características citadas no artigo.

5.3.2 Sequências caracterizadas por 3-mers

Os resultados obtidos com 3-mers tiveram um comportamento mais estável, com o S2FS obtendo uma vantagem considerável em relação ao algoritmo de referência para duas e

três características. Para um número maior de características, os dois métodos obtiveram resultados muito próximos de desempenho.

Uma observação a respeito dos resultados obtidos com o *dataset* de *Mus musculus* para esta representação por 3-mers é que, apesar de o gráfico de ganho de desempenho ser semelhante ao gráfico do *dataset* de *H. sapiens*, o valor limite para acrescentar novas características é menor do que o limite encontrado no *dataset* de *H. Sapiens*. Esse é um resultado esperado, uma vez que, como o *dataset* de *M. musculus* é menor, ou seja, possui menos transcritos, uma quantidade menor de características é necessária para classificar essas sequências sem que haja *overfitting*.

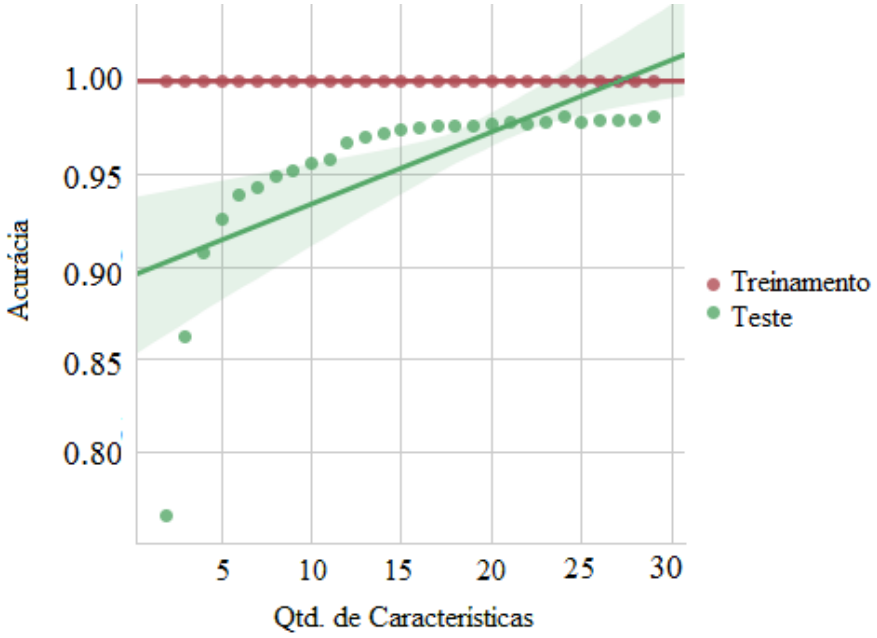

Figura 5.20: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 3-mers.

As Tabelas 5.10 e 5.11 apresentam as características mais relevantes ordenadas por ordem de importância para o modelo de classificação aplicado ao *dataset* de *Mus musculus*.

Diferente do caso com os transcritos representados por características de frequência 2-mer, a quantidade de características que foram consideradas pelos dois métodos é maior do que as características apresentadas em Ventola et. al [120]. Como consequência, ambos os métodos selecionaram as principais características mencionadas no artigo como relevantes.

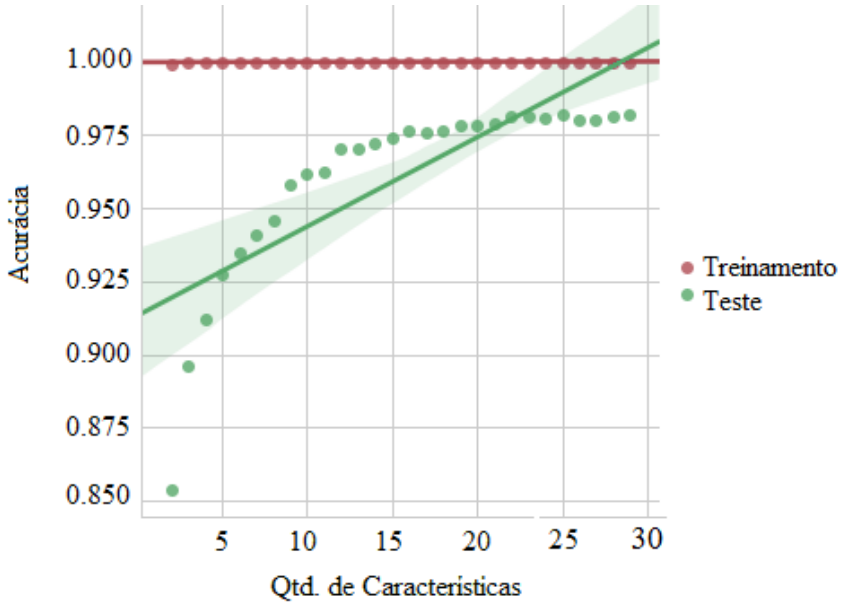

Figura 5.21: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 3-mers.

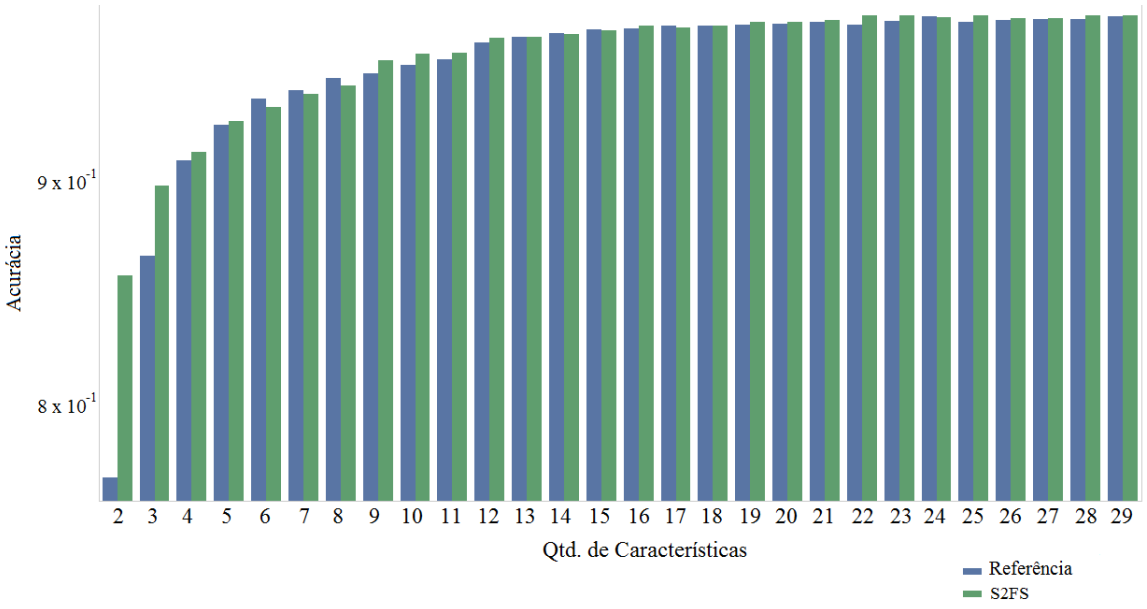

Figura 5.22: Comparação de desempenho dos dois métodos de seleção de características.

Tabela 5.10: Características 3-mer do *dataset Mus musculus* mais importantes escolhidas pelo algoritmo de referência. A ordem das características determina a importância que o modelo de classificação atribui a cada característica.

Características
TAC, TAT, CCG, GCG, CGC, TCG, CGG, CGT, GTA ACG, AGT, CGA, ACT, GTG, TGT, TTT, ATT, GTT, CTC, ACA, GGA, CTG, TTA, AGG, CAC, TCT, AGA, GAA, GAG, AAG

Tabela 5.11: Características 3-mer do *dataset Mus musculus* mais importantes escolhidas pelo S2FS. A ordem das características determina a importância que o modelo atribui a cada característica.

Características
TAC, GCG, GTA, CCG, TAT, CGG, TCG, CGC, CGT, CGA, AGT, TGT, ACG, GTG, ATT, TTT, GGT, GTT, TTG, GTC, GCA, TTA, ATA, GCC, AGC, CCC, GGC, GGG, AAT, TAG

5.3.3 Sequências caracterizadas por 4-mers

A diferença de desempenho entre os métodos de seleção estudados torna-se mais evidente quando os transcritos estão representados por 4-mers, ou seja, quando o conjunto do qual os métodos de seleção devem escolher as melhores características é maior. O S2FS possui uma vantagem considerável em relação ao algoritmo de referência em todos os testes realizados com os modelos de classificação conforme pode ser visto nas Figuras 5.23, 5.24 e 5.25.

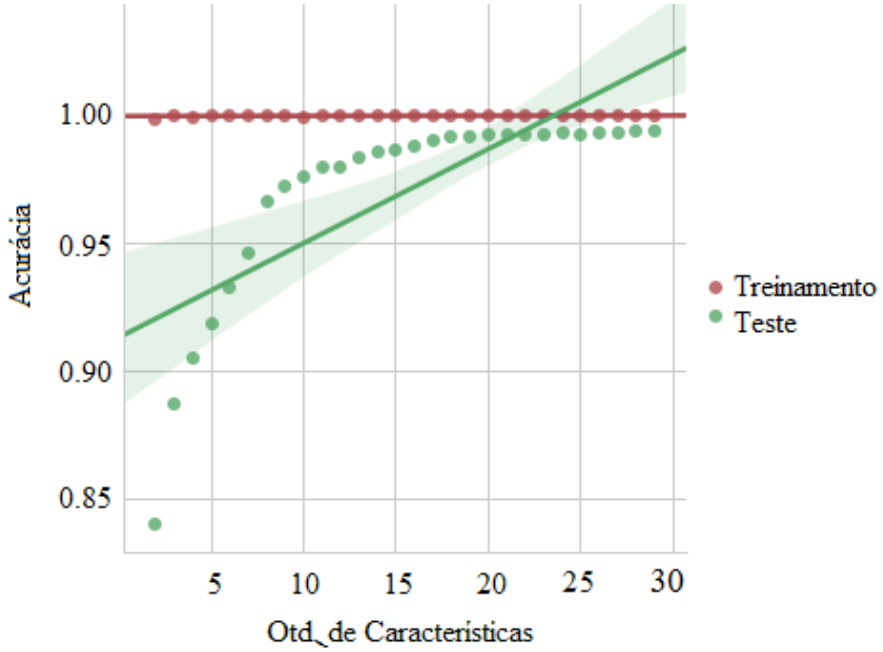

Figura 5.23: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 4-mers.

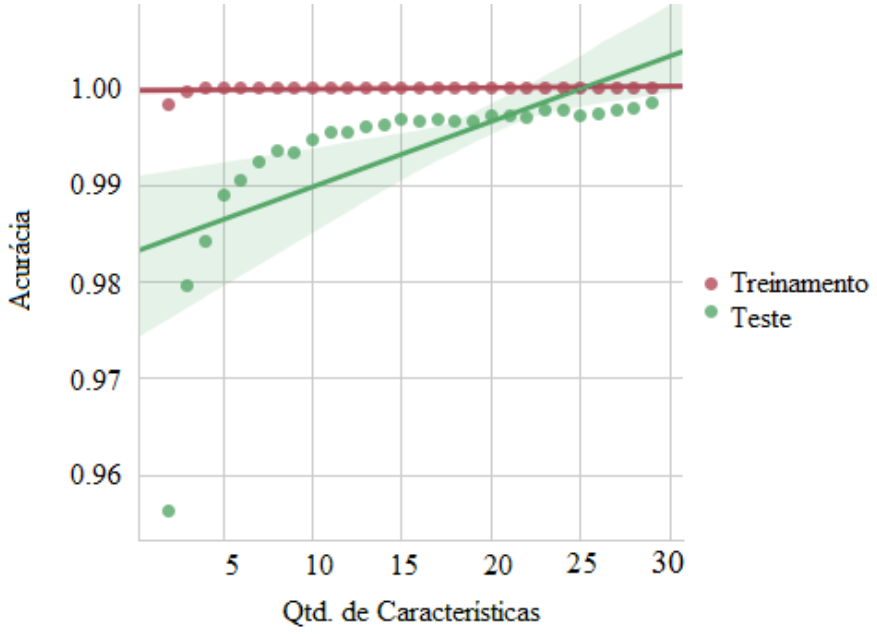

Figura 5.24: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 4-mers.

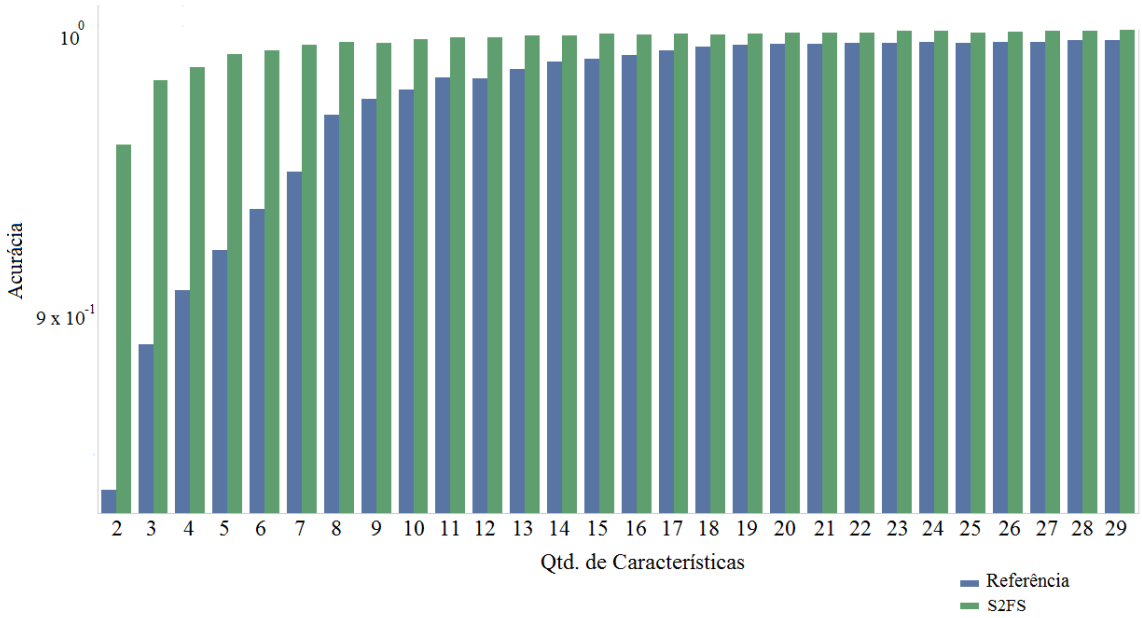

Figura 5.25: Comparação de desempenho dos dois métodos de seleção de características.

5.3.4 Sequências caracterizadas por 2-mers, 3-mers e 4-mers

Ao utilizar o conjunto total de características disponíveis para as sequências, 336 características, o S2FS apresenta resultados consideravelmente melhores do que o algoritmo de referência.

Ao comparar os resultados obtidos com as sequências caracterizadas por 2-mer, 3-mer e 4-mer com os obtidos com as sequências caracterizadas apenas por 4-mer, podemos observar que o S2FS teve um desempenho melhor. O método proposto, conforme pode ser visto nas Figuras 5.26 e 5.27, selecionou características que geraram uma diferença ainda maior de desempenho nos modelos de classificação, quando comparado ao desempenho obtido com as características escolhidas pelo algoritmo de referência. Em outras palavras, o S2FS teve um desempenho melhor ao escolher características de um conjunto maior de possíveis características, nas quais muitas delas estão correlacionadas entre si(Figura 5.28).

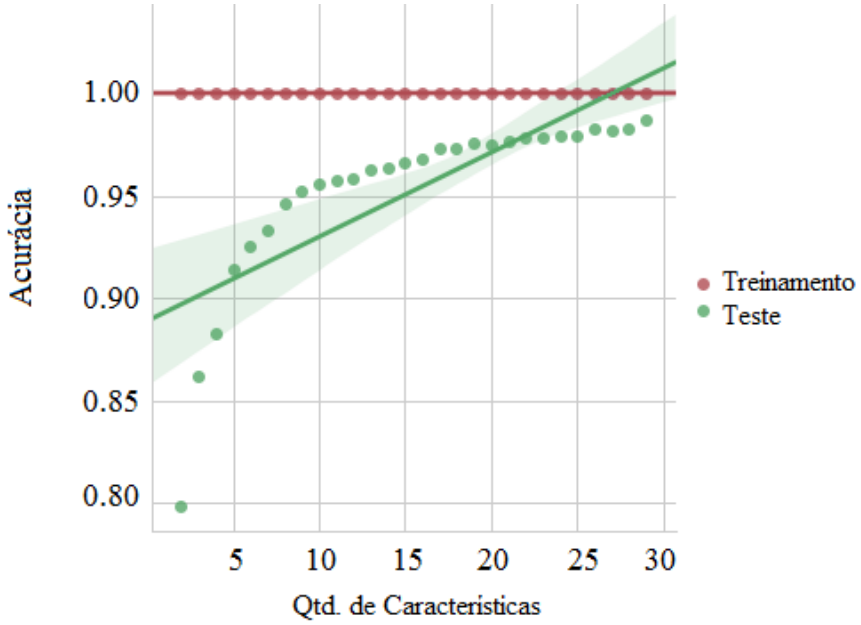

Figura 5.26: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 2-mers, 3-mers e 4-mers.

Outra observação em relação a esses resultados pode ser feita em relação às características que cada método seleciona como mais relevantes. No algoritmo de referência, características de 2-mers, 3-mers e 4-mers foram escolhidas como características relevantes. No entanto muitas das características escolhidas pelo algoritmo de referência são

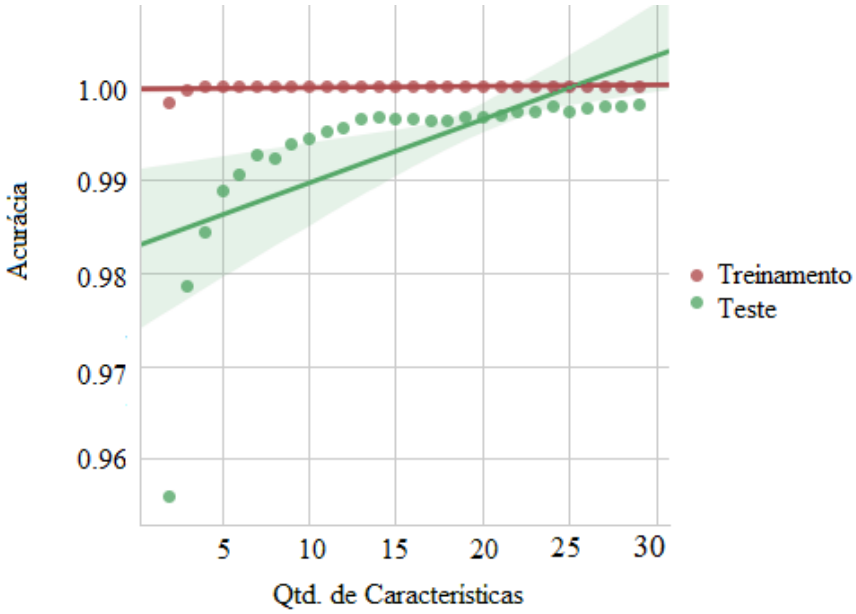

Figura 5.27: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset M. musculus* com características 2-mers, 3-mers e 4-mers.

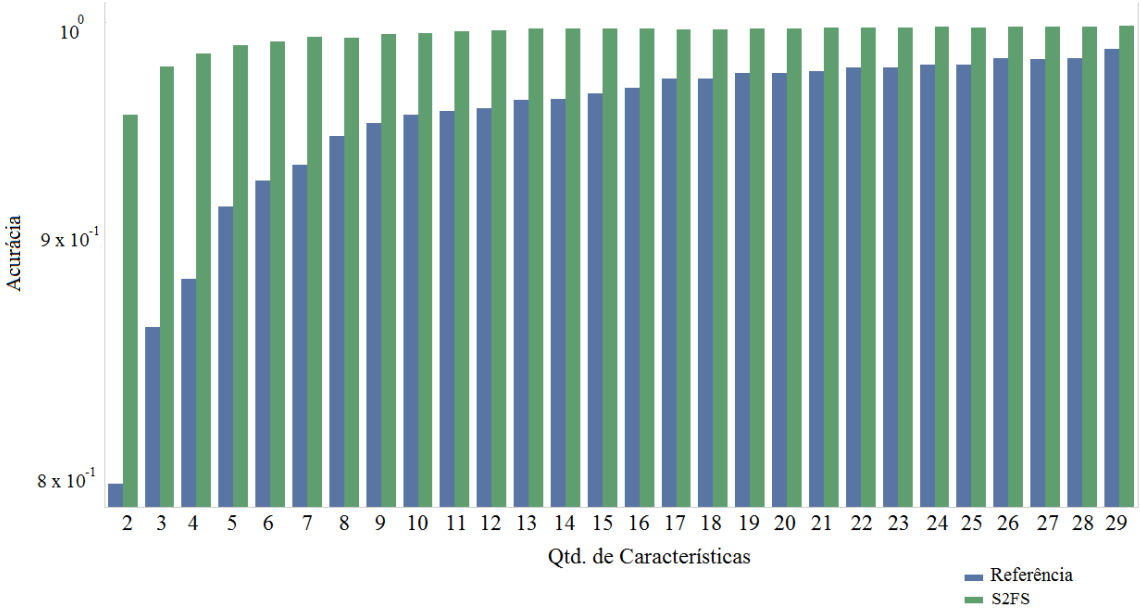

Figura 5.28: Comparação de desempenho dos dois métodos de seleção de características.

correlacionadas entre si como TT, TTT, TTTT ou CG, CGC, ou TA, TTA, TTTA, entre outras.

O S2FS, por outro lado, escolhe predominantemente características 4-mer, reduzindo a possibilidade de escolher características fortemente correlacionadas. As cinco características mais relevantes selecionadas pelo S2FS, por exemplo, são TACG, GTCG, TCGT, TCGC e CGAC. Todas as cinco características possuem CG em sua composição, mas nenhuma das características está inteiramente contida na outra, como é o caso de TT e TTTT escolhidos na seleção do algoritmo de referência. As Tabelas 5.12 e 5.13 apresentam as características mais importantes selecionadas pelo S2FS e pelo algoritmo de referência.

Tabela 5.12: Características 2, 3, 4-mers mais importantes escolhidas pelo algoritmo de referência.

Características
CGGC, CGC, TTAT, GCG, TTTT, GTA, TATT, CTGA, GT, TCG, CCG, TAT, CG, CGT, CGG, TTTA, CGA, ACG, TTT, AGGA, AAGA, CT, TTA, CTC, TA, TT, AGA, GGA, GA

Tabela 5.13: Características 2, 3, 4-mers mais importantes escolhidas pelo S2FS.

Características
TACG, GTCG, TCGT, TCGC, CGAC, GCGC,TTCG, TCGA, CGGT, GCGA, AACG, ACGA, GACG, ATCG, CGTC, CCGT, TGCG, ACGT, CGAA, ACCG, GGCG, CGCA, CGCC, TCGG, GCCG, CCGC, CCGA, GCGG, CGCT

5.4 *D. rerio*

O *dataset* de *D. rerio* é o menor dos *datasets* testados. É composto de 50.731 transcritos codificadores e 3.976 transcritos ncRNAs. Dessa forma, os resultados obtidos devem ser analisados levando em consideração que foram utilizados poucos transcritos em comparação com a quantidade de transcritos utilizados nos outros *datasets*.

Os testes realizados com *dataset* de *Danio rerio* seguiram a mesma metodologia dos outros *datasets*: inicialmente com as sequências caracterizadas por 2-mers, depois por 3-mers, 4-mers e por último, com todas as características de frequência juntas: 2-mers, 3-mers e 4-mers.

5.4.1 Amostras caracterizadas por 2-mers

Assim como nos outros *datasets*, os resultados obtidos utilizando apenas características de frequência 2-mer não foram muito satisfatórios. Primeiramente porque desempenho de classificação dos modelos treinados utilizando apenas as características 2-mer foi comparativamente menor do que quando os modelos de classificação foram treinados com as outras características de frequências 3-mer e 4-mer.

Outro problema em utilizar apenas características 2-mer é que a diferença dos resultados dos modelos de classificação treinados com as características escolhidas pelos dois métodos de seleção não é muito estável. Para alguns conjuntos de características, o algoritmo de referência obtém um desempenho ligeiramente melhor; para outros, o S2FS. Em outras palavras, não há como determinar qual método de seleção é melhor para poucas características.

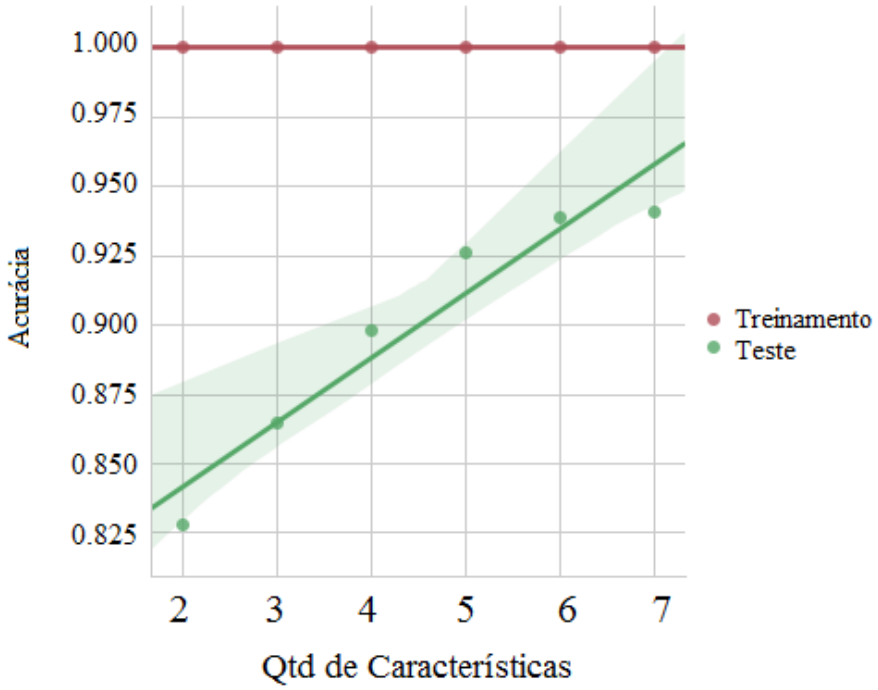

Figura 5.29: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 2-mers.

Conforme pode ser observado nas Figuras 5.29, 5.30 e 5.31, para *datasets* pequenos com poucas sequências e poucas características, a diferença de desempenho entre os dois métodos de seleção é pequena e não muito estável. Esse padrão pôde ser observado em todos os *datasets* testados neste trabalho.

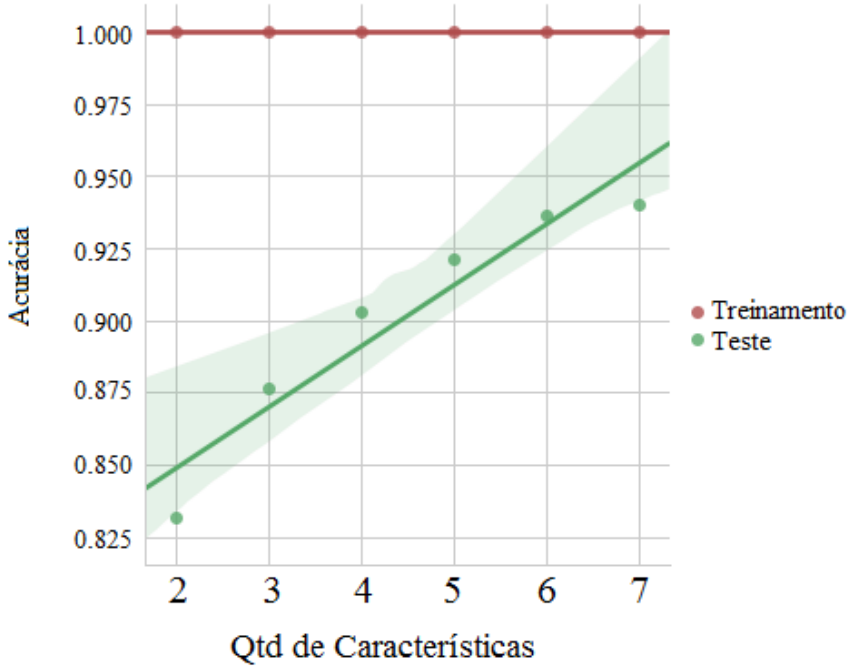

Figura 5.30: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 2-mers.

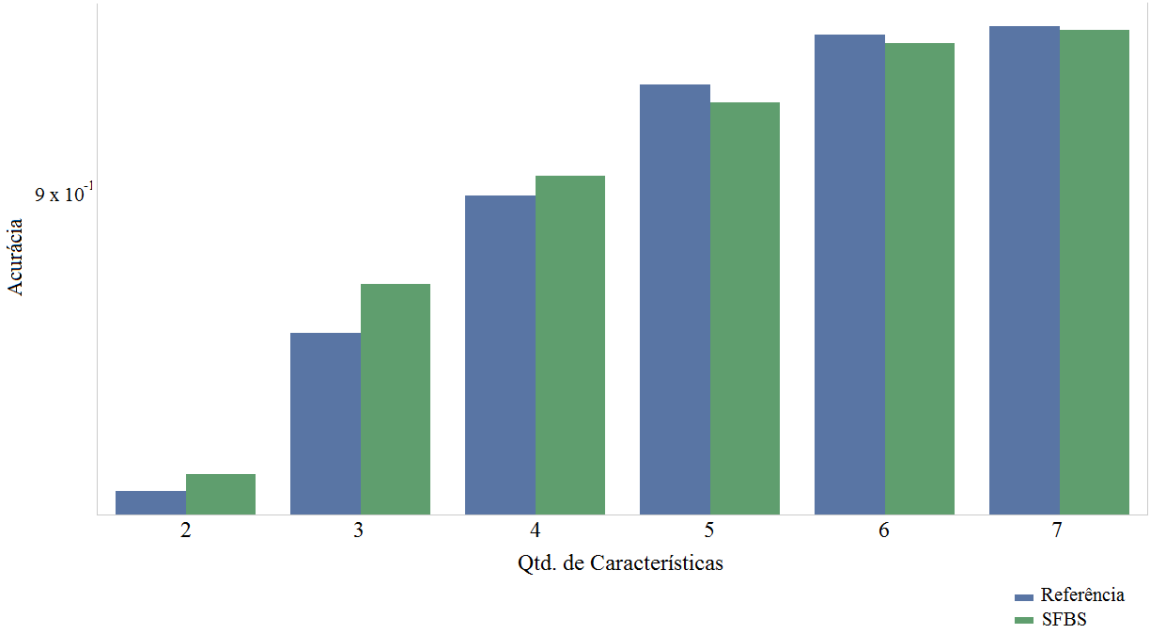

Figura 5.31: Comparação de desempenho dos dois métodos de seleção de características.

5.4.2 Sequências caracterizadas por 3-mers

Quando as sequências do *dataset* de *Danio rerio* estão representadas pelas características de frequências de 3-mers, as diferenças entre os dois métodos de seleção de característica ficam mais aparentes, conforme pode ser visto nas Figuras 5.32, 5.33 e 5.34.

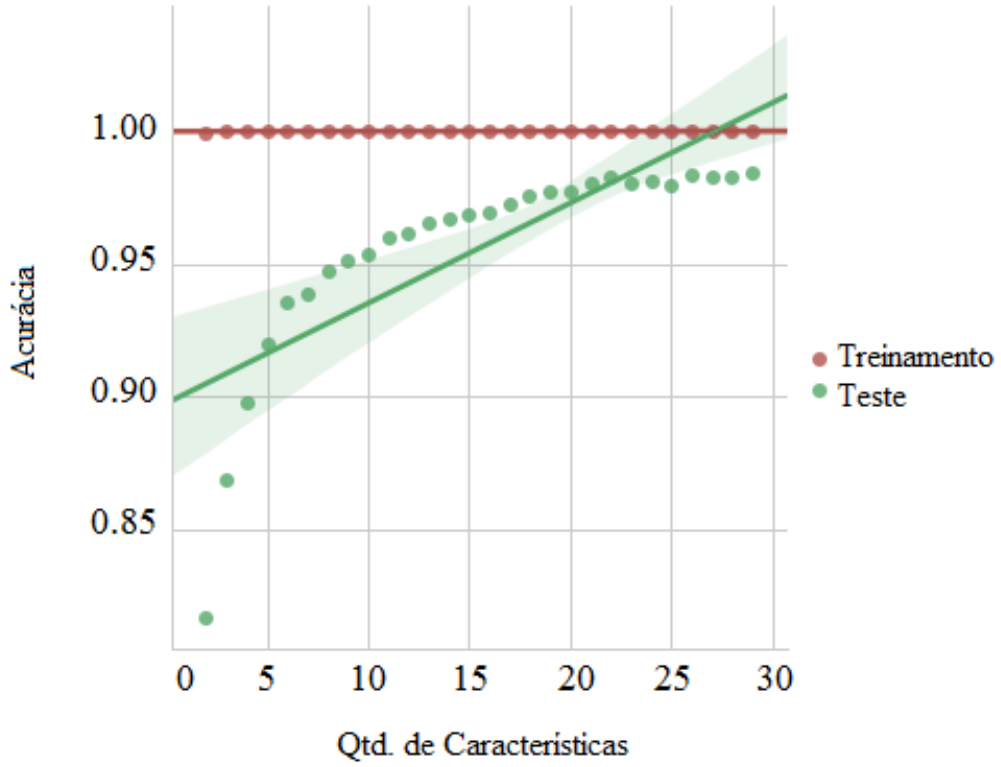

Figura 5.32: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 3-mers.

Das 29 características escolhidas por cada um dos dois métodos de seleção, 18 estão entre as 29 características de 3-mers utilizadas nas composições das assinaturas propostas por Ventola et al. [120]. Um fato interessante é que ambos os métodos encontraram 18 características mencionadas no artigo, mas essas características não necessariamente coincidem entre os dois métodos. Algumas características consideradas relevantes pelo artigo foram selecionadas pelo algoritmo de referência e outras pelo S2FS.

As Tabelas 5.14 e 5.15 apresentam as características mais importantes escolhidas pelos algoritmos de referência e pelo S2FS. As características encontram-se ordenadas pela ordem de importância de cada característica para o modelo classificador.

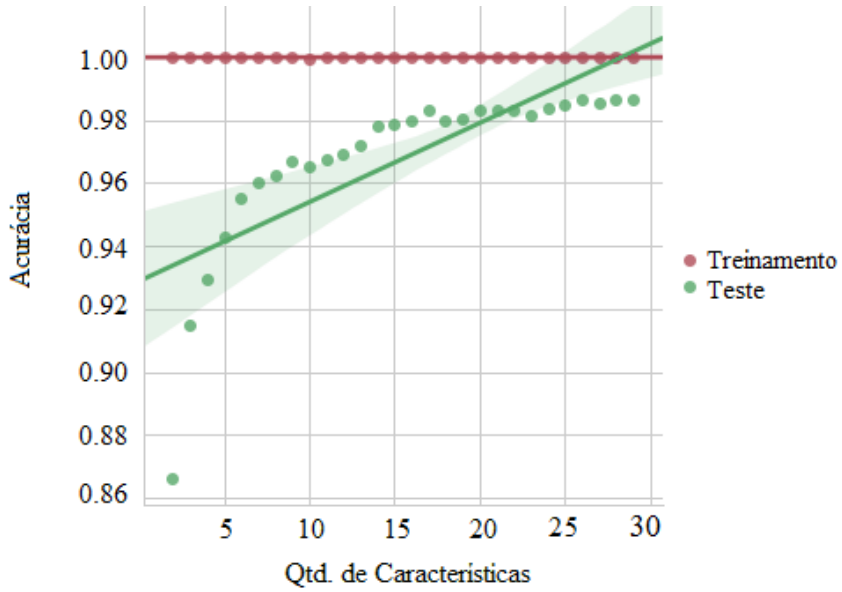

Figura 5.33: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 3-mers.

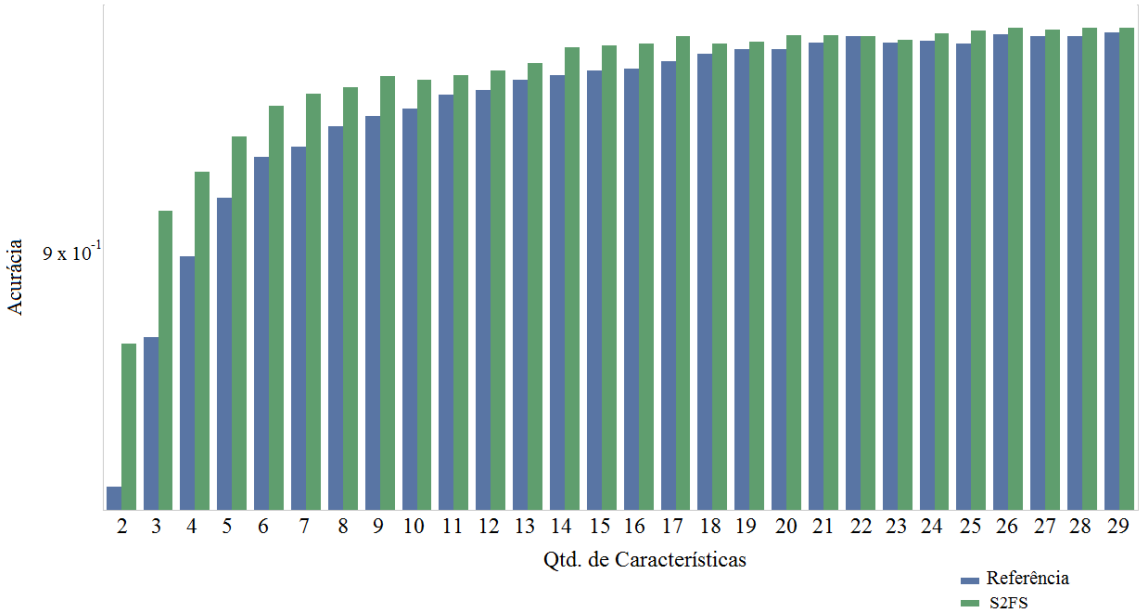

Figura 5.34: Comparação de desempenho dos dois métodos de seleção de características.

Tabela 5.14: Características 3-mer mais importantes escolhidas pelo algoritmo de referência para o *dataset D. rerio*.

Características
TCG, GGC, CCG, GGG, CCC, CGG, GCC, ACA ACC, CGA, TAT, ACG, CCT, TCC, CGT, CCA CTC, CGC, GAG, GCG, TTA, TGT, ATA, TTG TAA, AAT, ATT, TTT, AAA

Tabela 5.15: Características 3-mer mais importantes escolhidas pelo S2FS para o *dataset D. rerio*.

Características
CCG, CCC, TCG, GGC, CTA, GGG, CGA, GCC CGG, TCC, GGT, CGT, ACC, CGC, ACG, TGG CCT, CTC, GTC, CCA, GTG, GCT, ATC, AGG GGA, GCG, GAG, AGC, GAC

5.4.3 Sequências caracterizadas por 4-mers

Assim como nos outros *datasets* testados, os resultados obtidos com as características de frequências de 4-mers e com as sequências representadas pelos três tipos de características de frequência: 2-mers, 3-mers e 4-mers foram os mais significativos. A diferença entre os dois métodos de seleção fica bastante evidente nas Figuras 5.33, 5.34 e 5.35.

O menor número de sequências disponíveis para treinamento e teste dos modelos de classificação aparentemente não afetou o desempenho do S2FS, devido ao número de características disponíveis para escolha.

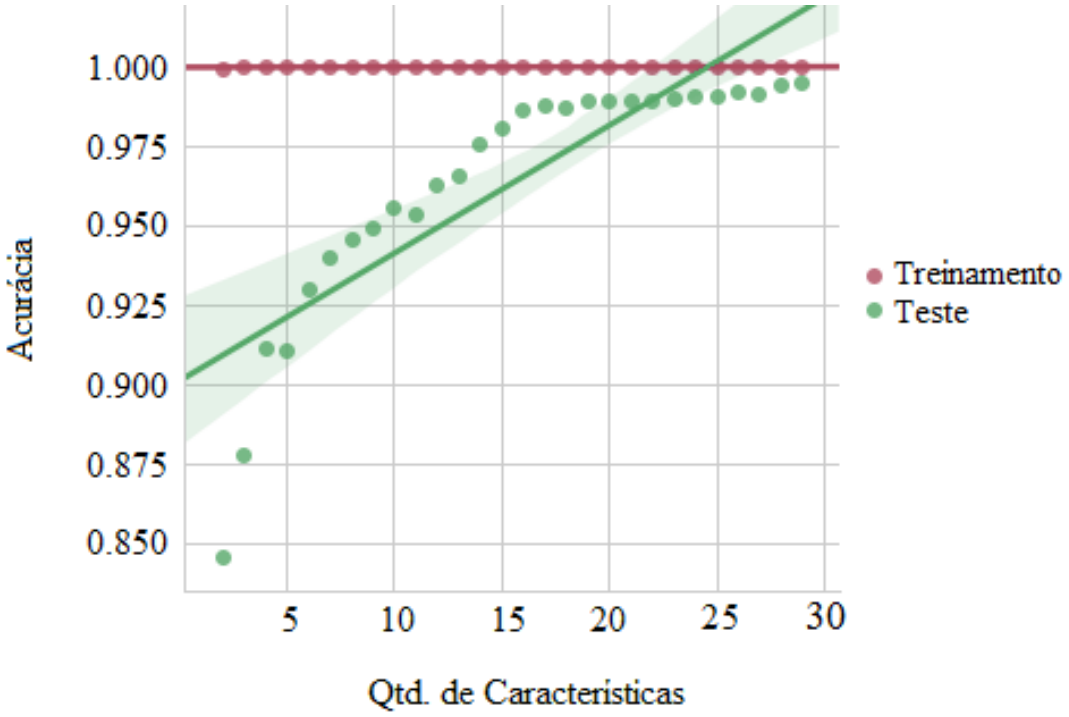

Figura 5.35: Desempenho dos modelos de classificação treinados com as características escolhidas pelo algoritmo de referência e testados em um conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 4-mers.

Outra observação interessante que pode ser feita na Figura 5.37 é que o S2FS alcança o limite do número de características que deve ser acrescentado ao modelo de classificação para que haja ganho de desempenho muito mais rápido do que o algoritmo de referência. Enquanto que com o S2FS os modelos de classificação não aparentam obter ganhos visíveis a partir de 7 características, com o algoritmo de referência os ganhos de desempenho passam a ser negligenciáveis a partir de 16 características. Note-se que não estamos propondo que 07 ou 16 características de frequências de 4-mers são suficientes para distinguir os PCTs dos lncRNAs do dataset *D. rerio*. Estamos comparando o desempenho de modelos de classificação de transcritos com uma pequena amostra do *dataset* disponível.

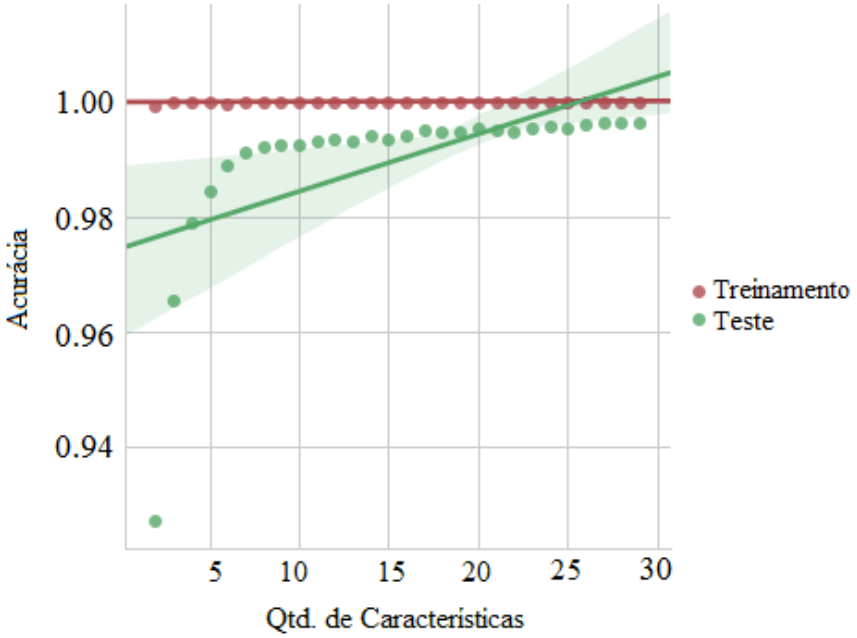

Figura 5.36: Desempenho dos modelos de classificação treinados com as características escolhidas pelo S2FS e testados em conjunto de dados de teste com as mesmas características. *Dataset D. rerio* com características 4-mers.

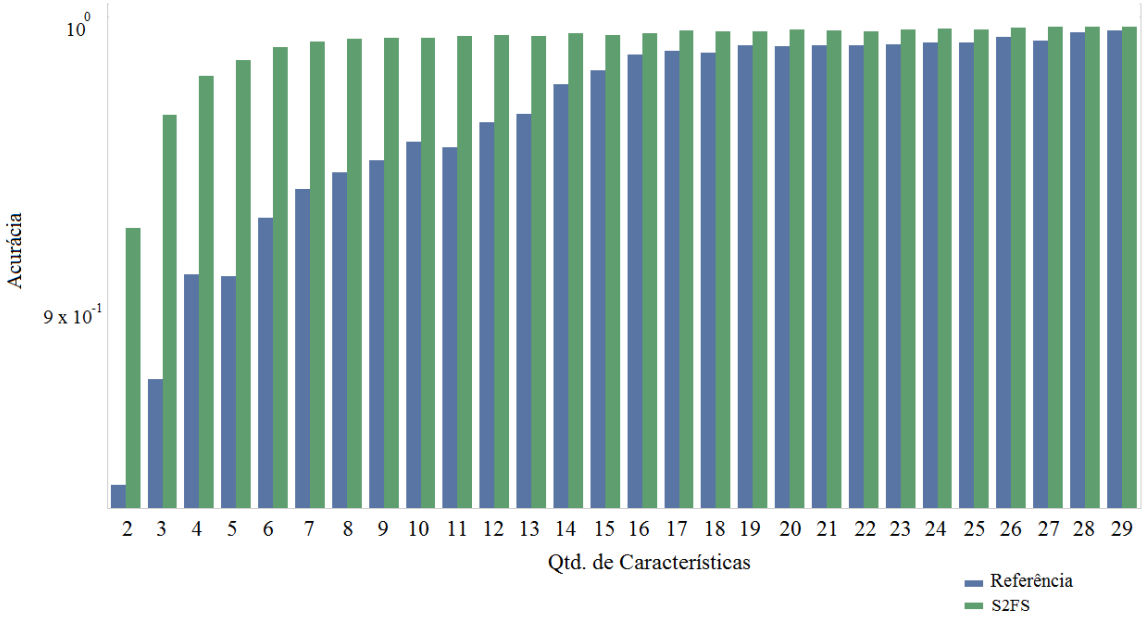

Figura 5.37: Comparação de desempenho dos dois métodos de seleção de características.

5.4.4 Sequências caracterizadas por 2-mers, 3-mers e 4-mers

O último teste realizado com o método de seleção proposto nesta dissertação foi no conjunto de características 2-mers, 3-mers e 4-mers (Figuras 5.38 e 5.39). De todos os *datasets* testados com a combinação de todas as características de frequências, o *dataset* de *Danio rerio* foi o conjunto de dados o qual o S2FS obteve o maior ganho de desempenho em relação ao algoritmo de referência, conforme pode ser visto na Figura 5.40.

Da mesma forma que com os testes aplicados as sequências representadas por 4-mer, o S2FS alcança o limite para o aumento do número de características com a finalidade de aumentar o desempenho dos modelos de classificação de forma consideravelmente mais rápida do que o algoritmo de referência.

Esses resultados são interessantes, pois o S2FS poderia ser utilizado como ferramenta para auxiliar no processo de encontrar o número mínimo de características para treinar modelos de classificação de forma a reduzir a possibilidade de treinar um modelo com *overfitting*. As Tabelas 5.16 e 5.17 apresentam as características mais importantes selecionadas pelo algoritmo de referência e pelo S2FS.

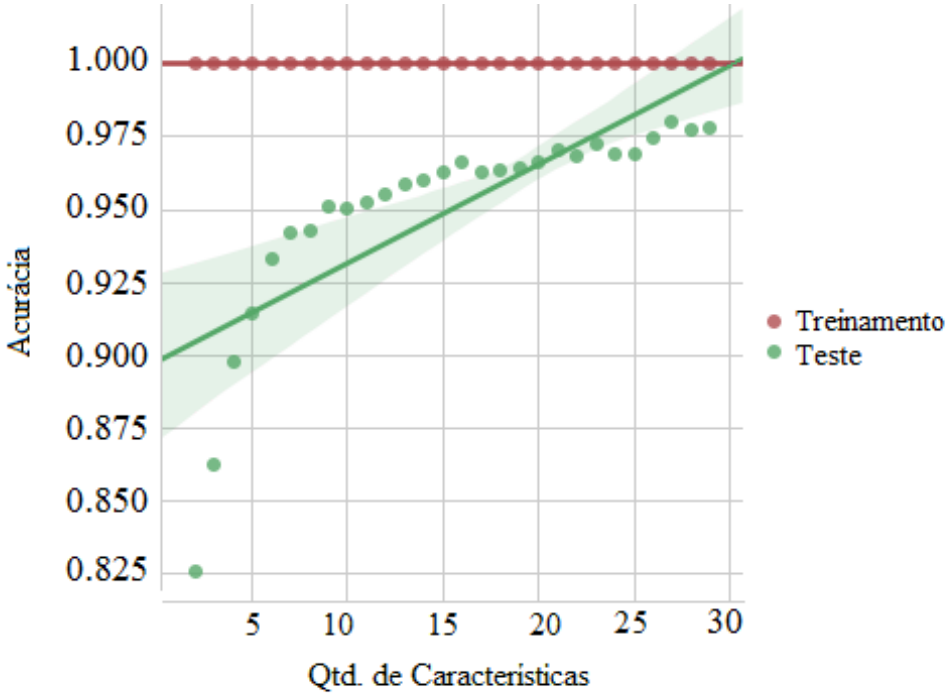

Figura 5.38: Resultado da classificação com o conjunto de dados de teste com as características escolhidas pelo algoritmo de referência. *Dataset D. rerio* com características 2-mers, 3-mers e 4-mers.

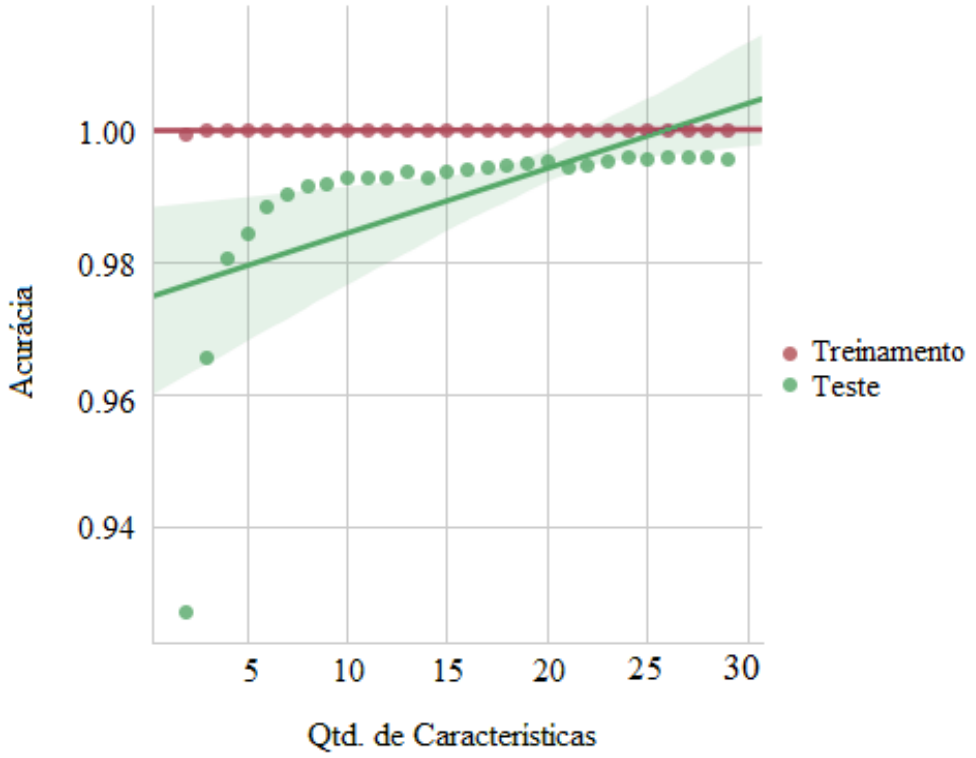

Figura 5.39: Resultado da classificação com o conjunto de dados de teste com as características escolhidas pelo S2FS. *Dataset D. rerio* com características 2-mers, 3-mers e 4-mers.

Tabela 5.16: Características 2, 3, 4-mers mais importantes escolhidas pelo algoritmo de referência.

Características
TCG, GGC, CCG, CCC, CC, CGG, GCC, CGA, CGC, GG, AATT, CG, AAAA, TTAA, GC, TTA, TAAA, TAA, AATA, TA, ATTT, AT, AA, TTT, ATA, AAT, AAA, TT, ATT

Tabela 5.17: Características 2, 3, 4-mers mais importantes escolhidas pelo S2FS.

Características
CCCG, TACG, CGGG, GTCG, TCGG, CGGC, GGCG, GGGG, CCGC, CCGG, TTCG, GCGC, CGCC, CCGA, TCCG, TCGC, CCCC, GCGA, GCCG, GCGG, CGAC, TCGA, CCGT, ATCG, ACCG, CGGA, ACGG, CGTC, CGAG

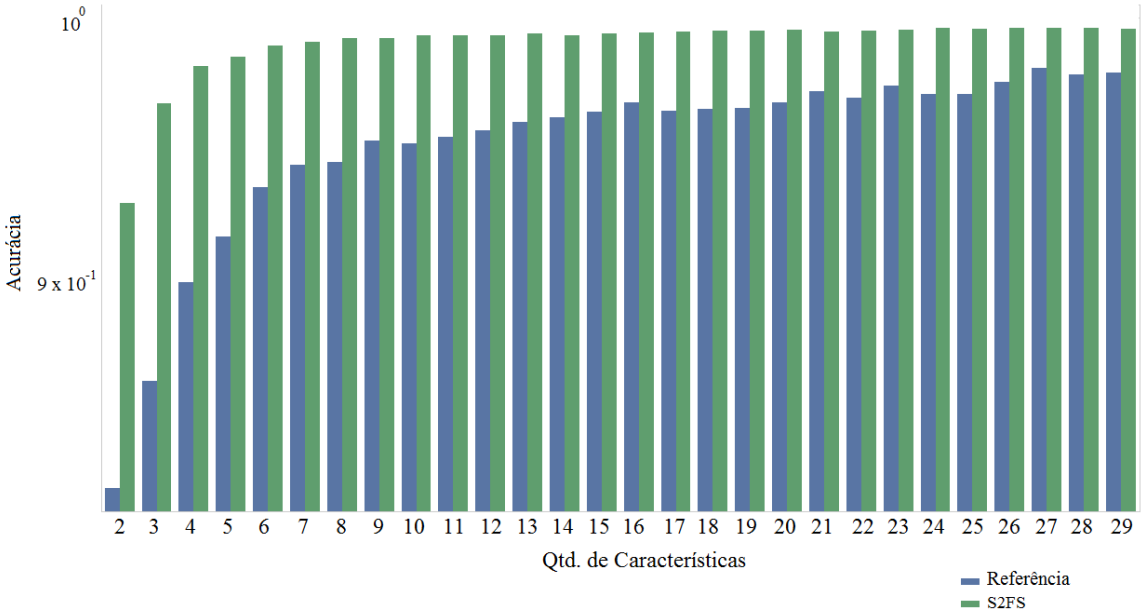

Figura 5.40: Comparação de desempenho dos dois métodos de seleção de características para o *dataset* de *D. rerio* com os transcritos caracterizados por 2-mer, 3-mer e 4-mer.

5.5 S2FS comparado com outro método: SFS

Os algoritmos SFS são uma família de algoritmos de busca gulosa utilizados para seleção de características [3]. É um método de seleção do tipo *wrapper*, com desempenho bastante superior ao algoritmo de referência utilizado neste trabalho para comparar com o desempenho do S2FS. O funcionamento do SFS *forward*, um dos tipos de algoritmo SFS, é selecionar uma característica por vez e testá-la em um classificador escolhido para avaliar a característica.

Na escolha da primeira característica, o S2FS é semelhante ao SFS, pois testa todas as características com um classificador pré-determinado. No entanto, o S2FS só passa uma vez por todas as características e utiliza a pontuação de cada uma como critério de escolha.

O SFS, por outro lado, após escolher a primeira melhor característica do conjunto total, testa novamente todas as características remanescentes para escolher a segunda melhor característica baseado no desempenho de duas características no classificador pré-determinado. E repete este processo até atingir o número desejado de características. Ou seja, o SFS *forward* e outros algoritmos da mesma família de busca sequencial executam em $O(d^2)$, onde d é o número de características. O S2FS, por outro lado, ao utilizar apenas a estratégia gulosa, executa o primeiro estágio do método em $O(d)$ e, em seguida, ordena as características em $O(d\log(d))$ e escolhe cada características em $O(1)$. Logo, o S2FS executa o seu processo em $O(d\log(d))$

Dessa forma, sabendo que o S2FS é mais rápido que o SFS, foi feito um teste com o *dataset* de *H. sapiens* utilizando os dois métodos de seleção, de forma a verificar a diferença de desempenho entre os dois métodos de seleção.

Na máquina utilizada para os testes, o SFS utilizou 17 horas para escolher 30 características das 336 disponíveis, enquanto que o S2FS utilizou 30 minutos para escolher 30 características. O desempenho do classificador RF treinado com as características escolhidas pelo SFS foi melhor que o desempenho do mesmo classificador treinado com as características escolhidas pelo S2FS utilizando a estratégia gulosa sem filtro de características correlacionadas, ou seja, a versão mais rápida do S2FS. Sem a filtragem de características correlacioandas, o S2FS só consegue alcançar o desempenho do SFS ao chegar perto do limite de 30 características, conforme pode ser visto na Figura 5.41.

Entretanto, ao se aplicar o filtro de características correlacionadas na escolha gulosa do S2FS, só selecionando as características com melhor pontuação que não são correlacionadas entre si, é possível alcançar o desempenho do SFS com muito menos características ainda utilizando uma fração do tempo utilizada pelo SFS, conforme mostra a Figura 5.42.

Ambos os métodos de seleção, S2FS e SFS, escolhem predominantemente 4-mer como características mais relevantes, conforme pode ser visto nas Tabelas 5.18 e 5.19.

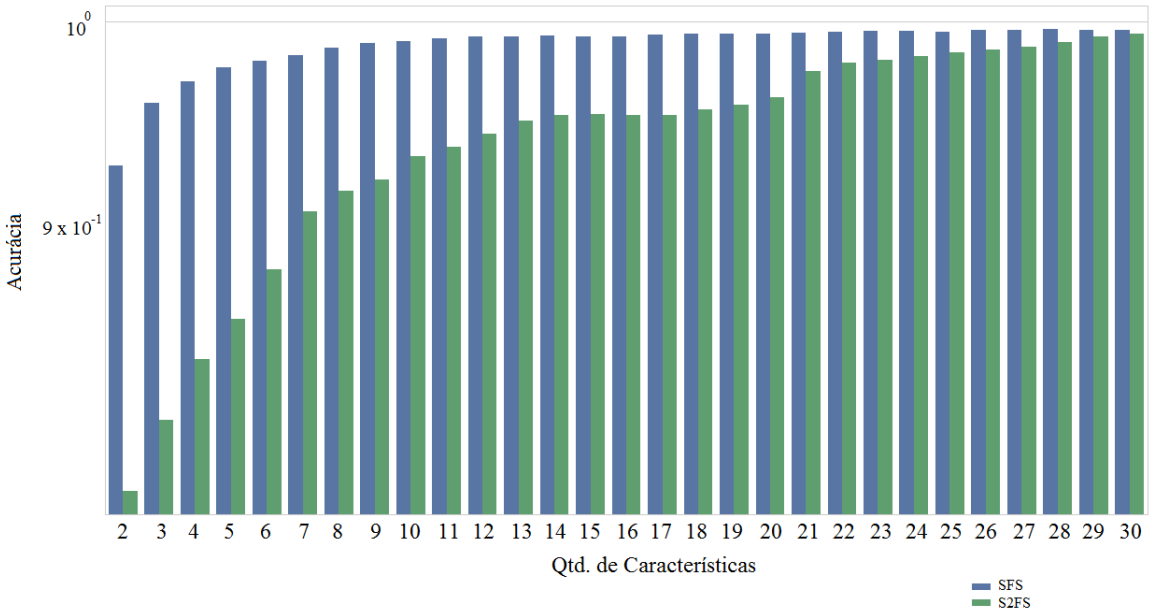

Figura 5.41: Comparação de desempenho entre o S2FS e o SFS para o *dataset* de *H. sapiens* com os transcritos caracterizados por 2-mer, 3-mer e 4-mer.

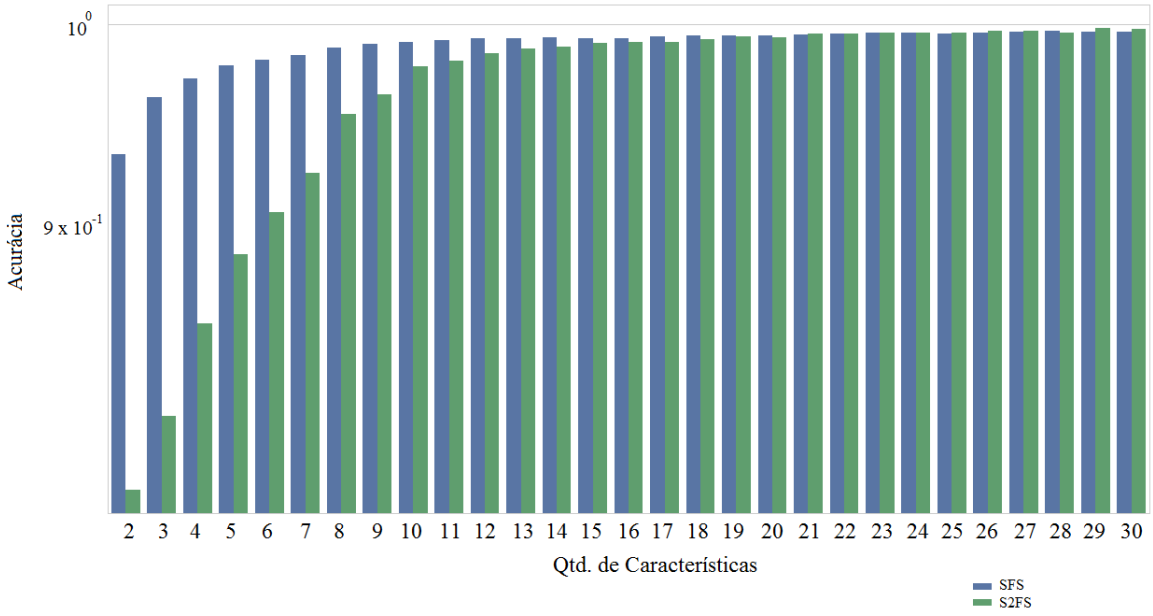

Figura 5.42: Comparação de desempenho entre o S2FS com filtro de correlação com SFS para o *dataset* de *H. sapiens* com os transcritos caracterizados por 2-mer, 3-mer e 4-mer.

Tabela 5.18: Características 2, 3, 4-mers mais importantes escolhidas pelo S2FS e rankeadas pelo RF.

Características
TACG, ACGG, CGAT, CCGT, CGTG, GTAT, GTAC, TAAC, TAGT, TATC, TACC, ATAC, TTAC, CTTA, TTAG, CAAT, TACT, TATG, ATTA, ATAT, ATTG, AAAA, CG, TTTC, CTT, TTTA, TTTG, GTT, ATC, TGT

Tabela 5.19: Características 2, 3, 4-mers mais importantes escolhidas pelo algoritmo SFS e rankeadas pelo RF.

Características
TTCG, ATCG, TACG, TCGA, CGAT, TCGT, AACG, ACGA, GTCG, GTAC, CGTT, CGAA, GGTA, CGGT, TATC, CGTA, ACGT, CGAC, TACC, CTCG, CCGA, CCGT, TCGC, TAGC, GCGA, CGAG, GCGT, CGCG, CGTG, CGTC

5.6 Discussão

O que pôde ser observado dos resultados apresentados é que o S2FS possui um desempenho igual ou melhor que o algoritmo de referência, se forem atendidos dois critérios principais: o número de características disponíveis e os hiperparâmetros de classificação.

Em relação ao número de características disponíveis, a quantidade de características deve ser suficientemente grande. Nos casos testados neste trabalho, por exemplo, quando as sequências estavam caracterizadas apenas por 2-mers, o número pequeno de características não foi suficiente nem para se obter bons resultados de classificação nem para distinguir claramente a diferença entre os métodos de seleção de características. Isso era esperado, considerando-se que muitas ferramentas existentes para predição de transcritos ncRNAs utilizam outras características além de 2-mers.

Por outro lado, ao aplicar os métodos de seleção nos conjuntos de características de frequências 3-mers, 4-mers e no conjunto com todos eles, 2-mers, 3-mers e 4-mers, o S2FS selecionou características que resultaram em um melhor desempenho dos modelos de classificação em todos os *datasets* de teste. O S2FS também alcançou o limite para se acrescentar novas características de modo a obter ganhos de desempenho com um número bem menor de características do que o algoritmo de referência. Este é um resultado interessante do ponto de vista da construção de modelos de classificação com o menor número possível de características.

Já em relação à escolha dos hiperparâmetros no primeiro estágio do método, é importante testar diferentes parâmetros para encontrar o valor adequado. Por exemplo, caso o hiperparâmetro de regularização esteja com o valor muito alto, poucas características vão estar disponíveis para as fases seguintes de seleção do *pipeline*. Por outro lado, se o valor estiver muito baixo, nenhuma característica vai ser descartada na primeira fase, o que pode aumentar o tempo de execução do método. Um valor adequado seria, então, aquele que conseguisse descartar características, mas que preservasse as característica que obtiveram melhor desempenho no primeiro estágio.

O foco deste trabalho está nos métodos de Aprendizado de Máquina e sua aplicação no contexto de Bioinformática. Por isso, é natural que algumas informações de interesse dessa área de pesquisa surjam nos resultados do trabalho.

A primeira observação que podemos constatar é que o conjunto de características k-mers mais relevante para classificação de lncRNAs é fortemente relacionado à espécie. Cada *dataset* testado gerou um conjunto diferente de características relevantes. Logo, a proposta de construção de um modelo de classificação amplo para diversas espécies deve levar em consideração essa informação. Extrair um conjunto comum de características relativos à informação primária das sequências de diferentes espécies para construir um classificador geral poderia ter um desempenho inferior.

Outra observação interessante dos resultados obtidos é que, de acordo com os testes realizados neste trabalho, quanto maior o tamanho do k-mers das características, melhor o desempenho dos classificadores utilizados para testar os métodos de seleção de características. Entretanto, deve-se levar em consideração os recursos computacionais disponíveis. Ao aumentarmos o tamanho dos k-mers, aumentamos exponencialmente a quantidade de memória necessária para armazenar as sequências em memória durante o processamento, e, como consequência, o tempo de processamento.

O último ponto a ser observado é que o S2FS é bastante eficiente e rápido quando comparado a outros bons métodos de seleção de características, como o SFS. Ao se utilizar um filtro de correlação entre as características que são escolhidas pela estratégia gulosa, é possível alcançar ótimos desempenhos em uma fração do tempo.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho, propusemos o S2FS, um método de seleção de características relacionadas às frequências dos k-mers das sequências de transcritos gerados por RNA-Seq. Os resultados de classificação obtidos com as características selecionadas pelo S2FS foram então comparados com resultados de classificação feitos com características escolhidas por um algoritmo de seleção univariada, de forma a verificar a qualidade das características escolhidas pelo S2FS.

A partir dos resultados observados neste trabalho, constatamos que o método S2FS, que utiliza o desempenho de cada característica como critério para seleção das melhores características, obteve resultados satisfatórios. Obtivemos bons desempenhos dos modelos de classificação nos subconjuntos de dados de teste dos *datasets* de *H. sapiens*, *M. musculus* e *D. rerio*. Os resultados confirmam que a etapa de seleção de características é bastante importante para a construção de bons modelos de distinção entre ncRNAs e transcritos codificadores de proteínas.

Em teste realizado por Vieira et al [121] com um *dataset* de lincRNAs de humanos semelhante ao usado neste trabalho, o S2FS selecionou um subconjunto de características 4-mer consideravelmente menor do que o conjunto de características de frequência utilizadas até então (as mesmas características de frequência utilizadas pelo iSeeRNA [108]). O resultado de desempenho do classificador utilizado no trabalho de Vieira et al [121] com as características escolhidas pelo S2FS foi melhor do que com as utilizadas anteriormente. Esse é um resultado coerente com o que apresentamos na seção 3.4 sobre a relação entre o número de sequências, número de características e *overfitting*.

Os resultados aqui apresentados mostram que o S2FS é eficiente para a escolha de características de frequências 3-mer e 4-mer para construção de modelos de predição de transcritos. Para esses conjuntos de características, o método proposto teve um desempenho superior ao obtido pelo método de seleção univariada de características nos três *datasets* testados. Por outro lado, para as características de frequência 2-mer, os resulta-

dos não indicam com clareza se o S2FS é melhor que o algoritmo de referência. Nos três *datasets* testados com características 2-mer, utilizando diferentes quantidades de características, o S2FS obteve resultados variáveis de desempenho em relação ao algoritmo de referência.

Ainda foi possível constatar que, nos três *datasets* testados, diferentes características de frequências de k-mers possuem diferentes níveis de importância para a distinção entre ncRNAs e PCTs em cada *dataset*. Do ponto de vista da Biologia Molecular, esse resultado precisa ser investigado. Do ponto de vista computacional, é uma indicação interessante de que provavelmente um conjunto heterogêneo de classificadores especializados que dependem de cada *dataset*, como *Ensemble*, teria resultados mais promissores do que construir um único classificador com as características comuns entre os *datasets*.

Outro ponto interessante a respeito dos resultados obtidos pelo S2FS é que, quando o método proposto escolhe características de um conjunto maior e mais heterogêneo de características de frequências: 2-mer, 3-mer e 4-mer, as características 4-mer são predominantemente escolhidas, as quais possibilitam o melhor desempenho dos modelos de classificação. Isso levanta a hipótese de que k-mers maiores, 5-mer ou 6-mer, poderiam gerar desempenhos ainda melhores do que os alcançados pelas características de frequência 4-mer. Por outro lado, k-mers maiores exigem recursos computacionais consideravelmente maiores, uma vez que o aumento dos k-mers é exponencial em relação à quantidade de memória para armazenar as sequências. Como consequência, o tempo de processamento para treinar e testar os modelos de classificação é significativamente maior.

Entretanto, cabe uma ressalva em relação aos resultados apresentados neste trabalho. Todos os testes foram realizados com conjuntos de características relacionados às frequências dos k-mers. Essas características possuem uma mesma faixa de valores possíveis, ou seja, apesar de distintas, são todas características do mesmo tipo e com as mesmas dimensões. Novos testes devem ser realizados com outros conjuntos de características e outras faixas de valores, de forma a estudar o comportamento do método em diferentes situações. O único teste em que puderam ser observadas características de diferentes dimensões foi quando se utilizou o tamanho da ORF e as características de frequências.

Os testes com os modelos de classificação com o tamanho da ORF mostraram uma predominância da importância do tamanho da ORF. Esse é um resultado já observado em outros trabalhos, como em Schneider et [101] e Vieira et al [121]. No caso deste trabalho, a importância do tamanho da ORF também é explicada devido ao processo inicial de seleção de sequências. Os transcritos escolhidos das classes distintas possuíam valores muito diferentes de tamanho de ORF, conforme vimos na subseção 5.1.4. De toda forma, a presença da característica ORF alterou o padrão de ganho de desempenho quando comparado aos outros testes. Talvez seja interessante aplicar alguma operação de

normalização nas características antes de se aplicar o S2FS como método de seleção de características.

Assim, uma proposta de trabalho futuro seria testar o S2FS com características heterogêneas, de forma a verificar o desempenho na escolha de características de maneira mais abrangente. Além disso, seria interessante testar o método proposto em *datasets* de outras espécies, ou até mesmo em outras áreas de aplicação, para verificar se o método proposto é generalizável. Outra possibilidade de trabalho a ser desenvolvida é otimizar o método para que ele possa ser executado de maneira paralela, utilizando todas as unidade de processamento disponíveis. Como cada característica é avaliada separadamente, é possível que elas sejam avaliadas por diversas *threads* ou processos distintos, de forma simultânea, acelerando o processo inicial da seleção das características. Nas fases seguintes, onde há a clusterização, a escolha gulosa das melhores característica ainda deve ser feita de maneira sequencial. No entanto, a fase que consome mais tempo de processamento é a primeira. Logo, uma paralelização do código aumentaria consideravelmente a velocidade do S2FS.

Referências

[1] H. Abdi e L. Williams. Principal Component Analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 26

[2] R. Achawanantakun, J. Chen, Y. Sun, e Y. Zhang. Lncrna-id: Long non-coding rna identification using balanced random forests. *Bioinformatics*, 31(24):3897–3905, 2015. 38

[3] D. W. Aha e R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from data*, pages 199–206. Springer, 1996. 85

[4] B. L. Aken, P. Achuthan, W. Akanni, M. R. Amode, F. Bernsdorff, J. Bhai, K. Billis, D. Carvalho-Silva, C. Cummins, P. Clapham, et al. Ensembl 2017. *Nucleic Acids Research*, 45(D1):D635–D642, 2017. 12, 35

[5] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, e P. Walter. *Molecular Biology of the Cell, Sixth Edition:*. Taylor & Francis Group, 2014. 1, 6, 7, 8, 13

[6] P. P. Amaral, M. B. Clark, D. K. Gascoigne, M. E. Dinger, e J. S. Mattick. lncRNAdb: a reference database for long noncoding RNAs. *Nucleic Acids Research*, 39(suppl 1):D146–D151, 2011. 12

[7] R. T. Arrial, R. C. 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):1, 2009. 17

[8] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991. 26, 27

[9] S. Balakrishnama e A. Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and Information Processing*, 18, 1998. 32

[10] J. M. Berg, J. L. Tymoczko, e L. Stryer. *Biochemistry, Fifth Edition*. W.H. Freeman, 2002. 9

[11] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 24

[12] R. J. Britten. Rates of DNA sequence evolution differ between taxonomic groups. *Science*, 231(4744):1393–1398, 1986. 14

[13] D. Bu, K. Yu, S. Sun, C. Xie, G. Skogerbø, R. Miao, H. Xiao, Q. Liao, H. Luo, G. Zhao, et al. NONCODE v3. 0: integrative annotation of long noncoding RNAs. *Nucleic Acids Research*, page gkr1175, 2011. 12

[14] B. Canard e R. S. Sarfati. DNA polymerase fluorescent substrates with reversible 3-tags. *Gene*, 148(1):1–6, 1994. 2

[15] L.J Cao, K. S. Chua, W.K. Chong, H.P Lee, e Q. Gu. A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1):321–336, 2003. 43

[16] S. M. Carr. DNA nucleotides. `http://www.mun.ca/biology/scarr/MGA2-02-06.jpg`, 2005. Acessado em: 2017-11-16. x, 7

[17] O. Chapelle, B. Scholkopf, e A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 20

[18] G. Chen, Z. Wang, D. Wang, C. Qiu, M. Liu, X. Chen, Q. Zhang, G. Yan, e Q. Cui. LncRNADisease: a database for long-non-coding RNA-associated diseases. *Nucleic acids research*, 41(D1):D983–D986, 2013. 16

[19] Wikimedia Commons. RNA codon table. `https://en.wikipedia.org/wiki/Genetic_code#RNA_codon_table`, 2006. Acessado em: 2017-11-16. x, 10

[20] Wikimedia Commons. Strukturmodell eines ausschnitts aus der DNA-doppelhelix (b-form) mit 20 basenpaarungen. `https://upload.wikimedia.org/wikipedia/commons/f/f0/DNA_Overview.png`, 2006. Acessado em: 2017-11-16. x, 8

[21] Wikimedia Commons. Graph of variation in estimated genome sizes in base pairs. `https://upload.wikimedia.org/wikipedia/commons/thumb/8/80/Genome_Sizes.png/800px-Genome_Sizes.png`, 2010. Acessado em: 2017-11-16. x, 12

[22] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994. 26

[23] ENCODE Project Consortium et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012. 11

[24] G. M. Cooper e R. E. Hausman. *The Cell: A Molecular Approach*. Cell, a Molecular Approach. Sinauer Associates, 2013. 11

[25] C. Cortes e V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 16

[26] F. Crick. On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8, 1958. 1

[27] M. Dash e H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997. 31

[28] Y. Devaux, J. Zangrando, B. Schroen, E. E. Creemers, T. Pedrazzini, C. P. Chang, I. Dorn, W. Gerald, T. Thum, S. Heymans, et al. Long noncoding RNAs in cardiac development and ageing. *Nature Reviews Cardiology*, 12(7):415–425, 2015. x, 15

[29] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. x, 22, 23, 24

[30] M. Esteller. Non-coding RNAs in human disease. *Nature Reviews Genetics*, 12(12):861–874, 2011. 16

[31] M. Ester, H. P. Kriegel, J. Sander, e X. Xu. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996. 26

[32] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, et al. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008. 39

[33] X. N. Fan e S. W. Zhang. lncRNA-MFDL: identification of human long non-coding RNAs by fusing multiple features and using deep learning. *Molecular BioSystems*, 11(3):892–897, 2015. 18, 37

[34] M. Fasold, D. Langenberger, H. Binder, P. F. Stadler, e S. Hoffmann. Dario: a ncrna detection and analysis tool for next-generation sequencing experiments. *Nucleic acids research*, 39(suppl_2):W112–W117, 2011. 38

[35] B. I. Fedeles, B. D. Freudenthal, E. Yau, V. Singh, S. C. Chang, et al. Intrinsic mutagenic properties of 5-chlorocytosine: A mechanistic connection between chronic inflammation and cancer. *Proceedings of the National Academy of Sciences*, 112(33):E4571–E4580, 2015. 14

[36] R. A. Fisher. *Statistical Methods For Research Workers*. Cosmo study guides. Cosmo Publications, 1925. 32

[37] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936. 32

[38] R. Flores, C. Hernández, A. E. M. Alba, J.A. Daròs, e F. D. Serio. Viroids and viroid-host interactions. *Annu. Rev. Phytopathol.*, 43:117–139, 2005. 11

[39] Y. Freund, R. Schapire, e N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999. 25

[40] J. Friedman, T. Hastie, e R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001. 25

[41] J. G. Gall. Chromosome structure and the C-value paradox. *The Journal of Cell Biology*, 91(3):3s–14s, 1981. 12

[42] E. A. Gibb, C. J. Brown, e W. L. Lam. The functional role of long non-coding RNA in human carcinomas. *Molecular Cancer*, 10(1):1, 2011. 16

[43] W. Gish et al. Identification of protein coding regions by database similarity search. *Nature genetics*, 3(3):266–272, 1993. 17, 19

[44] P. J. Grabowski, S. R. Seiler, e P. A. Sharp. A multicomponent complex is involved in the splicing of messenger RNA precursors. *Cell*, 42(1):345–353, 1985. 11

[45] C. W. Groetsch. The theory of tikhonov regularization for fredholm equations. *104p, Boston Pitman Publication*, 1984. 33

[46] R. A. Gupta, N. Shah, K. C. Wang, J. Kim, H. Horlings, et al. Long non-coding RNA HOTAIR reprograms chromatin state to promote cancer metastasis. *Nature*, 464(7291):1071–1076, 2010. 16

[47] I. Guyon e A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003. 32

[48] L. K. Hansen e P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990. 23

[49] Y. Hao, T. Crenshaw, T. Moulton, E. Newcomb, e B. Tycko. Tumour-suppressor activity of H19 RNA. *Nature*, 365(6448):764–767, 1993. 16

[50] K. Hibi, H. Nakamura, A. Hirai, Y. Fujikake, Y. Kasai, S. Akiyama, K. Ito, e H. Takagi. Loss of H19 imprinting in esophageal cancer. *Cancer research*, 56(3):480–482, 1996. 16

[51] T. K. Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995. 25

[52] R. W. Holley, J. Apgar, G. A. Everett, J. T. Madison, M. Marquisee, S. H. Merrill, J. R. Penswick, e A. Zamir. Structure of a ribonucleic acid. *Science*, 147(3664):1462–1465, 1965. 11

[53] G. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968. 30

[54] L. Hunter e J. Lederberg. Artificial intelligence and molecular biology. In *AI Magazine*. Citeseer, 1993. 3

[55] G. James, D. Witten, T. Hastie, e R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013. 21, 31

[56] C. A. Janeway, P. Travers, M. Walport, e M. J. Shlomchik. *Immunobiology: the immune system in health and disease*, volume 1. Current Biology, 1997. 14

[57] P. Johnsson, L. Lipovich, D. Grandér, e K. V. Morris. Evolutionary conservation of long non-coding RNAs; sequence, structure, function. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1840(3):1063–1071, 2014. 3, 14, 15

[58] L. Kaufman e P. Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987. 26

[59] E. Keogh e A. Mueen. *Curse of Dimensionality*, pages 257–258. Springer US, Boston, MA, 2010. 4

[60] D. Kim, W. K. Lee, S. Jeong, M. Y. Seol, H. Kim, K. S. Kim, E. J. Lee, J. Lee, e Y. S. Jo. Upregulation of long noncoding rna loc100507661 promotes tumor aggressiveness in thyroid cancer. *Molecular and Cellular Endocrinology*, 431:36–45, 2016. 3

[61] T. Kohonen e T. Honkela. Kohonen network. *Scholarpedia*, 2(1):1568, 2007. 33

[62] L. Kong, Y. Zhang, Z. Q. Ye, X. Q. O. Liu, S. Q. Zhao, L. Wei, e G. Gao. CPC: assess the protein-coding potential of transcripts using sequence features and Support Vector Machine. *Nucleic Acids Research*, 35(suppl 2):W345–W349, 2007. 16

[63] J. T. Y. Kung, D. Colognori, e J. T. Lee. Long noncoding RNAs: past, present, and future. *Genetics*, 193(3):651–669, 2013. 1, 14

[64] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. 2

[65] S. Lertampaiporn, C. Thammarongtham, C. Nukoolkit, B. Kaewkamnerdpong, e M. Ruengjitchatchawalya. Identification of non-coding rnas with a new composite feature in the hybrid random forest ensemble algorithm. *Nucleic acids research*, 42(11):e93–e93, 2014. 38

[66] A. Li, J. Zhang, e Z. Zhou. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. *BMC bioinformatics*, 15(1):311, 2014. 17, 37

[67] M. F. Lin, I. Jungreis, e M. Kellis. Phylocsf: a comparative genomics method to distinguish protein coding and non-coding regions. *Bioinformatics*, 27(13):i275–i282, 2011. 18

[68] D. C. Liu e J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. 39

[69] F. T. Liu, K. M. Ting, e Z. H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008. 26

[70] S. P. Liu, J. X. Yang, D. Y. Cao, e K. Shen. Identification of differentially expressed long non-coding rnas in human ovarian cancer cells with different metastatic potentials. *Cancer biology & medicine*, 10(3):138–141, 2013. 3

[71] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 26

[72] K. J. Locey e J. T. Lennon. Scaling laws predict global microbial diversity. *Proceedings of the National Academy of Sciences*, page 201521291, 2016. 3

[73] L. Ma, V. B. Bajic, e Z. Zhang. On the classification of long non-coding RNAs. *RNA Biology*, 10(6):924–933, 2013. 14

[74] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. 26

[75] C. A. Maher, C. Kumar-Sinha, X. Cao, S. Kalyana-Sundaram, B. Han, X. Jing, L. Sam, T. Barrette, N. Palanisamy, e A. M. Chinnaiyan. Transcriptome sequencing to detect gene fusions in cancer. *Nature*, 458(7234):97–101, 2009. 3

[76] O. Maimon e L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Series in Solid-State Sciences. Springer US, 2010. 28

[77] G. McLachlan e K. Basford. *Mixture models. Inference and applications to clustering*. 1988. 26

[78] L. McLaughlin. Processed mRNA leaves the nucleus. `http://slideplayer.com/slide/8115500/`, 2015. Acessado em: 2017-11-16. x, 11

[79] T. M. Mitchell. *The discipline of machine learning*, volume 3. Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006. 20

[80] B. B. Mulvey, U. Olcese, J. R. Cabrera, e J. I. Horabin. An interactive network of long non-coding RNAs facilitates the drosophila sex determination decision. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1839(9):773–784, 2014. 16

[81] P. Norvig e S. J. Russell. Inteligência artificial. *Editora Campus*, 2004. 20

[82] B. Panwar, A. Arora, e G. P. S. Raghava. Prediction and classification of ncRNAs using structural information. *BMC Genomics*, 15(1):127, 2014. 62

[83] J.-E Park, I. Heo, Y. Tian, D.K. Simanshu, H. Chang, D. Jee, D. J. Patel, e V. N. Kim. Dicer recognizes the 5 [prime] end of RNA for efficient and accurate processing. *Nature*, 475(7355):201–205, 2011. 13

[84] K. Pearson. Mathematical contributions to the theory of evolution.–on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the Royal Society of London*, 60(359-367):489–498, 1896. 32

[85] K. Pearson. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900. 32

[86] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 33

[87] K. Pearson. Determination of the coefficient of correlation. *Science*, 30(757):23–25, 1909. 44

[88] J. A. Poland e T. W. Rife. Genotyping-by-sequencing for plant breeding and genetics. *The Plant Genome*, 5(3):92–102, 2012. x, 2

[89] C. Ponting e W. Reik. Evolution and functions of long noncoding RNAs. *Cell*, Volume 136(4):629–641, feb 2009. doi:10.1016/j.cell.2009.02.006. 15

[90] J. R. Prensner e A. M. Chinnaiyan. The emergence of lncRNAs in cancer biology. *Cancer Discovery*, 1(5):391–407, 2011. 3, 14

[91] K. D. Pruitt, T. Tatusova, e D. R. Maglott. NCBI reference sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research*, 33(suppl 1):D501–D504, 2005. 12

[92] K. D. Pruitt, T. Tatusova, e D. R. Maglott. Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic acids research*, 35(suppl 1):D61–D65, 2007. 35

[93] C. J. Rawlings, J. P. Fox, E. A. Thompson, e B. Robson. Artificial Intelligence in Molecular Biology: A Review and Assessment [and Discussion]. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 344(1310):353–363, 1994. 3

[94] Y. P. Raykov, A. Boukouvalas, F. Baig, e M. A. Little. What to do when k-means clustering fails: a simple yet principled alternative algorithm. *PloS one*, 11(9):e0162259, 2016. 28

[95] D. Reddy e P. Jana. Initialization for k-means clustering using voronoi diagram. *Procedia Technology*, 4:395–400, 2012. 26

[96] J. L. Rinn e H. Y. Chang. Genome regulation by long noncoding RNAs. *Annual review of biochemistry*, 81, 2012. 11

[97] J. L. Rinn, M. Kertesz, J. K. Wang, S. L. Squazzo, X. Xu, S. A. Brugmann, L. H. Goodnough, et al. Functional demarcation of active and silent chromatin domains in human hox loci by noncoding RNAs. *cell*, 129(7):1311–1323, 2007. 16

[98] L. Rokach e O. Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005. 28

[99] M. Ronaghi, M. Uhlén, e P. Nyren. A sequencing method based on real-time pyrophosphate. *Science*, 281(5375):363, 1998. 2

[100] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 38

[101] H. W. Schneider, T. Raiol, M. M. Brigido, M. E. M. T. Walter, e P. F. Stadler. A Support Vector Machine based method to distinguish long non-coding RNAs from protein coding transcripts. *BMC Genomics*, 18(1):804, Oct 2017. 19, 91

[102] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A. J. Smola, e R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. 26

[103] J. C. Setubal. Como genes codificam proteínas. University Lecture, 2016. x, 9

[104] J. C. Setubal e J. Meidanis. *Introduction to computational molecular biology*. Computer Science Series. PWS Pub., 1997. 1

[105] S. A. Shabalina e N. A. Spiridonov. The mammalian transcriptome and the function of non-coding DNA sequences. *Genome Biology*, 5(4):1, 2004. 1, 12

[106] T. Spector. *Identically different: why you can change your genes*. Hachette UK, 2012. 1

[107] V. Spruyt. About the curse of dimensionality. `https://www.datasciencecentral.com/profiles/blogs/about-the-curse-of-dimensionality`. Acessado: 2017-11-16. xi, 31

[108] K. Sun, X. Chen, P. Jiang, X. Song, H. Wang, e H. Sun. iSeeRNA: identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data. *BMC Genomics*, 14(2):S7, 2013. 17, 62, 90

[109] L. Sun e L.and Meng J. Liu, H.and Zhang. lncrscan-svm: a tool for predicting long non-coding rnas using support vector machine. *PloS one*, 10(10):e0139654, 2015. 18

[110] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, e Y. Zhao. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic Acids Research*, page gkt646, 2013. 17

[111] J. M. Sutter e J. H. Kalivas. Comparison of forward selection, backward elimination, and generalized simulated annealing for variable selection. *Microchemical journal*, 47(1-2):60–66, 1993. 18

[112] Richard S Sutton e Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. 21

[113] A. L. Tarca, V. J. Carey, X. Chen, R. Romero, e S. Drăghici. Machine learning and its applications to biology. *PLoS Comput Biol*, 3(6):e116, 2007. 3

[114] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 33

[115] R. Tibshirani, G. Walther, e T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. 27

[116] R. Tripathi, S. Patel, V. Kumari, P. Chakraborty, e P. K. Varadwaj. Deeplnc, a long non-coding RNA prediction tool using deep neural network. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):1–14, 2016. 18, 36, 38

[117] A. B. Tucker. *Computer Science Handbook, Second Edition*. CRC Press, 2004. 26

[118] H. Varmus. Retroviruses. *Science*, 240(4858):1427–1435, 1988. 7

[119] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001. 2

[120] G. M. M. Ventola, T. M. R. Noviello, S. D'Aniello, A. Spagnuolo, M. Ceccarelli, e L. Cerulo. Identification of long non-coding transcripts with feature selection: a comparative study. *BMC bioinformatics*, 18(1):187, 2017. 4, 19, 53, 66, 67, 77

[121] L. M. Vieira. Machine Learning based Methods to discriminate long intergenic non-coding RNAs from protein coding transcripts. Dissertação (Mestrado), Universidade de Brasília, 2017. in progress. 90, 91

[122] S. Vinga. Information theory applications for biological sequence analysis. *Briefings in Bioinformatics*, page bbt068, 2013. 36

[123] P. J. Volders, K. Helsens, X. Wang, B. Menten, L. Martens, K. Gevaert, J. Vandesompele, e P. Mestdagh. LNCipedia: a database for annotated human lncRNA transcript sequences and structures. *Nucleic Acids Research*, 41(D1):D246–D251, 2013. 35

[124] L. Wang, H. J. Park, S. Dasari, S. Wang, J. P. Kocher, e W. Li. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. *Nucleic Acids Research*, 41(6):e74–e74, 2013. 17, 37, 46

[125] Z. Wang, M. Gerstein, e M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009. 3, 12

[126] J. Watson, F. Crick, e et al. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953. 1, 7, 8

[127] J. D. Watson, T. A. Baker, S. P. Bell, A. Gann, M. Levine, e R. Losicke. *Biologia Molecular do Gene - 7ed.:*. Artmed Editora, 2015. 13

[128] R. F. Weaver. *Molecular Biology*. McGraw-Hill, 2008. 8

[129] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. 25

[130] V. Wucher, F. Legeai, B. Hédan, et al. Feelnc: a tool for long non-coding rna annotation and its application to the dog transcriptome. *Nucleic acids research*, 45(8):e57–e57, 2017. 18

[131] T. Yoshimizu, A. Miroglio, M. A. Ripoche, et al. The H19 locus acts in vivo as a tumor suppressor. *Proceedings of the National Academy of Sciences*, 105(34):12417–12422, 2008. 16

[132] A. Zaha, H. B. Ferreira, e L. M. P. Passaglia. *Biologia Molecular Básica - 5.ed.:*. Artmed Editora, 2014. 13

[133] Z. H. Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012. 22